

REPUBLIQUE ALGERIENNE DEMOCRATIQUE ET POPULAIRE
MINISTRE DE L'ENSEIGNEMENT SUPERIEUR
ET DE LA RECHERCHE SCIENTIFIQUE



N° d'ordre :.....

Série :.....

**Université Mohammed Seddik
BENYAHIA - Jijel**

**Faculté des Sciences et de la Technologie
Département : Electronique**

Thèse Présentée
Pour l'obtention du Diplôme de Doctorat en Sciences
Spécialité : Electronique

Par : YACEF Fouad

Thème

**Commande Intelligente Adaptative de Drones
Miniatures à Voilure Tournante**

Soutenue le : **14 / 03 / 2018** Devant le Jury composé de :

Mr. BOUKABOU Abdelkrim	Professeur	Université de Jijel	Président
Mr. HAMERLAIN Mustapha	DR	CDTA-Alger	Rapporteur
Mr. BOUHALI Omar	Professeur	Université de Jijel	Co-rapporteur
Mr. BELHOCINE Mahmoud	DR	CDTA-Alger	Examineur
Mr. GUIATNI Mohamed	Professeur	EMP-Alger	Examineur
Mr. BOUBAKIR Ahsene	MCA	Université de Jijel	Examineur
Mr. RIZOUG Nassim	MCA	ESTACA-France	Invité

ملخص

توفر الطائرات بدون طيار الصغيرة الحجم فرصة وحيدة من اجل الانتقال من التحكم الغير خطي النظري الى التطبيق العملي كما تتوفر هذه الطائرات على مزايا اخرى بالمقارنة مع مثيلاتها من الطائرات بطيارمثل الصلابة، المرونة في الحركةبالاضافة الى حجم وزن صغيرين بالاضافة الى ان هذه الطائرات يمكنها القيام بمهام مثل المراقبة البحث التصوير و نقل الحمولات بنجاح.

ان انشاء المسارت الاقتصادية من حيث الطاقة تعتبر طريقة مفتاحية بالنسبة لطائرات بدون طيار لانها تضمن مدة طيران اكبر. توفر انظمة التحكم التلأمية الغامضة مزايا احسن بالاضافة الى المقاومة ضد المؤثرات الخارجية بفضل قدرتها على تعديل اعدادات طريقة التحكم. قامت ابحاث معمقة في مجال التحكم التلأمي الغامض النظري انشاء و تحليل انظمة تلأمية مستقرة.

تناقش هذه المذكرة عدة مسائل متعلقة بمحدودية الطاقة المحمولة في الطائرة بدون طيار ذات اربع محركات بالاضافة الى تطبيق طرق التحكم التلأمية الغاظة على هذه الانظمة. قمنا باقتراح نموذج طاقي و التصديق على هذا النموذج من اجل اظهار امكانية اقتصاد الطاقة. بالنسبة الى مشكل اقتصاد الطاقة قمنا بتحويله الى مشكل تحكم اقتصادي باستعمال ادوات رقمية، هذا الاخير تم حله من اجل حساب مداخل التحكم و المسارت الاقتصادية من حيث الطاقة. تم اقتراح طرق تحكم تلأمية غامضة من اجل معالجة مشكل متابعة المسار. تتمثل الطريقة الالى في قانون تحكم تلأمي غامض مباشر مكون بالاعتماد على ملاحظ الحالات. اما الطريقة الثانية فتمثل في مخطط تحكم تلأمي غامض قوي يعتمد على ملاحظ المؤثرات. كما استعملنا نظرية ليابونوف لدراسة استقرار طرق التحكم المقترحة وقوتها.

كلمات مفتاحية:

طائرة بدون طيار ذات اربع محركات، انشاء المسار، اقتصاد الطاقة، تحكم تلأمي غامض، ملاحظ الحالات، استقرار ليابونوف

Résumé

Les drones miniatures à voilure tournante (UAVs) offrent une opportunité unique pour la transition des contrôleurs non linéaires de la théorie à la pratique. Les drones miniatures offrent également plusieurs avantages par rapport à leurs homologues avec pilotes, notamment une extrême persistance, une maniabilité, un poids réduit et une taille réduite. En outre, plusieurs missions telles que la surveillance, l'exploration, la recherche, la photographie, et le transport de charges sont mieux accomplies par les UAV.

La génération de trajectoires optimales avec un minimum d'énergie est une technique clé pour les drones miniatures à voilure tournante, car elle assure une plus grande endurance. Les systèmes de commande adaptative floue offrent des performances améliorées et une robustesse accrue vis-à-vis les incertitudes et les perturbations grâce à leur capacité d'ajuster les paramètres de la commande en fonction des mesures en ligne. Des recherches approfondies dans le domaine de la théorie de la commande adaptative floue ont permis la conception, l'analyse et la synthèse des systèmes adaptatifs flous stables.

Cette thèse aborde certains défis liés à la limitation de l'énergie embarquée pour les UAVs précisément le Quadrotor et à l'application de la commande adaptative floue à ces véhicules. Un modèle énergétique a été introduit et validé par simulation pour montrer la possibilité d'économiser de l'énergie. Le problème de l'optimisation énergétique est transformé en un problème de commande optimale; en utilisant des outils numériques, ce problème a été résolu afin de calculer les entrées de commande et les trajectoires optimales en terme d'énergie. Des architectures de commande adaptative floue sont développées pour surmonter le problème de suivi de trajectoire du Quadrotor, et atténuer les perturbations externes. La première approche est une commande adaptative floue directe développée à base d'un observateur d'état pour le suivi de trajectoire du Quadrotor. Le second schéma de commande est un contrôleur adaptatif flou robuste basé sur un observateur de perturbation. L'analyse de stabilité des approches de commande proposées est effectuée en utilisant la théorie de Lyapunov. Les approches de commande proposées sont validées numériquement à l'aide d'une série de simulation afin de mettre en évidence leurs performances.

Mots-clé:

Quadrotor, Génération de trajectoire, Optimisation d'énergie, Commande adaptative floue, Observateur d'état, Stabilité de Lyapunov.

Abstract

Rotary Wings Unmanned aerial vehicles (UAVs) provide a unique opportunity for the transition of nonlinear controllers from theory to practice. The small, unmanned aerial vehicles (UAVs) examined in this thesis offer a low cost, low risk solution to applications in which safety is a critical concern. Unmanned aircraft also offer several benefits over their manned counterparts including extreme persistence, maneuverability, lower weight and smaller size. Furthermore, several missions such as surveillance, exploration, search-and-track, and lifting of heavy loads are best accomplished by UAVs.

Optimal trajectory generation with minimum energy for UAVs is considered to be one of the key enabling techniques for future high-performance, since it's ensure more endurance for UAVs. Adaptive fuzzy control systems offer improved performance and increased robustness to uncertainties by virtue of their ability to adjust control parameters as a function of online measurements. Extensive research in the field of adaptive fuzzy control theory has enabled the design, analysis, and synthesis of stable adaptive systems. We are now entering the stage in which adaptive fuzzy control for flight systems have reached the requisite level of maturity for application to hardware flight platforms.

This thesis addresses some of the challenges involved with the limitation of embedded energy for UAVs and the application of adaptive fuzzy control systems to UAVs. An energetic model was introduced and validated through simulation to show possibility of saving energy. The problem of energy optimization is transformed to an optimal control problem; using numerical tools, this problem was solved in order to compute minimum-energy control inputs and trajectories. Adaptive fuzzy control architectures are developed to overcome trajectory tracking performance limitation of quadrotor, the most significant of which is external disturbances. The first approach is a direct adaptive fuzzy controller developed based on state observer for trajectory tracking of quadrotor. The second control scheme is a robust adaptive fuzzy controller based on a disturbance observer. Stability analysis of the proposed control approaches are performed by using the Lyapunov theory. The proposed control approaches are validated numerically using a series of simulation studies in order to highlight their performances.

Keywords:

Quadrotor, Trajectory generation, Energy Optimisation, Adaptive fuzzy control, State observer, Lyapunov stability.

Dédicaces

A mes parents

A ma femme et mes enfants Abderrahmane et Zyneb

A ma sœur et mes frères

Remerciements

Le travail présenté dans cette thèse a été réalisé conjointement au Laboratoire de Mécatronique de l'Université de Jijel, au Centre de développement de technologies avancées (CDTA) et à l'École supérieure des techniques aéronautiques et de construction automobile (ESTACA) en France.

Ce sujet de recherche a été proposé et dirigé par Monsieur **Mustapha HAMER-LAIN**, en tant que directeur de thèse, directeur de recherche au centre de développement de technologies avancées (CDTA) et Monsieur **Omar BOUHALI**, le co-directeur, professeur à l'Université de Jijel. Qu'ils trouvent ici l'expression de ma sincère gratitude pour la confiance qu'ils m'ont accordée en me proposant ce sujet. Cette thèse n'aurait pas été achevée sans l'aide et précieux conseils de mes deux encadrants. Ils m'ont toujours encouragé et soutenu durant toutes ces années de préparation de cette thèse. Leurs nombreux conseils, orientations, engagements, et leurs disponibilités m'ont permis d'acquérir des connaissances complémentaires.

Je suis très reconnaissant à Monsieur **Nassim RIZOUG**, enseignant chercheur à l'École supérieure des techniques aéronautiques et de construction automobile (France), pour m'avoir accueilli au sein du laboratoire ESTACA'LAB pendant un séjour de 18 mois. Ses conseils judicieux et bienveillants ainsi que les nombreuses discussions scientifiques fructueuses que nous avons pu avoir étaient très bénéfiques pour ce travail. Je tiens à lui exprimer toute mon amitié et ma profonde gratitude.

Je remercie chaleureusement les membres de Jury qui m'ont fait l'honneur d'évaluer ce modeste travail de thèse. Je remercie chaleureusement Mr **Abdelkrim BOUK-ABOU**, Professeur à l'Université de Jijel de m'avoir fait l'honneur de présider ma soutenance. Mes vives remerciements vont également à Mr **Mahmoud BELHOCINE**, Directeur de recherche au centre de développement des technologies avancées (CDTA), à Mr **Mohamed GUIATNI**, Maître de conférence habilité à l'École militaire polytechnique d'avoir accepté d'agrémenter ce travail avec leurs remarques et conseils. Mes chaleureux remerciements s'adressent aussi à Mr **Ahsene BOUBAKIR**, Maître de conférence habilité à l'Université de Jijel d'avoir orné cette soutenance par sa présence.

Ce projet a été soutenu financièrement par la Direction Générale de la Recherche Scientifique et du Développement Technologique de l'Algérie (CNEPRU J0201720130005) et le Centre de développement des Technologies Avancées (CDTA).

Fouad Yacef

Table des matières

Résumé	iii
Abstract	iv
Dédicaces	v
Remerciements	vi
Table des matières	x
Introduction générale	1
1 État de l’art sur les drones miniatures et les systèmes flous	7
1.1 Introduction	8
1.2 Drones miniatures à voilure tournante	8
1.2.1 Contexte historique	8
1.2.2 Applications	14
1.3 Étude bibliographique sur la commande de Quadrotor	17
1.3.1 Groupes de recherche	17
1.3.2 Guidage et pilotage	19
1.3.3 Techniques de commande	21
1.4 Système flou	23
1.4.1 Description d’un système flou	23
1.4.2 Système flou de Takagi-Sugeno	25
1.4.3 Capacité d’approximation des systèmes Flous	27

1.5	Commande adaptative floue	28
1.5.1	Commande adaptative floue indirecte	29
1.5.2	Commande adaptative floue directe	35
1.6	Conclusion	40
2	Modélisation dynamique du Quadrotor	42
2.1	Introduction	43
2.2	Principe de vol du Quadrotor	43
2.3	Modélisation dynamique	44
2.3.1	Formalisme de Newton-Euler	46
2.3.2	Forces et moments	47
2.3.3	Modèle complet	50
2.3.4	Modèle simplifié	51
2.4	Modèle de rafale de vent	54
2.5	Simulation du modèle dynamique	56
2.6	Conclusion	58
3	Génération de trajectoire optimale en terme d'énergie pour un Quadrotor	59
3.1	Introduction	60
3.2	Travaux antérieurs	60
3.3	Modèle énergétique	62
3.3.1	Dynamique du mouvement	62
3.3.2	Dynamique des moteurs	62
3.3.3	Dynamique de la batterie	63
3.3.4	Énergie consommée et rendement du moteur	65
3.3.5	Effet des entrées de commande sur la consommation d'énergie	67
3.4	Optimisation d'énergie	68
3.5	Optimisation du temps	70
3.6	Méthode de programmation non linéaire	71
3.7	Simulation et comparaison	72

3.7.1	Résultats de simulation	72
3.7.2	Comparaison	73
3.8	Conclusion	76
4	Commande adaptative floue directe d'un Quadrotor avec observateur	78
4.1	Introduction	79
4.2	Formulation du problème	80
4.3	Synthèse de la commande adaptative floue directe	82
4.3.1	Description de système d'inférence flou	83
4.3.2	Structure de la loi de commande	84
4.3.3	Analyse de stabilité	86
4.4	Commande adaptative floue directe avec observateur	89
4.4.1	Synthèse de l'observateur d'état	89
4.4.2	Analyse de stabilité	90
4.5	Stratégie de commande	92
4.6	Résultat de simulation	96
4.6.1	Simulation de la commande adaptative floue directe	98
4.6.2	Simulation de la commande adaptative floue directe avec obser- vateur	103
4.7	Conclusion	109
5	Commande adaptative floue robuste à base d'observateur de perturbation	110
5.1	Introduction	111
5.2	Formulation de problème	112
5.3	Conception de la loi de commande	113
5.4	Synthèse de l'observateur de perturbation	114
5.5	Analyse de stabilité	115
5.6	Résultat de simulation	118
5.7	Conclusion	122
	Conclusion générale et perspectives	123

A Codes Matlab	126
A.1 Codes Matlab chapitre 1	127
A.2 Codes Matlab chapitre 2	132
 Bibliographie	 149

Liste des figures

1	État de l'art sur les drones miniatures et les systèmes flous	7
1.1	(a) véhicule aérien de Leonardo Da Vinci, (b) Bombardement par Balloon en 1848, (c) RP.1 aéronef de Reginald Denny	9
1.2	Gyroplane No.1	10
1.3	(a) Oehmichen No.2, (b) Convertawings Model A	10
1.4	(a) Curtiss-Wright VZ-7, (b) Curtiss-Wright X-19	10
1.5	Draganflyer X4-P	11
1.6	AscTec : (a) Hummingbird, (b) Pelican, (c) Falcon, (d) Firefly	12
1.7	(a) AR. Drone, (b) Phantom	13
1.8	Utilisation de drone dans l'agriculture: (a) Drone à voilures fixes, (b) Drone à voilures tournantes	15
1.9	Application de drone dans la photographie	16
1.10	Application de drone dans la photographie	18
1.11	Exemple de mission de navigation simple	20
1.12	Capteurs et modules requis pour un vol autonome	21
1.13	Configuration de base d'un système flou	24
1.14	Fonction d'appartenance	25
1.15	Commande adaptative floue indirecte	33
1.16	La sortie $x_1(t)$ et la sortie de référence $y_r(t) = \frac{\pi}{30} \sin(t)$ pour les conditions initiales $\mathbf{x}(0) = [\frac{\pi}{60}, 0]^T$,	35
1.17	Commande adaptative floue directe	38
1.18	Fonctions d'appartenance définies dans l'espace d'état	39

1.19	Sortie du système en boucle fermée $x(t)$	40
2	Modélisation dynamique du Quadrotor	42
2.1	Mouvements du Quadrotor	44
2.2	Représentation cartésienne du Quadrotor	45
2.3	Rafale de vent généré par le modèle de Dryden	55
2.4	Décollage vertical (vertical-takeoff)	56
2.5	Mouvement horizontal	57
3	Génération de trajectoire optimale en terme d'énergie pour un quadrotor	59
3.1	schéma de la batterie	64
3.2	Modèle énergétique	65
3.3	Rendement des moteurs	66
3.4	Entrées de commande, mission1 (a) mission2 (b)	67
3.5	Altitude du Quadrotor (a) et Énergie consommée (b)	68
3.6	Position du Quadrotor (a) et Vitesse linéaire (b)	73
3.7	Attitude du Quadrotor (a) et Vitesse angulaire (b)	73
3.8	Trajectoire optimale dans l'espace $(x-y-z)$ (a), et Entrées de commande (b)	74
3.9	fonction coût $E_c(t)$	74
3.10	Position du Quadrotor (a) et Vitesse linéaire (b)	75
3.11	Attitude du Quadrotor (a) et Vitesse angulaire (b)	75
3.12	Trajectoire optimale dans l'espace $(x-y-z)$ (a), et Entrées de commande (b)	76
4	Commande adaptative floue directe d'un Quadrotor avec observateur	78
4.1	Structure de la commande adaptative floue directe avec observateur	94
4.2	Structure de commande pour chaque $i^{\text{ème}}$ sous système	95
4.3	Système du Quadrotor complètement actionné	96
4.4	Fonctions d'appartenances, position	97

4.5	Fonctions d'appartenances, angles d'Euler	98
4.6	Suivi de position sans perturbations externes, CAFD	99
4.7	Suivi d'attitudes sans perturbations externes, CAFD	99
4.8	Entrées de commande sans perturbations externes, CAFD	100
4.9	Position en 3D sans perturbations externes, CAFD	100
4.10	Vitesse du vent	101
4.11	Suivi de position avec perturbations externes, CAFD	101
4.12	Suivi d'attitudes avec perturbations externes, CAFD	102
4.13	Entrées de commande avec perturbations externes, CAFD	102
4.14	Position en 3D avec perturbations externes, CAFD	103
4.15	Suivi de position sans perturbations externes, CAFDO	104
4.16	Suivi d'attitudes sans perturbations externes, CAFDO	105
4.17	Entrées de commande sans perturbations externes, CAFDO	105
4.18	Position en 3D sans perturbations externes, CAFDO	106
4.19	Suivi de position avec perturbations externes, CAFDO	107
4.20	Suivi d'attitudes avec perturbations externes, CAFDO	107
4.21	Entrées de commande avec perturbations externes, CAFDO	108
4.22	Position en 3D avec perturbations externes, CAFDO	108
5	Commande adaptative floue robuste à base d'observateur de perturbation	110
5.1	Suivi de position avec estimation de perturbations externes	119
5.2	Suivi d'attitudes avec estimation de perturbations externes	120
5.3	Entrées de commande	120
5.4	Position en 3D avec estimation de perturbations externes	121
5.5	Perturbations externes, $\hat{w}_i(\text{---})$, $w_i(\text{---})$	121
A	Codes Matlab	126

Liste des tableaux

1	État de l'art sur les drones miniatures et les systèmes flous	7
2	Modélisation dynamique du Quadrotor	42
2.1	Paramètres du Quadrotor	53
3	Génération de trajectoire optimale en terme d'énergie pour un quadrotor	59
3.1	Coefficients du moteur	64
3.2	Paramètres de la batterie	65
4	Commande adaptative floue directe d'un Quadrotor avec observateur	78
5	Commande adaptative floue robuste à base d'observateur de perturbation	110
A	Codes Matlab	126

Nomenclatures

Liste des Abréviations

<i>UAVs</i>	Unmanned aerial vehicles
<i>BTT</i>	Basic training target
<i>RC</i>	Radio commande
<i>LiPo</i>	Lithium polymer
<i>GUI</i>	Graphical user interface
<i>OSP</i>	Open source project
<i>VTOL</i>	vertical take-off and landing
<i>STARMAC</i>	Stanford testbed of autonomous rotorcraft for multi-agent control
<i>LQR</i>	Linear quadratic regulator
<i>BDLC</i>	Brushless direct current motor
<i>PWM</i>	Pulse width modulation
<i>PSD</i>	Power spectral density
<i>BS</i>	Beaufort scale
<i>GTSPN</i>	Generalized traveling salesman problem with neighborhoods
<i>SOC</i>	State of charge
<i>NLPP</i>	Non-Linear programming problems
<i>GPOPS</i>	General purpose OPTimal control softwar
<i>LGR</i>	Legendre gauss radau
<i>IPOPT</i>	Interior point OPTimizer
<i>SNOPT</i>	Sparse nonlinear OPTimizer
<i>CAFD</i>	Commande adaptative floue directe
<i>CAFDO</i>	Commande adaptative floue directe avec observateur

Liste des Symboles

ξ	Position du centre de masse de Quadrotor
η	Orientation de Quadrotor
V	Vitesse linéaire du corps rigide
Ω	Vitesse angulaire
R	Matrice de rotation

I	Matrice d'inertie
m	Masse de Quadrotor
F_{ext}	Forces extérieurs
T_{ext}	Moments extérieurs
F_{grav}	Force de poids
F_i	Force de poussée
Q_i	Traînée de l'hélice
ω_i	Vitesse des moteurs
T_{gyro}	Moment gyroscopique
C_T	Coefficient de portance
C_Q	Coefficient de traînée
I_r	Inertie du moteur
F_{aero}	Force aérodynamique
T_{roul}	Moment de roulis
T_{tang}	Moment de tangage
T_{lac}	Moment de lacet
T_{aero}	Moment aérodynamique
$\sigma_u, \sigma_v, \sigma_w$	Intensités de turbulence
L_u, L_v, L_w	Longueurs longitudinales de turbulence
v	Vitesse du véhicule relative par rapport au courant d'air
v_m	Tension du moteur
i	Courant du moteur
R	Résistance interne du moteur
L	Inductance du moteur
k_v	Constante de tension du moteur
τ	Couple moteur
k_T	Constante de couple
Q_f	Couple de frottement de charge
D_v	Coefficient d'amortissement visqueux du moteur
U_{bat}	Tension aux bornes de la batterie
E_m	Tension en circuit ouvert
E_0	Tension en circuit ouvert à pleine charge
R_{int}	Résistance interne
Q	Capacité de la cellule en Ah

Introduction générale

Les drones miniatures à voilure tournante ont le potentiel de changer le mode de vie de manière positive. Ils peuvent se déplacer dans l'espace (en trois dimensions) pour collecter des informations avec des capteurs embarqués et même interagir physiquement avec leur environnement en utilisant des pinces embarquées. Les drones à voilures tournantes sont actuellement utilisés pour de nombreuses tâches que les drones miniatures pourraient effectuer, comme la surveillance de la circulation, la culture et la météo; l'inspection des bâtiments, des ponts et des lignes électriques; activités générales de surveillance; ainsi que des applications de photographie aérienne et de vidéographie. Les véhicules aériens à voilure tournante n'ont pas besoin d'un pilote à bord, ce qui signifie qu'ils peuvent être plus petits, plus sécurisés et moins chers.

La taille petite de ces véhicules leur permet de fonctionner à l'intérieur, à l'extérieur et dans des espaces restreints. Cette capacité sera particulièrement utile dans des situations dangereuses telles que la recherche de survivants dans des bâtiments endommagés, pénétration et inspection de bâtiments avec des adversaires armés et la collecte d'informations dans des bâtiments contaminés par des substances biologiques ou nucléaires. Dans ces scénarios, la capacité de créer une conscience situationnelle sans jamais avoir à mettre une vie humaine en danger est extrêmement précieuse.

C'est vrai qu'un robot mobile pourrait accomplir certaines de ces tâches décrites. Dans certains cas, les robots terrestres ont un avantage considérable par rapport aux véhicules aériens, car ils peuvent transporter de plus grandes charges utiles, sont généralement plus robustes aux collisions avec l'environnement, ne nécessitent pas de détection active pour rester au même endroit et ne nécessitent pas une utilisation continue d'énergie pour rester en altitude. Cependant, la capacité à voler permet à un drone à voilure tournante d'accéder à des endroits impossibles à atteindre par un robot terrestre. De plus, la navigation sur un terrain accidenté et la montée d'escaliers sont des tâches difficiles pour les véhicules terrestres, mais dans la plupart des environnements, l'espace aérien est relativement libre, ce qui rend le problème de navigation plus facile pour les véhicules aériens.

Les véhicules aériens à voilure fixe sont une autre alternative aux drones à voilure tournante. Pour naviguer sur de longues distances, l'efficacité d'un véhicule à voilure fixe ne peut pas être dépassée par un véhicule à voilure tournante. Cependant, par rapport aux véhicules à voilure fixe, les drones à voilure tournante ont le net avantage de pouvoir flotter sur place (vol stationnaire). Les véhicules à voilure fixe, en comparaison, doivent constamment avancer pour produire une poussée. La capacité de vol stationnaire est précieuse pour naviguer dans des espaces restreints ainsi que pour ramasser et déposer

des charges utiles avec précision, et surtout pour la surveillance et la photographie.

Le Quadrotor est un type de drones miniatures à voilure tournante. Il a quatre hélices attachées à des moteurs typiquement montés dans une configuration croisée. Ils sont disponibles auprès de plusieurs entreprises en tant que véhicules de recherche et véhicules commerciaux ainsi que des jouets [1] [2] [3] [4]. Les bras à long moment sur lesquels reposent les hélices leur permettent de produire de grands moments de contrôle en effectuant des manœuvres agressives. L'un des plus grands avantages des Quadrotors est leur simplicité mécanique. En revanche, les hélicoptères classiques de petite taille et les hélicoptères coaxiaux nécessitent des mécanismes pour modifier l'inclinaison de l'hélice afin de produire des forces et des moments de contrôle.

Le Quadrotor est donc une excellente plateforme robotique pour les raisons mentionnées précédemment. La communauté de la robotique a vu une prolifération de recherches utilisant le Quadrotor. Des progrès ont été réalisés dans la conception, comme plusieurs groupes ont construit des Quadrotors de rang de 50 cm [5] [6] [7] et de rang de 10 cm [8] [9]. Il existe également des travaux sur la dynamique et le contrôle [5] [10] [11] [12], la planification [13] [14] [15], la génération de trajectoires [16] [17] [18] [19], apprentissage [18], transport de charge utile [20] [21] [22] et l'estimation d'état avec des capteurs embarqués [23] [24] [25]. Les travaux antérieurs seront discutés en détail dans le chapitre 1.

Les exigences croissantes en matière de sécurité et de performance ont nécessité le développement de systèmes de commande de vol (autopilot) sophistiqués. La commande adaptative est considéré comme l'une des techniques clés permettant le développement des technologies futures avec des performances et sécurité élevés, pour les drones miniatures à voilure tournante. Les systèmes de contrôle de vol adaptatifs offrent également de nombreux avantages par rapport à leurs homologues à gain fixe, notamment une meilleure performance, une robustesse accrue vis-à-vis les incertitudes et les perturbations externes, un temps de cycle de conception réduit, ainsi qu'un coût réduit. Ils sont capables d'atteindre ces objectifs grâce à leur capacité à ajuster les paramètres de la commande en fonction des mesures en ligne. Les premières utilisations de systèmes de commande de vol adaptatif ont utilisé des contrôleurs avec des propriétés de stabilité non éprouvées, parfois conduit à des conséquences désastreuses; par exemple le crash fatal du NASA X-15 en novembre 1967 [26]. Depuis ce temps, des recherches approfondies dans le domaine de la théorie de la commande adaptative ont permis la conception, l'analyse et la synthèse des systèmes adaptatifs stables. Nous entrons maintenant dans la phase où les systèmes de contrôle de vol adaptatifs ont atteint le niveau de maturité requis

pour l'application aux plates-formes de vol matérielles telle que les drones miniatures à voilure tournante.

MOTIVATIONS ET CONTRIBUTIONS

Bien qu'il n'y ait aucun doute que de grands progrès ont été réalisés sur tous les aspects des Quadrotors, ils ne sont pas encore utilisés pour de nombreuses tâches décrites précédemment et n'ont pas encore atteint leur plein potentiel. Pour atteindre cet objectif, beaucoup de travaux doivent être entamés pour faire avancer l'état de l'art dans tous les domaines de la technologie du Quadrotor. La motivation pour le travail présenté dans cette thèse est d'aborder certains défis liés au quadrotor et proposer des solutions à ces dernières. Plus précisément, le travail décrit ici représente des contributions à l'état de l'art en matière de commande et de génération de trajectoire optimale en terme d'énergie pour augmenter l'autonomie du véhicule.

Le travail présenté dans cette thèse a donné lieu aux contributions suivantes :

- La conception d'un modèle énergétique pour le Quadrotor, composé de la dynamique du mouvement du véhicule, la dynamique des moteurs et de la batterie. Le modèle proposé a été validé par simulation pour montrer la possibilité d'économiser de l'énergie.
- La formulation et la résolution de deux problèmes de commande optimale pour minimiser l'énergie et le temps d'une mission bien définie.
- La synthèse d'une loi de commande adaptative floue directe avec et sans observateur pour la poursuite de trajectoire d'un Quadrotor. L'approche de commande proposée est robuste vis-à-vis les perturbations externes telles que les rafales de vent et les perturbations internes telles que les variations paramétriques.
- La synthèse d'une loi de commande adaptative floue robuste basée sur l'observateur de perturbations. Cette approche a été validée en simulation pour montrer la robustesse contre les perturbations externes ainsi que les variations paramétriques.
- La proposition d'un observateur de perturbation pour l'estimation des perturbations externes à savoir les rafales de vent. L'estimateur sera utilisé dans la loi de commande dont le but est de compenser l'effet des perturbations externes.

PUBLICATIONS

Le travail réalisé dans cette thèse à fait l'objet des publications suivantes:

Publication internationale

- **F. Yacef**, O. Bouhali, M. Hamerlain and N. Rizoug, "Observer-based Adaptive Fuzzy Backstepping Tracking Control of Quadrotor Unmanned Aerial Vehicle Powered by Li-ion Battery" *Journal of Intelligent & Robotic Systems*, vol. 84, no. 1, pp. 179-197, 2016. **IF : 1.512**

Conférences internationales

- **F. Yacef**, N. Rizoug, O. Bouhali and M. Hamerlain, "Optimization of Energy Consumption for Quadrotor UAV" 9th International Micro Air Vehicules Conference and Flight Competition *IMAV'17*, Toulouse, France 2017, pp. 215-222.
- **F. Yacef**, N. Rizoug, L. Degaa, O. Bouhali and M. Hamerlain, "Trajectory Optimisation for a Quadrotor Helicopter Considering Energy Consumption" 4th International Conference on Control, Decision and Information Technologies *CoDiT'17*, Barcelona, Spain 2017, pp. 1030-1035.
- **F. Yacef**, O. Bouhali and M. Hamerlain, "Adaptive Fuzzy Tracking Control of Unmanned Quadrotor via Backstepping" 23st IEEE International Symposium on Industrial Electronics *ISIE'14*, Istanbul, Turkey 2014, pp. 40-45.
- **F. Yacef**, O. Bouhali and M. Hamerlain, "Adaptive Fuzzy Backstepping Control for Trajectory Tracking of Unmanned Aerial Quadrotor" IEEE International Conference on Unmanned Aircraft Systems *ICUAS'14*, Orlando, FL, USA 2014, pp. 920-927.

ORGANISATION DE LA THESE

Ce rapport de thèse est divisé en cinq chapitres:

D'abord, nous commençons par un état de l'art sur les drones miniatures dans le chapitre 1. Nous présentons brièvement l'historique des drones miniatures à voilure tournante et les applications civiles potentielles de ces véhicules spécifiquement le Quadrotor. Ensuite, nous donnons un état de l'art sur les travaux développés sur la commande

du Quadrotor et nous présentons aussi quelques notions sur le système flou utilisé dans ce travail ainsi qu'un rappel sur la commande adaptative floue.

Le chapitre 2 est consacré à la modélisation du Quadrotor. Nous y présentons le modèle cinématique ainsi que le modèle dynamique d'un corps rigide dans l'espace; ensuite, les forces et les moments importants qui agissent sur le Quadrotor. Un modèle dynamique complet du Quadrotor et un autre simplifié utilisé dans la synthèse des lois de commande sont aussi présentés dans ce chapitre.

Dans le chapitre 3, nous proposons un modèle énergétique pour le Quadrotor, composé de la dynamique du mouvement, la dynamique des moteurs ainsi que la dynamique de la batterie. Le modèle proposé est validé à travers deux missions pour montrer la possibilité d'économiser de l'énergie. Deux problèmes de commande optimale sont formulés et résolus afin de calculer l'énergie minimale ainsi que le temps minimal. Dans le premier problème, nous cherchons à trouver les entrées de commande et la trajectoire du véhicule entre deux configurations initiales et finales qui minimisent l'énergie consommée durant une mission spécifique. Alors que dans le second, le temps minimal et la trajectoire du véhicule sont calculés pour une énergie désirée.

Dans le chapitre 4, nous synthétisons deux lois de commande adaptatives floues directes pour une classe de système multi-rotors à savoir le Quadrotor. La première commande est sans observateur, tandis que la deuxième est avec observateur. L'idée de base est d'utiliser un système flou adaptatif pour approximer la dynamique incertaine du véhicule. Nous présentons aussi dans ce chapitre la conception d'un observateur d'état pour mesurer les états non mesurables. Les performances des deux schémas de commande proposés dans ce chapitre sont validées par simulation.

Le chapitre 5 introduit une loi de commande adaptative floue robuste à base d'observateur de perturbation, pour surmonter le problème des perturbations externes à savoir la rafale de vent. Il présente la conception de cette approche de commande qui repose sur l'estimation des perturbations externes, ainsi que la validation de cette approche de commande par simulation sur un Quadrotor.

Chapitre 1

État de l'art sur les drones miniatures et les systèmes flous

1.1 Introduction

Nous allons présenter dans ce chapitre un aperçu sur l'évolution, le développement et les applications de drones miniatures à voilure tournante, Ainsi que les travaux remarquables proposés dans la littérature pour ce type de drone. Nous allons restreindre notre étude sur un type de drones miniatures à voilure tournante à savoir le Quadrotor, où nous présentons les travaux des grands laboratoires qui travaillent sur cette plate-forme de recherche. Cette étude nous permettra de situer notre travail par rapport aux travaux antérieurs. Nous allons présenter aussi quelques descriptions des approximateurs flous et la commande adaptative floue.

Par la suite dans ce chapitre, nous allons introduire dans la section 1.2 un état de l'art sur les drones miniatures à voilure tournante. Dans la section 1.3 une étude bibliographique sur la commande de drone miniature à voilure tournante précisément le Quadrotor est exposée. Nous rappelons dans la section 1.4 quelques notions sur les systèmes flous, pour mieux présenter les approches de commande développées dans ce travail, . La section 1.5 présente un rappel sur la commande adaptative floue.

1.2 Drones miniatures à voilure tournante

1.2.1 Contexte historique

Le concept du drone a été créé par *Léonard De Vinci* 1488 (voir la figure 1.1a). Il a appelé le véhicule *Non il volo umano*, ce qui signifie *vol non humain*. Malgré le concept élaboré par Da Vinci, l'idée n'a pas été mise en exécution que beaucoup plus tard. La première utilisation enregistrée d'un véhicule aérien sans pilote pour le Combat dans la guerre a eu lieu le 22 Août 1849, lorsque les Autrichiens ont attaqué la ville italienne de Venise avec des ballons sans pilote chargés d'explosifs (voir la figure 1.1b). Le premier avion sans pilote, destiné à être utilisé comme des missiles, a été construit pendant et peu après la première guerre mondiale. En Septembre 1916, l'avion automatique de *Hewitt-Sperry*, autrement connu comme la "*bombe volante*" a fait son premier vol, démontrant le concept d'un aéronef sans pilote.

