Conception et commande de robots parallèles volants

Damien Six

▶ **To cite this version:**

Damien Six. Conception et commande de robots parallèles volants. Automatique / Robotique. École centrale de Nantes, 2018. Français. NNT : 2018ECDN0046 . tel-02384129

HAL Id: tel-02384129
https://theses.hal.science/tel-02384129

Submitted on 28 Nov 2019

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.

UNIVERSITE BRETAGNE LOIRE MATHSTIC

CENTRALE NANTES

THESE DE DOCTORAT DE

L'ÉCOLE CENTRALE DE NANTES
COMUE UNIVERSITE BRETAGNE LOIRE

ECOLE DOCTORALE N° 602
Mathématiques et Sciences et Technologies de l'Information et de la Communication
Spécialité : Automatique, productique et robotique

Par

Damien SIX

Conception et commande de robots parallèles volants

Thèse présentée et soutenue à Nantes, le 4 Décembre 2018
Unité de recherche : Laboratoire des Sciences du Numérique de Nantes

Rapporteurs avant soutenance :

Nicolas Andreff	Professeur des universités – Université de Franche-Comté
Pascal Morin	Professeur des universités – Sorbonne Université

Composition du Jury :

Jean-Pierre Merlet *Président*	Directeur de recherche - INRIA
Anibal Ollero	Professeur – Université de Séville
Isabelle Fantoni	Directrice de recherche – LS2N
Sébastien Briot *Directeur de thèse*	Chargé de recherche – LS2N
Philippe Martinet *Co-encadrant de thèse*	Directeur de recherche – INRIA
Abdelhamid Chriette *Co-encadrant de thèse*	Maître de conférence - ECN

À Denis, à papy et ses jeux de mots laids,

Remerciements

Ce manuscrit est le résultat de trois années de travail et je souhaite remercier toutes les personnes qui m'ont accompagné dans cette aventure.

Tout d'abord, je tiens à remercier mon superviseur de thèse Sébastien Briot pour ses conseils et sa confiance. Nos échanges ont toujours été riches d'enseignement. Sa passion pour la robotique et son implication ont alimenté ma motivation tout au long de la thèse. Merci aussi pour sa patience face à un thésard parfois têtu.

Je remercie également mon co-superviseur Abdelhamid Chriette. En particulier pour ses efforts qui ont permis de trouver les fonds nécessaires à la mise en place des installations expérimentales, sans lesquelles ces travaux de recherche n'auraient pu être menés à terme.

Je souhaite remercier mon second co-superviseur Philippe Martinet. Les circonstances l'ont amené à être moins présent au quotidien mais ses conseils se sont toujours avérés avisés.

Merci à Stéphane Jolivet pour la fabrication et l'assemblage des prototypes. Je m'excuse pour toutes les fois où je suis revenu avec un morceau de robot à réparer. Merci à Sébastien Rouquet pour m'avoir aidé à mettre l'activité expérimentale des drones en place et à Julian Erskine de reprendre le flambeau avec autant de passion.

Merci au comité de défense et en particulier à Nicolas Andreff et Pascal Morin pour leurs retours précis, bienveillants et constructifs sur mes travaux.

Merci à toute l'équipe ARMEN pour la bonne humeur, les échanges et la sympathie. De la bonne recherche dans une excellente ambiance, que demander de plus ?

Merci à Sylvain, Konstantin, Bogdan, Anoop, Abhilash pour ces quatre années passées ensemble.

Merci au personnel du LS2N et de l'Ecole Centrale de Nantes, pour les repas partagés, pour les sorties karting et pour l'aide apportée au quotidien.

Je tiens à remercier également mes amis, dont l'affection m'est indispensable. Merci aux rennais : Vincent, Salam, Ki, Mat, FX, etc. Merci à Clémence et Benoît pour les vacances à la Réunion et merci à ceux qui nous ont accompagné : Fanny, Flo, Hugues, Aurélie, Lydie et re Vincent. Merci à Dim, Emeline et Thomas qui font partie de la famille. Thomas, en partageant ton expérience, tu m'as aidé à prendre du recul sur les difficultés rencontrées et à trouver Sigmar au cœur du chaos. J'espère que ce remerciement me vaudra de l'XP.

En parlant de famille, je remercie aussi mes parents et leur palanquée de descendants pour leur amour et leur soutien. Vous avez construit ce que je suis aujourd'hui.

J'ai gardé mon plus grand remerciement pour la fin. À celle qui chaque jour ou presque était là. Celle qui a vécu de nombreux soirs ma longue liste des frustrations sachant que ça devait sortir, ou mes explications sur la traversée de singularités à l'aide de fourchettes. La première personne à qui j'avais envie de montrer la vidéo "Regarde ça vole !". Et celle qui était au premier rang quand j'étais cloué plusieurs semaines au lit. Pour ton soutien, pour ta patience, pour ton amour, pour avoir vécu cette expérience avec moi, Élise, je ne pourrai jamais assez te remercier.

Résumé

La manipulation aérienne est l'un des défis de la robotique au cours de cette dernière décennie. L'un des freins au développement des manipulateurs aériens est l'autonomie limitée des drones, réduite par la charge et la consommation électrique du manipulateur embarqué. Une solution pour dépasser cette limite passe par la collaboration de plusieurs drones dans une tâche de manipulation.

Cette thèse porte sur la conception et la commande d'un nouveau type de véhicule autonome aérien destiné à la manipulation. Le concept consiste à faire collaborer plusieurs drones, en particulier des quadricoptères, au travers d'une architecture passive polyarticulée. Le robot ainsi obtenu est une fusion entre l'architecture passive d'un robot parallèle et plusieurs drones.

L'étude du modèle dynamique de cette classe de robots met en avant un découplage dans le modèle dynamique. Ce découplage permet la conception d'une commande en cascade qui assure la stabilisation et le suivi de trajectoire pour ces robots.

Deux cas d'étude sont ensuite déclinés dans cette thèse : un robot parallèle volant à deux drones et un robot parallèle volant à trois drones. Pour ces deux robots, une simulation numérique est effectuée afin de valider le fonctionnement de la commande. Ces simulations ont également permis de valider la possibilité de modifier la configuration de l'architecture passive en vol.

Les travaux ont été portés avec succès jusqu'au stade expérimental pour le robot volant à deux drones.

Table des matières

Introduction **1**

1 État de l'art **5**
1.1 Modélisation et commande d'un quadricoptère 6
1.2 Les robots manipulateurs aériens 13
1.3 Robots parallèles, modèles cinématiques, dynamiques et singularités 17
1.4 Conclusion 33

2 Modélisation et commande d'un robot parallèle volant **35**
2.1 Un nouveau concept de manipulateur aérien 36
2.2 Cinématique d'un robot parallèle volant 38
2.3 Modèle dynamique d'un robot parallèle volant 43
2.4 Commande d'un robot parallèle volant 51
2.5 Conclusion 62

3 Étude de deux architectures de robots parallèles volants **65**
3.1 Robot parallèle volant à trois drones 66
3.2 Robot parallèle volant à deux drones 82
3.3 Conclusion 114

4 Mise en œuvre expérimentale **115**
4.1 Description du prototype 116
4.2 Identification des paramètres dynamiques des drones 123
4.3 Résultats expérimentaux 140
4.4 Conclusion 149

5 Singularités **151**
5.1 Singularités du modèle dynamique inverse et critère de traversée 152
5.2 Commande pour la traversée de la singularité 156
5.3 Simulation de traversée d'une singularité 162
5.4 Conclusion 169

Conclusion **171**

A Calcul des modèles cinématiques et dynamiques d'un robot défini à l'aide de la convention DHKK **177**
A.1 Calcul de la matrice jacobienne 177
A.2 Calcul du modèle dynamique 179

B Estimation de la position et orientation des drones **183**
B.1 Estimation des orientations par la centrale inertielle 183
B.2 Fusion de données pour l'estimation de la position et de l'orientation des drones 185

C Article sur la méthode de traversée des singularités pour les robots parallèles. **191**

D Démonstration pour la solution à l'équation cinématique d'ordre 2 en singularité **203**

Bibliographie **216**

Table des figures

1 Illustration de la première utilisation de drones. 2

1.1 Illustration du repère attaché au quadricoptère et de la position des rotors sur le quadricoptère PELICAN. 7
1.2 Schéma des forces et moments exercés par un rotor 8
1.3 Schéma général d'une commande classique pour un quadricoptère. 11
1.4 Exemples de drones munis d'un outil rigide. 14
1.5 Exemples de drones associés à un manipulateur à un degré de liberté. 14
1.6 Exemples de drones associés à un manipulateur à plusieurs degrés de liberté. 15
1.7 Exemples de manipulation en coopération par plusieurs quadricoptères. . . . 17
1.8 Schéma général d'un robot parallèle. 18
1.9 Exemples de robots parallèles plans à deux degrés de liberté. 19
1.10 Exemple de robot parallèle plan à trois degrés de liberté. 19
1.11 Exemples de robots parallèles permettant trois translations de l'effecteur dans l'espace. 20
1.12 Exemples de robots parallèles à six degrés de liberté. 21
1.13 Exemple de robot parallèle à câbles (Fraunhofer IPA). 21
1.14 Méthode de résolution du modèle géométrique inverse d'un robot parallèle. . 23
1.15 Modes de fonctionnement d'une jambe et modes d'assemblage pour le robot parallèle à cinq barres. 24
1.16 Architecture virtuelle pour le calcul du modèle dynamique d'un robot parallèle 26
1.17 Exemple de séparation de l'espace de travail d'un robot parallèle (cinq barres) par des singularités de Type 2 pour différents mode de fonctionnement. . . . 28
1.18 Exemple de robot parallèle (cinq barres) en singularité de Type 2. 29
1.19 Jambe du Triptéron en singularité LPJTS. L'articulation grisée est actionnée. 29
1.20 Robot cinq barres en singularité de Type 2. 32

2.1 Exemple de robot aérien à câble et schéma de structure virtuelle équivalente pour un câble. 37
2.2 Exemple de robot parallèle volant à trois drones. 37
2.3 Schéma général d'un robot volant à n drones. 38
2.4 Séparation virtuelle des drones et de l'architecture passive du robot parallèle volant. 40

2.5 Architecture virtuelle équivalente pour la définition de la position et de l'orientation de la plate-forme. 42
2.6 Illustration du bilan des forces s'exerçant au centre de gravité d'un drone. . . 45
2.7 Exemple, dans le plan, de rotation du drone solution de l'équation (E2.31). . 54
2.8 Schéma de la loi de commande pour le robot parallèle volant. 59

3.1 Vue CAO du robot parallèle volant à trois drones. 67
3.2 Paramétrisation du robot parallèle volant à trois drones. 67
3.3 Vue en coupe du robot parallèle volant à trois drones. 68
3.4 Exemple d'une configuration singulière pour le robot parallèle volant à trois drones. 71
3.5 Robot parallèle volant à trois drones : lieu des singularités. 73
3.6 Schéma de l'interaction entre les deux logiciels de simulation. 75
3.7 Simulation du robot parallèle volant à trois drones : trajectoire désirée et suivie pour les coordonnées $\mathbf{q}_p$. 78
3.8 Simulation du robot parallèle volant à trois drones : trajectoire désirée et suivie pour les coordonnées $\mathbf{q}_a$. 79
3.9 Simulation du robot parallèle volant à trois drones : convergence de l'erreur $\mathbf{e}_p$. 80
3.10 Simulation du robot paralèlle volant à trois drones : convergence de l'erreur $\mathbf{e}_a$. 81
3.11 Schéma du méchanisme Biglide. 83
3.12 Vue CAO du robot parallèle volant à deux drones. 84
3.13 Paramètres cinématiques du robot parallèle volant à deux drones. 85
3.14 Robot parallèle volant à deux drones : illustration des solutions du modèle géométrique dans le plan $\mathcal{P}_p$. 87
3.15 Exemple d'une configuration singulière du robot volant à deux drones. 90
3.16 Robot parallèle volant à deux drones dans une configuration telle que le plan $\mathcal{P}_p$ soit confondu avec le plan d'équation $x = 0$. 92
3.17 Robot parallèle volant à deux drones : bilan des forces agissants sur le drone 1. 93
3.18 Schéma de la commande du robot volant à deux drones. 102
3.19 Schéma cinématique du robot parallèle volant à deux drones muni d'un pivot supplémentaire. 105
3.20 Schéma de la commande du robot volant à deux drones avec liaison pivot supplémentaire. 107
3.21 Simulation du robot parallèle volant à deux drones : trajectoire désirée et suivi des coordonnées $\mathbf{q}_x$. 110
3.22 Simulation du robot parallèle volant à deux drones : trajectoire désirée et suivi des coordonnées $\mathbf{q}_a$. 111
3.23 Simulation du robot parallèle volant à deux drones : convergence de l'erreur $\mathbf{e}_x$. 112
3.24 Simulation du robot parallèle volant à deux drones : convergence de l'erreur $\mathbf{e}_a$. 112
3.25 Simulation du robot parallèle volant à deux drones : erreur des coordonnées ψ et x_G. 112

3.26 Simulation du robot parallèle volant à deux drones : vitesse de rotation des hélices sur les drones. 113

4.1 CAO du drone PELICAN. 117
4.2 CAO des pièces de raccordement aux drones du prototype. 118
4.3 CAO des bras de la structure passive du prototype. 118
4.4 CAO du prototype assemblé. 119
4.5 Photographie du prototype assemblé. 119
4.6 Schéma des équipements et communications du prototype. 120
4.7 Photographie d'une caméra infrarouge. 121
4.8 Photographie de la zone expérimentale. 122
4.9 Photographie d'un drone muni des marqueurs infrarouges. 122
4.10 Photographies des drones associés aux pièces de liaisons du prototype. 132
4.11 Validation croisée de l'identification du drone 1 à partir des données MOCAP (forces). 136
4.12 Validation croisée de l'identification du drone 1 à partir des données MOCAP (moments). 137
4.13 Validation croisée de l'identification du drone 1 à partir des données IMU (forces). 138
4.14 Validation croisée de l'identification du drone 1 à partir des données IMU (moments). 139
4.15 Exemple de trajectoire suivie par le robot pour les expérimentations. 142
4.16 Expérimentations : erreurs de suivi de la trajectoire en 54 s. 143
4.17 Expérimentations : erreurs de suivi de la trajectoire en 34 s. 144
4.18 Expérimentations : erreurs de suivi de la trajectoire en 26 s. 145
4.19 Expérimentations : commande envoyée aux moteurs du drone 1 pour la trajectoire en 54 secondes (filtrée). 146
4.20 Expérimentations : évolution des coefficients d'adaptation pour la trajectoire en 54 secondes. 147
4.21 Expérimentations : erreurs de suivi de la trajectoire en mode d'assemblage haut. 148

5.1 Architecture passive soumise aux forces $\mathbf{f}_{p,i}$ appliquées aux points O_i 153
5.2 Comparaison d'une singularité de Type 2 du Biglide (a) et d'une singularité du robot parallèle volant à deux drones (b). 154
5.3 Exemple de surcontrainte générée localement en singularité. 154
5.4 Accélération et projection de l'accélération dans le plan vertical de l'articulation ② en singularité. 158
5.5 Illustration des fonctions σ_s et $\mathcal{Q} \circ \sigma_s$ pour $\epsilon = 0.01$ et $n_s = 3$. 159
5.6 Schéma de la boucle lente (architecture passive) adaptée à la traversée des singularités. 162

5.7 Comparaison des configurations de traversée d'une singularité du robot parallèle volant à deux drones. 163
5.8 Trajectoire pour la traversée de la singularité pour les coordonnées $\mathbf{q}_p$. 165
5.9 Trajectoire de traversée de la singularité pour le robot parallèle volant à deux drones. 166
5.10 Forces de poussée et orientations des drones attendues lors de l'exécution de la trajectoire. 167
5.11 Simulation de traversée de singularité : erreur de suivi des positions $\mathbf{e}_x$. . . . 167
5.12 Simulation de traversée de singularité : erreur de suivi des orientations $\mathbf{e}_a$. . 168
5.13 Simulation de traversée de singularité : angles d'orientation calculés par la commande. 168
5.14 Simulation de traversée de singularité : forces de poussée calculées par la commande. 168

A.1 Repères définis selon la convention DHM pour le robot parallèle volant à deux drones. 178

Liste des tableaux

2.1 Coordonnées DHKK d'une strucure virtuelle équivalente, permettant de définir la position et l'orientation de la plate-forme dans l'espace. 42

3.1 Coordonnées DHKK de l'architecture passive du robot volant à trois drones. 69
3.2 Paramètres dynamiques du robot volant à trois drones. 76
3.3 Les paramètres DHKK du robot parallèle volant à deux drones. 86
3.4 Table des notations abrégées des fonctions sinus et cosinus. 89
3.5 Tableau des paramètres dynamique du robot volant à deux drones. 109

4.1 Résultats d'identification du drone 1 à partir des données MOCAP 133
4.2 Résultats d'identification du drone 1 à partir des données IMU 134
4.3 Résultats d'identification du drone 2 à partir des données IMU 135
4.4 Erreur sur le modèle lors de la validation croisée. 136
4.5 Tableau des paramètres dynamique du robot volant à deux drones en expérimentation. 141
4.6 Valeurs moyennes et maximales de la valeur absolue de l'erreur à différentes vitesses d'exécution d'une trajectoire pour le robot à deux drones. 146

5.1 Résultat de l'algorithme d'optimisation pour la traversée de la singularité. . 163
5.2 Expressions polynomiales définissant la trajectoire de traversée de la singularité.164

Nomenclature

Symboles généraux

$\dot{x}, \ddot{x}$	Première et seconde dérivée temporelle de la variable x
ℓ_{AB}	Distance du point A au point B
$\hat{x}$	L'estimation d'une variable x
$\mathbf{X}$	Une matrice
$\mathbf{x}$	Un vecteur
$\mathbf{X}^{+}$	La peudo-inverse de Moore Penrose de la matrice $\mathbf{X}$
$\mathbf{X}^{T}$	La transposée d'une matrice $\mathbf{X}$
x	Une variable scalaire
$\mathcal{I}_n$	La matrice identité de dimension n

Modélisation d'un robot parallèle volant

$\boldsymbol{\eta}_i$	Vecteur regroupant les coordonnées d'orientation du drone i : ϕ_i, θ_i et ψ_i
$\boldsymbol{\Omega}_i$	Vecteur regroupant le carré des vitesses de rotation des rotors du drone i
$\boldsymbol{J}_\omega$	Sous-matrice d'une matrice jacobienne $\mathbf{J}$ associée aux vitesses de rotations
$\boldsymbol{J}_i$	Matrice jacobienne cinématique de la jambe i
$\boldsymbol{J}_v$	Sous-matrice d'une matrice jacobienne $\mathbf{J}$ associée aux vitesses de translation
$\boldsymbol{t}_i$	Torseur cinématique de du dernier corps de la jambe i
$\mathbf{c}$	Vecteur regroupant les effets centrifuges, de Coriolis et de gravité d'un modèle dynamique
$\mathbf{f}$	Vecteur regroupant les forces de poussée des drones exprimées dans leurs repères locaux respectifs ${}^{i}\mathbf{f}_i$
$\mathbf{f}_z$	Vecteur regroupant les composantes selon $\mathbf{z}_i$ des forces de poussée des drones ${}^{i}\mathbf{f}_i$ exprimées dans leurs repères locaux respectifs

$\mathbf{g}$	Vecteur de gravité
$\mathbf{J}_p$	Matrice jacobienne cinématique de l'architecture passive, regroupant les matrices $\mathbf{J}_{v,i}$
$\mathbf{M}$	Matrice d'inertie généralisée d'un modèle dynamique
$\mathbf{m}$	Vecteur regroupant les moments dynamiques exercés par les hélices des drones exprimées dans leurs repères locaux respectifs ${}^{i}\mathbf{m}_i$
$\mathbf{q}$	Vecteur des coordonnées du robot
$\mathbf{q}_a$	Vecteur regroupant les coordonnées d'orientation des drones $\boldsymbol{\eta}_i$
$\mathbf{q}_p$	Vecteur des coordonnées de l'architecture passive du robot
$\mathbf{R}_r$	Forme réduite de la matrice $\mathbf{R}_t$
$\mathbf{R}_t$	Matrice regroupant les matrices de rotation $\mathbf{R}_i$ des repère locaux associés aux drones
$\mathbf{w}_{p,i}$	Torseur dynamique appliqué au point O_i sur le dernier corps de la jambe i, composé d'une force $\mathbf{f}_{p,i}$ et d'un moment $\mathbf{m}_{p,i}$
$k_{d,i}$	Coefficient de traînée du drone i
$k_{t,i}$	Coefficient de poussée du drone i
L_p	Lagrangien de l'architecture passive
m_i	Masse du drone associé à la jambe i
n	Nombre de jambes du robot parallèle volant
n_p	Nombre de coordonnées indépendantes de l'architecture passive
$\mathcal{B}_p$	La plate-forme du robot
$\mathcal{B}_{ij}$	Le corps j de la jambe i du robot
$\mathcal{D}_i$	Le drone associé à la jambe i
$\mathcal{F}_i$	Le repère associé au drone $\mathcal{D}_i$ d'origine O_i et d'axes $\mathbf{x}_i$, $\mathbf{y}_i$ et $\mathbf{z}_i$
$\mathcal{F}_o$	Le repère d'origine au sol d'origine O et d'axes $\mathbf{x}_o$, $\mathbf{y}_o$ et $\mathbf{z}_o$
$\mathcal{F}_p$	Le repère associé à la plate-forme $\mathcal{B}_p$ d'origine P et d'axes $\mathbf{x}_p$, $\mathbf{y}_p$ et $\mathbf{z}_p$
${}^{j}\mathbf{u}$	Un vecteur exprimé dans le repère associé $\mathcal{F}_j$

Commande

$\boldsymbol{\nu}$	Une commande auxiliaire
$\boldsymbol{\sigma}$	Un vecteur de variables glissantes

$\mathbf{e}$	Une erreur de suivi $\mathbf{e} = \mathbf{q} - \mathbf{q}^d$
$\mathbf{q}^d$	Le vecteur de trajectoire désirée pour le vecteur de coordonnées $\mathbf{q}$
k ou $\mathbf{K}$	Un gain ou une matrice de gain
V	Une fonction de Lyapunov

Paramètres dynamiques et identification

$\boldsymbol{\chi}$	Vecteur des paramètres dynamiques
$\boldsymbol{\rho}$	Vecteur d'erreur de l'identification dynamique
$\mathbf{I}$	Matrice d'inertie d'un corps rigide
$\mathbf{ms}$	Premier moment d'inertie d'un corps rigide
$\mathbf{W}$	Matrice de régression pour l'identification dyanmique
$\sigma_{\boldsymbol{\chi}_i}$ et $\%\sigma_{\boldsymbol{\chi}_i}$	Écart-type et écart type relatif d'un paramètre dynamique identifié $\boldsymbol{\chi}_i$
m	Masse d'un corps rigide
HLP	High Level Processor (microcontrôleur du PELICAN)
LLP	Low Level Processor (microcontrôleur du PELICAN)
SPI	Serial Peripheral Interface

Abréviations

CAO	Conception Assistée par Ordinateur
DHKK	Denavit Hartenberg modifiée par Khalil et Kleinfinger
IDIM	Inverse Dynamique Identification Model (Modèle dynamique inverse d'identification)
IMU	Inertial Measurement Unit (centrale inertielle)
LS	Least Squares (moindres carrés)
MOCAP	MOtion CAPture system (système de capture de mouvements)
PID	Proportionelle Intégrale Dérivée
ROS	Robot Operating System (ensemble d'outils informatiques open source permettant de développer des logiciels pour la robotique)
SVD	Singular Values Decomposition (décomposition en valeurs singulières)
TLS	Total Least Squares (moindres carrés totaux)
WLS	Weigthed Least Squares (moindres carrés pondérés)
WTLS	Weigthed Total Least Squares

Introduction

Origine et applications des robots volants autonomes

La première utilisation d'un véhicule autonome volant remonte à Juillet 1849 (Fig. 1). Les forces autrichiennes assiégeant Venise ont tenté d'attaquer la ville à l'aide de 200 ballons incendiaires, transportant des bombes associées à des mèches à retardement. La réussite de l'opération fut compromise car le vent tourna, ramenant une partie des bombes vers les lignes autrichiennes.

Les premiers prototypes d'avions radio-commandés ont vu le jour pendant et après la première guerre mondiale. Depuis, l'essentiel du développement des drones fut destiné à des applications militaires : cibles d'entraînement, récoltes de résultats dans le cadre d'essais nucléaires, plate-forme de reconnaissance, et plus récemment pour le combat.

Depuis les années 2000, les avancées en matière de matériel, électronique, capteurs et batteries ont alimenté le développement des micro-drones. Un micro-drone est un drone présentant une masse inférieure à 5 kg et une envergure inférieure à 0,7 m. L'émergence de plate-formes petites et peu chères a fait se multiplier les sujets de recherches et les applications civiles des drones. En France, l'essentiel du marché est celui des drones utilisés pour le loisir. Le micro-drone le plus courant est le quadricoptère. Les applications professionnelles sont principalement retrouvées dans [L'essor des drones aériens civils en France, 2016] :

- l'audiovisuel et les médias pour la majeure partie,
- l'inspection de bâtiments et d'infrastructures,
- l'agriculture pour la surveillance des cultures et marginalement pour l'épandage,
- la sécurité et la surveillance.

Ces usages évoluent avec l'apparition de nouvelles applications : livraison de colis, modélisation 3D de bâtiments, diffusion de données, sauvetage en mer, etc.

Un domaine de recherche actif

Les deux dernières décennies ont vu les recherches autour des drones se multiplier, notamment pour :

- la modélisation [Martin et Salaün, 2010; Bristeau et al., 2009] et la commande [Jayakrishnan, 2016; Carino et al., 2015; Chriette et al., 2015] des quadricoptères,

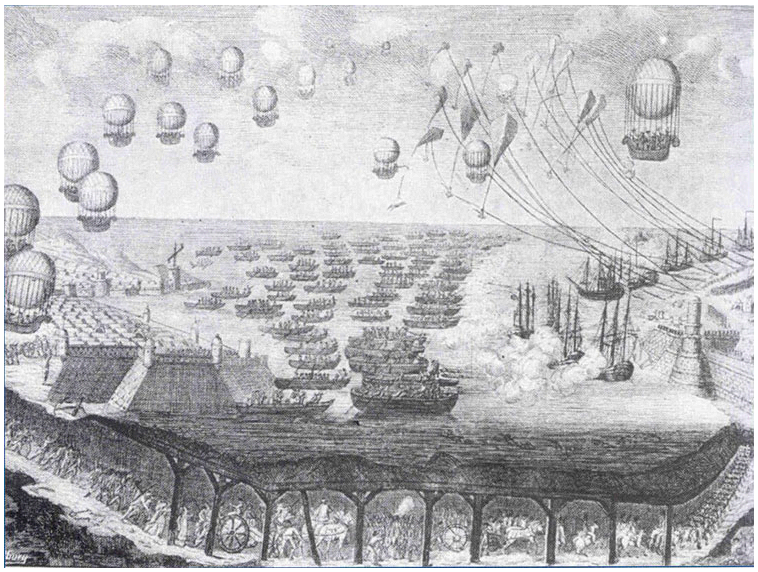


FIGURE 1 – Illustration de la première utilisation de drones.

- le développement de nouvelles architectures à voilures tournantes [Tognon et Franchi, 2018; Zhao et al., 2018; Rajappa et al., 2015; Ryll et al., 2013; Romero et al., 2007; Kendoul et al., 2006] ou bio-inspirées [Chirarattananon et al., 2016; Rosen et al., 2016],
- la navigation autonome [Eliker et al., 2018; Silva et al., 2017; Paternain et al., 2017; Hehn et D'Andrea, 2015],
- le vol en formation [Preiss et al., 2016; Alonso-mora et al., 2016; Yu et al., 2013],
- l'association d'un drone et d'un manipulateur (voir partie 1.2),
- la coopération pour le transport et la manipulation de charges par liaisons directes sur l'objet ou à l'aide de câbles (voir également partie 1.2).

La manipulation aérienne présente de nombreux défis scientifiques : autonomie des véhicules, stabilité de vol pendant la manipulation, sécurité, précision de la manipulation, charge utile... Au cours de la dernière décennie, des prototypes ont été développés et expérimentés et les applications industrielles sont en phase de développement. La coopération pour la manipulation est une piste pour augmenter la sécurité, la charge utile et l'autonomie de manipulateur aérien.

Cette thèse propose un nouveau type de manipulateur aérien impliquant la coopération de plusieurs drones. Ce manipulateur aérien se différencie des existants car la coopération s'effectue au travers d'une architecture poly-articulée composée de corps rigides. Cette particularité offre des capacités de manipulation inexistantes dans le cadre des autres solutions de coopération mises en œuvre.

Contributions de la thèse

Ce document présente les contributions suivantes :

- **La modélisation dynamique et la commande des robots parallèles volants :** les travaux réalisés lors de cette thèse ont permis de définir, au chapitre 2, un cadre pour le concept que nous appellerons *robot parallèle volant*. Ce robot est constitué d'une plate-forme volante sur laquelle sont fixées plusieurs jambes rigides articulées passives. Un drone est fixé sur chaque jambe, à l'extrémité opposée à la plate-forme. L'actionnement des drones permet la commande de la configuration de l'architecture passive. Nous proposerons une méthode pour calculer le modèle dynamique de ce robot. Ce modèle présente une structure découplée entre la dynamique de l'architecture passive et la dynamique de rotation des drones. Ce découplage a permis de proposer une loi de commande pour la stabilisation du système et le suivi de trajectoire. Ces développements, présentés au chapitre 2, s'appliquent à toute une classe de robots volants. Le nombre de jambes et l'arrangement des liaisons pour chaque jambe peuvent être modifiés, offrant un large choix dans la conception de robots basés sur ce concept.
- **La conception de deux robots parallèles volants :** le chapitre 3 présente deux architectures de robots parallèles volants. Le premier robot conçu est composé d'une plate-forme, de trois jambes passives et de trois drones. Le modèle dynamique développé dans le cas général n'assure aucunement que le robot sera commandable en pratique. Pour cela, une étude cinématique des mobilités de l'architecture passive est nécessaire. Elle permet de valider que les efforts générés sur la plate-forme permettent la commande de ses six degrés de libertés de la plate-forme. Cette étude permet également de déterminer les configurations singulières pour lesquelles la commande de tous les degrés de liberté de l'architecture passive n'est plus assurée. Les résultats d'une telle étude sur le robot volant à trois drones sont présentés. Le second robot volant a été conçu dans l'optique d'appliquer ce concept dans un cadre expérimental. Il a été cependant réalisé selon le matériel expérimental mis à disposition pour cette thèse. Ainsi, ce prototype ne respecte pas exactement toutes les hypothèses du modèle définis au chapitre 2. Des adaptations de la loi de commande, nécessaires pour ce robot, sont alors déclinées. Pour ces deux robots, des simulations numériques de suivi de trajectoires ont été menées afin de valider ces lois de commandes.
- **La preuve de concept expérimental :** au delà de la simulation numérique, un prototype expérimental a été développé et mis en oeuvre comme preuve de concept, tel que présenté au chapitre 4. Le vol stabilisé et des suivis de trajectoires ont été effectués avec succès, démontrant la faisabilité de la mise en œuvre du concept de robot volant présenté dans cette thèse.
- **L'identification dynamique des quadricoptères :** une bonne connaissance des paramètres dynamiques des quadricoptères utilisés dans le cadre des expérimentations assure une meilleure stabilité de la loi de commande établie pour le robot volant. Une méthode d'identification de ces paramètres dynamiques, basée sur des méthodes connues pour les robots manipulateurs classiques, a été établie et appliquée sur les drones actionneurs du prototype expérimental. Cette méthode d'identification ainsi

que son application sont présentées au chapitre 4.

- **Une loi de commande pour la traversée de singularités :** l'un des avantages du concept de robot parallèle volant est la possibilité de modifier le mode d'assemblage de l'architecture passive. Cette propriété est également présente sur certains robots parallèles classiques. Chaque mode d'assemblage du robot est séparé par des singularités. Pour un robot parallèle classique, ces singularités sont un handicap limitant l'espace de travail opérationnel du robot. Il existe cependant des méthodes permettant la traversée de ces singularités. Appliquées sur un robot parallèle volant, elles rendent possible la reconfiguration en vol de ces robots, augmentant ainsi l'espace des configurations atteignables. Pour les robots présentés dans cette thèse, cette reconfiguration offre, par exemple, la possibilité de placer l'effecteur au-dessus ou en-dessous des drones. Le chapitre 5 présente une loi de commande visant à effectuer cette reconfiguration. Cette loi de commande a été validée en simulation.

Nous allons, au prochain chapitre, présenter les travaux de recherche existants sur lesquels s'appuie cette thèse.

Chapitre 1

État de l'art

Sommaire

1.1 Modélisation et commande d'un quadricoptère 6
1.1.1 Choix des variables 6
1.1.2 Modèle dynamique 8
1.1.3 Commande d'un quadricoptère 10
1.2 Les robots manipulateurs aériens 13
1.3 Robots parallèles, modèles cinématiques, dynamiques et singularités 17
1.3.1 Généralités sur les robots parallèles 18
1.3.2 Modèles géométriques et cinématiques des robots parallèles 22
1.3.3 Modèle dynamique inverse d'un robot parallèle 25
1.3.4 Singularités des robots parallèles et traversée des singularités . . . 27
1.4 Conclusion 33

Cette thèse a pour but de développer la modélisation et la commande de robots parallèles volants. Elle se situe au croisement de deux domaines :

- *la modélisation et la commande des drones, en particulier des quadricoptères,*
- *la modélisation des robots parallèles et l'étude des singularités.*

Dans ce chapitre, nous passons en revue les précédents travaux sur lesquels s'appuie cette thèse. Tout d'abord, nous rappellerons les éléments théoriques de modélisation d'un quadricoptère et décrirons différentes méthodes mises en œuvre pour sa commande. Puis, nous présenterons des prototypes élaborés pour la manipulation aérienne. Dans une dernière partie, nous aborderons les robots parallèles. Différentes architectures existantes seront présentées. Puis, nous développerons le modèle dynamique d'un robot parallèle. Ces derniers présentent des singularités similaires à celles des robots parallèles volants. Nous décrirons ces singularités ainsi que différentes méthodes élaborées pour les gérer.

1.1 Modélisation et commande d'un quadricoptère

Le concept de robot aérien développé dans cette thèse est composé d'une architecture passive associée à plusieurs drones. Même si, a priori, le choix du drone reste libre, le quadricoptère, d'usage très répandu en tant que micro-drone, a été sélectionné pour assurer l'actionnement de notre structure aérienne. Dans cette partie, le modèle classique du quadricoptère, utilisé dans la plupart des études, sera développé. Ce modèle néglige un certain nombre d'effets de la dynamique des rotors et certains effets aérodynamiques. Il s'avère cependant efficace pour les applications de commande d'un quadricoptère et sera donc utilisé dans cette étude.

1.1.1 Choix des variables

Un quadricoptère peut se modéliser comme un corps rigide sur lequel s'appliquent des forces et moments générés par quatre rotors. Le corps rigide possède quatre bras formant une croix. Les rotors sont généralement placés aux extrémités de la croix de telle façon que les axes des rotors soient normaux au plan contenant la croix (Voir Fig. 1.1). Pour définir les paramètres du quadricoptère, il est nécessaire de poser deux repères.

- Le repère de référence au sol $\mathcal{F}_o$ d'origine le point O et d'axes $\mathbf{x}_o$, $\mathbf{y}_o$ et $\mathbf{z}_o$.
- Le repère attaché au drone $\mathcal{F}_d$ d'origine le point D et d'axes $\mathbf{x}_d$, $\mathbf{y}_d$ et $\mathbf{z}_d$. Le point D est placé au centre de la croix. L'axe $\mathbf{x}_d$ est inclus dans le plan formé par la croix et aligné le long de la bissectrice de l'angle formé par deux bras. L'axe $\mathbf{z}_d$ est normal au plan formé par la croix et pointe vers le haut. L'axe $\mathbf{y}_d$ complète le repère orthonormal direct.

Les rotors sont numérotés de 1 à 4 selon la convention suivante : dans le plan formé par la croix, l'axe $\mathbf{x}_d$ par rotation directe autour de l'axe $\mathbf{z}_d$ pointe successivement vers les rotors

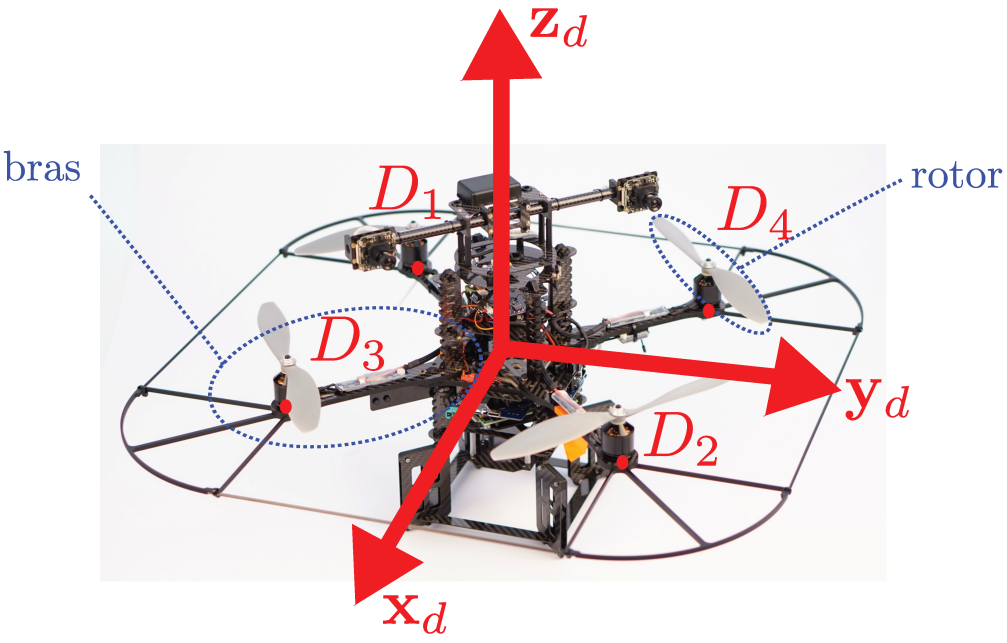


FIGURE 1.1 – Illustration du repère attaché au quadricoptère et de la position des rotors sur le quadricoptère PELICAN.

2, 4, 1 puis 3. Pour assurer la stabilité du lacet, deux hélices tournent dans un sens et deux hélices dans l'autre. Nous considérerons que les hélices 1 et 2 tournent dans le sens direct selon l'axe $\mathbf{z}_d$ et que les hélices 3 et 4 tournent en sens inverse. On appelle D_1 à D_4 les points formés par les intersections respectives des axes des rotors 1 à 4 avec le plan formé par $\mathbf{x}_d$, $\mathbf{y}_d$ et le point D.

Généralement, la distance entre le centre D du quadricoptère et chacun des points D_i est la même. De plus, l'angle entre deux bras du drone est généralement de 90°. Nous considérerons pour la suite que ces hypothèses sont respectées. Dans ce cas, une seule distance suffit alors à définir les caractéristiques géométriques du quadricoptère. La distance entre le point D_1 et le point D_3 est nommée d. La même distance sépare le point D_2 (respectivement D_3, D_4) du point D_4 (respectivement D_2, D_1).

La position et l'orientation du corps rigide dans l'espace sont définies par les coordonnées :

- $\boldsymbol{\xi} = [x\ y\ z]^T$ les coordonnées du point D dans le repère $\mathcal{F}_o$,
- $\boldsymbol{\eta}$ les coordonnées d'orientation du repère $\mathcal{F}_d$ par rapport au repère $\mathcal{F}_o$. Ces coordonnées dépendent du choix de représentation des orientations (matrices de rotation, angles d'Euler, de Bryant, quaternions,etc.) [Goldstein, 1980]. Pour cette partie, il n'est pas nécessaire de préciser la convention de représentation des orientations choisies.

La matrice de rotation du repère $\mathcal{F}_o$ vers le repère $\mathcal{F}_d$ est notée $\mathbf{R}_d$. Ainsi, la relation entre un vecteur ${}^o\mathbf{v}$, exprimé dans le repère d'origine $\mathcal{F}_o$, et le même vecteur ${}^d\mathbf{v}$, exprimé dans le repère $\mathcal{F}_d$, est donnée par :

$$ {}^o\mathbf{v} = \mathbf{R}_d\, {}^d\mathbf{v} \tag{E1.1}$$

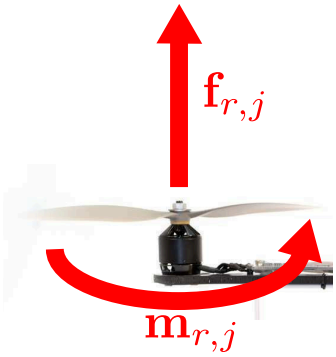


FIGURE 1.2 – Schéma des forces et moments exercés par un rotor

1.1.2 Modèle dynamique

Le torseur d'action mécanique résultant agissant sur le quadricoptère est la somme des torseurs appliqués par chacun de ses rotors. L'établissement du modèle dynamique du quadricoptère se fait alors en deux étapes. La première établit les torseurs d'actions mécaniques appliqués par chaque rotor et calcule la résultante obtenue pour les quatre rotors. La seconde calcule la dynamique du corps rigide à partir de cette résultante.

Calcul de la résultante des torseurs d'actions mécaniques appliqués par les rotors

Le modèle classique d'un rotor [Mahony et al., 2012; Bouabdallah et al., 2004a] considère que chaque rotor génère une force de poussée et un moment de traînée alignés avec son axe (Fig. 1.2). La force et le moment générés par le rotor j selon l'axe $\mathbf{z}_d$ sont respectivement notés $\mathbf{f}_{r,j}$ et $\mathbf{m}_{r,j}$ avec $j, (j = 1..4)$ le numéro du rotor. Ils sont proportionnels au carré de la vitesse de rotation de l'hélice associée. Leur expression est donnée par :

$$\begin{aligned} \mathbf{f}_{r,j} &= k_t \omega_j^2 \mathbf{z}_d \\ \mathbf{m}_{r,j} &= \pm k_d \omega_j^2 \mathbf{z}_d \end{aligned} \tag{E1.2}$$

avec

- k_t et k_d deux coefficients positifs appelés respectivement *coefficient de poussée* et *coefficient de traînée* des hélices,
- ω_j la norme de la vitesse de rotation du rotor j. ω_j est toujours positif quelque soit le sens de rotation du rotor. Celui-ci ne peut pas être modifié.

Le signe de $m_{r,j}$ dépend du sens de rotation de l'hélice selon $\mathbf{z}_d$. Il sera négatif si l'hélice tourne dans le sens direct et positif dans le cas inverse. Dans des conditions de vol réel, de nombreux effets aérodynamiques et gyroscopiques [Bristeau et al., 2009] modifient le modèle simple donné par l'équation (E1.2). Cependant, pour la commande des quadricoptères, ces effets sont généralement négligés ou considérés comme des perturbations pour la loi de commande

[Bouabdallah et al., 2004a; Mahony et al., 2012]. Ces effets ne seront donc pas modélisés non plus dans le cadre de cette thèse.

La dynamique des moteurs est également généralement négligée car considérée comme très rapide comparée à la dynamique de la commande du quadricoptère complet [Castillo et al., 2005]. Les vitesses de rotation au carré des rotors sont donc considérées comme les entrées du modèle dynamique. Elle sont regroupées dans un vecteur $\mathbf{\Omega} = [\omega_1^2 \ \omega_2^2 \ \omega_3^2 \ \omega_4^2]^T$.

Le torseur d'action mécanique $\mathbf{w}_d^T = [\mathbf{f}_d^T \ \mathbf{m}_d^T]$ agissant sur le corps rigide est la résultante des torseurs de chacun des rotors. Ce torseur exprimé au point D dans le repère $\mathcal{F}_d$ est alors donné par :

$$\begin{aligned} \mathbf{f}_d &= \begin{bmatrix} 0 \\ 0 \\ k_t \sum_{j=1}^{4} \omega_j^2 \end{bmatrix} \\ \mathbf{m}_d &= \begin{bmatrix} k_t \frac{d}{2}(-\omega_1^2 + \omega_2^2 - \omega_3^2 + \omega_4^2) \\ k_t \frac{d}{2}(\omega_1^2 - \omega_2^2 - \omega_3^2 + \omega_4^2) \\ k_d(-\omega_1^2 - \omega_2^2 + \omega_3^2 + \omega_4^2) \end{bmatrix} \end{aligned} \tag{E1.3}$$

L'expression sous forme matricielle de l'équation (E1.3) s'écrit :

$$^d\mathbf{w}_d = \mathbf{Q}\mathbf{\Omega} \tag{E1.4}$$

avec $\mathbf{Q}$ une matrice de répartition des efforts donnée par :

$$\mathbf{Q} = k_t \begin{bmatrix} 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 1 & 1 & 1 & 1 \\ -\frac{d}{2} & \frac{d}{2} & -\frac{d}{2} & \frac{d}{2} \\ \frac{d}{2} & -\frac{d}{2} & -\frac{d}{2} & \frac{d}{2} \\ -\frac{k_d}{k_t} & -\frac{k_d}{k_t} & \frac{k_d}{k_t} & \frac{k_d}{k_t} \end{bmatrix}$$

On constate en analysant la matrice $\mathbf{Q}$ que le degré d'actionnement d'un quadricoptère est limité. Le quadricoptère ne peut pas générer de poussée selon les axes locaux $\mathbf{x}_d$ et $\mathbf{y}_d$. Pour d, k_t et k_d non nuls, les quatre lignes non nulles de la matrice $\mathbf{Q}$ correspondent à la poussée selon $\mathbf{z}_d$, notée $f_{z,d}$, et aux trois moments générés par les hélices, i.e. $\mathbf{m}_d$. Elles sont linéairement indépendantes. Il est donc possible de calculer le carré de la vitesse de rotation des hélices d'un quadricoptère à partir de la poussée $f_{z,d}$ et du moment $\mathbf{m}_d$ appliqués au corps rigide. Cette relation s'écrit :

$$\mathbf{\Omega} = \mathbf{Q}_r^{-1} \begin{bmatrix} f_{z,d} \\ \mathbf{m}_d \end{bmatrix} \tag{E1.5}$$

avec $\mathbf{Q}_r$ la sous-matrice composée des quatre dernières lignes de la matrice $\mathbf{Q}$. L'équation (E1.5) établit une relation bijective entre les efforts appliqués au quadricoptère et la norme

de la vitesse de rotation de chaque rotor ω_j avec $\omega_j > 0$. Ainsi, il est possible de considérer les entrées du modèle dynamique du drone comme étant la force de poussée f_z et le moment $\mathbf{m}_d$. Les vitesses de rotation des hélices associées seront obtenues ensuite par la relation (E1.5).

Dynamique du corps rigide

Généralement, il est considéré que le centre de gravité du quadricoptère est confondu avec le point D. Le modèle dynamique du quadricoptère est alors donné par les équations décrivant la dynamique d'un corps rigide soumis au torseur $\mathbf{w}_d$ appliqué en son centre de gravité et à la force de gravité $\mathbf{g}$. Les équations de Newton-Euler pour ce cas particulier s'écrivent :

$$m_d {}^o\ddot{\boldsymbol{\xi}} = m_d {}^o\mathbf{g} + \mathbf{R}_d {}^d\mathbf{f}_d \tag{E1.6}$$

$$\boldsymbol{\Sigma}_d {}^d\dot{\boldsymbol{\omega}} + {}^d\boldsymbol{\omega} \times \boldsymbol{\Sigma}_d {}^d\boldsymbol{\omega} = {}^d\mathbf{m}_d \tag{E1.7}$$

avec :

- $\boldsymbol{\omega}$ le vecteur des vitesses angulaires du quadricoptère,
- m_d la masse du quadricoptère,
- $\boldsymbol{\Sigma}_d$ la matrice d'inertie du quadricoptère exprimée dans le repère $\mathcal{F}_d$.

Rappelons que dans l'équation (E1.6) la force ${}^d\mathbf{f}_d$ donnée par l'équation (E1.3) possède deux composantes nulles : ${}^d\mathbf{f}_d = [0\ 0\ f_{z,d}]^T$

Une dernière hypothèse généralement effectuée sur les quadricoptères est que la matrice d'inertie $\boldsymbol{\Sigma}_d$ est diagonale du fait des symétries de l'appareil. De plus, avec l'hypothèse de vol quasi-stationnaire, il est possible de linéariser le modèle dynamique (E1.7) des rotations autour du point d'équilibre [Bouabdallah et al., 2004a]. On obtient alors, en négligeant les effets gyroscopiques :

$$\begin{aligned} XX_d\ddot{\varphi} &= m_x \\ YY_d\ddot{\theta} &= m_y \\ ZZ_d\ddot{\psi} &= m_z \end{aligned} \tag{E1.8}$$

avec :

- XX_d, YY_d et ZZ_d les moments d'inertie selon les axes respectifs $\mathbf{x}_d$, $\mathbf{y}_d$ et $\mathbf{z}_d$,
- φ, θ et ψ les angles d'orientation (angles de Bryant) du repère $\mathcal{F}_d$ par rapport au repère $\mathcal{F}$.
- m_x, m_y, m_z les trois composantes du moment ${}^d\mathbf{m}_d$.

1.1.3 Commande d'un quadricoptère

Un quadricoptère est un système disposant de six degrés de liberté commandés par seulement quatre entrées composées de la force de poussée $f_{z,d}$ et des trois moments $m_{x,d}$, $m_{y,d}$ et

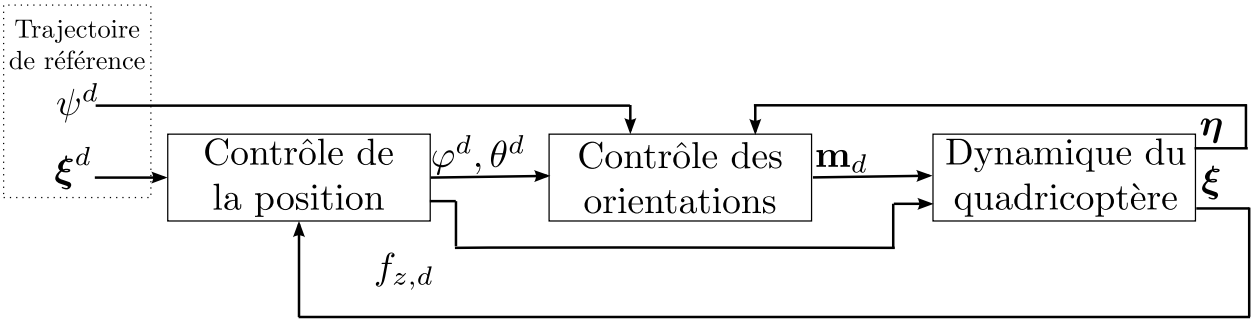


FIGURE 1.3 – Schéma général d'une commande classique pour un quadricoptère.

$m_{z,d}$. Il s'agit donc d'un système sous-actionné. Dans le cas d'une représentation des orientations de type "roulis (φ), tangage (θ), lacet (ψ)", il a été montré dans [Mellinger, 2011] que la dynamique d'un quadricoptère est un système différentiellement plat pour la sortie $\boldsymbol{\mu} = [x\ y\ z\ \psi]$. Cela signifie que le vecteur des variables d'état et le vecteur des entrées peuvent être exprimés comme des fonctions du vecteur de sortie $\boldsymbol{\mu}$ et de ses dérivées. Du fait de cette propriété, toute trajectoire suffisamment lisse générée pour la sortie $\boldsymbol{\mu}$ pourra être suivie par le quadricoptère [van Nieuwstadt et Murray, 1996].

Différentes stratégies ont été proposées pour la commande d'un quadricoptère. La plupart des commandes établies s'organisent selon la structure en cascade suivante (voir Fig. 1.3) :

- La trajectoire de référence est établie pour la sortie $\boldsymbol{\mu}$ donnant les coordonnées cartésiennes $\boldsymbol{\xi}^d$ et l'angle de lacet ψ^d désirés.
- Une boucle externe assure la commande de la position du centre de gravité du drone à partir de la trajectoire de référence. Cette boucle externe calcule la force de poussée $f_{z,d}$ et les angles de référence φ^d et θ^d pour le suivi de la trajectoire.
- Une boucle interne calcule les moments nécessaires $m_{x,d}$, $m_{y,d}$ et $m_{z,d}$ pour le suivi des angles de référence.

Pour que l'ensemble fonctionne, il est nécessaire de respecter le principe de séparation des échelles de temps entre les deux boucles de régulation. Ainsi, la boucle interne de commande des orientations devra converger rapidement ; elle sera donc appelée *boucle rapide*. La boucle externe de commande de la position devra converger plus lentement ; elle sera appelée *boucle lente*. La commande peut être d'approche linéaire autour du point d'équilibre en vol stationnaire ou non-linéaire. Différentes stratégies peuvent être abordées dans une même loi de commande pour la stabilisation de chacune des boucles.

Lois de commande linéaires

Les approches linéaires pour la commande d'un quadricoptère sont basées sur l'approximation de la dynamique du système autour du point de fonctionnement. Parmi ces méthodes, on peut citer la commande PID (Proportionelle Intégrale Dérivée) et la commande LQ (Linéaire Quadratique). L'approche PID est aujourd'hui la plus utilisée dans les autopilotes. Les

auteurs de [Bouabdallah et al., 2004b] comparent expérimentalement la commande LQ et la commande PID pour la commande des orientations d'un quadricoptère. Une approche LQ pour le suivi de trajectoires en coordonnées cartésiennes est proposée dans [Cowling et al., 2007]. Les auteurs de [Salih et al., 2010] et [Li et Li, 2011] développent une commande où les deux boucles sont basées sur une commande de type PID. L'approche linéaire n'est valable que pour un état proche du vol stationnaire. Elle ne sera donc pas valide pour de fortes inclinaisons du quadricoptère, nécessaires lors des vols très dynamiques.

Stabilité de Lyapunov

La recherche de meilleures performances et de configurations éloignées du vol stationnaire a mené aux développements de lois de commande non-linéaires. La preuve de stabilité de ces commandes repose généralement sur la théorie de stabilité de Lyapunov [Khalil, 2002]. Nous rappelons cette théorie car elle sera utilisée pour la démonstration de la stabilité de la commande établie pour cette thèse.

On considère un système $\mathcal{S}$ et son vecteur des variables d'état $\boldsymbol{\beta}$. Supposons $\boldsymbol{\beta} = \mathbf{0}$ un point d'équilibre de ce système. Une condition suffisante pour que l'état $\boldsymbol{\beta} = \mathbf{0}$ soit asymptotiquement stable est qu'il existe une fonction réelle $V(\boldsymbol{\beta})$ continuellement différentiable telle que :

$$V(\mathbf{0}) = 0 \quad \text{et} \quad V(\boldsymbol{\beta}) \text{ pour } \boldsymbol{\beta} \neq \mathbf{0}$$
$$\dot{V}(\boldsymbol{\beta}) < 0 \text{ pour } \boldsymbol{\beta} \neq \mathbf{0}$$

Pour un système en boucle fermée, si l'on veut prouver la convergence du vecteur d'erreur $\mathbf{e}$ vers $\mathbf{0}$, on cherchera en pratique une fonction $V(\mathbf{e}, \dot{\mathbf{e}})$ respectant ces critères. Le critère de Lyapunov est simple mais la fonction de Lyapunov peut être difficile à obtenir dans certains cas.

Lois de commande non linéaires

La liste suivante présente quelques lois de commande non linéaires utilisées pour la commande des quadricoptères.

- Commande en mode glissant (SMC : Sliding mode control) [Xu et Özguner, 2006; Sharifi et al., 2010; Besnard et al., 2012]. Cette commande présente l'avantage de garantir la stabilité en présence de perturbations bornées. Le temps de réponse est très rapide mais un phénomène de broutement (chattering) est généré. Des adaptations de la commande existent pour limiter ce phénomène aux alentours du point d'équilibre.
- Linéarisation par retour d'état [Carino et al., 2015; Voos, 2009; Mokhtari et Benallegue, 2004; Mistler et al., 2001]. La méthode consiste à transformer le système non linéaire en un équivalent linéaire par inversion du modèle dynamique du quadricoptère. L'inconvénient de cette méthode est la sensibilité aux erreurs de modèle. Diverses tech-

niques, dont l'adaptation des paramètres dynamiques en ligne, peuvent être employées pour améliorer la stabilité de cette commande face à ces erreurs. Les auteurs de [Carino et al., 2015] présentent une linéarisation doublée de l'usage des quaternions pour la représentation des orientations. L'avantage de la représentation en quaternions est l'absence de singularités de représentation contrairement aux angles d'Euler. Cette représentation est cependant plus complexe mathématiquement.

- Commande par backstepping [Madani et Benallegue, 2006; Lee et al., 2007]. Cette commande s'établit récursivement à l'aide de fonctions de Lyapunov successives. Elle contient également un modèle dynamique théorique et peut donc être sensible aux erreurs de modèle au même titre que la linéarisation par retour d'état.
- Algorithmes adaptatifs [Palunko et Fierro, 2011; Diao et al., 2011; Huang et al., 2010; Dierks et Jagannathan, 2010]. En complément des méthodes non linéaires décrites ci-avant, des algorithmes adaptatifs peuvent être mis en œuvre pour compenser les perturbations non modélisées ou les incertitudes du modèle dynamique.

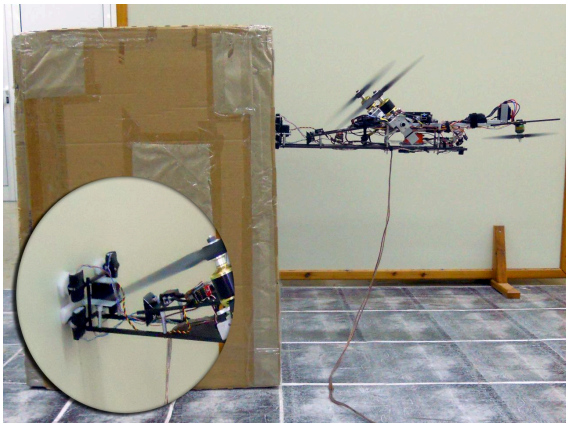
Nous allons ensuite présenter des exemples de robots manipulateurs aériens existants. La plupart d'entre eux sont basés sur des plate-formes de type quadricoptère.

1.2 Les robots manipulateurs aériens

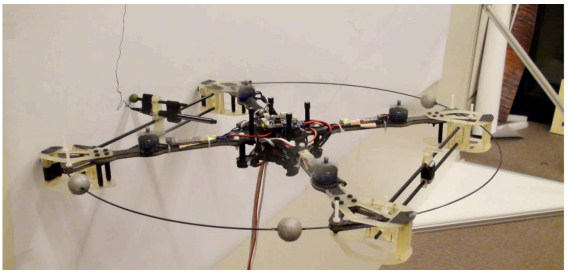
L'essentiel des applications existantes pour les drones se basent sur une interaction à distance avec l'environnement (par exemple à l'aide d'une caméra) en dehors de la propulsion de l'appareil. En effet, le vol du drone peut être facilement déstabilisé par les retours d'efforts générés au niveau des points de contacts. Au cours de la dernière décennie, plusieurs projets de manipulateurs aériens ont été développés pour faire interagir le robot volant avec son environnement. Quelques exemples sont présentés dans la suite de cette partie.

Outils rigides

L'interaction la plus simple consiste à faire interagir directement le drone avec l'environnement sans actionnement supplémentaire. Les auteurs de [Bellens et al., 2012; Nguyen et Lee, 2013; Mersha et al., 2014] proposent une loi de commande hybride position/effort pour l'interaction d'un quadricoptère avec son environnement. Des applications comme la génération d'une poussée à l'aide d'un tricoptère aux rotors inclinables [Papachristos et al., 2014] ou l'écriture aérienne à l'aide d'un quadricoptère [Darivianakis et al., 2014] ont été développées (voir Fig. 1.4). Si les expérimentations sur quadricoptères ont montré la capacité des lois de commande à gérer l'interaction avec l'environnement, ce type d'interaction se limite à un seul point de contact avec une précision de positionnement faible.



(a) Poussée à l'aide d'un tricoptère à rotors inclinables [Papachristos et al., 2014].



(b) Ecriture aérienne à l'aide d'un quadricoptère [Darivianakis et al., 2014].

FIGURE 1.4 – Exemples de drones munis d'un outil rigide.

(a) Hélicoptère associé à une pince [Pounds et al., 2011].

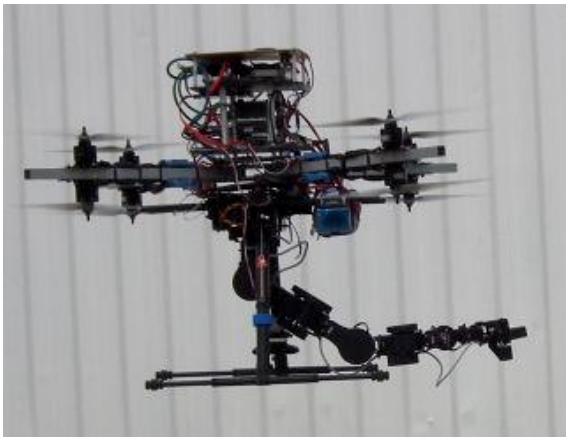
(b) Quadricoptère associé à une pince [Lindsey et al., 2011].

FIGURE 1.5 – Exemples de drones associés à un manipulateur à un degré de liberté.

Manipulateurs à 1 degré de liberté

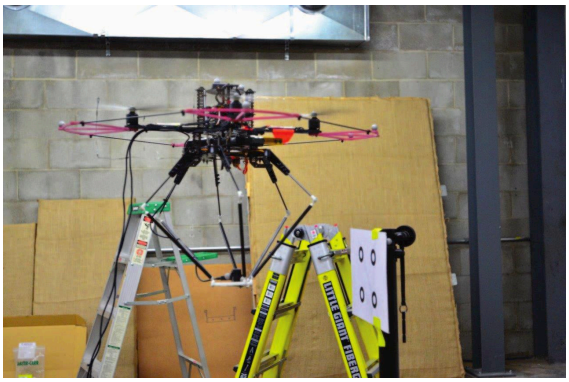
L'utilisation d'une pince actionnée à 1 degré de liberté (Fig. 1.5) est la solution la plus simple pour attraper un objet à l'aide d'un drone. Des pinces plus ou moins complexes ont été associées à des hélicoptères [Pounds et al., 2011; Backus et al., 2014] ou des quadricoptères [Lindsey et al., 2011; Ghadiok et al., 2011]. Les auteurs de [Lindsey et al., 2011] utilisent ce type de manipulateur aérien pour assembler des structures cubiques. Du fait de la faible charge utile des drones, cette solution convient pour attraper des objets de faible masse. À noter également que l'objet doit être suffisamment massif pour ne pas être soufflé par le drone lors de la prise.

(a) Hélicoptère associé à un bras Kuka à 7 degrés de liberté [Huber et al., 2013].



(b) Quadricoptère associé à un manipulateur à 7 degrés de liberté [Heredia et al., 2014].

(c) Quadricoptère associé à deux bras anthropomorphiques [Suarez et al., 2017].



(d) Quadricoptère associé à un robot parallèle [Danko et al., 2015].

FIGURE 1.6 – Exemples de drones associés à un manipulateur à plusieurs degrés de liberté.

Manipulateurs à plusieurs degrés de liberté

Par conception, les micro-drones à voilure tournante sont en général sous-actionnés. L'ajout d'une simple pince ne permet pas de compenser cette limitation. Afin d'augmenter les possibilités de manipulations, des robots aériens composés d'un drone et d'un manipulateur à plusieurs degrés de liberté ont été proposés (voir Fig. 1.6). Différentes architectures pour le manipulateur ont été envisagées et les exemples présentés ci-après ont été développés jusqu'au stade expérimental :

- **Manipulateurs sériels** : Les auteurs de [Kim et al., 2013] présentent un quadricoptère associé à un bras à 2 degrés de liberté munit d'une pince pour des applications de déplacement d'objets. Pour les mêmes applications à une autre échelle, les auteurs de [Huber et al., 2013] associent un hélicoptère à un bras industriel à 7 degrés de liberté de marque Kuka : l'étude indique qu'il a été nécessaire de restreindre les mouvements du bras pour ne pas déstabiliser le vol de l'hélicoptère. Plusieurs études se concentrent sur

la stabilisation du vol lors des mouvements du bras manipulateur. Ces études associent un quadricoptère à un manipulateur série à 7 [Heredia et al., 2014], 6 [Ruggiero et al., 2015] ou 2 [Tognon et al., 2017] degrés de liberté. Les auteurs de [Cataldi et al., 2016] ont développés une loi de commande pour un quadricoptère associé à un bras à 6 degrés de liberté optimisant la trajectoire du quadricoptère afin de diminuer les forces d'interaction avec l'environnement au niveau de l'effecteur. L'étude [Ryll et al., 2018] propose un robot manipulateur aérien muni de deux bras anthropomorphiques à 4 degrés de liberté plus une pince. Dans le cadre de cette étude, la commande de vol du quadricoptère est assuré par un opérateur humain ; seule l'opération de prise d'un objet par les bras est automatisée.

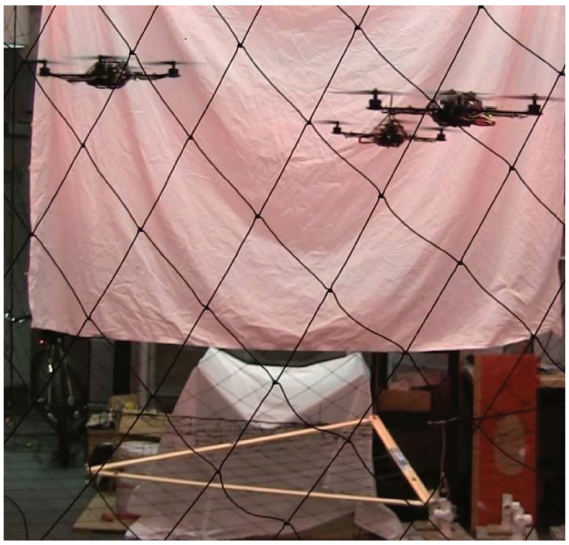
- **Manipulateurs parallèles** : L'étude [Fumagalli et al., 2014] associe un quadricoptère à un manipulateur parallèle de type Delta à trois degrés de liberté. L'objectif de la manipulation est d'obtenir un point de contact précis entre le robot aérien et l'environnement et de gérer les forces d'interaction. Les auteurs de [Danko et al., 2015] associent un quadricoptère à un robot parallèle pour obtenir une bonne précision de la position de l'effecteur. Développée à des fins d'observation, l'étude ne prévoie pas d'interaction de l'effecteur avec l'environnement. L'étude [Kamel et al., 2016] présente un quadricoptère associé à un robot parallèle de type mécanisme à cinq barres. Cette étude ne présente pas non plus d'interaction avec l'environnement.

On constate que ces projets restent récents et qu'il n'existe pas de solution expérimentale permettant une manipulation fine en interaction permanente avec l'environnement. Les problèmes majeurs de ce type d'application sont décrits ci-après :

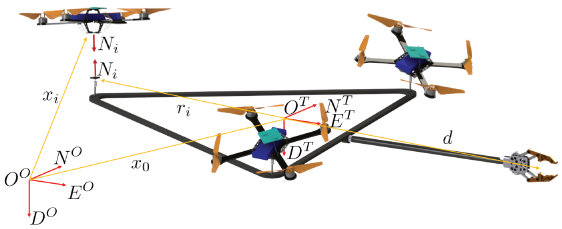
- Contrairement aux robots au sol, les drones n'ont pas de base stable. Ainsi, les efforts générés par la manipulation peuvent perturber la précision voire la stabilité du drone, d'autant que les drones utilisés sont en général des systèmes sous-actionnés. Une étude récente [Ryll et al., 2018] propose d'associer un manipulateur à deux degrés de liberté à un hexacoptère complètement actionné afin de lever cette difficulté.
- Les performances du robot peuvent être impactées par la proximité d'obstacles et, réciproquement, les éléments de l'environnement peuvent être impactés par le souffle du robot.
- La charge utile des robots est limitée et l'ajout d'un bras manipulateur peut diminuer amplement l'autonomie du drone porteur.

Coopération pour la manipulation

Afin de surmonter les limites en charge utile, plusieurs études se sont consacrées à la coopération entre plusieurs drones pour la manipulation et le transport de charges (voir Fig. 1.7). Une solution consiste à relier plusieurs quadricoptères à une charge à l'aide de câbles [Michael et al., 2011; Sreenath et Kumar, 2013; Erskine et al., 2018]. Toutefois, elle reste limitée en terme de manipulation. En effet, il est nécessaire qu'un opérateur attache les câbles



(a) Exemple de coopération pour la levée de charge par plusieurs quadricoptères à l'aide de câbles [Sreenath et Kumar, 2013].



(b) Outil aérien actionné par la coopération de plusieurs quadricoptères [Nguyen et al., 2015].

FIGURE 1.7 – Exemples de manipulation en coopération par plusieurs quadricoptères.

à la charge et le système ne peut pas exercer d'efforts de poussée au travers des câbles. Il s'agit néanmoins d'une solution efficace pour le transport de charges.

Les drones peuvent également être directement reliés à l'objet manipulé par des connecteurs magnétiques [Kim et al., 2014] ou des articulations sphériques [Nguyen et al., 2015]. Ce dernier exemple peut être considéré comme un cas particulier de notre étude. Il s'agit d'un robot parallèle volant pour lequel la structure cinématique des jambes est réduite à une seule articulation sphérique.

Nous allons maintenant présenter les robots parallèles et introduire leur modèle dynamique ainsi que les singularités de ce modèle.

1.3 Robots parallèles, modèles cinématiques, dynamiques et singularités

Le robot parallèle volant est similaire à un robot parallèle classique, à la différence que les actionneurs utilisés pour un robot parallèle sont remplacés par des drones dans le cas du robot parallèle volant. Les modèles géométriques, cinématiques et dynamiques des robots parallèles ont été beaucoup étudiés. Il sera possible de transférer une partie de la connaissance de ces modèles pour l'appliquer aux robots parallèles volants. Cette partie présente succinctement les robots parallèles. Elle permet d'introduire une méthode de calcul de leurs modèles. Elle présente aussi certaines singularités de configuration qui entraînent une dégénérescence du modèle dynamique. Des singularités similaires, présentées au chapitre 5, seront étudiées pour

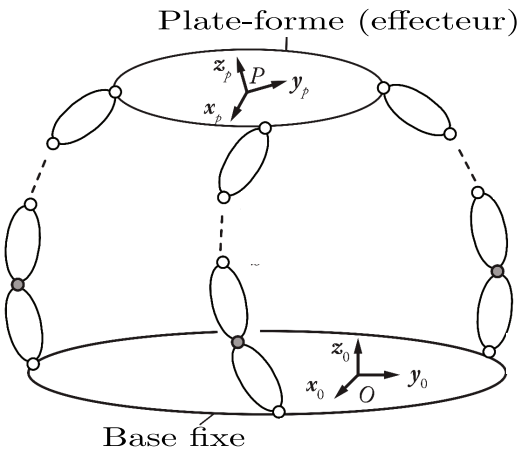


FIGURE 1.8 – Schéma général d'un robot parallèle [Briot, 2014]. Les articulations grisées sont les articulations actionnées.

les robots parallèles volants. La méthode développée pour la reconfiguration en vol s'inspire des travaux existants pour la traversée des singularités des robots parallèles classiques que nous aborderons également ici.

1.3.1 Généralités sur les robots parallèles

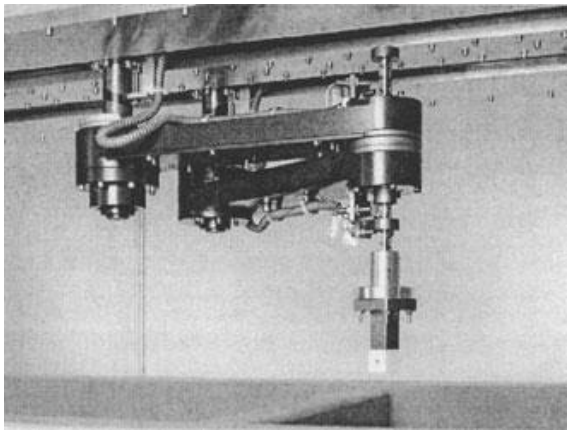
Les robots parallèles sont définis dans [Leinonen, 1991] comme des robots contrôlant les mouvements de leur effecteur à l'aide d'au moins deux chaînes cinématiques allant de l'effecteur à la base fixe. Ainsi, un robot parallèle est composé (Fig. 1.8) :

- d'une base fixe,
- d'une plate-forme mobile sur laquelle est habituellement monté l'effecteur,
- de chaînes cinématiques liant la base à la plate-forme, appelées *jambes* du robot.

Du fait de la répartition de la charge totale sur plusieurs chaînes cinématiques, les robots parallèles présentent généralement l'avantage d'une importante charge utile et de la capacité de générer de fortes dynamiques. Leur utilisation s'est répandue dans l'industrie dans le cadre d'applications aussi variées que le "pick and place", la simulation de vol, les systèmes de mesure, le micro-positionnement, les interfaces haptiques, etc. Les principaux défauts des robots parallèles sont leurs structures cinématiques qui peuvent être complexes et, en général, un espace de travail limité.

De nombreuses topologies des jambes ont été étudiées [Kong et Gosselin, 2007; Gogu, 2008], menant à un large choix d'architectures pour ces robots. En général, l'architecture cinématique est identique sur toutes les jambes. Nous présenterons ci-après quelques exemples de robots parallèles. La liste n'est pas exhaustive car les architectures existantes sont trop

(a) Mécanisme à cinq barres ($\underline{R}RRR\underline{R}$) par MECADEMIC



(b) Mécanisme biglide ($\underline{P}RRR\underline{P}$) [Hesselbach et al., 2002]

FIGURE 1.9 – Exemples de robots parallèles plans à deux degrés de liberté.

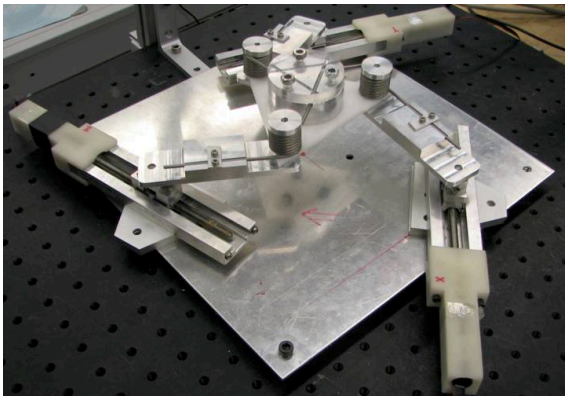
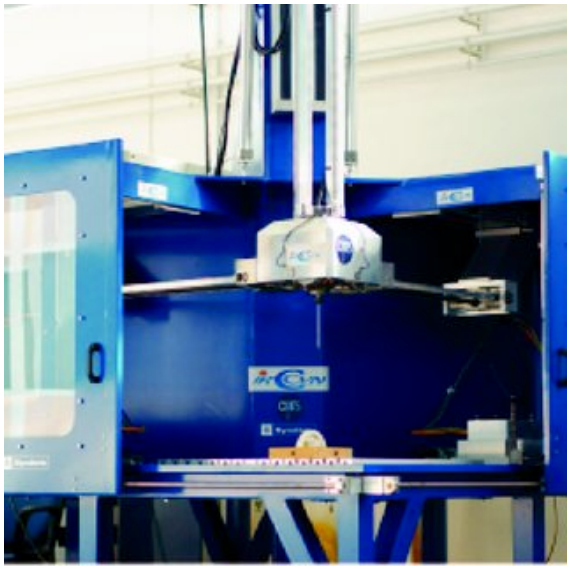


FIGURE 1.10 – Exemple de robot parallèle plan à trois degrés de liberté $3-\underline{P}RR$ [Wei et Simaan, 2010].

(a) Robot Delta ($3-\underline{R}RPaR$) par OMRON



(b) L'Orthoglide ($3-\underline{P}RPaR$) [Chablat et Wenger, 2003]

(b) The prototype.

(c) Le Tripteron ($3-\underline{P}RRR$) [Gosselin, 2009]

FIGURE 1.11 – Exemples de robots parallèles permettant trois translations de l'effecteur dans l'espace.

(a) Hexapode ($6-\underline{U}PS$) utilisé pour la simulation de vol (CAE).

(b) Robot Hexa ($6-\underline{R}US$) (Université de Tohoku)

FIGURE 1.12 – Exemples de robots parallèles à six degrés de liberté.

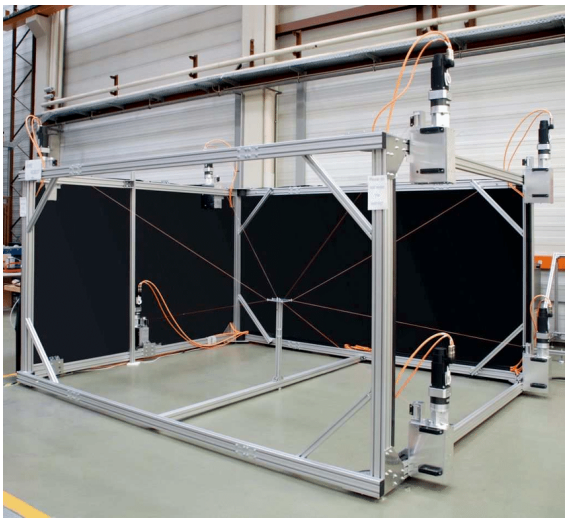


FIGURE 1.13 – Exemple de robot parallèle à câbles (Fraunhofer IPA).

nombreuses pour être toutes citées ici [1] :

- Robots parallèles plans à deux degrés de liberté tels que le mécanisme à cinq barres ($\underline{R}RRR\underline{R}$) ou le mécanisme bilglide ($\underline{P}RRR\underline{P}$) [Hesselbach et al., 2002]. Ces robots peuvent positionner un point dans un plan (Fig. 1.9).
- Robots parallèles plans à trois degrés de liberté permettant de commander la position et l'orientation d'une plate-forme dans un plan (Exemple : $3-\underline{P}RR$ [Wei et Simaan, 2010]) (Fig. 1.10).
- Robots parallèles permettant trois translations de l'effecteur dans l'espace. On peut citer le robot Delta ($3-\underline{R}RPaR$) [Clavel, 1980], l'Orthoglide ($3-\underline{P}RPaR$) [Chablat et Wenger, 2003] et le Tripteron ($3-\underline{P}RRR$) [Gosselin, 2009] (Fig. 1.11).
- Robots parallèles à six degrés de liberté comme l'Hexapode ($6-U\underline{P}S$, appelé également plate-forme Gough-Stewart) et l'Hexa ($6-\underline{R}US$) [Pierrot et al., 1991] (Fig. 1.12).
- Robots parallèles à câbles. Les robots parallèles à câbles sont un type de robots parallèles où l'effecteur est relié à la base fixe à l'aide de câbles dont la longueur varie sous l'effet de l'actionnement (voir Fig. 1.13).

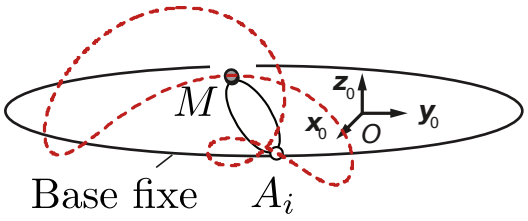
1.3.2 Modèles géométriques et cinématiques des robots parallèles

Soit un robot parallèle générique muni de n_a articulations actives, n_d articulations passives et n_x degrés de liberté indépendants pour la plate-forme. Nous considérerons que le robot possède autant de variables d'actionnement que de degrés de liberté ($n_a = n_x$), ce qui est le cas pour la plupart des robots parallèles classiques. Nous noterons $\mathbf{q}_a$ le vecteur des coordonnées des articulations actives, $\mathbf{q}_d$ le vecteur des coordonnées des articulations passives et $\mathbf{x}$ le vecteur des coordonnées de la plate-forme. Cette partie présente succinctement les modèles géométriques et cinématiques des robots parallèles, qui pourront être réutilisées pour le calcul de modèles géométriques ou cinématiques de robots parallèles volants, comme ceux présentés au chapitre 3.

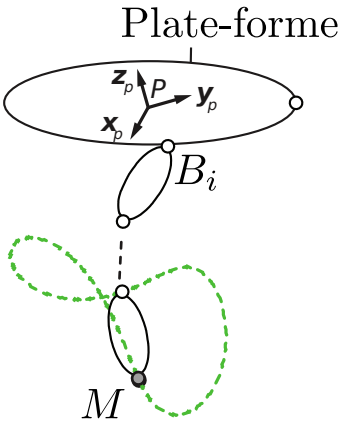
Modèle géométrique inverse

Le modèle géométrique inverse consiste à calculer les coordonnées des articulations $\mathbf{q}_a$ et $\mathbf{q}_d$ nécessaires pour obtenir une configuration $\mathbf{x}$ de la plate-forme donnée. J.P. Merlet présente deux approches pour calculer ce modèle [Merlet, 2006] : une méthode analytique et une méthode géométrique. Nous décrirons ici la méthode géométrique. On considère séparément chaque jambe i reliant la plate-forme à la base. L'extrémité de la jambe côté base, représentée par le point A_i, possède une position géométrique fixe. L'autre extrémité côté plate-forme,

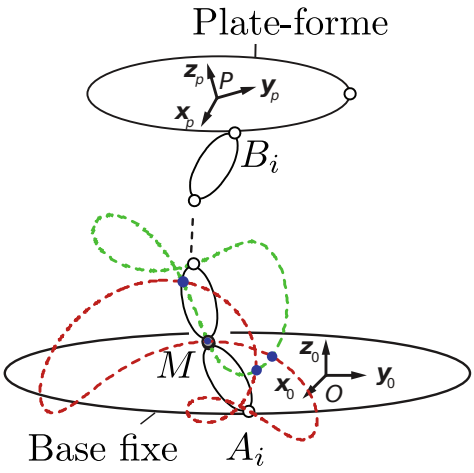
1. Pour cette liste, nous indiquerons la structure cinématique des jambes sous la forme d'une suite de lettres en partant de la base vers l'effecteur. Les articulations sont représentées par les lettres P (articulation prismatique), R (pivot), S (articulation sphérique), U (cardan) et Pa (type parallélogramme plan [Caro et al., 2010]). Les articulations actionnées sont soulignées. Le chiffre positionné avant la suite de lettre indique le nombre de jambes.



(a) Variété des positions possibles du point M pour le mécanisme $\mathcal{M}_1$.

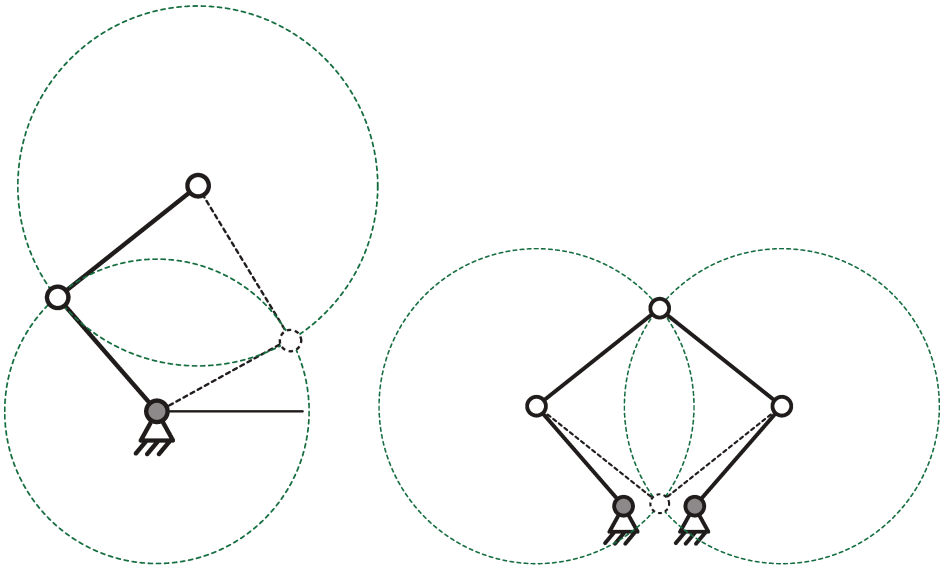


(b) Variété des positions possibles du point M pour le mécanisme $\mathcal{M}_2$.



(c) Les intersections entre les deux variétés donnent les solutions du modèle géométrique inverse.

FIGURE 1.14 – Méthode de résolution du modèle géométrique inverse d'un robot parallèle.



(a) (b)

FIGURE 1.15 – Modes de fonctionnement d'une jambe (a) et modes d'assemblage (b) pour le robot parallèle à cinq barres [Briot, 2014]. Les articulations actives du robot sont représentées en gris.

représentée par le point B_i, est déterminée par la position et l'orientation de la plate-forme, supposés connu pour le modèle géométrique inverse. La méthode consiste à couper la jambe en un point M donné. On obtient ainsi deux mécanismes :

- $\mathcal{M}_1$ entre le point A_i et le point M,
- $\mathcal{M}_2$ entre le point B_i et le point M.

Les mouvements des mécanismes vont déterminer chacun une variété représentant l'ensemble des positions possibles du point M. Les solutions du modèle géométrique inverse sont obtenues à l'intersection de la variété obtenue par les mouvements du mécanisme $\mathcal{M}_1$ et de celle obtenue par les mouvements du mécanisme $\mathcal{M}_2$ (voir Fig. 1.14). Plusieurs solutions peuvent être obtenues. Elles correspondent à différentes configurations possibles de la jambe appelées *modes de fonctionnement*. L'architecture des jambes est en général assez simple sur les robots parallèles existants et le modèle est calculé aisément à l'aide de cette méthode. La figure 1.15a illustre deux modes de fonctionnement pour une jambe d'un mécanisme à cinq barres.

Modèle géométrique direct

Le modèle géométrique direct du robot consiste à exprimer les coordonnées de la plate-forme $\mathbf{x}$ en fonction des coordonnées des articulations actives $\mathbf{q}_a$. En général, la solution à ce problème n'est pas unique. Pour une configuration des articulations actives donnée, il est possible d'assembler le robot de plusieurs façons. Ces configurations sont appelées les *modes d'assemblages*. La figure 1.15b illustre deux modes d'assemblage pour le mécanisme

à cinq barres. Dans le cas un robot parallèle à six degrés de liberté, le modèle géométrique direct est généralement un problème complexe, qui ne peut pas être exprimé de façon analytique. Plusieurs exemples de méthodes pour trouver ces solutions, appliquées à différents mécanismes sont présentés dans [Merlet, 2006]. Ces méthodes peuvent nécessiter un temps de calcul important et ne permettent pas de déterminer le mode d'assemblage à partir de la configuration des articulations actives uniquement. Il existe des méthodes numériques [Merlet, 2006], permettant un calcul plus rapide de la configuration de la plate-forme à partir d'une position proche connue. Il n'a pas été nécessaire, dans le cadre de cette thèse de faire appel à ces algorithmes. Le modèle géométrique du robot étudié dans le cadre expérimental a été exprimé analytiquement au chapitre 3.

Modèles cinématiques

Les méthodes pour obtenir les modèles cinématiques ne seront pas détaillées dans le cadre de cet état de l'art. Nous renvoyons le lecteur vers [Merlet, 2006; Briot et Khalil, 2015]. Nous allons cependant présenter les équations de ce modèle car elles interviennent dans le calcul du modèle dynamique décrit ci-après. La première équation cinématique s'écrit :

$$\mathbf{A}\dot{\mathbf{x}} + \mathbf{B}\dot{\mathbf{q}}_a = \mathbf{0} \tag{E1.9}$$

avec $\mathbf{A}$ et $\mathbf{B}$ deux matrices carrées $(n_a \times n_a)$. Cette équation relie la vitesse des articulations actives à la vitesse des coordonnées de la plate-forme. Tant que les matrices $\mathbf{A}$ et $\mathbf{B}$ sont de rang plein, il est aisé d'obtenir les modèles cinématiques direct et inverse du robot à partir de cette équation. Une seconde équation cinématique relie la vitesse des articulations actives, des articulations passives et des coordonnées de la plate-forme $\dot{\mathbf{x}}$. Cette équation s'écrit :

$$\mathbf{J}_x\dot{\mathbf{x}} + \mathbf{J}_a\dot{\mathbf{q}}_a + \mathbf{J}_d\dot{\mathbf{q}}_d = \mathbf{0} \tag{E1.10}$$

avec $\mathbf{J}_x$ et $\mathbf{J}_a$ deux matrices $(n_d \times n_a)$ et $\mathbf{J}_d$ $(n_d \times n_d)$. Tant que la matrice $\mathbf{J}_d$ est de rang plein, cette équation permet de calculer la vitesse des articulations passives, connaissant la vitesse des articulations actives et la vitesse des coordonnées de la plate-forme.

