THÈSE

En vue de l'obtention du

DOCTORAT DE L'UNIVERSITÉ FÉDÉRALE TOULOUSE MIDI-PYRÉNÉES

Délivré par :
l'École Nationale de l'Aviation Civile (ENAC)

Présentée et soutenue le *jj/mm/aaaa* **par :**
Florian SANSOU

Control actif de la turbulence sur un micro drone convertible

---

JURY

Mme Sophie TARBOURIECH	Directrice de recherche LAAS-CNRS	Présidente du jury
M. Paolo ROBUFFO GIORDANO	Directeur de recherche IRISA-CNRS	Rapporteur
M. Pascal MORIN	Professeur des universités Sorbonne Université	Rapporteur
Philippe CHEVREL	Enseignant chercheur IMT ATLANTIQUE	Examinateur
M. Fabrice DEMOURANT	Ingénieur de recherche ONERA	Directeur de thèse
M. Gautier HATTENBERGER	Enseignant chercheur ENAC	Co-directeur de thèse

---

École doctorale et spécialité :
EDSYS : Automatique 4200046
Unité de Recherche :
École Nationale d'Aviation Civile
Directeur(s) de Thèse :
M. Fabrice DEMOURANT et *M. Gautier HATTENBERGER*
Rapporteurs :
M. Paolo ROBUFFO GIORDANO et *M. Pascal MORIN*

Remerciements

A faire en dernier :-)

Table des matières

Liste des figures viii

Liste des tableaux ix

Liste des algorithmes xi

Liste des abréviations xiii

1 Généralités sur les drones 1

1.1 Drone autonome 1
1.2 Micros drone convertible 1
1.2.1 Domaine de vol 1
1.2.2 Types d'architecture des drones convertibles 2
1.3 Propriétés des *tailsitters* et des *freewings* 4
1.3.1 Actionnement 5
1.3.2 Aérodynamique 5
1.4 Modélisations 5
1.5 Commandes 5
1.6 Contexte de la thèse 5
1.7 Présentation de la thèse 5

2 Objectifs de commande 7

2.1 Contexte opérationnel 7
2.2 Perturbations 7
2.3 Résumée 7

3 Modélisation d'un drone convertible : DarkO 9

3.1 Modèle du drone DarkO 9
3.1.1 Modèle non-linéaire complet 11
3.1.2 Modèle non linéaire simplifiée à basse vitesse 14
3.2 Identification des paramètres du modèle 16
3.2.1 Modélisation des actionneurs 16
3.3 Équilibres stationnaires 17
3.3.1 Équilibre stationnaire sans vent 17
3.3.2 Équilibre stationnaire en présence de vent 19
3.4 Dynamiques linéarisés 24

3.4.1 Dynamique linéarisé sans vent 24
3.4.2 Dynamique linéarisé en présence de vent 25
3.5 Conclusion du Chapitre 3 32

4 Commande hybride 33
4.1 Motivation 33
4.1.1 Nonlinear dynamic feedback controller 34
4.1.2 Linear control design 36
4.1.3 Hysteresis-based local-global control design 37
4.2 Schéma de commande hybride 39
4.3 Simulations 39

5 Étude longitudinale sur une maquette à trois degrés de liberté 41
5.1 Motivation 41
5.2 Schéma de commande linéaire proportionnel intégral 41
5.3 Maquette expérimentale 41
5.4 Test bench presentation 41
5.4.1 Motivation for the design 41
5.4.2 Physical description, sensing, and actuation 42
5.4.3 Software-in-the-loop translational motion 45
5.5 Integral-based linear control 46
5.5.1 Description of the control scheme 46
5.5.2 H_∞-based optimization 48
5.6 Résultats 49

6 Commande proportionnelle intégrale d'un drone convertible à 6 degrés de liberté 51
6.1 Schéma de commande linéaire proportionnel intégral : 6 Dof 51
6.2 Integral-based linear control 51
6.2.1 Description of the control scheme 51
6.2.2 Zero-wind H_∞-based controller tuning 54
6.2.3 Multimodel H_∞-based controller tuning 56
6.3 Experimental flight with open wind tunnel 59
6.4 Maquette expérimentale : 6 Dof 60
6.5 Résultats 60

7 Modélisation d'un drone à aile libre rotation libre 67
7.1 Design et modélisation d'un drone : Colibri 67
7.2 Design and modelling of Colibri UAV 67

7.3 State estimation 72
7.3.1 Sensors placement 72
7.3.2 Angular speed estimation 73
7.3.3 Wing state estimation 74
7.3.4 Fuselage orientation estimation 76

8 Commande d'un drone à aile libre rotation libre 79
8.1 Inversion non linéaire incrémentale de la dynamique du drone 79
8.2 Experimentation 80
8.3 Commande Udwadia-Kalaba 81
8.4 Vols expérimentaux 81

Conclusion 83

A Exemple d'annexe 85
A.1 Exemple d'annexe 85

Bibliographie 87

Liste des figures

3.1 Vue de dessus d'un autopilote Apogee v1.00. 10

3.2 Repère de référence de DarkO avec une représentation schématique des actionneurs. 11

3.3 Input-output response of an Esc-Motor-Propeller assembly. 16

3.4 Bifilar pendulum mounting for the identication of $\boldsymbol{J}$. 17

3.5 Bifilar pendulum identication of $\boldsymbol{J}$. 18

3.6 Parameters (θ, δ, τ) of the equilibrium point (surface) established in Theorem 1 and Algorithm 1 for constant horizontal and vertical wind ($w_{\mathrm{r}x}$,$w_{\mathrm{r}z}$), and actuators saturation levels (red). 23

3.7 Parameters (θ, δ, τ) of the equilibrium point (blue) established in Theorem 1 and Algorithm 1 for a constant horizontal wind $w_{\mathrm{r}x}$, and actuators saturation levels (red). 24

4.1 Simulation with the nonlinear dynamic feedback controller. 35

4.2 Simulation of the full model (solid) and (3.8) (dashed) with $\boldsymbol{u} = \boldsymbol{u}_{\mathrm{lin}}$ as in (4.2) from an initial condition $\tilde{\boldsymbol{x}}_{\mathbf{0}}$ within the basin of attraction. 37

4.3 Diverging simulation of the full model with $\boldsymbol{u} = \boldsymbol{u}_{\mathrm{lin}}$ as in (4.2) from an initial condition $\tilde{\boldsymbol{x}}_{\mathbf{0}}$ outside the basin of attraction. 37

5.1 The one degree-of-freedom joint. 43

5.2 Single degree-of-freedom DarkO model in front of the WindShape. . 43

5.3 Virtual flight testing architecture : WindShape (a) ; Airspeed sensor (b) ; Force/Torque sensor (c) ; Rotary encoder (d) ; Servomotor (e) ; Brussless motor + ESC (f) ; LabJack (g) ; Control computer (h) . . . 44

5.4 Proposed integral-based controller. 47

6.1 Proposed integral-based controller with the wind perturbation $\boldsymbol{w}$, a plant-input perturbation $\boldsymbol{d}$ and a plant-output perturbation ν. . . . 52

6.2 Simulation of the non-linear model (3.1) (solid line) and the linearized model (3.31) (dashed line) with increasing constant wind steps with the controller tuned using the zero-wind optimization of Section 6.2.2. 61

6.3 Diagrams of the singular values of the transfer functions in (6.8) at the first iteration of Algorithm 3. 62

6.4 Simulation of the non-linear model (3.1) (solid line) and the linearized model (3.31) (dashed line) with increasing constant wind steps with the controller tuned using the multimodel optimization of Algorithm 3 in Section 6.2.3. 63
6.5 DarkO's experimental flight in front of the open wind tunnel. 64
6.6 Experiment of the DarkO UAV in front of the wind tunnel with increasing constant wind levels (lower plot). 65
6.7 Statistical visualization of the hovering performance. 66

7.1 Inertial (I) and wing (W) reference frames and the Colibri architecture. 68
7.2 Inertial (I), fuselage (F) and wing (W) reference frames and forces acting on the Colibri UAV. 70
7.3 Position and orientation simulation of the multi-body UAV Colibri in closed loop with a simple double-loop controller. 77
7.4 Angular position measurement (black,top plot), wing gyro velocity measurement (blue,bottom plot), finite difference velocity estimation (green, bottom plot) and high-gain estimates (red curves) 78

8.1 Colibri experimental prototype. 81
8.2 Position and orientation of the reference frame wing in the first six graphs and pivot angle measurement on the last graph below during real flight. 82

Liste des tableaux

3.1 Paramètres numériques identifiés du modèle DarkO. 12

6.1 Standard deviation of the modeled sensor noise added to the simulated measurements. 54
6.2 Values of the positive weighting scalars W_1–W_5 used in the execution of Algorithm 3. 58

7.1 Numerical parameters of the Colibri model. 69

Liste des algorithmes

1 Obtention des paramètres d'équilibre en (3.16). 20
2 Détermination des matrices de la linéarisation de (3.31) 31
3 Iterative multimodel controller gain tuning. 57

Liste des abréviations

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

Chapitre 1

Généralités sur les drones

1.1 Drone autonome

Ces dernières années, le domaine des drones s'est considérablement développé, de nombreux progrès ont été réalisé dans la conduite de vol autonome qui permette de réaliser de nombreuse tache longue, répétitive ou dangereuse de manière plus sûre que des avions ou des systèmes télépiloté. Les drones ont fait leurs preuves dans de nombreuses applications civiles, alors qu'ils étaient auparavant conçus à des fins de surveillance et de destruction dans le secteur militaire. La possibilité d'utiliser des systèmes de vol autonomes dans le secteur civil a commencé avec l'accessibilité croissante proposée par l'industrie commerciale grâce à des solutions à faible coût pour les applications d'imagerie aérienne. L'intérêt porté aux applications d'imagerie aérienne a motivé le développement de plusieurs projets, notamment dans les domaines de l'aide humanitaire, des secours en cas de catastrophe, de la recherche et du sauvetage, des opérations de sécurité, de la surveillance, de l'agriculture de précision et de l'inspection des infrastructures civiles.

1.2 Micros drone convertible

Les micros drone sont une gamme de drone intéressante par leurs petites tailles qui leur permettent d'intervenir dans des espaces confiné ou contraint. Ils n'ont cependant qu'une charge utile restreinte souvent limitée à l'emport d'une caméra ou d'un colis de faible masse.

1.2.1 Domaine de vol

Tout l'intérêt d'un drone convertible réside dans la capacité à décoller et atterrir verticalement, tout en conservant une bonne efficacité énergétique, en vol d'avancement, conféré par une aile. Cette aile a l'avantage de généré de la portance, qui s'oppose au poids un drone et permet d'assurer sa sustentation. La contrepartie de la génération de la portance est le trainé qui s'oppose à l'avancement et doit être contré par une force de traction générée par les hélices. Nous pouvons définir l'efficacité énergétique comme le ratio entre le temps de vol et l'énergie électrique nécessaire pour effectuer ce vol. Nous pouvons comparer l'efficacité d'un drone quadrirotor qui assure sa sustentation uniquement grâce à des hélices (à l'instar d'un hélicoptère), a celle d'un drone à voilure fixe et à celle d'un convertible. Drone à

Type d'architecture	Vitesse	Stationnaire	Temps de vol
Drone à voilure fixe			
Quadrirotor			
Drone convertible			

voilure fixe, stationnaire impossible, grande efficité ernegetique vitesse mini et max Quadrirotor, grande manouvrabilité, faible efficaité energetique Convertible, domaine de vol important, sensibilité aux pertubation, grand efficaticté energetique en vol d'avancement, faible en stationnaire

Nous observons qu'un drone convertible possède un domaine de vol bien plus important qu'un drone à voilure fixe qui ne va pas pouvoir voler à basse vitesse et qu'un quadrirotor dont l'autonomie vas être limité par sa consommation. Ainsi un drone convertible semble un outil tout à fait approprié pour de nombreuse mission.

1.2.2 Types d'architecture des drones convertibles

La conception structurelle et aérodynamique d'un drone est le facteur principal permettant des transitions stable et fluide. De plus, il est nécessaire d'optimiser l'architecture pour une mission de manière à être le plus efficace dans la tache principale du drone. Au vu de la diversité des missions, un grand nombre d'architectures ont été proposé et que nous pouvons catégoriser en quatre classes : *quadplanes*, *tiltrotor*, *tailsitter*, *tiltwing*. Nous ajoutons la catégorie *quadplanes* aux trois autres catégories (*tiltrotor*, *tailsitter* et *tiltwing*) classiquement utilisé lors des études bibliographiques sur les drones convertibles [Saeed 2018, Ducard 2021].

Quadplanes

Les *quadplanes* sont conçu par la fusion d'un avion et d'un quadrirotor, ce qui permet un découpage de l'actionnement en fonction de la phase de vol. Le premier système de propulsion est composé de quatre hélices générant une force verticale permet le contrôle lors de phase de décollage, d'atterrissage et de stationnaire. Le second système de propulsion est composé d'une propulsive supplémentaire afin d'atteindre des vitesses en vol d'avancement.

L'avantage de ce type d'architecture est sa grande robustesse. Effectivement, aucune pièce en mouvement est nécessaire pour réaliser la transition ce que réduit le risque de défaillance mécanique. L'inconvénient est le manque d'efficacité. Lors d'un vol d'avancement, la portance sera générée par l'aile, ainsi il est possible de désactiver les rotors qui génèrerons des perturbations aérodynamiques et des trainés parasites. Effectivement, 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 trainé. De plus, la surcharge engendrée par l'emport de moteur supplémentaire se traduit par une diminution de la charge utile transportable.

En termes de contrôle, un atout indéniable est la séparation des actionnements en fonction de phase 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. Ce critère est pertinent, car il est lié à l'efficacité 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 associé et dans les vitesses plus importante, la commutation de 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 *tiltrotor* nécessitent l'utilisation d'un actionneur supplémentaire afin d'effectuer la transition. 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. Ainsi l'angle entre le souffle des hélices et l'aile peut être ajusté à chaque instant. Cet angle joue un rôle important dans le contrôle des forces et des moments aérodynamiques : sa maîtrise permet de mieux gérer non seulement 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 *tiltrotor* embarqué 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 assurée la sustentation en stationnaire que pour générer la traction en mode avion

Tailsitter

Contrairement au *tiltrotor* qui se pose sur le fuselage de l'avion, les *tailsitter* se posent à la verticale. Durant la transition du mode stationnaire au vol d'avancement la structure entière bascule vers l'avant modifiant l'angle d'incidence de la voiture. Selon la configuration du *tailsitter*, la transition peut être réalisée soit par la génération du moment aérodynamique créé par les élevons, soit par le couple créé par le système de propulsion. Pendant le vol d'avancement, en position horizontale, le *tail-sitter* 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.

Ils semblent être la configuration le plus abouti des drones convertibles, car il utilise les mêmes actionneurs dans tout le domaine de vols. Ainsi, il n'embarque aucune masse superflue.

ation

Tiltwing

La particularité des *tiltwings* réside dans le fait que les rotors sont inclinées en même temps que les ailes. Un actionneur supplémentaire et puissant est donc nécessaire pour surmonter le couple de l'aile afin de la positionner dans l'orientation souhaitée. La commande de cet actionneur doit être prise en compte lors de la conception des lois de commande. 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 aux vents 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

Freewing Une gamme en cours de développement à l'intérieur de l'architecture des *tiltwing* est les *freewing*. Ils sont actionnés comme des *tiltwing* sauf au niveau de l'axe de rotation entre l'aile et le fuselage. Cette rotation est laissée libre, ce degrés de liberté permet à l'aile de changer librement son incidence.

1.3 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 particulière reste un processus important qui nécessite plus d'attention.

1.3.1 Actionnement

Dans ces deux architectures, il est courant de trouver des actionneurs basés sur des effets aérodynamiques. Ces actionneurs ont l'avantage d'être peut consommateur en énergies, ils ont mue par des servomoteurs qui consomme peu d'électricité proportionnellement aux couples qu'ils génèrent. Dans le cas des ailes volantes, les surfaces aérodynamiques sont souvent placé sur la partie arrière des ailes et peuvent être utilisé symétriquement similairement à des volets ou anti-symétriquement comme des ailerons. Nous utiliserons donc la contraction des deux mots anglais pour définir ces surfaces aérodynamiques qui porte le nom d'élevon. Dans les phases de stationnaire, atterrissage ou décollage, la plateforme est maintenue en vol par les hélices, ainsi il est nécessaire de dimensionner les groupes moteurs-hélices pour qui puisse 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 surface soufflé par le flux d'air des hélices, il existe un couplage des actionneurs qui complexifie la modélisation et le contrôle de ces architectures.

1.3.2 Aérodynamique

1.4 Modélisations

Compromis précision du modèle et commande

Modèle de turbulence dryden

1.5 Commandes

1.6 Contexte de la thèse

1.7 Présentation de la thèse

Chapitre 2

Objectifs de commande

Sommaire

2.1 Contexte opérationnel 7
2.2 Perturbations 7
2.3 Résumée 7

2.1 Contexte opérationnel

rejet de perturba-
tion, model based
control

2.2 Perturbations

2.3 Résumée

Chapitre 3

Modélisation d'un drone convertible : DarkO

Sommaire

3.1 Modèle du drone DarkO **9**
3.1.1 Modèle non-linéaire complet 11
3.1.2 Modèle non linéaire simplifiée à basse vitesse 14
3.2 Identification des paramètres du modèle **16**
3.2.1 Modélisation des actionneurs 16
3.3 Équilibres stationnaires **17**
3.3.1 Équilibre stationnaire sans vent 17
3.3.2 Équilibre stationnaire en présence de vent 19
3.4 Dynamiques linéarisés **24**
3.4.1 Dynamique linéarisé sans vent 24
3.4.2 Dynamique linéarisé en présence de vent 25
3.5 Conclusion du Chapitre 3 . . . . **32**

3.1 Modèle du drone DarkO

DarkO est un drone conçu et développé à l'École Nationale de l'Aviation Civile (ENAC) de Toulouse (France), est un exemple clair 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 Apogee [1] fabriquée à l'ENAC, voir Fig. 3.1.

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

---

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

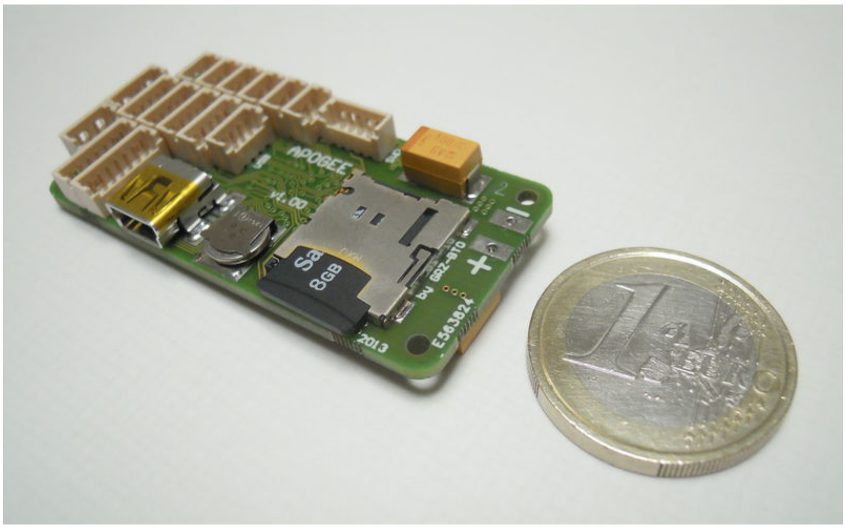

FIGURE 3.1 – Vue de dessus d'un autopilote Apogee v1.00.

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 flasher avec un firmware AM32. La communication sol-bord est réalisée via un canal bidirectionnel basé sur des modules XBee-PRO S1.

uver un syno-
ne

Les actionneurs de DarkO peuvent être décomposé en deux catégories. La première est composée de deux hélices (T-Motor T5147) placées symétriquement à l'avant de l'aile (illustrées en **noir** dans la Fig. 3.2) 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 élevons, placés à l'arrière de l'aile (illustrés en bleu dans la Fig. 3.2), agissant en tant que surfaces de contrôle. Les élevons génèrent des forces et des moments en modifiant leurs incidences relativement au flux d'air dans lequel ils sont placé. Ce flux d'air peut être généré par le vent relatif (liée à la vitesse du drone), le vent extérieur, mais aussi par le souffle des hélices. Les élevons sont commandés par deux servomoteurs MKS DS65K.

La figure 3.2 montre le modèle de DarkO, ainsi qu'un repère de référence inertiel 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 $z_\mathrm{b} = 0$), 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/élévateur gauche et droit sont désignés par les indices $i = 1$ (gauche) et $i = 2$ (droite). La convention de signe sera définie comme positive pour les positions des élevons δ_1, δ_2 lorsqu'ils créent un moment à cabrer avec les hélices tournant dans des directions

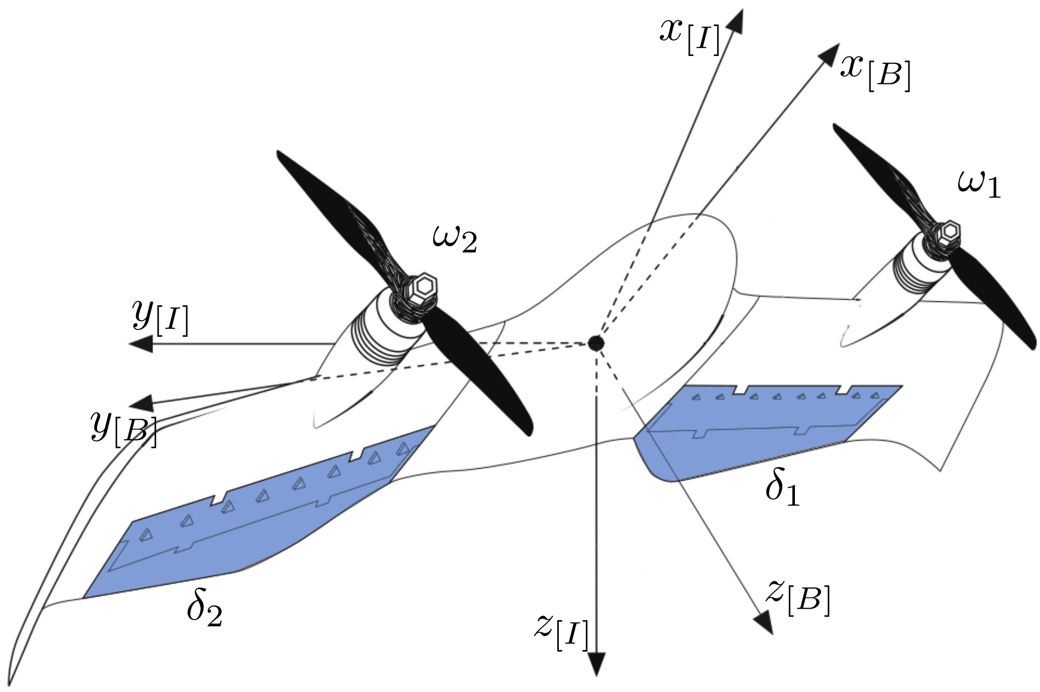

FIGURE 3.2 – Repère de référence de DarkO avec une représentation schématique des actionneurs.

opposées avec des vitesses angulaires $\omega_1 > 0$ et $\omega_2 < 0$, respectivement.

3.1.1 Modèle non-linéaire complet

En exploitant la modélisation présentée dans [Lustosa 2019] et [Olszanecki Barth 2020], un modèle précis de la dynamique de DarkO décrit la position $\boldsymbol{p} \in \mathbb{R}^3$ du centre de gravité et sa vitesse $\boldsymbol{v} = \dot{\boldsymbol{p}} \in \mathbb{R}^3$, en plus de son orientation, bien représentée par un quaternion $\boldsymbol{q} \in \mathbb{S}^3 := \{\boldsymbol{q} \in \mathbb{R}^4 : |\boldsymbol{q}| = 1\}$ et de sa vitesse angulaire $\boldsymbol{\omega}_\mathrm{b}$ représentée dans le repère du corps, qui satisfait $boldsymbol\dot{q} = \frac{1}{2}\boldsymbol{q} \otimes \left[\begin{smallmatrix} 0 \\ \boldsymbol{\omega}_\mathrm{b} \end{smallmatrix}\right]$, où $\otimes$ représente le produit de Hamilton (voir [Lustosa 2019, Olszanecki Barth 2020] ou le tutoriel [Hua 2013] pour plus de détails). En choisissant l'état global comme $\boldsymbol{x} := (\boldsymbol{p},\ \boldsymbol{v},\ \boldsymbol{q},\ \boldsymbol{\omega}_\mathrm{b})$, le modèle mathématique dérivé dans [Lustosa 2019], dépendent d'un ensemble de paramètres énumérés dans le tableau 7.1, où nous indiquons également la valeur obtenue à partir d'une identification du système [Sansou 2022]. Le modèle dynamique peut être écrit comme ci-dessous :

Paramètres et coefficients	Valeurs	Unités
m (Masse du drone)	0.519	kg
b (Envergure)	0.542	m
c (Corde aérodynamique)	0.13	m
$\boldsymbol{B} = \mathrm{diag}(b, c, b)$	$\mathrm{diag}(0.542, 0.13, 0.542)$	m
S (Surface de l'aile)	0.026936	m^2
S_{wet} (Surface soufflée)	0.0180	m^2
S_{p} (Surface de helice)	0.0127	m^2
$\boldsymbol{J} = \mathrm{diag}(J_x, J_y, J_z)$	$\mathrm{diag}(0.0067, 0.0012, 0.0082)$	$\mathrm{kg\,m}^2$
k_{f} (Poussée des hélices)	1.7800e-8	kg m
k_{m}(Moment des hélices)	2.1065e-10	$\mathrm{kg\,m}^2$
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 élevons)	0.2	–
ξ_{m} (Moment des élevons)	1.4	–
ρ (Densité de l'air)	1.225	$\mathrm{kg\,m}^{-3}$
C_{d} (Trainé)	0.1644	–
C_y (Lateral)	0	–
C_ℓ (Portance)	5.4001	–
Δ_{r} (Centrage du drone)	-0.0145	m

TABLE 3.1 – Paramètres numériques identifiés du modèle DarkO.

$$\begin{cases} m\dot{\boldsymbol{v}} = -m\boldsymbol{g} + \boldsymbol{R}(\boldsymbol{q})\boldsymbol{F}_{\mathrm{b}}, & (3.1a) \\ \boldsymbol{J}\dot{\boldsymbol{\omega}}_{\mathrm{b}} = -\left[\boldsymbol{\omega}_{\mathrm{b}}\right]_\times \boldsymbol{J}\boldsymbol{\omega}_{\mathrm{b}} + \boldsymbol{M}_{\mathrm{b}}, & (3.1b) \end{cases}$$

où $\boldsymbol{g} := \left[\,0\ 0\ 9.81\,\right]^\top$ désigne le vecteur de gravité, $m \in \mathbb{R}$ est la masse, $\boldsymbol{J} \in \mathbb{R}^{3\times 3}$ est le moment d'inertie diagonal (voir Tableau 7.1) et en partitionnant le quaternion $\boldsymbol{q} \in \mathbb{S}^3$ comme $\boldsymbol{q} := \left[\eta\ \boldsymbol{\epsilon}^\top\right]^\top$, la matrice de rotation correspondante est $\boldsymbol{R}(\boldsymbol{q}) \in SO(3) := \{\boldsymbol{R} \in \mathbb{R}^{3\times 3} :\ \boldsymbol{R}^\top \boldsymbol{R} = \mathbb{I}_3, \det(\boldsymbol{R}) = 1\}$ est défini comme (voir [Hua 2013])

$$\boldsymbol{R}(\boldsymbol{q}) := \mathbb{I}_3 + 2\eta\left[\boldsymbol{\epsilon}\right]_\times + 2\left[\boldsymbol{\epsilon}\right]_\times^2. \tag{3.2}$$

D'après [Lustosa 2019] le vecteur de force et de moment $\boldsymbol{F}_{\mathrm{b}}$ et $\boldsymbol{M}_{\mathrm{b}}$ dans (3.1) dépendent (i) de l'état du système $\boldsymbol{x}$, (ii) de la perturbation $\boldsymbol{w} \in \mathbb{R}^3$, représentant la vitesse du vent dans le référentiel inertiel, et (iii) de la commande des actionneurs (voir Figure 3.2), 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 d'abord l'effet des commandes des actionneurs. Chaque hélice génère une poussée $\boldsymbol{T}_i$ orienté dans la direction x du repère corps et un moment $\boldsymbol{N}_i$ selon le même axe :

$$\boldsymbol{T}_i := \begin{bmatrix} \tau_i \\ 0 \\ 0 \end{bmatrix} := \begin{bmatrix} k_\mathrm{f}\omega_i^2 \\ 0 \\ 0 \end{bmatrix}, \ \boldsymbol{N}_i := (-1)^i \frac{k_\mathrm{m}}{k_\mathrm{f}} \boldsymbol{T}_i, \quad i = 1, 2. \tag{3.3}$$

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

$$\boldsymbol{\Delta}_i^\mathrm{f} := \begin{bmatrix} 0 & 0 & \xi_\mathrm{f}\delta_i \\ 0 & 0 & 0 \\ -\xi_\mathrm{f}\delta_i & 0 & 0 \end{bmatrix}, \ \boldsymbol{\Delta}_i^\mathrm{m} := \begin{bmatrix} 0 & 0 & \xi_\mathrm{m}\delta_i \\ 0 & 0 & 0 \\ -\xi_\mathrm{m}\delta_i & 0 & 0 \end{bmatrix}, \tag{3.4}$$

$i = 1, 2$. Les paramètres constants k_f, k_m, ξ_f, ξ_m apparaissant dans (3.3) et (3.4) sont listés dans la Table 7.1.
Avec les quantités ci-dessus, nous pouvons réarranger la dynamique donnée dans le tableau suivant [Lustosa 2019, eqns (97), (98)] (voir aussi [Sansou 2022]) et exprimer $\boldsymbol{F}_\mathrm{b}$ et $\boldsymbol{M}_\mathrm{b}$ dans (3.1) comme

$$\begin{aligned} \boldsymbol{F}_\mathrm{b} := \boldsymbol{T}_1 + \boldsymbol{T}_2 &+ \frac{S_\mathrm{wet}}{4S_\mathrm{p}} \boldsymbol{\Phi}^{(\mathrm{fv})} \Big((\boldsymbol{\Delta}_1^\mathrm{f} - \mathbb{I}_3)\boldsymbol{T}_1 + (\boldsymbol{\Delta}_2^\mathrm{f} - \mathbb{I}_3)\boldsymbol{T}_2 \Big) \\ &+ \frac{1}{4}\rho S \boldsymbol{\Phi}^{(\mathrm{fv})} \Big(\boldsymbol{\Delta}_1^\mathrm{f} + \boldsymbol{\Delta}_2^\mathrm{f} - 2\mathbb{I}_3 \Big) \|\boldsymbol{v}_\mathrm{b}\| \boldsymbol{v}_\mathrm{b} \\ &+ \frac{1}{4}\rho S \boldsymbol{\Phi}^{(\mathrm{mv})} \Big(\boldsymbol{\Delta}_1^\mathrm{f} + \boldsymbol{\Delta}_2^\mathrm{f} - 2\mathbb{I}_3 \Big) \boldsymbol{B} \|\boldsymbol{v}_\mathrm{b}\| \boldsymbol{\omega}_\mathrm{b}, \end{aligned} \tag{3.5}$$

$$\begin{aligned} \boldsymbol{M}_\mathrm{b} :&= \boldsymbol{N}_1 + \boldsymbol{N}_2 + \begin{bmatrix} p_x \\ p_y \\ 0 \end{bmatrix}_\times \boldsymbol{T}_1 + \begin{bmatrix} p_x \\ -p_y \\ 0 \end{bmatrix}_\times \boldsymbol{T}_2 \\ &- \frac{S_\mathrm{wet}}{4S_\mathrm{p}} \left(\boldsymbol{B}\boldsymbol{\Phi}^{(\mathrm{mv})} (\boldsymbol{\Delta}_1^\mathrm{m} - \mathbb{I}_3) + \begin{bmatrix} 0 \\ a_y \\ 0 \end{bmatrix}_\times \boldsymbol{\Phi}^{(\mathrm{fv})} (\boldsymbol{\Delta}_1^\mathrm{m} + \mathbb{I}_3) \right) \boldsymbol{T}_1 \\ &- \frac{S_\mathrm{wet}}{4S_\mathrm{p}} \left(\boldsymbol{B}\boldsymbol{\Phi}^{(\mathrm{mv})} (\boldsymbol{\Delta}_2^\mathrm{m} - \mathbb{I}_3) + \begin{bmatrix} 0 \\ -a_y \\ 0 \end{bmatrix}_\times \boldsymbol{\Phi}^{(\mathrm{fv})} (\boldsymbol{\Delta}_2^\mathrm{m} + \mathbb{I}_3) \right) \boldsymbol{T}_2 \\ &+ \frac{1}{4}\rho S \Bigg(\left(\begin{bmatrix} 0 \\ a_y \\ 0 \end{bmatrix}_\times \boldsymbol{\Phi}^{(\mathrm{fv})} + \boldsymbol{B}\boldsymbol{\Phi}^{(\mathrm{mv})} \right) \boldsymbol{\Delta}_1^\mathrm{m} \\ &+ \left(\begin{bmatrix} 0 \\ -a_y \\ 0 \end{bmatrix}_\times \boldsymbol{\Phi}^{(\mathrm{fv})} + \boldsymbol{B}\boldsymbol{\Phi}^{(\mathrm{mv})} \right) \boldsymbol{\Delta}_2^\mathrm{m} - 2\boldsymbol{B}\boldsymbol{\Phi}^{(\mathrm{mv})} \Bigg) \|\boldsymbol{v}_\mathrm{b}\| \boldsymbol{v}_\mathrm{b} \\ &+ \frac{1}{4}\rho S \Bigg(\left(\begin{bmatrix} 0 \\ a_y \\ 0 \end{bmatrix}_\times \boldsymbol{\Phi}^{(\mathrm{mv})} + \boldsymbol{B}\boldsymbol{\Phi}^{(\mathrm{m}\omega)} \right) \boldsymbol{\Delta}_1^\mathrm{m} \\ &+ \left(\begin{bmatrix} 0 \\ -a_y \\ 0 \end{bmatrix}_\times \boldsymbol{\Phi}^{(\mathrm{mv})} + \boldsymbol{B}\boldsymbol{\Phi}^{(\mathrm{m}\omega)} \right) \boldsymbol{\Delta}_2^\mathrm{m} - 2\boldsymbol{B}\boldsymbol{\Phi}^{(\mathrm{m}\omega)} \Bigg) \boldsymbol{B} \|\boldsymbol{v}_\mathrm{b}\| \boldsymbol{\omega}_\mathrm{b}, \end{aligned} \tag{3.6}$$

où $\boldsymbol{v}_{\mathrm{b}} := \boldsymbol{R}^{\top}(\boldsymbol{q})(\boldsymbol{v} - \boldsymbol{w})$ représente la vitesse de l'air vu par le drone exprimé dans le repère du corps. Dans [Lustosa 2019], la valeur $\|\boldsymbol{v}_{\mathrm{b}}\|$ apparaissatn dans les expressions de $\boldsymbol{F}_{\mathrm{b}}$ et $\boldsymbol{M}_{\mathrm{b}}$ est remplacé par la valeur $\eta = \sqrt{\|\boldsymbol{v}_{\mathrm{b}}\|^2 + \mu c^2 \|\boldsymbol{\omega}_{\mathrm{b}}\|^2}$, avec $\mu \in \mathbb{R}$ étant un paramètre lié à l'identification du modèle, mais dans le cas de DarkO [Sansou 2022], l'identification fournit $\mu = 0$, c'est pourquoi nous présentons ici une description simplifiée. La matrice des coefficients aérodynamiques constants $\boldsymbol{\Phi} := \begin{bmatrix} \boldsymbol{\Phi}^{(\mathrm{fv})} & \boldsymbol{\Phi}^{(\mathrm{mv})\top} \\ \boldsymbol{\Phi}^{(\mathrm{mv})} & \boldsymbol{\Phi}^{(\mathrm{m}\omega)} \end{bmatrix} \in \mathbb{R}^{6\times 6}$, est défini dans [Olszanecki Barth 2020, eqs. (6)–(9)] comme $\boldsymbol{\Phi}^{(\mathrm{fv})} := \mathrm{diag}(C_{\mathrm{d}}, C_y, C_\ell)$ et

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

les valeurs numériques des constantes figurant dans le tableau 7.1 (ces valeurs numériques n'ont pas été indiquées dans [Lustosa 2019] et [Olszanecki Barth 2020] et sont données ici pour permettre de reproduire les résultats de nos simulations).