Probablement le premier aéronef contrôlé par Radio Commande (RC) réussi était *Fairey III F* qui était modifié avec plus dièdre dans le cadre d'un programme de l'établissement Royal d'aéronef en 1931. L'aéronef est devenu connu comme *Fairey Queens* qui explique pourquoi ultérieurement les drones RC ont été appelés *Queen Bee* et *Queen Wasp*. La première production de drone à grande échelle était le produit de *Reginald*

Denny, qui a produit un modèle d'avion RC en 1934 (voir la figure 1.1c). Dans la période après la seconde guerre mondiale, les avions RC ont suivi le succès de drone cible OQ-2 avec une autre série très réussie de drones cibles à piston, connus sous le nom de *Basic Training Target (BTT)*. Ils furent ensuite employés pour des missions de reconnaissance dans les années 1960 (drones *Firebee* et *Lightning Bug*). Leur utilisation pour des missions de surveillance s'est ensuite répandue lors de nombreux conflits. Depuis, de nombreux drones ont été développés, et leur usage à des fins civiles a également débuté.

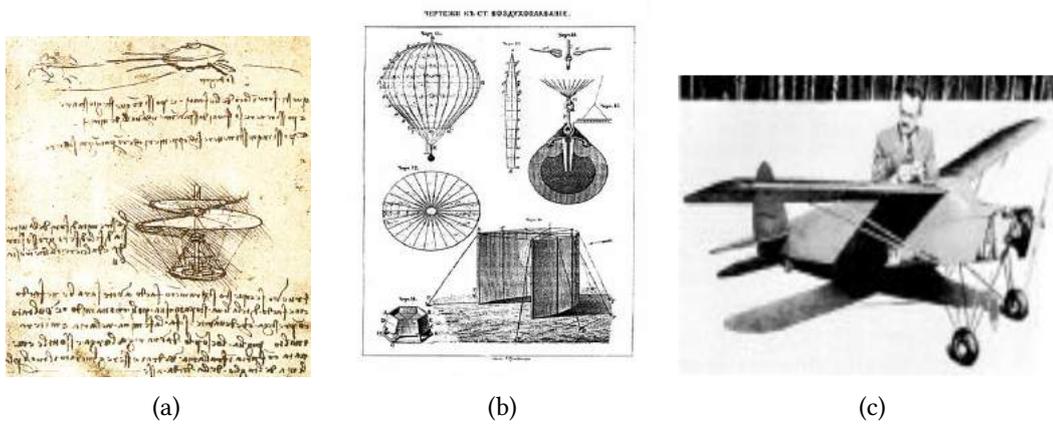


Figure 1.1: (a) véhicule aérien de Leonardo Da Vinci, (b) Bombardement par Balloon en 1848, (c) RP.1 aéronef de Reginald Denny

Le *Gyroplane No.1* est l'un des premiers Quadrotors avec pilote, construit par *Louis* et *Jacques Breguet* avec l'aide du professeur *Charles Richet* en 1907 en France (voir figure 1.2). Ce n'était pas un hélicoptère libre, ni contrôlable ni orientable. Mais c'était la première fois qu'un dispositif à voilure tournante s'était levé et un pilote dans l'air [27]. Dans les années 1920 en France, *Étienne Oehmichen* a conçu plusieurs Quadrotors [27]. Le *Oehmichen No.2* est probablement le premier hélicoptère fiable (voir figure 1.3a). L'américain *George de Bothezat* a construit *Convertawings Model A* (voir la figure 1.3b), et le premier vol a eu lieu en 1956. Il a également été le premier hélicoptère à quatre rotors qui a démontré un vol stationnaire [28].

Le *Curtiss-Wright VZ-7* a été conçu par la société *Curtiss-Wright* pour l'armée américaine en 1958 (voir la figure 1.4a). Il a été contrôlé en modifiant la poussée des quatre hélices [29]. En 1963, la société *Curtiss-Wright* a développé le *X-19* pour l'armée de l'air des États-Unis (voir la figure 1.4b). Cependant, il a été détruit lors d'un accident en 1965, et le programme de *X-19* a ensuite été annulé [30].

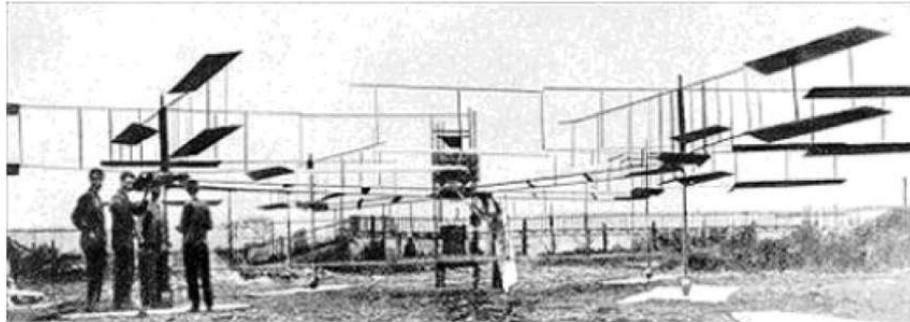


Figure 1.2: Gyroplane No.1



(a)



(b)

Figure 1.3: (a) Oehmichen No.2, (b) Convertawings Model A



(a)



(b)

Figure 1.4: (a) Curtiss-Wright VZ-7, (b) Curtiss-Wright X-19

Au cours des dernières décennies, divers drones miniatures à voilure tournante ont été conçus pour des recherches scientifiques et des applications civiles. Il existe de nombreux drones commercialisés avec succès, tels que *Draganflyer*, *Hummingbird*, *AR.Drone*, *Phantom*, *Gauji 330X*, etc.



Figure 1.5: Draganflyer X4-P

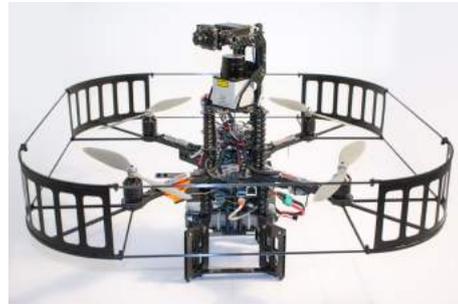
Draganfly Innovations Inc a conçu plusieurs Quadrotors depuis 1999. Il existe plusieurs types de Quadrotor: Draganflyer *X4-P*, Draganflyer *Guardian* et Draganflyer *Commander*. Ils sont construits en utilisant des fibres de carbone robustes, des pièces en nylon injectées en verre et des moteurs à courant continu sans balais (brushless DC motors). Les dimensions des Quadrotors Draganflyer sont entre 59cm et 87cm. Le poids avec la batterie varie de 1050g à 2750g, et la capacité de charge utile est de 420g à 1000g. La batterie embarquée est habituellement de type Lithium Polymer (LiPo) avec une tension maximale de 14,8V et une capacité varie de 2700mAh à 5400mAh, ce qui permet à un Quadrotor de voler environ 30 à 40 minutes. Le Draganflyer *X4-P* est équipé d'un puissant processeur embarqué avec onze capteurs: trois gyroscopes, trois accéléromètres, trois magnétomètres, un capteur de pression barométrique et un récepteur GPS. Tous ces capteurs fournissent constamment des données à l'ordinateur de vol embarqué. Les Quadrotors de chez Draganfly sont utilisés dans de nombreuses applications, telles que la sécurité publique, la maintenance et l'inspection industrielles, la photographie et la vidéographie, etc. De plus, ils sont utilisés dans la recherche scientifique dans de nombreuses universités, comme MIT, université North Dakota, université Texas A&M et l'université McGill [2].

Ascending Technologies propose également plusieurs Quadrotors et drones depuis 1998 [1]. L'AscTec *Flacon* est conçu pour les enregistrements professionnels d'images et de vidéos. L'AscTec *Firefly*, *Pelican* et *Hummingbird* sont conçus pour différents domaines de recherches scientifiques (voir la figure 1.6). Le *Pelican* est conçu avec une puissance maximisée et beaucoup d'espace pour un développement ultérieur. Le *Hummingbird* est conçu pour des manœuvres de vol très dynamiques et risquées, il est très robuste et sa structure simple est facile à réparer. Les poids de *Pelican* et *Hummingbird* avec batterie sont respectivement de 1650g et 710g et la charge utile maximale est de

200g et 650g. Le *Hummingbird* dispose de quatre moteurs à courant continu sans balais *HACKER X-BL 52s* et d'une batterie de 3 cellules 11.1V 2100mAh Lithium Polymère (LiPo) qui permet au véhicule de voler environ 20 minutes. Le *Hummingbird* dispose de deux microcontrôleurs ARM9 et dispose de neuf capteurs: trois gyroscopes, trois accéléromètres, un magnétomètre 3D, un capteur de pression barométrique et un récepteur GPS. Les drones de chez *AscTec* sont largement utilisés dans la recherche scientifique, tel que l'Université de Pennsylvanie et l'École polytechnique fédérale de Zurich. Ils fournissent un programme utilisateur qui fonctionne à 1kHz pour permettre aux utilisateurs de créer leur propre algorithme de commande.



(a)



(b)



(c)



(d)

Figure 1.6: AscTec : (a) Hummingbird, (b) Pelican, (c) Falcon, (d) Firefly

Le *AR. Drone* est produit par *Parrot SA* depuis 2010 (voir la figure 1.7a). Il équipe deux caméras HD intégrées et envoie des images en temps réel au contrôleur de vol par Wi-Fi. C'est un Quadrotor bien développé contrôlé à distance par une application d'interface graphique conviviale sur *iPhone*, ou *iPad* [31]. *AR. Drone*, nom complet Augmented Reality Drone, est conçu pour les jeux vidéo et les divertissements à domicile. Le prix est inférieur à 300 euros. Il est équipé d'une batterie de 3 cellules 11.1V 1000mAh Lithium Polymère (LiPo) qui permet au véhicule de voler environ 10 à 15 minutes. Le système

comporte deux caméras, trois gyroscopes, trois accéléromètres, un sonar et deux micro-contrôleurs. En 2011, Bristeau et al. ont exposé le système de navigation et de contrôle intérieur dans leur papier [32]. Le système est contrôlé par plusieurs contrôleurs PID.

DJI Innovations produit également des Quadrotors, des multicopters et des contrôleurs de vol à usage commercial et récréatif. Le Quadrotor *Phantom* a une dimension de 35cm et hauteur de 19cm (voir figure 1.7b). Il est équipé d'une batterie de 3 cellules avec 2200mAh Lithium Polymère (LiPo) qui permet au véhicule de voler environ 15 minutes. L'angle d'inclinaison maximal du *Phantom* est de 45 degrés. La vitesse maximale de montée ou de descente est de 6m/s et la vitesse maximale du vol est de 15m/s . Le *Phantom* dispose d'un système intégré de contrôle de vols, le système de pilote automatique Naza-M, pour assurer la stabilité du système. Il peut également être contrôlé par une radiocommande RC [4].



(a)



(b)

Figure 1.7: (a) AR. Drone, (b) Phantom

De nombreux projets open-source (OSP) sont également développés sur les Quadrotors, tels que *Arducopter* [33], *Paparazzi* [34], *Pixhawk* [35], *Mikrokopter* [3], etc. Les OSP utilisent des sites d'hébergement communautaire pour générer des codes ou des schémas librement. *Arducopter* fournit un logiciel basé sur une interface utilisateur graphique (GUI) pour ajuster les gains du contrôleur et afficher les informations de vol. *Paparazzi* fournit neuf Cartes de pilote automatique différentes qui sont développées par l'université ENAC et un GUI basé sur GCS avec scripts de chemin de vol. Le projet *Pixhawk* dispose d'un équipement de vision par ordinateur développé par le groupe de vision par ordinateur de ETHZ. *Mikrokopter* est fourni par *HiSystem GmbH* en 2006. Il dispose d'un logiciel basé sur l'interface graphique pour le réglage du gain.

1.2.2 Applications

L'utilisation des drones miniatures était très limitée dans la dernière décennie, essentiellement à cause des réglementations restreignant ou interdisant leur utilisation au sein de l'espace aérien. Dans les dernières années ce domaine a connu un intérêt considérable chez les industriels et les chercheurs, cet intérêt est motivé par les récentes avancées technologiques qui rendent possible la conception de systèmes performants dotés de réelles capacités de navigation autonome, à des coûts non prohibitifs.

Les applications sont nombreuses. Elles concernent en premier lieu les domaines de la sécurité (surveillance de l'espace aérien, du trafic urbain et interurbain), la gestion des risques naturels (surveillance de l'activité des volcans), la protection de l'environnement (mesure de la pollution de l'air, surveillance des forêts), l'intervention dans des sites hostiles (milieux radioactifs, déminage des terrains sans intervention humaine), la gestion des grandes infrastructures (barrages, lignes à haute tension, pipelines), l'agriculture (détection et traitement des cultures) et la prise de vue aérienne dans la production de films. Toutes ces missions nécessitent un contrôle performant de l'appareil et par conséquent des informations précises sur son état absolu et /ou relatif à son environnement [36].

1.2.2.1 Application de drones dans l'agriculture de précision

La valeur totale des solutions basées sur les Drones dans toutes les industries d'applications est significative, plus de 127 milliards de dollars, selon une récente analyse de PwC. Parmi les domaines les plus prometteurs, il y a l'agriculture, où les drones offrent la possibilité de relever plusieurs défis majeurs. La technologie Drone donne à l'industrie agricole une transformation de haute technologie, avec une planification et une stratégie basées sur la collecte et le traitement des données en temps réel. PwC estime à 32,4 milliards de dollars le marché des solutions à base de Drone dans l'agriculture de précision.

A l'échelle mondiale, les usages peuvent être classés en deux catégories principales : le traitement (pulvérisation en général), et le diagnostic (recueil de données géo-référencées sur l'état des cultures). Ces deux branches impliquent des différences technologiques sur l'architecture des drones :

1. pour le diagnostic, l'appareil emporte le bon capteur (en visible ou en infrarouge) à une altitude assez élevée pour éviter tous les obstacles et peut couvrir de grandes surfaces rapidement, ce qui est le cas des drones à voilure fixe de type *mini-avion*.
2. pour le traitement, l'appareil doit être capable de vols stationnaires, avec une

charge utile plus importante, tout en évitant les obstacles proches du sol, ce qui est le cas des drones à voilures tournantes de type *multi-rotors*. En France, l'épandage aérien de pesticides est interdit, et la tendance est plutôt à la suppression des dérogations existantes, compte tenu des risques de contamination pour les habitations et pour les cours d'eau en particulier.



Figure 1.8: Utilisation de drone dans l'agriculture: (a) Drone à voilures fixes, (b) Drone à voilures tournantes

1.2.2.2 Application de drones dans la photographie

Les drones sont également utilisés pour la photographie et la vidéo aérienne. Le drone *AscTec Falcon* est spécialement conçu pour cette application [1]. Un Quadrotor peut être utilisé à des fins commerciales, tels que le tournage d'un documentaire sur la faune, la vue aérienne pour une publicité d'un bien immobilier et la vidéo aérienne pour un événement sportif. Il peut également être utilisé à des fins personnelles pour enregistrer des photos et des vidéos pour souvenir. Pour la photographie et la vidéo aérienne commerciale, la haute qualité des photos et des vidéos devrait être assurée. Les images ne doivent pas être floues, et les vidéos devraient être prises pendant un mouvement lisse du véhicule.

L'obtention d'images et de vidéos aériennes de haute qualité peuvent être un grand défi. Cela dépend de la configuration de la caméra embarquée et, plus important, de la stabilité du véhicule lors du vol stationnaire ou du déplacement. Les drones doivent flotter ou se déplacer avec les plus petites vibrations pour assurer la qualité de l'image. Par conséquent, la commande du système devrait être capable de tolérer les bruits de mesures provenant des capteurs, tels que les accéléromètres et les gyroscopes, et de fournir les meilleures performances. De plus, comme les prises de vue sont prises dans

différentes circonstances et dans différentes conditions météorologiques, la stabilité du système est très importante. Il peut y avoir des perturbations météorologiques, telles que le vent, la température et l'humidité. Le système de commande de drone devrait être capable d'atténuer ces perturbations externes avec des erreurs de suivi et des vibrations les plus petites possibles.



Figure 1.9: (a) Bebop de chez parrot, (b) Phantom de chez DJI

Dans ces applications, les drones ont besoin d'une durée de vol assez longue pour terminer les tâches. Le temps de vol dépend de deux facteurs: la capacité de la batterie embarquée et la consommation d'énergie du système. Les drones commercialisés tels que les Quadrotors utilisent généralement des batteries au lithium-polymère (LiPo), qui ont une capacité beaucoup plus légère et plus élevée qu'une batterie rechargeable conventionnelle de forme équivalente. La batterie de 3 à 6 cellules, 12V et 3000mAh est la configuration la plus commune utilisée dans beaucoup de systèmes de Quadrotor, tel que *Draganfly X4-P* [2] et *AscTec Hummingbird* [1]. Elle permet à un drone de type Quadrotor de voler environ 15 à 20 minutes. Le *Draganfly X8* a une puissance améliorée, une batterie lithium-polymère de 5400mAh, pour assurer une durée de vol plus longue. Cependant, une capacité de batterie plus importante signifie une charge plus lourde et un prix plus cher. Afin d'étendre la durée du vol, nous pouvons également réduire la consommation énergétique du système en utilisant une CPU (Central Processing Unit) moins puissante. Cependant, un processeur moins puissant signifie moins de capacité de traitement. D'autres systèmes, comme le système de vision, nécessitent de nombreuses ressources de traitement. Par conséquent, le système de commande de drone devrait avoir une structure simple avec de bonnes performances. De plus, en particulier dans

les tâches de recherche et de sauvetage, il existe de nombreux obstacles inattendus. Le drone doit avoir une réponse rapide pour éviter les collisions. Donc, la méthode de commande devrait avoir une adaptation rapide du système aux changements brusques.

1.3 Étude bibliographique sur la commande de Quadrotor

1.3.1 Groupes de recherche

Depuis 2002, plusieurs centres de recherche et de groupes de spécialistes en aéronautique ont commencé à publier les premiers résultats à propos de la modélisation et de la commande de l'hélicoptère à quatre rotors (Quadrotor). Dans les paragraphes suivants, nous allons décrire très succinctement, les travaux de différents groupes qui sont intervenus dans la modélisation et la commande de Quadrotor. Il faut noter que cette liste n'est pas exhaustive.

Université Nationale d'Australie : cette équipe de recherche a été une des premières à aborder le problème de la stabilisation de l'hélicoptère à quatre rotors, ils ont construit la plate-forme *X-4* en 2004 (voir la figure 1.10a). Plus de détails sur *X-4* peuvent être trouvés dans la thèse et les documents de *Pounds* [37] [38]. Ensuite, ils se sont intéressés à donner une modélisation très détaillée en considérant les effets aérodynamiques et le battement des hélices de Quadrotor [39], ainsi que la modélisation de Quadrotor de taille large [40].

École Polytechnique Fédérale de Lausanne, Suisse : ce groupe de recherche s'est intéressé à la conception du "hardware" embarqué dans des hélicoptères à quatre rotors, en utilisant différentes architectures, telles que par exemple des microcontrôleurs, des DSP et des FPGAs [41]. En ce qui concerne la commande, ils ont reporté l'application de lois de type PID et LQ pour la stabilisation en attitude de l'hélicoptère à quatre rotors [42]. Ensuite, ils ont appliqué des approches plus complexes, telles que la commande backstepping et la commande à modes glissants [43]. Différents résultats en simulation et en temps réel ont été montrés. Cependant un modèle linéaire pour l'attitude a été utilisé et les aspects de robustesse par rapport aux perturbations externes et aux incertitudes du modèle n'ont pas été abordés. De plus, les lois de commande ne sont pas bornées.

Université de technologie de Compiègne, France : bien que cette équipe ait réalisé une modélisation de l'hélicoptère à quatre rotors basée sur un formalisme de Lagrange le point fort de leur travail concerne la commande. La stratégie de commande reportée dans [44] consiste à ramener les équations qui représentent la dynamique du système à une chaîne d'intégrateurs et ensuite à appliquer une stratégie proposée dans [45] qui

concerne la stabilisation des systèmes linéaires avec des commandes bornées. La plateforme expérimentale de cette équipe consiste en l'hélicoptère à quatre rotors *Dranganfly III* vendue chez *Dranganfly Innovation Inc* [2]. Une commande d'attitude bornée pour le Quadrotor a été proposée dans [46], où la loi de commande est obtenue en utilisant une technique backstepping. Néanmoins, les auteurs supposent un modèle linéaire pour la dynamique de la rotation.



Figure 1.10: (a) ANU X4, (b) STARMAC II

Université Lakehead, Canada : Le travail reporté par cette équipe concerne la stabilisation en attitude d'un robot volant de type VTOL à quatre rotors. L'approche de commande proposée dans [47] est basée sur les quaternions, et la compensation des couples gyroscopique et de Coriolis. On trouve deux contrôleurs, le premier est un PD et le deuxième un PD^2 , où l'action proportionnelle est en terme de vecteur quaternion et les deux actions dérivées sont en terme de la vitesse angulaire du véhicule et de la vitesse du vecteur quaternion. Les lois de commande montrent une stabilité asymptotique globale et une stabilité exponentielle locale. Les travaux récents de cette équipe de recherche porte sur l'estimation et la stabilisation d'attitude de drones miniatures de type VTOL [48] [49] [50].

Université de Stanford, USA : Stanford Testbed of Autonomous Rotorcraft for Multi-Agent Control (STARMAC) [6], est le produit réalisé par cette équipe de recherche, c'est un banc d'essai multi-véhicule utilisé pour valider de nouveaux concepts en matière de contrôle multi-agent. Les travaux menées sur STARMAC couvrent plusieurs domaines, y compris la recherche coopérative, le contrôle et réseau de capteur [51], l'évitement des obstacles coopératifs [52], planification de trajectoire optimale, et la génération de trajec-

toires acrobatiques. Les travaux en cours se concentrent sur l'intégration des capacités de perception sur STARMAC, pour permettre au véhicule de comprendre, de réagir et de raisonner sur leur environnement.

Université de Pennsylvanie, USA : ce groupe a réalisé un travail intéressant avec le professeur Vijay Kumar dans laboratoire GRASP, ils ont utilisé la plate-forme de Quadrotor destiné à la recherche *Hummingbird* (Figure 1.6a) pour développer plusieurs applications. Dans [53] les auteurs ont développé un algorithme de commande, une méthodologie de saisie, ainsi que la fabrication d'un mécanisme pour la prise des objets. Le problème de commande en coopératives de plusieurs Quadrotors pour une tâche de saisie et transportation est traité dans [54]. D'autres problèmes ont été traités telles que, la génération de trajectoire et réalisation des mouvements agressives [55], commande et génération d'une trajectoire demi-circulaire dans un environnement contraint [16].

École polytechnique fédérale de Zurich, Suisse : dans l'institut de systèmes dynamiques et de contrôle, Le professeur Raffaello D'Andrea et son équipe ont réalisé des travaux remarquables, ils ont utilisé la même plate-forme que le laboratoire GRASP (*Hummingbird*) pour traiter plusieurs problématiques et défis des drones miniatures, on peut citer : l'apprentissage itératif [56] [57], où le système autonome apprend de leur expérience passée à travers des algorithmes d'apprentissage qui lui permettent avec le temps de réaliser des manœuvres sophistiqués avec précision. Génération de trajectoire et commande en temps réel [17] [58], commande optimale en terme de temps [59], et la commande en formation pour des tâches d'assemblages, de lancement et attrape de balle [60] [61] [21].

1.3.2 Guidage et pilotage

Au cours des dix dernières années, les applications des drones à voilures tournantes sont demeurées comme des sources de motivation les plus influentes pour une analyse et un contrôle rigoureux des systèmes mécaniques et non linéaires. Les drones à voilures tournantes sont naturellement instables. La stabilisation de ces machines est donc une étape cruciale dans la conception des aéronefs autonomes. Développer un contrôleur de vol consiste à synthétiser des algorithmes qui vont produire les couples et les forces de commande pour deux raisons: 1) Stabiliser le véhicule et rejeter les perturbations internes et externes ; 2) fournir les accélérations nécessaires pour que le drone suive une trajectoire désirée et réaliser sa mission.

Prenons un exemple simple où le Quadrotor est chargé de se déplacer en toute sécurité à partir du point A au point B en évitant les obstacles (figure 1.11). On peut décomposer

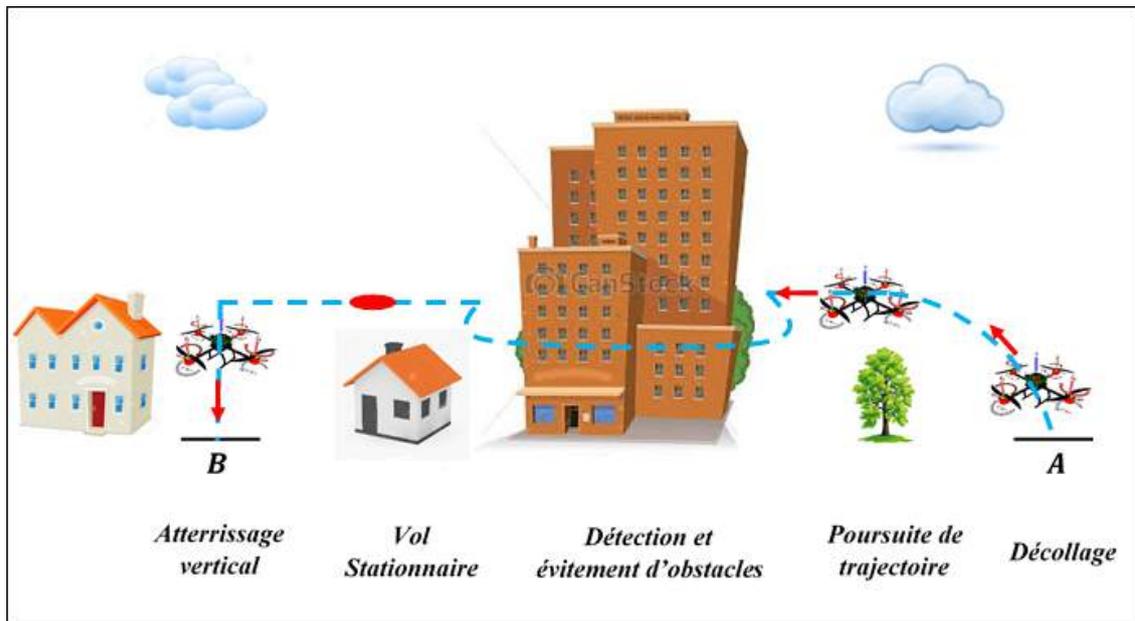


Figure 1.11: Exemple de mission de navigation simple

cette mission en tâches élémentaires comme le décollage autonome, suivi de trajectoire de référence, évitement et détection des obstacles, vol stationnaire, détection de zone d'atterrissage sécurisée et atterrissage autonome.

Pour qu'un robot volant autonome réalise cette mission, il a besoin d'avoir des capteurs pour l'auto-localisation, et des capteurs pour la perception de l'environnement et de l'interaction (voir figure 1.12). D'un autre côté, l'architecture de contrôleur embarqué doit intégrer différents modules : le contrôleur de bas niveau (stabilisation et suivi de trajectoire), comportements réactifs pour l'évitement des collisions, et un contrôleur de haut niveau (système de navigation) pour la planification de mission et la génération des trajectoires de référence.

Le contrôleur de bas niveau est dédié à l'augmentation de stabilité, rejet de perturbations et l'amélioration de la réponse dynamique lorsque le pilote ou le contrôleur de haut niveau (système de guidage) fournit des commandes et des trajectoires de référence. Les variables d'état sont généralement la position, les angles de rotation, la vitesse linéaire et les vitesses angulaires. Le contrôleur de bas niveau transforme ces mesures à des forces et des moments qui exercent sur le corps rigide du drone.

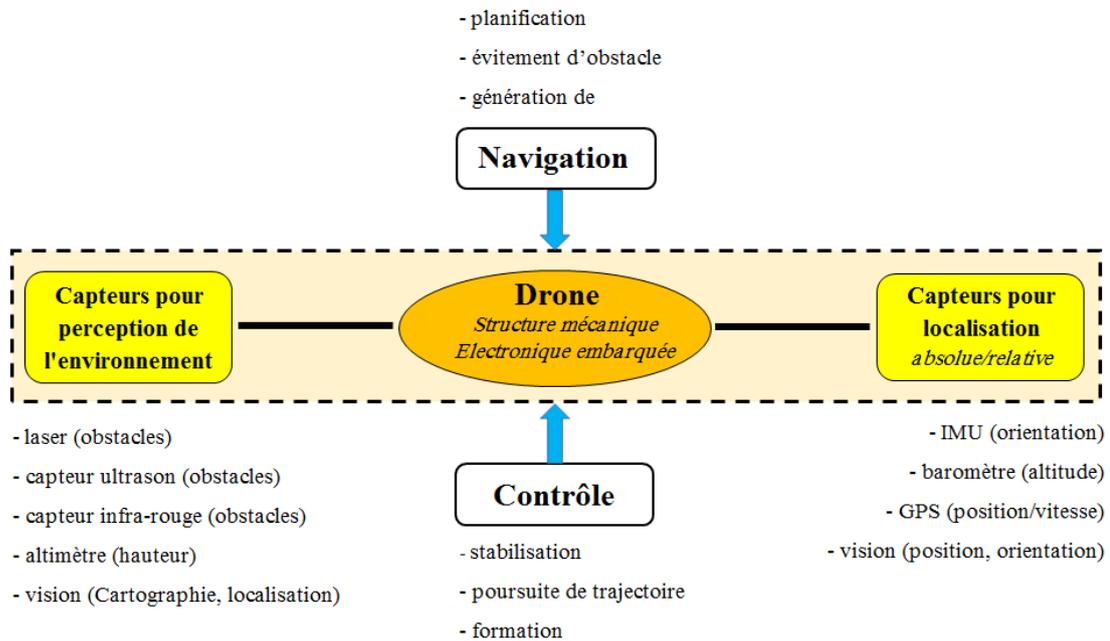


Figure 1.12: Capteurs et modules requis pour un vol autonome

1.3.3 Techniques de commande

Depuis les années 2000, différentes stratégies de commande et guidage ont été proposées. Ils peuvent être classés dans deux catégories: méthodes de contrôle basées sur l'apprentissage et techniques de contrôle basées sur la modélisation.

1.3.3.1 Techniques de commande basées sur l'apprentissage

Ces techniques de commande comprennent la logique floue et les approches des réseaux de neurones. Le modèle dynamique de l'UAV n'est pas nécessaire, mais plusieurs essais sont nécessaires pour former le système de contrôle. Les commandes basées sur la logique floue ont été mis en œuvre avec succès sur le Quadrotor. une commande adaptative basée sur la logique floue est proposée dans [62], le modèle de Takagi Sugeno est utilisé dans [63] [64] pour la stabilisation et l'estimation de l'attitude du Quadrotor, le problème de poursuite de trajectoire est traité [65], d'autres auteurs combine la logique floue avec les techniques de commande non linéaire tel que backstepping et mode glissant [66]. Plusieurs commandes basées sur les réseaux de neurones ont été développées pour le Quadrotor [67] [68] [69]. Dans [70], Madani, a développé une commande adaptative basée sur les techniques backstepping et les réseaux de neurones.

1.3.3.2 Techniques de commande basées sur le modèle

Ces méthodes de commande sont largement utilisées dans les systèmes aérospatiaux et sont basées sur le modèle dynamique des véhicules. Différents contrôleurs pour les UAVs ont été proposés en utilisant différentes techniques de commande linéaires classiques et les stratégies avancées de commande non linéaires. Il a été démontré que le modèle non linéaire de Quadrotor ne peut pas être converti en un système linéaire contrôlable via la linéarisation exacte, mais la commande par linéarisation entrées/sorties a été appliquée avec succès sur le Quadrotor [71]. Dans [42], le PID et les techniques de commande LQR ont été utilisées pour la commande d'un petit UAV. Un modèle de commande prédictive non linéaire a également été utilisé pour le suivi de trajectoire [72]. Il est à noter que les techniques de commande non linéaires basées sur la linéarisation sont plus générales et aucune garantie de robustesse contre les incertitudes des modèles ou des perturbations n'est fournie.

Les effets des incertitudes paramétriques et les dynamiques non modélisées ont été traités à l'aide des systèmes d'identifications [41] [44] [55] [73] [74] [75] ou des techniques de contrôle robustes [71] [76] [77]. Une boucle de commande H_∞ robuste [76] a été implémentée et testée sur l'hélicoptère à quatre rotors, effectuant ainsi un vol robuste et de suivi de trajectoire précis. Les résultats obtenus sont très satisfaisants. Toutefois, la commande robuste utilisant H_∞ est connue pour être conservateur et présente un compromis entre les performances et la robustesse. En outre, la conception de cette commande utilise un modèle linéarisé extrait du vol stationnaire. Ainsi, une dégradation des performances apparaît lorsque l'hélicoptère effectue des manœuvres agressives. La commande robuste devient inapplicable lorsque les incertitudes de modélisation sont grandes. La commande adaptative est ensuite nécessaire pour gérer ces incertitudes importantes et variables dans le temps. De nombreux contrôleurs adaptatifs ont été proposés pour la commande de Quadrotor [70] [78] [79] [80] [81] [12]. Récemment, les techniques de commande adaptative ont été intensivement étudiées et utilisées pour l'estimation des incertitudes paramétriques [79], la stabilisation de l'attitude [81] et la poursuite de trajectoire [78]. Les réseaux de neurones et les mécanismes d'adaptation ont été utilisés pour compenser les erreurs de modélisation via backstepping [70]. La commande des drones avec des entrées bornées est un autre problème difficile à relever dans la pratique en raison de saturation des actionneurs. Les auteurs [46] ont proposé un contrôleur qui traite le problème de commande avec les entrées bornées. Les techniques de saturation imbriquées (Nested saturation) a été appliquée avec succès pour stabiliser un drone miniature à voilure tournante de type VTOL (Vertical Takeoff and

Landing) [82] [83] et l'hélicoptère à quatre rotors [44]. D'autres méthodes de contrôle non linéaire tel que les techniques du backstepping [84] [85], mode glissant [86] [87] ont été appliquées pour les hélicoptères à quatre rotors.

1.4 Système flou

La logique floue a été introduite par le professeur Lotfi Zadeh en 1965 comme une généralisation de la logique binaire [88]. L'intérêt de la logique floue réside dans sa capacité à traiter l'imprécis, l'incertain et le vague. Elle est issue de la capacité de l'homme à décider et agir de façon pertinente malgré le flou des connaissances disponibles. En effet, la logique floue a été introduite pour approcher le raisonnement humain à l'aide d'une représentation adéquate des connaissances. Un système flou est une fonction non linéaire reliant un vecteur de données d'entrée à un vecteur de sortie.