1.3.3 Modèle dynamique inverse d'un robot parallèle

De nombreuses méthodes existent pour le calcul du modèle dynamique inverse d'un robot parallèle [Merlet, 2006]. Nous présenterons ici les grandes étapes de la méthode développée en [Briot et Khalil, 2015]. Le calcul du modèle dynamique inverse s'effectue en deux étapes.

1. La première étape consiste à séparer virtuellement chaque jambe de la plate-forme mobile. On obtient alors deux systèmes : une structure arborescente composée de la base du robot et des jambes du robot, considérées complètement actionnées pour la structure virtuelle, et une plate-forme libre indépendante. Le modèle dynamique de chacun des deux systèmes est alors calculé.

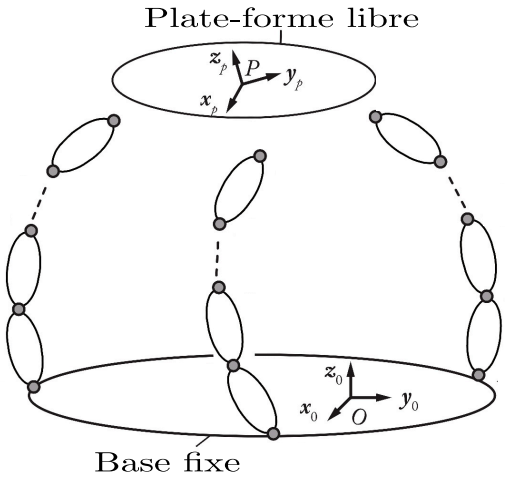


FIGURE 1.16 – Architecture virtuelle pour le calcul du modèle dynamique d'un robot parallèle [Briot, 2014]. Les articulations grisées sont les articulations actionnées.

Le Lagrangien de la structure arborescente virtuelle $L_t(\mathbf{q}_a, \dot{\mathbf{q}}_a, \mathbf{q}_d, \dot{\mathbf{q}}_d)$ est écrit en fonction des coordonnées des articulations actives et passives du robot. L'application des équations de Lagrange permet d'exprimer, pour cette structure arborescente, les efforts virtuels dans les articulations actives $\boldsymbol{\tau}_a$ et les efforts virtuels $\boldsymbol{\tau}_b$ dans les articulations passives :

$$\boldsymbol{\tau}_a = \frac{\mathrm{d}}{\mathrm{dt}} \left(\frac{\partial L_t}{\partial \dot{\mathbf{q}}_a} \right)^T - \left(\frac{\partial L_t}{\partial \mathbf{q}_a} \right)^T \tag{E1.11}$$

$$\boldsymbol{\tau}_d = \frac{\mathrm{d}}{\mathrm{dt}} \left(\frac{\partial L_t}{\partial \dot{\mathbf{q}}_d} \right)^T - \left(\frac{\partial L_t}{\partial \mathbf{q}_d} \right)^T \tag{E1.12}$$

Les vecteurs $\boldsymbol{\tau}_a$ et $\boldsymbol{\tau}_d$ sont respectivement de dimension n_a et n_d.

Le Lagrangien de la plate-forme libre $L_p(\mathbf{x}, \dot{\mathbf{x}})$ est écrit en fonction des coordonnées de cette dernière. Le torseur des efforts appliqués à la plate-forme est exprimé à l'aide des équations de Lagrange :

$$\mathbf{w}_x = \frac{\mathrm{d}}{\mathrm{dt}} \left(\frac{\partial L_p}{\partial \dot{\mathbf{x}}} \right)^T - \left(\frac{\partial L_p}{\partial \mathbf{x}} \right)^T \tag{E1.13}$$

2. La seconde étape consiste rassembler le robot en reliant chaque jambe à la plate-forme. L'expression du modèle dynamique est obtenu en tenant compte des équations cinématiques de fermeture de boucle (E1.9) et (E1.10) par la méthode des multiplicateurs

de Lagrange. Le système obtenu s'écrit :

$$\boldsymbol{\tau} = \boldsymbol{\tau}_a - \mathbf{B}^T\boldsymbol{\lambda}_1 - \mathbf{J}_a^T\boldsymbol{\lambda}_2 \tag{E1.14}$$
$$\mathbf{A}^T\boldsymbol{\lambda}_1 - \mathbf{J}_x^T\boldsymbol{\lambda}_2 = \mathbf{w}_x \tag{E1.15}$$
$$\mathbf{J}_d^T\boldsymbol{\lambda}_2 = \boldsymbol{\tau}_d \tag{E1.16}$$

avec :

- $\boldsymbol{\tau}$ le vecteur des efforts d'actionnement du robot, de dimension n_a,
- $\boldsymbol{\lambda}_1$ et $\boldsymbol{\lambda}_2$ les vecteurs multiplicateurs de Lagrange de dimensions respectives n_a et n_d.

En dehors des configurations singulières, la solution générale du modèle dynamique inverse s'obtient en calculant $\boldsymbol{\lambda}_2$ de (E1.16), $\boldsymbol{\lambda}_1$ de (E1.15). En les introduisant dans l'équation (E1.14), on obtient :

$$\boldsymbol{\tau} = \boldsymbol{\tau}_a - \mathbf{B}^T\mathbf{A}^{-T}\mathbf{w}_d - \mathbf{J}_a^T\mathbf{J}_d^{-T}\boldsymbol{\tau}_d \tag{E1.17}$$

avec $\mathbf{w}_d = \mathbf{w}_x + \mathbf{J}_x^T\mathbf{J}_d^{-T}\boldsymbol{\tau}_d$.

1.3.4 Singularités des robots parallèles et traversée des singularités

De nombreux types de singularités peuvent apparaître sur un robot parallèle. L'étude de l'intégralité de ces singularités sort du cadre de cette thèse et le lecteur est renvoyé vers deux publications de référence [Zlatanov et al., 1994] et [Conconi et Carricato, 2009]. Dans cette partie, nous décrirons deux types de singularités : les singularités de Type 2 et les singularités LPJTS (Leg Passive Joint Twist System). Elles sont associées à la dégénérescence du modèle dynamique inverse du robot parallèle et divisent généralement l'espace de travail du robot en plusieurs sous-espaces indépendants. Ainsi, elles limitent la zone de travail effective du robot en fonction de la configuration initiale (voir exemple Fig. 1.17). En particulier, les singularités de Type 2 séparent plusieurs sous-ensembles de l'espace de travail appelés *aspects*. Chaque aspect correspond aux différents mode d'assemblage du robot. La traversée de ces singularités, en particulier les singularités de Type 2, a fait l'objet de plusieurs études et des méthodes ont été proposées pour augmenter l'espace de travail opérationnel des robots concernés (voir ci-après). Les singularités présentes sur les robots parallèles volant sont similaires. Ainsi, les connaissances sur ces singularités et notamment sur l'extension de l'espace de travail d'un robot par la traversée de ces singularités, pourront être transposées et appliquées aux robots parallèles volants (voir chapitre 5).

Singularités de Type 2

Les singularités de Type 2 sont associées à la dégénérescence de la matrice $\mathbf{A}$ (E1.9). Lorsque le rang de cette matrice n'est pas plein, il existe un vecteur non nul $\dot{\mathbf{x}}_s$ tel que $\mathbf{A}\dot{\mathbf{x}}_s = \mathbf{0}$. L'interprétation physique de ce résultat est qu'un mouvement de la plate-forme

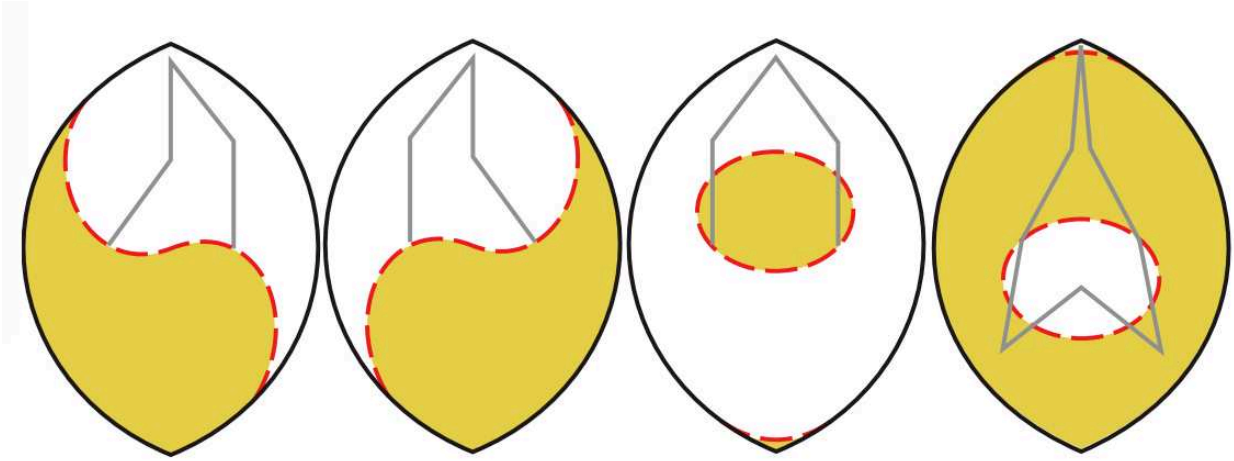


FIGURE 1.17 – Exemple de séparation de l'espace de travail d'un robot parallèle (cinq barres) par des singularités de Type 2 (pointillés rouges) pour différents mode de fonctionnement [Pagis, 2015].

est localement possible même si l'actionnement du robot est bloqué ; une direction de la plate-forme devient localement incontrôlée. Réciproquement [Briot et Khalil, 2015], certains efforts ne peuvent plus être transmis de l'actionnement à la plate-forme par les jambes.

Pour déterminer le lieu des singularités en pratique, le calcul analytique à l'aide du déterminant de la matrice $\mathbf{A}$ peut être très complexe. L'étude des singularités des robots parallèles peut être cependant effectuée à l'aide d'outils tels que la géométrie de Grassmann [Merlet, 2006] et l'algèbre de Grassmann-Cayley [Ben-Horin et Shoham, 2006].

A proximité de la singularité, le déterminant de la matrice $\mathbf{A}$ est proche de zéro. Si certaines précautions ne sont pas prises (voir ci-après), les efforts calculés par le modèle dynamique inverse (E1.17) vont tendre vers une norme infinie. De telles singularités peuvent ainsi mener à une détérioration physique du mécanisme ou à l'impossibilité de suivre la trajectoire désirée. La figure 1.18 montre un exemple de robot parallèle (mécanisme à cinq barres) en singularité de Type 2.

Singularité de type LPJTS

Les singularités de type LPJTS sont associées à la dégénérescence de la matrice $\mathbf{J}_d$ (E1.10). Lorsque le rang de cette matrice n'est pas plein, il existe un vecteur non nul $\dot{\mathbf{q}}_d$ tel que $\mathbf{J}_d\dot{\mathbf{q}}_d = \mathbf{0}$. L'interprétation physique de ce résultat est qu'un mouvement des articulations passives est localement possible même si l'actionnement et la plate-forme du robot sont fixes. La jambe gagne localement un mouvement incontrôlé. A proximité de la singularité, le déterminant de la matrice $\mathbf{J}_d$ est proche de zéro. Ainsi, si certaines précautions ne sont pas prises, les efforts calculés par le modèle dynamique inverse (E1.17) vont tendre vers une norme infinie avec les mêmes conséquences pour le mécanisme que les singularités de Type 2. La figure 1.19 montre un exemple de robot parallèle (Triptéron) en singularité LPJTS (voir

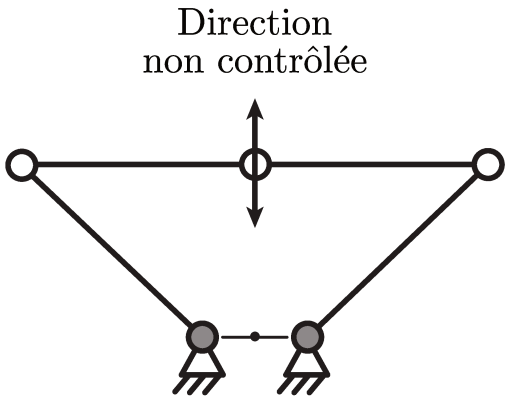


FIGURE 1.18 – Exemple de robot parallèle (cinq barres) en singularité de Type 2. Les articulations grisées sont les articulations actionnées.

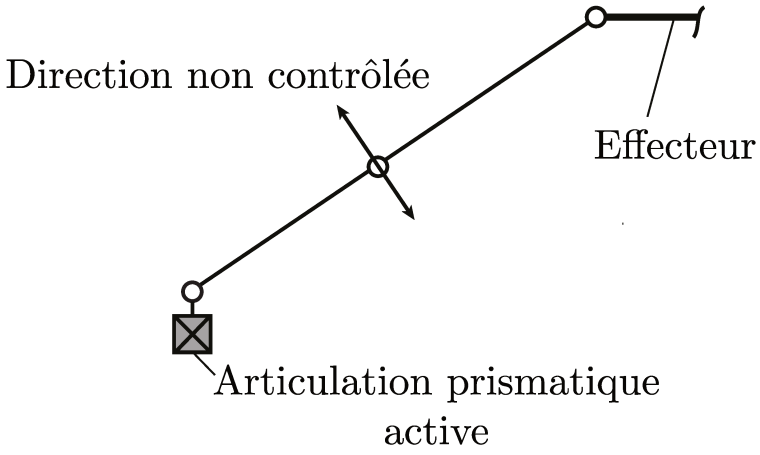


FIGURE 1.19 – Jambe du Triptéron en singularité LPJTS. L'articulation grisée est actionnée.

aussi Fig. 1.11 pour une photographie d'un robot Triptéron).

Traversée des singularités

Comme indiqué précédemment, l'espace de travail opérationnel réduit est l'une des limitations courantes des robots parallèles. Ainsi, plusieurs études se sont consacrées à l'extension de l'espace de travail opérationnel des robots parallèles vis à vis des singularités, telles que :

1. La conception de robots parallèles sans singularité [Liu et al., 2006; Gogu, 2004]. Cette solution mène souvent à la conception de robots avec un faible rapport espace de travail/encombrement du robot ou à la conception d'architectures complexes.
2. La conception de mécanismes redondants [Nahon et Angeles, 1989; Kurtz et Hayward, 1992; Rakotomanga et al., 2008]. Cette approche permet de diminuer, voire de supprimer, les singularités de Type 2 d'un robot parallèle.
3. Les mécanismes avec changement de mode d'actionnement [Arakelian et al., 2008]. Cette méthode ne nécessite pas d'actionneur supplémentaire mais le changement de

mode d'actionnement doit généralement s'effectuer lors que le robot est immobile, augmentant ainsi la durée pour effectuer une tache.

4. Le changement du mode de fonctionnement. Pour un robot parallèle, le changement de configuration d'une jambe peut changer le lieu des singularités de Type 2 [Campos et al., 2010] (Fig. 1.17). Cette reconfiguration s'effectue par la traversée d'autres singularités, situées en limite de l'espace de travail, qui ne provoquent pas la dégénérescence du modèle dynamique.
5. La planification de trajectoires modifiant le mode d'assemblage en contournant un point cuspidal (pour les singularités de Type 2) [Zein et al., 2008]. Cette solution ne concerne qu'un nombre limité de robots parallèles disposant de points cuspidaux.
6. La planification de trajectoires permettant de traverser directement la singularité sous contrainte du respect d'un critère dynamique de traversée [Ider, 2005; Briot et Arakelian, 2008; Briot et al., 2016]. Cette approche nécessite le développement de lois de commande spécifiques pour la traversée [Pagis et al., 2015; Six et al., 2017a; Hill et al., 2017].

Dans le cas des robots parallèles volants :

1. La redondance peut être exploitée en augmentant le nombre de drones pour un robot donné. Cette solution permet également de garder le contrôle du robot volant en cas de défaut sur l'un des drones. Le coût global du robot est cependant augmenté. Elle n'est pas développée dans cette thèse mais constitue une piste intéressante pour le développement de robots manipulateurs volants plus sécurisés.
2. L'actionnement est fourni uniquement par les drones. Ainsi, une solution nécessitant une modification du mode d'actionnement n'est pas envisagée.
3. Le changement de mode de fonctionnement ou le contournement d'un point cuspidal pourraient exister pour les robots parallèles volants mais nécessiterait, comme pour les robots parallèles classiques, la conception de trajectoires complexes spécifiques à ces applications. Cette solution n'a pas été explorée dans le cadre de cette thèse.
4. La traversée directe de la singularité est une solution adéquate, car elle peut s'appliquer directement sur, a priori, n'importe quel type d'architecture. Cette méthode sera étudiée pour son application sur les robots parallèles volants au chapitre 5. Nous allons donc développer plus en détail cette dernière solution.

Critère dynamique de traversée d'une singularité

Nous allons donner ci-après les conditions dynamiques nécessaires pour éviter des efforts en entrée infinis en singularités. Nous n'aborderons que le cas des singularités de Type 2. Des résultats similaires existent pour les singularités LPJTS (voir [Briot, 2014] et l'annexe C). Supposons donc que la matrice $\mathbf{A}$ n'est pas inversible (singularité de Type 2) et que la matrice $\mathbf{J}_d$ l'est (pas de singularité LPJTS). En résolvant $\boldsymbol{\lambda}_2$ dans (E1.16), les équations

(E1.14) et (E1.15) peuvent s'écrire sous la forme :

$$\boldsymbol{\tau} = \mathbf{w}_b - \mathbf{B}^T \boldsymbol{\lambda}_1 \tag{E1.18}$$

$$\mathbf{A}^T \boldsymbol{\lambda}_1 = \mathbf{w}_d \tag{E1.19}$$

avec $\mathbf{w}_b = \boldsymbol{\tau}_a - \mathbf{J}_a^T \mathbf{J}_d^{-T} \boldsymbol{\tau}_d$ et $\mathbf{w}_d = \mathbf{w}_x + \mathbf{J}_x^T \mathbf{J}_d^{-T} \boldsymbol{\tau}_d$.

Si la matrice $\mathbf{A}$ n'est pas de rang plein, comme indiqué précédemment, il existe un vecteur $\dot{\mathbf{x}}_s$ tel que [2] :

$$\mathbf{A}\dot{\mathbf{x}}_s = \mathbf{0} \tag{E1.20}$$

$\dot{\mathbf{x}}_s$ est un vecteur décrivant le mouvement incontrôlable lorsque le mécanisme est dans une configuration singulière de Type 2. En multipliant la partie gauche de (E1.19) par $\dot{\mathbf{x}}_s^T$, on obtient :

$$\dot{\mathbf{x}}_s^T \mathbf{A}^T \boldsymbol{\lambda}_1 = \mathbf{0} \tag{E1.21}$$

En conséquence, la partie droite de (E1.19) doit également respecter la condition

$$\dot{\mathbf{x}}_s^T \mathbf{w}_d = \mathbf{0} \tag{E1.22}$$

Le vecteur $\mathbf{w}_d$ représente la somme des efforts appliqués à la plate-forme par les jambes et les effets de gravité et d'inertie. La puissance développée par ces efforts le long de la direction non-contrôlée est nulle en singularité de Type 2.

La figure 1.20 illustre cette condition pour un mécanisme à cinq barres. Le mécanisme est composé de deux articulations actives situées aux points A_{11} et A_{21} et de trois articulations passives situées aux points A_{12}, A_{22} et E. E est également la position de l'effecteur. Dans [Pagis et al., 2015], une procédure d'identification dynamique réalisée sur un mécanisme à cinq barres aboutit à l'expression du vecteur $\mathbf{w}_d$ sous la forme $\mathbf{w}_d = m_E \ddot{\mathbf{x}}$ avec m_E une masse équivalente située au niveau de l'effecteur E. Le critère de traversée (E1.22) s'écrit alors $m_E \dot{\mathbf{x}}_s^T \ddot{\mathbf{x}} = \mathbf{0}$. L'accélération générée au niveau de l'effecteur doit alors être orthogonale à la direction non contrôlée. Soient $\mathbf{f}_{l,1}$ et $\mathbf{f}_{l,2}$ les forces transmises à l'effecteur au travers de chacune des jambes du mécanisme. Ces forces sont alignées respectivement selon les directions données par $\overrightarrow{A_{12}E}$ et $\overrightarrow{A_{22}E}$. Ainsi, il est impossible pour le mécanisme en configuration singulière de transmettre une force, donc une accélération, à l'effecteur dans une autre direction que celle des vecteurs $\overrightarrow{A_{12}E}$ et $\overrightarrow{A_{22}E}$, qui sont alors colinéaires. La direction orthogonale à ces deux vecteurs correspond à la direction non-contrôlée. De plus, le mécanisme ne peut pas résister à une force extérieure appliquée sur l'effecteur qui n'est pas alignée selon $\mathbf{f}_{l,2}$ et $\mathbf{f}_{l,2}$.

Plus qu'un critère nécessaire pour éviter les efforts en entrée infinis, la condition (E1.22) garantie l'existence d'au moins une solution exacte en $\boldsymbol{\lambda}_1$ à l'équation du modèle dynamique

2. On considère pour cette partie que le noyau de la matrice $\mathbf{A}$ est de dimension 1, ce qui sera généralement vérifié dans les cas d'application. Si la dimension du noyau de $\mathbf{A}$ est plus élevée, il conviendra alors de ne plus considérer seulement un vecteur $\dot{\mathbf{x}}_s$ mais une base du noyau de la matrice $\mathbf{A}$.

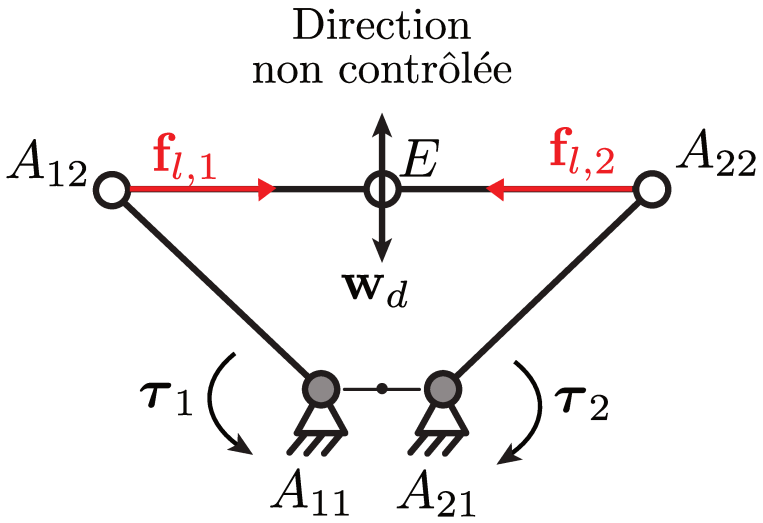


FIGURE 1.20 – Robot cinq barres en singularité de Type 2. Les forces $\mathbf{f}_{l,1}$ et $\mathbf{f}_{l,2}$ transmises à l'effecteur au travers du mécanisme sont orthogonales à la direction non contrôlée.

(E1.19). En effet, (E1.22) implique que $\mathbf{w}_d$ appartient à l'orthogonal du noyau de la matrice $\mathbf{A}$. Cela revient à dire que $\mathbf{w}_d$ appartient à l'image de la matrice $\mathbf{A}^T$ [3]. L'équation (E1.19) admet alors une infinité de solutions en $\boldsymbol{\lambda}_1$. Il existe localement une surcontrainte dans le mécanisme. Par exemple, sur la figure 1.20, $\mathbf{f}_{l,1}$ et $\mathbf{f}_{l,2}$ agissent dans la même direction. Plusieurs combinaisons de ces deux forces sont possibles pour générer une accélération de l'effecteur respectant le critère établi en (E1.22).

Si l'on désire obtenir la solution du modèle dynamique (E1.18)-(E1.19) minimisant la norme 2 des efforts en entrée, on pré-multiplie l'équation (E1.18) par $\mathbf{A}^T\mathbf{B}^{-T}$ et on introduit (E1.19) :

$$\mathbf{A}^T\mathbf{B}^{-T}\boldsymbol{\tau} = \mathbf{A}^T\mathbf{B}^{-T}\mathbf{w}_b - \mathbf{w}_d \tag{E1.23}$$

La solution du modèle dynamique (E1.23) minimisant $||\boldsymbol{\tau}||_2$ s'écrit alors :

$$\boldsymbol{\tau} = (\mathbf{J}^T)^+(\mathbf{J}^T\mathbf{w}_b - \mathbf{w}_d) \tag{E1.24}$$

avec $\mathbf{J} = \mathbf{B}^{-1}\mathbf{A}$ et $(\mathbf{J}^T)^+ = \mathbf{J}(\mathbf{J}^T\mathbf{J})^{-1}$ la pseudo-inverse de Moore-Penrose de la matrice $\mathbf{J}^T$. La solution obtenue par l'équation (E1.24) est une solution exacte du modèle dynamique inverse car $\mathbf{w}_d \in \mathrm{Im}\mathbf{A}^T \Rightarrow \mathbf{w}_d \in \mathrm{Im}\mathbf{J}^T \Rightarrow \mathbf{J}^T\mathbf{w}_b - \mathbf{w}_d \in \mathrm{Im}\mathbf{J}^T$.

Cette solution pourra être exploitée dans une loi de commande pour effectuer la traversée de la singularité. Les détails de la mise en œuvre d'une telle loi de commande sont donnés dans l'annexe C.

3. $(\mathrm{Ker}\mathbf{A})^\perp = \mathrm{Im}\mathbf{A}^T$ pour $\mathbf{A}$ une matrice de réels.

1.4 Conclusion

Dans ce chapitre, nous avons présenté un aperçu des robots manipulateurs aériens et des robots parallèles. Le concept développé dans cette thèse se situe au croisement de ces deux domaines. Nous avons introduit le modèle du quadricoptère et rappelé les méthodes de commande utilisées pour gérer le sous-actionnement de cet appareil. Ce sous-actionnement présente un enjeu pour la gestion de la stabilité d'un quadricoptère associé à un manipulateur aérien. Cette problématique, ainsi que l'autonomie limitée, restent un frein important pour le développement des manipulateurs aériens.

Des solutions de coopération entre plusieurs quadricoptères ont été étudiées pour augmenter la charge utile des manipulateurs aériens mais elles sont limitées dans leurs capacités de manipulation. Une liaison directe des quadricoptères avec l'objet manipulé restreint les possibilités de taille et de configuration de l'objet. Tandis que le transport de charges à l'aide de câbles n'offre pas la possibilité de produire des efforts de poussée via les câbles. C'est notamment cette limitation que nous cherchons à dépasser par la proposition d'un nouveau type de robot aérien, le robot parallèle volant.

Ce nouveau concept de robot est une alternative inédite aux manipulateurs aériens existants. Il présente des similarités avec les robots parallèles à base fixe. Nous avons donc introduit pour les robots parallèles des éléments théoriques qui pourront être transposés dans le cadre des robot parallèles volants : les modèles géométriques, cinématiques et dynamiques ainsi qu'une présentation de certaines singularités.

Le chapitre suivant présente les modèles et lois de commande établis pour les robots parallèles volants.

Chapitre 2

Modélisation et commande d'un robot parallèle volant

Sommaire

2.1 Un nouveau concept de manipulateur aérien 36
2.2 Cinématique d'un robot parallèle volant 38
2.2.1 Description cinématique 38
2.2.2 Choix des variables 39
2.2.3 Torseurs cinématiques et matrices jacobiennes 43
2.3 Modèle dynamique d'un robot parallèle volant 43
2.3.1 Modèle dynamique de l'architecture passive 43
2.3.2 Modèle dynamique de la rotation d'un drone attaché en son centre de gravité 47
2.3.3 Modèle dynamique du robot parallèle volant 49
2.4 Commande d'un robot parallèle volant 51
2.4.1 Entrées et sorties du modèle dynamique 52
2.4.2 Commande de l'architecture passive 55
2.4.3 Commande de l'orientation des drones 56
2.4.4 Commande du robot volant complet 58
2.4.5 Commande adaptative de l'architecture passive 59
2.5 Conclusion 62

Ce chapitre introduit le nouveau concept de robot parallèle volant. Une première partie présente le concept et les avantages attendus de ce nouveau robot aérien. Une seconde partie définit les paramètres cinématiques. La troisème partie de ce chapitre s'attarde sur l'étude du modèle dynamique et montre que celui-ci peut être séparé en deux systèmes d'équations : l'un décrivant la dynamique de l'architecture passive du robot volant et l'autre décrivant la dynamique des coordonnées d'orientation des drones actionneurs. La séparation de ces deux systèmes d'équations sera exploitée pour concevoir une loi de commande en cascade, qui fera l'objet d'une quatrième partie.

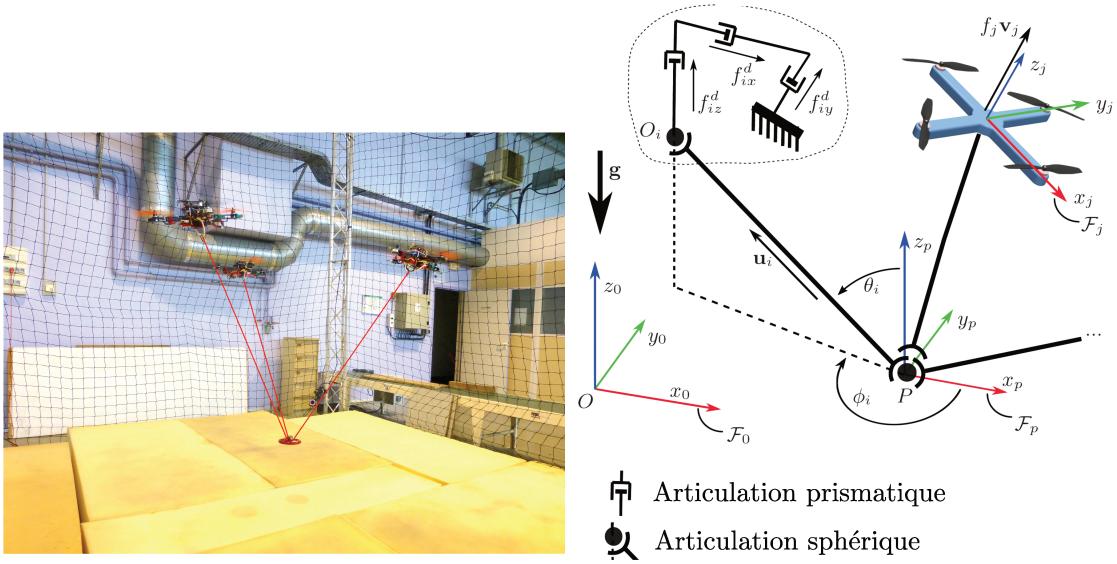
2.1 Un nouveau concept de manipulateur aérien

Dans la partie 1.2 nous avons présenté le transport de charge par drones à l'aide de câbles. Une autre façon de voir ce manipulateur aérien est de considérer qu'il s'agit un robot parallèle à câbles pour lequel les actionneurs des câbles ont été remplacés par des quadricoptères (Fig. 2.1). La figure 2.1 illustre cette analogie. A partir de cette analogie, il est possible d'imaginer remplacer les actionneurs d'un robot parallèle rigide par des quadricoptères pour obtenir un nouveau type de robot aérien. Le robot ainsi obtenu combine une architecture passive de robot parallèle avec la capacité de vol des drones, en particulier des quadricoptères (voir Fig. 2.2).

La réalisation d'un tel concept permettrait d'obtenir des propriétés intéressantes décrites ci-dessous :

- La répartition des efforts sur plusieurs drones, permettant d'augmenter la charge utile globale du robot. Il convient de limiter le poids du mécanisme passif pour conserver cet avantage.
- L'absence de moteurs supplémentaires en dehors des rotors déjà présents sur les drones. Dans le cas d'un système drone associé à un bras manipulateur, des moteurs supplémentaires sont nécessaires pour le contrôle de l'effecteur, diminuant l'énergie disponible pour le drone. Dans le cas du nouveau concept de robot parallèle volant, la structure est composée d'articulations passives uniquement commandées par la poussée des drones.
- L'effecteur peut être déporté loin du drone par l'usage d'une architecture passive adaptée, sans déstabiliser l'ensemble. Cela permet de ne pas perturber la zone de travail de l'effecteur par le flux d'air des drones.
- L'utilisation de corps rigides, combinée aux possibilités de reconfiguration de l'architecture passive, peut mener à effectuer des tâches au-dessus et en-dessous du plan contenant les drones à partir du même système.
- Un large choix d'architectures (voir partie 1.3.1), présentes dans la littérature des robots parallèles, permet d'envisager des robots volants ayant différentes propriétés cinématiques et dynamiques.

Nous allons maintenant aborder la modélisation de ce nouveau type de robot aérien.



(a) (b)

FIGURE 2.1 – Exemple de robot aérien à câbles (a) et schéma d'une architecture virtuelle équivalente pour l'un des quadricoptère (b) [Erskine, 2018].

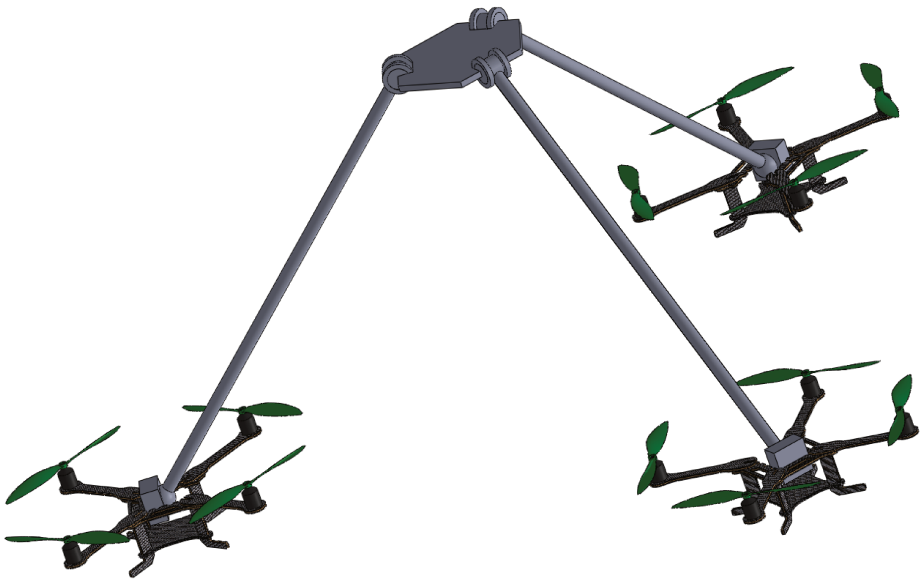


FIGURE 2.2 – Exemple de robot parallèle volant à trois drones.

2.2 Cinématique d'un robot parallèle volant

Nous allons effectuer dans cette partie une description cinématique du nouveau concept de robot parallèle volant. Ensuite, les paramètres cinématiques du robot seront donnés. Enfin, nous définirons les matrices jacobiennes cinématiques associées à chaque jambe.

2.2.1 Description cinématique

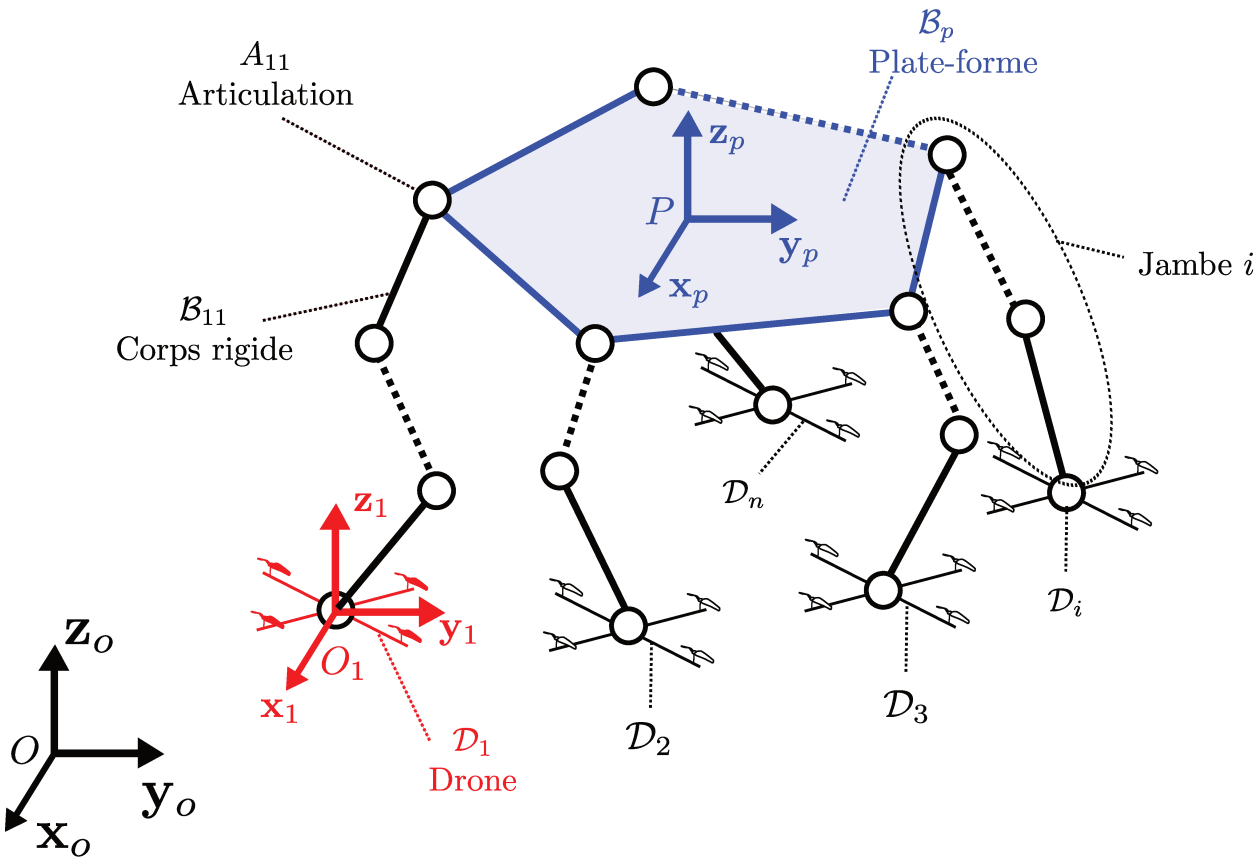


FIGURE 2.3 – Schéma général d'un robot volant à n drones.

Dans le cadre de cette thèse, le concept étudié est une architecture poly-articulée composée d'une plate-forme à laquelle sont reliées n chaînes cinématiques appelées jambes (voir Fig. 2.3). Chaque jambe est composée de corps rigides reliés par des articulations passives. Un drone est attaché à l'extrémité de chaque jambe opposée à la plate-forme. Pour cette étude, nous avons sélectionné des quadricoptères car ce drone est répandu, facilement personnalisable et présente des propriétés de vol en espace restreint adaptées au cadre des expérimentations souhaitées. L'extension de l'étude à d'autres types de drones peut être envisagée mais ne fait pas partie du cadre de cette thèse.

Dans ce chapitre, nous considérons que les drones sont reliés aux jambes de l'architecture

passive par des articulations sphériques situées au centre de gravité des drones. Cette hypothèse assure un découplage entre la rotation des drones et les mouvements de la structure. Ce découplage (voir partie 2.3) est indispensable pour la conception d'une commande robuste en cascade pour le robot. Un exemple de robot sans ce découplage ainsi qu'une discussion sur les problématiques qui en découlent seront présentés dans le chapitre 3.

2.2.2 Choix des variables

Les différents corps composant le robot parallèle volant sont nommés (voir Fig. 2.3) :

- $\mathcal{B}_p$ pour la plate-forme,
- $\mathcal{B}_{ij}$ pour le j-ième corps de la jambe i, en partant de la plate-forme,
- A_{ij} pour la j-ième articulation de la jambe i, en partant de la plate-forme,
- $\mathcal{D}_i$ pour le drone associé à la jambe i.

Les repères suivants sont également définis :

- $\mathcal{F}_o$ un repère orthonormé associé au sol d'origine le point O et d'axes $\mathbf{x}_o$, $\mathbf{y}_o$ et $\mathbf{z}_o$,
- $\mathcal{F}_p$ un repère orthonormé associé à la plate-forme $\mathcal{B}_p$ d'origine le point P et d'axes $\mathbf{x}_p$, $\mathbf{y}_p$ et $\mathbf{z}_p$.
- Pour chaque drone $\mathcal{D}_i$, $\mathcal{F}_i$ un repère associé au drone $\mathcal{D}_i$ d'origine le centre de gravité du drone O_i et d'axes $\mathbf{x}_i$, $\mathbf{y}_i$ et $\mathbf{z}_i$.

Dans la suite de ce chapitre, un vecteur $\mathbf{v}$ sera noté ${}^o\mathbf{v}$ s'il est exprimé dans le repère d'origine $\mathcal{F}_o$, ${}^p\mathbf{v}$ s'il est exprimé dans le repère associé à la plate-forme $\mathcal{F}_p$ et ${}^i\mathbf{v}$ s'il est exprimé dans le repère $\mathcal{F}_i$ associé au drone i.

La position et l'orientation du repère $\mathcal{F}_p$ sont données par six coordonnées $(x, y, z, \psi, \theta, \varphi)$. Les trois premières indiquent les trois coordonnées du point P selon les axes associés du repère $\mathcal{F}_o$ et les trois suivantes indiquent la représentation de l'orientation du repère $\mathcal{F}_p$ à l'aide de la convention des angles de Bryant [Diebel, 2006]. Soit un vecteur ${}^p\mathbf{v}$ défini dans le repère $\mathcal{F}_p$. Le même vecteur ${}^o\mathbf{v}$ exprimé dans le repère $\mathcal{F}_o$ sera donné par :

$$ {}^o\mathbf{v} = \mathbf{R}_{\mathbf{x}_o}(\varphi)\mathbf{R}_{\mathbf{y}_o}(\theta)\mathbf{R}_{\mathbf{z}_o}(\psi){}^p\mathbf{v} \tag{E2.1} $$

avec $\mathbf{R}_{\mathbf{u}}(\alpha)$ la rotation élémentaire d'axe $\mathbf{u}$ et d'angle α. Cette convention présente l'inconvénient de posséder des singularités de représentation. Une autre représentation des orientations sans singularité telle que les quaternions, déjà utilisée dans le cadre d'une loi de commande établie pour les quadricoptères [Carino et al., 2015], peut être envisagée. Cependant, les configurations de l'application expérimentale effectuée dans le cadre de cette thèse sont éloignées des singularités de la représentation choisie. De plus, la convention choisie présente l'avantage d'être aisément transposable vers une architecture virtuelle composée de trois pivots successifs (voir ci-après).

La position du point O_i et l'orientation du repère $\mathcal{F}_i$ du drone $\mathcal{D}_i$ sont exprimées par six coordonnées $(x_i, y_i, z_i, \varphi_i, \theta_i, \psi_i)$ définies de façon similaire à celles du repère $\mathcal{F}_p$. Ainsi,

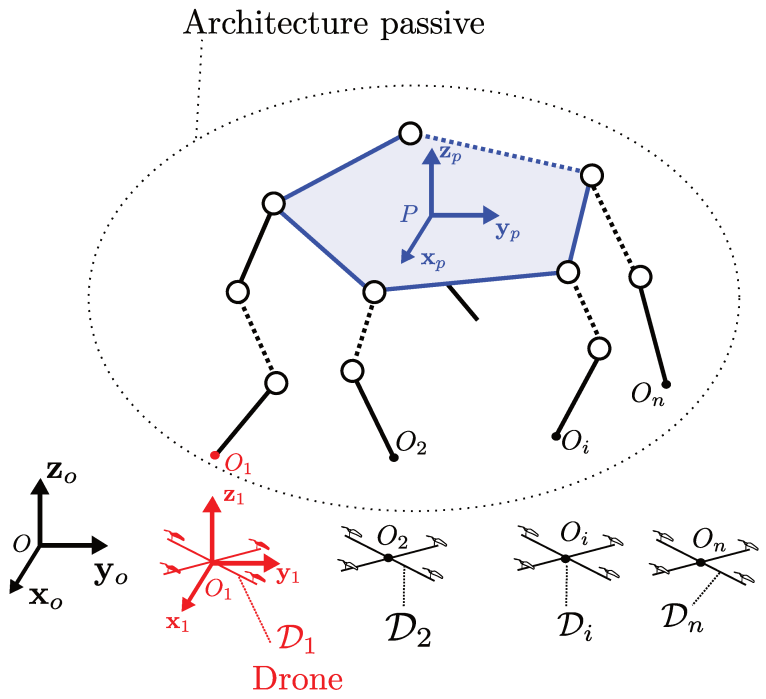


FIGURE 2.4 – Séparation virtuelle des drones et de l'architecture passive du robot parallèle volant.

l'expression développée de la matrice $\mathbf{R}_i = \mathbf{R}_{\mathbf{x}_o}(\varphi_i)\mathbf{R}_{\mathbf{y}_o}(\theta_i)\mathbf{R}_{\mathbf{z}_o}(\psi_i)$ de rotation du drone $\mathcal{D}_i$ s'écrit :

$$\mathbf{R}_i = \begin{bmatrix} \cos\theta_i \cos\psi_i & -\cos\theta_i \sin\psi_i & \sin\theta_i \\ \cos\varphi_i \sin\psi_i + \cos\psi_i \sin\varphi_i \sin\theta_i & \cos\varphi_i \cos\psi_i - \sin\varphi_i \sin\theta_i \sin\psi_i & -\cos\theta_i \sin\varphi_i \\ \sin\varphi_i \sin\psi_i - \cos\varphi_i \cos\psi_i \sin\theta_i & \cos\psi_i \sin\varphi_i + \cos\varphi_i \sin\theta_i \sin\psi_i & \cos\varphi_i \cos\theta_i \end{bmatrix} \quad \text{(E2.2)}$$

L'architecture passive du robot parallèle volant est définie comme étant composée :

- de la plate-forme,
- des jambes passives du robot, sans les drones actionneurs ni les articulations sphériques attachées à ces drones.

La modélisation des robots parallèles implique généralement de séparer virtuellement la plate-forme du reste du robot pour définir une architecture arborescente à partir de la base du robot (voir partie 1.3.3). Dans le cas de notre système, nous allons séparer virtuellement les drones du robot complet. Ainsi, nous obtiendrons une structure arborescente passive à partir de la plate-forme d'un côté et n drones de l'autre côté (voir Fig. 2.4). Le modèle dynamique (voir partie 2.3) sera alors calculé pour chacun des éléments virtuellement séparés puis l'expression des forces d'interactions entre ces éléments au niveau des articulations sphériques permettra d'établir le modèle du robot complet.

Pour définir l'architecture arborescente, nous choisirons d'utiliser la convention de Denavit-

Hartenberg modifiée par Khalil et Kleinfinger (DHKK) [Khalil et Kleinfinger, 1986]. La convention DHKK est un outil efficace pour définir la cinématique d'un robot sériel ou d'une structure arborescente. Cette convention utilise un ensemble de paramètres pour décrire la position et l'orientation d'un corps à partir du corps le précédant dans la chaîne cinématique : *son antécédent*. Chaque corps est relié à son antécédent par une articulation pivot ou glissière. Les liaisons plus complexes (sphériques, cardan,etc.) sont représentées par un enchaînement de plusieurs articulations. Le détail de cette convention et de la définition des repères associés à chaque corps est décrit dans [Khalil et Kleinfinger, 1986]. Les paramètres de cette convention sont, pour un corps rigide i donné :

- ant_i le corps précédant le corps i dans la chaîne cinématique,
- μ_i égal à 0 (respectivement 1) si l'articulation entre les deux corps est passive (respectivement active),
- σ_i égal à 0 (respectivement 1) si l'articulation entre les deux corps est un pivot (respectivement une glissière),
- α_i, d_i, θ_i et r_i des paramètres définissant la position et l'orientation du repère associé au corps i à partir du repère associé à son antécédent,
- γ_i et b_i deux paramètres supplémentaires nécessaires pour la définition des repères dans une structure arborescente.

La définition de la structure passive selon cette convention permet l'application directe des algorithmes connus pour le calcul des modèles géométriques, cinématiques et dynamiques [Khalil et Creusot, 1997].

Afin de représenter les mobilités de la plate-forme par la convention DHKK, nous allons définir une architecture virtuelle de type sérielle reliant le repère $\mathcal{F}_p$ au repère $\mathcal{F}_o$. Cette architecture virtuelle sans masse ni inertie, représentée en Fig. 2.5, est composée de corps sans masse reliés par six articulations successives (trois articulations glissières et trois pivots). Ces six articulations permettent à la plate-forme d'effectuer librement les trois translations et les trois rotations correspondant à ses six degrés de liberté. Le choix de l'ordre des articulations est lié à l'ordre des rotations pour la représentation des orientations choisies (E2.1). Notons que l'architecture virtuelle présente alors les mêmes singularités que la convention de représentation des orientations. Les paramètres selon la convention DHKK de cette architecture virtuelle équivalente sont décris dans la table (T2.1). Cette table peut être complétée depuis la plate-forme pour définir l'ensemble de l'architecture arborescente passive. Elle sera spécifique à chaque robot (voir les exemples tables (T3.3) et (T3.1) du chapitre 3).

Le vecteur des coordonnées généralisées de l'architecture passive sera ainsi constitué :

- des coordonnées de position et d'orientation de la plate-forme x, y, z, φ, θ et ψ,
- des coordonnées associées aux mobilités internes de chaque jambe, spécifiques à chaque robot.

Ces coordonnées généralisées de l'architecture passive sont regroupées dans un vecteur nommé

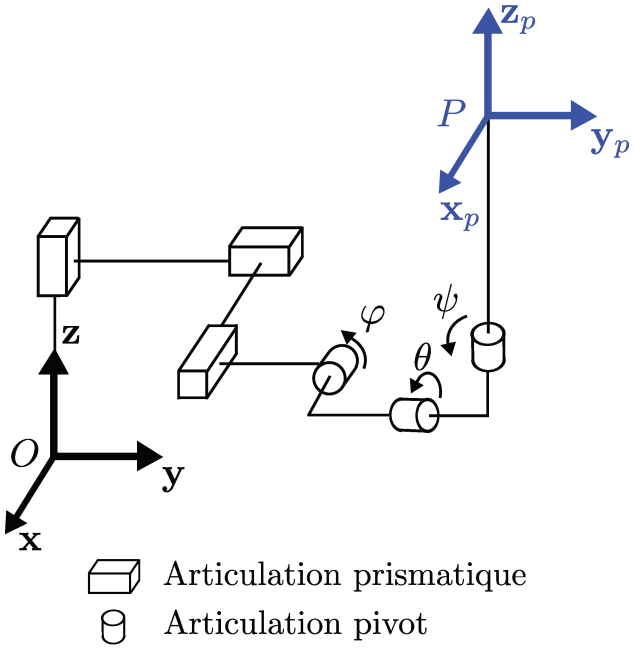


FIGURE 2.5 – Architecture virtuelle équivalente pour la définition de la position et de l'orientation de la plate-forme.

i	ant_i	μ_i	σ_i	γ_i	b_i	α_i	d_i	θ_i	r_i
$\mathcal{V}_1$	$\mathcal{F}_o$	0	1	0	0	0	0	0	z
$\mathcal{V}_2$	$\mathcal{V}_1$	0	1	0	0	$-\frac{\pi}{2}$	0	$-\frac{\pi}{2}$	y
$\mathcal{V}_3$	$\mathcal{V}_2$	0	1	0	0	$-\frac{\pi}{2}$	0	0	x
$\mathcal{V}_4$	$\mathcal{V}_3$	0	0	0	0	0	0	φ	0
$\mathcal{V}_5$	$\mathcal{V}_4$	0	0	0	0	$\frac{\pi}{2}$	0	$\theta + \frac{\pi}{2}$	0
$\mathcal{B}_p$	$\mathcal{V}_5$	0	0	0	0	$\frac{\pi}{2}$	0	ψ	0

TABLE 2.1 – Coordonnées DHKK d'une strucure virtuelle équivalente, permettant de définir la position et l'orientation de la plate-forme dans l'espace.

$\mathbf{q}_p$. Le nombre de coordonnées généralisées de l'architecture passive sera défini par le paramètre n_p.

2.2.3 Torseurs cinématiques et matrices jacobiennes

L'architecture passive a été définie comme une structure ouverte arborescente (voir Fig. 2.4). Soit ${}^o\mathbf{t}_i$ le torseur cinématique au point O_i du dernier corps (le plus éloigné de la plate-forme) de la jambe i exprimé dans le repère de référence $\mathcal{F}_o$. Il est possible de calculer l'expression de ce torseur en fonction de la dérivée des coordonnées généralisées de l'architecture passive $\mathbf{q}_p$ [Renaud, 1980]. Ce calcul prend la forme :

$$ {}^o\mathbf{t}_i = \mathbf{J}_i \dot{\mathbf{q}}_p \tag{E2.3} $$

$\mathbf{J}_i$ est une matrice $(6 \times n_p)$ appelée matrice jacobienne cinématique de la jambe i. En séparant les vitesses de translation et les vitesses de rotation dans le torseur ${}^o\mathbf{t}_i$, on écrit :

$$ \begin{bmatrix} {}^o\mathbf{v}_i \\ {}^o\boldsymbol{\omega}_i \end{bmatrix} = \begin{bmatrix} \mathbf{J}_{v,i} \\ \mathbf{J}_{\omega,i} \end{bmatrix} \dot{\mathbf{q}}_p \tag{E2.4} $$

Le calcul des matrices jacobiennes dépend de l'architecture mécanique mais son expression est connue de façon systématique pour tout robot défini selon la convention DHKK [Khalil et Dombre, 2002] (voir également l'annexe A.1).

2.3 Modèle dynamique d'un robot parallèle volant

Le modèle dynamique du robot sera calculé en trois étapes détaillées ci-après :

1. L'architecture passive est soumise aux torseurs d'actions mécaniques produits par les drones actionneurs sur le dernier corps de chaque jambe. Le modèle dynamique reliant ces torseurs d'action mécanique à l'accélération des coordonnées $\mathbf{q}_p$ sera établi en premier.
2. Le modèle dynamique des coordonnées d'orientation des drones actionneurs sous l'effet des moments produits par leurs hélices sera établi ensuite.
3. À partir des deux modèles précédemment calculés, un modèle dynamique complet est développé, mettant en évidence un découplage entre ces deux parties du modèle dynamique.

2.3.1 Modèle dynamique de l'architecture passive

Pour étudier le modèle dynamique de l'architecture passive, celle-ci est séparée virtuellement des drones actionneurs (voir Fig. 2.4). Établissons d'abord le modèle dynamique de l'architecture passive soumise aux torseurs d'action mécanique d'interaction entre les drones

et l'architecture passive, qui seront appelés *torseurs d'interaction*. Le principe fondamental de la dynamique sera ensuite appliqué sur chaque drone pour obtenir l'expression des torseurs d'interaction. Les deux équations combinées engendreront le modèle dynamique de l'architecture passive en fonction des forces de poussée des drones actionneurs.

Modèle dynamique de l'architecture passive soumise aux forces d'interaction

L'architecture passive est soumise aux torseurs d'action mécanique d'interaction avec les drones actionneurs. Pour chaque jambe i, un torseur d'interaction est appliqué sur le dernier corps de la jambe. Ce torseur d'interaction au point O_i exprimé dans le repère d'origine $\mathcal{F}_o$ sera noté ${}^o\mathbf{w}_{p,i}$ pour la jambe i.

Rappelons que le vecteur des coordonnées de l'architecture passive est noté $\mathbf{q}_p$. Chaque composante de ce vecteur donne la coordonnée de l'une des articulations passives de l'architecture. Le Lagrangien associé à l'architecture passive s'écrit $L_p(\mathbf{q}_p, \dot{\mathbf{q}}_p) = E_p(\mathbf{q}_p, \dot{\mathbf{q}}_p) - U_p(\mathbf{q}_p)$, où E_p et U_p sont respectivement ses énergies cinétique et potentielle. On considère un vecteur d'actionnements virtuels $\boldsymbol{\tau}_p$ exercé à chaque articulation de l'architecture passive. Les équations de Lagrange pour l'architecture passive s'écrivent [Khalil et Dombre, 2002] :

$$\frac{\mathrm{d}}{\mathrm{d}t}\left(\frac{\partial L_p}{\partial \dot{\mathbf{q}}_p}\right)^T - \left(\frac{\partial L_p}{\partial \mathbf{q}_p}\right)^T = \boldsymbol{\tau}_p \tag{E2.5}$$

Les actionnements virtuels $\boldsymbol{\tau}_p$ peuvent être exprimés en fonction des torseurs d'effort d'interaction $\mathbf{w}_{p,i} = \begin{bmatrix}\mathbf{f}_{p,i}^T & \mathbf{m}_{p,i}^T\end{bmatrix}^T$ par le principe des puissances virtuelles. Soit un mouvement virtuel infinitésimal quelconque de l'architecture passive. On définit $\dot{\mathbf{q}}_p^*$ le vecteur des vitesses virtuelles des coordonnées $\mathbf{q}_p$ et $\mathbf{t}_i^*$ le torseur cinématique virtuel du dernier corps de chaque jambe i appliqué au point O_i. La puissance virtuelle dans les articulations passives est égale à la somme des puissances fournies par chaque torseur d'interaction sur le dernier corps de chaque jambe i. On écrit alors :

$$\dot{\mathbf{q}}_p^{*T}\boldsymbol{\tau}_p = \sum_{i=1}^{n} {}^o\mathbf{t}_i^{*T}\,{}^o\mathbf{w}_{p,i} \tag{E2.6}$$

En introduisant la matrice cinématique jacobienne de chaque jambe (E2.3), le principe des puissances virtuelles s'écrit alors :

$$\dot{\mathbf{q}}_p^{*T}\boldsymbol{\tau}_p = \dot{\mathbf{q}}_p^{*T}\sum_{i=1}^{n} \mathbf{J}_i^{To}\mathbf{w}_{p,i} \tag{E2.7}$$

Par identification des deux membres de l'équation (E2.7), valables pour tout vecteur $\dot{\mathbf{q}}_p^*$, on obtient :

$$\boldsymbol{\tau}_p = \sum_{i=1}^{n} \mathbf{J}_i^{To}\mathbf{w}_{p,i} \tag{E2.8}$$

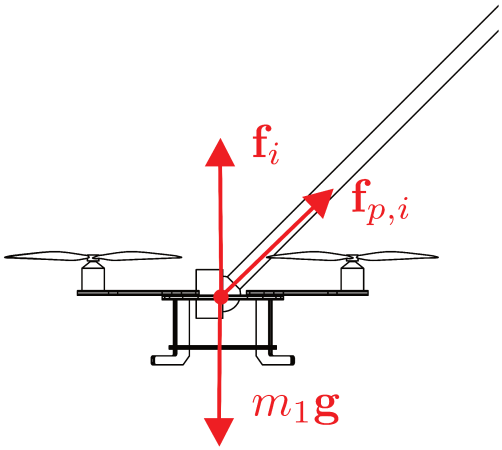


FIGURE 2.6 – Illustration du bilan des forces s'exerçant au centre de gravité d'un drone.

En introduisant (E2.8) dans (E2.5), on obtient :

$$\frac{\mathrm{d}}{\mathrm{dt}}\left(\frac{\partial L_p}{\partial \dot{\mathbf{q}}_p}\right)^T - \left(\frac{\partial L_p}{\partial \mathbf{q}_p}\right)^T = \sum_{i=1}^{n} \mathbf{J}_i^{T\,o}\mathbf{w}_{p,i} \tag{E2.9}$$

Cette équation peut être mise sous une forme linéaire en l'accélération des coordonnées $\mathbf{q}_p$ [Khalil et Dombre, 2002] (voir également annexe A.2) :

$$\mathbf{M}_p\ddot{\mathbf{q}}_p + \mathbf{c}_p = \sum_{i=1}^{n_i} \mathbf{J}_i^{T\,o}\mathbf{w}_{p,i} \tag{E2.10}$$

$\mathbf{M}_p$ est la matrice d'inertie généralisée de l'architecture passive, symétrique et définie positive. $\mathbf{c}_p$ est un vecteur regroupant les effets de gravité, centrifuges et de Coriolis.

Calcul du torseur d'interaction en fonction de l'actionnement des drones

Le torseur d'interaction entre le drone $\mathcal{D}_i$ et le dernier corps de la jambe i, ${}^o\mathbf{w}_{p,i}$, est composé d'un moment ${}^o\mathbf{m}_{p,i}$ et d'une force ${}^o\mathbf{f}_{p,i}$. Comme l'architecture passive est reliée au drone $\mathcal{D}_i$ par une articulation sphérique, les moments transmis sont nuls, i.e. ${}^o\mathbf{m}_{p,i} = \mathbf{0}$. La relation entre les forces ${}^o\mathbf{f}_{p,i}$ et ${}^o\mathbf{f}_i$ s'obtient en appliquant le principe fondamental de la dynamique au centre de gravité du drone, exprimé dans le repère de référence $\mathcal{F}_o$ (voir aussi Fig. 2.6) :

$$m_i{}^o\ddot{\mathbf{r}}_i = {}^o\mathbf{f}_i + m_i{}^o\mathbf{g} - {}^o\mathbf{f}_{p,i} \tag{E2.11}$$

avec :

- ${}^o\mathbf{r}_i$ la position du centre de gravité du drone $\mathcal{D}_i$ (point O_i) exprimé dans le repère $\mathcal{F}_o$, et donc ${}^o\ddot{\mathbf{r}}_i$ son accélération.

- ${}^o\mathbf{f}_i$ la force produite par les hélices du drone actionneur $\mathcal{D}_i$ exprimée dans le repère $\mathcal{F}_o$. Cette force est appelée force de poussée du drone $\mathcal{D}_i$.
- ${}^o\mathbf{g} = [0\ 0\ -g]^T$ le vecteur de gravité exprimé dans le repère $\mathcal{F}_o$.
- m_i la masse du drone $\mathcal{D}_i$

La vitesse translationnelle du point O_i, ${}^o\mathbf{v}_i = {}^o\dot{\mathbf{r}}_i$, s'exprime à l'aide de la matrice jacobienne cinématique associée donnée par l'équation (E2.4) :

$$ {}^o\dot{\mathbf{r}}_i = \mathbf{J}_{v,i}\dot{\mathbf{q}}_p \tag{E2.12}$$

En introduisant la dérivée de (E2.12) dans (E2.11), il est possible d'exprimer la force d'interaction entre le drone i et l'architecture passive par :

$$ {}^o\mathbf{f}_{p,i} = {}^o\mathbf{f}_i + m_i{}^o\mathbf{g} + m_i(\dot{\mathbf{J}}_{v,i}\dot{\mathbf{q}}_p + \mathbf{J}_{v,i}\ddot{\mathbf{q}}_p) \tag{E2.13}$$

Modèle dynamique de l'architecture passive en fonction de la poussée des drones actionneurs

L'équation (E2.10) décrivant le modèle dynamique de l'architecture passive s'exprime en fonction des efforts d'interaction $({}^o\mathbf{m}_{p,i}, {}^o\mathbf{f}_{p,i})$ qui ont été calculés ci-avant. À partir de l'expression de ${}^o\mathbf{f}_{p,i}$, donnée par (E2.13), et en utilisant le fait que ${}^o\mathbf{m}_{p,i} = \mathbf{0}$, l'équation (E2.10) s'écrit :

$$ \mathbf{M}_p\ddot{\mathbf{q}}_p + \mathbf{c}_p = \sum_{i=1}^{n} \mathbf{J}_{v,i}^T({}^o\mathbf{f}_i - m_i\mathbf{J}_{v,i}\ddot{\mathbf{q}}_p - m_i\dot{\mathbf{J}}_{v,i}\dot{\mathbf{q}}_p + m_i{}^o\mathbf{g}) \tag{E2.14}$$

En regroupant les termes liés à la dérivée seconde $\ddot{\mathbf{q}}_p$ des coordonnées généralisées, on obtient :

$$ \left(\mathbf{M}_p + \sum_{i=1}^{n} m_i\mathbf{J}_{v,i}^T\mathbf{J}_{v,i}\right)\ddot{\mathbf{q}}_p + \mathbf{c}_p + \sum_{i=1}^{n} m_j\mathbf{J}_{v,i}^T(\dot{\mathbf{J}}_{v,i}\dot{\mathbf{q}}_p - {}^o\mathbf{g}) = \sum_{i=1}^{n} \mathbf{J}_{v,i}^T{}^o\mathbf{f}_i \tag{E2.15}$$

Soit $\mathbf{R}_i$ (E2.2) la matrice de rotation du repère local $\mathcal{F}_i$ du drone i par rapport au repère $\mathcal{F}_o$. La force de poussée du drone i dans le repère $\mathcal{F}_o$ s'écrit alors :

$$ {}^o\mathbf{f}_i = \mathbf{R}_i{}^i\mathbf{f}_i \tag{E2.16}$$

avec ${}^i\mathbf{f}_i$ la force de poussée du drone dans son repère local. La matrice $\mathbf{R}_i$ ne dépend que des coordonnées d'orientation du drone. Il est possible alors d'écrire le modèle dynamique de l'architecture passive (E2.15) sous la forme :

$$ \mathbf{M}_t\ddot{\mathbf{q}}_p + \mathbf{c}_t = \mathbf{J}_p^T\mathbf{R}_t\mathbf{f} \tag{E2.17}$$

avec :

- $\mathbf{M_t} = \mathbf{M}_p + \sum_{i=1}^{n} m_i \mathbf{J}_{v,i}^T \mathbf{J}_{v,i}$
  Cette matrice est composée de la matrice d'inertie généralisée de l'architecture passive $\mathbf{M}_p$ et de la somme des termes d'inertie associés à la masse des drones aux extrémités des jambes. Notons que cette matrice est symétrique définie positive. Elle est de taille $(n_p \times n_p)$.
- $\mathbf{c}_t = \mathbf{c}_p + \sum_{i=1}^{n} m_i \mathbf{J}_{v,i}^T (\dot{\mathbf{J}}_{v,i} \dot{\mathbf{q}}_p - {}^o\mathbf{g})$
  Ce vecteur est composé du vecteur $\mathbf{c}_p$ augmenté des effets de gravité, centrifuges et de Coriolis générés par la masse des drones aux extrémités des jambes. Ce vecteur est de longueur n_p.
- $\mathbf{J}_p = \begin{bmatrix} \mathbf{J}_{v,1} \\ \mathbf{J}_{v,2} \\ \vdots \\ \mathbf{J}_{v,n} \end{bmatrix}$
  Cette matrice est la concaténation des matrices jacobiennes associées à la vitesse linéaire de chaque point O_i. Notons que cette matrice n'est a priori pas carrée. Elle est de taille $(3n \times n_p)$.
- $\mathbf{R}_t = \begin{bmatrix} \mathbf{R}_1 & \mathbf{0} & \dots & \mathbf{0} \\ \mathbf{0} & \mathbf{R}_2 & \dots & \mathbf{0} \\ \vdots & \vdots & \ddots & \vdots \\ \mathbf{0} & \mathbf{0} & \dots & \mathbf{R}_n \end{bmatrix}$
  Cette matrice est la concaténation dans une matrice diagonale par blocs des matrices de rotation des repères locaux de chaque jambe i. Cette matrice est de taille $(3n \times 3n)$.
- $\mathbf{f} = \begin{bmatrix} {}^1\mathbf{f}_1 \\ \vdots \\ {}^n\mathbf{f}_n \end{bmatrix}$
  Ce vecteur est la concaténation des forces de poussée des drones, exprimées chacune dans leur repère local. Il est de longueur $3n$.

Notons qu'une autre méthode de calcul possible de ce modèle est de transférer virtuellement la masse du drone i au dernier corps de la jambes i au point O_i. La masse du drone est alors considérée comme nulle et la masse et l'inertie du dernier corps de la jambe sont recalculées par l'ajout d'une masse ponctuelle m_i en O_i. Comme l'articulation entre le drone i et le dernier corps de la jambe i est sphérique, les deux architectures sont équivalentes. La matrice $\mathbf{M}_t$ et le vecteur $\mathbf{c}_t$ sont alors directement obtenus à partir des équations de Lagrange (E2.9).

2.3.2 Modèle dynamique de la rotation d'un drone attaché en son centre de gravité

Chaque drone $\mathcal{D}_i$ est relié à l'architecture passive par une articulation sphérique située au point O_i. Connaissant la configuration des coordonnées $\mathbf{q}_p$ du robot, il est possible de calculer

la coordonnée de chaque point O_i. Ainsi, la position du centre de gravité du drone dans l'espace est déterminée par les coordonnées $\mathbf{q}_p$. Par ailleurs, ses coordonnées d'orientation $\boldsymbol{\eta}_i = [\varphi_i\ \theta_i\ \psi_i]^T$ ne sont pas contraintes cinématiquement par les coordonnées de la structure passive car l'articulation reliant la structure à chacun des drones est à la fois sphérique et passive.

Le modèle dynamique des coordonnées d'orientation d'un drone est obtenu en appliquant les équations de Lagrange ou de Newton-Euler au centre de gravité du drone. Le drone étant relié à l'architecture passive par une articulation sphérique, aucun moment autre que ceux générés par les hélices n'influe la dynamique de rotation du drone. Les équations de Newton-Euler s'écrivent donc :

$$\boldsymbol{\Sigma}_i {}^i\dot{\boldsymbol{\omega}}_i = -{}^i\boldsymbol{\omega}_i \times \boldsymbol{\Sigma}_i {}^i\boldsymbol{\omega}_i + {}^i\mathbf{m}_i \tag{E2.18}$$

avec

- Σ_i la matrice d'inertie du drone i exprimée dans son repère local $\mathcal{F}_i$,
- ${}^i\boldsymbol{\omega}_i$ la vitesse angulaire du drone i au point O_i exprimée dans son repère local,
- ${}^i\mathbf{m}_i$ les moments dynamiques au point O_i générés par les hélices du drone dans son repère local.

La vitesse angulaire du drone i est reliée à la dérivée de ses coordonnées d'orientation $\boldsymbol{\eta}_i$ par l'équation [Khalil et Dombre, 2002] :

$${}^i\boldsymbol{\omega}_i = \mathbf{T}_i \dot{\boldsymbol{\eta}}_i \tag{E2.19}$$

La matrice $\mathbf{T}_i$ dépend du type de représentation choisie pour l'orientation des drones. Dans notre cas (E2.1), elle s'écrit :

$$\mathbf{T}_i = \begin{bmatrix} \cos\theta_i \cos\psi_i & \sin\psi_i & 0 \\ -\cos\theta_i \sin\psi_i & \cos\psi_i & 0 \\ \sin\theta_i & 0 & 1 \end{bmatrix} \tag{E2.20}$$

En introduisant (E2.19) dans (E2.18), on obtient :

$$\boldsymbol{\Sigma}_i \mathbf{T}_i \ddot{\boldsymbol{\eta}}_i = -\boldsymbol{\Sigma}_i \dot{\mathbf{T}}_i \dot{\boldsymbol{\eta}}_i - \mathbf{T}_i \dot{\boldsymbol{\eta}}_i \times \boldsymbol{\Sigma}_i \mathbf{T}_i \dot{\boldsymbol{\eta}}_i + {}^i\mathbf{m}_i \tag{E2.21}$$

La dynamique des orientations des drones est obtenue rassemblant le système d'équations (E2.21) pour chaque drone i, allant de 1 à n, sous la forme :

$$\mathbf{M}_a \ddot{\mathbf{q}}_a + \mathbf{c}_a = \mathbf{m} \tag{E2.22}$$

avec :

- $\mathbf{q}_a = \begin{bmatrix} \boldsymbol{\eta}_1 \\ \vdots \\ \boldsymbol{\eta}_n \end{bmatrix}$

Ce vecteur est la concaténation des coordonnées d'orientation des drones, de longueur $3n$.

- $\mathbf{M}_a = \begin{bmatrix} \Sigma_1\mathbf{T}_1 & \mathbf{0} & \dots & \mathbf{0} \\ \mathbf{0} & \Sigma_2\mathbf{T}_2 & \dots & \mathbf{0} \\ \vdots & \vdots & \ddots & \vdots \\ \mathbf{0} & \mathbf{0} & \dots & \Sigma_n\mathbf{T}_n \end{bmatrix}$
  Cette matrice est la concaténation dans une matrice diagonale par blocs des matrices d'inerties locales des drones associées aux matrices de changement de repère $\mathbf{T}_i$. Elle est de taille $(3n \times 3n)$.
- $\mathbf{c}_a = \begin{bmatrix} \Sigma_1\dot{\mathbf{T}}_1\dot{\boldsymbol{\eta}}_1 + \mathbf{T}_1\dot{\boldsymbol{\eta}}_1 \times \Sigma_1\mathbf{T}_1\dot{\boldsymbol{\eta}}_1 \\ \vdots \\ \Sigma_n\dot{\mathbf{T}}_n\dot{\boldsymbol{\eta}}_n + \mathbf{T}_n\dot{\boldsymbol{\eta}}_n \times \Sigma_n\mathbf{T}_n\dot{\boldsymbol{\eta}}_n \end{bmatrix}$
  Ce vecteur regroupe les effets centrifuges et de Coriolis associés à la rotation des drones. Il est de longueur $3n$.
- $\mathbf{m} = \begin{bmatrix} \mathbf{m}_1^1 \\ \vdots \\ \mathbf{m}_n^n \end{bmatrix}$
  Ce vecteur est la concaténation des moments appliqués par les hélices de chaque drone dans son repère local. Il est de longueur $3n$.

Notons que tous les termes de la dynamique de rotation des drones (E2.22) sont indépendants des coordonnées de l'architecture passive $\mathbf{q}_p$ ou de leurs dérivées. Ces termes sont également indépendants de la force de poussée $\mathbf{f}_i$ des drones actionneurs. En d'autres termes, cette partie du modèle dynamique du robot est totalement indépendante de la dynamique de l'architecture passive.

2.3.3 Modèle dynamique du robot parallèle volant

Les coordonnées de tout le système $\mathbf{q}$ (drone et architecture passive) sont composées des coordonnées généralisées de l'architecture passive $\mathbf{q}_p$ et des coordonnées d'orientation des drones $\mathbf{q}_a$. Le système composé des équations (E2.17) et (E2.22) définit le modèle dynamique complet du robot parallèle volant. Nous les rappelons ici

$$\mathbf{M}_t\ddot{\mathbf{q}}_p + \mathbf{c}_t = \mathbf{J}_p^T\mathbf{R}_t\mathbf{f}$$
$$\mathbf{M}_a\ddot{\mathbf{q}}_a + \mathbf{c}_a = \mathbf{m}$$

Comme décrit dans la partie 2.3.2, la dynamique de rotation des drones (E2.22) est totalement indépendante de la dynamique de l'architecture passive (E2.17).

La dynamique de l'architecture passive (E2.17) ne contient qu'un seul terme dépendant des coordonnées d'orientation des drones actionneurs $\mathbf{q}_a$ ou de leurs dérivées. Il s'agit de la matrice de rotation $\mathbf{R}_t$. Cette matrice est en effet la concaténation, en une matrice diagonale

par blocs, des matrices de rotation $\mathbf{R}_i$ de chaque drone $\mathcal{D}_i$. On constate dans l'équation (E2.2) que chaque matrice de rotation $\mathbf{R}_i$ ne dépend que des coordonnées d'orientation $\boldsymbol{\eta}_i$ du drone $\mathcal{D}_i$. Ainsi, la matrice $\mathbf{R}_t$ ne dépend que des coordonnées du vecteur $\mathbf{q}_a$ qui est, pour rappel, la concaténation des vecteurs de coordonnées d'orientation $\boldsymbol{\eta}_i$. Ce couplage des deux dynamiques uniquement par les coordonnées d'orientation des drones va permettre de concevoir une loi de commande en cascade similaire à celles existantes pour un quadricoptère seul.

Pour compléter le modèle dynamique, il faut intégrer le modèle des quadricoptères, développé dans la partie 1.1.2, pour exprimer la force et le moment générés par chaque drone en fonction de la vitesse de rotation des hélices. La définition du repère $\mathcal{F}_i$ et de la position des rotors est la même pour tous les drones, similaire à celle développée dans la partie 1.1.1. Les paramètres caractéristiques des drones k_t (coefficient de poussée), k_d (coefficient de trainée) et d (distance) sont accolés de l'indice i du drone concerné. Ils ne sont pas nécessairement égaux pour tous les drones. Ainsi, pour chaque drone $\mathcal{D}_i$, le torseur dynamique ${}^i\mathbf{w}_i$ exercé par les hélices dans le repère local $\mathcal{F}_i$ est exprimé en fonction de la vitesse de rotation des hélices à partir de l'équation (E1.4) :

$$ {}^i\mathbf{w}_i = \mathbf{Q}_i\boldsymbol{\Omega}_i \tag{E2.23}$$

avec :

- $\boldsymbol{\Omega}_i$ le vecteur regroupant la vitesse de rotation $\omega_{i,j}$, $(j = 1..4)$ de chaque rotor du drone $\mathcal{D}_i$ élevée au carré,
- $\mathbf{Q}_i$ est la matrice de répartition des efforts générés par les rotors dont l'expression est donnée dans la partie 1.1.2.

Notons que cette dernière étape est indépendante des développements précédents du chapitre. Ainsi, la théorie présentée reste valable avec d'autres drones que les quadricoptères classiques. Notons également que le modèle ne contraint pas à disposer de drones similaires à chaque extrémité de l'architecture passive.

Comme montré dans la partie 1.1.2, pour un drone $\mathcal{D}_i$, nous pouvons indifféremment considérer en entrée du modèle dynamique le vecteur de vitesse de rotation de ses rotors Ω_i ou le vecteur composé de sa force de poussée selon l'axe $\mathbf{z}_i$, $f_{z,i}$, et de ses moments $\mathbf{m}_i$. Les autres composantes de la force de poussée sont nuls, ${}^i\mathbf{f}_i = [0\ 0\ f_{z,i}]^T$. L'équation (E2.17) peut alors s'écrire :

$$ \mathbf{M}_t\ddot{\mathbf{q}}_p + \mathbf{c}_t = \mathbf{J}_p^T\mathbf{R}_r\mathbf{f}_z \tag{E2.24}$$

avec :

- $\mathbf{f}_z$ le vecteur regroupant les composantes selon $\mathbf{z}_i$ des forces de poussée des drones ${}^i\mathbf{f}_i$ exprimées dans leurs repères locaux $\mathcal{F}_i$ respectifs. $\mathbf{f}_z = [f_{z,1}\ \ldots\ f_{z,n}]^T$. Ce vecteur est de longueur n.

- $\mathbf{R}_r$ la matrice $\mathbf{R}_t$ réduite aux colonnes correspondants aux composantes de poussée non nulles (troisième colonne de chaque matrice $\mathbf{R}_i$). Cette matrice est de taille $(3n \times n)$ et se présente sous la forme diagonale par blocs suivante, obtenue à partir de (E2.2) et (E2.17) :

$$\mathbf{R}_r = \begin{bmatrix} \begin{matrix} \sin\theta_1 \\ -\sin\varphi_1\cos\theta_1 \\ \cos\varphi_1\cos\theta_1 \end{matrix} & \mathbf{0} & \dots & \mathbf{0} \\ \mathbf{0} & \begin{matrix} \sin\theta_2 \\ -\sin\varphi_2\cos\theta_2 \\ \cos\varphi_2\cos\theta_2 \end{matrix} & \dots & \mathbf{0} \\ \vdots & \vdots & \ddots & \vdots \\ \mathbf{0} & \mathbf{0} & \dots & \begin{matrix} \sin\theta_n \\ -\sin\varphi_n\cos\theta_n \\ \cos\varphi_n\cos\theta_n \end{matrix} \end{bmatrix} \tag{E2.25}$$

Ainsi, le modèle dynamique du robot volant complet est composé des équations (E2.24) et (E2.22). Dans la partie suivante nous allons établir une commande permettant de stabiliser le robot et de suivre une trajectoire donnée pour les coordonnées de l'architecture passive $\mathbf{q}_p$. Les entrées considérées du modèle pour la conception de la commande sont $\mathbf{f}_z$ et $\mathbf{m}$. Les vitesses de rotation des rotors nécessaires pour obtenir ces entrées sont calculées en appliquant la relation (E1.5) à chaque quadricoptère.

2.4 Commande d'un robot parallèle volant

Dans cette partie, nous allons établir la commande pour le robot parallèle volant. Cette partie sera développée en cinq sous-parties. Une première sous-partie se consacrera à l'étude des entrées et des sorties du modèle dynamique. Puis la commande de l'architecture passive sera établie. Ensuite, la commande de l'orientation des drones sera développée. Une quatrième partie présentera l'interfaçage des deux commandes précédemment établies dans la loi de commande du robot complet. Enfin, la cinquième partie développera une proposition de commande adaptative pour l'architecture passive permettant de gérer les éventuelles variations des coefficients de poussée.

2.4.1 Entrées et sorties du modèle dynamique

Le modèle dynamique d'un robot parallèle volant est décrit par les équations (E2.24) et (E2.22) que nous rappelons ici :

$$\mathbf{M}_t\ddot{\mathbf{q}}_p + \mathbf{c}_t = \mathbf{J}_p^T\mathbf{R}_r\mathbf{f}_z$$
$$\mathbf{M}_a\ddot{\mathbf{q}}_a + \mathbf{c}_a = \mathbf{m}$$

Comme expliqué dans la partie 2.3.3, les entrées du modèle considérées sont, pour chaque quadricoptère $\mathcal{D}_i$ $(i = 1..n)$:

- une force $f_{z,i}$ selon l'axe $\mathbf{z}_i$,
- le vecteur composé de trois moments indépendants $\mathbf{m}_i$.

Ces entrées sont regroupées respectivement dans le vecteur $\mathbf{f}_z$ et dans le vecteur $\mathbf{m}$.

Dynamique de rotation des drones

Intéressons-nous d'abord à la dynamique de rotation des drones. L'équation (E2.22) comprend un nombre de variables d'actionnement (vecteur $\mathbf{m}$) égal au nombre de variables d'orientations que l'on souhaite commander (vecteur $\mathbf{q}_a$), soit $3n$ variables. L'utilisation d'un modèle dynamique inverse nous permettra de ramener l'équation du modèle dynamique (E2.22) à un double intégrateur, comme dans une commande par couples calculés [Khalil et Dombre, 2002] (voir partie 2.4.3). Il n'y a pas de difficulté particulière pour la commande de cette dynamique.

Dynamique de l'architecture passive

Le système d'équations (E2.24), établissant le modèle dynamique de l'architecture passive, dispose de n variables d'actionnement (vecteur $\mathbf{f}_z$) pour la commande de n_p variables (vecteur $\mathbf{q}_p$). Considérons d'abord un vecteur de commande auxiliaire $\boldsymbol{\nu}$ de longueur n_p tel que $\boldsymbol{\nu} = \mathbf{J}_p^T\mathbf{R}_r\mathbf{f}_z$. L'équation

$$\mathbf{M_t}\ddot{\mathbf{q}}_p + \mathbf{c}_t = \boldsymbol{\nu} \tag{E2.26}$$

peut elle aussi se ramener à un double intégrateur par inversion du modèle dynamique. Ainsi, l'étude de la commandabilité de notre système s'effectue dans l'étude de la capacité à générer le vecteur $\boldsymbol{\nu}$ à partir des variables d'actionnement $\mathbf{f}_z$. Il s'agit de trouver un vecteur $\mathbf{f}_z$ solution de :

$$\mathbf{J}_p^T\mathbf{R}_r\mathbf{f}_z = \boldsymbol{\nu} \tag{E2.27}$$

Dans un premier temps, nous allons chercher une commande auxiliaire $\boldsymbol{\nu}_J$ solution de :

$$\mathbf{J}_p^T\boldsymbol{\nu}_J = \boldsymbol{\nu} \tag{E2.28}$$

La matrice $\mathbf{J}_p^T$ est de dimension $(n_p \times 3n)$. Son rang dépend de la cinématique de l'architecture passive et de ses singularités et il devra être étudié pour chaque système [Khalil et Dombre, 2002]. Nous considérerons dans la suite de la partie théorique sur la commande que $3n$ est supérieur ou égal à n_p et que la matrice $\mathbf{J}_p^T$ est de rang plein. Le chapitre 5 abordera une méthode pour gérer les singularités de la matrice $\mathbf{J}_p$. Ainsi, dans la suite de cette partie, nous considérons que le robot évolue hors des configurations singulières de l'architecture passive. Une solution exacte de l'équation (E2.28) est donnée par :

$$\boldsymbol{\nu}_J = (\mathbf{J}_p^T)^+ \boldsymbol{\nu} \tag{E2.29}$$

avec $(\mathbf{J}_p^T)^+$ la pseudo-inverse de Moore-Penrose de la matrice $\mathbf{J}_p^T$, $(\mathbf{J}_p^T)^+ = \mathbf{J}_p(\mathbf{J}_p^T\mathbf{J}_p)^{-1}$. La pseudo-inverse est utilisée ici car le nombre de colonnes de la matrice $\mathbf{J}_p^T$ peut être strictement supérieur à son nombre de lignes. Dans ce cas, l'architecture passive est sur-actionnée et la solution proposée est la solution minimisant la norme du vecteur $\boldsymbol{\nu}_J$. Il est possible de s'appuyer sur le noyau de la jacobienne $\mathbf{J}_p^T$ pour trouver d'autres solutions dans le cadre d'applications spécifiques.

Nous cherchons maintenant un vecteur $\mathbf{f}_z$ solution de :

$$\mathbf{R}_r \mathbf{f}_z = \boldsymbol{\nu}_J \tag{E2.30}$$

La matrice $\mathbf{R}_r$ est de dimension $(3n \times n)$. Ainsi, il n'est donc a priori pas possible de trouver une solution exacte à l'équation (E2.30).