3.1.2 Modèle non linéaire simplifiée à basse vitesse

Comme nous allons nous intéresser au maintien du drone en stationnaire, où la vitesse du drone est faible, nous pouvons simplifier la dynamique (3.1) en négligeant les effets aérodynamique quadratique dû à la vitesse $\boldsymbol{v}_{\mathrm{b}}$ et à la vitesse angulaire $\boldsymbol{\omega}_{\mathrm{b}}$ dans (3.5) et (3.6). Nous définissons le vecteur de commande :

$$\boldsymbol{u} := \begin{bmatrix} \tau_1 & \tau_2 & \delta_1 & \delta_2 \end{bmatrix}^{\top}, \tag{3.7}$$

qui permet d'obtenir le modèle basse vitesse comportant majeur les effets non linéaires du vent

$$\dot{\boldsymbol{p}} = \boldsymbol{v}, \tag{3.8a}$$

$$m\dot{\boldsymbol{v}} = -m\boldsymbol{g} + \boldsymbol{R}(\boldsymbol{q})\Big(\boldsymbol{M}_{\mathrm{f}}(\boldsymbol{u}) + \boldsymbol{D}_{\mathrm{f}}(\boldsymbol{u})\|\boldsymbol{w}\|\boldsymbol{R}^{\top}(\boldsymbol{q})(\boldsymbol{v} - \boldsymbol{w})\Big), \tag{3.8b}$$

$$\dot{\boldsymbol{q}} = \frac{1}{2}\boldsymbol{q} \otimes \left[\begin{smallmatrix} 0 \\ \boldsymbol{\omega}_{\mathrm{b}} \end{smallmatrix}\right], \tag{3.8c}$$

$$\boldsymbol{J}\dot{\boldsymbol{\omega}}_{\mathrm{b}} = -[\boldsymbol{\omega}_{\mathrm{b}}]_\times \boldsymbol{J}\boldsymbol{\omega}_{\mathrm{b}} + \boldsymbol{M}_{\mathrm{m}}(\boldsymbol{u}) + \boldsymbol{D}_{\mathrm{m}}(\boldsymbol{u})\|\boldsymbol{w}\|\boldsymbol{R}^{\top}(\boldsymbol{q})(\boldsymbol{v} - \boldsymbol{w}), \tag{3.8d}$$

où les vecteurs $\boldsymbol{M}_{\mathrm{f}}(\boldsymbol{u})$ et $\boldsymbol{M}_{\mathrm{m}}(\boldsymbol{u})$, et les matrices $\boldsymbol{D}_{\mathrm{f}}(\boldsymbol{u})$ et $\boldsymbol{D}_{\mathrm{m}}(\boldsymbol{u})$ proviennent de l'annulation des termes dépendant de la vitesse angulaire dans l'équation (3.5) et

(3.6). Ils peuvent etre développer en

$$
\begin{aligned}
\boldsymbol{M}_{\mathrm{f}}(\boldsymbol{u}) &:= \boldsymbol{T}_1+\boldsymbol{T}_2+\frac{S_{\mathrm{wet}}}{4S_{\mathrm{p}}}\boldsymbol{\Phi}^{(\mathrm{fv})}\Big((\boldsymbol{\Delta}_1^{\mathrm{f}} - \mathbb{I}_3)\boldsymbol{T}_1 + (\boldsymbol{\Delta}_2^{\mathrm{f}} - \mathbb{I}_3)\boldsymbol{T}_2\Big) \\
&= \begin{bmatrix} \left(1-\frac{S_{\mathrm{wet}}}{4S_{\mathrm{p}}}C_{\mathrm{d}}\right)(\tau_1+\tau_2) \\ 0 \\ -\frac{S_{\mathrm{wet}}}{4S_{\mathrm{p}}}C_\ell\xi_{\mathrm{f}}\left(\delta_1\tau_1+\delta_2\tau_2\right) \end{bmatrix} &(3.9) \\
\boldsymbol{M}_{\mathrm{m}}(\boldsymbol{u}) &:= \boldsymbol{N}_1+\boldsymbol{N}_2+\begin{bmatrix} p_x \\ p_y \\ 0\end{bmatrix}_\times \boldsymbol{T}_1 + \begin{bmatrix} p_x \\ -p_y \\ 0\end{bmatrix}_\times \boldsymbol{T}_2 \\
&\quad - \frac{S_{\mathrm{wet}}}{4S_{\mathrm{p}}}\Big(\boldsymbol{B}\boldsymbol{\Phi}^{(\mathrm{mv})}(\boldsymbol{\Delta}_1^{\mathrm{m}}-\mathbb{I}_3)+\begin{bmatrix} 0 \\ a_y \\ 0\end{bmatrix}_\times \boldsymbol{\Phi}^{(\mathrm{fv})}(\mathbb{I}_3+\boldsymbol{\Delta}_1^{\mathrm{m}})\Big)\boldsymbol{T}_1 \\
&\quad - \frac{S_{\mathrm{wet}}}{4S_{\mathrm{p}}}\Big(\boldsymbol{B}\boldsymbol{\Phi}^{(\mathrm{mv})}(\boldsymbol{\Delta}_2^{\mathrm{m}}-\mathbb{I}_3)+\begin{bmatrix} 0 \\ -a_y \\ 0\end{bmatrix}_\times \boldsymbol{\Phi}^{(\mathrm{fv})}(\mathbb{I}_3+\boldsymbol{\Delta}_2^{\mathrm{m}})\Big)\boldsymbol{T}_2 \\
&= \begin{bmatrix} \frac{k_{\mathrm{m}}}{k_{\mathrm{f}}}(\tau_1-\tau_2)+\frac{S_{\mathrm{wet}}}{4S_{\mathrm{p}}}a_yC_\ell\xi_{\mathrm{f}}(\delta_1\tau_1-\delta_2\tau_2) \\ \frac{S_{\mathrm{wet}}}{4S_{\mathrm{p}}}\Delta_{\mathrm{r}}C_\ell\xi_{\mathrm{m}}(\delta_1\tau_1+\delta_2\tau_2) \\ \left(p_y+\frac{S_{\mathrm{wet}}}{4S_{\mathrm{p}}}a_yC_{\mathrm{d}}\right)(\tau_1-\tau_2) \end{bmatrix} &(3.10) \\
\boldsymbol{D}_{\mathrm{f}}(\boldsymbol{u}) &:= \frac{1}{4}\rho S\boldsymbol{\Phi}^{(\mathrm{fv})}\Big(\boldsymbol{\Delta}_1^{\mathrm{f}}+\boldsymbol{\Delta}_2^{\mathrm{f}}-2\mathbb{I}_3\Big) &(3.11) \\
&= \frac{1}{4}\rho S\begin{bmatrix} -2C_{\mathrm{d}} & 0 & C_{\mathrm{d}}\xi_{\mathrm{f}}(\delta_1+\delta_2) \\ 0 & 0 & 0 \\ -C_\ell\xi_{\mathrm{f}}(\delta_1+\delta_2) & 0 & -2C_\ell \end{bmatrix} \\
\boldsymbol{D}_{\mathrm{m}}(\boldsymbol{u}) &:= \frac{1}{4}\rho S\Big(\Big(\begin{bmatrix} 0 \\ a_y \\ 0\end{bmatrix}_\times \boldsymbol{\Phi}^{(\mathrm{fv})}+\boldsymbol{B}\boldsymbol{\Phi}^{(\mathrm{mv})}\Big)\boldsymbol{\Delta}_1^{\mathrm{m}} &(3.12) \\
&\quad + \Big(\begin{bmatrix} 0 \\ -a_y \\ 0\end{bmatrix}_\times \boldsymbol{\Phi}^{(\mathrm{fv})}+\boldsymbol{B}\boldsymbol{\Phi}^{(\mathrm{mv})}\Big)\boldsymbol{\Delta}_2^{\mathrm{m}}-2\boldsymbol{B}\boldsymbol{\Phi}^{(\mathrm{mv})}\Big) \\
&= \frac{1}{4}\rho S\begin{bmatrix} -a_yC_{\mathrm{d}}\xi_{\mathrm{m}}(\delta_1-\delta_2) & 0 & 0 \\ \Delta_{\mathrm{r}}C_\ell\xi_{\mathrm{m}}(\delta_1+\delta_2) & 0 & 2\Delta_{\mathrm{r}}C_\ell \\ 0 & 0 & -a_yC_\ell\xi_{\mathrm{m}}(\delta_1-\delta_2) \end{bmatrix}
\end{aligned}
$$

où l'on observe l'effet non linéaire d'un vent non nul, qui est non linéaire avec $\boldsymbol{q}$, $\|\boldsymbol{v}_{\mathrm{b}}\|$ et $\boldsymbol{w}$. Comme dans [Olszanecki Barth 2020, eqn. (10)] et selon la formule de Diederich, nous obtenons $C_\ell = C_{\mathrm{d}} + \frac{\pi AR}{1+\sqrt{1+\left(\frac{AR}{2}\right)^2}}$ 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 matrices $\boldsymbol{M}_{\mathrm{f}}(\boldsymbol{u})$ et $\boldsymbol{M}_{\mathrm{m}}(\boldsymbol{u})$.

3.2 Identification des paramètres du modèle

Les valeurs numériques du tableau 7.1 ont été obtenues par une campagne d'identification du modèle [Sansou 2022]. En particulier, le coefficient k_{f} a été identifié à partir de l'équation (3.3), 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. Les éléments diagonaux de l'inertie $\boldsymbol{J}$ ont été mesurés à l'aide d'un système 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_{textb}, z_{b}) du drone suspendu par deux fils, ce qui forme un pendule de torsion comme le montre la Fig. 3.4. Il est intéressant de noter que la surface soufflée par les hélices représente 67 % de la surface totale du drone.

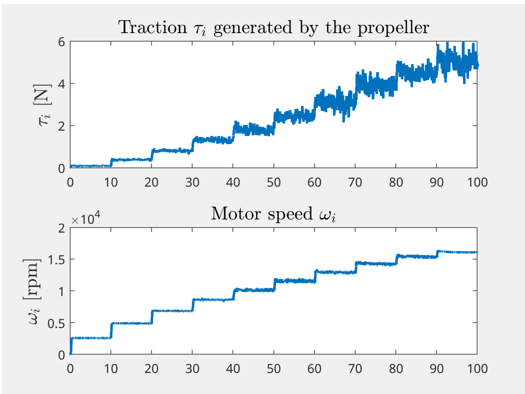

FIGURE 3.3 – Input-output response of an Esc-Motor-Propeller assembly.

uter explication
l'indentification

3.2.1 Modélisation des actionneurs

Les actionneurs de DakO ont des dynamqies qui limite leurs actions en terme 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 lié à la vitesse minimale de commutation de bobine du moteur pour maintenir la rotation. De plus, ces saturations permettent d'obtenir une modèle réaliste à énergie finie. Elle correspond à la contrainte suivante $\omega_i \in [2500,\ 16000]\ rpm = [262,\ 1675]\ \mathrm{rad\,s^{-1}}$, $i = 1, 2$. 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 $0.0125\,\mathrm{s}$, ce qui fournit un système d'actionnement assez

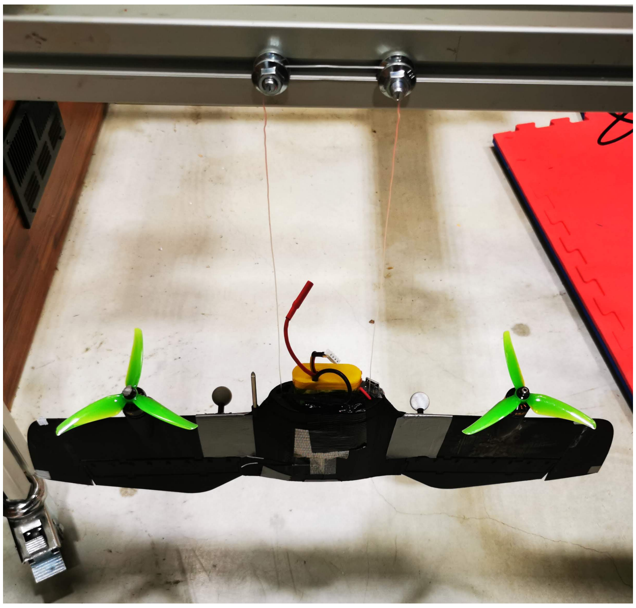

FIGURE 3.4 – Bifilar pendulum mounting for the identication of $\boldsymbol{J}$.

agressif.

Les saturations impactant les élevons proviennent des limites physiques des servomoteurs et du débattement limité par la forme de l'UAV, $\delta_i \in [-30\,;30]°$, $i = 1, 2$. La saturation la plus importante ici est peut-être 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.

3.3 Équilibres stationnaires

3.3.1 Équilibre stationnaire sans vent

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

$$\boldsymbol{u}_{\text{nowind}} := \begin{bmatrix} \tau_1 & \tau_2 & \delta_1\tau_1 & \delta_2\tau_2 \end{bmatrix}^\top \tag{3.13}$$

Nous obtenons un modèle linéaire vis-à-vis de sa commande, dérivé de (3.8) en imposant $\boldsymbol{w} = 0$,

$$\dot{\boldsymbol{p}} = \boldsymbol{v}, \qquad m\dot{\boldsymbol{v}} = -m\boldsymbol{g} + \boldsymbol{R}(\boldsymbol{q})\boldsymbol{F}\boldsymbol{u}_{\text{nowind}}, \tag{3.14a}$$

$$\dot{\boldsymbol{q}} = \frac{1}{2}\boldsymbol{q} \otimes \begin{bmatrix} 0 \\ \omega_{\text{b}} \end{bmatrix} \qquad \boldsymbol{J}\dot{\boldsymbol{\omega}}_{\text{b}} = -\,[\boldsymbol{\omega}_{\text{b}}]_\times J\boldsymbol{\omega}_{\text{b}} + \boldsymbol{M}\boldsymbol{u}_{\text{nowind}}, \tag{3.14b}$$

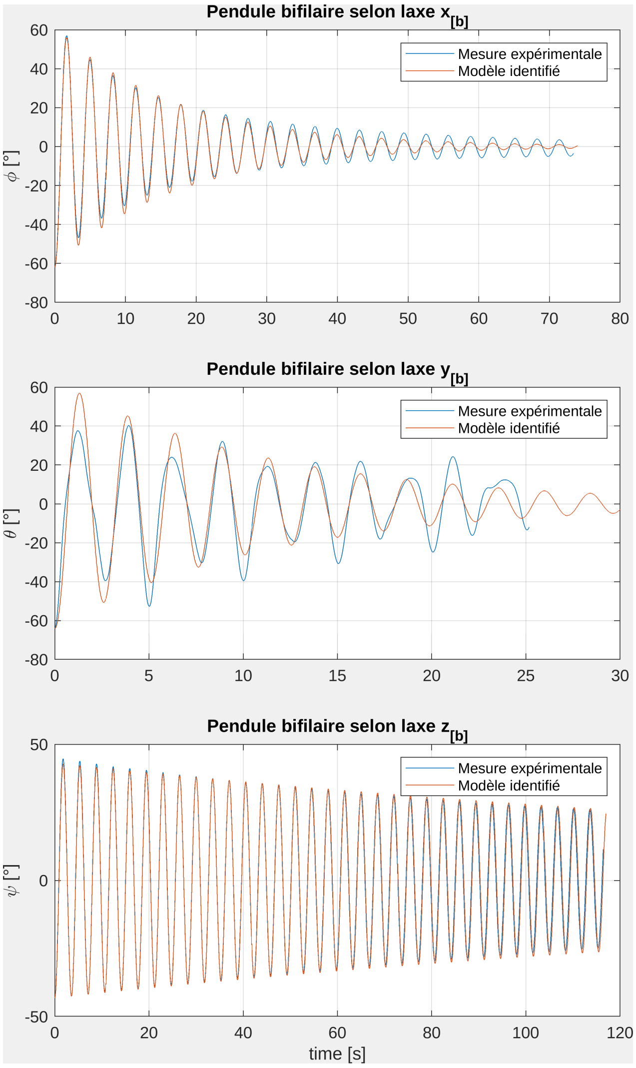

FIGURE 3.5 – Bifilar pendulum identication of $\boldsymbol{J}$.

avec les matrices

$$\left[\boldsymbol{F} \middle| \boldsymbol{M}\right] := \left[\begin{array}{cccc|cccc} a_{\mathrm{f}} & a_{\mathrm{f}} & 0 & 0 & a_{\mathrm{m}} & -a_{\mathrm{m}} & b_{\mathrm{m}} & -b_{\mathrm{m}} \\ 0 & 0 & 0 & 0 & 0 & 0 & c_{\mathrm{m}} & c_{\mathrm{m}} \\ 0 & 0 & b_{\mathrm{f}} & b_{\mathrm{f}} & d_{\mathrm{m}} & -d_{\mathrm{m}} & 0 & 0 \end{array}\right]$$

et les scalaires

$$\left[\begin{array}{c|c} a_{\mathrm{f}} & b_{\mathrm{f}} \\ \hline a_{\mathrm{m}} & b_{\mathrm{m}} \\ \hline c_{\mathrm{m}} & d_{\mathrm{m}} \end{array}\right] = \left[\begin{array}{c|c} 1 - \frac{S_{\mathrm{wet}}}{4S_{\mathrm{p}}} C_{\mathrm{d}} & -\frac{S_{\mathrm{wet}}}{4S_{\mathrm{p}}} C_{\ell} \xi_{\mathrm{f}} \\ \hline \frac{k_{\mathrm{m}}}{k_{\mathrm{f}}} & \frac{S_{\mathrm{wet}}}{4S_{\mathrm{p}}} a_y C_{\ell} \xi_{\mathrm{f}} \\ \hline \frac{S_{\mathrm{wet}}}{4S_{\mathrm{p}}} \Delta_{\mathrm{r}} C_{\ell} \xi_{\mathrm{m}} & p_y + \frac{S_{\mathrm{wet}}}{4S_{\mathrm{p}}} a_y C_{\mathrm{d}} \end{array}\right].$$

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

$$\boldsymbol{u}_{\text{nowind,eq}} = \frac{mg}{(1 - \frac{S_{\text{wet}}}{4S_{\text{p}}} C_{\text{d}})} [1\ 1\ 0\ 0]^\top \tag{3.15a}$$

$$\boldsymbol{q}_{\text{eq}} = [\eta_{\text{eq}}\ \boldsymbol{\epsilon}_{\text{eq}}^\top]^\top = \left[\sqrt{\tfrac{1}{2} - \beta}\ \ \beta\ \ \frac{2\beta^2 - 1}{2\sqrt{\frac{1}{2} - \beta}}\ \ \beta\right]^\top. \tag{3.15b}$$

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

3.3.2 Équilibre stationnaire en présence de vent

À partir des modèles (3.1) et (3.8), nous caractérisons un équilibre stationnaire en présence d'un vent constant $\boldsymbol{w}_{\text{eq}} = \left[\begin{smallmatrix} w_x \\ w_y \\ w_z \end{smallmatrix}\right] \in \mathbb{R}^3$ exprimé dans le repère inertiel, tel que $\left[\begin{smallmatrix} w_x \\ w_y \end{smallmatrix}\right] \neq 0$, c'est-à-dire qu'il existe toujours un vent horizontal non nul. Ainsi, pour chaque position de référence $\boldsymbol{p}_{\text{eq}} \in \mathbb{R}^3$, un ensemble de couple état/commande possible est $(\boldsymbol{u}_{\text{eq}}, \boldsymbol{x}_{\text{eq}}) = (\boldsymbol{u}_{\text{eq}}, \boldsymbol{p}_{\text{eq}}, \boldsymbol{v}_{\text{eq}}, \boldsymbol{q}_{\text{eq}}, \boldsymbol{\omega}_{\text{b,eq}})$ obtenu à l'aide de

$$\boldsymbol{u}_{\text{eq}} = \begin{bmatrix} \tau & \tau & \delta & \delta \end{bmatrix}^\top \tag{3.16a}$$

$$\boldsymbol{q}_{\text{eq}} = \boldsymbol{q}_{\text{eq}\psi} \otimes \boldsymbol{q}_{\text{eq}\theta} \tag{3.16b}$$

$$\boldsymbol{\omega}_{\text{b,eq}} = 0, \quad \boldsymbol{v}_{\text{eq}} = 0, \tag{3.16c}$$

où

$$\boldsymbol{q}_{\text{eq}\theta} := \begin{bmatrix} \cos(\frac{\theta}{2}) & 0 & \sin(\frac{\theta}{2}) & 0 \end{bmatrix}^\top \tag{3.17}$$

we define the quaternion $\boldsymbol{q}_{\text{eq}\psi}$ associated with a horizontal rotation $\psi = \arctan(w_x, w_y)$ of the inertial reference frame towards the (nonzero) horizontal wind direction :

traduire et amelio
l'explication

$$\boldsymbol{q}_{\text{eq}\psi} := \begin{bmatrix} \cos(\frac{\psi}{2}) & 0 & 0 & \sin(\frac{\psi}{2}) \end{bmatrix}^\top. \tag{3.18}$$

et l'angle d'inclinaison θ, la poussée des hélices τ, et la déflexion des élevons δ peuvent être obtenu à partir de l'algorithme 1.

Théorème 1. *Pour tout vent constant, $\boldsymbol{w} = \begin{bmatrix} w_x & w_y & w_z \end{bmatrix}^\top \in \mathbb{R}^3$ ayant une composante horizontale non nulle $\left[\begin{smallmatrix} w_x \\ w_y \end{smallmatrix}\right]$, les equations (3.18)–(3.17) avec θ, τ et δ sélectionné selon l'Algorithme 1 caractérisent un couple d'équilibre $(\boldsymbol{u}_{eq}, \boldsymbol{x}_{eq})$ pour la*

Algorithme 1 Obtention des paramètres d'équilibre en (3.16).

Entrée : Vecteur vent $\boldsymbol{w}_{\mathrm{eq}} = \left[\begin{smallmatrix} w_x & w_y & w_z \end{smallmatrix}\right]^\top$

Sortie : Paramètres ψ, θ, τ, δ dans (3.16)

1: Détermine l'angle $\psi = \operatorname{atan2}(w_x, w_y)$ de manière à obtenir $\boldsymbol{q}_{\mathrm{eq}\psi}$ dans (3.18)

2: Détermine la perturbation tournée $\boldsymbol{w}_{\mathrm{r}}$ avec la composante y nulle, en utilisant $\boldsymbol{R}_\psi := \left[\begin{smallmatrix} \cos\psi & \sin\psi & 0 \\ -\sin\psi & \cos\psi & 0 \\ 0 & 0 & 1 \end{smallmatrix}\right]$, selon

$$\boldsymbol{w}_{\mathrm{r,eq}} := \left[\begin{smallmatrix} w_{\mathrm{r}x} \\ 0 \\ w_{\mathrm{r}z} \end{smallmatrix}\right] := \boldsymbol{R}^\top(\boldsymbol{q}_{\mathrm{eq}\psi})\boldsymbol{w}_{\mathrm{eq}} = \boldsymbol{R}_\psi^\top \boldsymbol{w}_{\mathrm{eq}} \tag{3.19}$$

3: Détermine l'angle d'inclinaison θ de manière à obtenir $\boldsymbol{q}_{\mathrm{eq}\theta}$ dans (3.16b) :

$$\theta = -\tan^{-1}\left(\frac{w_{\mathrm{r}z}}{w_{\mathrm{r}x}} + \frac{2mg}{\rho S\|\boldsymbol{w}_{\mathrm{eq}}\|C_\ell(1-\frac{\xi_{\mathrm{f}}}{\xi_{\mathrm{m}}})w_{\mathrm{r}x}}\right) \tag{3.20}$$

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

$$\left[\begin{array}{c|c} a & b \\ \hline c & d \end{array}\right] := \left[\begin{array}{c|c} 2S_{\mathrm{wet}}C_\ell mg\sin\theta\xi_{\mathrm{f}} & 2S_{\mathrm{wet}}C_{\mathrm{d}}C_\ell\rho\|\boldsymbol{w}_{\mathrm{eq}}\|w_x^{\mathrm{b}} \\ \hline -4SS_{\mathrm{p}}C_\ell\rho\|\boldsymbol{w}_{\mathrm{eq}}\|w_x^{\mathrm{b}}\xi_{\mathrm{f}} & \frac{b\xi_{\mathrm{f}}}{2} \end{array}\right]$$

et grâce à ces scalaires (a, b, c, d), déterminons la traction des hélices τ dans (3.16a) comme

$$\begin{aligned}\tau = \frac{S_{\mathrm{p}}}{2S_{\mathrm{wet}}C_\ell\xi_{\mathrm{f}}(4S_{\mathrm{p}} - S_{\mathrm{wet}}C_{\mathrm{d}})}\Bigg(a+b+c+d+\Bigg[&(a+b+c-d)^2 - 4(d^2+ac-bd) \\ &- \frac{4{w_z^{\mathrm{b}}}^2 d}{{w_x^{\mathrm{b}}}^2}(d+c) + \frac{4w_z^{\mathrm{b}}ad\cos\theta}{w_x^{\mathrm{b}}C_\ell\sin\theta}\left(C_{\mathrm{d}} - \frac{4S_{\mathrm{p}}}{S_{\mathrm{wet}}}\right)\Bigg]^{\frac{1}{2}}\Bigg),\end{aligned} \tag{3.21}$$

où

$$\begin{bmatrix} w_x^{\mathrm{b}} \\ w_z^{\mathrm{b}} \end{bmatrix} = \begin{bmatrix} w_{\mathrm{r}x}\cos\theta - w_{\mathrm{r}z}\sin\theta \\ w_{\mathrm{r}x}\sin\theta + w_{\mathrm{r}z}\cos\theta \end{bmatrix}.$$

5: Déterminons la déflexion des élevons δ comme

$$\delta = \frac{2mg\sin\theta}{\rho S\|\boldsymbol{w}_{\mathrm{eq}}\|C_{\mathrm{d}}\xi_{\mathrm{f}}w_z^{\mathrm{b}}} + \frac{w_x^{\mathrm{b}}}{\xi_{\mathrm{f}}w_z^{\mathrm{b}}} - \frac{(4 - \frac{S_{\mathrm{wet}}}{S_{\mathrm{p}}}C_{\mathrm{d}})}{\rho S\|\boldsymbol{w}_{\mathrm{eq}}\|C_{\mathrm{d}}\xi_{\mathrm{f}}w_z^{\mathrm{b}}}\tau. \tag{3.22}$$

Retourne : ψ, θ, τ, δ

dynamique non linéaire (3.1) *et* (3.8).

Démonstration. Dans un premier temps, notons qu'avec l'expression de $\boldsymbol{R}$ (3.2) et l'expression de ψ dans l'étape 1 de l'Algorithme 1, on peut définir la perturbation à l'équilibre tournée $\boldsymbol{w}_{\mathrm{r,eq}} := \boldsymbol{R}_{\psi}^{\top}\boldsymbol{w}_{\mathrm{eq}} := \boldsymbol{R}^{\top}(\boldsymbol{q}_{\mathrm{eq}\psi})\boldsymbol{w}_{\mathrm{eq}}$ (voir (3.19) dans l'Algorithme 1), qui correpond à la rotation necessaire pour aligner l'axe $x_{[\mathrm{b}]}$ 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 il 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 repere corps comme étant

$$\begin{aligned}\boldsymbol{w}_{\mathrm{eq}}^{\mathrm{b}} &:= \begin{bmatrix} w_x^{\mathrm{b}} \\ 0 \\ w_z^{\mathrm{b}} \end{bmatrix} = \boldsymbol{R}^{\top}(\boldsymbol{q}_{\mathrm{eq}\theta})\boldsymbol{w}_{\mathrm{r,eq}} \\ &= \begin{bmatrix} \cos\theta & 0 & -\sin\theta \\ 0 & 1 & 0 \\ \sin\theta & 0 & \cos\theta \end{bmatrix}^{\top} \begin{bmatrix} w_{\mathrm{r}x} \\ 0 \\ w_{\mathrm{r}z} \end{bmatrix} = \begin{bmatrix} w_{\mathrm{r}x}\cos\theta - w_{\mathrm{r}z}\sin\theta \\ 0 \\ w_{\mathrm{r}x}\sin\theta + w_{\mathrm{r}z}\cos\theta \end{bmatrix}\end{aligned} \tag{3.23}$$

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 engendré par $\boldsymbol{q}_{\mathrm{eq}\psi}$, et suite à l'hypothèse $\left[\begin{smallmatrix} w_x \\ w_y \end{smallmatrix}\right] \neq 0$.