1.4.1 Description d'un système flou

Nous définissons d'abord une certaine terminologie. Un ensemble flou F est défini sur un univers de discours U , par une fonction d'appartenance μ_F qui prend ces valeurs dans l'intervalle $[0, 1]$. Cette fonction donne le degré d'appartenance de chaque élément $x \in U$ à l'ensemble flou F . Un ensemble flou peut être représenté comme un ensemble de paires ordonnées (F, μ_F) . La configuration de base d'un système flou est illustrée à la figure 1.13. Notant qu'il y a quatre modules principaux dans un système flou à savoir: interface de fuzzification, base de règles floues, moteur d'inférence flou et interface de défuzzification [89].

L'interface de fuzzification transforme la grandeur physique d'entrée en une grandeur floue. Plus précisément, étant donné (F, μ_F) un ensemble flou défini dans U , et $x \in U$ une entrée de l'interface de fuzzification alors, les sorties de l'interface de fuzzification sont $\mu_{F_i}(x_i)$ ($i = 1, 2, \dots, n$). Par exemple, la figure 1.14 définit une interface de fuzzification qui transforme la vitesse d'une voiture à des valeurs de fonction d'appartenance de la vitesse dans les trois ensembles flous étiquetés comme «lente», «moyenne» et «rapide». Il existe deux facteurs qui déterminent une interface de fuzzification: (1) le nombre d'ensembles flous définis dans l'univers de discours; et, (2) les fonctions d'appartenance spécifiques pour ces ensembles flous.

La base de règles floues est un ensemble d'énoncés linguistiques sous la forme de "SI un ensemble de conditions est satisfait, ALORS un ensemble de conséquences est déduit", où les conditions et les conséquences sont associées à des concepts flous (c.-à-d.,

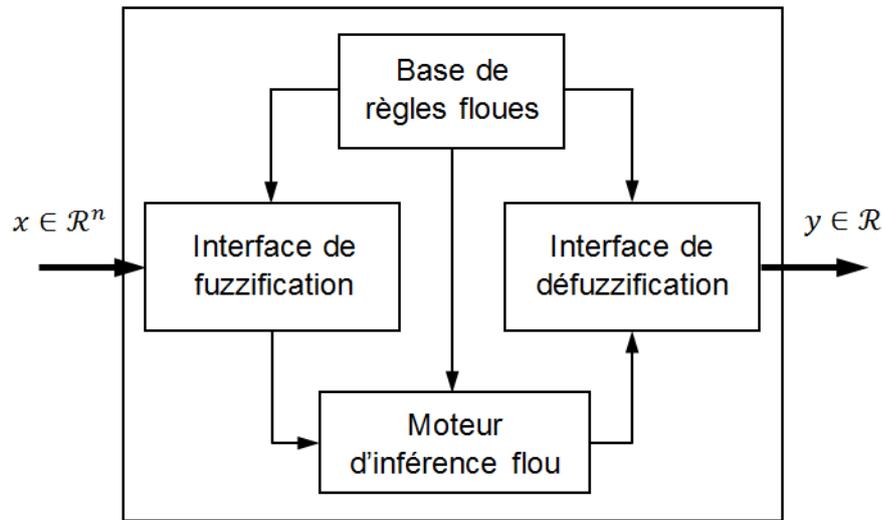


Figure 1.13: Configuration de base d'un système flou

Termes linguistiques). Par exemple, dans le cas d'un système flou à sortie unique de n -entrée, la base de règles floues peut consister en les règles suivantes:

$$R^k : \mathbf{Si} \ x_1 \text{ est } A_1^k \text{ et } x_2 \text{ est } A_2^k \text{ et } \dots \text{ et } x_n \text{ est } A_n^k \ \mathbf{Alors} \ z \text{ est } B^k$$

où x_i $i = 1, 2, \dots, n$ sont les entrées de système flou, z est la sortie du système flou, A_i^k et B^k sont des termes linguistiques, et k le nombre des règles floues. Ces règles floues **SI-ALORS** fournissent une forme naturelle dans laquelle les humains représentent leurs connaissances. En reliant chaque terme linguistique aux règles floues avec une fonction d'appartenance, nous spécifions la signification des règles floues dans un sens flou déterminé. Les paramètres de conception d'une base de règles floues sont: k , le nombre de règles floues dans la base de la règle floue; et, (2) l'énoncé spécifique de chaque règle floue.

Le moteur d'inférence flou est une logique de décision qui utilise des règles floues de la base de règles floue pour déterminer les sorties floues d'un système flou correspondant aux entrées du système flou. C'est le moteur d'inférence flou qui simule une procédure de prise de décision humaine basée sur des concepts flous et des énoncés linguistiques. Le paramètre de conception d'un moteur d'inférence flou est: quelle logique floue spécifique est utilisée.

L'interface de défuzzification transforme la grandeur floue issue de l'inférence en une grandeur physique. Cependant, il existe plusieurs méthodes de défuzzification, la méth-

ode la plus utilisée dans la commande est la méthode de défuzzification dite la moyenne des centres.

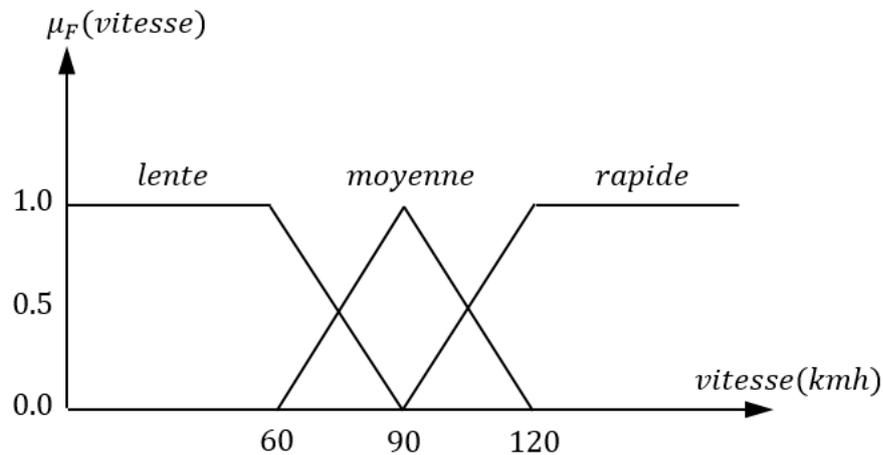


Figure 1.14: Fonction d'appartenance

Les fonctions d'appartenance des ensembles flous influencent fortement la «douceur» de la surface entrée-sortie déterminée par le système flou. En général, plus les fonctions sont pointues, moins la surface entrée-sortie est lisse. Le choix des fonctions d'appartenance est également très subjectif. Les fonctions d'appartenance les plus utilisées sont : fonction triangulaire, fonction trapézoïdale, fonction sigmoïde et fonction Gaussienne.

Le nombre d'ensembles flous définis dans les univers et le nombre de règles floues dans la base de règles floues influencent fortement la complexité d'un système flou, où la complexité comprend la complexité du temps, c'est-à-dire les exigences de calcul et la complexité de l'espace, c'est-à-dire les exigences de stockage. Ces paramètres peuvent être considérés comme des paramètres de structure d'un système flou. En général, plus ces paramètres sont grand; plus le système flou est complexe et plus les performances attendues du système flou sont élevées. Par conséquent, il existe toujours un compromis entre complexité et précision dans le choix de ces paramètres.

1.4.2 Système flou de Takagi-Sugeno

Dans la section précédente, nous avons présenté les systèmes flous à conclusions symboliques appelés systèmes flous standards, dits aussi de Mamdani. Ces systèmes utilisent en effet des conclusions symboliques de même nature que les prémisses et, de ce fait, des connaissances exprimées sous forme analytique décrivant la structure interne du

système ne peuvent pas être prises en compte directement. Sugeno et ces collaborateurs [90], ont proposé une classe de systèmes flous qui permet de représenter ce type d'information. Les systèmes flous de cette classe sont appelés systèmes flous à conclusion fonctionnelle où systèmes flous de Takagi-Sugeno (TS).

Notons par $x = [x_1, \dots, x_n]T$ les entrées du système flou, et par y sa sortie. Pour chaque x_i est associé m_i ensembles flous F_i^j dans X_i , tel que pour $x_i \in X_i$, il existe au moins un degré d'appartenance $\mu_{F_i^j}(x_i) \neq 0$, où $i = 1, 2, \dots, n$ et $j = 1, 2, \dots, m_i$. La base de règles du système flou comporte $N = \prod_{i=1}^n m_i$ règles floues de la forme:

$$R^k : \mathbf{Si} \ x_i \ \text{est} \ \widehat{F}_1^k \ \text{et} \ x_n \ \text{est} \ \widehat{F}_n^k \ \mathbf{Alors} \ y = f_k(x) \ (k = 1, \dots, N)$$

où $\widehat{F}_i^k \in \{F_i^1, \dots, F_i^{m_i}\}$ et $f_k(x)$ est une fonction numérique dans l'espace de sortie. En général, $f_k(x)$ est une fonction polynomiale en fonction des variables d'entrées, mais elle peut être aussi une fonction arbitraire tant qu'elle puisse décrire convenablement le comportement du système étudié. Si $f_k(x)$ est une fonction linéaire, i.e. $f_k(x) = a_0^k + \sum_{i=1}^n a_i^k x_i$ alors on a affaire à un système flou de Takagi-Sugeno d'ordre un. Si par contre $f_k(x)$ est un polynôme d'ordre zéro, i.e. $f_k(x) = a^k$ on a donc un système flou de Takagi-Sugeno d'ordre zéro (TS0).

Étant donné que chaque règle possède une conclusion numérique, la sortie totale du système flou est obtenue par le calcul d'une moyenne pondérée, et de cette manière, le temps consommé par la procédure de défuzzification est évité. En fait, la sortie du système flou est donnée par la relation suivante [88]:

$$y(x) = \frac{\sum_{k=1}^N \mu_k(x) f_k(x)}{\sum_{k=1}^N \mu_k(x)} \quad (1.1)$$

avec $\mu_k(x) = \prod_{i=1}^n \mu_{\widehat{F}_i^k}(x_i)$, et $\widehat{F}_i^k \in \{F_i^1, \dots, F_i^{m_i}\}$ représente le degré de confiance ou d'activation de la règle R^k .

Dans le cas d'un système flou TS0, la sortie est donnée par

$$y(x) = \frac{\sum_{k=1}^N \mu_k(x) a^k}{\sum_{k=1}^N \mu_k(x)} \quad (1.2)$$

Dans ce travail on limite l'utilisation des systèmes flous à ceux qui seront directement exploités dans les approches de commande développées, à savoir les systèmes flous de Takagi-Sugeno à conclusion constante, i.e. les systèmes flous de Takagi-Sugeno d'ordre zéro.

En introduisant la notion de fonctions floues de base [88], la sortie du système flou de TS0 (1.2) peut être écrite sous la forme :

$$y(x) = \varphi^T(x)\Theta \quad (1.3)$$

avec : $\Theta = [a^1, \dots, a^N]^T$ le vecteur des paramètres de la partie conclusion des règles floues. $\varphi(x) = [\varphi_1(x), \dots, \varphi_N(x)]^T$ est le vecteur des fonctions floues de base, dont chacune des composantes est donnée par :

$$\varphi_k(x) = \frac{\mu_k(x)}{\sum_{i=1}^N \mu_i(x)}, \quad k = 1, \dots, N \quad (1.4)$$

Les fonctions d'appartenance utilisées dans ce travail, sont choisies des fonction Gaussiennes définies par:

$$\mu_{F_i^j}(x_i) = \exp\left(-\frac{1}{2}\left(\frac{x_i - c_i^j}{\sigma_i^j}\right)^2\right) \quad (1.5)$$

avec c_i^j la moyenne et σ_i^j la variance.

1.4.3 Capacité d'approximation des systèmes Flous

Les systèmes flous sont construits à partir de règles floues de la forme *Si-Alors*, de ce fait, les informations linguistiques ou mathématiques disponibles, issues d'une expertise peuvent éventuellement être incorporées dans le contrôleur. De plus ils ont la propriété des approximateurs universels de fonctions continues. Ces deux caractéristiques ont poussé l'utilisation des systèmes flous dans la commande adaptative.

Il a été prouvé dans les travaux de *Li-Xin Wang* [89] [91] que les systèmes flous sont des approximateurs universels , c'est-à-dire, pour toute fonction continue $f(x)$ définie sur un ensemble compact Ω , et pour toute constante positive δ , il existe un système flou de la forme (1.3) tel que :

$$\sup_{x \in \Omega} |f(x) - y(x)| \leq \delta \quad (1.6)$$

Notons, cependant, que la propriété d'approximation universelle ne donne pas une méthode de construction du système flou $y(x)$, elle garantit seulement son existence. De plus, pour un degré de précision quelconque, on a besoin d'un nombre important de règles floues.

Dans ce travail la structure du système flou, les entrées du système flou, le nombre de fonctions d'appartenance pour chaque entrée, les paramètres des fonctions d'appartenance et le nombre de règles sont spécifiés par le designer a priori. Par-contre les paramètres de la conclusion seront calculés en ligne par les lois d'adaptation paramétrique.

1.5 Commande adaptative floue

Motivé par le succès pratique du contrôleur flou dans les produits de consommation et le contrôle des processus industriels, il y a eu de nombreux travaux sur les études théoriques rigoureuses des systèmes et des contrôleurs flous. Les chercheurs essaient d'expliquer pourquoi les résultats pratiques sont bons, systématiser l'approche existante, et développer d'autres plus puissants. À la suite de ces efforts, l'image des systèmes flous et de la théorie de la commande floue devient de plus en plus claire [92] [88].

Dans la conception d'un contrôleur flou, certains problèmes sont rencontrés pour des applications pratiques, notamment: (1) la conception des contrôleurs flous est effectuée de manière *ad-hoc* (objectif spécial et immédiat), donc il est souvent difficile de choisir au moins certains des paramètres du contrôleur. Par exemple, il est parfois difficile de savoir comment choisir les fonctions d'appartenances et la base de règles pour atteindre les performances souhaitées. (2) la loi de commande floue synthétisée pour le modèle du système peut fonctionner de manière inadéquate si des variations importantes et imprévisibles des paramètres du système se produisent, ou s'il existe un bruit ou un certain type de perturbation externe ou d'autres effets environnementales. Par conséquent, le modèle du système change lorsque la boucle fermée fonctionne, et les performances souhaitées ne peuvent pas être maintenues. Pour surmonter ces problèmes nous avons besoin d'un moyen pour régler automatiquement le contrôleur flou pour qu'il puisse s'adapter à différents changements du système. L'objectif principal de la commande adaptative est de maintenir une performance constante d'un système en présence de grandes incertitudes et variations paramétriques. Par conséquent, le contrôle flou avancé devrait être adaptatif [93].

Les principaux avantages de la commande adaptative floue sont: (1) une meilleure performance est généralement obtenue car le contrôleur flou adaptatif peut s'adapter au changement du système ou de l'environnement, et (2) moins d'informations sur le système sont nécessaires car la loi d'adaptation peut aider à apprendre la dynamique du système en temps réel [92]. La commande adaptative floue et la commande adaptative conventionnelle ont des similitudes et des différences. Ils sont similaires dans: (1) la

configuration de base et les principes sont plus ou moins les mêmes, et (2) les outils mathématiques utilisés dans l'analyse et la conception sont très semblables. Les principales différences sont les suivantes: (1) le contrôleur flou a une structure non linéaire spéciale qui est universelle pour différents systèmes, alors que la structure d'un contrôleur adaptatif classique change d'un système à un autre, et (2) la connaissance humaine sur la dynamique du système et la stratégie de commande peuvent être incorporées dans le contrôleur adaptatif flou, alors que ces connaissances ne sont pas considérées dans le contrôleur adaptatif conventionnel.

Selon la connaissance humaine utilisée et la structure du contrôleur flou, la commande adaptative floue peut être classée en deux catégories [92]:

- **Commande adaptative floue indirecte:** le contrôleur flou comprend un nombre des systèmes flous construits (initialement) à partir des connaissances du modèle du système.
- **Commande adaptative floue directe:** le contrôleur flou est un seul système flou construit (initialement) à partir des connaissances du contrôleur.

Dans les deux sections suivantes, nous allons développer le schéma de base de la commande adaptative floue indirecte ainsi que le schéma de la commande adaptative floue directe à travers un système non linéaire d'ordre n .

1.5.1 Commande adaptative floue indirecte

Supposons un système non linéaire d'ordre n représenté par l'équation différentielle suivante :

$$\begin{aligned} x^{(n)} &= f(x, \dot{x}, \dots, x^{(n-1)}) + g(x, \dot{x}, \dots, x^{(n-1)})u \\ y &= x \end{aligned} \tag{1.7}$$

avec f et g sont des fonctions inconnues, $u \in \mathbb{R}$ et $y \in \mathbb{R}$ sont l'entrée et la sortie du système respectivement, et $\mathbf{x} = [x_1, x_2, \dots, x_n]^T = [x, \dot{x}, \dots, x^{(n-1)}]^T$ est le vecteur d'état du système supposé mesurable. Pour que (1.7) soit commandable, nous exigeons que $g(\mathbf{x}) \neq 0$. Sans perte de généralité nous supposons que $g(\mathbf{x}) > 0$.

L'objectif de la commande est de concevoir un contrôleur $u(\mathbf{x}|\theta)$ basé sur les systèmes flous et une loi d'adaptation pour ajuster le vecteur des paramètres θ , de telle sorte que la sortie du système y suit la référence y_r , qui et ses dérivés sont connues et bornées.

Puisque nous concevons une commande floue adaptative indirecte, certaines connaissances du système sont disponibles. Plus précisément, nous supposons qu'il existe une collection de règles **SI-ALORS** floues qui décrivent le comportement entrée-sortie de $f(\mathbf{x})$ et $g(\mathbf{x})$; ces règles sont données comme suit:

$$\mathbf{SI} \ x_1 \text{ est } F_1^r \text{ et } \dots \text{ et } x_n \text{ est } F_n^r, \mathbf{ALORS} \ f(\mathbf{x}) \text{ est } C^r \quad (1.8)$$

qui décrit $f(\mathbf{x})$, et

$$\mathbf{SI} \ x_1 \text{ est } G_1^s \text{ et } \dots \text{ et } x_n \text{ est } G_n^s, \mathbf{ALORS} \ g(\mathbf{x}) \text{ est } D^s \quad (1.9)$$

qui décrit $g(\mathbf{x})$, avec F_i^r , C^r , G_i^s et D^s sont des ensembles flous, $r = 1, 2, \dots, L_f$ et $s = 1, 2, \dots, L_g$.

1.5.1.1 Conception du contrôleur flou

Si les fonctions non linéaires $f(\mathbf{x})$ et $g(\mathbf{x})$ sont connues, alors on peut synthétiser la loi de commande u sur la base de la théorie de la commande linéaire (par exemple, le placement des pôles) pour compenser les non linéarités. Définissons $e = y_r - y$, $\mathbf{e} = [e, \dot{e}, \dots, e^{(n-1)}]^T$ et $\mathbf{k} = [k_n, \dots, k_1]^T$ tel que toutes les racines du polynôme $s^n + k_1 s^{n-1} + \dots + k_n$ soient dans la partie gauche du plan complexe, et la loi de commande est choisi comme suit:

$$u^* = \frac{1}{g(\mathbf{x})} [-f(\mathbf{x}) + y_r^{(n)} + \mathbf{k}^T \mathbf{e}] \quad (1.10)$$

Substituant (1.10) dans (1.7), on obtient la dynamique de l'erreur en boucle fermée

$$e^{(n)} + k_1 e^{(n-1)} + \dots + k_n e = 0 \quad (1.11)$$

avec un choix judicieux de \mathbf{k} , on a $e(t) \rightarrow 0$ lorsque $t \rightarrow \infty$, et la sortie du système y converge asymptotiquement à la référence y_r .

Puisque les fonctions $f(\mathbf{x})$ et $g(\mathbf{x})$ sont inconnues, la loi de commande idéale (1.10) ne peut pas être implémentée. Néanmoins, nous avons les règles floues **SI-ALORS** (1.8)-(1.9) qui décrivent le comportement entrée-sortie de $f(\mathbf{x})$ et $g(\mathbf{x})$. Cependant, une idée raisonnable est de remplacer les fonctions $f(\mathbf{x})$ et $g(\mathbf{x})$ dans (1.10) par les systèmes flous $\hat{f}(\mathbf{x})$ et $\hat{g}(\mathbf{x})$, qui sont construits à partir des règles floues. Pour améliorer la précision de $\hat{f}(\mathbf{x})$ et $\hat{g}(\mathbf{x})$, une idée consiste à laisser certains paramètres dans $\hat{f}(\mathbf{x})$ et $\hat{g}(\mathbf{x})$

libres de changer pendant le fonctionnement en ligne afin que la précision de l'approximation s'améliore avec le temps. Soit $\theta_f \in \mathbb{R}^{M_f}$ et $\theta_g \in \mathbb{R}^{M_g}$ les paramètres libres dans $\hat{f}(\mathbf{x})$ et $\hat{g}(\mathbf{x})$, respectivement, donc nous notons $\hat{f}(\mathbf{x}) = \hat{f}(\mathbf{x}|\theta_f)$ et $\hat{g}(\mathbf{x}) = \hat{g}(\mathbf{x}|\theta_g)$. En remplaçant les fonctions $f(\mathbf{x})$ et $g(\mathbf{x})$ dans (1.9) par $\hat{f}(\mathbf{x}|\theta_f)$ et $\hat{g}(\mathbf{x}|\theta_g)$, respectivement, on obtient le contrôleur flou

$$u = u_I = \frac{1}{\hat{g}(\mathbf{x}|\theta_g)} [-\hat{f}(\mathbf{x}|\theta_f) + y_r^{(n)} + \mathbf{k}^T \mathbf{e}] \quad (1.12)$$

Pour implémenter la loi de commande (1.12) on doit spécifier les formules en détails de $\hat{f}(\mathbf{x}|\theta_f)$ et $\hat{g}(\mathbf{x}|\theta_g)$. Ces dernières sont construites en deux étapes:

- **Étape 1:** Pour la variable x_i ($i = 1, 2, \dots, n$) définir p_i ensembles flous $A_i^{l_i}$ ($l_i = 1, 2, \dots, p_i$) et q_i ensembles flous $B_i^{l_i}$ ($l_i = 1, 2, \dots, q_i$).
- **Étape 2:** Construire le système flou $\hat{f}(\mathbf{x}|\theta_f)$ à partir de $\prod_{i=1}^n p_i$ règles :

$$\mathbf{SI} \ x_1 \text{ est } A_1^{l_1} \text{ et } \dots \text{ et } x_n \text{ est } A_n^{l_n}, \mathbf{ALORS} \ \hat{f} \text{ est } E^{l_1, \dots, l_n} \quad (1.13)$$

avec $l_i = 1, 2, \dots, p_i$ $i = 1, 2, \dots, n$. De même, construire le système flou $\hat{g}(\mathbf{x}|\theta_g)$ à partir de $\prod_{i=1}^n q_i$ règles :

$$\mathbf{SI} \ x_1 \text{ est } B_1^{l_1} \text{ et } \dots \text{ et } x_n \text{ est } B_n^{l_n}, \mathbf{ALORS} \ \hat{g} \text{ est } H^{l_1, \dots, l_n} \quad (1.14)$$

avec $l_i = 1, 2, \dots, q_i$ $i = 1, 2, \dots, n$. En utilisant le moteur d'inférence de type produit, l'interface de fuzzification de type singleton et l'interface de défuzzification de type moyenne des centres, nous obtenons

$$\hat{f}(\mathbf{x}|\theta_f) = \frac{\sum_{l_1=1}^{p_1} \dots \sum_{l_n=1}^{p_n} \bar{y}_f^{l_1, \dots, l_n} \left(\prod_{i=1}^n \mu_{A_i^{l_i}}(x_i) \right)}{\sum_{l_1=1}^{p_1} \dots \sum_{l_n=1}^{p_n} \left(\prod_{i=1}^n \mu_{A_i^{l_i}}(x_i) \right)} \quad (1.15)$$

$$\hat{g}(\mathbf{x}|\theta_g) = \frac{\sum_{l_1=1}^{q_1} \dots \sum_{l_n=1}^{q_n} \bar{y}_g^{l_1, \dots, l_n} \left(\prod_{i=1}^n \mu_{B_i^{l_i}}(x_i) \right)}{\sum_{l_1=1}^{q_1} \dots \sum_{l_n=1}^{q_n} \left(\prod_{i=1}^n \mu_{B_i^{l_i}}(x_i) \right)} \quad (1.16)$$

Soit $\bar{y}_f^{l_1, \dots, l_n}$ et $\bar{y}_g^{l_1, \dots, l_n}$ les paramètres libres qui sont collectées dans $\theta_f \in \mathbb{R}^{\prod_{i=1}^n p_i}$ et $\theta_g \in \mathbb{R}^{\prod_{i=1}^n q_i}$, respectivement, donc nous pouvons réécrire (1.15) et (1.16) comme

$$\hat{f}(\mathbf{x}|\theta_f) = \theta_f^T \varphi_f(\mathbf{x}) \quad (1.17)$$

$$\hat{g}(\mathbf{x}|\theta_g) = \theta_g^T \varphi_g(\mathbf{x}) \quad (1.18)$$

avec $\varphi_f(\mathbf{x})$ est un vecteur de dimension $\prod_{i=1}^n p_i$.

$$\varphi_f(\mathbf{x}) = \frac{\prod_{i=1}^n \mu_{A_i^{l_i}}(x_i)}{\sum_{l_1=1}^{p_1} \cdots \sum_{l_n=1}^{p_n} \left(\prod_{i=1}^n \mu_{A_i^{l_i}}(x_i) \right)} \quad (1.19)$$

et $\varphi_g(\mathbf{x})$ est un vecteur de dimension $\prod_{i=1}^n q_i$.

$$\varphi_g(\mathbf{x}) = \frac{\prod_{i=1}^n \mu_{B_i^{l_i}}(x_i)}{\sum_{l_1=1}^{q_1} \cdots \sum_{l_n=1}^{q_n} \left(\prod_{i=1}^n \mu_{B_i^{l_i}}(x_i) \right)} \quad (1.20)$$

Puisque les paramètres θ_f et θ_g vont changer pendant le fonctionnement en ligne, notre tâche suivante consiste à concevoir une loi d'adaptation pour ces paramètres de telle sorte que l'erreur de suivi e soit minimisée.

1.5.1.2 Conception des lois d'adaptations

En substituant (1.12) dans (1.7) et après quelques manipulations, nous obtenons la dynamique de l'erreur en boucle fermée

$$e^{(n)} = -\mathbf{k}^T \mathbf{e} + [\hat{f}(\mathbf{x}|\theta_f) - f(\mathbf{x})] + [\hat{g}(\mathbf{x}|\theta_g) - g(\mathbf{x})] u_I \quad (1.21)$$

soit

$$\Lambda = \begin{bmatrix} 0 & 1 & 0 & 0 & \dots & 0 & 0 \\ 0 & 0 & 1 & 0 & \dots & 0 & 0 \\ \dots & \dots & \dots & \dots & \dots & \dots & \dots \\ 0 & 0 & 0 & 0 & \dots & 0 & 1 \\ -k_n & -k_{n-1} & \dots & \dots & \dots & \dots & -k_1 \end{bmatrix}, \quad \mathbf{b} = \begin{bmatrix} 0 \\ \dots \\ 0 \\ 1 \end{bmatrix} \quad (1.22)$$

alors l'équation dynamique (1.21) peut être réécrite dans la forme vectorielle

$$\dot{\mathbf{e}} = \Lambda \mathbf{e} + \mathbf{b} \left([\hat{f}(\mathbf{x}|\theta_f) - f(\mathbf{x})] + [\hat{g}(\mathbf{x}|\theta_g) - g(\mathbf{x})] u_I \right) \quad (1.23)$$

Définissons les paramètres optimaux comme suit:

$$\theta_f^* = \arg \min_{\theta_f \in \mathbb{R}^{\prod_{i=1}^n p_i}} \left[\sup_{\mathbf{x} \in \mathbb{R}^n} |\hat{f}(\mathbf{x}|\theta_f) - f(\mathbf{x})| \right] \quad (1.24)$$

$$\theta_g^* = \arg \min_{\theta_g \in \mathbb{R}^{\prod_{i=1}^n q_i}} \left[\sup_{\mathbf{x} \in \mathbb{R}^n} |\hat{g}(\mathbf{x}|\theta_g) - g(\mathbf{x})| \right] \quad (1.25)$$

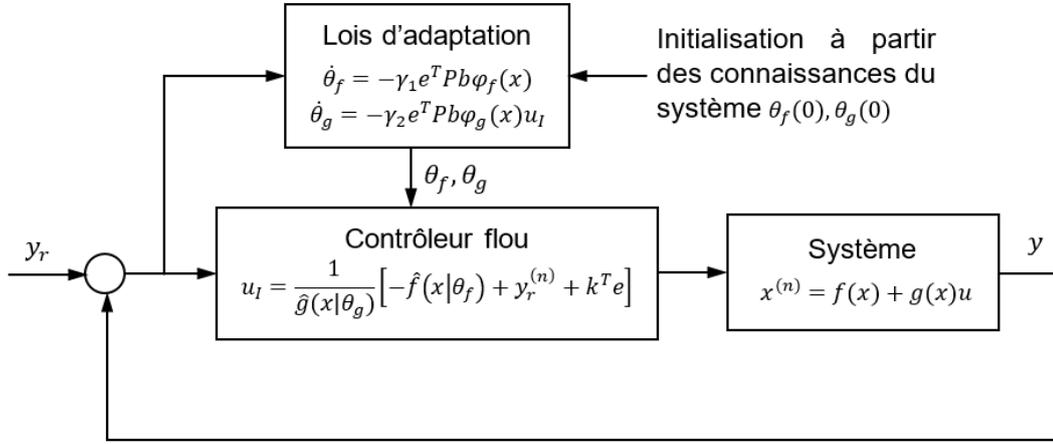


Figure 1.15: Commande adaptative floue indirecte

Ainsi $\hat{f}(\mathbf{x}|\theta_f^*)$ et $\hat{g}(\mathbf{x}|\theta_g^*)$ sont la meilleure approximation (min-max) de $f(\mathbf{x})$ et $g(\mathbf{x})$, respectivement, parmi tous les systèmes flous de la forme (1.17) et (1.18). Définissons aussi l'erreur d'approximation minimale

$$\epsilon = [\hat{f}(\mathbf{x}|\theta_f) - f(\mathbf{x})] + [\hat{g}(\mathbf{x}|\theta_g) - g(\mathbf{x})]u_I \quad (1.26)$$

En utilisant ϵ , nous pouvons réécrire (1.23) comme suit

$$\dot{\mathbf{e}} = \Lambda \mathbf{e} + \mathbf{b} \left([\hat{f}(\mathbf{x}|\theta_f) - \hat{f}(\mathbf{x}|\theta_f^*)] + [\hat{g}(\mathbf{x}|\theta_g) - \hat{g}(\mathbf{x}|\theta_g^*)]u_I + \epsilon \right) \quad (1.27)$$

En substituant (1.17) et (1.18) dans (1.27), on obtient la dynamique en boucle fermée suivante qui spécifie explicitement la relation entre l'erreur de suivi \mathbf{e} et les paramètres de la commande floue θ_f et θ_g :

$$\dot{\mathbf{e}} = \Lambda \mathbf{e} + \mathbf{b} [(\theta_f - \theta_f^*)^T \varphi_f(\mathbf{x}) + (\theta_g - \theta_g^*)^T \varphi_g(\mathbf{x})u_I + \epsilon] \quad (1.28)$$

La tâche d'une loi d'adaptation est de déterminer un mécanisme d'ajustement pour θ_f et θ_g tel que l'erreur de suivi \mathbf{e} et les erreurs des paramètres $\theta_f - \theta_f^*$ et $\theta_g - \theta_g^*$ sont minimisés.

Pour compléter cette tâche, considérons la fonction de Lyapunov suivante

$$V = \frac{1}{2} \mathbf{e}^T P \mathbf{e} + \frac{1}{2\gamma_1} (\theta_f - \theta_f^*)^T (\theta_f - \theta_f^*) + \frac{1}{2\gamma_2} (\theta_g - \theta_g^*)^T (\theta_g - \theta_g^*) \quad (1.29)$$

avec γ_1 et γ_2 sont des constants positifs, et P est une matrice définie positive satisfait l'équation de Lyapunov

$$\Lambda^T P + P \Lambda = -Q \quad (1.30)$$

avec Q est une matrice arbitraire définie positive, et Λ est donnée par (1.22). la dérivée temporelle de V est donnée par

$$\begin{aligned} \dot{V} = & -\frac{1}{2} \mathbf{e}^T Q \mathbf{e} + \mathbf{e}^T P \mathbf{b} \epsilon + \frac{1}{\gamma_1} (\theta_f - \theta_f^*)^T [\dot{\theta}_f + \gamma_1 \mathbf{e}^T P \mathbf{b} \varphi_f(\mathbf{x})] \\ & + \frac{1}{\gamma_2} (\theta_g - \theta_g^*)^T [\dot{\theta}_g + \gamma_2 \mathbf{e}^T P \mathbf{b} \varphi_g(\mathbf{x}) u_I] \end{aligned} \quad (1.31)$$

Pour minimiser l'erreur de poursuite \mathbf{e} et les erreurs $\theta_f - \theta_f^*$ et $\theta_g - \theta_g^*$ ou de manière équivalente, pour minimiser V , nous devrions choisir la loi d'adaptation tel que \dot{V} soit négative. Puisque le terme $-\frac{1}{2} \mathbf{e}^T Q \mathbf{e}$ est négatif et que nous pouvons choisir les systèmes flous de telle sorte que l'erreur d'approximation minimale ϵ soit faible, une bonne stratégie consiste à choisir la loi d'adaptation tel que les deux derniers termes de (1.31) soient nuls, c'est-à-dire les lois d'adaptations sont données par:

$$\dot{\theta}_f = -\gamma_1 \mathbf{e}^T P \mathbf{b} \varphi_f(\mathbf{x}) \quad (1.32)$$

$$\dot{\theta}_g = -\gamma_2 \mathbf{e}^T P \mathbf{b} \varphi_g(\mathbf{x}) u_I \quad (1.33)$$

Cette approche de la conception de la loi d'adaptation est appelée *l'approche de synthèse de Lyapunov*, car le but est de minimiser la fonction de Lyapunov V . En résumé, la loi de commande floue adaptative indirecte est illustrée dans la Figure 1.15.

