

Doctorat de l'Université de Toulouse

délivré par l'ENAC

Contrôle actif de la turbulence sur un micro-drone convertible

Thèse présentée et soutenue, le 6 décembre 2024 par

Florian SANSOU

École doctorale

SYSTEMES

Spécialité

Automatique et Systèmes Embarqués

Unité de recherche

ENAC-LAB - Laboratoire de Recherche ENAC

Thèse dirigée par

Fabrice DEMOURANT, Gautier HATTENBERGER et Thomas LOQUEN

Composition du jury

M. Paolo ROBUFFO GIORDANO, Rapporteur, IRISA - Institut de Recherche en Informatique et Systèmes Aléatoires

M. Pascal MORIN, Rapporteur, Institut des systèmes intelligents et robotique (ISIR) - CNRS - INSERM - Sorbonne Université

M. Philippe CHEVREL, Examinateur, IMT Atlantique

Mme Sophie TARBOURIECH, Examinatrice, LAAS-CNRS

M. Fabrice DEMOURANT, Directeur de thèse, Institut Supérieur de l'Aéronautique et de l'Espace

M. Gautier HATTENBERGER, Co-directeur de thèse, Ecole Nationale de l'Aviation Civile

Membres invités

M. Luca ZACCARIAN, LAAS-CNRS

M. Thomas LOQUEN, ONERA - DTIS

Remerciements

Table des matières

Liste des figures	x
Liste des tableaux	xi
Liste des algorithmes	xiii
Liste des abréviations	xv
Introduction	1
1 Généralités sur les drones convertibles	7
1.1 Micro-drones convertibles	7
1.1.1 Efficacité énergétique et domaine de vol	7
1.1.2 Types d'architectures des drones convertibles	8
1.2 Propriétés des <i>tailsitters</i> et des <i>freewings</i>	12
1.2.1 Actionnement	12
1.2.2 Aérodynamique	13
1.3 Modélisations	14
1.3.1 Dynamique du système	14
1.3.2 Modèle de perturbation	15
1.4 Actionnement	16
1.5 Architecture de commande de vol	17
1.6 Architecture de commande	17
1.7 Méthodes de résolution	22
1.8 Technologies et réalisations	25
1.9 Conclusion	27
2 Modélisation d'un drone convertible : DarkO	29
2.1 Modèle du drone DarkO	29
2.1.1 Modèle non-linéaire complet	32
2.1.2 Modèle non-linéaire simplifié à basse vitesse	34
2.2 Caractérisation de la maquette	36
2.2.1 Identification des paramètres du modèle	36
2.2.2 Modélisation des actionneurs	39
2.3 Équilibres stationnaires	40

2.3.1	Équilibre stationnaire sans vent	40
2.3.2	Équilibre stationnaire en présence de vent	41
2.4	Dynamiques linéarisées	47
2.4.1	Dynamique linéarisée sans vent	47
2.4.2	Dynamique linéarisée en présence de vent	48
2.5	Conclusion	54
3	Travail préliminaire : commande hybride de drone	57
3.1	Motivation	57
3.2	Contrôleur par retour d'état non-linéaire	58
3.3	Contrôleur par retour d'état linéaire	61
3.4	Conception d'une commande locale-globale	63
3.5	Conclusion	67
4	Commande robuste longitudinale d'une maquette	69
4.1	Motivation	69
4.2	Présentation de la maquette expérimentale	70
4.2.1	Description physique, capteurs et actionnements	70
4.2.2	Simulation des mouvements du drone	73
4.3	Contrôle linéaire, architecture PI sur maquette à 3 DOF	74
4.3.1	Description du schéma de contrôle	74
4.3.2	Schéma de synthèse H_∞	76
4.4	Résultats	78
4.5	Conclusion	79
5	Synthèse d'un retour statique de sortie par optimisation LMI	81
5.1	Motivation	81
5.2	Synthèse d'un contrôleur statique de sortie	82
5.2.1	Algorithme itératif pour la conception de retour statique de sortie	83
5.2.2	Architecture de commande	87
5.3	Résultats	88
5.3.1	Résultats de simulation	88
5.3.2	Résultats expérimentaux	90
5.4	Conclusion	93

6 Expérimentation d'un contrôle robuste pour DarkO	95
6.1 Motivation	95
6.2 Contrôleur structuré proportionnel intégral avec <i>roll off</i>	96
6.2.1 Description du schéma de contrôle	96
6.2.2 Contrôleur optimisé sous contraintes H_∞ , cas sans vent	99
6.2.3 Contrôleur optimisé sous contraintes H_∞ , cas multimodèle	101
6.3 Rafale de vent	107
6.4 Vol expérimental en soufflerie ouverte	108
6.5 Conclusion	111
7 Modèle, commande et expérimentation d'un <i>freewing</i>	113
7.1 Motivation	113
7.2 Réflexion autour de diverses architectures	114
7.3 Design et modélisation d'un drone : Colibri	117
7.3.1 Description de l'architecture	117
7.3.2 Modélisation	120
7.4 Estimation d'état	127
7.4.1 Placement des capteurs	127
7.4.2 Estimation de la vitesse angulaire	128
7.4.3 Estimation de l'état de l'aile	130
7.4.4 Estimation de l'orientation du fuselage	131
7.5 Inversion non-linéaire incrémentale de la dynamique du drone	132
7.6 Stabilisation du fuselage	132
7.7 Expérimentations	133
7.8 Vol avec un contrôleur INDI unifié	134
7.9 Conclusion	136
Conclusion	139
Annexe technique sur les drones	143
Bibliographie	149

Liste des figures

I.1	Phases de vol d'un drone <i>tailsitter</i> , DarkO.	2
I.2	Maquettes « T-Wing », « MavIon » et « KH-Lion ».	3
1.1	Structure <i>quadplane</i> proposée par [Avy 2023].	9
1.2	Structure <i>Tiltrotor</i> proposée par [Flores 2014], dans deux configurations, vol stationnaire et d'avancement.	10
1.3	Structure <i>tiltrotor</i> proposée par [Smeur 2020, Fernandez 2023b, Pfleimlin 2006].	11
1.4	Structure <i>tiltwing</i> proposée par [Dufour 2024, Ostermann 2012]. . . .	11
1.5	Représentation de la pression dynamique sur la voilure de DarkO lors d'un vol d'avancement.	14
1.6	Représentation temporelle des modèles de perturbation de vent. . . .	15
1.7	Actionneurs d'un <i>tailsitter</i> et leurs actions sur les axes principaux. .	16
1.8	Architecture classique de contrôle hiérarchique pour les drones. . . .	17
1.9	Méthodes de commande utilisées sur les architectures <i>tailsitter</i> et <i>freewing</i>	18
1.10	<i>Tailsitter</i> DarkO lors d'un test de robustesse [Olszanecki Barth 2020].	20
1.11	Contrôleur séquencé discret (a) et continu (b) [Ducard 2021]. . . .	21
1.12	Architecture d'un contrôleur MRAC.	21
1.13	Architecture d'un contrôleur H_∞	24
1.14	Architecture d'un autopilote et son environnement.	26
1.15	Contrôleur moteur et servomoteur.	26
2.1	Vue de dessus d'un autopilote Apogee v1.00.	30
2.2	Repère de référence de DarkO avec une représentation schématique des actionneurs.	30
2.3	Réponse entrée-sortie de l'ensemble moteur/hélice.	37
2.4	Montage d'un pendule bifilaire pour l'identification de l'inertie (\mathbf{J}) de DarkO.	37
2.5	Identification de l'inertie (\mathbf{J}), à partir des mesures issues du pendule bifilaire 2.4.	38
2.6	Montage de DarkO sur un banc de mesure face à une soufflerie ouverte.	39

2.7	Les paramètres (τ, δ, θ) de l'ensemble des points d'équilibre (surface) obtenus à l'aide du Théorème 1 et de l'Algorithm 1 pour un vent constant horizontal et vertical (w_{rx}, w_{rz}), avec les saturations des actionneurs (rose) (gauche) et coupe pour $w_{rx} = 0$ (droite).	45
3.1	Schéma de commande non-linéaire basée sur une direction de zéro moment [Michieletto 2020].	58
3.2	Simulation de la loi de commande non-linéaire avec la dynamique de DarkO (2.2).	60
3.3	Simulation du modèle complet (2.2) (ligne continue) et simplifié (2.15) (ligne en pointillé) avec $\mathbf{u} = \mathbf{u}_{\text{lin}}$ défini dans (3.2) et comme condition initiale $\tilde{\mathbf{x}}_0$ dans le bassin d'attraction.	62
3.4	Simulation divergente du modèle complet (2.2) avec $\mathbf{u} = \mathbf{u}_{\text{lin}}$ défini dans (3.2) et une condition initiale $\tilde{\mathbf{x}}_0$ en dehors du bassin d'attraction.	63
3.5	Exemple de l'évolution de $V(\tilde{\mathbf{x}})$, de la variable discrète ℓ et des ensembles continus \mathcal{C}_0 et \mathcal{C}_1 et discrets \mathcal{D}_0 et \mathcal{D}_1	64
3.6	Mécanisme d'hystérésis sur le changement local/global de loi de commande.	65
3.7	Simulation en boucle fermée avec le contrôleur hybride (3.4).	66
4.1	Montage à un degré de liberté.	71
4.2	Modèle de DarkO à un seul degré de liberté devant le <i>WindShape</i>	71
4.3	Architecture d'essai en vol virtuel : <i>WindShape</i> (a) ; capteur de vitesse (b) ; capteur de forces et moments (c) ; encodeur rotatif (d) ; servomoteur (e) ; moteur <i>brushless</i> + ESC (f) ; LabJack (g) ; ordinateur de contrôle (h)	72
4.4	Schéma de commande linéaire, proportionnelle-intégrale.	75
4.5	Résultats expérimentaux.	78
5.1	Structure de l'algorithme d'optimisation.	84
5.2	Augmentation du système.	87
5.3	Réponse du bouclage (5.13) avec une dynamique linéaire (2.30).	89
5.4	Réponse temporelle de l'angle d'incidence θ en boucle fermée.	90
5.5	Modèle expérimental du drone DarkO.	91
5.6	DarkO lors d'un vol expérimental.	92
5.7	Réponse expérimentale en boucle fermée et entrée de commande pour $h = 12$	93

6.1	Schéma de commande intégrale avec la perturbation de vent \mathbf{w} , la perturbation du système à l'entrée \mathbf{d} et à la sortie $\boldsymbol{\nu}$	97
6.2	Simulation du modèle non-linéaire (2.2) (ligne continue) et du modèle linéarisé (2.34) (ligne en pointillé) avec des incrément de vent constants croissants et le contrôleur basé sur l'optimisation sans vent de la section 6.2.2.	101
6.3	Diagrammes des valeurs singulières des fonctions de transfert dans (6.8), à la première itération de l'algorithme 3.	105
6.4	Simulation du modèle non-linéaire (2.2) (ligne continue) et du modèle linéarisé (2.34) (ligne en pointillé) avec des incrément de vent constants croissants, le contrôleur étant réglé à l'aide de l'optimisation multimodèle de l'algorithme 3 dans la section 6.2.3.	106
6.5	Simulation du modèle non-linéaire (2.2) face à une perturbation "Chapeau mexicain" (1.1) avec $f_g = 0.2 \text{ Hz}$	107
6.6	Simulation du modèle non-linéaire (2.2) face à une perturbation "Chapeau mexicain" (1.1) avec $f_g = 1.2 \text{ Hz}$	108
6.7	Simulation du modèle non-linéaire (2.2) face à une perturbation "ondelettes de Morlet" (1.2).	108
6.8	Vol expérimental de DarkO devant la soufflerie ouverte.	109
6.9	Expérience du drone DarkO devant la soufflerie avec des incrément de vent constants croissants (graphique du bas).	110
6.10	Visualisation statistique des performances de vol stationnaire (en rouge la valeur moyenne, en bleu l'écart type à 95% et en pointillé les bornes minimales et maximales).	111
7.1	Montage de la maquette du chapitre 4 avec un réglage de la position du centre de rotation (pièce jaune).	114
7.2	Positionnement avant et arrière du centre de rotation par rapport au centre de gravité.	115
7.3	Maquette de Colibri avec un montage des moteurs en H.	116
7.4	Les repères, inertiel (I) et aile (W), attachés à l'architecture de Colibri.	117
7.5	Modèle 3D de la liaison.	119
7.6	Visualisation de l'encodeur rotatif sur le modèle 3D (gauche) et modèle réel (droite).	119
7.7	Repère inertiel (I), du fuselage (F) et de l'aile (W) et forces agissant sur le drone Colibri.	121

7.8	Simulation de la position et de l'orientation du drone multicorps Colibri en boucle fermée avec un contrôleur à double boucle.	126
7.9	Mesure de l'angle (noir, graphique du haut), mesure de la vitesse angulaire avec le gyroscope de l'aile (bleu, graphique du bas), estimation de la vitesse angulaire par différence finie (vert, graphique du bas) et estimation avec le filtre haut gain (courbes rouges).	129
7.10	Maquette expérimentale du fuselage de Colibri.	133
7.11	Prototype : Colibri.	134
7.12	Position et orientation de l'aile (six premiers graphiques) et mesure de l'angle entre l'aile et le fuselage (dernier graphique), lors d'un vol réel.	135
C.1	Schéma de déplacement d'un drone convertible.	142
A.1	Interface graphique de la station de contrôle au sol.	146
A.2	Schéma de l'ordre d'exécution des codes embarqués [Team 2022]. . .	147
A.3	Algorithme de biais et de gain adaptatif (ABAG) [Franchi 2017]. . .	147

Liste des tableaux

1.1	Comparaison des architectures de drone [Saeed 2018, Table 1].	8
2.1	Paramètres numériques identifiés du modèle DarkO.	31
2.2	Longueur (h) et espacement (D) des fils du pendule pour chacun des axes.	38
6.1	Écart-type du bruit pour la modélisation des capteurs en simulation.	99
6.2	Valeurs des scalaires de pondération positifs W_1 – W_5 utilisés dans l'exécution de l'Algorithme 3.	104
7.1	Paramètres numériques du modèle Colibri.	118
A.1	Évolution des autopilotes Paparazzi sur dix ans.	145

Liste des algorithmes

1	Obtention des paramètres d'équilibre en (2.18).	46
2	Détermination des matrices de la linéarisation de (2.34)	55
3	Réglage itératif et multimodèle des gains du contrôleur.	103

Liste des abréviations

ABAG Biais adaptatif et gain adaptatif *Adaptive Bias and Adaptive Gain*

API Interface de programmation d'application (*Application Programming Interface*)

BMI Inégalités matricielles bilinéaires (*Bilinear Matrix Inequalities*)

CAO Conception assistée par ordinateur

DOF Degrés de liberté (*Degrees of Freedom*)

EKF Filtre de Kalman étendu (*Extended Kalman Filter*)

ENAC École Nationale de l'Aviation Civile

ESC Contrôleurs électroniques de vitesse (*Electronic Speed Controller*)

FDM Impression 3D dépôt de fil fondu (*Fused deposition modeling*)

GCS Station de contrôle au sol *Ground Control Station*

GPS Géo-positionnement par satellite (*Global Positioning System*)

IMU Centrales inertielles (*Inertial Measurement Units*)

LPV Système linéaire à paramètres variants (*Linear Parameter Varying*)

LQR Commande linéaire quadratique (*Linear Quadratic Regulators*)

MAV Micro-drone (*Micro Air Vehicle*)

MCU Microcontrôleurs (*Microcontroller Unit*)

MFC Commande sans modèle (*Model Free Control*)

MIMO Entrées et sorties multiples (*Multiple-Input Multiple-Output*)

MRAC Commande adaptative à référence de modèle (*Model Reference Adaptive Control*)

NED Nord Est Bas (*North, East, Down*)

OI Injection de sortie (*Output Injection*)

PCB Circuit imprimé *Printed Circuit Board*

PID Proportionnel Intégral Dérivé

- PLA* Thermoplastique : acide polylactique
- PWM* Modulation de largeur d'impulsion (*Pulse Width Modulation*)
- SF* Retour d'état (*State Feedback*)
- SISO* Entrée et sortie unique (*Single-Input Single-Output*)
- SI* Injection d'état (*State injection*)
- SLA* Impression 3D stéréolithographique (*StereoLithography Apparatus*)
- SPI* Bus de données série synchrone *Serial Peripheral Interface*
- UAV* Drone autonome (*Unmanned Aerial Vehicle*)
- VTOL* Drones à décollage et atterrissage vertical (*Vertical Take-Off and landing*)

Introduction

Contexte

Ces dernières années, le domaine des drones s'est considérablement développé. En effet, de nombreux progrès ont été réalisés dans la conduite de vols autonomes, lesquels permettent de réaliser de nombreuses tâches longues, répétitives ou dangereuses, de manière plus sûre que des avions ou des systèmes télépilotés. Les drones ont fait leurs preuves dans de nombreuses applications civiles, alors qu'ils étaient auparavant conçus à des fins de surveillance et d'opération dans le secteur militaire. Tout leur intérêt réside dans leur capacité à se maintenir stabilisé sans intervention humaine. Ainsi, les opérateurs peuvent se concentrer sur la mission, sans devoir consacrer une grande attention au pilotage du drone.

La possibilité d'utiliser des systèmes de vols autonomes dans le secteur civil a été rendue possible par l'accessibilité croissante, proposée par l'industrie, de solutions à faible coût pour les applications d'imagerie aérienne. Ainsi, ce sont dans des domaines aussi variés que l'agriculture de précision, l'inspection des infrastructures civiles ou encore les opérations de sécurité que les drones autonomes sont aujourd'hui mobilisés, devenant alors un riche sujet de recherche.

La miniaturisation des équipements électroniques et mécaniques est à l'origine de l'essor d'une classe de drones de plus en plus petits. Souvent qualifiés de *Micro Air Vehicle* (MAV) ou de *Unmanned Aerial Vehicle* (UAV), leur petite taille leur permet d'intervenir dans des espaces confinés ou contraints. Ils n'ont, cependant, qu'une charge utile restreinte, souvent limitée à l'emport d'une caméra ou d'un colis de faible masse. Leur faible autonomie restreignant leur usage, la recherche s'est alors concentrée sur une solution permettant d'optimiser leur utilisation. En cela, les drones à décollage et atterrissage vertical (*Vertical take-off and landing*; VTOL) répondent aux exigences.

Dans l'ensemble des VTOL, plusieurs architectures existent et seront détaillées dans la section 1.1.2. Toutefois, nos travaux se sont concentrés sur les classes des *tailsitters* et des *freewings*.

Phases d'un vol

De nombreux travaux ont été menés sur les *tailsitters*, avec l'objectif de couvrir l'intégralité du domaine de vol. Ce dernier est constitué des phases de vol suivantes :

1. Décollage vertical
2. Transition entre le vol stationnaire et le vol d'avancement
3. Vol d'avancement
4. Transition entre le vol d'avancement et le vol stationnaire
5. Atterrissage vertical

Bien que l'on puisse observer une symétrie entre la phase ① et ⑤, qui correspondent au décollage et à l'atterrissage vertical, une différence fondamentale est notable. Lors du décollage, la vitesse du drone engendrera un flux d'air sur l'aile, orienté dans le même sens que le flux d'air généré par les hélices. Cependant, lors de l'atterrissage, le flux d'air va se trouver inversé, le drone devant descendre, ce qui engendrera une vitesse opposée à la direction du flux d'air des hélices. Cette inversion génère une instabilité qui doit être compensée par le contrôleur.

Le vecteur \vec{W} représente la perturbation de vent qui peut affecter le vol sur l'intégralité des cinq phases de vol. Toutefois, on observe que dans les phases de décollage ①, de transition ② et ④ et d'atterrissage ⑤, le drone offre une grande surface verticale sujette au vent. Ainsi, il est nécessaire de traiter l'impact du vent sur cette architecture.

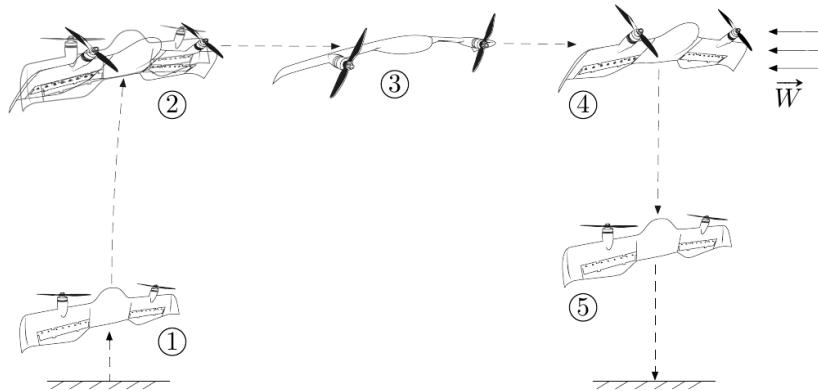


FIGURE I.1 – Phases de vol d'un drone *tailsitter*, DarkO.

Maquettes

De nombreuses maquettes réelles ont été assemblées dans le but de réaliser des vols expérimentaux. Citons en exemples le *tailsitter* à double rotor appelé « T-Wing » [Stone 2002, Stone 2008], le *tailsitter* appelé « MavIon » [Lustosa 2015], ou le « JLion » et le « KH-Lion » [Ke 2017b]. Ces trois maquettes sont illustrées sur la Figure I.2.



FIGURE I.2 – Maquettes « T-Wing », « MavIon » et « KH-Lion ».

Ces drones partagent une architecture similaire basée sur une aile supportant deux moteurs sur le bord d'attaque et soufflant deux elevons situés sur le bord de fuite. Cette architecture offre une plus grande robustesse que les *tiltrotors*, composés de pièces mobiles (ce qui les rend plus fragiles) et d'un actionneur puissant pour faire tourner l'ensemble moteur-hélice. La complexité inhérente à ces architectures nécessite un travail de modélisation en raison des nombreuses non-linéarités et couplages impliqués, en particulier en termes de modélisation des effets aérodynamiques. Dans ce contexte, l'interférence aérodynamique entre l'aile fixe et les rotors a été modélisée dans [Droandi 2015, Simmons 2022, Aref 2018], et les forces et moments d'hélice générés à des angles d'attaque élevés sont abordés dans [Fernandez 2023a]. Cependant, ces modèles sont complexes algorithmiquement et ne sont que partiellement utilisables pour la conception des commandes.

Un autre point important est la représentation de l'attitude du drone. Aussi, il est possible de représenter son orientation par des angles d'Euler [Escareno 2006, Guerrero 2009, Zhang 2017], ce qui permet une compréhension intuitive. Toutefois, une singularité apparaît dans certaines phases de vol. Compte tenu de la grande manœuvrabilité, il est préférable de représenter l'attitude par un quaternion unitaire, ce qui élimine toute singularité [Ke 2017a]. De nombreuses publications modélisent les effets aérodynamiques générés par les hélices en fonction de l'angle d'attaque et l'angle de dérapage [Escareno 2007, Chiappinelli 2018]. Il est possible de choisir un autre modèle pour les interactions aérodynamiques entre les moteurs, les ailes et les elevons, comme présenté dans [Lustosa 2019]. La technique de modélisation présentée dans [Lustosa 2019] permet de disposer d'un modèle global couvrant l'ensemble de l'enveloppe de vol, grâce à ce que l'on appelle l'approche Φ -théorie. Bien que cette dernière ne permette pas de prédire la chute brutale de la force de portance avec un angle d'attaque croissant (qui est causée par un flux d'air turbulent) [Tal 2022], elle permet de représenter le drone avec suffisamment de précision pour capturer le comportement lors de manœuvres agressives.

Les deux architectures sur lesquelles se sont concentrées nos recherches sont celles de DarkO, un *tailsitter* et de Colibri, un *frewing* basé sur une aile inspirée de DarkO, en rotation libre autour d'un fuselage qui sera maintenu horizontal.

Question de recherche

Comme expliqué précédemment, lors des phases de décollage et d'atterrissement, le drone est vertical (voir figure I.1), ce qui engendre une grande sensibilité aux perturbations. Il semble pertinent de concentrer nos travaux sur l'étude de la robustesse des drones convertibles face au vent. Notre étude se focalise sur la recherche d'un contrôleur de vol unifié, pour une architecture de drone fortement non-linéaire et couplée, sur l'intégralité du domaine de vol, en environnement perturbé.

Objectifs fixés pour la thèse

Les objectifs fixés sont pluriels : il s'agira d'étudier le comportement d'un drone *tailsitter* en environnement perturbé en présence de saturation des actionneurs pouvant engendrer des cycles limites. Nous utiliserons la linéarisation pour extraire la dynamique du drone autour de l'ensemble des points d'équilibre. Nous étudierons la précision des linéarisations face aux nombreuses non-linéarités du modèle.

De nos linéarisations, de notre compréhension du fonctionnement du drone et des limites analysées, il s'agira de proposer des architectures de commande basées sur un modèle pour un drone *tailsitter* permettant d'assurer une désensibilisation aux perturbations de vent. L'intérêt d'une architecture basée sur un modèle est la possibilité de certification de ce type d'architecture.

Pour finir, nous souhaitons utiliser des capteurs pour mesurer les perturbations en avance de phase pour les rejeter (sonde 5 trous, micro, Pitot, etc.). La connaissance d'une perturbation en amont de son impact sur le drone peut être un point clé dans la diminution de son impact. La mesure associée à un modèle peut être un moyen d'agir en anticipation plutôt qu'en réaction. Une action en anticipation pourrait se traduire par une modification de l'état du drone avant l'arrivée de la perturbation pour en minimiser son impact. À l'inverse, une action en réaction est une gestion de la perturbation suite à une modification de l'équilibre du drone (déplacement ou modification de son orientation) qui intervient après que la perturbation ait impacté le drone.

Au vu des contraintes des *tailsitters*, l'installation de capteurs de mesure de vent implique le développement d'une nouvelle architecture permettant l'installa-

tion d'un capteur capable de mesurer les perturbations sur l'intégralité du domaine de vol.

Plan et contribution

Notre exposé commencera par une description générale des architectures de drone convertible (Chapitre 1), avec une présentation des avantages et des inconvénients, ainsi que leur mode de fonctionnement. Une description générale de la modélisation, de l'actionnement et des lois de commande proposées sur les *tailsitters* et les *freewings* introduira nos propos sur ces architectures.

Le chapitre 2 détaillera le modèle non-linéaire d'un drone *tailsitter*, DarkO, à partir des travaux de [Lustosa 2019] et de [Olszanecki Barth 2020]. Nous proposerons un modèle simplifié pour les basses vitesses, ainsi que le détail des équilibres stationnaires en présence ou non de vent. De ces équilibres, nous présenterons la dynamique linéarisée du drone paramétrée par deux scalaires : le vent horizontal et vertical. Ce modèle étant le point de départ de chacun de nos travaux, il se retrouve expliqué dans [Sansou 2022a, Chapitre 2] et [Sansou 2022c, Section II] dans la condition de vent nul, dans [Sansou 2022b, Section 2] avec des conditions de vent non nul et dans [Sansou 2024a, Section II] sous sa forme la plus complète.

Le chapitre 3 fera l'objet d'une proposition de loi de commande hybride permettant d'augmenter le domaine de stabilité d'une loi linéaire avec une loi de commande non-linéaire basée sur une direction de zéro moment. Ces travaux ont été publiés dans [Sansou 2022a] et [Sansou 2022c].

Le chapitre 4 permettra de décrire une maquette expérimentale utilisée pour évaluer les performances d'une loi de commande basée sur une architecture proportionnelle dérivative. Cette loi est un retour de sortie permettant de stabiliser une position stationnaire en présence de vent. Cette maquette restreint les degrés de liberté classiques d'un drone pour se concentrer sur la réjection de perturbations de vent, grâce à un changement d'incidence de la maquette. La description de la loi de commande, son optimisation et les résultats sont disponibles dans [Sansou 2022b].

Le chapitre 5 propose une méthode différente d'obtention des gains de la boucle fermée, basée sur les inégalités linéaires matricielles et la théorie de Lyapunov. L'ensemble de l'optimisation sera réalisé à l'aide de la dynamique linéarisée du drone autour de la condition de vent nul. Ce contrôleur sera testé sur une maquette à six degrés de liberté en environnement contrôlé. Cette seconde méthode d'obtention des gains du contrôleur est disponible dans [Curpanaru 2024].

Le chapitre 6 étendra le contrôleur proposé dans le chapitre 4 pour stabiliser

la dynamique complète d'un drone *tailsitter*. La méthode d'obtention des gains du contrôleur est basée sur un algorithme itératif permettant de maintenir la complexité algorithmique, tout en identifiant les conditions de vol critiques. Les travaux ont été publiés dans [Sansou 2024a].

Le chapitre 7 propose une nouvelle architecture de *freewing*. Cette dernière est inspirée d'un *tailsitter* sur lequel on adjoint un fuselage pendulaire en rotation libre de manière à le maintenir horizontal dans toutes les configurations de vol. Nos travaux se sont concentrés sur l'obtention d'un modèle de simulation basé sur une dynamique multicorps. Des résultats de simulation et expérimentaux ont été proposés, basés sur un contrôle non-linéaire et publiés dans [Sansou 2024b].

CHAPITRE 1

Généralités sur les drones convertibles

Sommaire

1.1	Micro-drones convertibles	7
1.1.1	Efficacité énergétique et domaine de vol	7
1.1.2	Types d'architectures des drones convertibles	8
1.2	Propriétés des <i>tailsitters</i> et des <i>freewings</i>	12
1.2.1	Actionnement	12
1.2.2	Aérodynamique	13
1.3	Modélisations	14
1.3.1	Dynamique du système	14
1.3.2	Modèle de perturbation	15
1.4	Actionnement	16
1.5	Architecture de commande de vol	17
1.6	Architecture de commande	17
1.7	Méthodes de résolution	22
1.8	Technologies et réalisations	25
1.9	Conclusion	27

1.1 Micro-drones convertibles

1.1.1 Efficacité énergétique et domaine de vol

Tout l'intérêt d'un drone convertible réside dans sa capacité à décoller et atterrir verticalement, tout en conservant une bonne efficacité énergétique en vol horizontal, grâce à une aile. Cette aile a l'avantage de générer de la portance, laquelle s'oppose au poids du drone et permet d'assurer sa sustentation. La contrepartie de la génération de cette portance est la traînée qui s'oppose à l'avancement et doit être contrée par une force de traction engendrée par les hélices.

L'efficacité énergétique dépend du temps de vol et de l'énergie électrique nécessaire pour effectuer ce vol. Afin de souligner la prééminence de l'efficacité énergétique de ce modèle convertible, il convient de la comparer avec celle d'un drone quadrirotor qui assure sa sustentation uniquement grâce à des hélices (à l'instar d'un hélicoptère), et à celle d'un drone à voilure fixe.

Architecture	Vitesse m s^{-1}	Stationnaire	Temps de vol	Consommation
Convertible	[0 - 30]	Possible	Quelques heures	Faible
Quadrirotor	[0 - 16]	Possible	Quelques minutes	Forte
Voilure fixe	[8 - 30]	Impossible	Plusieurs heures	Variable

TABLE 1.1 – Comparaison des architectures de drone [Saeed 2018, Table 1].

Nous observons sur la Table 1.1 qu'un drone convertible possède un domaine de vol bien plus important qu'un drone à voilure fixe, lequel ne sera pas en mesure de voler à très basse vitesse, et qu'un quadrirotor, dont l'autonomie va être limitée par sa consommation. Toutefois, les drones convertibles sont nettement moins efficaces qu'une voilure fixe car ils sont plus lourds en raison des mécanismes nécessaires à la conversion et des redondances indispensables pour lutter contre les pannes. Il faut aussi noter que les voitures fixes ont une consommation inférieure pour les mêmes raisons qu'évoquées précédemment. Les drones convertibles ont une consommation variable qui dépend de l'architecture (liée à la masse ajoutée) et du temps passé en vol stationnaire et en vol d'avancement. Cette consommation est quand même inférieure à celle d'un quadrirotor qui se maintient en vol grâce à la rotation des hélices. Les convertibles utilisent une aile, ce qui diminue la consommation énergétique en vol d'avancement. En alliant autonomie et vol stationnaire, le modèle convertible répond aux exigences des missions civiles et militaires.

1.1.2 Types d'architectures des drones convertibles

La conception structurelle et aérodynamique d'un drone convertible est le facteur principal permettant des transitions stables et fluides. De plus, il est nécessaire d'optimiser l'architecture pour une mission, de manière à être le plus efficace. Au vu de la diversité des tâches à réaliser, un grand nombre de modèles ont été proposés, lesquels sont généralement catégorisables en trois classes : *tiltrotor*, *tailsitter*, *tiltwing*. En ce qu'il s'agit des prémisses des convertibles, il paraît opportun d'ajouter aux trois grandes catégories précédemment citées dans les états de l'art [Saeed 2018, Ducard 2021, Misra 2022], la catégorie *quadplane*.

Quadplane

Les *quadplanes* résultent de la fusion d'un avion et d'un quadrirotor (comme visible sur la Figure 1.1), ce qui permet un découplage de l'actionnement en fonction de la phase de vol. Le premier système de propulsion est composé de quatre hélices générant une force verticale permettant le contrôle lors des phases de décollage, d'atterrissage et stationnaire. Le second système de propulsion est composé d'une hélice propulsive supplémentaire afin de maintenir le vol d'avancement.



FIGURE 1.1 – Structure *quadplane* proposée par [Avy 2023].

L'avantage de ce type d'architecture est sa grande robustesse. Effectivement, aucune pièce en mouvement n'est nécessaire pour réaliser la transition, ce qui réduit le risque de défaillance mécanique. Toutefois, l'inconvénient réside dans son manque d'efficacité énergétique. Lors d'un vol d'avancement, la portance sera générée par l'aile. Ainsi, il est possible de désactiver les rotors qui génèrent des perturbations aérodynamiques et des trainées parasites. Les axes des moteurs se retrouvent orthogonaux au flux d'air généré par le déplacement du drone, ce qui correspond au cas le plus défavorable en termes de trainée. De plus, la surcharge engendrée par l'emport de moteurs supplémentaires se traduit par une diminution de la charge utile transportable.

Pour ce qui est du contrôle, un atout indéniable est la séparation des actionnements en fonction des phases de vol. Ainsi, l'architecture de commande sera basée sur un mécanisme de commutation permettant de choisir la loi de commande appropriée sur un critère de vitesse air [Li 2021, Mathur 2021, Okulski 2022]. Ce critère est pertinent en ce qu'il est lié à la capacité de l'aile à générer de la portance induite par le flux d'air. Ainsi, à basse vitesse, le drone se stabilise avec l'actionnement quadrirotor et la loi de commande associée. Dans les vitesses plus importantes, la

commutation de la loi permet de contrôler le drone en mode avion. Toutefois, le passage d'une loi à l'autre reste le point clé de la commande et demeure complexe et critique.

Tiltrotor

Les *tiltrotors* nécessitent l'utilisation d'un actionneur dédié à la modification de l'orientation des moteurs. Les rotors sont montés sur des arbres basculants actionnés et la transition du vol stationnaire au vol d'avancement (ou inversement) s'effectue progressivement en fonction de l'inclinaison du rotor. Les deux configurations sont représentées sur la Figure 1.2. Ainsi, l'angle entre le souffle des hélices et l'aile peut être ajusté à chaque instant [Mancinelli 2022, Du 2024, Nie 2024, Schlatter 2024]. Cet angle joue un rôle important dans le contrôle des forces et des moments aérodynamiques : sa maîtrise permet, non seulement, de mieux gérer les performances aérodynamiques du vol lors des transitions, mais aussi la stabilité du système sur l'ensemble du domaine de vol. Malgré le fait que les *tiltrotors* embarquent un actionneur uniquement dédié à la transition, ce qui augmente la masse du drone, cette architecture est intéressante car elle permet d'utiliser les mêmes actionneurs pour assurer la sustentation en stationnaire et pour générer la traction en mode avion.

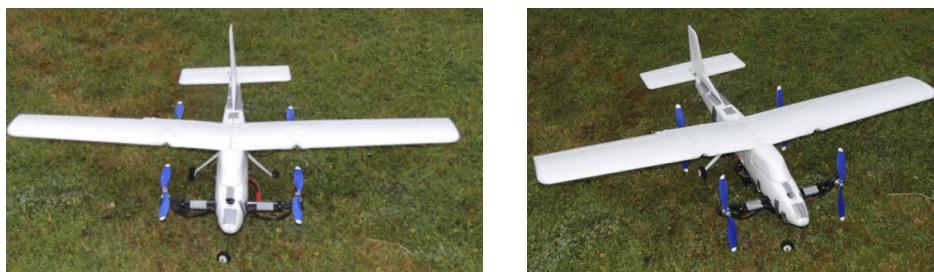


FIGURE 1.2 – Structure *Tiltrotor* proposée par [Flores 2014], dans deux configurations, vol stationnaire et d'avancement.

Tailsitter

Contrairement aux *tiltrotors* qui se posent sur le fuselage de l'avion (figure 1.2), les *tailsitters* se posent à la verticale (voir Figure centrale 1.3). Durant la transition du mode stationnaire au vol d'avancement, la structure entière bascule vers l'avant, modifiant ainsi l'angle d'incidence de la voilure [Ritz 2017, Verling 2016, Smeur 2020, Chiappinelli 2018, Tal 2022]. Selon la configuration du *tailsitter*, la transition peut être réalisée soit par la génération du moment aérodynamique créé par les élévons, soit par le couple créé par le système de propulsion. Pendant le vol

d'avancement, en position horizontale, le *tailsitter* vole comme un avion conventionnel sans dérive. En utilisant des techniques aérodynamiques classiques, les concepteurs peuvent optimiser le profil de l'aile du *tailsitter* pour le rendre plus endurant, afin de réduire la consommation d'énergie. Grâce à ce processus d'optimisation aérodynamique, le *tailsitter* peut effectuer des missions de vol de plus d'une heure.



FIGURE 1.3 – Structure *tiltrotor* proposée par [Smeur 2020, Fernandez 2023b, Flimlin 2006].

Ce modèle semble être la configuration la plus intéressante énergétiquement, car il utilise les mêmes actionneurs dans tout le domaine de vol. Ainsi, aucune masse superflue n'est embarquée.

Tiltwing

La particularité des *tiltwings* réside dans le fait que les rotors sont inclinés en même temps que les ailes (voir Figure 1.4).



FIGURE 1.4 – Structure *tiltwing* proposée par [Dufour 2024, Ostermann 2012].

Un actionneur supplémentaire et puissant est donc nécessaire pour surmonter le couple de l'aile, afin de la positionner dans l'orientation souhaitée [Holsten 2011, Rohr 2019, Çetinsoy 2012]. La commande de cet actionneur doit être prise en compte lors de la conception des lois de commande car ce servomoteur permet de fixer l'orientation de l'aile et par conséquent l'incidence vis-à-vis du flux d'air incident. Toutefois, la dynamique de cet actionneur et l'inertie de l'aile limitent les déplacements rapides, lesquels permettraient de rejeter des turbulences. Pendant le

décollage, l'atterrissage et les vols stationnaires, les ailes doivent être positionnées vers le haut afin de produire une force de poussée opposée au vecteur gravité. Dans ces phases de vol, lorsque les ailes sont orientées vers le haut, l'aéronef est plus vulnérable au vent et les lois de commande doivent rejeter ces perturbations. Dans la littérature, il existe plusieurs configurations d'ailes basculantes et différentes approches de contrôle conçues pour stabiliser leur dynamique de vol.

Par ailleurs, les *freewings* sont une gamme en cours de développement à l'intérieur de l'architecture des *tiltwings*. Ils sont actionnés comme des *tiltwings*, excepté au niveau de l'axe de rotation entre l'aile et le fuselage. Cette rotation est laissée libre, ce degré de liberté permettant à l'aile de changer librement son incidence. Il en résulte un gain de masse car il est possible de supprimer l'actionneur puissant et lourd nécessaire à la rotation de l'aile. De plus, l'aile étant libre de s'orienter, les turbulences ont un impact plus faible sur la structure, ce qui rend les vols plus stables [Welstead 2012, Axtén 2021, Axtén 2023].

1.2 Propriétés des *tailsitters* et des *freewings*

D'un point de vue mécanique, les *tailsitters* et les *freewings* sont caractérisés comme des systèmes sous-actionnés avec une dynamique fortement couplée. Ces caractéristiques mécaniques rendent le processus de modélisation et d'identification difficile. Cela peut s'expliquer par le fait que, pour ce type de système, une entrée de commande donnée agit simultanément sur différentes dynamiques. Ainsi, l'identification de l'influence d'une entrée de commande donnée sur une dynamique donnée nécessite une attention particulière.

1.2.1 Actionnement

Dans ces deux architectures, il est courant de trouver des actionneurs basés sur des effets aérodynamiques. Ces actionneurs ont l'avantage d'être peu consommateurs en énergie. Ils sont mis par des servomoteurs qui consomment peu d'électricité proportionnellement au couple qu'ils génèrent. Dans le cas des ailes volantes, les surfaces aérodynamiques sont souvent placées sur la partie arrière des ailes et peuvent être utilisées symétriquement à l'instar de volets, ou anti-symétriquement comme des ailerons. Dans les phases de décollage, de vol stationnaire ou d'atterrissage, la plateforme est maintenue en vol par les hélices : il est donc nécessaire de dimensionner les groupes moteurs-hélices pour qu'ils puissent générer assez de force. En fonction des configurations, les moments peuvent être obtenus par des

différentiels sur l'utilisation des moteurs ou bien par des surfaces aérodynamiques. Dans le cas de surfaces soufflées par le flux d'air des hélices, il existe un couplage des actionneurs qui complexifie la modélisation et le contrôle de ces architectures.

Définition 1.2.1 (Sous-actionnement). Considérons la dynamique d'un système décrite par l'équation différentielle du second ordre :

$$\ddot{\mathbf{q}} = f_1(\mathbf{q}, \dot{\mathbf{q}}, t) + f_2(\mathbf{q}, \dot{\mathbf{q}}, t)\mathbf{u}$$

où $\mathbf{q} \in \mathbb{R}^n$ est le vecteur d'état de dimension n , $\mathbf{u} \in \mathbb{R}^m$ est le vecteur de commande de dimension m et t le temps.

Lorsque le rang de f_2 est inférieur à la dimension du vecteur \mathbf{q} , on dit que le système est sous-actionné :

$$\text{rank}[f_2(\mathbf{q}, \dot{\mathbf{q}}, t)] < \dim[\mathbf{q}].$$

Au vu de la définition 1.2.1, beaucoup d'architectures présentées dans les catégories précédentes (*tiltrotor*, *tailsitter*, *tiltwing*, *quadplane*) sont sous-actionnées. La principale conséquence du sous-actionnement est l'impossibilité de générer un déplacement, sans modifier l'orientation du drone ou, autrement dit, tout déplacement sera engendré par un changement d'orientation du drone.

1.2.2 Aérodynamique

Lors d'un vol d'avancement, les *tailsitters* et les *freewings* assurent le maintien du vol en palier, en générant une force de portance grâce à leur surface alaire. Cette portance, qui s'oppose au poids, permet de maintenir une trajectoire. La force de trainée engendrée par l'aile et le fuselage est compensée par la composante horizontale de la poussée.

En vol stationnaire, le poids est compensé par la traction de l'hélice. La relation d'équilibre est plus complexe lorsque nous évaluons l'ensemble des points d'équilibre lors de la transition d'un mode à l'autre. La transition équilibrée est assurée par la composition correcte des forces aérodynamiques et de propulsion.

Les interactions aérodynamiques entre l'hélice et la surface de l'aile sont connues pour être complexes et difficiles à modéliser. La vitesse induite par le souffle de l'hélice entraîne une variation locale de l'angle d'attaque et des variations de pression dynamique au niveau des sections d'aile immergées, d'où une distribution différente de la portance (voir figure 1.5). L'analyse de l'interaction entre l'hélice, l'aile et les

élevons doit permettre d'obtenir de bonnes caractéristiques de vol et de tirer profit des combinaisons.

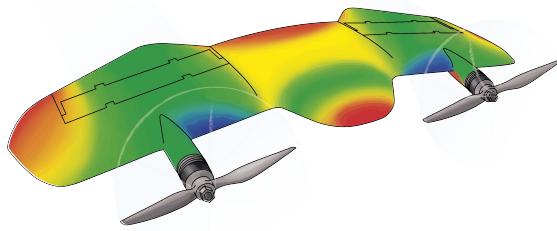


FIGURE 1.5 – Représentation de la pression dynamique sur la voilure de DarkO lors d'un vol d'avancement.

L'identification de ces effets aérodynamiques nécessite, pour chaque point de vol, les informations de l'air environnant, les valeurs des entrées de commande et les mesures des forces et moments aérodynamiques agissant sur le système. Le moyen le plus précis et le plus fiable d'identifier les phénomènes aérodynamiques est probablement de mener des campagnes en soufflerie, lesquelles sont particulièrement chronophages et coûteuses.

1.3 Modélisations

1.3.1 Dynamique du système

Dans le cas des drones convertibles basculant, le modèle aérodynamique doit être correct dans les phases de transition. Cela nécessite l'extension de l'aérodynamique classique (faible incidence) à un angle d'attaque élevé et à un fonctionnement à faible vitesse. En outre, l'effet du souffle de l'hélice sur les profils aérodynamiques du véhicule doit être compris et pris en compte dans le modèle du système. Par exemple, les travaux de [Rohr 2021] sur un *tiltwing* identifient les zones d'interaction entre l'hélice et l'aile. Les auteurs séparent clairement la génération de forces et de couples obtenues par la partie soufflée et non-soufflée de l'aile.

Des fonctions non-linéaires continues décrivant les coefficients de portance et de traînée aérodynamiques sur toute la plage de l'angle d'attaque pour les *tiltrotors* ont été dérivées dans [Ducard 2014] et pour les *tiltwings* dans [Lustosa 2017, Lustosa 2019].

Il existe de nombreuses conceptions possibles pour un drone convertible. Bien

que tous les modèles aient une structure commune, il existe des différences majeures dans la formulation des termes de forces et de couples. Celles-ci dépendent de la disposition des moteurs ou des hélices, de l'existence ou non de surfaces de contrôle aérodynamiques et de la forme du véhicule.

Un modèle précis est nécessaire pour les conceptions classiques de contrôle basées sur un modèle et en particulier pour les approches d'inversion dynamique ou de contrôle prédictif de modèle. Cependant, une modélisation précise va nécessiter une campagne d'identification approfondie en soufflerie ou un vol en environnement contraint. Il se peut également que la complexité du modèle ne permette pas une utilisation directe dans le contrôle, à cause d'une limitation matérielle des calculateurs embarqués, notamment dus aux temps de calcul réduits nécessaires à la stabilisation des drones.

1.3.2 Modèle de perturbation

Dans notre cas, nous nous sommes intéressés à l'impact du vent sur les architectures mentionnées précédemment. Il est donc nécessaire de modéliser le vent. Dans le cas de vent constant ou de cisaillement de vent, un modèle à échelons semble tout à fait approprié pour représenter le changement brusque de vitesse de vent. Pour les rafales, nous utiliserons plusieurs modèles, tel que le modèle "chapeau mexicain" ou les "ondelettes de Morlet".

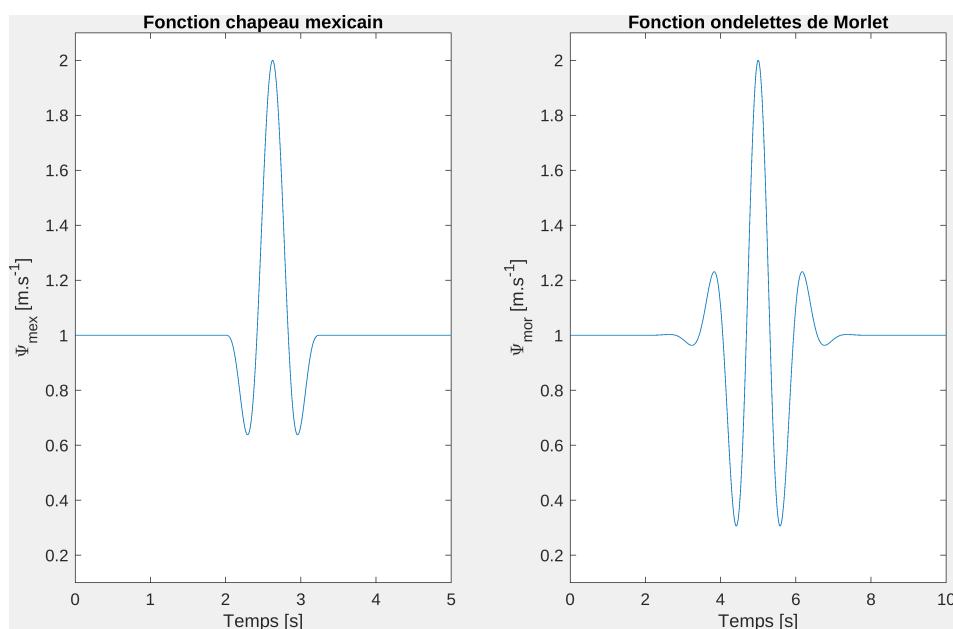


FIGURE 1.6 – Représentation temporelle des modèles de perturbation de vent.

La fonction définissant le "chapeau mexicain" est :

$$\Psi_{mex}(t) = w_m - \frac{A_g}{2} (1 - \cos(2\pi f_g(t - t_{0,mex}))) \sin(3\pi f_g(t - t_{0,mex})) \quad (1.1)$$

et la fonction "ondelettes de Morlet" est définie par :

$$\Psi_{mor}(t) = w_m + A_g \exp(-(t + t_{0,mor})) \cos(5 * (t - t_{0,mor})) \quad (1.2)$$

où w_m est le vent moyen sans perturbation, $t_{0,mex}$ représente l'instant de début de perturbation et $t_{0,mor}$ l'instant où la perturbation est maximale, A_g est l'intensité maximale de la perturbation et f_g est la fréquence de la perturbation. Les tracés de la Figure 1.6 montrent la représentation temporelle des deux fonctions pour les valeurs $w_m = 1 \text{ m s}^{-1}$, $t_{0,mex} = 2 \text{ s}$, $t_{0,mor} = 5 \text{ s}$, $A_g = 1 \text{ m s}^{-1}$ et $f_g = 0.8 \text{ rad s}^{-1}$.

1.4 Actionnement

La Figure 1.7 illustre les effets des actionneurs sur l'attitude d'un *tailsitter*.

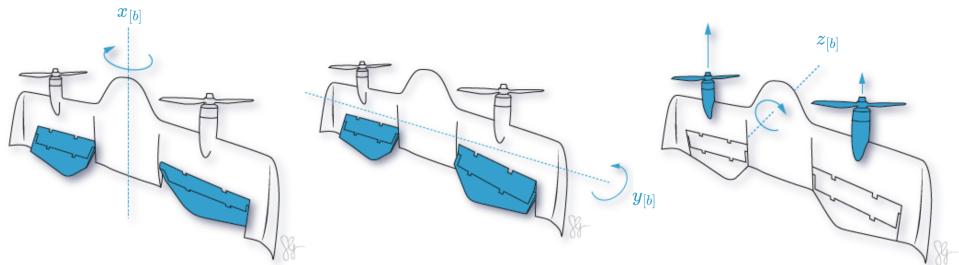


FIGURE 1.7 – Actionneurs d'un *tailsitter* et leurs actions sur les axes principaux.

La rotation autour de l'axe x_b peut être contrôlée par des braquages de volets asymétriques, l'angle de tangage (axe y_b) par des braquages de volets symétriques et la rotation autour de l'axe z_b est contrôlée par un dispositif de poussée différentielle, ce qui permet d'obtenir un angle de tangage plus élevé. La rotation autour de l'axe de lacet est contrôlée par une configuration de poussée différentielle, qui crée un couple temporaire non nul autour de son axe. Le dispositif de poussée différentielle engendre une différence entre les rotations de l'hélice gauche et de l'hélice droite, et modifie ainsi la vitesse de rotation de l'avion et donc le comportement aérodynamique autour des volets.

1.5 Architecture de commande de vol

Une approche en cascade est une méthode couramment utilisée dans le cadre de la commande de drone. Cette approche, illustrée sur la Figure 1.8, est basée sur l’imbrication de boucles de commande.

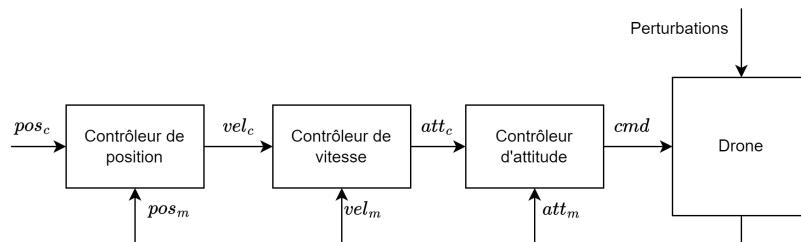


FIGURE 1.8 – Architecture classique de contrôle hiérarchique pour les drones.

Cette architecture s’appuie sur la séparation des dynamiques d’un drone sous actionné. Effectivement, le drone étant obligé de modifier son orientation pour changer de position, le schéma de contrôle prend en compte cette spécificité. Ainsi, la boucle interne est réglée pour avoir un temps de réponse faible vis-à-vis des autres lois. Cette boucle interne stabilise l’attitude du drone (att_m) en la faisant converger vers l’attitude de consigne (att_c). Cette attitude de consigne est le résultat de la boucle de vitesse faisant converger la vitesse du drone (vel_m) vers la vitesse de consigne (vel_c). La vitesse de consigne est la résultante de la boucle de position faisant converger la position du drone (pos_m) vers la position de consigne définie par le pilote (pos_c).

1.6 Architecture de commande

Plusieurs méthodes de commande ont été proposées sur les drones convertibles. Nous pouvons notamment distinguer les architectures linéaires et non-linéaires. La figure 1.9 propose une sélection des principales lois disponibles dans la littérature.

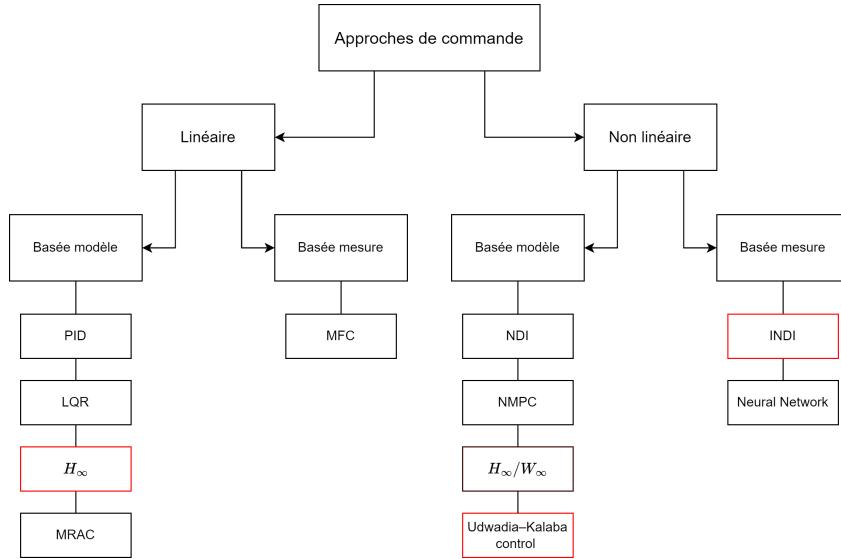


FIGURE 1.9 – Méthodes de commande utilisées sur les architectures *tailsitter* et *freewing*.

Contrôleur PID

Le contrôle Proportionnel Intégral Dérivé (PID) (ou P/PD/PI) est couramment utilisé, car son réglage est intuitif et ne nécessite qu'une connaissance limitée du système. Le contrôle PID donne souvent de très bons résultats et constitue un excellent point de départ pour la conception de contrôleurs plus avancés.

La forme générale d'un contrôleur PID est la suivante :

$$u_{PID}(t) = K_p e(t) + K_i \int_0^t e(\tau) d\tau + K_d \frac{de(t)}{dt} \quad (1.3)$$

où le vecteur d'erreur est $e(t) = x_{ref}(t) - x(t)$ et où les gains K_p , K_i et K_d sont respectivement le gain proportionnel, intégral et dérivateur. Ils peuvent être réglés ou calculés individuellement pour obtenir le comportement souhaité du système en boucle fermée.

Contrôleur LQR

Les contrôleurs LQR sont un type de contrôle linéaire basé sur les systèmes linéarisés de la forme $\dot{x} = Ax + Bu$. La commande u est choisie comme :

$$u = -Kx.$$

où la matrice de gain K est optimisée pour obtenir de bonnes performances pour le système en boucle fermée par rapport à la fonction de coût suivante :

$$J = \int_0^{\infty} x^T(\tau) Q x(\tau) + u^T(\tau) R u(\tau) d\tau$$

où Q et R sont, respectivement, les matrices de pondération de l'état et des commandes.

Bien que le contrôleur LQR présente généralement de bonnes propriétés de robustesse, l'optimalité n'est plus assurée si des erreurs de modélisation et des perturbations sont présentes dans le système. Une discussion détaillée et une comparaison entre le contrôle PID sans modèle et le contrôle LQR basé sur le modèle d'un *tailsitter* est disponible dans [Barth 2018]. Les auteurs de [Lustosa 2017] ont proposé l'approche de séquencement des gains obtenus pour un ensemble de modèles linéaires. Cette architecture de contrôle optimise le gain K en boucle fermée afin de répondre aux exigences de contrôle de la vitesse et de l'attitude définies par l'utilisateur. Grâce à des simulations de vol et à des vols expérimentaux, les auteurs soulignent et prouvent qu'une seule matrice de gains LQR n'est pas suffisante pour stabiliser le *tailsitter* dans toute son enveloppe de vol, ce qui justifie l'utilisation de méthodes de séquencement des gains.

