ÉCOLE DE TECHNOLOGIE SUPÉRIEURE
UNIVERSITÉ DU QUÉBEC

MÉMOIRE PRÉSENTÉ À
L'ÉCOLE DE TECHNOLOGIE SUPÉRIEURE

COMME EXIGENCE PARTIELLE
À L'OBTENTION DE LA
MAÎTRISE EN GÉNIE ÉLECTRIQUE
M. Ing.

PAR
Guillaume CHARLAND-ARCAND

# CONTRÔLE NON LINÉAIRE PAR BACKSTEPPING D'UN HÉLICOPTÈRE DE TYPE QUADROTOR POUR DES APPLICATIONS AUTONOMES

MONTRÉAL, LE 20 AOÛT 2014

**PRÉSENTATION DU JURY**

CE MÉMOIRE A ÉTÉ ÉVALUÉ

PAR UN JURY COMPOSÉ DE :

Ouassima Akhrif, directrice de mémoire
Département de génie électrique à l'École de technologie supérieure

François Gagnon, président du jury
Département de génie électrique à l'École de technologie supérieure

Lyne Woodward, membre du jury
Département de génie électrique à l'École de technologie supérieure

Denis Couillard, examinateur externe
Ultra Electronics TCS

IL A FAIT L'OBJET D'UNE SOUTENANCE DEVANT JURY ET PUBLIC

LE 21 MAI 2014

À L'ÉCOLE DE TECHNOLOGIE SUPÉRIEURE

## REMERCIEMENTS

Ce projet n'aurait jamais été un succès sans l'apport inestimable de plusieurs personnes.

J'aimerais premièrement remercier ma directrice, Ouassima Akhrif pour le support qu'elle m'a fourni durant mes études aux cycles supérieures et pour m'avoir recruté au sein de l'équipe du projet « Launch and Forget ». De plus, elle a fait preuve de beaucoup de flexibilité, pendant toute la durée de mon mémoire, en me permettant de continuer à m'investir dans le club Dronolab.

Je souhaite aussi remercier particulièrement Charles « LTS » Brunelle pour l'aide précieuse qu'il a apportée à ce projet, en effectuant, entre autre, la conception et l'implémentation de l'architecture du système sous ROS.

Il est également important de souligner l'apport inestimable de l'équipe de test, soit Abbas Chamseddine et Monia Mchirgui, pour toutes les exténuantes journées de test à l'extérieur. Je suis également très reconnaissant envers l'apport des membres du projet « Launch and Forget » soit le professeur François Gagnon, Samuel Gagné et Chahine Ben Moussa.

Les tests de vol n'auraient pas eu lieu sans l'appui de Martin Laporte de la compagnie KoptR Image ainsi que Roger Leblanc professeur à l'ÉNA.

Pour finir, je tiens à remercier tous les membres de Dronolab avec lesquels j'ai travaillé durant mes années à l'ÉTS, soit Pier-Marc Rivet, Florent Touchard, Frédéric Morin, Pascal Chiva, Guillaume Dorion, Mourad Dendane, Mukandila Mukandila, Joël Bourbonnais, Carl St-Laurent, Sam-Nicolai Johnston, Jonathan Pierrat, Jean-Philippe Maltais, Émile Abou Nsar, Junior Pierre, Olivier Massé, Dylan Delonglée, Jonathan Colins et Laurent Chagnon.

Pour finir, j'aimerais remercier Olivier Moses pour sa relecture attentive.

# CONTRÔLE NON LINÉAIRE PAR BACKSTEPPING D'UN HÉLICOPTÈRE DE TYPE QUADROTOR POUR DES APPLICATIONS AUTONOMES

Guillaume CHARLAND-ARCAND

## RÉSUMÉ

Le quadrotor est un aéronef faisant partie de la famille des hélicoptères, plus particulièrement de la famille des multirotors. Le quadrotor possède plusieurs caractéristiques (simplicité mécanique, décollage/atterrissage vertical, vol stationnaire, agilité) qui lui procurent plusieurs avantages opérationnels par rapport à d'autres types d'appareils. Cependant, ces caractéristiques proviennent de la dynamique hautement non linéaire, couplée et sous-actionnée du quadrotor, ce qui le rend impossible à commander sans l'action d'un contrôleur. Ce mémoire propose donc de concevoir un contrôleur permettant d'asservir précisément la position du quadrotor dans l'espace. Ce contrôleur pourra ensuite être utilisé pour effectuer des missions autonomes à l'aide d'un quadrotor.

L'application qui nous intéresse dans ce mémoire provient de la problématique de recherche du projet « Launch and Forget : Aerial Relay Node » de l'École de technologie supérieure (ÉTS) en collaboration avec la compagnie Ultra-Electronics. Celle-ci cherche à concevoir et implémenter une loi d'autonavigation innovatrice permettant d'utiliser un drone comme un relais de télécommunication.

L'objectif général du projet de recherche présenté dans ce mémoire est de concevoir un contrôleur non linéaire de type backstepping pour permettre la navigation du quadrotor selon des points de contrôle prédéfinis et d'offrir les fonctionnalités nécessaires pour implémenter la loi d'autonavigation développée dans le cadre du projet « Launch and Forget ». Pour ce faire, une modélisation mathématique du quadrotor a été effectuée. Par la suite, la conception du contrôleur backstepping a été effectuée à partir du modèle. Sa stabilité a ensuite été validée à l'aide de la théorie de stabilité de Lyapunov et à l'aide de la théorie de la stabilité entrées à états (Input to State Stability - ISS). Un estimateur exact de dérivée basé sur un algorithme de mode glissant d'ordre 2 est utilisé lors de la conception du contrôleur qui permet d'évaluer les variations des commandes virtuelles d'ordre élevé.

L'approche de commande choisie est validée en simulation à l'aide du modèle théorique et en pratique à l'aide d'un quadrotor de type Pelican fabriqué par la compagnie Ascending Technologies.

**Mots clés :** Contrôle non linéaire, stabilité de Lyapunov, backstepping, mode glissant d'ordre 2, stabilité entrées à états, quadrotor, drone

# CONTRÔLE NON LINÉAIRE PAR BACKSTEPPING D'UN HÉLICOPTÈRE DE TYPE QUADROTOR POUR DES APPLICATIONS AUTONOMES

Guillaume CHARLAND-ARCAND

## ABSTRACT

The quadrotor aircraft is a class of helicopter, more specifically of multirotors. The quadrotor has several characteristics (mechanically simple, vertical takeoff and landing, hovering capacities, agile) that give it several operational advantages over other types of aircraft. But its beneficts come at a cost : the quadrotor has a highly nonlinear dynamics, coupled and underactuated which makes it impossible to operate without a feedback controller action. This thesis proposes to design a nonlinear backstepping control law to control the exact position of the quadrotor in space. This controller can then be used to perform autonomous missions with a quadrotor.

The application we are interested in this thesis comes from the research project « Launch and Forget : Aerial Relay Node » of the École de technologie supérieure (ÉTS) in partnership with the company Ultra-Electronics. It seeks to develop and implement an innovative autonavigation law allowing the use of a drone as a telecommunication link.

The overall objective of the research project presented in the thesis is to design a backstepping controller for the navigation of a quadrotor according to predefined waypoints and provide the features necessary to implement the autonavigation law developped in the project « Launch and Forget : Aerial Relay Node ». To do this, a nonlinear model of the quadrotor is developped. A backstepping control based on this model is design and its stability is validated using Lyapunov theory and Input State Stability (ISS). An exact derivative estimator based on a second order sliding mode is used in the design of the controller to calculate the virtual control high order derivative.

The controller is validated by simulation and by practical flight tests using the Pelican quadrotor manufactured by the company Ascending Technologies.

**Keywords :** Nonlinear control, Lyapunov stability, backstepping, second order sliding mode, input output stability, quadrotor, unmanned aerial vehicle.

**TABLE DES MATIÈRES**

Page

## LISTE DES TABLEAUX

Page

## LISTE DES FIGURES

Page

## LISTE DES ABRÉVIATIONS, SIGLES ET ACRONYMES

| | |
|---|---|
| UAV | Unmanned Aerial Vehicule |
| ÉTS | École de technologie supérieure |
| LACIME | Laboratoire de communications et d'intégration de la microélectronique |
| CRSNG | Conseil de recherches en science naturelle et en génie du Canada |
| MEMS | Micro Electronic Mechanical System |
| NASA | National Aeronautics and Space Administration |
| GRAPS | General Robotics, Automation, Sensing & Perception |
| UNA | Université Nationale d'Australie |
| DGPS | Differential Global Positioning System |
| ETH | Institut des technologies fédéral suisse |
| MAST | Micro Autonomous System Technologies |
| NED | North, East, Down |
| ISS | Input to State Stability |
| ROS | Robotic Operating System |
| OBC | On-Board Computer |
| Asctec | Ascending Technologies |
| MMR | Multi-Mission Radio |
| Processeur HL | Processeur haut-niveau |
| Processeur LL | Processeur bas-niveau |
| GPS | Global Positioning System |
| DGPS | Differential Positioning System |
| RTCM | Radio Technical Commission for Maritime Services |

| | |
|---|---|
| FCU | Firmware Control Unit (node ROS) |
| MCU | Mobility Control Unit (node ROS) |
| ANS | Advanced Navigation Spatial (node ROS) |
| NL | Network Logistics (node ROS) |
| NAV | Navigation (node ROS) |
| FM | Flight Manager (node ROS) |
| ÉNA | École Nationale d'Aérotechnique |
| LIPO | Lithium-Polymère |
| MIT | Massachusetts Institute of Technology |
| MIMO | Multiple Input Multiple Output |
| IMU | Inertial Measurement Unit |
| LTS | Long Time Support |
| RC | Remote Control |
| ECEF | Earth Centered Earth Fixed |
| SPI | Serial Peripheral Interface Bus |
| INS | Inertial Navigation System |
| LLA | Longitude Latitude Altitude |
| TCP | Transmission Control Protocol |
| IP | Internet Protocol |
| RSSI | Received Signal Strenght Indication |

## LISTE DES SYMBOLES ET UNITÉS DE MESURE

| | |
|---|---|
| $\mathcal{F}_I$ | Repère inertiel |
| $O_I$ | Origine du repère inertiel |
| $\boldsymbol{i}_I$ | Axe *i* du repère inertiel |
| $\boldsymbol{j}_I$ | Axe *j* du repère inertiel |
| $\boldsymbol{k}_I$ | Axe *k* du repère inertiel |
| | |
| $\mathcal{F}_b$ | Repère du quadrotor |
| $O_b$ | Origine du repère du quadrotor |
| $\boldsymbol{i}_b$ | Axe *i* du repère du quadrotor |
| $\boldsymbol{j}_b$ | Axe j du repère du quadrotor |
| $\boldsymbol{k}_b$ | Axe *k* du repère du quadrotor |
| | |
| $\mathcal{F}_h$ | Repère de l'hélice |
| $O_h$ | Origine du repère de l'hélice |
| $\boldsymbol{i}_h$ | Axe *i* du repère de l'hélice |
| $\boldsymbol{j}_h$ | Axe j du repère de l'hélice |
| $\boldsymbol{k}_h$ | Axe *k* du repère de l'hélice |
| | |
| $\boldsymbol{k}$ | Vecteur unitaire $[0 \quad 0 \quad 1]^T$ |
| | |
| $\boldsymbol{\eta}$ | Vecteur de la position exprimé dans $\mathcal{F}_I$ $(m)$ |
| x | Position du quadrotor selon l'axe $\boldsymbol{i}_I$ $(m)$ |
| $y$ | Position du quadrotor selon l'axe $\boldsymbol{j}_I$ $(m)$ |
| $z$ | Position du quadrotor selon l'axe $\boldsymbol{k}_I$ $(m)$ |
| | |
| $\boldsymbol{\mu}$ | Vitesse linéaire exprimée dans $\mathcal{F}_I$ $(m/s)$ |
| $\boldsymbol{v}$ | Vecteur de la vitesse linéaire exprimée dans $\mathcal{F}_b$ $(m/s)$ |
| $u$ | Vitesse selon l'axe $\boldsymbol{i}_b$ $(m/s)$ |

| | |
|---|---|
| $v$ | Vitesse selon l'axe $\boldsymbol{j}_b$ $(m/s)$ |
| $w$ | Vitesse selon l'axe $\boldsymbol{k}_b$ $(m/s)$ |
| $\boldsymbol{\omega}$ | Vecteur des vitesses angulaires du repère $\mathcal{F}_b$ par rapport à $\mathcal{F}_I$ exprimées dans $\mathcal{F}_b$ $(rad/s)$ |
| $p$ | Vitesse angulaire autour de l'axe $\boldsymbol{i}_b$ $(rad/s)$ |
| $q$ | Vitesse angulaire autour de l'axe $\boldsymbol{j}_b$ $(rad/s)$ |
| $r$ | Vitesse angulaire autour de l'axe $\boldsymbol{k}_b$ $(rad/s)$ |
| $\boldsymbol{f}, \boldsymbol{f}^b$ | Vecteur des forces exprimé dans $\mathcal{F}_b$ $(N)$ |
| $\boldsymbol{f}^I$ | Vecteur des forces exprimé dans $\mathcal{F}_I$ $(N)$ |
| $\boldsymbol{\tau}, \boldsymbol{\tau}^b$ | Vecteur des couples exprimé dans $\mathcal{F}_b$ $(N)$ |
| $\boldsymbol{\tau}^I$ | Vecteur des couples exprimé dans $\mathcal{F}_I$ $(N)$ |
| $\boldsymbol{G}_r$ | Vecteur de précession gyroscopique exprimé dans $\mathcal{F}_b$ $(Nm)$ |
| $\boldsymbol{\Theta}$ | Vecteur des angles d'Euler $(rad)$ |
| $\phi$ | Roulis $(rad)$ |
| $\theta$ | Tangage $(rad)$ |
| $\psi$ | Lacet $(rad)$ |
| $\boldsymbol{\rho}(\boldsymbol{\Theta})_b^I, \boldsymbol{\rho}$ | Matrice de propagation des angles d'Euler $\mathcal{F}_b \rightarrow \mathcal{F}_I$ |
| $\boldsymbol{R}_b^I(\boldsymbol{\Theta}), \boldsymbol{R}$ | Matrice de rotation $\mathcal{F}_b \rightarrow \mathcal{F}_I$ |
| $\boldsymbol{I}$ | Tenseur d'inertie du quadrotor exprimé dans $\mathcal{F}_b$ $(kg \cdot m^2)$ |
| $I_{xx}$ | Inertie selon l'axe x $(kg \cdot m^2)$ |
| $I_{yy}$ | Inertie selon l'axe y $(kg \cdot m^2)$ |
| $I_{zz}$ | Inertie selon l'axe z $(kg \cdot m^2)$ |
| $m$ | Masse du quadrotor $(kg)$ |
| $g$ | Accélération gravitationnelle $(m/s^2)$ |

| | |
|---|---|
| $T$ | Force de poussée totale générée par les moteurs ($N$) |
| $\Omega_i$ | Vitesse de rotation du moteur *i* ($rad/s$) |
| $T_i$ | Force de poussée du moteur *i* ($N$) |
| $D_i$ | Moment de traînée du moteur *i* ($Nm$) |
| $I_m$ | Moment d'inertie du moteur ($kg \cdot m^2$) |
| | |
| $\lambda, \beta$ | Gain de l'estimateur de mode glissant d'ordre 2 |
| $l$ | Distance du centre de gravité à l'application de la force de portance ($m$) |
| $b$ | Coefficient linéaire de poussée ($Ns^2$) |
| $d$ | Coefficient linéaire de traînée ($Nms^2$) |
| $\Pi_1$ | Sous-système représentant la dynamique de positionnement |
| $\boldsymbol{X}_1$ | Vecteur des états du sous-système $\Pi_1$ |
| | |
| $\Pi_2$ | Sous-système représentant la dynamique angulaire |
| $\boldsymbol{X}_2$ | Vecteur des états du sous-système $\Pi_2$ |
| | |
| $\boldsymbol{\eta}_d(t)$ | Vecteur de la trajectoire de position désirée ($m$) |
| $x_d(t)$ | Trajectoire de la position en x désirée ($m$) |
| $y_d(t)$ | Trajectoire de la position en y désirée ($m$) |
| $z_d(t)$ | Trajectoire de la position en z désirée ($m$) |
| $\psi_d(t)$ | Cap désiré ($rad$) |
| | |
| $\boldsymbol{\Lambda}_1$,$\boldsymbol{\Lambda}_2$,$\boldsymbol{\Lambda}_3$,$\boldsymbol{\Lambda}_4$ | Matrice de gain |
| | |
| $\boldsymbol{\alpha}_\mu$ | Vecteur de la commande virtuelle de la vitesse linéaire ($m/s$) |
| $\boldsymbol{\alpha}_\Theta$ | Vecteur de la commande virtuelle de la position angulaire ($m/s$) |
| $\alpha_\phi$ | Commande virtuelle du roulis ($rad$) |
| $\alpha_\theta$ | Commande virtuelle du tangage ($rad$) |
| $\boldsymbol{\alpha}_\omega$ | Vecteur de la commande virtuelle de la vitesse angulaire ($rad/s$) |

| | |
|---|---|
| $\hat{\dot{\alpha}}_i$ | Estimation de la dérivée de la commande virtuelle de l'estimateur $i \in \{\phi \quad \theta \quad p \quad q \quad r\}$ |
| $\tilde{\dot{\alpha}}_i$ | Erreur d'estimation selon l'estimateur $i \in \{\phi \quad \theta \quad p \quad q \quad r\}$ |
| $V_i$ | Fonction de Lyapunov $i$ |
| $\boldsymbol{\eta}_e$ | Erreur de position linéaire $(m)$ |
| $\boldsymbol{\mu}_e$ | Erreur de vitesse linéaire $(m/s)$ |
| $\boldsymbol{\theta}_e$ | Erreur de position angulaire $(rad)$ |
| $\boldsymbol{\omega}_e$ | Erreur de vitesse angulaire $(rad/s)$ |
| $\boldsymbol{Q}(\boldsymbol{\Theta}_e, \boldsymbol{\alpha}_\Theta)$ | Matrice de séparation de la matrice de rotation |
| $\boldsymbol{S}(\cdot)$ | Matrice antisymétrique |
| $\boldsymbol{M}$ | Matrice de correspondance entre force/moment et les vitesses de moteur |
| $\lambda_{min}(\cdot)$ | Opérateur retournant la plus petite valeur propre d'une matrice |
| $\lambda_{max}(\cdot)$ | Opérateur retournant la plus grande valeur propre d'une matrice |

# INTRODUCTION

Le domaine des drones (Unmanned Aerial Vehicule - UAV) est en constante évolution depuis les débuts de l'aviation. À l'origine, les recherches dans ce domaine étaient principalement motivées par des applications militaires (Valavanis, 2007). En effet, les drones étaient, et demeurent aujourd'hui, la meilleure solution pour éviter la perte de pilotes lors de missions dangereuses. Cependant, compte tenu des complexités additionnelles inhérentes aux drones, le développement de ce domaine s'est effectué plus lentement que pour les systèmes avec pilote. L'apparition de capteurs de plus en plus précis, l'augmentation constante de la puissance de calcul des processeurs ainsi que l'avancement des connaissances dans le domaine de l'aéronautique ont permis à cette tendance de s'inverser. En effet, le domaine des drones croît de manière exponentielle depuis le début des années 80 (Newcome, 2004). L'utilisation de drones est désormais monnaie courante dans plusieurs domaines d'applications telles que l'arpentage, la surveillance de pipeline et la photographie aérienne (Austin, 2010). Aussi, les drones sont désormais très connus du public spécialement depuis l'intensification de l'utilisation de drones militaires, tels que le désormais célèbre Predator, par l'armée américaine.

Compte tenu du potentiel énorme des drones pour des applications civiles, un très grand nombre de compagnies concevant des drones civils sont apparues depuis les années 90 telles que, par exemple, Aeryon Labs et Aerodreams. En effet, pour la même application, les drones civils sont typiquement beaucoup moins volumineux que des véhicules avec pilote. Ceci permet de réduire drastiquement le coût de carburant, de fabrication, d'opération ainsi que de maintenance des appareils. Par ailleurs, l'apparition de capteurs bon marché, la multiplication de communautés hébergées sur internet portant sur les drones légers ainsi que l'apparition de projets à code libre sont des facteurs qui ont rendu accessible le domaine des drones à une très grande partie de la population.

De ce fait, il existe désormais une très grande variété de drones de différentes classes. Les UAV sont typiquement classés selon plusieurs critères tels que leur poids, leur taille, leur

rayon d'action, leur altitude de vol et leur endurance. La Figure 1.1 présente quelques modèles de drone connus sur le marché et classés par poids.

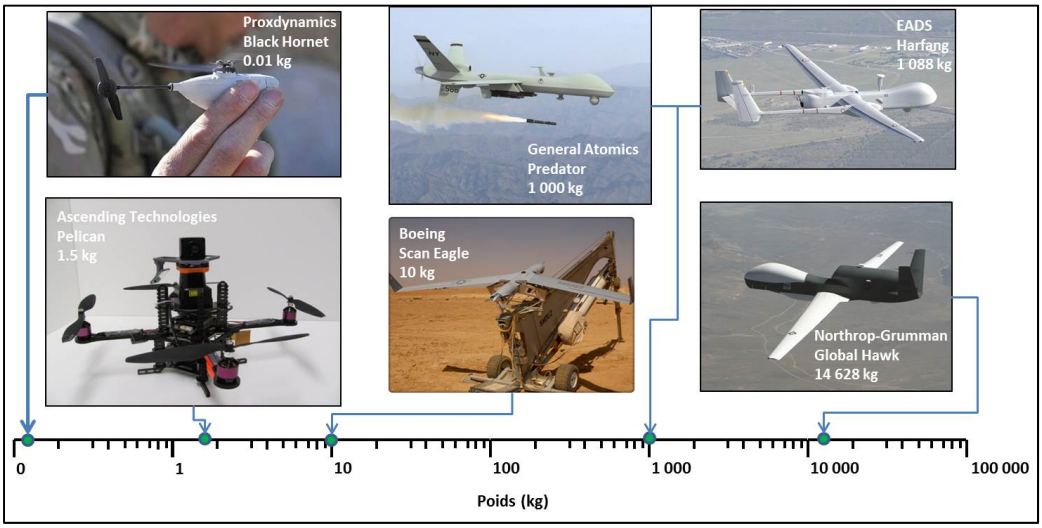



Figure 1.1 Différents modèles de drones classés par poids
Adaptée de Wikipédia (2014)

Le quadrotor, dont le contrôle fait l'objet de ce mémoire, fait partie de la famille des hélicoptères, plus particulièrement de la famille des appareils multirotors. Comme son nom l'indique, il s'agit d'un appareil qui possède quatre moteurs construits en forme de croix. La Figure 1.2 présente un prototype de quadrotor construit par le club scientifique Dronolab de l'École de technologie supérieure (ÉTS).

Figure 1.2 Quadrotor du club étudiant Dronolab de l'ÉTS en vol
Tirée de Dronolab (2014)

Comparativement à l'hélicoptère conventionnel où le contrôle de l'appareil est effectué en changeant l'angle d'incidence du rotor principal, le quadrotor est contrôlé par la variation de vitesse entre les différents moteurs. Ceci simplifie grandement la fabrication de ce type d'appareil en éliminant les pièces mécaniquement complexes constituant notamment le système de propulsion. Malgré ces différences mécaniques, le quadrotor conserve toutes les caractéristiques d'un hélicoptère conventionnel, ce qui lui permet d'être rapidement et facilement déployé sur une multitude de terrains. Il peut également effectuer des vols dans des environnements restreints tels que des bâtiments et maintenir un vol stationnaire. Les avantages et les désavantages du quadrotor sont résumés par le Tableau 1.1[1].

Tableau 1.1 Résumé des avantages et des désavantages du quadrotor

| **Avantages** | **Désavantages** |
|---|---|
| Décollage/Atterrissage vertical | Naturellement instable |
| Vol stationnaire | Dynamique couplée |
| Simple mécaniquement | Système sous-actionné |
| Agile | Petit rayon d'action ($\approx$ 1 à 3 km) |
| Petite taille | Faible autonomie ($\approx$ 10 à 30 min) |
| | Faible charge utile ($\approx$ 0.1 à 1 kg) |

Une multitude de tâches peuvent être envisageables pour un appareil ayant les caractéristiques précédentes, telles que :

- l'inspection de bâtiments, plantations, structures, lignes électrique, pipelines…;
- opération de déminage;
- exploration de lieux difficiles d'accès ou dangereux;
- arpentage;

[1] Les données du tableau 1.1 sont approximatives et dépendent du modèle du quadrotor

- localisation de feux de forêt;
- photographie aérienne.

**Problématique**

Pour l'ensemble de ces applications, le quadrotor doit être asservi pour être en mesure de demeurer en vol. En effet, un contrôleur robuste doit être conçu afin d'éliminer les principaux désavantages liés à la dynamique du quadrotor, dont principalement sa dynamique instable. Une telle tâche est complexe, car la dynamique du quadrotor offre de nombreux défis compte tenu de sa nature hautement non linéaire, couplée et sous-actionnée.

**Objectifs**

Le premier objectif de ce mémoire est de développer et d'implanter un contrôleur non linéaire de type backstepping permettant à un quadrotor d'effectuer des déplacements précis dans l'espace à partir de points de contrôle, pourvu qu'ils soient dans le domaine d'opération de l'appareil. Le backstepping a été choisi car il s'agit d'un contrôleur moderne, performant et bien adapté pour répondre aux problèmes inhérents à la dynamique sous-actionnée du quadrotor.

Ce projet de maîtrise a été développé dans le cadre du projet de recherche nommé « Launch and Forget : Aerial Relay Node » de l'ÉTS en partenariat avec la compagnie Ultra-Electronics et appuyé par le Conseil de recherches en science naturelle et en génie du Canada (CRSNG). L'objectif de « Launch and Forget » est de concevoir un relais de télécommunication aérien et autonome qui assure une bonne qualité de service et une augmentation de la portée pour des communications tactiques. Pour ce faire, une loi d'autonavigation innovatrice a été développée, utilisant seulement la puissance ainsi que l'angle d'incidence des signaux reçus, pour diriger un drone vers la position optimale au sens des télécommunications. La Figure 1.3 illustre le fonctionnement de la loi d'autonavigation.

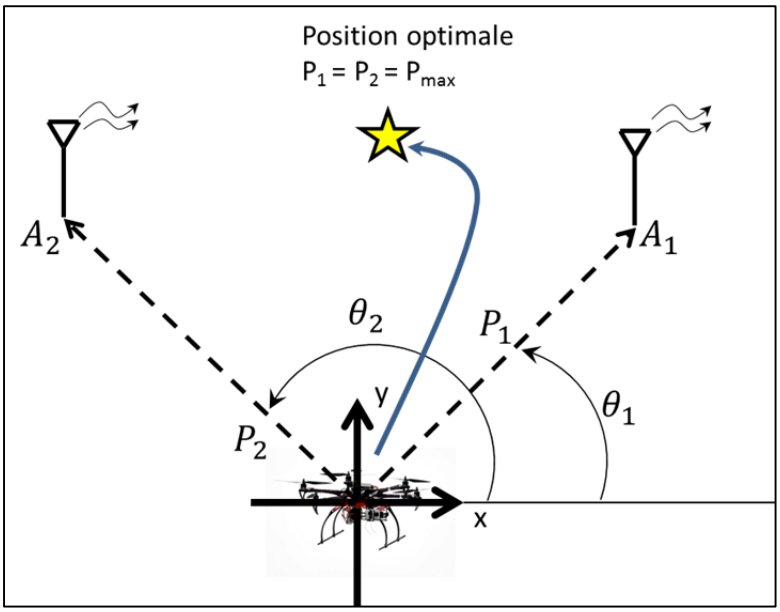



Figure 1.3 Schéma de principe de loi d'autonavigation

Le second objectif de ce mémoire est de concevoir le contrôleur backstepping dans l'optique d'assurer les déplacements demandés par l'algorithme d'autonavigation. Pour ce faire, une attention particulière est donnée aux aspects pratiques lors de la conception du contrôleur pour qu'il soit déployable sur un appareil pour effectuer des tests expérimentaux de la loi d'autonavigation.

Dans le cadre du projet « Launch and Forget », le quadrotor Asctec Pelican, développé par la compagnie Ascending Technologies, a été choisi comme appareil de test. Il s'agit d'un quadrotor de classe « mini », pesant 1kg et possédant une charge utile de 650g.

La Figure 1.4 illustre les principaux systèmes nécessaires pour réaliser l'objectif du projet. La partie encadrée est le sujet de ce mémoire.

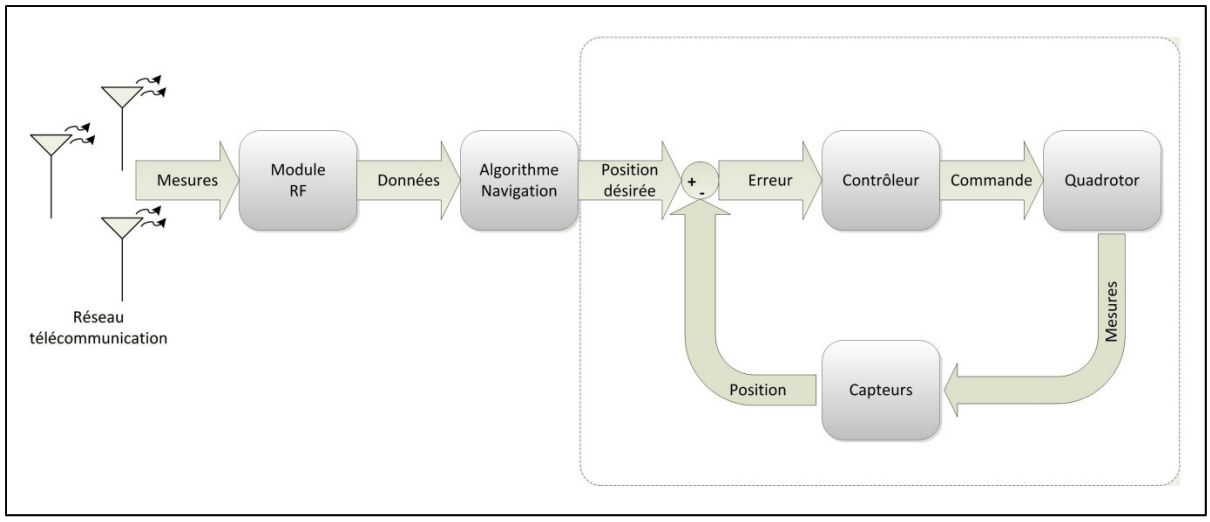



Figure 1.4 Schéma de principe de l'application du quadrotor

**Méthodologie**

Pour parvenir à notre objectif, nous devons effectuer les étapes suivantes :

- la modélisation de la dynamique du quadrotor;
- la conception d'un contrôleur non linéaire de type backstepping;
- le développement d'un modèle de simulation de la dynamique du quadrotor;
- la simulation de la dynamique du quadrotor et du contrôleur;
- l'implémentation du contrôleur au niveau du quadrotor Asctec Pelican;
- la validation expérimentale du contrôleur.

**Contributions scientifiques et techniques**

Les travaux réalisés dans le cadre de ce mémoire ont apporté plusieurs contributions d'ordre scientifique. En effet, la preuve mathématique de stabilité du contrôleur backstepping se distingue de plusieurs auteurs, notamment (Bouabdallah, 2007), par une modélisation complète de la dynamique; c'est-à-dire sans l'application de simplifications aux équations de base de la dynamique. Par ailleurs, la conception du contrôleur considère l'effet de l'erreur

de position angulaire sur la stabilité de la dynamique du quadrotor comparativement à d'autres travaux utilisant la dynamique complète, dont ceux de (Madani, Tarek *et al.*, 2006-2007; Huang, Xian *et al.* 2010)[2]. D'autre part, une majorité d'auteurs utilisent une forme d'estimation[3] pour calculer certaines dérivées de contrôleur virtuel ce qui est également le cas dans ce mémoire. Comparativement aux autres travaux, nous étudions en détail les effets des erreurs d'estimation sur la stabilité du système.

Il est également pertinent de noter que l'expérimentation pratique d'un contrôleur de vol est une tâche extrêmement complexe. En effet, la plupart des travaux portant sur les contrôleurs backstepping possèdent peu ou pas d'expérimentations pratiques (Madani, Tarek *et al.*, 2006-2007; Huang, Xian *et al.* 2010; Bouabdallah, 2007)[4]. Dans le cas où des expérimentations pratiques sont menées, celles-ci sont typiquement effectuées en laboratoires à l'aide de caméra à capture de mouvement mesurant la position du quadrotor avec une précision millimétrique. Le projet MAST du laboratoire GRAPS, le projet « Flying Machine Arena » du professeur Raffeallo D'Andrea et les travaux de (Wang, Raffler *et al.*, 2012) utilisent cette approche.

Plusieurs réalisations d'ordre technique ont été réalisées, dans le cadre de ce mémoire, pour réussir les démonstrations pratiques du contrôleur backstepping ainsi que de la loi d'autonavigation. En effet, comparativement à plusieurs travaux menant leurs essais en environnement contrôlé, nous avons effectué nos tests expérimentaux à l'extérieur. Ceci ajoute plusieurs éléments perturbateurs lors des vols, notamment le vent, et démontre la robustesse de notre approche. De plus, le capteur utilisé pour mesurer la position du quadrotor lors des expérimentations possède une précision significativement inférieure aux

---

[2] L'erreur de position angulaire a un effet direct sur la stabilité du quadrotor. Celle-ci est représentée dans ce mémoire par le terme $\frac{T}{m}\boldsymbol{Q}(\boldsymbol{\Theta}_e, \boldsymbol{\alpha}_\Theta)\boldsymbol{k}$ tel que présenter à la section 4.4.2 et par nos travaux (Ghommam, Charland-Arcand *et al*., 2014).

[3] Par exemple, un estimateur sliding mode d'ordre deux (Madani, Tarek *et al.*, 2006), une dérivée numérique (Madani, Tarek *et al.*, 2007) ou l'utilisation d'un filtre numérique (Wang, Raffler *et al.*, 2012).

[4] Certains de ses travaux ont des résultats expérimentaux, mais seulement concernant le contrôle de l'orientation du quadrotor.

caméras à capture de mouvement typiquement utilisées. Également, l'implémentation numérique et la calibration des gains du contrôleur backstepping ainsi que de l'estimateur de dérivé sont des tâches complexes. En effet, l'optimisation du code implémenté et les choix architecturaux se distinguent notamment par l'utilisation de plusieurs unités de calcul et par sa modularité.

Les travaux générés par ce mémoire et par le projet « Launch and Forget » ont généré plusieurs publications dont : (Ghommam, Charland-Arcand *et al*., 2014), (Chamseddine, Charland-Arcand, Akhrif *et.al*, 2014) et (Charland-Arcand, Akhrif *et.al*, 2014).

Ce mémoire est divisé en 6 parties :

Le **CHAPITRE 1** porte sur la revue de littérature des projets importants portant sur la modélisation et le contrôle de quadrotors.

Le **CHAPITRE 2** porte sur la modélisation de la dynamique du quadrotor. Celle-ci est effectuée en utilisant les lois de mouvement de Newton. On y présente aussi les principales forces et les principaux moments appliqués sur le quadrotor.

Le **CHAPITRE 3** effectue un résumé des concepts de commande non linéaire nécessaire pour effectuer la conception du contrôleur. On y présente la méthodologie de conception du backstepping. On y aborde aussi le concept de stabilité entrées à états ainsi que le fonctionnement d'un estimateur de dérivé basé sur un algorithme par mode glissant d'ordre 2.

Le **CHAPITRE 4** porte sur la conception du contrôleur backstepping. Le contrôleur est divisé en deux étapes, soit la conception du contrôleur de position et le contrôleur d'attitude. La preuve de la stabilité asymptotique du système est effectuée en se basant sur la théorie de Lyapunov.

Le **CHAPITRE 5** porte sur l'implémentation du contrôleur ainsi que du système de vol. Il présente les caractéristiques du quadrotor Asctec Pelican, les capteurs utilisés ainsi que l'architecture logicielle utilisée.

Le **CHAPITRE 6** porte sur les expérimentations du contrôleur. Dans un premier temps, nous présentons les résultats obtenus à l'aide du modèle de simulation pour plusieurs scénarios. Dans un deuxième temps, nous présentons les résultats obtenus à partir des essais de vol expérimentaux.

Nous finissons le mémoire avec une conclusion et des recommandations pour les projets futurs.

# CHAPITRE 1

# REVUE DE LITTÉRATURE

L'avancement technologique, notamment dans le domaine des centrales inertielles à faible coût basées sur la technologie des microsystèmes électromécaniques (MEMS – Micro Electronic Mechanical System), a permis au début des années 2000 à plusieurs laboratoires, de concevoir leur prototype de quadrotor. Compte tenu de son faible coût, de sa simplicité mécanique ainsi que de sa dynamique non linéaire complexe, sous-actionnée et couplée, le quadrotor est rapidement devenu un sujet de recherche populaire dans le domaine de la robotique ainsi que de la commande. L'apparition de compagnies telles que Ascending Technologies et Draganfly spécialisées dans la fabrication de quadrotors dédiés à la recherche n'a fait qu'accroître cet engouement.

Cette popularité fait en sorte que la littérature sur le sujet est vaste et élaborée. En effet, presque toutes les stratégies de commandes ont été appliquées par divers auteurs pour contrôler ce type d'appareil. Cependant, un grand nombre de travaux se base sur des modèles théoriques et ne présentent que des résultats de simulation. Dans un premier temps, nous présenterons ici une liste de projets qui nous paraissent les plus significatifs dans le domaine. (Bouabdallah, 2007), (Bresciani, 2008) ainsi que (Costa de Oliveira, 2011) présentent, dans leur ouvrage respectif, une liste détaillée exhaustive. Les projets sont présentés en ordre chronologique et comportent autant des résultats sur l'implémentation de lois de commande que sur la modélisation et la fabrication d'un quadrotor. De plus, la plupart présente des expérimentations pratiques. Le reste de la revue de littérature se consacre aux différentes méthodes de contrôle linéaire et non linéaire appliquées au quadrotor.

## 1.1 Projets importants

### 1.1.1 Mesicopter (1999-2001)

Le projet Mesicopter de l'université de Stanford s'est échelonné de 1999 à 2001 sous la supervision de M.Ilan Kroo. Ce projet, financé par la National Aeronautics and Space Administration (NASA), avait comme thématique principale la conception de quadrotor de très faible taille, ayant environ 10 cm de diamètre. Les objectifs du projet étaient d'analyser les problématiques liées à la conception et la faisabilité de systèmes volants miniatures, de développer des méthodes de fabrication et d'améliorer les connaissances aérodynamiques où la valeur du nombre de Reynolds est petite.

Plusieurs prototypes d'hélices et d'appareils ont été produits durant le projet, cependant aucun n'était capable de soulever le poids de sa source d'énergie. La plupart des prototypes étant plutôt des preuves de concepts (Kroo *et al.*, 2000). Par contre, plusieurs résultats intéressants sur l'analyse des effets aérodynamiques générés par l'hélice, résumés par (Fay, 2001), ont été utilisés par plusieurs auteurs tels que (Bouabdallah, 2007).

### 1.1.2 X4-flyer MARK (2002-)

Le projet MARK de l'université nationale d'Australie (UNA), dirigé par P. Pounds et Robert Manhony a débuté en 2002. Celui-ci consiste en la conception d'un quadrotor de 4kg pouvant soulever jusqu'à 1kg d'équipement. Celui-ci était très lourd comparativement aux autres quadrotors de sa génération. Les objectifs principaux du projet étaient, premièrement, de générer suffisamment de force de portance, à l'aide du groupe de propulsion, pour soulever l'appareil et, deuxièmement, de stabiliser celui-ci en vol.

Pour réaliser ces objectifs, l'équipe de l'UNA a choisi, comme dans le cas du projet Mesicopter, de concevoir un prototype de châssis et d'hélice. Ces recherches, basées sur les théories de commandes linéaires, leur ont permis de tirer plusieurs conclusions, notamment

que l'utilisation d'hélices rigides devrait être privilégiée (Pounds *et al.*, 2002). En effet, celles-ci avancent que la dynamique du quadrotor en déplacement latéral à moyenne et haute vitesse est affectée par un effet appelé de battement d'hélice modélisé par (Prouty, 2002). De plus, les auteurs concluent que la position du centre de gravité de l'appareil a un effet important sur la stabilité. Ils recommandent de le positionner dans le plan formé par les hélices pour augmenter la stabilité tout en éliminant les effets du « blade flapping » (Pounds *et al.*, 2010).

Un contrôleur proportionnel intégral dérivé (PID) est proposé pour contrôler l'attitude dans des conditions près du vol stationnaire. Ils soulignent que ce type de contrôleur est intéressant compte tenu de sa simplicité et de sa robustesse face aux variations de paramètres. Ils obtiennent de bons résultats en régime stationnaire ou quasi stationnaire avec des erreurs de ±1˚. Il est noté par les auteurs que l'élément limitant le plus la dynamique du prototype est la performance des moteurs (Pounds *et al.*, 2010).

Le projet a été le sujet de plusieurs publications. Le projet est toujours actif comme en fait foi la conception de leur 3e prototype, le MARK III.

### 1.1.3 OS4 (2004-2007)

Le projet OS4 de Samir Bouabdallah porte sur la conception et le contrôle de quadrotors. Ce projet est documenté par plusieurs articles rédigés pendant le projet ainsi que par la thèse de doctorat de M. Bouabdallah présentés à l'École Polytechnique de Lausanne. Celle-ci présente la méthodologie de conception d'un quadrotor en utilisant les informations provenant du poids total voulu, de la poussée générée par les moteurs ainsi que la capacité de la batterie.
Ce projet se concentre exclusivement sur le contrôle du quadrotor à l'intérieur, dans un environnement contrôlé et à basse vitesse. Ceci lui permet de négliger les effets aéronautiques au niveau du modèle. Bouabdallah simplifie le modèle à l'aide de l'hypothèse des petits angles d'attitude. Celle-ci est vérifiée par différents travaux, incluant le sien. Cette approximation permet de découpler le modèle du quadrotor pour obtenir une forme bien

adaptée pour appliquer plusieurs méthodes de contrôle. Ce modèle fut largement utilisé dans la littérature par la suite.

La thèse de Bouabdallah se concentre plus spécifiquement sur la comparaison de la performance de différentes méthodologies de commande appliquées au quadrotor.

Dans un premier temps, la performance de deux contrôleurs linéaires, soit le PID et LQ, est analysée. Il note que le contrôleur et la linéarisation du modèle fonctionnent grâce à la boucle interne contrôlant la vitesse des moteurs. Ceci permet à l'auteur de conclure qu'une dynamique rapide des moteurs est nécessaire au bon fonctionnement du quadrotor. Il note que le PID fonctionne bien autour du point d'équilibre, mais réagit mal en présence de perturbation. Pour ce qui est du contrôleur LQ, il indique que celui-ci donne des résultats moyens et qu'il est moins performant que le PID. Cependant, il note que Hoffman obtient de meilleurs résultats avec ce type de contrôleur (Hoffman *et al.*, 2004).

Dans un deuxième temps, Bouabdallah se penche sur la conception de contrôleurs non linéaires. Il en choisit trois : soit la méthode « Lyapunov Redesign », le contrôle en mode glissant et le backstepping. La méthode de « Lyapunov Redesign » est testée sur un banc d'essai où il obtient de bons résultats, principalement pour le contrôle du lacet. Malgré de bons résultats, les performances du contrôleur en mode glissant sont affectées par l'effet de commutation typiquement relié à ce type de contrôleur. Pour finir, il obtient d'excellents résultats à l'aide du backstepping, et cela, même pour des conditions initiales critiques. Il note aussi que le contrôleur effectue un très bon rejet des perturbations.

Suite à l'analyse de toutes ces méthodes, Bouabdallah conclut que la robustesse de l'action intégrale du PID face aux incertitudes du modèle et la capacité du contrôleur backstepping d'éliminer les perturbations sont les deux éléments essentiels pour obtenir un bon contrôle. Il propose donc de combiner les deux méthodes pour obtenir un contrôleur de type backstepping intégral. Les résultats obtenus sont bons, ce qui lui permet d'implémenter le contrôleur pour tous les degrés de liberté (Bouabdallah, 2007). Le contrôle de position selon

les axes x et y ne fut expérimenté qu'en simulation seulement. Cependant, des sonars ont été installés aux quatre extrémités des bras, permettant d'implémenter un algorithme d'évitement d'obstacle. L'auteur souligne qu'à sa connaissance, il s'agissait du premier quadrotor avec cette capacité.

### 1.1.4 STARMAC (2004-2012)

Le projet STARMAC de l'université de Stanford, dirigé par M.Hoffmann, a comme but l'analyse et le contrôle de quadrotor à l'extérieur dans une enveloppe de vol plus large que le vol stationnaire. L'équipe de M.Hoffmann a elle-même fabriqué un prototype multiagent, très flexible pouvant transporter une multitude de capteurs. Étant notamment équipé d'un Differential Global Positioning System (DGPS), ce quadrotor a permis au projet d'expérimenter sur le contrôle de la position ainsi que sur la génération et la poursuite de trajectoire (Hoffmann *et al.*, 2007, 2009).

Les premières publications de l'équipe STARMAC (Hoffmann *et al.*, 2004, 2007, 2009) portent principalement sur la modélisation et la conception du quadrotor. Celles-ci concluent tout comme (P.Pounds *et al.,* 2010) que le battement d'hélice (« blade flapping ») a des effets importants sur la dynamique de l'attitude du quadrotor lorsqu'il se déplace à haute vitesse. Comme première approche, un contrôleur PID est proposé pour stabiliser l'attitude avec une marge d'erreur d'environ ±2 à 3°. Un contrôleur PID est aussi proposé pour stabiliser la position. Cependant, les auteurs indiquent que les contrôleurs ne sont fonctionnels que dans des conditions où les effets aérodynamiques sont négligeables.

Les articles suivants portent sur la correction du précédent contrôleur pour le rendre plus robuste. L'ajout des effets aérodynamiques ainsi que la combinaison d'une stratégie d'inversion de dynamique avec le contrôleur PID sont utilisés pour augmenter la performance. Le nouveau contrôleur a été testé en soumettant le quadrotor à des manœuvres agressives telles que des changements de direction brusques (Hoffmann *et al*., 2009). Un contrôleur en mode glissant couplé avec une action intégrale est proposé dans un autre article

pour contrôler l'altitude, car les auteurs notent que celle-ci est la plus sujette aux incertitudes de paramètres ainsi qu'aux perturbations (Waslander *et al.*, 2005).

Le quadrotor de STARMAC a aussi été utilisé pour effectuer des recherches plus avancées sur la planification de trajectoire, l'évitement d'obstacles ainsi que la coopération multidrones (Hoffmann *et al*., 2008) et des vrilles (Gillula *et al.*, 2011).

### 1.1.5 « Flying Machine Arena » (2008-)

Le « Flying Machine Arena » est un projet de l'Institut des technologies fédéral suisse (ETH) mené par Raffeallo D'Andrea. Ce projet est une référence en termes d'équipement et de salle de test pour les expérimentations de drones miniatures. Ceux-ci utilisent des quadrotors commerciaux de type Hummingbird conçu par la compagnie Ascending Technologies. La chambre de test, d'une dimension de $10x10x10m^3$, est entourée par des filets et son plancher est constitué d'immenses coussins pour permettre de protéger les appareils en cas de chute. Le laboratoire est équipé de caméras à capture de mouvement de la compagnie Vicon, ayant une fréquence d'acquisition d'environ 250Hz et ayant une précision de quelques millimètres pour mesurer la position des appareils en vol.

Ce type d'installation a permis le développement de plusieurs projets. Voici quelques exemples : la poursuite de trajectoires complexes et agressives telles que le passage à travers des ouvertures, le vol coopératif coordonné par de la musique, l'exécution de vrilles, l'échange d'une balle de ping-pong entre deux quadrotors et la stabilisation d'un pendule inversé installé sur le quadrotor pour en nommer que quelques un. Chacun des projets de recherche utilise des contrôleurs non linéaires modernes.

### 1.1.6 GRAPS Labs : MAST (2009-)

L'approche du projet Micro Autonomous System Technologies (MAST) du laboratoire GRAPS de l'université de la Pennsylvanie est similaire à celle du projet « Flying Machine

Arena ». En effet, ceux-ci utilisent le même type d'équipement, soit des quadrotors de type Hummingbird ainsi que des caméras Vicon pour obtenir la position du quadrotor. Le MAST dirigé par M.Vijay Kumar se spécialise dans la génération de trajectoires agressives réalisables, ainsi que dans la coopération multiquadrotor. Leurs travaux ont fait l'objet d'une multitude de publications et de vidéos où plusieurs quadrotors effectuent des tâches complexes en équipe, tels que des vols en formation, des passages à travers obstacles ainsi que de jouer dans un orchestre. Le contrôleur conçu pour effectuer les travaux est un contrôleur basé sur la géométrie différentielle.

## 1.2 Littérature portant sur les contrôleurs linéaires

La plupart des contrôleurs linéaires développés pour les quadrotors ainsi que pour les hélicoptères miniatures sont obtenus par la linéarisation d'un modèle simplifié. La linéarisation est habituellement effectuée pour le point d'opération correspondant au vol stationnaire.

Les contrôleurs les plus employés sont le contrôleur PID (Kim H. et Shim D., 2003; Bouabdallah *et al.*,2004; Sadeghzadeh, Ankit *et al.*,2012), le contrôleur LQR (Shin J, Fujiwara *et al.*, 2005; Orsag, Popopat et al, 2010) et le contrôleur $\mathcal{H}_\infty$ (La Civita, Papageorgiou *et al.*, 2006).

La plupart des travaux obtiennent de bonnes performances soit une erreur d'environ ±1˚. Cependant celles-ci se dégradent au fur et à mesure que l'on s'éloigne du point d'opération.

## 1.3 Littérature portant sur les contrôleurs non linéaires

Plusieurs types de contrôleurs non linéaires ont été utilisés dans la littérature pour contrôler un appareil de type quadrotor. Nous présenterons ici quatre méthodologies de contrôle nonlinéaires soit : le contrôle par mode glissant, la linéarisation au sens entrées-sorties, le contrôle géométrique et le backstepping.

### 1.3.1 Contrôleur par mode glissant

Plusieurs auteurs ont réussi à développer un contrôleur par mode glissant (« Sliding mode control ») pour le quadrotor (Xu et Özgüner, 2006; Waslander, Hoffmann et al., 2005; Sharifi, Mirzaei et al. 2010; Bouabdallah et al., 2007). Les auteurs utilisent, soit la dynamique complète du quadrotor ou optent pour un modèle simplifié. Dans tous les cas cependant, la présence de l'effet de commutation inhérente au contrôleur par mode glissant affecte la dynamique générale du quadrotor. La stratégie des différents auteurs est de diminuer cet effet nuisible à l'implémentation du contrôleur en utilisant des variances de la fonction signe, telles qu'une fonction de forme sigmoïde ou une fonction de type saturation.

### 1.3.2 Linéarisation au sens entrées-sorties

Une linéarisation au sens entrées-sorties de la dynamique simplifiée du quadrotor est possible si celle-ci est appliquée en deux boucles distinctes, l'une contrôlant l'attitude du quadrotor tandis que la seconde contrôle la position du quadrotor (Das, Subbarao et al. 2008; Zhou et Zhang, 2010; Benallegue, Mokhtari et al. 2007). Celle-ci est ensuite combinée à un contrôleur PID pour assurer la stabilité du système linéarisé ainsi que la stabilité de la dynamique interne.