Pour résoudre ce problème, nous allons utiliser une méthode inspirée de la commande des quadricoptères [Mahony et al., 2012] [Wang et al., 2012]. L'entrée du modèle dynamique de l'architecture passive est étendue aux coordonnées d'orientations des drones. Ainsi, le problème posé devient : existe-il un vecteur $\mathbf{f}_z$ et des variables d'orientation $\varphi_i, \theta_i, \psi_i$ $(i = 1..n)$ solutions de l'équation (E2.30) ? Pour cela, développons l'équation (E2.30) à l'aide de (E2.25). Pour chaque jambe, les trois équations suivantes sont obtenues :

$$\begin{aligned} &\sin\theta_i f_{z,i} = \nu_{J,3i-2} \\ &-\sin\varphi_i \cos\theta_i f_{z,i} = \nu_{J,3i-1} \\ &\cos\varphi_i \cos\theta_i f_{z,i} = \nu_{J,3i} \end{aligned} \tag{E2.31}$$

avec $\nu_{J,k}$ la k-ième composante du vecteur $\boldsymbol{\nu}_J$. On constate que la solution ne dépend pas de l'orientation de la composante ψ_i du drone. Sachant que la poussée $f_{z,i}$ recherchée doit être positive, la solution de l'équation (E2.31) est donnée par :

$$\begin{aligned} f_{z,i} &= \sqrt{\nu_{J,3i-2}^2 + \nu_{J,3i-1}^2 + \nu_{J,3i}^2} \\ \varphi_i &= -\text{atan2}(\nu_{J,3i-1}, \nu_{J,3i}) \\ \theta_i &= \text{asin}\left(\frac{\nu_{J,3i-2}}{f_{z,i}}\right) \end{aligned} \tag{E2.32}$$

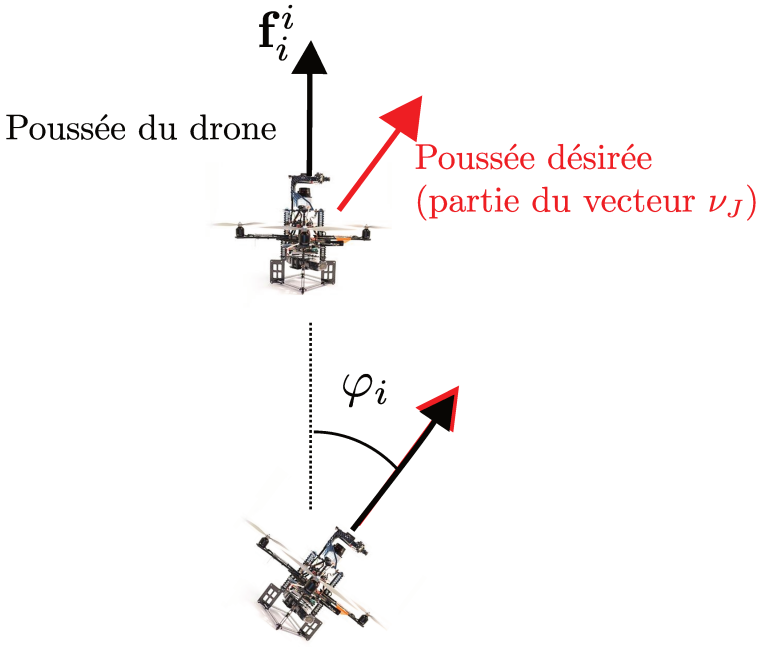


FIGURE 2.7 – Exemple, dans le plan, de rotation du drone solution de l'équation (E2.31).

Ce résultat peut être interprété géométriquement. En effet le vecteur $\boldsymbol{\nu}_J$ représente la concaténation de la poussée des drones exprimée dans le repère d'origine $\mathcal{F}_o$. Ainsi, trouver la solution de l'équation (E2.31) revient à trouver la poussée selon l'axe local $\mathbf{z}_i$ et l'orientation du drone $\mathcal{D}_i$ nécessaires à l'obtention de la poussée désirée $(\nu_{J,3i-2}, \nu_{J,3i-1}, \nu_{J,3i})^T$ exprimée dans le repère d'origine. Un exemple simplifié dans le plan est présenté Fig. 2.7. La solution (E2.32) présente une singularité pour $\nu_{J,3i} = 0$, i.e. pour une pousée horizontale. Elle ne sera pas atteinte en pratique car $\nu_{J,3i}$ représente la poussée désirée du drone $\mathcal{D}_i$ selon $\mathbf{z}_o$ dans le repère de référence $\mathcal{F}_o$. Cette poussée restera strictement positive tant que les dynamiques demandées ne compenseront pas le poids du drone i et la partie du poids de la structure exercée sur le drone.

La combinaison de l'équation (E2.29) et de l'équation (E2.32) donne une solution à l'équation (E2.27). Ainsi il est possible de contrôler l'architecture passive en considérant en entrée du système

- le vecteur $\mathbf{f}_z$ regroupant les poussées des drones,
- les coordonnées d'orientation des drones φ_i, θ_i $(i = 1..n)$.

Platitude différentielle

Une conséquence intéressante des développements ci-avant est que le modèle dynamique d'un robot parallèle volant est différentiellement plat pour la sortie $\boldsymbol{\mu} = \begin{bmatrix}\mathbf{q}_p^T & \psi_1 & \dots & \psi_n\end{bmatrix}^T$. En effet, il est possible d'exprimer le vecteur des états $[\mathbf{q}^T\ \dot{\mathbf{q}}^T]^T$ (rappel, $\mathbf{q}^T = [\mathbf{q}_p^T\ \mathbf{q}_a^T]$), ainsi

que les entrées $\mathbf{f}_z$ et $\mathbf{m}$, uniquement à partir du vecteur des sorties $\boldsymbol{\mu}$ et d'un nombre fini de ses dérivées. On obtient ce résultat de la façon suivante [1] :

- $\mathbf{q}_p$ et $\dot{\mathbf{q}}_p$ sont obtenus directement à partir des n_p premières composantes de $\boldsymbol{\mu}$ et $\dot{\boldsymbol{\mu}}$.
- Le vecteur $\boldsymbol{\nu}$ est calculé par l'équation (E2.26), qui est une fonction de $\mathbf{q}_p$, $\dot{\mathbf{q}}_p$ et $\ddot{\mathbf{q}}_p$. Le vecteur $\boldsymbol{\nu}_J = (\mathbf{J}_p^T)^+ \boldsymbol{\nu}$ (E2.29) l'est également car la matrice $\mathbf{J}_p$ ne dépend que de $\mathbf{q}_p$ [2].
- $\mathbf{f}_{z,i}$, ainsi que les angles φ_i et θ_i, pour $(i = 1 \dots n)$ sont obtenus à partir de $\boldsymbol{\nu}_J$ par l'équation (E2.32). Ainsi, le vecteur $\mathbf{f}_z$ est une fonction de $\mathbf{q}_p$, $\dot{\mathbf{q}}_p$ et $\ddot{\mathbf{q}}_p$ et, en conséquence, une fonction du vecteur $\boldsymbol{\mu}$ et de ses trois premières dérivées. Il en est de même pour $\mathbf{q}_a$ car les angles ψ_i sont les n dernières composantes du vecteur $\boldsymbol{\mu}$.
- Par dérivation de $\mathbf{q}_a$, $\dot{\mathbf{q}}_a$ est une fonction de la sortie $\boldsymbol{\mu}$ et de ses quatre premières dérivées.
- Par dérivation de $\dot{\mathbf{q}}_a$, $\ddot{\mathbf{q}}_a$ est une fonction de la sortie $\boldsymbol{\mu}$ et de ses cinq premières dérivées.
- Par l'équation du modèle dynamique des orientation (E2.22), le vecteur $\mathbf{m}$ est une fonction des vecteurs $\mathbf{q}_a$, $\dot{\mathbf{q}}_a$ et $\ddot{\mathbf{q}}_a$ et donc une fonction du vecteur $\boldsymbol{\mu}$ et de ses cinq premières dérivées.

Ces étapes suffisent à prouver que le système est différentiellement plat pour la sortie $\boldsymbol{\mu}$. Une conséquence utile de ce résultat est que toute trajectoire $\mathcal{C}^5$ définie pour la sortie $\boldsymbol{\mu}$ cinq fois continûment différentiable pourra être suivie par le robot volant.

En conclusion, deux dynamiques sont présentes dans le modèle. Une dynamique d'orientation des drones (E2.22), commandée par le vecteur $\mathbf{m}$, et une dynamique de l'architecture passive (E2.24), commandée par le vecteur des poussées $\mathbf{f}_z$ et les coordonnées d'orientation des drones. De plus, toute trajectoire suffisamment continue pour le vecteur de sortie $\boldsymbol{\mu} = \begin{bmatrix} \mathbf{q}_p^T & \psi_1 & \dots & \psi_n \end{bmatrix}^T$ pourra être suivie par le robot. Ces éléments permettent la mise en œuvre d'une loi de commande en cascade décrite ci-après.

2.4.2 Commande de l'architecture passive

Considérons le modèle dynamique de l'architecture passive avec pour entrée la commande auxiliaire $\boldsymbol{\nu}$ (E2.26) :

$$\mathbf{M}_t \ddot{\mathbf{q}}_p + \mathbf{c}_t = \boldsymbol{\nu}$$

Nous désirons faire suivre une trajectoire établie $\mathbf{q}_p^d$ aux coordonnées de l'architecture passive. L'erreur de suivi est définie par : $\mathbf{e}_p = \mathbf{q}_p - \mathbf{q}_p^d$. En effectuant une inversion dynamique, on peut

1. Les expressions développées de ces calculs ne sont pas données. Elles ne seront pas utilisées dans le cadre de cette thèse. Il est juste nécessaire de savoir qu'elles peuvent être calculées pour obtenir la propriété platitude différentielle.

2. Si la matrice $\mathbf{J}_p$ n'est pas carrée (robot sur-actionné), le vecteur $\mathbf{q}_p$ et ses dérivées ne permettent pas à eux seuls de calculer le vecteur $\boldsymbol{\nu}_J$. Il existe en effet plusieurs solutions pour le vecteur $\boldsymbol{\nu}_J$. La gestion de la sur-contrainte dans la commande détermine le choix de cette solution. Il est nécessaire d'en tenir compte pour connaître l'état du système à partir de la sortie.

ramener le système à un double intégrateur. Pour cela, définissons la commande auxiliaire $\boldsymbol{\nu}_p$ telle que :

$$\boldsymbol{\nu}_p = \mathbf{M}_t^{-1}(\boldsymbol{\nu} - \mathbf{c}_t) \tag{E2.33}$$

L'insertion de (E2.33) dans (E2.26) nous donne :

$$\ddot{\mathbf{q}}_p = \boldsymbol{\nu}_p \tag{E2.34}$$

Il est alors possible d'écrire pour ce système une loi de commande à action PID (Proportionnelle-Intégrale-Dérivée) telle que :

$$\boldsymbol{\nu}_p = \ddot{\mathbf{q}}_p^d - \mathbf{K}_p\mathbf{e}_p - \mathbf{K}_d\dot{\mathbf{e}}_p - \mathbf{K}_i \int \mathbf{e}_p \tag{E2.35}$$

avec $\mathbf{K}_p$, $\mathbf{K}_d$ et $\mathbf{K}_i$ les matrices respectives de gains pour l'action proportionnelle, intégrale et dérivée. Ces gains sont choisis pour assurer la convergence du double intégrateur (E2.34). Un correcteur PD suffit à assurer la convergence. Cependant, en l'absence d'un terme intégral une erreur statique apparaît du fait de l'erreur de modélisation. Le terme intégral permet de compenser cette erreur statique. La commande auxilaire $\boldsymbol{\nu}$ est alors obtenue en combinant (E2.35) et (E2.33), donnant ainsi :

$$\boldsymbol{\nu} = \mathbf{M}_t \left(\ddot{\mathbf{q}}_p^d - \mathbf{K}_p\mathbf{e}_p - \mathbf{K}_d\dot{\mathbf{e}}_p - \mathbf{K}_i \int \mathbf{e}_p \right) + \mathbf{c}_t \tag{E2.36}$$

En introduisant l'équation (E2.29), nous obtenons :

$$\boldsymbol{\nu}_J = (\mathbf{J}_p^T)^+ \left(\mathbf{M}_t \left(\ddot{\mathbf{q}}_p^d - \mathbf{K}_p\mathbf{e}_p - \mathbf{K}_d\dot{\mathbf{e}}_p - \mathbf{K}_i \int \mathbf{e}_p \right) + \mathbf{c}_t \right) \tag{E2.37}$$

Il est ensuite possible d'obtenir, pour chaque drone $\mathcal{D}_i$, $f_{z,i}$ la poussée, ainsi que les coordonnées d'orientation φ_i^d et θ_i^d, à partir de l'ensemble d'équations (E2.32).

La loi de commande (E2.36) est équivalente à une commande par couples calculés connue pour les robots manipulateurs [Khalil et Dombre, 2002]. Le choix des gains du PID (E2.35), permettant la convergence du double-intégrateur (E2.34), assure également la convergence théorique du système (E2.26) en boucle fermée. En pratique, il existe une différence entre le modèle dynamique théorique et le modèle dynamique réel du robot. La stabilité de la commande reste cependant assurée sous les conditions décrites dans [Samson, 1987], notamment par le choix des gains du PID.

2.4.3 Commande de l'orientation des drones

Considérons le modèle dynamique de l'orientation des drones (E2.22) :

$$\mathbf{M}_a\ddot{\mathbf{q}}_a + \mathbf{c}_a = \mathbf{m}$$

Nous désirons faire suivre une trajectoire désirée $\mathbf{q}_a^d$ aux coordonnées d'orientation des drones. L'erreur de suivi est définie par : $\mathbf{e}_a = \mathbf{q}_a - \mathbf{q}_a^d$. En effectuant une inversion dynamique, il est possible, là aussi, de ramener le système à un double intégrateur. Pour cela, définissons la commande auxiliaire $\boldsymbol{\nu}_a$ tel que :

$$\boldsymbol{\nu}_a = \mathbf{M}_a^{-1}(\mathbf{m} - \mathbf{c}_a) \tag{E2.38}$$

En insérant (E2.38) dans (E2.22), nous obtenons le double intégrateur suivant :

$$\ddot{\mathbf{q}}_a = \boldsymbol{\nu}_a \tag{E2.39}$$

Pour la conception d'une loi de commande en cascade (cf partie 2.4.4), il est crucial d'obtenir une convergence rapide et une bonne stabilité de la commande de l'orientation des drones. La commande par mode glissant présente ces deux avantages et a donc été sélectionné pour la conception de la loi de commande de l'orientation des drones. Cette commande sera également plus robuste face aux erreurs de modèle dynamique.

Définissons le vecteur des variables glissantes $\boldsymbol{\sigma}$ par :

$$\boldsymbol{\sigma} = \dot{\mathbf{e}}_a + \boldsymbol{\Lambda}\mathbf{e}_a + \boldsymbol{\Lambda}_i \int \mathbf{e}_a \tag{E2.40}$$

avec $\boldsymbol{\Lambda}$ et $\boldsymbol{\Lambda}_i$ deux matrices diagonales à coefficients strictement positifs. La commande par mode glissant est définie par :

$$\boldsymbol{\nu}_a = \ddot{\mathbf{q}}_a^d - \boldsymbol{\Lambda}\dot{\mathbf{e}}_a - \boldsymbol{\Lambda}_i\mathbf{e}_a - \mathbf{K}_s \text{signe}(\boldsymbol{\sigma}) \tag{E2.41}$$

avec $\mathbf{K}_s$ une matrice de gains diagonale à coefficients strictement positifs. La fonction signe(.) d'un vecteur $\boldsymbol{\sigma}$ est définie par un vecteur $\mathbf{s}$ de même dimension donné par :

$$s_k = \begin{cases} 1 & \text{si} \quad \sigma_k > 0 \\ 0 & \text{si} \quad \sigma_k = 0 \\ -1 & \text{si} \quad \sigma_k < 0 \end{cases} \tag{E2.42}$$

avec s_k, respectivement σ_k, le k-ième élément du vecteur $\mathbf{s}$, respectivement σ.

La stabilité de la commande par mode glissant se prouve à l'aide de la théorie de Lyapunov [Khalil, 2002]. Considérons la fonction de Lyapunov suivante :

$$V = \boldsymbol{\sigma}^T\boldsymbol{\sigma} \tag{E2.43}$$

La dérivée de cette fonction s'écrit alors :

$$\dot{V} = -2\boldsymbol{\sigma}^T\dot{\boldsymbol{\sigma}} \tag{E2.44}$$

En introduisant (E2.40) dans (E2.44), nous obtenons :

$$\dot{V} = -2\boldsymbol{\sigma}^T(\ddot{\mathbf{q}}_a - \ddot{\mathbf{q}}_a^d + \boldsymbol{\Lambda}\dot{\mathbf{e}}_a + \boldsymbol{\Lambda}_i\mathbf{e}_a) \tag{E2.45}$$

Et en introduisant ensuite (E2.39) et (E2.41) dans (E2.45), nous avons :

$$\dot{V} = -2\boldsymbol{\sigma}^T\mathbf{K}_s\text{signe}(\boldsymbol{\sigma}) \tag{E2.46}$$

$\dot{V}$ est alors strictement négative pour $\boldsymbol{\sigma}$ non nul. Ainsi, la stabilité et la convergence de $\boldsymbol{\sigma}$ vers zéro sont assurées. Cette stabilité est également assurée en présence de perturbations bornées par l'augmentation des gains de la matrice $\mathbf{K}_s$. La commande par mode glissant assure la convergence vers zéro de $\boldsymbol{\sigma}$ en temps fini. Une fois la convergence de la variable glissante assurée, l'évolution de l'erreur s'écrit à partir de (E2.40) :

$$\dot{\mathbf{e}}_a + \boldsymbol{\Lambda}\mathbf{e}_a + \boldsymbol{\Lambda}_i\int\mathbf{e}_a = 0 \tag{E2.47}$$

La convergence des variables d'orientation $\mathbf{q}_a$ est alors garantie car $\boldsymbol{\Lambda}$ et$\boldsymbol{\Lambda}_i$ sont des matrices diagonales à coefficients strictement positifs.

L'un des inconvénients du mode glissant est qu'il génère un phénomène de broutement, souvent désigné par le terme anglais "chattering". Pour limiter ce phénomène, la fonction signe(.) est remplacée par la fonction saturation sat(.). La fonction sat(.) d'un vecteur $\boldsymbol{\sigma}$ est définie par un vecteur $\mathbf{s}$ de même dimension donné par :

$$\mathbf{s}_k = \begin{cases} 1 & \text{si} \quad \sigma_k > \epsilon_k \\ \sigma_k/\epsilon_k & \text{si} \quad \sigma_k| \leq \epsilon_k \\ -1 & \text{si} \quad \sigma_k < -\epsilon_k \end{cases} \tag{E2.48}$$

Pour chaque variable glissante σ_k, ϵ_k défini une frontière autour de la surface glissante. La stabilité au delà de cette frontière est garantie et le système reste à l'intérieur cette frontière une fois celle-ci atteinte.

La commande finale de la dynamique d'orientation des drones est obtenue en combinant la loi de commande par mode glissant (E2.41) avec l'inversion dynamique (E2.38) :

$$\mathbf{m} = \mathbf{M}_a\left(\ddot{\mathbf{q}}_a^d - \boldsymbol{\Lambda}\dot{\mathbf{e}}_a - \boldsymbol{\Lambda}_i\mathbf{e}_a - \mathbf{K}_s\text{sat}(\boldsymbol{\sigma})\right) + \mathbf{c}_a \tag{E2.49}$$

2.4.4 Commande du robot volant complet

Nous avons établi dans la partie 2.4.2 une commande des coordonnées de l'architecture passive. Dans la partie 2.4.3, la commande de l'orientation des drones a été obtenue. La commande complète du robot consiste en une mise en cascade des deux commandes précédemment établies (voir Fig. 2.8). Pour cela, la trajectoire désirée de l'architecture passive $\mathbf{q}_p^d$ est définie. La commande de l'architecture passive ((E2.37) et (E2.32)) nous donne la pous-

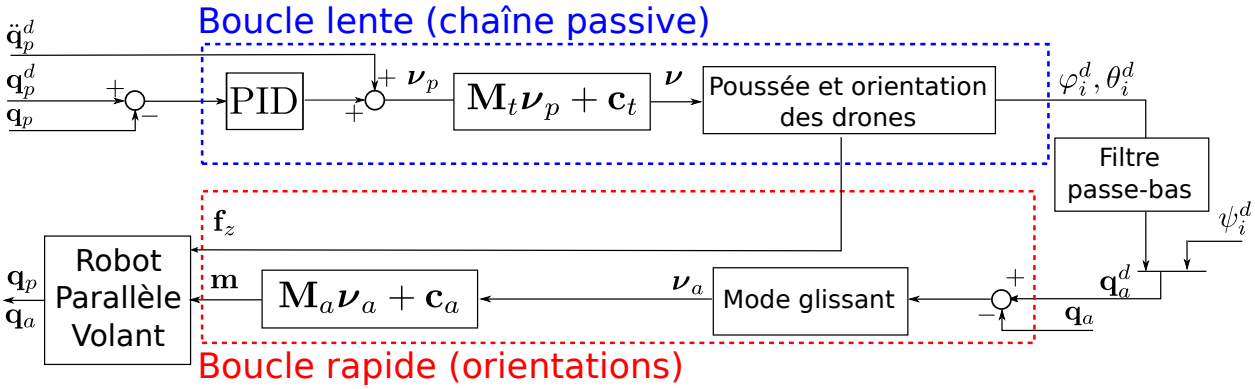


FIGURE 2.8 – Schéma de la loi de commande pour le robot parallèle volant.

sée $f_{z,i}$ et les coordonnées d'orientation φ_i, θ_i $(i = 1..n)$ pour chaque drone. Les poussées désirées sont regroupées dans le vecteur $\mathbf{f}$.

Ensuite, les coordonnées d'orientation à fournir pour la commande de l'architecture passive définissent l'entrée pour la commande d'orientation des drones. Seules les coordonnées φ_i et θ_i sont imposées pour chaque drone, laissant la coordonnée ψ_i libre. La trajectoire de la coordonnée ψ_i pourra en pratique servir à limiter les auto-collisions du robot lors de son évolution. Les trajectoires φ_i^d, θ_i^d et ψ_i^d des drones sont regroupées dans un vecteur de trajectoire désirée $\mathbf{q}_a^d$. La loi de commande (E2.49) nous donne alors le vecteur $\mathbf{m}$ contenant l'ensemble des couples à générer par les drones.

La convergence de l'erreur sur les variables d'orientation des drones $\mathbf{q}_a$ implique alors la convergence de l'erreur des coordonnées de l'architecture passive $\mathbf{q}_p$. Notons cependant certaines limites décrites ci-après. La régulation en cascade impose une séparation des échelles de temps entre la boucle interne rapide (orientation des drones) et la boucle externe plus lente (architecture passive). Ainsi, la fréquence de coupure de la commande de l'architecture passive devra être assez faible pour éviter de déstabiliser la commande en cascade obtenue. De plus, le calcul de la commande d'orientation nécessite de dériver deux fois l'orientation désirée $\mathbf{q}_a^d$. Les bruits sur la commande de l'architecture passive sont alors fortement amplifiés. Il est nécessaire de prévoir un filtre entre les deux étages pour limiter cet effet qui déstabiliserait le système.

2.4.5 Commande adaptative de l'architecture passive

Pour l'architecture passive, la stabilité et la performance de la commande sont soumises à la qualité du modèle dynamique établi. L'identification des paramètres dynamiques [Hollerbach et al., 2008] peut être utilisée pour affiner le modèle dynamique de l'architecture. Les paramètres dynamiques (masse, inertie) ne varient pas et peuvent être établis préalablement. Cependant, il existe un paramètre dynamique dont la valeur peut varier suivant les circonstances et qui impacte le modèle dynamique de l'architecture passive. Il s'agit du coefficient

de poussée du drone. En effet, dans les conditions d'expérimentation, celui-ci peut varier sous l'influence du niveau de charge de la batterie de l'appareil ou encore des effets de sol. C'est pourquoi une modification de la commande de l'architecture passive est proposée dans cette partie pour tenir compte de ce phénomène. Cette commande adaptative est inspirée d'une commande existante [Slotine et Weiping, 1988] transposée au cas du robot parallèle volant.

Considérons le modèle dynamique de l'architecture passive (E2.24) :

$$\mathbf{M}_t\ddot{\mathbf{q}}_p + \mathbf{c}_t = \mathbf{J}_p^T\mathbf{R}_r\mathbf{f}_z$$

En séparant la poussée de chaque drone dans le modèle dynamique, on obtient :

$$\mathbf{M}_t\ddot{\mathbf{q}}_p + \mathbf{c}_t = \sum_{i=1}^{n}\mathbf{J}_{v,i}^T\mathbf{R}_{r,i}f_{z,i} \tag{E2.50}$$

avec

- $\mathbf{R}_{r,i}$ la troisième colonne de la matrice $\mathbf{R}_i$ de rotation du drone $\mathcal{D}_i$,
- $f_{z,i}$ la poussée du drone $\mathcal{D}_i$.

La poussée $f_{z,i}$ du drone i est proportionnelle au coefficient de poussée $k_{t,i}$ par l'équation (E2.23). Supposons le coefficient réel de poussée $k_{t,i}$ différent de son estimation $\hat{k}_{t,i}$ et posons α_i tel que $\alpha_i = k_{t,i}/\hat{k}_{t,i}$. Alors le modèle dynamique (E2.50) s'écrit :

$$\mathbf{M}_t\ddot{\mathbf{q}}_p + \mathbf{c}_t = \sum_{i=1}^{n}\alpha_i\mathbf{J}_{v,i}^T\mathbf{R}_{r,i}\bar{f}_{z,i} \tag{E2.51}$$

avec $\bar{f}_{z,i}$ la poussée calculée à l'aide de l'estimation du coefficient de poussée $\hat{k}_{t,i}$:
$\bar{f}_{z,i} = \hat{k}_{t,i}\sum_{k=1}^{4}\omega_{i,j}$. Comme pour la commande non adaptative, le modèle dynamique sert à définir une commande auxiliaire ramenant le système à un double intégrateur :

$$\boldsymbol{\nu}_p = \mathbf{M}_t^{-1}\left(\sum_{i=1}^{n}\hat{\alpha}_i\mathbf{J}_{v,i}^T\mathbf{R}_{r,i}\bar{f}_{z,i} - \mathbf{c}_t\right) \tag{E2.52}$$

Dans la commande, une estimation $\hat{\alpha}_i$ de chaque coefficient α_i est utilisée, en remplacement de la valeur réelle qui est inconnue. Cette estimation sera effectuée dans la loi de commande. Nous définissons la variable glissante suivante :

$$\boldsymbol{\sigma}_p = \dot{\mathbf{e}}_p + \lambda_p\mathbf{e}_p + \lambda_i\int\mathbf{e}_p \tag{E2.53}$$

avec $\lambda_p > 0$ et $\lambda_i > 0$. La surface de glissement ainsi définie assure la convergence de l'erreur $\mathbf{e}_p$ vers zéro lorsque $\boldsymbol{\sigma}_p$ converge vers zéro. La loi de commande suivante pour la commande

auxiliaire $\boldsymbol{\nu}_p$ est alors définie par :

$$\boldsymbol{\nu}_p = \ddot{\mathbf{q}}_p^d - \lambda_p \dot{\mathbf{e}}_p - \lambda_i \mathbf{e}_p - k_\sigma \boldsymbol{\sigma}_p \tag{E2.54}$$

avec $k_\sigma > 0$. Afin de prouver la convergence de $\boldsymbol{\sigma}_p$ vers zéro, la théorie de stabilité de Lyapunov [Khalil, 2002] est à nouveau utilisée. La fonction de Lyapunov suivante est définie par :

$$V = \frac{1}{2} \boldsymbol{\sigma}_p^T \boldsymbol{\sigma}_p \tag{E2.55}$$

La dérivée de cette fonction est alors :

$$\dot{V} = \boldsymbol{\sigma}_p^T \dot{\boldsymbol{\sigma}}_p \tag{E2.56}$$

En introduisant l'équation (E2.53), nous obtenons :

$$\dot{V} = \boldsymbol{\sigma}_p^T (\ddot{\mathbf{q}}_p - \ddot{\mathbf{q}}_p^d + \lambda_p \dot{\mathbf{e}}_p + \lambda_i \mathbf{e}_p) \tag{E2.57}$$

Et à l'aide du modèle dynamique (E2.51), nous avons :

$$\dot{V} = \boldsymbol{\sigma}_p^T \left(\mathbf{M}_t^{-1} \left(\sum_{i=1}^{n} \alpha_i \mathbf{J}_{v,i}^T \mathbf{R}_{r,i} \bar{f}_{z,i} - \mathbf{c}_t \right) - \ddot{\mathbf{q}}_p^d + \lambda_p \dot{\mathbf{e}}_p + \lambda_i \mathbf{e}_p \right) \tag{E2.58}$$

Définissons alors $\tilde{\alpha}_i = \alpha_i - \hat{\alpha}$ la différence entre le coefficient estimé et le coefficient réel. En introduisant (E2.52) et (E2.54) dans (E2.58), nous obtenons :

$$\dot{V} = -k_\sigma \boldsymbol{\sigma}_p^T \boldsymbol{\sigma}_p + \boldsymbol{\sigma}_p^T \mathbf{M}_t^{-1} \left(\sum_{i=1}^{n} \tilde{\alpha}_i \mathbf{J}_{v,i}^T \mathbf{R}_{r,i} \bar{f}_{z,i} \right) \tag{E2.59}$$

Cette fonction n'est pas strictement négative. La différence entre le modèle réel et son estimation, représentée par $\tilde{\alpha}_i \neq 0$, peut engendrer une instabilité du système. Pour corriger ceci, rajoutons quelques termes à la fonction de Lyapunov (E2.55) :

$$V = \frac{1}{2} \boldsymbol{\sigma}_p^T \boldsymbol{\sigma}_p + \frac{1}{2\gamma} \sum_{i=1}^{n} \tilde{\alpha}_i \tag{E2.60}$$

avec γ un coefficient strictement positif. La dérivée de la fonction V est alors (en omettant les étapes de calcul similaires au cas précédent) :

$$\dot{V} = -k_\sigma \boldsymbol{\sigma}_p^T \boldsymbol{\sigma}_p + \boldsymbol{\sigma}_p^T \mathbf{M}_t^{-1} \left(\sum_{i=1}^{n} \tilde{\alpha}_i \mathbf{J}_{v,i}^T \mathbf{R}_{r,i} \bar{f}_{z,i} \right) + \frac{1}{\gamma} \sum_{i=1}^{n} \tilde{\alpha}_i \dot{\tilde{\alpha}}_i \tag{E2.61}$$

Pour obtenir la stabilité, nous cherchons à éliminer les deux derniers termes de $\dot{V}$. Pour cela,

il est suffisant d'avoir, pour $(i = 1..n)$:

$$\dot{\hat{\alpha}}_i = -\gamma \boldsymbol{\sigma}_p^T \mathbf{M}_t^{-1} \mathbf{J}_{v,i}^T \mathbf{R}_{r,i} \bar{f}_{z,i} \tag{E2.62}$$

Dans ce cas, l'équation (E2.61) devient :

$$\dot{V} = -k_\sigma \boldsymbol{\sigma}_p^T \boldsymbol{\sigma}_p \tag{E2.63}$$

$\dot{V}$ est strictement négative pour $\boldsymbol{\sigma}_p \neq 0$, ce qui prouve la stabilité. Cependant, il n'est pas possible d'agir sur la variation des termes $\tilde{\alpha}_i$. Nous allons donc ajouter l'hypothèse de faible variation des coefficients α_i devant la dynamique du système. Ainsi, il est possible écrire :

$$\dot{\tilde{\alpha}}_i = \dot{\alpha}_i - \dot{\hat{\alpha}}_i \simeq -\dot{\hat{\alpha}}_i \tag{E2.64}$$

Ainsi, en introduisant (E2.62) dans (E2.64), nous obtenons la loi d'adaptation pour l'estimation des coefficients α_i donnée par :

$$\dot{\hat{\alpha}}_i = \gamma \boldsymbol{\sigma}_p^T \mathbf{M}_t^{-1} \mathbf{J}_{v,i}^T \mathbf{R}_{r,i} \bar{f}_{z,i} \tag{E2.65}$$

L'estimation des coefficients $\hat{\alpha}_i$ est initialisée à 1 (pas d'erreur de modèle) lors de l'initialisation de la loi de commande, puis adaptée à chaque pas de temps selon la formule énoncée ci-dessus.

Le calcul de la poussée des angles désirés pour chaque drone à partir de la loi de commande pour la variable auxiliaire $\boldsymbol{\nu}_p$ (E2.54) est similaire à la commande sans adaptation (équations (E2.37) et (E2.32)). La seule différence étant la modification de la matrice $\mathbf{J}_p$ par la matrice $\mathbf{J}_{p,\alpha}$ définie par :

$$\mathbf{J}_{p,\alpha} = \begin{bmatrix} \hat{\alpha}_1 \mathbf{J}_{v,1} \\ \hat{\alpha}_2 \mathbf{J}_{v,2} \\ \vdots \\ \hat{\alpha}_n \mathbf{J}_{v,n} \end{bmatrix} \tag{E2.66}$$

Cette modification est la conséquence de la définition de la loi auxiliaire $\boldsymbol{\nu}_p$ (E2.52) tenant compte de l'estimation des coefficients α_i.

2.5 Conclusion

Dans ce chapitre, un nouveau concept de robot volant, appelé le robot parallèle volant, a été présenté. Ce robot est constitué d'une architecture passive articulée associée à plusieurs drones (dans notre cas il s'agit de quadricoptères). Le modèle dynamique générique de ce robot a été établi. Ce modèle montre des propriétés de découplage entre la dynamique des coordonnées généralisées de l'architecture passive et la dynamique de la rotation des

drones. Ce découplage est exploité pour établir une loi de commande en cascade qui assure la stabilité du système et permet le suivi d'une trajectoire imposée sur les coordonnées de la structure passive. Une loi d'adaptation a également été proposée afin de prendre en compte les variations du coefficient de poussée des drones par les effets de sol ou par la variation de la charge de la batterie.

Au chapitre suivant, nous allons détailler deux exemples de robots basés sur ce concept. Une simulation sera effectuée pour chacun de ces robots afin de confirmer la capacité de la loi de commande établie à stabiliser le robot et à suivre une trajectoire demandée.

Chapitre 3

Étude de deux architectures de robots parallèles volants

Sommaire

3.1 Robot parallèle volant à trois drones 66
3.1.1 Paramétrisation 66
3.1.2 Mobilités de la plate-forme et analyse des singularités 68
3.1.3 Simulation du robot volant à trois drones 74
3.2 Robot parallèle volant à deux drones 82
3.2.1 Présentation des paramètres cinématiques 83
3.2.2 Modèle géométrique direct 86
3.2.3 Modèle cinématique et singularités 88
3.2.4 Modèle dynamique 91
3.2.5 Loi de commande 101
3.2.6 Gestion de la commande latérale et ajout d'un pivot 104
3.2.7 Simulation 108
3.3 Conclusion 114

Ce chapitre présente l'étude de deux robots parallèles volants. Pendant le déroulement de cette thèse, un prototype de robot parallèle volant à deux drones, inspiré du mécanisme Biglide, a été d'abord conçu et fabriqué. Nous verrons au cours de ce chapitre et dans le chapitre 4 que cette configuration à deux drones présente de nombreuses particularités ayant induit des adaptations du modèle général décrit au chapitre 2. Pour une lecture plus facile des cas d'application, nous présenterons donc dans un premier temps une architecture à trois drones, pour lequel le modèle dynamique et la commande développés au chapitre 2 s'appliquent directement. Nous effectuerons pour ce robot une étude des mobilités de la plate-forme et des singularités afin de déterminer un espace de travail utile. Nous simulerons ensuite, dans cet espace de travail, un suivi de trajectoire effectuée par le robot afin de montrer les performances de suivi et de stabilité de la loi de commande développée. Nous présenterons, dans une seconde partie, l'architecture à deux drones du premier prototype de robot parallèle volant. Nous développerons les modifications de la loi de commande mises en œuvre pour ce prototype. Une simulation de suivi de trajectoire a également été effectuée pour ce robot.

3.1 Robot parallèle volant à trois drones

Dans cette partie, nous présenterons un robot volant composé de trois quadricoptères et d'une architecture passive à 9 degrés de liberté (voir Fig. 3.1). Nous décrirons ce robot volant et étudierons les singularités de son architecture passive. Pour ce robot, le modèle dynamique et la commande ne seront pas développés car il s'agit de l'application directe de la théorie présentée au chapitre 2. Cette partie se terminera par la présentation des résultats de simulation obtenus pour ce robot.

3.1.1 Paramétrisation

Le robot considéré dans cette partie est composé d'une plate-forme mobile connectée à trois quadricoptères au moyen de trois jambes identiques. Chaque jambe est connectée à la plate-forme par une articulation pivot et à un drone par une rotule. La figure 3.2 illustre l'application de la paramétrisation développée dans la partie 2.2.2.

Le repère $\mathcal{F}_p(P, \mathbf{x}_p, \mathbf{y}_p, \mathbf{z}_p)$ attaché à la plate-forme est défini ci-après. Son origine P est le barycentre de la plate-forme. L'axe $\mathbf{z}_p$ est normal au plan supérieur de la plate-forme. On écrit ℓ_{AB} la distance entre deux points A et B. Trois points R_i $(i = 1, 2, 3)$ sont définis par leurs coordonnées dans le repère $\mathcal{F}_p$:

$$ {}^p\overrightarrow{PR_i} = [\ell_{PR_i} \cos\gamma_i \quad \ell_{PR_i} \sin\gamma_i \quad 0]^T \tag{E3.1} $$

avec $\gamma_i = \{0, 2\pi/3, -2\pi/3\}$ rad. Dans le cas de notre robot, $\ell_{PR_1} = \ell_{PR_2} = \ell_{PR_3}$. Cette distance sera notée ℓ_p.

L'articulation pivot liant la plate-forme à la jambe $\mathcal{B}_i$ est située en R_i. Son axe est orienté suivant le vecteur ${}^p\mathbf{a}_i = [-\sin\gamma_i \ \cos\gamma_i \ 0]^T$ exprimé dans le repère $\mathcal{F}_p$. Chaque jambe

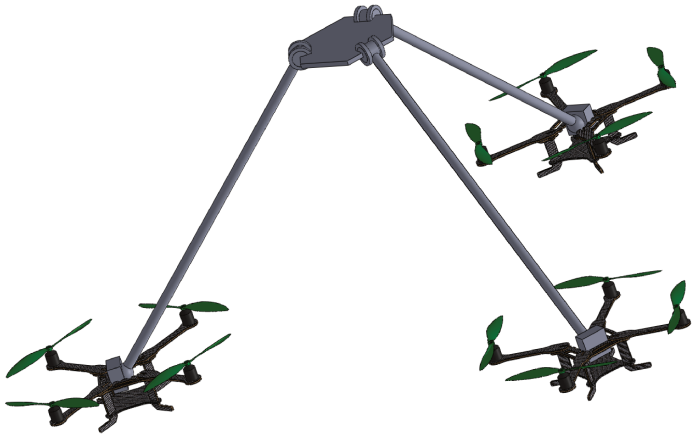


FIGURE 3.1 – Vue CAO du robot parallèle volant à trois drones.

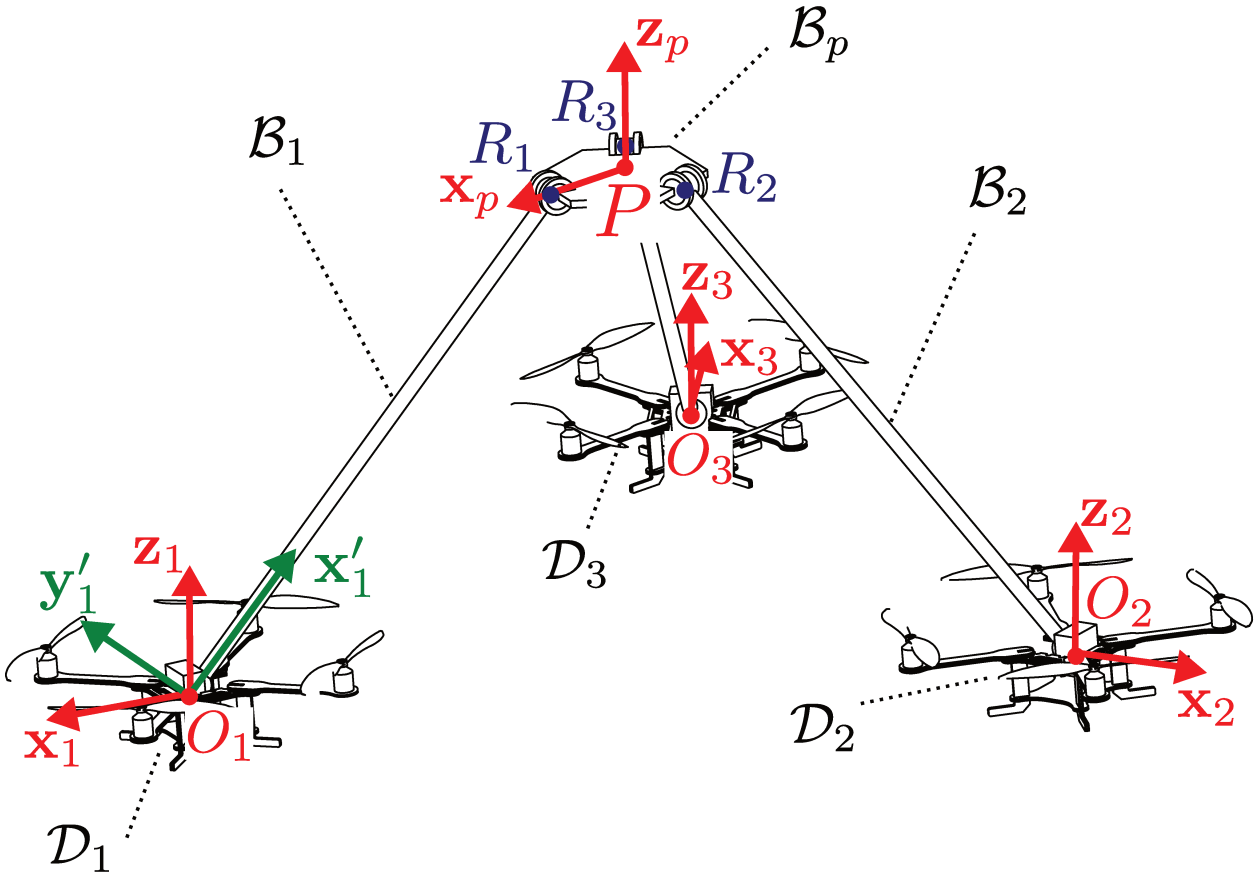


FIGURE 3.2 – Paramétrisation du robot parallèle volant à trois drones.

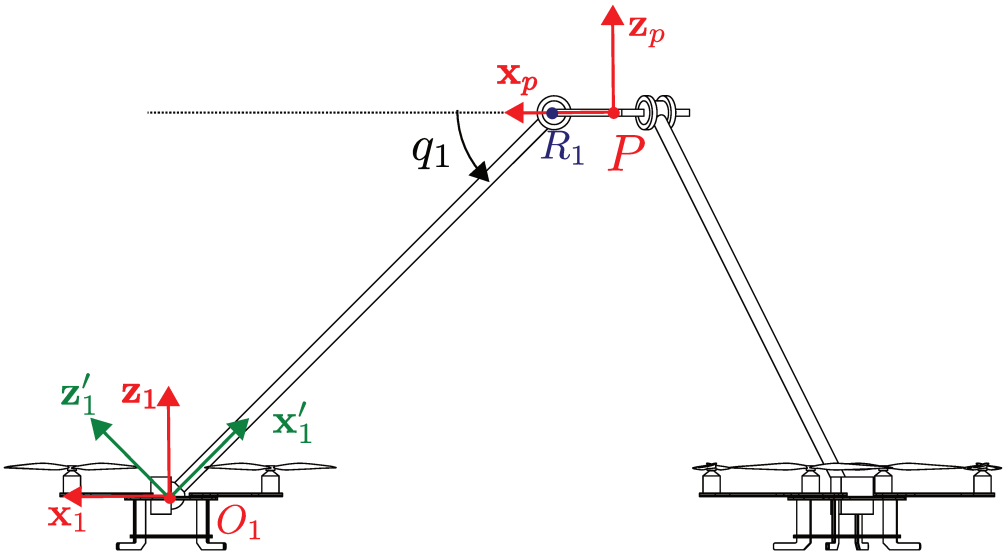


FIGURE 3.3 – Vue en coupe du robot parallèle volant à trois drones.

$\mathcal{B}_i$ est reliée au drone $\mathcal{D}_i$ par une articulation sphérique située en O_i $(i = 1, 2, 3)$ (voir aussi Fig. 3.3). Comme prévu pour le modèle général, O_i est également le centre de masse du drone $\mathcal{D}_i$. La longueur de la jambe i est définie par la distance $\ell_{R_iO_i}$. Le vecteur $\mathbf{q}_p$ des coordonnées généralisées de l'architecture passive est donné par :

- $x_p, y_p, z_p, \varphi_p, \theta_p, \psi_p$ les paramètres caractérisant la position et l'orientation de la plate-forme dans le repère $\mathcal{F}_o$,
- q_i $(i = 1, 2, 3)$ l'angle relatif entre le plan défini par $(\mathbf{x}_p, \mathbf{y}_p)$ et la jambe i (voir Fig. 3.3).

Le vecteur $\mathbf{q}_a$ est composé des coordonnées d'orientation des drones,
i.e. $\mathbf{q}_a = [\varphi_1\ \theta_1\ \psi_1\ \varphi_2\ \theta_2\ \psi_2\varphi_3\ \theta_3\ \psi_3]^T$.

En partie 2.2.2, l'utilisation de repères virtuels permettant la définition de l'architecture passive selon la convention DHM a été proposée. Le tableau (T3.1) décrit le robot selon cette convention. Elle permet le calcul de $\mathbf{M}_p$ et $\mathbf{c}_p$ (E2.10) de l'architecture passive par les algorithmes de Newton-Euler récursifs [Khalil, 2010].

Le robot dispose de 9 degrés de liberté correspondant à la position et l'orientation de la plate-forme dans l'espace et à la configuration de chacune des jambes.

3.1.2 Mobilités de la plate-forme et analyse des singularités

Dans cette partie, nous nous posons la question suivante : est-il possible de commander les degrés de liberté de la plate-forme avec cette architecture ? Pour répondre à cette question, considérons que le drone $\mathcal{D}_i$ peut produire une force au point O_i dans n'importe quelle direction. En réalité, les quadricoptères sont sous-actionnés mais nous avons traité cette problématique par la conception d'une loi de commande en cascade (voir partie 2.4.4). Ainsi l'étude des mobilités de la plate-forme est réduite à l'étude de l'architecture passive munie

i	ant$_i$	μ_i	σ_i	γ_i	b_i	α_i	d_i	θ_i	r_i
$\mathcal{V}_1$	$\mathcal{F}_o$	0	1	0	0	0	0	0	z
$\mathcal{V}_2$	$\mathcal{V}_1$	0	1	0	0	$-\frac{\pi}{2}$	0	$-\frac{\pi}{2}$	y
$\mathcal{V}_3$	$\mathcal{V}_2$	0	1	0	0	$-\frac{\pi}{2}$	0	0	x
$\mathcal{V}_4$	$\mathcal{V}_3$	0	0	0	0	0	0	φ	0
$\mathcal{V}_5$	$\mathcal{V}_4$	0	0	0	0	$\frac{\pi}{2}$	0	$\theta + \frac{\pi}{2}$	0
$\mathcal{B}_p$	$\mathcal{V}_5$	0	0	0	0	$\frac{\pi}{2}$	0	ψ	0
$\mathcal{B}_1$	$\mathcal{B}_p$	0	0	0	0	$\frac{\pi}{2}$	ℓ_p	q_1	0
$\mathcal{B}_2$	$\mathcal{B}_p$	0	0	$\frac{2\pi}{3}$	0	$\frac{\pi}{2}$	ℓ_p	q_2	0
$\mathcal{B}_3$	$\mathcal{B}_p$	0	0	$\frac{4\pi}{3}$	0	$\frac{\pi}{2}$	ℓ_p	q_3	0

TABLE 3.1 – Coordonnées DHKK de l'architecture passive du robot volant à trois drones.

des articulations sphériques aux extrémités de chaque jambe et actionnée par trois forces $\mathbf{f}_{p,i}$ appliquées à chacune de ces extrémités.

Mobilités de la plate-forme

La théorie des torseurs [Ball, 1900] [Hunt, 1978] est adaptée à l'étude des mobilités infinitésimales d'un robot parallèle [Kumar, 1992]. L'objectif de la théorie des torseurs est de trouver l'ensemble des efforts qui s'exercent sur la plate-forme à partir de la géométrie du mécanisme et du type de liaisons le composant. Elle permet de calculer un système de contraintes, représentées par des torseurs des actions mécaniques (ou torseurs d'effort), exercées par chaque jambe sur la plate-forme quand les actionnements sont bloqués. Le rang du système de contraintes global, obtenu par concaténation des contraintes de chaque jambe, définit les degrés de mobilité de la plate-forme. Si ce système est de rang 6, alors la plate-forme est entièrement contrainte. Sinon, au moins un degré de liberté de la plate-forme n'est pas contrôlé.

Pour chaque jambe i ($i = 1, 2, 3$), quatre torseurs cinématiques unitaires $\$_{ki}$ définissent les mouvements instantanés de la jambe, un pour le pivot ($k = 1$) et trois pour l'articulation sphérique ($k = 2, 3, 4$). Le système de contraintes de la jambe est composé des torseurs d'effort $\boldsymbol{\zeta}$ réciproques à l'ensemble de torseurs cinématiques $\$_{ki}$ [Kumar, 1992], i.e. $\boldsymbol{\zeta}^T \$_{ki} = 0$ pour tout k. Cela signifie que la puissance virtuelle développée par le torseur des actions mécaniques, représenté par $\boldsymbol{\zeta}$ selon la direction du mouvement donné par $\$_{ki}$, est nulle.

Considérons donc la jambe i. Pour simplifier l'expression des torseurs cinématiques de cette jambe, un nouveau repère est défini $\mathcal{F}'_i(O_i, \mathbf{x}'_i, \mathbf{y}'_i, \mathbf{z}'_i)$ tel que (voir aussi Fig. 3.1 et 3.3) :

- l'origine O_i est le centre de l'articulation sphérique,
- $\mathbf{x}'_i$ pointe vers R_i (le long de la jambe),
- $\mathbf{y}'_i$ est dans la direction de l'axe du pivot.

Le torseur cinématique unitaire définissant le mouvement du pivot situé en R_i dans le repère

$\mathcal{F}'_i$ au point O_i est donné par [Hunt, 1978] :

$$\$_{1i}^{i'} = \begin{bmatrix} 0 & 0 & l_{R_iO_i} & 0 & 1 & 0 \end{bmatrix}^T \tag{E3.2}$$

tandis que les trois torseurs cinématiques unitaires définissant les mouvements de l'articulation sphérique dans le repère $\mathcal{F}'_i$ au point O_i s'écrivent

$$\$_{2i}^{i'} = \begin{bmatrix} 0 & 0 & 0 & 1 & 0 & 0 \end{bmatrix}^T \tag{E3.3}$$

$$\$_{3i}^{i'} = \begin{bmatrix} 0 & 0 & 0 & 0 & 1 & 0 \end{bmatrix}^T \tag{E3.4}$$

$$\$_{4i}^{i'} = \begin{bmatrix} 0 & 0 & 0 & 0 & 0 & 1 \end{bmatrix}^T \tag{E3.5}$$

Pour chaque jambe i, seulement deux torseurs d'effort unitaires indépendants sont réciproques à l'ensemble des torseurs cinématiques $\$_{ki}$. Leur expression dans le repère $\mathcal{F}'_i$ au point O_i s'écrit

$$\boldsymbol{\zeta}_{1i}^{i'} = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 \end{bmatrix}^T \tag{E3.6}$$

$$\boldsymbol{\zeta}_{2i}^{i'} = \begin{bmatrix} 0 & 1 & 0 & 0 & 0 & 0 \end{bmatrix}^T \tag{E3.7}$$

$\boldsymbol{\zeta}_{1i}$ et $\boldsymbol{\zeta}_{2i}$ sont deux forces exercées sur la plate-forme par les jambes. Ces forces sont appliquées en O_i et dirigées selon $\mathbf{x}'_i$ et $\mathbf{y}'_i$. $\boldsymbol{\zeta}_{1i}$ et $\boldsymbol{\zeta}_{2i}$ peuvent être vues comme les deux composantes de la force $\mathbf{f}_{p,i}$, exercée par le drone $\mathcal{D}_i$ sur l'architecture passive, qui est transmise à la plate-forme. La troisième composante de la force $\mathbf{f}_{p,i}$ le long de $\mathbf{z}'_i$ ne provoque pas de mouvement de la plate-forme. Cette composante est un actionnement qui pourra être utilisé pour la commande de la coordonnée q_i de la jambe (voir partie 3.1.2).

Considérons maintenant l'ensemble des trois jambes. Le système global des contraintes est donné par $\boldsymbol{\Xi} = [{}^o\boldsymbol{\zeta}_{11}\ {}^o\boldsymbol{\zeta}_{21}\ {}^o\boldsymbol{\zeta}_{12}\ {}^o\boldsymbol{\zeta}_{22}\ {}^o\boldsymbol{\zeta}_{13}\ {}^o\boldsymbol{\zeta}_{23}]$, avec tous les torseurs exprimés dans le repère d'origine $\mathcal{F}_o$. Ce système est de dimension 6, ce qui rend la plate-forme entièrement contrôlable. La prochaine partie étudie les singularités cinématiques, i.e. les configurations pour lesquelles le système des contraintes globales est de rang strictement inférieur à 6.

Analyse des singularités de la plate-forme

Plusieurs outils sont utilisés pour définir les conditions de perte de rang du système $\boldsymbol{\Xi}$ parmi lesquels la géométrie de Grassmann [Merlet, 2006] et l'algèbre de Grassmann-Cayley [Ben-Horin et Shoham, 2006]. Dans notre cas particulier, l'algèbre de Grassmann-Cayley a déjà été utilisée dans [Ben-Horin et Shoham, 2006] pour prouver les conditions de singularité si le système des contraintes est composé d'un triplet de deux forces indépendantes $\boldsymbol{\zeta}_{1i}$ et $\boldsymbol{\zeta}_{2i}$ $(i = 1, 2, 3)$, chaque binôme $\boldsymbol{\zeta}_{1i}$ et $\boldsymbol{\zeta}_{2i}$ étant appliqué au même point O_i.

Définissons quatre plans

- $\mathcal{P}_i$ $(i = 1, 2, 3)$ est le plan passant respectivement par O_i et contenant les deux forces $\boldsymbol{\zeta}_{1i}$ et $\boldsymbol{\zeta}_{2i}$ appliquées au point O_i. C'est également le plan qui contient les vecteurs $\mathbf{x}'_i$

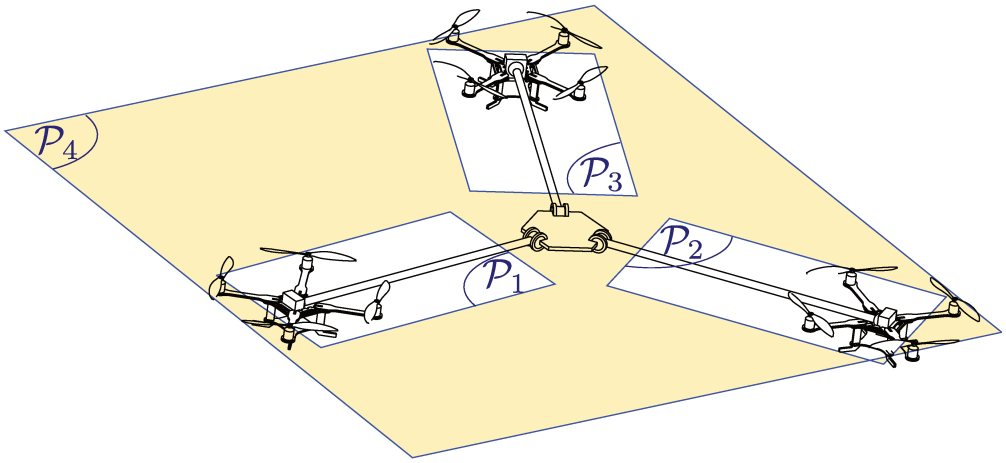


FIGURE 3.4 – Exemple d'une configuration singulière pour le robot parallèle volant à trois drones. $\mathcal{P}_1$, $\mathcal{P}_2$, $\mathcal{P}_3$ et $\mathcal{P}_4$ sont coplanaires.

et $\mathbf{y}'_i$;

- $\mathcal{P}_4$ est le plan contenant les points O_1, O_2 et O_3.

Les conditions de singularité apparaissent si les quatre plans $\mathcal{P}_i$ $(i = 1, 2, 3, 4)$ s'intersectent en un point au moins, au sens de coordonnées de Plücker [Ben-Horin et Shoham, 2006] (exemple Fig. 3.4). Cela inclut les cas suivants :

- les quatre plans s'intersectent en un point commun ;
- deux plans ou plus sont confondus ;
- trois plans ou plus s'intersectent le long d'une ligne commune ;
- l'un des plans dégénère en une ligne, par exemple O_1, O_2 et O_3 sont alignés ;
- l'un des plans dégénère en un point, par exemple O_1, O_2 et O_3 sont le même point.

Chaque plan $\mathcal{P}_i$ est défini par son équation :

$$a_i x + b_i y + c_i z + d_i = 0 \tag{E3.8}$$

Soit $\mathbf{D}$ la matrice composée des coefficients des quatre plans $\mathcal{P}_i$:

$$\mathbf{D} = \begin{bmatrix} a_1 & b_1 & c_1 & d_1 \\ a_2 & b_2 & c_2 & d_2 \\ a_3 & b_3 & c_3 & d_3 \\ a_4 & b_4 & c_4 & d_4 \end{bmatrix} \tag{E3.9}$$

La condition de singularité définie précédemment est équivalente à $\det(\mathbf{D}) = \mathbf{0}$. La position relative des quatre plans $\mathcal{P}_1$, $\mathcal{P}_2$, $\mathcal{P}_3$ et $\mathcal{P}_4$ n'est pas affectée par translation ou rotation dans l'espace et dépend donc seulement des degrés de mobilité internes de la plate-forme. Ainsi, les coordonnées q_1, q_2 et q_3 sont suffisantes pour étudier le lieu des singularités.

On cherche à montrer si le robot dispose d'un espace des configurations de ses jambes suffisant, libre de singularités, pour évoluer. Prenons le cas d'un robot ayant les dimensions $\ell_p = 0.1$ m et $\ell_{R_iO_i} = 0.8$ m identiques pour les trois jambes. Nous allons représenter numériquement les conditions de dégénérescence de la matrice $\mathbf{D}$ pour chaque q_i dans l'intervalle $[-\pi/2; \pi/2]$. Si celui-ci était plus grand, la possibilité d'auto-collision entre les jambes du robot serait élevée. La figure 3.5 montre la représentation 2D du lieu des singularités en fonction de q_2 et q_3 pour les valeurs suivantes de $q_1 = \pi/2$, $\pi/3$, $\pi/6$, 0, $-\pi/6$, $-\pi/3$ et $-\pi/2$. Le lieu des singularités est illustré pour les valeurs du déterminant de $\mathbf{D}$ proche de zéro ($|\det \mathbf{D}| < 5\text{e}{-4}$ dans notre analyse).

D'après ces graphiques, nous pouvons voir qu'en restreignant chaque coordonnée q_i à l'intervalle $[-\pi/2; 0]$, nous disposons d'un espace de travail libre de singularité. Cependant, un changement de configuration avec une plate-forme située sous les quadricoptères (tous les q_i positifs) nécessite de traverser une singularité. L'espace de travail effectif pourra être augmenté par l'usage d'une technique de traversée des singularités décrite au chapitre 5.

Nous allons maintenant étudier les singularités de l'ensemble de l'architecture passive, en tenant compte des jambes.

Analyse des singularités étendues aux jambes

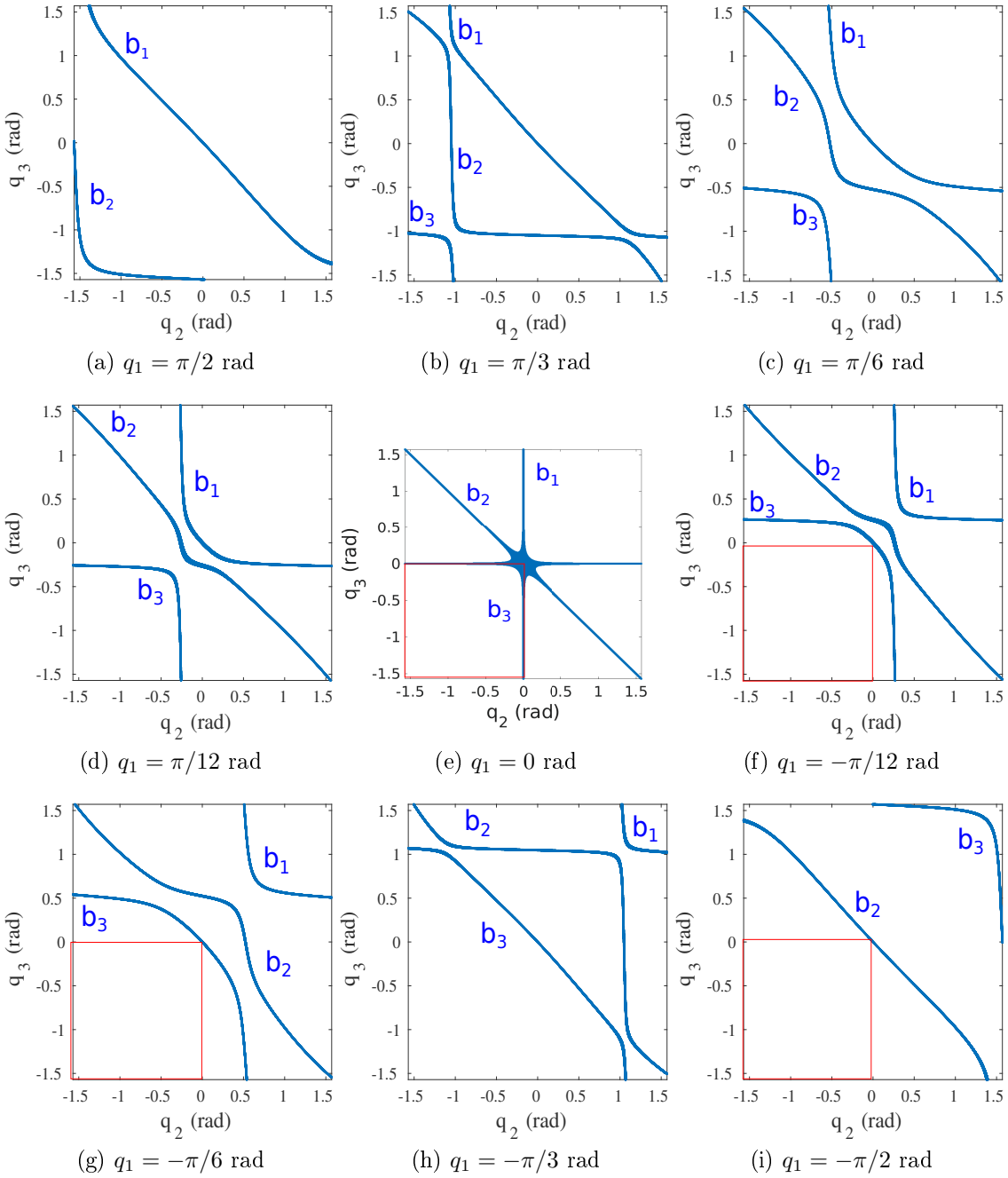
Nous avons vu dans la partie 3.1.2 que seule deux composantes de la force fournie par le drone i sont transmises à la plate-forme :

- une composante selon la direction x'_i,
- une composante selon la direction y'_i,

dans le repère associé à la jambe i. Ainsi, l'une des composantes reste disponible pour commander un degré de liberté supplémentaire. Cette composante est une force pure le long de $\mathbf{z}'_i$. Cette entrée supplémentaire va permettre la commande de la coordonnée q_i pour chaque jambe (voir Fig. 3.3). En partie 3.1.2, le lieu des singularités a été étudié pour la plate-forme uniquement. Dans cette partie, en exprimant la matrice jacobienne cinématique pour l'ensemble de l'architecture passive (plate-forme et jambes), nous pourrons prouver qu'il n'y a pas de singularités supplémentaires. Soit $\mathbf{q}_p = [x_p\ y_p\ z_p\ \varphi_p\ \theta_p\ \psi_p\ q_1\ q_2\ q_3]^T$ les coordonnées généralisées de l'architecture passive et ${}^{i'}\mathbf{v}_i$ $(i = 1, 2, 3)$ la vitesse du point O_i exprimée dans le repère $\mathcal{F}'_i$. On définit la matrice jacobienne cinématique inverse $\mathbf{J}'_{v,i}$ par :

$$ {}^{i'}\mathbf{v}_i = \mathbf{J}'_{v,i}\dot{\mathbf{q}}_p \tag{E3.10}$$

L'expression complète détaillée de $\mathbf{J}'_v$ n'est pas donnée ici mais son calcul a été effectué selon l'algorithme décrit en annexe A.1. Soit la matrice $\mathbf{J}'^T_v = [\mathbf{J}'^T_{v,1}\ \mathbf{J}'^T_{v,2}\ \mathbf{J}'^T_{v,3}]$. Afin d'étudier les singularités de cette matrice, il convient de réordonner ses lignes. Pour chaque $i = (1, 2, 3)$, définissons $v'_{i,x}$ (respectivement $v'_{i,y}$ et $v'_{i,z}$) la coordonnée de $\mathbf{v}_i^{i'}$ selon $\mathbf{x}'_i$ (respectivement $\mathbf{y}'_i$



(a) $q_1 = \pi/2$ rad (b) $q_1 = \pi/3$ rad (c) $q_1 = \pi/6$ rad

(d) $q_1 = \pi/12$ rad (e) $q_1 = 0$ rad (f) $q_1 = -\pi/12$ rad

(g) $q_1 = -\pi/6$ rad (h) $q_1 = -\pi/3$ rad (i) $q_1 = -\pi/2$ rad

FIGURE 3.5 – Robot parallèle volant à trois drones : lieu des singularités ($|\det \mathbf{A}| < 5\text{e}{-4}$) de la plate-forme avec $\ell_p = 0.1$ m et $\ell_{R_iO_i} = 0.8$ m. Le carré rouge représente une proposition d'espace de travail sans singularité.

et $\mathbf{z}'_i$). Le vecteur $\mathbf{v}_l$ est défini par :

$$\mathbf{v}_l = \begin{bmatrix} v'_{1,x} & v'_{2,x} & v'_{3,x} & v'_{1,y} & v'_{2,y} & v'_{3,y} & v'_{1,z} & v'_{2,z} & v'_{3,z} \end{bmatrix}^T \tag{E3.11}$$

Définissons $\mathbf{J}_l$ la matrice jacobienne $\mathbf{J}'_v$ réordonnée telle que :

$$\mathbf{v}_l = \mathbf{J}_l \dot{\mathbf{q}}_p \tag{E3.12}$$

Les trois dernières colonnes de la matrice $\mathbf{J}_v$ sont formées des composantes des torseurs cinématiques (E3.2), associés aux articulations pivots, contribuants à la vitesse des points O_i. Ainsi, la matrice $\mathbf{J}_l$ s'écrit sous la forme :

$$\mathbf{J}_l = \begin{bmatrix} \mathbf{J}_r & & \mathbf{0}_{6\times 3} & \\ & \ell_{R_1O_1} & & 0 \\ \star & 0 & \ell_{R_2O_2} & 0 \\ & 0 & 0 & \ell_{R_3O_3} \end{bmatrix} \tag{E3.13}$$

avec $\mathbf{J}_r$ reliant les vitesses $\mathbf{v}_i^{i'}$ selon $\mathbf{x}'_i$ et $\mathbf{y}'_i$ aux dérivées des coordonnées de la plate-forme, c'est à dire :

$$\begin{bmatrix} v'_{1,x} & v'_{2,x} & v'_{3,x} & v'_{1,y} & v'_{2,y} & v'_{3,y} \end{bmatrix}^T = \mathbf{J}_r \dot{\mathbf{q}}_e \tag{E3.14}$$

avec $\mathbf{q}_e = [x_p \; y_p \; z_p \; \varphi_p \; \theta_p \; \psi_p]^T$ les coordonnées de la plate-forme. $\mathbf{J}_r$ est ainsi la matrice jacobienne cinématique inverse de l'architecture passive dans le cas où les drones peuvent seulement produire des forces selon $\mathbf{x}'_i$ et $\mathbf{y}'_i$ ($i = 1, 2, 3$). Par l'équation (E3.13), le rang de la matrice $\mathbf{J}_l$ est égal au rang de la matrice $\mathbf{J}_r$ plus 3. Il est également connu [Merlet, 1989] que le rang de la matrice jacobienne cinématique $\mathbf{J}_r$ est égal au rang du système des contraintes $\boldsymbol{\Xi}$ appliquées à la plate-forme, étudié précédemment. L'architecture passive n'est pas dans une configuration singulière pour la plate-forme si et seulement si la matrice $\mathbf{J}_r$ est de rang plein.

Les deux matrices $\mathbf{J}_r$ et $\mathbf{J}_l$ ont les même conditions de dégénérescence. Ainsi le lieu de singularité pour l'ensemble de l'architecture passive est le même que celui de la plate-forme.

Dans la suite de cette partie, nous allons faire suivre à ce robot une trajectoire en simulation. Les configurations demandées au robot seront restreintes à l'espace de travail libre de configurations singulières déterminé ci-avant.

3.1.3 Simulation du robot volant à trois drones

Réalisation de la simulation

La loi de commande établie a été testée en simulation à l'aide des logiciels MATLAB-SIMULINK et ADAMS. La simulation dynamique des corps rigides est assurée par ADAMS tandis que la loi de commande a été mise en œuvre sous SIMULINK. La communication

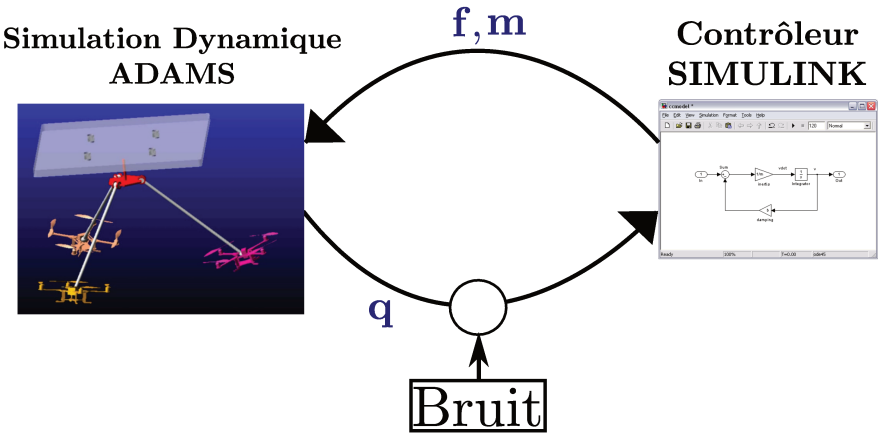


FIGURE 3.6 – Schéma de l'interaction entre les deux logiciels de simulation.

entre les deux logiciels est fixée à 1 kHz pour représenter le temps d'échantillonnage sur le robot réel (voir Fig. 3.6). A noter que la simulation de l'aérodynamique des hélices n'est pas gérée par ADAMS. Cette partie du modèle n'est donc pas simulée.

Afin de tester la stabilité de la loi de commande face aux bruits de mesure, un bruit uniforme a été ajouté pour chaque variable mesurée. L'intervalle de la distribution uniforme du bruit est $[-1e^{-2}; 1e^{-2}]$ rad pour les coordonnées de rotation, $[-1e^{-2}; 1e^{-2}]$ m pour les translations, $[-1e^{-2}; 1e^{-2}]$ m.s^{-1} pour les vitesses et $[-1e^{-2}; 1e^{-2}]$ rad.s^{-1} pour les vitesses angulaires. Ces bruits sont de natures différentes des bruits constatés en phase expérimentale (chapitre 4) mais suffisent à tester la stabilité de la loi de commande face au bruit de mesure dans le cadre de la simulation. Les entrées/sorties du modèle dynamique sont simplifiées et ne correspondent pas à des conditions d'expérimentation réelles. Ainsi, les entrées de la simulation dynamique sont représentées par le vecteur des forces de poussées $\mathbf{f}_z$ et le vecteur des moments $\mathbf{m}$ fournis par les drones. Nous considérons en sortie que l'ensemble des coordonnées du vecteur $\mathbf{q}$ est directement mesuré. Le modèle géométrique direct de ce robot n'est donc pas nécessaire pour cette application. Dans le cadre expérimental, il sera possible d'effectuer ces mesures par un système de capture de mouvements.

Paramètres physiques du robot

Les distances caractéristiques du robot sont

- la distance entre le centre de la plate-forme et les articulations pivot $\ell_p = 0.1$ m,
- la longueur des bras, choisie identique pour les trois bras : $\ell_{R_iO_i} = 0.8$ m.

Les paramètres dynamiques des corps sont donnés dans le tableau (T3.2) au centre de gravité des corps. Ces paramètres ont été choisis arbitrairement pour la simulation, la CAO du robot n'ayant pas été réalisée dans le cadre de cette thèse.

De plus, pour tester la robustesse de la loi de commande contre les erreurs de modèle, une perturbation de frottement aérodynamique est simulée sous ADAMS. Pour chaque drone,

Paramètre	Valeur
Masse de la plate-forme (kg)	1
Matrice d'inertie ($kg.m^2$) selon les axes du repère $\mathcal{F}_p$	$\begin{bmatrix} 0.03 & 0 & 0 \\ 0 & 0.03 & 0 \\ 0 & 0 & 0.05 \end{bmatrix}$
Masse d'une jambe (kg)	0.6
Matrice d'inertie ($kg.m^2$) selon les axes du repère $\mathcal{F}'_i$	$\begin{bmatrix} 0.00001 & 0 & 0 \\ 0 & 0.01 & 0 \\ 0 & 0 & 0.01 \end{bmatrix}$
Masse d'un drone (kg)	1.5
Matrice d'inertie ($kg.m^2$) selon les axes du repère $\mathcal{F}_i$	$\begin{bmatrix} 0.015 & 0 & 0 \\ 0 & 0.015 & 0 \\ 0 & 0 & 0.03 \end{bmatrix}$

TABLE 3.2 – Paramètres dynamiques du robot volant à trois drones.

cette perturbation est une force appliquée sur le drone et exprimée dans le repère local $\mathcal{F}_i$ par :

$$ {}^i\mathbf{f}_{a,i} = -\mathbf{A}\,{}^i\mathbf{v}_i \tag{E3.15}$$

avec $\mathbf{A}$ une matrice diagonale à coefficients de friction. Les coefficients sont choisis égaux dans toutes les directions (1 $kg.s^{-1}$).

Choix des gains et coefficients des filtres

Les gains sont sélectionnés afin de permettre une convergence rapide de l'erreur $\mathbf{e}_a$ sur les coordonnées d'orientation. Des gains assez faibles sont sélectionnés pour l'erreur de position des drones $\mathbf{e}_x$ afin de ne pas perturber la boucle rapide. Ces gains sont sélectionnés empiriquement. La définition d'une méthode de sélection des gains est à envisager dans l'extension des travaux de cette thèse. Les valeurs des gains pour cette simulation sont :

- pour la boucle lente (section 2.4.2) : les gains sont similaires sur toutes les coordonnées commandées. Ainsi, nous avons $\mathbf{K}_p$, $\mathbf{K}_d$ et $\mathbf{K}_i$ des matrices diagonales à coefficients respectifs 4, 4 et 1 ;
- pour la boucle rapide (section 2.4.3) : les coefficients diagonaux de Λ, Λ_i, $\mathbf{K}_s$ sont respectivement 15, 0 et 15. $\epsilon_k = 1$ est le même pour chaque variable commandée.

Comme indiqué dans la partie 2.4.4, surtout en présence de bruit important, il est nécessaire d'établir un filtre passe-bas entre la boucle lente et la boucle rapide. Sa fonction est de filtrer les coordonnées d'orientation désirées $\mathbf{q}_a^d$ issues du calcul de la boucle lente. Le filtre mis en œuvre est un filtre passe-bas de type Butterworth d'ordre 3 à fréquence de coupure 5 Hz.

Résultats de simulation

Le simulateur est utilisé pour le suivi d'une trajectoire qui se décompose en cinq parties

- de 0 à 5 secondes : stabilisation de la plate-forme,
- de 5 à 8 secondes : translation du robot selon l'axe $\mathbf{x}_o$,
- de 8 à 11 secondes : rotation combinée du robot selon les axes $\mathbf{z}_p$ et $\mathbf{x}_p$,
- de 11 à 14 secondes : changement d'orientation des jambes (degrés de mobilité internes de l'architecture passive),
- de 14 à 17 secondes : mouvement combiné sur plusieurs coordonnées.

Le suivi de la trajectoire de l'architecture passive est donné Fig. 3.7. La figure 3.8 montre les angles désirés et suivis pour chacun des trois drones. Les drones ne sont pas initialement positionnés de façon à assurer la stabilité du robot. Sur la figure 3.10, on constate la convergence vers les angles désirés pour chacun des drones. La convergence de la boucle externe de position (Fig. 3.9) n'est obtenue qu'après la convergence de la boucle interne. La nécessité de maintenir une séparation des échelles de temps entre les deux boucles implique une dynamique lente sur la convergence de la structure passive.

On constate la convergence de la commande en début de trajectoire sur les figures 3.9 et 3.10. Les résultats sont affichés filtrés pour une meilleure lecture des courbes. La loi de commande ne montre pas de difficulté pour suivre la trajectoire désirée. Une vidéo de la simulation est disponible au lien suivant
https://uncloud.univ-nantes.fr/index.php/s/Xna9LEBpzLogD7K

Nous allons maintenant étudier un robot parallèle volant à deux drones qui sera le prototype développé au LS2N dans le cadre de cette thèse.

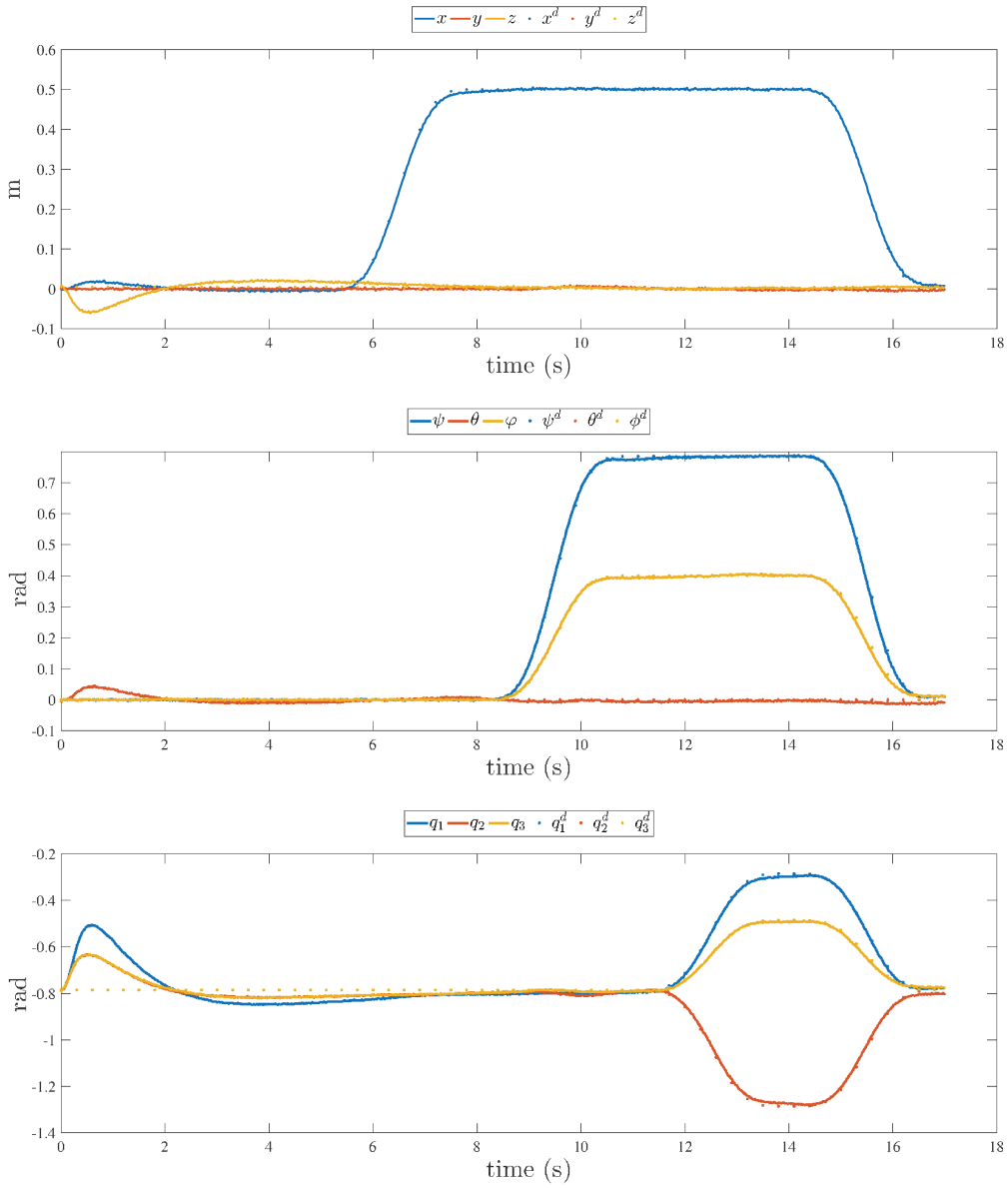


FIGURE 3.7 – Simulation du robot parallèle volant à trois drones : trajectoire désirée et suivie pour les coordonnées $\mathbf{q}_p$.

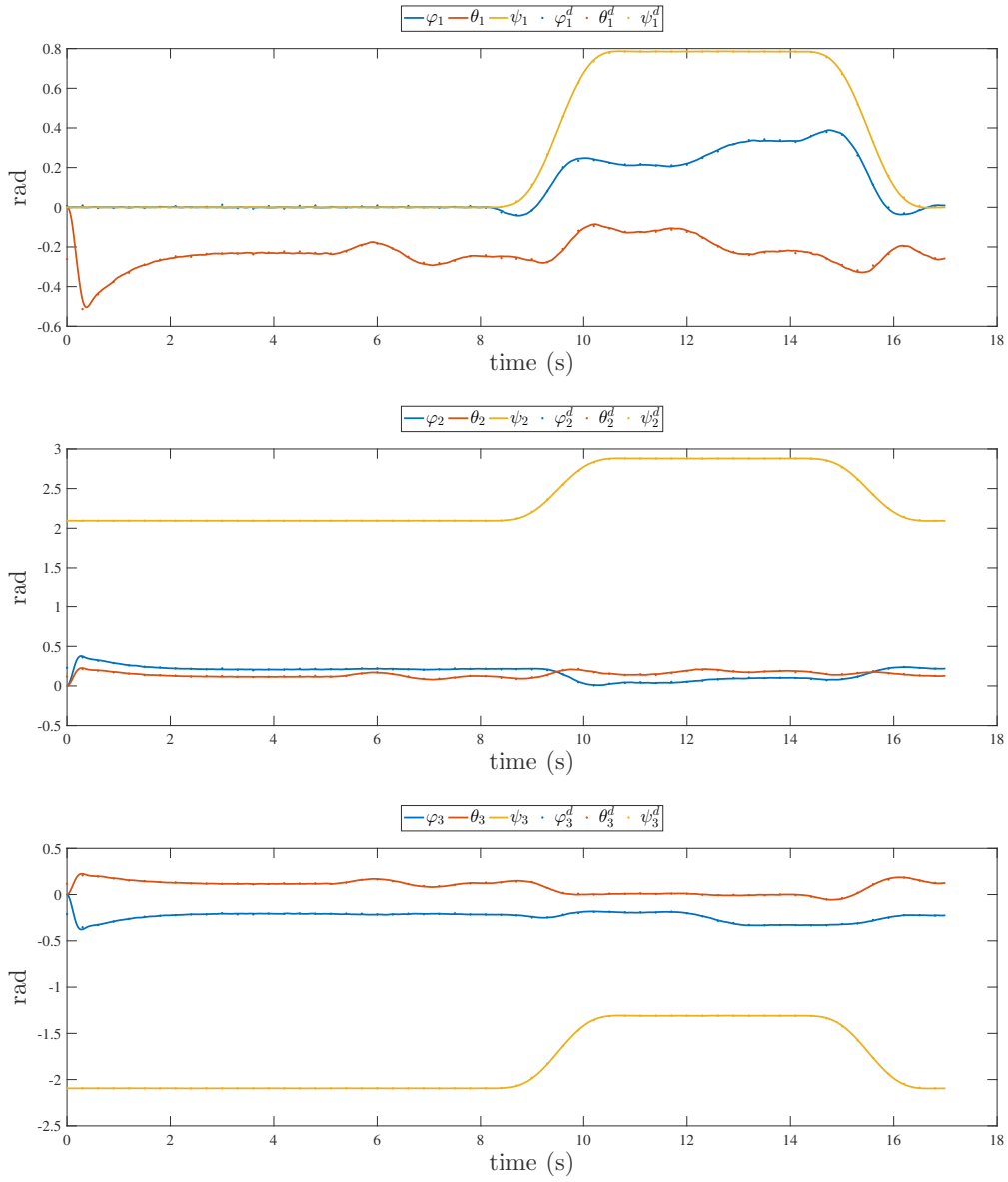


FIGURE 3.8 – Simulation du robot parallèle volant à trois drones : trajectoire désirée et suivie pour les coordonnées $\mathbf{q}_a$.

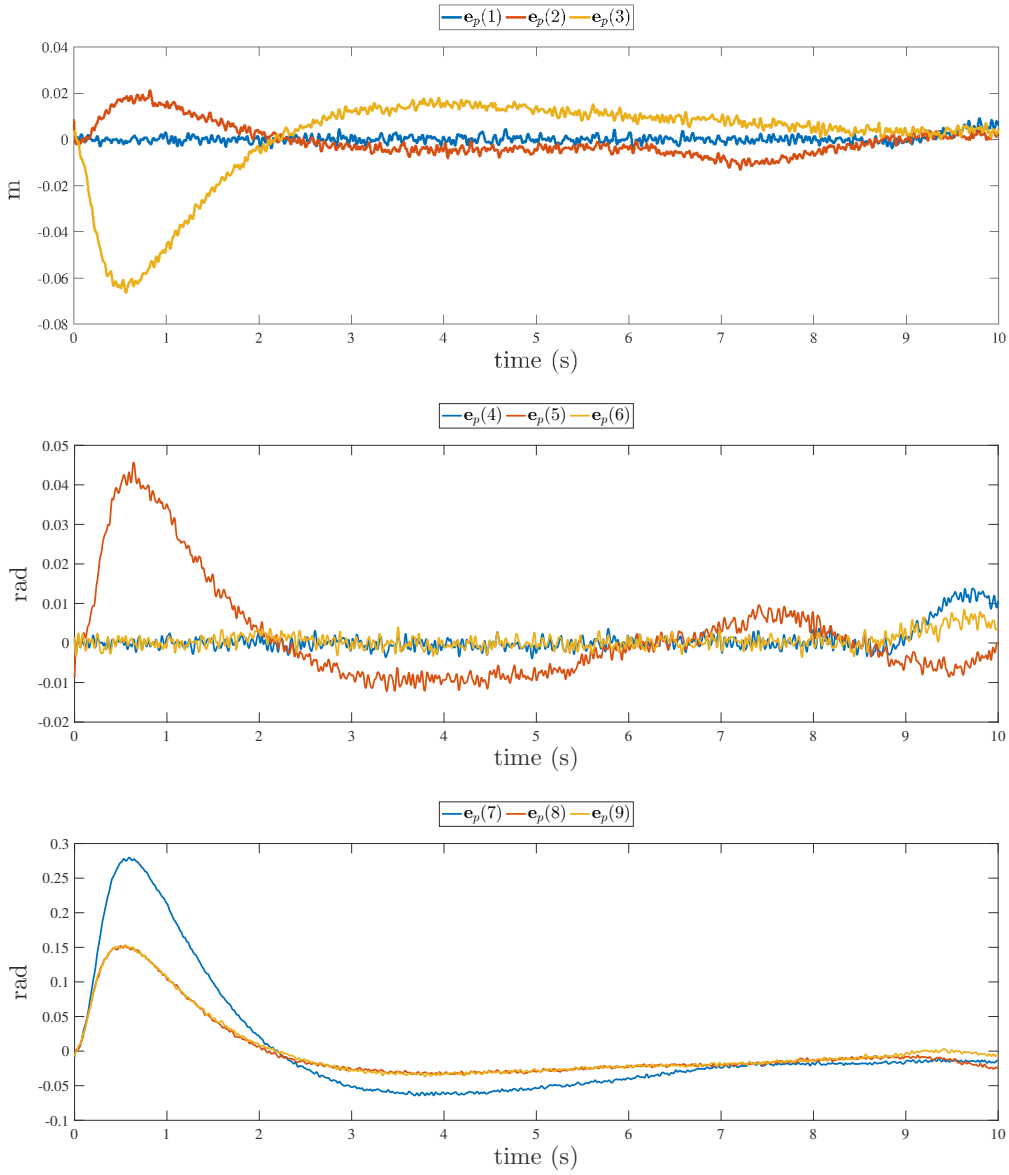


FIGURE 3.9 – Simulation du robot parallèle volant à trois drones : convergence de l'erreur $\mathbf{e}_p$.

