

Université de Limoges

ED 653 : SCIENCES ET INGENIERIE

Faculté des Sciences et Techniques – Institut de Recherche XLIM

Thèse pour obtenir le grade de
Docteur de l'Université de Limoges
Automatique, Robotique, Électronique

Présentée et soutenue par

Tariq ZIOUD

Le 11 décembre 2023

**CONCEPTION, MODÉLISATION ET COMMANDE ROBUSTE DE DRONES À ROTORS PIVOTANTS POUR
L'INTERACTION AÉRIENNE AUTONOME**

Thèse dirigée par Juan-Antonio Escareno et Ouiddad Labbani Igbida

JURY :

Président du jury

M. Nicolas Marchand, Directeur de Recherche - GIPSA-lab - Université de Grenoble Alpes

Rapporteurs

M. Pedro Castillo Garcia, Directeur de Recherche - CNRS - HEUDIASYC - UTC

M. Nicolas Marchand, Directeur de Recherche - GIPSA-lab - Université de Grenoble Alpes

Examineurs

M. Sylvain Bertrand, Maître de Recherche - ONERA - Université Paris-Saclay

M. Paolo Robuffo Giordano - Directeur de Recherche CNRS - Université de Rennes



À la mémoire de mon grand-père Mohamed Zouhry,

Remember to look up at the stars and not down at your feet. Try to make sense of what you see and wonder about what makes the universe exist. Be curious. And however difficult life may seem, there is always something you can do and succeed at. It matters that you don't just give up.

Stephen William Hawking

Remerciements

À mes directeurs de thèse, Juan-Antonio Escareno et Ouiddad Labbani-Igbida,

Je tiens à exprimer ma gratitude la plus sincère envers mes directeurs de thèse, Antonio et Ouiddad, pour leur mentorat exceptionnel. Vous étiez mes guides et gourous, toujours disponibles pour prodiguer des conseils avisés. Votre confiance en mes capacités m'a encouragé à repousser mes limites. Travailler à vos côtés durant ces trois années de thèse a été un grand honneur et a enrichi mon parcours académique.

À mes chers parents, mes sœurs et à toi, ma chère épouse Wiame,

Je souhaite vous remercier du fond du cœur pour votre soutien indéfectible tout au long de cette aventure. Votre compréhension et votre patience pendant les périodes intenses où la recherche m'a éloigné ont été inestimables. Votre soutien constant a été ma principale source de motivation, et je suis infiniment reconnaissant de vous avoir à mes côtés.

Ce travail n'aurait jamais vu le jour sans votre aide à tous, et je ressens une profonde gratitude envers chacun d'entre vous.

Tariq ZIOUD

Droits d'auteurs

Cette création est mise à disposition selon le Contrat :

« Attribution-Pas d'Utilisation Commerciale-Pas de modification 3.0 France »

disponible en ligne : <http://creativecommons.org/licenses/by-nc-nd/3.0/fr/>



Table des Matières

1	Introduction	21
1.1	Contextualisation et problématique	23
1.1.1	Perspective historique sur les drones	23
1.1.2	Les drones aujourd’hui	27
1.1.3	Revue de la littérature	30
1.1.4	Contexte de la thèse	35
1.1.5	Contributions principales de la thèse	36
1.2	Note de publications	39
2	Modèle dynamique du Xtilt	40
2.1	Les architectures de drones étudiées	42
2.1.1	Quadrirotor à propulseurs fixes	42
2.1.2	Quadrirotor à propulseurs pivotants	42
2.2	Définition des différents repères	43
2.3	Dynamique du drone	45
2.3.1	Paramètres cinématiques	45
2.3.2	Paramètres physiques	47
2.3.3	Équations de Lagrange	48
2.3.4	Bilan des forces et des couples	49
2.4	Limitation du torseur des actions mécaniques	55
2.4.1	Cas des bras pivotants Xtilt	58
2.4.2	Quadrirotor à bras fixes	62
2.5	Mécanique des fluides numérique	62
2.5.1	Modèle CFD	63
2.6	Conclusion	77
3	Contrôle et observation du quadrirotor	78
3.1	Modèle d’observation et de contrôle	80
3.2	Stratégies d’observation	82

3.2.1	Filtre de Kalman linéaire à état étendu	83
3.2.2	Super Twisting Observer	85
3.3	Stratégies de commande	86
3.3.1	Contrôle en orientation	86
3.3.2	Contrôle en position	86
3.4	Conclusion	99
4	Validation dans des environnements de simulation	100
4.1	Quadrirotor à bras fixes	101
4.1.1	ES-iLQG	101
4.2	Quadrirotor à bras pivotants (Xtilt)	106
4.2.1	Contrôleur PID	106
4.2.2	Intégrale MPC & MPSMC	109
4.2.3	ES-LKF & STO	117
4.3	Conclusion	119
5	Prototypes et résultats expérimentaux	120
5.1	Évaluation de l’approche d’estimation et de contrôle pour l’ES-iLQG	121
5.1.1	Cas 2D utilisant le drone Crazyflie	122
5.1.2	Cas 3D utilisant le drone Coex Clover 4	127
5.2	Validation expérimentale du drone à bras pivotant Xtilt	135
5.2.1	Architectures embarquées	135
5.2.2	Scénarios et résultats expérimentaux	141
5.3	Conclusion	147
6	Conclusion et perspectives	148
6.1	Résumé	149
6.2	Travaux futurs, quelques pistes	150
6.2.1	Génération de trajectoire	151
6.2.2	Une voilure fixe, pourquoi pas	157
6.3	Leçon tirée	157
A	Annexes	158
A.1	Quelques améliorations pour le système de propulsion	159
A.2	Piste d’atterrissage pour drone sur le robot Cinteo	160
B	Bibliographie	167
	Références	168

Liste des travaux 180

Table des Figures

1.1	(a) Un robot musical, (b) Horloge éléphant tirés d'un exemplaire du livre d'al-Jazari sur les automates, Kitab fi ma'arif al-hiyal al-handasiya	24
1.2	La "vis aérienne" conçue par Léonard de Vinci.	24
1.3	Planeur conçu par George Cayley.	25
1.4	Kettering Bug, une torpille aérienne sans pilote.	26
1.5	(a) Boeing B-17 à basse altitude, (b) V-1 en préparation pour un lancement . .	26
1.6	SD-1 Observer pour la reconnaissance tactique.	27
1.7	(a) MQ-1 Predator, (b) MQ-9 Reaper en vol	27
1.8	Décollage du drone Flyability dans la Grotte Eugenio Commission Boegan par ESA, Natalino Russo www.natalinorusso.it	28
1.9	(a) Amazon Prime Air, (b) Google Wing en vol.	28
1.10	Pulvérisation d'un champ par un drone dji, https://ag.dji.com/fr/about-us	29
1.11	Dji Matrice 300 avec l'échantillonneur d'eau de Nero https://nerosampler.com/	29
1.12	Classification des drones en fonction de leur nature : sous-actionnés, totalement actionnés.	30
1.13	Exemples de drones sous-actionnés (a) Xdrone quadricoptère, (b) DJI F550 hexarotor	31
1.14	Exemples de drones totalement actionnés (a) Tilt-Hex [19] source image , (b) Hélicoptère coaxial [18] source image	31
1.15	Quelques illustrations des différents drones tiltrotor décrits dans la littérature. . .	33
1.16	Prélèvement d'échantillons d'eau dans une rivière	36
1.17	Les deux versions du Xtilt développées dans les travaux de thèse, (a) la version 1 avec une carte Nvidia Jetson Nano, (b) la version 2 avec un contrôleur basée sur la Raspberry pi4. La distinction dans le positionnement des hélices doit également être relevée.	36
1.18	Description du ES-iLQG (Extended State Integral Linear Quadratic Gaussian).	37
1.19	Description du contrôleur basé sur le modèle prédictif.	38

2.1	Illustration des différentes forces, les quatre moteurs génèrent une poussée totale représentée par le vecteur rose, le vecteur vert représente la composante verticale de cette force et le vecteur bleu la composante horizontale (drone Crazyflie 2.1).	43
2.2	Description des différentes possibilités d'enveloppe de contrôle: (a) Contrôle différentiel du lacet (b) mouvement horizontal sans changer d'orientation (c) vol stationnaire non horizontal.	43
2.3	Description des différents repères considérés pour les deux structures étudiées .	45
2.4	Représentation des couples et des forces. Les différents repères et notations des forces/couples sont indiqués à droite.	50
2.5	Représentation des différentes distances moteurs par rapport au centre de gravité du drone.	51
2.6	Représentation du vecteur vitesse et des angles d'attaque ι et de dérive ν	52
2.7	$\ \mathbf{H}^{\mathcal{B}}\ $ en fonction de \tilde{u}_x pour $\phi = 0, \theta = 0$ et $\psi = 0, \tilde{u}_y = \tilde{u}_z = 0$	57
2.8	$\ \mathbf{H}^{\mathcal{B}}\ $ en fonction de \tilde{u}_x pour $\phi = 0, \theta = 0$ et $\psi = 0, \tilde{u}_y = 0$ et $\tilde{u}_z = 9.81$	57
2.9	$\ \mathbf{H}^{\mathcal{B}}\ $ en fonction de \tilde{u}_y pour $\phi = 0, \theta = 0$ et $\psi = 0, \tilde{u}_x = \tilde{u}_z = 0$	58
2.10	$\ \mathbf{H}^{\mathcal{B}}\ $ en fonction de \tilde{u}_z pour $\phi = 0, \theta = 0$ et $\psi = 0, \tilde{u}_x = \tilde{u}_y = 0$	58
2.11	$\ \mathbf{H}^{\mathcal{B}}\ $ en fonction de \tilde{u}_z pour $\phi = 0, \theta = \frac{\pi}{4}$ et $\psi = 0, \tilde{u}_x = \tilde{u}_y = 0$	59
2.12	Description des différentes possibilités d'enveloppe de contrôle: (a) Contrôle différentiel du lacet (b) mouvement horizontal sans changer d'orientation. . . .	61
2.13	Conditions aux limites du modèle CFD, une représentation des différents frontières.	64
2.14	Différentes perspectives après le raffinement du maillage par snappyHexMesh. Le maillage en zone (2) est plus affiné qu'en zone (1)	65
2.15	Coefficient de traînée C_d en fonction du temps	66
2.16	Représentation du coefficient de traînée en fonction de ι angle d'attaque et de ν angle de dérive	66
2.17	Vue en coupe sur le plan ($e_1 e_3$) de la pression de l'air pour deux angles d'attaque différents. Dans les deux cas on a $\nu = 0^\circ$: (a) pour $\iota = 0^\circ$ (b) pour $\iota = 60^\circ$	67
2.18	Le coefficient de traînée C_d et son modèle linéaire en fonction de l'angle d'attaque ι , en orange la courbe du modèle et en bleu les données réelles.	67
2.19	Représentation du coefficient de portance C_l en fonction de ι et de ν	68
2.20	Coefficient de portance C_l et son modèle linéaire en fonction de l'angle d'attaque ι , en orange la courbe du modèle et en bleu les données réelles	68
2.21	Représentation du coefficient de portance latérale C_s en fonction de l'angle d'attaque ι et de l'angle de dérive ν	69
2.22	Vue en coupe sur le plan ($e_1 e_2$) de la pression de l'air pour deux angles de dérives différents. Dans les deux cas on a $\iota = 0^\circ$: (a) pour $\nu = 0^\circ$ (b) pour $\nu = 60^\circ$	70

2.23	Représentation des coefficients des moments en fonction de l'angle d'attaque ι et de l'angle de dérive ν , (a) pour le moment sur l'axe de roulis, (b) pour l'axe de lacet	70
2.24	Représentation du coefficient du moment sur l'axe de tangage en fonction de l'angle d'attaque ι et de l'angle de dérive ν	71
2.25	Le coefficient de moment de tangage C_θ et son modèle en fonction de l'angle d'attaque ι , en orange la courbe du modèle et en bleu les données réelles	71
2.26	Conditions aux limites du modèle CFD pour une hélice du Xtilt, une représentation des différentes frontières	72
2.27	Différentes perspectives après le raffinement du maillage par <code>snappyHexMesh</code> autour de l'hélice	72
2.28	La poussée et le couple de trainée générés par l'hélice pour une vitesse de rotation de 1000 rad/s sans présence de vent.	73
2.29	Représentation volumique du flux d'air (U_z) sur l'axe vertical que l'hélice génère à différents instants	74
2.30	La poussée et le couple de trainée générés par l'hélice pour une vitesse de rotation de 1000 rad/s en présence d'un vent de 10 m/s	74
2.31	Représentation volumique du flux d'air que l'hélice génère à différents instants en présence d'un vent horizontal https://youtu.be/F2WQaSgTKbE	75
2.32	Représentation du cas de simulation: un angle d'inclinaison de $\frac{\pi}{3}$	76
2.33	La poussée et le couple de trainés générés par les deux hélices pour une vitesse de rotation de 1500rad/s et un angle d'inclinaison de $\frac{\pi}{3}$.	76
2.34	Représentation volumique du flux d'air généré pas les deux hélices pour une inclinaison $\frac{\pi}{3}$. https://youtu.be/EwUUabf-RZo	77
3.1	Description de l'architecture de contrôle ES-iLQG (Extended State Integral Linear Quadratic Gaussian).	87
3.2	Schéma de l'approche proposée basée MPC.	92
3.3	Schéma détaillé de l'approche proposée basée MPC.	93
3.4	Description du schéma de contrôleur intégral MPC.	94
3.5	Description du contrôleur MPSMC.	99
4.1	Organisation des simulations réalisées pour le quadrirotor à bras fixes.	102
4.2	Description de l'architecture de contrôle ES-iLQG (Extended State Integral Linear Quadratic Gaussian).	102

4.3	(a) Comportement du contrôleur en position et vitesse sur l'axe y face à différentes perturbations (<i>courbe supérieure</i>) : Bruit gaussien, perturbation constante de la valeur 0.5N sur l'intervalle de temps [40;80], perturbation sinusoïdale $\rho_y = 0.5 \sin(0.01\pi t)$ à partir de $t = 80s$. Les trajectoires désirées sont $y_d = 0.5 \sin(0.1\pi t)$ et $\dot{y}_d = 0.05\pi \cos(0.1\pi t)$, on retrouve aussi l'estimation de la perturbation, (b) Comportement du contrôleur en position et vitesse sur l'axe z face à différentes perturbations (<i>courbe supérieure</i>): Bruit gaussien, perturbation constante de la valeur 0.5N sur l'intervalle de temps [40;80], perturbation sinusoïdale $\rho_y = 0.5 \sin(0.01\pi t)$ à partir de $t = 80s$. Les trajectoires désirées sont $z_d = 0.5 \cos(0.1\pi t)$ et $\dot{z}_d = -0.05\pi \sin(0.1\pi t)$, on retrouve aussi l'estimation de la perturbation.	104
4.4	Description de l'environnement de simulation Ros/Gazebo.	104
4.5	(a) Suivi d'une trajectoire hélicoïdale le drone étant soumis à des perturbations, décrites par $x_d = 0.5 \sin(0.2\pi t)$, $y_d = 0.5 \cos(0.2\pi t)$, $z_d = 0.5 + 0.03t$, (b) Erreur en position de suivi d'une trajectoire hélicoïdale le drone étant soumis à des perturbations.	105
4.6	Erreur en vitesse de suivi d'une trajectoire hélicoïdale le drone étant soumis à des perturbations, la vitesse désirée est $\dot{x}_d = 0.1\pi \cos(0.2\pi t)$, $\dot{y}_d = -0.1\pi \sin(0.2\pi t)$, $\dot{z}_d = 0.03m/s$	105
4.7	Description du ES-iLQG (Extended State Integral Linear Quadratic Gaussian).	106
4.8	Suivi de trajectoire 3D pour la navigation holonomique (ϕ_d) et non holonomique (ψ_d)	108
4.9	(a) Performance de suivi de trajectoire lorsque la dynamique de l'axe y est pilotée par un mouvement de lacet, (c) L'évolution des angles d'inclinaison, les pics élevés en α_i sont dus à l'évolution rapide des ψ_d de -179° à 179° et l'absence de saturation sur la commande, (b) Performance de suivi de trajectoire lorsque la dynamique de l'axe y est commandée par le mouvement de roulis (ϕ_d), (d) Évolution des angles d'inclinaison.	108
4.10	Comparaison de la consommation d'énergie entre la configuration classique et Xtilt	109
4.11	Architecture de contrôle en boucle fermée proposée pour l'intégral MPC et MPSMC	110
4.12	Description complète de la boucle de contrôle de la simulation, détaillant les différents nœuds ROS2 et les messages échangés entre eux ainsi que l'environnement de simulation https://youtu.be/kjAZyqdocH4	110
4.13	Test de l'effet de l'horizon d'optimisation sur le comportement global du suivi en présence et en absence de perturbations constantes	111

4.14	Comparaison entre deux suivis de trajectoire à l'aide de l'intégral MPC. [<i>courbe verte</i>] correspond au modèle sans estimation de la perturbation, tandis que [<i>courbe bleu</i>] tient compte de cette estimation.	112
4.15	Comparaison en suivi de trajectoire en z entre l'intégral MPC et l'ES-iLQG . . .	113
4.16	Représentation 3D du suivi de trajectoire en présence de perturbations variables en considérant l'intégral MPC avec deux fonctions de coût différentes.	114
4.17	(a) Erreur de position en présence d'une perturbation variable en considérant un intégral MPC avec deux fonctions de coût différentes, (b) Les différents gains pour l'intégral MPC sur l'axe des x en présence de perturbations variables en considérant deux fonctions de coût différentes.	114
4.18	(a) Suivi de l'erreur de position en présence de perturbations variables dans le temps pour le MPSMC en tenant compte de deux fonctions de coût différentes, (b) Les différents gains pour la commande MPSMC sur l'axe z en présence de perturbations variables compte tenu de deux fonctions de coût différentes.	115
4.19	Erreur quadratique moyenne des deux contrôleurs proposés basés sur le MPSMC et l'intégral MPC pour la même fonction de coût	116
4.20	(a) Erreur quadratique moyenne des deux contrôleurs proposés basés sur le MPSMC et l'intégral MPC pour la même fonction de coût (coût 1), en présence de perturbation avec une forte composante dynamique, (b) Erreur quadratique moyenne des deux contrôleurs proposés basés sur le MPSMC et l'intégral MPC pour la même fonction de coût (coût 2), en présence de perturbation avec une forte composante dynamique.	117
4.21	Erreur de suivi et gain kp_z pour l'axe z en utilisant le contrôle intégral MPC, avec l'introduction d'une perturbation constante à l'instant $t = 40$	117
4.22	Perturbation estimée, une incertitude sur la masse du drone et une perturbation constante est introduite à l'instant $t = 40s$	118
4.23	(a) Erreur de suivi axe z pour l'intégral MPC en présence de perturbation en utilisant deux observateurs différents, (b) gain kp_z pour l'intégral MPC en présence de perturbation en utilisant deux observateurs différents.	118
5.1	Crazyflie 2.1 avec marqueur Optitrack	122
5.2	Crazyflie 2.1 avec un marqueur Optitrack	122
5.3	Architecture du système Crazyflie 2.1	123

5.4	(a) Suivi d'une trajectoire 2D circulaire avec et sans présence de perturbations constantes, décrite par $y_d = 0.5 \cos(0.01\pi t)$, $z_d = 1.5 + 0.5 \sin(0.01\pi t)$, (b) Erreur de suivi d'une trajectoire circulaire, décrite par $y_d = 0.5 \cos(0.01\pi t)$, $z_d = 1.5 + 0.5 \sin(0.01\pi t)$. https://youtu.be/sdiZnxvUEC4 https://youtu.be/kCSuzYDq_f0	125
5.5	(a) Suivi d'une trajectoire en vitesse circulaire avec et sans présence de perturbations constantes, décrite par $\dot{y}_d = -0.005\pi \sin(0.01\pi t)$, $\dot{z}_d = 0.005\pi \cos(0.01\pi t)$, (b) Erreur de suivi d'une trajectoire en vitesse circulaire, décrite par $\dot{y}_d = -0.005\pi \sin(0.01\pi t)$, $\dot{z}_d = 0.005\pi \cos(0.01\pi t)$.	126
5.6	(a) Perturbations estimées pour l'axe y et z, (b) Perturbation estimée en [g].	127
5.7	Architecture embarquée pour le drone Clover 4.2. Le pilote automatique PX4 exécute le contrôleur d'attitude à 250 Hz, tandis que le Raspberry Pi intègre la localisation du drone à 50 Hz, le filtre de Kalman linéaire à 100 Hz et le contrôleur de position à 50 Hz.	128
5.8	Configuration expérimentale du drone Clover 4.2 (vue dessous et dessus).	128
5.9	(a) Suivi d'une trajectoire 3D hélicoïdale, décrite par $x_d = 0.5 \sin(0.2\pi t)$, $y_d = 0.5 \cos(0.2\pi t)$, $z_d = 0.5 + 0.03t$, (b) Erreurs de suivi de position de la trajectoire hélicoïdale.	130
5.10	Estimation de la perturbation subie par le drone dans le référentiel fixe. Elle est estimée en termes d'accélération, la multiplication par la masse nominale donne l'équivalent en Newtons.	130
5.11	(a) Suivi de la trajectoire de la vitesse hélicoïdale sous différentes perturbations, la vitesse désirée est $\dot{x}_d = 0.1\pi \cos(0.2\pi t)$, $\dot{y}_d = -0.1\pi \sin(0.2\pi t)$, $\dot{z}_d = 0.03m/s$, (b) Erreurs de suivi de la vitesse de la trajectoire hélicoïdale.	131
5.12	Une séquence de photos du drone en suivie de la trajectoire hélicoïdale. (Vidéo https://youtu.be/M1x91ecZ5Ek).	131
5.13	(a) Suivi d'une trajectoire en forme de huit $x_d = 0.5 \sin(0.135\pi t)$, $y_d = 0.5 \sin(0.27\pi t)$ et $z_d = 1$ m, (b) Erreurs de suivi de la position de la trajectoire en forme de huit.	132
5.14	Estimation des perturbations lors du suivi de la trajectoire en forme de huit	132
5.15	(a) Suivi de la trajectoire de vitesse en forme de huit sous différentes perturbations, la vitesse souhaitée est la suivante $\dot{x}_d = 0.675\pi \cos(0.135\pi t)$, $\dot{y}_d = 0.135\pi \cos(0.27\pi t)$, (b) Erreurs de suivi de la vitesse de la trajectoire en forme de huit.	133
5.16	Suivi de la trajectoire 3D du drone examiné dans les scénarios expérimentaux, avec et sans compensation en direct des perturbations.	133
5.17	Ecart quadratique moyen de la trajectoire hélicoïdale. On remarque que l'erreur sur l'axe z est plus importante dans le cas où la perturbation n'est pas compensée.	134
5.18	Estimation des perturbations dans les deux cas d'études.	134

5.19	Les deux versions du Xtilt, (a) la version basée sur la carte Nvidia Jetson Nano, (b) la version basée sur la carte Raspberry pi4. Il convient également de noter la différence de positionnement des hélices.	135
5.20	Description globale de l'architecture des prototypes expérimentaux (a) version basée sur Nvidia Jetson Nano, (b) version basée sur la carte Raspberry pi4 . . .	136
5.21	Plate-forme de mesure des principaux moments d'inertie du drone à l'aide d'un pendule bifilaire	137
5.22	Identification des moments d'inertie, la période a été calculée par $t_0 = \frac{t}{N}$, t_t est le temps pris par les oscillations, N est le nombre d'oscillations, pour l'axe y $t_0 = 0,8482[s]$	138
5.23	Banc d'essai pour caractériser les coefficients de portance et de traînée https://youtu.be/v9b1Yvb3CFM	138
5.24	Mesures de la poussée et du couple générés par l'hélice en fonction de la vitesse de rotation des moteurs, effectuées sur la plateforme expérimentale.	139
5.25	Dispositif expérimental pour mesurer le couple gyroscopique.	140
5.26	Mesure expérimentale du couple généré par l'effet gyroscopique de l'hélice. . .	140
5.27	Suivi de la position du servomoteur Dynamixel pour une consigne désirée avec une fréquence variable dans le temps. $\alpha_d(t) = 17 \sin(2\pi 0.005t^2)$	141
5.28	Le Xtilt sur la plateforme de test pour ajuster les paramètres des différents contrôleurs. https://youtu.be/aDeMwgGhqEc	142
5.29	(a) Suivi d'une orientation désirée pour le tangage, la consigne désirée est $\theta_d = \frac{\pi}{3} \sin(0.2\pi t)$, $\dot{\theta}_d = 0.2 \frac{\pi^2}{3} \cos(0.2\pi t)$, (b) La consigne désirée pour le bras 1 ainsi que sa position mesurée en suivant une orientation désirée pour le tangage, la consigne désirée est $\theta_d = \frac{2\pi}{9} \sin(0.2\pi t)$, $\dot{\theta}_d = 0.2 \frac{2\pi^2}{9} \cos(0.2\pi t)$. https://youtu.be/scisQvd7nL4	142
5.30	Les essais en vol ont été réalisés en utilisant la plateforme illustrée à la figure 5.20a. Les résultats de ces essais sont présentés sous forme de courbe, où sont affichées les données de position et d'orientation mesurées par le drone, ainsi que les mesures obtenues à l'aide du système Optitrack. https://youtu.be/qkJjG5i-ZL8	143
5.31	Description détaillée de l'architecture de contrôle du prototype expérimental . .	144

5.32	Extrait des tests en vol : (a) Suivi d'une trajectoire circulaire pour l'axe x (b) Vol stationnaire non horizontal. https://youtu.be/SDeBwmfgua0	145
5.33	Résultats expérimentaux pour le contrôleur intégral MPC (a) Suivi en position et orientation, (b) Erreur de suivi pour la position et l'orientation.	145
5.34	Résultats expérimentaux pour le contrôleur intégral MPC (a) Suivi en vitesse, (b) Erreur de suivi en vitesse.	146
5.35	Résultats expérimentaux pour le contrôleur intégral MPC (a) Les gains pour l'axe x, (b) Les gains pour l'axe z.	146
5.36	Résultats expérimentaux pour le contrôleur intégral MPC, les gains pour l'axe y	147
6.1	La trajectoire suivie par un Milan royal lors de son approche de sa cible. source vidéo : https://youtu.be/AYOx-iCMZhk&t=127	151
6.2	(a) Suivi d'une trajectoire bio-inspirée $x_d = (0.04t^2 + 0.4t) \cos(\psi)$, $y_d = (0.04t^2 + 0.4t) \sin(\psi)$ et $z_d = 2 - 0.4t + 0.016t^2$, (b) Erreurs de suivi de la position de la trajectoire bio-inspirée.	152
6.3	(a) Suivi de la trajectoire de vitesse bio-inspirée, la vitesse souhaitée est $\dot{x}_d = (0.4 + 0.08t) \cos(\psi)$, $\dot{y}_d = (0.4 + 0.08t) \sin(\psi)$ et $\dot{z}_d = -0.4 + 0.032t$, (b) Erreurs de suivi de la vitesse de la trajectoire bio-inspirée.	152
6.4	Trajectoire de vol dynamique d'un Albatros exploitant le vent comme source d'énergie. [136]	153
6.5	Résultats d'optimisation énergétique de la trajectoire (a) [haut] La position dans le plan $x - y$, [bas] La position sur le plan $z - y$ (b) Les entrées de commande, [haut] l'angle de tangage, [bas] Le coefficient de portance	155
6.6	Une période de la trajectoire optimale.	156
6.7	L'océanite tempête en pleine chasse, source : https://youtu.be/X5pG1OQ0VfM	156
6.8	XWing une version améliorée du Xtilt.	157
A.1	Le schéma de contrôle pour l'asservissement de la force générée par les hélices	159
A.2	Schéma de contrôle pour l'asservissement de la force générée par les hélices, α est l'angle d'inclinaison, qui est connu, F^z représente la force mesurée portée par l'axe e_3 du repère du drone. Cela signifie que F^z est la composante de la force dans la direction verticale. F^d est la force désirée par le contrôleur, c'est-à-dire la valeur de force que le contrôleur souhaite atteindre pour réaliser une tâche donnée ou pour maintenir le drone dans une certaine configuration de vol. . . .	160
A.3	La plateforme expérimentale Cinteo avec la piste d'atterrissage et le drone (Clover 4).	160
A.4	Carte de marquage Aruco pour la piste d'atterrissage.	161

A.5	Lorsque la mesure de position n'est pas disponible, nous utilisons le second modèle d'observation. La principale différence avec le premier modèle réside dans le rang de la matrice d'observation. En cas de perte de la mesure de position, nous utilisons comme condition initiale la dernière mesure obtenue.	162
A.6	Simulation sous Gazebo de Cinteo et Clover 4.	163
A.7	Les résultats obtenus pour l'estimation de position en suivi d'une trajectoire hélicoïdale (a) On retrouve ici les positions souhaitées ainsi que les différentes mesures filtrées, (b) Les erreurs de mesure sont calculées en prenant comme référence la mesure réalisée par le flux optique et les marqueurs Aruco. https://youtu.be/VQ8nfh_NVd4	164
A.8	Ces images ont été prises par le drone lors de suivi d'une trajectoire hélicoïdale à différents instants. (a) Pour $t = 5s$ lors de la phase de décollage (b) Pour $t = 78s$ lors de la phase d'atterrissage. La ligne rouge correspond à l'axe des x , la ligne verte à l'axe des y et la ligne bleue à l'axe des z du drone.	165
A.9	Le robot Agilex Scout Mini portant la piste d'atterrissage et le drone.	165
A.10	Images prises par le drone en survolant le robot agricole Cinteo et la piste d'atterrissage à l'extérieur.	166

Liste des Tableaux

1.1	Classification des drones multirotors de la littérature, N_m est le nombre de moteurs, N_s est le nombre de servomoteurs, SSA pour sous-actionné, EA pour entièrement actionné, SRA pour sur-actionné, OM omnidirectionnel	34
2.1	Les différentes configurations et orientations testées.	57
4.1	Paramètres du filtre de Kalman et du LQR	102
4.2	Paramètres physiques du Coex Clover 4 données constructeur	103
4.3	Paramètres du drone Xtilt. Pour le coefficient de traînée, nous prenons la valeur moyenne, tpwm est la largeur d'impulsion multipliée par $1e^3$, pour les paramètres aérodynamiques, les valeurs ont été obtenues par la CFD, tandis que les autres paramètres ont été mesurés expérimentalement. Toutefois, il convient de noter que la matrice d'inertie pour les bras a été obtenue par la CAO.	106
4.4	Le RMSE en fonction de l'horizon et de l'intervalle de temps correspondant à la présence ou non de la perturbation	112
4.5	Paramètres des deux fonctions de coût utilisées. L'intérêt de tester ces deux fonctions de coût est de quantifier l'impact des matrices de coût sur les gains et l'erreur de suivi. Le coût 1 impose une contrainte plus importante sur l'erreur de suivi, tandis que le coût 2 impose une contrainte sur la commande.	113
4.6	Paramètres différents pour les deux fonctions de coût utilisées	116
5.1	Paramètres du filtre de Kalman et du LQR pour les expériences sur le Crazyflie	124
5.2	Paramètres du filtre de Kalman et du LQR pour le drone Clover 4.2	129

Organisation de la thèse

L'objectif de cette partie est de fournir un guide au lecteur, en offrant un aperçu de l'organisation de la thèse et en résumant le contenu de chaque chapitre. Cette structure globale vise à faciliter la compréhension du travail de recherche effectué et à permettre au lecteur de naviguer efficacement à travers les différents aspects de l'étude.

Chapitre 1 : Ce chapitre introduit le contexte général de la thèse, en mettant en lumière les défis et les enjeux liés au développement ainsi qu'au contrôle des drones. Une présentation des différentes architectures de drones à tilt-rotors de la littérature. De même un passage en revue des travaux de recherche antérieurs dans le domaine du contrôle des drones. Finalement, les principales contributions de la thèse.

Chapitre 2 : Dans ce chapitre, le modèle dynamique du drone à inclinaison Xtilt est élaboré en utilisant le formalisme d'Euler-Lagrange. Les effets aérodynamiques et gyroscopiques sont pris en compte pour obtenir une représentation précise du comportement du drone. Les expressions de la poussée des rotors et de la distribution des forces de l'angle d'inclinaison sont explicitement dérivées. De plus, une analyse CFD (dynamique des fluides computationale) approfondie est effectuée pour modéliser le comportement aérodynamique du drone dans différents scénarios, en tenant compte de l'interaction entre les flux d'air générés par les hélices.

Chapitre 3 : Ce chapitre présente la conception des contrôleurs et les observateurs proposés. Différentes architectures de contrôle sont détaillées, utilisant à la fois des approches basées sur le LQR et des processus prédictifs basés modèle (MPC).

Chapitre 4 : Ce chapitre joue un rôle central dans la validation des différentes lois de commande proposées, ainsi que les diverses approches pour l'estimation des perturbations. La modélisation réalisée vise à atteindre le plus haut niveau de réalisme possible en incluant la dynamique complète du drone, les interactions avec le vent et les effets aérodynamiques.

Chapitre 5 : Ce chapitre revêt une importance cruciale dans la validation des concepts théoriques et des modèles développés précédemment. Son objectif est de confirmer l'efficacité des approches de contrôle proposées en utilisant des prototypes réels spécialement développés au cours de cette thèse.

Chapitre 6 : Le dernier chapitre conclut la thèse en récapitulant les principales conclusions et en mettant en avant les contributions majeures. Les implications pratiques et les domaines d'application potentiels sont discutés. Des pistes pour des travaux futurs et des améliorations possibles sont également proposées.

1

Introduction

Sommaire

1.1	Contextualisation et problématique	23
1.1.1	Perspective historique sur les drones	23
1.1.2	Les drones aujourd'hui	27
1.1.3	Revue de la littérature	30
1.1.4	Contexte de la thèse	35
1.1.5	Contributions principales de la thèse	36
1.2	Note de publications	39

Liste des abréviations

Symbole	Description
UAS	Unmanned Aerial Systems
SAAMPLE	Système Aérien Autonome interactif pour l'étude environnementale
CFD	Computational fluid dynamics
SMC	Sliding mode control
MPC	Model predictive control
NMPC	Nonlinear Model predictive control
LQR	Linear quadratic regulator
LKF	linear kalman filter
ES-LKF	extended state linear kalman filter
SRA	Sur-actionné
SSA	Sous-actionné
OM	Omnidirectionnel
EA	Entièrement actionné
N_m	Nombre de moteurs
N_s	Nombre de servomoteurs

INTRODUCTION— Le présent chapitre revêt une importance fondamentale dans la structure de cette thèse, étant donné qu'il pose les bases conceptuelles qui guideront l'ensemble de la recherche. Dans un premier temps, nous débutons par exposer le domaine de recherche qui sera exploré tout au long de ce travail, à savoir l'utilisation novatrice des drones pour l'échantillonnage de l'eau. Cette introduction vise également à formuler la problématique à laquelle cette thèse s'attellera, dans le but de contribuer à l'amélioration de l'état de l'art.

1.1 Contextualisation et problématique

1.1.1 Perspective historique sur les drones

Les précurseurs des drones

L'histoire des robots est riche en découvertes et en créations qui ont émergé bien avant l'ère moderne. Remontant à des siècles reculés, l'idée de concevoir des mécanismes autonomes destinés à diverses applications a été présente, jetant ainsi les bases de l'évolution qui allait mener aux robots contemporains. Les prémices de cette passionnante histoire peuvent être tracées jusqu'à des esprits brillants tels que Pythagore et Archimède.

La première machine volante autonome connue trouve son origine à l'antiquité et a été attribuée à Achytas de la ville de Tarente en Italie, mathématicien et philosophe, il est aussi célèbre pour ses contributions à la géométrie et la théorie des nombres [1].

Sa création, souvent nommée "pigeon mécanique", était une sorte d'oiseau en bois. Cette petite machine était mûe par la vapeur d'eau et pouvait voler sur de courtes distances.

Cependant, il est généralement reconnu par de nombreux historiens des sciences [2][3] que c'est au XIII^e siècle qu'un des premiers exemples concrets d'un robot voit le jour grâce à l'inventeur et ingénieur Al-Jazari. Ce créateur visionnaire a réalisé un exploit remarquable en concevant le prototype d'un mécanisme automatisé, bien avant l'ère de l'électronique et des ordinateurs. Parmi ses réalisations, un robot particulièrement remarquable qui utilise une méthode hydraulique a vu le jour : une boîte à musique programmable (figure 1.1a) qui pouvait jouer le tambour à différents rythmes ainsi qu'une horloge où un oiseau mécanique gazouille et un automate humanoïde frappe la cymbale (figure 1.1b). Ces créations, à la fois complexes et ingénieuses, avaient pour principal objectif d'apporter du divertissement.

Le robot d'Al-Jazari marque une étape cruciale dans l'histoire des machines autonomes. Non seulement il démontre la capacité humaine à imaginer et à construire des dispositifs mécaniques capables de fonctionner de manière autonome, mais il préfigure également les développements futurs dans le domaine de l'automatisation. Cette innovation audacieuse dans l'art de la mécanique reflète l'ingéniosité et la créativité de l'esprit humain à travers les âges.

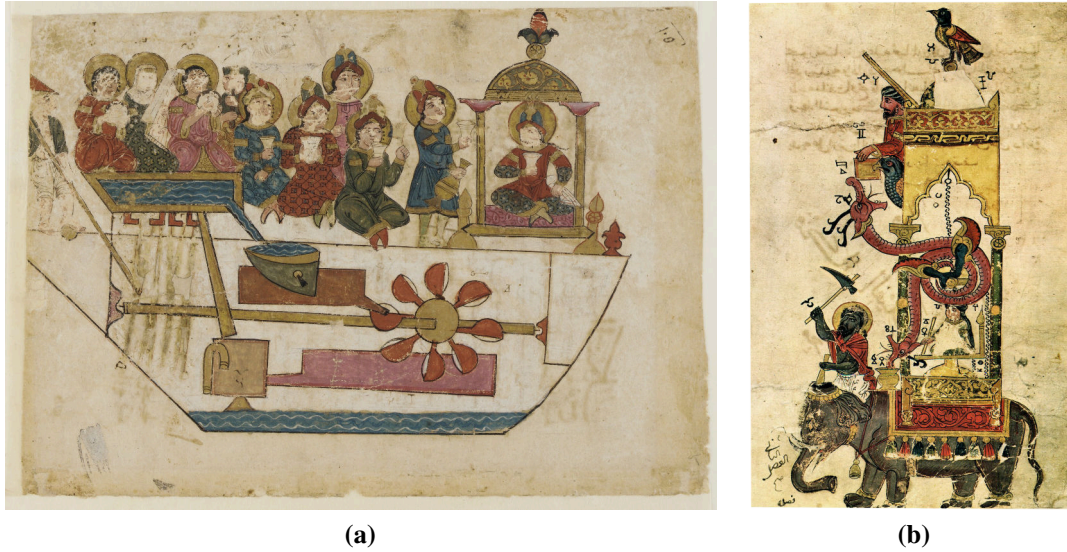


Figure 1.1: (a) Un robot musical, (b) Horloge éléphant tirés d'un exemplaire du livre d'al-Jazari sur les automates, Kitab fi ma'arif al-hiyal al-handasiya

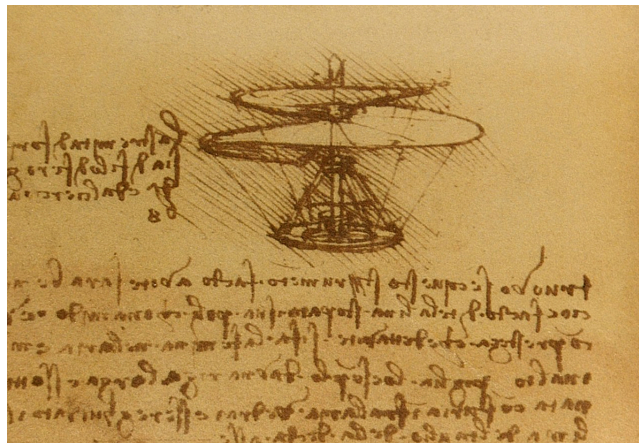


Figure 1.2: La "vis aérienne" conçue par Léonard de Vinci.

Léonard de Vinci, artiste, inventeur et scientifique de la Renaissance, a laissé son empreinte indélébile dans l'histoire de l'aviation. En 1483, il a élaboré un véhicule aérien connu sous le nom de "gyroscope aérien" ou encore "vis aérienne". Il se compose d'une partie centrale en forme de vis en la faisant tourner assez rapidement, elle était supposée générer assez de portance pour faire voler l'engin. Bien que cette machine n'ait jamais vu le jour de son vivant, néanmoins elle témoigne d'une étape importante dans l'exploration des possibilités de réaliser des vols. Certains spécialistes considèrent que la vis aérienne est l'ancêtre des hélicoptères [4] [5].

Les premiers vols

En 1783, les frères Montgolfier, Joseph-Michel et Jacques-Étienne Montgolfier, originaires d'Annonay, en France, ont conçu et fabriqué un ballon en papier et en toile, gonflé en utilisant l'air chaud provenant d'un feu situé en dessous. Le premier vol d'essai du ballon, baptisé le "Montgolfière" s'envola dans les airs à Annonay ne portant aucune charge humaine. Cependant, c'est le vol du 21 novembre 1783 qui a marqué l'histoire de l'aviation, le premier vol habité de l'histoire de l'humanité, Pilâtre de Rozier est le premier homme à l'avoir fait. Le succès de ce vol historique a captivé l'imagination du public et a suscité un grand intérêt pour les voyages aériens [6].

En 1804, George Cayley a conçu et testé un modèle réduit d'aéronef (figure 1.3) doté d'ailes fixes et d'une queue stabilisatrice, ses idées novatrices ont fondé les bases de l'aéronautique moderne. C'est qu'au XIXe siècle qu'un véritable essor de l'aviation a eu lieu, avec des inventeurs tels que Otto Lilienthal, qui a réalisé des vols en planeur avec succès et a contribué à une meilleure compréhension des principes de vol.

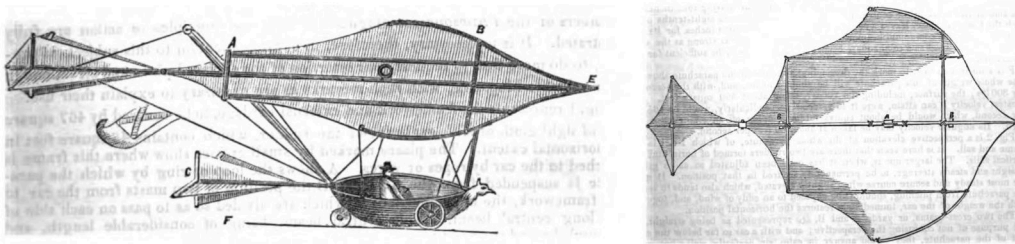


Figure 1.3: Planeur conçu par George Cayley.

En revanche, le premier vol motorisé soutenu et contrôlé, n'a eu lieu qu'au XXe siècle plus précisément en 1903, grâce à deux inventeurs de génie les frères Wright, Orville et Wilbur, cela a marqué un tournant important dans l'histoire de l'aviation.

Émergence des drones

En effet, tout au long de l'histoire de l'aviation, le moteur principale derrière le développement des avions a été la quête d'un moyen de transport pour les personnes comme pour les marchandises qui soit à la fois sûr et rapide. Les militaires ont rapidement reconnu les potentiels stratégiques et tactiques des aéronefs sans pilotes ou drones. La première guerre mondiale, qui débuta en 1914, a donné un bond en avant aux avancées dans l'aviation. Le "Kettering Bug" (figure 1.4) une torpille aérienne sans pilote, création de Charles Franklin Kettering, a réalisé son premier vol en 1918, marquant ainsi une époque d'avancées technologiques et d'expérimentations audacieuses, ouvrant la voie à des développements qui ont finalement abouti à l'aviation moderne.

La seconde guerre mondiale a connu une utilisation significative des drones marquant une nouvelle ère dans l'évolution de cette technologie. L'utilisation des drones a été multiple pour

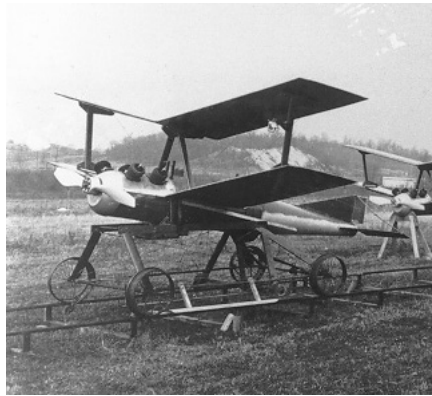


Figure 1.4: Kettering Bug, une torpille aérienne sans pilote.

diverses missions, notamment la reconnaissance, la diversion et même des attaques kamikazes. Prenant comme exemples, l'opération "Aphrodite" l'une des premières utilisations opérationnelles de drones, en réclamant des Boeing B-17 Flying Fortress et B-24 Liberator en fin de vie, ces derniers ont été convertis en drones chargés d'explosif, pilotés à distance par des équipages à bord d'autre avions, leurs missions étaient de détruire des cibles stratégiques allemandes comme les sites des V-1. Ces derniers étaient des missiles de croisière pilotés automatiquement développés par L'Allemagne nazie.



(a)



(b)

Figure 1.5: (a) Boeing B-17 à basse altitude, (b) V-1 en préparation pour un lancement

Pendant la Guerre froide, qui a duré de la fin de la Seconde Guerre Mondiale jusqu'à la dissolution de l'Union Soviétique en 1991, les drones ont continué à jouer un rôle important dans les opérations militaires de renseignement entre les deux superpuissances mondiales, L'Union Soviétique et les États-Unis. Les drones espions ont ainsi été développés et utilisés pour effectuer des missions de reconnaissance en territoire ennemi. De ce fait, les États-Unis ont développé le RP-71, également connu sous le nom de SD-1 Observer ou MQM-57 Falconer (figure 1.6), son premier vol a eu lieu en 1955. Ce dernier était équipé d'un système de pilotage automatique avec

une radio-commande de secours, ce qui lui permettait de fonctionner de manière autonome tout en offrant la possibilité d'une intervention humaine en cas de besoin. L'une des caractéristiques clés de ce drone était sa capacité à transporter des caméras pour la capture d'images à des fins de renseignement. Cela a permis aux militaires de recueillir des informations visuelles précieuses.



Figure 1.6: SD-1 Observer pour la reconnaissance tactique.

À l'ère moderne, les drones ont évolué pour devenir des outils incontournables et polyvalents utilisables dans une panoplie de domaines. Toujours dans le domaine militaire, les drones tels que le MQ-1 Predator et le MQ-9 Reaper (figure 1.7a - 1.7b) ont transformé les missions militaires en permettant des opérations d'attaques ciblées et de renseignement en temps réel.



Figure 1.7: (a) MQ-1 Predator, (b) MQ-9 Reaper en vol

1.1.2 Les drones aujourd'hui

L'évolution constante des technologies en termes de miniaturisation des capteurs, l'augmentation de puissance de calcul des microcontrôleurs et l'augmentation de la puissance des batteries ainsi que leurs autonomies contribuent largement à l'amélioration des performances des drones. Elles transforment ainsi notre capacité à effectuer des tâches comme l'exploration d'environnements difficilement accessibles (figure 1.8) ou dangereux pour l'être humain [7]–[9], la surveillance environnementale des feux de forêt et des catastrophes [10]. Un autre secteur aussi commence à explorer l'utilisation des drones, comme le géant Amazon (figure 1.9a) ou encore Google (figure

1.9b) avec sa filiale Wing en testant des systèmes de livraison aériens autonomes. En agriculture, les drones équipés de capteurs collectent des données pour la cartographie des cultures ou pour la pulvérisation de substances chimiques à des fins d'optimisation des pratiques agricoles. Dans le domaine de la recherche scientifique, les drones sont utilisés comme un moyen d'acquisition et de collecte de données scientifiques en embarquant différents capteurs, pour étudier par exemple des habitats marins ou les régions polaires. Équipés de caméra à haute résolution, les images aériennes des glaciers sont utilisées afin de créer des modèles 3D de la surface du glacier, permettant ainsi une surveillance des changements dans la topographie et les taux de fonte des glaciers. [11]–[13].



Figure 1.8: Décollage du drone Flyability dans la Grotte Eugenio Commission Boegan par ESA, Natalino Russo www.natalinorusso.it

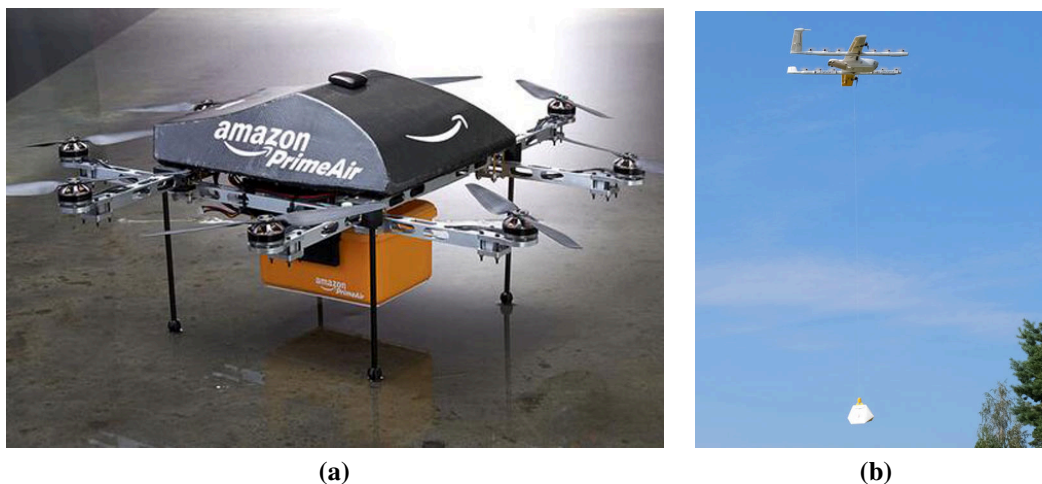


Figure 1.9: (a) Amazon Prime Air, (b) Google Wing en vol.

En demeurant dans le domaine scientifique, les drones trouvent également leur utilité dans le prélèvement d'échantillons d'eau, ayant pour objectif la surveillance de la qualité des ressources aquatiques ainsi que la cartographie des plans d'eau. Ils accomplissent cette tâche en effectuant divers relevés physico-chimiques [14]–[17], notamment la mesure de la température,



Figure 1.10: Pulvérisation d'un champ par un drone dji, <https://ag.dji.com/fr/about-us>

de la salinité et de la turbidité. Ces mesures offrent des indices quant aux concentrations en nutriments, procurant ainsi des informations précieuses quant à l'état des écosystèmes environnants. Cette approche s'avère aussi être un outil pour identifier les sources de pollution et préconiser des stratégies de prévention appropriées. En somme, cette technologie ouvre une réflexion profonde sur l'importance grandissante qu'elle revêt dans la compréhension et la sauvegarde des écosystèmes aquatiques.



Figure 1.11: Dji Matrice 300 avec l'échantillonneur d'eau de Nero <https://nerosampler.com/>

1.1.3 Revue de la littérature

L'avancement technologique des systèmes aériens sans pilote (UAS) a engendré un impact majeur au sein des secteurs industriels et scientifiques. Étant donné la constante évolution du champ des applications industrielles, il revêt une importance capitale que les UAS à rotors évoluent de manière concomitante afin de répondre aux missions les plus exigeantes. Dans cette optique, au sein de la communauté scientifique, émerge des initiatives visant à concevoir et développer de nouvelles configurations qui offrent un éventail élargi de capacités. Dans la suite, nous nous plongerons dans une revue de la littérature, explorant différentes approches et innovations émergentes dans le domaine des drones. Cette analyse exhaustive nous permettra de mieux comprendre les défis, les opportunités et les tendances actuelles. Pour mieux appréhender les différentes approches d'amélioration des drones à multirotors, il est essentiel de mettre en place une classification spécifique qui se focalise sur les capacités de mouvement et de positionnement dans l'espace, plutôt que sur le type de drone en lui-même (à voilure fixe ou tournante). Cette classification constituera le socle fondamental de notre analyse, nous autorisant ainsi à catégoriser de manière plus appropriée ces améliorations.

Classification des drones

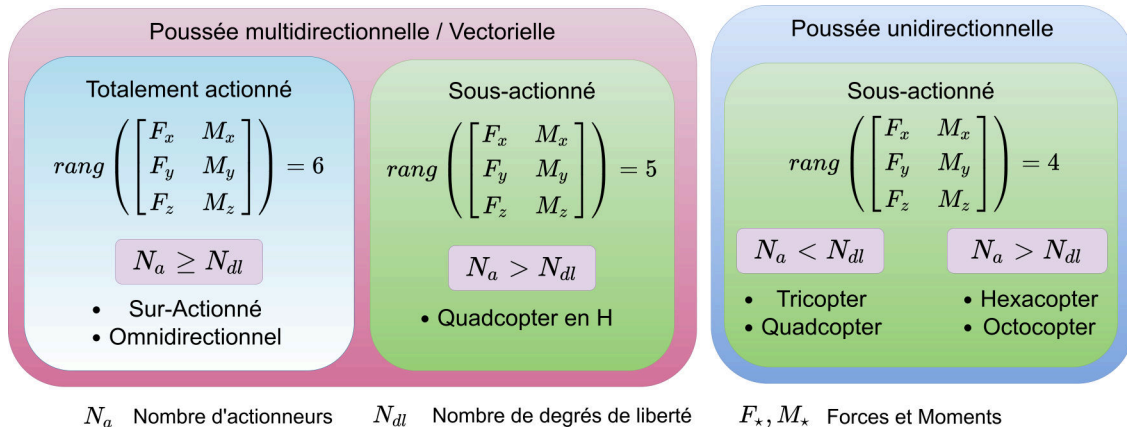


Figure 1.12: Classification des drones en fonction de leur nature : sous-actionnés, totalement actionnés.

La classification des drones multirotors repose fondamentalement sur le rapport entre le nombre d'actuateurs disponibles et le nombre de degrés de liberté inhérents à la plateforme (figure 1.12), tout en prenant en compte des nuances supplémentaires. Il est à noter que certaines configurations, bien qu'elles puissent disposer d'un nombre d'actionneurs supérieur, restent sous-actionnées. L'exemple de l'hexarotor présenté dans la figure 1.13b est pertinent à cet égard : bien que doté de six hélices, les vecteurs de poussée générés demeurent parallèles, limitant la capacité du système

à exercer des forces dans une seule direction. Cette particularité en fait un exemple de système sous-actionné. Les drones sous-actionnés, où le nombre d'actionneurs est inférieur aux DL, se heurtent à des contraintes lorsqu'ils cherchent à exécuter des mouvements complexes.



Figure 1.13: Exemples de drones sous-actionnés (a) Xdrone quadricoptère, (b) DJI F550 hexarotor

À l'opposé, les drones entièrement actionnés se distinguent par un équilibre entre le nombre d'actionneurs et de DL. Cette configuration confère au drone une agilité accrue ainsi qu'une palette de mouvements étendue. Par ailleurs, un cas particulier mérite d'être mentionné au sein de cette catégorie : les systèmes dits omnidirectionnels. Ces derniers offrent la capacité de maintenir un niveau minimal de force pour soutenir le poids tout en permettant au drone de s'orienter dans toutes les directions possibles [18] [19] (figure 1.14a - 1.14b).



Figure 1.14: Exemples de drones totalement actionnés (a) Tilt-Hex [19] [source image](#), (b) Hélicoptère coaxial [18] [source image](#)

Enfin, les drones sur-actionnés, qui disposent d'un surplus d'actionneurs par rapport aux DL, affichent une redondance de contrôle qui accroît leur capacité à tolérer les pannes et à maintenir leur stabilité en vol. Cette classification fournit une compréhension approfondie des atouts et

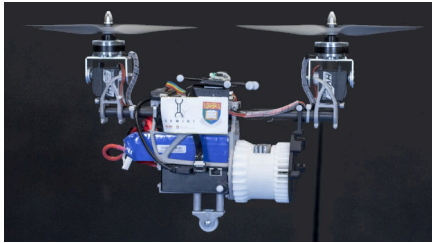
des limitations propres à chaque configuration de drone multirotors. Ces distinctions se révèlent cruciales pour guider le développement et l'optimisation des systèmes aériens sans pilote.

Architecture drones

Les configurations classiques des UAS se caractérisent par une sous-actionnabilité, c'est-à-dire que le nombre d'entrées de contrôle est inférieur au nombre de degrés de liberté (DL). Cette caractéristique présente des limitations dans diverses applications telles que l'échantillonnage de l'eau par drone et des tâches nécessitant une prise et une manipulation précises [20]. De telles situations requièrent une plateforme de manipulation dotée d'une certaine dextérité et flexibilité de mouvement. Ces problématiques se révèlent plus contraignantes pour les architectures conventionnelles à rotors fixes. Premièrement, dans ce type de drone, le contrôle direct de la position n'est pas possible, mais est plutôt réalisé à travers les mouvements de tangage et de roulis. Deuxièmement, l'utilisation d'une boucle de contrôle imbriquée pour gérer la position du drone restreint le temps de réponse ainsi que la capacité à rapidement atténuer les perturbations [21], [22]. Au cours des dernières années, plusieurs drones entièrement actionnés ont vu le jour, chacun étant adapté à des applications et des usages spécifiques. Diverses solutions ont été avancées, notamment la rotation des rotors en plein vol [23]–[29], ainsi que l'utilisation de rotors dotés d'un angle d'inclinaison fixe [30]–[34]. Ces travaux [35], [36] proposent une vue d'ensemble et une caractérisation générale des divers UAS multirotors. Le tableau 1.1 résume les divers types de drones multirotors mentionnés dans la littérature en fonction du nombre d'actionneurs et de leurs catégories.

Contrôleurs pour drones

Le transport de charges, l'interaction avec des objets, la collecte d'échantillons (comme le prélèvement d'eau), ainsi qu'une constante émergence de nouvelles applications, sont autant de tâches qui requièrent un contrôleur hautement performant et robuste pour les véhicules aériens autonomes, afin d'assurer un niveau élevé de stabilité face aux perturbations. En effet, malgré les progrès réalisés dans l'amélioration des performances opérationnelles des UAS actuels, ces systèmes demeurent sujets aux perturbations lors de l'exécution de leurs missions. Ces perturbations sont induites par une connaissance limitée du système, englobant à la fois les perturbations paramétriques et les perturbations externes dues à l'environnement. Parmi ces perturbations externes figurent les rafales de vent et les interactions environnementales en vol, telles que la collecte d'échantillons d'eau ou la manipulation/transport de charges. Il est donc impératif que le contrôleur soit capable de gérer efficacement ces perturbations pour éviter toute détérioration des performances du système. Dans la littérature, plusieurs lois de



(a) Gemini [37]



(b) Hélicoptère coaxial [18]



(c) Tricoptère en Y [38]



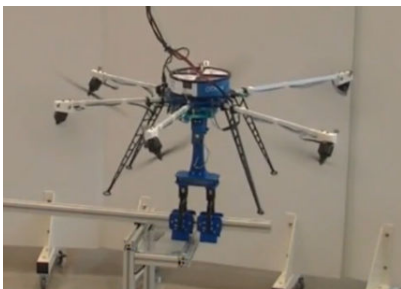
(d) Tricoptère en T Voliro [39]



(e) TiltDrone [40]



(f) Quadcoptère [41]



(g) OTHex [31]



(h) Hélicoptère Voliro [42]

Figure 1.15: Quelques illustrations des différents drones tiltrotor décrits dans la littérature.

	N_m	N_s	Classe	Description
Bicoptère	2	2	SSA	Le drone (figure 1.15a) est équipé de deux moteurs et deux servomoteurs, ce qui permet de générer trois types de couples : <ul style="list-style-type: none"> • Lacet en inclinant les propulseurs dans des directions opposées. • Roulis en inclinant les bras dans la même direction. • Tangage grâce à une variation de la poussée différentielle.
	2	0	EA	L'hélicoptère coaxial (figure 1.15b) est équipé de seulement deux actionneurs. Cependant, il est possible de diriger la poussée générée par les hélices en provoquant une réaction dynamique grâce à la modulation du couple appliqué par les moteurs. Cela permet d'effectuer une variation cyclique de l'inclinaison des pales du rotor.
Tricoptère	3	3	EA	Le drone (figure 1.15c) est équipé de trois moteurs et de trois servomoteurs disposés de manière à ce que les six degrés de liberté soient actionnés.
	6	4	SRA OM	Le drone (figure 1.15d) est équipé de cinq moteurs et de quatre servomoteurs disposés de manière à ce que les six degrés de liberté soient actionnés et la possibilité de réaliser un vol stationnaire dans n'importe quelle orientation.
Quadcoptère	4	2	EA	Le drone (figure 1.15e) est équipé de quatre moteurs et d'un mécanisme d'inclinaison de rotor biaxial, actionné par deux servomoteurs linéaires parallèle.
	4	4	SRA OM	Le drone (figure 1.15f) est équipé de quatre moteurs et quatre servomoteurs.
Héxacoptère	6	0	EA	Le drone (figure 1.15g) est équipé de six moteurs inclinés dans différentes directions, ce qui permet de générer une poussée multidirectionnelle en modifiant simplement les vitesses de rotation des moteurs.
	6	6	SRA OM	Le drone (figure 1.15h) est équipé de six moteurs et de six servomoteurs capables de maintenir une position stationnaire dans n'importe quelle orientation.

Table 1.1: Classification des drones multiroteurs de la littérature, N_m est le nombre de moteurs, N_s est le nombre de servomoteurs, SSA pour sous-actionné, EA pour entièrement actionné, SRA pour sur-actionné, OM omnidirectionnel

contrôle ont été déployées sur différentes configurations de drones. Par exemple, des contrôleurs PID ont été proposés dans les travaux [43]–[45]. Des méthodes de suivi de trajectoire basées sur le Backstepping sont employées dans [46] ou le contrôle non-linéaire basé énergie [47], tandis que le contrôle par mode glissant (2-SMC) est appliqué pour gérer à la fois la position et l'orientation dans [48]. Le régulateur linéaire quadratique LQG est utilisé pour réguler l'attitude des UAV dans [49] ainsi que le contrôle à mode glissant (SMC) [50], [51] ou la commande prédictive (MPC) [52]–[56].

Lorsque les drones interagissent avec l'environnement extérieur, il peut être nécessaire d'estimer les perturbations externes, notamment lorsque la tâche exige l'application d'un certain couple ou d'une certaine force sur un autre objet, ou pour évaluer la direction et l'intensité du vent. Ces forces en jeu peuvent être mesurées, soit par des capteurs de force ou de couple intégrés

au drone, mais cela ajoute du poids, soit par un observateur utilisant un modèle mathématique du drone qui prend en compte les perturbations extérieures.

En réalité, il existe plusieurs types d'observateurs, par exemple les observateurs basés sur le filtre de Kalman [57], [58], [59], les observateurs basés sur la quantité de mouvement (momentum-based observers) [60], ainsi que les observateurs hybrides [61]. Dans [62], les performances de divers observateurs sont évaluées par simulation dans diverses conditions. Selon cette étude, le filtre de Kalman étendu présente la meilleure robustesse au bruit et la plus grande précision d'estimation. Les approches dans la littérature concernant l'utilisation de LQR et de LQG sur les drones quadcoptères reposent souvent sur des modèles linéarisés, tels que cités dans [63]–[65] et [49]. En ce qui concerne les contrôleurs optimaux robustes, certains auteurs ont incorporé l'action intégrale, comme dans [66]. D'autre part, l'estimation des entrées inconnues repose souvent sur le cadre du filtre de Kalman, comme observé dans [57], [58]. Il convient toutefois de noter que ces approches n'ont été validées que sur le plan de la simulation. Le LQR est connu pour offrir une marge de gain et de phase significative, ce qui en fait un choix séduisant pour les systèmes linéaires sans bruit. Cependant, en conditions réelles, où le bruit et les incertitudes paramétriques sont prévalents, cette méthode montre ses limites. Pour pallier cette problématique, le LQR est associé à un estimateur quadratique linéaire (LQE), plus couramment appelé filtre de Kalman linéaire (LKF), formant ainsi la commande quadratique linéaire gaussienne (LQG). Toutefois, malgré ses avantages, le LQG demeure plus vulnérable aux perturbations externes. En effet, il n'y a pas de marge de stabilité garantie, comme exposé dans [67].