L'équation (3.8a) montre qu'il est nécessaire d'avoir $\boldsymbol{v}_{\mathrm{eq}} = 0$ pour maintenir l'équilibre stationnaire. En multipliant (3.8b) par $\boldsymbol{R}(\boldsymbol{q}_{\mathrm{eq}})$ donné dans (3.23), 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 au deux élevons $\delta_1 = \delta_2 = \delta$, nous obtenons pour les deux modèles (3.1) et (3.8), l'équilibre des forces selon l'axe $x_{[\mathrm{b}]}$ donné par

$$(2 - \frac{S_{\mathrm{wet}}}{2S_{\mathrm{p}}}C_{\mathrm{d}})\tau - \frac{1}{2}\rho S\|\boldsymbol{w}_{\mathrm{eq}}\|C_{\mathrm{d}}\left(w_x^{\mathrm{b}} - \xi_{\mathrm{f}}\delta w_z^{\mathrm{b}}\right) - mg\sin(\theta) = 0 \tag{3.24}$$

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

$$-\frac{S_{\mathrm{wet}}}{2S_{\mathrm{p}}}\xi_{\mathrm{f}}C_{\ell}\tau\delta - \frac{1}{2}\rho S\|\boldsymbol{w}_{\mathrm{eq}}\|C_{\ell}\left(w_z^{\mathrm{b}} + \xi_{\mathrm{f}}\delta w_x^{\mathrm{b}}\right) + mg\cos(\theta) = 0 \tag{3.25}$$

De manière similaire, à partir de (3.1b) et (3.8d), l'équilibre des moments autour de l'axe $y_{[\mathrm{b}]}$ permet d'obtenir

$$\frac{S_{\mathrm{wet}}}{2S_{\mathrm{p}}}\Delta_{\mathrm{r}}\xi_{\mathrm{m}}C_{\ell}\tau\delta + \frac{1}{2}\rho S\Delta_{\mathrm{r}}\|\boldsymbol{w}_{\mathrm{eq}}\|C_{\ell}\left(w_z^{\mathrm{b}} + \xi_{\mathrm{m}}\delta w_x^{\mathrm{b}}\right) = 0. \tag{3.26}$$

Pour calculer la solution du triplet (θ,τ,δ) des trois équations d'équilibre (3.24)–(3.26), ajoutons (3.25) multiplié par $\Delta_{\mathrm{r}}\xi_{\mathrm{m}}$, à (3.26) multiplié par ξ_{f}, de manière à annuler le premier terme et à obtenir

$$\begin{aligned}\Delta_{\mathrm{r}}\xi_{\mathrm{m}}\left(-\frac{1}{2}\rho S\|\boldsymbol{w}_{\mathrm{eq}}\|C_{\ell}(w_z^{\mathrm{b}} + \xi_{\mathrm{f}}\delta w_x^{\mathrm{b}}) + mg\cos(\theta)\right)&\\ + \xi_{\mathrm{f}}\left(\frac{1}{2}\rho S\Delta_{\mathrm{r}}\|\boldsymbol{w}_{\mathrm{eq}}\|C_{\ell}(w_z^{\mathrm{b}} + \xi_{\mathrm{m}}\delta w_x^{\mathrm{b}})\right) &= 0,\end{aligned}$$

qui est équivalent à

$$\frac{1}{2}\rho S\Delta_{\mathrm{r}}\|\boldsymbol{w}_{\mathrm{eq}}\|C_{\ell}(\xi_{\mathrm{f}} - \xi_{\mathrm{m}})w_z^{\mathrm{b}} + \Delta_{\mathrm{r}}\xi_{\mathrm{m}}mg\cos(\theta) = 0,$$

où $(w_x^{\mathrm{b}},w_z^{\mathrm{b}})$ sont les première et troisième composantes de $\boldsymbol{w}^{\mathrm{b}}$ dans (3.23). Ensuite, en utilisant (3.23) et en réarrangeant, nous obtenons

$$\begin{aligned}-\frac{1}{2}\rho S\Delta_{\mathrm{r}}\|\boldsymbol{w}_{\mathrm{eq}}\|C_{\ell}(\xi_{\mathrm{f}} - \xi_{\mathrm{m}})w_{\mathrm{r}x}\sin\theta + \Bigg(-\frac{1}{2}\rho S\Delta_{\mathrm{r}}&\\ \|\boldsymbol{w}_{\mathrm{eq}}\|C_{\ell}(\xi_{\mathrm{f}} - \xi_{\mathrm{m}})w_{\mathrm{r}z} + \Delta_{\mathrm{r}}\xi_{\mathrm{m}}mg\Bigg)\cos\theta &= 0,\end{aligned}$$

qui est satisfaite par

$$\theta = -\tan^{-1}\left(\frac{\rho S\|\boldsymbol{w}_{\mathrm{eq}}\|C_{\ell}(\xi_{\mathrm{f}} - \xi_{\mathrm{m}})w_{\mathrm{r}z} - 2\xi_{\mathrm{m}}mg}{\rho S\|\boldsymbol{w}_{\mathrm{eq}}\|C_{\ell}(\xi_{\mathrm{f}} - \xi_{\mathrm{m}})w_{\mathrm{r}x}}\right). \tag{3.27}$$

Cette dernière expression coïncide avec la sélection (3.20) dans l'Algorithme 1 après quelques manipulations. À partir de (3.20), nous pouvons calculer les commandes à l'équilibre en substituant (3.24) dans (3.25). Après quelques simplifications, la force nécessaire de traction des hélices τ pour maintenir la position d'équilibre corresponds à l'expression (3.21). Finalement, avec la valeur de τ dans (3.21), nous pouvons obtenir la déflexion des élevons nécessaire δ à partir de l'équation (3.25), ce qui nous donne la valeur obtenue dans (3.22). □

Il est intéressant de noter que pour chaque couple de vent$(w_{\mathrm{r}z},\, w_{\mathrm{r}x})$ correspond une orientation d'équilibre (3.16b), (3.20) est indépendante de l'entrée $\boldsymbol{u}_{\mathrm{eq}}$. En outre, il convient de souligner que pour toutes les valeurs de vent raisonnables, l'équation (3.21) 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 (3.16) de l'équilibre du drone pour différentes conditions de vent $\boldsymbol{w}$, nous reportons, sur la Fig. 3.6, les valeurs correspondantes de θ, δ, τ pour des valeurs de vitesse de vent horizontale allant de 0 à $-20\,\mathrm{m\,s^{-1}}$ et pour des valeurs de vitesse de vent verticale allant de -6 à $6\,\mathrm{m\,s^{-1}}$. L'angle d'incidence θ diminue de 90° à $-4.65°$. $\theta = 90°$ correspond à un vol stationnaire sans vent. La traction τ atteint son minimum à $w_{rx} = -12.8\,\mathrm{m\,s^{-1}}$, ce qui correspond à une condition de vol qui minimise la consommation d'énergie, car les moteurs sont la principale source de consommation électrique.

traduction captio

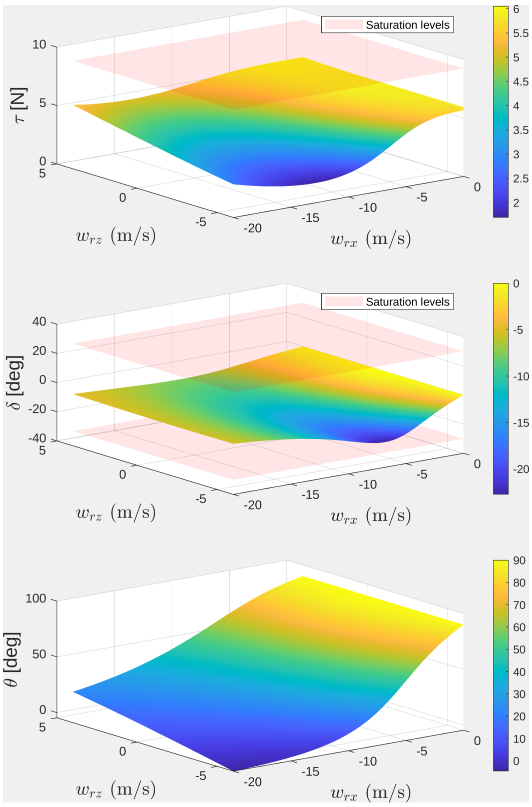

FIGURE 3.6 – Parameters (θ, δ, τ) of the equilibrium point (surface) established in Theorem 1 and Algorithm 1 for constant horizontal and vertical wind (w_{rx},w_{rz}), and actuators saturation levels (red).

Il est possible de faire une coupe des surfaces présentée dans (3.6) pour une vitesse verticale nulle $\boldsymbol{w}_{\mathrm{rx}} = 0$, ce qui nous donne le résultat de la Figure 3.7

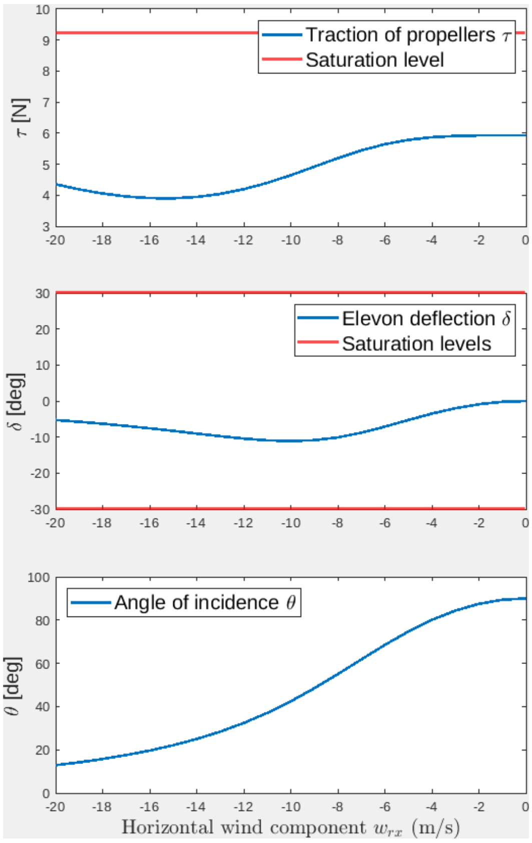

FIGURE 3.7 – Parameters (θ, δ, τ) of the equilibrium point (blue) established in Theorem 1 and Algorithm 1 for a constant horizontal wind $w_{\text{r}x}$, and actuators saturation levels (red).

3.4 Dynamiques linéarisés

3.4.1 Dynamique linéarisé sans vent

Considérons le cas sans vent discuté dans la section 3.3.1 pour lequel nous utilisons le vecteur de commande $\boldsymbol{u}_{\text{nowind}}$ et le vecteur de commande à l'équilibre $\boldsymbol{u}_{\text{nowind,eq}}$ défini dans l'équation (3.15a) et rapellons la transformation du vecteur de comamande suivante $\boldsymbol{u}_{\text{nowind}} := \begin{bmatrix} \tau_1 & \tau_2 & \delta_1\tau_1 & \delta_2\tau_2 \end{bmatrix}^\top$, la dynamqiue linearisé dans le cas sans vent est

$$\dot{\tilde{\boldsymbol{x}}} = \boldsymbol{A}_0\tilde{\boldsymbol{x}} + \boldsymbol{G}_0(\boldsymbol{u}_{\text{nowind}} - \boldsymbol{u}_{\text{nowind,eq}}), \tag{3.28}$$

où l'expression de $\boldsymbol{A}_0$ est

$$\boldsymbol{A}_0 = \boldsymbol{A}_w\Big|_{\boldsymbol{w}=0} = \begin{bmatrix} \mathbb{0}_3 & \mathbb{I}_3 & \mathbb{0}_3 & \mathbb{0}_3 \\ \mathbb{0}_3 & \mathbb{0}_3 & \boldsymbol{A}_{v\epsilon} & \mathbb{0}_3 \\ \mathbb{0}_3 & \mathbb{0}_3 & \mathbb{0}_3 & \boldsymbol{A}_{\epsilon\omega} \\ \mathbb{0}_3 & \mathbb{0}_3 & \mathbb{0}_3 & \mathbb{0}_3 \end{bmatrix}, \tag{3.29}$$

avec les matrices suivante

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

alors que l'expression de $\boldsymbol{G}_0$ est

$$\boldsymbol{G}_0 := \begin{bmatrix} \mathbb{0}_{3\times1} & \mathbb{0}_{3\times1} & \mathbb{0}_{3\times1} & \mathbb{0}_{3\times1} \\ 0 & 0 & a_\mathrm{g} & a_\mathrm{g} \\ 0 & 0 & 0 & 0 \\ b_\mathrm{g} & b_\mathrm{g} & 0 & 0 \\ \mathbb{0}_{3\times1} & \mathbb{0}_{3\times1} & \mathbb{0}_{3\times1} & \mathbb{0}_{3\times1} \\ c_\mathrm{g} & -c_\mathrm{g} & d_\mathrm{g} & -d_\mathrm{g} \\ 0 & 0 & e_\mathrm{g} & e_\mathrm{g} \\ f_\mathrm{g} & -f_\mathrm{g} & 0 & 0 \end{bmatrix},$$

avec

$$\left[\begin{array}{c|c} a_\mathrm{g} & b_\mathrm{g} \\ \hline c_\mathrm{g} & d_\mathrm{g} \\ \hline e_\mathrm{g} & f_\mathrm{g} \end{array}\right] = \left[\begin{array}{c|c} -\frac{S_\mathrm{wet}}{4mS_\mathrm{p}}C_\ell\xi_\mathrm{f} & \frac{1}{m}(1-\frac{S_\mathrm{wet}}{2S_\mathrm{p}}C_\mathrm{d}) \\ \hline \frac{k_\mathrm{m}}{J_xk_\mathrm{f}} & \frac{S_\mathrm{wet}a_y}{4J_xS_\mathrm{p}}C_\ell\xi_\mathrm{f} \\ \hline \frac{S_\mathrm{wet}\Delta_\mathrm{r}}{4J_yS_\mathrm{p}}C_\ell\xi_\mathrm{m} & \frac{1}{J_z}(p_y+\frac{S_\mathrm{wet}}{4S_\mathrm{p}}a_yC_\mathrm{d}) \end{array}\right].$$

3.4.2 Dynamique linéarisé 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 (3.8). 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. Plus précisément, pour chaque condition de vent d'équilibre $\boldsymbol{w}_\mathrm{eq}$ associé à l'équilibre $(\boldsymbol{u}_\mathrm{eq}, \boldsymbol{p}_\mathrm{eq}, \boldsymbol{v}_\mathrm{eq}, \boldsymbol{q}_\mathrm{eq}, \boldsymbol{\omega}_\mathrm{b,eq})$ caractérisé en (3.18)–(3.17), désignant les composantes scalaire et vectorielle du quaternion en (3.16b) comme $\boldsymbol{q}_\mathrm{eq} = (\eta_\mathrm{eq}, \boldsymbol{\epsilon}_\mathrm{eq})$, et à partir de la matrice de rotation $\boldsymbol{R}_\psi := \boldsymbol{R}(\boldsymbol{q}_{\text{eq}\psi})$ introduite au début de la preuve du Théorème 1, nous étudions ici la dynamique incrémental linéaire du vecteur d'état tourné :

$$\begin{aligned}\tilde{\boldsymbol{x}} &:= (\tilde{\boldsymbol{p}}, \tilde{\boldsymbol{v}}, \tilde{\boldsymbol{\epsilon}}, \tilde{\boldsymbol{\omega}}_{\text{b}}) = \left(\boldsymbol{R}_\psi^\top(\boldsymbol{p}-\boldsymbol{p}_{\text{eq}}), \boldsymbol{R}_\psi^\top \boldsymbol{v}, \boldsymbol{R}_\psi^\top(\boldsymbol{\epsilon}-\boldsymbol{\epsilon}_{\text{eq}}), \boldsymbol{\omega}_{\text{b}}\right), \\ \tilde{\boldsymbol{u}} &:= \boldsymbol{u} - \boldsymbol{u}_{\text{eq}}, \quad \tilde{\boldsymbol{w}} := \boldsymbol{R}_\psi^\top(\boldsymbol{w} - \boldsymbol{w}_{\text{eq}}).\end{aligned} \tag{3.30}$$

Notez que la rotation en (3.30) possède la propriété $\boldsymbol{R}_\psi^\top \boldsymbol{\epsilon}_{\text{eq}} = \left[\begin{smallmatrix} 0 & \sin(\frac{\theta}{2}) & 0 \end{smallmatrix}\right]^\top$, ce qui simplifie grandement le mouvement linéarisé.

En exploitant le fait que les vitesses lineaaire et angulaire ($\boldsymbol{v}_{\text{eq}}$, $\boldsymbol{\omega}_{\text{b,eq}}$) doit être nulle à l'équilibre (voir (3.16)), nous prouvons ci-dessous que la dynamique linéarisée de l'état (3.30) est donnée par

$$\begin{aligned}\dot{\tilde{\boldsymbol{x}}} &= \boldsymbol{A}_w \tilde{\boldsymbol{x}} + \boldsymbol{G}_w \tilde{\boldsymbol{u}} + \boldsymbol{E}_w \tilde{\boldsymbol{w}} \\ &= \begin{bmatrix} \mathbb{0}_3 & \mathbb{I}_3 & \mathbb{0}_3 & \mathbb{0}_3 \\ \mathbb{0}_3 & \boldsymbol{A}_{vv} & \boldsymbol{A}_{v\epsilon} & \mathbb{0}_3 \\ \mathbb{0}_3 & \mathbb{0}_3 & \mathbb{0}_3 & \boldsymbol{A}_{\epsilon\omega} \\ \mathbb{0}_3 & \mathbb{0}_3 & \boldsymbol{A}_{\omega\epsilon} & \mathbb{0}_3 \end{bmatrix} \tilde{\boldsymbol{x}} + \begin{bmatrix} \mathbb{0}_{3\times4} \\ \boldsymbol{G}_v \\ \mathbb{0}_{3\times4} \\ \boldsymbol{G}_\omega \end{bmatrix} \tilde{\boldsymbol{u}} + \begin{bmatrix} \mathbb{0}_{3\times3} \\ \boldsymbol{E}_v \\ \mathbb{0}_{3\times3} \\ \boldsymbol{E}_\omega \end{bmatrix} \tilde{\boldsymbol{w}},\end{aligned} \tag{3.31}$$

avec les matrices $\boldsymbol{A}_{vv}$, $\boldsymbol{A}_{v\epsilon}$, $\boldsymbol{A}_{\epsilon\omega_{\text{b}}}$, $\boldsymbol{A}_{\omega\epsilon}$, $\boldsymbol{G}_v$, $\boldsymbol{G}_\omega$ $\boldsymbol{E}_v$, $\boldsymbol{E}_\omega$ construite en suivant l'Algorithme 2.

Théorème 2. *Pour tout vent constant,* $\boldsymbol{w} = \left[\begin{smallmatrix} w_x & w_y & w_z \end{smallmatrix}\right]^\top \in \mathbb{R}^3$ *ayant une composante horizontale non nulle* $\left[\begin{smallmatrix} w_x \\ w_y \end{smallmatrix}\right]$*, et le doublet d'équilibre qui découle* $(\boldsymbol{u}_{eq}, \boldsymbol{x}_{eq})$ *de la dynamique* (3.8)*, tels que caractérisés dans* (3.18)-(3.17)*, la dynamique linéarisée du vecteur état incremental* (3.30) *est donné par* (3.31) *avec les matrices construites comme dans l'Algorithme 2.*

Démonstration. Tout d'abord, en exploitant la matrice de rotation $\boldsymbol{R}_\psi := \boldsymbol{R}(\boldsymbol{q}_{\text{eq}\psi})$ utilisé dans (3.30), nous transformons la dynamique non linéaire (3.8) en coordonnées tournées

$$(\boldsymbol{p}_{\text{r}}, \boldsymbol{v}_{\text{r}}, \boldsymbol{q}_{\text{r}}) := \left(\boldsymbol{R}_\psi^\top \boldsymbol{p}, \boldsymbol{R}_\psi^\top \boldsymbol{v}, \boldsymbol{q}_{\text{eq}\psi}^{-1} \otimes \boldsymbol{q}\right), \ \boldsymbol{w}_{\text{r}} := \boldsymbol{R}_\psi^\top \boldsymbol{w} \tag{3.32}$$

où $\boldsymbol{\omega}_{\text{b}}$ reste inchangée car elle est exprimée dans le repère du corps. Quelques observations permettent de simplifier la dynamique transformée (3.8) :

— nous avons $\boldsymbol{R}_\psi^\top m\boldsymbol{g} = m\boldsymbol{g}$ car la rotation de ψ est autour de l'axe $z_{[\text{i}]}$;
— comme $\boldsymbol{q}_{\text{r}} = \boldsymbol{q}_{\text{eq}\psi}^{-1} \otimes \boldsymbol{q}$, alors $\boldsymbol{R}_\psi^\top \boldsymbol{R}(\boldsymbol{q}) = \boldsymbol{R}(\boldsymbol{q}_{\text{r}})$;
— comme $\boldsymbol{v}_{\text{b}} := \boldsymbol{R}^\top(\boldsymbol{q})(\boldsymbol{v} - \boldsymbol{w})$ (comme défini après l'équation (3.6)), alors $\|\boldsymbol{v}_{\text{b}}\| = \|\boldsymbol{v} - \boldsymbol{w}\| - \|\boldsymbol{v}_{\text{r}} - \boldsymbol{w}_{\text{r}}\|$
— enfin $\boldsymbol{R}^\top(\boldsymbol{q})\boldsymbol{w} = \boldsymbol{R}^\top(\boldsymbol{q}_{\text{r}})\boldsymbol{R}_\psi^\top \boldsymbol{R}_\psi \boldsymbol{w}_{\text{r}} = \boldsymbol{R}^\top(\boldsymbol{q}_{\text{r}})\boldsymbol{w}_{\text{r}}$.

Sur la base des observations ci-dessus, nous pouvons dériver la version tournée des équations (3.8) comme étant

$$\dot{\boldsymbol{p}}_{\mathrm{r}} = \boldsymbol{v}_{\mathrm{r}}, \tag{3.33a}$$
$$m\dot{\boldsymbol{v}}_{\mathrm{r}} = \text{-}m\boldsymbol{g} + \boldsymbol{R}(\boldsymbol{q}_{\mathrm{r}})\left(\boldsymbol{M}_{\mathrm{f}}(\boldsymbol{u}) + \boldsymbol{D}_{\mathrm{f}}(\boldsymbol{u})\|\boldsymbol{w}_{\mathrm{r}}\|\boldsymbol{R}^{\top}(\boldsymbol{q}_{\mathrm{r}})(\boldsymbol{v}_{\mathrm{r}}\text{-}\boldsymbol{w}_{\mathrm{r}})\right), \tag{3.33b}$$
$$\dot{\boldsymbol{q}}_{\mathrm{r}} = \left(\frac{1}{2}\boldsymbol{q}_{\mathrm{r}} \otimes \left[\begin{smallmatrix} 0 \\ \boldsymbol{\omega}_{\mathrm{b}} \end{smallmatrix}\right]\right), \tag{3.33c}$$
$$\boldsymbol{J}\dot{\boldsymbol{\omega}}_{\mathrm{b}} = \text{-}\left[\boldsymbol{\omega}_{\mathrm{b}}\right]_{\times}\boldsymbol{J}\boldsymbol{\omega}_{\mathrm{b}} + \boldsymbol{M}_{\mathrm{m}}(\boldsymbol{u}) + \boldsymbol{D}_{\mathrm{m}}(\boldsymbol{u})\|\boldsymbol{w}_{\mathrm{r}}\|\boldsymbol{R}^{\top}(\boldsymbol{q}_{\mathrm{r}})(\boldsymbol{v}_{\mathrm{r}}\text{-}\boldsymbol{w}_{\mathrm{r}}) \tag{3.33d}$$

Avec ces nouvelles coordonnées, les vecteurs d'état incrémental (3.30) peut être exprimés comme étant

$$\begin{aligned} \tilde{\boldsymbol{x}} &= \left(\boldsymbol{p}_{\mathrm{r}}\text{-}\boldsymbol{R}_{\psi}^{\top}\boldsymbol{p}_{\mathrm{eq}}, \boldsymbol{v}_{\mathrm{r}}, \boldsymbol{\epsilon}_{\mathrm{r}}\text{-}\boldsymbol{R}_{\psi}^{\top}\boldsymbol{\epsilon}_{\mathrm{eq}}, \boldsymbol{\omega}_{\mathrm{b}}\right), \\ \tilde{\boldsymbol{u}} &:= \boldsymbol{u} - \boldsymbol{u}_{\mathrm{eq}}, \quad \tilde{\boldsymbol{w}} := \boldsymbol{w}_{\mathrm{r}} - \boldsymbol{w}_{\mathrm{r,eq}} \end{aligned} \tag{3.34}$$

où $\boldsymbol{w}_{\mathrm{r,eq}} = \boldsymbol{R}_{\psi}^{\top}\boldsymbol{w}_{\mathrm{eq}} = \left[\begin{smallmatrix} w_{\mathrm{r}x} \\ 0 \\ w_{\mathrm{r}z} \end{smallmatrix}\right]$, déjà défini dans (3.19), et $\boldsymbol{R}_{\psi}^{\top}\boldsymbol{\epsilon}_{\mathrm{eq}} = \left[\begin{smallmatrix} 0 & \sin(\frac{\theta}{2}) & 0 \end{smallmatrix}\right]^{\top}$ ont tous deux une structure peu dense intéréssante.

En se concentrant sur la dynamique tournée (3.33) et l'expression (3.34) des variables incrémentales, la preuve du théorème revient à montrer que la linéarisation de (3.33) autour de l'équilibre tourné

$$\begin{aligned} \boldsymbol{x}_{\mathrm{r,eq}} &= \left(\boldsymbol{p}_{\mathrm{r,eq}}, \boldsymbol{v}_{\mathrm{r,eq}}, \boldsymbol{\epsilon}_{\mathrm{r,eq}}, \boldsymbol{\omega}_{\mathrm{br,eq}}\right) \\ &= \left(\boldsymbol{R}_{\psi}^{\top}\boldsymbol{p}_{\mathrm{eq}}, \left[\begin{smallmatrix} 0 \\ 0 \\ 0 \end{smallmatrix}\right], \left[\begin{smallmatrix} 0 \\ \sin(\frac{\theta}{2}) \\ 0 \end{smallmatrix}\right], \left[\begin{smallmatrix} 0 \\ 0 \\ 0 \end{smallmatrix}\right]\right), \ \boldsymbol{w}_{\mathrm{r,eq}} = \left[\begin{smallmatrix} w_{\mathrm{r}x} \\ 0 \\ w_{\mathrm{r}z} \end{smallmatrix}\right] \end{aligned} \tag{3.35}$$

coïncide avec l'équation (3.31) et les expressions de l'Algorithme 2.

Dans ce but, inspirée par [Tregouet 2015, Proof of Lemma 1], pour linéariser la dynamique du quaternion $\boldsymbol{q}_{\mathrm{r}} = \begin{bmatrix} \eta_{\mathrm{r}} & \boldsymbol{\epsilon}_{\mathrm{r}}^{\top} \end{bmatrix}^{\top}$ evoluant dans $\mathbb{S}^3$, nous remplacons η_{r} par sa valeur positive lié à la norme unitaire du quaternion. Ainsi, $\eta_{\mathrm{r}} = (1 - \boldsymbol{\epsilon}_{\mathrm{r}}^{\top}\boldsymbol{\epsilon}_{\mathrm{r}})^{\frac{1}{2}}$. Concentrons-nous d'abord sur la matrice $\boldsymbol{A}_w$ dans (3.31). Les trois premieres lignes sont simplement $\left[\begin{smallmatrix} \mathbb{0}_3 & \mathbb{I}_3 & \mathbb{0}_3 & \mathbb{0}_3 \end{smallmatrix}\right]$, du fait de la linéarité de l'équation (3.33a). Pour le second block de ligne, nous nous concentrons sur l'équation (3.33b) et nous commencons par caractériser $\boldsymbol{R}(\boldsymbol{q}_{\mathrm{r,eq}})$, dont la structure est relativement vide de $\boldsymbol{\epsilon}_{\mathrm{r,eq}}$. En particulier, nous rappelons dans (3.23) en utilisant l'expression $\boldsymbol{R}$ dans (3.2), nous pouvons ecrire

$$\boldsymbol{R}(\boldsymbol{q}_{\mathrm{r,eq}}) = \boldsymbol{R}_{\theta} := \begin{bmatrix} 1 - 2\overline{\epsilon}_2^2 & 0 & 2\overline{\epsilon}_2\overline{\eta} \\ 0 & 1 & 0 \\ -2\overline{\epsilon}_2\overline{\eta} & 0 & 1 - 2\overline{\epsilon}_2^2 \end{bmatrix} = \left[\begin{smallmatrix} \cos\theta & 0 & \sin\theta \\ 0 & 1 & 0 \\ -\sin\theta & 0 & \cos\theta \end{smallmatrix}\right],$$

où $\bar{\epsilon}_2 = \sin\frac{\theta}{2}$ représente le deuxième élément de $\boldsymbol{\epsilon}_{\mathrm{r,eq}}$ selon (3.35) et $\overline{\eta} = \sqrt{1-\bar{\epsilon}_2^2} = \cos\frac{\theta}{2}$.

Avec cette expression de $\boldsymbol{R}_\theta$, nous pouvons dériver l'expression de (3.33b), en utilisant la notation abrégée $\cdot|_{\mathrm{eq}}$ pour caractériser l'évaluation d'une fonction (matricielle ou vectorielle) à l'équilibre (3.35),

$$\begin{aligned}\boldsymbol{A}_{vv} &= \frac{\partial}{\partial \boldsymbol{v}}\left(\frac{1}{m}\boldsymbol{R}(\boldsymbol{q}_{\mathrm{r}})\left(\boldsymbol{D}_{\mathrm{f}}(\boldsymbol{u})\|\boldsymbol{w}_{\mathrm{r}}\|\boldsymbol{R}^\top(\boldsymbol{q}_{\mathrm{r}})(\boldsymbol{v}_{\mathrm{r}}-\boldsymbol{w}_{\mathrm{r}})\right)\right)\Big|_{\mathrm{eq}} \\ &= \frac{\partial}{\partial \boldsymbol{v}}\left(\frac{1}{m}\boldsymbol{R}_\theta\boldsymbol{D}_{\mathrm{f}}(\boldsymbol{u}_{\mathrm{eq}})\|\boldsymbol{w}_{\mathrm{eq}}\|\boldsymbol{R}_\theta^\top\boldsymbol{v}_{\mathrm{r}}\right)\Big|_{\mathrm{eq}},\end{aligned} \tag{3.36}$$

qui, compte tenu de l'égalité $\boldsymbol{D}_{\mathrm{f,eq}} = \boldsymbol{D}_{\mathrm{f}}(\boldsymbol{u}_{\mathrm{eq}})$, il est facile de montrer qu'elle coïncide avec la matrice $\boldsymbol{A}_{vv}$ donnée en (3.49) dans l'Algorithme 2.