1.5.1.3 Exemple d'application

Nous appliquons maintenant le contrôleur flou adaptatif indirect au système de pendule inversé dont la dynamique est caractérisée par l'équation (1.34). Nous choisissons $k_1 = 2$ et $k_2 = 1$ (de telle sorte que $s^2 + k_1 s + k_2$ est stable), $Q = \text{diag}(10, 10)$, et résolvons l'équation de Lyapunov (1.30) pour obtenir la matrice P .

$$\begin{aligned} \dot{x}_1 &= x_2 \\ \dot{x}_2 &= \frac{g \sin x_1 - \frac{m l x_2^2 \cos x_1 \sin x_1}{m_c + m}}{l \left(\frac{4}{3} - \frac{m \cos^2 x_1}{m_c + m} \right)} + \frac{\frac{\cos x_1}{m_c + m}}{l \left(\frac{4}{3} - \frac{m \cos^2 x_1}{m_c + m} \right)} u \end{aligned} \quad (1.34)$$

où $g = 9.8m/s^2$ est l'accélération due à la gravité, $m_c = 1kg$ est la masse du chariot, $m = 0.1kg$ est la masse de la tige, $l = 0.5m$ est la demi-longueur de la tige, et u est la

force appliquée (entrée de commande). Nous supposons qu'il n'y a pas de règles linguistiques et nous choisissons $p_1 = p_2 = q_1 = q_2 = 5$, $A_1^l = A_2^l = B_1^l = B_2^l$ avec $\mu_{A_1^1}(x_1) = \exp[-(\frac{x_1+\pi/6}{\pi/24})^2]$, $\mu_{A_1^2}(x_1) = \exp[-(\frac{x_1+\pi/12}{\pi/24})^2]$, $\mu_{A_1^3}(x_1) = \exp[-(\frac{x_1}{\pi/24})^2]$, $\mu_{A_1^4}(x_1) = \exp[-(\frac{x_1-\pi/12}{\pi/24})^2]$, et $\mu_{A_1^5}(x_1) = \exp[-(\frac{x_1-\pi/6}{\pi/24})^2]$, qui couvrent l'intervalle $[-\pi/6, \pi/6]$. Puisque le rang de $f(x_1, x_2)$ est beaucoup plus grande que celui de $g(x_1, x_2)$, nous choisissons $\gamma_1 = 50$ et $\gamma_2 = 1$. La figure 1.16 montre l'évolution temporelle de la sortie $x_1(t)$ et la sortie de référence $y_r(t)$ pour les conditions initiales $\mathbf{x}(0) = [\frac{\pi}{60}, 0]^T$.

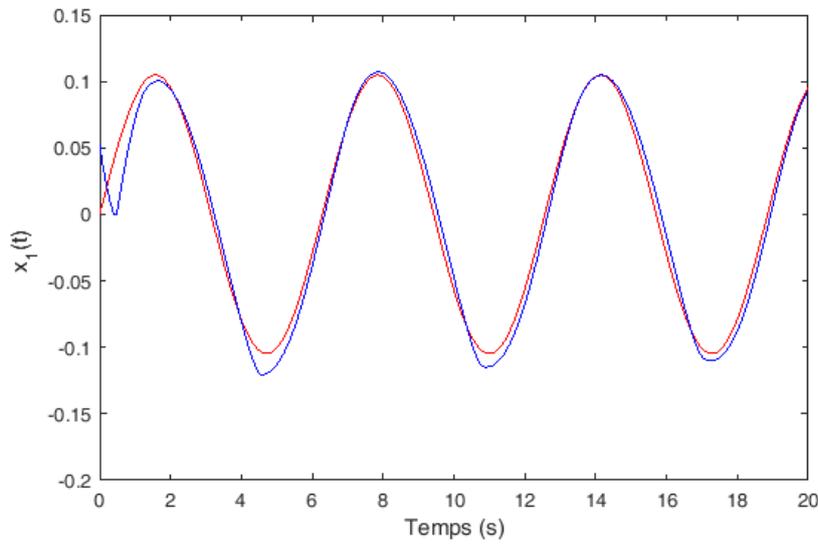


Figure 1.16: La sortie $x_1(t)$ et la sortie de référence $y_r(t) = \frac{\pi}{30} \sin(t)$ pour les conditions initiales $\mathbf{x}(0) = [\frac{\pi}{60}, 0]^T$,

1.5.2 Commande adaptative floue directe

Considérons le système non linéaire d'ordre n représenté par l'équation différentielle suivante :

$$\begin{aligned} x^{(n)} &= f(x, \dot{x}, \dots, x^{(n-1)}) + bu \\ y &= x \end{aligned} \quad (1.35)$$

avec $f(x, \dot{x}, \dots, x^{(n-1)})$ une fonction inconnue et b une constante positive. L'objectif de la commande reste le même que dans la commande floue adaptative indirecte; c'est-à-dire, concevoir un contrôleur $u(\mathbf{x}|\theta)$ basé sur les systèmes flous et une loi d'adaptation pour ajuster le vecteur des paramètres θ , de telle sorte que la sortie du système y suit la référence y_r aussi proche que possible. La principale différence réside dans l'hypothèse

concernant les connaissances humaines disponibles. Plus précisément, au lieu de connaître les connaissances du système (1.8) et (1.9), nous avons ici quelques connaissances sur la commande; c'est-à-dire les règles floues **SI-ALORS** qui décrivent les actions de commande.

$$\mathbf{SI} \ x_1 \text{ est } P_1^r \text{ et } \dots \text{ et } x_n \text{ est } P_n^r, \mathbf{ALORS} \ u \text{ est } Q^r \quad (1.36)$$

avec P_i^r et Q^r sont des ensembles flous dans \mathbb{R} , et $r = 1, 2, \dots, L_u$. Le contrôleur flou devrait être conçu de telle sorte que les règles (1.36) puissent être incorporées.

1.5.2.1 Conception du contrôleur flou

Pour incorporer les règles (1.36), un choix naturel consiste à utiliser un seul système flou comme contrôleur, c'est-à-dire que le contrôleur flou est dans ce cas

$$u = u_D(\mathbf{x}|\theta) \quad (1.37)$$

avec u_D est un système flou et θ est le vecteur des paramètres ajustables. Plus précisément, le système flou $u_D(\mathbf{x}|\theta)$ est construit à partir des deux étapes suivantes:

- **Étape 1:** Pour chaque variable x_i ($i = 1, 2, \dots, n$), définir m_i ensembles flou $A_i^{l_i}$ ($l_i = 1, 2, \dots, m_i$), qui comprennent les P_i^r dans (1.36).
- **Étape 2:** Construire le système flou $u_D(\mathbf{x}|\theta)$ à partir de $\prod_{i=1}^n m_i$ règles :

$$\mathbf{SI} \ x_1 \text{ est } A_1^{l_1} \text{ et } \dots \text{ et } x_n \text{ est } A_n^{l_n}, \mathbf{ALORS} \ u_D \text{ est } S^{l_1, \dots, l_n} \quad (1.38)$$

avec $l_i = 1, 2, \dots, m_i$, $i = 1, 2, \dots, n$, et S^{l_1, \dots, l_n} est égal à Q^r dans (1.36) si la partie **SI** de (1.36) correspond avec la partie **SI** de (1.38) et est égal à un certain ensembles flous arbitraires. En utilisant le moteur d'inférence de type produit, l'interface de fuzzification de type singleton et l'interface de défuzzification de type moyenne des centres, nous obtenons

$$u_D(\mathbf{x}|\theta) = \frac{\sum_{l_1=1}^{m_1} \dots \sum_{l_n=1}^{m_n} \bar{y}_u^{l_1, \dots, l_n} \left(\prod_{i=1}^n \mu_{A_i^{l_i}}(x_i) \right)}{\sum_{l_1=1}^{m_1} \dots \sum_{l_n=1}^{m_n} \left(\prod_{i=1}^n \mu_{A_i^{l_i}}(x_i) \right)} \quad (1.39)$$

Choisissons $\bar{y}_u^{l_1, \dots, l_n}$ comme paramètres ajustables qui sont collectés dans $\theta_f \in \mathbb{R}^{\prod_{i=1}^n m_i}$, le contrôleur flou devient

$$u_D(\mathbf{x}|\theta) = \theta^T \varphi(\mathbf{x}) \quad (1.40)$$

avec $\varphi(\mathbf{x})$ est la même que dans (1.19) sauf que p_i est maintenant remplacé par m_i .

1.5.2.2 Conception de loi d'adaptation

Soit u^* la même loi de commande idéale (1.10) que dans la section 1.5.1.1, avec $g(\mathbf{x}) = b$. En substituant (1.37) en (1.35) et par réarrangement, on obtient

$$e^{(n)} = -\mathbf{k}^T \mathbf{e} + b[u^* - u_D(\mathbf{x}|\theta)] \quad (1.41)$$

Soit Λ être tel que défini dans (1.22) et $\mathbf{b} = [0, \dots, 0, b]^T$, la dynamique de l'erreur en boucle fermée peut être écrite dans la forme vectorielle suivante

$$\dot{\mathbf{e}} = \Lambda \mathbf{e} + \mathbf{b}[u^* - u_D(\mathbf{x}|\theta)] \quad (1.42)$$

Définissons les paramètres optimaux

$$\theta^* = \arg \min_{\theta \in \mathbb{R}^{\prod_{i=1}^n m_i}} \left[\sup_{\mathbf{x} \in \mathbb{R}^n} |u_D(\mathbf{x}|\theta) - u^*| \right] \quad (1.43)$$

et l'erreur d'approximation minimale

$$\epsilon = u_D(\mathbf{x}|\theta^*) - u^* \quad (1.44)$$

En utilisant (1.44) et (1.40), nous pouvons réécrire l'équation d'erreur (1.42) comme suit

$$\dot{\mathbf{e}} = \Lambda \mathbf{e} + \mathbf{b}(\theta^* - \theta)^T \varphi(\mathbf{x}) - \mathbf{b}\epsilon \quad (1.45)$$

Considérons la fonction Lyapunov suivante

$$V = \frac{1}{2} \mathbf{e}^T P \mathbf{e} + \frac{1}{2\gamma} (\theta^* - \theta)^T (\theta^* - \theta) \quad (1.46)$$

avec P une matrice définie positive satisfait l'équation de Lyapunov (1.30), et γ est un constant positif (rappeler que $b > 0$ par hypothèse, donc V est positif). En utilisant (1.45) et (1.30), nous avons

$$\dot{V} = -\frac{1}{2} \mathbf{e}^T Q \mathbf{e} + \mathbf{e}^T P \mathbf{b} \left[(\theta^* - \theta)^T \varphi(\mathbf{x}) - \epsilon \right] - \frac{b}{\gamma} (\theta^* - \theta)^T \dot{\theta} \quad (1.47)$$

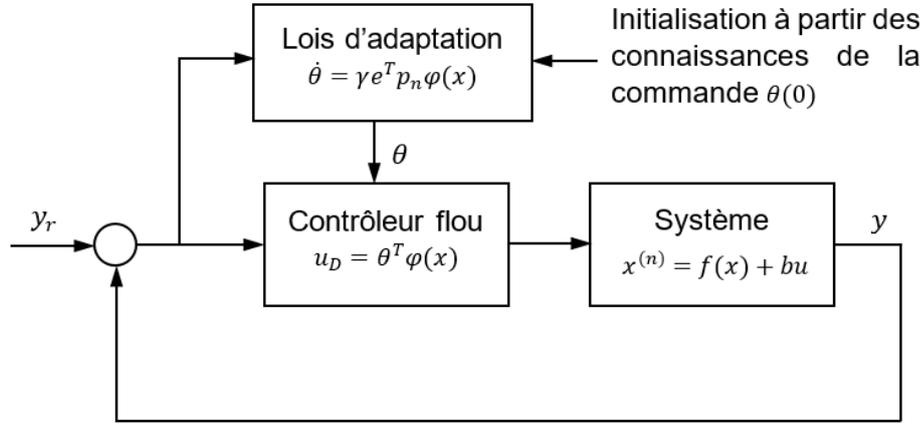


Figure 1.17: Commande adaptative floue directe

Soit p_n la dernière colonne de P , ensuite à partir de $\mathbf{b} = (0, \dots, 0, b)^T$ nous avons $\mathbf{e}^T P \mathbf{b} = \mathbf{e}^T p_n b$. Donc (1.47) peut être réécrit comme

$$\dot{V} = -\frac{1}{2} \mathbf{e}^T Q \mathbf{e} + \frac{b}{\gamma} (\theta^* - \theta)^T \left[\gamma \mathbf{e}^T p_n \varphi(\mathbf{x}) - \dot{\theta} \right] - \mathbf{e}^T p_n b \epsilon \quad (1.48)$$

Si nous choisissons la loi d'adaptation

$$\dot{\theta} = \gamma \mathbf{e}^T p_n \varphi(\mathbf{x}) \quad (1.49)$$

alors

$$\dot{V} = -\frac{1}{2} \mathbf{e}^T Q \mathbf{e} - \mathbf{e}^T p_n b \epsilon \quad (1.50)$$

Puisque $Q > 0$, on peut espérer qu'en concevant le système flou $u_D(\mathbf{x}|\theta)$ avec un nombre de règles suffisamment grand, l'erreur ϵ serait assez petit pour que $|\mathbf{e}^T p_n b \epsilon| < \frac{1}{2} \mathbf{e}^T Q \mathbf{e}$, ce qui aboutit à $\dot{V} < 0$.

1.5.2.3 Exemple d'application

Considérons le système non linéaire de premier ordre 1.51. Notre objectif est d'utiliser la commande floue adaptative directe pour réguler l'état $x(t)$ à zéro; donc, on a $y_r = 0$.

$$\dot{x}(t) = \frac{1 - \exp^{-x(t)}}{1 + \exp^{-x(t)}} + u(t) \quad (1.51)$$

Il est clair que le système (1.51) est instable si le contrôle est égal à zéro. En effet, si $u(t) = 0$, alors $\dot{x} = \frac{1-\exp^{-x}}{1+\exp^{-x}} > 0$ pour $x > 0$, et $\dot{x} = \frac{1-\exp^{-x}}{1+\exp^{-x}} < 0$ pour $x < 0$. Nous choisissons $\gamma = 3$ et définissons six ensembles flous $N3$, $N2$, $N1$, $P1$, $P2$ et $P3$ dans l'intervalle $[-3, 3]$ avec les fonctions d'appartenance $\mu_{N3}(x) = 1/(1 + \exp(5(x + 2)))$, $\mu_{N2}(x) = \exp(-(x + 1.5)^2)$, $\mu_{N1}(x) = \exp(-(x + 0.5)^2)$, $\mu_{P1}(x) = \exp(-(x - 0.5)^2)$, $\mu_{P2}(x) = \exp(-(x - 1.5)^2)$ et $\mu_{P3}(x) = 1/(1 + \exp(-5(x - 2)))$, qui sont montrées dans la figure 1.18.

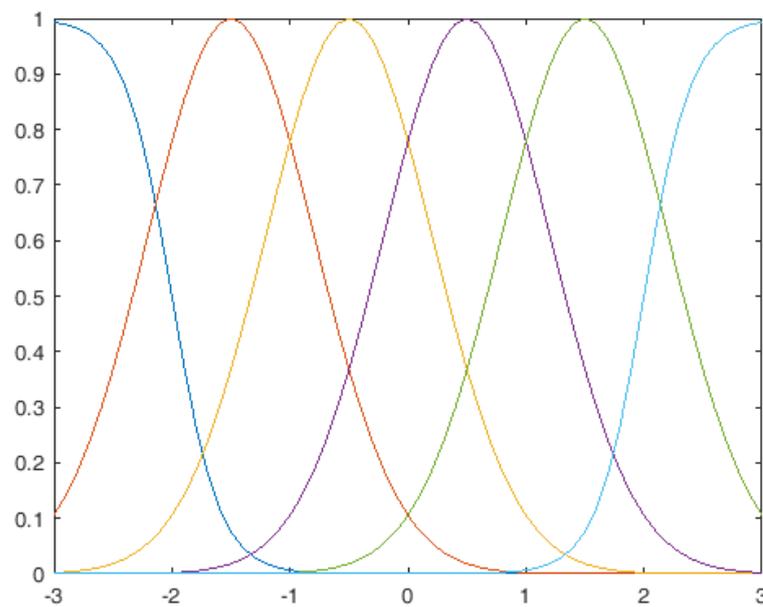


Figure 1.18: Fonctions d'appartenance définies dans l'espace d'état

La figure 1.19 montre la sortie $x(t)$ pour la condition initiale $x(0) = 1$. Nous voyons que le contrôleur flou adaptatif direct pourrait réguler le système à l'origine.

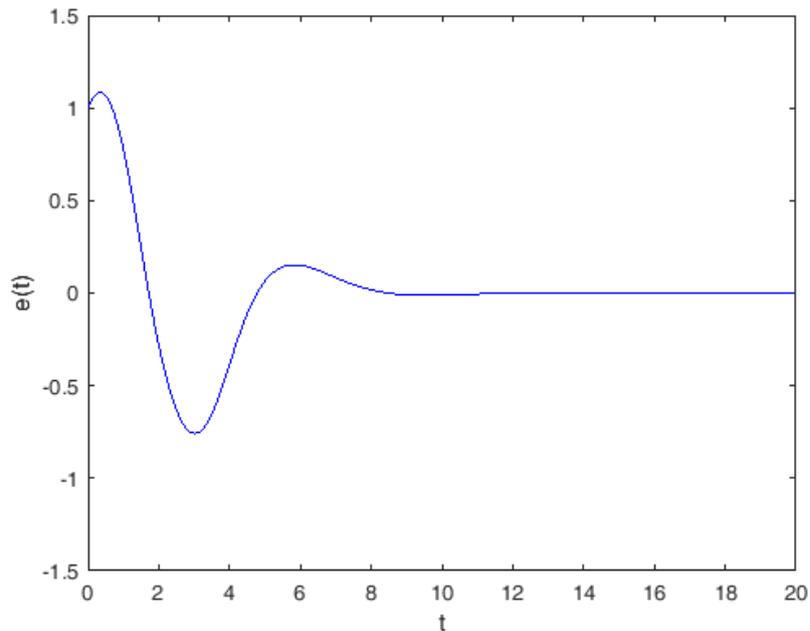


Figure 1.19: Sortie du système en boucle fermée $x(t)$

1.6 Conclusion

L'objectif principal de ce chapitre était de faire un état de l'art sur les drones miniatures à voilure tournante précisément le Quadrotor, d'introduire les système flous et la commande adaptative floue et de faire une analyse bibliographique sur la commande de Quadrotor. Les systèmes flous sont des approximateurs universels, ils peuvent approximer des fonctions non linéaires continues avec un degré de précision quelconque. De plus, une expertise humaine peut être introduite dans le contrôleur flou sous forme mathématique où linguistique.

Le Quadrotor est un système non linéaire complexe, sous actionné et fortement couplé. Obtenir un modèle qui représente précisément le comportement dynamique et les effets qui agissent sur ce type de robot volant, s'annonce souvent comme une tâche difficile, voire impossible. Ce qui dégrade considérablement les performances des approches de commande basées sur le modèle. Pour surmonter ce problème, nous sommes orientés vers les algorithmes de commande adaptative floue.

Dans l'état de l'art présenté sur les drones miniatures à voilure tournante, nous avons focalisé notre étude sur le Quadrotor et ces applications civiles potentielles. Aussi dans l'étude bibliographique sur la commande de Quadrotor, nous avons essayé de faire le tour

d'horizon des techniques de commande proposées pour résoudre certains problèmes liés à ce système.

Le modèle cinématique et le modèle dynamique du Quadrotor seront abordés dans le chapitre suivant, ainsi que les forces et les moments qui agissent sur ce dernier.

Chapitre 2

Modélisation dynamique du Quadrotor

2.1 Introduction

Un Quadrotor consiste en quatre moteurs (BLDC) sur lesquels des hélices sont fixées. Ces moteurs sont arrangés aux extrémités d'un châssis en forme de X , où chaque bras du châssis forme un angle de 90° avec l'autre. La vitesse de rotation des moteurs (i.e. la force de portance associée à l'hélice attachée à ce moteur) peut être modifiée individuellement, modifiant ainsi l'attitude et l'altitude du véhicule permettant au Quadrotor de naviguer dans l'espace.

Plusieurs modèles ont été proposés dans la littérature pour le Quadrotor. Dans [94] [95] [96] les auteurs considéraient le véhicule comme un corps rigide, dont la dynamique est décrite en utilisant la formulation d'Euler-Lagrange avec l'approximation de petits angles, afin d'en déduire un modèle simple pour des fins de commande. Dans [76] [97] le modèle de Quadrotor est encore déduit en utilisant la formulation d'Euler-Lagrange, mais les approximations pour la commande sont moins conservatives. Dans [98] [99], le modèle du système est obtenu en utilisant l'approche de Newton-Euler et les effets aérodynamiques les plus importants sont inclus dans le modèle. Dans [41] [100] un modèle global du Quadrotor est obtenu, prenant en compte la dynamique du corps rigide, l'effet gyroscopique sur les hélices et plusieurs forces aérodynamiques. Enfin, les auteurs de [11] et [7] mettent l'accent sur la modélisation des effets aérodynamiques et le battement des hélices pour le Quadrotor.

Les approches citées ci-dessus considèrent le Quadrotor comme un corps rigide dans l'espace soumis à des forces et des moments. Les forces qui agissent sur le Quadrotor sont nombreux et souvent difficile de les exprimer sous forme analytique à savoir les forces de portance et les effets aérodynamiques complexes. Tout d'abord un modèle théorique global est présenté, afin de donner un aperçu sur les effets qui agissent sur le Quadrotor. Ensuite ce modèle est simplifié afin d'obtenir un système d'équation plus adapté à la simulation.

2.2 Principe de vol du Quadrotor

Un Quadrotor possède, comme son nom l'indique, quatre moteurs, chaque moteur est constitué d'une hélice accouplée à un rotor. Les moteurs avant et arrière (M_1 et M_3) tournent dans le sens des aiguilles d'une montre avec les vitesses angulaires ω_1 et ω_3 , tandis que les deux autres moteurs (M_2 et M_4) tournent dans le sens contraire avec les vitesses angulaires ω_2 et ω_4 (figure 2.2). La figure 2.1 montre les mouvements du Quadrotor pour une configuration plus + (c-à-d le mouvement de roulis et tangage est

réalisé par la différence de poussée entre deux moteurs).

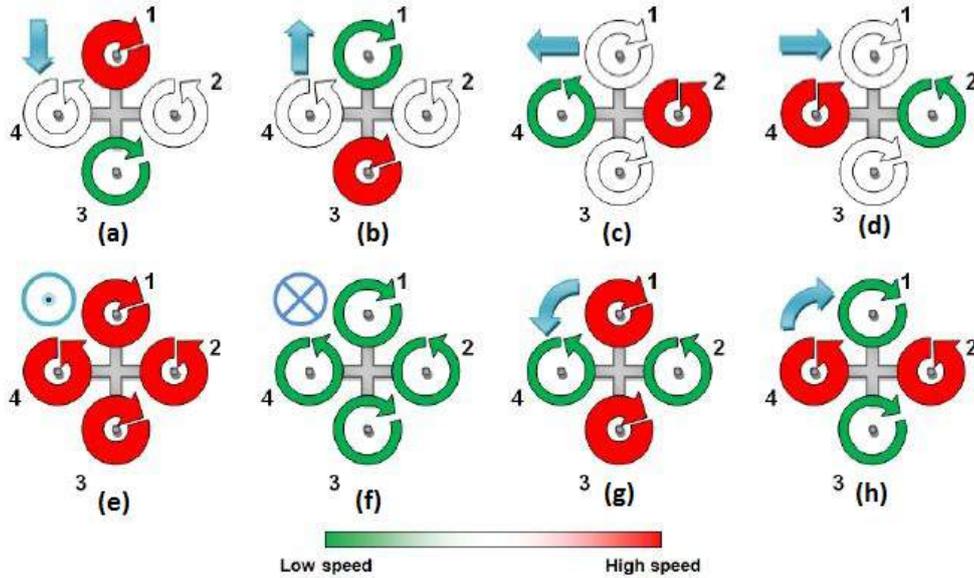


Figure 2.1: Mouvements du Quadrotor

La différence de portance entre les deux rotors avants et arrières produit un couple de tangage qui contrôle le mouvement de translation (avant/arrière) (figure 2.1a et 2.1b). Le déplacement (gauche/droite), quant à lui, est assuré par la différence de portance entre les rotors situés sur la gauche et sur la droite (figure 2.1c et 2.1d). Le mouvement vertical de montée et de descente est assuré par les quatre rotors à la fois (figures 2.1e et 2.1f). Enfin pour contrôler le lacet du système, c'est la somme des anti-couples produites par les quatre rotors qui définit le sens de rotation (figure 2.1g et 2.1h).

2.3 Modélisation dynamique

Le Quadrotor est modélisé en utilisant la notation présentée sur la figure 2.1. Deux repères sont utilisés pour étudier les mouvements du système, un repère inertiel lié à la terre $E^a(O^a, x_a, y_a, z_a)$ et un repère lié au corps rigide $E^b(O^b, x_b, y_b, z_b)$, où O^b est fixé au centre de masse du Quadrotor. La position absolue du Quadrotor est représenté par $\xi = [x \ y \ z]^T$ qui décrit la position du centre de masse de E^b par rapport à E^a , et son attitude par le vecteur $\eta = [\phi \ \theta \ \psi]^T$ qui représentent l'orientation du repère lié au corps rigide par rapport au repère inertiel. Cette notation des angles roulis tangage et lacet, généralement appelée Angles d'Euler, est basée sur l'hypothèse que le repère inertiel $E^a(O^a, e_{a1}, e_{a2}, e_{a3})$ peut atteindre la même orientation du repère $E^b(O^b, e_{b1}, e_{b2}, e_{b3})$

en effectuant d'abord une rotation d'un angle ψ autour de l'axe z (lacet), puis une rotation d'un angle θ autour de l'axe y (tangage), et enfin une rotation d'un angle ϕ autour de l'axe x (roulis). Les angles de roulis et de tangage sont limités par $(-\frac{\pi}{2} \leq \theta \leq \frac{\pi}{2})$, $(-\frac{\pi}{2} \leq \phi \leq \frac{\pi}{2})$ et l'angle de lacet ψ n'est pas limité. Cela implique que le véhicule doit voler en mode normal et non en mode acrobatique: lacet du Quadrotor peut varier librement, mais les angles de roulis et de tangage ne peuvent pas prendre une valeur telle que le véhicule subit un vol inverse. De cette façon, ξ et η décrivent complètement le mouvement de translation et le mouvement de rotation du véhicule par rapport au repère inertiel.

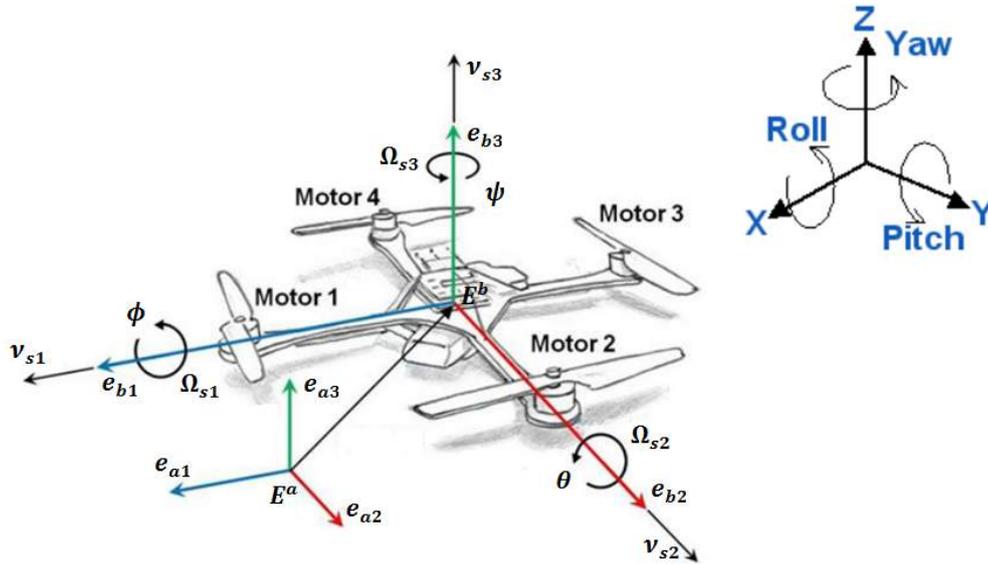


Figure 2.2: Représentation cartésienne du Quadrotor

Étant donné le vecteur $V = [u \ v \ w]^T \in E^b$ désigne la vitesse linéaire exprimée dans le repère lié au corps rigide E^b , donc la vitesse linéaire exprimée dans le repère inertiel est:

$$\dot{\xi} = R(\eta)V \quad (2.1)$$

Avec: $R(\eta)$ est la matrice de transformation du repère mobile au repère inertiel fixe, donnée par:

$$R(\eta) = \begin{bmatrix} C_\theta C_\psi & S_\phi S_\theta C_\psi - C_\phi S_\psi & C_\phi S_\theta C_\psi + S_\phi S_\psi \\ C_\theta S_\psi & S_\phi S_\theta S_\psi + C_\phi C_\psi & C_\phi S_\theta S_\psi - S_\phi C_\psi \\ -S_\theta & S_\phi C_\theta & C_\phi C_\theta \end{bmatrix} \quad (2.2)$$

D'une manière similaire, étant donné le vecteur de vitesse angulaire $\Omega = [p \ q \ r]^T \in E^b$ exprimée dans le repère lié au véhicule, où p , q et r représente les vitesses angulaires instantanées autour de l'axe X_b , l'axe Y_b et l'axe Z_b respectivement, elle est liée au taux de variation des angles roulis, tangage et lacet comme suit [101]:

$$\Omega = N(\eta)\dot{\eta} \quad (2.3)$$

Avec:

$$N(\eta) = \begin{bmatrix} 1 & 0 & -S_\theta \\ 0 & C_\phi & C_\theta S_\phi \\ 0 & -S_\phi & C_\theta C_\phi \end{bmatrix} \quad (2.4)$$

et $C_{(.)}$, $S_{(.)}$ représentent $\cos(.)$ et $\sin(.)$ respectivement.

2.3.1 Formalisme de Newton-Euler

Le Quadrotor est modélisé comme un corps rigide soumis aux forces d'actionnements et aux forces externes. Un résultat bien connu de la mécanique [102] [103] est que la dynamique des corps rigides peuvent être décrite en utilisant l'approche de Newton-Euler (basé sur les forces et les moments agissant sur le corps) et l'approche Euler-Lagrange (basée sur des hypothèses d'énergie). Avec l'approche Newton-Euler la dynamique est initialement formulée en terme des coordonnées du repère mobile (lié au corps rigide), puis exprimée en terme des coordonnées du repère inertiel en utilisant des transformations cinématiques. L'approche de Lagrange, au contraire, a besoin directement de l'utilisation de coordonnées généralisées (coordonnées du repère inertiel) et cela nécessite un symbolisme beaucoup plus lourd. Ainsi, le résultat final est le même, mais obtenu avec des notations différentes. Dans cette section, une approche de Newton-Euler est adoptée afin d'en tirer la dynamique de corps rigide du Quadrotor, car elle représente l'approche la plus simple pour la modélisation.

La dérivation des équations (2.1) et (2.3) en fonction du temps nous donne:

$$\begin{aligned} \ddot{\xi} &= R(\dot{V} + \dot{\Omega} \times V) \\ \dot{\Omega} &= N\ddot{\eta} + \left(\frac{\partial N}{\partial \phi} \dot{\phi} + \frac{\partial N}{\partial \theta} \dot{\theta} \right) \dot{\eta} \end{aligned} \quad (2.5)$$

En appliquant le formalisme de Newton-Euler sur le Quadrotor soumis aux forces $\sum F_{ext}$ et moments $\sum T_{ext}$ externes appliqués au centre de masse et exprimés au repère fixe du

corps rigide, on obtient les équations dynamiques suivantes:

$$\begin{bmatrix} mI_{3 \times 3} & 0_{3 \times 3} \\ 0_{3 \times 3} & I \end{bmatrix} \begin{bmatrix} \dot{V} \\ \dot{\Omega} \end{bmatrix} + \begin{bmatrix} \Omega \times mV \\ \Omega \times I\Omega \end{bmatrix} = \begin{bmatrix} \sum F_{ext} \\ \sum T_{ext} \end{bmatrix} \quad (2.6)$$

Où m est la masse totale du véhicule.

$I \in \mathcal{R}$ est la matrice d'inertie au centre de masse, exprimée dans le repère mobile lié au véhicule.

$$\begin{pmatrix} I_x & 0 & 0 \\ 0 & I_y & 0 \\ 0 & 0 & I_z \end{pmatrix}$$

$\sum F_{ext}$ et $\sum T_{ext}$ sont le vecteur de forces et de moments externes respectivement.

En utilisant les équations 2.4 et 2.5, les équations dynamiques du Quadrotor exprimées dans le repère inertiel sont données par:

$$\begin{aligned} \ddot{\xi} &= \frac{1}{m} R \sum F_{ext} \\ \ddot{\eta} &= (IN)^{-1} [\sum T_{ext} - I(\frac{\partial N}{\partial \phi} \dot{\phi} + \frac{\partial N}{\partial \theta} \dot{\theta}) \dot{\eta} - (N\dot{\eta}) \times (IN\dot{\eta})] \end{aligned} \quad (2.7)$$

2.3.2 Forces et moments

Le Quadrotor est un système mécanique sous actionné avec 6 sorties et possède seulement 4 entrées (PWM des moteurs). La dynamique des corps rigides décrit le mouvement du véhicule en fonction des forces et des moments qui agissent sur lui. La connaissance de ces forces généralisées permet de résoudre les équations (2.7) du mouvement du corps rigide. Les forces et les moments aérodynamiques sont obtenus en utilisant une combinaison de Théorie de Froude (momentum theory) et la théorie de l'élément lame [27] [40]. Ceci est basé sur le travail de Gary Fay durant le projet Mesicopter [104].

2.3.2.1 Force de gravité

La force de poids est appliquée au centre de gravité et dirigée le long de l'axe Z dans le sens négatif et exprimée dans le repère inertiel.

$$F_{grav} = R^T \begin{bmatrix} 0 \\ 0 \\ -mg \end{bmatrix} \quad (2.8)$$

Où R^T est la matrice de transformation des vecteurs exprimés dans le repère inertiel aux vecteurs exprimés dans le repère mobile lié au véhicule. Puisque le point d'application de la force est le centre de gravité, la force de gravité ne génère aucun moment.

2.3.2.2 Force de portance

La force de poussée F_i , avec $i = 1, 2, 3, 4$, est appliquée au centre du i -ème moteur, de distance l du centre de masse et dirigée le long de l'axe Z_b du repère mobile: $F_i \geq 0$. La force de poussée peut être considérée comme l'entrée du système Quadrotor, comme elle peut être modifiée en changeant la vitesse de rotation du moteur. D'un point de vue physique, la vitesse de rotation de chaque moteur est limitée, il y a donc aussi une borne supérieure des valeurs de chaque poussée: $F_i \leq F_{max}$ pour $i = 1, 4$. La variation des forces F_i , où $i = 1, 3$ ou $i = 2, 4$, produit des couples, avec une direction perpendiculaire au plan contenant les forces F_i . La poussée totale générée par les quatre moteurs exprimée dans le repère mobile est:

$$F = \begin{bmatrix} 0 \\ 0 \\ \sum_{i=1}^4 F_i \end{bmatrix} \quad (2.9)$$

L'expression de la poussée peut être obtenue à l'aide de la théorie des éléments de lame bien connue dans la littérature [30, 31, 16], qui conduit à

$$F_i = C_T \rho A r^2 \omega_i^2 \quad (2.10)$$

avec : C_T est le coefficient de portance, ρ densité de l'air, A est la surface du disque du rotor, r rayon du rotor et ω_i est la vitesse de rotation du moteur i .