1.1.4 Contexte de la thèse

La thèse s'inscrit dans le cadre du projet SAAMPLE, une initiative soutenue par la région Nouvelle-Aquitaine dans le cadre de sa stratégie ambitieuse pour l'eau. Cette stratégie vise à améliorer à la fois la qualité bactériologique et les propriétés physico-chimiques de l'eau, ainsi qu'à surveiller attentivement son utilisation et sa pollution par les entreprises. Le projet système aérien autonome interactif pour l'étude environnementale SAAMPLE vise particulièrement le prélèvement pour l'analyse de la qualité de l'eau dans des environnements naturels qualifiés de "non structurés". Plusieurs acteurs contribuent à ce projet, dont le laboratoire XLIM UMR CNRS 7252, qui a pour mission le développement d'une plateforme aérienne autonome destinée à effectuer des prélèvements d'eau ou des mesures ne nécessitant pas nécessairement un prélèvement.

En effet, l'utilisation de drones s'avère être une solution pertinente pour réaliser des prélèvements dans des zones considérées comme "non structurées", et ce pour plusieurs raisons. Il convient de noter que pour effectuer des prélèvements en vue d'analyses, la méthode traditionnelle implique des déplacements à pied le long de cours d'eau peu profonds (figure 1.16), ou en bateau pour



Figure 1.16: Prélèvement d'échantillons d'eau dans une rivière

les cours d'eau plus profonds. Cependant, les mouvements des pieds ou du bateau mélangent les eaux en surface avec celles des couches plus profondes, ce qui altère la fidélité de la mesure réalisée, particulièrement si celle-ci dépend de la profondeur.

1.1.5 Contributions principales de la thèse



Figure 1.17: Les deux versions du Xtilt développées dans les travaux de thèse, (a) la version 1 avec une carte Nvidia Jetson Nano, (b) la version 2 avec un contrôleur basée sur la Raspberry pi4. La distinction dans le positionnement des hélices doit également être relevée.

Le travail de cette thèse tire sa motivation et son inspiration des initiatives mentionnées précédemment. Nous proposons ainsi le développement d'un drone à quatre rotors pivotants (figure 1.17), avec pour objectif ultime d'atteindre une maîtrise totale et réactive basée sur la vectorisation de la poussée. La configuration présentée ici se démarque des approches connexes par sa capacité à suivre des trajectoires tant non holonomiques qu'holonomiques, en fonction du profil de vol et de l'usage. Cette configuration s'appuie sur un schéma en forme de H [68] [69]. Cette disposition nous offre la possibilité, d'une part, d'orienter la totalité de la poussée le

long d'un seul axe et, d'autre part, d'affronter des conditions venteuses à différentes inclinaisons, impactant la portance du drone en présence d'ailes. Un drone quadrirotor à bras inclinable entièrement opérationnel a été conçu, couvrant à la fois les aspects mécaniques (structure et mécanisme d'inclinaison) et l'architecture embarquée (système de commande de vol). La plate-forme expérimentale à propulseur pivotant est illustrée dans la (figure 1.17).

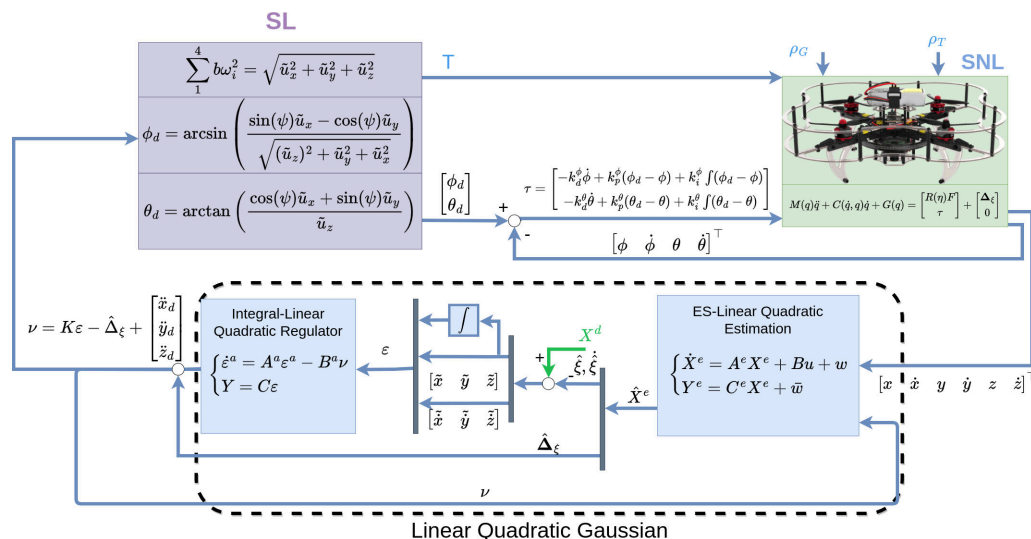


Figure 1.18: Description du ES-iLQG (Extended State Integral Linear Quadratic Gaussian).

Ce travail vise aussi à résoudre les problèmes de robustesse associés au LQG, tout en préservant les avantages du LQR et du LKF. Pour atteindre cet objectif, nous adoptons une approche en deux volets (figure 1.18):

- Nous renforçons le LQR en introduisant une composante intégrale, ce qui permet de mieux faire face aux incertitudes inhérentes au système.
- Nous utilisons une estimation de perturbation exogène obtenue par un ES-LKF (estimateur de Kalman) pour mettre à jour la sortie intégrale du LQR. Cette mise à jour vise à compenser les perturbations qui peuvent survenir pendant le fonctionnement.

L'objectif général de cette recherche est de présenter une stratégie de contrôle (figure 1.19) qui ne vise pas seulement une navigation optimale, mais également un rejet optimal des perturbations. Étant donné que la formulation du contrôle prédictif basé sur le modèle (MPC) exige un modèle précis pour satisfaire de manière optimale le profil de performance, deux schémas robustes sont intégrés dans l'approche MPC :

- Un état augmenté ou auxiliaire est incorporé pour élever l'ordre du modèle nominal. Ce modèle augmente intrinsèquement la robustesse.
- Un contrôle à mode glissant est adopté pour contrer les incertitudes bornées et appariées.

La structure du modèle utilisée pour résoudre le problème MPC est optimisée grâce à l'intégration de l'estimation des perturbations fournies par un estimateur de Kalman (ES-LKF). La combinaison

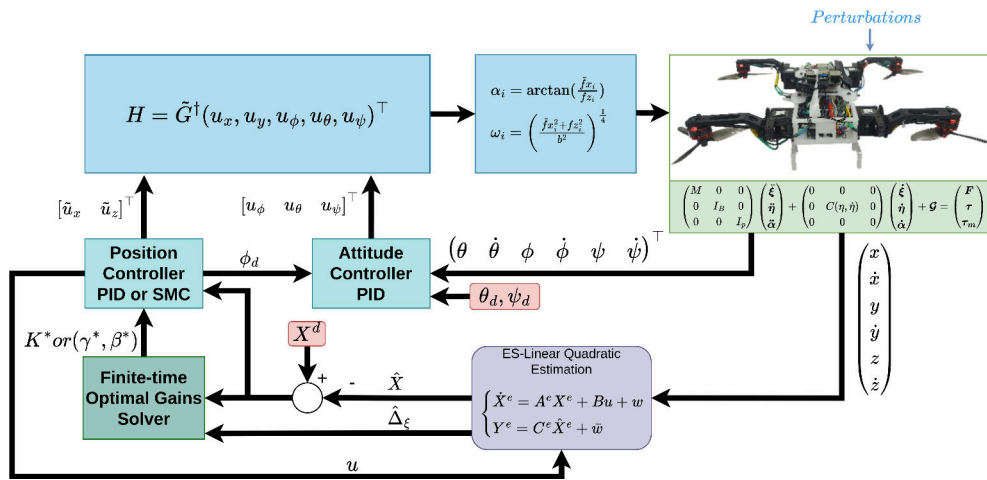


Figure 1.19: Description du contrôleur basé sur le modèle prédictif.

de ces éléments permet d'ajuster les gains du contrôleur de manière optimale en fonction du modèle mis à jour à chaque étape de temps.

Les principales contributions de cette étude sont résumées dans la liste ci-dessous :

- Conception de deux versions de la plateforme expérimentale (Xtilt), couvrant l'ensemble du processus, de la conception mécanique initiale au choix de l'architecture du système embarqué, en passant par le développement logiciel et la programmation, jusqu'à la réalisation des prototypes en fabrication additive.
- Le modèle dynamique complet du drone à bras pivotant (Xtilt) est formulé à l'aide du formalisme d'Euler-Lagrange, qui offre une représentation mathématique précise des mouvements, des forces et moments en jeu. Ce modèle prend en compte les effets aérodynamiques et gyroscopiques qui influent sur le comportement du drone.
- Une étude détaillée des actionneurs est entreprise afin de déterminer les distributions de forces pour différentes architectures de drones. Cette étude vise à comprendre en profondeur comment les actionneurs moteurs et hélices, génèrent des forces et des moments qui influencent le comportement de vol du drone.
- Une analyse approfondie de la dynamique des fluides numérique (CFD) est proposée pour modéliser le comportement aérodynamique à la fois d'un quadcoptère classique et d'un quadricoptère à propulseurs pivotants. Cette étude se concentre également sur l'analyse aérodynamique des hélices et sur l'interaction complexe entre les différents flux d'air générés par les hélices. L'objectif principal de cette analyse est de dériver des conclusions opérationnelles significatives pour les deux configurations de drones.
- Nous proposons plusieurs architectures de contrôle robustes conçues pour être pilotées de manière optimale, soit par un Régulateur Linéaire Quadratique (LQR), soit par un processus prédictif basé sur un modèle (MPC). Cette approche prédictive implique la

création d'une structure comprenant un modèle nominal ainsi qu'un terme additif pour tenir compte des perturbations inconnues, mais bornées. Ce modèle est continuellement mis à jour en utilisant une estimation du vecteur d'état étendue. Deux approches d'estimation des perturbations sont explorées, le Filtre de Kalman à état étendue et le Super Twisting Observer (STO).

- L'efficacité des lois de contrôle proposées est évaluée à travers une étude numérique détaillée en utilisant des environnements hautement réalistes. Pour ce faire, plusieurs plateformes logicielles sont utilisées, notamment Matlab Simulink, ROS (Robot Operating System), ROS2 et Gazebo. Ces plateformes permettent de modéliser les drones avec un haut degré de fidélité afin de reproduire au mieux les conditions réelles. L'approche d'évaluation comprend la création de modèles de drones précis, prenant en compte les caractéristiques géométriques, dynamiques et aérodynamiques spécifiques à chaque architecture de drone étudiée. Ces modèles sont ensuite intégrés dans les environnements de simulation pour refléter les conditions réelles dans lesquelles les drones opéreront. Une série d'expériences est menée en variant les scénarios pour inclure différentes perturbations et ajustements des paramètres du contrôleur. Ces expériences visent à évaluer la capacité des lois de contrôle à maintenir la stabilité, la précision et la réactivité du drone face à des conditions changeantes et perturbées.
- Une description détaillée des plateformes expérimentales est fournie pour éclairer le processus d'évaluation pratique. L'étape expérimentale a validé la pertinence des lois de commande développées et a mis en évidence la flexibilité du vol pour atteindre l'objectif de suivi, en passant d'un vol classique à un vol stationnaire à attitude non nulle. Les expérimentations ont été conçues pour mettre en évidence la performance et la robustesse des lois de commande dans des scénarios réels. Les drones ont été soumis à des conditions variées.

1.2 Note de publications

Les travaux effectués au cours de cette thèse ont conduit à des résultats, dont certains ont été présentés à des conférences internationales. Deux publications ont été acceptées pour présentation aux conférences suivante IEEE International Conference on Unmanned Aircraft Systems (ICUAS) 2022 et International Conference on Automation Science and Engineering (CASE) 2022. De plus, un journal a été soumis à la revue IEEE Transactions on Aerospace and Electronic Systems (TAES), actuellement en cours d'examen.

2

Modèle dynamique du Xtilt

Sommaire

2.1	Les architectures de drones étudiées	42
2.1.1	Quadrirotor à propulseurs fixes	42
2.1.2	Quadrirotor à propulseurs pivotants	42
2.2	Définition des différents repères	43
2.3	Dynamique du drone	45
2.3.1	Paramètres cinématiques	45
2.3.2	Paramètres physiques	47
2.3.3	Équations de Lagrange	48
2.3.4	Bilan des forces et des couples	49
2.4	Limitation du torseur des actions mécaniques	55
2.4.1	Cas des bras pivotants Xtilt	58
2.4.2	Quadrirotor à bras fixes	62
2.5	Mécanique des fluides numérique	62
2.5.1	Modèle CFD	63
2.6	Conclusion	77

Liste des variables

Symbole	Description
\mathcal{O}	Repère inertiel supposé galiléen
\mathcal{B}	Repère mobile
\mathcal{A}	Repère aérodynamique
\mathcal{P}_i	Repère associé à la i -ème hélice
$\xi \in \mathbb{R}^3$	Position du centre de masse dans le repère \mathcal{O}
$\eta \in \mathbb{R}^3$	Angles d'Euler orientation du repère \mathcal{B} par rapport au repère \mathcal{O}
$\Omega \in \mathbb{R}^3$	Vitesse de rotation instantanée entre \mathcal{O} et \mathcal{B} décrite en \mathcal{B}
$W_\eta \in \mathbb{R}^{3 \times 3}$	Wronskien permet la transformation de $\dot{\eta}$ en Ω
$\mathcal{D}(t)$	Vitesse linéaire exprimé dans \mathcal{B}
m	masse
$I_\star \in \mathbb{R}^3$	Matrice des inerties du drone
$q \in \mathbb{R}^3$	Vecteur des position généralisées
\mathcal{L}	Lagrangien
\mathcal{T}	Énergie cinétique totale
\mathcal{V}	Énergie potentielle totale
Q	Forces non conservatives et couples
$M \in \mathbb{R}^{3 \times 3}$	Matrice des masses
$\alpha_i \in \mathbb{R}$	Angle d'inclinaison du i -ème bras
τ_\star	Pour les couples
$\mathcal{F}_\star, f_\star$	Pour les forces
d, b	respectivement coefficient de traînée et de portance dépendent d'une hélice
$l_i \in \mathbb{R}^3$	est la distance du i -ème rotor par rapport au centre de masse
C_d	Coefficient de traînée du drone
C_l	Coefficient de portance du drone
C_s	Coefficient de portance latérale du drone
C_ϕ	Coefficient de couple aérodynamique axe roulis du drone
C_θ	Coefficient de couple aérodynamique axe tangage du drone
C_ψ	Coefficient de couple aérodynamique axe lacet du drone
H^\star	Vecteur des entrées de commande du système exprimé dans le référentiel \star
$G \in \mathbb{R}^{6 \times 8}$	Matrice d'allocation commandes

MODÉLISATION— Dans ce chapitre nous allons établir un modèle mathématique pour chacun des systèmes proposés. Dans un premier temps, cette modélisation comportera une description complète de la cinématique ainsi que de la dynamique du drone. Dans un second temps, un bilan des forces et des couples, le plus exhaustif possible, est établi qui inclut les effets aérodynamiques et leurs modélisations numériques CFD (computational fluid dynamics). L'intérêt est de construire des modèles de simulation complets des différents drones utilisés. Commençant dans un premier temps par la modélisation d'un quadrirotor à propulseurs fixes et concluant finalement par un quadrirotor à propulseurs pivotants. Cette étape est nécessaire, puisqu'elle permet à la fois d'établir des lois de commandes basées modèles et permet une première validation en simulation.

2.1 Les architectures de drones étudiées

2.1.1 Quadrirotor à propulseurs fixes

Les quadrirotors à bras fixes, de conception mécanique très basique sont dotés d'une grande dextérité et capacité à se mouvoir rapidement dans l'espace. Cependant, par leur nature, ils sont des systèmes dits sous-actionnés, dont le nombre de degrés de liberté est supérieur au nombre d'entrées de commande. Pour ce type de système, la poussée est unidirectionnelle. En effet, les quatre actionneurs étant coplanaires et orientés de la même manière, le mouvement translationnel horizontal est dû au changement d'orientation figure (2.1).

2.1.2 Quadrirotor à propulseurs pivotants

De conception mécanique plus complexe que précédemment, ces derniers, dans la plupart des cas, par le biais de servomoteurs offrent la possibilité d'orienter la poussée générée par chaque moteur, ce qui a pour effet d'augmenter le degré d'actionnement. Il est par exemple possible

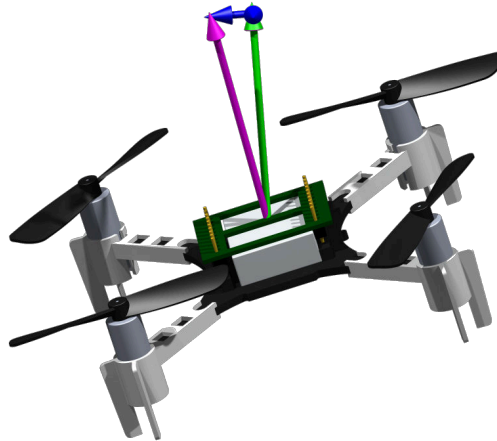


Figure 2.1: Illustration des différentes forces, les quatre moteurs génèrent une poussée totale représentée par le vecteur rose, le vecteur vert représente la composante verticale de cette force et le vecteur bleu la composante horizontale (drone Crazyflie 2.1).

avec ce type de système de faire un vol stationnaire non horizontal ou mouvement horizontal sans changer d'orientation figure (2.2). En effet, cela a pour conséquence d'augmenter le nombre d'entrées de commande. Cependant, l'augmentation du nombre d'entrées de commande ne signifie par forcément que le système soit totalement actionné, on en discutera de cela avec plus de détails dans la suite.

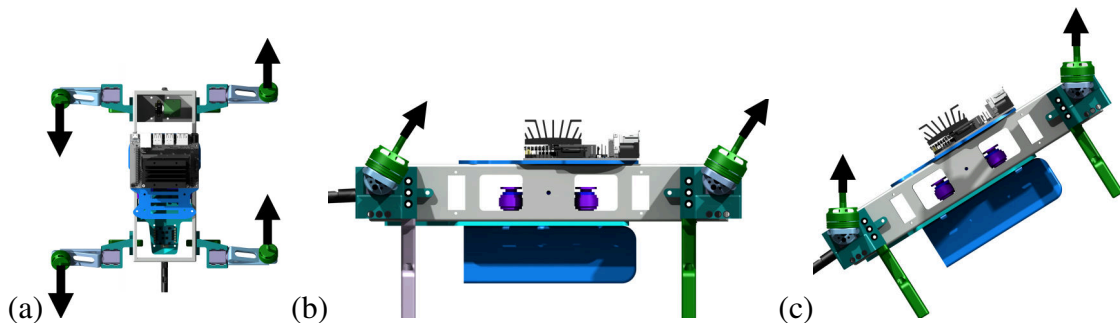


Figure 2.2: Description des différentes possibilités d'enveloppe de contrôle: (a) Contrôle différentiel du lacet (b) mouvement horizontal sans changer d'orientation (c) vol stationnaire non horizontal.

2.2 Définition des différents repères

Afin de décrire le mouvement du drone dans l'espace, on définit différents repères :

- O est le repère inertiel supposé galiléen, avec comme point d'origine le point O fixé au sol et a une base $\{e_x, e_y, e_z\}$ dont e_z pointe dans le sens inverse du centre de la Terre.

- \mathcal{B} est le repère mobile attaché au corps du drone, dont l'origine \mathcal{G} est le centre de masse de l'engin. le repère \mathcal{B} est lié à la base $\{e_1, e_2, e_3\}$, e_1 est l'axe de roulis et pointe vers l'avant, e_2 est l'axe de tangage et pointe vers la gauche, e_3 est l'axe de lacet et pointe vers le haut.
- \mathcal{A} est le repère aérodynamique lié au châssis du drone. Son origine O_a coïncide avec celle du repère \mathcal{B} , son orientation par rapport à ce dernier est définie par la matrice de rotation $R^{\mathcal{A}\mathcal{B}}(\iota, \nu) \in \mathbb{R}^{3 \times 3}$, ι est l'angle d'attaque, il décrit l'angle entre segments liant le bord d'attaque au bord de fuite et la direction du flux d'air et ν est l'angle de dérive qui correspond à l'angle entre le vecteur vitesse air et le plan $\{e_2, e_3\}$ du drone.

$$R^{\mathcal{A}\mathcal{B}}(\iota, \nu) = \begin{pmatrix} \cos(\iota) \cos(\nu) & -\sin(\nu) & \sin(\iota) \cos(\nu) \\ \cos(\iota) \sin(\nu) & \cos(\nu) & \sin(\iota) \sin(\nu) \\ -\sin(\iota) & 0 & \cos(\iota) \end{pmatrix} \quad (2.1)$$

- \mathcal{P}_i est i-ème repère associé à une hélice dont l'origine O_p est le centre de l'hélice associé à une base $\{e_{a_i}, e_{b_i}, e_{c_i}\}$. L'orientation de \mathcal{P}_i par rapport à \mathcal{B} est définie par la matrice de rotation $R^{\mathcal{P}_i\mathcal{B}}(\alpha_i, \beta_i) \in \mathbb{R}^{3 \times 3}$.
 - Une rotation d'angle α_i autour de l'axe e_{b_i} représentée par la matrice suivante:

$$R_{b_i}(\alpha_i) = \begin{pmatrix} \cos(\alpha_i) & 0 & \sin(\alpha_i) \\ 0 & 1 & 0 \\ -\sin(\alpha_i) & 0 & \cos(\alpha_i) \end{pmatrix} \quad (2.2)$$

- Une rotation d'angle β_i autour de l'axe e_{c_i} représentée par la matrice suivante:

$$R_{e_{c_i}}(\beta_i) = \begin{pmatrix} \cos(\beta_i) & -\sin(\beta_i) & 0 \\ \sin(\beta_i) & \cos(\beta_i) & 0 \\ 0 & 0 & 1 \end{pmatrix} \quad (2.3)$$

Ainsi l'orientation de \mathcal{P}_i par rapport à \mathcal{B} est défini par

$$\begin{aligned} R^{\mathcal{P}_i\mathcal{B}}(\alpha_i, \beta_i) &= R_{e_{c_i}}(\beta_i) R_{b_i}(\alpha_i) \\ &= \begin{pmatrix} \cos(\alpha_i) \cos(\beta_i) & -\sin(\beta_i) & \sin(\alpha_i) \cos(\beta_i) \\ \cos(\alpha_i) \sin(\beta_i) & \cos(\beta_i) & \sin(\alpha_i) \sin(\beta_i) \\ -\sin(\alpha_i) & 0 & \cos(\alpha_i) \end{pmatrix} \quad i \in \{1, 2, 3, 4\} \end{aligned} \quad (2.4)$$

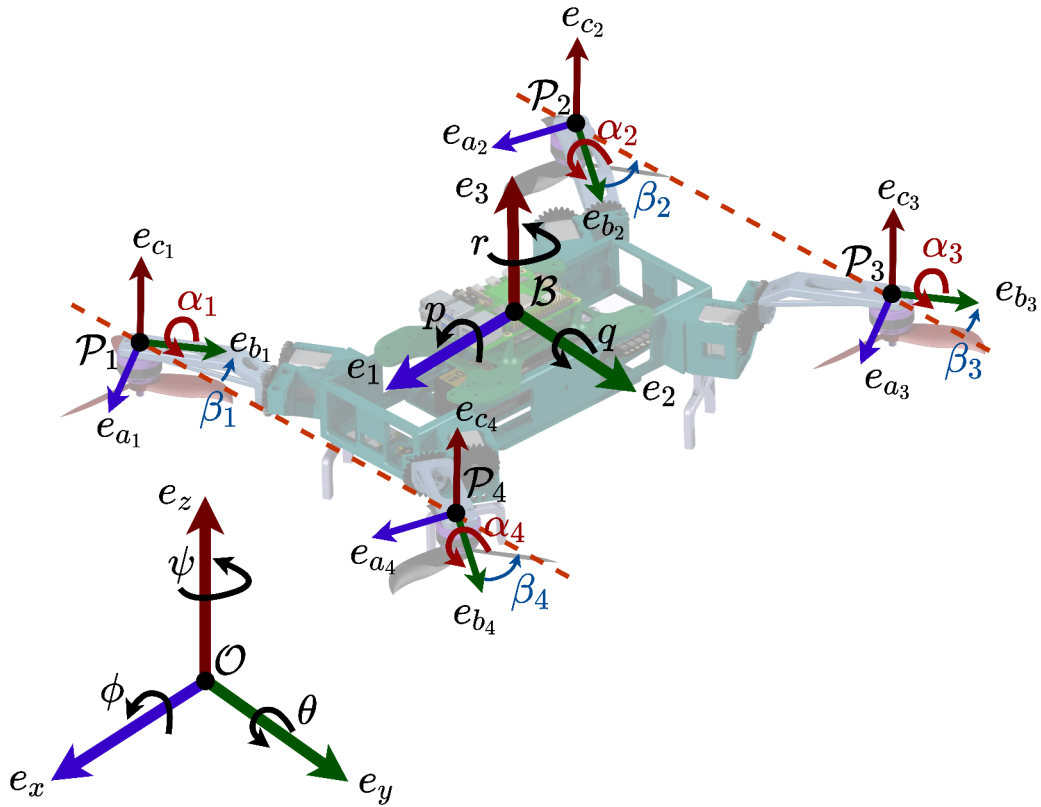


Figure 2.3: Description des différents repères considérés pour les deux structures étudiées

2.3 Dynamique du drone

Avant d'établir le modèle dynamique du drone, il est nécessaire de connaître et de décrire l'état du système.

2.3.1 Paramètres cinématiques

L'état du système peut être décrit par quatre vecteurs qui sont l'orientation et la vitesse angulaire du véhicule ainsi que la position et la vitesse du centre de masse. Ces vecteurs appartiennent à l'ensemble \mathbb{R}^3 et sont définis de la manière suivante :

- $\xi(t) = \begin{pmatrix} x & y & z \end{pmatrix}^\top$ est la position du centre de masse exprimée dans le repère O .
- $\alpha_i = \begin{pmatrix} 0 & \alpha_i & 0 \end{pmatrix}^\top$, est l'angle d'inclinaison du i -ème bras, exprimé dans le repère de l'hélice.
- $\eta = \begin{pmatrix} \phi & \theta & \psi \end{pmatrix}^\top$ constitué des angles d'Euler, il représente l'orientation du repère B par rapport au référentiel O

Le passage du référentiel O vers le repère B est réalisé par le biais de trois rotations.

– une rotation d'angle ϕ (roulis) autour de l'axe e_x représentée par la matrice suivante,

$$R_x(\phi) = \begin{pmatrix} 1 & 0 & 0 \\ 0 & \cos(\phi) & -\sin(\phi) \\ 0 & \sin(\phi) & \cos(\phi) \end{pmatrix} \quad (2.5)$$

– une rotation d'angle θ (tangage) autour de l'axe e_y représentée par la matrice suivante,

$$R_y(\theta) = \begin{pmatrix} \cos(\theta) & 0 & \sin(\theta) \\ 0 & 1 & 0 \\ -\sin(\theta) & 0 & \cos(\theta) \end{pmatrix} \quad (2.6)$$

– une rotation d'angle ψ (lacet) autour de l'axe e_z représentée par la matrice suivante,

$$R_z(\psi) = \begin{pmatrix} \cos(\psi) & -\sin(\psi) & 0 \\ \sin(\psi) & \cos(\psi) & 0 \\ 0 & 0 & 1 \end{pmatrix} \quad (2.7)$$

La matrice de rotation qui transforme \mathcal{B} à \mathcal{O} est alors définie par,

$$\begin{aligned} R^{\mathcal{O}\mathcal{B}}(\boldsymbol{\eta}) &= R_z(\psi)R_y(\theta)R_x(\phi) \\ &= \begin{pmatrix} \cos(\psi)\cos(\theta) & \cos(\psi)\sin(\phi)\sin(\theta) - \cos(\phi)\sin(\psi) & \sin(\psi)\sin(\phi) + \cos(\psi)\sin(\theta)\cos(\phi) \\ \cos(\theta)\sin(\psi) & \cos(\phi)\cos(\psi) + \sin(\phi)\sin(\psi)\sin(\theta) & \cos(\phi)\sin(\psi)\sin(\theta) - \cos(\psi)\sin(\phi) \\ -\sin(\theta) & \cos(\theta)\sin(\phi) & \cos(\theta)\cos(\phi) \end{pmatrix} \end{aligned} \quad (2.8)$$

• $\boldsymbol{\Omega}(t) = \begin{pmatrix} p & q & r \end{pmatrix}^\top$ est le vecteur de vitesse de rotation instantanée entre \mathcal{O} et \mathcal{B} décrit dans \mathcal{B}

$$\begin{aligned} \begin{pmatrix} p \\ q \\ r \end{pmatrix} &= \begin{pmatrix} \dot{\phi} \\ 0 \\ 0 \end{pmatrix} + R_x(\phi) \begin{pmatrix} 0 \\ \dot{\theta} \\ 0 \end{pmatrix} + R_x(\phi)R_y(\theta) \begin{pmatrix} 0 \\ 0 \\ \dot{\psi} \end{pmatrix} \\ &= \begin{pmatrix} 1 & 0 & \sin(\theta) \\ 0 & \cos(\phi) & -\cos(\theta)\sin(\phi) \\ 0 & \sin(\phi) & \cos(\phi)\cos(\theta) \end{pmatrix} \dot{\boldsymbol{\eta}}(t) = W_\eta \dot{\boldsymbol{\eta}}(t) \end{aligned} \quad (2.9)$$

W_η est la matrice Wronskienne, elle permet de transformer $\dot{\boldsymbol{\eta}}$ en $\boldsymbol{\Omega}$, cette dernière est défini pour $\theta \in \left] -\frac{\pi}{2}, \frac{\pi}{2} \right[$

- $\mathfrak{D}(t) = \begin{pmatrix} a & b & c \end{pmatrix}^\top$ correspond au vecteur de vitesse linéaire du centre de masse exprimé dans \mathcal{B} et par rapport à \mathcal{O}

$$\mathfrak{D}(t) = \begin{pmatrix} a \\ b \\ c \end{pmatrix} = (R^{O\mathcal{B}}(\boldsymbol{\eta}))^\top \dot{\boldsymbol{\xi}}(t) \quad (2.10)$$

2.3.2 Paramètres physiques

En matière de paramètres physiques, le quadrirotor à bras fixe et à bras pivotant se partage quelques caractéristiques cinétiques, comme la masse, le centre de gravité et la matrice d'inertie. Dans le premier cas, le drone est considéré comme un seul et unique corps rigide. Cependant pour la deuxième structure, ce n'est pas le cas, les quatre bras sont considérés comme des solides indépendants du châssis principal.

On notera :

- m la masse totale du véhicule
- \mathcal{G} le centre de gravité (CG) du véhicule
- \mathcal{G}_i le centre de gravité correspondant au i -ème ensemble qui comporte moteur, hélice et bras.
- $I_{\mathcal{B}}$ la matrice d'inertie du drone exprimée au centre de gravité G et dans le référentiel \mathcal{B} . On suppose que pour les deux systèmes étudiés, on n'a aucune asymétrie massique ou géométrique, sera considérée négligeable, ce qui implique que la matrice d'inertie est approximée par une matrice diagonale définie par :

$$I_{\mathcal{B}} = \begin{pmatrix} I_{xx} & 0 & 0 \\ 0 & I_{yy} & 0 \\ 0 & 0 & I_{zz} \end{pmatrix}_{\mathcal{B}} \quad (2.11)$$

- $I_{\mathcal{P}_i}^{\alpha_i}$ la matrice d'inertie du i -ème ensemble bras, moteur et hélice exprimée au centre de gravité G_i et dans le référentiel \mathcal{P}_i . De la même manière que précédemment, on suppose que l'ensemble est mécaniquement très bien équilibré ce qui laisse à supposer que la matrice $I_{\mathcal{P}_i}^{\alpha_i}$ est une matrice diagonale, définie par :

$$I_{\mathcal{P}_i}^{\alpha_i} = \begin{pmatrix} I_{xx}^{\alpha_i} & 0 & 0 \\ 0 & I_{yy}^{\alpha_i} & 0 \\ 0 & 0 & I_{zz}^{\alpha_i} \end{pmatrix}_{\mathcal{P}_i} \quad (2.12)$$

- $I_i^{\alpha_i} \in \mathbb{R}$ est l'inertie de l' i -ème ensemble {moteur et hélice} dans le repère \mathcal{P}_i portée par e_{ci} .

2.3.3 Équations de Lagrange

Le Xtilt est constitué de plusieurs solides indéformables, afin d'établir sa dynamique, il est judicieux d'utiliser le formalisme Euler-Lagrange. En effet, comme le lagrangien $\mathcal{L}(\mathbf{q}, \dot{\mathbf{q}}, t)$ est une fonction des positions généralisées \mathbf{q} , vitesses généralisées $\dot{\mathbf{q}}$ et du temps, cela simplifie la description dynamique d'un système constitué de plusieurs solides.

$$\mathcal{L}(\mathbf{q}, \dot{\mathbf{q}}, t) = \mathcal{T}(\mathbf{q}, \dot{\mathbf{q}}, t) - \mathcal{V}(\mathbf{q}, t) \quad (2.13)$$

Le Lagrangien est défini par la différence entre l'énergie cinétique totale du système $\mathcal{T}(\mathbf{q}, \dot{\mathbf{q}}, t)$ et l'énergie potentielle du système $\mathcal{V}(\mathbf{q}, t)$, les forces qui en dérivent sont supposées conservatives. Ainsi, la dynamique du système dans le cadre de la mécanique d'Euler-Lagrange est donnée par :

$$\frac{d}{dt} \left(\frac{\partial \mathcal{L}}{\partial \dot{\mathbf{q}}} \right) - \frac{\partial \mathcal{L}}{\partial \mathbf{q}} = \mathbf{Q} \quad (2.14)$$

\mathbf{Q} représente les forces non conservatives et les couples appliqués au système.

Pour le quadrirotor à propulseurs fixes

- Le vecteur des positions généralisées est défini par :

$$\mathbf{q} = \begin{pmatrix} \boldsymbol{\xi} & \boldsymbol{\eta} \end{pmatrix}^\top = \begin{pmatrix} x & y & z & \phi & \theta & \psi \end{pmatrix}^\top \in \mathbb{R}^6 \quad (2.15)$$

L'énergie cinétique totale translationnelle et rotationnelle du système est donnée par :

$$\mathcal{T} = \mathcal{T}_{trans} + \mathcal{T}_{rot} = \frac{1}{2} \dot{\boldsymbol{\xi}}^\top M \dot{\boldsymbol{\xi}} + \frac{1}{2} \dot{\boldsymbol{\eta}}^\top W_\eta^\top I_B W_\eta \dot{\boldsymbol{\eta}} \quad (2.16)$$

Où $M = \text{diag}(m, m, m) \in \mathbb{R}^{3 \times 3}$. En choisissant comme référence des potentiels le plan $\{e_x, e_y\}$, l'énergie potentielle du système est décrite par l'équation suivante :

$$\mathcal{V} = mgz \quad (2.17)$$

Où g est l'intensité du champ de pesanteur $g = 9.807 m/s^2$. En développant l'équation (2.14), on obtient la dynamique suivante :

$$\begin{pmatrix} M & 0 \\ 0 & W_\eta^\top I_B W_\eta \end{pmatrix} \begin{pmatrix} \ddot{\boldsymbol{\xi}} \\ \ddot{\boldsymbol{\eta}} \end{pmatrix} + \begin{pmatrix} 0 & 0 \\ 0 & C(\boldsymbol{\eta}, \dot{\boldsymbol{\eta}}) \end{pmatrix} \begin{pmatrix} \dot{\boldsymbol{\xi}} \\ \dot{\boldsymbol{\eta}} \end{pmatrix} + \mathcal{G} = \begin{pmatrix} \mathbf{F}^O \\ \boldsymbol{\tau}^O \end{pmatrix} \quad (2.18)$$

$\mathcal{G} = \begin{pmatrix} 0 & 0 & mg \end{pmatrix}^\top \in \mathbb{R}^3$, $\mathbf{F}^O = \begin{pmatrix} F_x & F_y & F_z \end{pmatrix}^\top \in \mathbb{R}^3$ et $\boldsymbol{\tau}^O = \begin{pmatrix} \tau_\phi & \tau_\theta & \tau_\psi \end{pmatrix}^\top \in \mathbb{R}^3$ sont l'ensemble des forces et des couples s'exerçant sur le drone exprimés dans le repère O . $C(\boldsymbol{\eta}, \dot{\boldsymbol{\eta}}) \in \mathbb{R}^{3 \times 3}$ est la matrice de Coriolis regroupant les termes gyroscopique et centripète

définie par :

$$C(\boldsymbol{\eta}, \dot{\boldsymbol{\eta}}) = \frac{d}{dt} \left(W_\eta^\top I_B W_\eta \right) \dot{\boldsymbol{\eta}} - \frac{\partial}{\partial \boldsymbol{\eta}} \left(\dot{\boldsymbol{\eta}}^\top W_\eta^\top I_B W_\eta \dot{\boldsymbol{\eta}} \right) \quad (2.19)$$

Pour le quadrirotor à propulseurs pivotants:

- le vecteur des positions généralisées est défini par :

$$\mathbf{q} = \left(\boldsymbol{\xi} \quad \boldsymbol{\eta} \quad \boldsymbol{\alpha} \right)^\top = \left(x \quad y \quad z \quad \phi \quad \theta \quad \psi \quad \alpha_1 \quad \alpha_2 \quad \alpha_3 \quad \alpha_4 \right)^\top \in \mathbb{R}^{10} \quad (2.20)$$

Où $\boldsymbol{\alpha} \in \mathbb{R}^4$ regroupe les positions angulaires des différents ensembles {bras, moteur et hélice}. À la différence du cas précédant, on ajoute l'énergie cinétique de rotation des ensembles {bras, moteur et hélice}, ainsi on définit l'énergie cinétique totale par :

$$\mathcal{T} = \frac{1}{2} \dot{\boldsymbol{\xi}}^\top M \dot{\boldsymbol{\xi}} + \frac{1}{2} \dot{\boldsymbol{\eta}}^\top W_\eta^\top I_B W_\eta \dot{\boldsymbol{\eta}} + \frac{1}{2} \sum_{i=1}^4 \dot{\boldsymbol{\alpha}}_i^\top I_{\mathcal{P}}^{\alpha_i} \dot{\boldsymbol{\alpha}}_i \quad (2.21)$$

L'énergie potentielle restant la même, on peut déduire la dynamique du système en développant l'équation (2.14), on obtient la dynamique suivante :

$$\begin{pmatrix} M & 0 & 0 \\ 0 & W_\eta^\top I_B W_\eta & 0 \\ 0 & 0 & I_p \end{pmatrix} \begin{pmatrix} \ddot{\boldsymbol{\xi}} \\ \ddot{\boldsymbol{\eta}} \\ \ddot{\boldsymbol{\alpha}} \end{pmatrix} + \begin{pmatrix} 0 & 0 & 0 \\ 0 & C(\boldsymbol{\eta}, \dot{\boldsymbol{\eta}}) & 0 \\ 0 & 0 & 0 \end{pmatrix} \begin{pmatrix} \dot{\boldsymbol{\xi}} \\ \dot{\boldsymbol{\eta}} \\ \dot{\boldsymbol{\alpha}} \end{pmatrix} + \begin{pmatrix} \mathcal{G} \\ 0 \end{pmatrix} = \begin{pmatrix} \mathbf{F}^O \\ \boldsymbol{\tau}^O \\ \boldsymbol{\tau}_m^{\mathcal{B}} \end{pmatrix} \quad (2.22)$$

$I_p = \text{diag}(I_{yy}^{\alpha_1}, I_{yy}^{\alpha_2}, I_{yy}^{\alpha_3}, I_{yy}^{\alpha_4})$ regroupe l'inertie sur axe e_{b_i} des quatres ensembles {bras, moteur et hélice}, $\boldsymbol{\tau}_m^{\mathcal{B}} = \left(\tau_{\alpha_1} \quad \tau_{\alpha_2} \quad \tau_{\alpha_3} \quad \tau_{\alpha_4} \right)^\top \in \mathbb{R}^4$ sont les couples subis par ces derniers.

2.3.4 Bilan des forces et des couples

Comme le mouvement du drone est décrit en fonction des forces et des couples appliqués à son centre de gravité, il est nécessaire d'explicitier et de décrire les différents couples et forces issus des actionneurs et des effets aérodynamiques affectant sa dynamique.

Le vecteur de force totale exercée sur le drone peut être décomposé en somme de deux termes : $\mathbf{F}_r^{\mathcal{B}}$ qui sont les forces générées par les rotors et $\mathbf{F}_a^{\mathcal{B}}$ les forces aérodynamiques tous exprimés dans le référentiel \mathcal{B} ,

$$\mathbf{F}^O = R(\boldsymbol{\eta}) \left(\mathbf{F}_r^{\mathcal{B}} + \mathbf{F}_a^{\mathcal{B}} \right) \quad (2.23)$$

On décompose le couple total $\boldsymbol{\tau}^{\mathcal{B}}$ subi par le drone en trois termes, $\boldsymbol{\tau}_r^{\mathcal{B}}$ est le couple total généré par les quatre rotors, $\boldsymbol{\tau}_a^{\mathcal{B}}$ sont les couples aérodynamiques et $\boldsymbol{\tau}_g^{\mathcal{B}}$ regroupe les couples générés

par la rotation de l'ensemble $\{\text{bras}, \text{moteur}, \text{hélice}\}$ tous exprimés dans le repère \mathcal{B} ,

$$\boldsymbol{\tau}^{\mathcal{B}} = \boldsymbol{\tau}_r^{\mathcal{B}} + \boldsymbol{\tau}_a^{\mathcal{B}} + \boldsymbol{\tau}_g^{\mathcal{B}} \quad (2.24)$$

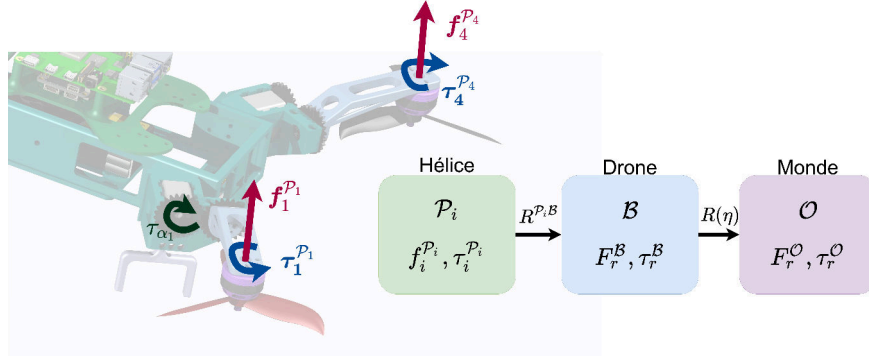


Figure 2.4: Représentation des couples et des forces. Les différents repères et notations des forces/couples sont indiqués à droite.

Portance et trainée des hélices

- La poussée ou force de portance $f_i^{\mathcal{P}_i}$ de l' i -ème hélice peut être exprimée comme une fonction du carré de la vitesse de rotation du moteur ω_i et d'un paramètre b qui dépend des propriétés aérodynamiques de l'hélice. Ajoutant à cela le fait que la rotation de l'hélice génère aussi des forces tangentielles dues à la résistance de l'air, ce qui génère un couple de frottement noté $\tau_i^{\mathcal{P}_i}$, d dépendant aussi des propriétés aérodynamiques de l'hélice.

$$\tau_i^{\mathcal{P}_i} = I_i^r \dot{\omega}_i + d\omega_i^2, \quad \mathbf{f}_i^{\mathcal{P}_i} = (0, 0, b\omega_i^2)^\top \quad (2.25)$$

On considère dans la suite que le terme $I_i^r \dot{\omega}_i$ est négligeable, on peut donc écrire que :

$$\boldsymbol{\tau}_i^{\mathcal{P}_i} = \begin{pmatrix} 0 & 0 & (-1)^{i-1} d\omega_i^2 \end{pmatrix}^\top \quad (2.26)$$

Il s'agit là d'un modèle bien établi qui plus est validé expérimentalement [28], [70]–[72]. Ainsi la totalité des couples et forces générés par les propulseurs est définie par :

$$\boldsymbol{\tau}_r^{\mathcal{B}} = \mathbf{l}_i \times \mathbf{F}_r^{\mathcal{B}} + \sum_{i=1}^4 \left(R^{\mathcal{P}_i \mathcal{B}}(\alpha_i, \beta_i) \boldsymbol{\tau}_i^{\mathcal{P}_i} \right), \quad \mathbf{F}_r^{\mathcal{B}} = \sum_{i=1}^4 \left(R^{\mathcal{P}_i \mathcal{B}}(\alpha_i, \beta_i) \mathbf{f}_i^{\mathcal{P}_i} \right) = \sum_{i=1}^4 \begin{pmatrix} f_{x_i} \\ f_{y_i} \\ f_{z_i} \end{pmatrix}, \quad \mathbf{l}_i = \begin{pmatrix} l_i^x \\ l_i^y \\ 0 \end{pmatrix} \quad (2.27)$$

$$l_i^x : \begin{cases} l^x & i \in \{1,4\} \\ -l^x & i \in \{2,3\} \end{cases}, \quad l_i^y : \begin{cases} l^y & i \in \{3,4\} \\ -l^y & i \in \{1,2\} \end{cases} \quad (2.28)$$

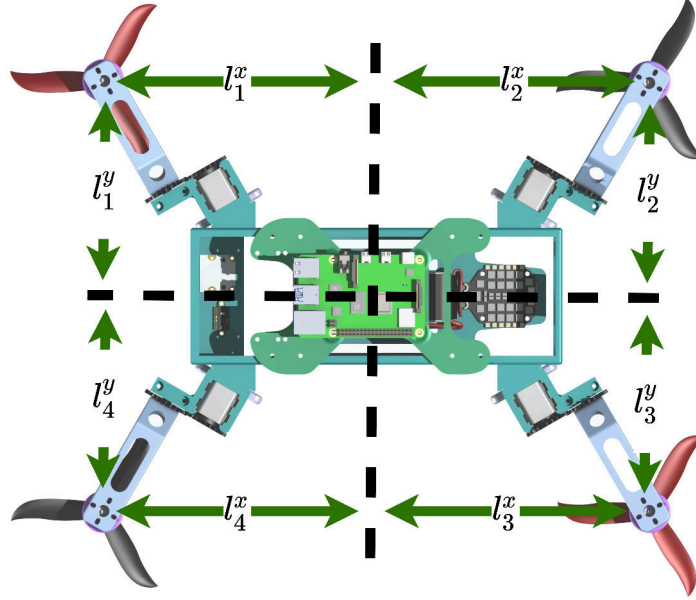


Figure 2.5: Représentation des différentes distances moteurs par rapport au centre de gravité du drone.

Couples gyroscopiques et servomoteurs :

- On considère que le rotor et l'hélice sont un unique corps rigide dont le moment angulaire L_i^r est uniquement dû à la rotation du rotor sur son axe de rotation. En faisant pivoter le rotor en générant un couple $(0 \quad \tau_{\alpha_i} \quad 0)^T$, on modifie son moment de rotation, le vecteur résultant ne change pas de magnitude, mais de direction. En se basant sur le principe de conservation du moment angulaire qui stipule que le moment angulaire reste constant, cela induit un couple gyroscopique exerçant une rotation déplaçant l'axe du rotor vers l'axe de rotation imposé, ou pour aligner son axe avec celui de la rotation transversale imposée [73]–[75]. τ_g^B est défini de la manière suivante :

$$\tau_g^B = \sum_{i=1}^4 \left(L_i^r \times \left(R_{e_{c_i}}(\beta_i) \dot{\alpha}_i + \Omega \right) + R_{e_{c_i}}(\beta_i) \begin{pmatrix} 0 \\ \tau_{\alpha_i} \\ 0 \end{pmatrix} \right), \quad L_i^r = R^{\mathcal{P}_i \mathcal{B}}(\alpha_i, \beta_i) \begin{pmatrix} 0 \\ 0 \\ (-1)^{i-1} I_i^r \omega_i \end{pmatrix} \quad (2.29)$$

On notera que dans notre cas d'étude $\dot{\beta}_i = 0$.

Forces et couples aérodynamiques :

- La modélisation aérodynamique du drone a pour but de caractériser l'influence du vent sur

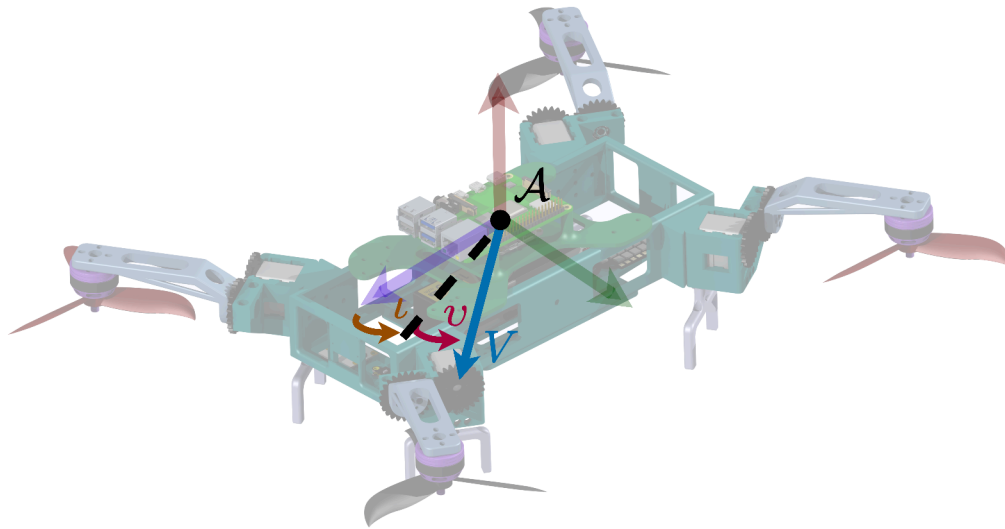


Figure 2.6: Représentation du vecteur vitesse et des angles d'attaque α et de dérive ν

la dynamique du drone. Les forces et couples qui en découlent ont des origines diverses et variées et dépendent de plusieurs paramètres. Ces phénomènes ont déjà fait l'objet d'études et de tentatives de modélisation, par exemple [76]–[80] donnent un aperçu global sur les efforts agissant sur le véhicule ainsi que sur les hélices. On retrouve parmi ces forces, la trainée induite. Pour une hélice considérée comme un solide rigide soumise au vent, la déformation de l'hélice ne suffit pas à balancer la différence de poussée entre les pales, ce qui engendre de la trainée. Les hélices subissent d'autres types de trainées, comme la trainée de profil et de translation lorsqu'elles sont en mouvement par rapport à l'air. Dans [78], l'étude de l'influence des effets aérodynamiques sur la dynamique du drone a permis de conclure que pendant un vol interactions aérodynamiques mutuelles entre les différents composants, entraîne par exemple une perte de poussée pour certaines hélices qui atteint les 11%. Néanmoins ces effets restent négligeable pour une plage de vitesses de 0 à 15 m/s

Notant aussi que, lorsque le drone vole à proximité d'une surface, d'autres phénomènes aérodynamiques apparaissent dit effets de plafond et effet de sol [81], [82].

Cependant, on notera que la plupart de ces effets peuvent être considérés comme négligeables pour des vitesses translationnelles basses. Pour cette raison et aussi, en prenant en considération tous ces phénomènes, cela augmenterait la complexité du modèle. Il faut également noter que l'idée derrière est d'avoir un modèle qu'on peut utiliser pour la commande ou l'observation. Dans la suite, on ne prendra en considération que les effets aérodynamiques découlant du profil aérodynamique du drone. On définit ainsi les forces aérodynamiques \mathbf{F}_a^B :

$$\mathbf{F}_a^{\mathcal{B}} = \frac{1}{2}\rho A R^{\mathcal{B}\mathcal{A}}(\iota, \nu) C(\iota, \nu) V^2 \quad (2.30)$$

A est la surface frontale de référence et ρ est la masse volumique de l'air. On considère que le drone évolue dans un environnement sans présence de vent, on note la norme de la vitesse du drone relative à l'air $V \geq 0$, ι est l'angle d'attaque et $\nu \in \left] -\frac{\pi}{2}, \frac{\pi}{2} \right[$ l'angle de dérive, définis par :

$$V = \sqrt{a^2 + b^2 + c^2}, \quad \iota = \arctan_4\left(\frac{c}{a}\right), \quad \nu = \arcsin\left(\frac{b}{V}\right) \quad (2.31)$$

\arctan_4 est la tangente inverse à quatre quadrants, $C(\iota, \nu) \in \mathbb{R}^3$ est le vecteur des coefficients aérodynamiques défini de la manière suivante :

$$C(\iota, \nu) = \begin{pmatrix} C_l(\iota, \nu) \\ -C_s(\iota, \nu) \\ -C_d(\iota, \nu) \end{pmatrix} \quad (2.32)$$

$C_l(\iota, \nu) \in \mathbb{R}$ est le coefficient de portance, $C_s(\iota, \nu) \in \mathbb{R}$ est le coefficient de portance latérale et $C_d(\iota, \nu) \in \mathbb{R}$ est le coefficient de traînée.

Le drone est aussi soumis à des couples aérodynamiques dus au fait que le centre de pression n'est pas aligné avec le centre de gravité du drone, $\boldsymbol{\tau}_a^{\mathcal{B}}$ les couples aérodynamiques sont définis par :

$$\boldsymbol{\tau}_a^{\mathcal{B}} = \frac{1}{2}\rho A s(\iota, \nu) C_\tau(\iota, \nu) V^2 \quad (2.33)$$

s est la corde, ligne entre le bord d'attaque et le bord de fuite du drone, $C_\tau(\iota, \nu) \in \mathbb{R}^3$ regroupe les coefficients des différents moments aérodynamiques,

$$C_\tau(\iota, \nu) = \begin{pmatrix} C_\phi(\iota, \nu) \\ C_\theta(\iota, \nu) \\ C_\psi(\iota, \nu) \end{pmatrix} \quad (2.34)$$

Modèle des dynamiques des actionneurs :

- Pour générer la poussée ou pour orienter les bras, des moteurs électriques sont utilisés. En effet, le drone est équipé de deux types de moteurs : moteur à courant continu MCC et BLDC (BrushLess Direct Current), les brushless (sans balais BLDC) sont utilisés pour faire tourner les hélices à très haute vitesse. Ces moteurs sont des machines synchrones qui reproduisent le comportement d'un MCC. À la différence d'une MCC, la commutation se fait électroniquement par des ESC "Electronic Speed Controller". Ce dernier maintient l'axe

magnétique du champ induit par les bobines du stator perpendiculaire à l'axe magnétique du rotor. Cela le rend plus robuste en comparaison à une MCC dont la commutation est faite mécaniquement par un collecteur, qui a tendance à s'user avec le temps.

Le plus souvent, on suppose que la dynamique des moteurs brushless est très rapide ou instantanée, ce qui est équivalent à prétendre que le moteur dispose d'un couple/courant infini. Cependant, la dynamique des moteurs peut être négligée dans le cas où la dynamique de la vitesse de rotation désirée reste relativement lente, c'est-à-dire on reste dans la bande passante des actionneurs. Afin de vérifier que notre commande respecte les limitations physiques des actionneurs, un modèle représentatif [83]–[86] des moteurs à courant continu a été établi, il faut noter que les deux types de moteurs peuvent être représentés de la même manière :

Équation électrique :

$$E_m(t) = K_E \omega_m(t) \quad (2.35a)$$

$$v_m(t) = E_m(t) + L_m \frac{d}{dt} i_m(t) + r_m i_m(t) \quad (2.35b)$$

Équation mécanique :

$$\tau_m(t) = K_\tau (i_m(t) - i_{m0}) \quad (2.36a)$$

$$I_m \dot{\omega}_m(t) = \tau_m(t) - \tau_d(t) \quad (2.36b)$$

Où,

- $E_m(t)$ est la force électromotrice en [V]
- K_E est la constante électromotrice en [Vs/rad]
- ω_m est la vitesse de rotation du moteur en [rad/s]
- L_m est l'inductance électrique en [H]
- r_m est la résistance électrique du moteur en [Ω]
- τ_m est le couple moteur en [Nm]
- τ_d est le couple de frottement et qui dans le cas du BLDC comporte le couple de résistance dû à la trainée d'une hélice en [Nm]
- i_{m0} est le courant à vide en [A]
- i_m est le courant à charge en [A]

Les BLDC sont généralement conçus pour avoir $L < 0.2$ mH, ce qui se traduit par une dynamique électrique très rapide en comparaison avec la dynamique mécanique du moteur qui dépend de l'inertie du rotor. Dans le cas des servomoteurs l'inertie de l'ensemble { moteur, bras et hélices }, n'est pas négligeable et affecte largement la dynamique mécanique.

2.4 Limitation du torseur des actions mécaniques

Après avoir détaillé le modèle du drone dans la section précédente (2.3), on s'intéresse dans cette partie à établir une stratégie d'actionnement du drone. Dans le cas du Xtilt, on dispose de 8 entrées de commande qui correspondent aux vitesses de rotation des moteurs et aux orientations des bras par rapport au châssis du drone. En effet, pour contrôler le drone, il est nécessaire de connaître la relation entre les entrées de commande et le torseur des actions mécaniques que l'on souhaite appliquer au drone, afin de le faire mouvoir dans un sens ou un autre. On considère comme entrée pour notre système le vecteur $\mathbf{H}^{\mathcal{B}}$ des forces des actionneurs :

$$\mathbf{H}^{\mathcal{B}} = \left(f_{z_1} \quad \tilde{f}_{x_1} \quad f_{z_2} \quad \tilde{f}_{x_2} \quad f_{z_3} \quad \tilde{f}_{x_3} \quad f_{z_4} \quad \tilde{f}_{x_4} \right)^{\top} \quad (2.37)$$

Les éléments sont définis dans l'équation Eq.(2.25) exprimés en \mathcal{B} de la manière suivante $\mathbf{f}_i^{\mathcal{B}} = (f_{x_i}, f_{y_i}, f_{z_i})^{\top} = (b\omega_i^2 s_{\alpha_i} c_{\beta_i}, b\omega_i^2 s_{\alpha_i} s_{\beta_i}, b\omega_i^2 c_{\alpha_i})^{\top}$, $\tilde{f}_{x_i} = b\omega_i^2 s_{\alpha_i}$, avec $c_{\star} = \cos(\star)$ et $s_{\star} = \sin(\star)$. On définit sous forme vectorielle le torseur des actions mécanique qu'on souhaite appliquer au drone exprimé dans le repère \mathcal{B} par,

$$\mathbf{U}^{\mathcal{B}} = \left(u_x \quad u_y \quad u_z \quad u_{\phi} \quad u_{\theta} \quad u_{\psi} \right)^{\top} = \begin{pmatrix} \sum_1^4 f_{x_i} \\ \sum_1^4 f_{y_i} \\ \sum_1^4 f_{z_i} \\ \boldsymbol{\tau}_r^{\mathcal{B}} \end{pmatrix} \quad (2.38)$$

Des équations précédentes, on déduit la relation qui lie les entrées du système au vecteur des actions mécaniques, on a donc :

$$\mathbf{U}^{\mathcal{B}} = \mathbf{G}^{\mathcal{B}} \mathbf{H}^{\mathcal{B}} \quad (2.39)$$

$\mathbf{G} \in \mathbb{R}^{6 \times 8}$ est dite matrice d'allocation (allocation matrix, mapping matrix), elle est la relation entre le torseur des actions mécaniques et les sorties des actionneurs. Le rang de la matrice \mathbf{G} nous permet de classifier notre système. En effet le rang de \mathbf{G} donne une indication sur la dimension de l'ensemble atteignable par $\mathbf{U}^{\mathcal{B}}$. Dans notre cas \mathbf{G} est défini par :

$$\mathbf{G} = \begin{pmatrix} 0 & c_{\beta_1} & 0 & c_{\beta_2} & 0 & c_{\beta_3} & 0 & c_{\beta_4} \\ 0 & s_{\beta_1} & 0 & s_{\beta_2} & 0 & s_{\beta_3} & 0 & s_{\beta_4} \\ 1 & 0 & 1 & 0 & 1 & 0 & 1 & 0 \\ l_1^y & \frac{d}{b} c_{\beta_1} & l_2^y & -\frac{d}{b} c_{\beta_2} & l_3^y & \frac{d}{b} c_{\beta_3} & l_4^y & -\frac{d}{b} c_{\beta_4} \\ -l_1^x & \frac{d}{b} s_{\beta_1} & -l_2^x & -\frac{d}{b} s_{\beta_2} & -l_3^x & \frac{d}{b} s_{\beta_3} & -l_4^x & -\frac{d}{b} s_{\beta_4} \\ \frac{d}{b} & l_1^x s_{\beta_1} - l_1^y c_{\beta_1} & -\frac{d}{b} & l_2^x s_{\beta_2} - l_2^y c_{\beta_2} & \frac{d}{b} & l_3^x s_{\beta_3} - l_3^y c_{\beta_3} & -\frac{d}{b} & l_4^x s_{\beta_4} - l_4^y c_{\beta_4} \end{pmatrix} \quad (2.40)$$

Comme on souhaite contrôler la position du drone par rapport au repère monde fixe \mathcal{O} , il est plus judicieux de décrire le torseur des forces subi par le centre de gravité du drone dans le référentiel \mathcal{O} . On définit ainsi le vecteur force :

$$\begin{pmatrix} u_x \\ u_y \\ u_z \end{pmatrix} = R^{O\mathcal{B}}(\boldsymbol{\eta})^\top \begin{pmatrix} \tilde{u}_x \\ \tilde{u}_y \\ \tilde{u}_z \end{pmatrix} \quad (2.41)$$

\tilde{u}_x, \tilde{u}_y et \tilde{u}_z sont respectivement les forces sur x,y et z que l'on souhaite appliquer au drone décrit dans le référentiel \mathcal{O} . Afin d'obtenir les différentes orientations des bras et les vitesses de rotation des moteurs en fonction du torseur mécanique que l'on souhaite appliquer à ce dernier. On utilise la méthode des moindres carrés pour la pseudoinverse qui donne la meilleure solution possible à l'équation Eq(2.39), $\mathbf{H}^{\mathcal{B}}$ étant le vecteur unique avec la norme la plus petite qui minimise $\|\mathbf{U}^{\mathcal{B}} - G^{\mathcal{B}}\mathbf{H}^{\mathcal{B}}\|$.

La pseudoinverse de G est calculée à l'aide de la décomposition en valeurs singulières, $G_v = \tilde{Q}\Sigma\tilde{R}^*$ ou $\tilde{Q} \in \mathbb{R}^{6 \times 6}$, $\tilde{R} \in \mathbb{R}^{8 \times 8}$ sont des matrices orthogonales et $\Sigma \in \mathbb{R}^{6 \times 8}$ est une matrice diagonale rectangulaire dont les valeurs diagonales sont positives. Cela conduit à :

$$G^\dagger = \tilde{R}\Sigma^{-1}\tilde{Q}^* \quad (2.42)$$

Ainsi, partant de $\mathbf{H}^{\mathcal{B}} = G^\dagger \mathbf{U}^{\mathcal{B}}$, nous pouvons calculer la vitesse de rotation pour chaque moteur et l'angle d'inclinaison de chaque bras :

$$\alpha_i = \arctan\left(\frac{\tilde{f}_{x_i}}{\tilde{f}_{z_i}}\right) \quad ; \quad \omega_i = \left(\frac{\tilde{f}_{x_i}^2 + \tilde{f}_{z_i}^2}{b^2}\right)^{\frac{1}{4}} \quad (2.43)$$

Plusieurs configurations existent en fonction des choix des β_i . Ces configurations diffèrent d'une part, dans leur possibilité à garantir de rendre le système totalement contrôlable ou non, dépendant du rang de la matrice G . D'autre part, pour appliquer une même force sur un des axes, la norme de $\|\mathbf{H}^{\mathcal{B}}\| = b\sqrt{\sum_1^4 \omega_i^4}$ dépend de la configuration choisie. En effet, certaines configurations ont tendance à minimiser $\|\mathbf{H}^{\mathcal{B}}\|$ ce qui représente un gain d'énergie, puisque pour une même force on a des vitesses de rotation des moteurs qui sont moindres. Afin d'illustrer cela, deux cas et trois configurations chacune ont été comparées, pour deux orientations données, on trace $\|\mathbf{H}^{\mathcal{B}}\|$ en fonction de \tilde{u}_x, \tilde{u}_y et \tilde{u}_z . Le tableau Tab.(2.1) résume les différentes configurations testées. Pour les deux cas de figure on a $\tilde{u}_x \in [-5, 5]$, $\tilde{u}_y \in [-5, 5]$ et $\tilde{u}_z \in [0, 10]$.