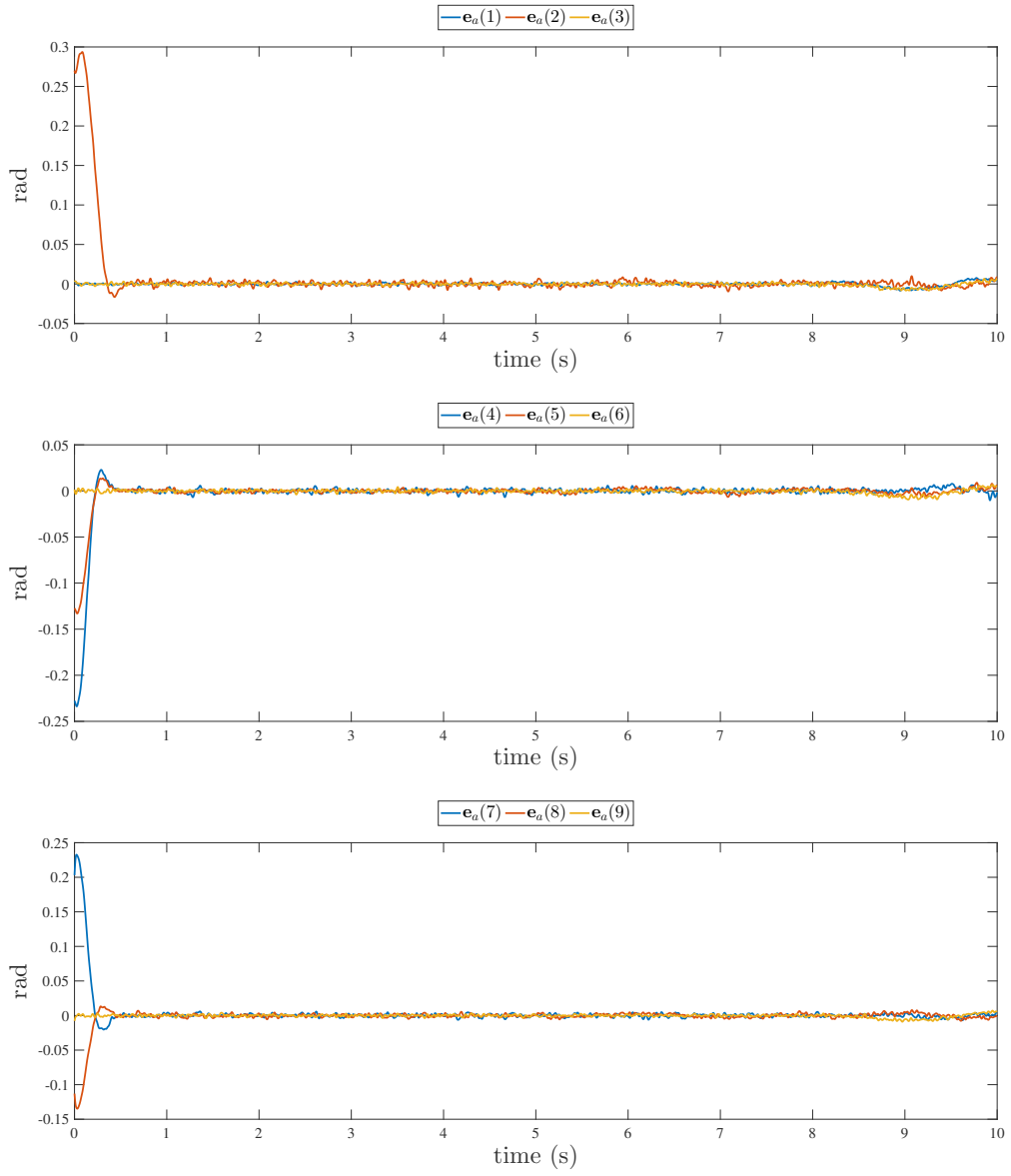


FIGURE 3.10 – Simulation du robot paralèlle volant à trois drones : convergence de l'erreur $\mathbf{e}_a$.

3.2 Robot parallèle volant à deux drones

Le second robot parallèle volant étudié est inspiré d'un robot parallèle plan : le mécanisme Biglide (Fig. 3.11). Le robot volant est obtenu en remplaçant les deux articulations prismatiques du Biglide par deux quadricoptères. Ainsi, il est composé (voir Fig. 3.12) :

- d'une chaîne cinématique passive formée de deux corps rigides reliés par une articulation pivot passive,
- de deux drones reliés à chaque extrémité de la chaîne cinématique par une articulation pivot passive.

Les axes des trois articulations pivots sont parallèles. Ce modèle possède deux différences importantes par rapport au modèle à n drones, établi au chapitre 2 :

- Le robot est conçu pour évoluer dans un plan vertical. Les liaisons entre les drones et l'architecture passive sont des articulations pivots. Nous décrirons partie 3.2.6 comment le maintien dans le plan vertical a été réalisé.
- Les liaisons entre les drones et l'architecture passive ne sont pas situées au centre de gravité des drones. Ainsi, le modèle dynamique de ce robot sera développé à nouveau afin de montrer les implications d'une telle conception.

Ces différences conduisent à une structure de modèle dynamique différente de celle présentée au chapitre 2. Cette conception a été effectuée afin de disposer rapidement d'un prototype à partir de deux drones disponibles au laboratoire LS2N dans le cadre de la thèse. Nous montrerons dans cette partie et dans le chapitre 4 que la théorie peut être adaptée à des cas moins idéaux que celui étudié au chapitre 2.

Cette partie se décompose comme suit :

1. Présentation des paramètres cinématiques du robot.
2. Modèle géométrique du robot, permettant d'obtenir ses coordonnées généralisées en fonction de la position et de l'orientation des drones.
3. Modèle cinématique et étude des singularités. Ce modèle établit la relation entre la vitesse des coordonnées généralisées du robot et les torseurs cinématiques des drones. La présence de singularités est illustrée au travers de l'étude du rang d'une matrice jacobienne.
4. Modèle dynamique du robot parallèle volant à deux drones. Dans cette sous-partie, sont abordées les différences du modèle dynamique de ce robot par rapport au concept présenté au chapitre 2.
5. Loi de commande. Cette sous-partie aborde les modifications de la loi de commande qui découlent des différences du modèle dynamique.
6. Gestion de la commande latérale et ajout d'un pivot. Dans le cadre de l'expérimentation et de simulations, il a été nécessaire d'ajouter un pivot supplémentaire dans la chaîne

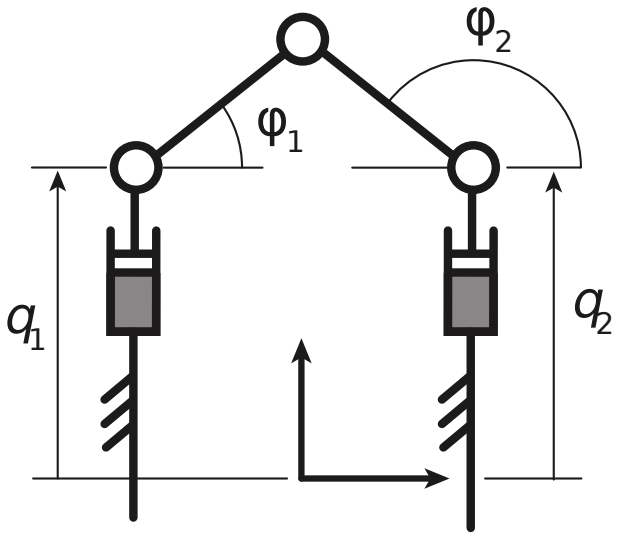


FIGURE 3.11 – Schéma du méchanisme Biglide. Les articulations grisées sont les articulations motorisées.

cinématique. Cette sous-partie développe les modifications de la loi de commande qui en découlent.

7. Simulation. Une dernière sous-partie présente les résultats de simulation, effectuée en co-simulation avec les logiciels ADAMS et SIMULINK.

3.2.1 Présentation des paramètres cinématiques

Le robot volant à deux drones est composé de quatre corps rigides (voir Fig. 3.13) :

- deux quadricoptères nommés $\mathcal{D}_1$ et $\mathcal{D}_2$,
- deux bras nommés $\mathcal{B}_1$ et $\mathcal{B}_2$.

Trois articulations passives d'axes parallèles entre eux relient ces corps et sont numérotées :

- ① pour l'articulation reliant $\mathcal{D}_1$ et $\mathcal{B}_1$,
- ② pour l'articulation reliant $\mathcal{B}_1$ et $\mathcal{B}_2$,
- ③ pour l'articulation reliant $\mathcal{D}_2$ et $\mathcal{B}_2$.

Les quadricoptères $\mathcal{D}_1$ et $\mathcal{D}_2$ sont similaires à ceux présentés dans la partie 1.1.1. Leurs paramètres caractéristiques (coefficient de poussée k_t, coefficient de traînée k_d, distance caractéristique d) sont définis de la même façon, accolés de l'indice 1 ou 2 suivant le drone considéré. Comme pour le quadricoptère seul décrit dans la partie 1.1.1, définissons les points D_1 et D_2 comme le centre des croix formées par les bras de chaque drone. La conception du robot est telle que le plan perpendiculaire à l'axe des articulations pivots et passant par D_1 contient

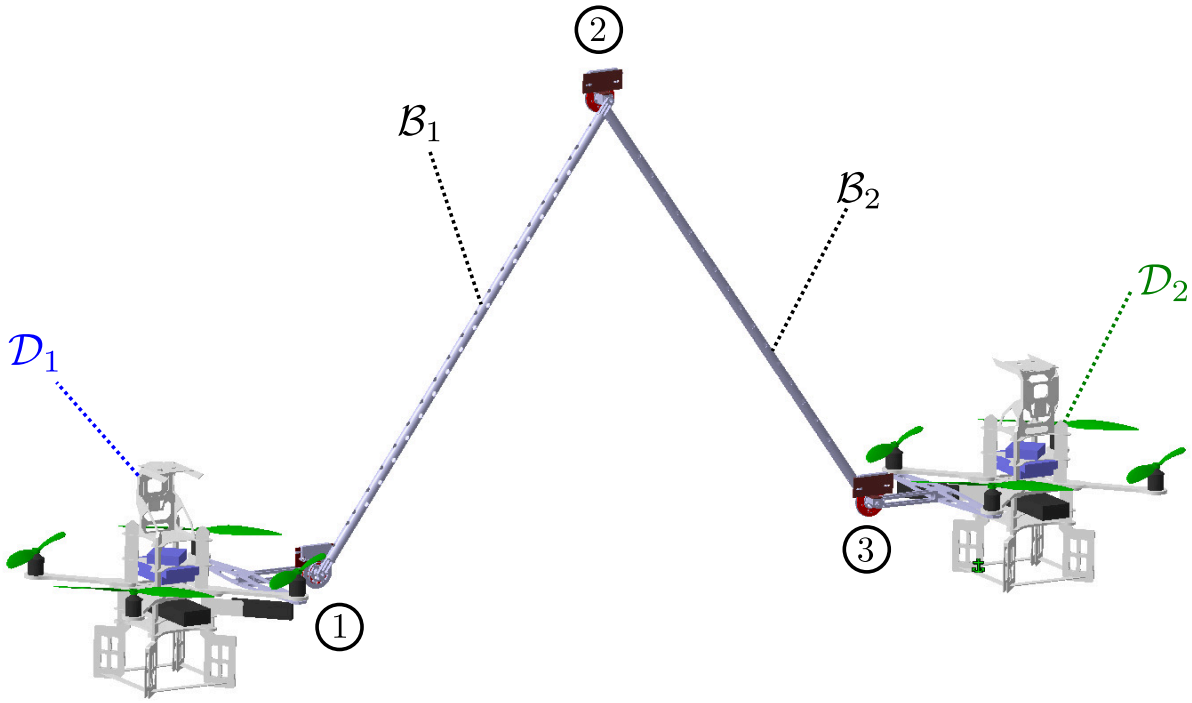


FIGURE 3.12 – Vue CAO du robot parallèle volant à deux drones.

aussi le point D_2. Nous nommerons ce plan $\mathcal{P}_p$. Les intersections de ce plan avec les axes des articulations ①, ② et ③ sont appelées respectivement O_1, O_2 et O_3.

Considérons un repère au sol $\mathcal{F}_o$ d'axes $\mathbf{x}_o$, $\mathbf{y}_o$ et $\mathbf{z}_o$. Afin de positionner le robot dans l'espace, définissons un repère $\mathcal{F}'$ attaché à $\mathcal{B}_1$ et d'origine O_1. L'axe $\mathbf{x}'$ du repère $\mathcal{F}'$ est aligné avec l'axe de l'articulation pivot ① et l'axe $\mathbf{y}'$ pointe vers O_2. Les deux repères attachés respectivement aux drones $\mathcal{D}_1$ et $\mathcal{D}_2$ sont nommés $\mathcal{F}_1$ et $\mathcal{F}_2$. Leurs centres coïncident respectivement avec O_1 et O_3. Leurs axes $\mathbf{x}_1$ et $\mathbf{x}_2$ sont alignés avec les axes des articulations respectives ① et ③. Leurs axes $\mathbf{y}_1$ et $\mathbf{y}_2$ pointent respectivement vers D_1 et D_2. L'orientation du repère $\mathcal{F}'$ par rapport au repère $\mathcal{F}_o$ est obtenue par trois rotations successives d'axes $\mathbf{x}_o$, $\mathbf{y}_o$ et $\mathbf{z}_o$:

$$\mathbf{v} = \mathbf{R}_{\mathbf{z}_o}(\psi)\mathbf{R}_{\mathbf{y}_o}(\theta)\mathbf{R}_{\mathbf{x}_o}(\varphi)\mathbf{v}' \tag{E3.16}$$

La même représentation permet de définir les coordonnées d'orientation des repères $\mathcal{F}_1$ (respectivement $\mathcal{F}_2$) par rapport au repère $\mathcal{F}_o$. Ces coordonnées sont notées $(\psi_1, \theta_1, \varphi_1)$ et respectivement $(\psi_2, \theta_2, \varphi_2)$.

Choisissons le corps $\mathcal{B}_1$ et son repère associé $\mathcal{F}'$ pour positionner et orienter le robot dans l'espace à l'aide des coordonnées

- x, y, z l'origine du repère $\mathcal{F}'$ exprimé dans le repère d'origine $\mathcal{F}_o$,
- ψ, θ, φ l'orientation de $\mathcal{F}'$ par rapport au repère $\mathcal{F}_o$.

Trois coordonnées supplémentaires sont nécessaires pour décrire la configuration des articu-

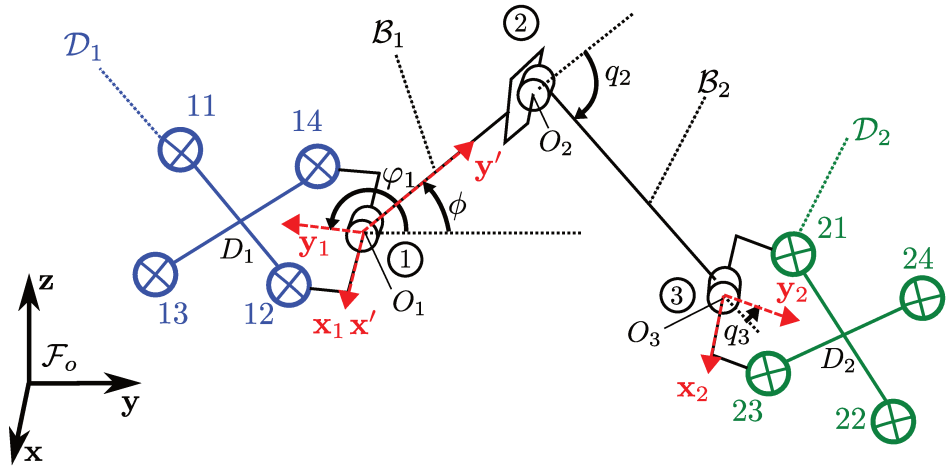


FIGURE 3.13 – Paramètres cinématiques du robot parallèle volant à deux drones.

lations pivots passives. Les coordonnées sélectionnées sont

- q_2 la coordonnée d'orientation entre les corps $\mathcal{B}_1$ et $\mathcal{B}_2$ reliés par l'articulation pivot ②,
- q_3 la coordonnée d'orientation entre les corps $\mathcal{B}_2$ et $\mathcal{D}_2$, reliés par l'articulation pivot ③,
- la coordonnée d'orientation φ_1 du drone $\mathcal{D}_1$ définie ci-dessus.

Le vecteur des coordonnées généralisées du robot est alors défini par $\mathbf{q} = [y\ x\ z\ \psi\ \theta\ \varphi\ q_2\ q_3\ \varphi_1]^T$. Le choix de commencer par la coordonnée y pour définir la position cartésienne facilite la définition du robot selon la convention DHKK. Notons également que la séquence des rotations (E3.16) diffère de celle choisie au chapitre 2. En effet, en utilisant cette séquence de rotations, du fait des contraintes cinématiques de l'architecture passive, nous disposons des relations suivantes :

$$\begin{aligned} \psi_1 &= \psi_2 = \psi \\ \theta_1 &= \theta_2 = \theta \\ \varphi_2 &= \varphi + q_2 + q_3 \end{aligned} \tag{E3.17}$$

La distance du point A au point B est notée ℓ_{AB}. Les longueurs caractéristiques du robot sont

- $\ell_1 = \ell_{O_1O_2}$ et $\ell_2 = \ell_{O_2O_3}$ définissant la longueur des corps $\mathcal{B}_1$ et $\mathcal{B}_2$,
- $\ell_{d1} = \ell_{O_1D_1}$ et $\ell_{d2} = \ell_{O_3D_2}$ la distance sur chaque drone entre le centre de sa croix et l'axe de l'articulation pivot.

La configuration du robot volant peut être représentée à l'aide de la convention Denhavit-Hartenberg modifiée par Khalil et Kleinfinger (DHKK). De façon similaire à la partie 2.2.2, une chaîne cinématique virtuelle est définie pour permettre la représentation de la position et de l'orientation du robot dans l'espace. Les paramètres DHKK du robot sont donnés dans le tableau (T3.3).

i	ant_i	μ_i	σ_i	γ_i	b_i	α_i	d_i	θ_i	r_i
$\mathcal{V}_1$	$\mathcal{F}_o$	0	1	0	0	$-\frac{\pi}{2}$	0	$-\frac{\pi}{2}$	y
$\mathcal{V}_2$	$\mathcal{V}_1$	0	1	0	0	$-\frac{\pi}{2}$	0	$-\frac{\pi}{2}$	x
$\mathcal{V}_3$	$\mathcal{V}_2$	0	1	0	0	$-\frac{\pi}{2}$	0	$-\frac{\pi}{2}$	z
$\mathcal{V}_4$	$\mathcal{V}_3$	0	0	0	0	0	0	ψ	0
$\mathcal{V}_5$	$\mathcal{V}_4$	0	0	0	0	$-\frac{\pi}{2}$	0	$-\frac{\pi}{2}+\theta$	0
$\mathcal{B}_1$	$\mathcal{V}_5$	0	0	0	0	$-\frac{\pi}{2}$	0	$-\frac{\pi}{2}+\varphi$	0
$\mathcal{B}_2$	$\mathcal{B}_1$	0	0	0	0	0	ℓ_1	q_2	0
$\mathcal{D}_2$	$\mathcal{B}_2$	0	0	0	0	0	ℓ_2	q_3	0
$\mathcal{D}_1$	$\mathcal{V}_5$	0	0	0	0	0	0	$-\frac{\pi}{2}+\varphi_1$	0

TABLE 3.3 – Les paramètres DHKK du robot parallèle volant à deux drones.

À partir de la définition des ces paramètres cinématiques, nous allons maintenant étudier le modèle géométrique du robot.

3.2.2 Modèle géométrique direct

Dans cette partie, nous considérons connues la position et l'orientation des drones, à savoir, les positions des points O_1 et O_3, données par les vecteurs ${}^o\mathbf{r}_1$ et ${}^o\mathbf{r}_2$ dans le repère $\mathcal{F}_o$, ainsi que les coordonnées d'orientation des drones ψ_1, θ_1, φ_1, ψ_2, θ_2 et φ_2. L'objectif du modèle géométrique développé ici est de calculer, en fonction de ces coordonnées connues, le vecteur des coordonnées généralisées du robot $\mathbf{q} = [y\ x\ z\ \psi\ \theta\ \varphi\ q_2\ q_3\ \varphi_1]^T$.

Les coordonnées x, y et z sont les coordonnées de position du point O_1. De plus, les contraintes cinématiques données en équation (E3.17) permettent d'obtenir $\theta = \theta_1$ et $\psi = \psi_1$. La coordonnée φ_1 est également connue. Il reste à obtenir les coordonnées φ, q_2 et q_3.

Pour cela, plaçons nous dans le plan $\mathcal{P}_p$ orthogonal aux axes des articulations pivots (voir Fig. 3.14). Soient le vecteur $\overrightarrow{O_1O_3}$ et R_p la matrice de rotation d'angles ψ et θ donnée par :

$$\mathbf{R}_p = \begin{bmatrix} C_\psi C_{\theta_1} & -S_\psi & C_\psi S_{\theta_1} \\ C_{\theta_1} S\psi & C_\psi & S_\psi S_{\theta_1} \\ -S_{\theta_1} & 0 & C_{\theta_1} \end{bmatrix} \tag{E3.18}$$

Le vecteur $\overrightarrow{O_1O_3}$ projeté dans le plan $\mathcal{P}_p$ est donné par l'équation :

$${}^p\overrightarrow{O_1O_3} = \mathbf{R}_p^T({}^o\mathbf{r}_2 - {}^o\mathbf{r}_1) \tag{E3.19}$$

Soient δ_y et δ_z les deux dernières coordonnées du vecteur ${}^p\overrightarrow{O_1O_3}$. Le modèle géométrique consiste à trouver la position de l'articulation ② (point O_2) dans le plan $\mathcal{P}_p$ (voir Fig.3.14). Cette position est définie par l'intersection de deux cercles : l'un de centre O_1 et de rayon ℓ_1 et l'autre de centre O_3 et de rayon ℓ_2. Hors des configurations singulières, deux solutions géométriques sont possibles. Le vecteur ${}^p\overrightarrow{O_1O_2}$ dans le plan $\mathcal{P}_p$ est alors donné par (en

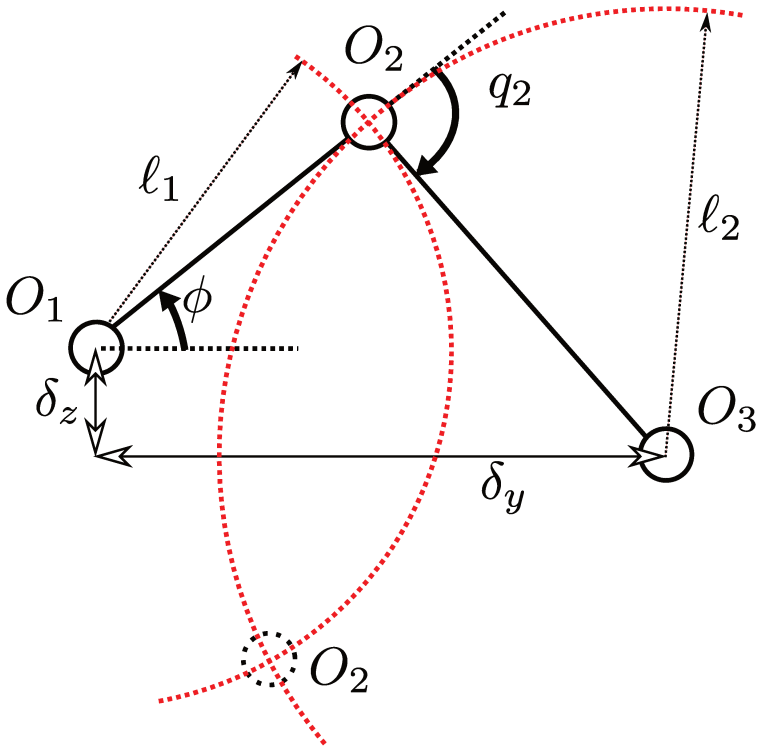


FIGURE 3.14 – Robot parallèle volant à deux drones : illustration des solutions du modèle géométrique dans le plan $\mathcal{P}_p$.

ignorant les étapes de calcul intermédiaires) :

$$ {}^p\overrightarrow{O_1O_2} = \begin{bmatrix} \frac{a}{\delta_y} - \frac{\delta_z}{\delta_y} D \\ D \end{bmatrix} \tag{E3.20} $$

avec $a = (\ell_1^2 - \ell_2^2 + \delta_y^2 + \delta_z^2)/2$, $D = -B \pm \sqrt{\Delta}/(2A)$, $B = -2a\delta_z$, $\Delta = B^2 - 4AC$, $A = \delta_z^2 + \delta_y^2$ et $C = a^2 - \ell_1^2\delta_y^2$. Les deux solutions du terme D correspondent aux deux solutions possibles du modèle géométrique, appelées *modes d'assemblages* par analogie aux robots parallèles classiques.

Soit y_2 et z_2 les deux coordonnées dans le plan $\mathcal{P}_p$ du vecteur ${}^p\overrightarrow{O_1O_2}$. Le calcul des coordonnées φ et q_2 s'obtient par :

$$ \begin{aligned} \varphi &= \text{atan2}(z_2, y_2) \\ q_2 &= \text{atan2}(\delta_z - z_2, \delta_y - y_2) - \varphi \end{aligned} \tag{E3.21} $$

Alors, il ne manque plus que la coordonnée q_3 qui est obtenue à partir de la relation (E3.17)

$$ q_3 = \varphi_2 - \varphi - q_2 \tag{E3.22} $$

pour obtenir le vecteur des coordonnées $\mathbf{q}$.

Après avoir établi le modèle géométrique du robot, la partie suivante s'intéresse à son modèle cinématique.

3.2.3 Modèle cinématique et singularités

Soient ${}^1\mathbf{t}_1$ et ${}^2\mathbf{t}_2$ les torseurs cinématiques des deux drones, exprimés dans leurs repères locaux respectifs $\mathcal{F}_1$ et $\mathcal{F}_2$ à l'origine de ces repères. Il est possible d'exprimer chacun de ces torseurs en fonction de la dérivée des coordonnées généralisées du robot $\dot{\mathbf{q}}$ et des matrices jacobiennes cinématiques $\mathbf{J}_1$ et $\mathbf{J}_2$:

$$ \begin{aligned} {}^1\mathbf{t}_1 &= \mathbf{J}_1\dot{\mathbf{q}} \\ {}^2\mathbf{t}_2 &= \mathbf{J}_2\dot{\mathbf{q}} \end{aligned} \tag{E3.23} $$

L'algorithme du calcul de ces matrices jacobiennes à partir des paramètres DHKK est donné en annexe A.1. L'expression développée de ces deux matrices est (avec les notations abrégées données table (T3.4))

$$ \mathbf{J}_1 = \begin{bmatrix} S_\psi C_\theta & C_\psi C_\theta & -S_\theta & 0 & 0 & 0 & 0 & 0 & 0 \\ S_\psi S_\theta S_{\varphi_1} + C_\psi C_{\varphi_1} & C_\psi S_\theta S_{\varphi_1} - S_\psi C_{\varphi_1} & C_\theta S_{\varphi_1} & 0 & 0 & 0 & 0 & 0 & 0 \\ S_\psi S_\theta C_{\varphi_1} - C_\psi S_{\varphi_1} & C_\psi S_\theta C_{\varphi_1} + S_\psi S_{\varphi_1} & C_\theta C_{\varphi_1} & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & -S_\theta & 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & S_{\varphi_1} C_\theta & C_{\varphi_1} & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & C_{\varphi_1} C_\theta & -S_{\varphi_1} & 0 & 0 & 0 & 0 \end{bmatrix} $$

x	$\sin(x)$	$\cos(x)$
ψ	S_ψ	C_ψ
φ	S_φ	C_φ
θ	S_θ	C_θ
q_2	S_2	C_3
q_3	S_2	C_3
φ_1	S_{φ_1}	C_{φ_1}
$\varphi + q_2 + q_3 = \varphi_2$	S_{φ_2}	C_{φ_2}
$\varphi + q_2$	S_{12}	C_{12}
$q_2 + q_3$	S_{23}	C_{23}

TABLE 3.4 – Table des notations abrégées des fonctions sinus et cosinus.

$$\mathbf{J}_2 = \begin{bmatrix} S_\psi C_\theta & C_\psi C_\theta & -S_\theta \\ S_\psi S_\theta S_{\varphi_2} + C_\psi C_{\varphi_2} & C_\psi S_\theta S_{\varphi_2} - S_\psi C_{\varphi_2} & C_\theta S_{\varphi_2} \\ S_\psi S_\theta C_{\varphi_2} - C_\psi S_{\varphi_2} & C_\psi S_\theta C_{\varphi_2} + S_\psi S_{\varphi_2} & C_\theta C_{\varphi_2} \\ 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix}$$

$$\begin{bmatrix} (-\ell_1 C_\varphi - \ell_2 C_{12}) C_\theta & \ell_1 S_\varphi + \ell_2 S_{12} & 0 & 0 & 0 & 0 \\ (-\ell_1 S_{23} - \ell_2 S_3) S_\theta & 0 & \ell_1 S_{23} + \ell_2 S_3 & \ell_2 S_3 & 0 & 0 \\ (-\ell_1 C_{23} - \ell_2 C_3) S_\theta & 0 & \ell_1 C_{23} + \ell_2 C_3 & \ell_2 C_3 & 0 & 0 \\ -S_\theta & 0 & 1 & 1 & 1 & 0 \\ S_{\varphi_2} C_\theta & C_{\varphi_2} & 0 & 0 & 0 & 0 \\ C_{\varphi_2} C_\theta & -S_{\varphi_2} & 0 & 0 & 0 & 0 \end{bmatrix}$$

Considérons maintenant la matrice $\mathbf{J}$ telle que $\mathbf{J}^T = [\mathbf{J}_1^T \ \mathbf{J}_2^T]$. Cette matrice jacobienne cinématique relie les torseurs cinématiques ${}^1\mathbf{t}_1$ et ${}^2\mathbf{t}_2$ à la dérivée des coordonnées généralisées du robot $\mathbf{q}$:

$$\begin{bmatrix} {}^1\mathbf{t}_1 \\ {}^2\mathbf{t}_2 \end{bmatrix} = \mathbf{J}\dot{\mathbf{q}} \tag{E3.24}$$

Étude des singularités

La matrice $\mathbf{J}$ est de dimension (12×9). Elle est également de rang plein sauf :

- dans le cas d'une singularité de représentation. Par exemple, $\theta = \pm\frac{\pi}{2}$ implique une dépendance linéaire entre :
  - la colonne 4 : $\pm[0\ 0\ 0\ -1\ 0\ 0\ 0\ -\ell_1 S_{23} - \ell_2 S_3 \quad -\ell_1 C_{23} - \ell_2 C_3\ -1\ 0\ 0]^T$
  - la colonne 6 : $[0\ 0\ 0\ 0\ 0\ 0\ 0\ \ell_1 S_{23} + \ell_2 S_3 \quad \ell_1 C_{23} + \ell_2 C_3\ 1\ 0\ 0]^T$
  - la colonne 9 : $[0\ 0\ 0\ 1\ 0\ 0\ 0\ 0\ 0\ 0\ 0\ 0]^T$

  Cette singularité est liée au choix de représentation des rotations. Dans le cadre des

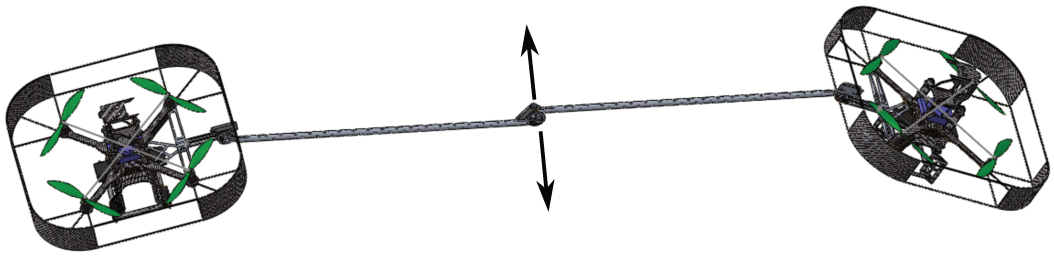


FIGURE 3.15 – Exemple d'une configuration singulière pour le robot volant à deux drones. Les flèches représentent le mouvement infinitésimal non contrôlé localement.

applications de cette étude, ces singularités sont éloignées de l'espace de travail considéré.

- dans le cas d'une singularité parallèle de la structure passive (similaire à une singularité de Type 2 pour un robot parallèle [Briot et Khalil, 2015]). Cette singularité se produit pour $q_2 = k\pi$, k entier. Dans ce cas, il existe une dépendance linéaire entre :
  - la colonne 6 : $[0\ 0\ 0\ 0\ 0\ 0\ (\ell_1 \pm \ell_2)S_3 \quad (\ell_1 \pm \ell_2)C_3\ 1\ 0\ 0]^T$
  - la colonne 7 : $[0\ 0\ 0\ 0\ 0\ 0\ \ell_2 S_3 \quad \ell_2 C_3\ 1\ 0\ 0]^T$
  - la colonne 8 :$[0\ 0\ 0\ 0\ 0\ 0\ 0\ 0\ 0\ 1\ 0\ 0]^T$

  Pour ce type de singularité, il existe un vecteur $\dot{\mathbf{q}}$ non nul tel que $\mathbf{J}\dot{\mathbf{q}} = \mathbf{0}$. Localement, un mouvement infinitésimal de la structure passive est possible même en ayant fixé la position et l'orientation des quadricoptères. Ce mouvement infinitésimal correspond à une direction non-contrôlée dans cette configuration. La figure 3.15 illustre le robot dans une telle configuration singulière.

La gestion des singularités pour la commande d'un robot parallèle volant est l'objet du chapitre 5. Dans un premier temps, seules des trajectoires de vol éloignées des configurations singulières seront effectuées par le robot. Nous considérerons donc pour la suite de cette partie que le robot n'est pas en configuration singulière.

Le modèle cinématique inverse donne la vitesse des coordonnées $\dot{\mathbf{q}}$ en fonction des torseurs cinématiques ${}^1\mathbf{t}_1$ et ${}^2\mathbf{t}_2$ exprimés dans leurs repères locaux respectifs. Ce modèle s'obtient par l'inversion de l'équation (E3.24) :

$$\dot{\mathbf{q}} = \mathbf{J}^+ \begin{bmatrix} {}^1\mathbf{t}_1 \\ {}^2\mathbf{t}_2 \end{bmatrix} \tag{E3.25}$$

La matrice Jacobienne $\mathbf{J}$ est de rang plein mais possède plus de lignes que de colonnes. Son inversion passe alors par le calcul de la pseudo-inverse $\mathbf{J}^+ = (\mathbf{J}^T\mathbf{J})^{-1}\mathbf{J}^T$. Elle offre, dans le cas général, une solution approchée de l'équation (E3.24). Cependant, des contraintes cinématiques existent sur ${}^1\mathbf{t}_1$ et ${}^2\mathbf{t}_2$ telles que (E3.25) est une solution exacte de l'équation (E3.24).

Dans la partie suivante, nous allons traiter du modèle dynamique du robot et des différences notables avec celui établi au chapitre 2.

3.2.4 Modèle dynamique

Le modèle dynamique d'un robot parallèle volant, établi dans la partie 2.3, se base sur l'hypothèse que les articulations sphériques sont situées au centre de gravité des drones pour relier les quadricoptères à la structure. Ces conditions ne sont pas remplies dans le modèle du robot volant à deux drones conçu pour l'expérimentation. Nous allons cependant montrer, sous certaines hypothèses, que le modèle dynamique peut être découplé dans une forme similaire à celle établie dans le cas du robot complet au chapitre 2. La première hypothèse, qui sera valable pour le reste de l'étude, est que le système est maintenu proche d'un plan vertical défini par $x = 0$, $\theta = 0$ et $\psi = 0$. Dans ce cas, le plan $\mathcal{P}_p$ (plan perpendiculaire aux axes des pivots et contenant D_1) est confondu avec le plan $x = 0$. L'idée de travailler dans un plan est issue du fait que l'espace de travail du mécanisme d'origine, le Biglide, est également contenu dans un plan. Notre robot présente des actionnements supplémentaires par rapport au Biglide qui seront utilisés pour maintenir le système dans ce plan vertical. Le fait de contraindre le robot à rester dans ce plan nous permet d'étudier la dynamique de la chaîne passive dans le plan vertical. Sans cette hypothèse, de nombreux couplages apparaissent et il n'est pas possible d'obtenir le modèle découplé souhaité.

L'étude du modèle dynamique du robot s'effectuera en quatre étapes :

1. Modèle dynamique de la chaîne passive dans le plan vertical : ce modèle permet d'établir une relation entre l'accélération des coordonnées de la chaîne passive et les forces de poussées produites par les drones.
2. Modèle dynamique des orientations des drones : ce modèle associe les couples produits par les drones à l'accélération des coordonnées définissant l'orientation des drones.
3. Bilan des modèles dynamiques montrant les similarités et les différences avec les modèles dynamiques décrits au chapitre 2.
4. Modèle dynamique le long de l'axe $\mathbf{x}$: la formalisation complémentaire du modèle dynamique du robot le long de cet axe sera nécessaire pour établir une commande de maintien du robot dans le plan $x = 0$.

Modèle restreint à la chaîne passive dans le plan vertical

Supposons donc la chaîne passive (corps $\mathcal{B}_1$ et $\mathcal{B}_2$) telle que le plan $\mathcal{P}_p$ soit confondu avec le plan vertical d'équation $x = 0$ (voir Fig. 3.16). Dans ce plan, la configuration de la chaîne passive est entièrement déterminée par la position et l'orientation du corps $\mathcal{B}_1$ (coordonnées y, z et φ) et par la coordonnée de l'articulation ② (q_2) située entre le corps $\mathcal{B}_1$ et $\mathcal{B}_2$. Soit $\mathbf{q}_p = [y\ z\ \varphi\ q_2]^T$ le vecteur des coordonnées de la chaîne passive dans le plan vertical.

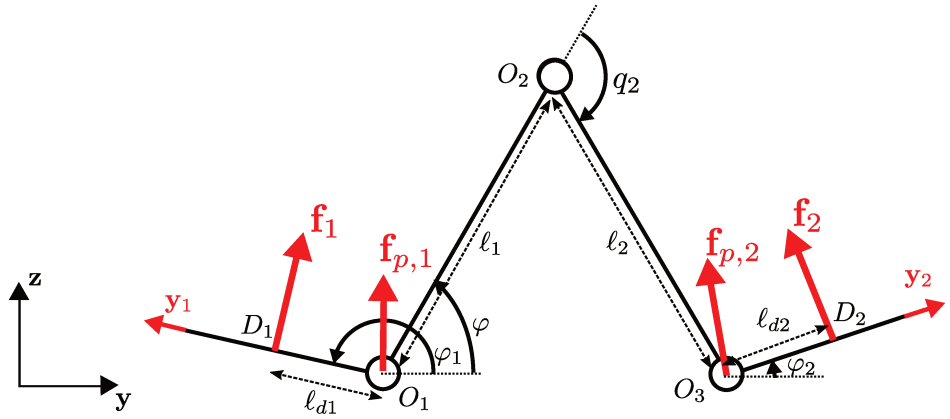


FIGURE 3.16 – Robot parallèle volant à deux drones dans une configuration telle que le plan $\mathcal{P}_p$ soit confondu avec le plan d'équation $x = 0$.

Pour la suite de cette étude, les équations seront établies dans le plan $\mathcal{P}_p$ confondu avec le plan d'équation $x = 0$. Le sous-vecteur des coordonnées dans ce plan d'un vecteur ${}^{o}\mathbf{v}$, exprimé dans le repère d'origine $\mathcal{F}_o$, est formé par les deux dernières coordonnées de ${}^{o}\mathbf{v}$ (selon $\mathbf{y}$ et $\mathbf{z}$). Nous écrirons ce sous-vecteur ${}^{o,p}\mathbf{v}$. De façon similaire, nous définirons ${}^{i,p}\mathbf{v}$ le sous-vecteur formé par les deux dernières coordonnées d'un vecteur ${}^{i}\mathbf{v}$ exprimé dans le repère $\mathcal{F}_i$ $(i = 1, 2)$. Sous la contrainte $\psi = 0$ et $\theta = 0$, le repère $\mathcal{F}_i$ est obtenu à partir du repère $\mathcal{F}_o$ par une rotation d'axe $\mathbf{x}$ et d'angle φ_i. Les vecteurs ${}^{o,p}\mathbf{v}$ et ${}^{i,p}\mathbf{v}$ sont alors reliés par l'équation :

$$ {}^{o,p}\mathbf{v} = {}^{p}\mathbf{R}_i \, {}^{i,p}\mathbf{v} \tag{E3.26} $$

avec la matrice de rotation dans le plan $x = 0$ et d'angle φ_i ${}^{p}\mathbf{R}_i = \begin{bmatrix} \cos\varphi_i & -\sin\varphi_i \\ \sin\varphi_i & \cos\varphi_i \end{bmatrix}$.

Matrice jacobienne cinématique de la structure passive dans le plan

Les vecteurs de position des points O_1 et O_2 exprimés dans le repère d'origine $\mathcal{F}_o$ sont notés ${}^{o}\mathbf{r}_1$ et ${}^{o}\mathbf{r}_2$. Les vecteurs ${}^{o,p}\mathbf{r}_1$ et ${}^{o,p}\mathbf{r}_2$ peuvent s'écrire en fonction des coordonnées $\mathbf{q}_p$ de la structure passive :

$$ \begin{aligned} {}^{o,p}\mathbf{r}_1 &= \begin{bmatrix} x \\ y \end{bmatrix} \\ {}^{o,p}\mathbf{r}_2 &= \begin{bmatrix} x + \ell_1 \cos\varphi + \ell_2 \cos(\varphi + q_2) \\ y + \ell_1 \sin\varphi + \ell_2 \sin(\varphi + q_2) \end{bmatrix} \end{aligned} \tag{E3.27} $$

Les vitesses des ces vecteurs sont obtenues en dérivant l'équation (E3.27). Elle peuvent être exprimées sous la forme suivante :

$$ \begin{bmatrix} {}^{o,p}\dot{\mathbf{r}}_1 \\ {}^{o,p}\dot{\mathbf{r}}_2 \end{bmatrix} = \mathbf{J}_p \dot{\mathbf{q}}_p \tag{E3.28} $$

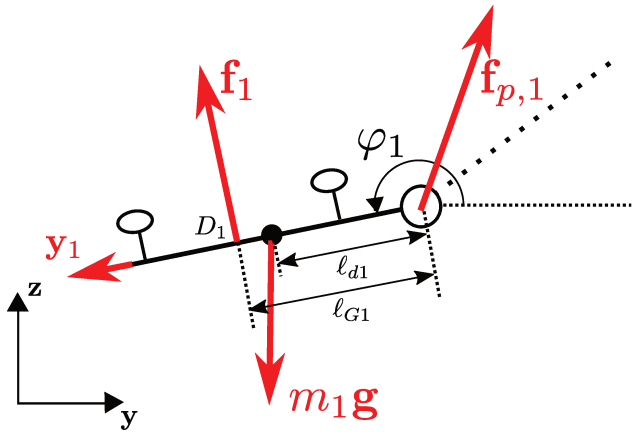


FIGURE 3.17 – Robot parallèle volant à deux drones : bilan des forces agissants sur le drone 1.

La matrice jacobienne cinématique $\mathbf{J}_p$ s'écrit (en utilisant les notations raccourcies données dans le tableau (T3.4)) :

$$\mathbf{J}_p = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 1 & 0 & -\ell_1 S_\varphi - \ell_2 S_{12} & -\ell_2 S_{12} \\ 0 & 1 & \ell_1 C_\varphi + \ell_2 C_{12} & \ell_2 C_{12} \end{bmatrix} \tag{E3.29}$$

Comme pour le modèle complet, la matrice jacobienne $\mathbf{J}_p$ est singulière pour $q_2 = k\pi$, k entier. Les deux dernières colonnes de la jacobienne deviennent alors linéairement dépendantes.

Modèle dynamique de l'architecture passive dans le plan vertical

La chaîne passive subit dans le plan les forces d'interaction avec les drones à ses deux extrémités ${}^{o,p}\mathbf{f}_{p,1}$ et ${}^{o,p}\mathbf{f}_{p,2}$. Les moments d'interaction selon l'axe $\mathbf{x}$ ne sont pas considérés car ils ne sont pas transmis à la structure passive par les articulations pivots. La première étape de calcul est similaire au modèle à n drones (E2.10) du chapitre 2. Ainsi, nous obtenons :

$$\mathbf{M}_p \ddot{\mathbf{q}}_p + \mathbf{c}_p = \mathbf{J}_{v,1}^T {}^{o,p}\mathbf{f}_{p,1} + \mathbf{J}_{v,2}^T {}^{o,p}\mathbf{f}_{p,2} = \mathbf{J}_p^T \begin{bmatrix} {}^{o,p}\mathbf{f}_{p,1} \\ {}^{o,p}\mathbf{f}_{p,2} \end{bmatrix} \tag{E3.30}$$

avec :

- ${}^{o,p}\mathbf{f}_{p,i}$ $(i = 1, 2)$ le sous-vecteur dans le plan $\mathcal{P}_p$ de la force d'interaction entre le drone $\mathcal{D}_i$ et la structure passive, exprimé dans le repère $\mathcal{F}_o$,
- $\mathbf{M}_p$ la matrice d'inertie généralisée de la chaîne passive de dimension (4×4),
- $\mathbf{c}_p$ le vecteur de dimension 4 comprenant les effets gravitationnels, centrifuges et de Coriolis.

La suite des calculs diffère du modèle général car les centres de rotation des liaisons ne

correspondent pas avec le centre de gravité des drones. Il est possible de relier les forces $\mathbf{f}_{p,i}$ appliquées aux extrémités de la chaîne passive aux forces $\mathbf{f}_i$ fournies par les drones grâce à l'application du principe fondamental de la dynamique au centre de gravité de chaque drone $\mathcal{D}_i$, comme illustré Fig. 3.17 :

$$ {}^{p}\mathbf{R}_i \, {}^{i,p}\mathbf{f}_i - {}^{o,p}\mathbf{f}_{p,i} + m_i \, {}^{o,p}\mathbf{g} = m_i \, {}^{o,p}\ddot{\boldsymbol{\xi}}_i \tag{E3.31} $$

avec

- ${}^{o,p}\boldsymbol{\xi}_i$ $(i = 1, 2)$ le sous-vecteur dans le plan $\mathcal{P}_p$ de la position du centre de gravité du drone $\mathcal{D}_i$, exprimé dans le repère $\mathcal{F}_o$, et donc ${}^{o,p}\ddot{\boldsymbol{\xi}}_p$ son accélération.
- m_i la masse du drone $\mathcal{D}_i$.

La position du centre de gravité du drone $\mathcal{D}_i$ dans le repère $\mathcal{F}_o$ en fonction de sa position dans le repère $\mathcal{F}_i$ est donné par :

$$ {}^{o,p}\boldsymbol{\xi}_i = {}^{o,p}\mathbf{r}_i + {}^{p}\mathbf{R}_i \, {}^{i,p}\boldsymbol{\xi}_i \tag{E3.32} $$

Le repère $\mathcal{F}_i$ étant un repère attaché au drone $\mathcal{D}_i$, le vecteur ${}^{i,p}\boldsymbol{\xi}_i$ est un vecteur constant ne dépendant que des caractéristiques physiques de chaque quadricoptère. Nous poserons l'hypothèse que le centre de gravité du drone $\mathcal{D}_i$ est situé selon le vecteur $\mathbf{y}_i$ à une distance ℓ_{Gi}. Cette hypothèse sera validée lors de l'identification des paramètres dynamiques des drones effectuée dans la partie 4.2. Nous verrons cependant que, du fait de l'ajout des éléments nécessaires aux liaisons pivots sur les drones, le centre de gravité n'est pas nécessairement confondu avec le point D_i, i.e. $\ell_{Gi} \neq \ell_{di}$ (voir Fig. 3.17). Le vecteur ${}^{i,p}\boldsymbol{\xi}_i$ s'écrit alors ${}^{i,p}\boldsymbol{\xi}_i = [\ell_{Gi} \; 0]^T$ et l'équation (E3.32) s'exprime sous la forme :

$$ {}^{o,p}\boldsymbol{\xi}_i = {}^{o,p}\mathbf{r}_i + \ell_{Gi} \, {}^{o,p}\mathbf{y}_i \tag{E3.33} $$

En dérivant deux fois (E3.33) par rapport au temps, on exprime ${}^{o,p}\ddot{\boldsymbol{\xi}}_i$ l'accélération du centre de gravité du drone $\mathcal{D}_i$ par :

$$ \begin{aligned} {}^{o,p}\ddot{\boldsymbol{\xi}}_i &= {}^{o,p}\ddot{\mathbf{r}}_i - \dot{\varphi}_i^{\,2} \ell_{Gi} \, {}^{o,p}\mathbf{y}_i + \ddot{\varphi}_i \ell_{Gi} \, {}^{o,p}\mathbf{z}_i \\ &= {}^{o,p}\ddot{\mathbf{r}}_i + {}^{p}\mathbf{R}_i \begin{bmatrix} -\dot{\varphi}_i^2 \ell_{Gi} \\ \ddot{\varphi}_i \ell_{Gi} \end{bmatrix} \end{aligned} \tag{E3.34} $$

En introduisant (E3.34) dans (E3.31), nous obtenons :

$$ {}^{p}\mathbf{R}_i \, {}^{i,p}\mathbf{f}_i - {}^{o,p}\mathbf{f}_{p,i} + m_i \, {}^{o,p}\mathbf{g} = m_i \, {}^{o,p}\ddot{\mathbf{r}}_i + {}^{p}\mathbf{R}_i \, {}^{p,i}\mathbf{p}_i \tag{E3.35} $$

avec ${}^{i,p}\mathbf{p}_i = \begin{bmatrix} -m_i \ell_{Gi} \dot{\varphi}_i^2 \\ m_i \ell_{Gi} \ddot{\varphi}_i \end{bmatrix}$.

Réorganisons les termes de l'équation (E3.35), donnant :

$$ {}^{p}\mathbf{R}_i({}^{i,p}\mathbf{f}_i - {}^{i,p}\mathbf{p}_i) = {}^{o,p}\mathbf{f}_{p,i} + m_i({}^{o,p}\ddot{\mathbf{r}}_i - {}^{o,p}\mathbf{g}) \tag{E3.36} $$

L'équation (E3.36) est similaire à l'équation (E2.11) du modèle général mais un terme ${}^{p}\mathbf{R}_i{}^{i,p}\mathbf{p}_i$ à gauche de l'équation est ajouté aux forces de poussée des drones. Nous pouvons alors utiliser l'équation (E2.17) et écrire :

$$ \mathbf{M}_t\ddot{\mathbf{q}}_p + \mathbf{c}_t = \mathbf{J}_p^T\mathbf{R}_t(\mathbf{f}_p - \mathbf{p}) \tag{E3.37} $$

avec

- $\mathbf{p} = \begin{bmatrix} {}^{1,p}\mathbf{p}_1 \\ {}^{2,p}\mathbf{p}_2 \end{bmatrix}$,
- $\mathbf{f}_p = \begin{bmatrix} {}^{1,p}\mathbf{f}_1 \\ {}^{2,p}\mathbf{f}_2 \end{bmatrix}$ le vecteur des composantes des forces de poussées des drones dans le plan $\mathcal{P}_p$,
- les matrices $\mathbf{M}_t$, $\mathbf{c}_t$, $\mathbf{R}_t$ définies de la même façon que pour (E2.17).

Le modèle dynamique obtenu (E3.36) exprime alors l'accélération des coordonnées de la structure passive en fonction de la poussée des drones. Il n'est relié aux coordonnées d'orientation des drones φ_1 et φ_2 et de leur dérivées par rapport au temps qu'au travers de la matrice $\mathbf{R}_t$ ET du vecteur $\mathbf{p}$.

Le vecteur $\mathbf{p}$, qui comprend les forces centrifuges et de Coriolis associés à la masse des drones, est une difficulté pour la commande de ce robot. Ce terme dépend de la vitesse au carré et de l'accélération angulaire des drones. Il induit un couplage supplémentaire entre la dynamique de la chaîne passive et la dynamique de rotation des drones. Dans le modèle général du chapitre 2, ce couplage n'était pas présent et a permis la conception d'une commande en cascade pour le robot. Dans le cadre du robot à deux drones, ce terme sera négligé pour établir une commande similaire au cas général. Ce terme sera alors considéré comme une perturbation pour le modèle.

Afin de donner une idée de la perturbation engendrée par $\mathbf{p}$, calculons une vitesse ou une accélération nécessaire pour obtenir une force de perturbation de l'ordre de 1 Newton. Prenons $m_i = 1.7$ kg et $\ell_d = 0.27$ m proches des valeurs du prototype réel. La vitesse angulaire nécessaire alors pour obtenir un effet de l'ordre du Newton serait de 1.48 rad.s^{-1} et l'accélération angulaire serait de 2.18 rad.s^{-2}. Garder un effet de perturbation limité implique une plage de fonctionnement réduite des vitesses et accélérations de rotation des drones. Nous avons constaté cependant, en simulation et en expérimentation, que la loi de commande n'était pas déstabilisée par cette perturbation pour les trajectoires envisagées. Il faut noter l'importance de réduire la distance entre l'articulation reliant le drone et la structure passive et son centre de gravité. Si $\ell_{di} = 0$ pour i=1 et 2, la relation $\mathbf{p} = \mathbf{0}$ est vérifiée.

Modèle dynamique des rotations des drones

Dans la partie précédente, nous avons décrit le modèle dynamique des coordonnées de la chaîne passive $\mathbf{q}_p = [y\ z\ \varphi\ q_2]^T$. Dans le cas du modèle général développé dans le chapitre 2, la seconde partie du modèle dynamique était composée de la dynamique de rotation des drones, indépendante de la dynamique de la structure passive. Dans le modèle à deux drones, il existe un bras de levier entre le centre de gravité du drone et l'articulation de liaison. Pour obtenir les équations exprimant le modèle dynamique de la rotation des drones en fonction des moments et forces appliquées, nous allons procéder en deux étapes.

1. Nous allons établir le modèle dynamique du robot complet (drones + architecture passive). Cette équation liera les torseurs d'actions mécaniques $\mathbf{w}_1$ et $\mathbf{w}_2$, produits respectivement par les drones 1 et 2, à l'accélération du vecteur $\mathbf{q}$ des coordonnées du robot.
2. Nous allons ensuite extraire du modèle dynamique du robot complet les équations associées aux coordonnées de rotation des drones : ψ_1, ψ_2, θ_1, θ_2, φ_1 et φ_2 . Elles peuvent se réduire au vecteur $\mathbf{q}_a = [\psi, \theta, \varphi_1, \varphi_2]^T$ du fait des relations imposées par les contraintes cinématiques de la chaîne passive données par (E3.17).

Commençons par écrire le modèle dynamique du robot complet. Nous considérons le Lagrangien du robot :

$$L(\mathbf{q}, \dot{\mathbf{q}}) = E(\dot{\mathbf{q}}, \mathbf{q}) - U(\mathbf{q}) \tag{E3.38}$$

où E représente l'énergie cinétique du robot et U son énergie potentielle. Les équations d'Euler-Lagrange appliquées au robot s'écrivent :

$$\frac{\mathrm{d}}{\mathrm{dt}}\left(\frac{\partial L}{\partial \dot{\mathbf{q}}}\right)^T - \left(\frac{\partial L}{\partial \mathbf{q}}\right)^T = \mathbf{J}^T \begin{bmatrix} {}^1\mathbf{w}_1 \\ {}^2\mathbf{w}_2 \end{bmatrix} \tag{E3.39}$$

avec $\mathbf{J}$ la matrice jacobienne cinématique issue de (E3.24). Cette équation peut être mise sous une forme linéaire en l'accélération des coordonnées $\mathbf{q}$ [Khalil et Dombre, 2002] (voir également annexe A.2) :

$$\mathbf{M}\ddot{\mathbf{q}} + \mathbf{c} = \mathbf{J}^T \begin{bmatrix} {}^1\mathbf{w}_1 \\ {}^2\mathbf{w}_2 \end{bmatrix} \tag{E3.40}$$

avec

- $\mathbf{M}$ une matrice d'inertie généralisée de dimension (9×9) définie positive,
- $\mathbf{c}$ le vecteur des effets gravitationnels, centrifuges et de Coriolis de dimension 9.

Séparons ensuite les efforts liés à la poussée des drones, déjà utilisés en entrée du modèle dynamique de la chaîne passive, des efforts liés aux moments générés par les drones. Pour cela, il est possible d'exprimer $\mathbf{f}_i$ (la force de poussée) et $\mathbf{m}_i$ (le vecteur des moments) du drone $\mathcal{D}_i$ dans son repère local :

$${}^i\mathbf{w}_i = \begin{bmatrix} {}^i\mathbf{f}_i \\ {}^i\mathbf{m}_i \end{bmatrix} \tag{E3.41}$$

Les matrices $\mathbf{J}_v$ et $\mathbf{J}_\omega$ sont définies par l'expression suivante :

$$\mathbf{J}^T \begin{bmatrix} {}^1\mathbf{w}_1 \\ {}^2\mathbf{w}_2 \end{bmatrix} = \mathbf{J}_v^T \begin{bmatrix} {}^1\mathbf{f}_1 \\ {}^2\mathbf{f}_2 \end{bmatrix} + \mathbf{J}_\omega^T \begin{bmatrix} {}^1\mathbf{m}_1 \\ {}^2\mathbf{m}_2 \end{bmatrix} \tag{E3.42}$$

L'expression développée des matrices $\mathbf{J}_v$ et $\mathbf{J}_\omega$ s'obtient par extraction des lignes de la matrice $\mathbf{J}$ (E3.24) :

$$\mathbf{J}_v^T = \begin{bmatrix} S_\psi C_\theta & S_\psi S\theta S_{\varphi_1} + C_\psi C_{\varphi_1} & S_\psi S\theta C_{\varphi_1} - C_\psi S_{\varphi_1} \\ C_\psi C_\theta & C_\psi S\theta S_{\varphi_1} - S_\psi C_{\varphi_1} & C_\psi S\theta C_{\varphi_1} - S_\psi S_{\varphi_1} \\ -S_\theta & C_\theta S_{\varphi_1} & C_\theta C_{\varphi_1} \\ 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix}$$

$$\begin{bmatrix} S_\psi C_\theta & S_\psi S\theta S_{\varphi_2} + C_\psi C_{\varphi_2} & S_\psi S\theta C_{\varphi_2} - C_\psi S_{\varphi_2} \\ C_\psi C_\theta & C_\psi S\theta S_{\varphi_2} - S_\psi C_{\varphi_2} & C_\psi S\theta C_{\varphi_2} - S_\psi S_{\varphi_2} \\ -S_\theta & C_\theta S_{\varphi_2} & C_\theta C_{\varphi_2} \\ (-\ell_1 C_\varphi - \ell_2 C_{12}) C_\theta & (-\ell_1 S_{23} - \ell_2 S_3) S_\theta & (-\ell_1 C_{23} - \ell_2 C_3) S_\theta \\ \ell_1 S_\varphi + \ell_2 S_{12} & 0 & 0 \\ 0 & \ell_1 S_{23} + \ell_2 S_3 & \ell_1 C_{23} + \ell_2 C_3 \\ 0 & \ell_2 S_3 & \ell_2 C_3 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix}$$

$$\mathbf{J}_\omega^T = \begin{bmatrix} 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ -S_\theta & S_{\varphi_1} C_\theta & C_{\varphi_1} C_\theta & -S_\theta & S_{\varphi_2} C_\theta & C_{\varphi_2} C_\theta \\ 0 & C_{\varphi_1} & -S_{\varphi_1} & 0 & C_{\varphi_2} & -S_{\varphi_2} \\ 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 \\ 1 & 0 & 0 & 0 & 0 & 0 \end{bmatrix}$$

En introduisant(E3.42) dans (E3.40) et en isolant le terme d'accélération $\ddot{\mathbf{q}}$, nous avons :

$$\ddot{\mathbf{q}} = \mathbf{M}^{-1} \left(\mathbf{J}_v^T \mathbf{f} + \mathbf{J}_\omega^T \mathbf{m} - \mathbf{c} \right) \tag{E3.43}$$

avec $\mathbf{m}^T = [{}^1\mathbf{m}_1{}^T, {}^2\mathbf{m}_2{}^T]$ et $\mathbf{f}^T = [{}^1\mathbf{f}_1{}^T, {}^2\mathbf{f}_2{}^T]$. Hors singularité de représentation ($\theta = \pm\pi/2$), la matrice $\mathbf{J}_\omega^T$ est de rang 4. Nous cherchons dans l'équation (E3.43) à lier les actionnements indépendants issus de $\mathbf{J}_\omega^T\mathbf{m}$ aux coordonnées $\mathbf{q}_a = [\psi\ \theta\ \varphi_1\ \varphi_2]^T$. En sélectionnant les lignes de l'équation (E3.43) correspondant à ces quatre coordonnées d'orientation, i.e. les lignes 4, 5, 9 et la somme des lignes 6, 7 et 8 ($\varphi_2 = \varphi + q_2 + q_3$), nous obtenons :

$$\ddot{\mathbf{q}}_a = \mathbf{\Psi}\mathbf{M}^{-1}\left(\mathbf{J}_v^T\mathbf{f} + \mathbf{J}_\omega^T\mathbf{m} - \mathbf{c}\right) \tag{E3.44}$$

avec

$$\mathbf{\Psi} = \begin{bmatrix} 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & 0 & 1 & 1 & 1 & 0 \end{bmatrix}$$

L'équation (E3.44) s'écrit sous la forme :

$$\mathbf{A}_m\mathbf{m} = \ddot{\mathbf{q}}_a - \mathbf{\Psi}\mathbf{M}^{-1}\mathbf{J}_v^T\mathbf{f} + \mathbf{\Psi}\mathbf{M}^{-1}\mathbf{c} \tag{E3.45}$$

avec $\mathbf{A}_m = \mathbf{\Psi}\mathbf{M}^{-1}\mathbf{J}_\omega^T$. $\mathbf{A}_m$ est de dimension (4 x 6). L'équation (E3.45) lie l'accélération des coordonnées d'orientation $\ddot{\mathbf{q}}_a$ aux moments $\mathbf{m}$ générés par le drone. Pour la commande, nous cherchons à exprimer cette équation sous la forme d'un modèle dynamique inverse, i.e. résoudre cette équation en $\mathbf{m}$. Pour que l'équation (E3.45) admette une solution en $\mathbf{m}$, il est nécessaire que la matrice $\mathbf{A}_m$ soit de rang plein.

Pour étudier ce rang, remarquons d'abord que la matrice $\mathbf{J}_\omega^T$ peut se décomposer sous la forme :

$$\mathbf{J}_\omega^T = \mathbf{\Psi}^T\mathbf{J}_r^T \tag{E3.46}$$

avec

$$\mathbf{J}_r^T = \begin{bmatrix} -S_\theta & S_{\varphi_1}C_\theta & C_{\varphi_1}C_\theta & -S_\theta & S_{\varphi_2}C_\theta & C_{\varphi_2}C_\theta \\ 0 & C_{\varphi_1} & -S_{\varphi_1} & 0 & C_{\varphi_2} & -S_{\varphi_2} \\ 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 \end{bmatrix}$$

Ainsi avec $\mathbf{H} = \mathbf{\Psi}\mathbf{M}^{-1}\mathbf{\Psi}^T$, la matrice $\mathbf{A}_m$ s'écrit :

$$\mathbf{A}_m = \mathbf{\Psi}\mathbf{M}^{-1}\mathbf{\Psi}^T\mathbf{J}_r^T = \mathbf{H}\mathbf{J}_r^T \tag{E3.47}$$

Or, la matrice $\mathbf{H}$ est une matrice (4×4) symétrique définie positive et la matrice $\mathbf{J}_r^T$ est une matrice (4×6) de rang 4. Ainsi, la matrice $\mathbf{A}_m$ est une matrice (4×6) de rang 4. L'équation (E3.45) possède au moins une solution pour l'inconnue $\mathbf{m}$. La matrice $\mathbf{A}_m$ possède deux colonnes supplémentaires à son rang. Ces deux colonnes correspondent à un sur-actionnement des moments générés par les drones. Ce sur-actionnement est provoqué par la contrainte cinématique de la chaîne passive. Ainsi, les moments du drone $\mathcal{D}_1$ selon $\mathbf{y}_1$ et $\mathbf{z}_1$

sont mécaniquement liés aux moments du drone $\mathcal{D}_2$ selon $\mathbf{y}_2$ et $\mathbf{z}_2$. Une solution minimisant la norme du vecteur des moments générés est obtenue par l'usage de la pseudo-inverse $\mathbf{A}_m^+$:

$$\mathbf{m} = \mathbf{A}_m^+ \left(\ddot{\mathbf{q}}_a - \mathbf{\Psi} \mathbf{M}^{-1} \mathbf{J}_v^T \mathbf{f} + \mathbf{\Psi} \mathbf{M}^{-1} \mathbf{c} \right) \tag{E3.48}$$

avec $\mathbf{A}_m^+ = \mathbf{A}_m^T (\mathbf{A}_m \mathbf{A}_m^T)^{-1}$.
L'équation (E3.48) est le modèle dynamique inverse exprimant les moments $\mathbf{m}$ à fournir par les drones pour produire une accélération des coordonnées de rotation $\ddot{\mathbf{q}}_a$ avec les forces de poussées des drones $\mathbf{f}$ imposées.

Bilan des modèles dynamiques

Le modèle dynamique en fonction du vecteur de coordonnées $\mathbf{q}_p$ et celui en fonction du vecteur $\mathbf{q}_a$ ont été établis par les équations (E3.37) et (E3.48) que nous rappelons ici :

$$\mathbf{M}_t \ddot{\mathbf{q}}_p + \mathbf{c}_t = \mathbf{J}_p^T \mathbf{R}_t (\mathbf{f} - \mathbf{p})$$

$$\mathbf{A}_m^+ \left(\ddot{\mathbf{q}}_a - \mathbf{\Psi} \mathbf{M}^{-1} \mathbf{J}_v^T \mathbf{f} + \mathbf{\Psi} \mathbf{M}^{-1} \mathbf{c} \right) = \mathbf{m}$$

Les hypothèse sur la conception du robot étant moins fortes que pour le modèle à n drones (voir partie 2.3.3), ces deux équations sont couplées de façon plus complexe qu'au chapitre 2.

Cependant, en négligeant le terme $\mathbf{p}$, l'équation de la dynamique de la structure passive (E3.37) ne dépend des coordonnées d'orientation $\mathbf{q}_a$ que via la matrice $\mathbf{R}_t$ et l'équation redevient totalement similaire à l'équation (E2.17).

L'équation de la dynamique des orientations des drones (E3.48) diffère de la forme de l'équation (E2.22) par la présence du terme dépendant des forces de poussées $-\mathbf{\Psi} \mathbf{M}^{-1} \mathbf{J}_v^T \mathbf{f}$. Ce terme additionnel est la conséquence du bras de levier entre le centre de masse du drone et le point d'application des forces d'interaction avec l'architecture passive. Cette nouvelle équation pourra être intégrée dans la commande en cascade. En effet, le modèle dynamique inverse des coordonnées orientations est utilisé pour établir la commande de la boucle interne (rapide). Or, le calcul des forces de poussées est lui effectué par la boucle externe (lente) (Fig. 3.18). Celle-ci pourra alors fournir à la boucle interne les valeurs de forces de poussées imposées pour le calcul du modèle dynamique inverse des coordonnées d'orientation.

Il est alors à nouveau possible d'établir une loi de commande en cascade de façon similaire à celle proposée dans la partie 2.4. Les spécificités de cette loi de commande seront données dans la sous-partie suivante.

Pour compléter le modèle dynamique, il faut exprimer les forces $\mathbf{f}$ et moments $\mathbf{m}$ en fonction de la vitesse de rotation des hélices. Considérons le modèle classique d'un quadricoptère

développé dans la partie 1.1, tel que :

$$ {}^{i}\mathbf{w}_i = \mathbf{Q}_i \mathbf{\Omega_i} \tag{E3.49} $$

$\mathbf{Q}_i$ sont les matrices de répartition de efforts générés par les hélices. Contrairement au cas du drone seul, le torseur des efforts n'est pas exprimé au centre du drone mais à l'origine du repère local $\mathcal{F}_i$. Les matrices $\mathbf{Q}_i$ sont alors données par :

$$ \mathbf{Q}_1 = k_{t,1} \begin{bmatrix} 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ -1 & -1 & -1 & -1 \\ -\ell_{d1} - \frac{d_1}{2} & -\ell_{d1} + \frac{d_1}{2} & -\ell_{d1} - \frac{d_1}{2} & -\ell_{d1} + \frac{d_1}{2} \\ -\frac{d_1}{2} & \frac{d_1}{2} & \frac{d_1}{2} & -\frac{d_1}{2} \\ -\frac{k_{d,1}}{k_{t,1}} & -\frac{k_{d,1}}{k_{t,1}} & \frac{k_{d,1}}{k_{t,1}} & \frac{k_{d,1}}{k_{t,1}} \end{bmatrix} $$

$$ \mathbf{Q}_2 = k_{t,2} \begin{bmatrix} 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 1 & 1 & 1 & 1 \\ \ell_{d,2} - \frac{d_2}{2} & \ell_{d,2} + \frac{d_2}{2} & \ell_{d,2} - \frac{d_2}{2} & \ell_{d,2} + \frac{d_2}{2} \\ \frac{d_2}{2} & -\frac{d_2}{2} & -\frac{d_2}{2} & \frac{d_2}{2} \\ \frac{k_{d,2}}{k_{t,2}} & \frac{k_{d,2}}{k_{t,2}} & -\frac{k_{d,2}}{k_{t,2}} & -\frac{k_{d,2}}{k_{t,2}} \end{bmatrix} $$

Comme pour le modèle général établit dans la partie 2.3.3, les entrées du modèle dynamique peuvent se réduire aux forces de poussées selon $\mathbf{z}_i$ pour chaque drone i et aux moments du vecteur $\mathbf{m}$. L'équation (E3.37) s'écrit alors (le vecteur $\mathbf{p}$ étant négligé) :

$$ \mathbf{M}_t \ddot{\mathbf{q}}_p + \mathbf{c}_t = \mathbf{J}_p^T \mathbf{R}_r \mathbf{f}_z \tag{E3.50} $$

avec

- $\mathbf{f}_z^T = [f_{z,1}, f_{z,2}]^T$,
- $\mathbf{R}_r$ contenant la deuxième et la quatrième colonne de $\mathbf{R}_t$: $\mathbf{R}_r = \begin{bmatrix} -\sin\varphi_1 & 0 \\ \cos\varphi_1 & 0 \\ 0 & -\sin\varphi_2 \\ 0 & \cos\varphi_2 \end{bmatrix}$.

Modèle dynamique selon l'axe $\mathbf{x}_o$

Précédemment dans cette partie, les modèles dynamiques ont été établis pour les coordonnées $\mathbf{q}_p = [y\ z\ \varphi\ q_2]^T$ et les coordonnées $\mathbf{q}_a = [\psi\ \theta\ \varphi_1\ \varphi_2]^T$. Les coordonnées généralisées du robot $\mathbf{q}$ sont au nombre de 9. En effet les modèles précédents ne décrivent pas la dynamique de la coordonnée x. Sous la contrainte $\psi = 0$ et $\theta = 0$, aucune force n'est produite

par le robot dans la direction $\mathbf{x}_o$ et le modèle dynamique de la coordonnée x s'écrit :

$$\ddot{x} = 0 \tag{E3.51}$$

Cette équation nous convient sur le plan théorique car nous souhaitons maintenir $x = 0$. Cependant, des perturbations peuvent faire évoluer la coordonnée x. Dans le cadre réel des expérimentations, il faudra prévoir une commande spécifique pour cette coordonnée. Pour cela, nous allons relâcher la contrainte $\theta = 0$. Avec $|\theta| << 1$ la dynamique de la translation selon $\mathbf{x}_o$ du centre de gravité du robot se déduit en projetant les forces $\mathbf{f}_1^1 = [0\ 0\ f_{z,1}]^T$ et $\mathbf{f}_2^2 = [0\ 0\ f_{z,2}]$ sur l'axe $\mathbf{x}_o$. Elle s'écrit :

$$\begin{aligned} m_t \ddot{x}_G &= \sin\theta(\cos\varphi_1 f_{z,1} + \cos\varphi_2 f_{z,2}) \\ &\simeq \theta(\cos\varphi_1 f_{z,1} + \cos\varphi_2 f_{z,2}) \end{aligned} \tag{E3.52}$$

avec m_t la masse totale du robot et x_G la position de son centre de gravité selon $\mathbf{x}_o$. Ce modèle dynamique latéral sera exploité dans le cadre de la commande de la coordonnée x_G, décrit dans la prochaine partie.

Nous avons ainsi établi les modèles dynamiques qui permettrons de concevoir la loi de commande du robot. Les particularités de cette loi de commande, notamment les différences avec le cas du chapitre 2, seront exposées dans la prochaine partie.

3.2.5 Loi de commande

La commande du robot volant à deux drones s'établit de façon similaire à celle du cas n drones 2.4. Il existe cependant quelques différences liées aux particularités de ce robot. La figure 3.18 présente la loi de commande pour le robot volant à deux drones. Les différences principales avec la loi de commande présenté au chapitre 2 sont :

- le calcul des forces de poussées et des angles désirés, qui s'effectue dans le plan vertical ;
- la commande de la coordonnée x, spécifique à ce robot pour effectuer une correction de la position selon l'axe $\mathbf{x}$, le reste de la commande n'intégrant pas cette coordonnée ;
- la commande de l'orientation des drones, qui diffère du modèle du chapitre 2 du fait de l'expression du modèle dynamique de rotation des drones spécifique à ce robot ;
- la commande dans le système des coordonnées du drone qui établit un suivi de trajectoire exprimé dans les coordonnées de position des drones plutôt que dans les coordonnées $\mathbf{q}_p$ de l'architecture passive.

Calcul de la poussée et des angles désirés des drones

Le calcul de la poussée des drones et des angles désirés, développé en partie 2.4.1 pour le cas à n drones, diffère légèrement dans le cas du robot volant à deux drones. Rappelons que le

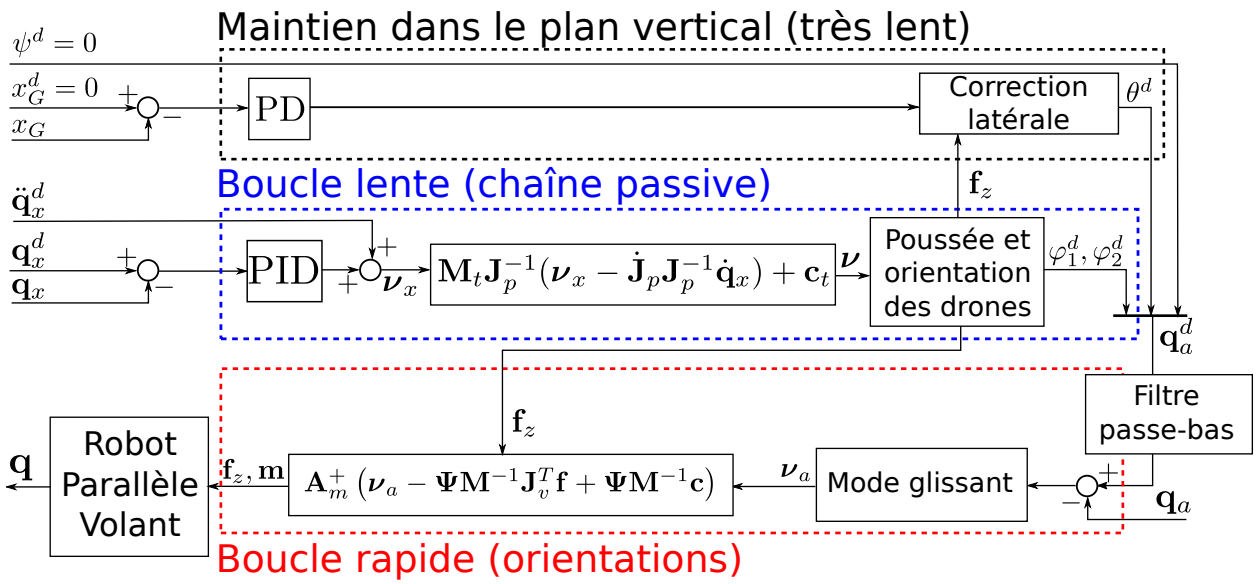


FIGURE 3.18 – Schéma de la commande du robot volant à deux drones.

calcul de la boucle lente s'effectue dans un plan vertical. Les forces désirées à chaque extrémité de la chaîne passive ne comprennent que deux composantes. L'expression développée du système d'équations (E2.30) est alors (sous l'hypothèse $\theta_i \simeq 0, i = 1, 2$) :

$$\begin{aligned} -\sin\varphi_i f_{z,i} &= \nu_{J,2i-1} \\ \cos\varphi_i f_{z,i} &= \nu_{J,2i} \end{aligned} \tag{E3.53}$$

Alors, la solution du système (E3.53) pour le drone 1 est telle que $f_{z,1} < 0$ (l'axe $\mathbf{z}_1$ est dans le sens opposé à la poussée du drone) et pour le drone 2 telle que $f_{z,2} > 0$:

$$\begin{aligned} f_{z,1} &= -\sqrt{\nu_{J,1}^2 + \nu_{J,2}^2} \\ \varphi_1 &= \text{atan2}(\nu_{J,1}, -\nu_{J,2}) \\ f_{z,2} &= \sqrt{\nu_{J,3}^2 + \nu_{J,4}^2} \\ \varphi_2 &= \text{atan2}(\nu_{J,3}, \nu_{J,4}) \end{aligned} \tag{E3.54}$$

Commande de la coordonnée x

Une commande spécifique additionnelle est établie pour maintenir le robot selon la coordonnée x souhaitée. Cette commande est basée sur la commande de la position du centre de gravité du robot x_G dont le modèle dynamique est exprimé par (E3.52). La variable θ est alors utilisée comme une commande auxiliaire pour converger vers $x_G = 0$. L'angle θ est

choisi pour obtenir l'équation du système en boucle fermée suivante :

$$\ddot{x}_G + k_{p,x} x_G + k_{d,x} \dot{x}_G = 0 \tag{E3.55}$$

avec les gains $k_{p,x} > 0$ et $k_{d,x} > 0$. En introduisant (E3.55) dans (E3.52), nous obtenons l'angle θ désiré :

$$\theta^d = \frac{-k_{p,x} x_G - k_{d,x} \dot{x_G}}{m_t(\cos\varphi_1 f_{z,1} + \cos\varphi_2 f_{z,2})} \tag{E3.56}$$

Les efforts de poussée pour le calcul de cette commande sont ceux obtenus par (E3.54). Pour que l'hypothèse de maintien du robot dans un plan vertical reste valide, les gains $k_{p,x}$ et $k_{d,x}$ doivent rester faibles. Ainsi l'évolution de θ sera suffisamment lente pour ne pas affecter le modèle dynamique de la chaîne passive établi sous hypothèse $\theta << 1$. Cette loi de commande n'a pas pour but de faire suivre une trajectoire à la coordonnée x mais uniquement de contrer d'éventuelles perturbations le long de l'axe $\mathbf{x}_o$. La fréquence de coupure de cette boucle de commande sera donc choisie plus lente que la commande de la chaîne passive.

Commande de l'orientation des drones

Le modèle dynamique établi pour les coordonnées d'orientation (3.2.4) diffère de celui du modèle général (E2.22). Cependant, il reste possible d'établir une loi de commande par mode glissant en définissant la commande auxiliaire $\boldsymbol{\nu}_a$ comme dans le cas du modèle général (E2.41). Cette commande auxiliaire sur l'accélération des coordonnées d'orientation $\mathbf{q}_a$ peut être introduite dans le modèle dynamique inverse (E3.48), donnant alors :

$$\mathbf{m} = \mathbf{A}_m^+ \left(\boldsymbol{\nu}_a - \boldsymbol{\Psi}\mathbf{M}^{-1}\mathbf{J}_v^T\mathbf{f} + \boldsymbol{\Psi}\mathbf{M}^{-1}\mathbf{c} \right) \tag{E3.57}$$

Les forces de poussées $\mathbf{f}$ nécessaires au calcul de cette loi de commande sont celles fournies par le calcul de la commande de la chaîne passive.

Commande de la boucle lente dans le système de coordonnées des drones

En conditions réelles d'expérimentations (voir annexe B.2), les mesures disponibles sont la position, l'orientation et le torseur cinématique de chaque drone. Considérons (au changement de repère près) que nous disposons de la position des points O_1 et O_3 dans le repère $\mathcal{F}_o$ données respectivement par les vecteurs ${}^o\mathbf{r}_1$ et ${}^o\mathbf{r}_2$, ainsi que des coordonnées d'orientation des drones et des torseurs cinématiques des drones ${}^1\mathbf{t}_1$ et ${}^2\mathbf{t}_2$, exprimés au point O_1 et O_3.

Partant de ces données disponibles, les parties 3.2.2 et 3.2.3 indiquent, au travers des modèles géométriques et cinématiques, les équations nécessaires de calcul du vecteur des coordonnées généralisées $\mathbf{q}$ et de sa dérivée $\dot{\mathbf{q}}$. Une fois ces vecteurs obtenus, le calcul des modèles dynamiques devient possible.

Pour la boucle lente, nous pouvons établir une loi de commande basée directement sur les

coordonnées de position des drones dans le plan vertical ${}^{o,p}\mathbf{r}_1$ et ${}^{o,p}\mathbf{r}_2$. Soit $\mathbf{q}_x^T = [{}^{o,p}\mathbf{r}_1^T \ {}^{o,p}\mathbf{r}_2^T]$ le vecteur des coordonnées des drones dans le plan vertical. L'équation (E3.28) établit la relation entre la dérivée de $\mathbf{q}_x$ et $\mathbf{q}_p$:

$$\dot{\mathbf{q}}_x = \mathbf{J}_p \dot{\mathbf{q}}_p \tag{E3.58}$$

Nous pouvons alors injecter cette relation et sa dérivée dans le modèle dynamique de la chaîne passive (E3.50) :

$$\mathbf{M}_t \mathbf{J}_p^{-1}(\ddot{\mathbf{q}}_x - \dot{\mathbf{J}}_p \mathbf{J}_p^{-1} \dot{\mathbf{q}}_x) + \mathbf{c}_t = \mathbf{J}_p^T \mathbf{R}_r \mathbf{f}_z \tag{E3.59}$$

Ce modèle dynamique ne dépend plus des coordonnées $\mathbf{q}_p$ de la chaîne passive mais des coordonnées $\mathbf{q}_x$ des drones. La modification de la commande passe alors par la modification de la commande auxiliaire $\boldsymbol{\nu}$ (E2.35). Définissons une nouvelle commande auxiliaire $\boldsymbol{\nu}_x$ par :

$$\boldsymbol{\nu}_x = \ddot{\mathbf{q}}_x^d - \mathbf{K}_p \mathbf{e}_x - \mathbf{K}_d \dot{\mathbf{e}}_x - \mathbf{K}_i \int \mathbf{e}_x \tag{E3.60}$$

Le calcul de la commande auxiliaire $\boldsymbol{\nu}$ (E2.36) est alors remplacé par :

$$\boldsymbol{\nu} = \mathbf{M}_t \mathbf{J}_p^{-1}(\boldsymbol{\nu}_x - \dot{\mathbf{J}}_p \mathbf{J}_p^{-1} \dot{\mathbf{q}}_x) + \mathbf{c}_t \tag{E3.61}$$

Les autres boucles ne sont pas impactées par la modification de la boucle lente.

Nous avons ainsi listé les modifications nécessaires de la loi de commande pour le robot volant à deux drones. Nous allons maintenant expliquer pourquoi il a été nécessaire d'ajouter une articulation pivot supplémentaire au robot et les conséquences que cela engendre sur la loi de commande.

3.2.6 Gestion de la commande latérale et ajout d'un pivot

La commande établie dans la partie 3.2.5 s'est avérée efficace dans le cadre des premières simulations. Cependant, en conditions réelles d'expérimentation, ou avec des conditions initiales de simulation moins idéales, il a été constaté une limitation de la commande en cas de forte déviation selon l'angle ψ. En effet, dans le cas d'un drone seul, cet angle est commandé par le couple selon l'axe vertical du drone. Celui-ci est généré par la traînée des hélices et s'écrit $\mathbf{m}_z = k_d(-\omega_1^2 - \omega_2^2 + \omega_3^2 + \omega_4^2)\mathbf{z}_d$. L'ordre de grandeur du coefficient k_d est $5e-7$ N.m.s^2. Ainsi, le couple maximal généré selon son axe vertical par un drone comme le Pélican est d'environ 1.5 N.m. Il suffit à commander l'angle selon cet axe car l'inertie du drone est également faible (0.007 kg.m^2).

Faisons maintenant un calcul approché de l'inertie suivant le même axe du robot volant à deux drones, toujours contraint dans le plan vertical. La majorité de la masse du robot est située au niveau des drones. Nous pouvons établir un modèle très simplifié du robot comme

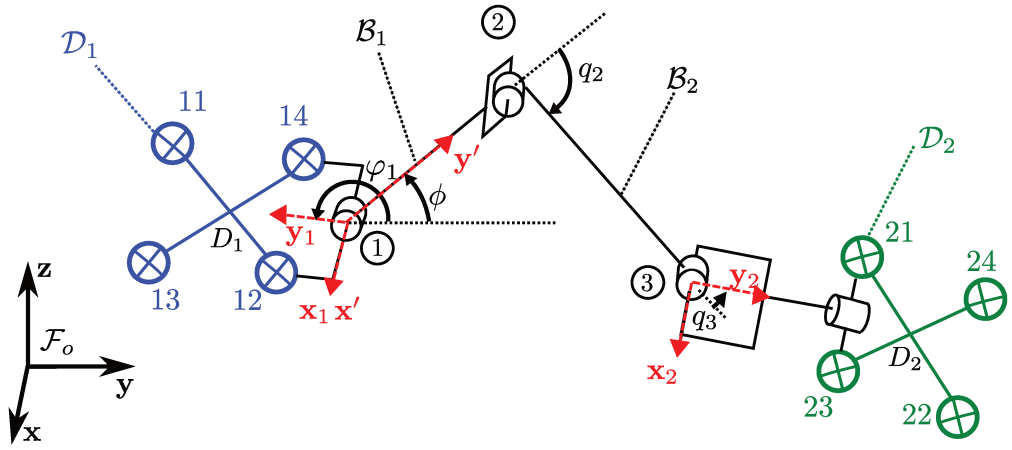


FIGURE 3.19 – Schéma cinématique du robot parallèle volant à deux drones muni d'un pivot supplémentaire.

deux masses ponctuelles situées au niveau du centre de gravité de chaque drone contenant chacune la moitié de la masse totale m_t du robot. La position du centre de gravité du robot selon l'axe $\mathbf{y}_o$ s'écrit :

$$y_{G,z} = \frac{y_{G_1} + y_{G_2}}{2} \tag{E3.62}$$

avec y_{G_1} et y_{G_2} les positions respectives selon l'axe $\mathbf{y}_o$ des centres de gravité des drones $\mathcal{D}_1$ et $\mathcal{D}_2$. L'inertie selon l'axe $\mathbf{z}$, calculée au centre de gravité, est donnée alors par (avec $x_{G_1} \simeq x_{G_2} \simeq 0$) :

$$I_{zz} \simeq \frac{m_t(y_{G_2} - y_{G_1})^2}{4} \tag{E3.63}$$

Avec une distance entre les deux drones de 1 m et une masse totale du robot de l'ordre de 4 kg, l'inertie obtenue est 1 kg/m^2, soit plus de cent fois plus élevée que dans le cas du drone seul. Le moment disponible pour agir selon cet axe est, quand à lui, la somme des moments de chaque drone, soit au maximum 3 N.m. Les actionneurs vont alors vite saturer en tentant de commander la variable ψ.

