



RÉPUBLIQUE ALGÉRIENNE DÉMOCRATIQUE ET POPULAIRE



MINISTÈRE DE L'ENSEIGNEMENT SUPÉRIEUR
ET DE LA RECHERCHE SCIENTIFIQUE
UNIVERSITÉ DES FRÈRES MENTOURI - CONSTANTINE 1

FACULTÉ DES SCIENCES DE LA TECHNOLOGIE
DÉPARTEMENT D'ELECTRONIQUE

N° Série :

N° d'ordre :

THÈSE

*Pour l'obtention du diplôme de Doctorat Troisième Cycle LMD
En Électronique*

Option: Contrôle et Traitement du Signal

Présentée par

BENZAID Karima

THÈME

**Stratégies de commande de véhicules à
déplacement vertical**

Soutenue le 26 Novembre 2018

devant le jury :

Président	BENNIA Abdelhak	Prof.	U. F. M. Constantine 1
Rapporteur	MANSOURI Noura	Prof.	U. F. M. Constantine 1
Co-rapporteur	LABBANI-IGBIDA Ouiddad	Prof.	U. de Limoges (Fr)
Examineurs	BOUDJEHEM Djalil	Prof.	Université de Guelma
	LADACI Samir	Prof.	É. N. P. de Constantine

Remerciements

Avant tout, je tiens à remercier les personnes qui ont contribué à la réalisation de ce travail de thèse. J'adresse mes premiers remerciements à mes directrices de thèse Mme. Noura Mansouri et Mme. Ouiddad Labbani-Igbida à qui je suis reconnaissante pour le temps qu'elles m'ont consacré, leurs conseils et leurs encouragements.

Je remercie sincèrement Mme. Noura Mansouri pour sa patience et sa rigueur dans mes moments de doute. Vous aviez toujours la bonne formule quand les efforts que je fournissais ne me semblaient pas porter fruit.

Je remercie profondément Mme. Ouiddad Labbani-Igbida de m'avoir accueillie chaleureusement durant mes séjours de recherche dans l'équipe de Robotique et Mécatronique de l'institut XLIM de Limoges. Je vous remercie pour votre sympathie et bonne humeur.

Je tiens à remercier Mr. BENNIA Abdelhak, Mr. BOUDJEHEM Djalil et Mr. LADACI Samir d'avoir accepté de participer à mon jury. Je les remercie également d'avoir lu ce mémoire et pour les remarques pertinentes qu'ils m'ont faites afin de l'améliorer.

Je tiens également à remercier Romain pour son aide et ses précieux conseils. Merci à Amani, Wafa, Hela B., Meriem, Bilel et Houssam pour leur amitié. Je remercie très spécialement Hela B.-S. pour son énergie contagieuse, sa présence permanente et sa générosité. Un grand merci pour ton soutien, Hela, et ta précieuse amitié.

Je remercie l'ensemble de ma famille pour leur souci constant de l'avancement de mon travail. Merci particulièrement à mes parents et ma petite sœur qui ont fait en sorte que j'effectue ce travail dans une bonne ambiance familiale. Un merci tout spécial aux personnes qui ont toujours cru en moi et qui m'ont encouragée pour mener à terme ce travail.

*À mes chers parents,
À la mémoire de mon oncle.*

Table des matières

Remerciements	iii
Table des matières	v
Liste des figures	vii
Liste des tableaux	ix
Liste des acronymes	xi
Introduction générale	1
1 État de l’art	5
1.1 Introduction	5
1.2 Commande des multi-rotors	11
1.2.1 Les contrôleurs linéaires	11
1.2.2 Contrôleurs non-linéaires	13
1.2.3 Contrôleurs intelligents basés sur l’apprentissage	18
1.3 Planification de chemins et évitement d’obstacles	18
1.3.1 Algorithmes à base des graphes	19
1.3.2 Algorithmes basés sur l’échantillonnage	20
1.3.3 Algorithmes basés sur des modèles mathématiques	21
1.3.4 Algorithmes bio-inspirés	22
1.3.5 Algorithmes multi-fusionnels	25
1.4 Conclusion	25
2 Modélisation des drones multi-rotors	27
2.1 Introduction	27
2.2 Référentiels et matrices de passage	28
2.3 Modélisation cinématique	30
2.4 Modélisation dynamique	31
2.5 Modèle généralisé d’un drone N -rotor	33
2.5.1 Forces et moments aérodynamiques	33
2.5.2 Effets gyroscopiques	35
2.5.3 Dynamique de translation	36
2.5.4 Dynamique de rotation	37
2.6 Identification des paramètres du multi-rotor	43
2.7 Conclusion	44
3 Commande et suivi de trajectoires	45
3.1 Introduction	45
3.2 Structure de la commande hiérarchique d’un drone multi-rotor	46

3.3	Commande hybride Back-stepping A.I. - PID	48
3.3.1	Boucle externe de la commande	48
3.3.2	Boucle interne de la commande	52
3.3.3	Étude comparative	53
3.4	Allocation des commandes sur les actionneurs	59
3.5	Résultats numériques	61
3.6	Étude en présence de pannes	66
3.7	Conclusion	76
4	Planification de chemins et évitement d'obstacles	77
4.1	Introduction	77
4.2	Planification de chemins par courbes de Bézier	78
4.2.1	Définitions et propriétés	78
4.2.2	Exemple de planification de chemin en 2D	80
4.2.3	Application pour des environnements 3D	81
4.3	Planification de chemin par squelettisation de l'environnement	87
4.3.1	Squelettisation de l'environnement	87
4.3.2	Calcul du chemin le plus court	90
4.3.3	Technique de lissage	90
4.4	Résultats numériques	92
4.4.1	résultats de l'approche proposée	92
4.4.2	Étude comparative	98
4.5	Conclusion	101
	Conclusion générale et perspectives	103
	Bibliographie	105
	Résumé	119
	Abstract	121
	Résumé arabe	123

Liste des figures

1.1	Différents modèles de bi-rotors. (a) Hélicoptère Yamaha RMAX, (b) Drone IT180 (rotors coaxiaux libres), (c) Drone HoverEye (rotors coaxiaux carénés).	6
1.2	Tri-coptère QuadLugs.	6
1.3	Drone Y4 Abzugeben.	7
1.4	Drone Y4 de l'université du Queensland.	7
1.5	Exemples de quadri-rotors à configuration standard.	7
1.6	Exemple d'un quadri-rotor à configuration H : drone Manta-Ray.	8
1.7	Drone Lynxmotion Hunter VTail.	8
1.8	Hexa-rotor standard.	8
1.9	Drone Y6.	9
1.10	Octo-rotor standard.	9
1.11	Octo-rotor à configuration V.	9
1.12	Drone X8.	10
1.13	Différentes configurations.	10
1.14	Configuration standard des multi-rotors : type ' + ' ligne 1, type ' × ' ligne 2.	11
1.15	Connectivité de nœuds dans une grille d'occupation.	20
2.1	Drone multi-rotor dans le référentiel inertiel fixe.	28
2.2	Angle d'Euler.	29
2.3	Forces de poussée générées par les rotors.	34
2.4	Illustration du moment de traînée généré par le rotor.	35
2.5	Représentation de l'angle λ des deuxième et quatrième rotors de l'octo-rotor à configuration ' + ' (à gauche) et configuration ' × ' (à droite).	38
2.6	Principe de mouvements de roulis et de tangage.	40
2.7	Principe de mouvement de lacet.	41
2.8	Identification expérimentale des coefficients de poussée (à gauche) et de traînée (à droite).	44
3.1	Illustration du couplage entre le mouvement horizontal et l'orientation de l'assiette d'un multi-rotor.	47
3.2	Structure de la commande hiérarchique d'un multi-rotor.	48
3.3	Stratégie de commande hybride proposée.	49
3.4	Commande de la position verticale.	49
3.5	Contrôleur PID de forme parallèle.	53
3.6	Contrôleur hybride implémenté sous Matlab/Simulink.	53
3.7	Représentation 3D de la trajectoire d'un quadri-rotor, obtenue par le Backstepping AI (en <i>vert</i>) et la stratégie hybride (en <i>noir</i>). La trajectoire désirée est représentée en <i>rouge</i> , avec un zoom sur le régime transitoire (en <i>haut à droite</i>).	54

3.8	Résultats de suivi de références (en <i>rouge</i>) de la position $(x, y, z)_m$ d'un quadri-rotor en appliquant le Back-stepping AI (en <i>vert</i>) et la stratégie hybride (en <i>noir</i>)	55
3.9	Résultats de stabilisation de l'orientation d'un quadri-rotor en appliquant le Back-stepping AI (en <i>vert</i>) et la stratégie hybride (en <i>noir</i>).	56
3.10	Erreur de suivi de références de la position $(x, y, z)_m$ d'un quadri-rotor en appliquant le Back-stepping AI (en <i>vert</i>) et la stratégie hybride (en <i>noir</i>)	57
3.11	Erreur de stabilisation de l'orientation d'un quadri-rotor en appliquant le Back-stepping AI (en <i>vert</i>) et la stratégie hybride (en <i>noir</i>).	58
3.12	Structure de contrôle de drone multi-rotor.	59
3.13	Résultats de suivi de trajectoire en utilisant un drone quadri-rotor.	62
3.14	Résultats de suivi de trajectoire en utilisant un drone hexa-rotor.	63
3.15	Résultats de suivi de trajectoire en utilisant un drone octo-rotor.	64
3.16	L'entrée de commande $U_1(t)$ en utilisant, (a) un quadrirotor, (b) un hexa-rotor et (c) un octo-rotor. Les valeurs numériques de U_1 dans les trois cas correspondent au poids des multi-rotors (<i>m.g</i>).	65
3.17	Stabilisation sur un point cible pour $N = 8$ rotors. Déclenchement de la panne à $t = 15s$ où la position de drone est représentée par le <i>point noir</i> . Le <i>point bleu</i> représente la position de drone à l'instant $t = 30s$ correspondant au temps de simulation dans la figure (3.19). A l'instant $t = 300s$, le système récupère presque le point 3D désiré.	67
3.18	Suivi de référence en 3D pour $N = 8$ après une panne de rotor.	68
3.19	Scénario.1 : position 3D d'un N -rotor ($6 \leq N \leq 16$) avec un rotor en panne (panne déclenchée à $t = 15s$).	69
3.20	Scénario.1 : orientation d'un N -rotor ($6 \leq N \leq 16$) avec un rotor en panne (panne déclenchée à $t = 15s$).	70
3.21	Scénario.2 : position 3D d'un N -rotor ($6 \leq N \leq 16$) avec un rotor en panne (panne déclenchée à $t = 20s$).	71
3.22	Scénario.2 : orientation d'un N -rotor ($6 \leq N \leq 16$) avec un rotor en panne (panne déclenchée à $t = 20s$).	72
3.23	Scénario.1 : allocation des entrées de commande sur les actionneurs d'un N -rotor avant et après le déclenchement de la panne totale d'un rotor ; (a) moyenne des vitesse des rotors, (b) écart type des vitesses.	73
3.24	Scénario.2 : allocation des entrées de commande sur les actionneurs d'un N -rotor avant et après le déclenchement de la panne totale d'un rotor ; (a) moyenne des vitesse des rotors, (b) écart type des vitesses.	74
3.25	Erreur en x, y, z ([m]) et ψ ([deg]) pour le cas d'un octo-rotor avant et après déclenchement de la panne totale d'un rotor.	75
4.1	Courbe de Bézier d'ordre 3.	80
4.2	Exemple de passage de la porte dans la robotique mobile.	81
4.3	Exemple d'une courbe de Bézier avec boucle.	83
4.4	Continuité de classe C^1 entre deux courbes de Bézier.	85
4.5	Exemple de boucle au point de raccordement de deux courbes de Bézier cubiques.	85
4.6	Scénario 1. Exemple de chemin planifié par courbes de Bézier.	86
4.7	Scénario 2. Exemple de chemin planifié par courbes de Bézier.	87
4.8	Représentation illustrative des concepts impliqués dans le calcul du DMA.	89
4.9	illustration du principe de l'approche de lissage sur la $k^{\text{ième}}$ courbe de Bézier.	91

4.10 Deux cas de squelette de l'environnement obtenus pour différentes tailles de drone.	93
4.11 La représentation 3D du chemin le plus court calculé sur le FMS pour différentes tailles de drone.	94
4.12 La représentation 2D du chemin le plus court calculé sur le FMS pour différentes tailles de drone.	95
4.13 Changement d'altitude tout le long du chemin le plus court calculé sur le FMS pour différentes valeurs de δ (20cm (à gauche), 40cm (à droite)).	95
4.14 Chemin final pour différentes tailles de drone représenté en 3D.	96
4.15 La représentation 2D du chemin final pour différentes tailles de drone.	97
4.16 Comparaison entre l'approche proposée (ligne continue) pour différentes valeurs de δ ($= 20\text{cm}$ (couleur cyan), $= 40\text{cm}$ (couleur magenta)) et le A^* appliqué sur la G.O. gonflée prenant en compte les deux cas : $\delta = 20\text{cm}$ (discontinue ligne couleur orange), $\delta = 40\text{cm}$ (ligne pointillée couleur noire).	99
4.17 Évolution de la distance aux obstacles ((a) : $\delta = 20\text{cm}$, (b) : $\delta = 40\text{cm}$).	100

Liste des tableaux

3.1 Valeurs numériques utilisées dans la simulation.	61
3.2 Paramètres de drones multi-rotors.	66
4.1 Résultats de lissage.	97
4.2 Comparaison quantitative entre le A^* appliqué directement sur la G.O. et l'approche proposée en considérant deux différents cas de taille du drone ($\delta = 20\text{cm}$ (cas 1), $\delta = 40\text{cm}$ (cas 2)).	98

Liste des acronymes

A.I. avec action intégrale. [4](#), [37](#), [40](#), [44](#), [45](#), [58](#)

ACS Admissible Control Set. [51](#)

ADAV Aéronefs à Décollage et Atterrissage Verticaux. [2–4](#), [12](#), [25](#)

DDL degrés de liberté. [4](#), [12](#), [18](#), [28](#), [40](#), [44](#), [45](#)

DMA Delta Medial Axis. [80](#), [81](#)

FMS Filtered Medial Surface. [79](#), [81](#), [82](#), [84](#), [90](#), [93](#)

G.O. grille d'occupation. [20](#), [90](#), [91](#)

GNC Guidage, Navigation et Contrôle. [2](#)

GVG Graphe de Voronoï généralisé. [79–81](#)

IMA Integer Medial Axis. [80](#), [81](#)

PID Proportional, Integral, Derivative. [111](#)

PRM Probabilistic Road Map. [20](#), [21](#)

RRT Rapidly-exploring Random Tree. [20](#), [21](#), [25](#)

UAV Unmanned Aerial Vehicle. [1](#), [111](#)

VTOL Vertical Take-Off and Landing. [2](#), [3](#), [111](#)

Introduction générale

Dans les milieux aéronautiques, le champ d'utilisation des drones est de plus en plus étendu. Cela est le fruit des avancées technologiques multidisciplinaires permettant la miniaturisation des actionneurs et de l'électronique embarquée. Nommés aussi **Unmanned Aerial Vehicles (UAVs)**, ces véhicules sont des aéronefs (engins plus lourds que l'air) sans pilote humain à bord, autonomes ou commandés à distance. Leurs domaines d'utilisation sont divers et variés. On peut en particulier citer leur utilisation pour des missions de surveillance, de renseignement ou de combat, mais aussi pour effectuer des missions dans les domaines de la sécurité civile et de l'industrie (inspection des lignes haute-tension par exemple).

Les drones sont classés selon plusieurs critères : altitude de vol, autonomie, taille ou charge utile [1, 2] :

- Les drones volant à haute altitude (> 20 000m) et de grande autonomie sont appelés *HALE* (High Altitude Long Endurance).
- Les drones volant à moyenne altitude (5 000m à 15 000m) et de grande autonomie (20 à 40 heures) sont appelés *MALE* (Médium Altitude Long Endurance).
- Les drones tactiques (*TUAV*) ont une portée d'action moyenne. Ils se situent entre les mini-drones à petite portée et les drones stratégiques à longue portée.
- Les drones de combat (*UCAV*, ce sont des drones particulièrement équipés pour le combat. Ils peuvent être tactiques ou stratégiques et sont capables d'emporter du matériel de reconnaissance ou de l'armement.

Selon le type de drones, la masse peut varier de quelques dizaines de grammes à une quinzaine de tonnes. La taille également peut aller de quelques centimètres pour les micro-drones et les drones miniatures à plusieurs mètres pour les drones spécialisés. C'est les performances requises par la mission et la nature et l'importance de la charge utile qui sont déterminantes.

Les drones sont également classés selon le type de leur voilure ; on distingue les drones à voilure fixe et les drones à voilure tournante. Dans la première catégorie, les mécanismes de sustentation et de propulsion sont séparés. L'aile (voilure fixe) permet la portance de l'avion. Alors que la propulsion est assurée par une installation motrice.

Pour les drones à voilure tournante, la sustentation et la propulsion sont toutes les deux assurées par les rotors. Cette caractéristique leur permet d'effectuer des décollages et des atterrissages verticaux, d'où vient leur appellation [Aéronefs à Décollage et Atterrissage Verticaux \(ADAV\)](#), en anglais [Vertical Take-Off and Landing \(VTOL\)](#) aircrafts. Les drones dits 'convertibles' combinent les deux types de voilures [3].

Pour leur fonctionnement et afin d'accomplir une mission spécifique dans un site de travail réel, les drones ont besoin d'un système de [Guidage, Navigation et Contrôle \(GNC\)](#), capable de fonctionner de manière efficace, en toute sécurité et de manière autonome. Dans un système [GNC](#) :

- Le guidage se réfère à la détermination du trajet où le véhicule doit se déplacer depuis son emplacement actuel jusqu'à une cible désignée.
- La navigation se réfère à la détermination du vecteur d'état (position, attitude et vitesses) du véhicule.
- Le contrôle se réfère à la manipulation des forces et des moments fournis par les actionneurs du véhicule afin de produire le mouvement attribué par le système de guidage tout en assurant la stabilité du véhicule.

La recherche sur les drones pour applications non-militaires n'a commencé qu'au début des années 90. Plusieurs projets ont été lancés sur le sujet et un grand nombre d'articles a été publié dans la littérature. Le projet Mesicopter de l'université de Stanford financé par la NASA a débuté en 1999. Il consiste en la conception d'un drone quadri-rotor de très petite taille. Le but de ce projet est de traiter les problématiques liées à la conception de drones miniatures et d'étudier les différents effets aérodynamiques. Des résultats intéressants sur les effets aérodynamiques générés par les hélices ont été publiés [4], et utilisés ensuite dans d'autres travaux de recherche [5]. Le projet X4-flyer MARK de l'université nationale d'Australie a commencé en 2002 sous la direction de Paul Pounds et Robert Mahony. Ce projet consiste en la fabrication d'un quadri-rotor de 4kg offrant une charge utile de 1kg. Des études sur la conception des hélices et l'effet de battement d'hélice ont été effectuées, et des conclusions ont été tirées. La stabilité du véhicule en vol était l'un des objectifs principaux de ce projets. Plusieurs papiers ont été publiés dans le cadre de ce projet, notamment sur le contrôle de l'attitude par PID [6, 7].

Le projet OS4 de la thèse de Samir Bouabdallah dirigée par Roland Siegwart a été initié en 2004 au laboratoire de systèmes autonomes de l'EPFL. Ce projet consiste à travailler sur la conception et le contrôle d'un quadri-rotor à l'intérieur. Après la fabrication de la plateforme quadri-rotor OS4, l'objectif était de développer un système de contrôle fiable pour la stabilisation du véhicule et le suivi des trajectoires. Dans le cadre de sa thèse, Bouabdallah a comparé les performances de différentes approches de commande linéaires et non-linéaires pour la stabilisation de l'assiette du véhicule. En se basant sur les résultats obtenus et les conclusions tirées, l'auteur a proposé un contrôleur de type Back-stepping inté-

gral pour la dynamique complète du quadri-rotor [5]. Le projet STARMAC (2004-2012) de l'université de Stanford dirigé par M.Hoffmann porte également sur la fabrication, l'analyse et le contrôle de quadri-rotor. Ce projet a été le sujet de plusieurs publications sur la modélisation, la conception, le contrôle du quadri-rotor ainsi que le suivi de trajectoire [8–10]. D'autres publications ont porté sur la coopération multi-drones [11, 12].

À la fin des années 2000, la recherche sur les drones [ADAV](#) a évolué vers l'utilisation des drones commerciaux. Vijay Kumar a utilisé dans ses recherches des quadri-rotors conçus par la compagnie Ascending Technologies. Ses travaux avec l'équipe du laboratoire GRAPS de l'université de la Pennsylvanie ont fait l'objet de plusieurs publications et démonstrations vidéos. Ils se sont concentrés sur la planification de trajectoires agressives, la réalisation de tâches par une équipe de drones quadri-rotors ainsi que la collaboration multi-robots mixtes terrestres / aériens. Des quadri-rotors Hummingbird et des hexa-rotors Firefly de l'Ascending Technologies ont été également utilisés dans d'autres projets de recherche [13, 14].

L'intérêt particulier accordé aux drones [ADAV](#) a fait en sorte que la littérature soit riche et élaborée. Diverses méthodes de planification de chemins pour robots terrestres ont été étendues en 3D pour applications aux drones. Cela a révélé de nombreuses problématiques de guidage telles que la complexité des algorithmes et la puissance de calcul exigée qui représentent un défi majeur lors des applications en temps réel. Bien que l'autonomie absolue reste une question ouverte, l'amélioration des algorithmes de guidage existants ou la recherche de nouvelles stratégies plus pratiques est vivement souhaitable afin de résoudre ces problématiques. Pour le contrôle, presque toutes les méthodologies de commande, classiques et modernes, ont été appliquées aux drones [ADAV](#). Selon la complexité de la mission et les conditions de vol imposées, certaines méthodes sont plus adaptées que d'autres. Une idée pertinente est de mixer plusieurs algorithmes de commande pour améliorer les performances du contrôleur.

Le travail présenté dans ce manuscrit se situe dans le cadre de problèmes de la planification de chemin, et de la commande et la stabilisation de véhicules miniatures de type [ADAV](#). Nous nous intéressons particulièrement aux [ADAV](#) à rotors multiples ayant une configuration standard ('+' ou '×'). Pour cela, nous développons un modèle dynamique unifié pour ce type de véhicules, que nous utilisons pour calculer notre contrôleur. Nous adoptons l'idée d'hybridation de deux algorithmes de commande (linéaire et non-linéaire) pour stabiliser le véhicule et effectuer un suivi de trajectoires. Pour le guidage, nous proposons une stratégie originale de planification de chemins 3D en prenant en compte les dimensions du robot aérien.

Les objectifs de ce travail sont les suivants :

- La modélisation dynamique des multi-rotors miniatures à décollage et atterrissage verticaux en proposant un modèle unifié,

- La conception d'un contrôleur pour la stabilisation et la commande de ce type d'engins,
- L'élaboration d'une approche de génération de chemins et d'évitement d'obstacles pour leur déplacement.

Le chapitre 1 introduit les différentes configurations de drones de type [ADAV](#). Une revue de littérature récente est fournie ensuite portant sur la commande et la génération de chemins en présentant des notions de base et discutant les méthodes utilisées pour le guidage et le contrôle.

Dans le chapitre 2, la modélisation d'un drone multi-rotor est détaillée dans l'objectif de concevoir les lois de commande et les valider. Le formalisme d'Euler-Newton est appliqué pour dériver le modèle mathématique. Une généralisation des expressions des forces et des différents couples en fonction du nombre de rotors est faite en vue de proposer un modèle unifié valable pour N rotors et pour les configurations ' + ' et ' × ' définies selon la position de la face avant du multi-rotor.

Le chapitre 3 traite la conception des lois de commande pour la stabilisation d'un multi-rotor. L'approche Back-stepping [avec action intégrale \(A.I.\)](#) et le contrôleur PID sont utilisés pour la stabilisation et la commande du véhicule. Nous proposons une commande hiérarchique selon une structure globale en cascade en décomposant la dynamique du système en deux sous-systèmes : translation et rotation. Le contrôleur hybride est appliqué pour élaborer des lois de commande pour la stabilisation des six [degrés de liberté \(DDL\)](#) du multi-rotor. Les lois de commande obtenues sont ensuite validées pour différents cas de multi-rotors. Le comportement des multi-rotors en présence d'une panne totale sur un des rotors est aussi abordé dans ce chapitre.

Le chapitre 4 porte sur la planification de chemins 3D en présence d'obstacles. Étant supposé que l'environnement est idéalement connu, des stratégies de planification de chemins sont proposées et validées ensuite sur des exemples d'environnements intérieurs. Utilisant la squelettisation de l'environnement, la solution présentée permet de chercher le chemin désiré en fonction de la taille du drone.

Chapitre 1

État de l'art

Sommaire

1.1	Introduction	5
1.2	Commande des multi-rotors	11
1.2.1	Les contrôleurs linéaires	11
1.2.2	Contrôleurs non-linéaires	13
1.2.3	Contrôleurs intelligents basés sur l'apprentissage	18
1.3	Planification de chemins et évitement d'obstacles	18
1.3.1	Algorithmes à base des graphes	19
1.3.2	Algorithmes basés sur l'échantillonnage	20
1.3.3	Algorithmes basés sur des modèles mathématiques	21
1.3.4	Algorithmes bio-inspirés	22
1.3.5	Algorithmes multi-fusionnels	25
1.4	Conclusion	25

1.1 Introduction

Il existe de nombreuses configurations de drones miniatures à déplacement vertical. Elles diffèrent les unes des autres par le nombre des rotors dont le drone est équipé et leur positionnement. On distingue en particulier les configurations à deux rotors et les configurations multi-rotors.

- Configurations à deux rotors

L'hélicoptère classique miniature fait partie de cette catégorie. Il est équipé d'un rotor principal et d'un rotor de queue contre-rotatif. Le rotor principal permet la sustentation et la propulsion. Le mouvement naturel en lacet produit par le couple de réaction du rotor principal est compensé par le couple du rotor de queue.

L'hélicoptère n'est pas le seul bi-rotor qui existe. D'autres modèles se distinguent par la position des rotors contrarotatifs : les bi-rotors coaxiaux libres ou carénés, en tandem, coté à coté, etc.

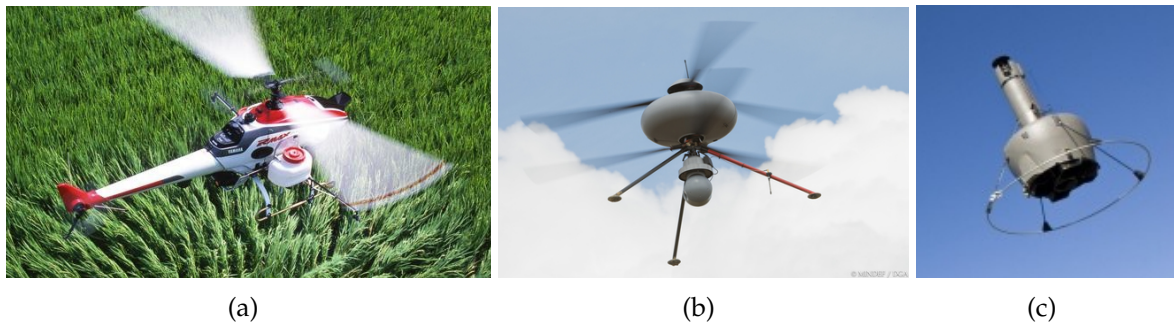


Figure 1.1. Différents modèles de bi-rotors. (a) Hélicoptère Yamaha RMAX, (b) Drone IT180 (rotors coaxiaux libres), (c) Drone HoverEye (rotors coaxiaux carénés).

- Configurations multi-rotors

Il existe des formes très différentes des drones multi-rotors. Ils sont équipés de trois à plusieurs rotors, positionnés de manière spécifique afin d'assurer la sustentation et la propulsion du véhicule. La structure d'un drone multi-rotors se compose en général d'une sorte de plateau central contenant les batteries et toute l'électronique, auquel est relié un ensemble de bras dont chacun porte à son extrémité un moteur et une hélice.

a) Drone tri-rotor. C'est un drone doté de trois bras dont chacun est connecté à un moteur. La face avant du drone est généralement placée entre deux des bras. Dans la configuration la plus simple d'un tri-rotor (**Y3**), le moteur arrière est à angle variable commandé par un servomoteur. Cela permet de contrecarrer l'effet gyroscopique du nombre impair de rotors, et modifier l'angle de lacet.

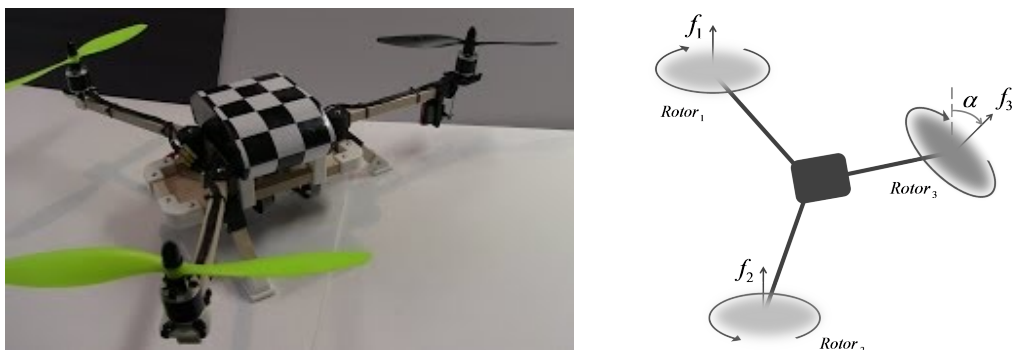


Figure 1.2. Tri-coptère QuadLugs.

Le modèle quadri-rotor triangulaire (**Y4**) fait également partie des tri-rotors même s'il est doté de quatre moteurs. Dans ce modèle, deux moteurs sont montés sur le bras arrière pour gérer les effets gyroscopiques.

Un autre modèle **Y4** doté d'un rotor principal au centre et de trois petits rotors à l'extrémité de chaque bras est développé à l'université du Queensland [15]. Dans ce modèle,

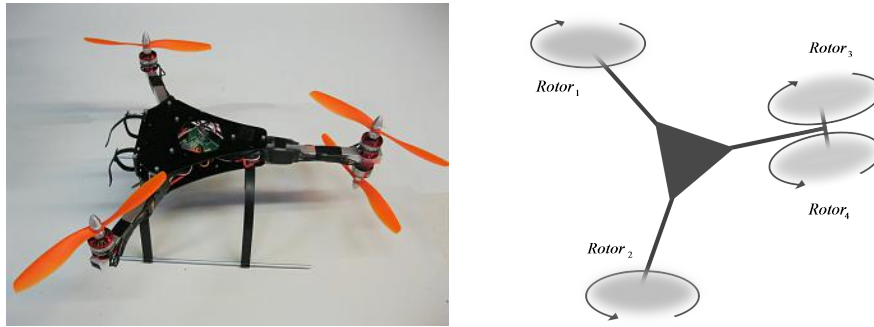


Figure 1.3. Drone Y4 Abzugeben.

la sustentation et la propulsion sont assurées par le rotor principal. Les trois petits rotors fournissent un contre-couple pour annuler l'effet gyroscopique.



Figure 1.4. Drone Y4 de l'université du Queensland.

- b) **Drone quadri-rotor.** C'est la conception multi-rotor la plus populaire. Il est doté de quatre moteurs et de quatre bras de support. l'avant du drone peut être placé entre deux bras comme il peut se trouver le long d'un bras. Dans la configuration standard (X4), les bras et les moteurs sont symétrique par rapport à deux axes.



(a) Drone HummingBird de AscTec (configuration '+')



(b) Drone Pelican de Asctec (configuration 'x')

Figure 1.5. Exemples de quadri-rotors à configuration standard.

Dans le modèle Manta-Ray de C.J. Youngblood Ent., quatre moteurs sont montés à l'extrémité gauche et droite de deux bras de support parallèles. Cette configuration est appelée : configuration **H**.



Figure 1.6. Exemple d'un quadri-rotor à configuration H : drone Manta-Ray.

Il existe également des configurations quadri-rotors non symétriques, comme le drone V-Tail avec une configuration unique à quatre bras, dont les deux à l'arrière forment un angle en **V**.

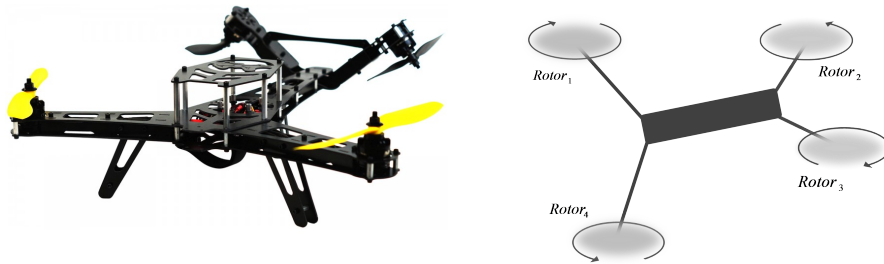


Figure 1.7. Drone Lynxmotion Hunter VTail.

- c) **Drone hexa-rotor.** C'est un drone doté de six bras de support et six rotors. La face avant est généralement placée entre deux bras, mais peut aussi se trouver le long d'un rotor. Dans sa configuration standard, les bras sont symétriques par rapport à deux axes et l'angle entre eux est égal à 60 degrés.



Figure 1.8. Hexa-rotor standard.

Il existe d'autres configurations à six rotors, notamment la **Y6**. Comme le **Y3**, un drone **Y6** est doté de trois bras de support uniquement, mais de deux moteurs par bras (un vers le haut, et un autre vers le bas).

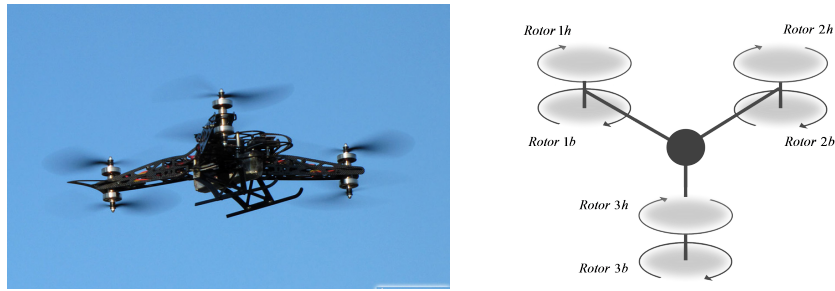


Figure 1.9. Drone Y6.

- d) **Drone octo-rotor.** C'est un véhicule à structure symétrique à deux axes, doté de huit rotors. Ces derniers peuvent être positionnés de différentes manières. La face avant peut être placée entre deux bras comme elle peut être au long d'un bras. La configuration standard d'un octo-rotor est montrée dans la figure (1.10).



Figure 1.10. Octo-rotor standard.

Un exemple d'une autre configuration à huit bras et huit rotors est représenté sur la figure (1.11). Dans cette configuration, chaque rotor dans le quadri-rotor est remplacé par un couple de rotors coté à coté formant un angle en V.



Figure 1.11. Octo-rotor à configuration V.

La même idée se trouve dans la configuration X8, où chaque rotor du drone X4 est remplacé par deux rotors co-axiaux (un vers le haut, et l'autre vers le bas).

- e) **Autres multi-rotors.** Il existe d'autres configurations multi-rotor avec un grand nombre de moteurs / hélices. Le choix d'une configuration multi-rotor dépend de ce qu'on va faire avec. Plus on a de moteurs, plus la poussée totale disponible est importante, ce qui signifie que l'appareil peut soulever plus de charge utile. Par ailleurs, si un moteur tombe en panne, il reste encore une chance pour que l'appareil puisse atterrir plutôt



Figure 1.12. Drone X8.

que de s'écraser. En revanche, augmenter le nombre des moteurs nécessite plus d'espace (taille/envergure plus grande) et un bloc de batteries plus important. En outre, avoir plus de moteurs signifie un prix plus élevé.

La figure (1.13) représente quelques exemples de multi-rotors avec différentes configurations et différents nombres de rotors.

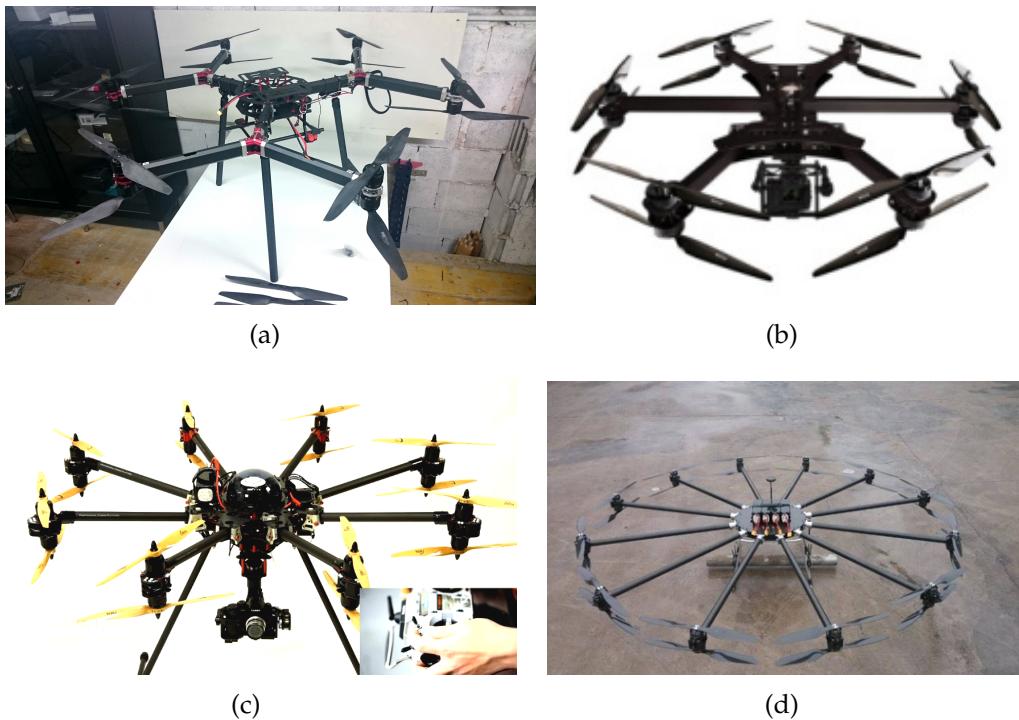


Figure 1.13. Différentes configurations.

• Configuration '+' et 'x'

Selon l'emplacement de la face avant du drone multi-rotor, on distingue deux configurations possibles : la configuration '+' et la configuration 'x'. Dans la première, la face avant du multi-rotor est défini par le 1^{er} rotor (ou bras) positionné sur l'axe du roulis dans le référentiel associé au robot. En revanche, dans la configuration 'x', la face avant du drone est défini par le 1^{er} et le N^{ème} rotor avec l'axe de roulis pris entre eux. La figure (1.14) représente des schémas illustratifs des multi-rotors standards classifiés selon l'emplacement de la face avant. Un exemple de systèmes drones réels pour chaque configuration '+' et 'x' est montré sur la figure (1.5).

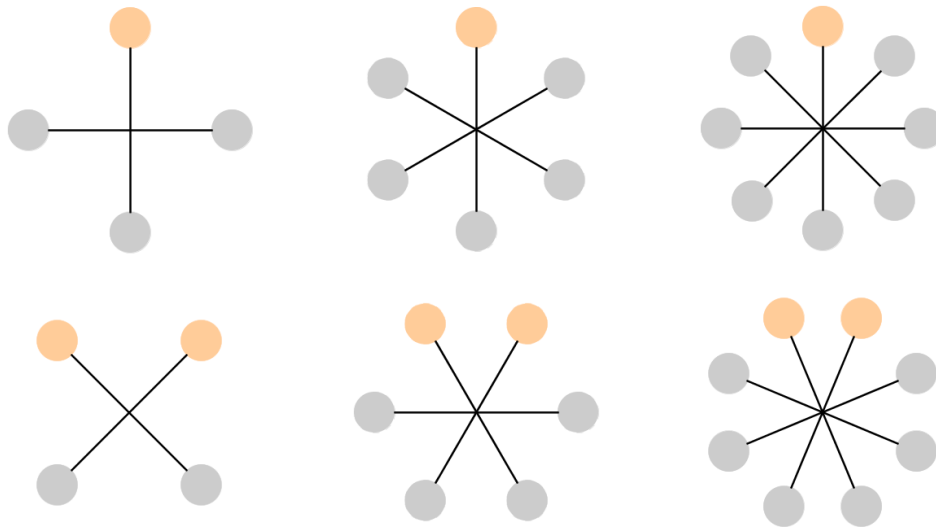


Figure 1.14. Configuration standard des multi-rotors : type ' + ' ligne 1, type ' × ' ligne 2.

1.2 Commande des multi-rotors

La conception d'un contrôleur de vol consiste à synthétiser des lois de commande qui calculent les entrées des actionneurs du véhicule pour produire les forces et les couples qui agissent et permettent d'obtenir un comportement désiré. Le domaine de la commande des multi-rotors est très vaste et ne peut pas être abordé ici dans son intégralité.

Les premiers travaux de recherche sur les multi-rotors se sont concentrés sur les drones quadri-rotors. Cela est en partie grâce aux diverses caractéristiques du quadri-rotor. Sa simplicité mécanique, son sous-actionnement ainsi que la non-linéarité et le couplage de sa dynamique ont rapidement fait du quadri-rotor un sujet de recherche très populaire. De plus, les avancées technologiques dans le domaine de la miniaturisation de l'électronique embarquée et l'apparition des micro IMU (Inertial Measurement Unit) ont permis de concevoir des quadri-rotors à taille réduite et à faible coût dédiés à la recherche. Au cours des dernières années, les drones hexa- et octo- rotors ont commencé à gagner l'attention de la communauté des chercheurs notamment dans le domaine de la commande tolérante aux défauts.

Les approches de commande de vol proposées dans la littérature pour le contrôle des multi-rotors peuvent être classées en trois catégories : les contrôleurs linéaires, les contrôleurs non-linéaires et les contrôleurs intelligents.

1.2.1 Les contrôleurs linéaires

Dans ce type de contrôleurs, la synthèse de la commande est basée sur une approximation linéaire du modèle dynamique du véhicule. Dans la pratique, les premiers contrôleurs proposés pour la commande des multi-rotors en vol autonome sont des contrôleurs

linéaires de type PID, LQR et H_∞ .

- **Contrôleur PID**

Pour la commande de vol des multi-rotors, le contrôleur PID [16] a été implémenté seul ou comme partie d'une structure hiérarchique d'un algorithme de commande.

Dans un processus de commande hiérarchique (commande en cascade) appliquée sur les drones ADAV d'une manière générale, la dynamique du véhicule est décomposée en deux sous systèmes : dynamique de rotation et dynamique de translation. La commande est conçue ensuite en deux boucles : une boucle interne de commande en attitude et une boucle externe de commande en position [17]. La caractéristique de sous-actionnement et le couplage dans la dynamique de ces systèmes font de la commande en cascade un choix approprié pour leur contrôle. Ce principe de loi de commande séparée a été adopté dans de nombreux travaux [18–21].

La stratégie de commande en cascade par contrôleur PID est la plus utilisée dans la pratique. Elle a été implémentée dans les systèmes PX4 [22] et ArduPilot [23]. Les auteurs dans [24] présentent une évaluation de trois structures différentes du PID pour la commande de l'attitude d'un quadri-rotor. Dans [6], le contrôleur PID a été implémenté sur un drone quadri-rotor plus lourd que la plupart des plates-formes expérimentales quadri-rotors. Le contrôleur a montré une bonne stabilisation des angles de roulis et tangage.

Le contrôleur PID classique a l'avantage que les valeurs des gains sont facile à ajuster, il est simple à concevoir et robuste. Cependant, la non-linéarité du modèle mathématique des multi-rotors, et les incertitudes liées à la modélisation inexacte de certaines dynamiques du système représentent un défi majeur. Cela peut limiter les performances du contrôleur PID appliqué aux mutli-rotors.

- **Commande linéaire quadratique (LQ)**

La commande LQ est une commande optimale permettant de calculer une matrice de gain sur un retour d'état du système. Elle a été appliquée pour le contrôle des multi-rotors. Une implémentation à structure hiérarchique de la commande LQ a été proposée dans [25] pour le contrôle des six DDL d'un quadri-rotor. La commande LQ a été également appliquée avec une action intégrale pour le contrôle d'un quadri-rotor [26] et d'un octo-rotor [27]. Des résultats satisfaisants ont été obtenus.

Cette méthode à été comparée à la commande par PID pour le contrôle de la boucle interne (orientation) d'un quadri-rotor dans le cadre du projet OS4 [28]. Le contrôleur PID a été appliqué en considérant la dynamique simplifiée du quadri-rotor. Cependant, la commande LQ a été appliquée sur un modèle plus complet du véhicule. Contrairement aux attentes, la commande LQ a été moins performante, et expérimentalement moins dynamique que le contrôleur PID. Ce dernier a prouvé son aptitude pour le contrôle des

angles d'orientation du quadri-rotor en vol stationnaire autonome en présence de perturbations mineures. Les auteurs ont signalé que le PID ne pourra pas stabiliser le véhicule en présence de fortes perturbations, et ont été fermement convaincus que la commande LQ devrait donner de meilleurs résultats.

- **Commande H_∞**

La synthèse de commande H_∞ [29] permet la conception de lois de commande optimales. Dans [30], les auteurs ont conçu un contrôleur basé sur la commande H_∞ pour le suivi de trajectoire d'un quadri-rotor. Les performances de cette approche ont été comparées à celles de la commande LQ-Servo, basée sur la commande LQ en combinant les avantages d'une action intégrale. Les résultats de simulation ont montré un bon suivi de trajectoire en présence de perturbation du vent par la commande LQ-Servo. Cependant, le contrôleur basé sur l'approche H_∞ montre des performances de suivi assez bonnes en absence de perturbations externes, mais il perd complètement le suivi en présence de perturbations.

On relève également le travail de Ortiz et *al.* [31], qui ont utilisé pour le contrôle de l'attitude, un contrôleur PID dont les gains sont obtenus en utilisant la synthèse H_∞ . Par comparaison aux résultats obtenus avec un PID classique, l'approche proposée donne de meilleurs résultats. Ils ont également démontré la robustesse de la méthode par des tests expérimentaux. Une idée similaire a été proposée quatre ans auparavant par García et *al.*, et validée en simulation [32]. Dans [33], la commande GH_∞ a été combinée avec la commande non-linéaire basée sur la linéarisation par retour d'état pour résoudre le problème de suivi de trajectoire d'un quadri-rotor. La différence essentielle entre H_∞ et GH_∞ est que la première méthode utilise l'algorithme d'itération pour calculer la loi de commande, alors que la seconde utilise le problème (de la / du) valeur propre/vecteur propre pour obtenir une solution plus facile à calculer.

