



**HAL**  
open science

# Conception et validation d'un robot apte à effectuer des mesures sur ouvrage d'art

Alexis Offermann

► **To cite this version:**

Alexis Offermann. Conception et validation d'un robot apte à effectuer des mesures sur ouvrage d'art. Automatique / Robotique. Université de Technologie de Compiègne, 2021. Français. NNT : 2021COMP2603 . tel-03338024

**HAL Id: tel-03338024**

**<https://theses.hal.science/tel-03338024v1>**

Submitted on 8 Sep 2021

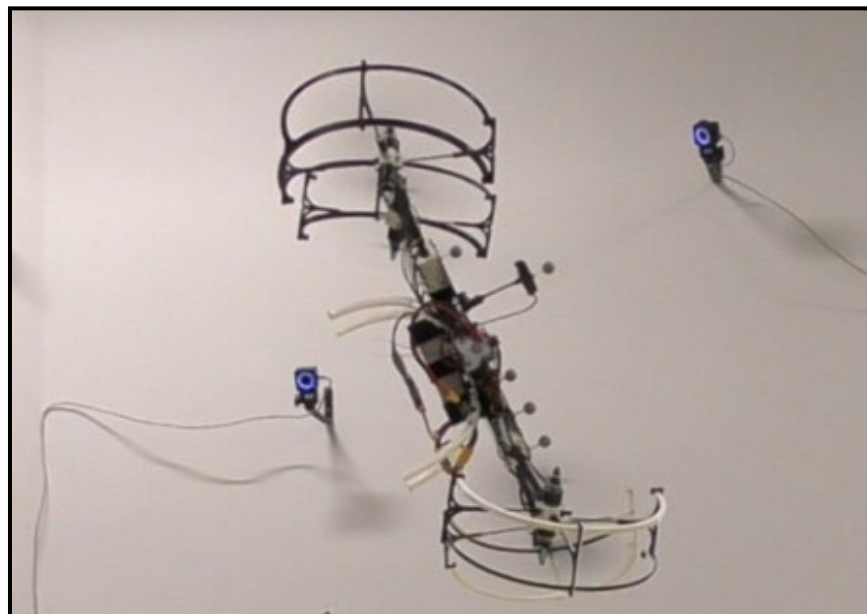
**HAL** is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

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

Par Alexis OFFERMANN

*Conception et validation d'un robot apte à effectuer  
des mesures sur ouvrage d'art*

Thèse présentée  
pour l'obtention du grade  
de Docteur de l'UTC



Soutenue le 30 mars 2021

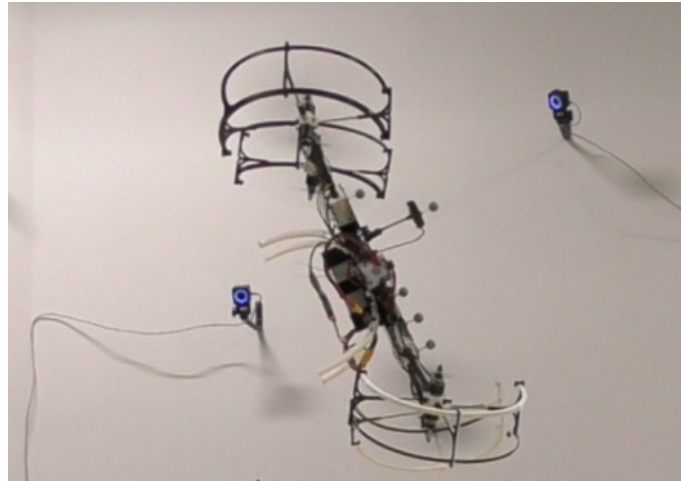
**Spécialité :** Mécatronique, Génie informatique, Automatique et  
Traitement du signal : Unité de recherche Heudysiac (UMR-7253)

D2603

UNIVERSITÉ DE TECHNOLOGIE DE COMPIÈGNE  
LABORATOIRE HEUDIASYC UMR CNRS/UTC 7253  
École doctorale École doctorale de l'UTC  
Unité de recherche Laboratoire Heudiasyc UMR CNRS/UTC 7253  
Thèse présentée par Alexis OFFERMANN  
Soutenue le 30 mars 2021

En vue de l'obtention du grade de docteur de l'Université de Technologie de Compiègne  
Discipline Mécatronique, Génie informatique, automatique et traitement du signal

# Conception et validation d'un robot apte à effectuer des mesures sur ouvrage d'art



---

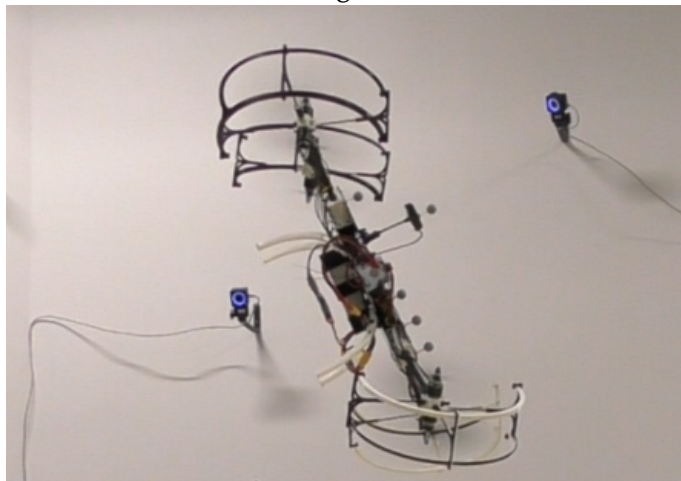
**Thèse dirigée par** Pedro CASTILLO GARCÍA Directeur  
Jérôme DE MIRAS Directeur  
Méchaal RIHANI Co-encadrant

## Composition du jury

|                            |  |   |
|----------------------------|--|---|
| <i>Rapporteurs</i>         | Stéphane VIOLLET<br>Sebastien BRIOT            | directeur de recherche au CNRS<br>chargé de recherche HDR au<br>CNRS                          |
| <i>Examineurs</i>          | Isabelle FANTONI<br>Pascal MORIN<br>Reine TALJ | directrice de recherche au CNRS<br>professeur à l'ISIR<br>chargée de recherche HDR au<br>CNRS |
|                            | Rogelio LOZANO                                 | directeur de recherche au CNRS  |
| <i>Invité</i>              | Stephan KESTELOOT                              | Maître de conférence au Ginger<br>CEBTP   |
| <i>Directeurs de thèse</i> | Pedro CASTILLO GARCÍA<br>Jérôme DE MIRAS       | chargé de recherche HDR au<br>CNRS<br>MCF HDR à l'UTC   |

## COLOPHON

Mémoire de thèse intitulé « Conception et validation d'un robot apte à effectuer des mesures sur ouvrage d'art »



», écrit par Alexis OFFERMANN, achevé le 30 juin 2021, composé au moyen du système de préparation de document  $\text{\LaTeX}$  et de la classe yathesis dédiée aux thèses préparées en France.



**UNIVERSITÉ DE TECHNOLOGIE DE COMPIÈGNE**  
**LABORATOIRE HEUDIASYC UMR CNRS/UTC 7253**

Doctoral School **École doctorale de l'UTC**  
University Department **Laboratoire Heudiasyc UMR CNRS/UTC 7253**

Thesis defended by **Alexis OFFERMANN**

Defended on **March 30, 2021**

In order to become Doctor from Université de Technologie de Compiègne

Academic Field **Mechatronics, Computer Engineering, Automation and Signal Processing**

---

# **Design and validation of a robot able to make measurements on buildings**

---

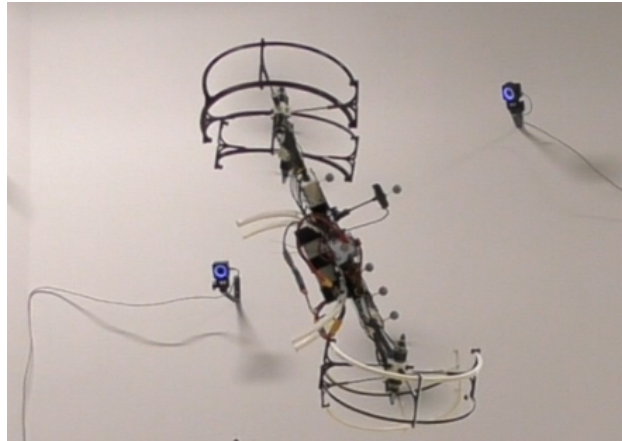
**Thesis supervised by** Pedro CASTILLO GARCÍA Supervisor  
Jérôme DE MIRAS Supervisor  
Méchaal RIHANI Co-supervisor

## **Committee members**

|                    |  |  |
|--------------------|--|--|
| <i>Referees</i>    | Stéphane VIOLLET<br>Sebastien BRIOT                              | Senior Researcher at CNRS<br>HDR Junior Researcher at CNRS   |
| <i>Examiners</i>   | Isabelle FANTONI<br>Pascal MORIN<br>Reine TALJ<br>Rogelio LOZANO | Senior Researcher at CNRS<br>Professor at ISIR<br>HDR Junior Researcher at CNRS<br>Senior Researcher at CNRS |
| <i>Guest</i>       | Stephan KESTELOOT  | Lecturer at Ginger CEBTP   |
| <i>Supervisors</i> | Pedro CASTILLO GARCÍA<br>Jérôme DE MIRAS                         | HDR Junior Researcher at CNRS<br>HDR Associate Professor at UTC  |



## CONCEPTION ET VALIDATION D'UN ROBOT APTE À EFFECTUER DES MESURES SUR OUVRAGE D'ART

**Résumé**

Cette thèse s'inscrit dans le domaine de la robotique pour l'analyse de la pathologie des structures pour le bâtiment et travaux publics. La réalisation de relevés photographiques est d'ores et déjà une technique largement utilisée pour faciliter le diagnostic des ouvrages. On cherche dans cette thèse à développer davantage la technique de relevé en permettant de faire un diagnostic au contact de la structure des ouvrages (allant d'un pont à une église en passant par les hôpitaux et écoles). La présentation de différentes solutions technologiques sera faite et permettra de faire un choix d'innovation technologique à investir. Le choix de solution retenu consiste à développer un drone à bras inclinable capable d'emporter la charge désirée (l'outil de mesure) sur la paroi. Cette étude pose un challenge à la fois sur le modèle mécanique non conventionnel mais également sur la conception d'un robot complexe. Une étude approfondie des lois de commande sera menée, implémentée et comparée dans cet ouvrage. Enfin, une plateforme informatique particulière a également été développée permettant le test rapide des différents algorithmes. Cette plateforme étant extrêmement versatile, elle pourra être réutilisée dans des travaux futurs. Enfin, ces travaux montrent le bon fonctionnement du vecteur drone pour réaliser la tâche qui lui incombe.

**Mots clés :** véhicule volant autonome, modélisation et analyse, système mécatronique complexe, commande non linéaire, conception mécanique, inversion numérique de modèle, expérimentations en temps réel, ros, ardupilot

---

**DESIGN AND VALIDATION OF A ROBOT ABLE TO MAKE MEASUREMENTS ON BUILDINGS****Abstract**

This PhD. applies in the field of robotics for building inspection. Buildings are understood here as huge constructions such as bridges or churches. Using photogrammetry to analyse the structure of these kind of buildings is nowadays a common technique. To extend the analysis methodologies, we propose in this document to imagine a way to bring tools directly in contact with the buildings. Multiple solution possibilities are presented for a final choice. The chosen solution consists of designing a drone able to tilt its arms to bring the building inspection tool directly onto the wall of the construction. This leads to a double challenge. Firstly, the mechanical part of designing such a complex and non-common drone. Secondly, the development and real time testing of the control algorithm. Eventually, this manuscript proposes to use a particular software architecture joining fast experimentations and validation of the control laws. This platform being very versatile, it is also possible to consider re-using it for different applications in future works. The experimentations reveal good results, promoting the usage of the drone as a carrying vector to bring the tools onto the construction.

**Keywords:** drone, uav, modeling and analysis, complex mechatronic system, nonlinear control, mechanical design, numerical model inversion, real-time experimentations, ros, ardupilot

---





# Remerciements

Pour commencer je souhaite remercier l'entreprise Ginger CEBTP ainsi que toutes les personnes qui m'ont aidé. C'est grâce à vous si j'ai pu effectuer cette étude. Un grand merci pour les moyens mis en œuvre permettant la réalisation de ce travail. Merci à tous les collègues directs et indirects de l'entreprise qui m'ont aidé et avec qui j'ai pu passer des moments agréables. Merci également sur un plan humain de m'avoir fait évoluer en quelqu'un de, je pense, plus robuste.

Je souhaite ensuite remercier tout particulièrement mes deux directeurs de thèse Pedro Castillo et Jérôme De Miras. Merci pour votre soutien professionnel. Vous étiez l'un comme l'autre toujours ouvert à la discussion et d'excellent conseil professionnels. J'ai particulièrement apprécié nos débats lors des réunions qui ont finalement énormément contribué à l'avancement. Merci également pour tous les conseils pédagogiques, sociaux, et les rires partagés! Je considère nos moments ensemble comme un privilège et j'en suis très reconnaissant.

Mes remerciements se tournent ensuite directement vers mes collègues du laboratoire qui ont pu m'aider de près ou de loin, ingénieurs de recherche, doctorants et surtout voisins de bureau. Merci pour l'aide mais également pour tous les bon moments passés en- et en-dehors du travail. Merci également de m'avoir fait goûter à votre culture provenant le plus souvent de l'autre coté de l'atlantique.

Pour continuer, je souhaite remercier tous les membres de ma famille, française et allemande, qui m'ont toujours soutenu dans mes travaux; en particulier Laura que j'admire par sa capacité à m'avoir supporté, écouté et aidé moralement dans les moments difficiles mais avec qui j'ai également adoré partager les bon moments. Merci également à vous tous, qui m'avez relu et corrigé tellement de fois. Une immense gratitude directement à mes deux parents qui m'ont tous les deux encouragé à effectuer ce travail mais également soutenu à de nombreux égards. Nochmal herzlichen Dank!

Et enfin et non des moindres, je souhaite remercier tous mes amis. Toutes ces relations que l'on a pu créer durant mon passage dans les Hauts-de-France mais également celles liées à l'Alsace. Vous comptez énormément pour moi et j'espère qu'à l'avenir nos chemin continueront à s'enchevêtrer comme ils le font, y compris quand je relirai ces phrases dans quelques années.



# Table des matières

|  |             |
|--|-------------|
| <b>Résumé</b>  | <b>ix</b>   |
| <b>Remerciements</b>   | <b>xi</b>   |
| <b>Table des matières</b>  | <b>xiii</b> |
| <b>Liste des tableaux</b>  | <b>xvii</b> |
| <b>Table des figures</b>   | <b>xix</b>  |
| <b>1 Avant-propos</b>  | <b>1</b>    |
| 1.1 Objectif et problématiques . . . . .                         | 2           |
| 1.2 Organisation du manuscrit . . . . .                          | 2           |
| <b>2 Contexte de l'étude</b>                                     | <b>5</b>    |
| 2.1 Présentation de l'entreprise . . . . .                       | 5           |
| 2.2 Motivations de l'entreprise . . . . .                        | 7           |
| 2.3 Cahier des charges . . . . .                                 | 9           |
| 2.4 État de l'art . . . . .                                      | 11          |
| 2.4.1 Les drones classiques . . . . .                            | 11          |
| 2.4.2 Drones avec configurations de vols particulières . . . . . | 13          |
| 2.4.3 Approches possibles et choix de solution . . . . .         | 16          |
| 2.5 Matériel et méthode . . . . .                                | 18          |
| 2.5.1 Présentation du laboratoire . . . . .                      | 18          |
| 2.5.2 Gestion du projet . . . . .                                | 19          |
| 2.5.3 Modélisation . . . . .                                     | 20          |
| 2.5.4 Loi de commande . . . . .                                  | 20          |
| 2.5.5 Plateforme . . . . .                                       | 20          |
| 2.5.5.1 Conception mécanique . . . . .                           | 20          |
| 2.5.5.2 Système embarqué . . . . .                               | 21          |
| 2.5.6 Résultats . . . . .  | 21          |
| 2.6 Conclusion . . . . .   | 22          |
| <b>3 Modélisation</b>  | <b>23</b>   |
| 3.1 PVTOL et PVTOL à bras inclinables . . . . .                  | 23          |
| 3.1.1 Modélisation mécanique PVTOL-MI . . . . .                  | 24          |
| 3.2 Définitions et rappels de modélisation mécanique . . . . .   | 29          |
| 3.2.1 Modélisation configuration classique X4 . . . . .          | 29          |
| 3.2.1.1 Définitions . . . . .                                    | 29          |
| 3.2.1.2 Dynamique du système . . . . .                           | 31          |
| 3.2.2 Repères et variables pour des bras inclinables . . . . .   | 34          |
| 3.3 Quadricoptère + . . . . .                                    | 37          |
| 3.3.1 Forces . . . . .   | 38          |

|          |   |           |
|----------|---|-----------|
| 3.3.2    | Moments . . . . .   | 39        |
| 3.3.3    | Synthèse . . . . .  | 41        |
| 3.4      | Quadricoptère X . . . . .   | 42        |
| 3.4.1    | Forces . . . . .  | 43        |
| 3.4.2    | Moments . . . . .   | 44        |
| 3.4.3    | Synthèse . . . . .  | 45        |
| 3.5      | Quadricoptère $\tilde{X}$ . . . . .   | 46        |
| 3.5.1    | Forces . . . . .  | 46        |
| 3.5.2    | Moments . . . . .   | 47        |
| 3.5.3    | Synthèse . . . . .  | 48        |
| 3.6      | Énergie . . . . .   | 49        |
| 3.7      | Conclusion . . . . .  | 51        |
| <b>4</b> | <b>Commande</b> . . . . .   | <b>53</b> |
| 4.1      | Algorithme de commande pour le PVTOL-MI . . . . .                             | 53        |
| 4.1.1    | Linéarisation exacte d'un système . . . . .                                   | 54        |
| 4.1.1.1  | Généralités . . . . .   | 54        |
| 4.1.1.2  | Application au système . . . . .  | 55        |
| 4.1.2    | Loi de commande linéaire . . . . .  | 56        |
| 4.1.3    | Algorithme de saturation emboîtés . . . . .                                   | 58        |
| 4.1.4    | Des entrées de commande virtuelles aux actionneurs . . . . .                  | 59        |
| 4.2      | Généralisation de la commande aux 3 dimensions . . . . .                      | 60        |
| 4.2.1    | Stabilisation du système . . . . .  | 61        |
| 4.2.1.1  | Accélérations linéaires . . . . .   | 61        |
| 4.2.1.2  | Accélérations angulaires . . . . .  | 62        |
| 4.2.1.3  | Calcul de l'erreur d'orientation . . . . .                                    | 62        |
| 4.2.2    | Mesure et observation des états . . . . .                                     | 63        |
| 4.2.2.1  | Vitesse linéaire . . . . .  | 63        |
| 4.2.2.2  | Orientation . . . . .   | 65        |
| 4.2.3    | Linéarisation du système . . . . .  | 66        |
| 4.2.4    | Contrôleur linéaire . . . . .   | 68        |
| 4.2.5    | Contrôleur Robust SMCST . . . . .   | 68        |
| 4.3      | Singularités . . . . .  | 70        |
| 4.4      | Balourd . . . . .   | 73        |
| 4.5      | Conclusion . . . . .  | 74        |
| <b>5</b> | <b>Plateforme</b> . . . . .   | <b>75</b> |
| 5.1      | Plateforme mécanique . . . . .  | 75        |
| 5.1.1    | Développements préliminaires . . . . .  | 76        |
| 5.1.1.1  | Conception d'un banc d'essai pour la mesure de $k_f$ et $k_\tau$ . . . . .    | 76        |
| 5.1.1.2  | Conception d'un banc d'essai pour le réglage des gains du drone . . . . .     | 81        |
| 5.1.1.3  | Rotation continue des servo-moteurs . . . . .                                 | 82        |
| 5.1.1.4  | Numérisation des outils de mesures . . . . .                                  | 83        |
| 5.1.2    | Conception pour les prototypes . . . . .                                      | 84        |
| 5.1.2.1  | Amortissement pour le Navio . . . . .   | 85        |
| 5.1.2.2  | Plaques pour la deuxième version du prototype . . . . .                       | 86        |
| 5.1.2.3  | Support pour les moteurs brushless . . . . .                                  | 87        |
| 5.1.2.4  | Dispositif de protection des biens et des personnes . . . . .                 | 88        |
| 5.1.2.5  | Optimisation de l'accouplement entre servomoteur et tube en carbone . . . . . | 89        |
| 5.1.2.6  | Support pour les servomoteurs . . . . .                                       | 90        |

|          |   |            |
|----------|---|------------|
| 5.1.2.7  | Support pour les batteries . . . . .  | 91         |
| 5.1.3    | Résultats . . . . .   | 92         |
| 5.2      | Plateforme informatique . . . . .   | 95         |
| 5.2.1    | Simulateur . . . . .  | 95         |
| 5.2.2    | Système embarqué . . . . .  | 97         |
| 5.2.2.1  | Choix de l'architecture logicielle/matérielle . . . . .                           | 97         |
| 5.2.2.2  | Première architecture, un hybride . . . . .                                       | 99         |
| 5.2.2.3  | Deuxième architecture, ROS sans Ardupilot . . . . .                               | 101        |
| 5.2.2.4  | Troisième architecture, le retour de l'hybridation . . . . .                      | 103        |
| 5.2.3    | Station sol . . . . .   | 103        |
| 5.2.4    | Système d'OptiTrack . . . . .   | 104        |
| 5.2.4.1  | Dynamixels et ROS . . . . .   | 105        |
| 5.2.5    | Conclusion . . . . .  | 107        |
| <b>6</b> | <b>Résultats de simulation et essais réels</b>                                    | <b>109</b> |
| 6.1      | PVTOL-MI résultats de simulation . . . . .  | 109        |
| 6.1.1    | Consigne proche de la position initiale . . . . .                                 | 111        |
| 6.1.2    | Consigne éloignée de la position initiale . . . . .                               | 113        |
| 6.1.3    | Discussion . . . . .  | 114        |
| 6.1.4    | Conclusions . . . . .   | 114        |
| 6.2      | Quadcopter en X . . . . .   | 114        |
| 6.2.1    | Cheminement des essais . . . . .  | 115        |
| 6.2.2    | 4 Degrés de liberté (DDL) bras inclinables . . . . .                              | 117        |
| 6.2.3    | 6 DDL bras inclinables . . . . .  | 119        |
| 6.2.4    | Suivi de carré . . . . .  | 121        |
| 6.2.4.1  | Simulation . . . . .  | 121        |
| 6.2.4.2  | Expérimentations . . . . .  | 123        |
| 6.2.5    | Inclinaison du corps . . . . .  | 125        |
| 6.2.5.1  | Simulation . . . . .  | 125        |
| 6.2.5.2  | Expérimentations Proportionnel Dérivé (PD) . . . . .                              | 127        |
| 6.2.5.3  | Expérimentations Sliding Mode Control Super Twisting<br>(SMCST) . . . . .         | 129        |
| 6.2.6    | Suivi de cercle et inclinaison . . . . .  | 130        |
| 6.2.6.1  | Simulation . . . . .  | 130        |
| 6.2.6.2  | Réalité - Contrôleur linéaire . . . . .   | 131        |
| 6.2.7    | Réalité - Contrôleur SMCST . . . . .  | 134        |
| 6.2.8    | Compensation gyroscopique . . . . .   | 135        |
| 6.2.8.1  | Simulation . . . . .  | 135        |
| 6.2.8.2  | Expérimentations . . . . .  | 136        |
| 6.3      | Configuration $\checkmark$ . . . . .  | 136        |
| 6.3.1    | Simulation . . . . .  | 137        |
| 6.3.2    | Expérimentations PD . . . . .   | 139        |
| 6.3.3    | Expérimentations PD compensant le balourd . . . . .                               | 141        |
| 6.3.4    | Expérimentations <i>Sliding Mode Super Twisting</i> (SMST) avec l'outil . . . . . | 142        |
| 6.4      | Conclusion . . . . .  | 143        |
| <b>7</b> | <b>Conclusion générale et Perspectives</b>  | <b>145</b> |
| 7.1      | Conclusion générale . . . . .   | 145        |
| 7.2      | Perspective . . . . .   | 147        |
|          | <b>Bibliographie</b>  | <b>149</b> |
|          | Liste de Publications . . . . .   | 149        |
|          | Références . . . . .  | 149        |

|   |            |
|---|------------|
| <b>Annexes</b>  | <b>160</b> |
| <b>Annexes</b>  | <b>160</b> |
| <b>A Formation, diffusion et travaux annexes</b>  | <b>161</b> |
| A.1 Formations . . . . .  | 161        |
| A.2 Diffusion scientifique et salons . . . . .  | 161        |
| <b>B Analyse des modes de défaillance, de leurs effets et de leur criticité (AMDEC)</b> | <b>163</b> |
| B.1 Criticité . . . . .   | 163        |
| B.2 Structure . . . . .   | 164        |
| B.3 Système embarqué . . . . .  | 165        |
| <b>C Précis mathématiques</b>   | <b>169</b> |
| C.1 Fonctions usuelles . . . . .  | 169        |
| C.1.1 Fonction de saturation . . . . .  | 169        |
| C.1.2 Fonction signe . . . . .  | 169        |
| C.1.3 Pseudoinverse de Moore-Penrose . . . . .  | 169        |
| C.2 Preuves de stabilité . . . . .  | 170        |
| C.2.1 Système linéaire . . . . .  | 170        |
| C.2.2 Saturations imbriquées . . . . .  | 170        |
| C.2.3 Sliding mode super-twisting . . . . .   | 172        |
| <b>D Plateforme</b>   | <b>174</b> |
| D.1 Mesure de l'angle de rotation des servomoteurs . . . . .                            | 174        |
| D.1.1 Boîtier Arduino . . . . .   | 176        |
| D.1.2 Accouplement entre bras et servomoteurs . . . . .                                 | 176        |
| D.2 Estimation de l'altitude . . . . .  | 177        |
| D.3 Précisions d'utilisation . . . . .  | 178        |
| D.3.1 Installation du framework développé . . . . .                                     | 178        |
| D.3.2 OptiTrack . . . . .   | 179        |
| D.3.3 Dynamixels . . . . .  | 179        |
| D.3.4 Problèmes et solutions rencontrés au cours de l'utilisation . . . . .             | 180        |

# Liste des tableaux

- 2.1 Fonction principale et contraintes du projet . . . . . 9
- 2.2 Classification des solutions technologiques . . . . . 10
- 2.3 Classification des solutions technologiques . . . . . 17
- 2.4 Échéance des tâches à réaliser au cours de la thèse . . . . . 19
- 2.5 Échéance des tâches à réaliser au cours de la thèse . . . . . 19
  
- 3.1 Définitions et noms des paramètres . . . . . 25
- 3.2 Définition des variables . . . . . 30
- 3.3 Définition des variables . . . . . 36
- 3.4 Définition des moments présents dans le drone . . . . . 37
- 3.5 Consommation énergétique dans 3 scénarii différents . . . . . 50
  
- B.1 Matrices de criticité Sécurité . . . . . 163
- B.2 Matrices de criticité disponibilité . . . . . 164
- B.3 Fonction principale et contraintes du projet . . . . . 164
- B.4 Fonction principale et contraintes de l'AMDEC système embarqué . . . . . 165





# Table des figures

|      |  |    |
|------|--|----|
| 2.1  | Château d'eau d'Auchy les mines . . . . .  | 6  |
| 2.2  | Prototypes pour prises de vues et relevés de Ginger Modeliris . . . . .  | 7  |
| 2.3  | Différents essais et appareils de mesure de la pathologie des structures . . . . .   | 8  |
| 2.4  | Exemple de nacelle inversée . . . . .  | 8  |
| 2.5  | Illustration du problème d'accès aux châteaux d'eau . . . . .  | 9  |
| 2.6  | Différents types d'utilisation des drones . . . . .  | 12 |
| 2.7  | Les trois axes de rotation du drone . . . . .  | 13 |
| 2.8  | Drone transformable . . . . .  | 13 |
| 2.9  | Avion convertible . . . . .  | 14 |
| 2.10 | L'omnicopter de l'ETH de Zurich . . . . .  | 14 |
| 2.11 | Bras manipulateurs volants . . . . .   | 15 |
| 2.12 | Prototypes pour manipulations aériennes . . . . .  | 15 |
| 2.13 | Drones commercialisés avec structure particulière . . . . .  | 16 |
| 2.14 | Projet Voliro . . . . .  | 16 |
| 2.15 | Démonstration avec une flotte de drones dans la volière . . . . .  | 18 |
| 3.1  | Représentation du PVTOL classique . . . . .  | 24 |
| 3.2  | Schéma du PVTOL à moteurs inclinables . . . . .  | 24 |
| 3.3  | Schéma des forces pour le PVTOL-MI . . . . .   | 25 |
| 3.4  | Schéma des distances dans le PVTOL-MI . . . . .  | 26 |
| 3.5  | Schéma des moments dans le PVTOL-MI . . . . .  | 27 |
| 3.6  | Représentation de l'effet des entrées virtuelles du PVTOL-MI . . . . .   | 28 |
| 3.7  | Définition des repères dans le système . . . . .   | 29 |
| 3.8  | Diagramme des changements de repères avec les angles d'Euler . . . . .   | 30 |
| 3.9  | Sens de rotations positifs dans les mesures angulaires . . . . .   | 32 |
| 3.10 | Différents types de configuration pour les drones à bras inclinables . . . . .   | 35 |
| 3.11 | Définition du repère lié à l'hélice et de l'angle $\beta$ . . . . .  | 36 |
| 3.12 | Quadricoptère dans une configuration en + . . . . .  | 37 |
| 3.13 | Schéma des forces et distances au niveau des hélices . . . . .   | 40 |
| 3.14 | Quadricoptère dans une configuration en $\times$ . . . . .   | 42 |
| 3.15 | Quadricoptère dans une configuration en $\text{X}$ . . . . .   | 46 |
| 3.16 | Parcours du drone pour comparaison énergétique dans 3 scénarii différents . . . . .  | 50 |
| 4.1  | Structure de commande d'un quadricoptère . . . . .   | 60 |
| 4.2  | Structure de commande d'un quadricoptère à bras inclinables . . . . .  | 64 |
| 4.3  | Structure de commande d'un quadricoptère à bras inclinables . . . . .  | 68 |
| 4.4  | Graphique de surface pour un contrôleur mode glissant . . . . .  | 69 |
| 4.5  | Explication de la représentation des singularités. En rouge, une absence de force, en blanc une force maximale et en jaune une force intermédiaire . . . . . | 71 |

|      |  |    |
|------|--|----|
| 4.6  | Singularités pour le drone en configuration "+" pour le premier bras. En rouge (gauche) et bleu foncé (droite), une absences de force; en blanc (gauche) et jaune (à droite) une force maximale et intermédiaire en jaune (gauche) et vert (droite) . . . . .      | 71 |
| 4.7  | Singularités pour le drone en configuration $\times$ pour le premier bras. En rouge (gauche) et bleu foncé (droite), une absences de force; en blanc (gauche) et jaune (à droite) une force maximale et intermédiaire en jaune (gauche) et vert (droite) . . . . . | 72 |
| 4.8  | Singularités pour la configuration $\checkmark$ pour le premier bras. En rouge (gauche) et bleu foncé (droite), une absences de force; en blanc (gauche) et jaune (à droite) une force maximale et intermédiaire en jaune (gauche) et vert (droite) . . . . .      | 72 |
| 4.9  | Régulation du balourd sur un disque en simulation . . . . .  | 74 |
| 5.1  | Support du banc d'essais . . . . .   | 77 |
| 5.2  | Embout pour le banc d'essai . . . . .  | 77 |
| 5.3  | Socle du banc d'essais . . . . .   | 78 |
| 5.4  | Bancs de tests pour estimer les constantes $k_f$ et $k_\tau$ . . . . .   | 78 |
| 5.5  | Estimation des constantes $k_f$ et $k_\tau$ . . . . .  | 79 |
| 5.6  | Vitesse de rotation des hélices en fonction de la <i>Pulse Width Modulation</i> (Modulation de largeur d'impulsion) (PWM) . . . . .  | 80 |
| 5.7  | Estimation de la constante $k_f$ . . . . .   | 80 |
| 5.8  | Estimation de la constante $k_\tau$ . . . . .  | 81 |
| 5.9  | Banc d'essai pour le réglage des gains - drone $\checkmark$ . . . . .  | 81 |
| 5.10 | Photo d'un servomoteur Dynamixel de Robotis™ . . . . .   | 82 |
| 5.11 | Procédure de scan de l'outil de mesure . . . . .   | 83 |
| 5.12 | Modélisation CAO, reproduction et véritable ferroskan . . . . .  | 83 |
| 5.13 | Modèle CAO du porte outils . . . . .   | 84 |
| 5.14 | Schéma et disposition des composants du système . . . . .  | 84 |
| 5.15 | Amortissement du Navio première version (gauche) et deuxième version (droite) . . . . .  | 85 |
| 5.16 | Amortissement du Navio2 deuxième version . . . . .   | 86 |
| 5.17 | Plaque en carbone de la 2ème version du prototype . . . . .  | 86 |
| 5.18 | Support de moteur pour le drone $\times$ . . . . .   | 87 |
| 5.19 | Support de moteur contra-rotatif final . . . . .   | 87 |
| 5.20 | Première version du bras du drone . . . . .  | 88 |
| 5.21 | Embout du protège bras . . . . .   | 89 |
| 5.22 | Embout du protège bras pour le drone $\checkmark$ . . . . .  | 89 |
| 5.23 | Extrémité du bras pour le drone $\times$ . . . . .   | 90 |
| 5.24 | Porte-servo et embout pour tube en carbone . . . . .   | 90 |
| 5.25 | Vue éclatée du bras inclinable coté servomoteur . . . . .  | 91 |
| 5.26 | Version finale du porte-servo accouplé au servomoteur Dynamixel . . . . .  | 91 |
| 5.27 | Extrémité du bras pour le drone $\times$ . . . . .   | 92 |
| 5.28 | Premier bras inclinable . . . . .  | 92 |
| 5.29 | Première version de <b>Robot aptes aux Mesures sur Ouvrages (RoMEO)</b> - CAO . . . . .  | 93 |
| 5.30 | Première version de RoMEO - Réel . . . . .   | 93 |
| 5.31 | 2ème version du prototype - CAO . . . . .  | 94 |
| 5.32 | 2ème version du prototype - Réel . . . . .   | 94 |
| 5.33 | Process de création du simulateur . . . . .  | 96 |
| 5.34 | BeagleBone® Blue . . . . .   | 98 |
| 5.35 | Drone DJI F450 équipé d'un Pixhawk . . . . .   | 98 |

|      |   |     |
|------|---|-----|
| 5.36 | Raspberry pi équipé d'un Navio 2 . . . . .  | 99  |
| 5.37 | Schéma de l'architecture informatique . . . . .   | 101 |
| 5.38 | Schéma de la deuxième architecture . . . . .  | 102 |
| 5.39 | Drone racer en vol . . . . .  | 103 |
| 5.40 | Station sol réalisée dans Matlab Simulink . . . . .   | 104 |
| 5.41 | Modèle du servomoteur en simulation . . . . .   | 106 |
| 5.42 | Comparaison des réponses des servomoteurs en réel et en simulation . . . . .  | 106 |
| 6.1  | Modèle Simulink du PVTOL-MI . . . . .   | 110 |
| 6.2  | Superposition d'images de la dynamique de vol du PVTOL-MI . . . . .   | 111 |
| 6.3  | PVTOL-MI : états du système pour une consigne proche. En bleu les états avec un contrôleur linéaire, en noir avec la commande par saturations et en rouge la référence. . . . .                                 | 112 |
| 6.4  | PVTOL-MI : comportement des actionneurs pour une consigne proche. En bleu les entrées des actionneurs avec un contrôleur linéaire, en noir avec la commande par saturations et en rouge la référence. . . . .   | 112 |
| 6.5  | PVTOL-MI : états du système pour une consigne éloignée. En bleu les états avec un contrôleur linéaire, en noir avec la commande par saturations et en rouge la référence. . . . .                               | 113 |
| 6.6  | PVTOL-MI : comportement des actionneurs pour une consigne éloignée. En bleu les entrées des actionneurs avec un contrôleur linéaire, en noir avec la commande par saturations et en rouge la référence. . . . . | 113 |
| 6.7  | Entrées de commandes du système en forme de $\times$ pour le deuxième scénario de vol. En noir la mesure d'état; en bleu le mode 4DDL; en rouge la référence. . . . .   | 117 |
| 6.8  | États du système en forme de $\times$ pour le deuxième scénario de vol. En noir la mesure d'état; en bleu le mode 4DDL. . . . .   | 117 |
| 6.9  | États des actionneurs pour le système en forme de $\times$ pour le deuxième scénario de vol. En noir l'entrée des actionneurs; en bleu le mode 4DDL. . . . .  | 118 |
| 6.10 | Entrées de commandes virtuelles du système en forme de $\times$ pour le deuxième scénario de vol. En noir l'entrée de commande; en bleu le mode 4DDL. . . . .   | 118 |
| 6.11 | États du système en forme de $\times$ pour le deuxième scénario de vol. En noir la mesure des états; en bleu le mode 4DDL; en gris le mode 6DDL. . . . .  | 119 |
| 6.12 | Entrées de commandes du système en forme de $\times$ pour le deuxième scénario de vol. En noir la mesure d'états; en bleu le mode 4DDL; en gris le mode 6DDL; en rouge la référence. . . . .                    | 119 |
| 6.13 | États des actionneurs pour le système en forme de $\times$ pour le deuxième scénario de vol. En noir l'entrée des actionneurs; en bleu le mode 4ddl; en gris le mode 6ddl. . . . .                              | 120 |
| 6.14 | Entrées de commandes du système en forme de $\times$ pour le deuxième scénario de vol. En noir l'entrée de commande; en bleu le mode 4DDL; en gris le mode 6DDL. . . . .  | 120 |
| 6.15 | Trajectoire du drone en simulation pour un suivi de 4 points de cheminement en forme de carré (3ème scénario de vol). En noir la mesure des états; en rouge les cercles de références. . . . .                  | 121 |
| 6.16 | États du système en simulation pour le suivi en trajectoire du carré. En noir la mesure des états; en rouge la référence. . . . .   | 121 |
| 6.17 | Entrée de commande du système en simulation pour le suivi en trajectoire du carré . . . . .   | 122 |
| 6.18 | États des actionneurs, en simulation pour le suivi en trajectoire du carré . . . . .  | 122 |

|      |   |     |
|------|---|-----|
| 6.19 | Position du drone durant les expérimentations pour un suivi de trajectoire en forme de carré. En noir la mesure des états; en bleu le mode manuel, le reste est en autonome; en rouge les cercles de références. . . . .            | 123 |
| 6.20 | États du drone durant les expérimentations pour un suivi de trajectoire en forme de carré. En noir la mesure des états; en bleu le mode manuel, le reste est en autonome; en rouge les cercles de références. . . . .               | 123 |
| 6.21 | États des actionneurs lors du suivi de trajectoire en forme de carré. En noir les entrées des actionneurs; en bleu le mode manuel, le reste est en autonome. . . . .  | 124 |
| 6.22 | États du système en forme de $\times$ pendant la simulation de l'inclinaison. En noir la mesure des états; en rouge les références. . . . .   | 125 |
| 6.23 | Dynamique de vol de RoMeO en simulation . . . . .   | 125 |
| 6.24 | États des actionneurs pour le système en forme de $\times$ pendant la simulation de l'inclinaison . . . . .   | 126 |
| 6.25 | Entrées de commandes du système en forme de $\times$ pendant la simulation de l'inclinaison . . . . .   | 126 |
| 6.26 | États du système en forme de $\times$ pendant les expérimentations. En noir la mesure des états; en bleu le mode manuel, le reste représente le vol incliné. . . . .  | 127 |
| 6.27 | États du système en forme de $\times$ pendant les expérimentations. En noir la mesure des états; en bleu le mode manuel, en gris le mode incliné. . . . .   | 127 |
| 6.28 | États des actionneurs pour le système en forme de $\times$ pendant les expérimentations. En noir l'entrée des actionneurs; en bleu le mode 4DDL, en gris le mode incliné 6 DDL. . . . .   | 128 |
| 6.29 | Entrées de commandes du système en forme de $\times$ pendant les expérimentations. En noir l'entrée de commande; en bleu le mode 4 DDL, en gris le mode incliné 6 DDL; en rouge les références. . . . .                             | 128 |
| 6.30 | États du système en forme de $\times$ pendant la simulation. En noir la mesure des états; en bleu le mode manuel, le reste représente le vol incliné. . . . .   | 129 |
| 6.31 | États du système en forme de $\times$ pendant la simulation. En noir la mesure des états; en bleu le mode 4 DDL, le reste représente le mode incliné 6 DDL; en rouge les références. . . . .  | 129 |
| 6.32 | États du système en forme de $\times$ pour le deuxième scénario de vol. En noir la mesure des états; en rouge la référence. . . . .   | 130 |
| 6.33 | États des actionneurs pour le système en forme de $\times$ pour le scénario de vol en cercle . . . . .  | 131 |
| 6.34 | Entrées de commandes du système en forme de $\times$ pour le deuxième scénario de vol . . . . .   | 131 |
| 6.35 | États du système en forme de $\times$ pour le deuxième scénario de vol. En noir la mesure des états; en bleu le mode 4 DDL, en gris le mode autonome. . . . .   | 132 |
| 6.36 | États du système en forme de $\times$ pour le deuxième scénario de vol. En noir la mesure des états; en bleu le mode 4DDL; en gris le mode 6DDL initialisation; sans couleur le mode autonome; les références en rouge. . . . .     | 132 |
| 6.37 | États des actionneurs pour le système en forme de $\times$ pour le deuxième scénario de vol. En noir les entrées des actionneurs; en bleu le mode 4DDL; en gris le mode 6DDL initialisation; sans couleur le mode autonome. . . . . | 133 |
| 6.38 | Entrées de commandes du système en forme de $\times$ pour le deuxième scénario de vol. En noir les entrées de commande; en bleu le mode 4DDL; en gris le mode 6DDL initialisation; sans couleur le mode autonome. . . . .           | 133 |
| 6.39 | États des actionneurs pour le système en forme de $\times$ pour le deuxième scénario de vol. En noir les mesures des états; en bleu le mode 4DDL; sans couleur le mode autonome; les références en rouge. . . . .                   | 134 |

|      |  |     |
|------|--|-----|
| 6.40 | Entrées de commandes du système en forme de $\times$ pour le deuxième scénario de vol. En noir les entrées des actionneurs ; en bleu le mode 4DDL ; sans couleur le mode autonome. . . . .                           | 134 |
| 6.41 | États du système en forme de $\times$ pour le deuxième scénario de vol. En noir les états sans compensation ; en bleu avec la compensation des couples gyroscopiques ; les références en rouge. . . . .              | 135 |
| 6.42 | États des actionneurs pour le système en forme de $\times$ pour le deuxième scénario de vol. En noir les entrées des actionneurs sans compensation ; en bleu avec la compensation des couples gyroscopiques. . . . . | 135 |
| 6.43 | Entrées de commandes du système en forme de $\times$ pour le deuxième scénario de vol. En noir les entrées de commande sans compensation ; en bleu avec la compensation des couples gyroscopiques. . . . .           | 136 |
| 6.44 | États du système en forme de $\lambda$ pendant la simulation. En noir les états ; les références en rouge. . . . .   | 137 |
| 6.45 | États des actionneurs du système en forme de $\lambda$ pendant la simulation   | 138 |
| 6.46 | Valeurs des entrées de commandes du système en forme de $\lambda$ pendant la simulation . . . . .  | 138 |
| 6.47 | États du système en forme de $\lambda$ pendant les tests réels. En noir les états ; en bleu le mode 4DDL ; en gris le mode 6DDL ; les références en rouge.   | 139 |
| 6.48 | Photographie du drone en configuration $\lambda$ pendant une inclinaison importante . . . . .  | 139 |
| 6.49 | États du système en forme de $\lambda$ pendant les tests réels. En noir la position du système ; en bleu le mode 4DDL ; sans couleur le mode 6DDL. . . . .   | 140 |
| 6.50 | États des actionneurs du système en forme de $\lambda$ pendant les tests réels. En noir les entrées des actionneurs ; en bleu le mode 4DDL ; en gris le mode 6DDL. . . . .   | 140 |
| 6.51 | Valeurs des entrées de commandes du système en forme de $\lambda$ pendant les tests réels. En noir les entrées de commande ; en bleu le mode 4DDL ; en gris le mode 6DDL. . . . .                                    | 141 |
| 6.52 | États du système en forme de $\lambda$ pendant les tests réels. En noir les états ; en bleu le mode 4DDL ; sans liseré, le mode 6DDL ; les références en rouge.  | 141 |
| 6.53 | Photographie du drone $\lambda$ en vol à plus de 80 degrés . . . . .   | 142 |
| 6.54 | Photographie du drone $\lambda$ en vol avec le porte outil . . . . .   | 142 |
| B.1  | Diagramme en pieuvre AMDEC structure . . . . .   | 164 |
| B.2  | Diagramme en pieuvre AMDEC système embarqué . . . . .  | 165 |
| B.3  | AMDEC FP - Structure . . . . .   | 166 |
| B.4  | AMDEC FC - Structure . . . . .   | 167 |
| B.5  | AMDEC FC - Système embarqué . . . . .  | 168 |
| D.1  | Image et photo des pièces pour le capteur de position servomoteur . . . . .  | 174 |
| D.2  | Photo du capteur à rotation continue seul (gauche) et implémenté (droite)  | 175 |
| D.3  | Angle du servomoteur par rapport à une consigne rapide (gauche) et lente (droite) . . . . .  | 175 |
| D.4  | Exemple de 3 cas pour la gestion de position des servos . . . . .  | 176 |
| D.5  | Photo du boîtier pour l'intégration de l'arduino . . . . .   | 176 |
| D.6  | Deuxième version de l'embout entre le servo et le tube en carbone . . . . .  | 177 |
| D.7  | Exemple d'utilisation du module real-time peacer . . . . .   | 181 |
| D.8  | Organisation d'un modèle Simulink . . . . .  | 181 |
| D.9  | Bascule D dans Simulink . . . . .  | 182 |



## Avant-propos

Dans ce manuscrit je présente mes travaux de thèse que j'ai eu l'honneur de réaliser avec l'entreprise Ginger Centre d'Expertise Bâtiment Travaux Publics (CEBTP) (anciennement Ginger Modeliris) à Béthune et le laboratoire Heudiasyc à Compiègne dans les Hauts-de-France. Après un an d'ingénierie au service de Modeliris (2016), les travaux de thèse ont réellement commencé en décembre 2017 pour finir début 2021. Ceux-ci ont été suivis par Méchaal Rihani dans l'entreprise et dirigés par Pedro Castillo-García et Jérôme De Miras au laboratoire. Le partenariat est né d'une alliance tétrapartite entre l'entreprise spécialisée dans l'expertise d'ouvrages d'art, un laboratoire ayant de nombreuses connaissances en robotique, l'Association Nationale Recherche Technologie (ANRT) et moi-même.

Depuis toujours passionné par la robotique et notamment par tous les aspects de la mécatronique, j'ai cherché à compléter mes connaissances acquises en master à Strasbourg par un doctorat dans la même discipline. Souhaitant travailler en entreprise, la thèse Conventions Industrielles de Formation par la REcherche (CIFRE) s'est présentée comme une évidence à mes yeux. L'offre proposée par Ginger correspondait en tout points à mes attentes. La première année dans l'entreprise m'a permis de comprendre les problèmes auxquels les ingénieurs du génie civil peuvent être confrontés. Ce fut également l'occasion de monter le dossier et d'avoir un premier échange avec le laboratoire et l'ANRT. Pour expliquer l'origine de cette thèse, revenons d'abord au domaine d'expertise de l'entreprise.

L'inspection des ouvrages d'art a toujours été importante et l'est d'avantage depuis quelques décennies, compte tenu de la croissance démographique et géographique de nos villes. Les outils et procédés d'inspection sont voués à évoluer et permettent de caractériser de plus en plus précisément l'état structurel de nos maisons, mais aussi des bâtiments industriels, des ouvrages pour le transport, les infrastructures pour l'énergie, etc. Une fois l'inspection effectuée, des mesures peuvent être prises sur les ouvrages pour garantir la sécurité des usagers.

Parmi les outils et procédés d'inspection les plus courants, la mesure de potentiel électrique du béton permet de caractériser l'état de corrosion des métaux internes. Les relevés au pacomètre permettent quant à eux de relever la présence de fers pour comprendre comment l'ouvrage a été construit. Le relevé de désordre, sert à caractériser l'état global de santé d'une structure et permet de prendre des mesures le cas échéant. Les procédés récents utilisent entre autre la création d'orthomosaiques. Ce puzzle de reconstitution d'image donne un relevé des désordres exhaustif et précis.



Cependant, le relevé à pied peut s'avérer compliqué si l'ouvrage est trop imposant, ou difficile d'accès. L'utilisation de nacelles ou de cordistes permet l'accès à ces parties délicates. Certaines parties des ouvrages restent, malgré ces techniques, difficiles d'accès. Ce contexte permet de poser l'objectif de la thèse.

## 1.1 Objectif et problématiques

L'objectif de cette thèse est de **rendre possible l'inspection d'ouvrages d'art dans des zones difficilement accessibles**. Au cours de la réflexion pour atteindre cet objectif, deux problématiques principales ont rapidement émergé. Par la suite, deux questions supplémentaires, liées à la plateforme, sont apparues.

- 🐜 **Quelle forme doit avoir le robot qui effectue les mesures sur ouvrage ?**
- 🐜 **Comment sera commandé ce robot ?**
- 🐜 **Quels points de conception mécanique devront être optimisés pendant la réalisation du dispositif ?**
- 🐜 **Quelle plateforme informatique permet d'effectuer des tests rapidement et efficacement ?**

## 1.2 Organisation du manuscrit

- 🐜 Le Chapitre 2 discute du type de mécanisme nécessaire pour effectuer le relevé sur ouvrage, ce qui donne des éléments de réponse à la première problématique. Après un rapide état de l'art, différentes solutions pour répondre au problème d'accès, et au besoin d'efficacité sur chantier sont proposées. De la grue radio-commandée, à l'ascenseur vertical, c'est finalement le drone qui sera retenu pour porter les outils. Deux types de drones sont susceptibles de répondre au besoin. Un drone classique avec bras manipulateur et un drone libre de ses mouvements indépendamment. Une analyse permet de présenter pourquoi c'est cette dernière solution qui sera retenue.
- 🐜 Le Chapitre 3 détaille le choix du type de solution retenue. Pour ce faire, une analyse de différents modèles mécaniques candidats est faite. Après une première approche sur un système 2D, plan, trois configurations différentes seront présentées pour finalement aboutir à deux systèmes viables pour ce projet. Ces modèles permettent d'apporter des éléments de réponse à la première problématique.
- 🐜 Le Chapitre 4 présentera différentes approches qui seront testées et validées en simulation comme en tests réels, permettant d'apporter des éléments de réponse à la deuxième problématique. Une méthode de linéarisation du système sera présentée permettant de tester un très grand nombre de lois de commandes sur le système. Un algorithme de commande linéaire classique sera détaillé dans un premier temps et principalement utilisé lors des tests. En complément, l'étude proposera l'emploi de lois de commande non linéaires comme les saturations imbriquées et un mode glissant du deuxième ordre. Ce dernier algorithme permet de garantir des performances de vol intéressantes, et possède la particularité d'être robuste aux perturbations (le vent par exemple).

✿ Pour répondre aux deux dernières problématiques, on propose un focus sur la plateforme (mécanique et informatique) développée dans le Chapitre 5.1. Deux des configurations décrites dans le Chapitre 3 seront effectivement construites et testées expérimentalement. La première partie du Chapitre 5.1 permettra de détailler tous les choix de conceptions effectués ainsi que toutes les réalisations annexes qui ont été nécessaires pour la création du dispositif. Cette première section permettra également d'expliquer le choix des matériaux, du type de réalisation et de l'intérêt, le cas échéant de proposer une solution propriétaire plutôt qu'une solution disponible dans le commerce.

La deuxième partie du chapitre permet d'exposer la plateforme informatique qui a été développée pour le système. En effet, le choix de la plateforme est cruciale pour la réalisation des essais pratiques. La solution la plus courante consiste à utiliser une infrastructure logicielle existante (framework) et de la modifier pour les besoins d'utilisation. Dans ce document on propose d'adapter un framework existant pour le rendre compatible avec quasiment n'importe quelle application robotique. Le dispositif pourra ainsi être testé pour les deux configurations proposés demandant un changement minimal entre les deux versions. Le deuxième intérêt de cette architecture est une programmation haut niveau, donc sans programmation à proprement parler permettant de tester beaucoup plus rapidement sans être gêné par une phase de débogage parfois longue et fastidieuse.

✿ Le Chapitre 6 permettra finalement de présenter et surtout d'analyser les différents résultats obtenus aussi bien en simulation qu'en expérimentations. Un comparatif des deux est fait mettant en évidence nombre de difficultés rencontrés. Un utilisateur futur pourra donc contourner ces points en utilisant directement les solutions proposées. Un certains nombres de problèmes apparaissant au cours des tests ont également permis des changements sur les autres parties (conception, commande et autre). Il s'agit donc finalement d'une analyse critique permettant de pouvoir proposer des perspectives pour des travaux futurs.

L'étude présentée dans ce document propose une réponse à l'objectif en choisissant de développer un RoMEO, un dispositif volant permettant d'effectuer les relevés au contact de l'ouvrage. On montre les solutions mais également les étapes qui ont permis la construction du dispositif avec toutes les études annexes. Afin de pouvoir continuer un développement futur, j'ai essayé d'écrire le document de la manière la plus didactique possible, reprenant pas à pas chaque étape pour enfin aboutir aux résultats. Cela permettra une prise en main du sujet facile par un lecteur qui n'est pas forcément du domaine de la mécatronique.

J'espère que ces lignes pourront transmettre au lecteur toutes la joie que l'on peut ressentir quand on construit un robot du début à la fin pour finalement le faire voler.

Bonne lecture.



## Contexte de l'étude

Après une rapide présentation de l'entreprise, ce chapitre explique le besoin qui a initié ce projet. Les problématiques présentées dans l'avant propos mènent aux contraintes inhérentes au projet. Un état de l'art des solutions existantes est proposé. Une attention particulière est portée aux drones possédant une configuration innovante. L'ensemble de ces recherches apporte des éléments de réponse à la première problématique ainsi qu'une orientation globale au projet, notamment sur le choix du laboratoire de recherche associé. Une fois celui-ci présenté, on s'attarde sur la gestion du projet. Le projet est découpé en différentes étapes à réaliser au cours des trois années de thèse. Les moyens mis en œuvre sont ensuite décrits pour répondre à la problématique, tant pour la conception mécanique que pour la partie concernant la commande ou encore les développements de la plateforme.

### 2.1 Présentation de l'entreprise

Initialement, l'entreprise s'appelait Do You Drone et était basée à Wasquehal (59). Celle-ci fut rachetée par le groupe Ginger CEBTP pour devenir Ginger Modeliris. Le terme venant de la concaténation de modélisme et d'iris (caméra). L'entreprise était initialement spécialisée dans la prise de vue d'ouvrages d'art. Cette inspection est réalisée à l'aide de différents moyens et milieux : en extérieur (monuments et ouvrages d'art, inspection de routes) et en intérieur (tunnels ou piles de pont). L'outil le plus utilisé pour les prises de vue reste à l'heure actuelle le drone, ou Unmanned Aerial Vehicle (UAV). [1] présente les enjeux du développement de cette nouvelle technique de prise de vue pour le domaine de l'inspection d'ouvrages.

Les photos prises d'un bâtiment, comme à gauche en Figure 2.1, permettent de créer des modèles 3D. On pourra en extraire des orthomosaïques (une mosaïque d'images), à l'échelle, telle qu'elle est présentée à droite en Figure 2.1. Ces images permettent de mettre en avant plus facilement les défauts des structures qui seront retracés manuellement dans un logiciel de dessin adéquat, au centre en Figure 2.1. Les plans réalisés seront ensuite utilisés pour permettre un diagnostic précis de l'ouvrage inspecté.

Le groupe international Ginger CEBTP est plus globalement spécialisé dans l'expertise et le diagnostic dans le domaine du Bâtiment et Travaux Publics (BTP). Les domaines d'expertise sont vastes et présentés ci-après :

- la géotechnique : étude des sols et des fondations des bâtiments ;
- la route : expertise des chaussées, diagnostic des pathologies rencontrées et propositions de solutions adaptées ;

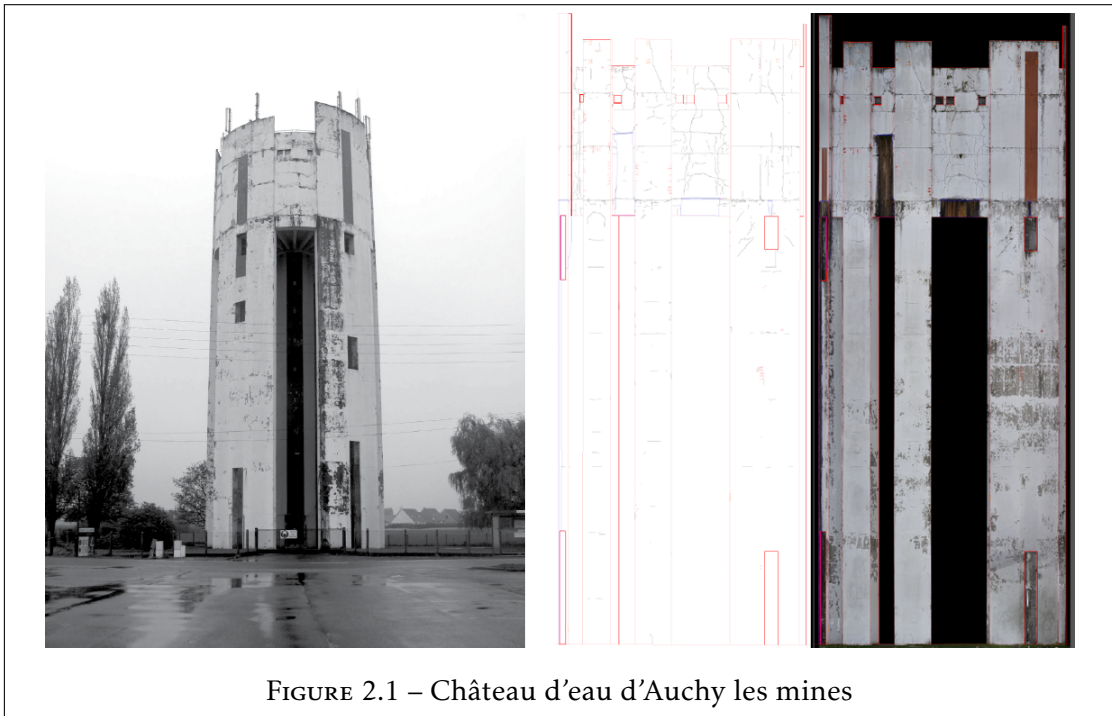


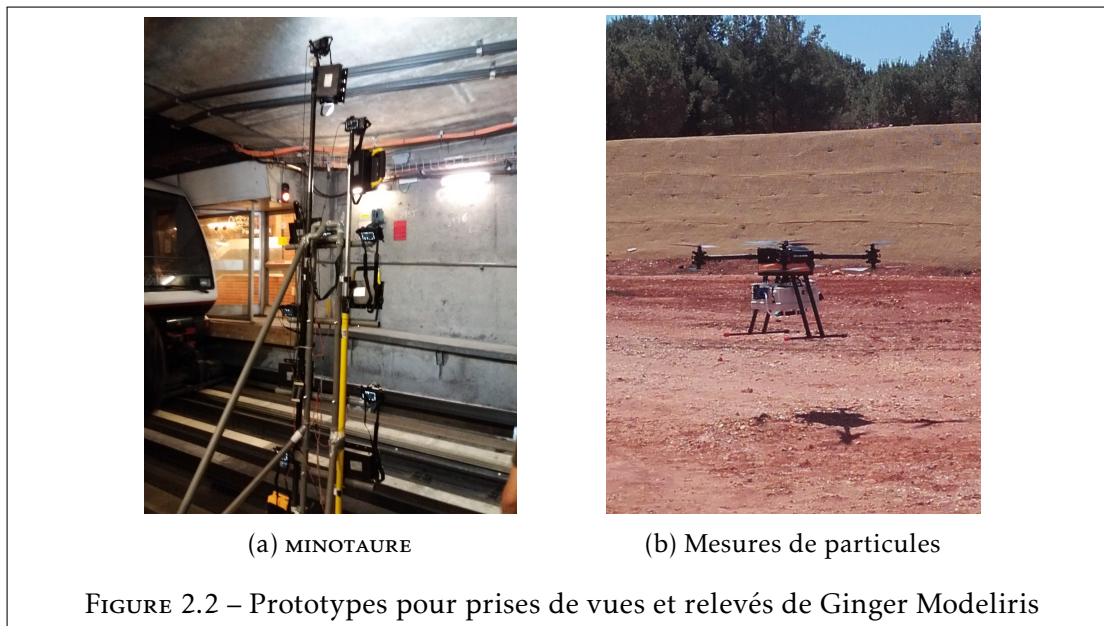
FIGURE 2.1 – Château d'eau d'Auchy les mines

- l'environnement (Burgeap) : gestion des sites pollués, protection de l'environnement, développement durable, énergie, analyses des sols ; cette filiale possède un centre de recherche et développement ce qui permet de proposer des solutions innovantes ;
- l'analyse des mouvements du sol : que ce soit pour prévenir les glissements de terrain ou les risques naturels, l'entreprise dispose un bureau d'étude dans le domaine ;
- le support : aussi bien sur les formations, la certification, que sur le support informatique mais aussi dans le domaine de la santé, le groupe possède des ressources permettant de réaliser des prestations personnalisées et rapides ;
- la déconstruction : filiale spécialisée dans la dé-construction et le démantèlement des bâtiments ;
- les réseaux : analyse et relevés des différents réseaux existants, principalement dans les sols (électriques, gaz, télécommunication, conduites d'eau...);
- **la pathologie des structures** : c'est dans ce dernier domaine que s'inscrit cette thèse. Il s'agit de faire un diagnostic de la structure des ouvrages. Cela peut concerner un pont, comme une église en passant par les hôpitaux et écoles. La réalisation des relevés photographiques décrits précédemment permettent en ce sens de faciliter le diagnostic de l'ouvrage, en étant à la fois rapide et exhaustif ;

Les missions de l'entreprise ne se limitent cependant pas qu'aux prises de vues par drone. Plusieurs prototypes et autres expérimentations ont également été développés. On pourra par exemple citer le Moyen d'Inspection Numérique pour les Ouvrages des Tunnels AUTomatique Rapide et Exhaustif (MINOTAURE). Ce dispositif destiné à prendre une grande quantité de photos dans les tunnels a été conçu en deux étapes, d'abord en tubes PVC, Figure 2.2a, puis en aluminium. En fonction de l'avancement du dispositif dans le tunnel, les photos sont prises sur l'ensemble des appareils photos. Des éclairages sont également disposés permettant d'améliorer la qualité de prises de vue.

D'autres projets sont également réalisés, notamment en partenariat avec la filiale Ginger Burgeap. On peut, par exemple, citer l'intégration de capteurs de mesures de particules sur un drone du commerce, montré en vol en Figure 2.2b.

De plus amples informations sur ces projets, ainsi que sur les formations suivies et la diffusion scientifique, sont présentées en Annexe A.



## 2.2 Motivations de l'entreprise

Les relevés visuels évoqués précédemment sont complétés par des relevés physiques pour préciser le diagnostic du bâtiment. Parmi ces relevés on citera :

- la présence et profondeur des fers dans le béton avec un pacomètre. La Figure 2.3a montre le Profometer® de l'entreprise Proceq™ et la Figure 2.3b montre le Ferros-can® de la marque Hilti™ ;
- la profondeur de pénétration du carbone dans le béton en utilisant la coloration à la phénolphthaléine, voir Figure 2.3c ;
- des mesures à ultrasons pour caractériser les matériaux ;
- des mesures de l'intérieur des parois avec le radar. Exemple du StructureScan Mini XT® de la marque GSSI™, en Figure 2.3d ;
- le potentiel électrique du béton pour mesurer la corrosion des fers internes, voir Figure 2.3e ;
- des prélèvements de poussières ou de carottes pour analyse ;
- des mesures de l'ouverture des fissures ;
- des test d'arrachement pour caractériser l'adhérence de surface ;

Les données pourront ensuite être exploitées pour réaliser un diagnostic plus complet de l'ouvrage par des experts du groupe. Enfin, ce diagnostic permettra de proposer des mesures pour corriger (le cas échéant) la pathologie de l'ouvrage.



Le document [2], complète les informations sur les différents procédés de mesure ainsi que les pathologies que peuvent avoir les bâtiments.

A l'heure actuelle, l'inspection des ouvrages se fait principalement manuellement. Les différents appareils de mesure demandent l'emploi d'un opérateur qualifié, qui devra aller au contact du bâtiment, parfois dans des conditions d'accès très compliquées. Si l'accès est le plus souvent réalisé à pied ou en nacelle de par le sol, certains ouvrages nécessitent un accès par le haut qui pourra se faire par des cordistes ou des nacelles inversées, voir Figure<sup>1</sup> 2.4 .



1. <https://www.dumont-clean.com/wp-content/uploads/2019/02/AB16-Nuit-3-1024x768.jpg>

L'opération peut cependant être très complexe, dangereuse ou même impossible dans certains cas. Comme exemple, on peut citer les tours aéroréfrigérantes des centrales électriques qui sont des ouvrages tellement immenses que leur inspection nécessite plusieurs jours d'arrêt. L'emploi d'un système autonome permet de pouvoir continuer l'exploitation simultanément. D'autres ouvrages comme certains châteaux d'eau présentent la même difficulté comme illustré en Figure 2.5. Cette thèse a donc été proposée afin de répondre à ce besoin, et pour offrir davantage de perspectives à l'entreprise. L'inspection visuelle par drone sera alors complétée par une inspection directement au contact des ouvrages.

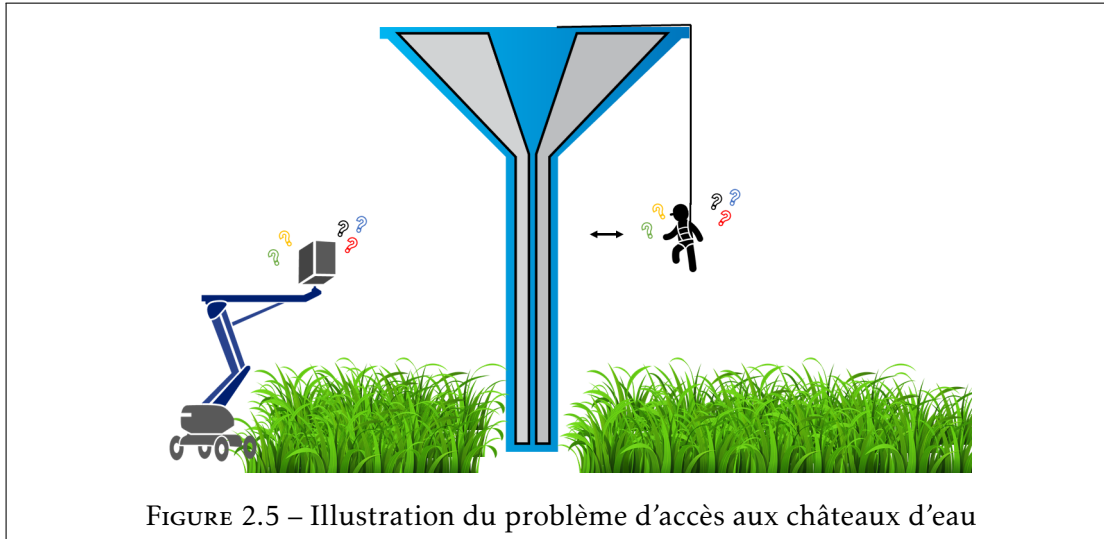


FIGURE 2.5 – Illustration du problème d'accès aux châteaux d'eau

## 2.3 Cahier des charges

L'objectif et le contexte du projet étant clarifiés, on propose de décomposer le problème en fonction de 8 contraintes permettant d'aider le choix du dispositif à réaliser. Celles-ci sont explicités dans le Tableau 2.1 suivant :

| Fonction            | Critère   | Niveau |
|---------------------|---|--------|
| FP1                 | Amener un outil de mesure à un endroit inaccessible             |        |
| FC1 : Autonomie     | Robot utilisable sur toute la durée de l'essai                  | 2      |
| FC2 : Résistance    | Robustesse face aux rafales de vent / poussières / ...          | 2      |
| FC3 : Sécurité      | Système sûr vis à vis des personnes et des biens                | 3      |
| FC4 : Accessibilité | Capacité à accéder facilement à un point de l'ouvrage           | 3      |
| FC5 : Encombrement  | Facilité d'amener-repli   | 2      |
| FC6 : Versatilité   | Système utilisable sur différents types d'ouvrages <sup>2</sup> | 2      |
| FC7 : Esthétique    | Rendu visuel et image de la société                             | 1      |
| FC8 : Ergonomie     | Facilité d'utilisation  | 2      |

TABLEAU 2.1 – Fonction principale et contraintes du projet

Les niveaux vont de 1 à 3.

F.P. : Fonction Principale.

F.C. : Fonction Contrainte.

<sup>2</sup> Mais également être capable de gérer des obstacles qui pourrait déranger le relevé (bords de fenêtre, câbles, armatures, ...)



On propose quatre solutions technologiques pour répondre à ces critères :

- Le "Lézard", ancien dispositif de l'entreprise semblable à un ascenseur nécessitant deux cordes avec une attache en haut et en bas de l'ouvrage
- Un drone avec un outil intégré
- Un robot "ventouse" qui se plaque sur l'édifice par succion
- Un bras manipulateur télescopique opérant depuis le sol

Le Tableau 2.2 classe le niveau du dispositif par rapport aux contraintes. En fonction de la contrainte, la valeur sera pondérée par un coefficient. Compte tenu du contexte d'utilisation, on choisit un coefficient à 3 pour la sécurité (FC3) et l'accessibilité (FC4) que doit avoir le dispositif. En effet, les ouvrages possédant une architecture souvent complexe, l'objectif premier du dispositif est d'accéder aux endroits difficiles pour l'homme. Les autres critères ont un coefficient normal (2), sauf le critère esthétique qui ne doit pas être au centre des préoccupations de l'utilisateur, la finalité étant une recherche de résultat.

| Solution    | FC1 | FC2 | FC3 | FC4 | FC5 | FC6 | FC7 | FC8 | Total |
|-------------|-----|-----|-----|-----|-----|-----|-----|-----|-------|
| Coefficient | (2) | (2) | (3) | (3) | (2) | (2) | (1) | (2) |       |
| Lézard      | 3   | 4   | 4   | 1   | 3   | 2   | 3   | 3   | 48    |
| Drone       | 2   | 2   | 3   | 4   | 4   | 4   | 4   | 3   | 55    |
| "ventouse"  | 2   | 2   | 1   | 3   | 4   | 3   | 4   | 3   | 44    |
| Bras        | 4   | 3   | 2   | 2   | 2   | 1   | 2   | 4   | 42    |

TABLEAU 2.2 – Classification des solutions technologiques

Note : les niveaux pour ce tableau vont de 1 à 4. On classe chaque solution les unes par rapport aux autres pour tous les critères.

- Pour FC1, le bras étant au sol, il aura la meilleure autonomie ; le lézard tenu par des câbles n'aura pas besoin de beaucoup d'énergie et sera donc 2nd ; les solutions "ventouse" et drone sont à égalité en troisième place avec une consommation d'énergie conséquente.
- Pour FC2, le lézard est directement plaqué sur la paroi et tenu par des câbles ce qui en fait le système le plus robuste face aux perturbations. Le bras sera très stable en base mais le sera de moins en moins en altitude. Enfin, la "ventouse" et le drone seront impactés de la même manière par le vent ou les aspérités de l'ouvrage.
- Pour FC3, Le lézard est un dispositif captif donc avec très peu de risques. Le drone peut certes être affecté par le vent mais il est conçu pour rester stable en l'air. Le bras télescopique risque de se balancer beaucoup à partir d'une certaine hauteur rendant le système dangereux, mais toujours moins que le système de "ventouse" qui risque de se décrocher.
- Pour FC4, le drone accède facilement et rapidement à n'importe quel point sur l'ouvrage, sans nécessité d'avoir un accès à l'édifice. Après un positionnement en bas de l'ouvrage et un déplacement vertical, la "ventouse" pourra en faire de même, nécessitant tout de même un accès à au moins un point de l'ouvrage. Le bras devra être placé à la verticale et donc un accès à la circonférence doit être possible. Enfin le lézard nécessite un accès par le toit pour accrocher les câbles, de plus il ne pourra pas accéder aux sections en dévers, lui octroyant la dernière place.

- Pour FC5, le drone et la "ventouse" ont un encombrement modéré. Le système de lézard avec les câbles vient donc en position suivante dans le classement. Le dispositif de bras quant à lui semble présenter le plus d'encombrement. Il faudra en effet considérer un système d'alimentation auxiliaire ainsi que la place pour rétracter les tubes.
- Pour FC6, Le drone peut être utilisé sur tous les types d'ouvrages. La "ventouse" sera gênée par les obstacles de surface la plaçant juste après. Comme déjà évoqué, le lézard ne pourra pas effectuer de relevé sur les sections en dévers le rendant moins versatile. Enfin, le bras va être limité à des ouvrages relativement peu hauts lui donnant à nouveau en dernière place.
- Pour FC7, le critère est noté de manière beaucoup plus subjective. Le drone et la "ventouse" sont à la pointe de la technologie et donc attractif pour un client. Le lézard peu témoigner d'une certaine élégance et simplicité. Quand au bras télescopique il sera placé en dernière position car il sera certainement très encombrant et reste malgré tout peu versatile, donc pas très esthétique.
- Pour FC8, on peut imaginer que le bras télescopique est commandable par un simple joystick alors que les autres dispositifs nécessiteront une formation qui les rend moins facile d'utilisation.

Suite à cette analyse on choisit donc le drone pour aborder cette problématique. Afin de ne pas tomber dans une solution déjà existante dans le commerce, on réalise un état de l'art. Il sera complété tout au long de l'étude, mais on propose d'exposer nombre de solutions existantes dans la section suivante.

## 2.4 État de l'art

### 2.4.1 Les drones classiques

Toutes les définitions concordent pour dire que les drones sont des aérodyne sans équipages (UAV), télé-pilotés ou encore autonomes. Les premiers drones sont apparus au début du siècle dernier et se sont énormément développés pendant la guerre froide à des fins militaires, et au début du millénaire pour des applications civiles.

Parmi les premiers drones on retrouve par exemple l'Aerial target, un avion cible télécommandé par des ondes télégraphiques, pendant la première guerre mondiale. Plus connues, les fusées V1 et 2 étaient des missiles munis d'un auto-pilote, pour plus de détails, consulter [3]

Comme A. Koehl le décrit largement dans son mémoire de thèse [4], de nombreuses versions différentes de drones existent (de l'hélicoptère au quadcopter en passant par le zeppelin). Les domaines d'utilisation se sont multipliés au cours des dernières années, dont voici quelques exemples (plus dans [5]) :

- le domaine militaire, voir Figure <sup>3</sup> 2.6a;
- le domaine humanitaire (aide suite à des catastrophes naturelles);
- la santé, voir Figure <sup>4</sup> 2.6b;

---

3. <https://www.af.mil/News/Photos.aspx?igphoto=2001601881>

4. [https://www.pourquoidoctor.fr/media/article/thunbs/uploded\\_drone-1438187062.jpg](https://www.pourquoidoctor.fr/media/article/thunbs/uploded_drone-1438187062.jpg)

- la protection civile, voir Figure <sup>5</sup> 2.6c ;
- l'agriculture, voir Figure <sup>6</sup> 2.6d ;
- la protection des espaces naturels ;
- l'inspection BTP exemple des débuts consultable dans [6] et illustré en Figure <sup>7</sup> 2.6e ;
- les expériences scientifiques ;
- la géodésie ;
- le transport, voir Figure <sup>8</sup> 2.6f ;
- les loisirs, voir Figure <sup>9</sup> 2.6g ;
- les médias et le journalisme, voir Figure 2.6h ;
- la télécommunication.



Il existe de nombreuses marques et modèles de drone civils. On peut citer le constructeur chinois DJI© qui est devenu leader mondial dans le domaine. Modeliris utilise d'ailleurs le Phantom 4 pour ses inspections visuelles, voir Figure 2.6h. Dans cette partie, on ne fera pas davantage de descriptif sur les différents drones classiques existants dans le domaine civil et militaire.

5. <https://www.france24.com/fr/20200506-covid-19-bataille-judiciaire-autour-de-l-utilisation-des-drones-par-la-police-française>

6. <https://www.studiosport.fr/guides/drones/a-quoi-servent-les-drones-en-agriculture.html>

7. <https://www.ginger-cebtp.com/photos/diagnostic-pathologie-structure/>

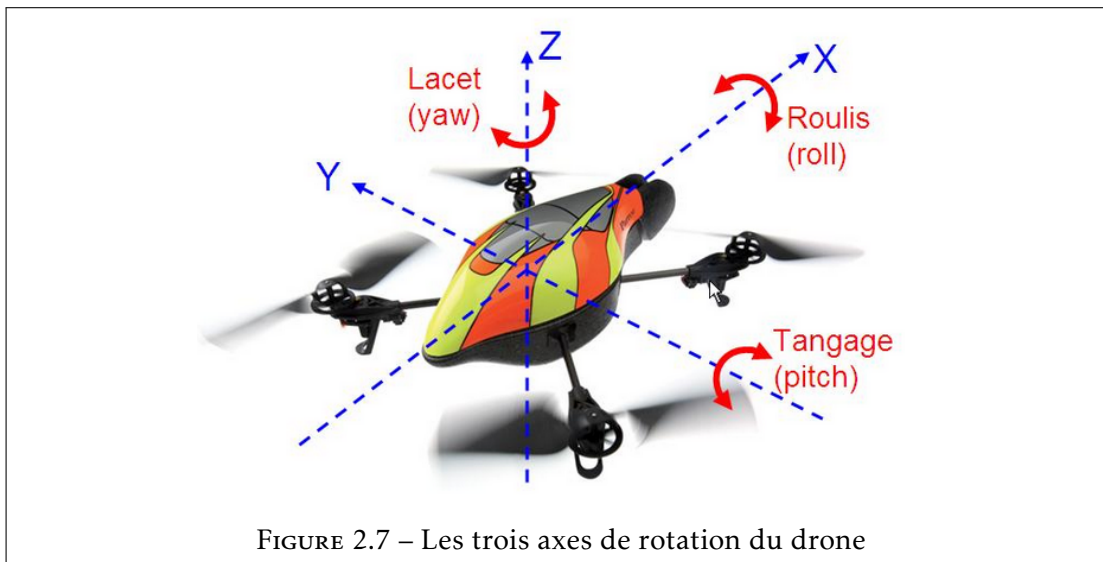
8. <https://www.franceinter.fr/societe/amazon-lance-un-centre-de-developpement-pour-la-livraison-par-drones-en-france>

9. [https://img.lemde.fr/2018/09/03/0/0/2048/1535/688/0/60/0/b9a7d9e\\_UMKcscSjaa5yV8Soip64XEfv.jpeg](https://img.lemde.fr/2018/09/03/0/0/2048/1535/688/0/60/0/b9a7d9e_UMKcscSjaa5yV8Soip64XEfv.jpeg)

Dans la section suivante, on présente les configurations de vols particulières. Parmi celles-ci on s'intéressera particulièrement aux solutions répondant directement à notre sujet. Cette base sera ensuite utile pour créer un drone permettant non seulement de répondre à la problématique, mais également à toutes les contraintes du projet.

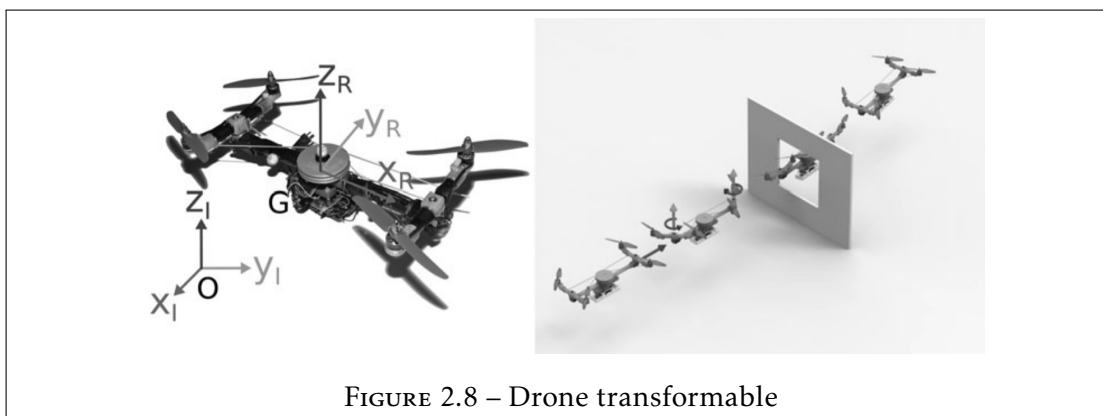
### 2.4.2 Drones avec configurations de vols particulières

Avant de commencer la présentation de différentes configurations de vols particulières présentées dans la littérature, on définit en Figure 2.7 (source <sup>10</sup>) les noms pour les rotations du drone. Cela permettra d'explicitier plus simplement les différents mouvements des l'aéronefs.



Note : les indications de la figure ne sont que valables pour de faibles orientations du drone.

Dans la littérature on retrouve de nombreuses études concernant le changement de configurations de vols dont on pourra s'inspirer. [7] propose un quadricoptère particulier, présenté en Figure 2.8. Le châssis est en trois parties formant un "I". Les parties supérieures et inférieures (bâses horizontales du "I") pivotent à 90° autour d'un axe vertical pour finalement obtenir un véhicule où les quatre hélices se suivent en ligne. Le véhicule se transforme alors temporairement en *Planar Vertical TakeOff and Landing aircraft* (PVTOL) pour se glisser dans des petits orifices.



10. <https://ekldata.com/3WH3hbe2YHmSXAtnPiqn4iPxrrM.png>

En termes de configuration particulière, on peut également citer les tri-rotors, décrits par exemple dans [8]. De par leur conception, le système possède le plus souvent une hélice inclinée ou inclinable pour compenser le couple sur le lacet. On pourra s'inspirer de cette particularité pour proposer une première solution à la problématique. On cite également [9], [10] qui présente un drone capable de changer sa configuration pour devenir un avion, voir Figure 2.9. Le châssis reste donc horizontal, et doit avancer pour garder une portance, ce qui ne correspond pas totalement au sujet. Cette configuration est cependant très intéressante pour faire des trajets de longue distance tout en proposant décollages et atterrissages verticaux.

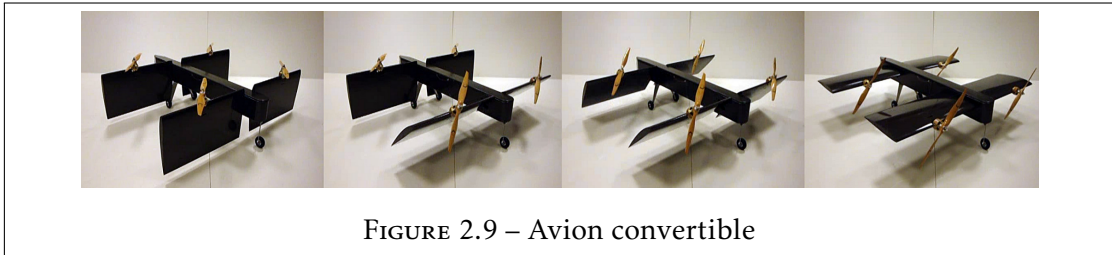


FIGURE 2.9 – Avion convertible

Dans [11], l'auteur propose de détailler un drone en forme de cube en positionnant les hélices de manière à obtenir une contrôlabilité totale sur les 6 DDL. Dans cette configuration, les hélices sont complètement fixes et peuvent, en fonction de l'orientation du drone, ne pas être utilisées du tout. Avoir des configurations de vol où tous les actionneurs ne sont pas utilisés implique un ajout de poids, point négatif en aéronautique. De plus, l'auteur propose 3 configurations différentes. Celle qui est retenue et construite, possède le plus de symétries, point essentiel pour assurer la stabilité mécanique. L'ajout d'actionneurs garantit une contrôlabilité totale du système, contrairement à une configuration de drone classique. Une image de ce dispositif est présenté en Figure 2.10,<sup>11</sup>. De même dans [12], l'auteur propose l'analyse d'un

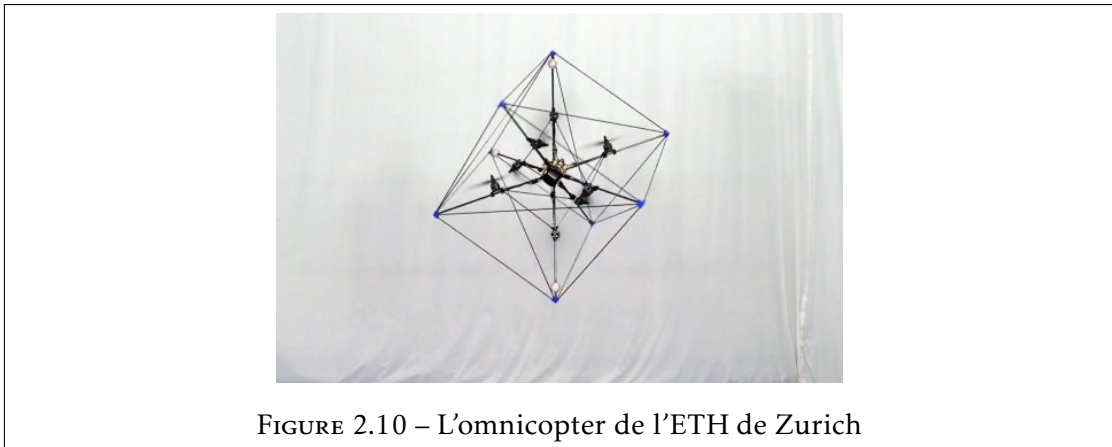


FIGURE 2.10 – L'omnicopter de l'ETH de Zurich

hexacopter avec des bras inclinés, fixes, permettant de voler indépendamment dans tous les DDL. De par sa configuration et les actionneurs choisis, une saturation sur l'indépendance de l'orientation du drone apparaît. On peut tout de même profiter de cette configuration pour pouvoir développer un vol plus acrobatique et stable que pour une configuration classique. Des travaux homologues peuvent être lus dans [13], [14]. On y présente un aéronef avec 6 hélices inclinées le long d'une tige permettant un déplacement omni-directionnel dans l'espace.

11. <https://www.helicomicro.com/wp-content/uploads/2016/08/theomnicopter-494x322.jpg>

Dans [15] (projet Aeroarms<sup>12</sup>), on propose l'utilisation d'un bras manipulateur piloté, présenté en Figure 2.11a. Celui-ci consiste en une barre sur laquelle est placée le capteur en bout. Cette barre peut tourner autour de l'axe de tangage du drone. Le couple en bout de bras est compensé par le placement des batteries de l'autre côté du bras. Celui-ci amène un capteur sur une paroi, permettant un positionnement des outils au contact de l'ouvrage. Cet exemple ouvre également des possibilités pour répondre d'une autre façon à la problématique. On cite également [16] proposant un bras à 2-DDL commandé par une approche de commande adaptative d'un modèle référence, présenté en Figure 2.11b. Cela permet de garantir le bon fonctionnement de la manipulation pour n'importe quelle charge utile, et donc l'inertie du système.

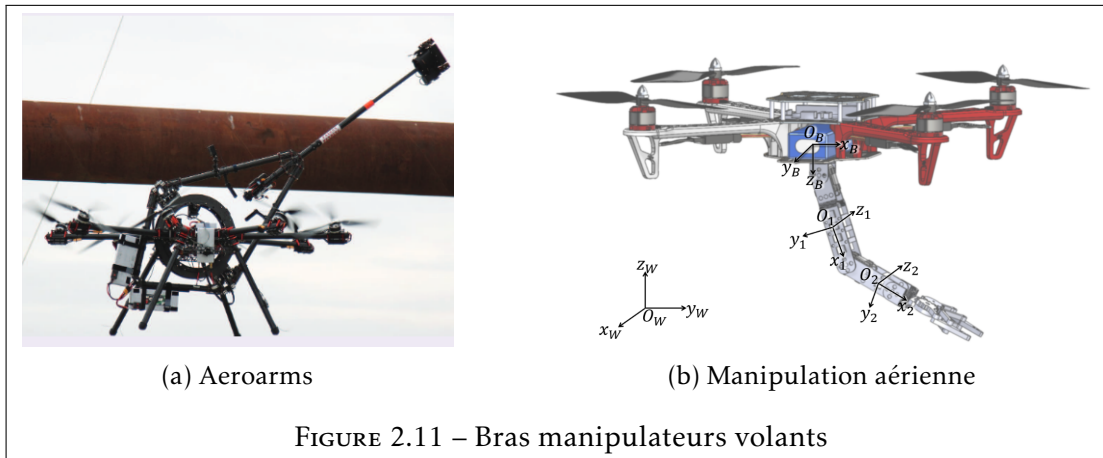


FIGURE 2.11 – Bras manipulateurs volants

Dans [17], [18] on emploie les drones pour la co-manipulation d'objets avec des bras industriels automatisés, présenté respectivement en Figure 2.12b et 2.12a. [19] propose la commande d'un bras manipulateur accroché sur un drone en vol. Deux lois de commandes linéaires sont proposées pour contrôler à la fois le drone et le bras. Celles-ci sont couplées pour garantir le suivi de position de l'extrémité du bras. Pour finir, une étude similaire dans [20] propose une commande par cinématique inverse, validée en simulation.

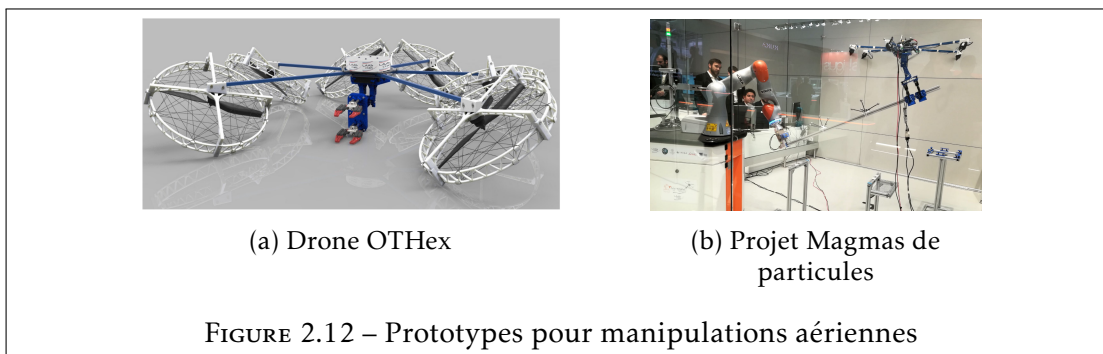
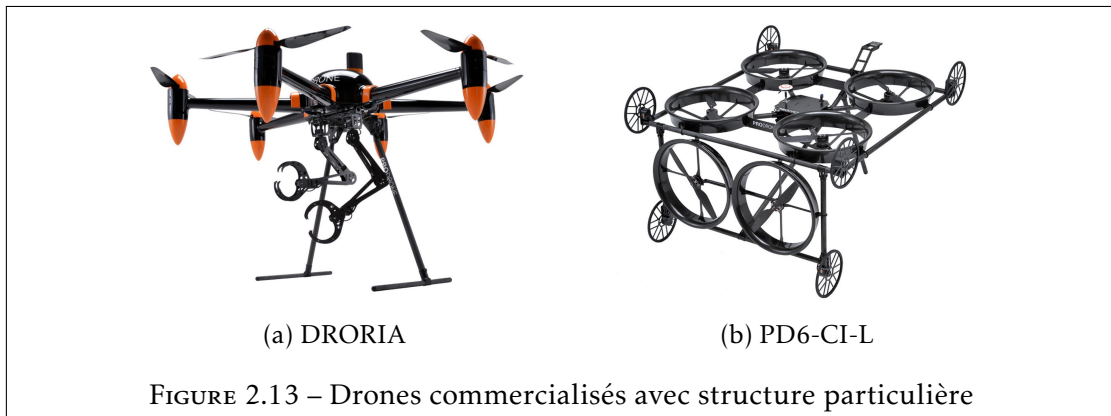


FIGURE 2.12 – Prototypes pour manipulations aériennes

Si l'on concatène les deux solutions précédentes, on obtiens un système avec des hélices qui ne sont pas co-planaires et qui utilise un bras manipulateur. [21] propose un système totalement actionné avec des bras inclinés de ce type. [22] aborde une approche similaire en développant une plateforme industrielle suspendue (par hélicoptère ou grue) ayant également des hélices inclinées et un bras manipulateur. Dans cette configuration, il est certes possible de se déplacer librement sur tous les DDL, mais dans certaines configurations, les poussées du drone ne seront pas optimisées.