- **Cas 1**, dans ce premier cas, on remarque dans les figures figure.(2.7) figure.(2.8), que pour générer une même force \tilde{u}_x , la configuration c1 nécessite une norme $\|\mathbf{H}^{\mathcal{B}}\|$ deux fois inférieures en comparaison avec la configuration c3 figure.(2.7). Cependant, d'après la

Orientation	Configuration
Cas 1 : $\phi = 0, \theta = 0, \psi = 0$	c1 : $\beta_1 = \beta_2 = \beta_3 = \beta_4 = 0$
	c2 : $\beta_1 = -\beta_2 = \beta_3 = -\beta_4 = \frac{\pi}{3}$
	c3 : $\beta_1 = -\beta_2 = \beta_3 = -\beta_4 = \frac{\pi}{4}$
Cas 2 : $\phi = 0, \theta = \frac{\pi}{4}, \psi = 0$	c4 : $\beta_1 = \beta_2 = \frac{\pi}{2}, \beta_3 = \beta_4 = 0$

Table 2.1: Les différentes configurations et orientations testées.

figure (2.9), pour la configuration c1, aucune force \tilde{u}_y ne peut être générée. En effet, pour cette configuration le rang de G est 5, le système est dit sous-actionné. En ce qui concerne la force \tilde{u}_x figure (2.10), pour ce premier cas, les différentes configurations produisent une norme $\|\mathbf{H}^{\mathcal{B}}\|$ de même valeur.

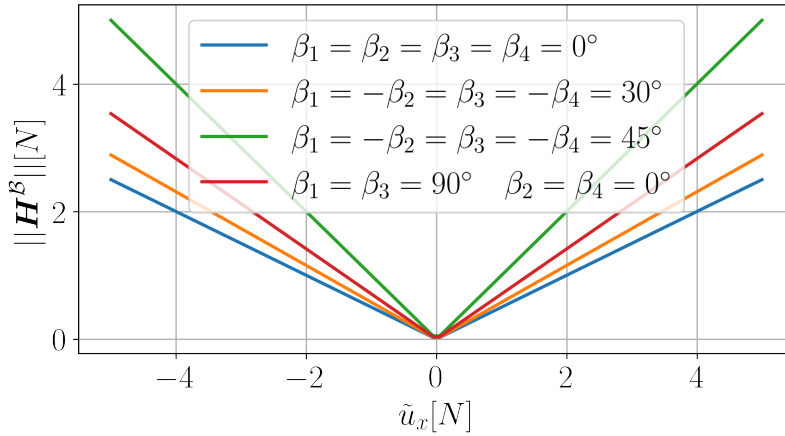


Figure 2.7: $\|\mathbf{H}^{\mathcal{B}}\|$ en fonction de \tilde{u}_x pour $\phi = 0, \theta = 0$ et $\psi = 0, \tilde{u}_y = \tilde{u}_z = 0$.

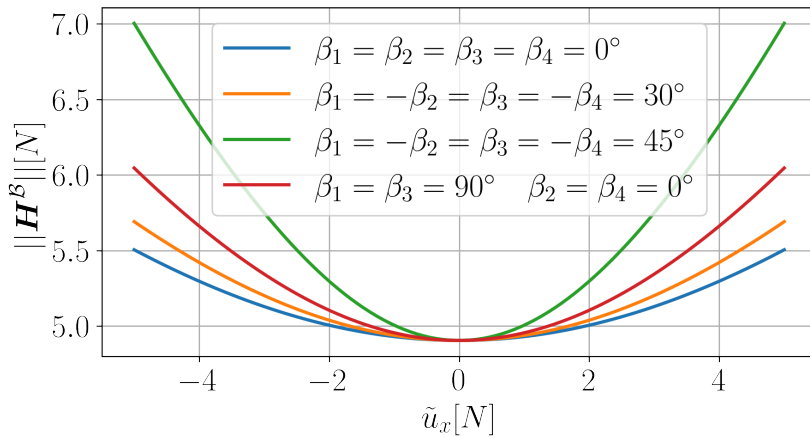


Figure 2.8: $\|\mathbf{H}^{\mathcal{B}}\|$ en fonction de \tilde{u}_x pour $\phi = 0, \theta = 0$ et $\psi = 0, \tilde{u}_y = 0$ et $\tilde{u}_z = 9.81$.

- **Cas 2** dans le second cas, on remarque la même chose pour \tilde{u}_x et \tilde{u}_y . Cependant, pour \tilde{u}_z les conclusions sont différentes. En effet, on remarque dans la figure (2.11) que pour

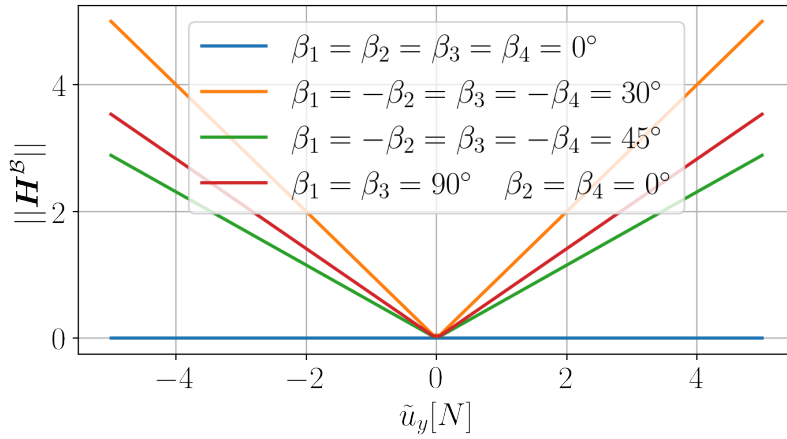


Figure 2.9: $\|\mathbf{H}^B\|$ en fonction de \tilde{u}_y pour $\phi = 0, \theta = 0$ et $\psi = 0, \tilde{u}_x = \tilde{u}_z = 0$.

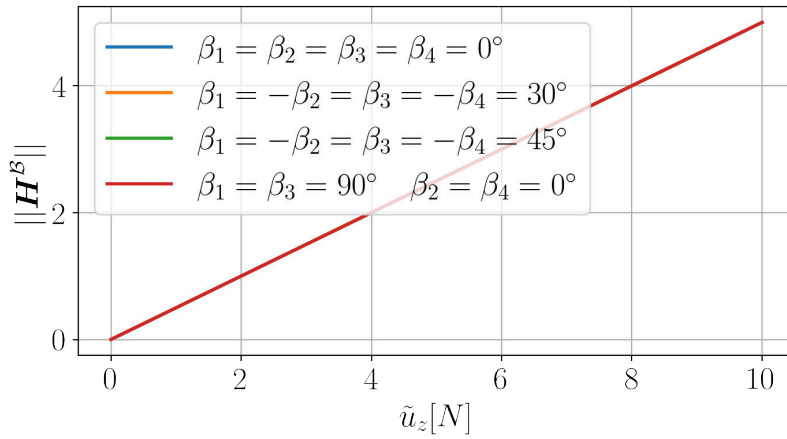


Figure 2.10: $\|\mathbf{H}^B\|$ en fonction de \tilde{u}_z pour $\phi = 0, \theta = 0$ et $\psi = 0, \tilde{u}_x = \tilde{u}_y = 0$.

générer une même force selon e_z , la configuration c1 nécessite un $\|\mathbf{H}^B\|$ inférieur en comparaison avec les autres configurations.

On a donc fait le choix pour le Xtilt d'utiliser la première configuration c1 : $\beta_1 = \beta_2 = \beta_3 = \beta_4 = 0$. En effet ce choix est motivé par plusieurs raisons. Même si pour cette configuration, le système est sous-actionné, on souhaite en effet utiliser pleinement sa capacité à garder une assiette différente en mouvement. La configuration c1 est la plus convenable puisque le gain en énergie est considérable. En ce qui concerne la partie sous-actionnée, on développera dans la suite les stratégies utilisées pour pallier cela.

2.4.1 Cas des bras pivotants Xtilt

Pour la configuration choisie pour ce drone, on a $\beta_1 = \beta_2 = \beta_3 = \beta_4 = 0$, la matrice d'allocation G est définie par :

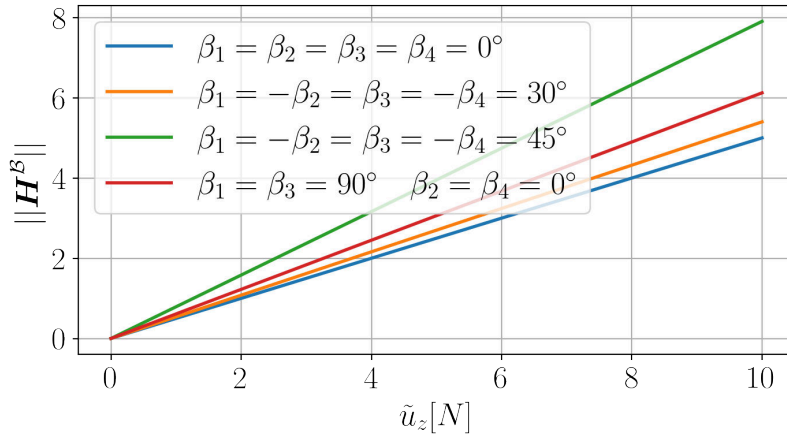


Figure 2.11: $\|H^B\|$ en fonction de \tilde{u}_z pour $\phi = 0, \theta = \frac{\pi}{4}$ et $\psi = 0, \tilde{u}_x = \tilde{u}_y = 0$.

$$G = \begin{pmatrix} 0 & 1 & 0 & 1 & 0 & 1 & 0 & 1 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 1 & 0 & 1 & 0 & 1 & 0 & 1 & 0 \\ l_1^y & \frac{d}{b} & l_2^y & -\frac{d}{b} & l_3^y & \frac{d}{b} & l_4^y & -\frac{d}{b} \\ -l_1^x & 0 & -l_2^x & 0 & -l_3^x & 0 & -l_4^x & 0 \\ \frac{d}{b} & -l_1^y & -\frac{d}{b} & -l_2^y & \frac{d}{b} & -l_3^y & -\frac{d}{b} & -l_4^y \end{pmatrix} \quad (2.44)$$

Le rang de G est inférieur au nombre de degrés de liberté (3 rotations et 3 translations), ainsi dans cette configuration les actionneurs ne sont pas suffisants pour contrôler tous les degrés de liberté. En effet, on peut voir dans 2.44 qu'avec cette configuration, on ne peut générer que deux forces et trois couples. Cependant, il est en effet possible de contrôler la position totale du drone en ajoutant une boucle externe assignant une attitude souhaitée pour appliquer une force désirée sur l'axe e_y dans le référentiel O . En partant des équations suivantes Eq.(2.38) et Eq.(2.41), cela conduit à la relation suivante :

$$\begin{pmatrix} \tilde{u}_x \\ \tilde{u}_y \\ \tilde{u}_z \end{pmatrix} = R^{OB}(\boldsymbol{\eta}) \begin{pmatrix} \sum_1^4 f_{x_i} \\ \sum_1^4 f_{y_i} \\ \sum_1^4 f_{z_i} \end{pmatrix} \quad (2.45)$$

On obtient ainsi la relation suivante,

$$\sin(\psi)\tilde{u}_x - \cos(\psi)\tilde{u}_y = \sin(\phi) \sum_1^4 f_{z_i} = \sin(\phi)u_z \quad (2.46)$$

Il est possible de résoudre (2.46) de deux manières différentes : soit en contrôlant le lacet ψ , ce qui implique une navigation nonholonomique, soit en contrôlant le roulis ϕ , qui présente

un profil de mouvement holonomique,

$$\begin{cases} \phi_d = \arcsin\left(\frac{\sin(\psi)\tilde{u}_x - \cos(\psi)\tilde{u}_y}{u_z}\right) \\ \psi_d = \arcsin\left(\frac{\sin(\phi)u_z}{\sqrt{\tilde{u}_x^2 + \tilde{u}_y^2}}\right) + \arctan\left(\frac{\tilde{u}_y}{\tilde{u}_x}\right) \end{cases} \quad (2.47)$$

Il est à noter que cette résolution est valable si les conditions suivantes sont accomplies.

$$\left| \frac{\sin(\psi)\tilde{u}_x - \cos(\psi)\tilde{u}_y}{u_z} \right| \leq 1, \quad \left| \frac{\sin(\phi)u_z}{\sqrt{\tilde{u}_x^2 + \tilde{u}_y^2}} \right| \leq 1 \quad (2.48)$$

Par volonté d'exhaustivité, présentons le cas nonholonomique, c'est-à-dire le contrôle de ϕ pour piloter le mouvement selon l'axe e_y . On a,

$$\begin{cases} u_x = \cos(\theta)\cos(\psi)\tilde{u}_x + \cos(\theta)\sin(\psi)\tilde{u}_y - \sin(\theta)\tilde{u}_z \\ u_y = 0 \\ u_z = (\sin(\phi)\sin(\psi) + \cos(\phi)\cos(\psi)\sin(\theta))\tilde{u}_x \\ \quad + (\cos(\phi)\sin(\psi)\sin(\theta) - \cos(\psi)\sin(\phi))\tilde{u}_y + \cos(\theta)\cos(\phi)\tilde{u}_z \end{cases} \quad (2.49)$$

Par exemple que pour le cas où $\psi = \frac{\pi}{2}$, le contrôle sur \tilde{u}_y passe sur u_x et par conséquent, le contrôle sur \tilde{u}_x passe sur ϕ_d . Nous pouvons donc contrôler la position complète du drone. Plus précisément, nous avons dans ce cas

$$\begin{cases} u_x = \cos(\theta)\tilde{u}_y - \sin(\theta)\tilde{u}_z \\ \phi_d = \arcsin\left(\frac{\tilde{u}_x}{u_z}\right) \end{cases} \quad (2.50)$$

Contraintes d'actionnement

La résolution précédente ne prend pas en considération quelques limitations physiques que le système subit. En effet, pour rester en vol le drone a besoin de garder une certaine force selon l'axe e_z pour compenser son poids. Il est nécessaire de prendre en considération cela, car la force \tilde{u}_x et le couple u_ψ vont avoir tendance à aligner e_{c_i} avec e_1 comme illustré dans la figure (2.12). Il est possible d'éviter cela en ajoutant une saturation sur l'orientation des propulseurs, en définissant une plage d'angle que les servomoteurs ne peuvent pas dépasser. Comme on souhaite que le drone garde toute sa capacité de tangage, une autre approche a été développée pour pallier cela. En effet, il est possible de saturer la force \tilde{u}_x et le couple u_ψ en fonction de

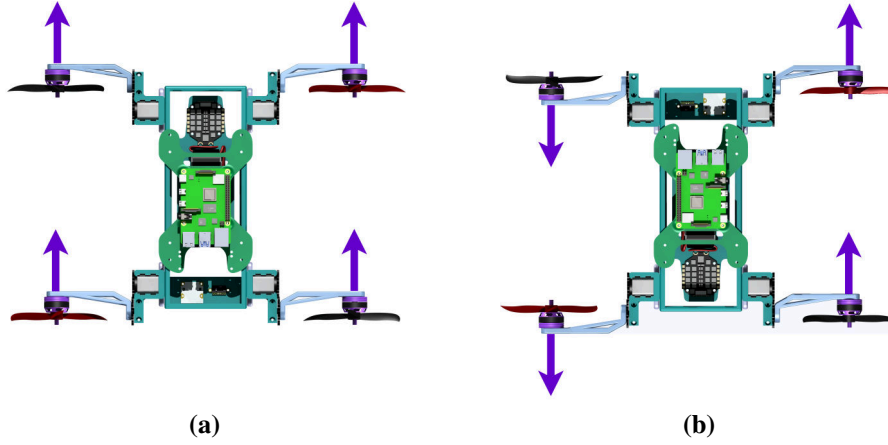


Figure 2.12: Description des différentes possibilités d'enveloppe de contrôle: (a) Contrôle différentiel du lacet (b) mouvement horizontal sans changer d'orientation.

u_z qui a tendance à ramener les propulseurs vers une position verticale. On définit ainsi nos gains de saturation $\mu_x > 0$ et $\mu_\psi > 0$, ainsi la matrice G est donnée par,

$$G = \begin{pmatrix} 0 & \mu_x & 0 & \mu_x & 0 & \mu_x & 0 & \mu_x \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 1 & 0 & 1 & 0 & 1 & 0 & 1 & 0 \\ l_1^y & \frac{d}{b} & l_2^y & -\frac{d}{b} & l_3^y & \frac{d}{b} & l_4^y & -\frac{d}{b} \\ -l_1^x & 0 & -l_2^x & 0 & -l_3^x & 0 & -l_4^x & 0 \\ \frac{d}{b} & -\mu_\psi l_1^y & -\frac{d}{b} & -\mu_\psi l_2^y & \frac{d}{b} & -\mu_\psi l_3^y & -\frac{d}{b} & -\mu_\psi l_4^y \end{pmatrix} \quad (2.51)$$

Les gains de saturation μ_x et μ_ψ sont définis par,

$$\mu_x = \begin{cases} 1 & \text{si } \left| \frac{u_x + s_\theta \tilde{u}_z}{u_z} \right| \leq \tan(\alpha_{max}^x) \\ \left| \frac{u_x - s_\theta \tilde{u}_z}{u_z} \right| \frac{1}{\tan(\alpha_{max}^x)} & \text{sinon} \end{cases} \quad (2.52)$$

$$\mu_\psi = \begin{cases} 1 & \text{si } \left| \frac{u_\psi}{u_z} \right| \leq \tan(\alpha_{max}^\psi) \\ \left| \frac{u_\psi}{u_z} \right| \frac{1}{\tan(\alpha_{max}^\psi)} & \text{sinon} \end{cases} \quad (2.53)$$

où $\alpha_{max}^x \in [\frac{\pi}{6}, \frac{\pi}{3}]$ et $\alpha_{max}^\psi \in [\frac{\pi}{6}, \frac{\pi}{3}]$ définissent les limites de la saturation. Notons que cette saturation ne concernera que la commande sur x et ψ , nous redéfinissons donc u_x et u_z par,

$$\begin{cases} u_x = \cos(\theta) \cos(\psi) \tilde{u}_x + \cos(\theta) \sin(\psi) \tilde{u}_y - \sin(\theta) \tilde{u}_z \mu_x \\ u_z = (\sin(\phi) \sin(\psi) + \cos(\phi) \cos(\psi) \sin(\theta)) \tilde{u}_x \\ \quad + (\cos(\phi) \sin(\psi) \sin(\theta) - \cos(\psi) \sin(\phi)) \tilde{u}_y + \cos(\theta) \cos(\phi) \tilde{u}_z \mu_x \end{cases} \quad (2.54)$$

2.4.2 Quadrirotor à bras fixes

Pour le quadricoptère à bras fixes on a $\alpha_i = 0$ pour $i \in \{1, 2, 3, 4\}$, la matrice G s'écrit donc,

$$G = \begin{pmatrix} 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 1 & 0 & 1 & 0 & 1 & 0 & 1 & 0 \\ l_1^y & 0 & l_2^y & 0 & l_3^y & 0 & l_4^y & 0 \\ -l_1^x & 0 & -l_2^x & 0 & -l_3^x & 0 & -l_4^x & 0 \\ \frac{d}{b} & 0 & -\frac{d}{b} & 0 & \frac{d}{b} & 0 & -\frac{d}{b} & 0 \end{pmatrix} \quad (2.55)$$

G est de rang 4, le système est sous actionné et ne peut générer qu'une seule force et trois couples. Cependant, il est possible de le contrôler en position en ajoutant une boucle externe qui contrôle l'attitude du drone, ceci a été largement abordé dans la littérature [87]–[89]. On définit ainsi,

$$\begin{pmatrix} \tilde{u}_x \\ \tilde{u}_y \\ \tilde{u}_z \end{pmatrix} = R^{OB}(\boldsymbol{\eta}) \begin{pmatrix} 0 \\ 0 \\ \sum_1^4 b\omega_i^2 \end{pmatrix} = \begin{pmatrix} \sum_1^4 b\omega_i^2 (c_\phi s_\theta c_\psi + s_\phi s_\psi) \\ \sum_1^4 b\omega_i^2 (c_\phi s_\theta s_\psi - s_\phi c_\psi) \\ \sum_1^4 b\omega_i^2 (c_\theta c_\phi) \end{pmatrix} \quad (2.56)$$

La poussée et l'attitude désirées sont exprimées ainsi

$$\begin{cases} \sum_1^4 b\omega_i^2 = \sqrt{\tilde{u}_x^2 + \tilde{u}_y^2 + \tilde{u}_z^2} \\ \phi_d = \arcsin\left(\frac{\sin(\psi)\tilde{u}_x - \cos(\psi)\tilde{u}_y}{\sqrt{\tilde{u}_x^2 + \tilde{u}_y^2 + \tilde{u}_z^2}}\right) \\ \theta_d = \arctan\left(\frac{\cos(\psi)\tilde{u}_x + \sin(\psi)\tilde{u}_y}{\tilde{u}_z}\right) \end{cases} \quad (2.57)$$

2.5 Mécanique des fluides numérique

En l'absence de moyens expérimentaux adéquats et nécessaires, la mécanique des fluides numérique (CFD) est le seul moyen qui permet de reproduire et substituer l'expérience. L'objectif de cette section est d'identifier à l'aide d'une simulation CFD en 3D, les différents paramètres aérodynamiques du Xtilt. Le modèle CFD du drone nous permettra aussi d'observer l'interaction aérodynamique entre un rotor et l'air ainsi que le châssis du drone et l'air. La modélisation CFD du Xtilt est tridimensionnelle, réalisée sous un logiciel libre d'accès OpenFOAM [90] v2112 (Open Field Operation and Manipulation) ¹. Ce dernier résout les équations de Navier-Stokes par discrétisation (méthode des volumes finis), en fonction des conditions aux limites choisies, la

¹<https://openfoam.org>

forme de l'objet et d'autres hypothèses. Ce logiciel est largement utilisé que dans le domaine de la recherche et de l'industrie, pour différentes applications, par exemple la modélisation d'un quadrirotor Cyclogyro [91] ou la modélisation de propagation de gaz basée CFD par un essaim de drones [92], ou plus classiquement pour l'optimisation des ailes d'un VTOL [93].

Comme pour n'importe quel objet en mouvement dans un fluide, les efforts aérodynamiques ne dépendent pas seulement de la forme de l'objet, mais aussi de quelques paramètres physiques liés aux propriétés du fluide. L'air est considéré un fluide Newtonien dont la viscosité est indépendante des contraintes mécaniques. Deux éléments qui permettent de définir la nature des écoulements, ce sont le nombre de Reynold et le nombre de Mach, on définit dans la suite ces différents paramètres et leurs dépendances : Soit

- $\nu = 15.6e^{-6} \text{ m}^2/\text{s}$ la viscosité cinématique de l'air à 20°C
- $\rho_{air} = 1.2 \text{ kg}/\text{m}^3$ la masse volumique de l'air à 20°C
- $R_e = \frac{LV}{\nu}$ est le nombre de Reynolds (viscosité relative de l'air) sans dimension. Il caractérise le type d'écoulement, V est la vitesse de l'écoulement défini (2.31), L en m est l'envergure du drone. Pour la gamme de vitesse du drone entre $0 - 15\text{m}/\text{set}$ en on considèrant que les hélices ne tourne pas. on a $R_e \approx 2.5e^5$, l'écoulement est dit laminaire [94].
- Cependant, en ce qui concerne les hélices le nombre de Reynolds est donné par [95]:

$$R_e^h = \frac{\rho_{air}\omega r_h c_h}{\nu}$$
ou ω et la vitesse de rotation de l'hélice, r_h et c_h sont respectivement le rayon et la corde de l'hélice positionnée au $3/4$ du centre de rotation. À basse vitesse de rotation on reste dans le même ordre de grandeur pour le nombre de Reynolds.
- $M_a = \frac{V}{v_s}$ le nombre de Mach ou v_s est la vitesse du son en considérant l'air comme un gaz parfait. À une basse altitude et pour une température de 20°C $v_s = 340.3\text{m}/\text{s}$. Pour la gamme de vitesses du drone, $M_a \approx 0.045 \ll 1$, ainsi les effets de compressibilité de l'air sont négligeables.

2.5.1 Modèle CFD

Deux modèles de CFD sont développés, le premier est celui du drone dans sa totalité incluant les hélices dans un état stationnaire et un second modèle qui ne comporte qu'une hélice tournante. Les deux cas de figure ont été soumis à des conditions différentes.

Châssis et hélices

OpenFOAM dispose de plusieurs solveurs en fonction du phénomène physique à simuler. Pour calculer les différents paramètres aérodynamiques du drone, on utilisera le solveur `simpleFoam`

pour des écoulements incompressibles et turbulents en régime permanent ce solveur est basé sur la méthode SIMPLE (Semi-Implicit Method for Pressure Linked Equations) [96].

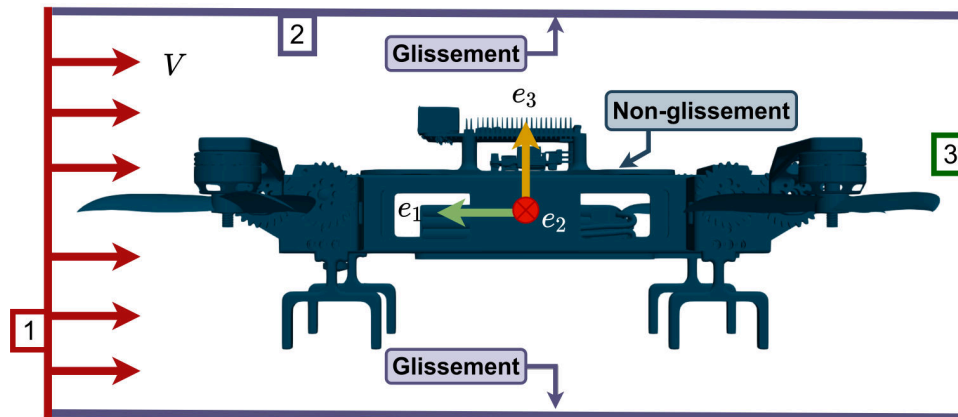


Figure 2.13: Conditions aux limites du modèle CFD, une représentation des différents frontières.

- **Conditions aux limites :** Le volume modélisé fait $1m \times 1m \times 1m$. Ce dernier est généré par le mailleur `blockmesh`, les parois de ce solide sont décomposées en trois domaines en fonction de leurs conditions aux limites figure (2.13):
 - La frontière numéro (1) correspond à l'entrée d'air, sa vitesse V est fixe et la pression a un gradient normal nul. (`inletOutlet`)
 - La frontière numéro (2) correspond aux parois extérieures, on considère une condition de glissement pour la vitesse et la pression a un gradient normal nul.
 - La frontière numéro (3) correspond à l'ouverture du volume modélisé sur l'extérieur (sortie), la pression est celle ambiante et le gradient de vitesse est normal nul. La vitesse est nulle lorsque l'écoulement pénètre dans le domaine. (`inletOutlet`)
 - Pour le drone, on considère une condition de non-glissement, la vitesse tangentielle à la surface du drone est considérée nulle.
- **Maillage :** `snappyHexMesh` est un outil de maillage 3D d'`OpenFoam`. Il permet la génération de maillage à partir de données surfaciques bien définies à partir d'un fichier STL et des fichiers `eMesh` extraits par l'outil `surfaceFeatureExtract`. Ces derniers contiennent les caractéristiques des arêtes de la géométrie du drone. `snappyHexMesh` est plus un 'sculpteur' de maillage qu'un générateur de maillage, car il requiert un maillage de base, celui généré par `blockmesh`. Le domaine initial doit être défini de façon à inclure la totalité des données importées du fichier STL du `xtilt`, contenant les données surfaciques. `snappyHexMesh` est paramétré via le fichier `snappyHexMeshDict` ou l'on peut choisir la zone de raffinement du maillage ainsi que le niveau de raffinement pour les volumes comme pour les surfaces. figure (2.14) représente deux vues de coupe différentes du domaine après raffinement du maillage par `snappyHexMesh`. Pour notre cas d'étude, on est plus

intéressé par les zones à proximité de la surface du drone. En effet, on choisit un niveau de raffinement plus élevé pour les zones proches de la surface extérieure du drone, comme on peut le remarquer sur la figure (2.14), où la zone (2) a un niveau de raffinement plus grand que la zone (1).

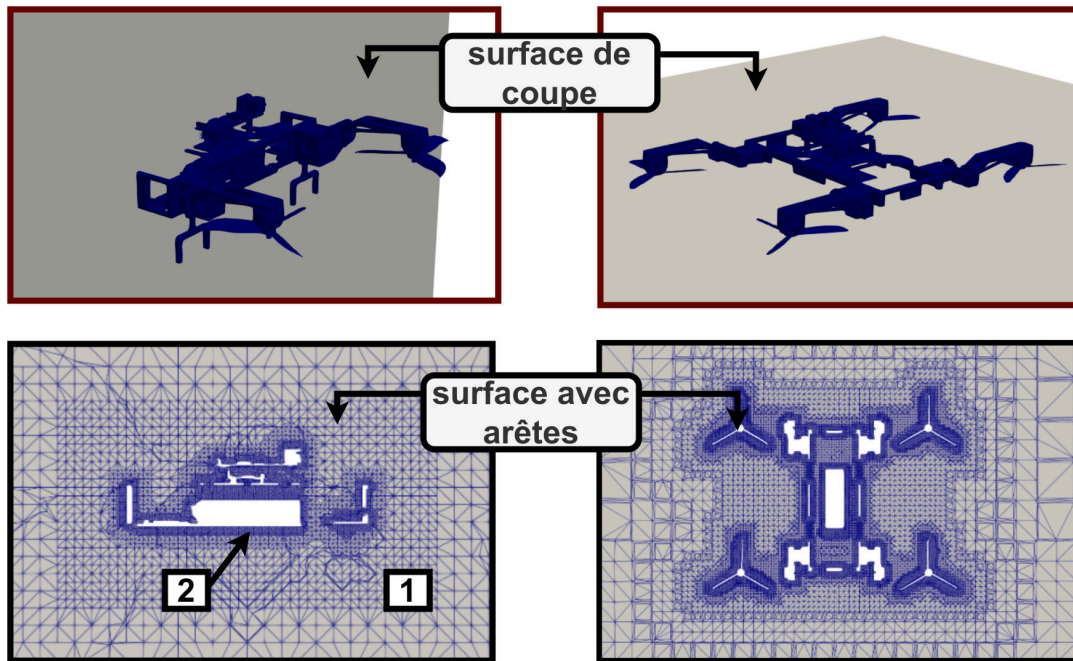


Figure 2.14: Différentes perspectives après le raffinement du maillage par snappyHexMesh. Le maillage en zone (2) est plus affiné qu'en zone (1)

- **Résultats et prétraitement des données :** La simulation a une durée de 5s avec un pas de 0.1s, la tolérance de convergence sur les résidus sont de 10^{-6} pour la pression et la vitesse, les coefficients aérodynamiques sont calculés pour $-\frac{\pi}{2} \leq \iota \leq \frac{\pi}{2}$ et $-\frac{\pi}{2} \leq \nu \leq \frac{\pi}{2}$, pour un pas de $\frac{\pi}{18} rad$. Les coefficients aérodynamiques sont calculés avec la fonction `forceCoeffs`, elle permet le calcul des coefficients des forces et des moments aerodynamiques pour une surface donnée. On peut aussi spécifier les surfaces de référence, on choisira comme surface de référence la surface frontale du drone pour $\iota = \nu = 0 rad$, on peut également définir le centre de rotation pour le calcul des moments, pour notre cas, les moments aérodynamiques seront calculés par rapport au centre de gravité du drone. Dans la figure (2.15), on remarque que les coefficients aérodynamiques convergent vers une valeur constante au bout de 5s. Pour garantir une convergence des coefficients, on prendra la valeur finale calculée des coefficients aérodynamiques. .
 - On remarque dans la figure (2.16) du coefficient de trainée, que l'effet de l'angle d'attaque ι a un grand effet sur la valeur du coefficient de trainée. En effet, cela est dû à la nature de la trainée qui prédomine sur notre système. Il se trouve que le

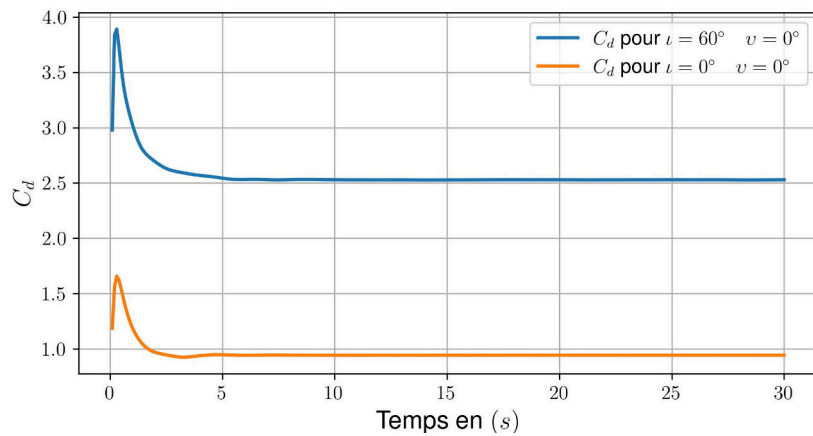


Figure 2.15: Coefficient de trainée C_d en fonction du temps

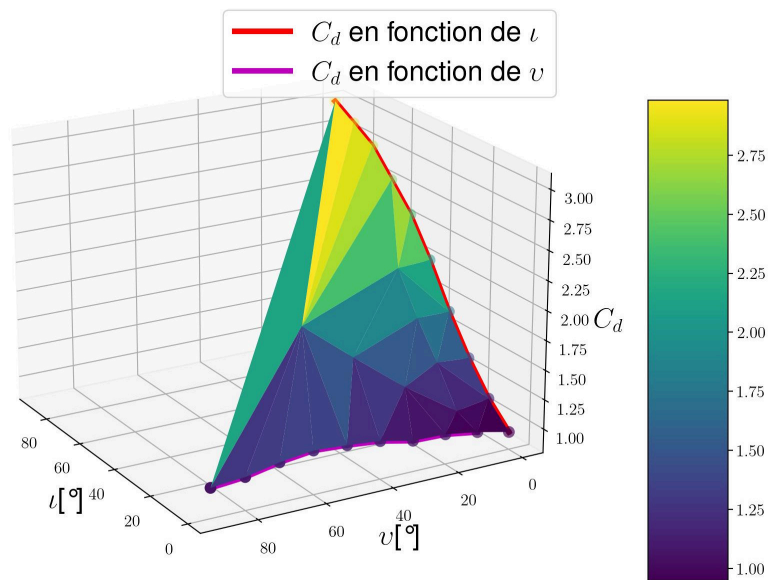


Figure 2.16: Représentation du coefficient de trainée en fonction de α angle d'attaque et de ν angle de dérive

Xtilt est un corps non profilé, cela implique que la trainée de pression (de forme) est prépondérante. Son effet s'accroît par la différence de pression entre le point d'attaque et le bord de fuite. On peut remarquer dans la figure (2.17) que cette différence de pression est plus élevée pour (2.17)(b) dans le cas où l'angle d'attaque $\alpha = 60^\circ$.

Par la suite, on négligera l'effet de l'angle de dérive sur le coefficient de trainée et on considérera que la trainée dépend seulement de l'angle d'attaque α . En modélisant $C_d(\alpha)$ par une régression polynomiale d'ordre quatre, on obtient le modèle suivant pour $C_d(\alpha)$:

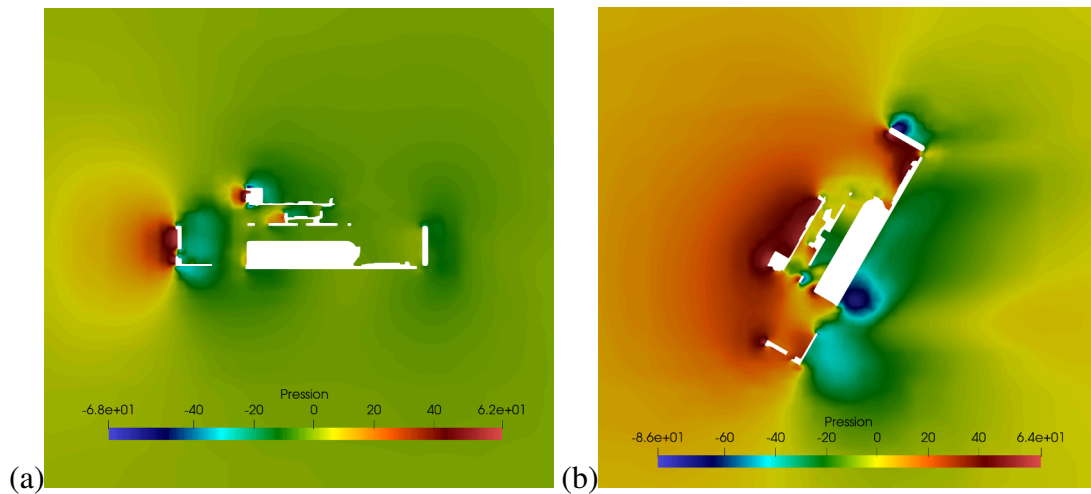


Figure 2.17: Vue en coupe sur le plan (e_1e_3) de la pression de l'air pour deux angles d'attaque différents. Dans les deux cas on a $\nu = 0^\circ$: (a) pour $\iota = 0^\circ$ (b) pour $\iota = 60^\circ$

$$C_d(\iota) = 7.95e^{-3}\iota^4 + 9.074e^{-4}\iota^3 - 1.314e^{-5}\iota^2 + 5.400e^{-7}\iota + 0.952 \quad (2.58)$$

Le modèle proposé a un écart quadratique moyen de 0.00014 et un coefficient de détermination de 0.9997. On retrouve dans la figure (2.18) une comparaison entre la courbe réelle et le modèle.

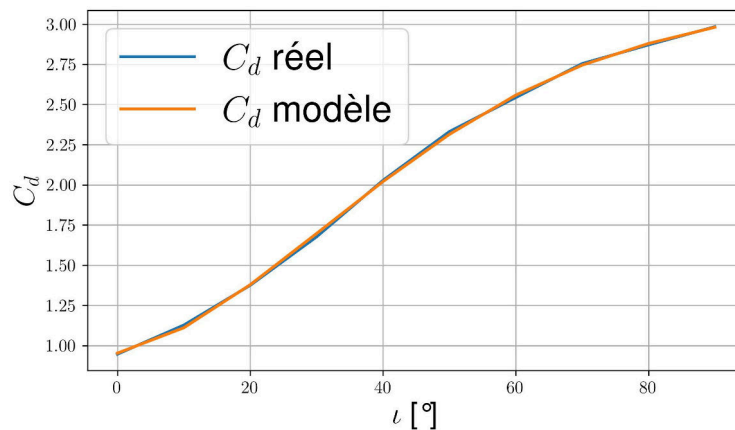


Figure 2.18: Le coefficient de trainée C_d et son modèle lineaire en fonction de l'angle d'attaque ι , en orange la courbe du modèle et en bleu les données réelles.

- On remarque aussi dans la figure (2.19) que le coefficient de portance C_l varie fortement en fonction de l'angle d'attaque ι et très peu pour l'angle de dérive ν . De même que pour la trainée, on considérera que le coefficient de portance ne dépend que de l'angle d'attaque.

On modélise alors $C_l(\iota)$ par une régression polynomiale d'ordre quatre, ce qui permet

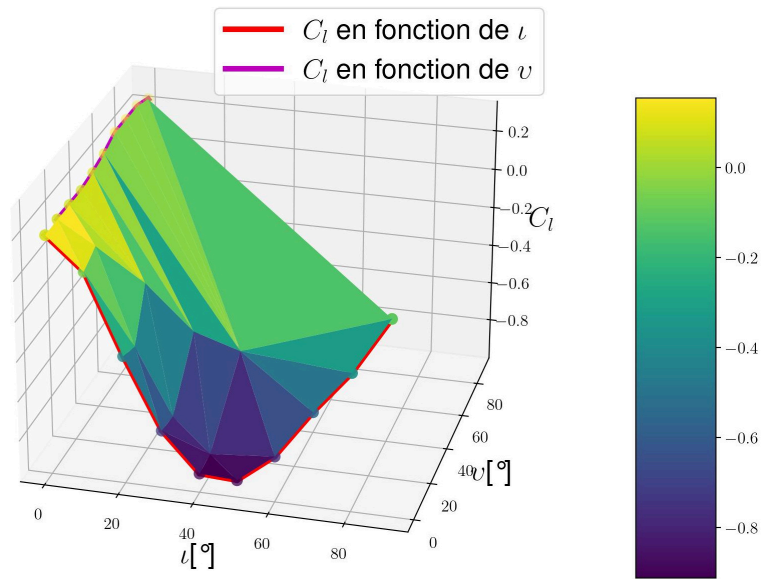


Figure 2.19: Représentation du coefficient de portance C_l en fonction de ι et de v .

d'obtenir le modèle suivant :

$$C_l(\iota) = -1.036e^{-2}\iota^4 - 1.53e^{-3}\iota^3 + 3.52e^{-5}\iota^2 - 1.92e^{-7}\iota + 0.2102 \quad (2.59)$$

Le modèle a un écart quadratique moyen de 0.00094 et un coefficient de détermination de 0.9935 figure (2.20). On retrouve dans la figure (2.20) une comparaison entre la courbe réelle et celle du modèle.

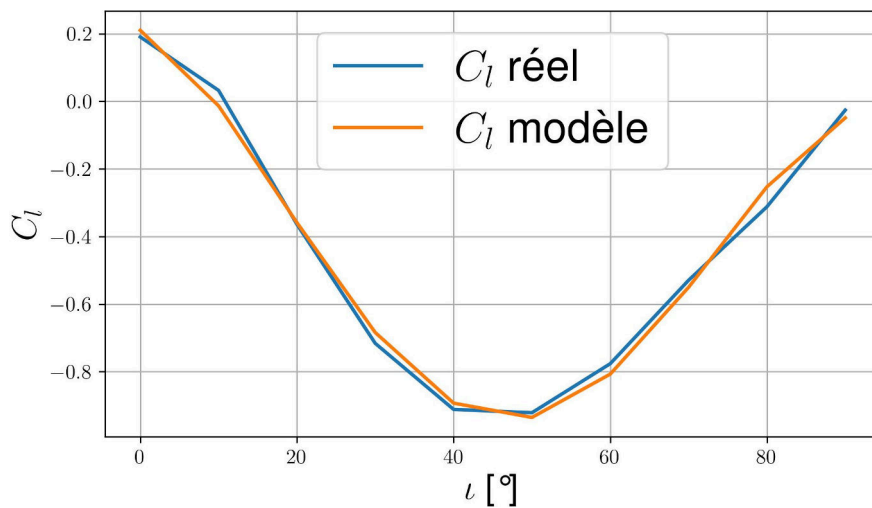


Figure 2.20: Coefficient de portance C_l et son modèle linéaire en fonction de l'angle d'attaque ι , en orange la courbe du modèle et en bleu les données réelles

– En ce qui concerne le coefficient de portance latérale figure (2.21), on remarque

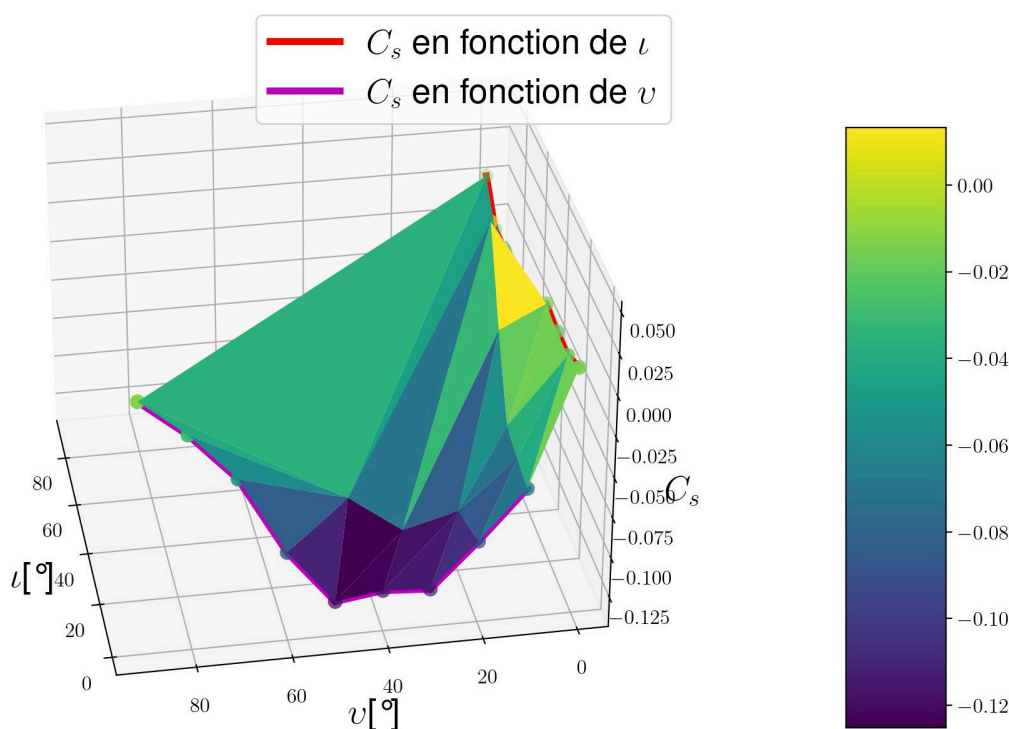


Figure 2.21: Représentation du coefficient de portance latérale C_s en fonction de l'angle d'attaque α et de l'angle de dérive ν .

cette fois que l'angle d'attaque n'a pas un grand impact sur sa variation. Cependant, l'angle de dérive a une certaine influence sur ce coefficient. En effet, l'augmentation de l'angle de dérive génère une variation de pression entre le côté gauche et droit du châssis, ce qui augmente la force aérodynamique latérale, voir figure (2.22). En comparaison avec les autres coefficients aérodynamiques, la variation du coefficient de portance latérale reste relativement petite et varie peu. Pour des raisons de simplification, on considérera que sa valeur est constante et on prendra la moyenne des valeurs de la figure (2.21).

- À l'égard des moments aérodynamiques sur le lacet et le roulis, on remarque que la variation ainsi que leurs valeurs ainsi que leurs variations restent relativement petites, on considérera par la suite que ces deux grandeurs sont constantes. On prendra la valeur moyenne pour ces deux dernières.
- Finalement, le moment aérodynamique sur le tangage est relativement important en comparaison avec les autres moments aérodynamiques. En effet, cela est dû à la forme supérieure du drone. En analysant la figure (2.17), on remarque que la pression n'est pas répartie uniformément, ce qui décale le centre de pression par rapport au centre de gravité (supposé au même endroit que le centre géométrique du drone).

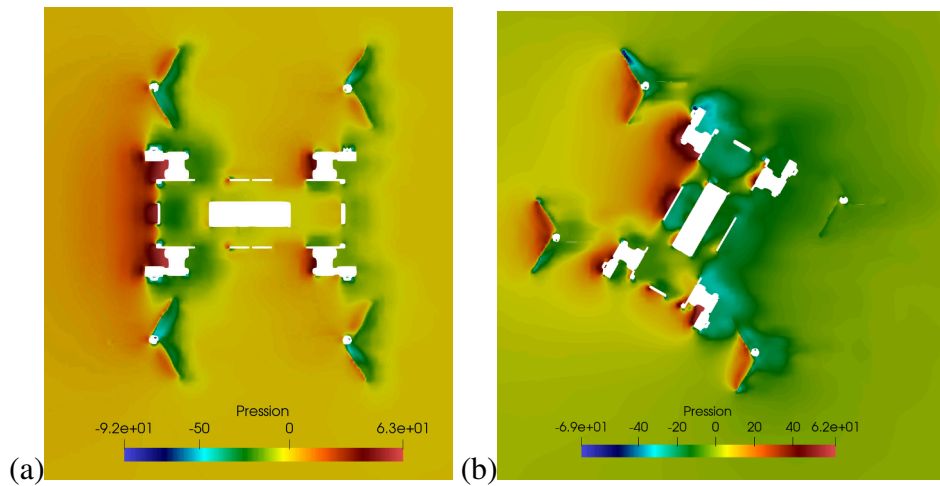


Figure 2.22: Vue en coupe sur le plan (e_1e_2) de la pression de l'air pour deux angles de dérives différents. Dans les deux cas on a $\iota = 0^\circ$: (a) pour $\nu = 0^\circ$ (b) pour $\nu = 60^\circ$.

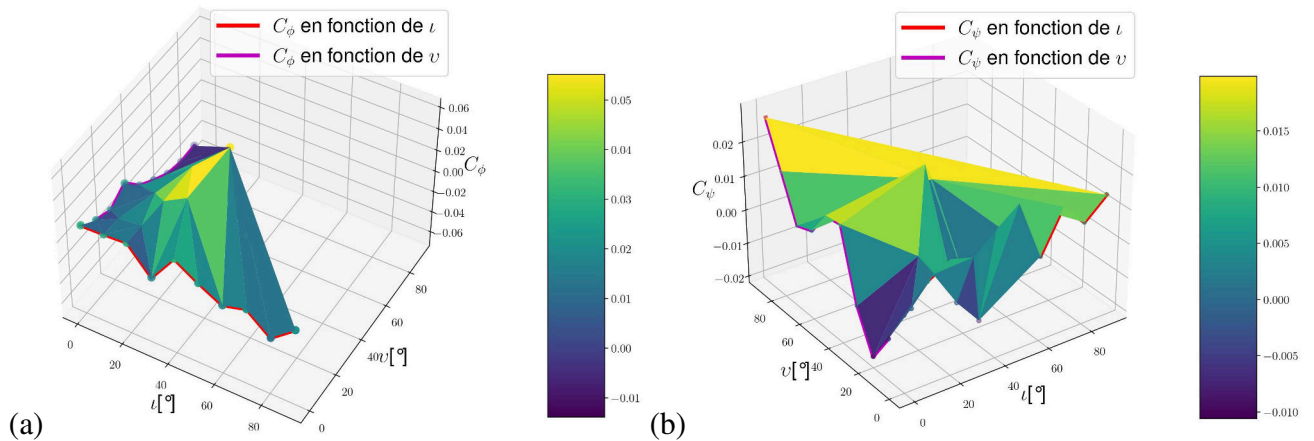


Figure 2.23: Représentation des coefficients des moments en fonction de l'angle d'attaque ι et de l'angle de dérive ν , (a) pour le moment sur l'axe de roulis, (b) pour l'axe de lacet

En s'appuyant sur la figure 2.24, on note que la variation de ce coefficient dépend largement de l'angle d'attaque. On suppose donc que le coefficient du moment aérodynamique sur l'axe de tangage ne dépend que de l'angle d'attaque ι . On modélisera $C_\theta(\iota)$ par une régression polynomiale d'ordre quatre figure (2.20). Le modèle a un écart quadratique moyen de 0.000114 et un coefficient de détermination de 0.9916. On obtient le modèle suivant pour $C_\theta(\iota)$:

$$C_\theta(\iota) = 0.0148\iota^4 - 1.211e^{-3}\iota^3 + 2.131e^{-5}\iota^2 - 1.09e^{-7}\iota + 0.03378 \quad (2.60)$$

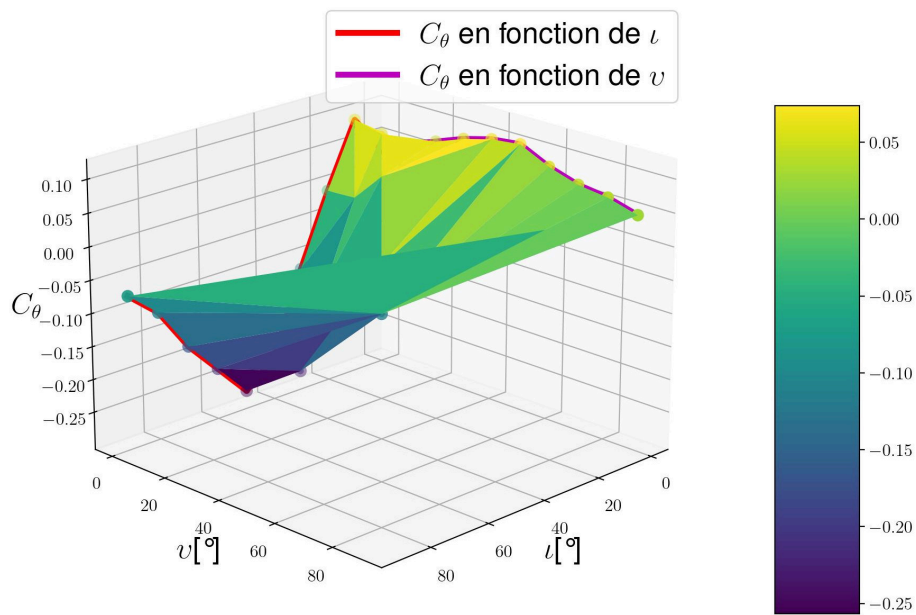


Figure 2.24: Représentation du coefficient du moment sur l'axe de tangage en fonction de l'angle d'attaque ι et de l'angle de dérive ν

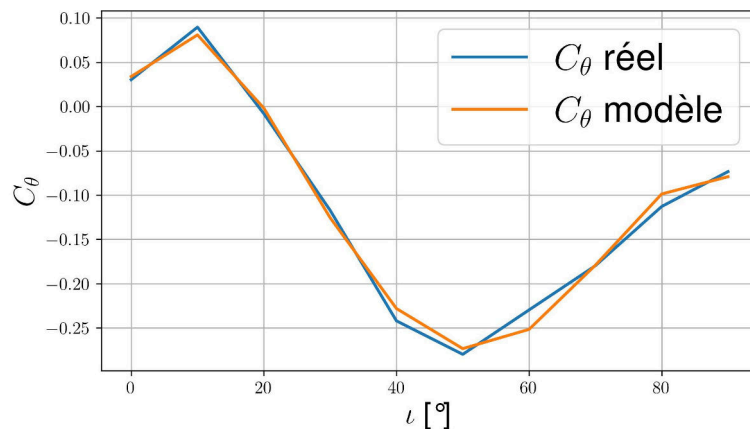


Figure 2.25: Le coefficient de moment de tangage C_θ et son modèle en fonction de l'angle d'attaque ι , en orange la courbe du modèle et en bleu les données réelles

Hélice en rotation

OpenFoam offre la possibilité de générer des maillages rotatifs, ce qui nous sera utile pour modéliser une hélice en rotation. Pour cela on dispose du solveur `pimpleFoam`, une combinaison de deux algorithmes SIMPLE (utilisé précédemment) et PISO (Pressure Implicit with Splitting of Operators) [97]. Pour les maillages rotatifs, on utilise l'algorithme d'interface de maillage arbitraire dit (AMI) [98]. Effet, la simulation d'un maillage rotatif nécessite une distinction entre les régions fixes et rotatives de la géométrie, ces dernières sont couplées aux limites de leurs domaines par le biais de la condition limite AMI cyclique, où on a les mêmes limites

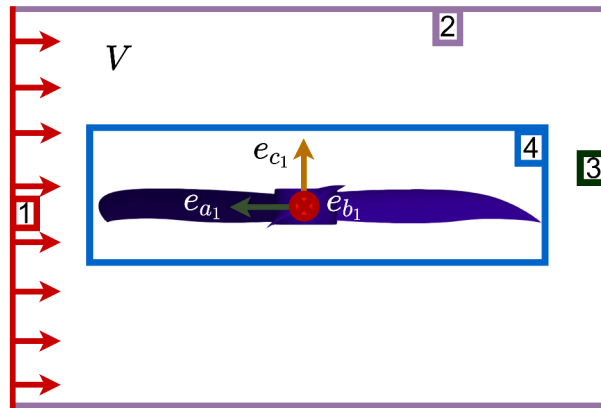


Figure 2.26: Conditions aux limites du modèle CFD pour une hélice du Xtilt, une représentation des différentes frontières

extérieures, mais une construction intérieure différente.

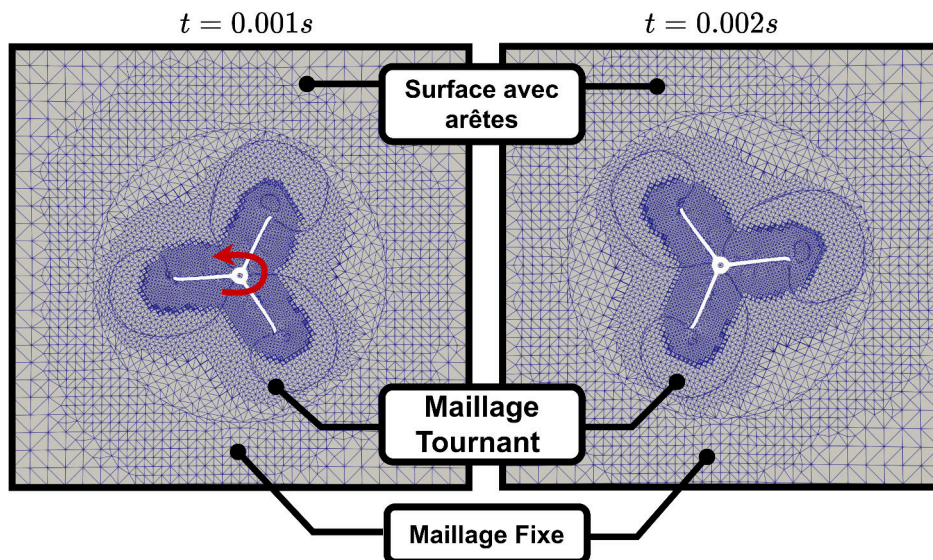


Figure 2.27: Différentes perspectives après le raffinement du maillage par snappyHexMesh autour de l'hélice

- **Conditions aux limites :** Le volume considéré pour cette modélisation fait $0.5m \times 0.5m \times 0.5m$, les conditions aux limites externes restent similaires à la simulation précédente. Cependant, comme on a deux types de maillage différents, un maillage fixe et un maillage rotatif. La zone (4) (figure (2.26)) représente cette délimitation, qui dans le jargon d'OpenFoam est définie comme une zone dite cyclicAMI, qui possède les mêmes conditions extérieures, mais une construction intérieure différente (contenant l'hélice)
- **Maillage** En utilisant le même outil que précédemment, on génère un premier maillage par blockMesh et on raffine ce maillage avec snappyHexMesh, dans la zone d'intérêt et dans

les zones à proximité de cette dernière, comme indiqué sur la figure (2.27)

- **Résultats et prétraitement des données :** Différentes simulations ont été réalisées sous diverses conditions, afin de connaître l'effet de ces dernières sur les efforts de poussées et de trainées que produit l'hélice.

- Dans la première batterie de tests, l'objectif est de connaître l'effet du vent sur la poussée et le couple de trainée qu'une hélice en rotation produit. Dans la figure (2.28), on trace la force de poussée et le couple qu'une hélice produit en absence de vent. Le temps de simulation est de 0.05s, ce qui est largement suffisant pour que la poussée et la trainée convergent vers des valeurs constantes, la vitesse de rotation de l'hélice est fixée à 1000 rad/s. La figure (2.29) représente le flux d'air généré par l'hélice à différents moments. Dans la figure (2.30) on trace la force de poussée et le couple qu'une hélice produit en présence d'un vent horizontal de vitesse 10 m/s. On remarque que pour une même vitesse de rotation de l'hélice et en présence de vent vertical, la poussée ainsi que la trainée générées par l'hélice sont plus grandes. On constate une augmentation de la poussée moyenne de 0.732 N et une augmentation de la norme du couple de trainée de 0.0139 Nm. La figure (2.31) représente le flux d'air généré par l'hélice à différents moments, on retrouve les mêmes conclusions que dans la littérature [99].

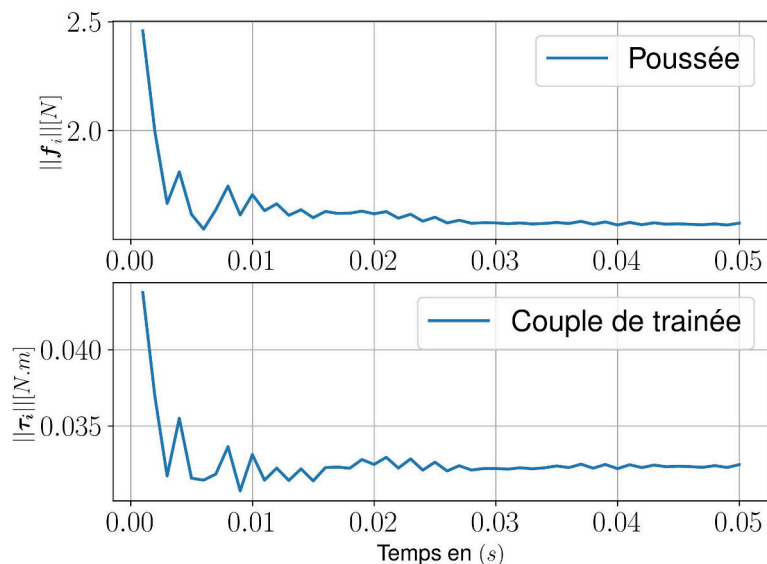


Figure 2.28: La poussée et le couple de trainée générés par l'hélice pour une vitesse de rotation de 1000 rad/s sans présence de vent.