Les auteurs ont considéré le contrôleur non-linéaire pour la stabilisation de la boucle interne, et la commande GH_∞ pour la boucle externe. La performance du contrôleur hybride à structure hiérarchique a été démontrée en simulation en présence des incertitudes de paramètres et des perturbations externes.

1.2.2 Contrôleurs non-linéaires

- **Commande basée sur la linéarisation par retour d'état**

La linéarisation par retour d'état est une approche couramment utilisée pour contrôler les systèmes non linéaires. Elle consiste à proposer une transformation du système non linéaire en un système linéaire équivalent sur lequel on peut appliquer des méthodes linéaires. La transformation inverse est utilisée après pour revenir au système original.

La méthode d'inversion dynamique est un cas spécifique de la linéarisation par retour d'état. Dans [34], un contrôleur non-linéaire basé sur ce type de linéarisation a été développé pour le suivi de trajectoires agressives, et a montré des résultats impressionnants. La technique d'inversion dynamique a été utilisée également dans [35] pour synthétiser des lois de commande en deux boucles hiérarchiques : boucle interne (attitude) et boucle externe (position). Le contrôleur de vol non-linéaire conçu a été implémenté à bord d'un quadri-rotor et validé grâce à plusieurs tests de vol.

- **Commande par saturations imbriquées**

La saturation des actionneurs limite l'espace des entrées de commande atteignables, ce qui peut provoquer l'instabilité du véhicule. Différentes stratégies de commandes ont été proposées dans la littérature pour traiter le problème de la saturation des actionneurs. L'objectif est de synthétiser des lois de commande bornées assurant la non-saturation des actionneurs. La commande par saturations imbriquées a été initialement proposée pour la stabilisation globale d'une chaîne d'intégrateurs [36]. Elle a été adaptée ensuite, en utilisant l'analyse de Lyapunov, pour la conception des lois de commande bornées afin de stabiliser un quadri-rotor [37]. Le contrôleur proposé a été testé en expérimentation et s'est avéré capable d'effectuer de manière autonome les tâches de décollage, de vol stationnaire et d'atterrissage. Sur la base de ce travail, une autre version a été développée [38]. Dans [38], la version améliorée a été implémentée à bord d'un quadri-rotor. Les résultats obtenus étaient meilleurs, en comparaison avec ceux obtenus dans [37], en termes de dépassements et de vitesse de convergence des variables d'état.

- **Commande Back-stepping**

La technique Back-stepping est basée sur un algorithme récursif pour la commande des systèmes linéaires et non-linéaires. Elle permet de synthétiser des lois de commande en définissant des fonctions de Lyapunov associées. Dans le processus de conception de lois de commande, certaines composantes du vecteur d'état sont considérées comme commande virtuelle et pour lesquelles une loi de commande intermédiaire est conçue. La technique Back-stepping a été développée par Petar V. Kokotovic dans les années 1990 pour le contrôle d'une classe particulière des systèmes dynamiques non-linéaires [39, 40]. Les auteurs dans [41] ont présenté différentes formes de l'action intégrale dans la conception d'un contrôleur non-linéaire en appliquant la méthode Back-stepping. Divers contrôleurs basés sur la technique Back-stepping ont été proposés dans la littérature pour le contrôle des drones [42–44]. La plupart d'entre eux ont été testés en simulation mais n'ont pas été implémentés en temps réel.

Dans [45], un contrôleur non-linéaire est dérivé, en utilisant des techniques de Back-stepping, et implémenté sur un robot volant de type quadri-rotor. Les résultats expérimentaux ont montré une bonne performance de la stratégie de contrôle proposée. L'algorithme

Back-stepping avec une action intégrale a été présenté dans [46] et implémenté en temps réel à l'aide du système de contrôle embarqué développé pour effectuer un suivi d'attitude d'un quadri-rotor. L'avantage de l'utilisation du terme intégral est d'améliorer les performances du système en régime permanent et augmenter la robustesse aux perturbations externes. Les résultats expérimentaux obtenus ont démontré de bonnes performances de stabilisation, de suivi et de rejet des perturbations.

Dans [47], les auteurs ont représenté la dynamique d'un quadri-rotor sous forme de trois sous-systèmes inter-connectés. Le premier sous-système est sous-actionné représentant la relation dynamique des positions horizontales avec les angles de tangage et de roulis (deux entrées et quatre sorties). Le deuxième sous-système entièrement actionné représente le comportement dynamique de la position verticale et de l'angle de lacet. Le dernier sous-système donne la dynamique des quatre rotors. Un contrôleur de vol a été conçu ensuite en utilisant la commande Back-stepping pour réaliser un suivi de trajectoire de la position 3D et de l'angle de lacet. Les résultats obtenus ont prouvé la performance du contrôleur proposé pour la stabilisation de la dynamique complète et le suivi de trajectoire du quadri-rotor. Enfin, des tests expérimentaux dans lesquels le véhicule est limité aux mouvements verticaux et de lacet ont été effectués pour valider (partiellement) le contrôleur proposé.

- **Commande par mode glissant**

La commande par mode glissant a fait son apparition dans les années 1960 pour répondre au besoin de robustesse par rapport aux perturbations externes et aux incertitudes des paramètres. Elle fait partie des commandes dites à *structure variable* et peut être appliquée aux systèmes linéaires et aux systèmes non-linéaires. Le principe de base consiste à amener le système sur une hypersurface stable appelée *surface de glissement*, et d'assurer ensuite la convergence sur la surface de glissement vers le point d'équilibre désiré. La commande par mode glissant a été appliquée dans quelques travaux pour le contrôle des multi-rotors.

Dans [48], une synthèse des lois de commande par mode glissant a été présentée. L'objectif était de contrôler le comportement d'un drone quadri-rotor pour réaliser un suivi de trajectoires désirées de la position 3D et de l'angle de lacet. Les auteurs se sont également intéressés au développement d'un observateur non-linéaire afin de pouvoir estimer les états non mesurés et les effets des perturbations externes. La stratégie de contrôle proposée a été testée en simulation. Les résultats obtenus ont démontré une performance moyenne de vitesse de convergence et de dépassement des réponses du système. Le contrôleur proposé dans [49] est conçu en utilisant l'approche de mode glissant pour contrôler un quadri-rotor en présence de perturbations externes et d'une panne partielle d'un actionneur. Les résultats obtenus en simulation ont prouvé la capacité du contrôleur à effectuer les tâches de décollage, vol stationnaire et atterrissage. Le cas de perte partielle de performance d'un

rotor (moitié de sa poussée) a été considéré en vol stationnaire, et le contrôleur proposé a démontré sa performance pour assurer un atterrissage en toute sécurité.

Dans [50], la synthèse des lois de commande basée sur la méthode de mode glissant pour la stabilisation de l'attitude et l'altitude d'un octo-rotor a été présentée. Le cas de l'incertitude de la masse du drone a été considéré dans les simulations numériques. Le contrôleur proposé a démontré sa performance en rejet de perturbation. Une comparaison entre les performances de différentes techniques de commande non-linéaires (Back-stepping, mode glissant et saturations imbriquées) a été présentée dans [51, 52]. L'étude présentée dans [51] porte sur la comparaison des trois algorithmes de contrôle pour la stabilisation de l'attitude d'un quadri-rotor en vol stationnaire. Tous les algorithmes ont garanti que les angles du véhicule restent très proches des valeurs souhaitées. Les résultats expérimentaux ont montré que l'approche de contrôle par saturations imbriquées est la stratégie la plus appropriée pour le système considéré. Elle a assuré un comportement plus fluide du véhicule et a permis de réduire la consommation d'énergie par rapport aux deux autres contrôleurs. Les performances de l'approche par mode glissant ont été comparées dans [52] à celles du Back-stepping pour la stabilisation des angles d'un quadri-rotor appelé OS4. Le contrôleur par mode glissant a fourni des résultats moyens. Cependant, le contrôleur Back-stepping a prouvé la capacité de contrôler les angles d'orientation en présence de perturbations relativement élevées.

L'inconvénient essentiel de la commande par mode glissant est le comportement dynamique particulier, autour de la surface de glissement, appelé phénomène de réticence due à la fonction discontinue "sign" dans la loi de commande. Pour résoudre ce problème, différentes solutions peuvent être envisagées. La fonction discontinue dans les lois de commande [51] a été remplacée par une fonction $\tanh(\cdot)$ qui est une approximation lisse de la fonction $\text{sign}(\cdot)$. Dans [49] et [50], les auteurs ont proposé de remplacer la fonction "sign" par une fonction de saturation pour éliminer cet inconvénient.

- **Commande prédictive**

Le principe de base de la commande prédictive est d'utiliser un modèle explicite du système dans l'algorithme de commande afin de prédire le comportement futur en sortie. Les lois de commande se calculent en résolvant des problèmes d'optimisation avec contraintes pour minimiser l'erreur de suivi sur un intervalle de temps fini appelé *horizon de prédiction*. Dans la robotique aérienne, l'avantage de la commande prédictive pour gérer les contraintes posées sur les variables d'état et de sortie du système (plages de fonctionnement) et les variables de commande (contraintes sur l'amplitude) a permis à cette commande d'être très utilisée dans la pratique pour des suivis de références [53, 54] et des vols en formation [26, 55].

Trois stratégies de contrôle utilisant la commande prédictive pour le suivi des trajectoires ont été présentées dans [14] pour les multi-rotors. Les différentes étapes de concep-

tion et d'intégration des contrôleurs proposés dans le système ROS (Robot Operating System) ont été détaillées dans cette référence, et divers conseils d'implémentation et suggestions pratiques ont été fournis. Les contrôleurs proposés ont été implémentés à bord de deux systèmes multi-rotors : un quadri-rotor (Hummingbird) et un hexa-rotor (Firefly) de l'Ascending Technologies. Les résultats expérimentaux obtenus ont validé les contrôleurs proposés et les performances pour effectuer des suivis de trajectoires agressives ont été démontrées.

Dans [56], un contrôleur basé sur la commande prédictive a été conçu et appliqué expérimentalement sur un hexa-rotor. L'objectif était de contrôler l'attitude du véhicule et les tests expérimentaux ont montré des résultats prometteurs. La commande prédictive basée sur un modèle linéaire d'un drone de type tri-rotor a été utilisée dans [57] pour la conception d'un contrôleur de la boucle interne (attitude). Les auteurs visaient à réaliser un suivi de trajectoire par contrôleur à structure en cascade dont la boucle externe (position) est assurée par un PID. Le contrôleur proposé a produit des commandes efficaces pour les références données en attitude, et fourni un suivi de trajectoire satisfaisant.

La résolution des problèmes d'optimisation dans l'algorithme de la commande prédictive exige une puissance de calcul élevée. Cela peut représenter un obstacle lors de l'implémentation à bord des contrôleurs développés pour des applications en temps réel. Les auteurs dans [58] ont proposé un contrôleur efficace (nommé en anglais *eMPC* pour "efficient Model Predictive Control") en modifiant la commande prédictive standard dans le but d'avoir un contrôleur applicable à un drone quadri-rotor : le Qball-X4 disponible au laboratoire NAVL de l'Université Concordia. La réduction de l'exigence de calcul a été obtenue principalement par l'utilisation d'un modèle linéaire pour la prédiction des sorties du véhicule et, un nombre réduit de points de prédiction qui ne sont pas uniformément espacés le long de l'horizon de prédiction [59]. En raison de la capacité de calcul limitée disponible à bord du drone Qball-X4, il a été constaté que le test expérimental de la commande prédictive standard ne peut pas être effectué avec succès. Des tests de vol réussis pour le suivi d'une trajectoire rectangulaire avec le contrôleur *eMPC* ont prouvé l'efficacité du contrôleur proposé.

Dans [60], les auteurs se sont intéressés à l'utilisation d'un quadri-rotor autonome pour larguer une charge utile d'une certaine hauteur. Cette application peut correspondre à une mission de recherche, sauvetage ou livraison des subsistances et de biens dans des environnements dangereux ou difficiles à atteindre par l'humain. Pour ce scénario, deux techniques de contrôle ont été utilisées et appliquées expérimentalement au quadri-rotor Qball-X4 : un PID basé sur une méthode de séquençage de gain et un contrôleur basé sur la commande prédictive. Les tests expérimentaux ont été menés avec succès pour le largage de la charge utile montrant l'efficacité des deux contrôleurs proposés.

Il existe d'autres travaux de recherche dans la littérature proposant des contrôleurs, basés sur la commande prédictive - en configuration autonome ainsi qu'en combinaison avec

d'autres techniques de contrôle - implémentés aux multi-rotors pour différentes applications [61–66].

1.2.3 Contrôleurs intelligents basés sur l'apprentissage

Dans cette catégorie de contrôleurs, la connaissance du modèle dynamique du véhicule n'est pas forcément requise, tandis que certaines données de vol sont nécessaires pour entraîner le système. Les contrôleurs basés sur la logique floue et les réseaux de neurones sont les plus utilisés pour les drones multi-rotors.

- **Commande à base de la logique floue**

La commande à base de la logique floue a été appliquée avec succès pour le contrôle des multi-rotors. Les variables d'entrée de l'algorithme de commande floue sont essentiellement les consignes, les sorties du système et l'ensemble des connaissances et des règles décrivant le comportement du véhicule. Dans [67], un contrôleur flou adaptatif robuste a été proposé pour répondre au problème de stabilisation du quadri-rotor avec une charge utile inconnue et soumis à une perturbation sinusoïdale du vent. Un contrôleur flou intelligent a été également appliqué dans [68] pour contrôler la position et l'orientation d'un quadri-rotor. Les résultats de simulation ont permis de valider l'approche. Cependant, le réglage des gains du contrôleur a été fait par la méthode *essai-erreur*. Une méthode efficace de réglage des gains du contrôleur est donc nécessaire.

- **Commande à base des réseaux de neurones**

Divers travaux de recherche existent dans la littérature sur l'utilisation des systèmes neuronaux pour l'identification, l'estimation et le contrôle des drones multi-rotors. Le schéma de conception d'une commande en utilisant les réseaux de neurones est à l'origine inspiré du fonctionnement des neurones biologiques. Dans [69], un contrôleur adaptatif basé sur les réseaux de neurones pour stabiliser un drone quadri-rotor avec une erreur de modélisation et une perturbation considérable du vent. Les auteurs dans [70], ont proposé un contrôleur non linéaire pour un drone quadri-rotor en utilisant les réseaux de neurones et le contrôle par retour de sortie. Les réseaux de neurones ont été utilisés pour apprendre la dynamique complète du véhicule, contrôler ses six DDL et estimer les vitesses translationnelles et angulaires. Il existe dans la littérature d'autres travaux proposant et appliquant des contrôleurs basés sur l'apprentissage aux drones multi-rotors [71–75].

1.3 Planification de chemins et évitement d'obstacles

Avant de donner un aperçu des travaux existants dans la littérature sur le problème de la planification de chemins en 3D, nous donnons les deux définitions suivantes [76] :

- *Génération de trajectoire* : Un générateur de trajectoire permet de calculer différentes fonctions de mouvement (position de référence, cap de référence, etc.) qui sont physiquement possibles et satisfont la dynamique et les contraintes du drone. Ces fonctions de mouvement sont directement utilisables comme trajectoires de référence par le contrôleur de vol.
- *Planification de chemin* : Un planificateur de chemin consiste à utiliser des données de navigation accumulées et des informations a priori pour permettre au drone de trouver le trajet meilleur et le plus sûr pour atteindre une position / configuration cible ou pour accomplir une tâche spécifique.

Deux concepts du problème de planification de mouvement se distinguent alors : la planification de chemin et la planification de trajectoire. La différence majeure entre une trajectoire et un chemin réside dans la dimension temporelle. Dans le problème de planification de chemin, la variable *temps* n'est pas prise en compte. Un planificateur de chemin permet de déterminer l'ensemble des lieux géométriques en 3D par lesquels le drone est censé passer. La planification de trajectoire 3D fait référence à la détermination de comment se déplacer le long du chemin 3D à chaque instant *t* en considérant les contraintes dynamiques du drone. La planification de trajectoire dans un espace 3D est donc l'utilisation de la solution de planification de chemin et des contraintes dynamiques du drone pour trouver une solution plus appropriée paramétrée par la variable *temps*.

Dans cet état de l'art, nous nous intéressons au problème de la planification de chemin en 3D. Les algorithmes de planification de chemin sont classés en cinq catégories [77] :

- algorithmes à base des graphes,
- algorithmes à base de l'échantillonnage,
- algorithmes basés sur des modèles mathématiques,
- algorithmes bio-inspirés,
- algorithme hybrides.

La plupart des solutions de planification de chemin ont été développées à l'origine pour des problèmes 2D, puis étendues à la troisième dimension. Cette extension en 3D implique l'augmentation de la complexité de l'approche et du coût de calcul. Dans certains travaux, des solutions 2D ont été appliquées dans des environnements 3D en associant seulement la 3^{ème} dimension au chemin 2D planifié (méthodes dites 2.5D). Dans ces solutions, le drone est censé suivre le chemin planifié à une hauteur constante.

1.3.1 Algorithmes à base des graphes

Le A* est l'algorithme de recherche à base des graphes le plus populaire. Initialement publié par Hart et *al.* en 1968 [78], l'algorithme A* est basé sur l'algorithme de Dijkstra

[79] et a été largement implémenté dans la robotique et les jeux vidéo. Le A* est un algorithme permettant de tracer un chemin dirigé efficacement entre plusieurs points, appelés "nœuds". Contrairement à l'algorithme de Dijkstra qui ne considère - dans le processus de recherche - que le coût (réel) du trajet déjà traversé entre le nœud de départ et le nœud actuel, le A* considère de plus le coût estimé du trajet restant à faire pour atteindre le nœud cible à partir du nœud actuel. Cela est une amélioration apportée à l'algorithme de Dijkstra permettant de vérifier moins de nœuds dans le processus de recherche du chemin le plus court. Dans ce type d'algorithmes, l'environnement est représenté comme une **grille d'occupation (G.O.)**. L'algorithme A* dans sa version de base, appliqué à une G.O. 2D, explore les 8 nœuds de voisinage (dont l'angle entre eux est égal à 45 degrés) pour trouver le plus court chemin. De même, l'extension en 3D implique la représentation de l'environnement par une G.O. 3D dont les 26 nœuds voisins sont à explorer. Cependant, cet algorithme ne peut pas trouver le chemin dans tout angle de connexion au voisinage. Cela implique une solution sous-optimale représentée comme un chemin avec des changements de direction continus.

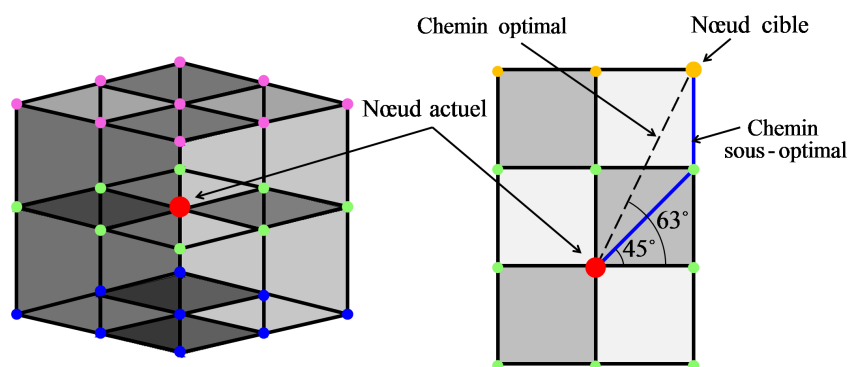


Figure 1.15. Connectivité de nœuds dans une grille d'occupation.

Dans le but de surmonter cette limite et / ou d'améliorer les résultats en termes de temps de traitement, de mémoire requise et de longueur de chemin, certaines variantes de l'algorithme A* ont été développées [80–82]. Pour les applications 3D, on peut citer [83–86]. Même si la solution a été améliorée, le chemin planifié a besoin de plus d'une phase de raffinements pour le rendre suffisamment lisse pour le suivi par véhicules aériens.

1.3.2 Algorithmes basés sur l'échantillonnage

Parmi les algorithmes basés sur l'échantillonnage les plus connus, on peut citer les algorithmes de diffusion incrémentale, connues aussi par les méthodes **Rapidly-exploring Random Tree (RRT)**, et les algorithmes de construction de réseaux probabilistes, appelés aussi méthodes **Probabilistic Road Map (PRM)**. Ces méthodes, basés sur l'échantillonnage de l'espace de configuration, sont initialement conçues pour les problèmes de planification de chemin 2D. Dans la méthode RRT initialement proposée par LaValle [87], un arbre est conçu en partant de la configuration initiale (position initiale du robot) qui est la racine

de l'arbre, jusqu'à atteindre la configuration finale. Cette version de base a été modifiée pour proposer une version bidirectionnelle [88] en diffusant simultanément deux arbres dans l'espace de configuration dont les racines sont la configuration initiale (départ) et la configuration finale (cible).

La méthode PRM [89] se compose de deux phases : une phase de construction et une phase de recherche. Dans la phase de construction, l'espace de configuration est échantillonné, et les configurations appartenant à l'espace libre sont conservées. Différentes méthodes d'échantillonnage ont été utilisées pour la génération de configurations dans l'espace libre. Un réseau (graphe), appelé aussi *road map* est conçu après en utilisant ces configurations dans l'objectif de capturer la connexité de l'espace libre. Dans la deuxième phase, les configurations initiale et finale sont connectées au graphe, et le chemin est obtenu en appliquant les méthodes de recherche du chemin le plus court tel que l'algorithme Dijkstra et A*.

L'inconvénient de ce type de méthodes basées sur l'échantillonnage est qu'elles renvoient des solutions non-optimales [90]. Dans le but d'améliorer la qualité de la solution, de nombreuses variantes de ces méthodes ont été proposées [90–94]. Pour le problème de planification de chemins pour les véhicules aériens, certaines versions ont été adaptées et appliquées aux environnements 3D. Dans [95], la méthode PRM a été utilisée pour planifier un chemin 3D dans un environnement urbain. La phase de construction a été faite hors ligne pour la génération du graphe. La phase de recherche du chemin 3D a été exécutée en ligne durant la mission en appliquant le A* sur le graphe généré précédemment hors ligne. Dans [96], les auteurs ont proposé une version améliorée de la méthode PRM pour traiter le problème de passages étroits (entre deux obstacles). Les expériences de planification de chemins ont été réalisées dans des environnements extérieurs réels. Les résultats ont démontré la capacité de la méthode proposée pour planifier des chemins de collision libre pour les drones.

La méthode RRT a été améliorée dans [97] pour développer un planificateur de chemins probabiliste robuste en temps réel pour drones. L'algorithme proposé fournit un moyen systématique pour traiter les incertitudes de l'environnement et gérer les obstacles dynamiques. La performance du planificateur de chemins proposé a été démontrée dans des environnements incertains simulés sans et avec des obstacles dynamiques. D'autres versions modifiées de la méthode RRT appliquées aux environnements 3D ont été proposées dans la littérature et validées par des études théoriques [98, 99].

1.3.3 Algorithmes basés sur des modèles mathématiques

Dans cette catégorie, l'environnement est décrit dans une forme mathématique et les contraintes sont traduites en équations différentielles. Ces algorithmes prennent en considération presque tous les facteurs et limitent étroitement la fonction de coût avec toutes les

inégalités et les équations afin d'obtenir une solution optimale. En raison de la complexité de calcul, cette catégorie d'algorithmes est appropriée pour fonctionner hors ligne. Différentes approches basées sur des modèles mathématiques ont été adoptées [100–102] et testées en simulation pour résoudre le problème de planification de chemins en 3D. Dans [100], le problème de planification de chemins 3D a été modélisé comme un problème de contrôle optimal. Dans le but d'obtenir un chemin optimal, le problème a été décomposé en deux parties. D'abord, un chemin 2D optimal est généré entre le point initial et le point final tout en satisfaisant les contraintes considérées ; puis le relief (la troisième dimension) est considéré tout le long du chemin 2D généré dans le but de déterminer le chemin 3D approximatif qui maximise la sécurité et satisfait d'autres contraintes. Les auteurs dans [101] ont présenté une approche de planification de trajectoire pour un drone quadri-rotor. L'approche proposée est basée sur la platitude différentielle et formule le problème de planification de trajectoire en tant que problème d'optimisation avec contraintes. La programmation linéaire en nombres entiers appartient également à cette catégorie, et a été appliquée dans [102] pour résoudre le problème de planification de chemins en 3D.

L'avantage de ces méthodes de résolution exacte est qu'elles fournissent des solutions optimales au problème de planification de chemins. Cependant, elles exigent une capacité de calcul élevée et un temps de calcul important. Cela représente un obstacle pour implémenter ces méthodes sur des systèmes drones réels en raison de la limite de la puissance de calcul disponible à bord. En outre, le temps de calcul important exigé par ces méthodes fait d'elles des méthodes adaptées pour des applications hors ligne. Le problème de planification de chemins 3D dans un environnement dynamique réel ne peut pas être résolu par cette classe de méthodes.

1.3.4 Algorithmes bio-inspirés

Ces méthodes, inspirées du monde biologique, ont été utilisées pour résoudre le problème de planification de chemin pour un ou plusieurs drones [103]. Cette catégorie inclut principalement deux groupes : les algorithmes évolutionnistes (génétiques ou d'intelligence en essaim) et les algorithmes à base des réseaux de neurones.

- **Les algorithmes génétiques**

Ce sont des algorithmes inspirés de la théorie d'évolution des espèces vivantes, appartenant donc à la famille des algorithmes évolutionnistes. Leur principe est basé sur les mécanismes de la sélection naturelle, l'enjambement (croisement) et la mutation de la génétique. Un algorithme génétique est une méta-heuristique de résolution des problèmes d'optimisation difficiles, pour lesquels les méthodes de résolution exactes ne sont pas adaptées. Les bases théoriques de ces algorithmes ont été développées à l'origine par John Holland [104]. Les algorithmes génétiques sont maintenant connus pour la résolution des problèmes de planification de chemin en 2D et 3D. Dans [105], les auteurs ont proposé deux planificateurs de chemin 3D en utili-

sant les algorithmes génétiques. Un planificateur hors ligne et un autre en ligne ont été conçus pour application dans un environnement connu et un autre inconnu. La solution a été donnée par une ou plusieurs courbes B-Spline produisant un chemin continu reliant les points de départ et de destination. L'efficacité des planificateurs fournissant des solutions *quasi optimales* a été prouvée par des tests selon différents scénarios. Un inconvénient des algorithmes génétiques est le coût de calcul qui entrave l'implémentation en temps réel. Des solutions de traitement en parallèle ont été proposées dans [106, 107] pour réduire le temps de calcul des algorithmes génétiques, afin d'obtenir un planificateur de chemin capable de recompiler un chemin en temps réel pour des missions dans des environnements incertains. Les auteurs dans [108] ont utilisé les algorithmes génétiques avec de nouveaux opérateurs dans l'objectif de maximiser une variable appelée *information collectée* depuis des *régions désirées*, tout en évitant de survoler des *régions interdites*.

- **Les algorithmes d'optimisation par essaim de particules (PSO)**

L'optimisation par essaim de particules est une méta-heuristique inventée par Kennedy et Eberhart en 1995 [109]. Elle s'inspire du comportement social de groupes d'animaux évoluant en essaim. Son principe se base sur la collaboration entre les différentes particules d'un essaim. Ces dernières s'entraident pour construire une solution à un problème complexe. De même que les algorithmes génétiques, l'optimisation par essaim de particules est un algorithme évolutionniste qui utilise une population de solutions candidates (chaque particule dans l'essaim est une solution potentielle dans l'espace de recherche) pour développer une solution optimale au problème posé. Cette méthode a été appliquée dans quelques travaux pour la planification de chemin 3D.

Dans [110], l'optimisation par essaim de particules a été utilisée pour la génération de chemins 3D sous forme de courbes B-Spline. La fonction objective a été définie en considérant trois objectifs : maximiser la sécurité (être suffisamment loin des obstacles), minimiser la consommation d'énergie (avoir le chemin le plus court possible) et rester le plus près possible des zones de reconnaissance. Cette approche a été testée par plusieurs scénarios simulés pour démontrer son efficacité.

Une comparaison entre les algorithmes génétiques et la méthode PSO pour la planification de chemin pour drone est présentée dans [111]. En se basant sur les résultats de simulation et en considérant un environnement 3D réel, il a été conclu que dans la plupart des cas considérés les algorithmes génétiques donnent de meilleures solutions que celles obtenues en utilisant la méthode PSO.

- **Les algorithmes d'optimisation par colonies de fourmis (ACO)**

Ce sont des méta-heuristiques faisant partie de la famille d'algorithmes d'intelligence en essaim utilisées pour la résolutions de problèmes complexes. Les algorithmes d'optimisation par colonies de fourmis sont inspirés du comportement des fourmis pour l'exploitation des ressources alimentaires. L'idée provient de l'action d'explo-

ration de l'environnement par une population de fourmis après la détection d'une source de nourriture (plus ou moins au hasard). Ces individus explorateurs ont la capacité de déposer des substances chimiques attractives (appelées *phéromones*) tout le long de leurs chemins parcourus entre le nid et la source alimentaire. Ces substances sont volatiles et donc disparaissent après un certain temps. Ce qui implique que les substances marquant les chemins longs deviennent moins attractives (ou disparaissent partiellement) avant d'être parcourus par d'autres individus et finissent par disparaître complètement. Cependant, les chemins courts deviennent de plus en plus renforcés (par les autres individus) et donc de plus en plus attractifs (grâce aux *phéromones*). Cela permet d'éliminer les chemins longs et converger vers le chemin le plus court (qui est le plus attractif). Cela est le principe de base de l'algorithme ACO initialement proposé en 1991 par Marco Dorigo dans le cadre de ses travaux de thèse de doctorat [112] publiée un an plus tard. À nos jours, peu de travaux ont été publiés dans la littérature proposant d'appliquer cet algorithme pour résoudre le problème de planification de chemin 3D pour drones, et uniquement des résultats de simulation y sont présentés [113, 114].

Au début des années 2000, de nouveaux algorithmes ont été développés en s'inspirant cette fois du comportement des abeilles. Un de ces algorithmes est l'optimisation par colonies d'abeilles (appelé BCO ou ABC en anglais) appartenant à la famille d'algorithmes d'intelligence en essaim. Une comparaison de différentes variantes de l'algorithme d'optimisation par colonies d'abeilles a été présentée dans [115] et la méthode dite *Balance-Evolution ABC* est considérée plus efficace en se basant sur des résultats de simulation.

Les algorithmes à base des réseaux de neurones ont été largement utilisés dans l'apprentissage, l'optimisation de la recherche, les problèmes de reconnaissance de formes et d'autres domaines d'application, grâce à leur capacité à fournir des solutions simples et optimales. Dans la planification de chemin 2D ou 3D, l'utilisation des algorithmes à base des réseaux de neurones (particulièrement en configuration autonome) semble relativement limitée. Une raison peut être leur nature nécessitant une phase d'apprentissage. La collecte des données et des informations requises pour entraîner les algorithmes à base de réseaux de neurones peut représenter un obstacle majeur à la conception de planificateurs de chemins basés sur ces algorithmes. Quelques travaux ont été proposés pour la planification de chemins en 2D [116–118], et en 3D [119]. Un autre travail peut être trouvé dans la référence [120].

Les algorithmes évolutionnistes étant des techniques méta-heuristiques sont des méthodologies de recherche puissantes et flexibles permettant de résoudre des problèmes pratiques difficiles. Ces algorithmes cherchent à produire des solutions de bonne qualité dans des délais de calcul raisonnables. La simplicité de ces méthodes (par rapport aux méthodes de résolution exacte qui utilisent des modèles mathématiques sophistiqués) permet de gagner en temps de calcul mais au détriment de l'optimalité de la solution. Un inconvé-

nient des algorithmes évolutionnistes est donc qu'ils ne peuvent pas garantir l'optimalité. La qualité de la solution dépend aussi de la population initiale (phase d'initialisation de l'algorithme). Pour surmonter ces limitations et d'autres des algorithmes de planification de chemin 3D, l'idée d'hybridation de deux algorithmes ou plus a été adoptée, faisant apparaître une nouvelle catégorie d'algorithmes appelés hybrides ou *multi-fusionnels*.

1.3.5 Algorithmes multi-fusionnels

Ces dernières années, les chercheurs ont trouvé dans l'hybridation une solution pour les limitations des algorithmes qu'ils utilisent. Elle consiste à combiner les caractéristiques de différents algorithmes, dans l'objectif de tirer profit des avantages respectifs. Cependant, l'algorithme hybride résultant risque de tenir également de leurs désavantages. Comme pour le contrôle des véhicules aériens, la planification de chemins 3D a aussi connu des méthodes hybrides afin d'améliorer les performances et la qualité des solutions obtenues. Dans la littérature, plusieurs exemples d'hybridations ont été proposés traitant le problème de planification de chemins pour drones. Dans [121], le résultat obtenu par l'application de l'algorithme génétique est utilisé pour entraîner l'algorithme de réseaux de neurones artificiels. L'objectif de cette combinaison est d'obtenir des solutions optimales tout en gagnant en temps de calcul. Un autre exemple est proposé dans [122]. Dans ce travail, les auteurs présentent un planificateur de chemins 3D basé sur les algorithmes génétiques en améliorant la phase d'initialisation de la population par le diagramme de Voronoï.

Dans [123], L'algorithme RRT* (variante de l'algorithme RRT) a été adopté pour planifier un chemin initial dans l'environnement 3D. L'algorithme d'optimisation par essais particuliers a été appliquée ensuite sur le résultat de l'algorithme RRT* pour une étape d'optimisation et de raffinement. Enfin dans [124], les auteurs proposent une méthode pour la planification de chemins pour un groupe de drones. Dans la première phase, le diagramme de Voronoï a été conçu. L'optimisation par essaim de particules a été utilisée dans la deuxième phase sur le diagramme de Voronoï pour planifier les chemins simultanés pour les drones simulés.

1.4 Conclusion

Dans ce chapitre, les différentes configurations des drones de type ADAV ont été introduites. Nous avons ensuite présenté un état de l'art sur les algorithmes de commande et les algorithmes de planifications de chemins appliqués pour les drones particulièrement à rotors multiples. Cette revue de la littérature révèle la diversité des méthodes utilisées pour assurer la commande et permettre la génération de chemins 3D. Le degré de simplicité, la facilité d'implémentation et la fiabilité varient d'une méthode à une autre. La nature de la mission à effectuer et les contraintes qui l'encadrent déterminent quelle méthode est plus appropriée.

Chapitre 2

Modélisation des drones multi-rotors

Sommaire

2.1	Introduction	27
2.2	Référentiels et matrices de passage	28
2.3	Modélisation cinématique	30
2.4	Modélisation dynamique	31
2.5	Modèle généralisé d'un drone N -rotor	33
2.5.1	Forces et moments aérodynamiques	33
2.5.2	Effets gyroscopiques	35
2.5.3	Dynamique de translation	36
2.5.4	Dynamique de rotation	37
2.6	Identification des paramètres du multi-rotor	43
2.7	Conclusion	44

2.1 Introduction

La modélisation mathématique d'un système physique a pour objectif d'établir l'ensemble des équations mathématiques décrivant son comportement. Cette étape est un élément indispensable pour tout travail de conception de lois de commande, détection des défauts, estimation de l'état, simulation, etc.

Deux approches sont utilisées dans la littérature pour la modélisation des drones : l'approche Lagrangienne et l'approche Newtonienne. Dans ce chapitre, nous nous intéressons à modéliser la dynamique des multi-rotors à configuration standard possédant deux plans de symétrie. En utilisant le formalisme d'Euler-Newton, nous proposons un modèle généralisé pour cette classe de drones [125]. Les multi-rotors à configurations originales ne sont pas considérés dans ce travail.

Un multi-rotor standard est un drone doté de N rotors dont $N/2$ sont en rotation en sens horaire et $N/2$ sont en rotation en sens anti-horaire. Ceux-ci sont montés à l'extrémité de

N bras de support dont l'angle entre eux est égal à $2\pi/N$. Nous définissons les hypothèses suivantes liées à la structure :

- La structure est rigide, de masse constante et parfaitement symétrique.
- Le centre de masse du système est exactement le centre de symétrie.
- Tous les rotors sont identiques et chacun tourne dans le sens opposé de celui du rotor voisin.

2.2 Référentiels et matrices de passage

Deux référentiels sont considérés dans la modélisation dynamique des multi-rotors, un global (fixe) et l'autre local (mobile). Soit $E = (O, \mathbf{e}_x, \mathbf{e}_y, \mathbf{e}_z)$ le référentiel inertiel fixe lié à la terre. La localisation choisie pour le référentiel E est souvent basée sur la position initiale du drone, mais peut aussi être arbitraire. Quant à son orientation, elle peut être choisie suivant les directions cardinales, en fonction de l'orientation initiale du drone ou encore arbitraire. Dans le référentiel inertiel E représenté sur la figure (2.1), \mathbf{e}_x est orienté vers le nord, \mathbf{e}_y vers l'ouest et \mathbf{e}_z vers le haut. Soit $B = (O', \mathbf{b}_x, \mathbf{b}_y, \mathbf{b}_z)$ le référentiel mobile associé au drone dont l'origine O' se situe au centre de masse du drone. Comme on peut le constater de la figure (2.1), E et B sont des référentiels orthonormés directs (cordonnées orientées "main droite").

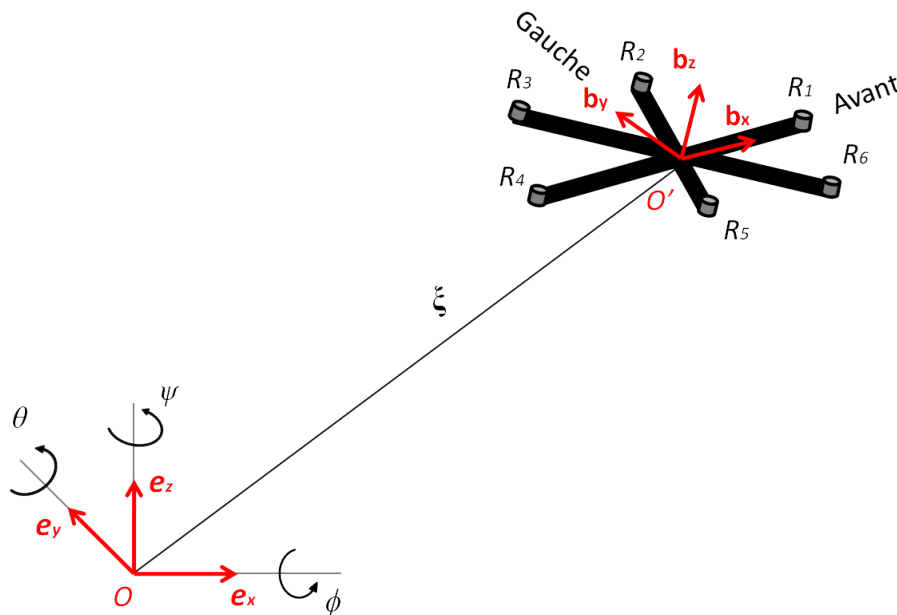


Figure 2.1. Drone multi-rotor dans le référentiel inertiel fixe.

Le positionnement du multi-rotor par rapport au référentiel E comporte six DDL (3 rotations indépendantes et 3 translations indépendantes). On doit alors définir trois variables de position et trois variables d'angles. La position du référentiel B associé au multi-rotor par rapport au référentiel inertiel E exprimée dans E est notée $\xi = (x, y, z)^T$ où x , y et z

sont respectivement les positions longitudinale, latérale et verticale. L'orientation du référentiel B par rapport au référentiel E exprimée dans E est donnée en fonction des angles d'Euler : roulis ϕ , tangage θ et lacet ψ représentés sur la figure (2.2).

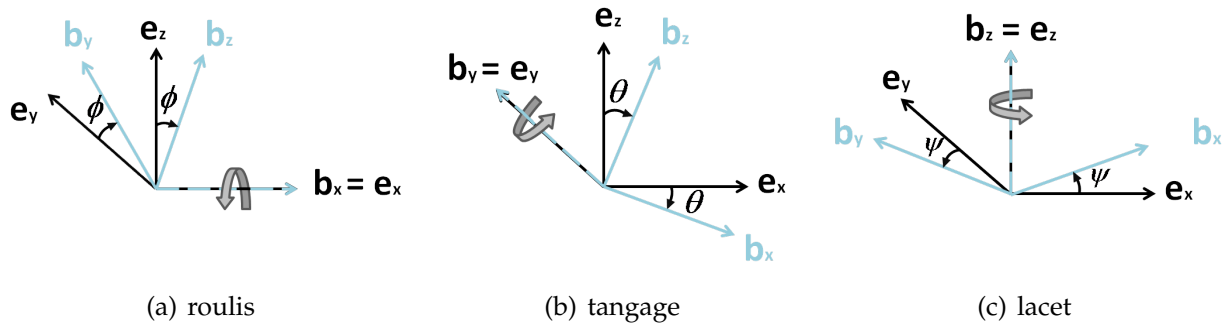


Figure 2.2. Angle d'Euler.

- **Mouvement de roulis** : rotation autour de l'axe e_x

$$\begin{cases} \mathbf{b}_x = \mathbf{e}_x \\ \mathbf{b}_y = c_\phi \mathbf{e}_y - s_\phi \mathbf{e}_z \\ \mathbf{b}_z = s_\phi \mathbf{e}_y + c_\phi \mathbf{e}_z \end{cases} \Rightarrow \mathbf{R}_\phi = \begin{bmatrix} 1 & 0 & 0 \\ 0 & c_\phi & -s_\phi \\ 0 & s_\phi & c_\phi \end{bmatrix} \quad (2.1)$$

avec $c_x = \cos x$ et $s_x = \sin x$.

- **Mouvement de tangage** : rotation autour de l'axe e_y

$$\begin{cases} \mathbf{b}_x = c_\theta \mathbf{e}_x + s_\theta \mathbf{e}_z \\ \mathbf{b}_y = \mathbf{e}_y \\ \mathbf{b}_z = -s_\theta \mathbf{e}_x + c_\theta \mathbf{e}_z \end{cases} \Rightarrow \mathbf{R}_\theta = \begin{bmatrix} c_\theta & 0 & s_\theta \\ 0 & 1 & 0 \\ -s_\theta & 0 & c_\theta \end{bmatrix} \quad (2.2)$$

- **Mouvement de lacet** : rotation autour de l'axe e_z

$$\begin{cases} \mathbf{b}_x = c_\psi \mathbf{e}_x - s_\psi \mathbf{e}_y \\ \mathbf{b}_y = s_\psi \mathbf{e}_x + c_\psi \mathbf{e}_y \\ \mathbf{b}_z = \mathbf{e}_z \end{cases} \Rightarrow \mathbf{R}_\psi = \begin{bmatrix} c_\psi & -s_\psi & 0 \\ s_\psi & c_\psi & 0 \\ 0 & 0 & 1 \end{bmatrix} \quad (2.3)$$

Les matrices \mathbf{R}_ϕ , \mathbf{R}_θ et \mathbf{R}_ψ sont des matrices orthogonales représentant chacun une rotation plane dans l'espace 3D. La transformation qui permet de passer de l'orientation du référentiel associé au drone B vers l'orientation du référentiel lié à la terre E est donnée par la matrice de rotation $\mathbf{R} = \mathbf{R}_\psi \mathbf{R}_\theta \mathbf{R}_\phi$.

$$\mathbf{R} = \begin{bmatrix} c_\theta c_\psi & s_\phi s_\theta c_\psi - c_\phi s_\psi & c_\phi s_\theta c_\psi + s_\phi s_\psi \\ c_\theta s_\psi & s_\phi s_\theta s_\psi + c_\phi c_\psi & c_\phi s_\theta s_\psi - s_\phi c_\psi \\ -s_\theta & s_\phi c_\theta & c_\phi c_\theta \end{bmatrix} \quad (2.4)$$

À noter que : $-\frac{\pi}{2} < \phi < \frac{\pi}{2}$, $-\frac{\pi}{2} < \theta < \frac{\pi}{2}$, $-\pi \leq \psi \leq \pi$.

$\mathbf{R} \in \mathbb{R}^{3 \times 3}$ est une matrice orthogonale de déterminant égal à 1. Elle appartient au *groupe spécial orthogonal* $SO(3)$ formé de toutes les matrices 3×3 qui vérifient les propriétés suivantes :

$$\begin{cases} \mathbf{R} \cdot \mathbf{R}^T = \mathbf{R}^T \mathbf{R} = \mathbf{I}_3 \\ \det(\mathbf{R}) = 1 \end{cases} \quad (2.5)$$

où \mathbf{R}^T est la matrice transposée de \mathbf{R} , et \mathbf{I} est la matrice d'identité.

Ainsi, $\mathbf{R}^{-1} = \mathbf{R}^T$ est la matrice de rotation qui permet le passage de l'orientation du référentiel E vers l'orientation du référentiel B .

2.3 Modélisation cinématique

La modélisation cinématique permet d'établir les vitesses de translation et de rotation du véhicule exprimées dans le référentiel inertiel fixe E en fonction de ses vitesses exprimées dans le référentiel local B . Dans cette section, nous présentons les équations cinématiques de translation et de rotation.

Soient :

- $\dot{\boldsymbol{\xi}} = (\dot{x}, \dot{y}, \dot{z})^T$: la vitesse linéaire du véhicule exprimée dans le référentiel E .
- $\mathbf{V} = (u, v, w)^T$: la vitesse linéaire du véhicule exprimée dans le référentiel B .
- $\dot{\boldsymbol{\eta}} = (\dot{\phi}, \dot{\theta}, \dot{\psi})^T$: la vitesse angulaire du véhicule exprimée dans le référentiel E .
- $\boldsymbol{\Omega} = (p, q, r)^T$: la vitesse angulaire du véhicule exprimée dans le référentiel B .