D'un autre côté, nous avons constaté dans la partie 3.2.4 que la commande des rotations du système est sur-actionnée. Il est donc possible de commander au moins un degré de liberté supplémentaire avec l'actionnement disponible. Nous ajoutons une articulation pivot au niveau de la liaison entre la chaîne passive et le drone $\mathcal{D}_2$. Cette articulation supplémentaire libère la coordonnée d'orientation θ_2 de ce drone (voir Fig. 3.19), qui sera exploitée pour commander, en cascade, la coordonnée d'orientation ψ.

Modèle dynamique latéral simplifié

L'ajout de l'articulation pivot modifie la conception de la commande et notamment de la loi de commande des variables x et ψ. En effet, nous allons agir sur ces deux variables en cascade par la modification des inclinaisons θ_1 et θ_2 des drones. Dans la loi de commande,

nous n'utiliserons plus les moments des traînées des drones, qui seront donc fixés à zéro. Nous allons établir un nouveau modèle dynamique pour la commande latérale à partir des forces latérales produites par les drones. Considérons les composantes selon $\mathbf{x}_o$ des forces produites par les drones : ${}^o f_{x,1}$ et ${}^o f_{x,2}$. L'évolution de la position du centre de gravité du robot sous l'influence de ces forces s'écrit :

$$m_t \ddot{x}_G = {}^o f_{x,1} + {}^o f_{x,2} \tag{E3.64}$$

Le moment selon l'axe $\mathbf{z}$ est principalement produit par ces forces. Les moments produits par les drones sont négligeables car $\theta_1 \simeq \theta_2 \simeq 0$ et les moments verticaux des drones dans leurs repères locaux ont été fixés à zéro. En reprenant le modèle simplifié à deux masses ponctuelles, écrivons le moment selon $\mathbf{z}$ appliqué au centre de gravité du robot :

$$m_z = ({}^o f_{x,1} - {}^o f_{x,2}) \frac{y_{G_2} - y_{G_1}}{2} \tag{E3.65}$$

En tenant compte de l'inertie calculée par (E3.63), on obtenons :

$$m_z = I_{zz} \ddot{\psi} \simeq \frac{m_t (y_{G_2} - y_{G_1})^2}{4} \ddot{\psi} \tag{E3.66}$$

En insérant (E3.65) dans (E3.66), nous avons :

$$\frac{m_t (y_{G_2} - y_{G_1})}{2} \ddot{\psi} \simeq {}^o f_{x,1} - {}^o f_{x,2} \tag{E3.67}$$

Les équations (E3.64) et (E3.67) expriment une relation entre l'accélération des coordonnées ψ et x_G d'une part et les forces latérales ${}^o f_{x,1}$ et ${}^o f_{x,2}$ produites par les drones d'autre part. Ainsi, ces forces latérales pourront être utilisées comme des commandes auxiliaires pour la commande des variables ψ et x_G.

Modification de la commande

Afin de tenir compte de l'ajout du pivot pour la commande latérale sur le robot, la loi de commande est modifiée de la façon suivante.

Deux commandes auxiliaires ν_x et ν_ψ sont établies sur les coordonnées respectives x et ψ. Ces deux lois de commande sont des lois PD à faibles gains. Les forces latérales ${}^o f_{x,1}$ et ${}^o f_{x,2}$ sont alors calculées à partir du modèle dynamique donné par les équations (E3.64) et (E3.67) :

$$\begin{aligned} {}^o f_{x,1} &= m_t \left(\frac{\nu_x}{2} + \frac{y_{G_2} - y_{G_1}}{4} \nu_\psi \right) \\ {}^o f_{x,2} &= m_t \left(\frac{\nu_x}{2} - \frac{y_{G_2} - y_{G_1}}{4} \nu_\psi \right) \end{aligned} \tag{E3.68}$$

En levant l'hypothèse $\theta_i \simeq 0, i = 1, 2$, le système (E3.53), à résoudre pour obtenir la poussée

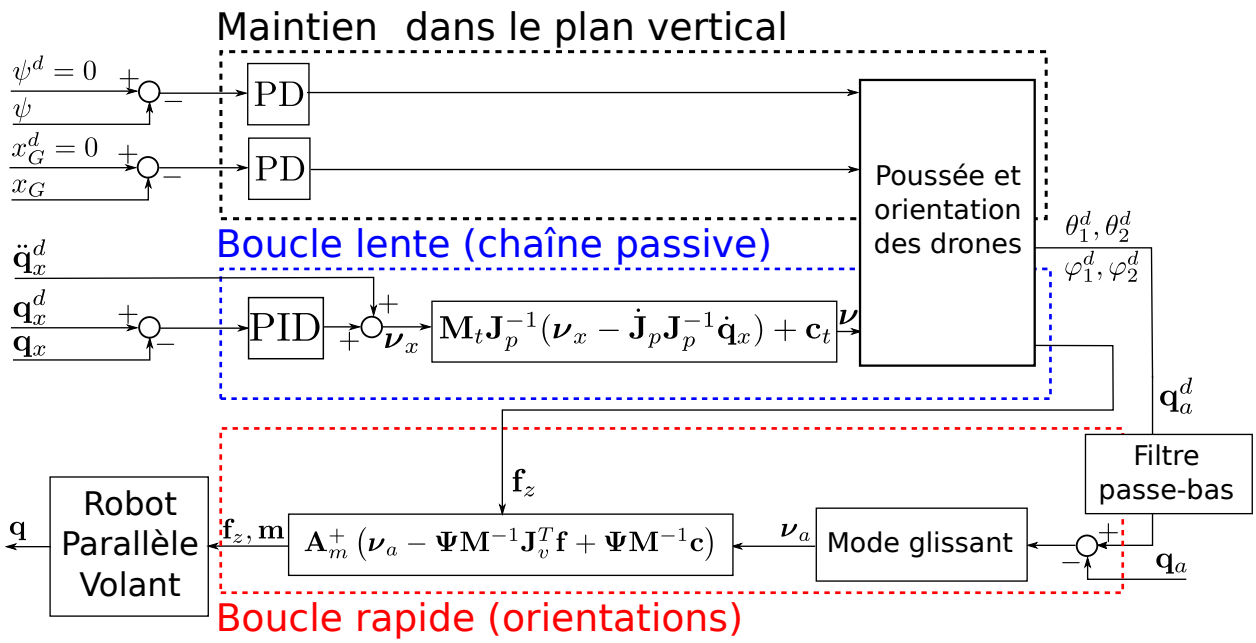


FIGURE 3.20 – Schéma de la commande du robot volant à deux drones avec liaison pivot supplémentaire.

et l'orientation du drone nécessaire à l'obtention les force désirées, s'écrit :

$$\begin{aligned} \cos\varphi_i \sin\theta_i f_{z,i} &= {}^o f_{x,i} \\ -\sin\varphi_i f_{z,i} &= \nu_{J,2i-1} \\ \cos\varphi_i \cos\theta_i f_{z,i} &= \nu_{J,2i} \end{aligned} \tag{E3.69}$$

La première ligne de l'équation (E3.69) permet la commande latérale, tandis que la seconde et la troisième ligne sont issues de la boucle de commande de la chaîne passive. Le système (E3.69) se résout de la façon suivante, en fonction du signe de la force de poussée :

$$\begin{aligned} f_{z,1} &= -\sqrt{\nu_{J,1}^2 + \nu_{J,2}^2 + {}^o f_{x,1}{}^2} \\ \varphi_1 &= \mathrm{asin}(-\nu_{J,1}/f_{z,1}) + \pi \\ \theta_1 &= \mathrm{atan2}({}^o f_{x,1}, \nu_{J,2}) \\ f_{z,2} &= \sqrt{\nu_3^2 + \nu_4^2 + {}^o f_{x,2}{}^2} \\ \varphi_2 &= \mathrm{asin}(-\nu_{J,3}/f_{z,2}) \\ \theta_2 &= \mathrm{atan2}({}^o f_{x,2}, \nu_{J,4}) \end{aligned} \tag{E3.70}$$

Le schéma de la commande du robot muni de la liaison pivot supplémentaire est donné Fig. 3.20. Les modifications par rapport à la commande sans cette articulation pivot sont :

- une nouvelle loi PD pour la commande auxiliaire sur ψ,

- un nouveau calcul des orientations désirées des drones (E3.70),
- un nouveau modèle dynamique des orientations intégrant θ_2 et retirant ψ dans le vecteur des coordonnées d'orientation $\mathbf{q}_a$. Le principe de calcul de ce modèle est le même que celui développé sans la liaison pivot supplémentaire (E3.48).

Notons que la commande latérale développée pour la commande de ψ (E3.68) ne fonctionne plus si le bras de levier entre les deux drones devient trop faible. Dans ce cas, l'inertie selon l'axe $\mathbf{z}$ du robot est faible et le robot pourra être commandé à l'aide des moments obtenus par la traînée des hélices comme initialement prévu. θ_2^d sera alors fixé à zéro. La commande du robot prévoit une transition continue entre les deux modes de commande latérale suivant la distance horizontale entre le centre de gravité du drone $\mathcal{D}_1$ et celui du drone $\mathcal{D}_2$.

Nous allons maintenant présenter les résultats obtenus en simulation pour la nouvelle loi de commande appliquée au robot parallèle volant à deux drones avec l'articulation pivot supplémentaire.

3.2.7 Simulation

Réalisation de la simulation

La simulation pour ce robot est effectuée avec les mêmes logiciels et dans les mêmes conditions (temps d'échantillonnage, bruit) que pour le robot à trois drones dans la partie 3.1.3. Dans le cadre de cette simulation, cependant, nous nous sommes approchés des entrées réelles mesurées dans le cadre expérimental. Les mesures disponibles en expérimentation (voir annexe B.2) sont la position et l'orientation de chaque drone, ainsi que leurs vitesses linéaires et angulaires. Ces mêmes mesures sont fournies par ADAMS à la commande. Une mesure supplémentaire de q_2 est également fournie pour indiquer le mode d'assemblage du robot, afin de déterminer la solution du modèle géométrique. Une solution vérifie $q_2 > 0$ et l'autre $q_2 < 0$. Cette mesure sera également mise en œuvre dans le cadre expérimental.

Comme indiqué dans la partie 3.2.5, la trajectoire déterminée et la commande s'effectuent sur le vecteur des coordonnées des drones $\mathbf{q}_x$. Ces conditions de simulation sont les plus proches possible des conditions de l'expérimentation réelle.

Paramètres physiques du robot

Les paramètres physiques du robot pour cette simulation sont ceux issus de la CAO du robot. Les distances caractéristiques sont

- la longueur des bras ℓ_1=ℓ_2=0.856 m,
- la distance entre les centres de gravité des drones et leurs articulations respectives $\ell_{d1} = 0.27365$ m et $\ell_{d2} = 0.27565$ m,
- $d = 0.3$ m la distance entre deux hélices d'un drone.

Les paramètres dynamiques de chaque corps $\mathcal{B}$ sont donnés par

	$\mathcal{B}_1$	$\mathcal{B}_2$	$\mathcal{D}_1$	$\mathcal{D}_2$
$m_{\mathcal{B}}$ (kg)	2.34e-01	2.18e-01	1.68e+00	1.71e+00
$MX_{\mathcal{B}}$ (kg.m)	1.09e-01	9.92e-02	4.21e-01	1.11e-03
$MY_{\mathcal{B}}$ (kg.m)	0.00e+00	0.00e+00	3.44e-03	0.00e+00
$MZ_{\mathcal{B}}$ (kg.m)	0.00e+00	0.00e+00	3.24e-03	4.45e-01
$XX_{\mathcal{B}}$ (kg.m^2)	2.68e-05	2.69e-05	1.61e-02	1.16e-01
$YY_{\mathcal{B}}$ (kg.m^2)	6.86e-02	6.63e-02	1.06e-01	1.34e-01
$ZZ_{\mathcal{B}}$ (kg.m^2)	6.86e-02	6.63e-02	1.25e-01	1.50e-02
$XY_{\mathcal{B}}$ (kg.m^2)	0.00e+00	0.00e+00	-8.66e-04	0.00e+00
$XZ_{\mathcal{B}}$ (kg.m^2)	0.00e+00	0.00e+00	-8.13e-04	-2.88e-04
$YZ_{\mathcal{B}}$ (kg.m^2)	0.00e+00	0.00e+00	-6.65e-06	0.00e+00

TABLE 3.5 – Tableau des paramètres dynamique du robot volant à deux drones (dans les repères locaux définis par la convention DHKK).

- la masse $m_{\mathcal{B}}$,
- le premier moment d'inertie $\mathbf{M}_{\mathcal{B}} = [MX_{\mathcal{B}}\ MY_{\mathcal{B}}\ MZ_{\mathcal{B}}]$,
- la matrice d'inertie $\mathbf{I}_{\mathcal{B}}$ symétrique à coefficients $XX_{\mathcal{B}}$, $YY_{\mathcal{B}}$, $ZZ_{\mathcal{B}}$, $XY_{\mathcal{B}}$, $XZ_{\mathcal{B}}$ et $YZ_{\mathcal{B}}$

Ces paramètres sont exprimés dans le repère de DHKK associé à chaque corps. Leurs valeurs sont indiquées dans le tableau (T3.5) Les coefficients de poussée et de traînée sont respectivement donnés par $k_t = 1.5\text{e}-05$ N.rad $^{-2}$ et $k_d = 5\text{e}-7$ N.m.rad $^{-2}$.

Choix des gains et coefficients des filtres

Le gains de la boucle rapide et de la boucle lente sont définis de façon similaire à la simulation précédente (voir partie 3.1.3). Des gains faibles sont utilisés pour le maintien dans le plan latéral, afin que l'hypothèse $\theta_1 \simeq 0$ et $\theta_2 \simeq 0$ reste valide. Les valeurs des gains de simulation sont :

- pour la coordonnée x_G (section 3.2.6) : $k_{p,x} = 1$, $k_{d,x} = 2$;
- pour la coordonnée ψ (section 3.2.6) : $k_{p,\psi} = 2$, $k_{d,\psi} = 3$;
- pour la boucle lente (commande adaptative partie 2.4.5) : $\lambda_p = 2$, $\lambda_i = 0.5$, $k_\sigma = 3$ et $\gamma = 0.01$;
- pour la boucle rapide (section 2.4.3) : les matrices de gains ont été prévues diagonales à coefficients positifs pour permettre un ajustement plus précis en fonction de la variable commandée. Dans le cadre de la simulation, il n'a pas été nécessaire de choisir des gains différents en fonction des orientations commandées. Les coefficients diagonaux de Λ, Λ_i, $\mathbf{K}_s$ sont respectivement 10, 2 et 20. $\epsilon_k = 1.33$ est le même pour chaque variable commandée.

Le filtre sur les orientations désirées θ_1^d et θ_2^d est un filtre passe-bas de type Butterworth d'ordre 4 à fréquence de coupure 1 Hz. Le filtre sur les orientations désirées φ_1^d et φ_2^d est un filtre passe-bas de type Butterworth d'ordre 3 à fréquence de coupure 3 Hz.

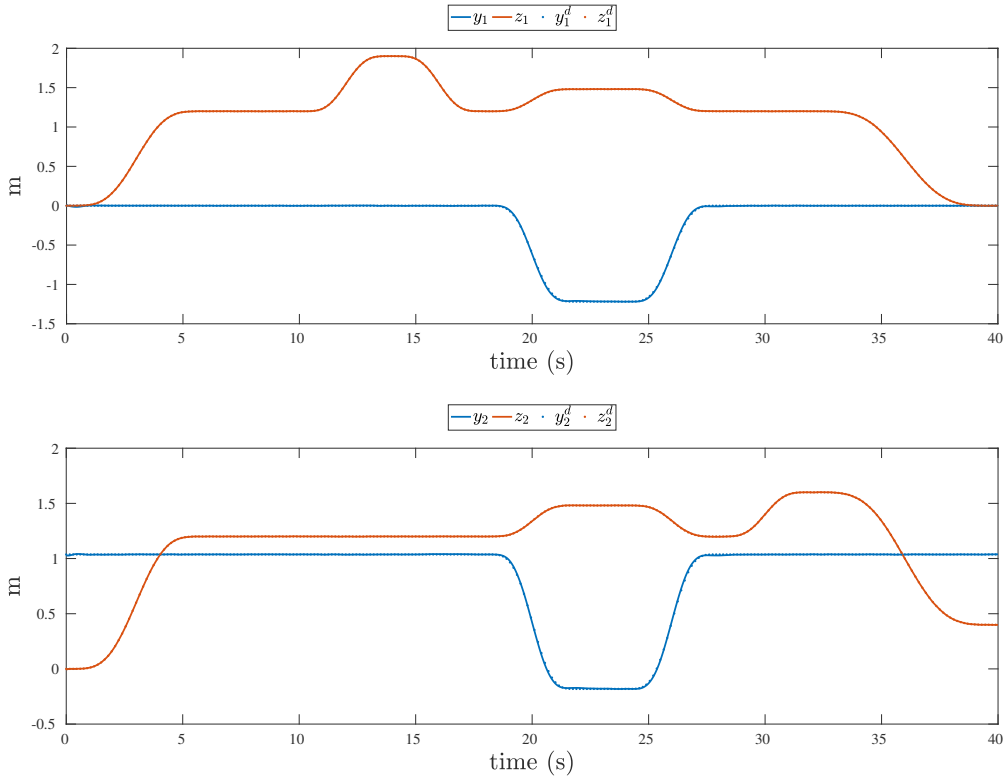


FIGURE 3.21 – Simulation du robot parallèle volant à deux drones : trajectoire désirée et suivi des coordonnées $\mathbf{q}_x$.

Résultats de simulation

Le simulation effectuée demande le suivi d'une trajectoire $\mathbf{q}_x^d$ dans l'espace des drones. La durée de la trajectoire est de soixante secondes. Les drones sont initialisés dans une position horizontale hors équilibre statique du système. La figure 3.21 illustre le suivi de la trajectoire par les drones. La figure 3.22 indique les angles d'orientation calculés par la loi decommande et le suivi de ces angles par la boucle rapide. La convergence en début de trajectoire des erreurs $\mathbf{e}_x$ et $\mathbf{e}_a$ est illustrée Fig. 3.23 et Fig. 3.24. L'erreur sur les variables de maintien dans le plan vertical désiré est donnée Fig. 3.25. Enfin, la figure 3.26 indique les vitesses de rotation des hélices sur chaque drone. Les résultats de suivi des trajectoires et d'erreur sont filtrés pour une meilleure lecture des courbes. La commande arrive à stabiliser le robot et à suivre la trajectoire désirée. Une vidéo de la simulation est disponible à l'adresse https://uncloud.univ-nantes.fr/index.php/s/BkB9n94WaBsjxJn

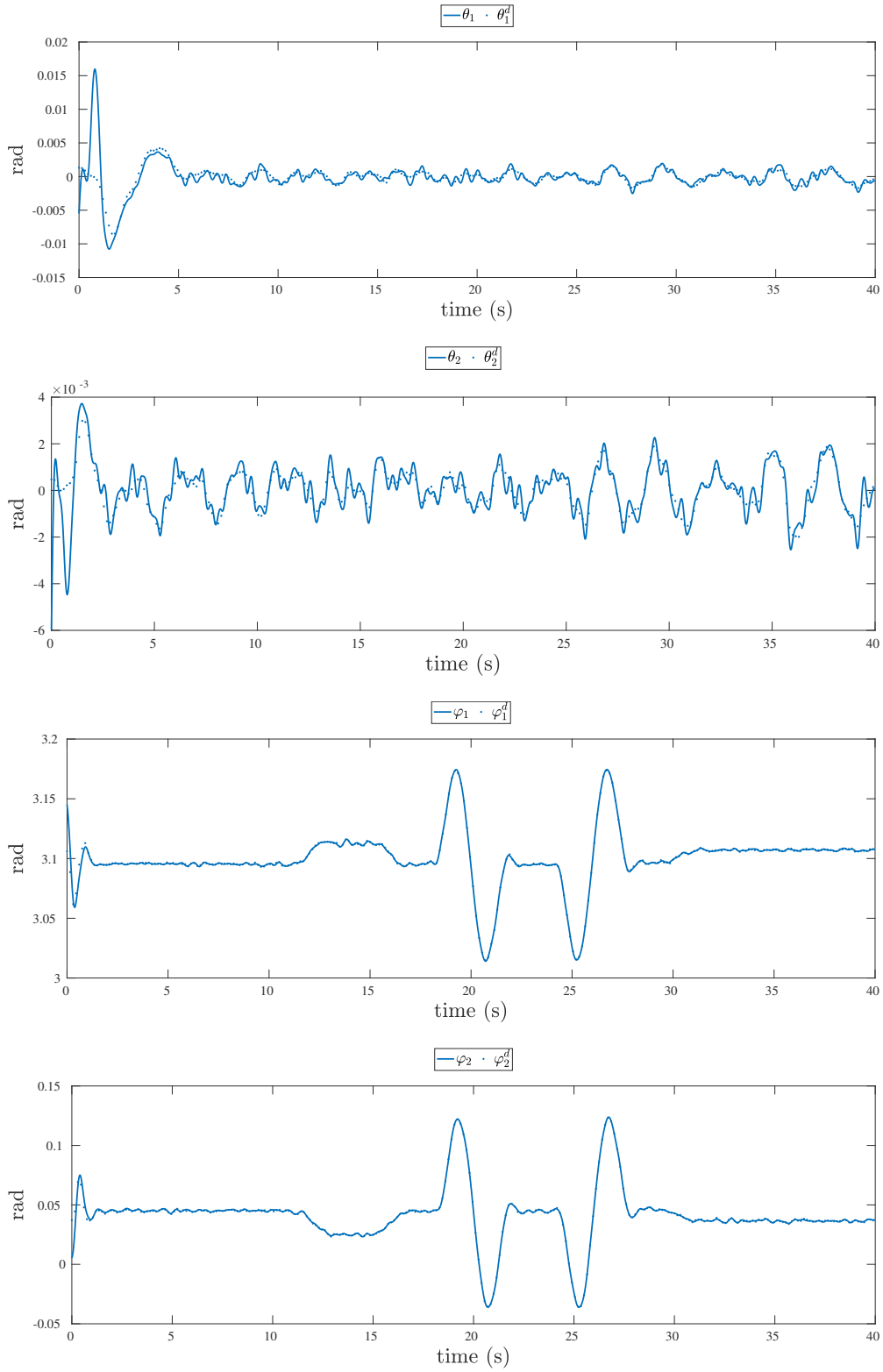


FIGURE 3.22 – Simulation du robot parallèle volant à deux drones : trajectoire désirée et suivi des coordonnées $\mathbf{q}_a$.

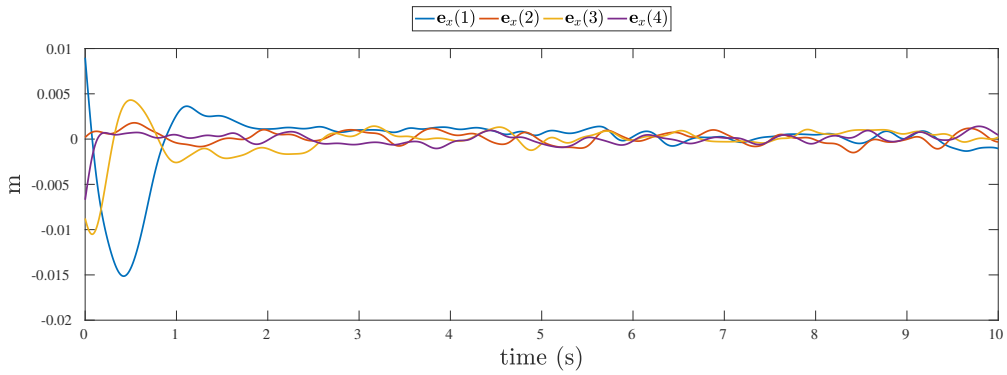


FIGURE 3.23 – Simulation du robot parallèle volant à deux drones : convergence de l'erreur $\mathbf{e}_x$.

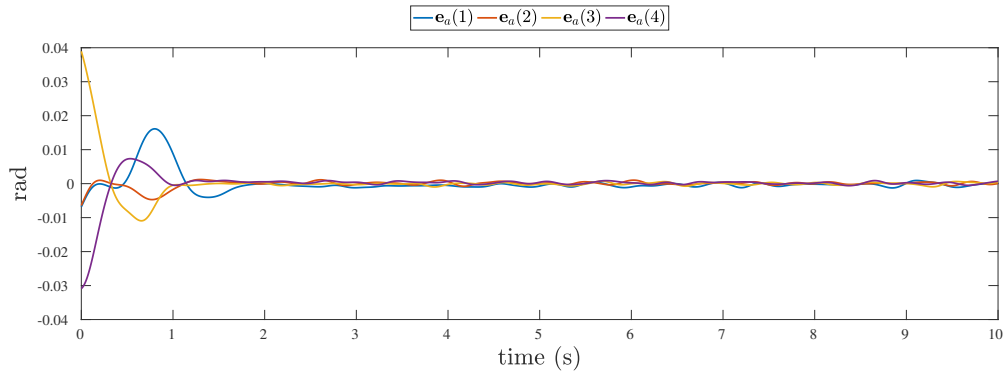


FIGURE 3.24 – Simulation du robot parallèle volant à deux drones : convergence de l'erreur $\mathbf{e}_a$.

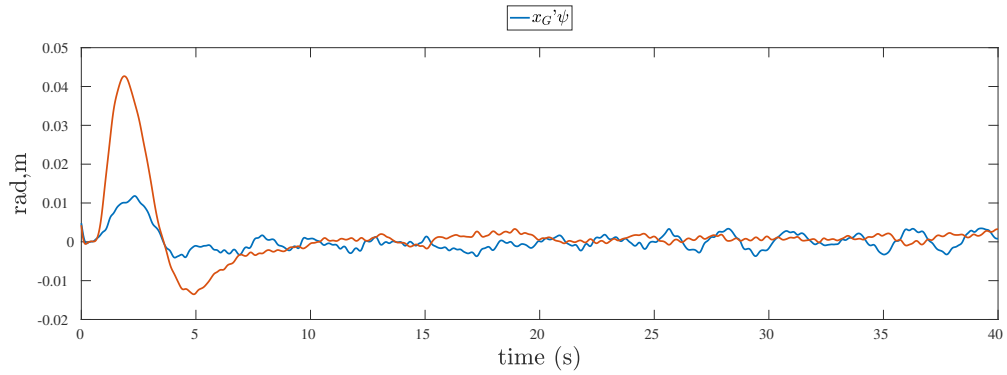


FIGURE 3.25 – Simulation du robot parallèle volant à deux drones : erreur des coordonnées ψ et x_G.

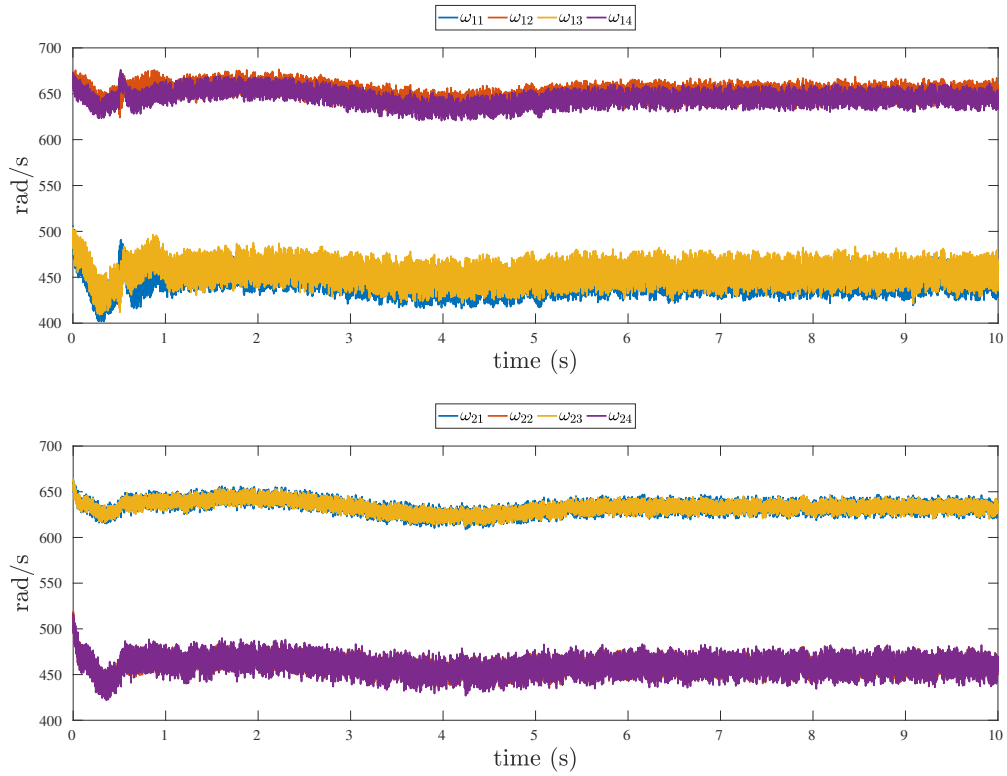


FIGURE 3.26 – Simulation du robot parallèle volant à deux drones : vitesse de rotation des hélices sur les drones.

3.3 Conclusion

Dans ce chapitre, nous avons décliné deux exemples de robots parallèles volants à partir du modèle général développé au chapitre 2. Pour le premier, un robot parallèle volant à trois drones, nous avons pu appliquer directement la loi de commande établie précédemment. De plus, l'étude des singularités de l'architecture passive a été nécessaire pour déterminer un espace de travail pour ce robot. Le robot volant à deux drones, modèle du robot réel utilisé en expérimentation, ne correspond pas parfaitement au cas idéal. Ainsi, il a été nécessaire d'adapter la commande à ce cas particulier, en concevant une commande limitée à un plan vertical dans un premier temps, puis en adoptant une autre approche pour le maintien du robot dans ce plan vertical. Dans les deux cas, des simulations ont été réalisées avec succès. Le chapitre suivant présente le prototype expérimental développé dans le cadre de cette thèse du robot parallèle volant à deux drones.

Chapitre 4

Mise en œuvre expérimentale

Sommaire

4.1 Description du prototype 116
4.1.1 CAO du robot parallèle volant 116
4.1.2 Équipements et communication 117
4.2 Identification des paramètres dynamiques des drones 123
4.2.1 Modèle dynamique d'identification du quadricoptère 124
4.2.2 Modèle dynamique inverse d'identification 126
4.2.3 Procédure d'identification 127
4.2.4 Analyse statistique, paramètres essentiels et moindres carrés totaux pondérés 130
4.2.5 Identification du drone PELICAN 131
4.3 Résultats expérimentaux 140
4.3.1 Paramètres du contrôleur 140
4.3.2 Résultats expérimentaux 141
4.4 Conclusion 149

Ce chapitre est consacré aux résultats expérimentaux obtenus dans le cadre de cette thèse. Dans un premier temps, nous allons décrire le prototype expérimental du robot volant à deux drones réalisé au LS2N. La loi de commande développée pour ce robot étant sensible aux erreurs de modèle dynamique, une seconde partie décrit le protocole expérimental de l'identification des paramètres dynamiques des quadricoptères nécessaires à la commande. Enfin, une troisième partie présente les résultats expérimentaux obtenus avec le prototype.

4.1 Description du prototype

Cette partie a pour vocation de présenter certains aspects techniques de la réalisation du prototype de robot parallèle volant, correspondant au modèle à deux drones présenté dans la partie 3.2. Ce prototype est basé sur l'utilisation de quadricoptères existants, deux PELICANs de la marque ASCTEC. La structure passive reliant les deux robots a été conçue à l'atelier du LS2N. Elle est essentiellement composée d'aluminium.

4.1.1 CAO du robot parallèle volant

La CAO du prototype a été réalisée avec les objectifs suivants :

- la compatibilité avec les drones PELICANs du laboratoire,
- une réduction maximale du poids afin de laisser une marge sur la puissance des rotors pour la commande dynamique du robot,
- une distance suffisante entre les deux robots afin de limiter les perturbations liées au souffle des drones,
- une facilité de maintenance par le démontage aisé de chaque élément.

Le prototype initial possédait deux bras de même longueur et uniquement trois articulations pivots passives conformément au modèle initial présenté dans la partie 3.2. Deux adaptations principales ont été réalisées au cours des expérimentations :

- l'ajout d'une articulation pivot passive supplémentaire (voir la partie 3.2.6) libérant un degré de liberté nécessaire à la commande latérale du robot ;
- suite à l'ajout de cette articulation (Fig. 3.19), il a été constaté que l'inertie le long de l'axe $\mathbf{z}_d$ vue par le drone $\mathcal{D}_1$ était trop importante, menant à une saturation rapide des actionneurs en cas d'écart avec la consigne de la variable θ_1. Ainsi, il a été décidé de réduire la longueur du bras $\mathcal{B}_1$.

La version finale du robot volant est constituée de :

- deux drones PELICANs (Fig. 4.1),
- les deux pièces de liaisons associées respectivement aux drones $\mathcal{D}_1$ et $\mathcal{D}_2$. Ces deux pièces contiennent respectivement les noix des liaisons pivot ① et ③ (voir Fig. 4.4). La pièce de liaison associée au drone $\mathcal{D}_2$ comprend également la liaison passive ajoutée pour la commande latérale (Fig. 4.2).

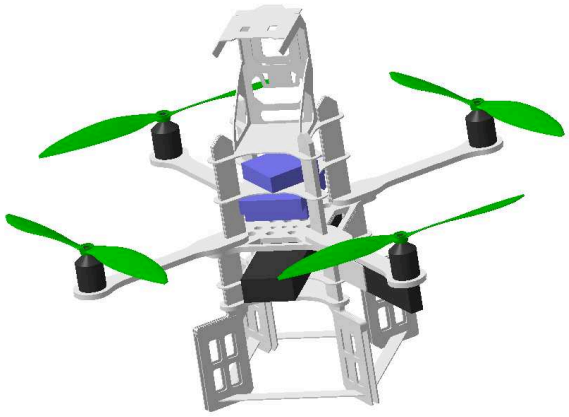


FIGURE 4.1 – CAO du drone PELICAN.

- deux bras correspondants aux corps $\mathcal{B}_1$ et $\mathcal{B}_2$ (voir Fig. 4.4). À chaque extrémité de $\mathcal{B}_1$, se trouve un axe de liaison pivot passive. L'extrémité de $\mathcal{B}_2$ en liaison avec $\mathcal{B}_1$ comprend la noix de la liaison pivot centrale ②. A l'autre extrémité, on trouve l'axe de la liaison ③ (Fig. 4.3).

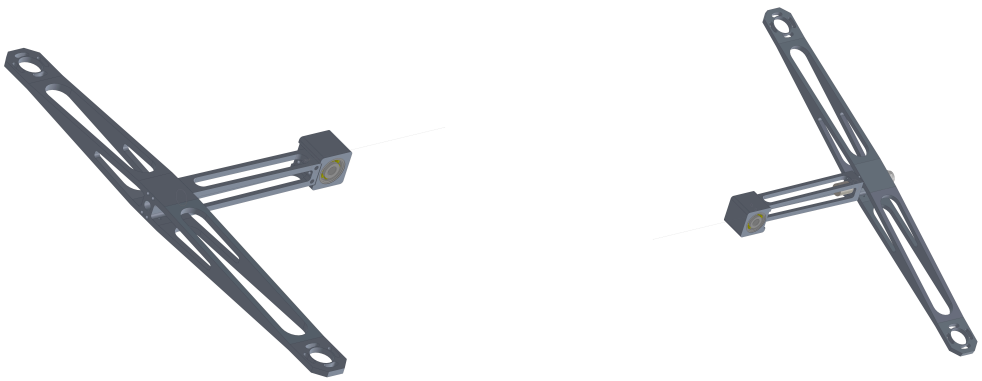
La figure 4.4 présente l'assemblage CAO de ce robot et la figure 4.5 une photographie du prototype réel.

4.1.2 Équipements et communication

Système embarqué du PELICAN

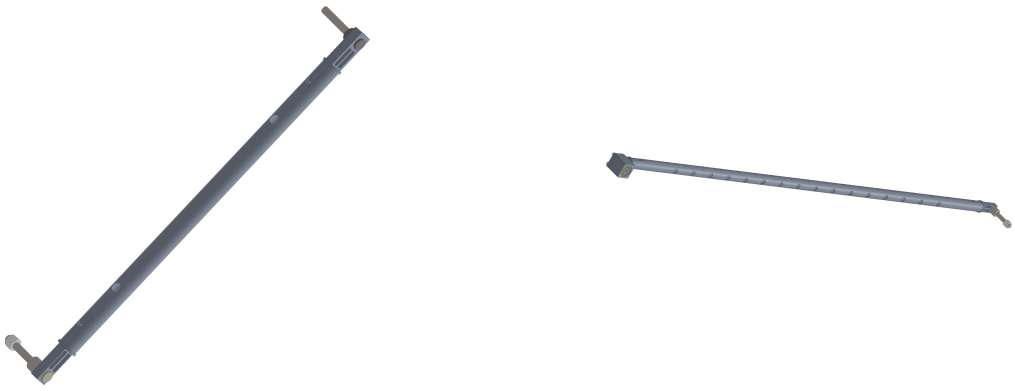
Le drone PELICAN dispose d'un système embarqué composé de deux microcontrôleurs appelés LLP (Low Level Processor) et HLP (High Level Processor).

Le LLP ne peut pas être modifié par l'utilisateur. Il contient la loi de commande développée par la société ASCTEC et présente plusieurs modes de contrôle du drône (manuel, altitude, GPS). Le LPP collecte les informations des capteurs GPS, gyroscope et accéléromètre. Il effectue également la commande des moteurs via un bus SPI (Serial Peripheral Interface). La fréquence maximale de collecte des données et d'envoi des commandes moteurs est 1kHz.



(a) Raccordement au drone 1 (b) Raccordement au drone 2

FIGURE 4.2 – CAO des pièces de raccordement aux drones du prototype.



(a) Bras $\mathcal{B}_1$ (b) Bras $\mathcal{B}_2$

FIGURE 4.3 – CAO des bras de la structure passive du prototype.

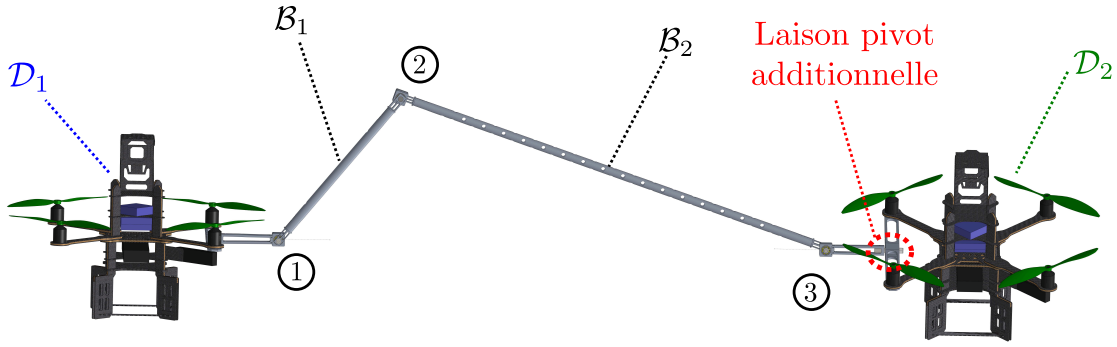


FIGURE 4.4 – CAO du prototype assemblé.

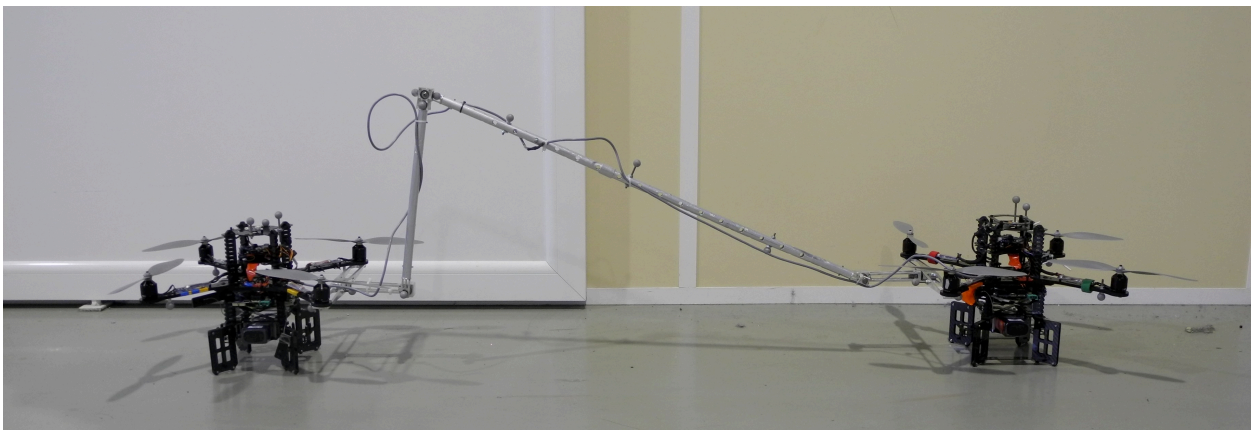


FIGURE 4.5 – Photographie du prototype assemblé.

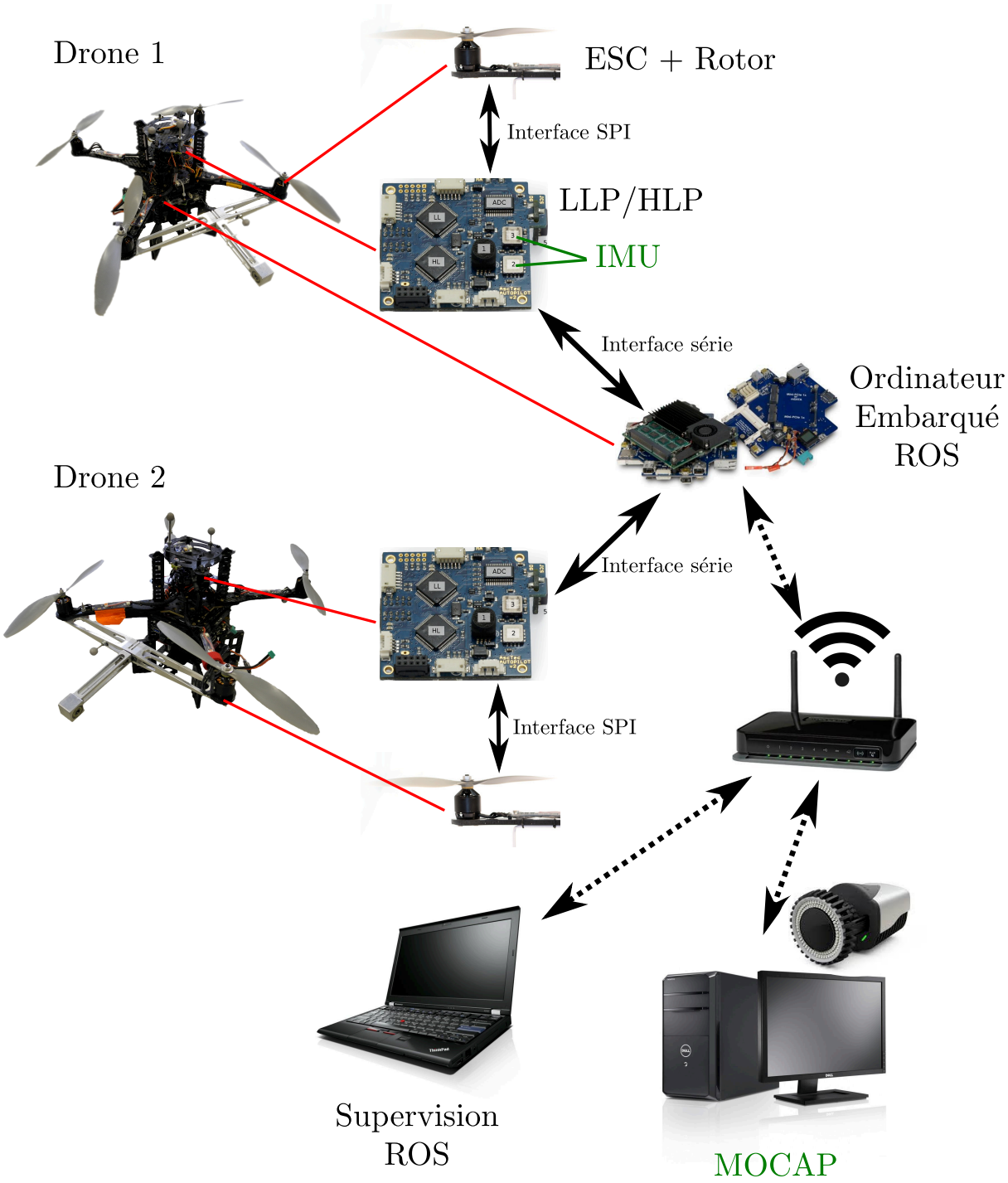


FIGURE 4.6 – Schéma des équipements et communications du prototype.

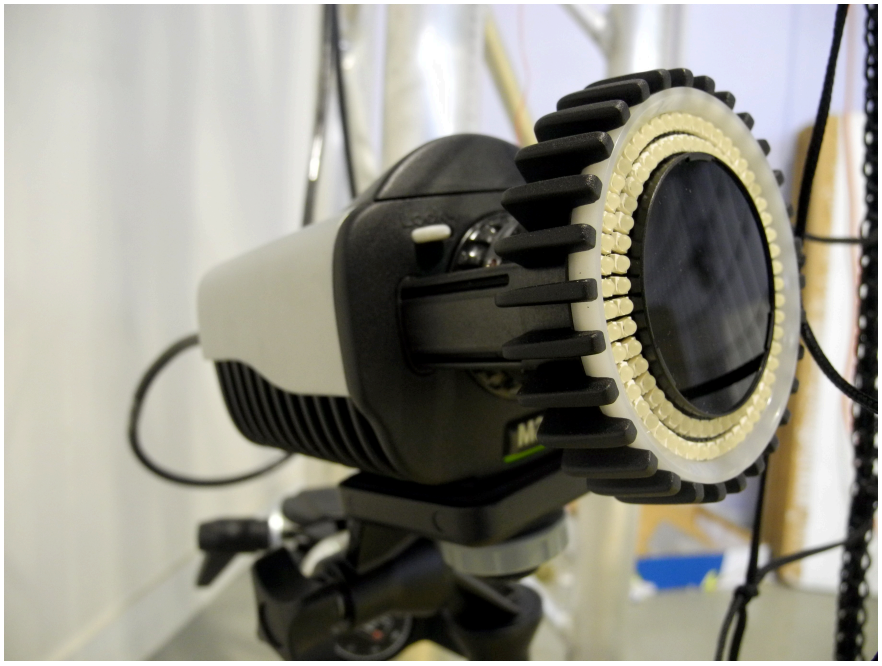


FIGURE 4.7 – Photographie d'une caméra infrarouge.

Le HLP peut être modifié par l'utilisateur et permet de développer une loi de commande pour les applications spécifiques. Une communication série est établie entre le LLP et le HLP afin de récupérer les informations capteur et de renvoyer la commande vers les moteurs. Plusieurs modes de contrôle sont disponibles :

- commande directe des vitesses moteurs,
- commande en couple/force,
- commande en altitude/orientation,
- commande par points de passage GPS.

Pour notre application, la commande en vitesses moteurs est sélectionnée. Elle permet de s'affranchir au maximum des modèles du constructeur intégrés dans le LLP.

Le PELICAN dispose également d'un ordinateur de bord embarqué. Il est muni d'une distribution GNU/LINUX et du middleware ROS (Robor Operating System). Dans le cadre de notre expérimentation, le calcul des commandes est entièrement effectué sur l'ordinateur de bord du drone 1 sous la forme d'un nœud ROS (Fig. 4.6). Cet ordinateur est relié au HLP de chacun des drones qui ne servent que de relais vers leurs LLPs respectifs. La fréquence de communication de collecte des données et de renvoi des commandes moteurs vers les LLP de chacun des drones est fixée à 1 kHz. L'utilisation du nœud ROS facilite la collecte des données et l'interfaçage avec le système de capture de mouvement.

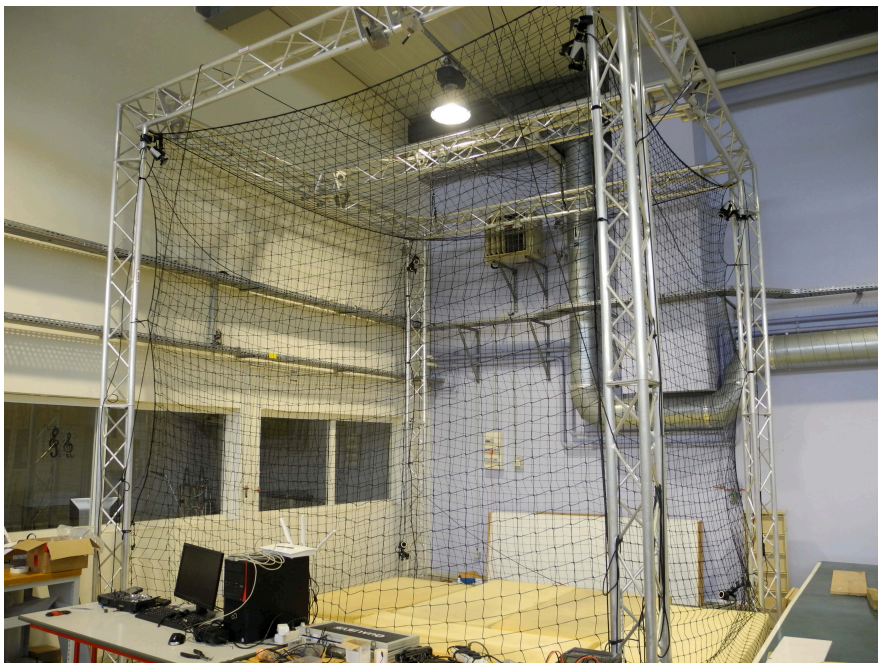


FIGURE 4.8 – Photographie de la zone expérimentale.

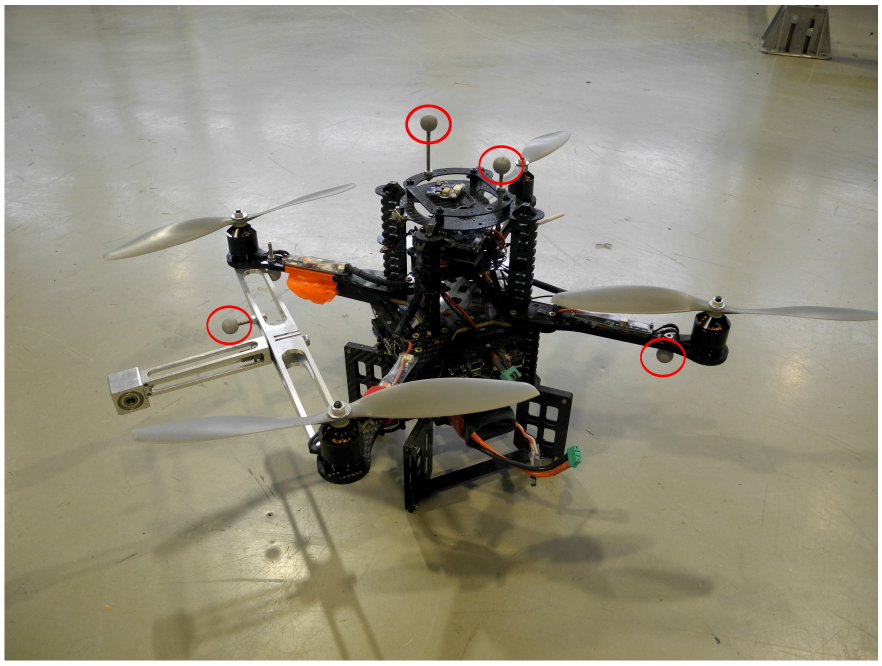


FIGURE 4.9 – Photographie d'un drone muni des marqueurs infrarouges (entourés en rouge).

Capture de mouvement

Les capteurs embarqués à bord de chaque drone ne mesurent que l'accélération et la vitesse angulaire de ceux-ci. A long terme, les erreurs d'intégration rendent impossible une mesure précise de la position des drones. Ainsi, il est nécessaire de disposer d'une mesure extéroceptive de la position des drones. Cette mesure s'effectue à l'aide d'un système de capture de mouvements (MOCAP). Le système utilisé dans le cadre de cette thèse est fourni par la société QUALISYS. Il comprend 8 caméras infrarouges (Fig. 4.7) disposées aux 8 coins de la zone expérimentale (Fig. 4.8). Ces caméras détectent les marqueurs infrarouges disposés sur les deux drones (Fig. 4.9) afin d'effectuer une estimation de la position et de l'orientation de ceux-ci. La mesure est effectuée à une fréquence de 250 Hz. L'utilisation de cette mesure, en association avec les capteurs de la centrale inertielle de chaque drone, s'effectue par une fusion de données décrite en annexe B.2 et calculée par l'ordinateur de bord du drone 1.

Avant de tester le contrôleur en conditions réelles, il est nécessaire de connaître au maximum les paramètres dynamiques du robot. La partie suivante décrit la méthodologie utilisée pour l'estimation des paramètres dynamiques des drones associés aux pièces de raccordement de la structure passive.

4.2 Identification des paramètres dynamiques des drones

La commande développée pour le contrôle du robot parallèle volant (section 2.4.4) comprend l'inversion du modèle dynamique de l'architecture passive et des drones actionneurs. Les paramètres dynamiques de l'architecture passive ont été estimés à partir de la CAO. Cependant, ceux des drones, notamment les coefficients de poussée et de traînée, ne sont pas connus. Les caractéristiques du fournisseur étant incertaines, il a été décidé d'identifier les paramètres dynamiques des quadricoptères utilisés dans le cadre des expérimentations.

A notre connaissance, l'estimation expérimentale des paramètres dynamiques d'un quadricoptère est effectuée à partir de bancs de tests spécifiques [Derafa et al., 2006; Elsamanty et al., 2013] ou d'une estimation issue de l'analyse des données de vol mais avec une connaissance supposée des paramètres dynamiques de l'actionnement des drones [Abas et al., 2011; Rashid et Akhtar, 2012].

Une méthode d'identification des paramètres dynamiques connue pour les robots manipulateurs classiques, nommée IDIM-LS, utilise un modèle dynamique inverse et une régression par moindres carrés [Khosla et Kanade, 1985; Lu et al., 1993; Kozlowski, 1996; Antonelli et al., 1999; Gautier et Poignet, 2001; Khalil et Dombre, 2002]. Une extension de cette méthode utilisant les moindres carrés totaux (IDIM-TLS) permet l'identification des paramètres dynamiques des corps et des gains d'actionnement d'un robot en une seule procédure [Briot et Gautier, 2015]. Comme nous recherchons à la fois des paramètres dynamiques et des gains d'actionnement pour le drone, l'adaptation de cette méthode semble appropriée.

L'identification sera effectuée à partir des données mesurées sur une trajectoire. Deux

systèmes de mesures sont disponibles pour l'expérimentation (voir annexe B.2). Le système de capture de mouvements (MOCAP) fournit les coordonnées du quadricoptère dans le repère de référence tandis que la centrale inertielle (IMU) nous donne les accélérations et vitesses angulaires dans le repère local du drone. Nous montrerons qu'il est possible d'effectuer la procédure d'identification à partir de chacun de ces deux systèmes de mesure. Les paramètres identifiés seront ensuite validés à l'aide d'une seconde trajectoire.

4.2.1 Modèle dynamique d'identification du quadricoptère

Considérons un quadricoptère $\mathcal{D}$ et définissons un repère local $\mathcal{F}_d$ associé à ce drone, de façon similaire à la partie 1.1.1. Les coordonnées du drone sont alors :

- ${}^d\boldsymbol{\xi}$ la position du point O_d dans le repère $\mathcal{F}_d$,
- $\boldsymbol{\eta} = [\psi, \theta, \varphi]^T$ les coordonnées d'orientation du repère $\mathcal{F}_d$ par rapport au repère $\mathcal{F}_o$. La matrice de rotation $\mathbf{R}_d$ telle que ${}^o\mathbf{v} = \mathbf{R}_d{}^d\mathbf{v}$ est donnée par :

$$\mathbf{R}_d = \mathbf{R}_\mathbf{z}(\psi)\mathbf{R}_\mathbf{y}(\theta)\mathbf{R}_\mathbf{x}(\varphi)$$

Le quadricoptère est un simple corps rigide sur lequel sont appliqués les forces et moments générés par les hélices. Ainsi, ses paramètres dynamiques sont ceux d'un corps rigide associés aux gains de poussée et de traînée des hélices. Les paramètres dynamiques du corps rigide sont :

- m_d sa masse,
- MX_d, MY_d et MZ_d les trois composantes du premier moment d'inertie ${}^d\mathbf{ms}$ en O_d, exprimé dans le repère $\mathcal{F}_d$,
- les six coefficients de la matrice d'inertie exprimée en O_d dans le repère $\mathcal{F}_d$:

$$\mathbf{I}_d = \begin{bmatrix} XX_d & XY_d & XZ_d \\ XY_d & YY_d & YZ_d \\ XZ_d & YZ_d & ZZ_d \end{bmatrix}$$

La vitesse angulaire du drone au point O_d est notée $\boldsymbol{\omega}$. Pour rappel, $\boldsymbol{\omega}$ est reliée à la dérivée des coordonnées d'orientation $\dot{\boldsymbol{\eta}}$ par :

$${}^d\boldsymbol{\omega} = \mathbf{T}_d\dot{\boldsymbol{\eta}} \tag{E4.1}$$

avec $\mathbf{T}_d$ une matrice (3×3) définie par l'équation (E2.19).

Dynamique du corps rigide

En partie 1.1.2, nous avons développé le modèle dynamique classique du quadricoptère. Pour ce modèle, la force de poussée $\mathbf{f}_d$ et le moment $\mathbf{m}_d$ sont appliqués au centre de gravité

du quadricoptère. Cette hypothèse, cependant, peut ne pas être vérifiée. Afin de s'assurer d'inclure tous les paramètres dynamiques dans la procédure d'identification, nous allons partir des équations de Newton-Euler d'un corps rigide soumis à une force $\mathbf{f}_d$, un moment $\mathbf{m}_d$ et à la force de gravité dans le cas général [Khalil et Dombre, 2002]. Elles s'écrivent :

$$m_d {}^d\ddot{\boldsymbol{\xi}} + {}^d\dot{\boldsymbol{\omega}} \times {}^d\mathbf{ms} + {}^d\boldsymbol{\omega} \times ({}^d\boldsymbol{\omega} \times {}^d\mathbf{ms}) = m_d \mathbf{R}_d^{-1o}\mathbf{g} + {}^d\mathbf{f}_d \tag{E4.2}$$

$$\mathbf{I}_d {}^d\dot{\boldsymbol{\omega}} + {}^d\boldsymbol{\omega} \times \mathbf{I}_d {}^d\boldsymbol{\omega} + {}^d\mathbf{ms} \times {}^d\boldsymbol{\xi}_d = {}^d\mathbf{ms} \times (\mathbf{R}_d^{-1o}\mathbf{g}) + {}^d\mathbf{m}_d \tag{E4.3}$$

Les données mesurées nécessaires au calcul des équations (E4.2) et (E4.3) sont

- l'accélération du drone dans son repère local ${}^d\ddot{\boldsymbol{\xi}}$,
- la matrice de rotation $\mathbf{R}_d$,
- la vitesse angulaire du drone dans son repère local ${}^d\boldsymbol{\omega}$,
- la dérivée de cette vitesse angulaire ${}^d\dot{\boldsymbol{\omega}}$.

L'IMU du drone mesure l'accélération spécifique ${}^d\ddot{\boldsymbol{\xi}} + \mathbf{R}_d^{-1o}\mathbf{g}$ et la vitesse de rotation ${}^d\boldsymbol{\omega}$ grâce aux instruments embarqués. Une estimation des orientations $\boldsymbol{\eta}$, et donc de la matrice $\mathbf{R}_d$, est également calculée (cf annexe B.1). ${}^d\dot{\boldsymbol{\omega}}$ est obtenu par une dérivée filtrée de ${}^d\boldsymbol{\omega}$ [Hollerbach et al., 2008].

Dans cette partie, nous souhaitons également établir un modèle dynamique en fonction des mesures obtenues par le MOCAP. Ces mesures sont :

- ${}^o\boldsymbol{\xi}$ la position du point O_d exprimée dans le repère $\mathcal{F}_o$,
- $\boldsymbol{\eta}$ les coordonnées d'orientation du drone dans le repère $\mathcal{F}_o$,

et leurs dérivées respectives. Les accélérations ${}^o\ddot{\boldsymbol{\xi}}$ et ${}^d\ddot{\boldsymbol{\xi}}$ sont reliées par la matrice de rotation $\mathbf{R}_d$ par :

$${}^d\ddot{\boldsymbol{\xi}} = \mathbf{R}_d^{-1o}\ddot{\boldsymbol{\xi}} \tag{E4.4}$$

La vitesse de rotation ${}^d\boldsymbol{\omega}$ et reliée à la dérivée des coordonnées $\boldsymbol{\eta}$ par (E4.1). La dérivée de cette équation nous donne :

$${}^d\dot{\boldsymbol{\omega}} = \dot{\mathbf{T}}_d\dot{\boldsymbol{\eta}} + \mathbf{T}_d\ddot{\boldsymbol{\eta}} \tag{E4.5}$$

En introduisant (E4.4), (E4.1) et (E4.5) dans les équations de la dynamique du drone (E4.2) et (E4.3), nous obtenons :

$$m_d \mathbf{R}_d^{-1o}\ddot{\boldsymbol{\xi}} + (\dot{\mathbf{T}}_d\dot{\boldsymbol{\eta}} + \mathbf{T}_d\ddot{\boldsymbol{\eta}}) \times \mathbf{ms} + \mathbf{T}_d\dot{\boldsymbol{\eta}} \times (\mathbf{T}_d\dot{\boldsymbol{\eta}} \times \mathbf{ms}) = m_d \mathbf{R}_d^{-1o}\mathbf{g} + {}^d\mathbf{f}_d \tag{E4.6}$$

$$\mathbf{I}_d(\dot{\mathbf{T}}_d\dot{\boldsymbol{\eta}} + \mathbf{T}_d\ddot{\boldsymbol{\eta}}) + \mathbf{T}_d\dot{\boldsymbol{\eta}} \times \mathbf{I}_d\mathbf{T}_d\dot{\boldsymbol{\eta}} + \mathbf{ms} \times \mathbf{R}_d^{-1o}\ddot{\boldsymbol{\xi}} = \mathbf{ms} \times (\mathbf{R}_d^{-1o}\mathbf{g}) + {}^d\mathbf{m}_d \tag{E4.7}$$

Paramètres dynamiques d'actionnement

Le modèle d'actionnement du quadricoptère est inchangé par rapport à celui donné par l'équation (E1.4) :

$${}^d\mathbf{w}_d = \begin{bmatrix} {}^d\mathbf{f}_d \\ {}^d\mathbf{m}_d \end{bmatrix} = \mathbf{Q}\boldsymbol{\Omega} \tag{E4.8}$$

Ainsi, le drone possède deux paramètres dynamiques liés à l'actionnement : son coefficient de poussée k_t et son coefficient de traînée k_d inclus dans la matrice $\mathbf{Q}$. Le vecteur des coefficients d'actionnement est définit par : $\mathbf{k} = [k_t \; k_d]^T$.

4.2.2 Modèle dynamique inverse d'identification

La masse d'un drone est le seul paramètre dynamique aisément mesurable à partir d'une balance. Cette masse servira donc de référence à l'estimation des autres paramètres dynamiques. L'ensemble des paramètres dynamiques du corps rigide inconnus sont concaténés dans un vecteur $\boldsymbol{\chi}$:

$$\boldsymbol{\chi}^T = [MX_d \; MY_d \; MZ_d \; XX_d \; YY_d \; ZZ_d \; XY_d \; XZ_d \; YZ_d]$$

La procédure d'identification IDIM-TLS [Briot et Gautier, 2015] nécessite d'exprimer le modèle dynamique inverse sous une forme linéaire en fonction des paramètres dynamiques. Cette forme est connue dans le cas des robots classiques et peut être également obtenue pour le modèle dynamique d'un quadricoptère.

Dans cette étude, nous regrouperons les termes des équations de la dynamique du quadricoptère ((E4.2), (E4.3) et (E4.8)) sous la forme suivante :

$$m_d \mathbf{d}_m + \mathbf{D}\boldsymbol{\chi} = \mathbf{D}_k \mathbf{k} \tag{E4.9}$$

le vecteur $\mathbf{d}_m$ et les matrices $\mathbf{D}$ et $\mathbf{D}_k$ contiennent respectivement les termes linéaires en m, $\boldsymbol{\chi}$ et $\mathbf{k}$ du modèle dynamique du quadricoptère. Leurs expressions développées sont données par :

$$\mathbf{d}_m = \begin{bmatrix} a_x - gS\theta & a_y + gS\varphi C\theta & a_z + gC\varphi C\theta & 0 & 0 & 0 \end{bmatrix}^T$$

$$\mathbf{D} = \begin{bmatrix} -\omega_y^2 - \omega_z^2 & \omega_x\omega_y - \dot{\omega}_z & \omega_x\omega_z + \dot{\omega}_y & 0 \\ \omega_x\omega_y + \dot{\omega}_z & -\omega_x^2 - \omega_z^2 & -\dot{\omega}_x + \omega_y\omega_z & 0 \\ \omega_x\omega_z - \dot{\omega}_y & \dot{\omega}_x + \omega_y\omega_z & -\omega_x^2 - \omega_y^2 & 0 \\ 0 & a_z + gC\varphi C\theta & -a_y - gS\varphi C\theta & \dot{\omega}_x \\ -a_z - gC\varphi C\theta & 0 & a_x - gS\theta & \omega_x\omega_z \\ a_y + gS\varphi c\theta & -a_x + gS\theta & 0 & -\omega_x\omega_y \end{bmatrix}$$

$$\begin{bmatrix} 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ -\omega_y\omega_z & \omega_y\omega_z & -\omega_x\omega_z + \dot{\omega}_y & \omega_x\omega_y + \dot{\omega}_z & \omega_y^2 - \omega_z^2 \\ \dot{\omega}_y & -\omega_x\omega_z & \dot{\omega}_x + \omega_y\omega_z & -\omega_x^2 + \omega_z^2 & -\omega_x\omega_y + \dot{\omega}_z \\ \omega_x\omega_y & \dot{\omega}_z & \omega_x^2 - \omega_y^2 & \dot{\omega}_x - \omega_y\omega_z & \omega_x\omega_z + \dot{\omega}_y \end{bmatrix}$$

$$\mathbf{D}_k = \begin{bmatrix} 0 & 0 \\ 0 & 0 \\ \sum_i \omega_i^2 & 0 \\ d(-\omega_1^2 - \omega_3^2 + \omega_2^2 + \omega_4^2) & 0 \\ d(\omega_1^2 + \omega_4^2 - \omega_2^2 - \omega_3^2) & 0 \\ 0 & \sum_i -\text{sign}(\omega_i)\omega_i^2 \end{bmatrix}$$

avec :

- a_x, a_y, a_z tels que ${}^d\ddot{\boldsymbol{\xi}} = [a_x\ a_y\ a_z]^T$
- ω_x, ω_y, ω_z tels que ${}^d\boldsymbol{\omega} = [\omega_x\ \omega_y\ \omega_z]^T$
- g la norme de la gravité ${}^o\mathbf{g} = [0\ 0\ -g]^T$,
- ω_i $(i = 1..4)$ la vitesse de rotation de chaque hélice $\Omega = [\omega_1\ \omega_2\ \omega_3\ \omega_4]^T$.

Les matrices $\mathbf{D}$ et $\mathbf{d}_m$ peuvent être obtenues à partir des données collectées lors d'une trajectoire effectuée par le drone :

1. *À partir des données de la centrale inertielle (IMU).* La vitesse angulaire ${}^d\boldsymbol{\omega}$ et l'accélération ressentie dans le repère local ${}^d\ddot{\boldsymbol{\xi}} - \mathbf{R}^{-1}{}^o\mathbf{g}$ sont mesurées directement par le gyroscope et l'accéléromètre. L'estimation des angles ψ, θ et φ est effectuée selon la procédure décrite dans l'annexe B.1. L'estimation de l'accélération angulaire ${}^o\hat{\dot{\boldsymbol{\omega}}}$ est obtenue par l'application d'un filtre Butterworth passe-bas à phase nulle et d'une dérivée numérique à différence centrée sur les données mesurées pour ${}^o\boldsymbol{\omega}$ [Hollerbach et al., 2008].
2. *À partir des données de la capture de mouvement (MOCAP).* La position ${}^o\boldsymbol{\xi}$ du point O_d dans le repère $\mathcal{F}_o$ et les coordonnées d'orientation du drone $\boldsymbol{\eta}$ sont mesurées par le système de capture de mouvement. L'estimation de leurs dérivées respectives ${}^o\hat{\dot{\boldsymbol{\xi}}}$ et $\hat{\dot{\boldsymbol{\eta}}}$ sont obtenues par l'application d'un filtre Butterworth passe-bas à phase nulle et d'une dérivée numérique à différence centrée sur les données mesurées. L'estimation des dérivées secondes $\hat{\ddot{\boldsymbol{\eta}}}$ et ${}^o\hat{\ddot{\boldsymbol{\xi}}}$ est obtenue en effectuant à nouveau ces opérations sur les dérivées premières ${}^o\hat{\dot{\boldsymbol{\xi}}}$ et $\hat{\dot{\boldsymbol{\eta}}}$. Ensuite, les vecteurs ${}^d\ddot{\boldsymbol{\xi}}$, $\boldsymbol{\omega}$ et $\dot{\boldsymbol{\omega}}$ sont calculés à grâce aux équations (E4.4), (E4.1) et (E4.5).

La matrice $\mathbf{D}_k$ est obtenue à partir des données de commande des moteurs. Un filtre Butterworth passe-bas à phase nulle est également appliqué sur ces mesures.

4.2.3 Procédure d'identification

Moindres carrés classiques

Nous cherchons à obtenir l'identification des paramètres dynamiques du quadricoptère $\boldsymbol{\chi}$ grâce au modèle dynamique inverse d'identification (E4.9). La méthode d'identification

par moindres carrés (IDIM-LS), telle qu'appliquée sur un manipulateur classique, requiert la connaissance des gains d'actionnement. Pour une application directe de cette méthode sur le quadricoptère, il serait nécessaire de disposer d'une estimation précise des paramètres de gains d'actionnement k_t et k_d. Pour cela, il est possible d'utiliser des bancs d'essais avec le risque d'une estimation biaisée par les conditions expérimentales de ces bancs.

Dans cette étude, nous effectuons l'identification en nous basant sur la connaissance d'un paramètre dynamique aisément mesurable, la masse du quadricoptère. Écrivons alors le modèle dynamique inverse (E4.9) sous la forme :

$$\begin{bmatrix}-\mathbf{D}_k & \mathbf{D}\end{bmatrix} \boldsymbol{\chi}_k = -m_d \mathbf{d}_m \tag{E4.10}$$

La partie gauche de (E4.10) regroupe les paramètres dynamiques inconnus, à savoir les gains d'actionnement et le vecteur des paramètres dynamiques $\boldsymbol{\xi}$ ($\boldsymbol{\chi}_k^T = [\mathbf{k}^T \quad \boldsymbol{\chi}^T]$), tandis que la partie droite contient les termes associés à la masse connue m_d. A partir de l'équation (E4.10), l'utilisation des moindres carrés permet d'estimer les paramètres dynamiques inconnus. La méthode consiste à trouver les paramètres dynamiques $\boldsymbol{\chi}_k^T$ minimisant l'erreur de modèle sur l'équation (E4.10) à partir d'un grand nombre de données de vol. Cette méthode, similaire à la méthode IDIM-LS, comprend les étapes suivantes [Hollerbach et al., 2008] :

1. Un vol est effectué par le quadricoptère. Lors de ce vol, les échantillons de données (vitesse des moteurs et mesures) sont collectées à la fréquence f_m.
2. Le calcul des matrices $\mathbf{D}_k$, $\mathbf{D}$ et du vecteur $\mathbf{d}_m$ (E4.9) est effectué pour chaque échantillon de données collecté. Pour chaque ligne i du système d'équation (E4.9), les données calculées pour chaque échantillon sont regroupées dans une matrice $\mathbf{W}_i^{f_m}$ et un vecteur $\mathbf{w}_{m,i}^{f_m}$. Ainsi, nous obtenons les systèmes surdéterminés pour chaque ligne i :

$$\mathbf{W}_i^{f_m} \boldsymbol{\chi}_k = m_d \mathbf{w}_{m,i}^{f_m} + \boldsymbol{\rho}_i^{f_m} \tag{E4.11}$$

   avec $\boldsymbol{\rho}_i^{f_m}$ le vecteur d'erreur contenant les bruits de mesure et les erreurs de modèle.
3. Afin de limiter les perturbations à hautes fréquences, notamment la dynamique du moteur non modélisée, un sous-échantillonage parallèle est effectué sur chaque colonne de $\mathbf{W}_i^{f_m}$ et sur $\mathbf{w}_{m,i}^{fm}$. Les matrices et vecteurs sous-échantillonnés de $\mathbf{W}_i^{f_m}$ et $\mathbf{w}_{m,i}^{fm}$ sont notés respectivement $\mathbf{W}_i$ et $\mathbf{w}_{m,i}$
4. Les matrices $\mathbf{W}_i$ et les vecteurs $\mathbf{w}_{m,i}$ sont concaténés respectivement en la matrice $\mathbf{W}$ et le vecteur $\mathbf{w}_m$, donnant alors le système surdéterminé à r équations, r étant le nombre d'échantillons collectés :

$$\mathbf{W} \boldsymbol{\chi}_k = m_d \mathbf{w}_m + \boldsymbol{\rho} \tag{E4.12}$$

L'identification des paramètres dynamiques pourrait alors être obtenue en appliquant la méthode des moindres carrés au système d'équations (E4.12). Cette solution est donnée

par :

$$\boldsymbol{\chi}_k = m_d \mathbf{W}^+ \mathbf{w}_m \tag{E4.13}$$

avec $\mathbf{W}^+ = (\mathbf{W}^T\mathbf{W})^{-1}\mathbf{W}^T$ la pseudo-inverse de Moore-Penrose de la matrice $\mathbf{W}$. La solution obtenue est celle minimisant la norme 2 du vecteur $\boldsymbol{\rho}$. Cependant, l'application de la méthode des moindres carrés fournit la "meilleure" estimation des paramètres dynamiques dans le cas où les erreurs de mesure ne sont contenues que dans le terme $\mathbf{w}_m$. Or, les données utilisées pour le calcul de la matrice $\mathbf{W}$ sont issues également de mesures bruitées. La méthode des moindres carrés totaux présentée ci-après est alors préférable pour l'estimation des paramètres dynamiques.

Moindres carrés totaux

La procédure d'identification par moindre carrés totaux est présentée dans [Van Huffel et Vandewalle, 1991] [Rao et Toutenburg, 1999]. Nous détaillerons ici son application à l'identification des paramètres dynamiques du quadricoptère. L'équation (E4.12) est écrite sous la forme :

$$\mathbf{W}_t \boldsymbol{\chi}_t = \boldsymbol{\rho}, \qquad \boldsymbol{\chi}_t^T = \begin{bmatrix} \boldsymbol{\chi}_k^T & 1 \end{bmatrix} \tag{E4.14}$$

avec $\mathbf{W}_t = \begin{bmatrix} \mathbf{W} & -m_d \mathbf{w}_m \end{bmatrix}$ une matrice de dimension $(r \times n_t)$ et $\boldsymbol{\chi}_t^T$ un vecteur de dimension (n_t). n_t est le nombre total de paramètres dynamiques du système que l'on cherche à identifier. Dans notre cas $n_t = 12$. En l'absence de perturbation, $\boldsymbol{\rho} = \mathbf{0}$. $\mathbf{W}_t$ devrait alors être de rang dégénéré pour admettre une solution non nulle. Dans ce cas, il existe une infinité de solutions dans le noyau de la matrice $\mathbf{W}_t$. Ces solutions sont données par :

$$\hat{\boldsymbol{\chi}}_t = \lambda \hat{\boldsymbol{\chi}}_t^* \tag{E4.15}$$

avec $\hat{\boldsymbol{\chi}}_t^*$ le vecteur unitaire tel que $\mathbf{W}_t \hat{\boldsymbol{\chi}}_t^* = \mathbf{0}$ et λ un nombre réel.

Cependant, du fait des pertubations (bruits de mesure, incertitudes du modèle), $\mathbf{W}_t$ est en pratique de rang plein. Ainsi, le système d'équations (E4.14) est modifié en un système compatible le plus proche au sens de la norme de Frobenius. Le nouveau système d'équations est obtenu à partir de la décomposition en valeurs propres de la matrice $\mathbf{W}_t$. Cette décomposition s'écrit :

$$\mathbf{W}_t = \mathbf{U}\mathbf{S}\mathbf{V}^T \tag{E4.16}$$

avec

- $\mathbf{U}$ et $\mathbf{V}$ des matrices orthonormales de tailles respectives $(r \times n_t)$ et $(n_t \times n_t)$
- $\mathbf{S}$ une matrice diagonale comportant les valeurs singulières s_j de $\mathbf{W}_t$ classées en ordre décroissant.

La matrice de rang dégénéré $\overline{\mathbf{W}}_t$ la plus proche au sens de la norme de Frobenius est donnée par l'expression suivante :

$$\overline{\mathbf{W}}_t = \mathbf{W}_t - s_{n_t} \mathbf{U}_{n_t} \mathbf{V}_{n_t}^T \tag{E4.17}$$

avec s_{n_t} la plus petite valeur singulière de $\mathbf{W}_t$ et $\mathbf{U}_{n_t}$ (respectivement $\mathbf{V}_{n_t}$) la dernière colonne de $\mathbf{U}$ (respectivement $\mathbf{V}$). $\overline{\mathbf{W}}_t$ est la matrice compatible avec le système d'équation $\mathbf{W}_t\boldsymbol{\xi}_t = \mathbf{0}$ minimisant la perturbation sur l'ensemble des termes de la matrice $\mathbf{W}_t$.

Le vecteur unitaire $\hat{\boldsymbol{\chi}}_t^*$ solution de l'équation

$$\overline{\mathbf{W}}_t\hat{\boldsymbol{\chi}}_t^* = \mathbf{0} \tag{E4.18}$$

est alors donné par la dernière colonne de $\mathbf{V}$, $\hat{\boldsymbol{\chi}}_t^* = \mathbf{V}_{n_t}$.

Il existe un nombre infini de vecteurs $\hat{\boldsymbol{\chi}}_t = \lambda\hat{\boldsymbol{\chi}}_t^*$ solution de l'équation (E4.18). Cependant, nous recherchons la solution à l'équation (E4.14) telle que la dernière composante du vecteur soit 1. Ainsi, la solution retenue est donnée par $\hat{\boldsymbol{\chi}}_t = \hat{\boldsymbol{\chi}}_t^*/\chi_n$, avec χ_n la dernière composante du vecteur $\hat{\boldsymbol{\chi}}_t^*$.

4.2.4 Analyse statistique, paramètres essentiels et moindres carrés totaux pondérés

Un outil statistique indiquant la précision de l'estimation des paramètres identifiés est l'écart-type. Les écarts-types $\sigma_{\hat{\boldsymbol{\chi}}_j}$ $(j = 1, \ldots, 11)$ sur l'estimation des paramètres dynamiques sont calculés en supposant que les perturbations dans le régresseur $\mathbf{W}_t$ sont indépendantes et distribuées de façon uniforme, avec une moyenne nulle et une matrice de covariance telle que :

$$\mathbf{C}_{WW} = \hat{\sigma}_W^2\mathcal{I}_{rn_t} \tag{E4.19}$$

avec $\mathcal{I}_{rn_t}$ la matrice identité de dimension $(rn_t \times rn_t)$. Une estimation non biaisée de l'écart type $\hat{\sigma}_W$ est donnée par l'expression [Van Huffel et Vandewalle, 1991] :

$$\hat{\sigma}_W = \frac{s_{n_t}}{\sqrt{r - n_t}} \tag{E4.20}$$

La matrice de covariance des erreurs d'estimation est donnée par [Van Huffel et Vandewalle, 1991] :

$$\mathbf{C}_{\hat{\chi}\hat{\chi}} \simeq \sigma_W^2\left(1 + ||\hat{\boldsymbol{\chi}}_{1:n-1}||_2^2\right)\left(\overline{\mathbf{W}}_{t_{1:n_t-1}}^T\overline{\mathbf{W}}_{t_{1:n_t-1}}\right)^{-1} \tag{E4.21}$$

avec $\hat{\boldsymbol{\chi}}_{1:n_t-1}$ le vecteur comprenant les $n_t - 1$ premiers coefficients de $\hat{\boldsymbol{\chi}}$ et $\overline{\mathbf{W}}_{t_{1:n_t-1}}$ une matrice composée des $n_t - 1$ premières colonnes de $\overline{\mathbf{W}}_t$. Ensuite, l'écart-type de l'estimation du j-ième paramètre dynamique est donné par l'expression suivante :

$$\sigma_{\hat{\boldsymbol{\chi}}_j} = \sqrt{\mathbf{C}_{\hat{\chi}\hat{\chi}}(j,j)} \tag{E4.22}$$

avec $\mathbf{C}_{\hat{\chi}\hat{\chi}}(j,j)$ le j-ème coefficient diagonal de $\mathbf{C}_{\hat{\chi}\hat{\chi}}$. L'écart-type relatif du j-ème paramètre dynamique est lui donné par :

$$\%\sigma_{\hat{\chi}_j} = 100\sigma_{\hat{\boldsymbol{\chi}}_j}/|\hat{\chi}_j| \tag{E4.23}$$

Pendant la procédure d'identification, certains paramètres sont mal identifiés car ils n'ont pas une contribution suffisamment significative dans le calcul du modèle dynamique. Ces paramètres peuvent être supprimés afin de simplifier le modèle dynamique et augmenter la précision de l'identification des autres paramètres. Ces paramètres sont écartés de façon itérative à partir de la première estimation effectuée. À chaque étape, le paramètre avec le plus grand écart-type est retiré. Une nouvelle estimation du modèle dynamique simplifié (sans ce paramètre) est effectuée. La procédure s'arrête lorsque le ratio entre le plus grand écart-type relatif et l'écart-type relatif le plus faible est inférieur à un seuil r_σ donné $\max(\%\sigma_{\hat{\chi}_j})/\min(\%\sigma_{\hat{\chi}_j}) < r_\sigma$, avec r_σ un ratio choisi idéalement entre 10 et 30 en fonction du niveau de perturbation dans la matrice $\mathbf{W}_t$. Les paramètres restants sont appelés paramètres essentiels et seront suffisants pour décrire le modèle dynamique du quadricoptère.

Afin d'améliorer l'estimation, la méthode des moindres carrés pondérés est utilisée sur les robots manipulateurs classiques (méthode IDIM-WTLS) [Gautier et Poignet, 2001]. Cette méthode est applicable également à notre cas. En effet, l'estimation des paramètres peut ne pas être aussi précise pour chacune des six lignes de l'équation (E4.10), correspondant aux six degrés de liberté du drone dans l'espace. Les lignes de la matrice $\mathbf{W}_t$ peuvent alors être pondérées en fonction de la confiance sur les données mesurées dans chaque direction. Cette étape s'effectue avant le regroupement des matrices $\mathbf{W}_i$ et $\mathbf{w}_{m,i}$ (voir partie 4.2.3) par la procédure suivante :

1. Six matrices $\mathbf{W}_{t,i} = [\mathbf{W}_i - m_d \mathbf{w}_{m,i}]$ (i=1..6), correspondant aux parties du régresseur $\mathbf{W}_t$ associé à chaque direction, sont définies.
2. Pour chaque matrice, l'estimation de la covariance est effectuée par :

$$\hat{\sigma}_{W_i} = \frac{s_{n_t,i}}{\sqrt{r/6 - n_t}} \tag{E4.24}$$

   avec $s_{n_t,i}$ la plus petite valeur singulière de la matrice $\mathbf{W}_{t,i}$. Cette covariance indique l'importance des perturbations de mesure et des erreurs de modèle pour la $i^{\text{ème}}$ ligne du modèle dynamique.
3. Chaque matrice $\mathbf{W}_{t,i}$ est alors respectivement pondérée par l'inverse de la covariance estimée $1/\hat{\sigma}_{W_i}$.
4. Les matrices $\mathbf{W}_{t,i}$ sont concaténées en la matrice $\mathbf{W}_t$.

Le reste de la procédure d'identification est inchangée. La partie suivante décrit les résultats d'identification du drone PELICAN produit par Asctec, qui seront utilisés pour les expérimentations dans le cadre de cette thèse.

4.2.5 Identification du drone PELICAN

La procédure d'identification a été appliquée sur deux drones PELICAN. Sur chacun de ces drones, nous avons fixé le corps rigide permettant la liaison avec la chaîne passive

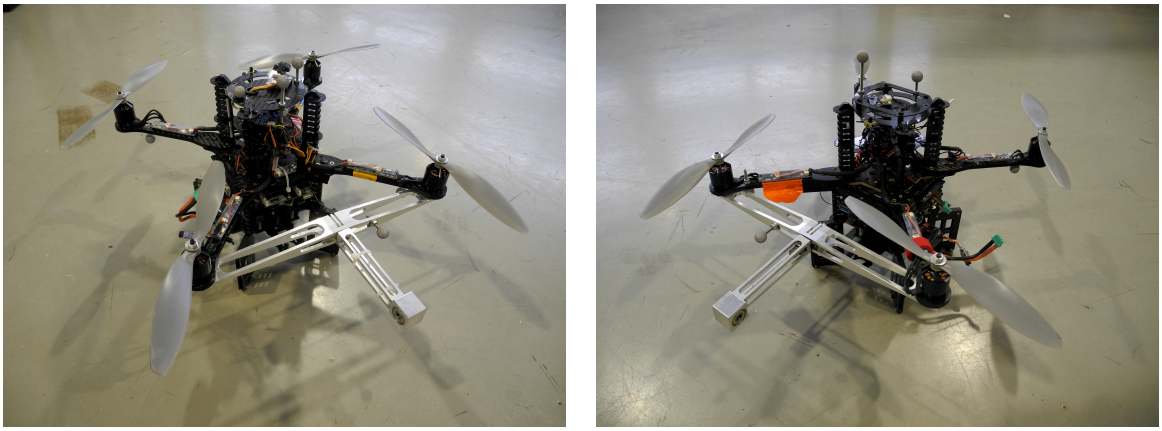


FIGURE 4.10 – Photographies des drones associés aux pièces de liaisons du prototype.

du prototype expérimental (voir partie 4.1.1). Ainsi, les paramètres dynamiques identifiés correspondent au corps respectifs $\mathcal{D}_1$ et $\mathcal{D}_2$ du robot complet (voir Fig. 4.10). Pour le premier drone, l'identification à partir des deux jeux de données (MOCAP et IMU) sera effectuée, ainsi qu'une comparaison des résultats obtenus. Pour le second drone, seule l'identification par le jeu de données issu de l'IMU sera effectuée, du fait d'une contrainte technique sur la vitesse de communication en WIFI ne permettant pas une collecte synchronisée de la vitesse de rotation des hélices et des données issues du MOCAP.

Les masses de référence pour l'identification sont :

- 1.6758 kg pour le drone N°1,
- 1.7069 kg pour le drone N°2.