12. <https://aeroarms-project.eu/>

Certaines solutions sont déjà commercialisées, on pourra citer l'entreprise ProDrone® qui propose deux produits intéressants. Le premier, "DRORIA" est un drone pouvant soulever une lourde charge utile à l'aide de bras manipulateurs, voir Figure 2.13a. L'autre, "PD6-CI-L", Figure 2.13b, permet un vol le long d'un mur vertical. Deux hélices verticales permettent de coller le drone directement sur la paroi et des roues permettent de le faire rouler lentement dessus, voir Figure 2.13.



On présente également quelques véhicules multi-rotors avec Moteurs Inclinaables (MI), ce qui permet de changer l'orientation de l'aéronef pour pouvoir se coller sur la paroi. L'étude [23] propose un hexacoptère avec bras inclinables, illustré en Figure 2.14. Deux quadricoptère également à bras inclinables sont décrits dans [24] ainsi que [25] qui présente un drone volant à la verticale. Ces exemples montrent la faisabilité du projet, mais laissent un champ de recherche possible, notamment sur l'optimisation mécanique, la commande et l'adaptation du drone pour une application spécifique. Deux solutions



distinctes se dessinent. D'une part un drone avec bras inclinables et de l'autre un drone équipé d'un bras manipulateur.

### 2.4.3 Approches possibles et choix de solution

Le choix ambitieux d'utiliser le drone comme vecteur, ouvre plusieurs questions secondaires. Comme cela a été vu dans la section précédente, aucun drone disponible sur le marché ne répond spécifiquement à cette problématique. Il faudra donc développer un système spécifique à l'application. De plus, afin de répondre aux différentes contraintes, on souhaite avoir un système robuste mécaniquement, et avec une commande performante. Dans tous les cas, le prototype doit absolument garantir sa fonction principale.

Différentes manières d'aborder les problématiques sont possibles. Le réel challenge consiste à pouvoir placer un capteur onéreux sur n'importe quelle surface et en toute sécurité. Deux approches différentes se distinguent et permettraient de résoudre le problème :

- (A) l'outil de mesure est agrippé par un bras mécanisé lui-même raccordé au drone ;
- (B) l'outil de mesure est inséré dans le cœur du drone. C'est le changement de sa configuration de vol qui permettra le plaquage de l'outil sur la paroi.

Dans les deux cas, cette approche implique un challenge important sur la commande du drone, notamment du fait du changement de centre de gravité. On résume dans le Tableau 2.3 le niveau de réponse des deux solutions face aux contraintes du système.

| Solution    | FC1 | FC2 | FC3 | FC4 | FC5 | FC6 | FC7 | FC8 | Total |
|-------------|-----|-----|-----|-----|-----|-----|-----|-----|-------|
| Coefficient | (2) | (2) | (3) | (3) | (2) | (2) | (1) | (2) |       |
| (A)         | 0   | 0   | 1   | 0   | 0   | 0   | 0   | 1   | 5     |
| (B)         | 0   | 1   | 0   | 1   | 0   | 0   | 1   | 0   | 7     |

TABLEAU 2.3 – Classification des solutions technologiques

L'avantage de la technique du bras manipulateur est la possibilité d'utiliser un drone du commerce en ajustant son auto-pilote. Le défi mécanique est plus simple sur le drone, mais pas sur la préhension du capteur. L'autonomie (FC1) des deux solutions est quasiment la même. la robustesse (FC2) sera moins grande que sur un drone directement plaqué sur la surface, en effet, un système de propulsion du drone qui changera sa configuration de vol limitera l'arrachement de la paroi ou le glissement du robot. Concernant la sécurité (FC3), un drone du commerce qu'implique la solution (A) sera nécessairement plus sûr qu'un prototype, il en va de même pour sa facilité d'utilisation (FC8). Le bras manipulateur ne pourra pas forcément accéder à toutes les surfaces (FC4), le drone pouvant faire obstruction. Pour l'encombrement (FC5), les deux types de drones se valent. Il en va de même pour la versatilité d'utilisation du dispositif (FC6). Pour l'esthétique (FC7), un système de bras inclinables semble plus impressionnant visuellement qu'un drone classique avec bras.

Dans tout les cas il faudra changer la loi de commande du système pour pouvoir le piloter. Ce n'est donc pas un critère à prendre en compte. Enfin, on précise un inconvénient supplémentaire pour la première solution proposée. Le système possédera un porte-à-faux dynamique important. Celui-ci implique une perte d'autonomie et d'efficacité qu'il faudra compenser d'une manière ou d'une autre.

Suite à ce comparatif, la réponse la plus adaptée au besoin semble être le changement de configuration. C'est donc cette solution qui sera retenue plutôt que la commande d'un drone équipé d'un bras robotisé. L'Annexe B présente l'AMDEC du système. Cette analyse permet de mettre en avant nombre de détails à considérer pendant la conception mécanique comme informatique. Les points les plus critiques qui découlent de cette analyse sont surtout la sécurité des tiers et de l'outil de mesure.

Répondre à ces critères sera une première étape pour utiliser le prototype en production. La sécurité des personnes pourra se faire en ajoutant un parachute alors que l'outil pourra être protégé par un carénage conséquent. Des systèmes de protection pour la batterie devront également être considérés.



La robustesse face aux rafales sort également comme un élément important et sera traité dans le manuscrit. Les perturbations magnétiques ne seront pas un problème comme explicité plus loin car on ne considèrera pas le magnétomètre.

On note que tous les points du cahier des charges ne pourront pas être traités dans cet ouvrage. Cependant on montrera dans les sections suivantes, et notamment dans le Chapitre 6, les résultats obtenus avec le drone conçu et développé. Un développement supplémentaire du système permettra de l'utiliser sur chantier en répondant aux contraintes non remplies.

## 2.5 Matériel et méthode

Dans cette section on décrit d'une part le matériel qui sera utilisé dans le travail de la thèse. Cela permettra à un utilisateur futur de pouvoir reproduire ces résultats dans les mêmes conditions. D'autre part on décrit la méthodologie appliquée pour répondre aux problématiques. Cela permettra également de mieux expliquer la structure du manuscrit en décrivant le cheminement de la réflexion qui a été faite.

Afin de pouvoir mener à bien ce projet, l'entreprise a sollicité un laboratoire spécialisé dans le domaine des drones. La section suivante présente ce laboratoire ainsi que les spécialités qu'il aborde.

### 2.5.1 Présentation du laboratoire

Le laboratoire Heuristique et Diagnostique des Systèmes Complexes (HEUDIASYC) touche à de nombreux et différents domaines.

Parmi ceux-ci on retrouve :

- Les véhicules intelligents
  1. L'aide à la prise de décision
  2. La conduite autonome
  3. La communication entre les véhicules
- le ferroviaire
  1. Logiciels de supervision
  2. Uniformisation de la signalisation entre les pays
- Réseaux de capteurs
- Logistique de transports
- Analyse et gestion de données
- Apprentissage statistique
- Les mini drones (dont cette thèse fait partie)
  1. Développement de plateforme
  2. Tests d'algorithmes de commande
  3. Test de la tolérance aux fautes
  4. Flotte de drones, voir Figure 2.5.1

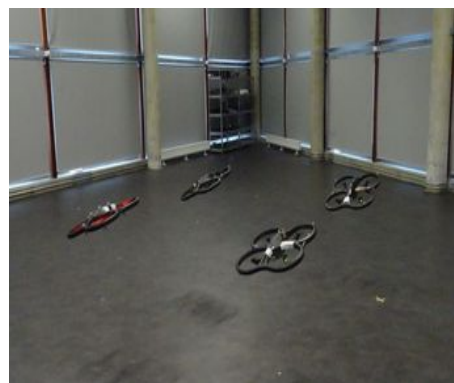


FIGURE 2.15 – Démonstration avec une flotte de drones dans la volière

### 2.5.2 Gestion du projet

Suite au choix de système décrit plus haut, on décompose le problème en plusieurs étapes. L'étude va être scindée en trois axes principaux menant finalement à la réalisation et aux résultats en simulations et expérimentations. Le premier axe consiste à développer la modélisation mécanique du système. Cela permet de le comprendre d'une part, et d'autre part de pouvoir proposer une loi de commande pour le contrôler. Cette première phase permettra donc d'orienter la deuxième partie de l'analyse qui consiste à proposer une loi de commande au système. Le couple modèle+commande doit ensuite être validé en simulation, troisième partie de l'étude. Cette dernière partie englobe à la fois le développement d'un simulateur mais également le développement d'une plateforme complète permettant le test en temps réel. La plateforme représente le travail sur le plan informatique mais aussi sur la conception mécanique des prototypes. Les objectifs prévisionnels du projet avaient été définis pour les 3 années. Dans le Tableau 2.4, on présente le planning initial établi en début de projet.

| Première année            | Deuxième année               | Troisième année            |
|---------------------------|------------------------------|----------------------------|
| État de l'art             | Premier tests                | Optimisation du système    |
| Analyse PVTOL-MI          | Ajustements lois de commande | Rédaction du mémoire       |
| Réalisation premier robot | Analyse de l'approche murale | Tests en extérieur         |
| Analyse Quadricoptère-MI  | Conception du porte outil    | Réalisation du robot final |

TABLEAU 2.4 – Échéance des tâches à réaliser au cours de la thèse

Par la suite on présente en Tableau 2.5, le planning effectif. La différence se fait principalement sur le développement de la plateforme informatique qui a pris beaucoup de temps en deuxième année. Cette partie décrite dans le Chapitre 5.2 a permis deux choses essentielles. Premièrement, la compréhension très bas niveau du dispositif, permettant d'obtenir de solides connaissances en robotique de manière générale. Deuxièmement, l'obtention d'un système extrêmement simple d'utilisation ouvert à beaucoup plus de personnes que les systèmes classiques, y compris des novices en matière de programmation.

| Première année            | Deuxième année            | Troisième année                    |
|---------------------------|---------------------------|------------------------------------|
| État de l'art             | Analyse Quadcopter-MI     | Tests différentes lois de commande |
| Analyse PVTOL-MI          | Réalisation premier robot | Compensation effets gyroscopiques  |
| Projets en entreprise     | Conception deuxième robot | Tests du deuxième prototype        |
| Conception premier robot  | Plateforme informatique   | Rédaction du mémoire               |
| Création du simulateur    | Premier tests             | Analyse de l'approche murale       |
| Rédaction et publications |                           |                                    |

TABLEAU 2.5 – Échéance des tâches à réaliser au cours de la thèse

Dans les sections suivantes on présente la plupart des outils utilisés au cours de la thèse. On détaillera ces éléments dans les différents chapitres qui suivent.

### 2.5.3 Modélisation

Dans un premier temps on choisit de faire une étude sur le modèle du PVTOL, mais en considérant des propulsions inclinables. Cela permet de modéliser une version simplifiée du système complet. D'une part, ce système largement exploité et présent dans la littérature, a été choisi car il permet de comprendre la dynamique affectant le drone. D'autre part, il permet aussi de faciliter le choix et la validation d'une loi de commande adaptée. Le système ne présente que 3 DDL car il est représenté dans un Plan 2D. Un rappel du système 3D classique sera fait afin de poser toutes les variables et rappeler certains des effets mécaniques du système. Une analyse du système libre d'évoluer sur 6 DDL sera faite par la suite. Celle-ci sera cruciale pour comprendre le système, le commander et enfin le concevoir.

Trois configurations différentes seront détaillées montrant la démarche pour finalement parvenir au prototype final qui permettra de réaliser l'objectif fixé plus haut. Une comparaison énergétique des différentes configurations sera également faite pour permettre d'apprécier celle qui sera la plus adaptée au problème. Certains effets supplémentaires seront également détaillés dans ce chapitre (comme le couple gyroscopique des hélices par exemple).

### 2.5.4 Loi de commande

Une fois la modélisation des différentes configurations effectuée, on propose de répondre à la deuxième problématique<sup>13</sup>. Pour se faire on choisit de tester un algorithme de commande par linéarisation entrée-états permettant d'obtenir un système linéaire. On pourra alors tester différentes lois de commande pour répondre aux contraintes énoncées ci-dessus. En commençant par une approche linéaire, simple, on détaillera ensuite deux approches non linéaires.

Parmi ces deux dernières lois, une loi de commande robuste permettant d'offrir une résistance aux perturbations extérieures sera proposée. Les capteurs actuels ne peuvent pas directement mesurer toutes les grandeurs dont on pourrait avoir besoin pour les calculs de la loi de commande. On propose donc de détailler la question de l'observation des états. Une analyse des points de singularité sera faite et une solution pour les limiter sera proposée. Il en sera fait de même pour les balourds liés au rajout du capteur.

### 2.5.5 Plateforme

#### 2.5.5.1 Conception mécanique

La configuration retenue demande une conception mécanique particulière. Comme cela a été évoqué précédemment, le drone sera équipé de bras inclinables. Cette modification demande donc, soit de repenser intégralement un drone, soit de modifier un modèle existant pour obtenir les résultats attendus. Dans tout les cas il faudra pouvoir réaliser des changements structurels et fonctionnels. Pour cela, on choisit d'utiliser le logiciel Autodesk Inventor pour la partie Conception Assistée par Ordinateur (CAO). Celui-ci permettra de créer des fichiers qui pourront ensuite être imprimés en 3D.

Les différentes fournitures seront achetées soit en magasin spécialisé, soit commandées en ligne. Pour les fournisseurs on pourra citer Radiospares<sup>TM</sup>, Studiosport© ou encore Génération Robots© et GoTronic©.

---

13. Comment sera commandé ce robot ?

Concernant l'impression, le laboratoire dispose de 3 types différents d'imprimantes 3D. La première, le modèle S5 d'Ultimaker®, permettra de faire des pièces rapidement par dépôt de fil extrêmement qualitatif. Les deux autres seront moins utilisés. Une imprimante à résine (Form 2 de Formlabs©) pour des pièces esthétiques et finalement une imprimante carbone (MarkTwo de Markforged®) permettront d'offrir des possibilités de prototypage intéressantes. La formation à ces techniques de prototypage rapides fait partie du travail effectué au cours de cette thèse.

### 2.5.5.2 Système embarqué

Pour le système embarqué, on choisit chaque élément avec soin, car cela aura de grandes répercussions sur le déroulement de la thèse (surtout au milieu et à la fin) mais aussi sur les performances du drone. Pour l'implémentation sur microcontrôleur, on choisit un système Raspberry Pi<sup>TM</sup> avec une extension NAVIO2<sup>TM</sup>. Ce système permettra d'accueillir le programme de commande d'une part, mais offre également bon nombre de capteurs qui pourront être utiles par la suite (*Inertial Measurement Unit* (IMU), Système de Géo-positionnement par satellite (*Global Positioning System*) (GPS), Entrées / Sorties (E/S) PWM, ...).

Les actionneurs seront plus largement décrits dans la partie sur la conception mécanique. En ce qui concerne les moteurs, on cherche à les dimensionner de façon à ce qu'ils puissent soulever le drone. Dans le cas présent on reprendra les T-Motors du drone F450 de DJI©. Le châssis du drone sera également la base structurelle du premier prototype. Pour aider dans le choix du moteur, on pourra également utiliser des sites comme [26], par exemple. Enfin, il faudra dimensionner le système d'alimentation. Celui-ci prendra d'abord la forme de batteries LiPo qui sont, à l'heure actuelle, le moyen qui offre le meilleur rapport poids puissance. Par la suite on pourra envisager d'utiliser des solutions d'alimentation filaire qui permettront une autonomie "illimitée" comme vendu par Elistair© [27].

### 2.5.6 Résultats

Un chapitre entier est dédié aux résultats obtenus aussi bien en simulation que durant les expérimentations en volière. Celle-ci, présente au Laboratoire permet d'effectuer des tests en ayant un grand espace équipé d'un système de *motion control*. Un certain nombre de matelas permet également de limiter l'endommagement du système pendant les phases de tests.

Les tests se font de manière successives en travaillant étape par étape. Après une grande phase de test en simulation, le principe consiste à avoir une commande sur la télécommande permettant de tester un point en gardant un moyen de revenir rapidement à la dernière étape stable utilisée. En lisant le chapitre on se rend réellement compte de cette progression en partant d'un vol quasi classique pour aboutir finalement à un vol où le drone est incliné mais fixe linéairement dans l'espace.

## 2.6 Conclusion

Ce premier chapitre permet de donner des éléments de réponse par rapport à la première problématique<sup>14</sup>. C'est donc un drone à bras inclinables qui est choisi pour y répondre. Cette décision a permis de structurer le travail de recherche par domaine mais également en étapes sur les trois ans du projet. Le chapitre suivant permettra d'analyser différents types de configurations de drones à bras inclinables. Cette étude permettra de statuer sur la configuration la plus appropriée pour atteindre l'objectif<sup>15</sup>. Cela permettra de compléter les réponses pour la première problématique et permettra aux chapitres suivants de donner des éléments de réponse aux trois autres problématiques posées dans l'avant propos.

---

14. Quelle forme doit avoir le robot qui effectue les mesures sur ouvrage

15. Rendre possible l'inspection d'ouvrages d'art dans des zones difficilement accessibles

## Modélisation

Dans ce chapitre, une modélisation mécanique de différentes configurations de véhicules types drone est présentée. Dans un premier temps on va étudier le PVTOL simple et avec bras inclinables. Cela permet d'avoir une première analyse en 2D du système que l'on se propose de tester. Par la suite, on propose une analyse tri-dimensionnelle. Un rappel sur la configuration classique "X4" sans bras inclinable qui correspond à la plupart des drones du commerce est faite. On adapte ensuite le modèle pour arriver progressivement à la configuration qui sera utilisée et testée in fine. Cela amène progressivement le lecteur vers la configuration de drone à bras inclinables. Trois configurations seront proposées, celles-ci se complexifiant de plus en plus. Une analyse des couples gyroscopiques qui ont un fort impact dans notre étude est également présentée. Une comparaison de consommation d'énergie entre les différentes configurations conclura le chapitre.

### 3.1 PVTOL et PVTOL à bras inclinables

Pour l'analyse mécanique qui suit, on considère uniquement le formalisme de Newton-Euler [28], celui-ci paraissant plus intuitif pour formaliser les équations du système. Le même résultat peut cependant être retrouvé par le formalisme d'Euler-Lagrange [29]. L'expression générale de la dynamique de translation des forces dans le repère du corps du système est :

$$\sum \vec{F}_B = m \dot{\mathbf{V}}_B + \boldsymbol{\Omega} \times m \mathbf{V}_B \quad (3.1)$$

$$\dot{\boldsymbol{\xi}} = {}^I \mathcal{R}_B \mathbf{V}_B \quad (3.2)$$

Avec  $\vec{F}_B$  le vecteur des forces dans le corps,  $m$  la masse du système,  $\mathbf{V}_B = [u, v, w]^T$  le vecteur des vitesses linéaires du système dans le repère du corps.  ${}^B \mathcal{R}_I$  est la matrice de rotation entre le repère du corps et le repère inertiel.  $\boldsymbol{\xi}_B = [x, y, z]^T$  est le vecteur des positions du système dans le repère inertiel.  $\boldsymbol{\Omega}$  correspond à la vitesse angulaire du système dans le repère du corps.

Remarque : les vecteurs sont écrits en **gras** ; les matrices en écriture Caligraphiée , les états et constantes en *italique* et les fonctions avec des parenthèses(\*).

Pour la dynamique de rotation dans le système, l'expression générale dans le corps est la suivante :

$$\sum \vec{M}_B = J \dot{\boldsymbol{\Omega}} + \boldsymbol{\Omega} \times J \boldsymbol{\Omega} \quad (3.3)$$

Avec  $M$  les moments et  $J$  l'inertie du système. On précise que l'on considère que les deux termes liés aux effets d'inertie  $\Omega \times m \mathbf{V}_B$  et  $\Omega \times J \Omega$  n'ont pas d'impact dans l'analyse plane qui suit. Dans la littérature on utilise très fréquemment le PVTOL pour tester différentes lois de commandes. Relativement simple de principe, ce système permet de considérer un système non-linéaire qui ressemble beaucoup à un drone que l'on ferait évoluer dans un plan.

Aidé par la littérature (dont on ne citera ici que [30], [31]), on retrouve rapidement les équations mécaniques du système PVTOL classique, représenté en Figure 3.1, ce qui donnera une base pour la suite. L'expression du système (en considérant que les moteurs sont parallèles et orthogonaux au support) est la suivante (cf. [32]) :

$$\begin{aligned} m\ddot{x} &= \sin(\phi)(T_1 + T_2) \\ m\ddot{y} &= \cos(\phi)(T_1 + T_2) - mg \\ J\ddot{\phi} &= L(T_2 - T_1) \end{aligned} \quad (3.4)$$

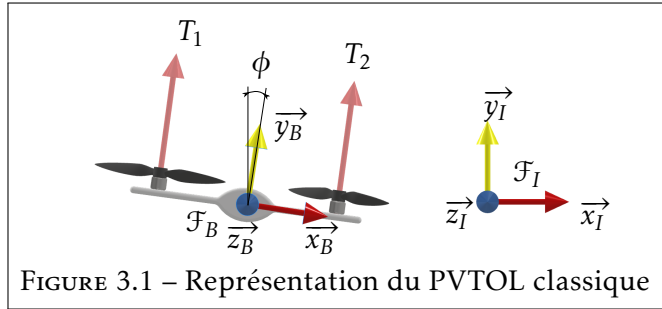


FIGURE 3.1 – Représentation du PVTOL classique

$\phi$  est l'état angulaire du système. Avec  $T_i$  la poussée exercée par le  $i^{\text{ème}}$  moteur et  $g$  l'accélération gravitationnelle. Soit  $L$  la distance horizontale entre le centre de masse du corps et l'origine de la poussée des moteurs.

### 3.1.1 Modélisation mécanique PVTOL-MI

Notre problématique nous conduit à réaliser la première étude sur un système PVTOL mais adaptée à notre besoin. On s'inspirera donc des travaux cités ci-dessus pour arriver au modèle mécanique du PVTOL avec des MI. L'étude qui suit ne décrit que la partie mécanique. La commande et les résultats de simulations viendront dans les chapitres suivants. Le changement mécanique que propose cette thèse implique également certains changements dans les équations. Généralement on considère l'inclinaison des moteurs comme étant fixe. Ici, il faudra prendre en compte cette inclinaison comme étant variable, et trouver des solutions pour commander le système. On présente le système PVTOL à Moteurs Inclinaibles (PVTOL-MI) en Figure 3.2.

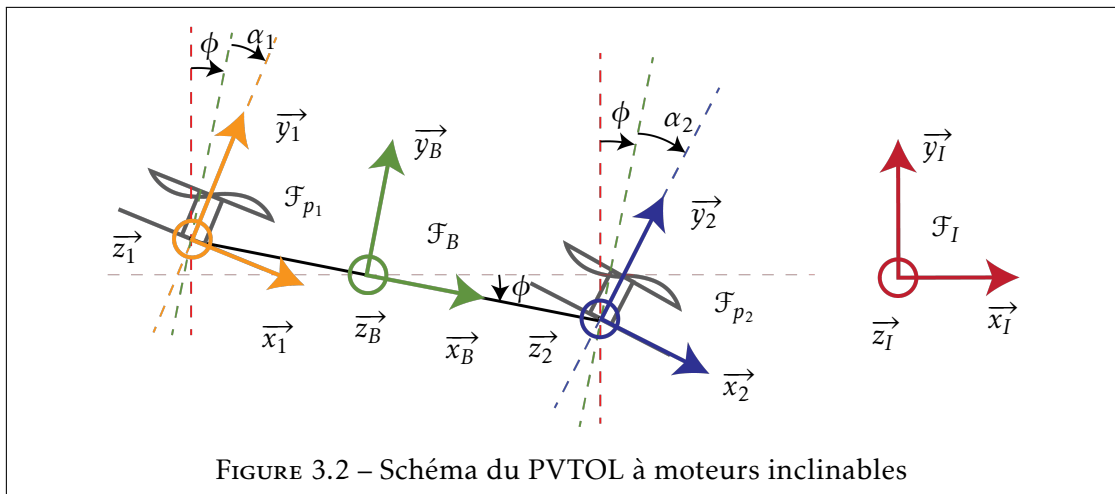


FIGURE 3.2 – Schéma du PVTOL à moteurs inclinables

Les différents éléments utiles à la modélisation mécanique du problème sont récapitulés dans le Tableau 3.1.

Ces variables sont définies au fur et à mesure, au moment le plus opportun. On note  $I$  l'indice correspondant au repère *inertiel*,  $B$  celui du corps (*Body*) du drone,  $a$  celui du bras (*arm*) et enfin  $p$  celui de l'hélice (*propeller*).

| Paramètre                       | Description   | Unité                         |
|---------------------------------|---|-------------------------------|
| $T_i$                           | Résultante de force due à la $i^{\text{ème}}$ hélice                    | $N$                           |
| $\mathcal{F}_I$                 | Repère initial  | -                             |
| $\mathcal{F}_B$                 | Repère du corps (body) du drone   | -                             |
| $\mathcal{F}_{p_i}$             | Repère du $i^{\text{ème}}$ moteur ( <i>propeller</i> ) aligné sur $T_i$ | -                             |
| $\phi, \dot{\phi}, \ddot{\phi}$ | Angle entre $\mathcal{F}_I$ et $\mathcal{F}_B$ et dérivées              | $rad, rad s^{-1}, rad s^{-2}$ |
| $\alpha_1$                      | Angle entre $\mathcal{F}_B$ et $\mathcal{F}_{p_1}$                      | $rad$                         |
| $\alpha_2$                      | Angle entre $\mathcal{F}_B$ et $\mathcal{F}_{p_2}$                      | $rad$                         |
| ${}^I\mathcal{R}_B$             | Matrice de rotation entre $\mathcal{F}_I$ et $\mathcal{F}_B$            | -                             |
| ${}^B\mathcal{R}_{p_i}$         | Matrice de rotation entre $\mathcal{F}_B$ et $\mathcal{F}_{p_i}$        | -                             |
| $h$                             | Distance entre la base des moteurs et l'origine de $T_i$                | $m$                           |
| $L$                             | Distance entre l'axe $\vec{y}_B$ et la base des moteurs                 | $m$                           |
| $m$                             | Masse du véhicule   | $kg$                          |
| $g$                             | L'accélération due à la gravité   | $m s^{-2}$                    |
| $J$                             | Moment d'inertie total du véhicule                                      | $kg m^2$                      |
| $x, \dot{x}, \ddot{x}$          | États du PVTOL-MI sur l'axe $\vec{x}_0$ et dérivées                     | $m, m s^{-1}, m s^{-2}$       |
| $y, \dot{y}, \ddot{y}$          | États du PVTOL-MI sur l'axe $\vec{y}_0$ et dérivées                     | $m, m s^{-1}, m s^{-2}$       |

TABLEAU 3.1 – Définitions et noms des paramètres

Pour la suite, on va être intéressé par l'expression des efforts suivants les axes  $\vec{x}_I$   $\vec{y}_I$  ainsi que par la rotation suivant l'axe  $\vec{z}_I$ . La force 1 ( $T_1$ ) correspond à la force créée par le moteur 1 dans le sens opposé au flux d'air. De même, la force 2 ( $T_2$ ) est créée à droite sur le deuxième moteur. Pour mieux visualiser les efforts en présence, on propose la Figure 3.3.

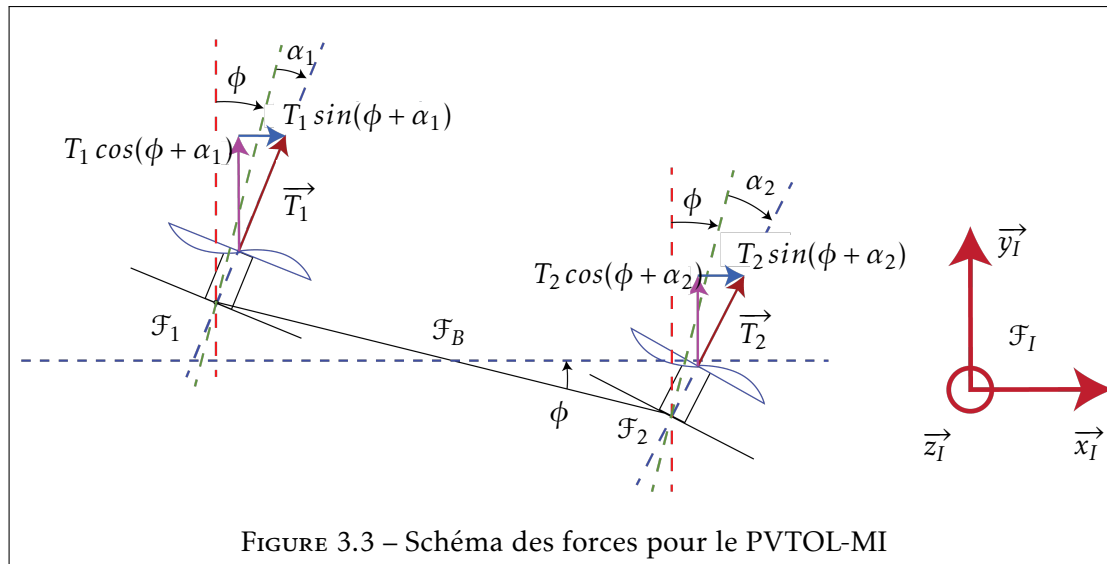


FIGURE 3.3 – Schéma des forces pour le PVTOL-MI

Les efforts suivants l'axe  $\vec{x}_I$  sont décrits en bleu et ceux suivant  $\vec{y}_I$  sont en magenta. Sur l'axe  $\vec{x}_I$  on a les efforts suivants :

$$\begin{aligned}
 m \ddot{x}_I &= T_1 \sin(\alpha_1 + \phi) + T_2 \sin(\alpha_2 + \phi) \\
 &= T_1 (\sin(\phi) \cos(\alpha_1) + \sin(\alpha_1) \cos(\phi)) + T_2 (\sin(\phi) \cos(\alpha_2) + \sin(\alpha_2) \cos(\phi)) \quad (3.5)
 \end{aligned}$$



En factorisant par  $\sin(\phi)$  et  $\cos(\phi)$  on obtient :

$$m\ddot{x}_I = \sin(\phi)(T_1 \cos(\alpha_1) + T_2 \cos(\alpha_2)) + \cos(\phi)(T_1 \sin(\alpha_1) + T_2 \sin(\alpha_2)) \quad (3.6)$$

Sur l'axe  $\vec{y}_I$  :

$$\begin{aligned} m\ddot{y}_I &= T_1 \cos(\alpha_1 + \phi) + T_2 \cos(\alpha_2 + \phi) - mg \\ &= T_1 (\cos(\phi) \cos(\alpha_1) - \sin(\alpha_1) \sin(\phi)) + T_2 (\cos(\phi) \cos(\alpha_2) - \sin(\alpha_2) \sin(\phi)) - mg \end{aligned} \quad (3.7)$$

En factorisant par  $\sin(\phi)$  et  $\cos(\phi)$  on obtient :

$$m\ddot{y}_I = \cos(\phi)(T_1 \cos(\alpha_1) + T_2 \cos(\alpha_2)) - \sin(\phi)(T_1 \sin(\alpha_1) + T_2 \sin(\alpha_2)) - mg \quad (3.8)$$

On peut également retrouver ces calculs en utilisant des matrices de rotations. Pour cela, on définit la matrice de rotation entre le repère  $\mathcal{F}_I$  et  $\mathcal{F}_B$  que l'on appelle  ${}^I\mathcal{R}_B$ . On remarque que la matrice a la même forme entre les repères  $\mathcal{F}_B$  et  $\mathcal{F}_{p_1}$  ainsi que  $\mathcal{F}_B$  et  $\mathcal{F}_{p_2}$  que l'on appelle respectivement  ${}^B\mathcal{R}_{p_1}$  et  ${}^B\mathcal{R}_{p_2}$  et que l'on définit de la manière suivante :

$${}^I\mathcal{R}_B = \begin{bmatrix} \cos(\phi) & \sin(\phi) \\ -\sin(\phi) & \cos(\phi) \end{bmatrix} \quad {}^B\mathcal{R}_{p_1} = \begin{bmatrix} \cos(\alpha_1) & \sin(\alpha_1) \\ -\sin(\alpha_1) & \cos(\alpha_1) \end{bmatrix} \quad {}^B\mathcal{R}_{p_2} = \begin{bmatrix} \cos(\alpha_2) & \sin(\alpha_2) \\ -\sin(\alpha_2) & \cos(\alpha_2) \end{bmatrix} \quad (3.9)$$

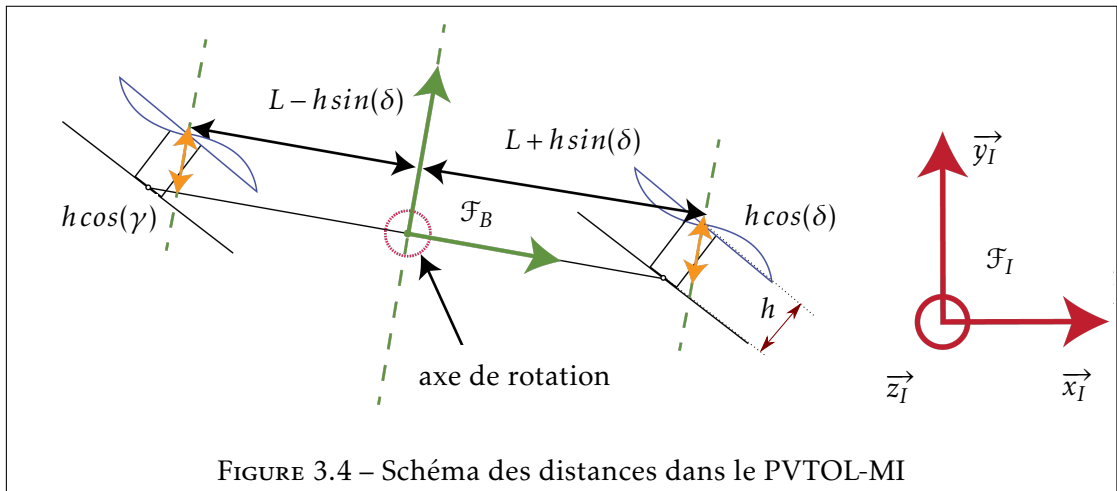
Appliquons maintenant ces matrices aux forces  $T_1$  et  $T_2$  qui sont toutes les deux sur les axes  $\vec{y}_1$  et  $\vec{y}_2$  respectivement. On cherche les efforts dans le repère  $\mathcal{F}_I$ . Les abréviations  $s()$  et  $c()$  correspondent au sinus et au cosinus.

$$\begin{aligned} m \begin{bmatrix} \ddot{x}_I \\ \ddot{y}_I \end{bmatrix} &= \begin{bmatrix} \cos(\phi) & \sin(\phi) \\ -\sin(\phi) & \cos(\phi) \end{bmatrix} \begin{bmatrix} \cos(\alpha_1) & \sin(\alpha_1) \\ -\sin(\alpha_1) & \cos(\alpha_1) \end{bmatrix} \begin{bmatrix} 0 \\ T_1 \end{bmatrix} \\ &+ \begin{bmatrix} \cos(\phi) & \sin(\phi) \\ -\sin(\phi) & \cos(\phi) \end{bmatrix} \begin{bmatrix} \cos(\alpha_2) & \sin(\alpha_2) \\ -\sin(\alpha_2) & \cos(\alpha_2) \end{bmatrix} \begin{bmatrix} 0 \\ T_2 \end{bmatrix} - \begin{bmatrix} 0 \\ mg \end{bmatrix} \end{aligned} \quad (3.10)$$

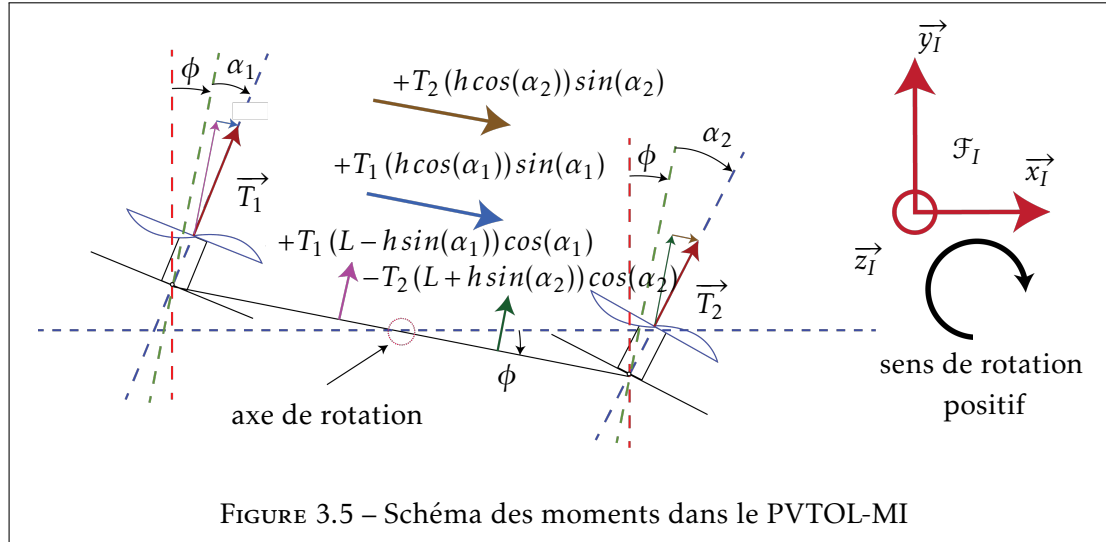
Ce qui donne, quand on ne considère que les deuxièmes colonnes :

$$m \begin{bmatrix} \ddot{x}_I \\ \ddot{y}_I \end{bmatrix} = \begin{bmatrix} c(\phi)s(\alpha_1) + s(\phi)c(\alpha_1) \\ -s(\phi)s(\alpha_1) + c(\phi)c(\alpha_1) \end{bmatrix} T_1 + \begin{bmatrix} c(\phi)s(\alpha_2) + s(\phi)c(\alpha_2) \\ -s(\phi)s(\alpha_2) + c(\phi)c(\alpha_2) \end{bmatrix} T_2 - \begin{bmatrix} 0 \\ mg \end{bmatrix} \quad (3.11)$$

On remarque que l'on retrouve bien les mêmes équations qu'en (3.6) et (3.8).



Les angles et la distance  $L$  ont été définis précédemment. On constate cependant un problème : que deviennent les projections de ces longueurs dans le repère du corps quand on incline les moteurs ? La Figure 3.4 permet d'exprimer ces distances projetées dans  $\mathcal{F}_B$ . Soit  $h$  la distance verticale entre le centre de masse du véhicule et l'origine de la poussée des moteurs, quand ils sont à leur position initiale. Sur l'axe  $\vec{z}_B$ , on peut maintenant calculer les efforts de couple liés à la poussée des hélices et obtenir l'expression de l'accélération angulaire  $\ddot{\phi}$ . On les représente sur la Figure 3.5.



On calcule le moment résultant au centre de gravité :

$$J\ddot{\phi} = (L - h \sin(\alpha_1)) T_1 \cos(\alpha_1) + T_1 h \cos(\alpha_1) \sin(\alpha_1) - T_2 (L + h \sin(\alpha_2)) \cos(\alpha_2) + T_2 h \cos(\alpha_2) \sin(\alpha_2) \quad (3.12)$$

En développant on obtient :

$$\begin{aligned} J\ddot{\phi} &= T_1 L \cos(\alpha_1) - \cancel{T_1 h \cos(\alpha_1) \sin(\alpha_1)} + \cancel{T_1 h \cos(\alpha_1) \sin(\alpha_1)} - T_2 L \cos(\alpha_2) \\ &\quad - \cancel{T_2 h \cos(\alpha_2) \sin(\alpha_2)} + \cancel{T_2 h \cos(\alpha_2) \sin(\alpha_2)} \\ J\ddot{\phi} &= T_1 L \cos(\alpha_1) - T_2 L \cos(\alpha_2) \end{aligned} \quad (3.13)$$

L'équation mécanique du système dans le repère initial s'écrit sous la forme suivante :

$$\begin{bmatrix} m \ddot{x}_I \\ m \ddot{y}_I \\ J \ddot{\phi} \end{bmatrix} = \begin{bmatrix} c(\phi)s(\alpha_1) + s(\phi)c(\alpha_1) \\ -s(\phi)s(\alpha_1) + c(\phi)c(\alpha_1) \\ Lc(\alpha_1) \end{bmatrix} T_1 + \begin{bmatrix} c(\phi)s(\alpha_2) + s(\phi)c(\alpha_2) \\ -s(\phi)s(\alpha_2) + c(\phi)c(\alpha_2) \\ -Lc(\alpha_2) \end{bmatrix} T_2 - \begin{bmatrix} 0 \\ mg \\ 0 \end{bmatrix} \quad (3.14)$$

On définit les variables suivantes :

$$\begin{aligned} u_1 &\triangleq T_1 \cos(\alpha_1) + T_2 \cos(\alpha_2) \\ u_2 &\triangleq T_1 \sin(\alpha_1) + T_2 \sin(\alpha_2) \\ u_3 &\triangleq T_1 \cos(\alpha_1) - T_2 \cos(\alpha_2) \end{aligned} \quad (3.15)$$

D'une part, ces équivalences simplifieront les calculs par la suite, mais d'autre part elles représentent réellement les actionnements du PVTOL-MI dans son repère propre. On peut en effet représenter ces équations comme étant les forces permettant le déplacement de l'objet volant. On retrouve cette représentation en Figure 3.6.

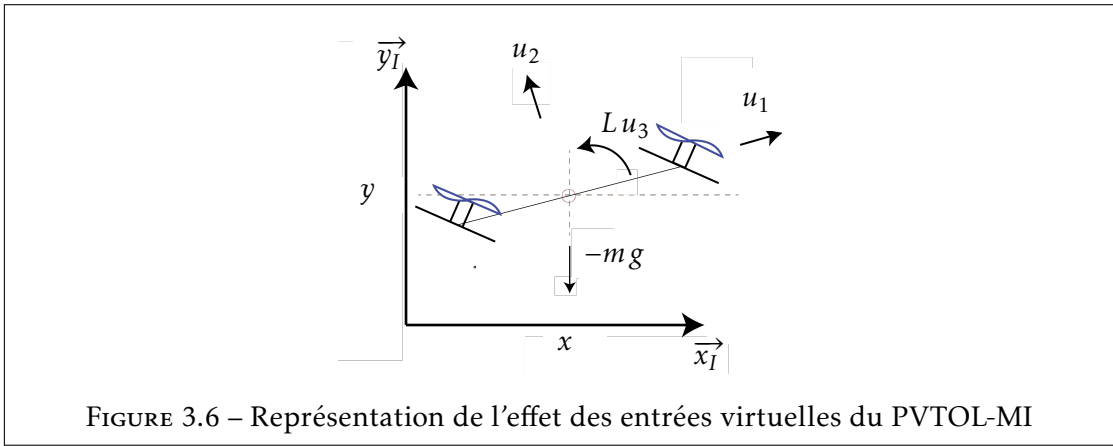


FIGURE 3.6 – Représentation de l'effet des entrées virtuelles du PVTOL-MI

Si on résume sous forme matricielle, on obtient pour le système l'équation mécanique suivante :

$$\begin{bmatrix} m \ddot{x}_I \\ m \ddot{y}_I \\ J \ddot{\phi} \end{bmatrix} = \mathcal{O} \begin{bmatrix} u_1 \\ u_2 \\ u_3 \end{bmatrix} - \begin{bmatrix} 0 \\ mg \\ 0 \end{bmatrix} \quad (3.16)$$

Avec  $\mathcal{O}$  la matrice suivante :

$$\mathcal{O} = \begin{bmatrix} \sin(\phi) & \cos(\phi) & 0 \\ \cos(\phi) & -\sin(\phi) & 0 \\ 0 & 0 & L \end{bmatrix} \quad (3.17)$$

Remarque : Si on compare cette modélisation mathématique à celle du PVTOL standard, on reprend les équations (3.15) avec  $\alpha_1 = 0$  et  $\alpha_2 = 0$ . Cela donne :

$$\begin{aligned} u_1 &\triangleq T_1 + T_2 \\ u_2 &\triangleq 0 \\ u_3 &\triangleq T_1 - T_2 \end{aligned} \quad (3.18)$$

Ce qui nous donne :

$$\begin{aligned} J \ddot{\phi} &= L(T_1 - T_2) \\ m \ddot{x}_I &= \sin(\phi)(T_1 + T_2) \\ m \ddot{y}_I &= \cos(\phi)(T_1 + T_2) - mg \end{aligned} \quad (3.19)$$

Dans le cas particulier présenté dans [32] on a un angle  $\phi$  négatif par rapport à celui présenté dans ce document. Soit  $\phi'$  définit de sorte à ce que  $\phi' = -\phi$  (par rapport au  $\phi$  de l'article), si on reprend nos calculs, on obtient donc :

$$\begin{aligned} J \ddot{\phi}' &= L(T_2 - T_1) \\ m \ddot{x}_I &= -\sin(\phi')(T_1 + T_2) \\ m \ddot{y}_I &= \cos(\phi')(T_1 + T_2) - mg \end{aligned} \quad (3.20)$$

On retrouve donc bien les mêmes équations que pour le modèle non-linéaire du PVTOL de [32] si on remplace  $-\phi$  par l'angle de l'article.

La section suivante reprend les travaux exposés ci-dessus en les généralisant aux trois dimensions. Un premier rappel de la configuration classique est fait, définissant toutes les variables utiles pour l'analyse. Trois configurations différentes sont ensuite explicitées. Celles-ci reprennent le cheminement de la pensée fait pour finalement aboutir à la dernière configuration.

## 3.2 Définitions et rappels de modélisation mécanique

Dans cette section on introduit l'ensemble des variables utilisées pour la modélisation mécanique des différentes configurations dans un repère en trois dimensions. On définit également tous les éléments mathématiques et physiques qui sont utiles pour l'analyse. La première sous-section se concentrera sur le modèle classique du quadricoptère en forme X. L'étude se base notamment sur les travaux [30], [33]. On met en évidence les forces et moments qui sont à l'origine de la dynamique du modèle.

Dans la sous-section qui suivra, on définit les variables pour le système avec le groupe moto-propulseur inclinable qui seront utilisées par la suite. Ce sera également l'occasion de présenter plus en détails les trois configurations qui sont présentées dans ce document.

### 3.2.1 Modélisation configuration classique X4

#### 3.2.1.1 Définitions

Outre le rappel de la dynamique classique d'un drone en X4 (X pour la forme et 4 pour le nombre de moteurs), cette partie permet également de définir une grande partie des variables complémentaires qui seront utilisées dans les sections suivantes. On renvoie le lecteur aux ouvrages [34]-[38] qui présentent exhaustivement la mécanique d'un quadricoptère classique par une approche Newton-Euler mais également Euler-Lagrange. Soit le système présenté en Figure 3.7.

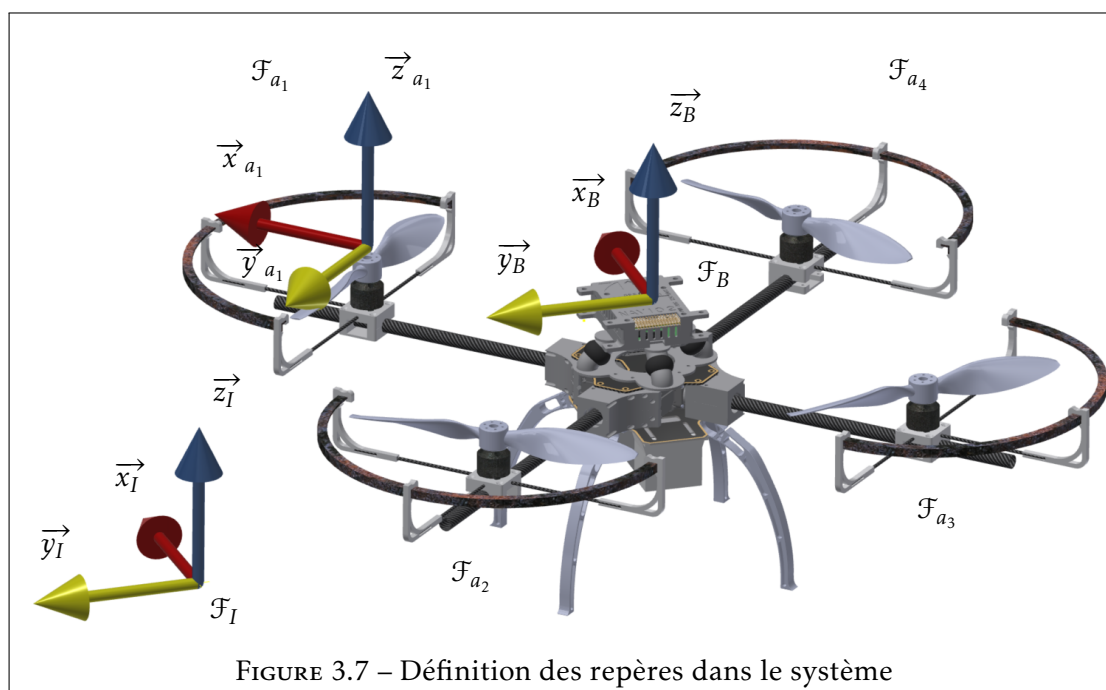


FIGURE 3.7 – Définition des repères dans le système

Celui-ci est composé de quatre moteurs avec hélices qui produiront une poussée verticale. On définit 3 repères dans le système. De manière analogue à la section précédente,  $\mathcal{F}_I$  correspond au repère inertiel. On définit ensuite le repère du corps qui est confondu avec le centre de masse du véhicule noté  $\mathcal{F}_B$ . Enfin, on définit le repère lié aux bras notés  $\mathcal{F}_{a_1 \rightarrow 4}$ . La plupart des variables étant déjà définies dans le Tableau 3.1, on complète celui-ci avec les variables dans le Tableau 3.2.

| Paramètres                    | Description   |
|-------------------------------|---|
| $\mathbf{T}$                  | Le vecteur des poussées des moteurs   |
| $\eta$                        | Angles d'Euler dans le repère inertiel  |
| $\omega$                      | Vecteur de rotation angulaire des hélices du drone                              |
| $\Omega$                      | Vecteur des vitesses angulaires du drone dans le repère du corps ( $p, q, r$ )  |
| ${}^a_i \mathcal{R}_{p_i}$    | Matrice de rotation entre le $i^{\text{ème}}$ bras et le $i^{\text{ème}}$ rotor |
| $J_a$                         | Moment d'inertie d'un bras ( <i>arm</i> ) du robot                              |
| $J_p$                         | Moment d'inertie d'un rotor ( <i>propeller</i> )                                |
| $\xi$                         | Vecteur de position du drone dans le repère initial                             |
| $k_f$                         | Coefficient de force de poussée du rotor  |
| $k_\tau$                      | Coefficient de couple de rotation du rotor                                      |
| $\delta = \frac{k_\tau}{k_f}$ | Rapport des coefficients précédents   |

TABLEAU 3.2 – Définition des variables

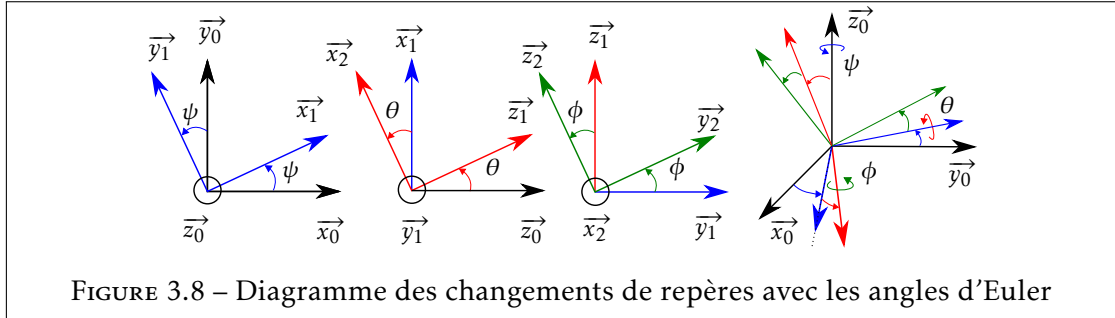


FIGURE 3.8 – Diagramme des changements de repères avec les angles d'Euler

On définit les rotations des angles d'Euler  $\phi$ ,  $\theta$  et  $\psi$  à partir de la Figure 3.8. Celles-ci permettent d'effectuer la rotation entre le repère initial et le repère du corps du drone. Cette matrice permettra d'exprimer la force de gravité (supposée verticale dans le repère inertiel) dans le repère du corps du drone. Plus de détails sont donnés dans [33] concernant la mécanique des véhicules terrestres et aériens (UAV).

$$\mathcal{R}_\psi = \begin{bmatrix} \cos(\psi) & -\sin(\psi) & 0 \\ \sin(\psi) & \cos(\psi) & 0 \\ 0 & 0 & 1 \end{bmatrix} \quad \mathcal{R}_\theta = \begin{bmatrix} \cos(\theta) & 0 & \sin(\theta) \\ 0 & 1 & 0 \\ -\sin(\theta) & 0 & \cos(\theta) \end{bmatrix} \quad \mathcal{R}_\phi = \begin{bmatrix} 1 & 0 & 0 \\ 0 & \cos(\phi) & -\sin(\phi) \\ 0 & \sin(\phi) & \cos(\phi) \end{bmatrix} \quad (3.21)$$

La matrice de rotation complète entre le repère inertiel et le repère du corps du drone s'écrit de la manière suivante :

$${}^I \mathcal{R}_B = \mathcal{R}_\psi \mathcal{R}_\phi \mathcal{R}_\theta = \begin{bmatrix} \cos(\psi)\cos(\theta) & -\cos(\phi)\sin(\psi) + \sin(\phi)\sin(\theta)\cos(\psi) & \sin(\psi)\sin(\phi) + \cos(\phi)\sin(\theta)\cos(\psi) \\ \sin(\psi)\cos(\theta) & \cos(\psi)\cos(\phi) + \sin(\psi)\sin(\theta)\sin(\phi) & -\sin(\phi)\cos(\psi) + \cos(\phi)\sin(\theta)\sin(\psi) \\ -\sin(\theta) & \cos(\theta)\sin(\phi) & \cos(\theta)\cos(\phi) \end{bmatrix} \quad (3.22)$$

On définit également la dérivée de la rotation entre le repère inertiel et du corps par l'équation suivante :

$${}^I \dot{\mathcal{R}}_B = {}^I \mathcal{R}_B \Omega^\times \quad (3.23)$$

Avec l'opérateur  $\text{vex} [\cdot]^\times$  basée sur la matrice antisymétrique définie comme :

$$x^\times = \begin{bmatrix} 0 & -x_3 & x_2 \\ x_3 & 0 & -x_1 \\ -x_2 & x_1 & 0 \end{bmatrix} \quad (3.24)$$

On propose alors le vecteur  $x = [x_1, x_2, x_3]^T$  qui permet de calculer le produit vectoriel de deux vecteurs en passant par le produit matriciel  $x \times y = [x]^\times y$ . L'opération inverse se note  $V$  et est telle que si  $[x]^\times = X$  alors  $X^V = x$ .

On définit la notation suivante pour le vecteur des angles d'Euler (3.25) ainsi que le vecteur des états de positions du drone (3.26) dans le repère inertiel :

$$\eta = \begin{bmatrix} \phi \\ \theta \\ \psi \end{bmatrix}_I \quad (3.25) \quad \xi = \begin{bmatrix} x \\ y \\ z \end{bmatrix}_I \quad (3.26)$$

On définit la notation suivante pour les vitesses de rotation (3.27) du drone et les vitesses linéaires (3.28) dans le repère du corps :

$$\Omega = \begin{bmatrix} p \\ q \\ r \end{bmatrix}_B \quad (3.27) \quad \mathbf{V}_B = \begin{bmatrix} u \\ v \\ w \end{bmatrix}_B \quad (3.28)$$

On définit les matrices de rotation du  $i^{\text{ème}}$  bras par rapport au corps du drone

$${}^B\mathcal{R}_{a_i} = \begin{bmatrix} \cos(\vartheta_i) & -\sin(\vartheta_i) & 0 \\ \sin(\vartheta_i) & \cos(\vartheta_i) & 0 \\ 0 & 0 & 1 \end{bmatrix} \quad (3.29)$$

La valeur de  $\vartheta_i$  dépend du bras concerné. En effet, le premier bras est incliné de  $45^\circ$  par rapport au nez de l'appareil, puis les bras suivant sont perpendiculaires au précédent. Les valeurs de  $\vartheta_i$  seront donc respectivement  $45, 135, -135, -45$  degrés pour les 4 bras en configuration  $\times$ .

On définit  $\omega_i$  la vitesse angulaire de la  $i^{\text{ème}}$  hélice. On définit  $k_f$  la constante de poussée proportionnelle au carré de la vitesse de rotation de l'hélice [ $\text{N.s}^2.\text{rad}^{-2}$ ] ainsi que l'expression de la poussée de la  $i^{\text{ème}}$  hélice :

$$T_i = k_f \omega_i^2 \quad (3.30)$$

On donne l'expression du vecteur des poussées correspondantes (3.31).

$$\mathbf{T} = \begin{bmatrix} T_1 \\ T_2 \\ T_3 \\ T_4 \end{bmatrix} \quad (3.31)$$

Pour la suite on suppose que les forces des hélices sont parfaitement orientées suivant l'axe  $\vec{z}_B$ . De plus on suppose que les constantes de poussées sont identiques pour tous les moteurs.

### 3.2.1.2 Dynamique du système

Afin d'obtenir les équations dynamiques du système, on exprime les forces du système dans le repère du corps du drone  $\mathcal{F}_B$ . La somme des forces dans le système est donnée par la relation suivante.

On développe l'équation en utilisant l'exemple du système quadricoptère classique :

$$\begin{aligned}
 \sum \vec{F}_B &= \sum {}^B \mathcal{R}_I F_{ext_I} + \sum_{i=0}^4 {}^B \mathcal{R}_{p_i} F_{p_i} + \mathbf{\Omega} \times m \mathbf{V}_B \\
 &= \sum {}^B \mathcal{R}_I F_{ext_I} + \sum_{i=0}^4 F_{p_i} + \mathbf{\Omega} \times m \mathbf{V}_B \\
 &= {}^B \mathcal{R}_I m \begin{bmatrix} 0 \\ 0 \\ g \end{bmatrix} + \sum_{i=1}^4 \begin{bmatrix} 0 \\ 0 \\ T_i \end{bmatrix} + \mathbf{\Omega} \times m \mathbf{V}_B
 \end{aligned} \tag{3.32}$$

Pour les moments, on définit les sens de rotations positifs comme sur la Figure 3.9. Cela veut dire que si le drone tourne dans le sens de la flèche, l'angle mesuré est positif et inversement. On précise ici que pendant les tests en simulation et en réel, les angles mesurés respectent également cette convention. La convention utilisée n'a pas forcément d'importance à partir du moment où elle est respectée du début à la fin.

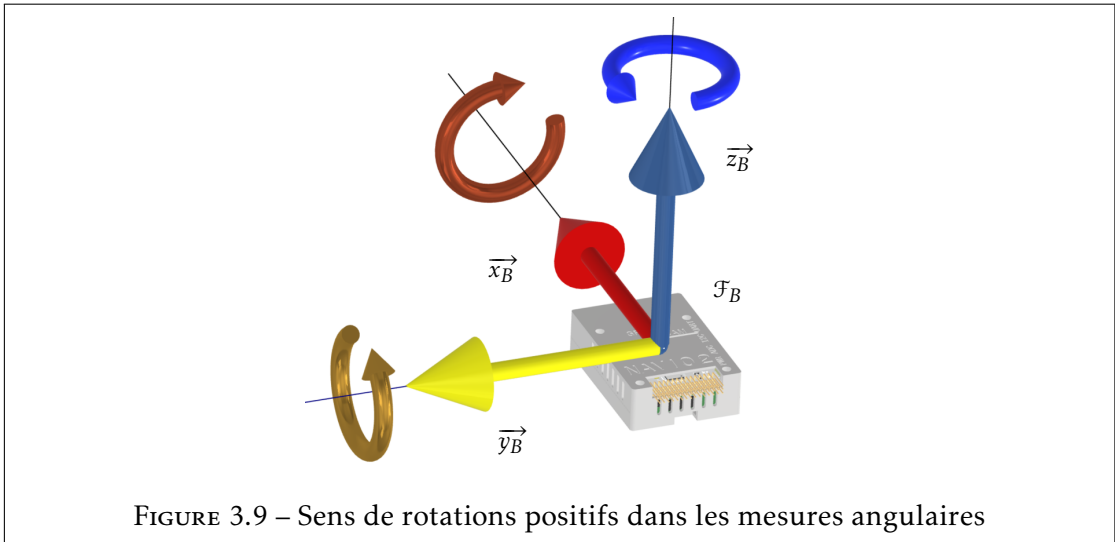


FIGURE 3.9 – Sens de rotations positifs dans les mesures angulaires

Le terme  $\sum \vec{M}_B$  est constitué de l'ensemble des couples qui s'exercent sur le corps du système. Pour plus de détails on considère d'abord le couple de rétroaction lié à la résistance de l'air face à la rotation des hélices noté  $\tau_{drag}$

$$\tau_{drag_i} \approx -k_\tau \omega_i^2 \tag{3.33}$$

avec  $k_\tau$  la constante de couple de trainée proportionnelle au carré de la vitesse de rotation de l'hélice [N.m.s<sup>2</sup>.rad<sup>-2</sup>]. On rappelle que sur un drone les hélices ont un sens de rotation qui s'alterne, une fois *Clock wise* (CW) et une fois *Counter Clock Wise* (CCW). Cela implique donc une trainée qui s'alternera également, négative puis positive. Dans un drone classique, le fait d'alterner le sens de rotation d'hélice permet de conserver une rotation nulle sur l'axe  $\vec{z}_B$ . Changer la vitesse de rotation de l'un des couples d'hélice (CW ou CCW) permet de faire tourner le drone sur son lacet.

En raison de la rotation du corps (et des bras par la suite) dans un système où les hélices tournent à très haute vitesse, on doit considérer le couple gyroscopique. Son expression est largement connue de la littérature, on ne citera que [39] ou encore [40] où l'effet est décrit.

On considère le couple gyroscopique des hélices dû à la rotation du corps noté  $\tau_{G\Omega}$  et définit comme suit :

$$\tau_{G\Omega} = \sum_{i=1}^4 p_i \mathcal{R}_B \Omega \times J_{prop} \begin{bmatrix} 0 \\ 0 \\ \omega_{(i)} \end{bmatrix}_{p_i} \quad (3.34)$$

On se rend compte que ce terme est négligeable globalement dans le drone. En effet, le couple gyroscopique crée par chaque bras va très largement se compenser deux à deux puisque le sens de rotation des hélices est opposé par paire. On ne réécrira donc pas ce terme dans cette section pour les calculs qui suivent.

On considère ensuite le couple dû à la propulsion lié à la rotation des hélices noté  $\tau_{T_i}$ .

$$\tau_{T_i} = \begin{bmatrix} L \\ 0 \\ h \end{bmatrix} \times \begin{bmatrix} 0 \\ 0 \\ T_i \end{bmatrix} = \begin{bmatrix} 0 \\ -T_i L \\ 0 \end{bmatrix} \quad (3.35)$$

Enfin, on considère les effets de Coriolis dûs à la rotation du corps et noté  $\tau_{CB}$ .

$$\tau_{CB} = \Omega \times J \Omega \quad (3.36)$$

Par la suite on va exprimer les forces du drone dans le repère inertiel et les couples dans le repère du corps. La raison en est que les mesures de positions et vitesses linéaires réalisées par les capteurs OptiTrack, GPS et baromètre sont réalisées dans le repère terrestre. Les mesures de vitesses et de position angulaires quant à elles sont faites à l'aide d'une centrale inertielle directement sur le drone, donc dans son repère.

On obtient les équations suivantes pour les forces exprimées dans le repère fixe :

$$m \ddot{\xi}_I = {}^I \mathcal{R}_B \begin{bmatrix} 0 \\ 0 \\ T_1 + T_2 + T_3 + T_4 \end{bmatrix} - \begin{bmatrix} 0 \\ 0 \\ mg \end{bmatrix} \quad (3.37)$$

On remarque que l'expression  $\Omega \times m \mathbf{V}_B$  n'est pas présente dans le repère inertiel. Pour illustrer ceci, prenons l'exemple d'un vaisseau dans l'espace, sans force gravitationnelle autour, qui se déplace à vitesse constante. Si ce vaisseau tourne sur lui-même, les membres de l'équipage auront tantôt l'impression de reculer, tantôt d'avancer. Ce phénomène est lié à ce terme mais n'intervient pas sur la dynamique du vaisseau dans le repère inertiel. La position et vitesse angulaire étant exprimées dans le repère du corps, les couples sont quant à eux donnés dans le repère du corps avec l'expression suivante :

$$\begin{aligned} \sum \overrightarrow{M}_B &= \sum_{i=1}^4 \tau_{drag_i} + \sum_{i=1}^4 {}^B \mathcal{R}_{a_i} \tau_{T_i} - \tau_{CB} \\ &= \sum_{i=1}^4 \tau_{drag_i} - \tau_{CB} \\ &+ \begin{bmatrix} \frac{\sqrt{2}}{2} & -\frac{\sqrt{2}}{2} & 0 \\ \frac{\sqrt{2}}{2} & \frac{\sqrt{2}}{2} & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 0 \\ -T_1 L \\ 0 \end{bmatrix} + \begin{bmatrix} -\frac{\sqrt{2}}{2} & -\frac{\sqrt{2}}{2} & 0 \\ \frac{\sqrt{2}}{2} & -\frac{\sqrt{2}}{2} & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 0 \\ -T_2 L \\ 0 \end{bmatrix} \\ &+ \begin{bmatrix} -\frac{\sqrt{2}}{2} & \frac{\sqrt{2}}{2} & 0 \\ -\frac{\sqrt{2}}{2} & -\frac{\sqrt{2}}{2} & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 0 \\ -T_3 L \\ 0 \end{bmatrix} + \begin{bmatrix} \frac{\sqrt{2}}{2} & \frac{\sqrt{2}}{2} & 0 \\ -\frac{\sqrt{2}}{2} & \frac{\sqrt{2}}{2} & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 0 \\ -T_4 L \\ 0 \end{bmatrix} \end{aligned} \quad (3.38)$$



Ce qui donne finalement :

$$\vec{M}_B = \delta \begin{bmatrix} 0 \\ 0 \\ -T_1 + T_2 - T_3 + T_4 \end{bmatrix} + \begin{bmatrix} \frac{\sqrt{2}}{2} L (T_1 + T_2 - T_3 - T_4) \\ \frac{\sqrt{2}}{2} L (-T_1 + T_2 + T_3 - T_4) \\ 0 \end{bmatrix} - \tau_{CB} \quad (3.39)$$

Les facteurs  $\frac{\sqrt{2}}{2}$  viennent du fait que dans cette configuration les bras sont à  $45^\circ$  du nez du drone. On note  $\delta = \frac{k_r}{k_f}$ , et on définit la constante suivante  $p = \frac{\sqrt{2}}{2}$ . L'expression précédente deviens alors :

$$\sum \vec{M}_B = \begin{bmatrix} pL & pL & -pL & -pL \\ -pL & pL & pL & -pL \\ -\delta & \delta & -\delta & \delta \end{bmatrix} \mathbf{T} - \tau_{CB} \quad (3.40)$$

On retrouve bien la dynamique classique d'un quadricoptère. Pour s'incliner autour de ses deux premiers axes de rotations, on augmentera (ou diminuera) la poussée des hélices. Pour faire tourner le drone sur lui-même il faudra cependant jouer sur une différence de couple liée à la trainée des hélices. Cette introduction permet de poser toutes les bases pour les sections suivantes où l'on considère que les bras portant le groupe moto-propulseur peuvent s'incliner vis-à-vis du corps du drone.

### 3.2.2 Repères et variables pour des bras inclinables

Passons à la modélisation du système avec bras inclinables. Une première configuration où l'utilisation d'un bras inclinable est quasiment une nécessité sont les tricopters. De par leur mécanique, il est intéressant de gérer le lacet par la rotation de l'un des bras. On trouve des exemples où ce procédé est appliqué dans [8], [41], [42].

Une configuration en quadricoptère plus classique mais également à bras inclinables présentée en [43] permet de passer d'un quadricoptère à un tricoptère en cas de faute de l'un des moteurs. L'étude se portera cependant sur un quadricoptère (généralisable à un octocoptère) en raison de l'imposante charge à embarquer. Celui-ci possèdera un degré de liberté supplémentaire sur chaque axe des bras contrairement à un drone conventionnel.

La littérature montre dans des travaux récents le potentiel de cette configuration. On se basera notamment sur les travaux de [24], [25], [44]-[49]. On peut également citer [23], [50], [51] pour une configuration en hexacoptère.

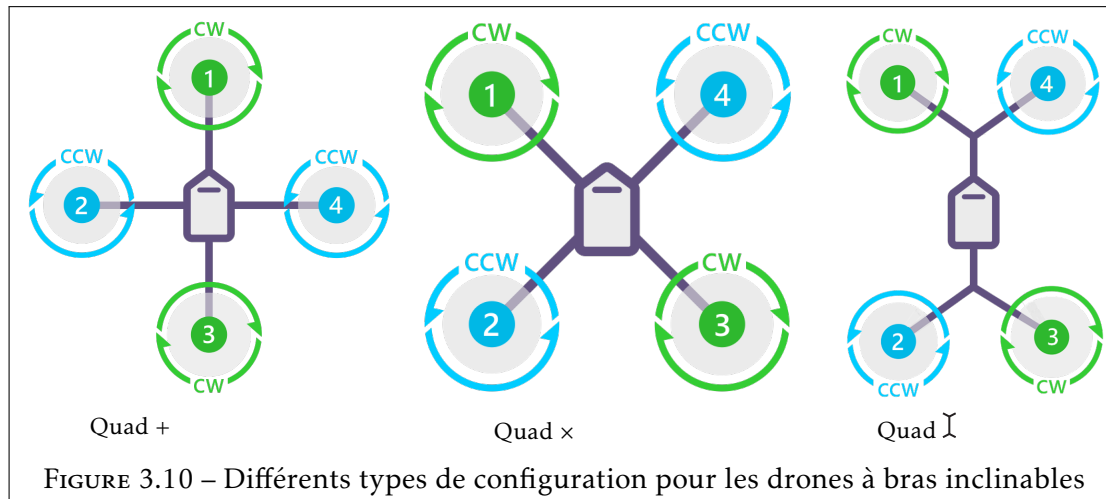
D'autres configurations intéressantes comme [52], [53] proposent non pas 1 mais 2 degrés de mobilités supplémentaires sur chaque moteur permettant notamment de limiter les problèmes de singularité, qui seront discutés dans le chapitre suivant.

Une configuration de drone convertible en avion est proposée dans [9], [10]. Cela permet d'utiliser tout les avantages du décollage vertical ainsi que ceux du vol de longue durée d'un avion. On remarque également que ce mode de vol est directement disponible dans les frameworks d'auto-pilotes comme ardupilot dont on reparlera plus tard.

[54] propose une structure massive avec deux moteurs inclinables, permettant une manœuvrabilité sur 5 degrés de liberté indépendants mais avec seulement deux moteurs et deux servomoteurs. [55] propose un drone capable de se mettre à la verticale pour passer un espace étroit par exemple, en inclinant les hélices à la verticale.

Dans [56], deux actionneurs inclineront toutes les hélices simultanément pour pouvoir accéder aux degrés de libertés indépendants manquants sur un drone classique. Ce drone pourra alors voler dans son plan horizontal sans devoir s'incliner. De manière analogue, [57], [58] proposent une configuration similaire en gardant le châssis d'un avion pour plus de maniabilité.

Dans cette étude, on détaille les calculs de la modélisation pour 3 configurations proposées en Figure 3.10.



Ces trois configurations correspondent à la chronologie de la réflexion faite dans le cadre de la thèse. On souhaite répondre le plus efficacement possible au besoin en n'arrivant pas dans des solutions entrant en conflit avec les contraintes fixées ou physiques du système (singularité).

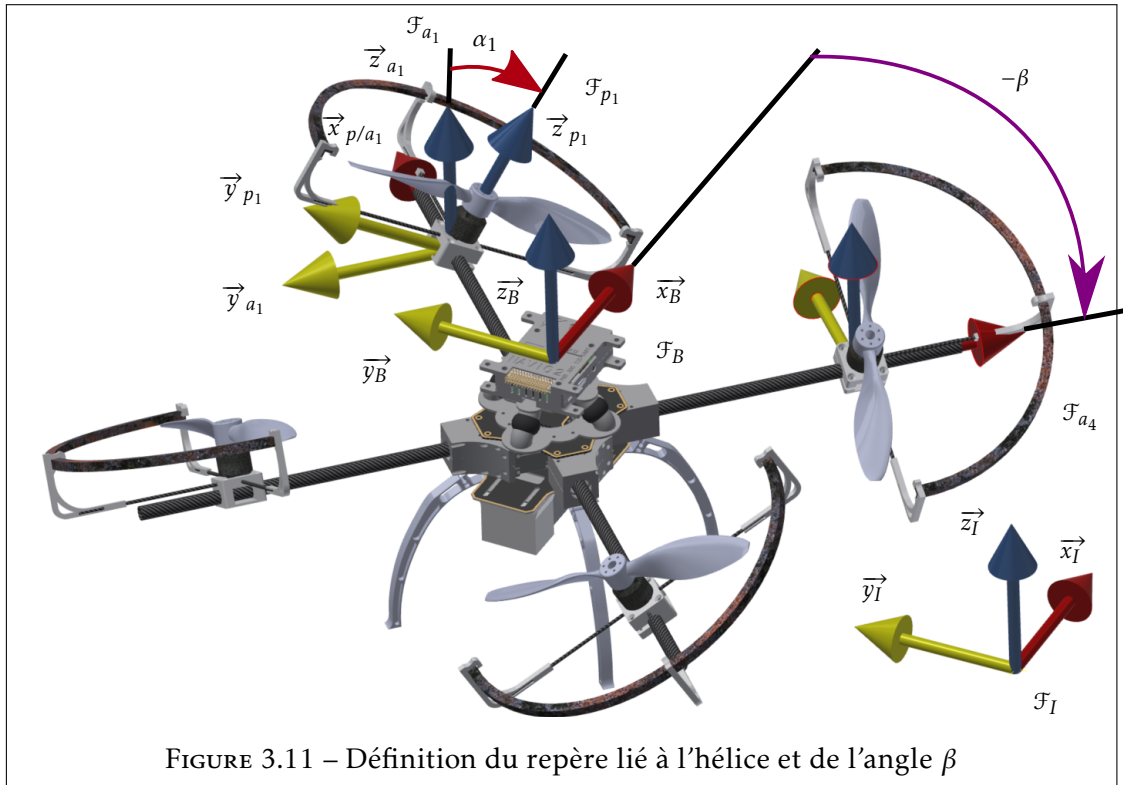
La première configuration en "+" permet une analyse sur un système simple car les axes des bras sont confondus avec les axes du corps. Celle-ci sera rapidement abandonnée car elle possède son point de singularité directement dans son point d'utilisation (voir Section 4.3).

La deuxième configuration correspond directement à la première version de RoMEO qui sera réalisée en prototype et testée pour valider la loi de commande et l'architecture mécanique. Cette configuration n'est pas totalement optimisée pour notre problème. En effet, en position verticale une grande quantité de l'énergie est contreproductive, on détaille ce point d'avantage en Section 3.16.

Enfin, la dernière configuration correspond à la version finale de RoMEO. On note également que cette variante inédite a été proposée suite à l'analyse des limites de la configuration précédente. A ce jour, cette configuration n'a pas encore été analysée dans la littérature, elle constitue donc une réelle innovation et un challenge à contrôler.

On pourra directement comparer la deuxième configuration avec un quadricoptère classique. En effet, si les bras sont inclinés à  $0^\circ$ , on se place dans le cas particulier du X4 conventionnel.

De nouvelles configurations impliquent également des nouvelles variables et repères que l'on va commencer par définir ci-dessous. En Figure 3.11 on rajoute le repère lié au  $i^{\text{ème}}$  groupe moto-pulseur noté  $\mathcal{F}_{p_i}$ .

FIGURE 3.11 – Définition du repère lié à l'hélice et de l'angle  $\beta$ 

On définit les matrices de rotation de la  $i^{\text{ème}}$  hélice par rapport au  $i^{\text{ème}}$  bras :

$${}^{a_i}\mathcal{R}_{p_i} = \begin{bmatrix} 1 & 0 & 0 \\ 0 & \cos(\alpha_i) & -\sin(\alpha_i) \\ 0 & \sin(\alpha_i) & \cos(\alpha_i) \end{bmatrix} \quad (3.41)$$

Avec  $\alpha_i$  l'angle permettant de passer du repère  $\mathcal{F}_{a_i}$  au repère  $\mathcal{F}_{p_i}$ , défini positif pour une rotation dans le sens représenté sur la Figure 3.11.

Remarque : la valeur de  $\vartheta_i$  dépend non seulement de la configuration mais également du bras considéré. Avec les trois configurations proposées, les valeurs de  $\vartheta$  sont présentées dans le tableau suivant pour les formes notées "+", "x", et "I"

| Configuration | Bras 1 (deg) | Bras 2 (deg) | Bras 3 (deg) | Bras 4 (deg) |
|---------------|--------------|--------------|--------------|--------------|
| +             | 0            | $\pi/2$      | $-\pi$       | $-\pi/2$     |
| x             | $\pi/4$      | $3\pi/4$     | $-3\pi/4$    | $-\pi/4$     |
| I             | $\beta$      | $\pi-\beta$  | $-\pi+\beta$ | $-\beta$     |

TABLEAU 3.3 – Définition des variables

Avec  $\beta$  une constante liée à la construction du drone. Sa valeur sera détaillée dans la Section 3.5. On note  $\pi - \beta$  et son opposé, les angles pour les deuxième et troisième bras. Le  $\pi$  vient du fait que l'angle  $\beta$  est fixe, alors que l'angle entre le premier et le deuxième bras n'est pas le même qu'entre le deuxième et le troisième. La symétrie du système (correspondant au  $\pi$ ) permet cependant d'exprimer cet angle en fonction de  $\beta$ .

Le fait d'opter pour une configuration à bras inclinables implique l'ajout d'effets mécaniques supplémentaires. Au niveau des couples, deux nouveaux couples s'ajoutent aux couples existants.  $\tau_\alpha$  qui représente le couple dû à la rotation des bras et  $\tau_{G\alpha}$  qui correspond au couple gyroscopique lié à la rotation des bras pendant que les hélices tournent.

Leurs expressions sont les suivantes :

$$\boldsymbol{\tau}_\alpha = J_a \ddot{\alpha} \quad (3.42)$$

$$\boldsymbol{\tau}_{G\alpha} = J_p \begin{bmatrix} 0 \\ 0 \\ \omega^{(i)} \end{bmatrix} \times \begin{bmatrix} \dot{\alpha}^{(i)} \\ 0 \\ 0 \end{bmatrix} = J_p \begin{bmatrix} 0 \\ \dot{\alpha}^{(i)} \omega^{(i)} \\ 0 \end{bmatrix} \quad (3.43)$$

[59] Propose d'ailleurs de se servir de cet effet gyroscopique pour contrôler un bi-rotor. Le principe utilisé est l'inclinaison des moteurs les uns vers les autres, pour créer un couple permettant une inclinaison du corps. Avec les éléments précédemment cités, on propose finalement le Tableau 3.4 qui récapitule tous les moments dans le système.

| Paramètres       | Description   |
|------------------|---|
| $\tau_{drag}$    | Moment de rétroaction des moteurs dans leur repère        |
| $\tau_T$         | Couple dû aux forces                                      |
| $\tau_\alpha$    | Moment dû à l'inertie de rotation du bras (dans le servo) |
| $\tau_{G\alpha}$ | Moment gyroscopique des hélices dû à la rotation des bras |
| $\tau_{G\Omega}$ | Moment gyroscopique des hélices dû à la rotation du drone |
| $\tau_{CB}$      | Moment de Coriolis dans le repère propre du drone         |

TABLEAU 3.4 – Définition des moments présents dans le drone

Par la suite on regroupera tous les termes gyroscopiques en un seul terme  $\mathbf{C}_{g-all}$  :

$$\mathbf{C}_{g-all} = \boldsymbol{\tau}_{CB} + \sum_{i=1}^4 {}^B \mathcal{R}_{p_i} (\boldsymbol{\tau}_{G\Omega} + \boldsymbol{\tau}_{G\alpha}) \quad (3.44)$$

Le moment lié à la rotation des bras  $\tau_\alpha$  sera négligé car l'inertie du bras sur son axe de rotation est très faible. On ajoutera l'inertie des bras à l'inertie du corps pour n'avoir qu'un seul terme noté  $J$ .

Les termes gyroscopiques dûs à la rotation des hélices ont une influence importante sur le système. Dans la suite de l'étude on cherchera à caractériser ce terme pour pouvoir ensuite le compenser dans la commande. Les autres termes de Coriolis quant à eux seront négligés dans un premiers temps. En effet, l'application du système n'incluant pas de rotations rapides sur plusieurs axes dans le repère du corps, on peut admettre qu' $\Omega$  est suffisamment petit. Enfin, le terme  $\tau_T$  qui correspond au couple dû aux forces est différent pour chaque configuration et sera donc détaillé.

### 3.3 Quadricoptère +

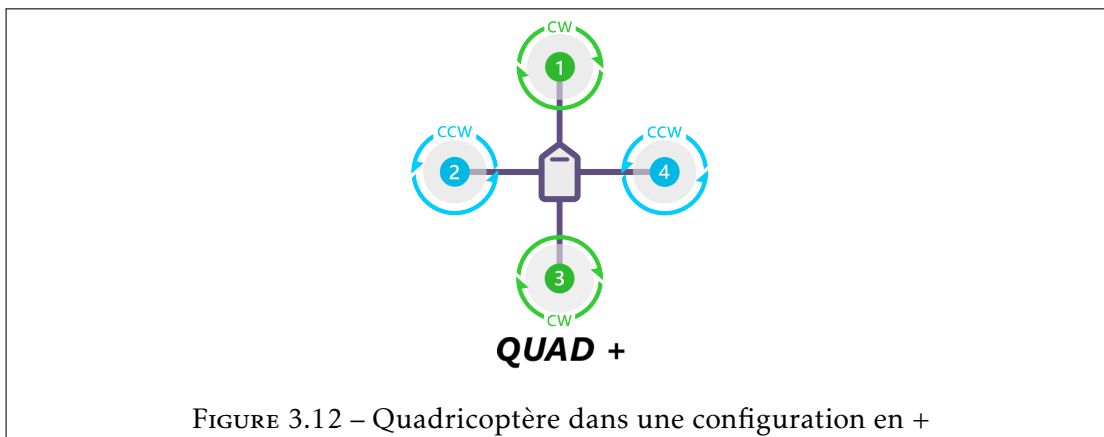


FIGURE 3.12 – Quadricoptère dans une configuration en +

On propose maintenant de passer à cette première configuration en + avec bras inclinables. L'étude se focalise d'abord sur l'expression des forces avant de passer à l'analyse des moments dans le système. Dans cette configuration le nez du drone pointe directement vers le premier bras comme le montre la Figure 3.12, ce qui facilite les calculs. Cette première approche permet de se familiariser avec les drones à bras inclinables.

### 3.3.1 Forces

L'expression des forces du système dans le repère inertiel est la suivante :

$$\sum \vec{F}_{I_+} = {}^I \mathcal{R}_B \sum_{i=0}^4 {}^B \mathcal{R}_{\alpha_i} {}^{\alpha_i} \mathcal{R}_{p_i} F_{prop(i)} + \sum F_{ext} \quad (3.45)$$

$$= {}^I \mathcal{R}_B \left( \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 1 & 0 & 0 \\ 0 & \cos(\alpha_1) & -\sin(\alpha_1) \\ 0 & \sin(\alpha_1) & \cos(\alpha_1) \end{bmatrix} \begin{bmatrix} 0 \\ 0 \\ T_1 \end{bmatrix} \right) \quad (3.46)$$

$$+ \begin{bmatrix} 0 & -1 & 0 \\ 1 & 0 & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 1 & 0 & 0 \\ 0 & \cos(\alpha_2) & -\sin(\alpha_2) \\ 0 & \sin(\alpha_2) & \cos(\alpha_2) \end{bmatrix} \begin{bmatrix} 0 \\ 0 \\ T_2 \end{bmatrix} \quad (3.47)$$

$$+ \begin{bmatrix} -1 & 0 & 0 \\ 0 & -1 & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 1 & 0 & 0 \\ 0 & \cos(\alpha_3) & -\sin(\alpha_3) \\ 0 & \sin(\alpha_3) & \cos(\alpha_3) \end{bmatrix} \begin{bmatrix} 0 \\ 0 \\ T_3 \end{bmatrix} \quad (3.48)$$

$$+ \begin{bmatrix} 0 & 1 & 0 \\ -1 & 0 & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 1 & 0 & 0 \\ 0 & \cos(\alpha_4) & -\sin(\alpha_4) \\ 0 & \sin(\alpha_4) & \cos(\alpha_4) \end{bmatrix} \begin{bmatrix} 0 \\ 0 \\ T_4 \end{bmatrix} + m \begin{bmatrix} 0 \\ 0 \\ g \end{bmatrix} \quad (3.49)$$

$$(3.50)$$

On rappelle l'expression de  $T_i$  dans l'équation (3.30). En développant on obtient :

$$\sum \vec{F}_{I_+} = {}^I \mathcal{R}_B \left( \mathcal{A}_{1+}(\alpha_1) \begin{bmatrix} 0 \\ 0 \\ T_1 \end{bmatrix} + \mathcal{A}_{2+}(\alpha_2) \begin{bmatrix} 0 \\ 0 \\ T_2 \end{bmatrix} + \mathcal{A}_{3+}(\alpha_3) \begin{bmatrix} 0 \\ 0 \\ T_3 \end{bmatrix} + \mathcal{A}_{4+}(\alpha_4) \begin{bmatrix} 0 \\ 0 \\ T_4 \end{bmatrix} \right) + m \begin{bmatrix} 0 \\ 0 \\ g \end{bmatrix} \quad (3.51)$$

Avec les matrices  $\mathcal{A}_{i+}(\alpha_i)$  définies de la manière suivante :

$$\begin{aligned} \mathcal{A}_{1+}(\alpha_1) &= \begin{bmatrix} 1 & 0 & 0 \\ 0 & \cos(\alpha_1) & -\sin(\alpha_1) \\ 0 & \sin(\alpha_1) & \cos(\alpha_1) \end{bmatrix} & \mathcal{A}_{2+}(\alpha_2) &= \begin{bmatrix} 0 & -\cos(\alpha_2) & \sin(\alpha_2) \\ 1 & 0 & 0 \\ 0 & \sin(\alpha_2) & \cos(\alpha_2) \end{bmatrix} \\ \mathcal{A}_{3+}(\alpha_3) &= \begin{bmatrix} -1 & 0 & 0 \\ 0 & -\cos(\alpha_3) & \sin(\alpha_3) \\ 0 & \sin(\alpha_3) & \cos(\alpha_3) \end{bmatrix} & \mathcal{A}_{4+}(\alpha_4) &= \begin{bmatrix} 0 & \cos(\alpha_4) & -\sin(\alpha_4) \\ -1 & 0 & 0 \\ 0 & \sin(\alpha_4) & \cos(\alpha_4) \end{bmatrix} \end{aligned} \quad (3.52)$$

En développant (3.51) on obtient :

$$\sum \vec{F}_{I_+} = {}^I \mathcal{R}_B \left( \begin{bmatrix} 0 \\ -\sin(\alpha_1) \\ \cos(\alpha_1) \end{bmatrix} T_1 + \begin{bmatrix} \sin(\alpha_2) \\ 0 \\ \cos(\alpha_2) \end{bmatrix} T_2 + \begin{bmatrix} 0 \\ \sin(\alpha_3) \\ \cos(\alpha_3) \end{bmatrix} T_3 + \begin{bmatrix} -\sin(\alpha_4) \\ 0 \\ \cos(\alpha_4) \end{bmatrix} T_4 \right) + m \begin{bmatrix} 0 \\ 0 \\ g \end{bmatrix} \quad (3.53)$$

On propose :

$$\kappa_+(\alpha) = \begin{bmatrix} 0 & s(\alpha_2) & 0 & -s(\alpha_4) \\ -s(\alpha_1) & 0 & s(\alpha_3) & 0 \\ c(\alpha_1) & c(\alpha_2) & c(\alpha_3) & c(\alpha_4) \end{bmatrix} \quad (3.54)$$

Ce qui donne, en factorisant par le vecteur  $\mathbf{T}$  :

$$\sum \overrightarrow{F_{L_+}} = {}^I \mathcal{R}_B \kappa_+(\alpha) \mathbf{T} + m \begin{bmatrix} 0 \\ 0 \\ g \end{bmatrix} \quad (3.55)$$

### 3.3.2 Moments

En configuration + on décompose les différents moments présents. Pour faire écho à la section précédente, on ne détaille dans un premier temps que les moments de réaction des moteurs ainsi que les couples induits par les forces créées par les hélices. Les couples gyroscopiques seront exprimés par la suite.

On décompose le calcul, d'une part par colonne, à gauche les termes liés à la trainée des hélices et à droite ceux des forces, d'autre part par ligne, avec la 1<sup>ère</sup> ligne pour la première hélice et ainsi de suite. On obtient alors :

$$\begin{aligned} \sum \overrightarrow{M_{B_+}} = & \mathcal{A}_{1+}(\alpha_1) \begin{bmatrix} 0 \\ 0 \\ -\delta T_1 \end{bmatrix} + \begin{bmatrix} L \\ 0 \\ h \end{bmatrix} \times \begin{bmatrix} 0 \\ 0 \\ T_1 \end{bmatrix} \\ & + \mathcal{A}_{2+}(\alpha_2) \begin{bmatrix} 0 \\ 0 \\ \delta T_2 \end{bmatrix} + \begin{bmatrix} L \\ 0 \\ h \end{bmatrix} \times \begin{bmatrix} 0 \\ 0 \\ T_2 \end{bmatrix} \\ & + \mathcal{A}_{3+}(\alpha_3) \begin{bmatrix} 0 \\ 0 \\ -\delta T_3 \end{bmatrix} + \begin{bmatrix} L \\ 0 \\ h \end{bmatrix} \times \begin{bmatrix} 0 \\ 0 \\ T_3 \end{bmatrix} \\ & + \mathcal{A}_{4+}(\alpha_4) \begin{bmatrix} 0 \\ 0 \\ \delta T_4 \end{bmatrix} + \begin{bmatrix} L \\ 0 \\ h \end{bmatrix} \times \begin{bmatrix} 0 \\ 0 \\ T_4 \end{bmatrix} + \mathbf{C}_{\mathbf{g-all}_+} \end{aligned} \quad (3.56)$$

Avec  $\delta = \frac{k_r}{k_f}$  et donc avec l'équation (3.33) on a  $\delta T_i = \tau_{drag}$ .

Quand on développe, l'équation devient :

$$\sum \overrightarrow{M_{B_+}} = \mathcal{A}_{1+}(\alpha_1) \begin{bmatrix} 0 \\ -L T_1 \\ -\delta T_1 \end{bmatrix} + \mathcal{A}_{2+}(\alpha_2) \begin{bmatrix} 0 \\ -L T_2 \\ \delta T_2 \end{bmatrix} + \mathcal{A}_{3+}(\alpha_3) \begin{bmatrix} 0 \\ -L T_3 \\ -\delta T_3 \end{bmatrix} + \mathcal{A}_{4+}(\alpha_4) \begin{bmatrix} 0 \\ -L T_4 \\ \delta T_4 \end{bmatrix} + \mathbf{C}_{\mathbf{g-all}_+} \quad (3.57)$$

Remarque 1 : dans notre cas on peut écrire le produit vectoriel de la manière suivante :

$$\begin{bmatrix} \mathcal{A}_{1+}(\alpha_1) \\ L \\ 0 \\ h \end{bmatrix} \times \begin{bmatrix} 0 \\ 0 \\ T_1 \end{bmatrix} = \mathcal{A}_{1+}(\alpha_1) \begin{bmatrix} L \\ 0 \\ h \end{bmatrix} \times \begin{bmatrix} 0 \\ 0 \\ T_1 \end{bmatrix} \quad (3.58)$$

Cette propriété est due aux matrices de rotations. En développant on se rend effectivement compte que nombre des termes s'annulent par les propriétés trigonométriques.

Remarque 2 : De même que pour l'analyse du PVTOL-MI, le terme en  $h$  de la Figure 3.13 se simplifie. Sur la même figure, on représente les forces et les distances pour les couples dus à cette distance.