### 1.3.3 Contrôle géométrique

Le contrôle géométrique se base sur la géométrie différentielle et la topologie pour concevoir des contrôleurs. La dynamique du quadrotor est membre d'une famille de systèmes nommée groupe euclidien spécial noté $SE(3)$. Cette propriété du système peut être exploitée pour concevoir un contrôleur non linéaire (Lee, Leok et al. 2011).

Aussi, en choisissant judicieusement les entrées et les sorties du système on peut démontrer que la dynamique du quadrotor est plate (« differentially flat »). Cette propriété permet un changement de variable qui facilite la conception du contrôleur (Mellinger et Kumar, 2011)[5].

### 1.3.4 Backstepping

Le backstepping est une méthode bien adaptée pour la dynamique du quadrotor, car celle-ci peut directement être exprimée sous une forme permettant la conception de ce type de contrôleur. Cette particularité rend le backstepping très populaire dans la littérature.

Outre (Bouabdallah et Siegwart, 2007) qui utilisent un modèle simplifié pour concevoir un contrôleur backstepping avec une action intégrale, la plupart des auteurs utilisent le modèle complet. (Madani et Benallegue, 2006, 2007), ont publié une série d'articles où la dynamique du quadrotor est décomposée en trois sous-systèmes interconnectés de telle sorte que le contrôleur développé est de type multientrées multisorties (Multiple Input Multiple Output - MIMO). (Huang, Xian et al. 2010) utilisent la même modélisation que (Madani et Benallegue, 2006, 2007) pour concevoir un contrôleur backstepping adaptatif. Le contrôleur est conçu sans connaître la masse du quadrotor. (Colorado, Barrientos et al., 2010) proposent quant à eux un contrôleur backstepping couplé à la théorie de Serret-Frenet pour améliorer le rejet des perturbations. (Jian, Thomas *et al.*, 2012) utilisent les quaternions unitaires pour exprimer la dynamique du quadrotor et concevoir le contrôleur backstepping.

L'ensemble des contrôleurs backstepping proposés dans la littérature fait face à la même problématique qui lui est inhérente. En effet, puisque la dynamique du quadrotor est complexe, il existe toujours une étape où la dérivation des contrôles virtuels devient analytiquement trop complexe. Les principales solutions pour résoudre ce problème sont soit de calculer numériquement la dérivée (Madani et Benallegue, 2006), l'estimer à l'aide d'un

---

[5] Le choix de Mellinger est de considérer comme entrées la poussée totale ainsi que les trois moments principaux et comme sortie la position ainsi que le lacet du quadrotor.

algorithme basé sur un mode glissant d'ordre 2 (Madani et Benallegue, 2006, 2007) ou d'utiliser un filtre numérique (Jian, Thomas *et al.*, 2012).

Cette revue de littérature nous a permis de présenter une grande variété de projets portant sur les quadrotors ainsi que la majorité des techniques de contrôle utilisée pour asservir ce type d'appareil. Celle-ci nous a également permis d'identifier la méthode de contrôle choisie pour ce mémoire : soit le backstepping.

# CHAPITRE 2

# MODÉLISATION DE LA DYNAMIQUE

Ce chapitre porte sur la modélisation mathématique de la dynamique du quadrotor en se basant sur les équations de mouvement de Newton. Dans un premier temps, nous présentons les repères de référence nécessaires pour exprimer l'orientation du quadrotor. Dans un deuxième temps nous donnons la définition de toutes les grandeurs physiques nécessaires à la modélisation. Pour finir, nous identifions toutes les forces et les moments ayant un impact sur le modèle du quadrotor.

## 2.1 Définition des repères

Pour décrire la position et l'orientation du quadrotor, nous avons besoin de deux repères. Le premier est nommé le repère inertiel. Il s'agit d'un référentiel orthogonal fixe de type galiléen, au sens que celui-ci n'accélère pas et ne tourne pas par rapport à un observateur étant dans un repère inertiel. Ainsi, il s'agit d'un repère dans lequel les lois de Newton s'appliquent (Arnol'd, 1978). Il est identifié par l'indice $I$, soit $\mathcal{F}_I = \{O_I, \boldsymbol{i}_I, \boldsymbol{j}_I, \boldsymbol{k}_I\}$. Une convention typiquement utilisée en aéronautique est le repère North, East, Down (NED)[6], tel que $\boldsymbol{i}_I$ pointe vers le nord géographique, $\boldsymbol{j}_I$ pointe vers l'est et $\boldsymbol{k}_I$ pointe vers le centre de la Terre. Nous considérerons ici que la surface de la Terre est plate[7], donc $\boldsymbol{i}_I$ et $\boldsymbol{j}_I$ sont parallèles au sol et $\boldsymbol{k}_I$ est perpendiculaire au sol. Le repère inertiel est de type main droite, tel que $\boldsymbol{i}_I \times \boldsymbol{j}_I = \boldsymbol{k}_I$. La position de l'origine $O_I$ est arbitraire. Typiquement, celle-ci est choisie comme étant la position initiale du quadrotor.

---

[6] Voir la norme iso : ISO 1151-2 :1985

[7] Hypothèse qui peut être validée facilement compte tenu du faible déplacement du quadrotor dans nos applications par rapport au rayon de la Terre.

La définition d'un deuxième repère est nécessaire pour décrire l'orientation du quadrotor[8]. Celui-ci est attaché au châssis du quadrotor et se déplace donc avec celui-ci. Il est dénommé le repère du quadrotor. Celui-ci est identifié par l'indice $b$, soit $\mathcal{F}_b = \{O_b, \boldsymbol{i}_b, \boldsymbol{j}_b, \boldsymbol{k}_b\}$. L'origine $O_b$ coïncide avec le centre de gravité du quadrotor, $\boldsymbol{i}_b$ pointe vers l'avant, $\boldsymbol{j}_b$ pointe vers le bras droit et $\boldsymbol{k}_b$ pointe vers le bas. Le repère du quadrotor est de type main droite, telle que $\boldsymbol{i}_b \times \boldsymbol{j}_b = \boldsymbol{k}_b$.

## 2.2 Décomposition de vecteur selon un repère

Un repère sert de base pour décomposer ou exprimer un vecteur. Les composantes du vecteur sont définies à partir des vecteurs unitaires constituant le repère de référence. Un vecteur peut donc avoir des composantes complètement différentes dépendamment du repère dans lequel celui-ci est exprimé. Puisque la valeur d'un vecteur peut changer d'un repère à l'autre, il est important d'utiliser une notation rigoureuse. Nous identifions un vecteur exprimé dans un repère à l'aide d'un exposant représentant le nom du repère. Par exemple, un vecteur $\boldsymbol{u}$ est exprimé dans le repère inertiel comme $\boldsymbol{u}^I = [u_1^I \quad u_2^I \quad u_3^I]^T$ ou dans le repère du quadrotor comme $\boldsymbol{u}^b = [u_1^b \quad u_2^b \quad u_3^b]^T$. Il est important de constater que les composantes des deux vecteurs ne sont habituellement pas égales ($\boldsymbol{u}^I \neq \boldsymbol{u}^b$). Malgré cela, les deux vecteurs, $\boldsymbol{u}^I$ et $\boldsymbol{u}^b$, représentent la même grandeur physique. Pour ne pas alourdir la notation, certains vecteurs, étant toujours exprimés par rapport au même repère, ne seront pas notés avec un exposant. Le lecteur peut alors se référer à la liste des symboles ou à la définition du vecteur pour connaître le repère dans lequel il est décomposé.

## 2.3 Définition du vecteur de position, de force et de moment

La position du quadrotor est définie par un vecteur allant de $O_I$ à $O_b$ et celui-ci est toujours exprimé dans le repère inertiel. Ce vecteur est noté $\boldsymbol{\eta} = [x \quad y \quad z]^T$. La vitesse linéaire exprimée dans le repère inertiel est notée $\dot{\boldsymbol{\eta}} = [\dot{x} \quad \dot{y} \quad \dot{z}]^T$.

---

[8] Cette notion est abordée à la section 2.4

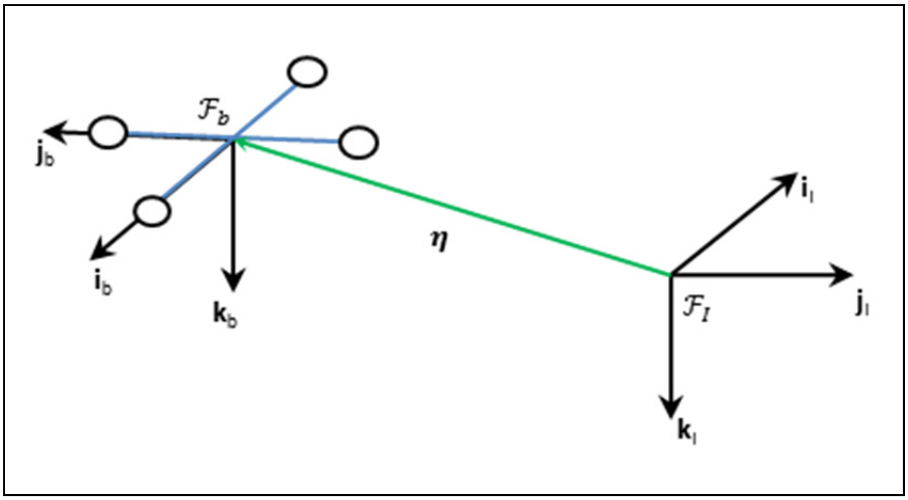



Figure 2.1 Repère $\mathcal{F}_b$ et $\mathcal{F}_I$

En suivant les notations du domaine de l'aérospatial, la vitesse linéaire exprimée dans le repère du quadrotor est notée, quant à elle, $\boldsymbol{v} = [u \quad v \quad w]^T$.

L'ensemble des forces et des moments appliqués au centre du quadrotor, exprimée dans le repère du quadrotor, sont respectivement $\boldsymbol{f} = [f_x \quad f_y \quad f_z]^T$ et $\boldsymbol{\tau} = [\tau_\phi \quad \tau_\theta \quad \tau_\psi]^T$. Le sens positif des moments est défini par la règle de la main droite, aussi appelée « sens trigonométrique »[9].

## 2.4 Matrice de rotation

Une façon d'exprimer l'orientation d'un repère $\mathcal{F}_B$ par rapport à un repère $\mathcal{F}_A$ est de décomposer les vecteurs unitaires formant $\mathcal{F}_B$ par rapport à $\mathcal{F}_A$. La décomposition s'effectue à l'aide d'une projection des vecteurs unitaires de $\mathcal{F}_B$ sur ceux de $\mathcal{F}_A$ à l'aide d'un produit scalaire. Puisque le produit scalaire est basé sur un cosinus, la matrice de rotation est parfois

[9] Lorsque l'axe de rotation pointe vers l'observateur, un moment positif autour de celui-ci est défini par une rotation dans le sens antihoraire.

appelée la matrice des cosinus de direction (Craig, 2005). En organisant les vecteurs résultants comme formant les colonnes d'une matrice, on obtient[10] :

$$\boldsymbol{R}_B^A = \begin{bmatrix} r_{11} & r_{12} & r_{13} \\ r_{21} & r_{22} & r_{23} \\ r_{31} & r_{32} & r_{33} \end{bmatrix} = \begin{bmatrix} \boldsymbol{i}_A \cdot \boldsymbol{i}_B & \boldsymbol{i}_A \cdot \boldsymbol{j}_B & \boldsymbol{i}_A \cdot \boldsymbol{k}_B \\ \boldsymbol{j}_A \cdot \boldsymbol{i}_B & \boldsymbol{j}_A \cdot \boldsymbol{j}_B & \boldsymbol{j}_A \cdot \boldsymbol{k}_B \\ \boldsymbol{k}_A \cdot \boldsymbol{i}_B & \boldsymbol{k}_A \cdot \boldsymbol{j}_B & \boldsymbol{k}_A \cdot \boldsymbol{k}_B \end{bmatrix} \tag{2.1}$$

Il existe trois interprétations physiques de la matrice de rotation $\boldsymbol{R}_B^A$ (DeSantis, 2011; Craig, 2005; Spong, Hutchinson et Vidyasagar 2005; Murray, Li et Sastry, 1994)

- orientation de $\mathcal{F}_B$ par rapport à $\mathcal{F}_A$;
- la rotation que $\mathcal{F}_A$ doit subir pour être orienté comme $\mathcal{F}_B$;
- la matrice qui permet de transformer un vecteur exprimé par rapport à $\mathcal{F}_B$ dans $\mathcal{F}_A$.

La dernière interprétation peut s'écrire mathématiquement ainsi :

$$\boldsymbol{u}^A = \boldsymbol{R}_B^A \boldsymbol{u}^B \tag{2.2}$$

La rotation inverse est obtenue à partir de l'inverse de la matrice de rotation, telle que :

$$\boldsymbol{u}^B = \boldsymbol{R}_B^{A^{-1}} \boldsymbol{u}^A = \boldsymbol{R}_A^B \boldsymbol{u}^A \tag{2.3}$$

Les matrices de rotation ont plusieurs caractéristiques intéressantes tel que :

- elles sont orthogonales. $\boldsymbol{R}\boldsymbol{R}^T = \boldsymbol{I} \leftrightarrow \boldsymbol{R}^T = \boldsymbol{R}^{-1}$;
- $\det(\boldsymbol{R}) = 1$;

[10] La base de décomposition des vecteurs unitaires ($\boldsymbol{i}_A\, \boldsymbol{j}_A\, \boldsymbol{k}_A\, \boldsymbol{i}_B\, \boldsymbol{j}_B\, \boldsymbol{k}_B$) est omise pour la clarté. La matrice $\boldsymbol{R}_B^A$ peut-être obtenue en décomposant ceux-ci soit par rapport à $\mathcal{F}_A$ ou $\mathcal{F}_B$ tant que la décomposition est cohérente pour l'ensemble des vecteurs unitaires.

- chaque colonne ou ligne est un vecteur unitaire;
- toutes les colonnes et toutes les lignes sont mutuellement orthogonales.

Toute matrice qui respecte ces propriétés est considérée comme étant une matrice de rotation et fait partie d'un ensemble de matrices nommé « groupe spécial orthogonal », noté $SO(\mathrm{n})$. Dans notre cas, puisque nous travaillons dans $\mathbb{R}^3$, $\boldsymbol{R} \in SO(3)$ (Bullo et Lewis, 2005; Spong, Hutchinson et Vidyasagar 2005, Murray, Li et Sastry, 1994).

Il existe trois matrices décrivant les rotations élémentaires autour d'un seul axe de rotation. Pour alléger la notation, nous noterons dans ce mémoire : $c_{(\cdot)} = \cos(\cdot)$, $s_{(\cdot)} = \sin(\cdot)$ et $t_{(\cdot)} = \tan(\cdot)$.

$$\boldsymbol{R_x}(\phi) = \begin{bmatrix} 1 & 0 & 0 \\ 0 & c_\phi & -s_\phi \\ 0 & s_\phi & c_\phi \end{bmatrix}$$

$$\boldsymbol{R_y}(\theta) = \begin{bmatrix} c_\theta & 0 & s_\theta \\ 0 & 1 & 0 \\ -s_\theta & 0 & c_\theta \end{bmatrix}$$

$$\boldsymbol{R_z}(\psi) = \begin{bmatrix} c_\psi & -s_\psi & 0 \\ s_\psi & c_\psi & 0 \\ 0 & 0 & 1 \end{bmatrix} \tag{2.4}$$

La combinaison de ces trois rotations permet d'obtenir la matrice de rotation $\boldsymbol{R}_b^I$ pouvant décrire l'orientation du quadrotor dans $\mathbb{R}^3$. Il existe 12 combinaisons de ces matrices pour décrire une orientation. Dans le cadre de ce projet, nous utiliserons le standard utilisé en aéronautique[11] tel que :

$$\boldsymbol{R}_b^I(\phi, \theta, \psi) = \boldsymbol{R_z}(\psi)\boldsymbol{R_y}(\theta)\boldsymbol{R_x}(\phi) \tag{2.5}$$

[11] Voir la norme iso : ISO 1151-2 :1985

$$\boldsymbol{R} \triangleq \boldsymbol{R}_b^I(\phi,\theta,\psi) = \begin{bmatrix} c_\theta c_\psi & s_\phi s_\theta c_\psi - c_\phi s_\psi & c_\phi s_\theta c_\psi + s_\phi s_\psi \\ c_\theta s_\psi & s_\phi s_\theta s_\psi + c_\phi c_\psi & c_\phi s_\theta s_\psi - s_\phi c_\psi \\ -s_\theta & s_\phi c_\theta & c_\phi c_\theta \end{bmatrix} \quad (2.6)$$

Les trois angles, qui sont les paramètres de la matrice $\boldsymbol{R}$, sont appelés les angles d'Euler et seront notés $\boldsymbol{\Theta} = [\phi \quad \theta \quad \psi]^T$. Ce triplet est utile, car ils ont un sens physique. En effet, ils permettent d'exprimer l'orientation du quadrotor dans l'espace.

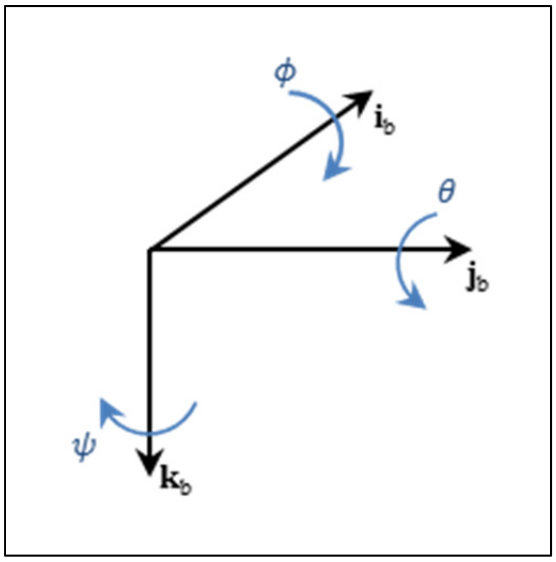



Figure 2.2 Les angles d'Euler

- le roulis ($\phi$) représente une rotation du quadrotor autour de l'axe $\boldsymbol{i}_b$;
- le tangage ($\theta$) représente une rotation du quadrotor autour de l'axe $\boldsymbol{j}_b$;
- le lacet ($\psi$) représente une rotation du quadrotor autour de l'axe $\boldsymbol{k}_b$.

## 2.5 Vitesse angulaire

Tout comme la vitesse linéaire, la vitesse angulaire est représentée par un vecteur. Cependant, contrairement à la vitesse linéaire, la vitesse angulaire ne peut s'appliquer qu'à

un repère[12]. Tout comme l'orientation, la vitesse angulaire d'un repère est exprimée par rapport à un repère de référence. La vitesse angulaire du repère $\mathcal{F}_B$ par rapport au repère $\mathcal{F}_A$ exprimée dans $\mathcal{F}_B$ est notée $\boldsymbol{\omega}_{AB}^{B}$. Celle-ci est reliée à la dérivée en fonction du temps de la matrice de rotation liant l'orientation des deux repères tels que (Craig, 2005; Spong, Hutchinson *et. al*, 2006; Murray, Li *et.al* 1994),

$$\dot{\boldsymbol{R}}_B^A = \boldsymbol{R}_B^A S(\boldsymbol{\omega}_{AB}^B) \tag{2.7}$$

Où $S(\boldsymbol{\omega}_{AB}^{A})$ est la matrice antisymétrique de la vitesse angulaire du repère $\boldsymbol{\omega}_{AB}^{A}$ exprimé dans $\mathcal{F}_A$. Nous introduisons ici l'opérateur $S(\cdot)$ qui transforme un vecteur défini dans $\mathbb{R}^n$ en une matrice antisymétrique membre de la famille des matrices antisymétriques notée $\mathfrak{so}(n)$ telle que pour tout vecteur $\boldsymbol{A}, \boldsymbol{B} \in \mathbb{R}^n$, $S(\boldsymbol{A})\,\boldsymbol{B} = \boldsymbol{A} \times \boldsymbol{B}$. Dans notre application, $S(\boldsymbol{\omega}_{AB}^{A}) \in \mathfrak{so}(3)$.

Pour simplifier la notation, nous noterons la vitesse angulaire du quadrotor par rapport au repère inertiel $\mathcal{F}_I$ exprimé dans le repère du quadrotor $\mathcal{F}_b$ par $\boldsymbol{\omega} \triangleq \boldsymbol{\omega}_{Ib}^{b} = [p \quad q \quad r]^T$. Ainsi, nous pouvons réécrire l'équation (2.7), comme :

$$\dot{\boldsymbol{R}} = \boldsymbol{R} S(\boldsymbol{\omega}) \tag{2.8}$$

## 2.6 Matrice de propagation des angles d'Euler

Maintenant que nous avons défini la vitesse angulaire et sa relation avec la dérivée par rapport au temps de la matrice de rotation, il nous intéresse de connaître la relation entre la vitesse angulaire et la variation des paramètres de la matrice de rotation; c'est-à-dire avec la variation des angles d'Euler. En effet, tel que nous le verrons dans le CHAPITRE 4 portant sur la conception du contrôleur, l'état qui nous intéresse n'est pas la vitesse angulaire

[12] En effet, un point dans l'espace effectuant une rotation n'a pas beaucoup de signification.

exprimée dans le repère du quadrotor, mais bien la position angulaire du quadrotor ainsi que sa variation dans le temps.

Tel que définis à la section 2.4, les angles d'Euler sont notés $\boldsymbol{\Theta} = [\phi \quad \theta \quad \psi]^T$ et leur dérivée par rapport au temps est donc $\dot{\boldsymbol{\Theta}} = [\dot{\phi} \quad \dot{\theta} \quad \dot{\psi}]^T$. En se basant sur (Craig, 2005), nous pouvons écrire la matrice de propagation des angles d'Euler comme étant :

$$\boldsymbol{\rho} \triangleq \begin{bmatrix} 1 & s_\phi t_\theta & c_\phi t_\theta \\ 0 & c_\phi & -s_\phi \\ 0 & s_\phi / c_\theta & c_\phi / c_\theta \end{bmatrix} \tag{2.9}$$

$$\dot{\boldsymbol{\Theta}} = \boldsymbol{\rho}(\boldsymbol{\Theta})\boldsymbol{\omega} \tag{2.10}$$

Il est important de noter que la matrice résultante de ces transformations n'est pas membre de *SO*(3). De plus, la matrice $\boldsymbol{\rho}$ possède une singularité lorsque $\theta = (2n-1)\frac{\pi}{2}$. Il s'agit d'un état appelé « blocage de cardan ». Dans cette configuration, le système perd un degré de liberté de telle sorte qu'une des rotations ne peut plus être représentée par les angles d'Euler. Il est intéressant de noter que les quaternions ne possèdent pas ce type de singularité.

## 2.7 Équation de mouvement

Il est typiquement convenu de considérer un objet, lors de la modélisation, comme étant un corps rigide. Ceci équivaut à considérer que toutes les particules constituant l'objet se déplacent également lorsque celui-ci subit une force. En d'autres termes, l'objet ne subit aucune déformation. En utilisant cette hypothèse, il est possible de modéliser l'objet comme étant une masse ponctuelle, sans volume, où toute la masse est concentrée et où l'ensemble des forces et des moments sont appliqués. Cette hypothèse est valide pour le quadrotor compte tenu de la rigidité de sa structure. Nous modéliserons donc, le quadrotor comme un corps rigide à six degrés de liberté, car il nécessite au minimum six états pour décrire sa position et son orientation dans l'espace.

À l'aide des définitions de la section 2.4 et 2.5, nous pouvons écrire :

$$\dot{\boldsymbol{\eta}} = \boldsymbol{R}\boldsymbol{v} \tag{2.11}$$

$$\dot{\boldsymbol{R}} = \boldsymbol{R}S(\boldsymbol{\omega}) \tag{2.12}$$

L'équation (2.11) donne la transformation de $\boldsymbol{v}$**,** la vitesse linéaire du quadrotor exprimée dans le repère $\mathcal{F}_b$, en $\dot{\boldsymbol{\eta}}$, la vitesse linéaire exprimée dans $\mathcal{F}_I$. Cette équation prend son importance du fait que la dynamique linéaire sera exprimée dans le repère du quadrotor. L'équation (2.12) provient de la définition de la vitesse angulaire (2.8).

Pour modéliser la dynamique du quadrotor, nous utiliserons ici les lois du mouvement de Newton. Comme nous l'avons présenté dans les sections précédentes, celles-ci ne sont valides que dans le repère inertiel $\mathcal{F}_I$.

La variation de la quantité de mouvements est donnée par :

$$\boldsymbol{f}^I = \frac{d}{dt} m\dot{\boldsymbol{\eta}} \tag{2.13}$$

Cependant, puisque les forces sont plus facilement exprimées dans le repère du quadrotor $\mathcal{F}_b$, nous devons transformer (2.13) en appliquant la matrice de rotation. En insérant (2.11) et puisque la masse du quadrotor est constante, nous obtenons :

$$\begin{aligned} \boldsymbol{R}^T\boldsymbol{f}^I = \boldsymbol{f} &= \boldsymbol{R}^T\left[m\frac{d}{dt}\,(\boldsymbol{R}\boldsymbol{v})\right] \\ &= \boldsymbol{R}^T\left[m\left(\dot{\boldsymbol{R}}\boldsymbol{v} + \boldsymbol{R}\dot{\boldsymbol{v}}\right)\right] \\ &= \boldsymbol{R}^T[m(\boldsymbol{R}\boldsymbol{S}(\boldsymbol{\omega})\boldsymbol{v} + \boldsymbol{R}\dot{\boldsymbol{v}})] \end{aligned} \tag{2.14}$$

$$\dot{\boldsymbol{v}} = -\boldsymbol{S}(\boldsymbol{\omega})\boldsymbol{v} + \frac{1}{m}\,\boldsymbol{f} \tag{2.15}$$

Puisque les forces sont appliquées directement sur le centre de gravité, cette équation n'est pas affectée par les mouvements rotatifs.

Comparativement au mouvement linéaire, où la masse sert de résistance au mouvement, les mouvements rotatifs introduisent le concept d'inertie. Dans le cas d'un mouvement rotatif dans $\mathbb{R}^3$, nous pouvons associer à un objet une matrice antisymétrique, définie positive, appelée « le tenseur d'inertie », noté $\boldsymbol{I}$.

$$\boldsymbol{I} = \begin{bmatrix} I_{\mathrm{xx}} & -I_{\mathrm{xy}} & -I_{\mathrm{xz}} \\ -I_{\mathrm{yx}} & I_{\mathrm{yy}} & -I_{\mathrm{yz}} \\ -I_{\mathrm{zx}} & -I_{\mathrm{zy}} & I_{\mathrm{zz}} \end{bmatrix} \tag{2.16}$$

Chaque terme du tenseur d'inertie dépend de la distribution de la masse de l'objet tel que :

$$\begin{aligned} I_{\mathrm{xx}} &= \iiint_V (y^2 + z^2)\partial m & I_{\mathrm{xy}} &= I_{\mathrm{yx}} = \iiint_V xy\ \partial m \\ I_{\mathrm{yy}} &= \iiint_V (x^2 + z^2)\partial m & I_{\mathrm{xz}} &= I_{\mathrm{zx}} = \iiint_V xz\ \partial m \\ I_{\mathrm{zz}} &= \iiint_V (x^2 + y^2)\partial m & I_{\mathrm{yz}} &= I_{\mathrm{zy}} = \iiint_V yz\ \partial m \end{aligned} \tag{2.17}$$

En observant (2.17), nous pouvons conclure que l'expression de $\boldsymbol{I}$ par rapport à $\mathcal{F}_b$ est toujours constante, car la distribution de la masse du quadrotor ne peut pas changer par rapport à ce repère puisque celui-ci est un corps rigide. Nous pouvons donc écrire $\boldsymbol{I} \triangleq \boldsymbol{I}^b$. Le tenseur d'inertie par rapport à $\mathcal{F}_I$ peut-être, quant à lui, noté ainsi (Murray, Li et Sastry, 1993) :

$$\boldsymbol{I}^I = \mathbf{R}\boldsymbol{I}\boldsymbol{R}^T \tag{2.18}$$

La dynamique des mouvements rotatifs est donnée par la variation du moment angulaire tel que

$$\boldsymbol{\tau}^I = \frac{d}{dt}(\boldsymbol{I}^I \boldsymbol{\omega}^I) \tag{2.19}$$

Où $\boldsymbol{\omega}^I$ est la vitesse angulaire du quadrotor par rapport à $\mathcal{F}_I$ exprimé dans $\mathcal{F}_I$. Puisque les moments sont plus facilement exprimés par rapport à $\mathcal{F}_b$ et que la matrice d'inertie y demeure constante, nous cherchons à exprimer (2.19) dans $\mathcal{F}_b$ tel que :

$$\begin{aligned} \boldsymbol{R}^T \boldsymbol{\tau}^I &= \boldsymbol{R}^T \frac{d}{dt}(\mathbf{R}\boldsymbol{I}\boldsymbol{\omega}) \\ \boldsymbol{\tau}^b = \boldsymbol{\tau} &= \boldsymbol{R}^T\left(\mathbf{R}\boldsymbol{I}\dot{\boldsymbol{\omega}} + \dot{\mathbf{R}}\boldsymbol{I}\boldsymbol{\omega}\right) \\ &= \boldsymbol{R}^T(\mathbf{R}\boldsymbol{I}\dot{\boldsymbol{\omega}} + \boldsymbol{R}S(\boldsymbol{\omega})\boldsymbol{I}\boldsymbol{\omega}) \\ &= \boldsymbol{I}\dot{\boldsymbol{\omega}} + \boldsymbol{\omega} \times \boldsymbol{I}\boldsymbol{\omega} \end{aligned} \tag{2.20}$$

$$\boldsymbol{I}\dot{\boldsymbol{\omega}} = \boldsymbol{\tau} - \boldsymbol{\omega} \times \boldsymbol{I}\boldsymbol{\omega} \tag{2.21}$$

## 2.8 Forces et moments principaux

Les forces et les moments appliqués au quadrotor proviennent des forces générées par la rotation des hélices, de la précession gyroscopique ainsi que des effets aérodynamiques. Nous présenterons ici chacune des forces et des moments séparément, ensuite nous les appliquerons au modèle du quadrotor et, pour finir, nous les ajouterons aux équations dynamiques.

Puisque plusieurs équations des sections suivantes demandent une analyse utilisant des notions de mécanique des fluides et d'analyse d'hélice qui dépassent le cadre de ce mémoire, nous ne présenterons ici qu'une vulgarisation des principes ainsi que les résultats nécessaires à notre modélisation. L'auteur propose (Fay, 2001; Bouabdallah, 2007; Leishman, 2006; Ioannis et Valavanis, 2011) pour plus détails sur le sujet.

### 2.8.1 Analyse des forces et moments appliqués à un élément d'hélice

L'analyse des forces générées par l'hélice est typiquement effectuée de deux manières. La première consiste à analyser l'hélice comme étant un disque infiniment mince, où il existe une différence de pression entre ses deux surfaces. L'analyse de pression et d'écoulement d'air à partir des lois de conservation de masse et d'énergie nous permet d'obtenir une formulation des forces. La deuxième consiste à décomposer l'hélice en éléments infiniment petits, de faire le traitement sur chacun des éléments et d'appliquer une intégration en fonction de la géométrie de l'hélice (Leishman, 2006; Raptis et Valavanis, 2011).

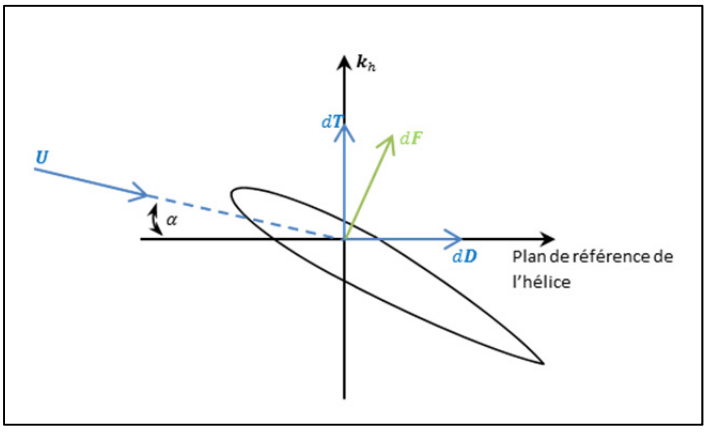



Figure 2.3 Forces appliquées et générées par rapport à une partie élémentaire d'hélice

En appliquant ces deux méthodes, nous pouvons obtenir la formulation de la force $\boldsymbol{dF}$ qui s'applique sur une partie élémentaire de l'hélice. Cette force est ensuite décomposée par rapport au repère de l'hélice, noté $\mathcal{F}_h = \{O_h, \boldsymbol{i}_h, \boldsymbol{j}_h, \boldsymbol{k}_h\}$. Celui-ci est placé dans le plan formé par la rotation de l'hélice où l'axe $\boldsymbol{k}_h$ correspond à l'axe du moteur. Dans notre cas, nous posons que l'axe $\boldsymbol{k}_h$ est parallèle à l'axe $\boldsymbol{k}_b$. L'orientation de l'axe $\boldsymbol{i}_h$ est définie par rapport au vecteur $V_\infty \cos \alpha_{hb}$ qui est la vitesse linéaire du quadrotor par rapport à l'air projetée dans le plan de l'hélice (Ioannis et Valavanis, 2011). Le disque formé par la rotation de l'hélice est situé dans le plan $\boldsymbol{i}_h - \boldsymbol{j}_h$.

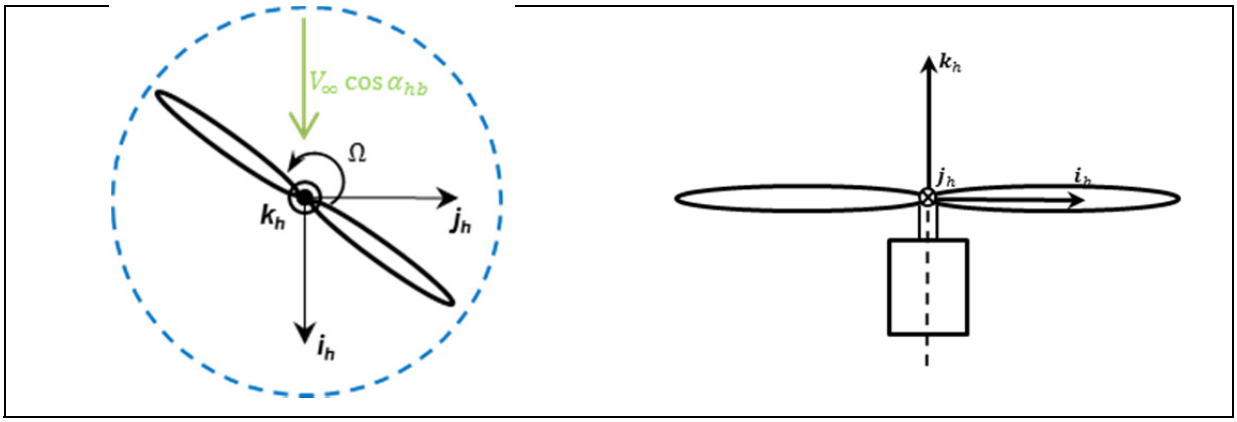



Figure 2.4 Repère de l'hélice $\mathcal{F}_h$

La force $d\boldsymbol{F}$ est décomposée en deux forces perpendiculaires. La première force, nommée « la force de portance » notée $d\boldsymbol{T}$, provient de la force de réaction, au sens de la troisième loi de Newton, générée par la perturbation du flot d'air dû au mouvement rotatif de l'hélice. La seconde, nommée « la force de traînée » notée $d\boldsymbol{D}$, provient de la résistance qu'offre la surface de contact de l'hélice à l'air. Cette dernière dépend de l'angle $\alpha$, nommé « angle d'attaque », entre le plan formé par l'hélice en rotation et la vitesse totale de l'air $\boldsymbol{U}$.

### 2.8.2 Forces de portance

En intégrant $d\boldsymbol{T}$ sur l'ensemble de la géométrie de l'hélice, nous obtenons la force de poussée totale générée par l'hélice (Fay, 2001; Bouabdallah, 2007; Pound *et al.*, 2004) :

$$T = C_T \rho A (\Omega r)^2 \tag{2.22}$$

Où $C_T$ est un terme aéronautique dépendant de la géométrie de l'hélice et des conditions aéronautiques, $\rho$ est la densité de l'air, $A$ est l'aire du disque généré par l'hélice, Ω est la vitesse de l'hélice et $r$ est la longueur de l'hélice. Plusieurs termes de l'équation peuvent être considérés comme étant constants si nous considérons que la vélocité du quadrotor est faible, que le vol est effectué à basse altitude et que l'hélice est suffisamment rigide. En appliquant ces hypothèses, l'équation peut être simplifiée ainsi :

$$T_i = b{\Omega_i}^2 \tag{2.23}$$

Où $b$ est le coefficient de poussée combinant tous les termes constants de l'équation (2.22) et $T_i$ est la force de poussée produite par l'hélice $i$. Cette simplification est généralement effectuée dans la littérature et est confirmée par de nombreuses publications (Bouabdallah, 2007; Costa de Oliveira, 2011; Pound *et al.*, 2004). Pour finir, nous posons que la force de portance générée par une hélice est toujours parallèle aux vecteurs $\boldsymbol{k}_h$ et $\boldsymbol{k}_b$.

### 2.8.3 Moment de traînée

L'hélice en rotation pousse l'air dans une direction spécifique, dans notre cas vers le bas, pour générer la force de portance. Cependant, puisque l'hélice a un certain angle d'attaque par rapport à l'air, une autre force perpendiculaire à la portance est créée.

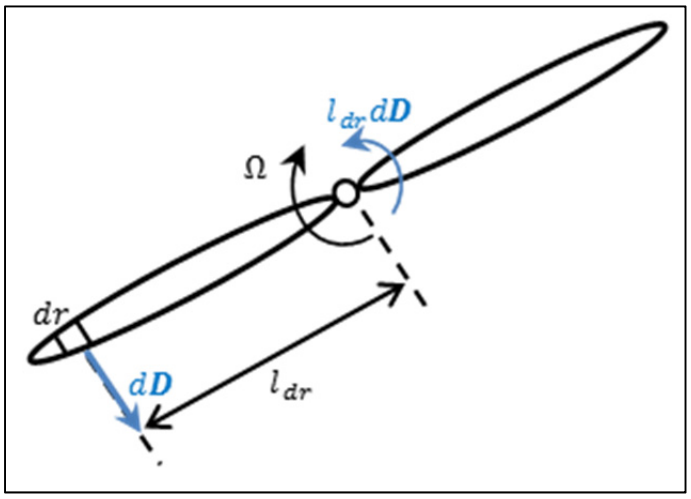



Figure 2.5 Moment de traînée d'une hélice

En faisant l'analyse sur un élément infiniment petit de l'hélice $dr$, nous obtenons un élément de la force de traînée $d\boldsymbol{D}$. Chaque élément de l'hélice génère une force de traînée $d\boldsymbol{D}$ à une distance $l_{dr}$ du centre de rotation. Ceci crée donc un moment $l_{dr} \cdot d\boldsymbol{D}$ par rapport à l'axe $\boldsymbol{k}_h$ dans le sens opposé à la rotation de l'hélice. En prenant la contribution de chaque terme $d\boldsymbol{D}$

en fonction de leur distance $l_{dr}$ sur l'ensemble de la géométrie de l'hélice, nous obtenons (Fay, 2001; Bouabdallah, 2007; Pound *et al.*, 2004) :

$$D = C_D \rho A (\Omega r)^2 r^2 \tag{2.24}$$

Où $C_D$ est un terme aéronautique dépendant des caractéristiques géométriques de l'hélice et des conditions aéronautiques.

En utilisant les mêmes hypothèses qu'à la section précédente, nous simplifions l'équation précédente pour obtenir :

$$D_i = d{\Omega_i}^2 \tag{2.25}$$

Où $d$ est le coefficient de traînée combinant tous les termes constants de l'équation (2.24) et $D_i$ est le moment de traînée généré par l'hélice $i$. Cette simplification est généralement effectuée dans la littérature et est confirmée par de nombreuses publications (Bouabdallah, 2007; Costa de Oliveira, 2011; Pound *et al.*, 2004). Il est important de noter que par définition, l'équation (2.25) génère un moment positif pour les rotations d'hélice antihoraire et un moment négatif pour les rotations horaires, lorsqu'exprimée dans le repère du quadrotor.

### 2.8.4 Précession gyroscopique

La précession gyroscopique est un moment obtenu lorsqu'on applique un couple perpendiculairement à l'axe de rotation d'un objet déjà en mouvement. Le moment gyroscopique généré est orienté de telle sorte qu'il est perpendiculaire au couple appliqué et à l'axe de rotation de l'objet de telle sorte que le trièdre formé par les trois vecteurs respecte la règle de la main droite (Beer et Johnston, 2005).

Plus précisément, ce moment s'applique, dans notre cas, car une hélice, possédant sa propre vitesse de rotation est perturbée par les mouvements rotatifs du quadrotor tel que :

$$\boldsymbol{G}_i = \boldsymbol{\omega} \times I_m \Omega_{\mathrm{i}} \boldsymbol{k}_{\mathrm{h}} \tag{2.26}$$

Où $I_m$ est l'inertie du rotor et $G_i$ est le moment gyroscopique généré par l'hélice $i$.

Il est important de noter que, par définition, on doit appliquer une inversion de signe aux termes de l'équation (2.26) pour les rotations d'hélice dans le sens horaire.

### 2.8.5 Force de gravité

La force de gravité s'applique au quadrotor proportionnellement à sa masse et est toujours perpendiculaire à la surface de la Terre. Exprimée dans le repère inertiel, nous obtenons :

$$\boldsymbol{g} = mg \cdot \boldsymbol{k}_I \tag{2.27}$$

Où $g$ est l'accélération gravitationnelle et est approximativement égale à $9.81\ {}^{m}/_{S}$.

## 2.9 Forces et moments non modélisés

Plusieurs autres effets aérodynamiques qui peuvent perturber le quadrotor ne seront pas modélisés dans le cadre de ce mémoire, car ceux-ci peuvent être négligés à faible vélocité (Bouabdallah, 2007). Nous ne donnerons ici qu'une brève description de ceux-ci. Le lecteur peut se référer à la littérature pour plus de détails sur ces effets.

### 2.9.1 Battement d'hélice

L'effet de battement d'hélice ou « blade flapping » est créé lorsque l'hélice se déplace horizontalement. Ce déplacement crée une différence de vitesse, et donc de poussée, entre la

partie de l'hélice qui attaque le flot d'air par rapport à celle qui se retire du flot d'air. Cette différence de poussée entre ces éléments de l'hélice cause le plan de l'hélice à s'incliner ce qui change la direction du vecteur de poussée.

### 2.9.2 Friction de l'air

Le châssis du quadrotor ainsi que les hélices offrent une résistance à l'air. Celle-ci génère une force de friction qui s'oppose au mouvement linéaire et rotatif du quadrotor. Cette force est proportionnelle au carré de la vitesse du quadrotor et dépend des conditions de l'air ainsi que de la géométrie du quadrotor.

### 2.9.3 Effet de sol

L'effet de sol est créé lorsqu'une surface, qui est suffisamment près de l'hélice, perturbe le flot d'air généré par celle-ci en plus d'améliorer la poussée de l'hélice. Typiquement, cet effet s'applique sur une distance d'environ une demi-fois la longueur de l'hélice (Bouabdallah, 2007), de telle sorte que cet effet s'applique seulement lors du décollage et de l'atterrissage de l'appareil.

### 2.9.4 Instabilité de l'air et vent

Le vent applique une force extérieure sur le quadrotor qui peut influencer la dynamique du quadrotor.

## 2.10 Dynamique des moteurs

Habituellement, les moteurs utilisés pour propulser des appareils de type multirotor sont de la famille des moteurs sans balai. Ceux-ci peuvent être modélisés à l'aide des équations

linéaires classiques des moteurs DC[13]. Il a été validé en pratique que la dynamique du moteur sans balai est beaucoup plus rapide que la dynamique du quadrotor, de telles sortes que celle-ci peut être négligée lors de la modélisation du quadrotor (Bouabdallah, 2007). Nous utilisons, dans ce mémoire, la même hypothèse car elle simplifie la conception du contrôleur.

## 2.11 Modèle du quadrotor

En nous basant sur les résultats des sections précédentes, nous pouvons maintenant écrire le modèle complet du quadrotor. La Figure 2.6 définit la rotation des moteurs pour le quadrotor. Les moteurs sont identifiés dans le sens horaire en considérant le moteur 1 étant celui du devant de l'appareil par rapport à $\mathcal{F}_b$. La paire de moteurs 1-3 tourne dans le sens horaire, tandis que la paire de moteurs 2-4 tourne dans le sens antihoraire.

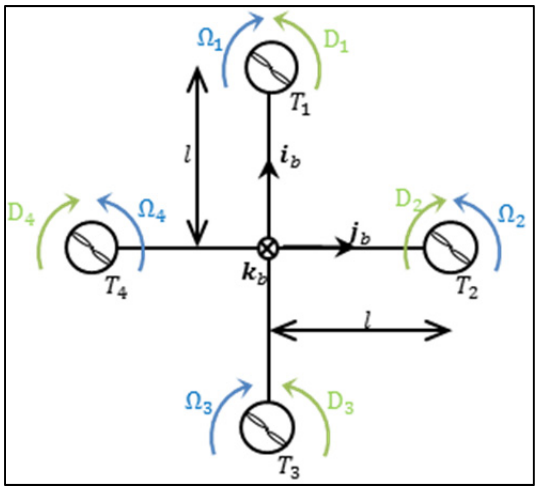



Figure 2.6 Identification du sens de rotation des moteurs

La Figure 2.7 présente une vue de profil du quadrotor dans laquelle nous identifions les deux forces principales agissant sur le quadrotor.

---

[13] Typiquement, on obtient un système linéaire d'ordre deux comprenant la dynamique électrique du moteur et la dynamique mécanique du moteur. Aussi, une simplification souvent effectuée au modèle est de négliger la dynamique électrique, car elle est plus rapide que la dynamique mécanique. La dynamique du moteur est alors exprimée par un système d'ordre un.

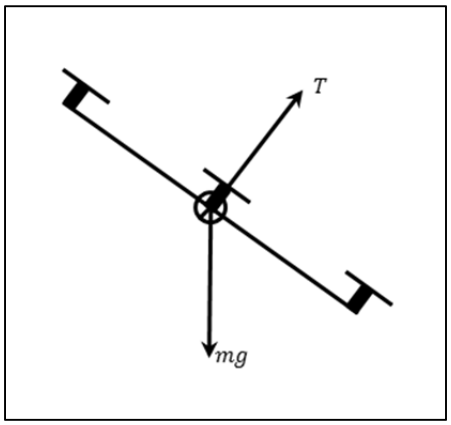



Figure 2.7 Diagramme des forces

En se basant sur les équations (2.23) et (2.27), nous pouvons écrire la somme des forces appliquées au quadrotor, exprimée dans $\mathcal{F}_b$ ainsi :

$$\boldsymbol{f} = \boldsymbol{R}^T mg \cdot \boldsymbol{k} - T\boldsymbol{k} \tag{2.28}$$

Où, $\boldsymbol{k} = [0 \quad 0 \quad 1]^{\boldsymbol{T}}$ est un vecteur unitaire et $T$ est la force totale de poussée telle que :

$$T = b(\Omega_1^2 + \Omega_2^2 + \Omega_3^2 + \Omega_4^2) \tag{2.29}$$

Les moments générés par les hélices proviennent de la différence de poussée générée par chaque paire de moteurs en fonction de $l$, la distance entre le point d'application de la force de poussée et le centre de gravité. Celui-ci se situe au centre géométrique du quadrotor tel qu'illustré par la Figure 2.6 et la Figure 2.7 En se basant sur les équations (2.23) et (2.25) ainsi que sur la Figure 2.6, nous pouvons écrire :

$$\boldsymbol{\tau} = \begin{bmatrix} \tau_\phi \\ \tau_\theta \\ \tau_\psi \end{bmatrix} = \begin{bmatrix} bl(\Omega_4^2 - \Omega_1^2) \\ bl(\Omega_1^2 - \Omega_3^2) \\ d(-\Omega_1^2 + \Omega_2^2 - \Omega_3^2 + \Omega_4^2) \end{bmatrix} \tag{2.30}$$

Le moment gyroscopique total, noté $\boldsymbol{G_r}$, est donné par la somme des moments gyroscopiques définie par (2.26) tel que :

$$\begin{aligned} \boldsymbol{G_r} &= I_m \sum_{i=1}^{4} (\boldsymbol{\omega} \times \boldsymbol{k})(-1)^i \Omega_{\mathrm{i}} \\ &= I_m [p \quad -q \quad 0]^T \Omega_{\mathrm{r}} \end{aligned} \tag{2.31}$$

Où $\Omega_{\mathrm{r}}$ est la vitesse résiduelle des moteurs définie par :

$$\Omega_{\mathrm{r}} = -\Omega_1 + \Omega_2 - \Omega_3 + \Omega_4 \tag{2.32}$$

En combinant l'ensemble des équations de ce chapitre, nous obtenons le modèle complet du quadrotor exprimé par rapport à $\mathcal{F}_b$.

$$\begin{aligned} \dot{\boldsymbol{R}} &= \boldsymbol{R} S(\boldsymbol{\omega}) \\ \dot{\boldsymbol{\eta}} &= \boldsymbol{R} \boldsymbol{v} \\ \dot{\boldsymbol{v}} &= -\boldsymbol{S}(\boldsymbol{\omega}) \boldsymbol{v} + \boldsymbol{R}^T g \cdot \boldsymbol{k} - \frac{T}{m} \boldsymbol{k} \\ \dot{\boldsymbol{\Theta}} &= \boldsymbol{\rho}(\boldsymbol{\Theta}) \boldsymbol{\omega} \\ \boldsymbol{I} \dot{\boldsymbol{\omega}} &= \boldsymbol{\tau} - \boldsymbol{\omega} \times \boldsymbol{I} \boldsymbol{\omega} - \boldsymbol{G_r} \end{aligned} \tag{2.33}$$

## 2.12 Correspondance forces/moments – vitesses de moteur

En se basant sur les équations (2.29) et (2.30), nous pouvons formuler une correspondance permettant de calculer la vitesse des moteurs à partir des forces et des moments appliqués au quadrotor. Cette correspondance est essentielle lors de l'implémentation du contrôleur au niveau de l'appareil.

Nous pouvons réécrire les équations sous la forme matricielle suivante :

$$\begin{bmatrix} T \\ \tau_\phi \\ \tau_\theta \\ \tau_\psi \end{bmatrix} = \begin{bmatrix} b & b & b & b \\ 0 & -bl & 0 & bl \\ bl & 0 & -bl & 0 \\ -d & d & -d & d \end{bmatrix} \begin{bmatrix} \Omega_1^2 \\ \Omega_2^2 \\ \Omega_3^2 \\ \Omega_4^2 \end{bmatrix} \tag{2.34}$$

En inversant cette matrice, nous obtenons la correspondance désirée, notée $\boldsymbol{M}$. Il est intéressant de noter que le déterminant de celle-ci est égal à $8b^3dl^2$ qui est toujours non nul, de telle sorte que l'inversion est toujours valide.

$$\begin{bmatrix} \Omega_1^2 \\ \Omega_2^2 \\ \Omega_3^2 \\ \Omega_4^2 \end{bmatrix} = \begin{bmatrix} 1/(4b) & 0 & 1/(2bl) & -1/(4d) \\ 1/(4b) & -1/(2bl) & 0 & 1/(4d) \\ 1/(4b) & 0 & -1/(2bl) & -1/(4d) \\ 1/(4b) & 1/(2bl) & 0 & 1/(4d) \end{bmatrix} \begin{bmatrix} T \\ \tau_\phi \\ \tau_\theta \\ \tau_\psi \end{bmatrix}$$

$$= \boldsymbol{M} \begin{bmatrix} T \\ \tau_\phi \\ \tau_\theta \\ \tau_\psi \end{bmatrix} \tag{2.35}$$

Dans ce chapitre, nous avons présenté les équations différentielles représentant la dynamique du quadrotor ainsi que les forces et moments générées par le groupe de propulsion. Dans le CHAPITRE 3, nous présenterons les notions nécessaires pour construire le contrôleur backtepping permettant l'asservissement du quadrotor.