- Après avoir établi l'effet du vent sur les efforts de poussée et de trainée générés par une hélice, on s'intéresse dans cette partie à l'interaction des différents flux d'airs générés par les hélices et leurs effets sur la trainée totale ainsi que la poussée totale

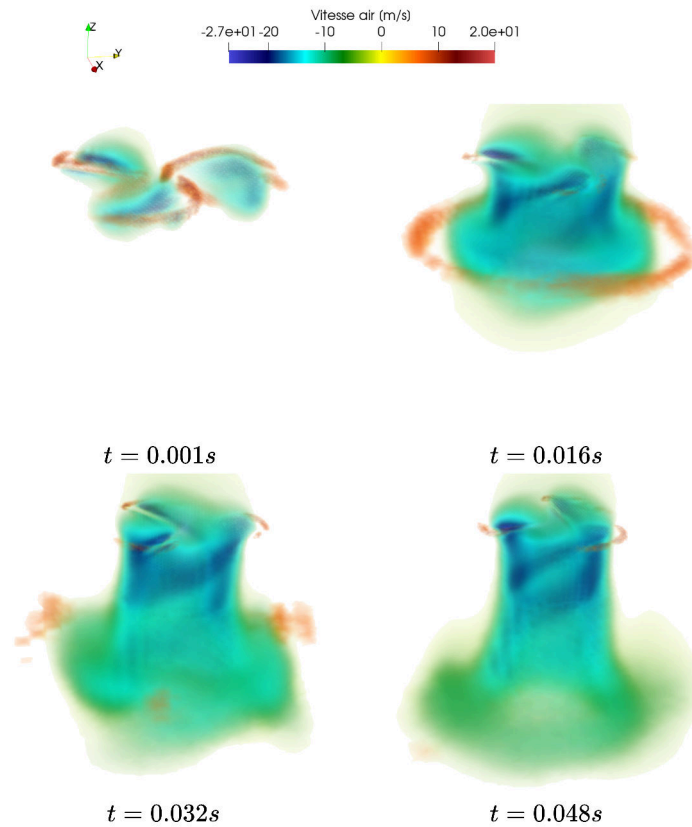


Figure 2.29: Représentation volumique du flux d'air (U_z) sur l'axe vertical que l'hélice génère à différents instants

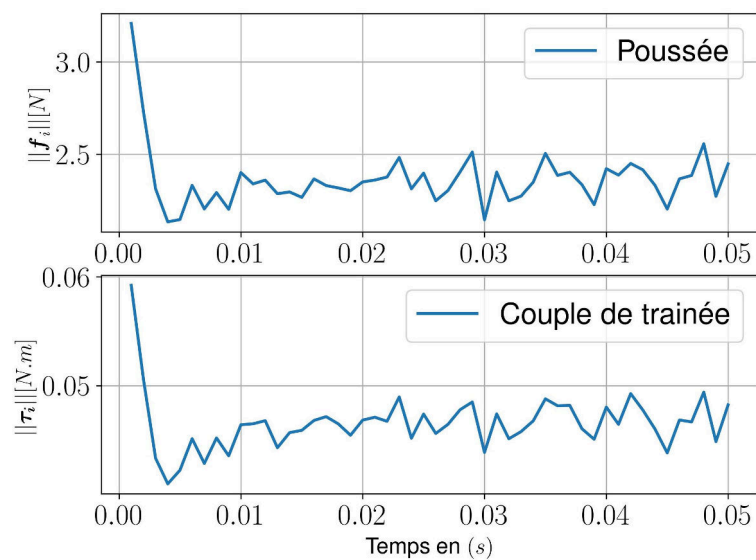


Figure 2.30: La poussée et le couple de trainée générés par l'hélice pour une vitesse de rotation de 1000 rad/s en présence d'un vent de 10 m/s

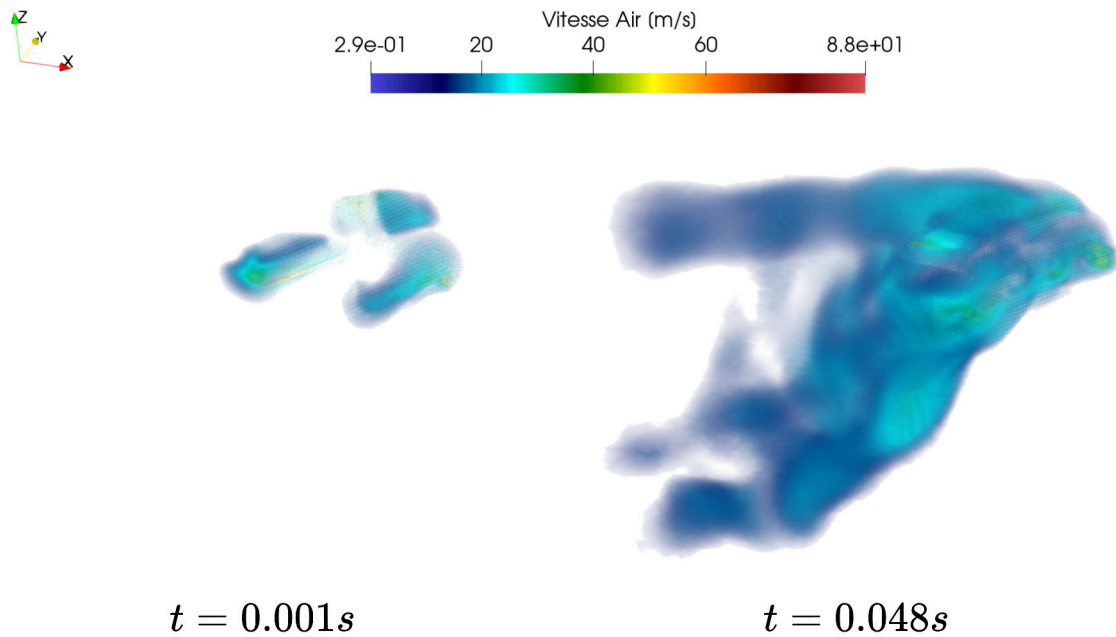


Figure 2.31: Représentation volumique du flux d'air que l'hélice génère à différents instants en présence d'un vent horizontal <https://youtu.be/F2WQaSgTKbE>

générée. En effet, pour le Xtilt, lorsque les propulseurs pivotent les flux d'air générés par les moteurs à l'avant et à l'arrière peuvent interférer. Dans cette partie, on considère les mêmes conditions aux limites que précédemment. Cependant, on a deux hélices en rotation à la même vitesse 1500 rad/s et de sens de rotation différents, une dans le sens horaire (CW) (correspond à l'hélice 1) et l'autre dans le sens contre horaire (CCW) (correspond à l'hélice 2). Une simulation a été réalisée pour un angle de tangage de $\frac{\pi}{3}$ comme représenté dans la figure (2.32). Il faut noter que ces interférences apparaissent pour des angles d'inclinaison supérieurs à $\frac{\pi}{4}$. Dans la figure (2.33), on retrouve la courbe de portance et de couple de trainés générés par chacune des hélices. On remarque qu'à partir de $t = 0.026s$, une différence notable entre les deux hélices en termes de poussée et de couple de trainé générés. En effet, pour l'hélice se trouvant en bas (hélice 2), comme on peut le voir dans la figure (2.34), $t = 0.026s$ correspond au moment où le flux d'air généré par l'hélice se trouvant en haut rejoint l'hélice se trouvant en bas. Non seulement cela cause une différence de poussée et de couple de trainée pour une vitesse de rotation, mais aussi une fluctuation de ces derniers.

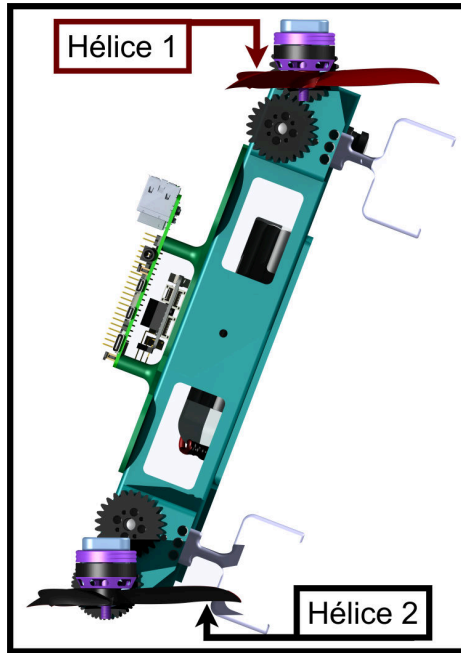


Figure 2.32: Représentation du cas de simulation: un angle d'inclinaison de $\frac{\pi}{3}$

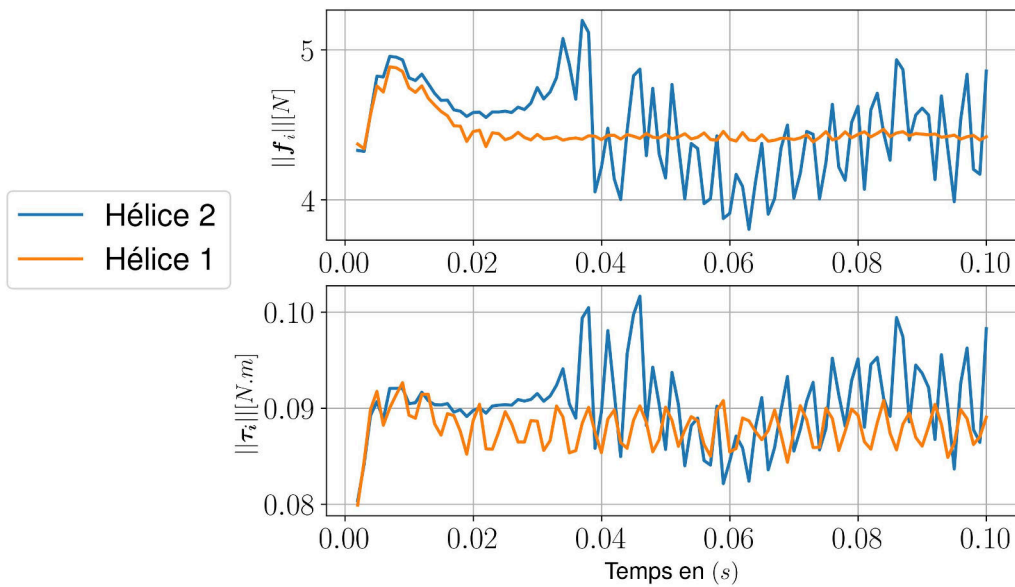


Figure 2.33: La poussée et le couple de trainés généré par les deux hélices pour une vitesse de rotation de $1500rad/s$ et un angle d'inclinaison de $\frac{\pi}{3}$.

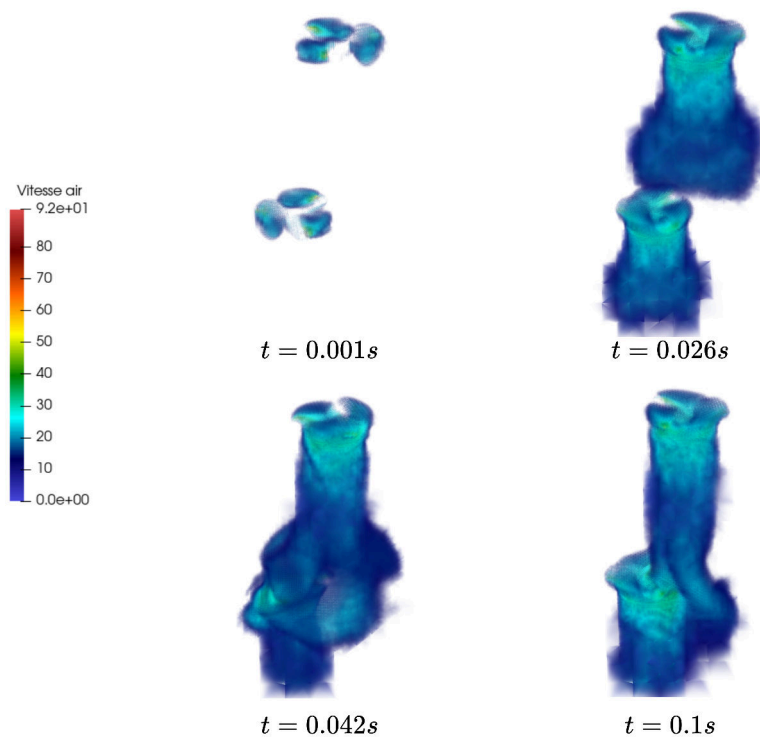


Figure 2.34: Représentation volumique du flux d'air généré par les deux hélices pour une inclinaison $\frac{\pi}{3}$. <https://youtu.be/EwUUabf-RZo>

2.6 Conclusion

Dans ce chapitre, un modèle théorique complet du drone a été établi par une approche énergétique (Euler-Lagrange) pour obtenir la dynamique du drone. Cette modélisation inclut les effets aérodynamiques ainsi qu'un bilan des forces et des couples que le système subit. Le bilan des forces nous a permis d'identifier nos entrées de commande afin de contrôler la dynamique du drone ainsi que la nature de ce dernier : suractionné ou sousactionné. Finalement, une étude CFD a été développée pour identifier les différents paramètres aérodynamiques du drone et aussi pour illustrer quelques interférences aérodynamiques entre une hélice en rotation et le flux d'air et entre deux hélices en rotation. Ces différentes interférences n'ont pas été prises en considération dans le modèle théorique, car cela rendrait le modèle très complexe. On verra dans la suite comment ces différents phénomènes non modélisés seront traités dans la partie contrôle.

3

Contrôle et observation du quadrirotor

Sommaire

3.1	Modèle d'observation et de contrôle	80
3.2	Stratégies d'observation	82
3.2.1	Filtre de Kalman linéaire à état étendu	83
3.2.2	Super Twisting Observer	85
3.3	Stratégies de commande	86
3.3.1	Contrôle en orientation	86
3.3.2	Contrôle en position	86
3.4	Conclusion	99

Liste des variables

Symbole	Description
$\mathbf{X} \in \mathbb{R}^6$	est le vecteur d'état
$\mathbf{X}^e \in \mathbb{R}^6$	est le vecteur d'état étendu avec la perturbation
$\mathbf{Y} \in \mathbb{R}^6$	est le vecteur des sorties
$\mathbf{Y}^e \in \mathbb{R}^6$	est le vecteur des sorties étendu
A^*	est la matrice d'état
B^*	est la matrice d'état
B_p^*	est la matrice des perturbations
C^*	est la matrice d'observation
D^*	est la matrice d'action directe
$\mathbf{w} \in \mathbb{R}^9$	est la variable aléatoire décrivant l'incertitude d'évolution de l'état
$\bar{\mathbf{w}} \in \mathbb{R}^9$	est la variable aléatoire décrivant l'incertitude dans la mesure
$Q \in \mathbb{R}^{9 \times 9}$	est la matrice de covariances de l'état
$R \in \mathbb{R}^{6 \times 6}$	est la matrice de covariances de mesure
$\boldsymbol{\varepsilon}^*$	est l'erreur de suivi
\mathbf{s}	est la variété de glissement
V	est la fonction de Lyapunov

CONTRÔLE ET OBSERVATION- Dans le chapitre précédent, un modèle théorique détaillé décrivant la dynamique du drone a été établi. Cependant, il est important de noter que ce modèle ne prend pas en compte tous les phénomènes physiques que le drone pourrait potentiellement rencontrer. En complément, l'objectif de ce chapitre est de développer des observateurs et des contrôleurs robustes capables de faire face aux incertitudes et aux lacunes du modèle. De manière spécifique, trois stratégies de commande sont proposées : l'ES-iLQG (Extended-State Integral Linear Quadratic Gaussian) et le MPC (Model Predictive Control) intégral, ainsi que le MPSMC (Model Predictive Sliding Mode Control). En ce qui concerne les stratégies d'observation, deux méthodes ont été adoptées. Tout d'abord, un filtre de Kalman à état étendu a été mis en œuvre pour estimer les états du système et les perturbations. En outre, un observateur Super Twisting (STO) a également été développé pour l'estimation des perturbations. L'ensemble de ces approches vise à renforcer la robustesse des systèmes de contrôle et d'observation du drone face aux diverses sources d'incertitude et aux phénomènes physiques non modélisés, contribuant ainsi à améliorer les performances opérationnelles du drone dans des situations réelles et dynamiques.

3.1 Modèle d'observation et de contrôle

Cette section est consacrée aux spécificités de la conception des contrôleurs et des observateurs pour les deux architectures de drones. Pour le Xtilt, on considérera seulement les dynamiques translationnelle et rotationnelle du corps principal. À cette fin, rappelons le système dynamique du drone (2.22) et (2.18).

$$\begin{pmatrix} M & 0 \\ 0 & W_\eta^\top I_B W_\eta \end{pmatrix} \begin{pmatrix} \ddot{\xi} \\ \ddot{\eta} \end{pmatrix} + \begin{pmatrix} 0 & 0 \\ 0 & C(\eta, \dot{\eta}) \end{pmatrix} \begin{pmatrix} \dot{\xi} \\ \dot{\eta} \end{pmatrix} + \mathcal{G} = \begin{pmatrix} F^O \\ \tau^O \end{pmatrix} \quad (3.1)$$

Pour s'approcher au mieux de la réalité des choses, on considère certaines hypothèses sur le modèle dynamique établi précédemment,

- **H1:** On s'intéresse seulement à la dynamique de translation, tandis que la dynamique de rotation est supposée stabilisée par un contrôle hiérarchique classique.
- **H2:** La matrice d'inertie M est supposée partiellement connue. Cela permet de réécrire M sous la forme de

$$M = M_n + M_\Delta \quad (3.2)$$

où M_n et M_Δ représentent respectivement les valeurs paramétriques nominales et incertaines.

- **H3:** On considère également le vecteur Δ_ρ regroupant les dynamiques non modélisées (couplages croisés translationnels/rotationnels, déformation structurelle, dés-alignement CoG).
- **H4:** La position translationnelle, ξ , et la vitesse, $\dot{\xi}$, sont supposées mesurables mais polluées par du bruit gaussien. .
- **H5:** On considère également qu'un vecteur de force exogène Δ_e provenant des fluctuations du vent atmosphérique s'exerce sur les rotors et sur le châssis.

En se basant sur ces hypothèses, on peut reformuler le modèle (3.1) de la manière suivante :

$$(M_n + M_\Delta)\ddot{\xi} + \mathcal{G} + \Delta_\rho = F_r^O + F_a^O + \Delta_e \quad (3.3)$$

Ainsi, on regroupe les incertitudes non paramétriques et endogènes dans un vecteur de perturbation unique.

$$\Delta_\xi = -M_\Delta \ddot{\xi} - \Delta_\rho + F_a^O + \Delta_e \quad (3.4)$$

Menant à,

$$\ddot{\xi} = M_n^{-1} (F_r^O - \mathcal{G} + \Delta_\xi) \quad (3.5)$$

On considère que la perturbation $\Delta_\xi \in \mathbb{R}^3$ est bornée, telle qu'il existe une constante $\Delta_{\xi_{max}} \in \mathbb{R}^+$ qui satisfait $\|\Delta_\xi\| \leq \Delta_{\xi_{max}} \forall t \geq 0$. En considérant l'entrée de contrôle $F_r^O = \mathcal{G} + \mathbf{u}$ mène à,

$$\ddot{\xi} = M_n^{-1} (\mathbf{u} + \Delta_\xi) \quad (3.6)$$

On définit $\mathbf{X} = (\xi, \dot{\xi})^\top = (\xi_1, \xi_2)^\top$ le vecteur de l'état actuel, on peut donc écrire,

$$\begin{cases} \dot{\xi}_1 = \xi_2 \\ \dot{\xi}_2 = M_n^{-1}(\mathbf{u} + \Delta_\xi) \end{cases} \quad (3.7)$$

Ainsi la représentation d'état du système (3.5) à temps continu s'écrit de la manière suivante,

$$\begin{cases} \dot{\mathbf{X}}(t) = A\mathbf{X}(t) + B\mathbf{u}(t) + B_p\Delta_\xi(t) \\ \mathbf{Y}(t) = C\mathbf{X}(t) + D\mathbf{u}(t) \end{cases} \quad (3.8)$$

Les différentes variables sont définies de la manière suivante,

- $A \in \mathbb{R}^{6 \times 6}$ est la matrice d'état
- $B \in \mathbb{R}^{6 \times 3}$ est la matrice de commande
- $B_p \in \mathbb{R}^{6 \times 3}$ est la matrice des perturbations
- $\mathbf{Y}(t) \in \mathbb{R}^6$ représente les sorties
- $C \in \mathbb{R}^{6 \times 6}$ est la matrice d'observation
- D est la matrice d'action directe

3.2 Stratégies d'observation

L'observabilité traduit la propriété d'un système où l'on peut reconstruire le vecteur d'état complet à partir de la connaissance de variables d'entrée et de sortie et une mesure partielle de quelques variables d'état par le biais de capteurs. Par exemple, pour notre cas en partant du modèle dynamique décrit dans l'équation (3.5), il est possible d'estimer les perturbations en construisant un observateur. Il existe dans la littérature différentes approches permettant de construire un observateur de perturbation. Le développement des observateurs dits robuste trouve son origine dans les années 1960 et 1970 pour remédier aux problèmes de sensibilité des observateurs d'état classiques (Luenberger). Parmi ces techniques d'estimation de perturbation on retrouve (UIO) pour Unknown Input Observer, le Disturbance Observer (DOb) [100], et plus récemment le (ESO) Extended State Observer [101] ou l'on peut estimer à la fois les perturbations externes et l'état du système. Le détail de ces différentes approches et leurs utilisations sont détaillés dans [102]. Il est aussi possible de construire un estimateur en se basant sur des modèles stochastiques type Filtre de Kalman. On retrouve dans la littérature plusieurs exemples de cas d'application d'un filtre de Kalman à état étendu pour à la fois filtrer le bruit gaussien des mesures et estimer les perturbations externes [103], [104]. Il existe aussi d'autres types d'observateurs de perturbation basés mode glissant (STO) pour Super Twisting Observer [105]. Dans [62] les performances de trois différents observateurs ont été évaluées (Momentum-based Observer, Hybrid Observer et Extended Kalman Filter) par simulation dans différentes conditions. Selon [62], le filtre de Kalman étendu présente les meilleures performances en termes de robustesse au bruit et de précision d'estimation.

3.2.1 Filtre de Kalman linéaire à état étendu

Le schéma d'estimation des perturbations est brièvement décrit ci-après. Dans cette perspective, considérons le modèle (3.8) qui inclut les perturbations $\Delta_\xi = (\rho_x, \rho_y, \rho_z)^\top$, il est supposé qu'aucune connaissance préalable sur ces perturbations n'est disponible. Toutefois, nous considérons que la perturbation présente une dynamique lente et varie peu dans le temps, on peut donc modéliser cette dernière par un processus de marche aléatoire, c'est-à-dire,

$$\dot{\Delta}_\xi = (\omega_x, \omega_y, \omega_z)^\top \quad (3.9)$$

En effet, on peut définir une autre représentation d'état pour le système différente de celle introduite dans l'équation (3.8), en considérant les perturbations comme des états du système. On peut alors définir le vecteur d'état étendu de la manière suivante,

$$\mathbf{X}^e = \left(\xi \quad \dot{\xi} \quad \Delta_\xi \right)^\top \quad (3.10)$$

Ainsi le modèle dynamique étendu est donné par,

$$\begin{cases} \dot{\mathbf{X}}^e(t) = A^e \mathbf{X}^e(t) + B^e \mathbf{u}(t) + \mathbf{w}(t) \\ \mathbf{Y}^e(t) = C^e \mathbf{X}^e(t) + \bar{\mathbf{w}}(t) \end{cases} \quad (3.11)$$

dont les variables sont définies de la manière suivante,

- $A^e \in \mathbb{R}^{9 \times 9}$ est la matrice d'état définie par,

$$A^e = \left(\begin{array}{c|c} & \begin{pmatrix} 0 & 0 & 0 \\ 1 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 1 \end{pmatrix} \\ \hline 0_{3 \times 6} & 0_{3 \times 3} \end{array} \right) \quad (3.12)$$

- $B^e \in \mathbb{R}^{9 \times 3}$ est la matrice de commande définie par,

$$B^e = \begin{pmatrix} B_{6 \times 3} \\ 0_{3 \times 3} \end{pmatrix} \quad (3.13)$$

- $C^e \in \mathbb{R}^{6 \times 9}$ est la matrice d'observation définie par,

$$C^e = \left(C_{6 \times 6} \mid 0_{6 \times 3} \right) \quad (3.14)$$

- $\mathbf{w}(t) \in \mathbb{R}^9$ est une variable aléatoire décrivant l'incertitude dans l'évolution de l'état du système. Elle est supposée issue d'une distribution gaussienne (normale) multidimensionnelle de moyenne nulle. On considère $Q \in \mathbb{R}^{9 \times 9}$ la matrice des covariances et on définit ainsi,

$$\mathbf{w}(t) \sim \mathcal{N}(0, Q) \quad (3.15)$$

- $\bar{\mathbf{w}}(t) \in \mathbb{R}^6$ est une variable aléatoire décrivant l'incertitude dans la mesure de l'état du système. Elle est supposée issue d'une distribution gaussienne (normale) multidimensionnelle de moyenne nulle. On considère $R \in \mathbb{R}^{6 \times 6}$ matrice des covariances et on définit ainsi,

$$\bar{\mathbf{w}}(t) \sim \mathcal{N}(0, R) \quad (3.16)$$

Cependant, le filtre de Kalman est implémenté en temps discret k . La version à temps discret de (3.11) est alors donnée par,

$$\begin{cases} \mathbf{X}_{k+1}^e = \bar{A}^e \mathbf{X}_k^e + \bar{B}^e \mathbf{u}_k + \mathbf{w}_k \\ \mathbf{Y}_k^e = \bar{C}^e \mathbf{X}_k^e + \bar{\mathbf{w}}_k \end{cases} \quad (3.17)$$

Les estimations de l'état produites par le filtre de Kalman minimisent l'erreur moyenne quadratique d'estimation en fonction d'une séquence d'observations donnée,

$$\hat{\mathbf{X}}_{i|j}^e \triangleq E \left[\mathbf{X}_{i|j}^e \mid \mathbf{Y}_1^e, \dots, \mathbf{Y}_j^e \right] \quad (3.18)$$

La variance de l'estimation est définie comme l'erreur quadratique moyenne de l'estimation,

$$P_{i|j} \triangleq E \left[\left(\mathbf{X}_{i|j}^e - \hat{\mathbf{X}}_{i|j}^e \right) \left(\mathbf{X}_{i|j}^e - \hat{\mathbf{X}}_{i|j}^e \right)^\top \right] \quad (3.19)$$

On notera, pour la suite, les variables de la manière suivante, $\hat{\mathbf{X}}_{k|k}^e$ signifie l'estimation de l'état pour un temps k considérant toutes les informations jusqu'au moment k . Le filtre de Kalman procède ensuite récursivement en deux étapes,

- **Prédiction** : À l'instant $k+1$, une prédiction $\hat{\mathbf{X}}_{k+1|k}^e$ et sa covariance $P_{k+1|k}$ sont calculées,

$$\hat{\mathbf{X}}_{k+1|k}^e = \bar{A}^e \hat{\mathbf{X}}_{k|k}^e + \bar{B}^e \mathbf{u}_k \quad (3.20)$$

$$P_{k+1|k} = \bar{A}^e P_{k|k} \bar{A}^{e\top} + Q \quad (3.21)$$

- **Actualisation des mesures :** À l'instant $k + 1$, une mesure est faite \mathbf{Y}_k^e ainsi que la mise à jour de la covariance estimée et l'estimation de l'état.

$$K_k = P_{k+1|k} \bar{C}^e (R + \bar{C}^e P_{k+1|k} \bar{C}^e)^{-1} \quad (3.22)$$

$$P_{k+1|k+1} = P_{k+1|k} - K_k (R + \bar{C}^e P_{k+1|k} \bar{C}^e) K_k^\top \quad (3.23)$$

$$\hat{\mathbf{X}}_{k+1|k+1}^e = \hat{\mathbf{X}}_{k+1|k}^e + K_k (\mathbf{Y}_k^e - \bar{C}^e \hat{\mathbf{X}}_{k+1|k}^e) \quad (3.24)$$

On obtient ainsi l'état estimé qui contient une estimation des perturbations,

$$\hat{\mathbf{X}}^e = \begin{pmatrix} \hat{\xi} & \dot{\hat{\xi}} & \hat{\Delta}_\xi \end{pmatrix}^\top \quad (3.25)$$

Il faut noter que le système (3.11) est observable puisque,

$$\text{rang} \begin{bmatrix} C^e & C^e A^e & C^e A^{e2} & \dots & C^e A^{e(n-1)} \end{bmatrix}^\top = 9, \quad n = 9 \quad (3.26)$$

3.2.2 Super Twisting Observer

Le Super Twisting Observer (STO) rapporté dans la littérature permet soit d'estimer les perturbations agissant sur le système ou/et l'état du système. Ce dernier garantit une convergence en temps fini de l'erreur d'estimation. En considérant le modèle dynamique introduit par l'équation (3.7) :

$$\begin{cases} \dot{\xi}_1 = \xi_2 \\ \dot{\xi}_2 = M_n^{-1} (\mathbf{u} + \Delta_\xi) \end{cases} \quad (3.27)$$

La dynamique de l'estimation du STO déjà reportée dans [106]–[108] est donnée par,

$$\begin{cases} \dot{\hat{\xi}}_1 = \hat{\xi}_2 + \mathbf{z}_1 \\ \dot{\hat{\xi}}_2 = M_n^{-1} (\mathbf{u} + \mathbf{z}_2) \end{cases} \quad (3.28)$$

Où \mathbf{z}_1 et \mathbf{z}_2 sont les termes de correction définis par,

$$\begin{cases} \mathbf{z}_1 = k_1 \text{sign}^* (\xi_1 - \hat{\xi}_1)^\top |\xi_1 - \hat{\xi}_1|^{\frac{1}{2}} \\ \mathbf{z}_2 = k_2 \text{sign}^* (\xi_1 - \hat{\xi}_1) \end{cases} \quad (3.29)$$

La fonction sign^* est définie pour un vecteur de n éléments, comme un vecteur de n élément

où pour chaque élément on prend la fonction sign.

On peut alors décrire la dynamique de l'erreur d'estimation de la manière suivante,

$$\begin{cases} \dot{\tilde{\xi}}_1 &= \dot{\xi}_1 - \hat{\dot{\xi}}_1 = -k_1 \text{sign}^*(\tilde{\xi}_1)^\top |\tilde{\xi}_1|^{\frac{1}{2}} + \tilde{\xi}_2 \\ \dot{\tilde{\xi}}_2 &= \dot{\xi}_2 - \hat{\dot{\xi}}_2 = M_n^{-1} (-k_2 \text{sign}^*(\tilde{\xi}_1) + \Delta_\xi) \end{cases} \quad (3.30)$$

Il a déjà été établi dans [107], [108] que pour des perturbations supposées bornées $\|\Delta_\xi\| \leq \Delta_{\xi_{max}}$ et pour $k_1 = 1.5\sqrt{\Delta_{\xi_{max}}}$ et $k_2 = 1.1\Delta_{\xi_{max}}$, les deux erreurs d'estimation $\tilde{\xi}_1$ et $\tilde{\xi}_2$ convergent vers zéro en un temps fini. Ainsi l'estimation de la perturbation est donnée par :

$$\hat{\Delta}_\xi = 1.1\Delta_{\xi_{max}} M_n^{-1} \text{sign}^*(\xi_1 - \hat{\xi}_1) \quad (3.31)$$

3.3 Stratégies de commande

3.3.1 Contrôle en orientation

En ce qui concerne le contrôleur d'attitude, un simple contrôleur PID (Proportionnel Intégral Dérivé) classique est utilisé et s'exprime comme suit

$$\begin{cases} u_\phi = k_d^\phi(\dot{\phi}_d - \dot{\phi}) + k_p^\phi(\phi_d - \phi) + k_i^\phi \int (\phi_d - \phi) \\ u_\theta = k_d^\theta(\dot{\theta}_d - \dot{\theta}) + k_p^\theta(\theta_d - \theta) + k_i^\theta \int (\theta_d - \theta) \\ u_\psi = k_d^\psi(\dot{\psi}_d - \dot{\psi}) + k_p^\psi(\psi_d - \psi) + k_i^\psi \int (\psi_d - \psi) \end{cases} \quad (3.32)$$

sur la base de l'erreur entre les points de consigne souhaités et les valeurs d'attitude mesurées, la simplicité du PID permet de faire tourner le contrôleur en attitude à très haute fréquence. Dans certains cas, le contrôleur en attitude servira comme boucle interne pour le contrôleur en position, il est essentiel de garantir une rapidité de convergence de la boucle interne.

3.3.2 Contrôle en position

Pour cette partie, les différentes stratégies de contrôle en position seront développées, en partant du modèle 3.5,

$$\ddot{\xi} = M_n^{-1} (\mathbf{F}_r^O - \mathcal{G} + \Delta_\xi) \quad (3.33)$$

Comme le but recherché est de réaliser un contrôle en suivi de trajectoire, on définit l'état désiré par $\mathbf{X}^d = (\xi^d, \dot{\xi}^d)^\top$, on définit ainsi l'erreur de suivi par $\boldsymbol{\varepsilon} = \mathbf{X}^d - \mathbf{X} = (\boldsymbol{\varepsilon}_1, \boldsymbol{\varepsilon}_2)^\top$, on peut donc déduire la dynamique d'erreur,

$$\begin{aligned}\dot{\boldsymbol{\varepsilon}}_1 &= \boldsymbol{\varepsilon}_2 \\ \dot{\boldsymbol{\varepsilon}}_2 &= M_n^{-1} \left(M_n \ddot{\boldsymbol{\xi}}^d - \mathbf{F}_r^O + \mathcal{G} - \Delta_\xi \right)\end{aligned}\quad (3.34)$$

En considérant comme entrée de commande $\mathbf{F}_r^O = M_n \ddot{\boldsymbol{\xi}}^d + \mathcal{G} + \boldsymbol{\nu}$, on obtient ainsi,

$$\begin{aligned}\dot{\boldsymbol{\varepsilon}}_1 &= \boldsymbol{\varepsilon}_2 \\ \dot{\boldsymbol{\varepsilon}}_2 &= -M_n^{-1} (\boldsymbol{\nu} + \Delta_\xi)\end{aligned}\quad (3.35)$$

Ainsi, la représentation d'état de la dynamique d'erreur est donnée par,

$$\begin{cases} \dot{\boldsymbol{\varepsilon}}(t) &= A\boldsymbol{\varepsilon}(t) - B\boldsymbol{\nu}(t) - B_p\Delta_\xi(t) \\ \mathbf{Y}_\varepsilon(t) &= C\boldsymbol{\varepsilon}(t) \end{cases}\quad (3.36)$$

ES-iLQG

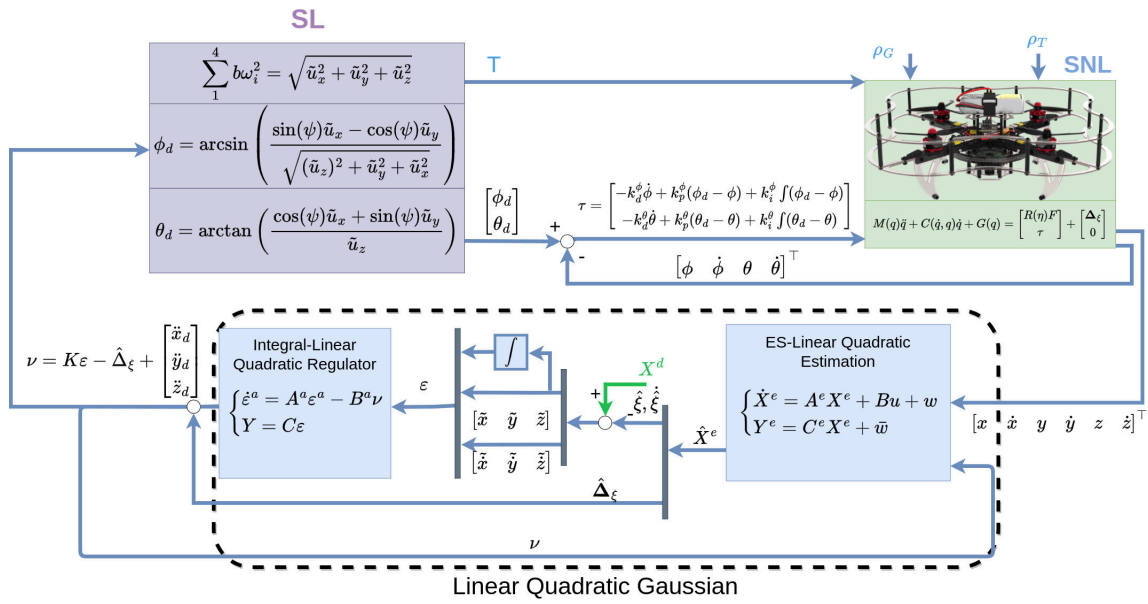


Figure 3.1: Description de l'architecture de contrôle ES-iLQG (Extended State Integral Linear Quadratic Gaussian).

Dans cette section, nous présentons le premier schéma de contrôle proposé pour stabiliser le mouvement en translation figure (3.1). L'intégral LQR (Linear Quadratic Regulator) permet un comportement robuste face aux incertitudes endogènes (paramétriques) et permet de suivre les trajectoires souhaitées tout en respectant des pénalités spécifiques sur l'entrée de contrôle et le vecteur d'état. Ce formalisme est largement utilisé pour les systèmes linéaires ou linéarisés afin de concevoir un contrôleur optimal. On retrouve par exemple dans la littérature [109] un contrôleur basé LQR dont le modèle dynamique du quadrirotor est linéarisé à chaque pas

pour calculer les gains. Dans [110] le modèle du quadrirotor est linéarisé pour concevoir le LQR et le contrôleur en position. Pour compenser les perturbations externes, une partie intégrale est ajoutée qui augmente l'état du système en intégrant dans sa totalité. Dans la suite, nous allons introduire le contrôleur utilisé. La stratégie globale considère un état auxiliaire $\boldsymbol{\varepsilon}_0 = \int^{t_c} \boldsymbol{\varepsilon}_1 dt$ (action intégrale), on introduit ainsi la variable d'état augmentée définie par $\boldsymbol{\varepsilon}^a = (\boldsymbol{\varepsilon}_0, \boldsymbol{\varepsilon}_1, \boldsymbol{\varepsilon}_2)^\top \in \mathbb{R}^9$, en se basant sur le modèle (3.36) et en négligeant la perturbation, la représentation d'état du système augmenté est donnée par,

$$\dot{\boldsymbol{\varepsilon}}^a(t) = A^a \boldsymbol{\varepsilon}^a(t) - B^a \boldsymbol{v}(t) \quad (3.37)$$

$A^a \in \mathbb{R}^{9 \times 9}$ est la matrice d'état pour le système augmenté, $B^a \in \mathbb{R}^{9 \times 3}$ est la matrice de commande du système augmenté. Le LQR fournit la matrice de gain optimale K de la loi de commande de sorte qu'une fonction objective J soit minimisée.

$$\boldsymbol{v} = K \boldsymbol{\varepsilon}^a + \ddot{\boldsymbol{\xi}}^d \quad (3.38)$$

$$J(\boldsymbol{\varepsilon}^a, \boldsymbol{v}) = \int_0^\infty \boldsymbol{\varepsilon}^{a\top} \tilde{Q} \boldsymbol{\varepsilon}^a + \boldsymbol{v}^\top \tilde{R} \boldsymbol{v} \quad (3.39)$$

$\tilde{Q} \in \mathbb{R}^{9 \times 9}$ et $\tilde{R} \in \mathbb{R}^{3 \times 3}$ sont les matrices de coût pour l'erreur d'état augmenté et la commande respectivement, le gain optimal est donné par,

$$K = \tilde{R}^{-1} B^{a\top} S \quad (3.40)$$

où S est obtenu en résolvant l'équation algébrique continue de Riccati :

$$0 = A^{a\top} S + S A^a - S B^a \tilde{R}^{-1} B^{a\top} S + \tilde{Q} \quad (3.41)$$

Les matrices susmentionnées vérifient les conditions d'observabilité et de contrôlabilité pour garantir l'existence d'une solution unique positive-définie S à l'équation de Riccati, $(A^a, \tilde{Q}^{\frac{1}{2}})$ doit être observable et (A^a, B^a) contrôlable. Pour compenser les perturbations, l'estimation de la perturbation est injectée dans la commande de la manière suivante:

$$\boldsymbol{v} = K \boldsymbol{\varepsilon}^a + \ddot{\boldsymbol{\xi}}^d - \hat{\Delta}_{\boldsymbol{\xi}} \quad (3.42)$$

Dans le cas du *LQR constraint*, des contraintes de performance sont considérées pour les états du système et/ou pour le contrôleur. L'objectif consiste à minimiser la fonction coût (3.39) tout en respectant ces spécifications de performance, c'est-à-dire,

$$\min J(\boldsymbol{\varepsilon}^a, \boldsymbol{\nu}) \quad (3.43)$$

sujet à,

$$\begin{aligned} \dot{\boldsymbol{\varepsilon}}^a(t) &= A^a \boldsymbol{\varepsilon}^a(t) - B^a \boldsymbol{\nu}(t) \\ \boldsymbol{\varepsilon}_{min}^a &\leq \boldsymbol{\varepsilon}^a(t) \leq \boldsymbol{\varepsilon}_{max}^a \quad \forall t \\ \boldsymbol{\nu}_{min} &\leq \boldsymbol{\nu}(t) \leq \boldsymbol{\nu}_{max} \quad \forall t \end{aligned} \quad (3.44)$$

Pour s'assurer que les entrées de contrôle et les états du système restent dans des limites prédéfinies. En raison de l'horizon temporel infini des contraintes et des décisions générées, la résolution du contrôleur optimal devient difficile à calculer. Au lieu de cela, il est possible d'utiliser des techniques d'approximation pour obtenir une solution réalisable qui satisfasse les contraintes et se rapproche de la commande optimale dans le cadre du problème LQR contraint. Dans ce sens, diverses méthodes telles que les algorithmes d'optimisation numérique, la commande prédictive (basée modèle) (MPC Model Predictive Control) ou les techniques de programmation linéaire peuvent être utilisées.

Commande prédictive

Dans cette partie, nous allons introduire quelques concepts de base pour la commande prédictive basée modèle. Le MPC introduit une fenêtre temporelle fuyante, ou un horizon temporel fini, le problème de contrôle optimal est formulé comme un problème de programmation quadratique (QP) avec un nombre fini (N) de variables de décision et de contraintes. Dans ce cas, la solution optimale est mise à jour fréquemment, alors que dans le LQR, la solution reste inchangée tout au long de l'horizon temporel. L'objectif est de minimiser une fonction coût tout en respectant les spécifications préalables sur la dynamique du système et des états, la résolution se fait à temps discret, on introduit ainsi le modèle discrétisé de (3.37)

$$\dot{\boldsymbol{\varepsilon}}_k^a = \bar{A}^a \boldsymbol{\varepsilon}_k^a - \bar{B}^a \boldsymbol{\nu}_k \quad (3.45)$$

À savoir la fonction coût est définie de la manière suivante:

$$\min_{\{\boldsymbol{\nu}_k, \dots, \boldsymbol{\nu}_{N-1}\}} \frac{1}{2} \sum_0^N (\boldsymbol{\nu}_k^\top \bar{R} \boldsymbol{\nu}_k + \boldsymbol{\varepsilon}_k^{a\top} \tilde{Q} \boldsymbol{\varepsilon}_k^a) \quad (3.46)$$

sujet à,

$$\begin{aligned}
 \dot{\boldsymbol{\varepsilon}}_{k+1}^a &= \bar{\mathbf{A}}^a \boldsymbol{\varepsilon}_k^a - \bar{\mathbf{B}}^a \boldsymbol{\nu}_k \\
 \boldsymbol{\varepsilon}_{min}^a &\leq \boldsymbol{\varepsilon}_k^a \leq \boldsymbol{\varepsilon}_{max}^a \quad \forall t \\
 \boldsymbol{\nu}_{min} &\leq \boldsymbol{\nu}_k \leq \boldsymbol{\nu}_{max} \quad \forall t
 \end{aligned} \tag{3.47}$$

Ce profil de performance est garanti pour l'ensemble de l'horizon temporel ($k \in \{0, N\}$). Avec un horizon temporel fini et un nombre fini de variables de décision et de contraintes, le problème peut être résolu à l'aide de méthodes QP (Quadratic programming) standard. On retrouve dans la littérature quelques implémentations d'un MPC pour contrôler des drones, dans [111] un simple MPC a été implémenté pour contrôler en orientation et position le drone, dans [112] un MPC avec une action intégrale et les contraintes d'actionnement sont pris en considération dans la solution optimale. Il convient de noter que les formulations susmentionnées supposent un système linéaire non perturbé. Cependant, les scénarios du monde réel comportent des incertitudes (internes et non paramétriques). C'est pourquoi, dans la section suivante, ces effets parasites sont explicitement pris en compte dans la formulation du MPC.

La formulation des lois de contrôle s'articule autour de deux perspectives incontournables. La première consiste à garantir la résilience du contrôleur face aux perturbations externes qu'un système peut rencontrer, tout en tenant compte des incertitudes du modèle. Le second paradigme est centré sur l'optimisation. Les systèmes mécatroniques, sans exception, sont confrontés aux limites imposées par leurs actionneurs en termes de puissance et sont en outre limités par l'autonomie de la batterie. Ces contraintes exigent une attention particulière.

La création d'une loi de commande capable d'harmoniser les attributs de robustesse et d'optimisation revêt une importance significative. Cela est particulièrement vrai pour les drones, où les ressources en énergie sont limitées et où l'influence des perturbations externes est importante.

Dans la littérature, de nombreuses approches ont émergé qui intègrent des lois de contrôle robustes avec les capacités d'un contrôleur prédictif basé modèle (MPC). Ce qui distingue le MPC des autres lois de contrôle optimales, comme le LQR, c'est sa capacité inhérente à prendre en compte les contraintes du système.

Une approche intéressante dans ce domaine est le contrôle prédictif en mode glissant (SMPC). Cette approche tire parti d'une combinaison d'attributs : l'aptitude à gérer les contraintes du système et à optimiser les actions de contrôle (MPC), ainsi que la robustesse inhérente à la commande en mode glissant, capable de gérer les perturbations.

Un aperçu complet des différentes approches de la commande prédictive à mode glissant (SMPC) est fourni dans [113]. Ces diverses approches peuvent être classées en deux groupes principaux en fonction de leur utilisation :

- **Mode de convergence** : La MPC est utilisée pendant la phase de convergence de la commande en mode glissant (SMC). Cette phase se produit lorsque l'état du système est en transition vers la surface de glissement, dans le but de l'atteindre. Dans la première catégorie d'approches, une méthode notable est la Dual-Mode Control Reaching Law (DMCRL), décrite dans [114]. La loi de contrôle de cette approche est résumée comme suit :

$$u_k = \begin{cases} u_k^{mpc}, & x_k \notin \mathcal{R}_i \\ u_k^{smc}, & x_k \in \mathcal{R}_i \end{cases} \quad (3.48)$$

Ici, \mathcal{R}_i représente la région de glissement, où le système fonctionne en mode de glissement. La commande appliquée au système est désignée par u_k . Lorsque l'état du système, représenté par x_k , se situe en dehors de la zone de glissement \mathcal{R}_i , la commande u_k^{mpc} est dérivée du cadre de la commande prédictive basée modèle (MPC). Inversement, lorsque le système se trouve dans la zone de glissement \mathcal{R}_i , la commande u_k^{smc} est déterminée à l'aide des principes de la commande en mode glissant (SMC).

- **Mode de glissement** : implique l'utilisation du MPC pendant la phase de glissement du processus de contrôle. Dans cette phase, le système fonctionne à proximité de la surface de glissement et le MPC est utilisée pour maintenir la stabilité et réguler le comportement du système dans cette région. Un exemple illustratif de cette approche est le contrôle prédictif à mode glissant intégral (ISMPC) présenté dans [115]. Dans l'ISMPC, la loi de contrôle est exprimée comme suit :

$$u_k = u_k^{mpc} + u_k^{smc} \quad (3.49)$$

Cette formulation combine les avantages des deux paradigmes de contrôle. Plus précisément, elle intègre un cadre MPC, désigné par u_k^{mpc} , qui assure des actions de contrôle optimales et contraintes, avec un SMC intégral, représenté par u_k^{smc} . La composante MPC gère l'optimisation et le respect des contraintes du système, tandis que le terme intégral du mode glissant traite des perturbations auxquelles le système est exposé.

Dans le but de renforcer la robustesse du contrôle prédictif basé modèle (MPC), diverses approches sont disponibles. Ces dernières ne modifient pas directement la formulation de base du MPC, mais introduisent des éléments supplémentaires pour améliorer sa résilience. L'une de ces approches consiste à intégrer une composante intégrale [112], visant à compenser l'impact des perturbations sur le système. Étant donné que le contrôleur MPC repose sur un modèle, certaines stratégies alternatives se concentrent sur l'amélioration de la précision du modèle utilisé comme référence pour le contrôleur. La commande prédictive non linéaire (NMPC) [116] est l'une de ces approches, elle repose sur un modèle non linéaire permettant de capturer de manière

plus précise la dynamique des systèmes. De plus, pour faire face aux perturbations externes, certaines techniques impliquent une estimation en temps réel des perturbations [117]. Dans ce cadre, le modèle linéaire nominal du système est complété par l'estimation des perturbations, et cela est pris en compte dans la commande prédictive.

Sur la base des recherches précédentes, l'approche que nous proposons est la suivante. Nous insistons sur la nécessité de maximiser la fréquence de contrôle et de veiller à ce que le modèle dynamique corresponde étroitement à la dynamique réelle du système en question. Cette importance découle de la compréhension du fait que des délais excessifs peuvent compromettre la robustesse du contrôleur, et qu'un modèle qui s'écarte de manière significative du comportement réel peut introduire une disparité substantielle entre la solution optimale calculée et la solution optimale réelle.

Notre proposition s'inscrit dans cette perspective : le modèle utilisé par la commande prédictive (MPC) est affiné par l'incorporation d'une estimation des perturbations externes. Des stratégies de contrôle robustes sont employées à un temps d'échantillonnage inférieur au temps nécessaire pour résoudre le problème de contrôle optimal. En termes plus simples, l'action du contrôleur appliquée correspond à la loi de contrôle robuste établie. La composante optimale intervient dans le calcul des gains de contrôle, garantissant le respect des contraintes et le maintien de la stabilité du système. De plus, ce processus d'optimisation prend en compte le modèle du drone, corrigé avec la perturbation estimée.

Le schéma ci-dessous illustre succinctement les composants clés de l'approche que nous proposons figure (3.2) :

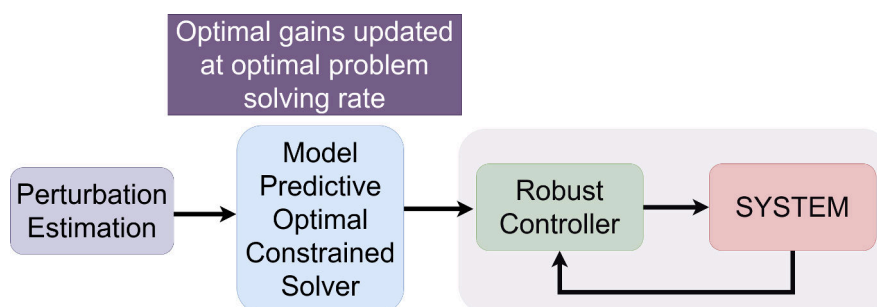


Figure 3.2: Schéma de l'approche proposée basée MPC.

Cette approche vise à trouver un équilibre entre robustesse et optimalité, en offrant une solution pratique qui combine la compensation des perturbations en temps réel avec une action de contrôle précise, améliorant ainsi la performance et la résilience du système de contrôle.

Nous allons dans la suite introduire deux approches basées MPC illustrées en détails dans la figure (3.3).

- **Intégrale MPC :** Pour cette approche, on enrichit le modèle pour le MPC par l'estimation

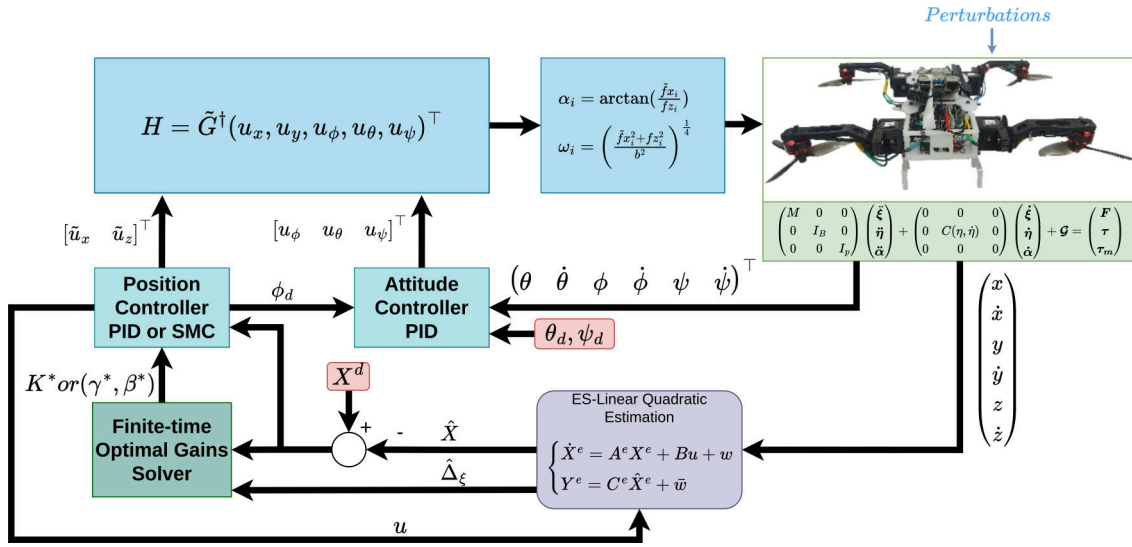


Figure 3.3: Schéma détaillé de l'approche proposée basée MPC.

de la perturbation $\hat{\Delta}_\xi$, ce dernier vise à améliorer le modèle de référence MPC, tandis que l'état intégral assure la robustesse contre les écarts d'estimation entre la valeur réelle Δ_ξ et l'estimation respective $\hat{\Delta}_\xi$. L'inclusion de l'estimation des perturbations modifie (3.45) conduisant à :

$$\boldsymbol{\varepsilon}_{k+1}^a = \bar{A}^a \boldsymbol{\varepsilon}_k^a - \bar{B}^a \mathbf{v}_k - \bar{B}_p^a \Delta_\xi \quad (3.50)$$

$\bar{B}_p^a \in \mathbb{R}^{9 \times 3}$, en partant du modèle dynamique précédent, nous pouvons formuler un problème de programmation quadratique (QP) comme suit :

$$\min_{\{\mathbf{v}_k, \dots, \mathbf{v}_{N-1}\}} \frac{1}{2} \sum_0^N (\mathbf{v}_k^\top \tilde{R} \mathbf{v}_k + \boldsymbol{\varepsilon}_k^{a\top} \tilde{Q} \boldsymbol{\varepsilon}_k^a) \quad (3.51)$$

sujet à,

$$\begin{aligned} C1: & \boldsymbol{\varepsilon}_{k+1}^a = \bar{A}^a \boldsymbol{\varepsilon}_k^a - \bar{B}^a \mathbf{v}_k - \bar{B}_p^a \Delta_\xi \quad \forall k \in \{0, N\} \\ C2: & \mathbf{v}_k = \tilde{K} \boldsymbol{\varepsilon}_k^a \quad \forall k \in \{0, N\} \\ C3: & K_{min} \leq \tilde{K} \leq K_{max} \\ C4: & \boldsymbol{\varepsilon}_0^a = \hat{\boldsymbol{\varepsilon}}^a(t) \\ C5: & \Delta_\xi = \hat{\Delta}_\xi \end{aligned} \quad (3.52)$$

les matrices $\tilde{Q} > 0$ et $\tilde{R} > 0$ pondèrent respectivement les écarts de coût de l'état et le coût d'actionnement, respectivement. La matrice des gains est définie comme suit

$$\tilde{K} = \text{diag}(k_{i_x}, k_{i_y}, k_{i_z}, k_{p_x}, k_{p_y}, k_{p_z}, k_{d_x}, k_{d_y}, k_{d_z}). \quad (3.53)$$

En outre, l'architecture de contrôle est dotée d'un schéma MPC d'optimisation multi-objectifs. Les contraintes sélectionnées (3.52) sont principalement destinées à (C1) définir le modèle de référence à des fins de prédiction, dont la structure englobe les perturbations, (C2) calculer les gains optimaux tout en respectant (C3) une plage de gain prédéfinie, (C4) est l'erreur initiale, nous prenons la dernière valeur connue de l'état estimée à l'instant t , et enfin (C5) pour mettre à jour (états filtrés et estimation des perturbations) le modèle simultanément via un ES-LKF, la loi ν_k , qui est appliquée au système à une fréquence plus élevée avec des gains obtenus par optimisation.

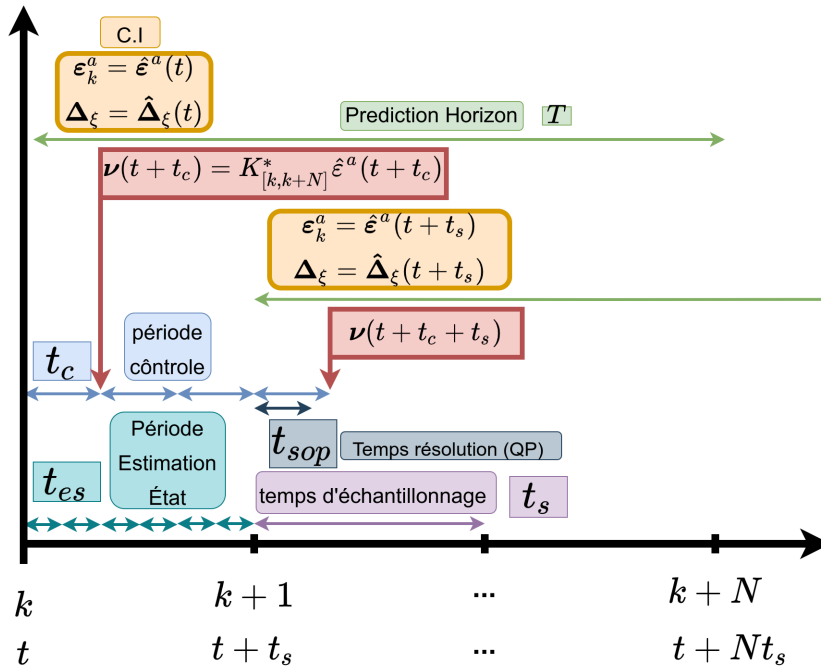


Figure 3.4: Description du schéma de contrôleur intégral MPC.

Le diagramme temporel représenté par la figure (3.4) détaille la séquence temporelle de la sortie MPC et de l'estimation des états/perturbations. La solution optimale est calculée à une fréquence de $f_s = 1/t_s$, en fixant simultanément les conditions initiales pour ϵ_k^a et Δ_ξ aux valeurs mesurées (ou aux valeurs estimées pour Δ_ξ), qui est supposée constante pendant l'horizon T , tandis que la commande ν est maintenue à une fréquence de $f_c = 1/t_c$, qui est plus grande que celle du QP.

Pendant l'intervalle $[k, k+1]$, la sortie MPC $\nu(t) = K_{[k,k+N]}^* \hat{\epsilon}^a(t)$ est appliquée ou $\hat{\epsilon}^a$ représente l'erreur de suivi en considérant l'état filtré/estimé et où $K_{[k,k+N]}^*$ représente la solution du problème d'optimisation (3.51) obtenue lors de l'itération précédente. Dans l'étape suivante à $t+t_c$, les gains correspondants sont conservés tandis que le vecteur des états d'erreur est mis à jour.

L'objectif principal de la distribution temporelle décrite ci-dessus est de fournir un contrôle réactif tout en assurant un comportement robuste. Cela implique que les gains optimaux respectent les contraintes susmentionnées et que le modèle de référence soit amélioré afin d'obtenir une structure proche de la réalité.

- **Sliding Mode Control MPC (MPSMC)** : On retrouve dans la littérature plusieurs implémentations du SMC pour le contrôle de drone [118], [119]. La présente section présente une variante du MPC qui incorpore le SMC dans le problème optimal. Cela permet de moduler les gains du SMC en fonction de la connaissance des perturbations (estimations), ce qui conduit à un rejet optimal des perturbations. Le SMC bien connu offre une certaine robustesse face aux perturbations bornées. On peut distinguer deux phases au cours du processus de stabilisation :

(i) la période au cours de laquelle les trajectoires d'état convergent vers la surface de glissement (hyperplan) $s(X)$ est connue sous le nom de mode de convergence (*reaching phase*), dont la conception de la commande doit satisfaire à

$$\mathbf{s}^\top \frac{\partial s}{\partial X} \dot{X} \leq -\eta \|s\| \quad (3.54)$$

où $\eta > 0 \in \mathbb{R}$ représente la constante de joignabilité (reachability constant).

De même, (ii) la trajectoire de l'état n'atteint pas seulement la variété de glissement, mais y reste pour $t > 0$, ce qui est appelé mode de glissement. Cela implique que

$$s \rightarrow 0 \Rightarrow \frac{\partial s}{\partial X} \dot{X} = 0 \quad (3.55)$$

La variété de glissement s est généralement exprimé en termes d'erreur d'état e comme suit :

$$s = \left(\frac{d}{dt} + \gamma \right)^{n-1} e \quad (3.56)$$

où $\gamma > 0$ correspond à un gain scalaire et n à l'ordre du système. Pour réduire le chattering inhérent au SMC, la fonction hyperbolique $\tanh(s)$ vient remplacer la fonction $\text{sign}(s)$ et l'analyse de stabilité est effectuée en conséquence.

On considère $\mathbf{s}_k = (s_k^x, s_k^y, s_k^z)^\top \in \mathbb{R}^3$ la variété de glissement discrète définie par

$$\mathbf{s}_k = \boldsymbol{\varepsilon}_{2k} + \gamma \boldsymbol{\varepsilon}_{1k} \quad (3.57)$$

Si l'on considère le système (3.35), le contrôleur qui stabilise cette dynamique est donné par

$$\mathbf{v}_k = M_n (\beta \tanh^*(\beta_1 \mathbf{s}_k) + \gamma \boldsymbol{\varepsilon}_{2k}) \quad (3.58)$$

Où $\beta = \text{diag}(\beta_x, \beta_y, \beta_z)$, $\gamma = \text{diag}(\gamma_x, \gamma_y, \gamma_z)$ et β_1 est un gain scalaire positif, tous destinés à moduler la réponse du SMC contre les perturbations.

Avec $\tanh^*(\beta_1 \mathbf{s}) = (\tanh(s_k^x), \tanh(s_k^y), \tanh(s_k^z))^\top$

Par conséquent, la loi de convergence en discret proposée est donnée par

$$\mathbf{s}_{k+1} = \mathbf{s}_k - t_c \left(\beta \tanh^*(\beta_1 \mathbf{s}_k) + M_n^{-1} \Delta_\xi \right) \quad (3.59)$$

t_c est le temps d'échantillonnage du contrôleur.

Lemma 3.3.1. *Si l'on tient compte de la loi d'atteinte proposée dans l'équation 3.59, s_k^i convergera vers la région définie par $\mathcal{R}_i = \{s_k^i \in \mathbb{R} : |s_k^i| \leq \epsilon\}$ et devient contraint à l'intérieur de cette région sans pouvoir s'en échapper. Nous définissons ϵ*

$$\epsilon = t_c \left(\frac{1}{m} (\Delta_{\xi_{max}}) + \frac{\lambda_i}{2} \right) \quad (3.60)$$

$\lambda_i > 0$ est un scalaire positif.