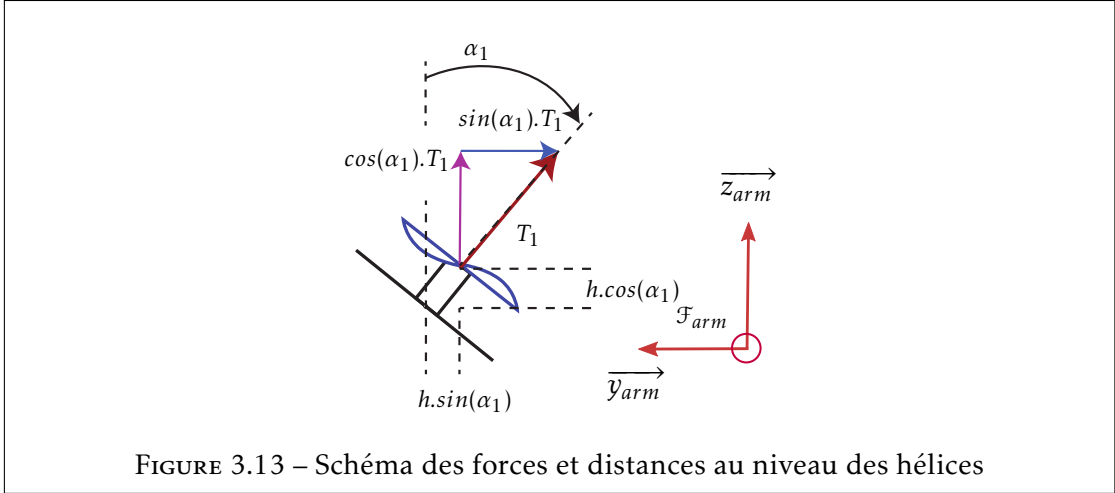


FIGURE 3.13 – Schéma des forces et distances au niveau des hélices

On retrouve cette simplification dans l'équation suivante en prenant l'exemple pour un seul bras :

$$\begin{bmatrix} 0 \\ -s(\alpha_1)h \\ c(\alpha_1)h \end{bmatrix} \times \begin{bmatrix} 0 \\ -s(\alpha_1)T_1 \\ c(\alpha_1)T_1 \end{bmatrix} = \begin{bmatrix} -s(\alpha_1)hc(\alpha_1)T_1 + c(\alpha_1)hs(\alpha_1)T_1 \\ 0 \\ 0 \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \\ 0 \end{bmatrix} \quad (3.59)$$

Par la suite on ne considèrera plus le couple lié à cette hauteur de moteur. On définit les matrices suivantes :

$$\chi_+(\alpha) = \begin{bmatrix} 0 & c(\alpha_2) & 0 & -c(\alpha_4) \\ -c(\alpha_1) & 0 & c(\alpha_3) & 0 \\ -s(\alpha_1) & -s(\alpha_2) & -s(\alpha_3) & -s(\alpha_4) \end{bmatrix} \quad (3.60)$$

$$\zeta_+(\alpha) = \begin{bmatrix} 0 & s(\alpha_2) & 0 & -s(\alpha_4) \\ s(\alpha_1) & 0 & -s(\alpha_3) & 0 \\ -c(\alpha_1) & c(\alpha_2) & -c(\alpha_3) & c(\alpha_4) \end{bmatrix} \quad (3.61)$$

On peut réécrire le système pour la partie des moments de la manière suivante :

$$\sum \overrightarrow{M_{B_+}} = L\chi_+(\alpha)\mathbf{T} + \delta\zeta_+(\alpha)\mathbf{T} + \mathbf{C}_{g\text{-all}_+} \quad (3.62)$$

Remarque : le terme  $L\chi_+(\alpha)$  correspond au couple induit par les forces des moteurs (noté précédemment  $\tau_T$ ) et  $\delta\zeta_+(\alpha)$  correspond au moment de réaction (noté précédemment  $\tau_{drag}$ ).

Les moments gyroscopiques induits par la rotation des bras sont exprimés de la manière suivante. On rappelle l'expression de ce terme dans le repère  $\mathcal{F}_{p_i}$  :

$$\tau_{G\alpha_i} = J_p \begin{bmatrix} 0 \\ \dot{\alpha}_{(i)} |\omega_{(i)}| (-1)^{i+1} \\ 0 \end{bmatrix} \quad (3.63)$$

On exprime ensuite ce terme dans le repère du corps et on fait la somme des 4 éléments

$$\begin{aligned} \tau_{G\alpha_+} &= \sum \tau_{G\alpha_{i+}} \\ &= J_p \left( \mathcal{A}_{1+}(\alpha_1) \begin{bmatrix} 0 \\ \dot{\alpha}_1 |\omega_1| \\ 0 \end{bmatrix} + \mathcal{A}_{2+}(\alpha_2) \begin{bmatrix} 0 \\ -\dot{\alpha}_2 |\omega_2| \\ 0 \end{bmatrix} + \mathcal{A}_{3+}(\alpha_3) \begin{bmatrix} 0 \\ \dot{\alpha}_3 |\omega_3| \\ 0 \end{bmatrix} + \mathcal{A}_{4+}(\alpha_4) \begin{bmatrix} 0 \\ -\dot{\alpha}_4 |\omega_4| \\ 0 \end{bmatrix} \right) \\ &= J_p \begin{bmatrix} 0 & e_2 c(\alpha_2) & 0 & -e_4 c(\alpha_4) \\ e_1 c(\alpha_1) & 0 & -e_3 c(\alpha_3) & 0 \\ e_1 s(\alpha_1) & -e_2 s(\alpha_2) & e_3 s(\alpha_3) & -e_4 s(\alpha_4) \end{bmatrix} \end{aligned} \quad (3.64)$$

Avec  $e_i = \dot{\alpha}_i |\omega_i|$ .

Comme on l'a dit précédemment on néglige les autres effets de Coriolis, car les vitesses angulaires du corps sont faibles et l'inertie des bras est très inférieure à celle du corps.

### 3.3.3 Synthèse

Finalement l'expression générale du système est de la forme suivante :

$$\begin{bmatrix} m \ddot{x} \\ m \ddot{y} \\ m \ddot{z} \\ J \dot{p} \\ J \dot{q} \\ J \dot{r} \end{bmatrix} = \begin{bmatrix} J \mathcal{R}_B & 0 \\ 0 & I^{[3 \times 3]} \end{bmatrix} \begin{bmatrix} T_2 s(\alpha_2) - T_4 s(\alpha_4) \\ -T_1 s(\alpha_1) + T_3 s(\alpha_3) \\ T_1 c(\alpha_1) + T_2 c(\alpha_2) + T_3 c(\alpha_3) + T_4 c(\alpha_4) \\ T_2 \delta s(\alpha_2) - T_4 \delta s(\alpha_4) + T_2 L c(\alpha_2) - T_4 L c(\alpha_4) \\ T_1 \delta s(\alpha_1) - T_3 \delta s(\alpha_3) - T_1 L c(\alpha_1) + T_3 L c(\alpha_3) \\ -T_1 L s(\alpha_1) - T_2 L s(\alpha_2) - T_3 L s(\alpha_3) - T_4 L s(\alpha_4) - T_1 \delta c(\alpha_1) + T_2 \delta c(\alpha_2) - T_3 \delta c(\alpha_3) + T_4 \delta c(\alpha_4) \end{bmatrix} - \begin{bmatrix} 0 \\ 0 \\ mg \\ 0 \\ 0 \\ 0 \end{bmatrix} + \mathbf{C}_{g\text{-all}+} \quad (3.65)$$

On propose de factoriser le vecteur avec les termes en  $\alpha$  en faisant un changement de variable comme suit :

$$\begin{bmatrix} u_1 \\ u_2 \\ u_3 \\ u_4 \\ u_5 \\ u_6 \end{bmatrix} \triangleq \mathcal{O}_+ \begin{bmatrix} T_1 \sin(\alpha_1) \\ T_1 \cos(\alpha_1) \\ T_2 \sin(\alpha_2) \\ T_2 \cos(\alpha_2) \\ T_3 \sin(\alpha_3) \\ T_3 \cos(\alpha_3) \\ T_4 \sin(\alpha_4) \\ T_4 \cos(\alpha_4) \end{bmatrix} \quad (3.66)$$



Avec :

$$\mathcal{O}_+ = \begin{bmatrix} 0 & 0 & 1 & 0 & 0 & 0 & -1 & 0 \\ -1 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 1 & 0 & 1 & 0 & 1 \\ 0 & 0 & \delta & L & 0 & 0 & -\delta & -L \\ \delta & -L & 0 & 0 & -\delta & L & 0 & 0 \\ -L & -\delta & -L & \delta & -L & -\delta & -L & \delta \end{bmatrix} \quad (3.67)$$

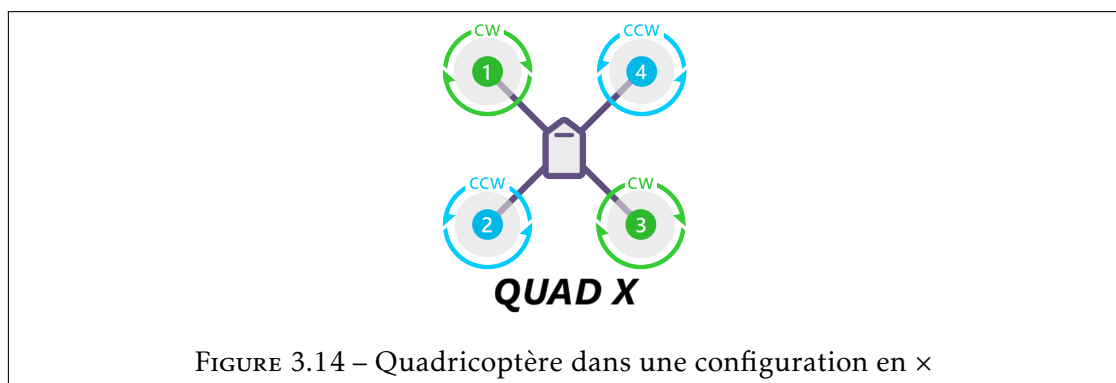
Ce qui donne quand on remplace dans l'équation précédente :

$$\begin{bmatrix} m\ddot{x} \\ m\ddot{y} \\ m\ddot{z} \\ J\dot{p} \\ J\dot{q} \\ J\dot{r} \end{bmatrix} = {}^I \mathcal{R}_B \begin{bmatrix} u_1 \\ u_2 \\ u_3 \\ u_4 \\ u_5 \\ u_6 \end{bmatrix} - \begin{bmatrix} 0 \\ 0 \\ mg \\ 0 \\ 0 \\ 0 \end{bmatrix} + \mathbf{C}_{g-\text{all}_+} \quad (3.68)$$

Remarque : si l'angle d'inclinaison des bras est nul on retrouve bien la configuration classique d'un quadricoptère, sans actionnement possible sur les axes  $\vec{x}_B$  et  $\vec{y}_B$ .

Cette section permet donc de poser tous les principes pour étudier les drones à bras inclinables. Quatre effets se distinguent, premièrement les forces liées aux hélices et liées à l'angle des bras. Deuxièmement les couples liés aux forces mais aussi, troisièmement, les couples de rétroactions qui ne sont plus seulement dans le plan horizontal du drone. Enfin, l'analyse des couples gyroscopiques permettra une analyse et une compensation de l'effet dans les chapitres suivants. La section suivante présente un modèle similaire avec la particularité de décaler le premier bras de  $45^\circ$  par rapport au nez de l'appareil.

### 3.4 Quadricoptère X



Cette section est dédiée à la configuration en forme de ×, voir Figure 3.14. C'est un cas particulier d'une configuration en + avec le nez du drone entre les bras 1 et 4. La plupart des drones sont construits sous cette forme. C'est pourquoi, cette configuration présente l'avantage de garder une certaine simplicité du système, non seulement sur l'analyse mécanique et la commande, mais également pour la réalisation. De plus, la littérature a pu très souvent explorer cette configuration, y compris pour des configurations avec bras inclinables.

Un grand avantage de cette configuration par rapport à la version précédente est le fait de garder plus de poussée sur les hélices. En position verticale par exemple, on pourra pousser à  $0.707 \times 4$  soit 2.828 fois la poussée maximale d'un groupe moto-propulseur. Cette poussée est à 2 pour la configuration +. D'autre part, comme évoqué, le problème des singularités apparaît à d'autres endroits que pour la section précédente. Un des points de fonctionnement principal du véhicule étant en position verticale, le décalage de la singularité sera un avantage supplémentaire, voir section 4.3. On détaille maintenant le modèle analytique qui correspond à la première version du prototype.

### 3.4.1 Forces

Les forces dans cette configurations sont de la forme suivante :

$$\begin{aligned}
\sum \vec{F}_{I_x} = {}^I \mathcal{R}_B & \left( k_f \begin{bmatrix} \frac{\sqrt{2}}{2} & -\frac{\sqrt{2}}{2} & 0 \\ \frac{\sqrt{2}}{2} & \frac{\sqrt{2}}{2} & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 1 & 0 & 0 \\ 0 & \cos(\alpha_1) & -\sin(\alpha_1) \\ 0 & \sin(\alpha_1) & \cos(\alpha_1) \end{bmatrix} \begin{bmatrix} 0 \\ 0 \\ \omega_1^2 \end{bmatrix} \right. \\
& + k_f \begin{bmatrix} -\frac{\sqrt{2}}{2} & -\frac{\sqrt{2}}{2} & 0 \\ \frac{\sqrt{2}}{2} & -\frac{\sqrt{2}}{2} & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 1 & 0 & 0 \\ 0 & \cos(\alpha_2) & -\sin(\alpha_2) \\ 0 & \sin(\alpha_2) & \cos(\alpha_2) \end{bmatrix} \begin{bmatrix} 0 \\ 0 \\ \omega_2^2 \end{bmatrix} \\
& + k_f \begin{bmatrix} -\frac{\sqrt{2}}{2} & \frac{\sqrt{2}}{2} & 0 \\ -\frac{\sqrt{2}}{2} & -\frac{\sqrt{2}}{2} & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 1 & 0 & 0 \\ 0 & \cos(\alpha_3) & -\sin(\alpha_3) \\ 0 & \sin(\alpha_3) & \cos(\alpha_3) \end{bmatrix} \begin{bmatrix} 0 \\ 0 \\ \omega_3^2 \end{bmatrix} \\
& \left. + k_f \begin{bmatrix} \frac{\sqrt{2}}{2} & \frac{\sqrt{2}}{2} & 0 \\ -\frac{\sqrt{2}}{2} & \frac{\sqrt{2}}{2} & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 1 & 0 & 0 \\ 0 & \cos(\alpha_4) & -\sin(\alpha_4) \\ 0 & \sin(\alpha_4) & \cos(\alpha_4) \end{bmatrix} \begin{bmatrix} 0 \\ 0 \\ \omega_4^2 \end{bmatrix} \right) + m \begin{bmatrix} 0 \\ 0 \\ g \end{bmatrix}
\end{aligned} \tag{3.69}$$

Ce qui donne :

$$\sum \vec{F}_{I_x} = {}^I \mathcal{R}_B \left( \mathcal{A}_{1 \times}(\alpha_1) \begin{bmatrix} 0 \\ 0 \\ T_1 \end{bmatrix} + \mathcal{A}_{2 \times}(\alpha_2) \begin{bmatrix} 0 \\ 0 \\ T_2 \end{bmatrix} + \mathcal{A}_{3 \times}(\alpha_3) \begin{bmatrix} 0 \\ 0 \\ T_3 \end{bmatrix} + \mathcal{A}_{4 \times}(\alpha_4) \begin{bmatrix} 0 \\ 0 \\ T_4 \end{bmatrix} \right) + m \begin{bmatrix} 0 \\ 0 \\ g \end{bmatrix} \tag{3.70}$$

Si on reprend la variable  $p = \frac{\sqrt{2}}{2}$ , on définit les matrices  $\mathcal{A}_{1 \times}(\alpha_1)$  de la manière suivante :

$$\begin{aligned}
\mathcal{A}_{1 \times}(\alpha_1) &= \begin{bmatrix} p & -p \cos(\alpha_1) & p \sin(\alpha_1) \\ p & p \cos(\alpha_1) & -p \sin(\alpha_1) \\ 0 & \sin(\alpha_1) & \cos(\alpha_1) \end{bmatrix} & \mathcal{A}_{2 \times}(\alpha_2) &= \begin{bmatrix} -p & -p \cos(\alpha_2) & p \sin(\alpha_2) \\ p & -p \cos(\alpha_2) & p \sin(\alpha_2) \\ 0 & \sin(\alpha_2) & \cos(\alpha_2) \end{bmatrix} \\
\mathcal{A}_{3 \times}(\alpha_3) &= \begin{bmatrix} -p & p \cos(\alpha_3) & -p \sin(\alpha_3) \\ -p & -p \cos(\alpha_3) & p \sin(\alpha_3) \\ 0 & \sin(\alpha_3) & \cos(\alpha_3) \end{bmatrix} & \mathcal{A}_{4 \times}(\alpha_4) &= \begin{bmatrix} p & p \cos(\alpha_4) & -p \sin(\alpha_4) \\ -p & p \cos(\alpha_4) & -p \sin(\alpha_4) \\ 0 & \sin(\alpha_4) & \cos(\alpha_4) \end{bmatrix}
\end{aligned} \tag{3.71}$$

On propose la notation suivante  $\odot$  pour l'opération mathématique qui, pour deux matrices produit une nouvelle matrice de même dimension que la deuxième, dont chaque élément  $i, j$  est le produit des éléments  $i, j$  des deux matrices originelles (c'est un produit de matrice qui ressemble au produit d'Hadamard).

On a par exemple :

$$\begin{bmatrix} a & b & c \end{bmatrix} \odot \begin{bmatrix} i & l \\ j & m \\ k & n \end{bmatrix} = \begin{bmatrix} ai & al \\ bj & bm \\ ck & cn \end{bmatrix} \quad (3.72)$$

On obtient pour le système de manière analogue :

$$\begin{aligned} \sum \overrightarrow{F_{I_x}} &= {}^I \mathcal{R}_B \left( \begin{bmatrix} p & p & 1 \end{bmatrix} \odot \begin{bmatrix} s(\alpha_1) & s(\alpha_2) & -s(\alpha_3) & -s(\alpha_4) \\ -s(\alpha_1) & s(\alpha_2) & s(\alpha_3) & -s(\alpha_4) \\ c(\alpha_1) & c(\alpha_2) & c(\alpha_3) & c(\alpha_4) \end{bmatrix} \mathbf{T} \right) + m \begin{bmatrix} 0 \\ 0 \\ g \end{bmatrix} \\ &= {}^I \mathcal{R}_B \kappa_x(\alpha) \mathbf{T} + m \begin{bmatrix} 0 \\ 0 \\ g \end{bmatrix} \end{aligned} \quad (3.73)$$

avec la matrice  $\kappa_x(\alpha)$  définit de la manière suivante :

$$\kappa_x(\alpha) = \begin{bmatrix} p & p & 1 \end{bmatrix} \odot \begin{bmatrix} s(\alpha_1) & s(\alpha_2) & -s(\alpha_3) & -s(\alpha_4) \\ -s(\alpha_1) & s(\alpha_2) & s(\alpha_3) & -s(\alpha_4) \\ c(\alpha_1) & c(\alpha_2) & c(\alpha_3) & c(\alpha_4) \end{bmatrix} \quad (3.74)$$

### 3.4.2 Moments

On utilise la même procédure qu'en Section 3.3.2. En configuration  $\times$  on obtient les équations suivantes pour les moments :

$$\sum \overrightarrow{M_{B_x}} = \mathcal{A}_{1\times}(\alpha_1) \begin{bmatrix} 0 \\ 0 \\ -\delta T_1 \end{bmatrix} + \begin{bmatrix} \mathcal{A}_{1\times}(\alpha_1) \\ 0 \\ h \end{bmatrix} \times \begin{bmatrix} \mathcal{A}_{1\times}(\alpha_1) \\ 0 \\ T_1 \end{bmatrix} \quad (3.75)$$

$$+ \mathcal{A}_{2\times}(\alpha_2) \begin{bmatrix} 0 \\ 0 \\ \delta T_2 \end{bmatrix} + \begin{bmatrix} \mathcal{A}_{2\times}(\alpha_2) \\ 0 \\ h \end{bmatrix} \times \begin{bmatrix} \mathcal{A}_{2\times}(\alpha_2) \\ 0 \\ T_2 \end{bmatrix} \quad (3.76)$$

$$+ \mathcal{A}_{3\times}(\alpha_3) \begin{bmatrix} 0 \\ 0 \\ -\delta T_3 \end{bmatrix} + \begin{bmatrix} \mathcal{A}_{3\times}(\alpha_3) \\ 0 \\ h \end{bmatrix} \times \begin{bmatrix} \mathcal{A}_{3\times}(\alpha_3) \\ 0 \\ T_3 \end{bmatrix} \quad (3.77)$$

$$+ \mathcal{A}_{4\times}(\alpha_4) \begin{bmatrix} 0 \\ 0 \\ \delta T_4 \end{bmatrix} + \begin{bmatrix} \mathcal{A}_{4\times}(\alpha_4) \\ 0 \\ h \end{bmatrix} \times \begin{bmatrix} \mathcal{A}_{4\times}(\alpha_4) \\ 0 \\ T_4 \end{bmatrix} + \mathbf{C}_{g-\text{all}_x} \quad (3.78)$$

Ce qui donne :

$$\sum \overrightarrow{M_{B_x}} = \mathcal{A}_{1\times}(\alpha_1) \begin{bmatrix} 0 \\ -L T_1 \\ -\delta T_1 \end{bmatrix} + \mathcal{A}_{2\times}(\alpha_2) \begin{bmatrix} 0 \\ -L T_2 \\ \delta T_2 \end{bmatrix} + \mathcal{A}_{3\times}(\alpha_3) \begin{bmatrix} 0 \\ -L T_3 \\ -\delta T_3 \end{bmatrix} + \mathcal{A}_{4\times}(\alpha_4) \begin{bmatrix} 0 \\ -L T_4 \\ \delta T_4 \end{bmatrix} + \mathbf{C}_{g-\text{all}_x} \quad (3.79)$$

On définit les matrices suivantes :

$$\chi_x(\alpha) = \begin{bmatrix} p & p & 1 \end{bmatrix} \odot \begin{bmatrix} c(\alpha_1) & c(\alpha_2) & -c(\alpha_3) & -c(\alpha_4) \\ -c(\alpha_1) & c(\alpha_2) & c(\alpha_3) & -c(\alpha_4) \\ -s(\alpha_1) & -s(\alpha_2) & -s(\alpha_3) & -s(\alpha_4) \end{bmatrix} \quad (3.80)$$

$$\zeta_x(\alpha) = \begin{bmatrix} p & p & 1 \end{bmatrix} \odot \begin{bmatrix} -s(\alpha_1) & s(\alpha_2) & s(\alpha_3) & -s(\alpha_4) \\ s(\alpha_1) & s(\alpha_2) & -s(\alpha_3) & -s(\alpha_4) \\ -c(\alpha_1) & c(\alpha_2) & -c(\alpha_3) & c(\alpha_4) \end{bmatrix} \quad (3.81)$$

On obtient alors :

$$\sum \overrightarrow{M_{B_x}} = L \chi_x(\alpha) \mathbf{T} + \delta \zeta_x(\alpha) \mathbf{T} + \mathbf{C}_{g-\text{all}_x} \quad (3.82)$$

On définit :

$$\begin{aligned} \tau_x(\alpha) &= (\delta \zeta_x(\alpha) + L \chi_x(\alpha)) \mathbf{T} \\ &= \begin{bmatrix} p & p & 1 \end{bmatrix} \odot \\ &\quad \begin{bmatrix} Lc(\alpha_1) - \delta s(\alpha_1) & Lc(\alpha_2) + \delta s(\alpha_2) & -Lc(\alpha_3) + \delta s(\alpha_3) & -Lc(\alpha_4) - \delta s(\alpha_4) \\ -Lc(\alpha_1) + \delta s(\alpha_1) & Lc(\alpha_2) + \delta s(\alpha_2) & Lc(\alpha_3) - \delta s(\alpha_3) & -Lc(\alpha_4) - \delta s(\alpha_4) \\ -Ls(\alpha_1) - \delta c(\alpha_1) & -Ls(\alpha_2) + \delta c(\alpha_2) & -Ls(\alpha_3) - \delta c(\alpha_3) & -Ls(\alpha_4) + \delta c(\alpha_4) \end{bmatrix} \mathbf{T} \end{aligned} \quad (3.83)$$

Ce qui donne finalement :

$$\sum \overrightarrow{M_{B_x}} = \tau_x(\alpha) + \mathbf{C}_{g-\text{all}_x} \quad (3.84)$$

Pour les couples gyroscopiques on donne :

$$\begin{aligned} \tau_{G\alpha_x} &= \sum \tau_{G\alpha_{i_x}} \\ &= J_p \left( \mathcal{A}_{1x}(\alpha_1) \begin{bmatrix} 0 \\ \dot{\alpha}_1 |\omega_1| \\ 0 \end{bmatrix} + \mathcal{A}_{2x}(\alpha_2) \begin{bmatrix} 0 \\ -\dot{\alpha}_2 |\omega_2| \\ 0 \end{bmatrix} + \mathcal{A}_{3x}(\alpha_3) \begin{bmatrix} 0 \\ \dot{\alpha}_3 |\omega_3| \\ 0 \end{bmatrix} + \mathcal{A}_{4x}(\alpha_4) \begin{bmatrix} 0 \\ -\dot{\alpha}_4 |\omega_4| \\ 0 \end{bmatrix} \right) \\ &= J_p \begin{bmatrix} p & p & 1 \end{bmatrix} \odot \begin{bmatrix} -e_1 c(\alpha_1) & e_2 c(\alpha_2) & e_3 c(\alpha_3) & -e_4 c(\alpha_4) \\ e_1 c(\alpha_1) & e_2 c(\alpha_2) & -e_3 c(\alpha_3) & -e_4 c(\alpha_4) \\ e_1 s(\alpha_1) & -e_2 s(\alpha_2) & e_3 s(\alpha_3) & -e_4 s(\alpha_4) \end{bmatrix} \end{aligned} \quad (3.85)$$

### 3.4.3 Synthèse

Si l'on applique le même changement de variable qu'en (3.66) de la Section 3.3.3 on obtient la matrice suivante :

$$\mathcal{O}_x = \begin{bmatrix} p & 0 & p & 0 & -p & 0 & -p & 0 \\ -p & 0 & p & 0 & p & 0 & -p & 0 \\ 0 & 1 & 0 & 1 & 0 & 1 & 0 & 1 \\ -p\delta & pl & p\delta & pl & p\delta & -pl & -p\delta & -pl \\ p\delta & -pl & p\delta & pl & -p\delta & pl & -p\delta & -pl \\ -l & -\delta & -l & \delta & -l & -\delta & -l & \delta \end{bmatrix} \quad (3.86)$$

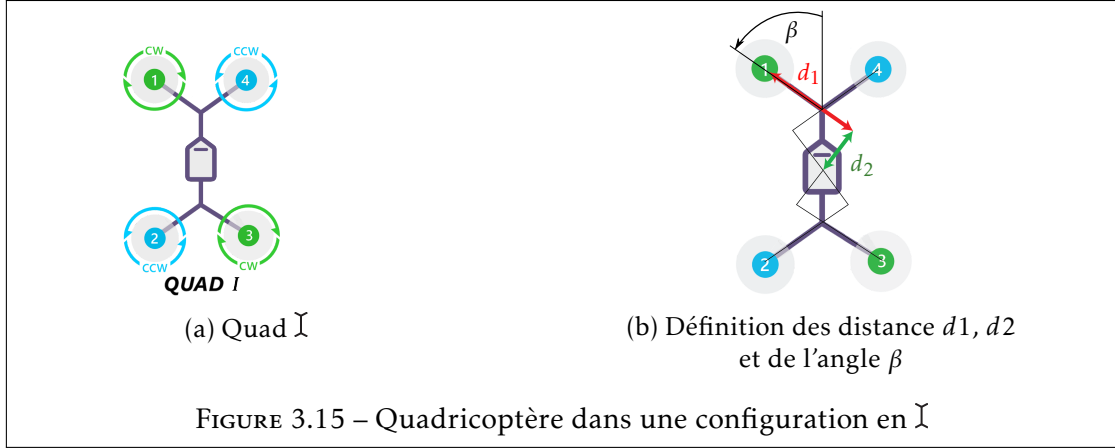
On fera le même changement de variables que précédemment. L'équation générale du système sera donc la même qu'en (3.68).

Remarque : on peut directement comparer ce modèle à la configuration en  $\times$  n'ayant pas de bras inclinable. Pour cela il suffit de considérer que les angles  $\alpha_i$  sont tous nuls. Concrètement cela implique que  $\sin(\alpha_i) = 0$  et donc qu'il ne faut considérer que les colonnes paires pour la matrice ci-dessus. Si l'on considère la matrice en (3.40), on retrouve bien la même expression que si l'on ne considère que les colonnes paires de l'équation (3.86).

A la différence du drone classique on voit directement que l'on possède des degrés d'actionnement sur les axes  $\vec{x}_B$  et  $\vec{y}_B$ .

Après cette configuration on propose de s'intéresser à une solution où les bras ne sont pas à  $45^\circ$  afin d'optimiser la portance que la propulsion offre.

### 3.5 Quadricoptère $\checkmark$



Dans cette section on présente le cas particulier d'une configuration en  $\checkmark$  avec le nez du drone entre les bras 1 et 4, voir Figure 3.15a (gauche). L'angle  $\beta$  devra être choisi arbitrairement en fonction des contraintes auquel le drone va être soumis. On reviendra sur le calcul de  $\beta$  à la fin de l'analyse du modèle. Pour ce dispositif, on définit les distances  $d_1$  et  $d_2$  exprimées dans le premier bras, sur la Figure 3.15b (droite), qui seront utilisées par la suite. Il est important de remarquer que la distance  $d_2$  est négative dans le repère des bras. En effet, si on rapporte le repère des bras dans le repère du corps on voit clairement que le sens de la distance est opposé à celui de l'axe  $\vec{y}_B$  des bras. Ce point est à garder en tête pour l'analyse, notamment sur les moments du système.

Comme précédemment on s'attarde sur l'expression des forces du système en fonction des actionneurs. La deuxième partie explicitera les moments pour finalement pouvoir synthétiser les résultats dans une seule matrice d'allocation globale en synthèse.

#### 3.5.1 Forces

L'expression des forces pour cette dernière configuration est la suivante :

$$\sum \vec{F}_{I\checkmark} = {}^I\mathcal{R}_B \left( k_f \begin{bmatrix} \cos(\beta) & -\sin(\beta) & 0 \\ \sin(\beta) & \cos(\beta) & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 1 & 0 & 0 \\ 0 & \cos(\alpha_1) & -\sin(\alpha_1) \\ 0 & \sin(\alpha_1) & \cos(\alpha_1) \end{bmatrix} \begin{bmatrix} 0 \\ 0 \\ \omega_1^2 \end{bmatrix} \right) \quad (3.87)$$

$$+ k_f \begin{bmatrix} \cos(\pi - \beta) & -\sin(\pi - \beta) & 0 \\ \sin(\pi - \beta) & \cos(\pi - \beta) & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 1 & 0 & 0 \\ 0 & \cos(\alpha_2) & -\sin(\alpha_2) \\ 0 & \sin(\alpha_2) & \cos(\alpha_2) \end{bmatrix} \begin{bmatrix} 0 \\ 0 \\ \omega_2^2 \end{bmatrix} \quad (3.88)$$

$$+ k_f \begin{bmatrix} \cos(-\pi + \beta) & -\sin(-\pi + \beta) & 0 \\ \sin(-\pi + \beta) & \cos(-\pi + \beta) & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 1 & 0 & 0 \\ 0 & \cos(\alpha_3) & -\sin(\alpha_3) \\ 0 & \sin(\alpha_3) & \cos(\alpha_3) \end{bmatrix} \begin{bmatrix} 0 \\ 0 \\ \omega_3^2 \end{bmatrix} \quad (3.89)$$

$$+ k_f \begin{bmatrix} \cos(-\beta) & -\sin(-\beta) & 0 \\ \sin(-\beta) & \cos(-\beta) & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 1 & 0 & 0 \\ 0 & \cos(\alpha_4) & -\sin(\alpha_4) \\ 0 & \sin(\alpha_4) & \cos(\alpha_4) \end{bmatrix} \begin{bmatrix} 0 \\ 0 \\ \omega_4^2 \end{bmatrix} \Bigg) + m \begin{bmatrix} 0 \\ 0 \\ g \end{bmatrix} \quad (3.90)$$

Ce qui donne :

$$\sum \overrightarrow{F_{I\mathcal{I}}} = {}^I \mathcal{R}_B \left( \mathcal{A}_{1\mathcal{I}}(\alpha_1) \begin{bmatrix} 0 \\ 0 \\ T_1 \end{bmatrix} + \mathcal{A}_{2\mathcal{I}}(\alpha_2) \begin{bmatrix} 0 \\ 0 \\ T_2 \end{bmatrix} + \mathcal{A}_{3\mathcal{I}}(\alpha_3) \begin{bmatrix} 0 \\ 0 \\ T_3 \end{bmatrix} + \mathcal{A}_{4\mathcal{I}}(\alpha_4) \begin{bmatrix} 0 \\ 0 \\ T_4 \end{bmatrix} \right) + m \begin{bmatrix} 0 \\ 0 \\ g \end{bmatrix} \quad (3.91)$$

On définit les matrices suivantes :

$$\begin{aligned} \mathcal{A}_{1\mathcal{I}}(\alpha_1) &= \begin{bmatrix} \cos(\beta) & -\sin(\beta) \cos(\alpha_1) & \sin(\beta) \sin(\alpha_1) \\ \sin(\beta) & \cos(\beta) \cos(\alpha_1) & -\cos(\beta) \sin(\alpha_1) \\ 0 & \sin(\alpha_1) & \cos(\alpha_1) \end{bmatrix} \\ \mathcal{A}_{2\mathcal{I}}(\alpha_2) &= \begin{bmatrix} -\cos(\beta) & -\sin(\beta) \cos(\alpha_2) & \sin(\beta) \sin(\alpha_2) \\ \sin(\beta) & -\cos(\beta) \cos(\alpha_2) & \cos(\beta) \sin(\alpha_2) \\ 0 & \sin(\alpha_2) & \cos(\alpha_2) \end{bmatrix} \\ \mathcal{A}_{3\mathcal{I}}(\alpha_3) &= \begin{bmatrix} -\cos(\beta) & \sin(\beta) \cos(\alpha_3) & -\sin(\beta) \sin(\alpha_3) \\ -\sin(\beta) & -\cos(\beta) \cos(\alpha_3) & \cos(\beta) \sin(\alpha_3) \\ 0 & \sin(\alpha_3) & \cos(\alpha_3) \end{bmatrix} \\ \mathcal{A}_{4\mathcal{I}}(\alpha_4) &= \begin{bmatrix} \cos(\beta) & \sin(\beta) \cos(\alpha_4) & -\sin(\beta) \sin(\alpha_4) \\ -\sin(\beta) & \cos(\beta) \cos(\alpha_4) & -\cos(\beta) \sin(\alpha_4) \\ 0 & \sin(\alpha_4) & \cos(\alpha_4) \end{bmatrix} \end{aligned} \quad (3.92)$$

On obtient pour le système de manière analogue à la configuration précédente :

$$\begin{aligned} \sum \overrightarrow{F_{I\mathcal{I}}} &= {}^I \mathcal{R}_B \left[ \begin{bmatrix} \sin(\beta) & \cos(\beta) & 1 \end{bmatrix} \odot \begin{bmatrix} s(\alpha_1) & s(\alpha_2) & -s(\alpha_3) & -s(\alpha_4) \\ -s(\alpha_1) & s(\alpha_2) & s(\alpha_3) & -s(\alpha_4) \\ c(\alpha_1) & c(\alpha_2) & c(\alpha_3) & c(\alpha_4) \end{bmatrix} \right] \mathbf{T} - m g \overrightarrow{z_B} \\ &= {}^I \mathcal{R}_B \kappa_{\mathcal{I}}(\alpha, \beta) \mathbf{T} - m g \overrightarrow{z_B} \end{aligned} \quad (3.93)$$

Avec la matrice  $\kappa_{\mathcal{I}}$  définie de la manière suivante :

$$\kappa_{\mathcal{I}}(\alpha, \beta) = \begin{bmatrix} s(\alpha_1)s(\beta) & s(\alpha_2)s(\beta) & -s(\alpha_3)s(\beta) & -s(\alpha_4)s(\beta) \\ -s(\alpha_1)c(\beta) & s(\alpha_2)c(\beta) & s(\alpha_3)c(\beta) & -s(\alpha_4)c(\beta) \\ c(\alpha_1) & c(\alpha_2) & c(\alpha_3) & c(\alpha_4) \end{bmatrix} \quad (3.94)$$

### 3.5.2 Moments

En configuration  $\mathcal{I}$  les moments sur le véhicule sont :

$$\sum \overrightarrow{M_{B\mathcal{I}}} = \mathcal{A}_{1\mathcal{I}}(\alpha_1) \begin{bmatrix} 0 \\ 0 \\ -\delta T_1 \end{bmatrix} + \begin{bmatrix} {}^B R(\beta)_{\alpha_1} \begin{bmatrix} d_1 \\ -d_2 \\ 0 \end{bmatrix} \end{bmatrix} \times \left[ \mathcal{A}_{1\mathcal{I}}(\alpha_1) \begin{bmatrix} 0 \\ 0 \\ T_1 \end{bmatrix} \right] \quad (3.95)$$

$$+ \mathcal{A}_{2\mathcal{I}}(\alpha_2) \begin{bmatrix} 0 \\ 0 \\ \delta T_2 \end{bmatrix} + \begin{bmatrix} {}^B R(\beta)_{\alpha_2} \begin{bmatrix} d_1 \\ d_2 \\ 0 \end{bmatrix} \end{bmatrix} \times \left[ \mathcal{A}_{2\mathcal{I}}(\alpha_2) \begin{bmatrix} 0 \\ 0 \\ T_2 \end{bmatrix} \right] \quad (3.96)$$

$$+ \mathcal{A}_{3\mathcal{I}}(\alpha_3) \begin{bmatrix} 0 \\ 0 \\ -\delta T_3 \end{bmatrix} + \begin{bmatrix} {}^B R(\beta)_{\alpha_3} \begin{bmatrix} d_1 \\ -d_2 \\ 0 \end{bmatrix} \end{bmatrix} \times \left[ \mathcal{A}_{3\mathcal{I}}(\alpha_3) \begin{bmatrix} 0 \\ 0 \\ T_3 \end{bmatrix} \right] \quad (3.97)$$

$$+ \mathcal{A}_{4\mathcal{I}}(\alpha_4) \begin{bmatrix} 0 \\ 0 \\ \delta T_4 \end{bmatrix} + \begin{bmatrix} {}^B R(\beta)_{\alpha_4} \begin{bmatrix} d_1 \\ d_2 \\ 0 \end{bmatrix} \end{bmatrix} \times \left[ \mathcal{A}_{4\mathcal{I}}(\alpha_4) \begin{bmatrix} 0 \\ 0 \\ T_4 \end{bmatrix} \right] + \mathbf{C}_{g-\text{all}\mathcal{I}} \quad (3.98)$$

Note : Le vecteur  $\begin{bmatrix} d_1 \\ d_2(-i) \\ 0 \end{bmatrix}$  est exprimé dans le repère du  $i^{\text{ème}}$  bras et non dans le repère de l'hélice. Pour être exprimé dans le repère du corps il faut donc le multiplier par la matrice de rotation  ${}^B\mathcal{R}_{\alpha_i}$  et non par  $\mathcal{A}_{i\Upsilon}(\alpha_i)$ .

De même que précédemment on définit les équations suivantes :

$$\sum \overrightarrow{M_{B\Upsilon}} = \chi_{\Upsilon}(\alpha, \beta) \mathbf{T} d_1 + \rho_{\Upsilon}(\alpha, \beta) \mathbf{T} d_2 + \varsigma_{\Upsilon}(\alpha, \beta) \mathbf{T} \delta + \mathbf{C}_{\mathbf{g}\text{-all}\Upsilon} \quad (3.99)$$

On définit les matrices suivantes :

$$\chi_{\Upsilon}(\alpha, \beta) = \begin{bmatrix} c(\alpha_1)s(\beta) & c(\alpha_2)s(\beta) & -c(\alpha_3)s(\beta) & -c(\alpha_4)s(\beta) \\ -c(\alpha_1)c(\beta) & c(\alpha_2)c(\beta) & c(\alpha_3)c(\beta) & -c(\alpha_4)c(\beta) \\ -s(\alpha_1) & -s(\alpha_2) & -s(\alpha_3) & -s(\alpha_4) \end{bmatrix} \quad (3.100)$$

$$\rho_{\Upsilon}(\alpha, \beta) = \begin{bmatrix} -c(\alpha_1)c(\beta) & -c(\alpha_2)c(\beta) & c(\alpha_3)c(\beta) & c(\alpha_4)c(\beta) \\ -c(\alpha_1)s(\beta) & c(\alpha_2)s(\beta) & c(\alpha_1)s(\beta) & -c(\alpha_4)s(\beta) \\ 0 & 0 & 0 & 0 \end{bmatrix} \quad (3.101)$$

$$\varsigma_{\Upsilon}(\alpha, \beta) = \begin{bmatrix} -s(\alpha_1)s(\beta) & s(\alpha_2)s(\beta) & s(\alpha_3)s(\beta) & -s(\alpha_4)s(\beta) \\ s(\alpha_1)c(\beta) & s(\alpha_2)c(\beta) & -s(\alpha_3)c(\beta) & -s(\alpha_4)c(\beta) \\ -c(\alpha_1) & c(\alpha_2) & -c(\alpha_3) & c(\alpha_4) \end{bmatrix} \quad (3.102)$$

L'expression des couples gyroscopiques est la suivante pour cette configuration :

$$\begin{aligned} \tau_{G\alpha\Upsilon} &= J_p \left( \mathcal{A}_{1\Upsilon}(\alpha_1) \begin{bmatrix} 0 \\ \dot{\alpha}_1 |\omega_1| \\ 0 \end{bmatrix} + \mathcal{A}_{2\Upsilon}(\alpha_2) \begin{bmatrix} 0 \\ -\dot{\alpha}_2 |\omega_2| \\ 0 \end{bmatrix} + \mathcal{A}_{3\Upsilon}(\alpha_3) \begin{bmatrix} 0 \\ \dot{\alpha}_3 |\omega_3| \\ 0 \end{bmatrix} + \mathcal{A}_{4\Upsilon}(\alpha_4) \begin{bmatrix} 0 \\ -\dot{\alpha}_4 |\omega_4| \\ 0 \end{bmatrix} \right) \\ &= J_p \begin{bmatrix} -e_1 c(\alpha_1) s(\beta) & e_2 c(\alpha_2) s(\beta) & e_3 c(\alpha_3) s(\beta) & -e_4 c(\alpha_4) s(\beta) \\ e_1 c(\alpha_1) c(\beta) & e_2 c(\alpha_2) c(\beta) & -e_3 c(\alpha_3) c(\beta) & -e_4 c(\alpha_4) c(\beta) \\ e_1 s(\alpha_1) & -e_2 s(\alpha_2) & e_3 s(\alpha_3) & -e_4 s(\alpha_4) \end{bmatrix} \end{aligned} \quad (3.103)$$

### 3.5.3 Synthèse

On propose de factoriser le vecteur avec les termes en  $\alpha$  en faisant un changement de variable comme suit :

$$\mathbf{u} = \begin{bmatrix} u_1 \\ u_2 \\ u_3 \\ u_4 \\ u_5 \\ u_6 \end{bmatrix} = \mathcal{O}_{\Upsilon} \begin{bmatrix} T_1 \sin(\alpha_1) \\ T_1 \cos(\alpha_1) \\ T_2 \sin(\alpha_2) \\ T_2 \cos(\alpha_2) \\ T_3 \sin(\alpha_3) \\ T_3 \cos(\alpha_3) \\ T_4 \sin(\alpha_4) \\ T_4 \cos(\alpha_4) \end{bmatrix} \quad (3.104)$$

Avec  $\mathcal{O}_{\Upsilon}$  définit de la manière suivante :

$$\mathcal{O}_{\Upsilon} = \begin{bmatrix} s(\beta) & 0 & s(\beta) & 0 & -s(\beta) & 0 & -s(\beta) & 0 \\ -c(\beta) & 0 & c(\beta) & 0 & c(\beta) & 0 & -c(\beta) & 0 \\ 0 & 1 & 0 & 1 & 0 & 1 & 0 & 1 \\ -s(\beta)\delta & d_1 s(\beta) - d_2 c(\beta) & s(\beta)\delta & d_1 s(\beta) - d_2 c(\beta) & s(\beta)\delta & -d_1 s(\beta) + d_2 c(\beta) & -s(\beta)\delta & -d_1 s(\beta) + d_2 c(\beta) \\ c(\beta)\delta & -d_1 c(\beta) - d_2 s(\beta) & c(\beta)\delta & d_1 c(\beta) + d_2 s(\beta) & -c(\beta)\delta & d_1 c(\beta) + d_2 s(\beta) & -c(\beta)\delta & -d_1 c(\beta) - d_2 s(\beta) \\ -d_1 & -\delta & -d_1 & \delta & -d_1 & -\delta & -d_1 & \delta \end{bmatrix} \quad (3.105)$$

Concernant l'angle  $\beta$ , on voit sur la matrice (3.105) que pour une rafale de vent latérale, c'est le terme en  $\cos(\beta)$  qui intervient. Soit une rafale de vent de 40 km/h (11 m/s), d'après [60] on peut en déduire l'expression de la force latérale appliquée au drone à partir de l'expression suivante :

$$F_v \approx \frac{1}{2} C_d \rho S V_v^2 \quad (3.106)$$

Avec  $F_v$  la force du vent sur le drone,  $C_d = 0.1$  le coefficient de traînée,  $\rho = 1.2$  la densité de l'air et  $S = 0.2$  la surface latérale du drone, et finalement  $V_v$  la vitesse du vent.

Ce calcul donne une force latérale de 300g appliquée sur le drone.

La matrice (3.105) ci-dessus montre que le drone, avec deux hélices qui poussent à 700g chacune pour un angle de  $60^\circ$ , peut rejeter une perturbation de  $2 * \cos(60) * 0.7 = 0.7$  700g. Cela permet donc de résister à des vents violents tout en gardant une mobilité sur l'axe  $\vec{y}_B$ .

Cette configuration est plus complexe que la précédente par deux points. Premièrement, l'utilisation d'un angle  $\beta$  change la forme des matrices du modèle. La subtilité des angles dans l'équation (3.87) est un point auquel il faut faire attention. Contrairement aux configurations précédentes, on ne peut pas simplement ajouter  $90^\circ$  d'un bras à l'autre. Cela explique l'apparition des termes  $\pi - \beta$  que l'on n'avait pas précédemment. Deuxièmement, le fait de désaxer les bras représente la vraie innovation du modèle et ouvre un champ de possibilités quasi-infini pour développer différents types de véhicules. Il a d'ailleurs été envisagé de faire une étude sur un système factice possédant une forme particulièrement complexe pour montrer le fonctionnement générique de cette méthode de calcul analytique. Cela aurait également permis de réappliquer la méthode de linéarisation du chapitre suivant, mais faute de temps cela n'a pas été fait.

La section suivante permet de faire un comparatif énergétique entre les deux dernières configurations proposées.

### 3.6 Énergie

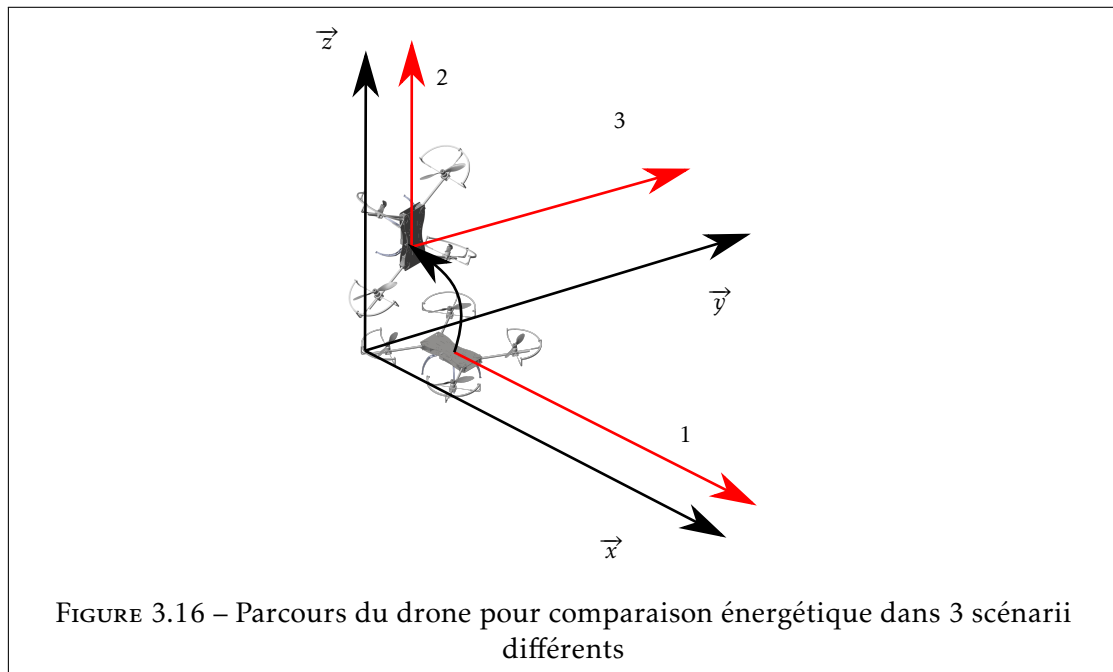
On choisit de comparer la configuration en  $\times$  avec celle en  $\Upsilon$  (qui correspondent aux deux prototypes réalisés). Pour se faire on mesure le travail dépensé par les machines pour effectuer un même trajet. L'expression de celui-ci est la suivante :

$$\mathcal{W} = \int_0^l \mathcal{F} dl \quad (3.107)$$

Avec  $\mathcal{F}$  la force (N) exercée par le système et  $\mathcal{W}$  le travail en Joules,  $l$  étant la distance à parcourir.



L'essai se fera par simulation en prenant un scénario de vol particulier. Celui-ci est décomposé en trois parties. Dans les trois cas la distance à parcourir est d'un mètre. Premièrement, un vol linéaire est fait en restant dans un plan horizontal. Deuxièmement, on placera le drone à la verticale sans vitesse linéaire initiale. Une fois stabilisé, le drone devra voler sur l'axe  $\vec{z}_I$ . Enfin, dans un troisième essai on répète l'expérience sur l'axe  $\vec{y}_I$ . En faisant l'intégrale de la distance parcourue avec la force, on peut évaluer le travail nécessaire pour effectuer le trajet et ainsi comparer les deux configurations sur le même graphique. On précise également que les deux drones ont la même masse durant la simulation et évoluent dans les mêmes conditions. Les résultats sont présentés en Figure 3.16 et dans le Tableau 3.5.



| Essai | Travail en configuration $\times$ (J) | Travail en configuration $\checkmark$ (J) |
|-------|---------------------------------------|---|
| 1     | 20.94                                 | 20.94                                     |
| 2     | 29.77                                 | 24.24                                     |
| 3     | 29.76                                 | 23.86                                     |

TABLEAU 3.5 – Consommation énergétique dans 3 scénarii différents

Ces graphiques montrent clairement que la configuration  $\checkmark$  est plus adaptée par deux aspects par rapport à la configuration en  $\times$ . Premièrement, on garde une versatilité du système en ayant un angle  $\beta$  ajustable au besoin du scénario de vol. Si lors d'une mission il est nécessaire d'effectuer un maximum de déplacements horizontaux c'est possible (ou tout simplement pour pouvoir parer un vent important). Deuxièmement, le système est plus optimisé pour un vol vertical que la configuration en  $\times$  mais permet tout de même de garder une contrôlabilité latérale. Cette configuration représente donc un bon compromis.

On remarque également qu'il aurait été tout à fait possible de développer une solution avec davantage d'actionneurs permettant d'optimiser encore l'utilisation énergétique, comme les configurations avec 2 servomoteurs par bras ; ou encore une configuration en  $\checkmark$  avec deux moteurs latéraux sur les cotés.

Cependant, le poids engendré par l'ajout d'actionneurs diminue les performances du système s'ils ne sont pas utilisés.

### 3.7 Conclusion

L'analyse mécanique faite dans ce chapitre met clairement en avant le fait que le système de drone à bras inclinable est totalement actionné et que chaque DDL peut être contrôlé indépendamment des autres. Une redondance sur le lacet peut également être mis en avant. Il est tout à fait possible d'orienter le drone en tournant les bras ou en utilisant l'effet de trainée aérodynamique. Concernant la configuration en "+", en vol à  $90^\circ$  (sur le roulis ou le tangage), deux des bras se trouvent en position verticale. Sans consigne de changement de position la seule force à compenser est la pesanteur. Or quand les bras sont mis à la verticale, ils ne peuvent pas compenser cette force et deviennent donc inutiles. En considérant cette restriction, on analyse la configuration suivante en  $\times$  qui donne des résultats intéressants et permet en théorie un vol vertical si les actionneurs sont dimensionnés pour. Enfin, l'analyse énergétique démontre l'intérêt de la configuration finale du drone en configuration  $\lambda$ , ce qui amène clairement une réponse à la deuxième problématique : Quelle forme doit avoir le robot qui effectue les mesures sur ouvrage ?

Le chapitre suivant propose une loi de commande permettant de commander les quatre configurations décrites ci-dessus. Le choix des chapitres s'est fait par thématique, il peut donc paraître surprenant de voir quatre configurations différentes que l'on cherchera toutes à commander. En réalité, le travail décrit aussi bien dans ce chapitre que dans le suivant suit une réelle progression chronologique. C'est pour cette raison que l'on ne construit pas qu'un seul prototype mais bien deux. Le premier en forme de  $\times$  permet d'identifier tous les points à optimiser qui sont alors inclus dans la deuxième version.

Pour commander le système on choisit de faire un changement de variables permettant d'obtenir un système totalement actionné au lieu de sur-actionné comme il est nativement. On appliquera par la suite la méthode de linéarisant entrée-état au système, pour faciliter le test de différentes loi de commandes. Plus de détails sur cette loi de commande seront donnés en Section 4.1.1. Une fois le système linéarisé, on peut appliquer une loi de commande sur le système simplifié. Pour le PVTOL-MI, On choisira d'utiliser une loi de commande linéaire PD (voir Section 4.1.2) ainsi qu'une approche par saturations imbriquées (voir Section 4.1.3) pour comparaison. Ces résultats ont donné lieu à une publication, voir [61]. Concernant les trois autres configurations on proposera de tester les performances du système avec une loi de commande linéaire puis robuste de type mode glissant super twisting.



# Commande

Afin de répondre à la deuxième problématique, on cherche tout d'abord à réaliser une stabilisation du système simplifié représenté par un PVTOL à bras inclinables. On présente une manière simple et intuitive de rendre le système simple à stabiliser par linéarisation entrée-états. Un rappel sur la technique est fait et est ensuite utilisé sur le système 2D. Une fois le système linéarisé, deux lois de commandes y seront appliquées. Pour finir sur ce système, on propose une méthode pour obtenir les valeurs à affecter aux actionneurs en fonction des signaux de commande donnés par l'algorithme.

Une généralisation aux trois dimensions permet par la suite de se rapprocher du système réel. Après le rappel du système quadricoptère classique, on propose de détailler la méthode d'observation des états qui ne sont pas directement mesurables par les capteurs.

La même méthode de linéarisation que pour le modèle 2D est employée pour commander le système. Une loi de commande robuste est présentée et appliquée en simulation et durant les expérimentations.

Une analyse sur les singularités du système avec les remèdes que l'on se propose d'apporter constitue également un point important qui a permis de choisir la forme finale du drone. Enfin, on considère une analyse de balourd dû à l'outil d'inspection, permettant de compenser dynamiquement le déséquilibre engendré par les masses déportés du système.

## 4.1 Algorithme de commande pour le PVTOL-MI

De par sa simplicité, le PVTOL est couramment utilisé en automatique pour tester différentes lois de commandes. Pour citer quelques exemples, [62], [63] utilise le principe de platitude des systèmes pour effectuer un suivi en trajectoire. [64]-[67] proposent un algorithme de saturations imbriquées, similaire à celui utilisé ci-après. Une analyse de stabilité y est également décrite. La commande sans modèle est abordée en [68] en appliquant un PID "intelligent".

[69] propose une méthode intéressante de découplage de système. Un exemple de linéarisation entrée-sortie en prenant en compte le désaxage des moteurs (souvent noté  $\epsilon$ ) est présenté et sera également une base pour les travaux ci-après. [31] sera également utilisé par la suite car un contrôleur PD avec une linéarisation du système y est proposé.

On décrit maintenant l'algorithme de commande utilisé pour contrôler le PVTOL-MI. On se basera sur le modèle mécanique exhaustif proposé dans la section précédente :

$$\begin{bmatrix} m.\ddot{x} \\ m.\ddot{y} \\ J.\dot{\phi} \end{bmatrix} = \mathcal{O} \cdot \begin{bmatrix} u_1 \\ u_2 \\ u_3 \end{bmatrix} - \begin{bmatrix} 0 \\ m.g \\ 0 \end{bmatrix} \quad (4.1)$$

Afin de garantir une stabilité du système, on souhaite obtenir la dynamique en boucle fermée suivante :

$$\ddot{\tilde{\Lambda}} + r(\tilde{\Lambda}, \dot{\tilde{\Lambda}}) = 0 \quad (4.2)$$

avec le vecteur des mesures d'états pour les positions en 2D, définit par  $\Lambda = [x, y, \phi]^T$  et  $r(\tilde{\Lambda}, \dot{\tilde{\Lambda}})$  une fonction de commande, qui sera définie par la suite. Le but de cette fonction est d'obtenir une équation différentielle qui mènera le système à une stabilité en boucle fermée. On étudiera par la suite plusieurs possibilités de loi de commande que l'on pourrait utiliser. On définit également  $\tilde{\Lambda}$  l'erreur d'état du système comme suit :

$$\tilde{\Lambda} = \Lambda - \Lambda_r \quad (4.3)$$

avec  $\Lambda_r$  le vecteur des références que le système doit suivre sur les trois états du système. Si on calcule la deuxième dérivée de l'équation (4.3) et qu'elle est injectée dans l'équation (4.2), on obtient alors le système suivant

$$\ddot{\tilde{\Lambda}} = \ddot{\Lambda}_r - r(\tilde{\Lambda}, \dot{\tilde{\Lambda}}) \quad (4.4)$$

Note : dans notre cas on considère que les références de degrés plus élevés sont constantes  $\ddot{\chi}_r, \dot{\Lambda}_r = 0$ . En effet, on effectue ici une régulation et non un suivi de trajectoire, donc la vitesse et l'accélération désirée sont nulles. Cette équation correspond à un système du deuxième ordre qui peut facilement être stabilisé si  $r(\tilde{\chi}, \dot{\tilde{\Lambda}})$  est bien choisi.

### 4.1.1 Linéarisation exacte d'un système

Pour notre système, l'algorithme proposé est celui de la linéarisation d'entrée-sortie. On commencera par un descriptif rapide de la méthode avec un exemple. On l'appliquera ensuite directement à notre système. L'objectif est de retrouver un système avec une forme similaire à celle décrite ci-dessus.

Remarque : ce n'est bien sûr pas la seule manière de commander le système. On pourrait tout à fait utiliser une approche par platitude ou backstepping par exemple.

#### 4.1.1.1 Généralités

Commençons par quelques généralités sur la méthode de linéarisation. Soit le système décrit sous la représentation d'état de la manière suivante :

$$\begin{cases} \mathbf{X}_{k+1} = \mathcal{A}\mathbf{X}_k + \mathcal{B}\mathbf{U}_k \\ \mathbf{Y}_k = \mathcal{C}\mathbf{X}_k + \mathcal{D}\mathbf{U}_k \end{cases} \quad (4.5)$$

Avec  $\mathbf{X}$  le vecteur d'états du système,  $\mathbf{Y}$  le vecteur des sorties et  $\mathbf{U}$  le vecteur des entrées de commande. Les fonctions  $f(\mathbf{X})$ ,  $g(\mathbf{X})$  et  $h(\mathbf{X})$  sont liées à la mécanique du système.

Le principe de la loi de commande consiste à transformer un système non linéaire en un système totalement ou partiellement linéaire afin de pouvoir le contrôler plus facilement. Deux possibilités s'offrent alors à nous, la linéarisation entrée-état et entrée-sortie. Dans le premier cas on cherche à dériver les états jusqu'à retrouver les entrées de commande. Dans le deuxième cas on dérive les sorties jusqu'à faire apparaître les entrées. On se concentrera sur le premier type de linéarisation qui sera celui utilisé dans les sections suivantes<sup>1</sup>. Pour ce faire, on cherche à exprimer le système dans la forme :

$$\dot{\mathbf{X}} = \mathcal{A}\mathbf{X} + \mathcal{B}\gamma(\mathbf{X})(\mathbf{U} - \alpha(\mathbf{X})) \quad (4.6)$$

avec  $\mathcal{A}$  et  $\mathcal{B}$  deux matrices définies par le système.

On pourra s'aider d'un changement de variable que l'on peut noter

$$\mathbf{Z} = M(\mathbf{X}) \quad (4.7)$$

On propose alors la loi de commande de la forme suivante :

$$\mathbf{U} = \alpha(\mathbf{X}) + \beta(\mathbf{X})v(\mathbf{X}) \quad (4.8)$$

avec  $\alpha(\mathbf{X})$  et  $\beta(\mathbf{X}) = \gamma^{-1}(\mathbf{X})$  à identifier après avoir fait les calculs, plus d'informations sur cette méthode dans [70]. La sous-section suivante propose d'appliquer cette méthode directement sur le système PVTOL-MI.

#### 4.1.1.2 Application au système

On réécrit le système (4.1) sous forme d'états décrite en (4.6) comme suit :

$$\begin{cases} \mathbb{J}_{2D} \dot{\mathbf{X}} = \mathbb{J}_{2D} \begin{bmatrix} 0 \\ g \\ 0 \end{bmatrix} + \mathcal{O}\mathbf{U} \\ \mathbf{Y} = h(\mathbf{X}) \end{cases} \quad (4.9)$$

avec  $h(\mathbf{X})$  une fonction liée au système. Le vecteur d'états est définis de la manière suivante :

$$\mathbf{X} = [x \quad y \quad \phi \quad \dot{x} \quad \dot{y} \quad \dot{\phi}]^T \quad (4.10)$$

On définit les termes d'inerties regroupés dans la matrice  $\mathbb{J}$  ainsi que le vecteur de pesanteur  $\mathbf{G}$  :

$$\mathbb{J}_{2D} = \begin{bmatrix} m & 0 & 0 \\ 0 & m & 0 \\ 0 & 0 & J \end{bmatrix} \quad \mathbf{G} = \begin{bmatrix} 0 \\ g \\ 0 \end{bmatrix} \quad (4.11)$$

On réécrit alors simplement l'équation d'états (4.9) de la manière suivante :

$$\dot{\mathbf{X}} = \mathbb{J}^{-1}\mathcal{O}(\mathbf{U} - \mathcal{O}^{-1}\mathbb{J}_{2D}\mathbf{G}) \quad (4.12)$$

L'identification des termes  $\alpha(\mathbf{X})$  et  $\gamma(\mathbf{X})$  peut se faire très facilement, à savoir :

$$\begin{aligned} \alpha(\mathbf{X}) &= -\mathcal{O}^{-1}\mathbb{J}\mathbf{G} \\ \gamma(\mathbf{X}) &= \mathbb{J}_{2D}^{-1}\mathcal{O} \end{aligned} \quad (4.13)$$

---

1. En robotique, les sorties coïncident très souvent avec les états

A partir de (4.8) on peut donc proposer la loi de fermeture de boucle suivante :

$$\mathbf{U} = \mathcal{O}^{-1} \mathbb{J}_{2D} \mathbf{G} + \mathcal{O}^{-1} \mathbb{J}_{2D} r(\tilde{\mathbf{\Lambda}}, \dot{\tilde{\mathbf{\Lambda}}}) \quad (4.14)$$

avec  $r(\chi, \dot{\chi})$  une loi de fermeture de boucle au choix, qui sera discutée plus tard.

On remarque que la matrice  $\mathcal{O}$  de l'équation (3.17) est directement inversible. En reprenant la forme d'équation de (4.1), on propose finalement la loi de commande suivante :

$$\begin{bmatrix} u_1 \\ u_2 \\ u_3 \end{bmatrix} = \mathcal{O}^{-1} \mathbb{J}_{2D} \left[ r(\tilde{\mathbf{\Lambda}}, \dot{\tilde{\mathbf{\Lambda}}}) + \begin{bmatrix} 0 \\ g \\ 0 \end{bmatrix} \right] \quad (4.15)$$

Le terme  $r(\tilde{\mathbf{\Lambda}}, \dot{\tilde{\mathbf{\Lambda}}})$  sera remplacé en fonction de l'algorithme choisi. Deux lois seront testées durant les simulations pour ce système. La première est une loi de commande linéaire du type PD<sup>2</sup>. Il est tout à fait possible de garantir une stabilité avec ce genre de commande puisque le système a été linéarisé (voir preuve de stabilité en Annexe C.2.1). On décrira également une loi de commande non-linéaire par saturations emboîtés.

#### 4.1.2 Loi de commande linéaire

Les contrôleurs linéaires types PD, Proportionnel Intégrale (PI) ou encore Proportionnel Intégrale Dérivé (PID) sont les contrôleurs les plus utilisés en automatique. Le principe étant très abordable, ils sont également faciles à implémenter. Ce type d'algorithme ne nécessite pas de grandes connaissances mathématiques ni ne demande trop de ressources aux contrôleurs sur lesquels ils sont implémentés.

Le principe de base consiste à créer une erreur de commande à laquelle on appliquera un gain et qui sera directement la loi de commande. L'erreur est construite en faisant la différence entre la référence que souhaite l'utilisateur et la mesure de la grandeur. Il choisit de calculer l'erreur sur l'état qu'il souhaite asservir (position, vitesse, accélération ou autre). En fonction de ce choix la signification des termes P, I et D est différente. Ces termes désignent les états du systèmes que l'on utilise pour effectuer la régulation.

Si on prend le problème du pendule, on peut régler sa vitesse de rotation ou sa position. Dans le premier cas le P désigne sa vitesse alors que dans le deuxième il s'agit de sa position. Si on cherche à le réguler en position on doit avoir une mesure (ou observation) de sa vitesse et de l'intégrale de sa position.

Les effets des facteurs P I et D sont bien connus de la littérature. Pour simplifier, P désigne la puissance générale appliquée au système pour aller de la condition initiale à la référence. Le système en boucle fermé avec un contrôleur P seul aurait tendance à osciller sauf si le système est mécaniquement amorti. D désigne la dérivée, donc liée à la vitesse du système. Si le gain pour ce facteur est élevé on va ralentir l'arrivée à la référence en amortissant le système. Enfin I permet d'intégrer l'erreur et donc de compenser une erreur statique. Globalement si les facteurs sont trop grands, des oscillations émergeront dans le système pour différentes raisons.

Dans le cas de notre PVTOL on utilise principalement un contrôleur PD<sup>3</sup>. La régulation se fait sur les positions.

2. Proportionnel dérivatif

3. On peut également proposer un PID pour compenser l'erreur statique mais un contrôleur PD seul permet de donner de bonnes performances plus rapidement

L'équation générale de la loi de commande s'écrit :

$$r(\tilde{\Lambda}, \dot{\tilde{\Lambda}}) = [-k_1 \dot{\tilde{\Lambda}} - k_2 \tilde{\Lambda}] = \begin{bmatrix} -k_{11} \dot{\tilde{x}} - k_{21} \tilde{x} \\ -k_{12} \dot{\tilde{y}} - k_{22} \tilde{y} \\ -k_{13} \dot{\tilde{\phi}} - k_{23} \tilde{\phi} \end{bmatrix} \quad (4.16)$$

en remplaçant (4.16) dans (4.15), elle-même remplacée dans (4.1) l'expression devient :

$$\begin{bmatrix} \ddot{x} + k_{11} \dot{\tilde{x}} + k_{21} \tilde{x} \\ \ddot{y} + k_{12} \dot{\tilde{y}} + k_{22} \tilde{y} \\ \ddot{\phi} + k_{13} \dot{\tilde{\phi}} + k_{23} \tilde{\phi} \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \\ 0 \end{bmatrix} \quad (4.17)$$

On obtient alors un système linéaire du deuxième ordre, où les gains  $k_{ij}$  de l'algorithme sont choisis de manière à obtenir une dynamique stable du système en boucle fermée.

Pour choisir les coefficients du contrôleur, on peut écrire l'équation dans le domaine de Laplace. Pour rappel on définit  $p$  tel qu'à une fonction  $f(t)$  on associe sa transformée  $F(p)$  par l'équation :

$$F(p) = \int_0^{\infty} f(t) e^{-pt} dt \quad (4.18)$$

Si on regarde l'équation (4.17) sur un seul axe on obtient alors :

$$p^2 X - p k_d X - k_p X = 0 \quad (4.19)$$

Avec  $X$  l'état sous forme discrète. Il est connu que la fonction de transfert d'un système du second ordre est de la forme suivante :

$$p^2 X + 2\zeta \omega_0 p X + \omega_0^2 X = 0 \quad (4.20)$$

Avec  $\zeta$  le coefficient d'amortissement du système et  $\omega_0$  la pulsation caractéristique du système. On peut noter :

$$\tau = \frac{1}{\zeta \omega_0} \quad (4.21)$$

La constante de temps du système à 5% de la référence. On peut donc facilement calculer  $k_d$  et  $k_p$  en choisissant une constante de temps et un coefficient d'amortissement, comme le montrent les équations suivantes :

$$\omega_0 = \frac{1}{\tau \zeta} \quad (4.22)$$

$$k_p = \omega^2 \quad (4.23)$$

$$k_d = 2\zeta \omega_0 \quad (4.24)$$

Par défaut on peut choisir une constante de temps d'une seconde et un coefficient d'amortissement à 0.7. Cet algorithme présente une première solution viable pour contrôler le système et en assurer sa stabilité. Cependant, celui-ci trouve ses limites sur différents aspects. La section suivante traite de la limite de la loi de commande linéaire lorsque la consigne est éloignée.



### 4.1.3 Algorithme de saturation emboîtés

L'intérêt de l'algorithme par saturations emboîtés (*nested saturation* en anglais) est de garantir un comportement borné sur les états. Contrairement à un algorithme linéaire, l'erreur n'est pas directement proportionnelle à la distance entre la référence et le points de mesure. Cette erreur sera plutôt saturée permettant de garantir un comportement moins violent pour des erreurs élevés.

Pour illustrer ce principe prenons un exemple simple. Soit un drone classique que l'on souhaite faire voler à une grande distance du point de départ. Dans cette configuration, sans saturation, l'angle d'inclinaison est proportionnel à la distance à parcourir. La surcompensation d'inclinaison du drone le fera tomber à la verticale ou peut-être même partir en arrière. Dans tous les cas le système ne rejoindra très probablement pas sa consigne. Rajouter simplement une saturation dans la loi de commande ne permet pas de garantir une stabilité du système. Au lieu de ça, A. Teel [71] propose une technique d'imbrication des états permettant de saturer le système état par état et de garantir la stabilité. Des exemples d'applications sont donnés dans [35], [72]. On propose d'utiliser la fonction de saturation  $\sigma$  (cf. Annexe C.1.1).

Remarque : le même algorithme est utilisable avec une fonction arc-tangente permettant d'obtenir un comportement plus doux en entrée et sortie du domaine de saturation.

On propose donc de définir la fonction  $r$  de l'équation (4.15) de la manière suivante pour imposer un comportement stable en boucle fermée :

$$r(\tilde{\Lambda}, \dot{\tilde{\Lambda}}) = \begin{bmatrix} -\sigma_{11}(k_{S11} \cdot \dot{\tilde{x}} + \sigma_{21}(k_{S11} \cdot \dot{\tilde{x}} + k_{S21} \cdot \tilde{x})) \\ -\sigma_{12}(k_{S12} \cdot \dot{\tilde{y}} + \sigma_{22}(k_{S12} \cdot \dot{\tilde{y}} + k_{S22} \cdot \tilde{y})) \\ -\sigma_{13}(k_{S13} \cdot \dot{\tilde{\phi}} + \sigma_{23}(k_{S13} \cdot \dot{\tilde{\phi}} + k_{S23} \cdot \tilde{\phi})) \end{bmatrix} \quad (4.25)$$

Les gains  $k_{Sij}$  seront choisis afin d'obtenir une convergence rapide vers les références d'états. Cette convergence est définie par un temps de réponse que l'utilisateur devra définir (<1 seconde par exemple). La preuve de stabilité est également donnée en Annexe C.2.2. Concernant les valeurs de saturations à imposer, pour garantir la stabilité du système, l'algorithme impose d'avoir  $\sigma_{1i} > 2 * \sigma_{2i}$  d'une part. D'autre part, on choisira les saturations en fonction des saturations réelles du système. Dans notre cas il s'agit des limites des actionneurs, et plus précisément de la poussée maximale possible par les moteurs.

Remarque : L'algorithme à été appliqué à deux états dans notre cas, mais on peut tout à fait généraliser la commande à  $n$  états :

$$u = -\sigma_n(k_{Sn} y_n + \sigma_{n-1}(k_{Sn-1} y_{n-1} + \sigma_{n-2}(\dots + \sigma_1(k_{S1} y_1)))) \quad (4.26)$$

avec  $y_{n-i}$  un changement de variable défini de la manière suivante :

$$y_{n-i} = \sum_{j=0}^i \frac{i!}{j!(i-j)!} x_{n-i} \quad (4.27)$$

$n$  étant le nombre d'états du système et  $0 \leq i \leq n-1$ .

Si on prend l'exemple d'un quadruple intégrateur qui correspond au modèle simplifié d'un quadricoptère classique, on obtiens alors pour cette équation la commande suivante :

$$\begin{aligned} u &= -\sigma_4(y_4 + \sigma_3(y_3 + \sigma_2(y_2 + \sigma_1(y_1)))) \\ &= -\sigma_4(x_4 + \sigma_3((x_4 + x_3) + \sigma_2((x_3 + 2 * x_2 + x_1) + \sigma_1(x_4 + 3 * x_3 + 3 * x_2 + x_1)))) \end{aligned} \quad (4.28)$$

#### 4.1.4 Des entrées de commande virtuelles aux actionneurs

On rappelle que ce sont les entrées de commandes virtuelles  $u_i$  qui sont utilisées dans l'algorithme. Cette entrée ne correspondent pas directement aux entrées des actionneurs  $T_1, T_2, \delta, \gamma$ . Pourtant, c'est bien des commandes sur les actionneurs qui sont nécessaires in fine pour contrôler le système. En simulation aussi, le système possède des entrées sur les liaisons pour symboliser les actionneur (voir chapitre suivant). Il faut donc proposer un lien direct entre les entrées virtuelles et celles des actionneurs. Pour commencer, on propose de réécrire l'équation (3.15) de la manière suivante :

$$\begin{bmatrix} u_1 \\ u_2 \\ u_3 \end{bmatrix} = \begin{bmatrix} 1 & 1 & 0 & 0 \\ 0 & 0 & 1 & 1 \\ 1 & -1 & 0 & 0 \end{bmatrix} \cdot \begin{bmatrix} T_1 \cdot \cos(\alpha_1) \\ T_2 \cdot \cos(\alpha_2) \\ T_1 \cdot \sin(\alpha_1) \\ T_2 \cdot \sin(\alpha_2) \end{bmatrix} \quad (4.29)$$

Par la suite on utilisera le principe de pseudo-inverse pour gérer le surnombre d'actionneurs comme abordé dans [73]. On notera la pseudo-inverse de Moore-Penrose <sup>†</sup>. On précise son expression mathématique en Annexe C.1.3. Cet outil mathématique permet d'inverser une matrice non-carrée, en faisant un choix permettant de gérer l'indétermination des solutions. Le résultat est le suivant :

$$\begin{bmatrix} 1 & 1 & 0 & 0 \\ 0 & 0 & 1 & 1 \\ 1 & -1 & 0 & 0 \end{bmatrix}^{\dagger} = \begin{bmatrix} 0.5 & 0 & 0.5 \\ 0.5 & 0 & -0.5 \\ 0 & 0.5 & 0 \\ 0 & 0.5 & 0 \end{bmatrix} \quad (4.30)$$

Enfin, en inversant (4.29) avec l'équation précédente, on obtient l'égalité suivante :

$$\begin{bmatrix} T_1 \cdot \cos(\alpha_1) \\ T_2 \cdot \cos(\alpha_2) \\ T_1 \cdot \sin(\alpha_1) \\ T_2 \cdot \sin(\alpha_2) \end{bmatrix} = \begin{bmatrix} 0.5 & 0 & 0.5 \\ 0.5 & 0 & -0.5 \\ 0 & 0.5 & 0 \\ 0 & 0.5 & 0 \end{bmatrix} \cdot \begin{bmatrix} u_1 \\ u_2 \\ u_3 \end{bmatrix} \quad (4.31)$$

Il devient alors évident de trouver les valeurs des actionneurs en fonction des entrées virtuelles de commande, il suffit d'effectuer les opérations comme suit :

$$\mathbf{Y}_{2D} = \begin{bmatrix} T_1 \\ T_2 \\ \alpha_1 \\ \alpha_2 \end{bmatrix} = \begin{bmatrix} \sqrt{(k_p \cdot (u_1 + u_3))^2 + (k_p \cdot u_2)^2} \\ \sqrt{(k_p \cdot (u_1 - u_3))^2 + (k_p \cdot u_2)^2} \\ \text{atan2}\left(\frac{k_p \cdot u_2}{k_p \cdot (u_1 + u_3)}\right) \\ \text{atan2}\left(\frac{k_p \cdot u_2}{k_p \cdot (u_1 - u_3)}\right) \end{bmatrix} \quad (4.32)$$

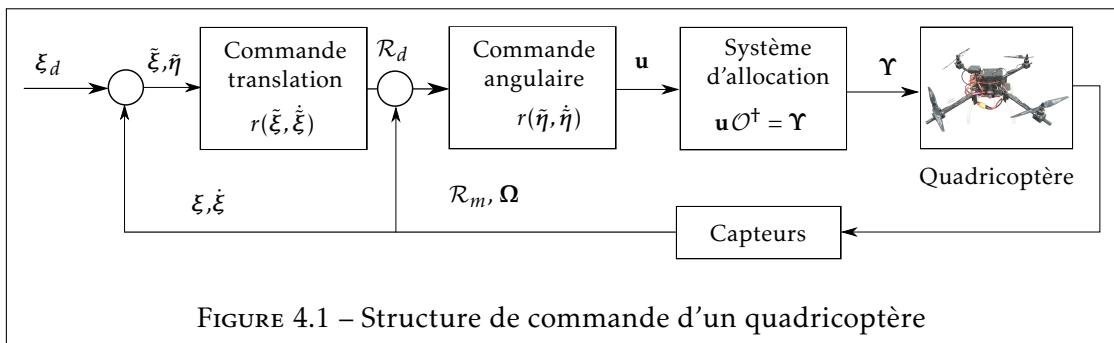
avec  $k_p = 0,5$  et  $\mathbf{Y}_{2D}$  le vecteur des entrées de commande des actionneurs pour le système 2D.

Note : à partir de l'équation (4.32) on peut calculer les valeurs des saturations à affecter pour garantir les limites des actionneurs.

Suite à cette description, des tests sont réalisés en simulation et présentés dans le chapitre suivant. On fera une comparaison entre les deux lois de commande. Les résultats ont donné lieu à un article scientifique [61]. Par la suite on propose de commander les configurations  $\times$  et  $\tilde{\chi}$  de manière analogue à celle proposée pour ce PVTOL particulier. Une généralisation aux 3 dimensions sera donc faite. Ce sera l'occasion de rappeler la méthode de contrôle d'un drone classique, ce qui sera un préliminaire pour la commande des deux configurations décrites.

## 4.2 Généralisation de la commande aux 3 dimensions

La commande d'un quadricoptère est tout aussi largement analysée et décrite dans la littérature que celle du PVTOL. Dans [74], [75] on retrouve un contrôle classique en cascade. Le principe général pour ces véhicules est de commander l'orientation du drone dans une boucle rapide ( $>50\text{Hz}$ , de préférence même  $>200\text{Hz}$ ); la commande en position se fera alors en donnant une consigne d'orientation de l'appareil. Une fois l'aéronef incliné, il se déplacera dans la direction souhaitée. L'appellation cascade vient du fait que l'on contrôle l'inclinaison pour obtenir une position. Le principe est rappelé en Figure 4.1.



Le principe résumé ci-dessus est directement lié au modèle mécanique où l'on voit clairement que la position sur  $\vec{x}_B$  et  $\vec{y}_B$  est liée à l'orientation du véhicule. Le drone se déplace en chutant de manière contrôlée. Afin d'améliorer les performances du système, plusieurs lois de commandes différentes ont été comparées. [76] propose par exemple d'utiliser une loi de commande linéaire optimale LQT/LQR permettant de trouver un optimum entre le suivi de trajectoire et la consommation énergétique. [72] généralise la commande par saturation utilisée sur le PVTOL dans un système 3D donc plus complexe. [77] propose une commande feedback-linéarisation permettant d'obtenir un système linéaire et donc plus facilement contrôlable. Une combinaison des deux dans [78] permet d'appliquer une loi de commande par saturation sur un système linéarisé.

Concernant les systèmes à bras inclinables, un article très intéressant en [23] propose de linéariser le modèle d'un hexacoptère pour ensuite lui appliquer un simple PD en utilisant les quaternions comme outil mathématique pour calculer l'erreur d'orientation. Cependant, d'autres configurations particulières de robots peuvent être trouvées, comme par exemple [79] qui propose un véhicule en forme de disque permettant d'effectuer des manœuvres sur les 6 degrés de libertés. On peut également citer [47] qui contrôle un quadricoptère à bras inclinables. Cependant, afin de limiter les in-homogénéités dans les niveaux différentiels dûs au modèle particulier, l'auteur propose de considérer les ordres supérieurs du modèle dans la commande.

Ainsi, le système est dérivé et contrôlé par "feedback-linearisation". D'autres formes de commandes auront été appliquées au système comme [24] qui propose une commande par platitude, ou encore [49] qui utilise une commande par mode glissant sur un drone à bras inclinables.

Pour notre problématique, la commande en trois dimensions se fera de manière analogue à celle en 2D pour le PVTOL-MI précédemment décrit. On procèdera par une linéarisation entrée-états afin d'obtenir un système linéaire auquel on pourra affecter n'importe quelle loi de commande. On reviendra notamment sur une loi de commande linéaire ainsi que sur une approche robuste par SMCST. Cette deuxième loi de commande est choisie pour sa robustesse face aux perturbations ce qui est un point important compte tenu des circonstances d'utilisation.

Concernant notre système, plusieurs points seront abordés, qui diffèrent de la configuration classique. La principale est le contrôle de l'orientation angulaire. Un autre point qui diffère de la section précédente est la présence des effets de Coriolis, les couples gyroscopiques en l'occurrence. Enfin, on présentera la différence entre les configurations décrites précédemment, notamment sur le résultat de la pseudo-inverse.

La section suivante présente comment stabiliser un tel système mais aussi comment obtenir certaines mesures qui ne sont pas directement disponibles par les capteurs, par observation d'états.

#### 4.2.1 Stabilisation du système

On rappelle les équations, avec les accélérations linéaires exprimées dans le repère inertiel et les accélérations angulaires exprimées dans le repère du drone.

$$\begin{bmatrix} m\ddot{x} \\ m\ddot{y} \\ m\ddot{z} \\ J_{xx}\dot{p} \\ J_{yy}\dot{q} \\ J_{zz}\dot{r} \end{bmatrix} = \begin{bmatrix} {}^I\mathcal{R}_B & 0 \\ 0 & I^{[3 \times 3]} \end{bmatrix} \begin{bmatrix} u_1 \\ u_2 \\ u_3 \\ u_4 \\ u_5 \\ u_6 \end{bmatrix} - \begin{bmatrix} 0 \\ 0 \\ mg \\ 0 \\ 0 \\ 0 \end{bmatrix} + \begin{bmatrix} 0 \\ \mathbf{C}_{g-all} \end{bmatrix} \quad (4.33)$$

##### 4.2.1.1 Accélérations linéaires

Pour les accélérations en translation, on souhaite obtenir la dynamique en boucle fermée suivante afin de stabiliser le système.

$$\ddot{\tilde{\xi}} + r(\tilde{\xi}, \dot{\tilde{\xi}}) = 0 \quad (4.34)$$

avec  $r(\tilde{\xi}, \dot{\tilde{\xi}})$  une fonction qui rend le système stable, qui sera défini plus tard. On note  $\tilde{\xi}$  l'erreur d'état du système définie de la manière suivante :

$$\tilde{\xi} = \xi - \xi_{ref} \quad (4.35)$$

avec  $\xi = [xyz]^T$  le vecteur des mesures d'états pour les positions en 3D. Celui-ci est obtenu soit par un système de Motion Capture (détaillé dans la section suivante) soit par un GPS.  $\xi_{ref}$  est la référence des états. Si l'on exprime la deuxième dérivée de (4.35) dans (4.34), on obtient :

$$\ddot{\xi} = \ddot{\xi}_{ref} - r(\tilde{\xi}, \dot{\tilde{\xi}}) \quad (4.36)$$

#### 4.2.1.2 Accélérations angulaires

Pour les accélérations angulaires, on souhaite obtenir la dynamique en boucle fermée suivante afin de stabiliser le système.

$$\dot{\tilde{\Omega}} + r(\tilde{\mathbf{R}}, \tilde{\Omega}) = 0 \quad (4.37)$$

avec  $r(\tilde{\mathbf{R}}, \tilde{\Omega})$  une fonction qui rend le système stable, qui sera défini plus tard. On note  $\tilde{\Omega}$  l'erreur d'état du système défini de la manière suivante :

$$\tilde{\Omega} = \Omega - \Omega_{ref} \quad (4.38)$$

avec  $\Omega_{ref}$  la référence des vitesses angulaires. On note  $\tilde{\mathbf{R}}$  l'erreur d'orientation. Si l'on exprime la première dérivée de (4.38) dans (4.37), on obtient :

$$\dot{\tilde{\Omega}} = \dot{\Omega}_{ref} - r(\tilde{\mathbf{R}}, \tilde{\Omega}) \quad (4.39)$$

Remarque : De même que précédemment, les termes  $\ddot{\xi}_{ref}$ ,  $\dot{\xi}_{ref}$ ,  $\dot{\Omega}_{ref}$  et  $\Omega_{ref}$  sont nuls dans notre cas. Pour stabiliser le système on propose finalement d'obtenir un système de la forme suivante :

$$\begin{bmatrix} \ddot{\xi} \\ \dot{\tilde{\Omega}} \end{bmatrix} - \begin{bmatrix} r(\tilde{\xi}, \dot{\xi}) \\ r(\tilde{\mathbf{R}}, \tilde{\Omega}) \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \end{bmatrix} \quad (4.40)$$

Dans la section suivante on explicite le calcul de l'erreur d'orientation. Celle-ci n'est pas triviale, et il existe différentes méthodes pour la calculer qui seront explicitées.

#### 4.2.1.3 Calcul de l'erreur d'orientation

On choisit de calculer l'erreur d'orientation basée sur les matrices de rotation. C'est un choix qui a été fait, il est tout à fait possible d'utiliser des quaternions pour déterminer cette erreur, ou encore les angles d'Euler. [80] explique l'emploi des quaternions pour calculer l'erreur de rotation du système. L'utilisation des angles d'Euler est cependant à éviter, pour ne pas avoir de problèmes liés à des discontinuités des angles.

Pour expliquer rapidement ce phénomène, la méthode des angles d'Euler consiste à effectuer trois rotations successives permettant d'orienter un objet dans l'espace. Deux des angles sont définis entre  $-180^\circ$  et  $180^\circ$  et le troisième entre  $-90^\circ$  et  $90^\circ$  et non sur  $360^\circ$ . Lorsque les angles limites sont atteints, donc pour des rotations de  $90^\circ$  par exemple, de fortes discontinuités apparaissent. Ce phénomène n'est pas gênant pour de faibles angles, car on ne rentre pas dans le domaine de discontinuité. Cependant, dans notre configuration de vol, on sera confronté à des angles de vols très importants par rapport à un drone classique ce qui exclut l'utilisation de cette méthode. On choisit plutôt de calculer l'erreur de rotation comme dans [81] en calculant une erreur de rotation entre deux matrices. Cette méthode permet d'exclure les problèmes décrits précédemment. La première matrice est calculée à partir des mesures alors que la deuxième est créée à partir des angles désirés. L'erreur d'orientation  $\tilde{\mathbf{R}}$  du système est définie de la manière suivante :

$$\tilde{\mathbf{R}} = \left( \frac{\mathcal{R}_d^T \mathcal{R}_m - \mathcal{R}_m^T \mathcal{R}_d}{2} \right)^V \quad (4.41)$$

avec  $\mathcal{R}_m$  la matrice d'orientation mesurée et  $\mathcal{R}_d$  la matrice d'orientation désirée. La notation  $[\ ]^V$  est celle définie avec l'opérateur  $\text{vex}^\times$  de l'équation (3.24).

### 4.2.2 Mesure et observation des états

Dans la sous-section précédente, on présente la manière de stabiliser le système. On remarque que pour fonctionner, le système nécessite une référence ainsi qu'une mesure des états. Dans notre système, la mesure de la vitesse angulaire est directement donnée par le gyromètre de la centrale inertielle.

De même, le système de motion capture OptiTrack plus amplement détaillé dans le chapitre suivant permet de mesurer la position du robot en temps réel. Cependant il reste les mesures de vitesses linéaires et de position angulaire à observer pour pouvoir commander le système.