Nous nous concentrons maintenant sur $\boldsymbol{A}_{v\epsilon}$ de la matrice $\boldsymbol{A}_w$, qui doit être calculée à partir de (3.33b) de manieère simialire à (3.36), comme

$$\boldsymbol{A}_{v\epsilon} = \frac{\partial}{\partial \boldsymbol{\epsilon}}\left(\frac{1}{m}\boldsymbol{R}(\boldsymbol{q}_{\mathrm{r}})\left(\boldsymbol{M}_{\mathrm{f}}(\boldsymbol{u}) + \boldsymbol{D}_{\mathrm{f}}(\boldsymbol{u})\|\boldsymbol{w}_{\mathrm{r}}\|\boldsymbol{R}^\top(\boldsymbol{q}_{\mathrm{r}})\boldsymbol{w}_{\mathrm{r}}\right)\right)\Big|_{\mathrm{eq}}. \tag{3.37}$$

Pour évaluer la partie droite de (3.37), nous demmarons de l'expression de $\boldsymbol{R}(\boldsymbol{q}) = \boldsymbol{R}\left(\left[\begin{smallmatrix}\eta\\ \epsilon\end{smallmatrix}\right]\right)$ dans (3.2), 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

$$\begin{aligned}\partial\boldsymbol{R}_{\boldsymbol{\epsilon}}(\boldsymbol{\epsilon},\mathfrak{v}) &:= \frac{\partial}{\partial\boldsymbol{\epsilon}}\boldsymbol{R}\left(\left[\begin{smallmatrix}\sqrt{1-\boldsymbol{\epsilon}^\top\boldsymbol{\epsilon}}\\ \boldsymbol{\epsilon}\end{smallmatrix}\right]\right)\mathfrak{v} \\ &= 2\eta\,[\mathfrak{v}]_\times\left(\frac{\boldsymbol{\epsilon}\boldsymbol{\epsilon}^\top}{1-\boldsymbol{\epsilon}^\top\boldsymbol{\epsilon}} - \mathbb{I}_3\right) - 4\mathfrak{v}\boldsymbol{\epsilon}^\top + 2\boldsymbol{\epsilon}\mathfrak{v}^\top + 2\boldsymbol{\epsilon}^\top\mathfrak{v}\mathbb{I}_3,\end{aligned} \tag{3.38}$$

qui implique donc

$$\frac{\partial}{\partial\boldsymbol{\epsilon}}\boldsymbol{R}^\top\left(\left[\begin{smallmatrix}\eta\\ \boldsymbol{\epsilon}\end{smallmatrix}\right]\right)\mathfrak{v} = \frac{\partial}{\partial\boldsymbol{\epsilon}}\boldsymbol{R}\left(\left[\begin{smallmatrix}\sqrt{1-\boldsymbol{\epsilon}^\top\boldsymbol{\epsilon}}\\ -\boldsymbol{\epsilon}\end{smallmatrix}\right]\right)\mathfrak{v} = \partial\boldsymbol{R}_{\boldsymbol{\epsilon}}(-\boldsymbol{\epsilon},\mathfrak{v}). \tag{3.39}$$

Pour évaluer (3.37), il sera utile de dériver la forme simplifiée suivante

$$\begin{aligned}&\partial\boldsymbol{R}_{\boldsymbol{\epsilon}}\left(\left[\begin{smallmatrix}0\\ \epsilon_2\\ 0\end{smallmatrix}\right],\left[\begin{smallmatrix}\mathfrak{v}_1\\ 0\\ \mathfrak{v}_3\end{smallmatrix}\right]\right) \\ &= 2\begin{bmatrix}0 & \left(\overline{\eta}-\frac{\bar{\epsilon}_2^2}{\overline{\eta}}\right)\mathfrak{v}_3 & 0\\ -\overline{\eta}\mathfrak{v}_3 & 0 & \overline{\eta}\mathfrak{v}_1\\ 0 & \left(\frac{\bar{\epsilon}_2^2}{\overline{\eta}}-\overline{\eta}\right)\mathfrak{v}_1 & 0\end{bmatrix} + 2\bar{\epsilon}_2\begin{bmatrix}0 & -2\mathfrak{v}_1 & 0\\ \mathfrak{v}_1 & 0 & \mathfrak{v}_3\\ 0 & -2\mathfrak{v}_3 & 0\end{bmatrix}.\end{aligned} \tag{3.40}$$

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 $\boldsymbol{w}$ et des deux entrées similaire des elevons 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 $\boldsymbol{D}_{\mathrm{f}}(\boldsymbol{u})\|\boldsymbol{v}_{\mathrm{b}}\|\boldsymbol{v}_{\mathrm{b}}$ provenant (3.33b) avec $\boldsymbol{D}_{\mathrm{f}}(\boldsymbol{u})$ De (3.11) :

$$\begin{bmatrix} f_{\mathrm{d}} \\ 0 \\ f_\ell \end{bmatrix} = -\boldsymbol{D}_{\mathrm{f}}(\boldsymbol{u}_{\mathrm{eq}})\|\boldsymbol{w}_{\mathrm{eq}}\|\boldsymbol{R}_\theta^\top \boldsymbol{w}_{\mathrm{r,eq}}, \tag{3.41}$$

qui apres calcul coincide avec l'expression (3.47) donner dans l'Algorithme 2.

A partir des deux forces (f_{d}, f_ℓ) dans (3.41), il est possible d determiner leurs dérivé partielle par rapport à la composante $\bar{\epsilon}_2$ du quaternion, qui reprensent le tangage du drone. En utilisant (3.39), nous obtenons

$$\begin{bmatrix} \frac{\partial f_{\mathrm{d}}}{\partial \epsilon_2} \\ 0 \\ \frac{\partial f_\ell}{\partial \epsilon_2} \end{bmatrix} = -\boldsymbol{D}_{\mathrm{f}}(\boldsymbol{u}_{\mathrm{eq}})\|\boldsymbol{w}_{\mathrm{eq}}\|\partial \boldsymbol{R}_{\boldsymbol{\epsilon}}(-\boldsymbol{\epsilon}, \boldsymbol{w}_{\mathrm{r,eq}}) \begin{bmatrix} 0 \\ 1 \\ 0 \end{bmatrix}, \tag{3.42}$$

qui apres calcul en utilisant l'égalité $\boldsymbol{D}_{\mathrm{f,eq}} = \boldsymbol{D}_{\mathrm{f}}(\boldsymbol{u}_{\mathrm{eq}})$, coincide avec l'équation (3.47), donné dans l'Algorithme 2.

En suivant des calculs similaire, 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 élevons, liée à l'écoulement de l'air créé par les hélices, sont obtenues à partir de (3.9) sont défini par

$$\begin{bmatrix} f_{\mathrm{m}} \\ 0 \\ f_{\mathrm{e}} \end{bmatrix} = \boldsymbol{M}_{\mathrm{f}}(\boldsymbol{u}_{\mathrm{eq}}), \tag{3.43}$$

qui, après calculs, coïncident avec les sélections de (3.48), données dans l'Algorithme 2.

En utilisant les définitions (3.38), (3.39), ainsi que les expressions (3.40), (3.41), (**??**), et leurs formes équivalentes indiquées dans (3.47), (3.48) données dans l'Algorithme 2, nous pouvons finalement calculer à partir de (3.37)

$$\begin{aligned} \boldsymbol{A}_{v\epsilon} = \frac{1}{m}\big(\partial \boldsymbol{R}_{\boldsymbol{\epsilon}}(\boldsymbol{\epsilon}, \boldsymbol{M}_{\mathrm{f}}(\boldsymbol{u}_{\mathrm{eq}})) \text{ - } \partial \boldsymbol{R}_{\boldsymbol{\epsilon}}(\boldsymbol{\epsilon}, \boldsymbol{D}_{\mathrm{f}}(\boldsymbol{u}_{\mathrm{eq}})\|\boldsymbol{w}_{\mathrm{eq}}\|\boldsymbol{R}_\theta^\top \boldsymbol{w}_{\mathrm{eq}}) \\ \text{-}\boldsymbol{R}_\theta \boldsymbol{D}_{\mathrm{f}}(\boldsymbol{u})\|\boldsymbol{w}_{\mathrm{r,eq}}\|\partial \boldsymbol{R}_{\boldsymbol{\epsilon}}(-\boldsymbol{\epsilon}, \boldsymbol{w}_{\mathrm{r,eq}}) \begin{bmatrix} 0 \\ 1 \\ 0 \end{bmatrix}\big)\Big|_{\mathrm{eq}}. \end{aligned}$$

qui fournit l'expression (3.50) dans l'Algorithme 2 après quelques calculs exploitant également $\boldsymbol{D}_{textf,eq} = \boldsymbol{D}_{textf}(\boldsymbol{u}_{\mathrm{eq}})$.

Nous nous concentrons maintenant sur la matrice $\boldsymbol{A}_{\epsilon\omega}$ de $\boldsymbol{A}_w$, et nous rappelons que, en raison des propriétés du produit de quaternion (voir, par exemple, [Hua 2013]), $\left[\begin{smallmatrix} \eta \\ \boldsymbol{\epsilon} \end{smallmatrix}\right] \otimes \left[\begin{smallmatrix} 0 \\ \boldsymbol{\omega}_{\mathrm{b}} \end{smallmatrix}\right] = \left[\begin{smallmatrix} -\boldsymbol{\epsilon}^\top \\ \eta \mathbb{I}_3 + [\boldsymbol{\epsilon}]_\times \end{smallmatrix}\right] \boldsymbol{\omega}_{\mathrm{b}}$. A partir des deux termes inférieurs de la matrice du côté droit de cette dernière équation, en développant (3.33c) et en calculant $\boldsymbol{A}_{\epsilon\omega} = \frac{\partial}{\partial \boldsymbol{\omega}_{\mathrm{b}}} \left(\frac{1}{2}\boldsymbol{q}_{\mathrm{r}} \otimes \left[\begin{smallmatrix} 0 \\ \boldsymbol{\omega}_{\mathrm{b}} \end{smallmatrix} \right] \right)\Big|_{\mathrm{eq}}$, nous obtenons les deux termes de l'expression (3.50) donnée dans l'Algorithme 2.

Nous nous concentrons maintenant sur la matrice $\boldsymbol{A}_{\omega\epsilon}$ de $\boldsymbol{A}_w$, qui doit être calculée à partir de (3.33d). Comme seul le dernier terme de la partie droite dépend de $\boldsymbol{\epsilon}$ (par l'intermédiaire de $\boldsymbol{q}_{\mathrm{r}}$), nous obtenons

$$\boldsymbol{A}_{\omega\epsilon} = \boldsymbol{J}^{-1}\boldsymbol{D}_{\mathrm{m}}(\boldsymbol{u}_{\mathrm{eq}})\|\boldsymbol{w}_{\mathrm{r}}\| \, \frac{\partial}{\partial \boldsymbol{\epsilon}} \left(\boldsymbol{R}^{\top}(\boldsymbol{q}_{\mathrm{r}})(\boldsymbol{v}_{\mathrm{r}} - \boldsymbol{w}_{\mathrm{r}}) \right)\Big|_{\mathrm{eq}} . \tag{3.44}$$

Pour calculer l'expression explicite de (3.44), nous exploitons à nouveau (3.39) et (3.40), et utilisons l'expression de $\boldsymbol{D}_{\mathrm{m}}$ dans (3.12), ainsi que les identités $\overline{\eta}^2 - \overline{\epsilon}_2^2 = \cos\theta$ et $2\overline{\eta}\overline{\epsilon}_2^2 = \sin\theta$, qui fournissent, après quelques simplifications, l'expression (3.51), donnée dans l'Algoirithme 2.

Passons maintenant à la dérivation des entrées de la matrice $\boldsymbol{G}_w$ dans (3.31), dont les composantes peuvent être dérivées de (3.33b) et (3.33d). En utilisant les quatre entrées de $\boldsymbol{u}$ dans (3.7), et en se basant également sur la structure de $\boldsymbol{M}_{\mathrm{f}}$, $\boldsymbol{D}_{\mathrm{f}}$, dans (3.9), (3.11), la forme explicite pour

$$\boldsymbol{G}_v = \frac{1}{m}\boldsymbol{R}_\theta \frac{\partial}{\partial \boldsymbol{u}} \left(\boldsymbol{M}_{\mathrm{f}}(\boldsymbol{u}) - \boldsymbol{D}_{\mathrm{f}}(\boldsymbol{u})\|\boldsymbol{w}_{\mathrm{r}}\|\boldsymbol{w}_{\mathrm{eq}}^{\mathrm{b}} \right)\Big|_{\mathrm{eq}} , \tag{3.45}$$

peut être calculée comme dans (3.52), après quelques factorisations.

De même, sur la base des matrices $\boldsymbol{M}_{\mathrm{m}}$, $\boldsymbol{D}_{\mathrm{m}}$ dans (3.10), (3.12), nous pouvons calculer

$$\boldsymbol{G}_\omega = \boldsymbol{J}^{-1} \frac{\partial}{\partial \boldsymbol{u}} \left(\boldsymbol{M}_{\mathrm{m}}(\boldsymbol{u}) - \boldsymbol{D}_{\mathrm{m}}(\boldsymbol{u})\|\boldsymbol{w}_{\mathrm{r}}\|\boldsymbol{w}_{\mathrm{eq}}^{\mathrm{b}} \right)\Big|_{\mathrm{eq}} \tag{3.46}$$

comme dans (3.53), après quelques factorisations.

Déterminons enfin l'expression de $\boldsymbol{E}_v$ dans (3.31). Notons d'abord que nous pouvons écrire $\|\boldsymbol{w}_{\mathrm{r}}\|\boldsymbol{w}_{\mathrm{r}} = \boldsymbol{w}_{\mathrm{r}}\sqrt{\boldsymbol{w}_{\mathrm{r}}^{\top}\boldsymbol{w}_{\mathrm{r}}}$, de sorte que

$$\frac{\partial}{\partial \boldsymbol{w}_{\mathrm{r}}} \|\boldsymbol{w}_{\mathrm{r}}\|\boldsymbol{w}_{\mathrm{r}} = \|\boldsymbol{w}_{\mathrm{r}}\|\mathbb{I}_3 + \frac{\boldsymbol{w}_{\mathrm{r}}\boldsymbol{w}_{\mathrm{r}}^{\top}}{\|\boldsymbol{w}_{\mathrm{r}}\|} = \|\boldsymbol{w}_{\mathrm{r}}\| \left(\mathbb{I}_3 + \frac{\boldsymbol{w}_{\mathrm{r}}\boldsymbol{w}_{\mathrm{r}}^{\top}}{\boldsymbol{w}_{\mathrm{r}}^{\top}\boldsymbol{w}_{\mathrm{r}}} \right).$$

Ensuite, à partir de (3.33b) et (3.33d) et en suivant des calculs similaires aux cas précédents, en utilisant également l'expression de $\boldsymbol{w}_{\mathrm{r}}$ dans (3.32), nous obtenons l'expression (3.54) (indiquée dans l'Algorithme 2), pour $\boldsymbol{E}_v = -\frac{1}{m}\boldsymbol{R}_\theta \frac{\partial}{\partial \boldsymbol{w}_{\mathrm{r}}} \left(\boldsymbol{D}_{\mathrm{f}}(\boldsymbol{u})\|\boldsymbol{w}_{\mathrm{r}}\|\boldsymbol{w}_{\mathrm{r}} \right)\Big|_{\mathrm{eq}}$ et $\boldsymbol{E}_w = -\boldsymbol{J}^{-1} \frac{\partial}{\partial \boldsymbol{w}_{\mathrm{r}}} \left(\boldsymbol{D}_{\mathrm{m}}(\boldsymbol{u})\|\boldsymbol{w}_{\mathrm{r}}\|\boldsymbol{w}_{\mathrm{r}} \right)\Big|_{\mathrm{eq}}$, où nous rappelons que $\boldsymbol{D}_{\mathrm{m,eq}} = \boldsymbol{D}_{\mathrm{m}}(\boldsymbol{u}_{\mathrm{eq}})$. □

aduire l'algo

Algorithme 2 Détermination des matrices de la linéarisation de (3.31)

Entrées : Vecteur de vent $\boldsymbol{w}_{\text{eq}} = \left[\begin{smallmatrix} w_x & w_y & w_z \end{smallmatrix}\right]^\top$ et d'équilibre $(\boldsymbol{u}_{\text{eq}}, \boldsymbol{x}_{\text{eq}})$ provenant de (3.16) et de l'Algorithme 1.

Sorties : Matrices $\boldsymbol{A}_w$, $\boldsymbol{G}_w$, $\boldsymbol{E}_w$ dans (3.31)

1: Sélectionner les paramètres ψ, θ, τ, δ de (3.16) 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 (3.23), (3.11), (3.12), définissons :

$$\boldsymbol{R}_\psi := \begin{bmatrix} \cos\psi & \sin\psi & 0 \\ -\sin\psi & \cos\psi & 0 \\ 0 & 0 & 1 \end{bmatrix}, \quad \boldsymbol{R}_\theta := \begin{bmatrix} \cos\theta & 0 & \sin\theta \\ 0 & 1 & 0 \\ -\sin\theta & 0 & \cos\theta \end{bmatrix},$$

$$\begin{bmatrix} w_{\text{r}x} \\ 0 \\ w_{\text{r}z} \end{bmatrix} := \boldsymbol{R}_\psi^\top \boldsymbol{w}_{\text{eq}}, \quad \begin{bmatrix} w_x^{\text{b}} \\ w_z^{\text{b}} \end{bmatrix} := \begin{bmatrix} w_{\text{r}x}\cos\theta - w_{\text{r}z}\sin\theta \\ w_{\text{r}z}\cos\theta + w_{\text{r}x}\sin\theta \end{bmatrix}$$

$$\left[\boldsymbol{D}_{\text{f,eq}} \middle| \boldsymbol{D}_{\text{m,eq}}\right] := \frac{\rho S}{2} \left[\begin{array}{ccc|ccc} -C_{\text{d}} & 0 & C_{\text{d}}\xi_{\text{f}}\delta & 0 & 0 & 0 \\ 0 & 0 & 0 & \Delta_{\text{r}} C_\ell \xi_{\text{m}}\delta & 0 & 2\Delta_{\text{r}} C_\ell \\ -C_\ell\xi_{\text{f}}\delta & 0 & -C_\ell & 0 & 0 & 0 \end{array}\right]$$

3: Definisson les forces de portance et de trainé ainsi que leurs dérivées par rapport à ϵ_2 (défini dans l'étape 1), comme

$$\begin{bmatrix} f_{\text{d}} & \frac{\partial f_{\text{d}}}{\partial \epsilon_2} \\ 0 & 0 \\ f_\ell & \frac{\partial f_\ell}{\partial \epsilon_2} \end{bmatrix} := -\|\boldsymbol{w}_{\text{eq}}\| \boldsymbol{D}_{\text{f,eq}} \begin{bmatrix} w_x^{\text{b}} & \left(4\bar{\eta} - \frac{2\bar{\epsilon}_2^2}{\bar{\eta}}\right) w_{\text{r}z} - 8\bar{\epsilon}_2 w_{\text{r}x} \\ 0 & 0 \\ w_z^{\text{b}} & \left(4\bar{\eta} - \frac{2\bar{\epsilon}_2^2}{\bar{\eta}}\right) w_{\text{r}x} - 8\bar{\epsilon}_2 w_{\text{r}z} \end{bmatrix}, \tag{3.47}$$

4: Définissons les forces des moteurs et des élevons comme

$$\begin{bmatrix} f_{\text{m}} \\ f_{\text{e}} \end{bmatrix} := \begin{bmatrix} \left(\frac{S_{\text{wet}} C_{\text{d}}}{2S_{\text{p}}} - 2\right)\tau \\ -\frac{S_{\text{wet}}\tau\delta\xi_{\text{f}} C_\ell}{2S_{\text{p}}} \end{bmatrix} \tag{3.48}$$

5: Sélectionnons les matrices $\boldsymbol{A}_w$ dans (3.31) comme :

$$\boldsymbol{A}_{vv} = \frac{\|\boldsymbol{w}_{\text{eq}}\|}{m} \boldsymbol{R}_\theta \boldsymbol{D}_{\text{f,eq}} \boldsymbol{R}_\theta^\top \tag{3.49}$$

$$\begin{bmatrix} \boldsymbol{A}_{v\epsilon}^{1,2} \\ \boldsymbol{A}_{v\epsilon}^{2,1} \\ \boldsymbol{A}_{v\epsilon}^{2,3} \\ \boldsymbol{A}_{v\epsilon}^{3,2} \end{bmatrix} := \begin{bmatrix} 2\bar{\eta} - \frac{\bar{\epsilon}_2^2}{\bar{\eta}} & 4\bar{\epsilon}_2 & 2\bar{\epsilon}_2^2 - 1 & 2\bar{\epsilon}_2\bar{\eta} \\ -2\bar{\eta} & -2\bar{\epsilon}_2 & 0 & 0 \\ 2\bar{\epsilon}_2 & -2\bar{\eta} & 0 & 0 \\ -4\bar{\epsilon}_2 & 2\bar{\eta} - \frac{\bar{\epsilon}_2^2}{\bar{\eta}} & -2\bar{\epsilon}_2\bar{\eta} & 1 - 2\bar{\epsilon}_2^2 \end{bmatrix} \begin{bmatrix} f_{\text{e}} + f_\ell \\ f_{\text{m}} + f_{\text{d}} \\ \frac{\partial f_{\text{d}}}{\partial \bar{\epsilon}_2} \\ \frac{\partial f_\ell}{\partial \bar{\epsilon}_2} \end{bmatrix}$$

$$\boldsymbol{A}_{v\epsilon} = \frac{1}{m} \begin{bmatrix} 0 & \boldsymbol{A}_{v\epsilon}^{1,2} & 0 \\ \boldsymbol{A}_{v\epsilon}^{2,1} & 0 & \boldsymbol{A}_{v\epsilon}^{2,3} \\ 0 & \boldsymbol{A}_{v\epsilon}^{3,2} & 0 \end{bmatrix}, \boldsymbol{A}_{\epsilon\omega} = \frac{\bar{\eta}}{2}\mathbb{I}_3 + \frac{\bar{\epsilon}_2}{2} \begin{bmatrix} 0 & 0 & 1 \\ 0 & 0 & 0 \\ -1 & 0 & 0 \end{bmatrix} \tag{3.50}$$

$$\boldsymbol{A}_{\omega\epsilon} = \frac{\rho S C_\ell \Delta_{\text{r}} \|\boldsymbol{w}_{\text{eq}}\| (w_x^{\text{b}} - \xi_{\text{m}}\delta w_z^{\text{b}})}{J_y \bar{\eta}} \begin{bmatrix} 0 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 0 \end{bmatrix} \tag{3.51}$$

6: Sélectionnons les matrices $\boldsymbol{G}_w$ dans (3.31) comme :

$$\boldsymbol{G}_v = \frac{1}{m}\boldsymbol{R}_\theta \left[\boldsymbol{G}_{v\tau} \middle| \boldsymbol{G}_{v\delta}\right], \; \boldsymbol{G}_{v\tau} := \begin{bmatrix} 1 - \frac{S_{\text{wet}} C_{\text{d}}}{4S_{\text{p}}} \\ 0 \\ -\frac{S_{\text{wet}} C_\ell \xi_{\text{f}}\delta}{2S_{\text{p}}} \end{bmatrix} \left[\begin{smallmatrix} 1 \\ 1 \end{smallmatrix}\right]^\top$$

$$\boldsymbol{G}_{v\delta} := \begin{bmatrix} -\frac{1}{4}\rho S C_{\text{d}} \xi_{\text{f}} \|\boldsymbol{w}_{\text{eq}}\| w_z^{\text{b}} \\ 0 \\ -\frac{S_{\text{wet}} C_\ell \xi_{\text{f}}\tau}{2S_{\text{p}}} + \frac{1}{4}\rho S C_\ell \xi_{\text{f}} \|\boldsymbol{w}_{\text{eq}}\| w_x^{\text{b}} \end{bmatrix} \left[\begin{smallmatrix} 1 \\ 1 \end{smallmatrix}\right]^\top \tag{3.52}$$

$$\boldsymbol{G}_\omega = \boldsymbol{J}^{-1} \left[\boldsymbol{G}_{\omega\tau} \; \boldsymbol{G}_{\omega\delta}\right], \boldsymbol{G}_{\omega\delta} := \frac{S_{\text{wet}} C_\ell \tau}{4S_{\text{p}}} \begin{bmatrix} a_y\xi_{\text{f}} & -a_y\xi_{\text{f}} \\ \Delta_{\text{r}}\xi_{\text{m}} & \Delta_{\text{r}}\xi_{\text{m}} \\ 0 & 0 \end{bmatrix} + $$

$$+ \frac{\rho S \|\boldsymbol{w}_{\text{eq}}\| \xi_{\text{m}}}{4} \begin{bmatrix} a_y C_{\text{d}} w_x^{\text{b}} & -a_y C_{\text{d}} w_x^{\text{b}} \\ \Delta_{\text{r}} C_\ell w_x^{\text{b}} & \Delta_{\text{r}} C_\ell w_x^{\text{b}} \\ a_y C_\ell w_z^{\text{b}} & -a_y C_\ell w_z^{\text{b}} \end{bmatrix} \tag{3.53}$$

3.5 Conclusion du Chapitre 3

nclusion sur la
dification du
teur de com-
nde

Chapitre 4

Commande hybride

Sommaire

4.1 Motivation **33**

4.1.1 Nonlinear dynamic feedback controller 34

4.1.2 Linear control design 36

4.1.3 Hysteresis-based local-global control design 37

4.2 Schéma de commande hybride **39**

4.3 Simulations **39**

4.1 Motivation

Basée sur les capacité d'un *tailsitter*, il est legitime de se poser la question du mode de vol utilise pour rejoindre un point. Effectivement, le drone a la possibilité de se déplacer en stationnaire ou bien en vol d'avancement.

We propose in this section two control design strategies for stabilizing a hovering position. The first one is inspired by the nonlinear stabilizer presented in [**?**] and provides a large region of attraction, and the second one is based on the linearized dynamics and allow for more effective gain tuning in the final approaching phase. The two controllers are united via a hybrid mechanism that allows retaining the steady-state performance of the linearized design with the large region of attraction guaranteed by the nonlinear design. Our solution is tested by simulating the full nonlinear model.

Remarque 1. We emphasize that vector $\boldsymbol{u}$ in (3.7) corresponds to a non-invertible transformation of the actual DarkO actuators corresponding to $\boldsymbol{u}_{\mathrm{act}} := [\omega_1,\ \omega_2,\ \delta_1,\ \delta_2]^\top$. Nevertheless, when imposing the saturation constraints discussed in Remark **??**, it is possible to uniquely determine $\boldsymbol{u}_{\mathrm{act}}$ from a desired value of $\boldsymbol{u}$ in (3.7), because nonzero positive values of ω_1 and ω_2 can be determined from the first two components of $\boldsymbol{u}$, and then δ_1 and δ_2 are easily constructed from the last two components of $\boldsymbol{u}$. ◦

4.1.1 Nonlinear dynamic feedback controller

We illustrate in this section a nonlinear dynamic control law inspired by the result of [**?**]. For the nonlinear control law of [**?**] to be applicable, matrices F and M reported in (**??**) must allow defining a so-called zero moment direction $\bar{\boldsymbol{u}} \in \mathbb{R}^4$ ensuring $|F\bar{\boldsymbol{u}}| = 1$ and $M\bar{\boldsymbol{u}} = 0$, and a right inverse M^r of M satisfying $MM^r = I$ and $FM^r = 0$. In our case, it is immediate to see that the zero-moment direction $\bar{\boldsymbol{u}} = \frac{\sqrt{2}}{2a_{\mathrm{f}}}\left[\begin{smallmatrix}1 & 1 & 0 & 0\end{smallmatrix}\right]^\top$ satisfies the required conditions, whereas the fact that $\operatorname{rank}(F) = 2$ (so that $\ker F$ has dimension 2) makes it impossible to find a right inverse M^r of M completely contained in $\ker F$. Due to this fact, we determine M^r by (conservatively) parametrizing the right pseudoinverses of M as $M^r := KM^\top(MKM^\top)^{-1}$ where parameter $K \in \mathbb{R}^{4\times 4}$ is symmetric and satisfies $MKM^\top \geq I$ (to ensure invertibility). Under this parametrization, the goal is to minimize the norm of $FM^r = FKM^\top(MKM^\top)^{-1}$, which is well achieved by minimizing the norm of $FKM^\top$, due to the fact that the constraint on $MKM^\top \geq I$ ensures that the factor $(MKM^\top)^{-1}$ has norm smaller than 1. Performing a Schur complement, this minimization is well obtained by solving the following semi-definite program :

$$\min_{K,\kappa} \kappa, \text{ subject to} : MKM^\top \geq I, \begin{bmatrix} \kappa I & FKM^\top \\ MK^\top F^\top & \kappa I \end{bmatrix} \geq 0,$$

which minimizes κ while ensuring $FKM^\top MK^\top F^\top \leq \kappa^2 I$. Solving this optimization, we obtain, for the specific matrices under consideration,

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

leading to $\kappa = 39.7$. With this optimality-based selection, the nonlinear dynamic design of [**?**] can be effectively applied by obtaining responses that are almost indistinguishable from the fully decoupled case $FM^r = 0$. Note that a similar approach, essentially neglecting the extra terms acting on the translational dynamics is also suggested in the survey paper [Hua 2013].

Based on the above-described choice of M^r and $\bar{\boldsymbol{u}}$, applying the feedback law in [**?**, eqn (19)], the input $\boldsymbol{u}$ becomes :

$$\boldsymbol{u} = \boldsymbol{u}_{\mathrm{nl}} := M^r \boldsymbol{\tau_r} + \bar{\boldsymbol{u}}\boldsymbol{f}, \tag{4.1}$$

where $\boldsymbol{\tau_r}$ and $\boldsymbol{f}$ are provided by the dynamic feedback controller proposed in [**?**].

The optimality-based selection of M^r is prone to a few interesting interpretations when observing the product $M^r\boldsymbol{\tau_r} = M^r\left[\begin{smallmatrix}\tau_{r,x} & \tau_{r,y} & \tau_{r,z}\end{smallmatrix}\right]^\top$. First, to produce a moment $\tau_{r,z}$ about the z-axis we mainly use the thrust differential action ; secondly, a moment $\tau_{r,y}$ about the y-axis is generated by an equal (additive) use of the two flaps, with great efficiency ; finally a moment $\tau_{r,x}$ about the x-axis comes from a differential use of the flaps.

As a final remark, as compared to the solution proposed in [?], to partially take into account the saturation effects highlighted in Remark ??, the error feedback interconnection of the outer loop in [?] has been augmented with a simple error governor strategy never allowing the translational position error $\mathbf{e}_p$ entering [?, eqn. (22)] to exceed the maximum value of 3 meters. The remaining tuning gains required in the solution of [?] have been selected following an intuitive PD tuning procedure as $k_{pp} = 0.5$, $k_{pd} = 1.2$, $k_{ap} = 0.08$, $k_{ad} = 0.1$ and $k_\Delta = 1$.

Figure 4.1 shows the response of the system in terms of linear and angular positions (top two rows) and actuators efforts (bottom two rows) when the system starts from the initial condition $\boldsymbol{x}(0) = [\boldsymbol{p}(0)\ \boldsymbol{v}(0)\ \boldsymbol{q}(0)\ \boldsymbol{\omega_b}(0)]^\top = [0\ 0\ 0\ 0\ 0\ 0\ 0.9140\ 0.1134\ \ -0.3728\ 0.1134\ 0\ 0\ 0]^\top$ with a target equilibrium position of $\boldsymbol{p}_{\text{eq}} = [4\ 5\ 6]^\top$ and $\boldsymbol{q}_{\text{eq}} = [\frac{\sqrt{2}}{2}\ 0\ \ -\frac{\sqrt{2}}{2}\ 0]^\top$. A graceful response can be seen, which remains quite far from the actuator saturations (see Remark ??). Increasing the tuning gains can speed up the response but provides undesired attitude oscillations. Therefore it is interesting to combine this nonlinear controller (providing a large region of attraction) with a more aggressive controller, designed based on the linearized dynamics (3.28) and to be used to improve the fail of the response.

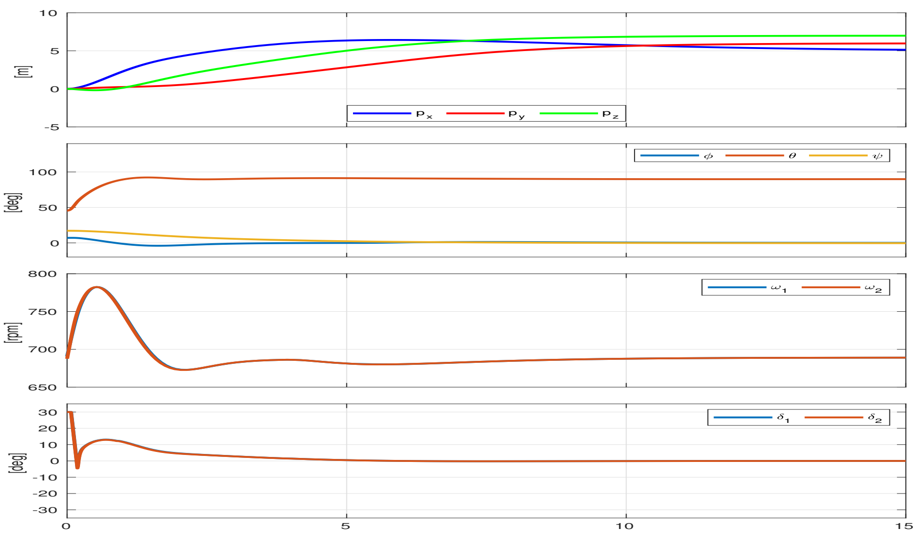

FIGURE 4.1 – Simulation with the nonlinear dynamic feedback controller.

4.1.2 Linear control design

Based on the performance-oriented observations of the previous section, given a target position corresponding to an equilibrium $\boldsymbol{p}_{\text{eq}}, \boldsymbol{q}_{\text{eq}}$ as characterized in Proposition **??**, we design here a local linear feedback controller capable of inducing a more aggressive response. To this end, we focus on the linearized dynamics (3.28) and recognize that we can design a state feedback controller

$$\boldsymbol{u}_{\text{lin}} := \boldsymbol{u}_{\text{eq}} - K\tilde{\boldsymbol{x}}, \tag{4.2}$$

where $\tilde{\boldsymbol{x}}$ has been introduced in (3.30) and $K \in \mathbb{R}^{4\times 12}$ is a state feedback gain that can be designed, based on the matrices A and G appearing in (3.28), in such a way that the closed-loop linear feedback $A_{\text{cl}} := A - GK$ be exponentially stable.