Résultats d'identification

L'identification des drones a été effectuée selon la procédure décrite précédemment. Les trajectoires ont été réalisées par un opérateur en mode manuel, de façon agressive pour permettre une bonne identification des paramètres inertiels. A noter cependant que l'espace disponible au laboratoire LS2N pour le vol intérieur limite les possibilités sur ce point. Il sera possible dans d'autres conditions expérimentales d'affiner l'identification par des trajectoires plus excitantes.

Les tableaux (T4.1) et (T4.2) donnent les résultats d'identification du drone 1, obtenus respectivement à partir des données MOCAP et IMU. Notons une très bonne identification du coefficient de poussée k_t. Pour les autres paramètres dynamiques, les résultats sont légèrement différents entre les deux méthodes. Cet écart peut s'expliquer par la différence d'origine entre le repère de la centrale inertielle et le repère local défini dans le logiciel de capture de mouvement. Cette hypothèse est confirmée par l'application du théorème de Huygens sur les

Paramètre	Valeur identifiée	$\%\sigma_{\hat{\chi}_j}$
MX_d (kg.m)	9.63e-03	0.88
MY_d (kg.m)	2.18e-02	0.44
MZ_d (kg.m)	-1.60e-01	0.42
XX_d (kg.m^2)	3.65e-02	0.51
YY_d (kg.m^2)	3.25e-02	0.50
ZZ_d (kg.m^2)	-	-
XY_d (kg.m^2)	-	-
YZ_d (kg.m^2)	-	-
XZ_d (kg.m^2)	-	-
k_t (N.rad^{-2})	1.43e-05	0.03
k_d (N.m.rad^{-2})	-	-

Effort	Erreur résiduelle	Erreur résiduelle relative (%)
F_x (N)	0.279	-
F_y (N)	0.205	-
F_z (N)	0.311	1.82
m_x (N.m)	0.065	29.37
m_y (N.m)	0.059	45.69

TABLE 4.1 – Résultats d'identification du drone 1 à partir des données MOCAP

inerties identifiées. Par exemple

$$\begin{aligned} {}^{\text{IMU}}XX_d &= {}^{\text{MOCAP}}XX_d + \left(-{}^{\text{MOCAP}}MY_d^2 - {}^{\text{MOCAP}}MZ_d^2 + {}^{\text{IMU}}MY_d^2 + {}^{\text{IMU}}MZ_d^2\right)/m_d \\ &= 3.21\text{e}{-}02 \text{ kg.m}^2 \end{aligned}$$

fournit un résultat plus proche de celui obtenu par les données IMU (3.11e-02 kg.m^2). Les inerties plus grandes mesurées pour les données MOCAP s'expliquent ainsi par une distance plus importante entre le centre de gravité et l'origine du référentiel que dans le cas du référentiel IMU (1.4 cm environ selon l'axe $\mathbf{z}_d$). Ces mesures peuvent ainsi permettre un recalage du référentiel entre ces deux systèmes de mesure. Notons la particularité du placement de la centrale inertielle du PELICAN, très éloignée du centre de gravité de l'appareil. Il est normalement préférable de rapprocher au maximum cette centrale du centre de gravité de l'appareil afin d'augmenter la qualité des mesures. Les mesures des accélérations ne sont alors pas impactées par les rotations du drone.

La procédure d'identification ne permet pas de déterminer les produits d'inertie. Ce résultat était prévisible du fait des symétries du quadricoptère et le corps rigide ajouté au drone est trop léger pour modifier de façon sensible ces symétries. Ces termes ont alors peu d'impact sur le modèle dynamique.

La méthode ne permet pas non plus d'identifier les paramètres dynamiques associés à l'axe $\mathbf{z}_d$ local du drone (moment d'inertie ZZ_d et coefficient k_d). En effet, dans le cas d'une matrice d'inertie diagonale et de faibles effets gyroscopiques, la dernière ligne de l'équation

Paramètre	Valeur identifiée	$\%\sigma_{\hat{\chi}_j}$
MX_d (kg.m)	8.32e-03	0.69
MY_d (kg.m)	2.09e-02	0.30
MZ_d (kg.m)	-1.35e-01	0.41
XX_d (kg.m^2)	3.11e-02	0.38
YY_d (kg.m^2)	2.66e-02	0.41
ZZ_d (kg.m^2)	-	-
XY_d (kg.m^2)	-	-
YZ_d (kg.m^2)	-	-
XZ_d (kg.m^2)	-	-
k_t (N.rad^{-2})	1.42e-05	0.03
k_d (N.m.rad^{-2})	-	-

Effort	Erreur résiduelle	Erreur résiduelle relative (%)
F_x (N)	0.351	-
F_y (N)	0.302	-
F_z (N)	0.480	2.81
m_x (N.m)	0.064	29.65
m_y (N.m)	0.061	48.60

TABLE 4.2 – Résultats d'identification du drone 1 à partir des données IMU

dynamique de rotation du drone au centre de gravité (E1.7) s'écrit :

$$ZZ_d\dot{\omega}_z = k_d \sum_i -\text{sign}(\omega_i)\omega_i^2 \tag{E4.25}$$

Il s'agit du seul terme contenant les paramètres dynamiques ZZ_d et k_d. Les autres paramètres dynamiques du quadricoptère n'apparaissent pas non plus dans cette équation. Cette indépendance rend impossible l'identification de ces paramètres à partir de la connaissance de la masse seule. Afin d'obtenir une identification complète du drone, il est nécessaire de connaître l'un de ces deux paramètres et de faire une identification spécifique sur l'axe $\mathbf{z}_d$. Dans le cadre de cette thèse, cette donnée n'est pas cruciale car les inerties selon l'axe $\mathbf{z}_d$ des drones sont négligeables devant l'inertie selon l'axe vertical du robot complet et les moments de traînée générés par les drones ne sont pas exploités dans la commande du robot.

Nous constatons également une erreur résiduelle relative élevée pour les moments, liée au fait que les moments restent faibles en moyenne sur la trajectoire exécutée. Une trajectoire avec une dynamique de rotation plus agressive permettrait d'obtenir de meilleurs résultats sur ce point, mais les conditions expérimentales au LS2N ne le permettent pas. L'erreur relative est plus faible pour le moment selon l'axe m_x plus sollicité. C'est sur cet axe qu'agit la charge créée par l'ajout de la liaison avec l'architecture passive du robot complet.

Les données d'identification obtenues pour le drone 2 sont données dans le tableau (T4.3).

Paramètre	Valeur identifiée	$\%\sigma_{\hat{\chi}_j}$
MX_d (kg.m)	-	-
MY_d (kg.m)	-2.92e-02	0.29
MZ_d (kg.m)	-1.38e-01	0.56
XX_d (kg.m^2)	2.97e-02	0.48
YY_d (kg.m^2)	2.64e-02	0.59
ZZ_d (kg.m^2)	-	-
XY_d (kg.m^2)	-	-
YZ_d (kg.m^2)	-	-
XZ_d (kg.m^2)	-	-
k_t (N.rad^{-2})	1.37e-05	0.03
k_d (N.m.rad^{-2})	-	-

Effort	Erreur résiduelle	Erreur résiduelle relative (%)
F_x (N)	0.299	-
F_y (N)	0.238	-
F_z (N)	0.361	2.16
m_x (N.m)	0.056	16.76
m_y (N.m)	0.067	56.73

TABLE 4.3 – Résultats d'identification du drone 2 à partir des données IMU

Validation croisée

Une validation croisée des résultats obtenus est effectuée. Elle consiste à vérifier, sur une seconde trajectoire, la cohérence des résultats d'identification obtenus. Pour chaque ligne du modèle dynamique (E4.2) et (E4.3), exprimée dans le repère local du drone, nous comparons :

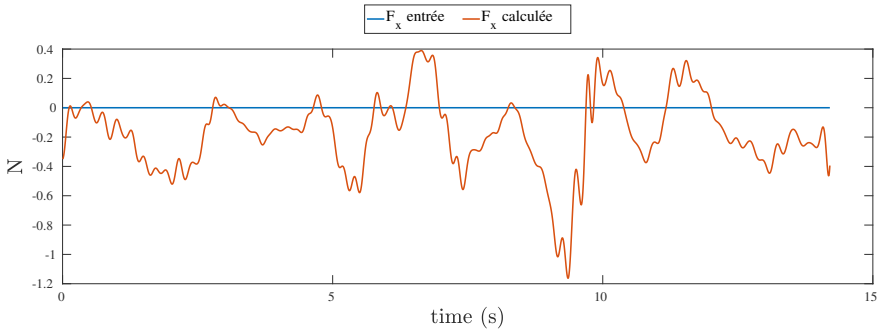
- les efforts en entrée fournis par le drone obtenus à l'aide des coefficients identifiés k_t et k_d et des vitesses de rotation des hélices,
- les efforts théoriques calculés pour obtenir les accélérations, mesurées à l'aide de la masse m_d et des autres paramètres dynamiques identifiés.

La comparaison, pour les données MOCAP, est donnée Fig. 4.11 et Fig. 4.12 et, pour les données IMU, Fig. 4.13 et Fig. 4.14 . Les résultats sont donnés pour le drone 1, ils sont similaires pour le drone 2. Les recollements sont satisfaisants en moyenne (voir tableau (T4.4)). Les erreurs relatives restent importantes sur les moments du fait des faibles valeurs moyennes de ceux-ci pour la trajectoire effectuée. Les valeurs relatives pour les forces selon l'axe $\mathbf{x}_d$ et $\mathbf{y}_d$ n'ont pas de sens car ces forces sont théoriquement de valeur nulle. Enfin, il n'y a pas de recollement du moment selon l'axe $\mathbf{z}_d$ car les paramètres dynamiques de cet axe n'ont pas été identifiés (voir ci-avant).

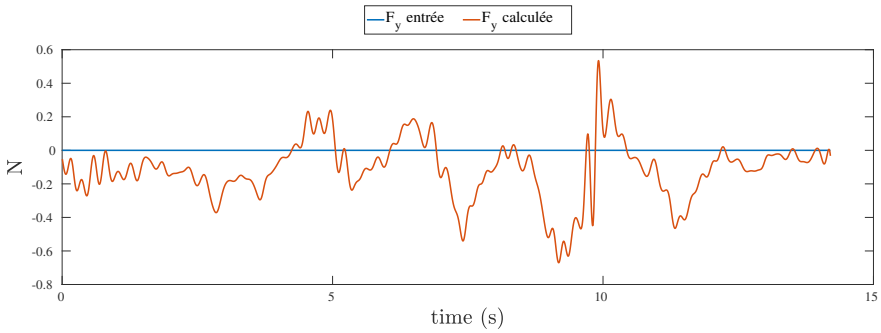
Lors de fortes variations de la dynamique, l'erreur de modèle peut être assez importante, jusqu'à 10 N par exemple pour la force F_z (Fig. 4.13). Cette erreur est essentiellement due à la présence d'un retard d'environ 50 millisecondes entre la valeur de rotation des hélices obtenue et la dynamique observée par les mesures. Deux explications peuvent être

	MOCAP	IMU
F_x (N)	0.241	0.350
F_y (N)	0.160	0.273
F_z (N)	0.444	0.531
$\%F_x$	2.67	5.7
m_x (N.m)	0.040	0.055
$\%m_x$	18.07	23.78
m_y (N.m)	0.092	0.104
$\%m_y$	98.33	116.3

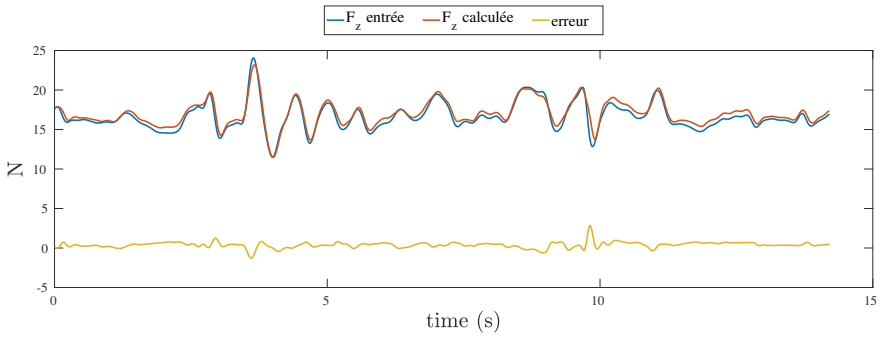
TABLE 4.4 – Erreur sur le modèle lors de la validation croisée.



(a) Force selon l'axe $\mathbf{x}_d$.

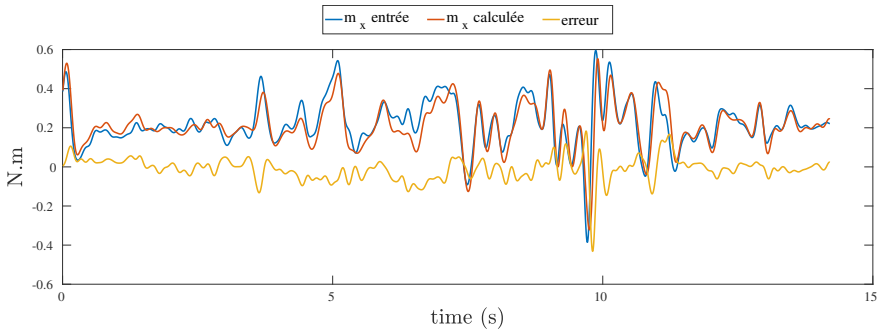


(b) Force selon l'axe $\mathbf{y}_d$.

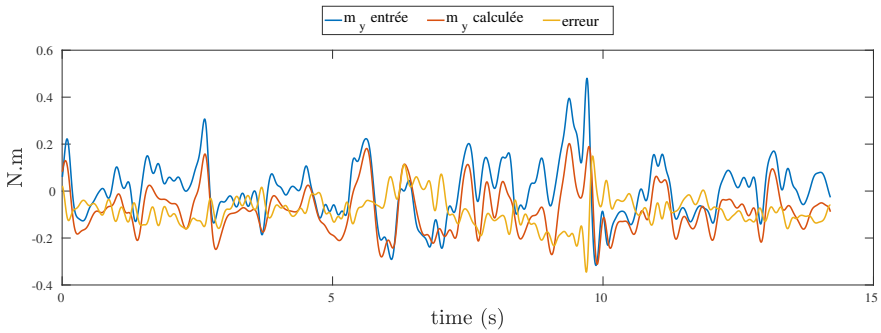


(c) Force selon l'axe $\mathbf{z}_d$.

FIGURE 4.11 – Validation croisée de l'identification du drone 1 à partir des données MOCAP (forces).

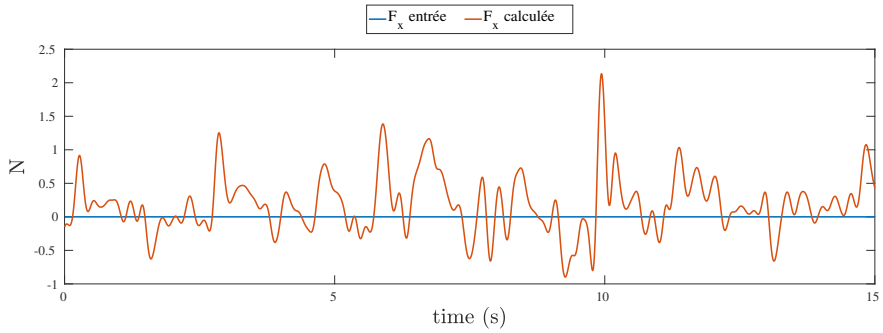


(a) Moment selon l'axe $\mathbf{x}_d$.

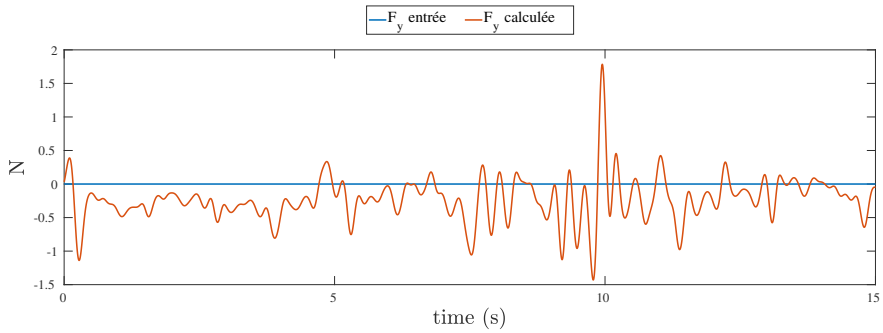


(b) Moment selon l'axe $\mathbf{y}_d$.

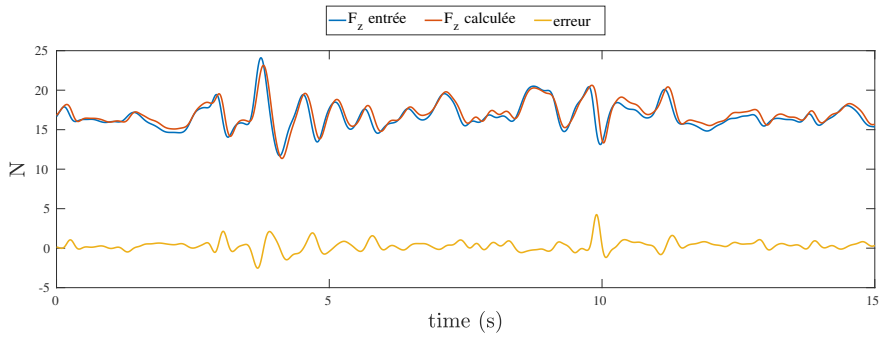
FIGURE 4.12 – Validation croisée de l'identification du drone 1 à partir des données MOCAP (moments).



(a) Force selon l'axe $\mathbf{x}_d$.

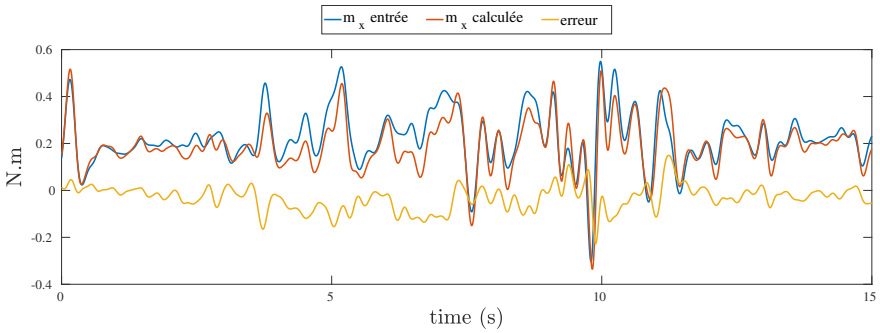


(b) Force selon l'axe $\mathbf{y}_d$.

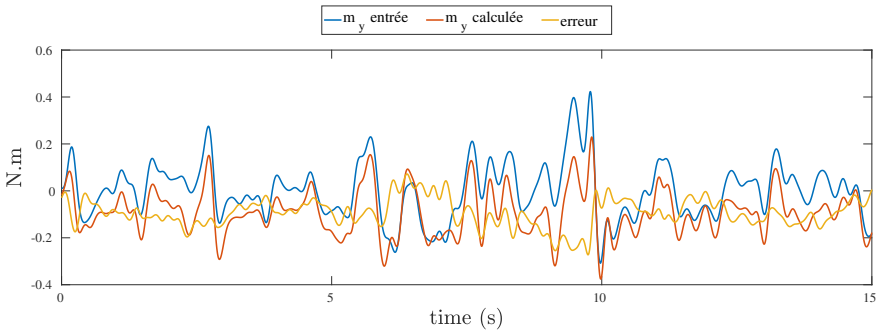


(c) Force selon l'axe $\mathbf{z}_d$.

FIGURE 4.13 – Validation croisée de l'identification du drone 1 à partir des données IMU (forces).



(a) Moment selon l'axe $\mathbf{x}_d$.



(b) Moment selon l'axe $\mathbf{y}_d$.

FIGURE 4.14 – Validation croisée de l'identification du drone 1 à partir des données IMU (moments).

fournies pour ce retard. Il peut être en partie dû à la dynamique transitoire des moteurs mais également à des phénomènes transitoires aérodynamiques lors des changements de vitesse de rotation des hélices. Ce retard amplifie l'erreur de recollement du modèle dynamique. Nous constatons aussi un biais sur le recollement pour le moment selon l'axe $\mathbf{y}_b$. Ce décalage est dû au changement de la batterie entre le vol d'identification et le vol de validation croisée. La position de la batterie n'étant pas strictement la même lorsqu'elle est replacée, elle peut induire une légère modification des paramètres dynamiques du quadricoptère avec un impact particulier pour ce moment.

Les résultats de l'identification des paramètres dynamiques vont servir de référence pour le modèle dynamique utilisé dans la loi de commande du prototype. Pour l'architecture passive, les estimations issues de la CAO sont conservées. Si elle est théoriquement envisageable, l'identification des paramètres de l'architecture passive nécessite d'effectuer des trajectoires excitantes sur le robot complet. Dans le cadre de cette thèse, l'espace disponible ne permet pas de telles manipulations. La partie suivante présente les résultats expérimentaux obtenus sur le prototype.

4.3 Résultats expérimentaux

Le contrôleur utilisé lors de l'expérimentation est celui présenté dans la partie 3.2. Ce contrôleur effectue le suivi d'une trajectoire, définie par la position des articulations ① et ③ du robot dans un plan vertical.

4.3.1 Paramètres du contrôleur

Paramètres physiques du robot

Dans sa dernière version (voir partie 4.1.1), le corps $\mathcal{B}_1$ du prototype a été modifié pour limiter la saturation des actionneurs et permettre ainsi au robot de se stabiliser, notamment dans le mode d'assemblage où les bras sont situés au dessus des drones. Ainsi, les paramètres physiques du robot réel sont les suivants

- la longueur des bras ℓ_1=0.356 m et ℓ_2=0.856 m,
- la distance entre les centres de gravité des drones et leur articulation respective $\ell_{d1} = 0.27365$ m et $\ell_{d2} = 0.27565$ m,
- $d = 0.3$ m la distance entre deux hélices d'un drone.

Les paramètres dynamiques sont issus des données de la CAO pour les bras et des données de l'identification (voir partie 4.2) pour les drones. Ces paramètres sont donnés en table (T4.5). Les coefficients de poussée sont ceux issus de la procédure d'identification $k_{t,1} =$ 1.42e-5 N.rad $^{-2}$ et $k_{t,2} = 1.37\text{e}-5$ N.rad $^{-2}$

	$\mathcal{B}_1$	$\mathcal{B}_2$	$\mathcal{D}_1$	$\mathcal{D}_2$
$m_{\mathcal{B}}$ (kg)	1.29e-01	2.18e-01	1.68e+00	1.71e+00
$MX_{\mathcal{B}}$ (kg.m)	2.73e-02	9.92e-02	4.31e-01	1.11e-03
$MY_{\mathcal{B}}$ (kg.m)	0.00e+00	0.00e+00	3.44e-03	0.00e+00
$MZ_{\mathcal{B}}$ (kg.m)	0.00e+00	0.00e+00	8.32e-03	4.44e-01
$XX_{\mathcal{B}}$ (kg.m^2)	2.68e-05	2.69e-05	1.57e-02	1.16e-01
$YY_{\mathcal{B}}$ (kg.m^2)	8.24e-03	6.63e-02	1.13e-01	1.34e-01
$ZZ_{\mathcal{B}}$ (kg.m^2)	8.24e-03	6.63e-02	1.25e-01	1.50e-02
$XY_{\mathcal{B}}$ (kg.m^2)	0.00e+00	0.00e+00	-8.66e-04	0.00e+00
$XZ_{\mathcal{B}}$ (kg.m^2)	0.00e+00	0.00e+00	-8.13e-04	-2.88e-04
$YZ_{\mathcal{B}}$ (kg.m^2)	0.00e+00	0.00e+00	-6.65e-06	0.00e+00

TABLE 4.5 – Tableau des paramètres dynamique du robot volant à deux drones en expérimentation (dans les repères locaux définis par la convention DHKK).

Gains et filtres

Les gains préalablement sélectionnés en simulation ont été adaptés en conditions expérimentales. Les gains utilisés pour les résultats présentés dans cette partie sont :

- pour la coordonnée x_G (section 3.2.6) : $k_{p,x} = 1$, $k_{d,x} = 2$;
- pour la coordonnée ψ (section 3.2.6) : $k_{p,\psi} = 2$, $k_{d,\psi} = 3$;
- pour la boucle lente (commande adaptative partie 2.4.5) : $\lambda_p = 2$, $\lambda_i = 0.5$, $k_\sigma = 3$ et $\gamma = 0.01$;
- les coefficients diagonaux de Λ, Λ_i, $\mathbf{K}_s$ sont respectivement 10, 2 et 15 pour toutes les coordonnées d'orientation φ_1, φ_2 et θ_2. Pour θ_1, les gains sont respectivement 10, 2 et 9. Le réglage est différent pour θ_1 car la correction de cette coordonnée provoque rapidement la saturation des actionneurs. Les gains ne peuvent donc pas être aussi élevés que pour les autres coordonnées. Les coefficients ϵ_k sont réglés de la façon suivante : $\epsilon_1 = 3.89$ et $\epsilon_2 = \epsilon_3 = \epsilon_4 = 1.33$.

Le filtre sur les orientations désirées θ_1^d et θ_2^d est un filtre passe-bas de type Butterworth d'ordre 4 à fréquence de coupure 1 Hz. Le filtre sur les orientations désirées φ_1^d et φ_2^d est un filtre passe-bas de type Butterworth d'ordre 3 à fréquence de coupure 3 Hz similaire à celui utilisé en simulation.

4.3.2 Résultats expérimentaux

Les résultats sont présentés pour le suivi d'une trajectoire de chacune des articulations ① et ③. La trajectoire proposée est une série de trajectoires polynomiales d'ordre 9 reliant plusieurs points dans le plan vertical (voir Fig. 4.15). La trajectoire telle que définie peut être exécutée par le robot dans les deux modes d'assemblages possibles de l'architecture passive, appelées haut (respectivement bas) si l'articulation centrale ② est située au dessus (resp. en dessous) des drones.

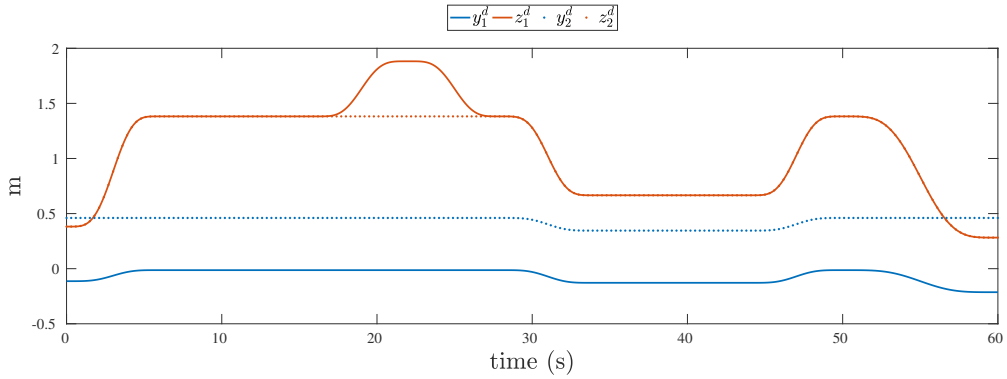


FIGURE 4.15 – Exemple de trajectoire suivie par le robot pour les expérimentations.

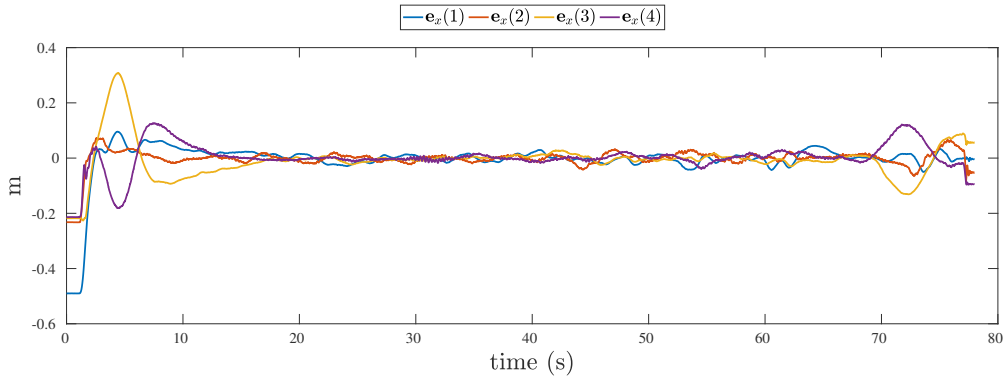
Les erreurs en mode d'assemblage bas pour une trajectoire sont données Fig. 4.16 à Fig. 4.18. La trajectoire est réalisée à plusieurs vitesses d'exécution afin de voir l'évolution de la précision en augmentant la dynamique du système. La trajectoire comprend toujours une phase de décollage et de stabilisation de 16 secondes, ainsi qu'une durée d'atterrissage de 6 secondes. Le cœur de la trajectoire est effectué respectivement en 54 (Fig. 4.16), 34 (Fig. 4.17) et 26 secondes (Fig. 4.18).

Nous observons de bonnes performances du suivi des coordonnées d'orientation. La convergence vers la trajectoire est également assurée. Les performances sont dégradées en phase de décollage et d'atterrissage, le modèle du contrôleur ne prenant pas en compte les effets de sol. Le tableau (T4.6) indique les valeurs moyennes et maximales de la valeur absolue de l'erreur des différentes coordonnées pour les trois trajectoires effectuées. Le suivi de position des articulations se dégrade légèrement avec l'augmentation de la dynamique de la trajectoire mais les performances ne sont pas drastiquement réduites. Le suivi des orientations n'est pas affecté par la dynamique de la trajectoire pour les trajectoires testées.

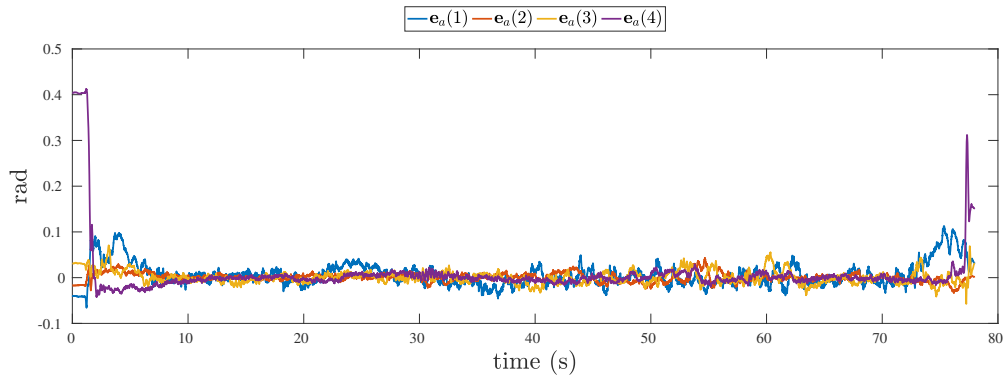
Les performances pour le contrôle latéral sont plus mauvaises pour l'ensemble des trajectoires. Cela n'est pas étonnant car le système n'a pas été conçu pour contrôler précisément ces coordonnées, mais limite les possibilités d'applications réelles du système à deux drones. Les perturbations dans cette direction seront également difficilement gérées.

La figure 4.19 montre les commandes envoyées au moteur pendant le vol. Même sur le résultat filtré pour l'affichage, nous constatons un phénomène de broutement ("chattering"). La limitation de ce broutement reste un enjeu, en obtenant un compromis entre celui-ci et la stabilité du système. Il peut être envisagé d'utiliser des techniques telles qu'un mode glissant d'ordre supérieur [Levant, 1993] pour améliorer les performances du système sur ce point.

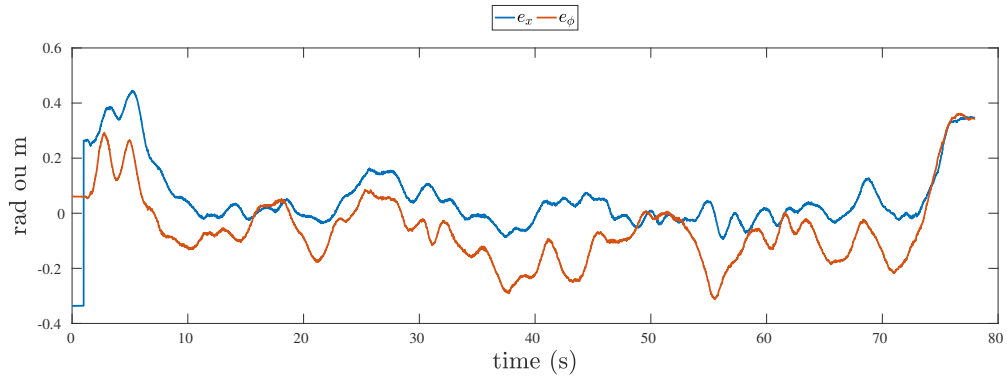
La figure 4.20 montre l'évolution des coefficients d'adaptation pendant la trajectoire effectuée. En dehors des phases de décollage et d'atterrissage où le modèle est très perturbé, nous constatons la convergence des coefficients d'adaptation. La valeur du coefficient pour le drone 1 diminue au cours du vol de manière plus significative que pour le drone 2. Ce résultat s'explique par le fait que le drone 1 embarque l'ordinateur de bord qui consomme beaucoup d'énergie. Il a été constaté lors d'expériences sur le drone seul que la charge de la



(a) Erreur de position des articulations $\mathbf{e}_x$.

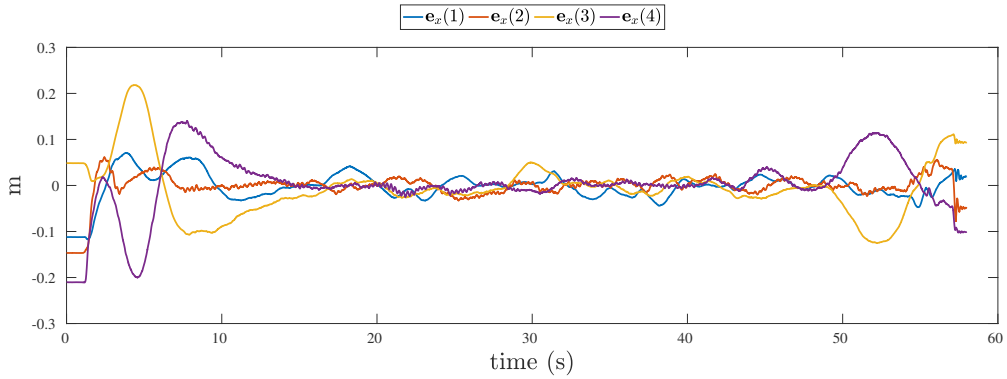


(b) Erreur sur les coordonnées d'orientation $\mathbf{e}_a$.

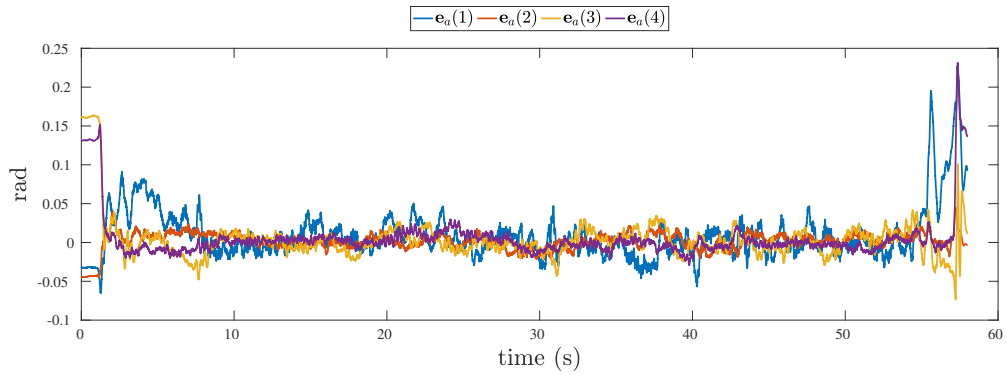


(c) Erreur sur les coordonnées de maintien dans le plan vertical.

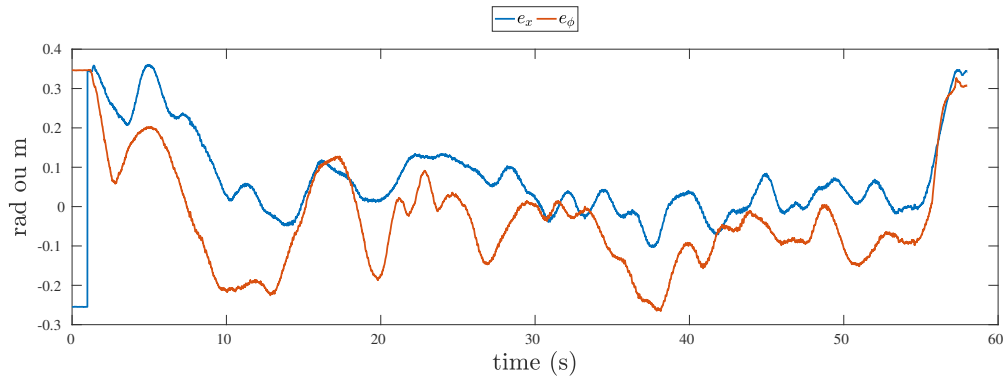
FIGURE 4.16 – Expérimentations : erreurs de suivi de la trajectoire en 54 s.



(a) Erreur de position des articulations $\mathbf{e}_x$.

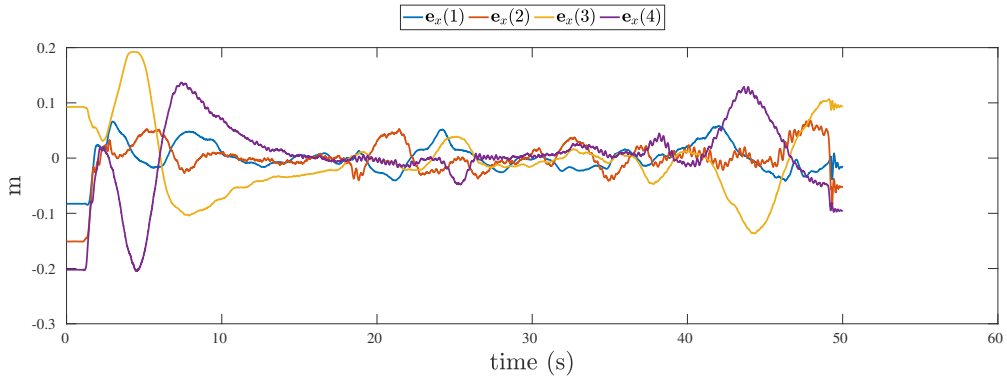


(b) Erreur sur les coordonnées d'orientation $\mathbf{e}_a$.

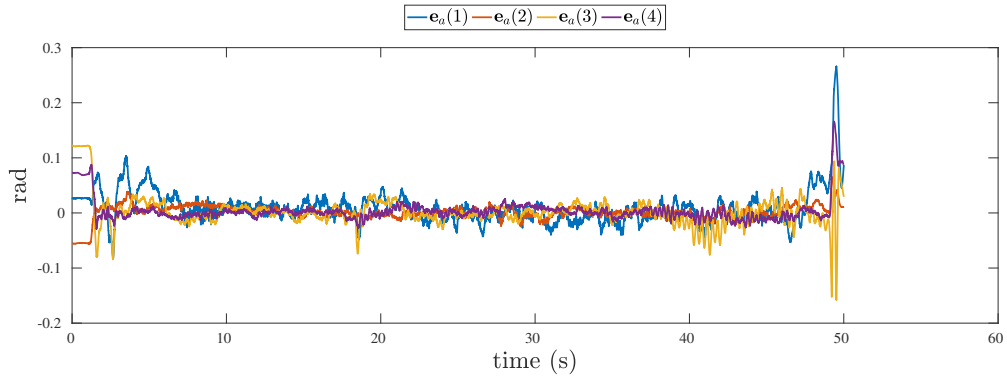


(c) Erreur sur les coordonnées de maintien dans le plan vertical.

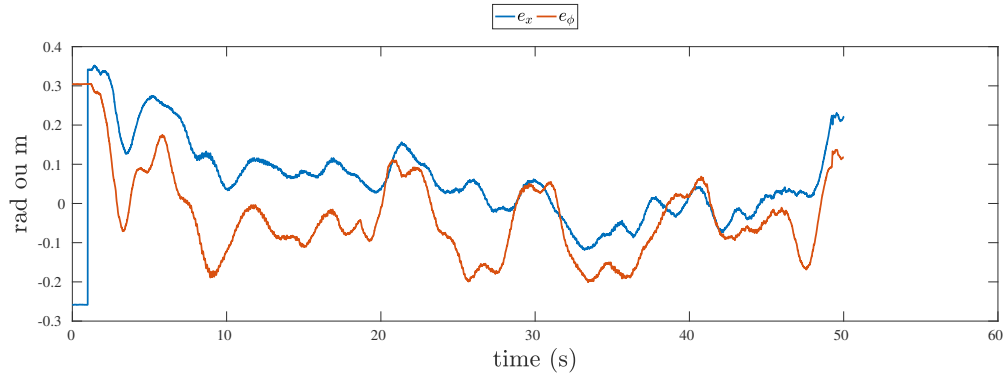
FIGURE 4.17 – Expérimentations : erreurs de suivi de la trajectoire en 34 s.



(a) Erreur de position des articulations $\mathbf{e}_x$.



(b) Erreur sur les coordonnées d'orientation $\mathbf{e}_a$.



(c) Erreur sur les coordonnées de maintien dans le plan vertical.

FIGURE 4.18 – Expérimentations : erreurs de suivi de la trajectoire en 26 s.

	54 s : Err. moy.	Err. max.	**34 s** : Err. moy.	Err. max.	**24 s** : Err. moy.	Err. max.
y_1	1.25e-02	4.44e-02	1.39e-02	4.41e-02	1.62e-02	5.84e-02
z_1	9.70e-03	4.22e-02	1.00e-02	3.26e-02	1.66e-02	5.22e-02
y_2	8.10e-03	3.79e-02	1.40e-02	4.99e-02	1.34e-02	4.73e-02
z_2	8.07e-03	4.36e-02	8.02e-03	3.99e-02	1.08e-02	5.10e-02
θ_1	1.06e-01	3.12e-01	7.68e-02	2.66e-01	9.18e-02	2.01e-01
θ_2	1.34e-02	4.93e-02	1.33e-02	5.68e-02	1.35e-02	4.84e-02
φ_1	6.39e-03	4.31e-02	6.81e-03	2.26e-02	5.31e-03	2.62e-02
φ_2	9.47e-03	5.57e-02	1.03e-02	4.34e-02	1.23e-02	7.64e-02
x	4.38e-02	1.63e-01	5.02e-02	1.34e-01	5.37e-02	1.56e-01
ψ	1.06e-01	3.12e-01	7.68e-02	2.66e-01	9.18e-02	2.01e-01

TABLE 4.6 – Valeurs moyennes et maximales de la valeur absolue de l'erreur à différentes vitesses d'exécution d'une trajectoire pour le robot à deux drones (unités m pour les positions et rad pour les orientations).

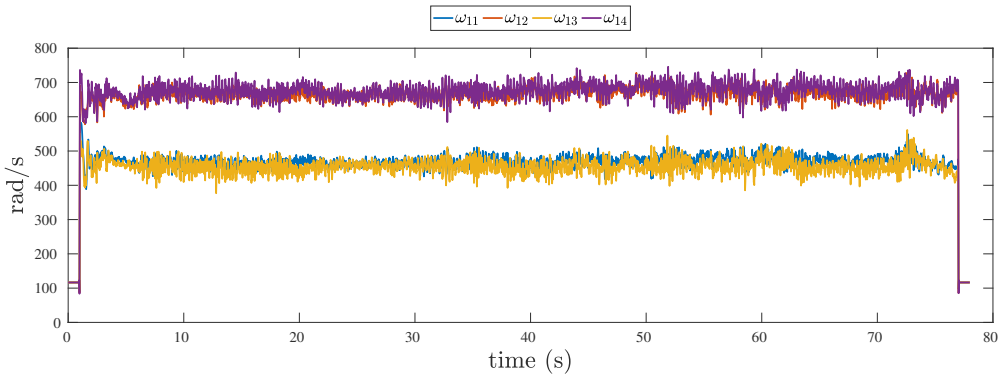


FIGURE 4.19 – Expérimentations : commande envoyée aux moteurs du drone 1 pour la trajectoire en 54 secondes (filtrée).

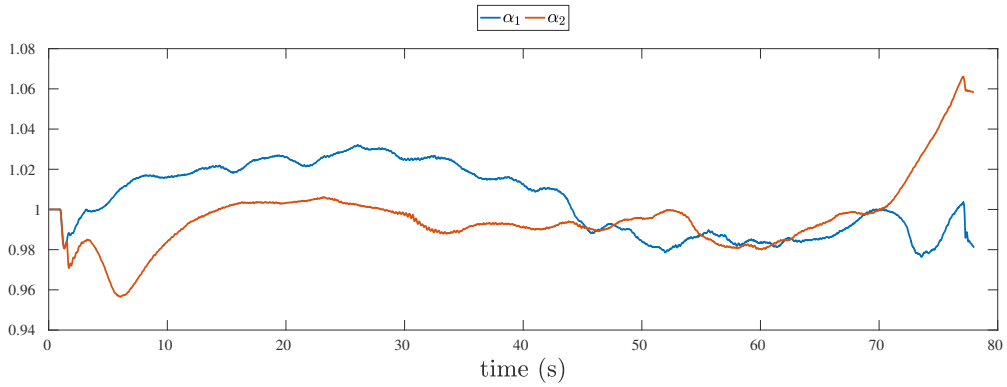
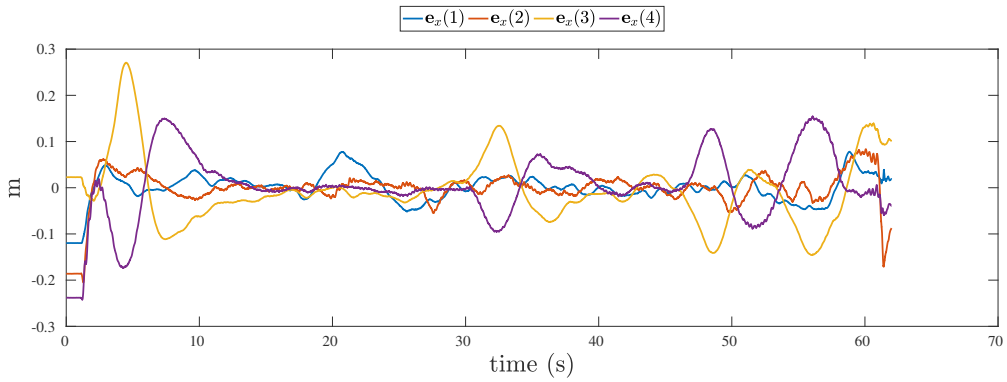


FIGURE 4.20 – Expérimentations : évolution des coefficients d'adaptation pour la trajectoire en 54 secondes.

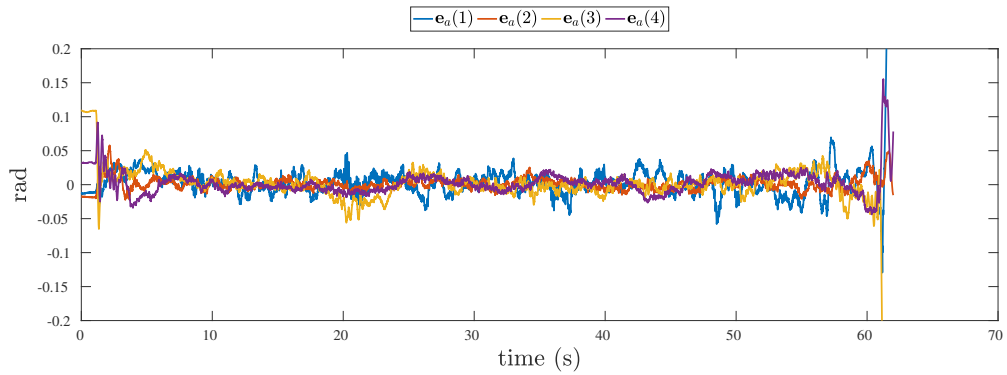
batterie a un impact significatif sur les performances de poussées du drone. C'est ce constat qui est à l'origine de la mise en œuvre des coefficients d'adaptation.

La figure 4.21 montre les résultats obtenus pour une trajectoire exécutée en mode d'assemblage haut. Le maintien dans le plan vertical est moins performant dans ce cas. La qualité de suivi de trajectoire est aussi affectée. Le mode d'assemblage haut est naturellement plus instable et les performances en sont affectées.

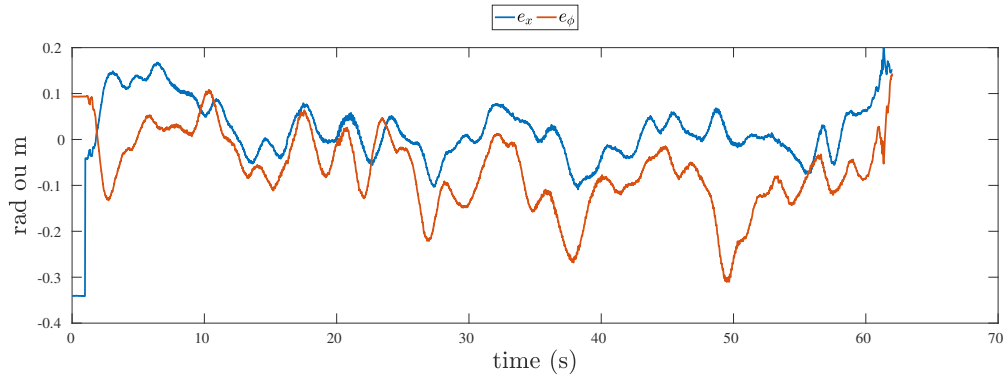
Une vidéo montrant l'ensemble de ces expériences est disponible au lien suivant https://uncloud.univ-nantes.fr/index.php/s/MPCSNEKEFxYfB8Z



(a) Erreur de position des articulations $\mathbf{e}_x$.



(b) Erreur sur les coordonnées d'orientation $\mathbf{e}_a$.



(c) Erreur sur les coordonnées de maintient dans le plan vertical.

FIGURE 4.21 – Expérimentations : erreurs de suivi de la trajectoire en mode d'assemblage haut.

4.4 Conclusion

Ce chapitre a présenté les résultats expérimentaux obtenus lors de cette thèse. L'identification des paramètres dynamiques des drones, nécessaires pour le contrôleur, a été effectuée à l'aide d'une méthode novatrice pour les quadricoptères, inspirée de l'identification des robots séries.

Les résultats montrent qu'il a été possible d'effectuer un suivi de trajectoire pour les deux modes d'assemblages possibles de l'architecture passive. La loi de commande conçue pour ce robot permet un contrôle dans les conditions d'expérimentations réelles. Cependant, les performances nécessitent une amélioration afin de parvenir à un niveau de précision suffisant pour la manipulation. Certaines améliorations peuvent être effectuées par :

- une maîtrise plus fine de la commande des moteurs et du retour de la vitesse réelle que ne le permet la plate-forme commerciale PELICAN,
- une amélioration de l'estimation de position et d'orientation par des capteurs IMU plus fiables que ceux du PELICAN (bruit et biais importants constatés),
- une loi de commande utilisant un mode glissant du second ordre, pour diminuer le phénomène de broutement.

Cependant, certaines limites restent intrinsèques à la conception du robot. Notamment :

- le maintien dans le plan vertical est complexe pour ce robot du fait de l'absence d'actionnement direct dans la direction $\mathbf{x}$,
- le fait que les centres de gravité des drones ne soient pas confondus avec leur articulation respective est une source de perturbations pour le contrôleur (voir partie 3.2.4).

Le robot à trois drones, présenté dans la partie 3.1, répond à ces limites de conception mais, faute de temps, le prototype expérimental n'a pas pu être développé dans le cadre de cette thèse. Le développement d'un tel prototype, basé sur un modèle de quadricoptère plus adéquat, permettra d'évaluer les performances réelles de ce type de système.

Ces considérations concluent le chapitre pour les résultats expérimentaux. Le chapitre suivant présente une technique de traversée des singularités qui peuvent se présenter dans certaines configurations de l'architecture passive. Cette technique permet de reconfigurer le robot d'un mode d'assemblage à l'autre en vol.

Chapitre 5

Singularités

Sommaire

5.1 Singularités du modèle dynamique inverse et critère de traversée 152

5.1.1 Singularités et modèle dynamique 152

5.1.2 Critère de traversée de la singularité 155

5.2 Commande pour la traversée de la singularité 156

5.2.1 Gestion de l'erreur de suivi aux alentours de la singularité 156

5.2.2 Cas de la commande sur les coordonnées cartésiennes des drones . 159

5.3 Simulation de traversée d'une singularité 162

5.3.1 Trajectoire de traversée 162

5.3.2 Résultats de simulation 165

5.4 Conclusion 169

Ce chapitre traite des singularités du robot parallèle volant. Une étude de ces singularités et de leur impact sur le modèle dynamique est présentée dans une première partie. Elle est à rapprocher des travaux existants sur les robots parallèles décrits en partie 1.3.4 et consiste à transposer ces résultats à notre mécanisme. Une seconde partie détaille les modifications de la loi de commande spécifiques à la traversée de la singularité. Cette adaptation est issue de travaux préliminaires sur un mécanisme à cinq barres réalisés dans le cadre de cette thèse. Ces travaux, généralisation de l'étude effectuée dans [Pagis et al., 2015], ne sont pas développés dans ce chapitre mais la publication associée est fournie en annexe C. Enfin, une troisième partie présente les résultats d'une simulation, effectuée à l'aide du logiciel ADAMS, de traversée de la singularité.

5.1 Singularités du modèle dynamique inverse et critère de traversée

5.1.1 Singularités et modèle dynamique

Dans le chapitre 2, nous avons établi le modèle dynamique du robot parallèle volant comme un ensemble de deux systèmes d'équations (E2.17) et (E2.22) que nous rappelons ici.

$$\mathbf{M}_t\ddot{\mathbf{q}}_p + \mathbf{c}_t = \mathbf{J}_p^T\mathbf{R}_t\mathbf{f}$$
$$\mathbf{M}_a\ddot{\mathbf{q}}_a + \mathbf{c}_a = \mathbf{m}$$

Le modèle dynamique inverse exprime les efforts à appliquer en entrée du système pour obtenir des vecteurs de coordonnées d'accélérations $\ddot{\mathbf{q}}_p$ et $\ddot{\mathbf{q}}_a$ donnés. Dans le cas de la dynamique d'orientation, l'équation (E2.22) donne immédiatement le résultat. Il n'y a pas de dégénérescence possible du modèle dynamique inverse. Dans le cas de la dynamique de l'architecture passive cependant, le rang de la matrice $\mathbf{J}_p^T$ est critique dans le calcul du modèle dynamique inverse. Le traitement de la matrice $\mathbf{R}_t$ a été détaillé dans la partie 2.4.1 et n'intervient pas pour cette étude. Il s'agit ici de répondre à la question : existe-t-il un vecteur $\mathbf{f}_J = \mathbf{R}_t\mathbf{f}$ solution de l'équation :

$$\mathbf{J}_p^T\mathbf{f}_J = \mathbf{M}_t\ddot{\mathbf{q}}_p + \mathbf{c}_t \tag{E5.1}$$

La réponse à cette question passe par l'étude de la matrice $\mathbf{J}_p$, définie à l'équation (E2.17) et composée, pour rappel, de la concaténation des matrices $\mathbf{J}_{v,i}$ vérifiant

$${}^o\mathbf{v}_i = \mathbf{J}_{v,i}\dot{\mathbf{q}}_p \tag{E5.2}$$

avec ${}^o\mathbf{v}_i$ la vitesse du point O_i situé à l'extrémité de la jambe i de l'architecture passive (Fig. 5.1). La matrice $\mathbf{J}_p$ est une matrice de dimension $(3n \times n_p)$ avec n_p le nombre de degrés de liberté de l'architecture passive et n le nombre de drones actionneurs.

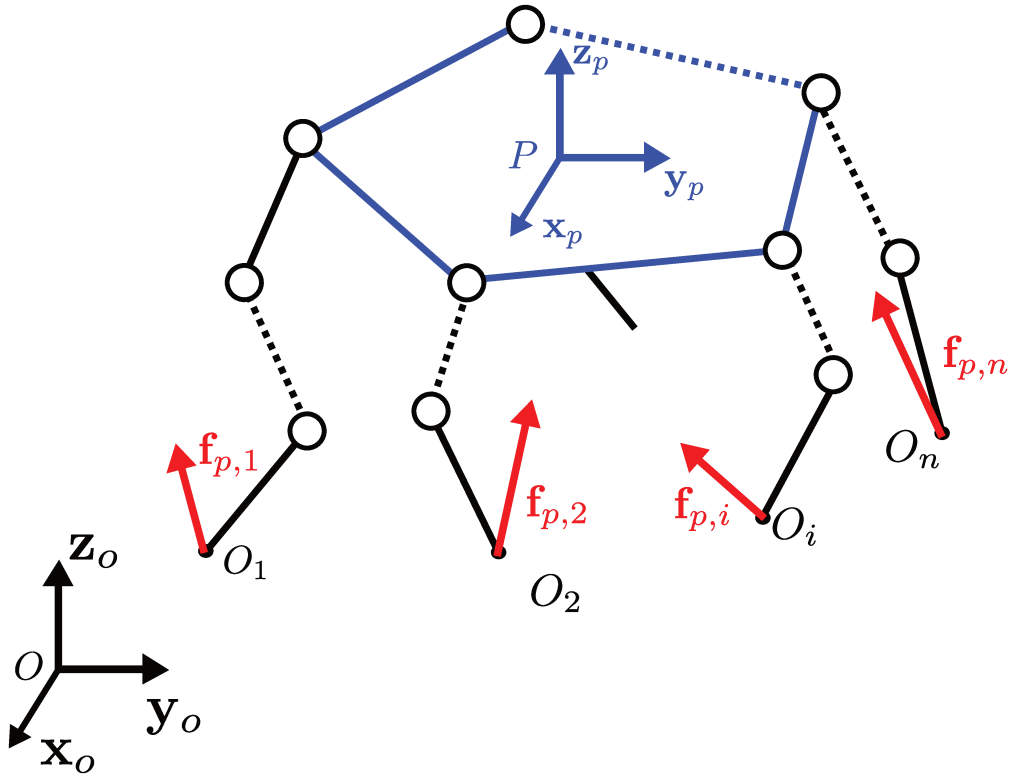


FIGURE 5.1 – Architecture passive soumise aux forces $\mathbf{f}_{p,i}$ appliquées aux points O_i

Résoudre l'équation (E5.1) revient à étudier le modèle dynamique de l'architecture passive seule, où l'on applique une force pure $\mathbf{f}_{p,i}$ en chacun des points O_i (Fig. 5.1). Nous supposerons pour la suite de cette partie que nous disposons au moins d'autant de degrés d'actionnement au travers des forces $\mathbf{f}_{p,i}$ que de degrés de liberté de la structure passive, i.e. $3n \geq n_p$.

L'étude du rang de $\mathbf{J}_p$ permet de savoir si l'équation (E5.1) possède au moins une solution. Si le rang de la matrice $\mathbf{J}_p$ est plein, alors une solution de l'équation (E5.1) est donnée par :

$$\mathbf{f}_J = (\mathbf{J}_p^T)^+(\mathbf{M}_t\ddot{\mathbf{q}}_p + \mathbf{c}_t) \tag{E5.3}$$

avec $(\mathbf{J}_p^T)^+ = \mathbf{J}_p(\mathbf{J}_p^T\mathbf{J}_p)^{-1}$ la pseudo-inverse de Moore-Penrose de la matrice $\mathbf{J}_p^T$. Si le robot possède autant de drones que de degrés de liberté de l'architecture passive ($3n = n_p$), nous avons $(\mathbf{J}_p^T)^+ = (\mathbf{J}_p^T)^{-1}$ et la solution donnée par l'équation (E5.3) est unique. Si $3n$ est strictement supérieur à n_p, le système est sur-actionné [1]. La solution (E5.3) est alors la solution minimisant la norme 2 du vecteur $\mathbf{f}_J$.

Si le rang de $\mathbf{J}_p$ n'est pas plein, alors il existe un vecteur $\dot{\mathbf{q}}_{p,s}$ non nul tel que $\mathbf{J}_p\dot{\mathbf{q}}_{p,s} = \mathbf{0}$. C'est-à-dire qu'il existe une configuration de l'architecture passive telle que, même si la position des drones actionneurs est fixe dans l'espace, un mouvement de l'architecture passive

1. Il faut comprendre ici que le sous-système considéré (architecture passive soumise à des forces pures) est sur-actionné. Cette propriété ne peut pas être étendue au robot complet du fait du sous-actionnement des quadricoptères.

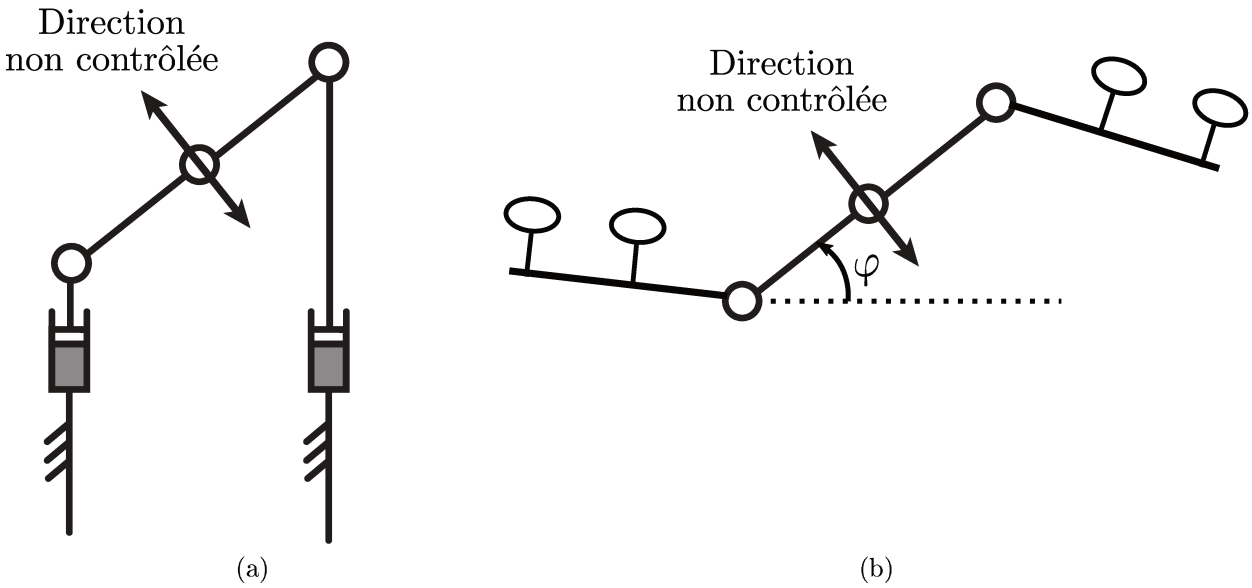


FIGURE 5.2 – Comparaison d'une singularité de Type 2 du Biglide (a) et d'une singularité du robot parallèle volant à deux drones (b).

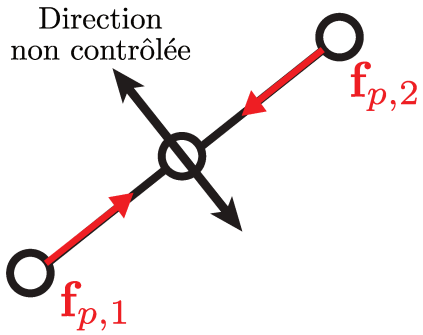


FIGURE 5.3 – Les forces $\mathbf{f}_{p,1}$ et $\mathbf{f}_{p,2}$ opposées et de même norme ne génèrent aucune dynamique sur l'architecture passive.

du robot est localement possible. Ce type de singularité faisant apparaître un mouvement incontrôlé sur l'architecture passive est similaire à une singularité de Type 2 d'un robot parallèle. La figure 5.2 illustre cette similitude. Comme dans le cas les singularités de Type 2 (voir partie 1.3.4), une singularité de la matrice $\mathbf{J}_p$ entraîne :

- le modèle dynamique inverse n'admet plus de solution finie. La norme de la solution (E5.3) diverge à l'approche de la singularité, car $\det(\mathbf{J}_p)$ est proche de zéro.
- localement, une surcontrainte apparaît dans le mécanisme. En effet, il existe un vecteur $\mathbf{f}_{J,s}$ non nul tel que $\mathbf{J}_p^T \mathbf{f}_{J,s} = \mathbf{0}$. Les forces données par le vecteur $\mathbf{f}_{J,s}$ peuvent être appliquées aux extrémités des jambes de l'architecture passive sans effet sur la dynamique de celle-ci (exemple Fig. 5.3).

5.1.2 Critère de traversée de la singularité

De façon similaire aux études sur les singularités de Type 2, il est possible de déterminer un critère garantissant une solution au modèle dynamique inverse. Pour la suite de ce chapitre, nous supposerons que le rang de la matrice $\mathbf{J}_p$ est $(n_p - 1)$ en singularité [2]. Ainsi, le noyau de la matrice $\mathbf{J}_p$ est de dimension 1. Tout vecteur $\boldsymbol{\nu}$ tel que $\mathbf{J}_p\boldsymbol{\nu} = \mathbf{0}$ sera alors proportionnel à $\dot{\mathbf{q}}_{p,s}$.

Afin d'obtenir le critère dynamique, le membre de gauche de l'équation (E5.1) est multiplié par le vecteur $\dot{\mathbf{q}}_{p,s}^T$ appartenant au noyau de la matrice $\mathbf{J}_p$:

$$\dot{\mathbf{q}}_{p,s}^T \mathbf{J}_p^T = \mathbf{0} \tag{E5.4}$$

Le membre de droite de l'équation (E5.1) doit alors vérifier :

$$\dot{\mathbf{q}}_{p,s}^T (\mathbf{M}_t \ddot{\mathbf{q}}_p + \mathbf{c}_t) = \mathbf{0} \tag{E5.5}$$

$\mathbf{M}_t\ddot{\mathbf{q}}_p + \mathbf{c}_t$ représente des efforts virtuels produits par l'inertie et la gravité dans la structure passive. La puissance fournie par ces efforts le long de la direction non contrôlée doit être nulle en singularité. L'équation (E5.5) est un critère sur le vecteur d'accélération $\ddot{\mathbf{q}}_p$ nécessaire pour que l'équation (E5.1) admette une solution. Ce critère est également suffisant car si $\ddot{\mathbf{q}}_p$ le vérifie, alors le vecteur $\mathbf{M}_t\ddot{\mathbf{q}}_p + \mathbf{c}_t$ appartient à l'orthogonal du noyau de $\mathbf{J}_p$. Ainsi, $\mathbf{M}_t\ddot{\mathbf{q}}_p + \mathbf{c}_t$ appartient à l'image de $\mathbf{J}_p^T$ et, en conséquence, l'équation (E5.1) admet au moins une solution. Ce raisonnement est le même que celui pour les singularités de Type 2, détaillé dans la partie 1.3.4.

Une trajectoire définie sur les coordonnées $\mathbf{q}_p$, respectant le critère (E5.5) aux alentours de la singularité, n'entraînera pas de divergence des couples calculés par l'équation (E5.3). Ainsi, cette trajectoire peut permettre la traversée de la singularité par le robot.

Exemple de configuration singulière du robot parallèle volant à deux drones

Dans le cas du robot parallèle volant à deux drones, la matrice $\mathbf{J}_p$ E3.27 s'écrit :

$$\mathbf{J}_p = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 1 & 0 & -\ell_1 S_\varphi - \ell_2 S_{12} & -\ell_2 S_{12} \\ 0 & 1 & \ell_1 C_\varphi + \ell_2 C_{12} & \ell_2 C_{12} \end{bmatrix}$$

2. Tous les résultats et preuves de ce chapitre et de l'annexe peuvent être aisément étendus aux cas où le rang de la matrice $\mathbf{J}_p$ est strictement inférieur à $n-1$ en considérant une base du noyau de $\mathbf{J}_p$ en remplacement du seul vecteur $\dot{\mathbf{q}}_{p,s}$. Dans ce cas, nous obtiendrons $n - \text{rang}(\mathbf{J}_p)$ équations pour le critère de singularité. Ce cas général n'est pas développé car il ne s'est pas présenté dans les exemples de robots abordés et compliquerait inutilement la lecture de ce chapitre.

Cette matrice est singulière pour $q_2 = k\pi$, k entier. Une configuration avec $q_2 = \pi$ n'est pas envisageable sur le robot réel (les deux bras sont alors superposés, il y a autocollision). La matrice $\mathbf{J}_p$ pour $q_2 = 0$ s'écrit :

$$\mathbf{J}_p = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 1 & 0 & -(\ell_1+\ell_2)S_\varphi & -\ell_2 S_\varphi \\ 0 & 1 & (\ell_1+\ell_2)C_\varphi & \ell_2 C_\varphi \end{bmatrix}$$

Le vecteur $\dot{\mathbf{q}}_s = [0\ 0\ -1/(\ell_1+\ell_2)\ 1/\ell_2]^T$ est alors un vecteur de vitesse non nul du noyau de $\mathbf{J_p}$. Supposons l'architecture passive en configuration singulière avec pour vitesse le vecteur $\dot{\mathbf{q}}_s$. Les vitesses des articulations ①, ② et ③ (voir Fig. 3.19 p.105) dans le plan vertical sont données respectivement par :

$$\mathbf{v}_1 = \begin{bmatrix} \dot{x} \\ \dot{y} \end{bmatrix} = \mathbf{0}$$

$$\mathbf{v}_2 = \begin{bmatrix} \dot{x} - \ell_1 S_\varphi \dot{\varphi} \\ \dot{y} + \ell_1 C_\varphi \dot{\varphi} \end{bmatrix} = \begin{bmatrix} \ell_1/(\ell_1+\ell_2)S_\varphi \\ -\ell_1/(\ell_1+\ell_2)C_\varphi \end{bmatrix}$$

$$\mathbf{v}_3 = \begin{bmatrix} \dot{x} - (\ell_1+\ell_2)S_\varphi\dot{\varphi} - \ell_2 S_\varphi \dot{q}_2 \\ \dot{y} + (\ell_1+\ell_2)C_\varphi\dot{\varphi} + \ell_2 C_\varphi \dot{q}_2 \end{bmatrix} = \mathbf{0}$$

La direction non contrôlée correspond à un mouvement local de l'articulation ② orthogonal à l'alignement des bras comme illustré en Fig. 5.2.

5.2 Commande pour la traversée de la singularité

Dans cette partie, considérons que l'on souhaite commander un robot parallèle volant sur une trajectoire traversant la singularité. La loi de commande utilisée est similaire à celui décrite dans la partie 2.4. Cependant, certaines adaptations sont nécessaires pour effectuer une traversée de singularité en conditions réelles.

5.2.1 Gestion de l'erreur de suivi aux alentours de la singularité

Il n'est pas suffisant d'établir une trajectoire respectant le critère de traversée de singularité pour parvenir à la traversée effective en conditions expérimentales. En effet, le critère de traversée garantit que les couples en entrée ne vont pas diverger si la trajectoire est strictement respectée. Or, dans le cadre d'une expérimentation réelle, il existe toujours une erreur de suivi de la trajectoire. Cette erreur sera amplifiée numériquement par l'inversion de la matrice $\mathbf{J}_p^T$ (voir équation E5.3) dans le calcul du modèle dynamique inverse et peut amener la divergence des couples calculés par l'équation (E2.29) correspondant à l'étape d'inversion de la dynamique effectuée par la commande. L'équation (E2.29) sous sa forme développée

s'écrit :

$$\boldsymbol{\nu}_J = (\mathbf{J}_p^T)^+ \boldsymbol{\nu} = (\mathbf{J}_p^T)^+ \left(\mathbf{M}_t \left(\ddot{\mathbf{q}}_p^d - \mathbf{K}_p \mathbf{e}_p - \mathbf{K}_d \dot{\mathbf{e}}_p - \mathbf{K}_i \int \mathbf{e}_p \right) + \mathbf{c}_t \right) \tag{E5.6}$$

Supposons que nous ayons défini une trajectoire respectant le critère de traversée et qu'ainsi l'équation

$$\mathbf{J}_p^T \boldsymbol{\nu}_J = \left(\mathbf{M}_t \ddot{\mathbf{q}}_p^d + \mathbf{c}_t \right) \tag{E5.7}$$

admette une solution exacte. L'ajout du terme de correction dans la commande (E5.6) ne permet plus d'assurer que le résultat ne diverge pas. Afin d'éviter ce problème, nous allons projeter orthogonalement le terme $\boldsymbol{\nu}$ dans l'image de la matrice $\mathbf{J}_p^T$. Ainsi, nous obtiendrons le vecteur le plus proche de $\boldsymbol{\nu}$ au sens de la norme 2 compatible avec l'équation :

$$\mathbf{J}_p^T \boldsymbol{\nu}_J = \boldsymbol{\nu} \tag{E5.8}$$

Pour obtenir cette projection, nous allons utiliser la décomposition en valeur singulières (SVD) de la matrice $\mathbf{J}_p$. Celle-ci s'écrit :

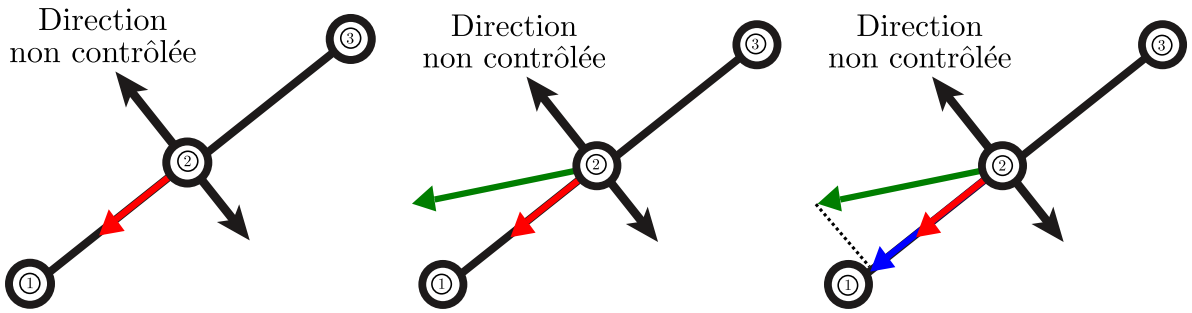
$$\mathbf{J}_p = \mathbf{U}\mathbf{S}\mathbf{V}^T \tag{E5.9}$$

avec $\mathbf{S}$ une matrice diagonale positive contenant les valeurs ordonnées des valeurs singulières de la matrice $\mathbf{J}_p$. $\mathbf{U}$ et $\mathbf{V}$ sont deux matrices unitaires. Au niveau de la singularité, la valeur singulière la plus faible est 0. En multipliant l'équation (E5.8) par $\mathbf{V}^T$ nous obtenons :

$$\mathbf{S}\mathbf{U}^T \boldsymbol{\nu}_J = \mathbf{V}^T \boldsymbol{\nu} \tag{E5.10}$$

La dernière ligne du membre de gauche de ce système d'équation est nulle (la dernière valeur diagonale de $\mathbf{S}$ est 0). Ainsi, pour que l'équation (E5.10) admette une solution, il est nécessaire que la dernière ligne du second membre $\mathbf{V}^T\boldsymbol{\nu}$ soit nulle également. Nous noterons $\mathbf{V}_0$ la matrice composée des $(n_p - 1)$ premières colonnes de $\mathbf{V}$ et d'une dernière colonne nulle. Ainsi, nous remplaçons le membre $\mathbf{V}^T\boldsymbol{\nu}$ par $\mathbf{V}_0^T\boldsymbol{\nu}$ dans l'équation (E5.10). L'opération revient à projeter orthogonalement le vecteur $\boldsymbol{\nu}$ dans l'image de $\mathbf{J}_p^T$ car les $(n_p - 1)$ premières lignes de $\mathbf{V}^T$ forment une base orthonormale de cet espace vectoriel. La problématique et la projection effectuée sont illustrées Fig. 5.4. Considérons pour cet exemple le modèle du robot parallèle volant à deux drones sans tenir compte de la gravité. Dans cet exemple simplifié, le critère de singularité revient à : "l'accélération de l'articulation ② doit être orthogonale à la direction non contrôlée". La figure illustre schématiquement la projection effectuée dans les coordonnées cartésiennes de l'articulation ②.

Afin d'éviter les discontinuités, une transition est effectuée à l'approche de la singularité. Cette transition doit être suffisamment douce pour ne pas perturber le fonctionnement de la commande. Afin de caractériser la "distance" à la singularité, nous allons utiliser $\kappa(\mathbf{J}_p)$



(a) Accélération obtenue par la trajectoire.

(b) Accélération obtenue avec les termes de correction de la loi de commande.

(c) Projection de l'accélération afin de respecter le critère de singularité.

FIGURE 5.4 – Accélération et projection de l'accélération de l'articulation ② en singularité (représentation dans les coordonnées cartésiennes de l'articulation ②, absence de gravité).

le conditionnement de la matrice $\mathbf{J}_p$ défini comme le rapport entre sa valeur propre la plus élevée et sa valeur propre la plus faible. A l'approche de la singularité $1/\kappa(\mathbf{J}_p)$ tend vers 0. Ainsi, nous définissons une fonction σ_s par :

$$\sigma_s(\mathbf{J}_p) = \begin{cases} 1 \text{ si } \dfrac{1}{\kappa(\mathbf{J}_p)} > n_s\epsilon, \\ \dfrac{1-\kappa(\mathbf{J}_p)\epsilon}{(n_s-1)\kappa(\mathbf{J}_p)\epsilon} \text{ si } \epsilon \leq \dfrac{1}{\kappa(\mathbf{J}_p)} \leq n_s\epsilon, \\ 0 \text{ si } \dfrac{1}{\kappa(\mathbf{J}_p)} < \epsilon. \end{cases} \tag{E5.11}$$

ϵ représente une "distance" à la singularité en dessous de laquelle la fonction σ_s est égale à 0 et n_s est un multiplicateur tel que la fonction σ_s est égale à 1 pour une "distance" à la singularité supérieure à $n_s\epsilon$. Il n'a pas été établi dans le cadre de cette thèse de méthode systématique pour déterminer ϵ et n_s. Ces valeurs ont été établies en simulation en observant l'évolution du conditionnement à l'approche de la singularité dans le cas du robot étudié. Afin de rendre la transition dérivable plusieurs fois, la fonction σ_s est combinée à une fonction polynomiale $\mathcal{Q}_s$ définie par :

$$\mathcal{Q}_s(r) = 35r^4 - 84r^5 + 70r^6 - 20r^7 \tag{E5.12}$$

La fonction $\mathcal{Q}_s$ est la fonction polynomiale de degré 7 prenant pour valeur 0 en 0 et 1 en 1 avec toutes les dérivées nulles en ces deux points. La figure 5.5 illustre l'évolution des fonctions que nous venons de définir.

Ainsi, la projection à l'approche de la singularité s'effectue en multipliant le vecteur $\mathbf{V}^T\boldsymbol{\nu}$

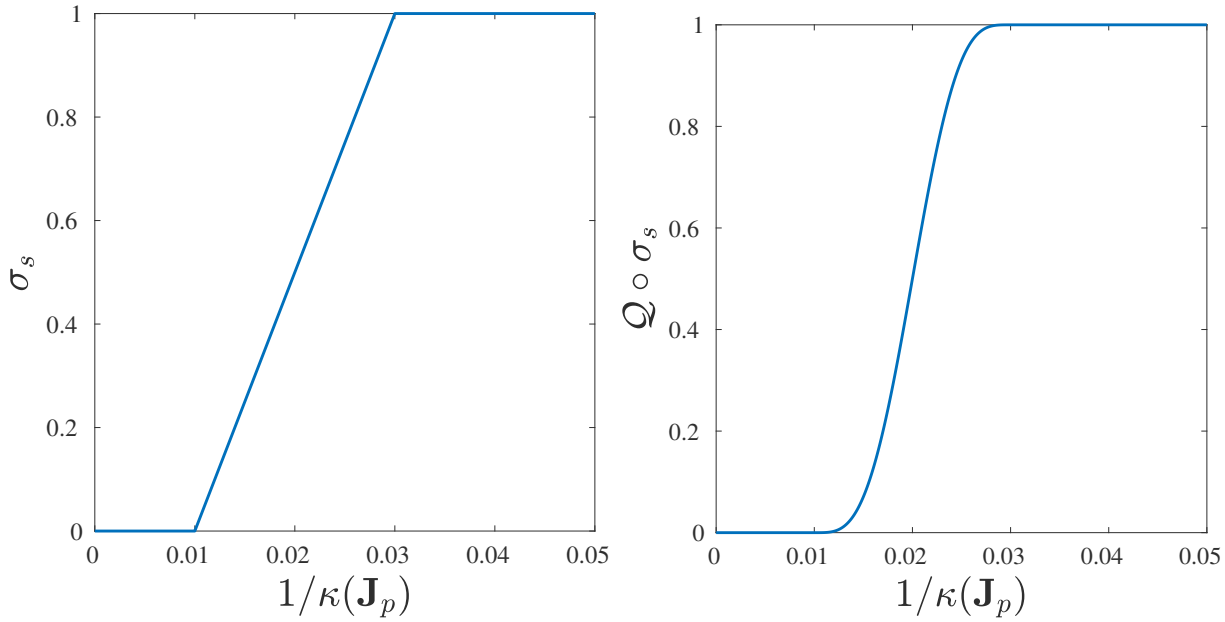


FIGURE 5.5 – Illustration des fonctions σ_s et $\mathcal{Q} \circ \sigma_s$ pour $\epsilon = 0.01$ et $n_s = 3$.

par la matrice diagonale :

$$\mathbf{Q} = \begin{bmatrix} 1 & 0 & \dots & 0 & 0 \\ 0 & 1 & \dots & 0 & 0 \\ \vdots & \vdots & \ddots & \vdots & \vdots \\ 0 & 0 & \dots & 1 & 0 \\ 0 & 0 & \dots & 0 & \mathcal{Q} \circ \sigma_s(\mathbf{J}_p) \end{bmatrix}$$

La matrice $\mathbf{Q}$ est égale à l'identité loin de la singularité ($\mathbf{Q}\mathbf{V}^T = \mathbf{V}^T$) et le dernier élément de la diagonale devient nul à l'approche de la singularité ($\mathbf{Q}\mathbf{V}^T = \mathbf{V}_0^T$). La solution de l'équation (E5.6), valable pour toutes les configurations, s'écrit alors :

$$\boldsymbol{\nu}_J = \mathbf{U}\mathbf{S}^T\mathbf{Q}\mathbf{V}^T\boldsymbol{\nu} \tag{E5.13}$$

Seule cette étape de calcul est modifiée dans la loi de commande établie dans la partie 2.4. Le reste de celui-ci n'est pas impacté par les singularités.

5.2.2 Cas de la commande sur les coordonnées cartésiennes des drones

Dans le cas où $n_p = 3n$, il y a autant de coordonnées de la structure passive $\mathbf{q}_p$ que de coordonnées de position des drones (points O_i aux extrémités des jambes de l'architecture passive), regroupées en un vecteur $\mathbf{q}_x$. C'est le cas des deux exemples de robots parallèles

volants développés au chapitre 3. La dérivée du vecteur $\mathbf{q}_x$ est reliée à la dérivée du vecteur $\mathbf{q}_p$ par l'équation :

$$\dot{\mathbf{q}}_x = \mathbf{J}_p \dot{\mathbf{q}}_p \tag{E5.14}$$

Dans ce cas, il est possible d'établir la loi de commande à partir des coordonnées de position des drones, de façon similaire à celle du robot volant à deux drones (voir partie 3.2.5). Le calcul du vecteur $\boldsymbol{\nu}$ dans la loi de commande est alors modifié par l'équation (E3.61) que nous rappelons ici :

$$\boldsymbol{\nu} = \mathbf{M}_t \mathbf{J}_p^{-1} (\boldsymbol{\nu}_x - \dot{\mathbf{J}}_p \mathbf{J}_p^{-1} \dot{\mathbf{q}}_x) + \mathbf{c}_t$$

Cette équation implique l'inversion de la matrice $\mathbf{J}_p$, impossible en singularité. Cette problématique provient de l'équation de relation entre l'accélération des coordonnées $\mathbf{q}_p$ et l'accélération des coordonnées $\mathbf{q}_x$. Cette équation est obtenue par la dérivée de l'équation (E5.14) :

$$\ddot{\mathbf{q}}_x = \mathbf{J}_p \ddot{\mathbf{q}}_p + \dot{\mathbf{J}}_p \dot{\mathbf{q}}_p \tag{E5.15}$$

Pour un système réel où nous mesurons l'accélération $\ddot{\mathbf{q}}_x$, comment retrouver l'accélération $\ddot{\mathbf{q}}_p$ en singularité à partir de l'équation (E5.15) ? Nous savons que l'équation admet au moins une solution (le robot réel possède physiquement une accélération). Cependant, en configuration singulière, le rang de $\mathbf{J}_p$ n'est pas plein. Il existe une infinité de solutions données par :

$$\ddot{\mathbf{q}}_p = \mathbf{J}_p^{+} (\ddot{\mathbf{q}}_x - \dot{\mathbf{J}}_p \dot{\mathbf{q}}_p) + (\mathcal{I}_n - \mathbf{J}_p^{+} \mathbf{J}_p) \boldsymbol{\beta} \tag{E5.16}$$

avec $\boldsymbol{\beta}$ un vecteur arbitraire de dimension n et $\mathcal{I}_n$ la matrice identité de dimension n. L'accélération $\ddot{\mathbf{q}}_x$ de drones ne suffit pas, en singularité, à connaître totalement l'accélération $\ddot{\mathbf{q}}_p$ de l'architecture passive. Il s'agit de la conséquence de la présence d'un mouvement incontrôlé en singularité. Il existe localement une dynamique libre de l'architecture passive. Cette dynamique libre a déjà été exprimée au travers du critère de traversée de la singularité (E5.5) :

$$\dot{\mathbf{q}}_{p,s}^{T} (\mathbf{M}_t \ddot{\mathbf{q}}_p + \mathbf{c}_t) = \mathbf{0}$$

En effet, il s'agit d'une équation sur les coordonnées d'accélération $\ddot{\mathbf{q}}_p$ indépendante des efforts fournis par les drones actionneurs. L'idée consiste alors à compléter le système d'équation (E5.15) avec le critère de traversée de la singularité (E5.5) pour obtenir la valeur de l'accélération $\ddot{\mathbf{q}}_p$. La preuve théorique de ce résultat, s'appuyant sur l'algèbre linéaire est fournie en annexe D. En pratique, le système complet à résoudre s'obtient en construisant l'équation cinématique de second ordre étendue, regroupant l'équation (E5.15) et (E5.5) :

$$\mathbf{J}_e \ddot{\mathbf{q}}_p = \mathbf{A}_e \ddot{\mathbf{q}}_x + \mathbf{b}_e \tag{E5.17}$$

avec

$$\mathbf{J}_e = \begin{bmatrix} \mathbf{J}_p \\ \dot{\mathbf{q}}_{p,s}^T \mathbf{M}_t \end{bmatrix} \quad \mathbf{A}_e = \begin{bmatrix} 1 & 0 & \dots & 0 \\ 0 & 1 & \dots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \dots & 1 \\ 0 & 0 & \dots & 0 \end{bmatrix} \quad \mathbf{b}_e = \begin{bmatrix} -\dot{\mathbf{J}}_p \dot{\mathbf{q}}_p \\ -\dot{\mathbf{q}}_{p,s}^T \mathbf{c}_t \end{bmatrix}$$

La solution de l'équation du second ordre est alors obtenue par l'expression :

$$\ddot{\mathbf{q}}_p = \mathbf{J}_e^+ (\mathbf{A}_e \ddot{\mathbf{q}}_x + \mathbf{b}_e) \tag{E5.18}$$

Cette expression n'est valable qu'en singularité. En dehors de celle-ci, la dynamique libre de la structure passive n'existe pas et $\mathbf{J}_p$ est de rang plein. L'équation (E5.15) se résout alors classiquement par :

$$\ddot{\mathbf{q}}_p = \mathbf{J}_p^{-1} (\ddot{\mathbf{q}}_x - \dot{\mathbf{J}}_p \dot{\mathbf{q}}_p) \tag{E5.19}$$

Notons que l'existence d'une solution donnée par (E5.18) n'est garantie que pour une accélération $\ddot{\mathbf{q}}_x$ *mesurée* sur le robot. En singularité, il est possible que certaines accélérations $\ddot{\mathbf{q}}_x$ ne soient pas réalisables par le robot. Or, l'inversion dynamique effectuée par la loi de commande est obtenue à partir du vecteur $\boldsymbol{\nu}_x$ représentant une accélération *désirée*. Si cette accélération n'est en pratique pas réalisable par le système, la solution de (E5.18) ne sera pas exacte mais celle la plus proche, au sens de la norme 2, admissible par le système.