#### 4.2.2.1 Vitesse linéaire

Pour rappel, la vitesse est la dérivée de la position. Intuitivement, on peut donner l'exemple du conducteur cherchant à évaluer la vitesse d'un autre automobiliste. Si il regarde au moins deux fois la position d'une voiture, à deux temps différents, il pourra en évaluer sa vitesse. La vitesse est donc bien une variation de position par rapport à une variation de temps. Pour le drone ce sera le même principe. Si on cherche à mesurer la vitesse linéaire du drone, une première approche consiste à utiliser la dérivée d'Euler depuis la mesure de position (obtenue dans notre cas par motion capture ou GPS). On en rappelle l'expression sous forme d'états à partir de l'équation (4.5) :

$$\dot{\mathbf{X}} = \begin{bmatrix} 0 & 1 \\ 0 & 0 \end{bmatrix} \mathbf{X} + \begin{bmatrix} 0 \\ 0 \end{bmatrix} \mathbf{U} + \alpha(t) \quad (4.42)$$

$$\mathbf{Y} = \begin{bmatrix} 1 & 0 \end{bmatrix} \mathbf{X} + \beta(t) \quad (4.43)$$

avec  $\dot{\mathbf{X}}$  le vecteur des vitesses du système et  $\mathbf{X}$  celui des positions.  $\alpha(t), \beta(t)$  sont des fonctions représentant le bruit de la mesure en temps continu. On définit les matrices  $\mathcal{A}$ ,  $\mathcal{B}$  et  $\mathcal{C}$  à partir de l'équation précédente comme suit :

$$\mathcal{A} = \begin{bmatrix} 0 & 1 \\ 0 & 0 \end{bmatrix} \quad \mathcal{B} = \begin{bmatrix} 0 \\ 0 \end{bmatrix} \quad \mathcal{C} = \begin{bmatrix} 1 & 0 \end{bmatrix} \quad (4.44)$$

Très rapidement on se rend compte que cette approche atteint des limites car la mesure de la position est très bruitée ce qui induit des comportements de sursauts lors des vols. Par la suite, on choisit d'utiliser un filtre de Kalman. Celui-ci se base sur la représentation d'état de la mesure de vitesse. On exprime la forme d'états sous forme discrète :

$$\begin{cases} \dot{\mathbf{X}}_{k+1} = \mathcal{F} \mathbf{X}_k + \mathcal{G} \mathbf{U}_k + \alpha_k \\ \mathbf{Y} = \mathcal{C} \mathbf{X}_k + \beta_k \end{cases} \quad (4.45)$$

Avec  $\mathcal{F}$  et  $\mathcal{G}$  définis de la manière suivante :

$$\mathcal{F} = \exp(\mathcal{A} T_e) \quad (4.46)$$

$$\mathcal{G} = \mathcal{A}^{-1} (\exp(\mathcal{A} T_e) - \mathcal{I}) \mathcal{B} \quad (4.47)$$

$T_e$  étant la période d'échantillonnage.  $\alpha_k, \beta_k$  sont des fonctions représentant le bruit de la mesure, cette fois-ci en temps discret. On calcule le vecteur d'estimation des états à l'instant  $k$ , noté  $\hat{\mathbf{X}}_k$ . On applique le même procédé pour la matrice des erreurs de covariance des états notée  $P$ .

$$\begin{aligned} \hat{\mathbf{X}}_k &= \hat{\mathbf{X}}_{k-1} + \mathcal{K}(\mathbf{Y}_k - \mathcal{C} \hat{\mathbf{X}}_{k-1}) \\ P_k &= (\mathcal{I} - \mathcal{K} \mathcal{C}) P_{k-1} (\mathcal{I} - \mathcal{K} \mathcal{C})^T + \mathcal{K} \mathbb{R} \mathcal{K}^T \end{aligned} \quad (4.48)$$

$\mathcal{K}$  désigne le gain du filtre de Kalman.  $\mathbb{R}$  et  $\mathbb{Q}$  sont des matrices que l'on pourra ajuster, avec  $\mathbb{R}$  qui désigne la matrice de covariance des capteurs et  $\mathbb{Q}$  la matrice de covariance du modèle. Une grande valeur dans ces matrices représente une covariance élevée et donc une moins grande confiance. Remarque : le terme  $(\mathbf{Y}_k - \mathcal{C}\mathbf{X}_{k-1})$  s'appelle l'innovation du filtre. L'étape suivante du filtre consiste à calculer la prédiction de l'estimation des états et de l'erreur de covariance ainsi que de mettre à jour le gain  $\mathcal{K}$ . Ce calcul se fera notamment à l'aide du modèle d'évolution du système  $C$ .

$$\begin{aligned}\hat{\mathbf{X}}_{k+1} &= \mathbf{A}\hat{\mathbf{X}}_k + \mathbf{B}\mathbf{U}_k \\ \mathcal{P}_{k+1} &= \mathbf{A}\mathcal{P}_k + \mathbb{Q} \\ \mathcal{K} &= \mathcal{P}_{k-1}\mathbf{C}^T (\mathbf{C}\mathcal{P}_{k-1}\mathbf{C}^T + \mathbb{R})^{-1}\end{aligned}\quad (4.49)$$

Ces valeurs calculées seront alors utilisées pour le calcul suivant de l'équation (4.48). Donc on aura

$$\begin{aligned}\hat{\mathbf{X}}_k &= \hat{\mathbf{X}}_{k+1} \\ \mathcal{P}_k &= \mathcal{P}_{k+1}\end{aligned}\quad (4.50)$$

On teste les deux filtres en pratique en ajoutant cependant un filtre du deuxième ordre sur le calcul de la dérivée d'Euler. Pour celui-ci on choisira une fréquence de coupure  $\omega = 2\pi f$  à  $f = 15$  Hz avec une période d'échantillonnage de 0.01s et un amortissement à  $\zeta = 0.9$ . Le résultat est présenté en Figure 4.2. Malgré un réglage du filtre deuxième

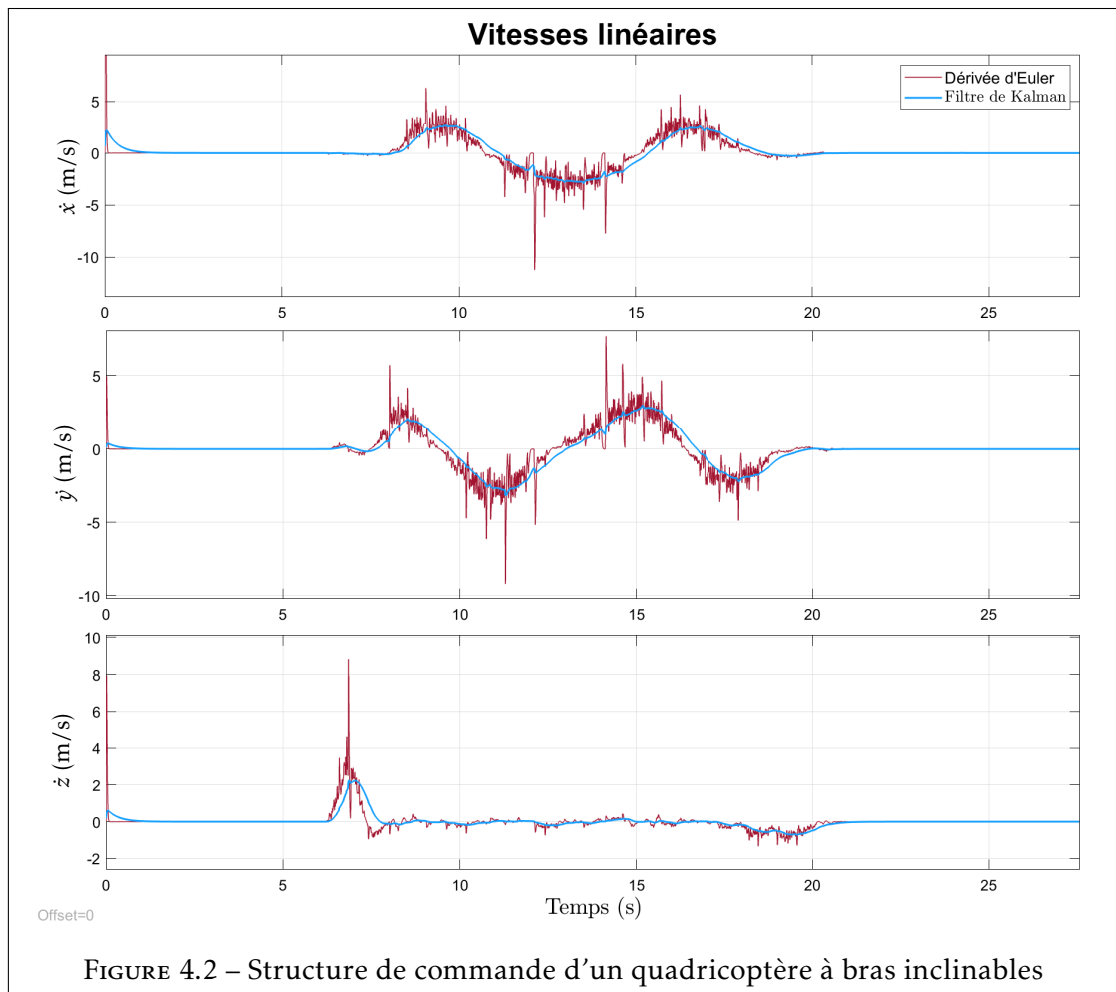


FIGURE 4.2 – Structure de commande d'un quadricoptère à bras inclinables

ordre qui ne doit pas être optimal, on remarque que le résultat avec le filtre de Kalman est nettement moins bruité que celui obtenu avec la dérivée d'Euler. C'est donc ce filtre que l'on gardera pour la suite.

Avec les équations 4.46 et 4.47, on utilisera les matrices suivantes pour le filtre (appliqué à un seul axe à la fois) :

$$\mathbf{X} = \begin{bmatrix} x \\ \dot{x} \end{bmatrix} \quad \mathcal{F} = \begin{bmatrix} 1 & T_e \\ 0 & 1 \end{bmatrix} \quad \mathcal{G} = \begin{bmatrix} 0 \\ 0 \end{bmatrix} \quad \mathcal{C} = \begin{bmatrix} 1 & 0 \end{bmatrix} \quad \mathcal{D} = 0 \quad (4.51)$$

On utilisera les gains suivants pour les matrices  $\mathbb{Q}$  et  $\mathbb{R}$

$$\mathbb{Q} = \begin{bmatrix} 0.01 & 0 \\ 0 & 0.1 \end{bmatrix} \quad \mathbb{R} = \begin{bmatrix} 0.1 \end{bmatrix} \quad (4.52)$$

Ce qui veut dire que l'on fait très confiance au modèle mais beaucoup moins à la mesure. Remarque : Dans cette section on présente deux manières d'estimer la vitesse linéaire du drone, mais il en existe d'autres. Une manière très utilisée est la mesure de vitesse par flux optique. Le principe consiste à analyser un flux d'images et d'estimer les déplacements d'une image à l'autre. Si un point se déplace d'une distance  $x$  entre deux photos prises à un intervalle de temps  $dt$ , on pourra en déduire la vitesse du véhicule.

En Annexe D.2 on présente une manière d'utiliser un filtre de Kalman pour estimer l'altitude du drone à l'aide de l'accéléromètre et du baromètre présents parmi les capteurs disponibles.

#### 4.2.2.2 Orientation

L'orientation du système n'est pas directement mesurée par le système. A la place on utilise les capteurs présents sur les centrales inertielles (plus amplement détaillés dans la partie 5.2). Les capteurs présents mesurent le champ magnétique, la vitesse angulaire et enfin l'accélération du système. Un filtre complémentaire est utilisé pour observer l'orientation du système aussi appelé *Attitude and Heading Reference System* (AHRS). Cette fusion de données est basée sur les travaux de Mahony [82]. Le principe basique de l'observateur se décompose en deux parties, l'accéléromètre (le cas échéant avec le magnétomètre) et le gyromètre. On note l'estimation de cette mesure  $\tilde{\eta}_m$

Premièrement, si le système est posé à plat et soumis à la force que celle de la pesanteur, l'accéléromètre mesure une force opposée à la gravité, donc orientée vers le haut, que l'on nommera  $\eta_{acc}$ . On peut alors définir un repère sans cap (donc non défini par rapport au nord magnétique). Si l'on rajoute la mesure du magnétomètre on obtient un cap permettant une mesure absolue de position angulaire. Cependant, la moindre accélération sur le système perturbe instantanément l'observation d'orientation. De même, l'ajout d'un champ magnétique corrompt les données du magnétomètre.

Deuxièmement, le gyromètre mesure la vitesse angulaire du système, il est donc possible d'intégrer cette mesure par rapport au temps, ce qui permet d'obtenir une mesure d'orientation relative, peu bruitée, que l'on nommera  $\tilde{\eta}_{gyro}$ . Les mesures sont cependant souvent biaisées et l'intégration de ce biais fait dériver la mesure au cours du temps. Le filtre complémentaire basique permet donc de mélanger les deux solutions précédentes en limitant à la fois les problèmes d'initialisation, de bruit, et de biais. Les deux estimations sont directement sommées avec un gain sur chacune comme suit (noté respectivement  $k_{acc}$  et  $k_{gyro}$ ) :

$$\tilde{\eta}_m = k_{gyro} \tilde{\eta}_{gyro} + k_{acc} \tilde{\eta}_{acc} \quad (4.53)$$





On répète le même procédé pour la configuration en  $\times$  on obtient :

$$\mathcal{O}_{\times}^{\dagger} = \begin{bmatrix} p & 0 & p & 0 & -p & 0 & -p & 0 \\ -p & 0 & p & 0 & p & 0 & -p & 0 \\ 0 & 1 & 0 & 1 & 0 & 1 & 0 & 1 \\ -p\delta & pL & p\delta & pL & p\delta & -pL & -p\delta & -pL \\ p\delta & -pL & p\delta & pL & -p\delta & pL & -p\delta & -pL \\ -L & -\delta & -L & \delta & -L & -\delta & -L & \delta \end{bmatrix}^{\dagger} = \begin{bmatrix} \frac{1}{4p} & -\frac{1}{4p} & 0 & 0 & 0 & -\frac{L^2}{4L(L^2+\delta^2)} \\ \frac{\delta}{4Lp} & -\frac{\delta}{4Lp} & \frac{1}{4} & \frac{1}{4Lp} & -\frac{1}{4Lp} & -\frac{\delta^2}{4\delta(L^2+\delta^2)} \\ \frac{1}{4p} & \frac{1}{4p} & 0 & 0 & 0 & -\frac{L^2}{4L(L^2+\delta^2)} \\ -\frac{\delta}{4Lp} & -\frac{\delta}{4Lp} & \frac{1}{4} & \frac{1}{4Lp} & \frac{1}{4Lp} & \frac{\delta^2}{4\delta(L^2+\delta^2)} \\ -\frac{1}{4p} & \frac{1}{4p} & 0 & 0 & 0 & -\frac{L^2}{4L(L^2+\delta^2)} \\ -\frac{\delta}{4Lp} & \frac{\delta}{4Lp} & \frac{1}{4} & -\frac{1}{4Lp} & \frac{1}{4Lp} & -\frac{\delta^2}{4\delta(L^2+\delta^2)} \\ -\frac{1}{4p} & -\frac{1}{4p} & 0 & 0 & 0 & -\frac{L^2}{4L(L^2+\delta^2)} \\ \frac{\delta}{4Lp} & \frac{\delta}{4Lp} & \frac{1}{4} & -\frac{1}{4Lp} & -\frac{1}{4Lp} & \frac{\delta^2}{4\delta(L^2+\delta^2)} \end{bmatrix} \quad (4.56)$$

Enfin, pour la configuration en  $\checkmark$  on obtient :

$$\mathcal{O}_{\checkmark}^{\dagger} = \begin{bmatrix} s & 0 & s & 0 & -s & 0 & -s & 0 \\ -c & 0 & c & 0 & c & 0 & -c & 0 \\ 0 & 1 & 0 & 1 & 0 & 1 & 0 & 1 \\ -s\delta & d_1*s - d_2*c & s\delta & d_1*s + d_2*c & s\delta & -d_1*s + d_2*c & -s\delta & -d_1*s - d_2*c \\ c\delta & -d_1*c - d_2*s & c\delta & d_1*c - d_2*s & -c\delta & d_1*c + d_2*s & -c\delta & -d_1*c + d_2*s \\ -d_1 & -\delta & -d_1 & \delta & -d_1 & -\delta & -d_1 & \delta \end{bmatrix}^{\dagger} = \begin{bmatrix} \frac{\sqrt{2}}{4} & -\frac{\sqrt{2}}{4} & 0 & 0 & 0 & -\frac{d_1^2}{4d_1(d_1^2+\delta^2)} \\ \frac{\sqrt{2}\delta(d_1+d_2)}{4(d_1^2+d_2^2)} & -\frac{\sqrt{2}\delta(d_1-d_2)}{4(d_1^2+d_2^2)} & \frac{1}{4} & \frac{\sqrt{2}(d_1-d_2)}{4(d_1^2+d_2^2)} & -\frac{\sqrt{2}(d_1+d_2)}{4(d_1^2+d_2^2)} & -\frac{\delta^2}{4\delta(d_1^2+\delta^2)} \\ \frac{\sqrt{2}}{4} & \frac{\sqrt{2}}{4} & 0 & 0 & 0 & -\frac{d_1^2}{4d_1(d_1^2+\delta^2)} \\ -\frac{\sqrt{2}\delta(d_1-d_2)}{4(d_1^2+d_2^2)} & -\frac{\sqrt{2}\delta(d_1+d_2)}{4(d_1^2+d_2^2)} & \frac{1}{4} & \frac{\sqrt{2}(d_1+d_2)}{4(d_1^2+d_2^2)} & \frac{\sqrt{2}(d_1-d_2)}{4(d_1^2+d_2^2)} & \frac{\delta^2}{4\delta(d_1^2+\delta^2)} \\ -\frac{\sqrt{2}}{4} & \frac{\sqrt{2}}{4} & 0 & 0 & 0 & -\frac{d_1^2}{4d_1(d_1^2+\delta^2)} \\ -\frac{\sqrt{2}\delta(d_1+d_2)}{4(d_1^2+d_2^2)} & \frac{\sqrt{2}\delta(d_1-d_2)}{4(d_1^2+d_2^2)} & \frac{1}{4} & -\frac{\sqrt{2}(d_1-d_2)}{4(d_1^2+d_2^2)} & \frac{\sqrt{2}(d_1+d_2)}{4(d_1^2+d_2^2)} & -\frac{\delta^2}{4\delta(d_1^2+\delta^2)} \\ -\frac{\sqrt{2}}{4} & -\frac{\sqrt{2}}{4} & 0 & 0 & 0 & -\frac{d_1^2}{4d_1(d_1^2+\delta^2)} \\ \frac{\sqrt{2}\delta(d_1-d_2)}{4(d_1^2+d_2^2)} & \frac{\sqrt{2}\delta(d_1+d_2)}{4(d_1^2+d_2^2)} & \frac{1}{4} & -\frac{\sqrt{2}(d_1+d_2)}{4(d_1^2+d_2^2)} & -\frac{\sqrt{2}(d_1-d_2)}{4(d_1^2+d_2^2)} & \frac{\delta^2}{4\delta(d_1^2+\delta^2)} \end{bmatrix} \quad (4.57)$$

On généralise l'équation (3.104) à toutes les configurations, de la manière suivante :

$$\begin{bmatrix} T_1 \sin(\alpha_1) \\ T_1 \cos(\alpha_1) \\ T_2 \sin(\alpha_2) \\ T_2 \cos(\alpha_2) \\ T_3 \sin(\alpha_3) \\ T_3 \cos(\alpha_3) \\ T_4 \sin(\alpha_4) \\ T_4 \cos(\alpha_4) \end{bmatrix} = \mathcal{O}_*^{\dagger} \begin{bmatrix} u_1 \\ u_2 \\ u_3 \\ u_4 \\ u_5 \\ u_6 \end{bmatrix} \quad (4.58)$$

Avec  $*$  à remplacer en fonction de la configuration choisie. On obtient ensuite les valeurs à affecter aux actionneurs par les deux équations suivantes :

$$T_i = \sqrt{(T_i \sin(\alpha_i))^2 + (T_i \cos(\alpha_i))^2} \quad \alpha_i = \text{atan2}\left(\frac{T_i \sin(\alpha_i)}{T_i \cos(\alpha_i)}\right) \quad (4.59)$$

On définit le vecteur des entrées de commande des actionneurs :

$$\Upsilon = [T_1, T_2, T_3, T_4, \alpha_1, \alpha_2, \alpha_3, \alpha_4]^T \quad (4.60)$$

Remarque : Les valeurs de  $T_i$  sont toujours positives mais peuvent être nulles. Dans ce cas l'angle du bras est indéterminé ce qui correspond à une singularité du système traité plus loin. En ces points, le bras aura tendance à tourner dans toutes les directions sans jamais servir au système car les hélices ne tourneront pas. Ces points de fonctionnement sont donc à éviter, car le drone n'utilise pas tous les actionneurs ce qui n'est pas optimisé mais ne sera pas non plus un danger de par l'absence de rotation des moteurs.

#### 4.2.4 Contrôleur linéaire

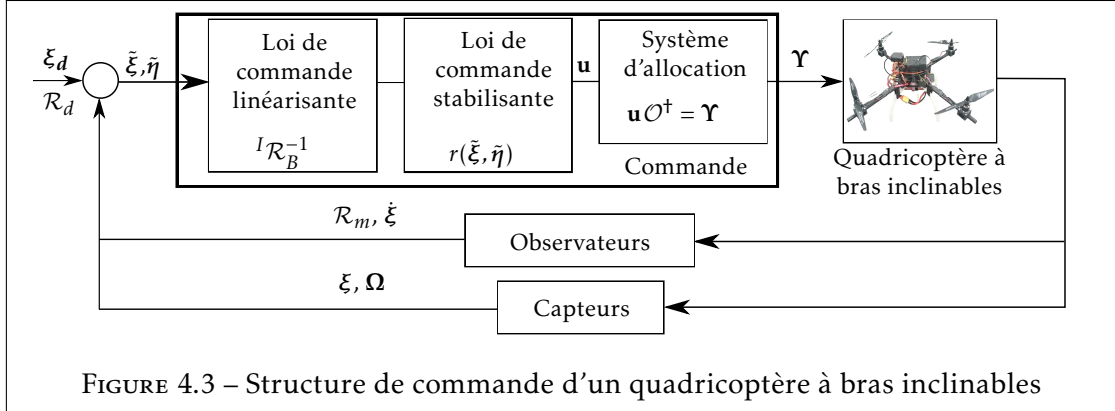


FIGURE 4.3 – Structure de commande d'un quadricoptère à bras inclinables

On résume le système graphiquement en Figure 4.3. La suite logique consiste à généraliser la commande linéaire proposée en Section 4.1.2 au système à 6 DDL. En utilisant un contrôleur linéaire PD, les fonctions  $r(\xi, \dot{\xi}) = r_{\xi_{PD}}$  et  $r(\mathbf{R}, \dot{\mathbf{R}}) = r_{\Omega_{PD}}$  prennent alors les formes suivantes :

$$\begin{bmatrix} r_{\xi_{PD}} \\ r_{\Omega_{PD}} \end{bmatrix} = \begin{bmatrix} -k_1 \dot{\xi} - k_2 \xi \\ -k_3 \dot{\mathbf{R}} - k_4 \mathbf{R} \end{bmatrix} = \begin{bmatrix} -k_{11} \dot{x} - k_{21} \bar{x} \\ -k_{12} \dot{y} - k_{22} \bar{y} \\ -k_{13} \dot{z} - k_{23} \bar{z} \\ -k_{31} \bar{p} \\ -k_{32} \bar{q} \\ -k_{33} \bar{r} \end{bmatrix} - k_4 \begin{bmatrix} 0 \\ 0 \\ 0 \\ \left[ \frac{\mathcal{R}_d^T \mathcal{R}_m - \mathcal{R}_m^T \mathcal{R}_d}{2} \right]^{V[3 \times 3]} \end{bmatrix} \quad (4.61)$$

Les gains  $k_{ij}$  de l'algorithme PD sont choisis de manière à garantir la stabilité du système. On reprendra les principes énoncés en Section 4.1.2 pour avoir un premier ordre de grandeur. Ceux-ci seront ajustés par la suite en pratique lors des tests.

On propose de considérer un autre algorithme de commande robuste aux perturbations permettant de répondre efficacement aux contraintes du système.

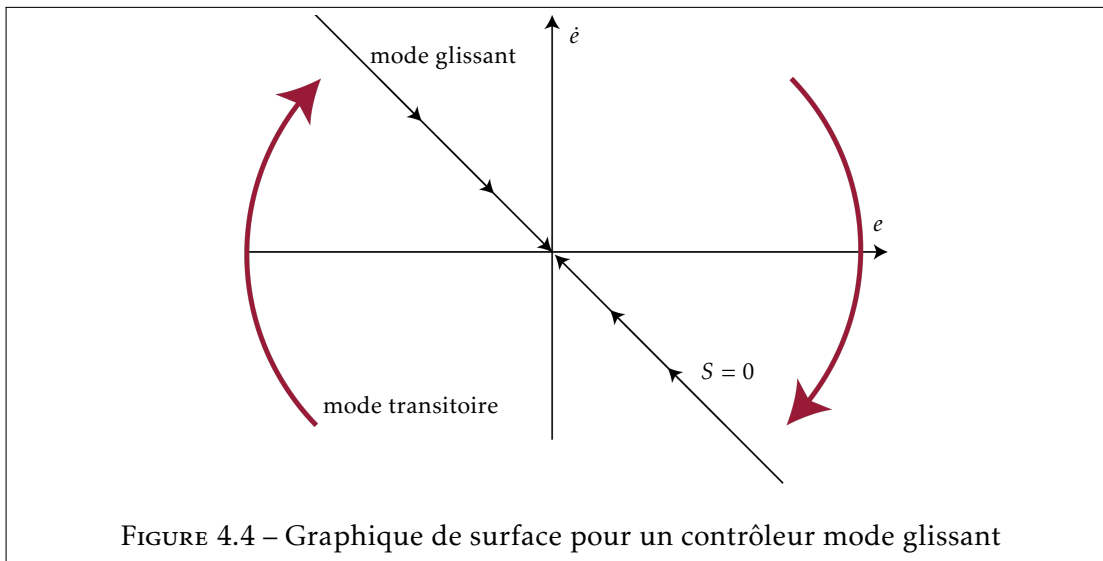
#### 4.2.5 Contrôleur Robust SMCST

La méthodologie des lois de commandes par mode glissant offrent plus de robustesse au système. Étant soumis à des perturbations extérieures importantes, il devient très intéressant d'implémenter ce genre d'algorithme dans notre système. Parmi les perturbations, on citera le vent mais également les différents moments des effets de Coriolis qui n'ont pas été prises en compte dans la loi de commande.

La commande par mode glissant simple (*Sliding Mode Control (SMC)*) est très largement connue dans la littérature et offre une bonne robustesse au système. [84] propose de l'appliquer directement sur un système de quadricoptère classique le rendant plus robuste vis à vis des perturbations extérieures. [85] propose une loi de commande similaire avec une adaptation de la fonction sign pour limiter le phénomène de *chattering*.

Pour les drones à bras inclinables, on pourra citer [86], [87] qui utilisent également un contrôle par mode glissant. Les deux travaux proposent une preuve de stabilité. De plus, [86] propose de considérer un mode glissant de rang supérieur pour réduire le phénomène de *chattering* (fait que la commande passe au dessus puis en-dessous de la valeur désirée à haute fréquence). [88] propose d'aller plus loin en appliquant cet algorithme pour pallier une faute moteur en reconfigurant le drone d'une forme X à une forme T. Cette capacité due au sur-actionnement du système peut donc s'avérer très utile en cas de pannes par exemple.

Avant d'introduire le fonctionnement d'un mode glissant, il faut expliciter ce qu'est une surface. Le concept de surface désigne le fait que le système va suivre un chemin qui va le faire converger vers sa référence dans l'espace de phase. En Figure 4.4 on montre le chemin que les états du système vont prendre pour rejoindre la surface et enfin avoir une erreur nulle. On précise que la surface est composée des erreurs (et dérivées) calculées à partir des états et références du système et notés  $e, \dot{e}$  sur la figure.



Pour notre système présenté en équation (4.33), l'entrée de commande d'un SMC classique s'écrit :

$$\mathbf{U} = \gamma \text{sign}(\mathbf{S}) \quad (4.62)$$

Avec  $\gamma$  le gains du système, et  $\mathbf{S}$  la surface que le système doit rejoindre, définit comme suit :

$$\mathbf{S} = k_p \tilde{\xi} \quad (4.63)$$

$\tilde{\xi}$  étant l'erreur de position du système. Cette équation correspond à un contrôle du premier ordre. On définit la fonction sign en Annexe C.1.2, avec  $\gamma$  un gain et  $\mathbf{S}$  la surface que le système doit rejoindre pour garantir la stabilité du système.

Pour comparer les performances du système obtenues avec la loi de commande linéaire présenté en équation (4.61), on propose d'utiliser une loi de commande robuste de type SMCST. Celle-ci présente beaucoup de similarités avec la loi de commande mode glissant classique précédente, en proposant cependant quelques améliorations. Couramment utilisée en commande, on peut retrouver son utilisation dans de nombreux domaines (exemple de la commande d'une machine outil [89]). [90] compare les performances d'un quadricoptère classique qui utilise un mode glissant super twisting avec les performances obtenues utilisant une commande linéaire LQR. [91], [92] proposent d'adapter le SMCST classique pour une version qui s'adapte en fonction de la perturbation rencontrée.

On rappelle en effet que volant le plus souvent en extérieur, une violente rafale de vent d'amplitude inconnue peut percuter le drone. Une commande de type robuste peut donc s'avérer très intéressante pour notre application, car elle permettra de compenser la perturbation rencontrée.

On propose le contrôleur SMCST pour notre système avec l'expression suivante :

$$\begin{bmatrix} r_{\xi_{SMCST}} \\ r_{\Omega_{SMCST}} \end{bmatrix} = \begin{bmatrix} \gamma_{1_\xi} |\mathbf{S}_\xi|^l \text{sign}(\mathbf{S}_\xi) + \gamma_{3_\xi} \int \text{sign}(\mathbf{S}_\xi) \\ \gamma_{1_\Omega} |\mathbf{S}_\Omega|^l \text{sign}(\mathbf{S}_\Omega) + \gamma_{3_\Omega} \int \text{sign}(\mathbf{S}_\Omega) \end{bmatrix} \quad (4.64)$$

avec :

$$\mathbf{S}_\xi = \ddot{\xi} + \gamma_{2_\xi} \dot{\xi} \quad (4.65)$$

$$\mathbf{S}_\Omega = \ddot{\Omega} + \gamma_{2_\Omega} \dot{\Omega} \quad (4.66)$$

On utilisera une adaptation particulière de la fonction  $\text{sign}$  qui permet de diminuer les problèmes de *chattering* quand l'erreur devient très faible, et donc que la surface est presque atteinte. Soit  $\epsilon$  une constante très petite, la fonction  $\text{sign}$  est définie tel que :

$$\text{sign}(\mathbf{S}) = \frac{\mathbf{S}}{|\mathbf{S}| + \epsilon} \quad (4.67)$$

Remarque : Dans l'équation (C.33), si  $\iota$  est égal à 1, le système se résume à un simple PID. De même, si  $\iota$  est nul, le système est identique à un mode glissant classique. On s'inspirera de [93], [94] pour faire la preuve de stabilité décrite en Annexe C.2.3.

Remarque 2 : le SMCST se différencie du SMC classique par le fait que c'est une loi de commande de rang plus élevé. En effet, on remarque la partie intégrale présente dans l'équation (C.33).

Remarque 3 : La commande proposée ne contient que des composantes continues contrairement à un mode glissant classique. En effet, la fonction  $\text{sign}$  est présente dans tout les cas mais pour le mode glissant super twisting, elle est multipliée par la surface qui est continue. Il en va de même pour le terme intégral qui par définition est continu. L'intérêt sur la commande est d'obtenir une régulation plus douce du système.

Ces différentes sections ont proposé une manière de contrôler le système avec différents algorithmes répondants aux contraintes que l'on s'est imposées.

La configuration à bras inclinables possède cependant une limite physique. Lorsque le drone se tourne dans certaines orientations, certains actionneurs deviennent inutiles. La section suivante explicite ce phénomène et propose une solution qui peut être mise en pratique.

### 4.3 Singularités

Le but de cette section est d'exposer les points de singularités du système pour les différentes configurations. On désigne ici par singularité une position dans laquelle le véhicule sera soumis à une discontinuité de manœuvrabilité due à une contrainte mécanique sur les actionneurs. En d'autres termes et dans notre cas, c'est une configuration où la force  $T_i$  du bras  $i$  sera tantôt positive, puis nulle puis négative. Cela aura pour conséquence de retourner au moins un de ses bras, rendant le véhicule potentiellement moins stable pendant le temps où le drone est dans cette position.

Une approche pratique permet de mettre en avant que ce phénomène se produit lorsque la force sur un bras est nulle, induisant par ailleurs une indétermination sur l'angle des bras. Ce genre de cas se trouve par exemple lorsque le drone est vertical en configuration "+". Le premier bras ne peut pas appliquer de force et on se rend vite compte qu'avec de petites variations d'orientation (vibrations par exemple), on peut avoir des variations d'angles très importantes dans le bras. Beaucoup de configurations de drones (y compris à angles inclinables) possèdent des singularités de par leur mécanique.

On souhaite mettre en avant graphiquement ces singularités. On se limitera au premier bras pour plus de visibilité. Pour ce faire on représente les champs de vecteurs de poussées du premier bras en fonction de l'orientation du nez de l'appareil. On présente en Figure 4.5 la position simulée du drone. On présente sur une sphère de rayon 1m les différentes zones de vol. En fonction de son orientation, on entre dans une singularité (zone rouge) ou dans une zone de travail (en jaune). Une vidéo associée à ce mouvement est donné en <https://youtu.be/70aKX6Z1Hpg>.

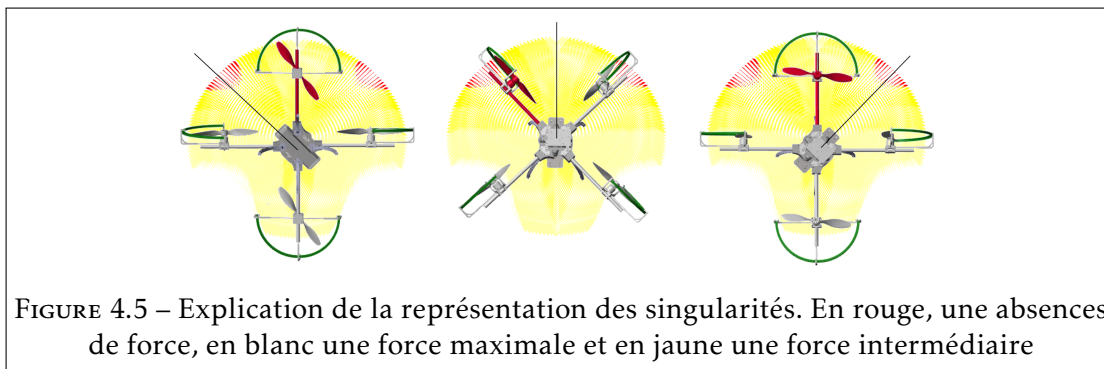


FIGURE 4.5 – Explication de la représentation des singularités. En rouge, une absence de force, en blanc une force maximale et en jaune une force intermédiaire

On détaille maintenant chacune des configurations 3D. On rappelle que dans la configuration en "+", le premier bras est confondu avec l'axe  $\vec{x}_B$ .

Le champ de vecteur du premier bras est donné en Figure 4.6. Sur cette figure le vecteur de poussée du premier bras est placé à l'endroit vers lequel le nez de l'appareil pointe. Autrement dit, sur la sphère de rayon 1, le champ de vecteurs au sommet correspond à un drone placé à la verticale. On précise que l'on considère le problème en quasi-statique. C'est à dire que les hélices ne tournent pas et on considère que le drone n'est soumis qu'à la seule force de pesanteur.

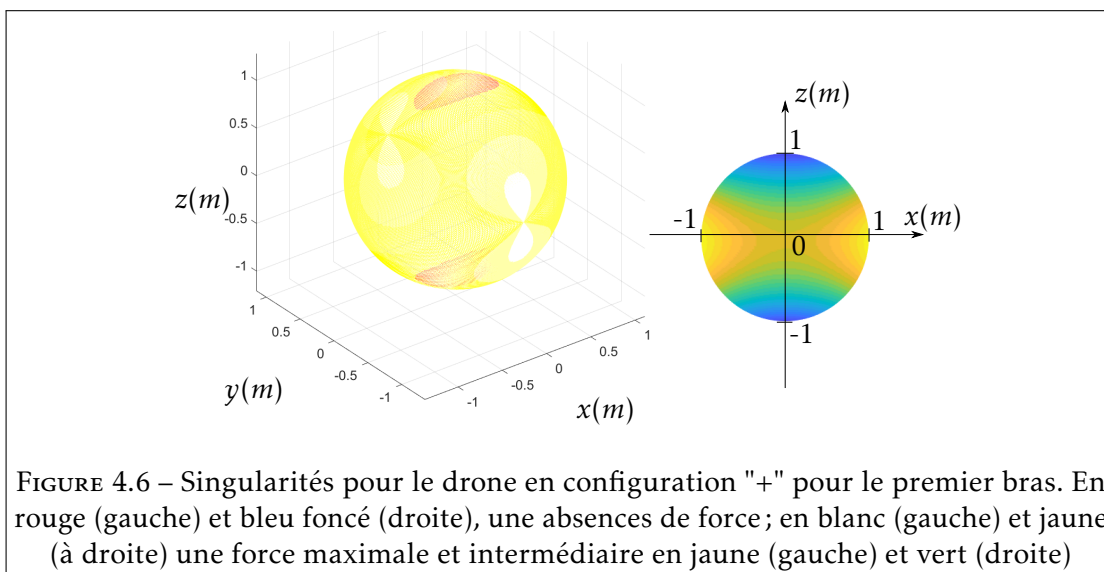


FIGURE 4.6 – Singularités pour le drone en configuration "+" pour le premier bras. En rouge (gauche) et bleu foncé (droite), une absence de force; en blanc (gauche) et jaune (à droite) une force maximale et intermédiaire en jaune (gauche) et vert (droite)

Sur la figure on distingue clairement les deux points de singularités quand le nez pointe vers le haut et le bas. La différence de couleur désigne en réalité la force de poussée imposée sur le bras (rouge est faible, jaune moyen et blanc important). Pour plus de clarté également, on ne simule pas toutes les configurations d'orientations possibles. Cela explique que l'on ne retrouve pas de tache blanche sur tous les côtés.

Remarque : en dynamique, les points de singularités sont changés. En effet, le fait de rajouter une consigne de position sur l'un des axes va globalement changer le comportement des bras. Si on prend l'exemple du cas particulier qui consiste à avoir une chute libre (force nulle sur  $\vec{z}_I$ ) et une force importante sur  $\vec{x}_I$  par exemple, on déplacera le point de singularité de  $\frac{\pi}{2}$  vers l'avant de l'appareil.

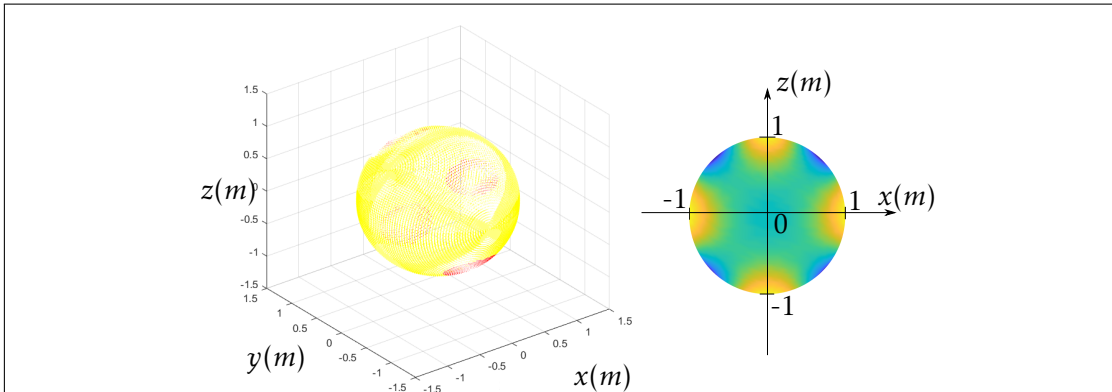


FIGURE 4.7 – Singularités pour le drone en configuration  $\times$  pour le premier bras. En rouge (gauche) et bleu foncé (droite), une absences de force ; en blanc (gauche) et jaune (à droite) une force maximale et intermédiaire en jaune (gauche) et vert (droite)

Pour la configuration en  $\times$ , on présente le champ de potentiel du premier bras en Figure 4.7. De par sa géométrie, on remarque tout de suite que les points de singularités ne sont plus seulement à la verticale mais qu'il faut également pivoter l'axe  $\vec{z}_B$  d'un angle de  $\frac{\pi}{4}$  pour atteindre cette singularité. En tournant de  $\frac{3\pi}{4}$  on obtient cependant un maximum de poussée. Idem si l'on fait tanguer l'appareil vers le bas et que l'on change son orientation sur l'axe  $\vec{z}_B$  de  $\frac{\pi}{4}$ . Enfin, si l'on retourne l'appareil d'un angle de  $\pi$  sur son axe  $\vec{z}_B$  puis que l'on refait les opérations précédentes, on retrouve les singularités de l'autre côté.

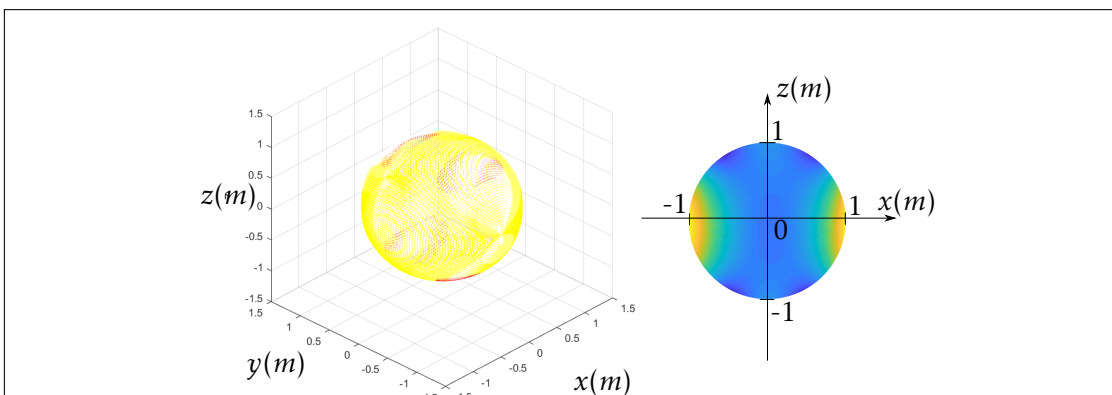


FIGURE 4.8 – Singularités pour la configuration  $\checkmark$  pour le premier bras. En rouge (gauche) et bleu foncé (droite), une absences de force ; en blanc (gauche) et jaune (à droite) une force maximale et intermédiaire en jaune (gauche) et vert (droite)

On étudie finalement les singularités dans la configuration en  $\tilde{\lambda}$ . On représente le résultat en Figure 4.8. La configuration est étudiée avec des bras ayant un angle  $\beta$  à  $\frac{\pi}{3}$ . On note que la singularité apparaît pour un angle de  $\frac{\pi}{6}$  et non de  $\frac{\pi}{3}$  ce qui peut paraître surprenant. En effet, il paraît intuitif que le bras se retourne lorsque l'axe de celui-ci est à la verticale. Cependant si c'était le cas, le drone partirait sur la gauche ou la droite et ne serait plus en équilibre sur l'axe  $\vec{y}_I$ . La singularité apparaît plutôt lorsque le vecteur de poussée du bras opposé s'inverse sur l'axe  $\vec{y}_B$ .

Le problème de singularité apporte un déséquilibre pour le drone par le retournement rapide des bras. Puisque ceux-ci ne produisent pas de poussée, on peut envisager une zone morte autour de la singularité, où le bras conserve son orientation et ne produit pas de poussée. La section suivante propose de se concentrer sur la compensation du balourd engendré par l'ajout d'une masse déportée du centre de gravité. Cela concerne à la fois les batteries que l'on peut mettre sur le système, mais aussi les différents capteurs d'inspection.

## 4.4 Balourd

Dans cette section on analyse le problème de non-équilibre du système, dû à l'ajout d'un capteur. L'équilibrage en mécanique désigne une "Répartition des masses d'un système en rotation (roue, rotor, machine, etc.) de manière que le centre de gravité de l'ensemble soit situé sur l'axe de rotation (équilibre statique) ou que les forces d'inertie se compensent (équilibre dynamique)"<sup>4</sup>. Le non-équilibre du système est ainsi induit par un balourd, ou masse qui ne se situe pas sur l'axe de rotation d'un système. En admettant par exemple que le drone est parfaitement équilibré, le fait de rajouter un outil de mesure entraîne un changement de centre de gravité pouvant introduire un déséquilibre.

L'outil de mesure possédant une masse non négligeable, il faudra compenser ce déséquilibre. Cependant, deux points positifs peuvent être soulignés. Premièrement, de par la configuration du prototype, le capteur ne bougera pas dans le référentiel du drone. Étant solidaire du corps, on peut considérer une perturbation statique. Deuxièmement, la faible vitesse de rotation du corps sur lui-même permet de négliger les effets de force centrifuge liés aux balourds. Finalement on exprime facilement la force liée au balourd statique à compenser comme suit :

$$F_{bal} = m_{bal} g \quad (4.68)$$

avec  $F_{bal}$  la force (N) exercée par le balourd et  $m_{bal}$  la masse du balourd. Attention ici à ne considérer que la seule masse du balourd et non la masse de l'outil en tant que tel. Si il existe par exemple une batterie d'un côté il faudra soustraire cette masse à celle du capteur pour connaître la masse du balourd. Dans la loi de commande il suffira donc de rajouter ce couple. Une subtilité est néanmoins à considérer. Le couple lié à la force (4.69) dépend directement de l'orientation du drone avec l'expression suivante :

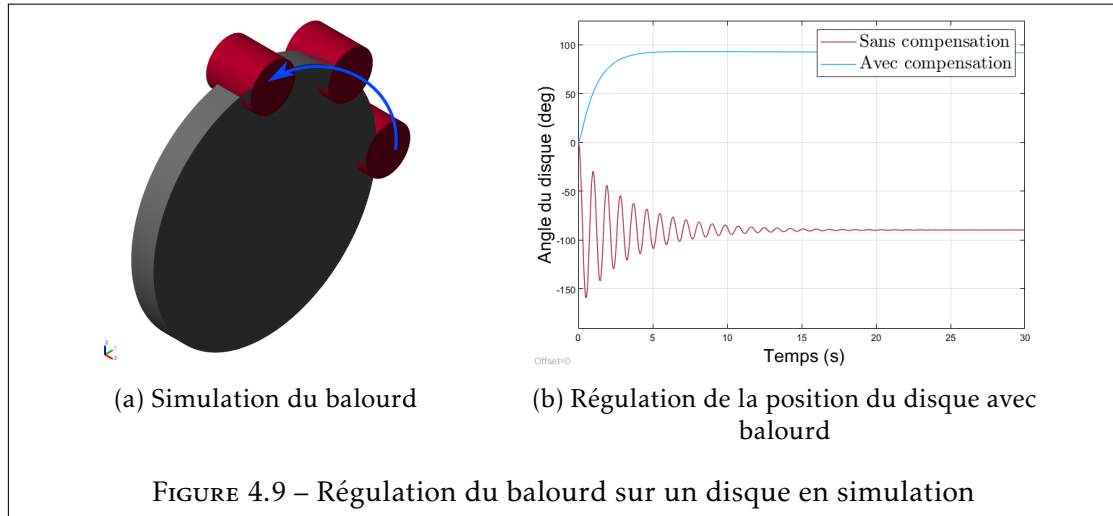
$$C_{bal} = \mathbf{L} \times^B \mathcal{R}_I F_{bal} \quad (4.69)$$

Avec  $\mathbf{L}$  le vecteur des distances entre le balourd et le centre de gravité. Cet algorithme a été testé sur un balourd fictif présenté sur la Figure 4.9. Le système consiste en un disque de répartition homogène auquel on a ajouté une masselotte (balourd). Le but de cette étude est de stabiliser la masselotte, initialement sur le coté, vers le haut du disque.

4. <https://www.larousse.fr/dictionnaires/francais/%C3%A9quilibrage/30671>



On note également que le disque ne possède qu'un seul DDL, à savoir une rotation sur son axe de révolution. Dans le scénario de test, on asservi le système par un contrôleur PD mais en appliquant une loi de commande avec et sans la compensation décrite ci-dessus. Les mêmes gains pour le PD sont utilisés pour asservir les deux systèmes.



Comme l'explique la figure, la compensation permet de stabiliser efficacement le système et sera donc implémentée dans le système.

On rediscutera de son impact dans le Chapitre 6.

## 4.5 Conclusion

Ce chapitre a permis d'aborder la question de la commande du système. Pour faire écho à la problématique, on propose une commande robuste qui semble correspondre directement aux besoins. Cependant les autres lois de commandes peuvent également stabiliser efficacement le système. De plus, ce chapitre a permis de soulever certaines questions supplémentaires. L'estimation des états a notamment été abordé et donne une passerelle avec le chapitre suivant qui est beaucoup plus pratique. A ce titre, l'analyse des singularités et des balourds est également un point à considérer pendant les essais. Enfin, la forme de commande par linéarisation permet une certaine flexibilité. Les couples gyroscopiques analysés au chapitre précédent peuvent par exemple simplement être compensés dans la loi de commande en soustrayant leur expression à la commande (comme pour les balourds). L'algorithme de linéarisation permet donc de stabiliser le système mais également de garder une certaine versatilité.

Un champ d'amélioration est possible notamment sur la compensation de la dynamique de l'actionnement des bras. En effet, les moteurs réagissent suffisamment rapidement pour que leur temps de réaction soit négligé, mais il en va autrement pour les servomoteurs responsables de l'inclinaison des bras. Leur latence induit des effets dynamiques sur le système, qui seront détaillés dans le chapitre de résultat. N'étant à ce jour pas compensé, ils représentent un champ d'amélioration possible pour le système.

Le chapitre suivant propose de se concentrer sur les plateformes informatiques et mécaniques du système. Dans un premier temps il sera décrit comment les prototypes ont été réalisés, ainsi que les infrastructures autour. Certains des points déjà traités dans ce chapitre comme le précédent, ont largement orienté la réalisation des véhicules. Dans un deuxième temps, on propose une solution informatique intéressante permettant de réaliser des tests très efficacement. Le système développé est extrêmement versatile (autant utilisable sur un véhicule terrestre qu'un drone classique ou celui de cette étude).

# Plateforme

Ce chapitre est destiné à présenter les différents choix faits au cours de la thèse pour réaliser deux prototypes. La première partie est dédiée à la réflexion faite pour concevoir mécaniquement le système. Les développements préliminaires permettent d’appréhender les actionneurs du système et de permettre des premiers tests.

Une deuxième partie présente ensuite plus en détail la conception des prototypes en configuration  $\times$  et  $\lambda$ . C’est l’occasion de présenter les éléments qui ont été spécialement réalisés pour le besoin spécifique de la thèse. La deuxième partie du chapitre s’axera sur la plateforme informatique développée. Après un rapide état de l’art sur les solutions existantes, une explication permettra de justifier le développement d’une plateforme personnalisée. Il est montré pourquoi celle-ci répond mieux à la dernière problématique que les solutions disponibles actuellement sur le marché

## 5.1 Plateforme mécanique

Une grande partie de l’étude était vouée à la conception et à la réalisation de prototypes. Ces développements ont été réalisés en parallèle des autres travaux durant les trois années. Un compromis a été fait le plus souvent possible entre le fait d’acheter les composants et de les réimprimer. La plupart du temps il était en effet plus intéressant d’acheter les éléments directement chez le vendeur quand c’était possible (exemple : les pieds d’atterrissage qui ne coutent pas cher à l’achat mais sont difficiles à imprimer de par leur forme). Cependant, on montre que ce n’est pas toujours possible de trouver une solution commercialisée, particulièrement pour des configurations de drones innovantes.

Le début de la phase de conception mécanique s’est fait en s’appuyant sur le matériel déjà disponible dans l’entreprise. Un ancien drone DJI F450 était disponible et permettait d’avoir une base intéressante pour une première version du prototype. Une réutilisation des moteurs et des plaques carbone constituant le châssis est faite, ainsi que les pieds d’atterrissage. Cette phase est faite en parallèle du développement d’un simulateur, décrit en Section 5.2.1. L’intérêt du simulateur est d’avoir un système qui reflète la réalité. Il est donc important de déterminer l’ensemble des variables du système avant de continuer. Ce point est discuté en Section 5.1.1.1.

Ensuite, une problématique concernant les rotations des bras est abordée. On fixe l’objectif de pouvoir tourner continuellement les bras sans avoir de blocage mécanique. Un temps conséquent a été dédié à ce sujet en proposant un premier système qui consiste à améliorer des servomoteurs classiques du marché pour garantir une rotation continue.

Une deuxième approche remplacera ensuite la première. Celle-ci se base sur une solution plus complexe et ne permet pas de rotation continue, mais laisse tout de même au système une grande ampleur de mouvement. L'ensemble des choix et difficultés liés à ces points sont discutés dans la Section 5.1.1.3. On a cherché à reproduire un outil en plastique pour avoir un mannequin à tester sur le drone. En cas de chute cela permettra de ne pas perdre le capteur. Plus de détails seront donnés en Section 5.1.1.4.

Suite à cette étude préliminaire on se concentre sur le châssis des deux prototypes. Des choix de conceptions sont faits nécessitant la conception de nombreuses pièces. La solution de prototypage au laboratoire est privilégiée car elle permet une versatilité totale par rapport à une solution sur étagère et permet d'obtenir les pièces très rapidement par impression 3D. Ce dernier point peut représenter un gain de temps conséquent.

On détaille toutes les pièces réalisées, en particulier les bras, de leur support structurel pour les servo-moteurs jusqu'aux protections d'hélices très facilement remplaçables. La conception mécanique visant réellement à obtenir un système optimisé vis à vis des contraintes (poids, résistance, réparabilité, ...). L'impression 3D permettra de donner des solutions personnalisées aux problèmes que l'on peut rencontrer. L'ensemble de ces données permet d'apporter des éléments de réponse à la troisième problématique <sup>1</sup>.

### 5.1.1 Développements préliminaires

Dans cette partie on donnera des précisions sur les développements indirectement liés au prototype, ou les phases de conceptions qui n'ont finalement pas abouti. Cela comprend :

- Banc d'essai pour la mesure des constantes  $k_f$  et  $k_\tau$  du groupe moteur-hélice
- Banc d'essai pour le réglage des gains du drone
- Conception de servomoteurs à rotation continue
- Numérisation des outils de mesure destinés à être embarqués sur le drone

#### 5.1.1.1 Conception d'un banc d'essai pour la mesure de $k_f$ et $k_\tau$

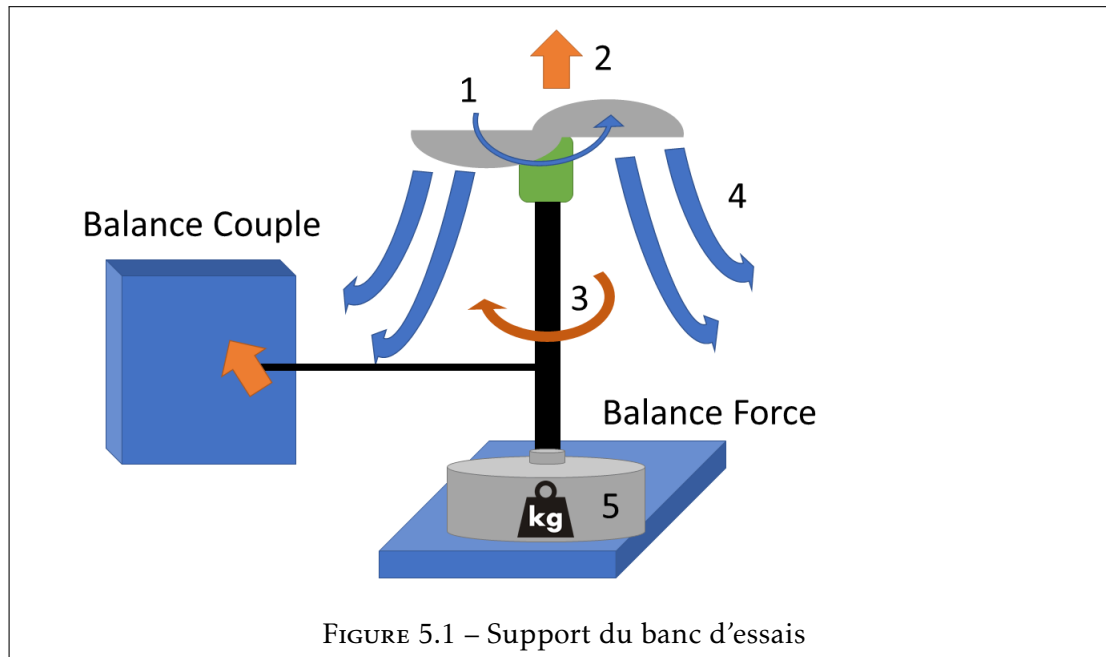
En aéronautique, un aéronef (avion, drone, montgolfière...) évoluant dans l'air sera soumis à des effets aérodynamiques. D'une part [95] détaille les effets auxquels un avion peut être généralement soumis ; la traînée, la portance, le centre de gravité et le centre aérodynamique.

D'autre part, [96]-[98] présentent plus précisément les effets auxquels sont soumis les drones (poussée et traînée parmi d'autres). On y trouve également des exemples de bancs d'essais pour mesurer les constantes, dont on s'inspirera pour réaliser le banc décrit ci-après. Pour la partie présentée ici, on propose de détailler la mesure des coefficients de traînée et de poussée des hélices en fonction de la vitesse de rotation des moteurs. Cela permettra de pouvoir cibler une loi de commande directement dans le micro-contrôleur, sans devoir réadapter les coefficients avant. Pour ce faire, on développe un banc d'essais qui permet d'une part de tenir le moteur, et d'autre part de mesurer les coefficients.

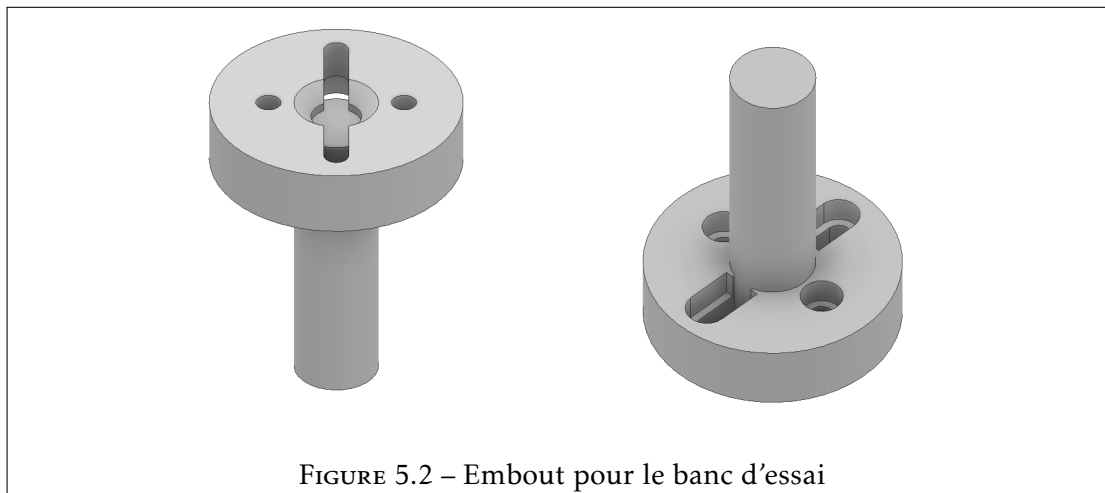
---

1. Quels points de conception mécanique devront être optimisés pendant la réalisation du dispositif?

Le banc aura la forme illustrée en Figure 5.1. L'hélice tournant en haut d'un mât (1) produira une force de poussée (2) mesurée par la balance en dessous et permettant d'estimer  $k_T$ . Ce processus produira également un couple de réaction. Le mât pouvant tourner vis à vis de son socle, une tige collée dessus pourra tourner avec la rotation engendrée par la trainée de l'hélice (3). La poussée de la tige pourra être mesurée par une autre balance. Connaissant sa distance il est alors possible d'en déduire le couple exercé. Le dispositif est placé en haut du mât pour que les perturbations aéronautiques (4) ne gênent pas la mesure. Enfin, on leste le dispositif (5) pour qu'il ne se soulève pas.



Dans la Section 5.1.2.3 on présente le support des moteurs, et dans la Section 5.1.2.5 on présente la liaison entre les tubes et les servomoteurs. Ces pièces seront adaptées pour le banc d'essais pour pouvoir à la fois tenir les moteurs, et les fixer à un support tournant. On réalise donc une concaténation des ces deux pièces pour pouvoir fixer un moteur à la verticale sur un tube en carbone. La Figure 5.2 montre cette pièce en CAO. La plupart des pièces imprimées en 3D sont semi-créuses et remplies avec un maillage souvent en forme de nid d'abeille. On note que contrairement aux autres pièces imprimées en 3D, celle-ci sera imprimée pleine afin de limiter les ruptures fortuites par torsion.



On réalise également un socle en deux parties pour solidariser la tige à un plateau tournant sur roulement. La pièce est présentée en Figure 5.3. Le même support que pour le moteur est également placé en bas du tube. Une vis traverse ensuite le support et la pièce à droite sur la figure, et vient se visser dans les deux emplacements de la pièce de gauche. Un roulement à bille sera alors pris au piège entre les deux pièces précédentes.

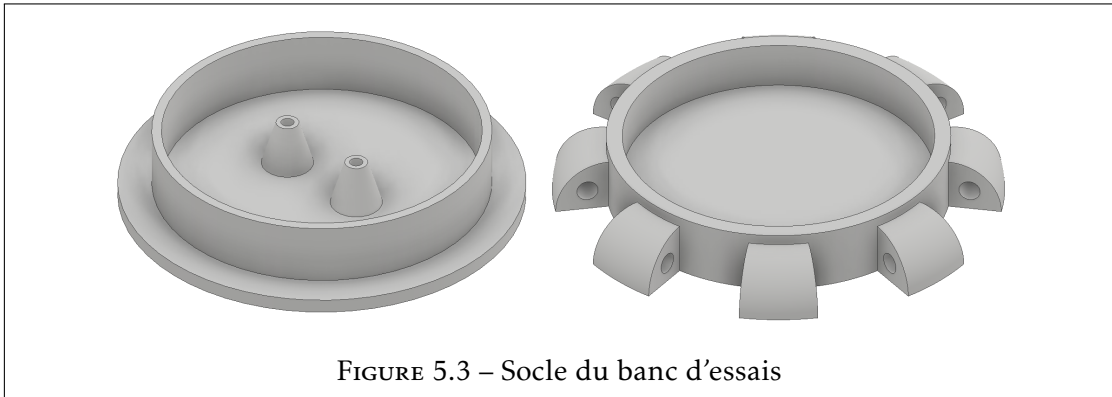
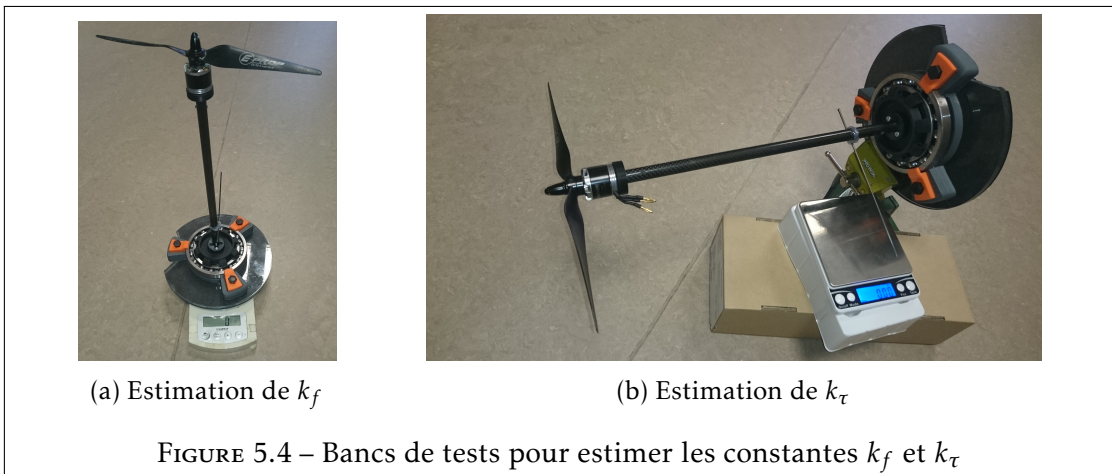


FIGURE 5.3 – Socle du banc d'essais

Le banc d'essais pour estimer les coefficients des moteurs décrit précédemment est représenté en Figure 5.4.



(a) Estimation de  $k_f$

(b) Estimation de  $k_\tau$

FIGURE 5.4 – Bancs de tests pour estimer les constantes  $k_f$  et  $k_\tau$

Remarque 1 : pour l'essai on utilise une tension constante à l'aide d'une alimentation stabilisée en 12V. Les PWM utilisés vont de 1100 $\mu$ s à 1900 $\mu$ s par pas de 50 $\mu$ s. De plus, chaque PWM possède sa propre fréquence. Pour des servomoteurs classiques de modélisme on utilise généralement 50Hz mais pour des Electronic Speed Controller (ESC) on peut monter à 200hz, garantissant la stabilité de vol du drone. Plus d'informations sur les PWM dans [99].

Remarque 2 : On choisit d'utiliser une alimentation stabilisée car cela permet de limiter le facteur de chute de tension pendant l'essai. Cependant en réalité c'est une batterie LiPo qui sera utilisée. Celle-ci verra sa tension chuter au fur et à mesure du vol, impactant directement la tension à l'entrée des moteurs. Ce point n'est pas un problème car le système est voué à être commandé en position. C'est donc la commande qui réglera directement ce problème.

L'essai est répété plusieurs fois pour garantir une mesure fiable. Pour l'essai de traction, on place directement le dispositif à l'aplomb de la balance et on mesure (par soustraction) la masse et donc la force de poussée de l'hélice.

On répète l'essai 3 fois pour chaque coefficient. On rappelle la définition de  $k_f$  et  $k_\tau$  :

$$F \approx k_f \omega^2 \quad (5.1)$$

$$\tau_{drag} \approx k_\tau \omega^2 \quad (5.2)$$

Avec  $F$  la force de traction verticale mesurée sur le banc d'essai.

Si l'on se réfère aux équations ci-dessus, on se rend compte que notre mesure ne correspond pas tout à fait. En effet, on mesure la force et le couple en fonction de la PWM et non de la vitesse de rotation des hélices. Dans un premier temps on se servira de cette mesure pour connaître la PWM à injecter dans le système en fonction de la force calculée par la loi de commande. Dans un deuxième temps on mesure la vitesse de rotation des hélices en fonction de la PWM pour estimer les facteurs  $k_f$  et  $k_\tau$ . On pourra en déduire la constante  $\delta$  qui est le rapport entre les deux facteurs précédents.

On montre en Figure 5.5 le résultat de cette mesure pour l'essai de traction. Les résultats pour l'essai de trainée seront montrés plus tard. On ne répète l'essai que trois fois, les résultats étant cohérents entre eux et la mesure assez fiable.

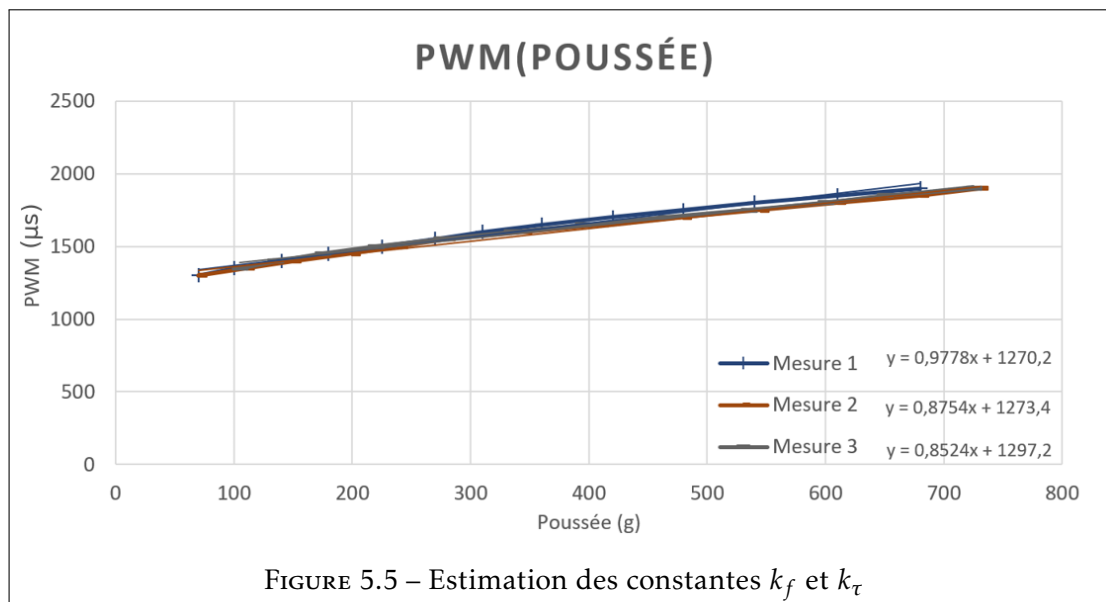


FIGURE 5.5 – Estimation des constantes  $k_f$  et  $k_\tau$

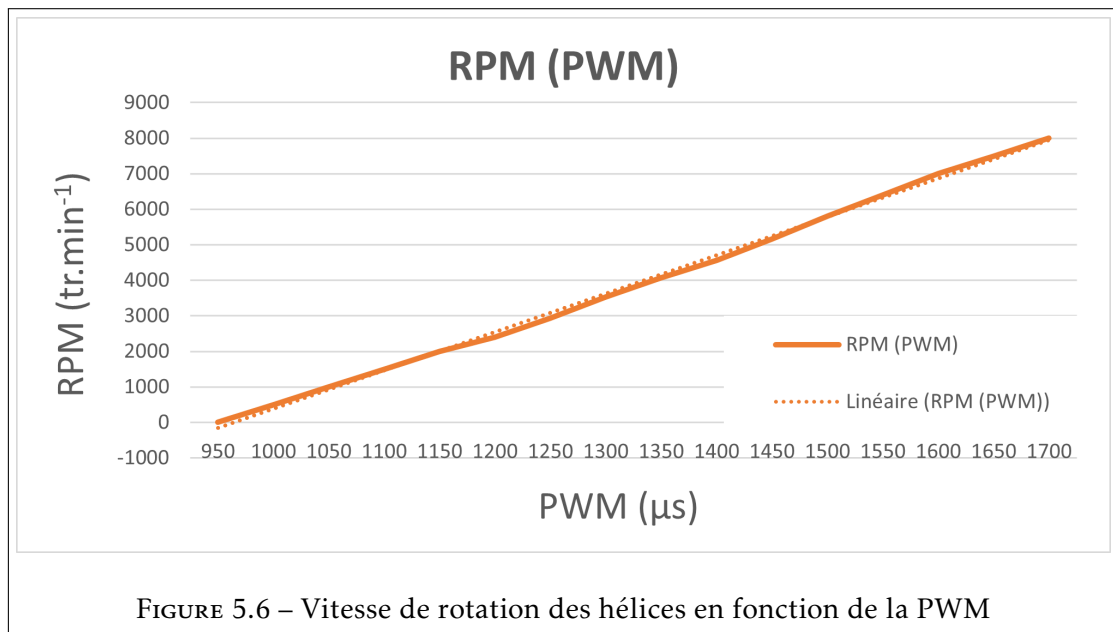
Sur la figure on identifie clairement une fonction qui semble affine mais légèrement inclinée ce qui correspondrait à un terme quadratique. On identifie alors que la PWM injectée serait directement proportionnelle à la vitesse de rotation. D'après la formule (5.1), il est alors logique d'avoir un terme quadratique qui apparaît. On propose l'expression suivante pour lier la PWM à la Force de traction (et de manière analogue au couple de rétroaction des hélices) :

$$PWM = \sqrt{aF + b} \quad \text{avec} \quad F = mg \quad (5.3)$$

$$\text{d'où} \quad (5.4)$$

$$PWM = \sqrt{amg + b} \quad (5.5)$$

Avec  $a$  et  $b$  deux constantes et  $m$  la masse soulevée par l'hélice, lue directement sur la balance. Pour valider la théorie précédente, on mesure la vitesse de rotation de l'hélice en fonction de la PWM injectée. Pour ce faire on utilise un tachymètre sur le dispositif. Le résultat est donné en Figure 5.6.



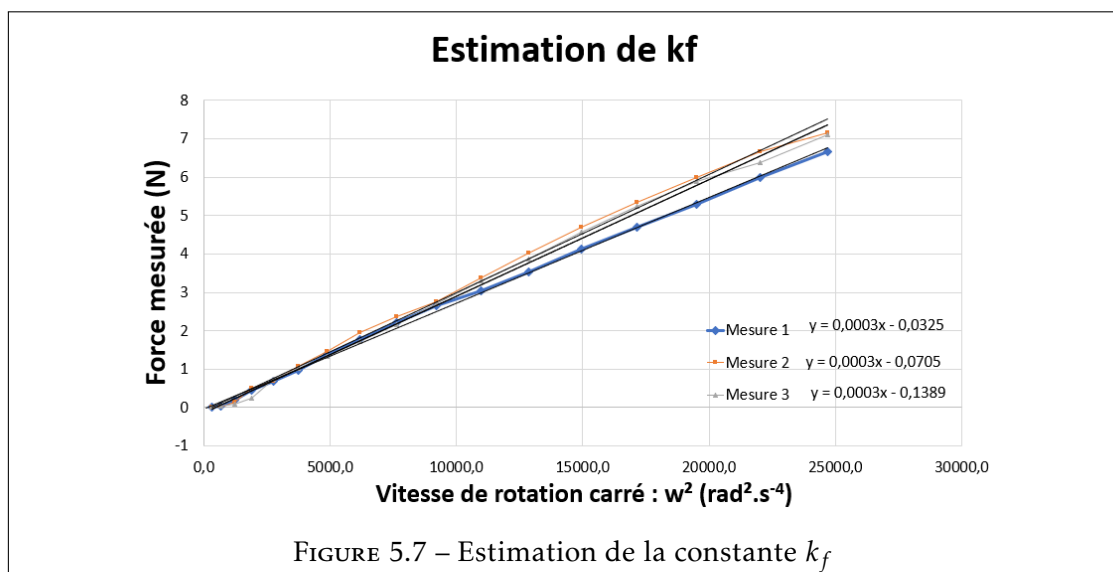
On en déduit une relation linéaire entre la vitesse de rotation des hélices et la PWM injectée.

Entre 950 et 1700  $\mu\text{s}$  de PWM, cette relation est comme suit :

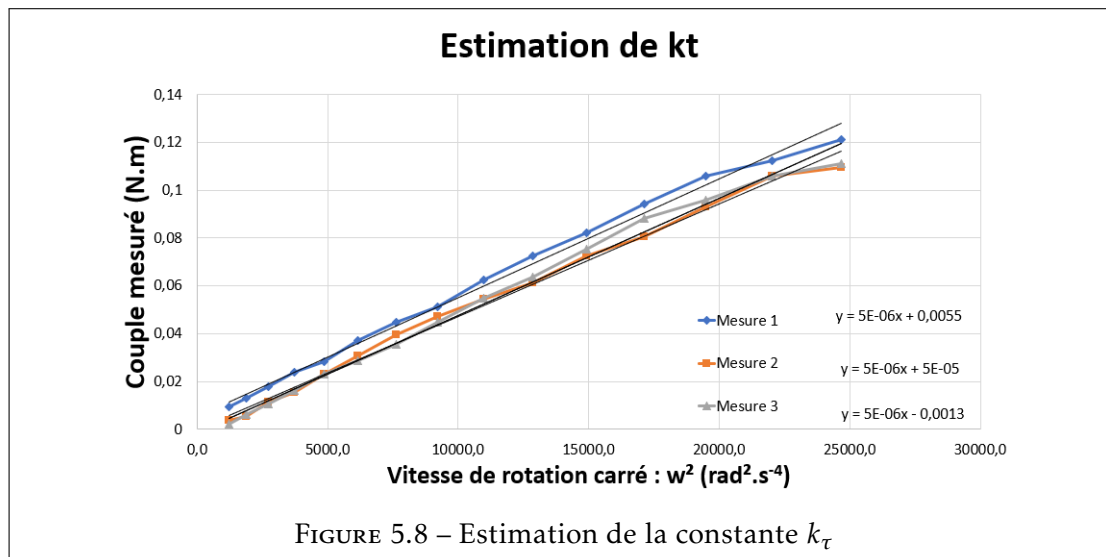
$$RPM = 10.(PWM - 950) \quad (5.6)$$

On déduit du graphique 5.5 les coefficients  $a$  et  $b$  de la fonction affine qui définit la PWM. On choisit  $a = 0.9$  et  $b = 1270$ . Par la suite on utilisera ces coefficients à la fois dans la simulation et dans les tests réels.

Connaissant la vitesse de rotation (en tours par minute) des hélices en fonction de la PWM injectée, on peut estimer les forces et couples du dispositif et en déduire une estimation de  $k_f$  et  $k_\tau$ . Le résultat est présenté en Figure 5.7 et en Figure 5.8.



On utilisera  $k_f = 0.003$  et  $k_\tau = 5e - 06$ . Ces résultats correspondent à l'ordre de grandeur que l'on peut trouver dans la littérature. On cite par exemple les coefficients dans [36] qui concernent un drone de grande ampleur, à savoir  $k_f = 0.0047$  et  $k_t = 0.000228$ .



### 5.1.1.2 Conception d'un banc d'essai pour le réglage des gains du drone



Pour régler les gains du système on réalise deux trépieds permettant de surélever le drone. La deuxième version est montrée en Figure 5.9. Il s'agit de placer le drone sur un axe, lui laissant la possibilité de pouvoir pivoter.

La liaison est réalisée au moyen d'une pièce faisant le liens entre des roulements sur l'axe et la structure du drone. Les deux autres rotations (en comptant le lacet) ne sont pas commandés pendant l'essai. Le trépied est lesté permettant d'utiliser une poussée nominale du drone, ce qui renforce l'acuité du test. Le fait d'avoir une infrastructure fixe permet également de pouvoir profiter d'une alimentation externe, ce qui limite la logistique et les problèmes de pertes de tension pour le système.

La première version du système ne permettait pas de réglage en hauteur, il s'agissait simplement de deux rectangles en PolyChlorure de Vinyle (PVC). La deuxième version, réalisée par Thierry MONGLON, que je remercie ici, permet d'ajuster la hauteur au moyen de réglettes en métal latérales que l'on peut distinguer sur les photographies.



De multiples orifices sont faits tout au long de la règle permettant d'ajuster la hauteur en fonction des besoins. Cette deuxième version est de fait pliable, ce qui permet de l'emporter sur des salons comme cela à été fait pour le salon SIDO de 2020.

Remarque : Le trépied change la dynamique du système ce qui biaise le réglage des gains mais permet tout de même d'offrir une base de réglage très intéressante qui ne sera que peut modifiée par la suite. De plus, elle permet d'offrir une sécurité et une ergonomie (alimentation filaire) qui ne sont pas négligeables. Après la phase de réglages sur ce trépied, sur les deux axes, la machine à été testée directement en vol libre. Le premier vol s'est très bien passé validant l'utilisation du dispositif.

### 5.1.1.3 Rotation continue des servo-moteurs

Pour le prototype de drone envisagé, il est nécessaire de pouvoir incliner les bras sur une grande plage d'angle. Pendant les 18 premiers mois de la thèse, on a cherché à se concentrer sur une solution d'intégration de capteurs de position angulaire dans des servomoteurs du commerce. Les capteurs sont en réalité des magnétomètres au-dessus desquels on place un aimant qui tournera avec l'axe de fin de réduction des servomoteurs. L'intérêt est d'une part de pouvoir utiliser n'importe quel servomoteur après l'avoir modifié. D'autre part, la commande est très simple puisque l'on utilise directement les PWM en sortie du micro-contrôleur. L'Annexe D.1 décrit plus en détails ces travaux.

Cependant, l'utilisation de ce système s'est trouvée être compliquée car plusieurs contraintes sont survenues. On cite par exemple le manque de place pour l'ajout des capteurs de positions, l'ajout d'un micro-contrôleur pour gérer la position des servomoteurs ou encore une mauvaise précision du capteurs due à la proximité directe avec les moteurs perturbant les mesures magnétiques. Le choix final s'est donc porté sur l'emploi de servomoteurs Dynamixels<sup>2</sup> qui possèdent un très grand nombre de fonctionnalités comme la variation de position angulaire à 300°, avec une grande précision. Ceux-ci possèdent également un réglage PID intégré et ajustable permettant d'obtenir un réponse très rapide à une consigne de position angulaire. Leur utilisation nécessite en revanche un certain nombre d'aménagements décrits dans la section 5.2.2. Une illustration de cet actionneur est donné en Figure 5.10, source<sup>3</sup>.



2. De la marque Robotis™

3. [https://static.generation-robots.com/15881-product\\_cover/servomoteur-dynamixel-ax-12-a-robotis.jpg](https://static.generation-robots.com/15881-product_cover/servomoteur-dynamixel-ax-12-a-robotis.jpg)

#### 5.1.1.4 Numérisation des outils de mesures

Souhaitant intégrer les outils de mesure du domaine BTP directement sur le drone, il était intéressant de les modéliser afin d'avoir un modèle fidèle en CAO. Ce modèle a ensuite pu être utilisé, d'une part pour réaliser une reproduction de l'outil à des fins de tests; d'autre part, elle contribue à réaliser un porte outil parfaitement ajusté.

On commence par la modélisation d'un ferroskan de la marque Hilti™. La numérisation se fait à l'aide d'un scanner 3D de la marque EinScan™ en suivant les instructions constructeur. La Figure 5.11 présente l'outil pendant la procédure de scan.



FIGURE 5.11 – Procédure de scan de l'outil de mesure

Le résultat de la numérisation est donné en Figure 5.12 à gauche et son impression 3D par dépôt de fil à droite (à côté de la version originale).



FIGURE 5.12 – Modélisation CAO, reproduction et véritable ferroskan

Par la suite on cherche à concevoir un porte outil capable de maintenir l'appareil de mesure en contact avec la paroi. On propose une version de celui-ci en Figure 5.13. L'attache de l'outil au drone se fera à l'aide d'une pince en métal élastique type porte-balais<sup>4</sup>. Cela permettra de tenir l'outil tout en gardant une rotation possible. On rajoute ensuite un système de maintien avec ressort permettant de plaquer l'outil sur la surface de la paroi. Même si le drone n'est pas parfaitement aligné, le système d'amortisseur réajustera le déséquilibre.

4. exemple de référence <https://www.leroymerlin.fr/v3/p/produits/support-bloque-manche-mottez-h-5-x-l-5-x-p-3-6-cm-e10906>

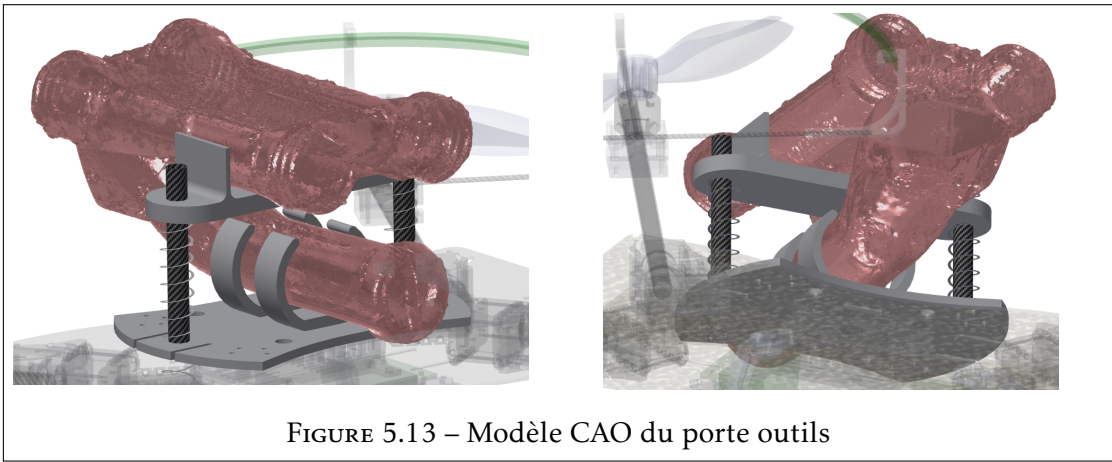


FIGURE 5.13 – Modèle CAO du porte outils

Les développements collatéraux étant explicités, on s'intéresse maintenant aux points de conceptions dédiés directement au châssis du drone.

### 5.1.2 Conception pour les prototypes

Dans cette deuxième section on propose de détailler les parties du système de drone qui ont été développées. Seules les configurations  $\times$  et  $\lambda$  ont été finalement produites. En effet, la configuration en "+" possède sa singularité au point de fonctionnement le plus utilisé du robot. On propose un schéma de principe permettant d'expliquer l'emplacement des différents éléments à placer sur le drone en Figure 5.14.

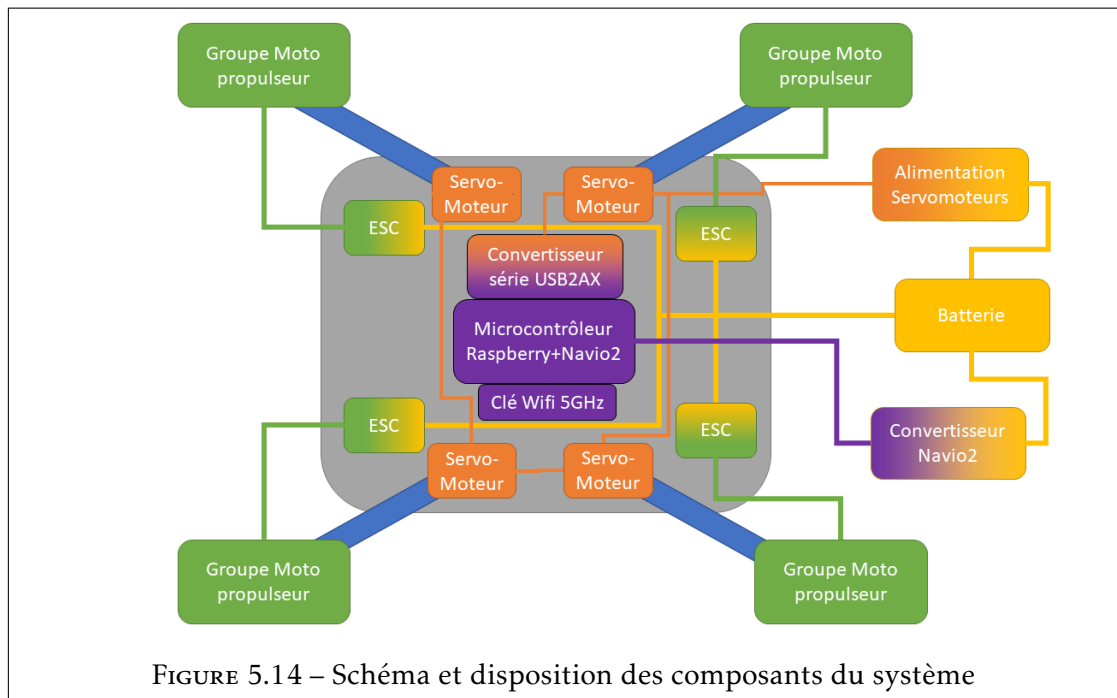


FIGURE 5.14 – Schéma et disposition des composants du système

La plupart des éléments se situent dans le cœur même du drone, limitant au maximum le poids sur les bras. Cela permet aux servomoteurs de pouvoir tourner les bras plus facilement tout en gardant un corps qui ne sera pas affecté par ces couples de rotation. Dans cette section on présente toutes les pièces faites en conception par ordinateur. L'intérêt de cette méthode est d'obtenir des pièces qui répondent parfaitement au besoin que l'on a tout en donnant une disponibilité immédiate. Ce dernier point est très intéressant en cas de casse sur le drone pour pouvoir continuer rapidement des essais.

Mais la conception permet un deuxième avantage. Les pièces peuvent être importées dans le simulateur décrit en Section 5.2.1. Dans le cas où l'on achète des pièces il faut le plus souvent les re-modéliser. Le fait d'être dans une démarche de conception puis de simulation et d'impression par dépôt de fil réduit l'intérêt des composants sur étagère. Certaines pièces seront cependant achetées car leur forme est difficilement imprimable (train d'atterrissage par exemple), que leur prix est très modeste et surtout que les plans sont disponibles sur internet.

Pour les prototypes on propose de détailler les points suivants :

- dispositif d'amortissement pour le support du Micro-contrôleur Navio2+Raspberry Pi;
- dispositif de protection des biens et des personnes;
- optimisation de l'accouplement entre servomoteur et tube en carbone;
- support pour les moteurs brushless;
- support pour les servomoteurs
- support pour les batteries;
- plaques pour la deuxième version du prototype.

Les impressions 3D prenant souvent un peu de temps, il est intéressant de n'imprimer qu'une seule pièce à la fois en journée. Cela permet d'identifier tous les problèmes de conception, de les corriger et de profiter de la nuit pour ré-imprimer les pièces optimisées, en série. En gardant cette discipline il était alors possible de ne pas perdre de temps et de travailler sur d'autres points en parallèle.

### 5.1.2.1 Amortissement pour le Navio

Afin de ne pas perturber les capteurs présents sur le Navio il est nécessaire de disposer d'un système d'amortissement permettant le découplage mécanique vibratoire entre le Navio et le drone. Dans la première version du prototype, le Navio est placé sur le dessus du drone. On s'inspire de l'amortisseur proposé sur le site internet de Emlid<sup>5</sup>, que l'on modifie pour pouvoir être intégré sur la plaque supérieure du drone F450. Le résultat est donné en Figure 5.16.

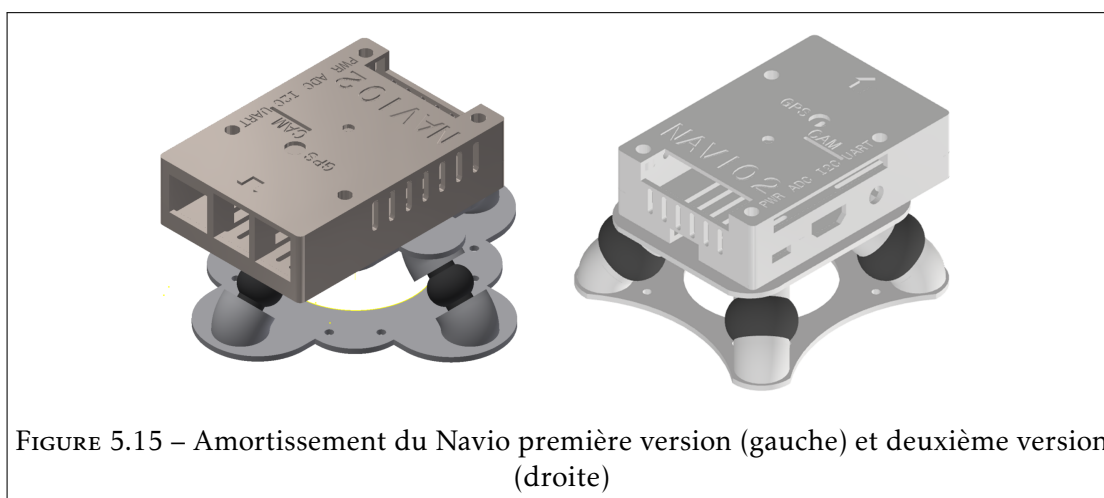


FIGURE 5.15 – Amortissement du Navio première version (gauche) et deuxième version (droite)

5. <https://docs.emlid.com/navio2/ardupilot/hardware-setup/>

La partie en noir représente des amortisseurs en caoutchouc permettant au système de bouger plus librement. Par la suite, pour la configuration en  $\tilde{\lambda}$  on souhaite réaliser une structure qui intégrera directement le Navio dans le centre de la structure. Pour cette deuxième version on propose d'utiliser également des caoutchoucs d'amortissement mais cette fois-ci c'est le boîtier du Navio qui sera modifié, voir Figure 5.16.

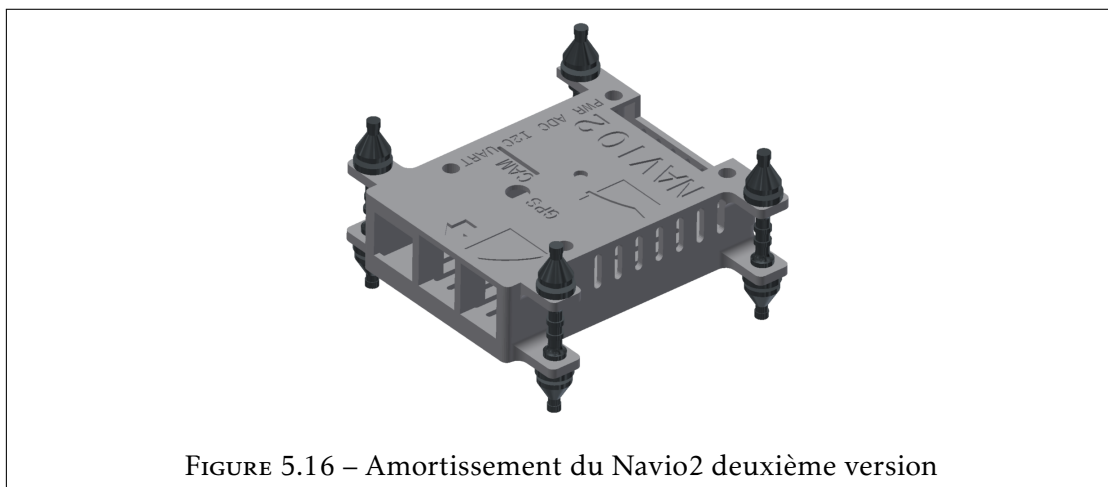


FIGURE 5.16 – Amortissement du Navio2 deuxième version

L'idée de ce boîtier est de contenir le micro-ordinateur entre deux plaques en carbone spécialement dessinées pour le prototype. Celles-ci sont d'ailleurs immédiatement décrites ci-après.

### 5.1.2.2 Plaques pour la deuxième version du prototype

La première version du prototype ne permet pas d'incliner le drone avec de grands angles. En effet, la configuration en "X" n'est pas optimisée car les forces latérales se compensent et engendrent donc une perte de poussée globale, voir Section 3.16. Ceci n'est pas un problème pour de faibles inclinaisons mais l'est pour des inclinaisons importantes, comme un vol vertical par exemple.