Contrôleur non-linéaire

L'approche "inversion dynamique non-linéaire incrémentale" (INDI) utilise la connaissance de la dynamique d'un système. Une inversion incrémentale est réalisée pour obtenir les accélérations angulaires cibles. Théoriquement, le bouclage des accélérations angulaires élimine la sensibilité aux imperfections du modèle, ce qui augmente considérablement la robustesse du système par rapport à l'inversion dynamique non-linéaire conventionnelle [Sieberling 2010, Binz 2019]. Ce contrôleur nécessite l'identification des actionneurs afin de régler les paramètres du contrôleur, telle que l'efficacité des actionneurs. Étant donné que l'efficacité des surfaces de contrôle aérodynamiques n'est pas constante pendant toute l'enveloppe de vol sur un *tiltwing*, une méthode de programmation des gains a été mise au point [Smeur 2016, Smeur 2020]. La limitation d'une telle loi de commande est la disponibilité et la qualité des accélérations angulaires. En effet, les capteurs mesurant la vitesse angulaire sont très bruités et nous sommes contraints de dériver ce signal pour obtenir les accélérations angulaires.

Contrôleur sans modèle basé mesure

Contrairement à l'INDI basée sur l'inversion d'un modèle, la loi de commande sans modèle (MFC) [Fliess 2013] nécessite peu de connaissances préalables du système contrôlé pour concevoir le contrôleur. On modélise la dynamique inconnue par une équation numérique, appelée modèle ultra-local :

$$y_m^{(\nu)} = F_y + \lambda u. \quad (1.4)$$

F_y représente la dynamique réelle du modèle ainsi que les différentes perturbations qui pourraient affecter la performance du système en sortie. Une estimation précise de F_y , définie comme \hat{F}_y , est cruciale et joue un rôle important dans les performances de contrôle de la MFC et est obtenue à l'aide d'un estimateur non-linéaire. Soulignons que cet estimateur ne modifie pas les gains de la commande en retour, contrairement aux méthodes de commande adaptative. Le processus adaptatif MFC est réalisé par l'estimateur qui fournit une sortie, laquelle est utilisée pour calculer la commande formant la boucle fermée. Avec cet estimateur, il est possible de mettre en œuvre un contrôleur robuste et adaptatif qui assure la stabilité des systèmes variables dans le temps, en estimant en temps réel leur dynamique à partir des mesures périodiques. Nous pouvons citer le travail de [Olszanecki Barth 2020] sur un *tailsitter*, lequel continue de voler malgré une modification importante de son actionnement et de sa dynamique (voir Figure 1.10).



FIGURE 1.10 – *Tailsitter* DarkO lors d'un test de robustesse [Olszanecki Barth 2020].

Contrôleur séquencé

L'approche "Diviser et Conquérir" de la Figure 1.11 (a) permet de sélectionner la loi de contrôle appropriée pour chaque phase de vol, de manière discrète, de sorte qu'un seul contrôleur ne fonctionne à la fois. Un exemple d'un contrôle séquentié hybride de *tailsitter* est donnée dans [Casau 2011].

La pondération de contrôle (Figure 1.11 (b)) fusionne en continu les commandes

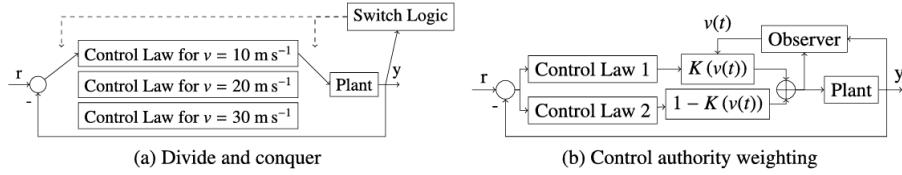


FIGURE 1.11 – Contrôleur séquencé discret (a) et continu (b) [Ducard 2021].

de deux contrôleurs, sur la base d'un poids dépendant d'une variable de sélection. Les auteurs de [Liang 2016] ont développé deux contrôleurs : un pour le mode VTOL et un second pour la croisière, lesquels fonctionnent simultanément. La commande globale est calculée comme :

$$u(\Delta t) = (1 - \frac{\Delta t}{T_T})u_{VTOL}(\Delta t) + \frac{\Delta t}{T_T}u_{cruise}(\Delta t).$$

Cet exemple est une transition entre le mode vol stationnaire et le mode croisière. Dans ce cas, T_T est le paramètre de durée de la transition et Δt est le temps écoulé depuis le début de la manœuvre.

Contrôleur adaptatif

La commande adaptative à référence de modèle (MRAC) utilise un modèle de référence avec les performances de suivi souhaitées et adapte les paramètres du contrôleur en fonction de la différence entre le modèle réel y et le modèle de référence y_m (voir Figure 1.12).

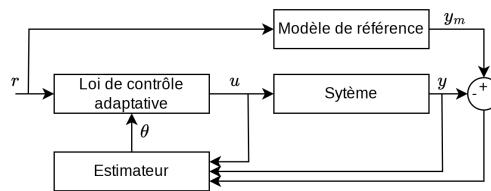


FIGURE 1.12 – Architecture d'un contrôleur MRAC.

Le mécanisme d'ajustement peut être développé à partir de la règle de MIT [Jain 2013], de la théorie de Lyapunov [Ge 1999, Chakrabarty 2016] et d'autres approches mathématiques, telles que les fonctions d'ajustement de la covariance, etc.

La technique de mécanisme d'ajustement proposée par [Chakrabarty 2016], ba-

sée sur les fonctions de Lyapunov, vise à suivre la sortie du système et ses états avec le modèle de référence. Pour ce faire, on suppose que tous les états sont disponibles pour la mesure, ce qui est rarement le cas dans les applications pratiques de contrôle.

Selon la règle du MIT [Shekhar 2018], le mécanisme d'ajustement fonctionne en minimisant l'erreur entre la sortie du système contrôlé y et la sortie du modèle de référence y_m via une fonction de coût qui met à jour le gain du contrôleur. Ces paramètres sont modifiés dans le sens négatif du gradient de la fonction de coût. Notons que le modèle de référence joue un rôle important dans les performances du contrôleur car la règle MIT tente de faire correspondre la sortie du système contrôlé avec la sortie du modèle de référence. Si les dynamiques du système sont mal modélisées, les performances de contrôle seront directement impactées. En outre, la règle MIT est très sensible aux changements d'amplitude de l'entrée de référence.

Pour l'estimation des paramètres en temps réel, les méthodes les plus couramment utilisées sont les dérivés de la méthode des moindres carrés moyens, telles que la méthode des moindres carrés moyens normalisés, la méthode des moindres carrés moyens récursifs et le filtre de Kalman étendu.

1.7 Méthodes de résolution

Il existe plusieurs méthodes pour obtenir les gains des contrôleurs évoqués précédemment (section 1.6) des plus naïves avec un réglage des gains d'un PID par essais erreur et jusqu'aux méthodes complexes d'optimisation (optimisation non lisse pour synthèse multiobjectif). Nous pouvons ajouter la résolution de l'équation algébrique de Riccati pour obtenir la matrice de gain d'un contrôleur LQR (section 1.6). Nous nous attarderons ici sur deux méthodes : le réglage des gains d'un PID par une méthode de placement de pôles et une synthèse basée sur des critères H_∞ .

Obtention des gains d'un PID par une méthode de placement de pôles [Wang 2008]

Les contrôleurs PID sont très largement utilisés dans l'industrie. Le réglage des gains (proportionnel, intégral et dérivatif) est souvent réalisé à l'aide d'une méthode manuelle des valeurs. Toutefois, à l'aide d'un modèle, il est possible d'obtenir les gains qui assurent des critères de performance (rapidité et dépassement). Cette méthode a été proposée par [Wang 2008].

Considérons un contrôleur PID de la forme :

$$C(s) = K_p + \frac{K_i}{s} + K_d s \quad (1.5)$$

utilisé pour contrôler un système $G(s)$ avec un retour unitaire d'équation caractéristique de bouclage :

$$1 + C(s)G(s) = 0. \quad (1.6)$$

Nous proposons de convertir les exigences de performance en une paire de pôles conjugués $\rho_{1,2} = -a \pm bj$.

La conception de la commande exige que le rapport entre a et la partie réelle de tous les pôles du système dépasse r , qui est généralement supérieur à 3. Cette contrainte impose l'emplacement de tous les autres pôles du système sur le côté gauche de la ligne $s = ra$.

En substituant $\rho_1 = -a + bj$ dans (1.6), nous obtenons :

$$K_p + \frac{K_i}{-a + bj} + K_d(-a + bj) = -\frac{1}{G(\rho_1)}. \quad (1.7)$$

Cette équation complexe peut être résolue en exprimant K_i et K_d en fonction de K_p :

$$\begin{cases} K_i = \frac{a^2 + b^2}{2a} K_p - (a^2 + b^2) X_1, \\ K_d = \frac{1}{2a} K_p + X_2 \end{cases} \quad (1.8)$$

avec

$$X_1 = \frac{1}{2b} \operatorname{Im}\left(\frac{-1}{G(\rho_1)}\right) + \frac{1}{2a} \operatorname{Re}\left(\frac{-1}{G(\rho_1)}\right), \quad X_2 = \frac{1}{2b} \operatorname{Im}\left(\frac{-1}{G(\rho_1)}\right) - \frac{1}{2a} \operatorname{Re}\left(\frac{-1}{G(\rho_1)}\right).$$

L'équation (1.8) simplifie la conception du contrôleur PID dépendant d'un paramètre K_p . Par conséquent, le gain K_p peut être déterminé pour ses valeurs positives à partir de la méthode du lieu racine. Selon le tracé des racines de l'équation caractéristique de la boucle fermée, nous sommes en mesure de déterminer la plage de K_p , de sorte que les racines se trouvent dans la région complexe requise.

Synthèse H_∞

La synthèse H_∞ est une méthode qui sert à la conception de commandes optimales, en imposant des contraintes sur la norme H_∞ d'un système [Apkarian 2006, Apkarian 2013]. En se basant sur une synthèse H_∞ , les auteurs de [Sun 2009] ont obtenu la stabilisation longitudinale d'un *tiltrotor*. En combinant une synthèse H_∞ à une approche de séquencement des gains, les auteurs de [Dickeson 2005, Dickeson 2006, Dickeson 2007] ont proposé la stabilisation d'un *tiltwing* sur l'ensemble de son enveloppe de vol.

Pour résoudre un problème par synthèse H_∞ , il est nécessaire de mettre le modèle du système dans la structure classique, représentée sur la Figure 1.13.

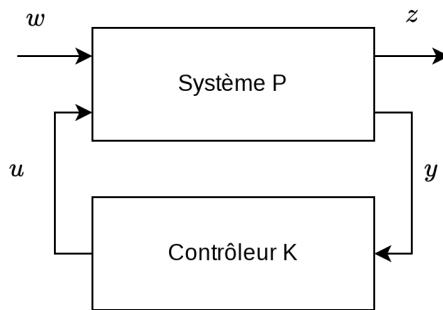


FIGURE 1.13 – Architecture d'un contrôleur H_∞ .

L'objectif est de minimiser l'effet de la perturbation w (cas le moins favorable possible) sur la variable de performance z , par le biais d'un contrôleur $u = K(s)y$. En d'autres termes, $K(s)$ est choisi de telle sorte que $\|F(P(s), K(s))\|_\infty$ soit minimisée, où $F(P(s), K(s))$ est la fonction de transfert de la perturbation w vers le signal d'erreur z . Dans ce formalisme, le contrôleur peut être de plusieurs formes PID, retour d'état ou de sortie, etc.

Nous pouvons aussi citer le travail de [Snyder 2021] qui utilise une approche de contrôle H_∞ , ainsi qu'une représentation du système *Linear Parameter Varying* (LPV). Cette représentation permet d'avoir la dynamique linéaire du drone sur l'ensemble du domaine de vol par interpolation des dynamiques paramétrées par la vitesse totale du drone.

Enfin, il existe une extension non-linéaire de l'approche H_∞ en utilisant l'espace de Sobolev pondéré W_∞ [Cardoso 2018, Cardoso 2019, Cardoso 2021, Cardoso 2024]. Cette méthode s'appuie sur la norme de Sobolev $W_{m,p}$ d'un signal,

laquelle est définie par :

$$\|z(t)\|_{W_{m,p}} = \left(\int_{t_0}^{\infty} (\|z(t)\|^p + \left\| \frac{dz(t)}{dt} \right\|^p + \dots + \left\| \frac{d^m z(t)}{dt^m} \right\|^p) dt \right)^{\frac{1}{p}}.$$

Grâce à la nature de la norme $W_{m,p}$, la variable de coût et ses dérivées temporelles sont prises en compte dans la fonction de coût, ce qui permet d'obtenir des contrôleurs offrant de meilleures performances en régime transitoire et en régime permanent.

1.8 Technologies et réalisations

Bien que les modèles proposés soient de plus en plus précis (section 1.3), que les simulations soient de plus en plus réalistes et que par conséquent, nous soyons en mesure de proposer des lois de commande de plus en plus élaborées (section 1.6 et 1.7), tout ce travail n'a pour validation finale que l'expérimentation. Ce travail nécessite le montage d'une plateforme comprenant plusieurs composants, l'identification des coefficients des modèles, le transfert des lois de commande dans un autopilote et la mise en place de vols expérimentaux.

Nos expérimentations sont réalisées grâce au système de drone Paparazzi (voir Annexe), lequel est constitué d'une partie matérielle et logicielle. La partie matérielle est principalement composée de l'autopilote (Apogee ou Tawaki) et la partie logicielle comporte l'ensemble des codes, embarqués et sol, permettant de faire voler un drone.

La Figure 1.14 présente de manière schématique les principaux composants nécessaires à la conduite d'un vol, tous étant reliés à l'autopilote. L'autopilote se charge d'exécuter périodiquement les codes associés aux différentes fonctions du drone (estimation d'état, navigation, guidage, contrôle (figure 1.8), charge utile, etc.).

Nos principales contributions dans cet environnement sont le codage des lois de commande proposées dans les chapitres 3 et 6 et le codage de l'estimation d'état du chapitre 7.

Un travail supplémentaire a été réalisé sur les contrôleurs moteurs. Les ESC ont été flashés avec le code open-source AM32 (voir Annexe). L'avantage de ce *firmware*, par rapport au code commercial, est qu'il exploite un retour PID de bas niveau de la vitesse de rotation du moteur, calculé à la même vitesse que la commutation de phase du moteur. Nous avons adapté le code de la boucle de vitesse dans le *firmware*, en suivant l'approche de [Franchi 2017], avec un algorithme de biais et de

gain adaptatif (ABAG).

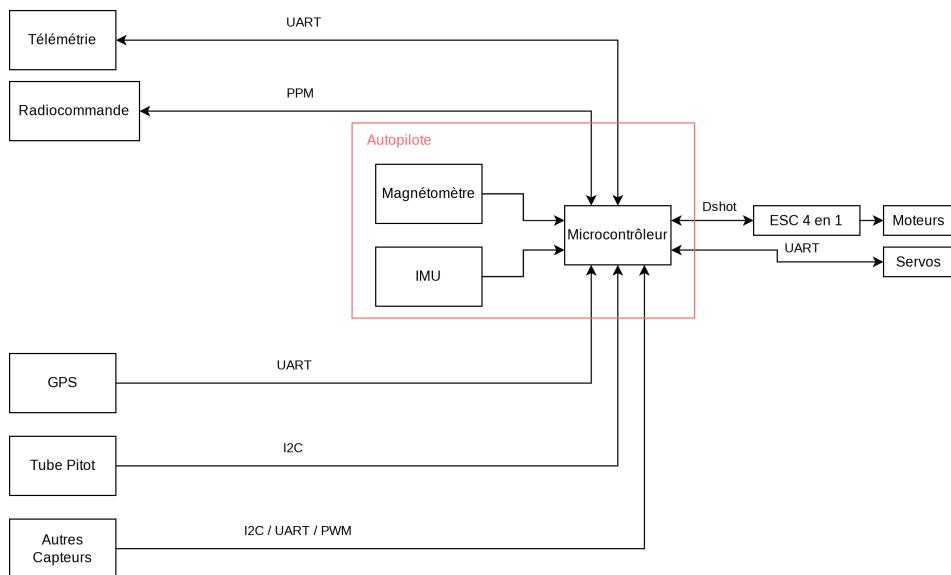


FIGURE 1.14 – Architecture d'un autopilote et son environnement.

Les avantages de la solution ABAG sont une grande réactivité et une grande adaptabilité, puisque les dimensions de l'hélice peuvent être modifiées sans qu'il ne soit nécessaire de modifier les gains d'actionnement. Un module a aussi été développé pour pouvoir utiliser des servomoteurs *Feetech* (le modèle STS3032), qui ont la possibilité d'être chaînés et d'être commandés via une liaison série bidirectionnelle. Cela permet notamment d'avoir un retour sur la position du servomoteur.

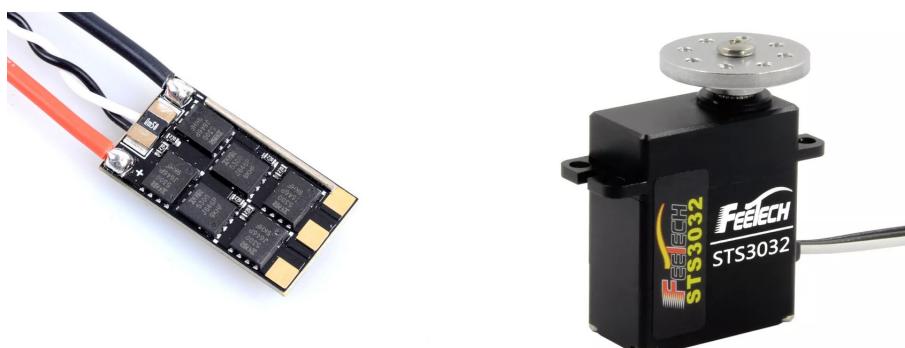


FIGURE 1.15 – Contrôleur moteur et servomoteur.

1.9 Conclusion

Ce chapitre présente un résumé de la littérature relative à la modélisation et au contrôle des *tailsitters*.

Le domaine des drones convertibles est vaste et chaque architecture cherche à répondre à une mission. Les diverses complexités engendrées sont autant de challenges pour assurer la stabilité du drone. Les défis aérodynamiques liés aux différents domaines de vol, tels que les écoulements à faible nombre de Reynolds, l'interaction hélice-ailes et l'identification des coefficients aérodynamiques par des campagnes en soufflerie, ont été résumés.

Plusieurs publications expliquent la complexité de développer un modèle dynamique pour les *tailsitters*. Cette difficulté notamment liée à la dynamique instable dans les vols de transition due à des changements rapides des forces et des moments aérodynamiques, appelle au développement de stratégies de contrôle innovantes et robustes.

L'analyse de la littérature sur les lois de commande indique que la mise en œuvre d'une loi robuste abordant directement l'impact du vent sur le drone n'a jamais été réalisée.

Nous allons donc nous focaliser sur un drone *tailsitter* et plus précisément sur sa modélisation dans le but de comprendre les effets des perturbations sur sa dynamique.

CHAPITRE 2

Modélisation d'un drone convertible : DarkO

Sommaire

2.1 Modèle du drone DarkO	29
2.1.1 Modèle non-linéaire complet	32
2.1.2 Modèle non-linéaire simplifié à basse vitesse	34
2.2 Caractérisation de la maquette	36
2.2.1 Identification des paramètres du modèle	36
2.2.2 Modélisation des actionneurs	39
2.3 Équilibres stationnaires	40
2.3.1 Équilibre stationnaire sans vent	40
2.3.2 Équilibre stationnaire en présence de vent	41
2.4 Dynamiques linéarisées	47
2.4.1 Dynamique linéarisée sans vent	47
2.4.2 Dynamique linéarisée en présence de vent	48
2.5 Conclusion	54

2.1 Modèle du drone DarkO

DarkO, drone conçu et développé à l'École Nationale de l'Aviation Civile (ENAC) de Toulouse (France), est un exemple de drone convertible avec une architecture dite *tailsitter*. DarkO est assemblé à partir de plusieurs pièces d'Onyx imprimées en 3D (un matériau très robuste composé de fibres de carbone omnidirectionnelles). Toutes les pièces sont emboîtées sur un seul axe, de sorte que le drone puisse facilement être démonté pour remplacer des pièces ou accéder à l'électronique embarquée.

L'autopilote embarqué est une carte Apogee¹ fabriquée à l'ENAC (voir Figure 2.1).

1. <https://wiki.paparazziuav.org/wiki/Apogee/v1.00>

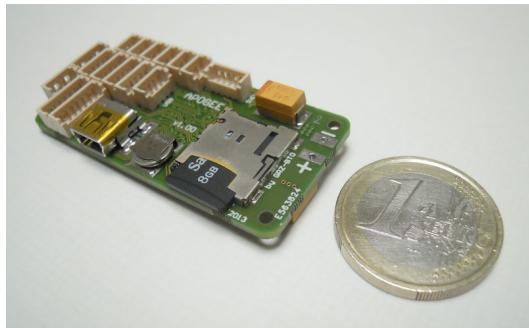


FIGURE 2.1 – Vue de dessus d'un autopilote Apogee v1.00.

L'autopilote offre la possibilité d'enregistrer les données de bord sur une carte mémoire SD, à la fréquence de contrôle de 500 Hz, ce qui permet un post-traitement efficace des données acquises. Le protocole de communication utilisé entre l'autopilote et les contrôleurs électroniques de vitesse (ESC) est le Dshot 600. Les ESC sont des AIKON AK32 35A avec un *firmware* AM32. La communication sol-bord est réalisée via un canal bidirectionnel basé sur des modules XBee-PRO S1.

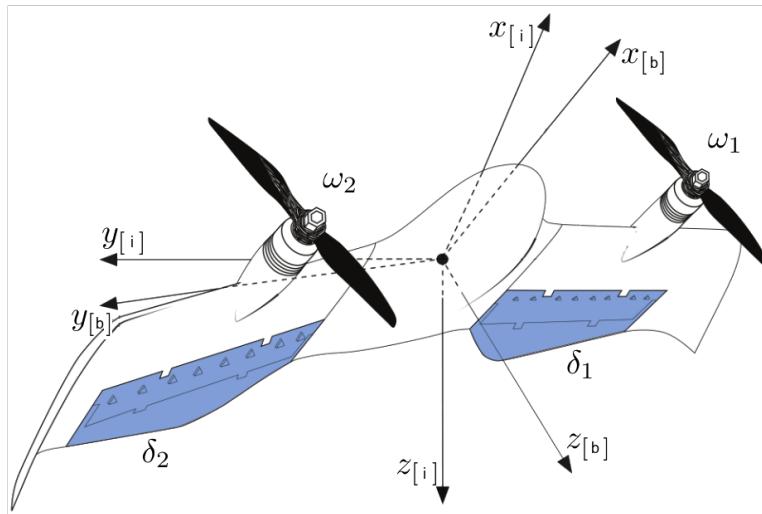


FIGURE 2.2 – Repère de référence de DarkO avec une représentation schématique des actionneurs.

Les actionneurs de DarkO sont classifiables en deux catégories. La première est liée à la propulsion et est constituée de deux hélices (T-Motor T5147) placées symétriquement à l'avant de l'aile (illustrées en **noir** dans la Figure 2.2) et alimentées par deux moteurs électriques (T-Motor F30 2300kv) générant une traction selon

l'axe x_b . La seconde catégorie est relative aux actionneurs aérodynamiques. Ainsi, le drone possède deux élévons, placés à l'arrière de l'aile (illustrés en bleu dans la Figure 2.2), agissant en tant que surface de contrôle. Les élévons génèrent des forces et des moments en modifiant leur incidence relativement au flux d'air dans lequel ils sont placés. Ce flux d'air peut être dû au vent relatif (lié à la vitesse du drone), au vent extérieur, mais aussi au souffle des hélices. Les élévons sont commandés par deux servomoteurs MKS DS65K qui apportent une grande rapidité lors des déplacements avec une grande robustesse et une faible masse.

La Figure 2.2 montre le modèle de DarkO, ainsi qu'un repère de référence inertiel *North, east, down* (NED) (ou repère terrestre) "i" lié à la surface de la Terre, et un repère corps 'b" attaché au drone, avec x_b correspondant à l'axe de roulis (l'axe des hélices dans le plan $z_b = 0$), y_b l'axe de tangage (la direction des ailes), z_b l'axe de lacet. En utilisant la même notation que dans [Lustosa 2019], le couple hélice/élévon gauche et droit est désigné par les indices $i = 1$ (gauche) et $i = 2$ (droite). La convention de signe sera définie comme positive pour les positions des élévons δ_1, δ_2 lorsqu'ils créent un moment à cabrer avec les hélices tournant dans des directions opposées avec des vitesses angulaires $\omega_1 > 0$ et $\omega_2 < 0$, respectivement.

Paramètres et coefficients	Valeurs	Unités
m (Masse du drone)	0.519	kg
b (Envergure)	0.542	m
c (Corde aérodynamique)	0.13	m
$B = \text{diag}(b, c, b)$	diag(0.542, 0.13, 0.542)	m
S (Surface de l'aile)	0.026936	m^2
S_{wet} (Surface soufflée)	0.0180	m^2
S_p (Surface des hélices)	0.0127	m^2
$J = \text{diag}(J_x, J_y, J_z)$	diag(0.0067, 0.0012, 0.0082)	kg m^2
k_f (Poussée des hélices)	1.7800e-8	kg m
k_m (Moment des hélices)	2.1065e-10	kg m^2
p_x (Position en x des hélices)	0.065	m
p_y (Position en y des hélices)	0.162	m
a_y (Position en y de la portance)	0.1504	m
ξ_f (Portance des élévons)	0.2	—
ξ_m (Moment des élévons)	1.4	—
ρ (Densité de l'air)	1.225	kg m^{-3}
C_d (Trainée)	0.1644	—
C_y (Latéral)	0	—
C_ℓ (Portance)	5.4001	—
Δ_r (Centrage du drone)	-0.0145	m

TABLE 2.1 – Paramètres numériques identifiés du modèle DarkO.

2.1.1 Modèle non-linéaire complet

En exploitant la modélisation présentée dans [Lustosa 2019, Olszanecki Barth 2020] et récemment validée par [Murali 2024], un modèle précis de la dynamique de DarkO permet de relier la position du centre de gravité $\mathbf{p} \in \mathbb{R}^3$, sa vitesse $\mathbf{v} = \dot{\mathbf{p}} \in \mathbb{R}^3$.

Son orientation est représentée par un quaternion unitaire $\mathbf{q} \in \mathbb{S}^3 := \{\mathbf{q} \in \mathbb{R}^4 : |\mathbf{q}| = 1\}$ et $\mathbf{q} := [\eta \ \boldsymbol{\epsilon}^\top]^\top$. Nous avons donc η qui représente la partie scalaire du quaternion et $\boldsymbol{\epsilon}$ qui représente la partie vectorielle du quaternion.

La vitesse angulaire $\boldsymbol{\omega}_b$, représentée dans le repère du corps telle que $\dot{\mathbf{q}} = \frac{1}{2}\mathbf{q} \otimes [\begin{smallmatrix} 0 \\ \boldsymbol{\omega}_b \end{smallmatrix}]$, où \otimes représente le produit hamiltonien (c'est à-dire le produit de quaternions) couramment utilisé dans le contexte des quaternions. Le produit hamiltonien est défini par :

$$\mathbf{p} \otimes \mathbf{q} = \begin{bmatrix} \eta_p \eta_q - \boldsymbol{\epsilon}_p \cdot \boldsymbol{\epsilon}_q \\ \eta_p \boldsymbol{\epsilon}_q + \eta_q \boldsymbol{\epsilon}_p + \boldsymbol{\epsilon}_p \times \boldsymbol{\epsilon}_q \end{bmatrix}. \quad (2.1)$$

En choisissant l'état global comme $\mathbf{x} := (\mathbf{p}, \mathbf{v}, \mathbf{q}, \boldsymbol{\omega}_b)$, le modèle mathématique, dérivé dans [Lustosa 2019], dépend d'un ensemble de paramètres énumérés dans la Table 2.1. Les valeurs numériques sont obtenues à partir d'une identification du système [Sansou 2022a] et la démarche est expliquée dans la section 2.2.

Le modèle dynamique peut être écrit comme ci-dessous :

$$\dot{\mathbf{p}} = \mathbf{v} \quad (2.2a)$$

$$m\dot{\mathbf{v}} = -m\mathbf{g} + \mathbf{R}(\mathbf{q})\mathbf{F}_b, \quad (2.2b)$$

$$\dot{\mathbf{q}} = \frac{1}{2}\mathbf{q} \otimes \boldsymbol{\omega}_b \quad (2.2c)$$

$$\mathbf{J}\dot{\boldsymbol{\omega}}_b = -[\boldsymbol{\omega}_b]_\times \mathbf{J}\boldsymbol{\omega}_b + \mathbf{M}_b, \quad (2.2d)$$

où $\mathbf{g} := [0 \ 0 \ 9.81]^\top$ désigne le vecteur de gravité, $m \in \mathbb{R}$ est la masse, $\mathbf{J} \in \mathbb{R}^{3 \times 3}$ est le moment d'inertie diagonal (voir Table 2.1), la matrice de rotation correspondant au quaternion q est $\mathbf{R}(\mathbf{q}) \in SO(3) := \{\mathbf{R} \in \mathbb{R}^{3 \times 3} : \mathbf{R}^\top \mathbf{R} = \mathbb{I}_3 \text{ et } \det(\mathbf{R}) = 1\}$ est défini comme (voir [Hua 2013]) :

$$\mathbf{R}(\mathbf{q}) := \mathbb{I}_3 + 2\eta [\boldsymbol{\epsilon}]_\times + 2 [\boldsymbol{\epsilon}]_\times^2. \quad (2.3)$$

D'après [Lustosa 2019], le vecteur de force \mathbf{F}_b et le vecteur de moment \mathbf{M}_b dans (2.2) dépendent premièrement de l'état du système \mathbf{x} , puis de la perturbation $\mathbf{w} \in \mathbb{R}^3$, représentant la vitesse du vent dans le référentiel inertiel, et enfin de la

commande des actionneurs (voir Figure 2.2), comprenant la vitesse de rotation des deux hélices $\omega_1, \omega_2 \in \mathbb{R}$ et la déflexion des élevons $\delta_1, \delta_2 \in \mathbb{R}$.

Considérons premièrement l'effet des commandes des actionneurs. Chaque hélice génère une poussée \mathbf{T}_i orientée dans la direction x du repère corps et un moment \mathbf{N}_i selon le même axe :

$$\mathbf{T}_i := \begin{bmatrix} \tau_i \\ 0 \\ 0 \end{bmatrix} := \begin{bmatrix} k_f \omega_i^2 \\ 0 \\ 0 \end{bmatrix}, \quad \mathbf{N}_i := (-1)^i \frac{k_m}{k_f} \mathbf{T}_i, \quad i = 1, 2. \quad (2.4)$$

La position de chaque élévon $\delta_i \in \mathbb{R}$ est assignée par un servomoteur qui impose un niveau d'efficacité (en termes de déviation du courant d'air) quantifié par deux matrices antisymétriques :

$$\Delta_i^f := \begin{bmatrix} 0 & 0 & \xi_f \delta_i \\ 0 & 0 & 0 \\ -\xi_f \delta_i & 0 & 0 \end{bmatrix}, \quad \Delta_i^m := \begin{bmatrix} 0 & 0 & \xi_m \delta_i \\ 0 & 0 & 0 \\ -\xi_m \delta_i & 0 & 0 \end{bmatrix}, \quad i = 1, 2. \quad (2.5)$$

Les paramètres constants k_f, k_m, ξ_f, ξ_m apparaissant dans (2.4) et (2.5) sont listés dans la Table 2.1.

Avec les quantités ci-dessus, nous pouvons réarranger la dynamique donnée dans [Lustosa 2019, eqns (97), (98)] (voir aussi [Sansou 2022a]) et exprimer \mathbf{F}_b et \mathbf{M}_b dans (2.2) comme :

$$\begin{aligned} \mathbf{F}_b := & \mathbf{T}_1 + \mathbf{T}_2 + \frac{S_{\text{wet}}}{4S_p} \Phi^{(\text{fv})} ((\Delta_1^f - \mathbb{I}_3) \mathbf{T}_1 + (\Delta_2^f - \mathbb{I}_3) \mathbf{T}_2) \\ & + \frac{1}{4} \rho S \Phi^{(\text{fv})} (\Delta_1^f + \Delta_2^f - 2\mathbb{I}_3) \|\mathbf{v}_b\| \mathbf{v}_b \\ & + \frac{1}{4} \rho S \Phi^{(\text{mv})} (\Delta_1^f + \Delta_2^f - 2\mathbb{I}_3) \mathbf{B} \|\mathbf{v}_b\| \boldsymbol{\omega}_b, \end{aligned} \quad (2.6)$$

$$\begin{aligned} \mathbf{M}_b := & \mathbf{N}_1 + \mathbf{N}_2 + \begin{bmatrix} p_x \\ p_y \\ 0 \end{bmatrix}_{\times} \mathbf{T}_1 + \begin{bmatrix} p_x \\ -p_y \\ 0 \end{bmatrix}_{\times} \mathbf{T}_2 \\ & - \frac{S_{\text{wet}}}{4S_p} \left(\mathbf{B} \Phi^{(\text{mv})} (\Delta_1^m - \mathbb{I}_3) + \begin{bmatrix} 0 \\ a_y \\ 0 \end{bmatrix}_{\times} \Phi^{(\text{fv})} (\Delta_1^m + \mathbb{I}_3) \right) \mathbf{T}_1 \\ & - \frac{S_{\text{wet}}}{4S_p} \left(\mathbf{B} \Phi^{(\text{mv})} (\Delta_2^m - \mathbb{I}_3) + \begin{bmatrix} 0 \\ -a_y \\ 0 \end{bmatrix}_{\times} \Phi^{(\text{fv})} (\Delta_2^m + \mathbb{I}_3) \right) \mathbf{T}_2 \\ & + \frac{1}{4} \rho S \left(\left(\begin{bmatrix} 0 \\ a_y \\ 0 \end{bmatrix}_{\times} \Phi^{(\text{fv})} + \mathbf{B} \Phi^{(\text{mv})} \right) \Delta_1^m \right. \end{aligned}$$

$$\begin{aligned}
& + \left(\begin{bmatrix} 0 \\ -a_y \\ 0 \end{bmatrix}_{\times} \Phi^{(\text{fv})} + \mathbf{B} \Phi^{(\text{mv})} \right) \Delta_2^{\text{m}} - 2 \mathbf{B} \Phi^{(\text{mv})} \Big) \| \mathbf{v}_b \| \mathbf{v}_b \\
& + \frac{1}{4} \rho S \left(\begin{bmatrix} 0 \\ a_y \\ 0 \end{bmatrix}_{\times} \Phi^{(\text{mv})} + \mathbf{B} \Phi^{(\text{m}\omega)} \right) \Delta_1^{\text{m}} \\
& + \left(\begin{bmatrix} 0 \\ -a_y \\ 0 \end{bmatrix}_{\times} \Phi^{(\text{mv})} + \mathbf{B} \Phi^{(\text{m}\omega)} \right) \Delta_2^{\text{m}} - 2 \mathbf{B} \Phi^{(\text{m}\omega)} \Big) \mathbf{B} \| \mathbf{v}_b \| \boldsymbol{\omega}_b,
\end{aligned} \tag{2.7}$$

où $\mathbf{v}_b := \mathbf{R}^T(\mathbf{q})(\mathbf{v} - \mathbf{w})$ représente la vitesse de l'air vue par le drone et exprimée dans le repère du corps.

Dans ce travail, la description de \mathbf{F}_b et \mathbf{M}_b est simplifiée par rapport à celle proposée dans la ϕ -théorie [Lustosa 2019]. En effet, la valeur de ϕ a été identifié comme nulle, ce qui engendre $\eta = \sqrt{\|\mathbf{v}_b\|^2 + \phi c^2 \|\boldsymbol{\omega}_b\|^2} = \|\mathbf{v}_b\|$ dans [Lustosa 2019, équation (17)].

La matrice des coefficients aérodynamiques constants $\Phi := \begin{bmatrix} \Phi^{(\text{fv})} & \Phi^{(\text{mv})} \\ \Phi^{(\text{mv})} & \Phi^{(\text{m}\omega)} \end{bmatrix} \in \mathbb{R}^{6 \times 6}$, est définie dans [Olszanecki Barth 2020, eqs. (6)–(9)] comme $\Phi^{(\text{fv})} := \text{diag}(C_d, C_y, C_\ell)$ et

$$\begin{aligned}
& \left[\begin{array}{c|c} \Phi^{(\text{mv})} & \Phi^{(\text{m}\omega)} \end{array} \right] := \\
& \left[\begin{array}{ccc|ccc} 0 & 0 & 0 & 0.1396 & 0 & 0.0573 \\ 0 & 0 & -\frac{\Delta_r}{c} C_\ell & 0 & 0.6358 & 0 \\ 0 & 0 & 0 & 0.0405 & 0 & 0.0019 \end{array} \right].
\end{aligned}$$

2.1.2 Modèle non-linéaire simplifié à basse vitesse

Dans la mesure où nous allons nous intéresser au maintien du drone en stationnaire, c'est-à-dire avec une vitesse du drone faible, nous pouvons simplifier la dynamique (2.2) en négligeant les effets aérodynamiques quadratiques dus à la vitesse \mathbf{v}_b et à la vitesse angulaire $\boldsymbol{\omega}_b$ dans (2.6) et (2.7). Nous définissons le vecteur de commande :

$$\mathbf{u} := [\tau_1 \ \tau_2 \ \delta_1 \ \delta_2]^T, \tag{2.8}$$

lequel permet d'obtenir le modèle basse vitesse comportant les effets majeurs non-linéaires du vent :

$$\dot{\mathbf{p}} = \mathbf{v}, \tag{2.9a}$$

$$m\dot{\mathbf{v}} = -mg + \mathbf{R}(\mathbf{q}) \left(\mathbf{M}_f(\mathbf{u}) + \mathbf{D}_f(\mathbf{u}) \| \mathbf{w} \| \mathbf{R}^T(\mathbf{q})(\mathbf{v} - \mathbf{w}) \right), \tag{2.9b}$$

$$\dot{\mathbf{q}} = \frac{1}{2} \mathbf{q} \otimes \begin{bmatrix} 0 \\ \omega_b \end{bmatrix}, \quad (2.9c)$$

$$\mathbf{J}\dot{\boldsymbol{\omega}}_b = -[\boldsymbol{\omega}_b]_{\times} \mathbf{J}\boldsymbol{\omega}_b + \mathbf{M}_m(\mathbf{u}) + \mathbf{D}_m(\mathbf{u})\|\mathbf{w}\| \mathbf{R}^{\top}(\mathbf{q})(\mathbf{v} - \mathbf{w}), \quad (2.9d)$$

où les vecteurs $\mathbf{M}_f(\mathbf{u})$ et $\mathbf{M}_m(\mathbf{u})$, et les matrices $\mathbf{D}_f(\mathbf{u})$ et $\mathbf{D}_m(\mathbf{u})$ proviennent de l'annulation des termes dépendant de la vitesse angulaire dans l'équation (2.6) et (2.7). Ils peuvent être développés en :

$$\begin{aligned} \mathbf{M}_f(\mathbf{u}) &:= \mathbf{T}_1 + \mathbf{T}_2 + \frac{S_{\text{wet}}}{4S_p} \boldsymbol{\Phi}^{(\text{fv})} \left((\Delta_1^f - \mathbb{I}_3) \mathbf{T}_1 + (\Delta_2^f - \mathbb{I}_3) \mathbf{T}_2 \right) \\ &= \begin{bmatrix} \left(1 - \frac{S_{\text{wet}}}{4S_p} C_d\right) (\tau_1 + \tau_2) \\ 0 \\ -\frac{S_{\text{wet}}}{4S_p} C_\ell \xi_f (\delta_1 \tau_1 + \delta_2 \tau_2) \end{bmatrix}, \end{aligned} \quad (2.10)$$

$$\begin{aligned} \mathbf{M}_m(\mathbf{u}) &:= \mathbf{N}_1 + \mathbf{N}_2 + \begin{bmatrix} p_x \\ p_y \\ 0 \end{bmatrix}_{\times} \mathbf{T}_1 + \begin{bmatrix} p_x \\ -p_y \\ 0 \end{bmatrix}_{\times} \mathbf{T}_2 \\ &\quad - \frac{S_{\text{wet}}}{4S_p} \left(\mathbf{B} \boldsymbol{\Phi}^{(\text{mv})} (\Delta_1^m - \mathbb{I}_3) + \begin{bmatrix} 0 \\ a_y \\ 0 \end{bmatrix}_{\times} \boldsymbol{\Phi}^{(\text{fv})} (\mathbb{I}_3 + \Delta_1^m) \right) \mathbf{T}_1 \\ &\quad - \frac{S_{\text{wet}}}{4S_p} \left(\mathbf{B} \boldsymbol{\Phi}^{(\text{mv})} (\Delta_2^m - \mathbb{I}_3) + \begin{bmatrix} 0 \\ -a_y \\ 0 \end{bmatrix}_{\times} \boldsymbol{\Phi}^{(\text{fv})} (\mathbb{I}_3 + \Delta_2^m) \right) \mathbf{T}_2 \\ &= \begin{bmatrix} \frac{k_m}{k_f} (\tau_1 - \tau_2) + \frac{S_{\text{wet}}}{4S_p} a_y C_\ell \xi_f (\delta_1 \tau_1 - \delta_2 \tau_2) \\ \frac{S_{\text{wet}}}{4S_p} \Delta_r C_\ell \xi_m (\delta_1 \tau_1 + \delta_2 \tau_2) \\ \left(p_y + \frac{S_{\text{wet}}}{4S_p} a_y C_d \right) (\tau_1 - \tau_2) \end{bmatrix}, \end{aligned} \quad (2.11)$$

$$\begin{aligned} \mathbf{D}_f(\mathbf{u}) &:= \frac{1}{4} \rho S \boldsymbol{\Phi}^{(\text{fv})} \left(\Delta_1^f + \Delta_2^f - 2\mathbb{I}_3 \right) \\ &= \frac{1}{4} \rho S \begin{bmatrix} -2C_d & 0 & C_d \xi_f (\delta_1 + \delta_2) \\ 0 & 0 & 0 \\ -C_\ell \xi_f (\delta_1 + \delta_2) & 0 & -2C_\ell \end{bmatrix}, \end{aligned} \quad (2.12)$$

$$\begin{aligned}
\mathbf{D}_m(\mathbf{u}) &:= \frac{1}{4}\rho S \left(\left(\begin{bmatrix} 0 \\ a_y \\ 0 \end{bmatrix} \times \Phi^{(fv)} + \mathbf{B}\Phi^{(mv)} \right) \Delta_1^m \right. \\
&\quad \left. + \left(\begin{bmatrix} 0 \\ -a_y \\ 0 \end{bmatrix} \times \Phi^{(fv)} + \mathbf{B}\Phi^{(mv)} \right) \Delta_2^m - 2\mathbf{B}\Phi^{(mv)} \right) \\
&= \frac{1}{4}\rho S \begin{bmatrix} -a_y C_d \xi_m (\delta_1 - \delta_2) & 0 & 0 \\ \Delta_r C_\ell \xi_m (\delta_1 + \delta_2) & 0 & 2\Delta_r C_\ell \\ 0 & 0 & -a_y C_\ell \xi_m (\delta_1 - \delta_2) \end{bmatrix}, \tag{2.13}
\end{aligned}$$

où l'on observe l'effet d'un vent non nul. De plus, la dynamique est non-linéaire vis-à-vis de \mathbf{q} , $\|\mathbf{v}_b\|$ et \mathbf{w} . Comme dans [Olszanecki Barth 2020, eqn. (10)] et selon la formule de Diederich, nous obtenons $C_\ell = C_d + \frac{\pi AR}{1 + \sqrt{1 + \left(\frac{AR}{2}\right)^2}}$ où $AR = \frac{b^2}{S}$ est l'allongement de l'aile. Nous observons le couplage des actionneurs ($\delta_1\tau_1 + \delta_2\tau_2$) dans les expressions des vecteurs $\mathbf{M}_f(\mathbf{u})$ et $\mathbf{M}_m(\mathbf{u})$.

2.2 Caractérisation de la maquette

L'obtention d'un modèle cohérent avec la réalité est liée à la caractérisation des paramètres constitutifs du modèle. Toutefois, lors de l'utilisation d'un modèle complexe avec de fortes composantes non-linéaires, il peut être difficile de caractériser une valeur. Nous avons donc choisi une méthode itérative basée sur l'identification individuelle des coefficients caractéristiques par des méthodes dédiées. Ainsi, nous avons obtenu les constantes des moteurs par une mesure sur un capteur de force et de moment, la matrice d'inertie à l'aide d'un montage de pendule bifilaire et un montage en soufflerie pour mesurer les coefficients aérodynamiques du modèle.

2.2.1 Identification des paramètres du modèle

Les valeurs numériques de la Table 2.1 ont été obtenues par une campagne d'identification du modèle [Sansou 2022a]. En particulier, le coefficient k_f a été identifié à partir de l'équation (2.4), qui relie la vitesse de rotation du moteur ω_i à la traction générée, à la vitesse de rotation minimale et maximale et à la constante de temps de la chaîne d'actionnement du moteur. La Figure 2.3 montre la génération de la force en fonction de la vitesse de rotation du moteur.

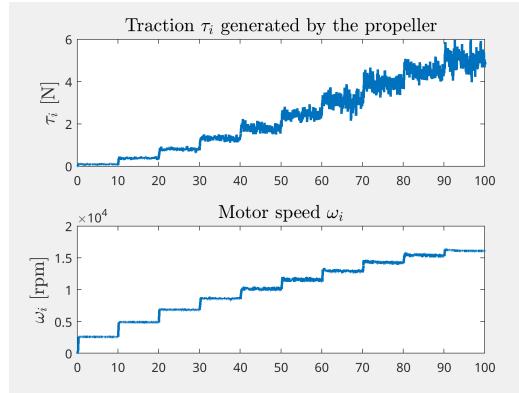


FIGURE 2.3 – Réponse entrée-sortie de l’ensemble moteur/hélice.

Pour effectuer l’identification des trois coefficients principaux (diagonaux) de la matrice d’inertie, nous avons réalisé un montage de pendule bifilaire. Cette méthode est largement utilisée dans le domaine des drones [Jardin 2007], et est basée sur la période d’oscillation autour de chacun des trois axes (x_b , y_b , z_b) du drone, lequel est suspendu par deux fils, ce qui forme un pendule de torsion comme le montre la Figure 2.4.

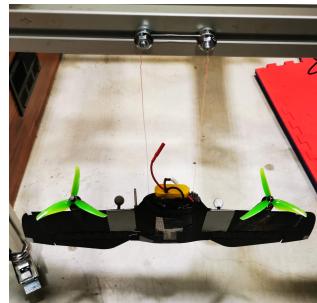


FIGURE 2.4 – Montage d’un pendule bifilaire pour l’identification de l’inertie (\mathbf{J}) de DarkO.

Lors de la mesure, l’autopilote est utilisé pour réaliser une acquisition de l’orientation du drone à 500 Hz. Le drone est positionné avec un angle non nul vis-à-vis de la position d’équilibre du pendule bifilaire puis il est lâché sans vitesse initiale. Le couple de rappel engendré par les deux fils produit des oscillations amorties (voir la Figure 2.5).

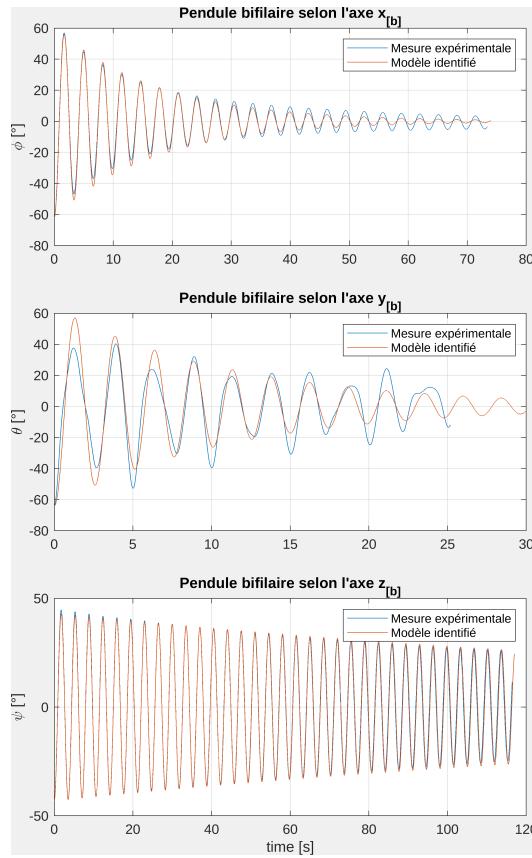


FIGURE 2.5 – Identification de l'inertie (J), à partir des mesures issues du pendule bifilaire 2.4.

Il est nécessaire de connaître la longueur des fils h ainsi que leur écartement D pour réaliser l'identification. Ces valeurs sont mesurées directement sur le banc de mesure pour chacune des trois configurations et reportées dans la table 2.2.

	h	D
J_x	0.962 m	0.142 m
J_y	0.415 m	0.051 m
J_z	1.018 m	0.149 m

TABLE 2.2 – Longueur (h) et espacement (D) des fils du pendule pour chacun des axes.

Une fois la mesure réalisée, nous utilisons l'outil *Simulink Design Optimization* pour obtenir les valeurs de l'amortissement visqueux C , et de l'inertie sur l'axe

mesuré I , à partir du modèle suivant :

$$\ddot{\theta} + \frac{C}{I}\dot{\theta} + \left(\frac{mgD^2}{4Ih}\right) \frac{\sin \theta}{\sqrt{1 - 0.5\left(\frac{D}{h}\right)^2(1 - \cos \theta)}} = 0$$

où θ est l'angle mesuré par l'autopilote à l'aide du code d'estimation d'état utilisant le gyroscope, l'accéléromètre et l'Optitrack.

Les autres coefficients ont été estimés à l'aide d'un montage sur un capteur de forces et moments à 6 degrés de liberté (DOF), observé sur la Figure 2.6.

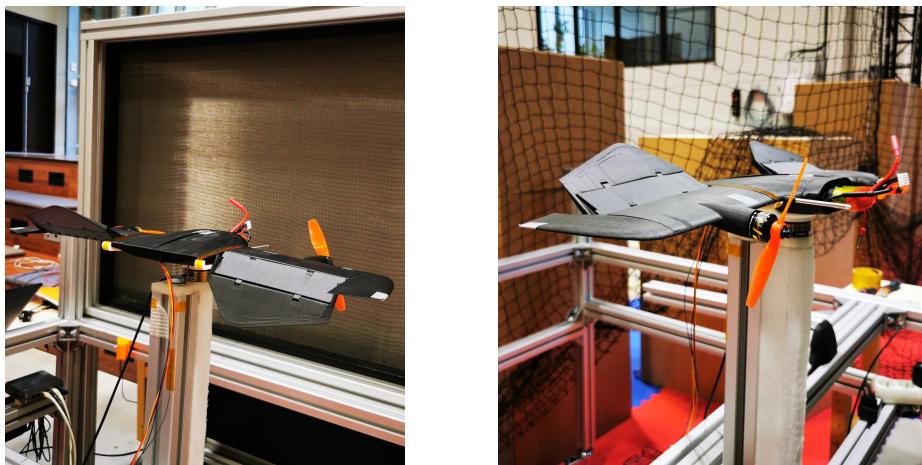


FIGURE 2.6 – Montage de DarkO sur un banc de mesure face à une soufflerie ouverte.

Ces mesures permettent d'estimer la surface soufflée par les hélices. Il est intéressant de noter que cette surface représente 67 % de la surface totale du drone.

2.2.2 Modélisation des actionneurs

Les actionneurs de DarkO ont des dynamiques qui limitent leurs actions en termes d'amplitude et de vitesse.

Pour les moteurs électriques générant la traction par les hélices, il existe deux causes de saturation. Une saturation à haute vitesse liée à la tension maximale du moteur et une saturation basse vitesse liée à la vitesse minimale de commutation de bobine du moteur, pour maintenir la rotation. De plus, ces saturations permettent d'obtenir un modèle réaliste à énergie finie. Elles correspondent à la contrainte suivante : $\omega_i \in [2500, 16000] \text{ rpm} = [262, 1675] \text{ rad s}^{-1}$, $i = 1, 2$.

En termes de dynamique, nous avons représenté la chaîne d'actionnement du moteur (composée de l'ESC, du moteur et de l'hélice) par une fonction de transfert

du premier ordre ayant une constante de temps égale à 0.0125 s, ce qui fournit un système d'actionnement assez agressif. Cette constante de temps a pu être identifiée lors du montage présenté sur la Figure 2.6, où nous avons piloté un démarrage des moteurs et mesuré la génération de force par le capteur.

Les saturations impactant les élevons proviennent des limites physiques des servomoteurs et du débattement limité par la forme de l'UAV, $\delta_i \in [-30 ; 30]^\circ$, $i = 1, 2$. La contrainte la plus importante, ici, est la bande passante de l'actionneur (due à l'actionnement du servomoteur), qui est modélisée par une fonction de transfert du premier ordre avec une constante de temps 0.05 s.

2.3 Équilibres stationnaires

Notre travail a pour point de départ les travaux réalisés par [Olszanecki Barth 2020, Smeur 2020] qui utilisent des lois de commande basées mesure. Ces lois de commande fournissent de très bonnes propriétés de stabilisation, même en présence de perturbations. Toutefois, elles permettent d'atteindre des équilibres qui n'ont jamais été caractérisés. Ce manque engendre une incompréhension des mécanismes en jeu lors de la déstabilisation. Effectivement, les expérimentations réalisées avec un bouclage INDI ont montré une oscillation de DarkO pour certaines vitesses de vent horizontal.

Ainsi, il se pose la question de savoir si ce comportement est dû à un cycle limite lié à des saturations des actionneurs ou bien à des effets aérodynamiques. Pour répondre à cette question, nous allons étudier l'ensemble des points d'équilibre du drone soumis ou non à du vent.

2.3.1 Équilibre stationnaire sans vent

Nous proposons une modification du vecteur de commande, dans le cas d'un équilibre sans vent $\mathbf{w}_{\text{eq}} = 0$, basé sur le couplage des actionneurs.

$$\mathbf{u}_{\text{nowind}} := [\tau_1 \ \tau_2 \ \delta_1 \tau_1 \ \delta_2 \tau_2]^\top \quad (2.14)$$

Nous soulignons que le vecteur $\mathbf{u}_{\text{nowind}}$ dans (2.14) correspond à une transformation non inversible des actionneurs de DarkO correspondant à $\mathbf{u} := [\tau_1 \ \tau_2 \ \delta_1 \ \delta_2]^\top$ ((2.8)). Néanmoins, si l'on impose les contraintes de saturation décrites dans la section 2.2.2, il est possible de déterminer de manière unique \mathbf{u} à partir d'une valeur souhaitée de $\mathbf{u}_{\text{nowind}}$ dans (2.14). Les valeurs positives non nulles de τ_1 et τ_2 peuvent être déterminées à partir des deux premières composantes de $\mathbf{u}_{\text{nowind}}$,

puis δ_1 et δ_2 sont facilement construites à partir des deux dernières composantes de $\mathbf{u}_{\text{nowind}}$.

Nous obtenons un modèle linéaire vis-à-vis de sa commande, dérivé de (2.9) en imposant $\mathbf{w} = 0$,

$$\dot{\mathbf{p}} = \mathbf{v}, \quad m\dot{\mathbf{v}} = -mg + \mathbf{R}(\mathbf{q})\mathbf{F}\mathbf{u}_{\text{nowind}}, \quad (2.15a)$$

$$\dot{\mathbf{q}} = \frac{1}{2}\mathbf{q} \otimes [\begin{smallmatrix} 0 \\ \omega_b \end{smallmatrix}] \quad \mathbf{J}\dot{\boldsymbol{\omega}}_b = -[\boldsymbol{\omega}_b]_{\times} J\boldsymbol{\omega}_b + M\mathbf{u}_{\text{nowind}}, \quad (2.15b)$$

avec les matrices :

$$[\mathbf{F} \mid \mathbf{M}] := \left[\begin{array}{cccc|cccc} a_f & a_f & 0 & 0 & a_m & -a_m & b_m & -b_m \\ 0 & 0 & 0 & 0 & 0 & 0 & c_m & c_m \\ 0 & 0 & b_f & b_f & d_m & -d_m & 0 & 0 \end{array} \right] \quad (2.16)$$

et les scalaires :

$$\left[\begin{array}{c|c} a_f & b_f \\ \hline a_m & b_m \\ c_m & d_m \end{array} \right] = \left[\begin{array}{c|c} 1 - \frac{S_{\text{wet}}}{4S_p} C_d & -\frac{S_{\text{wet}}}{4S_p} C_\ell \xi_f \\ \hline \frac{k_m}{k_f} & \frac{S_{\text{wet}}}{4S_p} a_y C_\ell \xi_f \\ \hline \frac{S_{\text{wet}}}{4S_p} \Delta_r C_\ell \xi_m & p_y + \frac{S_{\text{wet}}}{4S_p} a_y C_d \end{array} \right].$$

Tous les couples d'équilibre $(\mathbf{u}_{\text{nowind}}, \mathbf{x}) = (\mathbf{u}_{\text{nowind,eq}}, \mathbf{x}_{\text{eq}})$ sont paramétrés par une rotation arbitraire autour de l'axe $z_{[i]}$, définie par $\beta \in [-\sqrt{\frac{1}{2}}, \sqrt{\frac{1}{2}}]$. Le point d'équilibre a pour expression :

$$\mathbf{u}_{\text{nowind,eq}} = \frac{mg}{(1 - \frac{S_{\text{wet}}}{4S_p} C_d)} [1 \ 1 \ 0 \ 0]^T \quad (2.17a)$$

$$\mathbf{q}_{\text{eq}} = [\eta_{\text{eq}} \ \boldsymbol{\epsilon}_{\text{eq}}^T]^T = \left[\begin{array}{c} \sqrt{\frac{1}{2}-\beta} \ \beta \\ \frac{2\beta^2-1}{2\sqrt{\frac{1}{2}-\beta}} \ \beta \end{array} \right]^T. \quad (2.17b)$$

En présence d'un vent nul, le degré de liberté β permet d'orienter le drone dans n'importe quelle direction horizontale.