# CHAPITRE 3

# NOTIONS PRÉLÉMINAIRES DE COMMANDE NON LINÉAIRE

Dans ce chapitre, nous présenterons les notions nécessaires à la conception du contrôleur non linéaire. Dans un premier temps, la méthode de commande choisie, le backstepping, est abordée et appliquée sur un système non linéaire. Puisque le contrôleur conçu possède la propriété de stabilité entrées à états (ISS), cette notion est décrite, dans un second temps, en plus d'établir les conditions suffisantes pour l'obtenir. Pour finir, le contrôleur backstepping, développé dans ce mémoire, utilise des dérivées d'ordres supérieurs complexes à calculer analytiquement. Pour résoudre ce problème, un estimateur de dérivées basé sur un algorithme de mode glissant d'ordre deux est présenté, dans un troisième temps.

## 3.1 Backstepping

L'une des méthodes de conception des contrôleurs non linéaires se base sur la théorie de la stabilité des systèmes dynamiques d'Alexsandr Lyapunov (Lyapunov, 1992). Typiquement, le but du concepteur est de trouver une fonction définie positive, nommée « la fonction candidate de Lyapunov », dont la dérivée, par rapport au temps, est contrainte, à l'aide des entrées du système, à être une fonction définie négative. Bien entendu, cette tâche est complexe pour un grand nombre de systèmes. Le backstepping est une méthode de conception développée par plusieurs auteurs dont Petar V. Kokotovic[14] et applicable à certaines classes de systèmes, qui normalise la conception du contrôleur en une série d'étapes prédéfinies. Cette stratégie permet de construire au fur et à mesure l'expression de la commande pouvant stabiliser le système.

---

[14] L'origine du backstepping est difficile à retracer, car il est apparu, au départ, dans plusieurs publications indépendantes et ceci souvent de manière implicite (Krstić, Kanellakopoulos et al., 1995).

Pour illustrer le principe, prenons un système mécanique typique pouvant être modélisé par l'équation suivante, selon (Craig, 2005)[15]

$$\boldsymbol{M}(\boldsymbol{x})\ddot{\boldsymbol{x}} + \boldsymbol{C}(\boldsymbol{x}, \dot{\boldsymbol{x}})\dot{\boldsymbol{x}} + \boldsymbol{F}(\boldsymbol{x}, \dot{\boldsymbol{x}}) + \boldsymbol{g}(\boldsymbol{x}) = \boldsymbol{u}$$

Comme nous pouvons le constater, l'entrée du système $\boldsymbol{u}$ agit directement sur l'accélération, tandis que l'on s'intéresse, en général, à contrôler la position. Ceci augmente la difficulté, car il faut alors contrôler le système via deux intégrateurs.

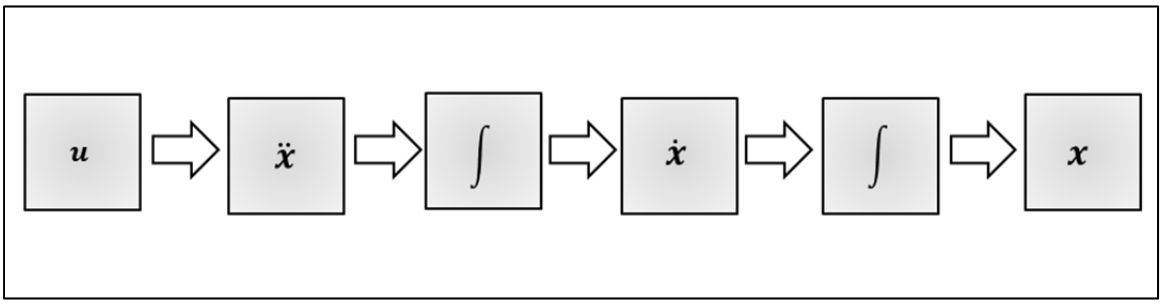



Figure 3.1 Chaîne de contrôle typique d'un système mécanique

Or, le backstepping utilise la relation cinématique entre les états pour simplifier la conception du contrôleur en l'effectuant par construction. On commence la conception du contrôleur au niveau de l'état que l'on désire commander et on remonte (« backstep ») la chaîne d'intégrateurs de telle sorte que la vitesse commande la position, l'accélération commande la vitesse et l'entrée de contrôle commande l'accélération[16].

---

[15] $\boldsymbol{M}(\boldsymbol{x})$ est la matrice d'inertie généralisée, $\boldsymbol{C}(\boldsymbol{x}, \dot{\boldsymbol{x}})$ est la matrice qui regroupe les termes provenant des accélérations centrifuges et des effets de Coriolis, $\boldsymbol{F}(\boldsymbol{x}, \dot{\boldsymbol{x}})$ est une matrice qui regroupe les termes de friction et $\boldsymbol{g}(\boldsymbol{x})$ est le vecteur regroupant les termes d'accélération gravitationnelle et $\boldsymbol{u}$ est le vecteur de commande.

[16] Cette explication se base ici sur une famille particulière de systèmes mécaniques. Comme nous le présenterons dans ce qui suit, ce concept peut s'appliquer à d'autres types de systèmes. Aussi, le lien entre les états n'est pas obligatoirement de type cinématique et peut être généralisé.

### 3.1.1 Description des étapes de conception.

La méthode de conception du contrôleur backstepping se base sur la théorie de stabilité des systèmes dynamiques de Lyapunov. L'essentiel de la théorie nécessaire est présenté dans (Khalil, 1992). Prenons le modèle d'état non linéaire suivant pour illustrer la méthode :

$$\begin{aligned} \dot{x}_1 &= cosx_1 + x_2 \\ \dot{x}_2 &= x_1 + u \end{aligned} \tag{3.1}$$

Le système homogène ($u = 0$) ne possède qu'un seul point d'équilibre à $(x_1, x_2) = (0, -1)$ qui est instable. Nous cherchons un contrôleur $u = \alpha(x_1, x_2)$ qui rend ce point d'équilibre du système globalement exponentiellement stable.

#### 3.1.1.1 Première étape

Concentrons-nous sur la première équation du système (3.1). Si $x_2$ était une entrée du système, alors un choix logique de contrôleur serait :

$$\alpha_{x_2} = -cosx_1 - k_1 x_1 \tag{3.2}$$

Où $k_1$ est un gain strictement positif choisi par le concepteur et où $\alpha_{x_2}$ est la commande virtuelle de $x_2$ de telle sorte que l'origine du système est globalement exponentiellement stable au sens de Lyapunov lorsque $x_2 = \alpha_{x_2}$. Nous pouvons confirmer ceci en choisissant la candidate de Lyapunov suivante :

$$V_1 = \frac{1}{2} x_1^2 \tag{3.3}$$

de telle sorte que sa dérivée par rapport au temps est :

$$\begin{aligned}\dot{V}_1 &= x_1\dot{x}_1 = x_1[cosx_1 - cosx_1 - k_1x_1] \\ &= -k_1x_1^2\end{aligned} \tag{3.4}$$

Cependant, comme $x_2$ n'est pas une entrée du système mais un état, le concepteur ne peut pas directement forcer $x_2$ à être égal à sa commande virtuelle $\alpha_{x_2}$. Néanmoins, le concepteur peut utiliser la seconde équation de (3.1) pour contrôler la variation de $x_2$, soit $\dot{x}_2$.

#### 3.1.1.2 Deuxième étape

Passons à la deuxième étape de conception en définissant l'erreur entre l'état $x_2$ et sa commande virtuelle $\alpha_{x_2}$ comme étant l'état suivant :

$$e = x_2 - \alpha_{x_2} \tag{3.5}$$

Il nous faut maintenant définir un nouveau système à partir de ce nouvel état. On le dénomme typiquement « le système augmenté ». Nous notons également que dans la deuxième étape de conception, l'état $x_2$ n'apparaît plus. Celui-ci est implicitement tenu en compte via l'état d'erreur $e$. Le système augmenté peut s'écrire ainsi :

$$\begin{aligned}\dot{x}_1 &= e - k_1x_1 \\ \dot{e} &= \dot{x}_2 - \dot{\alpha}_{x_2} \\ &= (x_1 + u) - (e - k_1x_1)(sinx_1 - k_1)\end{aligned} \tag{3.6}$$

Soit la fonction candidate de Lyapunov du système augmenté :

$$V_2 = V_1 + \frac{1}{2}e^2 \tag{3.7}$$

$$\begin{aligned}\dot{V}_2 &= x_1\dot{x}_1 + e\dot{e} \\ &= x_1[e - k_1x_1] + e[x_1 + u - (e - k_1x_1)(sinx_1 - k_1)] \\ &= \dot{V}_1 + e[2x_1 + u - (e - k_1x_1)(sinx_1 - k_1)]\end{aligned} \tag{3.8}$$

En choisissant le contrôleur suivant :

$$u = -2x_1 + (e - k_1x_1)(sinx_1 - k_1) - k_2e \tag{3.9}$$

Nous obtenons :

$$\dot{V}_2 = -k_1x_1^2 - k_2e^2 \tag{3.10}$$

Ce qui nous assure que le point d'équilibre $(x_1, x_2) = (0, -1)$ est globalement exponentiellement stable. Notre objectif de contrôle est donc atteint. Le système en boucle fermée a ainsi la forme suivante

$$\begin{aligned}\begin{bmatrix}\dot{x}_1 \\ \dot{e}\end{bmatrix} &= \begin{bmatrix}-k_1 & 1 \\ -1 & -k_2\end{bmatrix}\begin{bmatrix}x_1 \\ e\end{bmatrix} \\ &= \left(\begin{bmatrix}-k_1 & 0 \\ 0 & -k_2\end{bmatrix} + \begin{bmatrix}0 & 1 \\ -1 & 0\end{bmatrix}\right)\begin{bmatrix}x_1 \\ e\end{bmatrix}\end{aligned} \tag{3.11}$$

Il est intéressant de noter que malgré le fait que le système augmenté est de forme linéaire, celui-ci demeure non linéaire. Aussi, la matrice du système résultant de l'application d'un contrôleur backstepping est toujours la somme d'une matrice diagonale négative et d'une matrice antisymétrique.

## 3.2 Classes particulières de systèmes

L'exemple précédent peut se généraliser pour plusieurs classes de systèmes. Les classes qui nous intéressent particulièrement dans notre cas sont les systèmes de type « Strict-feedback »

et « Pure-feedback » tels que définis par (Krstić, Kanellakoupoulos et Kokotović, 1995). Nous présenterons ici la forme générale de chacune des familles de systèmes. (Krstić, Kanellakoupoulos et Kokotović, 1995) fournit une procédure de conception pour chacune des familles.

### 3.2.1 Systèmes « Strict-feedback »

$$\begin{aligned} \dot{x}_1 &= f_1(x_1) + g_1(x_1)x_2 \\ \dot{x}_2 &= f_2(x_1, x_2) + g_1(x_1, x_2)x_3 \\ &\vdots \\ \dot{x}_{n-1} &= f_{n-1}(x_1, x_2, \cdots, x_{n-1}) + g_{n-1}(x_1, x_2, \cdots, x_{n-1})x_n \\ \dot{x}_n &= f_n(x_1, x_2, \cdots, x_n) + g_n(x_1, x_2, \cdots, x_{n-1})u \end{aligned} \tag{3.12}$$

Où $x_1, \cdots, x_n$ sont les états du système et $u$ est l'entrée du système.

La caractéristique principale de cette classe de systèmes est que chaque équation $\dot{x}_i$ dépend seulement des états « précédents », soit $x_1, \cdots, x_{i+1}$. De plus, pour chaque équation $\dot{x}_i$, l'état $x_{i+1}$ est un terme appliqué linéairement et peut être utilisé comme commande virtuelle du système. La méthodologie de conception du contrôleur backstepping, telle que décrite dans la section précédente, est appliquée jusqu'à $\dot{x}_n$, où le contrôle est effectué directement via $u$ qui est l'entrée du système. On obtient alors la forme finale du contrôleur. Ce type de système est aussi appelé « système à forme triangulaire » dans la littérature. Il est intéressant de noter que le système de la section 3.1.1 appartient à cette catégorie.

### 3.2.2 Systèmes « Pure-feedback »

$$\begin{aligned} \dot{x}_1 &= f_1(x_1) + g_1(x_1)x_2 \\ \dot{x}_2 &= f_2(x_1, x_2, x_3) \\ \dot{x}_3 &= f_3(x_1, x_2, x_3, x_4) \\ &\vdots \\ \dot{x}_{n-1} &= f_{n-1}(x_1, x_2, \cdots, x_n) \\ \dot{x}_n &= f_n(x_1, x_2, \cdots, x_n, u) \end{aligned} \tag{3.13}$$

Où, $x_1, \cdots, x_n$, sont les états du système et $u$ est l'entrée du système. Comparativement aux systèmes « Strict-feedback », les systèmes « Pure-feedback » ne possèdent pas les états $x_{i+1}$, entrant linéairement dans les équations $\dot{x}_i$, pour effectuer la commande virtuelle.

L'idée, pour résoudre ce problème, est d'écrire l'erreur ainsi que la commande virtuelle sous la forme d'une fonction des états au lieu d'utiliser l'état directement. Comme nous l'avons présenté à la section 3.1.1.2, la relation entre un état $x_i$ et sa commande virtuelle $\alpha_{x_i}$, dans le cas d'un système « Strict-feedback », est en général

$$e_{x_i} = x_i - \alpha_{x_i}$$

On cherche ensuite une valeur de $\alpha_{x_i}$ pour forcer $\dot{V}_{i+1}$ à être définie négative.

Dans le cas d'un système « Pure-feedback », on cherche à écrire la relation entre la commande virtuelle et son état correspondant à l'aide de fonctions; soit l'erreur entre l'état $x_{i+1}$ et son contrôleur virtuel $\alpha_{x_{i+1}}$ pour la fonction $f_i(x_1, x_2, \cdots, x_{i+1})$. Typiquement, la fonction d'erreur aura la forme suivante :

$$f_{e_i}(x_1, x_2, \cdots, x_{i+1})\left[x_{i+1} - \alpha_{x_{i+1}}\right] = f_i(x_1, x_2, \cdots, x_{i+1}) - f_i\left(x_1, x_2, \cdots, \alpha_{x_{i+1}}\right)$$

On peut constater que les termes des deux fonctions $f_i$ sont les mêmes excepté pour l'état $x_{i+1}$ qui est remplacé par son contrôleur virtuel $\alpha_{x_{i+1}}$. Ceci permet de faire la correspondance entre l'état et son contrôleur virtuel sans qu'il ne soit introduit linéairement dans l'équation. Le but du concepteur est ici plus complexe : il doit trouver une fonction $f_{ei}(x_1, x_2, \cdots, x_{i+1})$ et, dans un même temps, la commande virtuelle $\alpha_{x_i}$ qui, combinées ensemble, stabilisent la dynamique de l'erreur. Cette procédure s'applique $n$ fois jusqu'à ce que le concepteur utilise $u$, l'entrée du système, pour effectuer la commande.

## 3.3 Stabilité entrées à états

Lors de la conception du contrôleur backstepping, nous devons démontrer la stabilité du système face à des perturbations provenant, notamment, de la dynamique sous-actionnée du quadrotor. Une des théories portant sur ce type de stabilité se nomme « la stabilité entrées à états » (ISS). Un système ISS est un système de la forme $\dot{\boldsymbol{x}} = \boldsymbol{f}(t, \boldsymbol{x}, \boldsymbol{u})$ dont l'origine est uniformément asymptotiquement stable lorsque l'entrée $\boldsymbol{u}$ est nulle et dont l'état est borné pour toutes entrées bornées appliquées à celui-ci, tel que :

$$\|\boldsymbol{x}(t)\| \leq \beta(\boldsymbol{x}(t_0), t_0 - t) + \gamma\left(\sup_{t_0 \leq \tau \leq t} \|\boldsymbol{u}(\tau)\|\right), \qquad \forall\, t > t_0 \tag{3.14}$$

où $\beta$ est une fonction classe $\mathcal{KL}$ et $\gamma$ est une fonction classe $\mathcal{K}$. Cette définition peut-être étendue au cas où l'entrée $u$ est considérée comme étant une perturbation appliquée à un système en boucle fermée, ce qui est notre cas (Sontag, 2008; Khalil, 2002).

La stabilité ISS nous permet non seulement de conclure que l'origine du système est stable, soit que l'état est borné, mais également de connaître la valeur de cette borne, puisqu'elle dépend de la grandeur de la perturbation. Une autre caractéristique essentielle de ce type de stabilité c'est qu'il assure la stabilité asymptotique de l'origine face à une perturbation bornée décroissante asymptotiquement vers zéro.

Dans ce qui suit, nous présentons une condition suffisante pour démontrer que l'origine d'un système est effectivement ISS face à une perturbation $\boldsymbol{u}$ donnée ainsi qu'une explication du principe.

### 3.3.1 Condition suffisante

Une condition suffisante que l'origine d'un système $\dot{\boldsymbol{x}} = \boldsymbol{f}(t, \boldsymbol{x}, \boldsymbol{u})$ doit remplir pour être ISS peut être énoncée sous la forme d'un théorème utilisant les concepts de stabilité de Lyapunov,

Soit $V(t, x)$ la candidate de Lyapunov du système telle que :

$$\begin{aligned} &\alpha_1(\|\boldsymbol{x}\|) \leq V(t, \boldsymbol{x}) \leq \alpha_2(\|\boldsymbol{x}\|) \\ &\frac{\partial V}{\partial t} + \frac{\partial V}{\partial \boldsymbol{x}} \boldsymbol{f}(t, \boldsymbol{x}, \boldsymbol{u}) \leq -W(\boldsymbol{x}), \qquad \forall \|\boldsymbol{x}\| \geq \rho(\|\boldsymbol{u}\|) > 0\,, t > 0 \end{aligned} \tag{3.15}$$

Où $\boldsymbol{x} \in \mathbb{R}^n$ est le vecteur d'état et $\boldsymbol{u} \in \mathbb{R}^m$ est le vecteur d'entrée[17], $\alpha_1$ et $\alpha_2$ sont des fonctions classes $\mathcal{K}_\infty$, $\rho$ est une fonction classe $\mathcal{K}$ et $W(\boldsymbol{x})$ est une fonction scalaire définie positive. Si une telle fonction de Lyapunov existe pour le système, alors l'origine de celui-ci est ISS et la fonction $\gamma$ de l'équation (3.14) peut être déduite ainsi (Sontag 2008; Khalil, 2002)[18] :

$$\gamma = {\alpha_1}^{-1} \circ \alpha_2 \circ \rho \tag{3.16}$$

Celle-ci nous permet alors d'évaluer la grandeur de la borne restreignant l'état.

### 3.3.2 Analyse de la stabilité entrées à états

Soient les ensembles $\left\{\boldsymbol{x} \in B_V | \frac{\partial V}{\partial t} + \frac{\partial V}{\partial \boldsymbol{x}} \boldsymbol{f} < -W(\boldsymbol{x})\,,\ \|\boldsymbol{x}\| > \rho(\|\boldsymbol{u}\|)\right\}$ et $\{\boldsymbol{x} \in B_u | \ \|\boldsymbol{x}\| < \rho(\|\boldsymbol{u}\|)\}$ au sens de la condition (3.15). À partir de ces définitions, nous pouvons conclure

---

[17] Comme indiqué précédemment, la théorie s'applique en considérant $u$ comme l'entrée du système ou comme un vecteur de perturbation.

[18] Où $\circ$ indique la composition de fonction, tel que $f \circ g = f(g)$

que toutes les trajectoires dont les conditions initiales appartiennent à l'ensemble $B_V$ sont asymptotiquement stables tandis que celles appartenant à l'ensemble $B_u$ sont stables. Les deux ensembles sont illustrés par la Figure 3.2.

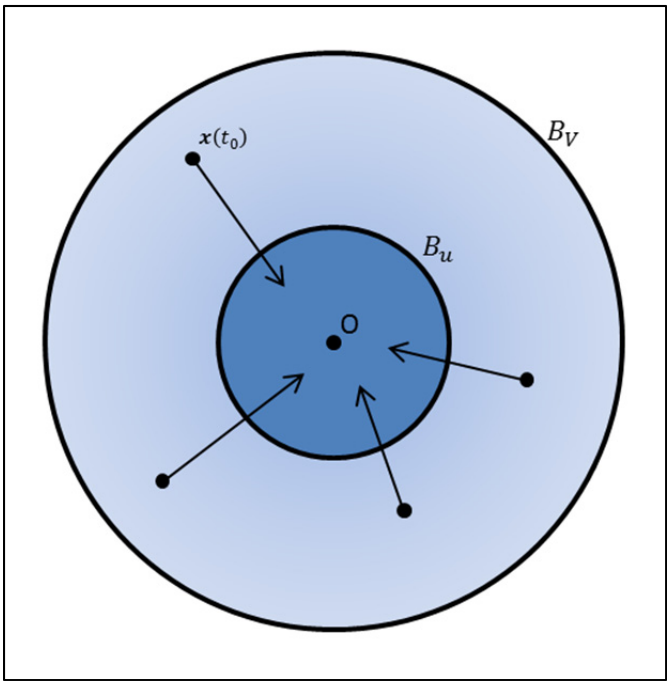



Figure 3.2 Dessin de principe de la condition suffisante pour les systèmes ISS

La condition (3.15) stipule que l'état d'un système ISS se retrouvant dans $B_V$ converge vers $B_u$ de façon asymptotique, de telle sorte qu'une fois compris dans l'ensemble $B_u$, il y demeure pour toujours. L'état demeure ensuite dans une sphère bornée par l'amplitude de la perturbation $\boldsymbol{u}$.

De plus, si $\|\boldsymbol{u}\|$ converge vers zéro lorsque $t \to \infty$, alors l'ensemble $B_u$ ne contient que l'origine, de telle sorte que l'état du système ISS converge asymptotiquement vers zéro. En effet, en posant $\|\boldsymbol{u}\| = 0$ dans l'équation (3.14) on obtient :

$$\|\boldsymbol{x}(t)\| \leq \beta(\boldsymbol{x}(t_0), t_0 - t)\,, \qquad \forall\, t > t_0 \tag{3.17}$$

Qui est la définition d'un système asymptotiquement stable au sens de Lyapunov.

### 3.3.3 Forme particulière d'un système ISS

Lors de la conception du contrôleur backstepping développé au chapitre 4, nous obtenons un cas particulier de système ISS. Nous allons identifier cette forme de système ici pour pouvoir y faire référence lors de la conception. Soit un système de la forme suivante :

$$\dot{\boldsymbol{x}} = -\Lambda \boldsymbol{x} - \boldsymbol{u} \tag{3.18}$$

où, $\boldsymbol{x} \in \mathbb{R}^n$ est le vecteur d'état et $\boldsymbol{u} \in \mathbb{R}^n$ est le vecteur de perturbation et $\Lambda$ est une matrice définie positive. Notre but est de démontrer qu'un système de cette forme est ISS. Soit la fonction candidate de Lyapunov suivante :

$$V = \frac{1}{2}\boldsymbol{x}^T\boldsymbol{x} \tag{3.19}$$

$$\begin{aligned} \dot{V} &= \boldsymbol{x}^T\dot{\boldsymbol{x}} \\ &= -\boldsymbol{x}^T\Lambda\boldsymbol{x} - \boldsymbol{x}^T\boldsymbol{u} \end{aligned} \tag{3.20}$$

Soit l'opérateur $\lambda_{min}(\cdot)$ qui retourne la plus petite valeur propre d'une matrice et soit $\lambda = \lambda_{min}(\Lambda)$. On peut réécrire (3.20) ainsi :

$$\dot{V} \leq -\lambda\|\boldsymbol{x}\|_2^2 + \|\boldsymbol{x}\|_2\|\boldsymbol{u}\|_2 \tag{3.21}$$

Soit le paramètre $\epsilon$ défini tel que $0 < \epsilon < 1$. Nous pouvons écrire :

$$\dot{V} \leq -\lambda(1-\epsilon)\|\boldsymbol{x}\|_2^2 - \lambda\epsilon\|\boldsymbol{x}\|_2^2 + \|\boldsymbol{x}\|_2\|\boldsymbol{u}\|_2 \tag{3.22}$$

L'équation (3.22) est toujours inférieure ou égale à zéro si,

$$-\lambda\epsilon\|\boldsymbol{x}\|_2^2 + \|\boldsymbol{x}\|_2\|\boldsymbol{u}\|_2 \leq 0 \tag{3.23}$$

Telle que :

$$\dot{V} \leq -\lambda(1-\epsilon)\|\boldsymbol{x}\|_2^2, \qquad \forall\|\boldsymbol{x}\|_2 \geq \frac{\|\boldsymbol{u}\|_2}{\lambda\epsilon} \tag{3.24}$$

L'équation (3.24) respecte la condition (3.15) de la section précédente. Nous pouvons donc conclure qu'un système de la forme de (3.18) ou dont $\dot{V}$ est de la forme (3.20) est ISS.

## 3.4 Estimateur exact de dérivée

Comme nous l'avons présenté à la section 3.1, le contrôleur backstepping est obtenu par construction. À chaque étape de conception, il est nécessaire d'effectuer la dérivée du contrôleur virtuel de l'étape précédente pour obtenir le système d'erreur augmenté, comme dans le cas de l'équation (3.6). Or, plus l'ordre du système est élevé, plus la conception du contrôleur backstepping nécessite d'étapes et plus la dérivée du contrôleur virtuel précédent devient complexe et dépend de dérivées d'état qui ne sont pas disponibles pour la commande. Il s'agit d'un des principaux défauts du backstepping (Farrel, Polycarpou *et al,* 2008).

Il existe plusieurs méthodes dans la littérature pour résoudre ce problème. La première solution proposée est d'utiliser des filtres numériques pour obtenir la dérivée des signaux nécessaires tels que présenté par (Farrel, Polycarpou *et al.,* 2008; Jian, Thomas *et al.*, 2012). Une deuxième solution est d'utiliser un estimateur exact de dérivée. Celui utilisé dans la littérature est basé sur un algorithme de mode glissant d'ordre 2 aussi appelé « Super Twisting Sliding Mode » (Levant, 1998). Celui-ci a déjà été utilisé par d'autres auteurs ayant la même problématique lors de la conception de contrôleur de type backstepping appliqué à des quadrotors (Madani et Benallegue, 2007).

L'analyse de l'estimateur exact de dérivée se base sur la théorie d'inclusion différentielle (Filipov, 1988) qui est complexe et dépasse le cadre de ce mémoire. Nous ne présenterons ici que la forme de l'estimateur, ses caractéristiques ainsi que les conditions suffisantes pour assurer la convergence de celui-ci. (Levant, 1993, 1998) présente une analyse complète de celui-ci.

En se basant sur (Levant, 1998), l'estimateur a la forme suivante :

$$\begin{aligned} \hat{\alpha} &= \xi - \lambda\sqrt{|\alpha - \hat{\alpha}|} \cdot \text{sign}(\alpha - \hat{\alpha}) \\ \dot{\xi} &= -\beta \cdot \text{sign}(\alpha - \hat{\alpha}) \end{aligned} \tag{3.25}$$

Où $\alpha$ est la variable dont on cherche à calculer la dérivée, $\hat{\alpha}$ est l'estimation de cette variable, $\xi$ est une variable d'état de l'estimateur, $\dot{\hat{\alpha}}$ est la dérivée estimée, sign est la fonction signe[19] et $\lambda$, $\beta$ sont des gains de l'estimateur.

L'erreur d'estimation $\left|\dot{\alpha} - \dot{\hat{\alpha}}\right|$ converge à zéro en temps fini, et ceci très rapidement si le signal dont on cherche la dérivée en fonction du temps est Lipschitz et si les conditions suivantes sont respectées (Levant, 1998)

$$\begin{aligned} &\beta > L \\ &\lambda^2 \geq 4L\frac{\beta + L}{\beta - L} \end{aligned} \tag{3.26}$$

Où $L$ est la constante de Lipschitz du signal $\alpha$. Puisque la fonction analytique du signal à dérivée est typiquement inconnue, il a été démontré que l'on peut choisir $L$, dans la condition (3.26), comme étant le double de l'accélération maximale de la variable dont nous estimons la dérivée (Davila, Fridman et Levant, 2005).

---

[19] Tel que $\text{sign}(x) = 1\,, \forall x > 0$ ; $\text{sign}(x) = -1\,, \forall < 0$ ; $\text{sign}(0) = 0$

Puisque le contrôleur est ISS par rapport aux erreurs d'estimation et que celles-ci convergent en temps fini, le principe de séparation toujours applicable dans le cas des systèmes linéaires (Krstić, Kanellakoupoulos et Kokotović, 1995) est satisfait dans le sens que l'on peut concevoir l'estimateur et le contrôleur du système séparément (Davila, Fridman et Levant, 2005). Cet aspect du contrôleur est présenté au CHAPITRE 4.

# CHAPITRE 4

# CONCEPTION DU CONTRÔLEUR BACKSTEPPING

Nous avons présenté lors des chapitres précédents tous les éléments nécessaires pour effectuer la conception du contrôleur backstepping. Dans ce chapitre, nous présenterons une forme du modèle du quadrotor plus appropriée pour effectuer la conception du contrôleur. Celle-ci est ensuite présentée en la séparant en deux parties. Pour finir, un résumé des signaux de contrôle est présenté.

## 4.1 Formulation du modèle de contrôle du quadrotor

Tel que défini au chapitre 3, le modèle du quadrotor est :

$$\begin{aligned}
\dot{\boldsymbol{R}} &= \boldsymbol{R}S(\boldsymbol{\omega}) \\
\dot{\boldsymbol{\eta}} &= \boldsymbol{R}\boldsymbol{v} \\
\dot{\boldsymbol{v}} &= -\boldsymbol{S}(\boldsymbol{\omega})\boldsymbol{v} + \boldsymbol{R}^T g \cdot \boldsymbol{k} - \frac{T}{m}\,\boldsymbol{k} \\
\dot{\boldsymbol{\Theta}} &= \boldsymbol{\rho}(\boldsymbol{\Theta})\boldsymbol{\omega} \\
\boldsymbol{I}\dot{\boldsymbol{\omega}} &= \boldsymbol{\tau} - \boldsymbol{\omega} \times \boldsymbol{I}\boldsymbol{\omega} - \boldsymbol{G}_r
\end{aligned} \tag{4.1}$$

Ce modèle est semblable à celui obtenu par (Xu, Xian et al., 2010; Raptis et Valavanis; 2011). Celui-ci néglige les effets aéronautiques suivants : la friction de l'air, l'effet de sol, l'effet du vent, le battement d'hélice et la variation de constante de poussée $b$ et de traînée $d$. De plus, on considère que le centre de gravité du quadrotor coïncide avec le centre géométrique de l'appareil. Notons que dans ce modèle, la dynamique de positionnement est exprimée par rapport à $\mathcal{F}_b$. Puisqu'il nous intéresse de contrôler la position du quadrotor par rapport au repère inertiel $\mathcal{F}_I$, nous devons effectuer un changement de variable.

Soit la vitesse linéaire exprimée dans $\mathcal{F}_I$ :

$$\boldsymbol{\mu} = \boldsymbol{R}\boldsymbol{v} \tag{4.2}$$

En utilisant celle-ci, on peut réécrire la dynamique du quadrotor ainsi :

$$\Pi_1: \begin{cases} \dot{\boldsymbol{\eta}} = \boldsymbol{\mu} \\ \dot{\boldsymbol{\mu}} = g \cdot \boldsymbol{k} - \dfrac{T}{m}\boldsymbol{R}\boldsymbol{k} \end{cases}$$
$$\Pi_2: \begin{cases} \dot{\boldsymbol{\Theta}} = \boldsymbol{\rho}\boldsymbol{\omega} \\ \boldsymbol{I}\dot{\boldsymbol{\omega}} = -\boldsymbol{\omega} \times \boldsymbol{I}\boldsymbol{\omega} - \boldsymbol{G}_r + \boldsymbol{\tau} \end{cases} \tag{4.3}$$

Nous avons choisi de diviser la dynamique en deux sous-systèmes pour simplifier la conception du contrôleur. La division est intuitive de façon à ce que le sous-système $\Pi_1$ représente la dynamique du positionnement du quadrotor tandis que $\Pi_2$ représente la dynamique de l'orientation du quadrotor.

Nous pouvons noter également que chacun des sous-systèmes est de forme « Strict-feedback system ». Ceux-ci sont interconnectés à l'aide de la matrice de rotation $\boldsymbol{R}$ et donc, du même coup, à l'aide des angles d'Euler $\boldsymbol{\Theta}$. L'interconnexion entre les deux systèmes peut être définie comme étant de la forme « Pure-feedback system ». Les deux formes sont définies aux sections 3.2.1 et 3.2.2.

## 4.2 Problématique de contrôle

Nous choisissons de contrôler la position et le cap du quadrotor tel que le contrôleur permet la poursuite des trajectoires désirées suivantes : $\boldsymbol{\eta}_d(t) = [x_d(t) \quad y_d(t) \quad z_d(t)]^t$ et $\psi_d(t)$. Nous posons que les trajectoires sont suffisamment lisses de telle façon que leurs dérivées soient bornées par des constantes[20]. Les valeurs de celles-ci sont directement liées aux

[20] Les trajectoires ne sont pas obligatoirement des fonctions de classe $C^\infty$. Pour que celles-ci respectent nos conditions, elles doivent être théoriquement de classe $C^3$ minimalement, de telle sorte que $\left\|\boldsymbol{\eta}_d(t)^{(i)}\right\| < \delta_i$ , $i = 0 \ldots 3$ et $\left\|\psi(t)^{(i)}\right\| < \varrho_i$ , $i = 0 \ldots 3$ où $\delta_i$ et $\varrho_i$ sont des constantes positives.

performances atteignables par le quadrotor qui dépendent principalement de sa dynamique ainsi que des caractéristiques des actionneurs[21].

Notre but est de concevoir un contrôleur qui nous permet de poursuivre les trajectoires de telle sorte que tous les états du quadrotor demeurent bornés en tout temps et, plus précisément, que la position ainsi que le cap du quadrotor convergent asymptotiquement aux valeurs désirées telles que :

$$\lim_{t\to\infty} \|\boldsymbol{\eta} - \boldsymbol{\eta}_d\| = 0$$
$$\lim_{t\to\infty} \|\psi - \psi_d\| = 0$$

Pour ce faire, le contrôleur doit générer quatre signaux de contrôle qui seront utilisés par les quatre entrées du quadrotor, soit la force de poussée totale $T$ ainsi que les trois moments $\tau_\phi$, $\tau_\theta$ et $\tau_\psi$.

## 4.3 Description des étapes de conception du contrôleur

Pour résoudre notre problématique de contrôle, nous choisissons d'utiliser un contrôleur de type backstepping tel que présenté à la section 3.1. Nous pouvons ainsi facilement tirer avantage du fait que la dynamique de $\Pi_1$ et $\Pi_2$ est déjà de la forme appropriée pour appliquer cette stratégie. De plus, l'application du backstepping permet de résoudre le problème créé par la nature sous-actionnée de la dynamique en la liant à l'aide de contrôleur virtuel.

La conception du contrôleur backstepping comprend quatre étapes, soit deux par sous-système. La structure de celui-ci est illustrée par la Figure 4.1. Nous commençons la conception du contrôleur par le sous-système $\Pi_1$. Cette partie nous permet de contrôler la position du quadrotor, nous permettant ainsi de déduire la valeur de la force de poussée totale

[21] Voir l'ANNEXE I pour plus de détails.

$T$ en plus de la valeur du contrôle virtuel pour les angles d'attitude, notés dans ce qui suit $\boldsymbol{\alpha}_{\Theta}$[22]. Ceux-ci représentent le lien avec le contrôleur du sous-système $\Pi_2$. Intuitivement, on peut interpréter l'action de cette partie du contrôleur comme étant le calcul de la force nécessaire pour maintenir l'appareil à l'altitude désirée tout en calculant l'orientation nécessaire de la force de poussée pour assurer le déplacement désiré dans le plan x-y.

Le contrôleur du sous-système $\Pi_2$ est dédié à la commande de l'attitude du quadrotor. Il assure donc la poursuite des contrôles virtuels $\alpha_\phi$ et $\alpha_\theta$ ainsi que de la trajectoire $\psi_d$ en maintenant les erreurs de roulis, de tangage et de lacet bornées et en les faisant tendre vers zéro. Cette partie du contrôleur nous permet de calculer les trois signaux de commande restants, soit les trois moments $\tau_\phi$, $\tau_\theta$ et $\tau_\psi$. Suite au calcul de $T$, $\tau_\phi$, $\tau_\theta$ et $\tau_\psi$, nous appliquons la correspondance (2.35) pour obtenir les vitesses de moteur désirées.

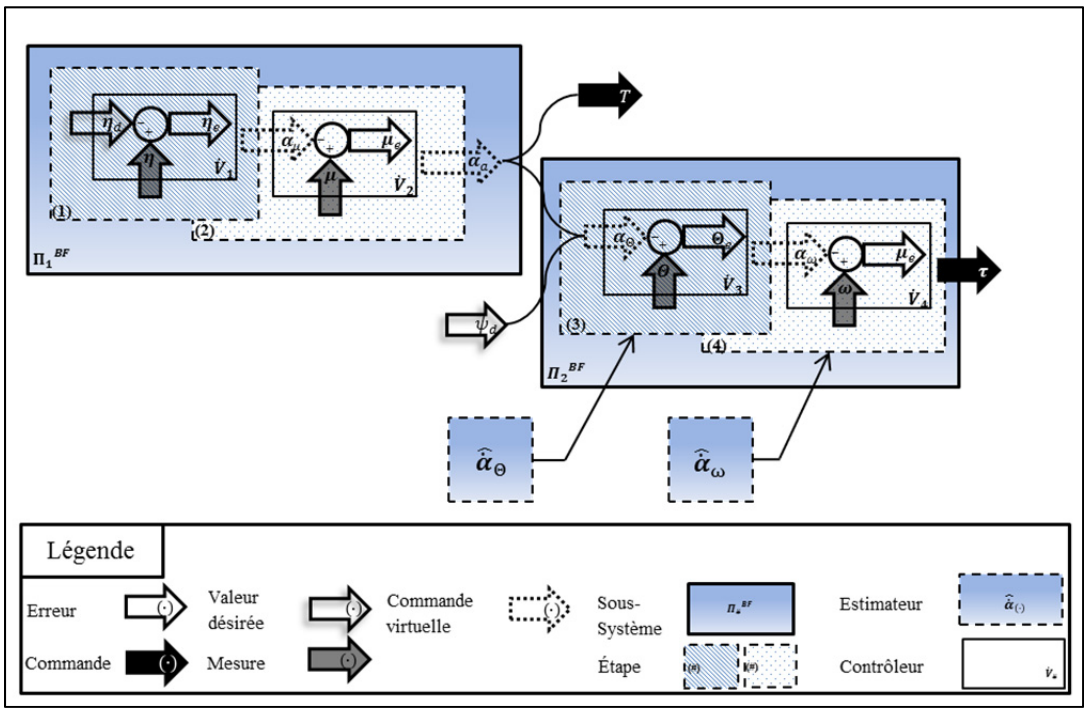



Figure 4.1 Structure du contrôleur

[22] Il est important de noter que $\boldsymbol{\alpha}_{\Theta} = \begin{bmatrix}\alpha_\phi & \alpha_\theta & \psi_d\end{bmatrix}^t$ et que la valeur de lacet désirée n'est pas une commande virtuelle, mais bien une trajectoire choisie par l'utilisateur.

## 4.4 Sous-système 1

### 4.4.1 Étape 1 : Contrôle de la position

Définissons l'erreur de position ainsi que sa dérivée par rapport au temps :

$$\boldsymbol{\eta}_e = \boldsymbol{\eta} - \boldsymbol{\eta}_d \tag{4.4}$$

$$\dot{\boldsymbol{\eta}}_e = \dot{\boldsymbol{\eta}} - \dot{\boldsymbol{\eta}}_d \tag{4.5}$$

Tel que présenté à la section 3.1, nous utilisons la vitesse pour contrôler l'erreur de position. Soit la fonction candidate de Lyapunov du système (4.4) :

$$V_1 = \frac{1}{2}\boldsymbol{\eta}_e{}^T\boldsymbol{\eta}_e \tag{4.6}$$

$$\dot{V}_1 = \boldsymbol{\eta}_e{}^T\dot{\boldsymbol{\eta}}_e = \boldsymbol{\eta}_e{}^T[\dot{\boldsymbol{\eta}} - \dot{\boldsymbol{\eta}}_d] = \boldsymbol{\eta}_e{}^T[\boldsymbol{\mu} - \dot{\boldsymbol{\eta}}_d] \tag{4.7}$$

Le choix de la commande virtuelle $\boldsymbol{\alpha}_\mu$, nous permet d'obtenir la fonction $\dot{V}_1$ définie négative lorsque $\boldsymbol{\mu} \rightarrow \boldsymbol{\alpha}_\mu$.

$$\boldsymbol{\alpha}_\mu = \dot{\boldsymbol{\eta}}_d - \boldsymbol{\Lambda}_1\boldsymbol{\eta}_e \tag{4.8}$$

où $\boldsymbol{\Lambda}_1$ est une matrice définie positive à être choisie par le concepteur.

### 4.4.2 Étape 2 : Contrôle de la vitesse linéaire

Définissons l'erreur de vitesse linéaire $\boldsymbol{\mu}_e$ entre la vitesse linéaire du quadrotor $\boldsymbol{\mu}$ et la commande virtuelle $\boldsymbol{\alpha}_\mu$ ainsi que sa dérivée par rapport au temps :

$$\boldsymbol{\mu}_e = \boldsymbol{\mu} - \boldsymbol{\alpha}_\mu \tag{4.9}$$

$$\dot{\boldsymbol{\mu}}_e = \dot{\boldsymbol{\mu}} - \dot{\boldsymbol{\alpha}}_\mu \tag{4.10}$$

En recombinant les équations précédentes, nous pouvons écrire le système :

$$\dot{\boldsymbol{\eta}}_e = \boldsymbol{\mu}_e - \boldsymbol{\Lambda}_1 \boldsymbol{\eta}_e \tag{4.11}$$

En introduisant l'équation de la vitesse linéaire $\dot{\boldsymbol{\mu}}$ du système $\Pi_1$ défini par la dynamique (4.3) et la dérivée de (4.8) dans (4.10) on obtient :

$$\dot{\boldsymbol{\mu}}_e = g \cdot \boldsymbol{k} - \frac{T}{m}\boldsymbol{R}\boldsymbol{k} - \ddot{\boldsymbol{\eta}}_d + \boldsymbol{\Lambda}_1(\boldsymbol{\mu}_e - \boldsymbol{\Lambda}_1 \boldsymbol{\eta}_e) \tag{4.12}$$

Les équations (4.11) et (4.12) constituent le système augmenté des erreurs de position et de vitesse que l'on veut stabiliser à l'origine. Soit la fonction candidate de Lyapunov du système augmenté :

$$V_2 = V_1 + \frac{1}{2}\boldsymbol{\mu}_e{}^T \boldsymbol{\mu}_e \tag{4.13}$$

$$\begin{aligned} \dot{V}_2 &= \boldsymbol{\eta}_e{}^T \dot{\boldsymbol{\eta}}_e + \boldsymbol{\mu}_e{}^T \dot{\boldsymbol{\mu}}_e \\ &= \boldsymbol{\eta}_e{}^T [\boldsymbol{\mu}_e - \boldsymbol{\Lambda}_1 \boldsymbol{\eta}_e] + \boldsymbol{\mu}_e{}^T \left[ g \cdot \boldsymbol{k} - \frac{T}{m}\boldsymbol{R}\boldsymbol{k} - \ddot{\boldsymbol{\eta}}_d + \boldsymbol{\Lambda}_1(\boldsymbol{\mu}_e - \boldsymbol{\Lambda}_1 \boldsymbol{\eta}_e) \right] \\ &= -\boldsymbol{\eta}_e{}^T \boldsymbol{\Lambda}_1 \boldsymbol{\eta}_e + \boldsymbol{\mu}_e{}^T \left[ \boldsymbol{\eta}_e + g \cdot \boldsymbol{k} - \frac{T}{m}\boldsymbol{R}\boldsymbol{k} - \ddot{\boldsymbol{\eta}}_d + \boldsymbol{\Lambda}_1(\boldsymbol{\mu}_e - \boldsymbol{\Lambda}_1 \boldsymbol{\eta}_e) \right] \end{aligned} \tag{4.14}$$

Nous devons maintenant utiliser le vecteur $\boldsymbol{\Theta}$ pour forcer $\dot{V}_2$ à être une fonction définie négative. Soit le contrôle virtuel $\boldsymbol{\alpha}_\Theta$ et définissons l'erreur de position angulaire :

$$\boldsymbol{\Theta}_e = \boldsymbol{\Theta} - \boldsymbol{\alpha}_\Theta \tag{4.15}$$

L'équation (4.14) ne possède pas de terme $\boldsymbol{\Theta}$ pouvant être directement utilisé pour effectuer le contrôle. L'approche employée ici est similaire à celle employée pour les systèmes ayant la forme « Pure-feedback system » telle que décrite à la section 3.2.2. Ainsi, plutôt qu'utiliser le vecteur $\boldsymbol{\Theta}$ directement, nous utilisons la matrice $\boldsymbol{R}(\boldsymbol{\Theta})$, dont les paramètres sont les angles d'Euler.

En se basant sur (Ghommam, Charland-Arcand *et al*., 2014), il est possible de décomposer la matrice de rotation en deux matrices telles que[23] :

$$\boldsymbol{R}(\boldsymbol{\alpha}_\Theta) = \boldsymbol{R}(\boldsymbol{\Theta}) - \boldsymbol{Q}(\boldsymbol{\Theta}_e, \boldsymbol{\alpha}_\Theta) \tag{4.16}$$

L'équation (4.16) est de la même forme que les autres équations d'erreur, par exemple (4.15), où $\boldsymbol{R}(\boldsymbol{\alpha}_\Theta)$ est la matrice de rotation ayant comme paramètre le contrôle virtuel $\boldsymbol{\alpha}_\Theta$ et où $\boldsymbol{Q}(\boldsymbol{\Theta}_e, \boldsymbol{\alpha}_\Theta)$ peut être considérée comme une matrice d'erreur.

En substituant (4.16) dans (4.14), nous obtenons :

$$\begin{aligned}\dot{V}_2 = & -\boldsymbol{\eta}_e{}^T \boldsymbol{\Lambda}_1 \boldsymbol{\eta}_e \\ & + \boldsymbol{\mu}_e{}^T \left[ \boldsymbol{\eta}_e + g \cdot \boldsymbol{k} - \frac{T}{m} \boldsymbol{R}(\boldsymbol{\alpha}_\Theta)\boldsymbol{k} - \ddot{\boldsymbol{\eta}}_d + \boldsymbol{\Lambda}_1(\boldsymbol{\mu}_e - \boldsymbol{\Lambda}_1 \boldsymbol{\eta}_e) \right] \\ & - \boldsymbol{\mu}_e{}^T \frac{T}{m} \boldsymbol{Q}(\boldsymbol{\Theta}_e, \boldsymbol{\alpha}_\Theta)\boldsymbol{k}\end{aligned} \tag{4.17}$$

Notre but est d'utiliser le terme $-\frac{T}{m}\boldsymbol{R}(\boldsymbol{\alpha}_\Theta)\boldsymbol{k}$ pour contraindre $\dot{V}_2$ à devenir une fonction définie négative.

---

[23] Voir l'ANNEXE II pour la démonstration de cette séparation ainsi que la liste des éléments de $\boldsymbol{Q}(\boldsymbol{\Theta}_e, \boldsymbol{\alpha}_\Theta)$.

$$\boldsymbol{\alpha_a} \triangleq \frac{T}{m}\boldsymbol{R}(\boldsymbol{\alpha}_\Theta)\boldsymbol{k} = \boldsymbol{\eta}_e + g \cdot \boldsymbol{k} - \ddot{\boldsymbol{\eta}}_d + \boldsymbol{\Lambda}_1(\boldsymbol{\mu}_e - \boldsymbol{\Lambda}_1\boldsymbol{\eta}_e) + \boldsymbol{\Lambda}_2\boldsymbol{\mu}_e \tag{4.18}$$

Où $\boldsymbol{\Lambda}_2$ est une matrice définie positive à être choisie par le concepteur. Le vecteur $\boldsymbol{\alpha_a}$, soit la commande virtuelle d'accélération, peut être interprété physiquement comme étant l'accélération linéaire du quadrotor nécessaire pour effectuer la poursuite de la trajectoire désirée. Cette accélération nous permet d'obtenir deux informations nécessaires à la conception du contrôleur, soit la force de poussée totale ainsi que l'orientation de celle-ci.

En insérant (4.18) dans (4.17), on obtient :

$$\dot{V}_2 = -\boldsymbol{\eta}_e{}^T\boldsymbol{\Lambda}_1\boldsymbol{\eta}_e - \boldsymbol{\mu}_e{}^T\boldsymbol{\Lambda}_2\boldsymbol{\mu}_e - \boldsymbol{\mu}_e{}^T\frac{T}{m}\boldsymbol{Q}(\boldsymbol{\Theta}_e, \boldsymbol{\alpha}_\Theta)\boldsymbol{k} \tag{4.19}$$

### 4.4.3 Sous-système 1 en boucle fermée

Le système en boucle fermée $\Pi_1{}^{BF}$ est obtenu en appliquant les commandes virtuelles à la dynamique de l'erreur du sous-système $\Pi_1$.

$$\Pi_1{}^{BF} : \begin{cases} \dot{\boldsymbol{\eta}}_e = \boldsymbol{\mu}_e - \boldsymbol{\Lambda}_1\boldsymbol{\eta}_e \\ \dot{\boldsymbol{\mu}}_e = -\boldsymbol{\eta}_e - \boldsymbol{\Lambda}_2\boldsymbol{\mu}_e - \frac{T}{m}\boldsymbol{Q}(\boldsymbol{\Theta}_e, \boldsymbol{\alpha}_\Theta)\boldsymbol{k} \end{cases} \tag{4.20}$$

Le terme $\frac{T}{m}\boldsymbol{Q}(\boldsymbol{\Theta}_e, \boldsymbol{\alpha}_\Theta)\boldsymbol{k}$ est considéré comme une perturbation par le contrôleur, car il contient des termes non mesurables. Cependant, nous pouvons constater que le système est ISS par rapport à la perturbation $\frac{T}{m}\boldsymbol{Q}(\boldsymbol{\Theta}_e, \boldsymbol{\alpha}_\Theta)\boldsymbol{k}$. En effet, le système en boucle fermée est d'une forme particulière, telle que définie à la section 3.3.3, identifiée comme étant un

système ISS[24], ce qui assure que les erreurs $\boldsymbol{\eta}_e$ et $\boldsymbol{\mu}_e$ sont bornées. De plus, en analysant en détail les éléments de la matrice $\boldsymbol{Q}(\boldsymbol{\Theta}_e, \boldsymbol{\alpha}_\Theta)$, on peut mettre en évidence deux propriétés intéressantes : la première est que la matrice $\boldsymbol{Q}(\boldsymbol{\Theta}_e, \boldsymbol{\alpha}_\Theta)$ est bornée, quelques soient ses entrées. Il existe donc une petite constante $\varepsilon$ telle que :

$$\|\boldsymbol{Q}(\boldsymbol{\Theta}_e, \boldsymbol{\alpha}_\Theta)\| < \varepsilon \ , \qquad \forall\, t \geq 0\ , \boldsymbol{\Theta}_e \in \mathbb{R}^3, \boldsymbol{\alpha}_\Theta \in \mathbb{R}^3 \tag{4.21}$$

La deuxième propriété fait en sorte que la norme de la matrice $\boldsymbol{Q}(\boldsymbol{\Theta}_e, \boldsymbol{\alpha}_\Theta)$ ne dépend que du terme $\Theta_e$ et décroit vers 0 si $\Theta_e$ décroit vers 0 tel que :

$$\lim_{t\to\infty} \boldsymbol{\Theta}_e \to 0 \ \Rightarrow \lim_{t\to\infty} \|\boldsymbol{Q}(\boldsymbol{\Theta}_e, \boldsymbol{\alpha}_\Theta)\| \to 0\ , \qquad \forall\, \boldsymbol{\alpha}_\Theta \in \mathbb{R}^3 \tag{4.22}$$

En combinant les deux propriétés (4.21) et (4.22) avec la propriété ISS du système ${\Pi_1}^{BF}$, on peut conclure que si nous parvenons à faire tendre l'erreur de position angulaire $\boldsymbol{\Theta}_e$ vers 0 alors le terme $\boldsymbol{Q}(\boldsymbol{\Theta}_e, \boldsymbol{\alpha}_\Theta)$**,** qui est la perturbation du système au sens ISS, tend lui aussi vers 0 et conséquemment l'origine du système en boucle fermé ${\Pi_1}^{BF}$ est asymptotiquement stable au sens de Lyapunov.