Afin d'effectuer une transition continue entre le calcul en singularité (E5.18) et hors singularité (E5.19), nous utiliserons à nouveau les fonctions σ_s (E5.11) et $\mathcal{Q}$ (E5.12) définies précédemment. La solution complète de la cinématique du second ordre, s'écrit alors :

$$\ddot{\mathbf{q}}_x = \mathcal{Q} \circ \sigma_s (\mathbf{J}_p) \left(\mathbf{J}_p^{-1} (\ddot{\mathbf{q}}_x - \dot{\mathbf{J}}_p \dot{\mathbf{q}}_p) \right) + (1 - \mathcal{Q} \circ \sigma_s (\mathbf{J}_p)) \left(\mathbf{J}_e^+ (\mathbf{A}_e \ddot{\mathbf{q}}_x + \mathbf{b}_e \right) \tag{E5.20}$$

avec, pour rappel, $\mathcal{Q} \circ \sigma_s (\mathbf{J}_p) = 1$ loin de la singularité et $\mathcal{Q} \circ \sigma_s (\mathbf{J}_p) = 0$ proche de la singularité.

Notons que l'équation (E5.20) possède des termes fonction de la vitesse des coordonnées de la structure passive $\dot{\mathbf{q}}_p$ qui n'est a priori pas mesurée. Cette vitesse peut être calculée à partir de la vitesse des coordonnées $\dot{\mathbf{q}}_x$ grâce à l'équation (E5.14). Cependant en singularité, ce calcul n'est plus possible. Plusieurs méthodes peuvent être envisagées afin d'obtenir cette vitesse sans ajout des capteurs supplémentaires :

- Utiliser localement une estimation de la vitesse par intégration de celle de l'accélération.
- Considérer l'erreur faible et utiliser la vitesse de la trajectoire désirée. Cette méthode est déjà appliquée dans le cas d'une commande en couples calculés lorsque les mesures de vitesse sont très bruitées [Khalil et Dombre, 2002]. Nous utiliserons cette seconde méthode dans le cadre des simulations mises en œuvre.

Les modifications décrites dans cette partie ont été intégrées à la loi de commande développée précédemment dans le cas du robot à deux drones (voir partie (3.2.5)). Cette modi-

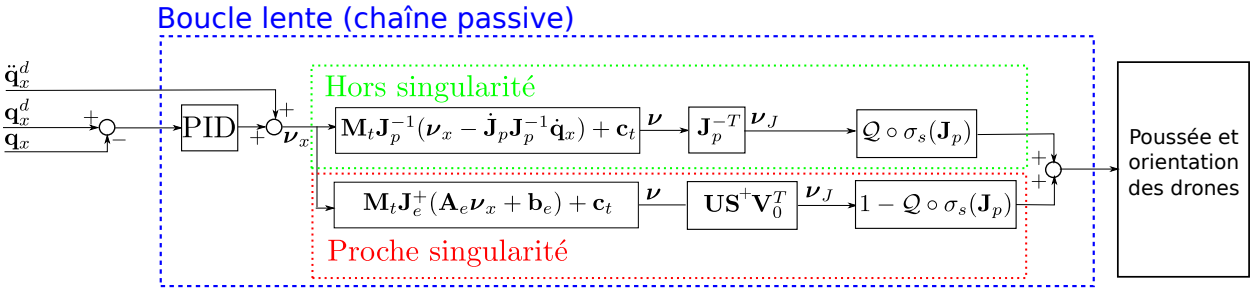


FIGURE 5.6 – Schéma de la boucle lente (architecture passive) adaptée à la traversée des singularités.

fication est illustrée Fig. 5.6. La partie suivante présente la trajectoire suivie et les résultats obtenus en simulation de la traversée d'une singularité par le robot parallèle volant à deux drones.

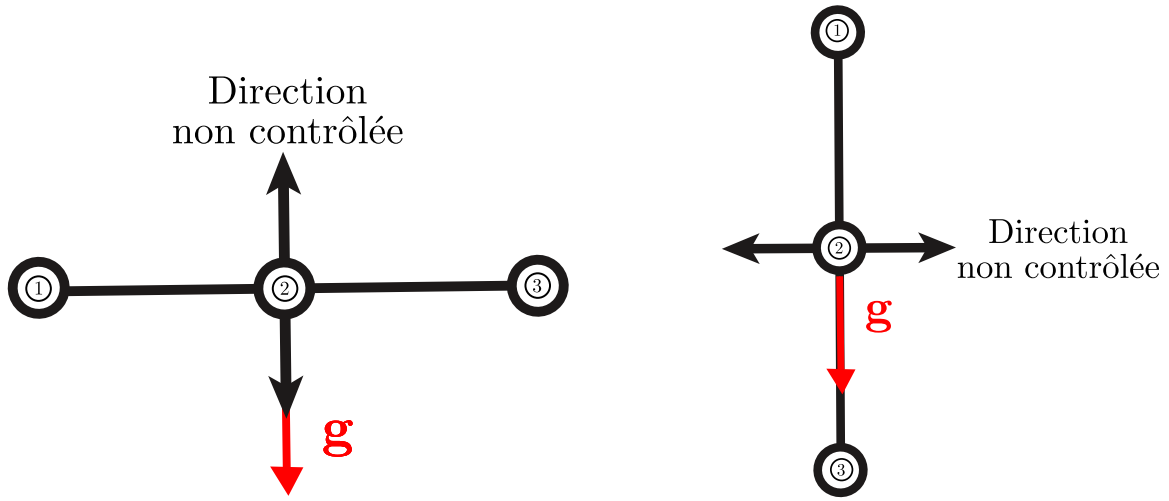
5.3 Simulation de traversée d'une singularité

La simulation pour cette partie est effectuée dans les mêmes conditions que celle réalisée dans la partie 3.2.7. Les paramètres physiques du robot, les gains et filtres sont totalement identiques. Le bruit simulé sur les mesures également.

5.3.1 Trajectoire de traversée

La trajectoire de traversée de la singularité est initialement définie pour les coordonnées de l'architecture passive $\mathbf{q}_p$. En effet, le critère de traversée de la singularité (E5.5) est établi pour ces coordonnées. La création de la trajectoire consiste à définir deux polynômes de degré 9 par coordonnée (donc huit polynômes en tout). Ces polynômes servent à définir deux moitiés de trajectoire : l'une partant d'une configuration initiale (en équilibre statique) à la singularité et l'autre de la singularité vers une configuration finale (en équilibre statique également). La trajectoire est ainsi définie grâce aux coordonnées des configurations initiales, finales et aux coordonnées de la configuration de singularité, ainsi qu'aux deux premières dérivées temporelles des coordonnées de l'architecture passive en singularité. Les troisièmes et quatrièmes dérivées temporelles sont choisies nulles pour les trois points. La durée de trajectoire de traversée est fixée à 4 secondes avec une traversée au bout de 2 secondes.

Un algorithme d'optimisation, cherchant à minimiser la norme de forces de poussées et l'accélération des angles sur les drones lors de l'exécution de la trajectoire, a été appliqué sous contrainte du critère de traversée de la singularité. Afin de diminuer le temps de calcul de la boucle d'optimisation, le choix arbitraire de garder fixe la position de l'articulation ① a été effectué. Le minimum local obtenu est donné dans le tableau (T5.1). Le résultat de l'optimisation indique que la position la moins "difficile" pour traverser la singularité est telle



(a) Cas 1 : La gravité est colinéaire avec la direction non contrôlée. Engendrant une accélération localement incontrôlable de l'articulation ②.

(b) Cas 2 : La gravité est orthogonale à la direction non contrôlée. La traversée de la singularité peut s'effectuer à faible accélération.

FIGURE 5.7 – Comparaison des configurations de traversée d'une singularité du robot parallèle volant à deux drones.

	$\mathbf{q}_p$	$\dot{\mathbf{q}}_p$	$\ddot{\mathbf{q}}_p$
Début	$[0 \quad 0 \quad \mathbf{1.57} \quad \mathbf{-0.3}]$	$[0 \quad 0 \quad 0 \quad 0]$	$[0 \quad 0 \quad 0 \quad 0]$
Singularité	$[0 \quad 0 \quad \mathbf{1.57} \quad 0]$	$[0 \quad 0 \quad 0 \quad \mathbf{0.26}]$	$[0 \quad 0 \quad \mathbf{-0.016} \quad \mathbf{-0.028}]$
Fin	$[0 \quad 0 \quad \mathbf{1.54} \quad \mathbf{0.3}]$	$[0 \quad 0 \quad 0 \quad 0]$	$[0 \quad 0 \quad 0 \quad 0]$

TABLE 5.1 – Résultat de l'algorithme d'optimisation pour la traversée de la singularité. Les paramètres en gras sont les paramètres optimisés.

que les deux barres soient en position verticale ($\varphi = \pi$). En effet, dans cette configuration, le vecteur de gravité est perpendiculaire à la direction non contrôlée, évitant une accélération non maîtrisée de l'articulation ② lors de la traversée de la singularité (voir Fig. 5.7). A partir de ce résultat, une trajectoire complète pour la simulation est établie. Elle comprend :

- une trajectoire polynomiale pour rejoindre la configuration de début de la trajectoire de traversée de la singularité,
- la trajectoire de traversée de la singularité, composée de deux polynômes par coordonnées et définie ci-avant,
- une trajectoire polynomiale de retour vers le point d'origine de la simulation.

Au total, quatre polynômes de degré 9 par coordonnée définissent la trajectoire. Le tableau (T5.2) donne les coefficients de ces polynômes pour les quatre étapes de la trajectoire. La Fig. 5.8 montre l'évolution des coordonnées $\mathbf{q}_p$ pendant la trajectoire. La visualisation de la trajectoire dans le plan vertical est illustrée Fig. 5.9.

La trajectoire est convertie dans l'espace des coordonnées $\mathbf{q}_x$ qui, dans le cas particulier

	a_9	a_8	a_7	a_6	a_5	a_4	a_3	a_2	a_1	a_0
y (toutes étapes)	0.0e+00	0.0e+00	0.0e+00	0.0e+00	0.0e+00	0.0e+00	0.0e+00	0.0e+00	0.0e+00	0.0e+00
z (toutes étapes)	0.0e+00	0.0e+00	0.0e+00	0.0e+00	0.0e+00	0.0e+00	0.0e+00	0.0e+00	0.0e+00	0.0e+00
q_1 (étape 1)	8.9e-11	-8.0e-09	2.7e-07	-4.3e-06	2.6e-05	0.0e+00	0.0e+00	0.0e+00	0.0e+00	9.2e-01
q_1 (étape 2)	-1.2e-03	1.1e-02	-3.5e-02	5.2e-02	-2.9e-02	0.0e+00	0.0e+00	0.0e+00	0.0e+00	1.6e+00
q_1 (étape 3)	-4.4e-03	3.9e-02	-1.3e-01	2.0e-01	-1.2e-01	0.0e+00	0.0e+00	-8.3e-03	0.0e+00	1.6e+00
q_1 (étape 4)	-3.3e-10	3.0e-08	-1.0e-06	1.6e-05	-9.6e-05	0.0e+00	0.0e+00	0.0e+00	0.0e+00	1.5e+00
q_2 (étape 1)	1.8e-10	-1.6e-08	5.7e-07	-8.8e-06	5.3e-05	0.0e+00	0.0e+00	0.0e+00	0.0e+00	-1.8e+00
q_2 (étape 2)	6.4e-03	-6.7e-02	2.7e-01	-5.1e-01	3.9e-01	0.0e+00	0.0e+00	0.0e+00	0.0e+00	-5.0e-01
q_2 (étape 3)	1.2e-02	-9.7e-02	3.0e-01	-3.9e-01	1.7e-01	0.0e+00	0.0e+00	-2.3e-02	4.3e-01	0.0e+00
q_2 (étape 4)	1.8e-10	-1.6e-08	5.7e-07	-8.8e-06	5.3e-05	0.0e+00	0.0e+00	0.0e+00	0.0e+00	5.0e-01

TABLE 5.2 – Expressions polynomiales définissant la trajectoire de traversée de la singularité.
$\mathcal{Q}(t) = a_9t^9 + a_8t^8 + a_7t^7 + a_6t^6 + a_5t^5 + a_4t^4 + a_3t^3 + a_2t^2 + a_1t + a_0$ avec t le temps écoulé en secondes depuis le début de la portion de trajectoire concernée.

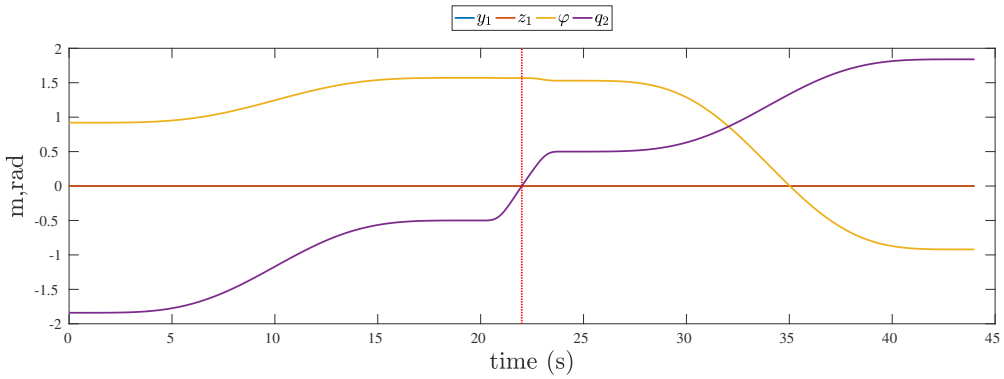


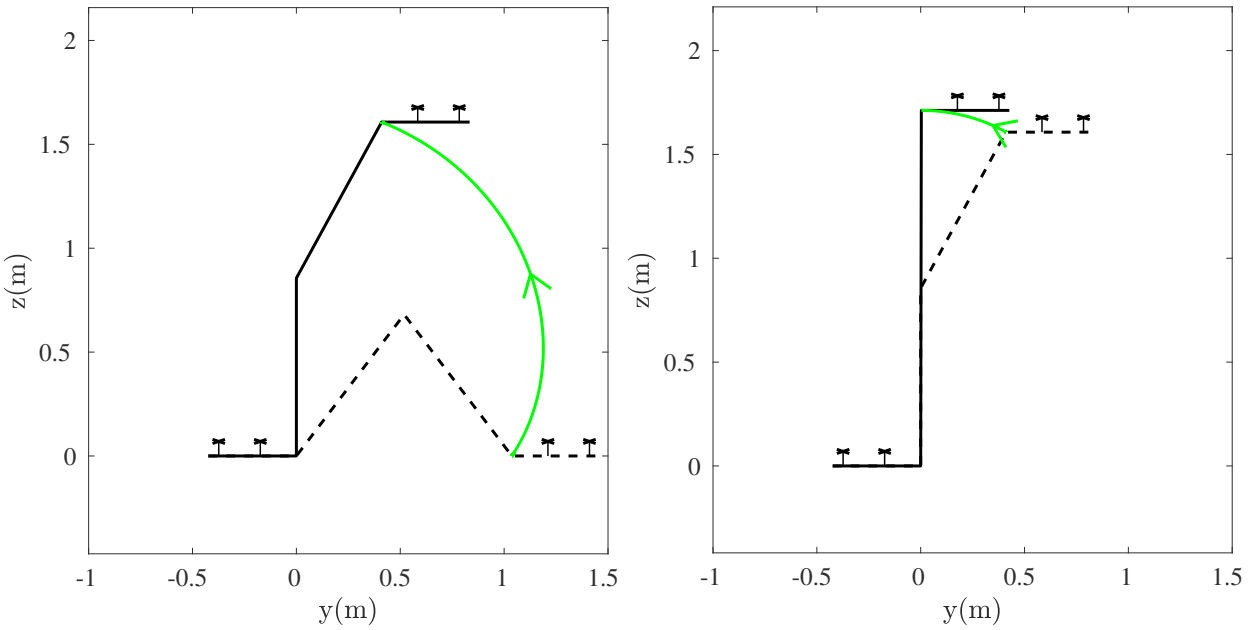
FIGURE 5.8 – Trajectoire pour la traversée de la singularité pour les coordonnées $\mathbf{q}_p$. La barre verticale rouge indique la traversée de la singularité.

du robot volant à deux drones, sont les coordonnées de position des articulations ① et ③ (Fig. 3.19) du robot dans le plan vertical. Cette conversion se fait grâce aux modèles géométriques et cinématiques établis dans la partie 3.2. Le suivi de la trajectoire sera assuré pour les coordonnées $\mathbf{q}_x$, afin de se rapprocher au mieux des conditions de l'expérimentation pour le robot réel. La figure 5.10 montre les forces de poussée et les orientations des quadricoptères attendues lors de l'exécution de la trajectoire.

5.3.2 Résultats de simulation

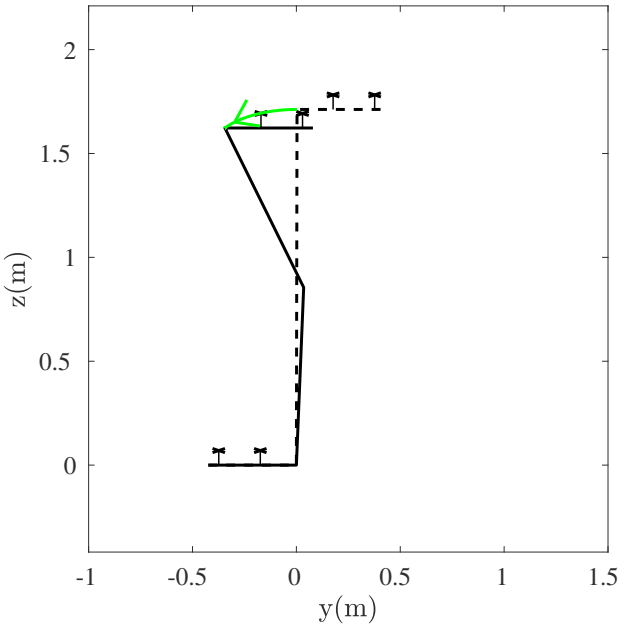
La durée totale de la trajectoire est de 44 secondes. Seule une partie de la trajectoire apporte de nouveaux éléments au regard des simulations déjà présentées dans la partie 3.2.7. Nous allons donc présenter les résultats sur les 12 secondes autour de la singularité afin d'étudier l'impact de la traversée. Les résultats présentés sont affichés filtrés pour une meilleure lecture des courbes.

Les figures 5.11 et 5.12 affichent respectivement les erreurs de suivi pour les coordonnées $\mathbf{q}_x$ et $\mathbf{q}_a$. Nous constatons une légère dégradation du suivi de l'angle φ_2 aux alentours de la singularité. Ceci est probablement dû à la variation rapide des angles désirés (voir également Fig. 5.13). Nous constatons également une légère augmentation locale de l'erreur de suivi des positions lors de la traversée de la singularité. Ce résultat est attendu car, d'une part, le suivi des orientations des drones est moins précis et affecte directement la précision de la boucle lente et, d'autre part, l'erreur dans la direction du mouvement non-contrôlé en singularité ne peut pas être corrigée. Cependant, la déviation reste faible et la commande récupère l'erreur après la traversée. La figure 5.14 affiche les forces de poussée des drones pendant la traversée de la singularité. Il n'y a pas de divergence des forces demandées lors de la traversée ce qui confirme l'efficacité des précautions prises dans la loi de commande. Une vidéo de la traversée de la singularité est disponible au lien https://uncloud.univ-nantes.fr/index.php/s/ZCxyMEoygM4kaZq.

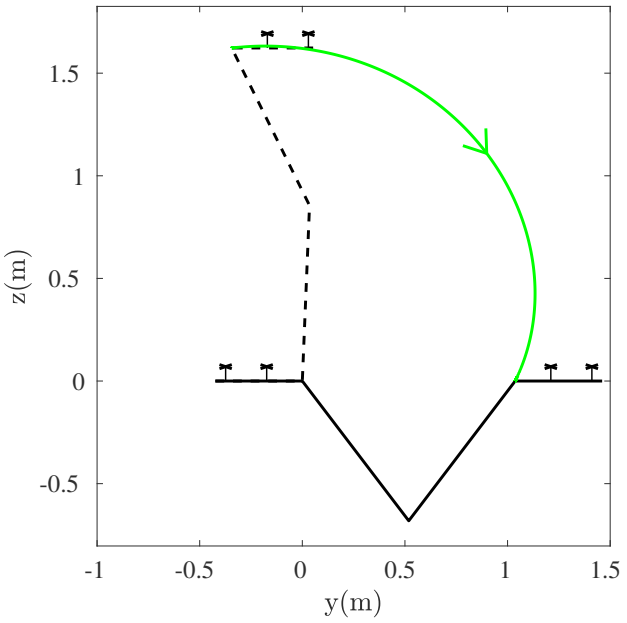


(a) Trajectoire d'approche

(b) Traversée de singularité (1)



(c) Traversée de singularité (2)



(d) Trajectoire de retour

FIGURE 5.9 – Trajectoire de traversée de la singularité pour le robot parallèle volant à deux drones.

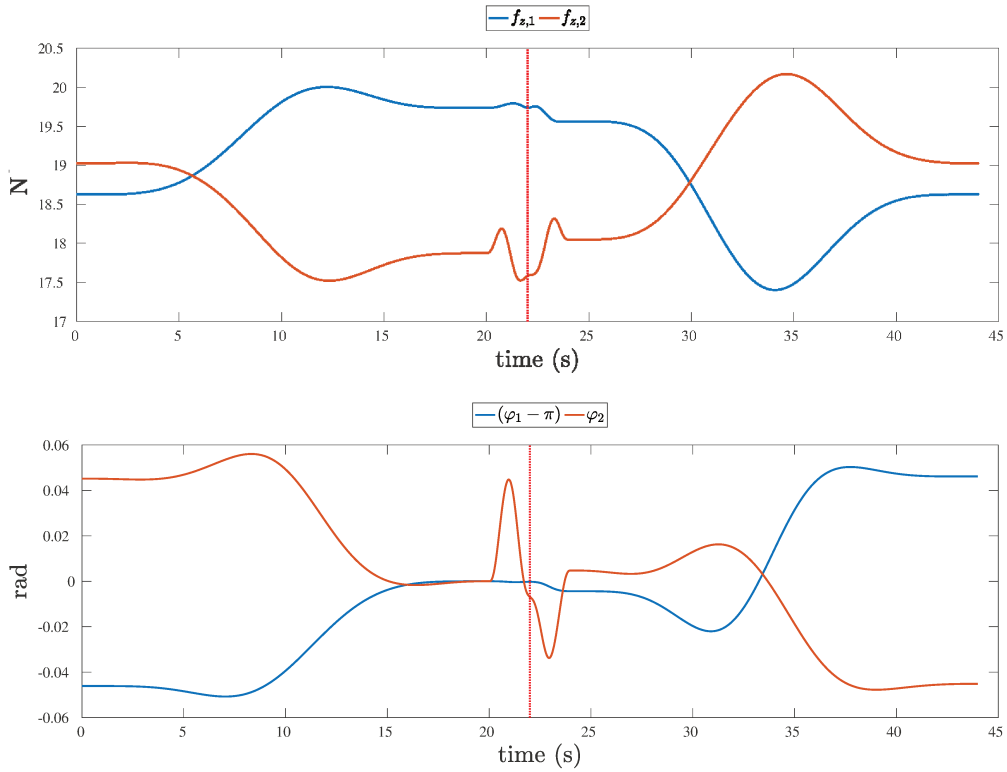


FIGURE 5.10 – Forces de poussée et orientations des drones attendues lors de l'exécution de la trajectoire. La barre vertical rouge indique la traversée de la singularité.

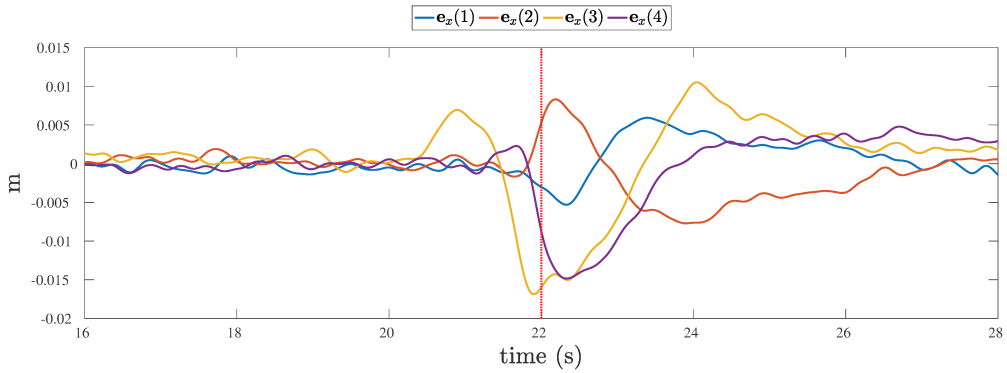


FIGURE 5.11 – Simulation de traversée de singularité : erreur de suivi des positions $\mathbf{e}_x$.

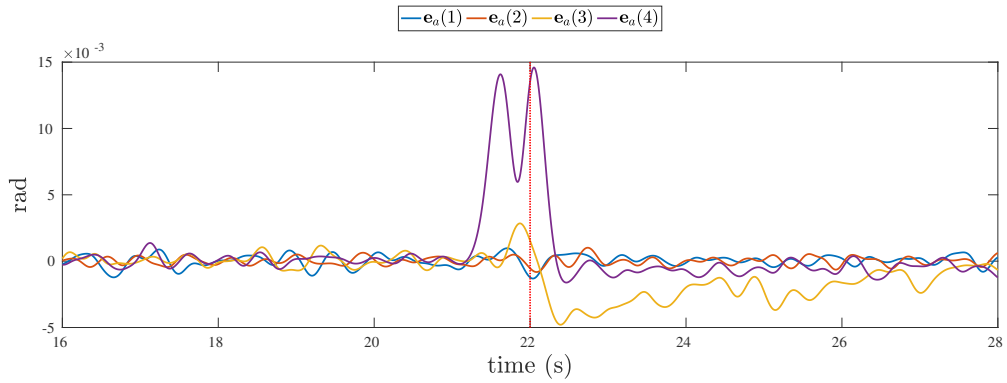


FIGURE 5.12 – Simulation de traversée de singularité : erreur de suivi des orientations $\mathbf{e}_a$.

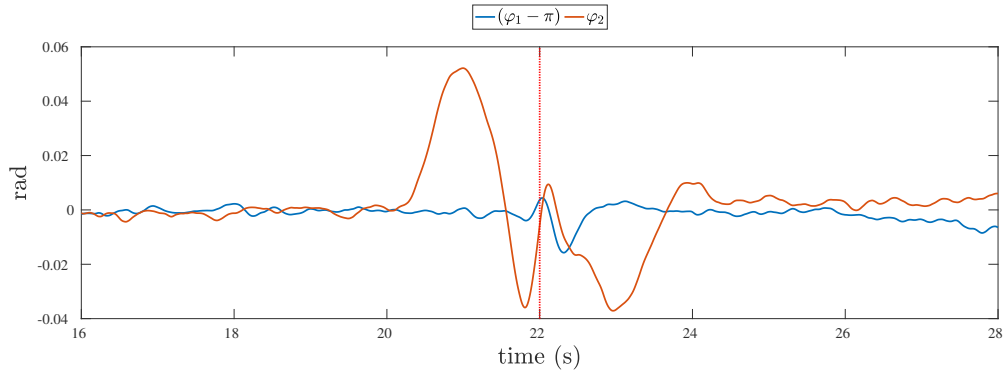


FIGURE 5.13 – Simulation de traversée de singularité : angles d'orientation calculés par la commande.

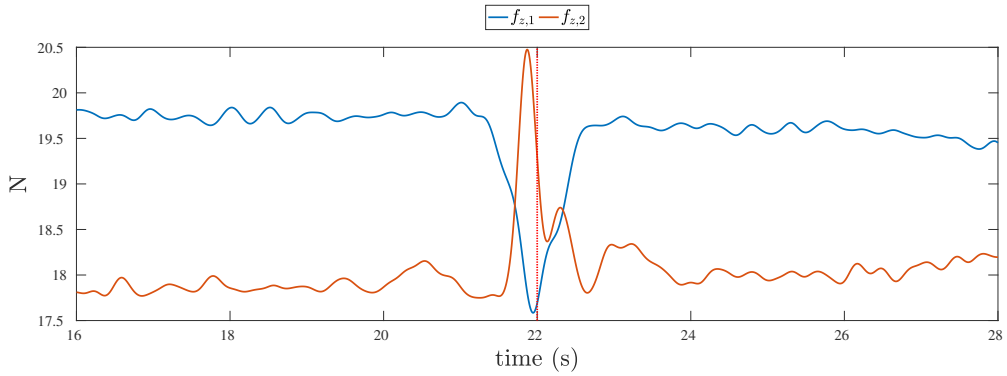


FIGURE 5.14 – Simulation de traversée de singularité : forces de poussée calculées par la commande.

5.4 Conclusion

Dans cette partie, nous avons étudié les singularités du modèle dynamique des robots parallèles volants. Il a été montré qu'en singularité une contrainte peut être établie sur la dynamique des coordonnées de l'architecture passive. Sous respect de cette contrainte, nous avons établi le modèle dynamique inverse du robot en singularité. Ces éléments ont permis la conception d'une loi de commande dédiée à la traversée d'une singularité du robot parallèle volant, à l'aide d'une trajectoire respectant la contrainte dynamique. La traversée de singularité a été testée avec succès en simulation à l'aide du logiciel ADAMS.

Conclusion

Synthèse et contributions de la thèse

Le sujet de cette thèse est la conception et la commande d'un nouveau type de manipulateur aérien. Les manipulateurs aériens existants font face, de manière générale, aux difficultés suivantes :

- une faible autonomie due à la charge embarquée et à la consommation énergétique du manipulateur,
- le maintien de la stabilité et de la précision du vol lors de la manipulation,
- une perturbation de la manipulation par le souffle du drone,
- une capacité de manipulation limitée observée sur les solutions de coopération existantes.

Afin de dépasser certaines de ces limites, un nouveau concept de robot aérien est proposé. Il consiste en une architecture passive poly-articulée composée d'une plate-forme sur laquelle sont fixées un nombre arbitraire de jambes. Un drone est attaché à chacune de ces jambes, à l'extrémité opposée à la plate-forme, par une articulation sphérique passive, située au niveau du centre de gravité du drone. Le mouvement de ces drones, en particulier des quadricoptères dans notre étude, actionne l'architecture passive. Le chapitre 2 présente la cinématique de ce robot et développe son modèle dynamique, faisant apparaître un découplage entre la dynamique de rotation des drones et la dynamique de l'architecture passive. Ce découplage a permis la conception d'une commande en cascade, assurant la stabilité du robot en vol et permettant le suivi d'une trajectoire.

À partir du concept générique, deux robots ont été étudiés au chapitre 3. Une étude cinématique spécifique à chaque robot est nécessaire afin de déterminer la mobilité de la plate-forme et les configurations singulières. Elle a été effectuée sur un premier robot parallèle volant à trois drones. Nous avons ensuite mis en œuvre une simulation afin de valider le fonctionnement de la commande lors de la stabilisation et du suivi de trajectoire. Dans le chapitre 3 nous avons également étudié un second robot à deux drones. Ce robot, dérivé d'un mécanisme parallèle plan, a été conçu dans le but de construire un prototype expérimental. Celui-ci apparaissait, a priori, comme un modèle simplifié, mais, du fait de sa conception, le modèle dynamique et la commande associée diffèrent du modèle générique développé au chapitre 2. Nous avons donc effectué des modifications de la loi de commande, afin de

maintenir le robot dans un plan vertical correspondant au plan de travail du mécanisme initial. Une simulation a permis de valider la loi de commande appliquée à ce robot.

Le prototype expérimental du robot parallèle volant à deux drones a été construit au laboratoire LS2N à partir de deux drones PELICANs disponibles, tel que présenté au chapitre 4. Nous l'avons associé à un système de capture de mouvements pour suivre la position du robot lors de l'expérimentation. Le prototype a permis de prouver la stabilité de la commande conçue en conditions réelles. Les performances en terme de précision restent cependant limitées. En effet, le maintien dans un plan vertical du robot sollicite beaucoup l'actionnement ce qui perturbe le fonctionnement de la commande.

La reconfiguration en vol de l'architecture passive est une propriété importante de ces systèmes. Elle permet, par exemple, d'effectuer un décollage dans de bonnes conditions, en plaçant la plate-forme au-dessus du plan des drones, puis de changer de configuration pour effectuer une tâche nécessitant de positionner la plate-forme sous le plan des drones. Cette reconfiguration de l'architecture passive s'effectue par la traversée d'une configuration singulière. Lors de la traversée d'une singularité, l'inversion du modèle dynamique dans la commande peut mener à une divergence des entrées pour une trajectoire arbitraire. Le chapitre 5 présente un critère dynamique à respecter par la trajectoire afin de traverser cette singularité. La commande conçue précédemment a été améliorée afin de pouvoir gérer le vol à proximité des configurations singulières. Cette capacité de la loi de commande à suivre la trajectoire de traversée de la singularité a été confirmée en simulation.

Perspectives

Cette thèse a établi une preuve de concept des robots parallèles volants. Le prototype mis en œuvre, dans les limites de durée de cette étude, ne permet cependant pas d'avoir une notion complète des performances atteignables en terme de précision. Une première voie d'amélioration serait d'effectuer une optimisation de la conception du robot parallèle volant à trois drones, basée sur un objectif de manipulation. Ensuite, la réalisation du prototype correspondant permettra d'évaluer les performances réelles de ce robot.

En simulation, nous supposons la configuration de l'architecture passive mesurée à la fréquence du contrôleur (1 kHz). En expérimentation, cette mesure, effectuée par le système de capture de mouvement, est obtenue à une fréquence de 250 Hz. Un filtre Kalman, tel que présenté en annexe B, permet d'estimer la mesure à 1kHz en fusionnant ces données avec les mesures issues des centrales inertielles des drones. Dans le cas du prototype expérimental à deux drones, l'estimation est effectuée sur la position des quadricoptères et la configuration de l'architecture passive est obtenue au travers du modèle géométrique direct du robot. Ce modèle géométrique dans le cas du robot parallèle volant à trois drones, équivalent à un système $3-SR$, est complexe et peut admettre jusqu'à 16 solutions [Merlet, 2006]. Cependant, il sera possible de s'en abstenir en concevant un filtre Kalman spécifique, fusionnant directement la mesure de la configuration de l'architecture, obtenue par la capture de mouvement, avec les données fournies par les centrales inertielles des trois drones.

Nous n'avons validé la reconfiguration de l'architecture passive par la traversée d'une singularité qu'en simulation. L'expérimentation sur le prototype existant s'avère délicate du fait du manque de précision du suivi de trajectoire. La mise en œuvre sur un prototype plus précis pourra permettre le test de ces reconfigurations dans le cadre expérimental.

Une autre étape importante serait la conception d'un contrôleur tenant compte de l'interaction avec l'environnement, au travers d'une mesure des efforts sur l'effecteur ou par la conception d'un observateur *ad hoc*. Le modèle dynamique du robot parallèle volant possédant une forme similaire au modèle du quadricoptère, il sera possible de s'inspirer des études existantes traitant de l'interaction avec l'environnement, telles que [Nguyen et Lee, 2013; Bellens et al., 2012; Darivianakis et al., 2014; Papachristos et al., 2014]. À l'aide de ce contrôleur, des tâches de manipulation en conditions expérimentales pourront être envisagées. Elles seront effectuées par :

- une interaction directe de la plate-forme avec l'environnement,
- l'ajout d'un outil spécifique sur la plate-forme, actionné indépendamment,
- l'ajout d'un outil lié à la configuration relative des jambes et de la plate-forme. En effet, dans le cas du robot volant à trois drones, nous disposons du contrôle des six degrés de liberté de la plate-forme mais également de trois degrés de liberté liés à la configuration des jambes. Ces degrés de liberté supplémentaires peuvent être utilisés pour répondre à des contraintes spatiales de l'environnement dans lequel le robot évolue mais aussi pour contribuer à l'actionnement d'un mécanisme.

D'autre part, les expérimentations effectuées lors de cette thèse ont toutes été réalisées en intérieur, avec une mesure de la position des drones par capteurs extéroceptifs. Dans le cadre d'applications industrielles, les conditions de vol ne sont pas aussi idéales. Ainsi, il conviendra de transposer au robot parallèle volant les connaissances en termes de vol en extérieur et de positionnement à l'aide de capteurs embarqués uniquement. Pour cela, il sera possible de s'inspirer des travaux réalisés dans le cadre de vols en formation à l'extérieur tels que ceux présentés dans [Vásárhelyi et al., 2014; Vetrella et al., 2015].

Lors de cette thèse, nous avons seulement étudié deux architectures alors que le développement d'autres robots parallèles volants pourrait mener à la découverte de propriétés d'intérêt dans diverses applications. Par exemple, d'autres chaînes cinématiques des jambes sont envisageables et il est également possible d'augmenter le nombre de drones afin de disposer de redondances. Cette dernière option permet de sécuriser le dispositif en cas de défaut de l'un des drones dans l'optique d'applications industrielles.

Enfin, dans le cadre de cette thèse, les robots parallèles volants sont soumis à une contrainte dans leur conception. La liaison entre le quadricoptère et l'architecture passive est une articulation sphérique située au centre de gravité du drone. Une telle conception peut être techniquement difficile à respecter. Dans ce cas, le découplage du modèle dynamique, servant de base au développement du contrôleur en cascade, n'est pas obtenu. Lors de l'étude du robot parallèle à deux drones, nous avons montré que le contrôleur établi arrivait néanmoins à stabiliser le vol. L'élaboration d'un contrôleur, permettant de s'affranchir de ces contraintes de conception, est également une piste de développement de ce concept. Une idée, à terme, serait de s'affranchir totalement des drones existants et de définir un cadre pour des structures volantes, composées de corps rigides reliés entre eux par des articulations passives, chaque corps pouvant être associé à un ou plusieurs rotors.

Publications

Conférences internationales

D. Six, S. Briot, A. Chriette, et P. Martinet. A controller for avoiding dynamic model degeneracy of parallel robots during type 2 singularity crossing. In *New Trends in Mechanism and Machine Science (EUCOMES proceedings)*, pages 589–597, Nantes, France, sep 2016

R.B. Hill, D. Six, A. Chriette, S. Briot, et P. Martinet. Crossing type 2 singularities of parallel robots without pre-planned trajectory with a virtual-constraint-based controller. In *Proceedings of the International Conference on Robotics and Automation (ICRA)*, pages 6080–6085, Singapore, may 2017

D. Six, A. Chriette, S. Briot, et P. Martinet. Dynamic modeling and trajectory tracking controller of a novel flying parallel robot. In *Proceedings of the IFAC World Congress*, pages 2241–2246, Toulouse, France, jul 2017b

Journaux

D. Six, S. Briot, A. Chriette, et P. Martinet. A controller avoiding dynamic model degeneracy of parallel robots during singularity crossing. *Journal of Mechanisms and Robotics*, 9(5) :051008–051008–8, 2017a

D. Six, S. Briot, A. Chriette, et P. Martinet. The kinematics , dynamics and control of a flying parallel robot with three quadrotors. *IEEE Robotics and Automation Letters*, 3(1) : 559–566, 2018

Publication hors thèse

C. Chevallereau, H. Razavi, D. Six, Y. Aoustin, et J. Grizzle. Self-synchronization and self-stabilization of 3D bipedal walking gaits. *Robotics and Autonomous Systems*, 100 :43–60, 2018

Annexe A

Calcul des modèles cinématiques et dynamiques d'un robot défini à l'aide de la convention DHKK

Cette annexe présente une méthode de calcul des matrices jacobiennes et du modèle dynamique pour une architecture aborescente défini à l'aide de la convention Denhavit-Hartenberg modifiée. En pratique, dans le cadre de cette thèse, le logiciel SYMORO+ [Khalil et Creusot, 1997] a été utilisé pour fournir un calcul numérique optimisé de ces modèles.

A.1 Calcul de la matrice jacobienne

Chaîne cinématique ouverte

On considère dans un premier temps une simple chaîne cinématique ouverte allant du corps $\mathcal{B}_0$ fixé au sol au corps $\mathcal{B}_{n_c}$ représentant l'effecteur. En convention DHM, chaque corps est relié à son antécédent, soit par une articulation prismatique, soit par une articulation pivot. Les liaisons plus complexes (sphériques, cardant) sont représentées comme un enchaînement de ces articulations. La convention DHM définit des repères $\mathcal{F}_0$ à $\mathcal{F}_{n_c}$ associés à chaque corps [Khalil et Kleinfinger, 1986]. L'origine du repère $\mathcal{F}_j$ est appelée R_j. L'axe $\mathbf{z}_j$ du repère $\mathcal{F}_j$ est aligné selon l'axe de l'articulation. La figure A.1 montre, à titre d'exemple, les repères définis selon la convention DHM pour le robot parallèle volant à deux drones.

La matrice jacobienne exprime la relation entre le torseur cinématique de l'effecteur $\mathbf{t}_{n_c}$ au point R_{n_c} et le vecteur des vitesse des coordonnées de la chaîne cinématique $\dot{\mathbf{q}}_c = [\dot{q}_1 \ \dots \dot{q}_{n_c}]^T$:

$$\mathbf{t}_{n_c} = \mathbf{J}_{c,n_c}\dot{\mathbf{q}}_c \tag{E1.1}$$

Pour calculer cette matrice jacobienne nous allons calculer indépendamment la contribution de chaque articulation.

- Si l'articulation j est prismatique, le torseur cinématique produit par cette articulation

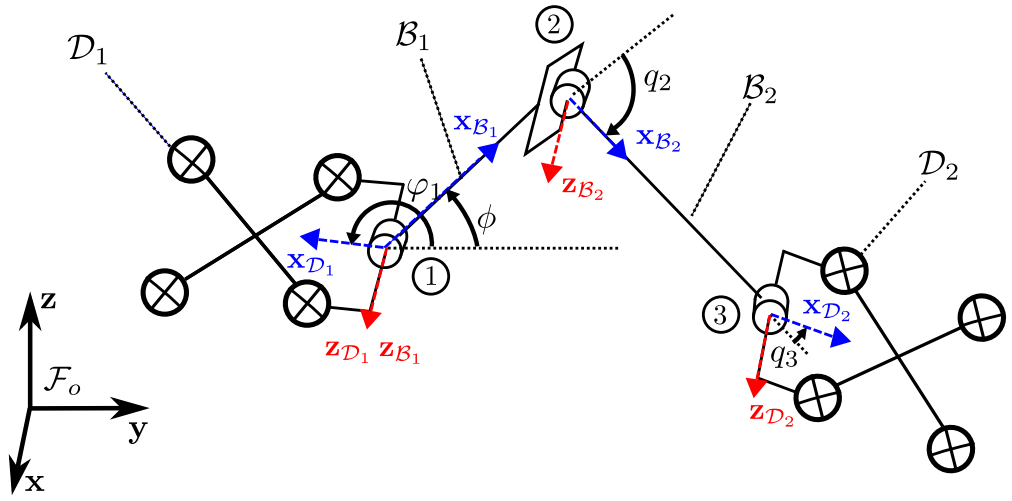


FIGURE A.1 – Repères définis selon la convention DHM pour le robot parallèle volant à deux drones. Les repères correspondent aux paramètres donnée en table (T3.3) (les repères virtuels ne sont pas représentés).

au point R_{n_c} s'écrit :

$$\mathbf{t}_{j,n_c} = \begin{bmatrix} \mathbf{z}_j \\ \mathbf{0} \end{bmatrix} \dot{q}_j \tag{E1.2}$$

- Si l'articulation j est un pivot, le torseur cinématique produit par cette articulation au point R_{n_c} s'écrit :

$$\mathbf{t}_{j,n_c} = \begin{bmatrix} \mathbf{z}_j \times \mathbf{r}_{R_j R_{n_c}} \\ \mathbf{z}_j \end{bmatrix} \dot{q}_j \tag{E1.3}$$

avec $\mathbf{r}_{R_j R_{n_c}}$ le vecteur reliant le point R_j au point R_{n_c}.

La matrice jacobienne s'obtient alors par la somme de la contribution de chaque articulation :

$$\mathbf{t}_{n_c} = \sum_{j=1}^{n_c} \mathbf{t}_{j,n_c} = \sum_{j=1}^{n_c} \begin{bmatrix} \sigma_j \mathbf{z}_j + \bar{\sigma}_j (\mathbf{z}_j \times \mathbf{r}_{R_j R_{n_c}}) \\ \bar{\sigma}_j \mathbf{z}_j \end{bmatrix} \dot{q}_j \tag{E1.4}$$

avec $\sigma_j = 0$ (respectivement 1) et $\bar{\sigma}_j = 1$ (resp. 0) si l'articulation j est un pivot (resp. une glissière). En identifiant (E1.1) et (E1.4) sous une forme matricielle, on obtient la matrice jacobienne de la chaîne cinématique :

$$\mathbf{J}_{c,n_c} = \begin{bmatrix} \sigma_1 \mathbf{z}_1 + \bar{\sigma}_1 (\mathbf{z}_1 \times \mathbf{r}_{R_1 R_{n_c}}) & \dots & \sigma_{n_c} \mathbf{z}_{n_c} + \bar{\sigma}_{n_c} (\mathbf{z}_{n_c} \times \mathbf{r}_{R_{n_c} R_{n_c}}) \\ \bar{\sigma}_1 \mathbf{z}_1 & \dots & \bar{\sigma}_{n_c} \mathbf{z}_{n_c} \end{bmatrix} \tag{E1.5}$$

Le torseur cinématique $\mathbf{t}_{n_c}$ et les vecteurs $\mathbf{z}_j$ et $\mathbf{r}_{R_j R_{n_c}}$ doivent être exprimés dans le même repère. L'expression finale de la matrice $\mathbf{J}_{c,n_c}$ dépend du repère choisi. Le calcul des vecteurs $\mathbf{z}_j$ et $\mathbf{r}_{R_j R_{n_c}}$ dans ce repère nécessitent le calcul de matrices de transition entre les repères définis par la convention DHM. Le détail de ces calculs sort du cadre de cette thèse, il peut

être trouvé notamment dans [Khalil et Dombre, 2002].

Par extension des équations (E1.1) et (E1.5), il est également possible d'exprimer le torseur cinématique du point R_k pour un corps $\mathcal{B}_k$:

$$\mathbf{t}_k = \mathbf{J}_{c,k}\dot{\mathbf{q}}_c \tag{E1.6}$$

avec

$$\mathbf{J}_{c,k} = \begin{bmatrix} \sigma_1\mathbf{z}_1 + \bar{\sigma}_1(\mathbf{z}_1 \times \mathbf{r}_{R_1R_k}) & \dots & \sigma_k\mathbf{z}_k + \bar{\sigma}_k(\mathbf{z}_k \times \mathbf{r}_{R_kR_k}) & \mathbf{0} & \dots & \mathbf{0} \\ \bar{\sigma}_1\mathbf{z}_1 & \dots & \bar{\sigma}_k\mathbf{z}_k & \mathbf{0} & \dots & \mathbf{0} \end{bmatrix}$$

Structure arborescente

Dans le cadre de cette thèse, la matrice jacobienne est exprimée pour chaque jambe de l'architecture arborescente passive du robot (E2.3). Pour effectuer ce calcul, on isole une chaîne cinématique ouverte allant du repère de référence $\mathcal{F}_o$ au corps situé à l'extrémité de la jambe. Par exemple, pour le robot volant à trois drones décrit en table (T3.1), on considère trois chaînes cinématiques composées respectivement des corps :

- $\mathcal{V}_1$, $\mathcal{V}_2$, $\mathcal{V}_3$, $\mathcal{V}_4$, $\mathcal{V}_5$, $\mathcal{B}_p$ et $\mathcal{B}_1$,
- $\mathcal{V}_1$, $\mathcal{V}_2$, $\mathcal{V}_3$, $\mathcal{V}_4$, $\mathcal{V}_5$, $\mathcal{B}_p$ et $\mathcal{B}_2$,
- $\mathcal{V}_1$, $\mathcal{V}_2$, $\mathcal{V}_3$, $\mathcal{V}_4$, $\mathcal{V}_5$, $\mathcal{B}_p$ et $\mathcal{B}_3$.

Chacune de ces chaînes contient les corps virtuels reliant le repère $\mathcal{F}_o$ à la plate-forme $\mathcal{B}_p$ du robot, puis les corps constituant la jambe choisie. Á l'aide de la méthode ci-avant, il est possible d'exprimer la matrice jacobienne $\mathbf{J}_{c,n_i}$ de la chaîne cinématique correspondant à la jambe i du robot.

$${}^0\mathbf{t}_i = \mathbf{J}_{c,n_i}\dot{\mathbf{q}}_{c,i} \tag{E1.7}$$

Le vecteur des coordonnées de ces articulations $\mathbf{q}_{c,i}$, est un sous-vecteur des coordonnées de l'architecture passive $\mathbf{q}_p$. En ajoutant à la matrice jacobienne $\mathbf{J}_{c,n_i}$ des vecteurs colonnes nuls pour les articulations ne faisant pas partie de la chaîne cinématique de la jambe i, on obtient :

$${}^o\mathbf{t}_i = \mathbf{J}_i\dot{\mathbf{q}}_p \tag{E1.8}$$

qui correspond à l'équation (E2.3) du modèle cinématique du robot parallèle volant.

Le même raisonnement permet, à partir de l'équation (E1.6), d'exprimer le torseur $\mathbf{t}_k$ pour n'importe quel corps $\mathcal{B}_k$ de la structure arborescente en fonction du vecteur $\dot{\mathbf{q}}_p$

$$\mathbf{t}_k = \mathbf{J}_k\dot{\mathbf{q}}_p \tag{E1.9}$$

A.2 Calcul du modèle dynamique

Cette partie explique le calcul du modèle dynamique de l'architecture arborescente passive, notamment le passage depuis l'expression du modèle dynamique sous la forme de l'équa-

tion (E2.9) à l'équation (E2.10). On considère donc une structure arborescente soumises aux efforts d'entrée $\boldsymbol{\tau}_p = \sum_{i=1}^{n} \mathbf{J}_i^{To} \mathbf{w}_{p,i}$. Les équations de Lagrange pour ce système s'écrivent :

$$\frac{\mathrm{d}}{\mathrm{dt}} \left(\frac{\partial L_p}{\partial \dot{\mathbf{q}}_p} \right)^T - \left(\frac{\partial L_p}{\partial \mathbf{q}_p} \right)^T = \boldsymbol{\tau}_p \tag{E1.10}$$

Exprimons un expression plus développée du Lagrangien L_p. Pour cela, nous allons introduire, pour chaque corps $\mathcal{B}_k$ du robot :

- m_k sa masse,
- G_k son centre de gravité,
- $\mathbf{ms}_k = m_k \mathbf{r}_{O_k G_k}$ son premier moment d'inertie au point R_k,
- $\mathbf{I}_k$ sa matrice d'inertie, exprimée dans le repère, $\mathcal{F}_k$ et $\mathbf{I}_{G_k}$ sa matrice d'inertie en G_k dans un repère parallèle au repère $\mathcal{F}_k$,
- $\mathbf{v}_k$ sa vitesse linéaire et $\boldsymbol{\omega}_k$ sa vitesse de rotation dans le repère $\mathcal{F}_k$,
- $\mathbf{v}_{G_k}$ sa vitesse linéaire au centre de gravité G_k.

Le Lagrangien L_p s'écrit :

$$L_p = \sum_k E_k - U_k \tag{E1.11}$$

avec E_k l'énergie cinétique et U_k l'énergie potentielle du corps $\mathcal{B}_k$.

Energie cinétique

L'énergie cinétique E_k du corps $\mathcal{B}_k$ en son centre de gravité s'écrit :

$$E_k = \frac{1}{2} (\boldsymbol{\omega}_k^T \mathbf{I}_{G_k} \boldsymbol{\omega}_k + m_k \mathbf{v}_{G_k}^T \mathbf{v}_{G_k}) \tag{E1.12}$$

Avec les relations de changement de repère

$$\mathbf{v}_{G_k} = \mathbf{v}_k - \widehat{\mathbf{r}}_{O_k G_k} \boldsymbol{\omega}_k \tag{E1.13}$$

$$\mathbf{I}_k = \mathbf{I}_{G_k} - m_k \widehat{\mathbf{r}}_{O_k G_k} \widehat{\mathbf{r}}_{O_k G_k} \tag{E1.14}$$

l'équation (E1.12) s'écrit :

$$E_k = \frac{1}{2} (\boldsymbol{\omega}_k^T \mathbf{I}_k \boldsymbol{\omega}_k + m_k \mathbf{v}_k^T \mathbf{v}_k - m_k \mathbf{v}_k^T \widehat{\mathbf{r}}_{O_k G_k} \boldsymbol{\omega}_k + m_k \boldsymbol{\omega}_k^T \widehat{\mathbf{r}}_{O_k G_k} \mathbf{v}_k) \tag{E1.15}$$

avec la notation $\widehat{\mathbf{r}}$ représentant une matrice telle que $\widehat{\mathbf{r}}\mathbf{u} = \mathbf{r} \times \mathbf{u}$. L'équation (E1.15) s'écrit sous forme matricielle :

$$E_k = \frac{1}{2} \mathbf{t}_k^T \mathbf{M}_k \mathbf{t}_k \tag{E1.16}$$

avec $\mathbf{M}_k = \begin{bmatrix} m_k\mathcal{I}_3 & -\widehat{\mathbf{ms}}_k \\ \widehat{\mathbf{ms}}_k & \mathbf{I}_k \end{bmatrix}$.
En introduisant (E1.9) dans (E1.16) on obtient :

$$E_k = \frac{1}{2}\dot{\mathbf{q}}_p^T \mathbf{J}_k^T \mathbf{M}_k \mathbf{J}_k \dot{\mathbf{q}}_p \tag{E1.17}$$

L'énergie cinétique pour l'ensemble du robot s'écrit alors :

$$E_p = \frac{1}{2}\dot{\mathbf{q}}_p^T \mathbf{M}_p \dot{\mathbf{q}}_p \tag{E1.18}$$

avec

$$\mathbf{M}_p = \sum_k \mathbf{J}_k^T \mathbf{M}_k \mathbf{J}_k$$

La matrice $\mathbf{M}_p$ est la matrice d'inertie généralisée du robot. Elle est symétrique définie positive.

Énergie potentielle

Seule les forces de gravitation interviennent dans l'énergie potentielle du robot. Celle-ci s'écrit alors pour le corps $\mathcal{B}_k$

$$U_k = -m_k \mathbf{g}^T \mathbf{r}_{\mathbf{OR_k}} \tag{E1.19}$$

Cette énergie est ne dépend que de la configuration du robot $\mathbf{q}_p$ et pas de la vitesse des articulation $\dot{\mathbf{q}}_p$. On note U_p l'énergie potentielle sommée sur tous les corps du robot.

Forme développée des équations de Lagrange

En introduisant (E1.11), (E1.18) et (E1.19) dans (E1.10), on obtient :

$$\mathbf{M}_p \ddot{\mathbf{q}}_p + \mathbf{c}_p = \boldsymbol{\tau}_p \tag{E1.20}$$

avec

$$\mathbf{c}_p = \dot{M}_p \dot{\mathbf{q}}_p - \left(\frac{\partial E_p}{\partial \mathbf{q}}\right)^T + \left(\frac{\partial U_p}{\partial \mathbf{q}}\right)^T$$

le vecteur $\mathbf{c}_p$ contient les termes d'effets centrifuges, de Coriolis et de gravité. Cette expression est utilisée dans l'équation (E2.10) pour exprimer le modèle dynamique de l'architecture passive du robot parallèle volant.

Cette méthode, pratique pour faire apparaître pour la structure du modèle dynamique du robot, n'est cependant pas la plus efficace en nombre d'opération. Le logiciel SYMORO+, utilisé dans le cadre de cette thèse, s'appuie sur un calcul récursif des équations de Newton-Euler. Le détail de cette méthode est donné dans [Khalil et Creusot, 1997].

Annexe B

Estimation de la position et orientation des drones

Pour l'estimation de la position et de l'orientation des drones, dans le cadre de l'expérimentation (voir chapitre 4), nous disposons, d'une part, des mesures fournies par la centrale inertielle (IMU) et, d'autre part, des mesures fournies par le système de capture de mouvement (MOCAP). La première partie de cette annexe présente une méthode pour l'estimation de l'orientation du drone uniquement à partir des mesures fournies par l'IMU. Cette estimation a été utilisée pour l'identification dynamique d'un drone. La seconde partie présente une méthode pour l'estimation de la position et de l'orientation des drones à partir des deux systèmes de mesure (IMU et MOCAP).

B.1 Estimation des orientations par la centrale inertielle

La centrale inertielle d'un quadricoptère est composée :

- d'un accéléromètre qui mesure l'accélération ${}^{d}\ddot{\boldsymbol{\xi}}$ du drone dans le repère local $\mathcal{F}_d$, (voir partie 1.1.1)
- d'un gyroscope qui mesure la vitesse angulaire ${}^{d}\boldsymbol{\omega}_d$ dans le repère local ${}^{d}\mathcal{F}_d$,
- d'un magnétomètre qui mesure l'intensité et l'orientation du champ magnétique terrestre ${}^{d}\mathbf{m}_{ag}$ dans le repère local ${}^{d}\mathcal{F}_d$.

A partir de ces données, il est possible d'estimer l'orientation du quadricoptère. Pour le PELICAN, cet algorithme a été programmé par le fournisseur du drone dans le LLP. Plusieurs algorithmes, appelés algorithmes AHRS (Attitude and Heading Reference Systems), existent et nous ne disposons pas du détail de l'implémentation du fournisseur. Nous rappelons cependant les principales étapes d'un algorithme AHRS ici.

Un algorithme AHRS calcule dans un premier temps l'estimation de l'orientation à partir des gyroscopes et l'estimation de l'orientation à partir l'accéléromètre et du magnétomètre. L'algorithme effectue ensuite la fusion de ces deux estimations.

Estimation de l'orientation à partir des gyroscopes

Les dérivés des coordonnées d'orientation du drone peuvent être reliées à la vitesse angulaire ${}^d\boldsymbol{\omega}_d$. Par exemple, pour une représentation par les angles de Bryant (E2.1), on a la relation donnée par l'équation (E2.19) :

$$ {}^d\boldsymbol{\omega}_d = \mathbf{T}_d \dot{\boldsymbol{\eta}}_d $$

avec $\mathbf{T}_d$ (E2.20) une matrice (3×3) fonction du vecteur $\boldsymbol{\eta}_d = [\varphi_d \ \theta_d \ \psi_d]^T$, le vecteur des coordonnées d'orientation du drone. Une relation similaire peut être obtenue pour une représentation des orientations par les quaternions [Madgwick, 2010].

Il est possible alors d'effectuer l'estimation des angles par intégration de l'équation (E2.19) à chaque pas de temps d'échantillonnage. Cette estimation, bien sûr, dérive au fur et à mesure que le temps passe.

Estimation de l'orientation à partir de l'accéléromètre et du magnétomètre

Pour l'algorithme AHRS, on effectue l'approximation que l'accéléromètre ne mesure que la gravité terrestre et que le magnétomètre ne mesure que le champ magnétique terrestre. Cette approximation est évidemment fausse dès que le drone subit une accélération mais s'avère suffisante, combinée à la mesure des gyroscopes, pour effectuer une estimation efficace de l'orientation du drone.

L'accélération $\mathbf{a}_d$ mesurée par l'accéléromètre, soumis à la gravitation, exprimée dans le repère local $\mathcal{F}_d$, s'écrit :

$$ \mathbf{a}_d = \mathbf{R}_d^{To} \mathbf{g} \tag{E2.1} $$

avec :

- ${}^o\mathbf{g} = [0 \ 0 \ -g]$ le vecteur de gravité, exprimé dans le repère $\mathcal{F}_o$,
- $\mathbf{R}_d$ la matrice de changement de repère donnée par (E2.2).

Seule la direction du vecteur de gravité est utile pour l'algorithme. On notera $\mathbf{u}_a$ le vecteur unitaire de la direction de l'accélération mesurée. À partir de l'équation (E2.1), $\mathbf{u}_a$ s'écrit :

$$ \mathbf{u}_a = -\mathbf{R}_d^T \mathbf{z}_o \tag{E2.2} $$

De même, on exprime le vecteur unitaire $\mathbf{u}_b$ de la direction du champ magnétique mesuré, ce vecteur s'écrit

$$ \mathbf{u}_a = \frac{\mathbf{R}_d^T \mathbf{b}_o}{||\mathbf{b}_o||_2} \tag{E2.3} $$

avec $\mathbf{b}_o$ le champ magnétique terrestre dans le repère $\mathcal{F}_o$.

L'estimation de l'orientation consiste à trouver la matrice de rotation $\mathbf{R}_d$ tel que

$$\mathbf{z}_o - \mathbf{R}_d \mathbf{u}_a = \mathbf{0} \tag{E2.4}$$

$$^o\mathbf{u}_b^* - \mathbf{R}_d \mathbf{u}_b = \mathbf{0} \tag{E2.5}$$

avec $^o\mathbf{u}_b^*$ la direction du champ magnétique local mesuré lors d'une phase de calibration préalable.

Le système d'équation (E2.4) peut être résolu par un algorithme de type descente de gradient [Mahony et al., 2008; Madgwick et al., 2011]. Il est également possible, suivant la représentation choisie des orientation, de calculer une solution algébrique à au système (E2.4) mais la méthode par descente de gradient peut être directement intégrée dans un algorithme complet fusionnant les deux estimations des orientations.

Fusion de données

L'estimation finale de l'algorithme AHRS consiste à fusionner les deux estimations précédemment introduites. Cette fusion de données peut également intégrer une estimation des biais de mesures. Deux algorithmes classiques pour effectuer cette fusion de données sont présentés dans [Mahony et al., 2008] et [Madgwick et al., 2011].

B.2 Fusion de données pour l'estimation de la position et de l'orientation des drones

Dans le cadre de cette thèse, la mesure de la position et de l'orientation des drones s'effectue à partir de deux systèmes de mesures (voir chapitre 4) :

- la centrale inertielle (IMU) qui fournit une vitesse angulaire $\boldsymbol{\omega}_m$ et une accélération $\mathbf{a}_m$ mesurées dans le repère local du drone $\mathcal{F}_d$,
- la capture de mouvement (MOCAP) qui fournit la position $^o\boldsymbol{\xi}$ et l'orientation $\boldsymbol{\eta}$ du drone par rapport au repère de référence $\mathcal{F}_o$.

Cette partie explique comment, dans la mise en œuvre de l'expérimentation, cette fusion de ces données a été effectuée. Elle est essentiellement inspirée de [Schinstock, 2010]. La fusion s'effectue à l'aide d'un filtre de Kalman étendu [Grewal et Andrews, 2008]. La théorie des filtres Kalman dépasse le cadre de cette thèse. Nous expliquerons ici les principes généraux et les détails de l'implémentation pour notre application. Le développement de cette partie s'effectue en trois étapes :

1. définition du modèle du système pour le filtre de Kalman,
2. linéarisation du modèle,
3. présentation de l'algorithme du filtre de Kalman.

B.2.1 Modèle du système

Le modèle du système pour le filtre de Kalman relie l'évolution des variables que nous cherchons à estimer aux mesures effectuées par la centrale inertielle. Dans cette partie, nous allons tenir compte du biais et du bruit sur les variables mesurées. Ainsi, le vecteur des mesures effectuées par l'IMU s'écrit :

$$\bar{\mathbf{u}} = \begin{bmatrix} \boldsymbol{\omega}_m \\ \mathbf{a}_m \end{bmatrix} = \begin{bmatrix} {}^d\boldsymbol{\omega} - \mathbf{w}_\omega + \mathbf{b}_\omega \\ {}_d\ddot{\boldsymbol{\xi}} - \mathbf{R}_d^{To}\mathbf{g} - \mathbf{w}_a + \mathbf{b}_a \end{bmatrix} \tag{E2.6}$$

avec $\mathbf{w}_\omega$ (respectivement $\mathbf{w}_a$) le vecteur de bruit et $\mathbf{b}_\omega$ (respectivement $\mathbf{b}_a$) le vecteur de biais pour le gyroscope (respectivement l'accéléromètre). Notons également que l'accéléromètre, en plus de l'accélération propre du drone, mesure en permanence une accélération due à la gravité $-\mathbf{R}_d^{To}\mathbf{g}$. Le vecteur $\bar{\mathbf{u}}$ est considéré comme l'entrée du système pour la conception du filtre de Kalman.

Les variables que nous cherchons à estimer par le filtre de Kalman sont assemblées dans un vecteur d'état :

$$\bar{\mathbf{x}} = \begin{bmatrix} {}^o\boldsymbol{\xi} \\ {}^o\dot{\boldsymbol{\xi}} \\ \boldsymbol{\eta} \\ \mathbf{b}_\omega \\ \mathbf{b}_a \end{bmatrix} \tag{E2.7}$$

Ce vecteur est composé de la position ${}^o\boldsymbol{\xi}$ et de la vitesse ${}^o\dot{\boldsymbol{\xi}}$ du drone dans le repère $\mathcal{F}_o$, des coordonnées d'orientation $\boldsymbol{\eta}$, et des vecteurs de biais de mesures.

L'évolution des biais est modélisé comme une marche aléatoire. Ainsi, la dérivée des vecteurs de biais est un vecteur de bruit aléatoire :

$$\begin{bmatrix} \dot{\mathbf{b}}_\omega \\ \dot{\mathbf{b}}_a \end{bmatrix} = \begin{bmatrix} \mathbf{w}_{b_\omega} \\ \mathbf{w}_{b_a} \end{bmatrix} \tag{E2.8}$$

Le vecteurs de bruit sont assemblés dans un vecteur $\bar{\mathbf{w}}$ qui s'écrit :

$$\bar{\mathbf{w}} = \begin{bmatrix} \mathbf{w}_\omega \\ \mathbf{w}_a \\ \mathbf{w}_{b_w} \\ \mathbf{w}_{b_a} \end{bmatrix} \tag{E2.9}$$

L'élaboration d'un filtre de Kalman nécessite une équation d'évolution des variables d'état. Celle ci est obtenue en insérant les équations (E2.19) et (E2.8) dans la dérivé de l'équation

(E2.7) :

$$\dot{\bar{\mathbf{x}}} = \begin{bmatrix} {}^{o}\dot{\boldsymbol{\xi}} \\ \mathbf{R}_d {}^{d}\ddot{\boldsymbol{\xi}} \\ \mathbf{T}_d^{-1\,d}\boldsymbol{\omega} \\ \mathbf{w}_{b_w} \\ \mathbf{w}_{b_a} \end{bmatrix} \tag{E2.10}$$

À l'aide des équations (E2.10) et (E2.6), il est possible d'exprimer l'évolution des variables d'état en fonction du vecteur des mesures $\bar{\mathbf{u}}$:

$$\dot{\bar{\mathbf{x}}} = \begin{bmatrix} {}^{o}\dot{\boldsymbol{\xi}} \\ \mathbf{R}_d(\mathbf{a}_m + \mathbf{w}_a - \mathbf{b}_a) + {}^{o}\mathbf{g} \\ \mathbf{T}_d^{-1}(\boldsymbol{\omega}_m + \mathbf{w}_\omega - \mathbf{b}_\omega) \\ \mathbf{w}_{b_w} \\ \mathbf{w}_{b_a} \end{bmatrix} \tag{E2.11}$$

L'équation (E2.11) est un système d'équation non linéaire qui décrit l'évolution des variables d'état en fonction des vecteurs des variables d'état $\bar{\mathbf{x}}$, des mesures $\bar{\mathbf{u}}$ et des bruits $\bar{\mathbf{w}}$. Elle sera notée sous la forme :

$$\dot{\bar{\mathbf{x}}} = \mathbf{f}(\bar{\mathbf{x}}, \bar{\mathbf{u}}, \bar{\mathbf{w}}) \tag{E2.12}$$

Les mesures fournies par la capture de mouvements vont servir à corriger l'estimation effectuée par le filtre de Kalman. Elles sont considérées comme les sorties du système, rassemblées dans un vecteur $\bar{\mathbf{y}}$, donné par :

$$\bar{\mathbf{y}} = \begin{bmatrix} {}^{o}\boldsymbol{\xi} \\ \boldsymbol{\eta} \end{bmatrix} + \bar{\mathbf{v}} \tag{E2.13}$$

avec $\mathbf{v}$ le vecteur de bruit de mesure sur les variables mesurées par la capture de mouvements.

B.2.2 Linéarisation du modèle

Le filtre de Kalman s'applique pour un système à états discrets. Le système donné par les équations (E2.11) et (E2.13) est assimilé à son équivalent discret pour une fréquence d'échantillonnage élevée :

$$\mathbf{x}_k = \mathbf{x}_{k-1} + T\mathbf{f}(\mathbf{x}_{k-1}, \mathbf{u}_k, \mathbf{w}_k) \tag{E2.14}$$

$$\mathbf{y}_k = \mathbf{H}\mathbf{x}_k + \mathbf{v}_k \tag{E2.15}$$

avec

- $\mathbf{x}_k$ (respectivement $\mathbf{w}_k$, $\mathbf{y}_k$ et $\mathbf{v}_k$) le vecteur $\bar{\mathbf{x}}$ (respectivement $\bar{\mathbf{w}}$, $\bar{\mathbf{y}}$ et $\bar{\mathbf{v}}$) discrétisé à l'instant k,
- T le temps d'échantillonnage (pour notre expérimentation T =1 ms).

- $\mathbf{H} = \left[\begin{array}{c|c|c|c|c} \mathcal{I}_3 & \mathbf{0}_{3\times3} & \mathbf{0}_{3\times3} & \mathbf{0}_{3\times3} & \mathbf{0}_{3\times3} \\ \hline \mathbf{0}_{3\times3} & \mathbf{0}_{3\times3} & \mathcal{I}_3 & \mathbf{0}_{3\times3} & \mathbf{0}_{3\times3} \end{array}\right]$

De plus, l'application du filtre Kalman étendu implique de linéariser la fonction $\mathbf{f}$ dans l'équation (E2.14). Pour cela, nous allons définir les deux matrices jacobiennes suivantes :

$$\mathbf{F} = \left.\frac{\partial \mathbf{f}}{\partial \bar{\mathbf{x}}}\right|_{\mathbf{x}_{k-1},\mathbf{u}_k,\mathbf{0}}$$
$$\mathbf{G} = \left.\frac{\partial \mathbf{f}}{\partial \bar{\mathbf{w}}}\right|_{\mathbf{x}_{k-1},\mathbf{u}_k,\mathbf{0}}$$

Les matrices jacobiennes prennent les formes suivantes :

$$\mathbf{F} = \left[\begin{array}{c|c|c|c|c} \mathbf{0}_{3\times3} & \mathcal{I}_3 & \mathbf{0}_{3\times3} & \mathbf{0}_{3\times3} & \mathbf{0}_{3\times3} \\ \hline \mathbf{0}_{3\times3} & \mathbf{0}_{3\times3} & \frac{\partial \mathbf{R}_d}{\partial \boldsymbol{\eta}}(\mathbf{a}_m - \mathbf{b}_a + \mathbf{w}_a) & \mathbf{0}_{3\times3} & -\mathbf{R}_d \\ \hline \mathbf{0}_{3\times3} & \mathbf{0}_{3\times3} & \frac{\partial \mathbf{T}_d^{-1}}{\partial \boldsymbol{\eta}}(\boldsymbol{\omega}_m - \mathbf{b}_\omega + \mathbf{w}_\omega) & -\mathbf{T}_d^{-1} & \mathbf{0}_{3\times3} \\ \hline \mathbf{0}_{3\times3} & \mathbf{0}_{3\times3} & \mathbf{0}_{3\times3} & \mathbf{0}_{3\times3} & \mathbf{0}_{3\times3} \\ \hline \mathbf{0}_{3\times3} & \mathbf{0}_{3\times3} & \mathbf{0}_{3\times3} & \mathbf{0}_{3\times3} & \mathbf{0}_{3\times3} \end{array}\right]$$

$$\mathbf{G} = \left[\begin{array}{c|c|c|c} \mathbf{0}_{3\times3} & \mathbf{0}_{3\times3} & \mathbf{0}_{3\times3} & \mathbf{0}_{3\times3} \\ \hline \mathbf{0}_{3\times3} & \mathbf{R}_d & \mathbf{0}_{3\times3} & \mathbf{0}_{3\times3} \\ \hline \mathbf{T}_d^{-1} & \mathbf{0}_{3\times3} & \mathbf{0}_{3\times3} & \mathbf{0}_{3\times3} \\ \hline \mathbf{0}_{3\times3} & \mathbf{0}_{3\times3} & \mathcal{I}_3 & \mathbf{0}_{3\times3} \\ \hline \mathbf{T}_d^{-1} & \mathbf{0}_{3\times3} & \mathbf{0}_{3\times3} & \mathcal{I}_3 \end{array}\right]$$

La linéarisation au premier ordre de l'équation (E2.14) s'écrit alors :

$$\mathbf{x}_k = (\mathcal{I}_{16} + T\mathbf{F}\mathbf{x}_{k-1})\mathbf{x}_{k-1} + T\mathbf{G}\mathbf{w}_k \tag{E2.16}$$

B.2.3 Algorithme du filtre de Kalman

Le filtre de Kalman dans le contexte discret est un estimateur récursif. Cela signifie que pour estimer l'état courant, seules l'estimation de l'état précédent et les mesures actuelles sont nécessaires. L'état du filtre est représenté par deux variables :

- $\hat{\mathbf{x}}_k$ l'estimation de l'état $\mathbf{x}_k$ à l'instant k,
- $\mathbf{P}_k$ la matrice de covariance de l'erreur, qui est une mesure de la précision de l'état estimé.

Le filtre de Kalman requiert les matrices de covariance des bruits de mesure $\mathbf{w}_k$ et $\mathbf{v}_k$. Ces matrices sont notées respectivement $\mathbf{Q}$ et $\mathbf{R}$. On émet l'hypothèse simplificatrice de l'indépendance des bruits de mesures. Ainsi, les matrice $\mathbf{Q}$ et $\mathbf{R}$ sont diagonales. Les six premières valeurs de la diagonales de $\mathbf{Q}$ sont les variances sur les bruits de mesure du gyroscope et de l'accéléromètre. Les six valeurs suivantes vont déterminer le pas pour l'estimation des biais

de ces instruments. Les six valeurs de la diagonale de $\mathbf{R}$ sont les variances sur les bruits de mesure de la capture de mouvement.

L'algorithme du filtre de Kalman s'effectue en deux étapes. Une étape de prédiction calcule l'évolution de l'état du filtre à partir de l'équation d'évolution du système (E2.14). Ensuite, une étape de correction vient adapter l'état du filtre en fonction de la sortie $\mathbf{y}_k$ mesurée. Dans le cas particulier de notre expérimentation, l'étape de prédiction s'effectue à chaque pas de temps du contrôleur (fréquence 1 kHz) tandis l'étape de correction ne s'effectue qu'à la réception des mesures de la capture de mouvement (fréquence 250 Hz).

Les équations de l'algorithme du filtre de Kalman sont :

- **Prédiction :**

$$\hat{\mathbf{x}}_k = \hat{\mathbf{x}}_{k-1} + T\mathbf{f}(\mathbf{x}_{k-1}, \mathbf{u}_k, \mathbf{0}) \tag{E2.17}$$

$$\mathbf{P}_k = (\mathcal{I}_{16} + \mathbf{F}T)\mathbf{P}_{k-1}(\mathcal{I}_{16} + \mathbf{F}T)^T + T^2\mathbf{G}\mathbf{Q}\mathbf{G}^T \tag{E2.18}$$

  L'équation (E2.17) effectue la mise à jour de l'état à partir de l'équation (E2.14) en supposant le vecteur de bruit égal à zéro. L'équation (E2.18) effectue la mise à jour de la confiance dans l'estimation grâce à la matrice de covariance sur les bruits de mesure $\mathbf{Q}$.

- **Correction :**

$$\mathbf{K} = \mathbf{P}_k\mathbf{H}^T(\mathbf{H}\mathbf{P}_k\mathbf{H}^T + \mathbf{R})^{-1} \tag{E2.19}$$

$$\hat{\mathbf{x}}_k \Leftarrow \hat{\mathbf{x}}_k + \mathbf{K}(\mathbf{y}_k - \mathbf{H}\hat{\mathbf{x}}_k) \tag{E2.20}$$

$$\mathbf{P}_k = \mathbf{P}_k - \mathbf{K}\mathbf{H}\mathbf{P}_k \tag{E2.21}$$

  L'équation (E2.19) détermine une matrice de gain $\mathbf{K}$ à partir de la confiance dans l'estimation $\mathbf{P}_k$ et de la confiance dans la mesure corrective $\mathbf{R}$. L'équation (E2.20) corrige la prédiction $\hat{\mathbf{x}}_k$ au regard de l'écart entre l'estimation et la mesure $\mathbf{y}_k - \mathbf{H}\hat{\mathbf{x}}_k$. La matrice de gain $\mathbf{K}$ intervient dans cette correction. L'équation (E2.21) met à jour la confiance dans l'estimation suite à cette correction.

Annexe C

Article sur la méthode de traversée des singularités pour les robots parallèles.

A Controller Avoiding Dynamic Model Degeneracy of Parallel Robots during Singularity Crossing

Damien Six[a,b]
Sébastien Briot[a]
Abdelhamid Chriette[a,b]
Philippe Martinet[a,b]

a. LS2N, UMR CNRS 6004
b. Ecole Centrale Nantes
Nantes 44300
France
{damien.six,sebastien.briot,abdelhamid.chriette,philippe.martinet}@ls2n.fr

Parallel robots present singular configurations that divide the operational workspace into several aspects. It was proven that Type 2 and Leg Passive Joint Twist System singularities can be crossed with a trajectory respecting a given dynamic criterion. However, the practical implementation of a controller able to track such trajectories is up to now limited to restrictive cases of Type 2 singularities crossing. Analyzing the structure of the inverse dynamic model, this paper proposes a global solution allowing the tracking of trajectories respecting the general criterion for any singularity that leads to potential issues of dynamic model degeneracy. The tracking is operated in the robot joint space. Experimental results on a five-bar mechanism showed the controller ability to successfully cross Type 2 singularities.

1 Introduction

Parallel robots present many advantages over serial robots as higher acceleration capacities, better stiffness and payload-to-weight ratio. However, their use in an industrial context remains limited due, in part, to the division of their workspace in several aspects by singularities [1]. Numerous singularities are prone to appear on a parallel mechanism. For a global overview the reader is referred to [2–4]. In some singularities, the inverse dynamic model of the mechanism does not admit a finite solution without the respect of a dynamic criterion [5]. Those singularities will be referred as Type 2 [6] and Leg Passive Joint Twist System (LPJTS) singularities, which are a kind of leg singularities [4]. The scope of this paper is restricted to the above mentioned singularities. The other singularities such as constraint singularities are not covered.

Several approaches have been explored in the literature to increase the size of the operational workspace of parallel robot, which is usually limited by the presence of singularities. The main approaches are:

- The design of parallel robots without singularities [7–9]. This solution is the most usual one, but it usually leads to the design of robots with a small workspace size or robot architectures with very low practicability.
- The use of redundancy [10–12], which increases the actuation cost.
- Mechanisms with variable actuation modes [13]. This solution is generally carried out when the mechanism is stopped.
- The modification of the Type 2 singularities locus by changing the robot working mode. The main way to proceed is to cross a Type 1 singularity by reaching the workspace boundary and changing the leg configuration [14]. This method however may not grant an access to the complete workspace for all robot architecture.
- The planning of assembly mode changing trajectories encircling cusp points (for Type 2 singularities) [15]. This solution requires the design of complex trajectories, which will not be the most efficient for fast operations, and only few mechanism have cusp points.
- The planning of assembly mode changing trajectories directly through the singularity [5, 16] with respect to a dynamic criterion when crossing the singularity.

Crossing the singularity seems a promising solution because a direct path between two points can be generated on any robot structure as long as this path respects the dynamic criterion for crossing. In practical implementation, the precision required on the trajectory tracking is not obtained with a simple PID controller and the effective crossing of the sin-

gularity is not guaranteed. Then, the use of a dynamic controller as computed-torque control have been explored. Unfortunately, the computation of the inverse dynamic model in such controller require the inversion of matrices that become singular in Type 2 or LPJTS singularities. This issue was partially solved in [17] and [18]. However, both solutions are limited to Type 2 singularities. Also the solution in [17] requires a restrictive trajectories and the solution in [18] is limited to a control of the Cartesian coordinates of the robot tool, which are rarely available on industrial robots.

This paper proposes a global solution allowing the crossing of Type 2 and LPJTS singularities with a computed-torque control, valid for any trajectory that crosses the singularity, as long as the dynamic criterion established in [5, 16] is respected. To obtain this solution, the following improvements are performed from existing solutions:

1. The proof that the dynamic criterion expressed in the previous studies is not only a necessary condition to cross the singularity but guarantee a solution to the inverse dynamic model, which is expressed for Type 2 and LPJTS singularities.
2. A projection of the desired wrench when solving the inverse dynamic model in the control law is developed to avoid unbounded input torques from numerical uncertainties and tracking errors.
3. The extension of the second order kinematic equations to singular configurations is obtained to allow a feedback control law in joint space.

The present paper is structured as follows. Section 2 discusses the expression of the inverse dynamic model that remains valid in singularity. Section 3 presents a control law designed to track a trajectory crossing a singularity. The validation of the designed controller is discussed through experimentation in section 4. Conclusions are given in section 5.

2 Dynamic model at singularity locus

2.1 Dynamic modeling of parallel mechanisms

In this section, the dynamic equations of a parallel manipulator are briefly recalled. For a more detailed analysis, the reader is referred to [19, 20]. The studied manipulator is composed of a fixed base, linked by several kinematic chains (the legs) to a mobile platform actuated along n_{dof} independent coordinates. Actuation is provided by n_{dof} active joints. n_d passive joints are presents in the manipulator legs. The configuration and velocity of the manipulator can be described using:

- $\mathbf{q}_a$ and $\dot{\mathbf{q}}_a$ two n_{dof}-dimensional vectors of active joint variables and active joint velocities respectively.
- $\mathbf{q}_d$ and $\dot{\mathbf{q}}_d$ two n_d-dimensional vectors of passive joint variables and passive joint velocities respectively.
- $\mathbf{x}$ and $\dot{\mathbf{x}}$ two n_{dof}-dimensional vectors of the independent platform coordinates and their time derivative respectively.

The relations between those coordinates are found by writing the closed-loop equations $\mathbf{h}(\mathbf{x},\mathbf{q}_a)=\mathbf{0}$ and $\mathbf{h}_a(\mathbf{x},\mathbf{q}_a,\mathbf{q}_d)=\mathbf{0}$. The time derivative of those closed-loop equations gives the kinematic constraints equations

$$\mathbf{A}\dot{\mathbf{x}}+\mathbf{B}\dot{\mathbf{q}}_a=\mathbf{0} \tag{1}$$

$$\mathbf{J}_x\dot{\mathbf{x}}-\mathbf{J}_a\dot{\mathbf{q}}_a-\mathbf{J}_d\dot{\mathbf{q}}_d=\mathbf{0} \tag{2}$$

with $\mathbf{A}$, $\mathbf{B}$, $\mathbf{J}_a$, $\mathbf{J}_x$ and $\mathbf{J}_d$ of respective size $(n_{dof}\times n_{dof})$, $(n_{dof}\times n_{dof})$, $(n_d\times n_{dof})$, $(n_d\times n_{dof})$ and $(n_d\times n_d)$. Let us also consider the Lagrangian L of the system and $\boldsymbol{\tau}_a$, $\boldsymbol{\tau}_d$ and $\mathbf{w}_x$ respectively n_{dof}, n_d and n_{dof}-dimensional vectors related to the Lagrangian by

$$\boldsymbol{\tau}_a=\frac{\mathrm{d}}{\mathrm{dt}}\left(\frac{\partial L}{\partial\dot{\mathbf{q}}_a}\right)^T-\left(\frac{\partial L}{\partial\mathbf{q}_a}\right)^T \tag{3}$$

$$\boldsymbol{\tau}_d=\frac{\mathrm{d}}{\mathrm{dt}}\left(\frac{\partial L}{\partial\dot{\mathbf{q}}_d}\right)^T-\left(\frac{\partial L}{\partial\mathbf{q}_d}\right)^T \tag{4}$$

$$\mathbf{w}_x=\frac{\mathrm{d}}{\mathrm{dt}}\left(\frac{\partial L}{\partial\dot{\mathbf{x}}}\right)^T-\left(\frac{\partial L}{\partial\mathbf{x}}\right)^T \tag{5}$$

Using method of Lagrange multipliers, the inverse dynamic model of the mechanism can be written as [20]

$$\boldsymbol{\tau}=\boldsymbol{\tau}_a-\mathbf{B}^T\boldsymbol{\lambda}_1-\mathbf{J}_a^T\boldsymbol{\lambda}_2 \tag{6}$$

$$\mathbf{A}^T\boldsymbol{\lambda}_1-\mathbf{J}_x^T\boldsymbol{\lambda}_2=\mathbf{w}_x \tag{7}$$

$$\mathbf{J}_d^T\boldsymbol{\lambda}_2=\boldsymbol{\tau}_d \tag{8}$$

where

- $\boldsymbol{\tau}$ is the n_{dof}-dimensional vector of the input efforts,
- $\boldsymbol{\lambda}_1$ and $\boldsymbol{\lambda}_2$ are a n_{dof}-dimensional and n_d-dimensional vectors of Lagrange multipliers respectively,

From the expression of the Lagrangian of the mechanical system, equations (3)-(5) can be obtained under the form [20]

$$\boldsymbol{\tau}_a=\mathbf{M}_a\begin{bmatrix}\ddot{\mathbf{q}}_a\\ \ddot{\mathbf{q}}_d\end{bmatrix}+\mathbf{c}_a \tag{9}$$

$$\boldsymbol{\tau}_d=\mathbf{M}_d\begin{bmatrix}\ddot{\mathbf{q}}_a\\ \ddot{\mathbf{q}}_d\end{bmatrix}+\mathbf{c}_d \tag{10}$$

$$\mathbf{w}_x=\mathbf{M}_x\ddot{\mathbf{x}}+\mathbf{c}_x \tag{11}$$

where

- $\mathbf{M}_a$ is a $(n_{dof}\times(n_{dof}+n_p))$ matrix and $\mathbf{M}_d$ a $(n_p\times(n_{dof}+n_p))$ matrix depending on the joint coordinates $\mathbf{q}_a$ and $\mathbf{q}_d$,
- $\mathbf{c}_a$ is a n_{dof}-dimensional vector and $\mathbf{c}_d$ a n_d-dimensional vector depending on the joint coordinates $\mathbf{q}_a$ and $\mathbf{q}_d$ and velocities $\dot{\mathbf{q}}_a$ and $\dot{\mathbf{q}}_d$,
- $\mathbf{M}_x$ is a $(n_{dof}\times n_{dof})$ definite positive matrix depending on the platform coordinates $\mathbf{x}$,
- $\mathbf{c}_r$ a n_{dof}-dimensional vector depending on the platform coordinates $\mathbf{x}$ and their time derivative $\dot{\mathbf{x}}$.