2.3.2 Équilibre stationnaire en présence de vent

À partir des modèles (2.2) et (2.9), nous caractérisons un équilibre stationnaire en présence d'un vent constant $\mathbf{w}_{\text{eq}} = \begin{bmatrix} w_x \\ w_y \\ w_z \end{bmatrix} \in \mathbb{R}^3$ exprimé dans le repère inertiel, tel que $[w_y] \neq 0$, c'est-à-dire qu'il existe toujours un vent horizontal non nul. Ainsi, pour chaque position de référence $\mathbf{p}_{\text{eq}} \in \mathbb{R}^3$, un ensemble de couples état/commande

possible est $(\mathbf{u}_{\text{eq}}, \mathbf{x}_{\text{eq}}) = (\mathbf{u}_{\text{eq}}, \mathbf{p}_{\text{eq}}, \mathbf{v}_{\text{eq}}, \mathbf{q}_{\text{eq}}, \boldsymbol{\omega}_{\text{b,eq}})$ obtenu à l'aide de :

$$\mathbf{u}_{\text{eq}} = \begin{bmatrix} \tau & \tau & \delta & \delta \end{bmatrix}^\top \quad (2.18a)$$

$$\mathbf{q}_{\text{eq}} = \mathbf{q}_{\text{eq}\psi} \otimes \mathbf{q}_{\text{eq}\theta} \quad (2.18b)$$

$$\boldsymbol{\omega}_{\text{b,eq}} = 0, \quad \mathbf{v}_{\text{eq}} = 0. \quad (2.18c)$$

Nous définissons deux quaternions $\mathbf{q}_{\text{eq}\psi}$ et $\mathbf{q}_{\text{eq}\theta}$ permettant d'exprimer l'ensemble des conditions de vent dans le repère inertiel vers un repère tourné où le vent est toujours contenu dans le plan $x - z$. Grâce à cette transformation, nous exprimons un ensemble continu d'équilibres en présence de vent :

$$\mathbf{q}_{\text{eq}\theta} := \begin{bmatrix} \cos(\frac{\theta}{2}) & 0 & \sin(\frac{\theta}{2}) & 0 \end{bmatrix}^\top \quad (2.19)$$

$$\mathbf{q}_{\text{eq}\psi} := \begin{bmatrix} \cos(\frac{\psi}{2}) & 0 & 0 & \sin(\frac{\psi}{2}) \end{bmatrix}^\top. \quad (2.20)$$

Les paramètres de l'équilibre sont la rotation horizontale $\psi = \arctan(w_x, w_y)$, l'angle d'inclinaison θ , la poussée des hélices τ , et la déflexion des élevons δ . Ils peuvent être obtenus à partir de l'algorithme 1.

Théorème 1. Pour tout vent constant, $\mathbf{w} = [w_x \ w_y \ w_z]^\top \in \mathbb{R}^3$ ayant une composante horizontale non nulle $[w_x]$, les équations (2.20)–(2.19) avec θ , τ et δ sélectionnées selon l'Algorithme 1 caractérisent un couple d'équilibre $(\mathbf{u}_{\text{eq}}, \mathbf{x}_{\text{eq}})$ pour la dynamique non-linéaire (2.2) et (2.9).

Démonstration. Dans un premier temps, notons qu'avec l'expression de \mathbf{R} (2.3) et l'expression de ψ dans l'étape 1 de l'Algorithme 1, on peut définir la perturbation à l'équilibre tournée $\mathbf{w}_{\text{r,eq}} := \mathbf{R}_\psi^\top \mathbf{w}_{\text{eq}} := \mathbf{R}^\top (\mathbf{q}_{\text{eq}\psi}) \mathbf{w}_{\text{eq}}$ (voir (2.26) dans l'Algorithme 1), correspondant à la rotation nécessaire pour aligner l'axe $x_{[\text{b}]}$ du repère corps avec la direction du vent. Une fois que le drone est face au vent, il subit un vent avec une composante latérale y nulle et peut ajuster son angle d'inclinaison θ afin de générer la poussée et la portance nécessaires pour compenser les effets du vent dans les directions longitudinale et verticale (l'effet latéral est nul en raison de l'orientation spécifique de l'appareil ψ). Avec cette rotation ψ , il est possible

d'exprimer le vent dans le repère corps comme étant :

$$\begin{aligned}\mathbf{w}_{\text{eq}}^{\text{b}} &:= \begin{bmatrix} w_x^{\text{b}} \\ 0 \\ w_z^{\text{b}} \end{bmatrix} = \mathbf{R}^{\top}(\mathbf{q}_{\text{eq}\theta}) \mathbf{w}_{\text{r,eq}} \\ &= \begin{bmatrix} \cos \theta & 0 & -\sin \theta \\ 0 & 1 & 0 \\ \sin \theta & 0 & \cos \theta \end{bmatrix}^{\top} \begin{bmatrix} w_{rx} \\ 0 \\ w_{rz} \end{bmatrix} = \begin{bmatrix} w_{rx} \cos \theta - w_{rz} \sin \theta \\ 0 \\ w_{rx} \sin \theta + w_{rz} \cos \theta \end{bmatrix}.\end{aligned}\quad (2.21)$$

Nous insistons sur le fait que w_x^{b} est toujours négatif et différent de zéro, car le drone est orienté dans la direction du vent grâce à la rotation engendrée par $\mathbf{q}_{\text{eq}\psi}$, et suite à l'hypothèse $[w_x^{\text{b}}] \neq 0$.

L'équation (2.9a) montre qu'il est nécessaire d'avoir $\mathbf{v}_{\text{eq}} = 0$ pour maintenir l'équilibre stationnaire. En multipliant (2.9b) par $\mathbf{R}(\mathbf{q}_{\text{eq}})$ donnée dans (2.21), nous l'exprimons dans le repère corps. Comme nous appliquons la même commande $\tau_1 = \tau_2 = \tau$ aux deux moteurs et la même commande aux deux élévons $\delta_1 = \delta_2 = \delta$, nous obtenons pour les deux modèles (2.2) et (2.9), l'équilibre des forces selon l'axe $x_{[\text{b}]}$ donné par :

$$(2 - \frac{S_{\text{wet}}}{2S_p} C_d) \tau - \frac{1}{2} \rho S \| \mathbf{w}_{\text{eq}} \| C_d (w_x^{\text{b}} - \xi_f \delta w_z^{\text{b}}) - mg \sin(\theta) = 0 \quad (2.22)$$

et l'équilibre des forces selon l'axe $z_{[\text{b}]}$ donné par :

$$-\frac{S_{\text{wet}}}{2S_p} \xi_f C_\ell \tau \delta - \frac{1}{2} \rho S \| \mathbf{w}_{\text{eq}} \| C_\ell (w_z^{\text{b}} + \xi_f \delta w_x^{\text{b}}) + mg \cos(\theta) = 0. \quad (2.23)$$

De manière similaire, à partir de (2.2d) et (2.9d), l'équilibre des moments autour de l'axe $y_{[\text{b}]}$ permet d'obtenir :

$$\frac{S_{\text{wet}}}{2S_p} \Delta_r \xi_m C_\ell \tau \delta + \frac{1}{2} \rho S \Delta_r \| \mathbf{w}_{\text{eq}} \| C_\ell (w_z^{\text{b}} + \xi_m \delta w_x^{\text{b}}) = 0. \quad (2.24)$$

Pour calculer la solution du triplet (θ, τ, δ) des trois équations d'équilibre (2.22)–(2.24), ajoutons (2.23) multipliée par $\Delta_r \xi_m$, à (2.24) multipliée par ξ_f , de manière à annuler le premier terme et à obtenir :

$$\begin{aligned}\Delta_r \xi_m \left(-\frac{1}{2} \rho S \| \mathbf{w}_{\text{eq}} \| C_\ell (w_z^{\text{b}} + \xi_f \delta w_x^{\text{b}}) + mg \cos(\theta) \right) \\ + \xi_f \left(\frac{1}{2} \rho S \Delta_r \| \mathbf{w}_{\text{eq}} \| C_\ell (w_z^{\text{b}} + \xi_m \delta w_x^{\text{b}}) \right) = 0,\end{aligned}$$

qui est équivalent à :

$$\frac{1}{2}\rho S \Delta_r \|\mathbf{w}_{eq}\| C_\ell (\xi_f - \xi_m) w_z^b + \Delta_r \xi_m m g \cos(\theta) = 0,$$

où (w_x^b, w_z^b) sont les première et troisième composantes de \mathbf{w}^b dans (2.21). Ensuite, en utilisant (2.21) et après calculs, nous obtenons :

$$-\frac{1}{2}\rho S \Delta_r \|\mathbf{w}_{eq}\| C_\ell (\xi_f - \xi_m) w_{rx} \sin \theta + \left(-\frac{1}{2}\rho S \Delta_r \|\mathbf{w}_{eq}\| C_\ell (\xi_f - \xi_m) w_{rz} + \Delta_r \xi_m m g \right) \cos \theta = 0,$$

qui est satisfaite par :

$$\theta = -\tan^{-1} \left(\frac{\rho S \|\mathbf{w}_{eq}\| C_\ell (\xi_f - \xi_m) w_{rz} - 2\xi_m m g}{\rho S \|\mathbf{w}_{eq}\| C_\ell (\xi_f - \xi_m) w_{rx}} \right). \quad (2.25)$$

Cette dernière expression coïncide avec la sélection (2.27) dans l'Algorithm 1. À partir de (2.27), nous pouvons calculer les commandes à l'équilibre en substituant (2.22) dans (2.23). Après simplifications, la force nécessaire de traction des hélices τ pour maintenir la position d'équilibre correspond à l'expression (2.28). Finalement, avec la valeur de τ dans (2.28), nous pouvons obtenir la déflexion des elevons nécessaire δ à partir de l'équation (2.23), ce qui nous donne la valeur obtenue dans (2.29). \square

Il est intéressant de noter que pour chaque couple de vent (w_{rz}, w_{rx}) correspond une orientation d'équilibre (2.18b), (2.27) étant indépendante de l'entrée \mathbf{u}_{eq} . En outre, il convient de souligner que pour toutes les valeurs de vent raisonnables, l'équation (2.28) correspond à la racine positive d'un polynôme du second ordre, l'autre racine étant toujours négative, ce qui conduit à une condition de poussée négative physiquement impossible.

À partir de l'expression analytique (2.18) de l'équilibre du drone pour différentes conditions de vent \mathbf{w} , nous reportons, sur la Figure 2.7, les valeurs correspondantes de θ , δ , τ pour des valeurs de vitesse de vent horizontal allant de 0 à -20 m s^{-1} et pour des valeurs de vitesse de vent vertical allant de -6 à 6 m s^{-1} .

L'angle d'incidence θ diminue de 90° à -4.65° . $\theta = 90^\circ$ correspond à un vol stationnaire sans vent. La traction τ atteint son minimum à $w_{rx} = -12.8 \text{ m s}^{-1}$, ce qui correspond à une condition de vol qui minimise la consommation d'énergie, car les moteurs sont la principale source de consommation électrique.

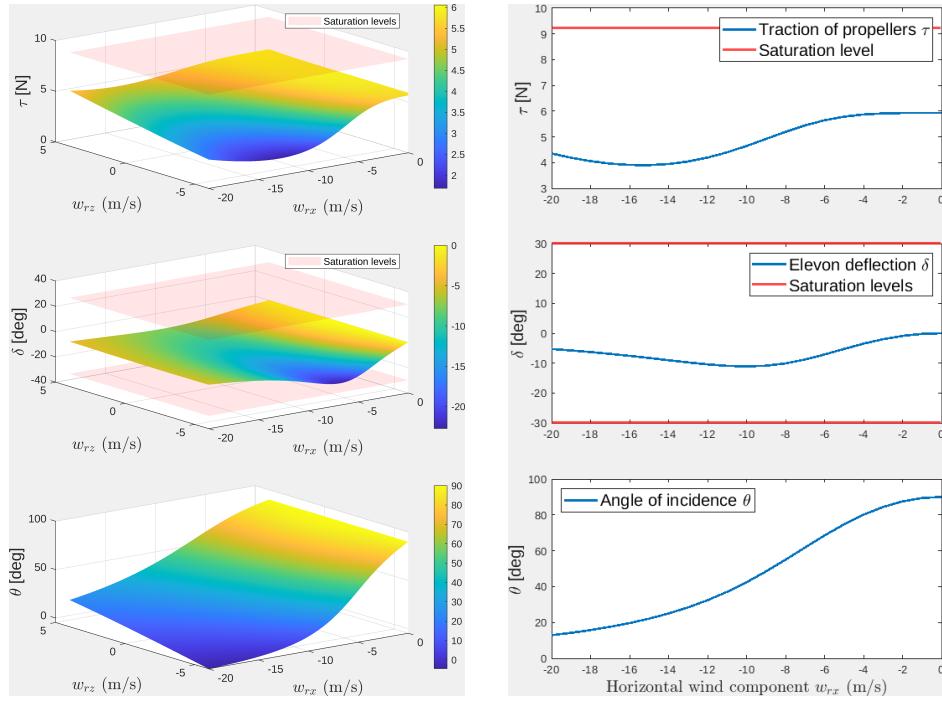


FIGURE 2.7 – Les paramètres (τ , δ , θ) de l’ensemble des points d’équilibre (surface) obtenus à l’aide du Théorème 1 et de l’Algorithm 1 pour un vent constant horizontal et vertical (w_{rx}, w_{rz}), avec les saturations des actionneurs (rose) (gauche) et coupe pour $w_{rx} = 0$ (droite).

Il est possible de faire une coupe des surfaces présentées dans (2.7) gauche pour une vitesse verticale nulle $w_{rx} = 0$, ce qui nous donne le résultat de la Figure (2.7) droite.

Algorithme 1 Obtention des paramètres d'équilibre en (2.18).

Entrée : Vecteur vent $\mathbf{w}_{\text{eq}} = [w_x \ w_y \ w_z]^\top$

Sortie : Paramètres $\psi, \theta, \tau, \delta$ dans (2.18)

- 1: Détermine l'angle $\psi = \text{atan}2(w_x, w_y)$ de manière à obtenir $\mathbf{q}_{\text{eq}\psi}$ dans (2.20)
- 2: Détermine la perturbation tournée \mathbf{w}_r avec la composante y nulle, en utilisant $\mathbf{R}_\psi := \begin{bmatrix} \cos \psi & \sin \psi & 0 \\ -\sin \psi & \cos \psi & 0 \\ 0 & 0 & 1 \end{bmatrix}$, selon :

$$\mathbf{w}_{r,\text{eq}} := \begin{bmatrix} w_{rx} \\ 0 \\ w_{rz} \end{bmatrix} := \mathbf{R}^\top(\mathbf{q}_{\text{eq}\psi}) \mathbf{w}_{\text{eq}} = \mathbf{R}_\psi^\top \mathbf{w}_{\text{eq}} \quad (2.26)$$

- 3: Détermine l'angle d'inclinaison θ de manière à obtenir $\mathbf{q}_{\text{eq}\theta}$ dans (2.18b) :

$$\theta = -\tan^{-1} \left(\frac{w_{rz}}{w_{rx}} + \frac{2mg}{\rho S \|\mathbf{w}_{\text{eq}}\| C_\ell (1 - \frac{\xi_f}{\xi_m}) w_{rx}} \right) \quad (2.27)$$

- 4: Pour des raisons de commodité, nous définissons les scalaires :

$$\begin{bmatrix} a & b \\ c & d \end{bmatrix} := \begin{bmatrix} 2S_{\text{wet}}C_\ell mg \sin \theta \xi_f & 2S_{\text{wet}}C_d C_\ell \rho \|\mathbf{w}_{\text{eq}}\| w_x^b \\ -4SS_p C_\ell \rho \|\mathbf{w}_{\text{eq}}\| w_x^b \xi_f & \frac{b\xi_f}{2} \end{bmatrix}$$

et grâce à ces scalaires (a, b, c, d) , déterminons la traction des hélices τ dans (2.18a) comme :

$$\begin{aligned} \tau = \frac{S_p}{2S_{\text{wet}}C_\ell \xi_f (4S_p - S_{\text{wet}}C_d)} & \left(a + b + c + d + \sqrt{(a + b + c - d)^2 - 4(d^2 + ac - bd)} \right. \\ & \left. - \frac{4w_z^b d}{w_x^b} (d + c) + \frac{4w_z^b ad \cos \theta}{w_x^b C_\ell \sin \theta} \left(C_d - \frac{4S_p}{S_{\text{wet}}} \right) \right)^{\frac{1}{2}}, \end{aligned} \quad (2.28)$$

où

$$\begin{bmatrix} w_x^b \\ w_z^b \end{bmatrix} = \begin{bmatrix} w_{rx} \cos \theta - w_{rz} \sin \theta \\ w_{rx} \sin \theta + w_{rz} \cos \theta \end{bmatrix}.$$

- 5: Déterminons la déflexion des élévons δ comme :

$$\delta = \frac{2mg \sin \theta}{\rho S \|\mathbf{w}_{\text{eq}}\| C_d \xi_f w_z^b} + \frac{w_x^b}{\xi_f w_z^b} - \frac{(4 - \frac{S_{\text{wet}}}{S_p} C_d)}{\rho S \|\mathbf{w}_{\text{eq}}\| C_d \xi_f w_z^b} \tau. \quad (2.29)$$

Retourne : $\psi, \theta, \tau, \delta$

2.4 Dynamiques linéarisées

Des points d'équilibres, obtenus dans la section 2.3 et notamment de l'Algorithm 1, il semble cohérent de poursuivre notre travail en obtenant les dynamiques linéarisées autour des points d'équilibre. Ces résultats permettront, par exemple, de concevoir des lois de commande linéaires basées sur des modèles. De nombreux outils existent pour le contrôle des modèles linéaires. Tout l'intérêt réside dans le fait d'avoir un modèle linéarisé autour d'un point d'équilibre assez consistant. Nous pouvons définir la consistance d'une linéarisation comme la précision du modèle linéaire vis-à-vis du modèle non-linéaire. Nous sommes intéressés par la meilleure prise en compte des phénomènes non-linéaires telles que les actions aérodynamiques.

2.4.1 Dynamique linéarisée sans vent

Considérons le cas sans vent discuté dans la section 2.3.1 pour lequel nous utilisons le vecteur de commande $\mathbf{u}_{\text{nowind}}$ et le vecteur de commande à l'équilibre $\mathbf{u}_{\text{nowind},\text{eq}}$ défini dans l'équation (2.17a) et rappelons la transformation du vecteur de commande suivant $\mathbf{u}_{\text{nowind}} := [\tau_1 \ \tau_2 \ \delta_1\tau_1 \ \delta_2\tau_2]^\top$. La dynamique linéarisée dans le cas sans vent est :

$$\dot{\tilde{\mathbf{x}}} = \mathbf{A}_0 \tilde{\mathbf{x}} + \mathbf{G}_0 (\mathbf{u}_{\text{nowind}} - \mathbf{u}_{\text{nowind},\text{eq}}), \quad (2.30)$$

où l'expression de \mathbf{A}_0 est :

$$\mathbf{A}_0 = \mathbf{A}_w \Big|_{\mathbf{w}=0} = \begin{bmatrix} \mathbf{0}_3 & \mathbb{I}_3 & \mathbf{0}_3 & \mathbf{0}_3 \\ \mathbf{0}_3 & \mathbf{0}_3 & \mathbf{A}_{v\epsilon} & \mathbf{0}_3 \\ \mathbf{0}_3 & \mathbf{0}_3 & \mathbf{0}_3 & \mathbf{A}_{\epsilon\omega} \\ \mathbf{0}_3 & \mathbf{0}_3 & \mathbf{0}_3 & \mathbf{0}_3 \end{bmatrix}, \quad (2.31)$$

avec les matrices suivantes :

$$\mathbf{A}_{\epsilon\omega} = \frac{\sqrt{2}}{4} \begin{bmatrix} 1 & 0 & -1 \\ 0 & 1 & 0 \\ 1 & 0 & 1 \end{bmatrix} \text{ et } \mathbf{A}_{v\epsilon} = \sqrt{2} \begin{bmatrix} 0 & -2g & 0 \\ g & 0 & g \\ 0 & -2g & 0 \end{bmatrix},$$

alors que l'expression de \mathbf{G}_0 est :

$$\mathbf{G}_0 := \begin{bmatrix} \mathbb{0}_{3 \times 1} & \mathbb{0}_{3 \times 1} & \mathbb{0}_{3 \times 1} & \mathbb{0}_{3 \times 1} \\ 0 & 0 & a_g & a_g \\ 0 & 0 & 0 & 0 \\ b_g & b_g & 0 & 0 \\ \mathbb{0}_{3 \times 1} & \mathbb{0}_{3 \times 1} & \mathbb{0}_{3 \times 1} & \mathbb{0}_{3 \times 1} \\ c_g & -c_g & d_g & -d_g \\ 0 & 0 & e_g & e_g \\ f_g & -f_g & 0 & 0 \end{bmatrix}, \quad (2.32)$$

avec :

$$\left[\begin{array}{c|c} a_g & b_g \\ \hline c_g & d_g \\ \hline e_g & f_g \end{array} \right] = \left[\begin{array}{c|c} -\frac{S_{\text{wet}}}{4mS_p} C_\ell \xi_f & \frac{1}{m}(1 - \frac{S_{\text{wet}}}{2S_p} C_d) \\ \frac{k_m}{J_x k_f} & \frac{S_{\text{wet}} a_y}{4J_x S_p} C_\ell \xi_f \\ \frac{S_{\text{wet}} \Delta_r}{4J_y S_p} C_\ell \xi_m & \frac{1}{J_z}(p_y + \frac{S_{\text{wet}}}{4S_p} a_y C_d) \end{array} \right].$$

2.4.2 Dynamique linéarisée en présence de vent

Pour chacun des équilibres caractérisés dans le Théorème 1, nous détaillons les équations linéarisées du mouvement par rapport au modèle non-linéaire simplifié à faible vitesse (2.9). Une approche directe conduirait à des équations linéarisées qui dépendent de l'angle ψ caractérisé à l'étape 1 de l'Algorithme 1. Au lieu de cela, nous définissons ici les coordonnées incrémentales dans un cadre de référence inertiel convenablement tourné, de sorte que la dynamique linéarisée soit indépendante de l'angle ψ . Pour chaque condition de vent d'équilibre \mathbf{w}_{eq} associée à l'équilibre $(\mathbf{u}_{\text{eq}}, \mathbf{p}_{\text{eq}}, \mathbf{v}_{\text{eq}}, \mathbf{q}_{\text{eq}}, \boldsymbol{\omega}_{\text{b,eq}})$ caractérisé en (2.20)–(2.19), nous étudions ici la dynamique incrémentale linéaire du vecteur d'état tourné :

$$\begin{aligned} \tilde{\mathbf{x}} &:= (\tilde{\mathbf{p}}, \tilde{\mathbf{v}}, \tilde{\boldsymbol{\epsilon}}, \tilde{\boldsymbol{\omega}}_{\text{b}}) = \left(\mathbf{R}_\psi^\top (\mathbf{p} - \mathbf{p}_{\text{eq}}), \mathbf{R}_\psi^\top \mathbf{v}, \mathbf{R}_\psi^\top (\boldsymbol{\epsilon} - \boldsymbol{\epsilon}_{\text{eq}}), \boldsymbol{\omega}_{\text{b}} \right), \\ \tilde{\mathbf{u}} &:= \mathbf{u} - \mathbf{u}_{\text{eq}}, \quad \tilde{\mathbf{w}} := \mathbf{R}_\psi^\top (\mathbf{w} - \mathbf{w}_{\text{eq}}). \end{aligned} \quad (2.33)$$

Nous désignons les composantes scalaire et vectorielle du quaternion en (2.18b) comme $\mathbf{q}_{\text{eq}} = (\eta_{\text{eq}}, \boldsymbol{\epsilon}_{\text{eq}})$, et nous utilisons la matrice de rotation $\mathbf{R}_\psi := \mathbf{R}(\mathbf{q}_{\text{eq}\psi})$ introduite au début de la preuve du Théorème 1. Nous notons que la rotation en (2.33) possède la propriété $\mathbf{R}_\psi^\top \boldsymbol{\epsilon}_{\text{eq}} = [0 \sin(\frac{\theta}{2}) 0]^\top$, ce qui simplifie grandement le mouvement linéarisé.

En exploitant le fait que les vitesses linéaires et angulaires $(\mathbf{v}_{\text{eq}}, \boldsymbol{\omega}_{\text{b,eq}})$ doivent être nulles à l'équilibre (voir (2.18)), nous prouvons ci-dessous que la dynamique

linéarisée de l'état (2.33) est donnée par :

$$\begin{aligned}\dot{\tilde{x}} &= \mathbf{A}_w \tilde{x} + \mathbf{G}_w \tilde{u} + \mathbf{E}_w \tilde{w} \\ &= \begin{bmatrix} \mathbf{0}_3 & \mathbb{I}_3 & \mathbf{0}_3 & \mathbf{0}_3 \\ \mathbf{0}_3 & \mathbf{A}_{vv} & \mathbf{A}_{v\epsilon} & \mathbf{0}_3 \\ \mathbf{0}_3 & \mathbf{0}_3 & \mathbf{0}_3 & \mathbf{A}_{\epsilon\omega} \\ \mathbf{0}_3 & \mathbf{0}_3 & \mathbf{A}_{\omega\epsilon} & \mathbf{0}_3 \end{bmatrix} \tilde{x} + \begin{bmatrix} \mathbf{0}_{3 \times 4} \\ \mathbf{G}_v \\ \mathbf{0}_{3 \times 4} \\ \mathbf{G}_\omega \end{bmatrix} \tilde{u} + \begin{bmatrix} \mathbf{0}_{3 \times 3} \\ \mathbf{E}_v \\ \mathbf{0}_{3 \times 3} \\ \mathbf{E}_\omega \end{bmatrix} \tilde{w},\end{aligned}\quad (2.34)$$

avec les matrices \mathbf{A}_{vv} , $\mathbf{A}_{v\epsilon}$, $\mathbf{A}_{\epsilon\omega_b}$, $\mathbf{A}_{\omega\epsilon}$, \mathbf{G}_v , \mathbf{G}_ω , \mathbf{E}_v , \mathbf{E}_ω construites en suivant l'Algorithme 2.

Théorème 2. *Pour tout vent constant, $\mathbf{w} = [w_x \ w_y \ w_z]^\top \in \mathbb{R}^3$ ayant une composante horizontale non nulle $[w_x]$, et pour le doublet d'équilibre qui découle $(\mathbf{u}_{eq}, \mathbf{x}_{eq})$ de la dynamique (2.9) tel que caractérisée dans (2.20)-(2.19), la dynamique linéarisée du vecteur état incrémental (2.33) est donnée par (2.34), avec les matrices construites comme dans l'Algorithme 2.*

Démonstration. Premièrement, en exploitant la matrice de rotation $\mathbf{R}_\psi := \mathbf{R}(\mathbf{q}_{eq\psi})$ utilisée dans (2.33), nous transformons la dynamique non-linéaire (2.9) en coordonnées tournées :

$$(\mathbf{p}_r, \mathbf{v}_r, \mathbf{q}_r) := (\mathbf{R}_\psi^\top \mathbf{p}, \mathbf{R}_\psi^\top \mathbf{v}, \mathbf{q}_{eq\psi}^{-1} \otimes \mathbf{q}), \quad \mathbf{w}_r := \mathbf{R}_\psi^\top \mathbf{w} \quad (2.35)$$

où $\boldsymbol{\omega}_b$ reste inchangée car elle est exprimée dans le repère du corps. Quelques observations permettent de simplifier la dynamique transformée (2.9) :

- nous avons $\mathbf{R}_\psi^\top m\mathbf{g} = m\mathbf{g}$ car la rotation de ψ est autour de l'axe $z_{[i]}$;
- comme $\mathbf{q}_r = \mathbf{q}_{eq\psi}^{-1} \otimes \mathbf{q}$, alors $\mathbf{R}_\psi^\top \mathbf{R}(\mathbf{q}) = \mathbf{R}(\mathbf{q}_r)$;
- comme $\mathbf{v}_b := \mathbf{R}^\top(\mathbf{q})(\mathbf{v} - \mathbf{w})$ (comme défini après l'équation (2.7)), alors $\|\mathbf{v}_b\| = \|\mathbf{v} - \mathbf{w}\| - \|\mathbf{v}_r - \mathbf{w}_r\|$
- enfin $\mathbf{R}^\top(\mathbf{q})\mathbf{w} = \mathbf{R}^\top(\mathbf{q}_r)\mathbf{R}_\psi^\top \mathbf{R}_\psi \mathbf{w}_r = \mathbf{R}^\top(\mathbf{q}_r)\mathbf{w}_r$.

Sur la base des observations ci-dessus, nous pouvons dériver la version tournée des équations (2.9) comme étant :

$$\dot{\mathbf{p}}_r = \mathbf{v}_r, \quad (2.36a)$$

$$m\dot{\mathbf{v}}_r = -m\mathbf{g} + \mathbf{R}(\mathbf{q}_r) \left(\mathbf{M}_f(\mathbf{u}) + \mathbf{D}_f(\mathbf{u})\|\mathbf{w}_r\| \mathbf{R}^\top(\mathbf{q}_r)(\mathbf{v}_r - \mathbf{w}_r) \right), \quad (2.36b)$$

$$\dot{\mathbf{q}}_r = \left(\frac{1}{2} \mathbf{q}_r \otimes [\mathbf{0}_{\omega_b}] \right), \quad (2.36c)$$

$$\mathbf{J}\dot{\boldsymbol{\omega}}_b = -[\boldsymbol{\omega}_b]_\times \mathbf{J}\boldsymbol{\omega}_b + \mathbf{M}_m(\mathbf{u}) + \mathbf{D}_m(\mathbf{u})\|\mathbf{w}_r\| \mathbf{R}^\top(\mathbf{q}_r)(\mathbf{v}_r - \mathbf{w}_r) \quad (2.36d)$$

Avec ces nouvelles coordonnées, le vecteur d'état incrémental (2.33) peut être ex-

primé comme étant :

$$\begin{aligned}\tilde{\boldsymbol{x}} &= \left(\boldsymbol{p}_r - \boldsymbol{R}_\psi^\top \boldsymbol{p}_{eq}, \boldsymbol{v}_r, \boldsymbol{\epsilon}_r - \boldsymbol{R}_\psi^\top \boldsymbol{\epsilon}_{eq}, \boldsymbol{\omega}_b \right), \\ \tilde{\boldsymbol{u}} &:= \boldsymbol{u} - \boldsymbol{u}_{eq}, \quad \tilde{\boldsymbol{w}} := \boldsymbol{w}_r - \boldsymbol{w}_{r,eq}\end{aligned}\tag{2.37}$$

où $\boldsymbol{w}_{r,eq} = \boldsymbol{R}_\psi^\top \boldsymbol{w}_{eq} = \begin{bmatrix} w_{rx} \\ 0 \\ w_{rz} \end{bmatrix}$, déjà défini dans (2.26), et $\boldsymbol{R}_\psi^\top \boldsymbol{\epsilon}_{eq} = [0 \sin(\frac{\theta}{2}) 0]^\top$ ont tous deux une structure peu dense intéressante.

En se concentrant sur la dynamique tournée (2.36) et sur l'expression (2.37) des variables incrémentales, la preuve du théorème revient à montrer que la linéarisation de (2.36) autour de l'équilibre tourné :

$$\begin{aligned}\boldsymbol{x}_{r,eq} &= \left(\boldsymbol{p}_{r,eq}, \boldsymbol{v}_{r,eq}, \boldsymbol{\epsilon}_{r,eq}, \boldsymbol{\omega}_{br,eq} \right) \\ &= \left(\boldsymbol{R}_\psi^\top \boldsymbol{p}_{eq}, \begin{bmatrix} 0 \\ 0 \\ 0 \end{bmatrix}, \begin{bmatrix} 0 \\ \sin(\frac{\theta}{2}) \\ 0 \end{bmatrix}, \begin{bmatrix} 0 \\ 0 \\ 0 \end{bmatrix} \right), \quad \boldsymbol{w}_{r,eq} = \begin{bmatrix} w_{rx} \\ 0 \\ w_{rz} \end{bmatrix}\end{aligned}\tag{2.38}$$

coïncide avec l'équation (2.34) et les expressions de l'Algorithme 2.

Dans ce but, afin de linéariser la dynamique du quaternion $\boldsymbol{q}_r = [\eta_r \ \boldsymbol{\epsilon}_r^\top]^\top$ évoluant dans \mathbb{S}^3 , nous remplaçons η_r par sa valeur positive liée à la norme unitaire du quaternion. Cette linéarisation est inspirée par [Tregouet 2015, Proof of Lemma 1]. Ainsi, $\eta_r = (1 - \boldsymbol{\epsilon}_r^\top \boldsymbol{\epsilon}_r)^{\frac{1}{2}}$.

Concentrons-nous d'abord sur la matrice \boldsymbol{A}_w dans (2.34). Les trois premières lignes sont simplement $[0_3 \ 1_3 \ 0_3 \ 0_3]$ du fait de la linéarité de l'équation (2.36a). Pour le second bloc de lignes, nous nous focalisons sur l'équation (2.36b) et nous commençons par caractériser $\boldsymbol{R}(\boldsymbol{q}_{r,eq})$, dont la structure est relativement vide puisque $\boldsymbol{\epsilon}_{r,eq}$. Comme rappelé dans (2.21) et en utilisant l'expression \boldsymbol{R} de (2.3), nous pouvons écrire :

$$\boldsymbol{R}(\boldsymbol{q}_{r,eq}) = \boldsymbol{R}_\theta := \begin{bmatrix} 1 - 2\bar{\epsilon}_2^2 & 0 & 2\bar{\epsilon}_2\bar{\eta} \\ 0 & 1 & 0 \\ -2\bar{\epsilon}_2\bar{\eta} & 0 & 1 - 2\bar{\epsilon}_2^2 \end{bmatrix} = \begin{bmatrix} \cos\theta & 0 & \sin\theta \\ 0 & 1 & 0 \\ -\sin\theta & 0 & \cos\theta \end{bmatrix},$$

où $\bar{\epsilon}_2 = \sin \frac{\theta}{2}$ représente le deuxième élément de $\boldsymbol{\epsilon}_{r,eq}$ selon (2.38) et $\bar{\eta} = \sqrt{1 - \bar{\epsilon}_2^2} = \cos \frac{\theta}{2}$.

Avec cette expression de \boldsymbol{R}_θ , nous pouvons dériver l'expression de (2.36b), en utilisant la notation abrégée $\cdot|_{eq}$ pour caractériser l'évaluation d'une fonction (ma-

tricelle ou vectorielle) à l'équilibre (2.38),

$$\begin{aligned}\mathbf{A}_{vv} &= \frac{\partial}{\partial \mathbf{v}} \left(\frac{1}{m} \mathbf{R}(\mathbf{q}_r) \left(\mathbf{D}_f(\mathbf{u}) \| \mathbf{w}_r \| \mathbf{R}^\top(\mathbf{q}_r) (\mathbf{v}_r - \mathbf{w}_r) \right) \right) \Big|_{eq} \\ &= \frac{\partial}{\partial \mathbf{v}} \left(\frac{1}{m} \mathbf{R}_\theta \mathbf{D}_f(\mathbf{u}_{eq}) \| \mathbf{w}_{eq} \| \mathbf{R}_\theta^\top \mathbf{v}_r \right) \Big|_{eq}.\end{aligned}\quad (2.39)$$

Avec cette fonction et compte tenu de l'égalité $\mathbf{D}_{f,eq} = \mathbf{D}_f(\mathbf{u}_{eq})$, il est possible de montrer qu'elle coïncide avec la matrice \mathbf{A}_{vv} donnée en (2.52) dans l'Algorithme 2.

Nous nous concentrons maintenant sur $\mathbf{A}_{v\epsilon}$ de la matrice \mathbf{A}_w , qui doit être calculée à partir de (2.36b) de manière similaire à (2.39), comme :

$$\mathbf{A}_{v\epsilon} = \frac{\partial}{\partial \boldsymbol{\epsilon}} \left(\frac{1}{m} \mathbf{R}(\mathbf{q}_r) \left(\mathbf{M}_f(\mathbf{u}) + \mathbf{D}_f(\mathbf{u}) \| \mathbf{w}_r \| \mathbf{R}^\top(\mathbf{q}_r) \mathbf{w}_r \right) \right) \Big|_{eq}.\quad (2.40)$$

Pour évaluer la partie droite de (2.40), nous démarrons de l'expression de $\mathbf{R}(\boldsymbol{\epsilon}) = \mathbf{R}([\eta \epsilon])$ dans (2.3). Après la substitution de $\eta = \sqrt{1 - \boldsymbol{\epsilon}^\top \boldsymbol{\epsilon}} \neq 0$ (nous rappelons que pour tous les équilibres caractérisés, nous avons $\eta \neq 0$), nous pouvons calculer la dérivée généralisée :

$$\begin{aligned}\partial \mathbf{R}_\epsilon(\boldsymbol{\epsilon}, \mathbf{v}) &:= \frac{\partial}{\partial \boldsymbol{\epsilon}} \mathbf{R} \left(\begin{bmatrix} \sqrt{1 - \boldsymbol{\epsilon}^\top \boldsymbol{\epsilon}} \\ \boldsymbol{\epsilon} \end{bmatrix} \right) \mathbf{v} \\ &= 2\eta [\mathbf{v}]_\times \left(\frac{\boldsymbol{\epsilon} \boldsymbol{\epsilon}^\top}{1 - \boldsymbol{\epsilon}^\top \boldsymbol{\epsilon}} - \mathbb{I}_3 \right) - 4\mathbf{v} \boldsymbol{\epsilon}^\top + 2\boldsymbol{\epsilon} \mathbf{v}^\top + 2\boldsymbol{\epsilon}^\top \mathbf{v} \mathbb{I}_3,\end{aligned}\quad (2.41)$$

qui implique donc :

$$\frac{\partial}{\partial \boldsymbol{\epsilon}} \mathbf{R}^\top([\eta \epsilon]) \mathbf{v} = \frac{\partial}{\partial \boldsymbol{\epsilon}} \mathbf{R} \left(\begin{bmatrix} \sqrt{1 - \boldsymbol{\epsilon}^\top \boldsymbol{\epsilon}} \\ -\boldsymbol{\epsilon} \end{bmatrix} \right) \mathbf{v} = \partial \mathbf{R}_\epsilon(-\boldsymbol{\epsilon}, \mathbf{v}).\quad (2.42)$$

Pour évaluer (2.40), il sera utile de dériver la forme simplifiée suivante :

$$\begin{aligned}\partial \mathbf{R}_\epsilon &\left(\begin{bmatrix} 0 \\ \epsilon_2 \\ 0 \end{bmatrix}, \begin{bmatrix} \mathbf{v}_1 \\ 0 \\ \mathbf{v}_3 \end{bmatrix} \right) \\ &= 2 \begin{bmatrix} 0 & \left(\bar{\eta} - \frac{\bar{\epsilon}_2^2}{\bar{\eta}} \right) \mathbf{v}_3 & 0 \\ -\bar{\eta} \mathbf{v}_3 & 0 & \bar{\eta} \mathbf{v}_1 \\ 0 & \left(\frac{\bar{\epsilon}_2^2}{\bar{\eta}} - \bar{\eta} \right) \mathbf{v}_1 & 0 \end{bmatrix} + 2\bar{\epsilon}_2 \begin{bmatrix} 0 & -2\mathbf{v}_1 & 0 \\ \mathbf{v}_1 & 0 & \mathbf{v}_3 \\ 0 & -2\mathbf{v}_3 & 0 \end{bmatrix}.\end{aligned}\quad (2.43)$$

Nous pouvons définir deux forces (f_d, f_ℓ) qui agissent sur le drone à l'équilibre, exprimées dans le repère corps, et qui dépendent du vent \mathbf{w} et des deux entrées similaires des élévons δ . Ces deux forces sont la traînée et la portance, générées par l'écoulement de l'air sur l'aile. Elles résultent du développement de l'expression

$\mathbf{D}_f(\mathbf{u})\|\mathbf{v}_b\|\mathbf{v}_b$ provenant de (2.36b) avec $\mathbf{D}_f(\mathbf{u})$ de (2.12) :

$$\begin{bmatrix} f_d \\ 0 \\ f_\ell \end{bmatrix} = -\mathbf{D}_f(\mathbf{u}_{eq})\|\mathbf{w}_{eq}\|\mathbf{R}_\theta^\top \mathbf{w}_{r,eq}. \quad (2.44)$$

Après calcul, cette expression coïncide avec celle de (2.50) donnée dans l'Algorithm 2.

À partir des deux forces (f_d, f_ℓ) dans (2.44), il est possible de déterminer leur dérivées partielle par rapport à la composante $\bar{\epsilon}_2$ du quaternion, qui représente le tangage du drone. En utilisant (2.42), nous obtenons :

$$\begin{bmatrix} \frac{\partial f_d}{\partial \epsilon_2} \\ 0 \\ \frac{\partial f_\ell}{\partial \epsilon_2} \end{bmatrix} = -\mathbf{D}_f(\mathbf{u}_{eq})\|\mathbf{w}_{eq}\|\partial \mathbf{R}_\epsilon(-\epsilon, \mathbf{w}_{r,eq}) \begin{bmatrix} 0 \\ 1 \\ 0 \end{bmatrix}, \quad (2.45)$$

qui, après calcul, en utilisant l'égalité $\mathbf{D}_{f,eq} = \mathbf{D}_f(\mathbf{u}_{eq})$, coïncide avec l'équation (2.50), donnée dans l'Algorithm 2.

En suivant des calculs similaires, la force f_m générée par les moteurs, liée à la traction des hélices et à la traînée générée par l'écoulement de l'air sur l'aile, et la force f_e générée par les élévons, liée à l'écoulement de l'air créé par les hélices, sont obtenues à partir de (2.10) et sont définies par :

$$\begin{bmatrix} f_m \\ 0 \\ f_e \end{bmatrix} = \mathbf{M}_f(\mathbf{u}_{eq}). \quad (2.46)$$

Cela coïncide, après calcul, avec les sélections de (2.51), données dans l'Algorithm 2.

En utilisant les définitions (2.41), (2.42), ainsi que les expressions (2.43), (2.44), (2.46), et leurs formes équivalentes indiquées dans (2.50), (2.51) données dans l'Algorithm 2, nous pouvons finalement calculer à partir de (2.40) :

$$\begin{aligned} \mathbf{A}_{v\epsilon} = & \frac{1}{m} (\partial \mathbf{R}_\epsilon(\epsilon, \mathbf{M}_f(\mathbf{u}_{eq})) - \partial \mathbf{R}_\epsilon(\epsilon, \mathbf{D}_f(\mathbf{u}_{eq}))\|\mathbf{w}_{eq}\|\mathbf{R}_\theta^\top \mathbf{w}_{eq}) \\ & - \mathbf{R}_\theta \mathbf{D}_f(\mathbf{u})\|\mathbf{w}_{r,eq}\|\partial \mathbf{R}_\epsilon(-\epsilon, \mathbf{w}_{r,eq}) \begin{bmatrix} 0 \\ 1 \\ 0 \end{bmatrix} \Big|_{eq}, \end{aligned}$$

qui fournit l'expression (2.53) dans l'Algorithm 2, après calculs exploitant également $\mathbf{D}_{text{f,eq}} = \mathbf{D}_{text{f}}(\mathbf{u}_{eq})$.

Nous évaluons maintenant la matrice $\mathbf{A}_{\epsilon\omega}$ de \mathbf{A}_w , et nous rappelons que $[\eta] \otimes [\omega_b] = \begin{bmatrix} -\epsilon^\top \\ \eta \mathbb{I}_3 + [\epsilon] \times \end{bmatrix} \omega_b$, en raison des propriétés du produit de quaternions (voir, par exemple, [Hua 2013] ou (2.1)). À partir des deux termes inférieurs de la matrice du côté droit de cette dernière équation, en développant (2.36c) et en calculant

$\mathbf{A}_{\epsilon\omega} = \frac{\partial}{\partial \omega_b} \left(\frac{1}{2} \mathbf{q}_r \otimes [\begin{smallmatrix} 0 \\ \omega_b \end{smallmatrix}] \right) \Big|_{eq}$, nous obtenons les deux termes de l'expression (2.53) donnée dans l'Algorithme 2.

À propos de la matrice $\mathbf{A}_{\omega\epsilon}$ de \mathbf{A}_w , qui doit être calculée à partir de (2.36d), nous observons que seul le dernier terme de la partie droite dépend de ϵ (par l'intermédiaire de \mathbf{q}_r), ainsi nous obtenons :

$$\mathbf{A}_{\omega\epsilon} = \mathbf{J}^{-1} \mathbf{D}_m(\mathbf{u}_{eq}) \|\mathbf{w}_r\| \frac{\partial}{\partial \epsilon} \left(\mathbf{R}^\top(\mathbf{q}_r)(\mathbf{v}_r - \mathbf{w}_r) \right) \Big|_{eq}. \quad (2.47)$$

Pour calculer l'expression explicite de (2.47), nous exploitons à nouveau (2.42) et (2.43), et utilisons l'expression de \mathbf{D}_m dans (2.13), ainsi que les identités $\bar{\eta}^2 - \bar{\epsilon}_2^2 = \cos \theta$ et $2\bar{\eta}\bar{\epsilon}_2 = \sin \theta$, qui fournissent, après simplifications, l'expression (2.54), donnée dans l'Algorithme 2.

Passons maintenant à la dérivation des entrées de la matrice \mathbf{G}_w dans (2.34), dont les composantes peuvent être dérivées de (2.36b) et (2.36d). En utilisant les quatre entrées de \mathbf{u} dans (2.8), et en se basant également sur la structure de \mathbf{M}_f , \mathbf{D}_f , dans (2.10), (2.12), la forme explicite pour :

$$\mathbf{G}_v = \frac{1}{m} \mathbf{R}_\theta \frac{\partial}{\partial \mathbf{u}} \left(\mathbf{M}_f(\mathbf{u}) - \mathbf{D}_f(\mathbf{u}) \|\mathbf{w}_r\| \mathbf{w}_{eq}^b \right) \Big|_{eq}, \quad (2.48)$$

peut être calculée comme dans (2.55), après factorisations.

De même, sur la base des matrices \mathbf{M}_m , \mathbf{D}_m dans (2.11), (2.13), nous pouvons calculer :

$$\mathbf{G}_\omega = \mathbf{J}^{-1} \frac{\partial}{\partial \mathbf{u}} \left(\mathbf{M}_m(\mathbf{u}) - \mathbf{D}_m(\mathbf{u}) \|\mathbf{w}_r\| \mathbf{w}_{eq}^b \right) \Big|_{eq} \quad (2.49)$$

comme dans (2.56), après factorisations.

Déterminons enfin l'expression de \mathbf{E}_v dans (2.34). Notons d'abord que nous pouvons écrire $\|\mathbf{w}_r\| \mathbf{w}_r = \mathbf{w}_r \sqrt{\mathbf{w}_r^\top \mathbf{w}_r}$, de sorte que :

$$\frac{\partial}{\partial \mathbf{w}_r} \|\mathbf{w}_r\| \mathbf{w}_r = \|\mathbf{w}_r\| \mathbb{I}_3 + \frac{\mathbf{w}_r \mathbf{w}_r^\top}{\|\mathbf{w}_r\|} = \|\mathbf{w}_r\| \left(\mathbb{I}_3 + \frac{\mathbf{w}_r \mathbf{w}_r^\top}{\mathbf{w}_r^\top \mathbf{w}_r} \right).$$

À l'aide de (2.36b) et (2.36d) et de l'expression de \mathbf{w}_r dans (2.35) et en suivant des calculs similaires aux cas précédents, nous obtenons l'expression (2.57) (indiquée dans l'Algorithme 2), pour $\mathbf{E}_v = -\frac{1}{m} \mathbf{R}_\theta \frac{\partial}{\partial \mathbf{w}_r} (\mathbf{D}_f(\mathbf{u}) \|\mathbf{w}_r\| \mathbf{w}_r) \Big|_{eq}$ et $\mathbf{E}_w = -\mathbf{J}^{-1} \frac{\partial}{\partial \mathbf{w}_r} (\mathbf{D}_m(\mathbf{u}) \|\mathbf{w}_r\| \mathbf{w}_r) \Big|_{eq}$, où nous rappelons que $\mathbf{D}_{m,eq} = \mathbf{D}_m(\mathbf{u}_{eq})$. \square

2.5 Conclusion

Le point important de ce chapitre est l'étude du modèle de drone convertible. La manipulation des équations, avec l'exposition des surfaces de contrôle virtuelles et l'obtention de la dynamique du drone, nous ont permis de clarifier le modèle, qui devient la base préliminaire pour toutes les lois de contrôle.

Ce chapitre décrit la modélisation d'un drone *tailsitter* avec notamment un modèle simplifié sans vent permettant une modification du vecteur de commande de manière à rendre le système linéaire en commande. Une étude poussée du modèle à basse vitesse permet aussi d'obtenir l'ensemble des points d'équilibre paramétrés par rapport à la vitesse du vent incident. Ces points d'équilibre permettent de vérifier la marge entre la commande à l'équilibre et les saturations des actionneurs. Cette marge est gage de manœuvrabilité. Des équilibres, il a été possible d'obtenir la dynamique linéarisée du drone en fonction des conditions de vent.

Il est donc possible d'utiliser la connaissance des équilibres et des dynamiques pour obtenir un contrôleur stabilisant le drone.

Algorithme 2 Détermination des matrices de la linéarisation de (2.34)

Entrées : Vecteur de vent $\mathbf{w}_{\text{eq}} = [w_x \ w_y \ w_z]^\top$ et
d'équilibre $(\mathbf{u}_{\text{eq}}, \mathbf{x}_{\text{eq}})$ provenant de (2.18) et de l'Algorithme 1.

Sorties : Matrices $\mathbf{A}_w, \mathbf{G}_w, \mathbf{E}_w$ dans (2.34)

- 1: Sélectionner les paramètres $\psi, \theta, \tau, \delta$ de (2.18) à l'aide de l'Algorithme 1 et de $\bar{\epsilon}_2 = \sin \frac{\theta}{2}, \bar{\eta} = \cos \frac{\theta}{2}$.
- 2: Avec les valeurs de (2.21), (2.12), (2.13), définissons :

$$\begin{aligned}\mathbf{R}_\psi &:= \begin{bmatrix} \cos \psi & \sin \psi & 0 \\ -\sin \psi & \cos \psi & 0 \\ 0 & 0 & 1 \end{bmatrix}, \quad \mathbf{R}_\theta := \begin{bmatrix} \cos \theta & 0 & \sin \theta \\ 0 & 1 & 0 \\ -\sin \theta & 0 & \cos \theta \end{bmatrix}, \\ \begin{bmatrix} w_{rx} \\ 0 \\ w_{rz} \end{bmatrix} &:= \mathbf{R}_\psi^\top \mathbf{w}_{\text{eq}}, \quad \begin{bmatrix} w_x^b \\ w_z^b \end{bmatrix} := \begin{bmatrix} w_{rx} \cos \theta - w_{rz} \sin \theta \\ w_{rz} \cos \theta + w_{rx} \sin \theta \end{bmatrix} \\ [\mathbf{D}_{f,\text{eq}} | \mathbf{D}_{m,\text{eq}}] &:= \frac{\rho S}{2} \begin{bmatrix} -C_d & 0 & C_d \xi_f \delta & 0 \\ 0 & 0 & 0 & \Delta_r C_\ell \xi_m \delta \\ -C_\ell \xi_f \delta & 0 & -C_\ell & 0 \\ 0 & 0 & 0 & 2\Delta_r C_\ell \end{bmatrix}\end{aligned}$$

- 3: Définissons les forces de portance et de trainée ainsi que leur dérivée par rapport à ϵ_2 (défini dans l'étape 1), comme :

$$\begin{bmatrix} f_d & \frac{\partial f_d}{\partial \epsilon_2} \\ 0 & \frac{\partial f_d}{\partial \epsilon_2} \\ f_\ell & \frac{\partial f_\ell}{\partial \epsilon_2} \end{bmatrix} := -\|\mathbf{w}_{\text{eq}}\| \mathbf{D}_{f,\text{eq}} \begin{bmatrix} w_x^b \left(4\bar{\eta} - \frac{2\bar{\epsilon}_2^2}{\bar{\eta}}\right) w_{rz} - 8\bar{\epsilon}_2 w_{rx} \\ 0 \\ w_z^b \left(4\bar{\eta} - \frac{2\bar{\epsilon}_2^2}{\bar{\eta}}\right) w_{rx} - 8\bar{\epsilon}_2 w_{rz} \end{bmatrix}, \quad (2.50)$$

- 4: Définissons les forces des moteurs et des élevons comme :

$$\begin{bmatrix} f_m \\ f_e \end{bmatrix} := \begin{bmatrix} \left(\frac{S_{\text{wet}} C_d}{2S_p} - 2\right) \tau \\ -\frac{S_{\text{wet}} \tau \delta \xi_f C_\ell}{2S_p} \end{bmatrix} \quad (2.51)$$

- 5: Sélectionnons les matrices \mathbf{A}_w dans (2.34) comme :

$$\mathbf{A}_{vv} = \frac{\|\mathbf{w}_{\text{eq}}\|}{m} \mathbf{R}_\theta \mathbf{D}_{f,\text{eq}} \mathbf{R}_\theta^\top \quad (2.52)$$

$$\begin{bmatrix} \mathbf{A}_{v\epsilon}^{1,2} \\ \mathbf{A}_{v\epsilon}^{2,1} \\ \mathbf{A}_{v\epsilon}^{2,3} \\ \mathbf{A}_{v\epsilon}^{3,2} \end{bmatrix} := \begin{bmatrix} 2\bar{\eta} - \frac{\bar{\epsilon}_2^2}{\bar{\eta}} & 4\bar{\epsilon}_2 & 2\bar{\epsilon}_2^2 - 1 & 2\bar{\epsilon}_2 \bar{\eta} \\ -2\bar{\eta} & -2\bar{\epsilon}_2 & 0 & 0 \\ 2\bar{\epsilon}_2 & -2\bar{\eta} & 0 & 0 \\ -4\bar{\epsilon}_2 & 2\bar{\eta} - \frac{\bar{\epsilon}_2^2}{\bar{\eta}} & -2\bar{\epsilon}_2 \bar{\eta} & 1 - 2\bar{\epsilon}_2^2 \end{bmatrix} \begin{bmatrix} f_e + f_\ell \\ f_m + f_d \\ \frac{\partial f_d}{\partial \epsilon_2} \\ \frac{\partial f_\ell}{\partial \epsilon_2} \end{bmatrix}$$

$$\mathbf{A}_{v\epsilon} = \frac{1}{m} \begin{bmatrix} 0 & \mathbf{A}_{v\epsilon}^{1,2} & 0 \\ \mathbf{A}_{v\epsilon}^{2,1} & 0 & \mathbf{A}_{v\epsilon}^{2,3} \\ 0 & \mathbf{A}_{v\epsilon}^{3,2} & 0 \end{bmatrix}, \quad \mathbf{A}_{\epsilon\omega} = \frac{\bar{\eta}}{2} \mathbb{I}_3 + \frac{\bar{\epsilon}_2}{2} \begin{bmatrix} 0 & 0 & 1 \\ 0 & 0 & 0 \\ -1 & 0 & 0 \end{bmatrix} \quad (2.53)$$

$$\mathbf{A}_{\omega\epsilon} = \frac{\rho S C_\ell \Delta_r \|\mathbf{w}_{\text{eq}}\| (w_x^b - \xi_m \delta w_z^b)}{J_y \bar{\eta}} \begin{bmatrix} 0 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 0 \end{bmatrix} \quad (2.54)$$

6: Sélectionnons les matrices \mathbf{G}_w dans (2.34) comme :

$$\begin{aligned}\mathbf{G}_v &= \frac{1}{m} \mathbf{R}_\theta [\mathbf{G}_{v\tau} | \mathbf{G}_{v\delta}], \quad \mathbf{G}_{v\tau} := \begin{bmatrix} 1 - \frac{S_{\text{wet}} C_d}{4 S_p} \\ 0 \\ -\frac{S_{\text{wet}} C_\ell \xi_f \delta}{2 S_p} \end{bmatrix} \begin{bmatrix} 1 \\ 1 \end{bmatrix}^\top \\ \mathbf{G}_{v\delta} &:= \begin{bmatrix} -\frac{1}{4} \rho S C_d \xi_f \|w_{\text{eq}}\| w_z^b \\ 0 \\ -\frac{S_{\text{wet}} C_\ell \xi_f \tau}{2 S_p} + \frac{1}{4} \rho S C_\ell \xi_f \|w_{\text{eq}}\| w_x^b \end{bmatrix} \begin{bmatrix} 1 \\ 1 \end{bmatrix}^\top\end{aligned}\quad (2.55)$$

$$\begin{aligned}\mathbf{G}_\omega &= \mathbf{J}^{-1} [\mathbf{G}_{\omega\tau} \ \mathbf{G}_{\omega\delta}], \quad \mathbf{G}_{\omega\delta} := \frac{S_{\text{wet}} C_\ell \tau}{4 S_p} \begin{bmatrix} a_y \xi_f & -a_y \xi_f \\ \Delta_r \xi_m & \Delta_r \xi_m \\ 0 & 0 \end{bmatrix} + \\ &\quad + \frac{\rho S \|w_{\text{eq}}\| \xi_m}{4} \begin{bmatrix} a_y C_d w_x^b & -a_y C_d w_x^b \\ \Delta_r C_\ell w_x^b & \Delta_r C_\ell w_x^b \\ a_y C_\ell w_z^b & -a_y C_\ell w_z^b \end{bmatrix} \\ \mathbf{G}_{\omega\tau} &:= \begin{bmatrix} \frac{k_m}{k_f} + \frac{S_{\text{wet}}}{4 S_p} a_y \xi_f C_\ell \delta \\ 0 \\ p_y + \frac{S_{\text{wet}}}{4 S_p} a_y C_d \end{bmatrix} \begin{bmatrix} 1 \\ -1 \end{bmatrix}^\top + \begin{bmatrix} 0 \\ \frac{S_{\text{wet}}}{4 S_p} \Delta_r \xi_m C_\ell \delta \\ 0 \end{bmatrix} \begin{bmatrix} 1 \\ 1 \end{bmatrix}^\top\end{aligned}\quad (2.56)$$

7: Sélectionnons les matrices \mathbf{E}_w de (2.34) comme :

$$\left[\begin{array}{c} \mathbf{E}_v \\ \mathbf{E}_\omega \end{array} \right] = - \left[\begin{array}{c} \mathbf{A}_{vv} \\ \mathbf{J} \|w_{\text{eq}}\| D_{m,\text{eq}} \mathbf{R}_\theta^\top \end{array} \right] \left(\mathbb{I}_3 + \frac{\mathbf{R}_\psi^\top w_{\text{eq}} w_{\text{eq}}^\top \mathbf{R}_\psi}{w_{\text{eq}}^\top w_{\text{eq}}} \right) \quad (2.57)$$

Retourne : $\mathbf{A}_w, \mathbf{G}_w, \mathbf{E}_w$

CHAPITRE 3

Travail préliminaire : commande hybride de drone

Sommaire

3.1	Motivation	57
3.2	Contrôleur par retour d'état non-linéaire	58
3.3	Contrôleur par retour d'état linéaire	61
3.4	Conception d'une commande locale-globale	63
3.5	Conclusion	67

3.1 Motivation

En prenant en compte les capacités d'un *tailsitter*, il est légitime de se poser la question du mode de vol devant être utilisé pour rejoindre un point. Effectivement, le drone a la possibilité de se déplacer en stationnaire ou bien en vol d'avancement. Lors d'un déplacement en stationnaire, le drone est vertical donc il se retrouve fortement sujet aux perturbations. Il est donc nécessaire d'avoir une grande région d'attraction autour de la position d'équilibre pour assurer un rejet des perturbations et une stabilisation. Lors d'un vol d'avancement, le drone se trouve dans une configuration proche d'une aile volante. Ainsi, il se trouve moins perturbé par les turbulences, dans la mesure où la surface projetée d'aile impactée par le flux d'air est plus faible. Toutefois, le drone doit voler à une vitesse assez importante pour être dans cette configuration donc il ne peut pas maintenir une position, mais seulement effectuer des cercles autour.