La dérivée de la matrice de rotation $\mathbf{R}(\boldsymbol{\eta})$ est donnée telle que :

$$\dot{\mathbf{R}} = \mathbf{R} \, sk(\boldsymbol{\Omega}) \quad (2.6)$$

avec $sk(\boldsymbol{\Omega})$ est la matrice antisymétrique, appelée également matrice de préproduit vectoriel, associée au vecteur $\boldsymbol{\Omega}$, telle que :

$$sk(\boldsymbol{\Omega}) = \begin{bmatrix} 0 & -r & q \\ r & 0 & -p \\ -q & p & 0 \end{bmatrix} \quad (2.7)$$

En multipliant les deux côtés de l'équation (2.6) par \mathbf{R}^T , on obtient :

$$sk(\boldsymbol{\Omega}) = \mathbf{R}^T \dot{\mathbf{R}} \quad (2.8)$$

La matrice $\mathbf{R}^T \dot{\mathbf{R}}$ a comme expression :

$$\mathbf{R}^T \dot{\mathbf{R}} = sk \left(\begin{bmatrix} \mathbf{R}_\phi^T & \mathbf{R}_\theta^T \\ 0 \\ \dot{\psi} \end{bmatrix} \right) + sk \left(\begin{bmatrix} \mathbf{R}_\phi^T \\ \dot{\theta} \\ 0 \end{bmatrix} \right) + sk \left(\begin{bmatrix} \dot{\phi} \\ 0 \\ 0 \end{bmatrix} \right) \quad (2.9)$$

Ainsi, on en déduit l'expression du vecteur Ω comme suit :

$$\begin{aligned}\Omega &= \begin{bmatrix} \dot{\phi} - s_\theta \dot{\psi} \\ c_\phi \dot{\theta} + s_\phi c_\theta \dot{\psi} \\ -s_\phi \dot{\theta} + c_\phi c_\theta \dot{\psi} \end{bmatrix} \\ &= \begin{bmatrix} 1 & 0 & -s_\theta \\ 0 & c_\phi & s_\phi c_\theta \\ 0 & -s_\phi & c_\phi c_\theta \end{bmatrix} \begin{bmatrix} \dot{\phi} \\ \dot{\theta} \\ \dot{\psi} \end{bmatrix}\end{aligned}\quad (2.10)$$

On peut exprimer la relation entre Ω et $\dot{\eta}$ par la matrice $S(\eta)$ telle que : $\Omega = S(\eta) \cdot \dot{\eta}$, et :

$$S(\eta) = \begin{bmatrix} 1 & 0 & -s_\theta \\ 0 & c_\phi & s_\phi c_\theta \\ 0 & -s_\phi & c_\phi c_\theta \end{bmatrix}\quad (2.11)$$

On note $Q(\eta)$ la matrice inverse de $S(\eta)$ donnée par :

$$Q(\eta) = \begin{bmatrix} 1 & s_\phi t_\theta & c_\phi t_\theta \\ 0 & c_\phi & -s_\phi \\ 0 & s_\phi / c_\theta & c_\phi / c_\theta \end{bmatrix}\quad (2.12)$$

avec : $t_\theta = \tan \theta$.

La singularité pour l'angle de tangage $\theta = \frac{\pi}{2} + k\pi$ ($k \in \mathbb{Z}$) est évidente. Cependant, cette singularité n'est généralement pas un problème. Physiquement, si l'angle de tangage θ (ou l'angle de roulis ϕ) atteint la valeur $\pm \frac{\pi}{2}$, c'est que le véhicule est orienté directement vers le haut ou vers le bas, et que l'axe vertical \mathbf{b}_z du véhicule devient parallèle au plan $(\mathbf{e}_x, \mathbf{e}_y)$. Cela bouleverse le comportement du véhicule et peut provoquer sa chute. Ces valeurs extrêmes sont donc à éviter et les angles de roulis et de tangage sont à définir tels que : $|\phi| < \frac{\pi}{2}$ et $|\theta| < \frac{\pi}{2}$.

Les équations cinétiques liant les vitesses du véhicule exprimées dans le référentiel E à ses vitesses locales V et Ω ont pour expressions :

$$\begin{cases} \dot{\xi} = \mathbf{R}(\eta) V \\ \dot{\eta} = \mathbf{Q}(\eta) \Omega \end{cases}\quad (2.13)$$

2.4 Modélisation dynamique

La modélisation dynamique s'appuie sur la modélisation cinématique. Elle permet d'établir les équations liant les forces et moments aux accélérations du véhicules. Un drone de type [ADAV](#) est un corps rigide de masse constante sur lequel on peut appliquer le principe

fondamental de la dynamique (la deuxième loi de Newton).

$$\begin{cases} \dot{\xi} &= \mathbf{R}\mathbf{V} \\ m\dot{\mathbf{V}} + \boldsymbol{\Omega} \wedge m\mathbf{V} &= \mathbf{F}_{ext} \\ \dot{\eta} &= \mathbf{Q}\boldsymbol{\Omega} \\ \mathbf{I}_r\dot{\boldsymbol{\Omega}} + \boldsymbol{\Omega} \wedge \mathbf{I}_r\boldsymbol{\Omega} &= \boldsymbol{\Gamma} \end{cases} \quad (2.14)$$

avec :

$m \in \mathbb{R}$: la masse totale du véhicule exprimée en kg.

$\mathbf{I}_r \in \mathbb{R}^{3 \times 3}$: la matrice d'inertie donnée dans le référentiel B .

$\mathbf{F}_{ext} \in \mathbb{R}^3$: le vecteur des forces exprimées dans le référentiel B .

$\boldsymbol{\Gamma} \in \mathbb{R}^3$: le vecteur des moments exprimés dans le référentiel B .

Pour le mouvement de translation, nous avons :

$$\dot{\xi} = \mathbf{R}(\eta) \mathbf{V} \quad \Rightarrow \quad \ddot{\xi} = \dot{\mathbf{R}}(\eta) \mathbf{V} + \mathbf{R}(\eta) \dot{\mathbf{V}} \quad (2.15)$$

Le vecteur des accélérations linéaires $\ddot{\xi}$ peut se réécrire tel que :

$$\ddot{\xi} = \mathbf{R}(\eta) \text{sk}(\boldsymbol{\Omega}) \mathbf{V} + \mathbf{R}(\eta) \dot{\mathbf{V}} \quad (2.16)$$

En remplaçant $\dot{\mathbf{V}}$ par son expression donnée dans l'équation (2.14), on obtient :

$$\begin{aligned} \ddot{\xi} &= \mathbf{R}(\eta) \text{sk}(\boldsymbol{\Omega}) \mathbf{V} + \mathbf{R}(\eta) \left(-\boldsymbol{\Omega} \wedge \mathbf{V} + \frac{1}{m} \mathbf{F}_{ext} \right) \\ &= \mathbf{R}(\eta) \text{sk}(\boldsymbol{\Omega}) \mathbf{V} - \mathbf{R}(\eta) \text{sk}(\boldsymbol{\Omega}) \mathbf{V} + \frac{1}{m} \mathbf{R}(\eta) \mathbf{F}_{ext} \\ &= \frac{1}{m} \mathbf{R}(\eta) \mathbf{F}_{ext} \end{aligned} \quad (2.17)$$

Le vecteur $\mathbf{R}(\eta) \mathbf{F}_{ext} \in \mathbb{R}^3$ donne les forces exprimées dans le référentiel E ($\mathbf{F}_{ext/E} = \mathbf{R}(\eta) \mathbf{F}_{ext}$).

$$\mathbf{F}_{ext/E} = \mathbf{F}_g + \mathbf{R} \mathbf{F}_{aero} \quad (2.18)$$

avec :

$\mathbf{F}_{aero} \in \mathbb{R}^3$: la somme des forces aérodynamiques exprimées dans le référentiel B .

$\mathbf{F}_g \in \mathbb{R}^3$: la force produite par la gravité et exprimée dans le référentiel E .

La gravité agit sur le véhicule dans la direction inverse de \mathbf{e}_z du référentiel E et génère la force \mathbf{F}_g telle que :

$$\mathbf{F}_g = (0, 0, -mg)^T = -mg \mathbf{e}_z \quad (2.19)$$

Ainsi, l'équation de la dynamique de translation du véhicule exprimée dans le référentiel E peut être écrite comme suit :

$$m \ddot{\xi} = -mg \mathbf{e}_z + \mathbf{R}(\boldsymbol{\eta}) F_{aero} \quad (2.20)$$

avec :

g : l'accélération de la pesanteur.

$\mathbf{R} F_{aero} \in \mathbb{R}^3$: la somme des forces aérodynamiques exprimées dans le référentiel E .

2.5 Modèle généralisé d'un drone N -rotor

Dans la section précédente, nous avons établi les équations de mouvement d'un corps rigide dans l'espace en utilisant le formalisme d'Euler-Newton. Cette forme invariante des lois de Newton et d'Euler est valable dans tout aéronef considéré comme corps rigide de masse constante.

Maintenant, nous modélisons les effets aérodynamiques et gyroscopiques spécifiques aux drones [ADAV](#) possédant deux plans de symétrie et N rotors. Les modèles les plus populaires de cette classe sont les quadri-rotors, les hexa-rotors et les octo-rotors. L'hélicoptère bi-rotor et le tri-rotor ont une structure et un principe de fonctionnement différents, et n'appartiennent pas à cette classe.

La proposition d'un modèle généralisé pour les multi-rotors ayant deux plans de symétrie est une des contributions de ce travail de thèse. L'objectif de cette partie est d'établir un modèle dynamique unifié, en fonction du nombre des rotors N . Cela permet essentiellement de généraliser les lois de commande, et de faciliter les implémentations pour différents systèmes multi-rotors dans des missions de vol en formation par exemple.

2.5.1 Forces et moments aérodynamiques

L'action de l'air sur les pales d'un actionneur en rotation produit deux types de forces : une force de poussée et une force de traînée. La vitesse angulaire des pales des actionneurs est beaucoup plus grande que la vitesse de déplacement. La force de poussée et la force de traînée sont donc supposées proportionnelles au carré de la vitesse de rotation de l'actionneur [19, 126].

- **Force de poussée (ou de portance)**

La force de poussée générée par chaque rotor est perpendiculaire au plan de rotation des pales, et orientée dans la direction de \mathbf{b}_z . En figure (2.3) sont montrées les forces de poussée générées par les actionneurs dans un drone hexa-rotor.

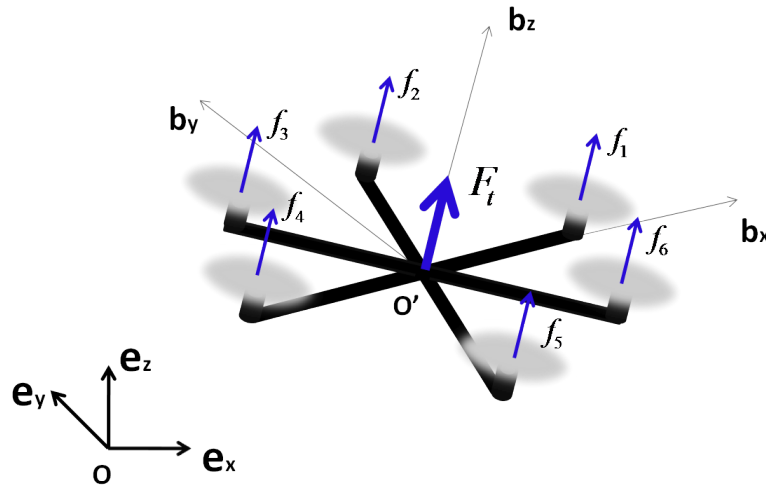


Figure 2.3. Forces de poussée générées par les rotors.

Tenant compte de l'hypothèse que tous les actionneurs sont identiques, l'expression de la force de poussée, notée $f \in \mathbb{R}$, générée par le rotor i est donnée comme suit :

$$f_i = b W_i^2 \mathbf{b}_z \quad (2.21)$$

avec :

$b > 0$: le coefficient de proportionnalité entre le carré de la vitesse de rotation du rotor et la force de poussée.

W_i : la vitesse de rotation du rotor i .

• Force de traînée

La force de traînée produite par la résistance de l'air sur les pales est parallèle au plan de rotation des pales, et orientée dans le sens opposé de la rotation des pales. Le moment de traînée $\tau \in \mathbb{R}$ (ou couple réactif), correspondant à la force de traînée, entraîne donc le véhicule en rotation autour de l'axe \mathbf{b}_z (mouvement de lacet) dans le sens opposé de W_i . Sur le schéma de la figure (2.4), le moment de traînée τ_i du rotor i est positif puisqu'il entraîne le véhicule en rotation en sens positif de lacet.

Le moment de traînée a pour expression :

$$\tau_i = -d W_i |W_i| \mathbf{b}_z \quad (2.22)$$

où $d > 0$ est le coefficient de proportionnalité entre le moment de traînée et le carré de la vitesse de rotation du rotor.

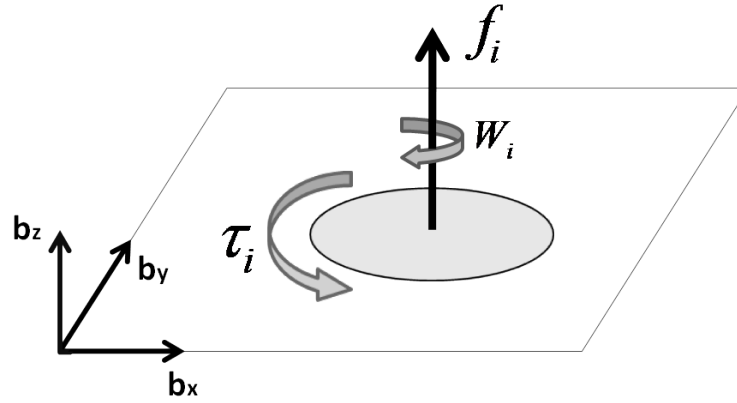


Figure 2.4. Illustration du moment de traînée généré par le rotor.

2.5.2 Effets gyroscopiques

L'effet gyroscopique correspond au phénomène de conservation de l'axe de rotation d'un objet en rotation. Chaque rotor est considéré comme un disque rigide en rotation autour de son axe vertical $\mathbf{r}_z = \mathbf{b}_z$ avec une vitesse angulaire W_i . L'axe de rotation du rotor se déplace (change de direction) avec les mouvements angulaires du véhicule N -rotor.¹ Le rotor exerce donc un couple (appelé couple gyroscopique) pour résister au changement de son axe de rotation.

Dans un multi-rotor standard, \mathbf{b}_z est l'axe de rotation de tout rotor i . Le vecteur du couple gyroscopique $\gamma \in \mathbb{R}^3$ généré par le rotor i a donc pour expression [126] :

$$\gamma_i = J_r(\boldsymbol{\Omega} \wedge \mathbf{b}_z)(-1)^{i+1} W_i \quad (2.23)$$

avec :

J_r : l'inertie du rotor autour de son axe de rotation exprimée en $kg.m^2$.

$J_r W_i \mathbf{b}_z$: le moment cinétique du rotor i .

En effet, il existe deux types des effets gyroscopiques appliqués au drone multi-rotor. Le premier est les effets gyroscopiques des rotors, donnés pour chaque rotor par le vecteur γ_i . Le deuxième type est les effets gyroscopiques du corps du véhicule. À noter que le terme $(\boldsymbol{\Omega} \wedge \mathbf{I}_r \boldsymbol{\Omega})$ de la dernière équation donnée dans (2.14) donne les trois composantes du vecteur des couples gyroscopiques du corps rigide, avec $\mathbf{I}_r \boldsymbol{\Omega}$ et le moment cinétique de ce dernier.

À présent, nous pouvons former le modèle dynamique unifié d'un drone N -rotor.

1. L'axe de rotation du rotor se déplace uniquement avec le roulis et le tangage. Le mouvement de lacet le conserve. Par conséquent, la troisième composante du vecteur des couples gyroscopiques générés par les rotors est nulle.

2.5.3 Dynamique de translation

L'équation différentielle du mouvement de translation donnée par (2.20) peut être réécrite comme suit :

$$\begin{bmatrix} \ddot{x} \\ \ddot{y} \\ \ddot{z} \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \\ -g \end{bmatrix} + \frac{1}{m} \begin{bmatrix} c_\theta c_\psi & s_\phi s_\theta c_\psi - c_\phi s_\psi & c_\phi s_\theta c_\psi + s_\phi s_\psi \\ c_\theta s_\psi & s_\phi s_\theta s_\psi + c_\phi c_\psi & c_\phi s_\theta s_\psi - s_\phi c_\psi \\ -s_\theta & s_\phi c_\theta & c_\phi c_\theta \end{bmatrix} \cdot F_{aero} \quad (2.24)$$

Dans la classe des N -rotors considérée, La force F_{aero} est la somme des forces de poussée f_i générées par les rotors. Notée F_t , elle est exprimée dans le référentiel associé au drone, en fonction du nombre des rotors N .

$$F_t = \sum_{i=1}^N f_i = b \sum_{i=1}^N W_i^2 \mathbf{b}_z \quad (2.25)$$

La force de poussée totale F_t est la seule entrée de commandes des actionneurs pour la dynamique de translation. Dans un vol stationnaire, elle permet de compenser la composante du poids du véhicule et le maintenir dans l'espace à une vitesses linéaires nulle ($\dot{\mathbf{z}} = 0$). L'augmentation de la force de poussée totale produit un déplacement vertical vers le haut. Sa diminution produit un déplacement vertical dans le sens opposé.

Nous pouvons écrire les équations de la dynamique de translation du système multi-rotor sous la forme suivante :

$$\begin{cases} \ddot{x} = \frac{1}{m} (\cos \phi \sin \theta \cos \psi + \sin \phi \sin \psi) F_t \\ \ddot{y} = \frac{1}{m} (\cos \phi \sin \theta \sin \psi - \sin \phi \cos \psi) F_t \\ \ddot{z} = -g + \frac{\cos \phi \cos \theta}{m} F_t \end{cases} \quad (2.26)$$

On note :

$$\begin{aligned} u_x &= \cos \phi \sin \theta \cos \psi + \sin \phi \sin \psi \\ u_y &= \cos \phi \sin \theta \sin \psi - \sin \phi \cos \psi \end{aligned} \quad (2.27)$$

Ces équations expriment les termes de couplage entre le déplacement (x, y) et l'orientation du véhicule. Nous revenons sur ce point dans le chapitre 3.

2.5.4 Dynamique de rotation

Dans l'équation de la dynamique de rotation d'un corps rigide dans l'espace donnée dans (2.14), la matrice d'inertie \mathbf{I}_r caractérise la difficulté à mettre le véhicule en mouvement de rotation. Elle est exprimée à l'origine du référentiel B telle que :

$$\mathbf{I}_r = \begin{bmatrix} I_{xx} & I_{xy} & I_{xz} \\ I_{yx} & I_{yy} & I_{yz} \\ I_{zx} & I_{zy} & I_{zz} \end{bmatrix} \quad (2.28)$$

Les quantités scalaires I_{xx} , I_{yy} et I_{zz} sont les moments d'inertie autour des axes \mathbf{b}_x , \mathbf{b}_y , \mathbf{b}_z respectivement dans le référentiel associé au véhicule. Les autres quantités sont les produits d'inertie par rapport aux plans $(\mathbf{b}_x, \mathbf{b}_y)$, $(\mathbf{b}_y, \mathbf{b}_z)$ et $(\mathbf{b}_x, \mathbf{b}_z)$. Tout comme J_r , les quantités de la matrice \mathbf{I}_r s'expriment en $kg.m^2$.

Les multi-rotors considérés sont des véhicules à structure admettant deux plans de symétrie orthogonaux : $(\mathbf{b}_x, \mathbf{b}_z)$ et $(\mathbf{b}_y, \mathbf{b}_z)$. Par conséquent, les produits d'inertie sont nuls, et la matrice d'inertie \mathbf{I}_r a pour expression :

$$\mathbf{I}_r = \begin{bmatrix} I_{xx} & 0 & 0 \\ 0 & I_{yy} & 0 \\ 0 & 0 & I_{zz} \end{bmatrix} \quad (2.29)$$

avec $I_{xx} = I_{yy}$.

Les moments extérieurs Γ s'exerçant sur le véhicule sont les moments des forces générées par les actionneurs (poussée et traînée) et les moments gyroscopiques.

$$\Gamma = \Gamma_1 + \Gamma_2 \quad (2.30)$$

avec :

$\Gamma_1 = (\Gamma_r, \Gamma_t, \Gamma_l)^T$: le vecteur des couples actifs résultants de la combinaison des forces de poussée f_i et des moments de traînée τ_i générés par les rotors.

$\Gamma_2 = (\Gamma_{g\phi}, \Gamma_{g\theta}, \Gamma_{g\psi})^T$: le vecteur des couples gyroscopiques des rotors.

Pour des petites valeurs d'angles loin des singularités, nous avons $\dot{\boldsymbol{\eta}} \approx \boldsymbol{\Omega}$. Ainsi, les équations différentielles de la dynamique de rotation du véhicule exprimée dans le référentiel E peuvent être écrites comme suit :

$$\begin{cases} \ddot{\phi} = \frac{(I_{yy} - I_{zz})}{I_{xx}} \dot{\theta} \dot{\psi} + \frac{\Gamma_r}{I_{xx}} - \frac{\Gamma_{g\phi}}{I_{xx}} \\ \ddot{\theta} = \frac{(I_{zz} - I_{xx})}{I_{yy}} \dot{\phi} \dot{\psi} + \frac{\Gamma_t}{I_{yy}} - \frac{\Gamma_{g\theta}}{I_{yy}} \\ \ddot{\psi} = \frac{(I_{xx} - I_{yy})}{I_{zz}} \dot{\phi} \dot{\theta} + \frac{\Gamma_l}{I_{zz}} - \frac{\Gamma_{g\psi}}{I_{zz}} \end{cases} \quad (2.31)$$

a) Couples actifs

Les couples actifs sont les couples induits de la somme des moments de poussée et de traînée produits par l'action de l'air sur les pales. Le vecteur des couples actifs a donc pour expression :

$$\mathbf{\Gamma}_1 = \sum_{i=1}^N (\vec{l}_i \wedge \vec{f}_i) + \sum_{i=1}^N \tau_i \quad (2.32)$$

avec :

N : le nombre des rotors que le drone possède.

\vec{l}_i : la position du centre du rotor i par rapport au référentiel B associé au véhicule.

f_i : la force de poussée générée par le rotor i .

τ_i : le moment de traînée généré par le rotor i .

Soient l la distance entre le centre de masse du drone et le centre de chaque rotor, et λ_i l'angle entre le bras du rotor i et l'axe \mathbf{b}_x du référentiel B , comme illustré sur la figure (2.5).

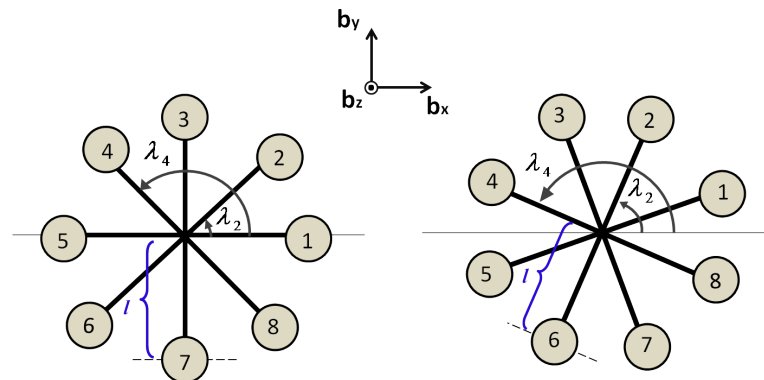


Figure 2.5. Représentation de l'angle λ des deuxième et quatrième rotors de l'octo-rotor à configuration '+' (à gauche) et configuration 'x' (à droite).

On obtient :

$$\begin{aligned} \mathbf{\Gamma}_1 &= \begin{pmatrix} \sum_{i=1}^N l \cos \lambda_i \\ \sum_{i=1}^N l \sin \lambda_i \\ 0 \end{pmatrix} \wedge \begin{pmatrix} 0 \\ 0 \\ f_i \end{pmatrix} + \begin{pmatrix} 0 \\ 0 \\ \sum_{i=1}^N \tau_i \end{pmatrix} \\ &\Rightarrow \mathbf{\Gamma}_1 = \begin{pmatrix} \sum_{i=1}^N l \sin \lambda_i f_i \\ -\sum_{i=1}^N l \cos \lambda_i f_i \\ \sum_{i=1}^N \tau_i \end{pmatrix} \end{aligned} \quad (2.33)$$

• **Couples actifs de roulis Γ_r et de tangage Γ_t**

Les couples actifs de roulis et de tangage permettent de faire tourner le multi-rotor autour des axes \mathbf{b}_x et \mathbf{b}_y respectivement. Leurs expressions sont données dans (2.33) telles que :

$$\begin{aligned} \Gamma_r &= l \sum_{i=1}^N \sin \lambda_i f_i \\ \Gamma_t &= -l \sum_{i=1}^N \cos \lambda_i f_i \end{aligned}$$

L'angle λ_i dépend de i le numéro du rotor et de N le nombre des rotors. Dans un multi-rotor standard, l'angle entre deux bras successifs est constant et égal à $\frac{2\pi}{N}$. Nous définissons donc λ_i comme suit :

$$\lambda_i = \begin{cases} (i-1) \frac{2\pi}{N} & \text{si '+' configuration} \\ \left(i - \frac{1}{2}\right) \frac{2\pi}{N} & \text{si 'x' configuration} \end{cases} \quad (2.34)$$

Nous obtenons ainsi les expressions des couples actifs de roulis et de tangage comme suit :

$$\begin{aligned} \Gamma_r &= bl \sum_{i=1}^N \sin \left((i-\epsilon) \frac{2\pi}{N} \right) W_i^2 \\ \Gamma_t &= bl \sum_{i=1}^N -\cos \left((i-\epsilon) \frac{2\pi}{N} \right) W_i^2 \end{aligned} \quad (2.35)$$

avec :

$$\epsilon = \begin{cases} 1 & \text{si '+' configuration} \\ \frac{1}{2} & \text{si 'x' configuration} \end{cases} \quad (2.36)$$

Pour comprendre le principe des mouvements de roulis et de tangage du multi-rotor, nous proposons les schémas de la figure (2.6) pour le cas d'un octo-rotor (vue de dessus). Les actionneurs représentés en gris foncés correspondent aux actionneurs ayant un moment de poussée positif, les actionneurs en pointillé ont un moment de poussée nul, et les actionneurs produisant un moment de poussée négatif sont représentés en gris clair. Les mouvements de roulis ϕ et de tangage θ sont assurés par la différence de la somme des moments positifs de poussée et la somme des moments négatifs. Il n'y a pas de rotation autour des axes \mathbf{b}_x et \mathbf{b}_y si les deux sommes sont égales.

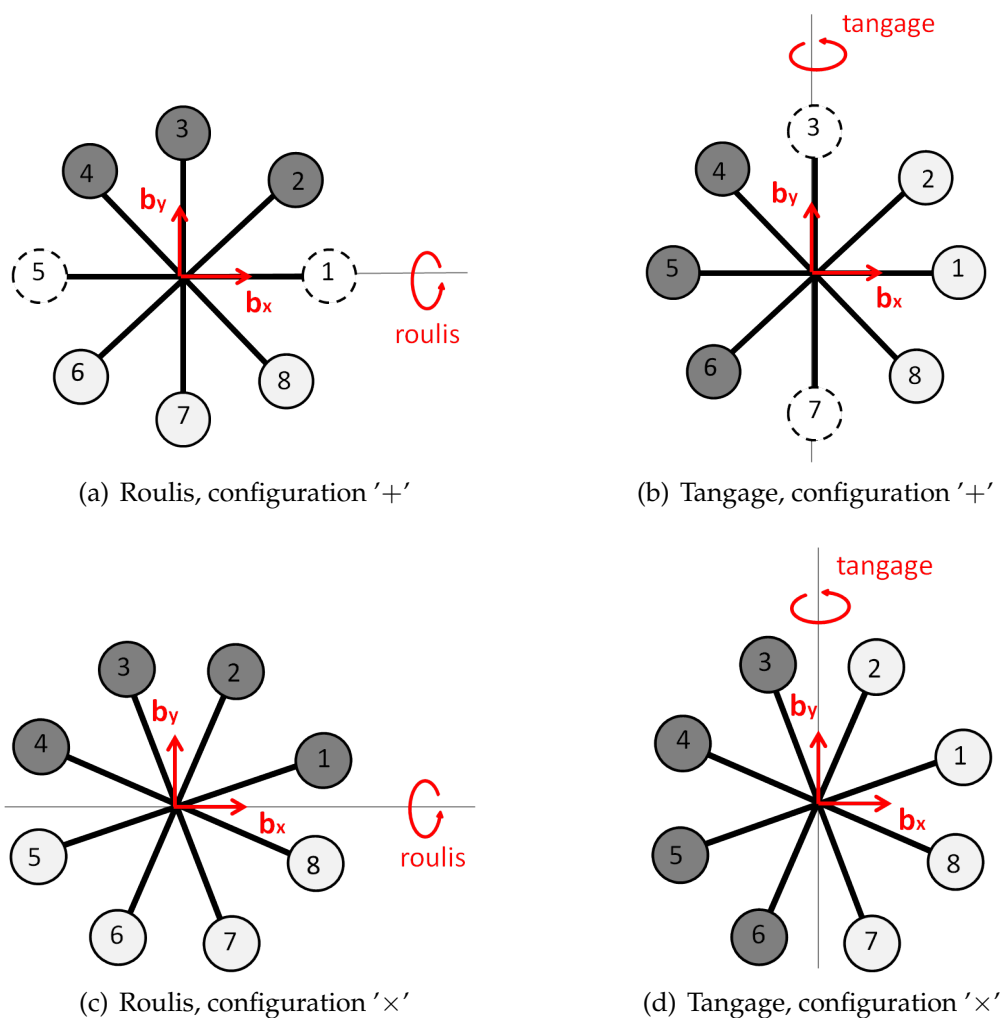


Figure 2.6. Principe de mouvements de roulis et de tangage.

On remarque que les rotors situés sur l'axe de roulis dans la configuration '+' ne peuvent pas jouer de rôle dans le mouvement de roulis. De même, les rotors situés sur l'axe de tangage ne peuvent pas jouer de rôle dans le mouvement de tangage. Cela représente une perte d'effort caractérisant les multi-rotors à configuration '+'.

- **Couple actif de lacet Γ_l**

Le couple actif de lacet permet de faire tourner le multi-rotor autour de l'axe \mathbf{b}_z . Son ex-

pression est donnée dans (2.33) telle que :

$$\Gamma_l = \sum_{i=1}^N \tau_i$$

Γ_l est produit par la combinaison des moments de traînée. Le signe du moment de traînée τ_i dépend du sens de rotation de l'actionneur i . Les rotors représentés en gris clair sur la figure (2.7) tournent dans le sens positif de lacet ($+\psi$), et produisent donc un moment (résistant) τ_i négatif. En revanche, les rotors représentés en gris foncé tournent dans le sens négatif de lacet ($-\psi$), et produisent un moment τ_i positif.

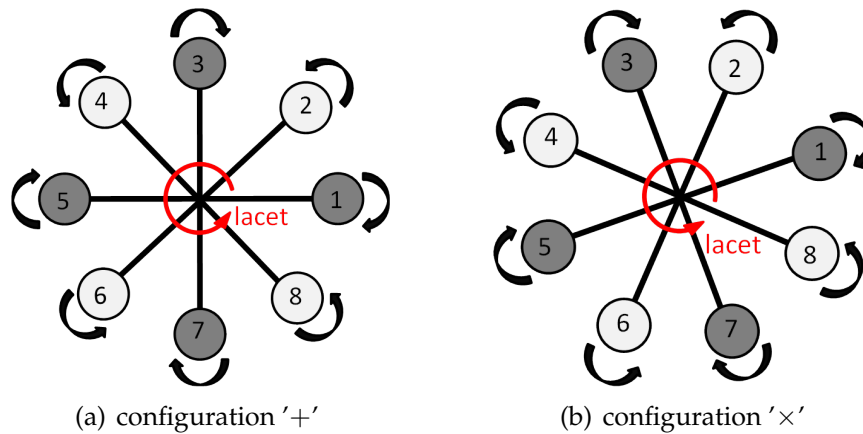


Figure 2.7. Principe de mouvement de lacet.

Par conséquent, le couple actif de lacet peut se réécrire comme suit :

$$\Gamma_l = d \left(\sum_{i=1}^{\frac{N}{2}} W_{2i-1}^2 - \sum_{i=1}^{\frac{N}{2}} W_{2i}^2 \right)$$

On en déduit :

$$\Gamma_l = d \sum_{i=1}^N (-1)^{i+1} W_i^2 \quad (2.37)$$

Le mouvement de lacet se génère en provoquant une différence de somme de vitesses de rotation des rotors en gris foncé (sens négatif de rotation) et ceux en gris clair (sens positif de rotation). À noter que pour produire un mouvement de lacet, il n'y a pas d'actionneur avec un moment résistant nul, que ce soit dans la configuration 'x' ou même '+'.²

Dans le cas où tous les rotors tournent à la même vitesse W_i , le couple de traînée résultant Γ_l , et donc l'accélération angulaire $\ddot{\psi}$ du véhicule multi-rotor, sont exactement égaux à zéro.²

2. Dans l'équation différentielle du mouvement de lacet donnée dans (2.31), le terme $\frac{(I_{xx} - I_{yy})}{I_{zz}} \dot{\phi} \dot{\theta}$ des effets gyroscopiques du corps du véhicule s'annule en raison de la symétrie des multi-rotors à configuration standard. Le couple $\Gamma_{g\psi}$ est aussi nul comme sera montré dans le paragraphe des couples gyroscopiques.

b) Couples gyroscopiques

Dans un drone N -rotor, le couple gyroscopique Γ_2 qui s'exerce sur le véhicule est la somme des N couples gyroscopiques générés par les rotors.

$$\Gamma_2 = \sum_{i=1}^N \gamma_i \quad \Rightarrow \quad \Gamma_2 = J_r (\boldsymbol{\Omega} \wedge \mathbf{b}_z) \sum_{i=1}^N (-1)^{i+1} W_i \quad (2.38)$$

Intuitivement, $\Gamma_2 = 0$ si tous les moteurs tournent à la même vitesse.

Le calcul vectoriel donne les trois composantes de Γ_2 comme suit :

$$\begin{aligned} \Gamma_{g\phi} &= J_r \dot{\theta} \sum_{i=1}^N (-1)^{i+1} W_i \\ \Gamma_{g\theta} &= -J_r \dot{\phi} \sum_{i=1}^N (-1)^{i+1} W_i \\ \Gamma_{g\psi} &= 0 \end{aligned} \quad (2.39)$$

Nous remplaçons les forces et les moments appliqués au multi-rotor par leurs expressions (2.25), (2.35), (2.37) et (2.39) établies en fonction de N , dans les équations différentielles de la dynamique de translation et de rotation. Nous obtenons ainsi le modèle dynamique généralisé à 6 DDL des drones multi-rotors ayant deux plans de symétrie ($\mathbf{b}_x, \mathbf{b}_z$) et ($\mathbf{b}_y, \mathbf{b}_z$) et N rotors, comme donné dans le système d'équations suivant :

$$\left\{ \begin{aligned} \ddot{x} &= \frac{b}{m} u_x \sum_{i=1}^N W_i^2 \\ \ddot{y} &= \frac{b}{m} u_y \sum_{i=1}^N W_i^2 \\ \ddot{z} &= -g + \frac{b}{m} \cos \phi \cos \theta \sum_{i=1}^N W_i^2 \\ \ddot{\phi} &= \frac{(I_{yy} - I_{zz})}{I_{xx}} \dot{\theta} \dot{\psi} + \frac{bl}{I_{xx}} \sum_{i=1}^N \sin \left((i - \epsilon) \frac{2\pi}{N} \right) W_i^2 - \frac{J_r}{I_{xx}} \dot{\theta} \sum_{i=1}^N (-1)^{i+1} W_i \\ \ddot{\theta} &= \frac{(I_{zz} - I_{xx})}{I_{yy}} \dot{\phi} \dot{\psi} - \frac{bl}{I_{yy}} \sum_{i=1}^N \cos \left((i - \epsilon) \frac{2\pi}{N} \right) W_i^2 + \frac{J_r}{I_{yy}} \dot{\phi} \sum_{i=1}^N (-1)^{i+1} W_i \\ \ddot{\psi} &= \frac{(I_{xx} - I_{yy})}{I_{zz}} \dot{\phi} \dot{\theta} + \frac{d}{I_{zz}} \sum_{i=1}^N (-1)^{i+1} W_i^2 \end{aligned} \right. \quad (2.40)$$

avec u_x et u_y sont donnés dans l'équation (2.27), et ϵ dans (2.36). Les constantes : m , I_{xx} , I_{yy} , I_{zz} , J_r , b , d et l sont les paramètres physiques du multi-rotor à déterminer.

A titre d'exemple, pour le cas d'un hexa-rotor à configuration '+' ($\epsilon = 1$), on obtient le

modèle dynamique comme suit [56] :

$$\left\{ \begin{array}{l} \ddot{x} = \frac{b}{m} u_x \sum_{i=1}^6 W_i^2 \\ \ddot{y} = \frac{b}{m} u_y \sum_{i=1}^6 W_i^2 \\ \ddot{z} = -g + \frac{b}{m} \cos \phi \cos \theta \sum_{i=1}^6 W_i^2 \\ I_{xx} \ddot{\phi} = (I_{yy} - I_{zz}) \dot{\theta} \dot{\psi} + bl \left(0, \frac{\sqrt{3}}{2}, \frac{\sqrt{3}}{2}, 0, -\frac{\sqrt{3}}{2}, -\frac{\sqrt{3}}{2} \right) \omega - J_r \dot{\theta} \omega_g \\ I_{yy} \ddot{\theta} = (I_{zz} - I_{xx}) \dot{\phi} \dot{\psi} + bl \left(-1, -\frac{1}{2}, \frac{1}{2}, 1, \frac{1}{2}, -\frac{1}{2} \right) \omega + J_r \dot{\phi} \omega_g \\ I_{zz} \ddot{\psi} = (I_{xx} - I_{yy}) \dot{\phi} \dot{\theta} + d \left(1, -1, 1, -1, 1, -1 \right) \omega \end{array} \right. \quad (2.41)$$

avec : $\omega = (W_1^2, W_2^2, \dots, W_6^2)^T$ et $\omega_g = W_1 - W_2 + W_3 - W_4 + W_5 - W_6$.

Le modèle unifié que nous proposons dans ce chapitre pour les multi-rotors standard peut être adapté à des configurations particulières telles que le drone X8 et l'octo-rotor à configuration V [50, 127].

2.6 Identification des paramètres du multi-rotor

Les valeurs numériques de la masse m , l'inertie du véhicule I_r et l'inertie du rotor J_r peuvent facilement être calculés ou mesurés. Les coefficients aérodynamiques de poussée b et de traînée d dépendent de plusieurs paramètres tels que : le nombre de pales, la densité de l'air, le rayon des pales, la longueur de la corde des pales, l'angle d'attaque, la forme du profil des pales, etc. Les déterminer théoriquement est donc difficile. Cependant, ils peuvent être identifiés expérimentalement sans s'occuper des valeurs numériques des paramètres dont ils dépendent.

À l'aide d'un dispositif de mesure (avec capteurs de vitesse, de force et de couple), on peut faire varier W_i la vitesse de rotation du rotor, et noter à chaque essai la valeur mesurée de la force et celle du couple. On représente ensuite les données mesurées en fonction du carré de la vitesse W_i , comme illustré sur la figure (2.8) [126]. Les coefficients directeurs des droites de régression $f = \sigma_1(W^2)$ (à gauche) et $\tau = \sigma_2(W^2)$ (à droite) sont respectivement les constantes b et d .

On peut également identifier le coefficient de traînée d en se basant sur l'équation différentielle du mouvement de lacet. Le processus expérimental consiste à poser le multi-rotor sur un support ne permettant qu'une seule rotation libre autour de son axe vertical z (mouvement de lacet). On a donc : $\phi(t) = 0$ et $\theta(t) = 0$. Pour générer un mouvement de lacet, on choisit d'entraîner les moteurs impairs M_{2i-1} (avec $i = 1, \dots, \frac{N}{2}$) à des vitesses connues.

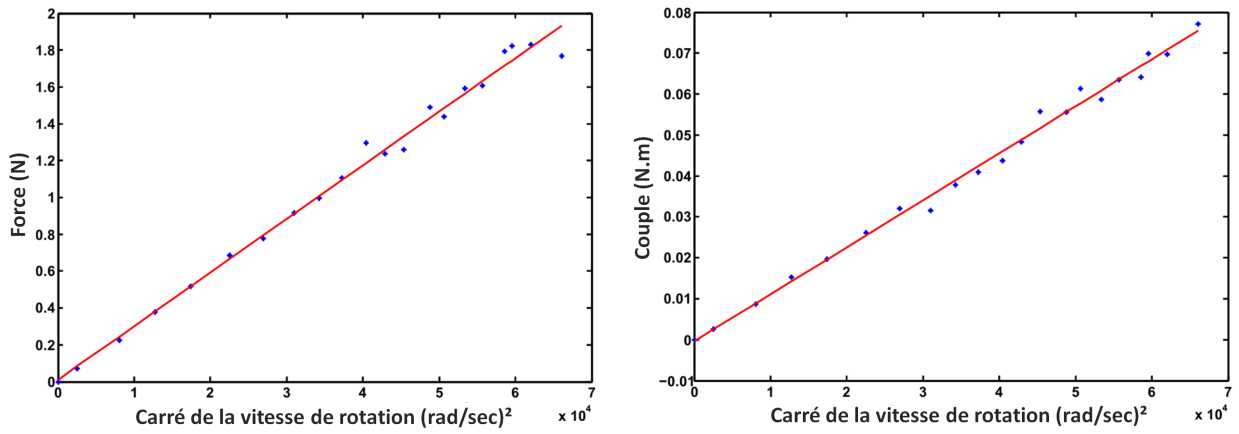


Figure 2.8. Identification expérimentale des coefficients de poussée (à gauche) et de traînée (à droite).

L'équation différentielle du mouvement de lacet donnée dans (2.40) peut alors être réécrite comme suit :

$$\begin{aligned}\ddot{\psi}(t) &= \frac{d}{I_{zz}} \sum_{i=1}^{\frac{N}{2}} W_{2i-1}^2 \\ &= \kappa \cdot d\end{aligned}\quad (2.42)$$

$$\text{avec : } \kappa = \frac{1}{I_{zz}} \sum_{i=1}^{\frac{N}{2}} W_{2i-1}^2.$$

En intégrant deux fois, on obtient $\psi(t) = \kappa \cdot d \frac{t^2}{2}$. Le coefficient d est donc donné par :

$$d = \frac{2\psi(t)}{\kappa t^2}\quad (2.43)$$

Enfin, on chronomètre le temps T mis par le multi-rotor pour effectuer un quart de tour en mouvement de lacet ($\psi = \pi/2$). On trouve la valeur numérique du coefficient de traînée d telle que : $d = \frac{\pi}{\kappa T^2}$.

2.7 Conclusion

Ce chapitre est consacré à la modélisation dynamique des multi-rotors. Nous avons utilisé le formalisme d'Euler-Newton pour établir les équations générales de la dynamique d'un véhicule considéré comme corps rigide dans l'espace. Ensuite, nous avons proposé un modèle dynamique unifié pour la classe des multi-rotors *ADAV* ayant deux plans de symétrie orthogonaux. Le modèle proposé est fonction du nombre des actionneurs du véhicule et valable pour une configuration '+' ou 'x'. Il peut également être légèrement modifié pour l'adapter à d'autres configurations. La conception de lois de commande est présentée dans le chapitre suivant en se basant sur ce modèle généralisé.

Chapitre 3

Commande et suivi de trajectoires

Sommaire

3.1	Introduction	45
3.2	Structure de la commande hiérarchique d'un drone multi-rotor	46
3.3	Commande hybride Back-stepping A.I. - PID	48
3.3.1	Boucle externe de la commande	48
3.3.2	Boucle interne de la commande	52
3.3.3	Étude comparative	53
3.4	Allocation des commandes sur les actionneurs	59
3.5	Résultats numériques	61
3.6	Étude en présence de pannes	66
3.7	Conclusion	76

3.1 Introduction

Ce chapitre s'intéresse au contrôle des multi-rotors pour la stabilisation et le suivi de trajectoires. La structure de commande en cascade est adoptée et la combinaison d'un contrôleur linéaire de type PID avec le contrôleur Back-stepping est proposée [128, 129]. La robustesse du PID pour la stabilisation de l'orientation de multi-rotors a été prouvée dans différents travaux dans la littérature. Nous appliquons donc le PID pour le contrôle de l'attitude du véhicule.

La méthode Back-stepping a été efficacement utilisée dans la littérature pour le contrôle de la position et l'orientation des multi-rotors. Nous proposons de renforcer d'une action intégrale les performances de l'algorithme Back-stepping afin de l'utiliser pour le contrôle de la position 3D du véhicule. Cette approche hybride nous permet de combiner les avantages du Back-stepping A.I. et du PID. Nous comparons les performances de l'approche proposée à celles de la méthode Back-stepping utilisée pour le contrôle des six DDL du système.

Dans ce chapitre, nous traitons également le problème d'allocation des commandes en se basant sur la matrice pseudo-inverse de Moore-Penrose. Dans cette partie, les développements mathématiques dans la forme généralisée pour un système multi-rotor font partie des contributions de ce travail. Nous nous intéressons aussi à l'étude de la réponse du système en cas de défaillance totale d'un rotor sans considérer une stratégie de contrôle tolérante aux défauts.

3.2 Structure de la commande hiérarchique d'un drone multi-rotor

Un système multi-rotor ayant N rotors possède N entrées de commande réelles qui sont les vitesses de rotation W_i ($i = 1, \dots, N$) des moteurs. En contrôlant les vitesses de rotation des moteurs, on peut contrôler efficacement le comportement du véhicule. Chaque variable de position ou de rotation du système possède une entrée de commande (force ou couple) donnée en fonction des W_i .

Considérons le système d'équations (2.26) (Chap. 2), on peut s'apercevoir que la force de poussée F_t est la seule entrée de commande de la dynamique de translation en 3D (longitudinale, latérale et verticale). Le vecteur Γ_1 des couples actifs ($\Gamma_r, \Gamma_t, \Gamma_l$) représente une entrée de commande complète pour la rotation du drone en 3D.