For our design, we have used an LQR selection, associated with the simplest possible weight matrices selection $Q = I_{12}$ and $R = I_4$, which gives desirable closed-loop responses. The LQR design also provides a positive definite Lyapunov certificate matrix $S \in \mathbb{R}^{12\times 12}$ (solution of the algebraic Riccati equation) ensuring that $A_{\text{cl}}^\top S + SA_{\text{cl}} < 0$. In particular, it is well known from the linear approximation theorem that function $V(\tilde{\boldsymbol{x}}) = \tilde{\boldsymbol{x}}^\top S\tilde{\boldsymbol{x}}$ is also a Lyapunov function certifying local exponential stability of $\boldsymbol{x}_{\mathbf{eq}}$ for the nonlinear dynamics. More specifically, there exits a positive scalar $\bar{v} \in \mathbb{R}$ such that, along dynamics (3.8), we have :

$$V(\tilde{\boldsymbol{x}}) \leq \bar{v} \quad \Rightarrow \quad \dot{V}(\tilde{\boldsymbol{x}}) := \langle \nabla V(\tilde{\boldsymbol{x}}), \dot{\tilde{\boldsymbol{x}}} \rangle < 0, \tag{4.3}$$

for all $\tilde{\boldsymbol{x}} \neq 0$; in other words, the sublevel set $V(\tilde{\boldsymbol{x}}) \leq \bar{v}$ is contained in the basin of attraction of the equilibrium $\boldsymbol{x}_{\text{eq}}$.

Determining the largest possible scalar $\bar{v}$ ensuring (4.3) is a challenging problem and conservative lower bounds of this quantity can be determined by quantifying the effect of the nonlinearities on the dynamics. Since $\dot{\tilde{\boldsymbol{x}}}$ is a function of $\boldsymbol{x}$, then it is fairly easy to algebraically evaluate $\dot{V}(\tilde{\boldsymbol{x}})$ for a large amount of random extractions of the variable $\tilde{\boldsymbol{x}}$, so as to get a probabilistic estimate of the largest $\bar{v}$. Rigorous guarantees about these selections can be obtained by applying the results in [**?**], which is out of the scope of this paper, but an evaluation of 10000 samples confirmed that the value $\bar{v} = 400$ is a good candidate selection satisfying (4.3).

Figure 4.2 shows a simulation starting at the origin with a zero orientation on the three axes (horizontal UAV) and zero initial velocities, with a target position $\boldsymbol{p}_{\text{eq}} = [4,\ 5,\ 6]$ with a hovering stabilization (vertical UAV) with $\beta = 0$. The dotted line represents the target position on each axis. Note that the initial linear and angular velocities are zero. The last graph shows the desirable exponential decay

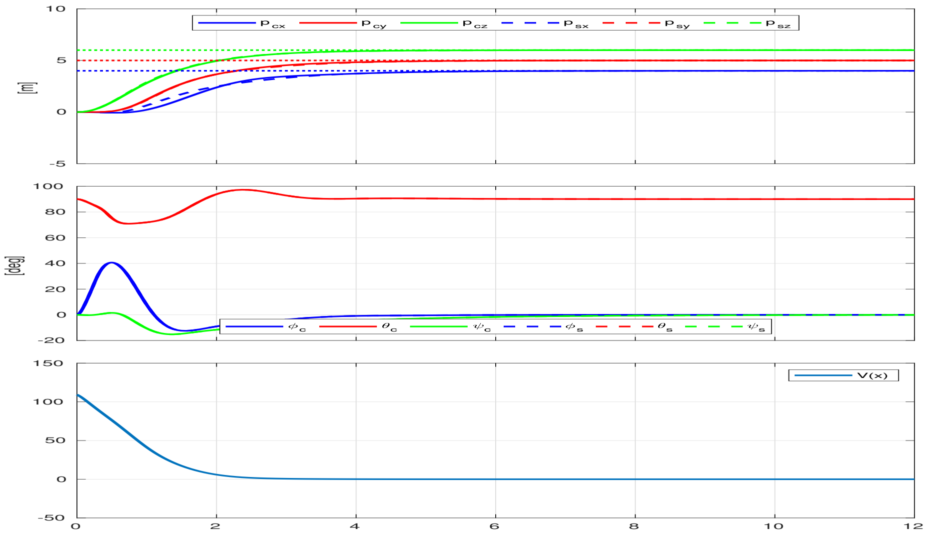

FIGURE 4.2 – Simulation of the full model (solid) and (3.8) (dashed) with $\boldsymbol{u} = \boldsymbol{u}_{\text{lin}}$ as in (4.2) from an initial condition $\tilde{\boldsymbol{x}}_{\mathbf{0}}$ within the basin of attraction.

of V Figure 4.2 shows both the simulation of the full model (solid) of [**?**] and of the simplified nonlinear model (3.8) (dashed) showing some significant differences in the initial response. When providing a larger target position $\boldsymbol{p}_{\text{eq}} = [8,\ 9,\ 10]$ (with the same orientation), the initial condition is outside the basin of attraction and diverging solutions are experienced as shown in Figure 4.3.

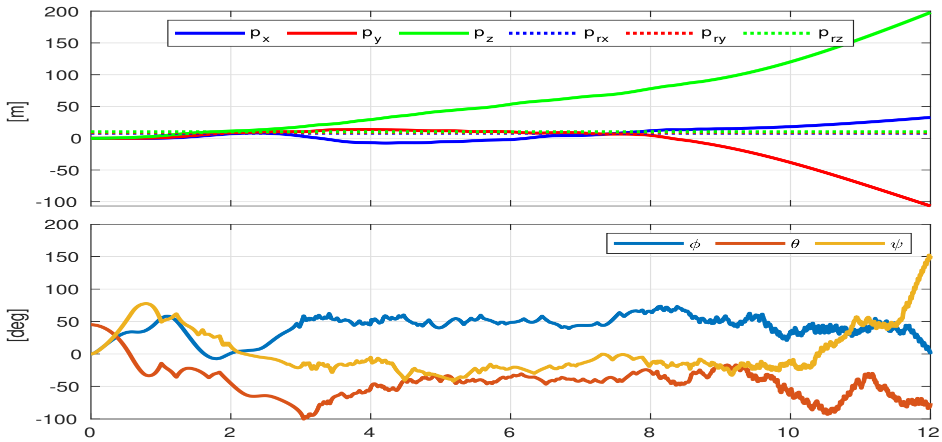

FIGURE 4.3 – Diverging simulation of the full model with $\boldsymbol{u} = \boldsymbol{u}_{\text{lin}}$ as in (4.2) from an initial condition $\tilde{\boldsymbol{x}}_{\mathbf{0}}$ outside the basin of attraction.

4.1.3 Hysteresis-based local-global control design

Inspired by the local/global strategies presented in [**?**, Ex. 1.7], similar to the solution presented in [**?**], we use a hybrid mechanism to switch between the high performance local feedback (4.2) (as long as the state is in the basin of attraction of the equilibrium) and the less aggressive nonlinear controller (4.1), which provides a larger region of attraction (and can be called with an abuse of notation the "global controller"). To this end, we augment the controller state with a logical state variable $\ell \in \{0, 1\}$, governing the choice of the control input between (4.1) and (4.2) as

$$\boldsymbol{u} = \boldsymbol{u}_{\text{hyb}} := \ell \boldsymbol{u}_{\text{nl}} + (1 - \ell)\boldsymbol{u}_{\text{lin}}, \tag{4.4}$$

We ensure, through the hybrid dynamics, that ℓ can only take values in $\{0, 1\}$. Its dynamics is defined by :

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

where $\chi = [\boldsymbol{p},\ \boldsymbol{v},\ \boldsymbol{q},\ \boldsymbol{\omega},\ l]$ is the complete closed loop state and $\mathcal{C}$ and $\mathcal{D}$ are, respectively, the flow and the jump sets, defined as

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

where $V(\tilde{\boldsymbol{x}}) := \tilde{\boldsymbol{x}}^\top S \tilde{\boldsymbol{x}}$ has been defined in the previous section, $\overline{v} = 400$ has been determined in the previous section to satisfy (4.3), and $\underline{v}$ is any positive constant satisfying $\underline{v} < \overline{v}$ (a smaller choice of $\underline{v}$ increases the hysteresis margin but postpones the desirable high performance tail of the feedback response). In our case we choose $\underline{v} = 350$. The following result is an immediate consequence of the results in [**?**, Ex. 1.7] and the properties of our linear and nonlinear designs.

Proposition 1. *Under the action of the hybrid feedback* (4.4)*, the closed loop exhibits the same basin of attraction as the one associated with the nonlinear controller* (4.1)*, while always using the high-performance linear feedback* (4.2) *in the tail of the response.*

We performed several simulations of the closed loop using the Matlab toolbox [**?**]. The simulations are carried out with the complete model of the UAV [**?**], including all the nonlinear aerodynamic effects. A sample simulation is reported in Figure **??**, where we initialize the UAV at the origin with zero roll and yaw orientation, and with a pitch angle of 45 degrees. The target orientation is in the vertical hovering configuration and the target position is assigned to $\boldsymbol{p}_{\mathrm{eq}} = [50,\ 25,\ 12.5]$.

We observe that in the time phase $t \in [0, 38]$, the UAV exhibits a graceful but slow convergence to the desired target position using the global controller ($\ell = 1$). At that time, the state enters set $\mathcal{D}_1$ and the more aggressive local controller is activated up to the convergence to the desired equilibrium.

To perform realistic simulations, the measurements are affected by sensors noise. The intrinsic robustness of the hybrid feedback, established in [**?**, Chapter 7] is confirmed by the graceful performance degradation as a function of the amplitude of the measurement noise.

4.2 Schéma de commande hybride

4.3 Simulations

Chapitre 5

Étude longitudinale sur une maquette à trois degrés de liberté

Sommaire

5.1 Motivation **41**
5.2 Schéma de commande linéaire proportionnel intégral **41**
5.3 Maquette expérimentale **41**
5.4 Test bench presentation **41**
5.4.1 Motivation for the design 41
5.4.2 Physical description, sensing, and actuation 42
5.4.3 Software-in-the-loop translational motion 45
5.5 Integral-based linear control **46**
5.5.1 Description of the control scheme 46
5.5.2 H_∞-based optimization 48
5.6 Résultats **49**

5.1 Motivation

5.2 Schéma de commande linéaire proportionnel intégral

5.3 Maquette expérimentale

5.4 Test bench presentation

5.4.1 Motivation for the design

Closed-loop simulations with the controller developed in [**?**,], show that in the presence of constant horizontal wind in the $(x_{[B]}, z_{[B]})$ plane, the UAV changes its pitch angle. This behaviour has also been observed in wind tunnel tests with the experimental device. Intuitively speaking, a reduced angle of attack leads to a smaller surface area facing the wind, so as to reduce the drag force, which strongly impacts the position. At the same time, the air flow due to the constant wind generates lift, compensated by a reduced propeller thrust and a consequent reduction of the UAV consumption. The goal of the prototype described here is to evaluate experimentally the effect of the wind on the DarkO device.

5.4.2 Physical description, sensing, and actuation

The developed prototype comprises 3D printed parts in Onyx and PLA (polylactic acid, a thermoplastic polyester). It is especially designed for running experiments in front of a wind tunnel with responses that resemble those of the DarkO due to the similar shape (see Fig. 5.2). The central part, which contains the onboard avionics (autopilot, GPS, etc.) in the DarkO, has been here replaced by a one-degree of freedom revolute joint (see Fig. 5.1). The wings are the same as those of the DarkO, with the electronic speed controllers (ESC), governing the brushless motor speed, placed in the wings. As described in Section 5.4.1, we wish to represent and study the y-axis degree of freedom of the DarkO UAV. The main carbon tube linking the two wings is used as the rotation axis. This tube is fixed to two bearings placed 28.5 mm apart in order to obtain a solid fixation of the rig. This rotation axis is equipped with an optical quadrature rotary encoder to accurately measure the orientation of the device. The advantage of this sensor is that it does not produce torque on the rotation axis. This encoder offers 4000 edges per revolution, which results in a resolution of $0.09°/pulse$.

As shown in Fig. 5.1, the indexer and the holder are drilled so that the rotation can be locked at known positions (0°, 90°, etc.) by a screw on the indexer that fits into the holes of the holder. Locking the device allows for a correct initialization of the incremental encoder. Locking also allows placing the device in specific exact positions in order to identify the aerodynamic coefficients.

The pivoting joint is also equipped with a 6 degrees of freedom (DOF) force-torque sensor, providing a measurement of the internal wrench exerted on the experimental device by the support. The experimental test bench is also equipped with a hot wire to measure the airspeed seen by the drone.

The photo reported in Fig. 5.2 shows the experimental device in its test environment. The drone is placed in front of an open wind tunnel, called WindShape, generating a horizontal wind between 2 and 16 $\mathrm{m\,s^{-1}}$. Thus, during our tests, we consider the vertical wind component to be zero. The drone is placed at the centre

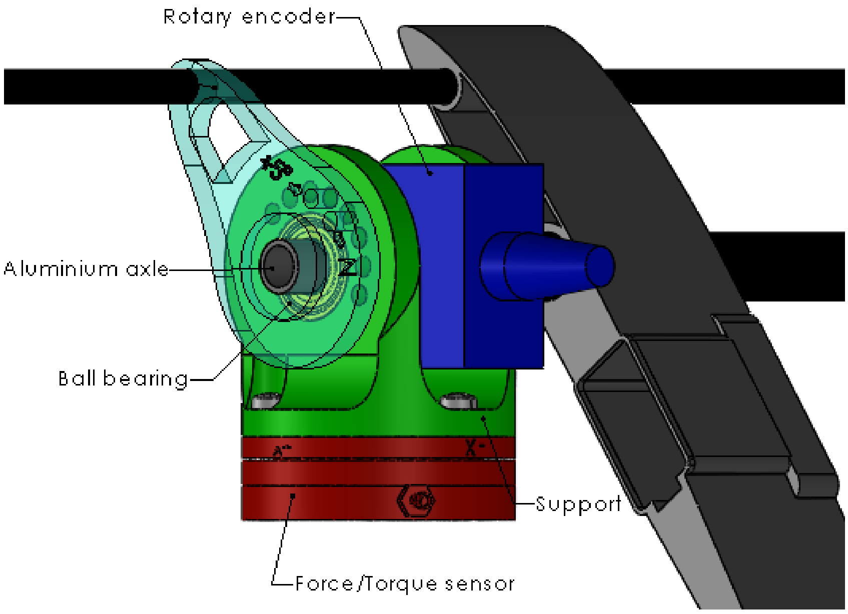

FIGURE 5.1 – The one degree-of-freedom joint.

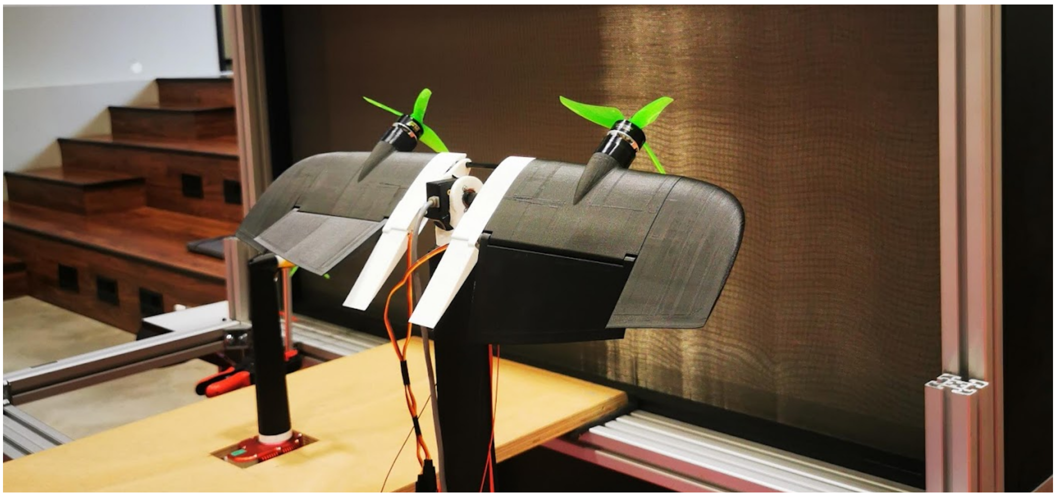

FIGURE 5.2 – Single degree-of-freedom DarkO model in front of the WindShape.

of the WindShape, in the most laminar flow area, while the hot wire sensor is placed as close as possible to the drone.

The geometry of the experimental setup allows placing the power and signal cables close to the centre of rotation so as to minimize their frictional effects on the structure. Despite this fact, the rotation system inevitably interferes with the drone, by creating parasitic forces, notably drag. The projected surface of the joint is small compared to the wing surface, so that the drag generated by this support is low compared to the drag of the wing and the propellers, thus it can be neglected.

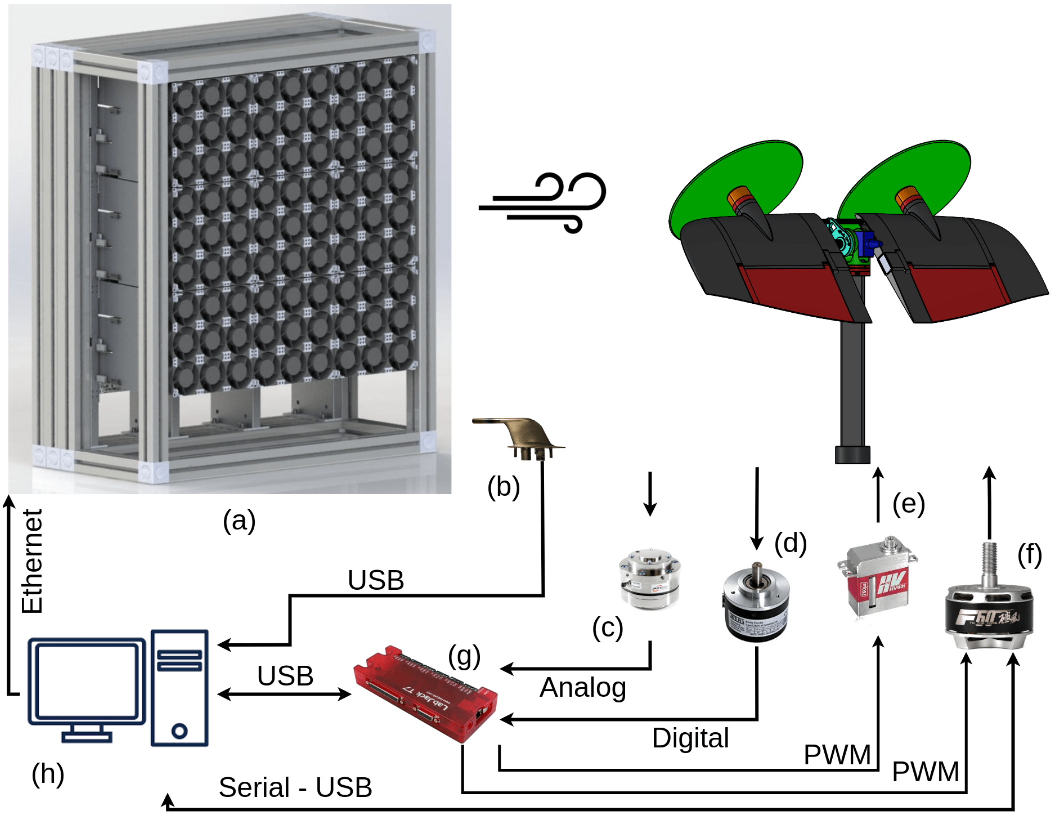

FIGURE 5.3 – Virtual flight testing architecture : WindShape (a) ; Airspeed sensor (b) ; Force/Torque sensor (c) ; Rotary encoder (d) ; Servomotor (e) ; Brussless motor + ESC (f) ; LabJack (g) ; Control computer (h)

A schematic diagram of the functional subcomponents of the experimental device and their interconnection is shown in Fig. 5.3, which is explained below by referring to the various subsystems with their corresponding letter (a)-(i).

The motors (f) are powered by an external 12v 20Ah battery and the servo motors (e) are powered by 5v via a LabJack T7 [?,] acquisition module (g). The LabJack module (g) concentrates most of the sensing/actuating signals : six analog inputs for the force/torque sensor (c), two digital quadrature inputs for the rotary encoder (d), one analog input (or serial link depending on the sensor) for the airspeed sensor (b), two digital PWM (Pulse Width Modulation) outputs for the motors (f) and two digital PWM outputs for the servomotors (e). The elevons are driven by servomotors that do not provide a position measurement signal, therefore

we use the setpoint, assuming a perfect actuator, which is reasonable, due to the software saturation imposed on the elevons commanded input and the correct sizing of the servomotors with respect to the involved forces.
The LabJack (g) has an application programming interface (API), allowing for a remote connection with a computer. We have developed a Python code that communicates with the LabJack in order to retrieve the sensor values, compute the command to be applied to the actuators according to the control scheme presented below, and generate the output signals for the actuators. The data collected from the LabJack is recorded to be used for post-processing and generate the plot reported in Section 8.2.
To generate the wind, we use a WindShape device, which also has an API, allowing it to be controlled through an Ethernet network. The developed Python code can assign the WindShape wind speed and therefore act on the model. It is thus possible to test a set of hovering configurations and their associated transients in the same test campaign, without any action on the model.

5.4.3 Software-in-the-loop translational motion

Since the prototype is connected to the fixed support, it is not possible to experimentally reproduce the translational motion. We have instead included a software-in-the-loop routine that simulates the translational motion by integrating the force measurements available at the joint. In particular, the translational velocity (resp. position) of the UAV is obtained by single (resp. double) integration of the data measured by the force sensor. We neglect the aerodynamic influence of the (simulated) speed on the wing for the sake of simplicity. In particular, from equations (3.8a) and (3.8b), we obtain the simplified model

$$\dot{\boldsymbol{v}} = \boldsymbol{g} + \frac{1}{m}\Big(R(\boldsymbol{q})(F\boldsymbol{u} + D_{\mathrm{f}}(\delta)R^{\top}(\boldsymbol{q})\|\boldsymbol{w}\|\boldsymbol{w})\Big) \tag{5.1a}$$

$$= \boldsymbol{g} + \frac{1}{m}\boldsymbol{F}_{meas}, \tag{5.1b}$$

where $\boldsymbol{F}_{meas}$ represents the forces measured by the sensor in the bias-corrected inertial reference frame. To calibrate the bias correction, at the initialization, the measured forces are averaged over 6000 samples, the model being blocked at a steady position (pitch angle at 0°, namely vertical orientation). Removing the bias from the measured force at each measurement, we subtract the gravity effect on the model from the measurement. An artificial mass m is instead assigned to the software-in-the-loop dynamics according to (5.1b), which allows testing several configurations to better appreciate the influence of the drone's mass on possible transient saturation events. This allows investigating scenarios involving the nontrivial mass of the battery, which is not present in our model. Although this manipulation is easy, it does not represent perfectly the reality because we do not take into account the distribution of the masses in the drone and thus the inertia modifications. The transitional velocity and position of the UAV are then obtained by single and double numerical integration of the acceleration as in (5.1), using a trapezoidal numerical integration.

5.5 Integral-based linear control

In our previous work [**?**, III.B] we proposed a proportional feedback stabilizing a hovering position in the absence of wind (perturbation). We propose here an extension including integral action, suitable for operating with a non-measured perturbation represented by a constant wind. The objective is to stabilize the drone at the reference position, rejecting the unknown constant wind disturbance.

5.5.1 Description of the control scheme

We experiment the situation with the wind only acting along the $x_{[I]}$ axis, with the drone oriented towards the wind, i.e. with zero roll and yaw angles. In this configuration, the wind only acts on the linear velocity along the $x_{[B]}$ and $z_{[B]}$ axes, and it only generates a moment about the $y_{[B]}$ axis. A careful inspection of the control and the disturbance input matrices F, M in (**??**) suggests an effective control architecture to reject a constant disturbance. Indeed, the ailerons and the propellers can be used symmetrically to generate respectively a moment about the $y_{[B]}$ axis and a force along the $x_{[B]}$ axis, thus compensating the disturbance effect. Nevertheless, there is still a force along the $z_{[B]}$ axis to be compensated, and an integral action can asymptotically converge to the desired force, even with a non-measured wind disturbance w. We may thus stabilize the UAV at a hovering position, different from the zero-wind equilibrium. The control solution exploits the pitch angle degree of freedom, for compensating the wind effect.
The proposed controller, shown in Fig. 6.1, corres-

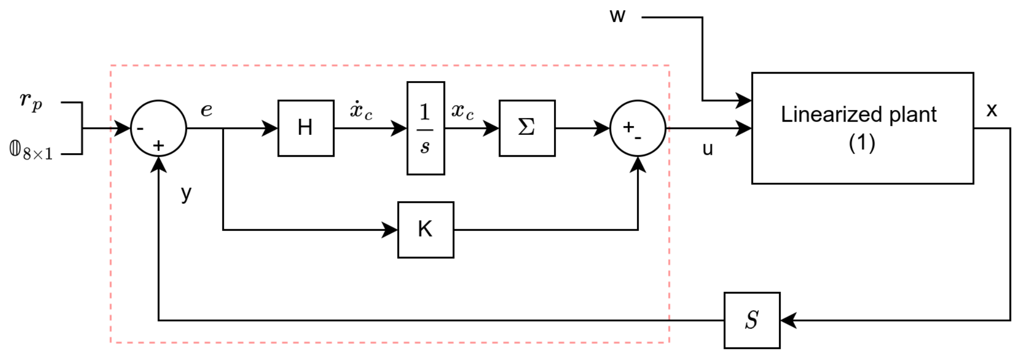

FIGURE 5.4 – Proposed integral-based controller.

ponds to

$$\dot{x_c} = H(y - \begin{bmatrix} r_p \\ \mathbb{0}_{8\times1} \end{bmatrix}), \tag{5.2}$$

$$y = Sx, \tag{5.3}$$

$$u = \Sigma x_c + K(y - \begin{bmatrix} r_p \\ \mathbb{0}_{8\times1} \end{bmatrix}), \tag{5.4}$$

$$S = \begin{bmatrix} \mathbb{I}_7 & \mathbb{0}_{7\times5} \\ \mathbb{0}_{4\times8} & \mathbb{I}_4 \end{bmatrix}, \tag{5.5}$$

$$\Sigma = \begin{bmatrix} 1 & 1 & 0 & 0 \\ 0 & 0 & 1 & 1 \end{bmatrix}^\top, \tag{5.6}$$

where $x_c \in \mathbb{R}^2$ is the integrator state ; $r_p \in \mathbb{R}^3$ is the constant reference comprising a target position for the translational motion ; S is an output selection matrix, which removes the pitch angle component from the measured output (impacting only the quaternion linearization) to form y ; Σ is an input allocation matrix that allows assigning the first component of the integrator state to the motor control and the second component to the elevon control. K, H are constant stabilizing gains to be selected in such a way that the linear closed loop matrix

$$A_{cl} = \begin{bmatrix} A & \mathbb{0}_{12\times2} \\ HS & \mathbb{0}_{2\times2} \end{bmatrix} - \begin{bmatrix} G \\ \mathbb{0}_{2\times4} \end{bmatrix} \left(K \begin{bmatrix} S & \mathbb{0}_{11\times2} \end{bmatrix} - \begin{bmatrix} \mathbb{0}_{4\times12} & \Sigma \end{bmatrix} \right), \tag{5.7}$$

characterizing the linearized closed loop be Hurwitz, to ensure stabilization with the linearized dynamics related to the zero-wind scenario (3.28).

In a nutshell, matrix (5.7), describes the closed loop shown in Fig. 6.1 : an output feedback with 11 outputs, consisting of the three positions, the three linear velocities, two out of three angles (ϕ and ψ) and the three angular velocities. This structure can be seen as a MIMO proportional-integral solution resulting from a careful observation of the UAV linearized dynamics, which allows a minimal number of integrators embedded in the controller. This control should allow constant disturbances rejection while having a satisfactory robustness. The gain K corresponds to the proportional term and the gain H weights the integral term, inducing convergence to the target. The allocation matrix Σ leads to a symmetrical use of the propellers and ailerons. We must then tune K and H to obtain a satisfactory trade-off between robustness and disturbance rejection. We implement a multi-objective synthesis based on an H_∞ optimization method, described next.

5.5.2 H_∞-based optimization

To perform a robust selection of K and H, we first characterize several transfers functions in Fig. 6.1. The measurement output y is used for feedback, the input u is the sum of the integral input Σx_c and the proportional action Ke. The output z corresponds to the output performance signals to control (e, w, u, y, r_p). Thanks to weighting functions $W = \text{diag}(W_1, ..., W_4)$, the design of H and K aims to reject a low frequency perturbation or step w acting on y. In short, the design goal is to bring y to zero despite the low frequency disturbance on w.

From the Nyquist criterion, we know that the margin corresponds to the minimal distance between the singularity (real point -1) and the product between the controller (C) and the plant (P). Consequently, we define the input modulus margin as $MM_u = \min_{\omega \in R} |1 - CP|$ and the output modulus margin as $MM_y = \min_{\omega \in R} |1 - PC|$ for a positive feedback. We first introduce the output sensitivity function $T_{r \to \epsilon} = S_y = (1 - PC)^{-1}$, so that $\|S_y\|_\infty = MM_y^{-1}$ and the input sensitivity function $T_{d \to u} = S_u = (1 - CP)^{-1}$, so that $\|S_u\|_\infty = MM_u^{-1}$. Consequently, the minimization of the H_∞-norm of S_u or S_y, leads to improving the input and output modulus margins. As our system is MIMO, we give importance to both the input and output sensitivity functions, because they do not commute. We also define the transfer functions $T_{r \to u} = CS_y = S_u C$ and $T_{w \to y}$. In order to guarantee a satisfactory trade-off between robustness and performance, we select the weighting functions W_1, W_2, W_3 and W_4 linked to $\|W_1 T_{r \to \epsilon}(s)\|_\infty \leq 1$ and $\|W_2 T_{d \to u}(s)\|_\infty \leq 1$, corresponding to robustness margins at the inputs and outputs, $\|W_3 T_{r \to u}(s)\|_\infty \leq 1$ limiting the control effort, $\|W_4 T_{w \to y}(s)\|_\infty \leq 1$ ensuring suitable wind disturbance rejection. Specifically, the weighting functions are tuned as

$$W_1 = 0.5, \quad W_2 = 0.5, \quad W_3 = 0.8, \qquad\qquad W_4 = 0.5. \tag{5.8}$$

The values of W_1 and W_2 ensure $MM_u > 6\ dB$ and $MM_y > 6\ dB$, W_3 and W_4 are tuned to obtain a satisfactory trade-off between the different specifications. The weight W_4 allows managing, among other things, the speed of the rejection.

With selections (5.8), we cast the design problem for K and H as an H_∞ synthesis under order constraint, providing good input and output specifications for the closed loop :

$$\begin{gathered} \min_{C} \left\| \begin{matrix} W_1 T_{r\to\epsilon}(P,C) \\ W_2 T_{d\to u}(P,C) \\ W_3 T_{r\to u}(P,C) \\ W_4 T_{w\to y}(P,C) \end{matrix} \right\|_\infty, \text{ subject to} \\ C \in \mathbb{R}^{11\times 4} \text{ stabilizes } P \text{ internally,} \end{gathered} \tag{5.9}$$

where P is the augmented plant containing the integral action and the linearized UAV dynamics. In addition, we impose constraints on the gains K and H ensuring that the closed loop with experimental device only evolves in the (x,z) plane, compatibly.

We solved (5.9) using Systune [**?**,]. Based on non-smooth optimization, Systune dealing with several non-convex scenarios, such as the structured control architecture where we optimize the gain matrices K, H. The optimization algorithm returns optimized selections of

$$\left[\begin{array}{c} H \\ \hline K \end{array}\right] = \left[\begin{array}{ccccccccccc} -1.902 & 0 & 7.201 & -9.043 & 0 & 33.244 & 0 & 0 & 0 & 4.696 & 0 \\ 0.425 & 0 & -1.620 & 2.024 & 0 & -7.480 & 0 & 0 & 0 & -1.045 & 0 \\ \hline 0.035 & 0 & -0.728 & -1.853 & 0 & -4.445 & 0 & 0 & 0 & -0.323 & 0 \\ 0.035 & 0 & -0.728 & -1.853 & 0 & -4.445 & 0 & 0 & 0 & -0.323 & 0 \\ 0.217 & 0 & -0.164 & 1.074 & 0 & -0.527 & 0 & 0 & 0 & -0.773 & 0 \\ 0.217 & 0 & -0.164 & 1.074 & 0 & -0.527 & 0 & 0 & 0 & -0.773 & 0 \end{array}\right] \tag{5.10}$$

Introducing a closed-loop spectral abscissa $\alpha = -0.2381$ for A_{cl} in (5.7).