2.3.2.3 Traînée de l'hélice

Les forces aérodynamiques agissant sur la lame de chaque rotor, dans une direction orthogonale à la force de poussée (réaction de la lame à la rotation), génèrent un moment supplémentaire qui doit être inclut dans le modèle. Ce moment est appelé moment de traînée et il est dirigé le long de l'axe Z_b . L'expression du moment de traînée peut être dérivée de manière similaire à celle utilisée pour dériver la force de poussée:

$$Q_i = C_Q \rho A r^3 \omega_i |\omega_i| \quad (2.11)$$

Où C_Q est le coefficient de traînée et ω_i est multiplié par sa valeur absolue pour préserver le signe de rotation pour les rotors à contre-rotation.

2.3.2.4 Traînée d'air

Le frottement de l'air (traînée d'air ou simplement traînée) est dû aux forces qui agissent sur le véhicule dans la direction de la vitesse relative d'écoulement d'air et représente la réaction de l'air au mouvement de translation et de rotation du corps dans l'espace. Les traînées de translation et de rotation sont généralement petites et sont approximées pour respecter la loi de Stoke [105]: la traînée de translation est donc considérée proportionnelle à la vitesse linéaire et le terme de traînée de rotation est proportionnel à la vitesse angulaire avec des coefficients k_t et k_r respectivement.

2.3.2.5 Effet gyroscopique

Lorsque le rotor tourne autour de son propre axe et d'un second axe, il tourne également autour d'un troisième axe qui est perpendiculaire au premier axe. Cet effet est appelé l'effet gyroscopique.

Le Quadrotor est propulsé par quatre hélices qui tournent à grande vitesse. Même si l'inertie de l'hélice est faible par rapport à celle du corps de véhicule, la vitesse de rotation élevée provoque un effet gyroscopique qui peut affecter le mouvement du véhicule lorsque les hélices tournent à des vitesses différentes. Dans la littérature, cet effet est négligé la plupart du temps, mais certains auteurs [106] l'incluent dans le modèle.

Le moment gyroscopique T_{gyro} provoqué par les rotations du véhicule avec rotors rotatifs peut être exprimé comme

$$T_{gyro} = I_r \left(\Omega \times \begin{bmatrix} 0 \\ 0 \\ 1 \end{bmatrix} \right) (\omega_1 - \omega_2 + \omega_3 - \omega_4) \quad (2.12)$$

Où Ω est la vitesse angulaire du véhicule exprimée dans le repère mobile, ω_i est la vitesse de rotation du moteur i et I_r est l'inertie du moteur.

2.3.2.6 Moment de roulis et moment de tangage

Comme chaque rotor génère une poussée, la différence entre chaque paire de rotors provoque les moments autour de l'axe X_b et Y_b dans le cadre du corps. La différence des poussées entre le rotor 2 et 4 provoque le moment de roulement autour de l'axe

X_b et la différence des poussées entre le rotor 1 et 3 provoque le moment de tangage autour de l'axe Y_b . La direction des moments est décidée selon la règle de la main droite. Par conséquent, le moment de roulage T_{roul} et le moment de tangage T_{tang} sont définis comme:

$$T_{roul} = l(F_2 - F_4) \quad (2.13)$$

$$T_{tang} = l(F_3 - F_1) \quad (2.14)$$

Où l est la distance entre le centre de gravité du véhicule et l'axe du moteur, F_i , $i = 1, \dots, 4$ sont les poussées des quatre moteurs.

2.3.2.7 Moment de lacet

La traînée Q_i fait tourner le Quadrotor autour de l'axe Z_b de repère mobile. La différence de traînée engendrée par les deux paires de moteurs (1,3) et (2,4) provoque le moment de lacet dans l'une des deux directions. Ainsi, le moment de lacet total est:

$$T_{lac} = Q_1 - Q_2 + Q_3 - Q_4 \quad (2.15)$$

2.3.3 Modèle complet

Dans les sections précédentes, nous avons déduit les équations de Newton-Euler pour un corps rigide dans l'espace. Ensuite, les forces et les moments externes importants qui agissent sur le Quadrotor sont analysés. Dans cette section, le modèle complet d'un Quadrotor sera d'abord proposé. Sur la base de certaines applications, ce modèle peut encore être simplifié.

A partir des équations (2.7), (2.8), (2.9) et (2.12), un modèle complet pour le Quadrotor est donné par les équations suivantes

$$\begin{aligned} \sum F_{ext} &= F - F_{aero} - F_{grav} \\ \sum T_{ext} &= T - T_{aero} - T_{gyro} \end{aligned} \quad (2.16)$$

$$\begin{aligned} \ddot{\xi} &= \frac{1}{m} RF - \frac{1}{m} RF_{aero} - G \\ \ddot{\eta} &= (IN)^{-1} \left(T - T_{aero} - I\Xi - (N\dot{\eta}) \times (IN\dot{\eta}) - (N\dot{\eta}) \times \sum_{i=1}^4 I_r (-1)^{i+1} \omega_i \right) \end{aligned} \quad (2.17)$$

avec

$$\Xi(\eta, \dot{\eta}) = \begin{bmatrix} -C_\theta \dot{\theta} \dot{\psi} \\ -S_\phi \dot{\phi} \dot{\theta} + C_\phi C_\theta \dot{\phi} \dot{\psi} - S_\phi S_\theta \dot{\theta} \dot{\psi} \\ -C_\phi \dot{\phi} \dot{\theta} - S_\phi C_\theta \dot{\phi} \dot{\psi} - C_\phi S_\theta \dot{\theta} \dot{\psi} \end{bmatrix} \quad (2.18)$$

$$T = \begin{bmatrix} l(F_2 - F_4) \\ l(F_3 - F_1) \\ \sum_{i=1}^4 (-1)^{i+1} Q_i \end{bmatrix} \quad (2.19)$$

et $F_{aero} = [k_t \dot{x}, k_t \dot{y}, k_t \dot{z}]^T$, $T_{aero} = [k_t \dot{\phi}, k_t \dot{\theta}, k_t \dot{\psi}]^T$

Dans (2.17), la première équation représente la dynamique de translation, et la deuxième équation représente la dynamique d'orientation. La dynamique angulaire n'est liée qu'à l'information angulaire. Alors que dans la dynamique de translation, les positions sont affectées par les changements des angles.

2.3.4 Modèle simplifié

Comme nous l'avons vu au chapitre 1, le Quadrotor a de nombreuses applications, comme la surveillance, la agriculture, la photographie aérienne et la vidéo, etc. Dans ces applications, les Quadrotors doivent généralement suivre une trajectoire ou survoler un point. Le mouvement de la machine est plutôt lent et sans déplacement agressive. Le Quadrotor Phantom [4] a un angle d'inclinaison maximum de 45 degrés et une vitesse de vol maximum de 10m/s. Toutefois, dans les applications, l'angle d'inclinaison est généralement inférieur à 14 degrés (0,25rad), la vitesse angulaire est inférieure à 0.2rad/s, et la vitesse de vol est inférieure à 2m/s.

$$\begin{aligned} \phi, \theta &\leq 0,25rad & \dot{\phi}, \dot{\theta}, \dot{\psi} &\leq 0,25rad/s \\ p, q, r &\leq 0.2rad/s & v &\leq 2m/s \end{aligned}$$

La relation entre le vecteur de vitesse angulaire exprimée dans le repère inertiel $\dot{\eta} = [\dot{\phi} \ \dot{\theta} \ \dot{\psi}]^T$ et la vitesse angulaire exprimée dans le repère mobile lié au véhicule $\Omega = [p \ q \ r]^T$ est donnée par

$$\Omega = \begin{bmatrix} 1 & 0 & -S_\theta \\ 0 & C_\phi & C_\theta S_\phi \\ 0 & -S_\phi & C_\theta C_\phi \end{bmatrix} \dot{\eta} \quad (2.20)$$

Donc la matrice de transformation $N(\eta)$ peut être bornée comme suit

$$N(\eta) = \begin{bmatrix} 1 & 0 & -S_\theta \\ 0 & C_\phi & C_\theta S_\phi \\ 0 & -S_\phi & C_\theta C_\phi \end{bmatrix} \preceq \begin{bmatrix} 1 & 0 & -0.247 \\ 0 & 0.969 & 0.239 \\ 0 & -0.247 & 0.939 \end{bmatrix} \quad (2.21)$$

On peut considérer les vitesses angulaires $\dot{\phi}$, $\dot{\theta}$, et $\dot{\psi}$ exprimées dans le repère mobile sont égales aux vitesses angulaires p , q , et r exprimées dans le repère inertiel respectivement

$$\dot{\phi} = p, \quad \dot{\theta} = q, \quad \dot{\psi} = r$$

Le Quadrotor est contrôlé généralement par les vitesses des moteurs ω_i . Comme les forces et les moments aérodynamiques sont proportionnels au carré de la vitesse de rotation du moteur, la force de poussée et les moment de lacet, roulis et tangage sont choisis comme des entrées de commande

$$\begin{aligned} u_1 &= \sum_{i=1}^4 F_i \\ u_2 &= l(F_2 - F_4) \\ u_3 &= l(F_3 - F_1) \\ u_4 &= \sum_{i=1}^4 (-1)^{i+1} Q_i \end{aligned} \quad (2.22)$$

ou encore

$$\begin{aligned} u_1 &= \kappa_b(\omega_1^2 + \omega_2^2 + \omega_3^2 + \omega_4^2) \\ u_2 &= l\kappa_b(\omega_2^2 - \omega_4^2) \\ u_3 &= l\kappa_b(\omega_3^2 - \omega_1^2) \\ u_4 &= \kappa_\tau(\omega_1^2 - \omega_2^2 + \omega_3^2 - \omega_4^2) \end{aligned} \quad (2.23)$$

où $\kappa_b = C_T \rho A r^2$ et $\kappa_\tau = C_Q \rho A r^3$ sont deux coefficients positifs.

Le modèle simplifié est donné par

$$\begin{aligned}
m\ddot{x} &= (\cos \phi \sin \theta \cos \psi + \sin \phi \sin \psi)u_1 \\
m\ddot{y} &= (\cos \phi \sin \theta \sin \psi - \sin \phi \cos \psi)u_1 \\
m\ddot{z} &= (\cos \phi \cos \theta)u_1 - mg \\
I_x\ddot{\phi} &= (I_y - I_z)\dot{\theta}\dot{\psi} + I_r\dot{\theta}\varpi + u_2 \\
I_y\ddot{\theta} &= (I_z - I_x)\dot{\phi}\dot{\psi} - I_r\dot{\phi}\varpi + u_3 \\
I_z\ddot{\psi} &= (I_x - I_y)\dot{\phi}\dot{\theta} + u_4
\end{aligned} \tag{2.24}$$

avec $\varpi = \omega_1 - \omega_2 + \omega_3 - \omega_4$, et $g = 9.81m/s^2$ la force de gravité.

Parameter	Value
l (m)	0.175
m (kg)	1.3
I_x (kgm ²)	0.081
I_y (kgm ²)	0.081
I_z (kgm ²)	0.142
κ_b (N/rad/s)	$3.8305e^{-6}$
κ_τ (Nm/rad/s)	$2.2518e^{-8}$

Tableau 2.1: Paramètres du Quadrotor

Le modèle simplifié (2.24) peut être exprimé sous une forme d'état comme suit:

$$\dot{X} = f(X, U) \tag{2.25}$$

avec le vecteur d'état $X = [x_{11}, x_{21}, \dots, x_{16}, x_{26}]^T \in \mathfrak{R}^{12}$, le vecteur des entrées $U = [u_1, u_2, u_3, u_4]^T$ et $x_{11} = x$, $x_{21} = \dot{x}$, $x_{12} = y$, $x_{22} = \dot{y}$, $x_{13} = z$, $x_{23} = \dot{z}$, $x_{14} = \phi$, $x_{24} = \dot{\phi}$, $x_{15} = \theta$, $x_{25} = \dot{\theta}$, $x_{16} = \psi$, $x_{26} = \dot{\psi}$.

En remplaçant le vecteur d'état X et la fonction non linéaire $f(X, U)$ on aura le

modèle d'état suivant:

$$\begin{aligned}
\dot{x}_{11} &= x_{21} \\
\dot{x}_{21} &= (\cos x_{14} \sin x_{15} \cos x_{16} + \sin x_{14} \sin x_{16}) \frac{1}{m} u_1 \\
\dot{x}_{12} &= x_{22} \\
\dot{x}_{22} &= (\cos x_{14} \sin x_{15} \sin x_{16} - \sin x_{14} \cos x_{16}) \frac{1}{m} u_1 \\
\dot{x}_{13} &= x_{23} \\
\dot{x}_{23} &= -g + \frac{\cos x_{14} \cos x_{15}}{m} u_1 \\
\dot{x}_{14} &= x_{24} \\
\dot{x}_{24} &= x_{25} x_{26} \left(\frac{I_y - I_z}{I_x} \right) + \frac{I_r}{I_x} x_{22} \varpi + \frac{1}{I_x} u_2 \\
\dot{x}_{15} &= x_{25} \\
\dot{x}_{25} &= x_{24} x_{26} \left(\frac{I_z - I_x}{I_y} \right) - \frac{I_r}{I_y} x_{21} \varpi + \frac{1}{I_y} u_3 \\
\dot{x}_{16} &= x_{26} \\
\dot{x}_{26} &= x_{24} x_{25} \left(\frac{I_x - I_y}{I_z} \right) + \frac{1}{I_z} u_4
\end{aligned} \tag{2.26}$$

2.4 Modèle de rafale de vent

La rafale de vent est un phénomène physique complexe. Il est typiquement modélisé en utilisant le modèle déterministe [107] ou le modèle stochastique [108]. Le second modèle suppose que la turbulence est un processus aléatoire gaussien stationnaire, ce qui est l'hypothèse la plus courante. La caractéristique stationnaire implique que la turbulence a une durée infinie, tandis que l'idée d'un processus gaussien est liée à la probabilité d'obtenir une vitesse de rafale donnée à un moment donné.

La densité spectrale de puissance (PSD) est une approche stochastique. Dans le modèle de turbulence atmosphérique basé sur la PSD, on suppose que l'intensité de la turbulence dépend de manière significative de circonstances spécifiques et elle est soumise à des changements dus aux conditions météorologiques, à l'altitude de vol et au gradient de température [109].

Le signal de rafale de vent est généré en faisant passer un bruit blanc à travers un filtre passe bas. Dans la littérature, deux principaux filtres peuvent être trouvés: le filtre de Dryden et le filtre de Von Karman. Comme le filtre Dryden a une forme plus simple que le Von Karman, nous préférons l'utiliser dans ce travail. Les filtres utilisés pour générer le modèle spectral de Dryden sont donnés par [110]:

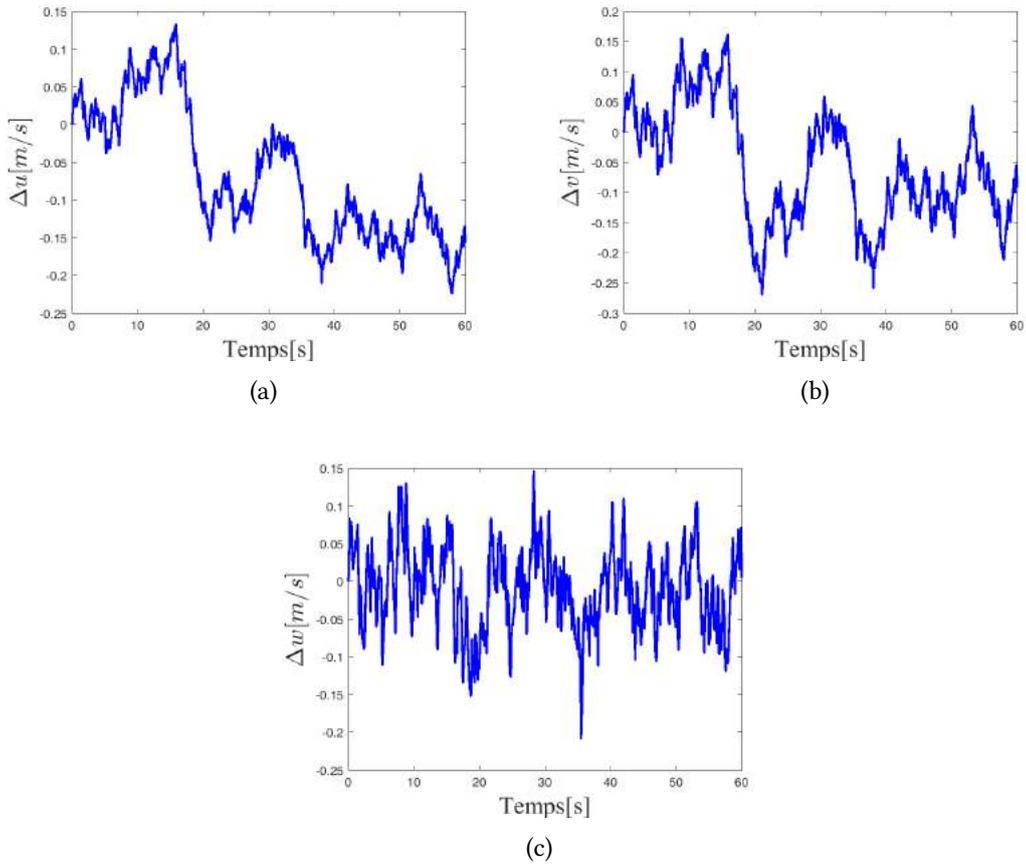


Figure 2.3: Rafale de vent généré par le modèle de Dryden

$$H_u(s) = \frac{\Delta u}{N_W} = \sigma_u \sqrt{\frac{2L_u}{\pi v}} \frac{1}{1 + \frac{L_u}{v}s} \quad (2.27)$$

$$H_v(s) = \frac{\Delta v}{N_W} = \sigma_v \sqrt{\frac{L_v}{\pi v}} \frac{1 + \frac{\sqrt{3}L_v}{v}s}{(1 + \frac{L_v}{v}s)^2} \quad (2.28)$$

$$H_w(s) = \frac{\Delta w}{N_W} = \sigma_w \sqrt{\frac{L_w}{\pi v}} \frac{1 + \frac{\sqrt{3}L_w}{v}s}{(1 + \frac{L_w}{v}s)^2} \quad (2.29)$$

avec N_W est un bruit blanc, v représente la vitesse relative du véhicule par rapport au courant d'air et $[\Delta u, \Delta v, \Delta w]^T$ est le changement dans les vitesses linéaires du véhicule dû à la rafale de vent. Les intensités de turbulence ($\sigma_u, \sigma_v, \sigma_w$) et les longueurs longitudinales de turbulence (L_u, L_v, L_w) décrivent le comportement de la rafale de vent. Dans la région de basse altitude (altitude $< 304.8m$), la longueur longitudinale et l'intensité

de turbulence sont données par:

$$L_w = h, L_u = L_v = \frac{h}{(0.177 + 0.000823h)^{1.2}} \quad (2.30)$$

$$\sigma_w = 0.1W_{20}, \frac{\sigma_u}{\sigma_w} = \frac{\sigma_v}{\sigma_w} = \frac{1}{(0.177 + 0.000823h)^{0.4}} \quad (2.31)$$

avec h est la hauteur du véhicule (altitude) et W_{20} est la vitesse du vent exprimée en échelle de Beaufort (BS). Dans notre travail, nous avons considéré une vitesse de vent typique de $3.4m/s$ (BS 3) et une altitude de $3m$. La figure 2.3 montre l'évolution temporelle de la rafale de vent le long des trois axes (X, Y, Z).

2.5 Simulation du modèle dynamique

Pour montrer le comportement non linéaire et couplé du Quadrotor, nous avons simulé le modèle dynamique en boucle ouverte. Le vecteur des vitesses des moteurs $\omega = [\omega_1, \omega_2, \omega_3, \omega_4]$ est utilisé comme vecteur d'entrée et le vecteur de sortie contient la position et l'attitude $y = [x, y, z, \phi, \theta, \psi]$.

Nous avons simulé deux mouvements du Quadrotor, le premier mouvement est un décollage vertical (vertical-takeoff) et le deuxième un mouvement horizontal. Pour le premier cas nous avons utilisé les entrées de système suivantes $\omega_1 = \omega_2 = \omega_3 = \omega_4 = 50\sin(0.9t) + 912$ pour une durée de $tf = 5s$, et les entrées $\omega_1 = 50\sin(0.9t) + 712, \omega_2 = 50\sin(0.9t) + 912, \omega_3 = 50\sin(0.9t) + 1112, \omega_4 = 50\sin(0.9t) + 912$ pour le deuxième cas.

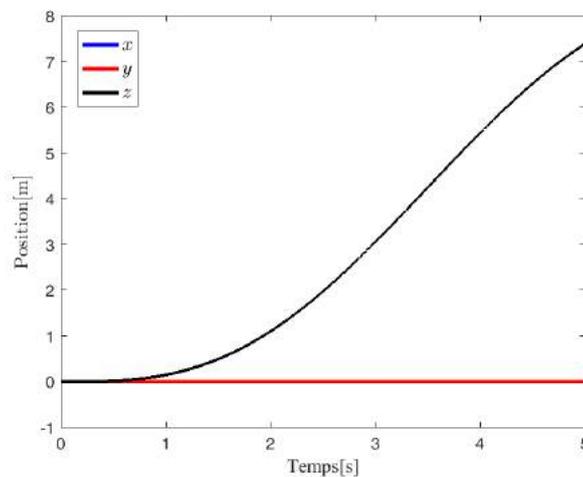
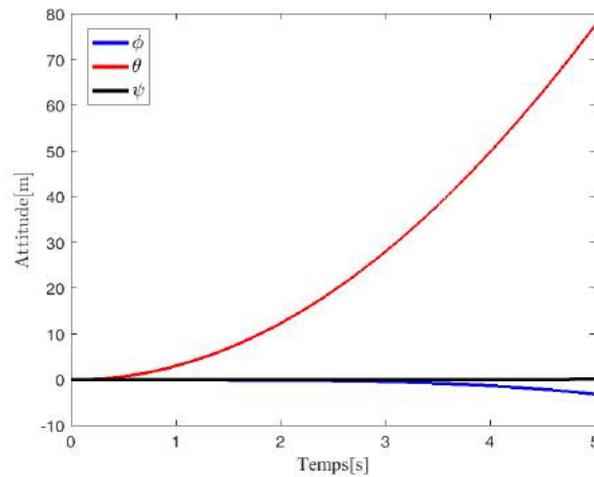
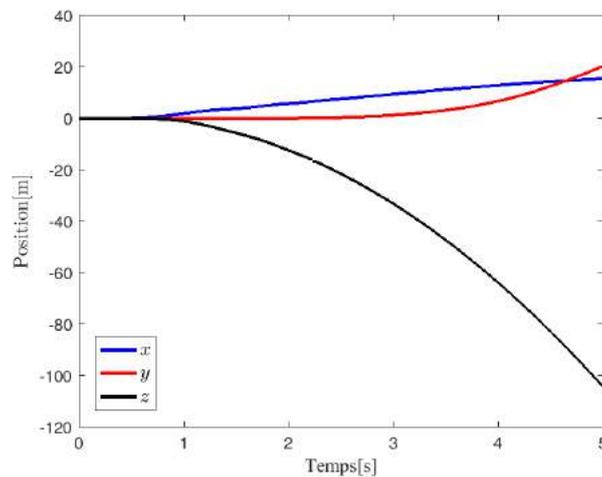


Figure 2.4: Décollage vertical (vertical-takeoff)

La figure 2.4 montre un déplacement le long de l'axe Z , alors que la figure 2.5 montre un déplacement dans l'espace 3D et un mouvement de rotation autour des trois axes (X, Y, Z). en réalité, si on implémente le premier cas sur une plate-forme expérimentale du Quadrotor on n'aura pas que le mouvement vertical, on aura une variation des six degrés de libertés du véhicule et ça a cause des dynamiques et effets non modélisés tels que l'aérodynamique de l'hélice, l'effet de sol et la rafale de vents dans les applications *outdoor* etc.



(a)



(b)

Figure 2.5: Mouvement horizontal

2.6 Conclusion

Dans ce chapitre, le modèle dynamique du Quadrotor a été établi par le formalisme de Newton-Euler. Nous avons présenté un modèle qui décrit presque tous les phénomènes physiques agissant sur le Quadrotor par un vecteur des équations différentielles non linéaires de deuxième ordre. Ce modèle montre la nature couplée, complexe, non linéaire, multi variable et sous-actionnée, ce qui rend son contrôle relativement difficile. Nous avons aussi présenté un modèle simplifié proposé dans la littérature. Un modèle de rafale de vent décrit les effets de la turbulence atmosphérique sur le véhicule est présenté. Une simulation du modèle dynamique est implémenté pour montrer le comportement non linéaire du Quadrotor.

Dans le prochain chapitre, nous allons introduire un algorithme de planification de trajectoire optimale en terme d'énergie pour le Quadrotor.

Chapitre 3

Génération de trajectoire optimale en terme d'énergie pour un Quadrotor

3.1 Introduction

Récemment, de nouvelles applications prometteuses ont émergé pour les drones miniatures à voilure tournante en particulier le Quadrotor, comme la livraison de colis, la cinématographie, la surveillance de l'agriculture et la manipulation aérienne. Cependant, ces applications sont encore limitées puisque l'énergie embarquée dont la source est la batterie Lithium-Polymère (LiPo) fournit un temps de vol entre 15 et 45 minutes ce qui limite la classe de mission qui peut être réalisée avec succès par le drone. Pour entreprendre ce problème, de nombreux efforts ont été faits pour réduire le poids des drones grâce à l'utilisation de châssis en fibre de carbone et l'amélioration du rapport énergie poids. Ces solutions réussissent à réduire la consommation d'énergie lors de fonctionnement dans des régimes affamés d'énergie; néanmoins, aucun progrès technologique significatif n'a été fait jusqu'à présent.

Dans ce chapitre, nous proposons une solution au problème de limitation de l'endurance de vol pour un type de véhicules aériens à voilure tournante à savoir le Quadrotor. Les Quadrotors sont des robots volants dotés de plusieurs rotors; ainsi, une grande partie de leur énergie est consommée par les rotors afin de maintenir le véhicule dans l'air. Dans ce concept, nous avons introduit un modèle énergétique composé de la dynamique du mouvement de véhicule, de la dynamique des moteurs et de la dynamique de batterie; ensuite, le modèle proposé a été validé par simulation pour montrer la possibilité d'économiser de l'énergie. Nous formulons deux problèmes de commande optimale afin de calculer l'énergie minimale et le temps minimal. Dans le premier problème, nous cherchons à trouver les entrées de commande et la trajectoire du véhicule entre deux configurations initiales et finales qui minimisent l'énergie consommée durant une mission bien spécifique. Alors que dans le second, le temps minimal et la trajectoire du véhicule sont calculés pour une énergie désirée.

3.2 Travaux antérieurs

De nombreuses études ont été proposées récemment contribuant à économiser de l'énergie et à augmenter l'endurance des drones miniatures à voilure tournante. Ces contributions ont principalement porté sur la conception du système de chargement et/ou remplacement automatique des batteries. Un système d'échange de batterie pour un groupe de drone miniature à voilure tournante a été proposé dans [111]. En plus du mécanisme d'échange de batterie, le système comprend un algorithme en ligne qui peut superviser le réapprovisionnement de plusieurs drones fonctionnant simultanément, déterminer

quand le véhicule nécessite une réalimentation et effectuer un atterrissage de précision sur la plate-forme d'atterrissage du mécanisme d'échange de batterie. Dans [112], la conception, l'essai et la construction d'une station de recharge au sol autonome pour un Quadrotor alimenté par batterie ont été présentés. Un algorithme de gestion de l'énergie a été implémenté pour un système multi-agent où la priorité est donnée au groupe pour assurer l'optimalité de la solution quel que soit le nombre, la position et la densité de l'environnement. Dans [113], un système de maintenance mécatronique de la batterie pour prolonger le temps de fonctionnement des batteries pour les drones miniatures à voilure tournante a été développé. D'autres études ont introduit un modèle d'estimation d'endurance de vol, dans [114] un modèle simple est proposé pour estimer l'endurance d'un Quadrotor à l'intérieur, alors que dans [115] une caractérisation de la consommation d'énergie d'un drone miniature à voilure tournante alimenté par une batterie LiPo et un modèle d'estimation d'endurance précis ont été introduits. Afin de prolonger le temps de vol des drones miniatures à voilure tournante, un contrôleur d'altitude basé sur l'état de charge de la batterie pour un drone miniature à six rotors a été proposé dans [116], où un système de surveillance de la batterie a été conçu pour estimer l'état de charge (SOC) et l'utiliser pour calculer la loi de commande synthétisée. Compte tenu de la commande en poursuite de trajectoire chemin avec consommation d'énergie minimale, les auteurs de [117] évaluent la relation entre la vitesse de navigation et la consommation d'énergie pour un Quadrotor, qui parcourt une trajectoire désirée au moyen d'un teste expérimental. Ensuite, un nouveau contrôleur de suivi de trajectoire est proposé dans lequel la vitesse du véhicule a un profil dynamique qui varie avec les exigences géométriques du chemin souhaité.

Le problème de la planification de trajectoire optimale en terme d'énergie pour les drones miniatures à voilure tournante a suscité moins d'intérêt dans la littérature. Dans [118], une approche a été proposée pour minimiser la consommation d'énergie pour un hexarotor dans une mission multi-cible. L'approche proposée consiste à une procédure heuristique pour résoudre 'Generalized Traveling Salesman Problem with Neighborhoods' (GTSPN), en utilisant une approche d'algorithme génétique qui incorpore un modèle dynamique affiné d'un hexarotor pour simuler la consommation d'énergie. Alors que dans [119] les trajectoires optimales en terme d'énergie ont été obtenues entre des configurations initiales et finales données pour un Quadrotor de 6-DOF en résolvant un problème de commande optimale. Le temps minimal d'une mission bien définie a été calculé en résolvant un problème de commande optimal associé.

La résolution de ces types de problèmes de commande optimale est compliquée en raison de la dynamique non linéaire du Quadrotor et de ses contraintes de faisabilité

sur les manœuvres. Les méthodes d'optimisation directe et indirecte sont les deux approches générales proposées pour résoudre les problèmes de planification de trajectoire optimale [120]. La méthode indirecte, qui est basée sur les équations de Hamilton et de Lagrange, a un faible taux de convergence et très sensible aux estimations initiales des solutions. Puisque la dynamique du système est complexe, il est très difficile d'obtenir une estimation initiale appropriée. Une autre approche est l'optimisation directe, tel que la méthode de collocation et la programmation non linéaire, qui discrétise la trajectoire en plusieurs segments, puis résolve le problème d'optimisation paramétré. Cette approche est très intéressante dans les applications complexes car elle ne nécessite pas de dériver les conditions nécessaires et fournit une estimation initiale pour les variables complémentaires [121]. En plus des deux approches, la cinématique de l'avion peut être simplifiée en fonction des caractéristiques des différentes manœuvres de vol, appelées approche de réduction de modèle. Les trajectoires générées par la suite sont composées de plusieurs modes [122]. Dans notre travail, nous utilisons la méthode de programmation non linéaire.

3.3 Modèle énergétique

Avant de procéder à l'optimisation énergétique, nous avons besoin d'un modèle énergétique pour avoir une idée sur l'énergie consommée par le véhicule pendant la mission et fournir une relation directe entre l'énergie et la dynamique du véhicule. Le modèle proposé dans cette section est composé de quatre blocs comme le montre la figure 3.2. Le premier bloc représente la dynamique du mouvement de véhicule, le deuxième contient la dynamique des moteurs, dans le troisième bloc on trouve la dynamique de la batterie et dans le quatrième la fonction de rendement des moteurs [123].

3.3.1 Dynamique du mouvement

Le modèle dynamique du Quadrotor (2.24) est présenté dans le chapitre précédent.

Nous utilisons la représentation d'état (2.25):

$$\dot{X} = f(X, U) \quad (3.1)$$

3.3.2 Dynamique des moteurs

Le système d'actionnement d'un Quadrotor est typiquement constitué d'une batterie LiPo, de quatre moteurs à courant continu sans balai (BLDC) et d'un étage de commande

pour contrôler la vitesse angulaire de chaque moteur. Les moteurs électriques DC sont bien modélisés par un circuit contenant une résistance, un inducteur et un générateur de tension en série [124].

$$v_m(t) = Ri(t) + L \frac{\partial i(t)}{\partial t} + \frac{\omega(t)}{k_v} \quad (3.2)$$

avec R la résistance interne du moteur, L l'inductance, $\omega(t)$ la vitesse du moteur et k_v la constante de tension du moteur, exprimée en $rad/s/volt$. De plus, le couple du moteur τ peut être modélisé comme étant proportionnel au courant $i(t)$ à travers la constante de couple, k_T , exprimée en Nm/A .

$$\tau(t) = k_T i(t) \quad (3.3)$$

La dynamique du moteur est modélisée comme une simple équation différentielle de premier ordre (3.4) où $\dot{\omega}$ est pilotée par le couple moteur et le couple de frottement de charge $Q_f(\omega(t))$. L'inertie, I_r inclut le moteur et l'hélice, le couple moteur vient du générateur de tension, et le couple de frottement de la charge résulte du glissement de l'hélice $Q_f(\omega(t)) = \kappa_\tau \omega^2(t)$, κ_τ est le coefficient de traînée.

$$I_r \frac{\partial \omega(t)}{\partial t} = \tau(t) - Q_f(\omega(t)) - D_v \omega(t) \quad (3.4)$$

où D_v est le coefficient d'amortissement visqueux du moteur Nms/rad . Typiquement, l'inductance des petits moteurs à courant continu est négligée par rapport à la réponse physique du système et peut donc être ignorée. En régime permanent, le courant $i(t)$ est constant, et l'équation (3.2) se réduit à :

$$v_m(t) = Ri(t) + \frac{\omega(t)}{k_v} \quad (3.5)$$

où le terme $\frac{1}{k_v} \omega(t)$ représente la force électromotrice du moteur. Le tableau 3.1 montre les coefficients du moteur BLDC utilisés dans notre étude.

3.3.3 Dynamique de la batterie

La batterie Li-ion possède un grand potentiel pour le développement et l'optimisation des systèmes de stockage d'énergie dans le future. En plus de leur petite taille et de leur faible poids, les batteries Li-ion offrent une densité d'énergie élevée et une efficacité de stockage proche de 100%, ce qui les rend idéales pour les applications embarquées. Mais

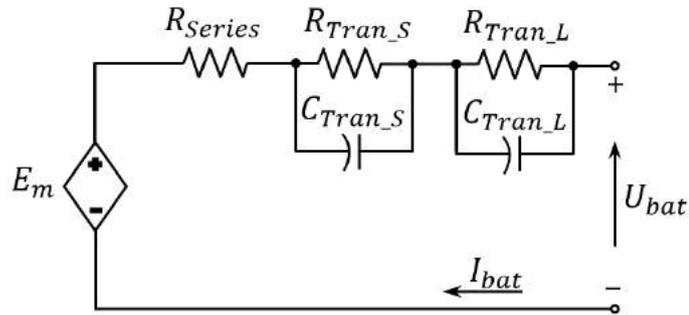


Figure 3.1: schéma de la batterie

l'inconvénient majeur de ce type de batterie est son coût élevé [125] [126].