Notant $\mathbf{U} = (U_1 \ U_2 \ U_3 \ U_4)^T$ le vecteur des entrées de commande avec : $U_1 = F_t$, $U_2 = \Gamma_r$, $U_3 = \Gamma_t$ et $U_4 = \Gamma_l$. Le système (2.40) peut alors se réécrire en fonction des entrées de commande U_i comme suit :

$$\left\{ \begin{array}{l} \ddot{x} = \frac{1}{m} u_x U_1 \\ \ddot{y} = \frac{1}{m} u_y U_1 \\ \ddot{z} = -g + \frac{\cos \phi \cos \theta}{m} U_1 \\ \ddot{\phi} = \frac{(I_{yy} - I_{zz})}{I_{xx}} \dot{\theta} \dot{\psi} - \frac{J_r}{I_{xx}} \dot{\theta} \sum_{i=1}^N (-1)^{i+1} W_i + \frac{1}{I_{xx}} U_2 \\ \ddot{\theta} = \frac{(I_{zz} - I_{xx})}{I_{yy}} \dot{\phi} \dot{\psi} + \frac{J_r}{I_{yy}} \dot{\phi} \sum_{i=1}^N (-1)^{i+1} W_i + \frac{1}{I_{yy}} U_3 \\ \ddot{\psi} = \frac{(I_{xx} - I_{yy})}{I_{zz}} \dot{\phi} \dot{\theta} + \frac{1}{I_{zz}} U_4 \end{array} \right. \quad (3.1)$$

Les multi-rotors ont six sorties ($x, y, z, \phi, \theta, \psi$) et uniquement quatre entrées de commande indépendantes. Cet aspect connu sous le terme de *sous-actionnement* caractérise les drones multi-rotors. Ce n'est donc pas possible de contrôler séparément toute la dynamique d'un système multi-rotor.

En effet, les multi-rotors doivent s'incliner pour avancer. Le mouvement longitudinal et le mouvement latéral, illustrés sur la figure (3.1), sont donc obtenus en orientant le vecteur de poussée totale vers la direction désirée du mouvement. Cela implique que les mouvements de rotation et de translation sont couplés. La relation entre les mouvements de rotation et de translation est donnée par u_x et u_y dans l'équation (2.27).

- **Mouvement de roulis :**

$(\phi \neq 0, \theta = 0, \psi = 0) \Rightarrow (u_x = 0, u_y = -\sin \phi) \Rightarrow$ translation selon l'axe y .

- **Mouvement de tangage :**

$(\phi = 0, \theta \neq 0, \psi = 0) \Rightarrow (u_x = \sin \theta, u_y = 0) \Rightarrow$ translation selon l'axe x .

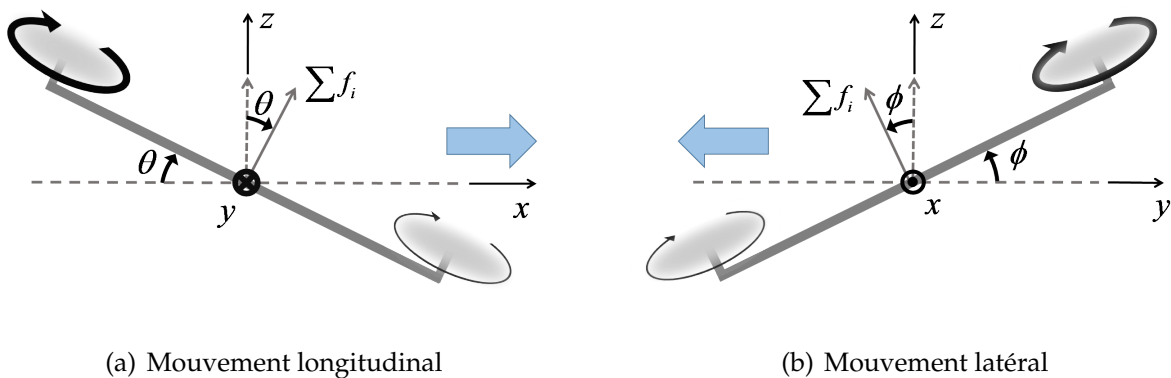


Figure 3.1. Illustration du couplage entre le mouvement horizontal et l'orientation de l'assiette d'un multi-rotor.

Le mouvement horizontal selon l'axe x est obtenu en faisant pivoter le multi-rotor sur lui-même autour de l'axe y (mouvement de tangage). De même, le mouvement horizontal selon l'axe y est obtenu en faisant pivoter le multi-rotor sur lui-même autour de l'axe x (mouvement de roulis). Le vol stationnaire est obtenu en forçant l'angle de roulis et l'angle de tangage à zéro.

Grâce à ce couplage entre les mouvements de rotation et de translation, la position (x,y) peut être contrôlée indirectement en contrôlant les angles de roulis et tangage.

Nous proposons une structure de commande hiérarchique pour le contrôle des multi-rotors comme le montre la figure (3.2). Pour cela, nous décomposons la dynamique du système en deux sous-systèmes : dynamique de translation et dynamique de rotation.

U_1 et U_4 sont respectivement les lois de commande de la position verticale z et l'angle de lacet ψ . Le bloc du contrôle de la position (x,y) permet de fournir les entrées u_x et u_y du bloc de calcul des valeurs désirées ϕ_d et θ_d pour les angles de roulis et de tangage. ϕ_d et θ_d sont ensuite introduites dans le bloc du contrôle des angles de roulis et tangage pour obtenir les lois de commande U_2 et U_3 dans le but final de contrôler la position (x,y) . En utilisant les expressions de u_x et u_y données dans (2.27), on obtient les références angulaires ϕ_d et θ_d comme suit :

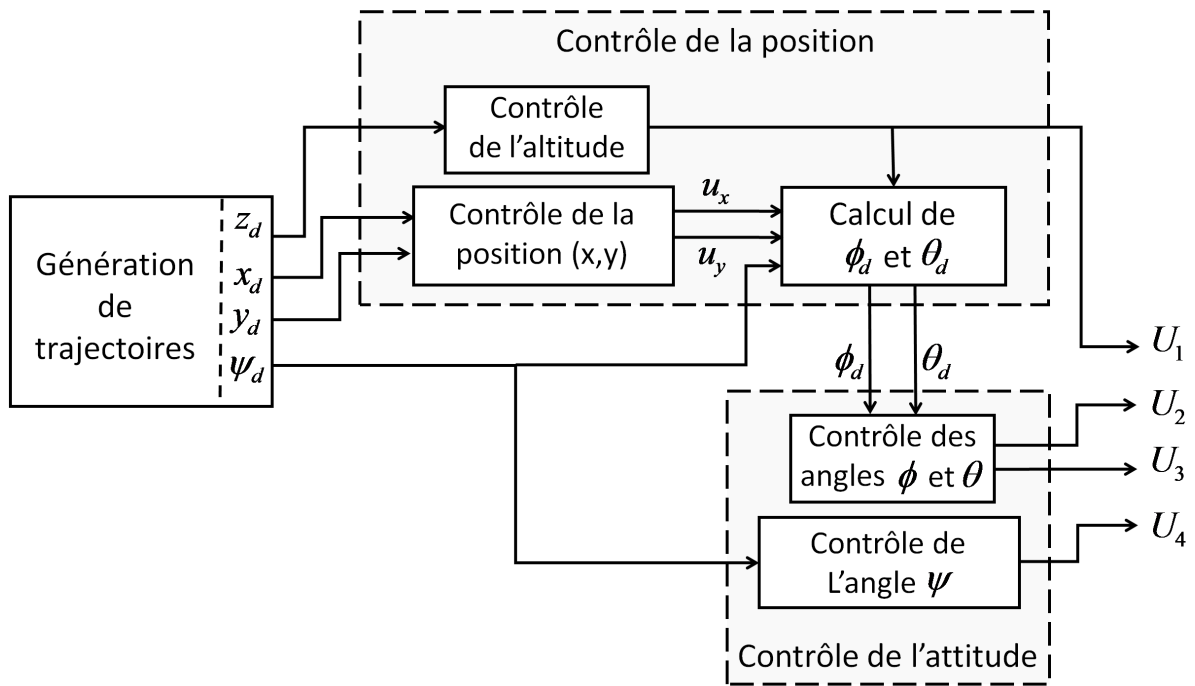


Figure 3.2. Structure de la commande hiérarchique d'un multi-rotor.

$$\begin{cases} \phi_d = \arcsin(u_x \cdot \sin \psi_d - u_y \cdot \cos \psi_d) \\ \theta_d = \arcsin\left(\frac{u_x \cdot \cos \psi_d + u_y \cdot \sin \psi_d}{\cos \phi_d}\right) \end{cases} \quad (3.2)$$

où ψ_d représente la référence de l'angle de lacet.

La singularité apparaît pour $\phi_d = \frac{\pi}{2} + k\pi$ ($k \in \mathbb{Z}$). L'ajout d'une fonction de saturation dans la boucle de commande de façon que cette singularité ne soit jamais atteinte résout le problème.

3.3 Commande hybride Back-stepping A.I. - PID

Nous proposons dans cette section une stratégie de commande hybride pour le contrôle de la dynamique du véhicule à six DDL. La stratégie de commande proposée est conçue dans une structure hiérarchique à deux boucles. La boucle externe assurant la commande de la position 3D en utilisant la méthode Back-stepping A.I. Tandis que, la boucle interne assure la commande de l'orientation en utilisant un contrôleur PID.

3.3.1 Boucle externe de la commande

Dans cette section, on s'intéresse à l'élaboration des lois de commande pour la stabilisation de la position (x, y, z) en appliquant la méthode Back-stepping A.I.

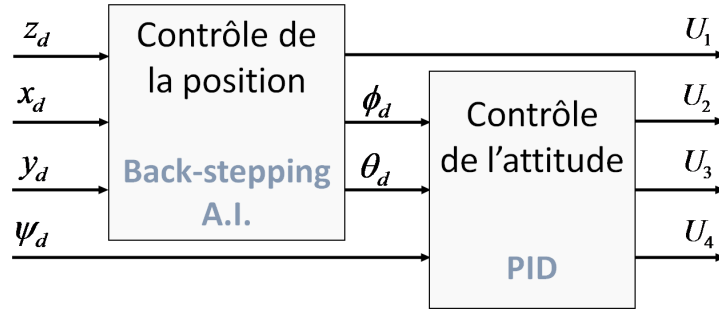


Figure 3.3. Stratégie de commande hybride proposée.

a- Commande de la position verticale

La dynamique de la position verticale d'un système multi-rotor est donnée dans le système d'équations (3.1) comme suit :

$$\ddot{z} = -g + \frac{\cos \phi \cos \theta}{m} U_1 \quad (3.3)$$

La figure (3.4) représente un schéma de la commande de la position verticale z par la méthode Back-stepping A.I. décomposée en deux étapes.

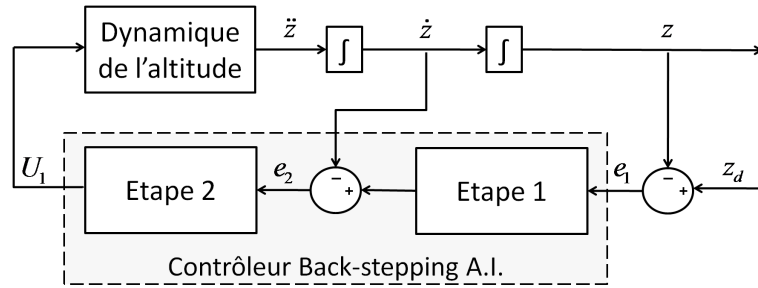


Figure 3.4. Commande de la position verticale.

La première étape dans l'algorithme Back-stepping A.I. pour la commande de l'altitude consiste à définir l'erreur de suivi de l'altitude e_{z1} et sa dérivée temporelle \dot{e}_{z1} :

$$e_{z1} = z_d - z, \quad \dot{e}_{z1} = \dot{z}_d - \dot{z} \quad (3.4)$$

En fonction de l'erreur e_{z1} , on considère une fonction de Lyapunov $V(e_{z1})$ définie positive et sa dérivée temporelle semi-définie négative.

$$V(e_{z1}) = \frac{1}{2} \left[e_{z1}^2 + \alpha_1 \left(\int_0^t e_{z1}(\tau) d\tau \right)^2 \right] \quad (3.5)$$

avec α_1 constante positive.

$$\begin{aligned}
\dot{V}(e_{z1}) &= e_{z1} \left(\dot{e}_{z1} + \alpha_1 \int_0^t e_{z1}(\tau) d\tau \right) \\
&= e_{z1} \left(\dot{z}_d - \dot{z} + \alpha_1 \int_0^t e_{z1}(\tau) d\tau \right)
\end{aligned} \tag{3.6}$$

Pour assurer la convergence de z vers z_d ($e_{z1} \rightarrow 0$), $\dot{V}(e_{z1})$ doit être semi-définie négative ($\dot{V}(e_{z1}) \leq 0$). Pour cela, on considère \dot{z} comme commande virtuelle, et on la choisit comme suit :

$$\dot{z} = \dot{z}_d + \alpha_1 \int_0^t e_{z1}(\tau) d\tau + \beta_1 e_{z1}, \quad \beta_1 > 0 \tag{3.7}$$

α_1 et β_1 sont des constantes à régler en fonction de la vitesse désirée de la convergence de l'erreur e_{z1} .

En remplaçant (3.7) dans (3.6), nous obtenons :

$$\dot{V}(e_{z1}) = -\beta_1 e_{z1}^2 \leq 0$$

Dans la deuxième étape, on définit e_{z2} l'erreur entre la vitesse d'altitude \dot{z} et la commande virtuelle correspondante telle qu'elle est choisie dans (3.7) :

$$e_{z2} = \dot{z}_d + \alpha_1 \int_0^t e_{z1}(\tau) d\tau + \beta_1 e_{z1} - \dot{z} \tag{3.8}$$

Sa dérivée en fonction du temps est :

$$\dot{e}_{z2} = \ddot{z}_d + \alpha_1 e_{z1} + \beta_1 \dot{e}_{z1} - \ddot{z} \tag{3.9}$$

A partir de (3.8), on peut déduire \dot{e}_{z1} en fonction de e_{z2} :

$$\dot{e}_{z1} = \dot{z}_d - \dot{z} = e_{z2} - \alpha_1 \int_0^t e_{z1}(\tau) d\tau - \beta_1 e_{z1} \tag{3.10}$$

En remplaçant (3.10) et (3.3) dans (3.9), on obtient :

$$\begin{aligned}
\dot{e}_{z2} &= \ddot{z}_d + \alpha_1 e_{z1} + \beta_1 \underbrace{\left(e_{z2} - \alpha_1 \int_0^t e_{z1}(\tau) d\tau - \beta_1 e_{z1} \right)}_{\dot{e}_{z1}} \\
&\quad + \underbrace{g - \frac{\cos \phi \cos \theta}{m} U_1}_{-\ddot{z}}
\end{aligned} \tag{3.11}$$

Afin d'assurer la convergence de e_{z2} vers zéro, une fonction augmentée de Lyapunov peut être définie comme suit :

$$V(e_{z1}, e_{z2}) = \frac{1}{2} \left[e_{z1}^2 + e_{z2}^2 + \alpha_1 \left(\int_0^t e_{z1}(\tau) d\tau \right)^2 \right]$$

dont la dynamique est donnée par :

$$\dot{V}(e_{z1}, e_{z2}) = e_{z1} \left(\dot{e}_{z1} + \alpha_1 \int_0^t e_{z1}(\tau) d\tau \right) + e_{z2} \dot{e}_{z2} \quad (3.12)$$

En substituant \dot{e}_{z1} et \dot{e}_{z2} par leurs expressions données dans (3.10) et (3.11), on obtient :

$$\begin{aligned} \dot{V}(e_{z1}, e_{z2}) = & -\beta_1 e_{z1}^2 + \beta_1 e_{z2}^2 + (1 + \alpha_1 - \beta_1^2) e_{z1} e_{z2} - \alpha_1 \beta_1 e_{z2} \int_0^t e_{z1}(\tau) d\tau \\ & + \ddot{z}_d e_{z2} + g e_{z2} - \frac{\cos \phi \cos \theta}{m} e_{z2} U_1 \end{aligned} \quad (3.13)$$

L'entrée de commande U_1 doit être choisie telle que $\dot{V}(e_{z1}, e_{z2})$ soit semi définie négative. En prenant :

$$U_1 = \frac{m}{\cos \phi \cos \theta} \left[g + \ddot{z}_d + (1 + \alpha_1 - \beta_1^2) e_{z1} + (\beta_1 + \beta_2) e_{z2} - \alpha_1 \beta_1 \int_0^t e_{z1}(\tau) d\tau \right] \quad (3.14)$$

où $\beta_2 > 0$, on obtient : $\dot{V}(e_{z1}, e_{z2}) = -\beta_1 e_{z1}^2 - \beta_2 e_{z2}^2$.

$$\begin{cases} \dot{V}(e_{z1}, e_{z2}) < 0, \quad \forall (e_{z1}, e_{z2}) \neq 0 \\ \dot{V}(0) = 0 \end{cases}$$

On conclut que z est asymptotiquement stable.

b- Commande de la position (x,y)

Une fois la force de poussée totale U_1 désirée obtenue, on reprend les mêmes étapes pour le contrôle des positions longitudinale et latérale.

Soient e_{x1} et e_{y1} les erreurs de suivi des positions x et y définies comme suit :

$$\begin{cases} e_{x1} = x_d - x \\ e_{y1} = y_d - y \end{cases}$$

De même que e_{z2} , soient e_{x2} et e_{y2} les erreurs de suivi des vitesses définies comme suit :

$$\begin{cases} e_{x2} = \dot{x}_d + \alpha_2 \int_0^t e_{x1}(\tau) d\tau + \beta_3 e_{x1} - \dot{x} \\ e_{y2} = \dot{y}_d + \alpha_3 \int_0^t e_{y1}(\tau) d\tau + \beta_5 e_{y1} - \dot{y} \end{cases}$$

où $\alpha_2, \alpha_3, \beta_3$ et β_5 sont des constantes positives à régler en fonction de la vitesse désirée de la convergence des erreurs.

En utilisant des fonctions de Lyapunov définies positives dont la dérivée est semi-définie négative, les lois de commande des positions longitudinale et latérale peuvent être choisies comme suit :

$$\begin{cases} u_x = \frac{m}{U_1} \left[\ddot{x}_d + (1 + \alpha_2 - \beta_3^2)e_{x1} + (\beta_3 + \beta_4)e_{x2} - \alpha_2\beta_3 \int_0^t e_{x1}(\tau)d\tau \right] \\ u_y = \frac{m}{U_1} \left[\ddot{y}_d + (1 + \alpha_3 - \beta_5^2)e_{y1} + (\beta_5 + \beta_6)e_{y2} - \alpha_3\beta_5 \int_0^t e_{y1}(\tau)d\tau \right] \end{cases} \quad (3.15)$$

3.3.2 Boucle interne de la commande

La boucle interne de la commande permet d'assurer la stabilisation des angles d'Euler du multi-rotor. Le contrôle de l'angle de roulis ϕ et de l'angle de tangage θ sert à stabiliser l'assiette du multi-rotor à une référence (ϕ_d, θ_d) dans le but de contrôler la position horizontale du système. Comme mentionné plus haut, les références ϕ_d et θ_d sont calculées en utilisant les expressions (3.2) en fonction de u_x et u_y obtenus dans (3.15). L'objectif du contrôle de l'angle de lacet ψ du multi-rotor est d'assurer le contrôle du cap (orientation de l'avant du drone (vecteur \mathbf{b}_x) par rapport au Nord par exemple (vecteur \mathbf{e}_x)). Dans une mission d'exploration ou de sauvetage, le contrôle du cap du multi-rotor permet de contrôler l'orientation de la caméra embarquée.

Pour la stabilisation des angles du multi-rotor, un contrôleur PID classique découplé est appliqué sur chacun des axes. Le PID appliqué est en structure parallèle et représenté dans le domaine du temps continu comme suit :

$$u(t) = K_P e(t) + K_I \int_0^t e(\tau)d\tau + K_D \frac{de(t)}{dt} \quad (3.16)$$

où K_P, K_I et K_d sont les gains du PID à régler par l'opérateur.

Le schéma de la figure (3.5) illustre la structure du contrôleur appliqué.

Les entrées de commande obtenues en utilisant le contrôleur PID pour suivre les références angulaires ϕ_d, θ_d et ψ_d sont alors exprimées comme suit :

$$\begin{cases} U_2 = K_{P\phi}e_\phi(t) + K_{I\phi} \int_0^t e_\phi(\tau)d\tau + K_{D\phi} \frac{de_\phi(t)}{dt} \\ U_3 = K_{P\theta}e_\theta(t) + K_{I\theta} \int_0^t e_\theta(\tau)d\tau + K_{D\theta} \frac{de_\theta(t)}{dt} \\ U_4 = K_{P\psi}e_\psi(t) + K_{I\psi} \int_0^t e_\psi(\tau)d\tau + K_{D\psi} \frac{de_\psi(t)}{dt} \end{cases} \quad (3.17)$$

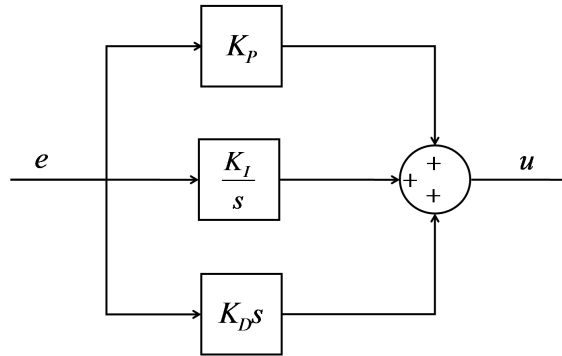


Figure 3.5. Contrôleur PID de forme parallèle.

avec $\mathbf{e} = (e_\phi, e_\theta, e_\psi)$ représente l'erreur et $e_\phi = \phi_d - \phi$, $e_\theta = \theta_d - \theta$, $e_\psi = \psi_d - \psi$. Les mesures de ϕ , θ , et ψ sont fournies par le magnétomètre.

La loi de commande globale est finalement donnée par le vecteur U , composé des entrées de commande $U_j, j = 1, \dots, 4$.

3.3.3 Étude comparative

Dans cette section, nous proposons de comparer l'approche hybride à la méthode Back-stepping A.I. en structure autonome (appliquée sur le système à 6 DDL : translation et rotation) pour effectuer un suivi de trajectoire. Les deux stratégies de commande sont implémentées sous l'environnement Matlab/Simulink pour le cas particulier d'un drone quadri-rotor avec les paramètres physiques [5] : $m = 0.65$ kg, $I_{xx} = I_{yy} = 7.5e - 3$ kgm², $I_{zz} = 1.3e - 2$ kgm², $J_r = 6e - 5$ kgm², $b = 3.13e - 5$, $d = 7.5e - 7$, $l = 0.23$ m and $g = 9.81$ m/s². Le cas général avec N rotors est considéré par la suite.

Le schéma bloc de la commande proposée implémentée sous Simulink est montré dans la figure (3.6).

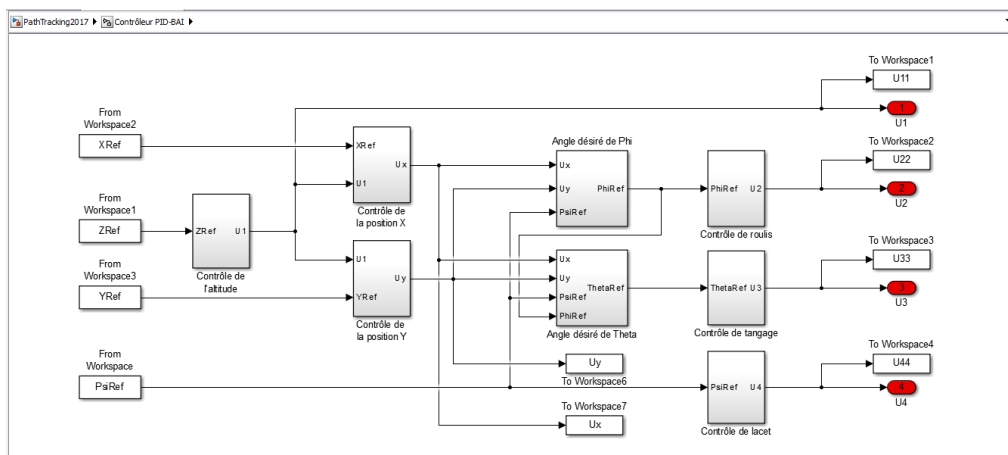


Figure 3.6. Contrôleur hybride implémenté sous Matlab/Simulink.

La trajectoire de référence est d'une forme hélicoidale contenant des virages serrés et des demi-cercles pour tester l'efficacité de la stratégie de commande hybride. La figure

(3.7) représente la trajectoire 3D souhaitée, et celles obtenues en utilisant les deux stratégies avec un zoom sur les réponses au régime transitoire.

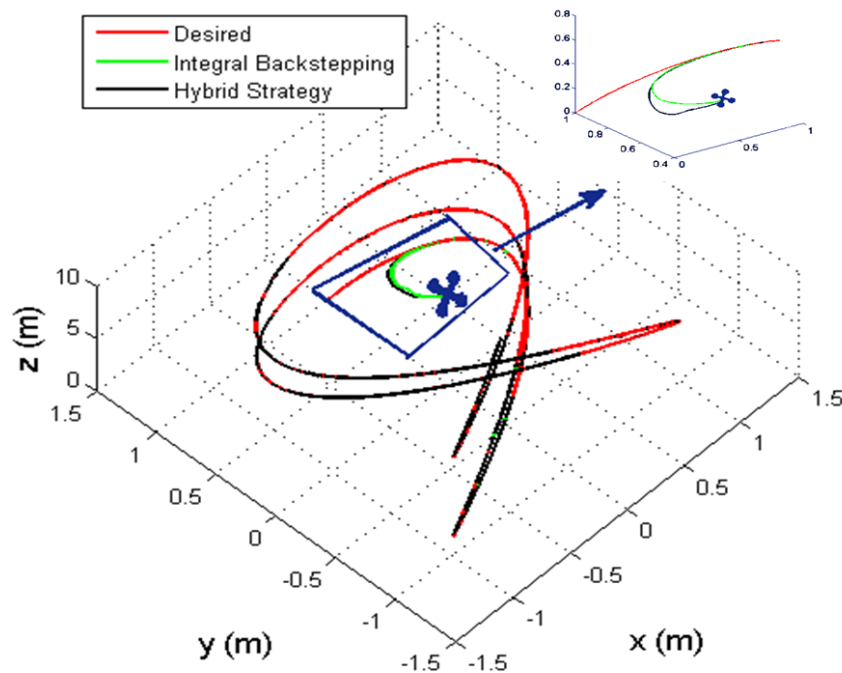


Figure 3.7. Représentation 3D de la trajectoire d'un quadri-rotor, obtenue par le Back-stepping AI (en vert) et la stratégie hybride (en noir). La trajectoire désirée est représentée en rouge, avec un zoom sur le régime transitoire (en haut à droite).

Les conditions initiales sont prises $(0.5, 0.5, 0.3)$ m pour le vecteur de position (x, y, z) et $\psi = 0$ pour l'angle de lacet. Les angles de roulis et tangage désirés sont calculés pour accomplir un suivi de trajectoires horizontales. Comme représenté dans la figure (3.9), les angles de roulis et de tangage prennent la forme sinusoïdale des trajectoires (x, y) désirées.

Les résultats du suivi de trajectoire sont illustrés dans les figures (3.8) et (3.9). Dans le régime transitoire, les contrôleurs montrent des comportements différents. La réponse devient stable après cinq secondes. Dans le régime permanent, les trajectoires des deux stratégies sont conformes à la trajectoire désirée.

Pour comparer les deux approches, les erreurs de suivi de références des six variables sont représentées dans les figures (3.10) et (3.11). L'erreur de stabilisation de la position z est absolument nulle dans les deux stratégies. En revanche, les erreurs de stabilisation de la position (x, y) sont de l'ordre de 10^{-4} . Pour la stabilisation de l'attitude, les erreurs des angles de roulis et tangage sont plus petites en utilisant le contrôleur PID (stratégie hybride) par rapport au Back-stepping A.I.. L'erreur de la stabilisation de l'angle de lacet est égale à zéro en utilisant la stratégie hybride. Elle est de l'ordre de 10^{-4} en appliquant la stratégie Back-stepping. En se basant sur ces résultats, la stratégie hybride a démontré sa capacité pour assurer le suivi de trajectoire des six DDL d'un quadri-rotor. Malgré la simplicité du PID par rapport au Back-stepping A.I., le PID a prouvé sa performance pour assurer la stabilité de la boucle interne.

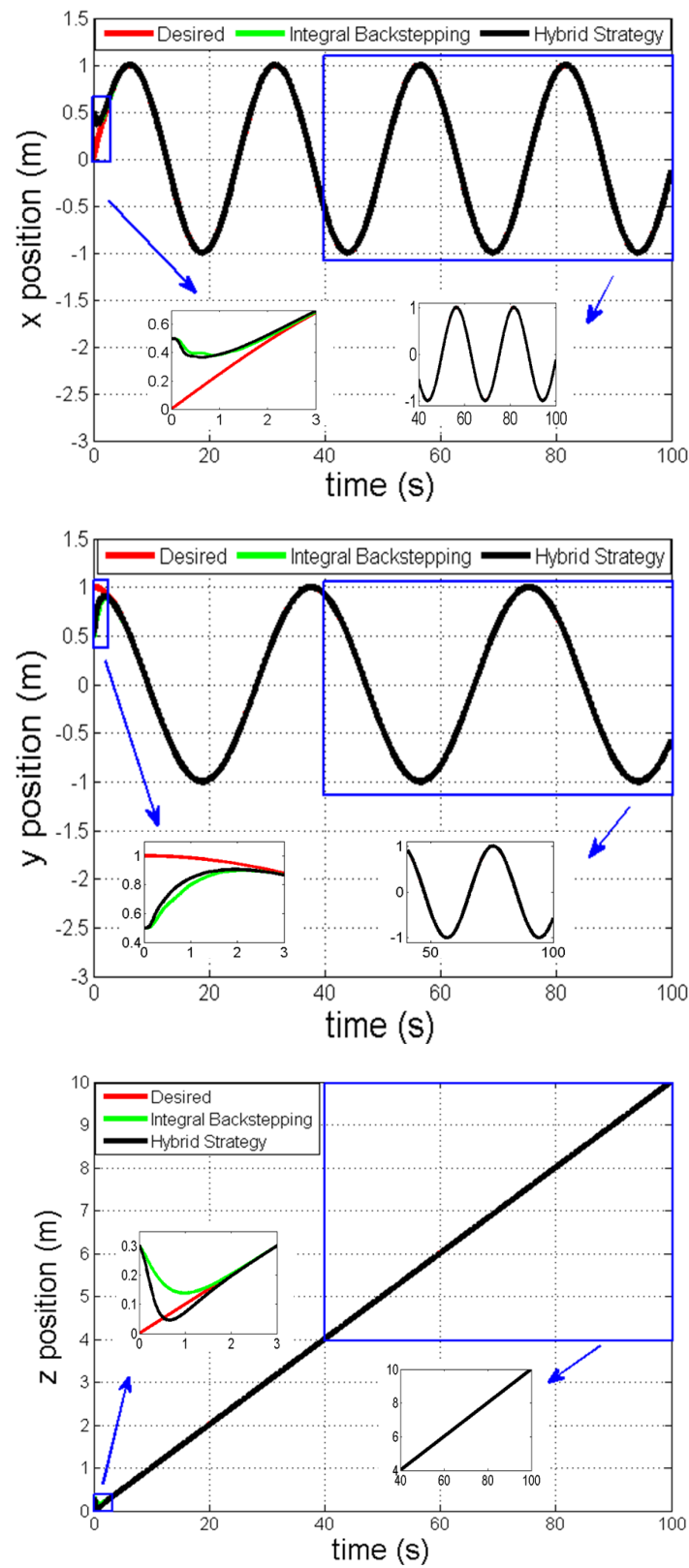


Figure 3.8. Résultats de suivi de références (en rouge) de la position (x,y,z) m d'un quadri-rotor en appliquant le Back-stepping AI (en vert) et la stratégie hybride (en noir)

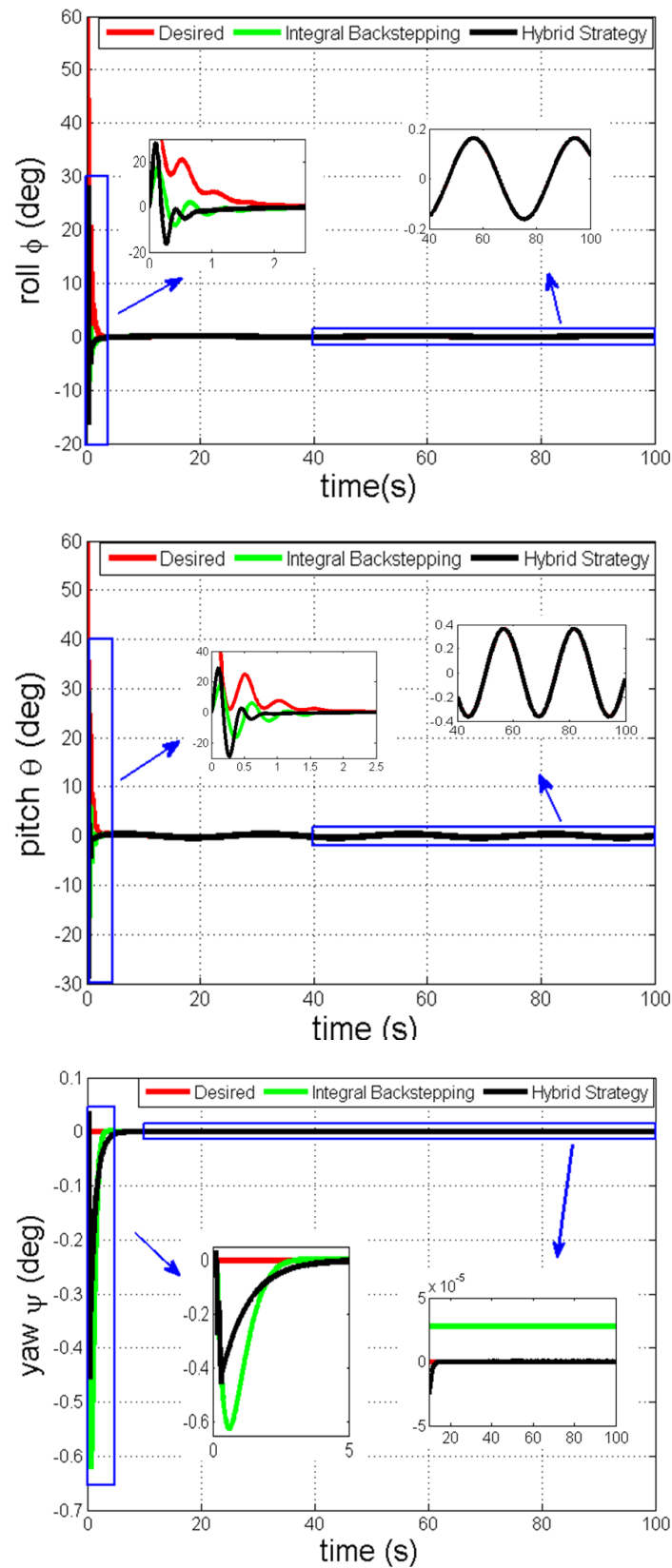


Figure 3.9. Résultats de stabilisation de l'orientation d'un quadri-rotor en appliquant le Back-stepping AI (en vert) et la stratégie hybride (en noir).

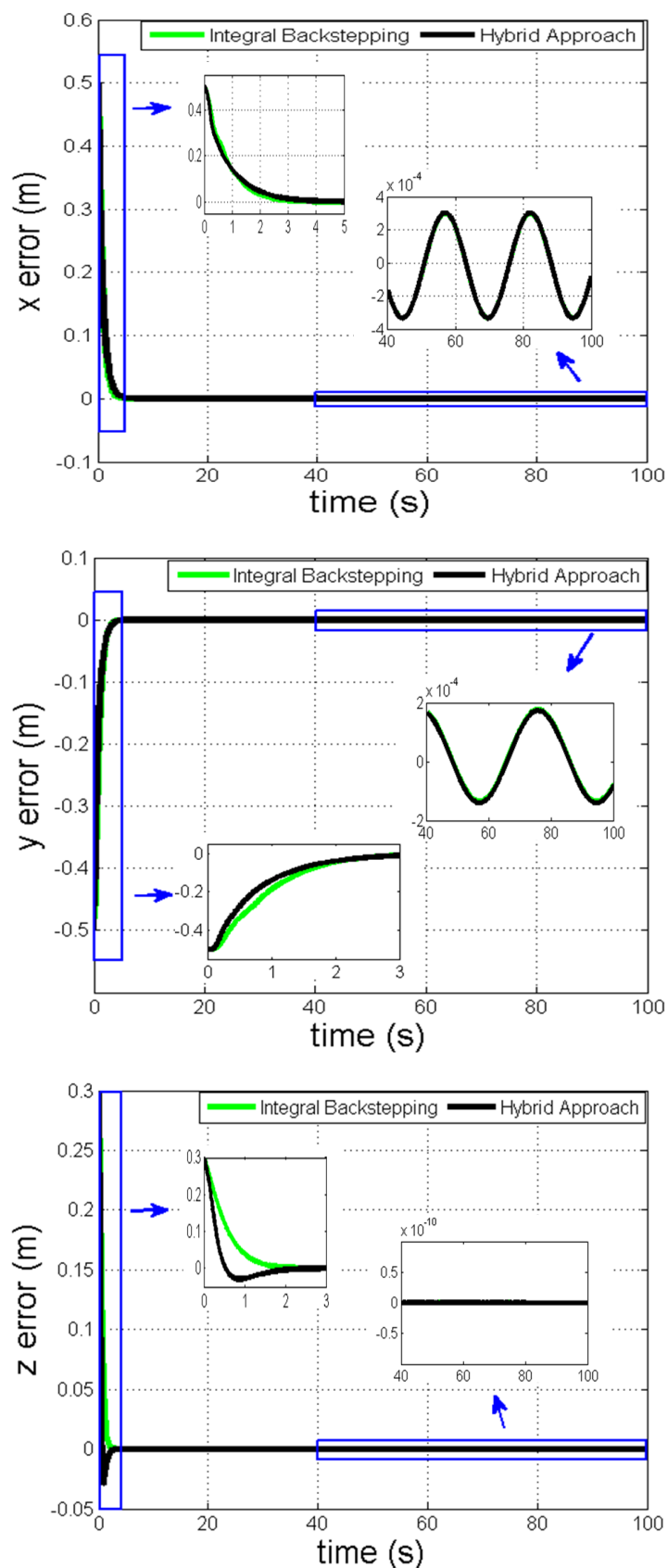


Figure 3.10. Erreur de suivi de références de la position (x,y,z) m d'un quadri-rotor en appliquant le Back-stepping AI (en *vert*) et la stratégie hybride (en *noir*)

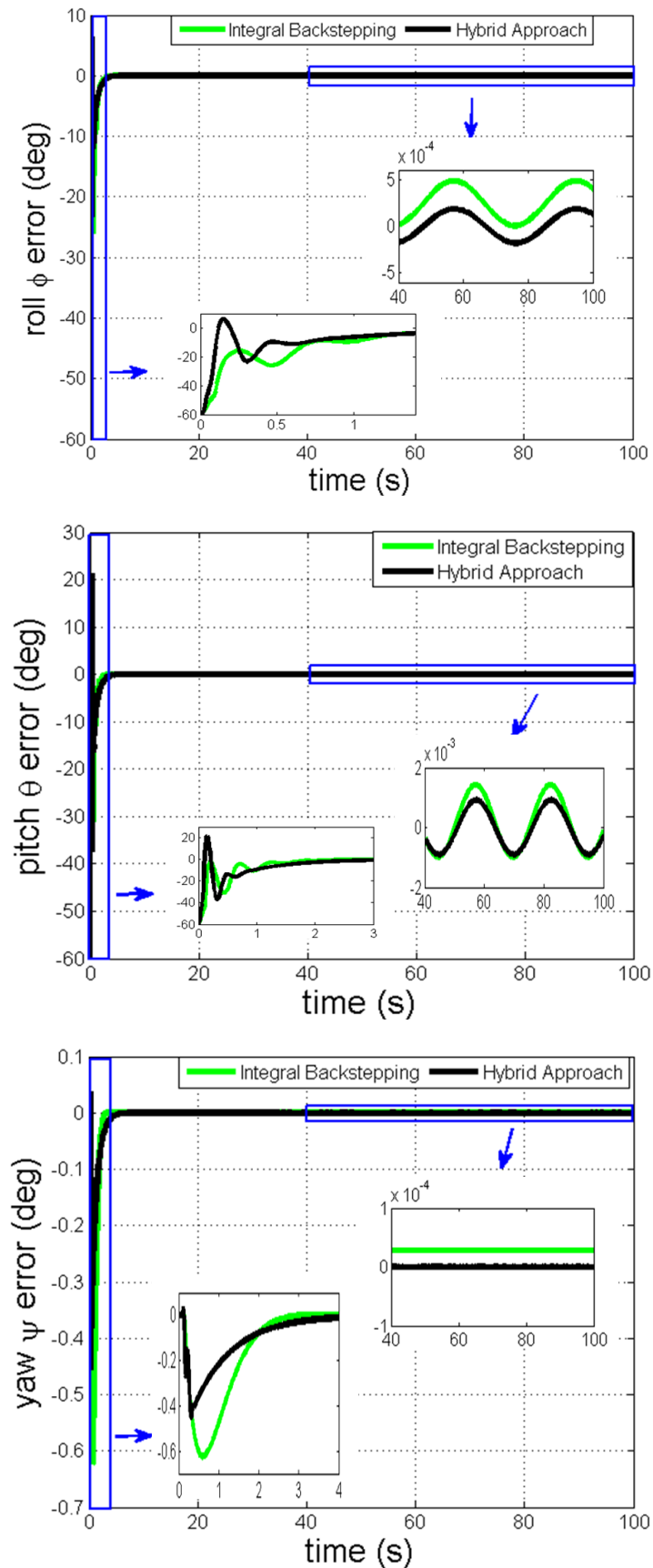


Figure 3.11. Erreur de stabilisation de l'orientation d'un quadri-rotor en appliquant le Back-stepping AI (en *vert*) et la stratégie hybride (en *noir*).

3.4 Allocation des commandes sur les actionneurs

Cette étape permet de répartir la force de poussée et les couples actifs désirés obtenus dans (3.14) et (3.17) sur les actionneurs du multi-rotor. Cela signifie que les entrées de commande U_i ($i = 1, \dots, 4$) assurant la stabilité du système doivent être traduites en des consignes d'actionneurs W_j ($j = 1, \dots, N$). W_j sont les vitesses de rotation des moteurs que le multi-rotor possède. Par conséquent, l'ajout d'un module d'allocation des commandes, comme le montre la figure (3.12), est indispensable avant d'envoyer les commandes au système multi-rotor.

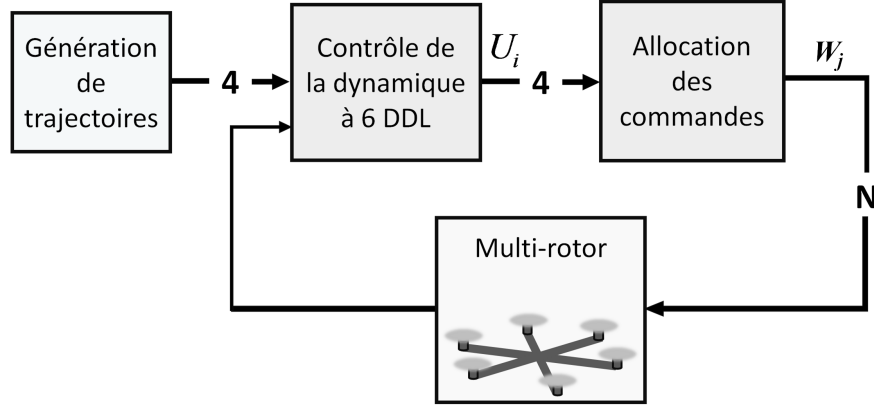


Figure 3.12. Structure de contrôle de drone multi-rotor.

La relation entre les entrées de commande et les vitesses de rotation des moteurs est exprimée par la matrice d'efficacité du contrôle A en tant que $\mathbf{U} = A_{4 \times N} \cdot \boldsymbol{\omega}$ avec :

$$\boldsymbol{\omega} = (W_1^2, W_2^2, \dots, W_N^2)^T$$

En se basant sur le modèle dynamique généralisé (2.40) proposé dans le chapitre 2, nous pouvons écrire la matrice A dans sa formule généralisée comme suit :

$$A = \begin{bmatrix} b & \dots & b & \dots & b \\ bl \sin \left((1-\epsilon) \frac{2\pi}{N} \right) & \dots & bl \sin \left((j-\epsilon) \frac{2\pi}{N} \right) & \dots & bl \sin \left((N-\epsilon) \frac{2\pi}{N} \right) \\ -bl \cos \left((1-\epsilon) \frac{2\pi}{N} \right) & \dots & -bl \cos \left((j-\epsilon) \frac{2\pi}{N} \right) & \dots & -bl \cos \left((N-\epsilon) \frac{2\pi}{N} \right) \\ d & \dots & (-1)^{j+1}d & \dots & -d \end{bmatrix} \quad (3.18)$$

Les paramètres b , d et l représentent respectivement le coefficient de poussée, le coefficient de trainée et la demi-envergure du multi-rotor. j est le numéro de colonne de la matrice A , et ϵ est défini dans (2.36) selon le type de la configuration considérée.

Les vitesses de rotation W_j sont contraintes par leurs limites supérieure W_{max} et inférieure W_{min} qui définissent un ensemble borné dans \mathbb{R}^N .

$$0 < W_{min} \leq W_j \leq W_{max}, \quad \forall j = 1, \dots, N \quad (3.19)$$

Pour le cas particulier des quadri-rotors, la matrice d'efficacité est carrée et inversible. Cependant, pour les multi-rotors avec $N \geq 6$ rotors, la matrice d'efficacité n'est pas carrée et ne peut pas être inversée. Il existe un nombre infini de matrices pseudo-inverses possibles. Dans ce travail, nous avons utilisé la pseudo-inverse de Moore-Penrose pour calculer $\mathbf{A}_{N \times 4}^+$ dans sa forme généralisée en fonction du nombre des actionneurs N et de ϵ . Nous avons obtenu :

$$\mathbf{A}^+ = \frac{1}{N} \begin{bmatrix} \frac{1}{b} & \frac{2 \sin \left((1 - \epsilon) \frac{2\pi}{N} \right)}{bl} & -\frac{2 \cos \left((1 - \epsilon) \frac{2\pi}{N} \right)}{bl} & \frac{1}{d} \\ \vdots & \vdots & \vdots & \vdots \\ \frac{1}{b} & \frac{2 \sin \left((j - \epsilon) \frac{2\pi}{N} \right)}{bl} & -\frac{2 \cos \left((j - \epsilon) \frac{2\pi}{N} \right)}{bl} & \frac{(-1)^{j+1}}{d} \\ \vdots & \vdots & \vdots & \vdots \\ \frac{1}{b} & \frac{2 \sin \left((N - \epsilon) \frac{2\pi}{N} \right)}{bl} & -\frac{2 \cos \left((N - \epsilon) \frac{2\pi}{N} \right)}{bl} & -\frac{1}{d} \end{bmatrix} \quad (3.20)$$

où $j = 1, \dots, N$ représente le numéro de ligne de la matrice A^+ .