À la lumière de cette conclusion, on peut donc constater que l'erreur de position est directement liée à l'erreur d'attitude du quadrotor via le terme $\boldsymbol{Q}(\boldsymbol{\Theta}_e, \boldsymbol{\alpha}_\Theta)$. Ceci confirme notre intuition. En effet, il s'agit de l'orientation du quadrotor qui dirige la force de poussée dans l'espace. Tant que celle-ci n'est pas orientée correctement, la position désirée ne peut être atteinte. Cependant, si un contrôleur peut garantir que l'erreur d'orientation est asymptotiquement stable, alors la force sera ultimement orientée correctement et l'erreur de position convergera elle aussi ultimement vers zéro, d'où la propriété ISS du système ${\Pi_1}^{BF}$ par rapport à l'erreur d'orientation.

---

[24] Nous tirons cette conclusion en supposant que $\|\boldsymbol{Q}\|$ est borné, ce qui est démontré par la suite. La démonstration est donnée à l'ANNEXE III.

On peut donc conclure que le contrôleur d'attitude a une grande influence sur la dynamique de position du quadrotor et devra donc être très performant et robuste pour assurer la bonne performance au contrôleur de position.

### 4.4.4 Calcul de la force de poussée totale et des contrôles virtuels

La commande virtuelle d'accélération $\boldsymbol{\alpha_a}$, déduite à la section précédente, contient deux informations complémentaires, soit la force de portance totale $T$ ainsi que l'orientation désirée du quadrotor via les paramètres de la matrice $\boldsymbol{R}(\boldsymbol{\alpha_\Theta})$.

Réarrangeons l'équation (4.18) :

$$\frac{T}{m}\boldsymbol{k} = \boldsymbol{R^{-1}}(\boldsymbol{\alpha_\Theta})\boldsymbol{\alpha_a} \tag{4.23}$$

En prenant le carré de chaque côté de l'équation, nous obtenons :

$$\frac{T^2}{m^2} = \boldsymbol{\alpha_a}^T\boldsymbol{\alpha_a} \tag{4.24}$$

La force de poussée totale des moteurs doit donc être[25] :

$$T = m\sqrt{\boldsymbol{\alpha_a}^T\boldsymbol{\alpha_a}} \tag{4.25}$$

Définissons les éléments du vecteur tels que $\boldsymbol{\alpha_a} = [\alpha_{a_x} \quad \alpha_{a_y} \quad \alpha_{a_z}]^T$. En reprenant l'équation (4.23), on peut utiliser chaque terme du vecteur $\boldsymbol{\alpha_a}$ pour extraire $\boldsymbol{\alpha_\Theta}$ de la matrice $\boldsymbol{R}(\boldsymbol{\alpha_\Theta})$.

[25] Le résultat de l'équation (4.25) est intuitif, car celle-ci est l'expression de la loi de Newton ($F = ma$).

$$\boldsymbol{R}^T(\boldsymbol{\alpha}_\Theta)\begin{bmatrix}\alpha_{a_x}\\ \alpha_{a_y}\\ \alpha_{a_z}\end{bmatrix}=\begin{bmatrix}0\\ 0\\ T/m\end{bmatrix} \tag{4.26}$$

L'équation (4.26) peut être réécrite sous la forme d'un système de trois équations :

$$0 = c_{\alpha_\theta}c_{\alpha_\psi}\alpha_{a_x} + c_{\alpha_\theta}s_{\alpha_\psi}\alpha_{a_y} - s_{\alpha_\theta}\alpha_{a_z} \tag{4.27}$$

$$\begin{aligned}0 = &\left(s_{\alpha_\phi}s_{\alpha_\theta}c_{\alpha_\psi} - c_{\alpha_\phi}s_{\alpha_\psi}\right)\alpha_{a_x}\\ &+\left(s_{\alpha_\phi}s_{\alpha_\theta}s_{\alpha_\psi} + c_{\alpha_\phi}c_{\alpha_\psi}\right)\alpha_{a_y}\\ &+\left(s_{\alpha_\phi}c_{\alpha_\theta}\right)\alpha_{a_z}\end{aligned} \tag{4.28}$$

$$\begin{aligned}T/m = &\left(c_{\alpha_\phi}s_{\alpha_\theta}c_{\alpha_\psi} + s_{\alpha_\phi}s_{\alpha_\psi}\right)\alpha_{\boldsymbol{a}_x}\\ &+\left(c_{\alpha_\phi}s_{\alpha_\theta}s_{\alpha_\psi} - s_{\alpha_\phi}c_{\alpha_\psi}\right)\alpha_{a_y}\\ &+\left(c_{\alpha_\phi}c_{\alpha_\theta}\right)\alpha_{a_z}\end{aligned} \tag{4.29}$$

En multipliant chaque côté de l'équation (4.28) par $-c_{\alpha_\phi}$ et chaque côté de l'équation (4.29) par $s_{\alpha_\phi}$ et en additionnant les deux résultats, nous obtenons :

$$\frac{T}{m}s_{\alpha_\phi} = \alpha_{a_x}s_{\alpha_\psi} - \alpha_{a_y}c_{\alpha_\psi} \tag{4.30}$$

En notant pour les étapes ultérieures que $\alpha_\theta \neq \frac{\pi}{2}$, on multiplie chaque côté de l'équation (4.27) par $\frac{1}{c_{\alpha_\theta}}$[26].

[26] Pour être plus exacte, $\alpha_\theta \neq \frac{\pi}{2}(2i-1)$, où $i$ est un entier.

$$t_{\alpha_\theta}\alpha_{a_z} = c_{\alpha_\psi}\alpha_{a_x} + s_{\alpha_\psi}\alpha_{a_y} \tag{4.31}$$

À partir des équations (4.30) et (4.31), nous pouvons facilement isoler les commandes virtuelles $\alpha_\phi$ et $\alpha_\theta$.

$$\alpha_\phi = \sin^{-1}\left(\frac{\alpha_{a_x}s_{\alpha_\psi} - \alpha_{a_y}c_{\alpha_\psi}}{\sqrt{\boldsymbol{\alpha_a}^T\boldsymbol{\alpha_a}}}\right) \tag{4.32}$$

$$\alpha_\theta = \tan^{-1}\left(\frac{\alpha_{a_x}c_{\alpha_\psi} + \alpha_{a_y}s_{\alpha_\psi}}{\alpha_{a_z}}\right) \tag{4.33}$$

Où $\alpha_\psi = \psi_d(t)$, est la trajectoire désirée du cap du quadrotor qui est disponible pour le contrôle.

### 4.4.5 Analyse du domaine de validité

Les équations précédentes possèdent plusieurs restrictions sur leur domaine qui doivent être respectées pour s'assurer que les commandes virtuelles soient valides.

Premièrement, le domaine de la fonction $\sin^{-1}$ de l'équation (4.32) est toujours respecté, car :

$$\left|\alpha_{a_x}s_{\alpha_\psi} - \alpha_{a_y}c_{\alpha_\psi}\right| \leq \sqrt{{\alpha_{a_x}}^2 + {\alpha_{a_y}}^2} \leq \sqrt{\boldsymbol{\alpha_a}^T\boldsymbol{\alpha_a}}$$

De telle sorte que le domaine $[-1,1]$ est toujours respecté puisque :

$$\frac{\alpha_{a_x}s_{\alpha_\psi} - \alpha_{a_y}c_{\alpha_\psi}}{\sqrt{\boldsymbol{\alpha_a}^T\boldsymbol{\alpha_a}}} \leq 1$$

Deuxièmement, le terme $\sqrt{\boldsymbol{\alpha_a}^T \boldsymbol{\alpha_a}}$, qui est la norme de l'accélération virtuelle, ne doit jamais être nul pour éviter une division par zéro. Cette condition est difficile à résoudre mathématiquement, mais peut-être interprétée intuitivement[27]. En effet, si la norme de l'accélération virtuelle est nulle, alors, en se basant sur (4.25), la force de poussé totale T est nulle, de telle sorte que le quadrotor se retrouve en chute libre. Il n'existe alors aucune solution d'orientation, soit les commandes virtuelles $\alpha_\phi$ et $\alpha_\theta$, pour permettre au quadrotor d'atteindre la position désirée, d'où l'inexistence de solution pour l'équation (4.32). Or, lors d'un vol, le quadrotor doit toujours combattre la gravité pour demeurer à l'équilibre de telle sorte que la norme de l'accélération virtuelle ne peut jamais être nulle. Ceci nous permet de conclure que cette condition est remplie. Le même raisonnement s'applique pour le terme $\alpha_{a_z}$, qui est l'accélération virtuelle selon l'axe $k_I$.

Aussi, en nous basant sur les résultats de l'ANNEXE I, nous concluons que nous pouvons toujours, lors de l'implémentation, ajouter des saturations au niveau des contrôles virtuels $\alpha_\phi$ et $\alpha_\theta$. De plus, un traitement numérique peut être utilisé pour éviter les divisions par zéro au niveau de (4.32) et (4.33), qui sera utilisé typiquement lors du décollage et de l'atterrissage de quadrotor. En faisant référence à la section 4.2, nous assumons que la trajectoire désirée génère des mouvements dans une enveloppe de vol typique de telle sorte que le quadrotor n'est jamais en chute libre. De ce fait, l'accélération maximale de descente du quadrotor doit donc toujours être inférieure à $1g$.

À partir de cette analyse, nous concluons que le domaine des équations (4.32) et (4.33) est toujours respecté, ce qui nous permet de passer à la conception du contrôleur pour le sous-système $\Pi_2$.

---

[27] Il faudrait prouver que le minimum de la norme de l'accélération virtuelle est toujours supérieur à 0 à partir de l'équation (4.18).

## 4.5 Sous-système 2

Le contrôleur est construit en deux étapes. La première consiste à faire tendre l'erreur de position angulaire $\boldsymbol{\Theta}_e$ vers zéro à l'aide de la vitesse angulaire. La deuxième consiste à contrôler la vitesse angulaire à l'aide du vecteur de couple $\boldsymbol{\tau}$.

Soit le vecteur de commande virtuel $\boldsymbol{\alpha}_\Theta$. Il est important de noter que celui-ci est formé de deux commandes virtuelles ainsi que de la trajectoire désirée du lacet.

$$\boldsymbol{\alpha}_\Theta = [\alpha_\phi \quad \alpha_\theta \quad \psi_d]^T \tag{4.34}$$

### 4.5.1 Étape 3 : Contrôle de la position angulaire

Reprenons l'équation (4.15) qui décrit l'erreur de position angulaire et prenons sa dérivée par rapport au temps :

$$\boldsymbol{\Theta}_e = \boldsymbol{\Theta} - \boldsymbol{\alpha}_\Theta \tag{4.35}$$

$$\dot{\boldsymbol{\Theta}}_e = \dot{\boldsymbol{\Theta}} - \dot{\boldsymbol{\alpha}}_\Theta \tag{4.36}$$

Tel que présenté à la section 3.1, nous utilisons la variation des angles d'angulaire pour contrôler l'erreur de position angulaire.

Soit la fonction candidate de Lyapunov pour le système précédant ainsi que sa dérivée par rapport au temps :

$$\mathrm{V}_3 = \frac{1}{2}\boldsymbol{\Theta}_e{}^T\boldsymbol{\Theta}_e \tag{4.37}$$

$$\dot{V}_3 = \boldsymbol{\Theta}_e{}^T \dot{\boldsymbol{\Theta}}_e = \boldsymbol{\Theta}_e{}^T \left[\dot{\boldsymbol{\Theta}} - \dot{\boldsymbol{\alpha}}_\Theta\right] = \boldsymbol{\Theta}_e{}^T [\boldsymbol{\rho}(\boldsymbol{\Theta})\boldsymbol{\omega} - \dot{\boldsymbol{\alpha}}_\Theta] \tag{4.38}$$

En suivant la méthode de conception, nous pouvons choisir le contrôle virtuel $\boldsymbol{\alpha}_\omega$ suivant, où $\boldsymbol{\Lambda}_3$ est une matrice définie positive à être choisie par le concepteur, de manière à transformer la fonction $\dot{V}_3$ en une fonction définie négative.

$$\boldsymbol{\alpha}_\omega = \boldsymbol{\rho}(\boldsymbol{\Theta})^{-1} [\dot{\boldsymbol{\alpha}}_\Theta - \boldsymbol{\Lambda}_3 \boldsymbol{\Theta}_e] \tag{4.39}$$

Or, le terme $\dot{\boldsymbol{\alpha}}_\Theta$ provient des dérivées des équations (4.32) et (4.33) qui sont clairement très complexes à résoudre analytiquement. Comme nous l'avons présenté à la section 3.4, ce phénomène est normal lors de la conception de contrôleur backstepping pour un système d'ordre élevé. Pour résoudre ce problème, nous choisissons d'utiliser l'estimateur de dérivée, présenté à la section 3.4.

Le vecteur $\boldsymbol{\alpha}_\Theta$ est formé de deux commandes virtuelles dont les dérivées devront être estimées, la valeur de $\dot{\psi}_d$ étant déjà disponible. Puisque l'estimateur est scalaire, il nous faut deux estimateurs pour évaluer la dérivée de $\boldsymbol{\alpha}_\Theta$. Les estimations sont notées $\widehat{(\cdot)}$ tandis que les erreurs d'estimation sont notées $\widetilde{(\cdot)}$. En se basant sur les équations (3.25), on peut écrire l'estimateur ainsi :

$$\begin{aligned} \hat{\alpha}_i &= \xi_i - \lambda_i \sqrt{|\alpha_i - \hat{\alpha}_i|} \cdot \operatorname{sign}(\alpha_i - \hat{\alpha}_i) \\ \dot{\xi}_i &= -\beta_i \cdot \operatorname{sign}(\alpha_i - \hat{\alpha}_i) \end{aligned} \tag{4.40}$$

Où l'indice $i \in \{\phi \quad \theta\}$ indique le contrôleur virtuel associé à l'estimateur. Le vecteur d'estimation ainsi que le vecteur d'erreur peuvent être notés ainsi :

$$\begin{bmatrix} \tilde{\alpha}_\phi \\ \tilde{\alpha}_\theta \\ 0 \end{bmatrix} = \begin{bmatrix} \dot{\alpha}_\phi \\ \dot{\alpha}_\theta \\ \dot{\psi}_d \end{bmatrix} - \begin{bmatrix} \hat{\dot{\alpha}}_\phi \\ \hat{\dot{\alpha}}_\theta \\ \dot{\psi}_d \end{bmatrix} \tag{4.41}$$

ou sous forme vectorielle suivante :

$$\tilde{\boldsymbol{\alpha}}_{\Theta} = \dot{\boldsymbol{\alpha}}_{\Theta} - \hat{\dot{\boldsymbol{\alpha}}}_{\Theta} \tag{4.42}$$

Nous pouvons maintenant réécrire (4.39) en fonction de l'estimation de la dérivée.

$$\boldsymbol{\alpha}_{\omega} = \boldsymbol{\rho}(\boldsymbol{\Theta})^{-1}\left[\hat{\dot{\boldsymbol{\alpha}}}_{\Theta} - \boldsymbol{\Lambda}_3\boldsymbol{\Theta}_e\right] \tag{4.43}$$

Où $\boldsymbol{\rho}(\boldsymbol{\Theta})^{-1}$, qui est défini par l'inverse de l'équation (2.9), est valide pour tout $\theta \neq \frac{\pi}{2}$. Ce qui est cohérent avec nos conclusions des sections 4.4.4 et 4.4.5. En appliquant (4.43), nous pouvons réécrire l'équation (4.38), lorsque $\boldsymbol{\omega} \rightarrow \boldsymbol{\alpha}_{\boldsymbol{\omega}}$ ainsi :

$$\begin{aligned}\dot{\mathrm{V}}_3 &= \boldsymbol{\Theta}_e{}^t\dot{\boldsymbol{\Theta}}_e = \boldsymbol{\Theta}_e{}^t\left[\dot{\boldsymbol{\Theta}} - \dot{\boldsymbol{\alpha}}_{\Theta}\right] = \boldsymbol{\Theta}_e{}^t[\boldsymbol{\rho}(\boldsymbol{\Theta})\boldsymbol{\omega} - \dot{\boldsymbol{\alpha}}_{\Theta}] \\ &= -\boldsymbol{\Theta}_e{}^t\boldsymbol{\Lambda}_3\boldsymbol{\Theta}_e - \boldsymbol{\Theta}_e{}^t\tilde{\boldsymbol{\alpha}}_{\Theta}\end{aligned} \tag{4.44}$$

En observant l'équation (4.44), nous pouvons conclure que celui-ci est de la forme particulière propre aux système ISS tel que défini à la section 3.3.3[28]. De plus, puisque l'estimateur converge en temps fini, le terme $\tilde{\boldsymbol{\alpha}}_{\Theta}$ tend rapidement vers zéro de telle sorte que l'origine de l'équation (4.35) est asymptotiquement stable si $\boldsymbol{\omega} \rightarrow \boldsymbol{\alpha}_{\boldsymbol{\omega}}$.

### 4.5.2 Étape 4 : Contrôle de la vitesse angulaire

Nous pouvons définir l'erreur entre la vitesse angulaire et sa commande virtuelle ainsi que sa dérivée par rapport au temps :

$$\boldsymbol{\omega}_e = \boldsymbol{\omega} - \boldsymbol{\alpha}_{\omega} \tag{4.45}$$

[28] La preuve complète de stabilité est présentée à l'ANNEXE III

$$\dot{\boldsymbol{\omega}}_e = \dot{\boldsymbol{\omega}} - \dot{\boldsymbol{\alpha}}_{\omega} \tag{4.46}$$

En introduisant (4.45) dans (4.35) et (4.3) dans (4.46), on obtient le système augmenté suivant :

$$\dot{\boldsymbol{\Theta}}_e = \boldsymbol{\rho}(\boldsymbol{\Theta})\boldsymbol{\omega}_e - \boldsymbol{\Lambda}_3\boldsymbol{\Theta}_e - \widetilde{\dot{\boldsymbol{\alpha}}}_{\Theta} \tag{4.47}$$

$$\dot{\boldsymbol{\omega}}_e = \boldsymbol{I}^{-1}[-\boldsymbol{\omega} \times \boldsymbol{I}\boldsymbol{\omega} - \boldsymbol{G}_r + \boldsymbol{\tau}] - \dot{\boldsymbol{\alpha}}_{\omega} \tag{4.48}$$

Où $\boldsymbol{I}$ est le tenseur d'inertie qui est, par définition, une matrice définie positive. Celle-ci est donc toujours inversible.

Soit la fonction candidate de Lyapunov du système augmenté :

$$\mathrm{V}_4 = \mathrm{V}_3 + \frac{1}{2}\boldsymbol{\omega}_e{}^T\boldsymbol{I}\boldsymbol{\omega}_e \tag{4.49}$$

Sa dérivée par rapport au temps est :

$$\begin{aligned}\dot{\mathrm{V}}_4 &= \boldsymbol{\Theta}_e{}^T\dot{\boldsymbol{\Theta}}_e + \boldsymbol{\omega}_e{}^T\boldsymbol{I}\dot{\boldsymbol{\omega}}_e \\ &= \boldsymbol{\Theta}_e{}^T\left[\boldsymbol{\rho}(\boldsymbol{\Theta})\boldsymbol{\omega}_e - \boldsymbol{\Lambda}_3\boldsymbol{\Theta}_e - \widetilde{\dot{\boldsymbol{\alpha}}}_{\Theta}\right] + \boldsymbol{\omega}_e{}^T\boldsymbol{I}[\boldsymbol{I}^{-1}[-\boldsymbol{\omega} \times \boldsymbol{I}\boldsymbol{\omega} - \boldsymbol{G}_a + \boldsymbol{\tau}] - \dot{\boldsymbol{\alpha}}_{\omega}] \\ &= \dot{\mathrm{V}}_3 + \boldsymbol{\omega}_e{}^T[\boldsymbol{\rho}(\boldsymbol{\Theta})^T\boldsymbol{\Theta}_e - \boldsymbol{\omega} \times \boldsymbol{I}\boldsymbol{\omega} - \boldsymbol{G}_r + \boldsymbol{\tau} - \boldsymbol{I}\dot{\boldsymbol{\alpha}}_{\omega}]\end{aligned} \tag{4.50}$$

On choisit le vecteur d'entrée $\boldsymbol{\tau}$ tel que $\dot{\mathrm{V}}_4$ soit une fonction définie négative. Le choix logique est d'éliminer tous les termes positifs à l'aide de l'entrée telle que :

$$\boldsymbol{\tau} = -\boldsymbol{\rho}(\boldsymbol{\Theta})^T\boldsymbol{\Theta}_e + \boldsymbol{\omega} \times \boldsymbol{I}\boldsymbol{\omega} + \boldsymbol{G}_r + \boldsymbol{I}\dot{\boldsymbol{\alpha}}_{\omega} - \boldsymbol{\Lambda}_4\boldsymbol{\omega}_e \tag{4.51}$$

Nous obtenons la même problématique qu'à la section précédente, dans le sens où le calcul de $\dot{\boldsymbol{\alpha}}_{\omega}$ est beaucoup trop complexe pour être effectué de manière analytique. Nous utilisons donc à nouveau l'estimateur de dérivée défini à la section 3.4. Puisque l'estimateur est scalaire, il nous faut trois estimateurs. En se basant sur les équations (3.25), on peut écrire l'estimateur ainsi :

$$\begin{aligned} \hat{\dot{\alpha}}_i &= \xi_i - \lambda_i \sqrt{|\alpha_i - \hat{\alpha}_i|} \cdot \operatorname{sign}(\alpha_i - \hat{\alpha}_i) \\ \dot{\xi}_i &= -\beta_i \cdot \operatorname{sign}(\alpha_i - \hat{\alpha}_i) \end{aligned} \tag{4.52}$$

où l'indice $i \in \{p \quad q \quad r\}$ indique le contrôleur virtuel associé à l'estimateur. Le vecteur d'estimation ainsi que le vecteur d'erreur sont notés ainsi :

$$\tilde{\dot{\boldsymbol{\alpha}}}_{\omega} = \dot{\boldsymbol{\alpha}}_{\omega} - \hat{\dot{\boldsymbol{\alpha}}}_{\omega} \tag{4.53}$$

À partir de (4.53), nous pouvons réécrire (4.51) ainsi :

$$\boldsymbol{\tau} = -\boldsymbol{\rho}(\boldsymbol{\Theta})^T \boldsymbol{\Theta}_e + \boldsymbol{\omega} \times \boldsymbol{I}\boldsymbol{\omega} + \boldsymbol{G}_r + \boldsymbol{I}\hat{\dot{\boldsymbol{\alpha}}}_{\omega} - \boldsymbol{\Lambda}_4 \boldsymbol{\omega}_e \tag{4.54}$$

où $\boldsymbol{\Lambda}_4$ est une matrice définie positive choisie par le concepteur tel qu'on peut aussi réécrire (4.50) :

$$\dot{V}_4 = -\boldsymbol{\Theta}_e{}^T \boldsymbol{\Lambda}_3 \boldsymbol{\Theta}_e - \boldsymbol{\omega}_e{}^T \boldsymbol{\Lambda}_4 \boldsymbol{\omega}_e - \boldsymbol{\Theta}_e{}^T \tilde{\dot{\boldsymbol{\alpha}}}_{\Theta} - \boldsymbol{\omega}_e{}^T \boldsymbol{I} \tilde{\dot{\boldsymbol{\alpha}}}_{\omega} \tag{4.55}$$

En introduisant (4.54) dans (4.48), on obtient le système en boucle fermée $\Pi_2{}^{BF}$ :

$$\Pi_2{}^{BF} : \begin{cases} \dot{\boldsymbol{\Theta}}_e = \boldsymbol{\rho}(\boldsymbol{\Theta})\boldsymbol{\omega}_e - \boldsymbol{\Lambda}_3 \boldsymbol{\Theta}_e - \tilde{\dot{\boldsymbol{\alpha}}}_{\Theta} \\ \boldsymbol{I}\dot{\boldsymbol{\omega}}_e = -\boldsymbol{\rho}(\boldsymbol{\Theta})^t \boldsymbol{\Theta}_e - \boldsymbol{\Lambda}_4 \boldsymbol{\omega}_e - \tilde{\dot{\boldsymbol{\alpha}}}_{\omega} \end{cases} \tag{4.56}$$

En observant l'équation (4.55), nous pouvons conclure que celui-ci est de la forme particulière propre aux systèmes ISS tel que défini à la section 3.3.3. De plus, puisque l'estimateur converge en temps fini, l'erreur $\tilde{\boldsymbol{\alpha}}_{\omega}$ tend rapidement vers zéro de telle sorte que l'origine du système en boucle fermé ${\Pi_2}^{BF}$ est asymptotiquement stable[29].

## 4.6 Résumé du contrôleur

Puisque la démarche du contrôleur est longue, nous allons résumer dans cette section l'essentiel de la démarche.

### 4.6.1 Système complet en boucle fermée

En combinant les systèmes en boucle fermée ${\Pi_1}^{BF}$ et ${\Pi_2}^{BF}$, on obtient le système en boucle fermée complet, qui peut être écrit sous la forme matricielle suivante :

$$\begin{bmatrix} \dot{\boldsymbol{\eta}}_e \\ \dot{\boldsymbol{\mu}}_e \\ \dot{\boldsymbol{\Theta}}_e \\ \boldsymbol{I}\dot{\boldsymbol{\omega}}_e \end{bmatrix} = \begin{bmatrix} -\boldsymbol{\Lambda}_1 & 1 & 0 & 0 \\ -1 & -\boldsymbol{\Lambda}_2 & 0 & 0 \\ 0 & 0 & -\boldsymbol{\Lambda}_3 & \boldsymbol{\rho}(\boldsymbol{\Theta}) \\ 0 & 0 & -\boldsymbol{\rho}(\boldsymbol{\Theta})^T & -\boldsymbol{\Lambda}_4 \end{bmatrix} \begin{bmatrix} \boldsymbol{\eta}_e \\ \boldsymbol{\mu}_e \\ \boldsymbol{\Theta}_e \\ \boldsymbol{\omega}_e \end{bmatrix} - \begin{bmatrix} 0 \\ \frac{T}{m}\boldsymbol{Q}(\boldsymbol{\Theta}_e, \boldsymbol{\alpha}_{\Theta})\boldsymbol{k} \\ \dot{\tilde{\boldsymbol{\alpha}}}_{\Theta} \\ \dot{\tilde{\boldsymbol{\alpha}}}_{\omega} \end{bmatrix} \tag{4.57}$$

Comme nous l'avons indiqué à la section 3.1, la matrice du système en boucle fermé est la somme d'une matrice dont la diagonale est négative et d'une matrice antisymétrique.

La fonction candidate de Lyapunov associée au système est la suivante :

$$V_{BF} = \frac{1}{2}{\boldsymbol{\eta}_e}^T\boldsymbol{\eta}_e + \frac{1}{2}{\boldsymbol{\mu}_e}^T\boldsymbol{\mu}_e + \frac{1}{2}{\boldsymbol{\Theta}_e}^T\boldsymbol{\Theta}_e + \frac{1}{2}{\boldsymbol{\omega}_e}^T\boldsymbol{I}\boldsymbol{\omega}_e \tag{4.58}$$

[29] La preuve de stabilité du système complet est présentée à l'ANNEXE III.

Sa dérivée par rapport au temps est :

$$\begin{aligned}\dot{V}_{BF} = &-\boldsymbol{\eta_e}^T\boldsymbol{\Lambda_1}\boldsymbol{\eta_e} - \boldsymbol{\mu_e}^T\boldsymbol{\Lambda_2}\boldsymbol{\mu_e} - \boldsymbol{\Theta_e}^T\boldsymbol{\Lambda_3}\boldsymbol{\Theta_e} - \boldsymbol{\omega_e}^T\boldsymbol{\Lambda_4}\boldsymbol{\omega_e} \\ &- \boldsymbol{\mu_e}^t\frac{T}{m}\boldsymbol{Q}(\boldsymbol{\Theta_e}, \boldsymbol{\alpha_\Theta})\boldsymbol{k} - \boldsymbol{\Theta_e}^T\tilde{\dot{\boldsymbol{\alpha}}}_\Theta - \boldsymbol{\omega_e}^T\boldsymbol{I}\tilde{\dot{\boldsymbol{\alpha}}}_\omega\end{aligned} \tag{4.59}$$

Suite aux développements des sections 4.4 et 4.5, nous concluons que le système en boucle fermée est asymptotiquement stable au sens de Lyapunov et est ISS par rapport aux perturbations, $\frac{T}{m}\boldsymbol{Q}(\boldsymbol{\Theta_e}, \boldsymbol{\alpha_\Theta})\boldsymbol{k}$, $\tilde{\dot{\boldsymbol{\alpha}}}_\Theta$ et $\tilde{\dot{\boldsymbol{\alpha}}}_\omega$. La preuve de ISS est présentée à l'ANNEXE III.

### 4.6.2 Signaux de contrôles

Tous les signaux nécessaires au contrôle sont résumés ici. Les erreurs sont calculées ainsi :

$$\begin{aligned}\boldsymbol{\eta_e} &= \boldsymbol{\eta} - \boldsymbol{\eta_d} \\ \boldsymbol{\mu_e} &= \boldsymbol{\mu} - \boldsymbol{\alpha_\mu} \\ \boldsymbol{\Theta_e} &= \boldsymbol{\Theta} - \boldsymbol{\alpha_\Theta} \\ \boldsymbol{\omega_e} &= \boldsymbol{\omega} - \boldsymbol{\alpha_\omega}\end{aligned} \tag{4.60}$$

Où les commandes virtuelles sont données par :

$$\begin{aligned}\boldsymbol{\alpha_\mu} &= \dot{\boldsymbol{\eta}}_d - \boldsymbol{\Lambda_1}\boldsymbol{\eta_e} \\ \boldsymbol{\alpha_a} &= [\alpha_{a_x} \quad \alpha_{a_y} \quad \alpha_{a_z}]^T = \boldsymbol{\eta_e} + g\cdot\boldsymbol{k} - \ddot{\boldsymbol{\eta}}_d + \boldsymbol{\Lambda_1}(\boldsymbol{\mu_e} - \boldsymbol{\Lambda_1}\boldsymbol{\eta_e}) + \boldsymbol{\Lambda_2}\boldsymbol{\mu_e} \\ \boldsymbol{\alpha_\omega} &= \boldsymbol{\rho}(\boldsymbol{\Theta})^{-1}\left[\hat{\dot{\boldsymbol{\alpha}}}_\Theta - \boldsymbol{\Lambda_3}\boldsymbol{\Theta_e}\right] \\ \alpha_\phi &= \sin^{-1}\left(\frac{\alpha_{a_x}s_{\alpha_\psi} - \alpha_{a_y}c_{\alpha_\psi}}{\sqrt{\boldsymbol{\alpha_a}^T\boldsymbol{\alpha_a}}}\right) \\ \alpha_\theta &= \tan^{-1}\left(\frac{\alpha_{a_x}c_{\alpha_\psi} + \alpha_{a_y}s_{\alpha_\psi}}{\alpha_{a_z}}\right) \\ \alpha_\Theta &= [\alpha_\phi \quad \alpha_\theta \quad \psi_d]^T\end{aligned} \tag{4.61}$$

Les quatre sorties du contrôleur sont :

$$T = m\sqrt{\boldsymbol{\alpha_a}^T \boldsymbol{\alpha_a}}$$
$$\boldsymbol{\tau} = -\boldsymbol{\rho}(\boldsymbol{\Theta})^t \boldsymbol{\Theta}_e + \boldsymbol{\omega} \times \boldsymbol{I\omega} + \boldsymbol{G}_r + \boldsymbol{I}\widehat{\boldsymbol{\alpha}}_\omega - \boldsymbol{\Lambda}_4 \boldsymbol{\omega}_e \quad (4.62)$$

Les estimateurs, où l'indice $i \in \{\phi \quad \theta \quad p \quad q \quad r\}$ indique le contrôleur virtuel associé à l'estimateur sont :

$$\hat{\dot{\alpha}}_i = \xi_i - \lambda_i \sqrt{|\alpha_i - \hat{\alpha}_i|} \cdot \text{sign}(\alpha_i - \hat{\alpha}_i)$$
$$\dot{\xi}_i = -\beta_i \cdot \text{sign}(\alpha_i - \hat{\alpha}_i) \quad (4.63)$$

Nous avons présenté dans ce chapitre la construction du contrôleur backstepping. Celle-ci a été effectuée en quatre étapes soit : le contrôle de la position linéaire, le contrôle de la vitesse linéaire, le contrôle de la position angulaire et le contrôle de la vitesse angulaire. À chaque étape, une fonction candidate de Lyapunov a été proposée. Ensuite, un contrôleur virtuel a été construit dans l'optique de forcer la dévirée par rapport au temps de la fonction de Lyapunov à être une fonction définie négative. Lors des étapes 3 et 4, il nous a été nécessaire d'utiliser estimateur de dévirée pour estimer la dévirée des contrôleurs virtuels puisque le résultat analytique de celle-ci était trop complexe. Le contrôleur obtenu respecte nos objectifs tels que présentés à la section 4.2. Une analyse de la stabilité de la boucle fermée par rapport aux perturbations générées par l'erreur d'alignement $\frac{T}{m} Q(\Theta_e, \alpha_\Theta) k$ ainsi que par rapport aux erreurs d'estimation $\widetilde{\boldsymbol{\alpha}}_\Theta$ et $\widetilde{\boldsymbol{\alpha}}_\omega$ est présentée à l'ANNEXE III.

# CHAPITRE 5

# IMPLÉMENTATION

Ce chapitre présente l'implémentation du contrôleur backstepping au niveau d'un quadrotor de type Pelican conçu par la compagnie Ascending Technologies que nous désignerons dans ce qui suit par le nom d'Asctec Pelican. Nous commencerons par présenter les unités de calcul ainsi que les capteurs embarqués sur le quadrotor puisque ceux-ci ont une influence directe sur l'architecture du contrôleur choisi. Par la suite, nous la présenterons en détail ainsi que le support logiciel utilisé pour effectuer l'implémentation.

## 5.1 Quadrotor Asctec Pelican

Comme mentionné précédemment, Ascending Technologies est une compagnie allemande de pointe dans la conception de mini-UAV de type quadrotor, dont plusieurs appareils sont dédiés à la recherche. Les particularités de ces produits sont la flexibilité, la modularité ainsi que l'existence de cadriciels permettant l'ajout rapide de code. Parmi la multitude d'appareils conçus par Ascending Technologies, nous avons choisi le Pelican puisqu'il s'agit du modèle de quadrotor offrant la meilleure charge utile.

### 5.1.1 Unités de calcul

Le Pelican possède plusieurs unités de calcul. La carte principale, présentée par la Figure 5.1, comprend deux microprocesseurs ARM LPC2146, soit le processeur haut niveau (processeur HL) et bas niveau (processeur LL). Le processeur LL gère l'acquisition de l'ensemble des capteurs de l'appareil, la fusion de données, la communication via Zigbee, la commande des moteurs et l'exécution du contrôleur conçu par Ascending Technologies. Le code au niveau du processeur LL est fermé et ne peut être modifié. Le processeur HL est ouvert aux modifications et est dédié à l'implémentation de code supplémentaire en l'occurrence le code du contrôleur backstepping conçu dans ce mémoire. Les deux processeurs communiquent

entre eux à l'aide d'un lien de type Serial Peripheral Interface Bus (SPI). Celui-ci permet d'acheminer les données des capteurs fusionnées du processeur LL vers le processeur HL et les commandes du processeur HL vers le processeur LL.

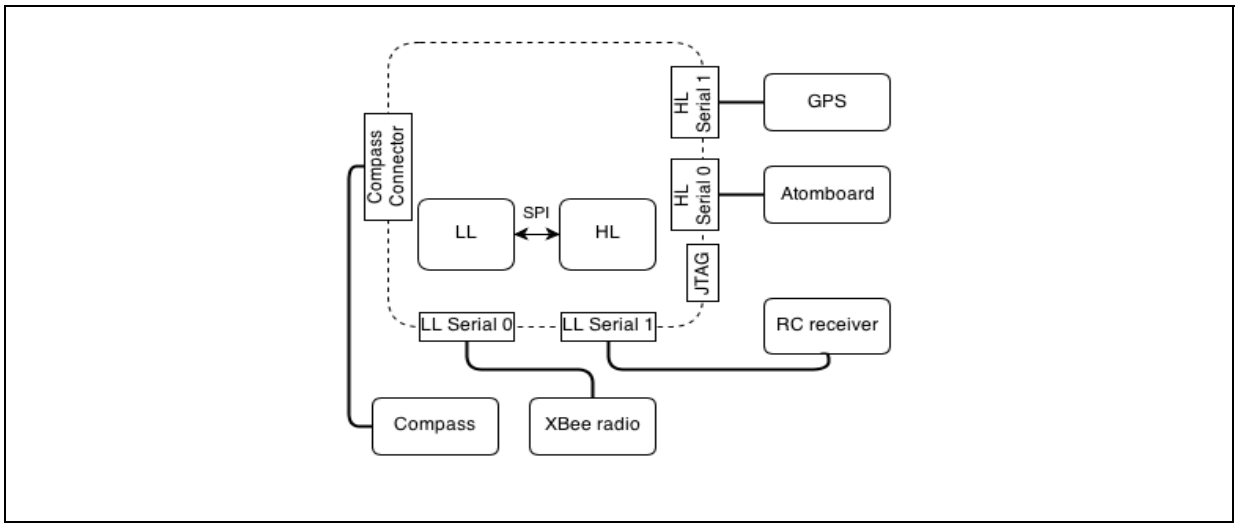



Figure 5.1 Configuration matérielle de la carte électronique du Asctec Pelican[30]

### 5.1.2 Ordinateur embarqué

Un ordinateur embarqué (On-Board Computer - OBC) muni d'un processeur Intel Atom 1.6Ghz est aussi disponible, sur le Pelican, pour effectuer des calculs plus complexes. L'ordinateur embarqué communique au processeur HL via un lien série, noté HL serial 0 sur la Figure 5.1. L'ensemble des systèmes et leur lien de communication sont illustrés par la Figure 5.2. Pour plus de détails sur le Pelican, l'auteur recommande au lecteur de consulter les manuels d'utilisation d'Ascending Technologies tout en notant que certaines modifications ont été apportées au système original. Celles-ci sont présentées dans les sections subséquentes.

[30] Il est important de noter qu'il s'agit du schéma du système tel qu'originalement offert par Ascending Technologies. Dans notre configuration, le module GPS n'est pas connecté sur le port HL serial 1. Image tirée de (Brunelle, 2013).

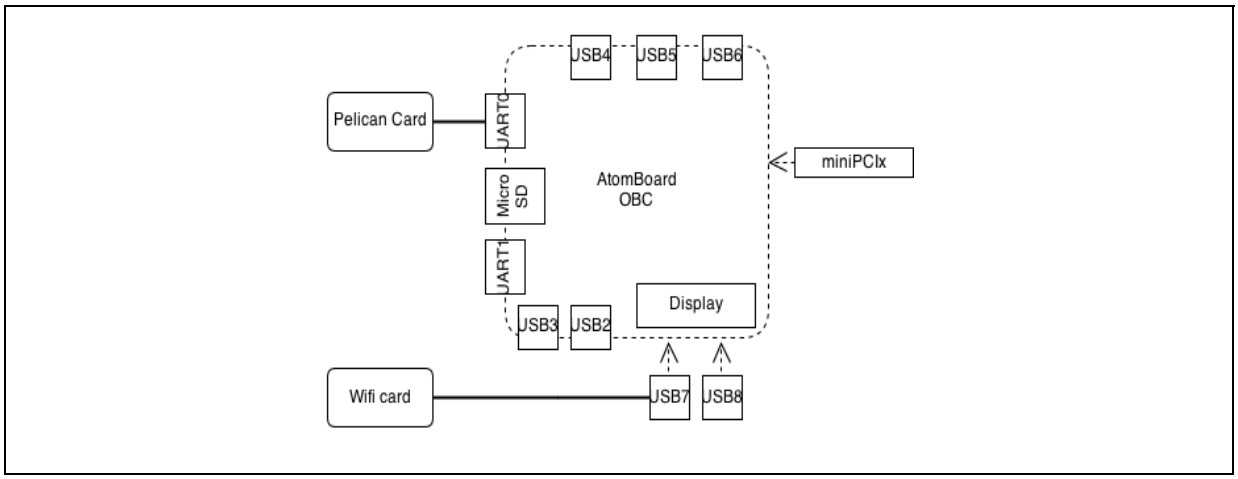



Figure 5.2 Configuration matérielle et photo de l'ordinateur Atom d'Ascending Technologies[31]

### 5.1.3 Capteurs

Le Pelican dispose de tous les capteurs nécessaires pour le vol. Les capteurs principaux sont les triplets de capteurs inertiels, soit les accéléromètres, les gyromètres ainsi que les magnétomètres qui ensemble forme une centrale inertielle (Inertial Measurement Unit - IMU). Un filtre de Kalman effectue la fusion des données inertielles et permet d'obtenir l'orientation du quadrotor sous la forme des angles d'Euler ainsi que la vitesse angulaire mesurée dans le repère du quadrotor. Les capteurs du Pelican mesurent également la vitesse des moteurs, la tension courante de la batterie ainsi que les signaux provenant de la télécommande. L'acquisition des capteurs ainsi que le calcul du filtre de Kalman sont effectués par le processeur LL et l'ensemble des mesures est transmis vers le processeur HL à une fréquence de 1KHz.

Un récepteur GPS, fourni avec le Pelican, permet de mesurer la position du quadrotor. Celle-ci est fusionnée avec les données inertielles pour obtenir une meilleure estimation de la position et à une fréquence plus élevée. Pour augmenter la précision du système, nous avons

[31] Image tirée de (Brunelle, 2013).

choisi d'utiliser un autre capteur couplant un receveur GPS ainsi qu'un système de navigation inertiel (Inertial Navigation System - INS) de la compagnie Advanced Navigation nommé « Spatial ». Les particularités de ce capteur sont présentées à la section 5.2.

### 5.1.4 Repère

Les capteurs du Pelican effectuent leurs mesures en fonction du repère quadrotor, tel que défini à la section 2.1. C'est-à-dire que l'axe $\boldsymbol{i}_b$ pointe vers l'avant, $\boldsymbol{j}_b$ pointe vers le bras droit et $\boldsymbol{k}_b$ pointe vers le bas. Le bras avant du Pelican est identifié à l'aide d'une bande rouge.

### 5.1.5 Alimentation

Nous avons à notre disposition quatre batteries Lithium-Polymère (LIPO) pour alimenter le quadrotor, soit deux de 6100 mAh, une de 6000mAh et une de 8000 mAh. En moyenne, nous pouvons obtenir une autonomie de vol de 15 minutes.

### 5.1.6 Architecture logiciel Asctec SDK

La compagnie Ascending Technologies fournit plusieurs outils de développement pour effectuer l'implémentation au niveau du processeur HL. Celui qui nous intéresse particulièrement est l'ensemble de développement logiciel (Asctec Software Development Kit – Asctec SDK). Celui-ci fournit des structures de données standardisées permettant au code du processeur HL de communiquer avec le processeur LL. Il fournit aussi l'ensemble des outils pour s'assurer que le code implémenté s'exécute à une période fixe prédéterminée.

La structure principale nommée «RO_ALL_DATA» contient toutes les données mesurées par les capteurs ainsi que le résultat du filtre de Kalman calculé par le processeur LL. La structure « WO_CTRL_INPUT » est dédiée au contrôle du quadrotor. Celle-ci comprend les

modes de contrôle suivants : le contrôle direct des moteurs, le contrôle par force/moment ou le contrôle à l'aide de l'autopilote du processeur LL.

## 5.2 Capteur GPS/INS Spatial de la compagnie Advanced Navigation

Comme précisé précédemment, nous avons choisi de remplacer le système de mesure GPS/INS du Pelican par un capteur plus performant. En effet, les spécifications du système de positionnement du Pelican indiquent une précision de 3m Root Mean Square (RMS) selon le plan vertical et une précision de 5m RMS selon le plan horizontal [32]. Or, pour assurer le fonctionnement d'un contrôleur performant tel que le backstepping, ce niveau de précision peut s'avérer insuffisant.

Pour obtenir de meilleures performances, nous avons choisi d'utiliser le capteur GPS/INS Spatial de la compagnie Advanced Navigation. Les spécifications de ce capteur indiquent une précision de 2m RMS en horizontal et 3m RMS en vertical. De plus, celui-ci peut être couplé à un système GPS différentiel (DGPS). Dans cette configuration, les performances du capteur augmentent de telle sorte que l'on obtient une précision de 0.6m RMS selon le plan horizontale et une précision de 1 m RMS selon le plan vertical. Une comparaison de la performance des deux systèmes de mesure est présentée à l'ANNEXE VII.

Puisqu'il comprend un système inertiel complet, le Spatial, en plus d'effectuer la mesure de la position, permet de mesurer l'orientation et la vitesse angulaire de l'appareil; ceci crée une redondance de données que nous devons gérer.

---

[32] Il est important de noter que les spécifications de système GPS/INS ne sont atteintes que dans les meilleures conditions possibles, soit lorsque la géométrie satellitaire est optimale et lorsque les erreurs de mesure GPS, dues principalement aux effets atmosphériques, sont minimales. De plus, compte tenu de la distribution spatiale des satellites, la précision de la mesure horizontale est généralement moins élevée que la verticale.

### 5.2.1 Traitement de la redondance des mesures

Avec l'ajout du capteur Spatial, nous avons créé une redondance des mesures de position angulaire et de vitesse angulaire[33]. Puisque les mesures proviennent de deux capteurs différents, il faut s'assurer que celles-ci soient cohérentes entre elles et, dans le cas contraire, choisir la mesure appropriée. Après vérification, les données mesurées par les deux capteurs est cohérent, sauf la mesure du lacet. Il a donc été choisi d'utiliser la mesure du lacet provenant du processeur LL puisque celle-ci est beaucoup plus précise et corrige adéquatement le facteur de déclinaison magnétique[34].

### 5.2.2 Système GPS différentiel (DGPS)

Typiquement, un système DGPS fonctionne à l'aide de deux récepteurs GPS. L'un des récepteurs est à une position fixe connue très précisément tandis que le deuxième est embarqué sur la plate-forme que l'on cherche à positionner. Puisque le receveur fixe connaît sa position, il peut estimer les erreurs de mesure des signaux GPS et calculer les corrections nécessaires pour augmenter la précision de la mesure. Les corrections sont ensuite acheminées au deuxième receveur à l'aide de message standardisé par le Radio Technical Commission for Maritime Services ou simplement RTCM.

Dans notre implémentation du système, le récepteur en mouvement est le capteur Spatial et le récepteur fixe est le OEM628 de la compagnie Novatel. Celui-ci génère les trois trames nécessaires pour effectuer une correction DGPS valide, soit les trames RTCM1 (correction DGPS), RTCM3 (paramètres du récepteur) et RTCM31 (correction différentielle GLONASS). Lors du déploiement du système, le récepteur OEM628 est connecté à une antenne fixe et un ordinateur.

---

[33] Nous ne remplaçons que le capteur GPS du Pelican de tel sorte que l'appareil conserve sa centrale inertielle.
[34] La mesure du Spatial est influencé davantage que celle du magnétomètre de Asctec par le bruit des moteurs compte tenu de sa proximité avec ceux-ci. De plus, l'architecture de contrôle a influencé notre décision en gardant comme philosophie de positionner les capteurs les plus près possible des unités de calcul qui en ont besoin. Ceci est présenté plus en détail dans les sections subséquentes.

### 5.2.3 Repère inertiel et conversion de la mesure de position

La mesure de la position effectuée par le Spatial est exprimée par rapport au repère Earth Centered Earth Fixed (ECEF) à l'aide de coordonnées sphériques, soit le triplet latitude, longitude et altitude aussi dénommé « la position LLA » (Farrel, 2008). Puisque le contrôleur a été conçu en utilisant un repère de référence NED, il est nécessaire d'effectuer une conversion de la mesure de la position LLA vers un repère NED local. Cette conversion utilise deux positions LLA, soit la position de l'origine du repère inertiel NED et la position courante du capteur. Comme indiqué précédemment, la position de l'origine du repère inertiel est choisie comme étant la position initiale du quadrotor. Les détails de la conversion sont donnés à l'ANNEXE V.

L'orientation du repère inertiel NED dépend de la position du nord géographique déterminée par le magnétomètre du quadrotor. Or, un magnétomètre mesure la position du nord magnétique et non le nord géographique. Il faut donc que le capteur et l'algorithme de fusion de données tiennent compte de la déclinaison magnétique. Une incohérence entre l'orientation du repère inertiel et le nord géographique peut considérablement influencer les performances du contrôleur[35].

Selon Transport Canada, la déclinaison magnétique a une valeur de -15.064˚ dans la région de Montréal. La déclinaison a été ajoutée lors de la calibration des capteurs du Pelican[36]. L'orientation du repère inertiel a ensuite été validée expérimentalement à l'aide de deux boussoles en positionnant l'axe $\boldsymbol{i}_b$ du quadrotor vers le nord géographique et en confirmant une mesure de 0 ˚ au niveau du lacet.

---

[35] De façon générale, cette erreur s'identifie en pratique lorsque le quadrotor effectue des mouvements circulaires plus ou moins prononcés lorsqu'il est en mode stationnaire.

[36] Le Spatial comprend son propre modèle interne du champ magnétique terrestre et effectue la correction pour la déclinaison automatiquement.

### 5.2.4 Acquisition des données de vol via le capteur Spatial

Le Spatial utilise deux liens de communications séries. L'un est dédié à la transmission des données choisies et à la réception de commande tandis que le deuxième est dédié à la transmission des trames RTCM. Le Spatial a été programmé pour retourner l'ensemble des données de vol, soit la position angulaire, la vitesse angulaire, la position ainsi que la vitess; le tout, à une fréquence de 250Hz. Compte tenu du haut débit de communication nécessaire, il a été choisi d'utiliser une vitesse de transmission de 921600 baud/s. Les trames RTCM doivent, quant à elles, être transmises environ à chaque seconde pour assurer une correction adéquate des erreurs de mesure GPS.