Nous nous concentrerons dans ce chapitre uniquement sur le vol stationnaire. Il s'agit donc de proposer une stratégie de commande pour stabiliser le drone en position stationnaire. Cette stratégie repose sur une dynamique discrète définissant l'usage d'une première loi de commande non-linéaire présentée dans [Michieletto 2020] et qui fournit une grande région d'attraction et d'une seconde

loi basée sur la dynamique linéarisée et fournissant une convergence plus rapide vers la cible. Les deux contrôleurs sont réunis par un mécanisme hybride qui permet de conserver les performances en régime permanent de la conception linéarisée, avec la grande région d'attraction garantie par la conception non-linéaire. Nous qualifions le contrôleur d'*hybride* car il possède deux dynamiques : une dynamique continue qui assure le bouclage (il s'agit précisément de deux lois de commande qui sont exécutées simultanément) et d'une dynamique discrète, laquelle sélectionne le contrôleur le plus adapté en fonction de la situation.

Notre solution est testée en simulant le modèle non-linéaire complet.

Dans cette partie, nous allons nous concentrer sur la stabilisation stationnaire du drone. Ainsi, nous nous appuyons sur la dynamique simplifiée décrite dans la section 2.1.2, avec la simplification $w = 0$ qui permet d'obtenir la dynamique simplifiée sans vent (2.15).

3.2 Contrôleur par retour d'état non-linéaire

Nous illustrons dans cette section une loi de contrôle dynamique non-linéaire inspirée du résultat de [Michieletto 2020] et présentée en Figure 3.1. Pour que cette loi de contrôle non-linéaire soit applicable, les matrices \mathbf{F} et \mathbf{M} mentionnées dans (2.15) doivent permettre de définir une direction dite de zéro moment $\bar{\mathbf{u}} \in \mathbb{R}^4$ garantissant $|\mathbf{F}\bar{\mathbf{u}}| = 1$ et $\mathbf{M}\bar{\mathbf{u}} = 0$, et la matrice inverse à droite \mathbf{M}^r de \mathbf{M} doit satisfaire $\mathbf{MM}^r = \mathbb{I}_3$ et $\mathbf{FM}^r = 0$.

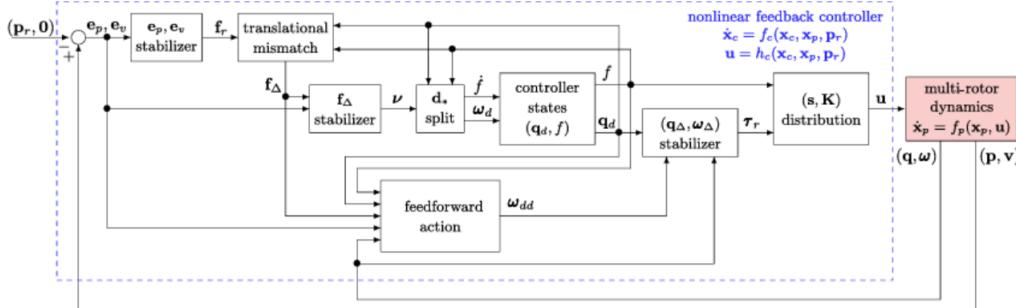


FIGURE 3.1 – Schéma de commande non-linéaire basée sur une direction de zéro moment [Michieletto 2020].

Dans notre cas, nous constatons que la direction du moment zéro $\bar{\mathbf{u}} = \frac{\sqrt{2}}{2a_f} [1 \ 1 \ 0 \ 0]^T$ satisfait les conditions, alors que le fait que $\text{rank}(\mathbf{F}) = 2$ (donc que le noyau de \mathbf{F} ($\ker \mathbf{F}$) soit de dimension 2) rend impossible l'obtention de la matrice inverse à droite \mathbf{M}^r de \mathbf{M} , laquelle est entièrement contenue dans $\ker \mathbf{F}$.

Nous déterminons \mathbf{M}^r en paramétrant (de manière conservatrice) les pseudo-inverses à droite de \mathbf{M} comme $\mathbf{M}^r := \mathbf{K}\mathbf{M}^\top(\mathbf{K}\mathbf{M}^\top)^{-1}$, où la matrice $\mathbf{K} \in \mathbb{R}^{4 \times 4}$ est symétrique et satisfait $\mathbf{K}\mathbf{M}^\top \geq \mathbb{I}_3$ (pour assurer l'inversibilité). Avec ce paramétrage, le but est de minimiser la norme de $\mathbf{F}\mathbf{M}^r = \mathbf{F}\mathbf{K}\mathbf{M}^\top(\mathbf{K}\mathbf{M}^\top)^{-1}$, ce qui est bien obtenu en minimisant la norme de $\mathbf{F}\mathbf{K}\mathbf{M}^\top$, du fait que la contrainte sur $\mathbf{K}\mathbf{M}^\top \geq \mathbb{I}_3$ garantit que le facteur $(\mathbf{K}\mathbf{M}^\top)^{-1}$ ait une norme plus petite que 1. En effectuant un complément de Schur, cette minimisation est obtenue en résolvant le programme semi-défini suivant :

$$\min_{\mathbf{K}, \kappa} \kappa, \text{ sous condition : } \mathbf{K}\mathbf{M}^\top \geq \mathbb{I}_3, \begin{bmatrix} \kappa\mathbb{I}_3 & \mathbf{F}\mathbf{K}\mathbf{M}^\top \\ \mathbf{K}^\top \mathbf{F}^\top & \kappa\mathbb{I}_3 \end{bmatrix} \geq 0, \text{ kappa} > 0,$$

lequel minimise κ tout en assurant $\mathbf{F}\mathbf{K}\mathbf{M}^\top \mathbf{K}^\top \mathbf{F}^\top \leq \kappa^2 \mathbb{I}_3$. En résolvant cette optimisation, on obtient, pour les matrices spécifiques considérées,

$$\mathbf{K} = \begin{bmatrix} 0 & -737 & 171 & -171 \\ -737 & 0 & -171 & 171 \\ 171 & -171 & 1583.5 & -43.73 \\ -171 & 171 & -43.73 & 1583.5 \end{bmatrix}, \mathbf{M}^r = \begin{bmatrix} 0 & 0 & -3.19 \\ 0 & 0 & 3.19 \\ -4.51 & -27.75 & -1.48 \\ 4.51 & -27.75 & 1.48 \end{bmatrix}$$

conduisant à $\kappa = 39.7$. Avec cette sélection basée sur l'optimalité, la conception dynamique non-linéaire de [Michieletto 2020] peut être appliquée efficacement en obtenant des réponses qui sont presque impossibles à distinguer du cas entièrement découplé $\mathbf{F}\mathbf{M}^r = 0$. Il convient de noter qu'une approche similaire, négligeant essentiellement les termes supplémentaires agissant sur la dynamique de translation, est également suggérée dans l'étude [Hua 2013]. Sur la base du choix de \mathbf{M}^r et de $\bar{\mathbf{u}}$ décrit ci-dessus, en appliquant la loi de commande de [Michieletto 2020, eqn (19)], l'entrée \mathbf{u} devient :

$$\mathbf{u} = \mathbf{u}_{\text{nl}} := \mathbf{M}^r \boldsymbol{\tau}_r + \bar{\mathbf{u}} \mathbf{f}, \quad (3.1)$$

où $\boldsymbol{\tau}_r = -k_{ap}\boldsymbol{\epsilon}_\Delta - k_{ad}\boldsymbol{\omega}_\Delta + \boldsymbol{\omega} \times J\boldsymbol{\omega} + J\boldsymbol{\omega}_{dd}$ est le moment de référence et \mathbf{f} est la force dans la direction de zéro moment.

La sélection de \mathbf{M}^r basée sur l'optimalité peut être interprétée de manière intéressante lorsqu'on observe le produit $\mathbf{M}^r \boldsymbol{\tau}_r = \mathbf{M}^r [\tau_{r,x} \ \tau_{r,y} \ \tau_{r,z}]^t op.$

Premièrement, pour obtenir un moment $\tau_{r,z}$ autour de l'axe z_b , nous utilisons principalement l'action différentielle de la poussée ; deuxièmement, un moment $\tau_{r,y}$ autour de l'axe y_b est généré par une utilisation symétrique des deux volets, avec une grande efficacité ; enfin, un moment $\tau_{r,x}$ autour de l'axe x_b provient d'une utilisation différentielle des volets.

Enfin, par rapport à la solution proposée dans [Michieletto 2020], pour prendre

partiellement en compte les effets de saturation énoncés dans la section 2.2.2, le bouclage décrit dans [Michieletto 2020] a été augmenté d'une stratégie de saturation d'erreur ne permettant jamais à l'erreur de position \mathbf{e}_p utilisée dans [Michieletto 2020, eqn. (22)] de dépasser la valeur maximale de 3 mètres. Les autres gains de réglage nécessaires à la solution de [Michieletto 2020] ont été sélectionnés comme $k_{pp} = 0.5$, $k_{pd} = 1.2$, $k_{ap} = 0.08$, $k_{ad} = 0.1$ et $k_\Delta = 1$.

La figure 3.2 montre la réponse du système en termes de positions, d'orientations (deux lignes du haut) et d'efforts des actionneurs (deux lignes du bas) lorsque le système part de la condition initiale $\mathbf{x}(0) = [\mathbf{p}(0) \ \mathbf{v}(0) \ \mathbf{q}(0) \ \boldsymbol{\omega}_b(0)]^\top = [0 \ 0 \ 0 \ 0 \ 0 \ 0 \ 0.9140 \ 0.1134 \ -0.3728 \ 0.1134 \ 0 \ 0 \ 0]^\top$ avec une position d'équilibre cible de $\mathbf{p}_{\text{eq}} = [4 \ 5 \ 6]^\top$ et $\mathbf{q}_{\text{eq}} = [\frac{\sqrt{2}}{2} \ 0 \ -\frac{\sqrt{2}}{2} \ 0]^\top$.

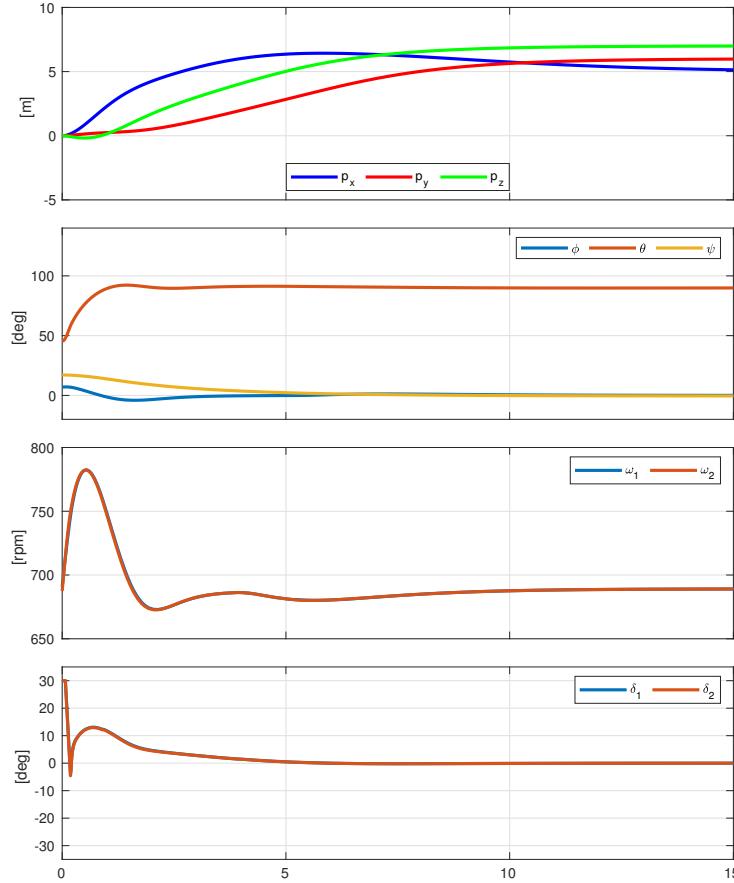


FIGURE 3.2 – Simulation de la loi de commande non-linéaire avec la dynamique de DarkO (2.2).

Une réponse adéquate peut être observée, laquelle reste assez éloignée des saturations des actionneurs (voir section 2.2.2). L'augmentation des gains peut accélérer

la réponse, mais produit toutefois des oscillations d'attitude indésirables. Il est donc intéressant de combiner ce contrôleur non-linéaire (qui fournit une grande région d'attraction) avec un contrôleur possédant une vitesse de convergence supérieure, conçu sur la base de la dynamique linéarisée sans vent (2.30).

3.3 Contrôleur par retour d'état linéaire

Sur la base des observations de la section précédente et étant donné une position cible correspondant à un équilibre $\mathbf{p}_{\text{eq}}, \mathbf{q}_{\text{eq}}$ tel que caractérisé dans l'équation (2.17), nous concevons ici un contrôleur par retour d'état linéaire capable de produire une réponse rapidement convergente. Pour cela, nous nous concentrerons sur la dynamique linéarisée (2.30) et proposons une loi de commande de la forme :

$$\mathbf{u}_{\text{lin}} := \mathbf{u}_{\text{eq}} - \mathbf{K}\tilde{\mathbf{x}}, \quad (3.2)$$

où $\tilde{\mathbf{x}}$ a été introduit dans (2.30) et $\mathbf{K} \in \mathbb{R}^{4 \times 12}$ est un gain de retour d'état qui peut être sélectionné, sur la base des matrices \mathbf{A}_0 et \mathbf{G}_0 apparaissant dans (2.30), de telle sorte que la boucle fermée du retour d'état $A_{\text{cl}} := \mathbf{A}_0 - \mathbf{G}_0 \mathbf{K}$ soit exponentiellement stable.

Dans notre cas, nous avons utilisé une sélection basée sur la commande linéaire quadratique, LQR, associée aux matrices de pondération $\mathbf{Q} = I_{12}$ et $\mathbf{R} = I_4$, qui donne une réponse en boucle fermée désirable. La conception LQR fournit également une matrice définie positive $\mathbf{S} \in \mathbb{R}^{12 \times 12}$ (solution de l'équation algébrique de Riccati) garantissant que $\mathbf{A}_{\text{cl}}^\top \mathbf{S} + \mathbf{S} \mathbf{A}_{\text{cl}} < 0$. Il est donc possible d'utiliser \mathbf{S} pour former une fonction de Lyapunov. En particulier, il est bien connu que la fonction $V(\tilde{\mathbf{x}}) = \tilde{\mathbf{x}}^\top \mathbf{S} \tilde{\mathbf{x}}$ est également une fonction de Lyapunov certifiant la stabilité exponentielle locale de \mathbf{x}_{eq} pour la dynamique non-linéaire. Plus précisément, il existe un scalaire positif $\bar{v} \in \mathbb{R}$ tel que, le long de la dynamique (2.2), nous avons :

$$V(\tilde{\mathbf{x}}) \leq \bar{v} \Rightarrow \dot{V}(\tilde{\mathbf{x}}) := \langle \nabla V(\tilde{\mathbf{x}}), \dot{\tilde{\mathbf{x}}} \rangle < 0, \quad (3.3)$$

pour tout $\tilde{\mathbf{x}} \neq 0$; en d'autres termes, le sous-ensemble de $V(\tilde{\mathbf{x}}) \leq \bar{v}$ est contenu dans le bassin d'attraction de l'équilibre \mathbf{x}_{eq} .

La détermination du plus grand scalaire \bar{v} assurant (3.3) est un problème complexe et des bornes inférieures conservatrices peuvent être déterminées en quantifiant l'effet des non-linéarités sur la dynamique. Puisque $\dot{\tilde{\mathbf{x}}}$ est une fonction de \mathbf{x} , il est assez facile d'évaluer algébriquement $\dot{V}(\tilde{\mathbf{x}})$ pour un grand nombre d'exactions aléatoires de la variable $\tilde{\mathbf{x}}$, afin d'obtenir une estimation probabiliste du plus grand

scalaire \bar{v} . Des garanties rigoureuses sur ces sélections peuvent être obtenues en appliquant les résultats de [Tempo 2012], mais une évaluation de 10000 échantillons a confirmé que la valeur $\bar{v} = 400$ est une sélection satisfaisante (3.3).

La Figure 3.3 montre une simulation commençant à l'origine, avec un drone vertical et des vitesses linéaires et angulaires initiales nulles. La position cible est $\mathbf{p}_{\text{eq}} = [4, 5, 6]$ avec une stabilisation en vol stationnaire (drone vertical) avec $\beta = 0$.

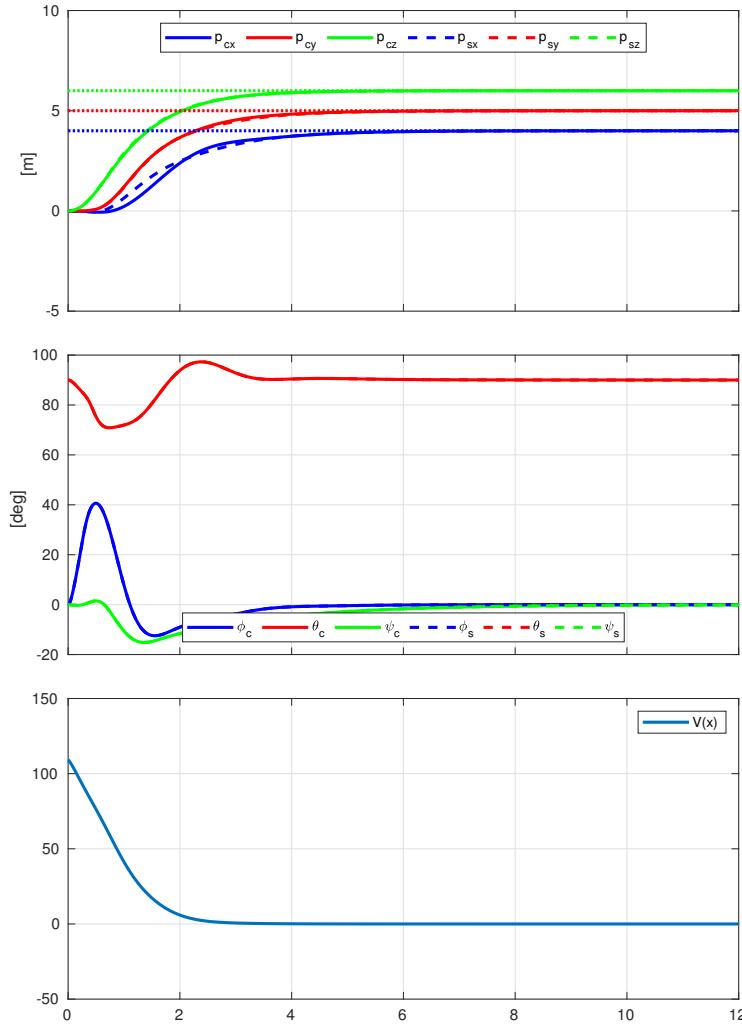


FIGURE 3.3 – Simulation du modèle complet (2.2) (ligne continue) et simplifié (2.15) (ligne en pointillé) avec $\mathbf{u} = \mathbf{u}_{\text{lin}}$ défini dans (3.2) et comme condition initiale $\tilde{\mathbf{x}}_0$ dans le bassin d'attraction.

La ligne en pointillé représente la position de la cible sur chaque axe. Le dernier graphique montre la décroissance exponentielle souhaitable de V . La Figure 3.3 illustre, à la fois, la simulation du modèle complet (ligne continue) (2.2) et du

modèle non-linéaire simplifié (2.15) (ligne en pointillé), ce qui met en évidence des différences dans la phase transitoire. Lorsqu'on fournit une position cible plus importante $\mathbf{p}_{\text{eq}} = [8, 9, 10]$ (avec la même orientation), la condition initiale se situe en dehors du bassin d'attraction et une divergence apparaît, comme le montre la figure 3.4.

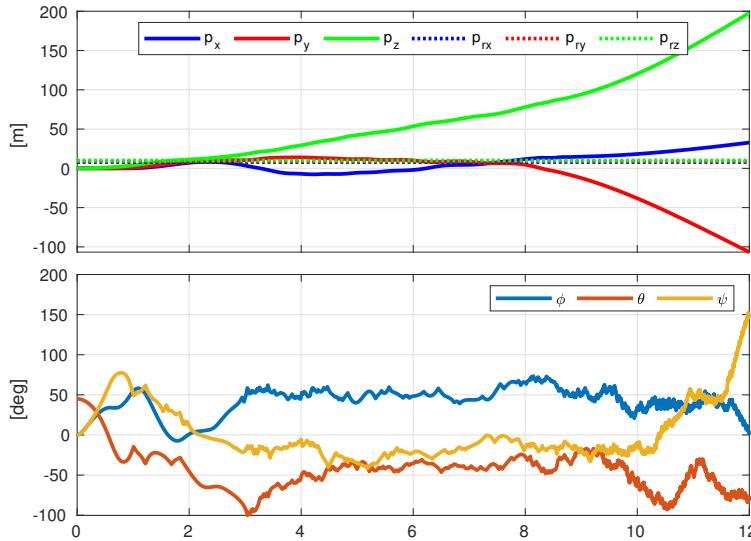


FIGURE 3.4 – Simulation divergente du modèle complet (2.2) avec $\mathbf{u} = \mathbf{u}_{\text{lin}}$ défini dans (3.2) et une condition initiale $\tilde{\mathbf{x}}_0$ en dehors du bassin d'attraction.

3.4 Conception d'une commande locale-globale

Considérant les stratégies locales-globales présentées dans [Goebel 2012, Ex. 1. 7], similaires à la solution présentée dans [Andreetto 2016], nous utilisons un mécanisme hybride. Il sélectionnera le contrôleur local avec une grande vitesse de convergence (3.2) (tant que l'état se trouve dans le bassin d'attraction de l'équilibre) ou le contrôleur non-linéaire plus lent (3.1), qui fournit une plus grande région d'attraction (et peut être appelé, par abus de langage, le "contrôleur global"). À cette fin, nous ajoutons à l'état du contrôleur une variable d'état logique $\ell \in \{0, 1\}$, qui régit le choix de l'entrée de contrôle entre (3.1) et (3.2) tel que :

$$\mathbf{u} = \mathbf{u}_{\text{hyb}} := \ell \mathbf{u}_{\text{nl}} + (1 - \ell) \mathbf{u}_{\text{lin}}. \quad (3.4)$$

Nous nous assurons, grâce à la dynamique hybride, que ℓ ne puisse prendre que des valeurs dans $\{0, 1\}$. Sa dynamique est définie par une évolution continue notée $\dot{\ell}$

dans \mathcal{C} et une évolution discrète notée ℓ^+ dans \mathcal{D} telle que :

$$\begin{cases} \dot{\ell} = 0, & \chi \in \mathcal{C} \\ \ell^+ = 1 - \ell, & \chi \in \mathcal{D} \end{cases}. \quad (3.5)$$

Ce formalisme mathématique représente un fonctionnement simple qui correspond à un maintien de la valeur de ℓ lors de l'évolution continue \mathcal{C} et à un changement de la valeur de ℓ de 1 vers 0 ou inversement lors des évolutions discrètes \mathcal{D} représentées par les croix rouges sur la Figure 3.5.

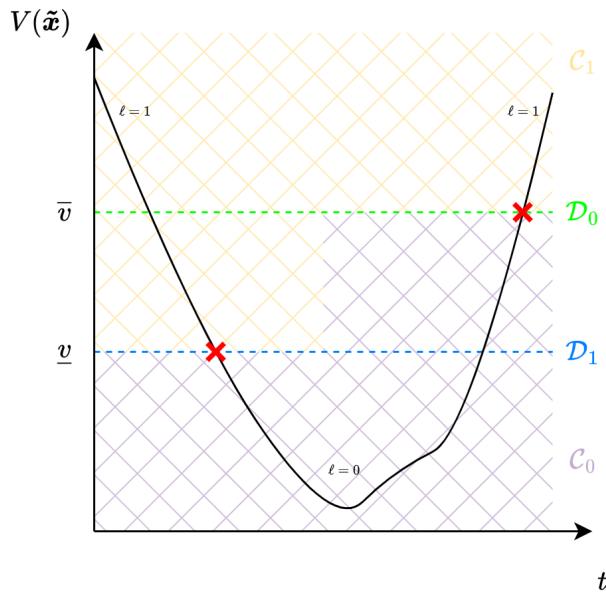


FIGURE 3.5 – Exemple de l'évolution de $V(\tilde{\mathbf{x}})$, de la variable discrète ℓ et des ensembles continus \mathcal{C}_0 et \mathcal{C}_1 et discrets \mathcal{D}_0 et \mathcal{D}_1 .

Nous avons l'état complet de la boucle fermée $\chi = [\mathbf{p}, \mathbf{v}, \mathbf{q}, \boldsymbol{\omega}, l]$ et les ensembles continus et discrets sont, respectivement, \mathcal{C} et \mathcal{D} définis par :

$$\begin{aligned} \mathcal{C} &:= \mathcal{C}_0 \cup \mathcal{C}_1, \quad \mathcal{D} := \mathcal{D}_0 \cup \mathcal{D}_1, \\ \mathcal{C}_0 &:= \{\chi \in \mathbb{R}^{14} : V(\tilde{\mathbf{x}}) \leq \bar{v} \text{ and } \ell = 0\} \\ \mathcal{C}_1 &:= \{\chi \in \mathbb{R}^{14} : V(\tilde{\mathbf{x}}) \geq \underline{v} \text{ and } \ell = 1\} \\ \mathcal{D}_0 &:= \{\chi \in \mathbb{R}^{14} : V(\tilde{\mathbf{x}}) \geq \bar{v} \text{ and } \ell = 0\} \\ \mathcal{D}_1 &:= \{\chi \in \mathbb{R}^{14} : V(\tilde{\mathbf{x}}) \leq \underline{v} \text{ and } \ell = 1\} \end{aligned}$$

où $V(\tilde{\mathbf{x}}) := \tilde{\mathbf{x}}^\top S \tilde{\mathbf{x}}$ a été définie dans la section précédente, $\bar{v} = 400$ a été déterminée

pour satisfaire (3.3) et \underline{v} est toute constante positive satisfaisant $\underline{v} < \bar{v}$.

L'intérêt de \underline{v} et de \bar{v} est de permettre une hystérésis sur le changement local/global de loi de commande de manière à éviter les oscillations entre les deux lois à proximité de la transition. Ce fonctionnement est représenté sur la Figure 3.6. Il retarde toutefois le changement de loi de commande.

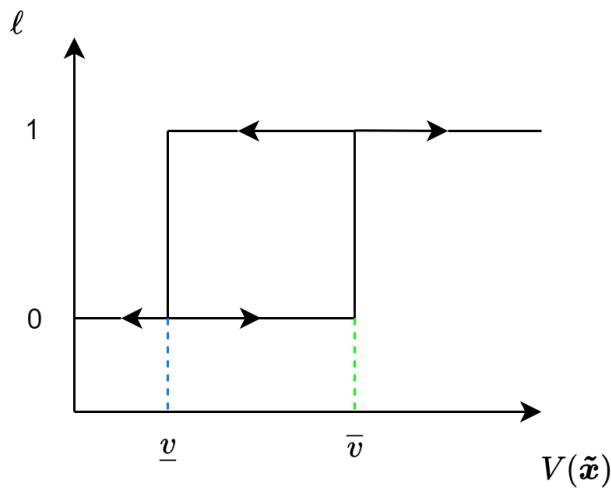


FIGURE 3.6 – Mécanisme d'hystérésis sur le changement local/global de loi de commande.

Dans notre cas, nous choisissons $\underline{v} = 350$. La Figure 3.5 présente un exemple du fonctionnement du mécanisme hybride en fonction de l'évolution de $V(\tilde{\mathbf{x}})$.

Le résultat suivant est une conséquence immédiate des résultats de [Goebel 2012, Ex. 1.7] et des propriétés de nos modèles linéaires et non-linéaires.

Proposition 1. *Avec l'action du bouclage hybride (3.4), la boucle fermée présente, quand elle utilise le contrôleur linéaire (3.2), le même bassin d'attraction que celui associé au contrôleur non-linéaire (3.1).*

Nous avons réalisé plusieurs simulations de la boucle fermée à l'aide de la *Hybrid Equations Toolbox* (HyEQ) de Matlab [Sanfelice 2013]. Les simulations sont effectuées avec le modèle complet du drone (2.2), comprenant tous les effets aérodynamiques non-linéaires. Un exemple de simulation est présenté dans la Figure 3.7, où nous initialisons le drone à l'origine avec une orientation nulle, sauf pour l'angle de tangage fixé à 45 degrés. L'orientation de la cible est en configuration de vol stationnaire vertical et la position de la cible est assignée à $\mathbf{p}_{\text{eq}} = [50, 25, 12.5]$.

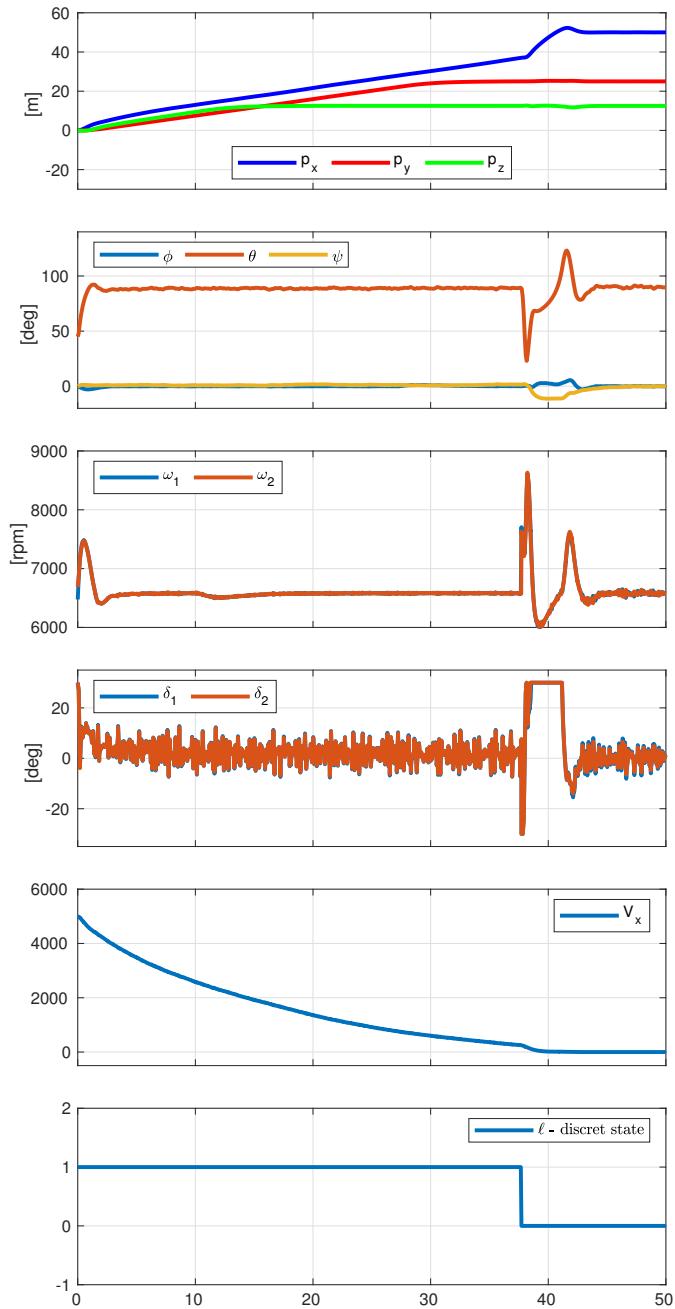


FIGURE 3.7 – Simulation en boucle fermée avec le contrôleur hybride (3.4).

Nous observons que sur la période $t \in [0, 38]$, le drone présente une convergence élégante mais lente vers la position cible souhaitée, en utilisant le contrôleur global ($\ell = 1$). Dès lors, l'état discret ℓ entre dans l'ensemble \mathcal{D}_1 et le contrôleur local est activé jusqu'à la convergence vers l'équilibre souhaité, avec une vitesse plus élevée.

Pour obtenir des simulations réalistes, les mesures sont affectées par le bruit des capteurs. La robustesse intrinsèque de la rétroaction hybride, établie dans [Goebel 2012, Chapitre 7], est confirmée par le maintien des performances, malgré le bruit de mesure.

3.5 Conclusion

L'objectif principal de ce chapitre est de proposer un mécanisme hybride capable d'utiliser deux modes de contrôle d'un drone convertible, pour naviguer entre deux points.

À partir du modèle simplifié décrit dans le chapitre 2, nous avons proposé deux lois de contrôle, basées sur une linéarisation du modèle et sur un contrôle non-linéaire. Pour le vol stationnaire, nous avons utilisé une linéarisation autour d'un point d'équilibre sans vent, afin d'obtenir une loi de commande linéaire par retour d'état. L'avantage de cette dernière réside dans sa capacité à optimiser le retour d'état et donc à ajuster sa vitesse de convergence et à rejeter des perturbations. Cependant, son domaine de stabilité est réduit, car il s'agit d'une loi de commande locale, valable dans le voisinage du point d'équilibre. Au contraire, la seconde loi proposée, non-linéaire, est globale. Celle-ci, stable sur l'ensemble du domaine de vol, ne permet cependant pas d'optimiser la vitesse de convergence, compte tenu de sa complexité.

Compte tenu des avantages et des inconvénients observés dans les lois de commande, il a semblé approprié d'envisager une transition hybride sur la base de la stabilité de la boucle. En d'autres termes, le contrôleur non-linéaire est utilisé là où le contrôleur linéaire est instable, afin de faire converger le drone vers la cible. Dès que le drone entre dans le domaine de stabilité du contrôleur linéaire, le mécanisme hybride prend en charge le changement de loi, en utilisant la vitesse de convergence maximale autorisée par l'optimisation linéaire. L'avantage de ce mécanisme est qu'il fournit un contrôleur ayant autorité sur l'ensemble du domaine de vol, assurant la convergence, tout en optimisant les capacités du drone. Bien que les premiers résultats ne soient obtenus que par simulation, ils semblent prometteurs pour une intégration sur l'architecture réelle.

Toutefois, ce travail ne prend pas en compte l'impact des turbulences ou du vent sur l'architecture *tailsitter*. Il est donc intéressant de se focaliser sur des mécanismes de stabilisation d'un drone face à des conditions de vol plus réalistes. L'ensemble de ces études sera mené dans le contexte d'un contrôleur local.

CHAPITRE 4

Commande robuste longitudinale d'une maquette

Sommaire

4.1	Motivation	69
4.2	Présentation de la maquette expérimentale	70
4.2.1	Description physique, capteurs et actionnements	70
4.2.2	Simulation des mouvements du drone	73
4.3	Contrôle linéaire, architecture PI sur maquette à 3 DOF	74
4.3.1	Description du schéma de contrôle	74
4.3.2	Schéma de synthèse H_∞	76
4.4	Résultats	78
4.5	Conclusion	79

4.1 Motivation

Les simulations en boucle fermée, avec le contrôleur (3.4) développé dans la section 3.4, montrent qu'en présence d'un vent horizontal constant dans le plan $(x_{[b]}, z_{[b]})$, le drone modifie son angle de tangage. Ce comportement a également été observé lors d'essais en soufflerie avec le dispositif expérimental [Olszanecki Barth 2020]. Intuitivement, une réduction de l'angle d'attaque entraîne une diminution de la surface exposée au vent, de manière à réduire la force de traînée, ce qui a une forte incidence sur la position. Dans le même temps, le flux d'air dû au vent constant génère une portance, compensée par une réduction de la poussée de l'hélice, et une réduction conséquente de la consommation du drone. L'objectif de la maquette décrite ici est d'évaluer expérimentalement l'effet du vent sur le dispositif DarkO.

De plus, il faut noter que le changement de l'angle d'attaque du drone a un impact sur la mesure du vent. Comme la sonde de Pitot est fixée sur le corps du

drone, cette dernière se trouve être en rotation lors de la transition. On comprend donc que la mesure du vent ne sera valide qu'en vol d'avancement, à haute vitesse. En vol stationnaire ou lors de la transition, nous n'avons pas de mesure du vent, ni des rafales impactant le drone.

Nous allons donc proposer dans ce chapitre une méthode de commande linéaire n'utilisant pas de mesure de vent pour stabiliser le drone dans une position de l'espace. Toutefois, nous commencerons par une stabilisation longitudinale sur une dynamique à trois degrés de liberté pour tester notre loi de commande. Cette loi de commande est locale et n'est valide que dans un environnement proche du point d'équilibre. Ce bouclage pourrait être utilisé en remplacement de la loi LQR proposée dans la section 3.3.

4.2 Présentation de la maquette expérimentale

4.2.1 Description physique, capteurs et actionnements

Le prototype développé comprend des pièces imprimées en 3D, en Onyx et PLA (acide polylactique, un polyester thermoplastique). Le drone est spécialement conçu pour réaliser des expériences devant une soufflerie, avec un comportement semblable à celui de DarkO en raison de leur forme similaire (voir Figure 4.2). La partie centrale, qui contient l'avionique embarquée (pilote automatique, GPS, etc.) dans DarkO, a été remplacée ici par un joint tournant à un degré de liberté (voir Figure 4.1). Les ailes sont les mêmes que celles du DarkO, avec les contrôleurs électroniques de vitesse (ESC), régissant la vitesse du moteur *brushless*, placés dans les ailes.

Comme décrit dans la section 4.1, nous souhaitons représenter et étudier le degré de liberté de l'axe y_b du drone DarkO. Le tube principal en carbone reliant les deux ailes est utilisé comme axe de rotation. Ce tube est fixé sur deux roulements espacés de 28.5 mm afin d'obtenir une fixation solide de l'ensemble. Cet axe de rotation est équipé d'un codeur optique rotatif en quadrature pour mesurer précisément l'orientation de l'appareil. L'avantage de ce capteur est qu'il ne produit pas de couple résistant sur l'axe de rotation. Ce codeur offre 4000 impulsions par tour, ce qui donne une résolution de $0.09^\circ / \text{impulsion}$.

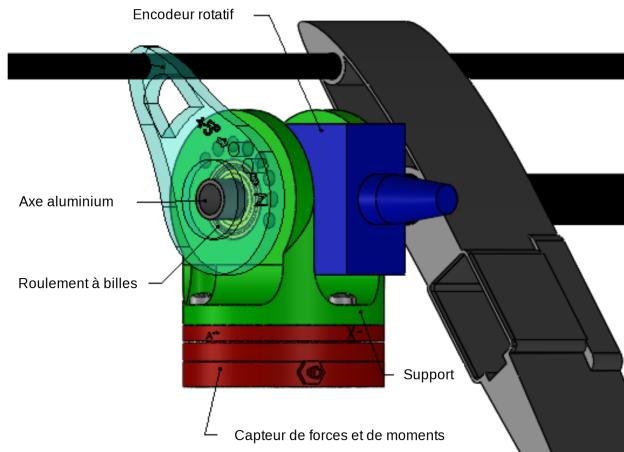


FIGURE 4.1 – Montage à un degré de liberté.

Comme le montre la Figure 4.1, l’indexeur et le support sont percés pour que la rotation puisse être bloquée, par une vis, à des positions connues (0° , 90° , etc.). Le verrouillage de l’appareil permet une initialisation correcte de l’encodeur incrémental. Le verrouillage permet également de placer l’appareil dans des positions exactes spécifiques afin d’identifier les coefficients aérodynamiques.

Le mécanisme est également équipé d’un capteur de forces et de moments à 6 degrés de liberté (DOF), qui permet de mesurer la force exercée sur le dispositif expérimental par le support. Le banc d’essai expérimental est également équipé d’un fil chaud pour mesurer la vitesse de l’air affectant la maquette.

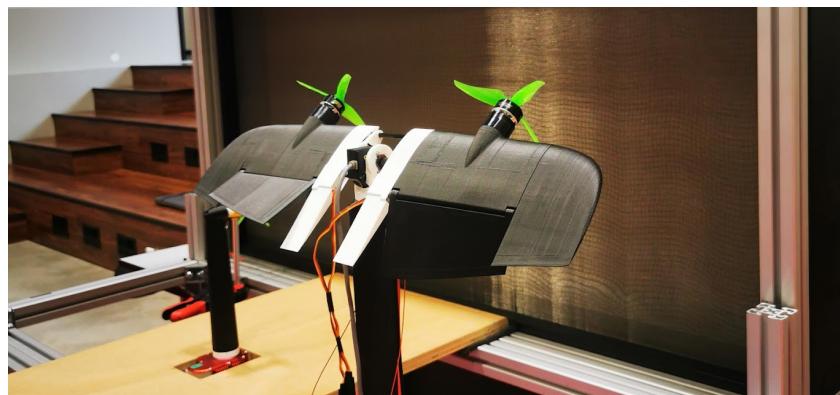


FIGURE 4.2 – Modèle de DarkO à un seul degré de liberté devant le *WindShape*.

La photo de la Figure 4.2 montre le dispositif expérimental dans son environnement de test. Le drone est placé devant une soufflerie ouverte, appelée *WindShape*, qui génère un vent horizontal compris entre 2 et 16 ms^{-1} . Ainsi, lors de nos tests,

nous considérons que la composante verticale du vent est nulle. Le drone est placé au centre du *WindShape*, dans la zone d'écoulement la plus laminaire, tandis que le capteur à fil chaud est placé aussi près que possible du drone.

La géométrie du dispositif expérimental permet de placer les câbles d'alimentation et de signal près du centre de rotation afin de minimiser leurs effets de friction sur la structure. Malgré cela, le système de rotation interfère inévitablement avec le drone, en créant des forces parasites, notamment de la traînée. La surface projetée de l'articulation étant faible par rapport à la surface de l'aile, la traînée générée par ce support est faible par rapport à la traînée de l'aile et des hélices, et peut donc être négligée. Un diagramme schématique des composants du dispositif expérimental et de leurs interconnexions est présenté à la Figure 4.3.

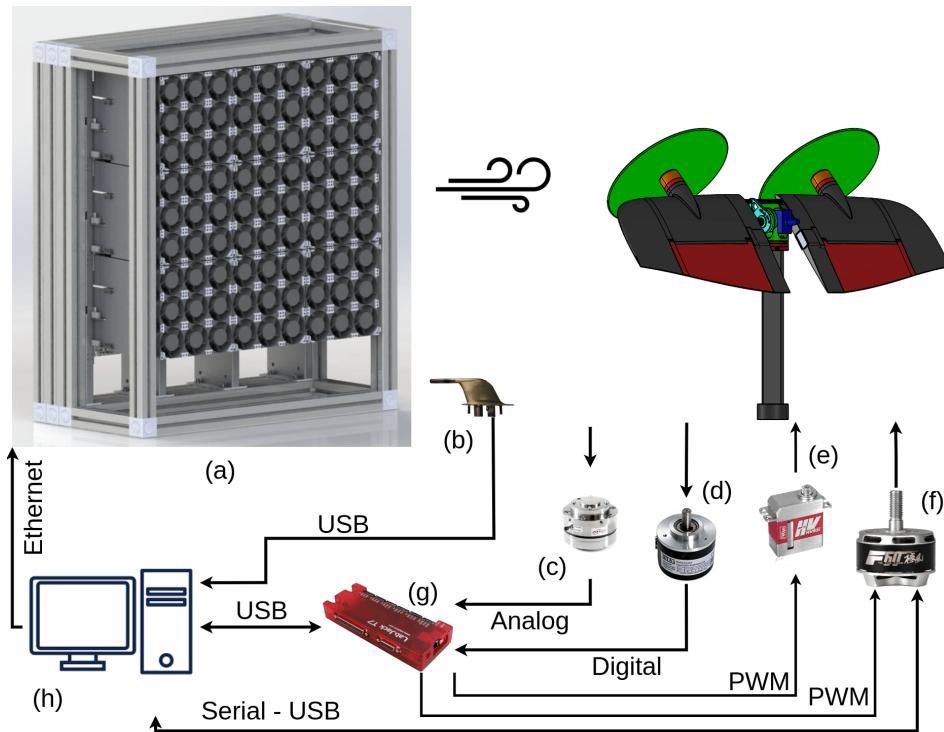


FIGURE 4.3 – Architecture d'essai en vol virtuel : *WindShape* (a) ; capteur de vitesse (b) ; capteur de forces et moments (c) ; encodeur rotatif (d) ; servomoteur (e) ; moteur *brushless* + ESC (f) ; LabJack (g) ; ordinateur de contrôle (h)

Les moteurs (f) sont alimentés par une batterie externe de 12v 20Ah et les servomoteurs (e) sont alimentés par du 5 V, via un module d'acquisition LabJack T7 [LabJack 2001,] (g). Le module LabJack (g) concentre la plupart des signaux des capteurs et des actionneurs : six entrées analogiques pour le capteur de force/couple

(c), deux entrées numériques en quadrature pour l'encodeur rotatif (d), une entrée analogique (ou une liaison série selon le capteur) pour le capteur de vitesse (b), deux sorties numériques PWM (Pulse Width Modulation) pour les moteurs (f) et deux sorties numériques PWM pour les servomoteurs (e).

Les élévons sont commandés par des servomoteurs qui ne fournissent pas de signal de mesure de la position. Nous utilisons donc le point de consigne, en supposant que les actionneurs soient parfaits. Cela est raisonnable en raison de la saturation logicielle imposée à l'entrée des élévons et du dimensionnement correct des servomoteurs par rapport aux forces impliquées. Le LabJack (g) possède une interface de programmation d'application (API), permettant une connexion avec un ordinateur. Nous avons développé un code Python qui communique avec le LabJack afin de récupérer les valeurs des capteurs, de calculer la commande à appliquer aux servomoteurs selon le schéma de contrôle présenté ci-dessous et de générer les signaux de sortie pour les servomoteurs. Les données collectées par le LabJack sont enregistrées afin d'être utilisées pour le post-traitement et de générer le graphique présenté dans la section 4.4.

Le *WindShape* dispose également d'une API lui permettant d'être contrôlé via un réseau Ethernet. Le code Python développé peut assigner la vitesse du vent généré par le *WindShape* et donc agir sur le modèle. Il est ainsi possible de tester un ensemble de configurations de vols stationnaires et les transitoires associés dans la même campagne d'essais, sans aucune action sur le modèle.

4.2.2 Simulation des mouvements du drone

Le prototype étant relié à un support fixe, il n'est pas possible de reproduire expérimentalement le mouvement de translation. Nous avons donc inclus une simulation logicielle du mouvement en intégrant les mesures de forces disponibles au niveau de la fixation. La vitesse de translation (respectivement la position) du drone est obtenue par intégration simple (respectivement double) des données mesurées par le capteur de forces. Par souci de simplicité, nous négligeons l'influence aérodynamique de la vitesse (simulée) sur l'aile. En particulier, à partir des équations (2.2a) et (2.2b), nous obtenons le modèle simplifié suivant :

$$\dot{\mathbf{v}} = \mathbf{g} + \frac{1}{m} \left(R(\mathbf{q})(F\mathbf{u} + D_f(\delta)R^\top(\mathbf{q})\|\mathbf{w}\|\mathbf{w}) \right) \quad (4.1a)$$

$$= \mathbf{g} + \frac{1}{m} \mathbf{F}_{meas}, \quad (4.1b)$$

où \mathbf{F}_{meas} représente les forces mesurées par le capteur dans le repère inertiel corrigé du biais. Pour calibrer la correction du biais, lors de l'initialisation, les forces mesurées sont moyennées sur 6000 échantillons, le modèle étant bloqué dans une position stable (angle de tangage à 0° , c'est-à-dire orientation verticale). Pour éliminer le biais de la force mesurée à chaque mesure, nous soustrayons l'effet de la gravité sur le modèle de la mesure. Une masse artificielle m est attribuée à la dynamique du logiciel dans la boucle conformément à (4.1b), ce qui permet de tester plusieurs configurations afin de mieux apprécier l'influence de la masse du drone sur d'éventuels événements transitoires de saturation. Cela permet d'étudier des scénarios impliquant la masse non négligeable de la batterie, qui n'est pas présente dans notre modèle. Bien que cette manipulation soit aisée, elle ne représente pas parfaitement la réalité car nous ne tenons pas compte de la répartition des masses dans le drone et donc des modifications de l'inertie. La vitesse et la position transitoire du drone sont ensuite obtenues par intégration numérique simple et double de l'accélération comme dans (4.1), en utilisant une intégration numérique trapézoïdale.

4.3 Contrôle linéaire, architecture PI sur maquette à 3 DOF

Dans la section 3.4, nous avons proposé un bouclage proportionnel stabilisant une position de vol stationnaire en l'absence de vent (sans perturbation). Nous proposons ici une extension incluant une action intégrale, adaptée au fonctionnement avec une perturbation non mesurée représentée par un vent constant. L'objectif est de stabiliser le drone à la position de référence, en rejetant une perturbation de vent constant inconnue. Les matrices de gain associées au gain intégral et proportionnel seront optimisées suivant un schéma de synthèse H_∞ combinant à la fois des critères de robustesse et de performance. Nous mettons donc en œuvre une synthèse robuste basée sur des critères H_∞ , sous contrainte d'une structure PI multidimensionnelle (MIMO).

4.3.1 Description du schéma de contrôle

Nous expérimentons la situation avec le vent agissant uniquement le long de l'axe x_b , avec le drone orienté vers le vent, c'est-à-dire avec des angles de roulis et de lacet nuls. Dans cette configuration, le vent n'agit que sur la vitesse linéaire le long des axes x_b et z_b , et ne génère qu'un moment autour de l'axe y_b . Un examen attentif de

la commande et des matrices d'entrée des perturbations \mathbf{F} , \mathbf{M} dans (2.16) suggère une architecture de commande efficace pour rejeter une perturbation constante. En effet, les élévons et les hélices peuvent être utilisés symétriquement pour générer respectivement un moment autour de l'axe y_b et une force le long de l'axe x_b , compensant ainsi l'effet de la perturbation. Néanmoins, il reste une force le long de l'axe z_b à compenser et une action intégrale peut converger asymptotiquement vers la force désirée, même avec une perturbation du vent non mesurée \mathbf{w} . Nous pouvons ainsi stabiliser le drone à une position de vol stationnaire, différente de l'équilibre sans vent. La solution de contrôle exploite le degré de liberté de l'angle de tangage pour compenser l'effet du vent.

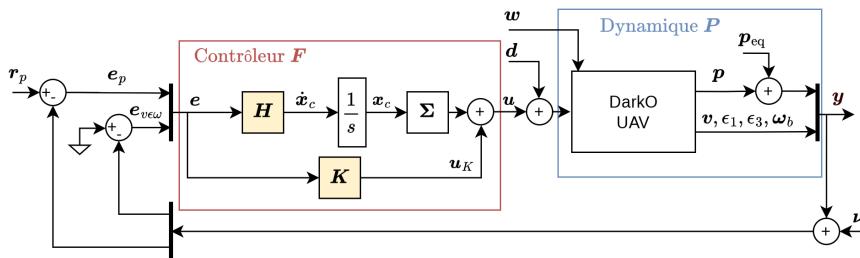


FIGURE 4.4 – Schéma de commande linéaire, proportionnelle-intégrale.

Le contrôleur proposé, représenté sur la Figure 4.4, correspond à :

$$\mathbf{e}_p = \mathbf{r}_p - \mathbf{p}, \quad (4.2a)$$

$$\mathbf{e} = [\mathbf{e}_p^\top \mathbf{e}_{v\omega}^\top]^\top, \quad \dot{\mathbf{x}}_c = \mathbf{H}\mathbf{e}, \quad \mathbf{u} = \Sigma \mathbf{x}_c + \mathbf{u}_K, \quad (4.2b)$$

$$\Sigma := \begin{bmatrix} 1 & 1 & 0 & 0 \\ 0 & 0 & 1 & 1 \end{bmatrix}^\top, \quad \mathbf{u}_K = \mathbf{K}\mathbf{e}, \quad (4.2c)$$

où $\mathbf{x}_c \in \mathbb{R}^2$ est l'état de l'intégrateur ; $\mathbf{r}_p \in \mathbb{R}^3$ est la référence constante comprenant une position cible pour le mouvement de translation ; Σ est une matrice d'allocation d'entrée qui permet d'affecter la première composante de l'état de l'intégrateur à la commande du moteur et la seconde composante à la commande de la gouverne de profondeur. \mathbf{K} , \mathbf{H} sont des gains constants à sélectionner pour que la matrice linéaire de la boucle fermée \mathbf{A}_{cl} caractérisant la boucle fermée linéaire soit Hurwitz, afin d'assurer la stabilisation avec la dynamique linéarisée liée au scénario sans vent (2.15).

De manière synthétique, la matrice (4.3) décrit la boucle fermée illustrée à la Figure 4.4 avec (4.2) : un retour de sortie avec 11 sorties, comprenant les trois positions, les trois vitesses linéaires, deux des trois angles (ϵ_1 et ϵ_3) et les trois

vitesses angulaires.

$$\mathbf{A}_{cl} = \begin{bmatrix} \mathbf{A} & \mathbf{0}_{12 \times 2} \\ \mathbf{H} & \mathbf{0}_{2 \times 2} \end{bmatrix} - \begin{bmatrix} \mathbf{G} \\ \mathbf{0}_{2 \times 4} \end{bmatrix} \left(\mathbf{K} - \begin{bmatrix} \mathbf{0}_{4 \times 12} & \mathbf{\Sigma} \end{bmatrix} \right), \quad (4.3)$$

Cette structure est une structure proportionnelle-intégrale MIMO résultant d'une observation attentive de la dynamique linéarisée du drone, ce qui permet un nombre minimal d'intégrateurs dans le contrôleur. Ce contrôle devrait permettre de rejeter les perturbations constantes tout en ayant une robustesse satisfaisante. Le gain \mathbf{K} correspond au terme proportionnel et le gain \mathbf{H} pondère le terme intégral, induisant une convergence vers la cible. La matrice d'allocation $\mathbf{\Sigma}$ conduit à une utilisation symétrique des hélices et des ailerons. Il faut alors ajuster \mathbf{K} et \mathbf{H} pour obtenir un compromis satisfaisant entre robustesse et rejet des perturbations. Nous mettons en œuvre une synthèse multiobjectif basée sur des contraintes H_∞ , décrite ci-après.

4.3.2 Schéma de synthèse H_∞

Pour effectuer une sélection robuste de \mathbf{K} et \mathbf{H} , nous introduisons des matrices de transfert qui correspondent aux objectifs de robustesse à partir de la Figure 4.4.

La sortie de mesure \mathbf{y} est utilisée pour la rétroaction, l'entrée \mathbf{u} est la somme de l'entrée intégrale $\mathbf{\Sigma}x_c$ et de l'action proportionnelle $\mathbf{K}\mathbf{e}$. La sortie \mathbf{z} correspond aux signaux de performance de sortie à contrôler ($\mathbf{e}, \mathbf{w}, \mathbf{u}, \mathbf{y}, \mathbf{r}_p$).

Nous notons la marge de module d'une matrice de transfert $s \mapsto T_{v \rightarrow z}$ comme $\Delta_m(T_{v \rightarrow z}) = \min_{\omega \in R} \sigma_{\min}(T_{v \rightarrow z}(j\omega))$. La fonction de sensibilité en sortie est définie par $T_{v \rightarrow e} = (\mathbb{I}_{11} + \mathbf{P}\mathbf{F})^{-1}$ de dimensions 11×11 , telle que $\|T_{v \rightarrow e}\|_\infty = \Delta_m(T_{v \rightarrow e})^{-1}$ et la fonction de sensibilité en entrée $T_{d \rightarrow u} = (\mathbb{I}_4 + \mathbf{F}\mathbf{P})^{-1}$ de dimensions 4×4 , est définie par $\|T_{d \rightarrow u}\|_\infty = \Delta_m(T_{d \rightarrow u})^{-1}$. Par conséquent, la minimisation de la norme H_∞ de $T_{v \rightarrow e}$ ou de $T_{d \rightarrow u}$ correspond à l'augmentation des marges de module en entrée et en sortie. Étant donné que le système \mathbf{P} est MIMO, nous accordons de l'importance aux fonctions de sensibilité en entrée et en sortie qui ne coïncident pas, car \mathbf{P} et \mathbf{F} ne commutent pas.

Nous définissons aussi la matrice de transfert $T_{v \rightarrow u}$ de dimensions 4×11 liée à l'impact du bruit de mesure ν sur la commande \mathbf{u} , et $T_{d \rightarrow y}$ de dimensions 11×4 représentant l'impact de la perturbation en entrée \mathbf{d} sur la sortie du système \mathbf{y} .