Un modèle physique de la batterie Li-ion a été présenté. Le modèle de batterie a été développé pour l'entrée comme étant le courant et comme sorties la tension et l'état de charge (SOC). Ce modèle ne prend pas en compte l'influence de la température et du phénomène autodécharge [125]. Cependant, il donne des résultats proches de la réalité. Le modèle est basé sur les deux équations (3.6) (3.7), l'état de charge (SOC) et la tension aux bornes de la cellule.

$$SOC = 100 \left[1 - \frac{\int I}{Q} \right] \quad (3.6)$$

$$U_{bat} = E_m - R_{int}I \quad (3.7)$$

E_m est la tension en circuit ouvert (VOC), son expression est la suivante:

$$E_m = E_0 - K \left[\frac{Q}{Q - \int I} \right] + Ae^{-B \int I} \quad (3.8)$$

E_0 est la tension en circuit ouvert à pleine charge, elle est différente de la tension nominale donnée par le fabricant. Q est la capacité de la cellule en Ah, le paramètre K

Paramètre	Valeur
I_r ($kg.m^2$)	$4.1904 \cdot 10^{-5}$
k_T ($N.m/A$)	$0.0104 \cdot 10^{-3}$
k_v ($rad/s/volt$)	96.342
D_v (Nms/rad)	$0.2 \cdot 10^{-3}$
R (Ohm)	0.2

Tableau 3.1: Coefficients du moteur

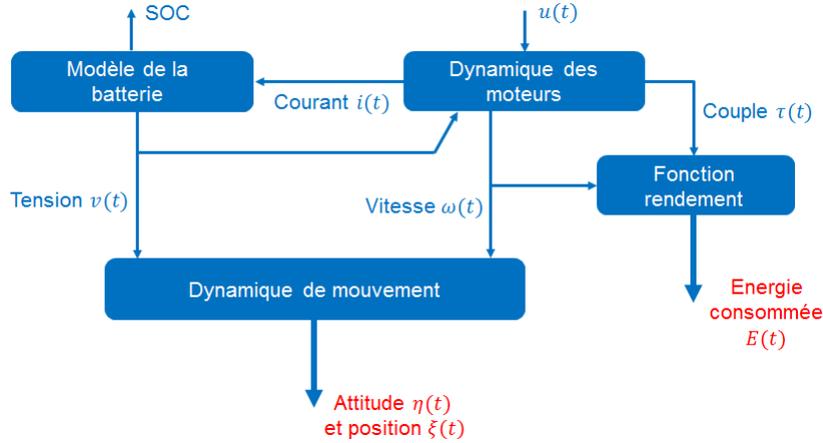


Figure 3.2: Modèle énergétique

représente le biais de voltage, A la tension exponentielle et B la capacité exponentielle sont des paramètres expérimentaux déterminés à partir de la courbe de décharge.

Le modèle est non linéaire, donc il a fallu adapter le courant d'entrée à celui vu par une cellule élémentaire en divisant le courant d'entrée par le nombre de branches parallèles M_{bat} . Pour la tension de sortie de notre batterie, nous avons simplement multiplié la tension de sortie de la cellule par le numéro de branche en série N_{bat} . Cette méthodologie nécessite de supposer que les cellules ont le même comportement.

Paramètre	Valeur
Q (Ah)	1.55
R_{int} (Ohm)	0.02
E_0 (volt)	1.24
K (volt)	$2.92 \cdot 10^{-3}$
A	0.156
B	2.35

Tableau 3.2: Paramètres de la batterie

3.3.4 Énergie consommée et rendement du moteur

L'énergie consommée par le véhicule pendant la mission est définie par:

$$E_c = \int_{t_0}^{t_f} \sum_{j=1}^4 \tau_j(t) \omega_j(t) dt \quad (3.9)$$

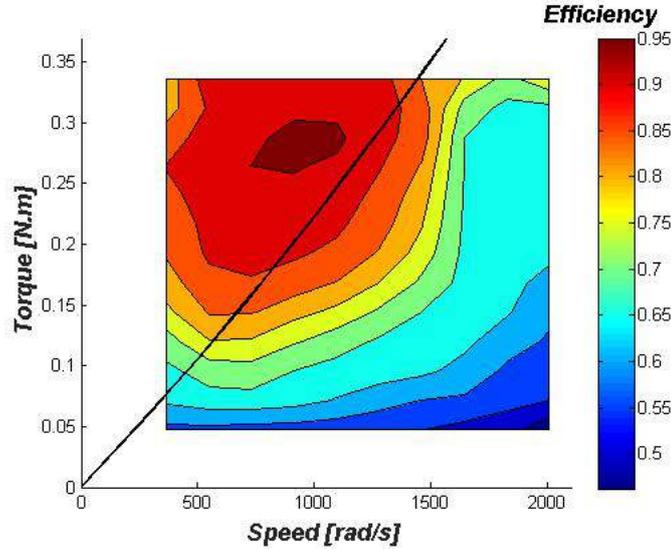


Figure 3.3: Rendement des moteurs

avec $\tau_j(t)$ est le couple généré par le moteur j et $\omega_j(t)$ est la vitesse du moteur à l'instant t . En utilisant l'équation (3.4) pour les quatre moteurs, l'équation (3.9) devient:

$$E_c = \int_{t_0}^{t_f} \sum_{j=1}^4 \left(J\dot{\omega}_j(t) + \kappa_\tau \omega_j^2(t) + D_v \omega_j(t) \right) \omega_j(t) dt \quad (3.10)$$

Afin de rendre notre modèle énergétique plus réaliste, une fonction de rendement est identifiée et ajoutée à la fonction d'énergie (3.10). Le rendement du moteur à courant continu sans balai utilisé pour actionner le Quadrotor est en fonction du couple moteur et de la vitesse de rotation $f_r(\tau(t), \omega(t))$. Nous avons utilisé l'interpolation polynomiale pour l'identification de la fonction de rendement, donc $f_r(\tau(t), \omega(t))$ peut être formulée comme suit:

$$f_r(\tau(t), \omega(t)) = a(\omega(t))\tau^3(t) + b(\omega(t))\tau^2(t) + c(\omega(t))\tau(t) + d(\omega(t)) \quad (3.11)$$

et

$$\begin{aligned} a(\omega(t)) &= a_1\omega^2(t) + b_1\omega(t) + c_1 \\ b(\omega(t)) &= a_2\omega^2(t) + b_2\omega(t) + c_2 \\ c(\omega(t)) &= a_3\omega^2(t) + b_3\omega(t) + c_3 \\ d(\omega(t)) &= a_4\omega^2(t) + b_4\omega(t) + c_4 \end{aligned}$$

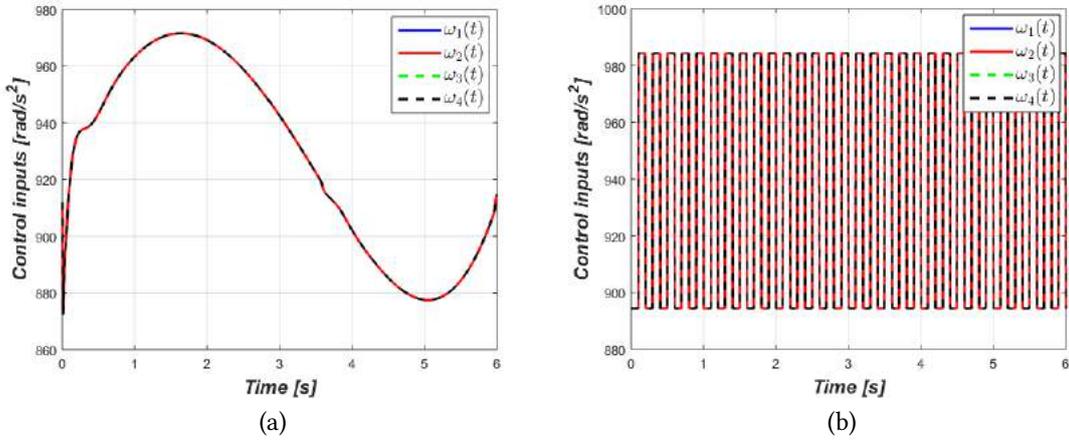


Figure 3.4: Entrées de commande, mission1 (a) mission2 (b)

Les paramètres du polynôme sont calculés en utilisant *Matlab* pour les quatre moteurs (nous supposons que les moteurs sont identiques).

$$\begin{array}{lll}
 a_1 = -1.72 \cdot 10^{-5} & b_1 = 0.014 & c_1 = -0.8796 \\
 a_2 = 1.95 \cdot 10^{-5} & b_2 = -0.0157 & c_2 = 0.3385 \\
 a_3 = -6.98 \cdot 10^{-6} & b_3 = 5.656 \cdot 10^{-3} & c_3 = 0.2890 \\
 a_4 = 4.09 \cdot 10^{-7} & b_4 = -3.908 \cdot 10^{-4} & c_4 = 0.1626
 \end{array}$$

Ensuite, l'énergie consommée (3.10) devient.

$$E_c = \int_{t_0}^{t_f} \sum_{j=1}^4 \frac{\left(J\omega_j(t) + \kappa_\tau \omega_j^2(t) + D_v \omega_j(t) \right)}{f_{r,j}(\tau_j(t), \omega_j(t))} \omega_j(t) dt \quad (3.12)$$

3.3.5 Effet des entrées de commande sur la consommation d'énergie

Afin de valider le modèle énergétique proposé, nous avons simulé deux missions de décollage vertical avec deux trajectoires d'altitude différentes, comme le montre la Fig. 3.4a. Les différentes entrées de commande utilisées pour les deux missions, sont indiquées sur la Fig. 3.5. Les deux missions ont la même durée $t_f = 6s$ et la même configuration initiale et finale.

Nous avons calculé l'énergie consommée pour les deux missions en utilisant l'équa-

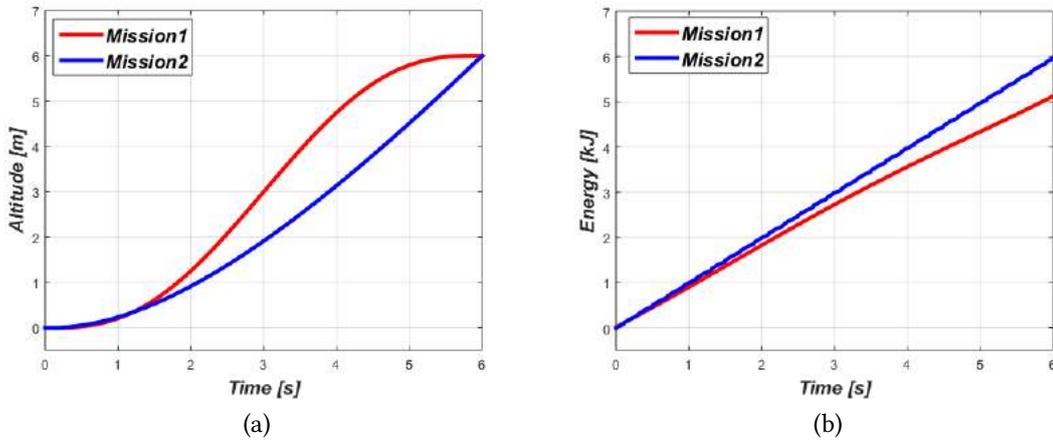


Figure 3.5: Altitude du Quadrotor (a) et Énergie consommée (b)

tion (3.12). Pour la première mission (ligne rouge), l'énergie consommée est $E_1 = 5.12kJ$, alors que dans la seconde mission (ligne bleue) l'énergie consommée est $E_2 = 5.97kJ$; ainsi le gain entre les deux missions avec la même configuration initiale et finale est de 17% de l'énergie totale. Pour une mission spécifique, l'optimisation de l'énergie peut être atteint; ce dont nous avons besoin, c'est de trouver les entrées de commande et la trajectoire qui permettent une consommation d'énergie optimale.

3.4 Optimisation d'énergie

Dans le problème d'optimisation de l'énergie, on cherche à trouver la trajectoire et les entrées de commande pour le Quadrotor qui minimisent l'énergie consommée tout en satisfaisant un ensemble de contraintes sur les états et les entrées de commande. D'un point de vue de commande optimale, nous essayons de calculer une solution en boucle ouverte pour un problème de commande optimale. Nous recherchons les entrées de commande du système qui minimisent l'énergie consommée lors d'une mission entre deux configurations initiales et finales spécifiques [127] [123].

Maintenant, le problème de génération de trajectoire optimale en terme d'énergie est formulé comme un problème de minimisation, dans lequel l'énergie consommée finale $E_c(t_f)$ est utilisée comme fonction de coût. De plus, les variables d'état dans $[x, \dot{x}, y, \dot{y}, z, \dot{z}, \phi, \dot{\phi}, \theta, \dot{\theta}, \psi, \dot{\psi}]^T$ et les variables de commande dans $[\omega_1, \omega_2, \omega_3, \omega_4]^T$ sont contraintes pour satisfaire la dynamique du véhicule et les conditions aux limites. La mission est de voler entre des positions initiales et finales spécifiées pendant un inter-

valle de temps $[t_0, t_f]$, tel que t_0 et t_f sont donnés.

Sur la base de la description ci-dessus, le problème de commande optimale est formulé comme suit:

$$\min_{(\omega_j, \tau_j)} E_c(t_f) \quad (3.13)$$

soumis aux contraintes

$$\begin{aligned} \dot{x}_{11} &= x_{21} \\ \dot{x}_{21} &= (\cos x_{14} \sin x_{15} \cos x_{16} + \sin x_{14} \sin x_{16}) \frac{1}{m} u_1 \\ \dot{x}_{12} &= x_{22} \\ \dot{x}_{22} &= (\cos x_{14} \sin x_{15} \sin x_{16} - \sin x_{14} \cos x_{16}) \frac{1}{m} u_1 \\ \dot{x}_{13} &= x_{23} \\ \dot{x}_{23} &= -g + \frac{\cos x_{14} \cos x_{15}}{m} u_1 \\ \dot{x}_{14} &= x_{24} \\ \dot{x}_{24} &= x_{25} x_{26} \left(\frac{I_y - I_z}{I_x} \right) + \frac{I_r}{I_x} x_{22} \varpi + \frac{1}{I_x} u_2 \\ \dot{x}_{15} &= x_{25} \\ \dot{x}_{25} &= x_{24} x_{26} \left(\frac{I_z - I_x}{I_y} \right) - \frac{I_r}{I_y} x_{21} \varpi + \frac{1}{I_y} u_3 \\ \dot{x}_{16} &= x_{26} \\ \dot{x}_{26} &= x_{24} x_{25} \left(\frac{I_x - I_y}{I_z} \right) + \frac{1}{I_z} u_4 \end{aligned} \quad (3.14)$$

et

$$\begin{aligned} |\phi| &\leq \frac{\pi}{2}, \quad |\theta| \leq \frac{\pi}{2}, \quad |\psi| \leq - \\ \omega_{min} &\leq \omega_j \leq \omega_{max} \\ 0 &\leq T \leq T_{max}, \quad |u_k| \leq u_{max}, \quad k = 1, 2, 3 \end{aligned} \quad (3.15)$$

avec les conditions aux limites:

$$\begin{aligned} [x(t_0), y(t_0), z(t_0), \phi(t_0), \theta(t_0), \psi(t_0)]^T &= [x_0, y_0, z_0, \phi_0, \theta_0, \psi_0]^T \\ [x(t_f), y(t_f), z(t_f)]^T &= [x_f, y_f, z_f]^T \end{aligned} \quad (3.16)$$

Les contraintes supplémentaires dans (3.15) sont associées à la dynamique du véhicule, avec ω_{min} and ω_{max} sont la vitesse minimale et maximale réalisable des rotors du véhicule, respectivement. Les angles de roulis et de tangage, ϕ , θ , doivent satisfaire $|\phi| \leq \frac{\pi}{2}$, $|\theta| \leq$

$\frac{\pi}{2}$ basés sur leur définition physique, et $|\dot{\psi}| \leq \Psi$ est nécessaire pour générer une trajectoire lisse, avec Ψ est le taux de variation maximal de l'angle de lacet.

3.5 Optimisation du temps

Le problème de la génération de trajectoire minimale en terme du temps est un sous-problème de commande optimale, où l'objectif étant de calculer le temps minimal passé par le véhicule en se déplaçant entre deux positions initiales et finales avec une énergie fixe. Les variables d'état sont $[x, \dot{x}, y, \dot{y}, z, \dot{z}, \phi, \dot{\phi}, \theta, \dot{\theta}, \psi, \dot{\psi}, \dot{E}_c]^T$ et les variables de commande $[\omega_1, \omega_2, \omega_3, \omega_4]^T$, alors que la mission est la même que dans le premier problème sauf le temps final t_f qui est libre dans ce cas [127].

Le problème de commande optimale est formulé comme suit:

$$\min_{(\omega_j, \tau_j)} t_f \quad (3.17)$$

soumis aux contraintes

$$\begin{aligned} \dot{x}_{11} &= x_{21} \\ \dot{x}_{21} &= (\cos x_{14} \sin x_{15} \cos x_{16} + \sin x_{14} \sin x_{16}) \frac{1}{m} u_1 \\ \dot{x}_{12} &= x_{22} \\ \dot{x}_{22} &= (\cos x_{14} \sin x_{15} \sin x_{16} - \sin x_{14} \cos x_{16}) \frac{1}{m} u_1 \\ \dot{x}_{13} &= x_{23} \\ \dot{x}_{23} &= -g + \frac{\cos x_{14} \cos x_{15}}{m} u_1 \\ \dot{x}_{14} &= x_{24} \\ \dot{x}_{24} &= x_{25} x_{26} \left(\frac{I_y - I_z}{I_x} \right) + \frac{I_r}{I_x} x_{22} \varpi + \frac{1}{I_x} u_2 \\ \dot{x}_{15} &= x_{25} \\ \dot{x}_{25} &= x_{24} x_{26} \left(\frac{I_z - I_x}{I_y} \right) - \frac{I_r}{I_y} x_{21} \varpi + \frac{1}{I_y} u_3 \\ \dot{x}_{16} &= x_{26} \\ \dot{x}_{26} &= x_{24} x_{25} \left(\frac{I_x - I_y}{I_z} \right) + \frac{1}{I_z} u_4 \\ \dot{E}_c &= \sum_{j=1}^4 \frac{1}{f_{r,j}(\tau_j, \omega_j)} \left(J \omega_j + \kappa_\tau \omega_j^2 + D_v \omega_j \right) \omega_j \end{aligned} \quad (3.18)$$

et

$$\begin{aligned}
|\phi| &\leq \frac{\pi}{2}, \quad |\theta| \leq \frac{\pi}{2}, \quad |\dot{\psi}| \leq - \\
\omega_{min} &\leq \omega_j \leq \omega_{max} \\
0 \leq T &\leq T_{max}, \quad |u_k| \leq u_{max}, \quad k = 1, 2, 3
\end{aligned} \tag{3.19}$$

avec les conditions aux limites:

$$\begin{aligned}
[x(t_0), y(t_0), z(t_0), \phi(t_0), \theta(t_0), \psi(t_0), E_c(t_0)]^T &= [x_0, y_0, z_0, \phi_0, \theta_0, \psi_0, E_{c,0}]^T \\
[x(t_f), y(t_f), z(t_f), E_c(t_f)]^T &= [x_f, y_f, z_f, E_{c,f}]^T
\end{aligned} \tag{3.20}$$

Il faut noter que la fonction d'énergie E_c a été traitée comme une variable d'état dans ce problème, avec les conditions aux limites $E_c(t_0) = E_{c,0}$ et $E_c(t_f) = E_{c,f}$.

3.6 Méthode de programmation non linéaire

Les problèmes de commande optimale (3.13)-(3.16) et (3.17)-(3.20) présentés dans les sections précédentes sont des problèmes d'optimisation complexes et non linéaires. L'approche générale pour résoudre ce type de problème est la méthode de collocation directe. L'idée de base de la collocation directe est de discrétiser une solution continue à un problème représenté par des variables d'état et de commande en utilisant une interpolation linéaire pour satisfaire les équations différentielles. De cette manière, un problème de commande optimale est transformé en un problème de programmation non linéaire (NLPP).

Dans notre étude, le problème de commande optimale proposé a été résolu numériquement en utilisant un logiciel Matlab appelé *GPOPS-II* [128]. Le logiciel utilise une méthode de collocation orthogonale en quadrature de Legendre-Gauss-Radau (LGR) [129] [130] dans laquelle le problème de commande optimale en temps continu est transcrit à un grand problème de programmation non linéaire éparpillé. Il utilise une méthode de raffinement de maillage adaptatif qui détermine le nombre d'intervalles de maillage et le degré du polynôme approximatif dans chaque intervalle de maillage afin d'obtenir une précision spécifiée. Le logiciel permet l'utilisation de deux solveurs de programmation non linéaire (NLP) pour résoudre le NLPP. Le premier est le solveur NLP open-source IPOPT (Interior Point OPTimizer) [131], et le second est le solveur NLP SNOPT (Sparse Nonlinear OPTimizer) [132].

3.7 Simulation et comparaison

3.7.1 Résultats de simulation

Les problèmes d'optimisation (3.13)-(3.16) et (3.17)-(3.20) ont été résolus en utilisant le solveur PNL open source IPOPT en mode dérivée seconde (Newton complet), avec le solveur linéaire direct massivement parallèle clairsemé multi-frontal MUMPS [133] disponible publiquement. Tous les résultats ont été obtenus en utilisant la forme d'intégration implicite de la méthode de Radau collocation et différentes formes de la méthode de raffinement du maillage ph en utilisant les paramètres par défaut du solveur NLP et la routine de mise à l'échelle automatique dans GPOPS-II.

Dans nos tests, nous avons considéré le Quadrotor *DJI Phantom 2* [4] avec un système de propulsion multi-rotor (moteurs 2212/920KV). Les paramètres physiques du Phantom 2 utilisés dans l'expérience de simulation sont indiqués dans le tableau 2.1.

Le problème (3.13)-(3.16) a été résolu numériquement pour trouver la trajectoire minimale et les entrées de commande $\omega_j(t)$ qui permet au Quadrotor de voler de la position initiale $[x, y, z]^T = [0, 0, 0]^T$ à l'instant $t_0 = 0$ au position finale $[x, y, z]^T = [4, 5, 6]^T$ à l'instant $t_f = 10s$, avec la condition initiale $[x_0, y_0, z_0, \phi_0, \theta_0, \psi_0]^T = [0, 0, 0, 0, 0, 0]^T$ et la condition finale $[x_f, y_f, z_f, \phi_f, \theta_f, \psi_f]^T = [4, 5, 6, 0, 0, 0]^T$. Les vitesses angulaires et linéaires initiales sont considérées nulles $[\dot{x}_0, \dot{y}_0, \dot{z}_0, \dot{\phi}_0, \dot{\theta}_0, \dot{\psi}_0]^T = [0, 0, 0, 0, 0, 0]^T$ et la même chose pour les vitesses angulaires et linéaires finales. En ce qui concerne les contraintes dans (12), l'estimation initiale pour les entrées de commande est donnée par $w_s = 912,32 \text{ rad/s}$ ce qui signifie que la poussée totale est $u_1 = 12,75 \text{ N}$ qui correspond à la poussée nécessaire pour contrebalancer l'accélération de gravité. La figure 3.6 montre l'évolution temporelle de la position du véhicule et de la vitesse linéaire. Alors que la figure 3.7 montre l'évolution temporelle de l'attitude du véhicule et de la vitesse angulaire. La figure 3.8 rapporte la trajectoire optimale dans l'espace $(x - y - z)$ et les entrées de commande $\omega_j(t)$. L'énergie consommée par le quadrotor pour parcourir cette trajectoire est $E_c(t_f) = 10.38 \text{ kJ}$ figure 3.9.

Le problème (3.17)-(3.20) a été résolu numériquement pour trouver les entrées de commande optimale en terme du temps $\omega_j(t)$ qui permet au Quadrotor de voler de la position initiale $[x, y, z]^T = [0, 0, 0]^T$ à l'instant $t_0 = 0$ au position finale $[x, y, z]^T = [4, 5, 6]^T$ avec une énergie totale fixe $E_c(t_f) = 3.5 \text{ kJ}$, et les mêmes conditions initiales et finales que le problème précédent $E_c(t_0) = 0 \text{ kJ}$.

Les résultats de simulation pour ce test sont présentés sur la figure 3.10, la figure

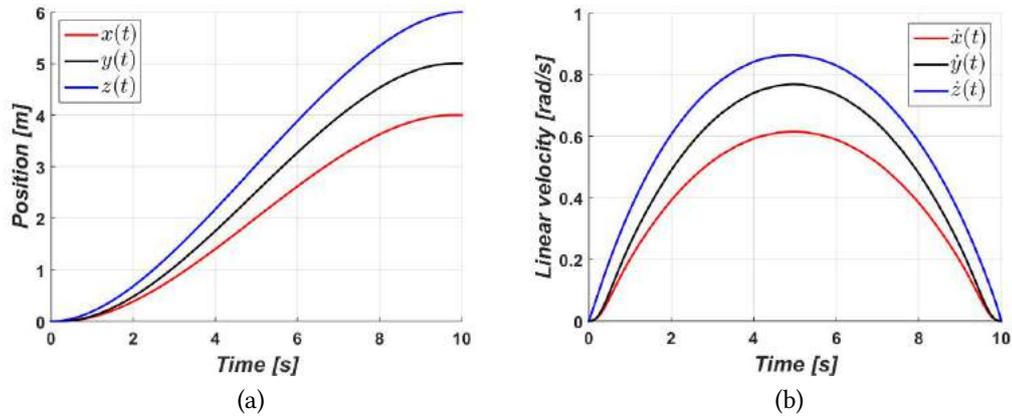


Figure 3.6: Position du Quadrotor (a) et Vitesse linéaire (b)

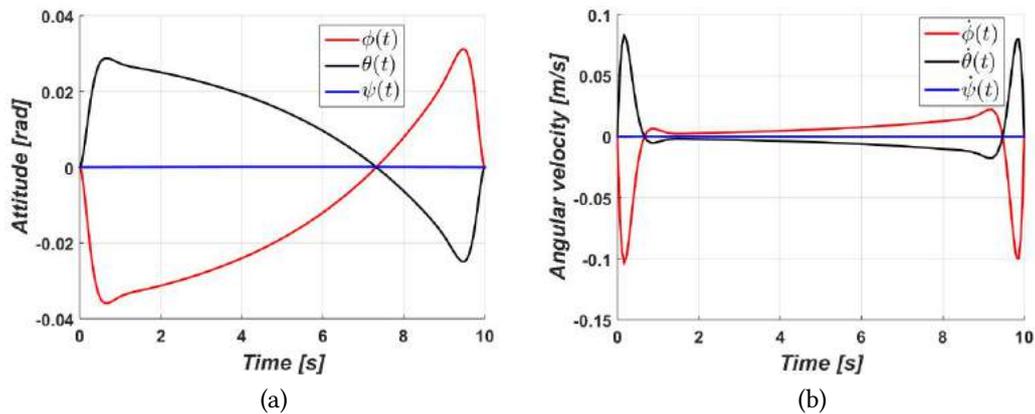


Figure 3.7: Attitude du Quadrotor (a) et Vitesse angulaire (b)

3.11 et la figure 3.12. Il est clair de la figure 3.12 que les entrées de commande $\omega_j(t)$ sont saturées et il est évident que les actionneurs sont plus sollicités pour permettre au véhicule de parcourir la trajectoire dans un temps minimal. Le temps optimal dépensé pour parcourir la trajectoire en respectant les contraintes dans (3.22) est de $t_f = 3.06s$.

3.7.2 Comparaison

Afin d'évaluer l'énergie consommée par le véhicule et avoir une idée sur l'énergie économisée avec l'approche proposée, nous avons comparé l'énergie consommée par le véhicule Quadrotor avec notre approche et l'énergie consommée en utilisant une approche de commande classique. L'approche de commande classique se compose de deux con-

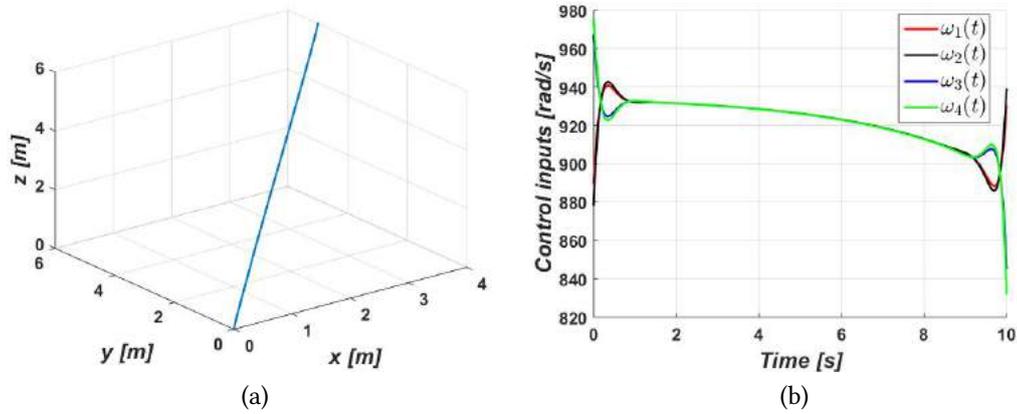


Figure 3.8: Trajectoire optimale dans l'espace $(x - y - z)$ (a), et Entrées de commande (b)

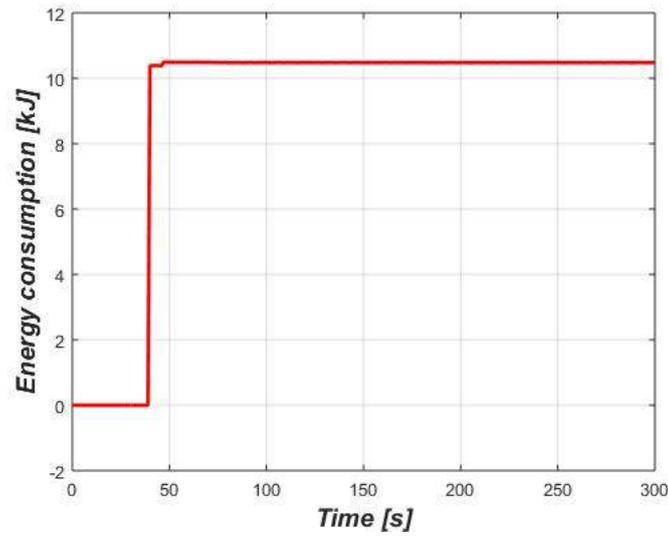


Figure 3.9: fonction coût $E_c(t)$

trôleurs, un contrôleur bas niveau et un algorithme de commande haut niveau.

Le contrôleur bas niveau consiste en une loi de commande adaptative floue directe [134] composé de trois termes, le terme adaptatif flou qui est synthétisé pour la dynamique du système $u_{f,j} = \Theta_j^T \varphi_j(x_s)$, avec $\varphi_j(x_s)$ le vecteur de fonctions de base floues, et Θ_j^T le vecteur des paramètres ajustables du système flou et x_s le vecteur d'état. Le terme robuste borné $u_{r,j} = \hat{\delta}_j \tanh\left(\frac{e_{2j}}{\epsilon_j}\right)$ employé pour compenser l'erreur d'approxima-

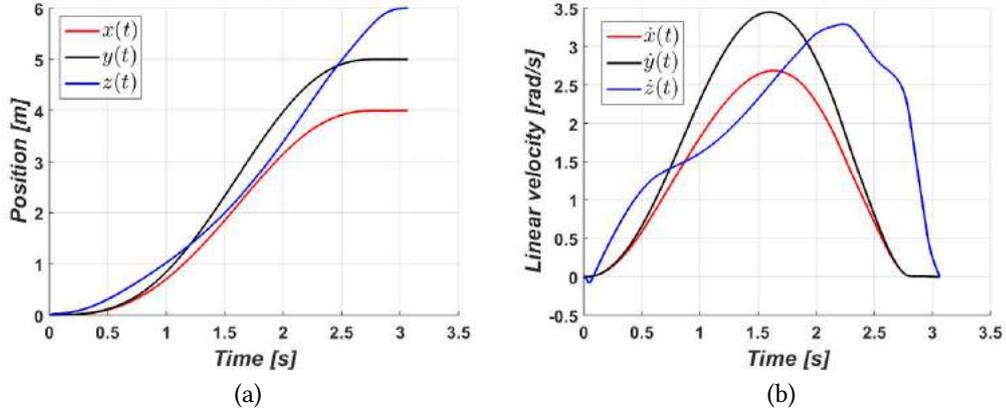


Figure 3.10: Position du Quadrotor (a) et Vitesse linéaire (b)

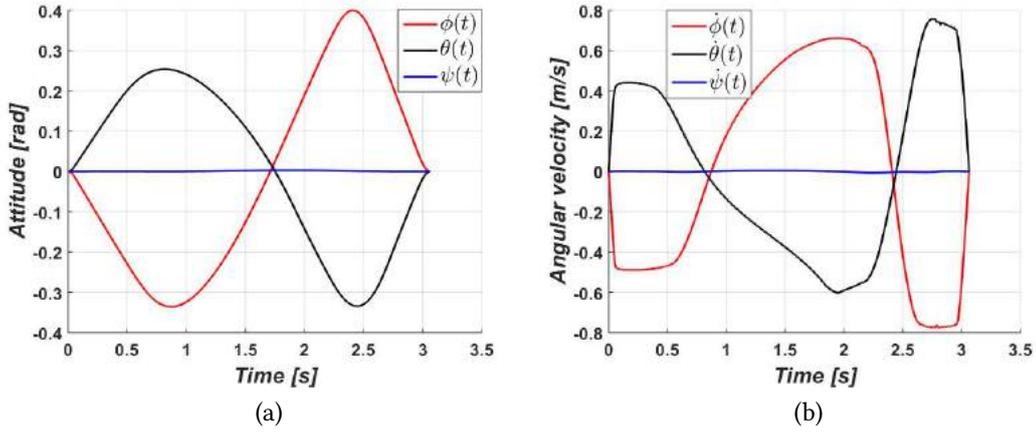


Figure 3.11: Attitude du Quadrotor (a) et Vitesse angulaire (b)

tion floue. Enfin, $u_{p,j} = k_{2j}e_{2j}$ le terme dérivé proportionnel.

$$u_j(t) = \Theta_j^T \varphi_j(x_s) + \hat{\delta}_j \tanh\left(\frac{e_{2j}}{\epsilon_j}\right) + k_{2j}e_{2j} \quad (3.21)$$

$$\dot{\Theta}_j = \gamma_j e_{2j} \varphi_j(x_s) - \sigma_{1j} \Theta_j \quad (3.22)$$

$$\dot{\hat{\delta}}_j = \eta_j e_{2j} \tanh\left(\frac{e_{2j}}{\epsilon_j}\right) - \sigma_{2j} \hat{\delta}_j \quad (3.23)$$

avec $e_{2j} = v_j - x_{2j}$ l'erreur en vitesse, $v_j = \dot{x}_{1j,d} + k_{1j}e_{1j}$ $k_{2j} > 0$, et $e_{1j} = x_{1j,d} - x_{1j}$, $j = 1, \dots, 6$ l'erreur en position et en attitude. $\epsilon_j > 0$, $\gamma_j > 0$, $\eta_j > 0$, $k_{1j} > 0$, σ_{1j} et σ_{2j} sont des constantes positives et $x_{1j,d}$ le signal de référence. Plus de détail sur cette approche de commande sera présenté dans le prochain chapitre.