5.6 Résultats

Chapitre 6

Commande proportionnelle intégrale d'un drone convertible à 6 degrés de liberté

Sommaire

6.1 Schéma de commande linéaire proportionnel intégral : 6 Dof 51

6.2 Integral-based linear control 51

6.2.1 Description of the control scheme 51

6.2.2 Zero-wind H_∞-based controller tuning 54

6.2.3 Multimodel H_∞-based controller tuning 56

6.3 Experimental flight with open wind tunnel 59

6.4 Maquette expérimentale : 6 Dof 60

6.5 Résultats 60

6.1 Schéma de commande linéaire proportionnel intégral : 6 Dof

6.2 Integral-based linear control

6.2.1 Description of the control scheme

A careful inspection of the control and the disturbance input matrices $\boldsymbol{G}_w$ and $\boldsymbol{E}_w$ in model (3.31) (see the output of Algorithm 2) suggests an effective control architecture to reject a constant wind disturbance $\boldsymbol{w}$. Indeed, the ailerons and the propellers can be used symmetrically to generate respectively a moment about the $y_{[\mathrm{b}]}$ axis, verifying equation (3.26) and a force along the $x_{[\mathrm{b}]}$ axis, verifying equation (3.24), thus compensating for the disturbance effect. Nevertheless, there is still a force along the $z_{[\mathrm{b}]}$ axis to be compensated for by verifying equation (3.25), and an integral action can asymptotically converge to the desired force, even with a

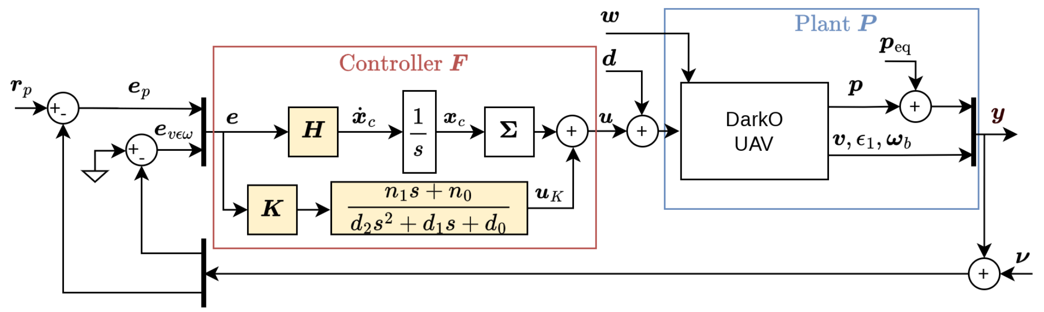

FIGURE 6.1 – Proposed integral-based controller with the wind perturbation $\boldsymbol{w}$, a plant-input perturbation $\boldsymbol{d}$ and a plant-output perturbation ν.

non-measured wind disturbance $\boldsymbol{w}$. We may thus stabilize the UAV at a hovering equilibrium as characterized in Theorem 1. Since we don't measure the wind $\boldsymbol{w}$, the values of ψ and θ in Algorithm 1 are unknown. The proposed controller, shown in Fig. 6.1, uses integral action to obtain these two unknown angles. Its feedback loop involves the following error variables output, which should converge to zero in any hovering position :

$$\boldsymbol{e}_p = \boldsymbol{r}_p - \boldsymbol{p}, \ \boldsymbol{e}_{v\epsilon\omega} = -\begin{bmatrix} \mathbb{I}_3 & \mathbb{0}_{3\times1} & \mathbb{0}_{3\times2} & \mathbb{0}_3 \\ \mathbb{0}_{1\times3} & 1 & \mathbb{0}_{1\times2} & \mathbb{0}_{1\times3} \\ \mathbb{0}_3 & \mathbb{0}_{3\times1} & \mathbb{0}_{3\times2} & \mathbb{I}_3 \end{bmatrix} \begin{bmatrix} \tilde{\boldsymbol{v}} \\ \tilde{\boldsymbol{\epsilon}} \\ \tilde{\boldsymbol{\omega}}_{\text{b}} \end{bmatrix}, \tag{6.1}$$

where $\boldsymbol{r}_p \in \mathbb{R}^3$ is the constant position reference comprising a target position for the translational motion (note that $\boldsymbol{r}_p$ is the reference input to the control scheme).

The error variables in (6.1) can be represented as in the block diagram of Fig. 6.1 by defining the output $\boldsymbol{y} \in \mathbb{R}^{10}$ of the linearized plant dynamics (3.31), having the incremental state vector $\tilde{\boldsymbol{x}} \in \mathbb{R}^{10\times1}$, as follows

$$\boldsymbol{y} = \boldsymbol{C}\tilde{\boldsymbol{x}} + \begin{bmatrix} \boldsymbol{p}_{\text{eq}} \\ \mathbb{0}_{7\times1} \end{bmatrix}, \quad \boldsymbol{C} := \begin{bmatrix} \mathbb{I}_6 & \mathbb{0}_{6\times1} & \mathbb{0}_{6\times2} & \mathbb{0}_{6\times3} \\ \mathbb{0}_{1\times6} & 1 & \mathbb{0}_{1\times2} & \mathbb{0}_{1\times3} \\ \mathbb{0}_{3\times6} & \mathbb{0}_{3\times1} & \mathbb{0}_{3\times2} & \mathbb{I}_3 \end{bmatrix}, \tag{6.2}$$

where the output matrix $\boldsymbol{C} \in \mathbb{R}^{10\times12}$ removes the $\tilde{\epsilon}_2$ and $\tilde{\epsilon}_3$ components from the state vector $\tilde{\boldsymbol{x}}$.

As shown in Fig. 6.1, the controller dynamic equations are based on the measured error $\boldsymbol{e}$ as follows

$$\boldsymbol{e} = \left[e_p^\top \ e_{v\epsilon\omega}^\top \right]^\top, \quad \dot{\boldsymbol{x}}_c = \boldsymbol{H}\boldsymbol{e}, \quad \boldsymbol{u} = \boldsymbol{\Sigma}\boldsymbol{x}_c + \boldsymbol{u}_K,$$
$$\boldsymbol{\Sigma} := \begin{bmatrix} 1 & 1 & 0 & 0 \\ 0 & 0 & 1 & 1 \end{bmatrix}^\top, \quad \boldsymbol{u}_K = \frac{n_1 s + n_0}{d_2 s^2 + d_1 s + d_0}\boldsymbol{K}\boldsymbol{e}, \tag{6.3}$$

where $\boldsymbol{x}_c \in \mathbb{R}^2$ is the integral action state ; $\boldsymbol{\Sigma}$ is an input allocation matrix that allows assigning the first component of the integrator state to the propellers action and the second component to the elevons action. Scalars n_1, n_0, d_2, d_1, d_0 are respectively the numerator and denominator coefficients of a filter used to avoid a direct input-output transmission that would amplify high-frequency measurement noise. This filter induces a strictly proper controller, for increased robustness to additive uncertainties. We define the controller $\boldsymbol{F}$ having dimensions 4×10 having transfer matrix $\boldsymbol{F}(s) = T_{\boldsymbol{e}\to\boldsymbol{u}}(s)$ as described in (6.3) and interconected as in Fig. 6.1. The plant $\boldsymbol{P}$ having dimensions 10×4 represents the linearized DarkO dynamics. The output of the plant $\boldsymbol{y} \in \mathbb{R}^{10\times1}$ is used as the input of controller $\boldsymbol{F}$.

In view of the symmetries of the actuators on the UAV, we have constrained the structure of matrix $\boldsymbol{K}$ in (6.3), associated with the controller's proportional action, in order to use the actuators in a physically meaningful way as follows :

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

In particular, a position error on the $z_{[\mathrm{i}]}$ axis of the NED world frame (see Fig. 3.2) results in a symmetric use of the two propellers that generates a force along the $x_{[\mathrm{b}]}$ axis of the UAV. The symmetric use of the two motors is reflected by the same-sign in coefficients k_3 and k_6 on columns 3 and 6 of $\boldsymbol{K}$, corresponding respectively to the position and velocity errors on the $z_{[\mathrm{i}]}$ axis. Similarly, a position or speed error along the drone's lateral axis $y_{[\mathrm{b}]}$ will be compensated for by an antisymmetric use of the motors, as reflected by the coefficients k_2 and k_5 and their opposite signs on columns 2 and 5 of $\boldsymbol{K}$. An angular velocity error about the $x_{[\mathrm{b}]}$ axis must be compensated for by an antisymmetric use of the elevons, as reflected by coefficient k_{18} having opposite signs on column 8 of $\boldsymbol{K}$. Parallel arguments explain the remaining coefficients of matrix $\boldsymbol{K}$ in (6.4). An advantage of the structure in (6.4) is the reduction of the number of variables to be optimized, from 40 to 20 scalar gains.

The closed loop shown in Fig. 6.1, is an output feedback with 10 outputs, consisting of the three linear positions, the three linear velocities, one of the three attitude angles (ϵ_1) and the three angular velocities. This structure can be seen as a MIMO proportional-integral feedback. The parameters to be tuned in controller $\boldsymbol{F}$ (6.3) are the proportional gain $\boldsymbol{K} \in \mathbb{R}^{4\times10}$ in (6.4), the integral gain $\boldsymbol{H} \in \mathbb{R}^{2\times10}$ and the filter parameters n_1, n_0, d_2, d_1, d_0, as highlighted in yellow in Fig. 6.1. A suitable tuning method should ensure desirable disturbance rejection and satisfactory robustness to unmodeled dynamics. These two goals lead to a trade-off because disturbance rejection requires an aggressive tuning while robustness properties are ensured by a frequency roll-off strategy. We discuss next two optimization-based tuning methods. The first one is issued from the ideas proposed in [?], which did not need the linearized dynamics of Theorems 1 and 2, and is summarized in Section 6.2.2. It is a multi-objective synthesis with H_∞ constraints based on the zero-wind model discussed in Remarks **??** and **??** and derived in [?]. We will show that this first method fails to stabilize the drone in certain wind ranges, due to the lack of knowledge of the dynamics characterized in Theorems 1 ans 2. The second tuning method, presented in Sec.6.2.3, is an iterative multi-objective synthesis with H_∞ constraints, based on a collection of models associated with different wind conditions and derived based on Theorems 1 and 2, through Algorithms 1 and 2. In our numerical valida-

Measurement	Value	Units
$\boldsymbol{p}$	2.5×10^{-4}	m
$\tilde{\boldsymbol{v}}$	1.2×10^{-3}	$\mathrm{m\,s^{-1}}$
$\tilde{\boldsymbol{\epsilon}}$	4.7×10^{-4}	
$\tilde{\boldsymbol{\omega}}_{\mathrm{b}}$	2.7×10^{-3}	$\mathrm{rad\,s^{-1}}$

Table 6.1 – Standard deviation of the modeled sensor noise added to the simulated measurements.

tion, reported in Sections 6.2.2 and 6.2.3 (see in particular Fig. 6.2 and Fig. 6.4), measurement noise is added to the output to produce practically reasonable numerical results. The standard deviations of the adopted noise levels are reported in Table 6.1. Moreover, in addition to reporting the simulation results of the linear feedback of Fig. 6.1 with the linearized model (3.31), in Sections 6.2.2 and 6.2.3, we also simulate the closed loop by replacing the linearized plant $\boldsymbol{P}$ with the nonlinear model (3.1) including many real-world effects. When replacing the linearized plant with the nonlinear dynamics (3.1), whose state is $\boldsymbol{x} = (\boldsymbol{p}, \boldsymbol{v}, \boldsymbol{q}, \boldsymbol{\omega}_{\mathrm{b}}) \in \mathbb{R}^{13}$, we replace the linear output $\boldsymbol{y}$ with the following surrogate nonlinear version

$$\boldsymbol{y}_{\mathrm{NL}} = \begin{bmatrix} \boldsymbol{p} \\ \boldsymbol{v} \\ \epsilon_1 \\ \boldsymbol{\omega}_{\mathrm{b}} \end{bmatrix} = \begin{bmatrix} \mathbb{I}_6 & \mathbb{0}_{6\times 1} & \mathbb{0}_{6\times 1} & \mathbb{0}_{6\times 2} & \mathbb{0}_3 \\ \mathbb{0}_{1\times 3} & 0 & 1 & \mathbb{0}_{1\times 2} & \mathbb{0}_{1\times 3} \\ \mathbb{0}_3 & \mathbb{0}_{3\times 1} & \mathbb{0}_{3\times 1} & \mathbb{0}_{3\times 2} & \mathbb{I}_3 \end{bmatrix} \begin{bmatrix} \boldsymbol{R}_\psi^\top \boldsymbol{p} \\ \boldsymbol{R}_\psi^\top \boldsymbol{v} \\ \boldsymbol{q}_{\mathrm{eq}\psi}^{-1} \otimes \boldsymbol{q} \\ \boldsymbol{\omega}_{\mathrm{b}} \end{bmatrix}. \tag{6.5}$$

In the next sections we denote the modulus margin of a transfer matrix $s \mapsto T_{v\to z}$ as $\Delta_m(T_{v\to z}) = \min_{\omega \in R} \sigma_{\min}(T_{v\to z}(j\omega))$.

6.2.2 Zero-wind H_∞-based controller tuning

For tuning the controller based on the zero wind, we use the linear plant model detailed in Remark **??**, $\boldsymbol{P}(s) = T_{\boldsymbol{u}\to\boldsymbol{y}}(s)$, obtained from equations (3.28) and (6.2) as

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

With reference to Fig. 6.1, we introduce transfer matrices that correspond to robustness objectives : the output sensitivity function as $T_{\nu\to e} = (\mathbb{I}_{10} + \boldsymbol{PF})^{-1}$ having dimensions 10×10, so that, $\|T_{\nu\to e}\|_\infty = \Delta_m(T_{\nu\to e})^{-1}$ and the input sensitivity function $T_{d\to u} = (\mathbb{I}_4 + \boldsymbol{FP})^{-1}$ having dimensions 4×4, so that $\|T_{d\to u}\|_\infty = \Delta_m(T_{d\to u})^{-1}$. Consequently, the minimization of the H_∞-norm of $T_{\nu\to e}$ or $T_{d\to u}$, corresponds to increasing the input and output modulus margins. Since plant $\boldsymbol{P}$ is MIMO, we give importance to both the input and the output sensitivity functions which do not coincide, because $\boldsymbol{P}$ and $\boldsymbol{F}$ do not commute. We also define the transfer matrix $T_{\nu\to u}$ having dimensions 4×10 linked to the impact of the measurement noise ν on the control input $\boldsymbol{u}$, and $T_{d\to y}$ having dimensions 10×4 representing the impact of the input disturbance $\boldsymbol{d}$ on the plant output $\boldsymbol{y}$. We solve the same problem as in our preliminary work [**?**, eqn. (13)] using the `Systune` software [**?**], however we use the control diagram presented in section 6.2.1 which includes a filter on the proportional action and a different number of outputs. We also include in the plant $\boldsymbol{P}$ the linear actuators dynamics discussed in Remark **??**.

Successive steps of increasing horizontal and vertical wind intensity (ranging from zero to $-6\,\mathrm{m\,s^{-1}}$) are applied, as shown in the lower plot of Fig. 6.2. The selected wind pairs (w_x, w_z) are represented by red dots on the surfaces in Fig. 3.6, where we can see that the equilibrium $(\boldsymbol{u}_{\mathrm{eq}}, \boldsymbol{x}_{\mathrm{eq}})$ is reached without saturating the actuators. We only focus on the negative part of the vertical wind speed because it is the most limiting one. Indeed, the drone is lifted by the rising vertical wind (whose sign is negative in the NED frame), so it needs less traction on the propellers to compensate for the gravity. The motors generate less airflow over the elevons, which reduces their efficiency, leads to saturation, and destabilizes the drone. The aim of the control system is to keep the UAV at the hovering position (defined as $\boldsymbol{r}_p = [0, 0, 0]^\top$), despite the increasing horizontal and vertical wind w_x and w_z. Fig. 6.2 both linear simulations with the linearized plant dynamics (3.31) (dashed) and nonlinear simulations with the accurate model (3.1) (solid). Both the linear and nonlinear simulations consistently show that the controller performs well at low wind speed (in fact, the tuning is performed based on the zero-wind model). However, when the wind speed w_x and w_z exceed $-5\,\mathrm{m\,s^{-1}}$, the hovering position becomes unstable and the drone oscillates and diverges. The tilt angles θ are used to represent the attitude to give a better insight of the vehicle behavior, however the nonlinear simulation of the nonlinear dynamics (3.1) is carried out with unit quaternions. The instability observed in the simulation results of Fig. 6.2 confirms the experimental instabilities reported in [?] where we used this same tuning method, and confirms the importance of Theorems 1 and 2 in Section 7.2, for an appropriate tuning of the controller gains, which is performed in the next section.

6.2.3 Multimodel H_∞-based controller tuning

The simulation results obtained with the zero-wind tuning method (see Fig. 6.2) together with the experimental instabilities observed in [?] confirm the need for a controller gain tuning procedure exploiting the parametrized non-zero wind linearizations of Theorems 1 and 2. Focusing again on the control scheme of Fig. 6.1, we now explicitly consider the (linearized) wind effect on the plant, and we consider the linearized plant dynamics (3.31) with output (6.2) and with the selections in Algorithm 2 as

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

whose input is the concatenation of the control input $\boldsymbol{u}$ and the wind disturbance input $\boldsymbol{w}$. As the model depends on the wind speed $\boldsymbol{w}$, we introduce a new transfer matrix $T_{w\to y}$ having dimensions 10×3, which corresponds to the transfer matrix between the wind input $\boldsymbol{w}$ and the plant output $\boldsymbol{y}$, quantifying the effect of the wind disturbance on the UAV feedback loop. With the set of transfers matrices defined in Sec. 6.2.2 and the new transfer matrix $T_{w\to y}$, we use the algorithmic approach in [?, ?], named "`systune`", which uses non-smooth optimization techniques to deal with non-convex tuning problems, such as our structured control architecture where we optimize the gain matrices $\boldsymbol{K}$, $\boldsymbol{H}$ and the filter parameters n_1, n_0, d_2, d_1, d_0 (in yellow on the figure 6.1). As reported in [?, eq. (2)], we solve a multi-objective optimization problem, by exploiting the Matlab implementation well explained in [?, §3]. In particular, based on a set $\mathcal{W}$ comprising a finite collection of pairs (w_x, w_z), with $w_x \in [0,\ 8]$ m s^{-1} and $w_z \in [-4,\ 4]$ m s^{-1}, we consider the ensuing set of linearized plants (6.6) and solve the following convex optimization, where scalars W_1, W_2, W_3, W_4 and W_5 are weighting factors to be tuned to obtain a satisfactory trade-off between robustness (associated with W_2, W_3 and W_4) and performance

(associated with W_1 and W_5)

$$\gamma^\star = \min_{\boldsymbol{F}} \max_{w\in\mathcal{W}} \left| \begin{matrix} \|W_1 T_{\nu\to e}(\boldsymbol{P}_w, \boldsymbol{F})\|_\infty \\ \|W_2 T_{d\to u}(\boldsymbol{P}_w, \boldsymbol{F})\|_\infty \\ \|W_3 T_{\nu\to u}(\boldsymbol{P}_w, \boldsymbol{F})\|_\infty \\ \|W_4 T_{d\to y}(\boldsymbol{P}_w, \boldsymbol{F})\|_\infty \\ \|W_5 T_{w\to y}(\boldsymbol{P}_w, \boldsymbol{F})\|_\infty \end{matrix} \right|_\infty, \text{ subject to} \tag{6.7}$$
$$\boldsymbol{F} \text{ stabilizes internally } \mathcal{F}_\ell(\boldsymbol{P}_w, \boldsymbol{F}), \forall w \in \mathcal{W},$$

where $\mathcal{F}_\ell(\boldsymbol{P}_w, \boldsymbol{F})$ denotes the linear feedback interconnection of Fig. 6.1 for a specific value of w (this is consistent with the classical robust control notation [**?**, **?**]). Notice that, with reference to [**?**, eq. (2)], we only specify soft constraints and we do not specify any hard constraint.

Algorithme 3 Iterative multimodel controller gain tuning.

Input : $\boldsymbol{A}_w, \boldsymbol{G}_w, \boldsymbol{E}_w$ the output matrices of Algorithm 2 and the positive weighting scalars W_1–W_5

Output : $\boldsymbol{K}$, $\boldsymbol{H}$ and the filter gains

1: (Initialization) Initialize $\mathcal{W}$ as a grid comprising all the pairs $w_x \in \{0,\ -4,\ -8\}$ and $w_z \in \{$-4, 0, 4$\}$

2: (Synthesis) Solve the optimization (6.7) with the software `systune`

3: (Analysis) Define a validation grid $\mathcal{W}_\text{v}$ by discretizing the interval $(w_x, w_y) \in [0, 8] \times [-4, 4]$ with a discretization step of 1 and using the controller $\boldsymbol{F}$ obtained from the previous step, compute, for each $w_\text{v} \in \mathcal{W}_\text{v}$,

$$\gamma_\text{v} = \left| \begin{matrix} \|W_1 T_{\nu\to e}(\boldsymbol{P}_{w_\text{v}}, \boldsymbol{F})\|_\infty \\ \|W_2 T_{d\to u}(\boldsymbol{P}_{w_\text{v}}, \boldsymbol{F})\|_\infty \\ \|W_3 T_{\nu\to u}(\boldsymbol{P}_{w_\text{v}}, \boldsymbol{F})\|_\infty \\ \|W_4 T_{d\to y}(\boldsymbol{P}_{w_\text{v}}, \boldsymbol{F})\|_\infty \\ \|W_5 T_{w\to y}(\boldsymbol{P}_{w_\text{v}}, \boldsymbol{F})\|_\infty \end{matrix} \right|_\infty, \tag{6.8}$$

and augment $\mathcal{W}$ with the corresponding point if $\gamma_\text{v} > 1$ or γ_v is undefined (namely if $\boldsymbol{F}$ is not internally stabilizing).

4: (Termination) If $\mathcal{W}$ has not been augmented at the previous step, then move to step 5, otherwise move to step 2.

5: **Return :** $\boldsymbol{K}$, $\boldsymbol{H}$ and filter parameters n_1, n_0, d_2, d_1, d_0

The optimization problem (6.7) becomes increasingly cumbersome, from a computational viewpoint, as we increase the cardinality of the set of wind conditions considered in $\mathcal{W}$. In fact, a brute force approach including a fine grid of points in $\mathcal{W}$ leads to a computationally intractable optimization. Instead, we follow here the iterative procedure overviewed in Algorithm 3, where $\mathcal{W}$ is initially selected as a sparse grid comprising $3 \times 3 = 9$ points (step 1) and then a synthesis step (step 2)

Weighting scalars	W_1	W_2	W_3	W_4	W_5
Values	18	16	11	26	5

TABLE 6.2 – Values of the positive weighting scalars W_1–W_5 used in the execution of Algorithm 3.

is repeatedly followed by a (computationally simple) analysis step (step 3) where controller $\boldsymbol{F}$ is fixed. Step 3 identifies the violating points by using a finer validation grid $\mathcal{W}_{\mathrm{v}}$ and adds them to the optimization set $\mathcal{W}$. The algorithm terminates after some iterations, when no points of the validation grid violate the constraints.

Executing Algorithm 3 for the DarkO models of Theorems 1 and 2 with the selection of the positive weighting scalars W_1–W_5 reported in Table 6.2.3, returned the following selection after 2 iterations :

$$\left[\boldsymbol{K}^\top \middle| \boldsymbol{H}^\top\right] = \left[\begin{array}{cccc|cc} -3.86 & -3.86 & 0.79 & 0.79 & 0.02 & 0.48 \\ 1.43 & -1.43 & 1.71 & -1.71 & -0.47 & -1.63 \\ 4.06 & 4.06 & -2.07 & -2.07 & -0.45 & 0.52 \\ -6.86 & -6.86 & -11.60 & -11.60 & -0.14 & 1.40 \\ -10.75 & 10.75 & -1.89 & 1.89 & 3.35 & 5.69 \\ 27.20 & 27.20 & -4.29 & 4.29 & -1.84 & 3.79 \\ -12.32 & 12.32 & -3.46 & 3.46 & 3.72 & 6.81 \\ -5.84 & 5.84 & -2.29 & 2.29 & 1.58 & 3.13 \\ -5.19 & 5.19 & 5.79 & 5.79 & 2.86 & -1.54 \\ -6.52 & 6.52 & 0.08 & -0.08 & 0.08 & 2.82 \end{array}\right],$$

$$\left[\begin{array}{c|c} n_1 & n_0 \\ \hline d_2 & d_1 \\ \hline d_0 & \end{array}\right] = \left[\begin{array}{c|c} -429 & -389 \\ \hline 1 & 6475 \\ \hline 4905 & \end{array}\right], \tag{6.9}$$

For the first iteration of Algorithm 3, after a candidate controller $\boldsymbol{F}$ has been evaluated at step 2, Fig. 6.3 shows in blue the bode diagrams of the maximum singular values of $T_{\nu\to e}$, $T_{d\to u}$, $T_{\nu\to u}$, $T_{d\to y}$, and $T_{w\to y}$ (associated with the value of γ_{v}) reported in (6.8) at the analysis step 3, to be compared to the inverse of the five weights W_1–W_5, represented by the green horizontal lines. The diagrams in red correspond to the points that violate the constraints and that are added to the set $\mathcal{W}$ for the next iteration. The few diagrams in magenta, instead, correspond to the 9 points considered in $\mathcal{W}$ for the first iteration of the synthesis step 2. The red diagrams in Fig. 6.3 clearly illustrate that the iterative algorithm manages to detect the critical values of wind speed (w_x, w_z) to be added to the optimization set $\mathcal{W}$.

The singular values of the output and the input sensitivity function (respectively $T_{r\to e}$ and $T_{d\to u}$) are shown in Fig. 6.3 top line. The graph in the third line represents the singular value of the transfer between the wind disturbance $\boldsymbol{w}$ and the drone output $\boldsymbol{y}$. The singular value tangent to the constraint is that for the highest wind condition a.g. $(w_x, w_z) = (-8, -4)\ \mathrm{m\,s^{-1}}$.

With the tuning reported in (6.9), as obtained with Algorithm 3, we report in

Fig. 6.4 parallel simulation results to those already shown in Fig. 6.2 for the zero-wind tuning method discussed in Section 6.2.2. Once again we simulate both the nonlinear plant (3.1) (solid lines) and the linearized plant (3.31) (dashed line). As compared to Fig. 6.2, the simulations of Fig. 6.4 show that the controller tuning based on Theorems 1 and 2 solves the instability issues and manages to stabilize the hovering condition in all of the considered wind scenarios. We also note from Fig. 6.4 shows a more aggressive action, indeed the control input u (both thrust and deflections) is more affected by the measurement noise. The effectiveness of the control scheme tuned on the basis of Algorithm 3 is also confirmed by the experimental results reported in the next section.

6.3 Experimental flight with open wind tunnel

DarkO's experimental flight took place in a dedicated space (see Fig. 6.5) with an Optitrack localization system based on a NED convention as per Figure 3.2. We used an open-vein wind generator to obtain wind steps that we measured with a hot-wire probe (the vertical bar in Fig. 6.5). Although this wind information is recorded on board the drone to synchronize the data, we do not use this measurement in the control law. The measurement frequency of this wind probe is only 0.5 Hz, so we only have one measurement every two seconds. The state estimation is carried out using an inertial navigation system to merge the Inertial Measurement Unit (IMU) + Optitrack sensor data in order to obtain an accurate estimation of the output $\boldsymbol{y}$ in Fig. 6.1. However, the drone's angular velocity $\boldsymbol{\omega}_{\mathrm{b}}$ is measured based on the IMU's gyrometer, which provides noisy measurements, therefore we added a second order Butterworth low-pass filter with cut-off frequency of 20 Hz to smoothen out the output $\boldsymbol{\omega}_{\mathrm{b}}$. The Butterworth filter is considered in the linearized dynamics when optimizing the controller gains following Algorithm 3.

comparaison lineaire

We also used the ESCs associated with the performance shown in Figure **??** for the propellers actuation. The two ESCs were flashed with the open-source code available in the GitHub repository AM32-MultiRotor-ESC-firmware [1]. The advantage of this firmware, as compared with the commercial code, is that it exploits a low-level PID feedback of the speed of rotation of the motor, which is calculated at the same speed as the motor phase commutation. We adapted the speed loop code in the firmware, following the approach of [**?**], featuring an adaptive bias and adaptive gain algorithm (ABAG). In this way, we compensate the battery discharge effects and obtain an accurate realization of the commanded speed. Before this modification, the integral action of the stabilizing feedback of Fig. 6.1 compensated for the

1. `https://github.com/FlorianSan/AM32-MultiRotor-ESC-firmware`

motor speed loss caused by the battery voltage reduction during flight. This integral compensation was indirectly generated by the altitude loss of the UAV caused by the reduced traction. The advantages of the ABAG solution are high responsiveness and adaptability, as the propeller dimensions can be changed without needing to modify the actuation gains.

We carried out a flight experiment where DarkO was manually put into a stabilized hovering mode in front of the wind tunnel, then we switched on the control law of Algorithm 3. As the drone had to be stabilized at least 30 cm away from the wind tunnel, a manual command was gradually applied to avoid overshooting, which could damage the wind tunnel. Once DarkO was close enough to the set-point $\boldsymbol{r}_p$ of Fig. 6.1, we switched on the proposed controller, obtaining the results in Fig. 6.6. During the follow-up experimentation phase, as shown in the lower plot of Fig. 6.6, we stepwise increased the wind speed, waiting 20 seconds between each pair of consecutive steps, up to a final wind speed of $7\,\mathrm{m\,s^{-1}}$.

Figs 6.6 and 6.7 show that the drone maintains its position despite the increasing wind speed. We can note a few important points, in agreement with the simulations : the motor traction decreases when increasing the wind speed. The control scheme takes advantage of the lift generated by the wind to support the drone, so that less energy is needed to stabilize the hovering position. The drone maintains its tilt angle at a value that is unknown a priori to the control law and naturally stems from the integral action that asymptotically attains the required value of the drone's pitch angle θ. To stabilize the position, the UAV uses the elevons to cancel the pitch moment generated by the shape of the wing, subjected to a horizontal wind, without reaching the saturation limits. We also notice a slight asymmetry of the effectiveness of the actuators, which is effectively compensated by the proportional action of the control scheme.

6.4 Maquette expérimentale : 6 Dof

6.5 Résultats

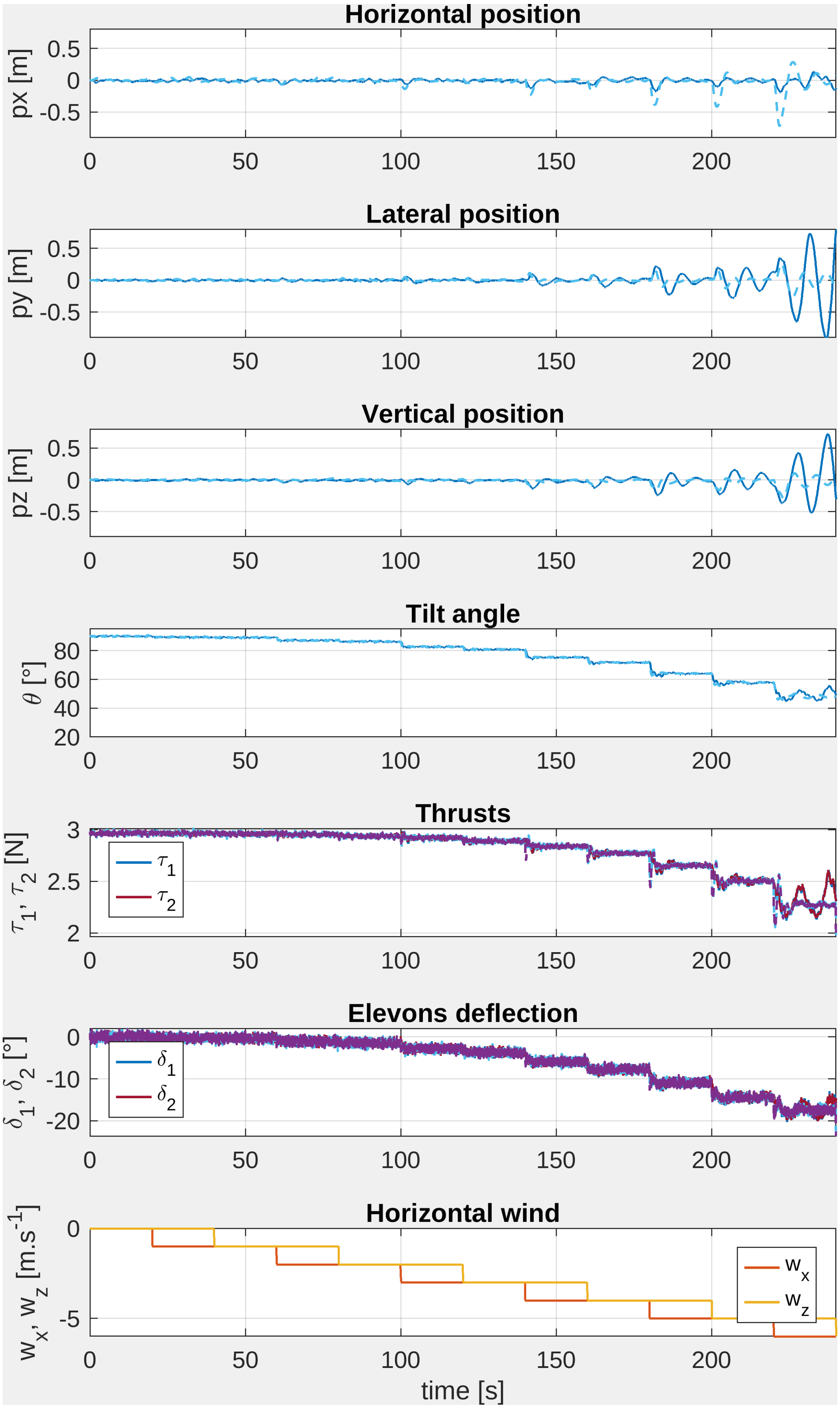

FIGURE 6.2 – Simulation of the non-linear model (3.1) (solid line) and the linearized model (3.31) (dashed line) with increasing constant wind steps with the controller tuned using the zero-wind optimization of Section 6.2.2.