L'ensemble de commandes atteignables ou **Admissible Control Set (ACS)** d'un multi-rotor peut être défini comme le sous-espace à quatre dimensions des entrées de commande (force de poussée et couples actifs) qui peuvent être atteints tout en satisfaisant les contraintes sur les vitesses de rotation des moteurs. Géométriquement, le **ACS** d'un multi-rotor est un polyèdre convexe défini dans \mathbb{R}^4 par l'intersection d'un nombre fini de demi-espaces. En utilisant (3.19) et la matrice pseudo-inverse donnée dans (3.20), nous pouvons écrire :

$$\begin{cases} \forall j = 1, \dots, N; \\ 0 < W_{min}^2 \leq \frac{1}{N} \left(\frac{1}{b} U_1 + \frac{2 \sin \left((j - \epsilon) \frac{2\pi}{N} \right)}{bl} U_2 - \frac{2 \cos \left((j - \epsilon) \frac{2\pi}{N} \right)}{bl} U_3 + \frac{(-1)^{j+1}}{d} U_4 \right) \leq W_{max}^2 \end{cases} \quad (3.21)$$

À partir de (3.21), nous obtenons le système d'inégalités suivant :

$$\begin{cases} \forall j = 1, \dots, N; \\ lU_{1min} \leq lU_1 + 2 \sin \left((j - 1) \frac{2\pi}{N} \right) U_2 - 2 \cos \left((j - 1) \frac{2\pi}{N} \right) U_3 + \frac{lb}{d} (-1)^{j+1} U_4 \leq lU_{1max} \end{cases} \quad (3.22)$$

où : $U_{1min} = NbW_{min}^2$ et $U_{1max} = NbW_{max}^2$ sont les bornes (supérieure et inférieure) de la force de poussée totale F_t atteignable. Le **ACS** d'un drone N -rotor en utilisant la méthode de pseudo-inversion de Moore-Penrose pour l'allocation des commandes est donc à obtenir à partir du système de N inégalités donné dans l'équation (3.22).

3.5 Résultats numériques

Afin de valider l'approche de commande hybride sur des systèmes multi-rotors, nous considérons un scénario de suivi de trajectoire en 3D, simulé sous l'environnement Matlab/Simulink, pour le cas d'un quadrirotor, hexarotor et octorotor. La vitesse maximale de rotor est fixée à 600 rad/s et les valeurs numériques des paramètres physiques utilisés dans la simulation sont donnés dans le tableau (3.1) comme dans [5], [130], et [131].

TABLE 3.1
Valeurs numériques utilisées dans la simulation.

	Quadri-rotor	Haxa-rotor	Octo-rotor	Unité
m	0.65	0.468	1.2	kg
l	0.23	0.225	0.4	m
b	3.13e-5	2.980e-6	3.13e-5	-
d	7.5e-7	1.140e-7	7.5e-7	-
I_{xx}	7.5e-3	4.856e-3	7.5e-3	$kg.m^2$
I_{yy}	7.5e-3	4.856e-3	7.5e-3	$kg.m^2$
I_{zz}	1.3e-2	8.801e-3	1.3e-2	$kg.m^2$
J_r	6e-5	3.357e-5	6e-5	$kg.m^2$
g	9.81	9.81	9.81	m/s^2

La trajectoire désirée et la réponse des systèmes quadri-, hexa- et octo-rotor sont données respectivement dans les figures (3.13), (3.14) et (3.15). Les déplacements selon x et y ont des formes sinusoïdales. La position verticale et l'angle de lacet désirés sont choisis en tant que constantes ($1m$ et 20° respectivement).

Les différentes figures montrent que la position désirée en (x,y) est très bien suivie. Nous ne pouvons absolument pas observer de différence entre le comportement des trois systèmes drones. Cela est dû au fait que la trajectoire est physiquement réalisable pour chacun des systèmes. L'altitude et le cap désirés sont atteints et stabilisés en quelques secondes (environ trois secondes pour le cap) avec un dépassement de 0%.

L'entrée de commande U_1 est représentée sur la figure (3.16). Elle est de la même forme pour les trois systèmes à des amplitudes différentes. Comme la position verticale est considérée constante, la poussée totale U_1 devrait être constante. Cependant, elle montre des oscillations négligeables ($\sim 10^{-8}[N]$) en raison de la forme sinusoïdale de la position (x,y) .

En nous fondant sur les résultats présentés, nous pouvons constater que si le système est un quadri-rotor, hexa-rotor ou octo-rotor, sa stabilisation est très bien assurée. Ceci valide l'application de l'approche de commande hybride pour le contrôle des multi-rotors et montre son efficacité. Quelque soit le nombre de rotors, les mêmes résultats de suivi sont obtenus.

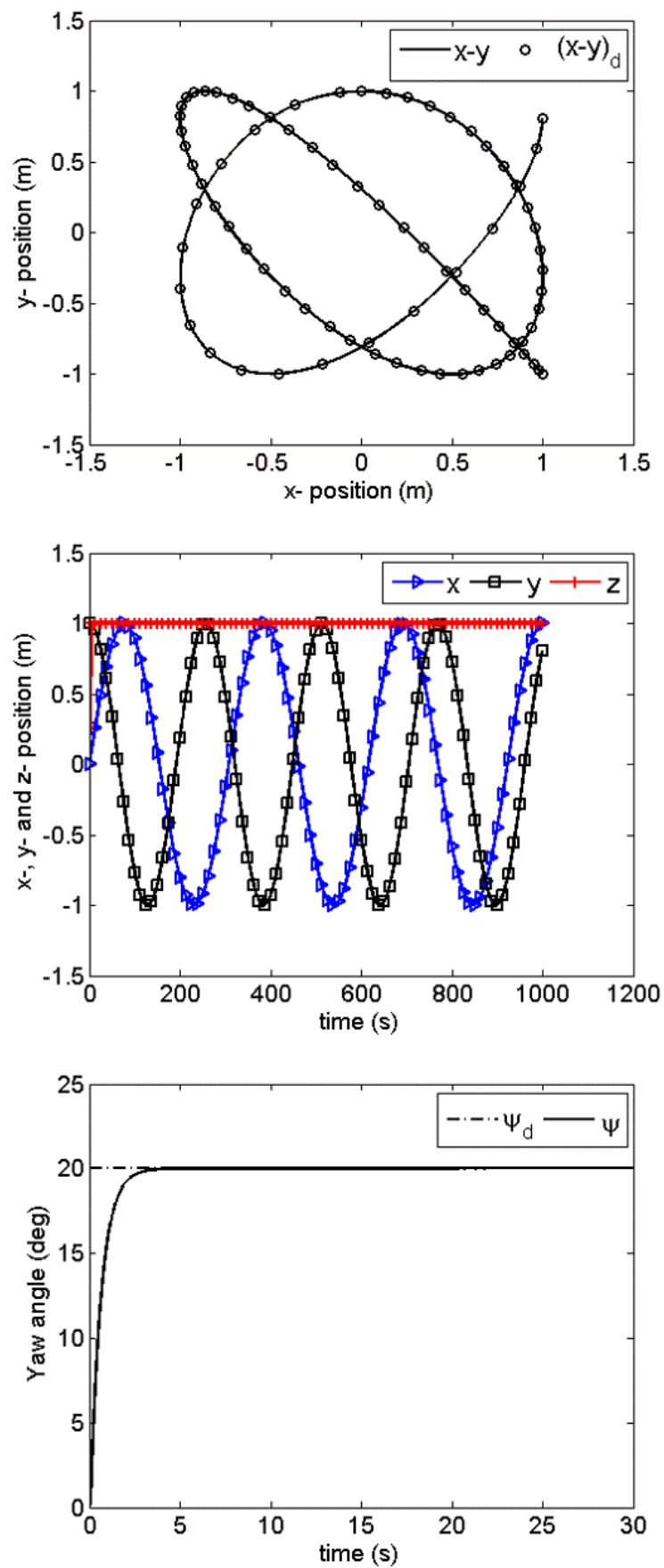


Figure 3.13. Résultats de suivi de trajectoire en utilisant un drone quadri-rotor.

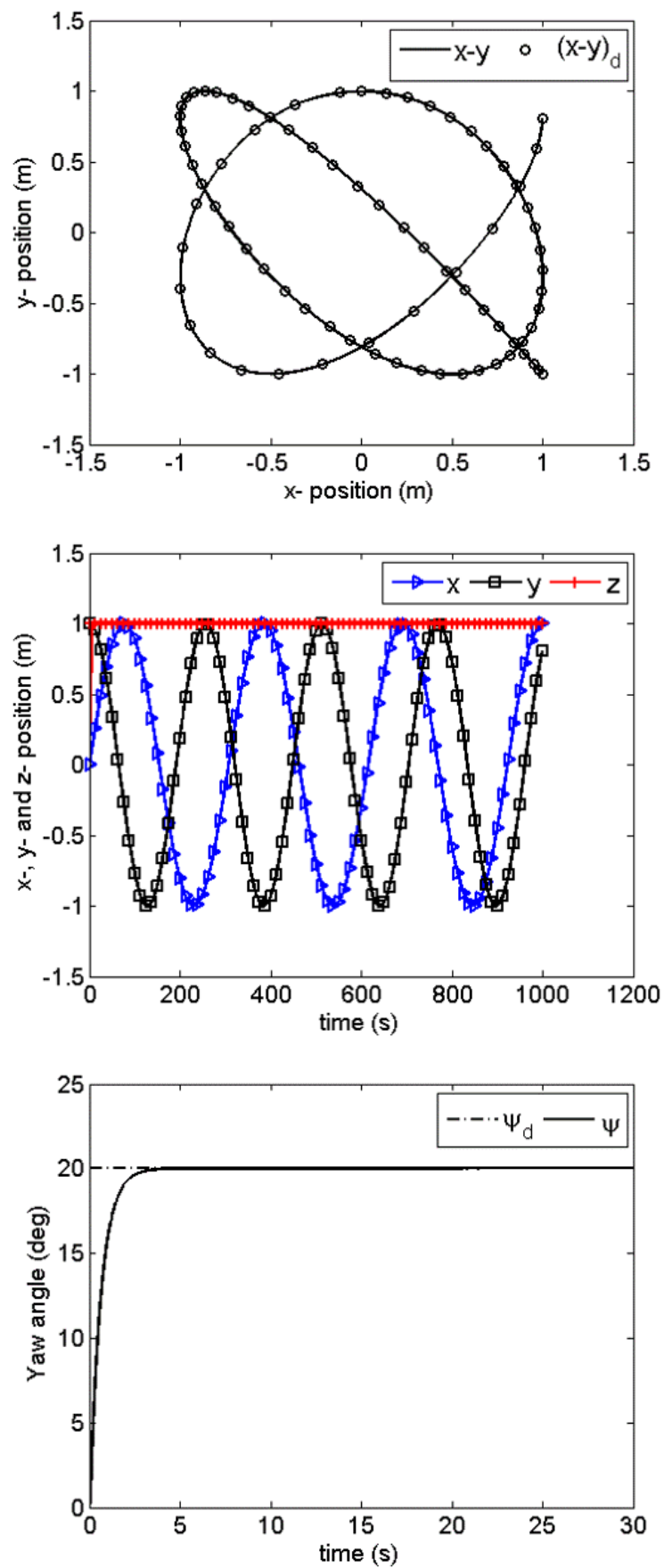


Figure 3.14. Résultats de suivi de trajectoire en utilisant un drone hexa-rotor.

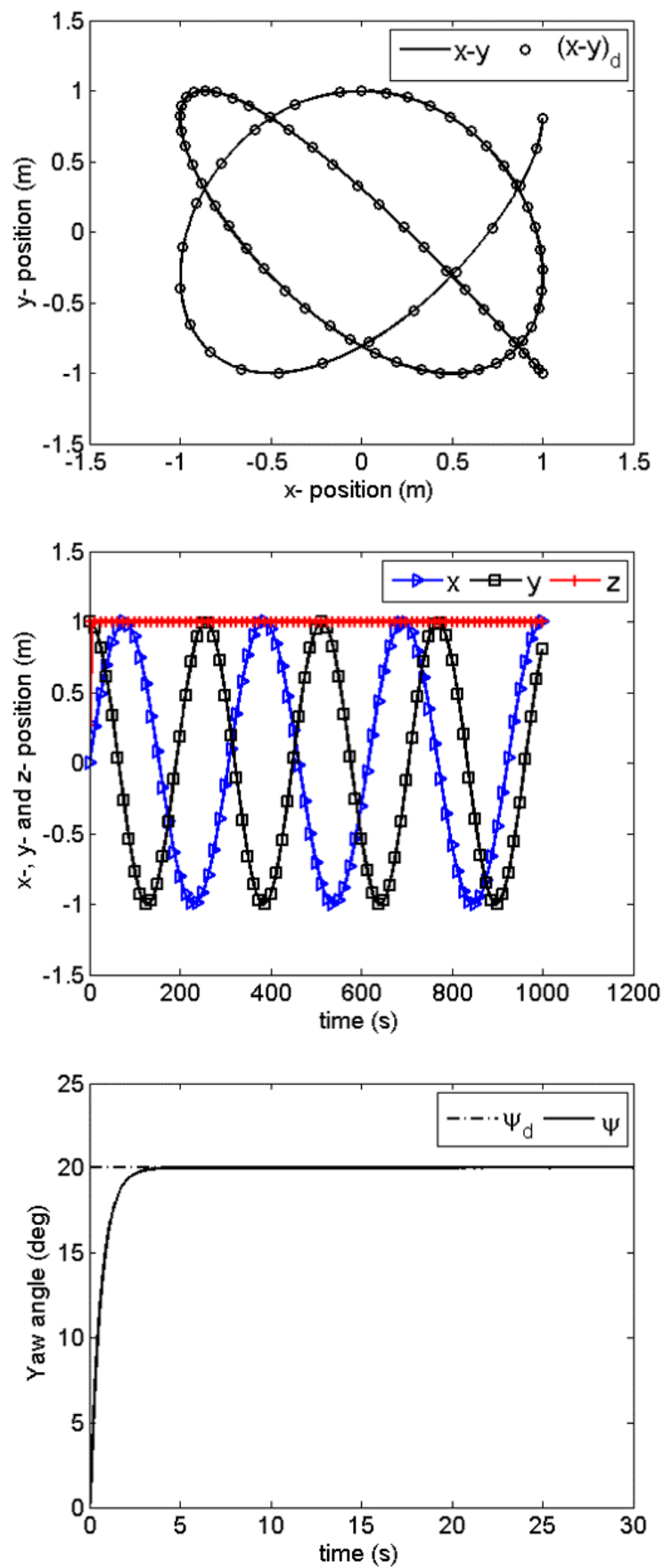
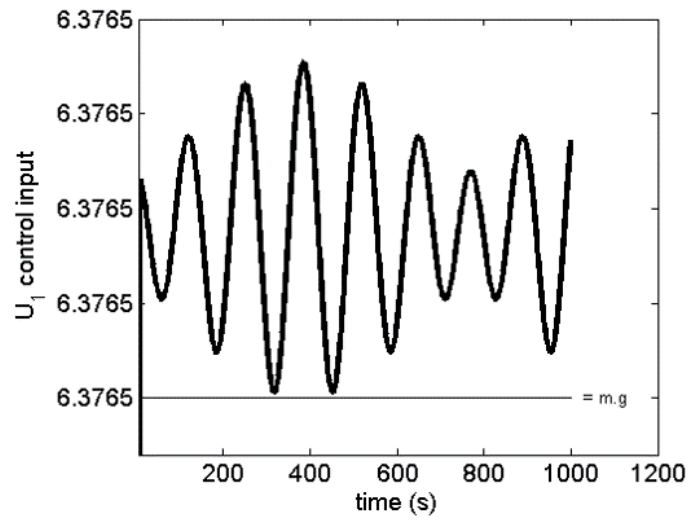
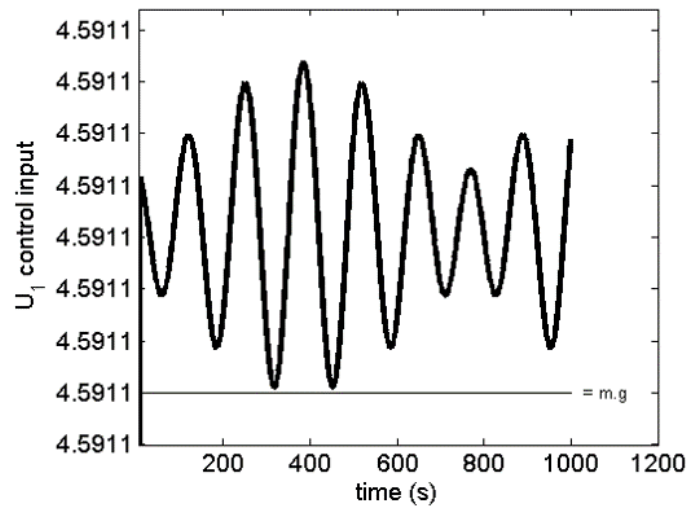


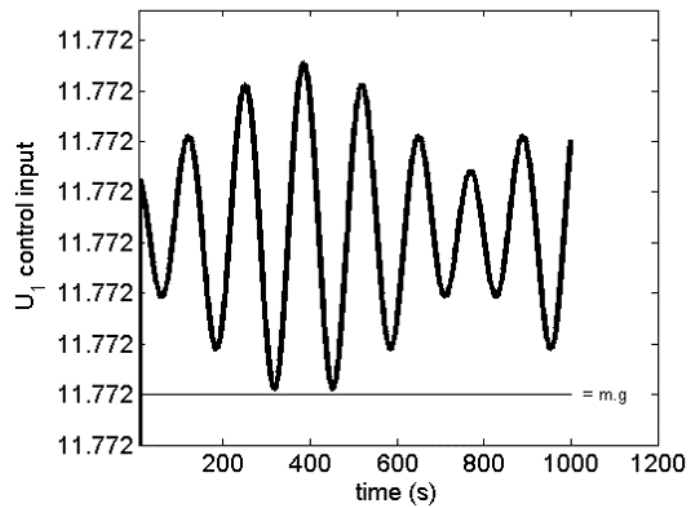
Figure 3.15. Résultats de suivi de trajectoire en utilisant un drone octo-rotor.



(a) quadri-rotor



(b) hexa-rotor



(c) octo-rotor

Figure 3.16. L'entrée de commande $U_1(t)$ en utilisant, (a) un quadrirotor, (b) un hexa-rotor et (c) un octo-rotor. Les valeurs numériques de U_1 dans les trois cas correspondent au poids des multi-rotors ($m.g$).

3.6 Étude en présence de pannes

Nous envisageons ici de démontrer l'effet de l'échec d'un rotor sur le comportement du drone N -rotor. Pour simuler l'échec du rotor, nous considérons deux scénarios de vol différents avec un rotor inactif :

- **Premier scénario** : Stabilisation sur un point cible (vol stationnaire).
- **Deuxième scénario** : Suivi de trajectoire en 3D.

Le rotor mis en échec est arbitrairement choisi pour être le 3^{ème} rotor. Cela va aussi montrer la robustesse du contrôleur pour automatiquement faire face à la dégradation des performances du système en présence de panne, sans considérer une approche de mise à jour dynamique de l'allocation des commandes. Le cas du quadri-rotor n'est pas considéré dans cette étude en présence de pannes. Dans un système quadri-rotor, la défaillance totale d'un rotor produit un mouvement de roulis (ou tangage) incontrôlable (si pas de contrôleur tolérant aux défauts considéré). Cela mène à une position horizontale (et aussi verticale) incontrôlable et conduit inévitablement le véhicule à l'écrasement.

Notant que les gains du contrôleur sont identiques pour tout système N -rotor avec $6 \leq N \leq 16$ et sont réglés pour montrer l'efficacité de la stratégie du contrôleur dans le cas de défaillance du rotor sans trop exciter le système dans le cas où tous les rotors sont opérationnels. La vitesse du rotor maximale est fixée à 600 rad/s et les valeurs numériques des paramètres des drones et de contrôleur sont résumées dans le tableau (3.2) [130].

TABLE 3.2
Paramètres de drones multi-rotors.

m	0.468 kg	α_1	4e-2	$K_{P\phi}$	1.8
l	0.225 m	β_1	3.5	$K_{I\phi}$	2.5e-2
b	2.98e-6	β_2	2.01	$K_{D\phi}$	0.4
d	1.14e-7	α_2	4e-2	$K_{P\theta}$	1.8
I_{xx}	5.856e-3 kg.m ²	β_3	5	$K_{I\theta}$	2.5e-2
I_{yy}	4.856e-3 kg.m ²	β_4	2.4	$K_{D\theta}$	0.4
I_{zz}	8.801e-3 kg.m ²	α_3	4e-2	$K_{P\psi}$	2
J_r	3.357e-5 kg.m ²	β_5	5	$K_{I\psi}$	2.5e-2
g	9.81 m/s ²	β_6	2.4	$K_{D\psi}$	0.5

En fait, les paramètres de drones, en particulier la masse totale et l'inertie, dépendent de la technologie du drone et du nombre de rotors. Nous considérons cependant dans l'étude suivante que les paramètres sont identiques pour tous les drones N -rotor simulés.

Le premier scénario consiste à un vol pour atteindre le point cible désiré $\zeta_d = (1, 1, 1)m$ avec $\psi_d = 20^\circ$ à partir d'un point de départ $\zeta_0 = (0, 0, 0.2)m$ avec une orientation ini-

tiale $\psi_0 = 0^\circ$. Dans le deuxième scénario, le système N -rotor est contrôlé pour suivre une trajectoire dans le plan (x,y) en maintenant l'altitude $z_d = 1m$ et l'orientation $\psi_d = 20^\circ$. Les lois de commande obtenues en appliquant l'approche hybride Back-stepping A.I. -PID introduite dans la section 3.3 sont utilisées pour la stabilisation de la dynamique du drone N -rotor, en utilisant le modèle généralisé proposé.

Dans les deux scénarios, les simulations débutent avec N rotors opérationnels, puis la panne sur le 3^{ème} rotor est déclenchée via un bloc switch, à l'instant $t = 15s$ dans le premier scénario et $t = 20s$ dans le deuxième. Les trajectoires 3D obtenues sont respectivement données dans les figures (3.17) et (3.18) pour le cas de $N = 8$ rotors.

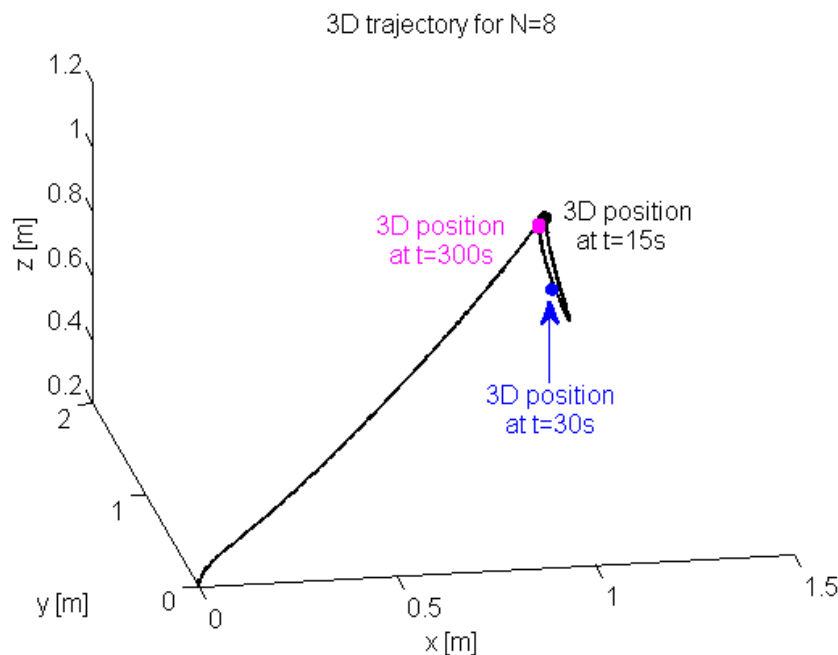


Figure 3.17. Stabilisation sur un point cible pour $N = 8$ rotors. Déclenchement de la panne à $t = 15s$ où la position de drone est représentée par le *point noir*. Le *point bleu* représente la position de drone à l'instant $t = 30s$ correspondant au temps de simulation dans la figure (3.19). A l'instant $t = 300s$, le système récupère presque le point 3D désiré.

Les séries temporelles des réponses des systèmes N -rotor sont représentées sur les figures (3.19) - (3.22). Tout d'abord, on discute la réponse du système dans la situation où tous les rotors sont opérationnels. Nous pouvons remarquer que le point cible (premier scénario) a été atteint avec l'angle de lacet désiré dans environ 3s. Dans le deuxième scénario, la trajectoire désirée en (x, y, z, ψ) a été très bien suivie avec z et ψ constantes, et x et y des trajectoires sinusoïdales. Nous examinons par la suite les réponses obtenues en situation de la panne du rotor.

• Réponse en altitude et en angle de lacet

La référence en altitude et en angle de lacet sont constantes dans les deux scénarios. Les réponses en altitude sont représentées dans les figures (3.19) et (3.21). Elles montrent que pour le cas de $N = 6$, le drone N -rotor n'est pas en mesure d'éviter l'écrasement après la survenance d'une panne de rotor. Le multi-rotor est capable de récupérer la hauteur

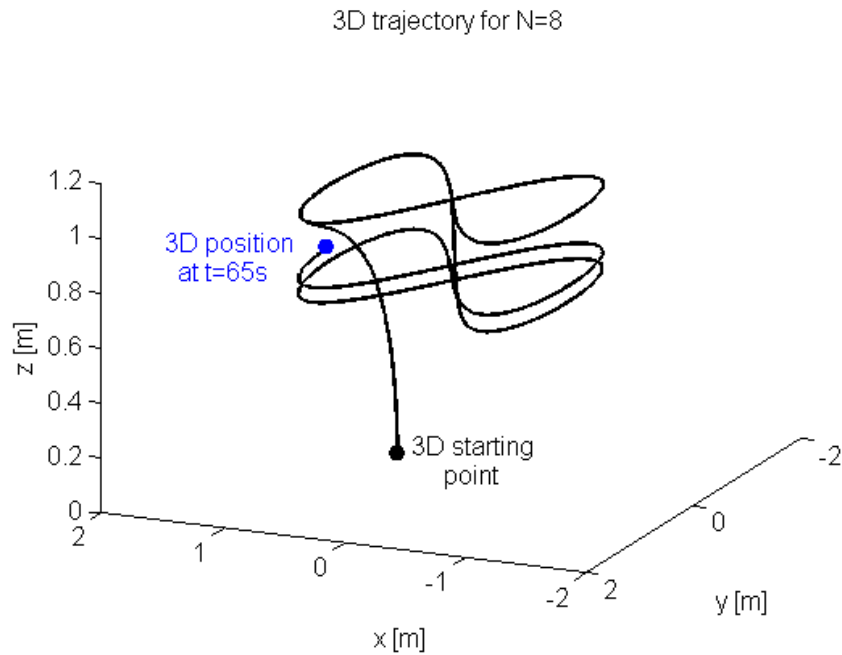


Figure 3.18. Suivi de référence en 3D pour $N = 8$ après une panne de rotor.

souhaitée s'il possède N rotors avec $N > 6$. Nous pouvons noter que plus le nombre de rotors augmente, plus la plage de la dérive de la hauteur diminue. Pour converger plus rapidement, nous avons besoin de régler les gains de contrôleur pour la situation de panne du rotor uniquement.

De la même manière, après l'apparition de la panne, la réponse de lacet converge lentement vers la référence pour tout $N > 6$ comme montré sur les figures (3.20) et (3.22). On peut remarquer que l'erreur instantanée de suivi de l'angle de lacet diminue à mesure que le nombre de rotors augmente.

• Réponse en (x,y)

Comme le montrent les figures (3.19) et (3.21), à l'exception de l'hexa-rotor, les systèmes multi-rotors sont stables dans le plan horizontal après le lancement de la panne du rotor et progressivement convergent vers les références (x_d, y_d) . La dérive causée par la perte d'un rotor est limitée à une petite plage relative. Elle est plus petite quand N augmente pour tout $N > 6$. Pour $N = 6$, la divergence de la réponse correspond à l'écrasement du système.

Nous pouvons observer l'effet de l'emplacement du rotor en panne en zoomant sur la réponse en x représentée sur la figure (3.19). On peut voir que la réponse en x du drone 8-rotors est légèrement meilleure que celle du drone 10-rotors. Cela est expliqué par le fait que le 3^{ème} rotor (en panne) se situe sur l'axe de tangage du système octo-rotor en configuration '++'. La panne n'a donc aucun effet sur le couple actif de tangage. Elle affecte pourtant la dynamique de tangage en raison de la présence des effets gyroscopiques ainsi la nature couplée du système. Voilà pourquoi la relation entre N et l'ordre de l'erreur de

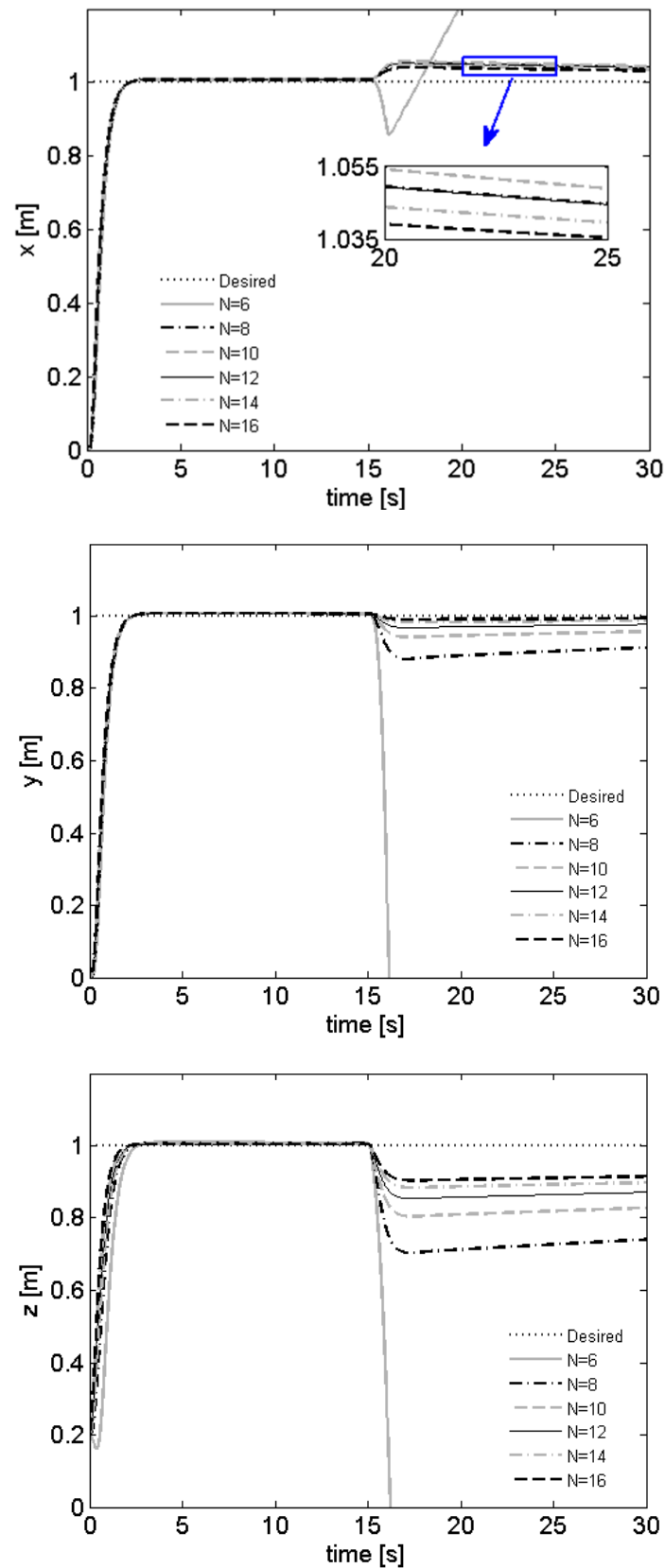


Figure 3.19. Scénario.1 : position 3D d'un N -rotor ($6 \leq N \leq 16$) avec un rotor en panne (panne déclenchée à $t = 15s$).

suivi en x n'est pas simplement une relation inverse. Nous pouvons également voir l'effet de l'emplacement du rotor en panne dans la légère différence de suivi entre les réponses x et y .

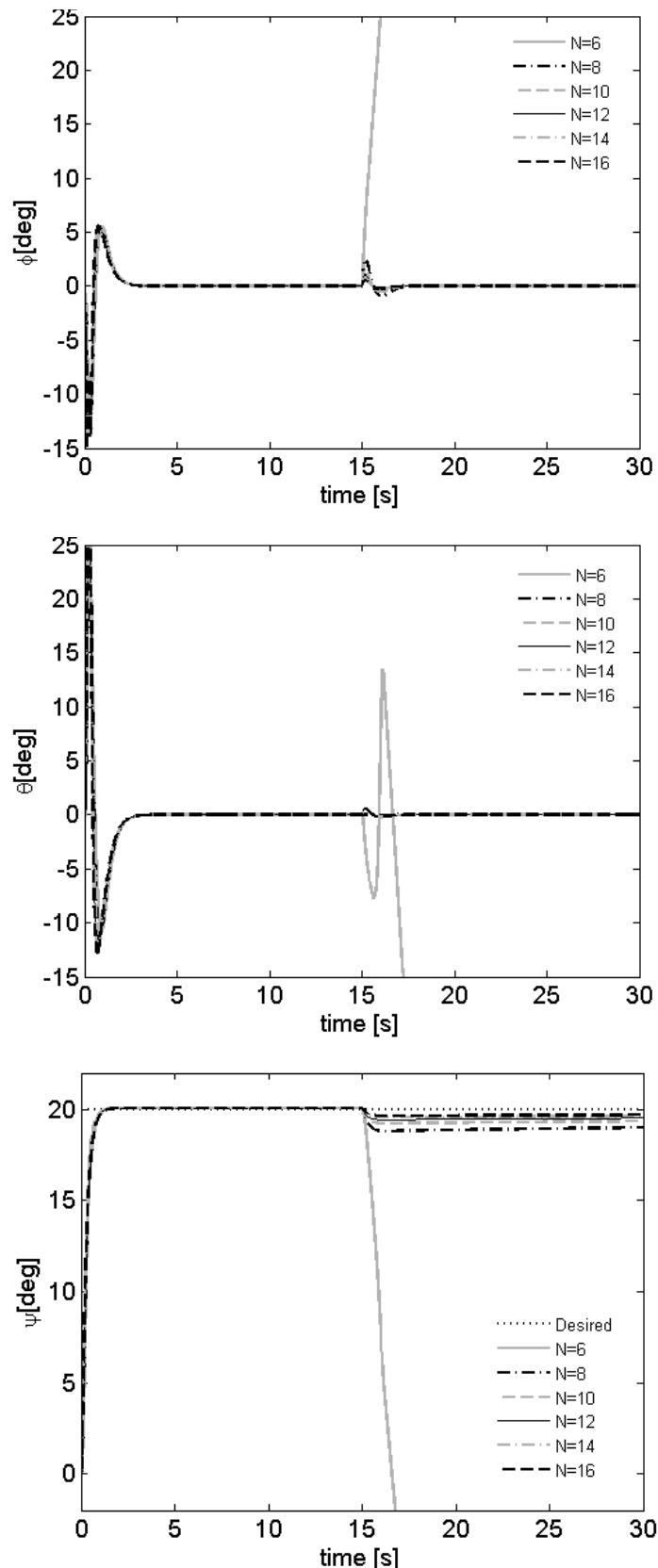


Figure 3.20. Scénario.1 : orientation d'un N -rotor ($6 \leq N \leq 16$) avec un rotor en panne (panne déclenchée à $t = 15$ s).

• Réponses des angles roulis et tangage

Comme les angles de roulis et de tangage sont définis en fonction de la position (x,y) désirée, ils prennent la forme de (x,y) . Comme représenté sur les figures (3.20) et (3.22),

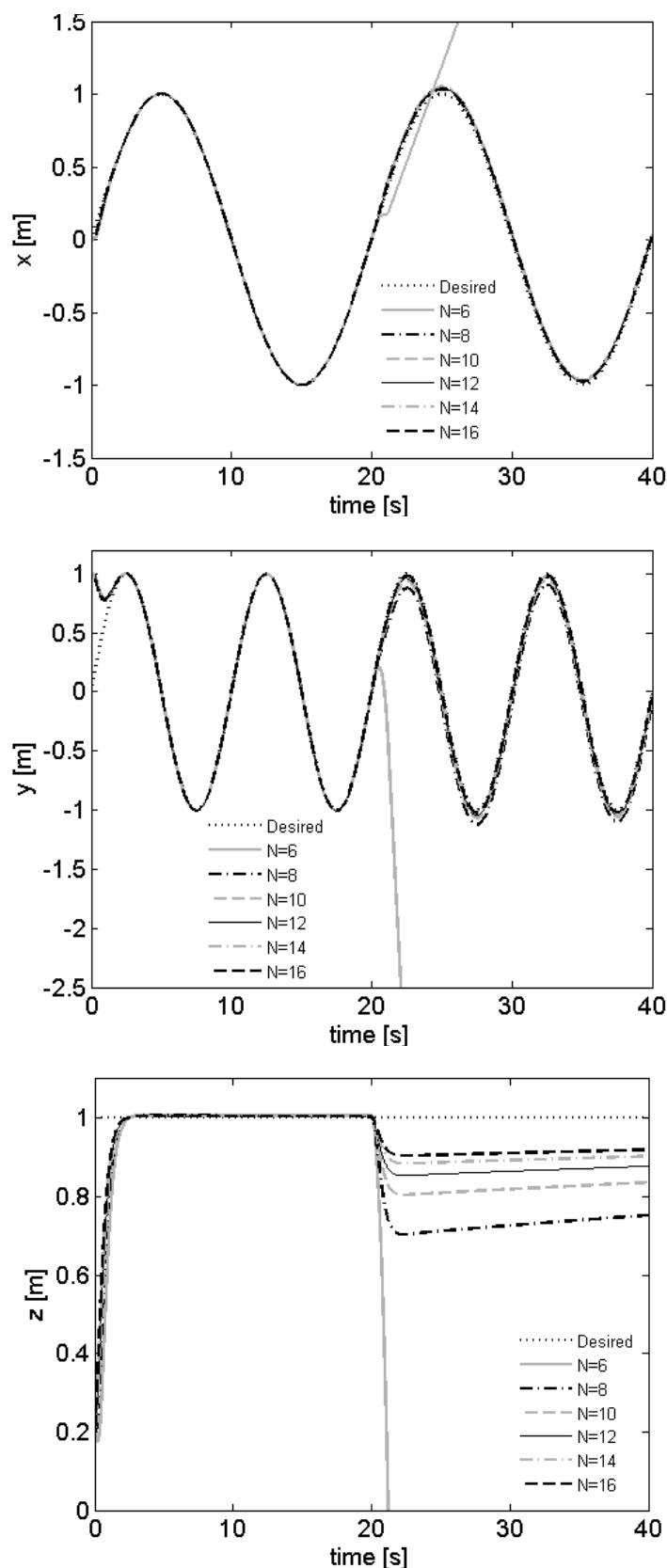


Figure 3.21. Scénario.2 : position 3D d'un N -rotor ($6 \leq N \leq 16$) avec un rotor en panne (panne déclenchée à $t = 20s$).

ϕ_d et θ_d sont constantes dans le scénario de vol stationnaire, et de forme sinusoïdale dans le scénario de suivi de trajectoire. Les cas de non-stabilisation des réponses en roulis et tangage correspondent à la situation d'écrasement du drone.

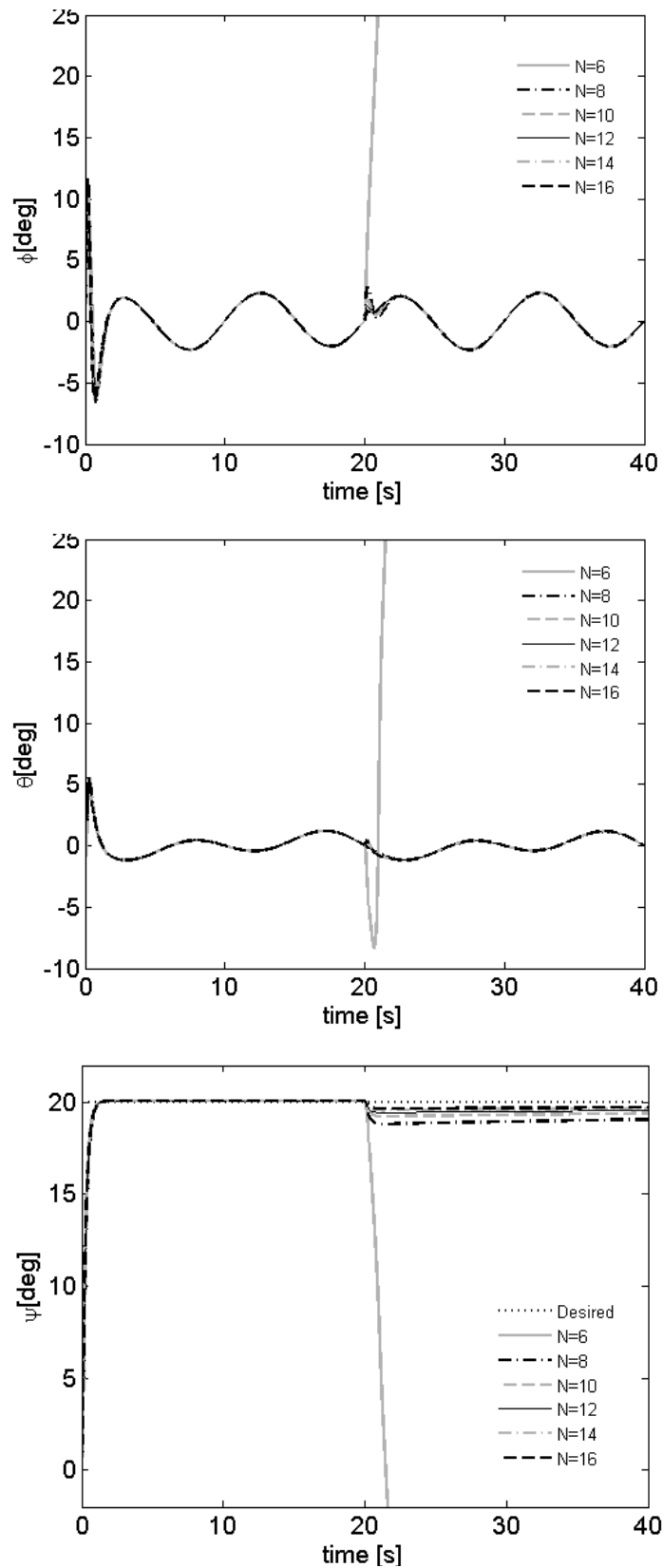
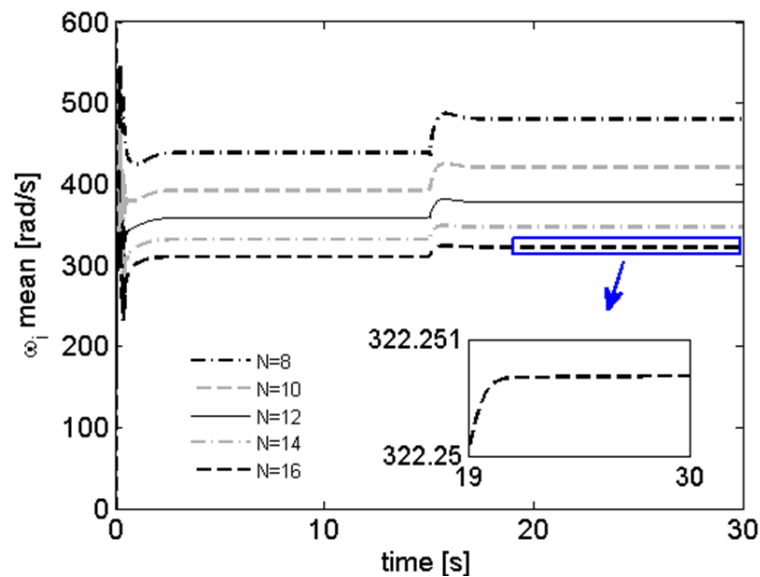


Figure 3.22. Scénario.2 : orientation d'un N -rotor ($6 \leq N \leq 16$) avec un rotor en panne (panne déclenchée à $t = 20$ s).

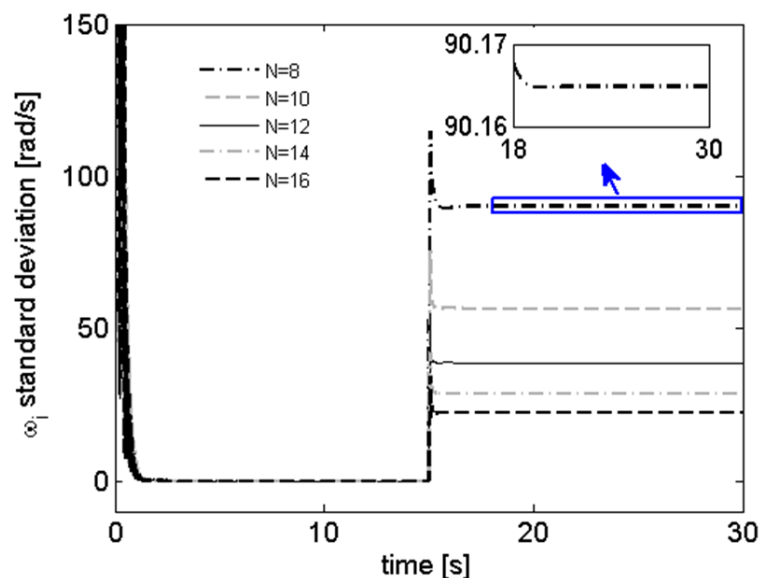
• Moyennes et écarts types des vitesses de rotors

Les figures (3.23) et (3.24) montrent que les moyennes et les écarts-types des vitesses des rotors augmentent lorsque la panne du rotor est déclenchée. Ceci pourrait être expliqué par

les sorties du contrôleur en essayant de maintenir les erreurs de suivi égales à zéro, ce qui produit implicitement une forme de compensation de l'échec. Ainsi, le système converge toujours vers la trajectoire désirée en augmentant les vitesses des rotors appropriés sans se rapprocher de la saturation.