Nous définissons des fonctions de pondération associées aux matrices de transfert précédemment explicitées. Sachant que $\|\cdot\|_\infty$ représente la norme H_∞ , la pondération $W_i(s)$ est associée à une matrice de transfert $T_i(s, x)$, qui permet de former la

fonction :

$$f_i(x) = \|W_i(s)T_i(s, x)\|_\infty \quad (4.4)$$

où x représente le vecteur de paramètres ajustables par l'optimisation. Avec cette formulation, il est possible d'utiliser un code d'optimisation permettant de minimiser $f(x)$.

En appliquant cette méthodologie dans notre cas et grâce aux fonctions de pondération $W = \text{diag}(W_1, \dots, W_4)$, la conception de \mathbf{H} et \mathbf{K} vise à rejeter une perturbation ou un échelon à basse fréquence \mathbf{w} agissant en entrée du système. L'objectif de la conception est d'amener \mathbf{y} à zéro, malgré la perturbation à basse fréquence sur \mathbf{w} . Plus précisément, nous avons sélectionné des fonctions de pondération constantes valant :

$$W_1 = 0.5, \quad W_2 = 0.5, \quad W_3 = 0.8, \quad W_4 = 0.5. \quad (4.5)$$

Ces valeurs ont été sélectionnées pour obtenir un bon compromis entre le rejet des perturbations et la vitesse de convergence du système.

Il est maintenant possible de former le problème d'optimisation multicritère suivant, en concaténant les contraintes exprimées sous la forme de (4.4)

$$\min_C \begin{vmatrix} \|W_1 T_{\nu \rightarrow e}(P, C)\| \\ \|W_2 T_{d \rightarrow u}(P, C)\| \\ \|W_3 T_{\nu \rightarrow u}(P, C)\| \\ \|W_4 T_{d \rightarrow y}(P, C)\| \end{vmatrix}_\infty, \text{ sous condition que} \\ C \in \mathbb{R}^{11 \times 4} \text{ stabilise } P \text{ en interne.} \quad (4.6)$$

Nous avons résolu (4.6) en utilisant la fonction Matlab **Systune** [Apkarian 2006]. Basé sur l'optimisation non lisse, **Systune** traite plusieurs scénarios non convexes, telle que l'architecture de contrôle structurée où nous optimisons les matrices de gain \mathbf{K} , \mathbf{H} . L'algorithme d'optimisation renvoie la sélection optimisée suivante :

$$\begin{bmatrix} \mathbf{H} \\ \mathbf{K} \end{bmatrix} = \begin{bmatrix} -1.902 & 0 & 7.201 & -9.043 & 0 & 33.244 & 0 & 0 & 0 & 4.696 & 0 \\ 0.425 & 0 & -1.620 & 2.024 & 0 & -7.480 & 0 & 0 & 0 & -1.045 & 0 \\ 0.035 & 0 & -0.728 & -1.853 & 0 & -4.445 & 0 & 0 & 0 & -0.323 & 0 \\ 0.035 & 0 & -0.728 & -1.853 & 0 & -4.445 & 0 & 0 & 0 & -0.323 & 0 \\ 0.217 & 0 & -0.164 & 1.074 & 0 & -0.527 & 0 & 0 & 0 & -0.773 & 0 \\ 0.217 & 0 & -0.164 & 1.074 & 0 & -0.527 & 0 & 0 & 0 & -0.773 & 0 \end{bmatrix}. \quad (4.7)$$

Nous obtenons une abscisse spectrale en boucle fermée pour \mathbf{A}_{cl} dans (4.3) valant $\alpha = -0.2381$.

4.4 Résultats

La figure 4.5 montre le résultat de la boucle fermée de la figure 4.4 avec la sélection des gains (4.7) et une valeur constante et croissante par morceaux du vent horizontal w (dernier graphique).

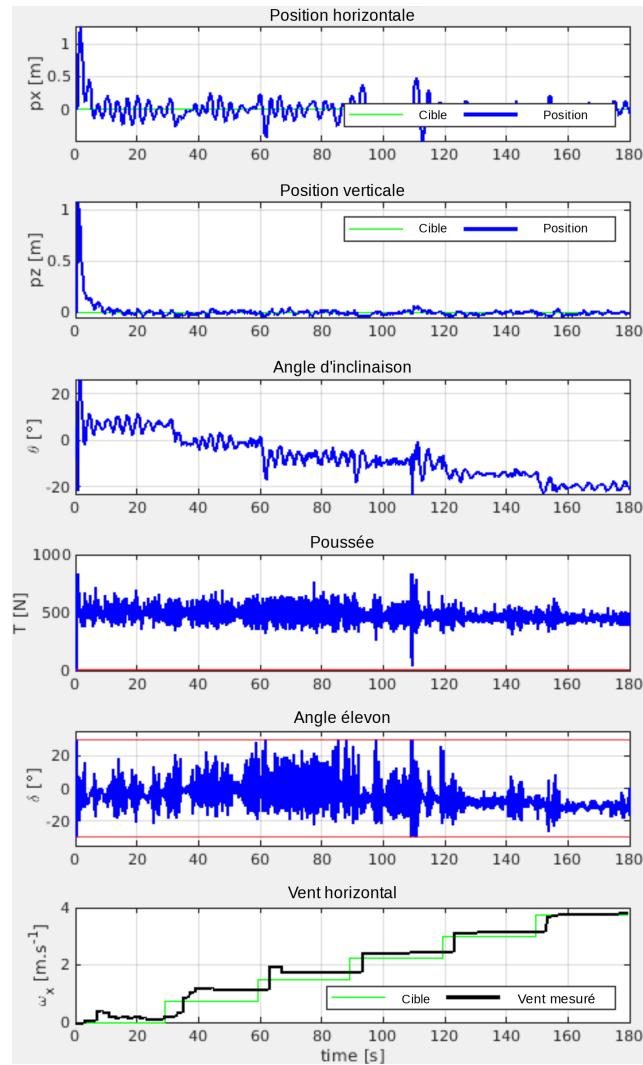


FIGURE 4.5 – Résultats expérimentaux.

Malgré quelques oscillations, le drone maintient sa position en dépit du vent, en inclinant convenablement l'angle de tangage. Les oscillations expérimentales sont absentes de nos simulations, ce qui suggère la présence de phénomènes non modélisés. Nous observons également un comportement qui pourrait être important pour de futures recherches : le drone semble se stabiliser plus facilement le long de l'axe

vertical que le long de l'axe horizontal.

Comme prévu, lorsque le vent augmente, l'angle d'inclinaison diminue, ce qui modifie la poussée nécessaire et la déflexion des élévons. En effet, le vent génère de la portance sur les ailes, ce qui compense l'effet de la gravité, et donc la poussée nécessaire devient plus faible. Pour chaque valeur de w , le modèle converge vers un équilibre, dont la caractérisation mathématique précise est détaillée dans 2.3.2. Il est donc nécessaire d'étendre la robustesse du contrôleur en effectuant une optimisation multimodèle du contrôleur (décrise dans 6.2.3). Il est également possible d'ajouter des degrés de liberté supplémentaires à la structure du contrôleur afin de lui donner plus de degrés de liberté dans l'optimisation.

La vidéo de la maquette, avec les résultats expérimentaux, est disponible via le lien : https://youtu.be/ce4_FUzeVzI.

4.5 Conclusion

Nous avons décrit une maquette permettant de simuler le vol d'un drone convertible, DarkO, en soufflerie. L'objectif de cette maquette est de tester le système de contrôle sur une représentation fidèle de la dynamique longitudinale, tout en simulant la dynamique de translation. Nous avons également présenté un contrôleur linéaire du type retour dynamique de sortie basé sur une architecture proportionnelle-intégrale pour la stabilisation du vol stationnaire dans des conditions de vent constant. Les gains du contrôleur ont été obtenus à l'aide d'une optimisation non convexe. Les résultats des essais expérimentaux montrent qu'il est possible de stabiliser l'équilibre en vol stationnaire dans la plage de vitesse du vent testée.

Toutefois, d'autres architectures de contrôle devraient être étudiées à l'avenir pour traiter les oscillations indésirables. Ces résultats indiquent qu'il est nécessaire de tester cette architecture de commande sur un modèle complet à six degrés de liberté.

CHAPITRE 5

Synthèse d'un retour statique de sortie par optimisation LMI

Sommaire

5.1	Motivation	81
5.2	Synthèse d'un contrôleur statique de sortie	82
5.2.1	Algorithme itératif pour la conception de retour statique de sortie	83
5.2.2	Architecture de commande	87
5.3	Résultats	88
5.3.1	Résultats de simulation	88
5.3.2	Résultats expérimentaux	90
5.4	Conclusion	93

5.1 Motivation

Dans le chapitre précédent, nous avons exploré la possibilité de stabiliser la dynamique longitudinale du drone à l'aide d'un contrôleur par retour dynamique de sortie. Les gains du contrôleur sont obtenus par une optimisation non-lisse basée sur des critères H_∞ . Nous proposons à présent d'étudier une manière différente de régler les gains du contrôleur, basée sur les inégalités linéaires matricielles.

Ce chapitre va permettre la mise en œuvre d'une formulation mathématique pour la synthèse du contrôleur basé modèle dans un cas pratique. Jusqu'à présent, cette approche a été testée et évaluée uniquement sur des cas théoriques [Arzelier 2018] avec des modèles dynamiques peu représentatifs des systèmes réels.

La méthode proposée pour contrôler le modèle de drone convertible étudié est basée sur la synthèse d'un contrôleur statique de sortie. La synthèse d'un contrôleur par retour de sortie statique pour la dynamique de DarkO augmentée s'inspire de [Syrmos 1997], où il est montré qu'un retour de sortie dynamique d'ordre $q \leq n$

(système d'ordre n), peut être converti en un retour de sortie statique à travers l'augmentation de l'espace d'état. Dans notre cas, des termes dynamiques tels que des intégrateurs et des filtres sont ajoutés à la dynamique du système, ce qui donne un espace d'état augmenté pour lequel une loi de bouclage statique (composée d'une matrice de gain statique) sera synthétisée.

La structure du correcteur permet de séparer très clairement la partie dynamique représentée notamment par l'intégrateur et la partie statique, représentée par les matrices de gain K et H . Dans le chapitre 4, les gains ont été obtenus par un outil d'optimisation non-lisse en se basant sur des contraintes fréquentielles H_∞ . Dans ce chapitre, on cherche à synthétiser K et H par un outil d'optimisation convexe du type LMI pour un problème qui est originellement du type BMI, ce qui engendre nécessairement un certain conservatisme dans la solution.

Le contrôleur est obtenu en convertissant algorithmiquement le problème d'inégalité matricielle bilinéaire (BMI) non convexe en trois problèmes convexes auxiliaires :

- un problème de retour d'état (SF),
- un problème d'injection de sortie (OI),
- un problème d'injection d'état (SI)

La résolution des problèmes (SF), (OI) et (SI) est une condition nécessaire à l'existence d'une solution pour le problème de retour statique de sortie. Par conséquent, la résolution de chacun des problèmes ci-dessus est abordée au moyen d'un algorithme en trois phases, présenté à la section 5.2. Cet algorithme est mis en œuvre sur la dynamique augmentée de DarkO, laquelle est incorporée dans la boucle ouverte pour imposer une forme souhaitée semblable aux principes du *loop shaping* [McFarlane 1992]. Cela conduit à des garanties multiples, notamment en termes de performance (avec un gain élevé en boucle ouverte à basse fréquence) et en termes de robustesse (avec une stratégie de gain faible en boucle ouverte à haute fréquence), tout en satisfaisant à l'exigence de suivi de la référence.

Les contrôleurs par retour statique de sortie seront validés par des simulations dans le domaine temporel. De plus, une validation sera également effectuée de manière expérimentale par le biais d'essais en vol réel, avec un modèle expérimental du drone DarkO.

5.2 Synthèse d'un contrôleur statique de sortie

La synthèse d'un contrôleur statique de sortie pose un problème non convexe en raison des multiplications entre les variables de décision, ce qui entraîne des inégali-

tés matricielles bilinéaires (BMI). Cette complexité rend le problème d'optimisation NP-difficile. En se référant à la formulation des BMI dans [Ebihara 2015], plusieurs reformulations équivalentes sont proposées dans [Arzelier 2018], ce qui conduit à l'inégalité matricielle présentée dans (5.1). La méthode proposée ici est extraite de [Ebihara 2015], s'appuyant notamment sur l'approche S-variable et les calculs duaux.

$$\begin{aligned} & \mathbf{P} \succ 0 \\ & He \left\{ \begin{bmatrix} 0 & 0 & \mathbf{P} \\ 0 & 0 & 0 \\ \mathbf{P} & 0 & 0 \end{bmatrix} \right\} \prec He \left\{ \begin{bmatrix} \mathbb{I}_p \\ -\left(\lambda \begin{bmatrix} C \\ 0_{p-n,n} \end{bmatrix} + \mathbf{M} \right) \\ -A \end{bmatrix} \right\} \mathbf{S}_1 + \begin{bmatrix} 0 \\ \mathbf{S}_2 \\ B\mathbf{Z} \end{bmatrix} \begin{bmatrix} 0 & \mathbb{I} & -\mathbf{H}^T \end{bmatrix} \end{aligned} \quad (5.1)$$

$$\mathbf{F} = -\mathbf{Z}\mathbf{S}_2^{-1} \begin{bmatrix} \mathbb{I}_p \\ 0_{n-p,p} \end{bmatrix} \quad (5.2)$$

Même si cette nouvelle reformulation est toujours non-convexe, elle permet d'obtenir un résultat. Si une solution est trouvée pour le système dans (5.1) avec $\lambda = 1$, $\mathbf{M} = 0$ et \mathbf{S}_2 non singulière, alors nous pouvons calculer une matrice de gain \mathbf{F} qui garantit la stabilité en boucle fermée (la preuve est disponible dans [Arzelier 2018]). Cette matrice de gain, qui n'est pas directement optimisée dans la synthèse convexe, est obtenue à l'aide de (5.2).

Cette preuve constitue l'objectif principal du processus d'optimisation. Relever les défis découlant de la nature non convexe du problème 5.1 nécessite des développements mathématiques et techniques spécifiques. La formulation mathématique introduite dans la littérature a donné des résultats théoriques initiaux prometteurs. L'algorithme, qui sera détaillé dans la sous-section suivante, sera appliqué à la dynamique du drone DarkO. L'analyse qui en résultera permettra de tirer des conclusions quant à l'efficacité de l'algorithme dans des scénarios d'essais pratiques.

5.2.1 Algorithme itératif pour la conception de retour statique de sortie

Le problème de la synthèse de gain d'un retour statique de sortie se caractérise par sa nature intrinsèquement non convexe qui s'accompagne de difficultés de calcul liées aux problèmes NP-difficile. Ceci, associé à une complexité inhérente qui provient de l'impossibilité d'optimiser toutes les variables de décision simultanément, rend le processus de division du problème non-convexe en de multiples sous-problèmes convexes auxiliaires extrêmement attrayants. L'inégalité matricielle bilinéaire (5.1) sera utilisée pour traduire ces objectifs dans un cadre d'optimisa-

tion. Chaque objectif particulier est atteint en fixant de manière optimale une partie des variables de décision, tout en affinant le reste par un processus itératif. Pour concevoir le contrôleur, un algorithme en trois phases introduit dans [Arzelier 2018] sera mis en œuvre Figure 5.1, lequel prend, en entrée, les matrices \mathbf{A} , \mathbf{B} , \mathbf{C} de la dynamique linéarisée du système.

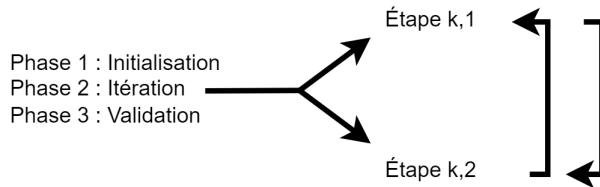


FIGURE 5.1 – Structure de l'algorithme d'optimisation.

5.2.1.1 Phase d'initialisation

L'objectif de la première phase est essentiellement de fournir une estimation initiale pour un contrôleur matriciel de gain à retour d'état \mathbf{K} qui stabilise la boucle fermée $\dot{\mathbf{x}} = (\mathbf{A} + \mathbf{B}\mathbf{K})\mathbf{x}$. Pour ce faire, la stratégie adoptée consiste à fixer les variables de décision λ , \mathbf{M} , \mathbf{H} de (5.1) de la manière suivante :

$$\lambda = \lambda = 0, \quad \mathbf{M} = M_0 = \begin{pmatrix} C^\circ C \\ C^{\perp T} \end{pmatrix}, \quad (5.3)$$

$$\mathbf{H} = H_0 = JM_0^{-1}, \quad J = (-\mu - h)\mathbb{I},$$

où h est un scalaire positif et $-\mu$ est la partie réelle maximale des valeurs propres de \mathbf{A} . L'étude de faisabilité menée dans la phase d'initialisation est la suivante :

$$\begin{aligned} \mathbf{P} &\succ 0 \\ He \left\{ \begin{bmatrix} 0 & 0 & \mathbf{P} \\ 0 & 0 & 0 \\ \mathbf{P} & 0 & 0 \end{bmatrix} \right\} &\prec He \left\{ \begin{bmatrix} \mathbb{I} \\ -M_0 \\ -\mathbf{A} \end{bmatrix} \mathbf{S}_1 + \begin{bmatrix} 0 \\ \mathbf{S}_2 \\ BZ \end{bmatrix} \begin{bmatrix} 0 & \mathbb{I} & -H_0^T \end{bmatrix} \right\} \end{aligned} \quad (5.4)$$

Si (5.4) est réalisable pour une combinaison de variables de décision \mathbf{S}_1 , \mathbf{S}_2 , \mathbf{Z} et pour le certificat de Lyapunov \mathbf{P} , alors une première estimation d'un gain de retour d'état stabilisant \mathbf{K} est trouvée, et nous pouvons passer à la phase d'itération. En revanche, si aucune solution n'est trouvée, le paramètre h est incrémenté de 1 et la phase d'initialisation est exécutée à nouveau. Cette phase résout les problèmes (SI) et (SF). Les résultats de cette phase sont les suivants :

$$S_{1.0} = \mathbf{S}_1, \quad \hat{K}_0 = -\mathbf{Z}\mathbf{S}_2^{-1}, \quad K = -\mathbf{Z}\mathbf{S}_2^{-1}M_0 \quad (5.5)$$

5.2.1.2 Phase d'itération - Étape k,1

Cette phase comprend deux étapes itératives. Dans la première étape, les variables d'optimisation λ et \mathbf{M} dans 5.1 précédemment fixées lors de l'initialisation, sont maintenant traitées comme des variables de décision. Pour maintenir la convexité, la variable de marge \mathbf{S}_1 est fixée à la valeur $S_{1,0}$ déterminée lors de la phase d'initialisation. Le système (5.6) est ensuite résolu.

$$\begin{aligned} \mathbf{P} &\succ 0, \quad \left[\begin{smallmatrix} (1-\lambda)\mathbb{I} & \mathbf{M}^T \\ \mathbf{M} & \mathbb{I} \end{smallmatrix} \right] \geq 0, \quad \lambda \geq 0 \\ H\mathbf{e}\left\{\left[\begin{smallmatrix} 0 & 0 & \mathbf{P} \\ \mathbf{P} & 0 & 0 \end{smallmatrix}\right]\right\} &\prec H\mathbf{e}\left\{\left[-\left(\lambda\left[\begin{smallmatrix} \mathbb{I} \\ 0_{p-n,n} \end{smallmatrix}\right] + \mathbf{M}\right)\right] S_{1,k-1} + \left[\begin{smallmatrix} 0 \\ -\mathbb{I} \\ B\hat{K}_{k-1} \end{smallmatrix}\right] [0 \ -\mathbf{S}_2 \ \mathbf{Y}^T]\right\} \quad (5.6) \end{aligned}$$

Dans (5.6), deux inégalités matricielles apparaissent en plus de l'inégalité du certificat de Lyapunov et de l'inégalité (5.1). Ces contraintes supplémentaires contribuent à la réalisation de l'objectif de la phase d'itération, qui consiste à trouver une solution pour (5.1) dans laquelle les variables λ et \mathbf{M} convergent respectivement vers 1 et 0, ce qui est une condition obligatoire pour atteindre l'objectif d'optimisation principal. Pour se faire, la maximisation de λ est définie comme un objectif dans le solveur d'optimisation. Une inégalité matricielle constraint la norme de \mathbf{M} et la réduit à 0, à mesure que λ augmente jusqu'à 1. La seconde inégalité est simplement une contrainte sur λ , ce dernier devant être positif.

Si une solution est trouvée pour (5.6) et si, pour cette solution, $1 - \lambda$ est inférieur à une tolérance définie, cela signifie que l'on peut passer à la troisième et dernière phase, à savoir la validation. Les résultats de cette phase sont les suivants :

$$\lambda_k = \lambda, \quad M_k = \mathbf{M}, \quad H_k^T = \mathbf{S}_2^{-1} \mathbf{Y}^T \quad (5.7)$$

Si ce n'est pas le cas, il y a une deuxième étape dans la phase d'itération.

5.2.1.3 Phase d'itération - Étape k,2

L'étape k,2 est relativement proche d'une étape d'initialisation. Comme pour la première phase, où λ , \mathbf{M} et H ont été fixés comme entrées pour générer un gain de retour d'état stabilisateur \mathbf{K} , la deuxième étape fixe les mêmes variables de décision avec des valeurs mises à jour à partir de l'étape k,1.

Le problème d'optimisation à résoudre dans cette phase est le suivant :

$$He \left\{ \begin{bmatrix} 0 & 0 & \textcolor{blue}{P} \\ 0 & 0 & 0 \\ \textcolor{blue}{P} & 0 & 0 \end{bmatrix} \right\} \prec He \left\{ \begin{bmatrix} \mathbb{I} \\ -\hat{M}(\alpha) \\ -A \end{bmatrix} \textcolor{blue}{S}_1 + \begin{bmatrix} 0 \\ \textcolor{blue}{S}_2 \\ BZ \end{bmatrix} \begin{bmatrix} 0 & \mathbb{I} & -H_k^T \end{bmatrix} \right\} \quad (5.8)$$

$$\hat{M}(\alpha) = \left((1 + \alpha(\lambda_k - 1)) \begin{bmatrix} C \\ 0_{p-n,n} \end{bmatrix} + \alpha M_k \right) \quad (5.9)$$

Lors de cette étape, un nouveau terme, $\hat{M}(\alpha)$ apparaît, dépendant d'une nouvelle variable de décision α . L'objectif est de minimiser α à l'aide de la méthode de bisection, en convertissant le problème non convexe en un problème convexe. Idéalement, α converge vers 0, signalant la fin de cette phase d'itération et la progression vers la phase de validation finale. Si $\alpha \approx 0$ selon une certaine tolérance, (5.9) se simplifie en (5.10). Par conséquent, les valeurs des variables de décision pour la solution de (5.8) lors l'étape k,2 sont identiques à la solution de (5.1), laquelle est résolue pour $\lambda = 1$ et $M = 0$. Cette solution permet le calcul d'un contrôleur par retour de sortie statique stabilisant.

$$\hat{M}(\alpha) = \left((1 + \alpha(\lambda_k - 1)) \begin{bmatrix} C \\ 0_{p-n,n} \end{bmatrix} + \alpha M_k \right) \stackrel{\alpha \approx 0}{=} \begin{bmatrix} C \\ 0_{p-n,n} \end{bmatrix} \quad (5.10)$$

Si, à l'étape k,2, α n'est pas inférieure à la tolérance fixée, l'algorithme commence une nouvelle étape d'itération à k,1. Les résultats de la phase actuelle sont les suivants :

$$\alpha_k = \alpha \quad \hat{K}_{k-1} = -Z \textcolor{blue}{S}_2^{-1} \quad S_{1,k} = \textcolor{blue}{S}_1 \quad (5.11)$$

5.2.1.4 Phase de validation

Dans les cas où l'étape d'itération fournit une solution pour laquelle $\lambda = 1$ et $M = 0$, une phase de validation n'est pas nécessaire car le contrôleur peut être calculé directement.

En pratique, l'algorithme quitte la phase d'itération avec des solutions $\lambda \approx 1$ et $M \approx 0$, en raison des limites des solveurs d'optimisation et des précisions numériques. Dans ces cas, les conditions pour atteindre l'objectif d'optimisation principal ne sont pas exactement remplies. Par conséquent, une phase de validation est nécessaire. Le problème à résoudre est défini par (5.12), où les variables sont

fixées $\lambda = 1$ et $M = 0$ et H est fixé à H_k (obtenu au cours de la phase d'itération). Si l'inégalité (5.12) est satisfaite alors la matrice de gain F peut être construite avec (5.2), ce qui garantit la stabilité de la boucle fermée ($A + BFC$). Cette dernière phase fournit la solution aux objectifs (OI) et (OF).

$$\begin{aligned} & P \succ 0 \\ & He \left\{ \begin{bmatrix} 0 & 0 & P \\ 0 & 0 & 0 \\ P & 0 & 0 \end{bmatrix} \right\} \prec He \left\{ \begin{bmatrix} \mathbb{I} \\ -[C \\ 0_{p-n,n}] \\ -A \end{bmatrix} \begin{bmatrix} S_1 \\ S_2 \\ BZ \end{bmatrix} + \begin{bmatrix} 0 \\ S_2 \\ BZ \end{bmatrix} [0 \ \mathbb{I} \ -H^T] \right\} \quad (5.12) \end{aligned}$$

5.2.2 Architecture de commande

L'architecture de contrôle du drone DarkO est conçue pour le stabiliser en vol stationnaire, avec et sans perturbations externes. Par exemple, pour contrer un vent de face affectant la vitesse linéaire le long des axes $x_{[b]}$ et $z_{[b]}$ et générant un moment autour de l'axe $y_{[b]}$, le schéma de commande utilise les ailerons et les hélices de manière symétrique pour générer un moment et une force compensatoires. Une action intégrale est employée pour contre-balancer la force le long de l'axe $z_{[b]}$, assurant une convergence asymptotique vers la force désirée en utilisant deux intégrateurs, un pour les moteurs et un pour les élevons.

L'architecture mise en œuvre, présentée à la Figure 5.2, est une reformulation de la loi de contrôle présentée dans le chapitre 4. Elle se compose du bloc dynamique linéarisé du drone DarkO (2.30) avec les matrices A_0 (2.31) et G_0 (2.32), du bloc de saturation pour les deux hélices et les deux élevons, comme présenté dans la section 2.2.2. La matrice de sélection des sorties élimine la mesure de l'état de l'angle de tangage, θ .

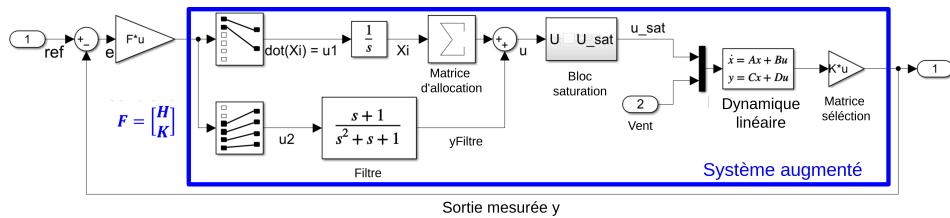


FIGURE 5.2 – Augmentation du système.

Pour la conception du contrôle, une structure de contrôleur de type PI est employée pour minimiser l'erreur statique et améliorer le rejet des perturbations ex-

ternes. Pour ce faire, nous ajoutons des filtres pré-compensateurs à l'entrée du système, sur la base de la méthodologie de *loop shaping*. Un intégrateur est ajouté pour assurer la performance (gain élevé) à basse fréquence, ce qui correspond à la spécification de suivi. De même, un filtre de deuxième ordre est ajouté pour assurer que le contrôleur soit strictement propre et robuste en ce qui concerne les dynamiques négligées grâce à un comportement d'atténuation. La fréquence de coupure du filtre est fixée à environ 5 Hz pour s'aligner sur la dynamique de la boucle fermée et rejeter le bruit dans les signaux mesurés. Pour limiter le nombre d'intégrateurs à 2 (un qui génère la commande d'intégration pour les deux hélices et l'autre pour les deux élévons), les sorties du bloc intégrateur sont doublées par une matrice d'allocation Σ . L'implémentation d'un filtre de second ordre présente l'avantage supplémentaire d'éviter un terme de transmission directe, qui peut amplifier les bruits de mesure indésirables sur \mathbf{y} et qui se répercutent sur les capteurs. La matrice de gain \mathbf{F} comprend les matrices de gain \mathbf{H} et \mathbf{K} respectivement pour l'action intégrale et proportionnelle. Les équations et la représentation de l'espace d'état du système augmenté se trouvent dans (5.13).

$$\begin{aligned}
 \dot{x}_i &= u_1, \\
 u &= \Sigma x_i + y_{filter} = \Sigma x_i + Filter \cdot u_2 \\
 \dot{x} &= Ax + Bu = Ax + B(\Sigma x_i + y_{filter}) = Ax + B\Sigma x_i + BCx_{filter}, \\
 \Sigma &= \begin{bmatrix} 1 & 0 \\ 0 & 1 \\ 0 & 1 \end{bmatrix}, \quad \begin{bmatrix} u_1 \\ u_2 \end{bmatrix} = Fe = F \left(\begin{bmatrix} ref \\ 0_{8 \times 1} \end{bmatrix} - y \right), \\
 \begin{bmatrix} \dot{x} \\ \dot{x}_c \\ \dot{x}_{filter} \end{bmatrix} &= \begin{bmatrix} A & B\Sigma & BC_{filter} \\ 0 & 0 & 0 \\ 0 & 0 & A_{filter} \end{bmatrix} \begin{bmatrix} x \\ x_c \\ x_{filter} \end{bmatrix} + \begin{bmatrix} 0 & 0 \\ 1 & 0 \\ 0 & B_{filter} \end{bmatrix} \begin{bmatrix} u_1 \\ u_2 \end{bmatrix}, \\
 y &= [C \ 0 \ 0] \begin{bmatrix} x \\ x_c \\ x_{filter} \end{bmatrix},
 \end{aligned} \tag{5.13}$$

où $\mathbf{x}_i \in \mathbb{R}^2$ sont les états des intégrateurs, $\mathbf{x}_f \in \mathbb{R}^4$ sont les états des filtres, $\mathbf{r} \in \mathbb{R}^3$ est le signal de référence, $\mathbf{y} \in \mathbb{R}^{11}$ est la sortie mesurée et $\mathbf{u} \in \mathbb{R}^4$ est l'entrée du système.

5.3 Résultats

5.3.1 Résultats de simulation

L'approche de la section 5.2.1 a été appliquée au système augmenté de la section 5.2.2, en se concentrant sur la dynamique linéarisée pour une vitesse de vent nulle (scénario de vol stationnaire). On a ainsi obtenu quatre contrôleurs stabilisateurs

pour $h = 12, 13, 14$ et 15 , h variant entre 1 et 40 . Plusieurs itérations avec différentes valeurs de h , servant de points d'initialisation pour l'algorithme d'optimisation, sont nécessaires pour améliorer la probabilité de convergence.

La Figure 5.3 montre la réponse temporelle en boucle fermée des quatre contrôleurs, avec des variations de consigne sur l'axe x à 5 s, l'axe y à 70 s et l'axe z à 40 s. Tous les contrôleurs stabilisent efficacement la dynamique en boucle fermée et suivent les références de position selon les trois axes. La réponse est lente sur les axes x et z et rapide sur l'axe y . Cela est dû à la capacité d'actionnement et l'utilisation différentielle des moteurs. De plus, une variation de consigne sur un axe n'entraîne pas de changement significatif sur les deux autres axes, ce qui indique un fort découplage entre les états de position. En outre, les actionneurs n'ont jamais été saturés.

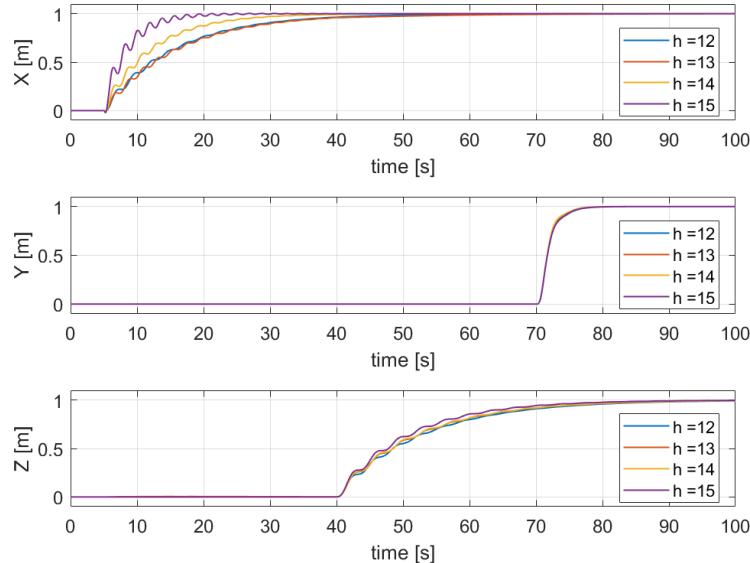


FIGURE 5.3 – Réponse du bouclage (5.13) avec une dynamique linéaire (2.30).

Le premier graphique de la Figure 5.4 montre que l'angle d'incidence θ subit des oscillations croissantes au fur et à mesure que h augmente. Bien que l'angle d'incidence θ ne soit pas directement contrôlé, il converge naturellement vers 0 lorsque les autres états se stabilisent, ce qui indique un vol stationnaire sans vent. Le second graphique montre que pour une dynamique linéarisée autour d'une vitesse de vent spécifique, θ converge vers une valeur non nulle, compensant les perturbations externes.

Les équations (5.14) et (5.15) présentent les gains du contrôleur synthétisés \mathbf{F} (bloc illustré dans 5.2) comprenant les gains matriciels \mathbf{H} et \mathbf{K} pour $h=12$ et une

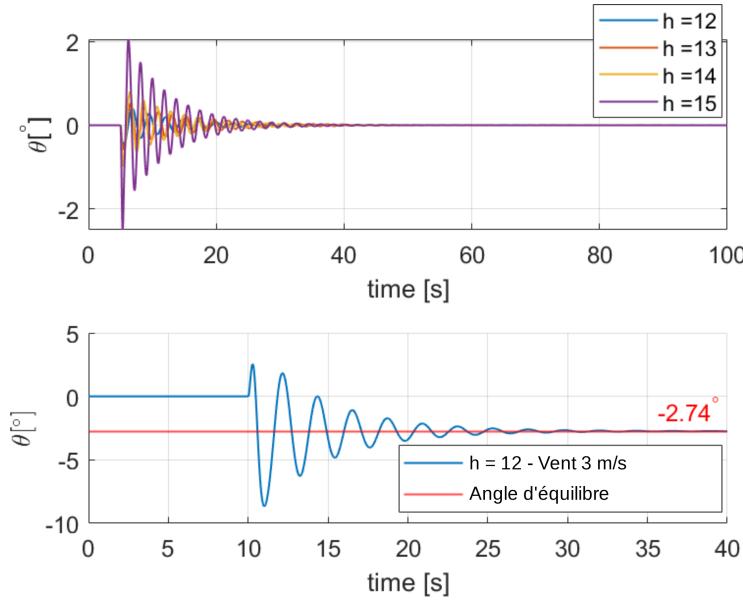


FIGURE 5.4 – Réponse temporelle de l'angle d'incidence θ en boucle fermée.

dynamique linéarisée à une vitesse de vent nulle. En examinant plus attentivement la matrice de gain \mathbf{K} qui produit quatre signaux de commande utilisés pour la commande asymétrique ou symétrique des actionneurs du drone, on peut observer que les lignes 1 et 2 (qui génèrent une commande de poussée sur les deux hélices) ainsi que les lignes 3 et 4 (qui génèrent une commande d'angle de déflexion pour les deux elevons) sont soit presque égales, soit opposées, à quelques exceptions près. Ce modèle a émergé naturellement du processus d'optimisation sans aucune contrainte, reflétant un alignement intuitif et cohérent avec la dynamique du drone.

$$\mathbf{H} = \begin{bmatrix} 3.5e-6 & -3.0e-7 & 1.5e-2 & -6.9e-6 & -7.4e-6 & 4.4e-1 & 1.8e-4 & 1.1e-4 & -8.4e-5 & -8.6e-5 & -2.3e-4 \\ -2.0e-2 & -7.5e-6 & 1.6e-5 & -2.0e-1 & 4.4e-5 & -1.1e-4 & -9.2e-5 & -9.1e-5 & 7.4e-6 & 7.6e-1 & -3.9e-5 \end{bmatrix} \quad (5.14)$$

$$\mathbf{K} = \begin{bmatrix} 7.6e-3 & -2.0e+1 & 9.9 & 3.1e-2 & -4.5e+1 & 8.2e+1 & -3.1e+2 & -3.1e+2 & 6.4e-1 & 3.0e-3 & -1.0e+2 \\ 5.4e-3 & 2.0e+1 & 9.9 & 8.8e-3 & 4.5e+1 & 8.2e+1 & 3.1e+2 & 3.1e+2 & -6.5e-1 & -1.9e-3 & 1.0e+2 \\ -4.2 & -3.2e-1 & 8.7e-3 & -4.5e+1 & -4.4e-1 & -3.1e-2 & -1.2e+1 & 1.6 & -3.1e+1 & 3.9e+1 & -1.6 \\ -4.2 & 3.2e-1 & 4.8e-3 & -4.5e+1 & 4.3e-1 & 3.3e-2 & 1.2e+1 & -1.7 & 3.1e+1 & 3.9e+1 & 1.6 \end{bmatrix} \quad (5.15)$$

5.3.2 Résultats expérimentaux

Pour la validation expérimentale, les contrôleurs ont été testés sur le modèle réel de drone DarkO construit à l'ENAC (Figure 5.5). Les essais en vol ont eu lieu dans la volière de l'ENAC, équipée d'un système de capture de mouvement Optitrack fournissant des données de position et d'attitude à 40 Hz, éliminant ainsi le besoin

d'un capteur GPS. La vitesse est obtenue par une différence finie.



FIGURE 5.5 – Modèle expérimental du drone DarkO.

Des algorithmes de fusion de données, y compris des filtres invariants, ont combiné les données d'Optitrack (position, vitesse et attitude) et de l'unité IMU du drone DarkO pour améliorer l'estimation de l'état. Cette estimation est utilisée pour créer le vecteur de sortie y , entrée de la loi de contrôle (5.13). Compte tenu de l'architecture de la boucle de contrôle, l'estimation doit être de très bonne qualité avec le délai le plus court possible.

La nature modulaire de Paparazzi permet d'utiliser l'implémentation existante du filtre invariant pour estimer le vecteur d'état de DarkO et d'ajouter un module de stabilisation basé sur la loi de contrôle décrite ci-dessus (voir 5.2.2). L'autopilote échantillonne les lois de commande et de contrôle à 500 Hz, générant des commandes de contrôle appropriées pour réaliser les manœuvres de vol souhaitées et stocker toutes les données, en vue d'une analyse a posteriori.

Les essais expérimentaux suivants constituent les expérimentations réalisées dans la volière de l'ENAC où des contrôleurs ont été obtenus avec des méthodes de synthèse basées sur une modélisation du drone convertible. Les vols précédents, pour ce type de drone, utilisaient des conceptions de contrôle combinées basées sur des modèles et des capteurs, comme les algorithmes INDI. Chaque vol commence par le décollage du drone et sa stabilisation à une position de référence à l'aide d'un contrôleur INDI. Après avoir atteint la position de vol stationnaire souhaitée, le contrôleur INDI est remplacé par le contrôleur par retour de sortie. La Figure 5.6 représente le drone DarkO pendant son vol d'essai. Les quatre contrôleurs ont été testés et validés sur le système réel, démontrant une performance stable et un suivi de position précis sur les trois axes.



FIGURE 5.6 – DarkO lors d'un vol expérimental.

La Figure 5.7 montre les résultats des essais en vol de l'un des quatre contrôleurs synthétisés. Comme cela a également été observé dans les simulations, sur les axes x et z , le temps de réponse est relativement élevé alors que sur l'axe y , il est plus rapide, en raison de l'actionnement différentiel important sur cet axe soutenu par la dynamique rapide de l'actionneur. La dynamique du drone sur l'axe z est caractérisée par de petites oscillations autour de la position de référence, qui peuvent également être observées dans le signal de commande des vitesses angulaires des hélices, illustrées à la Figure 5.7. Ces oscillations réduites sont dues au fait que la dynamique de l'actionneur n'est pas prise en compte dans la loi de commande. Au cours des vols expérimentaux, le contrôleur n'a pas saturé les actionneurs (voir la Figure 5.7).

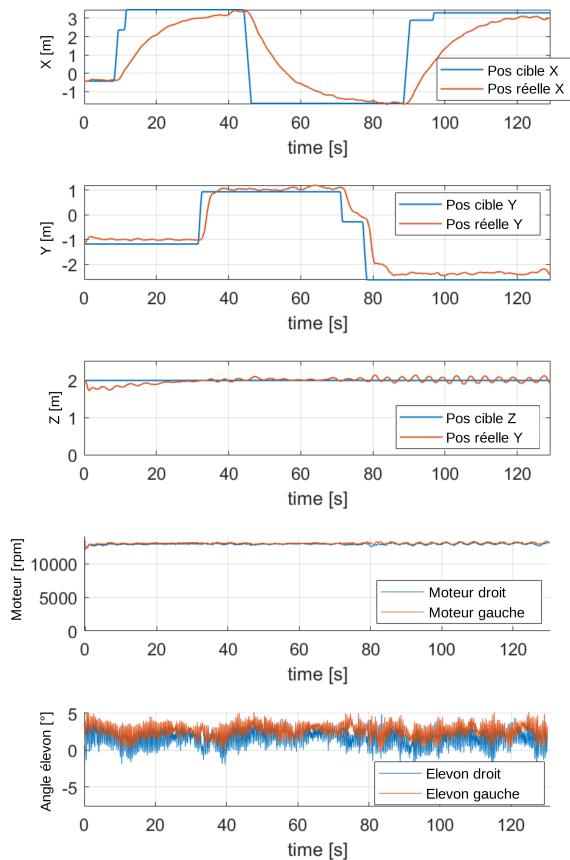


FIGURE 5.7 – Réponse expérimentale en boucle fermée et entrée de commande pour $h = 12$.

5.4 Conclusion

Pour traiter de la synthèse d'un retour statique de sortie, nous pouvons envisager deux approches étant donné le caractère non convexe et NP-difficile du problème. La première approche consiste à utiliser des outils d'optimisation non convexes pour directement optimiser les variables de décision du contrôleur, à savoir les matrices de gain. C'est ce qui est présenté dans les chapitres 4 et 6 en utilisant l'optimisation non-lisse et la fonction `Systune` de Matlab. La seconde approche consiste à utiliser des outils d'optimisation non convexes, ici l'outil LMI. Il s'agit alors de proposer un algorithme de résolution du problème BMI en une succession de problèmes LMI. C'est donc, par exemple, l'approche développée dans [Arzelier 2018], qui a été implémentée et testée en simulation et expérimentalement sur le drone.

Un algorithme d'optimisation convexe, utilisant le cadre LMI et la théorie de la stabilité de Lyapunov, a été employé pour synthétiser des contrôleurs à retour

de sortie statique pour le modèle de drone convertible DarkO. Cette technique de synthèse, basée sur le modèle, a permis de stabiliser efficacement la dynamique en boucle fermée du système, assurant une réponse temporelle satisfaisante et un suivi de la référence sans saturation des actionneurs. Malgré une prise en compte partielle des phénomènes non-linéaires, les contrôleurs ont démontré un bon comportement lors des premiers tests expérimentaux sur le modèle de drone DarkO. Avec les matrices de gain du contrôleur conçues et la structure de la loi de commande, des vols expérimentaux ont été menés avec succès pour le vol stationnaire et le suivi de la trajectoire. Ce résultat constitue une solide preuve de concept pour la loi de commande développée en termes de performance. Il est maintenant nécessaire de s'intéresser à l'impact du vent sur l'architecture du drone et aux impacts sur l'obtention des gains du contrôleur.

CHAPITRE 6

Expérimentation d'un contrôle robuste pour DarkO

Sommaire

6.1	Motivation	95
6.2	Contrôleur structuré proportionnel intégral avec <i>roll off</i>.	96
6.2.1	Description du schéma de contrôle	96
6.2.2	Contrôleur optimisé sous contraintes H_∞ , cas sans vent	99
6.2.3	Contrôleur optimisé sous contraintes H_∞ , cas multimodèle	101
6.3	Rafale de vent	107
6.4	Vol expérimental en soufflerie ouverte	108
6.5	Conclusion	111

6.1 Motivation

Nous avons observé expérimentalement dans le chapitre 4 qu'un drone *tailsitter* pouvait se stabiliser en stationnaire face à du vent, vers les points d'équilibre décrits dans la section 2.3.2, grâce à une architecture de commande linéaire. Cette dernière est basée sur un retour de sortie, proportionnel intégral (PI). Toutefois, ce type de contrôleur est un correcteur propre, ne garantissant pas la stabilité par rapport aux dynamiques négligées. Il a naturellement tendance à augmenter la sensibilité de la commande u au bruit de mesure sur y .

Pour résoudre ce problème, nous avons proposé une architecture que l'on peut qualifier de PI étendu par un filtre passe bas, lequel a pour fonction de répondre aux deux points précédents. Ce filtre n'est pas fixé a priori, mais est un paramètre de synthèse avec les matrices de gain K et H pour obtenir le meilleur compromis possible entre performance (réjection de la perturbation vent) et robustesse aux non-linéarités et aux dynamiques négligées au travers des marges de module et de la politique de *roll off* du filtre. Dans notre architecture, le retour de sortie libère

deux angles d'orientation du drone. Cette relaxation permet une convergence de ces angles vers des valeurs non connues, a priori, assurant le rejet des perturbations.

6.2 Contrôleur structuré proportionnel intégral avec *roll off*.

6.2.1 Description du schéma de contrôle

Une inspection minutieuse des matrices de commande et d'entrée des perturbations \mathbf{G}_w et \mathbf{E}_w du modèle (2.34) (voir la sortie de l'Algorithme 2) suggère une architecture de contrôle efficace pour rejeter une perturbation de vent constante \mathbf{w} . En effet, les ailerons et les hélices peuvent être utilisés symétriquement pour générer respectivement un moment autour de l'axe $y_{[b]}$, vérifiant l'équation (2.24), et une force le long de l'axe $x_{[b]}$, vérifiant l'équation (2.22), compensant ainsi l'effet de la perturbation. Néanmoins, il reste une force le long de l'axe $z_{[b]}$ à compenser en vérifiant l'équation (2.23). Une action intégrale peut converger asymptotiquement vers la force désirée, même avec une perturbation du vent non mesurée (\mathbf{w}). Nous pouvons donc stabiliser le drone à l'équilibre en vol stationnaire tel que caractérisé dans le Théorème 1. Comme nous ne mesurons pas le vent \mathbf{w} , les valeurs de ψ et θ dans l'Algorithme 1 sont inconnues. Le contrôleur proposé, illustré à la Figure 6.1, utilise l'action intégrale pour faire converger ces angles vers leur valeur d'équilibre. Le bouclage utilise les variables d'erreur suivantes :

$$\mathbf{e}_p = \mathbf{r}_p - \mathbf{p}, \quad \mathbf{e}_{v\epsilon\omega} = - \begin{bmatrix} \mathbb{I}_3 & \mathbf{0}_{3 \times 1} & \mathbf{0}_{3 \times 2} & \mathbf{0}_3 \\ \mathbf{0}_{1 \times 3} & 1 & \mathbf{0}_{1 \times 2} & \mathbf{0}_{1 \times 3} \\ \mathbf{0}_3 & \mathbf{0}_{3 \times 1} & \mathbf{0}_{3 \times 2} & \mathbb{I}_3 \end{bmatrix} \begin{bmatrix} \tilde{v} \\ \tilde{\epsilon} \\ \tilde{\omega}_b \end{bmatrix}. \quad (6.1)$$

Ces variables d'erreur doivent converger vers zéro dans le cas d'un vol stationnaire et pour n'importe quelle position de référence constante $\mathbf{r}_p \in \mathbb{R}^3$. Notons que \mathbf{r}_p est l'entrée de référence du schéma de contrôle.

Les variables d'erreur dans (6.1) peuvent être représentées comme dans le schéma de la Figure 6.1, c'est-à-dire, en définissant la sortie $\mathbf{y} \in \mathbb{R}^{10}$ de la dynamique du système linéarisé (2.34), ayant le vecteur d'état incrémental $\tilde{\mathbf{x}} \in \mathbb{R}^{10 \times 1}$ défini ci-dessous :

$$\mathbf{y} = \mathbf{C} \tilde{\mathbf{x}} + \begin{bmatrix} \mathbf{p}_{eq} \\ \mathbf{0}_{7 \times 1} \end{bmatrix}, \quad \mathbf{C} := \begin{bmatrix} \mathbb{I}_6 & \mathbf{0}_{6 \times 1} & \mathbf{0}_{6 \times 2} & \mathbf{0}_{6 \times 3} \\ \mathbf{0}_{1 \times 6} & 1 & \mathbf{0}_{1 \times 2} & \mathbf{0}_{1 \times 3} \\ \mathbf{0}_{3 \times 6} & \mathbf{0}_{3 \times 1} & \mathbf{0}_{3 \times 2} & \mathbb{I}_3 \end{bmatrix}, \quad (6.2)$$

où la matrice de sortie $\mathbf{C} \in \mathbb{R}^{10 \times 12}$ enlève la composante $\tilde{\epsilon}_2$ et $\tilde{\epsilon}_3$ du vecteur d'état $\tilde{\mathbf{x}}$.

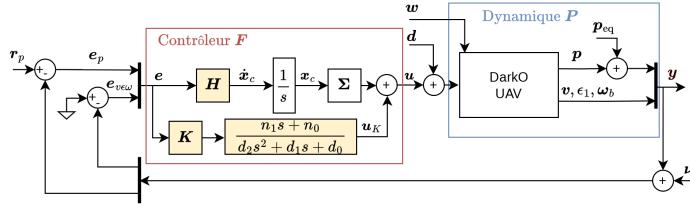


FIGURE 6.1 – Schéma de commande intégrale avec la perturbation de vent w , la perturbation du système à l'entrée d et à la sortie ν .

Comme le montre la Figure 6.1, les équations de la dynamique du contrôleur sont basées sur l'erreur e suivante :

$$\mathbf{e} = [e_p^\top \ e_{v\omega}^\top]^\top, \quad \dot{\mathbf{x}}_c = \mathbf{H}\mathbf{e}, \quad \mathbf{u} = \boldsymbol{\Sigma}\mathbf{x}_c + \mathbf{u}_K, \quad (6.3a)$$

$$\boldsymbol{\Sigma} := \begin{bmatrix} 1 & 1 & 0 & 0 \\ 0 & 0 & 1 & 1 \end{bmatrix}^\top, \quad \mathbf{u}_K = \frac{n_1s + n_0}{d_2s^2 + d_1s + d_0} \mathbf{K}\mathbf{e}, \quad (6.3b)$$

où $\mathbf{x}_c \in \mathbb{R}^2$ est l'état intégral, $\boldsymbol{\Sigma}$ est une matrice d'allocation des entrées qui permet d'affecter la première composante de l'état de l'intégrateur à l'action des hélices et la deuxième composante à l'action des elevons. Les scalaires n_1, n_0, d_2, d_1, d_0 sont respectivement les coefficients du numérateur et du dénominateur d'un filtre utilisé pour éviter une transmission directe entrée-sortie qui amplifierait le bruit de mesure à haute fréquence. Ce filtre induit un contrôleur strictement propre, pour une robustesse accrue aux incertitudes additives.

Nous définissons le contrôleur \mathbf{F} ayant pour dimensions 4×10 avec la matrice de transfert $\mathbf{F}(s) = T_{e \rightarrow u}(s)$ telle que décrite dans (6.3) et l'interconnexion détaillée dans la Figure 6.1. Le système \mathbf{P} de dimensions 10×4 représente la dynamique linéarisée de DarkO. La sortie du système $\mathbf{y} \in \mathbb{R}^{10 \times 1}$ est utilisée comme entrée du contrôleur \mathbf{F} .

Compte tenu des symétries des actionneurs du drone, nous avons constraint la structure de la matrice \mathbf{K} dans (6.3), associée à l'action proportionnelle du contrôleur, afin d'utiliser les actionneurs selon leur action physique. Ainsi, \mathbf{K} prend la forme :

$$\mathbf{K}_{\text{struct}} = \begin{bmatrix} k_1 & -k_2 & k_3 & k_4 & -k_5 & k_6 & -k_7 & k_8 & k_9 & -k_{10} \\ k_1 & k_2 & k_3 & k_4 & k_5 & k_6 & k_7 & -k_8 & -k_9 & k_{10} \\ -k_{11} & -k_{12} & k_{13} & -k_{14} & -k_{15} & -k_{16} & k_{17} & -k_{18} & k_{19} & -k_{20} \\ -k_{11} & k_{12} & k_{13} & -k_{14} & k_{15} & k_{16} & -k_{17} & k_{18} & k_{19} & k_{20} \end{bmatrix}. \quad (6.4)$$

La structure précédente s'explique par le comportement physique du drone. Une

erreur de position sur l'axe $z_{[i]}$ du repère inertiel NED (voir Figure 2.2) entraîne une utilisation symétrique des deux hélices, ce qui génère une force le long de l'axe $x_{[b]}$ du drone. L'utilisation symétrique des deux moteurs se traduit par le même signe dans les coefficients k_3 et k_6 des colonnes 3 et 6 de \mathbf{K} , qui correspondent respectivement aux erreurs de position et de vitesse sur l'axe $z_{[i]}$. De même, une erreur de position ou de vitesse le long de l'axe latéral du drone $y_{[b]}$ sera compensée par une utilisation antisymétrique des moteurs, comme le montrent les coefficients k_2 et k_5 et leur signe opposé dans les colonnes 2 et 5 de \mathbf{K} . Une erreur de vitesse angulaire autour de l'axe $x_{[b]}$ doit être compensée par une utilisation antisymétrique des élevons, comme le montre le coefficient k_{18} de signe opposé dans la colonne 8 de \mathbf{K} . Des arguments équivalents expliquent les coefficients restants de la matrice \mathbf{K} dans (6.4). Toutes ces explications ne sont valables que dans un voisinage de l'équilibre, où le drone se trouve être à la verticale. On comprend aisément que dans d'autres configurations, ces contraintes d'actionnement ne sont plus valides. Un avantage de la structure de (6.4) est la réduction du nombre de variables à optimiser, de 40 à 20 gains scalaires, cela se traduisant par une diminution du temps nécessaire à l'optimisation.

La boucle fermée illustrée à la Figure 6.1 est un retour de sortie à dix éléments, comprenant les trois positions, les trois vitesses linéaires, l'un des trois angles d'attitude (ϵ_1) et les trois vitesses angulaires. Cette structure peut être considérée comme un bouclage proportionnel-intégral étendu MIMO. Les paramètres à régler dans le contrôleur \mathbf{F} (6.3) sont le gain proportionnel $\mathbf{K} \in \mathbb{R}^{4 \times 10}$ dans (6.4), le gain intégral $\mathbf{H} \in \mathbb{R}^{2 \times 10}$ et les paramètres du filtre n_1, n_0, d_2, d_1, d_0 , comme repérés en jaune sur la Figure 6.1. Une méthode de réglage appropriée doit garantir une réjection adéquate des perturbations et une robustesse satisfaisante aux dynamiques non modélisées. Ces deux objectifs conduisent à un compromis car le rejet des perturbations nécessite un réglage agressif tandis que les propriétés de robustesse sont assurées par une stratégie d'atténuation des hautes fréquences et donc de la bande passante.

Nous examinons ensuite deux méthodes de réglage basées sur l'optimisation. La première est issue des idées proposées dans 4.3, méthode qui ne nécessitait pas la dynamique linéarisée des Théorèmes 1 et 2, et est résumée dans la section 6.2.2. Il s'agit d'une synthèse multiobjectif avec des contraintes H_∞ , basée sur le modèle de vent nul, discutée dans la section 2.3.1 et 2.4.1 et détaillée dans 4.3. Nous montrerons que cette première méthode ne permet pas de stabiliser le drone dans certaines plages de vent, en raison de la méconnaissance de la dynamique caractérisée par les Théorèmes 1 et 2. La deuxième méthode de réglage, présentée à la section 6.2.3, est une synthèse itérative multiobjectif et multimodèle avec des contraintes H_∞ , basée sur un ensemble de modèles associés à différentes conditions de vent, lesquels sont

obtenus des Théorèmes 1 et 2, par le biais des Algorithmes 1 et 2.

Dans notre validation numérique, présentée dans les sections 6.2.2 et 6.2.3 (voir en particulier les Figure 6.2 et Figure 6.4), un bruit de mesure est ajouté à la sortie pour produire des résultats numériques semblables aux expérimentaux. Les écarts types des niveaux de bruit adoptés sont indiqués dans la Table 6.1.

Grandeurs	Valeurs	Unités
\mathbf{p}	2.5×10^{-4}	m
$\tilde{\mathbf{v}}$	1.2×10^{-3}	m s^{-1}
$\tilde{\epsilon}$	4.7×10^{-4}	
$\tilde{\omega}_b$	2.7×10^{-3}	rad s^{-1}

TABLE 6.1 – Écart-type du bruit pour la modélisation des capteurs en simulation.

En plus de présenter les résultats de la simulation du bouclage linéaire de la Figure 6.1 avec le modèle linéarisé (2.34), dans les sections 6.2.2 et 6.2.3, nous simulons également la boucle fermée en remplaçant le modèle linéarisé \mathbf{P} par le modèle non-linéaire (2.2), comprenant la dynamique réelle du drone.