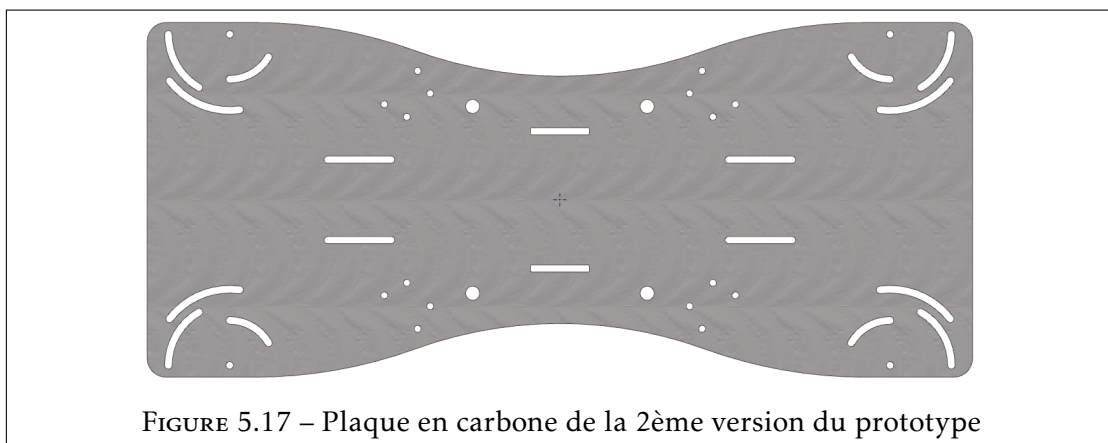


FIGURE 5.17 – Plaque en carbone de la 2ème version du prototype

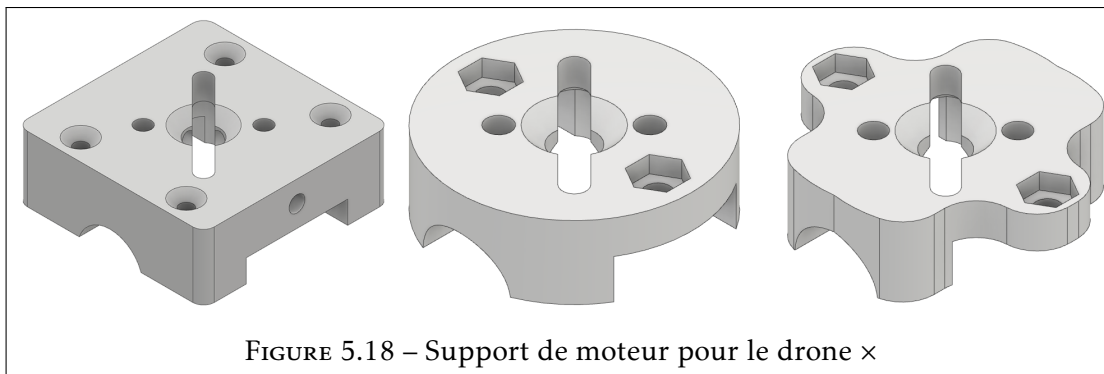
Pour palier ce problème, on propose de développer un deuxième châssis permettant une meilleure optimisation de la compensation de la force de pesanteur. Cette configuration ressemble davantage à un double "Y" permettant d'obtenir des angles de vol stationnaire bien plus importants en gardant la même plateforme électronique. L'intérêt de la configuration en  $\tilde{\lambda}$  est largement discuté dans le chapitre 3, notamment sur la partie décrivant la consommation énergétique.

On profite de l'élaboration du nouveau châssis pour y intégrer des optimisations de conceptions comme l'ajout de rainures pour serrer la batterie contre le châssis. On ajoute également des rainures permettant de régler l'angle des bras pivotants. Enfin, comme on l'a décrit précédemment, on profite de l'occasion pour intégrer le Navio directement entre les deux plaques de carbone constituant le châssis. Le résultat est montré en Figure 5.17. La plaque est en fibres de carbone tissé. Ce matériaux permet une rigidité incroyable pour un poids dérisoire et est donc très privilégié en aéronautique.

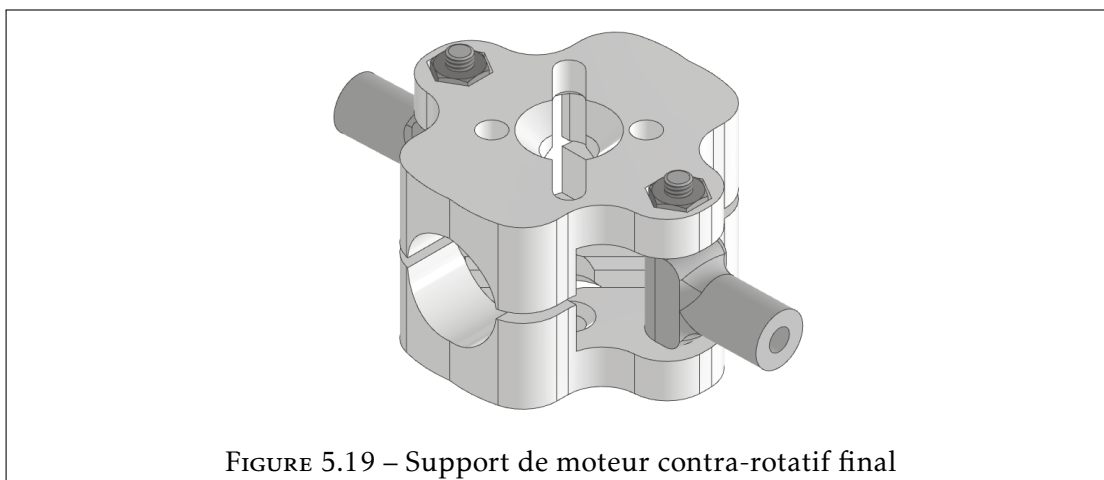
La partie suivante concerne les bras du drone avec les portes moteurs, les portes servos et l'accouplement entre servo et tube en carbone tissé dont sont constitués les bras.

### 5.1.2.3 Support pour les moteurs brushless

Un porte-moteur à été développé. Celui-ci permet non seulement de lier les moteurs brushless au tube en carbone, mais également d'accueillir la protection des bras grâce a des trous permettant d'y insérer des tiges en carbone. Plusieurs versions ont été développées, voir Figure 5.18, avec un dégradé de gauche à droite de la plus ancienne à la plus récente :



Parmi les plus grands changements, la première version ne nécessitait pas d'écrou. En effet, la vis était noyée dans le trou de la pièce et venait se visser dans une partie inférieure. La version suivante possède un emplacement spécifique pour un écrou en métal, renforçant le mécanisme. On réduit également par deux le nombre de vis nécessaires ce qui réduit le poids global. Enfin, dans la dernière version on réduit la quantité de matière. De plus on écarte les emplacements pour les vis, car ils étaient trop proches des moteurs.



Les essais sur le premier drone montrent que les perturbations liées à la trainée des hélices et les couples gyroscopique sont un problème pour le système. On choisit donc d'opter pour un système octorotor avec quatre paires de moteurs les uns au dessus des autres. La dernière version du porte moteur est donc imprimée deux fois pour au dessus et en dessous, cf Figure 5.19. Les deux pièces sont quasi identiques à l'exception des deux trous pour les vis. Dans l'un des cas on a l'emplacement pour les écrous et dans l'autre on creuse une rainure pour les têtes de vis.

Note 1 : on aurait pu réaliser une pièce avec une encoche d'écrou d'un côté et une tête de vis de l'autre. Cela aurait permis de réaliser qu'une seule pièce au lieu de 2 différentes, et donc de simplifier la conception de la pièce. Note : on réalise toutes les pièces en *PoLyactic Acid* (polymère et alternative naturelle au polyéthylène) (PLA) car ce plastique offre une meilleure rigidité que l'Acrylonitrile butadiène styrène (ABS) qui est plus souple.

#### 5.1.2.4 Dispositif de protection des biens et des personnes

Comme évoqué sur les porte-moteurs, on prévoit un emplacement pour insérer des baguettes en carbone visant à obtenir une protection des hélices. Sur les drones c'est une pratique répandue d'utiliser des protections pour limiter les dégâts avec les biens et les personnes. Dans le cas précis des protections d'hélices on aurait donc tout à fait pu acheter une solution du commerce. Cependant il aurait fallu réadapter le porte moteur, rajoutant certainement de la matière. De plus, la plupart des protections sont d'un seul tenant, il est donc nécessaire de changer l'ensemble ce qui peut représenter une perte de temps. On choisit de concevoir une protection où les pièces sont extrêmement faciles à changer et les plus optimisées possible. On présente en Figure 5.20, l'extrémité du bras du drone dans sa première version, vu du dessus (à gauche) et du dessous (à droite).

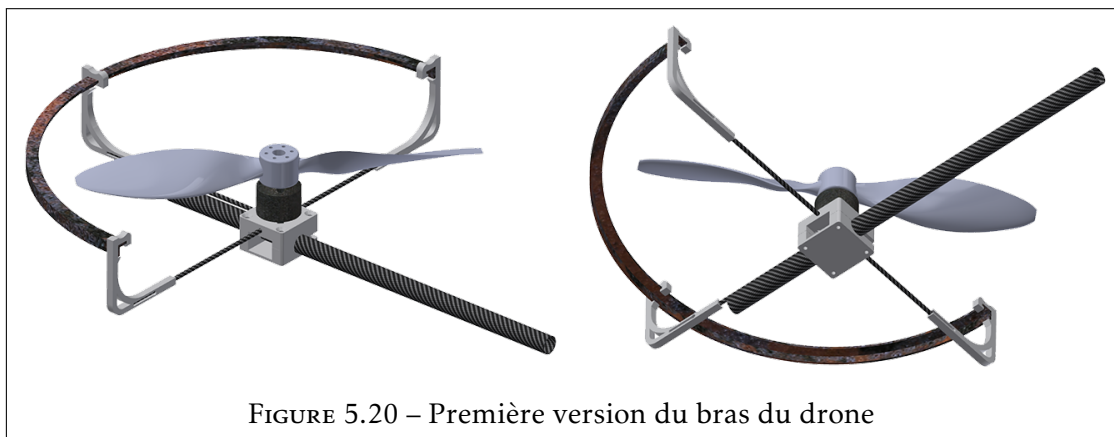


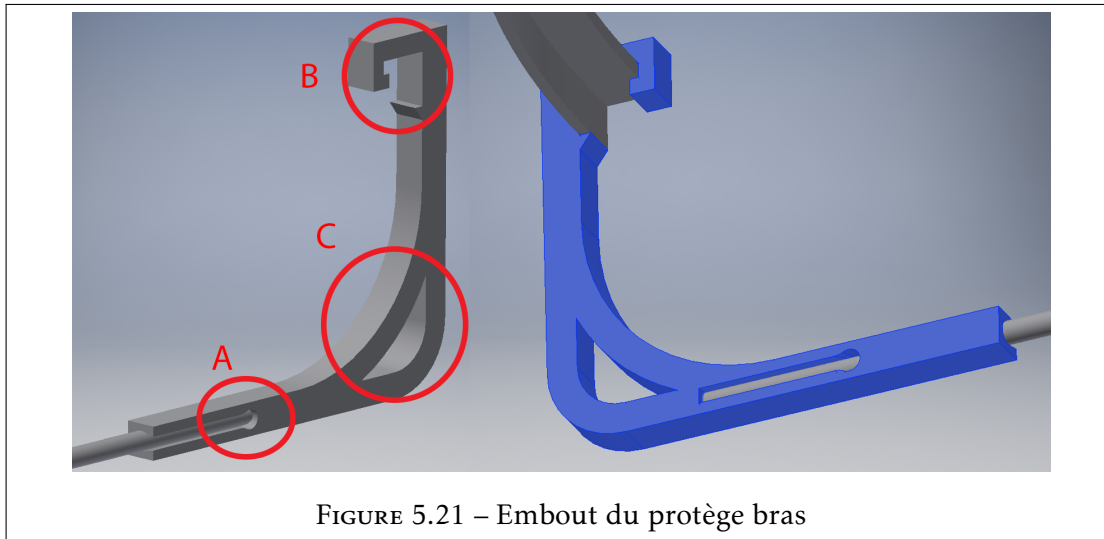
FIGURE 5.20 – Première version du bras du drone

Le tube et les tiges sont en fibre de carbone (tressé pour le tube et normal pour les tiges) permettant d'avoir un produit souple et robuste. Une partie intéressante est présente dans l'accouplement entre les tiges en carbone et l'arc de protection d'hélice.

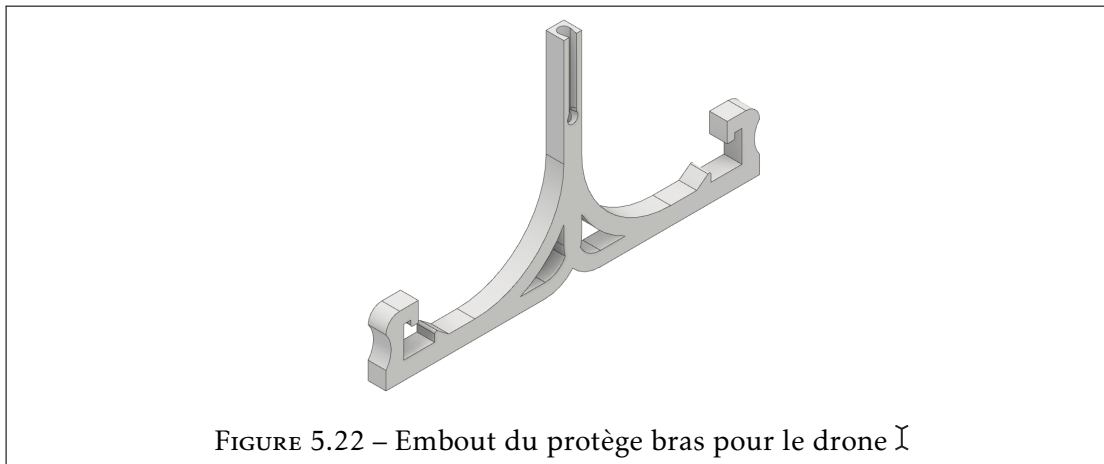
Cette pièce possède trois éléments particuliers, voir Figure 5.22. Premièrement, l'accouplement avec les joncs en carbone. Un trou est prévu pour accueillir la tige. L'assemblage se réalise en insérant le jonc perpendiculairement à la pièce dans le petit trou transversal (A), puis on rabat la pièce à 90° pour clipser l'ensemble.

Deuxièmement, en (B), on réalise l'accouplement de l'arc anti-chock également par clipsage. L'arc en forme de "L" est inséré dans l'encoche supérieure et reste en place après passage en force dans l'encoche inférieure.

La partie (C) enfin, permet d'enlever de la matière tout en gardant une certaine rigidité structurelle.



Pour la configuration en  $\Sigma$  on choisit de réaliser un octorotor qui aura donc besoin d'une protection d'hélice en haut comme en bas. On double donc tout simplement la pièce précédente pour protéger les deux hélices, voir Figure 5.22.



#### 5.1.2.5 Optimisation de l'accouplement entre servomoteur et tube en carbone

On présente en Figure 5.23 les différentes versions des embouts conçus pour accoupler les tubes en carbone avec les servomoteurs. Dans un premier temps on pensait partir sur l'idée d'utiliser des servomoteurs du commerce. Le premier embout de l'image devait donc s'emboîter directement dans le servomoteur et être fixé à lui à l'aide d'une petite vis insérée par l'autre extrémité. Cependant, l'expérience a rapidement montré que les bras pouvaient tourner à vide et que le couple n'était pas transmis correctement. La deuxième et troisième version prévoit donc d'utiliser les embouts des servomoteurs et de les insérer dans une encoche prévue à cet effet. On note que l'accouplement entre cet embout et le tube en carbone tressé se fait à l'aide d'une goupille en carbone insérée transversalement. Enfin, la dernière version représente un embout à accoupler aux Dynamixels. L'embout est directement vissé dessus, et on utilise également une goupille pour la liaison entre tube et embout. Pour limiter le jeu et les éventuels mouvements on utilisera du scelle-roulement entre le tube la goupille et l'embout.



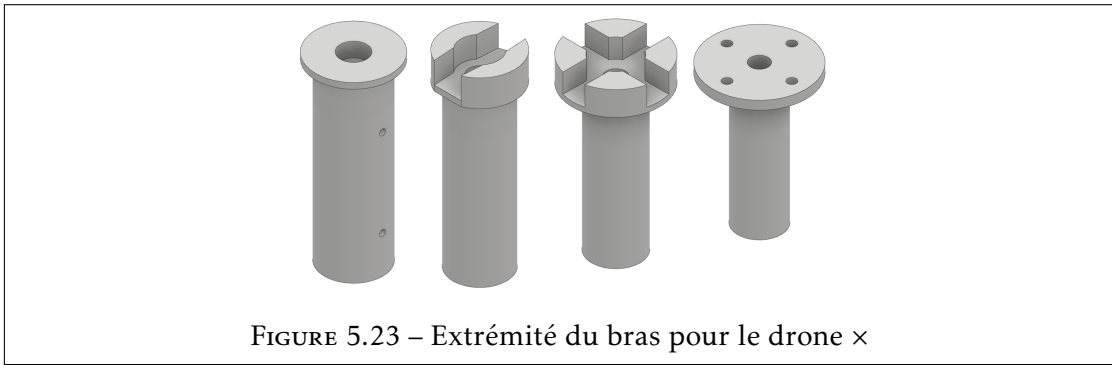


FIGURE 5.23 – Extrémité du bras pour le drone ×

En terme d'impression, il faut faire attention au sens. Le dépôt de fil permet de réaliser des pièces rapidement mais celles-ci sont sensibles au cisaillement dans le sens d'impression. Ainsi pour optimiser la structure des embouts décrits, on choisit d'imprimer les pièces dans le sens de la longueur et non de la hauteur. La laboratoire possède une solution pour imprimer du fil soluble en PolyVinyl Alcohol (PVA) ce qui rend l'impression tout à fait praticable.

Dans la description des pièces conçues pour les bras on arrive finalement au centre du drone. C'est dans la structure en elle-même que l'on place les portes-servos.

#### 5.1.2.6 Support pour les servomoteurs

Outre le fait de tenir les servomoteurs, les supports ont un réel intérêt structurel. Ils permettent de joindre les deux plaques en carbone décrites précédemment assurant une rigidité au drone. On présente en Figure 5.24 une des premières versions du porte-servo permettant le lien entre le tube en carbone tressé, le servomoteur et le châssis.

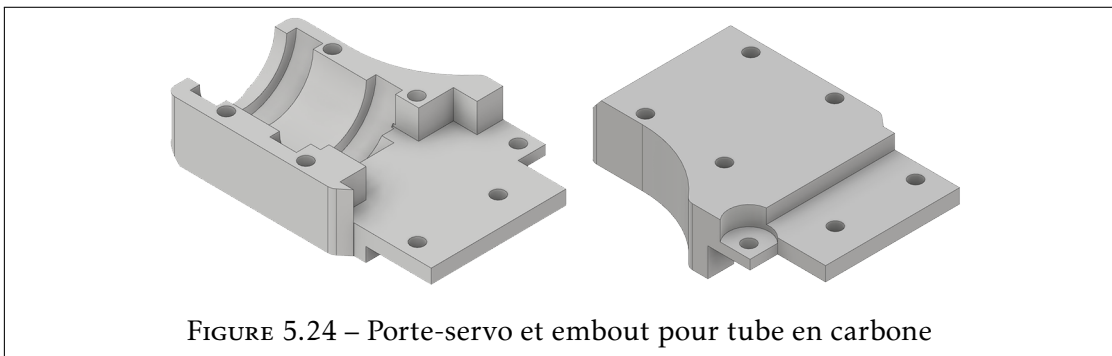


FIGURE 5.24 – Porte-servo et embout pour tube en carbone

On prévoit d'insérer deux roulements à billes entre cette pièce et le tube en carbone tressé pour que ce soit la structure qui subisse tout les efforts de flexion dû à la poussée des moteurs. Ainsi le couple (axe  $\vec{y}_{arm}$ ) n'est pas directement appliqué sur les servomoteurs. Le fait d'utiliser des roulements permet de conserver le moment que les servomoteurs appliquent sur les bras (axe  $\vec{x}_{arm}$ ). Cette première version pouvait accueillir des servos 9g. Les différents orifices servent à pouvoir visser les deux plaques créant la structure du drone.

En terme d'assemblage, la première plaque se positionne au-dessus et la deuxième en-dessous. On assemble les deux blocs avec le bras, les roulements et le servomoteur assemblés par la goupille. Une fois les deux blocs vissés par les plaques, on visse les servos transversalement dans les supports. On présente une vue éclatée en Figure 5.25.

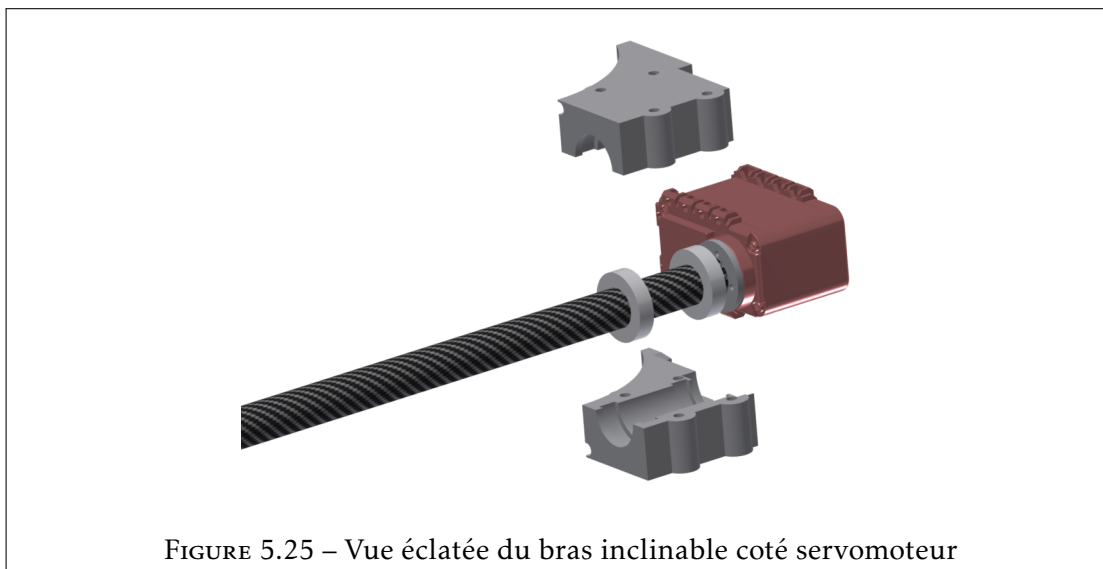


FIGURE 5.25 – Vue éclatée du bras inclinable coté servomoteur

Par la suite on re-dessine cette pièce pour pouvoir accueillir les Dynamixels. Ceux-ci étant relativement épais, on retire la partie au-dessus et en-dessous du servo. On prévoit des orifices pour accueillir les vis permettant de fixer la pièce frontalement sur les servos. Le résultat est proposé en Figure 5.26.

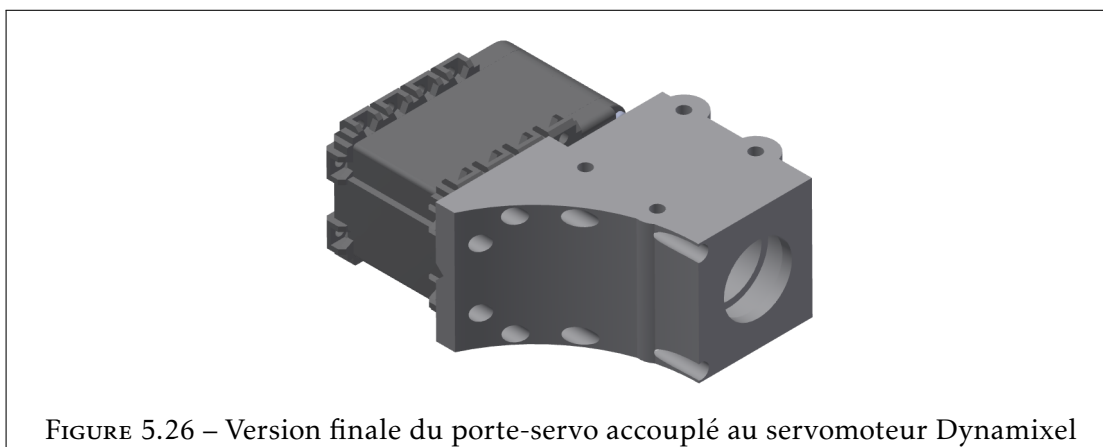


FIGURE 5.26 – Version finale du porte-servo accouplé au servomoteur Dynamixel

Une autre solution consiste à placer les servos en bout de bras. L'intérêt est un accès simplifié permettant un changement rapide le cas échéant.

### 5.1.2.7 Support pour les batteries

Le drone fonctionne avec des batteries Lithium Polymère (LiPo) de grande capacité (5000 mAh généralement). Celles-ci sont assez volumineuses et lourdes. Elles permettent cependant de fournir stocker une quantité d'énergie conséquente. Ne pouvant pas loger dans le châssis, et de par le manque de place lié au support des servomoteurs, il est nécessaire de les fixer à l'extérieur. Dans la configuration  $\times$  Le support du Navio est sur le dessus du drone, il ne reste donc que la partie inférieure de libre pour les batteries. Un support pour bloquer leur déplacements latéraux est dessiné. Celui-ci est montré en Figure 5.27.

Il permet de bloquer le déplacement de la batterie sur deux axes. Le dernier axe (vertical) reste libre pour pouvoir sortir la batterie. Elle sera serrée au support à l'aide d'un collier Velcro®.

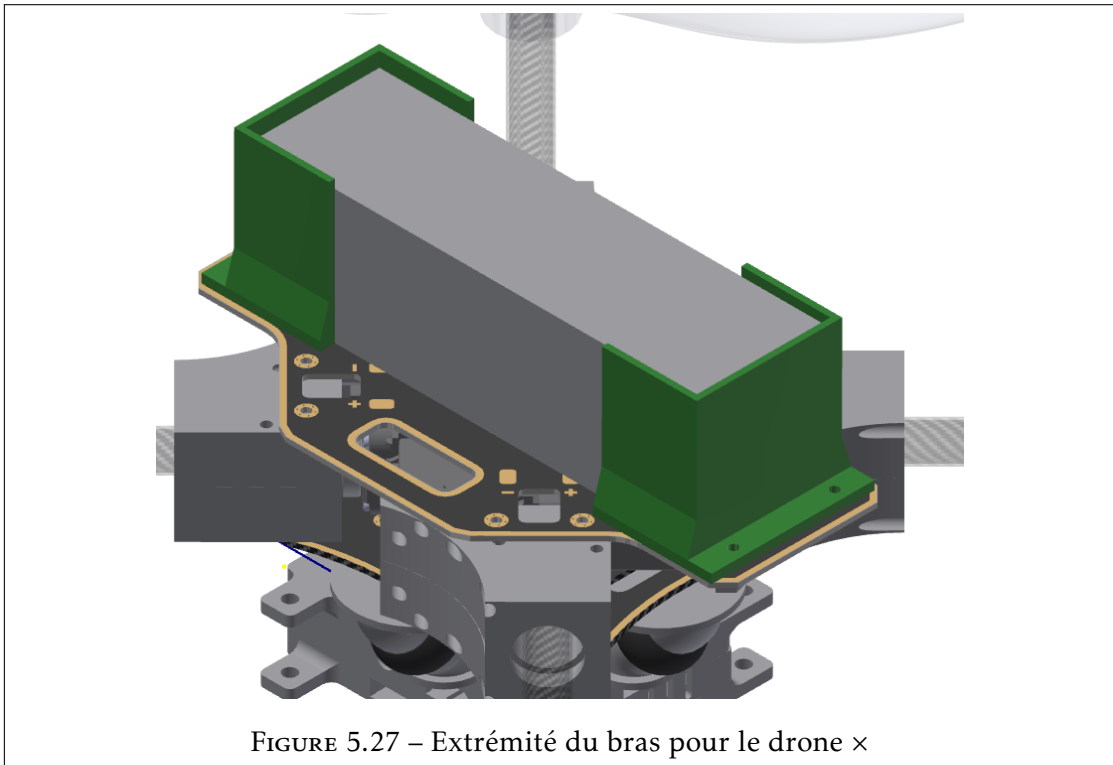


FIGURE 5.27 – Extrémité du bras pour le drone ×

Après ce détail des différents éléments constituant le drone, on propose de présenter les résultats de leur assemblage.

### 5.1.3 Résultats

On réalise une première version des bras pivotant avec les servomoteurs analogiques en première année. L'essai est fait avec les premier servos analogiques. On imprime et assemble les différentes pièces du bras que l'on présente en Figure 5.28. On effectue le test d'inclinaison du bras avec l'hélice en rotation rapide. Le test est concluant, le servo possède le couple suffisant pour changer l'angle de l'hélice. Cette réalisation pratique permet de donner des informations intéressantes sur le système, notamment concernant le temps de réponse, la réactivité ou encore la tenue à long terme.

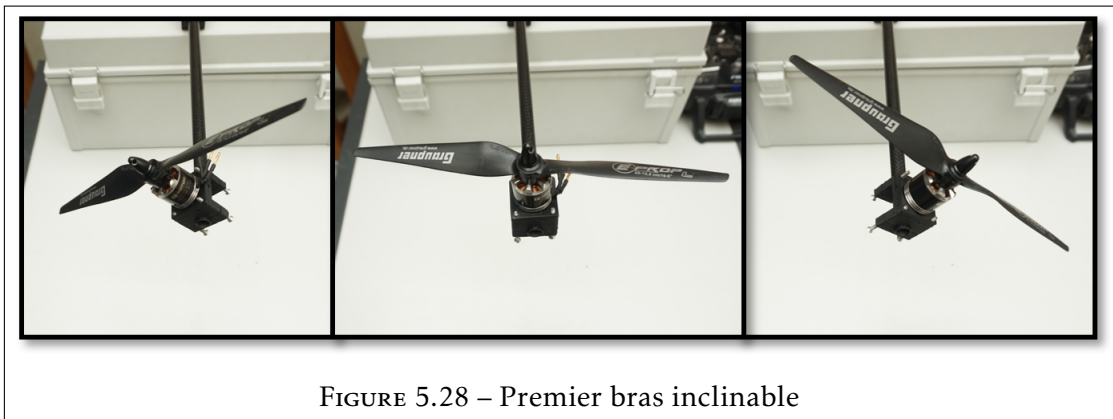


FIGURE 5.28 – Premier bras inclinable

Malgré le fait de pouvoir tenir la rotation des barres, les engrenages des micro servos étaient trop fragiles et cassaient fréquemment. De ce fait, et de par le manque de mobilité angulaire, on choisit les Dynamixel qui sont la solution finale pour ce projet.

Durant la première année, la modélisation CAO pour produire le drone complet a été réalisée. On présente cette modélisation en Figure 5.29. Les ESC sont scotchés aux bras sur cette version. Dans la version suivante on les place directement au sein de la structure. Globalement le prototype réalisé est grand mais pas très lourd avec 2Kg chargé de la batterie.

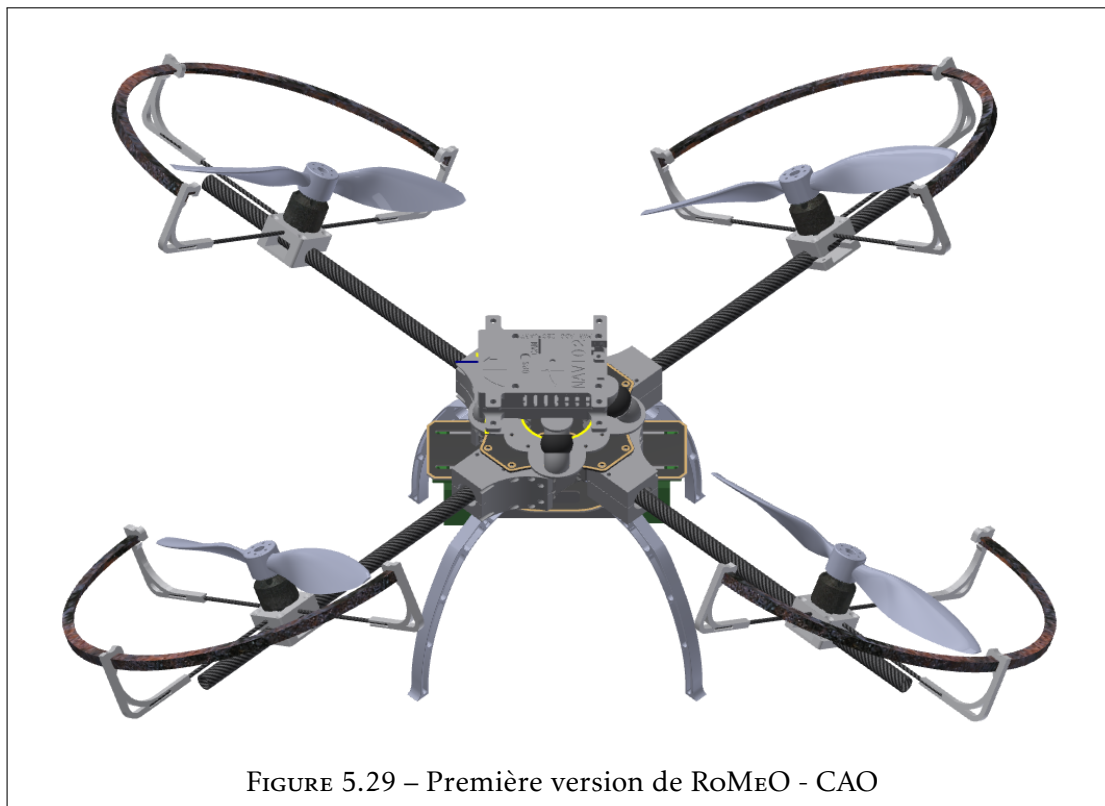


FIGURE 5.29 – Première version de RoMeO - CAO

Après avoir imprimé et assemblé le système, la première version de RoMeO a pris forme. Celle-ci est présentée en Figure 5.30.

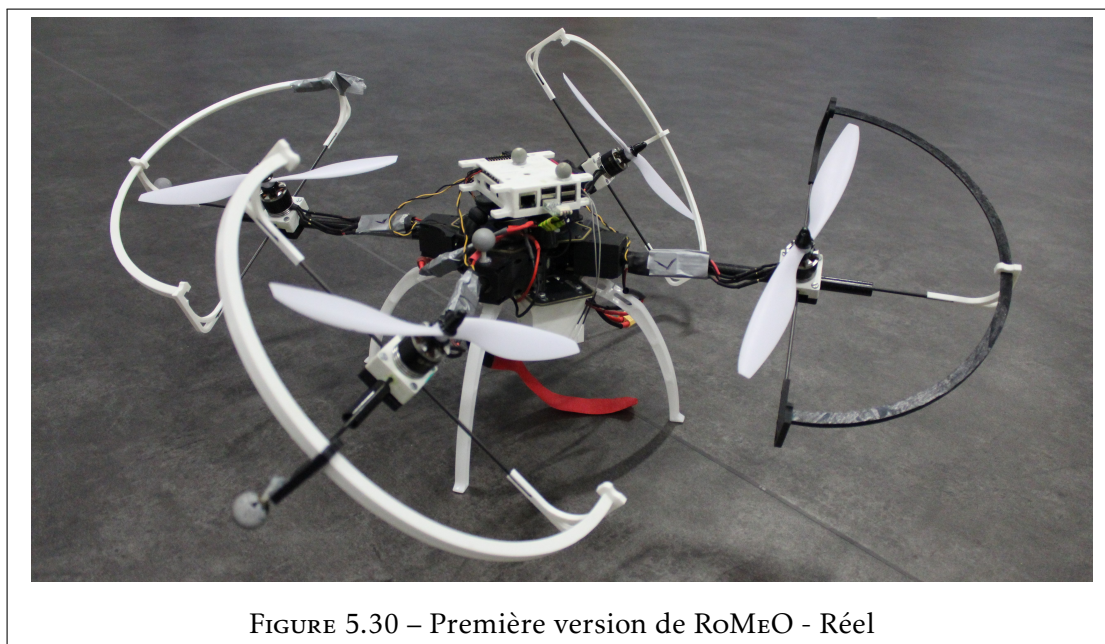


FIGURE 5.30 – Première version de RoMeO - Réel

Quand on assemble numériquement toutes les pièces on obtient pour la structure en  $\tilde{\chi}$  le modèle de la Figure 5.31.

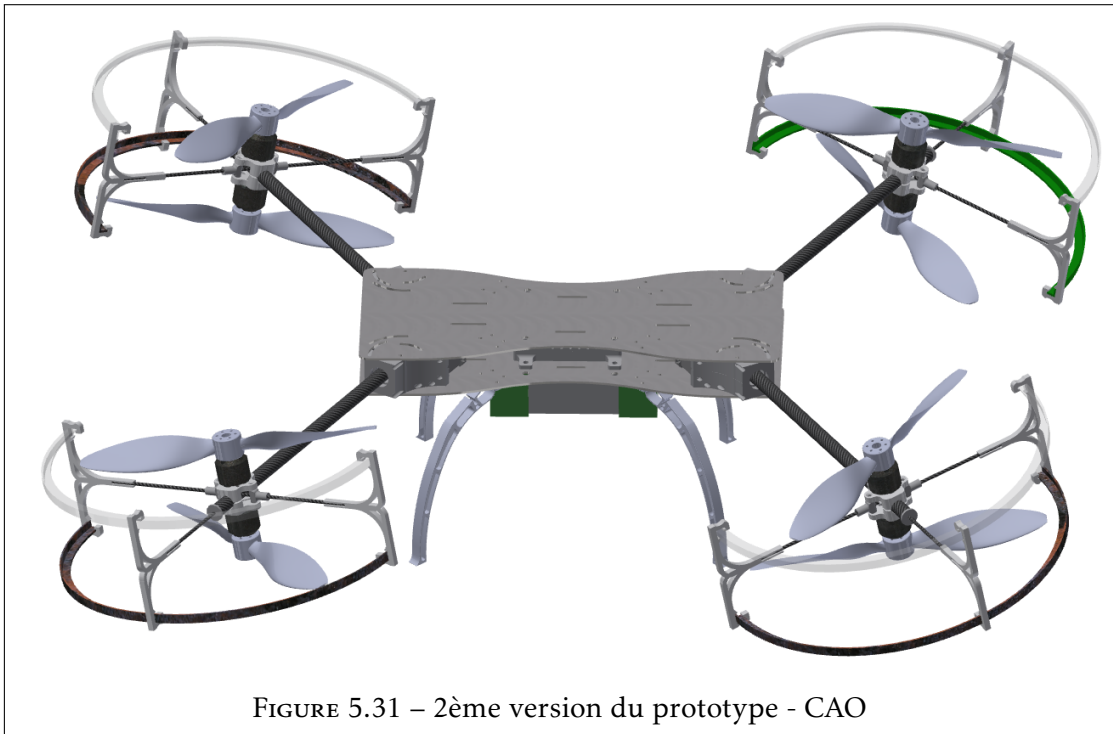


FIGURE 5.31 – 2ème version du prototype - CAO

Une fois imprimé et assemblé on obtient alors la deuxième version du prototype présentée en Figure 5.32.

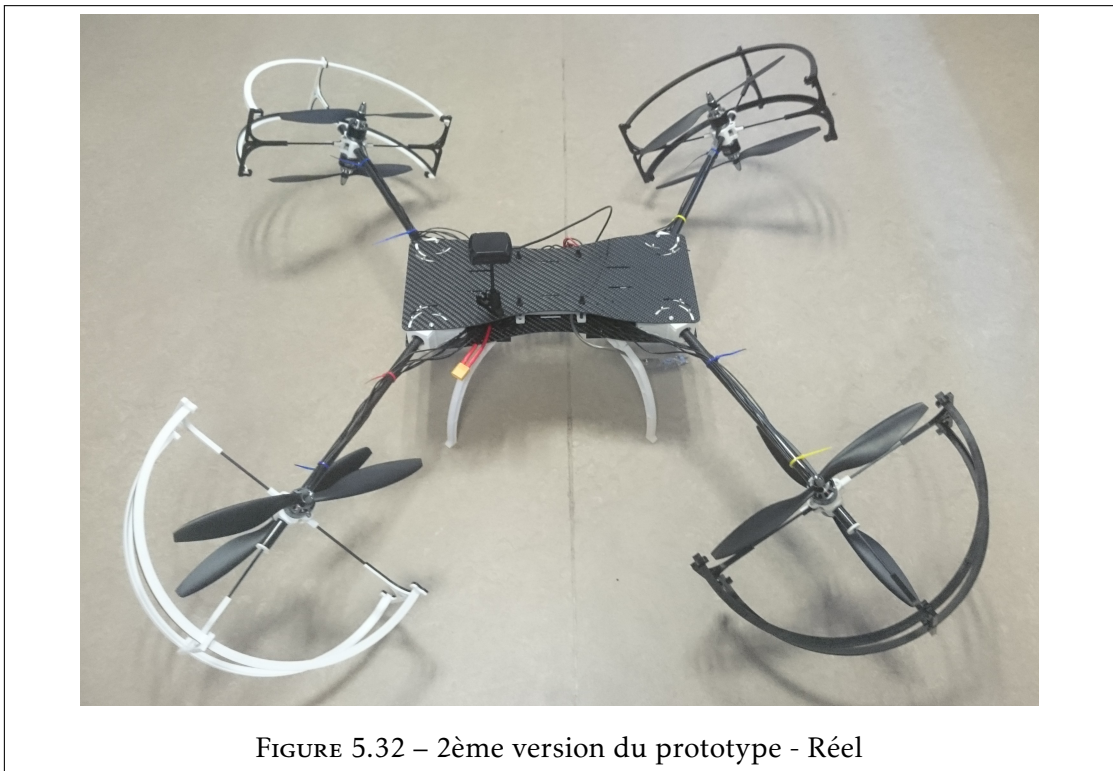


FIGURE 5.32 – 2ème version du prototype - Réel

Le résultat physique est extrêmement fidèle à la version dessinée sur ordinateur. Les essais de prototypages du premier drone ont permis un gain de temps considérable sur la création du deuxième prototype. En effet, la plupart des pièces ont pu être utilisées directement sans, ou avec très peu de changements (comme les porte-servos par exemple).

Un dernier point n'a pas été discuté, c'est celui du choix des moteurs électriques. En aéronautique plus l'hélice est grande meilleur est son rendement. La solution de l'hélicoptère possède donc un rendement supérieur à celle du drone, même si celle-ci n'est pas choisie pour répondre à notre objectif et possède d'autres problématiques qui lui sont propres<sup>6</sup>. On cherche donc à trouver un compromis entre de grandes hélices d'une part mais également un poids à respecter. Plus l'hélice est grande plus les moteurs doivent l'être aussi, mais également les batteries. Du fait que l'on choisit une structure avec deux moteurs l'un au-dessus de l'autre, on peut difficilement prendre des hélices très grandes. Finalement le choix se porte sur une taille raisonnable de 10"4.5 (10 pouces de diamètre et 4.5 pour le pas de l'hélice). Une taille raisonnable permet également de limiter les problèmes en cas de casse, surtout pour un prototype.

La section suivante traite plus en détails la plateforme informatique implémentée sur le drone avec une architecture versatile. On détaille le choix des différents micro-contrôleur et capteurs disponibles. Ce sera également l'occasion de présenter la réalisation du simulateur qui a permis de procéder à une multitude de tests avant de passer aux expérimentations présentées dans le Chapitre 6.

## 5.2 Plateforme informatique

Cette section propose d'explicitier tous les travaux effectués pour le développement du système informatique embarqué et au sol. Dans une première sous-section on présentera les travaux de la première année permettant la création d'un simulateur réaliste.

Principalement développée durant la deuxième année, la deuxième sous-section présentera différentes plateformes permettant l'implémentation des lois de commandes sur le drone. On détaillera trois configurations différentes en explicitant le fonctionnement ainsi que les fonctionnalités qu'elles offrent.

Un choix est fait, permettant une passerelle immédiate entre le modèle utilisé dans le simulateur et celui utilisé lors des tests réels.

Une dernière sous-section mettra en avant les points auxquels il faut prêter attention lorsque l'on développe un simulateur et/ou une plateforme pratique de vol. Suite au retour d'expérience, on synthétisera ici les problèmes et solutions rencontrés ainsi que plusieurs astuces à considérer si l'on souhaite entreprendre une démarche similaire.

### 5.2.1 Simulateur

L'emploi d'un simulateur est évident, avant d'implémenter une loi de commande dans un véhicule il est judicieux de la tester en simulation pour limiter les dégâts. En première année, la principale contrainte consistait à avoir un simulateur pour un modèle développé à partir des éléments de CAO et qui prenait en compte les lois de la physique. On note qu'il aurait également été possible d'utiliser un modèle de double intégrateur sur les 6 DDL pour le drone mais cela aurait eu l'inconvénient d'une trop grande simplification du modèle. Le principe consistait plutôt à tester les lois de commandes sur un système où tous les effets physiques étaient implicitement directement calculés. De nombreux simulateurs existent déjà. Gazebo par exemple est très utilisé dans la communauté et permet de nombreuses possibilités, notamment tester du code utilisé dans une architecture avec un framework *Robot Operating System* (ROS). On s'attardera d'ailleurs sur ROS plus en aval dans le texte. Il est aussi possible d'utiliser le simulateur propriétaire du laboratoire Framework Libre Air (FLAIR) qui propose un simulateur intégré.

---

6. Pour ne pas être exhaustif on peut citer le plateau cyclique qui reste un défi mécanique

Cependant la section précédente montre que l'on a développé un modèle CAO du robot. On choisit donc de se tourner vers une solution pouvant importer les fichiers et permettant une modélisation la plus réaliste possible, incluant la prise en charge des effets physiques d'interaction (couples, forces, etc). Un dernier critère décisif est la capacité à programmer le système graphiquement, donc une approche basée modèle. Ce dernier point représente un gain considérable en terme de respect des exigences fonctionnelles et de maintenance (débugage entre autre).

Le choix s'est alors porté vers l'utilisation de la toolbox Simscape<sup>TM</sup> de Matlab. Celle-ci permet l'importation rapide des fichiers CAO tout en garantissant un calcul fidèle à la réalité des lois physiques appliquées sur le modèle. La programmation est faite sous forme d'assemblage graphique de fonctions-bloc reliées entre elles sous forme de graphe.

Pour réaliser le simulateur, la procédure se fait en deux temps. On suivra d'abord la procédure proposée par Matlab<sup>7</sup> pour ajouter le plugin d'exportation du modèle dans le logiciel de CAO. Puis on importera le fichier généré dans Matlab avec la deuxième procédure également proposée par Matlab<sup>8</sup>. On présente la procédure de réalisation du simulateur en Figure 5.33.

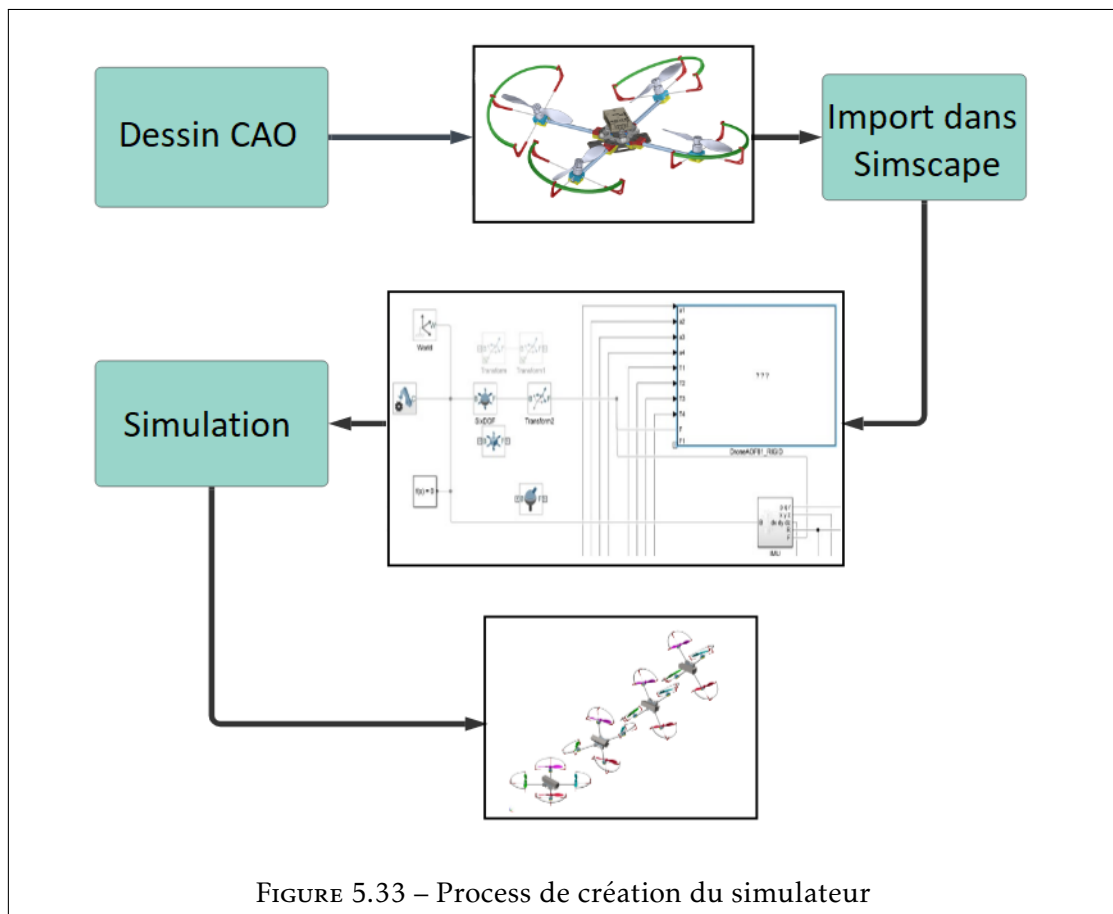


FIGURE 5.33 – Process de création du simulateur

Note : Le fichier généré par Inventor utilise des points au lieu de virgules ce qui empêche l'importation dans Matlab. Il faut simplement remplacer tous les mauvais caractères (en une seule fois ctrl+F et "remplacer tous").

7. <https://fr.mathworks.com/help/physmod/smlink/ug/installing-and-linking-simmechanics-link-software.html>

8. <https://fr.mathworks.com/help/physmod/sm/ref/smimport.html>

Note2 : L'opération génère un fichier Simulink Simscape. Un conseil pour la conception est de créer un ensemble par éléments fixes dans le modèle CAO. En effet, lors de l'importation, le logiciel fera des sous-groupes qui sont fonction des familles de pièces considérées encadrées. Il est donc plus facile de rajouter des liaisons entre ces ensembles, que si toutes les pièces sont indépendantes les unes par rapport aux autres. Dans l'exemple présent on possède 9 groupes (4 bras, 4 moto-propulseurs et un corps).

Note 3 : l'origine des repères des pièces possède son importance car les liaisons entre les ensembles se baseront directement dessus dans Simscape. Il est donc judicieux de placer les repères soit au centre de gravité, soit au centre des liaisons des pièces dessinées.

Une fois le modèle CAO importé dans Simscape, on doit rajouter les différents actionneurs du système. Le principe est d'utiliser un bloc de forces externe pour modéliser la poussée des moteurs. Celui-ci est directement appliquée sur l'axe de l'hélice. Sa rotation est directement contrôlée par une entrée sur le bloc de liaison pivot qu'elle possède avec le moteur. Le principe est le même pour la liaison pivot entre les servomoteurs et leurs bras.

On rajoute, pour plus de fidélité avec la réalité, un bloc simulant le comportement d'un servomoteur. Celui-ci est situé entre la sortie de commande et l'entrée de couple de la liaison entre bras et corps, il sera détaillé par la suite. Les états du système sont mesurés directement par un bloc capteur puis bruités pour plus de réalisme. De même, on sature les entrées des sources de force externes, car en réalité les moteurs n'ont pas la possibilité de produire une force infinie.

On précise également qu'il faut discrétiser la loi de commande du simulateur pour correspondre au modèle utilisé finalement en expérimentations (les données des capteurs sont bien discrètes). Ce sera également un gain de temps quand on passera du simulateur aux expérimentations. Les résultats obtenus à l'aide de ce simulateur sont donnés dans les sections correspondantes dans le Chapitre 6.

On présente ci-après le choix du micro-contrôleur utilisé pour piloter le drone. On met l'accent sur sa compatibilité avec le simulateur présenté précédemment, mais également sur les différentes fonctionnalités que le système offre.

## 5.2.2 Système embarqué

Cette partie a été un point important traité sur la deuxième année. Une architecture particulière a été développée afin de pouvoir tester des algorithmes de manière extrêmement simple et rapide. Après une présentation du matériel utilisé, trois configurations différentes de la plateforme seront détaillées.

### 5.2.2.1 Choix de l'architecture logicielle/matérielle

Pour le prototype on souhaite avoir un système à la fois léger et avec une grande capacité de calcul. De plus, il est préférable d'avoir accès au soutien d'une importante communauté pour pouvoir profiter de l'aide des forums. Plusieurs possibilités se sont alors ouvertes. Il faut faire la distinction entre la partie matérielle, donc le micro-contrôleur à proprement parlé, et la partie informatique.



D'un point de vue matériel on peut citer les solutions suivantes (sans être exhaustif) :

- BeagleBone® Blue
- Navio2 + Raspberry pi
- Pixhawk
- OpenPilot revolution
- Holybro Kakute
- Omnibus
- Mateksys

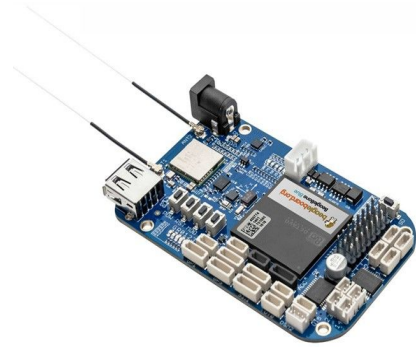


FIGURE 5.34 – BeagleBone® Blue

On présente l'image de la BeagleBone blue® en Figure 5.2.2.1, source <sup>9</sup>.

Concernant la partie informatique, les logiciels d'autopilote Ardupilot et PX4 sont les deux leaders actuels dans le domaine de la robotique. Ayant largement fait leur preuves en étant implémentés sur de nombreux robots, elles offrent un incroyable panel de possibilités d'utilisations. Les milieux d'applications s'étendent de l'espace terrestres à aérien en passant par le naval ou encore sous-marin. Une interface simple d'utilisation offre la possibilité de régler les gains et de planifier des missions de navigation autonome.

Si on combine les deux points évoqués précédemment, une première architecture possible est l'utilisation d'un micro-contrôleur Pixhawk. La Figure 5.35 montre le dispositif dans un drone avec les équipements associés.

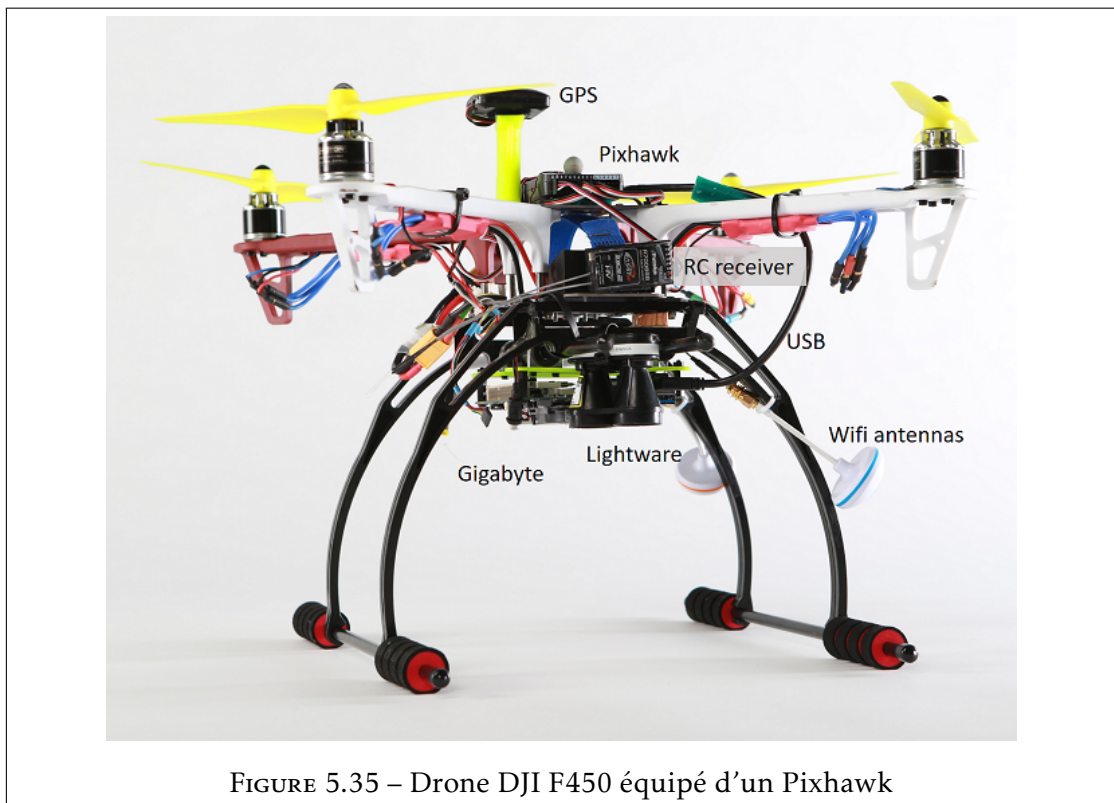


FIGURE 5.35 – Drone DJI F450 équipé d'un Pixhawk

9. <https://www.seedstudio.com/BeagleBone-Blue-p-2809.html>

Ce système ne propose cependant pas nativement de solutions pour la configuration de vol innovante décrite dans cet ouvrage. L'utilisation des logiciels implique donc nécessairement une modification du code source, ce qui demande un temps important ainsi qu'une bonne connaissance du langage informatique utilisé pour développer le logiciel. On remarque que la grande quantité de paramètres et d'options rendent ardue la lecture du code du logiciel et la compréhension du fonctionnement.

Une autre possibilité, consiste à utiliser le logiciel PX4 et à basculer le système en mode "offboard", pour que le micro-contrôleur permette d'accéder directement aux commandes des actionneurs via une interface ROS, plus accessible. Cette solution a été testée sans succès et ce, principalement parce qu'il manque une documentation solide sur le sujet.

Souhaitant avoir un dispositif capable de pouvoir tester rapidement différentes lois de commandes, la recherche s'est portée sur une solution sans codage. Grâce à Matlab Simulink Simscape il est possible de simuler le système mécaniquement (voir section précédente) et d'y tester différentes lois de commande. Cependant, il y avait encore un pas à franchir pour implémenter le code directement dans le micro-contrôleur depuis Simulink sans devoir changer le code de l'auto-pilote à chaque fois. [100] propose également d'utiliser une programmation via Matlab comme exposé ci-après à ceci près que ROS n'est pas utilisé.

On précise qu'il existe encore bien d'autres manières de faire, comme par exemple [101] qui envoie les informations de commandes depuis l'ordinateur via la radio-commande. Des commandes plus haut niveau via ROS sont également présentées dans [102].

Concernant le système physique, la solution retenue offre sensiblement les mêmes options que précédemment évoquées avec le Pixhawk, mais avec une puissance de calcul néanmoins supérieure. Le choix s'est tourné vers l'utilisation d'un Navio 2 + Raspberry Pi (NA+RPI) comme sur la Figure 5.36.



FIGURE 5.36 – Raspberry pi équipé d'un Navio 2

### 5.2.2.2 Première architecture, un hybride

Le système a été développé en trois phases qui se sont suivies chronologiquement. On explicite ici la première architecture. Le principe de celle-ci consiste à utiliser une couche du framework ROS pour commander les actionneurs et récupérer les données des capteurs.

Voici quelques définitions pour donner plus de clarté sur la suite du texte. On remarque que ces définitions ne sont pas académiques ni à prendre au sens propre. Il s'agit d'une re-formulation vulgarisatrice. Plus d'informations sont également directement disponible sur le site de ROS <sup>10</sup>.

La principale notion à appréhender dans ROS est l'emploi des "Nœuds". Un nœud est une entité qui effectue des traitements sur des données. Il pourra alors échanger des informations en envoyant (*publisher*) et recevant (*subscriber*) des données à d'autres nœuds. Cette publication et souscription se fait au travers d'un canal. Ce canal est appelé un "Topic". La publication et la réception des informations se font à une certaine fréquence que l'on définit dans le nœud. On peut également introduire la notion de "paramètre" qui consiste pour simplifier en un message que l'on ne va envoyer qu'une seule fois à chaque modification. C'est un outil très utile pour ne pas surcharger la bande passante si l'on n'a qu'une valeur à changer par moments (exemple : changement des gains). Les informations sont passées au travers de topics en ayant une certaine structure définie appelée "message". Un grand nombre de messages existent et d'autres peuvent être créés. Il faut comprendre un message comme un canevas que l'on va remplir de données. Enfin les services, similaires aux topics, sont destinés aux échanges bidirectionnels entre un client et un serveur. Contrairement aux topics, ils n'envoient pas d'information en continu, mais uniquement lorsque le client s'adresse au serveur. Ils ne seront cependant pas utilisés ici, car le principe consiste à envoyer ou recevoir des informations depuis les capteurs et actionneurs du système, en flux continu.

Une toolbox ROS a été développée par Mathwork<sup>TM</sup> et permet l'utilisation des nœuds ROS directement dans Simulink.

Nativement le NA+RPI propose de fonctionner avec un autopilote Ardupilot. Dans un premier temps car on a cherché à garder cet Ardupilot actif car il propose de nombreuses fonctionnalités intéressantes. Certains messages sont produits directement par l'autopilote et transmis par une passerelle appelée MAVROS. Celle-ci produit un nœud ROS qui sera un "publieur" d'information, notamment sur les données de l'IMU. Après configuration et récupération des messages MAVROS fournis par l'autopilote, on peut afficher les informations des capteurs en temps réel. En revanche, il n'est pas possible de commander directement les sorties PWM pour les actionneurs avec une commande ROS. Il a donc fallu développer un nœud particulier et propriétaire afin de commander les sorties. Cela a été réalisé en réutilisant les bibliothèques livrées avec le Navio et en les intégrant dans un nœud ROS.

On remarque que l'on ne peut pas récupérer d'information directement depuis les capteurs si l'autopilote est activé sur le micro-contrôleur. Il faudra passer par le biais du nœud MAVROS publié par le framework. Utiliser l'Ardupilot est à la fois un avantage et un inconvénient. Un avantage, car on peut récupérer graphiquement les données du micro-contrôleur sur une station au sol (on utilisera Mission Planner qui est très intuitif). C'est également un inconvénient car l'autopilote interfère avec les données envoyées par le nœud ROS publiant les informations vers les actionneurs, y compris quand l'autopilote n'envoie aucune donnée activement.

La solution proposée a consisté à désactiver l'écriture sur les sorties du Navio en changeant le code source de l'Ardupilot. Une fois recompilé, transféré et mis en service, on peut alors utiliser le nœud ROS sans interférence et écrire les commandes nécessaires sur les sorties du Navio. De manière plus graphique on propose d'illustrer le système en Figure 5.37.

10. <http://wiki.ros.org/Documentation>

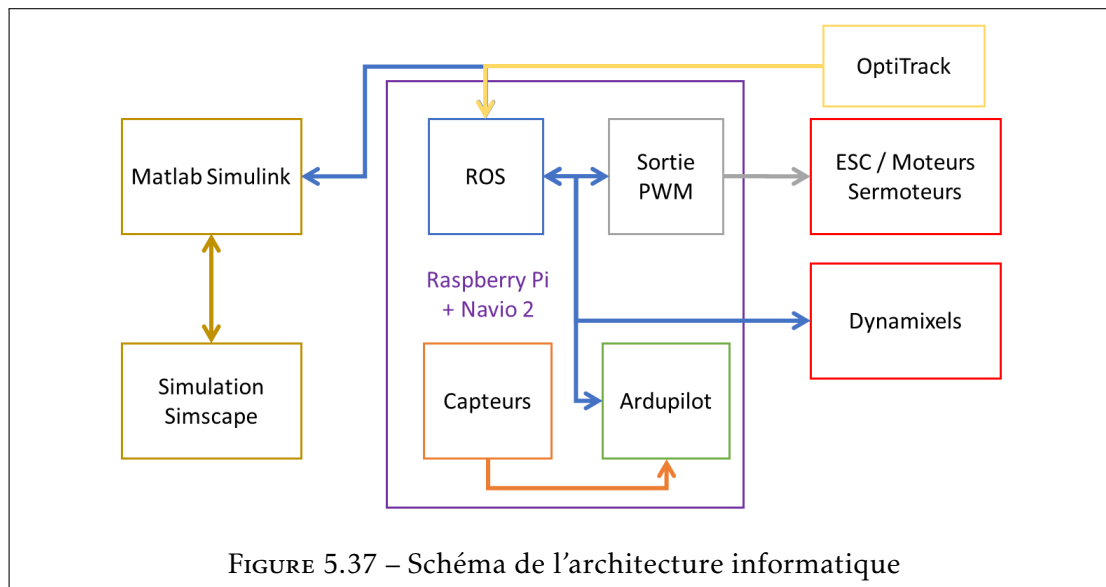


FIGURE 5.37 – Schéma de l'architecture informatique

L'utilisation de ROS permet l'emploi de blocs tiers, comme celui des Dynamixels (servomoteurs numériques) et de l'OptiTrack. Ce dernier est un système de mesure de position et d'orientation d'un objet dans l'espace, qui sera détaillé, plus en amont dans le texte. La possibilité de l'utiliser directement et simplement dans ROS et par extension dans Matlab illustre un autre avantage de cette configuration informatique.

Cette première étape semblait parfaite : l'emploi de l'Ardupilot en parallèle d'un nœud ROS propriétaire. Néanmoins, un problème majeur empêchant le système d'opérer en l'état a vite été constaté. Le nœud MAVROS publiant les données issues des capteurs, avait une vitesse de rafraîchissement beaucoup trop faible. Avec une telle latence sur les données de l'IMU, le système ne pouvait pas voler. Étant donné qu'il n'était pas possible de court-circuiter la réception des données des capteurs avec l'Ardupilot qui tournaient en arrière-plan, il a fallu penser à changer le système. On s'est alors tourné vers une deuxième solution qui a donné des résultats prometteurs.

### 5.2.2.3 Deuxième architecture, ROS sans Ardupilot

La deuxième approche se tourne vers une solution où l'on ne se sert plus du tout de l'Ardupilot. Dans cette solution on choisira de développer un nœud pour chaque entrée ou sortie du système et non plus uniquement pour la sortie des PWM. C'est une solution fastidieuse qui demande un peu de travail de programmation mais les bibliothèques étant déjà fournies par Emlid, ce procédé s'est finalement trouvé très abordable et instructif. D'autre part, il est à préciser que le développement des nœuds est quelque chose à ne faire qu'une seule fois pour toute. On propose de rentrer davantage dans le détail du système ROS utilisé.

On décrit les différents nœuds créés pour permettre l'utilisation de l'architecture. Pour les capteurs (qui publient des informations) on a :

- *Nœuds IMU* : Deux IMU sont présents sur le NA+RPI : le MPU9250 et le LSM9DS1. On crée un nœud pour chacune d'elle, qui donne les données brutes des capteurs, à savoir l'accéléromètre, le gyromètre et le magnétomètre. Avec une fusion de données décrite dans le Chapitre précédent on pourra observer l'orientation du drone. La dérive du cap devra être corrigée (voir Section D.3.4);
- *Nœud GPS* : Donnant des informations concernant la latitude et la longitude ainsi qu'une estimation de l'altitude;



#### 5.2.2.4 Troisième architecture, le retour de l'hybridation

Dans la troisième architecture on décide de mélanger les deux précédentes. On reprend l'idée du début en utilisant l'Ardupilot mais on rajoute dans le code source des *Publisher* et *Subscriber*, qui donne et reçoivent les mêmes informations que dans la section précédente. L'intérêt de cette architecture est le fait de garder une fréquence de publication utilisable pour le vol, tout en bénéficiant de tous les avantages de l'Ardupilot. On citera sans être exhaustif les filtres de Kalman, la gestion des pannes ou encore le repérage du drone sur la carte dans un programme de station sol dédié. Tout en pouvant utiliser toutes ces possibilités, on garde cependant l'avantage de la configuration Simulink avec le fait de choisir sa loi de commande testée au préalable dans le simulateur. Je remercie d'ailleurs Hassan Lhallabi pour sa large contribution lors de son stage. Notre collaboration a permis de réaliser cette dernière architecture. On présente en Figure 5.39 le résultat du vol avec cette architecture sur un drone classique, équipé d'un Navio et d'un Raspberry Pi.



FIGURE 5.39 – Drone racer en vol

L'image montre le vol du drone avec le programme Simulink. Les différentes informations de vols sont également récupérés dans le logiciel Mission Planner simultanément. Ces architectures présentent un fonctionnement très facile mais demandent une certaine configuration de la part de l'utilisateur. On détaille certains points de leur utilisation ainsi que les fonctionnalités qu'elles offrent dans la section suivante. A l'écriture de ce document, d'autres stages sont en cours afin de continuer ces développements. On donne les précisions d'installations et d'utilisation du framework développé, en Annexe D.3.1.

#### 5.2.3 Station sol

Pour le système on développe deux stations au sol différentes. La première est entièrement faite sous simulink et fonctionne en local sur un ordinateur. Elle permet d'envoyer et de recevoir des signaux de manière complètement asynchrone via le wifi ce qui est intéressant pour effectuer des tests. Le fait d'être asynchrone n'est pas un problème car tous les calculs sensibles seront faits directement avec le nœud du programme principal qui fonctionnera en local sur le Raspberry Pi.

Les paramètres et gains de la simulation pourront être changés dans cette station sol. Cependant, le taux de rafraichissement est assez lent et son utilisation n'est pas très fluide. On présente cette station en Figure 5.40.

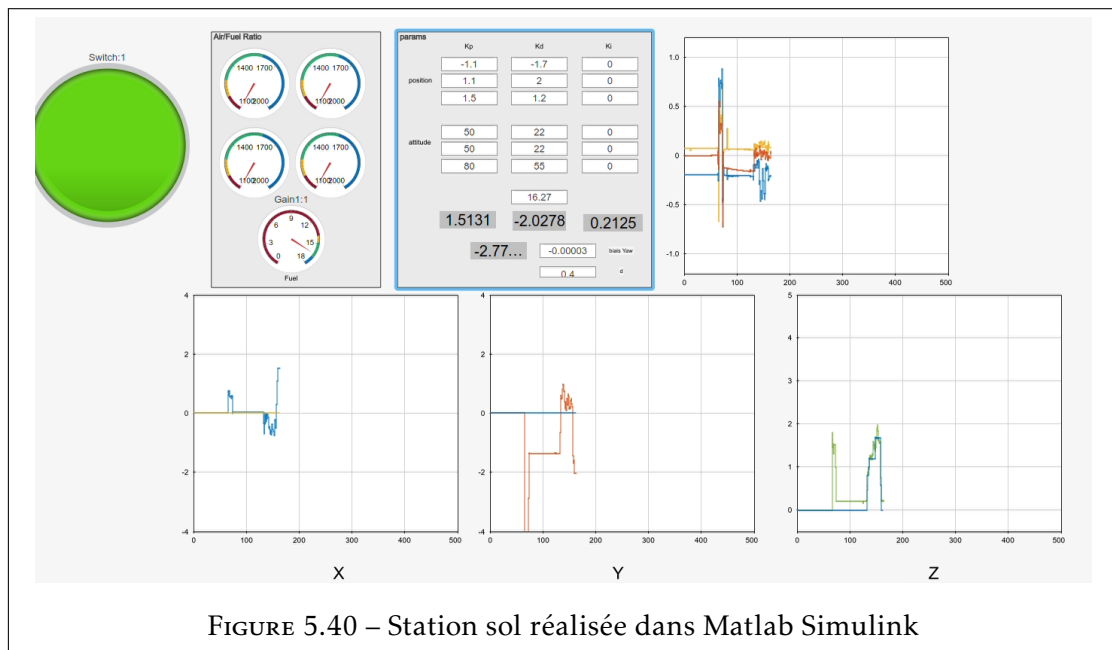


FIGURE 5.40 – Station sol réalisée dans Matlab Simulink

Une deuxième station au sol à été développée. Celle-ci permettant une utilisation graphiquement moins agréable mais beaucoup plus fluide. Il s'agit de l'emploi du plugin *rqt* de ROS. Le principe consiste à avoir un ordinateur qui fonctionne sous Linux avec la distribution ROS installée dessus. Le programme *rqt* permettra d'envoyer un certain nombre d'informations directement vers le système (comme le réglage des gains par exemple). Les données en temps réel seront affichées par *rqt plot* (ou *rqt multiplot*).

L'enregistrement des données est également un point qui a été traité. Pour Simulink, il suffira d'utiliser un block "*to workspace*" et d'enregistrer ensuite la structure de l'espace de travail. Pour l'autre station au sol on se servira du programme *rosbag*.

La commande manuelle du drone se fait directement à l'aide de la radio-commande garantissant une faible latence.

Enfin, cette architecture possède un dernier avantage. Le fait d'utiliser ROS donne accès à de nombreux composants fonctionnant déjà avec le framework. Parmi les exemples que l'on peut donner, les Dynamixels™ et l'OptiTrack™ qui seront compatibles avec le système après une petite configuration. L'utilisation de ces deux modules sera présentée dans les sections suivantes. Remarque : on pourra également interagir entre un Raspberry Pi et un Arduino par un canal série en utilisant la librairie *rosserial\_arduino*.

## 5.2.4 Système d'OptiTrack

Comme cela a été énoncé dans la section précédente, un des avantages d'utiliser une architecture ROS est de pouvoir importer directement des données de l'OptiTrack dans un nœud qui sera exploitable par le système en temps réel. L'utilisation de l'OptiTrack permet de localiser avec précision le drone dans un espace clos (on utilisera l'arène de vol du laboratoire) avant d'utiliser les données GPS en extérieur. Les premiers tests se faisant en intérieur, l'utilisation du GPS du Navio2 est impossible. De plus l'OptiTrack permet d'obtenir une précision très importante limitant les risques de collision, surtout quand le système n'est pas encore robuste. On donne plus d'informations en Annexe D.3.2.

Note : la latence de l'OptiTrack due à la communication et à l'échantillonnage du signal ne sera pas un problème. En effet, l'OptiTrack donne des informations sur la mesure de position linéaire dans l'espace. Le drone possède une instabilité sur son orientation qui est gérée avec un capteur local qui tourne à une fréquence suffisante. N'ayant pas d'instabilité qui pourrait directement être liée à la position linéaire on se contentera de la fréquence qu'impose le système, sans chercher à l'augmenter. Après les premiers tests de fonctionnement, on a souhaité obtenir des informations de position mais aussi de vitesse linéaire pour la loi de commande. On décrit dans le chapitre précédent l'algorithme du filtre de Kalman permettant de filtrer les données de l'OptiTrack pour obtenir la vitesse du drone. On implémentera ce filtre directement dans le modèle à l'aide du bloc Simulink correspondant.

#### 5.2.4.1 Dynamixels et ROS

Comme cela a été décrit dans la section précédente, le choix des servomoteurs pour incliner les bras s'est d'abord tourné vers une solution de développement avant de converger vers l'emploi des Dynamixels. Ces servomoteurs de la marque Robotis™ offrent énormément d'options de fonctionnement. Cependant, leur utilisation nécessite une architecture particulière et plus complexe que le simple emploi des PWM, comme pour un servomoteur classique. On citera ci-après tous les avantages et inconvénients rencontrés lors de leur utilisation. Commençons par les avantages :

- Mesure de courant (et donc de couple)
- Mesure de température
- Mesure de position angulaire
- Précision de positionnement à  $0.29^\circ$
- Connexions en série sur le même canal (et non en parallèle comme pour les PWM)
- Fort couple de sortie
- Possibilité d'utiliser la rotation continue
- Possibilité de les commander en position ou en vitesse
- Rapidité de fonctionnement ( $<0.2s$  pour  $60^\circ$ )
- Grand nombre de raccords mécaniques possibles

Inconvénients :

- Tension de fonctionnement LiPo 3S (11.1V), or notre système est en 4S (14.8V)
- Communication complexe et propriétaire (*Transistor-Transistor logic (TTL)*) nécessitant une interface supplémentaire
- Encombrant et lourd
- Plus onéreux

Le manuel des Dynamixels AX-12A est disponible sur le site [103]. Le principal problème de ces servomoteurs est la communication avec le système. L'interfaçage avec le micro-contrôleur se fait à l'aide du dispositif USB2AX qui permet de faire le lien entre l'USB du Raspberry Pi et un port TTL Dynamixel <sup>11</sup>.

---

11. Pendant la réalisation de cette interface on s'est également intéressé à un contrôle plus bas niveau Arduino qui remplacerait l'USB2AX. Le système étant plus encombrant, cette solution n'est pas retenue.



Finalement le système s'utilisera à l'aide du logiciel ROS Dynamixel workbench (cf. Manuel [104]). Pour plus d'informations, voir l'Annexe D.3.3.

Le nœud créé ne sera cependant pas utilisable simplement dans Matlab (à cause de la mauvaise prise en charge des chaînes de caractère dans le ROS de Matlab). On choisit donc de développer un nœud intermédiaire qui fera le lien entre le Dynamixel Workbench et Simulink. Ce nœud s'appelle "chatter" et est également plus amplement détaillé en Annexe D.3.3.

On choisit également de développer un modèle de servomoteur fidèle à la réalité pour le simulateur. Pour se faire on analyse les réponses en temps réel du servomoteur Dynamixel. On ajuste ensuite les différents gains du système modélisé dans le simulateur. Le modèle du servomoteur est donné sur la Figure 5.41.

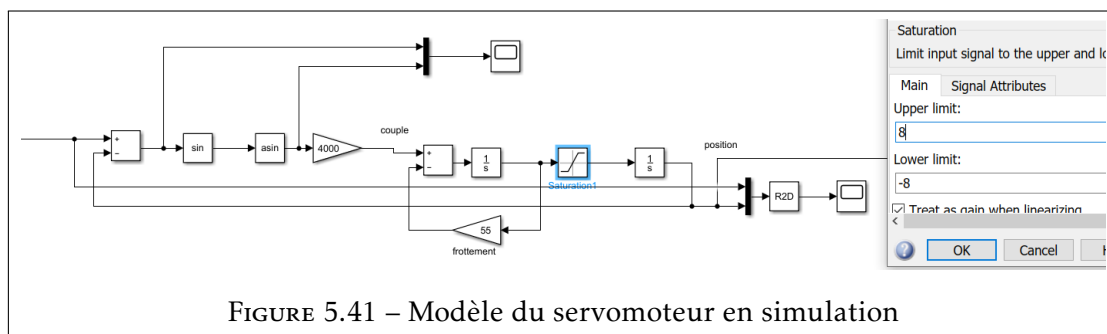


FIGURE 5.41 – Modèle du servomoteur en simulation

Quelques précisions concernant la Figure 5.41. Premièrement le fait de mettre  $\sin(\arcsin(\eta_d))$  permet de rendre la sortie vers le servomoteur continue, ce qui ne serait pas le cas en temps normal de par la fonction  $\text{atan2}$  utilisée pour calculer la consigne d'orientation  $\eta_d$ . D'autre part, le système est très similaire à un double intégrateur excepté que la vitesse du moteur est saturée.

Une comparaison entre la mesure réelle (à gauche) et les valeurs simulées pour une réponse à un signal en créneau est donnée en Figure 5.42. Les deux images sont quasiment identiques validant le modèle de servomoteur proposé.

La mesure fait ressortir que le moteur travaille bien dans son domaine de saturation, ce qui rend l'utilisation d'un prédicteur inutile. Il est impossible de faire tourner un moteur plus vite que sa saturation physique.

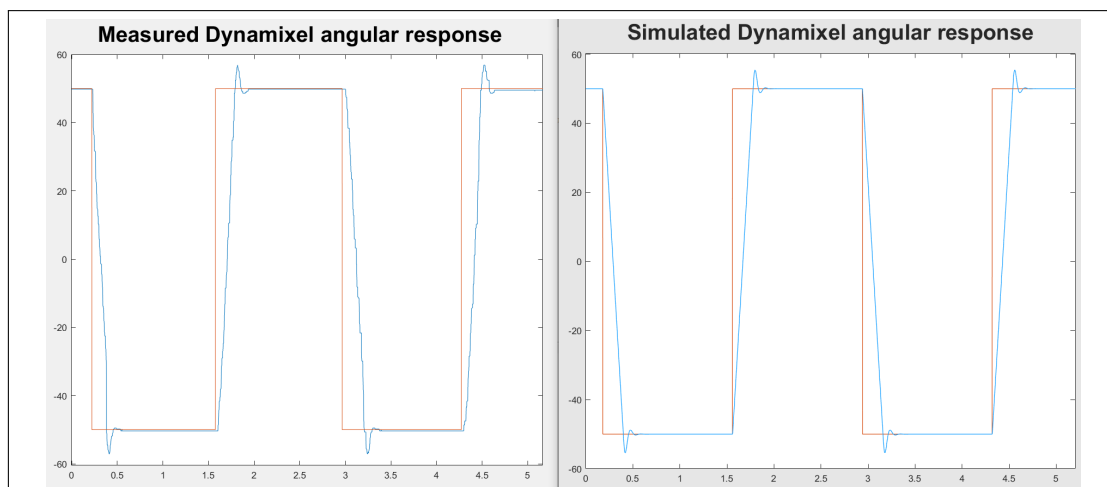


FIGURE 5.42 – Comparaison des réponses des servomoteurs en réel et en simulation

On trouvera en Annexe D.3.4 des aides pour un utilisateur voulant réutiliser le système. Il s'agit de détailler les problèmes qui ont pu survenir aussi bien au niveau de la simulation que pendant les expérimentations. Même si certains points peuvent paraître triviaux puisque la solution est donnée, ils ont parfois pris beaucoup de temps à être résolus.

### 5.2.5 Conclusion

La question concernant l'optimisation des pièces à concevoir pour répondre à l'objectif a été posée. Dans la première partie de chapitre, on y répond en décrivant principalement les pièces présentes sur les bras du drone. Leur forme a beaucoup évolué au cours de la thèse avec des début massifs et surdimensionnés. Les pièces ont petit à petit changés pour aboutir à une optimisation non seulement sur un plan mécanique mais aussi au niveau de la maintenance. L'impression 3D permet un grand pas en avant pour le prototypage rapide et les essais de conceptions. Les matériaux disponibles sont également un grand atout. La fibre de carbone permet de faire des pièces d'une robustesse incroyable pour un poids extrêmement léger optimisant encore le système. Concernant la deuxième partie de ce chapitre, on présente dans un premier temps le simulateur permettant de réaliser des tests validant les résultats obtenus aux trois chapitres précédents. On présente le cheminement de la réflexion faite sur la plateforme informatique. Plusieurs solutions sont évoquées et certaines retenues pour effectuer les expérimentations. Cette discussion permet finalement de mener à une plateforme permettant d'effectuer des tests rapidement et efficacement, pour reprendre la problématique posée. Des astuces et points particuliers exposés en annexe, permettent de compléter l'analyse avec l'espoir d'aider un lecteur voulant s'inspirer des travaux.

Ce chapitre montre toute la réflexion concernant le développement mécanique mais aussi informatique pour le drone que l'on conçoit. Pour cette thèse on possède maintenant un modèle mécanique qui a été linéarisé par un algorithme de commande. Un choix de lois de commande permet de garantir une stabilité du système. Les drones sont conçus, optimisés et finalement produits. La commande est implémentée dans une architecture logicielle performante.

Tous les ingrédients sont réunis pour faire finalement voler ces drones! Le chapitre suivant traite des résultats aussi bien de simulations que d'essais expérimentaux. Une analyse critique y sera faite permettant d'orienter le fil de discussion vers une conclusion générale ainsi que des perspectives pour des travaux futurs.



## Résultats de simulation et essais réels

Dans ce chapitre on propose de montrer les différentes simulations et expérimentations réelles pour les configurations de véhicules décrites dans les chapitres précédents. On présentera le comportement du PVTOL-MI avec une comparaison entre une loi de commande linéaire PD et une approche par saturations emboîtées. Les résultats seront donnés uniquement en simulation, la réalisation pratique ayant été volontairement omise. Bien que tout à fait réalisable, faute de temps, on propose de réaliser directement les essais pratiques sur les deux systèmes 3D retenus (en  $\times$  et en  $\tilde{\lambda}$ ).

La section qui suivra est donc dédiée à la configuration en  $\times$  qui à été la plus utilisée pour les tests aussi bien en simulation qu'en pratique. On présentera les différentes étapes entre la fin de conception du prototype et le vol du drone. La simulation donne une bonne base pour l'ajustement des gains. Celle-ci sera affinée avec des tests sur le trépied décrits dans la section précédente. Un exemple de test sur trépied est montré dans [https://youtu.be/UhT\\_-C3RuVM](https://youtu.be/UhT_-C3RuVM). Après cette étape préliminaire, on peut entamer la phase de vol libre. Ces expérimentations se font de manière progressive, il ne s'agit pas de voler avec toutes les fonctionnalités dès le début. Au lieu de ça on décompose les phases de tests en ajoutant des fonctionnalités à chaque étape. Plusieurs scénarii de vols seront également détaillés pour apprécier et améliorer la dynamique de vol du système.

Enfin, la configuration finale du drone en configuration  $\tilde{\lambda}$  est présentée dans la dernière section. Ici aussi, on présentera les résultats en simulation puis en pratique. Cette nouvelle configuration comprenant les améliorations issues des tests effectués sur le prototype en  $\times$

### 6.1 PVTOL-MI résultats de simulation

Les trois chapitres précédents ont permis de présenter non seulement l'architecture du simulateur mais aussi le modèle et la loi de commande appliquée au drone. Comme on l'a décrit précédemment il n'y aura pas de présentation de résultats pratiques. D'une part ce choix a été fait pour économiser du temps. D'autre part, le système étant plan, il faut prévoir des aménagements particuliers pour qu'il fonctionne en réalité. Ceci étant, [105] propose un prototype qui aurait pu être une base de tests. Enfin, les résultats n'auraient pas forcément été très instructifs car le système ne correspondait pas vraiment à la version finale (notamment au niveau des effets de Coriolis). La Figure 6.1 montre le modèle complet du simulateur utilisé pour tester le PVTOL-MI.

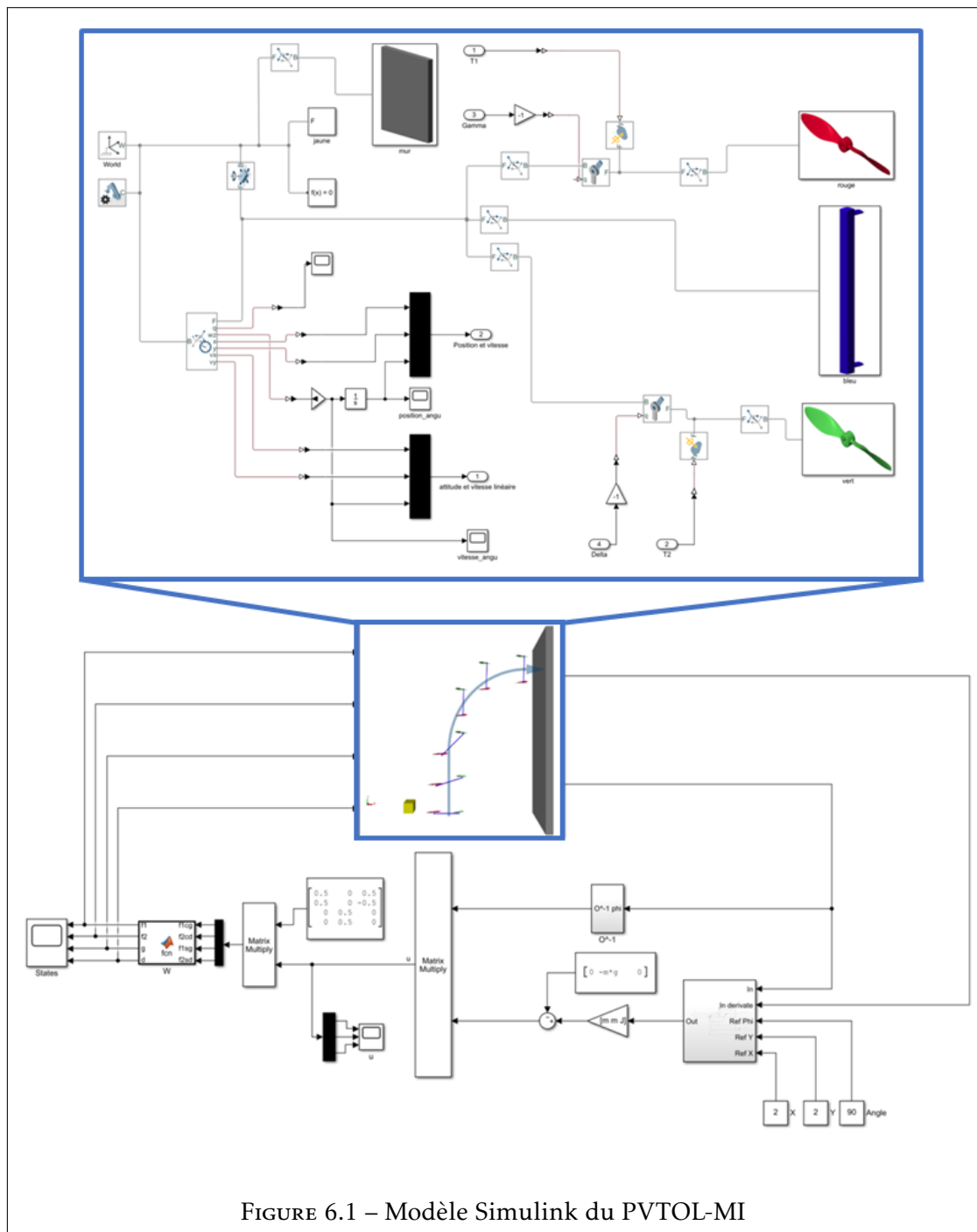


FIGURE 6.1 – Modèle Simulink du PVTOL-MI

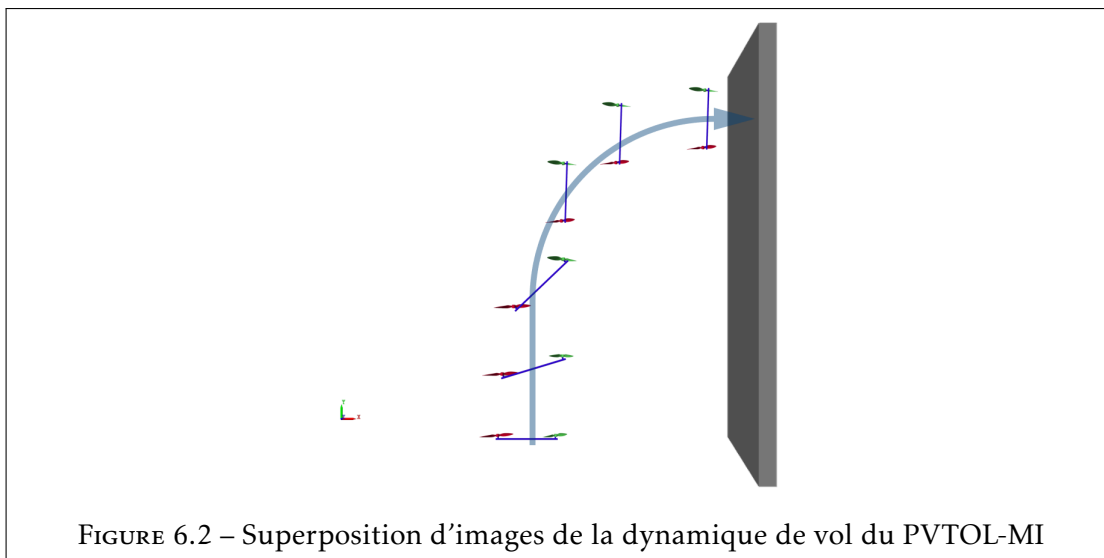
La partie inférieure de la figure montre la loi de commande que l'on applique sur le système. A droite, la référence avec la loi de commande PD. Après compensation de la gravité, on linéarise exactement le système et on extrait les entrées pour les actionneurs depuis les entrées virtuelles.

Le modèle Simulink se lit dans le sens horaire. La partie supérieure de la figure détaille le schéma Simscape représentant le système réel. On remarque bien les 4 entrées correspondant aux actionneurs ainsi que les deux vecteurs  $[1 \times 3]$  de sortie correspondant aux mesures des 3 états, en position et vitesse. Ces mesures sont comparées aux références (pour les positions) puis injectées dans l'algorithme de commande. On multiplie par l'inertie du système et on compense la gravité avant d'appliquer la matrice de rotation  $O^{-1}$ . Après le processus d'allocation on peut ré-injecter les valeurs aux entrées des actionneurs en entrée du système ; ce qui boucle le système.

En appliquant la loi de commande de l'équation (4.15) présentée en Section 4.1 dans le système on obtient un résultat de simulation illustré en Figure 6.2.

Le mur en gris ainsi que le repère inertiel ont été rajoutés pour plus de clarté. L'image décrite montre le drone avec une loi de commande linéaire. Par la suite on va montrer les graphiques correspondants à un comparatif entre les lois de commande linéaires et par saturation emboîtées. La première sous-section correspond à une consigne proche de la position initiale alors que la deuxième correspond à un scénario où la référence est plus éloignée. Sur les graphiques on montre directement les deux lois de commande ainsi que la référence demandée au système.

Après l'exposition brute de ces résultats avec un commentaire sommaire, on va proposer une analyse poussée pour mettre en avant les avantages et inconvénients des deux contrôleurs proposés.