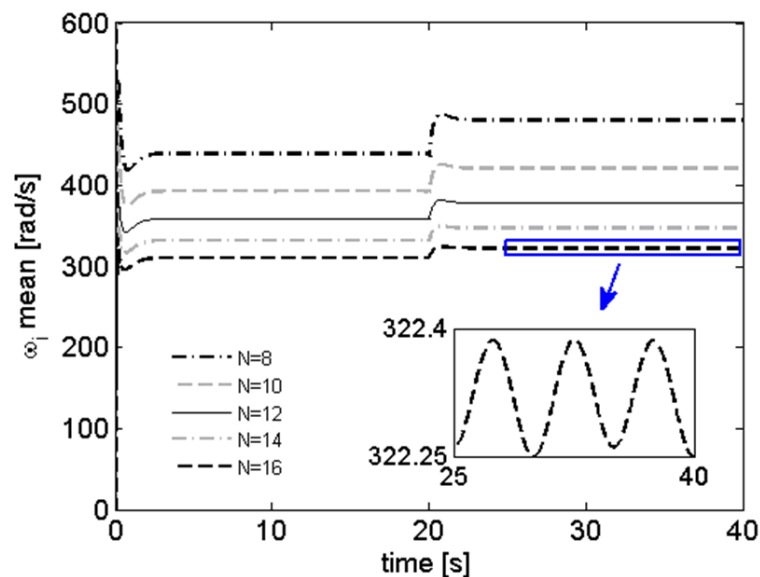
(a)



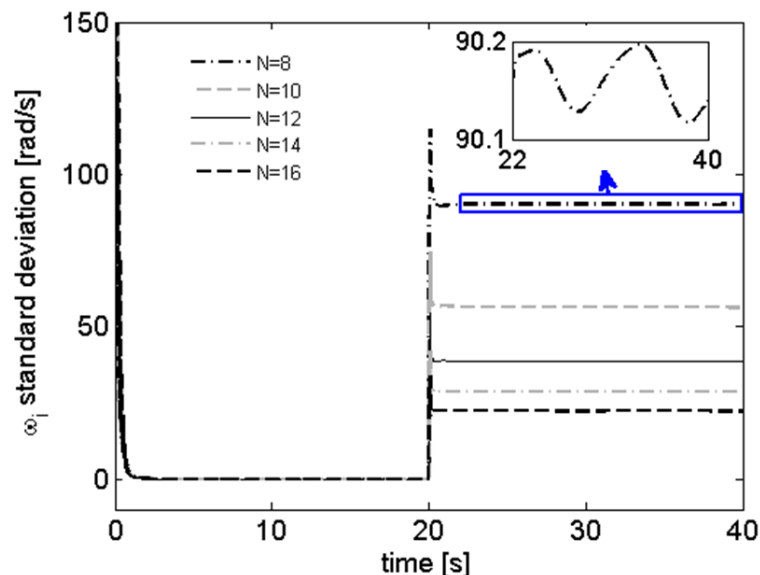
(b)

Figure 3.23. Scénario.1 : allocation des entrées de commande sur les actionneurs d'un N -rotor avant et après le déclenchement de la panne totale d'un rotor ; (a) moyenne des vitesses des rotors, (b) écart type des vitesses.

Pour conclure, en dépit de l'allocation statique des commandes qui ignore l'existence d'une panne sur le 3^{ème} rotor, les réponses du système convergent vers les trajectoires désirées, même si la convergence est lente. La figure (3.17) montre que le point 3D de référence est presque atteint à $t = 300s$. Ceci est confirmé dans les deux scénarios par le traçage des séries temporelles des erreurs en x , y , z et ψ représentées sur la figure (3.25). À noter que la convergence peut être accélérée en ajustant les gains du contrôleur. Sauf que cela induirait une excitation importante et inutile des rotors dans le cas nominal (les N rotors sont



(a)



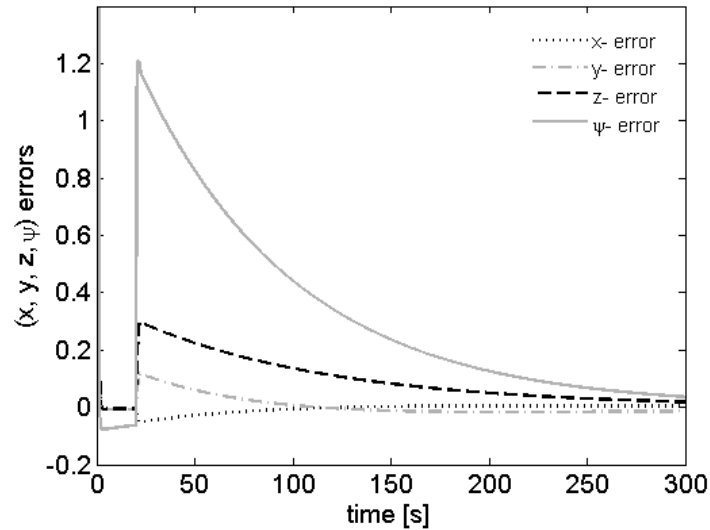
(b)

Figure 3.24. Scénario.2 : allocation des entrées de commande sur les actionneurs d'un N -rotor avant et après le déclenchement de la panne totale d'un rotor ; (a) moyenne des vitesses des rotors, (b) écart type des vitesses.

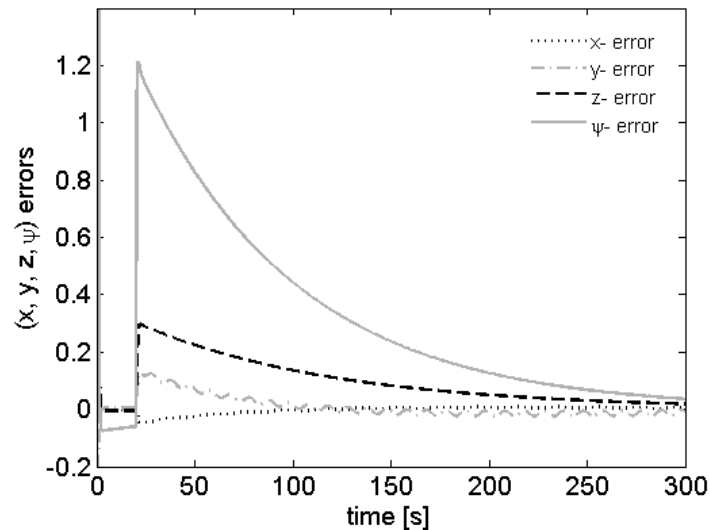
fonctionnels).

• Discussion du comportement de l'hexa-rotor

Dans les deux scénarios précédents, un seul rotor est mis en panne tant que tous les autres rotors sont fonctionnels. Les résultats présentés montrent que l'hexa-rotor considéré perd la contrôlabilité quand un rotor est en défaillance totale. Une analyse de la contrôlabilité d'un hexa-rotor de type PNPNP (où P indique que le rotor tourne dans le sens des aiguilles de la montre, et N indique qu'il tourne dans le sens inverse) soumis à une défaillance d'un seul rotor est présentée dans le travail [132]. Les auteurs ont démontré qu'un hexa-rotor de type PNPNP soumis à une défaillance d'un seul rotor est contrôlable si et seulement si : $K > \frac{5}{18}mg$, avec :



(a) scénario.1



(b) scénario.2

Figure 3.25. Erreur en x , y , z ([m]) et ψ ([deg]) pour le cas d'un octo-rotor avant et après déclenchement de la panne totale d'un rotor.

$K \in \mathbb{R}$: la force de poussée maximale pour chaque rotor ($K = b W_{max}^2 = \frac{U_{1max}}{6}$).

$m \in \mathbb{R}$: la masse totale de l'hexa-rotor.

$g \in \mathbb{R}$: la constante de gravité.

En considérant les paramètres physiques pris dans les simulations en présence de panne, nous trouvons :

$$K = 1.07(N) \Rightarrow K < \frac{5}{18}mg = 1.27(N)$$

On constate que l'hexa-rotor considéré est incontrôlable en présence d'échec total d'un rotor, et une approche de contrôle tolérante aux défauts est indisponible pour éviter l'écras-

sement de l'appareil.

La commande tolérante aux défauts des drones multi-rotors a attiré l'attention des chercheurs. Différentes stratégies sont proposées, dans le cas de défaillances totales ou partielles d'actionneurs dans le but d'offrir un atterrissage sécurisé [132] ou un vol normal [133–135]. Une nouvelle approche d'allocation de commande a été proposée par [136] afin d'améliorer les performances de vol d'un drone multi-rotor en offrant un ACS plus grand.

3.7 Conclusion

Dans ce chapitre, nous avons abordé le contrôle des drones multi-rotors pour assurer la stabilisation et le suivi de trajectoires. La caractéristique de sous actionnement de cette catégorie de drones impose le choix de quatre sorties à contrôler parmi les six sorties du système. Pour cela, nous avons proposé un schéma de commande hiérarchique pour le contrôle de la position 3D et de l'angle de lacet. La combinaison du contrôleur PID et de la technique Back-stepping avec action intégrale a permis d'obtenir de bonnes réponses pour le suivi de trajectoire et la stabilisation. Un autre point abordé dans ce chapitre est l'effet de la défaillance d'un actionneur sur le comportement global du drone multi-rotor. À l'exception du quadri-rotor (et de l'hexa-rotor dans certain cas), nous avons montré à travers les résultats obtenus que la panne totale d'un actionneur peut être compensée par les actionneurs restants. La force de poussée générée par l'actionneur perdu peut être répartie sur les actionneurs fonctionnels afin que le drone rattrape sa trajectoire initiale. La commande tolérante aux défauts n'a pas été abordée dans ce manuscrit.

Chapitre 4

Planification de chemins et évitement d'obstacles

Sommaire

4.1	Introduction	77
4.2	Planification de chemins par courbes de Bézier	78
4.2.1	Définitions et propriétés	78
4.2.2	Exemple de planification de chemin en 2D	80
4.2.3	Application pour des environnements 3D	81
4.3	Planification de chemin par squelettisation de l'environnement	87
4.3.1	Squelettisation de l'environnement	87
4.3.2	Calcul du chemin le plus court	90
4.3.3	Technique de lissage	90
4.4	Résultats numériques	92
4.4.1	résultats de l'approche proposée	92
4.4.2	Étude comparative	98
4.5	Conclusion	101

4.1 Introduction

Ce chapitre concerne le problème de planification de chemins 3D. Nous allons proposer deux méthodes pour planifier un chemin court et sûr (sans collision avec les obstacles) entre un point initial et un point final dans un environnement 3D connu. La première méthode est une extension en 3D d'une stratégie de planification de chemins par courbes de Bézier appliquée pour les robots terrestres. La deuxième méthode est originale et basée sur la squelettisation de l'environnement en tenant compte de la taille du drone [137].

4.2 Planification de chemins par courbes de Bézier

Les courbes de Bézier ont été utilisées dans la littérature comme un outil pour planifier un chemin 3D ou encore comme un moyen de lissage d'un chemin déjà calculé mais qui n'est pas lisse [138]. Cela est principalement grâce à la simplicité des courbes de Bézier et leurs caractéristiques aux points initial et final. Dans [139] et [140], les courbes de Bézier ont été utilisées dans une approche de planification de chemins pour une équipe de drones dans un environnement connu sans obstacles.

4.2.1 Définitions et propriétés

- Définition.1 (Courbe paramétrée)

On appelle courbe paramétrée, l'ensemble des points $M(t)$ de coordonnées $f(t)$ et $g(t)$ où t appartient à un intervalle I dans lequel f et g sont définies. On dit que la courbe admet pour représentation graphique :

$$\begin{cases} x(t) = f(t) \\ y(t) = g(t) \end{cases} \quad (4.1)$$

- Définition.2 (Courbe de Bézier)

On considère P_0, P_1, \dots, P_n ($n \geq 1$), $n + 1$ points. La courbe de Bézier d'ordre n associée à ces points est la courbe paramétrée C définie par $M(t)$ avec $t \in [0,1]$:

$$M(t) = B_0(t)P_0 + B_1(t)P_1 + \dots + B_n(t)P_n \quad (4.2)$$

Les coefficients B_i ($0 \leq i \leq n$) sont les *polynômes de Bernstein*. Les points P_i sont appelés les points de définition (ou de contrôle) de la courbe de Bézier.

- Définition.3 (Polynômes de Bernstein)

Les polynômes de Bernstein d'ordre n sont les polynômes :

$$B_{i,n}(t) = C_n^i t^i (1-t)^{n-i}, \quad i = 0, \dots, n \quad (4.3)$$

On rappelle que $C_n^i = \frac{n!}{i!(n-i)!}$. Lorsque n est fixé, on note simplement B_i au lieu de $B_{i,n}$.

Remarques. On note que :

- $B_{0,n}(t) = (1 - t)^n$ et $B_{n,n}(t) = t^n$.
- $B_{i,n}(t) \geq 0$ (positivité) et $\sum_{i=1}^n B_{i,n}(t) = 1$ (partition de l'unité).
- $B_{i,n}(t) = (1 - t)B_{i,n-1}(t) + t B_{i-1,n-1}(t)$ (relation de récurrence).

- Propriétés de la courbe de Bézier

Une courbe de Bézier d'ordre n associée aux points P_0, \dots, P_n satisfait les propriétés suivantes :

- Elle passe par les points P_0 et P_n , mais en général, elle ne passe pas par les autres points. On a :

$$\begin{cases} M(0) = P_0 \\ M(1) = P_n \end{cases} \quad (4.4)$$

- Pour $n = 1$, la courbe C est égale au segment de droite $[P_0, P_1]$.
- la courbe C est tangente en P_0 au segment de droite $[P_0, P_1]$, et en P_n au segment de droite $[P_{n-1}, P_n]$. On a :

$$\begin{cases} \dot{M}(0) = n(P_1 - P_0) = n\overrightarrow{P_0P_1} \\ \dot{M}(1) = n(P_n - P_{n-1}) = n\overrightarrow{P_{n-1}P_n} \end{cases} \quad (4.5)$$

- Elle se trouve à l'intérieur de l'enveloppe convexe des points de contrôle P_i .
- Elle est infiniment dérivable.

On note que dans un problème de planification de chemins, les propriétés (4.4) et (4.5) peuvent être utilisées pour définir deux points : point de départ et point cible.

Remarque. Lorsqu'on parle de courbe de Bézier dans un environnement 3D, P_i est le point (x_i, y_i, z_i) . Cela signifie que le point $M(t)$ a comme coordonnées :

$$x(t) = \sum_{i=0}^n B_i(t)x_i, \quad y(t) = \sum_{i=0}^n B_i(t)y_i \quad \text{et} \quad z(t) = \sum_{i=0}^n B_i(t)z_i.$$

- Courbe de Bézier d'ordre 3

Appelée aussi courbe cubique, elle est définie par quatre points P_0, P_1, P_2 et P_3 . Comme le montre la figure (4.1), les points P_1 et P_2 définissent le profil général de la courbe. Le polygone de contrôle est représenté en pointillés et est constitué des trois segments $\overrightarrow{P_0P_1}$, $\overrightarrow{P_1P_2}$ et $\overrightarrow{P_2P_3}$.

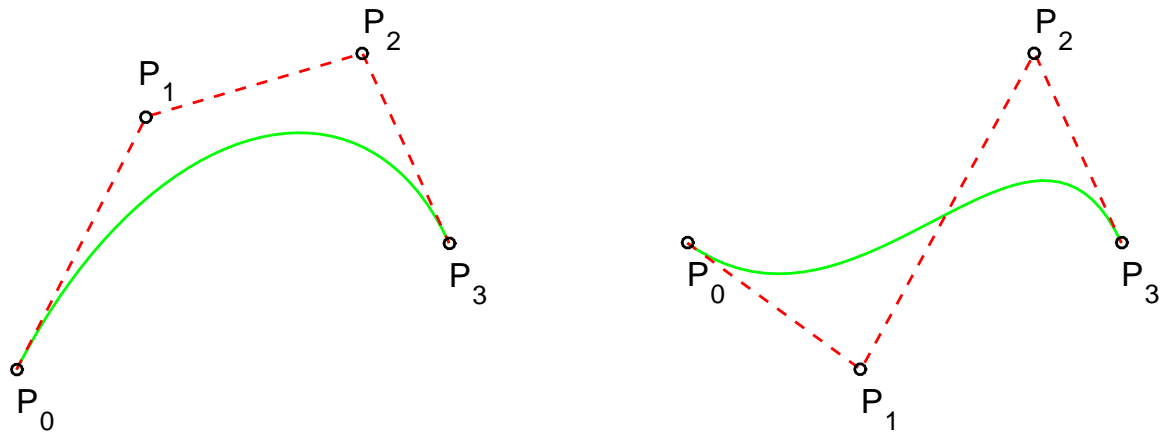


Figure 4.1. Courbe de Bézier d'ordre 3.

La forme paramétrique d'une courbe de Bézier d'ordre 3 s'écrit comme suit :

$$M(t) = (1-t)^3 P_0 + 3t(1-t)^2 P_1 + 3t^2(1-t) P_2 + t^3 P_3 \quad (4.6)$$

On peut facilement démontrer la propriété de partition de l'unité des polynômes de Bernstein :

$$(1-t)^3 + 3t(1-t)^2 + 3t^2(1-t) + t^3 = [t + (1-t)]^3 = 1^3 = 1 \quad (4.7)$$

4.2.2 Exemple de planification de chemin en 2D

Dans cette section, on va présenter une stratégie de planification de chemins basée sur les courbes de Bézier et le problème d'optimisation [141]. Cette approche a été améliorée et validée par des tests expérimentaux sur un robot mobile de l'équipe Mécatronique de l'Université de Limoges. Ce travail était le point de départ de notre étude sur la planification de chemins d'un multi-rotor.

Dans l'étude citée en référence, un fauteuil roulant est en mesure de suivre le chemin conçu qui passe à travers la porte, sans entrer en collision avec le mur de la porte ainsi que d'autres obstacles. Le problème de passage de porte d'un fauteuil roulant peut donc être considéré comme un problème de planification de chemins.

Comme on le voit dans la figure (4.2), le fauteuil roulant dans le couloir cherche à traverser la porte sur son côté gauche et atteindre la cible dans la chambre. Pour atteindre cet objectif, la stratégie de passage de la porte devrait respecter les critères suivants :

1. Le chemin planifié doit être lisse, sans oscillation, comme le montre la figure (4.2), où la ligne rouge est le chemin désiré pour un cas sans obstacles, et la ligne pointillée bleue représente le chemin désiré en évitant les obstacles.
2. En raison des contraintes mécaniques du fauteuil roulant, la courbure du chemin doit être limitée à certains intervalles.

3. Le cap du fauteuil roulant est perpendiculaire au plan de la porte lorsque le fauteuil roulant arrive au point P_d .

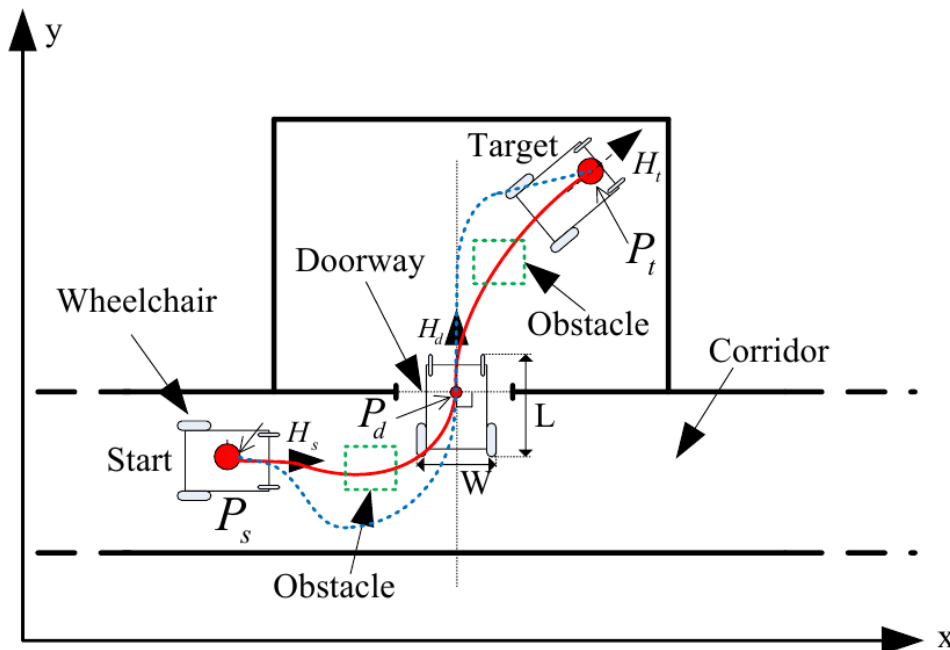


Figure 4.2. Exemple de passage de la porte dans la robotique mobile.

Le processus de génération de chemins proposé consiste d'abord à diviser le chemin $\widehat{P_s P_d P_t}$ en deux parties $\widehat{P_s P_d}$ et $\widehat{P_d P_t}$, puis générer les chemins désirés $\widehat{P_s P_d}$ et $\widehat{P_d P_t}$ basés indépendamment sur la courbe de Bézier afin d'assurer que chaque chemin vérifie les critères annoncés ci-dessus.

L'idée est de déterminer les points de contrôle qui définissent la forme de la courbe de Bézier associée à chacun des segments $\widehat{P_s P_d}$ et $\widehat{P_d P_t}$. On note P_{ci} et P_{ri} les points de contrôle dans le couloir et la chambre respectivement. Comme P_{c0} , P_{cn} , P_{r0} et P_{rn} correspondent respectivement à P_s , P_d , P_d et P_t qui sont connus, les points P_{ci} et P_{ri} avec $i = 1 : n - 1$ sont à déterminer. Dans la figure (4.2), H_s , H_d et H_t représentent les orientations du robot en positions P_s , P_d et P_t respectivement. Pour des raisons de gain en temps de calcul, l'ordre de la courbe de Bézier qui définit le chemin $\widehat{P_s P_d}$ et $\widehat{P_d P_t}$ est fixé à 3. Pour déterminer les points P_{c1} , P_{c2} , P_{r1} et P_{r2} , une méthode d'optimisation est utilisée en minimisant la courbure de la courbe de Bézier, et en respectant les critères décrits ci-dessus avec les orientations imposées en P_s , P_d et P_t .

4.2.3 Application pour des environnements 3D

Dans un environnement 3D avec obstacles, le vol d'un drone ne peut pas s'effectuer sans trouver un ensemble d'états admissibles lui permettant d'atteindre son objectif. le problème de planification de chemins fait référence à la recherche d'un ensemble de configurations dans l'espace navigable (libre) depuis un état de départ P_s vers un état d'arrivée P_T . L'ensemble de ces configurations est appelé chemin. Nous utilisons ici les courbes de

Bézier d'ordre 3 comme outil de représentation du chemin généré. Le point de contrôle P_0 et le point de départ P_S sont confondus, ainsi que les point P_3 et le point d'arrivée P_T . Les deux points de contrôle médians P_1 et P_2 de la courbe de Bézier sont à déterminer. Le problème de planification de chemin est formulé ici comme problème d'optimisation en minimisant la courbure et la dérivée de la courbure du chemin calculé :

$$\min_{P_1, P_2} \int_0^1 \left[(k(t))^2 + (\dot{k}(t))^2 \right] dt \quad (4.8)$$

avec $k(t)$ est la courbure de la courbe de Bézier en fonction du temps,

$$k(t) = \frac{\sqrt{(\dot{x}\ddot{y} - \dot{y}\ddot{x})^2 + (\dot{y}\ddot{z} - \dot{z}\ddot{y})^2 + (\dot{z}\ddot{x} - \dot{x}\ddot{z})^2}}{(\dot{x}^2 + \dot{y}^2 + \dot{z}^2)^{3/2}} \quad (4.9)$$

et $\dot{k}(t)$ est sa première dérivée.

$$\begin{aligned} \dot{k}(t) = & \frac{(\dot{x}\ddot{y} - \dot{y}\ddot{x})(\dot{x}\ddot{y} - \dot{y}\ddot{x}) + (\dot{y}\ddot{z} - \dot{z}\ddot{y})(\dot{y}\ddot{z} - \dot{z}\ddot{y}) + (\dot{z}\ddot{x} - \dot{x}\ddot{z})(\dot{z}\ddot{x} - \dot{x}\ddot{z})}{\sqrt{(\dot{x}\ddot{y} - \dot{y}\ddot{x})^2 + (\dot{y}\ddot{z} - \dot{z}\ddot{y})^2 + (\dot{z}\ddot{x} - \dot{x}\ddot{z})^2} (\dot{x}^2 + \dot{y}^2 + \dot{z}^2)^{3/2}} \\ & - 3 \frac{\sqrt{(\dot{x}\ddot{y} - \dot{y}\ddot{x})^2 + (\dot{y}\ddot{z} - \dot{z}\ddot{y})^2 + (\dot{z}\ddot{x} - \dot{x}\ddot{z})^2} \cdot (\dot{x}\ddot{x} + \dot{y}\ddot{y} + \dot{z}\ddot{z})}{(\dot{x}^2 + \dot{y}^2 + \dot{z}^2)^{5/2}} \end{aligned} \quad (4.10)$$

Ce travail a été validé sous l'environnement Matlab/Simulink en utilisant la fonction *fmincon* de la boîte à outil Optimization Toolbox. L'environnement est supposé connu avec des obstacles statiques. Certaines contraintes sont imposées dans le processus d'optimisation afin d'assurer les exigences liées à l'environnement et à la souplesse du chemin calculé.

- Le chemin doit être généré à l'intérieur de l'espace navigable.
- Les orientations aux points P_S et P_T peuvent être exigées. La courbe est tangente au polygone de contrôle dans les points d'extrémité. Pour une courbe de Bézier cubique $M(t)$ avec $t \in [0,1]$, $\dot{M}(0) = 3\overrightarrow{P_0P_1}$ et $\dot{M}(1) = 3\overrightarrow{P_2P_3}$.

Soient $\overrightarrow{V_S}$ et $\overrightarrow{V_T}$ les directions aux points P_S et P_T respectivement. P_1 est à chercher dans la direction de $\overrightarrow{V_S}$. P_2 est à chercher dans la direction opposée de $\overrightarrow{V_T}$. Cela signifie que $\overrightarrow{P_0P_1}$ et $\overrightarrow{V_S}$ sont colinéaires et ayant le même sens, de même pour $\overrightarrow{P_2P_3}$ et $\overrightarrow{V_T}$. L'angle entre chaque deux vecteurs est donc égale à zéro. Leur produit scalaire donne alors :

$$\overrightarrow{V_S} \cdot \overrightarrow{P_0P_1} = \|\overrightarrow{V_S}\| \cdot \|\overrightarrow{P_0P_1}\| \cdot \cos(0) \quad (4.11)$$

Cela implique que :

$$\frac{\vec{V}_S \cdot \overrightarrow{P_0P_1}}{\|\vec{V}_S\| \cdot \|\overrightarrow{P_0P_1}\|} = \cos(0) \quad (4.12)$$

Les contraintes liées aux directions en points P_S et P_T sont données alors comme suit :

$$\begin{cases} \arccos\left(\frac{\vec{V}_S \cdot \overrightarrow{P_0P_1}}{\|\vec{V}_S\| \cdot \|\overrightarrow{P_0P_1}\|}\right) = 0 \\ \arccos\left(\frac{\vec{V}_T \cdot \overrightarrow{P_2P_3}}{\|\vec{V}_T\| \cdot \|\overrightarrow{P_2P_3}\|}\right) = 0 \end{cases} \quad (4.13)$$

- Le chemin planifié doit éviter les obstacles. Soit (x_0, y_0, z_0) le centre de gravité d'un obstacle. La courbe de Bézier doit vérifier :

$$(x(t_k) - x_0)^2 + (y(t_k) - y_0)^2 + (z(t_k) - z_0)^2 > D_t \quad (4.14)$$

où $x(t_k)$, $y(t_k)$ et $z(t_k)$ sont les points de la courbe de Bézier. D_t est la distance minimale tolérée entre un point dans la courbe de Bézier et le centre de gravité de l'obstacle. Elle dépend des dimensions de l'obstacle, de la taille du drone et d'une distance prise comme marge de sécurité.

- La courbe de Bézier représentant le chemin planifié (ou seulement une partie) ne doit pas contenir des boucles produites par un mauvais placement des points de contrôle internes. Un cas de boucle indésirable dans une courbe de Bézier est représenté sur la figure (4.3).

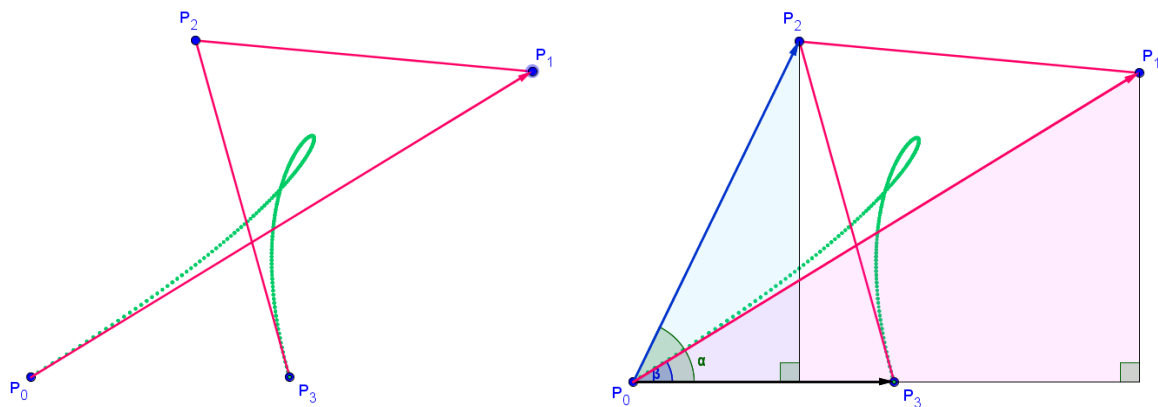


Figure 4.3. Exemple d'une courbe de Bézier avec boucle.

Pour éliminer ces boucles dans une courbe de Bézier cubique, deux exigences doivent être considérées. Premièrement, l'angle α entre $\overrightarrow{P_0P_3}$ et $\overrightarrow{P_0P_2}$ (sur la partie droite de la

figure (4.3)) doit être aigu, c-à-d, $\cos \alpha > 0$. Cela signifie que le produit scalaire de $\overrightarrow{P_0P_3}$ et $\overrightarrow{P_0P_2}$ soit positif. La deuxième exigence est que la projection de $\overrightarrow{P_0P_2}$ sur $\overrightarrow{P_0P_3}$ doit être supérieure de la projection de $\overrightarrow{P_0P_1}$ sur $\overrightarrow{P_0P_3}$.

Les contraintes suivantes doivent alors être ajoutées.

$$\begin{cases} \overrightarrow{P_0P_3} \cdot \overrightarrow{P_0P_2} \geq 0 \\ \overrightarrow{P_0P_3} \cdot \overrightarrow{P_0P_2} - \overrightarrow{P_0P_3} \cdot \overrightarrow{P_0P_1} \geq 0 \end{cases} \quad (4.15)$$

La résolution de ce problème d'optimisation avec contraintes permet la planification d'un chemin représenté par une courbe de Bézier. Cela peut être une tâche compliquée dans un environnement encombré avec un nombre considérable d'obstacles. Grâce aux propriétés des courbes de Bézier, deux courbes peuvent être raccordées formant une courbe globale. Dans ce cas, 5 points de contrôle sont à chercher : deux points internes par courbe et un point de raccordement. Dans le cas général, pour l courbes de Bézier à tracer, il y a $3l - 1$ points de contrôle à chercher. Le point initial de la première courbe de Bézier et le point final de la dernière sont respectivement P_S et P_T ; le point de départ et d'arrivée du chemin à générer. Dans ce qui suit, on note $P_0^k, P_1^k, P_2^k, P_3^k$ les points de contrôle de la $k^{\text{ième}}$ courbe de Bézier. Les contraintes (4.13) liées aux directions en P_S et P_T deviennent :

$$\begin{cases} \arccos \left(\frac{\overrightarrow{V_S} \cdot \overrightarrow{P_0^1 P_1^1}}{\|\overrightarrow{V_S}\| \cdot \|\overrightarrow{P_0^1 P_1^1}\|} \right) = 0 \\ \arccos \left(\frac{\overrightarrow{V_T} \cdot \overrightarrow{P_2^m P_3^m}}{\|\overrightarrow{V_T}\| \cdot \|\overrightarrow{P_2^m P_3^m}\|} \right) = 0 \end{cases} \quad (4.16)$$

Pour la première courbe de Bézier, les contraintes (4.15) deviennent :

$$\begin{cases} \overrightarrow{P_0^1 P_3^1} \cdot \overrightarrow{P_0^1 P_2^1} \geq 0 \\ \overrightarrow{P_0^1 P_3^1} \cdot \overrightarrow{P_0^1 P_2^1} - \overrightarrow{P_0^1 P_3^1} \cdot \overrightarrow{P_0^1 P_1^1} \geq 0 \end{cases} \quad (4.17)$$

De plus, pour chaque courbe de Bézier supplémentaire, on rajoute les contraintes suivantes pour éliminer les boucles :

$$\begin{cases} \forall k = 2, \dots, m : \\ \overrightarrow{P_0^{k-1} P_3^{k-1}} \cdot \overrightarrow{P_0^k P_2^k} \geq 0 \\ \overrightarrow{P_0^{k-1} P_3^{k-1}} \cdot \overrightarrow{P_0^k P_2^k} - \overrightarrow{P_0^{k-1} P_3^{k-1}} \cdot \overrightarrow{P_0^k P_1^k} \geq 0 \end{cases} \quad (4.18)$$

La continuité doit être assurée au point de raccordement de chaque couple de courbes de Bézier successives. La continuité de classe C^1 , représentée par le schéma de la figure (4.4), est considérée ici. Elle est vérifiée si et seulement si les points extrêmes sont confondus et situés au milieu du point qui les précède et de celui qui les suit.

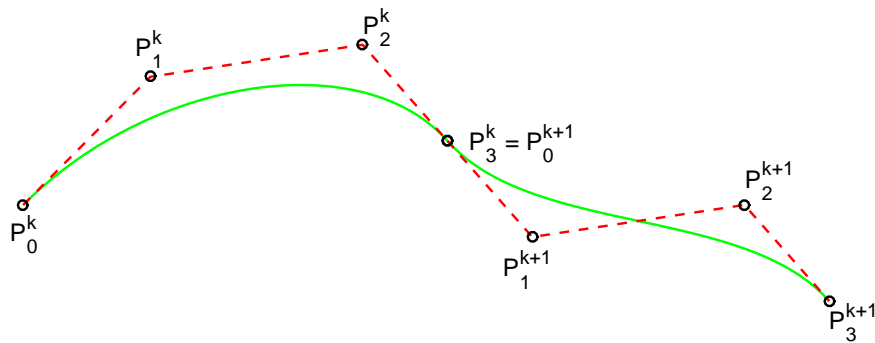


Figure 4.4. Continuité de classe C^1 entre deux courbes de Bézier.

Elle peut être traduite en contraintes dans le problème d'optimisation. Soient $P_0^k, P_1^k, P_2^k, P_3^k$ les points de contrôle de la $k^{\text{ième}}$ courbe de Bézier. La continuité de classe C^1 est vérifiée dans le point de raccordement de deux courbes de Bézier cubiques successives si :

$$\left\{ \begin{array}{l} \forall k = 2, \dots, m : \\ P_3^{k-1} = P_0^k \\ \overrightarrow{P_2^{k-1}P_3^{k-1}} = \overrightarrow{P_0^kP_1^k} \end{array} \right. \quad (4.19)$$

Même en respectant toutes les contraintes précédentes, un cas très particulier de boucles peut se produire au point de raccordement de deux courbes successives, comme le montre la figure (4.5).

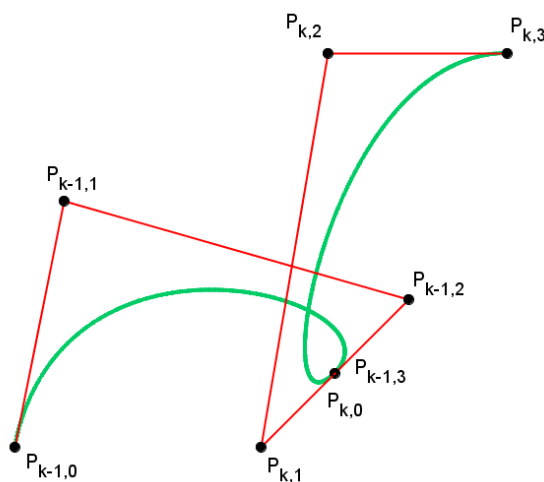


Figure 4.5. Exemple de boucle au point de raccordement de deux courbes de Bézier cubiques.

Pour éliminer ce cas entre deux courbes de Bézier successives, il faut imposer que l'angle entre $\overrightarrow{P_0^{k-1}P_3^{k-1}}$ et $\overrightarrow{P_0^kP_1^k}$ soit aigu, le cosinus de cet angle soit donc positif. La contrainte suivante doit être ajoutée.

$$\begin{cases} \forall k = 2, \dots, m : \\ \overrightarrow{P_0^{k-1}P_3^{k-1}} \cdot \overrightarrow{P_0^kP_1^k} \geq 0, & (P_3^{k-1} = P_0^k) \end{cases} \quad (4.20)$$

Les figures (4.6) et (4.7) représentent les résultats obtenus par l'approche proposée implémentée sous Matlab pour deux scénarios.

- **Scénario 1** : $P_S = (0,5,0)m$, $P_T = (9,5.5,1)m$, $\vec{V}_S = (0,1,1)$ et $\vec{V}_T = (1,0,0)$.
- **Scénario 2** : $P_S = (.6,1,0)m$, $P_T = (9,5.5,1)m$, $\vec{V}_S = (1,0,1)$ et $\vec{V}_T = (1,0,0)$.

P_S et P_T représentent respectivement le point de départ et le point d'arrivée. \vec{V}_S et \vec{V}_T sont les directions aux points P_S et P_T respectivement. Le chemin planifié est représenté en vert, et le polygone de contrôle avec une ligne pointillée de couleur magenta.

Dans cette approche, la taille du drone n'est pas prise en compte lors du calcul du chemin. Nous supposant que le drone est de taille inférieure à la largeur de la porte, et qu'il puisse passer la porte sans risque de collision.

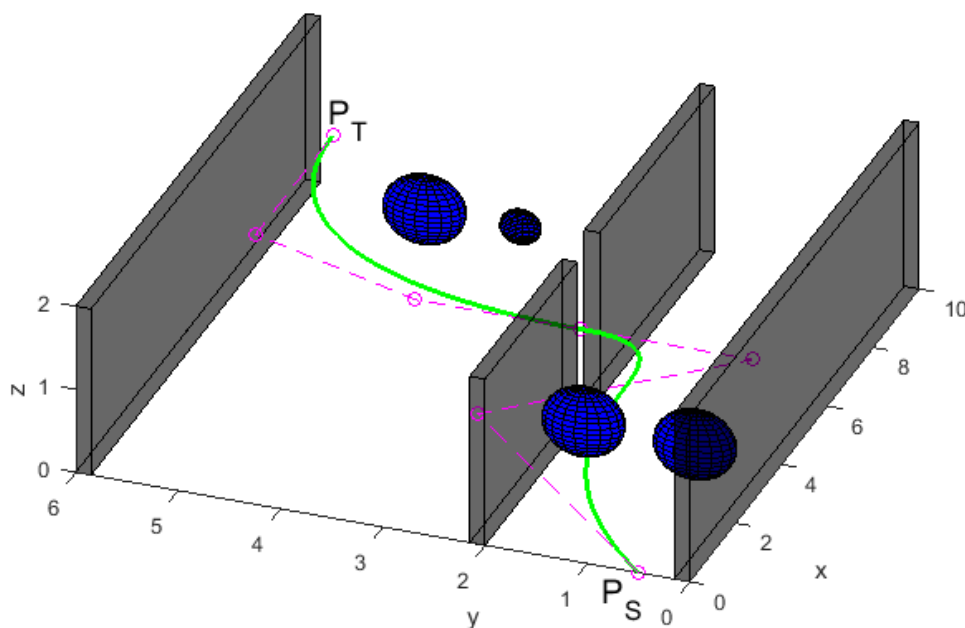


Figure 4.6. Scénario 1. Exemple de chemin planifié par courbes de Bézier.

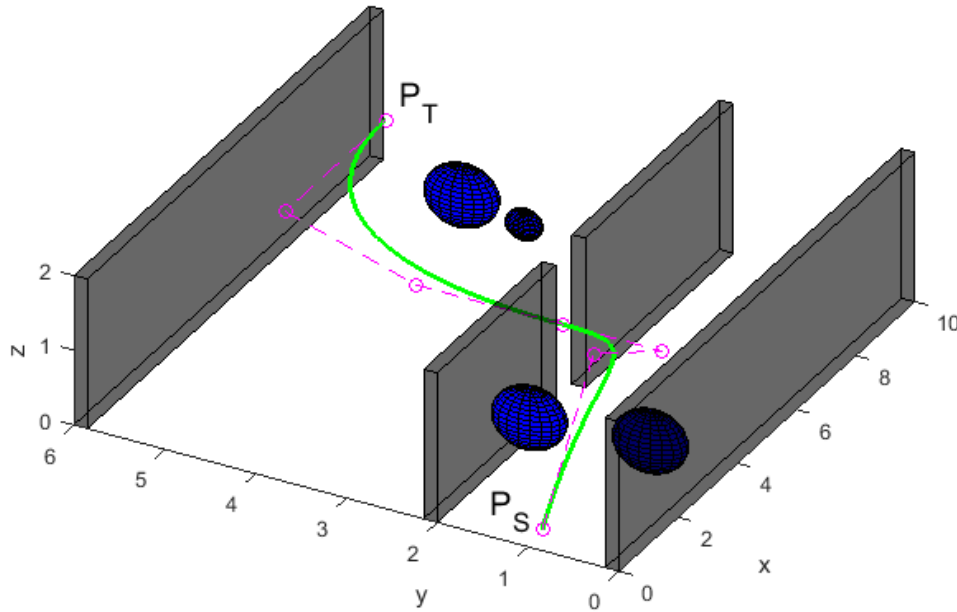


Figure 4.7. Scénario 2. Exemple de chemin planifié par courbes de Bézier.

4.3 Planification de chemin par squelettisation de l'environnement

Dans cette section, nous proposons une nouvelle approche de planification de chemins 3D. Cette approche est basée sur la squelettisation de l'environnement tout en prenant en considération la dimension du drone. Cela permet d'éliminer les passages étroits dans l'espace navigable et garantit une navigation sûre sans risque de collision. Cette partie représente un des apports de ce travail de thèse.

4.3.1 Squelettisation de l'environnement

Soit $\mathcal{X}_{\mathbb{R}}$ un sous-ensemble de \mathbb{R}^n et $\mathcal{X}_{\mathbb{R}}^c$ son complément. Initialement défini par Blum [142], l'axe médian réel $S_{\mathbb{R}}$ de $\mathcal{X}_{\mathbb{R}}$ (aussi appelé squelette) est l'ensemble des points $\mathbf{x} \in \mathcal{X}_{\mathbb{R}}$ à égale distance de leurs deux voisins les plus proches dans $\mathcal{X}_{\mathbb{R}}^c$. Comme indiqué dans [143], cette définition est conceptuellement équivalente au [Graphe de Voronoï généralisé \(GVG\)](#) (en anglais : Generalized Voronoï Graph) d'un environnement, et est donc utilisée ici comme une approximation précise. Dans le cas spécifique des formes 3D, il génère un ensemble de surfaces, et est donc appelé surface médiane ou également [Filtered Medial Surface \(FMS\)](#).

Pour chaque point de forme $\mathbf{x} \in \mathcal{X}_{\mathbb{R}}$, soit $\mathcal{P}_{\mathbf{x}} = \operatorname{argmin}_{\mathbf{y} \in \mathcal{X}_{\mathbb{R}}^c} d(\mathbf{x}, \mathbf{y})$ l'ensemble de ses points les plus proches sur $\mathcal{X}_{\mathbb{R}}^c$, appelé projection de \mathbf{x} . La distance euclidienne correspondante est notée $D_{\mathbf{x}} = d(\mathbf{x}, \mathcal{P}_{\mathbf{x}})$. En conséquence, le squelette $S_{\mathbb{R}}(\mathcal{X}_{\mathbb{R}})$ peut alors être défini par :

$$S_{\mathbb{R}}(\mathcal{X}_{\mathbb{R}}) = \{\mathbf{x} \in \mathcal{X}_{\mathbb{R}}, |\mathcal{P}_{\mathbf{x}}| \geq 2\} \quad (4.21)$$

Pour estimer et filtrer efficacement le **GVG** de l'environnement, deux préoccupations importantes doivent être prises en compte. Tout d'abord, l'espace navigable est exprimé dans ce travail dans un domaine discret (la grille d'occupation $\mathcal{F} \cup \mathcal{O}$). Appliquer (4.21) directement sur \mathcal{F} échouerait, puisque dans la plupart des cas, $|\mathcal{P}_{\mathbf{x}}| = 1$ même si une cellule donnée \mathbf{x} appartient réellement au **GVG** de l'environnement.