Proof. Considérons la fonction candidate de Lyapounov est donnée par

$$\mathbf{V}_k(\mathbf{s}_k) = \frac{1}{2} \mathbf{s}_k^\top \mathbf{s}_k = \sum (s_k^i)^2 \quad i \in \{x, y, z\} \quad (3.61)$$

Dont la variante est donnée comme suit

$$\begin{aligned} \Delta \mathbf{V}_{k+1} &= \mathbf{V}_{k+1} - \mathbf{V}_k \\ &= \sum \left((s_{k+1}^i - s_k^i)(s_{k+1}^i + s_k^i) \right) \quad i \in \{x, y, z\} \end{aligned} \quad (3.62)$$

Lorsque $s_k^i \neq 0$ pour assurer la stabilité du système en boucle fermée en faisant en sorte que s_k^i converge vers \mathcal{R}_i , il est essentiel que la variation de la fonction de Lyapounov soit négative $\Delta \mathbf{V}_{k+1} < 0$. Les conditions équivalentes qui doivent être satisfaites par la loi de convergence 3.59 sont les suivantes [120], [121]

$$(s_{k+1}^i - s_k^i) \text{sign}(s_k^i) < 0 \quad (3.63)$$

$$(s_{k+1}^i + s_k^i) \text{sign}(s_k^i) > 0 \quad (3.64)$$

Où $\text{sign}(s_k^i) = \frac{|s_k^i|}{s_k^i}$, 3.63 est appelé la condition de glissement et 3.63 la condition de convergence, nous avons

$$\begin{aligned} & (s_{k+1}^i - s_k^i) \text{sign}(s_k^i) \\ &= -t_c \left(\beta_i \tanh(\beta_1 s_k^i) + \frac{1}{m} \Delta_\xi^i \right) \text{sign}(s_k^i) \end{aligned} \quad (3.65)$$

$$\begin{aligned} & (s_{k+1}^i + s_k^i) \text{sign}(s_k^i) \\ &= 2|s_k^i| - t_c \left(\beta_i \tanh(\beta_1 s_k^i) + \frac{1}{m} (\Delta_\xi^i) \right) \text{sign}(s_k^i) \end{aligned} \quad (3.66)$$

Les paragraphes suivants couvrent deux cas distincts

– **Cas 1** : $s_k^i > 0$, en utilisant 3.65 on a

$$\begin{aligned} & (s_{k+1}^i - s_k^i) \text{sign}(s_k^i) \\ &= -t_c \left(\beta_i \tanh(\beta_1 s_k^i) + \frac{1}{m} \Delta_\xi^i \right) \\ &\leq -t_c \left(\beta_i \tanh(\beta_1 s_k^i) - \frac{1}{m} \Delta_{\xi_{max}} \right) \end{aligned} \quad (3.67)$$

Pour $\beta_i > \frac{1}{m} (\Delta_{\xi_{max}})$, la condition 3.63 est vérifiée. On définit ainsi,

$$\beta_i = \frac{1}{m} (\Delta_{\xi_{max}}) + \lambda_i \quad (3.68)$$

En partant de 3.66 on a ,

$$\begin{aligned} & (s_{k+1}^i + s_k^i) \text{sign}(s_k^i) \\ &= 2s_k^i - t_c \left(\beta_i \tanh(\beta_1 s_k^i) + \frac{1}{m} (\Delta_\xi^i) \right) \\ &\geq 2s_k^i - t_c \left(\beta_i + \frac{1}{m} (\Delta_{\xi_{max}}) \right) \\ &\geq 2s_k^i - t_c \left(\frac{2}{m} (\Delta_{\xi_{max}}) + \lambda_i \right) \end{aligned} \quad (3.69)$$

Pour s'assurer que la condition 3.64 est satisfaite,

$$s_k^i > t_c \left(\frac{1}{m} (\Delta_{\xi_{max}}) + \frac{\lambda_i}{2} \right) \quad (3.70)$$

– **Cas 2** : $s_k^i < 0$, en partant de 3.65 on a

$$\begin{aligned} & (s_{k+1}^i - s_k^i) \text{sign}(s_k^i) \\ &= t_c \left(\beta_i \tanh(\beta_1 s_k^i) + \frac{1}{m} \Delta_\xi^i \right) \\ &\geq t_c \left(\beta_i \tanh(\beta_1 s_k^i) - \frac{1}{m} \Delta_{\xi_{max}} \right) \end{aligned} \quad (3.71)$$

Puisque $\beta_i > \frac{1}{m} (\Delta_{\xi_{max}})$, la condition 3.63 est vérifiée. D'après 3.66, nous avons également ,

$$\begin{aligned}
& (s_{k+1}^i + s_k^i) \text{sign}(s_k^i) \\
&= -2s_k^i + t_c \left(\beta_i \tanh(\beta_1 s_k^i) + \frac{1}{m} (\Delta_\xi^i) \right) \\
&\leq -2s_k^i + t_c \left(\beta_i + \frac{1}{m} (\Delta_{\xi_{max}}) \right) \\
&\leq 2s_k^i + t_c \left(\frac{2}{m} (\Delta_{\xi_{max}}) + \lambda_i \right)
\end{aligned} \tag{3.72}$$

Pour s'assurer que la condition 3.64 est remplie,

$$s_k^i < -t_c \left(\frac{1}{m} (\Delta_{\xi_{max}}) + \frac{\lambda_i}{2} \right) \tag{3.73}$$

La condition de convergence 3.64 est remplie dans les deux cas lorsque $|s_k^i| > \epsilon$, ce qui signifie que s_k^i diminue en dehors de la région \mathcal{R}_i . En entrant dans \mathcal{R}_i , où $|s_k^i| \leq \epsilon$, et en considérant que la condition 3.63 reste constamment satisfaite pour les valeurs choisies de β_i , nous pouvons en déduire ce qui suit

$$|s_{k+1}^i| \leq \epsilon \tag{3.74}$$

Il s'ensuit que $s_{k+1}^i \in \mathcal{R}_i$, ce qui permet de conclure que le système est stable dans des limites restreintes.

□

Bien que l'analyse de stabilité en boucle fermée ci-dessus soit vérifiée, pour le système (3.35), une couche de contrôle complémentaire est conçue pour modeler la réponse du SMC afin de satisfaire aux conditions de performance (contraintes). À cette fin, considérons toujours l'estimation de la perturbation fournie par l'ES-LKF ainsi que la fonction de coût suivante,

$$\min_{\{\mathbf{v}_k, \dots, \mathbf{v}_{N-1}\}} \frac{1}{2} \sum_0^N (\mathbf{v}_k^\top \tilde{R}_m \mathbf{v}_k + \boldsymbol{\varepsilon}_k^\top \tilde{Q}_m \boldsymbol{\varepsilon}_k) \tag{3.75}$$

Sujet à,

$$\begin{aligned}
C1: & \boldsymbol{\varepsilon}_{k+1} = \bar{A} \boldsymbol{\varepsilon}_k - \bar{B} \mathbf{v}_k - \bar{B}_p \Delta_\xi \quad \forall k \in \{0, N\} \\
C2: & \mathbf{v}_k = \gamma \boldsymbol{\varepsilon}_{2k} + \beta \tanh\left(\frac{1}{2}(\gamma \boldsymbol{\varepsilon}_{1k} + \boldsymbol{\varepsilon}_{2k})\right) \quad \forall k \in \{0, N\} \\
C3: & \gamma_{min} \leq \gamma \leq \gamma_{max} \\
C4: & \beta_{min} \leq \beta \leq \beta_{max} \\
C5: & \boldsymbol{\varepsilon}_0 = \hat{\boldsymbol{\varepsilon}}(t) \\
C6: & \Delta_\xi = \hat{\Delta}_\xi
\end{aligned} \tag{3.76}$$

\bar{A}, \bar{B} et \bar{B}_p sont respectivement la discrétisation des matrices A, B et B_p , $\gamma = \text{diag}(\gamma_x, \gamma_y, \gamma_z)$,

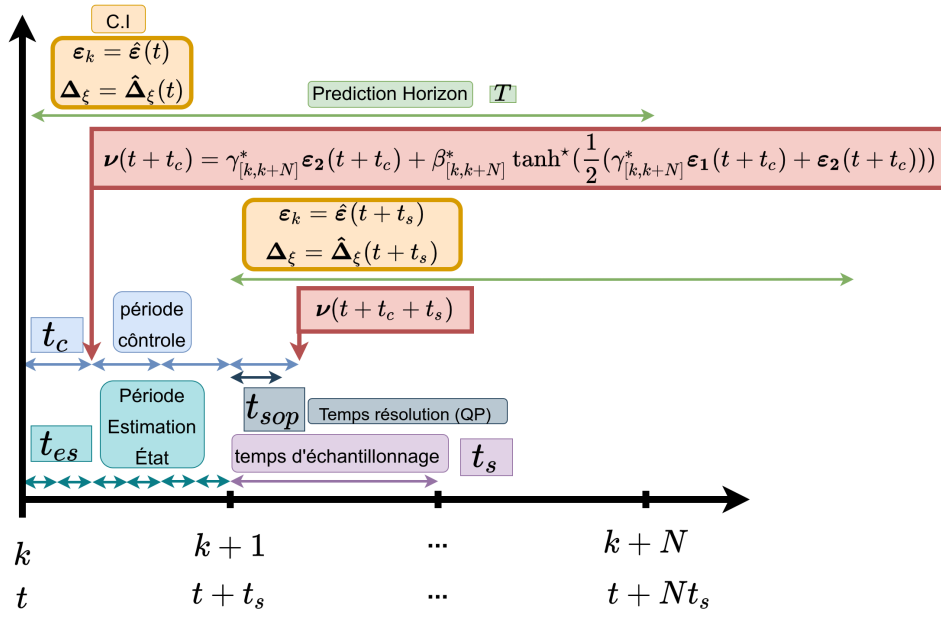


Figure 3.5: Description du contrôleur MPSMC.

$\beta = \text{diag}(\beta_x, \beta_y, \beta_z)$. Étant donné que le profil de performance exprimé par le problème d'optimisation multi-objectifs (3.75), les contraintes correspondantes (3.52) sont destinées à (C1) définir le modèle de référence à des fins de prédiction, dont la structure englobe les perturbations. (C2) calculer le contrôleur optimal tout en respectant (C3)/(C4) une plage de gains prédéfinie pour la surface de glissement, c'est-à-dire un gain proportionnel optimal $\gamma_{[k,k+N]}^*$) et d'amplitude ($\beta_{[k,k+N]}^*$) optimaux, respectivement. Enfin, comme dans l'approche MPC précédente, (C5) vise à mettre à jour les états filtrés et l'estimation des perturbations. La figure 3.5 schématise le contrôleur.

3.4 Conclusion

Dans cette section, nous avons établi différentes approches pour le contrôle de la position ainsi que le contrôle de l'attitude du drone. Nous avons également présenté des propositions pour l'estimation des perturbations externes et pour le filtrage des bruits de mesure. Ces fondements théoriques constituent une base solide qui sera utilisée dans la suite de notre étude. Cette phase comprendra une première étape de validation par simulation de ces diverses approches, suivie d'une deuxième étape de validation expérimentale. En combinant ces approches théoriques avec des simulations poussées et des essais concrets, nous visons à affiner et à valider nos méthodes de contrôle et d'estimation, en préparation des tests réels sur des plateformes de drones.

4

Validation dans des environnements de simulation

Sommaire

4.1	Quadrirotor à bras fixes	101
4.1.1	ES-iLQG	101
4.2	Quadrirotor à bras pivotants (Xtilt)	106
4.2.1	Contrôleur PID	106
4.2.2	Intégrale MPC & MPSMC	109
4.2.3	ES-LKF & STO	117
4.3	Conclusion	119

SIMULATION-Le chapitre précédent a introduit les différentes commandes adoptées pour contrôler le drone en orientation et en position. Il est nécessaire de tester ces différentes lois de commande en simulation afin d'établir leurs caractéristiques en termes de robustesse et de rejets de perturbations. Dans ce chapitre, les différents environnements de simulation sont introduits ainsi que les résultats obtenus pour les lois de contrôle proposées dans le chapitre précédent.

4.1 Quadrirotor à bras fixes

La section suivante est structurée conformément à l'illustration présentée dans la figure (4.1). Nous évaluons ici les performances du contrôleur ES-iLQG dans diverses conditions, en mettant particulièrement l'accent sur sa résilience face aux perturbations extérieures, quantifiée en termes d'erreur de suivi de trajectoire de référence. En ce qui concerne l'estimateur des perturbations (ES-LKF), nous mesurons sa précision en examinant l'erreur d'estimation des perturbations. Pour mener à bien ces évaluations, nous avons utilisé deux simulateurs distincts. La première simulation a été réalisée à l'aide de Matlab/Simulink. Cette simulation comprend un modèle 2D du drone évoluant dans le plan (zy) et intègre la modélisation des diverses perturbations. La deuxième simulation a été réalisée sous ROS 1/Gazebo. Dans cette configuration, la dynamique du drone est résolue grâce au solveur physique de Gazebo, et la simulation inclut à la fois la modélisation des perturbations et celle du matériel embarqué sur le prototype expérimental Coex Clover 4.

4.1.1 ES-iLQG

À des fins d'évaluation des performances, une étape de simulation a été réalisée sur la base du schéma de contrôle représenté sur la figure (4.2). Dans la partie inférieure, vous trouverez la structure du contrôleur ES-iLQG, tandis que dans la partie supérieure, la boucle d'asservissement


Plateforme	Contrôleur	Évaluation	Simulateur
Quadrirotor à bras fixes 	ES-iLQG	<ul style="list-style-type: none"> • Erreur de suivi <ul style="list-style-type: none"> ◦ Perturbation constante/variable ◦ Bruit Gaussien • Erreur d'estimation des perturbations 	Matlab/Simulink <ul style="list-style-type: none"> • 2D (yz) • Modélisation de la dynamique • Modélisation des perturbations ROS 1/ Gazebo <ul style="list-style-type: none"> • 3D • Solveur physique ODE • Simulation autopilote PX4 • Modélisation des perturbations

Figure 4.1: Organisation des simulations réalisées pour le quadrirotor à bras fixes.

interne pour l'orientation du drone est mise en évidence, utilisant un contrôleur PID. De plus, les différentes consignes d'orientation ainsi que la poussée désirée sont calculées en fonction de la commande du contrôleur en position (ES-iLQG). Des simulations en temps continu et discret de la commande en boucle fermée et de l'estimation ont été réalisées à l'aide de Matlab/Simulink dans un premier temps et sur Ros/Gazebo dans un second temps, avec les paramètres spécifiés dans le tableau (4.1).

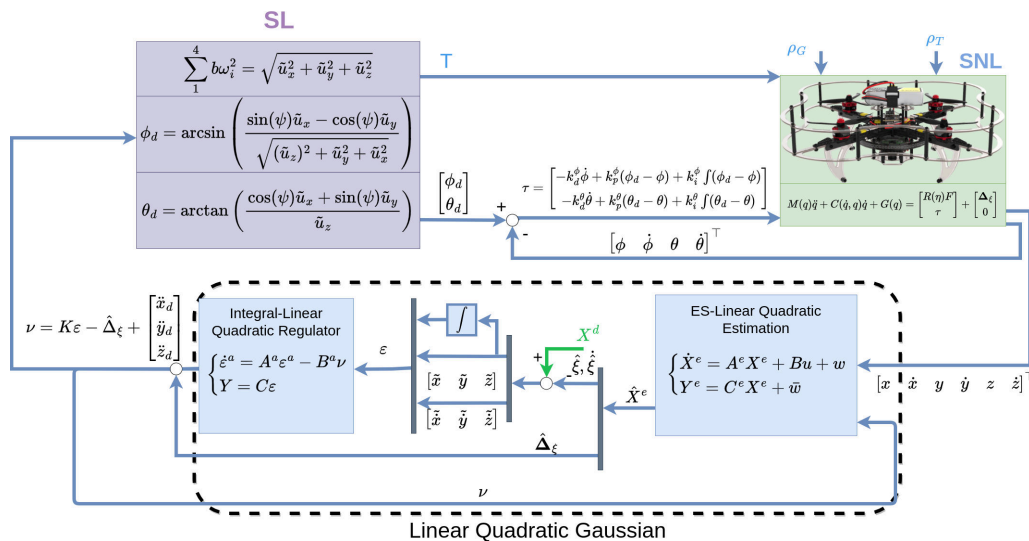


Figure 4.2: Description de l'architecture de contrôle ES-iLQG (Extended State Integral Linear Quadratic Gaussian).

	Description	Valeur
Kalman	w_k covariance (Q)	$diag(0.01, 0.01, 0.01, 0.01, 45, 45)$
	\bar{w}_k covariance (R)	$diag(1, 1, 1, 1)$
LQR	Matrice coût État (\tilde{Q})	$diag(15, 5, 5, 15, 1, 1)$
	Matrice coût contrôle (\tilde{R})	$diag(1, 1)$

Table 4.1: Paramètres du filtre de Kalman et du LQR

Le modèle de drone utilisé pour les simulations présente les paramètres décrits dans le tableau

(4.2) correspondant à la plateforme aérienne expérimentale Coex Clover 4.

	Description	Valeur	Unité
	Masse (m)	887	g
Quadcopter	moment d'inertie x (J_ϕ)	0.001	kgm ²
	moment d'inertie y (J_θ)	0.001	kgm ²
	moment d'inertie z (J_ψ)	0.013	kgm ²

Table 4.2: Paramètres physiques du Coex Clover 4 données constructeur

Évaluation de la robustesse, le rejet et l'estimation des perturbations

- **Matlab/Simulink** : Au cours de ces simulations, nous supposons que le drone se déplace dans le plan (zy) et nous considérons diverses perturbations à différents instants afin d'évaluer la réactivité du système. D'abord, une perturbation gaussienne est introduite sur les mesures de position à $t = 10s$. Une perturbation constante est ensuite introduite au temps 40s sur les axes y et z avec la même magnitude. Notre intérêt est d'évaluer comment le filtre de Kalman approxime la perturbation en présence d'un bruit non gaussien. Dans un dernier cas, une perturbation non constante est introduite dans le système, comme dans notre modèle les perturbations sont supposées constantes dont la dynamique est modélisée par un bruit gaussien. Notre intérêt ici est d'évaluer le comportement de l'observateur face à ce type de perturbation. Il faut noter que le modèle d'observation dans ce cas-là est évalué à un pôle réduit.

Les résultats (figures (4.3a) et (4.3b)) montrent qu'en présence d'une perturbation variable dans le temps et pour une grande variance pour (3.9), l'architecture proposée est capable de compenser toutes les perturbations introduites dans la dynamique translationnelle.

- **Ros/Gazebo Classics** : À la différence de la partie précédente où la simulation est réalisée dans un environnement 2D, le drone évolue dans un environnement 3D, dont la dynamique est régie par un moteur physique Open Dynamics Engine (Ode). Cette approche permet une première validation des lois de commande basées modèle, car la dynamique est résolue par le solveur physique.

De plus, cette simulation intègre la modélisation complète des différents capteurs embarqués sur le Coex Clover4, tels que la caméra pour le flux optique et le capteur de distance pour mesurer la distance par rapport au sol. Les moteurs brushless ainsi que les hélices sont également modélisés, prenant en compte le temps de réponse des moteurs et les coefficients aérodynamiques des hélices. Cette approche globale garantit une représentation fidèle de l'environnement du drone et des composants essentiels. La base de cette simulation repose sur les travaux référencés dans [122] et [123].

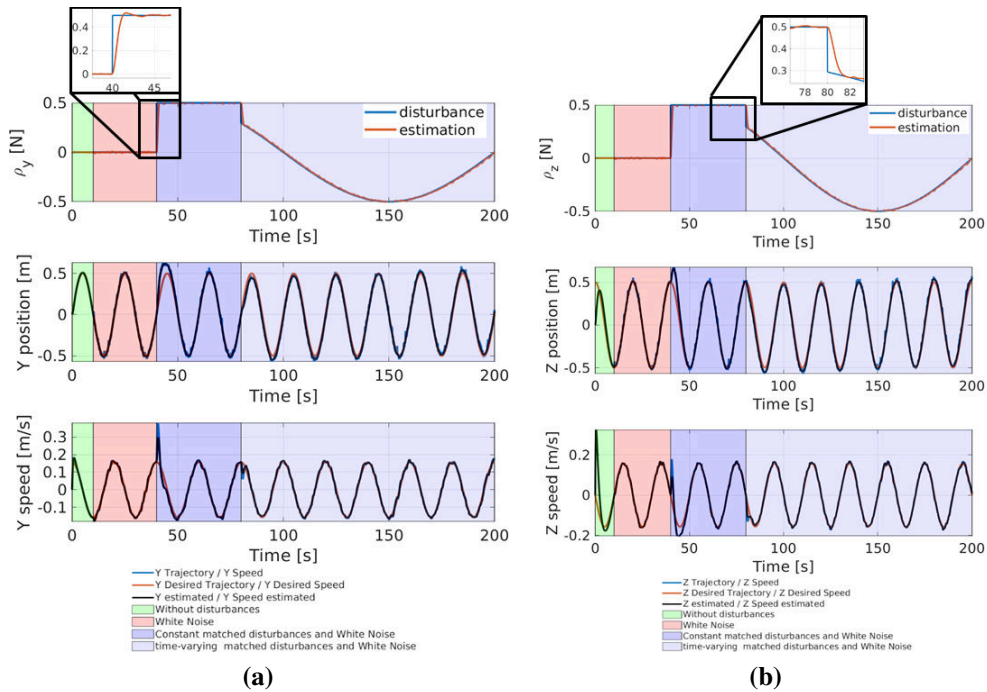


Figure 4.3: (a) Comportement du contrôleur en position et vitesse sur l'axe y face à différentes perturbations (*courbe supérieure*): Bruit gaussien, perturbation constante de la valeur 0.5N sur l'intervalle de temps [40;80], perturbation sinusoïdale $\rho_y = 0.5 \sin(0.01\pi t)$ à partir de $t = 80s$. Les trajectoires désirées sont $y_d = 0.5 \sin(0.1\pi t)$ et $\dot{y}_d = 0.05\pi \cos(0.1\pi t)$, on retrouve aussi l'estimation de la perturbation, (b) Comportement du contrôleur en position et vitesse sur l'axe z face à différentes perturbations (*courbe supérieure*): Bruit gaussien, perturbation constante de la valeur 0.5N sur l'intervalle de temps [40;80], perturbation sinusoïdale $\rho_z = 0.5 \sin(0.01\pi t)$ à partir de $t = 80s$. Les trajectoires désirées sont $z_d = 0.5 \cos(0.1\pi t)$ et $\dot{z}_d = -0.05\pi \sin(0.1\pi t)$, on retrouve aussi l'estimation de la perturbation.

Le contrôleur en orientation est celui de l'auto-pilote PX4, tandis que le contrôleur en position, basé sur ES-iLQG, est implémenté dans l'environnement ROS Noetic. La figure (4.4) illustre en détail l'environnement de simulation.

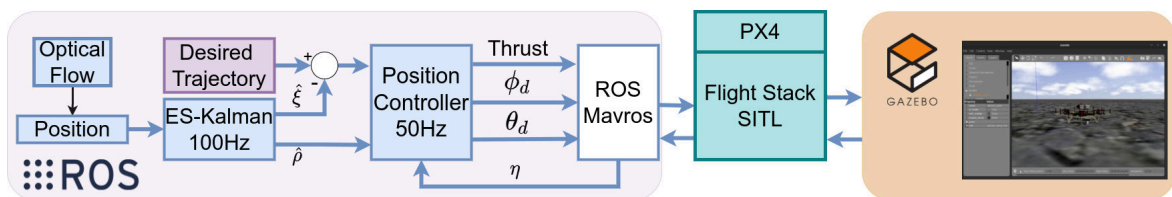


Figure 4.4: Description de l'environnement de simulation Ros/Gazebo.

Le drone suit une trajectoire hélicoïdale (4.5a), tout en étant soumis à une incertitude paramétrique de 100 grammes sur sa masse, représentant environ 11% de la masse totale du drone. La figure (4.5b) illustre l'erreur de suivi en position par rapport à la trajectoire désirée, démontrant la capacité du contrôleur à compenser la perturbation paramétrique

liée à l'incertitude de la masse, ainsi qu'aux incertitudes du modèle. De manière similaire, l'erreur de suivi en vitesse dans la figure (4.6), montre une performance similaire.

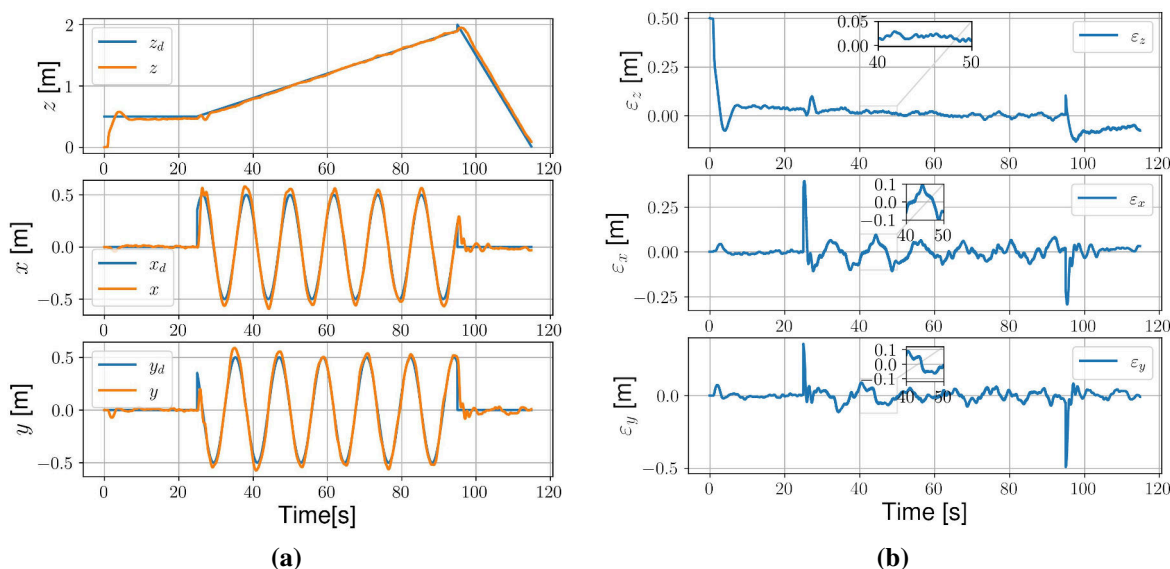


Figure 4.5: (a) Suivi d'une trajectoire hélicoïdale le drone étant soumis à des perturbations, décrites par $x_d = 0.5 \sin(0.2\pi t)$, $y_d = 0.5 \cos(0.2\pi t)$, $z_d = 0.5 + 0.03t$, (b) Erreur en position de suivi d'une trajectoire hélicoïdale le drone étant soumis à des perturbations.

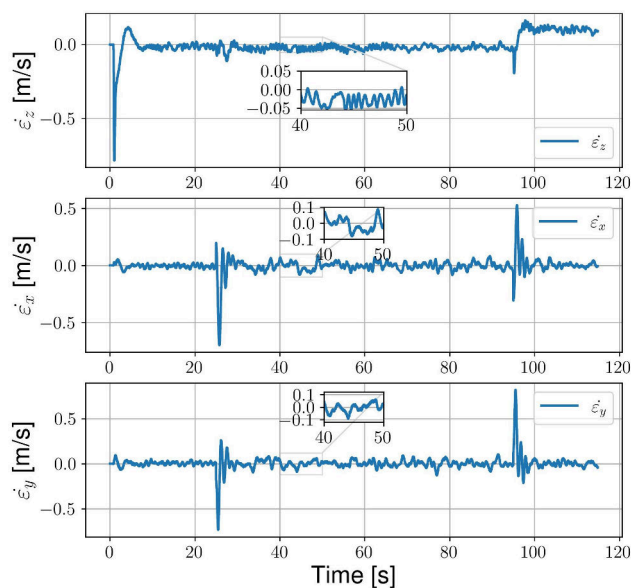


Figure 4.6: Erreur en vitesse de suivi d'une trajectoire hélicoïdale le drone étant soumis à des perturbations, la vitesse désirée est $\dot{x}_d = 0.1\pi \cos(0.2\pi t)$, $\dot{y}_d = -0.1\pi \sin(0.2\pi t)$, $\dot{z}_d = 0.03\text{m/s}$.

4.2 Quadrirotor à bras pivotants (Xtilt)

Cette section a pour but d'évaluer en simulation le comportement global de la plateforme Xtilt (quadrirotor à bras pivotant), afin d'observer et de collecter des données préliminaires concernant ce système pour remplir des missions de contrôle en position en présence de perturbations. Plusieurs simulations ont été réalisées pour le même système soumis à différentes perturbations, différents contrôleurs et sous différents environnements (voir figure (4.7)). Le modèle de simulation utilise les paramètres décrits dans le tableau (4.3) correspondant à la plateforme aérienne expérimentale.


	PID	<ul style="list-style-type: none"> • Navigation holonomique / non-holonomique • Consommation d'énergie Xtilt vs Quad 	Matlab/Simulink <ul style="list-style-type: none"> • 3D • Modélisation de la dynamique • Effets aérodynamiques • Modélisation des perturbations
	iLQG iMPC MPSC	<ul style="list-style-type: none"> • Erreur de suivi <ul style="list-style-type: none"> ◦ Effet du choix de l'horizon ◦ Effet de la correction du modèle nominal ◦ Compraison avec l'iLQG ◦ Perturbation constante/variable • Rapidité de convergence STO 	ROS 2/ Ignition Gazebo <ul style="list-style-type: none"> • 3D • Solveur physique ODE • Simulation des servomoteurs • Simulation des moteurs • Effets aérodynamiques • Modélisation des perturbations

Figure 4.7: Description du ES-iLQG (Extended State Integral Linear Quadratic Gaussian).

	Description	Valeur
Forces aérodynamiques	$C_d A$	0.04 m^2
Châssis	masse	0.815 kg
	I_b	$diag(6.8e^{-3}, 7.9e^{-3}, 9e^{-3}) \text{ kgm}^2$
Bras	masse	0.04 kg
	$I_A^{\alpha_i}$	$diag(2.0e^{-5}, 7.7e^{-6}, 2.2e^{-5}) \text{ kgm}^2$
Propulsion	b	$3.135e^{-7} \text{ kg/tpwm}^2$
	d	$3.8e^{-8} \text{ Nm/tpwm}^2$

Table 4.3: Paramètres du drone Xtilt. Pour le coefficient de traînée, nous prenons la valeur moyenne, tpwm est la largeur d'impulsion multipliée par $1e^3$, pour les paramètres aérodynamiques, les valeurs ont été obtenues par la CFD, tandis que les autres paramètres ont été mesurés expérimentalement. Toutefois, il convient de noter que la matrice d'inertie pour les bras a été obtenue par la CAO.

4.2.1 Contrôleur PID

Un ensemble de scénarios de simulations est réalisé en tenant compte d'un simple contrôleur PID. À ce stade préliminaire, la simulation ne tient pas compte des perturbations externes (par exemple, les rafales de vent) et des bruits des capteurs, mais elle tient compte du modèle dynamique

complet (2.22) du drone et des effets gyroscopiques et aérodynamiques. L'objectif de cette section est triple : (i) évaluer l'efficacité de la loi de contrôle pour la navigation holonomique et non holonomique, c'est-à-dire que nous utilisons ψ_d ou ϕ_d pour piloter le mouvement sur l'axe y , (ii) évaluer les capacités du quadriporteur à évoluer dans l'espace 3D, et (iii) une comparaison préliminaire entre les deux configurations en termes de consommation d'énergie. Pour cette partie, les simulations ont été réalisées sous Matlab/Simulink.

Premier scénario

Le scénario considéré ici englobe deux cas de navigation tout en suivant une trajectoire en forme hélicoïdale.

1. La navigation non-holonomique lorsque la dynamique de l'axe y est actionnée par l'intermédiaire de contrôle du lacet

$$\psi_d = \arcsin\left(\frac{\sin(\phi)u_z}{\sqrt{\tilde{u}_x^2 + \tilde{u}_y^2}}\right) + \arctan\left(\frac{\tilde{u}_y}{\tilde{u}_x}\right) \quad (4.1)$$

Les performances permettant d'atteindre l'objectif de contrôle du suivi de la trajectoire dans le cadre de cette configuration de navigation sont illustrées sur les figures (4.9a) et (4.8).

2. La navigation holonomique lorsque la dynamique de l'axe y est actionnée par l'intermédiaire de

$$\phi_d = \arcsin\left(\frac{\sin(\psi)\tilde{u}_x - \cos(\psi)\tilde{u}_y}{u_z}\right) \quad (4.2)$$

La performance de l'objectif de contrôle du suivi de la trajectoire dans le cadre de cette configuration de navigation est illustrée dans figure (4.9b) et figure (4.8).

On remarque que dans le cas de navigation non-holonyme, comme le montre la figure (4.9a), la transition du point de consigne ψ_d de -179° à 179° pose problème. Cette transition rapide fait effectuer au drone un tour complet en un laps de temps très court, ce qui entraîne les pics visibles dans la figure (4.9a). Une manière de remédier à cette situation est de passer d'un contrôle en position à un contrôle en vitesse pour ψ , en calculant la dérivée de la position désirée pour $\dot{\psi}_d$.

Cette approche permettrait de lisser la transition entre les points de consigne angulaires et d'éviter les variations brusques dans le contrôle, ce qui améliorerait la stabilité du drone.

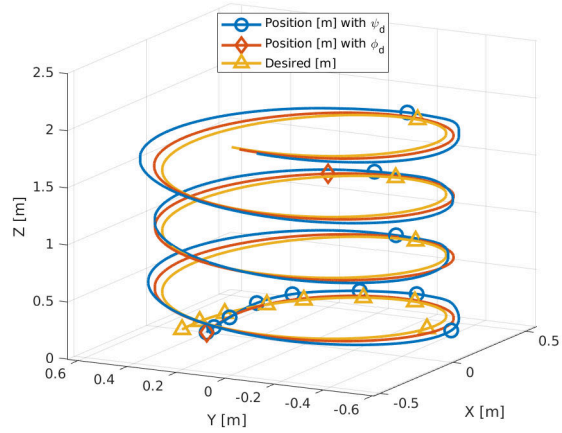


Figure 4.8: Suivi de trajectoire 3D pour la navigation holonomique (ϕ_d) et non holonomique (ψ_d)

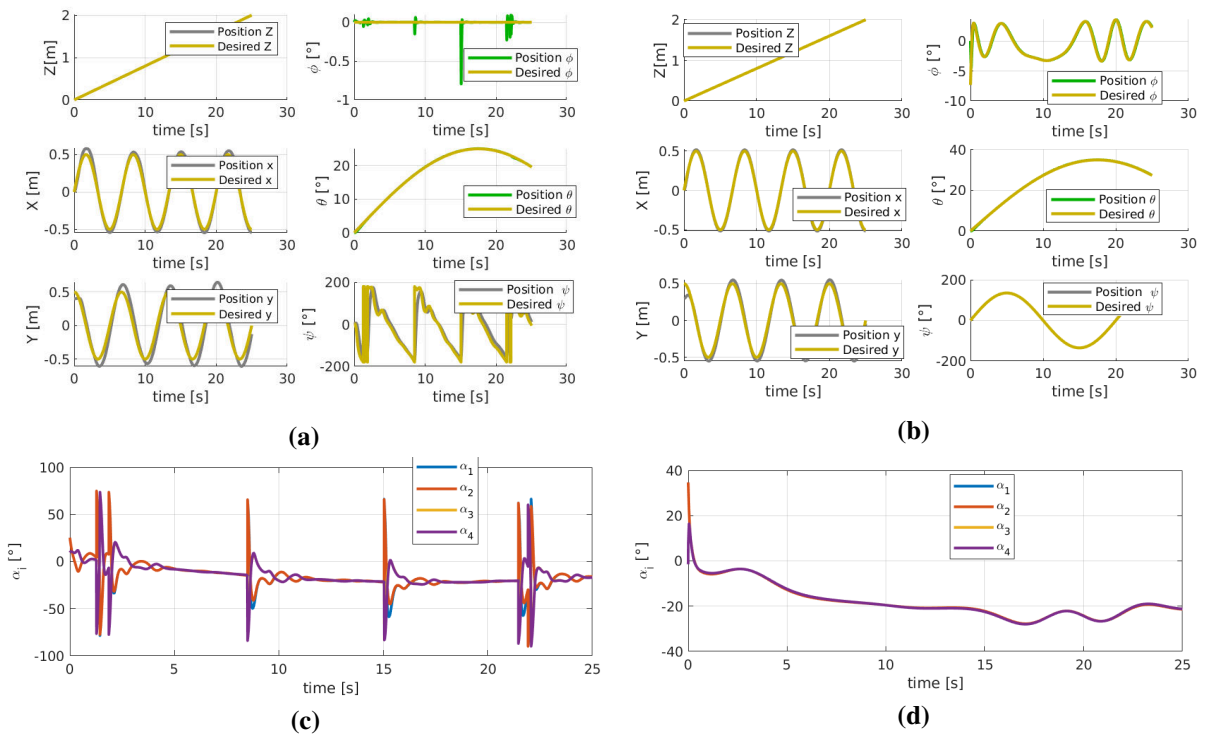


Figure 4.9: (a) Performance de suivi de trajectoire lorsque la dynamique de l'axe y est pilotée par un mouvement de lacet, (c) L'évolution des angles d'inclinaison, les pics élevés en α_i sont dus à l'évolution rapide des ψ_d de -179° à 179° et l'absence de saturation sur la commande, (b) Performance de suivi de trajectoire lorsque la dynamique de l'axe y est commandée par le mouvement de roulis (ϕ_d), (d) Évolution des angles d'inclinaison.

Deuxième scénario

Le deuxième scénario illustre la comparaison des deux configurations en termes de consommation d'énergie. La puissance mécanique à l'instant t générée par un moteur en considérant que le couple moteur est égal au couple induit par la traînée de l'hélice, est définie par,

$$P_i = d\omega_i^3 \quad (4.3)$$

Dans la figure (4.10), nous remarquons que les deux configurations atteignent l'objectif de contrôle. Cependant, à cette fin, le quadcoptère classique présente des angles de tangage importants, ce qui signifie une augmentation des vitesses des rotors pour compenser à la fois la perte de poussée verticale et la force de traînée qui devient plus importante en raison de l'augmentation de la surface frontale. Cependant, dans le cas du quadrirotor à bras pivotants le drone garde une assiette horizontale et une surface frontale minimale, ce qui réduit l'effet de la force de traînée.

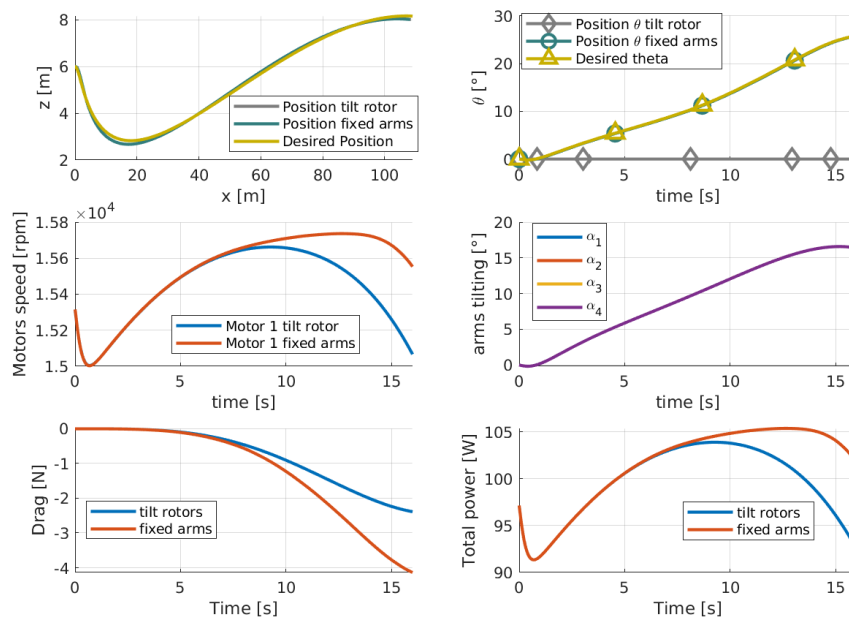


Figure 4.10: Comparaison de la consommation d'énergie entre la configuration classique et Xtilt

4.2.2 Intégrale MPC & MPSMC

La simulation a été réalisée sur la base du schéma de contrôle représenté sur la figure (4.11), sous l'environnement ROS2 Humble et Ignition Gazebo Fortress codé en C/C++. Une description complète du système a été réalisée sur Ignition Gazebo, qui comprend à la fois le modèle aérodynamique du drone, le modèle des servomoteurs, le système de propulsion (moteur brushless et hélice) ainsi que les différents capteurs. La figure (4.12) décrit la boucle de simulation

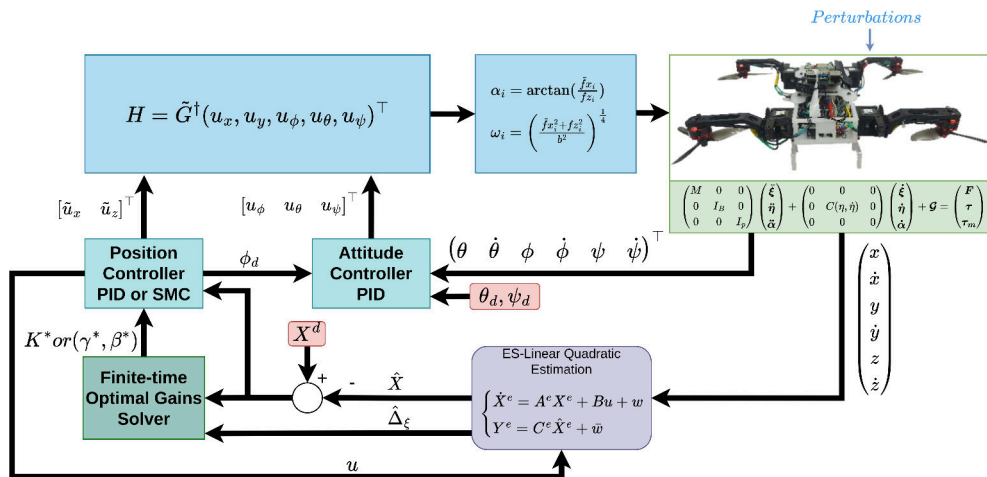


Figure 4.11: Architecture de contrôle en boucle fermée proposée pour l'intégral MPC et MPSMC

complète, toute la partie contrôle est codée en C/C++ sous l'environnement ROS2 humble, la bibliothèque de solveur optimal utilisée est COIN-OR IPOPT "Interior Point OPTimizer". La synchronisation temporelle ainsi que l'échange de messages entre ROS2 et Ignition se fait à travers le pont ROS2 Ignition Bridge.

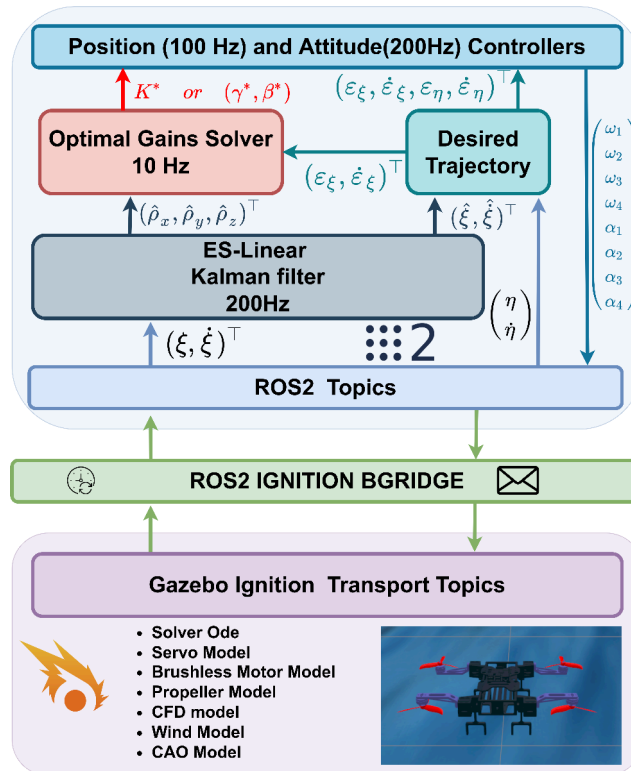


Figure 4.12: Description complète de la boucle de contrôle de la simulation, détaillant les différents nœuds ROS2 et les messages échangés entre eux ainsi que l'environnement de simulation <https://youtu.be/kjAZyqdocH4>

Afin d'évaluer les performances des schémas de contrôle proposés, deux cas sont considérés à des fins de comparaison, à savoir :

- Cas 1 : Perturbation constante
- Cas 2 : Perturbation due au vent
- Cas 3 : Perturbation due au vent fort

Cas 1: Perturbation constante

Pour ce premier scénario, le drone suit la trajectoire décrite dans la figure (4.13). Nous introduisons brusquement une perturbation constante au temps $t = 40s$ qui est maintenue jusqu'à $t = 80s$. Dans ce premier cas, nous appliquons la loi de contrôle (3.52) sans terme intégral et sur un seul axe (z). Dans la figure (4.13) pour la même fonction de coût et différents horizons ont été testés à différentes fréquences d'échantillonnages pour les gains, en raison du temps de calcul pour obtenir la solution optimale. En effet, en augmentant l'horizon de calcul de la solution optimale, on accroît le nombre de variables de décision, ce qui rend la résolution du problème plus complexe. Les fréquences affichées à droite de l'horizon dans la figure (4.13) représentent les fréquences moyennes mesurées sur une fenêtre de 300 secondes pour la résolution du problème optimal.

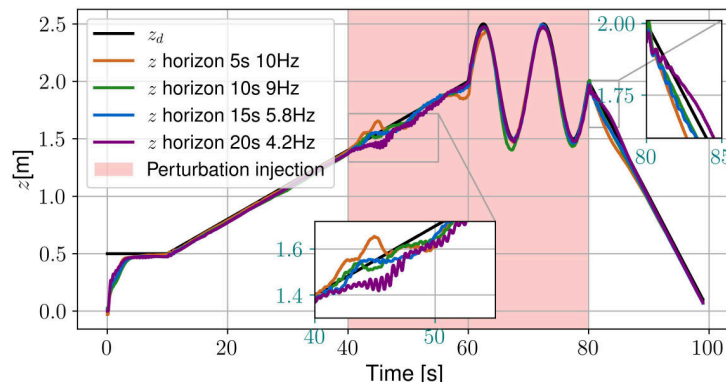


Figure 4.13: Test de l'effet de l'horizon d'optimisation sur le comportement global du suivi en présence et en absence de perturbations constantes

Le tableau (4.4) montre que pour le suivi de la trajectoire globale sans et avec perturbation externe, l'erreur quadratique moyenne la plus faible est obtenue avec un horizon de 20 secondes. Cependant l'erreur quadratique moyenne (RMSE) est minimale pour un horizon de $T = 15s$ lorsque le véhicule subit une perturbation exogène constante dans l'intervalle de temps $t \in [40s, 80s]$. Il convient également de noter que ce même intervalle, une augmentation de l'horizon à $T = 20s$ conduit à une augmentation de l'erreur quadratique moyenne. En effet, cela est dû au fait qu'en augmentant l'horizon, on réduit la fréquence d'échantillonnage pour les gains, ce qui diminue considérablement la réactivité du contrôleur en présence de perturbations. Dans de

telles situations, il est nécessaire de trouver un compromis entre l'horizon et la fréquence d'échantillonnage afin de maintenir un équilibre entre la validité de la solution optimale et la capacité à réagir rapidement face aux perturbations.

Time	Horizon	RMSE
[0, 100s]	5s	0.0634m
	10s	0.0617m
	15s	0.0536m
	20s	0.0528m
[40, 80s]	5s	0.0493m
	10s	0.0478m
	15s	0.0324m
	20s	0.0468m

Table 4.4: Le RMSE en fonction de l'horizon et de l'intervalle de temps correspondant à la présence ou non de la perturbation

Afin d'évaluer la contribution de l'estimation des perturbations au modèle nominal, nous avons réalisé deux simulations, utilisant le même horizon de temps et la même fonction de coût. Cependant, dans le premier cas, nous avons utilisé uniquement le modèle nominal, tandis que dans le second cas, nous avons utilisé le modèle nominal avec l'estimation des perturbations intégrée. Les figures (4.14) démontrent que l'inclusion de l'estimation des perturbations est plus performante que le cas qui n'inclut pas l'estimation des perturbations.

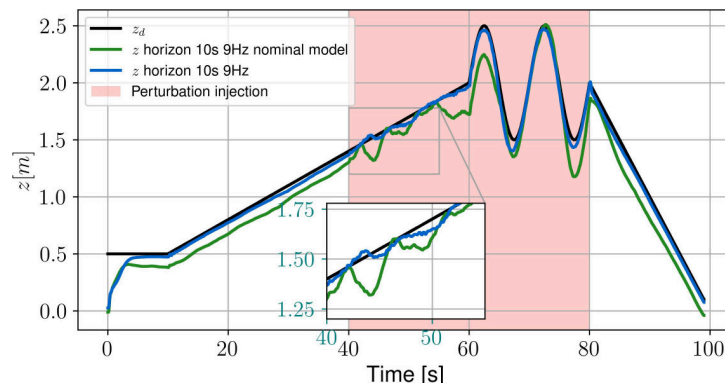


Figure 4.14: Comparaison entre deux suivis de trajectoire à l'aide de l'intégral MPC. [courbe verte] correspond au modèle sans estimation de la perturbation, tandis que [courbe bleue] tient compte de cette estimation.

Dans ce sens, une comparaison avec l'ES-iLQG est présentée sur la figure (4.15). Pour l'ES-iLQG, nous avons une erreur quadratique moyenne de 0,0814m alors que l'intégral MPC pour un horizon de 15s, nous avons une erreur quadratique moyenne de 0,0536m.

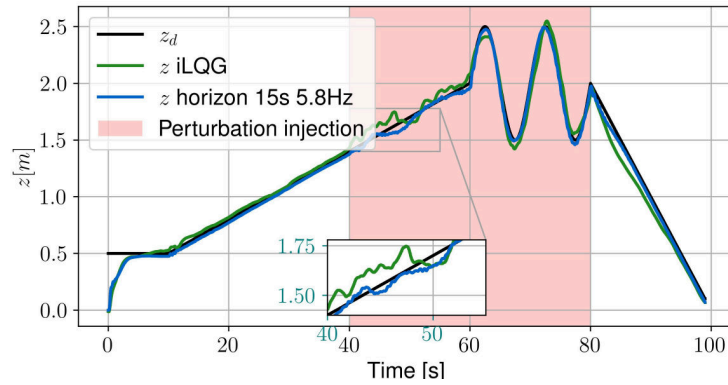


Figure 4.15: Comparaison en suivi de trajectoire en z entre l'intégral MPC et l'ES-iLQG

Cas 2 : Perturbation due au vent

Dans ce cas, on considère la perturbation due au vent qui comporte une composante constante et une composante variable dans le temps, variant à la fois en direction et en magnitude. L'intérêt de cet essai est d'évaluer la robustesse du contrôleur face aux perturbations externes. Les deux schémas MPC proposés sont testés en tenant compte de contraintes similaires et en suivant une trajectoire hélicoïdale. À la différence du scénario précédent, dans ce cas, le drone évolue dans environnement 3D.

- **Intégral MPC** : Le drone est amené à suivre une trajectoire hélicoïdale figure (4.16), le problème optimal (3.52) est résolu en considérant deux variantes de fonctions de coûts décrites par le tableau tableau (4.5), ou $\tilde{Q} = \text{diag}(Q_x, Q_y, Q_z)$ avec $Q_x, Q_y, Q_z \in \mathbb{R}^{3 \times 3}$.

Fonction coût	Variable	Valeur
coût 1	R_{pid}	$\text{diag}(0.025, 0.025, 0.025)$
	Q_x	$\text{diag}(0.25, 127.625, 20.475)$
	Q_y	$\text{diag}(0.25, 127.625, 20.475)$
	Q_z	$\text{diag}(0.25, 127.625, 20.475)$
coût 2	R_{pid}	$\text{diag}(127.75, 127.75, 127.75)$
	Q_x	$\text{diag}(0.25, 0.125, 0.475)$
	Q_y	$\text{diag}(0.25, 0.125, 0.475)$
	Q_z	$\text{diag}(0.25, 0.125, 0.475)$

Table 4.5: Paramètres des deux fonctions de coût utilisées. L'intérêt de tester ces deux fonctions de coût est de quantifier l'impact des matrices de coût sur les gains et l'erreur de suivi. Le coût 1 impose une contrainte plus importante sur l'erreur de suivi, tandis que le coût 2 impose une contrainte sur la commande.

On remarque figure (4.17a) que le suivi de trajectoire est meilleur pour la première fonction de coût, dû au fait que la contrainte sur le contrôle est plus élevée pour la seconde fonction

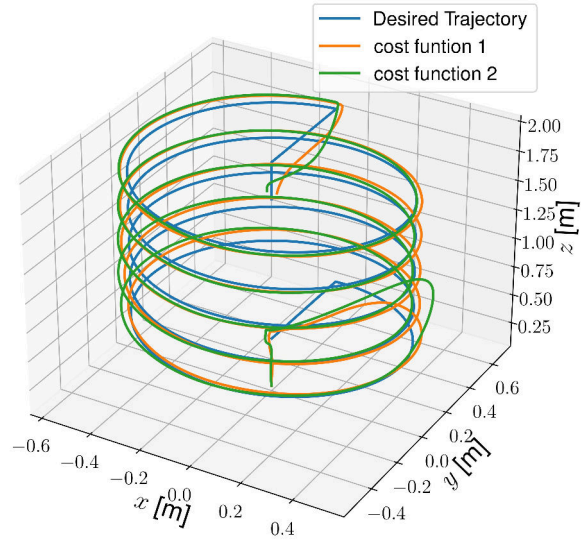


Figure 4.16: Représentation 3D du suivi de trajectoire en présence de perturbations variables en considérant l'intégral MPC avec deux fonctions de coût différentes.

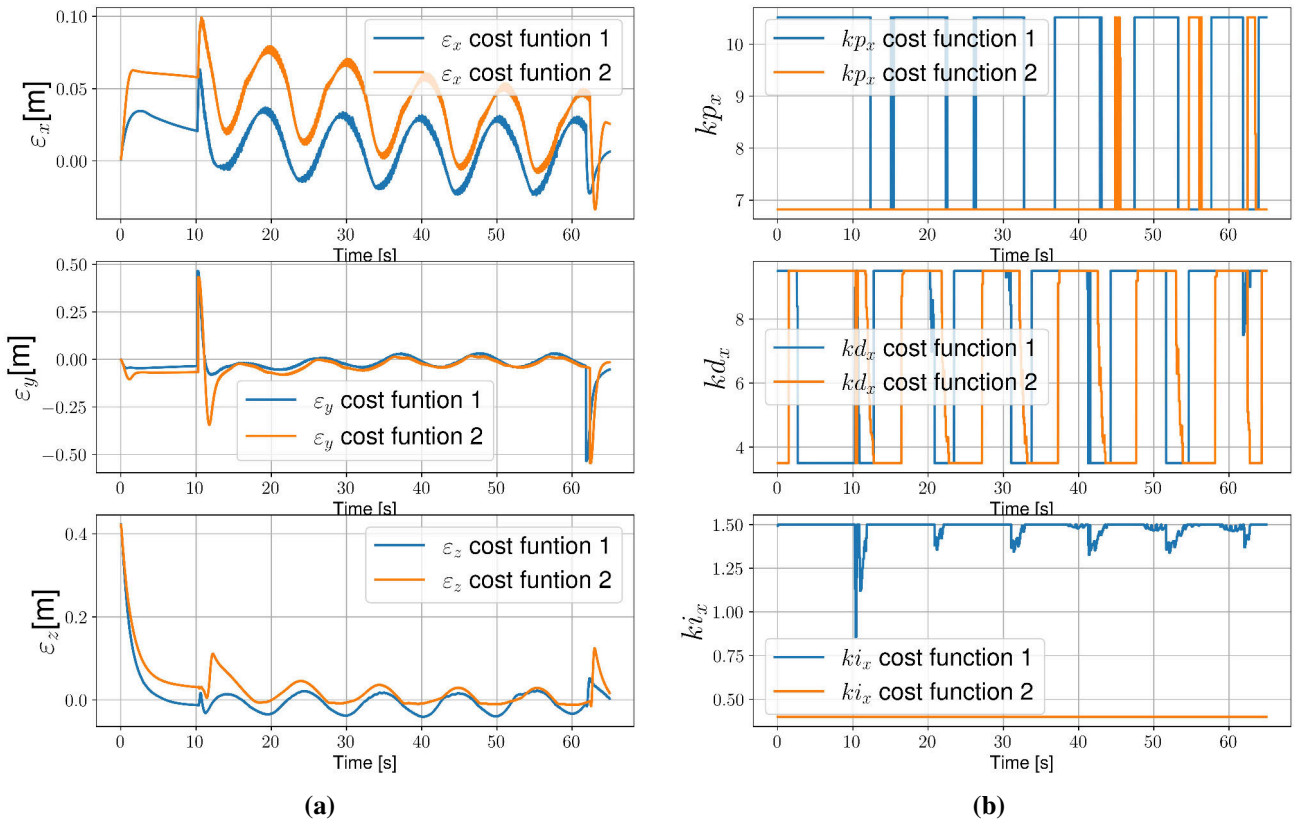


Figure 4.17: (a) Erreur de position en présence d'une perturbation variable en considérant un intégral MPC avec deux fonctions de coût différentes, (b) Les différents gains pour l'intégral MPC sur l'axe des x en présence de perturbations variables en considérant deux fonctions de coût différentes.

de coût. Ceci se traduit par une diminution des valeurs des gains figure (4.17b) avec une valeur moyenne pour kp_x respectivement pour le coût 1 et le coût 2 (9.28,6.98) ainsi que pour kd_x (7.5,7.15) et ki_x (1.48,0.4).

- **MPSMC** : Pour cet essai, deux profils de fonctions de coût sont considérés afin que le drone suive la même trajectoire hélicoïdale en appliquant la loi de contrôle (3.76). Nous observons le même effet des contraintes de contrôle sur les erreurs de position et de vitesse, voir figure (4.18a). On observe également le même effet sur les gains de contrôle, voir figure (4.18b). Cependant, nous remarquons que pour le même horizon et les mêmes contraintes sur l'erreur de position et l'erreur de vitesse, le comportement global du contrôleur (3.52) est meilleur, l'erreur moyenne quadratique étant plus faible figure (4.19). Les valeurs utilisées ici sont indiquées dans le tableau (4.6), où $\tilde{Q} = \text{diag}(Q_x, Q_y, Q_z)$, avec Q_x, Q_y et $Q_z \in \mathbb{R}^{2 \times 2}$.

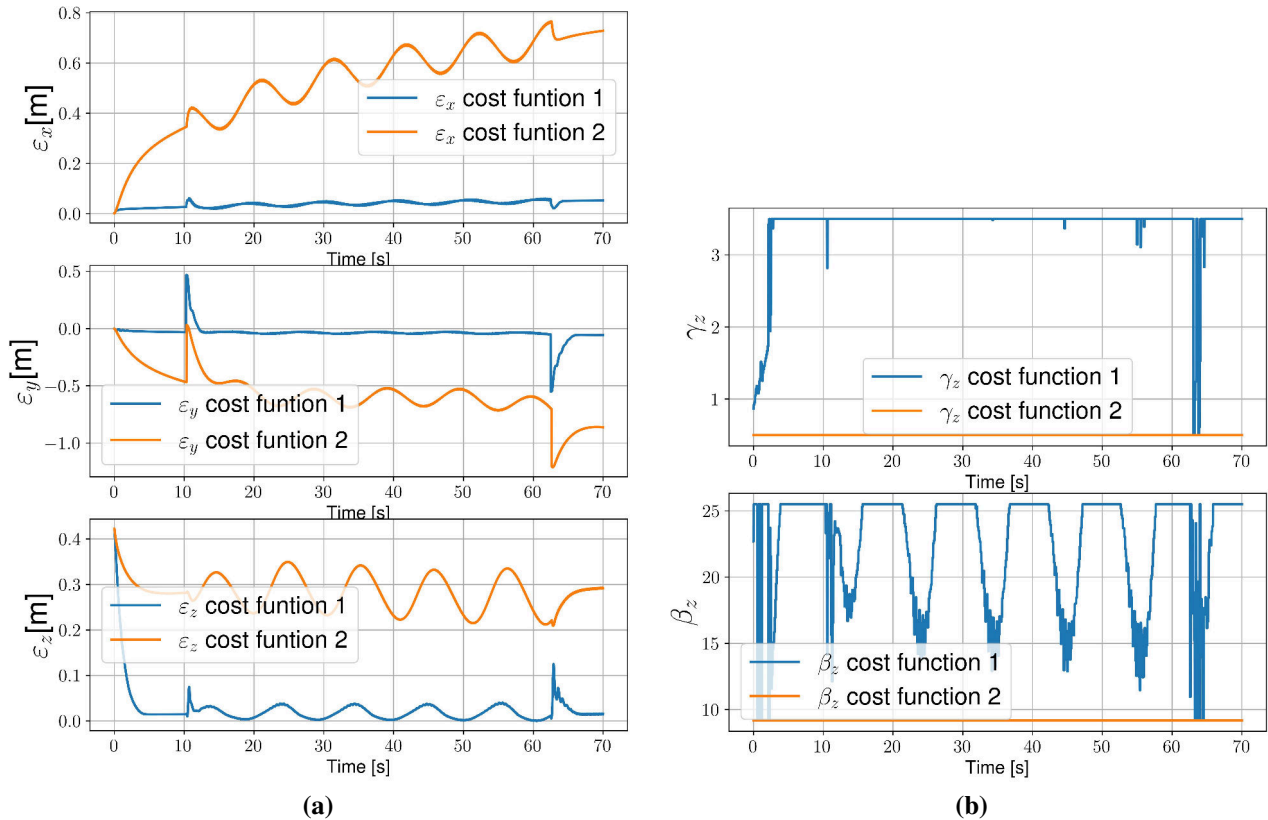


Figure 4.18: (a) Suivi de l'erreur de position en présence de perturbations variables dans le temps pour le MPSMC en tenant compte de deux fonctions de coût différentes, (b) Les différents gains pour la commande MPSMC sur l'axe z en présence de perturbations variables compte tenu de deux fonctions de coût différentes.

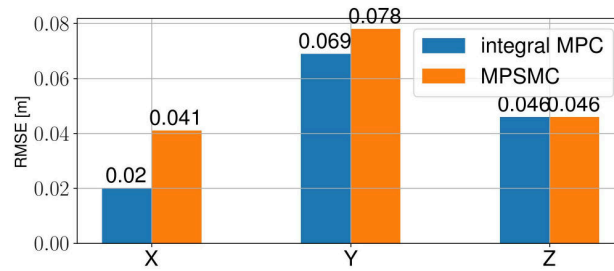


Figure 4.19: Erreur quadratique moyenne des deux contrôleurs proposés basés sur le MPSMC et l'intégral MPC pour la même fonction de coût

Fonction coût	Variable	Value
coût 1	R	$diag(0.025, 0.025, 0.025)$
	\tilde{Q}_x	$diag(127.625, 20.475)$
	\tilde{Q}_y	$diag(127.625, 20.475)$
	\tilde{Q}_z	$diag(127.625, 20.475)$
coût 2	R	$diag(127.75, 127.75, 127.75)$
	\tilde{Q}_x	$diag(0.125, 0.475)$
	\tilde{Q}_y	$diag(0.125, 0.475)$
	\tilde{Q}_z	$diag(0.125, 0.475)$

Table 4.6: Paramètres différents pour les deux fonctions de coût utilisées

Cas 3 : Perturbation due au vent fort

Dans cette section, la perturbation due au vent se caractérise par une magnitude élevée de la composante variable dans le temps. L'objectif de cet essai est donc d'évaluer la robustesse des schémas MPC proposés en présence de perturbations à forte fluctuation. La figure (4.20a) montre l'erreur quadratique moyenne de la trajectoire lors du suivi d'une trajectoire 3D. En considérant la fonction de coût 1 pour l'intégral MPC et le MPSMC, on observe qu'en présence de fortes perturbations variables, la loi de commande (3.76) est plus performante que (3.52). Cependant, dans les mêmes conditions que la figure (4.20b) et compte tenu de la fonction de coût 2, où la commande est plus contrainte que l'erreur de position et de vitesse, la loi de commande (3.52) est plus performante.

Remarque :

Il convient de noter quelques avantages de ces deux approches basées MPC. L'un des avantages majeurs réside dans la facilité avec laquelle on peut imposer des contraintes sur l'erreur de suivi ou sur la commande. Par exemple, si l'on souhaite améliorer le suivi, il suffit d'augmenter la contrainte sur l'erreur. Un autre avantage important est que les gains sont calculés en prenant en

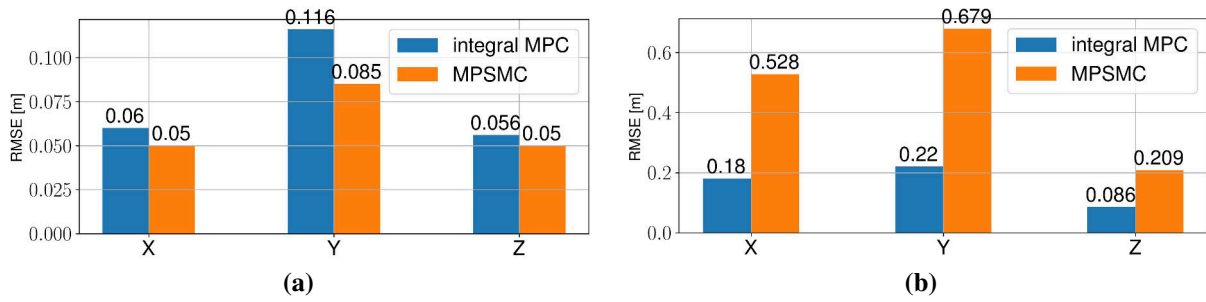


Figure 4.20: (a) Erreur quadratique moyenne des deux contrôleurs proposés basés sur le MPSMC et l'intégral MPC pour la même fonction de coût (coût 1), en présence de perturbation avec une forte composante dynamique, (b) Erreur quadratique moyenne des deux contrôleurs proposés basés sur le MPSMC et l'intégral MPC pour la même fonction de coût (coût 2), en présence de perturbation avec une forte composante dynamique.

compte à la fois les perturbations et l'erreur. Comme on peut le remarquer dans la figure (4.21), dans le cas non perturbé à gauche, lorsque l'erreur de suivi est grande, un gain faible suffit à générer une commande suffisante pour la réduire. Cependant, en présence de perturbations, le gain kp_x est maintenu pour compenser l'effet de la perturbation sur l'erreur de suivi. Il est à noter que dans ce cas, seule l'action proportionnelle est utilisée.