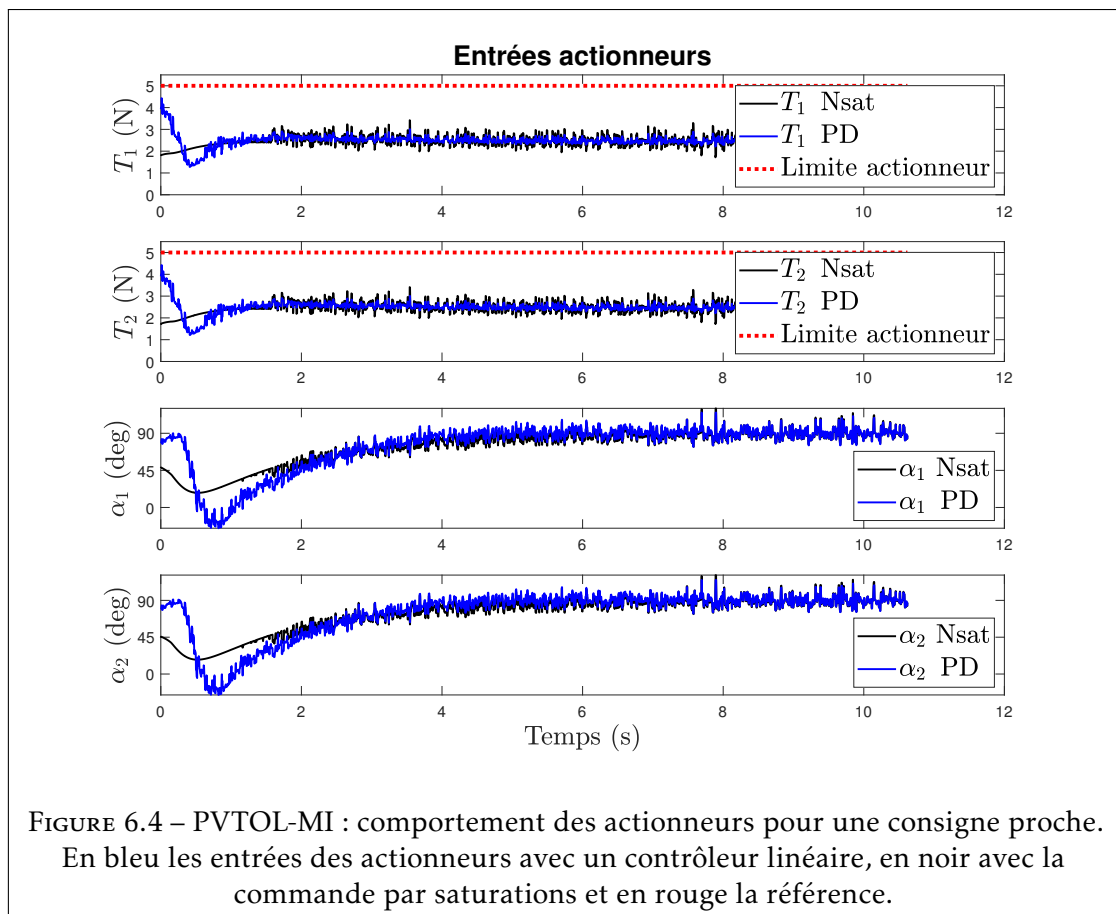
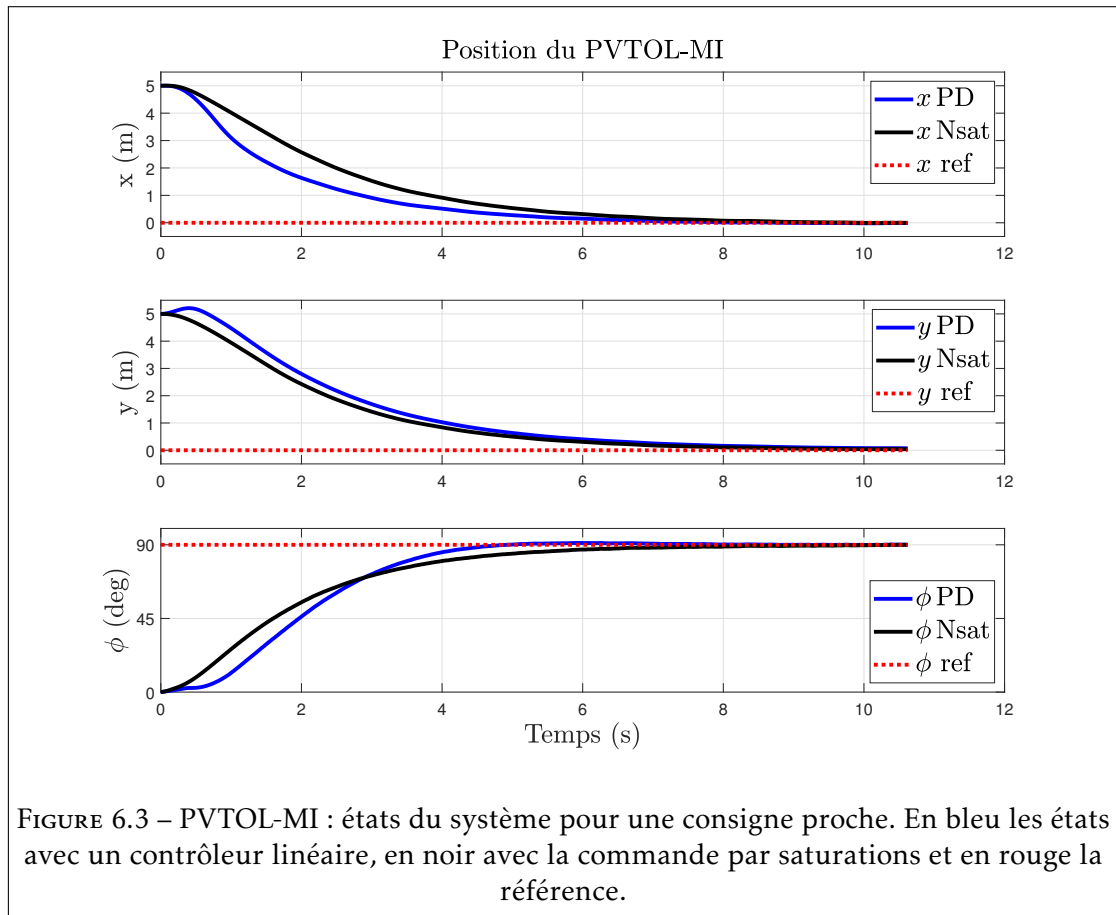
### 6.1.1 Consigne proche de la position initiale

Dans ce scénario la position initiale du véhicule est à 5 m en  $x$  et  $y$ , la consigne étant à 0. Pour les angles on commence à  $0^\circ$  et on souhaite avoir une inclinaison finale de  $90^\circ$  pour l'angle  $\phi$ .

On présente en Figure 6.3 les états du système, avec deux courbes, l'une pour la commande linéaire (PD), en bleu et l'autre pour la non-linéaire (Nsat), en noir.

On voit que le comportement des deux lois de commande est assez similaire. On n'entre pas dans le domaine de saturation de l'algorithme et le système rejoint la consigne assez rapidement et sans problèmes.

En Figure 6.4 on présente les entrées de commandes pour les actionneurs. On précise que l'on rajoute du bruit dans le système. La commande non-linéaire paraît plus douce au début de la simulation mais pour le reste, les deux graphiques sont très semblables.



### 6.1.2 Consigne éloignée de la position initiale

Pour le deuxième scénario on éloigne davantage le véhicule avec une position initiale à 10 m sur  $x$  et  $y$ . On procède à la même opération que précédemment pour  $\phi$ , a savoir un départ à  $0^\circ$  et une arrivée à  $90^\circ$ .

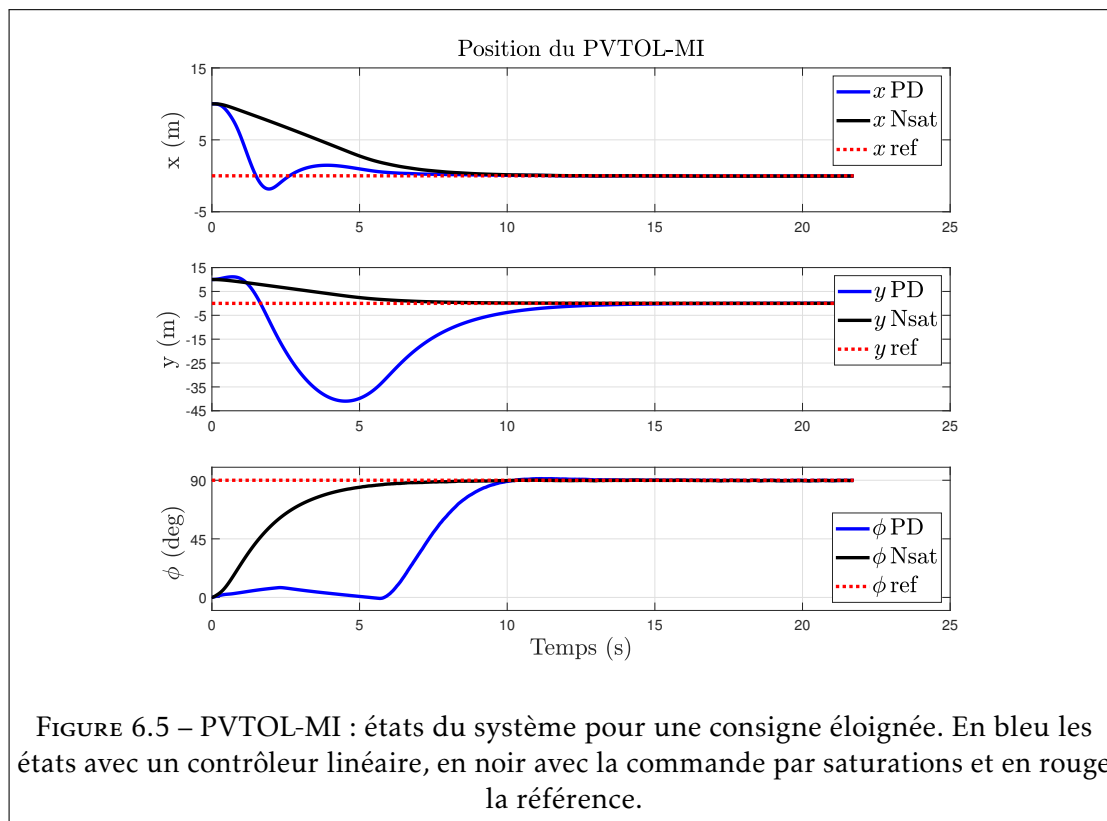


FIGURE 6.5 – PVTOL-MI : états du système pour une consigne éloignée. En bleu les états avec un contrôleur linéaire, en noir avec la commande par saturations et en rouge la référence.

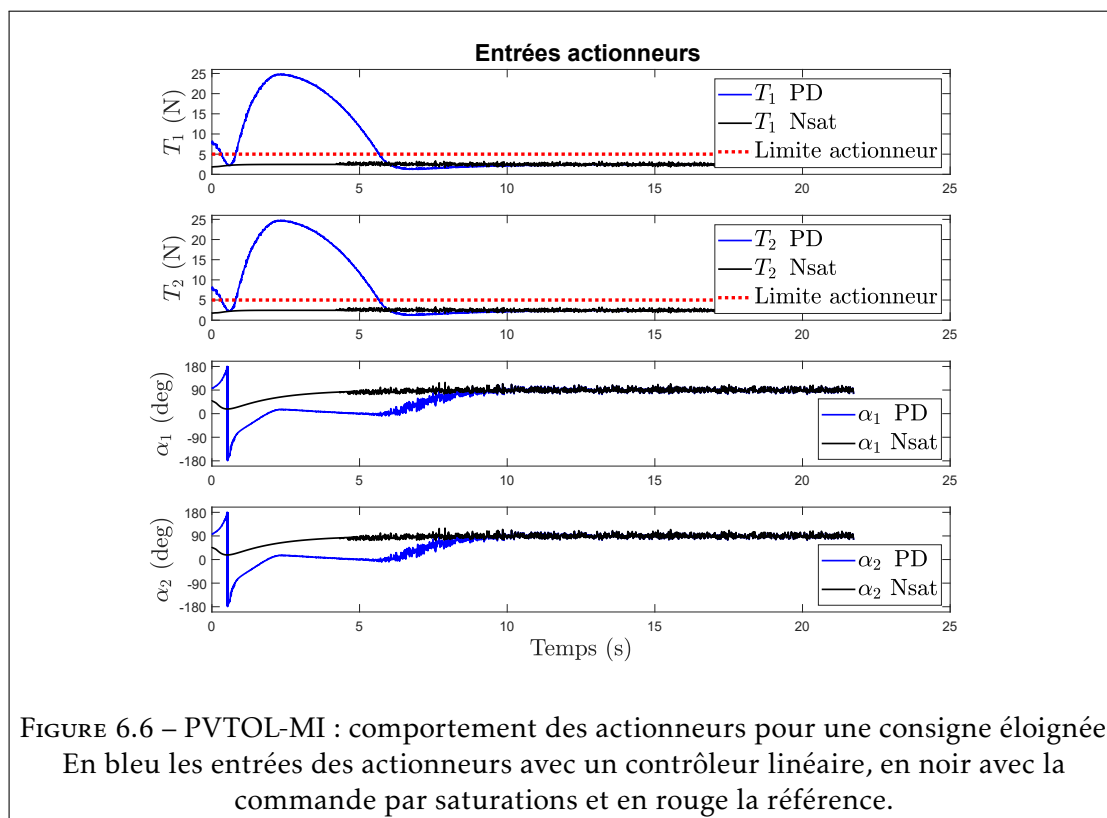


FIGURE 6.6 – PVTOL-MI : comportement des actionneurs pour une consigne éloignée. En bleu les entrées des actionneurs avec un contrôleur linéaire, en noir avec la commande par saturations et en rouge la référence.



### 6.1.3 Discussion

Les graphiques de la sous-section précédentes montrent le comportement du véhicule pour deux scénarii de positions initiales différentes et deux lois de commande différentes. On a montré sur les graphiques précédents le comportement des entrées de commandes et des actionneurs.

On peut remarquer une réelle différence entre les deux contrôleurs. Le contrôleur linéaire est plus rapide pour rejoindre la consigne. L'algorithme de saturations imbriqué quant à lui permet de garantir une approche plus robuste face aux consignes éloignées. Cela à cependant l'inconvénient de ralentir la dynamique du système. On voit très clairement que dans le cas d'une consigne trop éloignée on obtient un dépassement de la commande pour le contrôleur linéaire qui peut être un vrai problème menant dans le pire cas à une destruction du matériel.

Un autre avantage de la commande par saturation est une limitation de l'énergie consommée. Enfin, on peut garantir, en choisissant correctement les saturations, que le système n'atteindra jamais de consignes de poussées hors des limites physiques du système.

On précise aussi que dans cette simulation les couples gyroscopiques ne sont pas pris en comptes car ceux-ci seraient orientés sur un axe qui ne fait pas partie des degrés de liberté du système. On reviendra cependant sur ce point dans la section suivante. On a également négligé l'influence de l'inertie des hélices qui est très petite dans le système.

### 6.1.4 Conclusions

Une architecture particulière du PVTOL à été présentée et expliquée en partant du modèle, en passant par une proposition de loi de commande et enfin en finissant par des essais en simulation. Cette architecture est un préambule pour l'étude d'un drone possédant la capacité d'incliner les bras. Le principe des entrées virtuelles et de la linéarisation par pseudo inverse a permis une appréhension du problème sur un système simplifié, avant de travailler sur un système plus complexe. Ce galop d'essai a également permis d'appréhender les avantages et inconvénients d'une loi de commande du type saturation imbriquée. Le bon fonctionnement en boucle fermée du système a également permis de donner des résultats encourageants pour poursuivre en 3D.

Dans la section suivante, c'est justement les résultats de la configuration en forme de  $\times$  qui seront montrés avec également un comparatif de loi de commande mais aussi une étude sur l'impact des couples gyroscopiques sur le système, problème inhérent à cette configuration particulière, contrairement à un drone classique, à bras fixes.

## 6.2 Quadcopter en X

Dans cette section on présente et analyse les résultats des travaux menés sur le quadcopter à bras inclinables en forme de  $\times$ , principal prototype utilisé lors des tests. On montre dans un premier temps le système inclinable en simulation puis en essais réels. Pour les essais, la progression des essais sera montrée du 1<sup>er</sup> mode de vol au plus récent. Les premiers essais montrent un vol classique mais avec le lacet du drone qui est principalement géré par les bras et non les couples de trainée des hélices.

On complète les expérimentations pour avoir un drone complètement actionné dans le repère du corps. On montre un vol avec une inclinaison désirée statique pour un vol stationnaire. Les vols ont permis de mettre en évidence un certains nombre de facteurs intervenant sur le drone qui seront discutés.

Deux scénarii de vols seront finalement proposés, en simulation puis en réalité. Le premier montrera le comportement du drone quand il doit suivre 4 points formant un carré. Ensuite on changera la consigne pour avoir un drone volant le long d'un cercle, en étant incliné dans le sens opposé à celui qu'un drone classique aurait. Les résultats seront analysés et discutés.

### 6.2.1 Cheminement des essais

La simulation numérique a donné lieu à des résultats intéressants. Sont intérêt est triple. Premièrement elle a permis de tester la loi de commande et de valider son fonctionnement sur un modèle mécanique réaliste. Deuxièmement, les gains ont pu être adaptés et ont constitué une base solide pour les expérimentations. Enfin, la simulation a permis de résoudre des problèmes qui sont survenus pendant les tests ou avant qu'ils n'interviennent. Deux exemples pour ce dernier point.

Pendant les tests on s'est rendu compte que le système divergeait quand il était incliné dans un sens et est sur-amorti dans l'autre. Cela était dû à une inversion d'hélices (les hélices horaires et anti-horaires). La simulation a permis de montrer rapidement cet effet. L'autre exemple concerne la mesure des coefficients d'inertie du modèle CAO fourni par Inventor®. La mesure initiale était fautive et la simulation montrait très clairement un comportement instable. Après recherche, le problème venait du modèle de CAO. Le centre de gravité du modèle n'était pas centré sur l'origine du dessin ce qui biaisait le calcul d'inertie.

Pour les expérimentations, on réalise un cheminement progressif permettant d'aller vers une configuration de vol de plus en plus compliquée. Ce chapitre est également l'occasion de faire un point sur les difficultés rencontrées au cours des expérimentations et les conclusions que l'on a pu en tirer. Comme décrit en introduction la toute première phase est de faire des tests sur trépied permettant d'ajuster les gains du système. Mais en parallèle de ces tests, et avant d'utiliser un système aussi complexe que le drone, on valide le fonctionnement de la plateforme informatique sur un modèle réduit de voiture. Le résultat est donné en sur <https://youtu.be/P9MBziFGBYo>. Sur cette vidéo on montre la voiture qui suit un cercle par la première méthode de suivi de trajectoire décrit dans la Section D.3.4.

Pour les expérimentations avec le drone, on découpe les essais par scénarii de vol comme suit :

1. On commence avec un vol où le roulis, le tangage et la force de poussée sont contrôlés de façon classique. Le lacet en revanche est géré par l'inclinaison des bras comme on le voit clairement sur le premier essais dans la vidéo suivante <https://youtu.be/w7wioNsx3yI>. Après un réglage des gains on obtient un vol qui ressemble à un vol classique, sauf quand on change beaucoup le lacet du drone. Ceci est discuté en Section 6.2.2.
2. Dans un deuxième temps, on choisit de positionner latéralement le drone en permettant l'actionnement sur les axes  $\vec{X}_B$  et  $\vec{Y}_B$ . Cependant on ne vole pas directement avec la configuration finale du drone. La matrice  ${}^I\mathcal{R}_B^{-1}$  de l'équation (4.40) est considéré comme étant l'identité pour les premiers essais. La raison en est que le drone devient très instable en étant complètement actionné si les gains sont mal réglés. L'essai permet donc de rejoindre un point dans l'espace sans permettre de garder une inclinaison fixe. En effet quand on regarde l'équation précédemment cité, la gravité n'est pas projetée sur les axes  $\vec{X}_B$  et  $\vec{Y}_B$ . Ces éléments sont présentés dans la Section 6.2.3.

3. La troisième étape consiste à exploiter la configuration précédente en décrivant un carré. Cela est fait avec la méthode des waypoints décrite dans la Section D.3.4. Le résultat est décrit et analysé dans la Section 6.2.4. Ces travaux ont également permis une deuxième publication [106].

On entre ensuite dans le cœur du sujet de la thèse après ces premiers essais fructueux.

4. On décrit en Section 6.2.5 le vol du drone en maintenant une position fixe tout en le faisant s'incliner sur son axe de roulis et tangage. Les tests décrivent le comportement en simulation puis en expérimentations avec un algorithme PD puis SMCST.
5. Finalement, on souhaite réaliser un essai complexe mêlant une position fixe, avec une inclinaison donnée. Cela permet de simuler le comportement du drone en futur condition réelle. Un scénario de vol complexe est également l'occasion d'optimiser les gains. Un scénario de vol où le drone vole autour d'un cercle est proposé. Cela permet d'effectuer un trajet qui n'est pas évident tout en restant dans les limites physiques qu'offre la volière. Dans cette configuration, une consigne d'inclinaison donnée au drone implique un vol qui serait impossible avec un drone conventionnel. On incline le drone vers la gauche (ou vers -la droite réciproquement) pour aller vers la droite (-gauche). Le cap du drone ne change pas.
6. Une analyse complémentaire sur les couples gyroscopiques a également été faite en simulation et présentée en Section 6.2.8. Des tests expérimentaux ont été menés mais sans succès à ce jour. De plus, la configuration en  $\chi$  n'ayant pas ce problème, l'étude a été abandonnée.

A partir de ce point, la légende sur les figures sera toujours la même. La mesure sera représentée par un trait **noir** fin. Les différentes parties de l'expérimentation sont ensuite surlignées en couleur.

La partie en **bleu** correspond au vol décrit dans la section ci-dessus. Il s'agit donc d'un vol classique avec la particularité que le drone gère le lacet avec l'inclinaison des bras. La partie en **gris** correspond au vol avec l'accès aux 6 DDL. Enfin, les **pointillés rouges** représentent la référence donnée par l'utilisateur.

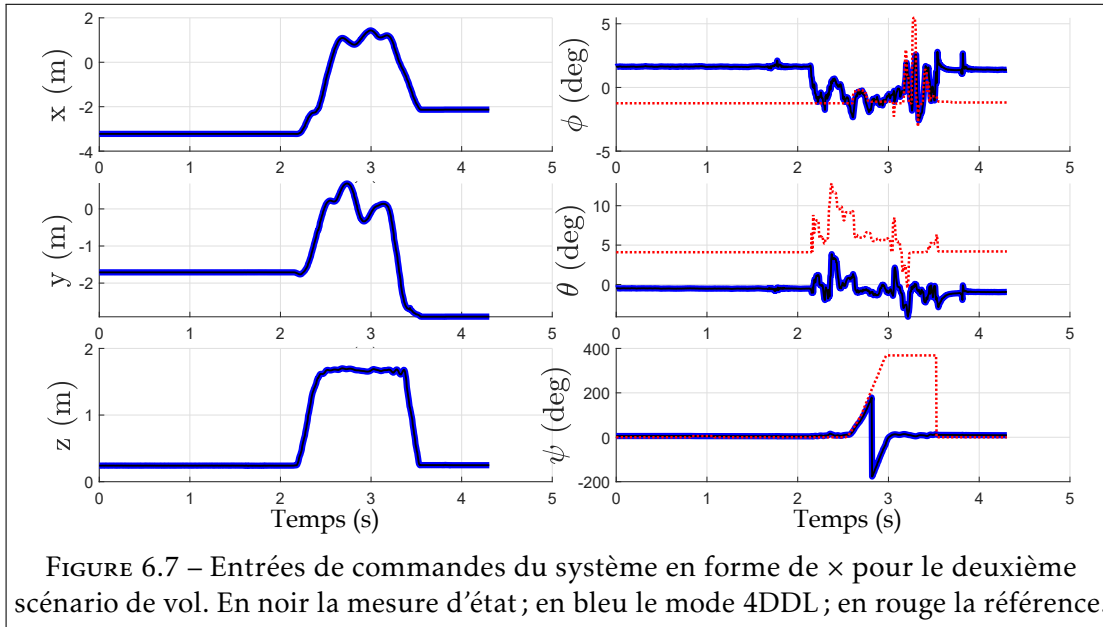
Sur les graphiques on remarque quasi-systématiquement une erreur sur l'échelle de temps. Celle-ci vient du fait que la mesure de temps est faite sur un modèle Simulink correspondant à la station sol. Le pas de temps est donc celui du modèle et non de la réalité. Cette erreur sera corrigée par la suite en utilisant la mesure du temps réel au lieu du temps du modèle.

On peut considérer que l'ensemble des gains pour ce drone comme suit :

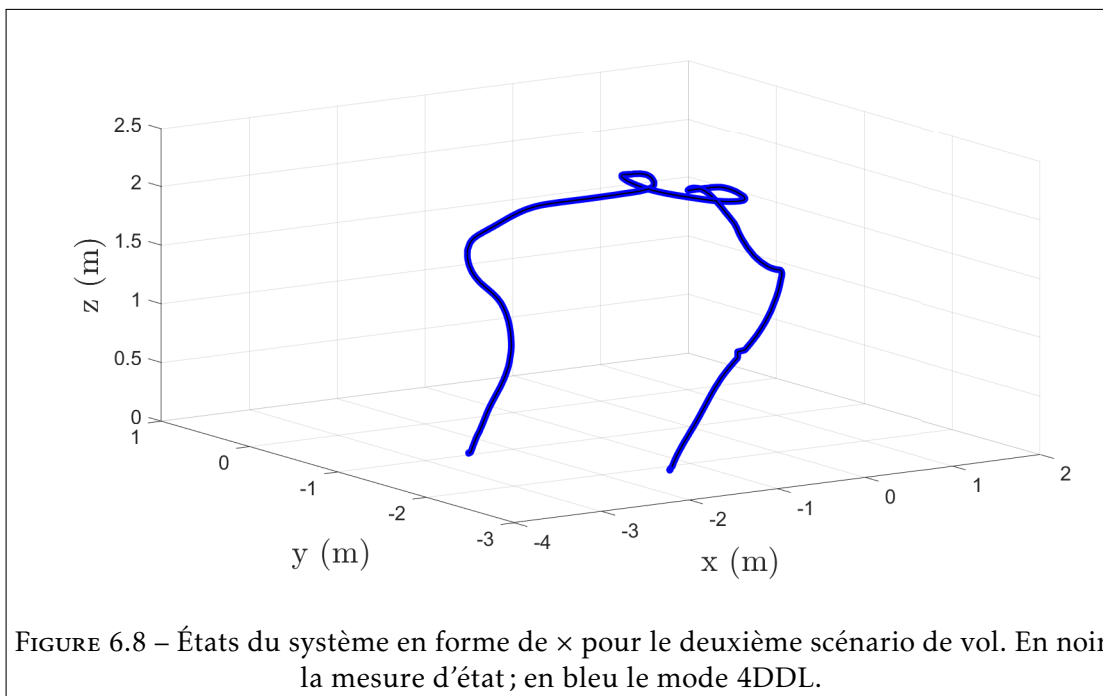
| Expé. | x    | y    | z    | $\phi$ | $\theta$ | $\psi$ |            | $\phi_{SMCST}$ | $\theta_{SMCST}$ | $\psi_{SMCST}$ |
|-------|------|------|------|--------|----------|--------|------------|----------------|------------------|----------------|
| $k_p$ | 0.7  | 0.7  | 0.65 | 5.7    | 5.7      | 7.5    | $\gamma_1$ | 3              | 3                | 2.48           |
| $k_d$ | 1.6  | 1.6  | 1    | 1.66   | 1.66     | 4.66   | $\gamma_2$ | 4              | 4                | 3.54           |
| Simu. | x    | y    | z    | $\phi$ | $\theta$ | $\psi$ |            | $\phi_{SMCST}$ | $\theta_{SMCST}$ | $\psi_{SMCST}$ |
| $k_p$ | 0.85 | 0.85 | 0.85 | 20     | 20       | 6.4    | $\gamma_1$ | 3              | 3                | 2.5            |
| $k_d$ | 1.5  | 1.5  | 1.5  | 10     | 10       | 4.4    | $\gamma_2$ | 10             | 10               | 3.5            |

### 6.2.2 4 DDL bras inclinables

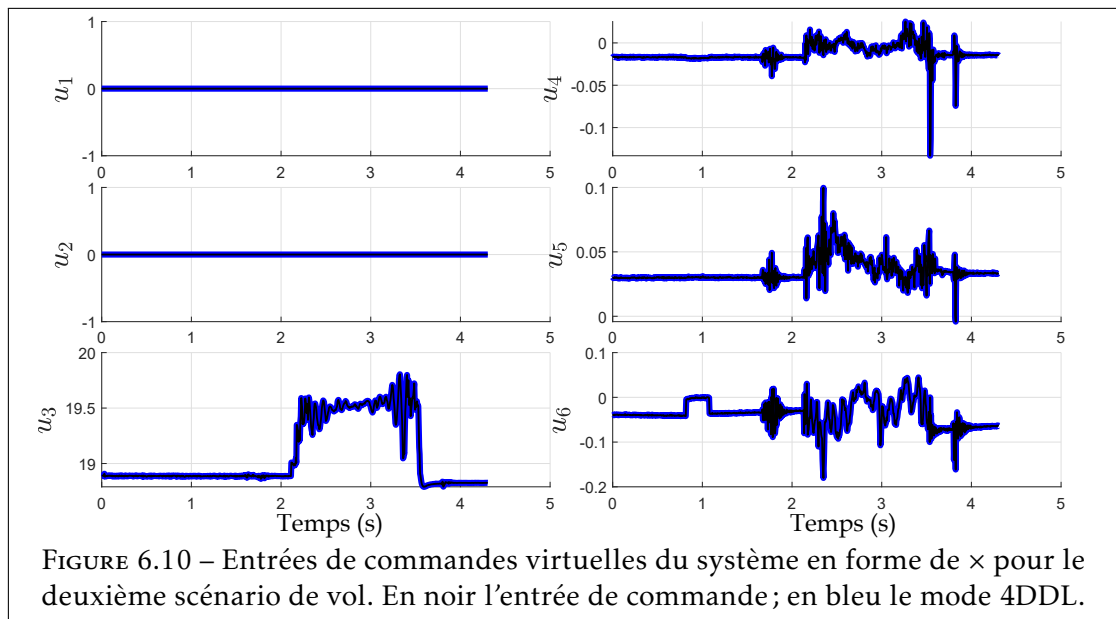
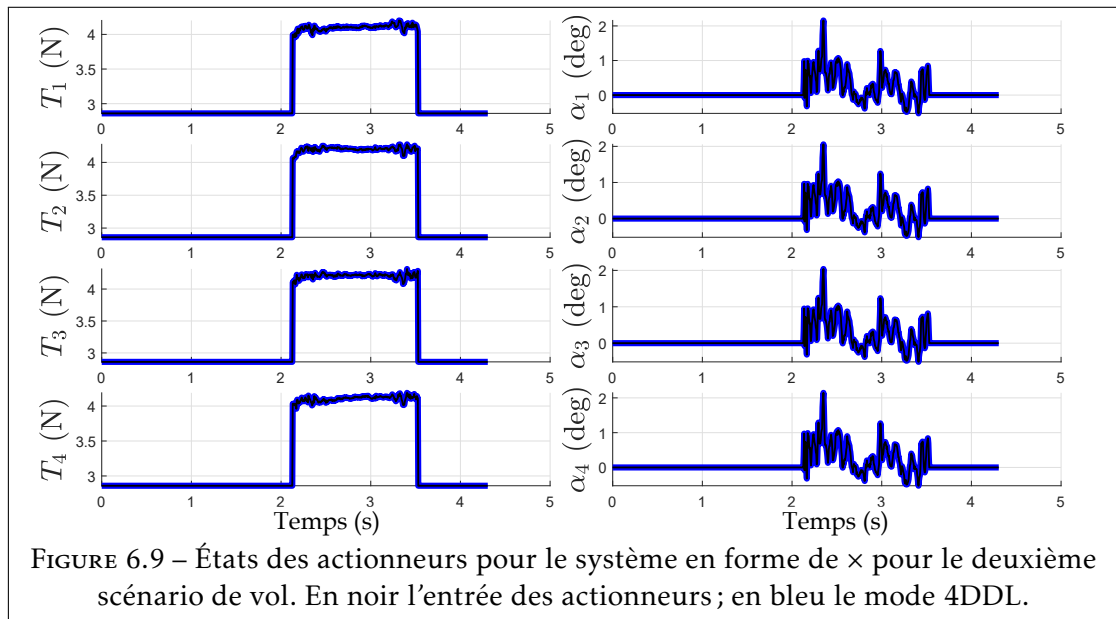
Durant les premiers tests réels, on choisit de désactiver l'actionnement suivant les axes  $\vec{X}$  et  $\vec{Y}$ . Dans l'essai proposé sur les images suivantes on fait tourner le drone sur son axe de lacet pour montrer l'effet des bras inclinables. On rappelle que les trois autres axes sont commandés de la même manière qu'un drone classique. Le vol se fait avec des références en attitude et altitude par contrôle manuel au moyen de la radiocommande.



La Figure 6.7 présente les états du système. On considère bien une rotation sur l'axe  $\vec{z}_B$  de  $360^\circ$ . La mesure est définie entre  $-180^\circ$  et  $+180^\circ$  alors que la référence est l'intégrale de la consigne donnée sur la télécommande. Cette dernière suit donc linéairement une consigne de lacet sans subir la périodicité des fonctions de rotations. On illustre donc bien l'intérêt du calcul d'erreur par matrice de rotations plutôt que par les angles d'Euler.



La Figure 6.8 montre le vol effectué en manuel pour ce drone. Malgré sa rotation sur  $360^\circ$ , la position du drone reste stable dans l'espace montrant un système robuste. La Figure 6.9 présente les entrées de commande pour les actionneurs du système. On remarque que les bras s'inclinent légèrement permettant au drone de tourner sur lui-même. Les gains étant relativement faibles, la consigne sur les angles reste assez petite.



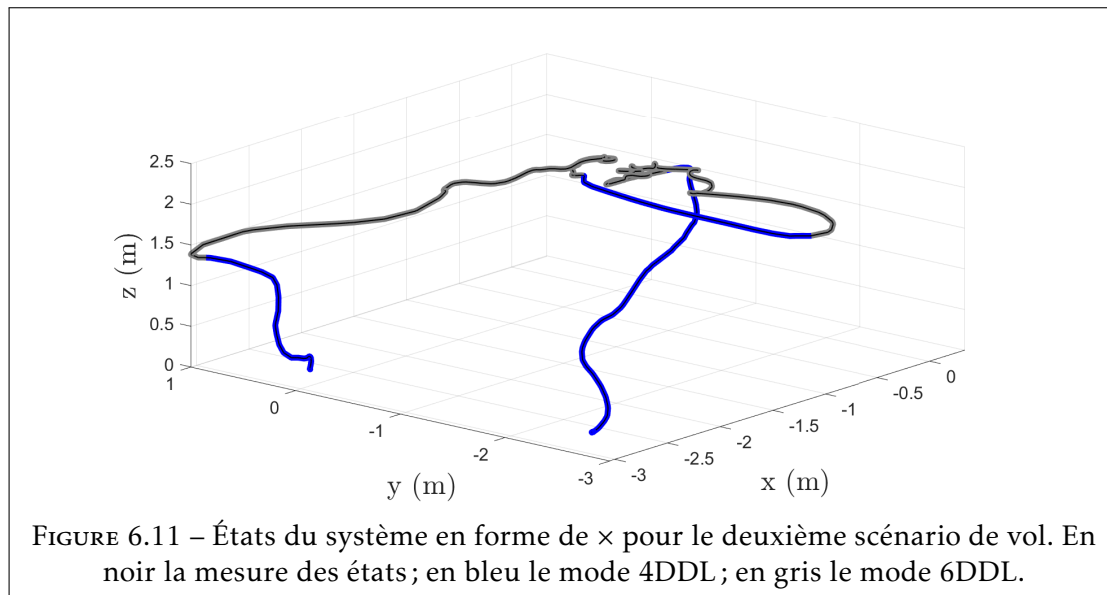
Sur la Figure 6.10 on présente les entrées de commandes du système. Comme énoncé précédemment, on n'applique pas de commande sur les degrés de liberté correspondants aux déplacements linéaires sur les axes  $\vec{X}_B$  et  $\vec{Y}_B$ , qui correspondent aux entrées  $u_1$  et  $u_2$ .

On présente la vidéo <https://youtu.be/2L59Xfg8bxw> d'un vol homologue sur laquelle on peut voir le drone en vol dans la volière.

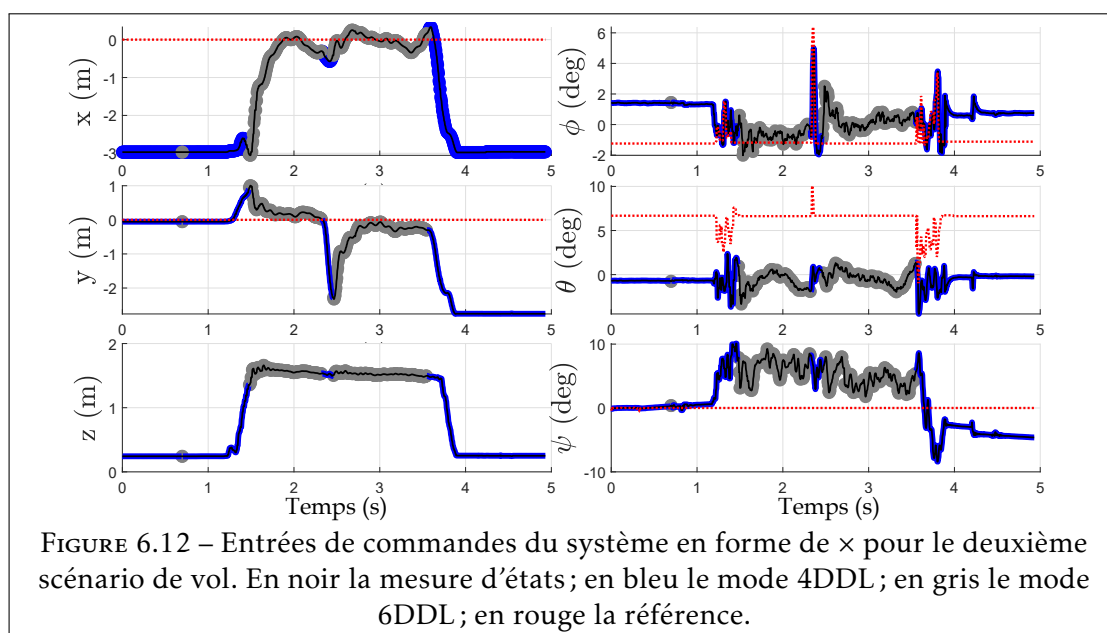
### 6.2.3 6 DDL bras inclinables

On présente dans cette section le comportement du système quand on active l'actionnement du système sur les axes  $\vec{X}$  et  $\vec{Y}$ . Après une première approche en vol manuel (4ddl), on active le mode 6ddl qui permet de ramener le drone à la position désirée. Dans notre cas il s'agit du centre de la volière.

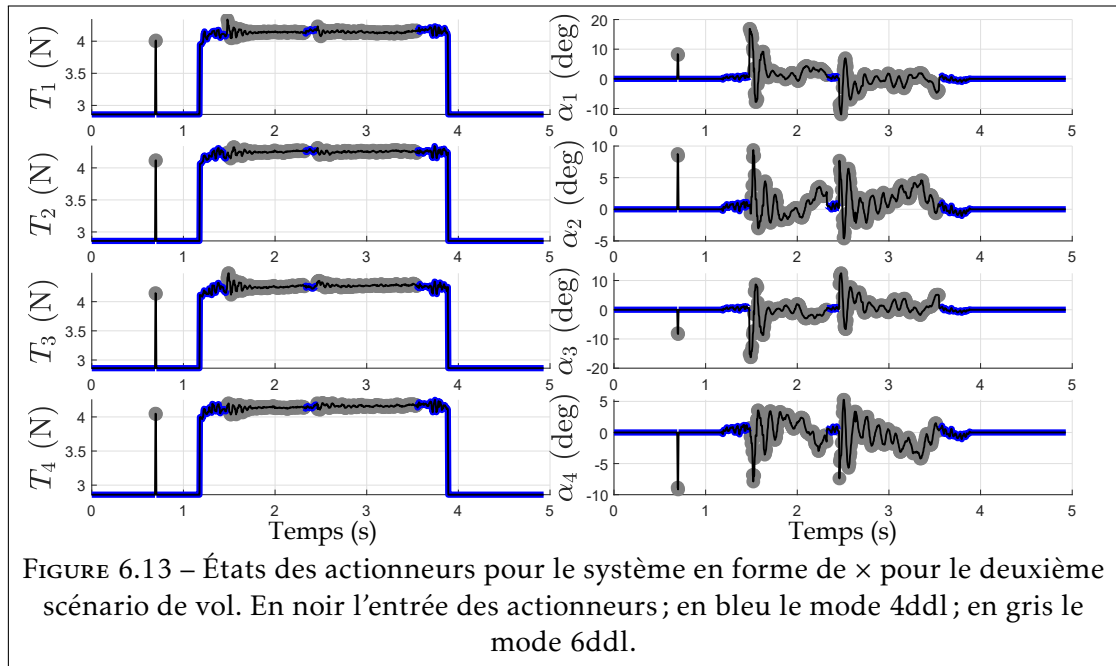
La Figure 6.11 montre le trajet de vol du drone. La position initiale est située à gauche (-3,0) pour finir au centre, en bas en (-3,-3). On distingue clairement qu'après une première phase de vol ramenant le drone au centre on reprend le contrôle en manuel. Ceci permet d'éloigner le drone de son centre pour qu'il puisse re-converger vers sa référence une deuxième fois.



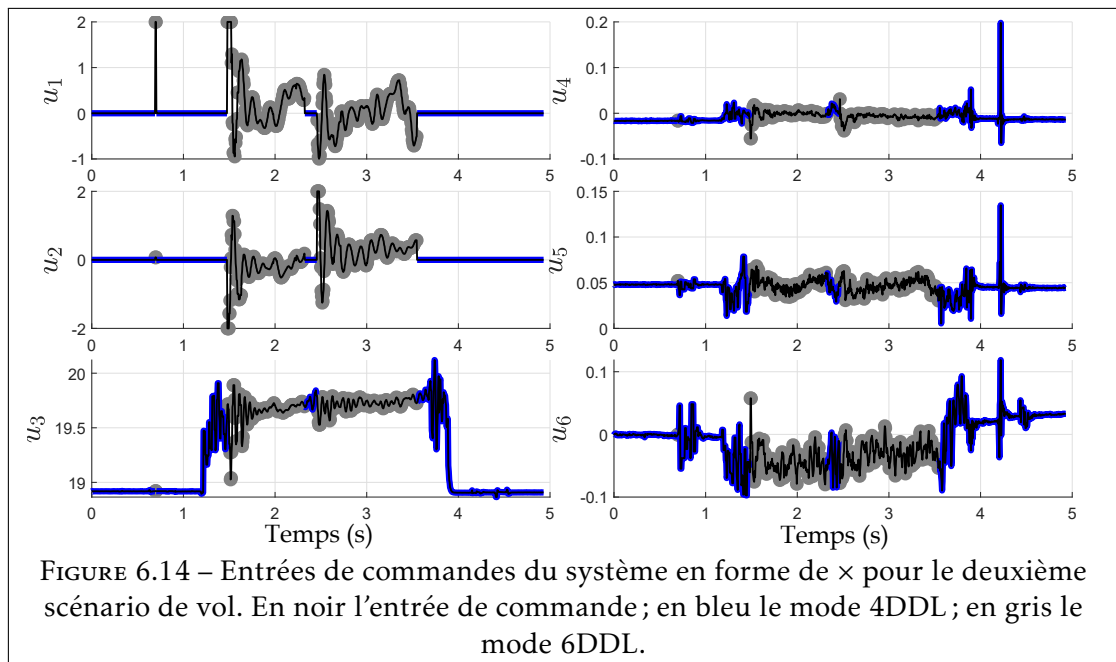
La Figure 6.12 montre les états du système pendant le vol. Le drone converge rapidement vers sa référence sans grandes difficultés. De plus, les consignes sur le roulis et le tangage sont désaxées. Cela vient d'un balourd inhérent à une légère asymétrie du système. La correction est donc faite de manière permanente pour garder un système horizontal.



La Figure 6.13 montre les valeurs des actionneurs pendant le vol. On voit distinctement l'actionnement des bras quand le drone est ramené au centre de la volière, tout en gardant une poussée quasi statique.



Sur la Figure 6.14 on voit clairement que les entrées de commandes sur les axes  $\vec{x}_I$  et  $\vec{y}_I$  sont utilisées. Cela permet d'ajouter que l'on sature leurs valeurs pour les mêmes raisons que celles évoquées pendant l'analyse du PVTOL-MI, à savoir qu'un angle trop important peut causer des problèmes sur le système, en cas d'initialisation trop éloignée.



### 6.2.4 Suivi de carré

Le suivi de trajectoire en forme de carré se fait au moyen de références dans l'espace à atteindre. Une tolérance autour de ces points est dessinée par un cercle rouge et permet une plus grande fluidité lors des essais. On décrit dans un premier temps les résultats de la simulation du système. Cette première étape permet de réaliser et tester les blocs Simulink permettant de réaliser le suivi de trajectoire.

#### 6.2.4.1 Simulation

Une première simulation, qui permet d'effectuer le trajet représenté en Figure 6.15. On remarque que le robot suit correctement la trajectoire désirée, et est arrêté dès qu'il entre dans le cercle rouge. Une vidéo de la simulation est disponible sur [https://youtu.be/wX\\_h0gjCfU0](https://youtu.be/wX_h0gjCfU0)

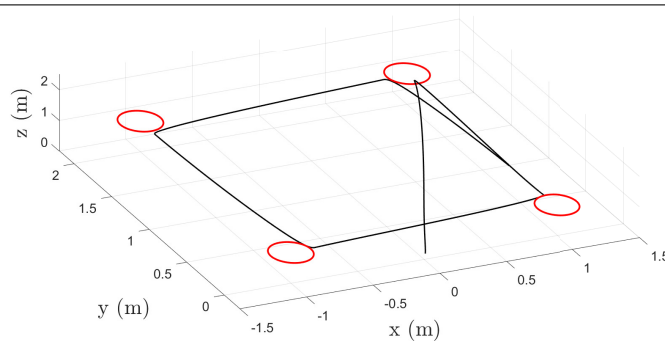


FIGURE 6.15 – Trajectoire du drone en simulation pour un suivi de 4 points de cheminement en forme de carré (3ème scénario de vol). En noir la mesure des états ; en rouge les cercles de références.

Sur la Figure 6.16 on remarque de faibles inclinaisons du corps dès que le drone change de référence de vol. Ce phénomène vient des couples engendrés par la rotation des bras qui fait légèrement basculer le véhicule. Sur les commandes linéaires, le facteur d'amortissement des gains est choisi pour limiter l'effet de dépassement que l'on pourrait avoir. La consigne est ainsi atteinte relativement lentement, mais sûrement.

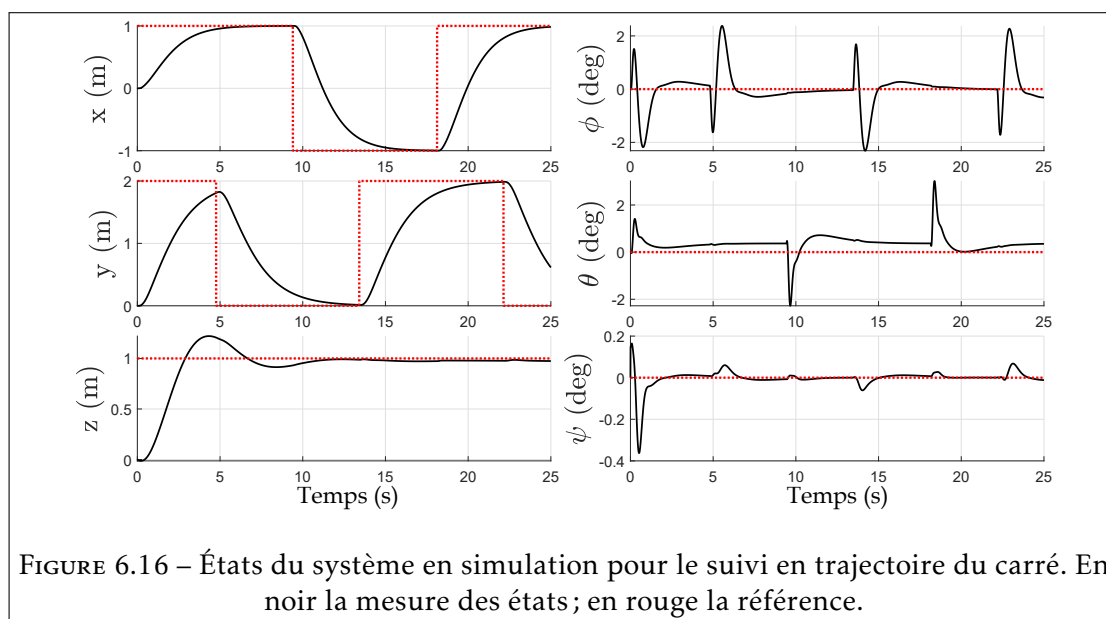
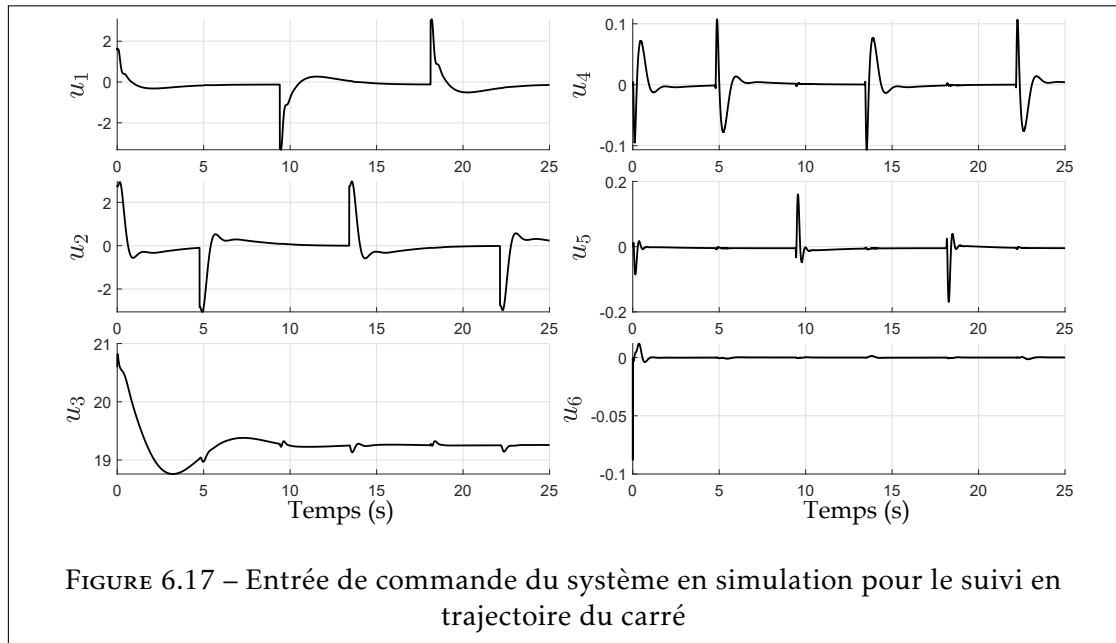


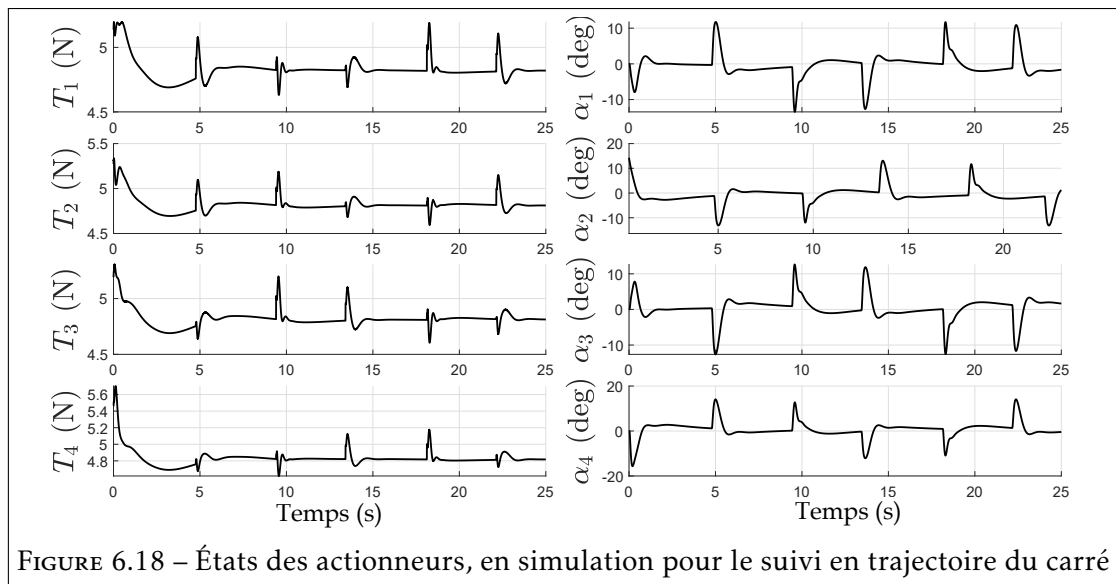
FIGURE 6.16 – États du système en simulation pour le suivi en trajectoire du carré. En noir la mesure des états ; en rouge la référence.



Dès lors que le drone s'incline légèrement, on voit clairement sur la Figure 6.17, que la loi de commande corrige l'orientation du dispositif ( $u_4, u_5$ ).



La Figure 6.18 montre l'inclinaison par à-coups des bras à chaque fois que le drone atteint l'une des consignes désirées.

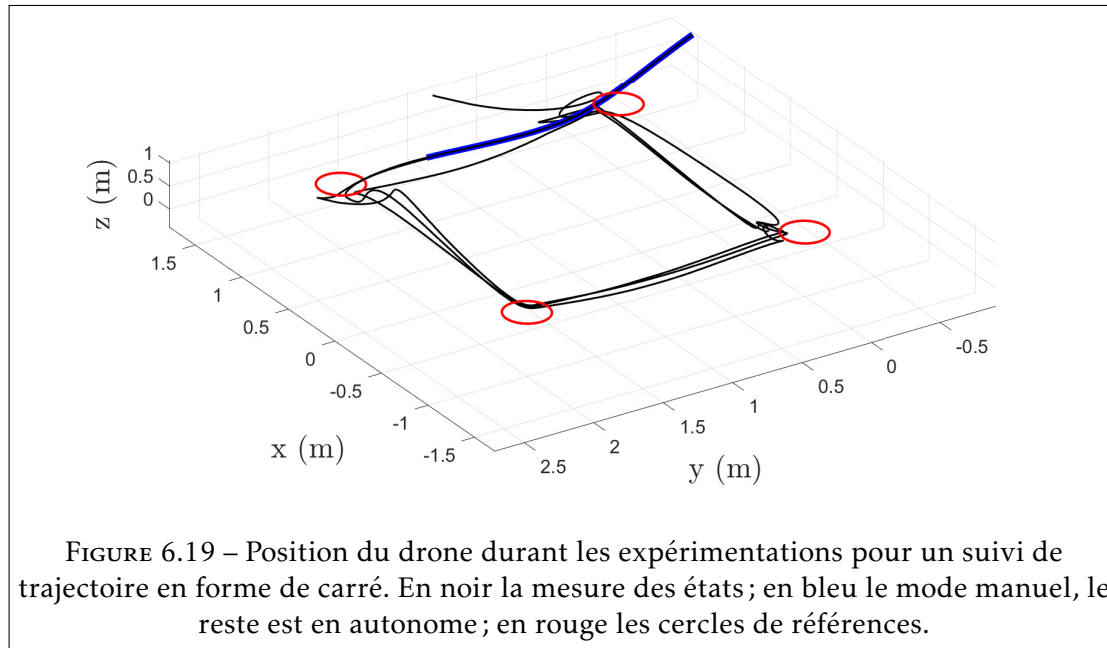


L'ensemble des résultats montre une dynamique rapide faisant apparaître quelques effets mécanique énoncés au Chapitre 3, comme le couple associé à la rotation des bras ou encore les couples gyroscopiques.

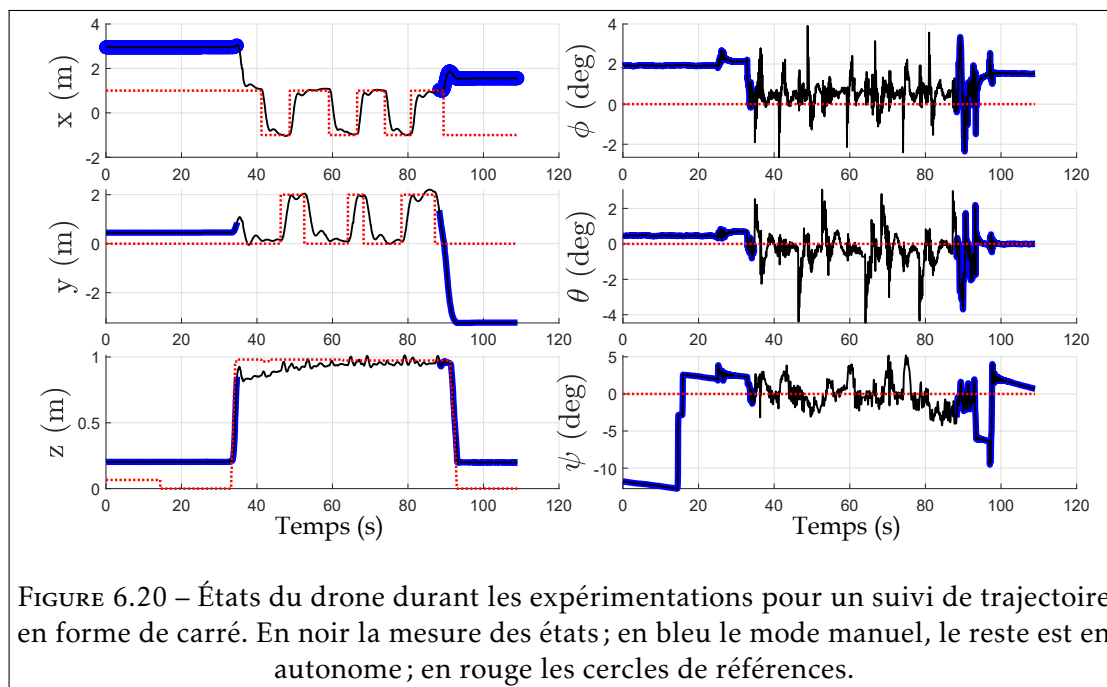
On remarque également que le système n'a pas été bruité, contrairement au système réel qui suit. Le bruit des mesures a un effet important sur le bon positionnement du drone ainsi que sur les différents effets mécaniques qui peuvent en être amplifiés.

### 6.2.4.2 Expérimentations

On effectue finalement les expérimentations, une vidéo dans ce scénario est proposée sur <https://youtu.be/zyVreEYNrys>. La Figure 6.19 décrit le trajet réel qu'effectue le drone. On remarque tout de suite une petite vague entre le point à gauche et celui en bas. Cet effet provient d'une part de l'hypothèse faite sur la matrice de rotation décrit en début de section ( ${}^I\mathcal{R}_B^{-1} = \mathcal{I}$ ). D'autre part l'effet est lié aux couples résiduels (gyroscope et couple de rotation des bras) comme décrits en simulation.



Sur la Figure 6.20, on constate que le système a un comportement très homologué à celui décrit en simulation. Outre le fait que cela renforce la confiance sur le simulateur conçu, les essais montrent une bonne stabilité de vol. On retrouve également les effets d'inclinaison du drone décrit précédemment sur le roulis et le tangage de l'appareil.



On ne détaille pas les entrées de commande car il n’y a pas de point remarquable dessus. En revanche, sur la Figure 6.21 on présente les états des actionneurs et notamment le comportement des bras qui s’inclinent d’une manière très similaire aux tests effectués en simulation (mais bruités).

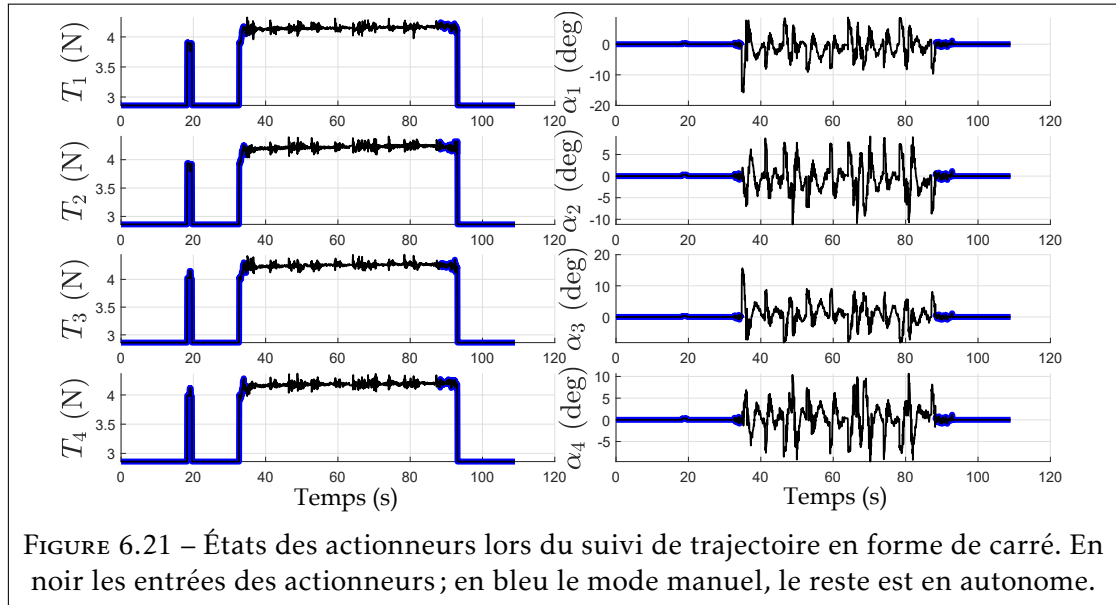


FIGURE 6.21 – États des actionneurs lors du suivi de trajectoire en forme de carré. En noir les entrées des actionneurs; en bleu le mode manuel, le reste est en autonome.

Un point important peut être souligné pour cette expérimentation. Les déplacements sur les axes  $\vec{x}_I$  et  $\vec{y}_I$  du drone doivent correspondre à ceux de l’OptiTrack re-projetés dans le repère du véhicule. Pour se faire, on choisit tout simplement d’aligner le cap du drone (donc un lacet à  $0^\circ$ ) sur l’orientation de l’OptiTrack. En cas de non-respect de ce point, le drone essaiera d’aller en avant alors qu’il ira en diagonale en réalité ce qui fera faire un cercle au drone pour une référence de position et le fera dévier de la trajectoire dans l’expérience décrite ici. De plus, certaines difficultés ont été rencontrées de par l’inertie du drone. Si celui-ci a trop dévié de sa trajectoire (par l’effet cité précédemment par exemple), le drone dépassera largement la consigne avant de re-converger vers sa référence. Une première solution consiste à régler les gains (ce qui a été fait dans le cas présent). Une deuxième mesure prise est l’utilisation du système de trajectoire basé sur le temps et non la position. Et finalement, il est intéressant d’utiliser la commande par mode glissant également sur la commande en position du drone.

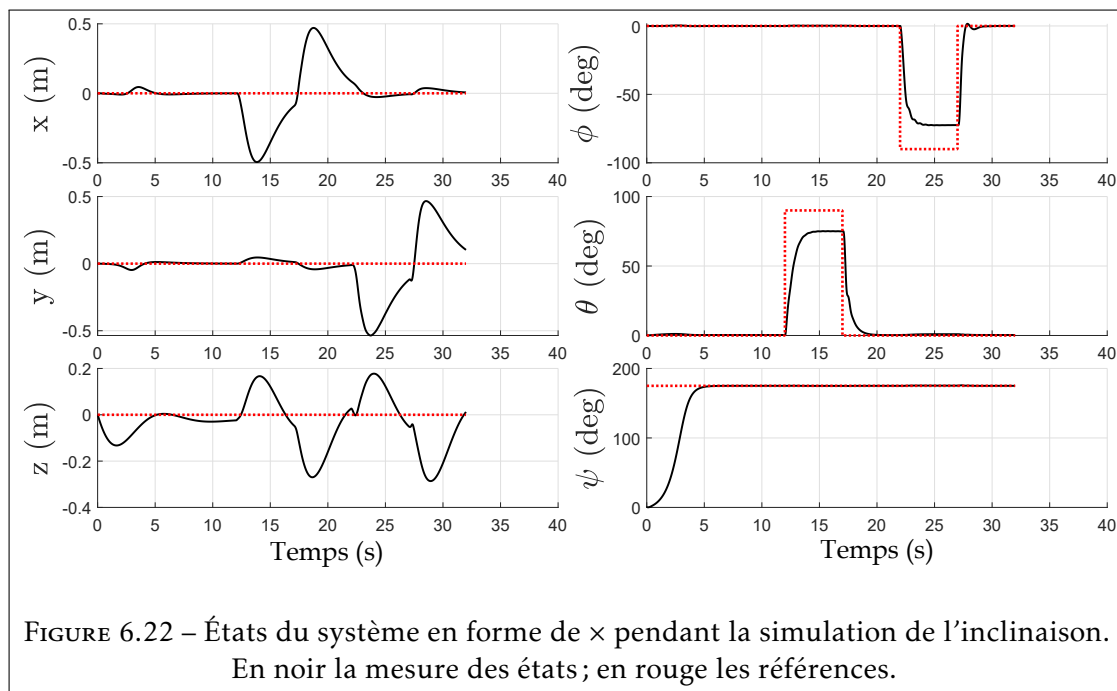
Les essais précédents permettent une première validation du prototype ainsi que des lois de commande et du modèle utilisé. Par la suite on enlève l’hypothèse énoncée précédemment sur la matrice de rotation (qui était  ${}^I\mathcal{R}_B^{-1} = \mathcal{I}$ ). Cela permet au drone de conserver une position donnée tout en s’inclinant. C’est la porte d’entrée vers un vol vertical, ou du moins incliné dans un premier temps. La sous-section suivante présente justement un vol, toujours en configuration  $\times$ , incliné en conservant une position fixe au centre de la volière. C’est durant cette étape que l’on se rend compte de la liaison de tous les chapitres. En effet, il est bien sûr important de bien régler les gains du système mais cela n’a que peu de valeur si les bras ne sont pas parfaitement contrôlés (on fait référence au premier type de servomoteurs). Il en va de même pour nombre de points mécaniques; sans être exhaustif on peut citer l’importance de l’absence de jeu dans la liaison entre servo et tube en carbone. De plus, les éléments constituant la matrice d’inerties doivent également être les plus justes possibles. Mais la partie concernant la plateforme informatique a également son importance, notamment au niveau des latences comme on le discute dans le scénario de vol le long du cercle.

## 6.2.5 Inclinaison du corps

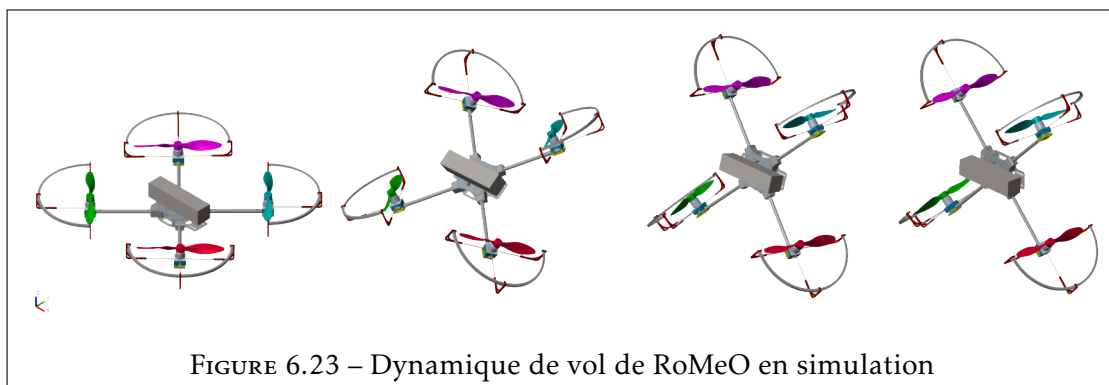
### 6.2.5.1 Simulation

Dans le scénario de cette simulation on teste directement le système à 6 DDL en faisant faire une rotation de  $180^\circ$  au drone sur son lacet, puis une rotation de  $90^\circ$  sur son roulis et son tangage. La Figure 6.22 montre que le système répond bien à la consigne en gardant cependant une erreur statique. Celle-ci peut être compensée d'une part par une composante intégrale, donc en ayant un contrôleur PID. D'autre part, il est également possible d'augmenter les gains pour limiter ce phénomène.

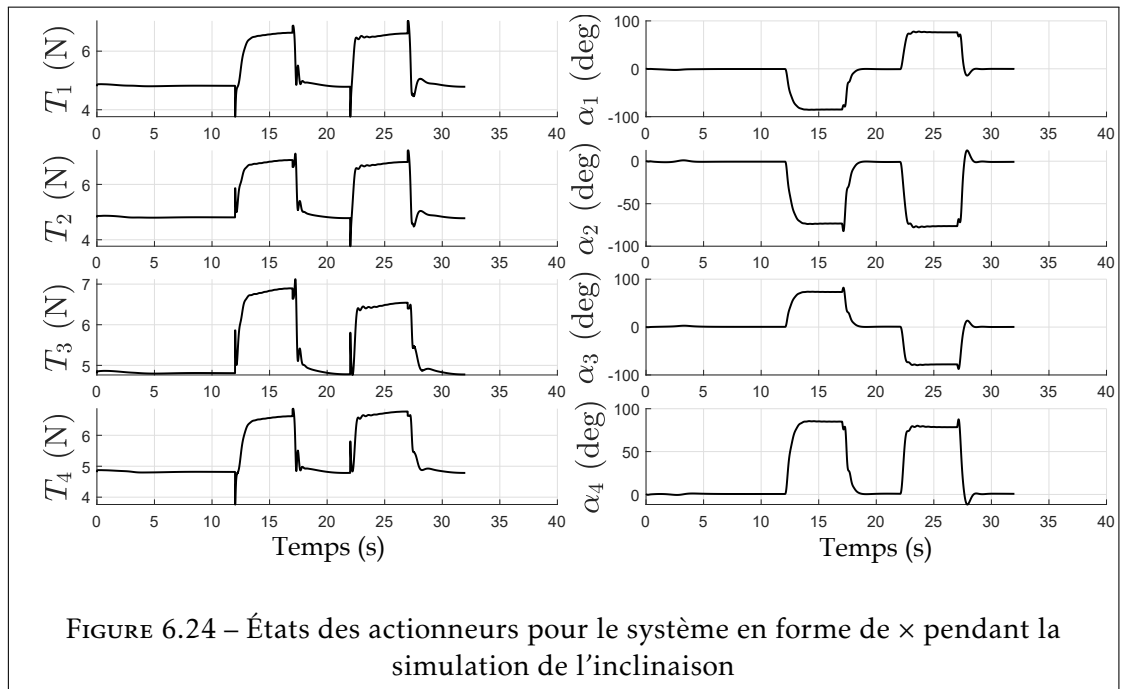
Sur la même figure on remarque également que le drone a tendance à dévier de sa référence de position. Cela provient du retard qui a été implémenté sur les servomoteurs. La poussée des moteurs commence à faire incliner le drone alors que les bras n'ont pas encore commencé à tourner faisant partir le drone. L'erreur est ensuite rapidement corrigée. Il serait d'ailleurs intéressant d'implémenter un prédictiveur (de Smith par exemple) pour prendre de l'avance sur la compensation à effectuer. Ainsi le drone pourrait anticiper l'inclinaison des bras sur l'augmentation de la poussée des moteurs pour obtenir une dynamique plus souple.



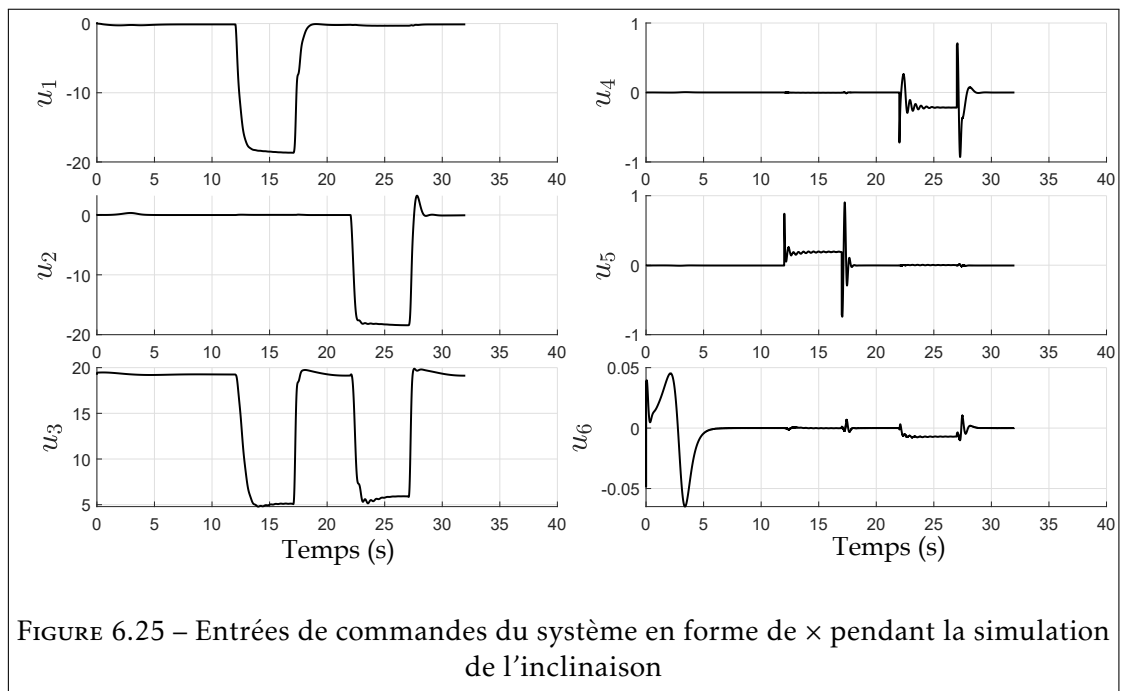
On présente également en Figure 6.23 une superposition des positions adoptées par le drone avec une commande en PID décrite en Section 4.2



La Figure 6.24 montre très clairement une augmentation de la force de poussée nécessaire pour maintenir le drone à une altitude constante. Cela est dû à la configuration en  $\times$  qui implique une certaine contre productivité en vol incliné (les bras poussent en partie l'un contre l'autre).



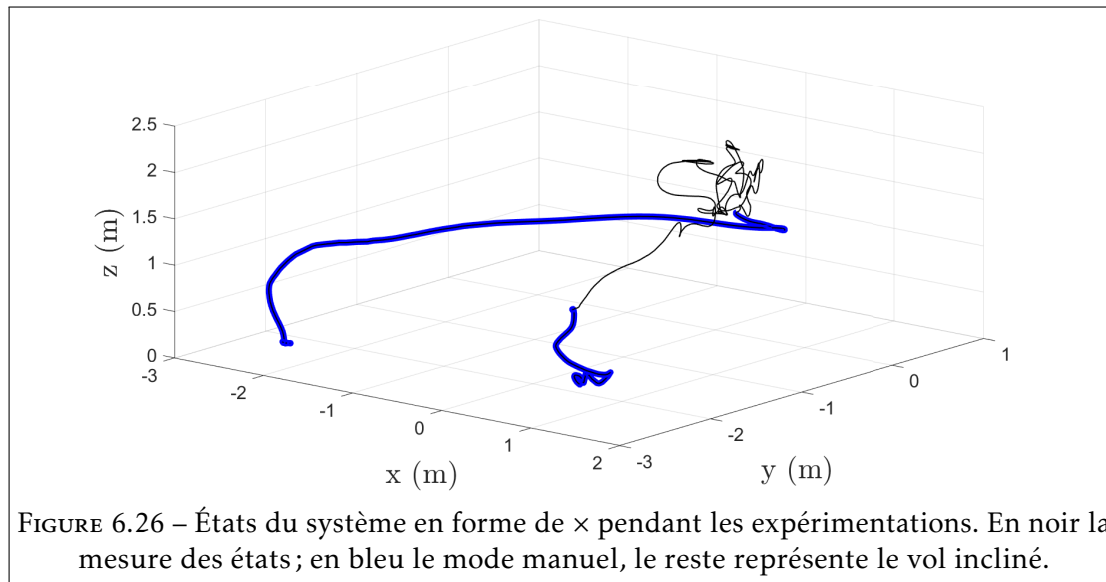
Sur la Figure 6.25 on remarque bien le basculement du drone et l'effet de la matrice de rotation  ${}^I R_B^{-1}$ . Celle-ci transfère la gravité de la loi de commande  $u_3$  vers les autres entrées de commande. Physiquement, on se rend donc compte que la poussée ne s'applique plus sur la verticale mais plutôt sur le côté du drone (dans son référentiel).



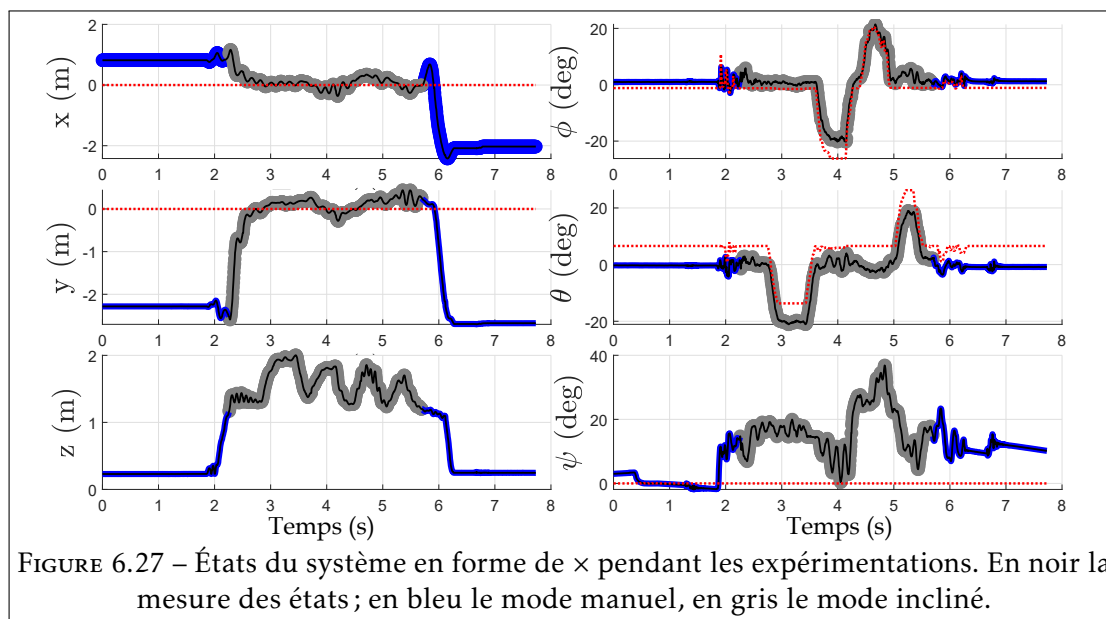
Les deux sections suivantes présentent les résultats obtenus lors des essais réels avec une loi de commande linéaire puis robuste.

### 6.2.5.2 Expérimentations PD

Les expérimentations nécessaires pour arriver au résultat qui suit ont nécessités nombres d'heures. D'une part, le problème d'inversion des hélices évoqué précédemment a été un facteur limitant sur lequel on ne reviendra pas. D'autre part, la dynamique complexe du système et les difficultés liées au biais du cap ont rendu les essais complexes mais également très instructifs. Pour commencer on propose la vidéo <https://youtu.be/kGhH6eN7poE> sur laquelle on présente le drone s'inclinant tout en montrant les données recueillies par la station sol. Sans être le même vol que sur la vidéo on propose en Figure 6.26 le parcours du vol effectué pour ce scénario.

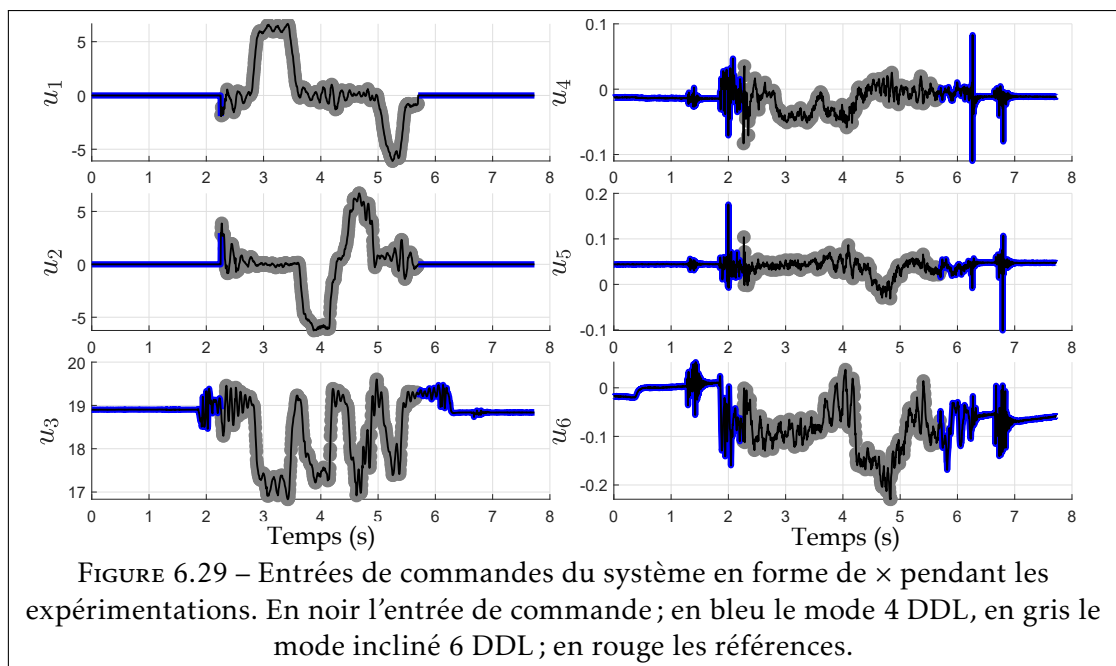
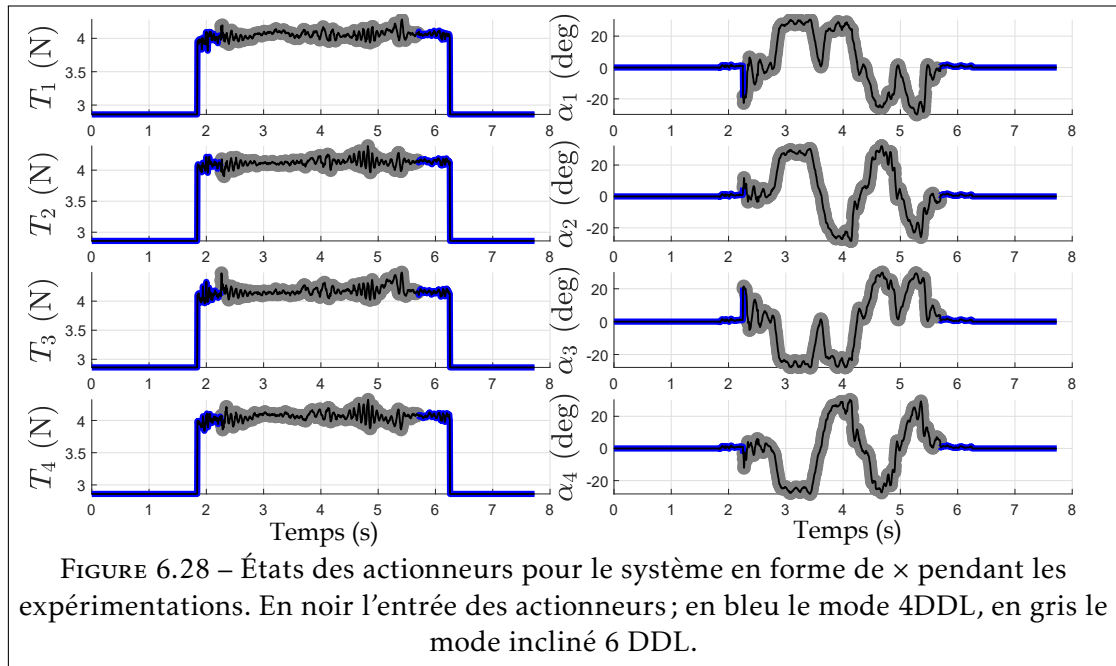


Sur la Figure 6.27 on remarque que la référence de position est rapidement atteinte avec cependant une altitude qui augment lorsque le drone s'incline. Cela vient d'une erreur sur le coefficient  $k_f$ . Ce problème à été réglé dans la configuration  $\checkmark$ .



En Figure 6.28, on observe une bonne dynamique pour les actionneurs. Les bras s'inclinent ce qui permet aux moteurs de conserver une poussée constante tout au long du vol.

C'est justement au cours de ce vol que l'on ressent l'apparition des couples gyroscopiques. Même si ils ne sont pas visibles sur la figure, le changement rapide d'inclinaison à un impact différent selon l'axe de rotation sur lequel il est fait.



Sur la Figure 6.46 des entrées de commande virtuelles, on retrouve clairement le transfert de la masse d'un degré de liberté à un autre, comme en simulation. Le lacet n'est d'ailleurs pas très bien géré, comme on le voit sur la Figure 6.27 et gagnerait à être réglé de manière plus agressive, ou plus robuste. Ce qui nous amène au même scénario mais traité avec le mode glissant super twisting.

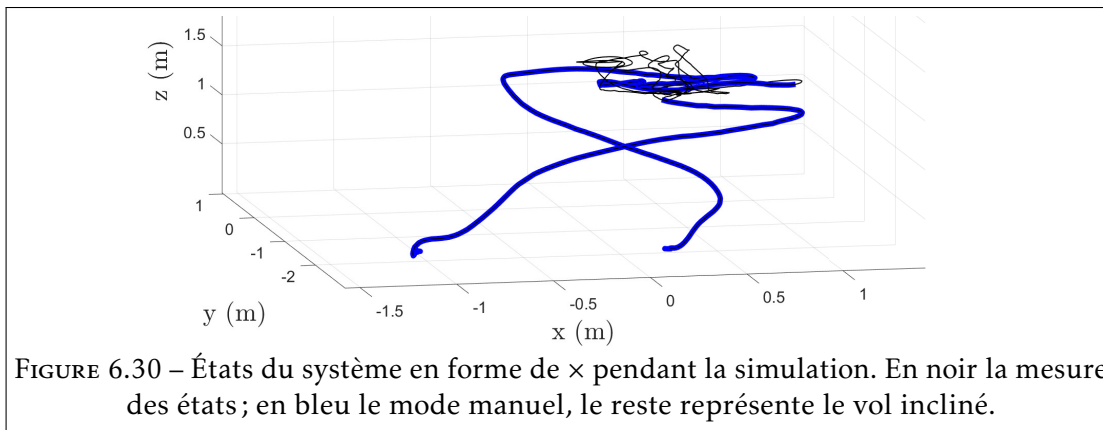
### 6.2.5.3 Expérimentations SMCST

L'utilisation de l'algorithme par mode glissant n'est pas très compliqué à implémenter. On choisit un facteur  $\iota$  à 0.5 (voir eq(C.33)) qui permet d'avoir un bon compromis entre un contrôleur linéaire et un mode glissant classique. On choisit les gains pour ce système comme si c'était un PD classique. On a donc la relation équivalente suivante :

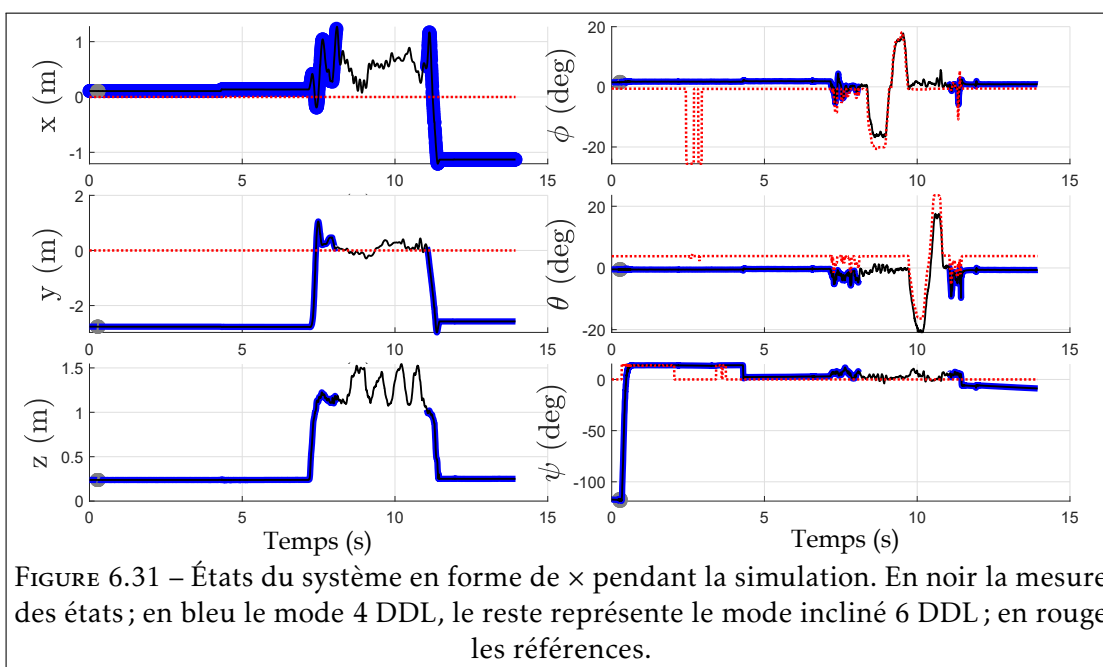
$$-k_d \dot{\xi} - k_p \xi - k_i \int \xi = -\gamma_{1_\xi} (\dot{\xi} - \gamma_{2_\xi} \xi) + \gamma_{3_\xi} \int S_\xi \quad (6.1)$$

on en identifie facilement  $\gamma_{1_\xi} = k_d$ ;  $\gamma_{2_\xi} = \frac{k_p}{\gamma_{1_\xi}}$  et enfin  $\gamma_{3_\xi}$  qui permettra de retirer l'erreur statique est à régler comme le terme  $I$  du PID, donc avec prudence et une valeur faible.

La Figure 6.30 montre un vol qui ressemble à la configuration précédente avec une convergence rapide vers la référence, puis de légères oscillations dues au bruit et biais sur les mesures.



Il en va de même pour les états présentés en Figure 6.31 qui ressemblent énormément à la commande précédente.





On ne remontre pas les graphiques sur les actionneurs ni les entrées de commande car les graphiques n'apportent rien de nouveau par rapport au contrôleur précédent. Cette loi de commande est robuste vis à vis des perturbations, or le système n'est pas perturbé en volière. Il serait donc intéressant d'effectuer des tests de tenue de position tout en appliquant une perturbation. Ceci pourrait être envisagé par un souffleur (comme ceux utilisés pour pousser les feuilles mortes) par exemple.