Par conséquent, la définition de l'axe médian discret doit être légèrement modifiée. Bien que plusieurs solutions aient été proposées au cours des années, ce travail est basé sur l'axe médian (**Integer Medial Axis (IMA)**) [144]. Soit $\mathcal{P}_{\mathbf{x}}^*$ la projection réduite de \mathbf{x} , correspondant au premier élément de $\mathcal{P}_{\mathbf{x}}$ par rapport à un ordre lexical. Étant donné une forme numérique $\mathcal{X}_{\mathbb{Z}} \subset \mathbb{Z}^n$ et son complément $\mathcal{X}_{\mathbb{Z}}^c$, le squelette discret $S_{\mathbb{Z}}$ de $\mathcal{X}_{\mathbb{Z}}$ peut alors être exprimé comme :

$$S_{\mathbb{Z}}(\mathcal{X}_{\mathbb{Z}}) = \{\mathbf{x} \in \mathcal{X}_{\mathbb{Z}}, \exists \mathbf{y} \in N_{\mathbf{x}} \mid \begin{aligned} d(\mathcal{P}_{\mathbf{x}}^*, \mathcal{P}_{\mathbf{y}}^*) &> 1, \\ d(\mathbf{m}, \mathcal{P}_{\mathbf{x}}^*) &\geq d(\mathbf{m}, \mathcal{P}_{\mathbf{y}}^*) \end{aligned}\} \quad (4.22)$$

où $N_{\mathbf{x}} = \{\mathbf{y} \in \mathcal{X}_{\mathbb{Z}}, d(\mathbf{x}, \mathbf{y}) = 1\}$ est l'ensemble des voisins directs de \mathbf{x} , et \mathbf{m} est le point médian du segment de ligne $[\mathbf{x}, \mathbf{y}]$. La première condition mesure la distance (Euclidienne) entre les projections de \mathbf{x} et \mathbf{y} pour supprimer les points squelettiques liés à l'aliasing, tandis que la seconde condition est utilisée pour produire un squelette aussi fin que possible, car elle ne sélectionne que la meilleure approximation au squelette réel. Notons qu'il est théoriquement prouvé [144] que cette définition se rapproche efficacement de l'axe médian de la forme correspondante dans \mathbb{R}^n tout en supprimant les points squelettiques induits par la discrétisation.

La deuxième préoccupation majeure est le fait que les squelettes sont très sensibles aux petites déformations et tendent à produire des branches parasites. Le **IMA** se rapproche efficacement de l'axe médian réel de Blum, mais n'est pas conçu pour filtrer les branches bruyantes. Nous proposons dans ce travail de traiter ce problème en utilisant le **Delta Medial Axis (DMA)** [145], une stratégie d'élagage en temps réel capable d'identifier et de rejeter les branches squelettiques parasites en temps réel. L'algorithme a été initialement conçu pour des formes $2D$, mais peut être étendu directement à des dimensions plus élevées.

Les concepts de base de l'algorithme, illustrés sur la figure (4.8) sont les suivants :

- Étant donné $\mathbf{x} \in \mathcal{X}_{\mathbb{Z}}, \mathbf{y} \in N_{\mathbf{x}}$, soit $r_{\mathbf{x}, \mathbf{y}}$ dénote le rayon du disque le plus grand disjoint de \mathcal{X}^c et centré le long de $[\mathcal{P}_{\mathbf{x}}, \mathcal{P}_{\mathbf{y}}]$:

$$r_{\mathbf{x}, \mathbf{y}} = \sup\{\mathcal{D}_{\mathbf{z}} \mid \mathbf{z} \in [\mathcal{P}_{\mathbf{x}}, \mathcal{P}_{\mathbf{y}}]\} \quad (4.23)$$

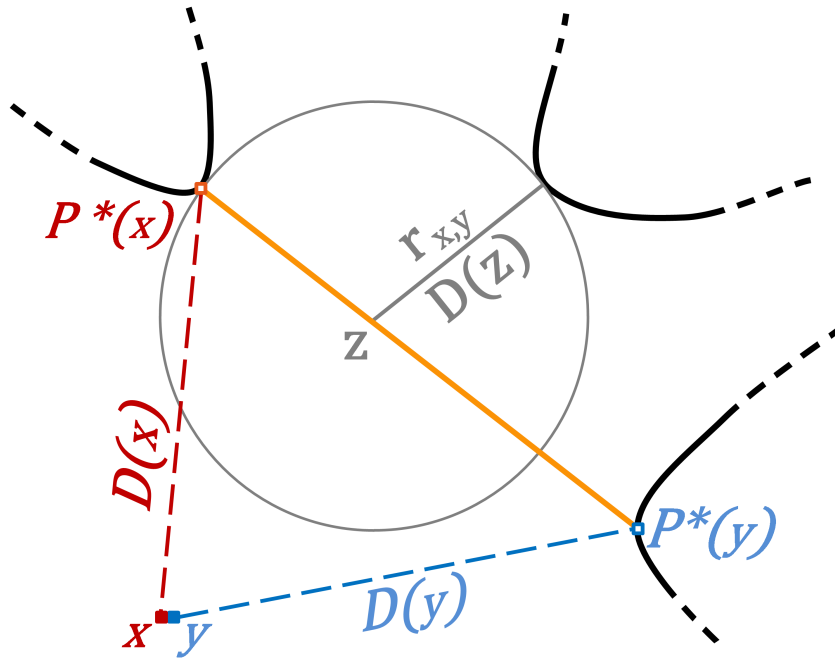


Figure 4.8. Représentation illustrative des concepts impliqués dans le calcul du DMA.

- Le critère d'élagage $\delta^*(\mathbf{x})$ est alors défini comme la valeur maximale de $r_{\mathbf{x},\mathbf{y}}$ parmi ses voisins directs $\mathbf{y} \in N_{\mathbf{x}}$:

$$\delta^*(\mathbf{x}) = \sup\{r_{\mathbf{x},\mathbf{y}} \mid \mathbf{y} \in N_{\mathbf{x}}, d(\mathbf{m}, \mathcal{P}_{\mathbf{x}}) \geq d(\mathbf{m}, \mathcal{P}_{\mathbf{y}})\} \quad (4.24)$$

Notez que la deuxième condition est similaire à l'algorithme **IMA**. Elle contribue à générer un squelette mince, et donc une meilleure approximation du **GVG**. Le **DMA** est alors défini comme suit :

- Définition (Axe Médian Delta)

Soit $\delta \in \mathbb{R}_+^*$, et $\mathcal{X}_{\mathcal{Z}} \subset \mathbb{Z}^n$. L'Axe Médian Delta $MA_{\delta}(\mathcal{X}_{\mathcal{Z}})$ de $\mathcal{X}_{\mathcal{Z}}$ est l'ensemble de points $\mathbf{x} \in \mathcal{X}_{\mathcal{Z}}$ avec une valeur delta maximale $\delta^*(\mathbf{x})$ d'au moins δ :

$$MA_{\delta}(\mathcal{X}_{\mathcal{Z}}) = \{\mathbf{x} \in \mathcal{X}_{\mathcal{Z}} \mid \delta_M(\mathbf{x}) \geq \delta\} \quad (4.25)$$

Intuitivement, le **DMA** rejette les branches squelettiques générées par des déformations de forme inférieures au paramètre δ . Basé sur la dimension du robot, il peut donc être réglé pour ne conserver que des branches squelettes (donc des chemins navigables du **GVG**) suffisamment grandes pour assurer une navigation sûre.

Appliqué sur la représentation de l'environnement $\mathcal{F} \cup \mathcal{O}$, le **GVG** est donc finalement approché et filtré pour une taille donnée de drone δ , en calculant le **FMS** $MA_{\delta}(\mathcal{F})$. Des exemples de différentes valeurs de δ sont donnés dans la section 4.4.

4.3.2 Calcul du chemin le plus court

Le **FMS** définit un sous-espace de l'espace navigable qui garantit des chemins sûrs en fonction de la dimension de l'UAV. L'algorithme A-star est ensuite utilisé pour calculer un chemin 3D affiné sur le **FMS**. Il permet de trouver le chemin avec le plus petit coût à partir d'un ensemble de chemins candidats menant à une destination. Le principe de base de l'algorithme A* consiste à définir une fonction coût $F(n)$:

$$F(n) = G(n) + H(n) \quad (4.26)$$

où n est le nœud actuel, $G(n)$ dénote le coût de déplacement du point de départ (nœud) P_S au nœud actuel n et $H(n)$ (fonction heuristique) indique le coût estimé pour atteindre le point cible (nœud) P_T à partir du nœud courant n .

Étant donné les coordonnées de P_S et P_T , le but ici est de calculer le chemin le plus court sur le **FMS**. La fonction coût est alors la distance euclidienne dans l'espace 3D. Une distance tolérable D_t doit être définie pour considérer deux points dans le **FMS** connectés. Dans une grille d'occupation 3D, l'algorithme A* utilise la recherche restreinte à 26 connectivités (donnant les cellules les plus proches dans un environnement 3D), ou dans tous les angles pour certaines variantes de l'A*. Ici, la recherche avance sur le **FMS** en se basant sur D_t . Pour un nœud courant, les voisins sont déterminés en fonction de leur distance au nœud actuel. Ils seront examinés si leur distance au nœud actuel est inférieure ou égale à D_t .

Le chemin généré est un ensemble de points de passage ordonnés définis sur le **FMS** partant du point de départ P_S jusqu'au point d'arrivée P_T . Les résultats obtenus sont représentés sur les figures (4.11)–(4.13) de la sections 4.4 pour deux cas différents de δ .

4.3.3 Technique de lissage

La question du lissage du chemin obtenu est formulé comme un problème d'optimisation avec des contraintes. L'objectif est de calculer un chemin approximatif au chemin réel qui permet au drone de faire le moins de manœuvres. Notez qu'il existe un compromis entre la faisabilité du chemin et son optimalité. En lissant, l'optimalité du trajet en termes de sécurité et de longueur de trajet peut se dégrader. L'idée que nous proposons ici est de définir un corridor de sécurité autour du chemin d'origine comme suit :

$$(Q_j - p(t_j)) \leq (D_j - \delta) \quad (4.27)$$

avec Q_j sont les données réelles (avant lissage), $p(t_j)$ sont des valeurs de données approximées calculées par l'interpolation de courbes de Bézier, et D_j est la distance Euclidienne à l'obstacle le plus proche dans chaque point de chemin original (avant lissage).

Le chemin lissé formé par les courbes de Bézier connectées doit être dans le couloir de sécurité. Pour cela, nous devons déterminer les points de contrôle qui le définissent par la résolution du problème d'optimisation. La propriété d'interpolation du point final des courbes de Bézier garantit que le chemin débute par P_S et se termine par P_T . La fonction objective à minimiser est la moyenne de la distance au carré entre les données réelles et les données lissées.

Supposons qu'il y ait m points de passage Q_j ($j = 1 : m$) dans les données réelles. Pour les données lissées $p(t_j)$, nous notons que $0 \leq t_j \leq 1$ avec division uniforme, donc $Q_1 = p(0) = P_S$ et $Q_m = p(1) = P_T$. La fonction objectif est donnée par :

$$J = \min_{p(t_j)} \left(\frac{1}{m-2} \sqrt{\sum_{j=2}^{m-1} (Q_j - p(t_j))^2} \right) \quad (4.28)$$

Les données d'origine peuvent être divisées en plusieurs segments en donnant l'ensemble initial de points de rupture. Chaque segment sera approximé par une courbe de Bézier cubique en identifiant les points de contrôle intermédiaires. S'il y a l segments (donc $l - 1$ points de rupture), en résolvant le problème d'optimisation (voir Fig. 4.9), $3l - 1$ points de contrôle doivent être déterminés entre P_S et P_T pour définir les l courbes de Bézier. Il est important de préciser que les points de contrôle joignant les courbes de Bézier ne sont pas fixes et doivent également être déterminés.

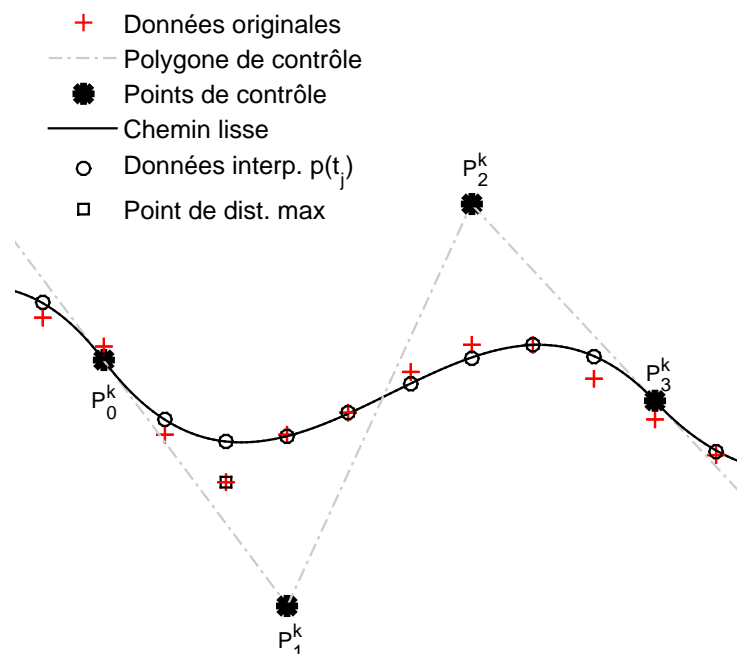


Figure 4.9. illustration du principe de l'approche de lissage sur la $k^{\text{ième}}$ courbe de Bézier.

Dans l'approche de lissage que nous proposons ici, nous utilisons un raccordement de classe C^1 entre les courbes de Bézier. L'orientation du drone dans le point de départ et le point cible peut être imposée dans le chemin planifié. Grâce à la propriété de la tangente

aux points d'extrémité de la courbe de Bézier, les directions aux points P_S et P_T peuvent être introduites dans les contraintes d'optimisation.

Dans le processus d'optimisation, nous considérons le cas où aucune solution n'est trouvée tout en respectant la contrainte que le chemin lissé ne doit pas être en dehors du couloir de sécurité. Dans ce cas, si aucune solution vérifiant les contraintes n'est trouvée, une étape supplémentaire est ajoutée. Le segment où le problème se pose sera divisé en deux segments et le point de jonction des nouveaux segments générés sera ajouté à la liste des points de rupture. Cela donne un segment de plus (par conséquent, une autre courbe de Bézier à définir) en remplaçant un segment par deux nouveaux segments. Nous essayons à nouveau de trouver une solution au problème d'optimisation définissant un chemin approximatif dans le couloir de sécurité. L'étape de division est répétée jusqu'à ce que les contraintes seront vérifiées.

4.4 Résultats numériques

4.4.1 résultats de l'approche proposée

- **Squelettisation de l'environnement.** Afin d'illustrer comment le **FMS** extrait par la squelettisation dépend de la taille du drone δ , nous avons considéré la même représentation de l'environnement pour deux différentes valeurs de δ (20cm et 40cm). Comme nous pouvons le voir sur la figure (4.10), le **FMS** se rétrécit lorsque δ augmente. Cela signifie qu'il y a plus de chemins candidats pour atteindre la destination à partir d'une position initiale lorsque la taille du drone est petite.

- **Génération du chemin le plus court.** Nous considérons les deux cas précédents de taille de drone δ (20cm et 40cm). Les figures (4.11) et (4.12) représentent respectivement la représentation 3D et 2D du chemin le plus court trouvé pour chaque cas de δ . Nous pouvons voir que le chemin planifié est sous-optimisé avec des oscillations qui nécessitent une étape de post-traitement pour l'affiner. La figure (4.13) donne une information sur les changements d'altitude tout le long du chemin.

- **Lissage.** Les figures (4.14) et (4.15) montre le chemin affiné après l'utilisation de l'approche basée sur les courbes de Bézier. Nous pouvons voir que le chemin planifié par la méthode proposée est optimal en termes de sécurité et de souplesse.

Le tableau 4.1 indique les résultats de l'écart entre les chemins d'origine (obtenus par la squelettisation et l'application de l'algorithme A^*) et les chemins lissés. Intuitivement, cet écart peut atteindre sa valeur maximale lorsque la largeur du couloir de sécurité est maximale. Nous notons que la méthode de lissage satisfait la condition sur l'écart de lissage maximal toléré.

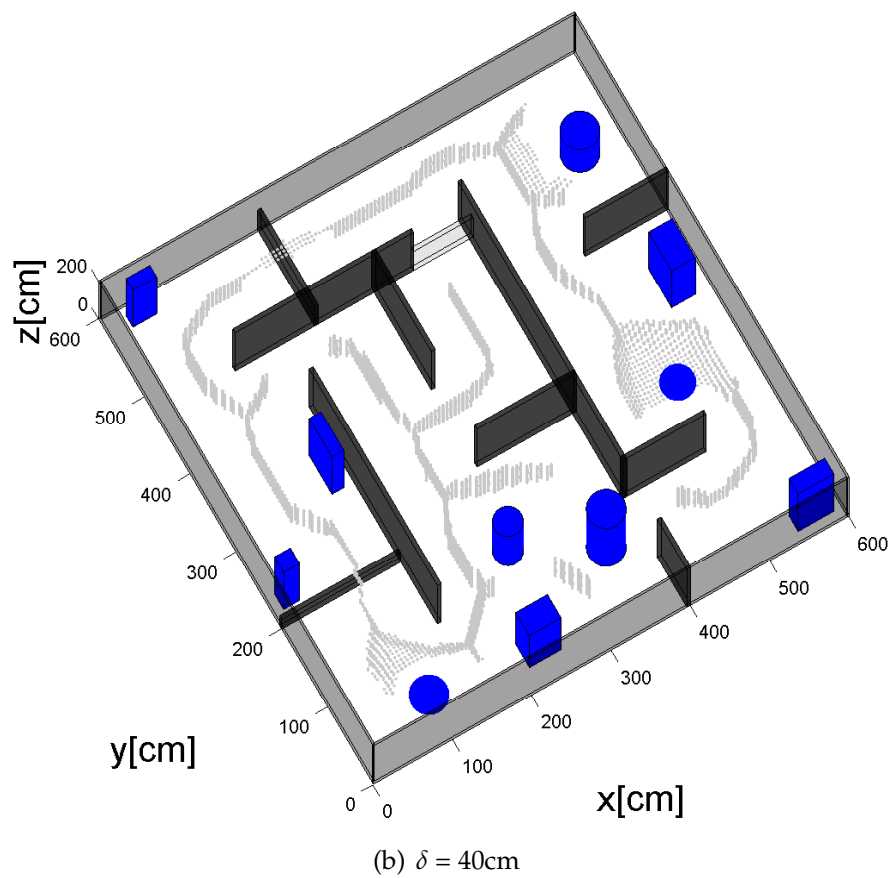
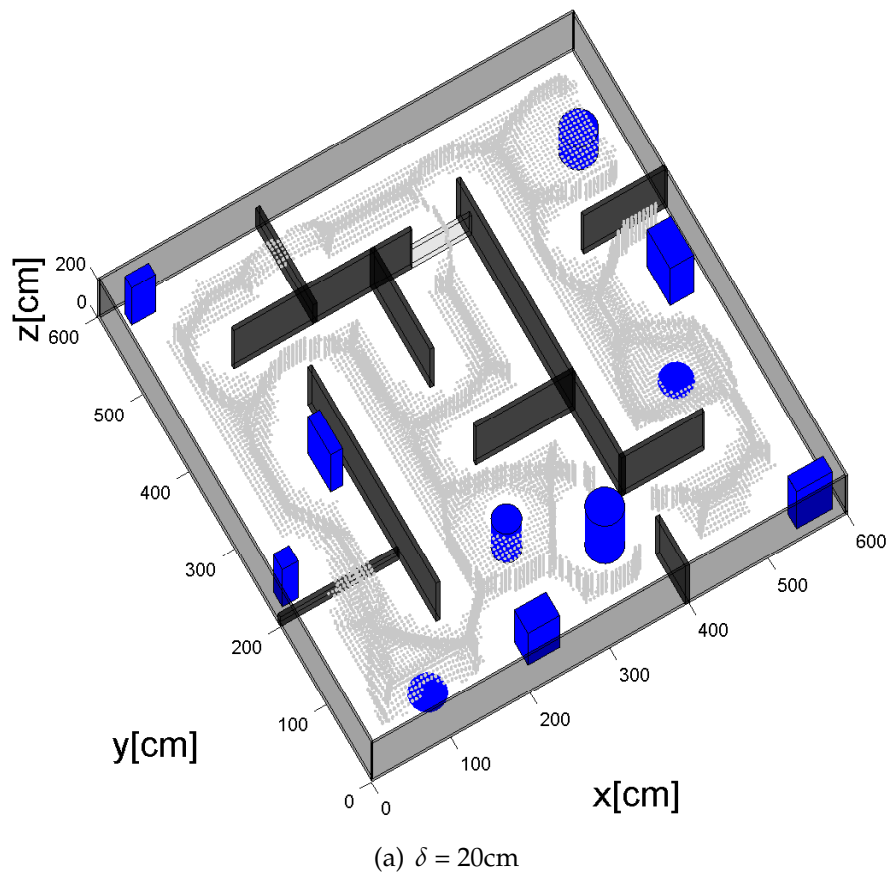


Figure 4.10. Deux cas de squelette de l'environnement obtenus pour différentes tailles de drone.

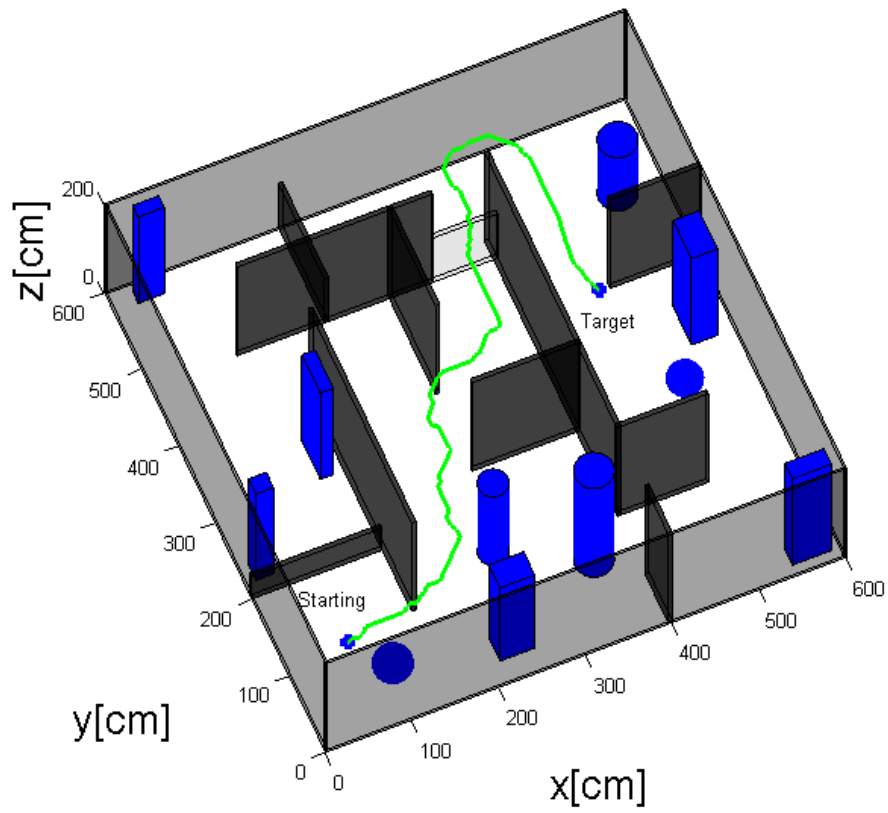
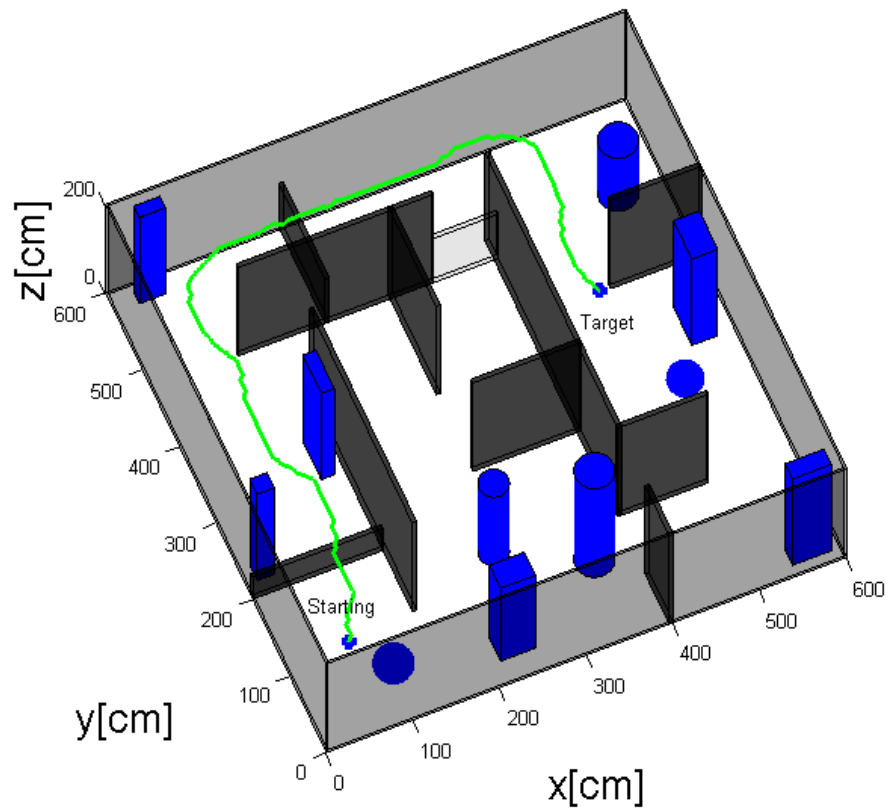
(a) $\delta = 20\text{cm}$ (b) $\delta = 40\text{cm}$

Figure 4.11. La représentation 3D du chemin le plus court calculé sur le FMS pour différentes tailles de drone.

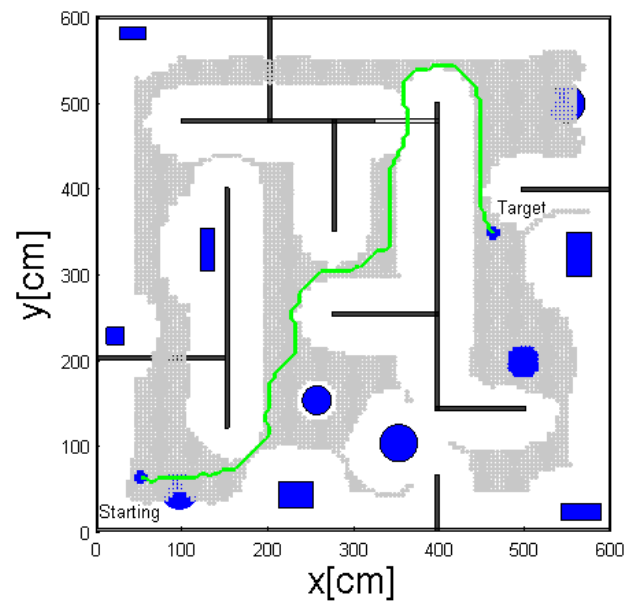
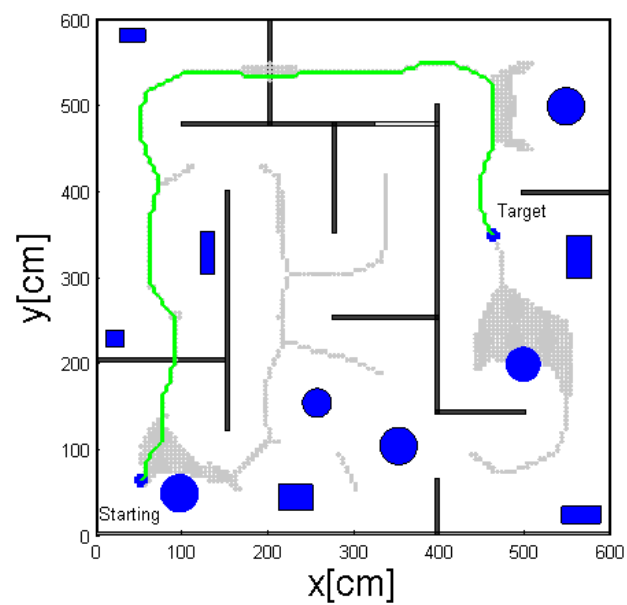
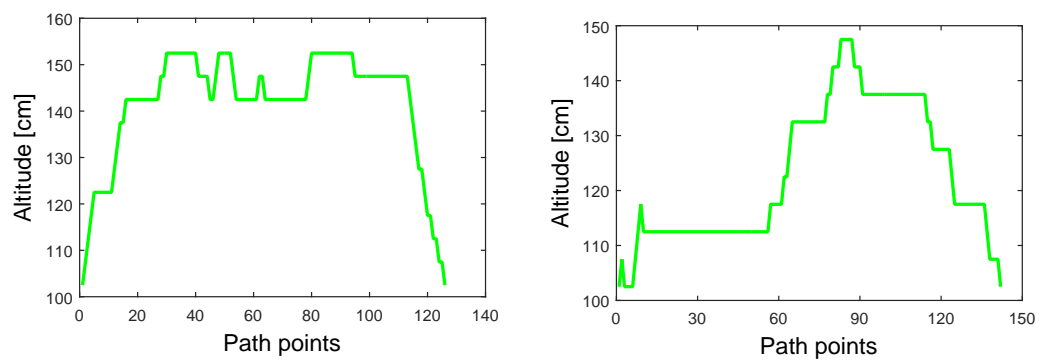
(a) $\delta = 20\text{cm}$ (b) $\delta = 40\text{cm}$

Figure 4.12. La représentation 2D du chemin le plus court calculé sur le FMS pour différentes tailles de drone.

Figure 4.13. Changement d'altitude tout le long du chemin le plus court calculé sur le FMS pour différentes valeurs de δ (20cm (à gauche), 40cm (à droite)).

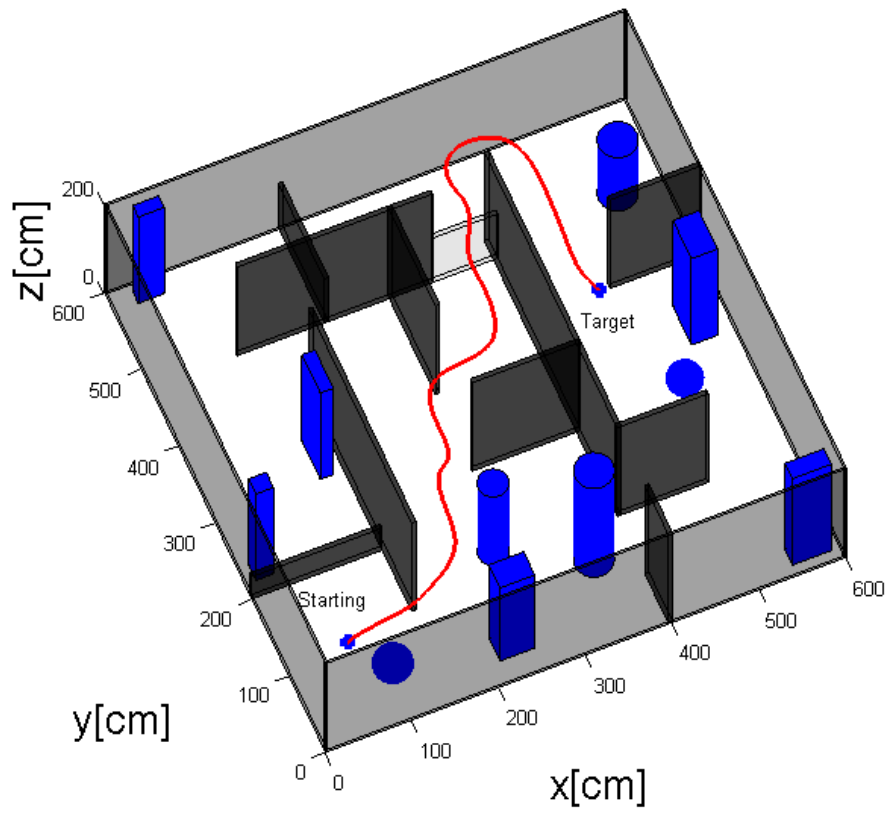
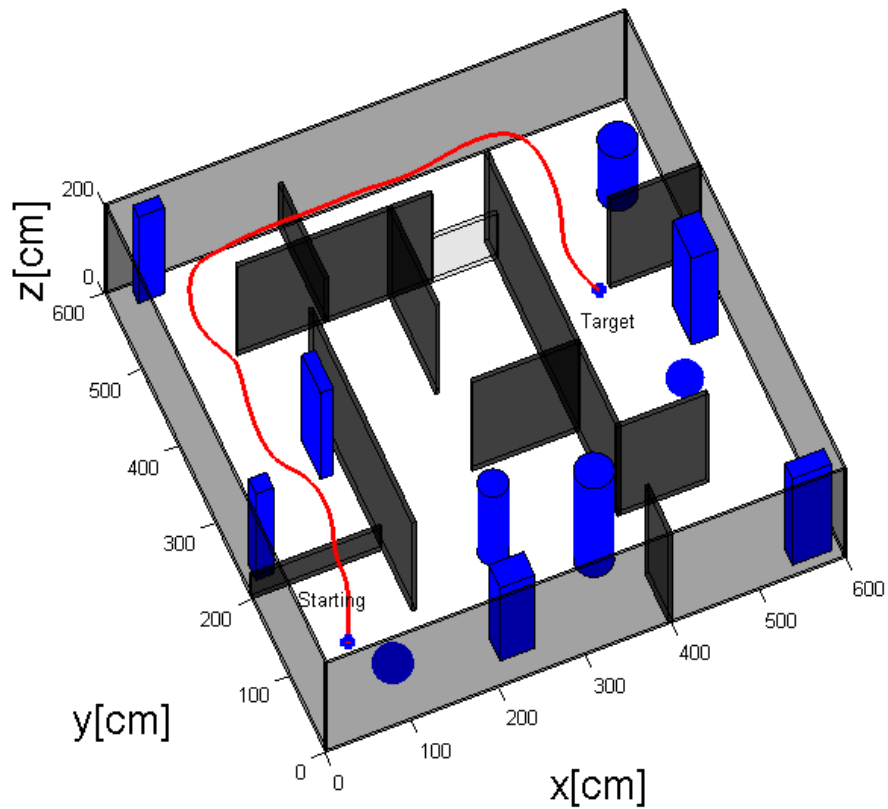
(a) $\delta = 20\text{cm}$ (b) $\delta = 40\text{cm}$

Figure 4.14. Chemin final pour différentes tailles de drone représenté en 3D.

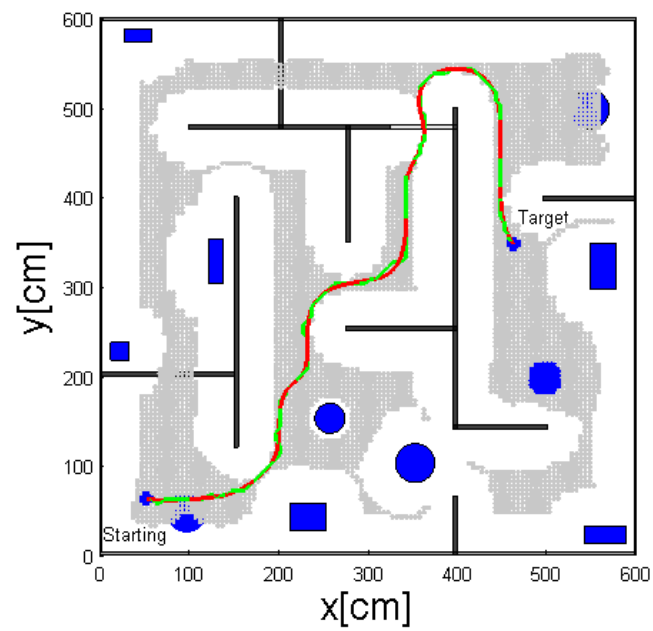
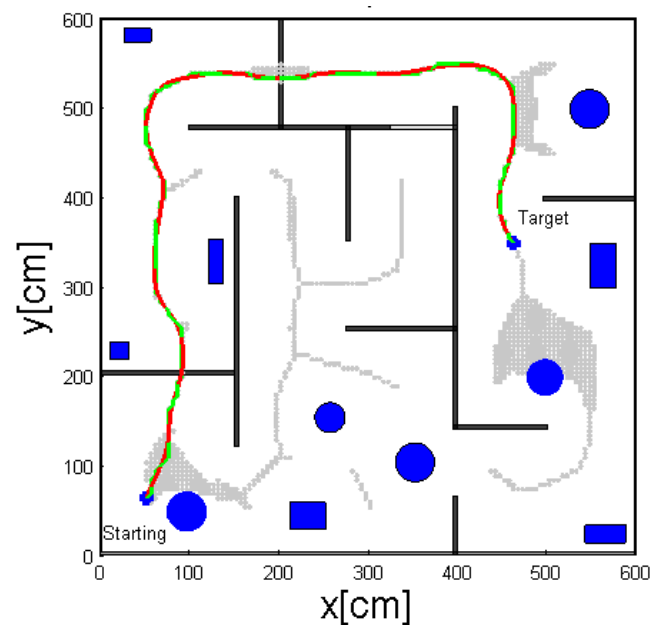
(a) $\delta = 20\text{cm}$ (b) $\delta = 40\text{cm}$

Figure 4.15. La représentation 2D du chemin final pour différentes tailles de drone.

TABLE 4.1
Résultats de lissage.

	cas 1	cas 2
	$\delta = 20\text{cm}$	$\delta = 40\text{cm}$
Erreur de lissage maximale autorisée (cm)	42.59	32.58
Erreur de lissage maximale mesurée (cm)	8.13	8.98
Moyenne de l'erreur de lissage (cm)	2.70	2.55

4.4.2 Étude comparative

Pour fournir une étude comparative de l'approche proposée, nous appliquons l'algorithme A* directement sur la G.O. en 3D pour trouver le chemin le plus court possible. Pour cela, nous fournissons une G.O. gonflée qui prend en compte les différentes tailles de drone considérées précédemment ($\delta = 20cm$ et $\delta = 40cm$). Les résultats graphiques sont représentés dans la figure (4.16) comparés au résultats de l'approche proposée.

Pour analyser efficacement les résultats, nous définissons un ensemble de critères quantitatifs pour l'étude comparative. Les résultats de la comparaison sont donnés dans le tableau 4.2.

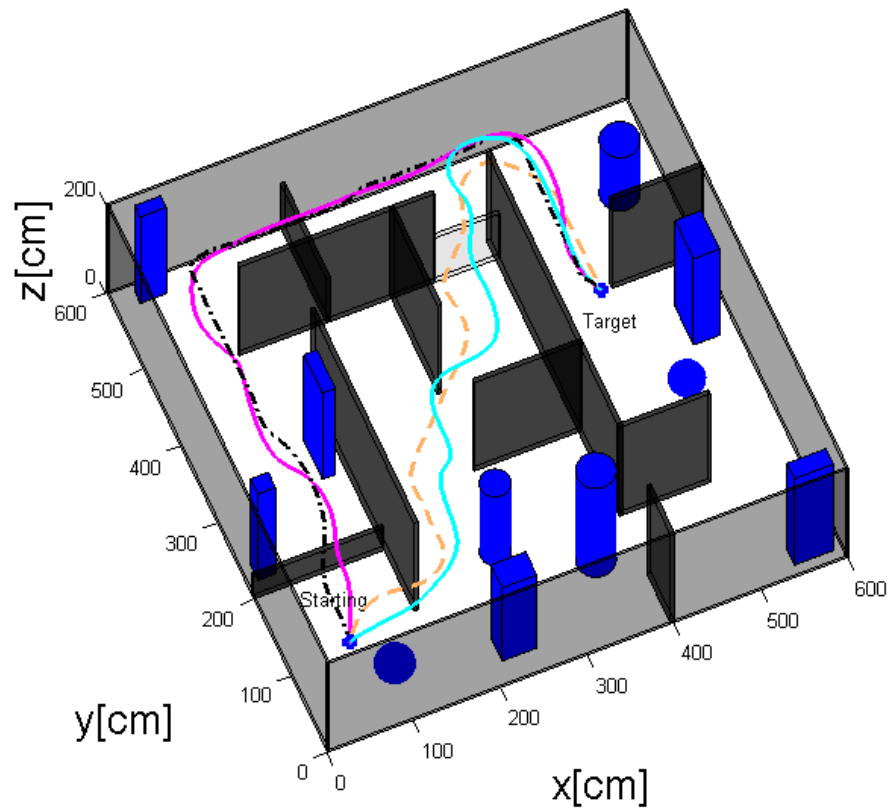
TABLE 4.2

Comparaison quantitative entre le A* appliqué directement sur la G.O. et l'approche proposée en considérant deux différents cas de taille du drone ($\delta = 20cm$ (cas 1), $\delta = 40cm$ (cas 2)).

	A* sur la G.O.		Approche proposée	
	cas 1	cas 2	cas 1	cas 2
Nombre de données d'entrée	236644	61841	23324	7913
Longueur du chemin (m)	9.03	11.08	9.95	11.41
Nombre de points du chemin	155	209	126	142
Minimale dist. aux obstacles (cm)	25.63	42.86	31.13	46.28
Moyenne de la dist. aux obstacles (cm)	40.18	50.51	48.42	56.88

Les données d'entrée (tableau. 4.2) dépendent du type de l'approche appliquée et de la dimension du drone. Lorsque l'algorithme A* est utilisé directement sur la G.O. (sans squelettisation), les données d'entrée sont l'espace navigable dans la grille d'occupation gonflée. En utilisant l'approche proposée, c'est le FMS de l'espace navigable dans la G.O.. Les conclusions tirées du tableau 4.2 sont les suivantes :

- L'approche proposée offre une diminution importante du volume des données d'entrée dans les deux cas (90,15 % et 87,21 % de baisse respectivement) pour une augmentation de respectivement 10,18 % et 2,97 % de la longueur du trajet. La réduction de l'espace de recherche est due à l'étape de squelettisation. Ceci est d'un grand avantage lors de la considération d'applications en temps réel.
- Dans tous les cas, la plus petite distance aux obstacles est supérieure de la taille du drone δ , ce qui confirme que tous les trajets planifiés sont exempts de collision. L'approche proposée donne plus de distance aux obstacles fournissant un meilleur chemin en termes de sécurité que l'algorithme A* appliqué directement sur la G.O.. L'évolution de la distance aux obstacles le long du chemin que le drone est censé suivre est représentée sur la figure (4.17).



XY plot

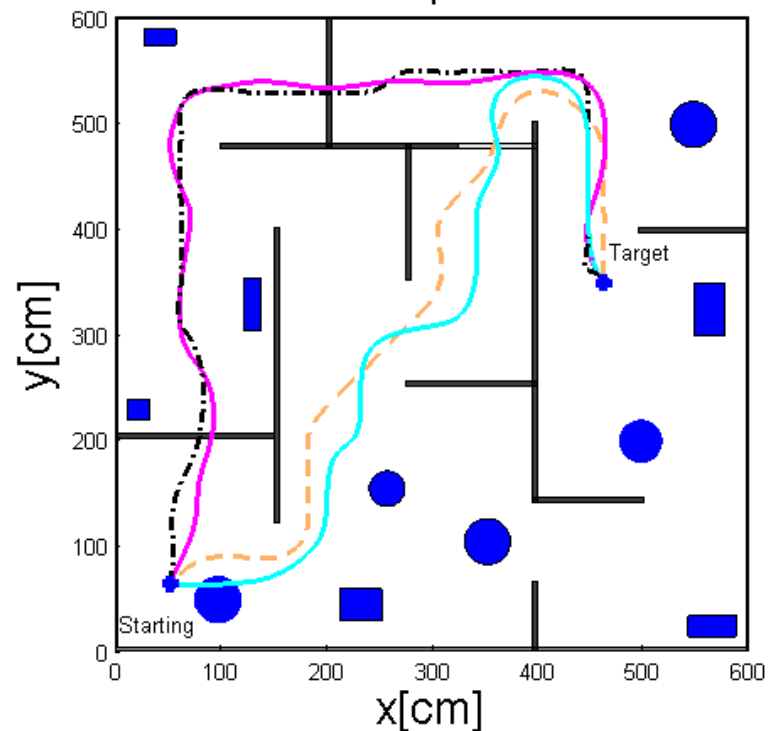
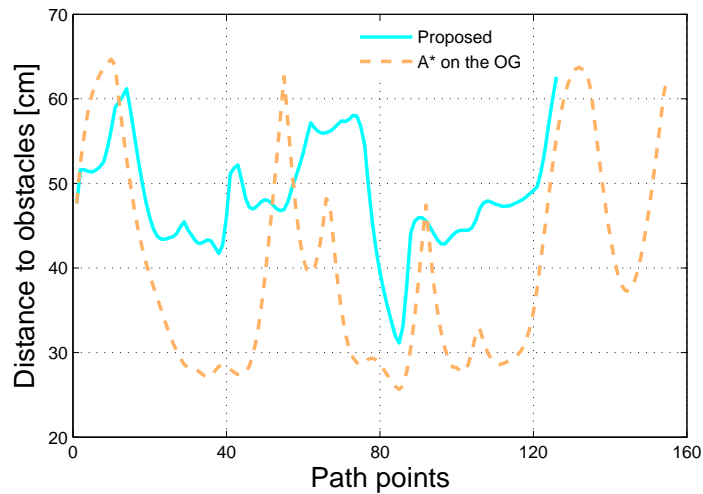
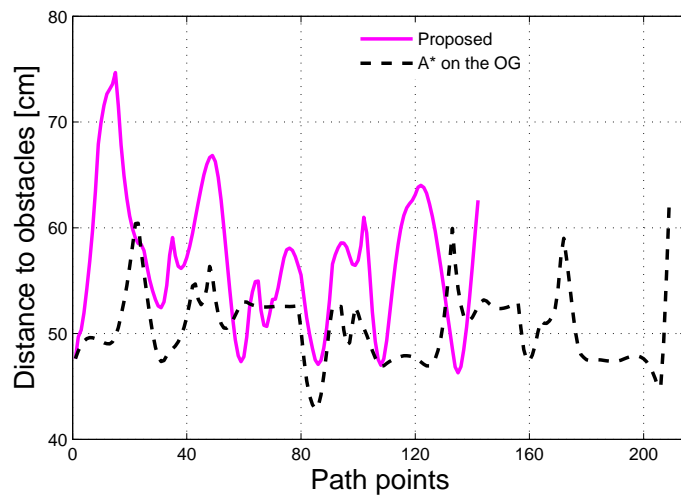


Figure 4.16. Comparaison entre l'approche proposée (ligne continue) pour différentes valeurs de δ ($= 20\text{cm}$ (couleur cyan), $= 40\text{cm}$ (couleur magenta)) et le A* appliqué sur la G.O. gonflée prenant en compte les deux cas : $\delta = 20\text{cm}$ (discontinue ligne couleur orange), $\delta = 40\text{cm}$ (ligne pointillée couleur noire).

Nous percevons que le chemin généré par l'approche proposée n'est pas optimal en ce qui concerne sa longueur. Cependant, il est plus optimal que le chemin planifié en utilisant le A* directement sur la G.O. en ce qui concerne le critère de sécurité.



(a) cas 1



(b) cas 2

Figure 4.17. Évolution de la distance aux obstacles ((a) : $\delta = 20\text{cm}$, (b) : $\delta = 40\text{cm}$).