Lorsque l'on remplace le modèle linéarisé par la dynamique non-linéaire (2.2), dont l'état est $\mathbf{x} = (\mathbf{p}, \mathbf{v}, \mathbf{q}, \omega_b) \in \mathbb{R}^{13}$, nous remplaçons la sortie linéaire \mathbf{y} avec la version non-linéaire suivante :

$$\mathbf{y}_{NL} = \begin{bmatrix} \mathbf{p} \\ \mathbf{v} \\ \epsilon_1 \\ \omega_b \end{bmatrix} = \begin{bmatrix} \mathbb{I}_6 & 0_{6 \times 1} & 0_{6 \times 1} & 0_{6 \times 2} & 0_3 \\ 0_{1 \times 3} & 0 & 1 & 0_{1 \times 2} & 0_{1 \times 3} \\ 0_3 & 0_{3 \times 1} & 0_{3 \times 1} & 0_{3 \times 2} & \mathbb{I}_3 \end{bmatrix} \begin{bmatrix} \mathbf{R}_\psi^\top \mathbf{p} \\ \mathbf{R}_\psi^\top \mathbf{v} \\ \mathbf{q}_{eq\psi}^{-1} \otimes \mathbf{q} \\ \omega_b \end{bmatrix}. \quad (6.5)$$

6.2.2 Contrôleur optimisé sous contraintes H_∞ , cas sans vent

Pour régler le contrôleur dans le cas sans vent, nous utilisons le modèle linéaire du système décrit dans la section 2.4.1, $\mathbf{P}(s) = T_{u \rightarrow y}(s)$, obtenu à partir des équations (2.30) et (6.2), tel que :

$$\mathbf{P}(s) = \mathbf{C}(s\mathbb{I}_{12} - \mathbf{A}_0)^{-1} \mathbf{G}_0.$$

En lien avec la Figure 6.1 et de manière similaire à ce qui a été réalisé à la section 4.3.2, nous introduisons des matrices de transfert qui correspondent aux objectifs de robustesse : la fonction de sensibilité en sortie définie par $T_{\nu \rightarrow e} = (\mathbb{I}_{10} + \mathbf{P}\mathbf{F})^{-1}$ de dimensions 10×10 , telle que $\|T_{\nu \rightarrow e}\|_\infty = \Delta_m(T_{\nu \rightarrow e})^{-1}$ et la fonction de sensibilité en entrée $T_{d \rightarrow u} = (\mathbb{I}_4 + \mathbf{F}\mathbf{P})^{-1}$ de dimensions 4×4 , définie par $\|T_{d \rightarrow u}\|_\infty = \Delta_m(T_{d \rightarrow u})^{-1}$. Par conséquent, la minimisation de la norme H_∞ de $T_{\nu \rightarrow e}$ ou de

$T_{d \rightarrow u}$ correspond à l'augmentation des marges de module en entrée et en sortie. Étant donné que le système \mathbf{P} est MIMO, nous accordons de l'importance aux fonctions de sensibilité en entrée et en sortie qui ne coïncident pas, car \mathbf{P} et \mathbf{F} ne commutent pas.

Nous définissons aussi la matrice de transfert $T_{\nu \rightarrow u}$, de dimensions 4×10 , liée à l'impact du bruit de mesure ν sur la commande u , et $T_{d \rightarrow y}$, de dimensions 10×4 , représentant l'impact de la perturbation en entrée d sur la sortie du système y .

Nous résolvons le même problème que dans notre travail précédent (4.6) en utilisant la fonction **Systune** de Matlab [Apkarian 2006], mais nous utilisons le diagramme de contrôle présenté dans la section 6.2.1 qui comprend un filtre sur l'action proportionnelle et un nombre différent de sorties. Nous incluons également dans le système \mathbf{P} la dynamique des actionneurs linéaires discutée dans la section 2.2.2.

Une augmentation successive de l'intensité horizontale et verticale du vent (allant de 0 à -6 m s^{-1}) est appliquée, comme le montre le tracé inférieur de la Figure 6.2. Les couples de vent sélectionnés (w_x, w_z) sont représentés par des points rouges sur les surfaces de la figure 2.7, où l'on peut voir que l'équilibre $(\mathbf{u}_{\text{eq}}, \mathbf{x}_{\text{eq}})$ est atteint sans que les actionneurs ne soient saturés. Nous ne nous intéressons qu'à la partie négative de la vitesse verticale du vent car elle est la plus limitante. En effet, le drone est soulevé par le vent vertical ascendant (dont le signe est négatif dans le cadre de la NED), nécessitant ainsi moins de traction sur les hélices pour compenser la gravité. Les moteurs génèrent moins de flux d'air sur les élévons, ce qui réduit leur efficacité, conduit à la saturation et déstabilise le drone. L'objectif du système de contrôle est de maintenir le drone en position de vol stationnaire (définie comme $\mathbf{r}_p = [0, 0, 0]^\top$), malgré l'augmentation du vent horizontal et vertical w_x et w_z .

La Figure 6.2 présente à la fois des simulations linéaires avec la dynamique linéarisée (2.34) (en pointillé) et des simulations non-linéaires avec le modèle non-linéaire (2.2) (en traits pleins). Les simulations linéaires et non-linéaires montrent systématiquement que le contrôleur fonctionne bien à faible vitesse de vent (effectivement, le réglage est effectué sur la base du modèle de vent nul). Cependant, lorsque la vitesse du vent w_x et w_z dépasse -5 m s^{-1} , la position de vol stationnaire devient instable et le drone oscille et diverge. Les angles d'inclinaison θ sont utilisés pour représenter l'attitude afin de donner un meilleur aperçu du comportement du véhicule, mais la simulation de la dynamique non-linéaire (2.2) est effectuée avec un quaternion unitaire. L'instabilité observée dans les résultats de la simulation de la figure 6.2 confirme les instabilités expérimentales rapportées dans la section 4.4, où nous avons utilisé cette même méthode de réglage. Est également confirmée l'importance des Théorèmes 1 et 2 dans la Section 2.1, pour un réglage approprié

des gains du contrôleur, comme cela est effectué dans la section suivante.

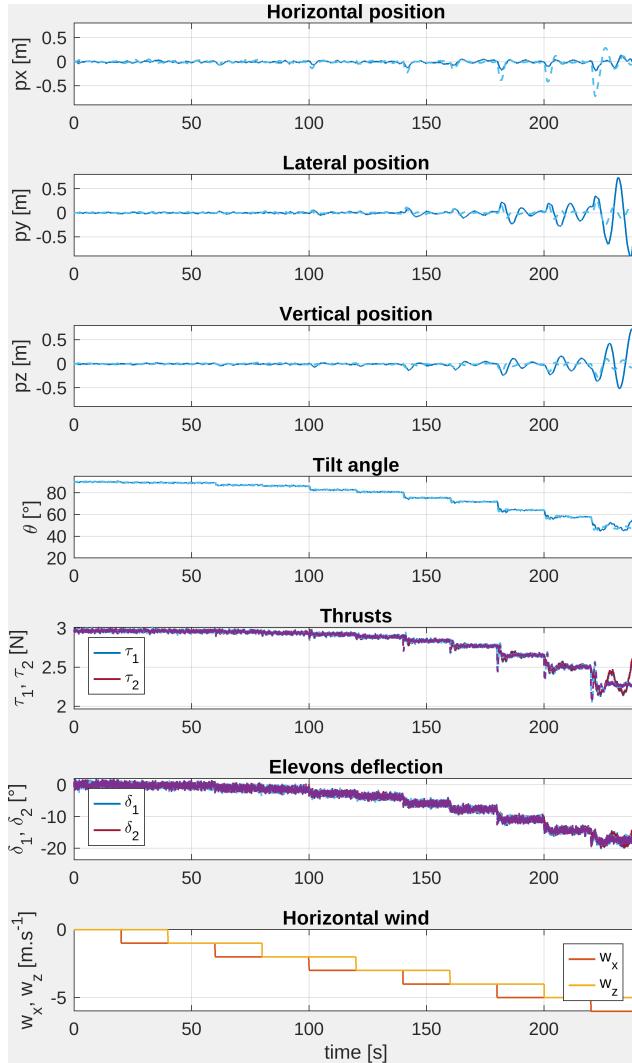


FIGURE 6.2 – Simulation du modèle non-linéaire (2.2) (ligne continue) et du modèle linéarisé (2.34) (ligne en pointillés) avec des incrément de vent constants croissants et le contrôleur basé sur l’optimisation sans vent de la section 6.2.2.

6.2.3 Contrôleur optimisé sous contraintes H_∞ , cas multimodèle

Les résultats de simulation obtenus avec la méthode de réglage sans vent (voir la Figure 6.2), ainsi que les instabilités expérimentales observées dans la section 4.4 confirment la nécessité d’une procédure de réglage du gain du contrôleur exploitant les linéarisations paramétrées en fonction du vent des Théorèmes 1 et 2. En nous concentrant à nouveau sur le schéma de contrôle de la Figure 6.1, nous considérons

maintenant explicitement l'effet du vent (linéarisé) sur le système (2.34) avec la sortie (6.2) et avec les sélections de l'algorithme 2 comme définies ci-dessous :

$$\begin{aligned}\mathbf{P}_w(s) &= \begin{bmatrix} \mathbf{P}_u(s, w) & \mathbf{P}_w(s, w) \end{bmatrix} \\ &:= \mathbf{C}(s\mathbb{I}_{12} - \mathbf{A}_w)^{-1} \begin{bmatrix} \mathbf{G}_w & \mathbf{E}_w \end{bmatrix},\end{aligned}\quad (6.6)$$

où l'entrée est la concaténation de l'entrée de commande \mathbf{u} et de l'entrée de perturbation de vent \mathbf{w} . Comme le modèle dépend de la vitesse du vent \mathbf{w} , nous introduisons une nouvelle matrice de transfert $T_{w \rightarrow y}$ ayant des dimensions de 10×3 , laquelle correspond à la matrice de transfert entre l'entrée du vent \mathbf{w} et la sortie du système \mathbf{y} , quantifiant l'effet de la perturbation du vent sur la boucle de commande du drone.

Avec l'ensemble des matrices de transfert définies dans la section 6.2.2 et la nouvelle matrice de transfert $T_{w \rightarrow y}$, nous utilisons l'approche proposée dans [Apkarian 2006, Apkarian 2013]. Ainsi, nous pouvons régler notre architecture de contrôle structurée, pour laquelle nous optimisons les matrices de gain \mathbf{K} , \mathbf{H} et les paramètres de filtrage n_1, n_0, d_2, d_1, d_0 (en jaune sur la Figure 6.1). Nous résolvons le problème d'optimisation multiobjectif, en exploitant l'implémentation Matlab [Apkarian 2013, §3].

En particulier, sur la base d'un ensemble \mathcal{W} comprenant une collection finie de couples (w_x, w_z) , avec $w_x \in [0, 8] \text{ m s}^{-1}$ et $w_z \in [-4, 4] \text{ m s}^{-1}$, nous considérons l'ensemble des systèmes linéarisés qui résulte de (6.6) et nous résolvons l'optimisation convexe suivante, où les scalaires W_1, W_2, W_3, W_4 et W_5 sont des facteurs de pondération à ajuster pour obtenir un compromis satisfaisant entre la robustesse (associée à W_1, W_2, W_3 et W_4) et la performance (associée à W_1, W_2 et W_5) :

$$\gamma^* = \min_{\mathbf{F}} \max_{w \in \mathcal{W}} \left| \begin{array}{c} \|W_1 T_{v \rightarrow e}(\mathbf{P}_w, \mathbf{F})\|_\infty \\ \|W_2 T_{d \rightarrow u}(\mathbf{P}_w, \mathbf{F})\|_\infty \\ \|W_3 T_{v \rightarrow u}(\mathbf{P}_w, \mathbf{F})\|_\infty \\ \|W_4 T_{d \rightarrow y}(\mathbf{P}_w, \mathbf{F})\|_\infty \\ \|W_5 T_{w \rightarrow y}(\mathbf{P}_w, \mathbf{F})\|_\infty \end{array} \right|, \text{ sous condition que} \quad (6.7)$$

\mathbf{F} stabilise en interne $\mathcal{F}_\ell(\mathbf{P}_w, \mathbf{F})$, $\forall w \in \mathcal{W}$,

où $\mathcal{F}_\ell(\mathbf{P}_w, \mathbf{F})$ désigne l'interconnexion de bouclage linéaire de la Figure 6.1 pour une valeur spécifique de w (ceci est cohérent avec la notation classique de contrôle robuste [Apkarian 2006, Apkarian 2013]).

Il est intéressant de noter que W_1 et W_2 peuvent simultanément servir à la

performance et à la robustesse (MM). Effectivement, ces deux pondérations agissent sur les sensibilités en sortie et en entrée du système, lesquelles auront une action sur la bande passante du bouclage et un impact dérivatif en basses fréquences, ce qui permettra d'ajuster la performance. De plus, ces transferts pondérés assureront la robustesse du système grâce à la minimisation de la norme H_∞ de $W_1 T_{\nu \rightarrow e}(\mathbf{P}_w, \mathbf{F})$ ou de $W_2 T_{d \rightarrow u}(\mathbf{P}_w, \mathbf{F})$ correspond à l'augmentation des marges de module en entrée et en sortie. Cette marge de module est la plus petite distance du point critique -1 au lieu de transfert en boucle ouverte dans le cas SISO, selon le critère de Nyquist. Par extension dans le cas MIMO, nous en déduisons que la maximisation de la marge de module permet de s'éloigner de l'instabilité.

Algorithme 3 Réglage itératif et multimodèle des gains du contrôleur.

Entrées : $\mathbf{A}_w, \mathbf{G}_w, \mathbf{E}_w$ les matrices de sortie de l'algorithme 2 et les scalaires de pondération positifs W_1-W_5

Sorties : \mathbf{K}, \mathbf{H} et les gains du filtre

- 1: (Initialisation) Initialiser \mathcal{W} comme un maillage comprenant toutes les paires $w_x \in \{0, -4, -8\}$ et $w_z \in \{-4, 0, 4\}$, soit initialement neuf modèles de synthèse.
- 2: (Synthèse) Résoudre l'optimisation (6.7) avec la fonction **Systune**.
- 3: (Analyse) Définir un maillage de validation \mathcal{W}_v en discrétilisant l'intervalle $(w_x, w_y) \in [0, 8] \times [-4, 4]$ avec un pas de discrétilisation de 1, soit 81 modèles d'analyse. En utilisant le contrôleur \mathbf{F} obtenu à l'étape précédente et pour chaque $w_v \in \mathcal{W}_v$, nous calculons :

$$\gamma_v = \left\| \begin{array}{l} \|W_1 T_{\nu \rightarrow e}(\mathbf{P}_{w_v}, \mathbf{F})\|_\infty \\ \|W_2 T_{d \rightarrow u}(\mathbf{P}_{w_v}, \mathbf{F})\|_\infty \\ \|W_3 T_{\nu \rightarrow u}(\mathbf{P}_{w_v}, \mathbf{F})\|_\infty \\ \|W_4 T_{d \rightarrow y}(\mathbf{P}_{w_v}, \mathbf{F})\|_\infty \\ \|W_5 T_{w \rightarrow y}(\mathbf{P}_{w_v}, \mathbf{F})\|_\infty \end{array} \right\|_\infty. \quad (6.8)$$

Augmenter \mathcal{W} avec le point correspondant si $\gamma_v > 1$ ou si γ_v est indéfini (à savoir si aucune solution n'est trouvée).

- 4: (Conclusion) Si \mathcal{W} n'a pas été augmenté à l'étape précédente, passer à l'étape 5, sinon passer à l'étape 2.
 - 5: **Retourne :** \mathbf{K}, \mathbf{H} et les paramètres du filtre n_1, n_0, d_2, d_1, d_0
-

Le problème d'optimisation (6.7) devient de plus en plus lourd d'un point de vue informatique, à mesure que nous augmentons la cardinalité de l'ensemble des conditions de vent considérées dans \mathcal{W} . En fait, une approche de force brute incluant un maillage fin de points dans \mathcal{W} conduit à une optimisation difficile à calculer. Au lieu de cela, nous suivons ici la procédure itérative décrite dans l'algorithme 3, où \mathcal{W} est initialement sélectionné comme un maillage grossier comprenant $3 \times 3 = 9$

points (étape 1); une étape de synthèse (étape 2) est ensuite suivie de manière répétée par une étape d'analyse (simple du point de vue du calcul) (étape 3) où le contrôleur \mathbf{F} est fixé. L'étape 3 identifie les points de violation en utilisant un maillage de validation plus fin \mathcal{W}_v et les ajoute à l'ensemble d'optimisation \mathcal{W}_v . L'algorithme se termine après quelques itérations, lorsque aucun point du maillage de validation ne viole les contraintes.

Pondération	W_1	W_2	W_3	W_4	W_5
Valeurs	18	16	11	26	5

TABLE 6.2 – Valeurs des scalaires de pondération positifs W_1 – W_5 utilisés dans l'exécution de l'Algorithme 3.

L'exécution de l'algorithme 3, pour les modèles de DarkO issus des théorèmes 1 et 2 avec la sélection des scalaires de pondération positifs W_1 – W_5 indiqués dans la Table 6.2, a donné la sélection suivante après 2 itérations :

$$\begin{aligned} \left[\mathbf{K}^\top \mid \mathbf{H}^\top \right] &= \left[\begin{array}{ccccc|cc} -3.86 & -3.86 & 0.79 & 0.79 & 0.02 & 0.48 \\ 1.43 & -1.43 & 1.71 & -1.71 & -0.47 & -1.63 \\ 4.06 & 4.06 & -2.07 & -2.07 & -0.45 & 0.52 \\ -6.86 & -6.86 & -11.60 & -11.60 & -0.14 & 1.40 \\ -10.75 & 10.75 & -1.89 & 1.89 & 3.35 & 5.69 \\ 27.20 & 27.20 & -4.29 & 4.29 & -1.84 & 3.79 \\ -12.32 & 12.32 & -3.46 & 3.46 & 3.72 & 6.81 \\ -5.84 & 5.84 & -2.29 & 2.29 & 1.58 & 3.13 \\ -5.19 & 5.19 & 5.79 & 5.79 & 2.86 & -1.54 \\ -6.52 & 6.52 & 0.08 & -0.08 & 0.08 & 2.82 \end{array} \right], \\ \left[\begin{array}{c|c} n_1 & n_0 \\ \hline d_2 & d_1 \\ \hline d_0 & \end{array} \right] &= \left[\begin{array}{c|c} -429 & -389 \\ \hline 1 & 6475 \\ \hline 4905 & \end{array} \right]. \end{aligned} \tag{6.9}$$

Après la première itération de l'algorithme 3 et qu'un contrôleur candidat \mathbf{F} ait été évalué à l'étape 2, nous pouvons tracer la Figure 6.3. Elle montre en bleu les diagrammes de bode des valeurs singulières maximales des matrices de transferts $T_{\nu \rightarrow e}$, $T_{d \rightarrow u}$, $T_{\nu \rightarrow u}$, $T_{d \rightarrow y}$, et $T_{w \rightarrow y}$ (associées à la valeur de γ_v) reportées dans (6.8) à l'étape d'analyse 3. L'inverse des cinq poids W_1 à W_5 est représenté par les lignes horizontales vertes. Les diagrammes en rouge correspondent aux points qui violent les contraintes et qui sont ajoutés à l'ensemble \mathcal{W} pour l'itération suivante. Dans notre cas, ils correspondent aux conditions de vent suivantes : $(w_x, w_z) = (-1, 4), (-1, 3), (-1, 2), (-1, 1), (-1, 0), (-1, 1)$ m s⁻¹. Les quelques diagrammes en magenta correspondent aux 9 points considérés dans \mathcal{W} pour la première itération de l'étape de synthèse 2. Les diagrammes rouges de la Figure 6.3 illustrent clairement que l'algorithme itératif parvient à détecter les valeurs critiques de la vitesse du vent (w_x, w_z) que nous ajoutons à l'ensemble d'optimisation \mathcal{W} .

Les valeurs singulières de la fonction de sensibilité de la sortie et de l'entrée (respectivement $T_{r \rightarrow e}$ et $T_{d \rightarrow u}$) sont représentées sur la Figure 6.3 (ligne supérieure). Le graphique de la troisième ligne représente la valeur singulière du transfert entre la perturbation du vent w et la sortie du drone y . La valeur singulière qui tangente la contrainte est celle de la condition de vent la plus élevée du modèle de synthèse, $(w_x, w_z) = (-8, -4) \text{ m s}^{-1}$.

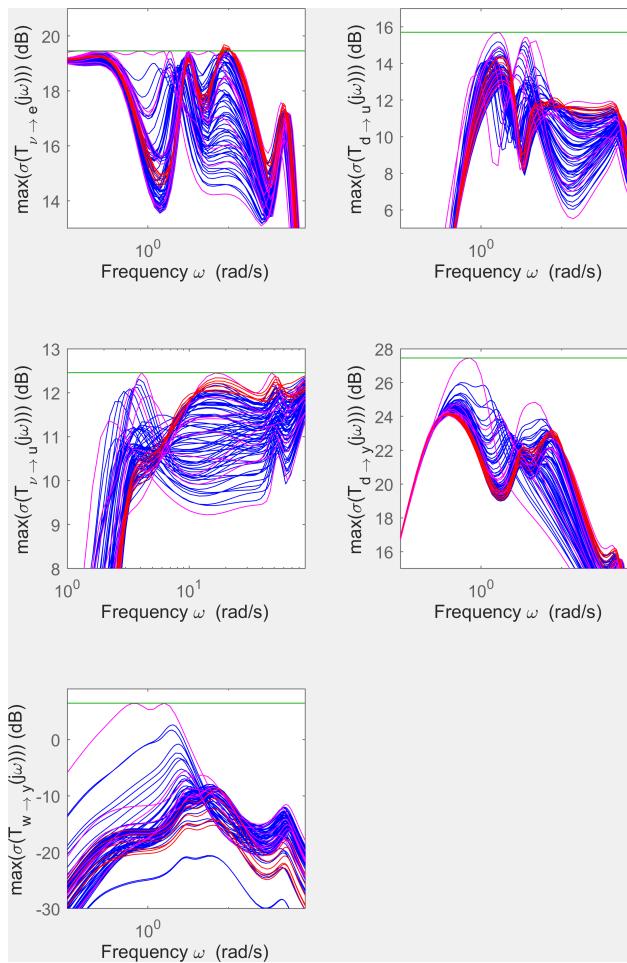


FIGURE 6.3 – Diagrammes des valeurs singulières des fonctions de transfert dans (6.8), à la première itération de l'algorithme 3.

Avec le réglage indiqué dans (6.9), tel qu'il est obtenu avec l'Algorithm 3, nous présentons dans la Figure 6.4 des résultats de simulation similaires à ceux déjà présentés dans la Figure 6.2 pour la méthode de réglage sans vent discutée dans la Section 6.2.2. Une fois de plus, nous simulons à la fois le modèle non-linéaire (2.2) (lignes pleines) et le modèle linéarisé (2.34) (lignes en pointillés).

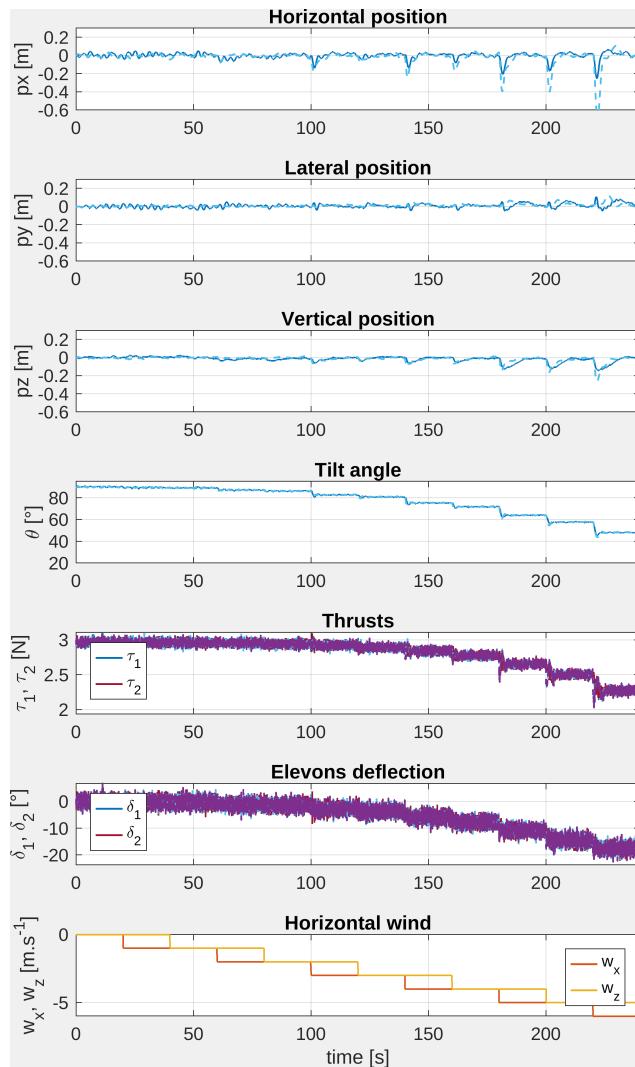


FIGURE 6.4 – Simulation du modèle non-linéaire (2.2) (ligne continue) et du modèle linéarisé (2.34) (ligne en pointillé) avec des incrément de vent constants croissants, le contrôleur étant réglé à l'aide de l'optimisation multimodèle de l'algorithme 3 dans la section 6.2.3.

Par rapport à la Figure 6.2, les simulations de la Figure 6.4 montrent que le réglage du contrôleur basé sur les Théorèmes 1 et 2 résout les problèmes d'instabilité et parvient à stabiliser le vol stationnaire dans tous les scénarios de vent considérés. Nous notons également que la Figure 6.4 montre une action plus agressive. En effet, l'entrée de contrôle u (à la fois la poussée et les déflexions) est plus affectée par le bruit de mesure.

L'efficacité du schéma de contrôle réglé sur la base de l'Algorithme 3 est également confirmée par les résultats expérimentaux présentés dans la section suivante.

6.3 Rafale de vent

Dans la section 1.3.2, nous avons présenté des modèles de perturbation représentant une modélisation d'une rafale de vent. Nous allons les utiliser pour évaluer la réponse du drone face à cette perturbation.

Nous avons appliqué un vent horizontal w_x avec trois valeurs moyennes $w_m \in [0, 0.3, 0.6] \text{ m s}^{-1}$ et avec les caractéristiques suivantes pour la fonction "Chapeau mexicain" $A_g = -5 \text{ m s}^{-1}$, $f_g = \frac{1}{T_g} = 0.2 \text{ Hz}$ soit une période T_g de 5 secondes. Les résultats sont visibles sur la Figure 6.5.

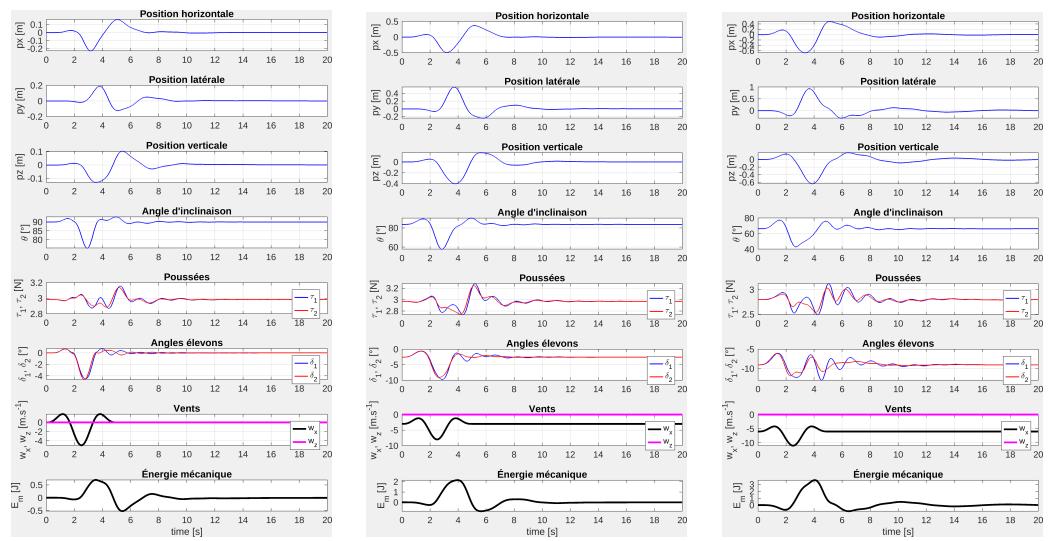


FIGURE 6.5 – Simulation du modèle non-linéaire (2.2) face à une perturbation "Chapeau mexicain" (1.1) avec $f_g = 0.2 \text{ Hz}$.

Nous avons aussi réalisé une simulation similaire à la précédente avec une fréquence égale à $f_g = 1.2 \text{ Hz}$ soit une période T_g de 0.83 seconde. Cette valeur est utilisée dans [Gillebaart 2014] pour représenter une rafale. Les résultats sont visibles sur la Figure 6.6.

Enfin, nous avons réalisé une simulation avec un vent horizontal w_x avec trois valeurs moyennes $w_m \in [0, 0.3, 0.6] \text{ m s}^{-1}$ et avec la caractéristique suivante pour la fonction "ondelettes de Morlet" $A_g = -5 \text{ m s}^{-1}$. Les résultats sont visibles sur la Figure 6.7. Nous observons que le drone et son contrôleur sont en mesure de rejeter les différentes perturbations. Nous observons, sur l'ensemble des simulations, que le drone lutte contre le vent incident en modifiant son incidence : il s'aligne dans le flux du vent, ce qui a pour impact majeur de faire prendre de l'altitude au drone.

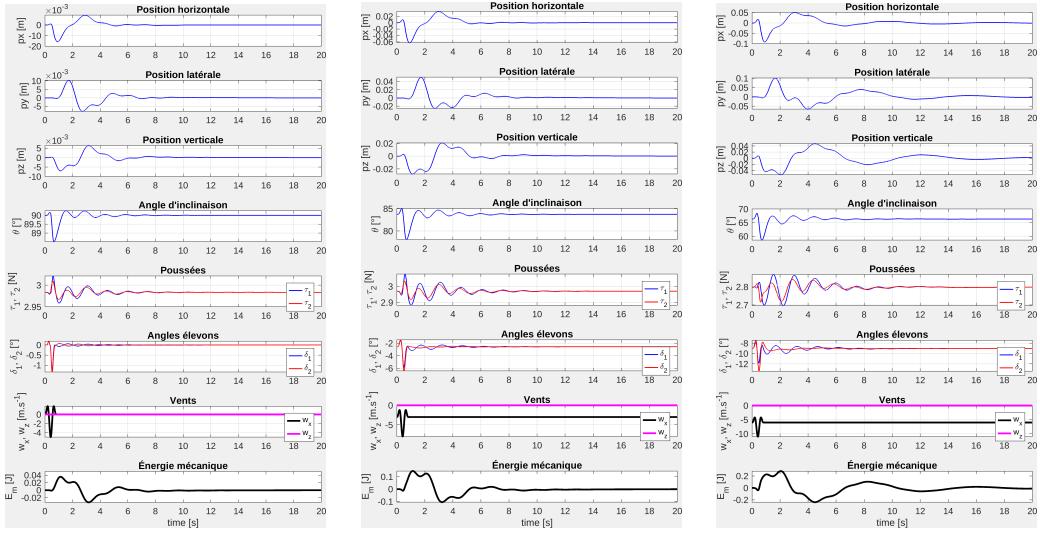


FIGURE 6.6 – Simulation du modèle non-linéaire (2.2) face à une perturbation "Chapeau mexicain" (1.1) avec $f_g = 1.2 \text{ Hz}$.

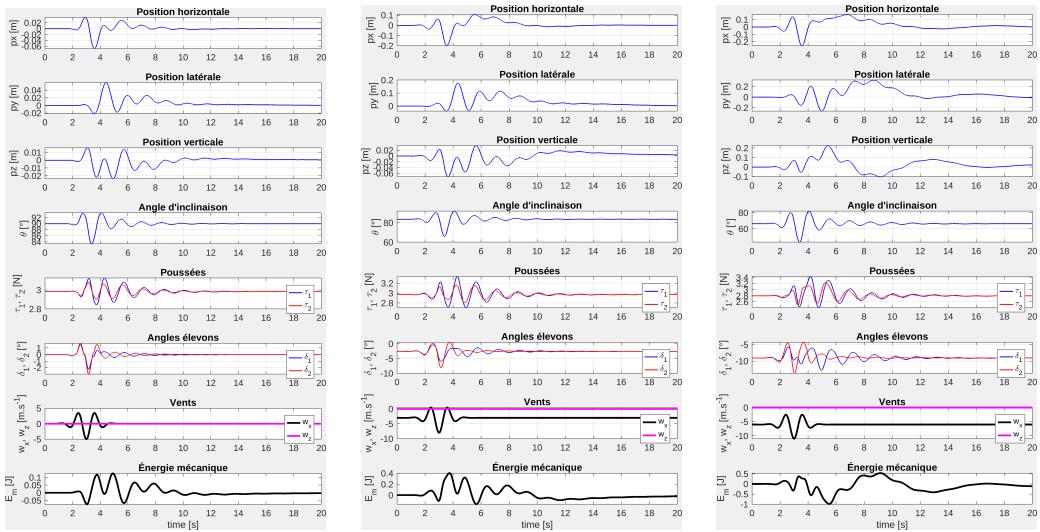


FIGURE 6.7 – Simulation du modèle non-linéaire (2.2) face à une perturbation "ondelettes de Morlet" (1.2).

6.4 Vol expérimental en soufflerie ouverte

Les vols expérimentaux de DarkO se sont déroulés dans un espace dédié (voir Figure 6.8) avec un système de localisation Optitrack basé sur une convention NED selon la Figure 2.2. Nous avons utilisé un générateur de vent à veine ouverte pour obtenir des incrément de vent que nous avons mesurés à l'aide d'une sonde à fil chaud (la barre verticale dans la Figure 6.8).



FIGURE 6.8 – Vol expérimental de DarkO devant la soufflerie ouverte.

Bien que cette information sur le vent soit enregistrée à bord du drone pour synchroniser les données, nous n'utilisons pas cette mesure dans la loi de commande. La fréquence de mesure de cette sonde de vent n'est que de 0,5 Hz, de sorte que nous n'avons qu'une mesure toutes les deux secondes. L'estimation de l'état est effectuée à l'aide d'un système de navigation inertielles pour fusionner les données de l'unité de mesure inertielles (IMU) et du système de localisation Optitrack afin d'obtenir une estimation précise de la sortie \mathbf{y} dans la Figure 6.1. Cependant, la vitesse angulaire du drone $\boldsymbol{\omega}_b$ est mesurée sur la base du gyromètre de l'IMU, qui fournit des mesures bruitées. Nous avons donc ajouté un filtre passe-bas *Butterworth* de second ordre avec une fréquence de coupure de 20 Hz pour lisser la sortie $\boldsymbol{\omega}_{textb}$. Le filtre de *Butterworth* est pris en compte dans la dynamique linéarisée lors de l'optimisation des gains du contrôleur en suivant l'Algorithme 3.

Nous avons également utilisé les ESC similaires à ceux utilisés lors de l'identification montrée dans la Figure 2.3 pour l'actionnement des hélices. Les deux ESC ont été flashés avec le code open-source AM32 (voir Annexe). De cette manière, nous compensons les effets de décharge de la batterie et obtenons un suivi précis de la vitesse commandée. Avant cette modification, l'action intégrale de la rétroaction stabilisatrice de la Figure 6.1 compensait la perte de vitesse du moteur causée par la réduction de la tension de la batterie pendant le vol. Cette compensation intégrale a été indirectement générée par la perte d'altitude du drone causée par la réduction de la traction.

Nous avons réalisé une expérience de vol au cours de laquelle DarkO a été mis manuellement en mode de vol stationnaire stabilisé devant la soufflerie, puis nous avons activé la loi de contrôle de l'Algorithme 3. Comme le drone devait être stabilisé à au moins 30 cm de la soufflerie, un pilotage manuel a été réalisé pour éviter tout dépassement qui pourrait endommager la soufflerie. Une fois que DarkO était suffisamment proche du point de consigne \mathbf{r}_p de la Figure 6.1, nous avons

activé le contrôleur proposé, obtenant les résultats de la Figure 6.9.

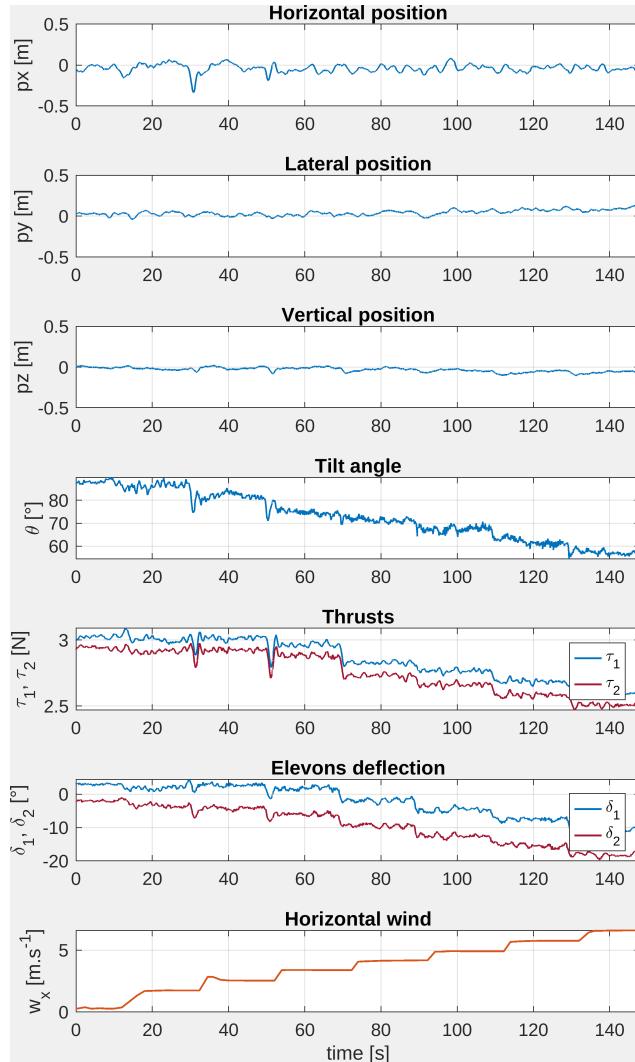


FIGURE 6.9 – Expérience du drone DarkO devant la soufflerie avec des incrémentations de vent constants croissants (graphique du bas).

Au cours de la phase d'expérimentation, comme le montre le graphique inférieur de la Figure 6.9, nous avons augmenté progressivement la vitesse du vent, en attendant 20 secondes entre chaque évolution de vent jusqu'à une vitesse finale de 7 m s^{-1} .

Les figures 6.9 et 6.10 montrent que le drone maintient sa position malgré l'augmentation de la vitesse du vent. Nous pouvons noter quelques points importants, en accord avec les simulations : la traction du moteur diminue lorsque la vitesse du vent augmente. Le schéma de contrôle profite de la portance générée par le vent

pour soutenir le drone, de sorte que moins d'énergie soit nécessaire pour stabiliser la position en vol stationnaire. Le drone maintient son angle d'inclinaison à une valeur inconnue, a priori, pour la loi de commande et qui découle naturellement de l'action intégrale. Cette valeur est atteinte asymptotiquement et converge vers la valeur requise de θ . Pour stabiliser la position, le drone utilise les elevons pour annuler le moment de tangage généré par la forme de l'aile, soumise à un vent horizontal, sans atteindre les limites de saturation. On note également une légère asymétrie de l'efficacité des actionneurs, qui est efficacement compensée par l'action proportionnelle du schéma de contrôle. Cette asymétrie est visible sur le cinquième et sixième graphique de la figure 6.9, où l'on observe un écart entre la vitesse de rotation du moteur droit et du moteur gauche. Cet écart peut être dû à des variations légères de la forme de l'hélice ou aux performances des moteurs et contrôleurs. Il en est de même pour les elevons.

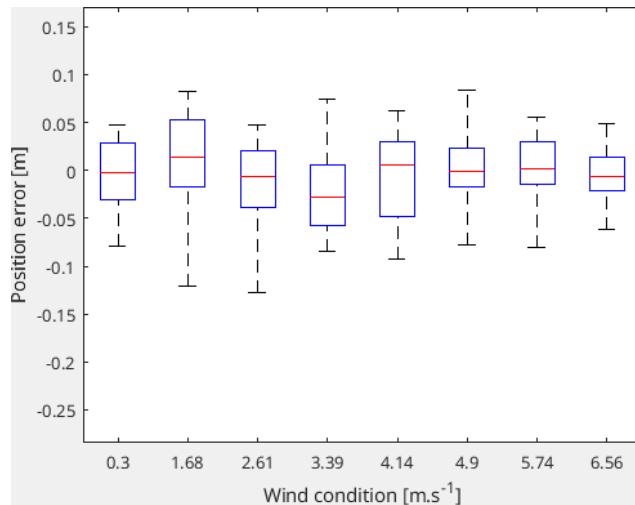


FIGURE 6.10 – Visualisation statistique des performances de vol stationnaire (en rouge la valeur moyenne, en bleu l'écart type à 95% et en pointillé les bornes minimales et maximales).

6.5 Conclusion

Nous avons proposé un schéma de contrôle pour la stabilisation d'un *tailsitter* lors d'un vol stationnaire en présence d'un vent constant inconnu. Notre bouclage contient une action intégrale et ne nécessite pas la mesure de la vitesse du vent. Les modèles paramétriques linéarisés se sont avérés être un instrument clé pour effectuer la synthèse d'un contrôleur robuste aux différents cas de vent. Après avoir étudié les

résultats de la simulation, nous avons mené une campagne de vols expérimentaux dans un environnement contrôlé pour valider notre solution de contrôle.

La principale limitation de cette architecture réside dans l'impossibilité de mesurer le vent dans les phases de vol stationnaire ou de vitesse faible. Pour tenter de résoudre ce problème, nous avons proposé une nouvelle architecture.

CHAPITRE 7

Modèle, commande et expérimentation d'un *freewing*

Sommaire

7.1	Motivation	113
7.2	Réflexion autour de diverses architectures	114
7.3	Design et modélisation d'un drone : Colibri	117
7.3.1	Description de l'architecture	117
7.3.2	Modélisation	120
7.4	Estimation d'état	127
7.4.1	Placement des capteurs	127
7.4.2	Estimation de la vitesse angulaire	128
7.4.3	Estimation de l'état de l'aile	130
7.4.4	Estimation de l'orientation du fuselage	131
7.5	Inversion non-linéaire incrémentale de la dynamique du drone	132
7.6	Stabilisation du fuselage	132
7.7	Expérimentations	133
7.8	Vol avec un contrôleur INDI unifié	134
7.9	Conclusion	136

7.1 Motivation

Les nombreux avantages d'une architecture *tailsitter* nous ont poussé à nous intéresser à leur stabilisation. Nous avons toutefois observé une limitation dans leur usage. Effectivement, l'aile change d'incidence en fonction de la vitesse du drone ou du vent environnant et toute charge utile se trouve en rotation avec l'aile. Ce comportement est inapproprié pour une caméra ou une sonde Pitot, lesquelles doivent pouvoir maintenir une direction constante. Ainsi, nous avons pensé à une nouvelle

architecture, à mi-chemin entre un *tailsitter* et un *tiltwing*, appelée *freewing*. Cette dernière conserve les avantages du *tiltwing* en ce qu'elle permet d'avoir un fuselage que l'on peut maintenir constamment horizontal. Elle s'inspire des *tailsitters* en permettant de modifier librement l'incidence de l'aile pour se désensibiliser des turbulences. L'aile étant libre de s'orienter vis-à-vis du fuselage, une augmentation de portance engendrée par une augmentation de la vitesse du flux d'air (turbulences), modifiera l'angle d'incidence de l'aile ce qui entraînera une diminution de la portance et donc un nouvel équilibre.

De plus, le fuselage permettra l'emport d'une charge utile fragile (car elle pourra être maintenue horizontale en toutes circonstances), mais aussi de capteurs et de la batterie. L'ensemble de la masse du drone sera concentré sur le fuselage de manière à minimiser l'inertie de l'aile et à faciliter sa rotation face à des turbulences faibles.

7.2 Réflexion autour de diverses architectures

La maquette utilisée dans le chapitre 4 nous a permis d'expérimenter le comportement de l'aile en fonction de la position du centre de rotation.

Nous avons réalisé un montage permettant de modifier la position du centre de rotation par rapport au centre de gravité (figure 7.1). Cela impacte la stabilité de l'aile, mais aussi son comportement face au vent.

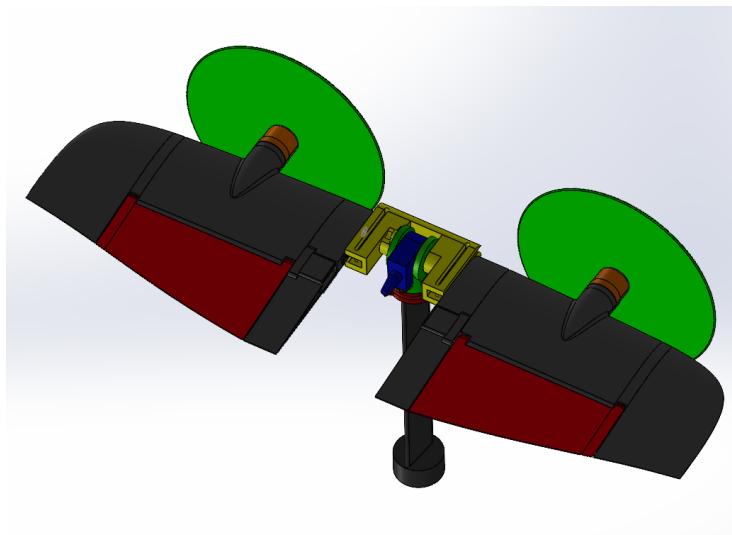


FIGURE 7.1 – Montage de la maquette du chapitre 4 avec un réglage de la position du centre de rotation (pièce jaune).

La pièce jaune de la Figure 7.1 assure la liaison entre les demi-ailes (droite et

gauche) et le système de rotation central. De plus, la pièce est conçue pour permettre un glissement dans le plan des ailes, ce qui a pour conséquence de déplacer le centre de gravité de l'ensemble par rapport au centre de rotation, ce dernier étant bloqué. Nous observons sur la Figure 7.2 que le réglage permet un débattement d'environ 30 mm. Le montage est verrouillé par deux vis par aile, assurant la rigidité.

Le centre de gravité doit se trouver en arrière du centre de rotation pour assurer que l'aile soit verticale sans commande.

De ce montage et en utilisant la loi de stabilisation de la Figure 4.4, nous avons observé l'intérêt de positionner un fuselage pendulaire fixé en rotation libre. Ainsi l'aile se trouve libre de tourner autour de l'axe de liaison pour obtenir l'incidence nécessaire à la stabilisation du drone dans la phase de vol considérée. Nous avons choisi d'expérimenter le cas où la charge utile, ainsi que la batterie, se situent sur le fuselage de manière à diminuer l'inertie de l'aile : cela permet d'augmenter sa réactivité. Ainsi, la maquette de la Figure 7.1 était alimentée par une batterie au plomb installée au sol avec des câbles alimentant le système installé au niveau de la fixation. Ces câbles génèrent un moment résistant, permettant d'approcher au mieux le système réel.

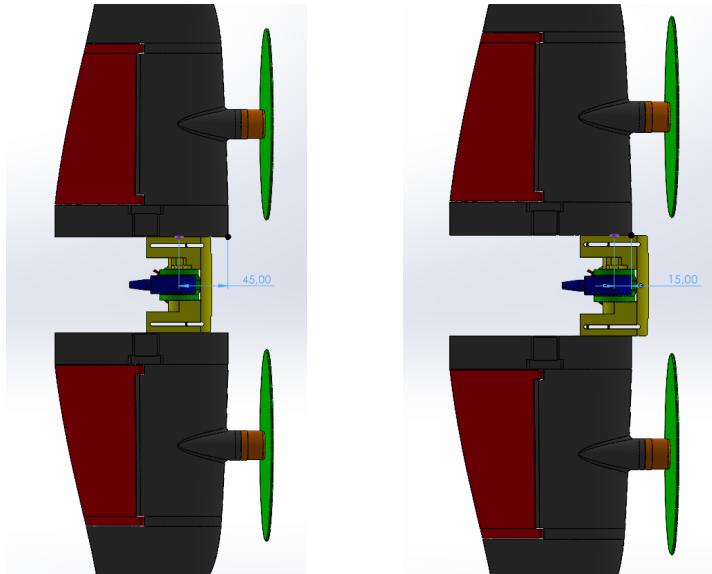


FIGURE 7.2 – Positionnement avant et arrière du centre de rotation par rapport au centre de gravité.

Au niveau de l'actionnement, le parti pris a été d'utiliser un mécanisme similaire à celui de DarkO, étant donné les études préliminaires réalisées sur la modélisation, l'étude des couplages et des saturations. Toutefois, il existait d'autres possibilités.

Nous pouvions monter les moteurs sur des nacelles pour supprimer les élévons et générer de la poussée vectorielle. Ce mécanisme ne supprime aucun actionneur car le servomoteur présent sur l'élévon se retrouve à actionner la nacelle. Ce servomoteur doit être dimensionné pour vaincre les effets gyroscopiques de l'hélice, ce qui peut diminuer la réactivité de l'actionnement. De plus, le mécanisme de nacelle augmente le nombre de pièces en mouvement, ce qui pose des problèmes de fiabilité. L'absence de commande aérodynamique ne permet pas de vol plané en cas de panne moteur, donc le contrôle du drone ne peut pas être assuré.

Une autre possibilité était de motoriser l'intégralité d'une demi-aile, ce qui permettait de modifier la direction de poussée de moteur, mais aussi de modifier l'incidence de l'aile. Toutefois, considérant les dimensions de l'aile, les servomoteurs doivent être dimensionnés au vu des effets gyroscopiques des hélices, couplés à l'inertie de la demi aile. Cela aurait induit des servomoteurs puissants qui auraient alourdi la structure.

Nous avons aussi évalué la possibilité de rendre les deux ailes droite et gauche indépendantes, ce qui aurait ajouté un degré de liberté dans la dynamique du drone. Ce fonctionnement permet de se soustraire de l'impact d'une différence de vitesse air sur chacune des demi-ailes. Toutefois, il est nécessaire de mesurer l'orientation de chacune des demi-ailes, ce qui impliquait un capteur en plus dans un espace assez contraint.

Bien que nous ayons convergé vers une architecture d'actionnement proche de celle de DarkO, nous avons réalisé une maquette avec une installation des moteurs en H (voir Figure 7.3).

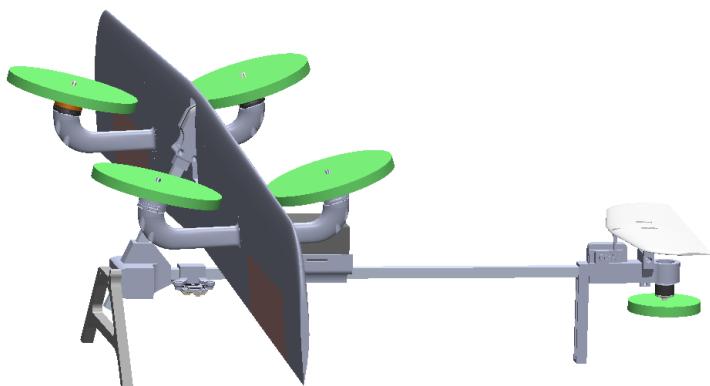


FIGURE 7.3 – Maquette de Colibri avec un montage des moteurs en H.

Cette architecture permet d'utiliser un différentiel moteur entre ceux placés sous l'aile et ceux sur l'aile, pour contrôler le moment de tangage du drone. Le principal intérêt est la suppression de la nécessité d'utiliser les élévons lors de la transition. Les élévons sont utilisés uniquement lors d'un vol plané. Effectivement, l'installation des moteurs permet d'avoir un contrôle total de la voilure (de manière similaire à un quadrirotor). Un angle a été ajouté lors de l'installation du moteur pour augmenter l'efficacité sur la rotation, autour de l'axe vertical en stationnaire. Nous n'avons pas poursuivi dans cette direction, au profit de l'architecture linéaire plus alignée sur nos précédents travaux.

7.3 Design et modélisation d'un drone : Colibri

7.3.1 Description de l'architecture

Le drone Colibri est dérivé d'un drone *tailsitter* qui génère de la portance pendant le vol d'avancement. Cette aile possède plusieurs actionneurs : quatre moteurs u_i , $i = 1, 2, 3, 4$ et deux élévons δ_l et δ_r . Nous pouvons définir le vecteur de contrôle \mathbf{u}_W de l'aile d'après la Figure 7.4 comme :

$$\mathbf{u}_W = [u_1 \ u_2 \ u_3 \ u_4 \ \delta_l \ \delta_r]^\top. \quad (7.1)$$

Un fuselage relié par un pivot est fixé au centre aérodynamique de l'aile. Ce fuselage supporte l'autopilote, la batterie, un moteur et un empennage pour le maintenir horizontal.

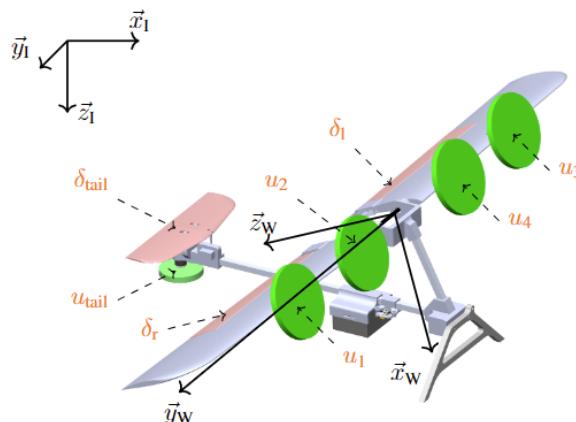


FIGURE 7.4 – Les repères, inertiel (I) et aile (W), attachés à l'architecture de Colibri.

Dans la Figure 7.4, toutes les surfaces de contrôle aérodynamiques sont représentées en rose et les hélices en vert. Trois repères sont attachés au drone tels que : (I) est un repère inertiel NED lié à la surface de la terre, (W) est un repère attaché à l'aile du drone et (F) est un repère attaché au fuselage du drone.

Certaines des dimensions caractéristiques sont indiquées dans la Table 7.1. Notons que les moteurs sont positionnés symétriquement sur l'aile, ce qui signifie que la position peut être décrite en se concentrant sur un seul côté.

Paramètres	Valeurs	Unités
m_W (masse de l'aile)	0.53	kg
m_F (masse du fuselage)	1.17	kg
$\mathbf{J}_W = diag(J_x^W, J_y^W, J_z^W)$	$diag(0.1677, 0.0052, 0.1634)$	$kg\ m^2$
$\mathbf{J}_F = diag(J_x^F, J_y^F, J_z^F)$	$diag(0.0191, 0.0161, 0.0343)$	$kg\ m^2$
k_p (coefficients hélice fuselage)	5.13e-6	$kg\ m$
k_f (coefficients de poussée des hélices)	5.13e-6	$kg\ m$
k_m (coefficients de moment des hélices)	2.640e-7	$kg\ m^2$
ξ_f (génération de force élévon)	0.48	—
ξ_m (Génération de moment élévon)	0.93	—
\mathbf{d}_{MO_W}	$[0.383, 0, -0.167]^\top$	m
\mathbf{d}_{GO_W}	$[0.052, 0, -0.171]^\top$	m

TABLE 7.1 – Paramètres numériques du modèle Colibri.

Nous avons déjà conçu un système de rotation libre dans le chapitre 4.1, toutefois il était fixé à un banc de test. Pour Colibri, il est nécessaire de concevoir une liaison robuste, mais légère pour être embarquée sur le drone.

Nous avons donc conçu un système de rotation basé sur deux roulements à aiguille (en violet sur l'image 7.5) et deux butées à billes F6-12M pour minimiser le couple résistant (en jaune sur l'image 7.5). Les butées à billes sont bloquées par des pièces de verrouillage (en bleu sur l'image 7.5) obtenues par impression 3D stéréolithographique (SLA). Ces verrouilleurs pincent le tube en carbone pour éviter qu'il coulisse de droite à gauche.

L'impression 3D SLA est un procédé de photopolymérisation qui utilise un laser UV pour polymériser une résine liquide en plastique durci. Cette technologie permet d'obtenir des pièces d'une grande précision et plus résistantes que l'impression 3D à dépôt de fil fondu (FDM).

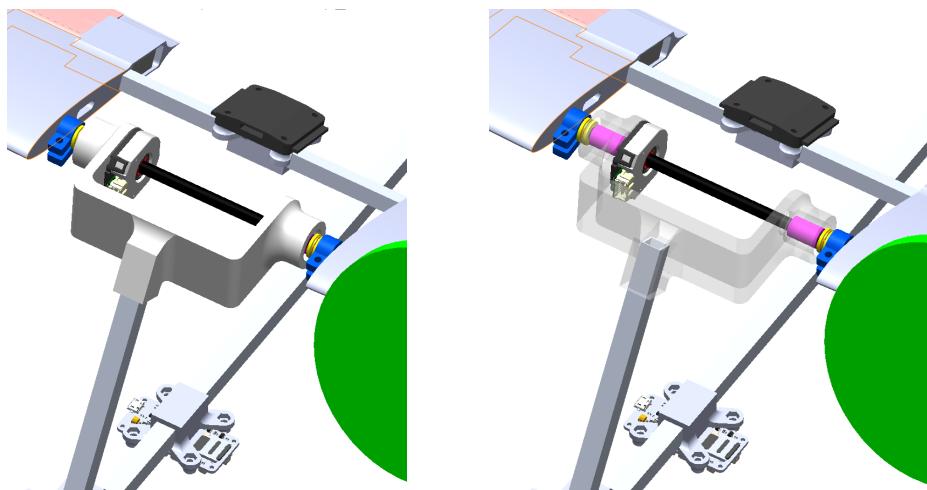


FIGURE 7.5 – Modèle 3D de la liaison.

L'encodeur rotatif, CUI Devices AMT22 est fixé au bâti qui maintient les roulements (figure 7.6 gauche). Un système de pince est installé sur le tube en carbone et permet au capteur de mesurer l'angle de rotation de l'ensemble aile par rapport au fuselage.