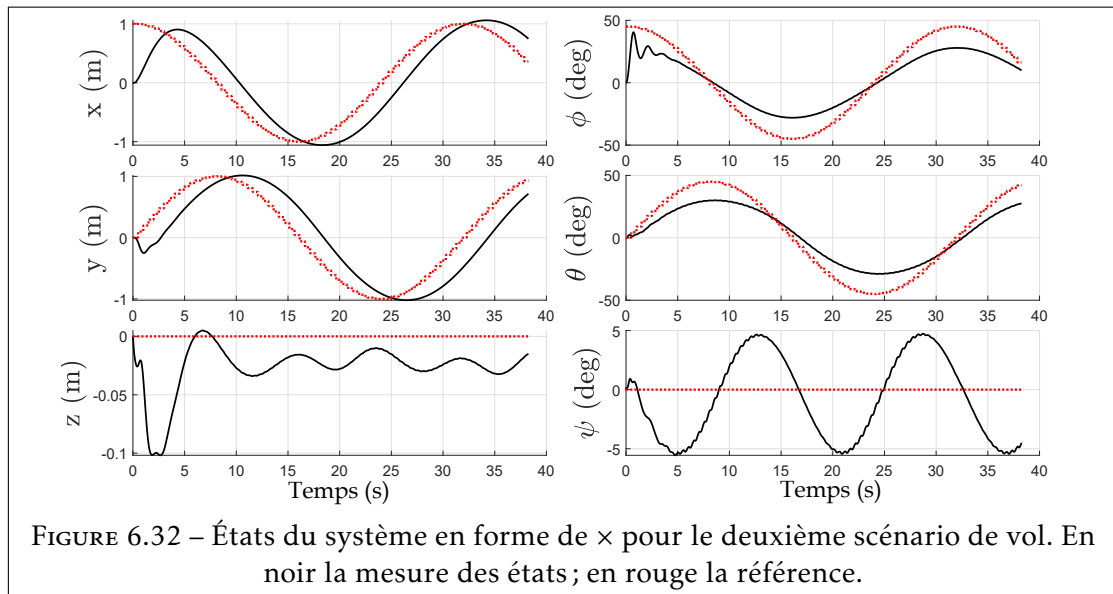
### 6.2.6 Suivi de cercle et inclinaison

Sur cet essai le drone incline son assiette dans le sens opposé à ce que devrait faire un drone classique. Cela est tout à fait possible dans notre cas et montre clairement le découplage entre les différents degrés de liberté du système. La gestion de la position est faite à l'aide de la poussée et des bras, et non pas grâce à l'orientation du drone comme fait classiquement. Après une démonstration des résultats en simulation, on s'attarde sur les expérimentations avec un contrôle linéaire puis robuste. La méthode de suivi de trajectoire utilisée est celle des waypoints définie en fonction du temps, et non plus celle utilisée pour effectuer le carré. Il s'agit de la 3ème méthode de suivi de trajectoire décrite dans la Section D.3.4. Cela permet d'obtenir un trajet beaucoup plus fluide pour le drone.

#### 6.2.6.1 Simulation

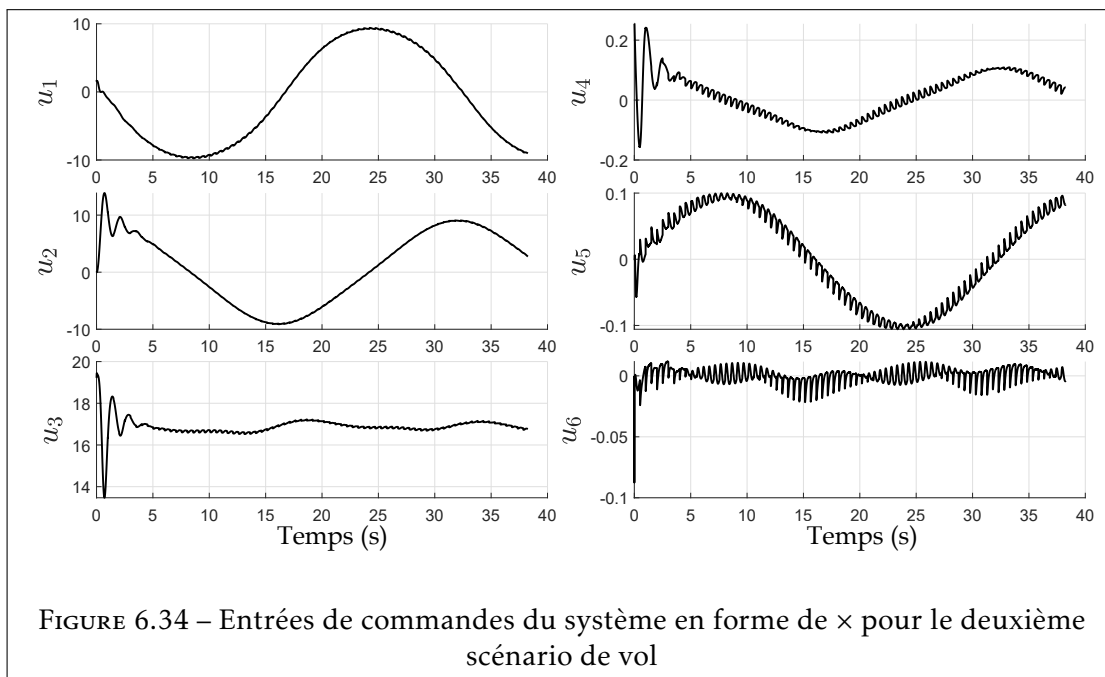
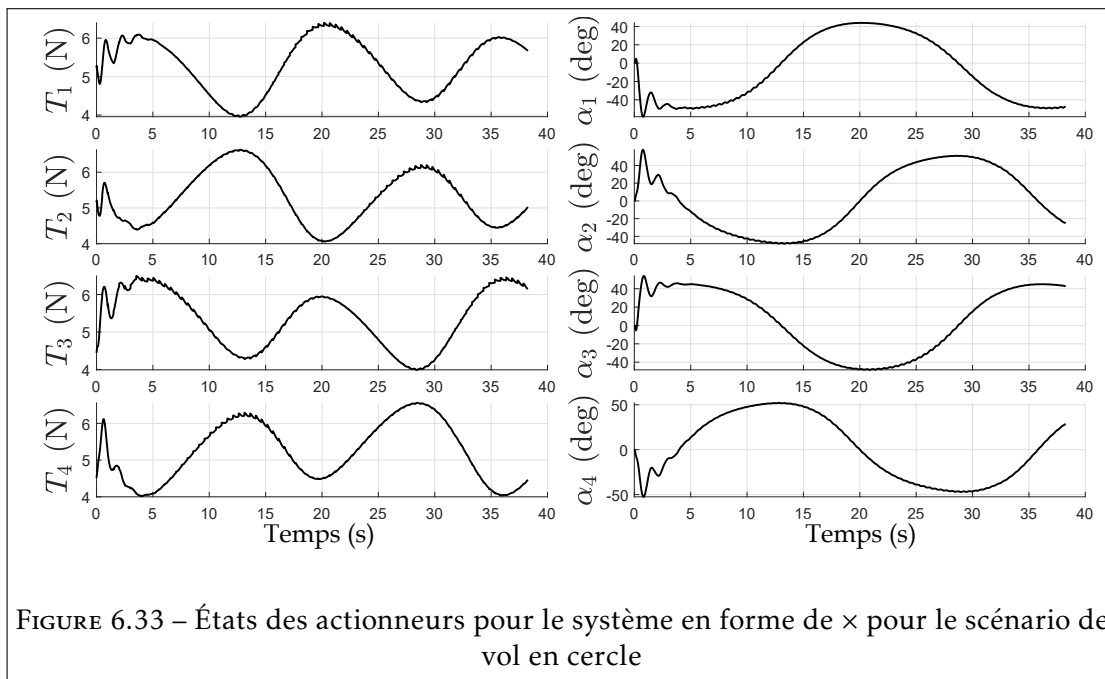
Pour cette simulation on ajuste les gains en prenant  $k_{p_\phi} = 10 \mid k_{p_\theta} = 10 \mid k_{p_\psi} = 6.4$  pour l'attitude en position et  $k_{d_\phi} = 5 \mid k_{d_\theta} = 5 \mid k_{d_\psi} = 4.4$  pour les vitesses angulaires. Cela permet d'obtenir une dynamique plus proche de la réalité.

En Figure 6.32 on présente les états du système pendant la simulation.



On remarque que le système arrive sans difficulté à suivre la consigne moyennant un léger retard. Un effet intéressant est montré sur l'état du lacet  $\psi$ . Le drone s'incline légèrement sur son axe  $\vec{z}_B$  par l'effet des couples de Coriolis provenant de l'équation (3.36). On peut répéter ce phénomène avec des drones grand public en faisant varier l'orientation rapidement sur les deux axes ce qui provoquera un changement de cap du drone.

La Figure 6.33 montre un comportement très doux des actionneurs, ce qui est confirmé par la bonne dynamique du système.

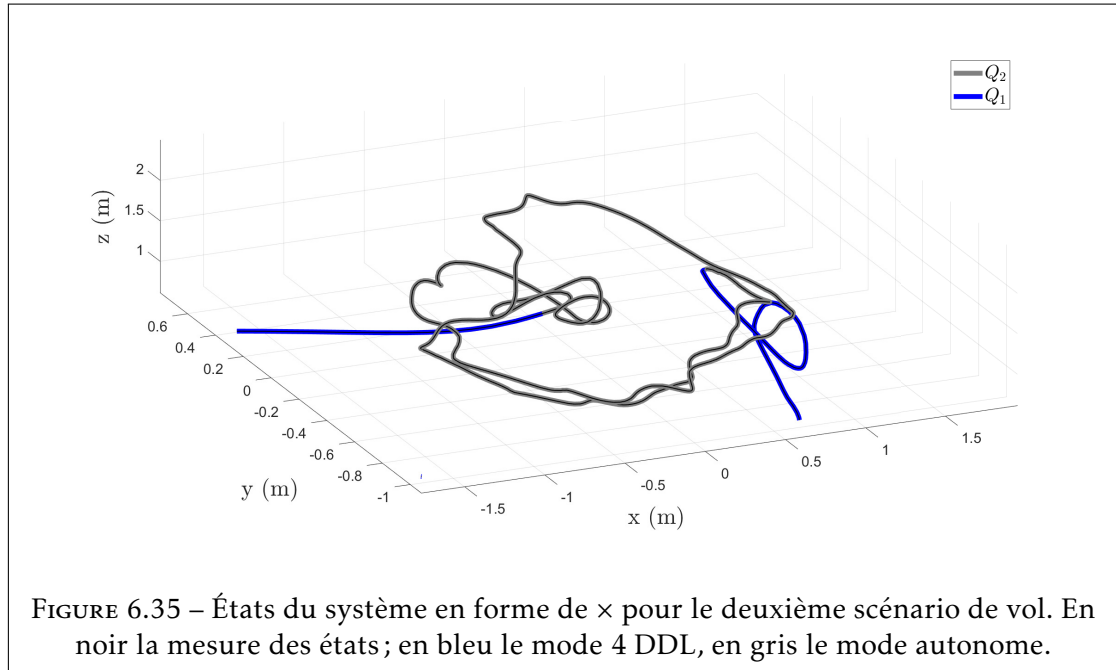


Enfin, la Figure 6.34 présente de légères oscillations d'un ordre supérieur. Cela vient certainement d'une petite asymétrie dans le modèle provoquant un déséquilibre, sans que cela ne soit réellement un problème. D'autre part les valeurs sont négligeables quand on observe l'échelle.

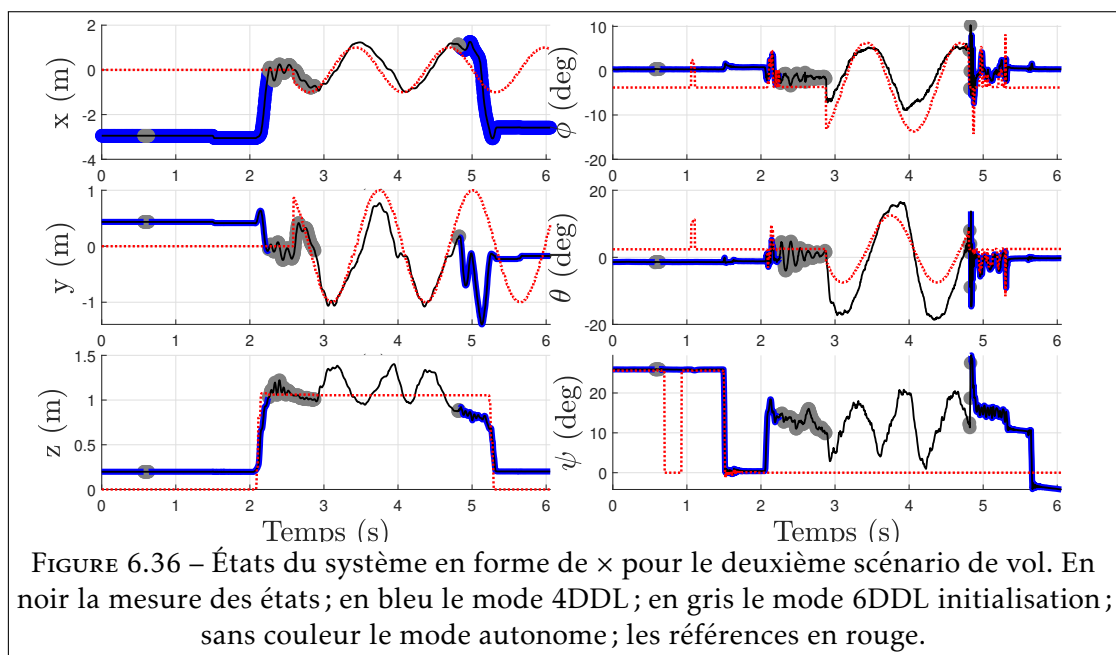
### 6.2.6.2 Réalité - Contrôleur linéaire

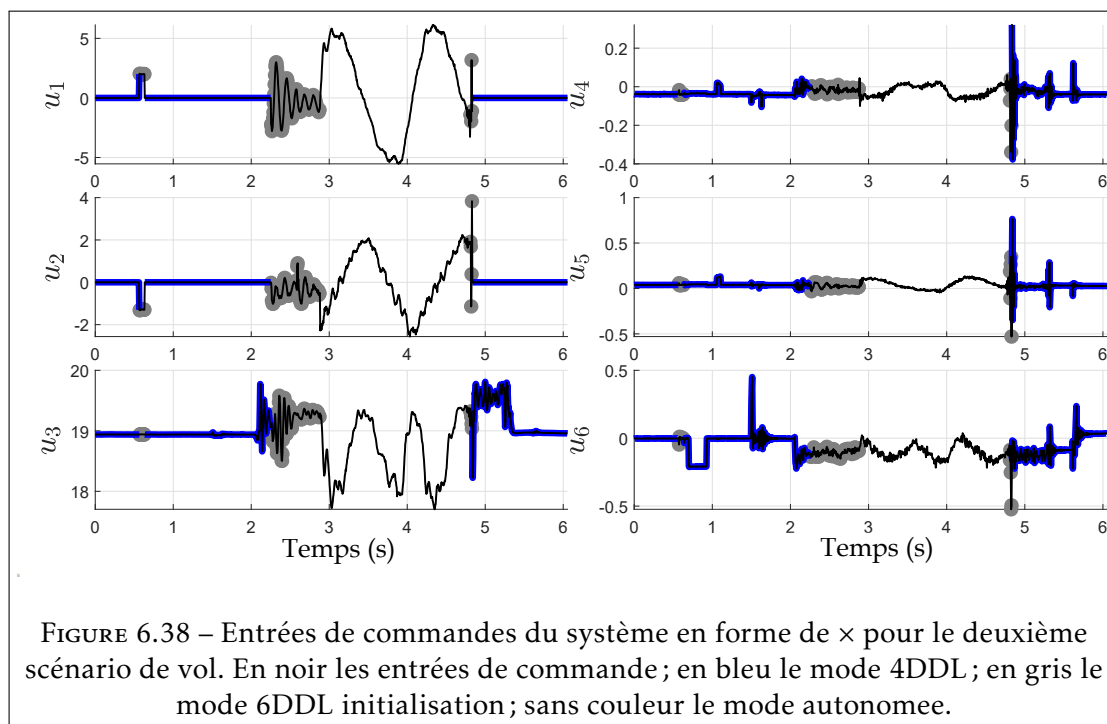
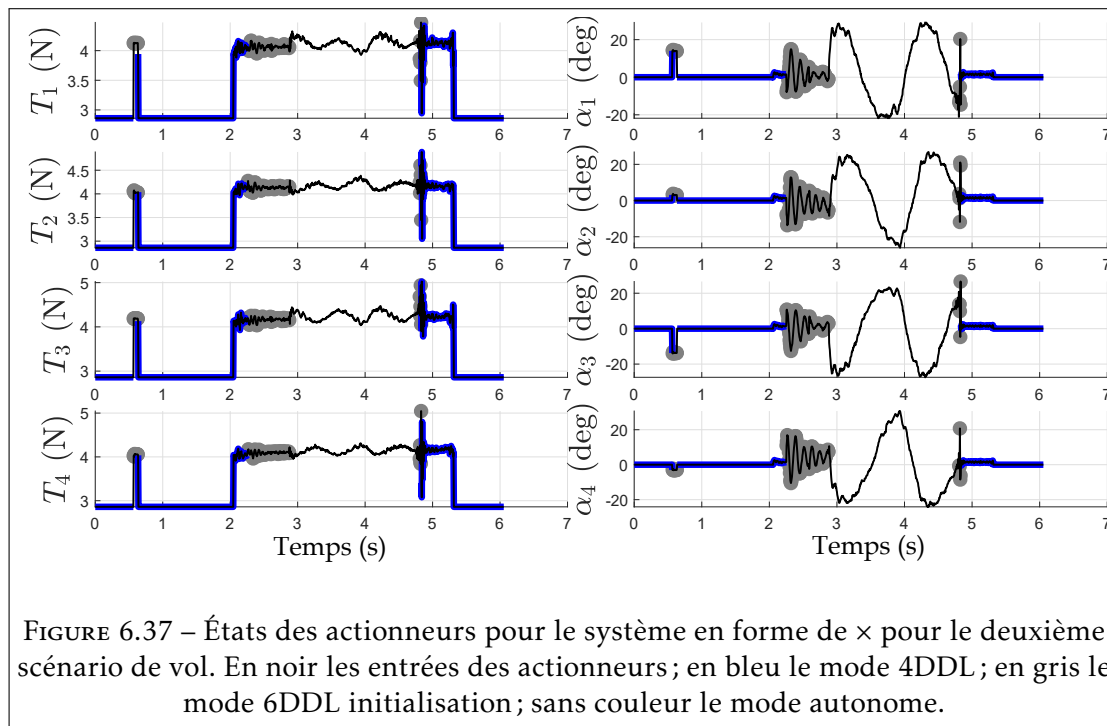
Un point important qui est apparu lors de ces essais est l'impact du délai sur les servomoteurs. Après une confirmation en faisant des tests en simulation on se rend compte que la latence des servomoteurs impacte directement la qualité de réponse à l'inclinaison du système. On se rend alors compte que les dynamiques étaient réglés sur une fréquence de 50Hz et ont donc été ajustés à 200. Ce point est encore un avantage pour ces servomoteurs numériques et donc plus réactifs que les servos conventionnels.

On présente le trajet effectué en forme de cercle sur la Figure 6.35. Après une initialisation en amenant le drone au centre on démarre l'expérimentation et le drone converge rapidement vers sa référence.



En Figure 6.36 le système répond correctement aux consignes demandées pour le positionnement linéaire. Cependant, l'angle  $\phi$  et  $\theta$  ne convergent pas totalement vers la consigne. Sur ce point un effet concernant le couple de rétroaction est apparu. La différence de réponse sur les deux axes est en partie due au coefficient  $k_\tau$  qui est inexact. L'effet est corroboré en simulation, il semble que les coefficients ne soient pas tout à fait corrects menant le drone à une sur- ou sous-inclinaison. En étant incliné, le drone cherchera à atténuer (ou amplifier) l'effet de rétroaction en fonction du facteur, ce qui crée cette altération d'inclinaison. Par la suite on réduit ce coefficient pour donner de meilleurs résultats.

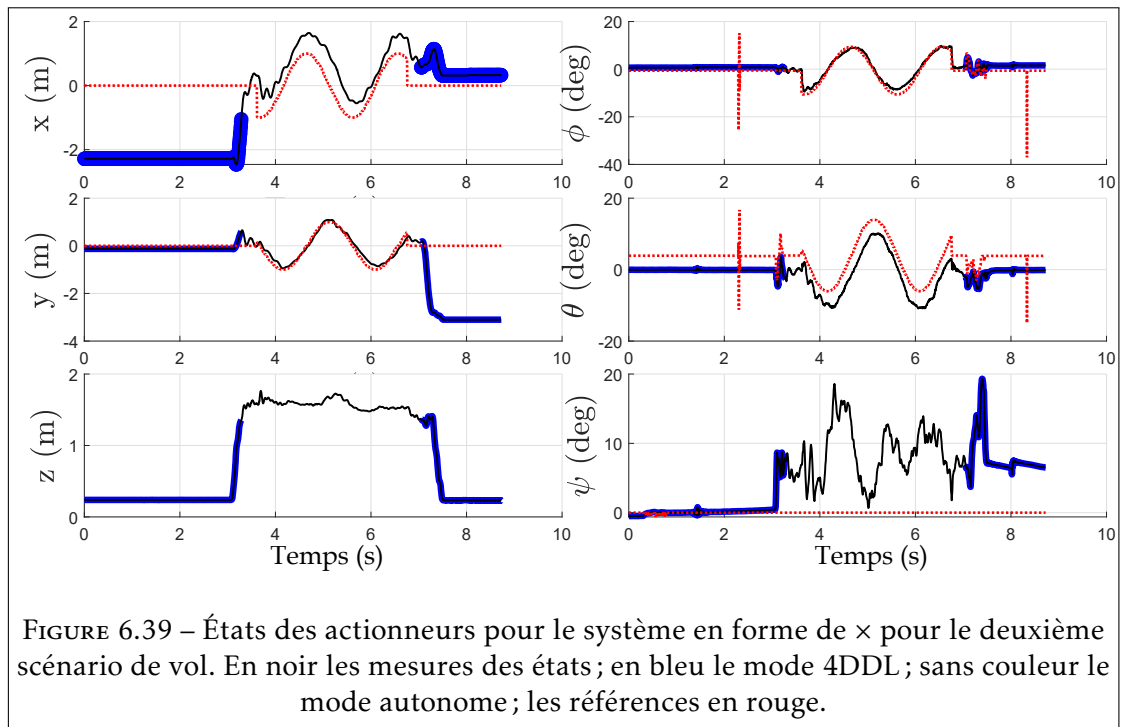




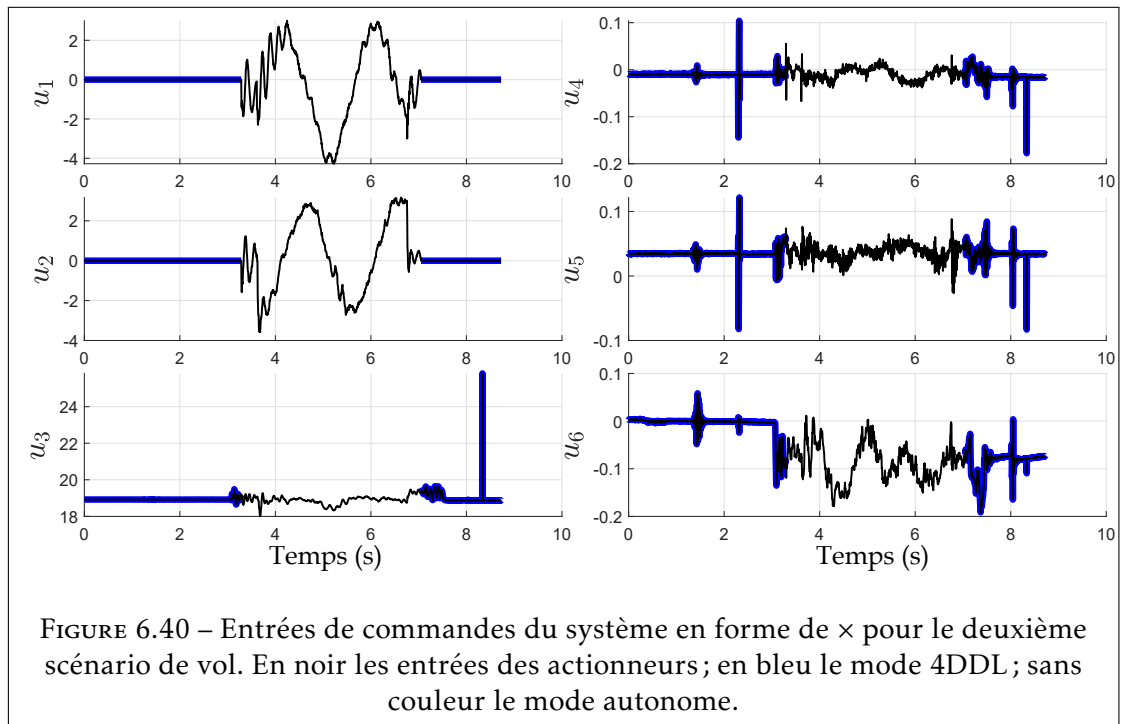
Les expérimentations ont permis de mettre en évidence deux points importants auxquels il faut faire attention pour la suite, la latence des servomoteurs et le coefficient des couples de rétroaction. Ce deuxième point est probablement lié à une mesure inexacte sur le banc de test. En effet on remarque que le roulement permettant la rotation peut corrompre la mesure. A l'avenir on propose de faire attention à ce point pour les tests des coefficients aéronautiques. Ce point sera cependant négligé dans l'étude en configuration  $\tilde{\lambda}$  de par la configuration en hélices contrarotatives.

### 6.2.7 Réalité - Contrôleur SMCST

Avec le contrôleur robuste, on montre en Figure 6.39 que les performances du système sont très semblables à celles avec le contrôleur décrit précédemment.



On montre sur la Figure 6.40 que malgré la bonne convergence du système, le phénomène de chattering n'est pas présent. Cela vient du fait d'une part que la commande est d'un ordre supérieur à un mode glissant classique. D'autre part, l'emploi d'un  $\epsilon = 0.2$  retire définitivement l'effet, comme décrit dans la Section 4.2.5.

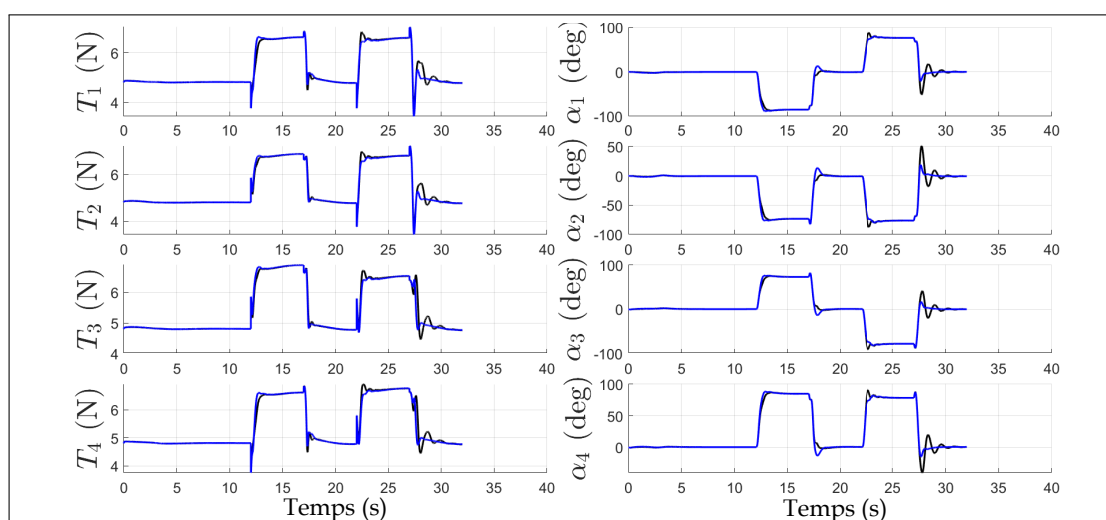
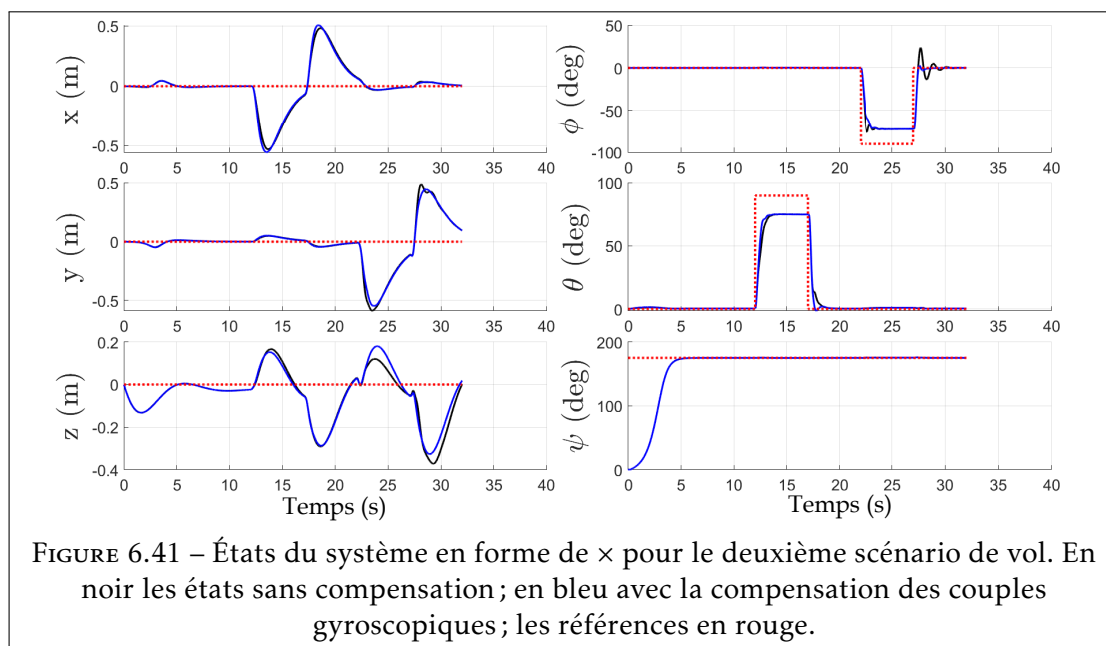


### 6.2.8 Compensation gyroscopique

Pour la mesure d'effets gyroscopiques, on choisit d'effectuer une rotation importante de  $90^\circ$  sur le tangage puis sur le roulis. Entre les deux rotations on remet le drone en position neutre. L'essai a pour but d'imposer une dynamique rapide au drone permettant de remarquer l'impact des couples gyroscopiques. On désigne en noir le comportement du drone sans compensation et en bleu avec.

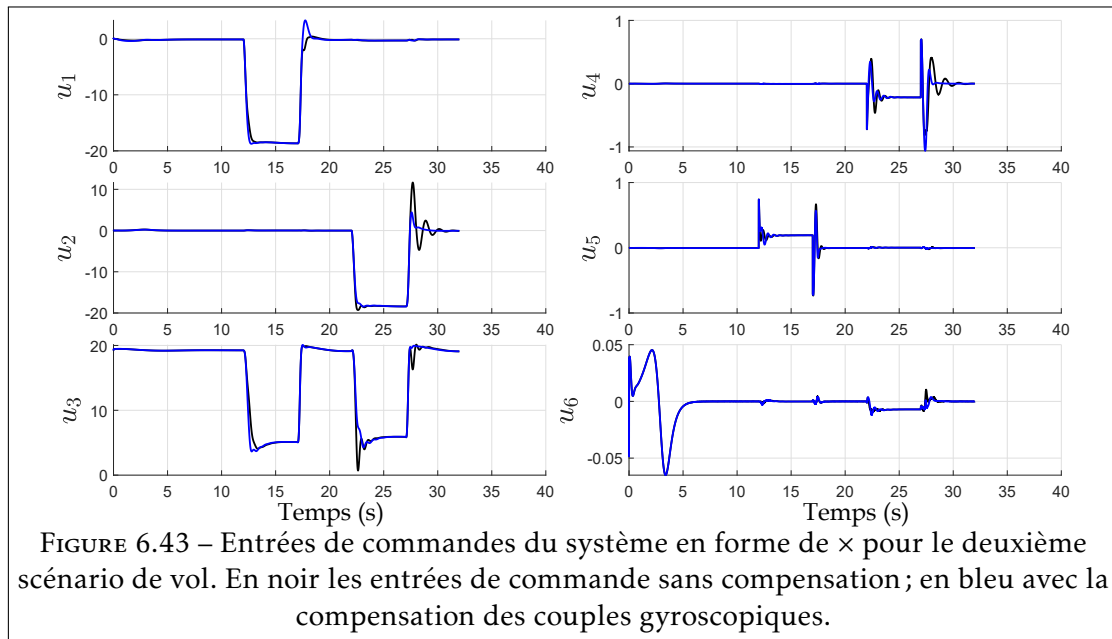
#### 6.2.8.1 Simulation

Pour cette simulation on utilise des gains avec une valeur sur le D inférieure pour pouvoir mieux visualiser l'effet. On utilise  $k_{d_\phi} = 7 \mid k_{d_\theta} = 7 \mid k_{d_\psi} = 4.4$  pour le coefficient dérivatif de l'orientation. La Figure 6.41 montre le comportement du drone dans le scénario de vol décrit précédemment. On remarque que la réponse du système est d'une part sur-amortie (sur  $\theta$ ) et d'autre part oscillatoire (sur  $\phi$ ).



La Figure 6.42 montre les mêmes effets au niveau des actionneurs. Quand on regarde l'équation (6.1) on comprend que les couples gyroscopiques entraînent le drone dans l'un des sens alors qu'ils vont à son encontre dans l'autre. C'est de ce phénomène que résulte l'oscillation et l'amortissement respectivement.

La Figure 6.43 montre un effet intéressant sur la commande. On remarque que la compensation gyroscopique induit un facteur important qui fait augmenter  $u_4$  et  $u_5$  pour permettre au système de converger beaucoup plus linéairement vers son objectif.



### 6.2.8.2 Expérimentations

Comme on l'a déjà évoqué, les expérimentations n'ont pas donné de résultats acceptables. Cependant on donne quelques pistes si l'étude est reprise.

Pour l'observation des couples, on a besoin de mesurer la vitesse de rotation des hélices. Dans les expérimentations que l'on a faites, on s'est basé sur une mesure sur banc de la vitesse de rotation en fonction de la PWM injectée. On a montré une linéarité totale entre les deux facteurs dans la Section 5.1.1.1. On mesure 10 tr/min par ms en plus sur la PWM, avec un offset de 950. Par la suite on recommande de mesurer la valeur en direct au moyen d'un capteur barrage (type TCRT5000). La mesure de vitesse de rotation des servomoteurs est quand à elle directement publiée par les Dynamixel et facilement utilisable dans le modèle.

## 6.3 Configuration $\tilde{\chi}$

La configuration en  $\tilde{\chi}$ , tant discutée dans ce manuscrit a finalement été testée. De la section précédente on apprend qu'il est préférable d'avoir deux hélices contra-rotatives l'une au-dessus de l'autre. Cela permet non seulement de limiter les effets gyroscopiques mais également les effets liés à la trainée des hélices. Dans cette configuration, le lacet est géré à 100% par la rotation des bras. Le Navio est également placé à l'intérieur du drone ce qui permet de garder de l'espace pour l'outil de mesure. L'ensemble des autres pièces électroniques est soigneusement placé entre les deux plaques en carbone permettant d'obtenir un produit optimisé en terme de poids et d'espace.

Dans les expérimentations suivantes on ne montre pas le vol avec l'outil de mesure, faute de temps, ces tests seront faits durant les derniers mois. De même, les tests avec le mode glissant n'ont pas encore pu être faits à ce jour mais ne devraient tarder. Il en va de même pour la compensation des balourds. On peut considérer que l'ensemble des gains utilisés en simulations et lors des expérimentations pour ce drone est comme suit :

|    | $x$ | $y$ | $z$  | $\phi$ | $\theta$ | $\psi$ | $\phi$     | $\phi_{SMCST}$ | $\theta_{SMCST}$ | $\psi_{SMCST}$ |
|----|-----|-----|------|--------|----------|--------|------------|----------------|------------------|----------------|
| kp | 3   | 3   | 2.48 | 20     | 20       | 7.5    | $\gamma_1$ | 4              | 4                | 1.5            |
| kd | 4   | 4   | 3.54 | 5.1    | 5.1      | 4.65   | $\gamma_2$ | 5              | 5                | 2              |
| ki | 0   | 0   | 0    | 0      | 0        | 0      | $\gamma_3$ | 0              | 0                | 0              |

Les expérimentations utilisant le Sliding mode n'ont pas pu être testées, faute de temps. Elles devraient suivre rapidement, permettant un comparatif avec la loi de commande linéaire présentée ci-dessous. Les gains proposés correspondent à ceux fonctionnant en simulation.

### 6.3.1 Simulation

Dans une première phase en simulation, on montre la bonne réponse du système aux références données. Pour cela, on propose un scénario de vol incliné où la position ne change pas au cours du temps. On incline le drone à  $90^\circ$  sur son axe de rotation principal, tout en effectuant une rotation de  $360^\circ$  sur lui-même. Une fois l'opération effectuée, le drone s'incline légèrement sur son deuxième axe mais seulement de  $40^\circ$ . La réponse du système est donnée pour les états sur la Figure 6.44.

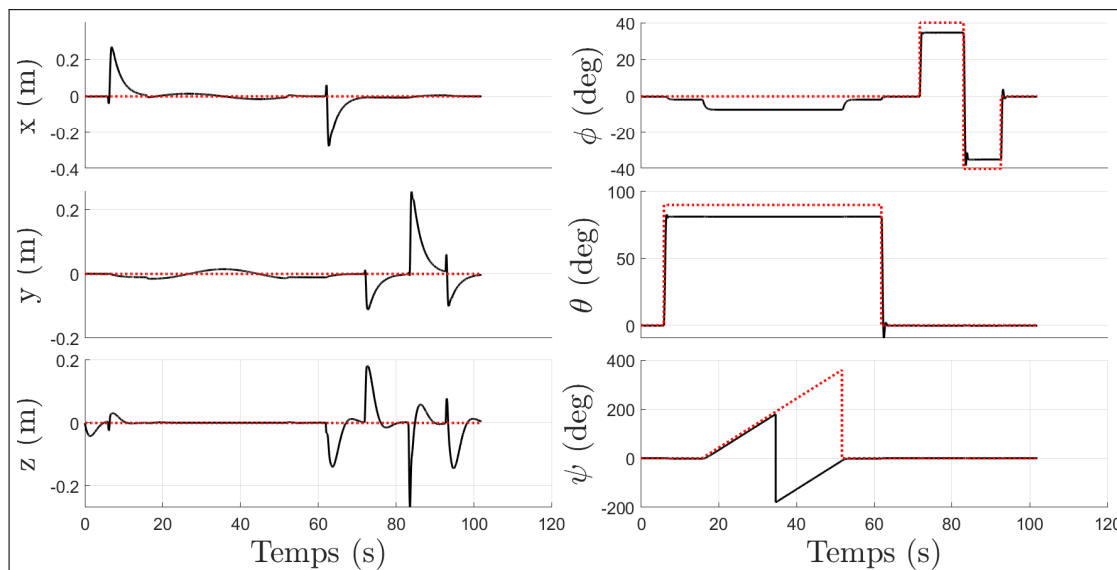
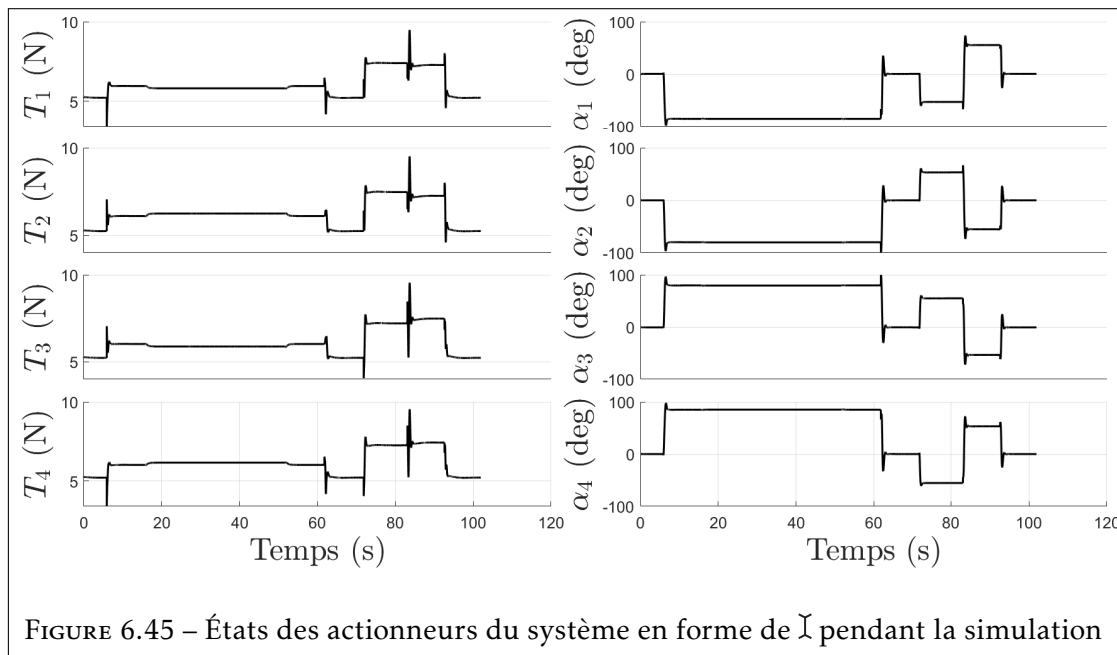


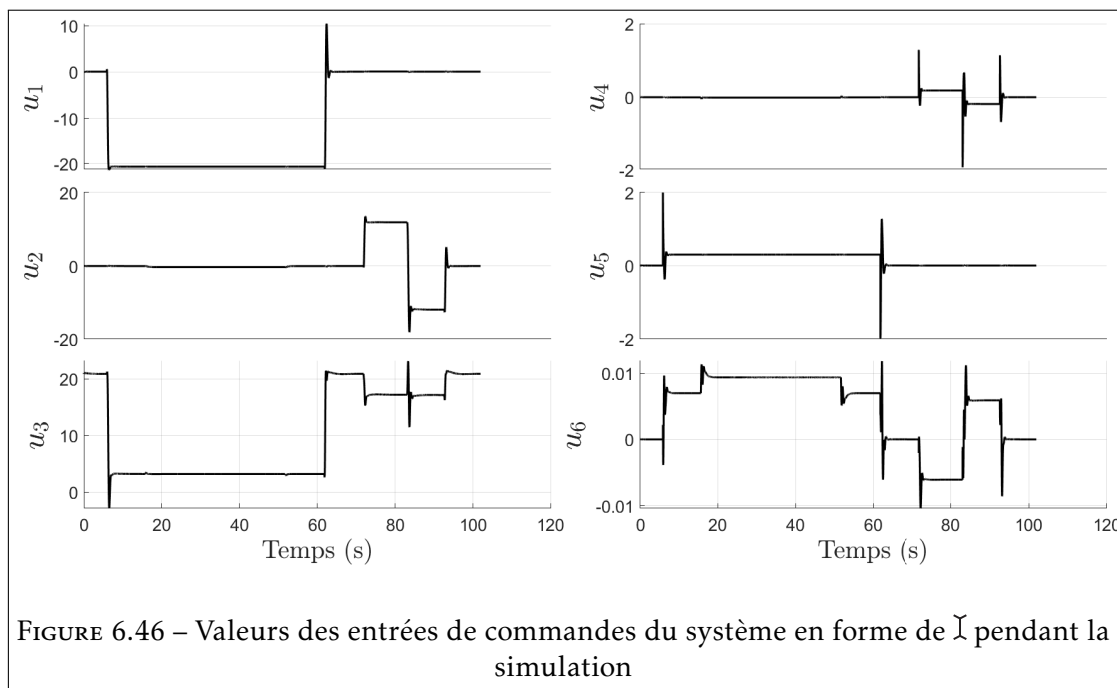
FIGURE 6.44 – États du système en forme de  $\Upsilon$  pendant la simulation. En noir les états ; les références en rouge.

Sur la Figure 6.45 on remarque bien la différence de poussée nécessaire entre la première rotation dans le sens prévu et la deuxième. On remarque aussi que malgré le changement d'inclinaison importante, la force nécessaire pour tenir le drone n'est pas beaucoup plus importante, on possède donc bien une architecture optimisée à notre problème.





Sur la Figure 6.46 des entrées de commande virtuelles, on voit clairement le transfert de la masse d'un degré de liberté à un autre en fonction de l'orientation du drone. D'autre part on voit des à-coups liés à l'arrêt net du drone.



Les tests sur la configuration en forme de  $\times$  ont pris beaucoup de temps avant d'avoir des résultats concluants, alors que les premiers tests sur le drone en  $\checkmark$  ont tout de suite fonctionnés. Après quelques ajustements de gains sur trépied (voir Section 5.1.1.2), on procède aux essais en volière. Malgré sa taille imposante le drone se laisse bien contrôler offrant un vol souple et puissant.

### 6.3.2 Expérimentations PD

Avec cette configuration, on cherche à atteindre des inclinaisons plus importantes qu'avec le drone en configuration  $\times$ . Sur la Figure 6.47 on peut voir une très bonne réponse à la consigne de l'axe  $\vec{y}_B$ . L'angle important sur le roulis présenté sur l'image est due à la représentations des angles d'Euler qui fausse les angles pour des rotations importantes (proche de  $90^\circ$ ). Cependant, on remarque tout de même une rotation sur l'axe  $\vec{z}_B$  pendant le vol. Cela provient du déséquilibre lié au balourd, dont la compensation n'a pas encore été implémentée.

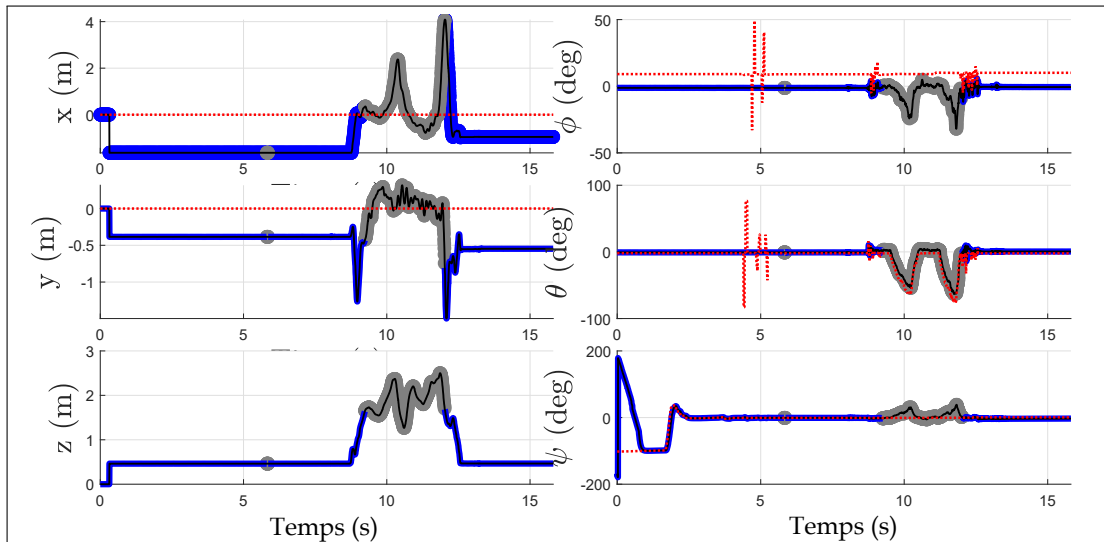


FIGURE 6.47 – États du système en forme de  $\tilde{\lambda}$  pendant les tests réels. En noir les états; en bleu le mode 4DDL; en gris le mode 6DDL; les références en rouge.

Sur la Figure on peut enregistrer un record d'inclinaison à  $63^\circ$  qui est très prometteur. Une photo de ce vol est justement présentée en Figure 6.48. Une vidéo correspondant à ce vol est disponible à l'adresse <https://youtu.be/PLzCfWYW56Q>. Les tests futurs visent à obtenir un vol stable à  $90^\circ$ .

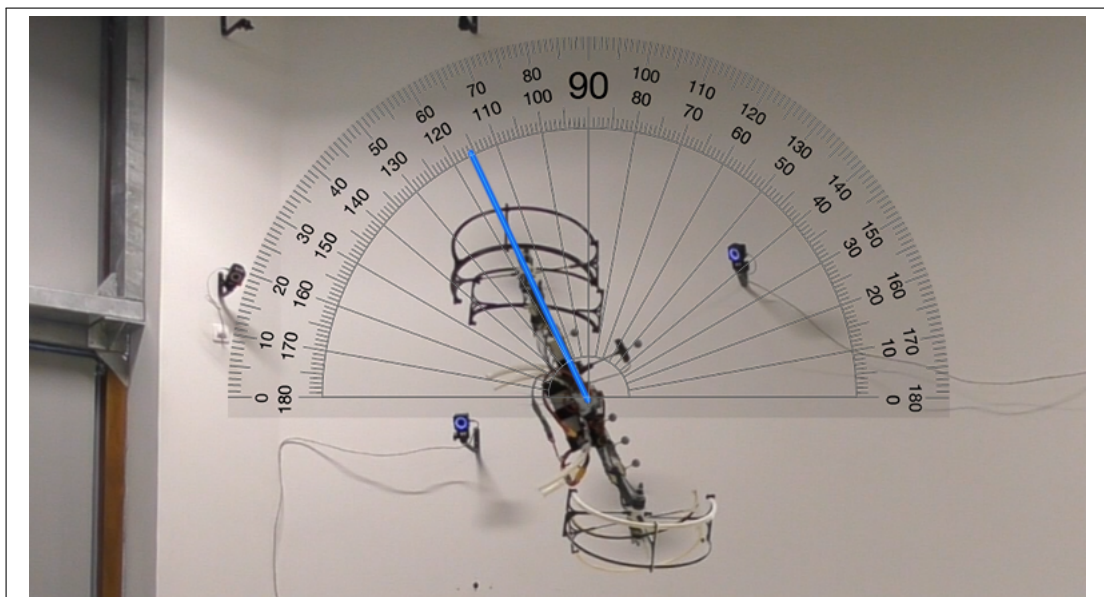
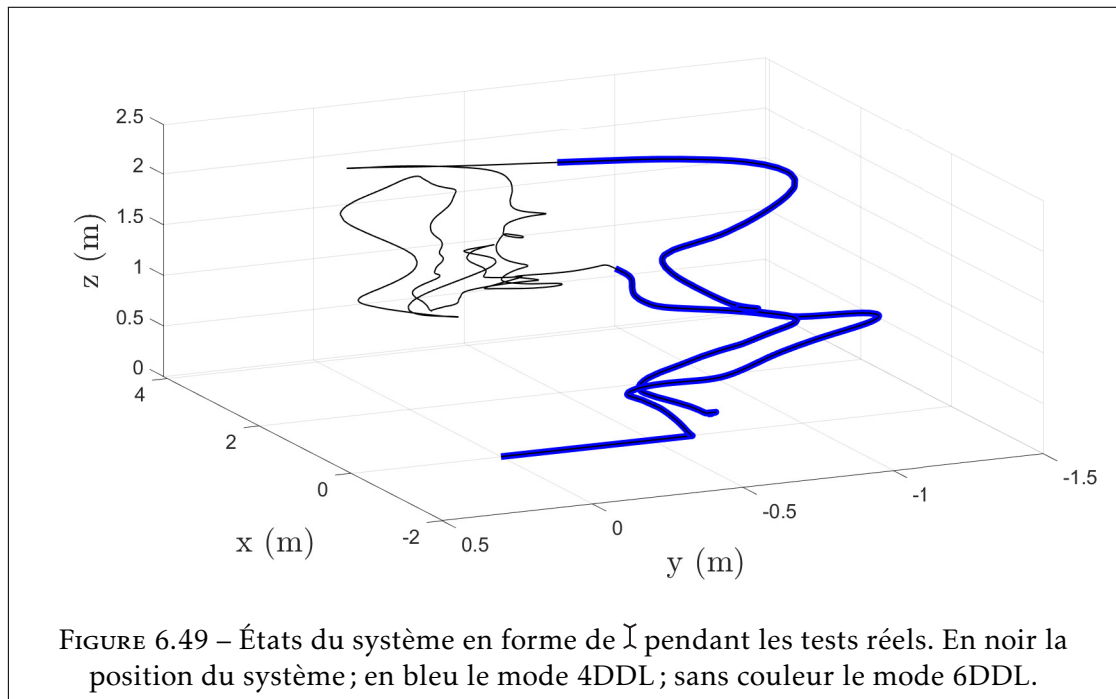
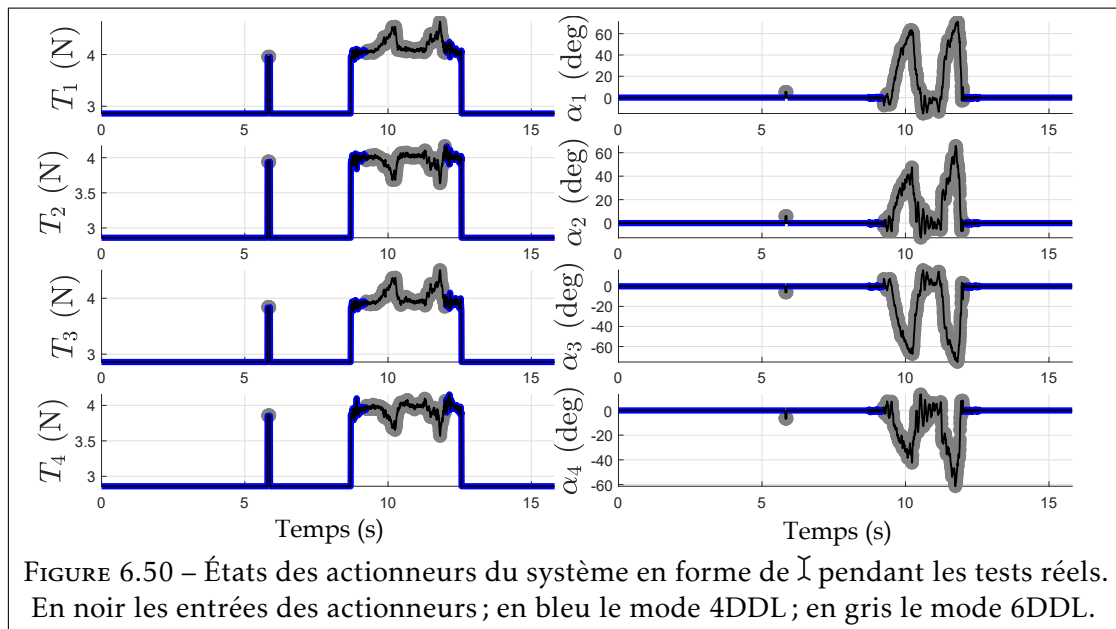


FIGURE 6.48 – Photographie du drone en configuration  $\tilde{\lambda}$  pendant une inclinaison importante

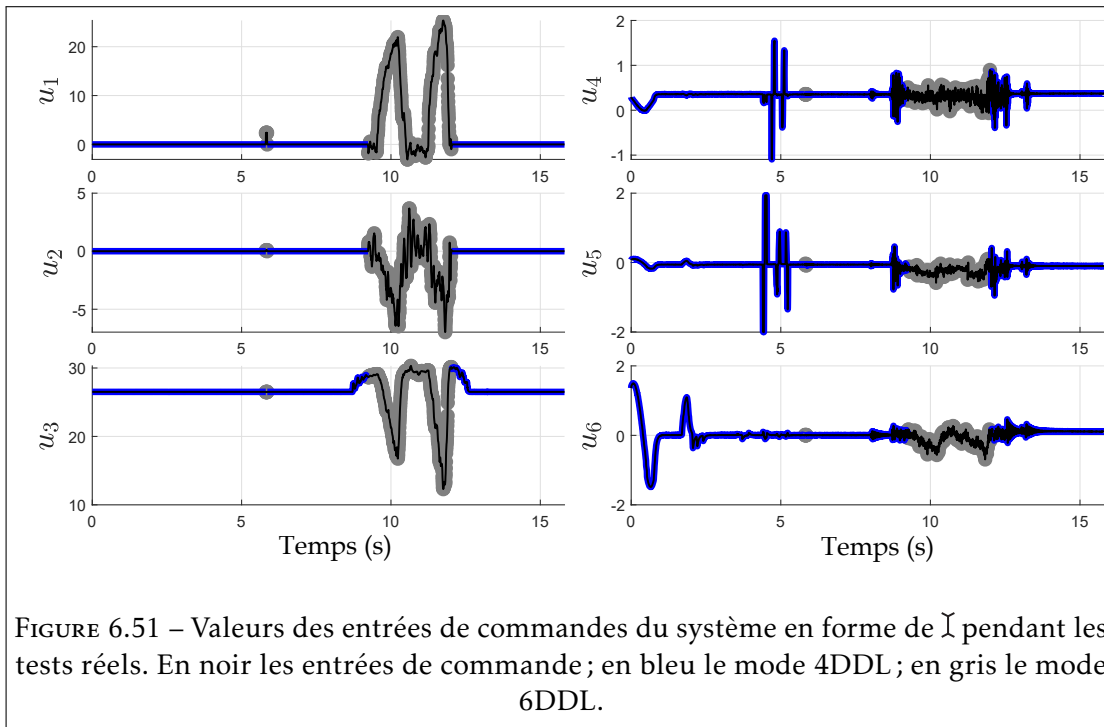
La Figure 6.49 montre le trajet du drone. Comme on le voit également sur les états, le drone n'est pas encore très stable pour maintenir une position fixe. Cela est dû d'une part à des gains mal ajustés, et d'autre part à la présence d'un reliquat des expériences passées. Il s'agit des saturations sur les déplacements linéaires qu'il faut modifier.



Sur la Figure 6.50 on remarque une augmentation de la force de poussée qui n'était pas présente en simulation. Cela s'explique par la présence de l'inclinaison sur l'axe de roulis décrit précédemment.

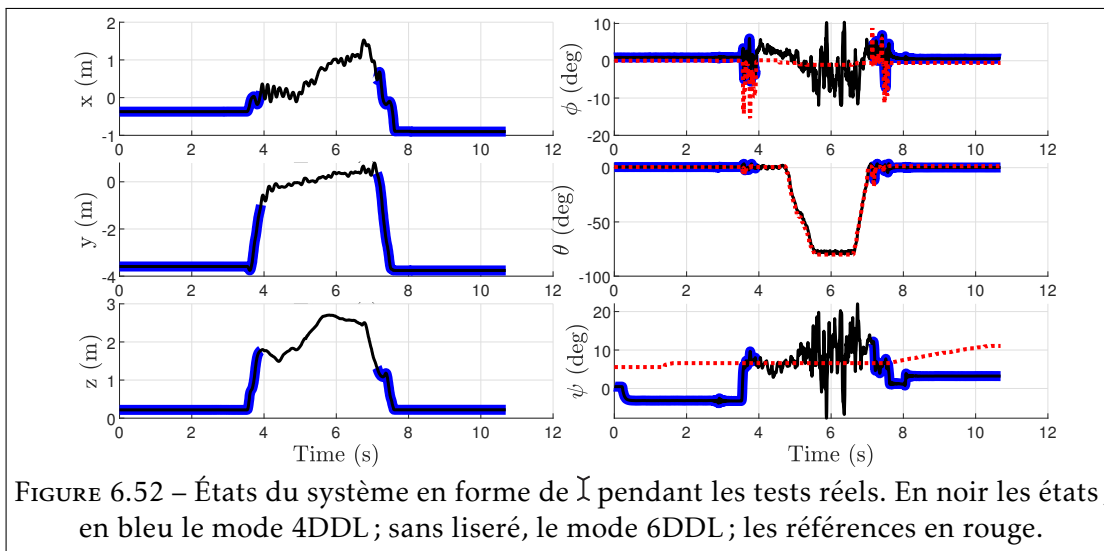


On profite de la Figure 6.51 pour décrire les pics relevés sur  $u_4$  et  $u_5$ . Il s'agit tout simplement d'un test d'initialisation, les moteurs ne sont pas alimentés dans cette configuration. Il en est de même pour l'orientation du lacet, la première vague correspond à la remise à 0 par rapport au référentiel de l'OptiTrack décrit précédemment.



### 6.3.3 Expérimentations PD compensant le balourd

En compensant le balourd du drone grâce à l'estimateur décrit en section 6.48, on réalise de nouvelles expérimentations. Les états mesurés durant cette expérimentation sont donnés en Figure 6.52.



Comme on peut le constater, la compensation du balourd permet largement d'améliorer les performances du système. Premièrement on obtiens un vol nettement plus stable, avec une rotation sur le roulis et le lacet très acceptable. Deuxièmement, on obtiens une rotation sur le tangage à plus de 80 degrés et qui suit parfaitement la consigne. Une vidéo montrant un vol avec cette inclinaison est donnée à l'adresse suivante <https://youtu.be/Ss9UJaZk0Yc>. Une capture d'écran est donnée en Figure 6.53.

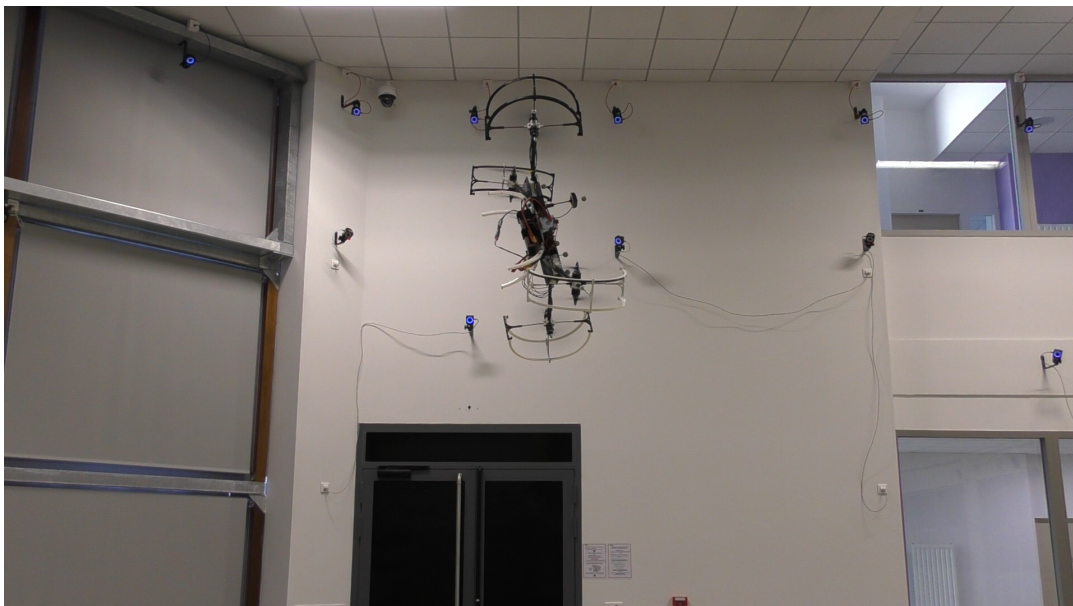


FIGURE 6.53 – Photographie du drone X en vol à plus de 80 degrés

#### 6.3.4 Expérimentations SMST avec l'outil

Pour finir on réalise des tests avec l'outil placé sur le drone. L'essai est réalisé avec la loi de commande robuste. Une vidéo de tests est réalisée et disponible à l'adresse suivante <https://youtu.be/UXjLIbqm-qs>. Une capture d'écran est montrée en Figure 6.54.

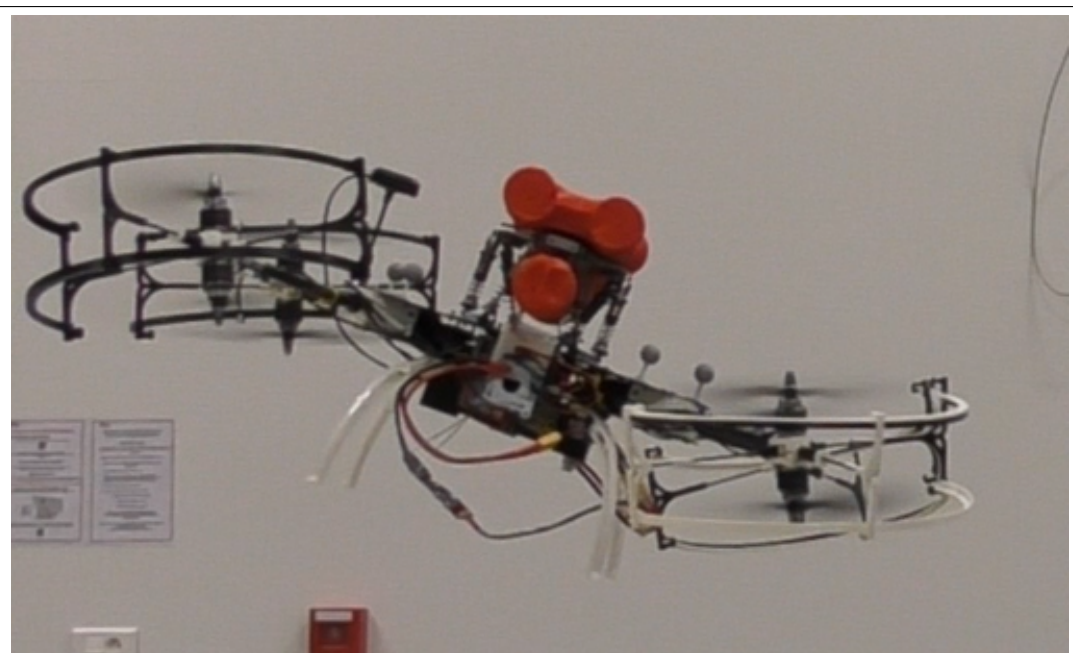


FIGURE 6.54 – Photographie du drone X en vol avec le porte outil

On constate que le drone oscille beaucoup dès que l'inclinaison dépasse une dizaine de degrés. Plusieurs hypothèses sur la cause de ce comportement sont proposées. La première concerne le balourd estimé du porte outil qui n'est peut-être pas convenablement réglé. Une deuxième hypothèse se porte sur la latence induite par les servomoteurs. En effet on remarque que le phénomène oscillatoire est assez caractéristique de système avec des retards.

## 6.4 Conclusion

On constate avec satisfaction, que les tests avec la configuration en  $\Sigma$  se sont extrêmement bien passés. Dès les premières expérimentations, des résultats sont apparus. Cela permet de valider la plateforme mécanique décrite exhaustivement dans le chapitre précédent. La configuration avec deux hélices contra-rotatives résout bon nombre de difficultés.

Le simulateur proposé est fidèle à la réalité. Il a permis d'étudier et d'analyser le problème gyroscopique plus rapidement que par expérimentation. De plus, les gains utilisés dans la simulation ont été faiblement changés durant la phase expérimentale. Il a également donné l'opportunité de tester différentes lois de commande de manière simple et efficace. Le modèle Simulink a pu ensuite être adapté très rapidement pour finalement être transféré dans le drone pour les essais réels.

On remarque que les gains en simulation et en tests réels ne sont pas exactement les mêmes, ceux des expérimentations étant plus faibles. Après plusieurs essais, on se rend compte que si l'on utilise les gains de la simulation, le drone se met à osciller. Ce phénomène vient non seulement de certaines latences qui n'ont pas été prises en compte dans la simulation. L'effet vient également de quelques imperfections mécaniques lors de la réalisation, et enfin de l'imperfection des coefficients d'inertie.

Une réserve peut cependant être faite. L'expérience montre que la mesure de l'orientation dévie beaucoup quand le drone est à la verticale. Cela vient de l'absence de magnétomètre dans la mesure. Pour remédier à ce problème on propose d'utiliser dans le futur la configuration informatique utilisant l'Ardupilot qui offre toute une procédure de calibration pour ce capteur.

De plus amples utilisations de la loi de commande robuste feront l'objet des tests futurs. Ce sera également l'opportunité de tester le dispositif vis à vis des perturbations extérieures. Parmi celles-ci on propose l'emploi de souffleurs pour simuler les rafales de vents qu'il peut y avoir en extérieur.

Enfin, les premières expérimentations faites avec le porte outil, sont encourageantes mais ne remplissent pas totalement le cahier des charge. Par la suite on propose de poursuivre les expérimentations avec une meilleure compensation des balourds et une prise en compte des latences liés aux servomoteurs. Celle-ci pourraient induire un comportement oscillatoire divergeant comme montré sur la vidéo de la section précédente. On pourra également envisager de compenser ce phénomène avec un estimateur de perturbation.

La différence n'est pourtant pas immense et ne limite pas les tests que l'on peut faire avec l'appareil.



# Conclusion générale et Perspectives

## 7.1 Conclusion générale

Dans cette thèse on propose de donner une réponse à la question : comment **rendre possible l'inspection d'ouvrages d'art dans des zones difficilement accessibles**. Après une analyse du problème on choisit d'orienter l'étude vers la réalisation d'un drone. Ce drone peu conventionnel propose d'utiliser un système de bras inclinables pour pouvoir s'orienter librement dans l'espace tout en conservant une position fixe, contrairement à un drone classique qui en est incapable. Cette particularité permet de répondre de manière innovante à l'objectif donné, en souhaitant placer l'outil de mesure directement sur le drone, qui entrera alors en contact avec l'ouvrage.

Après un rappel de la mécanique classique d'un quadricoptère sur un système à 3 DDL puis 6DDL, l'étude propose une analyse complète du système avec bras inclinables. Quatre configurations sont proposés et analysés mécaniquement. La première, en 2D, propose de reconsidérer le système du PVTOL en permettant l'inclinaison des hélices indépendamment. Cette introduction permet d'étudier une deuxième configuration 6DDL en forme de "+" qui permet de définir toutes les variables et les principes mécaniques utilisés. Cette configuration ne permet cependant pas de réaliser efficacement l'objectif que l'on se fixe, car, en vol vertical, le drone ne peut utiliser que la moitié des ses hélices pour le porter. Une troisième configuration en forme de  $\times$  est donc proposée et sera retenue pour réaliser un premier prototype. La dernière configuration consiste à optimiser la précédente en étudiant une configuration novatrice en forme de  $\lambda$ . Une étude énergétique démontre l'intérêt de cette configuration par rapport aux deux précédentes, ce qui constitue la réelle innovation technologique qu'apporte ce document.

Cette première approche permet de répondre à la première problématique que l'on s'est posée, à savoir "Quelle forme doit avoir le robot qui effectue les mesures sur ouvrage?". La forme en  $\lambda$  permet à la fois d'optimiser la force des moteurs utilisée mais également de laisser une latitude de mouvement latérale possible.

Vient ensuite le chapitre sur la commande du système. Ici, il s'agit de proposer une solution de commande pour ce système hautement non-linéaire et sur-actionné. Après un rappel sur les loi de commandes classiques, on propose d'utiliser des entrées de commande virtuelles pour rendre le système totalement actionné. Celles-ci permettent de découpler la dynamique des 6 DDL du système rendant la modélisation à la fois plus intuitive et le système plus facile à contrôler.



On propose d'utiliser la méthode de linéarisations pour pouvoir commander le système avec n'importe quel algorithme. Dans cette étude on propose de détailler trois algorithmes, une commande linéaire, un algorithme par saturations emboîtées et finalement une commande robuste. Ces trois lois de commande sont détaillées et utilisées sur le système.

L'intérêt d'une loi de commande robuste est la résistance du dispositif face aux perturbations externes. Étant donné que le système est destiné à être utilisé en extérieur, où les perturbations sont importantes, ce contrôleur permet de répondre à la deuxième question à savoir : "Comment sera commandé ce drone?".

Une analyse des singularités du système permet également de compléter l'étude en proposant des solutions pour compenser cet effet, mais aussi le problème de balourd lié à l'implémentation d'un outil de mesure dans le drone.

Le chapitre suivant décrit en détail la plateforme développée pour réaliser les deux prototypes de drone. Dans un premier temps on cherche à répondre la question "Quels points de conception mécanique devront être optimisés pendant la réalisation du dispositif?". Une grande partie de l'étude est liée aux bras qui sont la principale différence entre un drone classique et le nouveau drone proposé dans cette étude. On choisit de concevoir les pièces en CAO afin de répondre à un besoin d'optimisation mécanique, ce qui permet également d'obtenir une solution pour remplacer les pièces facilement en cas de problème lors des essais. Une recherche d'optimisation est donc dédiée aux pièces les plus vulnérables (protections d'hélices) mais également aux pièces structurelles (porte-servos). Cette démarche a été étendue à toutes les pièces constituant le système en cherchant à obtenir une structure fonctionnelle et légère, facilitée par l'impression 3D qui permet de faire des pièces creuses mais très rigides.

Le dernier point traité concerne la partie informatique du système. On rappelle la question que l'on s'est posée au début de l'étude : "Quelle plateforme informatique permet d'effectuer des tests rapidement et efficacement?". Après l'exposition des solutions existantes qui peuvent répondre au besoin de cette thèse, on recherche une méthode offrant une plus grande ergonomie. Pour cela on propose un système basé sur une interface Matlab Simulink et permettant deux choses. D'une part elle constitue la possibilité de réaliser des simulations réalistes avec une importation possible du modèle CAO du système. D'autre part, la programmation est remplacée par une interface graphique permettant de réaliser rapidement des changements, tout en conservant la possibilité de cibler un micro-contrôleur sur le drone. C'est donc par ce système que l'on choisit de répondre à la problématique posée. On remarque que c'est une architecture innovante pour le laboratoire et permettra à l'avenir d'améliorer la pratique d'expérimentations aussi bien sur les drones que pour d'autres types de véhicules, comme les véhicules terrestres par exemple.

Les tests de simulations et expérimentaux ont permis de valider le système pas à pas. On montre la progression des essais en commençant dans une configuration très proche d'un drone classique. Les étapes suivantes débloquent des fonctionnalités poussant le drone vers ses limites physiques. Un premier essai consiste à incliner les bras plutôt que le corps pour atteindre une position donnée. Par la suite on améliore l'essai en réalisant un véritable trajet. Cette étape permet par la suite d'entrer dans le vif du sujet en réalisant l'opération inverse : garder une position fixe mais s'incliner à une orientation donnée. Des tests concluants ouvrent la voie vers un deuxième scénario de vol complexe, permettant un vol le long d'un cercle tout en étant incliné dans le sens inverse à ce que ferait un drone conventionnel pour réaliser la même tâche.

Les essais, et notamment les scénarii de vols, permettent de mettre en évidence nombre de détails directement liés à la mécanique complexe du dispositif. Cette phase à pris énormément de temps mais est également très instructive et complète la formation.

Globalement dans cette thèse on reprend un concept de bras inclinables existant déjà depuis quelques années. Mais on apporte une réelle contribution scientifique dans le domaine, en proposant à la fois une configuration novatrice, ainsi qu'une plateforme permettant le prototypage extrêmement rapide de systèmes très complexes, avec simulateur à l'appui.

Afin de réaliser une critique la plus objective possible je dirais que l'objectif que l'on s'est fixé est partiellement atteint. Un dispositif ayant le potentiel de réaliser les mesures est décrit dans cet ouvrage. Des essais concrets avec un outil de mesure doivent encore être effectués pour valider totalement l'objectif fixé, ce qui nous amène aux travaux restants mais également aux perspectives que l'on propose pour la suite du projet.

## 7.2 Perspective

La suite imminente de l'étude concerne l'utilisation du prototype avec un outil de mesure en vol. Après cette phase préliminaire et en validant un vol robuste, on pourra envisager un vol au contact d'une paroi.

Pour continuer le projet on propose plusieurs axes de réflexion. Pour commencer, on pourra considérer les parties plus haut niveau de l'étude. On donnera par exemple une attention particulière à la partie sécurité du système. Ceci doit se faire au travers du processus d'homologation du drone. Parmi les dispositifs à intégrer on note notamment la coupure d'urgence des moteurs et l'ajout d'un parachute.

Au niveau logiciel on pourra également améliorer le dispositif. Une fois que la loi de commande sera parfaitement validée, on propose d'utiliser l'ardupilot pour offrir à l'utilisateur une interface plus intuitive que Matlab. Cela permettra également de pouvoir rajouter facilement une caméra sur le Raspberry pi pour une expérience de vol plus agréable.

Toujours dans le volet de la partie informatique, l'adaptation de l'autopilote (comme ardupilot) permettra d'avoir une machine à état. En fonction de la situation dans laquelle le drone se trouve, différentes sécurités pourront prévenir des problèmes de vol. Pour le moment, le changement des états est manuel en utilisant les boutons sur la télécommande ; à l'avenir il devrait être le plus automatique possible. Pour illustrer ce point, prenons l'exemple du décollage du drone. A l'heure actuelle il suffit d'utiliser un interrupteur pour faire tourner les hélices. Sur la plupart des drones du commerce le démarrage se fait par un positionnement particulier des deux joysticks de la télécommande.

Un point de perspective supplémentaire pourrais consister à changer la matrice d'allocation quand le drone se situe dans les zones de singularités. En effet, les moteurs n'étant pas utiles, voir même néfaste, leur utilisation pourrait être modifiée, voir réduite à néant.

Pour finir on propose une liste non exhaustive de sujets de réflexions directement liés au thème qui peuvent contribuer à l'amélioration du système :

- système modulaire (multi-outils)
- ergonomie pour l'utilisateur
- optimisation mécanique (par méthode des éléments finis par exemple)
- optimisation aérodynamique par une analyse multiphysique
- optimisation du système pour la maintenance et la mise en service
- sécurité et norme du produit

Le mot de la fin : cette thèse m'a permis d'aborder énormément de points différents liés au domaine de la mécatronique. Trois années de ma vie sont passées à une vitesse fulgurante, et sans regrets. Des moments forts, beaucoup d'émotions de tous types et surtout un projet incroyablement riche permettant un développement aussi bien intellectuel que personnel.

Merci de m'avoir lu jusqu'ici et de m'avoir laissé la chance de partager mon expérience du mieux que j'ai pu, avec vous, lecteur.

# Bibliographie

## Liste de Publications

1. A. Offermann, P. Castillo, et J. D. Miras, en, in 2019 International Conference on Unmanned Aircraft Systems (ICUAS) (Juin 2019), Atlanta, USA, pp. 1451-1457.
2. A. Offermann, P. Castillo, et J. De Miras, in 2020 28th Mediterranean Conference on Control and Automation (MED), ISSN : 2473-3504 (Sept. 2020), Saint-Raphaël, France, pp. 50-55.

## Références

- [1] François DERKX et Jean Luc SORIN. « Inspection des ouvrages d'art par drone : Bilan et perspectives des travaux du LCPC ». In : *BLPC* (déc. 2008), p. 19.
- [2] D. POINEAU. *Inspection Des Ouvrages d'art En Béton Armé*. Ginger CEBTP formation, 2014.
- [3] Peter VAN BLYENBURGH. « Uavs Where Do We Stand? » In : *Military Technology* (mars 1999), p. 29-30.
- [4] Arnaud KOEHL. « Modélisation, observation et commande d'un drone miniature à birotor coaxial ». fr. In : (), p. 211.
- [5] Pascal BRISSET. *Drones civils Perspectives et réalités*. Août 2004.
- [6] F DERKX, J DUMOULIN, J-L SORIN et al. « Concept de Plate-forme Mobile Instrumentée (PMI) pour l'inspection des ouvrages d'art ». fr. In : (), p. 6.
- [7] Valentin RIVIERE, Augustin MANECY et Stéphane VIOLLET. « Agile Robotic Fliers : A Morphing-Based Approach ». en. In : *Soft Robotics* 5.5 (oct. 2018), p. 541-553. ISSN : 2169-5172, 2169-5180. DOI : 10.1089/soro.2017.0120<sup>1</sup>.
- [8] J. ESCARENO, A. SANCHEZ, O. GARCIA et al. « Triple Tilting Rotor Mini-UAV : Modeling and Embedded Control of the Attitude ». en. In : *2008 American Control Conference*. Seattle, WA : IEEE, juin 2008, p. 3476-3481. ISBN : 978-1-4244-2078-0. DOI : 10.1109/ACC.2008.4587031<sup>2</sup>.
- [9] Kaan T ONER, Ertugrul CETINSOY, Mustafa UNEL et al. « Dynamic Model and Control of a New Quadrotor Unmanned Aerial Vehicle with Tilt-Wing Mechanism ». en. In : (2008), p. 6.
- [10] Kaan T ONER, Ertugrul CETINSOY, Efe SIRIMOGLU et al. « LQR and SMC Stabilization of a New Unmanned Aerial Vehicle ». en. In : (2009), p. 6.

---

1. <https://doi.org/10.1089/soro.2017.0120>

2. <https://doi.org/10.1109/ACC.2008.4587031>

- [11] Dario BRESCIANINI et Raffaello D'ANDREA. « Design, Modeling and Control of an Omni-Directional Aerial Vehicle ». en. In : *2016 IEEE International Conference on Robotics and Automation (ICRA)*. Stockholm, Sweden : IEEE, mai 2016, p. 3261-3266. ISBN : 978-1-4673-8026-3. DOI : 10.1109/ICRA.2016.7487497 <sup>3</sup>.
- [12] Markus RYLL, Giuseppe MUSCIO, Francesco PIERRI et al. « 6D Physical Interaction with a Fully Actuated Aerial Robot ». en. In : *2017 IEEE International Conference on Robotics and Automation (ICRA)*. Singapore, Singapore : IEEE, mai 2017, p. 5190-5195. ISBN : 978-1-5090-4633-1. DOI : 10.1109/ICRA.2017.7989608 <sup>4</sup>.
- [13] Sangyul PARK, Jongbeom HER, Juhyeok KIM et al. « Design, Modeling and Control of Omni-Directional Aerial Robot ». en. In : *2016 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. Daejeon, South Korea : IEEE, oct. 2016, p. 1570-1575. ISBN : 978-1-5090-3762-9. DOI : 10.1109/IROS.2016.7759254 <sup>5</sup>.
- [14] Nicolas STAUB, Davide BICEGO, Quentin SABLE et al. « Towards a Flying Assistant Paradigm : The OTHex ». en. In : *2018 IEEE International Conference on Robotics and Automation (ICRA)*. Brisbane, QLD : IEEE, mai 2018, p. 6997-7002. ISBN : 978-1-5386-3081-5. DOI : 10.1109/ICRA.2018.8460877 <sup>6</sup>.
- [15] Anibal OLLERO, Juan CORTES, Angel SANTAMARIA-NAVARRO et al. « The AE-ROARMS Project : Aerial Robots with Advanced Manipulation Capabilities for Inspection and Maintenance ». en. In : *IEEE Robotics & Automation Magazine* 25.4 (déc. 2018), p. 12-23. ISSN : 1070-9932, 1558-223X. DOI : 10.1109/MRA.2018.2852789 <sup>7</sup>.
- [16] Emre YILMAZ, Hammad ZAKI et Mustafa UNEL. « Nonlinear Adaptive Control of an Aerial Manipulation System ». en. In : *2019 18th European Control Conference (ECC)*. Naples, Italy : IEEE, juin 2019, p. 3916-3921. ISBN : 978-3-907144-00-8. DOI : 10.23919/ECC.2019.8795709 <sup>8</sup>.
- [17] Nicolas STAUB, Mostafa MOHAMMADI, Davide BICEGO et al. « Towards Robotic MAGMaS : Multiple Aerial-Ground Manipulator Systems ». en. In : *2017 IEEE International Conference on Robotics and Automation (ICRA)*. Singapore, Singapore : IEEE, mai 2017, p. 1307-1312. ISBN : 978-1-5090-4633-1. DOI : 10.1109/ICRA.2017.7989154 <sup>9</sup>.
- [18] Nicolas STAUB, Mostafa MOHAMMADI, Davide BICEGO et al. « The Tele-MAGMaS : An Aerial-Ground Comanipulator System ». en. In : *IEEE Robotics & Automation Magazine* 25.4 (déc. 2018), p. 66-75. ISSN : 1070-9932, 1558-223X. DOI : 10.1109/MRA.2018.2871344 <sup>10</sup>.
- [19] Dimitris CHAIKALIS, Farshad KHORRAMI et Anthony TZES. « Adaptive Control Approaches for an Unmanned Aerial Manipulation System ». en. In : *Aircraft Systems* (2020), p. 10.

---

3. <https://doi.org/10.1109/ICRA.2016.7487497>

4. <https://doi.org/10.1109/ICRA.2017.7989608>

5. <https://doi.org/10.1109/IROS.2016.7759254>

6. <https://doi.org/10.1109/ICRA.2018.8460877>

7. <https://doi.org/10.1109/MRA.2018.2852789>

8. <https://doi.org/10.23919/ECC.2019.8795709>

9. <https://doi.org/10.1109/ICRA.2017.7989154>

10. <https://doi.org/10.1109/MRA.2018.2871344>

- [20] G. ARLEO, F. CACCAVALE, G. MUSCIO et al. « Control of Quadrotor Aerial Vehicles Equipped with a Robotic Arm ». en. In : *21st Mediterranean Conference on Control and Automation*. Platanias, Chania - Crete, Greece : IEEE, juin 2013, p. 1174-1180. ISBN : 978-1-4799-0997-1 978-1-4799-0995-7. DOI : 10.1109/MED.2013.6608869<sup>11</sup>.
- [21] Markus RYLL, Davide BICEGO et Antonio FRANCHI. « A Truly Redundant Aerial Manipulator Exploiting a Multi-Directional Thrust Base ». en. In : *IFAC-PapersOnLine* 51.22 (2018), p. 138-143. ISSN : 24058963. DOI : 10.1016/j.ifacol.2018.11.531<sup>12</sup>.
- [22] Yuri S. SARKISOV, Min Jun KIM, Davide BICEGO et al. « Development of SAM : Cable-Suspended Aerial Manipulator \* ». en. In : *2019 International Conference on Robotics and Automation (ICRA)*. Montreal, QC, Canada : IEEE, mai 2019, p. 5323-5329. ISBN : 978-1-5386-6027-0. DOI : 10.1109/ICRA.2019.8793592<sup>13</sup>.
- [23] Mina KAMEL, Sebastian VERLING, Omar ELKHATIB et al. « Voliro : An Omnidirectional Hexacopter With Tilttable Rotors ». en. In : *IEEE Robotics & Automation Magazine* 25.4 (déc. 2018), p. 34-44. ISSN : 1070-9932, 1558-223X. DOI : 10.1109/MRA.2018.2866758<sup>14</sup>. arXiv : 1801.04581<sup>15</sup>.
- [24] Runit KUMAR, Alireza NEMATI, Manish KUMAR et al. « Tilting-Rotor Quadcopter for Aggressive Flight Maneuvers Using Differential Flatness Based Flight Controller ». en. In : *Volume 3 : Vibration in Mechanical Systems; Modeling and Validation; Dynamic Systems and Control Education; Vibrations and Control of Systems; Modeling and Estimation for Vehicle Safety and Integrity; Modeling and Control of IC Engines and Aftertreatment Systems; Unmanned Aerial Vehicles (UAVs) and Their Applications; Dynamics and Control of Renewable Energy Systems; Energy Harvesting; Control of Smart Buildings and Microgrids; Energy Systems*. Tysons, Virginia, USA : American Society of Mechanical Engineers, oct. 2017, V003T39A006. ISBN : 978-0-7918-5829-5. DOI : 10.1115/DSCC2017-5241<sup>16</sup>.
- [25] Atsushi OOSEDO, Satoko ABIKO, Shota NARASAKI et al. « Flight Control Systems of a Quad Tilt Rotor Unmanned Aerial Vehicle for a Large Attitude Change ». en. In : *2015 IEEE International Conference on Robotics and Automation (ICRA)*. Seattle, WA, USA : IEEE, mai 2015, p. 2326-2331. ISBN : 978-1-4799-6923-4. DOI : 10.1109/ICRA.2015.7139508<sup>17</sup>.
- [26] Markus MÜLLER. *Eclac*. <https://www.ecalc.ch/index.htm>.
- [27] *Elistair*. <https://elistair.com/>. URL : <https://www.ecalc.ch/index.htm>.
- [28] Hubert HAHN. *Rigid Body Dynamics of Mechanisms*. Berlin New York : Springer, 2002. ISBN : 3-540-42373-7.
- [29] Herbert GOLDSTEIN. *Classical Mechanics*. San Francisco : Addison Wesley, 2002. ISBN : 978-0-201-31611-7.
- [30] P. CASTILLO, R. LOZANO et Alejandro E. DZUL. *Modelling and Control of Mini-Flying Machines*. en. *Advances in Industrial Control*. London ; New York : Springer, 2005. ISBN : 978-1-85233-957-9.

11. <https://doi.org/10.1109/MED.2013.6608869>

12. <https://doi.org/10.1016/j.ifacol.2018.11.531>

13. <https://doi.org/10.1109/ICRA.2019.8793592>

14. <https://doi.org/10.1109/MRA.2018.2866758>

15. <https://arxiv.org/abs/1801.04581>

16. <https://doi.org/10.1115/DSCC2017-5241>

17. <https://doi.org/10.1109/ICRA.2015.7139508>

- [31] Isabelle FANTONI, Rogelio LOZANO et Pedro CASTILLO. « A SIMPLE STABILIZATION ALGORITHM FOR THE PVTOL AIRCRAFT ». en. In : *IFAC Proceedings Volumes* 35.1 (2002), p. 7-11. ISSN : 14746670. DOI : 10.3182/20020721-6-ES-1901.01227<sup>18</sup>.
- [32] P Castillo GARCÍA. « Nonlinear Model of the PVTOL Aircraft ». en. In : (), p. 7.
- [33] Isabelle FANTONI. « Modelling of Terrestrial and Aerial Vehicles ». en. In : (), p. 18.
- [34] P CASTILLO et R LOZANO. « Dynamic Model of a Quad-Rotor Rotorcraft ». en. In : (), p. 16.
- [35] Luis Rodolfo GARCÍA CARRILLO, Alejandro Enrique DzUL LÓPEZ, Rogelio LOZANO et al. « Modeling the Quad-Rotor Mini-Rotorcraft ». en. In : *Quad Rotorcraft Control*. London : Springer London, 2013, p. 23-34. ISBN : 978-1-4471-4398-7 978-1-4471-4399-4. DOI : 10.1007/978-1-4471-4399-4\_2<sup>19</sup>.
- [36] P. POUNDS, R. MAHONY et P. CORKE. « Modelling and Control of a Large Quadrotor Robot ». en. In : *Control Engineering Practice* 18.7 (juill. 2010), p. 691-699. ISSN : 09670661. DOI : 10.1016/j.conengprac.2010.02.008<sup>20</sup>.
- [37] Paul POUNDS, Robert MAHONY et Peter CORKE. « Modelling and Control of a Quad-Rotor Robot ». en. In : (), p. 11.
- [38] Rejane Cavalcante SA, Andre Luiz C. de ARAUJO, Antonio T. VARELA et al. « Construction and PID Control for Stability of an Unmanned Aerial Vehicle of the Type Quadrotor ». en. In : *2013 Latin American Robotics Symposium and Competition*. Arequipa, Peru : IEEE, oct. 2013, p. 95-99. ISBN : 978-0-7695-5139-5. DOI : 10.1109/LARS.2013.64<sup>21</sup>.
- [39] Pierre AGATI. *Mécanique Du Solide : Applications Industrielles : [Cours et Exercices Corrigés]*. Paris : Dunod, 2003. ISBN : 978-2-10-007945-2.
- [40] *Effet Gyroscopique*. <https://sensphysique.com/mecanique-du-solide/effet-gyroscopique/>.
- [41] Juan ESCARENO, Anand SANCHEZ et Octavio GARCIA. « Control of Different UAVs with Tilting Rotors ». en. In : *Unmanned Aerial Vehicles*. Sous la dir. de Rogelio LOZANO. Hoboken, NJ, USA : John Wiley & Sons, Inc., fév. 2013, p. 115-132. ISBN : 978-1-118-59993-8 978-1-84821-127-8. DOI : 10.1002/9781118599938.ch6<sup>22</sup>.
- [42] D. KASTELAN, M. KONZ et J. RUDOLPH. « Fully Actuated Tricopter with Pilot-Supporting Control\*\*This Work Is Supported in Part by the German Research Foundation (DFG) in the Framework of Project “Regelung von Trikopfern Beim Gemeinsamen Transport von an Seilen Aufgehängten Lasten”. » en. In : *IFAC-PapersOnLine* 48.9 (2015), p. 79-84. ISSN : 24058963. DOI : 10.1016/j.ifacol.2015.08.063<sup>23</sup>.
- [43] Rमित KUMAR, Siddharth SRIDHAR, Franck CAZAURANG et al. « Reconfigurable Fault-Tolerant Tilt-Rotor Quadcopter System ». en. In : *Volume 3 : Modeling and Validation; Multi-Agent and Networked Systems; Path Planning and Motion Control; Tracking Control Systems; Unmanned Aerial Vehicles (UAVs) and Application; Unmanned Ground and Aerial Vehicles; Vibration in Mechanical Systems; Vibrations and Control of Systems; Vibrations : Modeling, Analysis, and Control*.