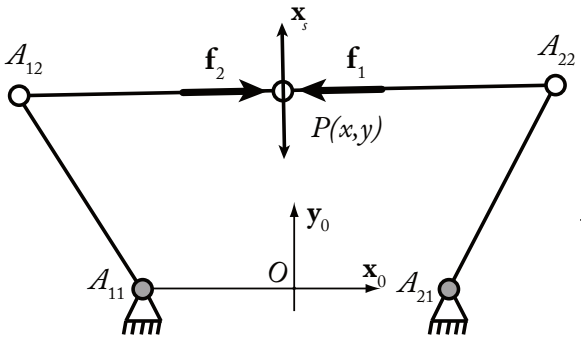


Figure 1: Example of a five-bar mechanism in a Type 2 singularity configuration. The vector $\dot{\mathbf{x}}_s$ represents the direction of the uncontrolled platform motion. $\mathbf{f}_1$ and $\mathbf{f}_2$ are the forces that can be generated through actuation on the platform. In this configuration, the resultant force that can be applied by the system on the platform is collinear to $\mathbf{f}_1$ and $\mathbf{f}_2$. Two actuators are used to generate an effort along one direction, an overconstraint is consequently generated in the mechanism.

2.2 Solution to the inverse dynamic model and degeneracy conditions

In this section, the main results concerning the general solution to the inverse dynamic model equations (6)-(8) and the degeneracy conditions of this model are recalled. For a detailed study of the dynamic model and its degeneracy conditions, the reader is referred to [5]. The general solution to the inverse dynamic model of a parallel robot can be computed by expressing $\boldsymbol{\lambda}_2$ from (8) and $\boldsymbol{\lambda}_1$ from (7) and by introducing them into (6)

$$\boldsymbol{\tau} = \boldsymbol{\tau}_a - \mathbf{J}_a^T \mathbf{J}_d^{-T} \boldsymbol{\tau}_d - \mathbf{B}^T \mathbf{A}^{-T} \mathbf{w}_d \tag{12}$$

with $\mathbf{w}_d = \mathbf{w}_x + \mathbf{J}_x^T \mathbf{J}_d^{-T} \boldsymbol{\tau}_d$. The expression of the dynamic model under this form requires a full rank of the matrices $\mathbf{A}$ and $\mathbf{J}_d$ [5]. However, this requirement is not always met. The rank degeneracy of those matrices is associated to kinematic singularities of parallel mechanisms identified in the literature:

- The degeneracy of the matrix $\mathbf{A}$ corresponds to a kinematic singularity named Type 2 singularity [6]. In such singularity, one (or more) of the degrees of freedom of the platform becomes uncontrollable (see Fig. 1).
- The degeneracy of the matrix $\mathbf{J}_d$ corresponds to a kinematic singularity named LPJTS singularity [4, 20]. In such singularity, one (or more) of the leg gains an uncontrollable motion on a passive joint, even if the platform is fully controlled.

For clarity purposes, it is assumed that only one direction is uncontrollable at singularity locus, corresponding to rank degeneracy of the matrix $\mathbf{A}$ or $\mathbf{J}_d$ of order one[1]. It was previously proved that the inverse dynamic model remains consistent under the respect of a dynamic criterion when crossing a singularity. The criteria established are summarized in Table 1. Proofs and considerations on those criteria can be found in [21] and extended to LPJTS singularities in [5].

Table 1: Dynamic criterion to avoid degeneracy of the inverse dynamic model in a singularity

Singularity	Type 2 [21]	LPJTS [5]
Criterion	$\dot{\mathbf{x}}_s^T \mathbf{w}_d = 0$	$\dot{\mathbf{q}}_{ds}^T \boldsymbol{\tau}_d = 0$
	$\dot{\mathbf{x}}_s \in \ker(\mathbf{A})$	$\dot{\mathbf{q}}_{ds} \in \ker(\mathbf{J}_d)$

2.3 Solution to the inverse dynamic model at singularity locus

Equation (12) gives a solution to the dynamic model away from singularities. In this section, a solution to the inverse dynamic model in a singularity is given under the respect of the criteria described in Table 1.

2.3.1 Type 2 singularity

Let us assume a parallel robot on a Type 2 singularity with no other singularity on the same locus[2]. Then, the inverse dynamic equations (6)-(8) can be written under the form

$$\boldsymbol{\tau} = \boldsymbol{\tau}_a - \mathbf{B}^T \boldsymbol{\lambda}_1 - \mathbf{J}_a^T \mathbf{J}_d^{-T} \boldsymbol{\tau}_d \tag{13}$$

$$\mathbf{A}^T \boldsymbol{\lambda}_1 = \mathbf{w}_d \tag{14}$$

with $\mathbf{w}_d = \mathbf{w}_x + \mathbf{J}_x^T \mathbf{J}_d^{-T} \boldsymbol{\tau}_d$. The rank degeneracy of the matrix $\mathbf{A}$ implies that the equation (14) will give at least one exact solution $\boldsymbol{\lambda}_1$ if and only if $\mathbf{w}_d$ is included in the range of the matrix $\mathbf{A}^T$. The range of $\mathbf{A}^T$ represents all the total wrenches $\mathbf{w}_d$ that can be applied by the legs through actuation and external forces on the platform. From linear algebra, it is known that the range of the matrix $\mathbf{A}^T$ is the orthogonal complement of the kernel of the matrix $\mathbf{A}$

$$\ker(\mathbf{A})^{\perp} = \mathrm{Im}(\mathbf{A}^T) \tag{15}$$

If the dynamic criterion expressed in table 1 is respected, then $\mathbf{w}_d$ is in the range of the matrix $\mathbf{A}$ and the inverse dynamic equations admits at least an exact solution. In order to find it, the inverse dynamic model is projected on the platform efforts by pre-multiplying equation (13) by $\mathbf{A}^T \mathbf{B}^{-T}$

$$\mathbf{A}^T \mathbf{B}^{-T} \boldsymbol{\tau} = \mathbf{A}^T \mathbf{B}^{-T} \boldsymbol{\tau}_a - \mathbf{A}^T \boldsymbol{\lambda}_1 - \mathbf{A}^T \mathbf{B}^{-T} \mathbf{J}_a^T \mathbf{J}_d^{-T} \boldsymbol{\tau}_d \tag{16}$$

Then, $\mathbf{A}^T \boldsymbol{\lambda}_1$ can be replaced in (16) from (14), giving

$$\mathbf{J}_{T2}^T \boldsymbol{\tau} = \mathbf{w}_{T2} \tag{17}$$

[1] Even if the case rarely appears on existing parallel robots, the results can be extended to to a higher order degeneracy.

[2] This hypothesis is taken given that the case of the coincidence of two singularities is extremely rare and generally avoided in the design of a parallel robot.

with $\mathbf{J}_{T2} = \mathbf{B}^{-1}\mathbf{A}$ and
$\mathbf{w}_{T2} = \mathbf{J}_{T2}^T\boldsymbol{\tau}_a - \mathbf{w}_x - (\mathbf{J}_x^T + \mathbf{J}_{T2}^T\mathbf{J}_a^T)\mathbf{J}_d^{-T}\boldsymbol{\tau}_d$.
Knowing that a solution to the inverse dynamic model (13)-(14) is garanteed if the dynamic criterion is respected, one solution can be expressed using the Moore-Penrose pseudo-inverse

$$\boldsymbol{\tau} = {\mathbf{J}_{T2}^T}^{+}\mathbf{w}_{T2} \tag{18}$$

Since the matrix $\mathbf{J}_{T2}^T$ is not full rank, its kernel is not empty, and so there is an infinite number of solutions to the inverse dynamic model including (18). An illustration is given Fig. 1, on a five-bar mechanism. The wrenches generated by the system on the platform at singularity are two collinear forces. Their opposite action generates an overconstraint in the system. Several combination of those forces (and so several input torques) can generate a desired acceleration of the platform. Equation (18) gives the solution to the inverse dynamic model minimizing the input torques, which is usually the desired solution to limit the power consumption of the mechanical system. However, for specific applications, it is possible to express a general solution to the inverse dynamic model under the form

$$\boldsymbol{\tau} = {\mathbf{J}_{T2}^T}^{+}\mathbf{w}_{T2} + (\mathbf{I}_{n_{dof}} - {\mathbf{J}_{T2}^T}^{+}\mathbf{J}_{T2}^T)\boldsymbol{\nu} \tag{19}$$

with $\boldsymbol{\nu}$ a n_{dof}-dimensional arbitrary vector called the *overconstraint*. Among the possible solutions to the inverse dynamic model, the solution minimizing the input torques, given by equation (18), will be retained. Note also that, out of a singularity, the following equality is obtained ${\mathbf{J}_{T2}^T}^{+} = \mathbf{J}_{T2}^{-T}$. Then, the solution to the inverse dynamic model under the form (18) is also valid out of the singularity (equivalent to equation (12)).

2.3.2 LPJTS singularity

An equivalent inverse dynamic model can be expressed for a LPJTS singularity. In this section, a solution to the inverse dynamic model on a LPJTS singularity is expressed under the respect of the criterion given in Table 1. The inverse dynamic equations (6)-(8) can be written under the form

$$\boldsymbol{\tau} = \boldsymbol{\tau}_a - \mathbf{B}^T\mathbf{A}^{-T}\mathbf{w}_x - \mathbf{J}_{da}^T\boldsymbol{\lambda}_2 \tag{20}$$
$$\mathbf{J}_d^T\boldsymbol{\lambda}_2 = \boldsymbol{\tau}_d \tag{21}$$

with $\mathbf{J}_{da} = \mathbf{J}_x\mathbf{A}^{-1}\mathbf{B} + \mathbf{J}_a$. Note that $\mathbf{J}_{da}$ is a Jacobian matrix linking the velocity of the active joints to the velocity of the passive joints

$$\mathbf{J}_{da}\dot{\mathbf{q}}_a + \mathbf{J}_d\dot{\mathbf{q}}_d = \mathbf{0} \tag{22}$$

The degeneracy of the matrix $\mathbf{J}_{da}$ corresponds to a rare singularity where the mechanism can generate a non-null active joint velocity $\dot{\mathbf{q}}_a$ with no motion of the passive joints. Let us consider that the LPJTS singularity does not coincide with such singularity or a Type 2 singularity[3]. The rank degeneracy of the matrix $\mathbf{J}_d^T$ implies that the equation (21) admits at least one exact solution $\boldsymbol{\lambda}_2$ if and only if $\boldsymbol{\tau}_d$ is included in the range of $\mathbf{J}_d^T$ which is an equivalent condition to the criterion previously defined to cross a LPJTS singularity in Table 1. Projecting the inverse dynamic model (20)-(21) on the efforts exerted on the passive joints gives

$$\mathbf{J}_p^T\boldsymbol{\tau} = \mathbf{w}_p \tag{23}$$

with $\mathbf{J}_p = \mathbf{J}_{da}^{-1}\mathbf{J}_d$ and
$\mathbf{w}_p = -\boldsymbol{\tau}_d + \mathbf{J}_d^T\mathbf{J}_{da}^{-T}(\boldsymbol{\tau}_a - \mathbf{B}^T\mathbf{A}^{-T}\mathbf{w}_x)$
With the respect of the criterion (21), the exact solution to the inverse dynamic model minimizing the input torques is

$$\boldsymbol{\tau} = {\mathbf{J}_p^T}^{+}\mathbf{w}_p \tag{24}$$

As in the case of a Type 2 singularity, the general solution to the inverse dynamic model (20)-(21) can be expressed under the form

$$\boldsymbol{\tau} = {\mathbf{J}_p^T}^{+}\mathbf{w}_p + (\mathbf{I}_{n_{dof}} - {\mathbf{J}_p^T}^{+}\mathbf{J}_p^T)\boldsymbol{\nu} \tag{25}$$

with $\boldsymbol{\nu}$ an arbitrary vector of overconstraint.

3 Control law

In this section, the design of a computed torque control law in order to track trajectories crossing a singularity is discussed. As industrial robot sensors usually measure the active joint coordinates, the control designed is based on the joint coordinates of the mechanical system.

3.1 Extended second-order loop-closure equations in the neighborhood of a singularity

The inverse dynamic model of a parallel mechanism contains terms function of the platform, passive and active joints accelerations. To design a computed torque control in joint space, it is necessary to express those accelerations as function of the controlled coordinates acceleration, i.e. the active joint accelerations. The time derivative of loop-closure equations (1) and (2) gives

$$\mathbf{A}\ddot{\mathbf{x}} + \mathbf{B}\ddot{\mathbf{q}}_a + \mathbf{b} = \mathbf{0} \tag{26}$$
$$\mathbf{J}_x\ddot{\mathbf{x}} - \mathbf{J}_a\ddot{\mathbf{q}}_a - \mathbf{J}_d\ddot{\mathbf{q}}_d + \mathbf{d} = \mathbf{0} \tag{27}$$

with $\mathbf{b} = \dot{\mathbf{A}}\mathbf{t}_r + \dot{\mathbf{B}}\dot{\mathbf{q}}_a$ and $\mathbf{d} = \dot{\mathbf{J}}_x\dot{\mathbf{x}} - \dot{\mathbf{J}}_{ta}\dot{\mathbf{q}}_a - \dot{\mathbf{J}}_{td}\dot{\mathbf{q}}_d$. Out of a Type 2 singularity, those two equations can be combined

[3]This hypothesis is taken given that the case of the coincidence of two singularities is extremely rare and generally avoided in the design of a parallel robot.

giving

$$\mathbf{J}_{da}\ddot{\mathbf{q}}_a + \mathbf{J}_d\ddot{\mathbf{q}}_d + \mathbf{n} = \mathbf{0} \tag{28}$$

with $\mathbf{J}_{da} = \mathbf{J}_x\mathbf{A}^{-1}\mathbf{B} + \mathbf{J}_a$ and $\mathbf{n} = \mathbf{J}_x\mathbf{A}^{-1}\mathbf{b} - \mathbf{d}$. Then, out of singularities, the passive joints acceleration and platform acceleration can be expressed as function of the active joint acceleration with

$$\ddot{\mathbf{x}} = -\mathbf{A}^{-1}\mathbf{B}\ddot{\mathbf{q}}_a - \mathbf{A}^{-1}\mathbf{b} \tag{29}$$

$$\ddot{\mathbf{q}}_d = -\mathbf{J}_d^{-1}\mathbf{J}_{da}\ddot{\mathbf{q}}_a - \mathbf{J}_d^{-1}\mathbf{n} \tag{30}$$

However, those expressions are not valid in singularities, as the matrices $\mathbf{A}$ and $\mathbf{J}_d$ are singular. To compute those expressions in singularity, it is necessary to extend the second-order loop-closure equations.

3.1.1 Type 2 singularity

When crossing a Type 2 singularity the inversion of the matrix $\mathbf{A}$ required to solve equation (26) is not possible anymore. In a Type 2 singularity the appearance of one uncontrolled degree of freedom on the platform is related to a kinematic constraint on the joint coordinates. This constraint is respected if and only if $-\mathbf{B}\ddot{\mathbf{q}}_a - \mathbf{b}$ is included in the range of $\mathbf{A}$. Let us consider this constraint respected, then equation (26) will give an infinite number of solutions

$$\ddot{\mathbf{x}} = -\mathbf{A}^{+}(\mathbf{B}\ddot{\mathbf{q}}_a + \mathbf{b}) + (\mathbf{I}_{n_a} - \mathbf{A}^{+}\mathbf{A})\boldsymbol{\nu} \tag{31}$$

with $\boldsymbol{\nu}$ a n_a-dimensional arbitrary vector. This is due to the fact that the platform coordinates acceleration vector $\ddot{\mathbf{x}}$ do not only depends on joint accelerations but also on free dynamics associated to the uncontrollable platform motion. The dynamic criterion expressed in Table 1

$$\dot{\mathbf{x}}_s^T\mathbf{w}_d = 0 \tag{32}$$

is not only a necessary criterion to be respected by a trajectory that crosses the singularity but also an expression of the free dynamics of the platform. This free dynamics can be developed as function of $\ddot{\mathbf{x}}$. Let us consider the Lagrangian of the robot as function of the active joint and platform coordinates and velocities $L_{ax} = f(\dot{\mathbf{q}}_a, \mathbf{q}_a, \dot{\mathbf{x}}, \mathbf{x})$. Then $\mathbf{w}_d$ is given by

$$\begin{aligned}\mathbf{w}_d &= \frac{\mathrm{d}}{\mathrm{dt}}\left(\frac{\partial L_{ax}}{\partial \dot{\mathbf{x}}}\right)^T - \left(\frac{\partial L_{ax}}{\partial \mathbf{x}}\right)^T \\ &= \mathbf{M}_{ax}\ddot{\mathbf{x}} + \mathbf{c}_{ax}\end{aligned} \tag{33}$$

with $\mathbf{M}_{ax}$ a definite positive matrix $(n_a \times n_a)$. Equation (33) introduced in equation (32) gives

$$\dot{\mathbf{x}}_s^T\mathbf{M}_{ax}\ddot{\mathbf{x}} + \dot{\mathbf{x}}_s^T\mathbf{c}_{ax} = 0 \tag{34}$$

The second-order loop-closure equation (26) can be completed with equation (34) giving

$$\mathbf{A}_e\ddot{\mathbf{x}} = -\mathbf{B}_e\ddot{\mathbf{q}}_a - \mathbf{b}_e \tag{35}$$

with

$$\mathbf{A}_e = \begin{bmatrix}\mathbf{A} \\ \dot{\mathbf{x}}_s^T\mathbf{M}_{ax}\end{bmatrix} \qquad \mathbf{B}_e = \begin{bmatrix}\mathbf{B} \\ \mathbf{0}_{1\times n_{dof}}\end{bmatrix} \qquad \mathbf{b}_e = \begin{bmatrix}\mathbf{b} \\ \dot{\mathbf{x}}_s^T\mathbf{c}_{ax}\end{bmatrix}$$

$\mathbf{A}_e$ is a $(n_{dof}+1 \times n_{dof})$ matrix. Even at singularity, matrix $\mathbf{A}_e$ rank remains equal to n_{dof} (see proof below). If a solution to this system exists, it is unique and given by the expression

$$\ddot{\mathbf{x}} = -\mathbf{A}_e^{+}(\mathbf{B}_e\ddot{\mathbf{q}}_a + \mathbf{b}_e) \tag{36}$$

The existence of the solution can be guaranteed for a trajectory designed in Cartesian space and then mapped to joint space. However, when using this expression in the controller design, the existence of an exact solution is not guaranteed. The computation of the solution through the Moore-Penrose pseudo-inverse will ensure the computation of a solution minimizing the quadratic error in $\ddot{\mathbf{x}}$, thus leading to a good estimation of the inverse dynamic model.

Proof. The solutions to equation (26) expressed in equation (31) span an affine subspace $\mathcal{S}$ of the platform accelerations of direction $\ker\mathbf{A}$ (considered dimension 1). We will prove that the affine subspace spanned by the solutions in $\ddot{\mathbf{x}}$ to the dynamic criterion equation (34) intersect in one unique point with $\mathcal{S}$. The solutions to equation (34) is an affine subspace of dimension $n_{dof} - 1$. The direction of this affine subspace is given by

$$\dot{\mathbf{x}}_s^T\mathbf{M}_{ax}\ddot{\mathbf{x}} = 0 \tag{37}$$

As $\mathbf{M}_{ax}$ is a positive definite matrices, $\dot{\mathbf{x}}_s^T\mathbf{M}_{ax}\dot{\mathbf{x}}_s > 0$. Thus, any $\dot{\mathbf{x}}_s$ in $\ker\mathbf{A}$ is not solution to the equation (37), meaning that $\ker\mathbf{A}$ is not included in the direction given by (37). Then, the affine subspaces spanned by the solutions in $\dot{\mathbf{x}}$ to the equations (26) and (34) are two supplementary subspaces. Their intersection is a point in the platform accelerations space. This point is the unique possible platform acceleration that can be performed by the robot in the singularity, completing the proof.

The solution to the extended loop-closure equations (35) gives an exact computation of the platform acceleration if and only if the dynamic criterion is respected by the platform acceleration vector. However, in the design of our trajectories, the criterion is only respected in the neighborhood of a singularity. Away from it, it will be necessary to use the classical loop-closure equations (26). For the purpose of designing a controller in experimental conditions, the transition between those two equations have to be continuous in

the neighborhood of the singularity. A Type 2 singularity is characterized by the degeneracy of the matrix $\mathbf{A}$. The matrix condition number $\kappa(\mathbf{A}) = \frac{e_{\max}(\mathbf{A})}{e_{\min}(\mathbf{A})}$ is chosen as an indicator of the proximity to the singularity, with $e_{\max}()$ and $e_{\min}()$ being respectively the maximal and minimal singular values of a matrix, leading to

$$\begin{aligned}\ddot{\mathbf{x}} = &-(\sigma\mathbf{A}^{-1}\mathbf{B} + (1-\sigma)\mathbf{A}_e^{+}\mathbf{B}_e)\ddot{\mathbf{q}}_a \\ &- \sigma\mathbf{A}^{-1}\mathbf{b} - (1-\sigma)\mathbf{A}_e^{+}\mathbf{b}_e \end{aligned} \tag{38}$$

with σ a continuous transition function equal to 0 in the neighborhood of a singularity and to 1 away from it.

$$\sigma = \begin{cases} 1 \text{ if } \dfrac{1}{\kappa(\mathbf{A})} > 2\varepsilon, \\ \dfrac{1-\kappa(\mathbf{A})\varepsilon}{\kappa(\mathbf{A})\varepsilon} \text{ if } \varepsilon < \dfrac{1}{\kappa(\mathbf{A})} < 2\varepsilon, \\ 0 \text{ if } \dfrac{1}{\kappa(\mathbf{A})} < \varepsilon. \end{cases} \tag{39}$$

with ε a threshold characterizing the proximity of the singularity determined experimentally.

3.1.2 LPJTS singularity

A similar result can be obtained for a LPJTS singularity. The equation (27) cannot be solved at singularity locus. In this case, the extended second-order loop-closure equation will take the form

$$\mathbf{J}_{d,e}\ddot{\mathbf{q}}_d = -\mathbf{J}_{da,e}\ddot{\mathbf{q}}_a - \mathbf{n}_e \tag{40}$$

with

$$\mathbf{J}_{d,e} = \begin{bmatrix} \mathbf{J}_d \\ \dot{\mathbf{q}}_{ds}^T\mathbf{M}_{de} \end{bmatrix} \qquad \mathbf{J}_{da,e} = \begin{bmatrix} \mathbf{J}_{da} \\ \mathbf{0}_{1\times n_{dof}} \end{bmatrix} \qquad \mathbf{n}_e = \begin{bmatrix} \mathbf{n} \\ \dot{\mathbf{q}}_{ds}^T\mathbf{c}_{de} \end{bmatrix}$$
$$\mathbf{M}_{de} = \mathbf{M}_d \begin{bmatrix} -\mathbf{J}_{da}^{-1}\mathbf{J}_d \\ \mathbf{1}_{n_p\times 1} \end{bmatrix} \qquad \mathbf{c}_{de} = \mathbf{c}_d + \mathbf{M}_d \begin{bmatrix} -\mathbf{J}_{da}^{-1}\mathbf{n} \\ \mathbf{0}_{n_p\times 1} \end{bmatrix}$$

With a reasoning similar to the one applied for Type 2 singularities (section (3.1.1)), it can be proved that the system (40) will give a unique solution in LPJTS singularity. The computation of the passive joint acceleration is then a combination of the classical loop-closure equations (30) and the extended loop-closure equation (40),

$$\begin{aligned}\ddot{\mathbf{q}}_d = &-(\sigma_p\mathbf{J}_d^{-1}\mathbf{J}_{da} - (1-\sigma_p)\mathbf{J}_{d,e}^{+}\mathbf{J}_{da,e})\ddot{\mathbf{q}}_a \\ &- \sigma_p\mathbf{J}_d^{-1}\mathbf{n} - (1-\sigma_p)\mathbf{J}_{td,e}^{+}\mathbf{n}_e \end{aligned} \tag{41}$$

where

$$\sigma_p = \begin{cases} 1 \text{ if } \dfrac{1}{\kappa(\mathbf{J}_d)} > 2\varepsilon_p, \\ \dfrac{1-\kappa(\mathbf{J}_d)\varepsilon_p}{\kappa(\mathbf{J}_d)\varepsilon_p} \text{ if } \varepsilon < \dfrac{1}{\kappa(\mathbf{J}_d)} < 2\varepsilon_p, \\ 0 \text{ if } \dfrac{1}{\kappa(\mathbf{J}_d)} < \varepsilon_p. \end{cases} \tag{42}$$

with ε_p a threshold characterizing the proximity of the singularity.

3.2 Expression of the inverse dynamic model for a control in joint space

In section 2.3.1, general solutions to the inverse dynamic model that are valid in and out of a singularity are given by equation (18) for Type 2 singularities and (24) for LPJTS singularities. Those models will be used for the design of a the computed torque control law to cross the singularity. The inverse dynamic model is expressed as function of the vectors $\boldsymbol{\tau}_a$, $\boldsymbol{\tau}_d$ and $\mathbf{w}_x$ issued from the Lagrangian of the system by equations (3)-(5). For encoder-based computed torque control purpose, this inverse dynamic model must be expressed as a function of the acceleration of the measured coordinates $\ddot{\mathbf{q}}_a$. This expression will be computed using equations (9)-(11) and the extended second-order loop-closure equations expressed in section 3.1.

3.2.1 Type 2 singularity

Combining the second order kinematic equations (27) and (38) with the equations issued from the Lagrangian (11)-(10) and the inverse dynamic model (18) leads to the expression of the model as function of the joint acceleration $\ddot{\mathbf{q}}_a$. The model can be expressed under the form

$$\boldsymbol{\tau} = \mathbf{M}_{T2}\ddot{\mathbf{q}}_a + \mathbf{h}_{T2} \tag{43}$$

where

- $\mathbf{M}_{T2}$ is the $(n_{dof} \times n_{dof})$ inertia matrix of the system. $\mathbf{M}_{T2}$ depends on the joints coordinates $\mathbf{q}$ and platform coordinates $\mathbf{x}$. Skipping all mathematical derivations, it comes

$$\begin{aligned}\mathbf{M}_{T2} = &\mathbf{J}_{T2}^{T\,+}\big(\mathbf{J}_{T2}^T\mathbf{M}_a\mathbf{J}_\sigma \\ &+ \mathbf{M}_x(\sigma\mathbf{A}^{-1}\mathbf{B} + (1-\sigma)\mathbf{A}_e^{+}\mathbf{B}_e) \\ &-(\mathbf{J}_x^T + \mathbf{J}_{T2}^T\mathbf{J}_a^T)\mathbf{J}_d^{-T}\mathbf{J}_\sigma\big) \end{aligned} \tag{44}$$

  with $\mathbf{J}_\sigma = \begin{bmatrix} \mathbf{I}_{n_{dof}} \\ -\mathbf{J}_d^{-1}(\mathbf{J}_x(\sigma\mathbf{A}^{-1}\mathbf{B} + (1-\sigma)\mathbf{A}_e^{+}\mathbf{B}_\mathbf{e}) + \mathbf{J}_a) \end{bmatrix}$.
- $\mathbf{h}_{T2}$ is a n_{dof}-dimensional vector regrouping the gravitational, centrifugal and Coriolis terms. $\mathbf{h}_{T2}$ depends on the joints coordinates $\mathbf{q}$ and velocities $\dot{\mathbf{q}}$ and platform

coordinates $\mathbf{x}$ and their time derivative $\dot{\mathbf{x}}$. Skipping all mathematical derivations, one can obtain

$$\begin{aligned}\mathbf{h}_{T2} = \mathbf{J}_{T2}^{T}{}^{+} \big(&\mathbf{J}_{T2}^{T}(\mathbf{M}_a\mathbf{n}_\sigma + \mathbf{c}_a) \\ &+ \mathbf{M}_x(\sigma\mathbf{A}^{-1}\mathbf{b} + (1-\sigma)\mathbf{A}_e^{+}\mathbf{b}_e) - \mathbf{c}_x \\ &-(\mathbf{J}_x^T + \mathbf{J}_{T2}^T\mathbf{J}_a^T)\mathbf{J}_d^{-T}(\mathbf{M}_d\mathbf{n}_\sigma - \mathbf{c}_d)\big) \end{aligned} \tag{45}$$

with $\mathbf{n}_\sigma = \begin{bmatrix} \mathbf{0}_{n_{dof}\times 1} \\ -\mathbf{J}_d^{-1}\mathbf{J}_x(\sigma\mathbf{A}^{-1}\mathbf{b} + (1-\sigma)\mathbf{A}_e^{+}\mathbf{b}_e) + \mathbf{d} \end{bmatrix}$.

3.2.2 LPJTS singularity

An expression similar to equation (43) can be derived from the inverse dynamic model obtained for a LPJTS singularity. Combining the second order kinematic equations (29) and (41) with the equations issued from the Lagrangian (11)-(10) and the inverse dynamic model equation (24) leads to the expression of the model as function of the joint acceleration $\ddot{\mathbf{q}}_a$.

$$\boldsymbol{\tau} = \mathbf{M}_p\ddot{\mathbf{q}}_a + \mathbf{h}_p \tag{46}$$

Skipping the mathematical derivations, $\mathbf{M}_p$ is given by

$$\begin{aligned}\mathbf{M}_p = \mathbf{J}_p^{T}{}^{+} \big(&-\mathbf{M}_d\mathbf{J}_{\sigma_p} \\ &+\mathbf{J}_p^T\left(\mathbf{M}_a\mathbf{J}_{\sigma_p} + \mathbf{B}^T\mathbf{A}^{-T}\mathbf{M}_x\mathbf{A}^{-1}\mathbf{B}\right)\big) \end{aligned} \tag{47}$$

with $\mathbf{J}_{\sigma p} = \begin{bmatrix} \mathbf{I}_{n_{dof}} \\ -\sigma_p\mathbf{J}_d^{-1}\mathbf{J}_{da} - (1-\sigma_p)\mathbf{J}_{td,e}^{+}\mathbf{J}_e \end{bmatrix}$
and $\mathbf{h}_p$ is given by

$$\begin{aligned}\mathbf{h}_p = \mathbf{J}_p^{T}{}^{+} \big(&-\left(\mathbf{M}_d\mathbf{n}_{\sigma_p} + \mathbf{c}_d\right) + \mathbf{J}_p^T\left(\mathbf{M}_a\mathbf{n}_{\sigma_p} + \mathbf{c}_a\right. \\ &\left.+\mathbf{B}^T\mathbf{A}^{-T}(\mathbf{M}_x\mathbf{A}^{-1}\mathbf{b} - \mathbf{c}_r)\right)\big) \end{aligned} \tag{48}$$

with $\mathbf{n}_{\sigma p} = \begin{bmatrix} \mathbf{0}_{n_{dof}\times 1} \\ -\sigma_p\mathbf{J}_d^{-1}\mathbf{n} - (1-\sigma_p)\mathbf{J}_{td,e}^{+}\mathbf{n}_e \end{bmatrix}$

3.3 Wrench projection in the neighborhood of a singularity

Around the singularity locus, the dynamic criterion established (Table 1) will not be exactly respected by a controller in real conditions. Unfortunately, only the strict respect of this criterion guarantee an exact solution to the inverse dynamic model. The existence of numerical and tracking errors may lead to the computation of huge torques in the neighborhood of a singularity. Error amplification around the singularity is due to the presence of a singular value near zero in the inverted matrix. Based on the assumption that the criterion is respected in the neighborhood of the singularity, a projection of the desired wrench in the dynamic model is performed. The design of the error inhibitor is given in this section for a Type 2 singularity from equation (18). The design of a similar inhibitor for a LPJTS singularity can be obtained from equation (24).

Let us consider the inverse dynamic model equation to solve under the form (18)

$$\mathbf{A}^T\mathbf{B}^{-T}\boldsymbol{\tau} = \mathbf{w}_{T2} \tag{49}$$

The matrix $\mathbf{A}$ degenerates in a Type 2 singularity. Let consider its singular value decomposition

$$\mathbf{A} = \mathbf{U}\Sigma\mathbf{V}^T \tag{50}$$

where Σ is the diagonal matrix of ordered positive singular values $(e_1,\ldots,e_n)$ and $\mathbf{U}$, $\mathbf{V}$ two unitary matrices. At singularity locus, the last singular value is null $e_n = 0$. Multiplying equation (49) by the matrix $\mathbf{V}^T$ gives

$$\Sigma\mathbf{U}^T\mathbf{B}^{-T}\boldsymbol{\tau} = \mathbf{V}^T\mathbf{w}_{T2} \tag{51}$$

In order for the dynamic model to give an exact solution, the last component of the vector $\mathbf{w}_s = \mathbf{V}^T\mathbf{w}_{T2}$ must be equal to zero. This is an other formulation of the dynamic criterion necessary to cross the singularity (note that the last line of the matrix $\mathbf{V}^T$ is included in $\ker\mathbf{A}$). As the dynamic criterion is not exactly respected, the last component of the vector $\mathbf{w}_s$ has to be driven toward zero in the neighborhood of the singularity to avoid the numerical degeneracy of the inverse dynamic model. As in section 3.1.1, the proximity to the singularity is characterized by the condition number using the norm-2 of the matrix $\mathbf{A}$. A function σ_s is defined by

$$\sigma_s = \begin{cases} 1 \text{ if } \dfrac{1}{\kappa(\mathbf{A})} > 2\varepsilon_s, \\ \dfrac{1-\kappa(\mathbf{A})\varepsilon_s}{\kappa(\mathbf{A})\varepsilon_s} \text{ if } \varepsilon_s < \dfrac{1}{\kappa(\mathbf{A})} < 2\varepsilon_s, \\ 0 \text{ if } \dfrac{1}{\kappa(\mathbf{A})} < \varepsilon_s. \end{cases} \tag{52}$$

with ε_s threshold determined experimentally. σ_s is a function equals to 1 and continuously driven towards 0 as the system get closer to the singularity. Multiplying the last component of the vector $\mathbf{w}_s$ by σ_s ensures that the desired wrench in equation (49) respects the dynamic criterion. This will avoid exponential increase of the torques consequent to minimal errors on the trajectory tracking.

3.4 Computed torque control law

The computed torque control (CTC) is an advanced control law for non-linear systems where the inverse dynamic model is used to realize an exact linearization of the system. Computed torque control allows the design of considerably more precise, energy-efficient and compliant controls for robots. The inclusion of the mechanism dynamic model

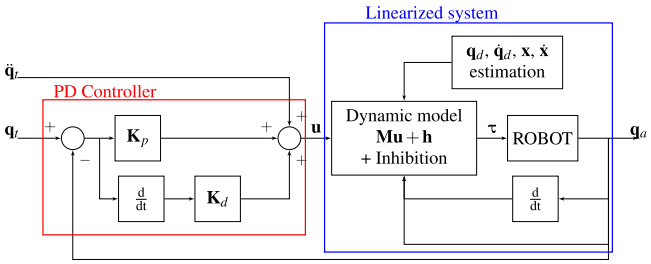


Figure 2: Computed Torque controller in joint space

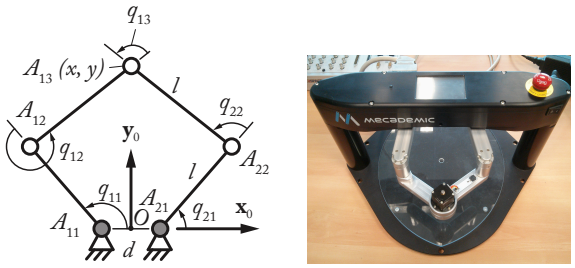


Figure 3: Five-bar mechanism and parametrization scheme

into the controller is desirable to enforce a dynamical criterion such as the criterion required for singularity crossing (Table 1). The convergence and the stability of such controllers is well known [22]. For all aforementioned reasons, this control law is an excellent candidate to track trajectories crossing the singularity. The following developments are suited both for Type 2 and LPJTS singularities.

From the previous section results, the inverse dynamic model of the robot can be expressed as a function of the active joint accelerations even at singular configuration $\ddot{\mathbf{q}}_a$:

$$\boldsymbol{\tau} = \mathbf{M}\ddot{\mathbf{q}}_a + \mathbf{h} \tag{53}$$

The computed torque control law is obtained with the design of an auxiliary control signal $\mathbf{u}$ corresponding to the joint acceleration $\ddot{\mathbf{q}}_a$ from which the input torques are computed

$$\boldsymbol{\tau} = \mathbf{Mu} + \mathbf{h} \tag{54}$$

The following PD control law is applied

$$\mathbf{u} = \ddot{\mathbf{q}}_t + \mathbf{K}_d(\dot{\mathbf{q}}_t - \dot{\mathbf{q}}_a) + \mathbf{K}_p(\mathbf{q}_t - \mathbf{q}_a) \tag{55}$$

where

- $\mathbf{q}_t$, $\dot{\mathbf{q}}_t$ and $\ddot{\mathbf{q}}_t$ are respectively the desired active joints position, velocity and acceleration along the tracked trajectory,
- $\mathbf{K}_d$ and $\mathbf{K}_p$ are positive definite gain matrices.

This control law ensures a second-order convergence of the joint position error $\mathbf{e} = \mathbf{q}_t - \mathbf{q}_a$. Fig. 2 represents the classical CTC controller that can be applied to a parallel mechanism in order to track a trajectory. The computation of the matrices $\mathbf{M}$ and $\mathbf{h}$ requires an estimation of the platform and passive joints position and velocity. Several techniques may be used to get this estimation such as a computation through the geometric model, interval analysis method [23] or exteroceptive measures [24]. The estimation may also be computed from the desired trajectory, as long as the tracking is reliable [19]. A computed torque control in Cartesian coordinates may also be designed to limit the issues with solutions to the second-order kinematic equations for Type 2 singularities (see section 3.1.1). However, to be efficient, such controller requires a precise external measure of the end-effector position, which is currently rarely available on robots. For a review of dynamic control of parallel robots, the reader is referred to [25].

4 Case study

The proposed approach was validated on an experimental platform. As the models and controllers are similar for Type 2 and LPJTS singularities, the experimental validation is only presented for the crossing of Type 2 singularities with a planar five-bar mechanism.

A five-bar mechanism is a planar parallel mechanism composed of two actuated joints located points A_{11} and A_{21} and three passive joints located at points A_{12}, A_{22} and A_{13} (see Fig. 3). This mechanism is able to generate a motion of the end-effector situated in A_{13} ($\mathbf{x} = (x,y)^T$) through the actuation of the joints q_{11} and q_{21}. The passive joints coordinates are respectively noted q_{12} on the first arm, q_{22} on the second arm and q_{13} for the joint linking those two arms.

4.1 Identified dynamic model

A full dynamic model identification was computed using a weighted least square method applied on exciting trajectories described in [26, 27]. The base parameters identified for the five-bar mechanism were

- grouped inertia for the two proximal links $zz_{11R} = 0.0133$ kg/m^2 and $zz_{21R} = 0.0142$ kg/m^2,
- grouped point mass at the end-effector $m_{3R} = 0.537$ kg,
- frictions (coulomb model) static $f_{off11} = 0.0855$ Nm, $f_{off21} = 0$ Nm and kinetic $f_{s11} = 0.345$ Nm/s, $f_{s21} = 0.430$ Nm/s parameters for the active joints.

The base parameters lead to the computation of the dynamic model (6)-(8) under the form

$$\begin{cases} \boldsymbol{\tau} = \mathbf{Z}\ddot{\mathbf{q}}_a + \mathbf{f} - \mathbf{B}^T\boldsymbol{\lambda}_1 \\ \mathbf{A}^T\boldsymbol{\lambda}_1 = m_{3R}\ddot{\mathbf{x}} \end{cases} \tag{56}$$

with

- $\ddot{\mathbf{q}}_a = (\ddot{q}_{11}, \ddot{q}_{21})^T$ the active joint accelerations vector,
- $\mathbf{A}$ and $\mathbf{B}$ issued from the loop-closure equations (1) and (2),

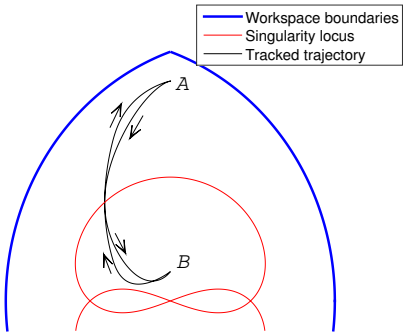


Figure 4: Tracked trajectory in Cartesian space (to scale)

- $\mathbf{Z}$ a positive diagonal matrix

$$\mathbf{Z} = \begin{bmatrix} zz_{11R} & 0 \\ 0 & zz_{21R} \end{bmatrix} \tag{57}$$

- $\mathbf{f}$ the active joint friction terms

$$\begin{bmatrix} f_{s11}\text{sign}(\dot{q}_{11}) \\ f_{s21}\text{sign}(\dot{q}_{21}) \end{bmatrix} + \begin{bmatrix} f_{off11} \\ f_{off21} \end{bmatrix} \tag{58}$$

  with sign a mathematical function extracting the sign of a real number.

The five-bar mechanism presents Type 2 singularities corresponding to the degeneracy of the matrix $\mathbf{A}$. Type 2 singularity configurations happen when the two distal links of the five-bar mechanism are aligned. In this case, the uncontrolled motion is orthogonal to this alignment (see Fig. 1). The dynamic criterion to be applied when crossing a Type 2 singularity (Table 1), related to the dynamic model (56), is then given by the equation

$$\ddot{y} = \ddot{x}\tan(q_{i1,s} + q_{i2,s}) \tag{59}$$

with $q_{i1,s}$ and $q_{i2,s}$ joint $i1$ and $i2$ positions of arm i in singular configuration. The expression of the inverse dynamic model that does not degenerate at singularity locus (18), as long as the criterion (59) is respected, is given by

$$\boldsymbol{\tau} = (\mathbf{J}_{T2}^T)^+(\mathbf{J}_{T2}^T\mathbf{Z}\ddot{\mathbf{q}}_a - m_{3R}\ddot{\mathbf{x}}) \tag{60}$$

4.2 Trajectory generation

To test the controller a trajectory crossing twice a Type 2 singularity has been generated with the five-bar mechanism. This trajectory (shown in Fig. 4) is a return trip from a point $A = [0;0.15]$ m to a point $B = [0;0.02]$ m. The trajectory crosses twice the singularity point $S = [-0.048;0.067]$ m. A point to point trajectory is computed in Cartesian coordinates and composed of

- A trajectory during a time interval $[t_{si};t_{so}]$ around the singularity with constant acceleration $(\ddot{x}_s,\ddot{y}_s)$ that respects the criterion (59), given by (with t the time variable)

$$\begin{aligned} x &= \frac{\ddot{x}_s}{2}t^2 + \alpha_x t + \beta_x \\ y &= \frac{\ddot{y}_s}{2}t^2 + \alpha_y t + \beta_y \end{aligned} \tag{61}$$

  α_x, α_y, β_x and β_y are coefficients computed such that the trajectory respect a desired position and velocity at singularity point.
- Two five degree polynomials ensuring a continuous transition form the starting point to the singularity trajectory and from the singularity trajectory to the end point. Those polynomials are defined by initial and final position, a null velocity and null acceleration of the end-effector at those points and by the the trajectory duration t_f.

Table 2 gives the parameters used to define the trajectory from point A to point B and from point B to point A. The trajectory in joint space is computed through the inverse geometric model of the robot.

Table 2: Trajectory parameters

	t_f [sec]	t_s [sec]	t_{si} [sec]	t_{so} [sec]	$\dot{x}_s$ [m/s]	$\dot{y}_s$ [m/s]	$\ddot{x}_s$ [m/s^2]	$\ddot{y}_s$ [m/s^2]
$A \to B$	0.7	0.35	0.25	0.45	0	-0.374	2	0.469
$B \to A$	1.4	1.15	1.05	1.25	0	-0.374	1	0.234

4.3 Experimental results

This section presents the experimental results of the trajectory defined in section 4.2. Figure 5 shows the torque computed by the controller and the joint tracking error during the trajectory. The thresholds ε and ε_s are both set to 0.1 for this experiment. With the dynamic model computed, there is no discontinuity in torque computation when crossing the singularity. The tracking error may locally increase just after the singularity crossing. Several factors generate this effect. First, the dynamic model around the singularity is an approximation which considers that the criterion established (59) is strictly respected. In real conditions, a strict respect of the criterion cannot be ensured, leading to a less efficient estimation of the dynamic model around the singularity and a deterioration of the tracking quality. It must also be considered that locally, a Type 2 singularity constraints the gap between the two proximal legs of the system, if the calibration of the proximal joint positions is not of perfect accuracy, an error will be generated by this constraint when crossing. And finally, it must be considered that in the neighborhood of a singularity, the system is locally underactuated.

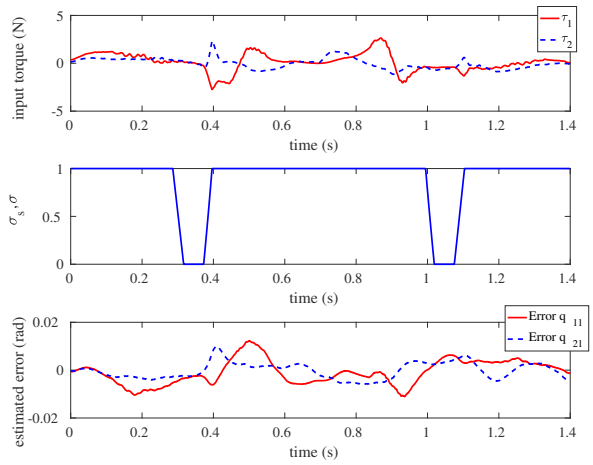


Figure 5: Input torque applied, σ and σ_s values and tracking error on a singularity crossing trajectory. $\sigma = \sigma_s = 0$ means that the robot is close to a singularity and the wrench projection in the controller is active.

This aspect makes a part of the tracking errors locally impossible to be corrected by the controller. The appropriate precautions taken make that the torque input remain bounded at singularity, at the cost of a local angular tracking error.

Nevertheless, a trajectory have been generated with several singularity points and accelerations. The trajectory have been tested at least 10 times with success in crossing the singularities each time. Experiment videos are available at https://www.youtube.com/watch?v=xVAaF4ycfho

5 Conclusion

The presence of singularities in the workspace of parallel robot reduces the effective workspace. To increase this workspace, a promising solution that consists in designing trajectories able to cross singularities has been developed in a previous study. However, the controllers developed to track those trajectories were limited to restrictive cases of Type 2 singularities crossing. This paper proposes new expressions of the inverse dynamic model, adapted for computation even in the singularity. The inverse dynamic model developed also handle the local overconstraint that appears in the singularity. Combined with the extension of the second order loop-closure equations in the singularity, it allows the design of a computed torque control able to track the desired trajectories. The theoretical contributions of this paper also extended the results to the crossing of LPJTS singularities. The controller was validated experimentally on a five-bar mechanism.

Future work on this subject includes the development of techniques to validate the operating mode change when crossing a singularity without the use of exteroceptive sensors. Also, one limitation of the controller is its dependency to a threshold determined experimentally. An improvement of the approach would be to develop a formal method to compute an appropriate value. Work related to a better consideration of the underactuation into the controller when crossing the singularity is also in progress.

References

[1] Merlet, J., 2006. *Parallel robots*. Springer- ISBN: 978-1-4020-4133-4.

[2] Zlatanov, D., Fenton, R., and Benhabib, B., 1994. "Singularity analysis of mechanisms and robots via a velocity-equation model of the instantaneous kinematics". In Proceedings of IEEE International Conference on Robotics and Automation, pp. 986–991.

[3] Zlatanov, D., Bonev, I. A., and Gosselin, C. M., 2002. "Constraint singularities of parallel mechanisms". In Proceedings of IEEE International Conference on Robotics and Automation, pp. 496–502.

[4] Conconi, M., and Carricato, M., 2009. "A New Assessment of Singularities of Parallel Kinematic Chains". *IEEE Transactions on Robotics,* **25**(4), aug, pp. 757–770.

[5] Briot, S., Pagis, G., Bouton, N., and Martinet, P., 2015. "Degeneracy conditions of the dynamic model of parallel robots". *Multibody System Dynamics*, oct, pp. 1–42.

[6] Gosselin, C., and Angeles, J., 1990. "Singularity analysis of closed-loop kinematic chains". *IEEE Transactions on Robotics and Automation,* **6**(3), jun, pp. 281–290.

[7] Liu, X.-J., Wang, J., and Pritschow, G., 2006. "Kinematics, singularity and workspace of planar 5R symmetrical parallel mechanisms". *Mechanism and Machine Theory,* **41**(2), feb, pp. 145–169.

[8] Gogu, G., 2004. "Structural synthesis of fully-isotropic parallel robots via theory of linear transformations and evolutionary morphology". *European Journal of Mechanics, A/Solids,* **23**, pp. 1021–1039.

[9] Briot, S., and Arakelian, V., 2010. "On the Dynamic Properties of Rigid-Link Flexible-Joint Parallel Manipulators in the Presence of Type 2 Singularities". *ASME. J. Mechanisms Robotics.,* **2**(2), pp. 021004, 6 pages.

[10] Nahon, M. A., and Angeles, J., 1989. "Force optimization in redundantly-actuated closed kinematic chains". In Proceedings of IEEE International Conference on Robotics and Automation, pp. 951–956.

[11] Kurtz, R., and Hayward, V., 1992. "Multiple-goal kinematic optimization of a parallel spherical mechanism with actuator redundancy". *IEEE Transactions on Robotics and Automation,* **8**(5), pp. 644–651.

[12] Rakotomanga, N., Chablat, D., and Caro, S., 2008. *Kinetostatic Performance of a Planar Parallel Mechanism with Variable Actuation*. Springer - ISBN: 978-1-402-04940-8.

[13] Arakelian, V., Briot, S., and Glazunov, V., 2008. "Increase of singularity-free zones in the workspace of parallel manipulators using mechanisms of variable structure". *Mechanism and Machine Theory,* **43**(9), sep, pp. 1129–1140.

[14] Campos, L., Bourbonnais, F., Bonev, I. A., and Bigras, P., 2010. "Development of a Five-Bar Parallel Robot

With Large Workspace". In Proceedings of Annual Mechanisms and Robotics Conference, pp. 917–922.
[15] Zein, M., Wenger, P., and Chablat, D., 2008. "Non-Singular Assembly-mode Changing Motions for 3-RPR Parallel Manipulators". *Mechanism and Machine Theory,* **43**(4), mar, pp. 480–490.
[16] Ider, S. K., 2005. "Inverse dynamics of parallel manipulators in the presence of drive singularities". *Mechanism and Machine Theory,* **40**(1), jan, pp. 33–44.
[17] Pagis, G., Bouton, N., Briot, S., and Martinet, P., 2015. "Enlarging parallel robot workspace through Type-2 singularity crossing". *Control Engineering Practice,* **39**, jun, pp. 1–11.
[18] Six, D., Briot, S., Chriette, A., and Martinet, P., 2016. "A Controller for Avoiding Dynamic Model Degeneracy of Parallel Robots During Type 2 Singularity Crossing". In New Trends in Mechanism and Machine Science (EUCOMES proc.), pp. 589–597.
[19] Khalil, W., and Dombre, E., 2002. *Modeling, Identification and Control of Robots*. Kogan Page Science.
[20] Briot, S., and Khalil, W., 2015. *Dynamics of Parallel Robots: From Rigid Bodies to Flexible Elements*. Springer - ISBN: 978-3-319-19787-6.
[21] Briot, S., and Arakelian, V., 2008. "Optimal Force Generation in Parallel Manipulators for Passing through the Singular Positions". *The International Journal of Robotics Research,* **27**(8), pp. 967–983.
[22] Samson, C., 1987. "Robust control of a class of nonlinear systems and applications to robotics". *International Journal of Adaptive Control and Signal Processing,* **1**, pp. 49–68.
[23] Merlet, J. P., 2004. "Solving the Forward Kinematics of a Gough-Type Parallel Manipulator with Interval Analysis". *The International Journal of Robotics Research,* **23**(3), mar, pp. 221–235.
[24] Özgür, E., Dahmouche, R., Andreff, N., and Martinet, P., 2014. "A vision-based generic dynamic model of PKMs and its experimental validation on the Quattro parallel robot". In Proceedings of IEEE/ASME International Conference on Advanced Intelligent Mechatronics, pp. 937–942.
[25] Paccot, F., Andreff, N., and Martinet, P., 2009. "A Review on the Dynamic Control of Parallel Kinematic Machines: Theory and Experiments". *The International Journal of Robotics Research,* **28**(3), pp. 395–416.
[26] Gautier, M., Khalil, W., and Restrepo, P. P., 1995. "Identification of the dynamic parameters of a closed loop robot". In Proceedings of IEEE International Conference on Robotics and Automation, pp. 3045–3050.
[27] Renaud, P., Vivas, A., Andreff, N., Poignet, P., Martinet, P., Pierrot, F., and Company, O., 2006. "Kinematic and dynamic identification of parallel mechanisms". *Control Engineering Practice,* **14**(9), sep, pp. 1099–1109.

List of Figures

1 Example of a five-bar mechanism in a Type 2 singularity configuration 3
2 Computed Torque controller in joint space . . 8
3 Five-bar mechanism and parametrization scheme 8
4 Tracked trajectory in Cartesian space (to scale) 9
5 Input torque applied, σ and σ_s values and tracking error on a singularity crossing trajectory 10

List of Tables

1 Dynamic criterion to avoid degeneracy of the inverse dynamic model in a singularity 3
2 Trajectory parameters 9

Annexe D

Démonstration pour la solution à l'équation cinématique d'ordre 2 en singularité

Dans la partie 5.2.2, il est annoncé que l'équation cinématique du second ordre (E5.15) complétée de l'équation de critère de traversée de singularité (E1.22) admet une solution unique. Nous développons la preuve de cette affirmation dans cette annexe. Cette preuve s'appuie sur les propriétés des sous-espaces affines solutions des équations.

Soit $\mathcal{S}_1$ le sous-espace affine de $\mathbb{R}^n$ solution de l'équation (E5.15). Nous supposons qu'il existe au moins une solution à cette équation. Dans ce cas, comme la matrice $\mathbf{J}_p$ est de dimension $n-1$ l'espace $\mathcal{S}_1$ est un espace affine de dimension 1 et de direction $\ker(\mathbf{J}_p)$.

Soit $\mathcal{S}_2$ le sous-espace affine $\mathbb{R}^n$ solution de l'équation (E1.22). $\mathcal{S}_2$ est de dimension $n-1$ et la direction de $\mathcal{S}_2$ est un hyperplan d'équation

$$\dot{\mathbf{q}}_{p,s}^T \mathbf{M}_t \ddot{\mathbf{q}}_p = 0 \tag{E4.1}$$

La matrice $\mathbf{M}_t$ est symétrique définie positive donc $\dot{\mathbf{q}}_{p,s}^T \mathbf{M}_t \dot{\mathbf{q}}_{p,s} > 0$. $\dot{\mathbf{q}}_{p,s}$ n'est donc pas inclus dans la direction de $\mathcal{S}_2$.

$\dot{\mathbf{q}}_{p,s}$ est inclus dans $\ker(\mathbf{J}_p)$ de dimension 1 on peut donc en déduire que $\ker(\mathbf{J}_p)$ n'est pas inclus dans l'hyperplan solution de (E4.1).

Ainsi les directions des sous-espaces affines $\mathcal{S}_1$ et $\mathcal{S}_2$ solutions des équations (E5.15) et (E1.22) sont supplémentaires dans $\mathbb{R}^n$, ils admettent une unique intersection que sera la solution des deux systèmes d'équation (E5.15) et (E1.22).

Bibliographie

L'essor des drones aériens civils en France. Rapport technique, Secrétariat général de la défense et de la sécurité nationale, 2016.

N. Abas, A. Legowo, et R. Akmeliawati. Parameter identification of an autonomous quadrotor. In *Proceedings of the 4th International Conference on Mechatronics (ICOM)*, Kuala Lumpur, Malaysia, may 2011.

J. Alonso-mora, E. Montijano, M. Schwager, et D. Rus. Distributed multi-robot Formation control among obstacles : A geometric and optimization approach with consensus. In *Proceedings of IEEE International Conference on Robotics and Automation (ICRA)*, pages 5356–5363, Stockholm, Sweden, may 2016.

G. Antonelli, F. Caccavale, et P. Chiacchio. A systematic procedure for the identification of dynamic parameters of robot manipulators. *Robotica*, 17(4) :427–435, 1999.

V. Arakelian, S. Briot, et V. Glazunov. Increase of singularity-free zones in the workspace of parallel manipulators using mechanisms of variable structure. *Mechanism and Machine Theory*, 43(9) :1129–1140, 2008.

S.B. Backus, L.U. Odhner, et A.M. Dollar. Design of hands for aerial manipulation : actuator number and routing for grasping and perching. In *Proceedings of the IEEE International Conference on Intelligent Robots and Systems (IROS)*, pages 34–40, Chicago, Illinois, USA, sep 2014.

S.R.S. Ball. *A treatise on the theory of screws*. Cambridge University Press, 1900.

S. Bellens, J. De Schutter, et H. Bruyninckx. A hybrid pose / wrench control framework for quadrotor helicopters. In *Proceedings of the IEEE International Conference on Robotics and Automation (ICRA)*, pages 2269–2274, Saint Paul, Minnesota, USA, apr 2012.

P. Ben-Horin et M. Shoham. Singularity condition of six-degree-of-freedom three-legged parallel robots based on Grassmann - Cayley Algebra. *IEEE Transactions on Robotics*, 22(4) :577–590, 2006.

L. Besnard, Y.B. Shtessel, et B. Landrum. Quadrotor vehicle control via sliding mode controller driven by sliding mode disturbance observer. *Journal of the Franklin Institute*, 349(2) :658–684, 2012.

S. Bouabdallah, P. Murrieri, et R. Siegwart. Design and control of an indoor micro quadrotor. In *Proceedings of the IEEE International Conference on Robotics and Automation (ICRA)*, pages 4393–4398, New Orleans, LA, USA, apr 2004a.

S. Bouabdallah, A. Noth, et R. Siegwart. PID vs LQ control techniques applied to an indoor micro quadrotor. In *Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, volume 3, pages 2451–2456, Sendai, Japan, oct 2004b.

S. Briot. *Advanced modelling of Robots.* Support du cours AMORO, Ecole Centrale de Nantes, 2014.

S. Briot et V. Arakelian. Optimal Force Generation in Parallel Manipulators for Passing through the Singular Positions. *The International Journal of Robotics Research*, 27(8) : 967–983, 2008.

S. Briot et M. Gautier. Global identification of joint drive gains and dynamic parameters of parallel robots. *Multibody System Dynamics*, 33(1) :3–26, 2015.

S. Briot et W. Khalil. *Dynamics of parallel robots : from rigid bodies to flexible elements.* Springer, 2015.

S. Briot, G. Pagis, N. Bouton, et P. Martinet. Degeneracy conditions of the dynamic model of parallel robots. *Multibody System Dynamics*, 37(4) :371–412, 2016.

P.J. Bristeau, P. Martin, E. Salaün, et N. Petit. The role of propeller aerodynamics in the model of a quadrotor UAV. In *Proceedings of the European Control Conference*, pages 683–688, Budapest, Hungary, aug 2009.

L. Campos, F. Bourbonnais, I.A. Bonev, et P. Bigras. Development of a five-bar parallel robot with large workspace. In *Proceedings of the Annual Mechanisms and Robotics Conference*, pages 917–922, Montreal, Quebec, Canada, aug 2010.

J. Carino, H. Abaunza, et P. Castillo. Quadrotor quaternion control. In *Proceeding of International Conference on Unmanned Aircraft Systems (ICUAS)*, pages 825–831, Denver, Colorado, USA, jun 2015.

S. Caro, W.A. Khan, D. Pasini, et J. Angeles. The rule-based conceptual design of the architecture of serial Schönflies-motion generators. *Mechanism and Machine Theory*, 45 (2) :251–260, 2010.

P. Castillo, R. Lozano, et A. Dzul. Stabilization of a mini rotorcraft with four rotors : Experimental implementation of linear and nonlinear control laws. *IEEE Control Systems Magazine*, 25(6) :45–55, 2005.

E. Cataldi, G. Muscio, M.A. Trujillo, Y. Rodriguez, F. Pierri, G. Antonelli, F. Caccavale, A. Viguria, S. Chiaverini, et A. Ollero. Impedance control of an aerial-manipulator : Preliminary results. In *Proceedings of the IEEE International Conference on Intelligent Robots and Systems (IROS)*, pages 3848–3853, Daejeon, Korea, nov 2016.

D. Chablat et P. Wenger. Architecture optimization of a 3-DOF translational parallel mechanism for machining applications, the orthoglide. *IEEE Transactions on Robotics and Automation*, 19(3) :403–410, 2003.

C. Chevallereau, H. Razavi, D. Six, Y. Aoustin, et J. Grizzle. Self-synchronization and self-stabilization of 3D bipedal walking gaits. *Robotics and Autonomous Systems*, 100 :43–60, 2018.

P. Chirarattananon, K.Y. Ma, et R.J. Wood. Perching with a robotic insect using adaptive tracking control and iterative learning control. *The International Journal of Robotics Research*, 35(10) :1185–1206, 2016.

A. Chriette, F. Plestan, H. Castañeda, M. Pal, M. Guillo, M. Oldega, S. Rajappa, et R. Chandra. Adaptive robust attitude control for UAVs - design and experimental validation. *International Journal of Adaptive Control and Signal Processing*, 30 :1478–1493, 2015.

R. Clavel. Device for the movement and positioning of an element in space, US Patent 4976582, 1980.

M. Conconi et M. Carricato. A new assessment of singularities of parallel kinematic chains. *IEEE Transactions on Robotics*, 25(4) :757–770, 2009.

I.D. Cowling, O.a. Yakimenko, J.F. Whidborne, et A.K. Cooke. A prototype of an autonomous controller for a quadrotor UAV. In *Proceedings of the European Control Conference*, pages 4001–4008, Kos, Grece, jul 2007.

T.W. Danko, K.P. Chaney, et P.Y. Oh. A parallel manipulator for mobile manipulating UAVs. In *Proceedings of the IEEE Conference on Technologies for Practical Robot Applications*, pages 1–6, Woburn, MA, USA, aug 2015.

G. Darivianakis, K. Alexis, M. Burri, et R. Siegwart. Hybrid predictive control for aerial robotic physical interaction towards inspection operations. In *Proceedings of the IEEE International Conference on Robotics and Automation (ICRA)*, pages 53–58, Hong Kong, China, jun 2014.

L. Derafa, T. Madani, et A. Benallegue. Dynamic modelling and experimental identification of four rotors helicopter parameters. In *Proceedings of the IEEE International Conference on Industrial Technology*, pages 1834–1839, Mumbai, India, jan 2006.

C. Diao, B. Xian, Q. Yin, W. Zeng, H. Li, et Y. Yang. A nonlinear adaptive control approach for quadrotor UAVs. In *8th Asian Control Conference (ASCC)*, pages 223–228, Kaohsiung, Taiwan, may 2011.

J. Diebel. Representing attitude : Euler angles, unit quaternions, and rotation vectors. Rapport technique, Standford University, 2006.

T. Dierks et S. Jagannathan. Output feedback control of a quadrotor UAV using neural networks. *IEEE Transactions on Neural Networks*, 21(1) :50–66, 2010.

K. Eliker, G. Zhang, S. Grouni, et W. Zhang. An optimization problem for quadcopter reference flight trajectory generation. *Journal of Advanced Transportation*, 2018 :1–15, jul 2018.

M. Elsamanty, A. Khalifa, M. Fanni, A. Ramadan, et A. Abo-Ismail. Methodology for identifying quadrotor parameters, attitude estimation and control. In *Proceedings of the IEEE/ASME International Conference on Advanced Intelligent Mechatronics (AIM)*, pages 1343–1348, Wollongong, Australia, jan 2013.

J.C. Erskine. Dynamics , planning and control of an aerial towed cable system, Thèse de master, Ecole Centrale de Nantes, 2018.

J.C. Erskine, A. Chriette, et S. Caro. Wrench capability analysis of aerial cable towed systems. In *Proceedings of the Computer and Information in Engineering Conference (IDETC/CIE)*, Québec, Canada, aug 2018.

M. Fumagalli, R. Naldi, A. Macchelli, F. Forte, A.Q. Keemink, S. Stramigioli, R. Carloni, et L. Marconi. Developing an aerial manipulator prototype. *Robotics and Automation*, 21 (3) :41–50, 2014.

M. Gautier et P. Poignet. Extended Kalman filtering and weighted least squares dynamic identification of robot. *Control Engineering Practice*, 9(12) :1361–1372, 2001.

V. Ghadiok, J. Goldin, et W. Ren. Autonomous indoor aerial gripping using a quadrotor. In *Proceedings of the IEEE International Conference on Intelligent Robots and Systems (IROS)*, pages 4645–4651, 2011.

G. Gogu. Structural synthesis of fully-isotropic parallel robots via theory of linear transformations and evolutionary morphology. *European Journal of Mechanics, A/Solids*, 23(6) : 1021–1039, 2004.

G. Gogu. *Structural synthesis of parallel robots*. Springer, 2008.

H. Goldstein. *Classical mechanics*. Addison-Wesley, 1980.

C. Gosselin. Compact dynamic models for the tripteron and quadrupteron parallel manipulators. *Proceedings of the Institution of Mechanical Engineers. Part I : Journal of Systems and Control Engineering*, 223(1) :1–11, 2009.

M.S. Grewal et A.P. Andrews. *Kalman filtering : theory and practice using MATLAB (Third Edition)*. Wiley, 2008.

M. Hehn et R. D'Andrea. Real-time trajectory generation for quadrocopters. *IEEE Transactions on Robotics*, 31(4) :877–892, aug 2015.

G. Heredia, A.E. Jimenez-Cano, I. Sanchez, D. Llorente, V. Vega, J. Braga, J.A. Acosta, et A. Ollero. Control of a multirotor outdoor aerial manipulator. In *Proceedings of the IEEE International Conference on Intelligent Robots and Systems (IROS)*, pages 3417–3422, Chicago, Illinois, USA, sep 2014.

J. Hesselbach, M. Helm, et S. Soetebier. Connecting assembly modes for workspace enlargement. In J. Lenarčič et F. ThomasIn , *Advances in Robot Kinematics (ARK)*, pages 347–356. Springer, 2002.

R.B. Hill, D. Six, A. Chriette, S. Briot, et P. Martinet. Crossing type 2 singularities of parallel robots without pre-planned trajectory with a virtual-constraint-based controller. In *Proceedings of the International Conference on Robotics and Automation (ICRA)*, pages 6080–6085, Singapore, may 2017.

J. Hollerbach, W. Khalil, et M. Gautier. Model Identification. In B. Siciliano et O. KhatibIn , *Springer Handbook of Robotics*, chapter 14. Springer, 2008.

M. Huang, B. Xian, C. Diao, K. Yang, et Y. Feng. Adaptive tracking control of underactuated quadrotor unmanned aerial vehicles via backstepping. In *Proceedings of American Control Conference*, pages 2076–2081, Baltimore, MD, USA, jul 2010.

F. Huber, K. Kondak, K. Krieger, D. Sommer, M. Schwarzbach, M. Laiacker, I. Kossyk, S. Parusel, S. Haddadin, et A. Albu-Schaffer. First analysis and experiments in aerial manipulation using fully actuated redundant robot arm. In *Proceedings of the IEEE International Conference on Intelligent Robots and Systems (IROS)*, pages 3452–3457, Tokyo, Japan, nov 2013.

K. Hunt. *Kinematic geometry of mechanisms*. Oxford University Press, 1978.

S.K. Ider. Inverse dynamics of parallel manipulators in the presence of drive singularities. *Mechanism and Machine Theory*, 40(1) :33–44, 2005.

H. Jayakrishnan. Position and attitude control of a quadrotor UAV using super twisting sliding mode. *IFAC-PapersOnLine*, 49(1) :284–289, 2016.

M. Kamel, K. Alexis, et R. Siegwart. Design and modeling of dexterous aerial manipulator. In *Proceedings of the IEEE International Conference on Intelligent Robots and Systems (IROS)*, pages 4870–4876, Daejeon, Korea, oct 2016.

F. Kendoul, I. Fantoni, et R. Lozano. Modeling and Control of a Small Autonomous Aircraft Having Two Tilting Rotors. *IEEE Transactions on Robotics*, 22(6) :1297–1302, 2006.

H.K. Khalil. *Nonlinear systems (Third edition)*. Pearson, 2002.

W. Khalil. Dynamic modeling of robots using recursive Newton-Euler techniques. In *Proceedings of the International Conference on Informatics in Control, Automation and Robotics*, Madeira, Portugal, jun 2010.

W. Khalil et D. Creusot. SYMORO + : A system for the symbolic modelling of robots. *Robotica*, 15(2) :153–161, 1997.

W. Khalil et E. Dombre. *Modeling, identification and control of robots*. Kogan Page Science, 2002.

W. Khalil et J. Kleinfinger. A new geometric notation for open and closed-loop robots. In *Proceedings of the IEEE International Conference on Robotics and Automation (ICRA)*, pages 1174–1179, San Francisco, CA, USA, may 1986.

P. Khosla et T. Kanade. Parameter identification of robot dynamics. In *Proceedings of the 24th IEEE Conference on Decision and Control*, pages 1754–1760, Ft. Lauderdale, Finland, dec 1985.

S. Kim, S. Choi, et H.J. Kim. Aerial manipulation using a quadrotor with a two DOF robotic arm. In *Proceedings of the IEEE International Conference on Intelligent Robots and Systems (IROS)*, pages 4990–4995, Tokyo, Japan, nov 2013.

S. Kim, S. Choi, H. Lee, et H.J. Kim. Vision-based collaborative lifting using quadrotor UAVs. In *Proceedings of the International Conference on Control, Automation and Systems*, pages 1169–1174, Seoul, Korea, oct 2014.

X. Kong et C. Gosselin. *Type synthesis of parallel mechanisms*. Springer, 2007.

K. Kozlowski. *Modeling and Identification in Robotics*. Springer, 1996.

V. Kumar. Instantaneous kinematics of parallel-chain robotic mechanisms. *Journal of Mechanical Design*, 114 :349–358, 1992.

R. Kurtz et V. Hayward. Multiple-goal kinematic optimization of a parallel spherical mechanism with actuator redundancy. *IEEE Transactions on Robotics and Automation*, 8 (5) :644–651, 1992.

D. Lee, T.C. Burg, B. Xian, et D.M. Dawson. Output feedback tracking control of an underactuated quad-rotor UAV. In *Proceedings of the American Control Conference*, pages 1775–1780, New York City, USA, jul 2007.

T. Leinonen. Terminology for the theory of machines and mechanisms. *Mechanism and Machine Theory*, 26, 1991.

A. Levant. Sliding order and sliding accuracy in sliding mode control. *International Journal of Control*, 58(6) :1247–1263, 1993.

J. Li et Y. Li. Dynamic analysis and PID control for a quadrotor. In *Proceedings of the International Conference on Mechatronics and Automation*, pages 573–578, Beijin, China, aug 2011.

Q. Lindsey, D. Mellinger, et V. Kumar. Construction of Cubic Structures with Quadrotor Teams. In *Proceedings of Robotics : Science and Systems (RSS)*, Los Angeles, California, USA, jun 2011.

X.J. Liu, J. Wang, et G. Pritschow. Kinematics, singularity and workspace of planar 5R symmetrical parallel mechanisms. *Mechanism and Machine Theory*, 41(2) :145–169, 2006.

Z. Lu, K.B. Shlrnoga, et A.A. Goldenberg. Experimental determination of dynamic parameters of robotic arms. *Journal of Robotic Systems*, 10(8) :1009–1029, 1993.

T. Madani et A. Benallegue. Backstepping control for a quadrotor helicopter. In *Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 3255–3260, Beijin, China, oct 2006.

S.O.H. Madgwick, A.J.L. Harrison, et R. Vaidyanathan. Estimation of IMU and MARG orientation using a gradient descent algorithm. In *Proceedings of the IEEE International Conference on Rehabilitation Robotics*, pages 1–7, Zurich, Switzerland, jun 2011. IEEE.

S. Madgwick. An efficient orientation filter for inertial and inertial/magnetic sensor arrays. Rapport technique, X-IO and University of Bristol, 2010.

R. Mahony, T. Hamel, J.M. Pflimlin, J.M. Pflimlin Nonlinear, et R. Mahony. Complementary Filters on the Special Orthogonal Group. *IEEE Transactions on Automatic Control, Institute of Electrical and Electronics Engineers*, 53(5) :1203–1217, 2008.

R. Mahony, V. Kumar, et P. Corke. Multirotor Aerial Vehicles : Modeling, estimation, and control of quadrotor. *IEEE Robotics and Automation Magazine*, 19(3) :20–32, 2012.

P. Martin et E. Salaün. The true role of accelerometer feedback in quadrotor control. In *Proceedings of the IEEE International Conference on Robotics and Automation (ICRA)*, pages 1623–1629, Anchorage, AK, USA, may 2010.

D. Mellinger. Minimum snap trajectory generation and control for quadrotors. In *Proceedings of the IEEE International Conference on Robotics and Automation (ICRA)*, pages 2520–2525, Shangai, China, may 2011.

J.P. Merlet. Singular configurations of parallel manipulators and Grassmann geometry. *The International Journal of Robotics Research*, 8(5) :45–56, 1989.

J.P. Merlet. *Parallel Robots*. Springer, 2006.

A.Y. Mersha, S. Stramigioli, et R. Carloni. Variable impedance control for aerial interaction. In *Proceedings of the IEEE International Conference on Intelligent Robots and Systems (IROS)*, pages 3435–3440, Chicago, Illinois, USA, sep 2014.

N. Michael, J. Fink, et V. Kumar. Cooperative manipulation and transportation with aerial robots. *Autonomous Robots*, 30(1) :73–86, 2011.

V. Mistler, A. Benallegue, et N. M'Sirdi. Exact linearization and noninteracting control of a 4 rotors helicopter via dynamic feedback. In *Proceedings of the 10th IEEE International Workshop on Robot and Human Interactive Communication*, pages 586–593, Bordeaux, France, sep 2001.

A. Mokhtari et A. Benallegue. Dynamic feedback controller of Euler angles and wind parameters estimation for a quadrotor unmanned aerial vehicle. In *Proceedings of IEEE International Conference on Robotics and Automation (ICRA)*, pages 2359–2366, New Orleans, LA, USA, apr 2004.

M.A. Nahon et J. Angeles. Force optimization in redundantly-actuated closed kinematic chains. In *Proceedings of IEEE International Conference on Robotics and Automation (ICRA)*, pages 951–956, Scottsdale,Arizona, USA, may 1989.

H.N. Nguyen et D. Lee. Hybrid force/motion control of quadrotor for tool operation. In *Proceedings of the International Conference on Intelligent Robots and Systems (IROS)*, pages 3458–3464, Tokyo, Japan, 2013.

H.N. Nguyen, S. Park, et D. Lee. Aerial tool operation system using quadrotors as Rotating Thrust Generators. In *Proceedings of the International Conference on Intelligent Robots and Systems (IROS)*, pages 1285–1291, Hamburg, Germany, sep 2015.

G. Pagis. *Augmentation de la taille de l'espace de travail opérationnel des robots parallèles en traversant les singularités de type 2*. PhD thesis, Ecole Centrale de Nantes, 2015.

G. Pagis, N. Bouton, S. Briot, et P. Martinet. Enlarging parallel robot workspace through Type-2 singularity crossing. *Control Engineering Practice*, 39 :1–11, jun 2015.

I. Palunko et R. Fierro. Adaptive control of a quadrotor with dynamic changes in the center of gravity. In *Proceedings of the IFAC World Congress*, pages 2626–2631, Milano, Italy, 2011.

C. Papachristos, K. Alexis, et A. Tzes. Efficient force exertion for aerial robotic manipulation : Exploiting the thrust-vectoring authority of a tri-tiltRotor UAV. In *Proceedings of the IEEE International Conference on Robotics and Automation (ICRA)*, pages 4500–4505, Hong Kong, China, jun 2014.

S. Paternain, M. Tailanian, R. Rosa, et R. Canetti. Design and implementation of flight control for an autonomous quadrotor. In *Proceedings of IEEE URUCON*, pages 1–4, Montevideo, Uruguay, oct 2017.

F. Pierrot, P. Dauchez, et A. Fournier. HEXA : a fast six-DOF fully-parallel robot. In *Proceedings of the Fifth International Conference on Advanced Robotics*, Pisa, Italy, jun 1991.

P.E.I. Pounds, D.R. Bersak, et A.M. Dollar. Grasping from the air : Hovering capture and load stability. In *Proceedings of the IEEE International Conference on Robotics and Automation (ICRA)*, pages 2491–2498, Shangai, China, may 2011.

J.A. Preiss, W. Hönig, G.S. Sukhatme, et N. Ayanian. Crazyswarm : A Large Nano-Quadcopter Swarm. In *Proceedings of the International Conference on Intelligent Robots and Systems (IROS)*, pages 3449–3450, Daejeon, Korea, oct 2016.

S. Rajappa, M. Ryll, H.H. Bulthoff, et A. Franchi. Modeling, control and design optimization for a fully-actuated hexarotor aerial vehicle with tilted propellers. In *Proceedings of the IEEE International Conference on Robotics and Automation (ICRA)*, pages 4006–4013, Seattle, WA, USA, may 2015.

N. Rakotomanga, D. Chablat, et S. Caro. Kinetostatic performance of a planar parallel mechanism with variable actuation. In J. Lenarčič et P. WengerIn , *Advances in Robot Kinematics : Analysis and Design*, pages 311–320. Springer, 2008.

C.R. Rao et H. Toutenburg. *Linear models : Least squares and alternatives, second edition.* Springer, 1999.

M.I. Rashid et S. Akhtar. Adaptive control of quadcopter with unknown model parameters. In *Proceedings of the 9th International Bhurban Conference on Applied Sciences & Technology (IBCAST)*, pages 8–14, Islamabad, Pakistan, jan 2012.

M. Renaud. Calcul de la matrice Jacobienne nécessaire à la commande coordonnée d'un manipulateur. *Mechanism and Machine Theory*, 15(2) :81–91, jan 1980.

H. Romero, S. Salazar, A. Sanchez, et R. Lozano. A new UAV configuration having eight rotors : Dynamical model and real-time control. In *Proceedings of the IEEE Conference on Decision and Control*, pages 6418–6423, New Orleans, LA, USA, dec 2007. IEEE.

M.H. Rosen, G. le Pivain, R. Sahai, N.T. Jafferis, et R.J. Wood. Development of a 3.2g untethered flapping-wing platform for flight energetics and control experiments. In *Proceeding of the IEEE International Conference on Robotics and Automation (ICRA)*, pages 3227–3233, Stockholm, Sweden, may 2016.

F. Ruggiero, M.A. Trujillo, R. Cano, H. Ascorbe, A. Viguria, C. Perez, V. Lippiello, A. Ollero, et B. Siciliano. A multilayer control for multirotor UAVs equipped with a servo robot arm. In *Proceedings of the IEEE International Conference on Robotics and Automation (ICRA)*, pages 4014–4020, Seattle, WA, USA, jun 2015.

M. Ryll, H.H. Bulthoff, et P.R. Giordano. First flight tests for a quadrotor UAV with tilting propellers. In *Proceedings of the IEEE International Conference on Robotics and Automation (ICRA)*, pages 295–302, Karlsruhe, Germany, may 2013.

M. Ryll, D. Bicego, et A. Franchi. A truly redundant aerial manipulator exploiting a multi-directional thrust base. In *Proceedings of the IFAC Symposium on Robot Control*, Budapest, Hungary, aug 2018.

A.L. Salih, M. Moghavvemi, H.A.F. Mohamed, et K.S. Gaeid. Flight PID controller design for a UAV quadrotor. *Scientific Research and Essays*, 5(23) :3660–3667, 2010.

C. Samson. Robust control of a class of non-linear systems and applications to robotics. *International Journal of Adaptive Control and Signal Processing*, 1 :49–68, 1987.

D.E. Schinstock. GPS-aided INS Solution for OpenPilot. Rapport technique, Kansas State University, 2010.

F. Sharifi, M. Mirzaei, G.W. Brandon, et Y. Zhang. Fault tolerant control of a quadrotor UAV using sliding mode control. In *Proceedings of the Conference on Control and Fault Tolerant Systems*, pages 239–244, Nice, France, aug 2010.

J.P. Silva, C. De Wagter, et G. de Croon. Quadrotor thrust vectoring control with time and jerk optimal trajectory planning in constant wind fields. *Unmanned Systems*, 6(1) :15–37, jan 2017.

D. Six, S. Briot, A. Chriette, et P. Martinet. A controller for avoiding dynamic model degeneracy of parallel robots during type 2 singularity crossing. In *New Trends in Mechanism and Machine Science (EUCOMES proceedings)*, pages 589–597, Nantes, France, sep 2016.

D. Six, S. Briot, A. Chriette, et P. Martinet. A controller avoiding dynamic model degeneracy of parallel robots during singularity crossing. *Journal of Mechanisms and Robotics*, 9(5) : 051008–051008–8, 2017a.

D. Six, A. Chriette, S. Briot, et P. Martinet. Dynamic modeling and trajectory tracking controller of a novel flying parallel robot. In *Proceedings of the IFAC World Congress*, pages 2241–2246, Toulouse, France, jul 2017b.

D. Six, S. Briot, A. Chriette, et P. Martinet. The kinematics , dynamics and control of a flying parallel robot with three quadrotors. *IEEE Robotics and Automation Letters*, 3(1) : 559–566, 2018.

J.J.E. Slotine et L. Weiping. Adaptive manipulator control : A case study. *IEEE Transactions on Automatic Control*, 33(11) :995–1003, 1988.

K. Sreenath et V. Kumar. Dynamics, control and planning for cooperative manipulation of payloads suspended by cables from multiple quadrotor robots. In *Proceedings of Robotics : Science and Systems*, Berlin, Germany, may 2013.

A. Suarez, P.R. Soria, G. Heredia, B.C. Arrue, et A. Ollero. Anthropomorphic, compliant and lightweight dual arm system for aerial manipulation. In *Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 992–997, Vancouver, BC, Canada, sep 2017.

M. Tognon et A. Franchi. Omnidirectional Aerial Vehicles With Unidirectional Thrusters : Theory, Optimal Design, and Control. *IEEE Robotics and Automation Letters*, 3(3) : 2277–2282, jul 2018.

M. Tognon, B. Yuksel, G. Buondonno, et A. Franchi. Dynamic decentralized control for protocentric aerial manipulators. In *Proceedings of the IEEE International Conference on Robotics and Automation (ICRA)*, pages 6375–6380, Singapore, may 2017.

S. Van Huffel et J. Vandewalle. *The total least squares problem : Computational aspects and analysis*. SIAM, 1991.

M. van Nieuwstadt et R.M. Murray. Real time trajectory generation for differentially flat systems. *IFAC Proceedings Volumes*, 29(1) :2301–2306, 1996.

G. Vásárhelyi, C. Virágh, G. Somorjai, N. Tarcai, T. Szörényi, T. Nepusz, et T. Vicsek. Outdoor flocking and formation flight with autonomous aerial robots. In *Proceedings of the IEEE International Conference on Intelligent Robots and Systems (IROS)*, pages 3866–3873, Chicago, Illinois, USA, sep 2014.

A.R. Vetrella, G. Fasano, A. Renga, et D. Accardo. Cooperative UAV navigation based on distributed multi-antenna GNSS, vision, and MEMS sensors. In *Proceedings of the International Conference on Unmanned Aircraft Systems (ICUAS)*, pages 1128–1137, Denver, Colorado, USA, jun 2015.

H. Voos. Nonlinear control of a quadrotor micro-uav using feedback-linearization. In *Proceedings of the IEEE International Conference on Mechatronics*, pages 1–6, Malaga, Spain, apr 2009.

J. Wang, T. Raffler, et F. Holzapfel. Nonlinear position control approaches for quadcopters using a novel state representation. In *Proceedings of the AIAA Guidance, Navigation, and Control Conference*, pages 1–17, Minneapolis, Minnesota, USA, aug 2012.

W. Wei et N. Simaan. Design of planar parallel robots with preloaded flexures for guaranteed backlash prevention. *Journal of Mechanisms and Robotics*, 2(1) :011012–1–9, 2010.

R. Xu et Ü. Özguner. Sliding Mode Control of a Quadrotor Helicopter. In *Proceedings of IEEE Conference on Decision and Control*, pages 4957–4962, San Diego, CA, USA, dec 2006.

B. Yu, X. Dong, Z. Shi, et Y. Zhong. Formation control for quadrotor swarm systems : Algorithms and experiments. In *Proceedings of the Chinese Control Conference*, pages 7099–7104, Xi'an, China, jul 2013.

M. Zein, P. Wenger, et D. Chablat. Non-singular assembly-mode changing motions for 3-RPR parallel manipulators. *Mechanism and Machine Theory*, 43(4) :480–490, mar 2008.

M. Zhao, T. Anzai, F. Shi, X. Chen, K. Okada, et M. Inaba. Design, modeling, and control of an aerial robot DRAGON : A dual-rotor-embedded multilink robot with the ability of multi-degree-of-freedom aerial transformation. *IEEE Robotics and Automation Letters*, 3 (2) :1176–1183, 2018.

D. Zlatanov, R.G. Fenton, et B. Benhabib. Singularity analysis of mechanisms and robots via a velocity-equation model of the instantaneous kinematics. In *Proceedings of the IEEE International Conference on Robotics and Automation (ICRA)*, pages 986–991, San Diego, CA, USA, jun 1994.

UNIVERSITE BRETAGNE LOIRE / MATHSTIC

CENTRALE NANTES

Titre : Conception et commande de robots parallèles volants

Mots clés : Véhicule aérien autonome ; Robots parallèles ; Modèle Dynamique ; Commande par retour d'état

Résumé :

La manipulation aérienne est l'un des défis de la robotique au cours de cette dernière décennie. L'un des freins au développement des manipulateurs aériens est l'autonomie limitée des drones, réduite par la charge et la consommation électrique du manipulateur embarqué. Une solution pour dépasser cette limite passe par la collaboration de plusieurs drones dans une tâche de manipulation.

Cette thèse porte sur la conception et la commande d'un nouveau type de véhicule autonome aérien destiné à la manipulation. Le concept consiste à faire collaborer plusieurs drones, en particulier des quadricoptères, au travers d'une architecture passive polyarticulée. Le robot ainsi obtenu est une fusion entre l'architecture passive d'un robot parallèle et plusieurs drones.

L'étude du modèle dynamique de cette classe de robots met en avant un découplage dans le modèle dynamique. Ce découplage permet la conception d'une commande en cascade qui assure la stabilisation et le suivi de trajectoire pour ces robots.

Deux cas d'étude sont ensuite déclinés dans cette thèse : un robot parallèle volant à deux drones et un robot parallèle volant à trois drones. Pour ces deux robots, une simulation numérique est effectuée afin de valider le fonctionnement de la commande. Ces simulations ont également permis de valider la possibilité de changer la configuration de l'architecture passive en vol.

Les travaux ont été portés avec succès jusqu'au stade expérimental pour le robot volant à deux drones.

Title : Design and control of flying parallel robots

Keywords : Unmanned Aerial Vehicle ; Parallel robots ; Dynamic model ; Feedback control

Abstract :

Aerial manipulation is one of the challenges of robotics over the last decade. One of the constraints on the development of aerial manipulators is the limited autonomy of drones, reduced by the load and energy consumption of the on-board manipulator. One way to overcome this limit is to have several drones collaborate on a manipulation task.

This thesis deals with the design and control of a new type of autonomous aerial vehicle for manipulation tasks. The concept consists in the collaboration of several drones, in particular quadrotors, through a polyarticulated passive architecture. The robot thus obtained is a fusion between the passive architecture of a parallel robot and several drones.

The study of the dynamic model of this robot class highlights a decoupling in the dynamic model. This decoupling allows the design of a cascade control law. This controller provides stabilization and trajectory tracking for these robots.

Two study cases are then presented in this thesis: a flying parallel robot with two drones and a flying parallel robot with three drones. For these two robots, a numerical simulation is performed to validate the controller performances. These simulations also allowed to validate the reconfiguration abilities of passive architecture in flight.

The work was successfully carried to the experimental stage for the flying robot with two drones.