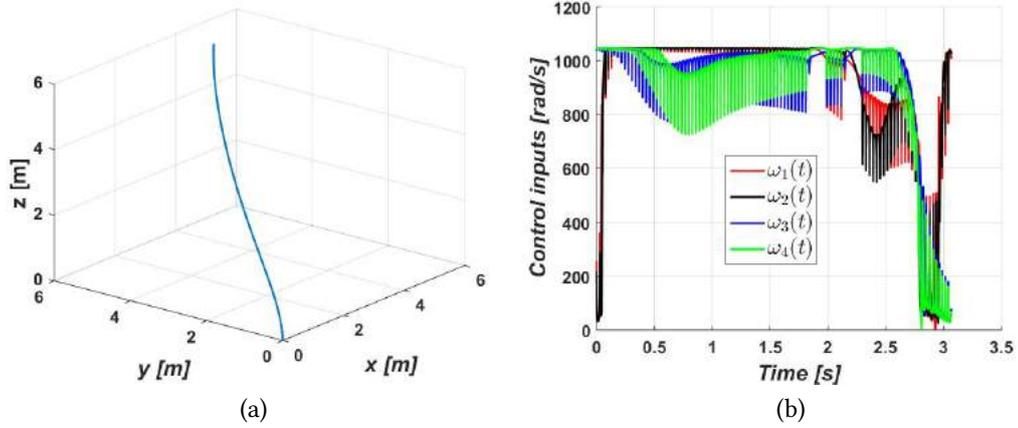


Figure 3.12: Trajectoire optimale dans l'espace $(x - y - z)$ (a), et Entrées de commande (b)

L'algorithme de commande haut niveau consiste à un polynôme du degré trois, $q(t) = \alpha_0 + \alpha_1 t + \alpha_2 t^2 + \alpha_3 t^3$, avec les conditions initiales et finales sur la position et la vitesse $q(t_0) = q_0$, $q(t_f) = q_f$, $\dot{q}(t_0) = \dot{q}_0$, $\dot{q}(t_f) = \dot{q}_f$ nous pouvons calculer les paramètres du polynôme.

L'énergie consommée obtenue avec l'approche de commande classique est $E_c(t_f) = 10.49 \text{ kJ}$, comparée à l'énergie consommée par l'approche proposée $E_c(t_f) = 10.38 \text{ kJ}$, nous observons une augmentation de 1% de l'énergie totale, ce qui représente une petite quantité d'énergie économisée. Pour améliorer la quantité d'énergie économisée, nous devons améliorer le modèle énergétique, ce qui sera l'objectif du travail futur.

Nous avons fait une étude comparative entre notre approche et l'approche proposée par [119] pour un temps final de $t_f = 20 \text{ s}$. L'énergie consommée obtenue avec l'approche proposée dans [119] est $E_{c1}(t_f) = 26.23 \text{ kJ}$, Alors que l'énergie consommée par notre approche pour les mêmes conditions aux limites est $E_{c2}(t_f) = 20,72 \text{ kJ}$. L'énergie économisée par l'approche proposée par rapport à celle proposée dans [119] est de 26.59% de l'énergie totale.

3.8 Conclusion

Dans ce chapitre, un algorithme de génération de trajectoire optimale en terme d'énergie et du temps a été proposé par le bais de deux problèmes de commande optimale présentés et résolus numériquement. Au début nous avons présenté un modèle énergétique composé de toutes les dynamiques qui influencent le comportement du véhicule. En-

suite, nous avons validé le modèle énergétique à travers deux missions différentes avec les mêmes configurations initiales et finales pour montrer la possibilité d'économiser de l'énergie. Dans un but d'optimisation énergétique, un problème de commande optimale a été introduit et résolu à l'aide d'un logiciel d'optimisation dynamique *GPOPS-II*. Dans le problème de commande optimale, nous cherchons à trouver la trajectoire et les entrées de commande du véhicule qui minimisent l'énergie consommée durant une mission spécifique. Les expériences numériques ont illustré les solutions du problème de commande optimale proposé, et l'étude comparative fournit une quantification de l'énergie qui peut être sauvegardée dans une mission simple.

En effet le nombre d'application émergé par les drones miniatures à voilure tournante précisément le Quadrotor ne cesse pas d'augmenter. Actuellement, les applications potentielles qui dominent le marché des drones miniatures nécessitent une grande autonomie (longue durée de vol) à savoir l'agriculture et la photographie. De ce fait le développement des algorithmes de génération de trajectoire et les approches de commande qui minimisent l'énergie du drone durant sa mission reste un problème ouvert.

Dans le prochain chapitre, nous allons introduire une commande adaptative floue directe à base d'un observateur d'état pour le Quadrotor.

Chapitre 4

Commande adaptative floue directe d'un Quadrotor avec observateur

4.1 Introduction

Deux approches de commande intelligentes seront introduites dans ce chapitre pour une classe de système multi-rotors à savoir le Quadrotor. Cette classe des robots volants est devenu récemment un domaine intéressant qui a motivé la communauté de chercheurs à cause de large gamme d'application couvrir par ce type de robot volant. En raison de sa grande maniabilité et de sa structure mécanique simple, le Quadrotor est devenu une plate-forme de recherche de référence. Cependant, la conception de commande pour ce type de véhicule est une étape compliquée car ce sont des systèmes mécaniques non linéaires, sous-actionnés et instable. La conception d'un contrôleur pour ces aéronefs nécessite quelques considérations importantes à prendre en compte. Ces types de robots volants ont un comportement non linéaire et variable dans le temps, des incertitudes paramétriques (perturbations internes) et ils sont affectés par les turbulences atmosphériques (perturbations externes). Par conséquent, une stratégie de commande adaptative robuste est nécessaire pour obtenir un suivi de trajectoire précis et des performances élevées dans un vol autonome.

Au cours de la dernière décennie, la technique de commande par le backstepping a été une solution intuitive au problème de commande des systèmes sous-actionnés les UAVs (Unmanned Aerial Vehicles). En asservissant le sous-système actionné (attitude) et en fournissant un contrôleur virtuel, on peut stabiliser le sous-système sous-actionné (position). Malheureusement, cette approche systématique nécessite une connaissance complète de la dynamique du système qui est une hypothèse extrêmement difficile à avoir en pratique pour les UAVs. Dans le but de résoudre ce problème et étendre l'applicabilité du backstepping, nous allons introduire dans ce chapitre une commande adaptative floue par le backstepping pour le Quadrotor sans et avec observateur d'état. L'idée de base est d'utiliser un système adaptatif flou pour estimer en ligne une loi de commande idéale partiellement inconnue basé sur le modèle dynamique. Dans l'approche proposée, le contrôleur est composé de trois termes, le terme adaptatif flou synthétisé pour approximer une loi de commande backstepping basée sur le modèle dynamique. Un terme robuste borné utilisé pour compenser l'erreur d'approximation floue. Le terme proportionnel dérivé pour la commande de la variation des états du système. La stabilité de la boucle fermée est étudiée par l'approche de Lyapunov. Les stratégies de commande adaptative floue proposée ici garantissent la bornitude de tous les signaux de la boucle fermée ainsi que la convergence des erreurs de poursuite vers un petit voisinage de zéro. Les performances de ces lois de commande sont examinées par simulation.

Ce chapitre est construit ainsi : nous commençons par la formulation de problème

de commande pour le système étudié à savoir le Quadrotor dans la section 4.2. Nous procédons ensuite dans la section 4.3 à la synthèse de la loi de commande adaptative floue directe, la conception de son mécanisme d'adaptation et nous dégagons aussi les conditions nécessaires pour la stabilité de la boucle fermée. La section 4.4 introduit la synthèse de la loi de commande adaptative floue directe avec observateur d'état. La section 4.5 est consacrée à la présentation et l'analyse des résultats de simulation issus de l'application des lois de commande adaptatives floues proposées dans ce chapitre sur le Quadrotor.

4.2 Formulation du problème

L'objectif principal de l'étape de synthèse de contrôle est de développer une approche de commande adaptative floue basée sur l'observateur capable de générer les signaux d'entrée pour les moteurs de Quadrotor en assurant que les sorties $(x(t), y(t), z(t), \psi(t))$ suivent la trajectoire de référence $(x_r(t), y_r(t), z_r(t), \psi_r(t))$ en garantissant la bornitude de tous les signaux de la boucle fermée.

Le modèle dynamique du Quadrotor représenté sous forme d'état (2.20), peut être réorganisé sous forme de $n = 6$ sous système $\sum_i, i = 1, \dots, n$ comme suit

$$\begin{cases} \dot{x}_{1i} = x_{2i} \\ \dot{x}_{2i} = f_i(x_s) + g_i(x_s)u_i(t) \\ y_i = x_{1i} \end{cases} \quad (4.1)$$

avec: $x_s = [x_{11}, x_{21}, \dots, x_{1n}, x_{2n}]^T \in \mathbb{R}^{2n}$, est le vecteur d'état complet, $u = [u_1, \dots, u_n]^T \in \mathbb{R}^n$ est le vecteur d'entrée de commande $y = [y_1, \dots, y_n]^T \in \mathbb{R}^n$ est le vecteur de sortie, $f_i(x_s)$ et $g_i(x_s)$ sont des fonctions non linéaires continues supposées inconnues. Afin d'atteindre l'objectif de synthèse de la loi de commande, les hypothèses suivantes sont retenues dans le reste de ce chapitre.

Hypothèse 4.1. Les trajectoires de références $x_{1i,r}, i = 1, \dots, n$ ainsi que leurs dérivées $\dot{x}_{1i,r}$, et $\ddot{x}_{1i,r}$ sont supposées connues, continues et bornées.

Hypothèse 4.2. les angles de roulis, tangage et lacet sont supposées bornées $(-\pi \leq \psi \leq \pi)$, $(-\frac{\pi}{2} \leq \theta \leq \frac{\pi}{2})$ et $(-\frac{\pi}{2} \leq \phi \leq \frac{\pi}{2})$. Cette hypothèse est courante dans la pratique tant que le Quadrotor n'effectue pas de manœuvres agressives dans un vol libre.

Hypothèse 4.3. Les fonctions $g_i(x_s)$ ainsi que ses dérivées $\dot{g}_i(x_s)$, $i = 1, \dots, n$ sont définies positives et lipschitziennes.

Définissons l'erreur de poursuite pour le signal de référence $x_{1i,r}$, $i = 1, \dots, n$ comme suit

$$e_{1i} = x_{1i,r} - x_{1i} \quad (4.2)$$

et la deuxième erreur

$$e_{2i} = v_i - x_{2i} \quad (4.3)$$

avec: $v_i = \dot{x}_{1i,r} + \kappa_{1i}e_{1i}$.

L'erreur en vitesse dans (4.3) peut être réécrite comme suit $e_{2i} = \dot{e}_{1i} + \kappa_{1i}e_{1i}$, si $e_{2i} = 0$ nous aurons un système linéaire du premier ordre dont la solution implique la convergence de e_{1i} , \dot{e}_{1i} vers zéro [135]. Il est possible donc de stabiliser le vecteur $[e_{1i}, \dot{e}_{1i}]^T$ en synthétisant une loi de commande garantissant la convergence vers zéro du signal e_{2i} .

Proposition 4.1. Considérons le système (4.1) affine en la commande et les erreurs e_{2i} , $i = 1, \dots, n$ données par (4.3). En utilisant les techniques de commande par le backstepping, une loi de commande peut être synthétisée $u_i^*(t)$, garantissant la convergence vers zéro de e_{2i} , $i = 1, \dots, n$, quand $t \rightarrow \infty$, et par conséquent la convergence vers zéro de e_{1i} et sa dérivée quand $t \rightarrow \infty$.

Prouve. On procède en deux étapes:

Étape 1. A partir de la première équation différentielle du système (4.1), la dynamique de l'erreur de poursuite est donnée par $\dot{e}_{1i} = \dot{x}_{1i,r} - x_{2i}$. Maintenant, choisissant une fonction de Lyapunov V_{1i} , et sa dérivée temporelle \dot{V}_{1i} comme suit

$$V_{1i} = \frac{1}{2}e_{1i}^2 \quad (4.4)$$

$$\dot{V}_{1i} = e_{1i}\dot{e}_{1i} = e_{1i}(\dot{x}_{1i,r} - x_{2i}) \quad (4.5)$$

nous proposons la commande virtuelle v_i , donnée par

$$v_i = \dot{x}_{1i,r} + \kappa_{1i}e_{1i} \quad (4.6)$$

avec κ_{1i} est un paramètre de conception. En utilisant (4.6), la dérivée temporelle de V_{1i} (4.5) peut être reformulée comme suit

$$\dot{V}_{1i} = -\kappa_{1i}e_{1i}^2 + e_{1i}e_{2i} \quad (4.7)$$

Étape 2. par la dérivée de e_{2i} et l'utilisation de (4.1) nous aurons

$$\dot{e}_{2i} = \dot{v}_i - f_i(x_s) - g_i(x_s)u_i(t) \quad (4.8)$$

Choisissant la fonction de Lyapunov augmentée $V_{2i} = V_{1i} + \frac{1}{2g_i(x_s)}e_{2i}^2$; alors, la dérivée temporelle de V_{2i} sera comme suit

$$\dot{V}_{2i} = \dot{V}_{1i} + \frac{1}{g_i(x_s)}e_{2i}\dot{e}_{2i} - \frac{\dot{g}_i(x_s)}{2g_i^2(x_s)}e_{2i}^2 \quad (4.9)$$

En remplaçant (4.7), (4.8) et (4.9), on obtient

$$\dot{V}_{2i} = -\kappa_{1i}e_{1i}^2 + e_{2i} \left[\frac{\dot{v}_i + g_i(x_s)e_{1i} - f_i(x_s)}{g_i(x_s)} - \frac{\dot{g}_i(x_s)}{2g_i^2(x_s)}e_{2i} - u_i(t) \right] \quad (4.10)$$

La loi de commande $u_i^*(t)$ est conçue comme suit

$$u_i^*(t) = \frac{1}{g_i(x_s)} \left[\dot{v}_i + g_i(x_s)e_{1i} - f_i(x_s) - \frac{\dot{g}_i(x_s)}{2g_i(x_s)}e_{2i} \right] + \kappa_{2i}e_{2i} \quad (4.11)$$

avec $\dot{v}_i = \ddot{x}_{1i,r} + \kappa_{1i}(x_{1i,r} - \dot{x}_{1i})$. En utilisant (4.11), on peut facilement vérifier que

$$\dot{V}_{2i} < -\kappa_{1i}e_{1i}^2 - \kappa_{2i}e_{2i}^2 \quad (4.12)$$

Étant donné que $\dot{V}_{2i}(t)$ est définie négative, alors $V_{2i}(t) \leq V_{2i}(0)$. Ce qui implique que e_{1i} et e_{2i} convergence vers zéro quand $t \rightarrow \infty$, par conséquent l'erreur de poursuite e_{1i} et sa première dérivée convergent aussi vers zéro.

La loi de commande $u_{b,i}^*(t)$ est idéale dans le cas où les fonctions non linéaires $f_i(x_s)$ et $g_i(x_s)$ sont inconnues. Dans ce qui suit nous cherchons à approximer cette loi de commande par un système adaptatif floue robuste.

Remarque 4.1. Le terme proportionnel $\kappa_{2i}e_{2i}$ dans la loi de commande (4.11) n'est pas approximé avec la commande idéale $u_i^*(t)$ par le système adaptatif flou, nous utilisons ce terme comme une action proportionnelle dans la loi de commande proposée pour améliorer les performances fournies par le système adaptatif flou.

4.3 Synthèse de la commande adaptative floue directe

Nous avons prouvé dans la section précédente l'existence d'une commande idéale $u_i^*(t)$ basée sur le modèle dynamique du système décrite par (4.11), celle-ci peut stabiliser le

système et assurer les performances désirées. Néanmoins, cette commande dépend des dynamiques partiellement inconnues du système, du coup elle ne peut être implémentée. Pour surmonter cette limitation, un mécanisme adaptative flou est utilisé pour estimer cette commande u^* avec l'aide d'un terme robuste pour compenser les erreurs d'approximation floue.

4.3.1 Description de système d'inférence flou

Le système d'inférence flou utilisé dans ce travail est un système flou de type Takagi-Sugeno d'ordre zéro, multi entrées mono sortie. Un tel système se compose de quatre parties: la base de connaissances (où de règles), la fuzzification, le moteur d'inférence floue et la défuzzification. La base de connaissances se compose d'une collection de règles floues *if-then* sous la forme suivante [136]

$$R^k : \mathbf{Si} \ x_1 \text{ is } F_1^k \text{ and } \dots \ x_n \text{ is } F_n^k, \ \mathbf{Alors} \ y_f \text{ is } y_f^k, \ k = 1, \dots, N$$

avec: $x_f = [x_1, \dots, x_n]^T \in \Omega_{x_f} \subset \mathbb{R}^n$ est l'ensemble des entrées, $y_f \in \mathbb{R}$ la sortie avec $\Omega_{x_f} = \Omega_{x_1} \times \dots \times \Omega_{x_n}$ et $\Omega_{x_1} \in \mathbb{R}$. Pour chaque entrée x_i on définit m_i ensemble flou $G_i^j, j = 1, \dots, m_i$ et $F_i^k \in \{G_i^1, \dots, G_i^{m_i}\}$, y_f^k est une constante réelle associée à la partie conclusion de la k -th règle et N le nombre total des règles.

En utilisant la défuzzification de type centroïde, la sortie du système flou est exprimée comme suit

$$y_f(x_f | \Theta) = \frac{\sum_{k=1}^N y_f^k \prod_{i=1}^n \mu_{F_i^k}(x_f)}{\sum_{k=1}^N [\prod_{l=1}^n \mu_{F_l^k}(x_f)]} \quad (4.13)$$

où: $\mu_{F_i^k} \in \{\mu_{G_i^1}, \dots, \mu_{G_i^{m_i}}\}$ et $\mu_{G_i^j}(x_i)$ est la fonction d'appartenance associée à l'ensemble flou G_i^j . Définissons

$$\varphi_k(x_f) = \frac{\prod_{i=1}^n \mu_{F_i^k}(x_f)}{\sum_{k=1}^N [\prod_{i=1}^n \mu_{F_i^k}(x_f)]} \quad (4.14)$$

avec $\varphi(x_f) = [\varphi_1(x_f), \dots, \varphi_N(x_f)]^T$ est le vecteur des fonctions floues de base, et $\Theta^T = [y_f^1, \dots, y_f^N]$ est le vecteur des paramètres de la partie conclusion des règles floues. Ensuite, la sortie du système flou (4.13) peut être réécrite comme suit

$$y_f(x_f | \Theta) = \Theta^T \varphi(x_f) \quad (4.15)$$

Lemme 4.1. [137] Pour toute fonction continue réelle $f(x_f)$ définie sur l'ensemble com-

subset $\Omega_f \subset \mathbb{R}^n$ et pour $\delta > 0$, il existe un système flou tel que

$$\sup_{x_f \in \Omega_f} |f(x_f) - \Theta^T \varphi(x_f)| \leq \delta$$

Remarque 4.2. Le système flou (4.15) est supposé être bien défini et ses paramètres convenablement choisis de telle sorte que $\sum_{k=1}^N \prod_{i=1}^n \mu_{F_i^k}(x_f) \neq 0$ pour tout $x_f \in \Omega_{x_f}$.

4.3.2 Structure de la loi de commande

Dans cette section, nous allons concevoir une commande adaptative floue robuste, ainsi que les lois d'adaptation des paramètres de telle sorte que tous les signaux de la boucle fermée soient uniformément stables et bornés et que les erreurs de poursuite $e_{1i} = x_{1i,r} - x_{1i}$ soient aussi petites que possible. La loi de commande proposée est constituée de trois termes, un terme adaptatif flou employé pour approcher les dynamiques inconnues du système, un terme robuste conçu pour compenser les erreurs d'approximation floue, et un terme proportionnel dérivé.

Hypothèse 4.4. En utilisant le lemme 4.1 et la démonstration donnée par [137], les systèmes adaptatifs flous sont des approximateurs universels, ils peuvent approximer toute fonction lisse sur un ensemble compact. En se basant sur la capacité d'approximation des systèmes flous, on peut supposer que les lois de commande $u_i^*(t)$ peuvent être approximées par le système adaptatif flou $u_f(x_s | \Theta_i)$ comme suit

$$u_f(x_s | \Theta_i) = \Theta_i^T \varphi_i(x_s), i = 1, \dots, n \quad (4.16)$$

Nous supposons, qu'il existe un vecteur des paramètres Θ_i^* lentement variables dans le temps tel que la commande idéale $u_i^*(t)$ vérifie

$$u_i^*(t) = \Theta_i^{*T} \varphi_i(x_s) + \varepsilon_i(x_s), i = 1, \dots, n \quad (4.17)$$

avec $\varepsilon_i(x_s)$ l'erreur d'approximation floue minimale, Θ_i^* est le vecteur des paramètres optimaux défini par [136]:

$$\Theta_i^* = \arg \min_{\Theta_i \in \Omega_\theta} \left\{ \sup_{x_s \in \Omega_{x_s}} |u_f(x_s | \Theta_i) - u_i^*(t)| \right\} \quad (4.18)$$

avec: Ω_{x_s} et Ω_θ sont deux ensembles compact pour x_s et Θ_i respectivement. Donc, nous

pouvons écrire

$$\varepsilon_i(x_s) = u_i^*(t) - \Theta_i^{*T} \varphi_i(x_s) \quad (4.19)$$

Hypothèse 4.5. Les erreurs d'approximation floue $\varepsilon_i(x_s)$ sont supposée bornées tel que $|\varepsilon_i(x_s)| \leq \bar{\varepsilon}_i$ pour tout $x_s \in \Omega_{x_s}$ et $\bar{\varepsilon}_i > 0$ est une constante.

Puisque le vecteur optimal Θ_i^* est inconnu, nous allons l'estimer par une loi d'adaptation. Soit Θ_i une estimation du vecteur optimal Θ_i^* ; la loi de commande pour le système (4.15) est choisie comme suit

$$u_i(t) = u_{f,i} + u_{r,i} + u_{p,i}, i = 1, \dots, n \quad (4.20)$$

$u_{f,i}$ est le terme de commande adaptatif flou, conçu pour approcher la loi de commande idéale basée sur le modèle dynamique $u_i^*(t)$ (4.11), avec $\varphi_i(x_s)$ est le vecteur des fonction floues de base, Θ_i est le vecteur des paramètres réglables du système flou.

$$u_{f,i} = \Theta_i^T \varphi_i(x_s) \quad (4.21)$$

$$\dot{\Theta}_i = \gamma_i e_{2i} \varphi_i(x_s) - \sigma_{1i} \Theta_i \quad (4.22)$$

avec $\gamma_i > 0$ et $\sigma_{1i} > 0$ sont des constantes positives, $e_{2i} = \dot{e}_{1i} + \kappa_i e_{1i}$ et $\Theta_i(0) = 0$.

$u_{r,i}$ est le terme de commande robuste borné employé pour compenser l'erreur d'approximation floue donné par

$$u_{r,i} = \hat{\varepsilon}_i \tanh\left(\frac{e_{2i}}{\epsilon_i}\right) \quad (4.23)$$

avec

$$\dot{\hat{\varepsilon}}_i = \eta_i e_{2i} \tanh\left(\frac{e_{2i}}{\epsilon_i}\right) - \sigma_{2i} \hat{\varepsilon}_i \quad (4.24)$$

avec $\eta_i > 0$, $\sigma_{2i} > 0$ et $\epsilon_i > 0$ sont des constantes positives, et $\hat{\varepsilon}_i(0) = 0$.

$u_{p,i}$ est le terme proportionnel dérivé, il est donné par

$$u_{p,i} = \kappa_{2i} e_{2i} \quad (4.25)$$

avec $\kappa_{2i} > 0$ une constante positive.

La loi de commande proposé peut être formulée explicitement comme suit

$$u_i = \Theta_i^T \varphi_i(x_s) + \hat{\varepsilon}_i \tanh\left(\frac{e_{2i}}{\epsilon_i}\right) + \kappa_{2i} e_{2i} \quad (4.26)$$

$$\dot{\Theta}_i = \gamma_i e_{2i} \varphi_i(x_s) - \sigma_{1i} \Theta_i \quad (4.27)$$

$$\dot{\hat{\epsilon}}_i = \eta_i e_{2i} \tanh\left(\frac{e_{2i}}{\epsilon_i}\right) - \sigma_{2i} \hat{\epsilon}_i \quad (4.28)$$

Il est important de noter que les lois d'adaptations (4.27) et (4.28) sont conçues de telle sorte que la dérivée de la fonction de Lyapunov candidate, utilisée dans la section suivante pour analyser la stabilité en boucle fermée, soit définie négative dans l'espace des paramètres estimés lorsque ces paramètres dépassent certaines bornes [138].

4.3.3 Analyse de stabilité

Considérons la fonction de Lyapunov candidate suivante

$$V = \frac{1}{2} \sum_{i=1}^n e_{1i}^2 + \frac{1}{g_i(x_s)} e_{2i}^2 + \frac{1}{\gamma_i} \tilde{\Theta}_i^T \tilde{\Theta}_i + \frac{1}{\eta_i} \tilde{\epsilon}_i^T \tilde{\epsilon}_i \quad (4.29)$$

avec $\tilde{\epsilon}_i = \epsilon_i - \hat{\epsilon}_i$, $\hat{\epsilon}_i$ est l'estimé de ϵ_i et $\tilde{\Theta}_i = \Theta_i^* - \Theta_i$ est l'erreur d'estimation.

La dérivée temporelle de V on utilisant (4.10) est

$$\dot{V} = \sum_{i=1}^n -\kappa_{1i} e_{1i}^2 + e_{2i} [u_i^*(t) - u_i(t)] + \frac{1}{\gamma_i} \tilde{\Theta}_i^T \dot{\tilde{\Theta}}_i + \frac{1}{\eta_i} \tilde{\epsilon}_i^T \dot{\tilde{\epsilon}}_i \quad (4.30)$$

En remplaçant par (4.17) et (4.26) dans (4.30), on obtient

$$\begin{aligned} \dot{V} = & \sum_{i=1}^n -\kappa_{1i} e_{1i}^2 + e_{2i} \left[\Theta_i^{*T} \varphi_i(x_s) + \epsilon_i(x_s) - \Theta_i^T \varphi_i(x_s) - \hat{\epsilon}_i \tanh\left(\frac{e_{2i}}{\epsilon_i}\right) - \kappa_{2i} e_{2i} \right] \\ & + \frac{1}{\gamma_i} \tilde{\Theta}_i^T \dot{\tilde{\Theta}}_i + \frac{1}{\eta_i} \tilde{\epsilon}_i^T \dot{\tilde{\epsilon}}_i \end{aligned} \quad (4.31)$$

Les vecteurs des paramètres optimaux Θ_i^* et ϵ_i sont supposés lentement variables dans le temps, par conséquent, la dérivée de l'erreur d'estimation sera $\dot{\tilde{\Theta}}_i = -\dot{\Theta}_i$, et $\dot{\tilde{\epsilon}}_i = -\dot{\hat{\epsilon}}_i$. On a donc,

$$\begin{aligned} \dot{V} = & \sum_{i=1}^n -\kappa_{1i} e_{1i}^2 - \kappa_{2i} e_{2i}^2 + e_{2i} \tilde{\Theta}_i^T \varphi_i(x_s) + e_{2i} \left[\epsilon_i(x_s) - \hat{\epsilon}_i \tanh\left(\frac{e_{2i}}{\epsilon_i}\right) \right] \\ & - \frac{1}{\gamma_i} \tilde{\Theta}_i^T \dot{\Theta}_i - \frac{1}{\eta_i} \tilde{\epsilon}_i^T \dot{\hat{\epsilon}}_i \end{aligned} \quad (4.32)$$

L'équation (4.32) peut être bornée de la manière suivante

$$\begin{aligned}
\dot{V} \leq & \sum_{i=1}^n -\kappa_{1i}e_{1i}^2 - \kappa_{2i}e_{2i}^2 + \frac{1}{\gamma_i}\tilde{\Theta}_i^T \left[\gamma_i e_{2i} \varphi_i(x_s) - \dot{\Theta}_i \right] \\
& + |e_{2i}| \varepsilon_i - e_{2i} \hat{\varepsilon}_i \tanh\left(\frac{e_{2i}}{\varepsilon_i}\right) + \frac{1}{\eta_i} \tilde{\varepsilon}_i \left[\eta_i e_{2i} \tanh\left(\frac{e_{2i}}{\varepsilon_i}\right) - \dot{\hat{\varepsilon}}_i \right] \\
& - e_{2i} \varepsilon_i \tanh\left(\frac{e_{2i}}{\varepsilon_i}\right) + e_{2i} \hat{\varepsilon}_i \tanh\left(\frac{e_{2i}}{\varepsilon_i}\right)
\end{aligned} \tag{4.33}$$

Notons que l'inégalité suivante est valable pour tout $\zeta > 0$

$$|e_{2i}| - e_{2i} \tanh\left(\frac{e_{2i}}{\varepsilon_i}\right) \leq \zeta \varepsilon_i = \varsigma \tag{4.34}$$

avec ζ une constante qui satisfait $\zeta = e^{-(\zeta+1)}$, à savoir $\zeta = 0.2785$. Ensuite, l'équation (4.33) peut être réécrite comme suit

$$\begin{aligned}
\dot{V} \leq & \sum_{i=1}^n -\kappa_{1i}e_{1i}^2 - \kappa_{2i}e_{2i}^2 + \frac{1}{\gamma_i}\tilde{\Theta}_i^T \left[\gamma_i e_{2i} \varphi_i(x_s) - \dot{\Theta}_i \right] \\
& + \varepsilon_i \varsigma + \frac{1}{\eta_i} \tilde{\varepsilon}_i \left[\eta_i e_{2i} \tanh\left(\frac{e_{2i}}{\varepsilon_i}\right) - \dot{\hat{\varepsilon}}_i \right]
\end{aligned} \tag{4.35}$$

Remplaçons les lois d'adaptation paramétriques $\dot{\Theta}_i$ (4.22), et $\dot{\hat{\varepsilon}}_i$ (4.24), on obtient

$$\dot{V} \leq \sum_{i=1}^n -\kappa_{1i}e_{1i}^2 - \kappa_{2i}e_{2i}^2 + \varepsilon_i \varsigma + \frac{\sigma_{1i}}{\gamma_i} \tilde{\Theta}_i^T \Theta_i + \frac{\sigma_{2i}}{\eta_i} \tilde{\varepsilon}_i \hat{\varepsilon}_i \tag{4.36}$$

En utilisant l'inégalité de Young pour les termes $\frac{\sigma_{1i}}{\gamma_i} \tilde{\Theta}_i^T \Theta_i$ et $\frac{\sigma_{2i}}{\eta_i} \tilde{\varepsilon}_i \hat{\varepsilon}_i$, on obtient

$$\frac{\sigma_{1i}}{\gamma_i} \tilde{\Theta}_i^T \Theta_i \leq -\frac{\sigma_{1i}}{2\gamma_i} \tilde{\Theta}_i^T \tilde{\Theta}_i + \frac{\sigma_{1i}}{2\gamma_i} \Theta_i^{*T} \Theta_i^* \tag{4.37}$$

$$\frac{\sigma_{2i}}{\eta_i} \tilde{\varepsilon}_i \hat{\varepsilon}_i \leq -\frac{\sigma_{2i}}{2\eta_i} \tilde{\varepsilon}_i^2 + \frac{\sigma_{2i}}{2\eta_i} |\varepsilon_i|^2 \tag{4.38}$$

Par conséquent, l'équation (4.36) devient

$$\dot{V} \leq \sum_{i=1}^n -\kappa_{1i}e_{1i}^2 - \kappa_{2i}e_{2i}^2 + \bar{\varepsilon}_i \varsigma - \frac{\sigma_{1i}}{2\gamma_i} \tilde{\Theta}_i^T \tilde{\Theta}_i + \frac{\sigma_{1i}}{2\gamma_i} \Theta_i^{*T} \Theta_i^* - \frac{\sigma_{2i}}{2\eta_i} \tilde{\varepsilon}_i^2 + \frac{\sigma_{2i}}{2\eta_i} |\varepsilon_i|^2 \tag{4.39}$$

Définissant $0 < \lambda < \min[\sigma_{1i}, \sigma_{2i}, 2\kappa_{1i}, 2\kappa_{2i}]$, (4.39) s'écrit

$$\dot{V} \leq -\lambda V + \bar{\varepsilon}_i \varsigma + \frac{\sigma_{1i}}{2\gamma_i} \Theta_i^{*T} \Theta_i^* + \frac{\sigma_{2i}}{2\eta_i} \bar{\varepsilon}_i^2 \tag{4.40}$$

Comme les vecteurs des paramètres Θ_i^* et ε_i sont supposés bornés, nous pouvons donc définir une borne constante et positive ρ , tel que

$$\rho = \sup_t \left(\sum_{i=1}^n \bar{\varepsilon}_i \varsigma + \frac{\sigma_{1i}}{2\gamma_i} \Theta_i^{*T} \Theta_i^* + \frac{\sigma_{2i}}{2\eta_i} \bar{\varepsilon}_i^2 \right) \quad (4.41)$$

avec ce choix, (4.40) devient

$$\dot{V} \leq -\lambda V + \rho \quad (4.42)$$

Avec le théorème suivant nous pouvons prouver la bornitude de tous les signaux de la boucle fermée ainsi que la convergence de l'erreur de poursuite e_{1i} vers un petit voisinage de zéro.