## 5.3 Robotic Operating System (ROS)

La compagnie Willow Garage développe depuis quelques années un métasystème d'exploitation libre dédié à la robotique nommée Robotic Operating System (ROS). Celui-ci fournit plusieurs outils et implémentations de systèmes pour accélérer le déploiement de systèmes robotiques, comme le traitement de transformation de repère, la gestion d'échange de messages, l'abstraction matérielle, la gestion de capteurs, le traitement multimachine, etc. Compte tenu de la versatilité des outils déjà disponibles sous ROS et de la familiarité de l'auteur avec ce système, il a été choisi d'utiliser ROS pour effectuer l'implémentation du contrôleur ainsi que du système de vol. Cette section a pour but de décrire brièvement le fonctionnement de ROS.

### 5.3.1 Principe de fonctionnement

Chaque programme s'exécutant sous ROS est appelé une « node ». Chaque node s'exécute en parallèle et est indépendante des autres nodes. Pour échanger de l'information, chaque node peut publier (transmettre) ou s'inscrire (recevoir) des messages. Un processus unique, nommé « roscore », nécessaire dans chaque utilisation de ROS, gère la gestion des messages. Celui-ci connaît l'ensemble des messages publiés ou inscrits, par les nodes et effectuent la

liaison entre celles-ci à l'aide de fonctions de rappel. Le roscore est toujours actif, de telle sorte que l'ajout de messages publiés ou l'inscription à de nouveaux messages peut changer dynamiquement au cours de l'exécution du système. L'ensemble des communications ROS s'effectue via le protocole TCP/IP, de telle sorte qu'un système ROS peut-être facilement déployé sur plusieurs machines et ceci est transparent pour l'ensemble des nodes. En effet, il suffit pour cela que l'ensemble des nodes connaissent l'adresse IP du roscore. Celui-ci assure ensuite la connexion entre les différentes nodes à travers le réseau ROS. Chaque message ROS possède un nom unique commençant par le symbole « / ». Lorsqu'un message n'est pas unique à une seule node, on ajoute typiquement le nom de la node avant le nom du message, par exemple « /nom_node/nom_message ».

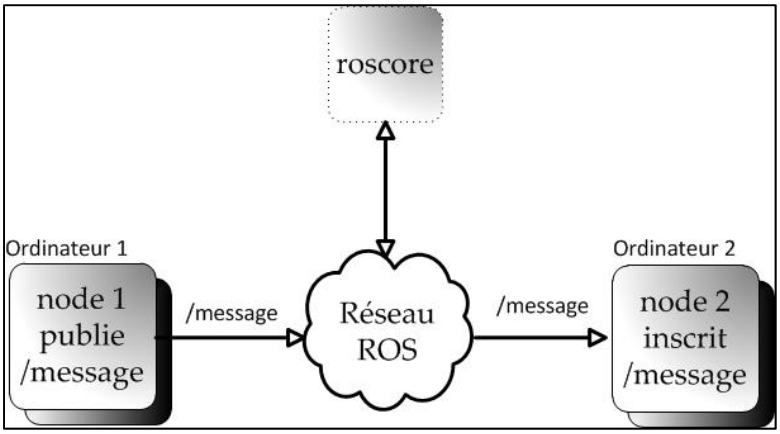



Figure 5.3 Exemple d'un système ROS simple

### 5.3.2 Temps réel

Il est important de noter que ROS n'est pas un système temps réel dur. En effet, ROS est typiquement utilisé sous un système d'exploitation de type LINUX, dans notre cas Ubuntu 12.04 LTS, de telle sorte que la gestion de l'exécution de l'ensemble des nodes ROS ainsi que le roscore sont dépendants du système d'exploitation. Néanmoins, toutes les nodes peuvent être configurées pour agir comme des processus prioritaires. Après plusieurs tests de faisabilité, nous avons conclu que puisque les processus développés dans le cadre de ce projet demandent peu de puissance de calcul et compte tenu de la puissance de l'OBC, nous

pouvions atteindre des performances temps réel suffisantes tant que l'OBC n'est pas surchargé. En effet, nous obtenons une précision de l'ordre des dixièmes de Hz lorsque le processeur de l'OBC est à 50% de sa capacité lors du fonctionnement du système de vol complet.

### 5.3.3 Outils

Plusieurs outils sont déjà implémentés sous ROS pour accélérer le développement. Les outils principaux utilisés lors du projet sont nommés « rosbag » et « rqt_gui ». La node rqt_gui offre la possibilité de s'inscrire à tous les messages publiés sur le réseau ROS et d'afficher leur contenu, leur fréquence ainsi que leur bande passante. De plus, le node rqt_gui offre la possibilité de publier n'importe quel message ROS à une fréquence voulue ou lors de l'appui d'un bouton par l'utilisateur. Cette fonctionnalité est très utile pour ajouter rapidement une interface usagée au système ou pour effectuer des tests. La node ROS rosbag permet, quant à elle, d'enregistrer les messages publiés sur le réseau ROS dans une structure de données nommées « bag ». Chaque bag peut-être revisionné à l'aide de l'outil rxbag. De plus, les données contenues dans le fichier bag peuvent être extraites dans un fichier texte pour effectuer d'autres analyses.

Plusieurs autres outils ROS, tel que « rostopic » (outil de gestion des messages ROS), « rxgraph » (outil d'analyse du réseau ROS) et « rosparam » (outil de gestion des paramètres), pour ne nommer que ceux-ci, sont disponibles pour valider l'implémentation sous ROS.

## 5.4 Implémentation du contrôleur backstepping

L'implémentation de l'autopilote backstepping a été divisée en deux parties qui ont été implémentées sur des unités de calcul différentes : l'utilisation des capteurs, la proximité de ceux-ci par rapport aux unités de calcul qui les utilisent ainsi que la puissance de calcul nécessaire sont les principaux facteurs qui ont influencé cette décision. En effet, l'ajout du

Spatial au Pelican est l'élément principal qui a influencé l'ensemble de nos choix architecturaux.

Comme mentionné précédemment, le capteur Spatial demande une grande bande passante, ce qui n'était pas possible d'obtenir au niveau du port série du processeur HL. De plus, le Spatial demande l'utilisation de deux ports séries, lors de son utilisation avec un DGPS, tandis que la carte du processeur HL ne nous en fournit qu'un seul. Ceci explique son installation au niveau de l'OBC.

En nous basant sur les équations de la section 4.6, nous pouvons constater que le contrôleur backstepping se divise facilement en deux parties, en séparant le contrôleur de position et le contrôleur d'attitude de l'appareil. Ceci a été effectué auparavant dans plusieurs implémentations telles que le prototype réalisé par le Massachusetts Institute of Technology (MIT) (Achtelik, Bachrac *et al.*, 2009). Puisque le Spatial est connecté à l'OBC, nous avons choisi d'implémenter le contrôleur de position au niveau de l'OBC. Le contrôleur d'attitude est, quant à lui, implémenté au niveau du processeur HL puisque cette partie du contrôleur est critique et qu'il est garanti d'être exécuté en temps réel dur sur le processeur ARM.

### 5.4.1 Contrôleur de position

Le contrôleur de position est implémenté au niveau de l'OBC. La figure suivante présente un schéma conceptuel des liens de communication utilisés pour l'implémentation.

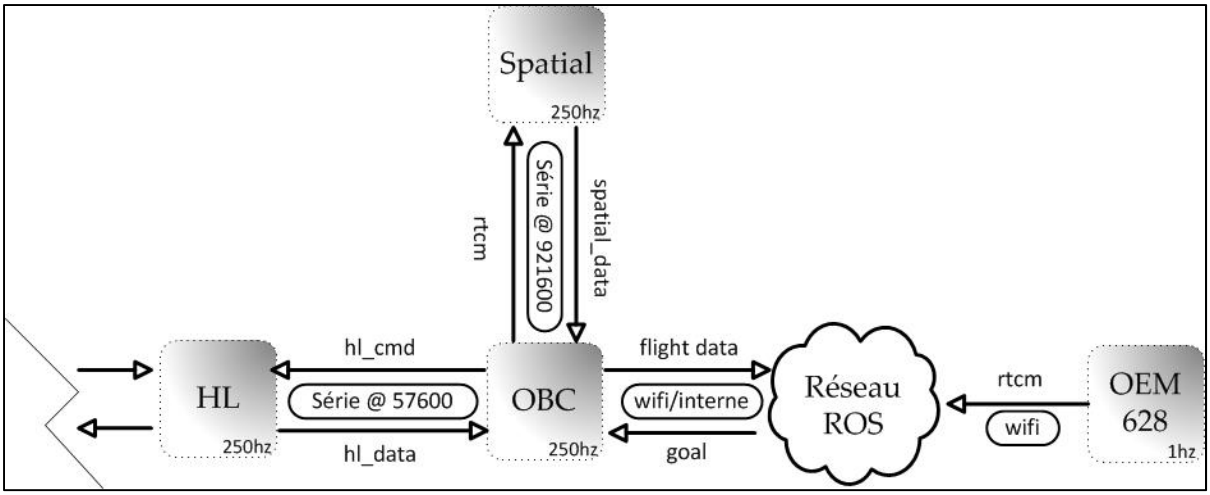



Figure 5.4 Schéma conceptuel du contrôleur de position[37]

Le contrôleur de position effectue les calculs des contrôleurs virtuels $\boldsymbol{\alpha}_{\mu}$, $\boldsymbol{\alpha}_{a}$, $\alpha_{\phi}$, $\alpha_{\theta}$ et de la puissance de poussée totale $T$ tel qu'indiqué par les équations (4.61) et (4.62) à partir des données de vol provenant du capteur Spatial et des trajectoires désirées calculées à partir des positions finales désirées reçues du réseau ROS. Les commandes virtuelles $\alpha_{\phi}$ et $\alpha_{\theta}$ la valeur désirée du lacet $\psi_{d}(t)$ en plus de la poussée totale $T$ sont ensuite envoyées vers le processeur HL pour effectuer le contrôle de l'attitude du quadrotor.

Tout comme l'ensemble des processus implémentés au niveau de l'OBC, le contrôleur de position est intégré à l'aide d'une node ROS nommée « Mobility Control Unit » (MCU). L'ensemble du système ROS implémenté au niveau de l'OBC est présenté à la section 5.6.

### 5.4.2 Contrôleur d'attitude

Le contrôleur d'attitude est implémenté au niveau du processeur HL en suivant les instructions définies dans le Asctec SDK. La Figure 5.5 présente un schéma conceptuel des liens de communication utilisés pour l'implémentation.

[37] La partie à gauche du schéma est complétée par la Figure 5.5.

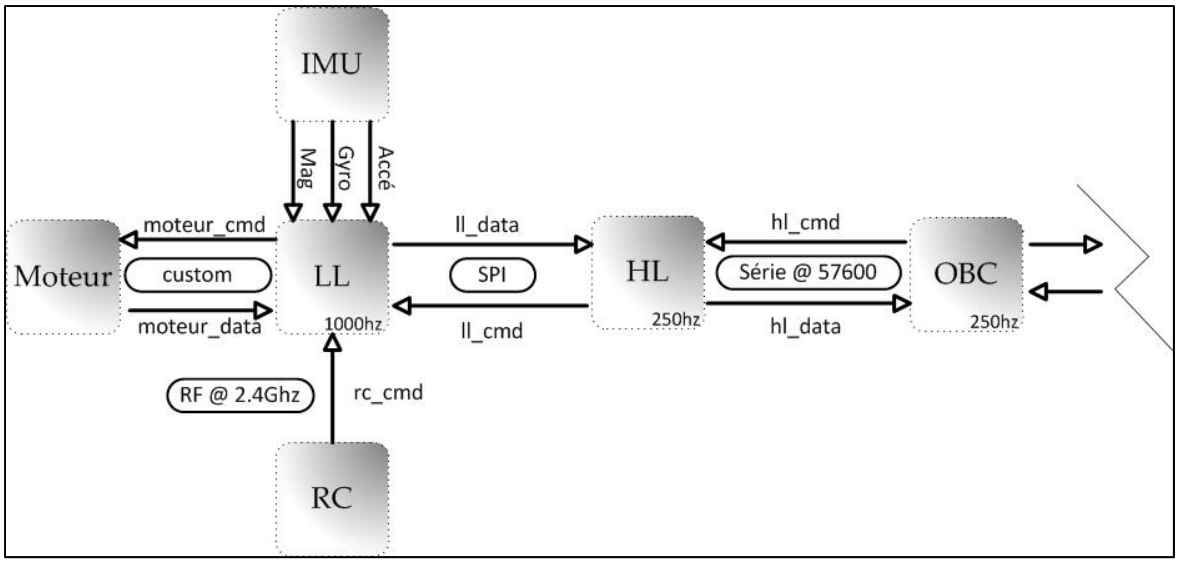



Figure 5.5 Schéma conceptuel du contrôleur d'attitude

Le contrôleur d'attitude effectue le calcul du contrôleur virtuel $\boldsymbol{\alpha}_{\omega}$, du vecteur de couple $\boldsymbol{\tau}$ ainsi que le calcul de tous les estimateurs en se basant sur les équations de la section 4.6.2. L'ensemble du code du contrôleur est inclus dans la fonction « SDK_mainloop », conçue pour être exécutée à une fréquence de 1KHz; cependant, il a été choisi de l'exécuter à 250Hz pour être synchronisé avec le contrôleur de position. Le traitement d'une exécution du code peut être résumé par le pseudocode suivant :

SDK_mainloop()

```
1.  soit data, data_SI        // Contient les données de vol
2.  soit cmd_RC               // Structure qui contient les commandes de la télécommande
3.  soit cmd_LL               // Commande à envoyer au processeur LL
4.  soit cmd_ctrl             // Structure qui contient les commandes pour le contrôleur
5.  soit cmd_OBC              // Structure qui contient les commandes de l'OBC
6.  soit hl_obc_status        // Status envoyé à l'OBC
7.
8.  data = read_LL_donnees()
9.  data_SI = convert_2_SI(data)
10. si cmd_RC->hl_actif est vrai
11.             cmd_LL->hl_actif = vrai
12.             cmd_LL->hl_ctrl_mode = force/moment
13.             si cmd_RC->auto est vrai
14.                     hl_obc_status = auto
```

```
15.                 cmd_ctrl = cmd_OBC
16.             sinon
17.                 hl_obc_status = manuel
18.                 cmd_ctrl = cmd_RC
19.             fin
20.             cmd_LL->force_moment = execution_ctrl_attitude(cmd_ctrl , data_SI)
21.     sinon
22.             cmd_LL->hl_actif = faux
23.     fin
24.     send_cmd_2_LL(cmd_LL)
25.     send_message_2_OBC(cmd_OBC , data_SI)
```

Le schéma de principe (Figure 5.5) et le pseudocode précédents résument le fonctionnement du contrôleur d'attitude. Le processeur LL recueille l'ensemble des mesures des capteurs, les signaux provenant de la télécommande (Remote Control - RC) ainsi que la vitesse des moteurs et les transmet via SPI vers le processeur HL. Une fois l'appel de la fonction SDK_mainloop complété, le processeur HL transmet les commandes vers le processeur LL.

Nous avons choisi d'utiliser le mode de contrôle force/moment de telle sorte que nous spécifions la force totale générée par les moteurs ainsi que les moments désirés autour des trois axes principaux et le processeur LL s'assure de faire la correspondance nécessaire pour obtenir la vitesse des moteurs à appliquer. En d'autres termes, le processeur LL s'assure d'effectuer le calcul de l'équation (2.35). Ceci élimine la nécessité de mesurer expérimentalement les paramètres $b$, $d$ et $l$.

Pour assurer au pilote un contrôle complet sur le système, deux boutons de la télécommande sont dédiés à la gestion du contrôleur. Ceux-ci sont liés à des canaux particuliers. Le canal « 4 » permet d'activer le contrôleur d'attitude implémenté au niveau du processeur HL. Ce canal indique au processeur LL d'accepter les commandes provenant du contrôleur HL en utilisant le mode choisi[38]; ceci peut être utile en cas de défaillance du contrôleur backstepping. Le canal « 5 » permet, quant à lui, de spécifier si le contrôleur d'attitude

[38] Dans notre cas force/moment

fonctionne en mode autonome ou en mode manuel. Lorsque le contrôleur d'attitude fonctionne en mode manuel, les entrées de commande, soient le roulis, le tangage, le lacet et la puissance des moteurs totale, proviennent de la télécommande, tandis qu'en mode automatique, ses commandes proviennent du contrôle de position, implémenté au niveau de l'OBC.

Le processeur HL achemine aussi des messages vers l'OBC. En effet, l'ensemble des données de vol de l'appareil est envoyé vers l'OBC pour être transmis sur le réseau ROS. Les deux messages qui nous intéressent ici sont le message de statut du contrôleur, nommé « /fcu/status », qui indique au contrôleur de position si le contrôleur d'attitude est en mode autonome ou en mode manuel pour effectuer des décisions ainsi que le message « fcu/yaw_feedback » qui transmet la mesure du lacet vers l'OBC[39].

## 5.5 Implémentation de la génération de trajectoire

Dans des applications robotiques, une génération de trajectoire est habituellement utilisée pour diriger un mobile en éliminant les écarts trop élevés entre la valeur désirée et l'état contrôlé. En effet, une erreur trop grande entre ces deux valeurs peut entraîner une saturation au niveau de la commande ou même, dans certains cas, déstabiliser le système.

Comme présenté à la section 4.2, le contrôleur doit être en mesure d'effectuer la poursuite de quatre trajectoires, soit une par degré de liberté contrôlée. Nous avons implémenté une trajectoire parabolique par partie, telle que décrite par (Biagiotti et Melchiorri, 2009)[40]. Un exemple de ce type de trajectoire est illustré par la Figure 5.6.

---

[39] Ceci est présenté plus en détail à la section 5.6.

[40] Nous ne tenons pas en compte ici des contraintes, définies à la section 4.2, applicables aux trajectoires. En effet, l'assemblage des trajectoires paraboliques, qui individuellement sont des fonctions de classe $C^3$, ne forme qu'une fonction de classe $C^0$ car elle est non dérivable aux de raccordement.

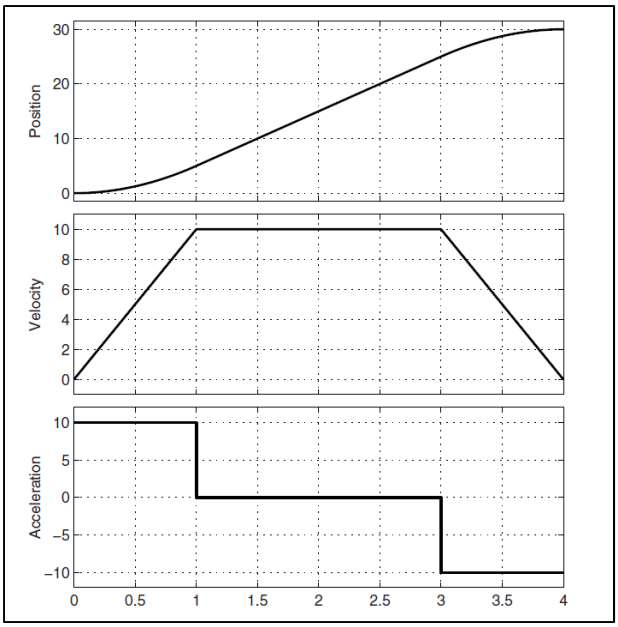



Figure 5.6 Exemple de trajectoire parabolique à trois parties[41]

La trajectoire est une fonction du temps divisée en trois phases : une phase d'accélération, une phase de transport et une phase de décélération. L'accélération maximale ainsi que la vitesse maximale atteinte lors de la phase de transport sont des paramètres de la trajectoire. À chaque fois qu'une nouvelle valeur finale désirée est demandée, le générateur de trajectoire calcule les coefficients définissant chaque phase de la trajectoire, le temps d'accélération/décélération requis et la durée totale de la trajectoire[42]. Ces calculs sont présentés à l'ANNEXE VI.

Lors de l'implémentation, nous avons choisi d'utiliser deux modes de guidage pour le quadrotor, soit un mode de guidage par position et un mode de guidage par vitesse[43]. Pour les deux modes, nous utilisons la même génération de trajectoire, mais les entrées ainsi que les sorties de celle-ci sont utilisées différemment.

---

[41] Cette figure est tirée de (Biagiotti et Melchiorri, 2009) p. 63
[42] Nous considérons la vitesse et l'accélération comme étant nulles à la fin d'une trajectoire.
[43] Le mode de guidage par vitesse a été implémenté pour être utilisé par la loi d'autonavigation conçue lors du projet « Launch and Forget »

### 5.5.1 Trajectoire à partir de la position finale désirée

Dans le mode de guidage par position, une trajectoire en fonction d'une position finale désirée est calculée pour chaque degré de liberté contrôlé. Puisque la mesure de la vitesse peut-être très bruitée pour de petites valeurs, il a été choisi de forcer la vitesse initiale de chaque trajectoire à zéro pour ne pas corrompre le calcul de la trajectoire.

### 5.5.2 Trajectoire à partir de la vitesse finale désirée

Puisque la loi d'autonavigation, développée dans le cadre du projet « Launch and Forget » génère des vitesses désirées, nous avons choisi d'implémenter un mode de guidage par vitesse. Puisque la loi d'autonagivation n'est valide qu'en deux dimensions, la génération des trajectoires pour les axes « X » et « Y » utilise les vitesses désirées tandis que la génération des trajectoires pour l'axe « Z » et le lacet utilise des positions désirées.

La génération de trajectoires est identique à celle présentée à la section 5.5.1 sauf que tous les paramètres, les entrées et les sorties « augmentent d'ordre »; c'est-à-dire que la position devient la vitesse, la vitesse devient l'accélération, etc. Dans ce cas, puisque la sortie de la trajectoire est désormais une vitesse désirée, nous ajoutons à la trajectoire un intégrateur pour obtenir la position désirée. Le processus de calcul est illustré par la Figure 5.7.

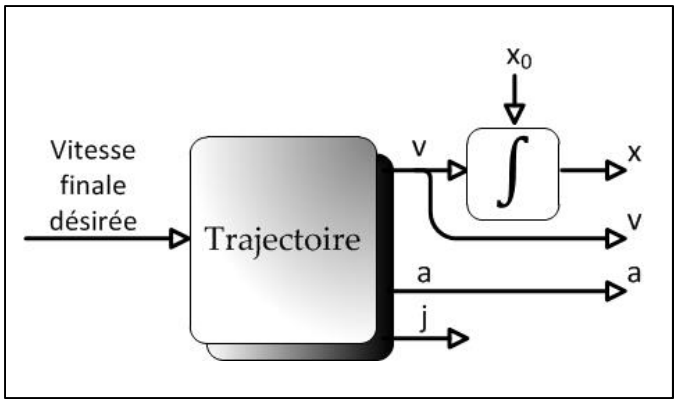



Figure 5.7 Calcul de la trajectoire[44]

La position initiale de la trajectoire est la position courante du quadrotor lors de la première génération de la trajectoire. La vitesse initiale ainsi que l'accélération initiale proviennent quant à elles du calcul de la trajectoire précédente.

## 5.6 Implémentation du système de vol sous ROS

Le système de vol comprend tous les processus nécessaires pour assurer le vol autonome du quadrotor. Celui-ci est implémenté par plusieurs nodes ROS aillant chacune un but précis. Nous présenterons ici le fonctionnement de chacune des nodes. Comme mentionné précédemment, l'ensemble des nodes communiquent entre-elles via le réseau ROS soit par WiFi, soit à l'interne de l'OBC via des sockets TCP-IP. Le schéma de la Figure 5.8 présente l'interaction entre les différentes nodes ROS.

[44] Où « x » représente la position désirée, « v » la vitesse désirée, « a » l'accélération désirée, « j » le jerk désiré et $x_0$ est la position initiale de la trajectoire.

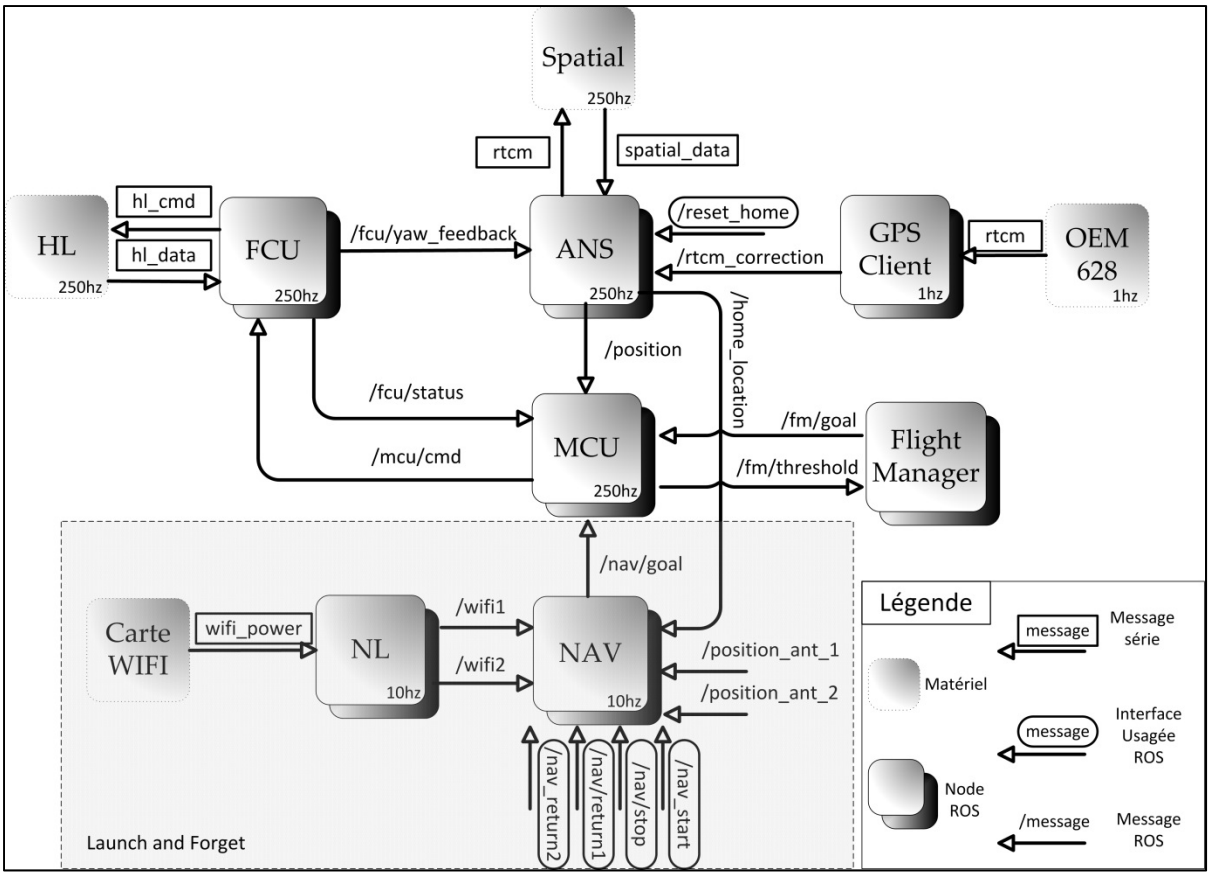



Figure 5.8 Schéma du système de vol complet

### 5.6.1 Node « Firmware Control Unit » (FCU)

La node ROS « Firmware Control Unit » (FCU) [45] sert d'interface de communication entre le réseau ROS et le processeur HL, en d'autres termes, entre ROS et le contrôleur d'attitude. La node FCU a comme tâche : la gestion du port série de l'OBC connecté au processeur HL, l'exécution de l'algorithme d'encodage/décodage de message, le traitement des données envoyées par le processeur HL et l'envoi de commandes au processeur HL. Voici la liste des messages reçus par la node FCU et transmis sur le réseau ROS.

[45] Cette node est basée sur la node ROS du même nom développée par CCNY Robotics Lab.

Tableau 5.1 Liste des messages ROS envoyés par le FCU

| Nom | Fréquence (hz) | Description |
|---|---|---|
| /fcu/imu_data | 5 | Données de la centrale inertielle (position et vitesse angulaire) |
| /fcu/status | 250 | Statut du contrôleur d'attitude (niveau de batterie, mode d'opération du contrôleur; soit manuel ou auto) |
| /fcu/rc_data | 5 | Données de la télécommande |
| /fcu/debug | 5 | Champ utilisé pour le déverminage |
| /fcu/mag | 5 | Données brutes du magnétomètre |
| /fcu/u_commands | 5 | Commande de force et moment – Sortie du contrôleur |
| /fcu/ctrl_debug_roll | 5 | Champ utilisé pour le déverminage |
| /fcu/ctrl_debug_pitch | 5 | Champ utilisé pour le déverminage |
| /fcu/ctrl_debug_yaw | 5 | Champ utilisé pour le déverminage |
| /fcu/yaw_feedback | 250 | Retour de la mesure du lacet |

Le module FCU ne reçoit qu'un seul message soit le message « /mcu/control » publié par la node MCU.

### 5.6.2 Node « Advance Navigation Spatial » (ANS)

La node « Advance Navigation Spatial » (ANS) sert d'interface de communication entre le réseau ROS et le capteur Spatial. Celle-ci comprend la gestion de deux ports séries connectés au Spatial, l'algorithme d'encodage/décodage des messages ainsi que le traitement des données reçues et envoyées. Le capteur Spatial a été configuré pour envoyer trois messages à des fréquences précises à l'OBC. Ceux-ci sont présentés dans le Tableau 5.2 :

Tableau 5.2 Messages provenant du capteur Spatial

| Nom | Fréquence (Hz) | Description |
|---|---|---|
| ans_state | 250 | Données résultantes de la fusion de donnée |
| ans_satellite | 1 | Statut de la mesure des satellites GPS |
| ans_raw_data | 1 | Données brutes des capteurs inertiels |

Nous nous concentrerons ici sur le message « ans_state » puisqu'il contient l'ensemble des données nécessaires au vol; les autres messages sont plutôt utilisés pour tester le système. Chaque réception d'un message ans_state déclenche la fonction de conversion calculant la position du quadrotor dans un repère NED local à partir de la position LLA provenant du Spatial, tel que décrit à la section 5.2.3. Une fois la conversion terminée, un nouveau message ROS « ans/position » est publié sur le réseau ROS en utilisant la dernière mesure du lacet provenant du processeur HL via le message « /fcu/yaw_feedback »[46].

Pour finir, la node ANS est inscrite au message ROS « rtcm_correction » contenant les trames de corrections DGPS provenant du receveur OEM628. Celles-ci sont acheminées directement au Spatial à l'aide du port série dédié à cet effet.

### 5.6.3 Node « Flight Manager » (FM)

La node « Flight Manager » (FM) est le gestionnaire de vol. Celui-ci permet de créer un plan de vol à partir d'une liste de points de contrôle présentée sous la forme d'un tableau. La Figure 5.9 présente l'interface graphique du « Flight Manager » lors de la création d'un plan de vol.

[46] Le message « ans/position » contient la position et la vitesse linéaire du quadrotor selon le repère local de référence NED ainsi que la position et la vitesse angulaire.

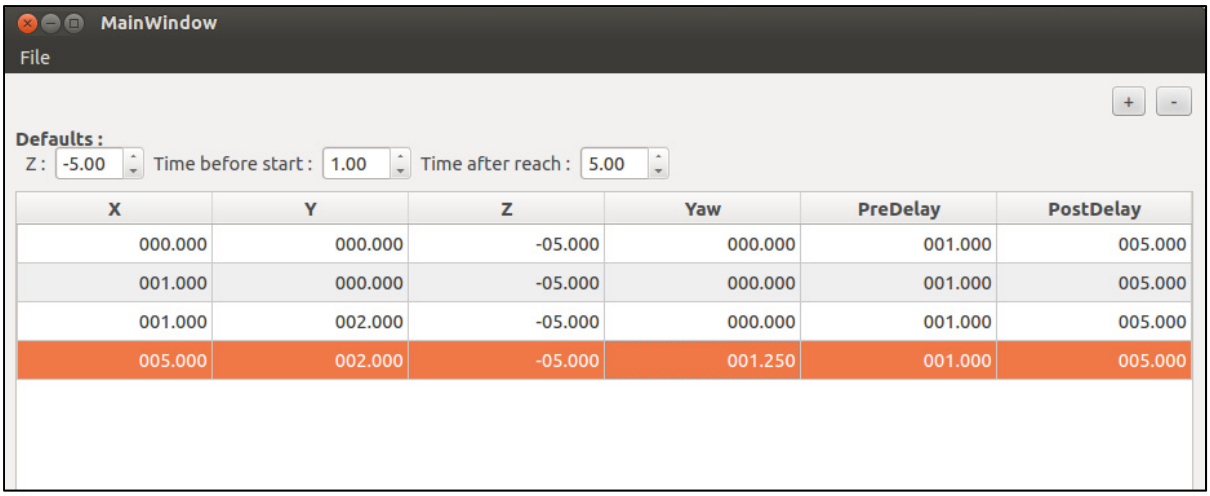



Figure 5.9 Interface du Flight Manager

Le « Flight Manager » est habituellement déployé sur un ordinateur au sol. Une fois démarré, celui-ci envoie le point de contrôle courant, via WIFI, à l'aide du message « /fm/goal » vers la node MCU lorsque le quadrotor a atteint le point de contrôle précédent[47]. Le Flight Manager détermine si le quadrotor a atteint le point de contrôle de chaque axe selon une marge d'erreur dont le seuil a été déterminé à priori. L'erreur est acheminée vers le Flight Manager via le message « /fm/threshold » provenant de la node MCU.

### 5.6.4 Node « Navigation » (NAV)

La node ROS « Navigation » (NAV) sert d'interface entre le réseau ROS et l'algorithme de navigation. Elle utilise la puissance mesurée des deux antennes provenant des messages « /wifi1 » et « /wifi2 » publiés par la node NL. L'algorithme d'autonavigation nécessite également l'angle d'arrivée des signaux pour fonctionner. Or, ceux-ci ne sont pas disponibles dans la configuration matérielle des essais expérimentaux. Pour passer outre ce problème, nous avons choisi de calculer les angles d'arrivée en utilisant la position des antennes. Celles-ci pourrait être reçue via le réseau ROS, lors de l'expérimentation.

---

[47] Sauf pour le premier point de contrôle qui est envoyé dès le démarrage de la node.

La node Nav utilise quatre messages ROS comme interface usagé. À partir de rqt_gui, l'opérateur du drone peut donc démarrer la loi de navigation (« /nav_start »), arrêter la loi de navigation (« /nav_stop ») ou retourner le drone vers l'une des deux antennes émettrices (« /nav/return1 » ou « /nav/return2 »). La loi de navigation calcule les vitesses désirées selon les axes « X » et « Y » qui sont ensuite envoyées vers la node MCU via le message « /nav/goal/ ».

### 5.6.5 Node « Mobility Control Unit » (MCU)

La node « Mobility Control Unit » (MCU) sert d'interface entre le contrôleur de position et le réseau ROS. Celle-ci gère principalement les modes de fonctionnement du contrôleur ainsi que la génération de la trajectoire désirée.

Le contrôleur de position est exécuté par le MCU à chaque réception d'un message « /ans/position » provenant de la node ANS. Puisque le Spatial envoie les données de position à une fréquence de 250Hz, le contrôleur s'exécute à la même fréquence. Pour s'assurer qu'il n'existe aucun délai lors des transitions entre le mode manuel et le mode automatique, le contrôleur de position est toujours actif, même lorsque le mode manuel est en fonction.

Comme présenté précédemment, le contrôleur assure la poursuite des trajectoires désirées. Celles-ci sont calculées par le générateur de trajectoire implémenté dans la node MCU à partir des positions finales désirées (message « /fm/goal ») provenant de la node « Flight Manager » ou à partir des vitesses finales désirées (« /nav/goal ») provenant de l'algorithme de navigation.

La node MCU implémente une machine à états, illustrée par la Figure 5.10, pour gérer le calcul de la trajectoire, et dans un même temps, le comportement du quadrotor en vol. Celui-ci est divisé en deux modes distincts, soit le vol stationnaire (« Vol Stat ») et la poursuite de trajectoire (« Poursuite Traj »). Dans le cas du vol stationnaire, la trajectoire calculée par le

MCU se base sur la position actuelle du quadrotor de telle sorte que le quadrotor demeure sur place.

Lors du démarrage du MCU, celui-ci calcule, par défaut, une trajectoire de vol stationnaire. Le pilote peut, à tout moment, choisir de passer en mode automatique. Un signal est alors lancé par le processeur HL via le message « fcu/status », transmis sur le réseau ROS par la node FCU et reçu par la node MCU. Lors du passage du mode manuel au mode automatique du contrôleur, le MCU tombe dans le mode vol stationnaire et calcule une nouvelle trajectoire.

Une poursuite de trajectoire est déclenchée si la node MCU reçoit une position finale désirée provenant de la node Flight Manager via le message « fm/goal » ou si la node MCU reçoit une vitesse finale désirée provenant de la node NAV via le message « nav/goal ». Le calcul de la trajectoire est transparent au système quelque soit le type de message de but reçu[48].

À chaque réception d'un message « ans/position » provenant de la node ANS, le MCU exécute le générateur de trajectoire pour obtenir une nouvelle position désirée et effectue le calcul du contrôleur de position. Le résultat est ensuite envoyé vers le processeur HL, via le message « /mcu/cmd ».

[48] Bien que la génération de trajectoire utilise le même mécanisme, la node « Flight Manager » et la node « Navigation » n'ont pas été conçues pour être exécutées simultanément.

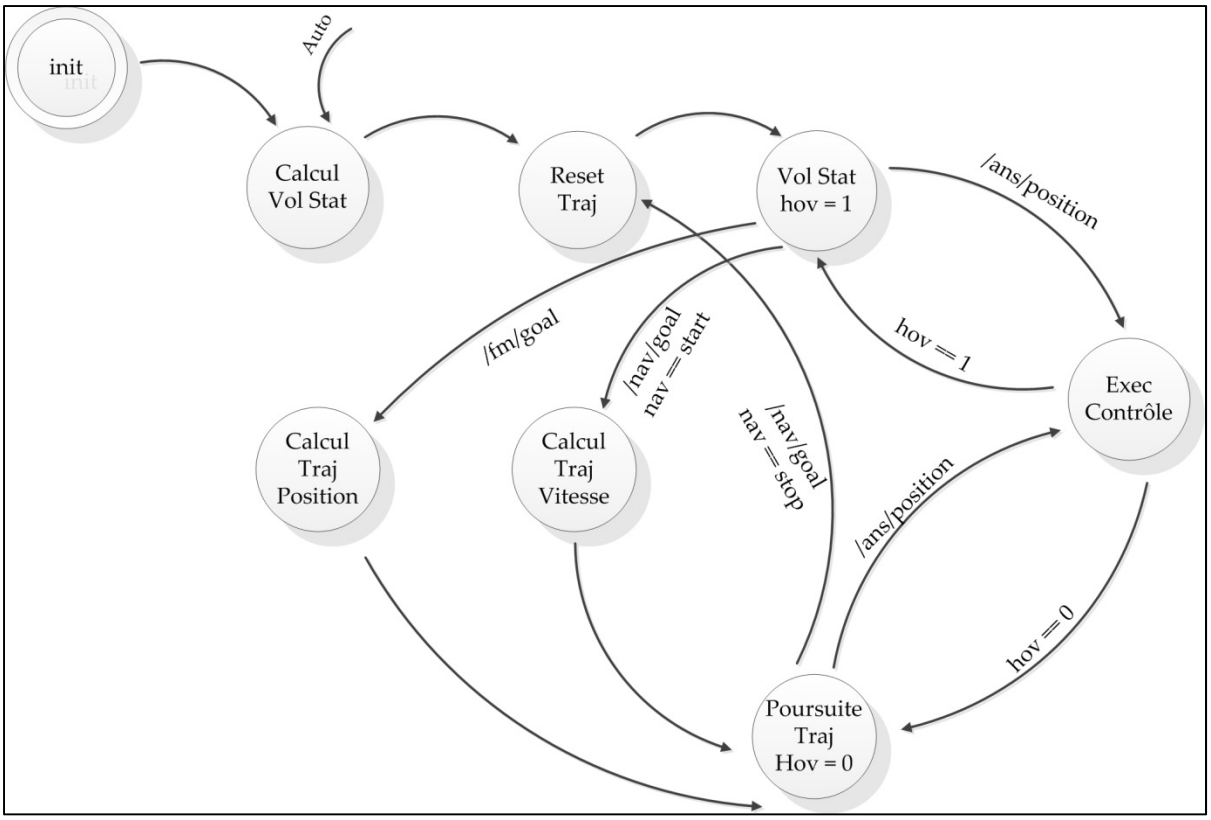



Figure 5.10 Machine à états du MCU

### 5.6.6 Node « Network Lib »

La node « Network Lib » (NL) se charge de communiquer avec la carte WiFi embarquée sur le drone pour mesurer la puissance du signal reçu par les deux antennes. L'adresse MAC de chaque antenne est utilisée pour les identifier au niveau de la node NL.

### 5.6.7 Node « Client GPS »

La node « Client GPS » sert d'interface entre le receveur GPS OEM628 et le réseau ROS. Celle-ci permet de publier les corrections RTCM qui sont acheminées vers le capteur Spatial.

## 5.7 Résumé de l'implémentation

Nous avons abordé dans ce chapitre toutes les décisions techniques relatives à l'implémentation du contrôleur backstepping. Tout d'abord, il a été constaté que la précision du système de navigation inertielle fusionné avec les données GPS fournies avec le Pelican s'avérait insuffisante pour implémenter le backstepping. Nous avons choisi de remplacer le système fourni par Ascending Technologies par le capteur INS/GPS Spatial qui est plus précis et offre la possibilité d'utiliser les signaux de correction RTCM. Cependant, le choix de ce capteur a influencé l'ensemble de nos décisions architecturales par la suite. En effet, pour obtenir le débit nécessaire de communication, nous avons choisi de séparer le contrôleur en deux; le contrôleur d'attitude est exécuté au niveau du processeur HL tandis que le contrôleur de position est exécuté au niveau de l'OBC. L'ensemble du système de vol a été développé en utilisant ROS comme cadriciel.

# CHAPITRE 6

# VALIDATION ET EXPÉRIMENTATION

Ce chapitre se consacre à la présentation des résultats d'expérimentation du contrôleur backstepping présenté au CHAPITRE 4. Deux méthodes de validation ont été effectuées : la première consiste à effectuer des simulations de la dynamique du quadrotor jumelée au contrôleur backstepping sous Matlab/Simulink tandis que la deuxième consiste à effectuer des essais expérimentaux à l'aide du Asctec Pelican modifié, tel que présenté au chapitre CHAPITRE 5. Pour les deux approches, nous présenterons la méthodologie utilisée pour obtenir les résultats ainsi que les scénarios de test exécutés.

## 6.1 Simulation

Le modèle de simulation du quadrotor a été implémenté sous Matlab/Simulink. La dynamique du quadrotor, représentée par l'équation (2.33), est intégrée à Simulink à l'aide de S-Function tandis que le contrôleur, le planificateur de mission, ainsi que le générateur de trajectoire sont intégrés à l'aide de « Matlab-Function ». La simulation inclut également la dynamique des moteurs modélisés à l'aide d'une fonction de transfert d'ordre 1. Les capteurs sont modélisés comme étant parfaits. La Figure 6.1 présente l'architecture du modèle de simulation.

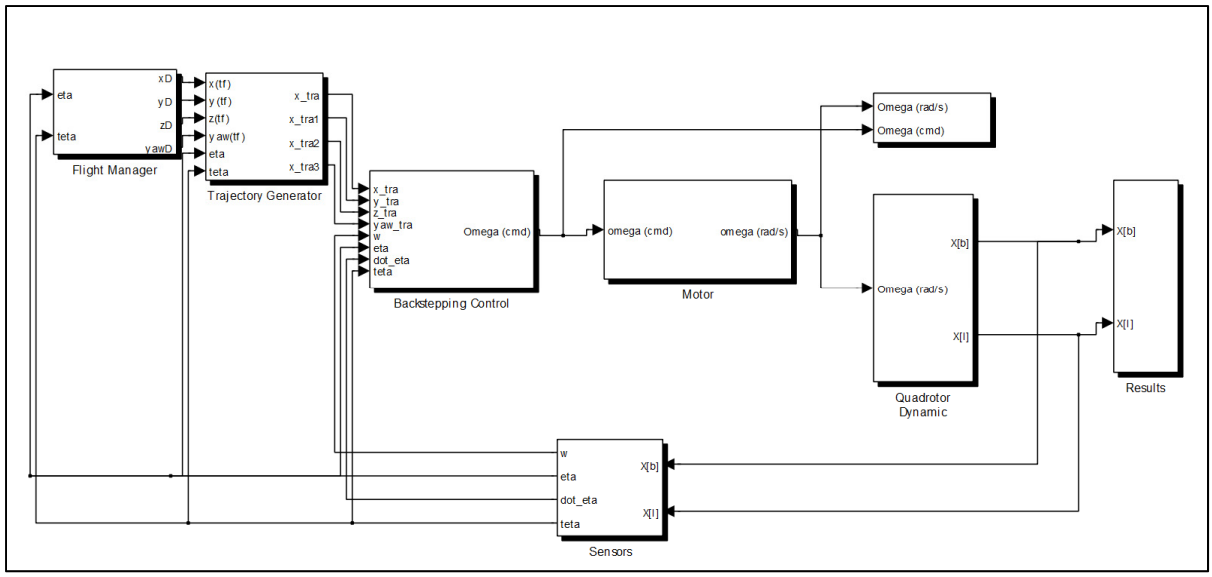



Figure 6.1 Schéma du modèle de simulation

Nous avons choisi de présenter ici deux scénarios de simulation. Le premier scénario consiste à faire une simulation dite « parfaite » en modélisant la dynamique des moteurs comme étant infiniment rapide[49] et en assumant que les paramètres du drone sont parfaitement connus par le contrôleur. Ce scénario nous permet de démontrer les performances théoriques atteignables par le contrôleur. Le deuxième scénario présente une utilisation du contrôleur dans un cas applicatif, soit avec la loi d'autonavigation. Dans ce scénario la dynamique des moteurs est considérée. Ceci nous permet de démontrer l'efficacité du contrôleur dans un cadre d'application réelle.

### 6.1.1 Paramètres de simulation

Cette section présente les différents paramètres utilisés lors des simulations. Le Tableau 6.1 spécifie les paramètres de Matlab/Simulink utilisés pour les simulations.

[49] La dynamique des moteurs est alors représentée par un gain unitaire.

Tableau 6.1 Paramètres de Matlab/Simulink

| Paramètre Matlab/Simulink | Valeur |
|---|---|
| Solveur | Ode4 (Runger-Kutta) |
| Pas de calcul | 0.005s |

Les paramètres du quadrotor utilisés correspondent à ceux du Asctec Pelican tel que présenté par (Wang, Raffler *et al.*, 2012). L'inertie de l'hélice a, quant à elle, été mesurée à l'aide d'un modèle CAO produit par le logiciel Catia 5. Le Tableau 6.2 présente ces paramètres.

Tableau 6.2 Paramètre du quadrotor Asctec Pelican[50]

| Nom | Symbole | Valeur |
|---|---|---|
| Masse | $m$ | $1.36\ kg$ |
| Inertie autour de l'axe $\boldsymbol{i_b}$ | $I_{xx}$ | $0.024\ kg \cdot m^2$ |
| Inertie autour de l'axe $\boldsymbol{j_b}$ | $I_{yy}$ | $0.024\ kg \cdot m^2$ |
| Inertie autour de l'axe $\boldsymbol{k_b}$ | $I_{zz}$ | $0.032\ kg \cdot m^2$ |
| Coefficient de poussée | $b$ | $0.00001912444476702\ N \cdot s^2$ |
| Coefficient de trainé | $d$ | $0.00000184\ N \cdot m \cdot s^2$ |
| Longueur des bras | $l$ | $0.21\ m$ |
| Inertie de l'hélice | $I_m$ | $0.000078187598\ kg \cdot m^2$ |

Les gains du contrôleur ainsi que le gain des estimateurs de dérivée sont présentés dans le Tableau 6.3. Ceux-ci correspondent également aux gains utilisés lors des essais expérimentaux.

[50] Comme indiqué précédemment, les paramètres b et d ne sont pas nécessaires lors de l'implémentation pratique du contrôleur, cependant, ceux-ci sont utilisés en simulation pour calculer la vitesse des moteurs.

Tableau 6.3 Gains du contrôleur backstepping[51]

| Gain | Valeur |
|---|---|
| $\Lambda_1$ : Gain position | $diag(2 \quad 2 \quad 2)$ |
| $\Lambda_2$ : Gain vitesse | $diag(1 \quad 1 \quad 0.35)$ |
| $\Lambda_3$ : Gain position angulaire | $diag(7 \quad 7 \quad 2)$ |
| $\Lambda_4$ : Gain vitesse angulaire | $diag(5 \quad 5 \quad 1.8)$ |
| $\lambda$ | 5 |
| $\beta$ | 70 |

### 6.1.2 Scénario 1 : Modèle parfait

Le premier scénario de simulation présente la poursuite d'une trajectoire très agressive par le quadrotor. Pour obtenir le profil de trajectoire désiré, nous avons choisi d'utiliser une trajectoire polynomiale d'ordre 8. Ceci nous permet de contrôler les positions, vitesses accélérations et jerks initiaux et finaux[52]. Le temps choisi pour effectuer la trajectoire et la distance à parcourir agissent sur l'agressivité de la trajectoire. Dans cette simulation, nous avons choisi d'utiliser trois secondes par segments de trajectoire.

Le Tableau 6.4 présente le plan de vol de ce scénario.

[51] Où $diag(\cdot)$ est une matrice diagonale constituée du vecteur des paramètres.
[52] Les vitesses, accélérations et jerks initiaux et finaux sont choisis comme étant nuls.

Tableau 6.4 Scénario de vol

| Point | X (m) | Y (m) | Z (m) | Lacet (rad) |
|---|---|---|---|---|
| 1 | 0 | 0 | 0 | 0 |
| 2 | 16 | 0 | $-5$ | $3\pi/4$ |
| 3 | 16 | $-16$ | $-2$ | $3\pi/2$ |
| 4 | 0 | $-16$ | $-5$ | $\pi/4$ |
| 5 | 0 | 0 | 0 | 0 |

Le Tableau 6.5 présente les conditions initiales du quadrotor. Les vitesses ainsi que les accélérations sont considérées comme étant nulles pour tous les axes.

Tableau 6.5 Condition initiale du quadrotor dans la simulation

| État | Valeur |
|---|---|
| $x(0)$ | $0.5\ m$ |
| $y(0)$ | $0.5\ m$ |
| $z(0)$ | $-0.5\ m$ |
| $\phi(0)$ | $0\ rad$ |
| $\theta(0)$ | $0\ rad$ |
| $\psi(0)$ | $0\ rad$ |

La Figure 6.2 présente la poursuite de la trajectoire effectuée par le quadrotor en trois dimensions. Pour plus de clarté, l'axe z est inversé pour être positif. Malgré une erreur initiale de 0.5m selon les trois axes principaux, le quadrotor converge rapidement et poursuit efficacement la trajectoire pendant l'ensemble de la mission.