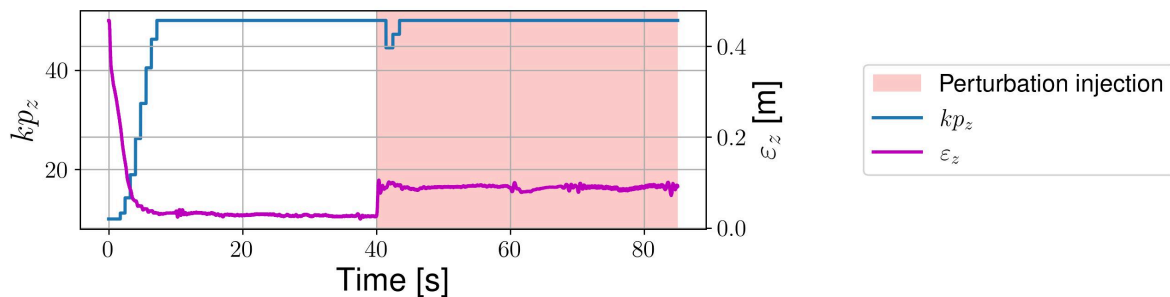


Figure 4.21: Erreur de suivi et gain kp_z pour l'axe z en utilisant le contrôle intégral MPC, avec l'introduction d'une perturbation constante à l'instant $t = 40$

4.2.3 ES-LKF & STO

Dans cette partie une brève comparaison sera établie entre le filtre de Kalman linéaire et le Super Twisting Observer, pour l'estimation des perturbations. Les deux observateurs sont soumis aux mêmes conditions et à la même fréquence d'échantillonnage. Ils partent d'une condition initiale nulle pour la perturbation estimée, le drone subit différentes sources de perturbations qui sont les suivantes : une incertitude sur la masse du drone et une perturbation constante est brusquement introduite au temps $t = 40s$, qui est maintenue jusqu'à $t = 80s$. La figure

(4.22) représente l'estimation de la perturbation pour les deux observateur. On remarque que le STO converge plus rapidement que le ES-LKF, pour limiter le *chattering* pour le STO un filtre passe bas est appliqué à la sortie du STO.

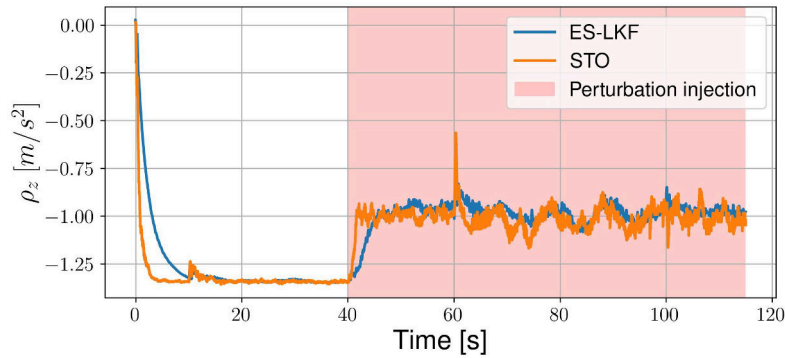


Figure 4.22: Perturbation estimée, une incertitude sur la masse du drone et une perturbation constante est introduite à l'instant $t = 40s$.

Il convient de noter que le STO apporte une certaine amélioration en réduisant l'erreur de suivi, comme le montre la figure (4.23a). Dans des conditions identiques et en suivant la même trajectoire, le STO présente une erreur quadratique moyenne de 0.050 m, tandis que l'ES-LKF affiche 0.054 m. Cette amélioration est due à la rapidité de convergence du STO, qui réduit rapidement l'erreur d'estimation de la perturbation. Comme le montre la figure (4.23b), cette différence de convergence affecte également les valeurs des gains, notamment dans le cas de l'ES-LKF, où les gains ont un retard par rapport au STO.

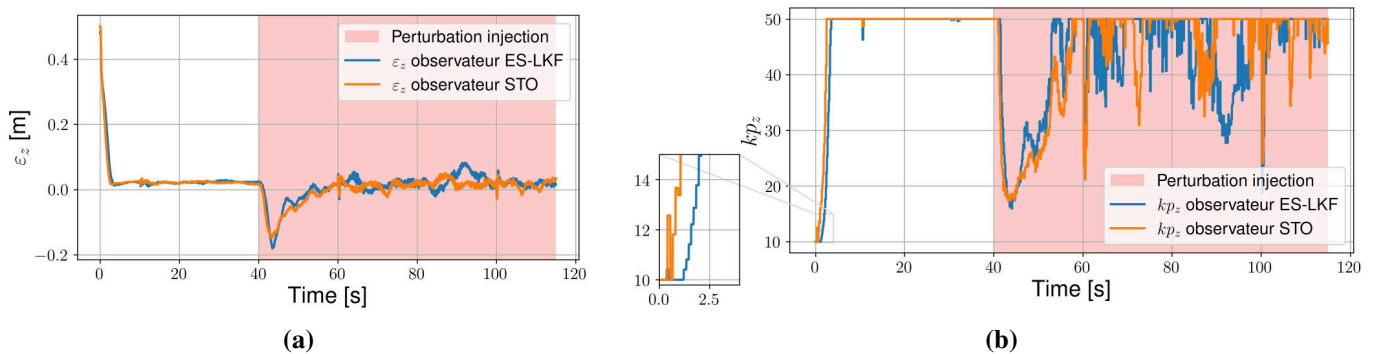


Figure 4.23: (a) Erreur de suivi axe z pour l'intégral MPC en présence de perturbation en utilisant deux observateurs différents, (b) gain kp_z pour l'intégral MPC en présence de perturbation en utilisant deux observateurs différents.

4.3 Conclusion

Dans ce chapitre, nous avons examiné en détail les lois de commande proposées et évalué leur robustesse face aux perturbations à travers divers scénarios de simulation. Pour ce faire, nous avons utilisé différents environnements de simulation.

Ces environnements de simulation reproduisent fidèlement la réalité en modélisant non seulement la physique du système (dynamique, aérodynamique, etc.), mais également l'architecture embarquée, y compris l'autopilote, les capteurs, les actionneurs et d'autres composants essentiels. Ainsi, ils fournissent une représentation précise et complète de la réalité que nous cherchons à simuler.

5

Prototypes et résultats expérimentaux

Sommaire

5.1	Évaluation de l'approche d'estimation et de contrôle pour l'ES-iLQG	121
5.1.1	Cas 2D utilisant le drone Crazyflie	122
5.1.2	Cas 3D utilisant le drone Coex Clover 4	127
5.2	Validation expérimentale du drone à bras pivotant Xtilt	135
5.2.1	Architectures embarquées	135
5.2.2	Scénarios et résultats expérimentaux	141
5.3	Conclusion	147

PROTOTYPES EXPÉRIMENTAUX- Après avoir réalisé une première validation en simulation dans le chapitre précédent, ce chapitre offre une présentation détaillée de l'évaluation expérimentale des différentes approches de commande et des prototypes utilisés ou développés, ainsi que de leurs caractéristiques. Il expose également les résultats expérimentaux obtenus en appliquant les lois de commande dans différentes conditions expérimentales.

Trois plateformes de drones ont été mises en œuvre pour cette étude. Parmi elles, deux suivent une architecture classique : des quadrirotors à bras fixes. Le Crazyflie 2.1 de Bitcraze, et le Clover 4 de la marque Coex. La troisième plateforme, nommée Xtilt, a été élaborée et fabriquée au sein du laboratoire. Toutes sont contrôlées sous l'environnement ROS.

Ce chapitre propose ainsi un aperçu complet des dispositifs expérimentaux utilisés, établissant un lien concret entre les simulations préliminaires et les essais pratiques sur des plateformes réelles.

5.1 Évaluation de l'approche d'estimation et de contrôle pour l'ES-iLQG

Nous commençons cette étude par la validation de l'ES-iLQG, en utilisant deux architectures classiques de drones (quadricoptères à bras fixes). L'objectif principal est d'évaluer la robustesse de l'estimation et la capacité de rejet des perturbations. Par la suite, nous aborderons deux cas spécifiques :

- **Cas 1** : La plateforme expérimentale Crazyflie est contrôlée uniquement dans le plan (yz).
- **Cas 2** : La plateforme expérimentale Clover 4 est contrôlée en 3D.

La figure (5.1) illustre la structure de cette évaluation.



Contrôleur	Plateforme	Évaluation
ES-iLQG		<ul style="list-style-type: none"> • 2D (yz) • Avec un modèle identifié • Erreur de suivi <ul style="list-style-type: none"> ◦ Présence de perturbations <ul style="list-style-type: none"> ▪ Sans et avec compensation • Validation de l'estimation
		<ul style="list-style-type: none"> • 3D (yz) • Sans identification du modèle • Erreur de suivi <ul style="list-style-type: none"> ◦ Présence de perturbations <ul style="list-style-type: none"> ▪ Sans et avec compensation • Validation de l'estimation

Figure 5.1: Crazyflie 2.1 avec marqueur Optitrack

5.1.1 Cas 2D utilisant le drone Crazyflie

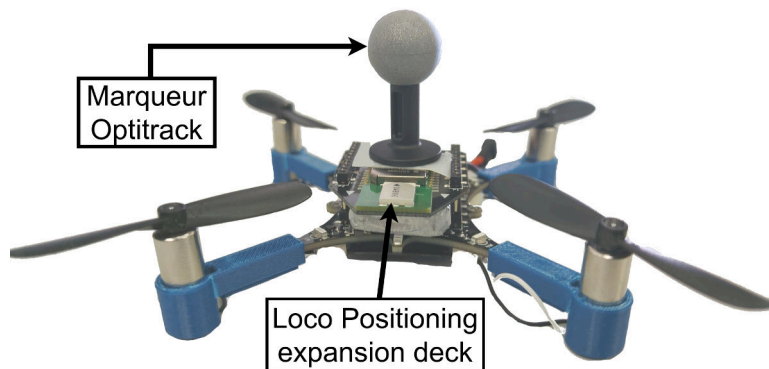


Figure 5.2: Crazyflie 2.1 avec un marqueur Optitrack

Le drone Crazyflie 2.1 est un microquadricoptère de Bitcraze, son code est en libre accès. Ce dernier est équipé d'une radio à longue portée et faible latence, ce qui offre la possibilité de le contrôler à distance. Il existe plusieurs bibliothèques qui permettent de développer du code en python ou C/C++ à des fins de contrôler le drone, comme la bibliothèque officielle sous python [Crazyflie Micro Quadcopter library API](#) et le package C/C++ [Crazyflie_cpp](#) qui contient les bibliothèques pour la radio (Crazyradio) et le drone.

Le code source du [firmware](#) utilisé sur la plateforme est en libre accès, ce qui permet si l'on désire de modifier le contrôleur bas niveau utilisé et d'agir directement à la source ou de modifier par exemple les paramètres physiques (masse...), gains des contrôleurs PID ou même le type de filtre, Kalman ou complémentaire.

Architecture du système

Pour cette première plateforme expérimentale, l'architecture globale se présente de la manière suivante figure (5.3). La communication entre le Crazyflie et l'ordinateur s'effectue via un module RF, le Crazyflie driver permet de rendre disponible les informations reçues par le module RF en ROS topics. En effet, il offre la possibilité de récupérer l'odométrie du drone ainsi que d'envoyer une poussée et des orientations désirées en passant par des ROS topics. Le contrôleur en position ainsi que l'observateur des perturbations sont implémentés en C++ sous l'environnement ROS sur le PC distant. En ce qui concerne le contrôleur en orientation, ce dernier est embarqué sur le Crazyflie. La position est mesurée de deux manières différentes via le capteur flux optique (**Flow deck V2**) ou via le **Loco Positioning system** en utilisant des balises Decawave UWB (ultra wideband).

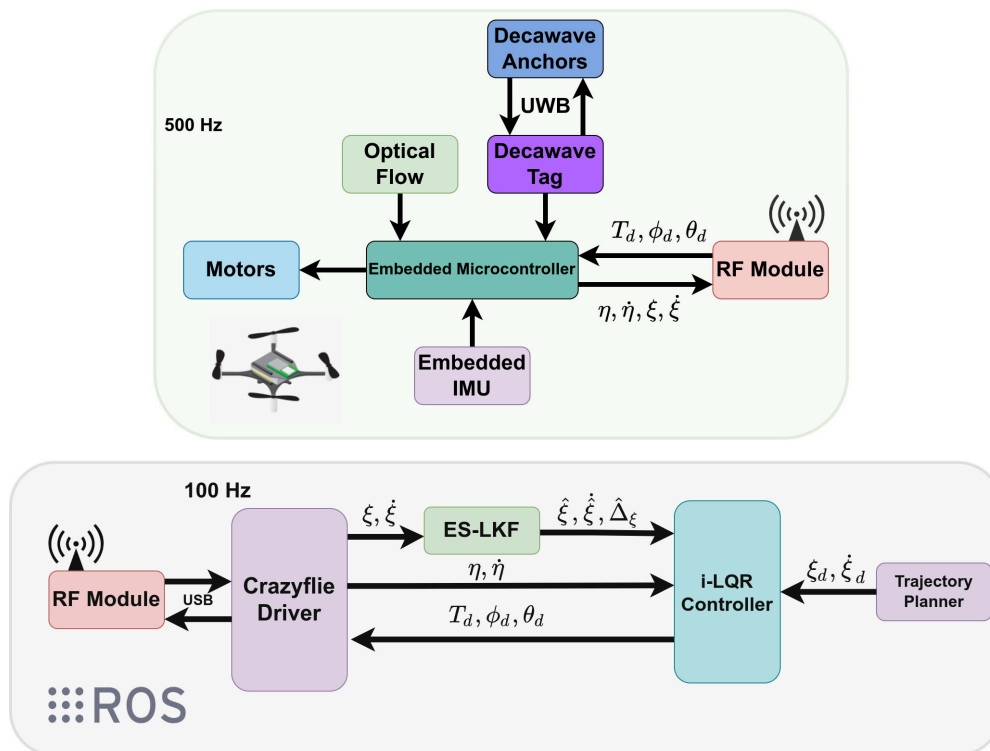


Figure 5.3: Architecture du système Crazyflie 2.1

Scénarios résultats et expérimentaux

La présente section s'attache à décrire la méthodologie et les résultats expérimentaux pour le contrôleur (ES-iLQR) introduit dans la section (3.3) et résumé dans la figure (4.2). L'objectif est d'amasser une compréhension approfondie des performances en vol du drone soumis à différentes conditions. En outre, une évaluation de l'impact des perturbations sur la précision du contrôle

en position ainsi que la réaction du contrôleur face à ces perturbations.

Dans les scénarios expérimentaux, trois cas de figures ont été testés.

- **Scénario 1, sans perturbation avec compensation** : Un premier suivi d'une trajectoire consigne sans présence de perturbations externes, ce qui représente une référence pour le reste des tests.
- **Scénario 2, avec perturbation sans compensation** : Le drone suit la même trajectoire du scénario précédent, mais on introduit une perturbation constante externe au système. La perturbation introduite est une masse centrée d'environ 2 grammes (poids du marqueur Optitrack), ce qui représente 7.5% de la masse du drone. Comme on peut voir dans la figure (5.2), le contrôleur ignore la nouvelle masse du drone. Il faut aussi noter qu'aucune compensation ne sera appliquée pour contrer cette perturbation, c'est-à-dire l'estimation de la perturbation n'est pas injectée dans la commande.
- **Scénario 3, avec perturbation avec compensation** : Pour le troisième scénario, la différence avec le deuxième scénario réside dans le fait que cette fois-ci l'estimation de la perturbation est injectée dans la commande pour compenser les perturbations.

Il convient de noter que, pour l'ensemble de ces scénarios, le contrôle de la position est réalisé exclusivement dans le plan 2D (yz). Afin de se rapprocher au mieux de la réalité physique, le modèle employé pour estimer l'état et les perturbations, ainsi que celui utilisé pour le processus d'optimisation iLQR, a été déterminé par le biais d'une identification en boucle ouverte du système. Le Tableau (5.1) résume les différents paramètres pour l'ES-LKF et le iLQR.

	Description	Valeur
Modèle	Matrice d'état (A)	$\begin{pmatrix} 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & -0.4244 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & -0.0393 & -0.2751 \end{pmatrix}$
	État	$\left(\int \varepsilon_y \quad \varepsilon_y \quad \dot{\varepsilon}_y \quad \int \varepsilon_z \quad \varepsilon_z \quad \dot{\varepsilon}_z \right)^T$
Kalman	\mathbf{w}_k covariance (Q)	$diag(7e-1, 1e-3, 4.5e-2, 1e-3, 1.5e-4, 2.5e-4)$
	$\bar{\mathbf{w}}_k$ covariance (R)	$diag(1e-2, 1e-1, 1e-1, 1e-1)$
LQR	Coût État (\tilde{Q})	$diag(1.15, 2.5e-2, 4.5e-3, 2.55, 2.4e-2, 5.5e-3)$
	Coût Contrôle (\tilde{R})	$diag(0.1, 0.1)$

Table 5.1: Paramètres du filtre de Kalman et du LQR pour les expériences sur le Crazyflie

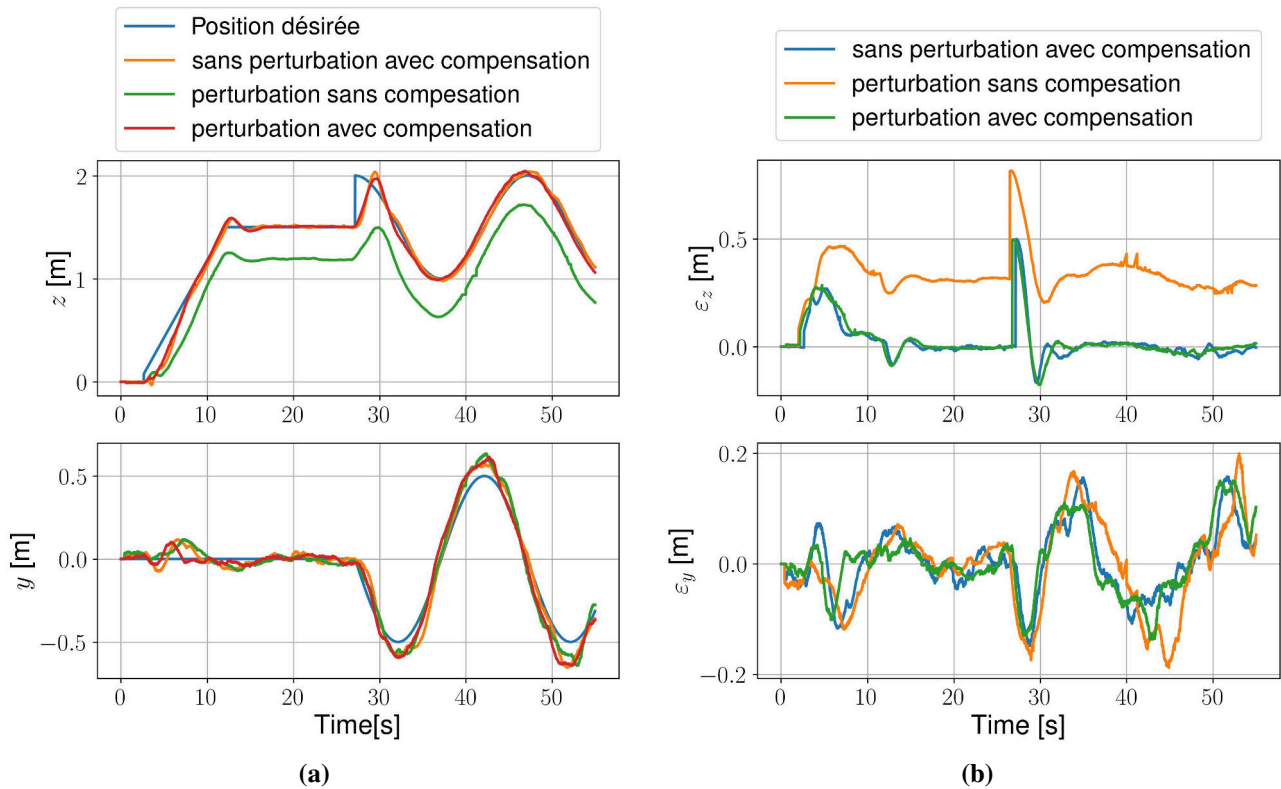


Figure 5.4: (a) Suivi d'une trajectoire 2D circulaire avec et sans présence de perturbations constantes, décrite par $y_d = 0.5 \cos(0.01\pi t)$, $z_d = 1.5 + 0.5 \sin(0.01\pi t)$, (b) Erreur de suivi d'une trajectoire circulaire, décrite par $y_d = 0.5 \cos(0.01\pi t)$, $z_d = 1.5 + 0.5 \sin(0.01\pi t)$. <https://youtu.be/sdiZnxvUEC4> https://youtu.be/kCSuzYDq_f0

Les figures suivantes (5.4a) et (5.4b) illustrent les trajectoires de référence, les trajectoires réelles du drone, ainsi que les erreurs de suivi observées pour chacun des trois scénarios. Il est intéressant de constater que les deux scénarios 1 et 3 ont le même comportement, même en présence de perturbations le comportement du drone demeure relativement proche de celui du scénario non perturbé. De plus, il est évident que l'absence de compensation des perturbations entrave la capacité du contrôleur à annuler l'effet de la perturbation et à éliminer l'écart par rapport à la trajectoire de référence.

Par ailleurs, il est essentiel de souligner que l'effet de la perturbation constante a un impact minime sur le suivi de la consigne de vitesse. Cela est corroboré par les observations tirées des deux graphiques (figure (5.5a) et figure (5.5b)) qui démontrent une similarité presque totale dans les comportements pour les trois scénarios.

En somme, les données présentées dans les figures mettent en lumière la capacité du système à maintenir une performance stable malgré les perturbations et à illustrer l'importance de la compensation pour atténuer les erreurs de suivi. De plus, l'influence négligeable de la perturbation

constante sur le suivi de la vitesse renforce la robustesse du système.

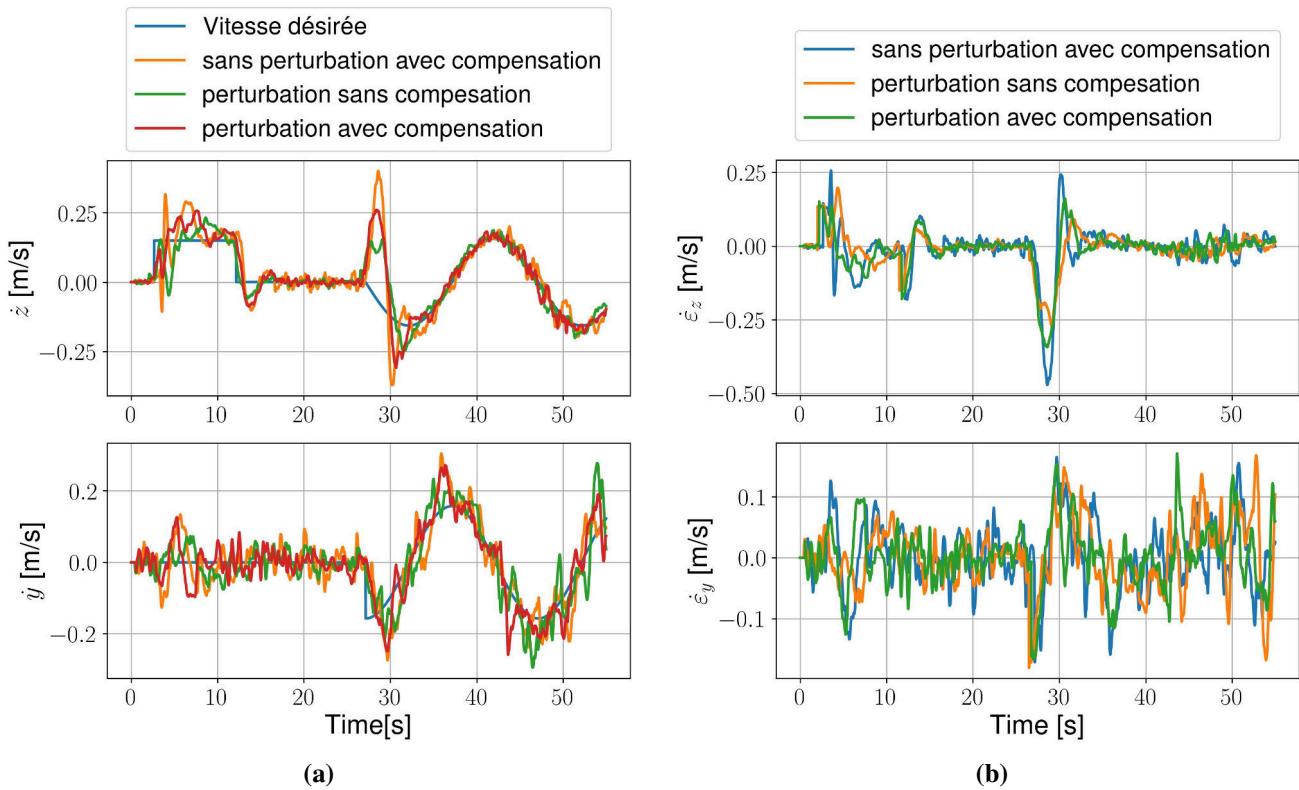


Figure 5.5: (a) Suivi d'une trajectoire en vitesse circulaire avec et sans présence de perturbations constantes, décrite par $\dot{y}_d = -0.005\pi \sin(0.01\pi t)$, $\dot{z}_d = 0.005\pi \cos(0.01\pi t)$, (b) Erreur de suivi d'une trajectoire en vitesse circulaire, décrite par $\dot{y}_d = -0.005\pi \sin(0.01\pi t)$, $\dot{z}_d = 0.005\pi \cos(0.01\pi t)$.

La figure (5.6a) présente l'estimation de la perturbation dans les cas où le système subit ou non des perturbations. Une observation notable est que, lorsque nous excluons l'ajout de la petite masse supplémentaire sur le drone, certaines perturbations persistent dans le système. Cette persistance est attribuable à une variété de facteurs, parmi lesquels figurent l'incertitude liée au modèle, les variations de la tension de la batterie et d'autres sources de perturbation.

L'ajout de la masse au drone, étant relativement bien connu, offre une opportunité de vérifier la précision de l'estimation des perturbations externes. La figure (5.6b) illustre l'estimation de la perturbation en grammes. Le filtre de Kalman initie l'estimation de la perturbation à partir d'une valeur nulle, et au fil du temps, la valeur estimée de la perturbation converge vers une moyenne constante d'environ 2 grammes. Cette courbe a été générée en prenant comme référence le scénario 1. En effet, la simple soustraction de l'estimation de la perturbation dans le scénario 1 de celle dans les scénarios 2 ou 3, en présence de la masse ajoutée, permet d'obtenir une estimation du poids. Cette approche a pour but d'enlever les perturbations omniprésentes dans tous les cas.

L'estimation de la masse ajoutée au drone est donnée par

$$m_{\Delta} = \frac{m_n \Delta \rho_z}{g} \quad \text{avec} \quad \Delta \rho_z = (\rho_z)_{\text{scenario3}} - (\rho_z)_{\text{scenario1}} \quad (5.1)$$

Où m_{Δ} est la masse ajoutée au drone, m_n est la masse nominale du drone et g est la pesanteur. En synthèse, les graphiques offrent un aperçu de l'estimation des perturbations et mettent en évidence la manière dont l'ajout de masse (perturbation constante) influence cette estimation. L'exactitude de la valeur estimée de la perturbation au fil du temps renforce la validité de l'approche utilisée et sa pertinence pour compenser les perturbations.

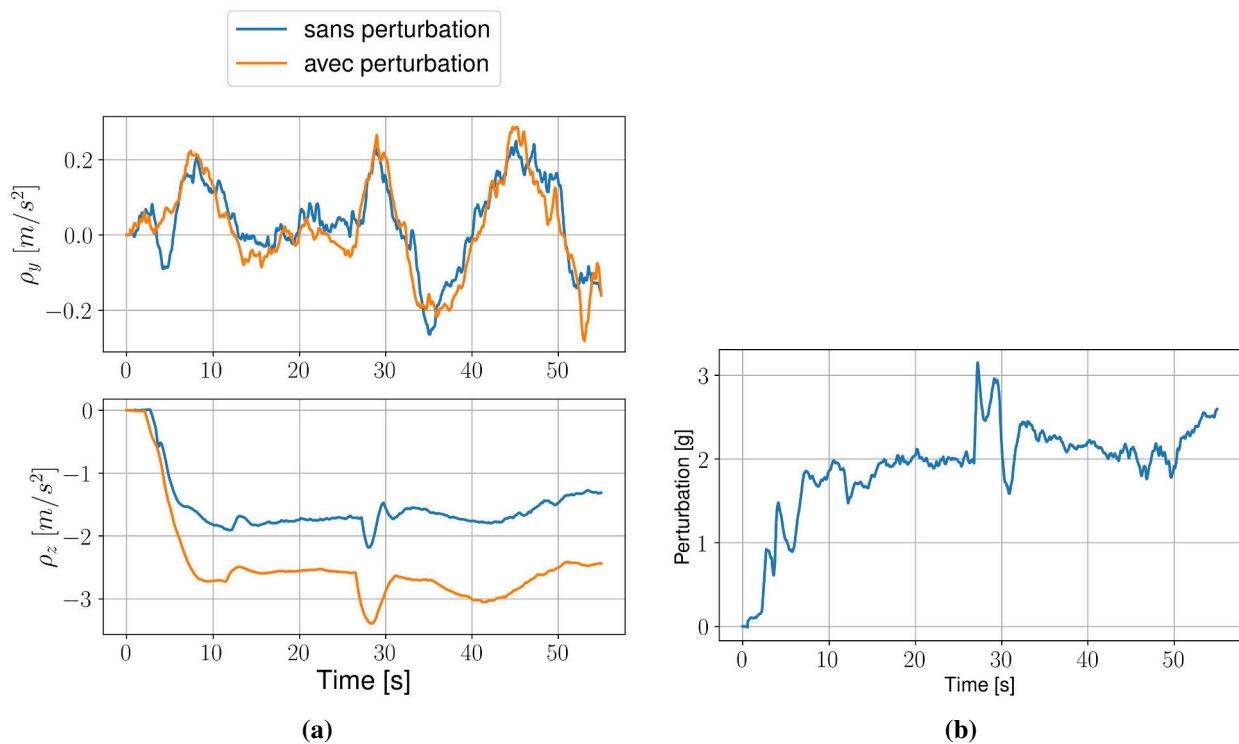


Figure 5.6: (a) Perturbations estimées pour l'axe y et z, (b) Perturbation estimée en [g].

5.1.2 Cas 3D utilisant le drone Coex Clover 4

Cette section constitue un complément à la partie expérimentale précédente. En effet, l'architecture qui suit présente une configuration similaire à celle du Crazyflie, bien que quelques différences existent en termes de caractéristiques physiques, (la masse, l'inertie, le système de propulsion...) et aussi d'architecture embarquée. À la différence du cas précédent, le contrôleur ES-iLQG est mis à l'épreuve dans un contexte 3D, où les déplacements selon les trois axes sont pris en compte.

Architecture du système

Pour réaliser l'étape expérimentale, nous avons utilisé une plateforme Coex Clover 4.2 figure (5.8 et 5.12). Le drone Coex est composé d'un contrôleur de vol autopilote PX4, d'une carte Raspberry Pi4, d'une caméra embarquée Raspberry Pi-v2.1 et d'un capteur d'altitude ToF VL53L1x. Les paramètres physiques intrinsèques du drone sont indiqués dans le tableau 4.2. L'architecture globale du système est décrite dans la figure (5.7). Le contrôleur de position et l'estimateur du filtre de Kalman ont été mis en œuvre sur le micro-ordinateur Raspberry Pi4 codé en C++ sous le logiciel intermédiaire ROS. Le pilote automatique PX4 gère le contrôle d'attitude du drone sur la base des valeurs de contrôle souhaitées communiquées via l'interface de communication MAVROS. La localisation du drone est assurée par la caméra embarquée où la vitesse du drone dans le plan horizontal xy est obtenue en calculant le flux optique à l'aide d'une corrélation de phase entre deux images superposées.

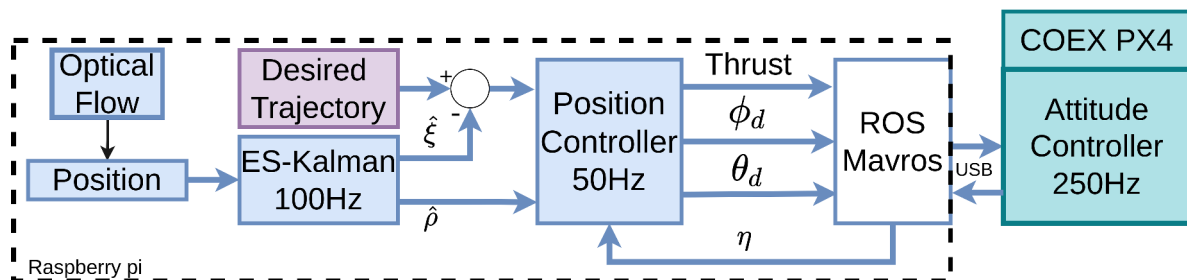


Figure 5.7: Architecture embarquée pour le drone Clover 4.2. Le pilote automatique PX4 exécute le contrôleur d'attitude à 250 Hz, tandis que le Raspberry Pi intègre la localisation du drone à 50 Hz, le filtre de Kalman linéaire à 100 Hz et le contrôleur de position à 50 Hz.

Scénarios et résultats expérimentaux

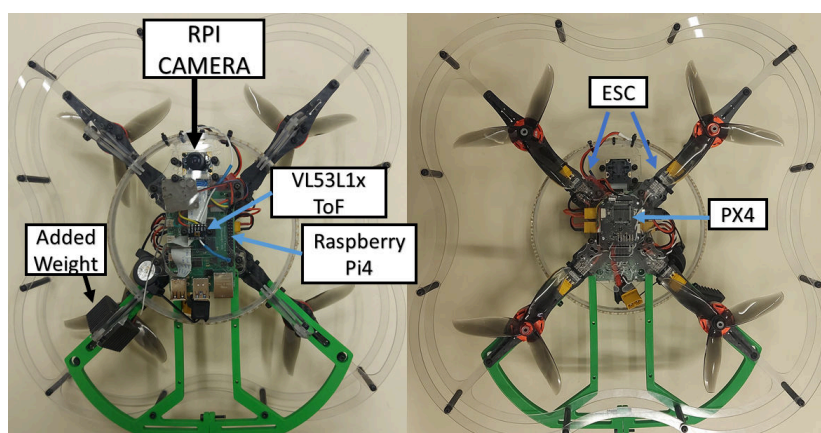


Figure 5.8: Configuration expérimentale du drone Clover 4.2 (vue dessous et dessus).

La première expérience consiste à faire voler le drone sans aucune perturbation extérieure, comme le vent ou toute autre charge supplémentaire. Cette partie sera notre référence pour le reste des tests. Même en l’absence de vent et de surpoids, d’autres types de perturbations persistent, comme les frottements de l’air dus au déplacement du drone, ainsi que les incertitudes de mesure de paramètres physiques tels que le poids, sans oublier que notre modèle est incomplet. Il est à noter que les approches habituelles s’appuient sur une identification du modèle du système pour avoir la représentation de l’état. Ici, nous utilisons le même modèle que dans la modélisation théorique, l’incomplétude du modèle est compensée par l’estimation de la perturbation.

Dans la deuxième expérience, le drone est chargé d’un poids d’environ 52g qui représente 5.8% du poids du drone. L’intérêt est d’évaluer si le filtre de Kalman serait capable d’estimer cette perturbation et de la compenser au niveau de la commande. A titre de comparaison, un autre test a été réalisé, voir figure (5.9a), mais cette fois, l’estimation de la perturbation n’est pas injectée dans la commande. Deux types de trajectoires ont été utilisés dans cette série d’expériences, une première trajectoire hélicoïdale et une trajectoire en forme de huit. Pour toutes les expériences, les paramètres de l’estimateur de Kalman et du régulateur linéaire quadratique sont indiqués dans le tableau (5.2).

	Description	Valeur
Modèle	Matrice d’état (A)	$\begin{pmatrix} 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \end{pmatrix}$
	État	$\left(\int \varepsilon_x \quad \varepsilon_x \quad \dot{\varepsilon}_x \quad \int \varepsilon_y \quad \varepsilon_y \quad \dot{\varepsilon}_y \quad \int \varepsilon_z \quad \varepsilon_z \quad \dot{\varepsilon}_z \right)^T$
Kalman	\mathbf{w}_k covariance (Q)	$diag(2e-3, 1, 5e-2, 2e-2, 1.5e-2, 1e-2, \dots, 1.5e-2, 2.5e-6, 1.5e-6, 8.5e-6)$
	$\bar{\mathbf{w}}_k$ covariance (R)	$diag(1.7e-4, 1.7e-4, 1.7e-4, 1.7e-4, \dots, 1.7e-4, 1.7e-3)$
LQR	Coût État (\tilde{Q})	$diag(2.45, 9.5e-2, 6.5e-3, 2.45, 9.5e-2, \dots, 6.5e-3, 2.955, 8.5e-2, 5.5e-3)$
	Coût Contrôle (\tilde{R})	$diag(0.1, 0.1, 0.1)$

Table 5.2: Paramètres du filtre de Kalman et du LQR pour le drone Clover 4.2

Les erreurs de suivi sont données dans la figure (5.9b). On peut voir qu'un simple contrôleur intégral-proportionnel-dérivé a compensé la perturbation statique. Cependant, cette erreur est compensée dans un temps plus long que dans la troisième expérience où la perturbation estimée est réinjectée dans la commande.

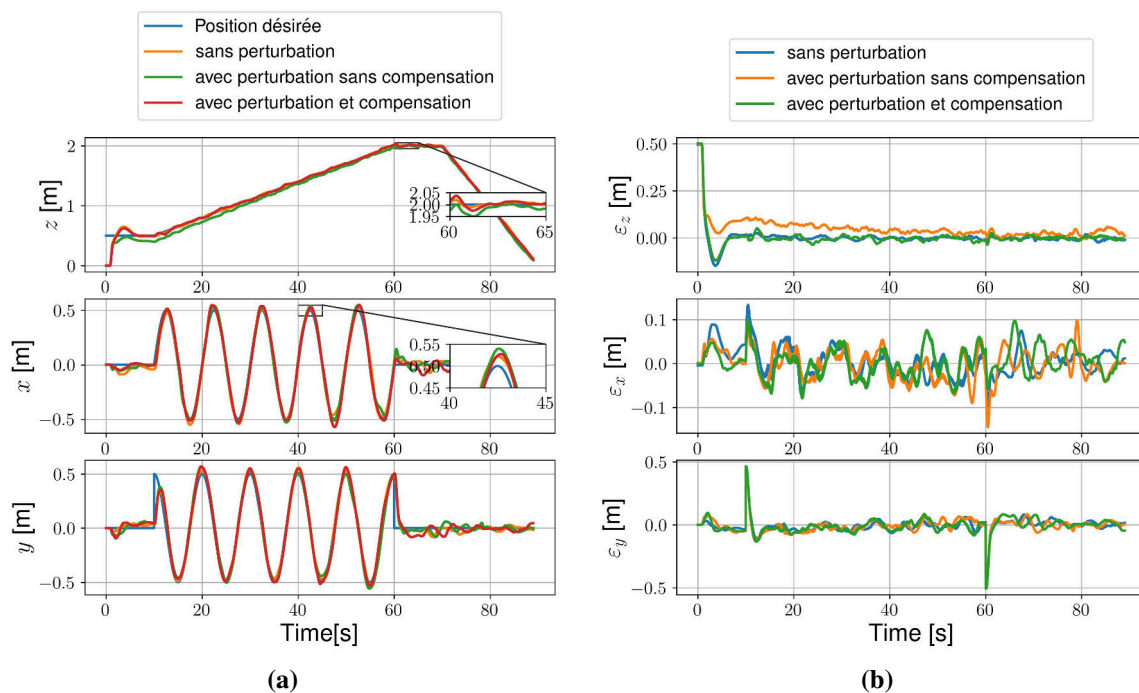


Figure 5.9: (a) Suivi d'une trajectoire 3D hélicoïdale, décrite par $x_d = 0.5 \sin(0.2\pi t)$, $y_d = 0.5 \cos(0.2\pi t)$, $z_d = 0.5 + 0.03t$, (b) Erreurs de suivi de position de la trajectoire hélicoïdale.

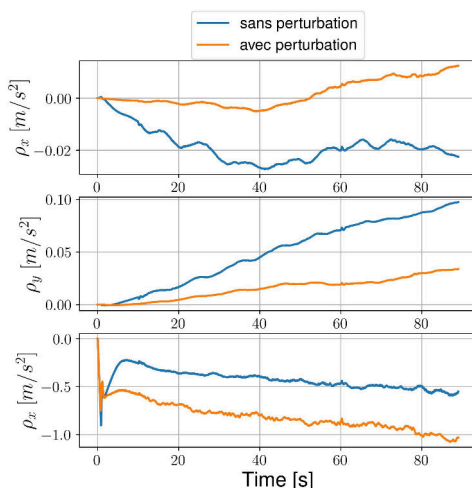


Figure 5.10: Estimation de la perturbation subie par le drone dans le référentiel fixe. Elle est estimée en termes d'accélération, la multiplication par la masse nominale donne l'équivalent en Newtons.

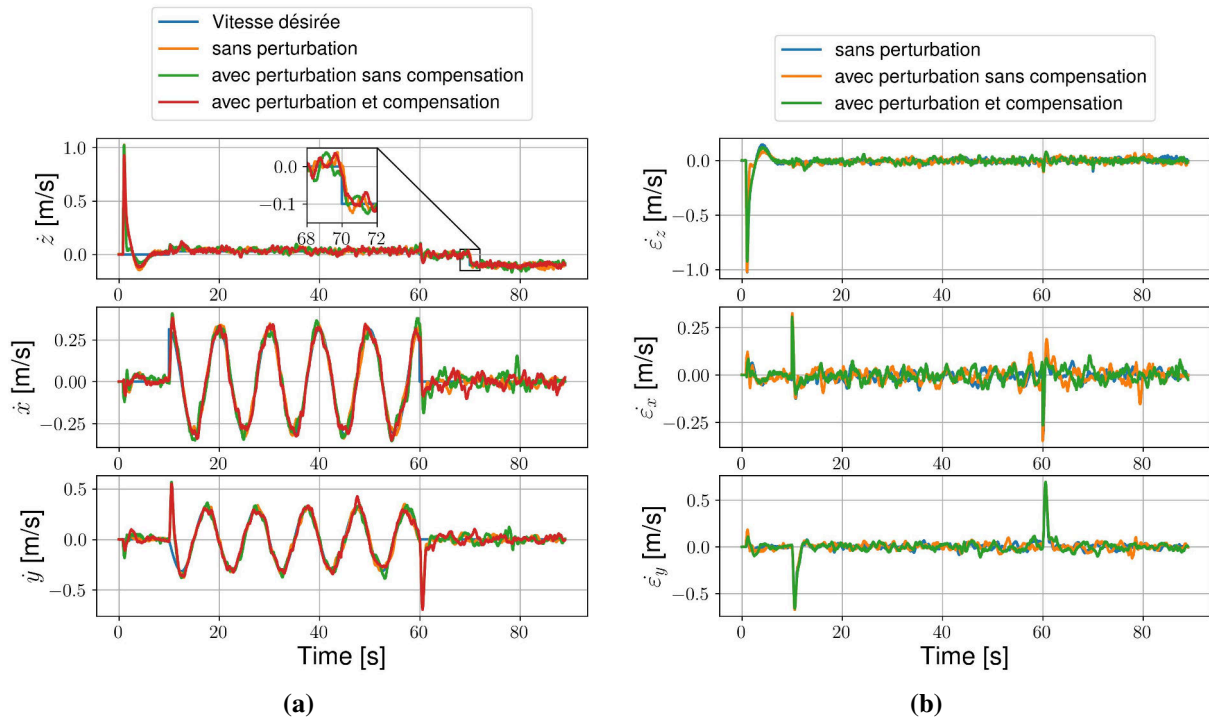


Figure 5.11: (a) Suivi de la trajectoire de la vitesse hélicoïdale sous différentes perturbations, la vitesse désirée est $\dot{x}_d = 0.1\pi \cos(0.2\pi t)$, $\dot{y}_d = -0.1\pi \sin(0.2\pi t)$, $\dot{z}_d = 0.03m/s$, (b) Erreurs de suivi de la vitesse de la trajectoire hélicoïdale.



Figure 5.12: Une séquence de photos du drone en suivie de la trajectoire hélicoïdale. (Vidéo <https://youtu.be/M1x91ecZ5Ek>).

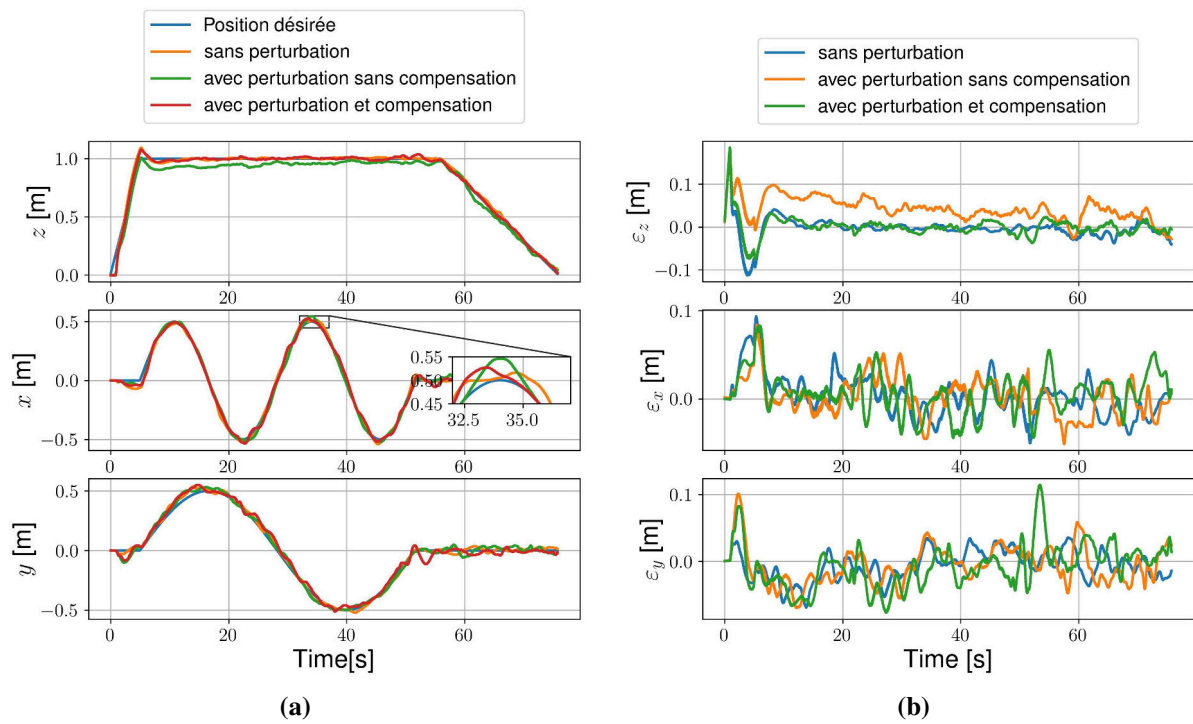


Figure 5.13: (a) Suivi d'une trajectoire en forme de huit $x_d = 0.5 \sin(0.135\pi t)$, $y_d = 0.5 \sin(0.27\pi t)$ et $z_d = 1$ m, (b) Erreurs de suivi de la position de la trajectoire en forme de huit.

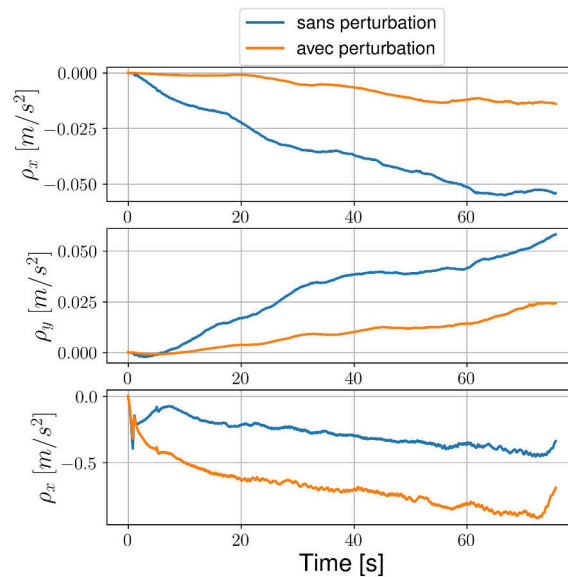


Figure 5.14: Estimation des perturbations lors du suivi de la trajectoire en forme de huit

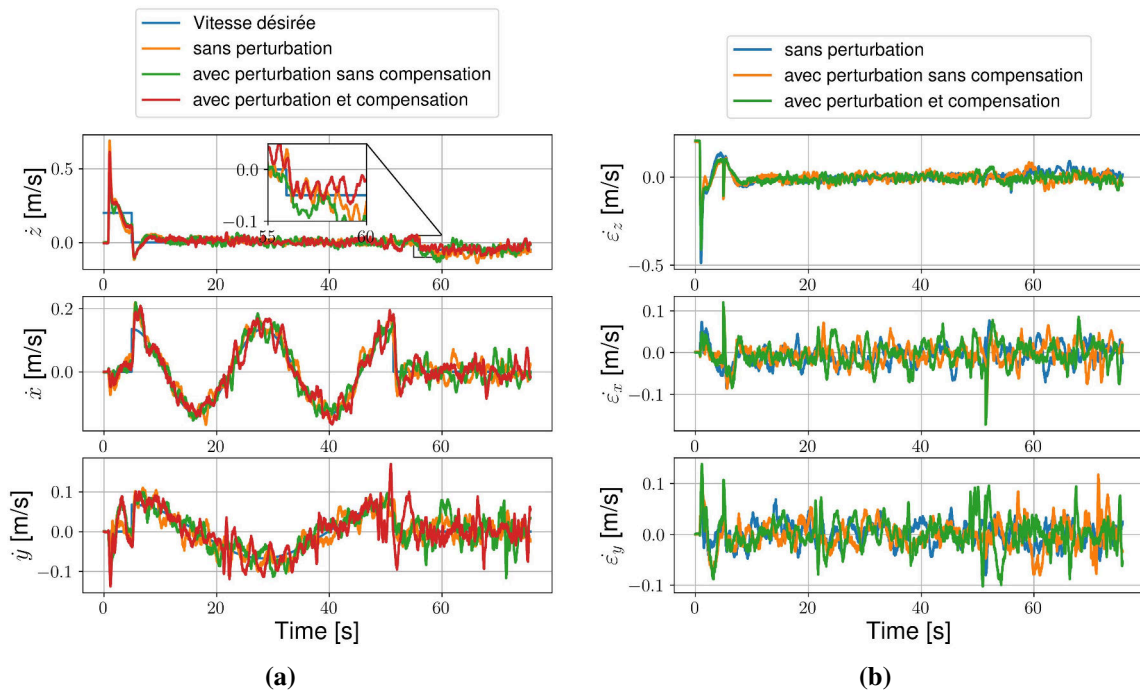


Figure 5.15: (a) Suivi de la trajectoire de vitesse en forme de huit sous différentes perturbations, la vitesse souhaitée est la suivante $\dot{x}_d = 0.675\pi \cos(0.135\pi t)$, $\dot{y}_d = 0.135\pi \cos(0.27\pi t)$, (b) Erreurs de suivi de la vitesse de la trajectoire en forme de huit.

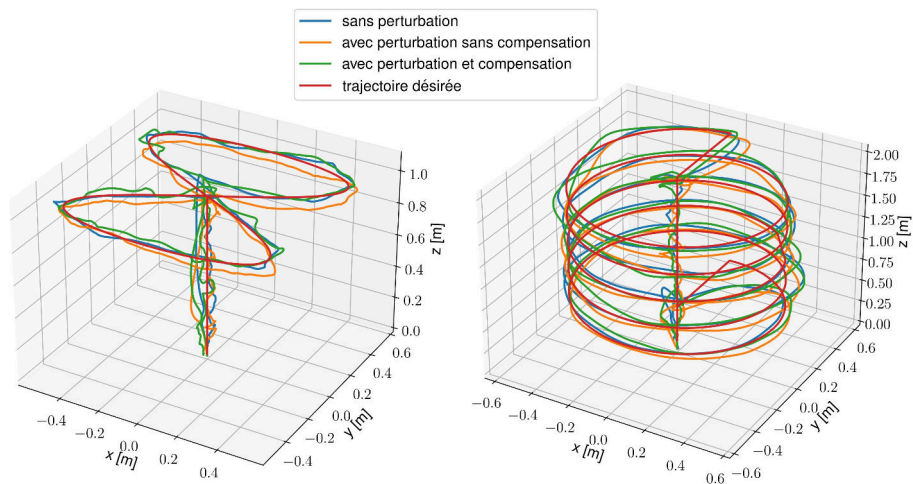


Figure 5.16: Suivi de la trajectoire 3D du drone examiné dans les scénarios expérimentaux, avec et sans compensation en direct des perturbations.

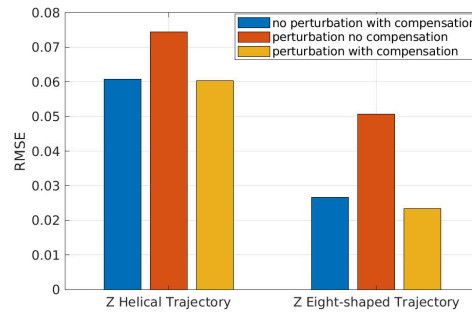


Figure 5.17: Ecart quadratique moyen de la trajectoire hélicoïdale. On remarque que l’erreur sur l’axe z est plus importante dans le cas où la perturbation n’est pas compensée.

Dans les figures (5.9a) et (5.11b), nous trouvons la perturbation estimée en termes d’accélération, dans le cas où nous avons ajouté un poids sur le drone, la perturbation estimée est plus grande en norme que pour le cas où nous ne perturbons pas notre système. À noter qu’une perturbation négative résulte d’une force qui agit dans la direction opposée du référentiel fixe. On note également que même dans le premier cas, nous remarquons que la perturbation n’est pas nulle, elle est en fait due à la fluctuation de la tension de la batterie. En effet, comme les moteurs sont contrôlés en boucle ouverte avec un signal PWM, le changement de la tension de la batterie affecte la vitesse de rotation des moteurs et aussi la poussée. Les figures (5.13a) et (5.15b) permettent de conclure la même chose que pour le premier cas mais en suivant une trajectoire différente. Comme nous avons quantifié les autres perturbations, grâce à la première expérience, sans le poids ajouté, nous pouvons calculer la perturbation générée par le poids ajouté, en supposant que les deux expériences ont été faites dans les mêmes conditions (niveau de batterie...). En soustrayant la perturbation estimée sans le poids de la perturbation estimée avec le poids, nous pouvons avoir la perturbation causée uniquement par le poids ajouté (Éq.5.1) et nous faisons une conversion pour avoir l’équivalent de cette perturbation en grammes figure (5.18). On observe que l’estimation de la masse ajoutée est moins précise que dans le cas précédent (Cas 2D avec le Crazyflie). Ceci est principalement dû au fait que, dans le cas précédent, le modèle utilisé pour l’ES-LKF avait été identifié, tandis que dans la situation actuelle, le modèle utilisé est le modèle nominal.

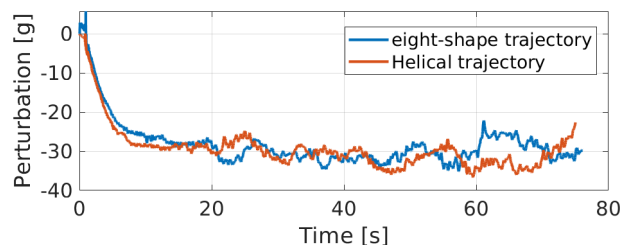


Figure 5.18: Estimation des perturbations dans les deux cas d’études.

5.2 Validation expérimentale du drone à bras pivotant Xtilt



Figure 5.19: Les deux versions du Xtilt, (a) la version basée sur la carte Nvidia Jetson Nano, (b) la version basée sur la carte Raspberry pi4. Il convient également de noter la différence de positionnement des hélices.

Afin de concrétiser la validation de l'architecture Xtilt proposée, une plateforme expérimentale a été développée. Cette section s'attardera en détail sur la conception mécanique et électronique du Xtilt, fournissant une vue approfondie de ces aspects. Au cours du processus de développement du drone, plusieurs architectures embarquées ont été examinées et soumises à des tests expérimentaux. Dans un premier temps, l'architecture globale du drone Xtilt est présentée pour les deux versions développées. Ensuite, nous détaillons les diverses méthodes et moyens utilisés pour identifier les paramètres physiques du drone, tels que l'inertie et les coefficients aérodynamiques des hélices. Enfin, nous procédons à une validation expérimentale des différents contrôleurs proposés, à la fois sur un banc de test et en vol.

5.2.1 Architectures embarquées

Pour la première version du Xtilt (5.19a), le contrôleur de vol embarqué est mis en œuvre sur une carte Jetson Nano, une carte informatique compacte de NVIDIA. Pour garantir une efficacité optimale et réduire au maximum la surcharge système, nous avons conçu un environnement Linux personnalisé avec ROS Noetic à l'aide de BuildRoot[124], adapté spécifiquement aux besoins du drone. Cette approche nous permet de minimiser la taille de l'image système en éliminant les composants superflus et en ne conservant que les éléments essentiels pour assurer le fonctionnement du drone. L'utilisation de cette configuration sur mesure assure une exécution rapide et efficace de l'algorithme de contrôle, tout en préservant les ressources pour les tâches essentielles de vol. En somme, la mise en place d'un environnement Linux personnalisé pour le contrôleur de vol embarqué reflète notre engagement à optimiser chaque aspect de l'architecture pour répondre aux besoins spécifiques du drone tout en maximisant son efficacité opérationnelle. L'algorithme de contrôle fonctionne à 100 Hz et prend en entrée une trajectoire souhaitée pour la position et l'orientation ξ^d, η^d , une vitesse désirée $\dot{\xi}^d, \dot{\eta}^d$, les positions/orientations du Xtilt dans

le repère monde ξ, η ainsi que les vitesses de rotation/linéaire $\dot{\xi}, \dot{\eta}$. L'orientation η du Xtilt est acquise depuis l'IMU Microstrain 3dm-gx5-25 à 500 Hz, communiquant avec la carte Jetson via USB à 921600 bauds. Pour la détection de position ξ , deux systèmes de positionnement différents sont utilisés. Le premier est un système de capture de mouvement (Optitrack) avec un taux de rafraîchissement de positionnement de 200 Hz. Le second utilise une caméra RPi v2 associée à un module de télémétrie laser *vl53l0x* pour obtenir la hauteur afin de calculer le flux optique à 50 Hz. Pour contrôler les moteurs brushless, la Jetson Nano communique via un générateur de signal PWM (PCA9685), avec une résolution de 12 bits pour chaque sortie, ce qui équivaut à une résolution d'environ 4 μ s à 60 Hz, via le port I2C en Fast-mode plus (horloge à 1 MHz). La boucle de contrôle des moteurs brushless fonctionne à 300 Hz. Les informations d'inclinaison des rotors sont envoyées via le port série à 1 Mbps, et les bras sont contrôlés à 100 Hz. La figure 5.20a illustre la répartition des éléments mentionnés précédemment.

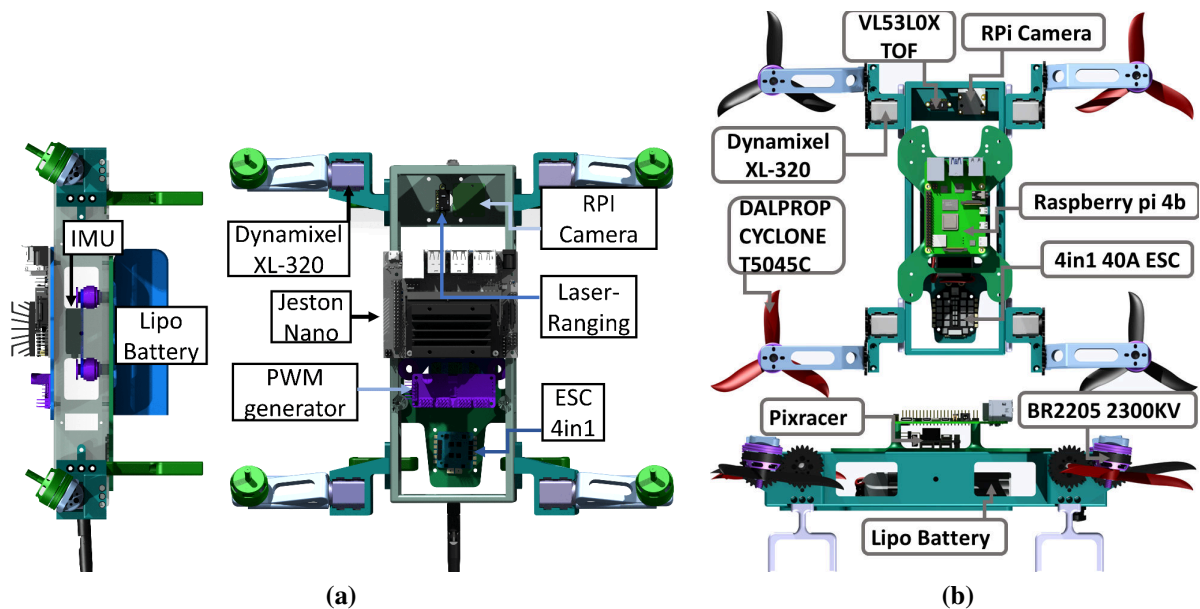


Figure 5.20: Description globale de l'architecture des prototypes expérimentaux (a) version basée sur Nvidia Jetson Nano, (b) version basée sur la carte Raspberry pi4

En ce qui concerne la deuxième version du Xtilt (5.19b), le système embarqué intègre une Raspberry Pi 4B exécutant Debian 11 avec ROS Noetic. Cette carte est connectée d'une part à l'autopilote Pixracer via USB, et d'autre part, elle communique via un port série avec quatre servomoteurs Dynamixel XL-320 qui contrôlent l'orientation des rotors. La caméra Waveshare Fisheye Lens et le capteur de distance à temps de vol VL53L0X sont respectivement connectés au Raspberry Pi via le bus CSI (Camera Serial Interface) et le bus I2C.

Les quatre moteurs brushless sont commandés directement par l'autopilote PX4 [125] à l'aide d'un ESC (contrôleur de vitesse électronique) 4en1 de 40A. Ces moteurs sont pilotés en modulant la

largeur d’impulsion (PWM) par la PixRacer à une fréquence de 400 Hz. Dans cette configuration, le contrôleur en orientation est intégré dans la PixRacer, une configuration spécifique pour le Xtilt a été codé et ajouté à l’autopilote PX4. Cependant, le contrôle en position est exécuté sur la Raspberry Pi 4. La figure (5.20b) présente une vue d’ensemble du système.