18. <https://doi.org/10.3182/20020721-6-ES-1901.01227>

19. [https://doi.org/10.1007/978-1-4471-4399-4\\_2](https://doi.org/10.1007/978-1-4471-4399-4_2)

20. <https://doi.org/10.1016/j.conengprac.2010.02.008>

21. <https://doi.org/10.1109/LARS.2013.64>

22. <https://doi.org/10.1002/9781118599938.ch6>

23. <https://doi.org/10.1016/j.ifacol.2015.08.063>

- Atlanta, Georgia, USA : American Society of Mechanical Engineers, sept. 2018, V003T37A008. ISBN : 978-0-7918-5191-3. DOI : 10.1115/DSCC2018-9197<sup>24</sup>.
- [44] A. NEMATI et M. KUMAR. « Modeling and Control of a Single Axis Tilting Quadcopter ». en. In : *2014 American Control Conference*. Portland, OR, USA : IEEE, juin 2014, p. 3077-3082. ISBN : 978-1-4799-3274-0 978-1-4799-3272-6 978-1-4799-3271-9. DOI : 10.1109/ACC.2014.6859328<sup>25</sup>.
- [45] Markus RYLL, Heinrich H. BULTHOFF et Paolo Robuffo GIORDANO. « A Novel Overactuated Quadrotor Unmanned Aerial Vehicle : Modeling, Control, and Experimental Validation ». en. In : *IEEE Transactions on Control Systems Technology* 23.2 (mars 2015), p. 540-556. ISSN : 1063-6536, 1558-0865. DOI : 10.1109/TCST.2014.2330999<sup>26</sup>.
- [46] Markus RYLL, Heinrich H. BULTHOFF et Paolo Robuffo GIORDANO. « First Flight Tests for a Quadrotor UAV with Tilting Propellers ». en. In : *2013 IEEE International Conference on Robotics and Automation*. Karlsruhe, Germany : IEEE, mai 2013, p. 295-302. ISBN : 978-1-4673-5643-5 978-1-4673-5641-1. DOI : 10.1109/ICRA.2013.6630591<sup>27</sup>.
- [47] Markus RYLL, Heinrich H. BULTHOFF et Paolo Robuffo GIORDANO. « Modeling and Control of a Quadrotor UAV with Tilting Propellers ». en. In : *2012 IEEE International Conference on Robotics and Automation*. St Paul, MN, USA : IEEE, mai 2012, p. 4606-4613. ISBN : 978-1-4673-1405-3 978-1-4673-1403-9 978-1-4673-1578-4 978-1-4673-1404-6. DOI : 10.1109/ICRA.2012.6225129<sup>28</sup>.
- [48] Fatih SENKUL et Erdinc ALTUG. « Modeling and Control of a Novel Tilt & Roll Rotor Quadrotor UAV ». en. In : *2013 International Conference on Unmanned Aircraft Systems (ICUAS)*. Atlanta, GA, USA : IEEE, mai 2013, p. 1071-1076. ISBN : 978-1-4799-0817-2 978-1-4799-0815-8 978-1-4799-0816-5. DOI : 10.1109/ICUAS.2013.6564796<sup>29</sup>.
- [49] Siddharth SRIDHAR, Gaurang GUPTA, Runit KUMAR et al. « Tilt-Rotor Quadcopter Explored : Hardware Based Dynamics, Smart Sliding Mode Controller, Attitude Hold & Wind Disturbance Scenarios ». en. In : *2019 American Control Conference (ACC)*. Philadelphia, PA, USA : IEEE, juill. 2019, p. 2005-2010. ISBN : 978-1-5386-7926-5. DOI : 10.23919/ACC.2019.8815013<sup>30</sup>.
- [50] Omar ELKHATIB. « Control Allocation of a Tilting Rotor Hexacopter ». en. In : (2017), 44 p. DOI : 10.3929/ETHZ-B-000224598<sup>31</sup>.
- [51] Sujit RAJAPPA, Markus RYLL, Heinrich H. BULTHOFF et al. « Modeling, Control and Design Optimization for a Fully-Actuated Hexarotor Aerial Vehicle with Tilted Propellers ». en. In : *2015 IEEE International Conference on Robotics and Automation (ICRA)*. Seattle, WA, USA : IEEE, mai 2015, p. 4006-4013. ISBN : 978-1-4799-6923-4. DOI : 10.1109/ICRA.2015.7139759<sup>32</sup>.
- [52] Ali JUNAID, Alejandro SANCHEZ, Javier BOSCH et al. « Design and Implementation of a Dual-Axis Tilting Quadcopter ». en. In : *Robotics 7.4* (oct. 2018), p. 65. ISSN : 2218-6581. DOI : 10.3390/robotics7040065<sup>33</sup>.

24. <https://doi.org/10.1115/DSCC2018-9197>

25. <https://doi.org/10.1109/ACC.2014.6859328>

26. <https://doi.org/10.1109/TCST.2014.2330999>

27. <https://doi.org/10.1109/ICRA.2013.6630591>

28. <https://doi.org/10.1109/ICRA.2012.6225129>

29. <https://doi.org/10.1109/ICUAS.2013.6564796>

30. <https://doi.org/10.23919/ACC.2019.8815013>

31. <https://doi.org/10.3929/ETHZ-B-000224598>

32. <https://doi.org/10.1109/ICRA.2015.7139759>

33. <https://doi.org/10.3390/robotics7040065>



- [53] Mahmoud ELFEKY, Moustafa ELSHAFAEI, Abdul-Wahid A. SAIF et al. « Modeling and Simulation of Quadrotor UAV with Tilting Rotors ». en. In : *International Journal of Control, Automation and Systems* 14.4 (août 2016), p. 1047-1055. ISSN : 1598-6446, 2005-4092. DOI : 10.1007/s12555-015-0064-5<sup>34</sup>.
- [54] Shlok AGARWAL, Apoorva MOHAN et Kamlesh KUMAR. « Design and Fabrication of Twinrotor UAV ». en. In : *Computer Science & Information Technology (CS & IT)*. Academy & Industry Research Collaboration Center (AIRCC), nov. 2013, p. 369-377. ISBN : 978-1-921987-15-1. DOI : 10.5121/cs.it.2013.3830<sup>35</sup>.
- [55] Giovanni Ortega VARGAS, Christoph HINTZ, Luis Rodolfo Garcia CARRILLO et al. « Dynamic Modeling of a Multi-Rotorcraft UAS with Morphing Capabilities ». en. In : *2015 International Conference on Unmanned Aircraft Systems (ICUAS)*. Denver, CO, USA : IEEE, juin 2015, p. 963-971. ISBN : 978-1-4799-6010-1. DOI : 10.1109/ICUAS.2015.7152385<sup>36</sup>.
- [56] Marcin ODELGA, Paolo STEGAGNO et Heinrich H. BULTHOFF. « A Fully Actuated Quadrotor UAV with a Propeller Tilting Mechanism : Modeling and Control ». en. In : *2016 IEEE International Conference on Advanced Intelligent Mechatronics (AIM)*. Banff, AB, Canada : IEEE, juill. 2016, p. 306-311. ISBN : 978-1-5090-2065-2. DOI : 10.1109/AIM.2016.7576784<sup>37</sup>.
- [57] Arindam BHANJA CHOWDHURY, Anil KULHARE et Gaurav RAINA. « A Generalized Control Method for a Tilt-Rotor UAV Stabilization ». en. In : *2012 IEEE International Conference on Cyber Technology in Automation, Control, and Intelligent Systems (CYBER)*. Bangkok : IEEE, mai 2012, p. 309-314. ISBN : 978-1-4673-1421-3 978-1-4673-1420-6 978-1-4673-1419-0. DOI : 10.1109/CYBER.2012.6392571<sup>38</sup>.
- [58] Anand SANCHEZ, Juan ESCARENO et Octavio GARCIA. « Autonomous Hovering of a Two-Rotor UAV ». en. In : *Unmanned Aerial Vehicles*. Sous la dir. de Rogelio LOZANO. Hoboken, NJ, USA : John Wiley & Sons, Inc., fév. 2013, p. 59-78. ISBN : 978-1-118-59993-8 978-1-84821-127-8. DOI : 10.1002/9781118599938.ch4<sup>39</sup>.
- [59] Gary GRESS. « Using Dual Propellers as Gyroscopes for Tilt-Prop Hover Control ». en. In : *2002 Biennial International Powered Lift Conference and Exhibit*. Williamsburg, Virginia : American Institute of Aeronautics and Astronautics, nov. 2002. ISBN : 978-1-62410-109-0. DOI : 10.2514/6.2002-5968<sup>40</sup>.
- [60] Pedro CASTILLO GARCIA, Laura Elena MUNOZ HERNANDEZ et Pedro GARCIA GIL. *Indoor Navigation Strategies for Aerial Autonomous Systems*. Sous la dir. d'ELSEVIER. Fév. 2017.
- [61] A. OFFERMANN, P. CASTILLO et J. De MIRAS. « Control of a PVTOL \* with Tilting Rotors \* ». en. In : *2019 International Conference on Unmanned Aircraft Systems (ICUAS)*. Atlanta, GA, USA : IEEE, juin 2019, p. 1451-1457. ISBN : 978-1-72810-333-4. DOI : 10.1109/ICUAS.2019.8798341<sup>41</sup>.
- [62] Santosh DEVASIA. « A Different Look at Output Tracking : Control of a VTOL Aircraft ». en. In : (), p. 7.

34. <https://doi.org/10.1007/s12555-015-0064-5>

35. <https://doi.org/10.5121/cs.it.2013.3830>

36. <https://doi.org/10.1109/ICUAS.2015.7152385>

37. <https://doi.org/10.1109/AIM.2016.7576784>

38. <https://doi.org/10.1109/CYBER.2012.6392571>

39. <https://doi.org/10.1002/9781118599938.ch4>

40. <https://doi.org/10.2514/6.2002-5968>

41. <https://doi.org/10.1109/ICUAS.2019.8798341>

- [63] H. SIRA-RAMFREZ. « Non-Minimum Phase Output Reference Trajectory Tracking for a PVTOL Aircraft ». In : *Proceedings of the 2000. IEEE International Conference on Control Applications. Conference Proceedings (Cat. No.00CH37162)*. Anchorage, AK, USA : IEEE, 2000, p. 838-843. ISBN : 978-0-7803-6562-9. DOI : 10.1109/CCA.2000.897544 <sup>42</sup>.
- [64] A. SANCHEZ, P. CASTILLO, J. ESCARENO et al. « Simple Real-Time Control Strategy to Stabilize the PVTOL Aircraft Using Bounded Inputs ». en. In : *2007 European Control Conference (ECC)*. Kos : IEEE, juill. 2007, p. 3486-3491. ISBN : 978-3-9524173-8-6. DOI : 10.23919/ECC.2007.7068815 <sup>43</sup>.
- [65] Daniela Juanita LÓPEZ-ARAUJO, Arturo ZAVALA-RÍO, Isabelle FANTONI et al. « Global Stabilization of the PVTOL Aircraft with Lateral Force Coupling and Bounded Inputs ». en. In : (), p. 27.
- [66] I. FANTONI, R. LOZANO et A. PALOMINO. « Global Stabilizing Control Design for the PVTOL Aircraft Using Saturation Functions on the Inputs ». en. In : *2003 European Control Conference (ECC)*. Cambridge, UK : IEEE, sept. 2003, p. 1810-1815. ISBN : 978-3-9524173-7-9. DOI : 10.23919/ECC.2003.7085228 <sup>44</sup>.
- [67] I. FANTONI, A. ZAVALA et R. LOZANO. « Global Stabilization of a PVTOL Aircraft with Bounded Thrust ». en. In : *Proceedings of the 41st IEEE Conference on Decision and Control, 2002*. T. 4. Las Vegas, NV, USA : IEEE, 2002, p. 4462-4467. ISBN : 978-0-7803-7516-1. DOI : 10.1109/CDC.2002.1185076 <sup>45</sup>.
- [68] Samer RIACHY, Michel FLIESS, Cédric JOIN et al. « Vers une simplification de la commande non linéaire : l'exemple d'un avion à décollage vertical ». fr. In : (), p. 7.
- [69] R. OLFATI-SABER. « Global Configuration Stabilization for the VTOL Aircraft with Strong Input Coupling ». en. In : *IEEE Transactions on Automatic Control* 47.11 (nov. 2002), p. 1949-1952. ISSN : 0018-9286. DOI : 10.1109/TAC.2002.804457 <sup>46</sup>.
- [70] J Karl HEDRICK, Anouck GIRARD, J K HEDRICK et al. « Control of Nonlinear Dynamic Systems ». en. In : (2015), p. 284.
- [71] Andrew R. TEEL. « Global Stabilization and Restricted Tracking for Multiple Integrators with Bounded Controls ». en. In : *Systems & Control Letters* 18.3 (mars 1992), p. 165-171. ISSN : 01676911. DOI : 10.1016/0167-6911(92)90001-9 <sup>47</sup>.
- [72] P CASTILLO, A DZUL et R LOZANO. « Real-Time Stabilization and Tracking of a Four Rotor Mini-Rotorcraft ». en. In : (), p. 6.
- [73] Michael W OPPENHEIMER. « Control Allocation for Overactuated Systems ». en. In : (), p. 9.
- [74] Lifeng WANG, Yichong HE, Zhixiang ZHANG et al. « Trajectory Tracking of Quadrotor Aerial Robot Using Improved Dynamic Inversion Method ». en. In : *Intelligent Control and Automation* 04.04 (2013), p. 343-348. ISSN : 2153-0653, 2153-0661. DOI : 10.4236/ica.2013.44040 <sup>48</sup>.

---

42. <https://doi.org/10.1109/CCA.2000.897544>

43. <https://doi.org/10.23919/ECC.2007.7068815>

44. <https://doi.org/10.23919/ECC.2003.7085228>

45. <https://doi.org/10.1109/CDC.2002.1185076>

46. <https://doi.org/10.1109/TAC.2002.804457>

47. [https://doi.org/10.1016/0167-6911\(92\)90001-9](https://doi.org/10.1016/0167-6911(92)90001-9)

48. <https://doi.org/10.4236/ica.2013.44040>

- [75] Minh NGUYEN DUC, Thang Nguyen TRONG et Yang Sheng XUAN. « The Quadrotor MAV System Using PID Control ». en. In : *2015 IEEE International Conference on Mechatronics and Automation (ICMA)*. Beijing, China : IEEE, août 2015, p. 506-510. ISBN : 978-1-4799-7097-1 978-1-4799-7098-8. DOI : 10.1109/ICMA.2015.7237537<sup>49</sup>.
- [76] Emre Can SUICMEZ et Ali Turker KUTAY. « Optimal Path Tracking Control of a Quadrotor UAV ». en. In : *2014 International Conference on Unmanned Aircraft Systems (ICUAS)*. Orlando, FL, USA : IEEE, mai 2014, p. 115-125. ISBN : 978-1-4799-2376-2. DOI : 10.1109/ICUAS.2014.6842246<sup>50</sup>.
- [77] Holger Voos. « Nonlinear Control of a Quadrotor Micro-UAV Using Feedback-Linearization ». en. In : *2009 IEEE International Conference on Mechatronics*. Malaga, Spain : IEEE, 2009, p. 1-6. ISBN : 978-1-4244-4194-5. DOI : 10.1109/ICMECH.2009.4957154<sup>51</sup>.
- [78] Yongeun YOON, Eric N. JOHNSON et Liling REN. « Autonomous Flight Control for Multirotors by a Simple Input-Output Linearization with Nested Saturation ». en. In : *2018 IEEE Conference on Control Technology and Applications (CCTA)*. Copenhagen : IEEE, août 2018, p. 355-360. ISBN : 978-1-5386-7698-1. DOI : 10.1109/CCTA.2018.8511557<sup>52</sup>.
- [79] Markus RYLL, Davide BICEGO et Antonio FRANCHI. « Modeling and Control of FAST-Hex : A Fully-Actuated by Synchronized-Tilting Hexarotor ». en. In : *2016 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. Daejeon, South Korea : IEEE, oct. 2016, p. 1689-1694. ISBN : 978-1-5090-3762-9. DOI : 10.1109/IROS.2016.7759271<sup>53</sup>.
- [80] Emil FRESK et George NIKOLAKOPOULOS. « Full Quaternion Based Attitude Control for a Quadrotor ». en. In : *2013 European Control Conference (ECC)*. Zurich : IEEE, juill. 2013, p. 3864-3869. ISBN : 978-3-033-03962-9. DOI : 10.23919/ECC.2013.6669617<sup>54</sup>.
- [81] Taeyoung LEE, Melvin LEOK et N. Harris McCLAMROCH. « Geometric Tracking Control of a Quadrotor UAV on SE(3) ». en. In : *49th IEEE Conference on Decision and Control (CDC)*. Atlanta, GA : IEEE, déc. 2010, p. 5420-5425. ISBN : 978-1-4244-7745-6 978-1-4244-7746-3. DOI : 10.1109/CDC.2010.5717652<sup>55</sup>.
- [82] Robert MAHONY, Tarek HAMEL et Jean-Michel PFLIMLIN. « Nonlinear Complementary Filters on the Special Orthogonal Group ». en. In : *IEEE Transactions on Automatic Control* 53.5 (juin 2008), p. 1203-1218. ISSN : 0018-9286. DOI : 10.1109/TAC.2008.923738<sup>56</sup>.
- [83] Simone LUDWIG, Kaleb BURNHAM, Antonio JIMÉNEZ et al. « Comparison of Attitude and Heading Reference Systems Using Foot Mounted MIMU Sensor Data : Basic, Madgwick, and Mahony ». In : mars 2018, p. 96. DOI : 10.1117/12.2296568<sup>57</sup>.

49. <https://doi.org/10.1109/ICMA.2015.7237537>

50. <https://doi.org/10.1109/ICUAS.2014.6842246>

51. <https://doi.org/10.1109/ICMECH.2009.4957154>

52. <https://doi.org/10.1109/CCTA.2018.8511557>

53. <https://doi.org/10.1109/IROS.2016.7759271>

54. <https://doi.org/10.23919/ECC.2013.6669617>

55. <https://doi.org/10.1109/CDC.2010.5717652>

56. <https://doi.org/10.1109/TAC.2008.923738>

57. <https://doi.org/10.1117/12.2296568>

- [84] Krittaya RUNCHAROON et Varawan SRICHATRAPIMUK. « Sliding Mode Control of Quadrotor ». en. In : *2013 The International Conference on Technological Advances in Electrical, Electronics and Computer Engineering (TAECE)*. Konya, Turkey : IEEE, mai 2013, p. 552-557. ISBN : 978-1-4673-5613-8 978-1-4673-5612-1 978-1-4673-5611-4. DOI : 10.1109/TAECE.2013.6557334<sup>58</sup>.
- [85] Marco HERRERA, Alejandro P GÓMEZ, William CHAMORRO et al. « Sliding Mode Control : An Approach to Control a Quadrotor ». en. In : (), p. 6.
- [86] Chih-Chen YIH. « Flight Control of a Tilt-Rotor Quadcopter via Sliding Mode ». en. In : *2016 International Automatic Control Conference (CACS)*. Taichung : IEEE, nov. 2016, p. 65-70. ISBN : 978-1-5090-4108-4. DOI : 10.1109/CACS.2016.7973885<sup>59</sup>.
- [87] Siddharth SRIDHAR, Runit KUMAR, Mohammadreza RADMANESH et al. « Non-Linear Sliding Mode Control of a Tilting-Rotor Quadcopter ». en. In : *Volume 1 : Aerospace Applications; Advances in Control Design Methods; Bio Engineering Applications; Advances in Non-Linear Control; Adaptive and Intelligent Systems Control; Advances in Wind Energy Systems; Advances in Robotics; Assistive and Rehabilitation Robotics; Biomedical and Neural Systems Modeling, Diagnostics, and Control; Bio-Mechatronics and Physical Human Robot; Advanced Driver Assistance Systems and Autonomous Vehicles; Automotive Systems*. Tysons, Virginia, USA : American Society of Mechanical Engineers, oct. 2017, V001T09A007. ISBN : 978-0-7918-5827-1. DOI : 10.1115/DSCC2017-5375<sup>60</sup>.
- [88] Siddharth SRIDHAR, Runit KUMAR, Kelly COHEN et al. « Fault Tolerance of a Reconfigurable Tilt-Rotor Quadcopter Using Sliding Mode Control ». en. In : *Volume 3 : Modeling and Validation; Multi-Agent and Networked Systems; Path Planning and Motion Control; Tracking Control Systems; Unmanned Aerial Vehicles (UAVs) and Application; Unmanned Ground and Aerial Vehicles; Vibration in Mechanical Systems; Vibrations and Control of Systems; Vibrations : Modeling, Analysis, and Control*. Atlanta, Georgia, USA : American Society of Mechanical Engineers, sept. 2018, V003T37A009. ISBN : 978-0-7918-5191-3. DOI : 10.1115/DSCC2018-9199<sup>61</sup>.
- [89] Luis RUBIO, Asier IBEAS et Xichun LUO. « P-PI and Super Twisting Sliding Mode Control Schemes Comparison for High-Precision CNC Machining ». en. In : *2016 24th Iranian Conference on Electrical Engineering (ICEE)*. Shiraz, Iran : IEEE, mai 2016, p. 1825-1830. ISBN : 978-1-4673-8789-7. DOI : 10.1109/IranianCEE.2016.7585818<sup>62</sup>.
- [90] H.J. JAYAKRISHNAN. « Position and Attitude Control of a Quadrotor UAV Using Super Twisting Sliding Mode ». en. In : *IFAC-PapersOnLine* 49.1 (2016), p. 284-289. ISSN : 24058963. DOI : 10.1016/j.ifacol.2016.03.067<sup>63</sup>.
- [91] Van Truong HOANG et Quang Hieu PHAM. « Adaptive Super-Twisting Second-Order Sliding Mode for Attitude Control of Quadcopter UAVs ». en. In : (), p. 7.
- [92] Reesa AKBAR et Naoki UCHIYAMA. « Adaptive Modified Super-Twisting Control for a Quadrotor Helicopter with a Nonlinear Sliding Surface ». en. In : (), p. 6.

58. <https://doi.org/10.1109/TAECE.2013.6557334>

59. <https://doi.org/10.1109/CACS.2016.7973885>

60. <https://doi.org/10.1115/DSCC2017-5375>

61. <https://doi.org/10.1115/DSCC2018-9199>

62. <https://doi.org/10.1109/IranianCEE.2016.7585818>

63. <https://doi.org/10.1016/j.ifacol.2016.03.067>

- [93] Jaime A. MORENO et Marisol OSORIO. « A Lyapunov Approach to Second-Order Sliding Mode Controllers and Observers ». en. In : *2008 47th IEEE Conference on Decision and Control*. Cancun, Mexico : IEEE, 2008, p. 2856-2861. ISBN : 978-1-4244-3123-6. DOI : 10.1109/CDC.2008.4739356<sup>64</sup>.
- [94] Chaoxu MU, Changyin SUN, Chengshan QIAN et al. « Super-Twisting Sliding Mode Control Based on Lyapunov Analysis for the Cursing Flight of Hypersonic Vehicles ». en. In : *2013 10th IEEE International Conference on Control and Automation (ICCA)*. Hangzhou, China : IEEE, juin 2013, p. 522-527. ISBN : 978-1-4673-4708-2 978-1-4673-4707-5 978-1-4673-4706-8. DOI : 10.1109/ICCA.2013.6565017<sup>65</sup>.
- [95] Bernard ETKIN et T. TEICHMANN. « *Dynamics of Flight : Stability and Control* ». en. In : *Physics Today* 12.9 (sept. 1959), p. 54-56. ISSN : 0031-9228, 1945-0699. DOI : 10.1063/1.3060977<sup>66</sup>.
- [96] Armando S. SANCA, Pablo J. ALSINA et J s de Jesus F. CERQUEIRA. « Dynamic Modelling of a Quadrotor Aerial Vehicle with Nonlinear Inputs ». en. In : *2008 IEEE Latin American Robotic Symposium*. Salvador, Bahia, Brazil : IEEE, oct. 2008, p. 143-148. ISBN : 978-1-4244-3379-7. DOI : 10.1109/LARS.2008.17<sup>67</sup>.
- [97] T CRUL. « Etude de la propulsion d'un drone ». fr. In : (), p. 15.
- [98] Moses BANGURA et Robert MAHONY. « Nonlinear Dynamic Modeling for High Performance Control of a Quadrotor ». en. In : *New Zealand*. (2012), p. 10.
- [99] Jian SUN. « Pulse-Width Modulation ». In : *Dynamics and Control of Switched Electronic Systems : Advanced Perspectives for Modeling, Simulation and Control of Power Converters*. Sous la dir. de Francesco VASCA et Luigi IANNELLI. London : Springer London, 2012, p. 25-61. ISBN : 978-1-4471-2885-4. DOI : 10.1007/978-1-4471-2885-4   <sup>68</sup>.
- [100] Augustin MANECY, Nicolas MARCHAND, Franck RUFFIER et al. « X4-MaG : A Low-Cost Open-Source Micro-Quadrotor and Its Linux-Based Controller ». en. In : *International Journal of Micro Air Vehicles* 7.2 (juin 2015), p. 89-109. ISSN : 1756-8293, 1756-8307. DOI : 10.1260/1756-8293.7.2.89<sup>69</sup>.
- [101] A. BERNA, P. CASTILLO, G. SANAHUJA et al. « Development of a Test-Bed to Implement and Validate Real-Time Control Strategies for Aerial Vehicles\* ». en. In : *IFAC Proceedings Volumes* 44.1 (jan. 2011), p. 7660-7665. ISSN : 14746670. DOI : 10.3182/20110828-6-IT-1002.02743<sup>70</sup>.
- [102] M. GALLI, R. BARBER, S. GARRIDO et al. « Path Planning Using Matlab-ROS Integration Applied to Mobile Robots ». en. In : *2017 IEEE International Conference on Autonomous Robot Systems and Competitions (ICARSC)*. Coimbra, Portugal : IEEE, avr. 2017, p. 98-103. ISBN : 978-1-5090-6234-8. DOI : 10.1109/ICARSC.2017.7964059<sup>71</sup>.
- [103] *Dynamixels Manual*. <http://www.crustcrawler.com/products/bioloid/docs/AX-12.pdf>.
- [104] *Dynamixels Workbench Tutorial*. [https://manual.robotis.com/docs/en/software/dynamixel/dynamixel\\_workbench/](https://manual.robotis.com/docs/en/software/dynamixel/dynamixel_workbench/).

64. <https://doi.org/10.1109/CDC.2008.4739356>

65. <https://doi.org/10.1109/ICCA.2013.6565017>

66. <https://doi.org/10.1063/1.3060977>

67. <https://doi.org/10.1109/LARS.2008.17>

68. <https://doi.org/10.1007/978-1-4471-2885-4   >

69. <https://doi.org/10.1260/1756-8293.7.2.89>

70. <https://doi.org/10.3182/20110828-6-IT-1002.02743>

71. <https://doi.org/10.1109/ICARSC.2017.7964059>

- [105] Laura E MUNOZ, Omar SANTOS et Pedro CASTILLO. « Robust Nonlinear Real-Time Control Strategy to Stabilize a PVTOL Aircraft in Crosswind ». en. In : *2010 IEEE/RSJ International Conference on Intelligent Robots and Systems*. Taipei : IEEE, oct. 2010, p. 1606-1611. ISBN : 978-1-4244-6674-0. DOI : 10.1109/IROS.2010.5654396<sup>72</sup>.
- [107] Hassan KHALIL. *Nonlinear Systems*. Upper Saddle River, New Jersey : Prentice Hall, 2002. ISBN : 0-13-067389-7.

---

72. <https://doi.org/10.1109/IROS.2010.5654396>

# Annexes

## Sommaire du présent chapitre

---

|          |   |            |
|----------|---|------------|
| <b>A</b> | <b>Formation, diffusion et travaux annexes</b>              | <b>161</b> |
| A.1      | Formations  | 161        |
| A.2      | Diffusion scientifique et salons                            | 161        |
| <b>B</b> | <b>AMDEC</b>  | <b>163</b> |
| B.1      | Criticité   | 163        |
| B.2      | Structure   | 164        |
| B.3      | Système embarqué  | 165        |
| <b>C</b> | <b>Précis mathématiques</b>                                 | <b>169</b> |
| C.1      | <b>Fonctions usuelles</b>                                   | <b>169</b> |
| C.1.1    | Fonction de saturation . . . . .                            | 169        |
| C.1.2    | Fonction signe . . . . .                                    | 169        |
| C.1.3    | Pseudoinverse de Moore-Penrose . . . . .                    | 169        |
| C.2      | <b>Preuves de stabilité</b>                                 | <b>170</b> |
| C.2.1    | Système linéaire . . . . .                                  | 170        |
| C.2.2    | Saturations imbriquées . . . . .                            | 170        |
| C.2.3    | Sliding mode super-twisting . . . . .                       | 172        |
| <b>D</b> | <b>Plateforme</b>   | <b>174</b> |
| D.1      | <b>Mesure de l'angle de rotation des servomoteurs</b>       | <b>174</b> |
| D.1.1    | Boîtier Arduino . . . . .                                   | 176        |
| D.1.2    | Accouplement entre bras et servomoteurs . . . . .           | 176        |
| D.2      | <b>Estimation de l'altitude</b>                             | <b>177</b> |
| D.3      | <b>Précisions d'utilisation</b>                             | <b>178</b> |
| D.3.1    | Installation du framework développé . . . . .               | 178        |
| D.3.2    | OptiTrack . . . . .   | 179        |
| D.3.3    | Dynamixels . . . . .  | 179        |
| D.3.4    | Problèmes et solutions rencontrés au cours de l'utilisation | 180        |

---

# Formation, diffusion et travaux annexes

## A.1 Formations

Dans le cadre de cette thèse, plusieurs formations et journées scientifiques ont été suivies :

- RobMooc, commande non linéaire avec L. Jaulin, Université de Bretagne occidentale
- Formation théorique ULM pour le télé-pilotage de drone
- Formation à l'éthique de la recherche et à l'intégrité scientifique.
- Séminaires de l'équipe

Les formations qui ont été suivies :

- TIS05, "Commande de robots autonomes en coopération", 32h
- TIS04, "Estimation pour la navigation robotique", 32h
- TOEIC, validation du niveau de compétence en langue anglaise

## A.2 Diffusion scientifique et salons

- A. Offermann, Conception et validation d'un ROBOT apte à effectuer des Mesures sur Ouvrages (C.Ro.Me.O), JRDA2018 : Journée Régionale de Doctorant en Automatique, 03 juillet 2018, Logis du Roy à Amiens, Poster
- Présentation des travaux lors de la journée des doctorants en septembre 2018, 2019 et 2020
- Article scientifique présenté à la conférence internationale MED 2020, voir [106]
- Article scientifique présenté à l'ICUAS 2019, voir [61]
- Article scientifique soumis à l'ICRA 2021
- Fêtes de la science de Compiègne 2018, 2019, et 2020. Vulgarisation des travaux devant un public jeune



- JNRR 2019 Journées Nationales de la Recherche en Robotique, Vittel, 2019 présentations top-actuels dans le domaine de la robotique
- SIDO, Lyon, 2019 salon sur l'Internet des objets et intelligence artificielle - Participation
- SIDO, Lyon, 2020 salon sur l'Internet des objets et intelligence artificielle - Tenue de stand

### A.3 Contributions

Parmi les travaux entrepris au sein de l'entreprise ayant un lien avec ce sujet de thèse on citera :

- Réparation du Léopard : ancien dispositif de mesure pacométrique et de potentiel sur parois. Le dispositif ressemble à un ascenseur que l'on vient fixer sur l'ouvrage afin de mesurer les fers et le potentiel électrique du béton. La modernisation consiste, d'une part, à changer le système d'alimentation filaire par des batteries. D'autre part, une adaptation est réalisée afin de pouvoir utiliser les capteurs actuels.
- Intégration de capteurs sur drone gros porteur en vue de faire une mesure de particules fines. La mission est réalisée à l'usine Cristal à Thann.
- Intégration de capteurs (différents) sur drone gros porteur en vue de faire une mesure de particule de boxaline au dessus de l'usine et zones de stockage d'Alteo à Gardanne.

# Annexe B

## AMDEC

Dans cette Annexe on présente l'AMDEC lié au système de drone que l'on développe. On détaillera cette analyse pour les différentes parties du drone afin de mettre le point sur les différents problèmes que l'utilisateur pourrait rencontrer et y proposer des solutions.

### B.1 Criticité

Il s'agit dans un premier temps de caractériser le niveau de criticité du système par rapport à un mode de défaillance. On proposera ensuite un certain nombre de solutions à ces points critiques afin de rendre le système le plus sûr possible. Parmi les critères de sécurité on distingue la disponibilité de la sécurité. Le premier cas désigne le fait que le système soit plus ou moins facile à réparer (1 : irréparable; 4 : Maintenance de base). Pour la sécurité on met en évidence si le système présente des danger liés à la sécurité (1 : très dangereux à 4 : sans gravité). On liera ces deux aspects avec la probabilité d'apparition du problème (1 : très fréquent à 4 : rare).

On note que pour notre système on n'a pas le même niveau d'exigence entre la disponibilité et la sécurité. On considère qu'il est plus important que le système préserve les tiers (biens ou personnes) que lui-même. Ainsi on présente les Tableaux B.1,B.2de criticité. Le code couleur est donné dans le tableau ?? suivant :

|                               |         |                                    |        |
|-------------------------------|---------|------------------------------------|--------|
| Très critique (TC) - Sécurité | 1 à 4   | Très critique (TC) - disponibilité | 1 à 3  |
| Critique (C) - Sécurité       | 5 à 10  | Critique (C) - disponibilité       | 4 à 7  |
| Peu critique (PC)             | 11 à 16 | Peu critique (PC)                  | 8 à 16 |

| Probabilité \ Gravité Sécurité | Gravité Sécurité |   |    |    |
|--------------------------------|------------------|---|----|----|
|                                | 1                | 2 | 3  | 4  |
| 1                              | 1                | 2 | 3  | 4  |
| 2                              | 2                | 4 | 6  | 8  |
| 3                              | 3                | 6 | 9  | 12 |
| 4                              | 4                | 8 | 12 | 16 |

TABLEAU B.1 – Matrices de criticité Sécurité

| Probabilité \ Gravité<br>Disponibilité | 1 | 2 | 3  | 4  |
|--|---|---|----|----|
|  | 1 | 1 | 2  | 3  |
| 2                                      | 2 | 4 | 6  | 8  |
| 3                                      | 3 | 6 | 9  | 12 |
| 4                                      | 4 | 8 | 12 | 16 |

TABLEAU B.2 – Matrices de criticité disponibilité

## B.2 Structure

On commencera par détailler les défaillances possibles pour la structure du drone. On continuera ensuite avec les défaillances possibles dues au système embarqué. Celui-ci contiendra la batterie, le récepteur, les servo-moteurs, et le groupe moto-propulseur. On détaille dans le Tableau B.3 suivant les fonctions principales (F.P.) et contraintes (F.C.) du système.

| Fonction | Critère  |
|----------|--|
| FP 1.1   | Assurer la stabilité en vol des équipements embarqués          |
| FP 1.2   | Préserver l'intégrité de l'opérateur et des intervenants       |
| FP 1.3   | Préserver l'intégrité des outils de mesure                     |
| FP 1.4   | Garantir une bonne approche de l'outil par rapport à l'ouvrage |
| FC 1.1   | Supporter la chaleur et vibrations dégagées par le moteur      |
| FC 1.2   | Assurer la sécurité du système par rapport à l'environnement   |
| FC 1.3   | Supporter les servo-moteurs                                    |
| FC 1.4   | Protéger la batterie des chocs                                 |
| FC 1.5   | Supporter l'alimentation du système                            |
| FC 1.6   | Ne pas perturber la communication RC                           |
| FC 1.7   | Supporter le système embarqué                                  |
| FC 1.8   | Amortir les vibrations du système embarqué                     |

TABLEAU B.3 – Fonction principale et contraintes du projet

La Figure B.1 présente les différentes liaisons critiques et principales de la partie structure du système.

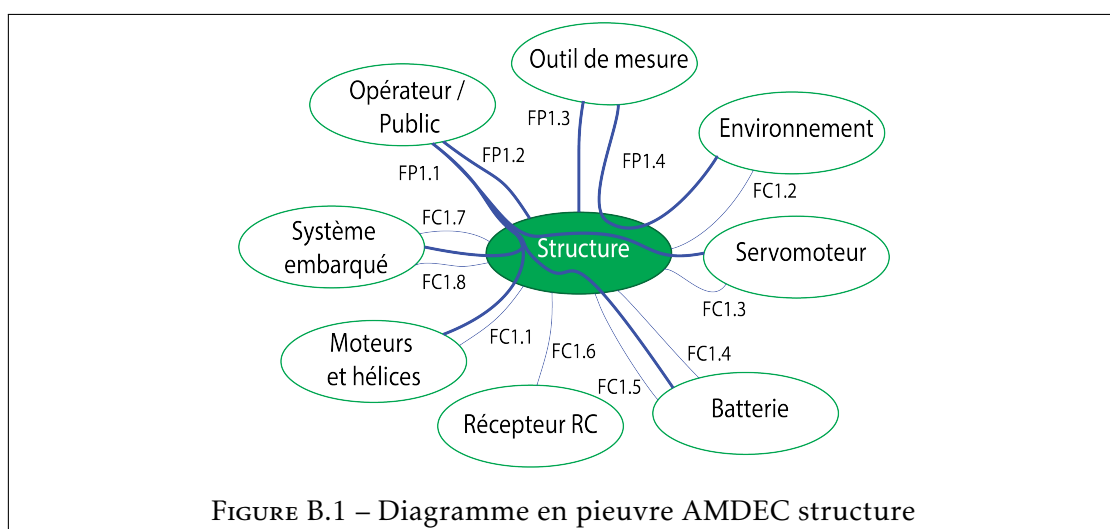


FIGURE B.1 – Diagramme en pieuvre AMDEC structure

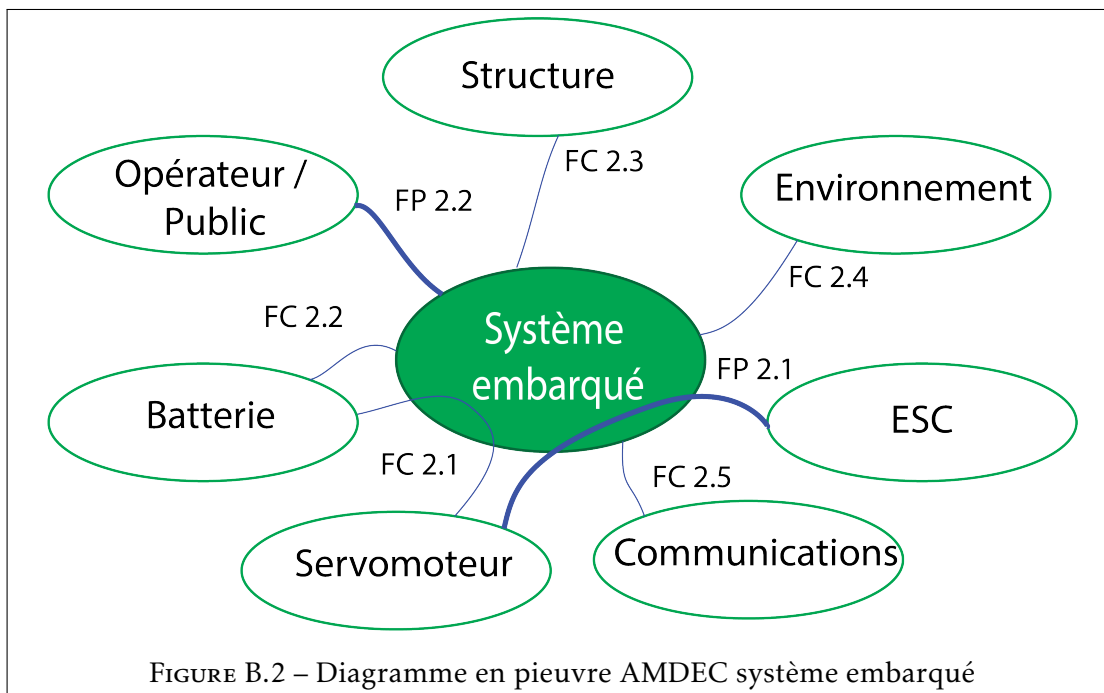
Les deux Figures B.3,B.4 détaillent les modes de criticité ainsi que les moyens de les détecter, d'y répondre et de les prévenir.

### B.3 Système embarqué

Pour le système embarqué on présente le Tableau B.4 et le Schéma B.2.

| Fonction | Critère                                    |
|----------|--|
| FP 2.1   | Commander les actionneurs                  |
| FP 2.2   | Assurer l'interface homme-machine          |
| FC 2.1   | Garantir l'alimentation des servomoteurs   |
| FC 2.2   | Garantir l'alimentation du microcontrôleur |
| FC 2.3   | Garantir la fixation                       |
| FC 2.4   | Gérer les perturbations dues au vent       |
| FC 2.5   | Résistance aux interférences radios        |

TABLEAU B.4 – Fonction principale et contraintes de l'AMDEC système embarqué



| Fonction | Modes de défaillance                                       | Origine | Conséquence                             | Cause                                       | GS | GD | P | Criticité |               | Détection   | Correction                                    | Prévention   | Remarque             |
|----------|--|---------|---|---|----|----|---|-----------|---------------|---|---|--|----------------------|
|          |  |         |   |   |    |    |   | Sécurité  | Disponibilité |   |   |  |                      |
| FP 1.1   | Instabilité du vol   | H/T     | Perte de contrôle et collision du drone | Turbulence aéronautique                     | 3  | 2  | 2 | C         | C             | Centrale inertielle, GPS                              | Optimiser l'aérodynamisme                     | Tests en soufflerie  |                      |
|          |  |         |   | Mauvaise répartition de l'équipement        | 3  | 2  | 4 | C         | PC            | Mesure de balourd                                     | Déplacement de l'équipement                   | Test manuel d'équilibrage                                      |                      |
| FP 1.2   | L'aéronef entre en contact avec un tiers                   | H/A/T   | Dommages Corporels / Dommages matériels | Structure mal dimensionnée                  | 1  | 1  | 4 | TC        | C             | Centrale inertielle                                   | déclenchement de l'arrêt d'urgence, parachute | Mise en place de dispositifs de sécurité (Carénage, parachute) | Maillage <2          |
|          |  |         |   | Mauvaise protection du capteur              | 3  | 3  | 3 | C         | PC            | Tests en soufflerie pour détecter les infrastructures | Rajout de protections                         | Révision du drone  |                      |
| FP 1.3   | Perte outil / interaction destructive avec l'environnement | A/H/T   | Endommagement / Destruction du capteur  | Mauvaise fixation                           | 2  | 1  | 4 | C         | C             | Vérification de la fixation de l'outil                | Maintenance mécanique                         | Révision du drone  | Utilisation dragonne |
|          |  |         |   | Non-amortissement de l'approche à l'ouvrage | 2  | 2  | 2 | TC        | C             | Maintenance mécanique                                 |   | Révision du drone  | Ajustement ressort   |

FIGURE B.3 – AMDEC FP - Structure

| Fonction | Modes de défaillance                        | Origine | Conséquence  | Cause  | GS | GD | P | Criticité |               | Détection                          | Correction                                    | Prévention   | Remarque    |
|----------|---|---------|--|--|----|----|---|-----------|---------------|------------------------------------|---|--|-------------|
|          |   |         |  |  |    |    |   | Sécurité  | Disponibilité |                                    |   |  |             |
| FC 1.1   | Surchauffe / déserrage des composants       | T       | Dégradation / Rupture / perte des composants                           | Mauvaise fixation / raisonnante / Mauvaise ventilation | 2  | 2  | 2 | TC        | C             | Test manuel de serrage des moteurs | Reserrage vis                                 | Application de frein filet sur les vis                         |             |
| FC 1.2   | Collision avec élément extérieur            | A/T/H   | Déviations de la trajectoire / Dommages Corporels / Dommages matériels | Pas de protection des parties actives                  | 3  | 3  | 2 | C         | C             | Centrale inertielle                | Déclenchement de l'arrêt d'urgence, parachute | Mise en place de dispositifs de sécurité (Carénage, parachute) | Maillage <2 |
| FC 1.3   | Sortie des servomoteurs de leur emplacement | T       | Déviations de la trajectoire / Dommages Corporels / Dommages matériels | Mauvaise fixation / Mauvaise conception                | 2  | 2  | 4 | C         | C             | Centrale inertielle                | Atterissage d'urgence                         | Mise en place de d'un parachute                                |             |
| FC 1.4   | Combustion de la batterie                   | T/A     | Destruction du drone et danger corporel                                | Choc violent de la batterie                            | 1  | 2  | 4 | TC        | C             | Système de sécurité                | Coupe moteurs + ouverture parachute           | Test du dispositif de sécurité / maintenance mécanique         |             |
| FC 1.5   | Sortie de la batterie de son emplacement    | A/T/H   | Balourd / coupure d'alimentation / collision du drone / incendie       | Mauvaise fixation / Mauvaise conception                | 2  | 3  | 4 | C         | C             | Centrale inertielle                | Atterissage d'urgence / coupe moteurs         | Maintenance mécanique  |             |
| FC 1.6   | Interférences avec le système de RC         | T/A     | Mauvaises commandes du drone   | Non-réception / altération des commandes de vol        | 2  | 2  | 3 | C         | C             | Télépilote                         | Atterissage d'urgence                         | Changement des fréquences radio                                |             |
| FC 1.7   | Perte du système embarqué                   | T/A     | Balourd / coupure de la commande/ Crash du drone                       | Mauvaise fixation / Mauvaise conception                | 2  | 2  | 4 | C         | C             | Système de sécurité                | Atterissage d'urgence / coupe moteurs         | Maintenance mécanique  |             |
| FC 1.8   | Perturbation des mesures angulaires         | T       | Vol instable du drone  | Vibrations dans le système embarqué                    | 3  | 3  | 4 | C         | PC            | Centrale inertielle                | Atterissage d'urgence                         | Maintenance mécanique  |             |

FIGURE B.4 – AMDEC FC - Structure

| Fonction | Modes de défaillance                      | Origine | Conséquence  | Cause                                   | GS | GD | P | Criticité |               | Détection                 | Correction  | Prévention   | Remarque              |
|----------|---|---------|--|---|----|----|---|-----------|---------------|---------------------------|---|--|-----------------------|
|          |   |         |  |   |    |    |   | Sécurité  | Disponibilité |                           |   |  |                       |
| FP 2.1   | Mauvaise commande des actionneurs         | H/A/T   | Vol non maîtrisé et aléatoire                              | Rupture du lien de communication        | 2  | 2  | 3 | C         | C             | Visuel                    | Maintenance électrique                                    | Tests des actionneurs avant le vol                   |                       |
|          |   |         |  | Envoi de la mauvaise commande           | 2  | 2  | 4 | C         | C             | Centrale inertielle       | Utilisation du système d'urgence                          | Tests des actionneurs avant le vol                   |                       |
| FP 2.2   | Problème d'interaction homme/Machine      | T       | Perte de contrôle du drone                                 | Surcharge du processeur                 | 3  | 3  | 2 | C         | C             | Visuel                    | Optimisation des programmes et recompilation              | Optimisation des programmes                          |                       |
| FC 2.1   | absence d'alimentation des servos         | T/A/H   | Perte de contrôle du drone                                 | Coupure du système électrique           | 2  | 2  | 4 | C         | C             | Visuel                    | Maintenance électrique                                    | Tests des actionneurs avant le vol                   |                       |
| FC 2.2   | absence d'alimentation du microcontrôleur | T/A     | Perte de contrôle du drone                                 | Coupure du système électrique           | 2  | 2  | 4 | C         | C             | Système d'urgence         | Coupure des moteurs et parachute                          | Maintenance préventive                               |                       |
| FC 2.3   | Détachement d'un dispositif               | T/H/A   | Perte totale ou partielle de contrôle du drone             | Mauvaise fixation d'un élément          | 2  | 3  | 4 | C         | PC            | Visuel, système d'urgence | Utilisation du système d'urgence   atterrissage d'urgence | Maintenance Mécanique                                | Voir schéma structure |
| FC 2.4   | Vol instable du drone                     | T/A     | Oscillations du drone pouvant mener à sa perte de contrôle | Raffale de vent trop importante         | 2  | 2  | 2 | TC        | C             | Centrale inertielle       | Atterrissage d'urgence                                    | Loi de commande robuste                              |                       |
| FC 2.5   | Instabilité du drone / fly away           | T/A     | Déviations de la trajectoire / perte du drone              | Perturbation magnétique / interférences | 2  | 2  | 2 | TC        | C             | Programme                 | Retour à la base / atterrissage d'urgence                 | Création d'un programme de retour maison si problème |                       |

FIGURE B.5 – AMDEC FC - Système embarqué

# Précis mathématiques

## C.1 Fonctions usuelles

### C.1.1 Fonction de saturation

On définit la fonction de saturation noté  $\sigma_i$  avec  $i > 0$  comme suit

$$\sigma_i(x) = \begin{cases} -i & \text{si } x < -i \\ x & \text{si } -i < x < i \\ i & \text{si } x > i \end{cases} \quad (\text{C.1})$$

### C.1.2 Fonction signe

On définit la fonction signe noté  $sign(x)$  comme suit

$$sign(x) = \begin{cases} -1 & \text{si } x < 0 \\ 0 & \text{si } x = 0 \\ 1 & \text{si } x > 0 \end{cases} \quad (\text{C.2})$$

### C.1.3 Pseudoinverse de Moore-Penrose

Soit la fonction de pseudo-inverse pour une matrice  $A$  notée  $A^\dagger$  qui satisfait les conditions suivantes :

$$\begin{aligned} AA^\dagger A &= A \\ A^\dagger AA^\dagger &= A^\dagger \\ (AA^\dagger)^* &= AA^\dagger \\ (A^\dagger A)^* &= A^\dagger A \end{aligned} \quad (\text{C.3})$$

Avec  $A^*$  le conjugué de la matrice  $A$ . On donne également l'expression pour calculer son expression comme suit, si  $(AA^*)^{-1}$  est inversible :

$$A^\dagger = A^* (AA^*)^{-1} \quad (\text{C.4})$$

Note : on a alors  $A^\dagger A = I$ ,  $I$  étant la matrice identité.

Si la condition précédente n'est pas remplie, on choisira plutôt d'inverser  $(A^* A)$ . L'expression de la pseudo inverse sera alors calculé de la manière suivante :

$$A^\dagger = (A^* A)^{-1} A^* \quad (\text{C.5})$$



Note : on aura alors  $AA^\dagger = I$

## C.2 Preuves de stabilité

Afin de prouver la stabilité des loi de commandes utilisés, on choisit de faire des preuves de stabilité en utilisant les théorèmes de Lyapunov. Soit une fonction de Lyapunov candidate  $V$ .

les conditions de stabilité sont les suivantes

- Stabilité :  $V(0) = 0, V(x) > 0, \dot{V}(x) \leq 0$
- Stabilité asymptotique :  $V(0) = 0, V(x) > 0, \dot{V}(x) < 0$  pour  $x \neq 0$
- Globalement Stable :  $V(0) = 0, V(x) > 0$ , radialement non borné  $\|x\| \rightarrow 0 \implies f(x) \rightarrow \infty$
- Globalement Asymptotiquement Stable (G.A.S.) : globalement stable et stable asymptotiquement

Dans les chapitres de commande on applique l'algorithme de saturations imbriquées sur un système qui correspond à un double intégrateur de la forme

$$\begin{aligned}\dot{x}_1 &= x_2 \\ \dot{x}_2 &= u\end{aligned}\tag{C.6}$$

Note : les termes d'inertie sont négligés pour plus de simplicité. On note  $x_1$  l'état correspondant à la position,  $x_2$  la vitesse,  $\dot{x}_2$  l'accélération et  $u$  l'entrée de commande du système.

### C.2.1 Système linéaire

Soit la loi de commande de type P.D. suivante :

$$\ddot{x} = u = -\dot{x} - x\tag{C.7}$$

On propose la fonction de Lyapunov candidate suivante :

$$V = \frac{x^2}{2} + \frac{\dot{x}^2}{2}\tag{C.8}$$

d'où sa dérivée

$$\begin{aligned}\dot{V} &= x\dot{x} + \dot{x}\ddot{x} \\ \dot{V} &= \dot{x}(x + u) \\ \dot{V} &= -\dot{x}^2\end{aligned}\tag{C.9}$$

le système est G.A.S.

### C.2.2 Saturations imbriquées

Dans la section précédente on définit la fonction de saturation  $\sigma_i$  avec  $i > 0$  une constante positive qui représente les bornes de la saturation.

On propose la commande suivante pour stabiliser le système :

$$\dot{x}_2 = u = -\sigma_a(x_2 + \sigma_b(\cdot))\tag{C.10}$$

avec  $a$  et  $b$  des constantes qui seront définis par la suite.  
On propose la fonction de Lyapounov candidate suivante.

$$V_1 = \frac{1}{2}x_2^2 \quad (\text{C.11})$$

et sa dérivée

$$\dot{V}_1 = u x_2 = -x_2 \sigma_a(x_2 + \sigma_b(\cdot)) \quad (\text{C.12})$$

On remarque que si  $|x_2| \geq b$  alors

$$x_2 + \sigma_b(\cdot) = \text{sign}(x_2)n \quad , n > 0 \quad (\text{C.13})$$

et donc

$$\dot{V}_1 < 0 \quad (\text{C.14})$$

Ce qui par définition implique de l'état  $x_2$  converge vers 0. Cela a pour conséquence  $\exists$  un temps  $t \geq t_1$  où  $|x_2| \leq b$  ce qui implique de choisir la saturation  $a$  de sorte que

$$\begin{matrix} |x_2 + \sigma_b(\cdot)| \leq 2b = a \\ (\leq b) \quad (\leq b) \end{matrix} \quad (\text{C.15})$$

$a$  pourra être choisi en fonction de la saturation des actionneurs, ou au choix de l'utilisateur. à partir du temps  $t_1$  on peut donc écrire

$$u = -x_2 - \sigma(\cdot) \quad (\text{C.16})$$

On définit les fonctions suivantes

$$\begin{aligned} \dot{v} &= u + x_2 \\ v &= x_2 + x_1 \\ \sigma_b(v) &= \sigma_b(x_2 + x_1) \quad , b > 0 \end{aligned} \quad (\text{C.17})$$

ce qui implique pour  $t \geq t_1$

$$\begin{aligned} \dot{v} &= -x_2 - \sigma_b(x_2 + x_1) + x_2 \\ \dot{v} &= -\sigma_b(x_2 + x_1) \end{aligned} \quad (\text{C.18})$$

En définissant la fonction de Lyapounov candidate suivante et sa dérivée

$$\begin{aligned} V_2 &= \frac{1}{2}v^2 \\ \dot{V}_2 &= v \dot{v} = -v \sigma_b(v) \end{aligned} \quad (\text{C.19})$$

si  $|v| > b$  alors on a

$$\dot{V}_2 = -v b \text{sign}(v) \leq 0 \quad (\text{C.20})$$

Par conséquent  $v$  converge vers 0.  $\exists t > t_2 > t_1$  où  $|v| \leq b$ . Dans ce cas on peut alors écrire

$$\dot{V}_2 = -v^2 \leq 0 \quad (\text{C.21})$$

Donc  $v$  converge exponentiellement vers 0. Or  $x_2$  converge également vers 0. D'après la définition de  $v$  en équation (C.17),  $x_1$  converge aussi vers 0. Puisque les états du système convergent vers 0, on considère que la stabilité du système est garantie.

### C.2.3 Sliding mode super-twisting

Soit la surface  $S = x_1$ , on propose le contrôleur mode glissant super twisting suivant :

$$u = \dot{x}_1 = -\gamma_1 |x_1|^l \text{sign}(x_1) + x_2 \quad (\text{C.22})$$

$$\dot{x}_2 = -\gamma_3 \text{sign}(x_1) \quad (\text{C.23})$$

Note : le système est considéré sans perturbations. On propose la fonction de Lyapunov candidate suivante :

$$V(x) = 2\gamma_3 |x_1| + \frac{1}{2} x_2^2 + \frac{1}{2} (\gamma_1 |x_1|^l \text{sign}(x_1) - x_2)^2 \quad (\text{C.24})$$

On peut réécrire cette équation sous la forme quadratique suivante :

$$V(x) = \mathcal{N}^T \mathcal{M} \mathcal{N} \quad (\text{C.25})$$

avec

$$\mathcal{M} = \frac{1}{2} \begin{bmatrix} 4\gamma_3 + \gamma_1^2 & -\gamma_1 \\ -\gamma_1 & 2 \end{bmatrix} \quad (\text{C.26})$$

$$\mathcal{N}^T = \begin{bmatrix} |x_1|^{1/2} \text{sign}(x_1) & x_2 \end{bmatrix} \quad (\text{C.27})$$

La fonction  $V(x)$  est continue et dérivable en tout point sauf  $x_1 = 0$ . Note qu'en ce point le système atteint son objectif. De plus cette fonction est bornée, on peut donc écrire :

$$\lambda_{\min}(\mathcal{M}) \|\mathcal{N}\|^2 \leq V(x) \leq \lambda_{\max}(\mathcal{M}) \|\mathcal{N}\|^2 \quad (\text{C.28})$$

Avec  $\lambda$  une fonction qui donne les valeurs propres d'une matrice, et  $\|\mathcal{N}\|^2 = |x_1| + x_2^2$ . On note que  $\|\mathcal{N}\|^2 \geq |x_1|$ . D'où on peut écrire :

$$|x_1| \leq \|\mathcal{N}\|^2 \leq \frac{V(x)}{\lambda_{\min}(\mathcal{M})} \quad (\text{C.29})$$

$$|x_1|^{1/2} \leq \|\mathcal{N}\| \leq \frac{V(x)^{1/2}}{\lambda_{\min}(\mathcal{M})^{1/2}} \quad (\text{C.30})$$

On dérive  $V(x)$  pour obtenir :

$$\dot{V}(x) = -\frac{1}{|x_1|^{1/2}} \mathcal{N}^T \mathcal{L} \mathcal{N} \quad (\text{C.31})$$

Avec

$$\mathcal{L} = \frac{\gamma_1}{2} \begin{bmatrix} 2\gamma_3 + \gamma_1^2 & -\gamma_1 \\ -\gamma_1 & 1 \end{bmatrix} \quad (\text{C.32})$$

De même qu'en équation (C.28) on peut noter

$$-\frac{1}{|x_1|^{1/2}} \mathcal{N}^T \mathcal{L} \mathcal{N} \leq -\frac{1}{|x_1|^{1/2}} \|\mathcal{N}\|^2 \lambda_{\max}(\mathcal{L}) \quad (\text{C.33})$$

En remplaçant par (C.30) et (C.29) on obtient :

$$\dot{V}(x) \leq -\frac{\lambda_{\min}(\mathcal{M})^{1/2}}{V(x)^{1/2}} \|\mathcal{N}\|^2 \lambda_{\max}(\mathcal{L}) \quad (\text{C.34})$$

$$\leq -\frac{V(x)}{\lambda_{\min}(\mathcal{M})} \frac{\lambda_{\min}(\mathcal{M})^{1/2}}{V(x)^{1/2}} \lambda_{\max}(\mathcal{L}) \quad (\text{C.35})$$

$$\leq -\frac{V(x)^{1/2} \lambda_{\max}(\mathcal{L})}{\lambda_{\min}(\mathcal{M})^{1/2}} \quad (\text{C.36})$$

On définit :

$$\mathcal{U} = -\frac{\lambda_{\max}(\mathcal{L})}{\lambda_{\min}(\mathcal{M})^{1/2}} \quad (\text{C.37})$$

On définit une fonction  $v(x)$  qui pour une équation différentielle de la forme suivante :

$$\dot{v}(x) = -\frac{v(x)^{1/2} \lambda_{\max}(\mathcal{L})}{\lambda_{\min}(\mathcal{M})^{1/2}} \quad (\text{C.38})$$

possède une solution comme suit :

$$\dot{v}(x) = \left( v(0)^{1/2} - \frac{\mathcal{U}}{2} t \right)^2 \quad (\text{C.39})$$

D'après le principe de comparaison donné en [107] on peut écrire que  $v(x) \leq V(x)$ . Ce qui implique, d'après l'équation (C.28) que  $V(x)$  converge vers 0 et donc que les états convergent également vers 0 après un temps  $T = \frac{2V^{1/2}(0)}{\mathcal{U}}$ .

## Plateforme

### D.1 Mesure de l'angle de rotation des servomoteurs

On décrit ici les travaux réalisés pour intégrer les capteurs de position angulaire dans les servo-moteurs. La plupart de ceux-ci n'ont généralement qu'une rotation sur  $180^\circ$ . Il est donc nécessaire de changer leur fonctionnement pour avoir une rotation continue. Pour se faire on choisit d'utiliser un capteur de position angulaire en deux parties (aimant et magnétomètre).

On propose de concevoir deux pièces permettant de contenir l'aimant et le capteur de position angulaire pour le servomoteur, voir Figure D.1 (gauche). Le porte-aimant est ensuite directement placé sur l'axe de sortie de roulement du servo-moteur. Le porte-capteur quand à lui vient remplacer l'emplacement de la partie commande du servomoteur. Une fois l'intégration achevée, on obtient un positionnement coaxial de l'aimant et du capteur de position angulaire, voir Figure D.1(droite).

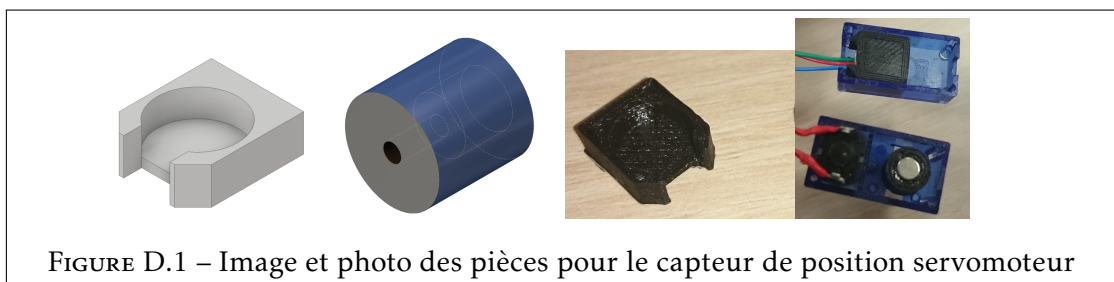


FIGURE D.1 – Image et photo des pièces pour le capteur de position servomoteur

Comme expliqué précédemment, on utilisera des capteurs de position angulaire à rotation continue référence RLS RM08 comme sur la Figure D.2 à gauche.

On les intègre directement dans les servomoteurs à l'aide des pièces réalisées par CAO cités précédemment. Le résultat est donné à droite sur la même figure.

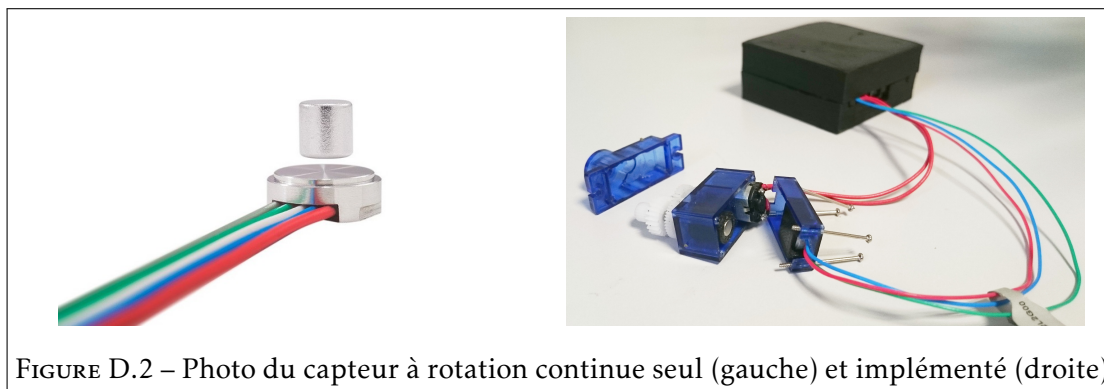


FIGURE D.2 – Photo du capteur à rotation continue seul (gauche) et implémenté (droite)

La phase suivante consiste à contrôler cet actionneur dans sa nouvelle configuration. Pour ce faire, on propose d'utiliser un micro-contrôleur Arduino. On développe un algorithme permettant de contrôler le servomoteur à partir des données collectées par le capteur. Le dispositif fonctionne avec un temps de retard comme on peut le voir sur la Figure D.3.

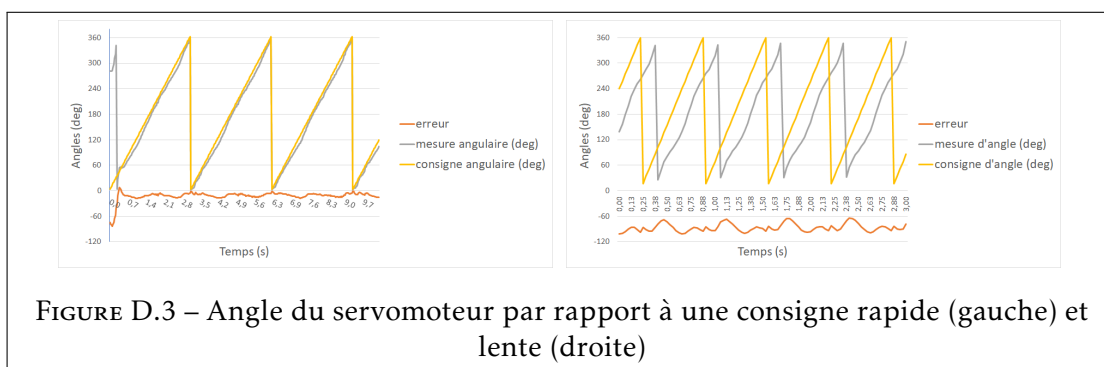


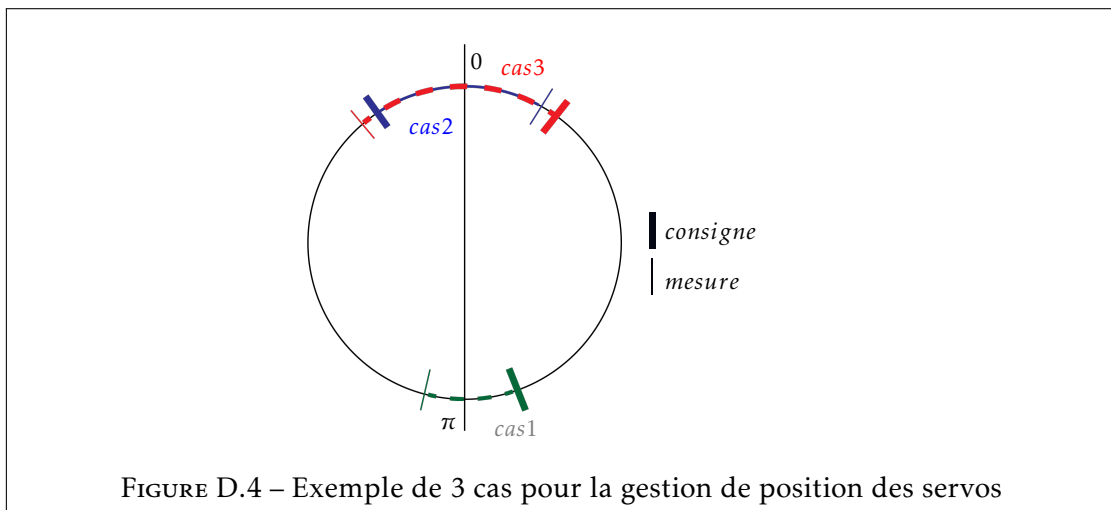
FIGURE D.3 – Angle du servomoteur par rapport à une consigne rapide (gauche) et lente (droite)

L'algorithme utilisé n'est pas trivial. En effet, en fonction de la position de la consigne et de la mesure, l'erreur ne doit pas forcément être la même. On a ici une fonction cyclique ce qui a pour conséquence que l'erreur, donc le chemin entre la consigne et la mesure est plus grande si on tourne dans un sens ou dans l'autre. Au niveau du capteur de position, la mesure va de 0 à 1024 avec 1024 qui correspond 360°. On va donc chercher la distance la plus courte entre ces valeurs. Pour décrire l'algorithme on décrit 3 cas possibles entre la position de la consigne et celle du servo-moteur. En fonction du cas, on choisit de définir l'erreur de manière à avoir sa valeur absolue minimale.

- Cas1 : consigne-mesure
- Cas2 : (consigne+1023)-mesure
- Cas3 : consigne-(mesure+1023)

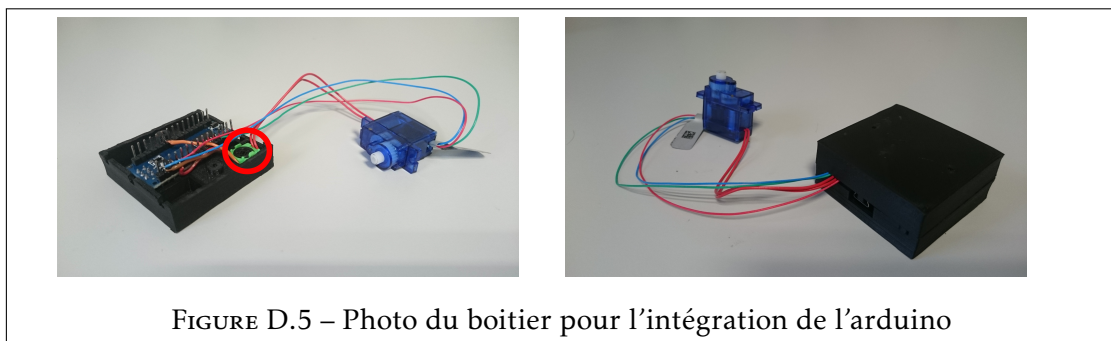
Le fait de rajouter 1023 permet de comparer la première valeur avec la seconde qui aurait fait un tour de plus.

Sur la Figure D.4 on donne un exemple où chacun des cas décrit précédemment est un optimal.



### D.1.1 Boîtier Arduino

On propose de dessiner un boîtier permettant de contenir l'Arduino qui contrôlera les servomoteurs. De plus, comme décrit dans la section précédente, on retire la partie commande des servomoteurs pour obtenir un gain de place. Ce faisant on propose de replacer cette partie (encerclé en rouge sur la figure) dans le boîtier. Le dessin ne possède aucune particularité, il s'agit simplement de pouvoir accueillir tous les composants. On prévoit également la place pour faire passer les différents câbles. Enfin, on rajoute des cylindres afin de pouvoir visser les deux moitiés avec le châssis. Le résultat est donné en Figure D.5



### D.1.2 Accouplement entre bras et servomoteurs

Suite à des tests, on constate que l'accouplement détaillé en première année n'est pas parfait. En effet, la rotation du servomoteur par rapport à l'embout n'est pas bloquée. Les servos étant vendus avec une pièce s'accouplant parfaitement sur la tête, on adapte l'embout à cette tête de servo pour bloquer la rotation. On vis ensuite la tête de servo sur l'embout imprimé, voir Figure D.6.

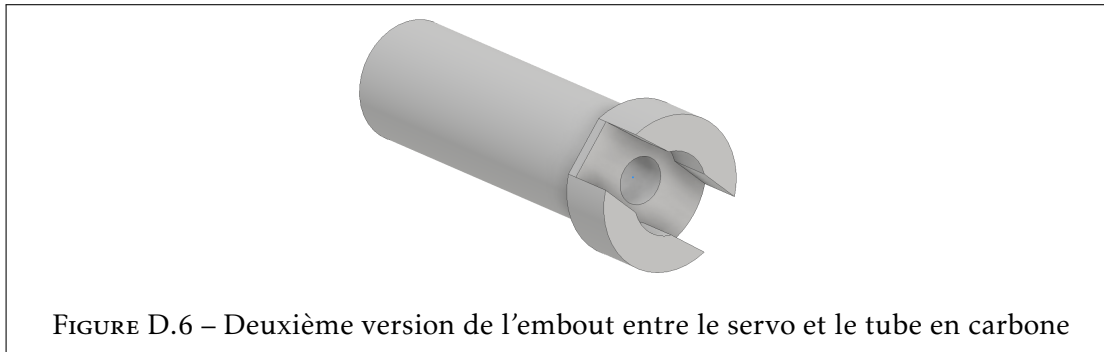


FIGURE D.6 – Deuxième version de l'embout entre le servo et le tube en carbone

## D.2 Estimation de l'altitude

Cette section est une ouverture pour un utilisateur ultérieur. Une manière supplémentaire de mesurer l'altitude (autre que l'OptiTrack) du drone est d'utiliser un baromètre. Cela peut être très intéressant quand l'OptiTrack ou encore le GPS ne sont pas disponibles. Afin d'estimer l'altitude du dispositif, on peut mesurer la hauteur avec le baromètre et filtrer le résultat avec l'accéléromètre. Cette méthode n'a pas été implémentée faute de temps mais on laisse au lecteur quelques pistes. Une méthode de filtrage par Kalman comme décrite précédemment est appliquée à ce capteur et permet d'améliorer la mesure que l'on aurait avec l'accéléromètre seul. Le système prend la forme suivante :

$$\dot{\mathbf{X}} = \begin{bmatrix} 0 & 1 \\ 0 & 0 \end{bmatrix} \mathbf{X} + \begin{bmatrix} 0 \\ 1 \end{bmatrix} \mathbf{U} + \alpha(t) \quad (\text{D.1})$$

$$\mathbf{Y} = \begin{bmatrix} 1 & 0 \end{bmatrix} \mathbf{X} + \beta(t) \quad (\text{D.2})$$

On peut utiliser les valeurs suivantes pour le filtre :

$$\mathbf{X} = \begin{bmatrix} z \\ \dot{z} \end{bmatrix} \quad \mathcal{F} = \begin{bmatrix} 1 & T_{ech} \\ 0 & 1 \end{bmatrix} \quad \mathcal{G} = \begin{bmatrix} 0 \\ T_{ech} \end{bmatrix} \quad \mathcal{C} = \begin{bmatrix} 1 & 0 \end{bmatrix} \quad \mathcal{D} = 0 \quad (\text{D.3})$$

La matrice  $\mathbb{R}$  sera dans le cas présent un scalaire dont la valeur est l'écart type de la mesure du baromètre, noté  $\sigma_b$ . De plus, faisant davantage confiance au modèle du baromètre qu'à celui de l'accéléromètre, on choisira les gains suivants (avec  $a \ll b$ , deux constantes) :

$$\mathbb{Q} = \begin{bmatrix} a & 0 \\ 0 & b \end{bmatrix} \quad \mathbb{R} = \sigma_b \quad (\text{D.4})$$

Le gain  $a$  correspond à la confiance du baromètre et  $b$  à celle de l'accéléromètre. Note : le fait d'utiliser l'accéléromètre en tant qu'entrée de commande permet d'effectuer une fusion des données en plus du filtrage.



## D.3 Précisions d'utilisation

### D.3.1 Installation du framework développé

L'immense intérêt de cette architecture est le fait de ne pas être obligé de coder le système. Un autre avantage est le fait de pouvoir utiliser directement le modèle utilisé avec le simulateur décrit précédemment. Le processus d'essai-erreur peut donc être accéléré pour obtenir un système robuste plus rapidement. L'inconvénient est que l'on n'a pas de prise directe sur ce qui a été codé. Cependant aucun problème dû à l'interprétation de Matlab n'a pu être identifié à ce jour.

Pour se servir du système décrit ci-dessus on commence par suivre le tutoriel d'installation donné par Emlid. Il est impératif de configurer l'adresse IP dans le fichier :

```
/etc/environment
```

En ajoutant la ligne

```
ROS_IP=192.168.1.1
```

L'IP ci-dessus doit correspondre à celle du Raspberry Pi utilisé.

Astuce : on pourra afficher l'ensemble des services et leurs états avec la commande :

```
systemctl list-unit-files --type=service
```

Une fois le système en place on peut se servir librement des nœuds créés dans Matlab. Une petite configuration doit cependant être effectuée au préalable pour que le logiciel puisse les reconnaître. Pour commencer il faut installer une toolbox nommée "*ROS Toolbox interface for ROS Custom Messages*". Ensuite, il faut importer l'ensemble des messages qui seront utilisés. Pour cela il faut récupérer les messages utilisés aux emplacements suivants sur le Raspberry.

On trouvera le message PWM.msg directement dans le dossier du nœud créé pour écrire sur les sorties du Navio :

```
/home/pi/catkin_ws/src/Capteurs/msg/PWM.msg
```

On pourra trouver les autres messages dans le dossier :

```
/opt/ros/kinetic/share/mavros_msgs/.
```

On les importera ensuite dans Matlab en utilisant la commande

```
rosgenmsg(folderpath)
```

On suivra ensuite les indications à l'écran.

Note : attention, avant l'importation, il faut changer au préalable les fichiers package.xml. Tous les "depend" doivent être convertis en "build-depend" sinon l'opération ne fonctionnera pas.

Note2 : si, pour quelque raison que ce soit, on doit effectuer l'importation plusieurs fois il est alors nécessaire de supprimer les fichiers dans le dossier matlab\_gen au préalable.

Ceci fait, on pourra commencer la "programmation" via Simulink avec les nœuds. Pour tester le modèle directement on peut appuyer sur le bouton play après configuration de l'hôte ROS.

Le système tournera en local sur l'ordinateur et non sur le Raspberry Pi ce qui permettra de faire des tests rapides mais ne donnera pas la possibilité d'effectuer des vols. En effet dans ce mode, il est impossible d'avoir un taux de rafraichissement suffisant pour garantir la stabilité de vol.

Une fois le code testé en local on peut alors le cibler sur la machine avec le bouton "*deploy to hardware*". Lors de la première compilation, il faudra renseigner les informations nécessaires à la compilation comme le nom d'utilisateur et le mot de passe par exemple. Le modèle est alors converti en code C++ ROS qui sera compilé en local sur la machine. Le processus est assez lent du fait de la puissance de calcul du Raspberry Pi. On note que l'on ne peut pas cross-compiler à l'heure actuelle. Cela devrait être envisageable à l'avenir (notamment avec ROS2).

On note également que le fait d'utiliser ROS permet de jouir du soutien de la communauté ce qui n'est pas négligeable.

### D.3.2 OptiTrack

On installera le package ROS du VRPN correspondant, à l'aide de la commande suivante :

```
sudo apt-get install ros-kinetic-vrpn-client-ros
```

Le manuel peut être trouvé sur le wiki ROS <sup>1</sup>.

Le nœud peut ensuite être démarré en utilisant la commande suivante :

```
roslaunch vrpn_client_ros sample.launch server:=172.26.21.1:3883
```

L'adresse IP est à adapter en fonction du serveur de l'Optitrack. Cette commande devra être lancée après le démarrage du système d'Optitrack et ne fait donc pas partie des commandes lancées au démarrage du système.

### D.3.3 Dynamixels

Celui-ci sera configuré de la manière suivante :

- Dans le fichier Dynamixel *controller.launch*, les périodes sont mises à 0.005s soit 200 Hz.
- Dans le dossier *config* on change le fichier *basic.yaml* où l'on crée le nom des joints : "un", "deux", "trois", "quatre"

On détail le nœud *chatter*. Il est démarré manuellement avec la commande :

```
roslaunch dynamixel_workbench_controllers chatter
```

Il permet d'envoyer les consignes de position via un message de type flottant sous forme de radian. Le nœud qui permet le dialogue avec les Dynamixels quant à lui sera également démarré manuellement à l'aide de la commande :

```
roslaunch dynamixel_workbench_controllers dynamixel_controllers.launch
```

Note : Pendant les tests on remarque que le système final fonctionne mais avec une restriction, dont la cause n'a pas encore été trouvée. Lorsque l'on démarre le nœud des Dynamixels il a tendance à s'effondrer ou à ne pas détecter les Dynamixels.

1. [http://wiki.ros.org/vrpn\\_client\\_ros](http://wiki.ros.org/vrpn_client_ros)

Pour remédier à ça, il suffit de débrancher et de rebrancher l'USB2AX 2 fois sur le Raspberry Pi. La connexion se fait alors sans problème. On propose également une solution en ligne de commande en installant le paquet "hub-ctrl"<sup>2</sup>. On peut alors éteindre et allumer le port USB avec les commandes suivantes :

```
cd ~/hub-ctrl.c
sudo ./hub-ctrl -h 0 -P 3 -p 0
sudo ./hub-ctrl -h 0 -P 3 -p 1
sudo ./hub-ctrl -h 0 -P 3 -p 0
sudo ./hub-ctrl -h 0 -P 3 -p 1
```

Note 2 : Même avec une bonne connexion il arrive que certains paquets n'arrivent pas à destination. C'est notamment le cas lorsque les bras ont des comportements rapides. Un message d'erreur est alors affiché mais il n'aura pas d'incidence sur le fonctionnement du système.

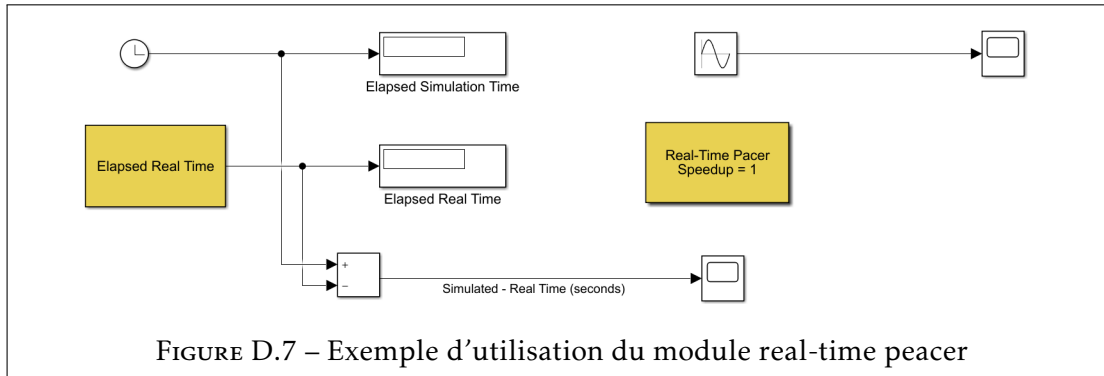
### D.3.4 Problèmes et solutions rencontrés au cours de l'utilisation

On détaille dans cette sous-section tous les problèmes, astuces et solutions rencontrés et proposées au cours des différents essais.

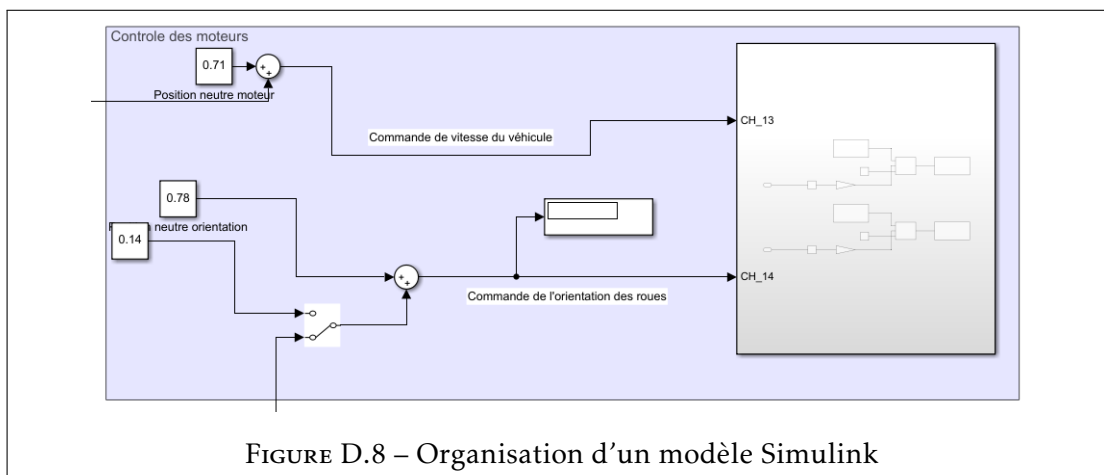
- *Inversion des hélices* : Lors des expérimentations, un effet surprenant est apparu. Le drone s'inclinait bien dans les deux sens en gardant une position constante. Cependant, sur l'un des deux axes, le drone était amorti dans son orientation alors qu'il divergeait sur l'autre. Une analyse en simulation a permis de mettre en évidence exactement le même phénomène. La cause était une inversion de l'emplacement des hélices horaires et des hélices anti-horaires. Sur un drone classique cela aurait eu une conséquence désastreuse sur le lacet en ayant un système divergeant (donc tournant de plus en plus vite sur lui-même). Dans notre cas, le lacet est principalement géré par l'orientation des bras et cela n'a donc pas eu d'effet dramatique à ce niveau-là. En revanche, lors de l'inclinaison le terme de rétroaction des hélices a influencé le système dans le mauvais sens, ce qui a conduit à la divergence sur l'un des axes et au sur-amortissement sur l'autre.
- *Biais sur le lacet* : Comme cela a été décrit précédemment, le biais n'est pas compensé par le magnétomètre, il faut donc le réinitialiser manuellement. En volière on se sert de l'orientation du drone au sol mesurée par l'OptiTrack. Cette valeur permet de créer un biais pour compenser l'erreur sur le lacet. Dans le détail, la valeur de biais est calculée à partir de l'intégrale de la moyenne glissante de l'erreur entre la mesure de l'Optitrack et celle de l'IMU. Le biais est ensuite converti en matrice de rotation qui est multipliée à la matrice de rotation mesurée par l'IMU. Cela permet de faire converger l'erreur de rotation à 0;
- *Impact des publishers sur la fréquence du système* : Pour afficher ou enregistrer les données depuis le nœud local du drone, on utilise des publishers qui vont envoyer des informations à la station sol. Un publisher par signal était utilisé. Un effet intéressant est alors survenu : la fréquence du nœud de commande sur le drone a chuté aléatoirement provoquant de grands problèmes de latences et donc d'oscillations du drone. On propose de pallier ce problème en limitant le nombre de publishers et en augmentant la taille des messages. A l'avenir on peut également proposer d'imaginer un message d'une taille importante contenant toutes les informations dont on a besoin. Ainsi le système enverra des informations sur un seul publisher limitant l'impact sur la bande passante;

2. <https://www.baptiste-donaux.fr/raspberry-pi-gerer-l'alimentation-des-ports-usb/>

- *real time pacer* : Dans l'application dédiée au système embarqué décrite, on est souvent amené à tester le système en ligne avant de le télécharger sur le système (avec le bouton play). Deux cas se présentent alors, soit le programme tourne plus lentement que la réalité (à cause de la lenteur du réseau par exemple) soit l'inverse. Dans le deuxième cas il est alors intéressant de ralentir la simulation pour avoir un système fonctionnant en temps réel. Cela est possible à l'aide du module de Matlab : real-time pacer<sup>3</sup>. Un exemple d'utilisation de ce module peut être trouvé en Figure D.7 ;



- *Organisation du modèle Simulink* : Afin de pouvoir mieux organiser le modèle, il est conseillé de respecter quelques astuces simples. Pour commencer, on peut créer des sous-groupes dans le modèle. On pourra par exemple en faire un pour la commande, un pour la loi d'allocation puis un pour tous les capteurs et enfin un autre pour les actionneurs. Il est également conseillé de faire des sous-groupes pour les fonctions (si l'on a plusieurs contrôleurs PD par exemple). D'autre part, on peut également mettre des aires afin de déplacer toute une partie du modèle en une fois. On montre un exemple en Figure D.8 (l'aire est en bleu).



- *Echantillonnage* : Comme cela a été décrit précédemment, il est important d'échantillonner le modèle avec le bon intervalle de temps. Un intervalle de temps trop long engendre des latences au cœur du nœud du NA+RPI ce qui donne une dynamique oscillatoire qui peut même diverger et être destructrice. À l'inverse, une dynamique trop rapide ne pourra pas être gérée par l'ordinateur de bord car il doit communiquer des informations par wifi via les publisher. Une impossibilité de délivrer les messages peut également faire globalement ralentir le système ;

3. <https://fr.mathworks.com/matlabcentral/fileexchange/29107-real-time-pacer-for-simulink>

- *SpaceMouse®* : La souris spatiale est un produit de l'entreprise 3DConnexion© qui permet la rotation d'un objet sur les 6 DDL tout comme le drone. Outre son intérêt pour la CAO, le produit est également directement compatible avec Simulink et Ardupilot. Cette souris spéciale peut donc être utilisée pour piloter le drone, y compris en simulation ;
- *Maintien d'allumage de bouton* : Lors des simulations ou encore des tests pratiques il est souvent intéressant (ou inévitable en fonction du matériel) de pouvoir utiliser des boutons poussoirs. La SpaceMouse® par exemple possède deux boutons poussoirs qui peuvent être très utiles. Le moyen le plus simple pour avoir un signal binaire qui reste allumé avec une première pression et qui s'éteint avec un deuxième appui sur le bouton, est la bascule D. Le schéma Simulink pour celle-ci est donné en Figure D.9 ;

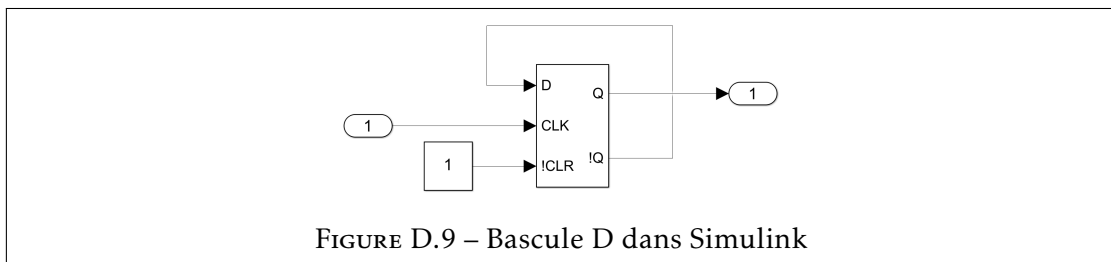


FIGURE D.9 – Bascule D dans Simulink

- *Rotation des hélices* : Pour simuler la rotation des hélices dans Simscape, deux possibilités existent. Premièrement, on peut utiliser l'actionnement de la liaison entre hélice et moteur, en contrôlant directement la position de l'hélice. Deuxièmement on peut commander le couple de rotation du système. Pour que le système ne tourne pas de plus en plus vite, et rester fidèle à la réalité, il faut ajouter un coefficient de frottement lié à la résistance de l'air ;
- *Suivi de trajectoire* : Effectuer un suivi en trajectoire présente un intérêt important pour valider les performances du système. Trois manières différentes pour effectuer un suivi en trajectoire ont été testées au cours des tests réels et en simulation, ils sont détaillés ci-après :
  1. *Recherche des waypoints* : Le principe consiste à échantillonner la fonction correspondant à la forme à suivre (par exemple un cercle). Chaque échantillon sera un point dans l'espace à suivre par le drone. Une fois le point atteint le drone cherchera le point suivant. Pour fluidifier le vol on pourra choisir de créer une zone dans laquelle le drone peut se situer autour du point ciblé avant de passer au point suivant. Cela permet de limiter le temps entre les points en cas de faibles erreurs de positionnement. Cette méthode, bien qu'efficace, entraîne parfois une non fluidité du vol du drone. En effet, par son élan, le drone continue son vol et ne rejoindra pas tout de suite le point suivant. Il faudra alors attendre avant qu'il ne corrige sa position, entraînant des périodes de suivis de trajectoires non fluides.
  2. *vitesse constante avec changement de cap* : Dans ce mode de fonctionnement, la vitesse horizontale du drone est constante, et c'est son cap qui changera en fonction de sa position sur le cercle. L'inconvénient de cette technique est une difficulté pour reprendre le contrôle du drone en cas de problème. Si le drone n'est pas orienté dans le bon sens il est plus compliqué de pouvoir intervenir en urgence.

3. *Waypoints échantillonnés par le temps* : Cette dernière méthode est similaire à la première à l'exception du fait que la cible du drone changera en fonction du temps et non pas du fait (ou non) que le drone ait atteint le point. Cela fluidifiera grandement le suivi de trajectoire, cependant la consigne de position du drone changera quoi qu'il arrive, peu importe la position actuelle du drone. Le cas échéant, cette méthode est donc plus dangereuse car l'on n'attendra pas que le drone soit en position avant de continuer le suivi de trajectoire.
- *Effectuer une sauvegarde du NA+RPI* : De temps en temps il peut être important de faire une sauvegarde totale du système. Pour ce faire le plus simple est de créer une image du système avec un logiciel comme Win32 Disk Imager. Il sera alors possible de recharger cette image sur une carte SD de taille identique ou supérieure. En cas de carte SD plus petite il faudra alors utiliser une astuce supplémentaire. L'image doit d'abord être montée virtuellement, réduite puis démontée. Il est ensuite possible de réduire la taille de l'image en tant que telle. Cependant, l'opération change l'*Universal Unique Identifier* (UUID) de l'image. Pendant qu'elle est montée et qu'elle est redimensionnée, il faut donc également changer cet identifiant. Voilà les commandes à effectuer pour cette opération (mode super-utilisateur) :

```
losetup --find --partscan save_romeo.img  
gparted /dev/loop0
```

Pour monter l'image et ouvrir l'éditeur de partitions. On redimensionne la partition puis change l'UUID en faisant un clic droit sur la partition et en sélectionnant l'onglet concerné dans le menu déroulant. Puis, en console :

```
fdisk /dev/loop0
```

entrer  $p$  et noter le `BlockSize` et `EndBlockNumberOfLastLoopPartition` (que l'on nommera ici `EBNOLLP`).

Enfin, toujours en invité de commande on exporte la nouvelle image :

```
dd if=/dev/loop0 of=shrunk.img bs=BlockSize count=EBNOLLP+1  
losetup --detach /dev/loop0
```

Il est ensuite possible de charger l'image réduite sur la carte SD.