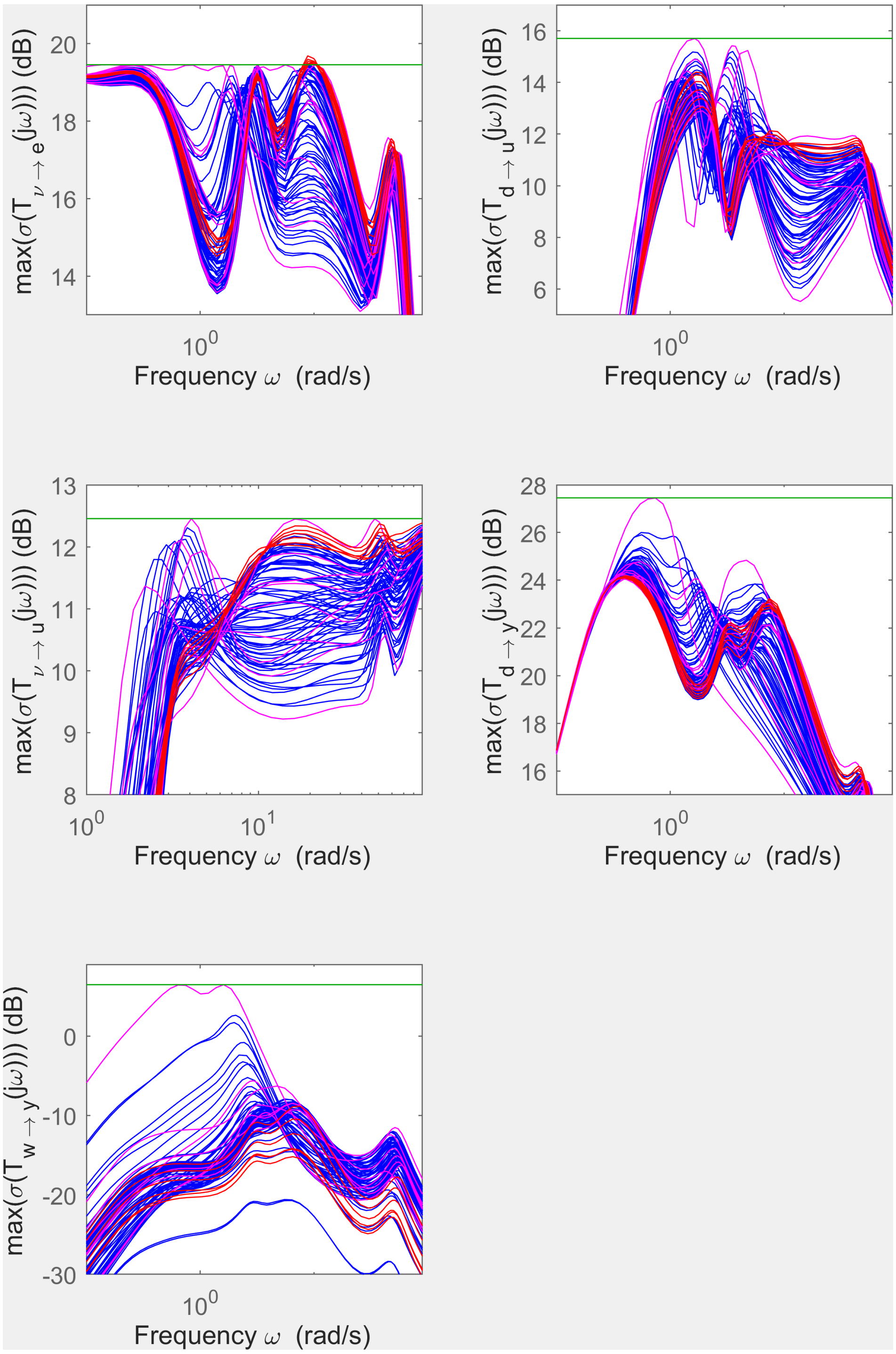

FIGURE 6.3 – Diagrams of the singular values of the transfer functions in (6.8) at the first iteration of Algorithm 3.

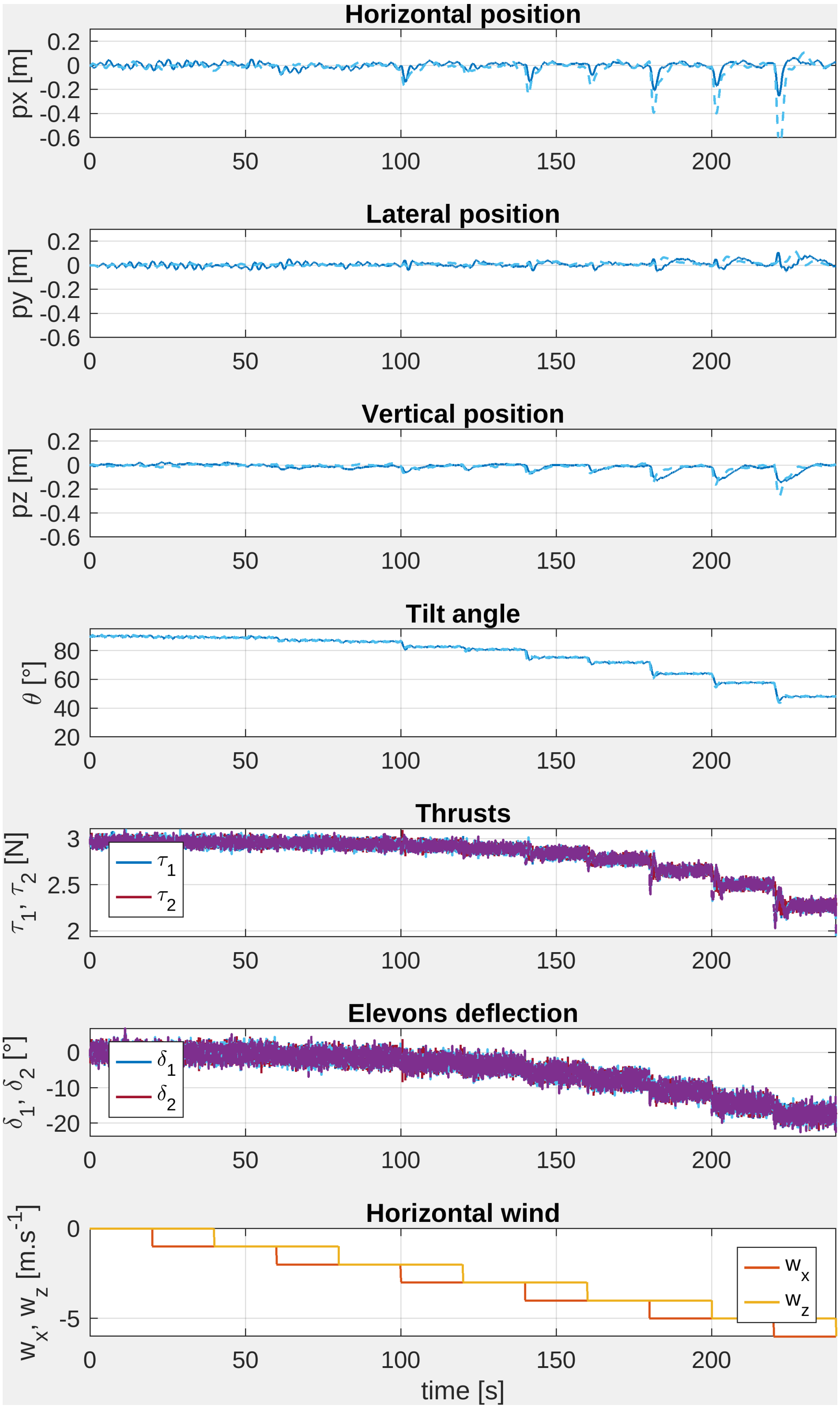

FIGURE 6.4 – Simulation of the non-linear model (3.1) (solid line) and the linearized model (3.31) (dashed line) with increasing constant wind steps with the controller tuned using the multimodel optimization of Algorithm 3 in Section 6.2.3.

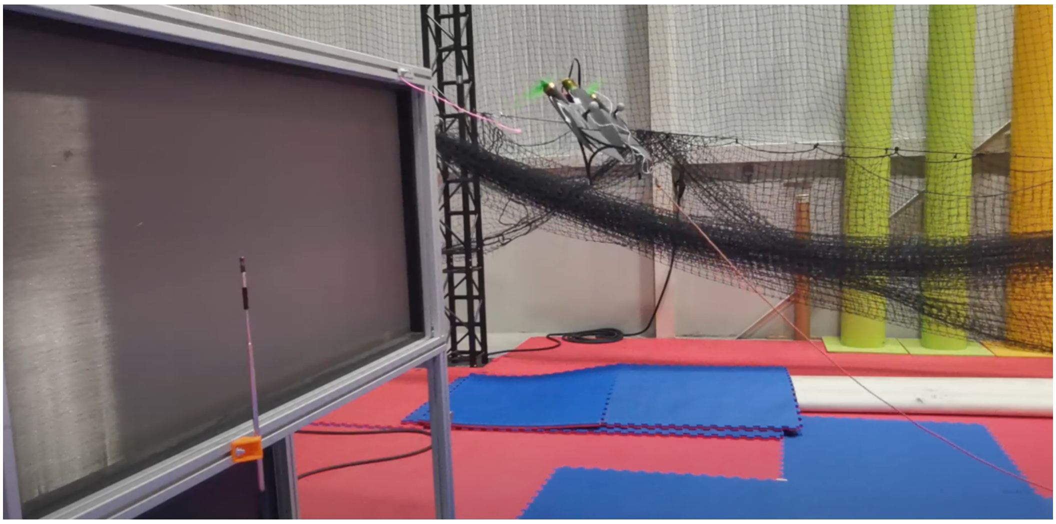

Figure 6.5 – DarkO's experimental flight in front of the open wind tunnel.

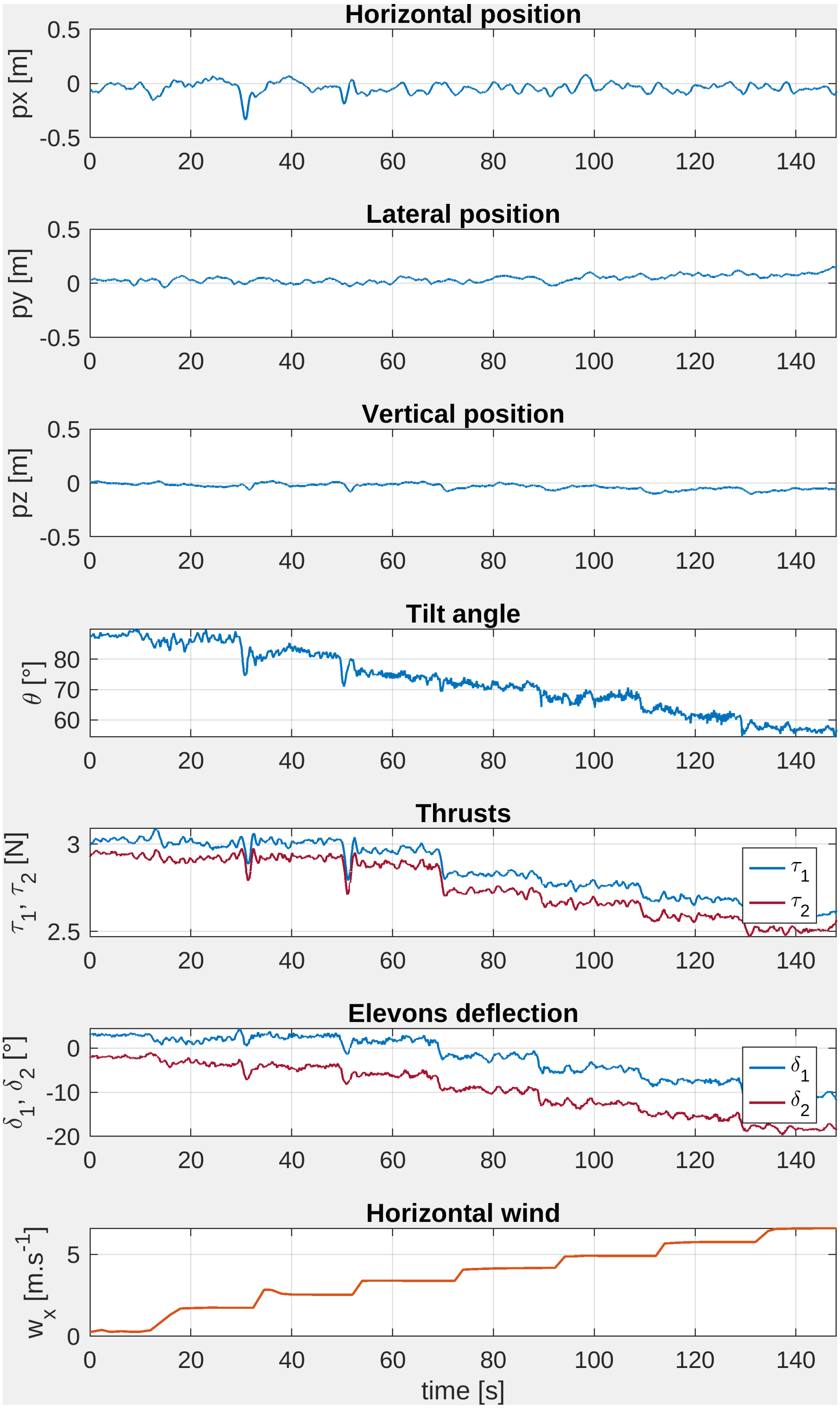

FIGURE 6.6 – Experiment of the DarkO UAV in front of the wind tunnel with increasing constant wind levels (lower plot).

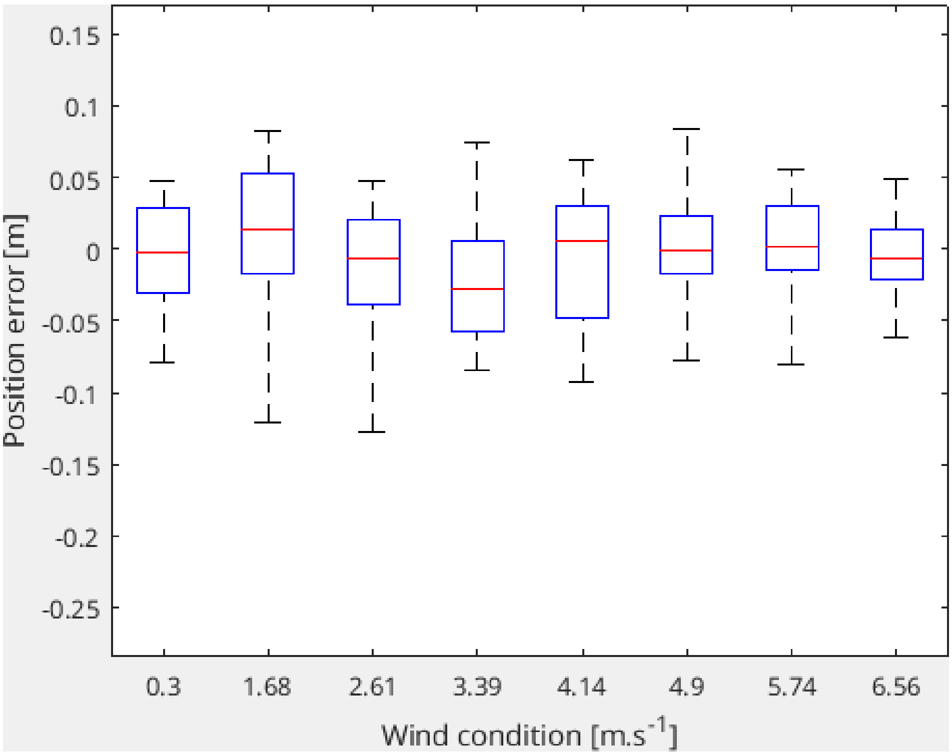

FIGURE 6.7 – Statistical visualization of the hovering performance.

Chapitre 7

Modélisation d'un drone à aile libre rotation libre

Sommaire

7.1 Design et modélisation d'un drone : Colibri 67
7.2 Design and modelling of Colibri UAV 67
7.3 State estimation 72
7.3.1 Sensors placement 72
7.3.2 Angular speed estimation 73
7.3.3 Wing state estimation 74
7.3.4 Fuselage orientation estimation 76

7.1 Design et modélisation d'un drone : Colibri

7.2 Design and modelling of Colibri UAV

The Colibri drone is derived from a tail-sitter drone with a wing that generates lift during the forward flight. This wing has several actuators : four motors $u_i,\ i = 1, 2, 3, 4$ and two elevons δ_l and δ_r. We can define the control vector u_W of the wing based on Figure 7.1 as $u_\mathrm{W} = [u_1\ u_2\ u_3\ u_4\ \delta_\mathrm{l}\ \delta_\mathrm{r}]^\top$. A fuselage linked by a pivot is secured at the aerodynamic centre of the wing. This fuselage supports the autopilot, the battery, a motor and a tail to keep it horizontal. In Figure 7.1, all the aerodynamic control surfaces are shown in pink and the propellers are shown in green. There are three reference frames attach to the drone. (I) is a NED inertial reference frame (or world frame) linked to the earth's surface, (W) is a wing reference frame attached to the drone wing and (F) is a fuselage reference frame attached to the drone fuselage.

Some of the characteristic dimensions are shown in Table 7.1. Note that the motors are positioned symmetrically on the wing, which means that the position can be described by focusing on one side.

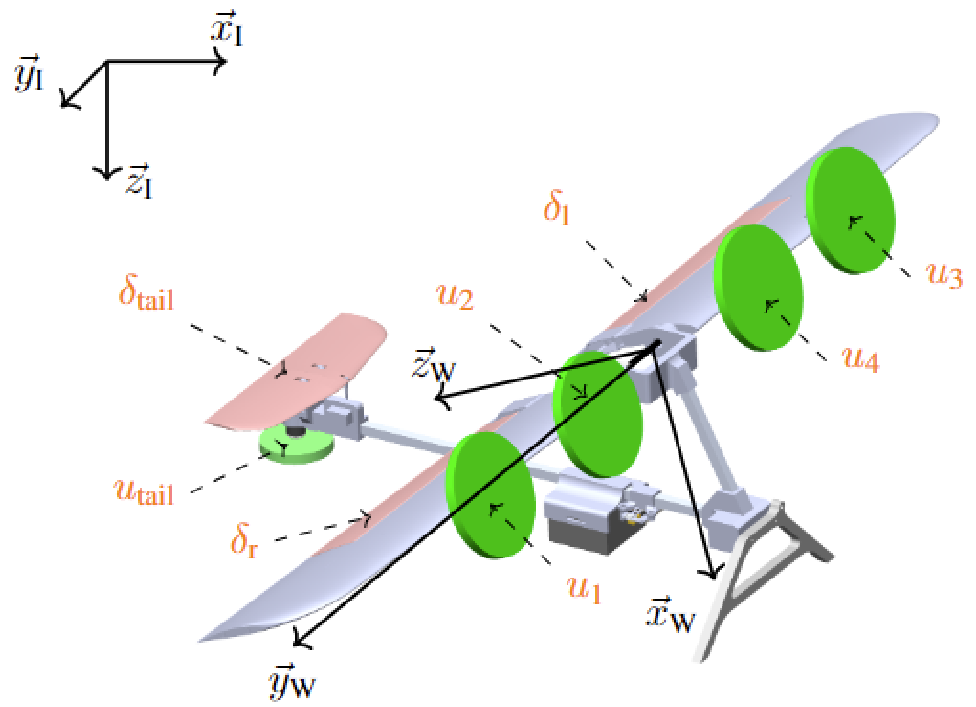

FIGURE 7.1 – Inertial (I) and wing (W) reference frames and the Colibri architecture.

The modelling is based on the results of [?, Section 2.15]. The algorithm for computing matrices M, A, Q and B is in [?], which provides us the equations of motion of a constrained multibody system :

$$\ddot{x} = \hat{M}^\dagger \begin{bmatrix} Q \\ B \end{bmatrix} = \begin{bmatrix} (I - A^\dagger A)M \\ A \end{bmatrix}^\dagger \begin{bmatrix} Q \\ B \end{bmatrix} \tag{7.1}$$

whose expression is valid as long as $\hat{M}$ has full rank and where A, M, Q and B are described next.

We will use quaternions $q = \begin{bmatrix} \eta \ \epsilon^\top \end{bmatrix}^\top \in \mathbb{S}^3 := \{q \in \mathbb{R}^4 : |q| = 1\}$ to represent the orientations of the two bodies. The ensuing rotation matrix $R(q) \in SO(3) := \{R \in \mathbb{R}^{3\times3} : R^\top R = I, \det(R) = 1\}$ is uniquely defined as $R(q) := I + 2\eta\,[\epsilon]_\times + 2\,[\epsilon]_\times^2 = [R_1 \ R_2 \ R_3]$.

According to Figure 7.1 and 7.2, define the vectors $p_\mathrm{F} = \overrightarrow{O_\mathrm{I}O_\mathrm{F}}$, $p_\mathrm{W} = \overrightarrow{O_\mathrm{I}O_\mathrm{W}}$, $d_\mathrm{FW} = \overrightarrow{O_\mathrm{F}O_\mathrm{W}}$ satisfying $d_\mathrm{FW} = p_\mathrm{W} - p_\mathrm{F}$ and $d_{\mathrm{MO_W}} = \overrightarrow{\mathrm{MO_W}}$, $d_{\mathrm{GO_W}} = \overrightarrow{\mathrm{GO_W}}$.

The overall state vector is $(x, v) \in \mathbb{R}^{28}$ with $x = (p_\mathrm{W},\ q_\mathrm{W},\ p_\mathrm{F},\ q_\mathrm{F}) \in \mathbb{R}^{14}$ and $v = (v_\mathrm{W},\ \dot{q}_\mathrm{W},\ v_\mathrm{F},\ \dot{q}_\mathrm{F}) = (\dot{p}_\mathrm{W},\ \dot{q}_\mathrm{W},\ \dot{p}_\mathrm{F},\ \dot{q}_\mathrm{F}) = \dot{x} \in \mathbb{R}^{14}$, where $v_\mathrm{W} = \dot{p}_\mathrm{W} \in \mathbb{R}^3$

Parameter	Value	Units
m_{W} (wing mass)	0.53	kg
m_{F} (fuselage mass)	1.17	kg
$J_{\mathrm{W}} = diag(J_x^{\mathrm{W}}, J_y^{\mathrm{W}}, J_z^{\mathrm{W}})$	$\mathrm{diag}(0.1677, 0.0052, 0.1634)$	$\mathrm{kg\,m^2}$
$J_{\mathrm{F}} = diag(J_x^{\mathrm{F}}, J_y^{\mathrm{F}}, J_z^{\mathrm{F}})$	$\mathrm{diag}(0.0191, 0.0161, 0.0343)$	$\mathrm{kg\,m^2}$
k_{f} (propeller thrust coeff.)	1.7800e-8	kg m
$d_{\mathrm{MO_W}}$	$[0.383, 0, -0.167]^\top$	m
$d_{\mathrm{GO_W}}$	$[0.052, 0, -0.171]^\top$	m

TABLE 7.1 – Numerical parameters of the Colibri model.

represents the linear velocity of the wing in the inertial reference frame, $\dot{q}_{\mathrm{W}} \in \mathbb{R}^4$ is the derivative of the quaternion, $q_{\mathrm{W}} \in \mathbb{R}^4$ representing the orientation of the wing, $v_{\mathrm{F}} = p_{\mathrm{F}} \in \mathbb{R}^3$ is the linear velocity of the fuselage in the inertial reference frame and $\dot{q}_{\mathrm{F}} \in \mathbb{R}^4$ is the derivative of the quaternion $q_{\mathrm{F}} \in \mathbb{R}^4$ representing the fuselage orientation. It can be seen that the state vector is not minimal. It should be noted that the angular velocity $\omega \in \mathbb{R}^3$ can be obtained from the quaternion derivative $\dot{q}$ using equation [**?**, equation (2.7)] recalled here :

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

where $H(q) \in \mathbb{R}^{3\times4}$ is a matrix defined by $H(q) = 2\begin{bmatrix}-\epsilon & \eta I_3 - [\epsilon]_\times\end{bmatrix}$. For deriving the equations of motion, recalling that $R_i(q) \in \mathbb{R}^3, i = 1, 2, 3$ are the three columns of a rotation matrix associated with quaternion q, define matrices $L_{\mathrm{i}}^{\mathrm{W}}(q_{\mathrm{W}}) = \frac{\partial R_i}{\partial q}(q_{\mathrm{W}}) \in \mathbb{R}^{3\times4}$, $L_{\mathrm{i}}^{\mathrm{F}}(q_{\mathrm{F}}) = \frac{\partial R_i}{\partial q}(q_{\mathrm{F}}) \in \mathbb{R}^{3\times4}$ and $L_{O_{\mathrm{F}}^{\mathrm{W}}} = \sum_{i=1}^{3} d_{\mathrm{FW}}(i) L_{\mathrm{i}}^{\mathrm{F}}(q_{\mathrm{F}})$, $i \in 1, 2, 3$, where $d_{\mathrm{FW}}(i)$ denotes the i-th component of vector $d_{\mathrm{FW}} = p_{\mathrm{W}} - p_{\mathrm{F}}$. Since O_{W} is located at the wing's center of rotation, the distance d_{FW} is a constant, since O_{W} and O_{F} can be assumed to belong to the same solid (the fuselage). We deduce, with homogeneity, $\dot{L}_{O_{\mathrm{F}}^{\mathrm{W}}} = \sum_{i=1}^{3} d_{\mathrm{FW}}(i) L_{\mathrm{i}}^{\mathrm{F}}(\dot{q}_{\mathrm{F}})$. With these definitions, select the matrices in (7.1) as

$$M = \begin{bmatrix} m_{\mathrm{W}} I_3 & \mathbb{0}_{3\times4} & \mathbb{0}_3 & \mathbb{0}_{3\times4} \\ \mathbb{0}_{4\times3} & H_{\mathrm{W}}^\top J_{\mathrm{W}} H_{\mathrm{W}} & \mathbb{0}_{4\times3} & \mathbb{0}_4 \\ \mathbb{0}_3 & \mathbb{0}_{3\times4} & m_{\mathrm{F}} I_3 & \mathbb{0}_{3\times4} \\ \mathbb{0}_{4\times3} & \mathbb{0}_4 & \mathbb{0}_{4\times3} & H_{\mathrm{F}}^\top J_{\mathrm{F}} H_{\mathrm{F}} \end{bmatrix} \in \mathbb{R}^{14\times14}, \tag{7.2}$$

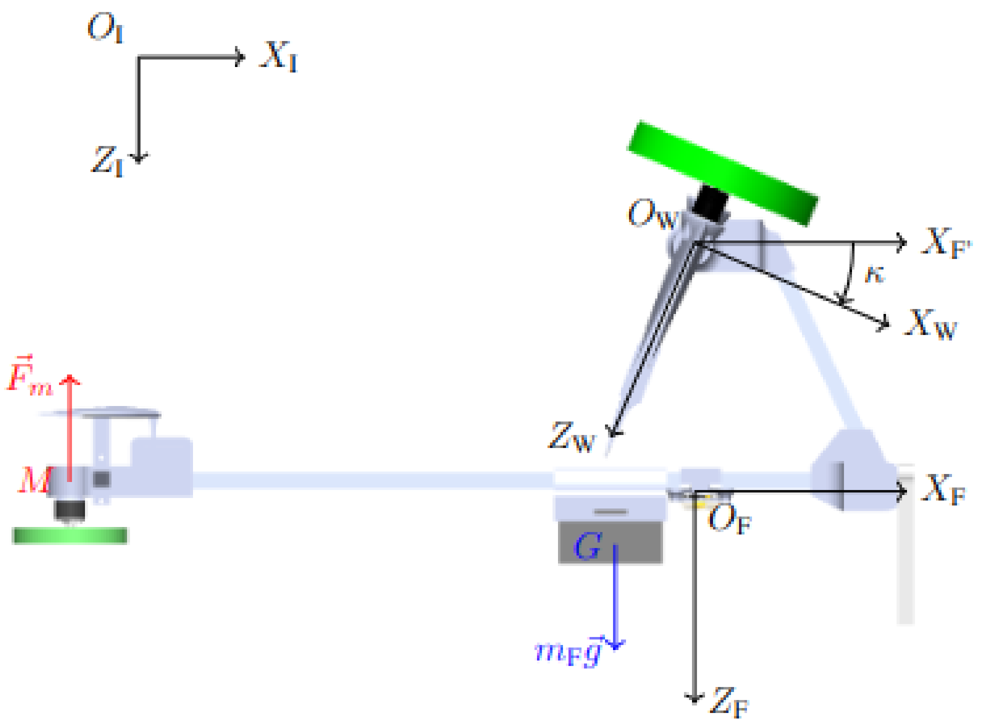

FIGURE 7.2 – Inertial (I), fuselage (F) and wing (W) reference frames and forces acting on the Colibri UAV.

where we denoted $H_W = H(q_W)$, $H_F = H(q_F)$, and

$$Q = \begin{bmatrix} m_W g e_3 + R(q_W) F_b \\ -2\dot{H}_W^\top J_W \dot{H}_W \dot{q}_W + H_W^\top M_W \\ m_F g e_3 + R(q_F) F_F \\ -2\dot{H}_F^\top J_F \dot{H}_F \dot{q}_F + H_F^\top M_F \end{bmatrix} \in \mathbb{R}^{14}, \tag{7.3}$$

where $\dot{H}_W$ denote $H(\dot{q}_W)$, coinciding with the time derivative of $H(q_W)$ and $\dot{H}_F$ denote $H(\dot{q}_F)$, coinciding with the time derivative of $H(q_F)$. Moreover, F_b and M_b represent, respectively, all the forces and moments acting on the wing. The expressions of M and Q are taken from [**?**, equations (45) and (57)] where the ϕ theory is developed, a parametrisation that allows the classical angles of incidence and sideslip to be subtracted and the hover singularity to be avoided. For lack of space, they will not be more detailed. Finally, $F_F = F_m$ et M_F represent respectively the set of non-gravitational forces and moments acting on the fuselage expressed in the frame O_{W}. In particular, $F_m = -k_f {u_{\mathrm{tail}}}^2$ is the force generated by the motor located at the tail of the fuselage and u_{tail} is the motor rotation speed, while

$$M_{\mathrm{F}} = m_{\mathrm{F}} g e_3 \times d_{\mathrm{GO_W}} + F_m \times d_{\mathrm{MO_W}}, \tag{7.4}$$

where $d_{\mathrm{MO_W}}$ is the distance between the motor location and the center of rotation and $d_{\mathrm{GO_W}}$ is the distance between the location of the fuselage's center of gravity and the center of rotation.