Spécificités structurelles

Le volet mécanique du prototype expérimental se compose de plusieurs composants, chacun d’eux étant réalisé en impression 3D avec du Z-Nylon sur une imprimante Zortrax M300 Dual [126]. Le drone présente une masse totale de 1.02 kg, tandis que son inertie a été mesurée à l’aide d’un pendule bifilaire [127]–[129], comme illustré dans la figure (5.21). Le processus de mesure consiste à incliner manuellement le drone d’un léger angle autour de l’axe y , puis à le relâcher. Cette manipulation induit un mouvement de rotation périodique autour de l’axe y , dont la période dépend de l’inertie de l’axe en rotation. L’inertie est calculée selon l’équation :

$$I_{yy} = \frac{mgd^2}{16\pi^2l} t_0^2 \quad (5.2)$$

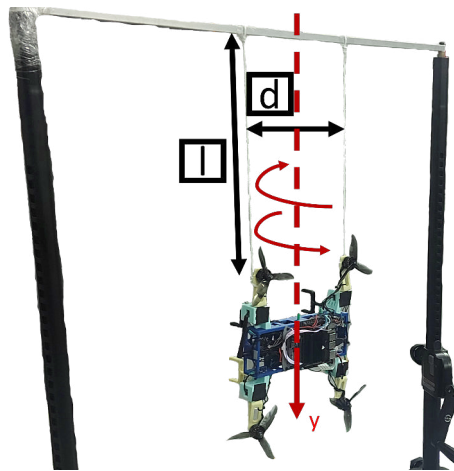


Figure 5.21: Plate-forme de mesure des principaux moments d’inertie du drone à l’aide d’un pendule bifilaire

où m est la masse du drone, l est la longueur du fil, d est la distance entre les deux fils, t_0 est la période du pendule bifilaire. De même, on mesure l’inertie sur les autres axes en faisant coïncider leurs axes avec l’axe d’oscillation du pendule bifilaire, soit

$$I_B = \begin{pmatrix} 6.8e^{-3} & 0 & 0 \\ 0 & 7.9e^{-3} & 0 \\ 0 & 0 & 9e^{-3} \end{pmatrix} \left[\text{kgm}^2 \right] \quad (5.3)$$

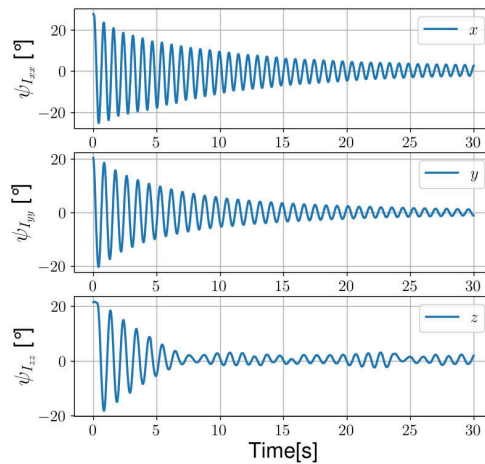


Figure 5.22: Identification des moments d’inertie, la période a été calculée par $t_0 = \frac{t_t}{N}$, t_t est le temps pris par les oscillations, N est le nombre d’oscillations, pour l’axe y $t_0 = 0,8482[s]$.

Pour l’ensemble bras, moteur et hélice, l’inertie a été mesurée à l’aide du modèle CAO.

$$I_{\mathcal{P}_i}^{\alpha_i} = \begin{pmatrix} 2.003e^{-5} & 0 & 0 \\ 0 & 7.77e^{-6} & 0 \\ 0 & 0 & 2.247e^{-5} \end{pmatrix} \left[\text{kgm}^2 \right] \quad (5.4)$$

Système de propulsion

Pour déterminer les valeurs des paramètres aérodynamique b et d , qui correspondent respectivement aux coefficients de poussée et de traînée du rotor, nous avons conçu une banc d’essai complet. Ce dispositif intègre un MPU-9250 pour les relevés de vibrations, deux jauges de contrainte, l’ensemble bras-moteur-hélice-servomoteur ainsi qu’une carte Arduino Due pour le traitement des données, comme illustré sur la figure (5.23).

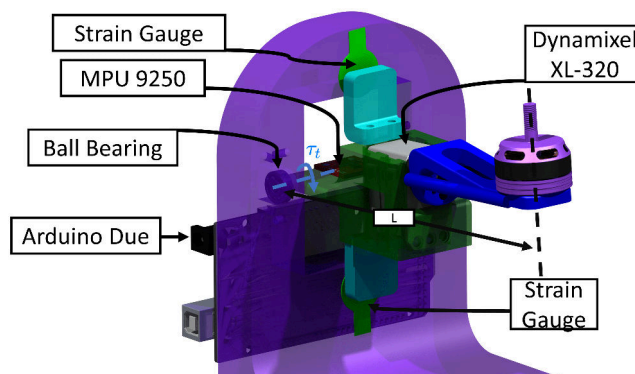


Figure 5.23: Banc d’essai pour caractériser les coefficients de portance et de traînée <https://youtu.be/v9b1Yvb3CFM>

Les deux palettes supérieure et inférieure fixées sur le support du servomoteur sont montées serrées, c'est-à-dire qu'une force nominale est appliquée au préalable sur les jauges supérieure et inférieure.

$$\tau_t = d\omega^2 \sin(\alpha) + b\omega^2 \cos(\alpha)L \quad (5.5)$$

L est la distance entre le vecteur d'action de la poussée et l'axe de rotation, ω est la vitesse de rotation du moteur, α est l'angle d'inclinaison du bras, pour $\alpha = 0$ rad le couple généré par la traînée n'agit pas sur l'axe mesuré, nous mesurerons b avec $\alpha = 0$ rad. Pour $\alpha = \frac{\pi}{2}$ rad la poussée du moteur ne génère aucun couple sur l'axe mesuré, nous mesurerons d avec $\alpha = \frac{\pi}{2}$ rad.

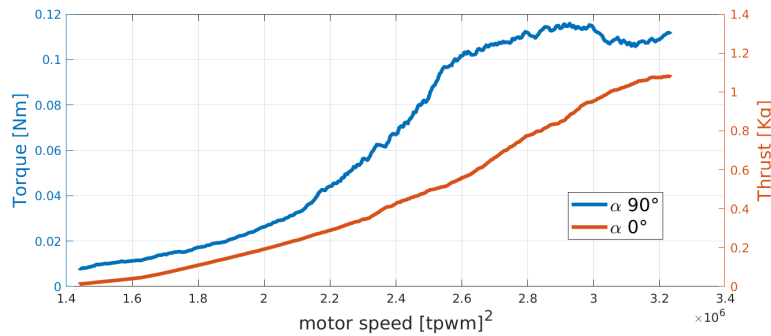


Figure 5.24: Mesures de la poussée et du couple générés par l'hélice en fonction de la vitesse de rotation des moteurs, effectuées sur la plateforme expérimentale.

Un modèle polynomial a été utilisé pour représenter la poussée et la traînée générées par l'hélice en fonction de la vitesse de rotation du moteur. Cette modélisation a été basée sur les données expérimentales mesurées (voir figure 5.24), ce qui a conduit aux résultats suivants :

$$\begin{cases} b = 3.135e^{-7} & \text{kg}/(tpwm)^2 \\ d = 3.8e^{-8} & \text{Nm}/(tpwm)^2 \end{cases} \quad (5.6)$$

$tpwm$ est la largeur d'impulsion du signal PWM (Pulse Width Modulation) multipliée par 1000. En utilisant cette même plateforme d'essai, il est en effet possible de confirmer expérimentalement l'équation établie pour les couples gyroscopiques. Étant donné que les mesures précédentes ont été effectuées pour un angle α constant, les effets des couples gyroscopiques étaient absents. Afin d'annuler l'impact de la force de portance générée par l'hélice sur les couples détectés par les capteurs, nous avons conçu et imprimé en 3D une hélice ayant une inertie équivalente à celle utilisée sur le drone. Cette hélice spécifique ne génère aucune force de portance. La figure (5.25) illustre la configuration expérimentale mise en place.

En faisant tourner le moteur à une vitesse constante, on utilise $\alpha = \frac{\pi}{3} \sin(2\pi t)$ et $\dot{\alpha} = \frac{2\pi^2}{3} \cos(2\pi t)$ comme consignes. Étant donné que l'hélice ne produit aucune portance et en négligeant la traînée qu'elle engendre, on peut exprimer le couple τ_t de la manière suivante :

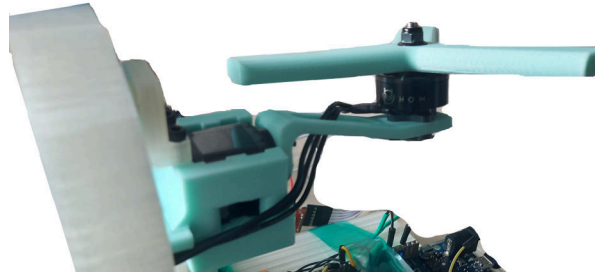


Figure 5.25: Dispositif expérimental pour mesurer le couple gyroscopique.

$$\tau_t = \cos(\alpha)\dot{\alpha}I'\omega \quad (5.7)$$

I' représente l'inertie de l'ensemble moteur-hélice. Étant donné que ω est constante, nous obtenons $\frac{\tau_t}{\cos(\alpha)\dot{\alpha}} = I'\omega = \text{cst}$. Les résultats de cette relation sont présentés dans la figure 5.26, où sont tracés le couple mesuré ainsi que le produit $\cos(\alpha)\dot{\alpha}$.

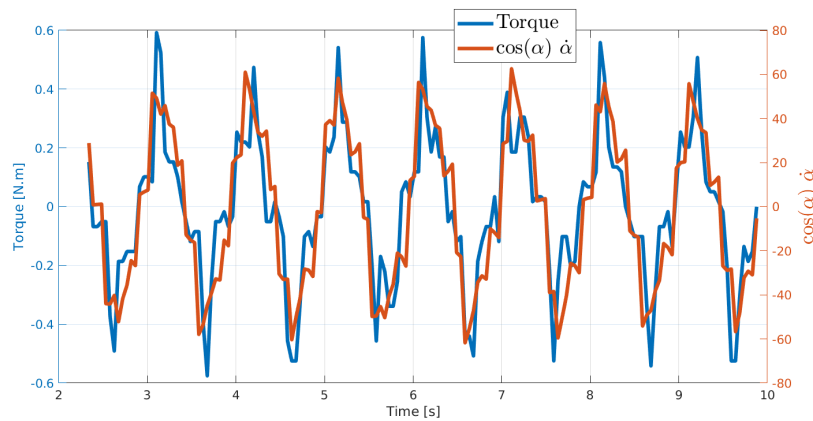


Figure 5.26: Mesure expérimentale du couple généré par l'effet gyroscopique de l'hélice.

Système d'actionnement des bras

Le drone est équipé de quatre servomoteurs Dynamixel XL-320, spécialement choisis pour leur capacité à réaliser un contrôle précis de l'angle d'inclinaison en position et en vitesse. Chaque servomoteur est doté d'un régulateur PID intégré qui prend en charge cette fonction. Les paramètres de ces régulateurs ont été soigneusement ajustés par des expérimentations.

Pour garantir un comportement stable du drone, des étapes de configuration ont été entreprises. Tout d'abord, une limitation de la vitesse d'inclinaison a été introduite dans les contrôleurs. Cette saturation a été appliquée en double, sur le régulateur des Dynamixels et sur la sortie de commande. Ensuite, chaque servomoteur Dynamixel a été minutieusement calibré afin d'assurer

que l'erreur de suivi de l'angle d'inclinaison reste en deçà d'une marge de tolérance, même pour des consignes de vitesse deux fois supérieures à la valeur maximale admissible $\dot{\alpha}_{max}$. Cette approche garantit que le drone soit capable de suivre les commandes d'inclinaison de manière précise et stable. On considère que pour une marge d'erreur d'inclinaison inférieure à environ 5° , on peut garantir un comportement satisfaisant (voir figure (5.27)). L'objectif ici est d'assurer une bande passante adéquate pour les servomoteurs, qui soit largement suffisante pour répondre aux caractéristiques de contrôle souhaitées et imposées.

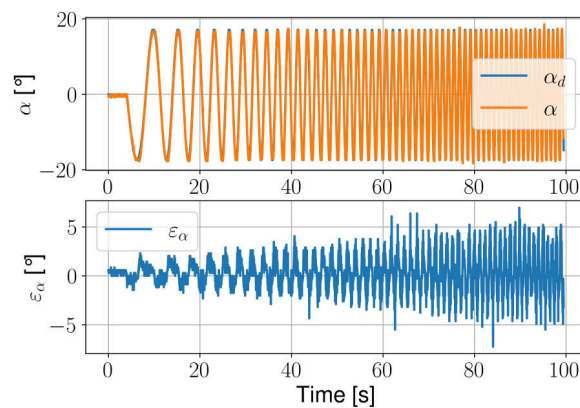


Figure 5.27: Suivi de la position du servomoteur Dynamixel pour une consigne désirée avec une fréquence variable dans le temps. $\alpha_d(t) = 17 \sin(2\pi 0.005t^2)$.

5.2.2 Scénarios et résultats expérimentaux

Tests sur banc d'essai

Pour ajuster les gains des divers contrôleurs intégrés au drone, une plateforme de test à deux degrés de liberté (ψ, θ) a été conçue et fabriquée en 3D (voir figure 5.28). Cette plateforme de test revêt une importance particulière, car elle permet non seulement une validation expérimentale initiale des architectures embarquées décrites précédemment, mais aussi l'optimisation des paramètres des contrôleurs en attitude pour obtenir le comportement souhaité.

Grâce à cette plateforme, les performances des différents contrôleurs en réponse aux variations de consignes angulaires peuvent être évaluées de manière isolée. Ces tests sur la plateforme ont contribué à affiner les réglages des contrôleurs en vue d'obtenir les performances et la réactivité attendues.

Les résultats des essais réalisés sur la plateforme de test sont illustrés dans la figure 5.29a, mettant en évidence le suivi d'une orientation souhaitée pour l'axe de tangage. Les performances en termes d'erreur de suivi sont relativement satisfaisantes, avec un écart quadratique moyen (RMSE)

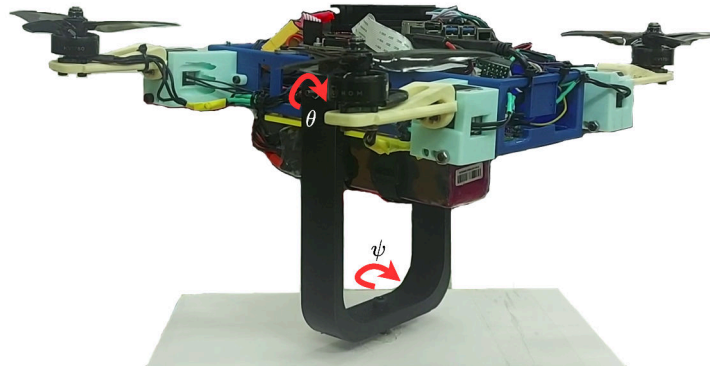


Figure 5.28: Le Xtilt sur la plateforme de test pour ajuster les paramètres des différents contrôleurs. <https://youtu.be/aDeMwgGhqEc>

de 6.63° . Cette plateforme a également joué un rôle essentiel dans la validation des gains choisis pour les servomoteurs Dynamixel, comme le démontre la figure 5.29b. En observant le suivi de la consigne de commande α_1 , nous obtenons un écart quadratique moyen (RMSE) de 1.085° .

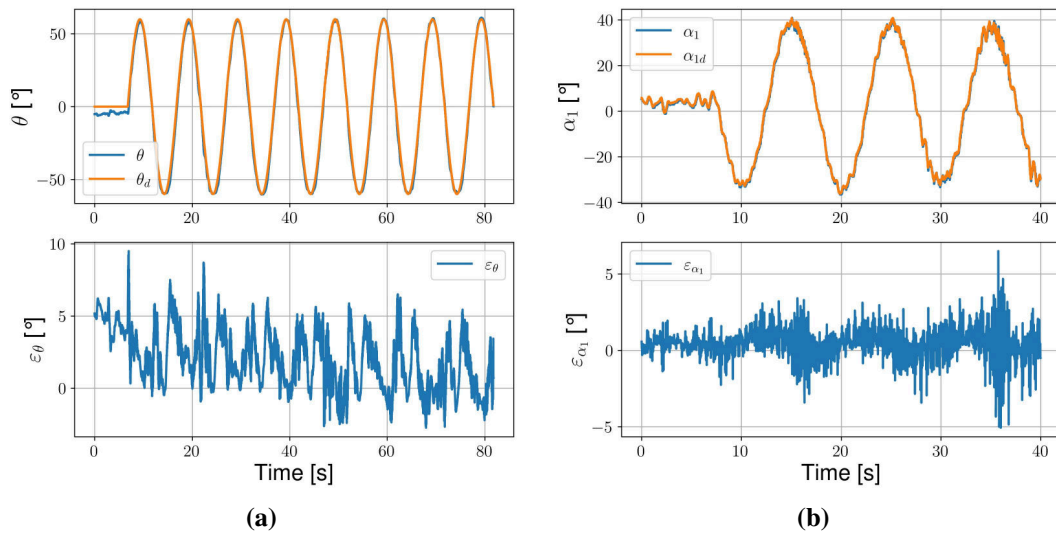


Figure 5.29: (a) Suivi d'une orientation désirée pour le tangage, la consigne désirée est $\theta_d = \frac{\pi}{3} \sin(0.2\pi t)$, $\dot{\theta}_d = 0.2 \frac{\pi^2}{3} \cos(0.2\pi t)$, (b) La consigne désirée pour le bras 1 ainsi que sa position mesurée en suivant une orientation désirée pour le tangage, la consigne désirée est $\theta_d = \frac{2\pi}{9} \sin(0.2\pi t)$, $\dot{\theta}_d = 0.2 \frac{2\pi^2}{9} \cos(0.2\pi t)$. <https://youtu.be/scisQvd7nL4>

Ces résultats, obtenus grâce à des essais sur la plateforme de test, démontrent l'efficacité des contrôleurs et des paramètres choisis, ainsi que la capacité du drone à maintenir des orientations désirées de manière précise. Les performances prometteuses sur la plateforme de test vont servir de base pour les essais en situation réelle du système de contrôle en situation de vol.

Tests en vol

- **Contrôleur PID pour la position et l'attitude** : Pour les essais en vol, le drone a été équipé de 6 marqueurs Optitrack M4, positionnés à différents endroits et orientations sur le drone, comme illustré dans la figure 5.20a. La zone d'essai était équipée de 4 caméras (*Prime^x41*) [130], couvrant un volume cubique de 2 mètres de côté.

Les résultats des essais en vol pour les consignes suivantes : $x_d = 0.5 \cos(0.2\pi t)$, $y_d = 0.5 \sin(0.2\pi t)$, $z_d = 1\text{m}$, $\psi = 0^\circ$ et $\theta = 0^\circ$, sont présentés dans la figure 5.30. Les essais en vol ont été réalisés avec les mêmes gains d'attitude que ceux utilisés lors des tests sur la plateforme 5.29a. Une observation importante est que le comportement du contrôleur d'attitude se dégrade en situation de vol réel. Ceci est dû au changement d'inertie lorsque le drone passe du banc d'essai à un vol réel. En effet, lors des essais en banc de test, l'inertie du support s'ajoute à celle du drone. De plus, même si les frottements sur les liaisons pivots sont minimales, leur effet reste non négligeable sur le comportement du drone. Les vibrations générées par les rotations des hélices ont également un impact plus marqué en vol, car elles sont moins bien absorbées.

Malgré ces différences, le drone parvient à se stabiliser en vol, ce qui est essentiel pour la suite des ajustements de paramètres. En prenant la position mesurée par l'Optitrack comme référence absolue, nous pouvons examiner les erreurs de mesure de position et d'attitude. Ces erreurs sont également illustrées dans la figure 5.30. On constate que les erreurs demeurent minimales pour θ , ϕ et x , mais une divergence se manifeste pour le lacet (ψ), cela est dû aux perturbations magnétiques générées par les moteurs brushless [131].

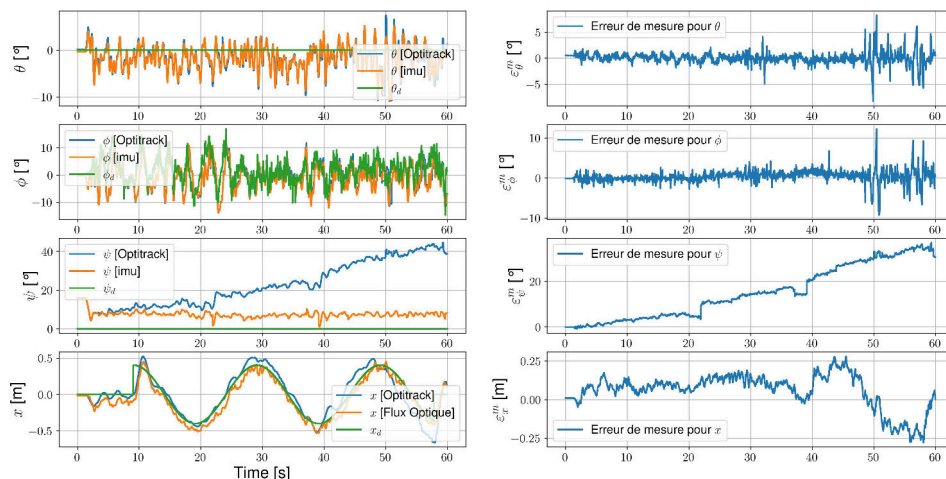


Figure 5.30: Les essais en vol ont été réalisés en utilisant la plateforme illustrée à la figure 5.20a. Les résultats de ces essais sont présentés sous forme de courbe, où sont affichées les données de position et d'orientation mesurées par le drone, ainsi que les mesures obtenues à l'aide du système Optitrack.

<https://youtu.be/qkJG5i-ZL8>

• Contrôleur intégral MPC pour la position

Pour cette partie, la plateforme représentée dans la figure 5.19b a été utilisée pour effectuer une validation expérimentale de la loi de commande intégrale MPC introduite par le problème d'optimisation (3.52). Le contrôle en position repose sur un ensemble de composants comprenant le solveur optimal Ipopt, l'estimateur ES-LKF et l'odométrie basée sur le flux optique (Optical Flow). Tous ces éléments sont exécutés dans la carte Raspberry-Pi embarquée. Leur implémentation a été réalisée en langage C++ sous le middleware ROS Noetic.

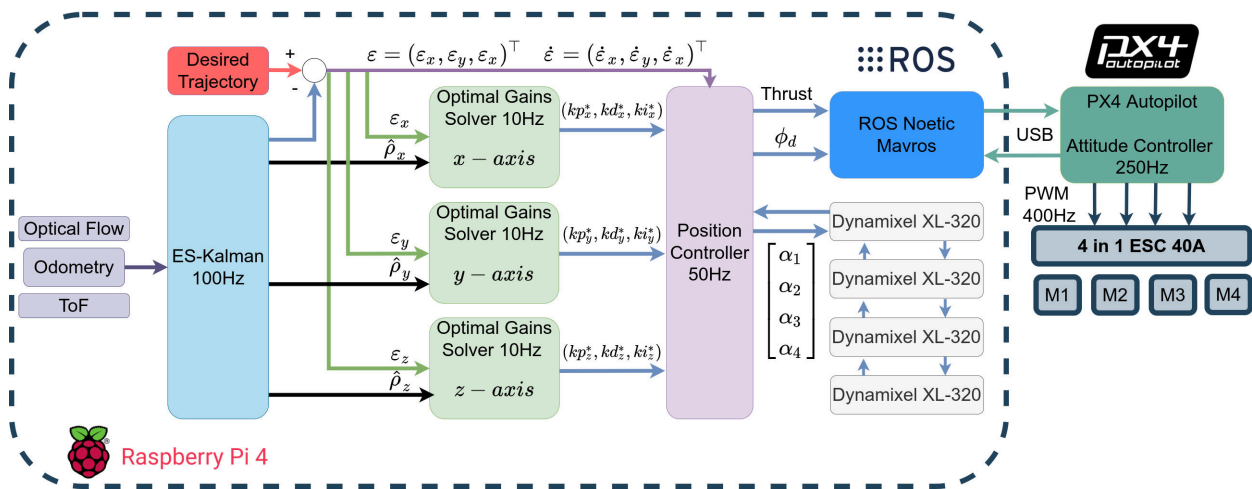


Figure 5.31: Description détaillée de l'architecture de contrôle du prototype expérimental

Pour satisfaire les exigences de temps réel avec un taux de rafraîchissement de 10 Hz, le problème d'optimisation est décomposé en trois parties distinctes, la dynamique selon l'axe x , y et z , dont la résolution est effectuée en parallèle. Cette stratégie de décomposition permet d'atteindre les performances nécessaires tout en respectant les contraintes de temps. Il est important de noter que le solveur optimal Ipopt utilisé dans cette phase expérimentale correspond au même solveur employé dans l'étude numérique préalable. La figure 5.31 ci-dessous illustre l'architecture globale de contrôle. Dans cette configuration, le drone suit une trajectoire qui comporte deux segments distincts : tout d'abord, une trajectoire circulaire (voir figure (5.32a)) puis le drone effectue un vol stationnaire non horizontal (voir figure (5.32b)). L'intégralité de la trajectoire est détaillée dans la figure (5.33a). Cependant, le robot aérien est soumis à une perturbation constante résultant de l'ajout d'une masse additionnelle pesant 58 grammes. Sans pour autant oublier la présence de termes parasites non modélisés, tels que la déformation du châssis ou le désalignement des rotors, qui peuvent influencer la commande du drone.

Les figures (5.33b) et (5.34b) mettent en évidence le comportement satisfaisant du système

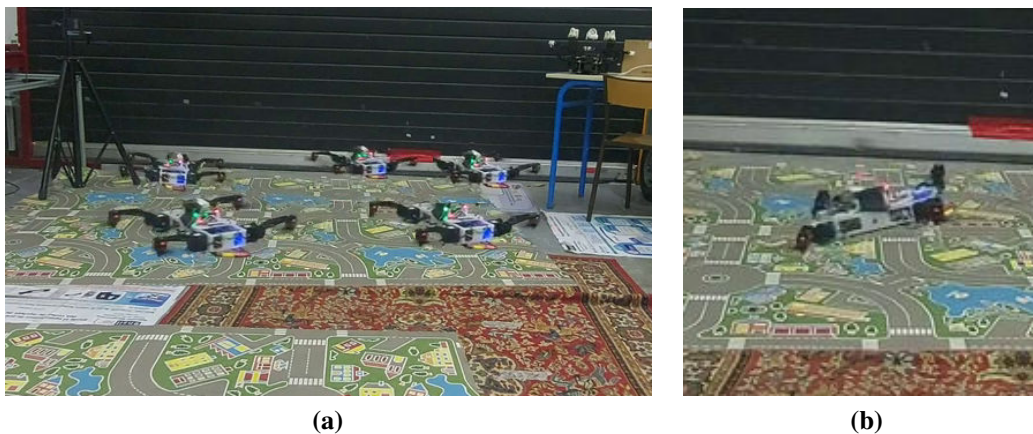


Figure 5.32: Extrait des tests en vol : (a) Suivi d'une trajectoire circulaire pour l'axe x (b) Vol stationnaire non horizontal. <https://youtu.be/SDeBwmfgua0>

en termes d'erreur de suivi de la position et de la vitesse. Ces résultats démontrent l'efficacité de la loi de commande proposée malgré les perturbations et les facteurs non modélisés. Les figures (5.35a, 5.35b) représente les gains optimaux solution du problème optimal (3.52).

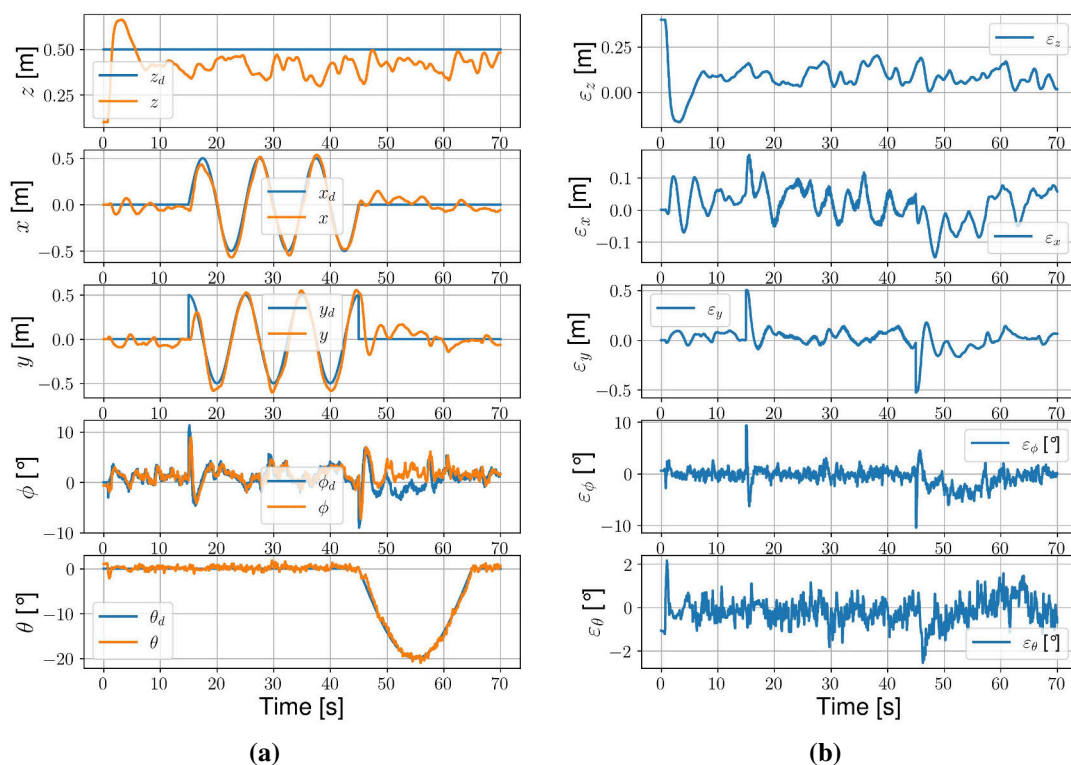


Figure 5.33: Résultats expérimentaux pour le contrôleur intégral MPC (a) Suivi en position et orientation, (b) Erreur de suivi pour la position et l'orientation.

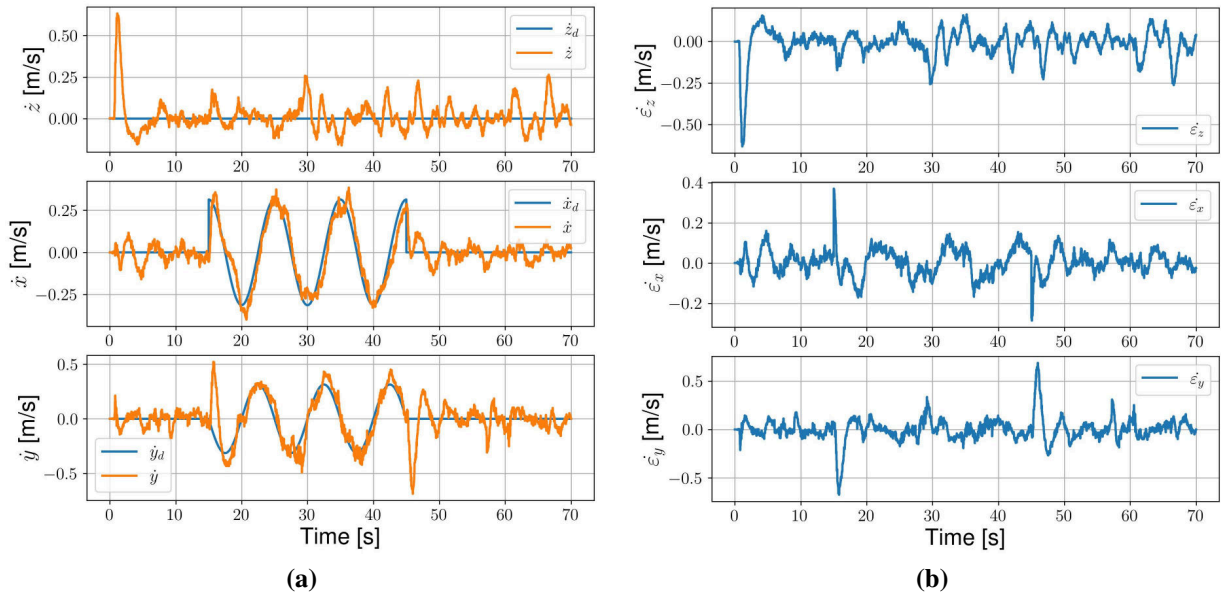


Figure 5.34: Résultats expérimentaux pour le contrôleur intégral MPC (a) Suivi en vitesse, (b) Erreur de suivi en vitesse.

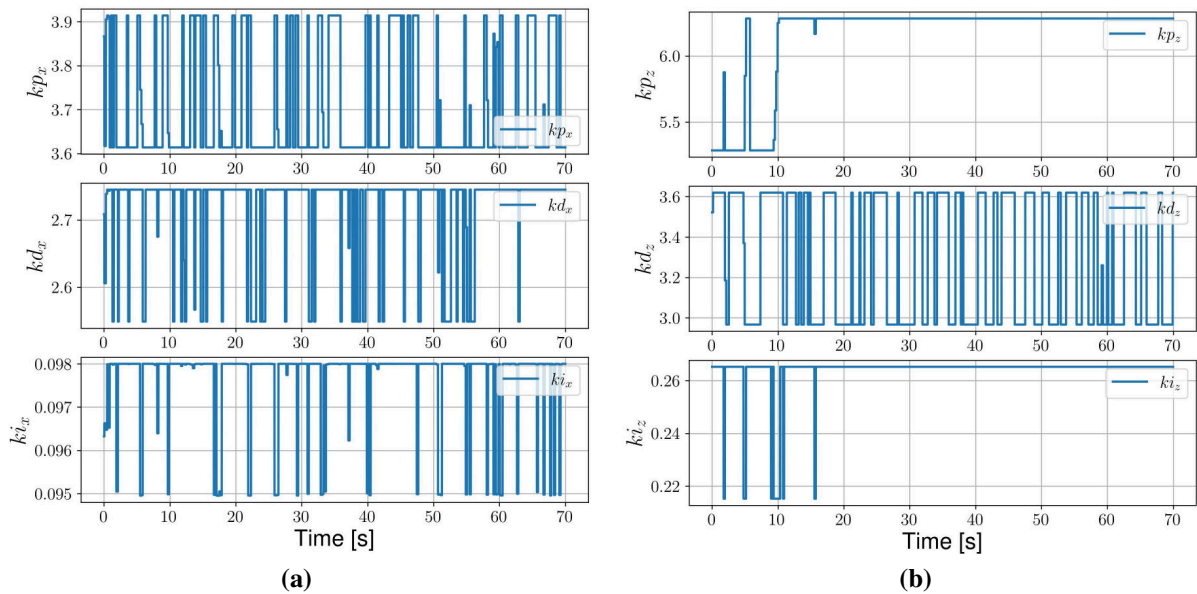


Figure 5.35: Résultats expérimentaux pour le contrôleur intégral MPC (a) Les gains pour l'axe x , (b) Les gains pour l'axe z .

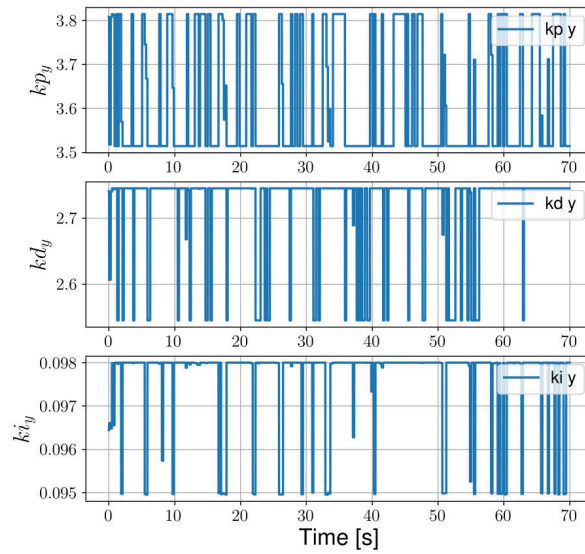


Figure 5.36: Résultats expérimentaux pour le contrôleur intégral MPC, les gains pour l'axe y

5.3 Conclusion

En résumé, cette section s'est concentrée sur la validation expérimentale des divers contrôleurs proposés, couvrant à la fois les drones à architecture Xtilt (avec bras pivotants) et les quadrirotors à bras fixes.

L'utilisation de plateformes de test spécifiquement conçues pour cela a joué un rôle crucial dans la validation expérimentale des approches proposées. Les essais en vol ont permis de démontrer la robustesse et l'efficacité des contrôleurs et des observateurs face aux variations réelles de l'environnement. Les résultats obtenus ont mis en évidence une bonne adaptation aux perturbations, validant ainsi les choix et les approches proposées.

6

Conclusion et perspectives

Sommaire

6.1	Résumé	149
6.2	Travaux futurs, quelques pistes	150
6.2.1	Génération de trajectoire	151
6.2.2	Une voilure fixe, pourquoi pas	157
6.3	Leçon tirée	157

CONCLUSION— Le chapitre de conclusion marque la fin de notre exploration approfondie sur le contrôle et la stabilisation des drones en environnements perturbés. Après avoir plongé dans les détails de la modélisation, de la conception de contrôle et des validations expérimentales, nous sommes à présent prêts à récapituler les principaux résultats et les contributions majeures de cette thèse. Cette conclusion nous offre l’opportunité de rassembler les éléments clés et de mettre en perspective les avancées que nous avons réalisées dans la quête d’une navigation autonome et stable des drones en présence de perturbations.

6.1 Résumé

Au cours de cette étude, nous avons entrepris une conception et modélisation complète de la plateforme Xtilt, en analysant chaque aspect de sa dynamique et de son comportement aérodynamique. Nous avons débuté en dérivant un modèle dynamique détaillé au moyen du formalisme d’Euler-Lagrange, englobant les efforts aérodynamiques et les couples gyroscopiques qui agissent sur le drone. En parallèle, nous avons également élaboré une modélisation CFD pour le Xtilt, élargissant ainsi notre compréhension du comportement aérodynamique du châssis et des rotors. Le Xtilt intègre un mécanisme de direction des propulseurs à servomoteurs, conférant une capacité de vectorisation variable de la poussée. Une analyse minutieuse de la distribution des forces des actionneurs a également été menée pour déduire les entrées de contrôle engendrées par la vitesse des rotors et l’angle d’inclinaison des servomoteurs. En complément, une évaluation comparative de la consommation d’énergie a été entreprise, offrant des données cruciales pour des simulations fidèles à la réalité et permettant d’évaluer l’efficacité du contrôleur ainsi que du concept de configuration. Les résultats de notre étude numérique mettent en évidence que la configuration Xtilt, avec propulseurs pivotants, affiche une consommation

d'énergie inférieure à celle du quadcoptère classique.

L'objectif central de notre travail a résidé dans la stabilisation du suivi de trajectoire des drones dans des environnements perturbés, en misant exclusivement sur un contrôleur de vol embarqué. Dans cette optique, nous avons élaboré une stratégie de contrôle robuste et optimale qui fusionne un LQR intégral avec un ES-LKF (ES-iLQG). Cette approche transcende les vulnérabilités inhérentes au LQG en termes de robustesse, tout en tenant compte des critères optimaux relatifs aux limites des actionneurs et à la convergence des états. Nous avons mené une série étendue de simulations numériques et expérimentales afin d'évaluer les performances de cette stratégie de contrôle. Les résultats obtenus, qu'ils proviennent des simulations ou des expérimentations, témoignent de l'efficacité de notre approche.

Pour compléter la modélisation en vue du contrôle et en présence de perturbations, nous avons enrichi le modèle en le segmentant en une version nominale et une perturbation intégrant des incertitudes paramétriques et exogènes. Cette démarche a préparé le terrain pour la mise en place de méthodes de contrôle optimal robuste, incluant un MPC intégral et des schémas MPSMC, tout en reposant sur l'estimation de l'ES-LKF pour maintenir à jour tant les états que les perturbations. Une étude numérique approfondie a ainsi fourni une évaluation solide de notre approche, dans un environnement virtuel étroitement fidèle à la réalité, au sein de ROS2 Humble et Ignition Gazebo Fortress. Cet environnement de simulation intègre le véritable châssis du drone (moteurs et servomoteurs) ainsi que les conditions atmosphériques. Enfin, nous avons présenté en détail les prototypes expérimentaux, couvrant à la fois l'architecture embarquée et les caractéristiques structurelles des drones. De plus, nous avons détaillé la conception et la mise en œuvre du banc d'essai expérimental, effectuant ainsi une validation préliminaire de l'architecture globale du Xtilt, suivie d'une validation en conditions de vol réel.

6.2 Travaux futurs, quelques pistes

Une fois la plateforme expérimentale prête, le travail restant doit se concentrer sur divers aspects. D'une part, il s'agit d'améliorer la génération de trajectoires afin d'optimiser différents critères en fonction des préférences de l'utilisateur. D'autre part, il faut poursuivre le développement du préhenseur, en cherchant comment intégrer un système capable à la fois de réaliser des prélèvements et d'installer différents capteurs pour effectuer des mesures, tout en respectant la limite de la charge utile que le drone peut supporter.

6.2.1 Génération de trajectoire

L'un des premiers aspects qui reste à travailler a été brièvement abordé en puisant principalement notre inspiration dans la nature. Les créatures vivantes sont une grande source d'inspiration pour la recherche, en particulier dans le domaine de la robotique moderne. En effet, selon le type des mesures à effectuer, différentes approches peuvent être utilisées. Pour réaliser des mesures de température à la surface ou une cartographie ultra-sonore, il n'est pas nécessaire d'être en contact avec l'eau ; la mesure peut être effectuée à distance. Cependant, pour des mesures de pH ou d'autres paramètres chimiques et biologiques, il est essentiel d'être en contact avec l'eau. C'est pourquoi nous avons exploré deux approches de génération de trajectoires bio-inspirées.

Trajectoire avec contact à la surface de l'eau :

La première approche s'inspire du Milan royal. Le Milan royal, en particulier, se révèle être un prédateur exceptionnel grâce à sa capacité à parcourir de longues distances en vol, ainsi qu'à sa maîtrise pour chasser des proies, même à la surface de l'eau, démontrant une agilité et une vitesse remarquables [132].

Son aptitude à descendre rapidement pour capturer une proie, puis à s'envoler rapidement par la suite, offre plusieurs avantages pour les prélèvements et mesures effectués à la surface de l'eau. Cela permet, par exemple, de déposer rapidement des sondes de mesure ou d'établir un contact direct entre le capteur et l'eau, ou encore de recueillir de petits échantillons d'eau.



Figure 6.1: La trajectoire suivie par un Milan royal lors de son approche de sa cible. source vidéo : <https://youtu.be/AYOx-iCMZhk&t=127>

Pour ce faire, nous avons utilisé une vidéo du Milan royal, filmée par The Slow Mo Guys [133], dans laquelle plusieurs Milan royaux sont en train de capturer de la nourriture au sol. Cette vidéo a été enregistrée au ralenti, ce qui nous a permis de reconstituer l'une des trajectoires du Milan royal lors de sa descente pour approcher sa proie. En utilisant le logiciel Kinovea [134], nous avons pu extraire les coordonnées des différents points de passage (voire figure 6.1). La trajectoire complète a été reconstruite en utilisant une approximation par courbe de Bézier d'ordre 2 pour 5 points de contrôle. Cette trajectoire a été testée expérimentalement en utilisant le

quadricoptère Clover 4, on retrouve dans les figures 6.2a-6.2b les courbes de suivi de la trajectoire bio-inspirée et l'erreur de suivi. On retrouve aussi dans les figures 6.2a-6.2b le suivi en vitesse et l'erreur de suivi pour cette même trajectoire <https://youtu.be/73hcMzgMsOQ>.

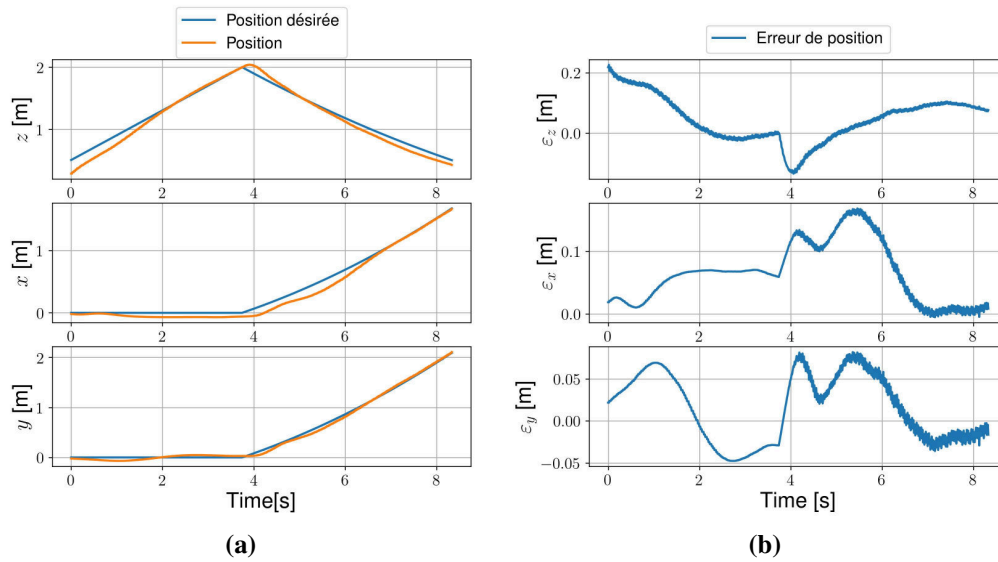


Figure 6.2: (a) Suivi d'une trajectoire bio-inspirée $x_d = (0.04t^2 + 0.4t) \cos(\psi)$, $y_d = (0.04t^2 + 0.4t) \sin(\psi)$ et $z_d = 2 - 0.4t + 0.016t^2$, (b) Erreurs de suivi de la position de la trajectoire bio-inspirée.

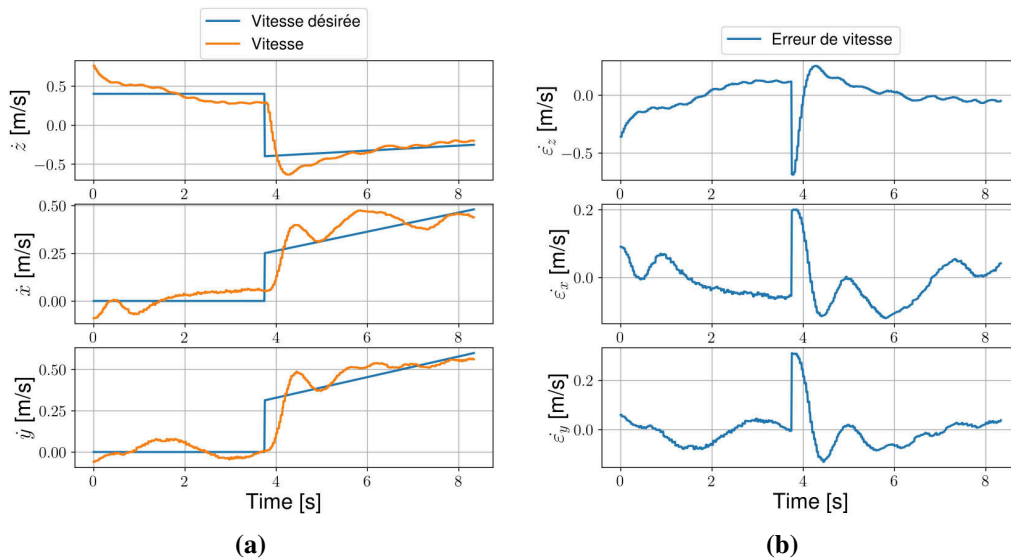


Figure 6.3: (a) Suivi de la trajectoire de vitesse bio-inspirée, la vitesse souhaitée est $\dot{x}_d = (0.4 + 0.08t) \cos(\psi)$, $\dot{y}_d = (0.4 + 0.08t) \sin(\psi)$ et $\dot{z}_d = -0.4 + 0.032t$, (b) Erreurs de suivi de la vitesse de la trajectoire bio-inspirée.

Trajectoire sans contact à la surface de l'eau :

La deuxième approche s'inspire de l'Albatros. L'autre type de prélèvement ne nécessite pas nécessairement un contact physique entre le capteur et l'eau. Ce dernier, peut être effectué à une certaine distance. Pour réaliser des cartographies de mesures, le drone doit être capable de couvrir de grandes distances tout en consommant le moins d'énergie possible. Dans la nature, un oiseau comme l'Albatros est capable de parcourir des milliers de kilomètres. Cette capacité (vol de gradient) provient de son utilisation intelligente des phénomènes naturels tels que le vent, lui permettant de récupérer de l'énergie pour rester en vol le plus longtemps possible, tout en consommant peu d'énergie en minimisant les battements de ses ailes [135].

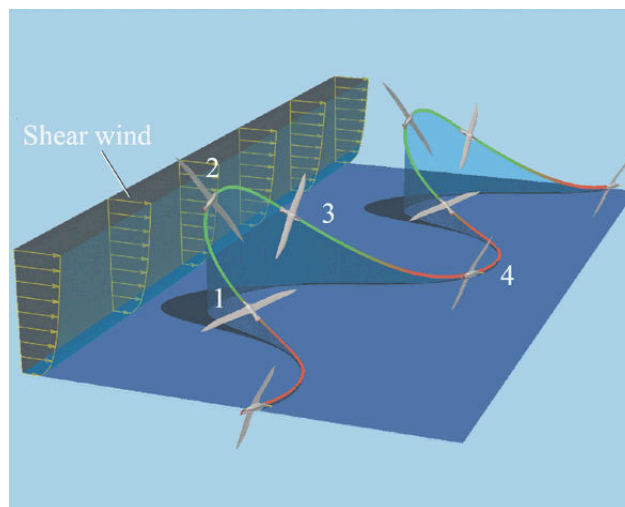


Figure 6.4: Trajectoire de vol dynamique d'un Albatros exploitant le vent comme source d'énergie. [136]

Différentes études ont été menées pour mieux comprendre comment l'Albatros réalise ces trajectoires optimales en termes de consommation énergétique [137]–[139]. Certaines de ces recherches ont recueilli des données expérimentales en équipant des Albatros de capteurs [135]. Ces études sur les techniques de vol de l'Albatros sont d'un grand intérêt, en particulier en ce qui concerne la gestion de l'énergie et l'optimisation des trajectoires de vol sur de longues distances. Il est en effet possible de résoudre ce problème en posant une question clé : quelle est la trajectoire optimale à suivre pour minimiser la perte totale d'énergie du système ? L'énergie totale du système peut être calculée comme la somme de l'énergie cinétique (Eq.2.16) et de l'énergie potentielle (Eq.2.17). Il est possible de formuler ce problème comme un problème d'optimisation, où la fonction à minimiser est la différence entre l'énergie totale initiale du système et l'énergie finale de celui-ci. La résolution se fait à temps discret, ce qui nous conduit à introduire N le nombre de variables de décision. On peut définir la différence d'énergie entre l'état initial et l'état final comme suit :

$$\Delta_E = \mathcal{T}(\mathbf{q}, \dot{\mathbf{q}}, 0) + \mathcal{V}(\mathbf{q}, 0) - \mathcal{T}(\mathbf{q}, \dot{\mathbf{q}}, N) + \mathcal{V}(\mathbf{q}, N) = E_0 - E_N \quad (6.1)$$

Comme l'objectif est de minimiser la perte d'énergie totale du système, nous pouvons définir la fonction coût comme suit :

$$\min_{0, \dots, N-1} \Delta_E \quad (6.2)$$

En considérant le modèle établi dans le **Chapitre 2** (Eq.(2.18)), on introduit ainsi le modèle discrétisé par :

$$\mathbf{X}_{k+1} = F(\mathbf{X}_k, \mathbf{U}_k) \quad (6.3)$$

\mathbf{X} est le vecteur d'état du système et \mathbf{U} est l'entrée de commande, le modèle est discrétisé par la méthode de Runge-Kutta d'ordre 4. Nous prendrons comme entrée de commande pour le système l'angle de tangage et le coefficient de portance C_l . Nous ignorerons totalement l'existence des actionneurs (moteurs). Pour modéliser le comportement de l'Albatros, nous récupérerons ses paramètres aérodynamiques et physiques dans la littérature [140]. On formalise les contraintes d'optimisation de la manière suivante :

$$\begin{aligned} C1 : & \quad \mathbf{X}_{k+1} = F(\mathbf{X}_k, \mathbf{U}_k) \\ C2 : & \quad E_0 - E_N \leq \Delta_E \\ C3 : & \quad z_0 = 3m \\ C4 : & \quad y_0 = 0m \\ C5 : & \quad x_0 = 0m \\ C6 : & \quad \phi_0 = 0rad \quad -\pi/2 < \phi_k < \pi/2 \quad \forall k \\ C7 : & \quad \psi_0 = 0rad \quad -\pi/2 < \psi_k < \pi/2 \quad \forall k \\ C8 : & \quad \theta_0 = \iota_0 = 0rad \quad -\pi/2 < \theta_k - \iota_k < \pi/2 \quad \forall k \\ C9 : & \quad z_N = z_0 \\ C10 : & \quad x_N = x_0 \\ C11 : & \quad y_N = 45m \\ C12 : & \quad T = 5.5s \\ C13 : & \quad V_0 = 15m/s \end{aligned} \quad (6.4)$$

La première condition, C1, représente la contrainte dynamique de notre système, tandis que la seconde condition, C2, vise à garantir que la différence entre l'énergie initiale et finale soit minimale, puisque l'objectif de la minimisation est de réduire Δ_E . Les autres conditions représentent les conditions aux limites que nous souhaitons imposer au système. Par exemple, V_0 représente la vitesse de l'air du système à l'instant initial, et y_N représente la contrainte

sur la position finale selon l'axe y . Nous imposons également des limites temporelles et d'orientation pour le fonctionnement du système.

Il convient de noter que le système est soumis à un vent le long de l'axe des x , et le modèle utilisé pour la vitesse moyenne du vent en fonction de la hauteur z est le suivant :

$$V_w = 6.5 \left(\frac{\log\left(\frac{z}{0.03}\right)}{\log\left(\frac{10}{0.03}\right)} \right) \quad (6.5)$$

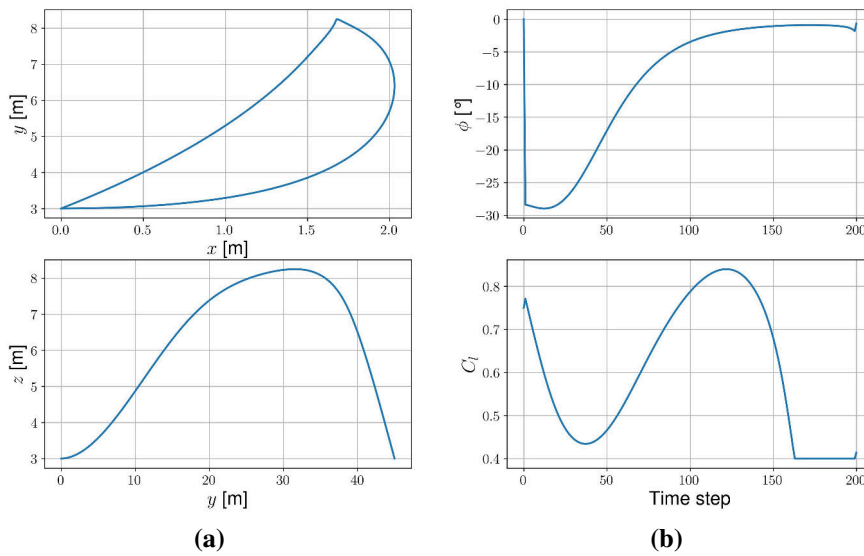


Figure 6.5: Résultats d'optimisation énergétique de la trajectoire (a) [haut] La position dans le plan $x - y$, [bas] La position sur le plan $z - y$ (b) Les entrées de commande, [haut] l'angle de tangage, [bas] Le coefficient de portance

En s'inspirant de [138] on utilisera un polynôme quadratique pour modéliser le coefficient de traînée C_d :

$$C_d = 0.0173 - 0.00224C_l + 0.0629C_l^2 - 0.0578C_l^3 + 0.0314C_l^4 \quad (6.6)$$

Pour résoudre ce problème d'optimisation, nous avons utilisé le même solveur que dans le **Chapitre 3** et **Chapitre 4** (IPOPT). La simulation a été réalisée en utilisant Python 3 avec CasADi. Les solutions optimales obtenues dans les figures (6.5) et (6.6). Étant donné que le modèle est relativement complexe, la résolution du problème d'optimisation a pris 8.9 secondes. Il est important de noter que cette durée de résolution reste relativement longue pour des applications en temps réel. Dans le cas de l'application au drone, la connaissance des paramètres aérodynamiques du drone ainsi que de la vitesse air, doit être disponible, que ce soit par observation ou mesure.

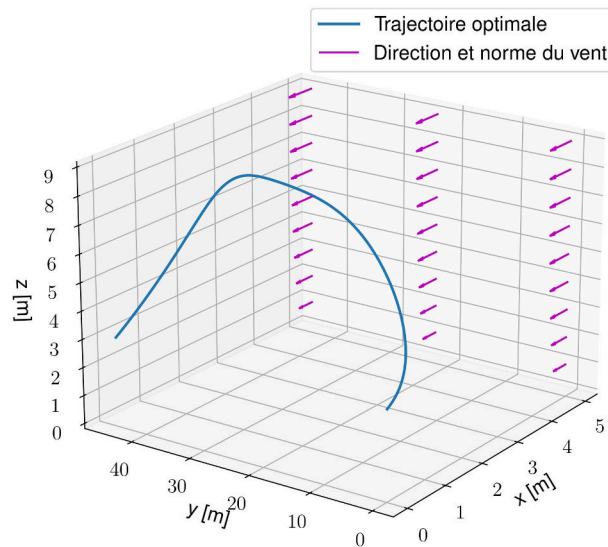


Figure 6.6: Une période de la trajectoire optimale.

Autres scénarios :

Vol stationnaire inspiré de l’Océanite tempête : Dans la nature, il existe également des sources d’inspiration pour réaliser des vols stationnaires en minimisant la consommation d’énergie. Un troisième scénario ou type de prélèvement n’a pas été abordé précédemment, il implique la réalisation de mesures à différentes profondeurs, nécessitant ainsi que le drone se maintienne en position stationnaire pendant la collecte. Dans la nature, il existe un exemple d’oiseau capable de rester en position stationnaire en utilisant à la fois la traînée générée par le vent et le courant d’eau, comme illustré dans la figure suivante.

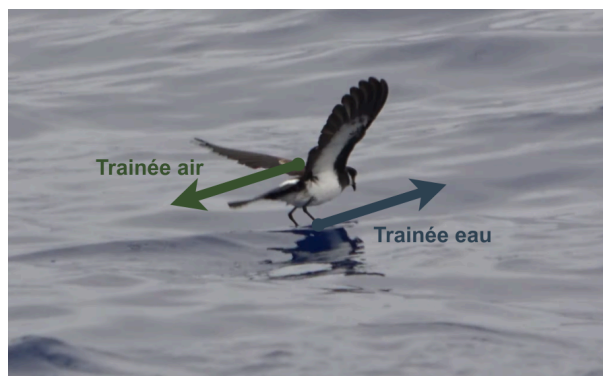


Figure 6.7: L’océanite tempête en pleine chasse, source : <https://youtu.be/X5pG1OQ0VfM>

Les forces de traînée aérodynamique poussent l’oiseau vers l’arrière, tandis que ses pattes, immergées dans l’eau, subissent un écoulement relatif en sens inverse du vent, créant ainsi une

traînée hydrodynamique qui agit dans le sens opposé sur les pattes de l'oiseau. Lorsque ces forces de traînée aérienne et aquatique s'équilibrent, l'oiseau parvient à maintenir une position stationnaire à une hauteur qui lui permet de chasser à la surface de l'eau [140], voir la figure (6.7).

6.2.2 Une voilure fixe, pourquoi pas

Afin de minimiser la consommation énergétique du Xtilt, compte tenu de ce qui a été discuté précédemment, la meilleure solution semble être l'ajout d'ailes au drone. Le vent peut être considéré comme une source d'énergie exploitable par le drone grâce à une optimisation appropriée. De plus, la conception de ces ailes peut également être optimisée pour garantir une portance élevée tout en maintenant une vitesse de l'air minimale. En effet, il existe dans la littérature plusieurs travaux de recherche qui ont abordé cette problématique [141].



Figure 6.8: XWing une version améliorée du Xtilt.

6.3 Leçon tirée

Une leçon essentielle à tirer de ces années de thèse est la distinction entre le monde de la théorie et le monde de la pratique. Même si une étude est menée avec la plus grande rigueur dans la conceptualisation théorique, il persiste toujours des différences lorsqu'on passe à l'application réelle. Cependant, il est crucial de comprendre que ces deux mondes sont complémentaires. L'expérience pratique permet non seulement de valider la théorie, mais elle peut également servir de boucle de rétroaction pour affiner ou corriger celle-ci. Cette approche a été centrale dans cette thèse, où j'ai suivi un processus itératif de modélisation, de simulation, d'expérimentation, puis de retour à la modélisation, et ainsi de suite.



Annexes

Sommaire

A.1	Quelques améliorations pour le système de propulsion	159
A.2	Piste d’atterrissage pour drone sur le robot Cinteo	160

A.1 Quelques améliorations pour le système de propulsion

Dans la littérature scientifique, l'approche classique pour interagir avec les systèmes de propulsion (hélice et moteur) consiste à identifier les différents paramètres aérodynamiques de l'hélice, tels que la poussée et la traînée en fonction de la vitesse de rotation de l'hélice. De plus, l'identification des moteurs est réalisée pour connaître la vitesse de rotation des moteurs en fonction de la tension d'alimentation. En plaçant un capteur de tension et un capteur de vitesse de rotation des moteurs, il est possible d'asservir le système de propulsion. Cependant, ces identifications restent valables uniquement pour des points de fonctionnement spécifiques. Le drone peut être exposé à des perturbations externes telles que des rafales de vent, qui peuvent altérer ces estimations, sans oublier l'interaction entre les différents flux d'air générés par les hélices.

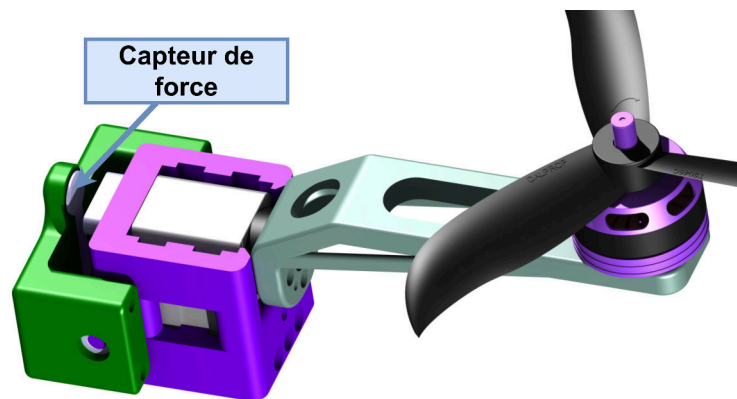


Figure A.1: Le schéma de contrôle pour l'asservissement de la force générée par les hélices

En s'inspirant de la plateforme expérimentale réalisée au cours de cette thèse pour identifier le système de propulsion, il est possible de l'embarquer sur le drone. Cependant, quelques modifications sont nécessaires. Le schéma suivant (figure A.1) montre un bras équipé d'un capteur de force. Ce capteur de force est en réalité une simple résistance variable qui ne nécessite pas beaucoup de ressources au micro-contrôleur pour être lue. Cette configuration permet de mesurer la force générée selon l'axe e_3 dans la base \mathcal{B} . Étant donné que les orientations des bras sont connues, il est possible d'observer la poussée totale générée par le système de propulsion. Ainsi, il est possible d'asservir la force générée par l'hélice, où la consigne est la force désirée par le contrôleur (voir figure A.2). Cela permettra également d'enrichir l'estimation du filtre de Kalman. En effet, au lieu de donner comme entrée de commande la sortie du contrôleur, qui peut différer de ce que le système subit, il serait plus judicieux de remplacer la sortie de commande par la mesure de la commande réellement réalisée directement sur l'actionneur.