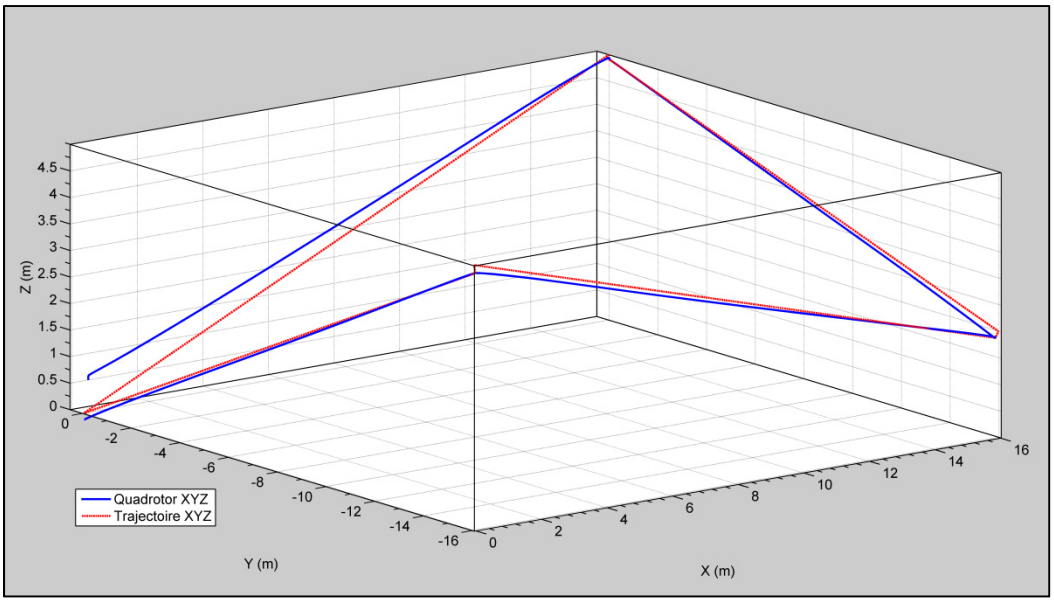



Figure 6.2 Poursuite de la trajectoire tridimensionnelle par le quadrotor

La Figure 6.3 présente le résultat de la simulation selon chacun des degrés de liberté contrôlé par le backstepping soit la position tridimensionnelle ainsi que le lacet du quadrotor. Puisque le contrôleur utilise directement les mêmes paramètres que le modèle du quadrotor, nous obtenons le résultat attendu, soit une poursuite parfaite de la trajectoire.

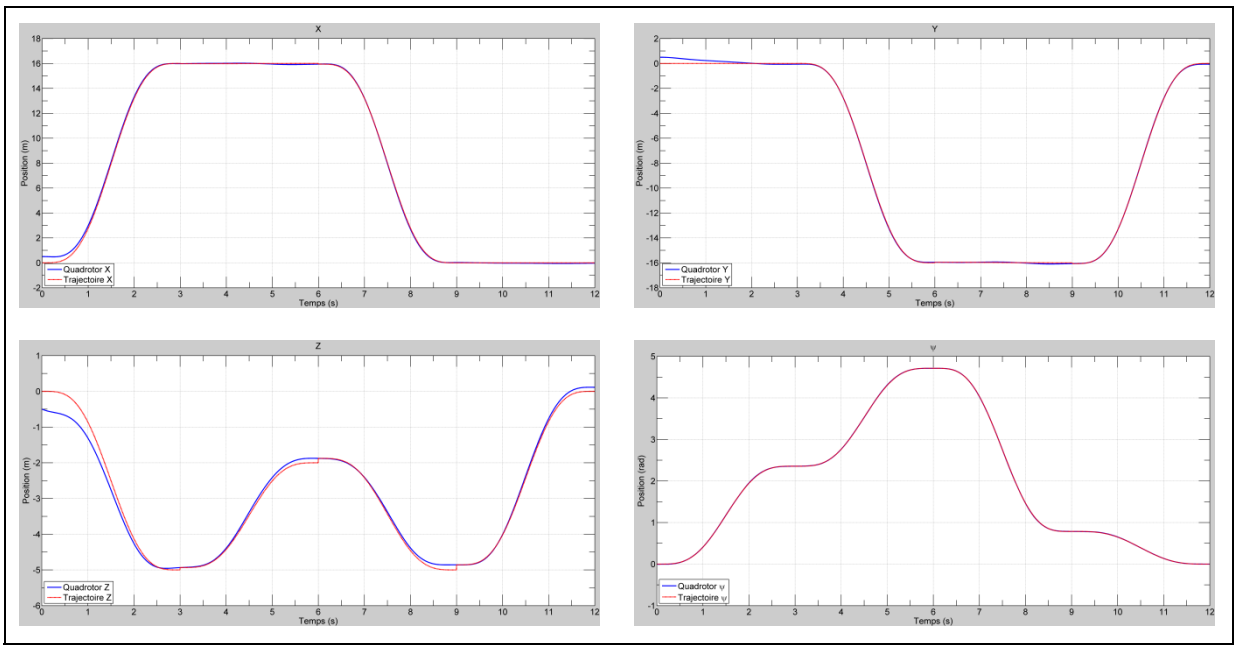



Figure 6.3 Poursuite de trajectoires par le quadrotor selon $x$, $y$, $z$ et $\psi$

La Figure 6.4 présente le résultat de la poursuite pour les degrés de liberté sous-actionnés du quadrotor, soit le roulis et le tangage par rapport aux commandes virtuelles générées par le contrôleur de position.

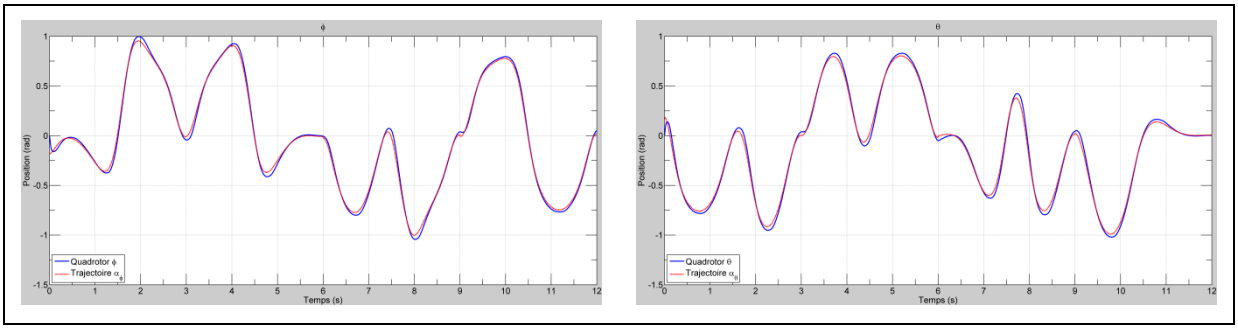



Figure 6.4 Poursuite de trajectoires par le quadrotor selon $\phi$ et $\theta$

La Figure 6.5 présente le résultat de la poursuite des trajectoires pour les quatre degrés de liberté asservis par le contrôleur backstepping.

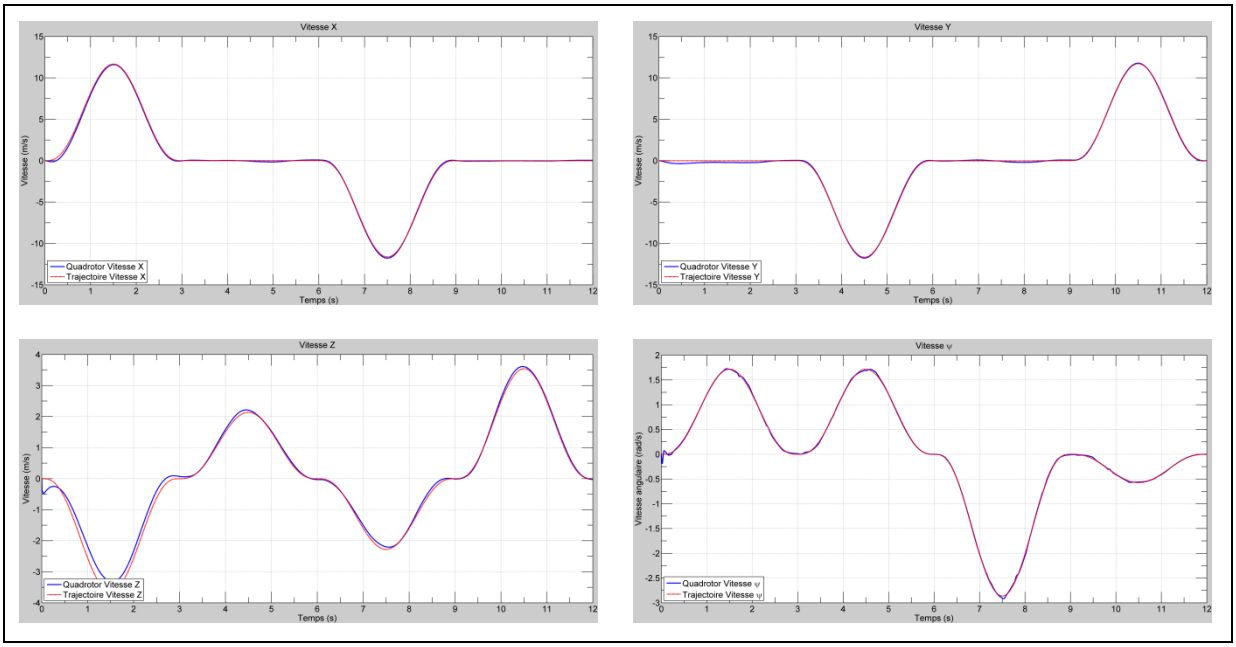



Figure 6.5 Poursuite de trajectoires par le quadrotor selon $\dot{x}$, $\dot{y}$, $\dot{z}$ et $\dot{\psi}$

La Figure 6.6 présente le résultat de la poursuite de trajectoire pour la vitesse de variation des angles d'attitude et la dérivée estimée des contrôleurs virtuels.

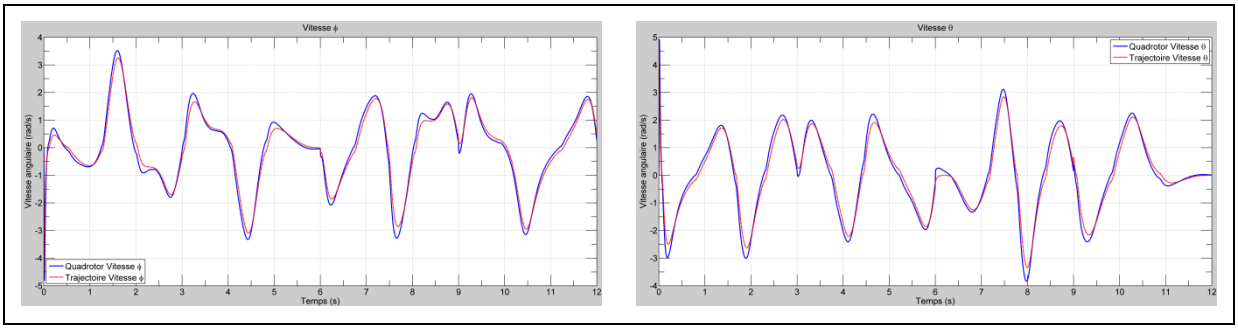



Figure 6.6 Poursuite de trajectoires par le quadrotor selon $\dot{\phi}$ et $\dot{\theta}$

La Figure 6.7 présente une comparaison entre les estimateurs de dérivée par mode glissant d'ordre 2 et la dérivée numérique de Matlab/Simulink. Tel que décrit par (Levant, 1998), l'estimateur est robuste par rapport au bruit. Comme nous pouvons le constater, l'estimateur génère de meilleurs résultats que la dérivée numérique de Matlab, car il élimine les discontinuités crées par des variations importantes du signal.

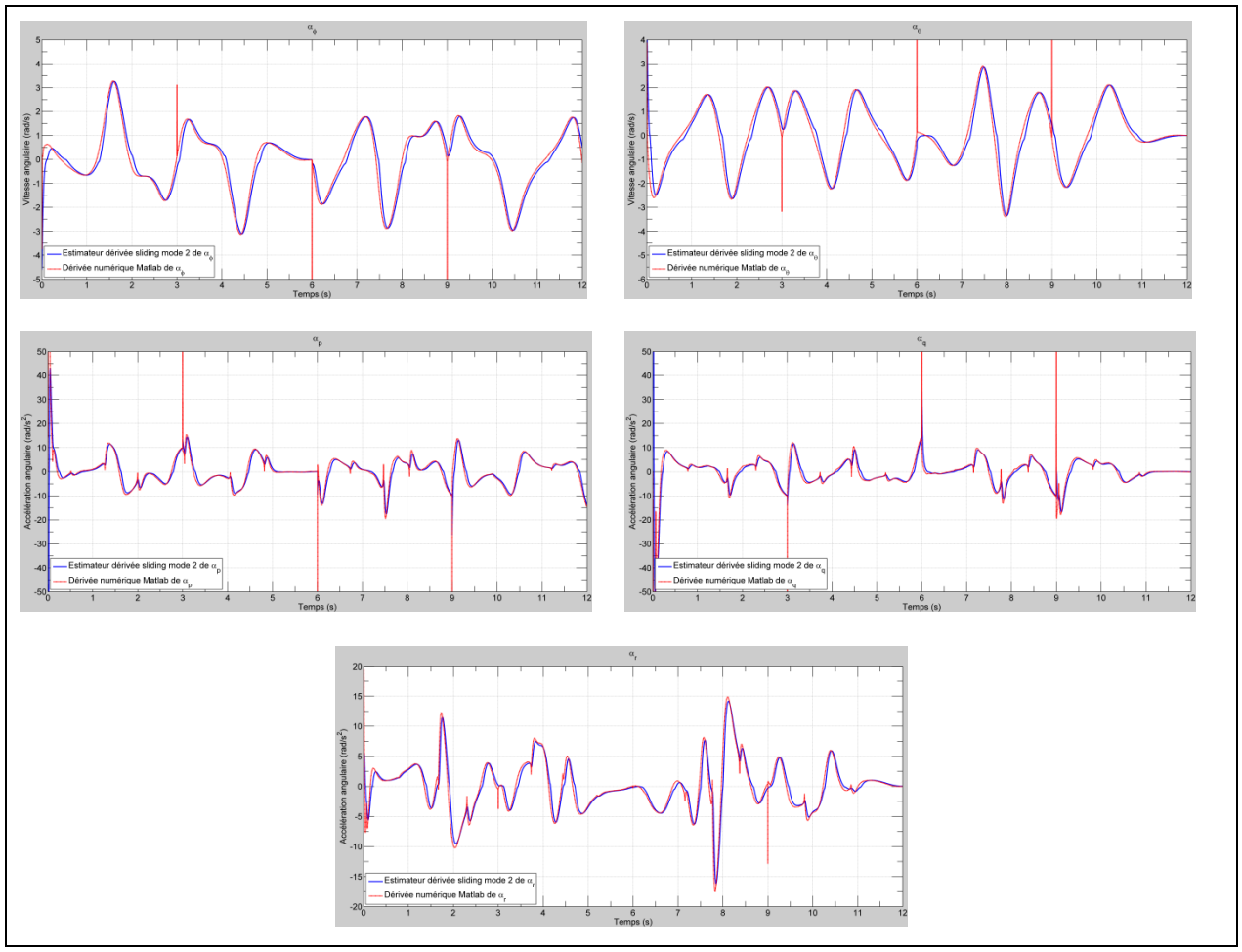



Figure 6.7 Comparaison entre l'estimateur de dérivée par mode glissant d'ordre 2 et la dérivée numérique Matlab/Simulink pour les commandes virtuelles

La Figure 6.8 présente les efforts demandés au niveau du groupe de propulsion du quadrotor. Puisque la vitesse maximale des moteurs est d'environ 850 $rad/s$, nous pouvons constater que le contrôleur génère des entrées de commande atteignables en pratique.

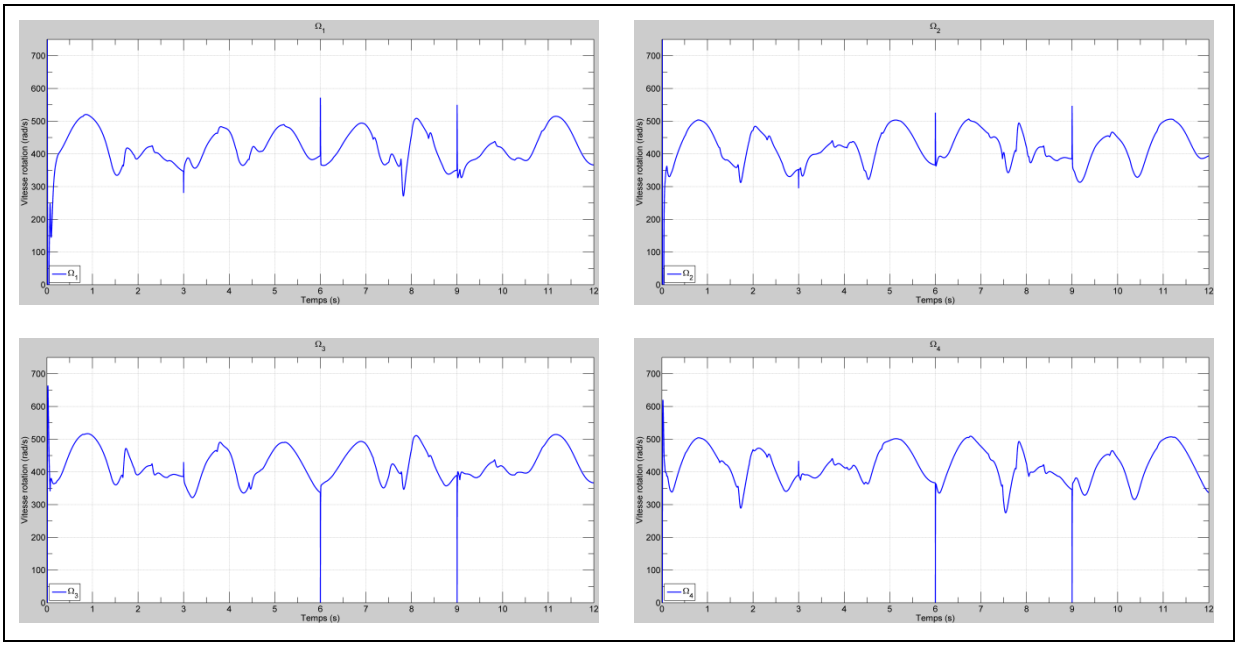



Figure 6.8 Vitesse des moteurs

À l'aide des figures précédentes, nous avons démontré l'efficacité du contrôleur backstepping à poursuivre une trajectoire agressive. En effet, le profil de la trajectoire choisie pousse le quadrotor à des vitesses de pointe de $12\ m/s$ et à des accélérations maximales de $13\ m/s^2$. Il est intéressant de noter que plusieurs discontinuités sont présentes dans les résultats. Celles-ci correspondent au point de jonction entre deux trajectoires soit, lors de la 3$^{e}$ 6$^{e}$ et 9$^{e}$ seconde: celles-ci proviennent d'une instabilité numérique en ces points. En pratique, ce problème pourrait être réglé en assurant un meilleur lien de continuité entre les différentes parties de la trajectoire.

### 6.1.3 Scénario 2 : Autonavigation

Ce deuxième scénario de simulation est utilisé pour démontrer l'efficacité du contrôleur à l'aide d'une application réelle. Nous avons choisi d'utiliser la loi d'autonavigation développée durant le projet « Launch and Forget ». Comme indiqué précédemment, celle-ci permet au quadrotor effectuant un relais de communication, de se diriger vers le point optimal, au sens des télécommunications, pour améliorer le service entre deux antennes. Il est

aussi important de noter que la loi de navigation génère des vitesses désirées seulement pour les axes x et y. Dans un cas d'application typique, l'altitude ainsi que le lacet du quadrotor sont maintenus à leur position initiale durant l'ensemble du vol.

Pour cette simulation, nous avons choisi d'ajouter la dynamique des moteurs. Celle-ci est modélisée à l'aide d'une fonction de transfert d'ordre 1 suivie d'une saturation dont les paramètres sont présentés au le Tableau 6.6. Il est important de noter que ces valeurs n'ont pas été obtenues expérimentalement, mais ont été choisies dans l'optique d'être le pire cas[53].

Tableau 6.6 Paramètres des moteurs en simulation

| **Paramètres** | **Valeurs** |
|---|---|
| $K$ : Gain | 1 |
| $\tau$ : constante de temps | $0.2\ s$ |
| Saturation | $850\ rad/s$ |

Les trajectoires choisies pour ce scénario de simulation correspondent à celles utilisées en pratique, telles que présentées à la section 5.5.2. Le Tableau 6.7 présente les paramètres des trajectoires.

Tableau 6.7 Paramètres des trajectoires

| **Paramètres** | **Valeurs** |
|---|---|
| Accélération maximale | $3\ m/s^2$ |
| Jerk maximal | $3\ m/s^3$ |

Les conditions initiales du quadrotor pour ce scénario sont présentées au Tableau 6.8.

[53] En effet, les moteurs sans balai utilisés conçus pour des appareils multirotors sont typiquement très rapides, de telle sorte que la dynamique de ceux-ci est négligée par plusieurs auteurs tels que (Bouabdallah, 2007)

Tableau 6.8 Conditions initiales du quadrotor dans la simulation

| **État** | **Valeur** |
|---|---|
| $x(0)$ | $0\ m$ |
| $y(0)$ | $0\ m$ |
| $z(0)$ | $-10\ m$ |
| $\phi(0)$ | $0\ rad$ |
| $\theta(0)$ | $0\ rad$ |
| $\psi(0)$ | $0\ rad$ |

Le Tableau 6.9 présente la position des antennes.

Tableau 6.9 Position des antennes

| **Antenne** | **X (m)** | **Y (m)** |
|---|---|---|
| 1 | $30\ m$ | $35\ m$ |
| 2 | $0\ m$ | $-5\ m$ |

La Figure 6.9 présente la poursuite de la trajectoire par le quadrotor dans le plan x-y. La position des deux antennes est également présentée. Dans la simulation, nous considérons que les paramètres des deux antennes sont identiques de façon à ce que le point optimal soit positionné à équidistance des deux antennes et directement entre les deux.

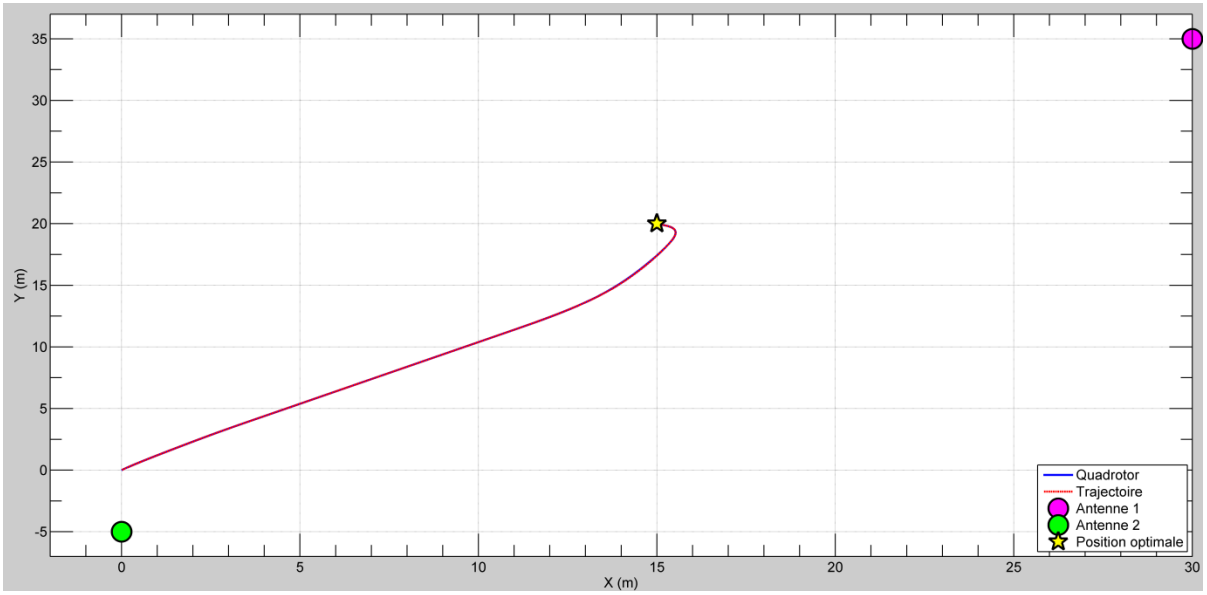



Figure 6.9 Poursuite de la trajectoire générée par la loi d'autonavigation

La Figure 6.10 présente la poursuite des trajectoires individuellement. Tel qu'on peut le constater, le contrôleur assure une poursuite parfaite de la trajectoire.

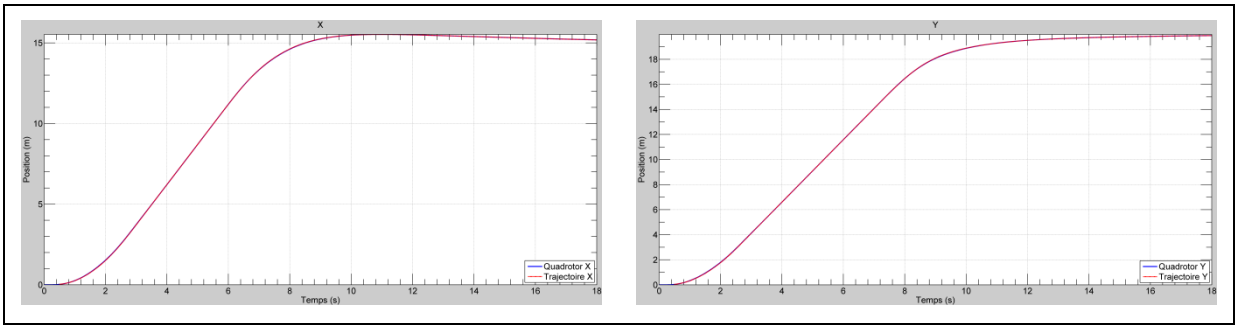



Figure 6.10 Poursuite de trajectoires par le quadrotor selon $x$ et $y$

La Figure 6.11 présente les degrés de liberté sous-actionnés du quadrotor.

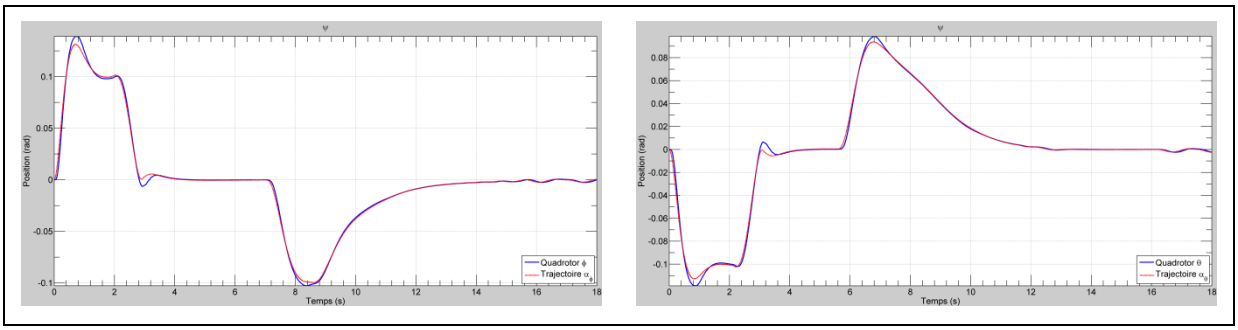



Figure 6.11 Poursuite de trajectoires par le quadrotor selon $\phi$ et $\theta$

La Figure 6.12 présente les axes qui ne sont pas contrôlés par la loi d'autonavigation. Comme on peut le constater, ceux-ci ne sont pas affectés par le déplacement et maintiennent leur valeur initiale durant le vol.

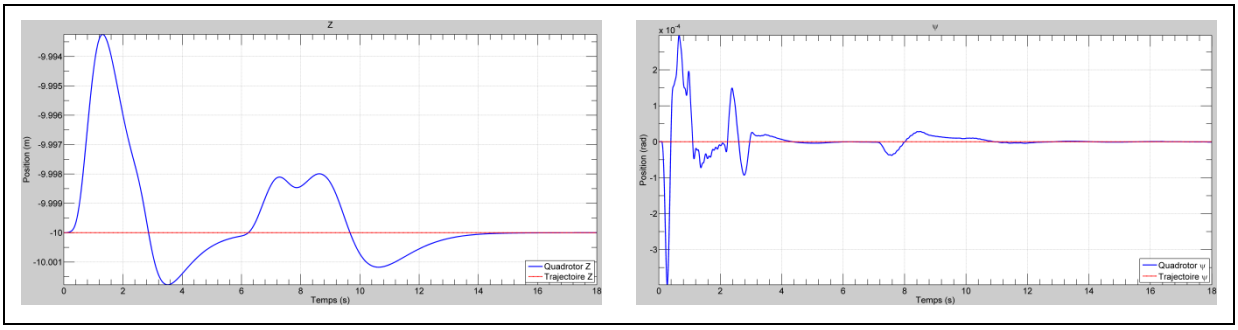



Figure 6.12 Poursuite de trajectoires par le quadrotor selon $z$ et $\psi$

La Figure 6.13 présente les profils des vitesses désirées ainsi que les vitesses linéaires du quadrotor. Il est important de noter que la trajectoire de vitesse est construite à l'aide des vitesses désirées par la loi d'autonavigation. Puisque cette trajectoire n'est pas connue à priori, nous avons choisi d'ajouter, tout comme en pratique, une saturation pour limiter la vitesse commandée par la loi de navigation. Dans notre cas, celle-ci correspond à $2.5m/s$.

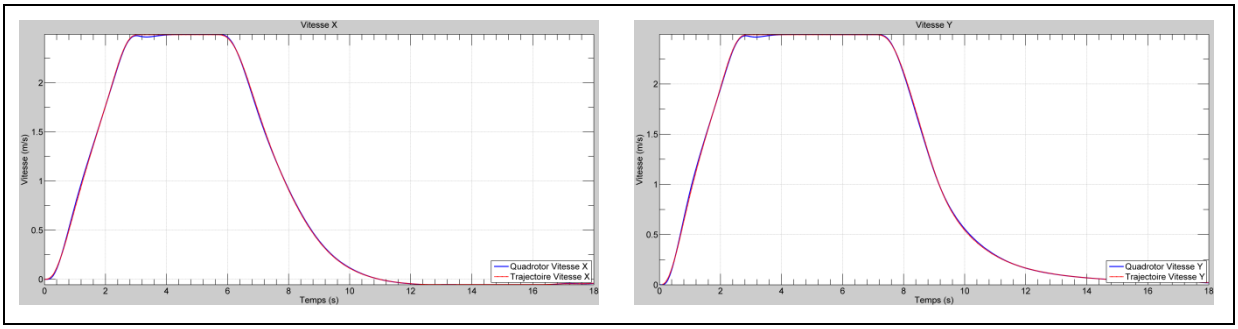



Figure 6.13 Poursuite de trajectoires par le quadrotor selon $\dot{x}$ et $\dot{y}$

La Figure 6.14 présente la variation des positions angulaires désirées ainsi que la poursuite effectuée par le contrôleur.

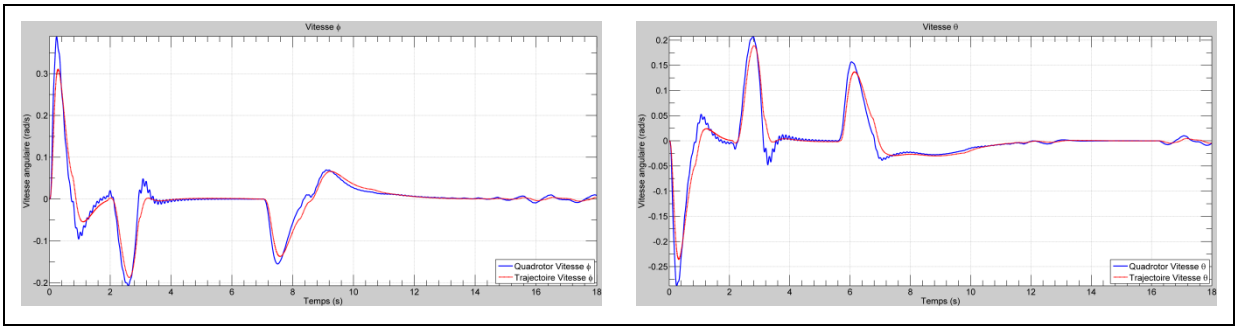
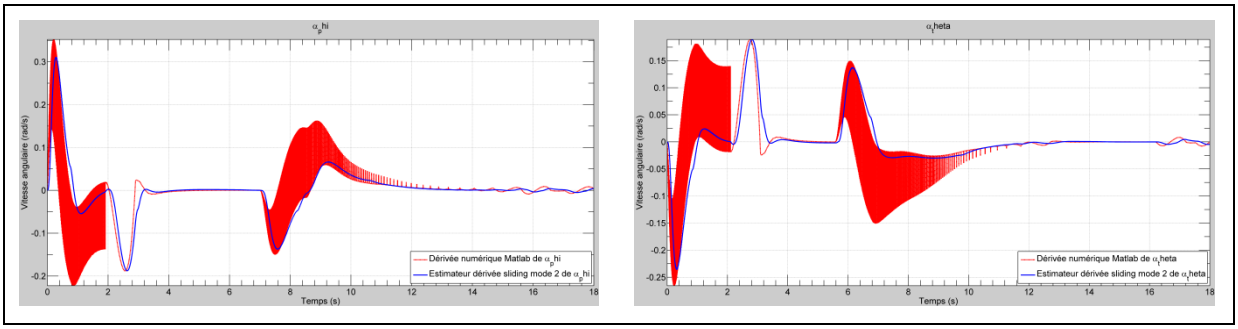



Figure 6.14 Poursuite de trajectoires par le quadrotor selon $\dot{\phi}$ et $\dot{\theta}$

La Figure 6.15 présente la comparaison entre l'estimateur de dérivée par mode glissant d'ordre 2 et la dérivée numérique de Matlab. Comme dans le cas du premier scénario de simulation, nous pouvons constater que l'estimateur rejette efficacement le bruit à haute fréquence inhérent à ce type de calcul.



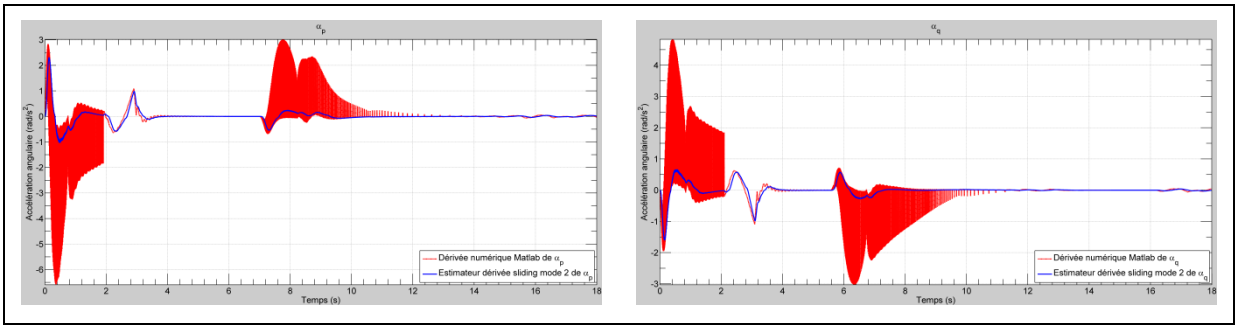



Figure 6.15 Comparaison entre l'estimateur de dérivée par mode glissant d'ordre 2 et la dérivée numérique Matlab/Simulink pour les commandes virtuelles

La Figure 6.16 présente la vitesse des moteurs. Comme on peut le constater, la dynamique des moteurs agit comme un filtre passe-bas en éliminant la composante haute fréquence des commandes générées par le contrôleur. Ceci ralentit considérablement l'application des forces et des moments désirés par le contrôleur.

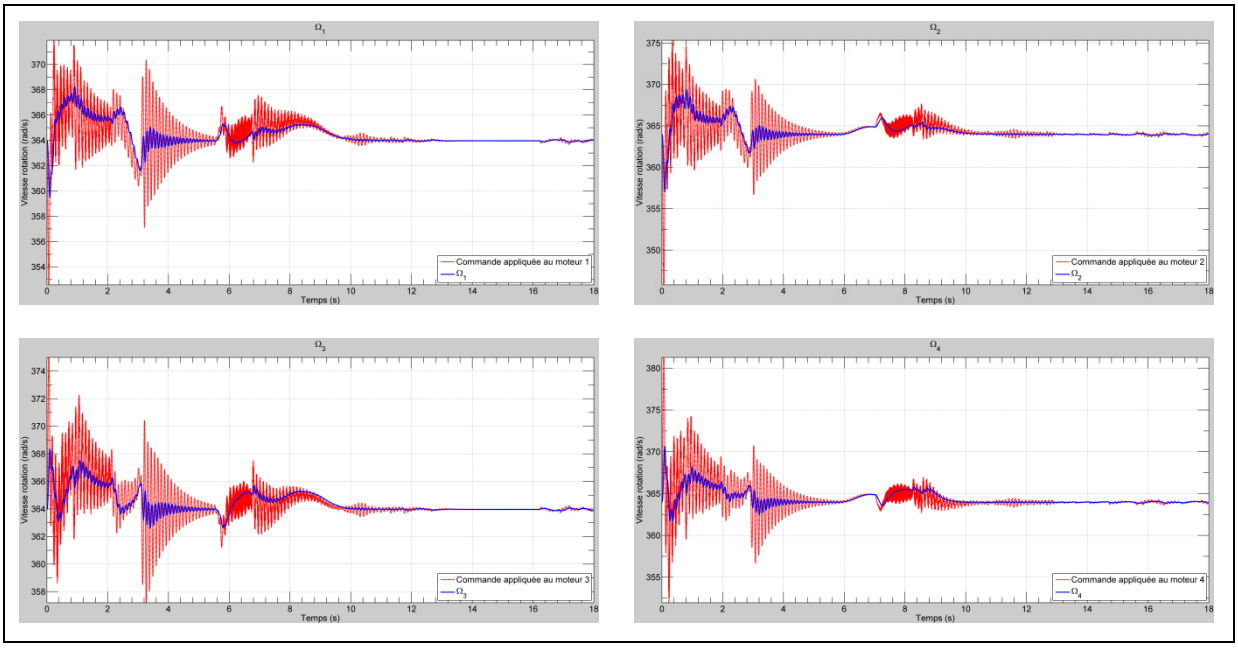



Figure 6.16 Vitesse des moteurs

Les performances obtenues dans cette simulation sont inférieures à celles présentées dans le premier scénario de simulation. En effet, dans ce deuxième cas, la vitesse de pointe atteinte est égale à $2.5m/s$ et l'accélération maximale est égale à $1.5m/s^2$. La diminution des

performances peut être expliquée par deux facteurs. Le premier facteur est l'ajout d'une dynamique de moteur peu rapide. Ceci limite la vitesse d'application des efforts de commande. Augmenter l'agressivité du profil des trajectoires aurait pour effet d'ajouter d'importantes oscillations au niveau des angles d'attitude du quadrotor. Le second facteur est l'application d'une saturation aux vitesses désirées générées par la loi d'autonavigation. Comme mentionné précédemment, cette saturation est nécessaire puisque nous n'avons aucun contrôle sur le profil de la trajectoire générée. Cependant, malgré les performances moins agressives, le contrôleur backstepping permet d'implémenter efficacement la loi d'autonavigation.

## 6.2 Expérimentation

Cette section a pour but de présenter les résultats expérimentaux. Ils ont été effectués à l'aide du quadrotor Asctec Pelican ainsi que des outils logiciels, tel que présenté au CHAPITRE 6.

Chaque vol est effectué selon la même méthodologie. Premièrement, le quadrotor est posé au sol. Sa position initiale est alors utilisée pour définir la position de l'origine du repère NED local. Une fois tous les systèmes en fonction, le pilote fait décoller le quadrotor manuellement. Une fois la position initiale atteinte, le pilote passe en mode automatique et le quadrotor effectue un vol stationnaire. La mission est ensuite lancée. Une fois la mission complétée, le pilote reprend le contrôle du quadrotor et effectue l'atterrissage de l'appareil.

Les données de vol sont mesurées à l'aide du processus ROS ros_bag. Nous pouvons donc enregistrer tous les messages publiés sur le réseau ROS. Puisque le processus d'enregistrement est lourd en terme de puissance de calcul et gourmand en espace mémoire, nous avons choisi d'effectuer l'enregistrement à partir d'un ordinateur au sol[54]. Cette configuration comporte cependant plusieurs désavantages : en effet, il a été constaté lors des tests que l'enregistrement demande une charge assez élevée sur le réseau. Puisque,

[54] Celui-ci exécute également la node « Flight Manager » et la node « Client GPS ».

l'utilisation du réseau WiFi est essentielle pour le fonctionnement du planificateur de mission et pour la loi d'autonavigation[55]; elle limite la quantité de messages pouvant être enregistrés tout en conservant la cohérence des données[56]. Nous avons donc choisi d'enregistrer le minimum possible de données tout en diminuant la fréquence d'enregistrement.

Nous avons sélectionné trois tests expérimentaux. Le premier test présente le vol du quadrotor à l'intérieur et en mode manuel. Celui-ci a pour but de démontrer l'efficacité du contrôleur d'attitude. Ce premier test est nécessaire car, comme nous l'avons présenté à la section 4.4, la performance du contrôleur de position est directement affectée par la performance du contrôleur d'attitude. De plus, le vol en mode manuel permet au pilote d'effectuer des trajectoires plus agressives par rapport aux angles d'attitude que lors du vol autonome. Dans le cadre de ce test, nous avons choisi d'enregistrer toutes les données de vol spécifiques au contrôleur d'attitude, car le réseau n'est pas nécessaire dans ce type d'application. Le deuxième test présente un vol complètement autonome du quadrotor. Celui-ci a pour but de démontrer l'efficacité de l'ensemble du contrôleur backstepping. Pour finir, un troisième test présente le fonctionnement de la loi d'autonavigation dans un cadre expérimental.

Il est important de noter que pour tous les tests expérimentaux, les valeurs des gains choisies sont les mêmes que pour la simulation et sont présentées par le Tableau 6.3. Les paramètres du quadrotor choisis pour concevoir le contrôleur sont également les mêmes qu'en simulation et sont présentées par le Tableau 6.2.

---

[55] Pour le gestionnaire de mission, voir la section 5.6.3. Pour la loi d'autonavigation, la charge sur le réseau du système ROS est encore plus importante, car celle-ci nécessite plus de communication intersystème via WiFi, principalement pour transférer la position des antennes.

[56] Puisque l'enregistrement est en temps réel, la perte de paquets WiFi engendre la perte de données. Selon la charge sur le réseau, un enregistrement peut perdre plusieurs secondes de données de manière sporadique.

### 6.2.1 Test de vol en mode manuel : Contrôleur d'attitude seulement

Le test de vol en mode manuel a été effectué à l'ÉTS à l'intérieur de la cage de vol du club Dronolab qui a une dimension de 6m X 6m X 15m. Le test s'est déroulé en effectuant des trajectoires de forme carrée très agressives[57] en suivant les bordures de la cage. Le Tableau 6.10 présente les messages ROS enregistrés durant le test ainsi que la fréquence d'acquisition.

Tableau 6.10 Message ROS et données enregistrées durant le test de vol manuel

| Message | Données | Fréquence (Hz) |
|---|---|---|
| /fcu/imu_data | $\phi \quad \theta \quad \psi \quad p \quad q \quad r$ | 250 |
| /fcu/calcul_ctrl | $\hat{\dot{\alpha}}_\phi \quad \hat{\dot{\alpha}}_\theta \quad \hat{\dot{\alpha}}_p \quad \hat{\dot{\alpha}}_q \quad \hat{\dot{\alpha}}_r \quad \dot{\phi} \quad \dot{\theta} \quad \dot{\psi}$ | 250 |
| /fcu/moteur | $\Omega_1 \quad \Omega_2 \quad \Omega_3 \quad \Omega_4$ | 250 |
| /fcu/rc | $T \quad \alpha_\phi \quad \alpha_\theta \quad \alpha_\psi$ | 250 |

La Figure 6.17 présente la poursuite du quadrotor pour les trois axes asservis par le contrôleur d'attitude, soit le roulis, le tangage et le lacet. On peut observer une très bonne poursuite au niveau du roulis et du tangage. Nous pouvons également constater que la mesure du lacet est plus bruitée ce qui explique une plus grande erreur pour cet axe.

[57] À chaque changement de direction, soit à chaque coin du carré, la trajectoire exige une variation des angles d'Euler d'environ $115°/s$. Ceci peut-être constaté sur la Figure 6.17.

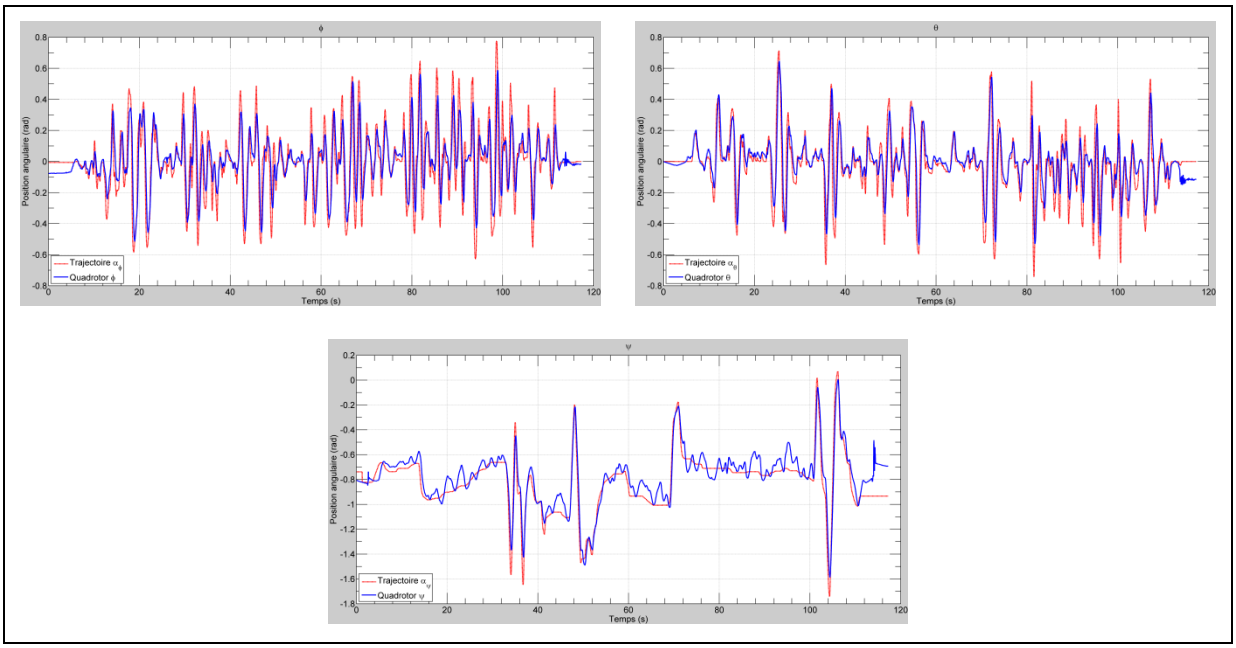



Figure 6.17 Poursuite de trajectoires par le quadrotor selon $\phi$, $\theta$ et $\psi$

La Figure 6.18 présente la variation des commandes de position angulaire et leur poursuite par le quadrotor. Nous pouvons constater une bonne poursuite de la trajectoire et que les courbes sont cohérentes avec les positions angulaires illustrées par la Figure 6.17.

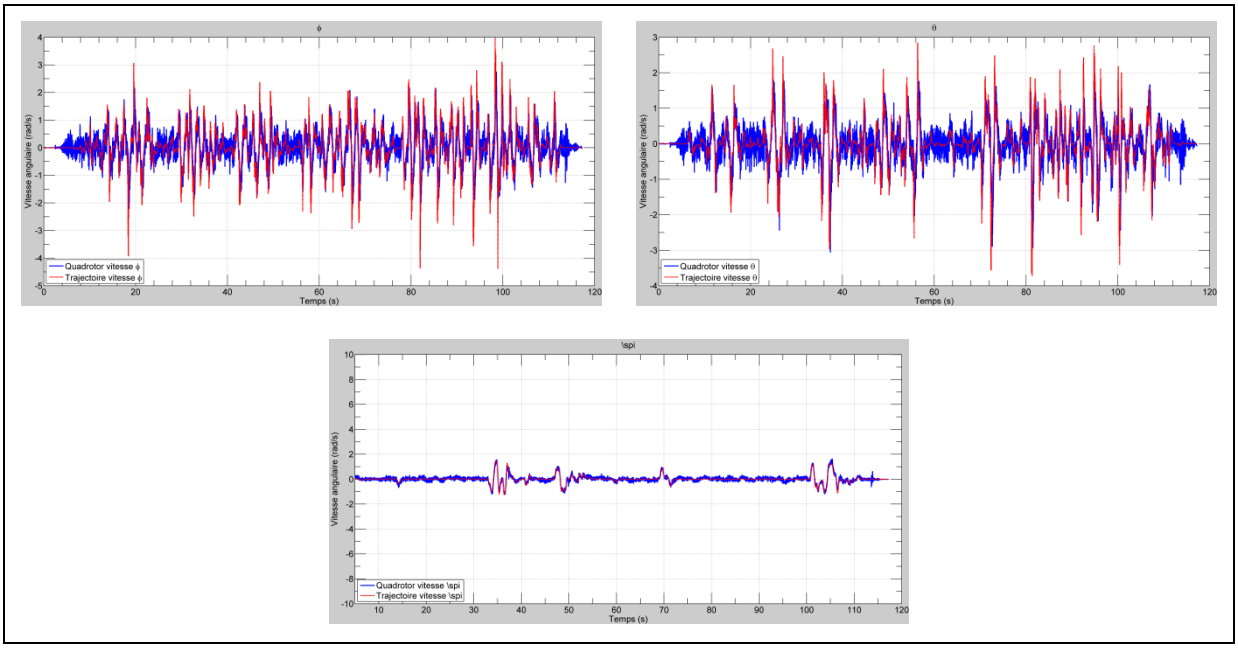



Figure 6.18 Poursuite de trajectoires par le quadrotor selon $\dot{\phi}$, $\dot{\theta}$ et $\dot{\psi}$

La Figure 6.19 présente la comparaison entre l'estimateur de dérivée par mode glissant d'ordre 2 et une dérivée numérique classique[58]. Les estimations des dérivées des signaux $\alpha_p$, $\alpha_q$ et $\alpha_r$ sont présentées seules, car le résultat des dérivées numériques est inutilisable[59].