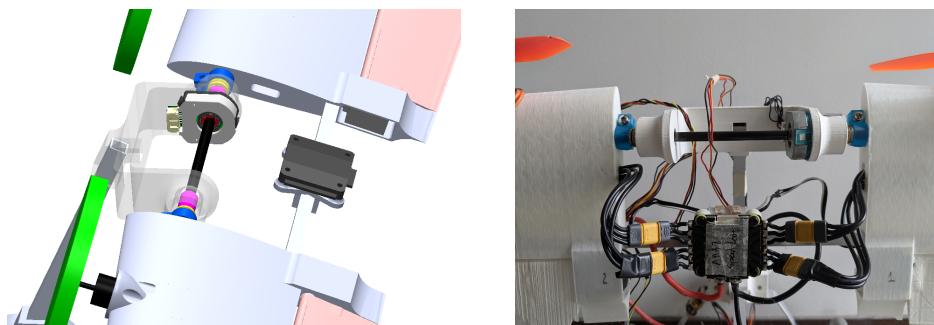


FIGURE 7.6 – Visualisation de l'encodeur rotatif sur le modèle 3D (gauche) et modèle réel (droite).

Le montage réel est proposé sur la Figure 7.6 de droite. Nous observons le montage du contrôleur moteur avec les fils qui vont jusqu'au moteur (3 par moteur soit un total de 12 fils) et les câbles de commande de l'ESC, les câbles de l'encodeur rotatif et les câbles de commandes des servomoteurs (3 fils, car les servomoteurs sont chainés) et enfin les câbles de l'IMU installée sur l'aile.

7.3.2 Modélisation

La modélisation est basée sur les résultats de [Udwadia 2006, Équation (2.15)]. Elle est obtenue à partir de l'équation de Lagrange décrivant le mouvement d'un système à chaque instant t :

$$\mathbf{M}(\mathbf{x}, t)\ddot{\mathbf{x}} = \mathbf{Q}(\mathbf{x}, \dot{\mathbf{x}}, t) \quad (7.2)$$

Cette équation correspond au système non contraint, où les n coordonnées de \mathbf{x} sont indépendantes les unes vis-à-vis des autres ou du moins traitées comme telles. Nous allons adjoindre à ce système non contraint m contraintes :

$$\varphi_i(\mathbf{x}, \dot{\mathbf{x}}, t) = 0, \quad i = 1, 2, \dots, m. \quad (7.3)$$

En supposant que ces contraintes soient assez lisses, nous pouvons les différencier (7.3) par rapport au temps pour obtenir :

$$\mathbf{A}(\mathbf{x}, \dot{\mathbf{x}}, t)\ddot{\mathbf{x}} = \mathbf{B}(\mathbf{x}, \dot{\mathbf{x}}, t). \quad (7.4)$$

En utilisant (7.2) et (7.4), nous pouvons obtenir :

$$\begin{bmatrix} (\mathbb{I} - \mathbf{A}^\dagger \mathbf{A})\mathbf{M} \\ \mathbf{A} \end{bmatrix} \ddot{\mathbf{x}} = \begin{bmatrix} (\mathbb{I} - \mathbf{A}^\dagger \mathbf{A})\mathbf{Q} \\ \mathbf{B} \end{bmatrix}. \quad (7.5)$$

L'équation (7.5) nous fournit l'équation de mouvement d'un système multicorps contraint :

$$\ddot{\mathbf{x}} := \hat{\mathbf{M}}^\dagger \begin{bmatrix} \mathbf{Q} \\ \mathbf{B} \end{bmatrix} = \begin{bmatrix} (\mathbb{I} - \mathbf{A}^\dagger \mathbf{A})\mathbf{M} \\ \mathbf{A} \end{bmatrix}^\dagger \begin{bmatrix} \mathbf{Q} \\ \mathbf{B} \end{bmatrix} \quad (7.6)$$

dont l'expression est valide tant que $\hat{\mathbf{M}}$ a un rang complet et où \mathbf{A} , \mathbf{M} , \mathbf{Q} et \mathbf{B} sont décrits par la suite. L'algorithme de calcul de ces matrices se trouve dans [Udwadia 2012].

D'après [Tangirala 2015], nous avons une définition d'une réalisation minimale :

Définition 7.3.1 (Réalisation minimale). Une réalisation est dite "minimale" si elle décrit le système avec le nombre minimum d'états. Le nombre minimum de variables d'état nécessaire pour décrire un système est égal à l'ordre de l'équation différentielle.

Nous utiliserons les quaternions $\mathbf{q} = [\eta \ \boldsymbol{\epsilon}^\top]^\top \in \mathbb{S}^3 := \{\mathbf{q} \in \mathbb{R}^4 : |\mathbf{q}| = 1\}$ pour

représenter les orientations des deux corps. La matrice de rotation qui en résulte $\mathbf{R}(\mathbf{q}) \in SO(3) := \{\mathbf{R} \in \mathbb{R}^{3 \times 3} : \mathbf{R}^\top \mathbf{R} = \mathbb{I}_3 \text{ et } \det(\mathbf{R}) = 1\}$ est définie de manière unique comme $\mathbf{R}(\mathbf{q}) := \mathbb{I}_3 + 2\eta [\epsilon]_x + 2[\epsilon]_x^2 = [\mathbf{R}_1 \ \mathbf{R}_2 \ \mathbf{R}_3]$.

D'après les Figures 7.4 et 7.7, nous définissons les vecteurs $\mathbf{p}_F = \overrightarrow{O_I O_F}$, $\mathbf{p}_W = \overrightarrow{O_I O_W}$, $\mathbf{d}_{FW} = \overrightarrow{O_F O_W}$ satisfaisant $\mathbf{d}_{FW} = \mathbf{p}_W - \mathbf{p}_F$ et $\mathbf{d}_{MO_W} = \overrightarrow{MO_W}$, $\mathbf{d}_{GO_W} = \overrightarrow{GO_W}$.

Le vecteur d'état global est $(\mathbf{x}, \mathbf{v}) \in \mathbb{R}^{28}$ avec $\mathbf{x} = (\mathbf{p}_W, \mathbf{q}_W, \mathbf{p}_F, \mathbf{q}_F) \in \mathbb{R}^{14}$ et $\mathbf{v} = (\mathbf{v}_W, \dot{\mathbf{q}}_W, \mathbf{v}_F, \dot{\mathbf{q}}_F) = (\dot{\mathbf{p}}_W, \dot{\mathbf{q}}_W, \dot{\mathbf{p}}_F, \dot{\mathbf{q}}_F) = \dot{\mathbf{x}} \in \mathbb{R}^{14}$. Nous avons $\mathbf{v}_W = \dot{\mathbf{p}}_W \in \mathbb{R}^3$ qui représente la vitesse linéaire de l'aile dans le repère inertiel; $\dot{\mathbf{q}}_W \in \mathbb{R}^4$ est la dérivée du quaternion. $\mathbf{q}_W \in \mathbb{R}^4$ représente l'orientation de l'aile; $\mathbf{v}_F = \mathbf{p}_F \in \mathbb{R}^3$ est la vitesse linéaire du fuselage dans le repère inertiel et $\dot{\mathbf{q}}_F \in \mathbb{R}^4$ est la dérivée du quaternion. $\mathbf{q}_F \in \mathbb{R}^4$ représente l'orientation du fuselage. Nous observons que le vecteur d'état n'est pas minimal d'après la définition 7.3.1. Effectivement, le système possède sept degrés de liberté (six degrés pour l'aile et un degré pour le fuselage) et nous le représentons avec un vecteur d'état de dimension quatorze.

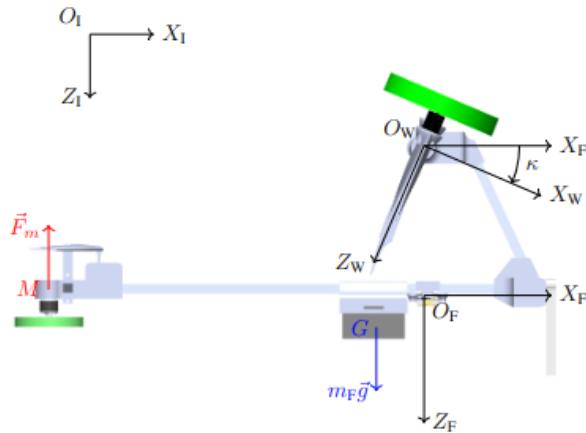


FIGURE 7.7 – Repère inertiel (I), du fuselage (F) et de l'aile (W) et forces agissant sur le drone Colibri.

Nous devons aussi définir le vecteur de commande de dimension huit $\mathbf{u} = (u_1, u_2, u_3, u_4, \delta_l, \delta_r, u_{tail}, \delta_{tail})$, où les actionneurs sont repérés en vert et rose sur la Figure 7.4. Les actionneurs verts sont des groupes moteurs-hélices et les actionneurs rose sont des surfaces aérodynamiques.

Il convient de noter que la vitesse angulaire $\boldsymbol{\omega} \in \mathbb{R}^3$ peut être obtenue à partir de la dérivée du quaternion $\dot{\mathbf{q}}$ en utilisant l'équation [Udwadia 2012, équation (2.7)]

rappelée ici :

$$\boldsymbol{\omega} = \mathbf{H}(\mathbf{q})\dot{\mathbf{q}}$$

où $\mathbf{H}(\mathbf{q}) \in \mathbb{R}^{3 \times 4}$ est une matrice définie par $\mathbf{H}(\mathbf{q}) = 2 \begin{bmatrix} -\epsilon & \eta \mathbb{I}_3 - [\boldsymbol{\epsilon}]_\times \end{bmatrix}$. Pour dériver les équations du mouvement, rappelons que $\mathbf{R}_i(\mathbf{q}) \in \mathbb{R}^3, i = 1, 2, 3$ sont les trois colonnes d'une matrice de rotation associée au quaternion q , lesquelles définissent les matrices $\mathbf{L}_i^W(\mathbf{q}_W) = \frac{\partial \mathbf{R}_i}{\partial q}(\mathbf{q}_W) \in \mathbb{R}^{3 \times 4}$, $\mathbf{L}_i^F(\mathbf{q}_F) = \frac{\partial \mathbf{R}_i}{\partial q}(\mathbf{q}_F) \in \mathbb{R}^{3 \times 4}$ et $\mathbf{L}_{O_F^W} = \sum_{i=1}^3 \mathbf{d}_{FW}(i) \mathbf{L}_i^F(\mathbf{q}_F)$, $i \in 1, 2, 3$, où $\mathbf{d}_{FW}(i)$ désigne la i -ème composante du vecteur $\mathbf{d}_{FW} = \mathbf{p}_W - \mathbf{p}_F$. Comme le point O_W est situé au centre de rotation de l'aile, la distance \mathbf{d}_{FW} est une constante, puisque O_W et O_F appartiennent au même solide (le fuselage).

Nous en déduisons, avec l'homogénéité, $\dot{\mathbf{L}}_{O_F^W} = \sum_{i=1}^3 \mathbf{d}_{FW}(i) \mathbf{L}_i^F(\dot{\mathbf{q}}_F)$. Avec ces définitions, les matrices de (7.6) sont :

$$\mathbf{M} = \begin{bmatrix} m_W \mathbb{I}_3 & \mathbb{0}_{3 \times 4} & \mathbb{0}_3 & \mathbb{0}_{3 \times 4} \\ \mathbb{0}_{4 \times 3} & \mathbf{H}_W^\top \mathbf{J}_W \mathbf{H}_W & \mathbb{0}_{4 \times 3} & \mathbb{0}_4 \\ \mathbb{0}_3 & \mathbb{0}_{3 \times 4} & m_F \mathbb{I}_3 & \mathbb{0}_{3 \times 4} \\ \mathbb{0}_{4 \times 3} & \mathbb{0}_4 & \mathbb{0}_{4 \times 3} & \mathbf{H}_F^\top \mathbf{J}_F \mathbf{H}_F \end{bmatrix} \in \mathbb{R}^{14 \times 14}, \quad (7.7)$$

où nous avons noté $\mathbf{H}_W = \mathbf{H}(\mathbf{q}_W)$, $\mathbf{H}_F = \mathbf{H}(\mathbf{q}_F)$, et

$$\mathbf{Q} = \begin{bmatrix} m_W g e_3 + \mathbf{R}(\mathbf{q}_W) \mathbf{F}_W(x, \mathbf{u}) \\ -2 \dot{\mathbf{H}}_W^\top \mathbf{J}_W \dot{\mathbf{H}}_W \dot{\mathbf{q}}_W + \mathbf{H}_W^\top \mathbf{M}_W(x, \mathbf{u}) \\ m_F g e_3 + \mathbf{R}(\mathbf{q}_F) \mathbf{F}_F(\mathbf{u}) \\ -2 \dot{\mathbf{H}}_F^\top \mathbf{J}_F \dot{\mathbf{H}}_F \dot{\mathbf{q}}_F + \mathbf{H}_F^\top \mathbf{M}_F(\mathbf{u}) \end{bmatrix} \in \mathbb{R}^{14}, \quad (7.8)$$

où $\dot{\mathbf{H}}_W$ désigne $\mathbf{H}(\dot{\mathbf{q}}_W)$, coïncidant avec la dérivée temporelle de $\mathbf{H}(\mathbf{q}_W)$ et $\dot{\mathbf{H}}_F$ désigne $\mathbf{H}(\dot{\mathbf{q}}_F)$, coïncidant avec la dérivée temporelle de $\mathbf{H}(\mathbf{q}_F)$. De plus, $\mathbf{F}_W(x)$ et $\mathbf{M}_b(x)$ représentent respectivement toutes les forces et tous les moments agissant sur l'aile. Leur expression est tirée de [Lustosa 2019, équations (45) et (57)] où est développée la ϕ -théorie, paramétrage qui permet de soustraire les angles classiques d'incidence et de dérapage et d'éviter la singularité du vol stationnaire. De manière similaire aux équations (2.6) et (2.7) exprimant les forces et les moments agissant sur DarkO, nous pouvons exprimer $\mathbf{F}_W(x)$ et $\mathbf{M}_b(x)$ relatifs à l'aile de Colibri.

Chaque hélice génère une poussée \mathbf{T}_i orientée dans la direction $-zW$ du repère

de l'aile et un moment \mathbf{N}_i selon le même axe :

$$\mathbf{T}_i := \begin{bmatrix} \tau_i \\ 0 \\ 0 \end{bmatrix} := \begin{bmatrix} k_f \omega_i^2 \\ 0 \\ 0 \end{bmatrix}, \quad \mathbf{N}_i := (-1)^i \frac{k_m}{k_f} \mathbf{T}_i, \quad i = 1, 2, 3, 4, \text{tail.} \quad (7.9)$$

La position de chaque élévon $\delta_i \in \mathbb{R}$ est assignée par un servomoteur qui impose un niveau d'efficacité (en termes de déviation du courant d'air) quantifié par deux matrices antisymétriques :

$$\Delta_i^f := \begin{bmatrix} 0 & 0 & \xi_f \delta_i \\ 0 & 0 & 0 \\ -\xi_f \delta_i & 0 & 0 \end{bmatrix}, \quad \Delta_i^m := \begin{bmatrix} 0 & 0 & \xi_m \delta_i \\ 0 & 0 & 0 \\ -\xi_m \delta_i & 0 & 0 \end{bmatrix}, \quad i = l, r. \quad (7.10)$$

Les paramètres constants k_f , k_m , ξ_f , ξ_m apparaissant dans (2.4) et (2.5) sont listés dans la Table 7.1.

Avec les quantités ci-dessus, nous pouvons nous inspirer de la dynamique donnée dans [Lustosa 2019, eqns (97), (98)] et exprimer $\mathbf{F}_W(x, u)$ et $\mathbf{M}_W(x, u)$ dans (7.8) comme :

$$\begin{aligned} \mathbf{F}_W(x, u) := & \sum_{i=1}^4 \mathbf{T}_i + \frac{S_{\text{wet}}}{4S_p} \Phi^{(\text{fv})} \left((\Delta_1^f - \mathbb{I}_3)(\mathbf{T}_1 + \mathbf{T}_2) + (\Delta_2^f - \mathbb{I}_3)(\mathbf{T}_3 + \mathbf{T}_4) \right) \\ & + \frac{1}{4} \rho S \Phi^{(\text{fv})} (\Delta_1^f + \Delta_2^f - 2\mathbb{I}_3) \|\mathbf{v}_b\| \mathbf{v}_b \\ & + \frac{1}{4} \rho S \Phi^{(\text{mv})} (\Delta_1^f + \Delta_2^f - 2\mathbb{I}_3) \mathbf{B} \|\mathbf{v}_b\| \boldsymbol{\omega}_b, \end{aligned} \quad (7.11)$$

$$\begin{aligned} \mathbf{M}_W(x, u) := & \sum_{i=1}^4 \mathbf{N}_i + + \begin{bmatrix} p_x \\ p_{y1} \\ 0 \end{bmatrix} \times \mathbf{T}_1 + \begin{bmatrix} p_x \\ p_{y2} \\ 0 \end{bmatrix} \times \mathbf{T}_2 + \begin{bmatrix} p_x \\ -p_{y3} \\ 0 \end{bmatrix} \times \mathbf{T}_3 + \begin{bmatrix} p_x \\ -p_{y4} \\ 0 \end{bmatrix} \times \mathbf{T}_4 \\ & - \frac{S_{\text{wet}}}{4S_p} \left(\mathbf{B} \Phi^{(\text{mv})} (\Delta_1^m - \mathbb{I}_3) + \begin{bmatrix} 0 \\ a_y \\ 0 \end{bmatrix} \times \Phi^{(\text{fv})} (\Delta_1^m + \mathbb{I}_3) \right) (\mathbf{T}_1 + \mathbf{T}_2) \\ & - \frac{S_{\text{wet}}}{4S_p} \left(\mathbf{B} \Phi^{(\text{mv})} (\Delta_2^m - \mathbb{I}_3) + \begin{bmatrix} 0 \\ -a_y \\ 0 \end{bmatrix} \times \Phi^{(\text{fv})} (\Delta_2^m + \mathbb{I}_3) \right) (\mathbf{T}_3 + \mathbf{T}_4) \\ & + \frac{1}{4} \rho S \left(\left(\begin{bmatrix} 0 \\ a_y \\ 0 \end{bmatrix} \times \Phi^{(\text{fv})} + \mathbf{B} \Phi^{(\text{mv})} \right) \Delta_1^m \right. \\ & \left. + \left(\begin{bmatrix} 0 \\ -a_y \\ 0 \end{bmatrix} \times \Phi^{(\text{fv})} + \mathbf{B} \Phi^{(\text{mv})} \right) \Delta_2^m - 2 \mathbf{B} \Phi^{(\text{mv})} \right) \|\mathbf{v}_b\| \mathbf{v}_b \\ & + \frac{1}{4} \rho S \left(\left(\begin{bmatrix} 0 \\ a_y \\ 0 \end{bmatrix} \times \Phi^{(\text{mv})} + \mathbf{B} \Phi^{(\text{m}\omega)} \right) \Delta_1^m \right. \end{aligned}$$

$$+ \left(\begin{bmatrix} 0 \\ -a_y \\ 0 \end{bmatrix}_x \Phi^{(mv)} + B\Phi^{(m\omega)} \right) \Delta_2^m - 2B\Phi^{(m\omega)} \right) B \|v_b\| \omega_b, \quad (7.12)$$

où $v_b := R^\top(q)(v - w)$ représente la vitesse de l'air vue par le drone et exprimée dans le repère du corps. Dans [Lustosa 2019], la valeur $\|v_b\|$, apparaissant dans les expressions de F_b et M_b , est remplacée par la valeur $\eta = \sqrt{\|v_b\|^2 + \mu c^2 \|\omega_b\|^2}$, avec $\mu \in \mathbb{R}$ étant un paramètre lié à l'identification du modèle. Toutefois, dans le cas de Colibri, l'identification fournit $\mu = 0$.

Dès lors, nous présentons ici une description simplifiée. La matrice des coefficients aérodynamiques constants $\Phi := \begin{bmatrix} \Phi^{(fv)} & \Phi^{(mv)\top} \\ \Phi^{(mv)} & \Phi^{(m\omega)} \end{bmatrix} \in \mathbb{R}^{6 \times 6}$, est définie dans [Olszanecki Barth 2020, eqs. (6)–(9)] comme $\Phi^{(fv)} := \text{diag}(C_d, C_y, C_\ell)$ et

$$\left[\begin{array}{c|c} \Phi^{(mv)} & \Phi^{(m\omega)} \end{array} \right] := \left[\begin{array}{ccc|ccc} 0 & 0 & 0 & 0.1396 & 0 & 0.0573 \\ 0 & 0 & -\frac{\Delta_r}{c} C_\ell & 0 & 0.6358 & 0 \\ 0 & 0 & 0 & 0.0405 & 0 & 0.0019 \end{array} \right].$$

Enfin, $F_F = F_m$ et M_F représentent respectivement l'ensemble des forces et moments non gravitationnels agissant sur le fuselage exprimés dans le repère O_W . En particulier, $F_m = -k_p u_{tail}^2$ est la force générée par le moteur situé à l'arrière du fuselage et u_{tail} est la vitesse de rotation du moteur.

Nous utilisons le théorème fondamental de la statique et nous isolons le fuselage. Les deux moments qui agissent sur le fuselage sont le couple dû à la force de poussée du moteur (en rouge sur la Figure (7.7)) et le couple dû à la position du centre de gravité du fuselage (en bleu sur la Figure (7.7)). Il en résulte l'équation suivante :

$$M_F = m_F g e_3 \times d_{GO_W} + F_m \times d_{MO_W}, \quad (7.13)$$

où d_{MO_W} est la distance entre l'emplacement du moteur et le centre de rotation et d_{GO_W} est la distance entre l'emplacement du centre de gravité du fuselage et le centre de rotation.

L'ensemble des contraintes, associé à la non-minimalité de l'état (x, v) et à la

liaison pivot entre les deux corps, est donné par :

$$\begin{cases} \varphi_1 := \mathbf{q}_W^\top \mathbf{q}_W - 1 = 0, & \varphi_2 := \mathbf{q}_F^\top \mathbf{q}_F - 1 = 0, \\ \varphi_3 := \mathbf{R}_2(\mathbf{q}_W)^\top \mathbf{R}_3(\mathbf{q}_F) = 0, & \varphi_4 := \mathbf{R}_2(\mathbf{q}_W)^\top \mathbf{R}_1(\mathbf{q}_F) = 0, \\ \varphi_5 := \mathbf{p}_F + \mathbf{d}_{FA} + \mathbf{p}_W = 0 & \end{cases} \quad (7.14)$$

Les deux premières contraintes imposent la norme unitaire des quaternions \mathbf{q}_F et \mathbf{q}_W . Les troisième et quatrième contraintes sont liées à une contrainte pivot, c'est-à-dire que l'orthogonalité de deux vecteurs est imposée. La dernière est une contrainte de position, de sorte que le point du centre de rotation de l'aile coïncide avec le point défini dans le fuselage. Cette contrainte est basée sur une fermeture géométrique tridimensionnelle. Il est plus commode d'exprimer l'ensemble des contraintes sous la forme d'un système dynamique stable convergeant vers zéro. Nous convertissons donc chacune des contraintes sous la forme suivante :

$$\ddot{\varphi}_i + \delta_1 \dot{\varphi}_i + \delta_2 \varphi_i = 0, \quad i \in \{1, 2, 3, 4, 5\}, \quad (7.15)$$

avec les sélections $(\delta_1, \delta_2) = (0.5, 8)$ étant les coefficients d'un polynôme stable, de sorte que, quelle que soit la sélection $\varphi_i(0) = 0$, nous ayons $\lim_{t \rightarrow \infty} \varphi_i(t) = 0$. En différenciant deux fois les contraintes (7.14) et en les factorisant sous la forme de l'équation (7.4), nous obtenons l'expression de $\mathbf{A}(\mathbf{x}, \dot{\mathbf{x}})$ reportée dans l'équation (7.16) et de $\mathbf{B}(\mathbf{x}, \dot{\mathbf{x}})$ reportée dans l'équation (7.17).

$$\mathbf{A} = \begin{bmatrix} 0_{1 \times 3} & \mathbf{q}_W^\top & 0_{1 \times 3} & 0_{1 \times 4} \\ 0_{1 \times 3} & 0_{1 \times 4} & 0_{1 \times 3} & \mathbf{q}_F^\top \\ 0_{1 \times 3} & \mathbf{R}_3(\mathbf{q}_F)^\top \mathbf{L}_2^W(\mathbf{q}_W) & 0_{1 \times 3} & \mathbf{R}_2(\mathbf{q}_W)^\top \mathbf{L}_3^F(\mathbf{q}_F) \\ 0_{1 \times 3} & \mathbf{R}_1(\mathbf{q}_F)^\top \mathbf{L}_2^W(\mathbf{q}_W) & 0_{1 \times 3} & \mathbf{R}_2(\mathbf{q}_W)^\top \mathbf{L}_3^F(\mathbf{q}_F) \\ \mathbb{I}_3 & \mathbf{L}_{O_F^W} & -\mathbb{I}_3 & 0_{3 \times 4} \end{bmatrix} \quad (7.16)$$

$$\mathbf{B} = \begin{bmatrix} -\delta_1 \mathbf{q}_W^\top \dot{\mathbf{q}}_W - \frac{\delta_2}{2} (\mathbf{q}_W^\top \mathbf{q}_W - 1) - \dot{\mathbf{q}}_W^\top \dot{\mathbf{q}}_W \\ -\delta_1 \mathbf{q}_F^\top \dot{\mathbf{q}}_F - \frac{\delta_2}{2} (\mathbf{q}_F^\top \mathbf{q}_F - 1) - \dot{\mathbf{q}}_F^\top \dot{\mathbf{q}}_F \\ B_3 \\ B_4 \\ \dot{\mathbf{L}}_{O_F^W} \dot{\mathbf{q}}_F - \delta_1 (\mathbf{v}_W + \dot{\mathbf{L}}_{O_F^W} \dot{\mathbf{q}}_W - \mathbf{v}_F) - \delta_1 \varphi_5 \end{bmatrix}. \quad (7.17)$$

avec

$$\begin{aligned} B_3 &= -\mathbf{R}_3(\mathbf{q}_F)^\top \dot{\mathbf{L}}_2^W \dot{\mathbf{q}}_W - \mathbf{R}_2(\mathbf{q}_W)^\top \dot{\mathbf{L}}_3^F \dot{\mathbf{q}}_F - 2\dot{\mathbf{q}}_W^\top \mathbf{L}_2^W{}^\top \mathbf{L}_3^F \dot{\mathbf{q}}_F - \delta_1(\mathbf{R}_3(\mathbf{q}_F)^\top \mathbf{L}_2^W \dot{\mathbf{q}}_W \\ &\quad + \mathbf{R}_2(\mathbf{q}_W)^\top \mathbf{L}_3^F \dot{\mathbf{q}}_F) - \delta_2 \varphi_3, \\ B_4 &= -\mathbf{R}_1(\mathbf{q}_F)^\top \dot{\mathbf{L}}_2^W \dot{\mathbf{q}}_W - \mathbf{R}_2(\mathbf{q}_W)^\top \dot{\mathbf{L}}_1^F \dot{\mathbf{q}}_F - 2\dot{\mathbf{q}}_W^\top \mathbf{L}_2^W{}^\top \mathbf{L}_1^F \dot{\mathbf{q}}_F - \delta_1(\mathbf{R}_1(\mathbf{q}_F)^\top \mathbf{L}_2^W \dot{\mathbf{q}}_W \\ &\quad + \mathbf{R}_2(\mathbf{q}_W)^\top \mathbf{L}_1^F \dot{\mathbf{q}}_F) - \delta_2 \varphi_4. \end{aligned}$$

Les résultats de la simulation en boucle fermée sont présentés à la Figure 7.8.

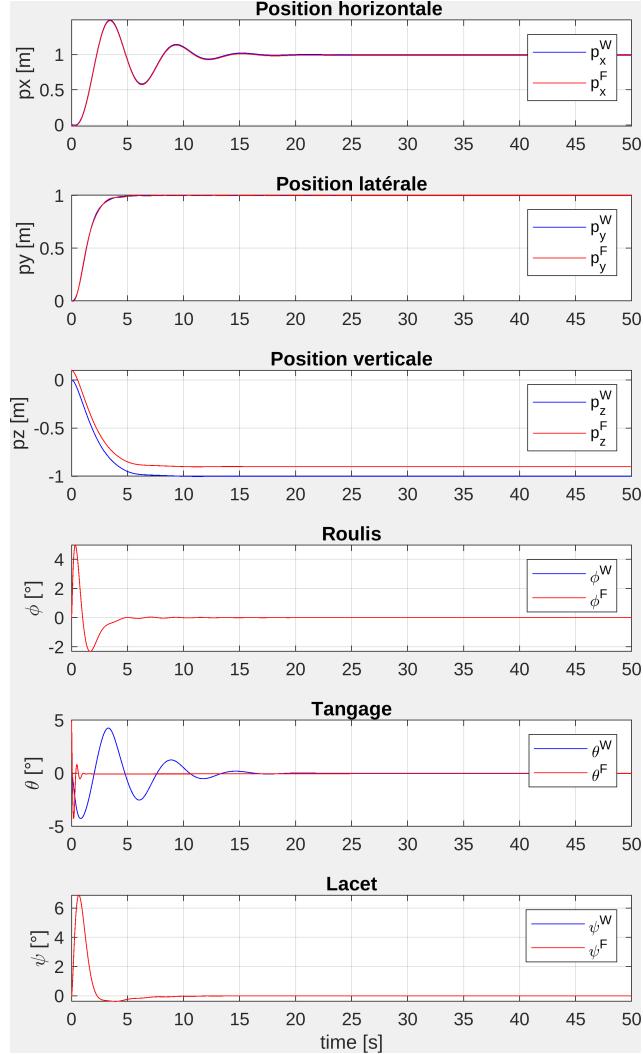


FIGURE 7.8 – Simulation de la position et de l'orientation du drone multicorps Colibri en boucle fermée avec un contrôleur à double boucle.

La simulation d'un drone reste complexe, car il est naturellement instable. Nous

avons choisi d'utiliser la loi de contrôle proposée dans 6.2.3 pour stabiliser le système. Ce contrôle stabilise l'aile. Une autre loi de contrôle basée sur une rétroaction proportionnelle-dérivée stabilise le fuselage pour le maintenir horizontal, développée dans la section 7.6.

En considérant les degrés de liberté de la liaison pivot, le couplage entre les deux corps est clairement visible sur les trois graphiques du bas. En effet, les angles de roulis et de lacet (ϕ_F, ψ_F) et (ϕ_W, ψ_W) du fuselage et de l'aile coïncident parfaitement, alors que les angles de tangage (θ_F, θ_W) sont différents.

Par la suite, nous souhaitons réaliser des expérimentations avec un système réel. Nous devons donc étudier la possibilité d'obtention de l'estimation des corps pour stabiliser le drone.

7.4 Estimation d'état

Un des problèmes engendré par cette architecture multicorps est l'estimation de l'état de chacun des composants. Pour stabiliser ce système de drone à deux corps, il est nécessaire de connaître la position et l'orientation des deux corps. Grâce à la liaison pivot entre l'aile et le fuselage, la différence entre l'orientation de l'aile et l'orientation du fuselage est simplement une rotation autour de l'axe de tangage de l'aile. Les deux autres orientations (roulis et lacet) coïncident. La position du centre de gravité du fuselage peut être déduite de la position du centre de gravité de l'aile et de l'angle entre le fuselage et l'aile. Cet angle est mesuré par un codeur rotatif en quadrature (CUI Devices AMT22, codeurs absolus, 12 bits, SPI), qui renvoie une mesure angulaire quantifiée avec un pas de 0.09° . Compte tenu de cette mesure angulaire, nous examinons ci-dessous l'estimation des informations relatives à la vitesse, afin de reconstruire l'état du drone.

7.4.1 Placement des capteurs

Une première question concerne l'emplacement des capteurs : l'IMU (accéléromètre, gyroscope et magnétomètre) peut être installée sur le fuselage ou sur l'aile. L'installation de l'IMU sur l'aile permet d'effectuer les mesures directement dans le référentiel souhaité, mais les mesures sont plus bruitées car l'IMU est attachée à la structure supportant les moteurs. Compte tenu de la taille des ailes, leur flexibilité peut générer des résonances et perturber les mesures.

L'installation de l'IMU sur le fuselage réduit les vibrations, mais implique que les mesures soient transformées dans le référentiel de l'aile. La transformation corres-

pondante peut être calculée à partir de la mesure de l'encodeur rotatif, qui fournit l'angle entre l'aile et le fuselage, ainsi qu'à partir des mesures prises avec le logiciel de CAO, qui fournissent des informations précises sur les distances entre les référentiels de l'aile et du fuselage. Notre choix final est de fixer l'IMU au fuselage. Une autre considération est que la carte de pilotage automatique, qui a déjà une IMU intégrée, est également supposée être connectée à la charge utile et à d'autres capteurs fixés sur le fuselage. Il s'agit donc de limiter le nombre de câbles au point de pivot pour les commandes des actionneurs et l'alimentation électrique.

7.4.2 Estimation de la vitesse angulaire

Comme expliqué ci-dessus, nous pouvons mesurer l'angle $\kappa \in \mathbb{R}$ entre l'aile et le fuselage à l'aide de l'encodeur rotatif. Ensuite, pour estimer la vitesse angulaire, nous utilisons un observateur haut gain proposé dans [Nicosia 1990] (voir également [Chitour 2002] pour l'utilisation d'observateurs haut gain pour estimer les dérivées temporelles). Cette méthode est préférable à une dérivée par différence finie, car les informations quantifiées générées par l'encodeur rotatif peuvent donner lieu à du bruit numérique dans les valeurs estimées de vitesse angulaire.

Désignons par $\kappa \in \mathbb{R}$ l'angle mesuré, par $\omega_\kappa := \dot{\kappa} \in \mathbb{R}$ sa dérivée à estimer, et par $\xi = [\kappa, \omega_\kappa]^\top \in \mathbb{R}^2$ leur juxtaposition en un seul vecteur. Notons également $\hat{\xi}$ l'estimation de ξ définie par :

$$\hat{\xi} = [\hat{\kappa}, \hat{\omega}_\kappa]^\top \in \mathbb{R}^2.$$

Suivant [Nicosia 1990], la dynamique de l'estimateur est donnée par :

$$\dot{\hat{\xi}} = \begin{bmatrix} 0 & 1 \\ 0 & 0 \end{bmatrix} \hat{\xi} + \begin{bmatrix} k_p \\ \frac{\epsilon_\kappa}{k_v} \\ \frac{k_v}{\epsilon_\kappa^2} \end{bmatrix} (\kappa - \hat{\kappa}), \quad (7.18)$$

où κ est la mesure angulaire récupérée du capteur, k_p et k_v sont deux gains scalaires positifs tels que l'équation caractéristique $s^2 + k_v s + k_p = 0$ ait des racines à partie réelle négative. Pour notre estimateur, nous avons choisi $k_p = 1$ et $k_v = 1,3$ de manière à obtenir un facteur d'amortissement $\zeta = 0,65$ conduisant à une réponse légèrement sous-amortie comme compromis approprié entre un temps de montée rapide et une réponse légèrement oscillatoire. Le gain ϵ_κ peut être ajusté de manière pratique afin d'obtenir un compromis entre l'action de lissage (obtenue en augmentant ϵ_κ) et la réduction du retard temporel de l'estimateur (obtenue en réduisant ϵ_κ). En outre, l'action de lissage de l'approche proposée atténue l'effet de la quanti-

fication de la mesure angulaire. Nous avons choisi $\epsilon_\kappa = 0,05$ pour nos expériences. La Figure 7.9 montre les résultats expérimentaux obtenus après la mise en œuvre du filtre haut gain (7.18) dans le cas d'un vol générant des oscillations angulaires de grande amplitude.

Nous avons effectué une dérivation par différence finie (en vert) en post-traitement pour comparer les résultats. En raison de la nature quantifiée de l'encodeur rotatif, nous observons que la vitesse angulaire obtenue par différence finie est très bruitée. Nous constatons que le filtre haut gain permet d'estimer la vitesse angulaire avec plus de précisions (en rouge), bien qu'avec un léger retard. Grâce à l'ajout d'une IMU supplémentaire sur l'aile lors d'un essai en vol, il est possible de comparer l'estimation de la vitesse avec les mesures du gyroscope de l'aile (MPU9250), visibles sur le graphique du bas de la Figure 7.9 (trace bleue). Nous constatons que les mesures du gyroscope sont quelque peu bruitées, notamment en raison des vibrations générées par les moteurs.

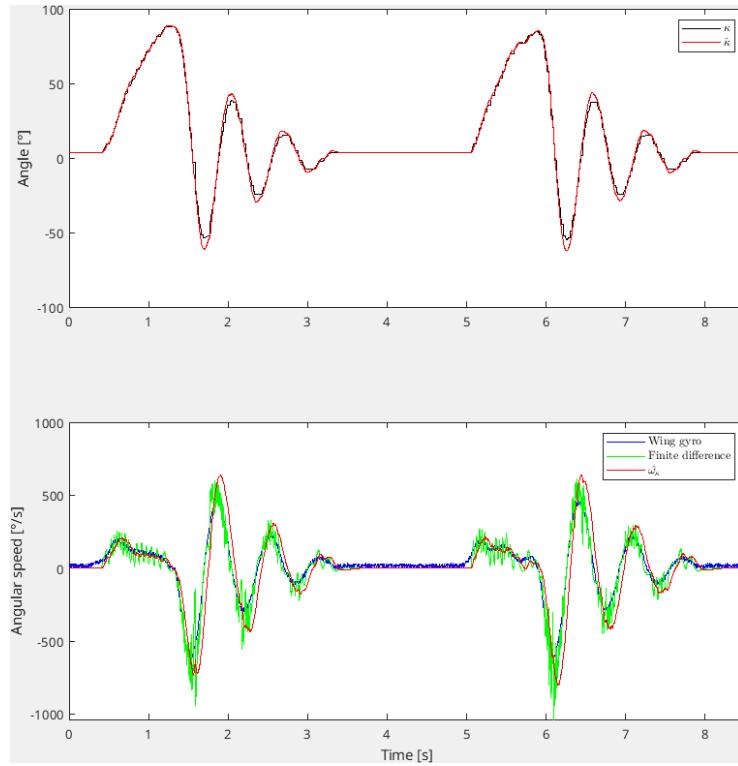


FIGURE 7.9 – Mesure de l'angle (noir, graphique du haut), mesure de la vitesse angulaire avec le gyroscope de l'aile (bleu, graphique du bas), estimation de la vitesse angulaire par différence finie (vert, graphique du bas) et estimation avec le filtre haut gain (courbes rouges).

Afin d'effectuer la transformation nécessaire entre les repères, nous définissons le quaternion $\mathbf{q}_{\hat{\kappa}} \in \mathbb{S}^3$ suivant :

$$\mathbf{q}_{\hat{\kappa}} = \left[\cos\left(\frac{\hat{\kappa}}{2}\right) \ 0 \ \sin\left(\frac{\hat{\kappa}}{2}\right) \ 0 \right]^\top \quad (7.19)$$

7.4.3 Estimation de l'état de l'aile

Sur la base de l'angle estimé $\hat{\kappa}$ et de la vitesse angulaire estimée $\hat{\omega}_\kappa$, il est possible de transformer les mesures réalisées dans le repère du fuselage vers le repère de l'aile. Tous les capteurs sont installés sur la carte de l'autopilote, qui est elle-même fixée au fuselage. Cependant, nous voulons utiliser INDI pour stabiliser l'aile. Cette loi de contrôle nécessite donc des informations sur l'état du drone dans le repère de l'aile, où toutes les forces sont appliquées (aérodynamiques et de traction).

Deux solutions viables sont alors possibles : effectuer l'estimation d'état dans le repère du fuselage et faire pivoter l'estimation en utilisant l'estimation de l'angle $\hat{\kappa}$, ou faire pivoter les mesures brutes à l'avance pour les exprimer dans le repère de l'aile, et ensuite effectuer l'estimation d'état dans ce dernier.

Compte tenu de l'architecture actuelle du logiciel du système Paparazzi (voir Annexe), il est impossible d'avoir deux structures conjointes d'estimation de l'état. Il est donc complexe de mettre en œuvre la première solution, dans laquelle le contrôleur récupère directement l'estimation de l'état actuel. C'est pourquoi nous avons choisi d'estimer l'état de l'aile à partir des données mesurées sur le fuselage. À cette fin, nous détaillons ci-dessous les transformations pour les trois capteurs : gyroscope, accéléromètre et magnétomètre.

Pour les mesures de la vitesse angulaire basées sur le gyroscope, nous pouvons calculer la vitesse angulaire de l'aile exprimée dans le repère de l'aile :

$$\boldsymbol{\omega}_W = \mathbf{R}(\mathbf{q}_{\hat{\kappa}}) \left(\boldsymbol{\omega}_{gyro}^F + \begin{bmatrix} 0 \\ \hat{\omega}_\kappa \\ 0 \end{bmatrix} \right) \quad (7.20)$$

où $\boldsymbol{\omega}_{gyro}^F$ est la vitesse angulaire mesurée par le gyroscope sur le fuselage, exprimée dans le repère du fuselage, $\hat{\omega}_\kappa$ est la vitesse angulaire estimée de l'aile par rapport au fuselage, conformément à (7.18), et $\mathbf{q}_{\hat{\kappa}}$ est le quaternion défini dans (7.19). L'expression (7.20) est similaire à une composition de vitesse angulaire et à un changement du repère.

Pour la mesure de l'accélération avec l'accéléromètre, nous pouvons utiliser la relation de l'équation (7.21), qui est obtenue à partir du théorème de transport du

taux de variation [Brizard 2004], où nous trouvons le terme d'accélération d'Euler $\dot{\boldsymbol{\omega}}_F \times \mathbf{d}_{AF}$ et le terme d'accélération centripète $\boldsymbol{\omega}_F \times (\boldsymbol{\omega}_F \times \mathbf{d}_{AF})$. L'accélération de Coriolis $2\boldsymbol{\omega}_F \times \frac{d(\mathbf{d}_{FW})}{dt} \Big|_{O_F}$ et le taux d'accélération $\frac{d^2(\mathbf{d}_{FW})}{d^2t} \Big|_{O_F}$ sont nuls car \mathbf{d}_{FW} est constant.

$$\mathbf{a}_W = \mathbf{R}(\mathbf{q}_{\hat{\kappa}}) \left(\mathbf{a}_{acc}^F + \dot{\boldsymbol{\omega}}_{gyro}^F \times \mathbf{d}_{FW} + \boldsymbol{\omega}_{gyro}^F \times (\boldsymbol{\omega}_{gyro}^F \times \mathbf{d}_{FW}) \right) \quad (7.21)$$

où $\mathbf{a}_{acc}^F \in \mathbb{R}^3$ est l'accélération mesurée par l'accéléromètre sur le fuselage, exprimée dans le repère du fuselage et $\boldsymbol{\omega}_{gyro}^F \in \mathbb{R}^3$ est la vitesse angulaire du fuselage, identique à l'équation (7.20). L'accélération angulaire $\dot{\boldsymbol{\omega}}_{gyro}^F \in \mathbb{R}^3$ dans (7.21) est calculée par différence finie.

Pour les mesures du magnétomètre, nous avons :

$$\mathbf{E}_W = \mathbf{R}(\mathbf{q}_{\hat{\kappa}}) \mathbf{E}_{mag} \quad (7.22)$$

où $\mathbf{E}_{mag} \in \mathbb{R}^3$ est la sortie du magnétomètre, exprimée dans le repère du fuselage et $\mathbf{E}_W \in \mathbb{R}^3$ est la mesure exprimée dans le repère de l'aile.

Pour obtenir une estimation de l'état de l'aile, nous utilisons un algorithme de fusion des mesures des capteurs : le filtre de Kalman étendu (EKF) qui fournit une estimation des états suivants : \mathbf{p}_W , \mathbf{v}_W , \mathbf{q}_W à partir de mesures transformées dans le repère de l'aile $\boldsymbol{\omega}_W$ (équation (7.20)), \mathbf{a}_W (équation (7.21)), \mathbf{E}_W (équation (7.22)) et les données du système de vision externe, qui fournissent une mesure précise de la position \mathbf{p}_W et de la vitesse \mathbf{v}_W du drone dans le repère inertiel (I).

7.4.4 Estimation de l'orientation du fuselage

Pour déterminer l'orientation du fuselage, nous pouvons effectuer une composition entre le quaternion représentant l'orientation de l'aile \mathbf{q}_W résultat de l'EKF et le quaternion construit à partir de la mesure filtrée de l'encodeur rotatif $\mathbf{q}_{\hat{\kappa}}$ dans (7.19),

$$\mathbf{q}_F = \mathbf{q}_W \otimes \mathbf{q}_{\hat{\kappa}} \quad (7.23)$$

où l'opérateur \otimes désigne le produit hamiltonien. La connaissance de \mathbf{q}_F est nécessaire pour maintenir le fuselage parfaitement horizontal.

7.5 Inversion non-linéaire incrémentale de la dynamique du drone

La théorie de l'inversion dynamique non-linéaire incrémentale (INDI) utilisée dans le contexte des micro-drones est présentée dans [Smeur 2016]. Nous utilisons la notation proposée dans [Smeur 2020]. L'hypothèse centrale sous-jacente est que le principe de séparation des échelles de temps s'applique à la dynamique de l'actionneur et à la dynamique des forces et des moments aérodynamiques. Le signal de commande \boldsymbol{u}_W définit dans (7.1) peut alors être calculé de manière incrémentale en utilisant la matrice d'efficacité de l'actionneur \boldsymbol{G} .

$$\boldsymbol{u}_W = \boldsymbol{u}_W + \boldsymbol{G}^\dagger (\boldsymbol{\nu} - \begin{bmatrix} \dot{\boldsymbol{\omega}}_W \\ T_W \end{bmatrix}) \quad (7.24)$$

où $\dot{\boldsymbol{\omega}}_W \in \mathbb{R}^3$ est l'accélération angulaire obtenue par différence finie à partir de l'équation (7.20), $T_W \in \mathbb{R}$ est la poussée actuelle.

$$\boldsymbol{\nu} = \begin{bmatrix} K_\Omega(\Omega_{ref} - \boldsymbol{\omega}_W) \\ T_d \end{bmatrix} \quad (7.25)$$

La matrice d'efficacité du contrôle \boldsymbol{G} est définie par :

$$\begin{bmatrix} \partial\phi \\ \partial\theta \\ \partial\psi \\ \partial T \end{bmatrix} = \boldsymbol{G}\boldsymbol{u}_W = \begin{bmatrix} -7.5 & -15 & 7.5 & 15 & 0 & 0 \\ 0 & 0 & 0 & 0 & 15 & 15 \\ 0 & 0 & 0 & 0 & 4 & -4 \\ -0.6 & -0.6 & -0.6 & -0.6 & 0 & 0 \end{bmatrix} \boldsymbol{u}_W$$

Cette sélection de la matrice d'efficacité a été déterminée pour les vols stationnaires, mais il est nécessaire d'effectuer une étude différente pour le vol vers l'avant.

7.6 Stabilisation du fuselage

Pour stabiliser le fuselage, nous utilisons un bouclage proportionnel-dérivé de l'angle θ_F formé entre le fuselage et l'horizontale, que nous voulons maintenir à zéro. Cet angle est obtenu en convertissant le quaternion q_F de l'équation (7.23) en un angle d'Euler en suivant la convention d'Euler 'ZYX'. Le bouclage fournit la commande u_{tail} pour la vitesse angulaire du moteur générant la force F_m (voir

Figure 7.7), suivant l'expression :

$$u_{\text{tail}} = u_{\text{tail},eq} + k_p \theta_F + k_d \dot{\theta}_F, \quad (7.26)$$

où u_{eq} est la commande, à l'équilibre, du moteur pour maintenir le fuselage horizontal en l'absence de perturbation et k_p , k_d sont des gains scalaires ajustables. Nous obtenons $\dot{\theta}_F$ à partir de $\boldsymbol{\omega}_{gyro}^F = [\dot{\phi}_F \ \dot{\theta}_F \ \dot{\psi}_F]^\top$.

La valeur $u_{\text{tail},eq}$ a été obtenue en utilisant l'équation (7.13). Nous annulons cette dernière $\sum \vec{M}_{O_W} = \vec{0}$ et nous obtenons :

$$m_F g e_3 \times \mathbf{d}_{GO_W} + \mathbf{F}_m \times \mathbf{d}_{MO_W} = \vec{0} \quad (7.27)$$

En utilisant (7.27) et $\mathbf{F}_{m,eq} = -k_p u_{\text{tail},eq}^2$, nous obtenons :

$$u_{\text{tail},eq} = \sqrt{\left(\frac{mgz_{GO_W}}{k_p z_{MO_W}}\right)} \quad (7.28)$$

avec z_{GO_W} et z_{MO_W} , respectivement, la troisième composante de \mathbf{d}_{GO_W} et \mathbf{d}_{MO_W} .

Les gains k_p et k_d ont été ajustés lors d'un montage sur un banc de test pour assurer un comportement satisfaisant.

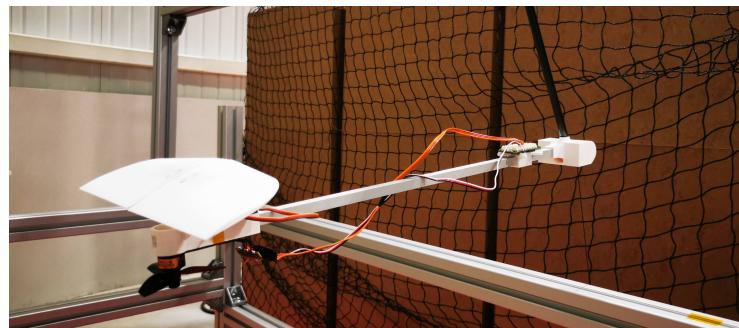


FIGURE 7.10 – Maquette expérimentale du fuselage de Colibri.

Le banc de test de la Figure (7.10) est un montage du fuselage sur un pivot fixe devant la soufflerie ouverte. Ce montage permet d'apprécier le fonctionnement de la commande (7.26) et son comportement face au vent.

7.7 Expérimentations

Un prototype a été mis au point, comme le montre la Figure 7.11. La Figure 7.12 présente une sélection des résultats expérimentaux en environnement contrôlé.



FIGURE 7.11 – Prototype : Colibri.

Sur la Figure 7.12, de 0 s à 8 s, le drone est au sol. De 8 s à 16 s, le drone décolle pour atteindre une hauteur de 2 mètres, visible sur le troisième graphique. Cette hauteur est atteinte après un dépassement de 10 %. Le drone est maintenu dans cette position pendant 54 s. Des oscillations d'incidence sont observées dans le cinquième et le dernier graphique, générant des oscillations dans la position horizontale du drone. Ce phénomène est dû au couplage entre les deux corps, qui n'est pas correctement stabilisé. À partir de 70 s, le drone commence à se diriger vers le point $\mathbf{p}_c = [3 \ 0.9 \ -1.5]^\top$ et $\psi_c = 90^\circ$.

7.8 Vol avec un contrôleur INDI unifié

Au vu du vol expérimental de la Figure 7.12, nous avons supprimé le contrôleur proportionnel-dérivé du fuselage pour utiliser l'INDI. Nous ajoutons le moteur du fuselage comme actionneur dans le vecteur de commande \mathbf{u}_W et nous ajoutons un nouvel objectif de stabilisation, lequel correspond à la stabilisation du fuselage u_{tail} .

Ainsi, la matrice \mathbf{G} définie dans (7.24) devient :

$$\begin{bmatrix} \partial\phi \\ \partial\theta \\ \partial\psi \\ \partial T \\ \partial\theta_F \end{bmatrix} = \mathbf{G}\mathbf{u}_W = \begin{bmatrix} -7.5 & -15 & 7.5 & 15 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 15 & 15 & 0 \\ 0 & 0 & 0 & 0 & 4 & -4 & 0 \\ -0.6 & -0.6 & -0.6 & -0.6 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 15 \end{bmatrix} \mathbf{u}_W.$$

Dans le cas dynamique, le mouvement de l'aile (et donc du point d'attache) a un impact sur le fuselage. Il est donc nécessaire de réaliser une étude dynamique (7.29). Toutefois, en l'absence de vent, l'actionneur utilisé pour stabiliser le fuselage est le moteur qui ne peut générer qu'une force vers le bas (comme développé dans la section 7.6).

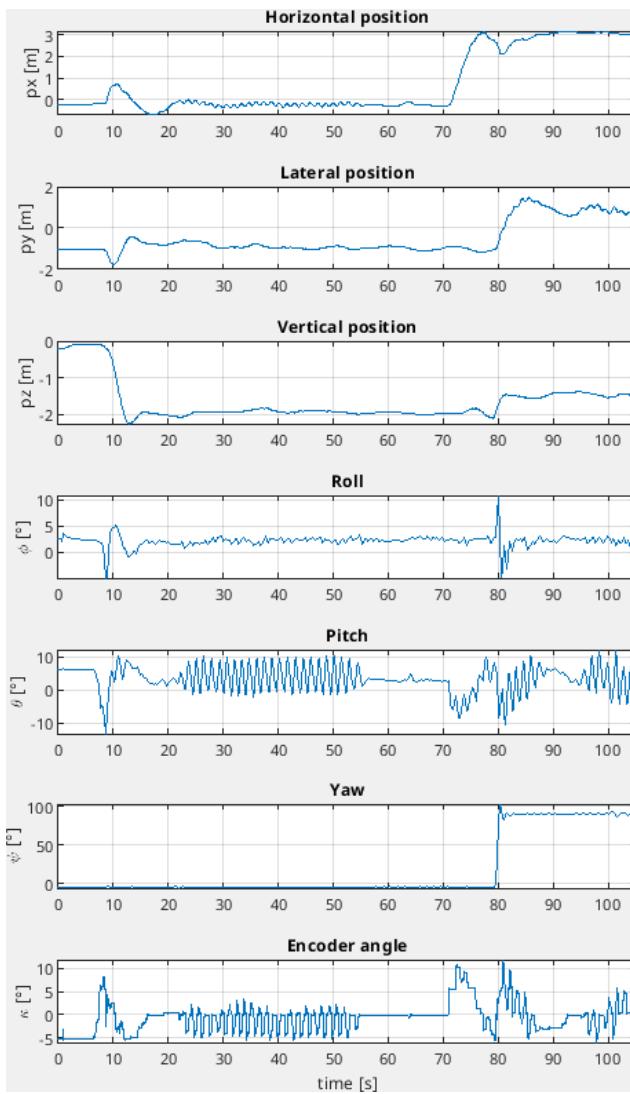


FIGURE 7.12 – Position et orientation de l'aile (six premiers graphiques) et mesure de l'angle entre l'aile et le fuselage (dernier graphique), lors d'un vol réel.

Remarque 1. En raison de l'irréversibilité de la génération de forces par le moteur du fuselage, une accélération positive ne pourra pas être contrebalancée par le moteur. Lors d'une décélération, une augmentation de la vitesse de rotation du moteur permet de compenser les accélérations linéaires et donc de maintenir le fuselage horizontal. ○

Il est possible d'anticiper la commande de vitesse angulaire du fuselage en fonction des accélérations linéaires verticale \ddot{z} et horizontale \ddot{x} de l'aile.

$$ml^2\ddot{\theta}_p + ml(g - \ddot{z}) \sin \theta_p + ml\ddot{x} \cos \theta_p = 0 \quad (7.29)$$

$$\implies \ddot{\theta}_p = -\frac{1}{l} ((g - \ddot{z}) \sin \theta_p + \ddot{x} \cos \theta_p) \quad (7.30)$$

avec $l = \|\mathbf{d}_{GO_W}\|$ et $\theta_p = \theta_F - \frac{\pi}{2} - \arctan \frac{x_{GO_W}}{z_{GO_W}}$. En régime stationnaire, nous avons un équilibre qui assure que l'accélération de la gravité g soit compensée par la traction de l'hélice \mathbf{F}_m . Nous discrétisons cette équation pour obtenir la commande *feedforward* de vitesse angulaire :

$$\dot{\theta}_p(t+1) = -\frac{1}{l} ((-\ddot{z}) \sin \theta_p + \ddot{x} \cos \theta_p) + dot\theta_p(t). \quad (7.31)$$

De cette équation (7.31), nous observons qu'il est possible de générer une trajectoire qui permette d'accélérer horizontalement sans générer un changement de vitesse angulaire du fuselage (qui ne pourrait pas être compensée voir remarque 1). Cette trajectoire utilise une accélération verticale \ddot{z} pour assurer $(-\ddot{z}) \sin \theta_p + \ddot{x} \cos \theta_p = 0$. Cette trajectoire est donc ascendante. Une autre solution à la remarque 1 est de permettre la génération d'une force bidirectionnelle (en ajoutant un second moteur ou un servomoteur pour diriger la force).

7.9 Conclusion

Ce chapitre propose une nouvelle architecture de drone *frewing* qui possède des caractéristiques intéressantes, tel qu'un fuselage maintenu horizontal pendant toutes les phases de vol, un décollage et un atterrissage vertical, et une passivité naturelle aux perturbations de vent. Toutefois, de nombreuses complexités apparaissent avec cette architecture. La modélisation multicorps menée a permis d'obtenir un modèle représentant le comportement du drone et une identification a isolé les coefficients pour que le modèle soit le plus réaliste. Un travail a été mené sur l'estimation de l'état d'un tel drone, permettant de conclure sur la meilleure position pour l'installation des capteurs et l'implémentation de la démarche nécessaire pour réaliser l'estimation de l'état.

À l'aide de cette dernière, une stabilisation basée sur une double boucle INDI et proportionnelle dérivée a permis de réaliser une validation expérimentale de la maquette, de l'estimation de l'état. Toutefois, cette implémentation a montré les limites d'une architecture à deux boucles. Nous avons donc proposé une architecture

de commande unifiée basée sur une rétroaction INDI, contrôlant les deux corps et permettant le contrôle expérimental d'un drone à fuselage pendulaire non actionné.

Conclusion

Cette thèse est consacrée à la modélisation, à l'étude et au contrôle d'un drone convertible sujet à des forces aérodynamiques, des couplages entre les actionneurs et des dynamiques non-linéaires. Elle propose, au travers de l'utilisation d'un modèle unifié représentant les forces aérodynamiques sur l'ensemble du domaine de vol, d'analyser le comportement d'un *tailsitter* et de proposer des méthodes de commande. Notre travail a débouché sur la proposition d'une nouvelle architecture de type *freewing*.