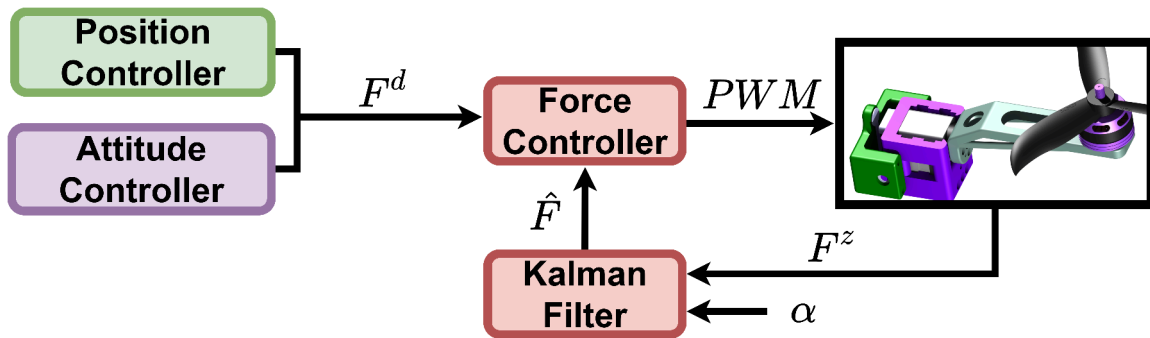


Figure A.2: Schéma de contrôle pour l'asservissement de la force générée par les hélices, α est l'angle d'inclinaison, qui est connu, F^z représente la force mesurée portée par l'axe e_3 du repère du drone. Cela signifie que F^z est la composante de la force dans la direction verticale. F^d est la force désirée par le contrôleur, c'est-à-dire la valeur de force que le contrôleur souhaite atteindre pour réaliser une tâche donnée ou pour maintenir le drone dans une certaine configuration de vol.

A.2 Piste d'atterrissage pour drone sur le robot Cinteo

Dans cette section, nous aborderons certains travaux réalisés pendant la thèse qui sont liés au sujet de cette dernière.

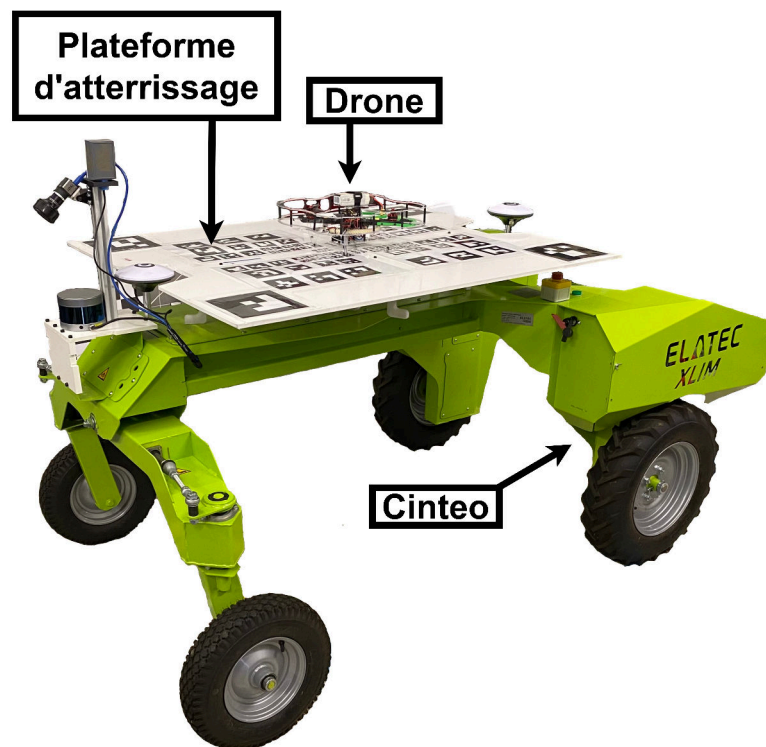


Figure A.3: La plateforme expérimentale Cinteo avec la piste d'atterrissage et le drone (Clover 4).

Le robot Cinteo (voir figure A.3) a été développé en collaboration avec Elatec et est dédié à la

détection des cultures et des adventices, tout en naviguant de manière autonome dans un champ agricole [142]. Le lien avec les travaux de ma thèse s'est établi en raison du besoin du robot terrestre Cinto d'obtenir une perception augmentée de son environnement, similaire à une vue aérienne. Il convient de noter que le travail entrepris s'est principalement concentré sur la phase finale, à savoir l'approche finale du drone vers la piste d'atterrissage. L'approche utilisée repose sur la vision. Cependant, pour des missions où le drone effectue des vols à une certaine distance du robot terrestre, au point de perdre complètement de vue ce dernier, d'autres moyens de localisation et de communication doivent être utilisés. Nonobstant cela, afin d'accroître la fiabilité du positionnement du drone, une solution a été proposée pour compenser la perte de vue de la plateforme.

Afin de permettre la localisation du drone par rapport au robot mobile, une piste d'atterrissage a été équipée de codes Aruco [143], en utilisant la bibliothèque OpenCV [144]. Cette approche permet d'obtenir la position du drone par rapport à ces marqueurs en supposant ces derniers dans le champ de vision de la caméra du drone. Étant donné que la caméra du drone a un champ de vision limité, il est possible que certaines parties des marqueurs soient occultées lorsque le drone s'approche de la piste d'atterrissage, notamment lors de l'atterrissage. Pour éviter cela et garantir qu'au moins un certain nombre minimum de marqueurs restent visibles dans le champ de vision de la caméra tout au long du processus d'atterrissage, différents marqueurs de tailles variées ont été utilisés. À proximité de la cible d'atterrissage, les marqueurs sont de petite taille, tandis que leur taille augmente à mesure que l'on s'éloigne du point central. La figure suivante A.4 illustre l'agencement des marqueurs ainsi que leurs positions par rapport au centre choisi.

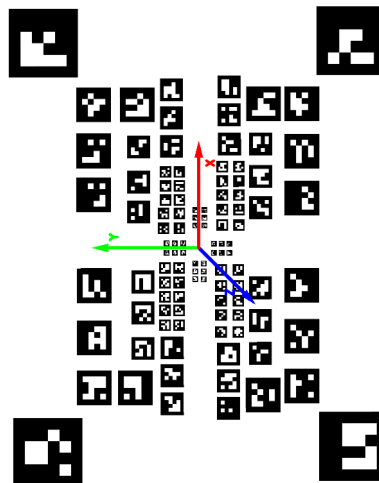


Figure A.4: Carte de marqueurs Aruco pour la piste d'atterrissage.

En utilisant la bibliothèque Aruco d'OpenCV, il est possible de créer un tableau Aruco qui correspond à un regroupement de marqueurs Aruco et qui fonctionne comme un marqueur unique en fournissant une position unique par rapport à la caméra [145]. De cette manière, il est possible de créer une carte d'Aruco, et la position mesurée sera relative au point de référence choisi. Il convient de noter qu'avec cette approche, il n'est pas nécessaire que tous les marqueurs Aruco soient dans le champ de vision de la caméra, ce qui améliore la robustesse de la détection. Il faut prendre en compte que l'atterrissage doit être effectué à l'extérieur, en présence de soleil et d'autres perturbations lumineuses susceptibles d'altérer la détection. En augmentant le nombre de marqueurs, on réduit cette probabilité. Cependant, il est important de se rappeler que la caméra embarquée sur le drone a une limite en termes de fréquence d'acquisition (images par secondes), ce qui signifie que pour des mouvements rapides, les images capturées peuvent être floues. Cela représente un problème sérieux pour la détection. En effet, la caméra embarquée sur le drone permet également de calculer le flux optique. L'idée pour améliorer la robustesse de l'estimation de la position consiste à combiner les deux approches en utilisant un filtre de Kalman linéaire basé sur le modèle du drone et la mesure de la vitesse. Il est également possible de combiner la mesure d'accélération fournie par l'accéléromètre, mais dans ce cas, nous nous contenterons de la vitesse uniquement. Le schéma A.5 suivant résume l'approche utilisée.

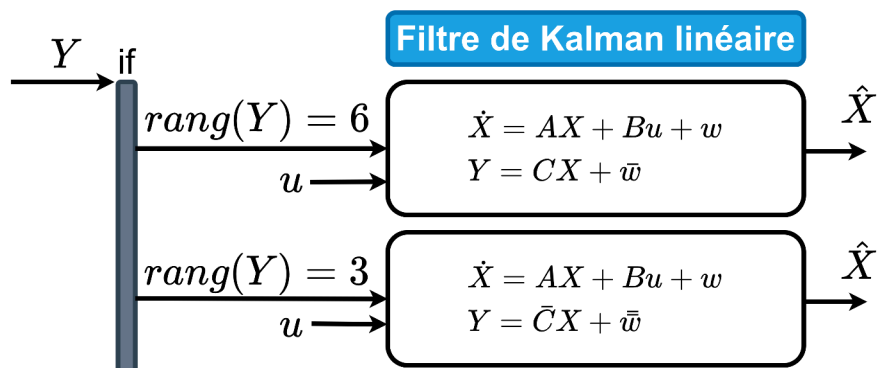


Figure A.5: Lorsque la mesure de position n'est pas disponible, nous utilisons le second modèle d'observation. La principale différence avec le premier modèle réside dans le rang de la matrice d'observation. En cas de perte de la mesure de position, nous utilisons comme condition initiale la dernière mesure obtenue.

Plusieurs essais en simulation ont été effectués pour valider préliminairement la solution proposée. La simulation a été réalisée dans l'environnement ROS Gazebo (voir figure A.6), intégrant à la fois le drone et un jumeau numérique du robot Cinto. Un contrôleur MPC a été utilisé pour le suivi de trajectoire du robot Cinto, tandis que le drone était équipé d'un contrôleur ES-iLQG, comme détaillé dans la thèse.

Un simple test de suivi a été effectué, où le drone avait pour consigne de maintenir une position $x = y = 0$ et $z = 0.5m$ dans le référentiel de la carte des marqueurs, tandis que le robot Cinteo suivait une trajectoire en point consigne. On retrouve dans la vidéo suivante le résultat obtenu en terme de suivi de trajectoire <https://youtu.be/LcsNbCaNavg>.

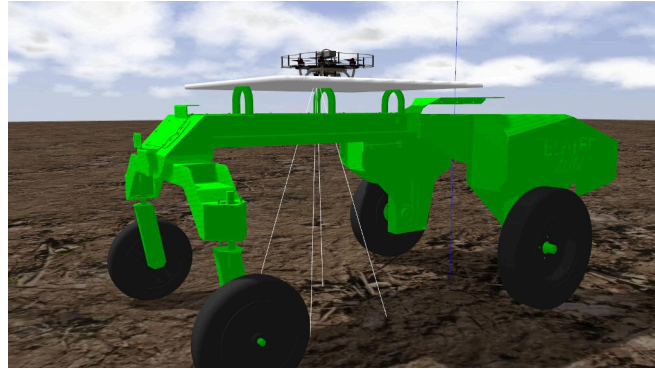


Figure A.6: Simulation sous Gazebo de Cinteo et Clover 4.

Pour la partie expérimentale, nous avons utilisé le quadricoptère Coex Clover, qui a déjà été introduit dans la partie expérimentale de la thèse **Chapitre 5**. Les traitements d'image ont été effectués à une résolution de 320p à une fréquence de 40 images par seconde. La validation expérimentale s'est déroulée en deux étapes. Dans un premier temps, nous avons validé l'estimation de la position, puis dans un second temps, nous avons réalisé un test de suivi en position d'un robot mobile. Dans ce cas, le robot mobile était un Agilex Scout Mini.

- **Estimation de la pose du drone relativement à la piste d'atterrissage** Dans cette première partie, le drone doit suivre une trajectoire hélicoïdale (déjà décrite dans le **Chapitre 5**). Le suivi de la trajectoire s'effectue en utilisant la même odométrie que celle du chapitre expérimental, c'est-à-dire que la position mesurée par l'approche proposée basée sur les marqueurs Aruco n'intervient pas au niveau de la commande. L'objectif ici est de valider l'exactitude de cette mesure en la comparant à un système de positionnement fonctionnel. Dans les figures suivantes, la figure A.7a montre les positions mesurées (marqueurs Aruco) et filtrées par un filtre de kalman, tandis que la figure A.7b présente les erreurs de mesure par rapport à la mesure du flux optique filtrée par un filtre de Kalman. Voici quelques observations :
 - On remarque que lorsque le drone s'éloigne de la piste d'atterrissage tout en gagnant de l'altitude, la mesure devient bruitée. Cela est normal car la résolution d'image utilisée est limitée à 320p en raison de la puissance de calcul réduite de la Raspberry Pi. Une mauvaise résolution d'image affecte à la fois la détection et la précision de la mesure.
 - On observe également une certaine dérive avec le temps entre les mesures du flux

optique et par les marqueurs Aruco. En effet, la figure A.8b montre que lorsque le drone commence à atterrir, il n'est pas centré, comme le montre la position estimée par le flux optique. Il existe un décalage entre les deux mesures. Comme la position des marqueurs Aruco sur la plateforme d'atterrissage est parfaitement connue, la mesure la plus fiable en termes de précision de localisation reste celle des Aruco. Le flux optique a tendance à diverger dans le temps en raison de petites erreurs de mesure sur la vitesse, qui, bien que très fines, s'accroissent lors de l'intégration pour obtenir la position.

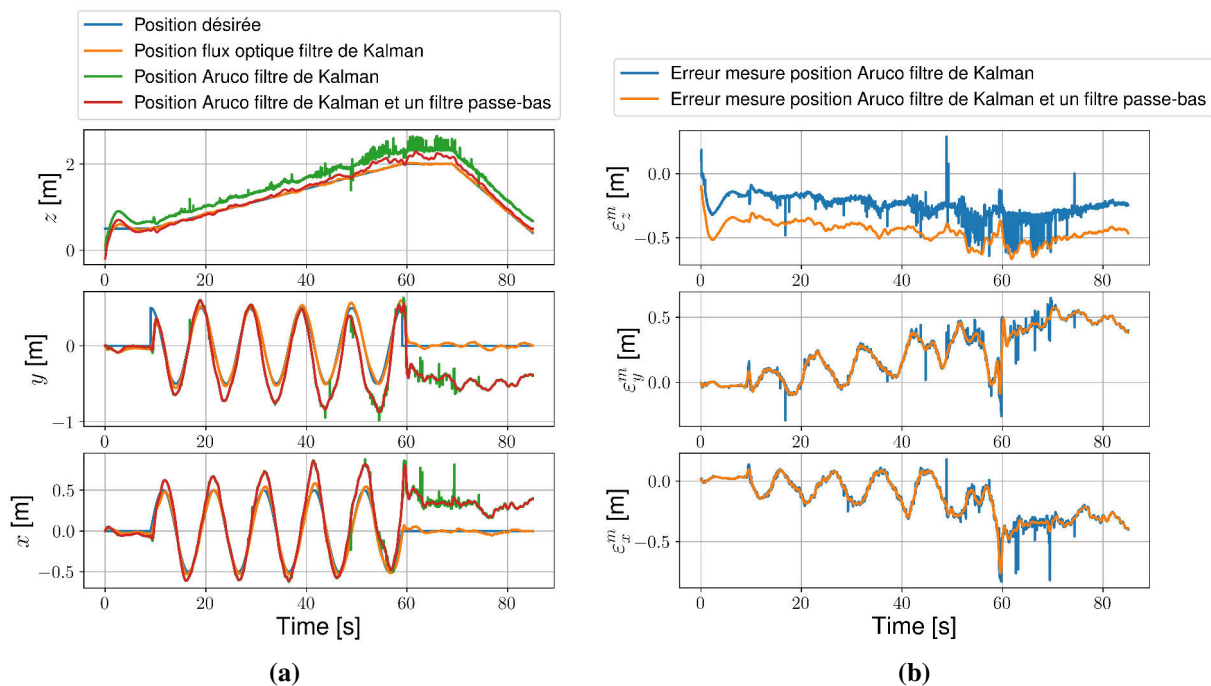


Figure A.7: Les résultats obtenus pour l'estimation de position en suivi d'une trajectoire hélicoïdale (a) On retrouve ici les positions souhaitées ainsi que les différentes mesures filtrées, (b) Les erreurs de mesure sont calculées en prenant comme référence la mesure réalisée par le flux optique et les marqueurs Aruco. https://youtu.be/VQ8nfh_NVd4

- **Suivi de la plateforme mobile** Afin de valider le suivi de cible mobile, la piste d'atterrissage a été montée sur le robot Agilex Scout Mini. Le drone est contrôlé pour maintenir un vol stationnaire, la position selon x et y à zéro, la seule consigne que l'opérateur peut modifier est la hauteur. Cela a permis de valider la phase finale d'atterrissage, même lorsque le robot terrestre est en mouvement. Le drone doit être capable de suivre le robot en mouvement. Une vidéo illustrant cela : <https://youtu.be/ZhNtfMeZAEo>. À l'instant $t = 16s$ dans la vidéo, le drone perd la détection des marqueurs Aruco, cependant, cela n'affecte pas son bon fonctionnement. Une fois que les marqueurs Aruco sont à nouveau détectés, le drone

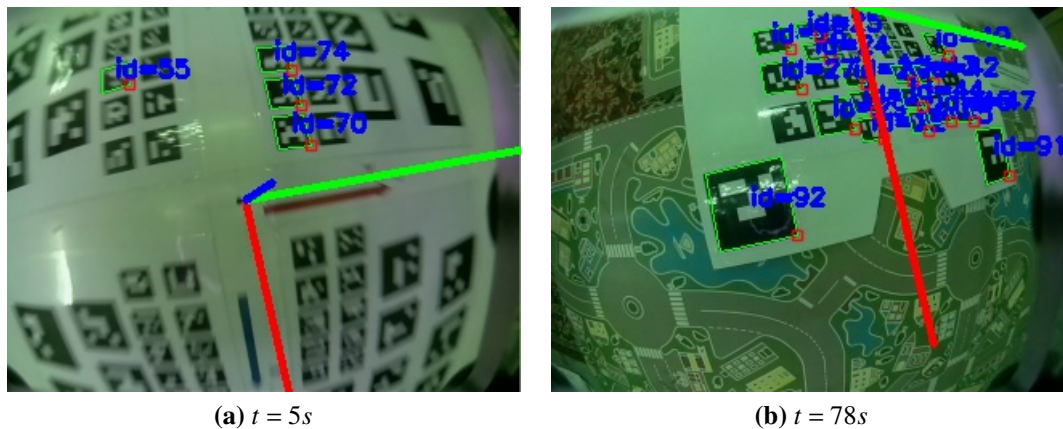


Figure A.8: Ces images ont été prises par le drone lors de suivi d'une trajectoire hélicoïdale à différents instants. (a) Pour $t = 5s$ lors de la phase de décollage (b) Pour $t = 78s$ lors de la phase d'atterrissage. La ligne rouge correspond à l'axe des x , la ligne verte à l'axe des y et la ligne bleue à l'axe des z du drone.

rejoint sa consigne.

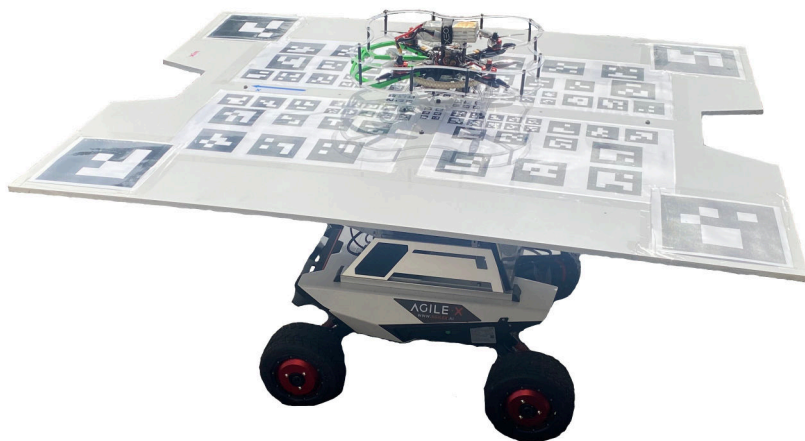


Figure A.9: Le robot Agilex Scout Mini portant la piste d'atterrissage et le drone.

Pour finir, quelques tests de vol ont été réalisés à l'extérieur dans des conditions réelles, y compris des variations de luminosité. La figure (A.10) illustre quelques images prises par le drone lors de ces vols télécommandés en mode manuel.

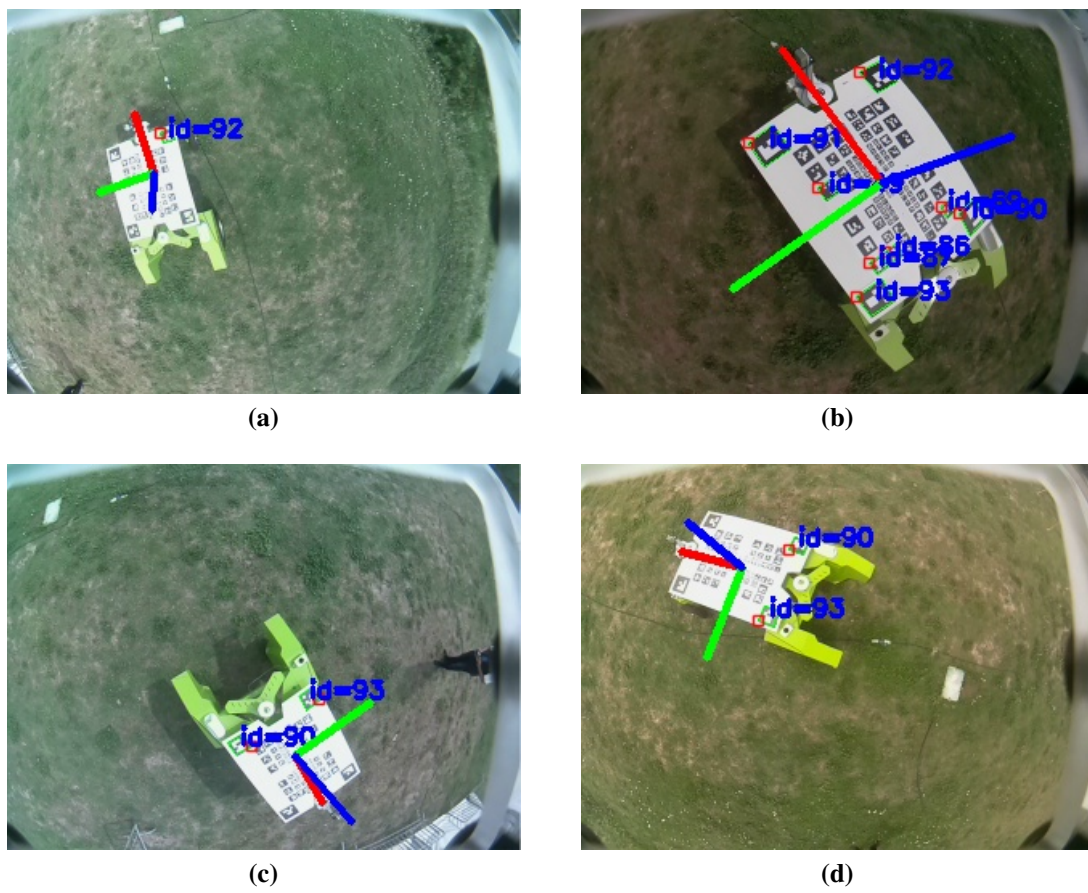


Figure A.10: Images prises par le drone en survolant le robot agricole Cinteo et la piste d’atterrissage à l’extérieur.

B

Bibliographie

Sommaire

Références	168
Liste des travaux	180

Références

- [1] K. Valavanis, G. Vachtsevanos, and P. Antsaklis, “Technology and autonomous mechanisms in the mediterranean: from ancient greece to byzantium,” in *2007 European Control Conference (ECC)*, IEEE, 2007, pp. 263–270.
- [2] D. R. Hill, “Mechanical engineering in the medieval near east,” *Scientific American*, vol. 264, no. 5, pp. 100–105, 1991. [Online]. Available: <http://www.jstor.org/stable/24936907> (visited on 08/28/2023).
- [3] O. B. Tibbetts, “Donald r. hill (tr.): the book of knowledge of ingenious mechanical devices (kitab fi ma‘rifat al-ḥiyal alhandasiyya), by ibn al-razzaz al-jazari. xxv, 285 pp. dordrecht, boston: d. reidel publishing co., [c1974]. guilders 240.,” *Bulletin of the School of Oriental and African Studies*, vol. 38, no. 1, pp. 151–153, 1975.
- [4] [Online]. Available: <https://www.theguardian.com/world/2008/may/16/italy.japan>.
- [5] [Online]. Available: <https://www.heli-archive.ch/en/helicopters/helicopters>.
- [6] [Online]. Available: <https://www.chateauversailles.fr/decouvrir/histoire/grandes-dates/premier-vol-montgolfiere>.
- [7] G. Zhang, B. Shang, Y. Chen, and H. Moyes, “Smartcavedrone: 3d cave mapping using uavs as robotic co-archaeologists,” in *2017 International Conference on Unmanned Aircraft Systems (ICUAS)*, 2017, pp. 1052–1057.
- [8] A. Singletary, T. Gurriet, P. Nilsson, and A. D. Ames, “Safety-critical rapid aerial exploration of unknown environments,” in *2020 IEEE International Conference on Robotics and Automation (ICRA)*, 2020, pp. 10 270–10 276.
- [9] [Online]. Available: https://www.esa.int/Science_Exploration/Human_and_Robotic_Exploration/CAVES_and_Pangaea/Exploring_underground_with_a_colliding_drone.
- [10] [Online]. Available: <https://news.cnrs.fr/articles/new-tools-to-help-predict-forest-fires>.

- [11] [Online]. Available: <https://www.theguardian.com/world/2013/aug/27/drones-polar-melting-ice-antarctica-greenland>.
- [12] P. Harder, M. Schirmer, J. Pomeroy, and W. Helgason, “Accuracy of snow depth estimation in mountain and prairie environments by an unmanned aerial vehicle,” *The Cryosphere*, vol. 10, no. 6, pp. 2559–2571, 2016.
- [13] I. Eischeid, E. M. Soininen, J. J. Assmann, *et al.*, “Disturbance mapping in arctic tundra improved by a planning workflow for drone studies: advancing tools for future ecosystem monitoring,” *Remote Sensing*, vol. 13, no. 21, 2021. [Online]. Available: <https://www.mdpi.com/2072-4292/13/21/4466>.
- [14] J.-P. Ore, S. Elbaum, A. Burgin, and C. Detweiler, “Autonomous aerial water sampling,” *Journal of Field Robotics*, vol. 32, no. 8, pp. 1095–1113, 2015.
- [15] C. Koparan, A. B. Koc, C. V. Privette, and C. B. Sawyer, “In situ water quality measurements using an unmanned aerial vehicle (uav) system,” *Water*, vol. 10, no. 3, 2018. [Online]. Available: <https://www.mdpi.com/2073-4441/10/3/264>.
- [16] K. Song, A. Brewer, S. Ahmadian, A. Shankar, C. Detweiler, and A. J. Burgin, “Using unmanned aerial vehicles to sample aquatic ecosystems,” *Limnology and Oceanography: Methods*, vol. 15, no. 12, pp. 1021–1030, 2017.
- [17] S. Yuan, Y. Li, F. Bao, *et al.*, “Marine environmental monitoring with unmanned vehicle platforms: present applications and future prospects,” *Science of The Total Environment*, vol. 858, p. 159741, 2023.
- [18] J. Paulos, B. Caraher, and M. Yim, “Emulating a fully actuated aerial vehicle using two actuators,” in *2018 IEEE International Conference on Robotics and Automation (ICRA)*, IEEE, 2018, pp. 7011–7016.
- [19] M. Ryll, G. Muscio, F. Pierri, *et al.*, “6d physical interaction with a fully actuated aerial robot,” in *2017 IEEE International Conference on Robotics and Automation (ICRA)*, IEEE, 2017, pp. 5190–5195.
- [20] P. E. I. Pounds, D. R. Bersak, and A. M. Dollar, “Grasping from the air: Hovering capture and load stability,” in *2011 IEEE International Conference on Robotics and Automation*, Shanghai, China: IEEE, May 2011, pp. 2491–2498. (visited on 04/20/2022).
- [21] R. Mahony, V. Kumar, and P. Corke, “Multirotor Aerial Vehicles: Modeling, Estimation, and Control of Quadrotor,” *IEEE Robotics Automation Magazine*, vol. 19, no. 3, 2012, Conference Name: IEEE Robotics Automation Magazine.
- [22] S. Bouabdallah, M. Becker, and R. Siegwart, “Autonomous miniature flying robots: coming soon! - Research, Development, and Results,” *Robotics & Automation Magazine, IEEE*, vol. 14, pp. 88–98, Oct. 2007.

- [23] A. Offermann, P. Castillo, and J. De Miras, “Nonlinear model and control validation of a tilting quadcopter,” in *2020 28th Mediterranean Conference on Control and Automation (MED)*, ISSN: 2473-3504, Sep. 2020, pp. 50–55.
- [24] S. Sridhar, G. Gupta, R. Kumar, M. Kumar, and K. Cohen, “Tilt-Rotor Quadcopter Xplored: Hardware based Dynamics, Smart Sliding Mode Controller, Attitude Hold amp; Wind Disturbance Scenarios,” in *2019 American Control Conference (ACC)*, ISSN: 2378-5861, Jul. 2019, pp. 2005–2010.
- [25] M. Ryll, H. H. Bühlhoff, and P. R. Giordano, “A Novel Overactuated Quadrotor UAV: Modeling, Control and Experimental Validation,” en, *IEEE Transactions on Control Systems Technology*, vol. 23, no. 2, p. 510, Feb. 2015. (visited on 04/20/2022).
- [26] A. Sanchez, J. Escareño, O. Garcia, and R. Lozano, “Autonomous Hovering of a Noncyclic Tiltrotor UAV: Modeling, Control and Implementation,” en, *IFAC Proceedings Volumes*, 17th IFAC World Congress, vol. 41, no. 2, pp. 803–808, Jan. 2008. (visited on 04/20/2022).
- [27] M. Allenspach, K. Bodie, M. Brunner, *et al.*, “Design and optimal control of a tiltrotor micro-aerial vehicle for efficient omnidirectional flight,” *The International Journal of Robotics Research*, vol. 39, no. 10-11, pp. 1305–1325, 2020.
- [28] A. Sanchez, J. Escareño, O. Garcia, and R. Lozano, “Autonomous hovering of a noncyclic tiltrotor uav: modeling, control and implementation,” *IFAC Proceedings Volumes*, vol. 41, no. 2, pp. 803–808, 2008, 17th IFAC World Congress. [Online]. Available: <https://www.sciencedirect.com/science/article/pii/S1474667016390486>.
- [29] G. R. Flores Colunga, “Design, modeling and control of a convertible mini airplane having four tilting rotors,” Theses, Université de Technologie de Compiègne, Oct. 2014. [Online]. Available: <https://theses.hal.science/tel-01297721>.
- [30] D. Kotarski, P. Piljek, J. Kasać, and D. Majetić, “Performance analysis of fully actuated multirotor unmanned aerial vehicle configurations with passively tilted rotors,” *Applied Sciences*, vol. 11, no. 18, p. 8786, 2021.
- [31] N. Staub, D. Bicego, Q. Sablé, V. Arellano, S. Mishra, and A. Franchi, “Towards a flying assistant paradigm: the othex,” pp. 6997–7002, 2018.
- [32] E. Kaufman, K. Caldwell, D. Lee, and T. Lee, “Design and development of a free-floating hexrotor UAV for 6-DOF maneuvers,” Mar. 2014, pp. 1–10.
- [33] B. Crowther, A. Lanzon, M. Maya-Gonzalez, and D. Langkamp, “Kinematic analysis and control design for a nonplanar multirotor vehicle,” *Journal of Guidance, Control, and Dynamics*, vol. 34, no. 4, pp. 1157–1171, 2011.

- [34] J. Escareno, A. Sanchez, O. Garcia, and R. Lozano, “Triple tilting rotor mini-uav: modeling and embedded control of the attitude,” in *2008 American Control Conference*, 2008, pp. 3476–3481.
- [35] M. Hamandi, F. Usai, Q. Sablé, N. Staub, M. Tognon, and A. Franchi, “Design of multicopter aerial vehicles: a taxonomy based on input allocation,” *The International Journal of Robotics Research*, vol. 40, no. 8-9, pp. 1015–1044, 2021.
- [36] R. Rashad, J. Goerres, R. Aarts, J. B. C. Engelen, and S. Stramigioli, “Fully actuated multicopter uavs: a literature review,” *IEEE Robotics Automation Magazine*, vol. 27, no. 3, pp. 97–107, 2020.
- [37] Y. Qin, W. Xu, A. Lee, and F. Zhang, “Gemini: a compact yet efficient bi-copter uav for indoor applications,” *IEEE Robotics and Automation Letters*, vol. 5, no. 2, pp. 3213–3220, 2020.
- [38] D. Kastelan, M. Konz, and J. Rudolph, “Fully actuated tri-copter with pilot-supporting control,” *IFAC-PapersOnLine*, vol. 48, no. 9, pp. 79–84, 2015, 1st IFAC Workshop on Advanced Control and Navigation for Autonomous Aerospace Vehicles ACNAAV’15. [Online]. Available: <https://www.sciencedirect.com/science/article/pii/S2405896315009301>.
- [39] R. Watson, M. Kamel, D. Zhang, *et al.*, “Dry coupled ultrasonic non-destructive evaluation using an over-actuated unmanned aerial vehicle,” *IEEE Transactions on Automation Science and Engineering*, vol. 19, no. 4, pp. 2874–2889, 2022.
- [40] P. Zheng, X. Tan, B. B. Kocer, E. Yang, and M. Kovac, “Tilt-drone: a fully-actuated tilting quadrotor platform,” *IEEE Robotics and Automation Letters*, vol. 5, no. 4, pp. 6845–6852, 2020.
- [41] M. Ryll, H. H. Bühlhoff, and P. R. Giordano, “A novel overactuated quadrotor unmanned aerial vehicle: modeling, control, and experimental validation,” *IEEE Transactions on Control Systems Technology*, vol. 23, no. 2, pp. 540–556, 2015.
- [42] M. Kamel, S. Verling, O. Elkhatib, *et al.*, “The voliro omniorientational hexacopter: an agile and maneuverable tiltable-rotor aerial vehicle,” *IEEE Robotics Automation Magazine*, vol. 25, no. 4, pp. 34–44, 2018.
- [43] T. Shim and H. Bang, “Autonomous landing of uav using vision based approach and pid controller based outer loop,” in *2018 18th International Conference on Control, Automation and Systems (ICCAS)*, IEEE, 2018, pp. 876–879.
- [44] P. Castillo, R. Lozano, and A. E. Dzul, *Modelling and control of mini-flying machines*. Springer Science & Business Media, 2005.

- [45] Z. Peng and L. Jikai, “On new uav flight control system based on kalman & pid,” in *2011 2nd International Conference on Intelligent Control and Information Processing*, IEEE, vol. 2, 2011, pp. 819–823.
- [46] A. M. Elhennawy and M. K. Habib, “Nonlinear robust control of a quadcopter: implementation and evaluation,” in *IECON 2018-44th Annual Conference of the IEEE Industrial Electronics Society*, IEEE, 2018, pp. 3782–3787.
- [47] L. E. Muñoz, O. Santos, P. Castillo, and I. Fantoni, “Energy-based nonlinear control for a quadrotor rotorcraft,” in *2013 american control conference*, IEEE, 2013, pp. 1177–1182.
- [48] G. V. Raffo, M. G. Ortega, and F. R. Rubio, “Backstepping/nonlinear h_∞ control for path tracking of a quadrotor unmanned aerial vehicle,” in *2008 American Control Conference*, IEEE, 2008, pp. 3356–3361.
- [49] K. Nonami, F. Kendoul, S. Suzuki, *et al.*, “Autonomous control of a mini quadrotor vehicle using lqg controllers,” *Autonomous flying robots: unmanned aerial vehicles and micro aerial vehicles*, pp. 61–76, 2010.
- [50] L. Yu, G. He, S. Zhao, and X. Wang, “Dynamic inversion-based sliding mode control of a tilt tri-rotor uav,” in *2019 12th Asian Control Conference (ASCC)*, IEEE, 2019, pp. 1637–1642.
- [51] V. D. Quy, N. Q. Hung, P. X. Tung, *et al.*, “Sliding mode control for the hovering flight of a tilt tri-rotor uav,” in *2022 11th International Conference on Control, Automation and Information Sciences (ICCAIS)*, IEEE, 2022, pp. 448–454.
- [52] L. Bauersfeld, L. Spannagl, G. J. Ducard, and C. H. Onder, “Mpc flight control for a tilt-rotor vtol aircraft,” *IEEE Transactions on Aerospace and Electronic Systems*, vol. 57, no. 4, pp. 2395–2409, 2021.
- [53] K. Benkhoud and S. Bouallègue, “Model predictive control design for a convertible quad tilt-wing uav,” in *2016 4th International Conference on Control Engineering & Information Technology (CEIT)*, IEEE, 2016, pp. 1–6.
- [54] R. Andrade, G. V. Raffo, and J. E. Normey-Rico, “Model predictive control of a tilt-rotor uav for load transportation,” in *2016 European Control Conference (ECC)*, IEEE, 2016, pp. 2165–2170.
- [55] S. Bertrand, “Commande de drone miniature à voilure tournante,” 2007NICE4035, Ph.D. dissertation, 2007, 1 vol. (224 p.) [Online]. Available: <http://www.theses.fr/2007NICE4035/document>.
- [56] J. Marzat, S. Bertrand, A. Eudes, M. Sanfourche, and J. Moras, “Reactive mpc for autonomous mav navigation in indoor cluttered environments: flight experiments,” *IFAC-PapersOnLine*, vol. 50, no. 1, pp. 15 996–16 002, 2017.

- [57] J. F. Cayero Becerra, J. Cugueró Escofet, and B. Morcego Seix, “Impedance control of a planar quadrotor with an extended kalman filter external wrench estimator,” in *euRathlon/SHERPA summer school 2015 on field robotics*, 2015, pp. 1–5.
- [58] J. Cayero, B. Morcego, and J. Cugueró, “Estimating external disturbances in uavs using linear and spin momenta,” in *2018 Annual American Control Conference (ACC)*, IEEE, 2018, pp. 4317–4323.
- [59] J.-A. Escareño, A. Belbachir, T. Raharijaona, and S. Bouchafa, “Generalized disturbance estimation via eslkf for the motion control of rotorcraft having a rod-suspended load,” in *ICINCO (2)*, 2016, pp. 526–533.
- [60] A. Mohammadi, M. Tavakoli, H. J. Marquez, and F. Hashemzadeh, “Nonlinear disturbance observer design for robotic manipulators,” *Control Engineering Practice*, vol. 21, no. 3, pp. 253–267, 2013.
- [61] T. Tomić and S. Haddadin, “A unified framework for external wrench estimation, interaction control and collision reflexes for flying robots,” in *2014 IEEE/RSJ international conference on intelligent robots and systems*, IEEE, 2014, pp. 4197–4204.
- [62] M. Wilmsen, C. Yao, M. Schuster, S. Li, and K. Janschek, “Nonlinear wrench observer design for an aerial manipulator,” *IFAC-PapersOnLine*, vol. 52, no. 22, pp. 1–6, 2019, 1st IFAC Workshop on Robot Control WROCO 2019. [Online]. Available: <https://www.sciencedirect.com/science/article/pii/S2405896319309735>.
- [63] E. Reyes-Valeria, R. Enriquez-Caldera, S. Camacho-Lara, and J. Guichard, “Lqr control for a quadrotor using unit quaternions: modeling and simulation,” in *CONIELECOMP 2013, 23rd International Conference on Electronics, Communications and Computing*, IEEE, 2013, pp. 172–178.
- [64] L. Martins, C. Cardeira, and P. Oliveira, “Linear quadratic regulator for trajectory tracking of a quadrotor,” *IFAC-PapersOnLine*, vol. 52, no. 12, pp. 176–181, 2019.
- [65] E. Abouselima, D. Ichalal, and S. Mammar, “Quadrotor control and actuator fault detection: lqg versus robust H_2/H_∞ observer,” in *2019 4th Conference on Control and Fault Tolerant Systems (SysTol)*, IEEE, 2019, pp. 86–91.
- [66] H. G. Malkapure and M. Chidambaram, “Comparison of two methods of incorporating an integral action in linear quadratic regulator,” *IFAC Proceedings Volumes*, vol. 47, no. 1, pp. 55–61, 2014.
- [67] J. C. Doyle, “Guaranteed margins for lqg regulators,” *IEEE Transactions on automatic Control*, vol. 23, no. 4, pp. 756–757, 1978.
- [68] R. B. Anderson, J. A. Marshall, and A. L’Afflitto, “Constrained Robust Model Reference Adaptive Control of a Tilt-Rotor Quadcopter Pulling an Unmodeled Cart,” *IEEE*

- Transactions on Aerospace and Electronic Systems*, vol. 57, no. 1, pp. 39–54, Feb. 2021, Conference Name: IEEE Transactions on Aerospace and Electronic Systems.
- [69] R. B. Anderson, J.-P. Burke, J. A. Marshall, and A. L’Afflitto, “Robust Adaptive Control for Constrained Tilt-Rotor Quadcopters of Unknown Inertial Properties,” in *2019 American Control Conference (ACC)*, ISSN: 2378-5861, Jul. 2019, pp. 2922–2927.
- [70] S. K. Phang, K. Li, B. M. Chen, T. H. Lee, K. P. Valavanis, and G. J. Vachtsevanos, *Systematic design methodology and construction of micro aerial quadrotor vehicles*. Springer, 2014, pp. 181–206.
- [71] D. Mellinger and V. Kumar, “Minimum snap trajectory generation and control for quadrotors,” pp. 2520–2525, 2011.
- [72] M. Ryll, H. H. Bühlhoff, and P. R. Giordano, “Modeling and control of a quadrotor uav with tilting propellers,” pp. 4606–4613, 2012.
- [73] T. Hamel, R. Mahony, R. Lozano, and J. Ostrowski, “Dynamic modelling and configuration stabilization for an x4-flyer,” *IFAC Proceedings Volumes*, vol. 35, no. 1, pp. 217–222, 2002, 15th IFAC World Congress. [Online]. Available: <https://www.sciencedirect.com/science/article/pii/S1474667015392697>.
- [74] L. Haixu, Q. Xiangju, and W. Weijun, “Multi-body motion modeling and simulation for tilt rotor aircraft,” *Chinese Journal of Aeronautics*, vol. 23, no. 4, pp. 415–422, 2010. [Online]. Available: <https://www.sciencedirect.com/science/article/pii/S1000936109602363>.
- [75] R. B. Anderson, J. A. Marshall, and A. L’Afflitto, “Constrained robust model reference adaptive control of a tilt-rotor quadcopter pulling an unmodeled cart,” *IEEE Transactions on Aerospace and Electronic Systems*, vol. 57, no. 1, pp. 39–54, 2021.
- [76] M. Bangura, R. Mahony, *et al.*, “Nonlinear dynamic modeling for high performance control of a quadrotor,” 2012.
- [77] K. P. Jain, T. Fortmuller, J. Byun, S. A. Mäkiharju, and M. W. Mueller, “Modeling of aerodynamic disturbances for proximity flight of multirotors,” pp. 1261–1269, 2019.
- [78] J. Ye, J. Wang, T. Song, Z. Wu, and P. Tang, “Nonlinear modeling the quadcopter considering the aerodynamic interaction,” *IEEE Access*, vol. 9, pp. 134 716–134 732, 2021.
- [79] G. Hoffmann, H. Huang, S. Waslander, and C. Tomlin, “Quadrotor helicopter flight dynamics and control: theory and experiment,” p. 6461, 2007.
- [80] M. Faessler, A. Franchi, and D. Scaramuzza, “Differential flatness of quadrotor dynamics subject to rotor drag for accurate tracking of high-speed trajectories,” *IEEE Robotics and Automation Letters*, vol. 3, no. 2, pp. 620–626, 2017.

- [81] D. D. C. Bernard, F. Riccardi, M. Giurato, and M. Lovera, “A dynamic analysis of ground effect for a quadrotor platform,” *IFAC-PapersOnLine*, vol. 50, no. 1, pp. 10 311–10 316, 2017, 20th IFAC World Congress. [Online]. Available: <https://www.sciencedirect.com/science/article/pii/S2405896317320803>.
- [82] P. Sanchez-Cuevas, G. Heredia, and A. Ollero, “Characterization of the aerodynamic ground effect and its influence in multirotor control,” *International Journal of Aerospace Engineering*, vol. 2017, 2017.
- [83] A. Gong, R. MacNeill, and D. Verstraete, “Performance testing and modeling of a brushless dc motor, electronic speed controller and propeller for a small uav application,” in *2018 joint propulsion conference*, 2018, p. 4584.
- [84] M. Bangura, H. Lim, H. J. Kim, and R. Mahony, “Aerodynamic power control for multirotor aerial vehicles,” pp. 529–536, 2014.
- [85] M. Bangura and R. Mahony, “Thrust control for multirotor aerial vehicles,” *IEEE Transactions on Robotics*, vol. 33, no. 2, pp. 390–405, 2017.
- [86] M. Cunningham and J. E. Hubbard, “System identification of a multi-rotor uav actuator,” in *AIAA Atmospheric Flight Mechanics Conference*, 2017, p. 1187.
- [87] L. Besnard, Y. B. Shtessel, and B. Landrum, “Quadrotor vehicle control via sliding mode controller driven by sliding mode disturbance observer,” *Journal of the Franklin Institute*, vol. 349, no. 2, pp. 658–684, 2012, Advances in Guidance and Control of Aerospace Vehicles using Sliding Mode Control and Observation Techniques. [Online]. Available: <https://www.sciencedirect.com/science/article/pii/S0016003211001839>.
- [88] M. Hamandi, M. Tognon, and A. Franchi, “Direct acceleration feedback control of quadrotor aerial vehicles,” in *2020 IEEE International Conference on Robotics and Automation (ICRA)*, 2020, pp. 5335–5341.
- [89] T. Madani and A. Benallegue, “Backstepping control for a quadrotor helicopter,” in *2006 IEEE/RSJ International Conference on Intelligent Robots and Systems*, 2006, pp. 3255–3260.
- [90] H. Jasak, “Openfoam: open source cfd in research and industry,” *International Journal of Naval Architecture and Ocean Engineering*, vol. 1, no. 2, pp. 89–94, 2009.
- [91] L. Gagnon, G. Quaranta, and M. Schwaiger, “Open-source 3d cfd of a quadrotor cyclogyro aircraft,” in *OpenFOAM®: Selected Papers of the 11th Workshop*, Springer, 2019, pp. 373–388.
- [92] B. P. Duisterhof, S. Li, J. Burgués, V. J. Reddi, and G. C. H. E. de Croon, “Sniffy bug: a fully autonomous swarm of gas-seeking nano quadcopters in cluttered environments,” in *2021 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 2021, pp. 9099–9106.

- [93] M. Prieto, M. S. Escarti-Guillem, and S. Hoyas, “Aerodynamic optimization of a vtol drone using winglets,” *Results in Engineering*, vol. 17, p. 100 855, 2023. [Online]. Available: <https://www.sciencedirect.com/science/article/pii/S2590123022005254>.
- [94] B. Carmichael, “Low reynolds number airfoil survey, volume 1,” Tech. Rep., 1981.
- [95] A. Garofano-Soldado, P. J. Sanchez-Cuevas, G. Heredia, and A. Ollero, “Numerical-experimental evaluation and modelling of aerodynamic ground effect for small-scale tilted propellers at low reynolds numbers,” *Aerospace Science and Technology*, p. 107 625, 2022.
- [96] L. Caretto, A. Gosman, S. Patankar, and D. Spalding, “Two calculation procedures for steady, three-dimensional flows with recirculation,” in *Proceedings of the Third International Conference on Numerical Methods in Fluid Mechanics: Vol. II Problems of Fluid Mechanics*, Springer, 1973, pp. 60–68.
- [97] R. I. Issa, “Solution of the implicitly discretised fluid flow equations by operator-splitting,” *Journal of computational physics*, vol. 62, no. 1, pp. 40–65, 1986.
- [98] P. Farrell and J. Maddison, “Conservative interpolation between volume meshes by local galerkin projection,” *Computer Methods in Applied Mechanics and Engineering*, vol. 200, no. 1-4, pp. 89–100, 2011.
- [99] Y. Lei, Y. Li, and J. Wang, “Aerodynamic analysis of an orthogonal octorotor uav considering horizontal wind disturbance,” *Aerospace*, vol. 10, no. 6, 2023. [Online]. Available: <https://www.mdpi.com/2226-4310/10/6/525>.
- [100] S. J. Lee, S. H. Kim, and H. J. Kim, “Robust translational force control of multi-rotor uav for precise acceleration tracking,” *IEEE Transactions on Automation Science and Engineering*, vol. 17, no. 2, pp. 562–573, 2019.
- [101] J. F. Guerrero-Castellanos, S. Durand, G. A. Munoz-Hernandez, *et al.*, “Bounded attitude control with active disturbance rejection capabilities for multirotor uavs,” *Applied Sciences*, vol. 11, no. 13, 2021. [Online]. Available: <https://www.mdpi.com/2076-3417/11/13/5960>.
- [102] E. Sariyildiz, R. Oboe, and K. Ohnishi, “Disturbance observer-based robust control and its applications: 35th anniversary overview,” *IEEE Transactions on Industrial Electronics*, vol. 67, no. 3, pp. 2042–2053, 2019.
- [103] D. Hentzen, T. Stastny, R. Siegwart, and R. Brockers, “Disturbance estimation and rejection for high-precision multirotor position control,” in *2019 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 2019, pp. 2797–2804.
- [104] K. Coleman, H. Bai, and C. N. Taylor, “Extended invariant-ekf designs for state and additive disturbance estimation,” *Automatica*, vol. 125, p. 109 464, 2021. [On-

- line]. Available: <https://www.sciencedirect.com/science/article/pii/S0005109820306622>.
- [105] H. Hamadi, B. Lussier, I. Fantoni, C. Francis, and H. Shraim, “Observer-based super twisting controller robust to wind perturbation for multirotor uav,” in *2019 International Conference on Unmanned Aircraft Systems (ICUAS)*, IEEE, 2019, pp. 397–405.
- [106] A. Chalanga, S. Kamal, L. M. Fridman, B. Bandyopadhyay, and J. A. Moreno, “Implementation of super-twisting control: super-twisting and higher order sliding-mode observer-based approaches,” *IEEE Transactions on Industrial Electronics*, vol. 63, no. 6, pp. 3677–3685, 2016.
- [107] J. Davila, L. Fridman, and A. Levant, “Second-order sliding-mode observer for mechanical systems,” *IEEE Transactions on Automatic Control*, vol. 50, no. 11, pp. 1785–1789, 2005.
- [108] Y. Shtessel, C. Edwards, L. Fridman, and A. Levant, “Observation and identification via hoshm observers,” in *Sliding Mode Control and Observation*. New York, NY: Springer New York, 2014, pp. 251–290. [Online]. Available: https://doi.org/10.1007/978-0-8176-4893-0_7.
- [109] P. Foehn and D. Scaramuzza, “Onboard state dependent lqr for agile quadrotors,” in *2018 IEEE International Conference on Robotics and Automation (ICRA)*, 2018, pp. 6566–6572.
- [110] L. Martins, C. Carneira, and P. Oliveira, “Linear quadratic regulator for trajectory tracking of a quadrotor,” *IFAC-PapersOnLine*, vol. 52, no. 12, pp. 176–181, 2019, 21st IFAC Symposium on Automatic Control in Aerospace ACA 2019. [Online]. Available: <https://www.sciencedirect.com/science/article/pii/S2405896319311450>.
- [111] K. Benkhoud and S. Bouallègue, “Model predictive control design for a convertible quad tilt-wing uav,” in *2016 4th International Conference on Control Engineering Information Technology (CEIT)*, 2016, pp. 1–6.
- [112] R. Andrade, G. V. Raffo, and J. E. Normey-Rico, “Model predictive control of a tilt-rotor uav for load transportation,” in *2016 European Control Conference (ECC)*, 2016, pp. 2165–2170.
- [113] H. Xiao, D. Zhao, S. Gao, and S. K. Spurgeon, “Sliding mode predictive control: a survey,” *Annual Reviews in Control*, vol. 54, pp. 148–166, 2022. [Online]. Available: <https://www.sciencedirect.com/science/article/pii/S1367578822000876>.
- [114] J.-S. Zhou, Z.-Y. Liu, and R. Pei, “A new nonlinear model predictive control scheme for discrete-time system based on sliding mode control,” in *Proceedings of the 2001 American Control Conference. (Cat. No.01CH37148)*, vol. 4, 2001, 3079–3084 vol.4.

- [115] G. P. Incremona, A. Ferrara, and L. Magni, “Mpc for robot manipulators with integral sliding modes generation,” *IEEE/ASME Transactions on Mechatronics*, vol. 22, no. 3, pp. 1299–1307, 2017.
- [116] T. J. Stastny, A. Dash, and R. Siegwart, “Nonlinear mpc for fixed-wing uav trajectory tracking: implementation and flight experiments,” in *AIAA guidance, navigation, and control conference*, 2017, p. 1512.
- [117] P. Sistu, R. Gopinath, and B. Bequette, “A disturbance estimator for model predictive control,” *IFAC Proceedings Volumes*, vol. 27, no. 2, pp. 155–160, 1994, IFAC Symposium on Advanced Control of Chemical Processes, Kyoto, Japan, 25-27 May 1994. [Online]. Available: <https://www.sciencedirect.com/science/article/pii/S1474667017481468>.
- [118] L. Yu, G. He, S. Zhao, and X. Wang, “Dynamic inversion-based sliding mode control of a tilt tri-rotor uav,” in *2019 12th Asian Control Conference (ASCC)*, 2019, pp. 1637–1642.
- [119] L. T. Tuyet Nhung, V. Dinh Quy, N. Q. Hung, and P. Xuan Tung, “Sliding mode control for the hovering flight of a tilt tri-rotor uav,” in *2022 11th International Conference on Control, Automation and Information Sciences (ICCAIS)*, 2022, pp. 448–454.
- [120] S. Sarpturk, Y. Istefanopulos, and O. Kaynak, “On the stability of discrete-time sliding mode control systems,” *IEEE Transactions on Automatic Control*, vol. 32, no. 10, pp. 930–932, 1987.
- [121] J. Zhang, P. Shi, Y. Xia, and H. Yang, “Discrete-time sliding mode control with disturbance rejection,” *IEEE Transactions on Industrial Electronics*, vol. 66, no. 10, pp. 7967–7975, 2018.
- [122] [Online]. Available: <https://docs.px4.io/main/en/simulation/>.
- [123] [Online]. Available: https://clover.coex.tech/en/simulation_native.html.
- [124] [Online]. Available: <https://buildroot.org/>.
- [125] [Online]. Available: <https://github.com/PX4/PX4-Autopilot>.
- [126] [Online]. Available: <https://zortrax.com/3d-printers/m300-dual/>.
- [127] J. W. Then and K.-r. Chiang, “Experimental determination of moments of inertia by the bifilar pendulum method,” *American Journal of Physics*, vol. 38, no. 4, pp. 537–539, 1970.
- [128] F. Bos, A. A. Meera, D. Benders, and M. Wisse, “Free energy principle for state and input estimation of a quadcopter flying in wind,” in *2022 International Conference on Robotics and Automation (ICRA)*, 2022, pp. 5389–5395.
- [129] H. C. Choi, I. Wee, M. Corah, *et al.*, “Baxter: bi-modal aerial-terrestrial hybrid vehicle for long-endurance versatile mobility,” in *Experimental Robotics: The 17th International Symposium*, Springer, 2021, pp. 60–72.

- [130] [Online]. Available: <https://optitrack.com/cameras/primex-41/>.
- [131] N. J. Unwin and A. J. Postula, “Motor bourn magnetic noise filtering for magnetometers in micro-rotary aerial vehicles,” *Sensors & Transducers*, vol. 193, no. 10, p. 161, 2015.
- [132] J. Thomas, G. Loianno, K. Sreenath, and V. Kumar, “Toward image based visual servoing for aerial grasping and perching,” in *2014 IEEE International Conference on Robotics and Automation (ICRA)*, 2014, pp. 2113–2118.
- [133] [Online]. Available: <https://www.youtube.com/@theslowmoguys>.
- [134] [Online]. Available: <https://www.kinovea.org/>.
- [135] G. Sachs, J. Traugott, A. Nesterova, and F. Bonadonna, “Experimental verification of dynamic soaring in albatrosses,” *Journal of Experimental Biology*, vol. 216, no. 22, pp. 4222–4232, 2013.
- [136] G. Sachs, “Minimum shear wind strength required for dynamic soaring of albatrosses,” *Ibis*, vol. 147, no. 1, pp. 1–10, 2005.
- [137] G. Sachs, “Kinetic energy in dynamic soaring—inertial speed and airspeed,” *Journal of Guidance, Control, and Dynamics*, vol. 42, no. 8, pp. 1812–1821, 2019.
- [138] T. Flanzer, G. Bower, and I. Kroo, “Robust trajectory optimization for dynamic soaring,” in *AIAA guidance, navigation, and control conference*, 2012, p. 4603.
- [139] M. Nekoui, J. Khaghani, R. Nasiri, and M. N. Ahmadabadi, “Natural dynamics exploitation of dynamic soaring: towards bio-inspired and energy efficient flying locomotion,” in *2018 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, IEEE, 2018, pp. 8171–8176.
- [140] C. J. Pennycuik, *Modelling the flying bird*. Elsevier, 2008.
- [141] A. M. Smith, “High-lift aerodynamics,” *Journal of Aircraft*, vol. 12, no. 6, pp. 501–530, 1975.
- [142] J. E. Aviles Mejia, “Robotique mobile autonome en milieu ouvert,” 2023LIMO0017, Ph.D. dissertation, 2023. [Online]. Available: <http://www.theses.fr/2023LIMO0017/document>.
- [143] S. Garrido-Jurado, R. Muñoz-Salinas, F. Madrid-Cuevas, and M. Marín-Jiménez, “Automatic generation and detection of highly reliable fiducial markers under occlusion,” *Pattern Recognition*, vol. 47, no. 6, pp. 2280–2292, 2014. [Online]. Available: <https://www.sciencedirect.com/science/article/pii/S0031320314000235>.
- [144] [Online]. Available: https://docs.opencv.org/4.x/d5/dae/tutorial_aruco_detection.html.
- [145] [Online]. Available: https://docs.opencv.org/4.x/db/da9/tutorial_aruco_board_detection.html.

Liste des Travaux

Conférences internationales à comité de lecture

T. Zioud, J. Escareno, and O. Labbani-Igbida, “Real-time of-based trajectory control of a uas rotorcraft based on integral extended-state lqg,” *2022 IEEE 18th International Conference on Automation Science and Engineering (CASE)*, pp. 1423–1430, Aug. 2022.

T. Zioud, J. Escareno, and O. Labbani-Igbida, “Dynamic/cfd modeling, control and energy-consumption comparative analysis of a quad-tilting rotor vtol uas,” *2022 International Conference on Unmanned Aircraft Systems (ICUAS)*, pp. 849–858, Jun. 2022.

Journal internationale à comité de lecture

T. Zioud, J. Escareno, and O. Labbani-Igbida, “Dynamic modeling and mpc driven robust control of a rotorcraft having four tilting-rotors,” *IEEE Transactions on Aerospace and Electronic Systems*, vol. Under review, 2023.

Conception, modélisation et commande robuste de drones à rotors pivotants pour l'interaction aérienne autonome

Résumé : Ce travail s'inscrit dans le cadre du projet régional qui a pour objectif de concevoir un drone échantillonneur polyvalent, capable à la fois de prélever des échantillons d'eau et d'embarquer divers capteurs pour effectuer des mesures physiques et chimiques sur site. Selon la nature de la tâche à accomplir, le drone doit être en mesure de maintenir une position stationnaire (par exemple, lors de la collecte d'un échantillon d'eau) ou de se déplacer en vol (par exemple, lors de mesures de température ou de pH).

Au cours de cette thèse, nous avons développé un quadrirotor baptisé Xtilt, doté de propulseurs pivotants, lui permettant de combiner deux modes de vol, à savoir le vol stationnaire et le vol de croisière. Sur le plan scientifique, notre travail a consisté en une modélisation du Xtilt en utilisant le formalisme d'Euler-Lagrange, en prenant en compte les effets aérodynamiques, ainsi qu'une étude détaillée en mécanique des fluides computationnelle (CFD) du châssis du drone.

Pour faire face aux perturbations internes et externes, nous avons proposé des lois de commande robustes, notamment la commande linéaire intégrale quadratique gaussienne à état étendu (ES-iLQG). Cette méthode comprend une amélioration de l'approche LQG intégrant un filtre de Kalman linéaire à état étendu (ES-LKF) pour l'estimation des perturbations et un régulateur intégral linéaire quadratique (iLQR). Nous avons également testé d'autres lois de commande robustes basées sur la commande prédictive MPC (Model Predictive Control), y compris l'intégral MPC et MPSMC. Ces approches visent à combiner à la fois des lois de commande robustes et optimales, en tenant compte des limitations du drone.

Enfin, nous avons réalisé un prototype opérationnel capable de suivre des trajectoires en position et en vitesse. Le châssis a été conçu en CAO et fabriqué en impression 3D. Notre architecture système intègre les différentes lois de commande développées en langage C/C++ au sein de l'écosystème ROS sur calculateur embarqué. Des résultats expérimentaux prometteurs ont été obtenus lors de scénarios de suivi de trajectoire : une trajectoire hélicoïdale et une trajectoire en forme de 8, en présence de perturbations.

Mots clés : Drones, UAV, Multirotor, Propulseurs pivotants, Contrôle, Commande Optimale, Commande basée sur le modèle, Commande robuste, Observation, Perturbations.

Design, Modeling, and Robust Control of a Quad Tilting-Rotor Drone for Autonomous Aerial Interaction

Abstract: This research is part of a regional project aimed at designing a versatile sampling drone capable of both collecting water samples and carrying various sensors for on-site physical and chemical measurements. Depending on the task at hand, the drone must be able to maintain a stationary position (e.g., during water sample collection) or move in flight (e.g., while measuring temperature or pH). During this doctoral study, we developed a quadrotor named Xtilt, equipped with tilting thrusters, enabling it to seamlessly transition between two flight modes: hovering and cruising. From a scientific perspective, our work encompassed modeling Xtilt using the Euler-Lagrange formalism, taking into account aerodynamic effects, and conducting an extensive study in computational fluid dynamics (CFD) of the drone's chassis.

To address internal and external disturbances, we introduced robust control laws, including the Extended State Integral Quadratic Gaussian Control (ES-iLQG). This methodology represents an enhancement of the LQG approach by incorporating an Extended State Kalman Filter (ES-LKF) for disturbance estimation and a Linear Quadratic Integral Regulator (iLQR). We also explored alternative robust control laws based on Model Predictive Control (MPC), such as Integral MPC and MPSMC. These approaches aim to harmonize robustness with optimality while accommodating the drone's limitations. Lastly, we constructed an operational prototype capable of tracking position and velocity trajectories. The chassis was designed using Computer-Aided Design (CAD) software and manufactured using 3D printing technology. Our system architecture integrates the various control laws developed in C/C++ within the ROS ecosystem on an embedded computing platform. Promising experimental results were achieved during trajectory tracking scenarios, which included helical and figure-eight trajectories, even in the presence of disturbances.

Keywords: Drones, UAS, Multirotor, Tilting Thruster, Control, Optimal Control, Model-Based Control, Robust Control, Observation, Disturbances.