L'étape de lissage est effectuée pour calculer un chemin approximatif nécessitant le moins de manœuvres du drone sans affecter sa sécurité. Pour cela, deux nouveaux paramètres quantitatifs sont définis afin de mieux évaluer la solution proposée :

- ΔZ_{max} : valeur maximale de changement de l'altitude,
- $\Delta\psi_{max}$: valeur maximale de changement de l'angle de lacet.

Pour les deux tailles de drone considérées précédemment ($\delta = 20\text{cm}$ et $\delta = 40\text{cm}$), les chiffres suivants sont obtenus :

- **Dans le cas 1** ($\delta = 20\text{ cm}$) : $\Delta Z_{max} = 08.78\text{ cm}$, $\Delta\psi_{max} = 28.43\text{ deg}$.
- **Dans le cas 2** ($\delta = 40\text{ cm}$) : $\Delta Z_{max} = 05.41\text{ cm}$, $\Delta\psi_{max} = 23.84\text{ deg}$.

Ces valeurs de paramètres confirment qu'il n'y a pas de changements agressifs d'altitude et d'angle de lacet entre deux points du chemin obtenu.

Nous notons que pour tous les cas simulés, nous considérons le cas particulier où les points de départ et les points cibles sont sur le **FMS**. Ce n'est pas forcément le cas. L'idée à adopter est alors d'atteindre (depuis un point de départ) le point le plus proche sur le **FMS**, et/ou d'abandonner le **FMS** au point le plus proche du point cible. Ces portions supplémentaires au début et/ou à la fin du chemin peuvent être générées en utilisant les courbes de Bézier. Un processus d'optimisation peut être appliqué pour garantir la souplesse du chemin aux points de jonction.

4.5 Conclusion

Dans ce chapitre, nous avons abordé le problème de planification de chemins pour drones. Nous avons considéré le cas des environnements intérieurs avec obstacles statiques et proposé deux stratégies. Dans la première, le chemin calculé est représenté par une courbe de Bézier obtenue par la résolution d'un problème d'optimisation sous contraintes. Dans la deuxième, le chemin le plus court est calculé par l'algorithme A* appliqué sur le squelette 3D de l'environnement. Nous avons utilisé les courbes de Bézier pour le lissage du chemin calculé. Contrairement à la première stratégie, la planification de chemins par squelettisation de l'environnement permet d'obtenir une solution adaptée à la taille du drone en éliminant les passages étroits dans l'espace de navigation. Une étude comparative a démontré l'avantage de l'étape de squelettisation pour réduire l'espace de recherche dans l'environnement.

Conclusion générale et perspectives

Dans ce manuscrit, nous avons abordé le problème de guidage et de contrôle pour un véhicule aérien autonome ayant de multiples rotors. L'objectif de la thèse était de synthétiser des lois de commande simples et efficaces, en s'appuyant sur les résultats des travaux récents, en vue d'une implémentation en temps réel.

D'abord, nous avons fait une étude bibliographique sur les techniques de commande appliquées aux drones à décollage et atterrissage verticaux, en particulier les multi-rotors. La deuxième partie de cette étude de la littérature a porté sur l'analyse des stratégies de planification de chemins dans des environnements 3D pour drones. Ce travail bibliographique est le cœur du premier chapitre. Nous avons conclu la richesse de la littérature sur ce sujet et la diversité des approches proposées. Les recherches effectuées sur les multi-rotors s'intéressaient en grande partie au contrôle de ses véhicules, et ont permis de proposer des solutions efficaces. Quant au guidage, la majorité des solutions proposées ont été vérifiées en simulation et des problèmes liés à la possibilité d'implémentation en temps réel ne sont pas encore résolus.

Dans un deuxième temps, il était nécessaire d'établir un modèle dynamique pour toute étape ultérieure du travail. Nous avons présenté la modélisation dynamique dans le deuxième chapitre dans le but de réaliser des simulations et d'étudier le comportement du véhicule en appliquant des approches de commande. Pour cela, nous avons utilisé l'approche Newtonienne pour obtenir les équations différentielles de la dynamique de translation et de rotation. Notre contribution de cette partie est la proposition d'un modèle dynamique générique pour la commande de cette classe des multi-rotors.

Le modèle dynamique obtenu a été utilisé ensuite dans une structure de commande en cascade en vue de contrôler la dynamique du multi-rotor à six degrés de liberté. L'approche proposée est une hybridation de deux méthodes de commande appliquées séparément sur la dynamique de rotation et celle de translation. Le contrôleur PID classique a été adopté pour la stabilisation de l'orientation, et le Back-stepping à action intégrale pour le suivi de trajectoires. Le comportement du multi-rotor a été également simulé dans la situation de présence de panne totale sur un actionneur pour des systèmes à différents nombres d'actionneurs. La synthèse des lois de commande et les résultats obtenus ont été introduits dans le troisième chapitre.

Le dernier chapitre présente une partie importante de la contribution de ce travail de thèse. Elle concerne la planification de chemins 3D et l'évitement d'obstacles. Nous avons proposé deux stratégies pour ce problème dans la continuité des travaux de recherche de l'équipe Mécatronique de l'université de Limoges. Il s'agit de l'extension en 3D d'une approche de planification de chemins initialement proposée pour les robots terrestres, et la proposition d'une approche originale pour robots aériens. La particularité de l'approche proposée réside dans l'utilisation du squelette tenant compte de la dimension du robot. Cela a permis d'un côté de limiter l'espace de recherche et de gagner donc en temps de calcul, et d'autre côté d'obtenir des solutions efficaces adaptées à la taille du robot considéré.

Pour la suite du travail, nous pouvons lister plusieurs perspectives :

- Dans ce travail, nous avons abordé le problème de guidage en considérant un environnement statique. Dans ce cas là, le traitement a été fait hors ligne et un chemin prédéfini a été obtenu sans considérer la notion du temps. Une amélioration peut être la considération du cas dynamique des obstacles et la planification d'une trajectoire faisable en termes de contraintes dynamique du robot aérien.
- Il s'avère que les méthodes de commande et de planification de chemins proposées dans ce travail sont performantes au niveau théorique en simulation. Une perspective envisageable serait de les implémenter à bord d'un drone pour réaliser des tests expérimentaux réels. Un système de Guidage-Navigation-Contrôle sera donc nécessaire y compris la composante de navigation qui n'est pas abordée dans ce travail. Le système de navigation permettrait la collecte des informations nécessaires à la planification de trajectoires et la conception des lois de commande du déplacement du véhicule.
- Dans le même contexte, il serait également intéressant de concevoir des commandes tolérantes aux défauts. La panne totale ou partielle d'un ou de plusieurs actionneurs lors d'un vol réel peut conduire à des situations très critiques. Programmer le robot aérien de sorte que même si un actionneur tombe en panne, les autres actionneurs compensent pour terminer la mission avec succès ou effectuer un atterrissage urgent sans écrasement.
- Une extension envisageable serait d'étendre nos travaux pour application dans un contexte multi-robots. L'idée de base est que les systèmes multi-robots peuvent effectuer des tâches plus efficacement qu'un seul robot ou peuvent accomplir des tâches non exécutables par un seul. Il peut s'agir d'un groupe de drones ou d'une équipe mixte de robots aériens et robots terrestres.
- Une amélioration pourrait être apportée au modèle mathématique du véhicule pour le rendre plus fidèle au comportement réel en prenant en considération la dynamique des actionneurs, des fonctions de l'aéronautique et des fonctions de perturbations externes.

Bibliographie

- [1] OFFICE OF THE SECRETARY OF DEFENSE, « Unmanned aircraft systems roadmap 2005-2030 », rap. tech., Département de la défense des États-Unis, 2005.
- [2] ARMY UAS COE STAFF, « "eyes of the army" u. s. army unmanned aircraft systems roadmap 2010-2035 », rap. tech., U. S. ARMY, 2010.
- [3] D. K. PHUNG, *Conception, Modeling, and Control of a Convertible Mini-Drone*. Thèse doctorat, Université Pierre et Marie Curie, 2015.
- [4] G. FAY, « Derivation of the aerodynamic forces for the Mesicopter simulation », rap. tech., Université de Stanford, 2001.
- [5] S. BOUABDALLAH, *Design and control of quadrotors with application to autonomous flying*. Thèse doctorat, École Polytechnique Fédérale Lausanne, 2007.
- [6] P. POUNDS, R. MAHONY et P. CORKE, « Modelling and control of a large quadrotor robot control », *Engineering Practice*, vol. 18, no. 7, p. 691—699, 2010.
- [7] P. E. I. POUNDS, D. R. BERSAK et A. M. DOLLAR, « Stability of small-scale UAV helicopters and quadrotors with added payload mass under PID control », *Autonomous Robots*, vol. 33, no. 1-2, p. 129—142, 2012.
- [8] G. M. HOFFMANN, H. HUANG, S. L. WASLANDER et C. J. TOMLIN, « Quadrotor helicopter flight dynamics and control : Theory and experiment », in *Proceedings of the AIAA Guidance, Navigation, and Control Conference*, p. 1–20, Août 2007.
- [9] G. M. HOFFMANN, S. L. WASLANDER et C. J. TOMLIN, « Quadrotor helicopter trajectory tracking control », in *Proceedings of the AIAA Guidance, Navigation, and Control Conference*, p. 1–14, Août 2008.
- [10] H. HUANG, G. HOFFMANN, S. L. WASLANDER et C. TOMLIN, « Aerodynamics and control of autonomous quadrotor helicopters in aggressive maneuvering », in *Proceedings of the IEEE International Conference on Robotics and Automation (ICRA)*, p. 3277–3282, Mai 2009.
- [11] G. M. HOFFMANN, D. G. RAJNARAYAN, S. L. WASLANDER, D. DOSTAL, J. S. JANG et C. J. TOMLIN, « The stanford testbed of autonomous rotorcraft for multi agent

- control (STARMAC) », in *Proceedings of the 23rd Digital Avionics System Conference*, p. 12.E.4/1–10, Novembre 2004.
- [12] S. L. WASLANDER, G. HOFFMANN, J. S. JANG et C. J. TOMLIN, « Multi-agent quadrotor testbed design : Integral sliding mode vs. reinforcement learning », in *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, p. 468–473, Août 2005.
- [13] S. LUPASHIN, M. HEHN, M. W. MUELLER, A. P. SCHOELLIG, M. SHERBACK et R. D'ANDREA, « A platform for aerial robotics research and demonstration : The Flying Machine Arena », *Mechatronics*, vol. 24, p. 41–54, Février 2014.
- [14] M. KAMEL, T. STASTNY, K. ALEXIS et R. SIEGWART, « Model predictive control for trajectory tracking of unmanned aerial vehicles using robot operating system », in *Robot Operating System (ROS)* (A. KOUBAA, éd.), vol. 707, in *Studies in Computational Intelligence*, p. 3–39, Springer, Cham, 2017.
- [15] S. DRIESSENS et P. E. I. POUNDS, « Towards a more efficient quadrotor configuration », in *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, p. 1386–1392, Novembre 2013.
- [16] K. J. ÅSTRÖM et T. HÄGGLUND, *PID Controllers - Theory, Design, and Tuning*. International Society for Measurement and Control, 1995.
- [17] E. ROUSSEL, *Contribution à la modélisation, l'identification et la commande d'un hélicoptère miniature*. Thèse doctorat, Université de Strasbourg, 2017.
- [18] T. HAMEL, R. MAHONY, R. LOZANO et J. OSTROWSKI, « Dynamic modelling and configuration stabilization for an X4-Flyer », in *15th Triennial IFAC World Congress*, 2002.
- [19] N. GUENARD, T. HAMEL et V. MOREAU, « Dynamic modeling and intuitive control strategy for an "X4-Flyer" », in *the IEEE International Conference on Robotics and Automation (ICRA)*, p. 141–146, Juin 2005.
- [20] J. M. PFLIMLIN, T. HAMEL, P. SOUERÈS et R. MAHONY, « A hierarchical control strategy for the autonomous navigation of a ducted fan flying robot », in *International Conference on Robotics and Automation*, p. 2491–2496, IEEE, Mai 2006.
- [21] A. HABLY et N. MARCHAND, « Global stabilization of a four rotor helicopter with bounded inputs », in *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, p. 129—134.
- [22] L. MEIER, D. HONEGGER et M. POLLEFEYS, « PX4 : A node-based multithreaded open source robotics framework for deeply embedded platforms », in *2015 IEEE International Conference on Robotics and Automation (ICRA)*, p. 6235—6240, IEEE, Mai 2015.

- [23] C. ANDERSON et DIY DRONES TEAM, « Ardupilot open-source project ». <http://ardupilot.org/>, 2018. Consulté le 06 Mai 2018.
- [24] G. SZAFRANSKI et R. CZYBA, « Different approaches of PID control UAV type quadrotor », in *International Micro Air Vehicle conference and competitions 2011 (IMAV 2011)*, Septembre 2011.
- [25] J. P. HOW, B. BEHREKE, A. FRANK, D. DALE et J. VIANA, « Real-time indoor autonomous vehicle test environment », *IEEE Control Systems Magazine*, vol. 28, no. 2, p. 52–64, 2008.
- [26] K. NONAMI, F. KENDOUL, S. S. W. WANG et D. NAKAZAWA, *Autonomous Flying Robots : Unmanned Aerial Vehicles and Micro Aerial Vehicles*. Tokyo : Springer-Verlag, 2010.
- [27] V. G. ADÎR et A. M. STOICA, « Integral LQR control of a star-shaped octorotor », *INCAS bulletin*, vol. 4, no. 2, p. 3–18, 2012.
- [28] S. BOUABDALLAH, A. NOTH et R. SIEGWART, « PID vs LQ control techniques applied to an indoor micro quadrotor », in *2004 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, p. 2451–2456, 2004.
- [29] H. KWAKERNAAK, « H_2 -optimization : Theory and applications to robust control design », *Annual Reviews in Control*, vol. 26, no. 1, p. 45–56, 2002.
- [30] O. ARAAR et N. AOUF, « Full linear control of a quadrotor UAV, LQ vs H_∞ », in *2014 UKACC International Conference on Control*, p. 133–138, Juillet 2014.
- [31] J. P. ORTIZ, L. I. MINCHALA et M. J. REINOSO, « Nonlinear robust H-Infinity PID controller for the multivariable system quadrotor », *IEEE Latin America Transactions*, vol. 14, no. 3, p. 1176–1183, 2016.
- [32] R. A. GARCÍA, F. RUBIO et M. ORTEGA, « Robust PID control of the quadrotor helicopter », vol. 45, no. 3, p. 229–234, 2012.
- [33] A. MOKHTARI, A. BENALLEGUE et B. DAACHI, « Robust feedback linearization and GH_∞ controller for a quadrotor unmanned aerial vehicle », *Electrical Engineering*, vol. 57, no. 1, p. 20–27, 2006.
- [34] D. MELLINGER et V. KUMAR, « Minimum snap trajectory generation and control for quadrotors », in *2011 IEEE International Conference on Robotics and Automation*, p. 2520–2525, IEEE, Mai 2011.
- [35] F. KENDOUL, Z. YU et K. NONAMI, « Guidance and nonlinear control system for autonomous flight of mini rotorcraft unmanned aerial vehicles », *Journal of Field Robotics*, vol. 27, no. 3, p. 311–334.

- [36] A. R. TEEL, « Global stabilization and restricted tracking for multiple integrators with bounded controls », *Systems and Control Letters*, vol. 18, p. 165—171, 1992.
- [37] P. CASTILLO, A. DZUL et R. LOZANO, « Real-time stabilization and tracking of a four-rotor mini rotorcraft », *IEEE Transactions on Control Systems Technology*, vol. 12, no. 4, p. 510–516, 2004.
- [38] F. KENDOUL, D. LARA, I. FANTONI et R. LOZANO, « Real-time nonlinear embedded control for an autonomous quad-rotor helicopter », *AIAA Journal of Guidance, Control, and Dynamics*, vol. 30, no. 4, p. 1049—1061, 2007.
- [39] P. V. KOKOTOVIC, « The joy of feedback : nonlinear and adaptive », *IEEE Control Systems*, vol. 12, no. 3, p. 7–17, 1992.
- [40] P. V. KOKOTOVIC, M. KRSTIC et I. KANELLAKOPOULOS, « Backstepping to passivity : recursive design of adaptive systems », in *the 31st IEEE Conference on Decision and Control*, IEEE, Décembre 1992.
- [41] R. SKJETNE et T. I. FOSSEN, « On integral control in backstepping : Analysis of different techniques », in *American Control Conference*, p. 1899–1904, 2004.
- [42] R. MAHONY et T. HAMEL, « Robust trajectory tracking for a scale model autonomous helicopter », *International journal of robust and nonlinear control*, vol. 14, no. 12, p. 1035–1059, 2004.
- [43] B. ZHU et W. HUO, « Adaptive backstepping control for a miniature autonomous helicopter », in *50th IEEE Conference on Decision and Control and European Control Conference*, p. 5413–5418, IEEE, 2011.
- [44] L. MEDERREG, F. DIAZ et K. N. M'SIRDI, « Nonlinear backstepping control with observer design for a 4 rotors helicopter », in *International Conference on Advances in Vehicle Control and Safety*, Octobre 2004.
- [45] N. GUENARD, T. HAMEL et R. MAHONY, « A practical visual servo control for an unmanned aerial vehicle », *IEEE Transactions on Robotics*, vol. 24, no. 2, p. 331—340, 2008.
- [46] M. BOUCHOUCHA, S. SEGHOURE, H. OSMANI et M. BOURI, « Integral backstepping for attitude tracking of a quadrotor system », *Elektronika ir Elektrotechnika*, vol. 116, no. 10, p. 75–80, 2011.
- [47] T. MADANI et A. BENALLEGUE, « Control of a quadrotor mini-helicopter via full state backstepping technique », in *45th IEEE Conference on Decision and Control*, p. 1515–1520, IEEE, Décembre 2006.

- [48] H. BOUADI et M. TADJINE, « Nonlinear observer design and sliding mode control of four rotors helicopter », *International Journal of Aerospace and Mechanical Engineering*, vol. 1, no. 7, 2007.
- [49] F. SHARIFI, M. MIRZAEI, B. W. GORDON et Y. ZHANG, « Fault tolerant control of a quadrotor UAV using sliding mode control », in *Conference on Control and Fault-Tolerant Systems*, p. 239–244, 2010.
- [50] V. G. ADÎR, A. M. STOICA et J. F. WHIDBORNE, « Sliding mode control of a 4Y octo-rotor », *UPB Scientific Bulletin, Series D : Mechanical Engineering*, vol. 74, no. 4, p. 37–52, 2012.
- [51] L. R. GARCÍA CARRILLO, A. DZUL et R. LOZANO, « Hovering quad-rotor control : A comparison of nonlinear controllers using visual feedback », *IEEE Transactions on Aerospace and Electronic Systems*, vol. 48, p. 3159–3170, Octobre 2012.
- [52] S. BOUABDALLAH et R. SIEGWART, « Backstepping and sliding-mode techniques applied to an indoor micro quadrotor », in *the 2005 IEEE International Conference on Robotics and Automation*, p. 2259–2264, IEEE, April 2005.
- [53] H. KIM, D. SHIM et S. SASTRY, « Nonlinear model predictive tracking control for rotorcraft-based unmanned aerial vehicles », in *Proceedings of the 2002 American Control Conference*, p. 3576—3581, IEEE, Mai 2002.
- [54] J. QI, D. SONG, L. DAI, J. HAN et Y. WANG, « The new evolution for SIA rotorcraft UAV project », *Journal of Robotics*, vol. 2010, p. 1–9, 2010.
- [55] E. SHAW, H. CHUNG, J. K. HEDRICK et S. SASTRY, « Unmanned helicopter formation flight experiment for the study of mesh stability », in *Cooperative Systems* (D. GRUNDEL, R. MURPHEY, P. PARDALOS et O. PROKOPYEV, éds), vol. 588, in *Lecture Notes in Economics and Mathematical Systems*, p. 37–56, Springer, Berlin, Heidelberg, 2007.
- [56] J. A. J. LIGTHART, P. POKSAWAT, L. WANG et H. NIJMEIJER, « Experimentally validated model predictive controller for a hexacopter », *IFAC PapersOnLine*, vol. 50, p. 4076—4081, Juillet 2017.
- [57] A. PRACH et E. KAYACAN, « An MPC-based position controller for a tilt-rotor tricopter VTOL UAV », *Optimal Control Applications and Methods*, vol. 39, no. 1, p. 343–356, 2018.
- [58] M. ABDOLHOSSEINI, Y. M. ZHANG et C. A. RABBATH, « An efficient model predictive control scheme for an unmanned quadrotor helicopter », *Journal of Intelligent and Robotic Systems*, vol. 70, p. 27–38, Avril 2013.
- [59] P. W. GIBBENS et E. D. B. MEDAGODA, « Efficient model predictive control algorithm for aircraft », *Journal of Guidance, Control, and Dynamics*, vol. 34, no. 7, p. 1909—1915, 2011.

- [60] I. SADEGHZADEH, M. ABDOLHOSSEINI et Y. ZHANG, « Payload drop application using an unmanned quadrotor helicopter based on gain-scheduled PID and model predictive control », *Unmanned Systems*, vol. 02, no. 01, p. 39–52, 2014.
- [61] G. V. RAFFO, M. G. ORTEGA et F. R. RUBIO, « MPC with nonlinear H_∞ control for path tracking of a quad-rotor helicopter », *IFAC Proceedings Volumes*, vol. 41, no. 2, p. 8564–8569, 2008.
- [62] P. BOUFFARD, A. ASWANI et C. TOMLIN, « Learning-based model predictive control on a quadrotor : Onboard implementation and experimental results », in *2012 IEEE International Conference on Robotics and Automation (ICRA)*, p. 279—284, IEEE, Mai 2012.
- [63] K. ALEXIS, C. PAPACHRISTOS, G. NIKOLAKOPOULOS et A. TZES, « Model predictive quadrotor indoor position control », in *2011 19th Mediterranean Conference on Control and Automation (MED)*, p. 1247—1252, IEEE, Juin 2011.
- [64] K. ALEXIS, G. NIKOLAKOPOULOS et A. TZES, « On trajectory tracking model predictive control of an unmanned quadrotor helicopter subject to aerodynamic disturbances », *Asian Journal of Control*, vol. 16, no. 1, p. 209–224, 2014.
- [65] M. NEUNERT, C. de CROUSAZ, F. FURRER, M. KAMEL, F. FARSHIDIAN, R. SIEGWART et J. BUCHLI, « Fast nonlinear model predictive control for unified trajectory optimization and tracking », in *2016 IEEE International Conference on Robotics and Automation (ICRA)*, p. 1398—1404, IEEE, Mai 2016.
- [66] M. KAMEL, J. ALONSO-MORA, R. SIEGWART et J. I. NIETO, « Nonlinear model predictive control for multi-micro aerial vehicle robust collision avoidance », *arXiv preprint*, 2017. arXiv :1703.01164.
- [67] C. COZA et C. MACNAB, « A new robust adaptive-fuzzy control method applied to quadrotor helicopter stabilization », in *Annual Conference of the North American Fuzzy Information Processing Society (NAFIPS)*, p. 475–479, IEEE, Juin 2006.
- [68] M. SANTOS, V. LÓPEZ et F. MORATA, « Intelligent fuzzy controller of a quadrotor », in *2010 International Conference on Intelligent Systems and Knowledge Engineering (ISKE)*, p. 141–146, IEEE, Novembre 2010.
- [69] C. NICOL, C. MACNAB et A. RAMIREZ-SERRANO, « Robust neural network control of a quadrotor helicopter », in *2008 Canadian Conference on Electrical and Computer Engineering (CCECE)*, p. 1233–1237, IEEE, Mai 2008.
- [70] T. DIERKS et S. JAGANNATHAN, « Output feedback control of a quadrotor uav using neural networks », *IEEE Transactions on Neural Networks*, vol. 21, no. 1, p. 50–66, 2010.

- [71] M. COLLOTTA, G. PAU et R. CAPONETTO, « A real-time system based on a neural network model to control hexacopter trajectories », in *2014 International Symposium on Power Electronics, Electrical Drives, Automation and Motion (SPEEDAM)*, p. 225–230, IEEE, Juin 2014.
- [72] P. BHATKHANDI et T. C. HAVENS, « Real time fuzzy controller for quadrotor stability control », in *2014 IEEE International Conference on Fuzzy Systems (FUZZ-IEEE)*, p. 913–919, IEEE, Juillet 2014.
- [73] X. CHEN, D. LI, Z. XU et Y. BAI, « Robust control of quadrotor MAV using self-organizing interval type-II fuzzy neural networks (SOIT-IIFNNs) controller », *International Journal of Intelligent Computing and Cybernetics*, vol. 4, no. 3, p. 397–412, 2011.
- [74] O. BOUHALI et H. BOUDJEDIR, « Neural network control with neuro-sliding mode observer applied to quadrotor helicopter », in *2011 International Symposium on Innovations in Intelligent Systems and Applications (INISTA)*, p. 24–28, IEEE, Juin 2011.
- [75] J. F. SHEPHERD et K. TUMER, « Robust neuro-control for a micro quadrotor », in *12th Annual Conference on Genetic and Evolutionary Computation (GECCO' 10)*, p. 1131–1138, ACM, Juillet 2010.
- [76] F. KENDOUL, « Survey of advances in guidance, navigation, and control of unmanned rotorcraft systems », *Journal of Field Robotics*, vol. 29, no. 2, p. 315–378, 2012.
- [77] L. YANG, J. QI, D. SONG, J. XIAO, J. HAN et Y. XIA, « Survey of robot 3d path planning algorithms », *Journal of Control Science and Engineering*, vol. 2016, 2016. 22 pages.
- [78] P. E. HART, N. J. NILSSON et B. RAPHAEL, « A formal basis for the heuristic determination of minimum cost paths », *IEEE Transactions on Systems, Science, and Cybernetics*, vol. SSC-4, no. 2, p. 100–107, 1968.
- [79] E. W. DIJKSTRA, « A note on two problems in connexion with graphs », *Numerische Mathematik*, vol. 1, no. 1, p. 269–271, 1959.
- [80] S. KOENIG et M. LIKHACHEV, « D* Lite », in *AAAI Conference on Artificial Intelligence*, p. 476–483, Juillet-Août 2002.
- [81] A. NASH, S. KOENIG et M. LIKHACHEV, « Incremental Phi* : Incremental any-angle path planning on grids », in *International Joint Conference on Artificial Intelligence (IJCAI)*, p. 1824–1830, Juillet 2009.
- [82] A. K. GURUJI, H. AGARWAL et D. K. PARSEDIYA, « Time-efficient A* algorithm for robot path planning », *Procedia Technology*, vol. 23, no. 2016, p. 144–149, 2016.
- [83] L. DE FILIPPIS, G. GUGLIERI et F. QUAGLIOTTI, « Path planning strategies for UAVS in 3D environments », *Journal of Intelligent and Robotic Systems*, vol. 65, no. 1–4, p. 247–264, 2012.

- [84] J. CARSTEN, D. FERGUSON et A. STENTZ, « 3D field D* : Improved path planning and replanning in three dimensions », in *IEEE/RSJ International Conference on Intelligent Robots and Systems*, p. 3381–3386, Octobre 2006.
- [85] Z. QI, Z. SHAO, Y. S. PING, L. M. HIOT et Y. K. LEONG, « An improved heuristic algorithm for UAV path planning in 3D environment », in *International Conference on Intelligent Human-Machine Systems and Cybernetics*, p. 258–261, IEEE, Août 2010.
- [86] A. NASH, S. KOENIG et C. TOVEY, « Lazy Theta* : Any-angle path planning and path length analysis in 3D », in *AAAI Conference on Artificial Intelligence*, p. 147–154, Juillet 2010.
- [87] S. M. LAVALLE, « Rapidly-exploring random trees : a new tool for path planning », rap. tech., Computer Science Department, Iowa State University, Octobre 1998.
- [88] S. M. LAVALLE et J. J. KUFFNER, « Randomized kinodynamic planning », in *IEEE International Conference on Robotics and Automation*, p. 473–479, Mai 1999.
- [89] L. E. KAVRAKI, P. SVESTKA, J.-C. LATOMBE et M. H. OVERMARS, « Probabilistic roadmaps for path planning in high-dimensional configuration spaces », *IEEE Transactions on Robotics and Automation*, vol. 12, no. 4, p. 566–580, 1996.
- [90] S. KARAMAN et E. FRAZZOLI, « Sampling-based algorithms for optimal motion planning », *The International Journal of Robotics Research*, vol. 30, no. 7, p. 846–894, 2011.
- [91] R. ROHLIN et L. E. KAVRAKI, « Path planning using lazy PRM », in *International Conference on Robotics and Automation (ICRA)*, p. 521–528, IEEE, Avril 2000.
- [92] E. FRAZZOLI, M. A. DAHLEH et E. FERON, « Real-time motion planning for agile autonomous vehicles », *Journal of Guidance, Control, and Dynamics*, vol. 25, no. 1, p. 116–129, 2002.
- [93] K. E. BEKRIS, B. Y. CHEN, A. M. LADD, E. PLAKU et L. E. KAVRAKI, « Multiple query probabilistic roadmap planning using single query planning primitives », in *IEEE/RJS International Conference on Intelligent Robots and Systems*, p. 656–661, Octobre 2003.
- [94] D. AARNO, D. KRAGIC et H. I. CHRISTENSEN, « Artificial potential biased probabilistic roadmap method », in *International Conference on Robotics and Automation*, p. 461–466, IEEE, Avril 2004.
- [95] P. O. PETTERSSON et P. DOHERTY, « Probabilistic roadmap based path planning for an autonomous unmanned helicopter », *Journal of Intelligent and Fuzzy Systems*, vol. 17, no. 4, p. 395–405, 2006.

- [96] F. YAN, Y.-S. LIU et J.-Z. XIAO, « Path planning in complex 3D environments using a probabilistic roadmap method », *International Journal of Automation and Computing*, vol. 10, no. 6, p. 525–533, 2013.
- [97] M. KOTHARI et I. POSTLETHWAITE, « A probabilistically robust path planning algorithm for UAVs using rapidly-exploring random trees », *Journal of Intelligent and Robotic Systems*, vol. 71, no. 2, p. 231–253, 2013.
- [98] K. YANG et S. SUKKARIEH, « 3D smooth path planning for a UAV in cluttered natural environments », in *2008 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, p. 794–800, Septembre 2008.
- [99] X. WU, C. GUO, Y. LI, W. CHEN et Y. WANG, « The research of RRT route planning algorithm for UAV that based on kinematic equation », *International Journal of Control and Automation*, vol. 8, no. 1, p. 287–296, 2015.
- [100] B. MILLER, K. STEPANYAN, A. MILLER et M. ANDREEV, « 3D path planning in a threat environment », in *Conference on Decision and Control and European Control Conference*, p. 6864–6869, IEEE, Decembre 2011.
- [101] A. CHAMSEDDINE, Y. ZHANG, C. A. RABBATH, C. JOIN et D. THEILLIOL, « Flatness-based trajectory planning/replanning for a quadrotor unmanned aerial vehicle », *IEEE Transactions On Aerospace And Electronic Systems*, vol. 48, no. 4, p. 2832–2848, 2012.
- [102] E. MASEHIAN et G. HABIBI, « Robot path planning in 3D space using binary integer programming », *International Journal of Computer and Information Engineering*, vol. 1, no. 5, p. 1255–1260, 2007.
- [103] H. DUAN et P. LI, *Bio-inspired Computation in Unmanned Aerial Vehicles*. Springer-Verlag Berlin Heidelberg, 1 éd., 2014.
- [104] J. H. HOLLAND, « Adaptation in natural and artificial systems : an introductory analysis with applications to biology, control, and artificial intelligence », rap. tech., University of Michigan Press, 1975.
- [105] I. K. NIKOLOS, K. P. VALAVANIS, N. C. TSOURVELOUDIS et A. N. KOSTARAS, « Evolutionary algorithm based offline/online path planner for UAV navigation », *IEEE Transactions on Systems, Man, and Cybernetics - Part B (Cybernetics)*, vol. 33, no. 6, p. 898–912, 2003.
- [106] F. C. J. ALLAIRE, M. TARBOUCHI, G. LABONTÉ et G. FUSINA, « FPGA implementation of genetic algorithm for UAV real-time path planning », *Journal of Intelligent and Robotic Systems*, vol. 54, no. 1–3, p. 495—510, 2009.

- [107] N. OZALP et O. K. SAHINGOZ, « Optimal UAV path planning in a 3D threat environment by using parallel evolutionary algorithms », in *2013 International Conference on Unmanned Aircraft Systems (ICUAS)*, p. 308–317, IEEE, Mai 2013.
- [108] H. ERGEZER et K. LEBLEBICIOGLU, « Path planning for UAVs for maximum information collection », *IEEE Transactions on Aerospace and Electronic Systems*, vol. 49, no. 1, p. 502–520, 2013.
- [109] J. KENNEDY et R. C. EBERHART, « Particle swarm optimization », in *IEEE International Conference on Neural Networks*, p. 1942–1948, IEEE, Novembre 1995.
- [110] J. L. FOO, J. KNUTZON, V. KALIVARAPU, J. OLIVER et E. WINER, « Path planning of unmanned aerial vehicles using B-Splines and particle swarm optimization », *Journal of Aerospace Computing, Information, And Communication*, vol. 6, no. 4, p. 271–290, 2009.
- [111] V. ROBERGE, M. TARBOUCHI et G. LABONTÉ, « Comparison of parallel genetic algorithm and particle swarm optimization for real-time UAV path planning », *IEEE Transactions On Industrial Informatics*, vol. 9, no. 1, p. 132–141, 2013.
- [112] M. DORIGO, *Optimization, Learning and Natural Algorithms*. Thèse doctorat, Politecnico di Milano, 1992.
- [113] Y. HE, Q. ZENG, J. LIU, G. XU et X. DENG, « Path planning for indoor UAV based on ant colony optimization », in *25th Chinese Control and Decision Conference (CCDC)*, p. 2919–2923, IEEE, Mai 2013.
- [114] T. ZAZA et A. RICHARDS, « Ant colony optimization for routing and tasking problems for teams of UAVs », in *2014 UKACC International Conference on Control*, p. 652–655, IEEE, Juillet 2014.
- [115] B. LI, L.-G. GONG et W.-L. YANG, « An improved artificial bee colony algorithm based on balance-evolution strategy for unmanned combat aerial vehicle path planning », *The Scientific World Journal*, vol. 2014, p. 1–10, 2014.
- [116] A. A. KASSIM et B. V. K. V. KUMAR, « A neural network architecture for path planning », in *International Joint Conference on Neural Networks (IJCNN)*, p. 787–792, IEEE, Juin 1992.
- [117] R. GLASIUS, A. KOMODA et S. C. A. M. GIELEN, « Neural network dynamics for path planning and obstacle avoidance », *Neural Networks*, vol. 8, no. 1, p. 125–133, 1995.
- [118] S. X. YANG et C. LUO, « A neural network approach to complete coverage path planning », *IEEE Transactions on Systems, Man, and Cybernetics, Part B (Cybernetics)*, vol. 34, no. 1, p. 718–724, 2004.

- [119] V. KROUMOV et J. YU, « 3D path planning for mobile robots using annealing neural network », in *International Conference on Networking, Sensing and Control*, IEEE, March 2009.
- [120] N. WANG, X. GU, J. CHEN, L. SHEN et M. REN, « A hybrid neural network method for UAV attack route integrated planning », in *Advances in Neural Networks* (Y. W., H. H. et Z. N., eds), vol. 5553, in *Lecture Notes in Computer Science*, p. 226–235, Springer, Berlin, Heidelberg, 2009.
- [121] S. A. GAUTAM et N. VERMA, « Path planning for unmanned aerial vehicle based on genetic algorithm & artificial neural network in 3D », in *2014 International Conference on Data Mining and Intelligent Computing (ICDMIC)*, p. 1–5, IEEE, Septembre 2014.
- [122] Y. V. PEHLIVANOGLU, « A new vibrational genetic algorithm enhanced with a voronoi diagram for path planning of autonomous UAV », *Aerospace Science and Technology*, vol. 16, no. 1, p. 47–55, 2012.
- [123] K. WU, T. XI et H. WANG, « Real-time three-dimensional smooth path planning for unmanned aerial vehicles in completely unknown cluttered environments », in *2017 IEEE Region 10 Conference (TENCON)*, p. 2017–2022, IEEE, Novembre 2017.
- [124] H. TONG, W. W. CHAO, H. C. QIANG et X. Y. BO, « Path planning of UAV based on voronoi diagram and DPSO (discrete PSO) », *Procedia Engineering*, vol. 29, p. 4198–4203, 2012.
- [125] K. BENZAID, N. MANSOURI et O. LABBANI-IGBIDA, « A generalized dynamical model and control approach applied to multicopter aerial systems », in *International Conference on Modelling, Identification and Control*, IEEE, Novembre 2016.
- [126] J. F. GUERRERO CASTELLANOS, *Estimation de l'attitude et commande bornée en attitude d'un corps rigide : Application à un mini hélicoptère à quatre rotors*. Thèse doctorat, Université Joseph-Fourier - Grenoble I, 2008.
- [127] S. J. HADDADI et P. ZARAFSHAN, « Attitude control of an autonomous octocopter », in *IEEE International Conference on Robotics and Mechatronics*, p. 540–545, Octobre 2014.
- [128] K. BENZAID, N. MANSOURI et O. LABBANI-IGBIDA, « Robust trajectory tracking control of a quadrotor UAV », in *International Conference on Systems and Control*, IEEE, Octobre 2013.
- [129] K. BENZAID, N. MANSOURI et O. LABBANI-IGBIDA, « Analyse comparative de différentes stratégies de commande d'un quadrirotor », in *International Conference on Electronics Engineering*, Novembre 2013.
- [130] A. ALAIMO, V. ARTALE, C. L. ROSA MILAZZO et A. RICCIARDELLO, « PID controller applied to hexacopter flight », *Journal of Intelligent and Robotic Systems*, vol. 73, p. 261–270, Janvier 2014.

- [131] A. MARKS, J. F. WHIDBORNE et I. YAMAMOTO, « Control allocation for fault tolerant control of a VTOL octorotor », in *UKACC International Conference on Control*, p. 357–362, IEEE, Septembre 2012.
- [132] G.-X. DU, Q. QUAN et K.-Y. CAI, « Controllability analysis and degraded control for a class of hexacopters subject to rotor failures », *Journal of Intelligent and Robotic Systems*, vol. 78, p. 143–157, Avril 2015.
- [133] T. LI, Y. ZHANG et B. W. GORDON, « Passive and active nonlinear fault-tolerant control of a quadrotor UAV based on sliding mode control technique », *Proceedings of the Institution of Mechanical Engineers, Part I - Journal of Systems and Control Engineering*, vol. 277, no. 1, p. 12–23, 2013.
- [134] A.-R. MERHEB, H. NOURA et F. BATEMAN, « Design of passive fault - tolerant controllers of a quadrotor based on sliding mode theory », *International journal of applied Mathematics and computer science*, vol. 25, no. 3, p. 561—576, 2015.
- [135] D. ROTONDO, F. NEJJARI et V. PUIG, « Robust quasi - LPV model reference FTC of a quadrotor UAV subject to actuator faults », *International journal of applied Mathematics and computer science*, vol. 25, no. 1, p. 7–22, 2015.
- [136] G. DUCARD et M.-D. HUA, « Discussion and practical aspects on control allocation for a multi-rotor helicopter », in *Proceedings of the 1st International Conference on Unmanned Aerial Vehicles in Geomatics, UAV-g 2011*, p. 1–6, Septembre 2011.
- [137] K. BENZAID, R. MARIE, N. MANSOURI et O. LABBANI-IGBIDA, « Filtered medial surface based approach for 3D collision-free path planning problem », *Journal of Robotics*, vol. 2018, 2018. 9 pages.
- [138] B. T. INGERSOLL, J. K. INGERSOLL, P. DEFRANCO et A. NING, « UAV path-planning using bézier curves and a receding horizon approach », in *AIAA Modeling and Simulation Technologies Conference*, p. 1–14, Juin 2016.
- [139] O. K. SAHINGOZ, « Generation of Bezier curve-based flyable trajectories for multi-UAV systems with parallel genetic algorithm », *Journal of Intelligent and Robotic Systems*, vol. 74, no. 1–2, p. 499—511, 2014.
- [140] M. LIZARRAGA et G. H. ELKAIM, « Spatially deconflicted path generation for multiple UAVs in a bounded airspace », in *IEEE/ION Position, Location and Navigation Symposium*, p. 1213–1218, Mai 2008.
- [141] L. CHEN, S. WANG, H. HU et K. MCDONALD-MAIER, « Bézier curve based trajectory planning for an intelligent wheelchair to pass a doorway », in *UKACC International Conference on Control*, p. 339–344, IEEE, Septembre 2012.

- [142] H. BLUM, « A transformation for extracting new descriptors of shape », in *Models for the Perception of Speech and Visual Form*, p. 362–380, MIT Press, 1967.
- [143] R. OGNIEWICZ et M. ILG, « Voronoi skeletons : theory and applications », in *Computer Society Conference on Computer Vision and Pattern Recognition*, p. 63–69, IEEE, 1992.
- [144] W. H. HESSELINK et J. B. ROERDINK, « Euclidean skeletons of digital image and volume data in linear time by the integer medial axis transform », *IEEE Transactions on Pattern Analysis And Machine Intelligence*, vol. 30, no. 12, p. 2204–2217, 2008.
- [145] R. MARIE, O. LABBANI-IGBIDA et E. M. MOUADDIB, « The delta medial axis : A fast and robust algorithm for filtered skeleton extraction », *Pattern Recognition*, vol. 56, p. 26–39, 2016.

Titre : Stratégies de commande de véhicules à déplacement vertical

Résumé :

Ce travail porte essentiellement sur l'étude de la commande des véhicules aériens autonomes à décollage et atterrissage verticaux. Nous nous concentrons sur les drones multi-rotors. Nous étudions d'abord la modélisation dynamique de ce type de drones. En utilisant le formalisme d'Euler-Newton, les équations différentielles sont dérivées sous forme d'un modèle généralisé pour décrire le comportement d'un drone multi-rotor possédant un nombre pair N de rotors. Dans un deuxième temps, nous analysons les approches de commandes utilisées dans la littérature pour la stabilisation et le suivi de chemins / trajectoires des multi-rotors. En se basant sur le modèle dynamique développé, nous proposons une structure de commande hiérarchique pour la stabilisation du véhicule. L'approche de commande proposée consiste à combiner l'algorithme PID avec l'algorithme Back-stepping à action intégrale. Nous discutons également l'effet de panne sur le comportement du véhicule. Le guidage des drones est un autre élément abordé dans ce travail. Nous présentons une étude des différents algorithmes appliqués pour la planification de chemins 3D. Nous présentons ensuite notre contribution dans la planification de chemins pour drones. Il s'agit d'une méthode de planification de chemins, minimisant la longueur du chemin et maximisant la distance aux obstacles, basée sur la squelettisation de l'environnement 3D tout en prenant en considération la taille du véhicule. La méthode proposée a été testée et s'avère capable de produire des solutions optimales en termes de sûreté et de longueur de chemin.

Mots-clés : drone, modélisation, guidage, contrôle, squelettisation, PID, Back-stepping.

Title : Control strategies for vertical take-off and landing vehicles

Abstract:

This work addresses the study of the control of autonomous **Vertical Take-Off and Landing (VTOL)** aircrafts. We focus on multi-rotor **Unmanned Aerial Vehicles (UAVs)**. We first study the dynamic modeling of this class of **UAVs**. Using the Euler-Newton formalism, the differential equations are derived as a generalized model to describe the behavior of a multi-rotor **UAV** with an even number N of rotors. In a second step, we analyze the control approaches used in the literature for multi-rotors stabilization and path / trajectory tracking. Based on the dynamic model developed, we propose a hierarchical structure for the vehicle control. The proposed control approach consists of combining the **Proportional, Integral, Derivative (PID)** algorithm with the integral Back-stepping algorithm. We also discuss the effect of a complete rotor fault on the behavior of the vehicle. The guidance of UAVs is another element addressed in this work. We present a study of the different algorithms applied for the planning of 3D paths. We then present our contribution in the path planning for UAVs. It is a method of path planning, minimizing the path length and maximizing the distance to obstacles, based on the skeletonization of the 3D environment while taking into account the size of the vehicle. The proposed method has been tested and proves capable of producing optimal solutions in terms of safety and path length.

Keywords: UAV, modeling, guidance, control, skeletonization, PID, Back-stepping.

العنوان : استراتيجيات التحكم في الطائرات بدون طيار ذات الإقلاع و الهبوط العمودي

ملخص :

يتناول هذا العمل دراسة التحكم في الطائرات بدون طيار ذات الإقلاع و الهبوط العمودي، مع التركيز على الطائرات بدون طيار متعددة الدورات. ندرس أولاً التّمدجة الديناميكية لهذا النوع من الطائرات بدون طيار. نقوم باستنباط المعادلات التفاضلية على شكل نموذج معمم لوصف سلوك طائرة بدون طيار تحتوي على عدد زوجي N من الدورات. في الخطوة الثانية، نقوم بتحليل أساليب و تقنيات التحكم المعتمدة في الأدبيات لتحقيق استقرار و تتبع المسارات لدى الطائرات بدون طيار ذات الدورات المتعددة. وبناءً على النموذج الديناميكي المعمم الذي تمّ تطويره، نقترح هيكلًا هرميًا من أجل التحكم في الطائرة بدون طيار. يعتمد منهج التحكم المقترح على دمج خوارزميتي PID و Integral Back-steppping . كما نتطرق أيضا لمناقشة تأثير العطل الكلي لأحد الدورات على سلوك الطائرة. إنّ توجيه الطائرات بدون طيار هو عنصر آخر تمّ تناوله في هذا العمل. حيث نقدّم دراسة للخوارزميات المختلفة المطبقة على تخطيط المسارات ثلاثية الأبعاد. ثمّ نقدّم مساهمتنا في تخطيط مسارات الطائرات بدون طيار. إنّها طريقة لتخطيط المسار، مع خفض طول المسار ورفع المسافة إلى العقبات قدر الامكان، بناءً على استخراج الهيكل العظمي للبيئة ثلاثية الأبعاد مع أخذ حجم الطائرة في الاعتبار. تمّ اختبار الطريقة المقترحة وتمكنا من تقديم مسارات جيدة من حيث السلامة وطول المسار.

الكلمات الدالة : طائرة بدون طيار، نمذجة، توجيه، تحكم، هيكل عظمية،

Back-steppping ، PID