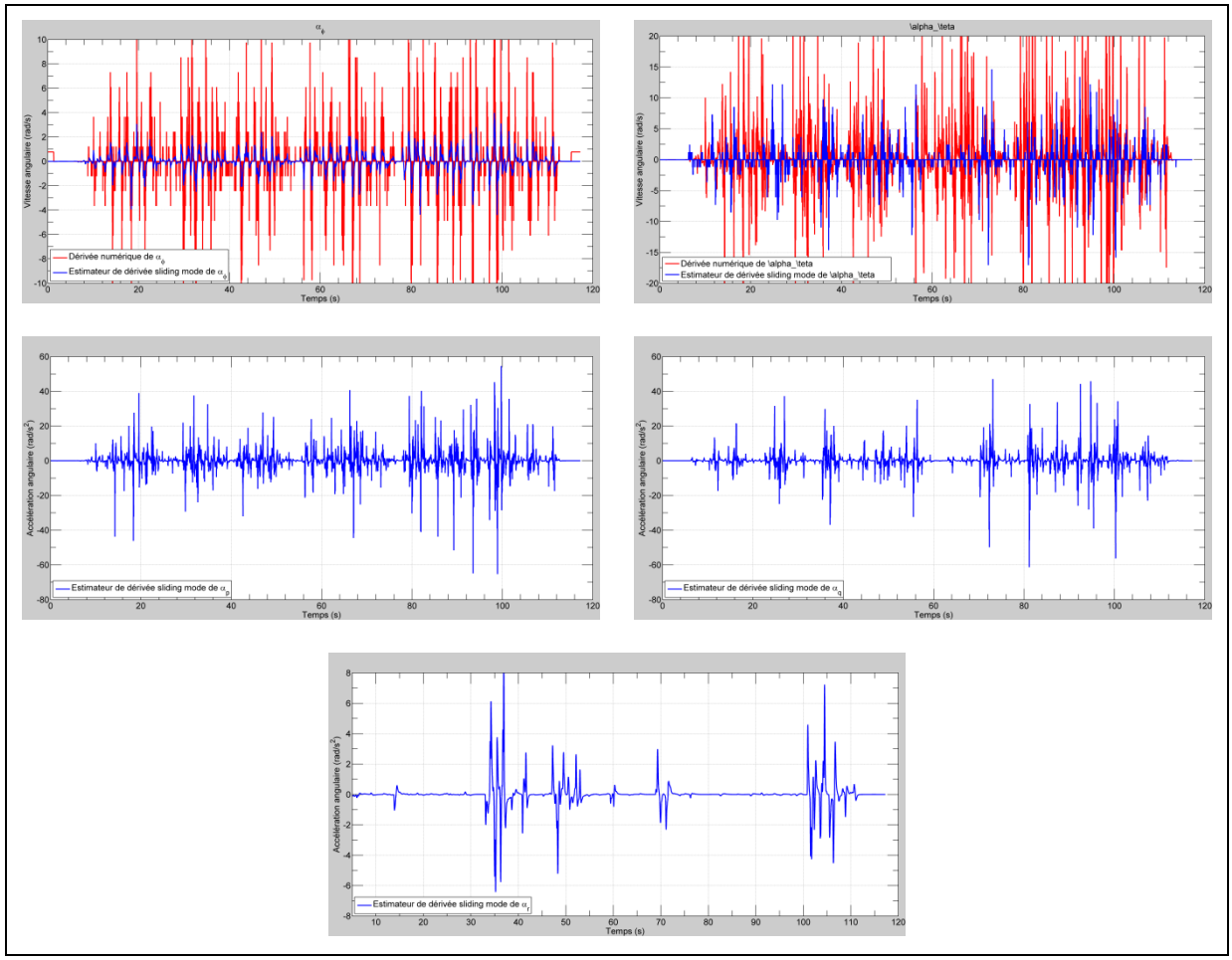



Figure 6.19 Comparaison entre l'estimateur de dérivée par mode glissant d'ordre 2 et la dérivée numérique Matlab/Simulink pour les commandes virtuelles

[58] Celle-ci est calculée tel que $\Delta x(i) = \big(x(i) - x(i-1)\big)/dt$ où $dt = 0.004$ et $i$ est le numéro de l'échantillon.
[59] Leur valeur oscille entre 10 à 100 fois par rapport à leur valeur théorique.

La Figure 6.20 présente la vitesse des moteurs mesurée. Puisque la mesure reçue est normalisée et que la valeur de conversion n'est pas fournie, nous avons effectué des mesures pour connaître la correspondance. L'ANNEXE IV présente la formule de conversion.

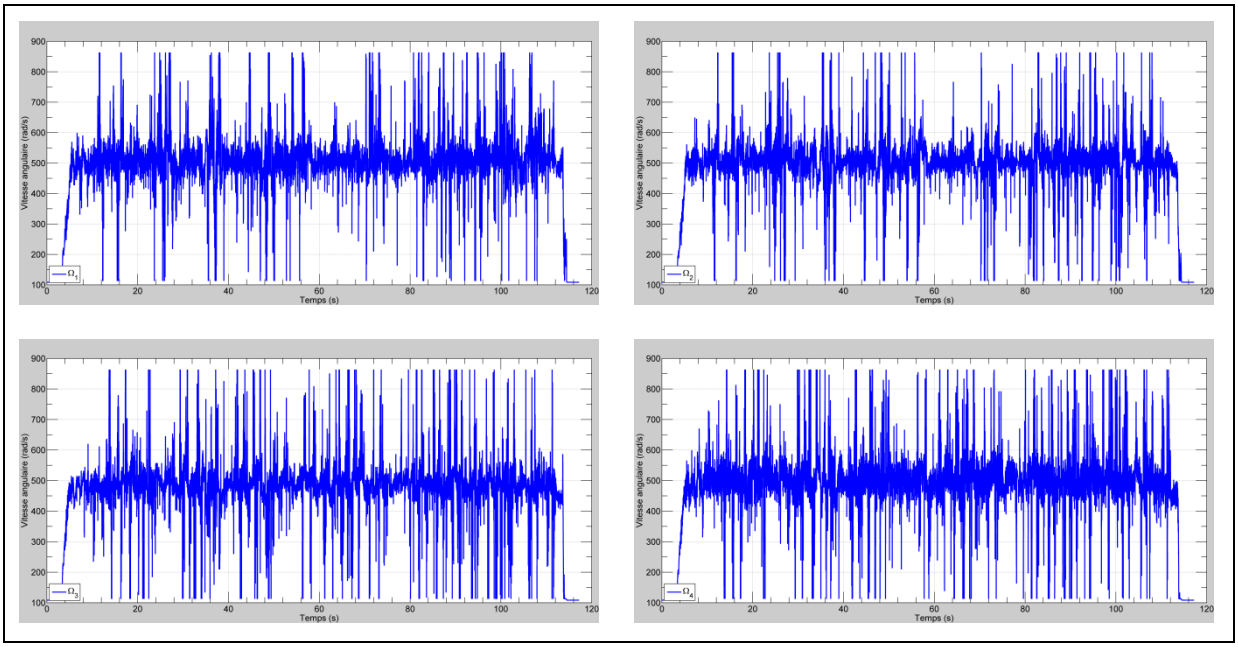



Figure 6.20 Vitesse des moteurs

Comme on peut le constater à l'aide de ces résultats, le contrôleur d'attitude peut efficacement suivre des trajectoires agressives variant d'approximativement $\pm 25°$ et pouvant atteindre des vitesses angulaires d'approximativement $\pm 115°/s$. On peut aussi observer la robustesse des estimateurs de dérivée par rapport à une dérivée numérique dans un contexte d'application pratique. Pour finir, il est intéressant de noter que la vitesse des moteurs varie de manière beaucoup plus importante qu'en simulation ce qui justifie la simplification de leur dynamique.

### 6.2.2 Test de vol en mode autonome : Contrôleur complet

Les tests de vol autonome ont eu lieu sur les terrains de l'École Nationale d'Aérotechnique (ÉNA) de Longueuil. Comparativement au vol mené à l'intérieur ou lors de simulations, les conditions climatiques, principalement le vent, peuvent avoir un impact majeur sur les

performances du système. En effet, les perturbations générées par le vent ne sont pas prises en compte par le contrôleur proposé. Après plusieurs essais, nous avons constaté que le contrôleur devient peu performant en présence de vent aillant une vitesse de plus de 15km/h. Les résultats présentés ici ont été obtenus en présence d'un vent allant entre 5 et 10km/h direction nord-est[60].

Puisque nous disposons d'un seul Asctec Pelican pour effectuer les tests, il est important de prendre le maximum de précautions pour s'assurer que le contrôleur backstepping demeure en fonction suffisamment longtemps pour permettre au pilote de rattraper l'appareil en cas de défaillance. Pour ce faire, nous avons ajouté des saturations au niveau des commandes générées par le contrôleur de position. Les commandes virtuelles des angles de roulis et de tangage ainsi que la poussée totale ont été saturées par les valeurs présentées par le Tableau 6.11. Les valeurs ont été choisies pour ne pas nuire aux performances du contrôleur tout en assurant un maximum de protection. De plus, nous avons choisi d'utiliser un profil de trajectoire peu agressive pour assurer un maximum de sécurité. Les trajectoires choisies sont celles présentées à la section 5.5.1. Les paramètres de celles-ci sont présentés par le Tableau 6.12.

Tableau 6.11 Saturations appliquées aux commandes générées par le contrôleur de position

| **Commande** | **Valeur Max** | **Valeur Min** |
|---|---|---|
| $\alpha_\phi$ | $15°$ | $-15°$ |
| $\alpha_\theta$ | $15°$ | $-15°$ |
| $T$ | $15N$ | $12N$ |

Tableau 6.12 Paramètres des trajectoires

| **Paramètres** | **Valeurs** |
|---|---|
| Vitesse maximale (x-y-z) | $1\,m/s$ |

[60] Puisque nous n'avions pas l'équipement nécessaire pour effectuer une mesure précise de la vitesse du vent de sa direction, ses informations proviennent de la tour de contrôle de l'aéroport de Longueuil.

| | |
|---|---|
| Accélération maximale (x-y-z) | $0.25\, m/s^2$ |
| Vitesse maximale ($\boldsymbol{\psi}$) | $0.17\, rad/s$ |
| Accélération maximale ($\boldsymbol{\psi}$) | $0.05\, rad/s^2$ |

Le Tableau 6.13 présente les messages ROS et les données enregistrées durant le test de vol.

Tableau 6.13 Messages ROS et les données enregistrées durant le vol autonome

| **Message** | **Données** | **Fréquence (hz)** |
|---|---|---|
| /fcu/debug_ctrl | $\lvert\phi - \alpha_\phi\rvert \quad \lvert\theta - \alpha_\theta\rvert \quad \lvert\psi - \alpha_\psi\rvert$ | 25 |
| /ans/position | $x \quad y \quad z \quad \dot{x} \quad \dot{y} \quad \dot{z}$ | 250 |
| /mcu/trajectory_X | $x_d \quad \dot{x}_d \quad \ddot{x}_d$ | 50 |
| /mcu/trajectory_Y | $y_d \quad \dot{y}_d \quad \ddot{y}_d$ | 50 |
| /mcu/trajectory_Y | $z_d \quad \dot{z}_d \quad \ddot{z}_d$ | 50 |

Le Tableau 6.14 présente le scénario de vol entré dans le planificateur de mission. Le lacet est conservé à zéro pour l'ensemble du vol pour permettre de bien voir la correspondance entre le mouvement dans le plan x-y et la variation des angles d'attitude roulis-tangage.

Tableau 6.14 Scénario de vol

| **Point** | **X (m)** | **Y (m)** | **Z (m)** | **Lacet (rad)** |
|---|---|---|---|---|
| 1 | 2 | 0 | $-3.5$ | 0 |
| 2 | 16 | 0 | $-5$ | 0 |
| 3 | 16 | $-16$ | $-3.5$ | 0 |
| 4 | 2 | $-16$ | $-3.5$ | 0 |
| 5 | 2 | 0 | $-3.5$ | 0 |

La Figure 6.21 présente en trois dimensions la poursuite de la trajectoire par le quadrotor. Pour plus de clarté, l'axe z est inversé pour être positif. L'enregistrement débute dès le passage en mode autonome. On peut voir qu'à ce moment le quadrotor perd 1m d'altitude

pour ensuite aller se stabiliser à sa position initiale. Une fois stabilisée, la mission est démarrée.

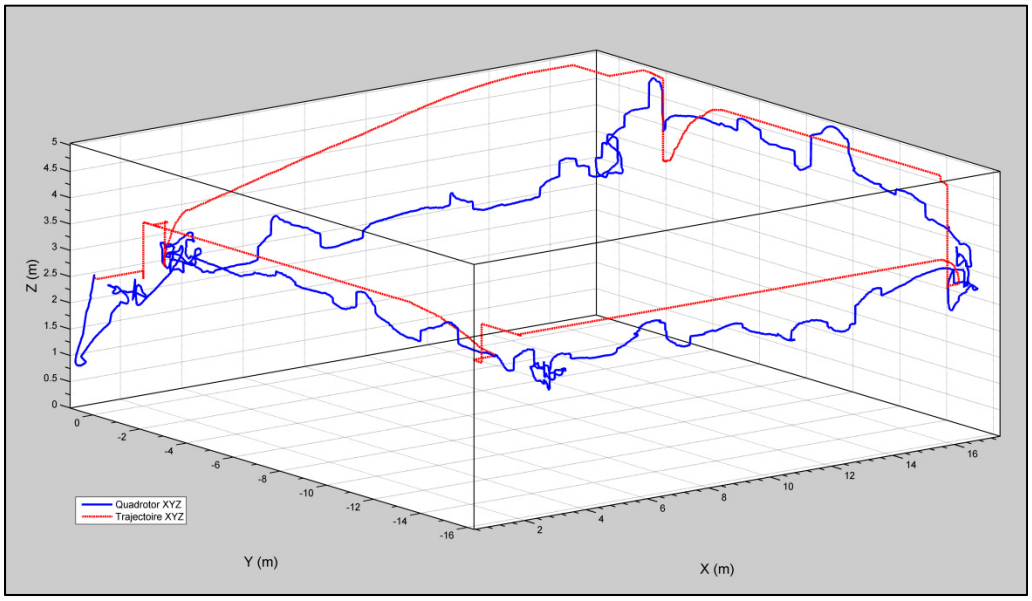
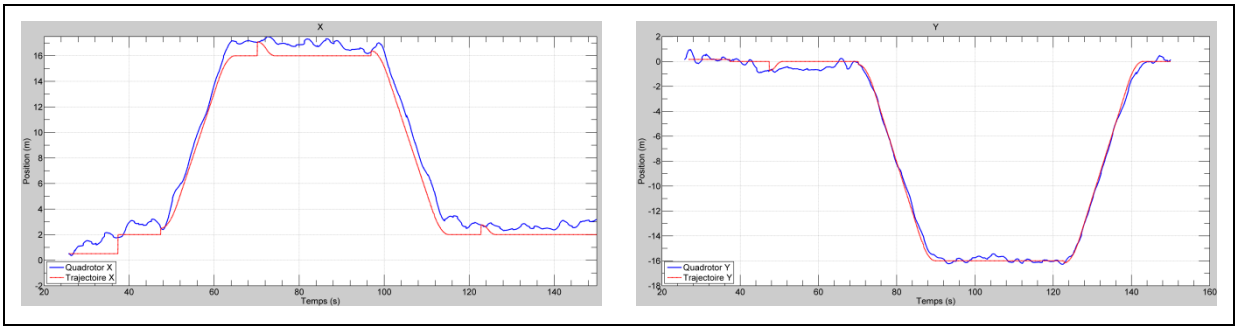



Figure 6.21 Poursuite de la trajectoire tridimensionnelle par le quadrotor

La Figure 6.22 présente les poursuites de trajectoire selon les axes x, y et z. On peut voir que les trajectoires sont recalculées pour tous les axes à chaque réception d'un nouveau point envoyé par le gestionnaire de mission.



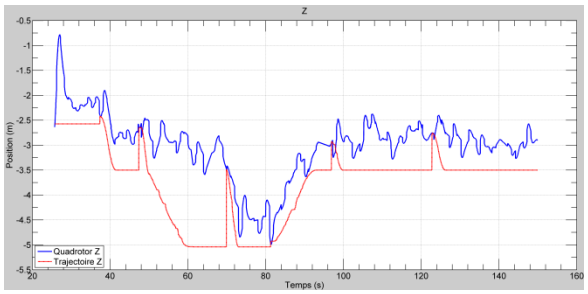



Figure 6.22 Poursuite de trajectoires par le quadrotor selon $x$, $y$ et $z$

La Figure 6.23 présente la poursuite des trajectoires des vitesses pour les axes x, y et z.

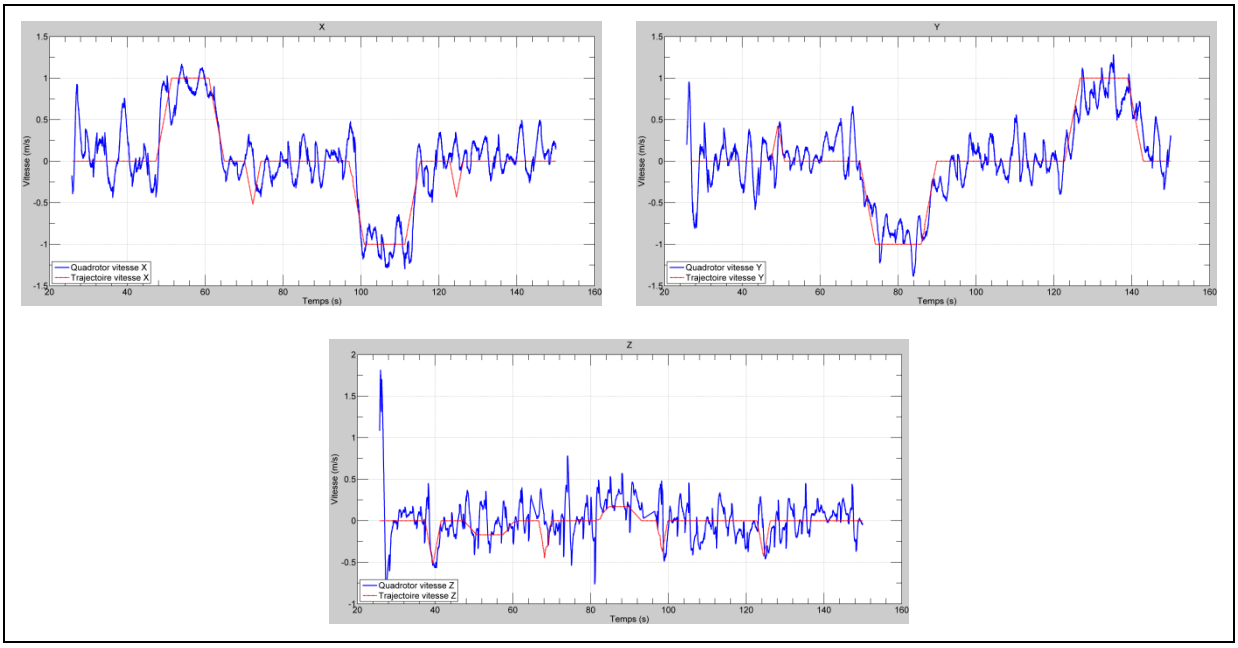



Figure 6.23 Poursuite de trajectoires par le quadrotor selon $\dot{x}$, $\dot{y}$ et $\dot{z}$

La Figure 6.24 présente les erreurs de poursuite entre les trois angles d'attitude et leur signal de référence provenant du contrôleur de position.

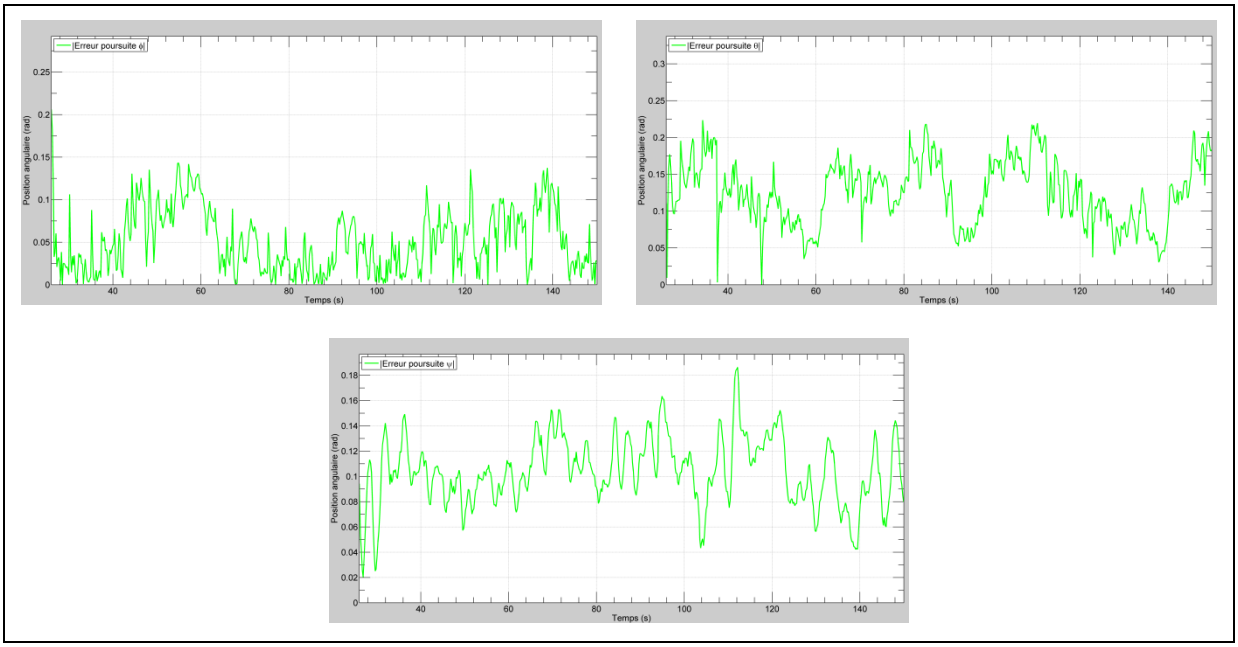



Figure 6.24 Erreur de poursuite des trajectoires par le quadrotor selon $\phi$, $\theta$ et $\psi$

Les résultats suivants démontrent une bonne poursuite de trajectoire pour les axes x et y. Nous pouvons tout de même observer une erreur d'environ 1 m selon x. L'origine de l'erreur provient, selon nous, de deux facteurs : le premier étant les perturbations générées par le vent et le deuxième, le fait que le centre de gravité du quadrotor ne coïncide pas avec le centre géométrique de l'appareil. Puisque ceci n'est pas modélisé, cette erreur affecte la performance du contrôleur de position, car la commande virtuelle qu'il calcule ne correspond pas à la position angulaire finale voulue; il existe alors une erreur en régime permanent au niveau de la position. En effet, nous pouvons observer que l'erreur de 1m selon l'axe x correspondant à une erreur de tangage d'environ $0.125\,rad$ tandis qu'une erreur pratiquement nulle selon l'axe y correspond à une erreur de roulis d'environ $0.005\,rad$. La poursuite de trajectoire selon l'axe z comporte une erreur plus importante. Plusieurs éléments peuvent expliquer ceci : premièrement, le capteur Spatial possède une précision moins élevée selon le plan vertical que selon le plan horizontal et deuxièmement, puisque nous n'effectuons pas la correspondance entre les forces/moments et la vitesse des moteurs, il est

possible que la poussée totale fournie au processeur LL ne soit pas correctement normalisée[61].

### 6.2.3 Test de vol avec la loi d'autonavigation

Cette section présente les performances du contrôleur backstepping lorsqu'il est employé avec la loi d'autonavigation. L'analyse du fonctionnement de celle-ci dépasse les cadres de ce document et ne sera pas présentée ici. Nous avons choisi un scénario de test où la première antenne est fixe et la deuxième antenne est en mouvement. Les paramètres pour les trajectoires de vitesse sont présentés dans le Tableau 6.15. Il est important de noter que le quadrotor maintient une altitude et un lacet fixe durant l'ensemble de la mission.

Tableau 6.15 Paramètres des trajectoires

| Paramètres | Valeurs |
|---|---|
| Accélération maximale (x-y) | $1\,m/s^2$ |
| Jerk maximal (x-y) | $0.5\,m/s^3$ |

Le Tableau 6.16 présente les messages ROS enregistrés durant le vol.

Tableau 6.16 Messages ROS et les données enregistrées

| Message | Données | Fréquence (hz) |
|---|---|---|
| /wifi1 | RSSI | 10 |
| /wifi2 | RSSI | 10 |
| /ans/position | $x \quad y \quad z \quad \dot{x} \quad \dot{y} \quad \dot{z}$ | 250 |
| /mcu/trajectory_X | $x_d \quad \dot{x}_d \quad \ddot{x}_d$ | 50 |

[61] Toutes les données transmises au processeur LL doivent être normalisées pour être envoyées dans le format d'un nombre entier étant dans l'intervalle $[0 \quad 200]$. Dans le cas de la force totale de poussée, celle-ci est normalisée avec la valeur de la force totale pouvant être générée par les moteurs, soit $32N$ en se basant sur (Wang, Raffler *et.al.*, 2012). Il est possible que cette valeur ne soit pas exacte.

| | | |
|---|---|---|
| /mcu/trajectory_Y | $y_d \quad \dot{y}_d \quad \ddot{y}_d$ | 50 |
| /ant1/position | Position LLA | 20 |
| /ant2/position | Position LLA | 20 |

La Figure 6.25 présente la poursuite de trajectoire, générée par la loi d'autonavigation, par le quadrotor dans le plan x-y. La position optimale est calculée comme étant la position située directement au milieu des deux antennes.

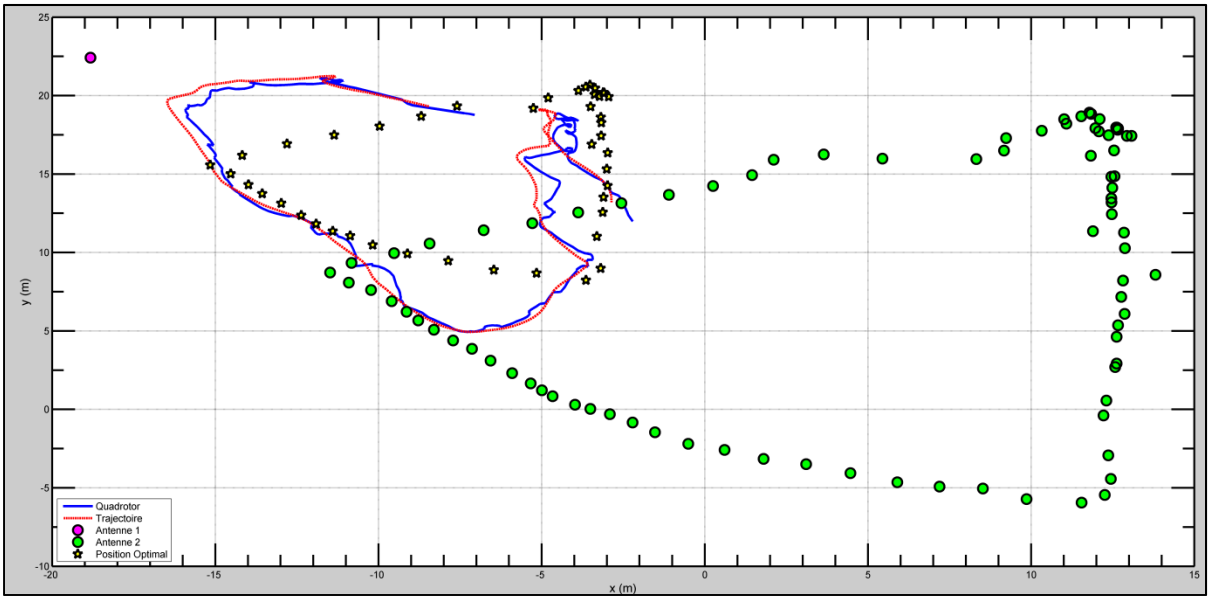
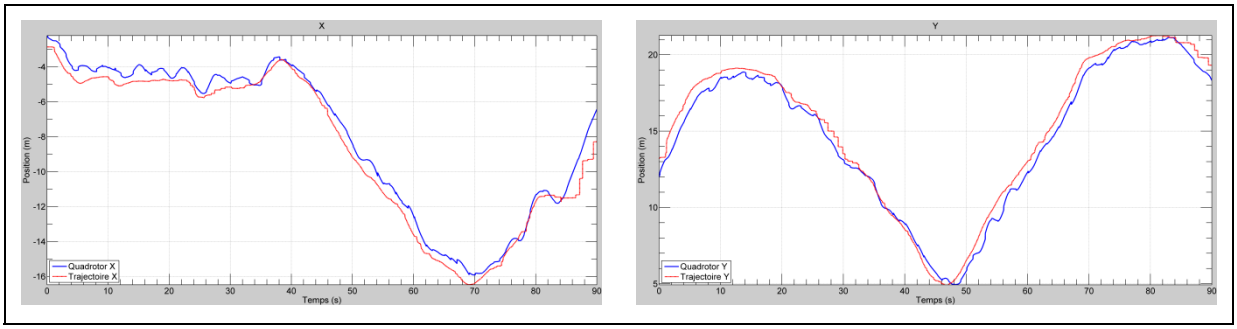



Figure 6.25 Poursuite de la trajectoire générée par la loi d'autonavigation

La Figure 6.26 présente la poursuite de trajectoire selon chacun des axes. L'axe z est maintenu à sa position initiale durant l'ensemble du vol.



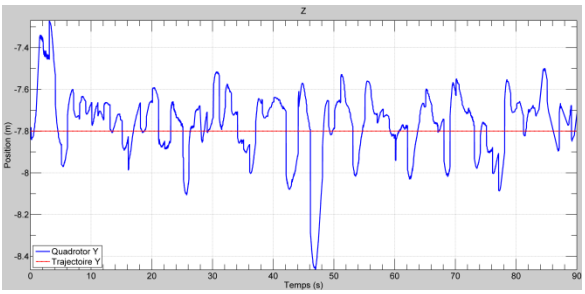



Figure 6.26 Poursuite de trajectoires par le quadrotor selon $x$, $y$ et $z$

La Figure 6.27 présente les poursuites de vitesse. Le profil de celui-ci est déterminé par les commandes de vitesses provenant de la loi d'autonavigation et des paramètres du Tableau 6.15. Tout comme en simulation, une saturation est aussi ajoutée aux commandes de vitesse. Pour ce test de vol, la vitesse est saturée à 1 $m/s$.

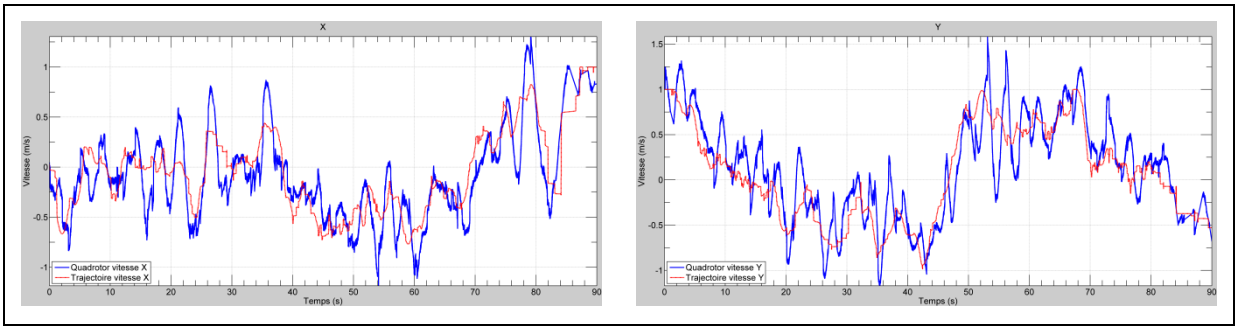



Figure 6.27 Poursuite de trajectoires par le quadrotor selon $\dot{x}$ et $\dot{y}$

Comme nous pouvons le constater, le contrôleur backstepping proposé est efficace dans le cadre d'une application réelle. En outre, il permet d'opérer le quadrotor de façon autonome tout en le couplant à l'algorithme d'autonavigation.

# CONCLUSION

Nous avons abordé, dans ce mémoire, la problématique du contrôle d'un hélicoptère à quatre hélices de type quadrotor dans le but d'effectuer des missions autonomes. Ce type d'appareil possède une dynamique hautement non linéaire, couplée et sous-actionnée, ce qui augmente le défi rencontré par le concepteur.

L'objectif de ce mémoire était de concevoir un contrôleur non linéaire permettant de contrôler quatre degrés de liberté de l'appareil, soit la position tridimensionnelle et le lacet, selon des trajectoires prédéfinies. Pour ce faire, un modèle non linéaire du quadrotor a été mis sur pied en se basant sur les équations de mouvement de Newton. À partir de ce modèle, la construction d'un contrôleur backstepping a été présentée. Puisque la dynamique du quadrotor est d'un ordre élevé, l'utilisation d'un estimateur de dérivée par mode glissant d'ordre 2 est nécessaire pour calculer la dérivée en fonction du temps de certains contrôleurs virtuels. La stabilité du contrôleur backstepping est analysée face aux erreurs d'estimation ainsi que face à l'effet de la dynamique sous-actionnée du système à l'aide de la théorie des systèmes ISS.

Le contrôleur a été validé par deux méthodes distinctes : la première consiste en une simulation de la dynamique et du contrôleur et où plusieurs scénarios de simulation nous ont permis de valider la conception du contrôleur ainsi que d'analyser ses performances. La deuxième méthode consiste en une validation expérimentale du contrôleur. Ceci a requis l'utilisation d'un quadrotor de type Asctec Pelican dont le capteur de position a été remplacé afin d'améliorer les performances. L'implémentation d'un système de vol complet a été nécessaire pour pouvoir déployer l'appareil de manière autonome. Plusieurs essais de vol ont été effectués avec succès, malgré la présence de perturbations, tel que le vent. Pour finir, le contrôleur a été validé dans une opération réelle, alors qu'il servait de système de contrôle à un algorithme d'autonavigation.

# RECOMMANDATIONS

Le vol expérimental d'aéronef, particulièrement celui d'un drone, est toujours une opération complexe et risquée. Dans le cadre de ce mémoire, nous avons réussi à mettre en place l'architecture nécessaire pour effectuer les tests de vol de manière sécuritaire autant pour les usagers que pour le prototype, et ceci, principalement, au prix de performance. Maintenant que le système a été validé et testé, il est envisageable, dans des travaux subséquents, d'améliorer les capacités du système.

Premièrement, il est désormais possible d'utiliser le système original pour effectuer des tests de vol pour des vitesses et accélérations plus élevées. Aussi, une variation plus importante de l'altitude lors des tests démontrerait plus adéquatement la poursuite de trajectoire selon l'axe Z. Une calibration plus exhaustive des gains du contrôleur auraient comme effet d'améliorer les performances. De plus, de nouvelles fonctionnalités sont envisageables pour le quadrotor en utilisant les systèmes déjà en place tel que par exemple : l'implémentation d'un algorithme de génération de trajectoire pouvant gérer le décollage et l'atterrissage du contrôleur de façon autonome et un algorithme de retour à la base automatique.

Il est intéressant de noter que le contrôleur proposé ne tient pas en compte de la configuration des moteurs. En effet, le modèle ne considère que la force et les moments appliqués sur le point de masse, mais pas les éléments qui les génèrent. Il est de notre avis que le contrôleur peut être utilisé pour des appareils possédant la même dynamique, mais pas la même configuration de moteurs, par exemple un hexacoptère ou un hélicoptère classique. Il suffit au concepteur de modifier la matrice de correspondance force/moment à moteur, l'équation (2.35), de manière à tenir compte de la nouvelle configuration.

Une modification du contrôleur est aussi envisageable. En effet, le backstepping est une excellente stratégie de contrôle qui s'applique bien à la dynamique du quadrotor, mais elle comporte tout de même certains défauts. Premièrement, elle exige une précision assez élevée des capteurs, ce qui, dans des applications extérieures, peut parfois laisser à désirer.

Deuxièmement, elle est peu robuste, car elle dépend directement des paramètres du système qui peuvent parfois être difficilement mesurables, comme par exemple l'inertie. Troisièmement, le modèle devrait être modifié pour tenir en compte de l'erreur d'alignement entre le centre de gravité et le centre géométrique de l'appareil. Quatrièmement, le backstepping devient de plus en plus complexe mathématiquement lorsque l'ordre du système augmente. Pour pallier à ces défauts, nous croyons qu'une simplification de la dynamique couplée à un contrôleur backstepping adaptatif pourrait rendre le contrôleur plus robuste. Compte tenu de la précision du capteur disponible, les trajectoires choisies ne peuvent être agressives, de telle sorte que les angles d'attitude du quadrotor demeurent peu élevés. Cette hypothèse, déjà utilisée par plusieurs publications, notamment (Bouabdallah, 2007) permet de diminuer la complexité du modèle. Ceci nous donnerait la flexibilité nécessaire pour ajouter facilement un contrôleur adaptatif, jumelé au contrôleur backstepping, qui pourrait estimer les perturbations du vent ainsi que les erreurs de modélisation. L'ensemble du contrôleur devrait ensuite être analysé afin d'assurer la stabilité du système face aux erreurs de modélisation et d'estimation.

Nous notons aussi que l'implémentation du contrôleur sur deux unités de calcul différentes apporte une dégradation importante de la performance. Idéalement, le contrôleur devrait être implémenté sur une seule unité de calcul, en temps réel dur. L'unité de calcul devrait idéalement effectuer l'ensemble des mesures de tous les capteurs et appliquer directement les signaux de contrôle.

Pour finir, nous notons que la communication Wifi possède une très faible portée et devrait être remplacée par d'autres formes d'ondes[62] pour effectuer des tests sur de plus grandes distances tout en permettant de mesurer toutes les données de vol.

---

[62] Par exemple, un module de télécommunication série via RF tel que les modules hautes performances de la compagnie Digi International.

## ANNEXE I

## INVERSION DE LA DYNAMIQUE DE TRANSLATION

Nous cherchons à inverser la dynamique de translation linéaire du quadrotor pour trouver la valeur des angles d'Euler en fonction de l'accélération du quadrotor.

En nous basant sur la dynamique de translation du quadrotor, soit la deuxième équation de $\Pi_1$ (4.3), nous pouvons écrire :

$$\boldsymbol{R}^T\left(\begin{bmatrix}0\\0\\g\end{bmatrix}-\begin{bmatrix}\ddot{x}\\\ddot{y}\\\ddot{z}\end{bmatrix}\right)=\begin{bmatrix}0\\0\\T/m\end{bmatrix} \tag{A I-1}$$

La première ligne de (A I-1) s'écrit ainsi :

$$0=-\ddot{x}c_\theta c_\psi-\ddot{y}c_\theta s_\psi-(g-\ddot{z})s_\theta \tag{A I-2}$$

On peut réorganiser (A I-2) tel que :

$$\theta=\tan^{-1}\left(\frac{\ddot{x}c_\psi+\ddot{y}s_\psi}{\ddot{z}-g}\right) \tag{A I-3}$$

La deuxième ligne de (A I-1) s'écrit ainsi :

$$0=\ddot{x}\left(c_\phi s_\psi-s_\phi s_\theta c_\psi\right)-\ddot{y}\left(c_\phi c_\psi+s_\phi s_\theta s_\psi\right)+(g-\ddot{z})\left(c_\phi s_\psi\right) \tag{A I-4}$$

On peut réorganiser (A I-4) tel que :

$$\left(c_\theta s_\phi\right)(\ddot{z}-g)=c_\phi\left(\ddot{x}s_\psi-\ddot{y}c_\psi\right)-\left(s_\phi s_\theta\right)\left(\ddot{x}c_\psi+\ddot{y}s_\psi\right)$$

$$\frac{c_\phi}{s_\phi}=\frac{c_\theta(\ddot{z}-g)+s_\theta\left(\ddot{x}c_\psi+\ddot{y}s_\psi\right)}{\ddot{x}s_\psi-\ddot{y}c_\psi}$$

$$\phi=\tan^{-1}\left(\frac{\ddot{x}s_\psi-\ddot{y}c_\psi}{c_\theta(\ddot{z}-g)+s_\theta\left(\ddot{x}c_\psi+\ddot{y}s_\psi\right)}\right) \tag{A I-5}$$

Les équations (A I-3) et (A I-5) nous permettent de constater que pour générer de grandes accélérations, les valeurs de tangage et roulis approchent les valeurs de $\pm\frac{\pi}{2}$ qui sont des points de singularité, tel que définis aux sections 2.6 et 4.4.5.

Cette analyse nous permet de constater que certaines trajectoires très agressives ne peuvent pas être accomplies dans un régime stable d'opération.

Une manière de mitiger cette propriété est de choisir des accélérations maximales au niveau de la trajectoire désirée qui respecteraient un sous-ensemble de la plage théorique d'opération de l'appareil. De plus, une saturation des angles de tangage et de roulis peut être ajoutée pour s'assurer que les singularités ne sont jamais atteintes en pratique. Par exemple, une limite de tangage de $\frac{\pi}{4}$ implique, approximativement, une accélération de un g, ce qui est suffisant pour grand nombre d'applications[63].

[63] On obtient cette approximation en posant une condition de vol à altitude constante ($\ddot{z} = 0$) et un lacet nul à l'équation (A I-3).

## ANNEXE II

## COMPOSITION DE LA MATRICE DE SÉPARATION $\boldsymbol{Q}(\boldsymbol{\Theta}_e, \boldsymbol{\alpha}_\Theta)$

Notre but est de trouver les éléments de la matrice $\boldsymbol{Q}(\boldsymbol{\Theta}_e, \boldsymbol{\alpha}_\Theta)$. Puisque la démonstration prend beaucoup d'espace, celle-ci est effectuée pour le premier élément de la matrice $\boldsymbol{Q}(\boldsymbol{\Theta}_e, \boldsymbol{\alpha}_\Theta)$. Chacun des premiers éléments des matrices est noté $(\cdot)_{11}$ dans cette annexe.

Reprenons l'équation (4.16) :

$$\boldsymbol{R}(\boldsymbol{\alpha}_\Theta) = \boldsymbol{R}(\boldsymbol{\Theta}) - \boldsymbol{Q}(\boldsymbol{\Theta}_e, \boldsymbol{\alpha}_\Theta) \tag{A II-1}$$

L'équation (A II-1) peut être réécrite pour le premier élément de chaque matrice ainsi :

$$Q_{11}(\boldsymbol{\Theta}_e, \boldsymbol{\alpha}_\Theta) = R_{11}(\boldsymbol{\Theta}) - R_{11}(\boldsymbol{\alpha}_\Theta) \tag{A II-2}$$

En notant, à partir de (4.15) que $\boldsymbol{\Theta} = \boldsymbol{\alpha}_\Theta + \boldsymbol{\Theta}_e$ et à partir de (2.6), nous pouvons réécrire (A II-1) ainsi :

$$Q_{11}(\boldsymbol{\Theta}_e, \boldsymbol{\alpha}_\Theta) = \cos(\alpha_\theta + \theta_e) \cdot \cos\left(\alpha_\psi + \psi_e\right) - c_{\boldsymbol{\alpha}_\theta} c_{\boldsymbol{\alpha}_\psi} \tag{A II-3}$$

Par trigonométrie on peut écrire :

$$\begin{aligned} &\cos(\alpha_\theta + \theta_e) \cdot \cos\left(\alpha_\psi + \psi_e\right) \\ &= c_{\boldsymbol{\alpha}_\theta} c_{\theta_e} c_{\boldsymbol{\alpha}_\psi} c_{\psi_e} - c_{\boldsymbol{\alpha}_\theta} c_{\theta_e} s_{\boldsymbol{\alpha}_\psi} s_{\psi_e} - c_{\boldsymbol{\alpha}_\psi} c_{\psi_e} s_{\boldsymbol{\alpha}_\theta} s_{\theta_e} + s_{\boldsymbol{\alpha}_\theta} s_{\theta_e} s_{\boldsymbol{\alpha}_\psi} s_{\psi_e} \end{aligned} \tag{A II-4}$$

On peut voir que le premier terme de (A II-4) contient le terme $c_{\boldsymbol{\alpha}_\theta} c_{\boldsymbol{\alpha}_\psi}$. Écrivons (A II-3) sous la forme (A II-4) en mettant en évidence le premier terme de (A II-4) ainsi que $c_{\boldsymbol{\alpha}_\theta} c_{\boldsymbol{\alpha}_\psi}$.

$$Q_{11}(\boldsymbol{\Theta}_e, \boldsymbol{\alpha}_\Theta) = c_{\boldsymbol{\alpha}_\theta} c_{\boldsymbol{\alpha}_\psi}\left(c_{\theta_e} c_{\psi_e} - 1\right) - c_{\boldsymbol{\alpha}_\theta} c_{\theta_e} s_{\boldsymbol{\alpha}_\psi} s_{\psi_e} - c_{\boldsymbol{\alpha}_\psi} c_{\psi_e} s_{\boldsymbol{\alpha}_\theta} s_{\theta_e} + s_{\boldsymbol{\alpha}_\theta} s_{\theta_e} s_{\boldsymbol{\alpha}_\psi} s_{\psi_e} \quad \text{(A II-5)}$$

L'analyse de l'élément $Q_{11}(\boldsymbol{\Theta}_e, \boldsymbol{\alpha}_\Theta)$ peut être généralisée à l'ensemble de la matrice par la suite.

Premièrement, puisque $Q_{11}(\boldsymbol{\Theta}_e, \boldsymbol{\alpha}_\Theta)$ n'est constitué que de fonctions cosinus et sinus, nous pouvons conclure que cet élément est borné, et ceci, quelle que soit la valeur des vecteurs $\boldsymbol{\Theta}_e$ et $\boldsymbol{\alpha}_\Theta$.

Deuxièmement, si $\boldsymbol{\Theta}_e$ tend vers zéro, alors l'élément $Q_{11}(\boldsymbol{\Theta}_e, \boldsymbol{\alpha}_\Theta)$ tend vers zéro. Ceci est facilement vérifiable en observant chaque terme de $Q_{11}(\boldsymbol{\Theta}_e, \boldsymbol{\alpha}_\Theta)$.