The set of constraints associated with the nonminimality or the state (x, v) and by the pivot connection between the two bodies is given by :

$$\begin{cases} \varphi_1 \coloneqq q_{\mathrm{W}}^\top q_{\mathrm{W}} - 1 = 0 \\ \varphi_2 \coloneqq q_{\mathrm{F}}^\top q_{\mathrm{F}} - 1 = 0 \\ \varphi_3 \coloneqq R_2(q_{\mathrm{W}})^\top R_3(q_{\mathrm{F}}) = 0 \\ \varphi_4 \coloneqq R_2(q_{\mathrm{W}})^\top R_1(q_{\mathrm{F}}) = 0 \\ \varphi_5 \coloneqq p_{\mathrm{F}} + d_{\mathrm{FA}} + p_{\mathrm{W}} = 0 \end{cases} \tag{7.5}$$

The first two constraints impose the unit norm of the quaternions q_{F} and q_{W}. The third and fourth constraints are related to a moving pivot constraint, i.e. the orthogonality of two vectors is imposed. The last one is a positional constraint so that the point of the centre of rotation belonging to the wing coincides with the point defined in the fuselage. This constraint is based on a three-dimensional geometric closure.
It is more convenient to express the set of constraints as a stable dynamical system converging to zero, so we convert each one of the constraints in the form :

$$\ddot{\varphi}_i + \delta_1 \dot{\varphi}_i + \delta_2 \varphi_i = 0, i \in 1, 2, 3, 4, 5, \tag{7.6}$$

with the selections $(\delta_1, \delta_2) = (0.5,\ 8)$ being the coefficients of a stable polynomial, so that, regardless of the selection $\varphi_i(0) = 0$, we have $\lim\limits_{t\to\infty} \varphi_i(t) = 0$. By differentiating constraints (7.5) twice and factoring them out in the form $A(x, \dot{x})\ddot{x} = B(x, \dot{x})$, we obtain the expression of $A(x, \dot{x})$ reported in equation (7.7) and $B(x, \dot{x})$ reported in the equation (7.8) at the start of the next page.

$$A = \begin{bmatrix} \mathbb{0}_{1\times 3} & q_{\mathrm{W}}^{\top} & \mathbb{0}_{1\times 3} & \mathbb{0}_{1\times 4} \\ \mathbb{0}_{1\times 3} & \mathbb{0}_{1\times 4} & \mathbb{0}_{1\times 3} & q_{\mathrm{F}}^{\top} \\ \mathbb{0}_{1\times 3} & R_3(q_{\mathrm{F}})^{\top} L_2^{\mathrm{W}}(q_{\mathrm{W}}) & \mathbb{0}_{1\times 3} & R_2(q_{\mathrm{W}})^{\top} L_3^{\mathrm{F}}(q_{\mathrm{F}}) \\ \mathbb{0}_{1\times 3} & R_1(q_{\mathrm{F}})^{\top} L_2^{\mathrm{W}}(q_{\mathrm{W}}) & \mathbb{0}_{1\times 3} & R_2(q_{\mathrm{W}})^{\top} L_3^{\mathrm{F}}(q_{\mathrm{F}}) \\ \mathbb{I}_3 & L_{O_{\mathrm{F}}^{\mathrm{W}}} & -\mathbb{I}_3 & \mathbb{0}_{3\times 4} \end{bmatrix} \tag{7.7}$$

The simulation of a drone remains complex, as it is naturally unstable. We have chosen to use the control law proposed in [?] extended to 6 DOF dynamics to stabilize the system. This PI-based control stabilises the wing. Another control law based on a proportional-derivative feedback stabilises the fuselage to keep it horizontal. The closed-loop simulation results are shown in Figure 7.3. Considering the degrees of freedom of the pivot link, the coupling between the two bodies is clearly visible from the lower three plots. Indeed, the roll and yaw angles $(\phi_{\mathrm{F}}, \psi_{\mathrm{F}})$ and $(\phi_{\mathrm{W}}, \psi_{\mathrm{W}})$ of the fuselage and wing coincide perfectly, while the pitch angles $(\theta\mathrm{F}, \theta\mathrm{F})$ are radically different.

7.3 State estimation

In order to stabilise this two-body UAV system, it is necessary to know the position and orientation of the two bodies. Due to the pivot link between the wing and the fuselage, the difference between the orientation of the wing and the orientation of the fuselage is simply a rotation about the pitch axis of the wing. The two other orientations (roll and yaw) coincide. The position of the fuselage's centre of gravity can be deduced from the position of the wing's centre of gravity and the angle between the fuselage and the wing. This angle is measured by a quadrature rotary encoder (CUI Devices AMT22, Absolute Encoders, 12 bit, SPI), which returns a quantized angular measurement with a step size of 0.09°. Given this angular measurement, we discuss below the estimation of the speed information, so as to reconstruct the state of the UAV.

7.3.1 Sensors placement

A first question pertains to the sensors placements : the IMU (accelerometer, gyroscope and magnetometer) can be installed on the fuselage or on the wing. Installing the IMU on the wing means that the measurements can be taken directly in the desired reference frame, but the measurements are noisier because the IMU is attached to the structure supporting the motors. Given the size of the wing, their flexibility can generate resonances and can perturb the measurements. Installing the IMU on the fuselage reduces vibrations, but means that the measurements must be transformed in the wing reference frame. The corresponding transformation can be computed from the rotary encoder measurement, providing the angle between the wing and the fuselage, and also from the measurements taken with the CAD software, providing precise information about the distances between the wing and fuselage frames. Our final choice is to attach the IMU to the fuselage. Another consideration is that the autopilot board, which already have an integrated IMU, is also supposed to be connected to the payload and other sensors attached to the fuselage. It is thus limiting the number of cables at the pivot point to the actuators commands and power supply.

7.3.2 Angular speed estimation

As explained above, we can measure the angle $\kappa \in \mathbb{R}$ between the wing and the fuselage using the rotary encoder. Then, to estimate the angular velocity we use the high-gain observer proposed in [**?**] (see also [**?**] for the use of high-gain observers to estimate time derivatives). This method is preferable to a finite difference derivative, as the quantized information generated by the rotary encoder can result in bursts in the estimated angular velocity values.
Denote by $\kappa \in \mathbb{R}$ the measured position variable, by $\omega_\kappa := \dot{\kappa} \in \mathbb{R}$ its derivative, to be estimated, and by $\xi = [\kappa,\ \omega_\kappa]^\top \in \mathbb{R}^2$ their juxtaposition in a single vector. Denote also $\hat{\xi}$ the estimate of ξ as follows :

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

Following [**?**], the estimator dynamics is given by

$$\dot{\hat{\xi}} = \begin{bmatrix} 0 & 1 \\ 0 & 0 \end{bmatrix} \hat{\xi} + \begin{bmatrix} \frac{k_p}{\epsilon_\kappa} \\ \frac{k_v}{\epsilon_\kappa^2} \end{bmatrix} (\kappa - \hat{\kappa}), \tag{7.9}$$

where κ is the angular measurement recovering from the sensors, k_p and k_v are two positive scalars gains such that the characteristic equation $s^2 + k_v s + k_p = 0$ has roots with negative real part. For our estimators, we have selected $k_p = 1$ and $k_v = 1.3$ so as to get a damping factor $\zeta = 0.65$ leading to a slightly underdamped response as a suitable trade-off between a fast rise time and a mildly oscillatory response. The high-gain scaling factor ϵ_κ can be conveniently adjusted in order to obtain a trade-off between smoothing action (obtained by increasing ϵ_κ) and reduction of the time lag of the estimator (obtained by reducing ϵ_κ). Moreover, the smoothing action of

the proposed approach mitigates the effect of the quantized position measurements. We have selected $\epsilon_\kappa = 0.05$ for our experiments. Figure 7.4 shows the experimental results obtained after implementation of the high-gain filter (7.9) in the case of a flight generating high-amplitude angular oscillations. We carried out differentiation by finite difference (in green) in post-treatment to compare the results. Due to the quantized nature of the rotary encoder, we observe that the angular velocity obtained by finite difference is very noisy. We can see that the high-gain filter makes it possible to estimate the angular velocity more accurately (in red), albeit with a slight delay. Thanks to the addition of an extra IMU on the wing in a specific flight test, it is possible to compare the velocity estimate with the wing's gyroscope (MPU9250) measurements, visible on the bottom graph of Figure 7.4 (blue trace). We can see that the gyroscope readings are somewhat noisy, due in particular to the vibrations generated by the motors.
In order to perform the necessary transformation among the reference frames, define the quaternion $q_{\hat{\kappa}} \in \mathbb{S}^3$ as follows :

$$q_{\hat{\kappa}} = \left[\cos\left(\frac{\hat{\kappa}}{2}\right) \;\; 0 \;\; \sin\left(\frac{\hat{\kappa}}{2}\right) \;\; 0\right]^\top \tag{7.10}$$

7.3.3 Wing state estimation

Based on the estimated angle $\hat{\kappa}$ and the estimated angular velocity $\hat{\omega}_\kappa$, it is possible to transform the measurements from the fuselage to the wing frame. All the sensors are installed on the autopilot board, which is itself attached to the fuselage. However, as mention in introduction, we want to use INDI to stabilize the wing. So this control law requires the state information in the wing reference frame, where all the forces are applied (aerodynamic and traction). Then, two viable solution are possible : perform the state estimation in the fuselage reference frame and rotate the estimation, using the estimate of the angle $\hat{\kappa}$, or rotate the raw measurements in advance to express them in the wing reference frame, and then perform the state estimation on the latter. Given the current architecture of the software in the Paparazzi [1] system, it is cumbersome to have two joint state estimation structures, so it is difficult to implement the first solution, where the controller directly retrieves the current state estimation. For this reason, we have chosen to estimate the state of the wing from data measured on the fuselage. To this end, we detail below the coordinate transformation for the three sensors : gyroscope, accelerometer and magnetometer.

For the gyroscope-based angular rate measurements, we may compute the an-

1. `https://github.com/enacuavlab/paparazzi/tree/rot_state_est`

gular velocity of the wing expressed in the wing frame as

$$\omega_{\mathrm{W}} = R(q_{\hat{\kappa}}) \left(\omega^{\mathrm{F}}_{gyro} + \begin{bmatrix} 0 \\ \omega_{\kappa} \\ 0 \end{bmatrix} \right) \tag{7.11}$$

where $\omega^{\mathrm{F}}_{gyro}$ is the angular velocity measured by the gyro on the fuselage, expressed in the fuselage frame, $\hat{\omega}_{\kappa}$ is the estimated angular velocity of the wing relative to the fuselage, as per (7.9), and $q_{\hat{\kappa}}$ is the quaternion defined in (7.10). Expression (7.11) is similar to a composition of angular velocities and a reference frame transformation.

For the acceleration measurement with the accelerometer, we may use the following relation Expression (7.12) is obtained from the rate of change transport theorem [**?**], where we find the Euler acceleration term $\dot{\omega_{\mathrm{F}}} \times d_{AF}$ and the centripetal acceleration term $\omega_{\mathrm{F}} \times (\omega_{\mathrm{F}} \times d_{AF})$. Coriolis Acceleration $2\omega_{\mathrm{F}} \times \frac{d(d_{\mathrm{FW}})}{dt}\Big|_{O_{\mathrm{F}}}$ and the rate of acceleration $\frac{d^2(d_{\mathrm{FW}})}{d^2t}\Big|_{O_{\mathrm{F}}}$ are zero because d_{FW} is contant.

$$a_{\mathrm{W}} = R(q_{\hat{\kappa}}) \left(a^F_{acc} + \dot{\omega}^F_{gyro} \times d_{\mathrm{FW}} + \omega^F_{gyro} \times (\omega^F_{gyro} \times d_{\mathrm{FW}}) \right) \tag{7.12}$$

where $a^F_{acc} \in \mathbb{R}^3$ is the acceleration measured by the accelerometer on the fuselage, expressed in the fuselage frame and ω^F_{gyro}, the angular velocity of the fuselage, same as the equation (7.11). The angular acceleration $\dot{\omega}^F_{gyro}$ in (7.12) is computed by a finite difference.

For the magnetometer measurements, we have

$$E_{\mathrm{W}} = R(q_{\hat{\kappa}}) E_{mag} \tag{7.13}$$

where $E_{mag} \in \mathbb{R}^3$ is the magnetometer output, expressed in the fuselage frame and $E_{\mathrm{W}} \in \mathbb{R}^3$ is the computed measurement expressed in the wing frame.

To obtain the wing state estimate, we use a sensor measurement fusion algorithm : extended Kalman filter [2] (EKF) which provide an estimate of the following states : p_{W}, v_{W}, q_{W} from measurements transformed in the wing reference frame ω_{W} (eq. (7.11)), a_{W} (eq. (7.12)), E_{W} (eq. (7.13)) and external vision system pose data, which provides a precise measurement of the drone's position p_{W} and speed v_{W} in the inertial reference frame (I).

2. `https://github.com/PX4/PX4-ECL/tree/master`

7.3.4 Fuselage orientation estimation

To determine the orientation of the fuselage, we may perform a composition between the quaternion representing the orientation of the wing q_{W} result of EKF and the quaternion constructed from the filtered measurement of the rotary encoder $q_{\hat{\kappa}}$ in (7.10),

$$q_{\mathrm{F}} = q_{\mathrm{W}} \otimes q_{\hat{\kappa}} \tag{7.14}$$

where the operator $\otimes$ denotes the qaternion product. The knowledge of q_{F} is needed to keep the fuselage perfectly horizontal.

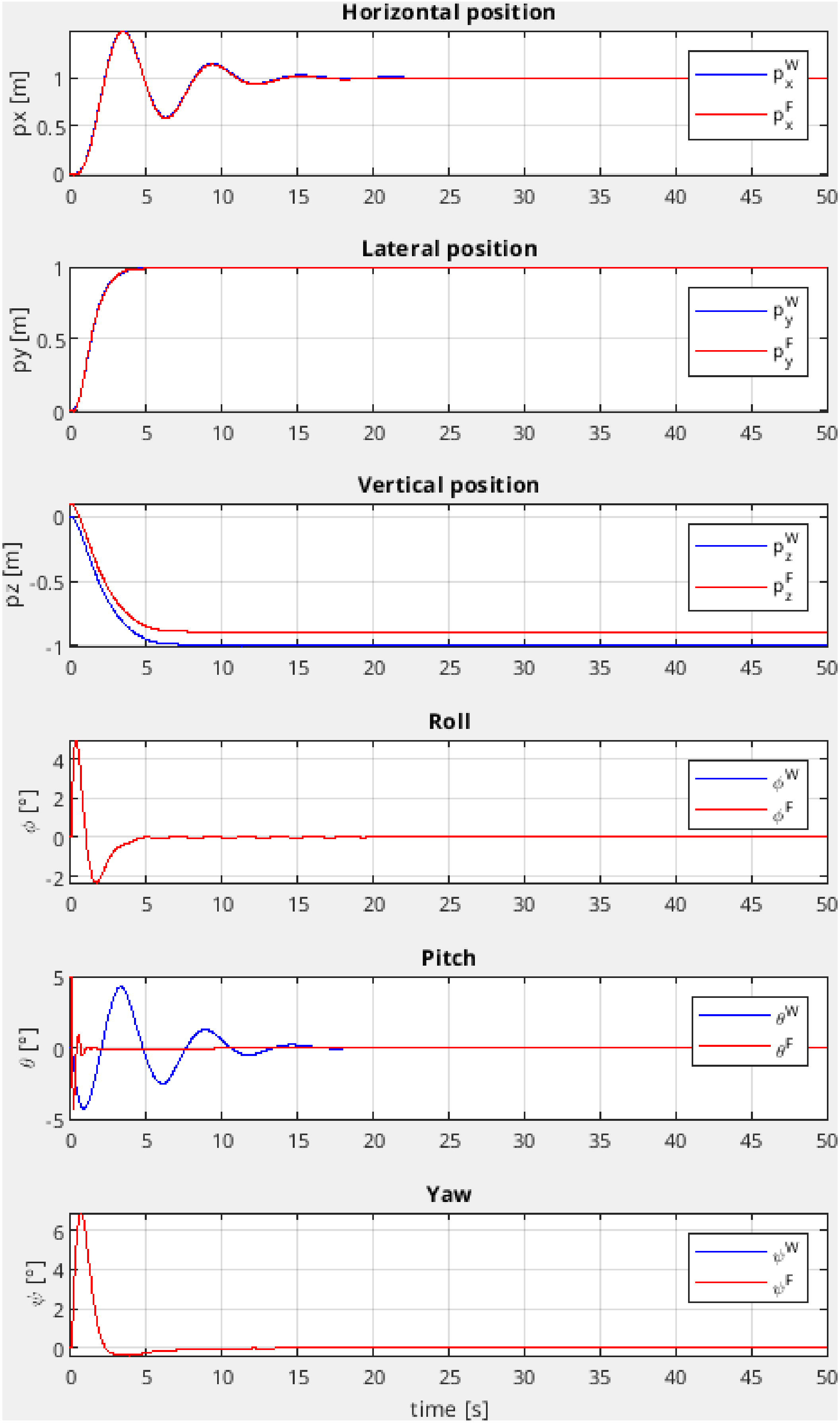

FIGURE 7.3 – Position and orientation simulation of the multi-body UAV Colibri in closed loop with a simple double-loop controller.

$$B = \begin{bmatrix} -\delta_1 q_{\mathrm{W}}^\top \dot{q}_{\mathrm{W}} - \frac{\delta_2}{2}(q_{\mathrm{W}}^\top q_{\mathrm{W}} - 1) - \dot{q}_{\mathrm{W}}^\top \dot{q}_{\mathrm{W}} \\ -\delta_1 q_{\mathrm{F}}^\top \dot{q}_{\mathrm{F}} - \frac{\delta_2}{2}(q_{\mathrm{F}}^\top q_{\mathrm{F}} - 1) - \dot{q}_{\mathrm{F}}^\top \dot{q}_{\mathrm{F}} \\ -R_3(q_{\mathrm{F}})^\top \dot{L}_2^{\mathrm{W}} \dot{q}_{\mathrm{W}} - R_2(q_{\mathrm{W}})^\top \dot{L}_3^{\mathrm{F}} \dot{q}_{\mathrm{F}} - 2\dot{q}_{\mathrm{W}}^\top {L_2^{\mathrm{W}}}^\top L_3^{\mathrm{F}} \dot{q}_{\mathrm{F}} - \delta_1 (R_3(q_{\mathrm{F}})^\top L_2^{\mathrm{W}} \dot{q}_{\mathrm{W}} + R_2(q_{\mathrm{W}})^\top L_3^{\mathrm{F}} \dot{q}_{\mathrm{F}}) - \delta_2 \varphi_3 \\ -R_1(q_{\mathrm{F}})^\top \dot{L}_2^{\mathrm{W}} \dot{q}_{\mathrm{W}} - R_2(q_{\mathrm{W}})^\top \dot{L}_1^{\mathrm{F}} \dot{q}_{\mathrm{F}} - 2\dot{q}_{\mathrm{W}}^\top {L_2^{\mathrm{W}}}^\top L_1^{\mathrm{F}} \dot{q}_{\mathrm{F}} - \delta_1 (R_1(q_{\mathrm{F}})^\top L_2^{\mathrm{W}} \dot{q}_{\mathrm{W}} + R_2(q_{\mathrm{W}})^\top L_1^{\mathrm{F}} \dot{q}_{\mathrm{F}}) - \delta_2 \varphi_4 \\ \dot{L}_{O_{\mathrm{F}}^{\mathrm{W}}} \dot{q}_{\mathrm{F}} - \delta_1 (v_{\mathrm{W}} + \dot{L}_{O_{\mathrm{F}}^{\mathrm{W}}} \dot{q}_{\mathrm{W}} - v_{\mathrm{F}}) - \delta_1 \varphi_5 \end{bmatrix} \tag{7.8}$$

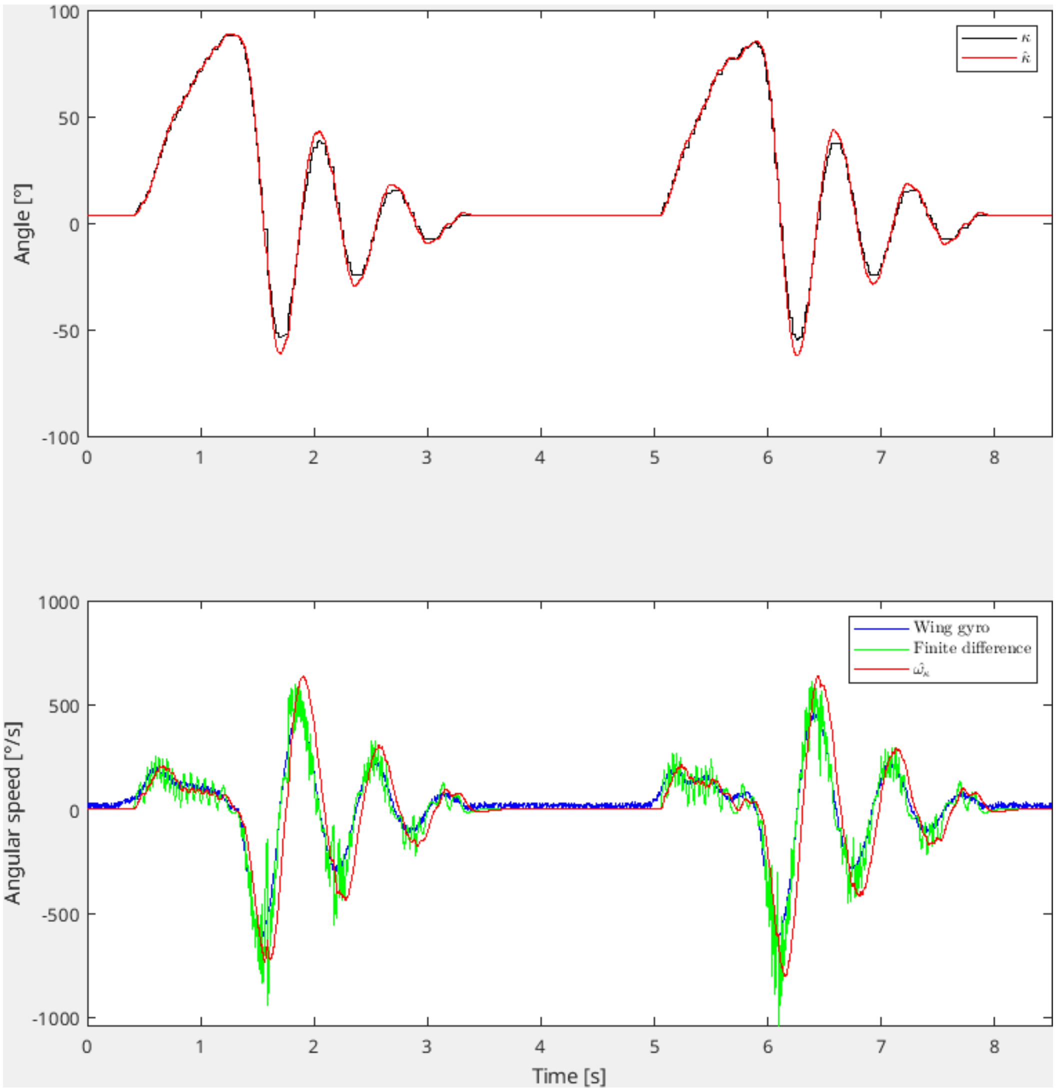

FIGURE 7.4 – Angular position measurement (black,top plot), wing gyro velocity measurement (blue,bottom plot), finite difference velocity estimation (green, bottom plot) and high-gain estimates (red curves)

Chapitre 8

Commande d'un drone à aile libre rotation libre

Sommaire

8.1 Inversion non linéaire incrémentale de la dynamique du drone . . . 79

8.2 Experimentation . . . 80

8.3 Commande Udwadia-Kalaba . . . 81

8.4 Vols expérimentaux . . . 81

8.1 Inversion non linéaire incrémentale de la dynamique du drone

The theory of Incremental Nonlinear Dynamic Inversion (INDI) used in the context of micro-UAVs is presented in [**?**]. We use the notation proposed in [**?**], without providing extra details, due to length constraints. The central underlying assumption is that the so-called timescale separation principle holds w.r.t. the actuator dynamics and the dynamics of aerodynamic forces and moments. The control signal can then be computed incrementally using the actuator effectiveness matrix G.

$$u_{\mathrm{W}} = u_{\mathrm{W}} + G^{\dagger}(\nu - \begin{bmatrix} \dot{\omega}_{\mathrm{W}} \\ T_{\mathrm{W}} \end{bmatrix}) \tag{8.1}$$

where $\dot{\omega}_{\mathrm{W}} \in \mathbb{R}^3$ is the measured angular acceleration obtain by finite difference from equation (7.11), $T_{\mathrm{W}} \in \mathbb{R}$ is the current thrust, ν is define in [**?**, equation (4)] and G

is the control effectiveness matrix, determined as follows :

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

This selection of efficiency matrix has been determined for the hovering flights, but it is necessary to carry out a different study for the forward flight.

To stabilise the fuselage, we use a PD feedback from the angle θ_{F} formed between the fuselage and the horizontal, which we want to keep at zero. This is obtained by converting the quaternion q_{F} of equation (7.14) into an Euler angle by following the 'ZYX' Euler convention. The PD feedback provides the reference u_{tail} for the angular speed of the motor generating the force F_m (see Figure 7.2), as follows

$$u_{\mathrm{tail}} = u_{eq} + k_p \theta_{\mathrm{F}} + k_d \dot{\theta}_{\mathrm{F}},$$

where u_{eq} is the equilibrium motor command to keep the fuselage horizontal in the absence of disturbance and k_p, k_d are tunable scalar gains. The value u_{eq} was obtained by applying a moment theorem to the fuselage at the point O_{W}. In fact, the two moments that come into effect on the fuselage are the torque due to the thrust force of the tail motor and the torque due to the position of the fuselage's centre of gravity. The gains k_p et k_d were adjusted in flight to ensure satisfactory flight behaviour. We obtain $\dot{\theta}_{\mathrm{F}}$ from $\omega^F_{gyro} = [\dot{\phi}_{\mathrm{F}}\ \dot{\theta}_{\mathrm{F}}\ \dot{\psi}_{\mathrm{F}}]^\top$.

8.2 Experimentation

An experimental prototype was developed, as shown in Figure 8.1. A selection of the experimental results in controlled flight is shown in Figure 8.2.

About Figure 8.2, from 0 s to 8 s, the drone is on the ground. From 8 s to 16 s, the drone takes off to reach a height of 2 metres visible from the third plot. This height is reached after a 10 % overshoot. The drone is held in this position for 54 s. Incidence oscillations are observed in the fifth and last plot, generating oscillations in the drone's horizontal position. This is due to the coupling between the two bodies, which is not properly stabilized. From 70 s, the UAV starts heading towards the point $p_c = \begin{bmatrix} 3 & 0.9 & -1.5 \end{bmatrix}^\top$ and $\psi_c = 90°$.

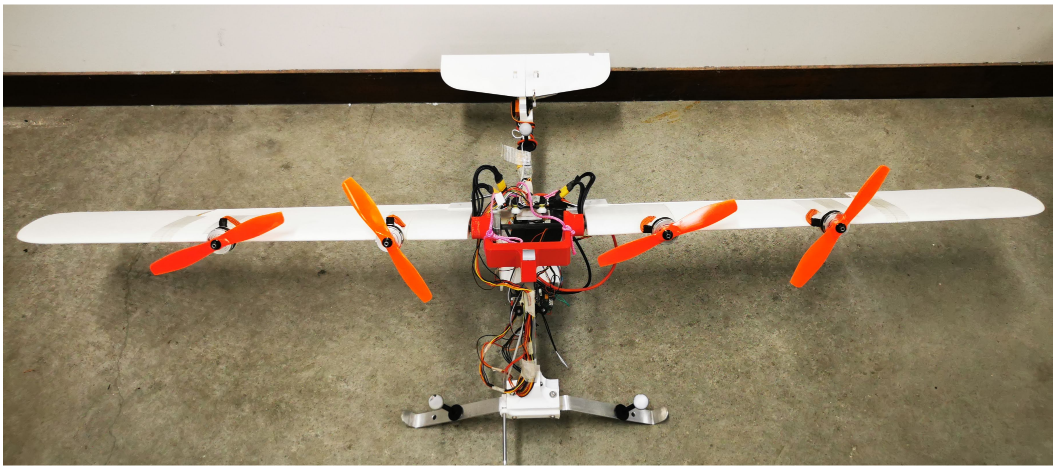

FIGURE 8.1 – Colibri experimental prototype.

8.3 Commande Udwadia-Kalaba

8.4 Vols expérimentaux

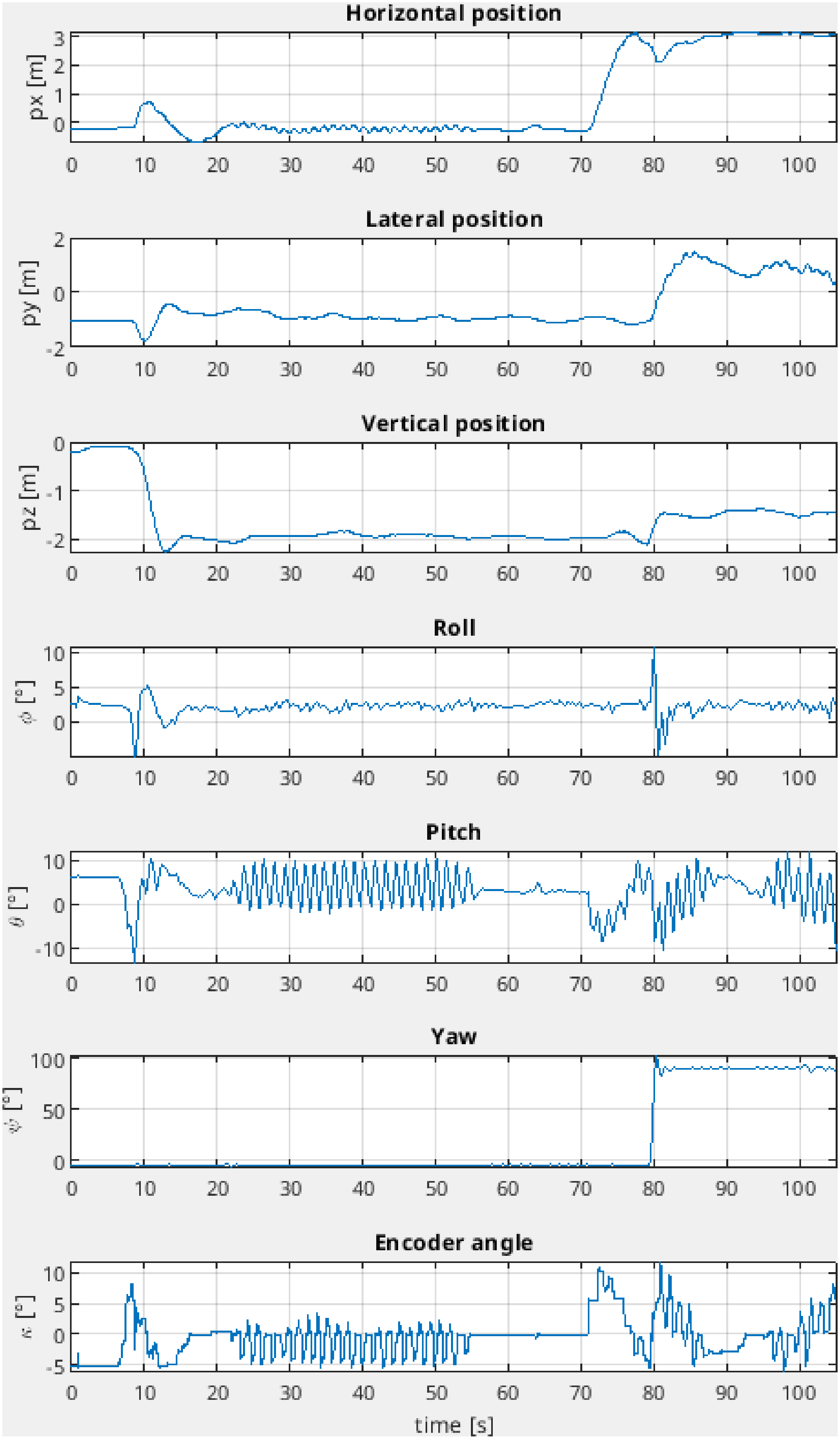

FIGURE 8.2 – Position and orientation of the reference frame wing in the first six graphs and pivot angle measurement on the last graph below during real flight.

Conclusion

Ce manuscrit de thèse rapporte

Annexe A

Exemple d’annexe

A.1 Exemple d’annexe

Bibliographie

[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 page 2.)

[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 11, 12, 29 et 34.)

[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 16.)

[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 10, 11, 12, 13 et 14.)

[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 11, 14 et 15.)

[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 page 2.)

[Sansou 2022] Sansou, F. *Commande hybride d'un drone convertible pour des déplacements sous optimaux.* 2022. [En ligne]. Disponible : `https://arxiv.org/abs/2203.15387`. (Cité en pages 11, 13, 14 et 16.)

[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 27.)

Résumé : resume
Mots clés : mots, clefs

Abstract : abstrat
Keywords : key, words