La première partie du manuscrit propose un rapide aperçu des architectures de drones convertibles avant de se focaliser sur les *tailsitters* et les *freewings*. Nous avons développé les principales caractéristiques d'actionnement, le comportement des drones ainsi que les méthodes de modélisation. Ce chapitre introductif a permis de développer l'architecture conventionnelle de commande, ainsi qu'un tour d'horizon des méthodes de commande et d'optimisation des contrôleurs. Notre travail ayant une forte composante expérimentale, une vision de l'architecture nécessaire à ces expérimentations a été évoquée avant d'être développée plus en profondeur dans l'Annexe.

Nous avons utilisé un modèle de la littérature, sans singularité sur l'ensemble du domaine de vol, ne faisant pas appel aux angles aérodynamiques α et β , appelé ϕ -théorie. Ce modèle mathématique n'a d'utilité pratique que s'il est cohérent avec la réalité, ce qui a pu être démontré dans la littérature et qui est confirmé par nos travaux. De ce modèle non-linéaire, nous avons pu extraire des caractéristiques intéressantes pour les drones à décollage et atterrissage vertical. Nous avons caractérisé l'ensemble des points d'équilibre avec ou sans vent pour un *tailsitter*. De ces équilibres, nous avons extrait la dynamique linéarisée, point de départ de la conception de toute loi de commande linéaire. Notre compréhension du comportement du drone a été augmentée par ces résultats qui nous informent sur la commandabilité du drone, sur les marges vis-à-vis des saturations et sur la capture du comportement du drone par une linéarisation autour des points d'équilibre. Nous avons donc pu valider la précision des linéarisations face aux nombreuses non-linéarités du modèle. Pour cela, nous avons effectué des simulations en boucle fermée, au vu du comportement instable du drone, du modèle linéaire et non-linéaire.

Un travail préliminaire a permis de proposer une architecture de commande hybride avec un mécanisme d'hystérésis, basée sur une variable discrète sélectionnant la loi de commande la plus appropriée en fonction de la phase de vol. Les

deux lois proposées dans ce cas sont une loi non-linéaire basée sur une direction de zéro-moment et une loi linéaire LQR. Cette loi LQR est optimisée grâce au modèle obtenu précédemment.

De ce travail et à l'aide de la linéarisation, nous avons observé un comportement intéressant pour le rejet de perturbations de vent sur un *tailsitter*. Ce comportement repose sur le changement de l'angle de tangage du drone pour compenser l'augmentation de la vitesse air qui engendre un déplacement du drone. Nous avons donc expérimenté, à l'aide d'une maquette à trois degrés de liberté, une loi de commande proportionnelle intégrale. Cette maquette, utilisant une architecture physique associée à un modèle de dynamique transactionnelle simulée, a permis de valider l'architecture de commande ainsi que son optimisation basée sur des contraintes H_∞ . Bien que les résultats obtenus soient prometteurs, nous avons étudié une méthode différente d'obtention des gains du contrôleur PI. Cette méthode, plus conservative, est basée sur une résolution successive de LMI. Les résultats ont pu être évalués sur l'architecture complète du drone, par un vol expérimental en volière.

Cette expérimentation a permis d'identifier des problèmes de sensibilité de la boucle fermée aux dynamiques non modélisées et aux bruits. Nous avons donc proposé une extension du contrôleur PI pour augmenter sa robustesse. Une expérimentation face à un vent croissant par palier a validé notre travail.

Nos travaux nous ont amené à vouloir installer un capteur de vent sur le drone pour pouvoir utiliser la mesure pour la transition. Toutefois, le corps du *tailsitter* étant en rotation lors de la transition, nous ne pouvions pas fixer le capteur de manière satisfaisante. Nous avons donc étudié et développé une architecture *freewing* procurant un fuselage maintenu horizontal permettant d'installer n'importe quel capteur ou charge utile. L'aile étant en rotation libre autour du fuselage, nous conservons de nombreuses propriétés des *tailsitters*. Dans cette démarche, nous avons modélisé le drone avec une dynamique multicorps, identifié les paramètres, fabriqué la maquette et réalisé des vols expérimentaux à l'aide de l'INDI.

Limite de l'étude

Les travaux préliminaires, menés au chapitre 3, ne sont que des résultats de simulation. Il serait souhaitable de réaliser des expérimentations du contrôleur non-linéaire basé sur une direction de zéro-moment ainsi que de son utilisation dans l'architecture hybride avec une transition entre le contrôleur basé sur une direction de zéro-moment et le contrôleur PI étendu développé au chapitre 6.

Bien que nous souhaitions utiliser la mesure du vent, notre travail n'a pu aboutir,

étant donné la richesse des questions que nous avons souhaité développer en amont et par le temps nécessaire au développement de l'architecture nécessaire. De plus, de nombreuses architectures auraient pu répondre au besoin. Nous avons choisi de nous concentrer sur une architecture inspirée du *tailsitter* DarkO car nous avions de l'expérience dans sa modélisation et sa dynamique.

Tous nos résultats ont été expérimentés dans une atmosphère contrôlée avec un générateur de perturbations. Il serait maintenant intéressant d'évaluer la précision de nos contrôleurs en extérieur. Ce travail possède une double complexité car le drone évoluerait dans un environnement plus turbulent, mais aussi avec une estimation d'état moins précise. Effectivement, en intérieur, nous avons accès à un système de positionnement millimétrique alors qu'en extérieur, les GPS ne peuvent nous fournir une information de position aussi précise.

Travaux futurs

La modélisation de Udwadia-Kalaba permet d'obtenir un modèle d'un drone multicorps. Il serait intéressant d'utiliser ces travaux pour concevoir un contrôleur assurant la stabilisation de l'aile et du fuselage en prenant en compte leurs interactions.

De plus, ce contrôleur pourrait profiter de la mesure du vent réalisée par une sonde cinq trous installée sur le fuselage. Comme le fuselage est maintenu horizontal et avec une faible variation d'incidence, la mesure sera disponible dans toutes les phases de vol. Il est nécessaire de caractériser la sonde. Effectivement, la mesure étant réalisée par une différence de pression statique et dynamique, nous devons donc étudier la sensibilité des capteurs pour déterminer la plus petite variation de vent mesurable ainsi que le temps nécessaire à la mesure.

Enfin, demeure un sujet de recherche, non abordé à ce jour : si les drones convertibles possèdent 2 modes de déplacement (stationnaire et vol d'avancement), il conviendra de s'interroger sur la stratégie de vol à adopter, en fonction des caractéristiques de la mission.

En effet, en fonction de la distance entre les points de l'espace, il s'agit de choisir entre :

- effectuer une transition du mode stationnaire vers le mode d'avancement (efficace énergétiquement)
- ou rester en mode stationnaire et se déplacer de proche en proche (très énergivore).

Si le mode d'avancement est énergétiquement intéressant, il implique deux tran-

sitions et la réalisation d'un cercle (nécessaire pour de courtes distances et pour avoir le temps de réaliser les transitions).

Ce problème est schématisé sur la Figure C.1.

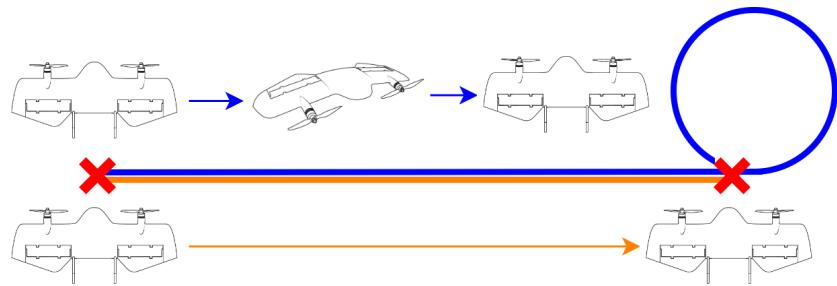


FIGURE C.1 – Schéma de déplacement d'un drone convertible.

Annexe technique sur les drones

Système de drone : Paparazzi

Un drone est composé de plusieurs pièces assemblées entre elles pour former la structure sur laquelle sont fixés des actionneurs, un autopilote et une charge utile (colis, caméra, capteur, etc.). L'élément central est l'autopilote qui assure la communication entre tous les éléments. Nous pouvons décomposer l'autopilote en deux parties : la partie matérielle et la partie logicielle. La partie matérielle est constituée d'un circuit imprimé (PCB) sur lequel des composants sont installés pour assurer les tâches relatives au vol. La partie logicielle se décompose en deux éléments : le segment sol et le logiciel embarqué.

Nous pouvons détailler les capteurs embarqués et le microcontrôleur avec l'ensemble de ses ports de communication.

Les capteurs d'un autopilote

Un autopilote comporte généralement un accéléromètre, un gyroscope, un magnétomètre et un baromètre.

L'accéléromètre à trois axes permet de mesurer l'ensemble des forces appliquées sur le véhicule, à l'exception du poids. Il est possible d'obtenir la position du drone par double intégration de la mesure de l'accéléromètre. Toutefois, il convient de souligner que la position dérive rapidement en raison des bruits de mesure.

Le gyroscope à trois axes permet de mesurer les vitesses de rotation du véhicule. Il est possible d'obtenir l'orientation du drone par intégration de la mesure du gyroscope. Toutefois, comme précédemment, l'orientation dérive rapidement en raison des bruits de mesure.

Le magnétomètre à trois axes indique la direction du nord magnétique. Il permet de se diriger par rapport à une référence connue. Le principal inconvénient de ce capteur est sa perturbation par les masses magnétiques environnantes, ainsi que par les champs magnétiques parasites induits par la proximité des moteurs électriques par exemple. Il est donc difficile de les utiliser à l'intérieur d'un bâtiment. L'influence magnétique de l'engin porteur et les perturbations dues à d'éventuels moteurs électriques peuvent être éliminées en qualifiant, de manière statique, les

erreurs dues aux masses métalliques du véhicule et aux moteurs électriques (en fonction des tensions et courants d'alimentation).

Le baromètre est un capteur d'altitude basé sur la mesure de la pression atmosphérique. Cette pression est mesurée par un système électronique basé sur la résonance naturelle d'une pièce en alliage de nickel ou sur la modification de l'équilibre d'un pont de Wheatstone associé à un cristal de quartz sur lequel, par l'intermédiaire d'une capsule souple, s'exerce la pression atmosphérique. On déduit de la variation de pression atmosphérique une variation d'altitude à l'aide du modèle d'atmosphère standard qui nous indique qu'au niveau de la mer, la pression diminue de 1 hPa tous les 8.5 m.

Le GPS est monté en extérieur de l'autopilote. Ce système de géopositionnement par satellite (*Global Positioning System*) permet d'obtenir un positionnement absolu du drone.

Il est courant de retrouver plusieurs capteurs dans un même boitier, que l'on nomme centrale inertie (Inertial Measurement Units, IMU). Ces dernières sont composées au minimum d'un accéléromètre 3-axes et d'un gyroscope 3-axes, mais il est courant de les trouver avec un magnétomètre 3-axes.

Le microcontrôleur d'un autopilote

Le microcontrôleur (Microcontroller Unit, MCU) est la pièce maîtresse de l'autopilote en ce qu'il permet d'effectuer l'ensemble des traitements nécessaires à la conduite du vol.

De plus, il possède plusieurs ports de communication pour récupérer les données des capteurs ou envoyer des ordres aux actionneurs.

La liaison série permet de relier deux équipements numériques pour qu'ils puissent s'échanger des informations. C'est le moyen de communication le plus simple. Toutefois, il contient un moyen de détection des erreurs tel que le bit de parité.

Le protocole *CAN* provient de l'industrie automobile. Il permet de raccorder à un même câble un grand nombre de calculateurs qui communiqueront donc à tour de rôle. Cette technique élimine le besoin de câbler des lignes dédiées pour chaque information à faire transiter (connexion point-à-point).

Nous pouvons citer le *Dshot* qui est un protocole de communication défini entre l'autopilote et l'ESC pour envoyer les commandes des moteurs. Les avancées sur

ce protocole ont notamment permis la communication bidirectionnelle, permettant d'obtenir la vitesse des moteurs, leur consommation et d'autres informations.

Enfin, le protocole *I²C* est un bus de communication série simplifiant l'interconnexion de circuit intégré sur une même carte. Ce bus ne nécessite que deux fils pour être mis en place. Il n'est conçu que pour faire communiquer des équipements relativement proches (quelques centimètres).

Évolutions

Les nombreux progrès dans les systèmes d'estimation état permettent de connaître précisément l'orientation et la position des drones pour assurer la stabilisation, le guidage et la navigation. Les progrès sont liés à l'amélioration continue des capteurs, notamment des centrales inertielles constituées d'un accéléromètre, d'un gyroscope et d'un magnétomètre.

La Table A.1 montre l'évolution des vitesses des microcontrôleurs (Microcontroller Unit, MCU) embarqués sur les autopilotes et de la réduction du bruit des capteurs inertIELS.

Type	Date	MCU	Vitesse	Capteur	Bruit RMS
Apogee	2013	STM32F4	168 MHz	MPU-9150	Gyro : 0.06 dps Accel : 4 mg
Chimera	2016	STM32F7	216 MHz	MPU-9250	Gyro : 0.1 dps Accel : 8 mg
Tawaki 1	2019	STM32F7	216 MHz	ICM-20600	Gyro : 0.04 dps Accel : 1 mg
Tawaki 2	2023	STM32H7	480 MHz	ICM-42688-P	Gyro : 0.028 dps Accel : 0.70 mg

TABLE A.1 – Évolution des autopilotes Paparazzi sur dix ans.

Sur une période de dix ans, nous pouvons observer que les microcontrôleurs ont doublé leur vitesse d'exécution, que les fabricants ont divisé par deux le bruit moyen sur les gyroscopes et par quatre le bruit moyen des accéléromètres. Ces évolutions continues permettent une amélioration de l'estimation du drone utilisée pour la stabilisation. Il en résulte une stabilité accrue et de nouvelles possibilités pour la commande des drones.

Les logiciels d'un autopilote

Tout le fonctionnement d'un drone repose sur le logiciel qui permet de le faire voler. Il se décompose en deux catégories : la partie sol et la partie embarquée.

Le segment sol

Le segment sol est un ensemble de logiciels permettant de moniterer l'état du drone et de lui envoyer des ordres. Il repose sur les informations échangées avec le drone au travers de la télémétrie. L'interface principale est la GCS *Ground Control Station* (voir Figure A.1), laquelle assure la visualisation du drone sur la carte, ainsi que toutes les commandes nécessaires au vol (modification de point de passage, atterrissage, etc.). Elle est écrite en C++.

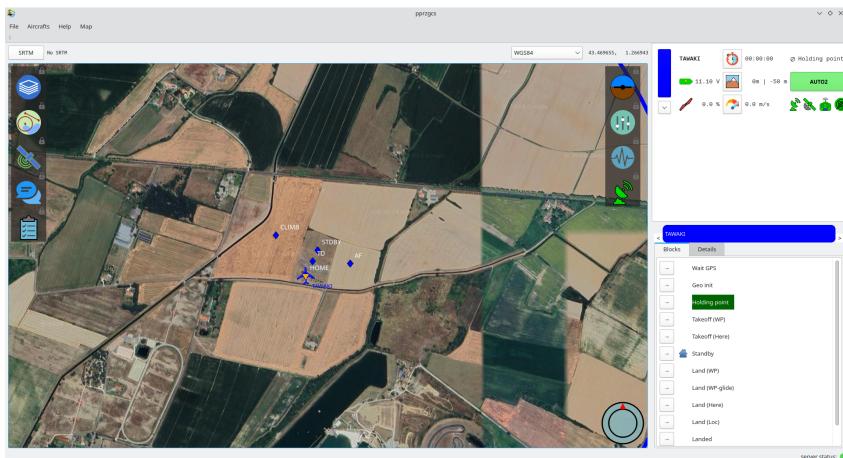


FIGURE A.1 – Interface graphique de la station de contrôle au sol.

Une autre partie du segment sol est le code serveur qui gère les messages échangés entre les différentes applications. Le code est en Ocaml.

Enfin, un code assure la compilation croisée du logiciel embarqué qui doit être téléchargé sur le drone, basée sur des Makefile et du code Ocaml. Le logiciel embarqué est décrit par la suite.

Le logiciel embarqué

Le logiciel embarqué est un code écrit en C, intégré dans un système d'exploitation temps réel "Chibios". Il est téléchargé sur le microcontrôleur au travers d'une sonde de programmation ou de la prise USB présente sur l'autopilote.

L'ensemble du code est organisé sous la forme de modules que l'on change au besoin. Chaque module assure des fonctionnalités telles que l'estimation d'état, la stabilisation, le guidage, la navigation ou encore la gestion de la charge utile (voir Figure A.2). Grâce à un mécanisme de gestion de dépendance, les modules ont la possibilité de charger d'autres modules nécessaires à leur fonctionnement. L'ordre de compilation et d'édition des liens seront gérés par le logiciel de compilation.

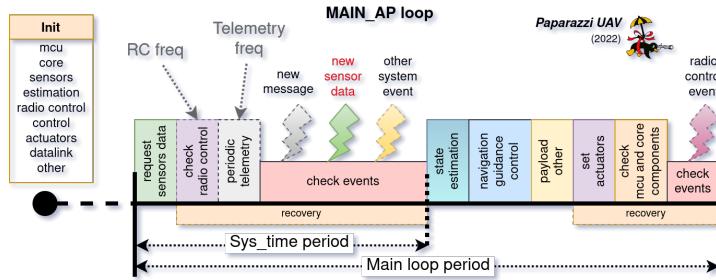


FIGURE A.2 – Schéma de l'ordre d'exécution des codes embarqués [Team 2022].

AM32

Le logiciel AM32 est conçu pour les microprocesseurs ARM STM32 afin de contrôler un moteur *brushless*, couramment utilisé pour les drones. Le logiciel est conçu pour être sûr et rapide, avec des démaragements rapides et sans à-coups et une accélération linéaire. Il est destiné à être utilisé avec plusieurs types de véhicules et de contrôleurs de vol.

L'intérêt de ce logiciel est qu'il est ouvert, permettant de contribuer, en proposant des évolutions. Nous avons ainsi implémenté l'approche de [Franchi 2017], avec un algorithme de biais et de gain adaptatif (ABAG) (voir A.3).

Algorithm 1: ABAG Propeller Speed Control Algorithm

Inputs	$y_k \in \mathbb{R}^+$	%Measured rotation period, [s]
	$y_k^d \in \mathbb{R}^+$	%Desired rotation period, [s]
Output	$u_k \in [0, 1]$	%PWM duty cycle, [adim.]
Parameters	$\alpha \in (0, 1)$	%error sign filtering factor, [adim.]
	$\bar{e}_b \in (0, 1)$	%threshold for bias adaptation, [s]
	$\delta_b \in (0, 1)$	%bias adaptation step, [adim.]
	$\bar{e}_g \in (0, 1)$	%threshold for gain adaptation, [s]
	$\delta_g \in (0, 1)$	%gain adaptation step, [adim.]
Variables	$\bar{e}_k \in [-1, 1]$	% low-pass filtered error sign, [adim.]
	$b_k \in [0, 1]$	% adaptive bias, [adim.]
	$g_k \in [0, 1]$	% adaptive gain, [adim.]

```

1  $k = 0, u_0 = \bar{e}_0 = g_0 = b_0 = 0$ 
2 while  $+k$  do
3    $\bar{e}_k = \alpha \bar{e}_{k-1} + (1 - \alpha) \text{sgn}(y_k - y_k^d)$            %error sign filtering
4    $b_k = \text{sat}_{[0,1]}(b_{k-1} + \delta_b \text{hside}(|\bar{e}_k| - \bar{e}_b) \text{sgn}(\bar{e}_k - \bar{e}_b))$  %bias update
5    $g_k = \text{sat}_{[0,1]}(g_{k-1} + \delta_g \text{sgn}(|\bar{e}_k| - \bar{e}_g))$            %gain update
6    $u_k = \text{sat}_{[0,1]}(b_k + g_k \text{sgn}(y_k - y_k^d))$            %control input computation

```

FIGURE A.3 – Algorithme de biais et de gain adaptatif (ABAG) [Franchi 2017].

L'algorithme ABAG est adaptatif et robuste en ce qu'il ne nécessite pas la connaissance des paramètres mécaniques ou électriques du groupe moteur et hélice et qu'il n'est pas nécessaire de procéder à une identification, ni de connaître l'entrée

nominale. De plus, l'algorithme ABAG ne nécessite que très peu de ressources de calcul, ce qui en fait un atout important pour un système embarqué.

Bibliographie

- [Andreetto 2016] Andreetto, M., Fontanelli, D. et Zaccarian, L. *Quasi time-optimal hybrid trajectory tracking of an n-dimensional saturated double integrator.* Dans IEEE Conference on Control Applications, pages 550–555, 2016. (Cité en page 63.)
- [Apkarian 2006] Apkarian, P. et Noll, D. *Nonsmooth H_∞ Synthesis.* IEEE Transactions on Automatic Control, vol. 51, no. 1, pages 71–86, 2006. [En ligne]. Disponible : <http://dx.doi.org/10.1109/TAC.2005.860290>. (Cité en pages 24, 77, 100 et 102.)
- [Apkarian 2013] Apkarian, P. *Tuning controllers against multiple design requirements.* Dans 2013 American Control Conference, pages 3888–3893, 2013. (Cité en pages 24 et 102.)
- [Aref 2018] Aref, P., Ghoreyshi, M., Jirasek, A., Satchell, M. J. et Bergeron, K. *Computational Study of Propeller-Wing Aerodynamic Interaction.* Aerospace, vol. 5, no. 3, 2018. [En ligne]. Disponible : <http://dx.doi.org/10.3390/aerospace5030079>. (Cité en page 3.)
- [Arzelier 2018] Arzelier, D., Dabbene, F., Formentin, S., Peaucelle, D. et Zaccarian, L. *Robust Static Output Feedback Design with Deterministic and Probabilistic Certificates.* Dans Uncertainty in Networked Systems : In Honor of Roberto Tempo, pages 121–148. Springer, December 2018. (Cité en pages 81, 83, 84 et 93.)
- [Avy 2023] Avy. 2023. [En ligne]. Disponible : <https://avy.eu/>. (Cité en pages vii et 9.)
- [Axten 2021] Axten, R. et Johnson, E. *Vtol freewing testbed vehicle in hover and forward flight.* Dans AIAA Scitech 2021 Forum, AIAA Scitech 2021 Forum, pages 1–19. American Institute of Aeronautics and Astronautics Inc, AIAA, 2021. Publisher Copyright : © 2021, American Institute of Aeronautics and Astronautics Inc, AIAA. All rights reserved. ; AIAA Science and Technology Forum and Exposition, AIAA SciTech Forum 2021 ; Conference date : 11-01-2021 Through 15-01-2021. (Cité en page 12.)
- [Axten 2023] Axten, R. M., Khamvilai, T. et Johnson, E. N. *VTOL Freewing Design and Adaptive Controller Development.* Dans AIAA SCITECH 2023 Forum, 2023. (Cité en page 12.)

- [Barth 2018] Barth, J. M. O., Condomines, J.-P., Bronz, M., Lustosa, L. R., Moeschetta, J.-M., Join, C. et Fliess, M. *Fixed-wing UAV with transitioning flight capabilities : Model-Based or Model-Free Control approach ? A preliminary study.* Dans 2018 International Conference on Unmanned Aircraft Systems (ICUAS), pages 1157–1164, 2018. (Cité en page 19.)
- [Binz 2019] Binz, F., Islam, T. et Moormann, D. *Attitude control of tiltwing aircraft using a wing-fixed coordinate system and incremental nonlinear dynamic inversion.* International Journal of Micro Air Vehicles, vol. 11, page 1756829319861370, 2019. [En ligne]. Disponible : <https://doi.org/10.177/1756829319861370>. (Cité en page 19.)
- [Brizard 2004] Brizard, A. J. *Motion in a non-inertial frame.* Saint Michael's College, Colchester, VT, 2004. (Cité en page 131.)
- [Cardoso 2018] Cardoso, D. N., Esteban, S. et Raffo, G. V. *Nonlinear \mathcal{H}_2 and \mathcal{H}_∞ control formulated in the Weighted Sobolev space for underactuated mechanical systems with input coupling.* Dans 2018 IEEE Conference on Decision and Control (CDC), pages 3812–3817, 2018. (Cité en page 24.)
- [Cardoso 2019] Cardoso, D. N., Esteban, S. et Raffo, G. V. *A Nonlinear \mathcal{W}_∞ Controller of a Tilt-rotor UAV for trajectory tracking.* Dans 2019 18th European Control Conference (ECC), pages 928–934, 2019. (Cité en page 24.)
- [Cardoso 2021] Cardoso, D. N., Esteban, S. et Raffo, G. V. *A robust optimal control approach in the weighted Sobolev space for underactuated mechanical systems.* Automatica, vol. 125, page 109474, 2021. (Cité en page 24.)
- [Cardoso 2024] Cardoso, D. N., Terra, M. H. et Raffo, G. V. *Robust Nonlinear \mathcal{W}_∞ Optimal Control for Input Nonaffine Systems.* Dans 2024 European Control Conference (ECC), pages 65–71, 2024. (Cité en page 24.)
- [Casau 2011] Casau, P., Cabecinhas, D. et Silvestre, C. *Autonomous Transition Flight for a Vertical Take-Off and Landing aircraft.* Dans 2011 50th IEEE Conference on Decision and Control and European Control Conference, pages 3974–3979, 2011. (Cité en page 20.)
- [Çetinsoy 2012] Çetinsoy, E., Dikyar, S., Hançer, C., Oner, K., Sirimoglu, E., Unel, M. et Aksit, M. *Design and construction of a novel quad tilt-wing UAV.* Mechatronics, vol. 22, no. 6, pages 723–745, 2012. (Cité en page 11.)
- [Chakrabarty 2016] Chakrabarty, A. K. et Bhattacharya, S. *Lyapunov Based Two-stage Robust Model Reference Adaptive Controller for Linear Plants with*

- Time Varying Bounded Uncertainties.* IFAC-PapersOnLine, vol. 49, no. 1, pages 213–218, 2016. [En ligne]. Disponible : <https://www.sciencedirect.com/science/article/pii/S2405896316300556>. 4th IFAC Conference on Advances in Control and Optimization of Dynamical Systems ACODS 2016. (Cité en page 21.)
- [Chiappinelli 2018] Chiappinelli, R. et Nahon, M. *Modeling and Control of a Tail-sitter UAV.* Dans 2018 International Conference on Unmanned Aircraft Systems (ICUAS), pages 400–409, 2018. (Cité en pages 3 et 10.)
- [Chitour 2002] Chitour, Y. *Time-varying high-gain observers for numerical differentiation.* IEEE Transactions on Automatic Control, vol. 47, no. 9, pages 1565–1569, 2002. [En ligne]. Disponible : <http://dx.doi.org/10.1109/TAC.2002.802740>. (Cité en page 128.)
- [Curpanaru 2024] Curpanaru, A.-I., Demourant, F. et Sansou, F. *Control of the DarkO Tail-Sitter Drone through an LMI-Based Static Output Feedback Design.* Dans Richardson, T., éditeur, 15th annual International Micro Air Vehicle Conference and Competition, pages 254–261, Bristol, United Kingdom, Sep 2024. Paper no. IMAV2024-30. (Cité en page 5.)
- [Dickeson 2005] Dickeson, J., Mix, D., Koenig, J., Linda, K., Cifdaloz, O., Wells, V. et Rodriguez, A. *H ∞ Hover-to-Cruise Conversion for a Tilt-Wing Rotorcraft.* Dans Proceedings of the 44th IEEE Conference on Decision and Control, pages 6486–6491, 2005. (Cité en page 24.)
- [Dickeson 2006] Dickeson, J. J., Cifdaloz, O., Miles, D. W., Koziol, P. M., Wells, V. L. et Rodriguez, A. A. *Robust H ∞ Gain-Scheduled Conversion for a Tilt-Wing Rotorcraft.* Dans Proceedings of the 45th IEEE Conference on Decision and Control, pages 5882–5887, 2006. (Cité en page 24.)
- [Dickeson 2007] Dickeson, J. J., Miles, D., Cifdaloz, O., Wells, V. L. et Rodriguez, A. A. *Robust LPV H ∞ gain-scheduled hover-to-cruise conversion for a tilt-wing rotorcraft in the presence of CG variations.* Dans 2007 46th IEEE Conference on Decision and Control, pages 2773–2778, 2007. (Cité en page 24.)
- [Droandi 2015] Droandi, G., Zanotti, A., Gibertini, G., Grassi, D. et Campanardi, G. *Experimental investigation of the rotor-wing aerodynamic interaction in a tiltwing aircraft in hover.* The Aeronautical Journal, vol. 119, no. 1215, page 591–612, 2015. [En ligne]. Disponible : <http://dx.doi.org/10.1017/S0001924000010708>. (Cité en page 3.)

- [Du 2024] Du, S. et Zha, Y. *Numerical simulation of the transition flight aerodynamics of cross-shaped quad-tiltrotor UAV*. Scientific Reports, vol. 14, no. 1, page 17878, 2024. (Cité en page 10.)
- [Ducard 2014] Ducard, G. et Hua, M.-D. *Modeling of an unmanned hybrid aerial vehicle*. Dans 2014 IEEE Conference on Control Applications (CCA), pages 1011–1016, 2014. (Cité en page 14.)
- [Ducard 2021] Ducard, G. J. J. et Allenspach, M. *Review of designs and flight control techniques of hybrid and convertible VTOL UAVs*. Aerospace Science and Technology, vol. 118, page 107035, November 2021. [En ligne]. Disponible : <http://dx.doi.org/10.1016/j.ast.2021.107035>. (Cité en pages vii, 8 et 21.)
- [Dufour 2024] Dufour. 2024. [En ligne]. Disponible : <https://www.dufour.aero/aero2>. (Cité en pages vii et 11.)
- [Ebihara 2015] Ebihara, Y., Peaucelle, D. et Arzelier, D. *S-Variable Approach to LMI-Based Robust Control*. Springer, January 2015. (Cité en page 83.)
- [Escareno 2006] Escareno, J., Salazar, S. et Lozano, R. *Modelling and Control of a Convertible VTOL Aircraft*. Dans Proceedings of the 45th IEEE Conference on Decision and Control, pages 69–74, 2006. (Cité en page 3.)
- [Escareno 2007] Escareno, J., Stone, R., Sanchez, A. et Lozano, R. *Modeling and control strategy for the transition of a convertible tail-sitter UAV*. Dans European Control Conference, pages 3385–3390, 2007. (Cité en page 3.)
- [Fernandez 2023a] Fernandez, L. F., Bronz, M., Bartoli, N. et Lefebvre, T. *Assessment of Methods for Propeller Performance Calculation at High Incidence Angles*. Dans AIAA SCITECH 2023 Forum, 2023. (Cité en page 3.)
- [Fernandez 2023b] Fernandez, L. F. T., Bronz, M., Bartoli, N. et Lefebvre, T. *Development of a Mission-Tailored Tail-Sitter MAV*. Unmanned systems, vol. 12, no. 03, pages 611–625, November 2023. [En ligne]. Disponible : <https://hal.science/hal-04612206>. (Cité en pages vii et 11.)
- [Fliess 2013] Fliess, M. et Join, C. *Model-free control*. International Journal of Control, vol. 86, no. 12, page 2228–2252, December 2013. [En ligne]. Disponible : <http://dx.doi.org/10.1080/00207179.2013.810345>. (Cité en page 20.)
- [Flores 2014] Flores, G., Lugo, I. et Lozano, R. *6-DOF hovering controller design of the Quad Tiltrotor aircraft : Simulations and experiments*. Dans 53rd IEEE Conference on Decision and Control, pages 6123–6128, 2014. (Cité en pages vii et 10.)

- [Franchi 2017] Franchi, A. et Mallet, A. *Adaptive Closed-loop Speed Control of BLDC Motors with Applications to Multi-rotor Aerial Vehicles*. Dans IEEE International Conference on Robotics and Automation, Singapour, Singapour, May 2017. (Cité en pages x, 25 et 147.)
- [Ge 1999] Ge, S., Hang, C. et Zhang, T. *A direct adaptive controller for dynamic systems with a class of nonlinear parameterizations*. Automatica, vol. 35, no. 4, pages 741–747, 1999. [En ligne]. Disponible : <https://www.sciencedirect.com/science/article/pii/S0005109898002155>. (Cité en page 21.)
- [Gillebaart 2014] Gillebaart, T., Bernhammer, L., Zuijlen, A. et Kuik, G. *Active flap control on an aeroelastic wind turbine airfoil in gust conditions using both a CFD and an engineering model*. Journal of Physics : Conference Series, vol. 524, page 012060, 06 2014. [En ligne]. Disponible : <http://dx.doi.org/10.1088/1742-6596/524/1/012060>. (Cité en page 107.)
- [Goebel 2012] Goebel, R., Sanfelice, R. G. et Teel, A. R. Hybrid dynamical systems : Modeling, stability, and robustness. Princeton University Press, New Jersey, 2012. (Cité en pages 63, 65 et 67.)
- [Guerrero 2009] Guerrero, J. A., Lozano, R., Romero, G., Lara-Alabazares, D. et Wong, K. C. *Robust control design based on sliding mode control for hover flight of a mini tail-sitter Unmanned Aerial Vehicle*. Dans 2009 35th Annual Conference of IEEE Industrial Electronics, pages 2342–2347, 2009. (Cité en page 3.)
- [Holsten 2011] Holsten, J., Ostermann, T. et Moormann, D. *Design and wind tunnel tests of a tiltwing UAV*. CEAS aeronautical journal, vol. 2, pages 69–79, 2011. (Cité en page 11.)
- [Hua 2013] Hua, M.-D., Hamel, T., Morin, P. et Samson, C. *Introduction to feedback control of underactuated VTOL vehicles : A review of basic control design ideas and principles*. IEEE Control Systems, vol. 33, pages 61–75, February 2013. [En ligne]. Disponible : <http://dx.doi.org/10.1109/MCS.2012.2225931>. (Cité en pages 32, 52 et 59.)
- [Jain 2013] Jain, P. et Nigam, M. J. *Design of a Model Reference Adaptive Controller Using Modified MIT Rule for a Second Order System 1*. Dans Advance in Electronic and Electric Engineering, 2013. (Cité en page 21.)
- [Jardin 2007] Jardin, M. et Mueller, E. *Optimized Measurements of UAV Mass Moment of Inertia with a Bifilar Pendulum*. Dans AIAA Guidance, Navigation and Control Conference and Exhibit, 2007. (Cité en page 37.)

- [Ke 2017a] Ke, Y. et Chen, B. M. *Full envelope dynamics modeling and simulation for tail-sitter hybrid UAVs.* Dans 2017 36th Chinese Control Conference (CCC), pages 2242–2247, 2017. (Cité en page 3.)
- [Ke 2017b] Ke, Y., Wang, K., Gong, K., Lai, S. et Chen, B. M. *Model based robust forward transition control for tail-sitter hybrid unmanned aerial vehicles.* Dans 2017 13th IEEE International Conference on Control and Automation (ICCA), pages 828–833, 2017. (Cité en page 2.)
- [LabJack 2001] LabJack. T7. 2001. [En ligne]. Disponible : <https://labjack.com/products/t7>. Accessed : 2022-09-19. (Cité en page 72.)
- [Li 2021] Li, F., Song, W.-P., Song, B.-F. et Zhang, H. *Dynamic modeling, simulation, and parameter study of electric quadrotor system of Quad-Plane UAV in wind disturbance environment.* International Journal of Micro Air Vehicles, vol. 13, page 17568293211022211, 2021. [En ligne]. Disponible : <https://doi.org/10.1177/17568293211022211>. (Cité en page 9.)
- [Liang 2016] Liang, J., Fei, Q., Wang, B. et Geng, Q. *Tailsitter VTOL flying wing aircraft attitude control.* Dans 2016 31st Youth Academic Annual Conference of Chinese Association of Automation (YAC), pages 439–443, 2016. (Cité en page 21.)
- [Lustosa 2015] Lustosa, L., Defaÿ, F. et Moschetta, J. *Longitudinal study of a tilt-body vehicle : modeling, control and stability analysis.* Dans Proc. of International Conference on Unmanned Aircraft Systems, pages 816–824, Denver, Colorado, US, June 2015. (Cité en page 2.)
- [Lustosa 2017] Lustosa, L. R. *La Phi-théorie : une approche pour la conception de lois de commande de vol des véhicules convertibles.* PhD thesis, Toulouse, ISAE, 2017. Thèse de doctorat dirigée par Moschetta, Jean-Marc et Defaÿ, François Automatique, Dynamique des fluides Toulouse, ISAE 2017. (Cité en pages 14 et 19.)
- [Lustosa 2019] Lustosa, L. R., Defaÿ, F. et Moschetta, J.-M. *Global Singularity-Free Aerodynamic Model for Algorithmic Flight Control of Tail Sitters.* Journal of Guidance, Control, and Dynamics, vol. 42, no. 2, pages 303–316, February 2019. [En ligne]. Disponible : <http://dx.doi.org/10.2514/1.G003374>. (Cité en pages 3, 5, 14, 31, 32, 33, 34, 122, 123 et 124.)
- [Mancinelli 2022] Mancinelli, A., Smeur, E. J., Remes, B. et Croon, G. d. *Dual-axis tilting rotor quad-plane design, simulation, flight and performance comparison with a conventional quad-plane design.* Dans 2022 International Confe-

- rence on Unmanned Aircraft Systems (ICUAS), pages 197–206, 2022. (Cité en page 10.)
- [Mathur 2021] Mathur, A. et Atkins, E. *Design, Modeling and Hybrid Control of a QuadPlane*. Dans AIAA Scitech 2021 Forum, 01 2021. (Cité en page 9.)
- [McFarlane 1992] McFarlane, D. et Glover, K. *A loop-shaping design procedure using H_{∞} / synthesis*. IEEE Transactions on Automatic Control, vol. 37, no. 6, pages 759–769, 1992. [En ligne]. Disponible : <http://dx.doi.org/10.1109/9.256330>. (Cité en page 82.)
- [Michieletto 2020] Michieletto, G., Cenedese, A., Zaccarian, L. et Franchi, A. *Hierarchical nonlinear control for multi-rotor asymptotic stabilization based on zero-moment direction*. Automatica, vol. 117, page 108991, 2020. (Cité en pages viii, 57, 58, 59 et 60.)
- [Misra 2022] Misra, A., Jayachandran, S., Kenche, S., Katoch, A., Suresh, A., Gundabattini, E., Selvaraj, S. K. et Legesse, A. A. *A Review on Vertical Take-Off and Landing (VTOL) Tilt-Rotor and Tilt Wing Unmanned Aerial Vehicles (UAVs)*. Journal of Engineering, vol. 2022, no. 1, page 1803638, 2022. [En ligne]. Disponible : <https://onlinelibrary.wiley.com/doi/abs/10.1155/2022/1803638>. (Cité en page 8.)
- [Murali 2024] Murali, K., Moreno, E. P. et Lustosa, L. R. *Flight Validation of a Global Singularity-Free Aerodynamic Model for Flight Control of Tail Sitters*. Dans 2024 IEEE International Conference on Robotics and Automation (ICRA), pages 3861–3867, 2024. (Cité en page 32.)
- [Nicosia 1990] Nicosia, S., Tornambe, A. et Valigi, P. *Experimental results in state estimation of industrial robots*. Dans 29th IEEE Conference on Decision and Control, pages 360–365 vol.1, 1990. (Cité en page 128.)
- [Nie 2024] Nie, H., Gu, F. et He, Y. *Hierarchical gain scheduling based tilt angle guided robust control during mode transition for tilt-rotor unmanned aircraft vehicle*. International Journal of Advanced Robotic Systems, vol. 21, no. 3, page 17298806241246334, 2024. (Cité en page 10.)
- [Okulski 2022] Okulski, M. et Ławryńczuk, M. *A small UAV optimized for efficient long-range and VTOL missions : An experimental tandem-wing Quadplane Drone*. applied sciences, vol. 12, no. 14, page 7059, 2022. (Cité en page 9.)
- [Olszanecki Barth 2020] Olszanecki Barth, J. M., Condomines, J.-P., Bronz, M., Moschetta, J.-M., Join, C. et Fliess, M. *Model-free control algorithms for micro air vehicles with transitioning flight capabilities*. International Journal

- of Micro Air Vehicles, vol. 12, pages 1–22, April 2020. [En ligne]. Disponible : <http://dx.doi.org/10.1177/1756829320914264>. (Cité en pages vii, 5, 20, 32, 34, 36, 40, 69 et 124.)
- [Ostermann 2012] Ostermann, T., Holsten, J., Dobrev, Y. et Moormann, D. *Control concept of a tiltwing uav during low speed manoeuvring*. Dans Proceeding of the 28th International Congress of the Aeronautical Sciences : ICAS Brisbane, Australia, volume 1190, 2012. (Cité en pages vii et 11.)
- [Pfimlin 2006] Pfimlin, J.-M. *Commande d'un minidrone à hélice carénée : de la stabilisation dans le vent à la navigation autonome*. Theses, SUPAERO, November 2006. (Cité en pages vii et 11.)
- [Ritz 2017] Ritz, R. et D'Andrea, R. *A global controller for flying wing tailsitter vehicles*. Dans 2017 IEEE International Conference on Robotics and Automation (ICRA), pages 2731–2738, 2017. (Cité en page 10.)
- [Rohr 2019] Rohr, D., Stastny, T., Verling, S. et Siegwart, R. *Attitude and cruise control of a VTOL tiltwing UAV*. IEEE Robotics and Automation Letters, vol. 4, no. 3, pages 2683–2690, 2019. (Cité en page 11.)
- [Rohr 2021] Rohr, D., Studiger, M., Stastny, T., Lawrance, N. R. J. et Siegwart, R. *Nonlinear Model Predictive Velocity Control of a VTOL Tiltwing UAV*. IEEE Robotics and Automation Letters, vol. 6, no. 3, pages 5776–5783, 2021. [En ligne]. Disponible : <http://dx.doi.org/10.1109/LRA.2021.3084888>. (Cité en page 14.)
- [Saeed 2018] Saeed, A. S., Younes, A. B., Cai, C. et Cai, G. *A survey of hybrid Unmanned Aerial Vehicles*. Progress in Aerospace Sciences, vol. 98, pages 91–105, April 2018. [En ligne]. Disponible : <http://dx.doi.org/10.1016/j.paerosci.2018.03.007>. (Cité en pages xi et 8.)
- [Sanfelice 2013] Sanfelice, R., Copp, D. et Nanez, P. *A toolbox for simulation of hybrid systems in Matlab/Simulink : Hybrid Equations (HyEQ) Toolbox*, 2013. (Cité en page 65.)
- [Sansou 2022a] Sansou, F. *Commande hybride d'un drone convertible pour des déplacements sous optimaux*. Master Thesis. ENAC Toulouse, 2022. [En ligne]. Disponible : <http://dx.doi.org/10.48550/ARXIV.2203.15387>. (Cité en pages 5, 32, 33 et 36.)
- [Sansou 2022b] Sansou, F., Demourant, F., Hattenberger, G., Loquen, T. et Zaccarian, L. *Open wind tunnel experiments of the DarkO tailsitter longitudinal stabilization with constant wind*. IFAC-PapersOnLine,

- vol. 55, no. 22, pages 1–6, 2022. [En ligne]. Disponible : <http://dx.doi.org/https://doi.org/10.1016/j.ifacol.2023.03.001>. 22nd IFAC Symposium on Automatic Control in Aerospace ACA 2022. (Cité en page 5.)
- [Sansou 2022c] Sansou, F. et Zaccarian, L. *On local-global hysteresis-based hovering stabilization of the DarkO convertible UAV*. Dans 2022 European Control Conference (ECC), pages 40–45, 2022. (Cité en page 5.)
- [Sansou 2024a] Sansou, F., Demourant, F., Hattenberger, G., Loquen, T. et Zaccarian, L. *Hovering stabilization of the DarkO tail-sitter drone with constant wind*. IEEE Transactions on Control Systems Technology, 2024. (Cité en pages 5 et 6.)
- [Sansou 2024b] Sansou, F., Hattenberger, G., Zaccarian, L., Demourant, F. et Loquen, T. *Modelling and Hovering Stabilisation of a Free-Rotating Wing UAV*. Dans 2024 International Conference on Unmanned Aircraft Systems (ICUAS), pages 779–785, La Canée (Crete), Greece, June 2024. IEEE. (Cité en page 6.)
- [Schlatter 2024] Schlatter, M., Ducard, G., Rohr, D. et Onder, C. *Longitudinal Control of a Tilt-rotor VTOL UAV using Incremental Nonlinear Dynamic Inversion*. Dans 2024 International Conference on Control, Automation and Diagnosis (ICCAD), pages 1–6. IEEE, 2024. (Cité en page 10.)
- [Shekhar 2018] Shekhar, A. et Sharma, A. *Review of Model Reference Adaptive Control*. Dans 2018 International Conference on Information , Communication, Engineering and Technology (ICICET), pages 1–5, 2018. (Cité en page 22.)
- [Sieberling 2010] Sieberling, S., Chu, Q. P. et Mulder, J. A. *Robust Flight Control Using Incremental Nonlinear Dynamic Inversion and Angular Acceleration Prediction*. Journal of Guidance, Control, and Dynamics, vol. 33, no. 6, pages 1732–1742, 2010. [En ligne]. Disponible : <https://doi.org/10.2514/1.49978>. (Cité en page 19.)
- [Simmons 2022] Simmons, B. M. et Murphy, P. C. *Aero-Propulsive Modeling for Tilt-Wing, Distributed Propulsion Aircraft Using Wind Tunnel Data*. Journal of Aircraft, vol. 59, no. 5, pages 1162–1178, 2022. (Cité en page 3.)
- [Smeur 2016] Smeur, E. J. J., Chu, Q. et de Croon, G. C. H. E. *Adaptive Incremental Nonlinear Dynamic Inversion for Attitude Control of Micro Air Vehicles*. Journal of Guidance, Control, and Dynamics, vol. 39, no. 3, pages 450–461, 2016. [En ligne]. Disponible : <https://doi.org/10.2514/1.G001490>. (Cité en pages 19 et 132.)

- [Smeur 2020] Smeur, E. J. J., Bronz, M. et de Croon, G. C. H. E. *Incremental Control and Guidance of Hybrid Aircraft Applied to a Tailsitter Unmanned Air Vehicle*. Journal of Guidance, Control, and Dynamics, vol. 43, no. 2, pages 274–287, 2020. [En ligne]. Disponible : <https://doi.org/10.2514/1.G004520>. (Cité en pages vii, 10, 11, 19, 40 et 132.)
- [Snyder 2021] Snyder, S., Zhao, P. et Hovakimyan, N. *Adaptive control for linear parameter-varying systems with application to a VTOL aircraft*. Aerospace Science and Technology, vol. 112, page 106621, 2021. [En ligne]. Disponible : <https://www.sciencedirect.com/science/article/pii/S1270963821001310>. (Cité en page 24.)
- [Stone 2002] Stone, H. et Wong, K. C. *Preliminary Design of a Tandem-Wing Tail-Sitter UAV Using Multi-Disciplinary Design Optimisation*. Dans International Aerospace Congress, 2002. (Cité en page 2.)
- [Stone 2008] Stone, R. H., Anderson, P., Hutchison, C., Tsai, A., Gibbens, P. et Wong, K. C. *Flight Testing of the T-Wing Tail-Sitter Unmanned Air Vehicle*. Journal of Aircraft, vol. 45, no. 2, pages 673–685, 2008. [En ligne]. Disponible : <http://dx.doi.org/10.2514/1.32750>. (Cité en page 2.)
- [Sun 2009] Sun, J., Yang, J. et Zhu, X. *Robust Flight Control Law Development for Tiltrotor Conversion*. Dans 2009 International Conference on Intelligent Human-Machine Systems and Cybernetics, volume 2, pages 481–484, 2009. (Cité en page 24.)
- [Syrmos 1997] Syrmos, V., Abdallah, C., Dorato, P. et Grigoriadis, K. *Static output feedback—A survey*. Automatica, vol. 33, no. 2, pages 125–137, 1997. [En ligne]. Disponible : <https://www.sciencedirect.com/science/article/pii/S0005109896001410>. (Cité en page 81.)
- [Tal 2022] Tal, E. et Karaman, S. *Global Incremental Flight Control for Agile Maneuvering of a Tailsitter Flying Wing*. arXiv preprint arXiv :2207.13218, 2022. (Cité en pages 3 et 10.)
- [Tangirala 2015] Tangirala, A. Principles of system identification : Theory and practice. CRC Press, 10 2015. (Cité en page 120.)
- [Team 2022] Team, P. *PaparazziUAV : System Architecture*, 2022. (Cité en pages x et 147.)
- [Tempo 2012] Tempo, R., Calafio, G. et Dabbene, F. Randomized algorithms for analysis and control of uncertain systems : With applications. Springer Publishing Company, Incorporated, 2nd édition, 2012. (Cité en page 62.)

- [Tregouet 2015] Tregouet, J.-F., Arzelier, D., Peaucelle, D., Pittet, C. et Zaccarian, L. *Reaction Wheels Desaturation Using Magnetorquers and Static Input Allocation*. IEEE Transactions on Control Systems Technology, vol. 23, no. 2, pages 525–539, February 2015. [En ligne]. Disponible : <http://dx.doi.org/10.1109/TCST.2014.2326037>. (Cité en page 50.)
- [Udwadia 2006] Udwadia, F. E. et Phohomsiri, P. *Explicit Equations of Motion for Constrained Mechanical Systems with Singular Mass Matrices and Applications to Multi-Body Dynamics*. Proceedings : Mathematical, Physical and Engineering Sciences, vol. 462, no. 2071, pages 2097–2117, 2006. [En ligne]. Disponible : <http://www.jstor.org/stable/20208995>. (Cité en page 120.)
- [Udwadia 2012] Udwadia, F. E. et Schutte, A. D. *A unified approach to rigid body rotational dynamics and control*. Proceedings : Mathematical, Physical and Engineering Sciences, vol. 468, no. 2138, pages 395–414, 2012. [En ligne]. Disponible : <http://www.jstor.org/stable/41345881>. (Cité en pages 120 et 121.)
- [Verling 2016] Verling, S., Weibel, B., Boosfeld, M., Alexis, K., Burri, M. et Siegwart, R. *Full Attitude Control of a VTOL tailsitter UAV*. Dans 2016 IEEE International Conference on Robotics and Automation (ICRA), pages 3006–3012, 2016. (Cité en page 10.)
- [Wang 2008] Wang, Q.-G., Zhang, Z., Astrom, K. J., Zhang, Y. et Zhang, Y. *Guaranteed Dominant Pole Placement with PID Controllers*. IFAC Proceedings Volumes, vol. 41, no. 2, pages 5842–5845, 2008. [En ligne]. Disponible : <https://www.sciencedirect.com/science/article/pii/S1474667016398779>. 17th IFAC World Congress. (Cité en page 22.)
- [Welstead 2012] Welstead, J. et Crouse, G. *A Segmented Freewing Concept for UAS Gust Alleviation in Adverse Environments*. Dans AIAA Infotech@Aerospace Conference, 2012. (Cité en page 12.)
- [Zhang 2017] Zhang, S., Fei, Q., Liang, J. et Geng, Q. *Modeling and control for longitudinal attitude of a twin-rotor tail-sitter unmanned aerial vehicle*. Dans 2017 13th IEEE International Conference on Control and Automation (ICCA), pages 816–821, 2017. (Cité en page 3.)

Résumé : Les drones sont devenus un outil dans de nombreux domaines tels que l'inspection, la surveillance ou la maintenance. Cependant, ils souffrent d'une autonomie limitée. Les *tailsitters* apportent une solution grâce à une grande enveloppe de vol et à une efficacité énergétique. Toutefois, les *tailsitters* sont grandement sujets aux perturbations aérologiques et notamment aux turbulences dans les phases stationnaires principalement. Cela est dû à la grande surface d'aile verticale, laquelle possède une grande prise au vent. De plus, leur corps tournant lors de la transition, il est donc compliqué de mesurer la vitesse de l'air. Ainsi, en stationnaire ou à faible vitesse, le vent n'est pas connu. Ce type de drone est sous-actionné puisque l'on trouve deux moteurs sur l'aile et deux surfaces aérodynamiques sur le bord de fuite. Le flux d'air des hélices soufflant les élévons, nous avons un couplage entre les actionneurs.

Cette thèse cherche à étudier la commande de drones dans des environnements perturbés ou en présence de vent. Les premiers travaux se sont concentrés sur la dynamique sans vent pour appréhender une dynamique simplifiée. Nous avons pu proposer une modification non-linéaire du vecteur de commande pour rendre ce modèle linéaire en commande. De ce modèle, nous avons proposé une loi de commande locale-globale fondée sur une dynamique hybride à hystérésis. Elle permet d'étendre le domaine de stabilité de la loi de commande linéaire agressive à l'aide d'une loi non-linéaire avec une grande région d'attraction, mais moins agressive.

La suite des travaux s'est concentrée sur la stabilisation d'un *tailsitter* soumis à des échelons de vent. Il en résulte une caractérisation des équilibres stationnaires pour un ensemble de conditions de vent et l'obtention de la représentation linéarisée de la dynamique du drone. À l'aide de ce modèle, il a été possible d'analyser les saturations des actionneurs et l'autorité disponible aux environs des points d'équilibre. Nous avons réalisé une stabilisation établie sur un retour de sortie, avec une action proportionnelle et intégrale. Cette commande n'utilise pas la mesure de l'angle de tangage du drone, car nous ne pouvons pas, a priori, connaître la valeur cible qui nécessiterait une estimation de la vitesse et de la direction du vent. L'optimisation de ce bouclage est effectuée à l'aide du logiciel "Systune" pour obtenir de bonnes propriétés de réjection de perturbation. Une approche incrémentale a été suivie, la loi de commande ayant été testée dans un premier temps sur une maquette à un degré de liberté face à une soufflerie ouverte. Une fois validée, la loi de commande a été implémentée dans le système de drone Paparazzi. Grâce à son architecture modulaire, il a été possible de nous interfaçer avec les codes d'estimation et de commande des actionneurs. Ainsi, nous avons pu réaliser des vols sur le modèle complet à six degrés de liberté.

Enfin, nous avons proposé une architecture inspirée du *tailsitter*, nommée *free-wing*. Nous avons développé un drone multicorps basé sur une aile en rotation libre sur son axe de tangage autour d'un fuselage. L'actionnement de l'aile est sensiblement le même que pour le *tailsitter* et le fuselage possède deux actionneurs pour se maintenir horizontal. Nous recherchons, dans cette architecture, une passivité naturelle à la turbulence induite par le changement naturel de l'incidence de l'aile en fonction du vent incident. Il s'agit aussi d'installer une charge utile sur le fuselage horizontal sur le domaine de vol. De plus, nous avons réalisé un modèle de simulation où la dynamique est obtenue à l'aide des équations de Udwadia-Kalaba

et de la phi-théorie. Enfin, nous nous sommes concentrés sur la stabilisation et le guidage du drone en utilisant une inversion incrémentale non-linéaire de la dynamique (INDI). Nous utilisons les actionneurs de l'aile et du fuselage pour obtenir une loi de stabilisation globale. Des vols ont validé l'intérêt de cette architecture.

Mots clés : Planification et contrôle des drones ; validation de modèles ; contrôle robuste ; systèmes électromécaniques

Abstract : Drones have become a prevalent tool in numerous fields, including inspection, surveillance, and maintenance. However, one area where they are currently lacking is autonomy. Tailsitter offer a viable solution, combining large flight envelopes with energy efficiency. Nonetheless, tailsitter unmanned aerial vehicles are particularly susceptible to disturbances, specifically turbulence during hover phases. This is because of the large vertical surface area of the wing, which offers a high degree of wind resistance. Furthermore, the drone's body rotates during transition, making it difficult to accurately measure airspeed. It is not possible to accurately measure wind speed, whether the drone is hovering or moving at a low speed. Additionally, this underactuated drone, has two motors on the wing and two aerodynamic surfaces on the trailing edge. The interaction of the airflow from the propellers with the elevons results in a coupling between the actuators.

The aim of this thesis is to examine the control of drone in challenging environments, with a particular emphasis on the impact of wind. However, initial research focused on windless dynamics in order to gain a better understanding of the simplified dynamics. We were able to propose a non-linear modification of the control vector to transform this nonlinear model into an input-affine model. Based on this model, we proposed a local-global control law based on hybrid dynamics with hysteresis, which allowed us to extend the stability domain of an aggressive linear control law by means of a non-linear law with a large region of attraction, but which was less aggressive.

Further work was conducted on the stabilization of a tailsitter drone under wind steps. This resulted in a characterization of stationary equilibria for a range of wind conditions and a linearized representation of the drone's dynamics. Using this model, we were able to analyze the saturation levels of actuators and the authority available close to the equilibrium points. We have implemented a stabilization system based on output feedback with proportional and integral action, derived from the model. This control does not utilize the drones pitch angle measurement, since the target value is not known a priori. It would require an estimation of wind speed and direction, which is not feasible. The loop is optimized using "Systune" software in order to achieve effective disturbance rejection properties. We adopted an incremental approach by initially evaluating the control law on a one-degree-of-freedom model against an open wind tunnel. Upon validation, we proceeded to implement the control law within the Paparazzi UAV system. Due to its modular design, we were able to establish a connection with the state estimation and actuators, enabling us to execute flights on the entire model with six degrees of freedom.

We then proposed an architecture inspired by the tailsitter, called freewing. We have developed a multi-body drone that is based on a wing that can freely rotate on its pitch axis around a fuselage. The wing is driven the same way as the tailsitter, and the fuselage has a motor and an aerodynamic surface to keep it horizontal. With this architecture, we aim to achieve a natural passivity to turbulence induced by the natural change in wing incidence as a function of the incident wind. However, there is also the possibility of installing a payload on the horizontal fuselage. To obtain a simulation model, we have modeled the drone's dynamics using the Udwadia-Kalaba equations and the phi-theory. We focused on stabilization and guidance of the UAV through the use of incremental nonlinear dynamic inversion (INDI). The

wing and fuselage actuators are used to achieve a global stabilization law. In order to evaluate the benefits of this architecture, we conducted flight tests.

Keywords : UAV Planning and Control ; Model Validation ; Robust control ; Electromechanical systems