En appliquant sur l'ensemble des éléments de $\boldsymbol{Q}(\boldsymbol{\Theta}_e, \boldsymbol{\alpha}_\Theta)$, on peut déduire les deux propriétés suivantes. Il existe une petite constante positive $\varepsilon$ telle que :

$$\|\boldsymbol{Q}(\boldsymbol{\Theta}_e, \boldsymbol{\alpha}_\Theta)\| < \varepsilon\,, \qquad \forall\, t \geq 0\,,\ \boldsymbol{\Theta}_e \in \mathbb{R}^3,\ \boldsymbol{\alpha}_\Theta \in \mathbb{R}^3 \quad \text{(A II-6)}$$

$$\boldsymbol{\Theta}_e \to \mathbf{0} \Rightarrow \lim_{t\to\infty} \boldsymbol{Q}(\boldsymbol{\Theta}_e, \boldsymbol{\alpha}_\Theta) = 0\,, \qquad \forall\, \boldsymbol{\alpha}_\Theta \in \mathbb{R}^3 \quad \text{(A II-7)}$$

En appliquant la décomposition sur l'ensemble des éléments, on obtient les éléments de la matrice $\boldsymbol{Q}(\boldsymbol{\Theta}_e, \boldsymbol{\alpha}_\Theta)$ :

$$
\begin{aligned}
Q_{12}(\boldsymbol{\Theta}_e, \boldsymbol{\alpha}_\Theta) = & -c_{\boldsymbol{\alpha}_\phi} c_{\boldsymbol{\alpha}_\psi} c_{\phi_e} s_{\psi_e} - c_{\boldsymbol{\alpha}_\phi} s_{\boldsymbol{\alpha}_\psi} \left(c_{\phi_e} s_{\psi_e} - 1\right) \\
& + s_{\boldsymbol{\alpha}_\phi} c_{\boldsymbol{\alpha}_\psi} s_{\phi_e} s_{\psi_e} \\
& + s_{\boldsymbol{\alpha}_\phi} s_{\boldsymbol{\alpha}_\psi} s_{\phi_e} c_{\psi_e} + c_{\boldsymbol{\alpha}_\phi} c_{\boldsymbol{\alpha}_\theta} c_{\boldsymbol{\alpha}_\psi} s_{\phi_e} s_{\theta_e} c_{\psi_e} \\
& - s_{\boldsymbol{\alpha}_\phi} s_{\boldsymbol{\alpha}_\theta} s_{\boldsymbol{\alpha}_\psi} c_{\phi_e} c_{\theta_e} s_{\psi_e} + s_{\boldsymbol{\alpha}_\phi} s_{\boldsymbol{\alpha}_\theta} c_{\boldsymbol{\alpha}_\psi} \left(c_{\phi_e} s_{\theta_e} s_{\psi_e} - 1\right) \\
& + c_{\boldsymbol{\alpha}_\phi} s_{\boldsymbol{\alpha}_\theta} c_{\boldsymbol{\alpha}_\psi} s_{\phi_e} c_{\theta_e} c_{\psi_e} + s_{\boldsymbol{\alpha}_\phi} s_{\boldsymbol{\alpha}_\theta} c_{\boldsymbol{\alpha}_\psi} c_{\phi_e} c_{\theta_e} c_{\psi_e} \\
& - c_{\boldsymbol{\alpha}_\phi} c_{\boldsymbol{\alpha}_\theta} c_{\boldsymbol{\alpha}_\psi} c_{\phi_e} s_{\theta_e} s_{\psi_e} - c_{\boldsymbol{\alpha}_\phi} c_{\boldsymbol{\alpha}_\theta} c_{\boldsymbol{\alpha}_\psi} c_{\phi_e} s_{\theta_e} s_{\psi_e} \\
& - c_{\boldsymbol{\alpha}_\phi} c_{\boldsymbol{\alpha}_\theta} c_{\boldsymbol{\alpha}_\psi} c_{\phi_e} s_{\theta_e} s_{\psi_e}
\end{aligned} \tag{A II-8}
$$

$$
\begin{aligned}
Q_{13}(\boldsymbol{\Theta}_e, \boldsymbol{\alpha}_\Theta) = & +c_{\boldsymbol{\alpha}_\phi} c_{\boldsymbol{\alpha}_\psi} s_{\phi_e} s_{\psi_e} + s_{\boldsymbol{\alpha}_\phi} s_{\boldsymbol{\alpha}_\psi} \left(c_{\phi_e} c_{\psi_e} - 1\right) \\
& + c_{\boldsymbol{\alpha}_\phi} s_{\boldsymbol{\alpha}_\psi} s_{\phi_e} c_{\psi_e} + s_{\boldsymbol{\alpha}_\phi} c_{\boldsymbol{\alpha}_\psi} c_{\phi_e} s_{\psi_e} \\
& + c_{\boldsymbol{\alpha}_\phi} c_{\boldsymbol{\alpha}_\theta} c_{\boldsymbol{\alpha}_\psi} c_{\phi_e} s_{\theta_e} c_{\psi_e} \\
& - c_{\boldsymbol{\alpha}_\phi} c_{\boldsymbol{\alpha}_\theta} s_{\boldsymbol{\alpha}_\psi} c_{\phi_e} s_{\theta_e} s_{\psi_e} + c_{\boldsymbol{\alpha}_\phi} s_{\boldsymbol{\alpha}_\theta} c_{\boldsymbol{\alpha}_\psi} \left(c_{\phi_e} c_{\theta_e} c_{\psi_e} - 1\right) \\
& - c_{\boldsymbol{\alpha}_\phi} s_{\boldsymbol{\alpha}_\theta} s_{\boldsymbol{\alpha}_\psi} c_{\phi_e} c_{\theta_e} s_{\psi_e} - s_{\boldsymbol{\alpha}_\phi} c_{\boldsymbol{\alpha}_\theta} c_{\boldsymbol{\alpha}_\psi} s_{\phi_e} s_{\theta_e} c_{\psi_e} \\
& - s_{\boldsymbol{\alpha}_\phi} s_{\boldsymbol{\alpha}_\theta} c_{\boldsymbol{\alpha}_\psi} c_{\phi_e} c_{\theta_e} s_{\psi_e} + s_{\boldsymbol{\alpha}_\phi} c_{\boldsymbol{\alpha}_\theta} s_{\boldsymbol{\alpha}_\psi} s_{\phi_e} s_{\theta_e} s_{\psi_e} \\
& + s_{\boldsymbol{\alpha}_\phi} s_{\boldsymbol{\alpha}_\theta} s_{\boldsymbol{\alpha}_\psi} s_{\phi_e} c_{\theta_e} s_{\psi_e}
\end{aligned} \tag{A II-9}
$$

$$
\begin{aligned}
Q_{21}(\boldsymbol{\Theta}_e, \boldsymbol{\alpha}_\Theta) = & \, c_{\boldsymbol{\alpha}_\theta} s_{\boldsymbol{\alpha}_\psi} \left(c_{\theta_e} c_{\psi_e} - 1\right) + c_{\boldsymbol{\alpha}_\theta} c_{\boldsymbol{\alpha}_\psi} c_{\theta_e} s_{\psi_e} - s_{\boldsymbol{\alpha}_\theta} c_{\boldsymbol{\alpha}_\psi} s_{\theta_e} s_{\psi_e} \\
& - s_{\boldsymbol{\alpha}_\theta} s_{\boldsymbol{\alpha}_\psi} s_{\theta_e} c_{\psi_e}
\end{aligned} \tag{A II-10}
$$

$$
\begin{aligned}
Q_{22}(\boldsymbol{\Theta}_e, \boldsymbol{\alpha}_\Theta) = & -c_{\boldsymbol{\alpha}_\phi} s_{\boldsymbol{\alpha}_\psi} c_{\phi_e} s_{\psi_e} + c_{\boldsymbol{\alpha}_\phi} c_{\boldsymbol{\alpha}_\psi} \left(c_{\phi_e} c_{\psi_e} - 1\right) \\
& + s_{\boldsymbol{\alpha}_\phi} s_{\boldsymbol{\alpha}_\psi} s_{\phi_e} s_{\psi_e} - s_{\boldsymbol{\alpha}_\phi} c_{\boldsymbol{\alpha}_\psi} s_{\phi_e} c_{\psi_e} \\
& + c_{\boldsymbol{\alpha}_\phi} c_{\boldsymbol{\alpha}_\theta} c_{\boldsymbol{\alpha}_\psi} s_{\phi_e} s_{\theta_e} s_{\psi_e} \\
& + c_{\boldsymbol{\alpha}_\phi} s_{\boldsymbol{\alpha}_\theta} c_{\boldsymbol{\alpha}_\psi} s_{\phi_e} c_{\theta_e} s_{\psi_e} + s_{\boldsymbol{\alpha}_\phi} s_{\boldsymbol{\alpha}_\theta} s_{\boldsymbol{\alpha}_\psi} \left(c_{\phi_e} c_{\theta_e} c_{\psi_e} - 1\right) \\
& + s_{\boldsymbol{\alpha}_\phi} s_{\boldsymbol{\alpha}_\theta} c_{\boldsymbol{\alpha}_\psi} c_{\phi_e} c_{\theta_e} s_{\psi_e} + c_{\boldsymbol{\alpha}_\phi} c_{\boldsymbol{\alpha}_\theta} s_{\boldsymbol{\alpha}_\psi} s_{\phi_e} s_{\theta_e} c_{\psi_e} \\
& + s_{\boldsymbol{\alpha}_\phi} c_{\boldsymbol{\alpha}_\theta} s_{\boldsymbol{\alpha}_\psi} c_{\phi_e} s_{\theta_e} c_{\psi_e} + c_{\boldsymbol{\alpha}_\phi} s_{\boldsymbol{\alpha}_\theta} s_{\boldsymbol{\alpha}_\psi} s_{\phi_e} c_{\theta_e} c_{\psi_e} \\
& + s_{\boldsymbol{\alpha}_\phi} c_{\boldsymbol{\alpha}_\theta} c_{\boldsymbol{\alpha}_\psi} c_{\phi_e} s_{\theta_e} s_{\psi_e}
\end{aligned} \tag{A II-11}
$$

$$
\begin{aligned}
Q_{23}(\boldsymbol{\Theta}_e, \boldsymbol{\alpha}_\Theta) = & -c_{\boldsymbol{\alpha}_\phi} c_{\boldsymbol{\alpha}_\psi} s_{\phi_e} c_{\psi_e} - s_{\boldsymbol{\alpha}_\phi} c_{\boldsymbol{\alpha}_\psi} \left(c_{\phi_e} c_{\psi_e} - 1\right) \\
& + c_{\boldsymbol{\alpha}_\phi} s_{\boldsymbol{\alpha}_\psi} s_{\phi_e} s_{\psi_e} - s_{\boldsymbol{\alpha}_\phi} s_{\boldsymbol{\alpha}_\psi} c_{\phi_e} s_{\psi_e} \\
& + c_{\boldsymbol{\alpha}_\phi} c_{\boldsymbol{\alpha}_\theta} c_{\boldsymbol{\alpha}_\psi} c_{\phi_e} s_{\theta_e} s_{\psi_e} \\
& + c_{\boldsymbol{\alpha}_\phi} s_{\boldsymbol{\alpha}_\theta} c_{\boldsymbol{\alpha}_\psi} c_{\phi_e} c_{\theta_e} s_{\psi_e} + c_{\boldsymbol{\alpha}_\phi} s_{\boldsymbol{\alpha}_\theta} s_{\boldsymbol{\alpha}_\psi} \left(c_{\phi_e} c_{\theta_e} c_{\psi_e} - 1\right) \\
& + c_{\boldsymbol{\alpha}_\phi} c_{\boldsymbol{\alpha}_\theta} s_{\boldsymbol{\alpha}_\psi} c_{\phi_e} s_{\theta_e} c_{\psi_e} - s_{\boldsymbol{\alpha}_\phi} c_{\boldsymbol{\alpha}_\theta} c_{\boldsymbol{\alpha}_\psi} s_{\phi_e} s_{\theta_e} s_{\psi_e} \\
& - s_{\boldsymbol{\alpha}_\phi} s_{\boldsymbol{\alpha}_\theta} c_{\boldsymbol{\alpha}_\psi} s_{\phi_e} c_{\theta_e} s_{\psi_e} - s_{\boldsymbol{\alpha}_\phi} c_{\boldsymbol{\alpha}_\theta} s_{\boldsymbol{\alpha}_\psi} s_{\phi_e} s_{\theta_e} c_{\psi_e} \\
& - s_{\boldsymbol{\alpha}_\phi} s_{\boldsymbol{\alpha}_\theta} s_{\boldsymbol{\alpha}_\psi} s_{\phi_e} c_{\theta_e} c_{\psi_e}
\end{aligned} \tag{A II-12}
$$

$$
Q_{31}(\boldsymbol{\Theta}_e, \boldsymbol{\alpha}_\Theta) = -s_{\boldsymbol{\alpha}_\theta} \left(c_{\theta_e} - 1\right) - c_{\boldsymbol{\alpha}_\theta} s_{\theta_e} \tag{A II-13}
$$

$$
\begin{aligned}
Q_{32}(\boldsymbol{\Theta}_e, \boldsymbol{\alpha}_\Theta) = & \, c_{\boldsymbol{\alpha}_\phi} s_{\boldsymbol{\alpha}_\theta} \left(c_{\phi_e} c_{\theta_e} - 1\right) + c_{\boldsymbol{\alpha}_\phi} c_{\boldsymbol{\alpha}_\theta} s_{\phi_e} c_{\theta_e} - c_{\boldsymbol{\alpha}_\phi} s_{\boldsymbol{\alpha}_\theta} s_{\phi_e} s_{\theta_e} \\
& - s_{\boldsymbol{\alpha}_\phi} s_{\boldsymbol{\alpha}_\theta} c_{\phi_e} s_{\theta_e}
\end{aligned} \tag{A II-14}
$$

$$
\begin{aligned}
Q_{33}(\boldsymbol{\Theta}_e, \boldsymbol{\alpha}_\Theta) = & \, c_{\boldsymbol{\alpha}_\phi} c_{\boldsymbol{\alpha}_\theta} \left(c_{\phi_e} c_{\theta_e} - 1\right) - s_{\boldsymbol{\alpha}_\phi} c_{\boldsymbol{\alpha}_\theta} s_{\phi_e} c_{\theta_e} - c_{\boldsymbol{\alpha}_\phi} s_{\boldsymbol{\alpha}_\theta} c_{\phi_e} s_{\theta_e} \\
& + s_{\boldsymbol{\alpha}_\phi} s_{\boldsymbol{\alpha}_\theta} s_{\phi_e} s_{\theta_e}
\end{aligned} \tag{A II-15}
$$

## ANNEXE III

## ANALYSE ISS DU SYSTÈME EN BOUCLE FERMÉE

Le système en boucle fermée possède plusieurs perturbations dont les effets doivent être étudiés pour s'assurer que l'état du système demeure borné en tout temps face à ceux-ci. De plus, on doit s'assurer que l'état tend vers zéro pour atteindre nos objectifs de conception.

Le système en boucle fermée peut s'écrire sous la forme matricielle suivante :

$$\begin{bmatrix} \dot{\boldsymbol{\eta}}_e \\ \dot{\boldsymbol{\mu}}_e \\ \dot{\boldsymbol{\Theta}}_e \\ \boldsymbol{I}\dot{\boldsymbol{\omega}}_e \end{bmatrix} = \begin{bmatrix} -\boldsymbol{\Lambda}_1 & 1 & 0 & 0 \\ -1 & -\boldsymbol{\Lambda}_2 & 0 & 0 \\ 0 & 0 & -\boldsymbol{\Lambda}_3 & \boldsymbol{\rho}(\boldsymbol{\Theta}) \\ 0 & 0 & -\boldsymbol{\rho}(\boldsymbol{\Theta})^T & -\boldsymbol{\Lambda}_4 \end{bmatrix} \begin{bmatrix} \boldsymbol{\eta}_e \\ \boldsymbol{\mu}_e \\ \boldsymbol{\Theta}_e \\ \boldsymbol{\omega}_e \end{bmatrix} - \begin{bmatrix} 0 \\ \frac{T}{m}\boldsymbol{Q}(\boldsymbol{\Theta}_e, \boldsymbol{\alpha}_\Theta)\boldsymbol{k} \\ \widetilde{\dot{\boldsymbol{\alpha}}}_\Theta \\ \widetilde{\dot{\boldsymbol{\alpha}}}_\omega \end{bmatrix} \quad \text{(A III-1)}$$

Soit le vecteur d'état total $\boldsymbol{X} = [\boldsymbol{\eta}_e{}^T \quad \boldsymbol{\mu}_e{}^T \quad \boldsymbol{\Theta}_e{}^T \quad \boldsymbol{\omega}_e{}^T]^T$ et le vecteur de perturbation $\boldsymbol{u} = \left[0 \quad \left(\frac{T}{m}\boldsymbol{Q}(\boldsymbol{\Theta}_e, \boldsymbol{\alpha}_\Theta)\boldsymbol{k}\right)^T \quad \widetilde{\dot{\boldsymbol{\alpha}}}_\Theta{}^T \quad \widetilde{\dot{\boldsymbol{\alpha}}}_\omega{}^T\right]^T$. À l'aide de ceux-ci, nous pouvons réécrire la fonction candidate de Lyapunov du système total ainsi :

$$V_{BF} = \frac{1}{2}\boldsymbol{\eta}_e{}^T\boldsymbol{\eta}_e + \frac{1}{2}\boldsymbol{\mu}_e{}^T\boldsymbol{\mu}_e + \frac{1}{2}\boldsymbol{\Theta}_e{}^T\boldsymbol{\Theta}_e + \frac{1}{2}\boldsymbol{\omega}_e{}^T\boldsymbol{I}\boldsymbol{\omega}_e \quad \text{(A III-2)}$$

$$\frac{1}{2}\lambda_{min}(\boldsymbol{I})\|\boldsymbol{X}\|_2^2 \leq V_{BF} \leq \frac{1}{2}\lambda_{max}(\boldsymbol{I})\|\boldsymbol{X}\|_2^2 \quad \text{(A III-3)}$$

Où les opérateurs $\lambda_{min}(\boldsymbol{M})$ et $\lambda_{max}(\boldsymbol{M})$ retournent respectivement la plus petite et la plus grande valeur propre de la matrice $\boldsymbol{M}$.

À partir du même principe, nous pouvons réécrire la fonction $\dot{V}_{BF}$ ainsi :

$$\dot{V}_{BF} = -\boldsymbol{\eta_e}^T \boldsymbol{\Lambda_1} \boldsymbol{\eta_e} - \boldsymbol{\mu_e}^T \boldsymbol{\Lambda_2} \boldsymbol{\mu_e} - \boldsymbol{\Theta_e}^T \boldsymbol{\Lambda_3} \boldsymbol{\Theta_e} - \boldsymbol{\omega_e}^T \boldsymbol{\Lambda_4} \boldsymbol{\omega_e} - \boldsymbol{\mu_e}^T \frac{T}{m} \boldsymbol{Q}(\boldsymbol{\Theta_e}, \boldsymbol{\alpha_\Theta}) \boldsymbol{k} - \boldsymbol{\Theta_e}^T \widetilde{\dot{\boldsymbol{\alpha}}}_\Theta - \boldsymbol{\omega_e}^T \boldsymbol{I} \widetilde{\dot{\boldsymbol{\alpha}}}_\omega \quad \text{(A III-4)}$$

$$\dot{V}_{BF} \leq -\lambda_{min}(\boldsymbol{\Lambda_1}) \|\boldsymbol{\eta_e}\|_2^2 - \lambda_{min}(\boldsymbol{\Lambda_2}) \|\boldsymbol{\mu_e}\|_2^2 - \lambda_{min}(\boldsymbol{\Lambda_3}) \|\boldsymbol{\Theta_e}\|_2^2 - \lambda_{min}(\boldsymbol{\Lambda_4}) \|\boldsymbol{\omega_e}| - \boldsymbol{\mu_e}^T \frac{T}{m} \boldsymbol{Q}(\boldsymbol{\Theta_e}, \boldsymbol{\alpha_\Theta}) \boldsymbol{k} - \boldsymbol{\Theta_e}^T \widetilde{\dot{\boldsymbol{\alpha}}}_\Theta - \boldsymbol{\omega_e}^T \boldsymbol{I} \widetilde{\dot{\boldsymbol{\alpha}}}_\omega \quad \text{(A III-5)}$$

$$\dot{V}_{BF} \leq -\lambda_{\boldsymbol{\Lambda}} \|\boldsymbol{X}\|_2^2 - \boldsymbol{X}^T \boldsymbol{u} \quad \text{(A III-6)}$$

Où $\lambda_{\boldsymbol{\Lambda}} = min\{\lambda_{min}(\boldsymbol{\Lambda_1})\,,\ \lambda_{min}(\boldsymbol{\Lambda_2})\,,\ \lambda_{min}(\boldsymbol{\Lambda_3})\,,\ \lambda_{min}(\boldsymbol{\Lambda_4})\}$. On peut voir que (A III-6) est de la forme particulière présentée à la section 3.3.3. Nous pouvons compléter la preuve en suivant la procédure telle que :

$$\dot{V}_{BF} \leq -\lambda_{\boldsymbol{\Lambda}} \|\boldsymbol{X}\|_2^2 + \|\boldsymbol{X}\|_2 \|\boldsymbol{u}\|_{\boldsymbol{2}} \quad \text{(A III-7)}$$

Soit la constante $0 < \varepsilon < 1$ telle qu'on peut écrire :

$$\dot{V}_{BF} \leq -\lambda_{\boldsymbol{\Lambda}}(1-\varepsilon) \|\boldsymbol{X}\|_2^2 - \lambda_{\boldsymbol{\Lambda}} \varepsilon \|\boldsymbol{X}\|_2^2 + \|\boldsymbol{X}\|_2 \|\boldsymbol{u}\|_{\boldsymbol{2}} \quad \text{(A III-8)}$$

L'inéquation (A III-8) est toujours inférieure ou égale à zéro si :

$$-\lambda_{\boldsymbol{\Lambda}} \varepsilon \|\boldsymbol{X}\|_2^2 + \|\boldsymbol{X}\|_2 \|\boldsymbol{u}\|_{\boldsymbol{2}} \leq 0 \quad \text{(A III-9)}$$

Cette condition peut être réécrite ainsi :

$$\|\boldsymbol{X}\|_2 \geq \frac{\|\boldsymbol{u}\|_{\boldsymbol{2}}}{\lambda_{\boldsymbol{\Lambda}} \varepsilon} \quad \text{(A III-10)}$$

tel que nous pouvons réécrire la dérivée de la fonction de Lyapunov du système en boucle fermée réponds aux conditions de l'équation (3.15) tel que :

$$\dot{V}_{BF} \leq -\lambda_{\boldsymbol{\Lambda}}(1-\varepsilon)\|\boldsymbol{X}\|_2^2, \qquad \|\boldsymbol{X}\|_2 \geq \frac{\|\boldsymbol{u}\|_2}{\lambda_{\boldsymbol{\Lambda}}\varepsilon}, t > 0 \tag{A III-11}$$

Ceci nous permet de définir la fonction classe $\mathcal{K}_\infty$ $\rho(x)$ suivante :

$$\rho(r) = \frac{r}{\lambda_{\boldsymbol{\Lambda}}\varepsilon} \tag{A III-12}$$

La fonction $\gamma$ associée au système ISS peut donc être écrite telle que :

$$\begin{aligned}\gamma(r) &= {\alpha_1}^{-1} \circ \alpha_2 \circ \rho \\ &= \sqrt{\frac{\lambda_{max}(\boldsymbol{I})}{\lambda_{min}(\boldsymbol{I})}}\left(\frac{r}{\lambda_{\boldsymbol{\Lambda}}\varepsilon}\right)\end{aligned} \tag{A III-13}$$

Où $\alpha_1$ et $\alpha_2$ sont les fonctions classes $\mathcal{K}$ définies dans l'inéquation (A III-3).

De plus, compte tenu des propriétés ISS du système et du fait que les erreurs d'estimation tendent vers zéro en temps fini grâce aux propriétés de l'estimateur les erreurs, nous pouvons conclure que les états $\boldsymbol{\Theta}_e$ et $\boldsymbol{\omega}_e$ tendent asymptotiquement vers zéro. De plus puisque $\boldsymbol{\Theta}_e \to \boldsymbol{0}$ lorsque $t \to \infty$, alors $\|\boldsymbol{Q}(\boldsymbol{\Theta}_e, \boldsymbol{\alpha}_\Theta)\| \to 0$, tel que présenté à l'ANNEXE II, nous pouvons conclure que l'état $\boldsymbol{\mu}_e$ tend également asymptotiquement vers zéro de telle sorte que l'origine du système en boucle fermée est globalement asymptotiquement stable.

## ANNEXE IV

## CONVERSION VITESSE DE MOTEUR ASCTEC À SI

La vitesse des moteurs mesurée par la carte LL est normalisée à l'aide d'une équation dont nous ne connaissons pas les paramètres. Pour connaître la correspondance entre la vitesse normalisée et la vitesse en rad/s, nous avons choisi de faire des mesures expérimentales sur l'ensemble de la plage du moteur. Pour ce faire nous avons programmé le Pelican pour contrôler directement la vitesse du moteur 1. La Figure A IV-1 présente le résultat des mesures.

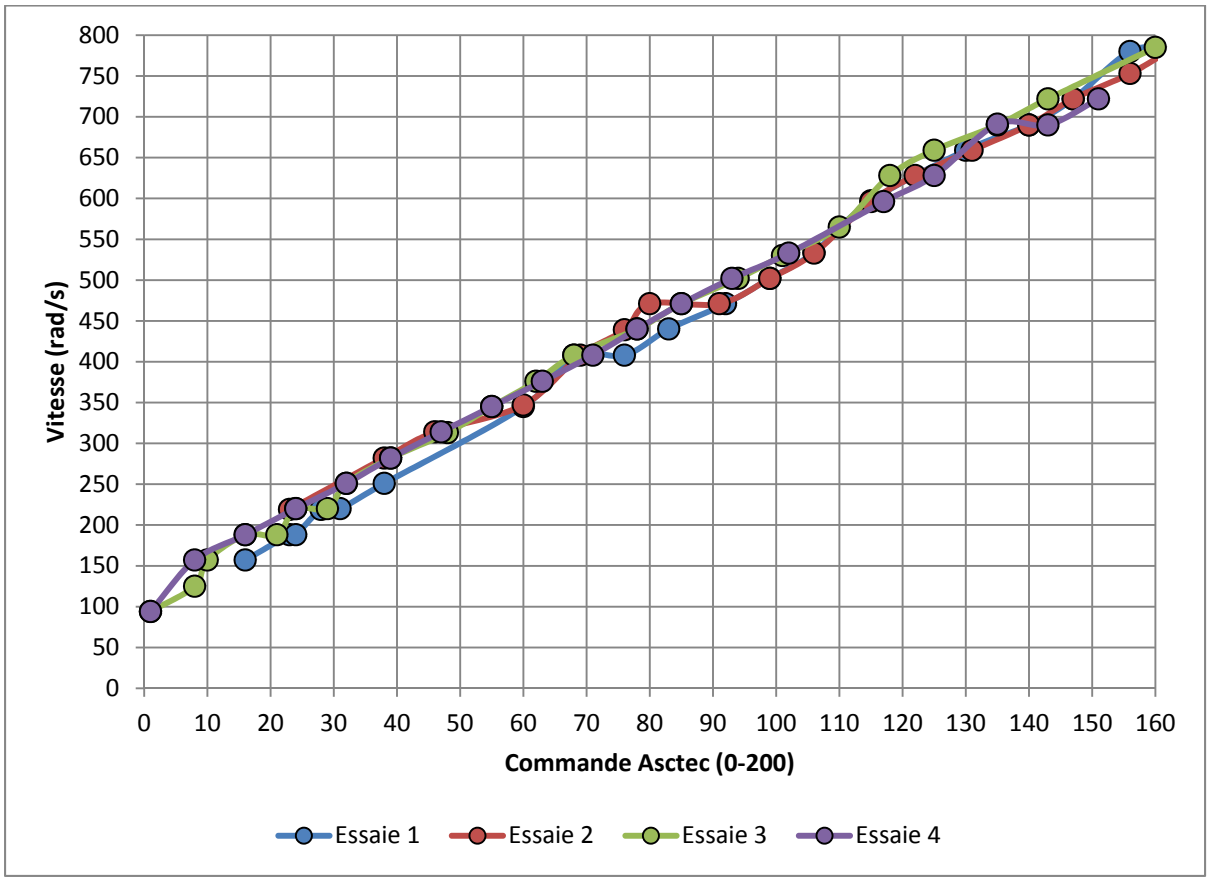



Figure-A IV-1 Mesure de la vitesse de rotation d'un moteur par rapport à la commande

Suite à ses mesures, nous avons utilisé un algorithme de régression linéaire pour déterminer l'équation de correspondance. La Figure A IV-2 présente l'ensemble des régressions.

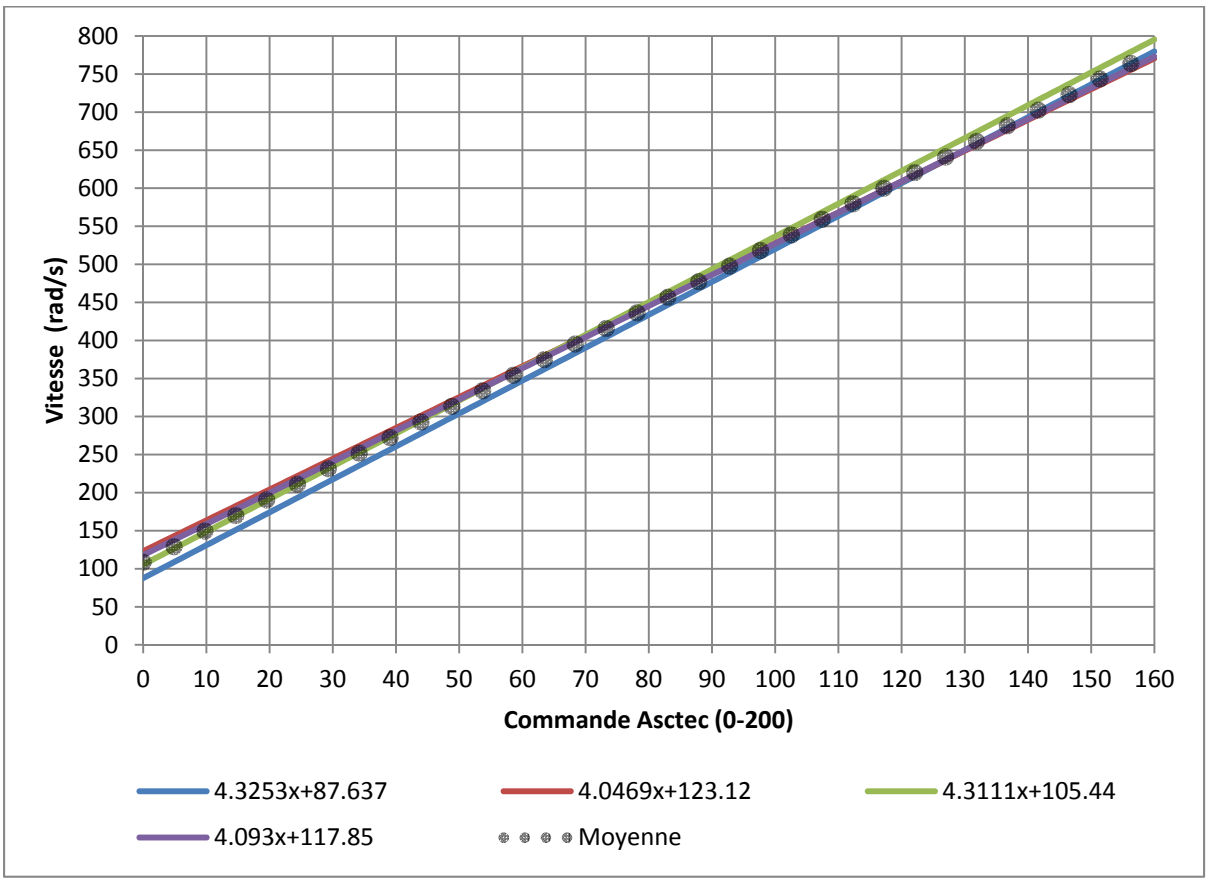



Figure-A IV-2 Corrélation linéaire entre la vitesse d'un moteur et la commande

Nous avons choisi les paramètres de l'équation en faisant la moyenne des paramètres obtenus lors des essais. L'équation finale est la suivante

$$V[rad] = 4.194075 \cdot V[asctec] + 108.5118 \qquad \text{(A IV-1)}$$

# ANNEXE V

## CONVERSION POSITION LLA À NED

L'origine du repère NED utilisé pour effectuer la conversion de la position LLA à NED est fixée à l'aide du signal « /reset_home ». Celui-ci est émis par l'utilisateur via l'interface rqt_gui. Lors de l'activation de ce signal, la node ANS conserve en mémoire la dernière position reçue du Spatial comme étant l'origine du repère NED local et la publie sur le réseau ROS à l'aide du message « /home_location ».

La conversion LLA à NED utilisée est basée sur les équations tirées de (Farrel, 2008). Dans un premier temps, définissons les paramètres permettant de modéliser la Terre comme ellipse, soit $a$ le grand axe principal, $f$ l'aplatissement et $e_2$ l'excentricité carrée dont les valeurs sont les suivantes :

$$a = 6378137 \tag{A V-1}$$

$$f = 0.00335281066474748071984552861852 06 \tag{A V-2}$$

$$e_2 = 2f - f^2 \tag{A V-3}$$

Soit la position de l'origine du repère local en LLA exprimé par le triplet $(\varphi_0 \quad \lambda_0 \quad h_0)$ étant respectivement la latitude, la longitude et l'altitude. Nous commençons la convention en exprimant cette position dans le repère ECEF à l'aide du triplet $(X_0 \quad Y_0 \quad Z_0)$. Le rayon $R_0$ correspond à la valeur du rayon de la Terre à l'origine du repère local.

$$R_0 = a/\sqrt{1 - e_2 \sin^2(\varphi_0)} \tag{A V-4}$$

$$X_0 = (R_0 + h_0) \cdot \cos\varphi_0 \cdot \cos\lambda_0 \tag{A V-5}$$

$$Y_0 = (R_0 + h_0) \cdot \cos\varphi_0 \cdot \sin\lambda_0 \tag{A V-6}$$

$$Z_0 = (R_0 \cdot (1 - e_2) + h_0) \sin\varphi_0 \tag{A V-7}$$

Ces calculs sont effectués qu'une seule fois à la réception d'un nouveau message « /home_location ». Par la suite chaque position LLA reçue exprimée par le triplet $(\varphi \quad \lambda \quad h)$ peut être transformée dans le repère ECEF en utilisant les mêmes équations tel que :

$$R = a/\sqrt{1 - e_2 \sin^2(\varphi)} \tag{A V-8}$$

$$X = (R + h) \cdot \cos\varphi \cdot \cos\lambda \tag{A V-9}$$

$$Y = (R + h) \cdot \cos\varphi \cdot \sin\lambda \tag{A V-10}$$

$$Z = (R \cdot (1 - e_2) + h) \sin\varphi \tag{A V-11}$$

Nous pouvons ensuite la transformer dans le repère NED local ainsi :

$$\begin{bmatrix} x \\ y \\ z \end{bmatrix} = \begin{bmatrix} -s_{\varphi_0} c_{\lambda_0} & -s_{\varphi_0} s_{\lambda_0} & c_{\varphi_0} \\ -s_{\lambda_0} & c_{\lambda_0} & 0 \\ -c_{\varphi_0} c_{\lambda_0} & -c_{\varphi_0} s_{\lambda_0} & -s_{\varphi_0} \end{bmatrix} \begin{bmatrix} X - X_0 \\ Y - Y_0 \\ Z - Z_0 \end{bmatrix} \tag{A V-12}$$

# ANNEXE VI

## CALCUL DE TRAJECTOIRE

Le but du calcul de trajectoire est de construire une courbe lisse commençant par une position initiale $q_0$ et une vitesse initiale $v_0$ et terminant par la position finale désirée $q_f$ et une vitesse finale désirée $v_f$. La trajectoire choisie est de type parabolique divisée en trois parties distinctes, soit une phase d'accélération, une phase d'accélération nulle et une phase de décélération telle que décrite par (Biagiotti et Melchiorri, 2009). Le temps d'accélération est $T_a$ et le temps de décélération est noté $T_d$. On suppose que le temps où commence la trajectoire est toujours égal à zéro. Le temps total pour effectuer la trajectoire est notée par $T_f$. De plus, nous posons que la vitesse finale désirée $v_f$ est nulle pour toutes les trajectoires. La Figure A VI-1 présente un exemple de trajectoire où l'on a identifié les périodes de temps.

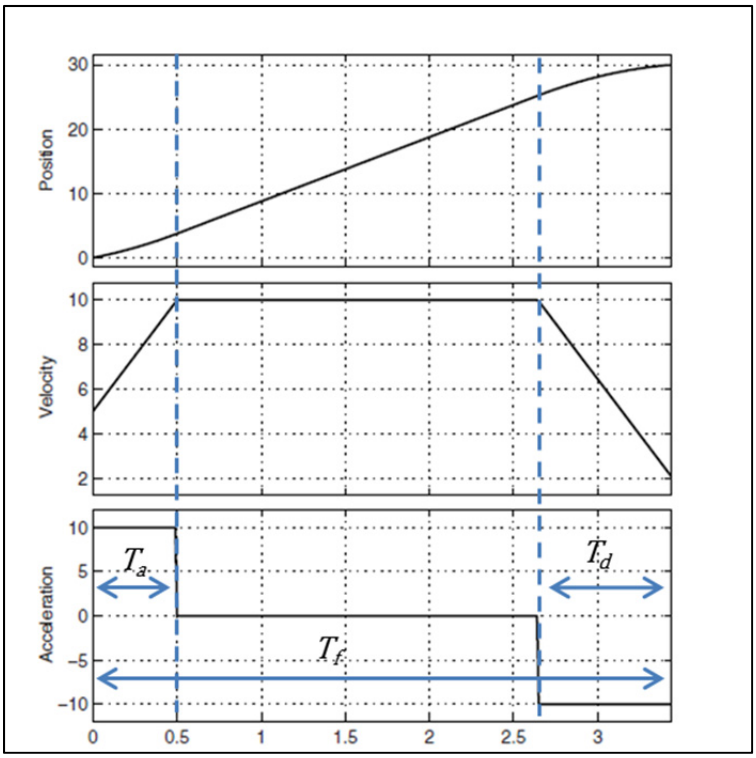



Figure-A VI-1 Profil d'une trajectoire typique

Nous nous basons sur une trajectoire parabolique où la valeur de l'accélération maximale $a_{max}$ et la valeur de la vitesse maximale $v_{max}$ sont spécifiées. À partir de $a_{max}$ et $v_{max}$ nous pouvons définir le déplacement accompli par la trajectoire, notée $h$ [64].

$$h = q_f - q_0 \tag{A VI-1}$$

La trajectoire est réalisable seulement si l'inéquation suivante est respectée :

$$a_{max} h > \frac{{v_0}^2}{2} \tag{A VI-2}$$

Si l'inéquation (A VI-2) n'est pas respectée, alors le déplacement $h$ est trop petit, de telle sorte qu'aucune trajectoire n'est calculée. Si la trajectoire est réalisable, alors il existe deux cas possibles : soit la trajectoire atteint la vitesse $v_{max}$ tel que la trajectoire contient un segment où l'accélération est nulle ou soit la trajectoire n'atteint pas $v_{max}$, mais une vitesse intermédiaire notée $v_{lim}$, tel que $v_{lim} < v_{max}$. Dans ce cas, la trajectoire ne comporte aucun segment où l'accélération est nulle. Si l'inéquation suivante est vraie, alors la vitesse $v_{max}$ est atteinte.

$$a_{max} h > {v_{max}}^2 - \frac{{v_0}^2}{2} \tag{A VI-3}$$

Dans le cas contraire, la vitesse limite peut être calculée ainsi :

---

[64] Pour ce qui suit, nous posons que le déplacement $h$ est positif, tel que $q_f > q_0$. L'ensemble des calculs s'applique également lorsque $h$ est négatif en effectuant quelques modifications au résultat final. Ceci est présenté par la suite.

$$v_{lim} = \sqrt{a_{max}h + \frac{{v_0}^2}{2}} \tag{A VI-4}$$

Pour les deux cas, nous notons $v_v$ comme étant la vitesse maximale atteinte par la trajectoire. Soit le premier cas tel que $v_v = v_{max}$. Les paramètres de la trajectoire peuvent être calculés ainsi :

$$T_a = \frac{v_{max} - v_0}{a_{max}} \tag{A VI-5}$$

$$T_d = \frac{v_{max}}{a_{max}} \tag{A VI-6}$$

$$T_f = \frac{h}{v_{max}} + \frac{v_{max}}{2a_{max}}\left(1 - \frac{v_0}{v_{max}}\right)^2 + \frac{v_{max}}{2a_{max}} \tag{A VI-7}$$

Soit le deuxième cas tel que $v_v = v_{lim}$ Les paramètres de la trajectoire peuvent être calculés ainsi :

$$T_a = \frac{v_{lim} - v_0}{a_{max}} \tag{A VI-8}$$

$$T_d = \frac{v_{lim}}{a_{max}} \tag{A VI-9}$$

$$T_f = T_a + T_d \tag{A VI-10}$$

Les trajectoires $q_d(t)$, $\dot{q}_d(t)$ et $\ddot{q}_d(t)$ peuvent alors être exprimées comme étant :

Phase d'accélération $t \leq T_a$ :

$$\begin{cases} q_d(t) = q_0 + v_0 t + \dfrac{v_v - v_0}{2T_a} t^2 \\ \dot{q}_d(t) = v_0 + \dfrac{v_v - v_0}{T_a} t \\ \ddot{q}_d(t) = a_{max} \end{cases} \quad \text{(A VI-11)}$$

Phase d'accélération nulle $T_a < t \leq T_f - T_d$ :

$$\begin{cases} q_d(t) = q_0 + v_0 t + \dfrac{v_v - v_0}{2T_a} t^2 \\ \dot{q}_d(t) = v_v \\ \ddot{q}_d(t) = 0 \end{cases} \quad \text{(A VI-12)}$$

Phase de décélération $T_f - T_d < t \leq T_f$ :

$$\begin{cases} q_d(t) = q_f - v_f(T_f - t) - \dfrac{v_v}{2T_d}(T_f - t)^2 \\ \dot{q}_d(t) = v_f + \dfrac{v_v}{T_d}(T_f - t) \\ \ddot{q}_d(t) = -a_{max} \end{cases} \quad \text{(A VI-13)}$$

Dans le cas où $h < 0$, il suffit de changer le signe de $q_0$, $v_0$, $q_f$, $q_d(t)$, $\dot{q}_d(t)$ et $\ddot{q}_d(t)$ dans les équations précédentes. On peut donc généraliser les calculs en multipliant ses valeurs par $sign(h)$[65].

[65] Où $sign(h)$ est la fonction signe tel que $h > 0 \rightarrow sign(h) = 1$ $h < 0 \rightarrow sign(h) = -1$.

## ANNEXE VII

## COMPARAISON ENTRE LE CAPTEUR INS/GPS SPATIAL ET LE CAPTEUR INS/GPS ASCTEC

Cette annexe présente une comparaison entre le capteur INS/GPS Spatial et le capteur INS/GPS Asctec. Pour ce faire, nous avons choisi d'effectuer, pour les deux capteurs, le même parcours sur un terrain. Il est important de noter que le terrain choisi constitue un environnement de type « canyon urbain » de tel sorte qu'il n'est pas propice à une bonne réception des signaux GPS[66]. La Figure A VII-1 présente les mesures des deux capteurs après avoir appliqué la transformation LLA à NED telle que décrite par l'ANNEXE V.

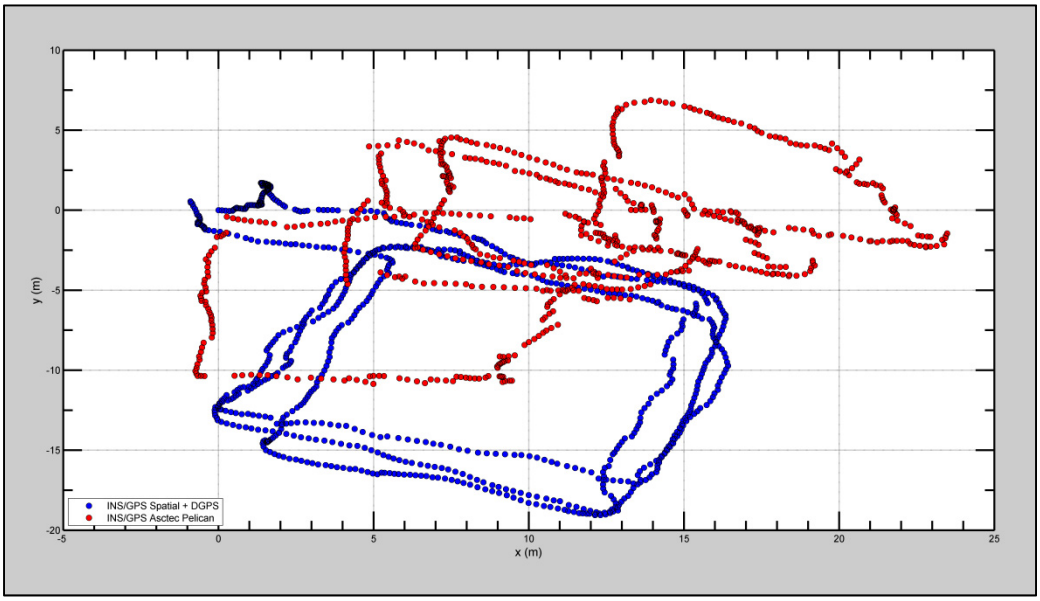



Figure-A VII-1 Comparaison entre le capteur INS/GPS Spatial et le capteur INS/GPS Asctec dans le plan x-y

[66] Pour les deux essais, les receveurs GPS poursuivaient quatre satellites, ce qui constitue le minimum requis. Ceci est sans compter la présence de plusieurs bâtiments, augmentant les perturbations du signal provenant des réflexions.

## LISTE DE RÉFÉRENCES BIBLIOGRAPHIQUES


Arnol'd. 1978. « Mathematical Methods of Classical Mechanics », first edition, number 60 in Graduate Texts in Mathematics, Springet-Verlag, New York-Heidelberg-Berlin.

Austin Reg. 2010. « Unmanned Aircraft Systems : UAV's Design, Development and Deployment ». Aerospace Series. John Wiley & Sons.

Beer, Johnstron. 2005. «Mécanique pour ingénieurs, volume 2 dynamique». Chenelière McGraw-Hill.

Benallegue, A., A. Mokhtari et L. Fridman. 2007. «High-order sliding-mode observer for a quadrotor UAV». In International Journal of Robust and Nonlinear Control, John Wiley & Sons, Ltd.

Biagiotti Luigi, Melchiorri Claudio. 2009. « Trajectory Planning for Automatic Machines and Robots ». Springer.

Bouabdallah, Samir. 2007. «Design and control of quadrotors with application to autonomous flying». Thèse de doctorat no 3727, École Polytechnique Fédérale de Lausanne. 155p.

Bouabdallah, Samir, Noth André, Siegwart Roland. 2004 « PID vs LQ Control Techniques Applied to an Indoor Micro Quadrotor ». In IEEE International Conference on Intelligent Robots and Systems. Sendal, Japan.

Bresciani, Tommaso. 2008. «Modelling, Identification and control of a Quadrotor Helicopter». Mémoire de maîtrise, Lund University.

Brunelle Charles. 2013. « Projet Launch & Forget : Spécification d'installation ». ÉTS.

Brunelle Charles. 2013. « Projet Launch & Forget : Architecture logicielle ». ÉTS.

Chamseddine Abbas, Charland-Arcand Guillaume, Akhrif Ouassima, Gagné Samuel, Gagnon François, Couillard Denis. « Communication Relay for Moving Ground Units with Unkown Positions ». Submitted to the IEEE Transaction on Aerospace and Electronic Systems, 2014.

Charland-Arcand Guillaume, Akhrif Ouassima, Chamseddine Abbas and Gagnon François. « Backstepping Control design applied to a quadrotor aircraft ». Submitted to the ICIUS, 2014.

Craig J. John. 2005. «Introduction to Robotics, Mechanics and Control». 3e edition. Pearson Prentice Hall.

Das, A., K. Subbarao et F. Lewis. 2008. «Dynamic inversion with zero-dynamics stabilisation for quadrotor control». In *IET Control Theory and Applications (Aug. 7, 2008), p.303-314.*

Davila, J. and Fridman, L. and Levant, A. 2005. ”Secondorder sliding-mode observer for mechanical systems”, IEEE Trans. Automat. Cont., 50, pp. 1785–1789, 2005.

DeSantis Romano. 2000. «Elements of Kinematics With Applications to Problems in Navigation and Robotics». Note de cours ELE6209, Département de génie électrique, Polytechnique Montréal.

DeSantis Romano. 2011. «Systèmes de navigation aérienne». 7e édition. Note de cours ELE6209, Département de génie électrique, Polytechnique Montréal.

Dronolab. 2013. « Archives et Images ». *In Dronolab : Aéronef autonome | Club étudiant de l'ÉTS*. En ligne. < http://dronolab.etsmtl.ca/ >. Consulté 15 février 2013.

Farrel., Jay A. 2008 « Aided Navigation, GPS with High Rate Sensors ». The McGraw-Hill.

Fay, Gary. 2001. «Derivation of the Aerodynamic forces for the Mesicopter Simulation».

Filipov, A.F. 1988. « Differential Equations with Discontinuous Righthand Sides ». In Mathematics and Its Applications (Sovier Series), Kluwer Academic Publishers.

Gillula, Jeremy H., Haomiao Huang, Michael P. Vitus et Claire J. Tomlin. 2009. «Design of Guaranteed Safe Maneuvers Using Rachable Sets : Autonomous Quadrotor Aerobatics in Theory and Practice».

Ghommam Jawhar, Guillaume Charland-Arcand and Maarouf Saad. « Three-Dimensional Constrained Tracking Control via Exact Differential Estimator of a Quadrotor Helicopter ». To be published in Asian Journal of Control, AID ASJC951, 2014.

Hamel, Tarek, Rogelio Lozano et James Ostrowski. 2002. «Dynamic modelling and configuration stabilization for an X4-Flyer». In 15th Triennial World Congress, (Barcelona, IFAC 2002).

Huang, Mu, Bin Xian, Chen Diao, Kaiyan Yang et Yu Feng. 2010. «Adaptative Tracking Control of Underactuated Quadrotor Unmanned Aerial Vehicles via Backstepping». In American Control Conference ( Baltimore, June 30 – July 2 2010), p. 2076-2081.

Jay A. Farrell, Marios Polycarpou, Manu Sharma, and Wenjie Dong , 2008 , Command Filtered Backstepping, 2008 American Control Conference Westin Seattle Hotel, Seattle, Washington, USA.

Khalil, K. Hassan. 2002. « NonLinear Systems : Third Edition ». Prentice Hall.

Kim, H. J. and Shim, D. H. 2003. «A flight control system for aerial robots: Algorithms and experiments». Control Engineering Practice, 11(2), 13891400.

Koks Don. 2006. « Explorations In Mathematical Physics, The concepts behind and elegant language». Springer.

Krishnan R. 2010. « Permanent Magnet Synchronous and Brushless DC Motor Drives ». CRC Press.

Kroo Ilan, Shantz Michael, Kunz Peter, Fay Gary, Cheng Shelley, Fabian Tibor, Partridge Chad. 2000. « The Mesicopter : A Miniature Rotorcraft Concept : Phase II Interim Report ». Standford University.

Krstić Miroslav, Kanellakoupoulos Ioannis et Kokotović Petar. 1995. « Nonlinear and Adaptative Control Design ». A volume in the Wiley Series on Adaptive and Learning Systems for Signal Processing, Communications, and Control. John Wiley & Sons.

La Civita M. and Papageorgiou,G. andMessner,W. C. and Kanade, T.2006. «Design and flight testing of an $H_\infty$ controller for a robotic helicopter». Journal of Guidance, Control, and Dynamics, 29(2), 485–494.

Lavoie Philippe. 2012. «Système de navigation hybride GPS/INS à faible coût pour la navigation robuste en environnement urbain». École de technologie supérieure.

Leishman J. Gordon. 2006. «Principles of Helicopter Aerodynamics». Second editions, number 12 in Cambridge Aerospace Series, Cambridge Press.

Levant Arie. 1998. « Robust Exact Differentiation via Sliding Mode Technique ». Automatica Vol.34, No.3. Elsevier Science.

Li Zhou, Qing, Youmin Zhang, Camille-Alain Rabbath et Didier Theilliol. 2010. «Design of Feedback Linearization Control and Reconfigurable Control Allocation with Application to a Quadrotor UAV». In *Conference on Control and Fault Tolerant Systems (Nice, Oct. 6-8 2010), p.371-376.*

Lyapunov Aleksandr. 1992. « The General Problem of the Stability of Motion ». Taylor & Francis.

Madani, Tarek et Abdelaziz Benallegue. 2006.«Backstepping Control for a Quadrotor Helicopter». In *Proc. (IEEE/RSJ) International Conference on Intelligent Robots and Systems (Beijing, Oct. 9-15 2006), p.3255-3260.*

Madani, Tarek et Abdelaziz Benallegue. 2007. «Backstepping Control with Exact 2 Sliding Mode Estimation for a Quadrotor Unmanned Aerial Vehicle». In *Proc. (IEEE/RSJ) International Conference on Intelligence Robots and System (San Diego, Oct. 29 – Nov. 2 2007), p.141-146.*

Madani, Tarek et Abdelaziz Benallegue. 2006. «Control of a Quadrotor Mini-Helicopter via Full State Backstepping Technique». In *Proc. (IEEE) 45th Conference on Decision & Control (San Diego, Dec. 13-15 2006), p.1515-1520.*

Madani, Tarek et Abdelaziz Benallegue. 2007. «Sliding mode observer and backstepping Control for a Quadrotor Unamanned Aerial Vehicles». In *Proc. American Control Conference (New York, July 11-13 2007), p.5887-5892.*

Marcelo De Lellis Costa de Oliveira. 2011. «Modeling, Identification and Control of a Quadrotor Aircraft». Mémoire de maîtrise, Prague, Czech Technical University, 75 p.

Markus Achtelik, Abraham Bachrach, Ruijie He, Samuel Prentice, Nicholas Roy. 2009. « Autonomous Navigation and Exploration of a Quadrotor Helicopter in GPS-denied Indoor Environments ». First Symposium on Indoor Flight Issues. International Aerial Robotics Competition.

Mellinger, Daniel et Vijay Kumar. «Minimum Snap Trajectory Generation and Control for Quadrotors». In *Proc. (IEEE) International conference on Robotics and Automation (ICRA 2011) (May 9-13 2011).*

Murray M. Richard, Li Zexiang et Sastry S. Shankar. 1994. «A Mathematical Introduction to Robotic Manipulation». CRC Press.

M.Hoffmann, Gabriel, Haomiao Huang, Steven L. Waslander et Claire J. Tomlin. 2007. «Quadrotor Helicopter Flight Dynamics and Control: Theory and Experiment». In *AIAA Guidance, Navigation and Control Conference and Exhibit (South Carolina, Aug. 20-23 2007)*: American Institute of Aeronautics and Astronautics.

M.Hoffmann, Gabriel, Haomiao Huang, Steven L. Waslander et Claire J. Tomlin. 2009. «Aerodynamics and Control of Autonomous Quadrotor Helicopters in Aggressive Maneuvering». In *Proc. (IEEE) International conference on Robotics and Automation (ICRA'09) (May 12-17 2009), p. 3277-3282*: IEEE Press Piscataway.

M.Hoffmann, Gabriel, Steven L. Waslander et Claire J. Tomlin. 2004. «Quadrotor Helicopter Trajectory Tracking Control». American Institute of Aeronautics and Astronautics.

Newcome R. Laurence. 2004. « Unmanned Aviation : A Brief History of Unmanned Aerial Vehicles ». AIAA Technology & Engineering.

Orsag Matko, Popopat Marina et Bogdan Stjepan. 2010. « Hybrid fly-by-wire quadrotor controller ». IEEE International Symposium on Industrial Electronics.

Pounds, P., R. Mahony et P. Corke. 2010. «Modelling and control of a large quadrotor robot». In *Control Engineering Practice, p.691-699.*

Pounds, P., R. Mahony, P. Hynes et J. Roberts. 2002. «Design of a Four-Rotor Aerial Robot». In Proc. Australian Conference on Robotics and Automation (Auckland, 27-19 Nov. 2002), p.145-150.

Raptis A. Ioannis, Valavanis Kimon P. 2011. «Linear and Nonlinear Control of Small-Scale Unamanned Helicopters». Springer.

Sadeghzadeh Iman, Mehta Ankit, Chamseddine Abbas et Zhang Youmin. 2012. « Active Fault Tolerant Control of a Quadrotor UAV Based On Gain-Scheduled PID Control ».

Sharifi, Farid, Mostafa Mirzaei, Brandon W. Gordon et Youmin Zhang. 2010. «Fault Tolerant Control of a Quadrotor UAV using Sliding Mode Control». In *Conference on Control and Fault Tolerant Systems (Oct. 6-8 2010), p.239-244.*

Shin J. and Fujiwara and D., Nonami, K., and Hazawa, K. 2005. «Model based optimal attitude and positioning control of small scale unmanned helicopter». Robotica, 23, 51-63.

Spong W. Mark, Hutchinson Seth et Vidyasagar M. 2006. «Robot Modeling and Control». First Edition. John Wiley & Sons.

Valavanis P. Kimon. 2007. « Advances in Unmanned Aerial Vehicles : State of the Art and the Road to Autonomy ». International Series on Intelligent Systems, Control, and Automation : Science and Engineering. University of South Florida, Springer.

Wang Jian, Raffler Thomas, Holzapfel Florian. 2012. « Nonlinear Position Control Approaches for Quadcopters Using a Novel State Representation ». In AIAA Guidance, Navigation and Control Conference, Minneapolis, Minnesota.

Waslander, L. Steven, Gabriel M. Hoffmann, Jung Soon Jang et Claire J. Tomlin. 2005. «Multi-Agent Quadrotor Testbed Control Design : Integral Sliding Mode vs Reinforcement Learning».

Wikipedia. 2014. «List of unmanned aerial vehicles». In *Wikipedia*. En ligne. <http://en.wikipedia.org/wiki/List_of_unmanned_aerial_vehicles>. Consulté le 14 février 2013.

Xu, Rong et Umit Ozguner. 2006. «Sliding Mode Control of a Quadrotor Helicopter». In *Proc. (IEEE) 45th Conference on Decision & Control (San Diego, Dec. 13-15 2006), p.4957-7962.*