Théorème 4.1. *Considérons le système non linéaire du Quadrotor (4.1). Supposons que les hypothèses 4.1-4.5 et le lemme 4.1 sont satisfaites. La loi de commande adaptative floue (4.26) avec les lois d'adaptation (4.22) et (4.24) garantit la stabilité de la boucle fermée et la bornitude de tous les signaux ainsi que la convergence de l'erreur de poursuite vers un petit voisinage autour de zéro.*

Prouve. Par l'intégration de (4.42) sur $[0, t]$, on obtient

$$V(t) \leq V(0)e^{-\lambda t} + \frac{\rho}{\lambda} \quad (4.43)$$

L'inégalité (4.43), implique que pour $V \geq \frac{\rho}{\lambda}$ nous avons $\dot{V} \leq 0$. Par conséquent, en utilisant le théorème de Lyapunov, les signaux $e_{1i}(t)$, $e_{2i}(t)$, $\tilde{\Theta}_i(t)$, $\tilde{\varepsilon}_i$ et $u_i(t)$ de la boucle fermée sont bornés; Par ailleurs, pour toute $\mu \geq \sqrt{\frac{\rho}{\lambda}}$, il existe un temps $T > 0$, tel que $|e_{2i}(t)| \leq \mu$ pour tout $t \geq T$. Afin d'atteindre la convergence d'erreur de poursuite vers un petit voisinage autour de zéro et de rendre $\sqrt{\frac{\rho}{\lambda}}$ aussi petit que désiré, les paramètres de conception γ_i , η_i , ε_i , σ_{1i} , σ_{2i} , κ_{1i} et κ_{2i} devrait être choisi de manière appropriée. Ainsi, il est facile de voir que $\lim_{t \rightarrow \infty} |e_{1i}(t)| \leq \mu$. \square

Il est important de noter que la loi de commande (4.26) peut être implémentée lorsque le vecteur d'état du système $x_s(t)$ est mesurable. La connaissance du modèle dynamique n'est pas nécessaire pour la conception de cette commande, de plus avec un choix approprié des paramètres γ_i , η_i , κ_{1i} et κ_{2i} nous pouvons assurer ainsi la poursuite de trajectoires de références. Dans ce qui suit, nous allons étendre notre étude au cas où le vecteur d'état $x_s(t)$ est reconstruit par un observateur d'état synthétisé pour ce but. Par conséquent, la loi de commande adaptative floue directe à base d'observateur résultante requiert uniquement la mesure de la sortie $y_i(t)$ pour le calcul de la commande $u_i(t)$.

4.4 Commande adaptative floue directe avec observateur

Dans la section précédente, nous avons montré que la commande adaptative floue (4.26) doter d'une certaine robustesse peut garantir la stabilité du système en boucle fermée ainsi que la poursuite de trajectoire de référence; cependant le système flou utilisé dans cette approche dépend du vecteur d'état. Par conséquent lorsque une partie du vecteur d'état ou le vecteur d'état complet est indisponible, l'implémentation de la loi de commande (4.26) sera difficile. Maintenant, après avoir vu le principe de la synthèse de la commande adaptative floue, nous allons nous intéresser au cas où un observateur d'état est utilisé afin d'estimer convenablement le vecteur d'état $x_s(t)$.

4.4.1 Synthèse de l'observateur d'état

Définissons un observateur d'état pour le système (4.1) de la forme suivante

$$\begin{cases} \dot{\hat{x}}_{1i} = \hat{x}_{2i} + k_{1i}(y_i - \hat{x}_{1i}) \\ \dot{\hat{x}}_{2i} = f_i(\hat{x}_s) + g_i(\hat{x}_s)u_i(t) + k_{2i}(y_i - \hat{x}_{1i}) \\ \hat{y}_i = \hat{x}_{1i} \end{cases} \quad (4.44)$$

Ce qui peut être réécrit encore comme suit

$$\begin{cases} \dot{\hat{x}}_i = A_i \hat{x}_i + K_i y_i + B_i + G_i u_i(t) \\ \hat{y}_i = C^T \hat{x}_i \end{cases} \quad (4.45)$$

où

$$A_i = \begin{bmatrix} -k_{1i} & 1 \\ -k_{2i} & 0 \end{bmatrix}, \quad K_i = \begin{bmatrix} k_{1i} \\ k_{2i} \end{bmatrix}, \quad \hat{x}_i = \begin{bmatrix} \hat{x}_{1i} \\ \hat{x}_{2i} \end{bmatrix}$$

$$B_i = \begin{bmatrix} 0 \\ f_i(\hat{x}_s) \end{bmatrix}, \quad G_i = \begin{bmatrix} 0 \\ g_i(\hat{x}_s) \end{bmatrix}, \quad C_i = \begin{bmatrix} 1 \\ 0 \end{bmatrix}$$

Les vecteurs K_i sont choisies de tel sorte que A_i soit une matrice Hurwitz. Notons $Q_i^T = Q_i > 0$, il existe des matrices $P_i^T = P_i > 0$ symétriques et définies positives solutions du système d'équations suivant

$$A_i^T P_i + P_i A_i = -2Q_i \quad (4.46)$$

Nous définissons l'erreur d'observation $e_i = x_i - \hat{x}_i$; donc à partir de (4.1) et (4.44), la

dynamique de l'erreur d'observation s'écrit comme suit

$$\dot{e}_i = A_i e_i + \delta_i \quad (4.47)$$

avec $x_i = [x_{1i}, x_{2i}]^T$, $\delta_i = [0, \delta_{fi} + \delta_{gi}]^T$, $\delta_{fi} = f_i(x_s) - f_i(\hat{x}_s)$ et $\delta_{gi} = g_i(x_s) - g_i(\hat{x}_s)$.

Hypothèse 4.6. Les erreurs d'estimation non linéaires δ_{fi} , δ_{gi} sont lipschitziennes et satisfait $|\delta_{fi}(x_1) - \delta_{fi}(x_2)| \leq \bar{\delta}_{fi} |x_1 - x_2|$, $|\delta_{gi}(x_1) - \delta_{gi}(x_2)| \leq \bar{\delta}_{gi} |x_1 - x_2|$, avec $\bar{\delta}_{fi}$ et $\bar{\delta}_{gi}$ sont deux constantes positives.

Nous adoptons dans ce cas la loi de commande suivante

$$u_i(t) = \Theta_i^T \varphi_i(\hat{x}_s) + \hat{\varepsilon}_i \tanh\left(\frac{\hat{e}_{2i}}{\epsilon_i}\right) + \kappa_{2i} \hat{e}_{2i} \quad (4.48)$$

$$\dot{\Theta}_i = \gamma_i \hat{e}_{2i} \varphi_i(\hat{x}_s) - \sigma_{1i} \Theta_i \quad (4.49)$$

$$\dot{\hat{\varepsilon}}_i = \eta_i \hat{e}_{2i} \tanh\left(\frac{\hat{e}_{2i}}{\epsilon_i}\right) - \sigma_{2i} \hat{\varepsilon}_i \quad (4.50)$$

avec $\hat{e}_{2i} = \dot{\hat{e}}_{1i} + \kappa_i \hat{e}_{1i}$ et $\hat{e}_{1i} = x_{1i,r} - \hat{x}_{1i}$.

4.4.2 Analyse de stabilité

Considérons la fonction de Lyapunov candidate suivante

$$V = \frac{1}{2} \sum_{i=1}^n \left(e_i^T P_i e_i + \hat{e}_{1i}^2 + \frac{1}{g_i(\hat{x}_s)} \hat{e}_{2i}^2 + \frac{1}{\gamma_i} \tilde{\Theta}_i^T \tilde{\Theta}_i + \frac{1}{\eta_i} \tilde{\varepsilon}_i^T \tilde{\varepsilon}_i \right) \quad (4.51)$$

La dérivée temporelle de V le long de la solution de (4.47) est

$$\begin{aligned} \dot{V} = \sum_{i=1}^n & \left(\frac{1}{2} \dot{e}_i^T P_i e_i + \frac{1}{2} e_i^T P_i \dot{e}_i - \kappa_{1i} \hat{e}_{1i}^2 + \hat{e}_{2i} \left(u_i^*(t) - u_i(t) \right) \right. \\ & \left. + \frac{1}{\gamma_i} \tilde{\Theta}_i^T \dot{\tilde{\Theta}}_i + \frac{1}{\eta_i} \tilde{\varepsilon}_i^T \dot{\tilde{\varepsilon}}_i \right) \end{aligned} \quad (4.52)$$

ou encore

$$\begin{aligned} \dot{V} = \sum_{i=1}^n & \left(\frac{1}{2} e_i^T \left(A_i^T P_i + P_i A_i \right) e_i + e_i^T P_i \delta_i - \kappa_{1i} \hat{e}_{1i}^2 + \hat{e}_{2i} \left(u_i^*(t) - u_i(t) \right) \right. \\ & \left. + \frac{1}{\gamma_i} \tilde{\Theta}_i^T \dot{\tilde{\Theta}}_i + \frac{1}{\eta_i} \tilde{\varepsilon}_i^T \dot{\tilde{\varepsilon}}_i \right) \end{aligned} \quad (4.53)$$

En remplaçant par (4.17) et (4.48) dans (4.53), on obtient

$$\begin{aligned} \dot{V} = & \sum_{i=1}^n -e_i^T Q_i e_i + e_i^T P_i \delta_i - \kappa_{1i} \hat{e}_{1i}^2 \\ & + \hat{e}_{2i} \left(\Theta_i^{*T} \varphi_i(\hat{x}_s) + \varepsilon_i(\hat{x}_s) - \Theta_i^T \varphi_i(\hat{x}_s) - \hat{\varepsilon}_i \tanh\left(\frac{\hat{e}_{2i}}{\epsilon_i}\right) - \kappa_{2i} \hat{e}_{2i} \right) \\ & - \frac{1}{\gamma_i} \tilde{\Theta}_i^T \Theta_i - \frac{1}{\eta_i} \tilde{\varepsilon}_i^T \hat{\varepsilon}_i \end{aligned} \quad (4.54)$$

avec les inégalités suivantes

$$\begin{aligned} e_i^T P_i \delta_i & \leq \frac{1}{2} \|e_i\|^2 + \frac{1}{2} \|P_i\|^2 (\delta_{fi}^2 + \delta_{gi}^2) \\ & \leq \frac{1}{2} \|e_i\|^2 + \frac{1}{2} \|P_i\|^2 (\bar{\delta}_{fi}^2 + \bar{\delta}_{gi}^2) \end{aligned} \quad (4.55)$$

L'équation (4.54) peut être bornée de la manière suivante

$$\begin{aligned} \dot{V} \leq & \sum_{i=1}^n -e_i^T Q_i e_i + \frac{1}{2} \|e_i\|^2 + \frac{1}{2} \|P_i\|^2 (\bar{\delta}_{fi}^2 + \bar{\delta}_{gi}^2) - \kappa_{1i} \hat{e}_{1i}^2 - \kappa_{2i} \hat{e}_{2i}^2 \\ & + |\hat{e}_{2i}| \varepsilon_i - \hat{e}_{2i} \hat{\varepsilon}_i \tanh\left(\frac{\hat{e}_{2i}}{\epsilon_i}\right) + \frac{1}{\gamma_i} \tilde{\Theta}_i^T \left(\gamma_i \hat{e}_{2i} \varphi_i(\hat{x}_s) - \Theta_i \right) \\ & - \hat{e}_{2i} \varepsilon_i \tanh\left(\frac{\hat{e}_{2i}}{\epsilon_i}\right) + \hat{e}_{2i} \hat{\varepsilon}_i \tanh\left(\frac{\hat{e}_{2i}}{\epsilon_i}\right) + \frac{1}{\eta_i} \tilde{\varepsilon}_i \left(\eta_i \hat{e}_{2i} \tanh\left(\frac{\hat{e}_{2i}}{\epsilon_i}\right) - \hat{\varepsilon}_i \right) \end{aligned} \quad (4.56)$$

En utilisant (4.49) et (4.50), (4.56) s'écrit

$$\begin{aligned} \dot{V} \leq & \sum_{i=1}^n -e_i^T Q_i e_i - \kappa_{1i} \hat{e}_{1i}^2 - \kappa_{2i} \hat{e}_{2i}^2 + \frac{1}{2} \|e_i\|^2 + \frac{1}{2} \|P_i\|^2 (\bar{\delta}_{fi}^2 + \bar{\delta}_{gi}^2) \\ & + \varepsilon_i \varsigma + \frac{\sigma_{1i}}{\gamma_i} \tilde{\Theta}_i^T \Theta_i + \frac{\sigma_{2i}}{\eta_i} \tilde{\varepsilon}_i \hat{\varepsilon}_i \end{aligned} \quad (4.57)$$

avec (4.37) et (4.38), (4.57) devient

$$\begin{aligned} \dot{V} \leq & \sum_{i=1}^n -e_i^T Q_i e_i - \kappa_{1i} \hat{e}_{1i}^2 - \kappa_{2i} \hat{e}_{2i}^2 + \frac{1}{2} \|e_i\|^2 + \frac{1}{2} \|P_i\|^2 (\bar{\delta}_{fi}^2 + \bar{\delta}_{gi}^2) \\ & + \bar{\delta}_i \varsigma - \frac{\sigma_{1i}}{2\gamma_i} \tilde{\Theta}_i^T \tilde{\Theta}_i + \frac{\sigma_{1i}}{2\gamma_i} \Theta_i^{*T} \Theta_i^* - \frac{\sigma_{2i}}{2\eta_i} \tilde{\varepsilon}_i^2 + \frac{\sigma_{2i}}{2\eta_i} |\varepsilon_i|^2 \end{aligned} \quad (4.58)$$

Choisissons $\bar{\lambda} = \min \{2 \min(\lambda_{\min}(Q_i) - 1/2), \sigma_{1i}, \sigma_{2i}, 2\kappa_{1i}, 2\kappa_{2i}\}$, (4.57) peut être réécrite ainsi

$$\dot{V} \leq -\bar{\lambda} V + \varrho \quad (4.59)$$

avec

$$\begin{aligned} \varrho = & \sum_{i=1}^6 \frac{\sigma_{1i}}{2\gamma_i} \Theta_i^{*T} \Theta_i^* + \frac{\sigma_{2i}}{2\eta_i} |\varepsilon_i^*|^2 \\ & + \frac{1}{2} \|P_i\|^2 (\bar{\delta}_{fi}^2 + \bar{\delta}_{gi}^2) + \varepsilon_i^* \varsigma \end{aligned} \quad (4.60)$$

Ainsi la stabilité du système global contrôleur/observateur peut être formellement prouvée à travers le théorème suivant

Théorème 4.2. *Considérons le système non linéaire du Quadrotor (4.1). Supposons que les hypothèses 4.5 et 4.6 sont satisfaites. Alors, la loi de commande (4.48) avec l'observateur d'état (4.44) garantissent la bornitude de tous les signaux de la boucle fermée ainsi que la convergence des erreurs \hat{e}_{1i} et e_i vers un petit voisinage autour de zéro.*

Prouve. A partir de (4.59), on déduit

$$V(t) \leq V(0)e^{-\bar{\lambda}t} + \frac{\varrho}{\bar{\lambda}} \quad (4.61)$$

A partir de (4.61), on conclut que pour $V \geq \varrho/\bar{\lambda}$, nous avons $\dot{V} \leq 0$. Donc, en se basant sur le théorème de Lyapunov, les signaux en boucle fermée $\hat{e}_{1i}(t)$, $\hat{e}_{2i}(t)$, $e_i(t)$, $\tilde{\Theta}_i(t)$, $\tilde{\varepsilon}_i$ et $u_i(t)$, ainsi que les états du système x_{1i} , x_{2i} , \hat{x}_{1i} et \hat{x}_{2i} ; en plus, pour $\mu \geq \sqrt{\varrho/\bar{\lambda}}$, il existe un temps $T > 0$, tel que $|\hat{e}_{2i}(t)| \leq \bar{\mu}$ pour tout $t \geq T$. Afin d'assurer la convergence de l'erreur de poursuite vers un petit voisinage de zéro, il suffit de choisir des valeurs appropriées pour les paramètres γ_i , η_i , ε_i , σ_{1i} , σ_{2i} , κ_{1i} et κ_{2i} . Par conséquent, il est possible de rendre $\sqrt{\varrho/\bar{\lambda}}$ aussi petit que souhaité. Donc, les erreurs de poursuite convergent vers des ensembles résiduels tels que $|\hat{e}_{1i}(t)| \leq 2\alpha_i\bar{\mu}$. ■

4.5 Stratégie de commande

Le modèle dynamique du Quadrotor décrit par (4.1) a six sorties $(x, y, z, \phi, \theta, \psi)$ alors qu'il n'a que quatre entrées indépendantes (u_1, u_2, u_3, u_4) . Par conséquent, le Quadrotor est un système sous-actionnée. Nous ne pouvons pas contrôler tous les états en même temps [101]. Une combinaison possible de sorties contrôlées peut être (x, y, z, ψ) afin de suivre les positions désirés et de stabiliser l'angle de roulis et de tangage (ϕ, θ) qui introduit une dynamique zéro stable dans le système. Pour résoudre ce problème, nous supposons l'existence de deux entrées de commande auxiliaires (u_5, u_6) en plus de quatre entrées de commande du Quadrotor, de tel sorte que chaque sortie du système sera contrôlée indépendamment. les entrées de commande auxiliaires u_5 et u_6 représentent

la relation entre l'angle de tangage et la position x , l'angle de roulis et la position y respectivement [139] [134].

La stratégie de commande adoptée est basée principalement sur deux boucles (boucle interne et boucle externe). La boucle interne contient la dynamique de l'attitude avec trois lois de commande : commande de roulis, commande de tangage et commande de lacet. La boucle externe inclut la dynamique de position avec trois lois de commande: la commande d'altitude et la commande de positions (x, y) . La boucle de commande externe génère une désirée pour le système de roulis (ϕ) et de tangage (θ) en passant par le bloc de correction. Ce bloc corrige la rotation de roulis et de tangage en fonction de l'angle de lacet (ψ) et les lois de commande de la position (x, y) u_5 et u_6 . Les schémas des figures 4.1 et 4.2 représente la stratégie de commande adaptative floue directe avec observateur d'état adoptée.

$$u_5 = \frac{u_1}{m} (C_{x_{11}} S_{x_{12}} C_{x_{13}} + S_{x_{11}} S_{x_{13}}) \quad (4.62)$$

$$u_6 = \frac{u_1}{m} (C_{x_{11}} S_{x_{12}} S_{x_{13}} - S_{x_{11}} C_{x_{13}}) \quad (4.63)$$

avec $C_{(.)}$ et $S_{(.)}$ représente $\cos(.)$ et $\sin(.)$ respectivement. En utilisant l'approximation de la loi de commande u_4

$$u_1 = \frac{mg}{C_{x_{11}} C_{x_{12}}} \quad (4.64)$$

remplaçons (4.64) dans (4.62) et (4.63), et appliquant $(C_{x_{13}}(4.62) + S_{x_{13}}(4.63))$, on obtient

$$\left(\frac{u_5 C_{x_{13}} + u_6 S_{x_{13}}}{g} \right) = \tan(x_{12}) \quad (4.65)$$

en appliquant $(S_{x_{13}}(4.62) - C_{x_{13}}(4.63))$, on a

$$\left(\frac{u_5 S_{x_{13}} - u_6 C_{x_{13}}}{g} C_{x_{12}} \right) = \tan(x_{11}) \quad (4.66)$$

à partir de (4.65) et (4.66), avec $x_{11} = \phi_r$, $x_{12} = \theta_r$, $x_{13} = \psi$ on obtient

$$\begin{aligned} \theta_r &= \text{atan} \left(\frac{u_5 \cos \psi + u_6 \sin \psi}{g} \right) \\ \phi_r &= \text{atan} \left(\frac{u_5 \sin \psi - u_6 \cos \psi}{g} \cos \theta_r \right) \end{aligned} \quad (4.67)$$

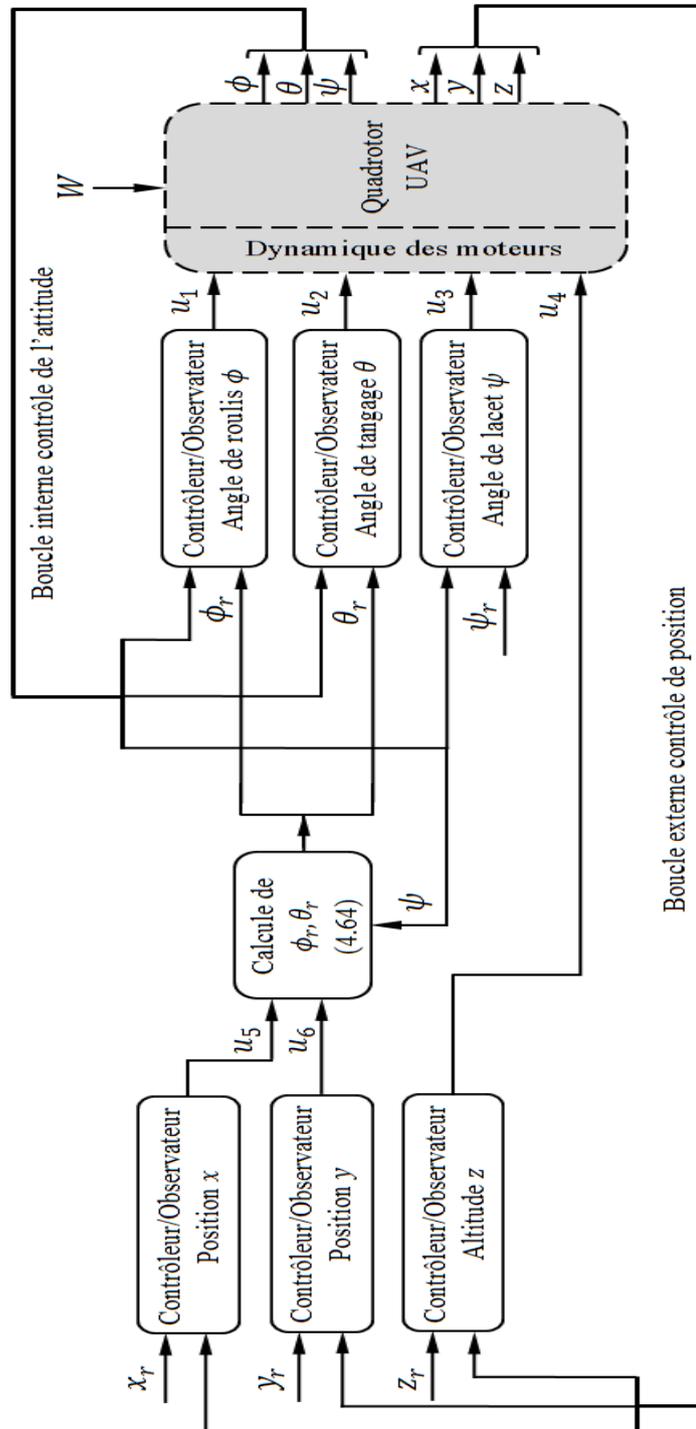


Figure 4.1: Structure de la commande adaptative floue directe avec observateur

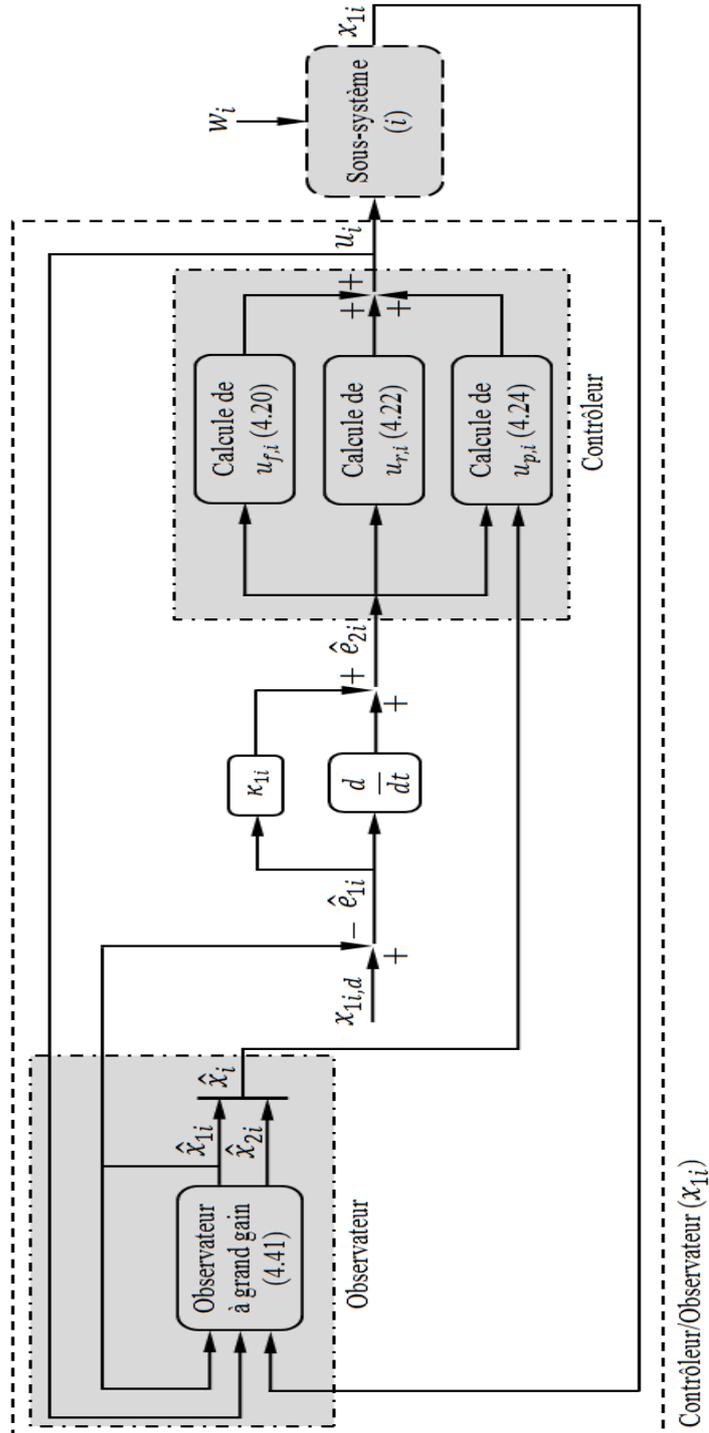


Figure 4.2: Structure de commande pour chaque $i^{\text{ème}}$ sous système

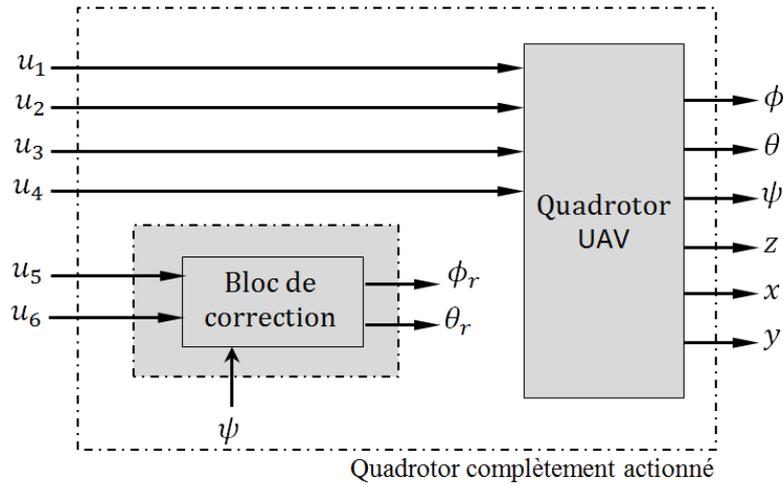


Figure 4.3: Système du Quadrotor complètement actionné

La signification physique des commandes auxiliaires est que le mouvement latéral (x, y) est commandé par les angles de roulis et de tangage. C'est à dire que les mouvements de roulis et de tangage doivent prendre les angles désirées (ϕ_r, θ_r) pour garantir le contrôle du mouvement latéral. Figure 4.3 représente le système du Quadrotor complètement actionné avec six entrées (u_1, \dots, u_6) , et six sorties $(\phi, \theta, \psi, z, x, y)$.

4.6 Résultat de simulation

Afin de valider les stratégies de commande adaptative floue directe synthétise, des simulations sont réalisées pour vérifier la robustesse et les performances obtenues par le contrôleur proposé pour le problème de suivi de trajectoire d'un Quadrotor. Deux scénarios de simulation sont implémentés. Dans le premier scénario, une trajectoire circulaire est considérée sans tenir compte des perturbations externes. Dans le deuxième scénario, les perturbations externes (rafales de vent) sont considérées. Les simulation ont été implémentée sous l'environnement *Simulink* de *Matlab* avec une période d'échantillonnage $t = 0.001sec$.

Les paramètres du modèle de Quadrotor utilisé dans la simulation sont illustrés sur le tableau 2.1. Le modèle de rafales de vent présenté dans le chapitre 1 est utilisé pour générer la vitesse du vent le long des trois axes (latéral, longitudinal et vertical). Ensuite, ils sont considérés comme des perturbations externes additives à la vitesse de translation du Quadrotor. Les paramètres du modèle de vent sont: $L_u = L_v = 23.568$, $L_w = 3$, $\sigma_u = \sigma_v = 0.68$, $\sigma_w = 0.34$, avec une vitesse typique du vent de $3.4m/s$. Une variation

de 40% pour les paramètres d'inertie (I_x, I_y, I_z) est considéré pour tester la robustesse fournie par la stratégie de commande proposée vis à vis les incertitudes paramétriques.

Les variables d'entrée du système flou sont $e_{Fi} = [e_{1i}, \dot{e}_{1i}]^T$ ($i = 1, 2, 3$) pour la position, et $x_{Fi} = [x_{1i}, \dot{x}_{1i}]^T$ ($i = 4, 5, 6$) pour les angles d'Euler. Pour chaque variable x_{Fi} et e_{Fi} , trois ensembles flous N_f, Z_f, P_f sont choisis, avec les fonctions d'appartenance suivantes:

$$\begin{aligned}\mu_{F_{x_i}^1} &= \exp\left(-\frac{1}{2}\left(\frac{x_{Fi} + \pi/12}{\pi/18}\right)^2\right), & \mu_{F_{e_i}^1} &= \exp\left(-\frac{1}{2}\left(\frac{e_{Fi} + 10}{4}\right)^2\right) \\ \mu_{F_{x_i}^2} &= \exp\left(-\frac{1}{2}\left(\frac{x_{Fi}}{\pi/18}\right)^2\right), & \mu_{F_{e_i}^2} &= \exp\left(-\frac{1}{2}\left(\frac{e_{Fi}}{4}\right)^2\right) \\ \mu_{F_{x_i}^3} &= \exp\left(-\frac{1}{2}\left(\frac{x_{Fi} - \pi/12}{\pi/18}\right)^2\right), & \mu_{F_{e_i}^3} &= \exp\left(-\frac{1}{2}\left(\frac{e_{Fi} - 10}{4}\right)^2\right)\end{aligned}$$

ils sont définis dans l'intervalle $[-\pi/4, \pi/4]$ pour les angles d'Euler et $[-15, 15]$ pour la position. La trajectoire de référence est un cercle de 8 mètre de hauteur évoluant dans l'espace cartésien défini par

$$\begin{aligned}x_d &= 1 - \sin\left(\frac{\pi t}{30} + \frac{\pi}{2}\right) m \\ y_d &= -2 \sin\left(\frac{\pi t}{30} + \pi\right) m \\ z_d &= 5 - 3 \sin\left(\frac{\pi t}{30} + \frac{\pi}{2}\right) m\end{aligned}$$

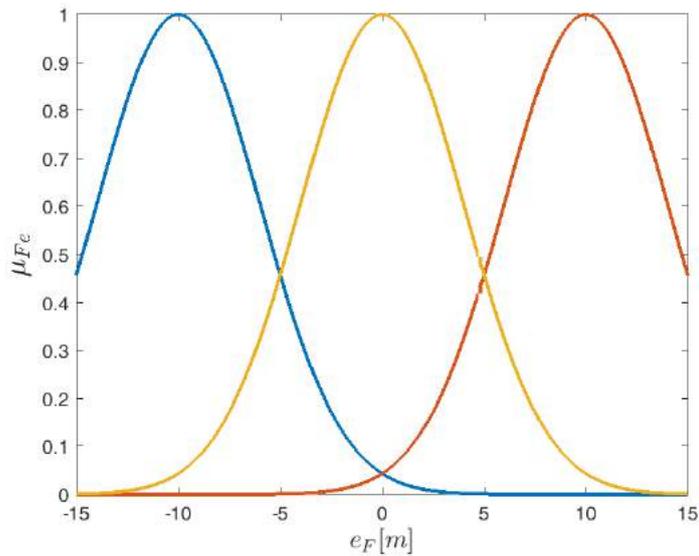


Figure 4.4: Fonctions d'appartenance, position

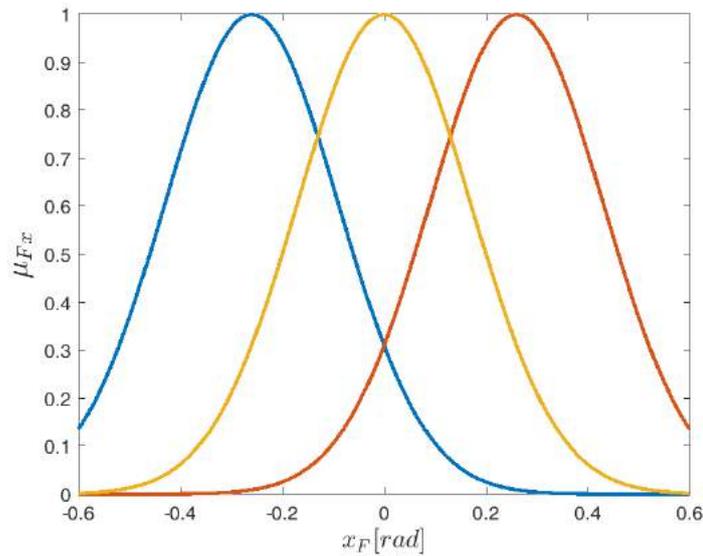


Figure 4.5: Fonctions d'appartenance, angles d'Euler

4.6.1 Simulation de la commande adaptative floue directe

Les paramètres de conception pour la commande adaptative floue directe CAFD sont choisis comme suit $\kappa_{1i} = 1$ ($i = 1, 2, 4, 5, 6$), $\kappa_{13} = 3$, $\kappa_{2i} = 2$ ($i = 1, 2, 4, 5, 6$), $\kappa_{23} = 6$. Les gains de la loi d'adaptation floue sont choisis comme suit $\gamma_i = 1.2$ ($i = 1, 2, 4, 5, 6$), $\gamma_3 = 2.8$, $\sigma_{1i} = 0.01$ et les gains du terme robuste $\epsilon_i = 0.001$, $\eta_i = 0.03$ et $\sigma_{2i} = 0.01$. Les valeurs initiales pour la position et les angles d'Euler sont $x(0) = 0.5$, $y(0) = 0$, $z(0) = 0.5$, $\phi(0) = 0$, $\theta(0) = 0$, $\psi(0) = 0.35$, et les valeurs initiales des lois d'adaptation paramétrique $\hat{\epsilon}_i(0) = 0$, $\Theta_i(0) = 0_{9 \times 1}$. Les vitesses linéaires et angulaires initiales sont choisies égales à zéro.

Suivi de trajectoire sans perturbations externes

Les résultats de la simulation pour ce cas sont présentés sur les figures 4.6-4.9. La figure 4.6 et 4.7 représente respectivement la position et l'attitude du véhicule, l'évolution du drone dans l'espace est montrée sur la figure 4.9. Ces figures montre que la commande CAFD permet au drone de suivre la trajectoire de référence malgré qu'il démarre d'une position initial différent du celle de la référence. La commande CAFD génère des entrées lisses représenté sur la figure 4.8.

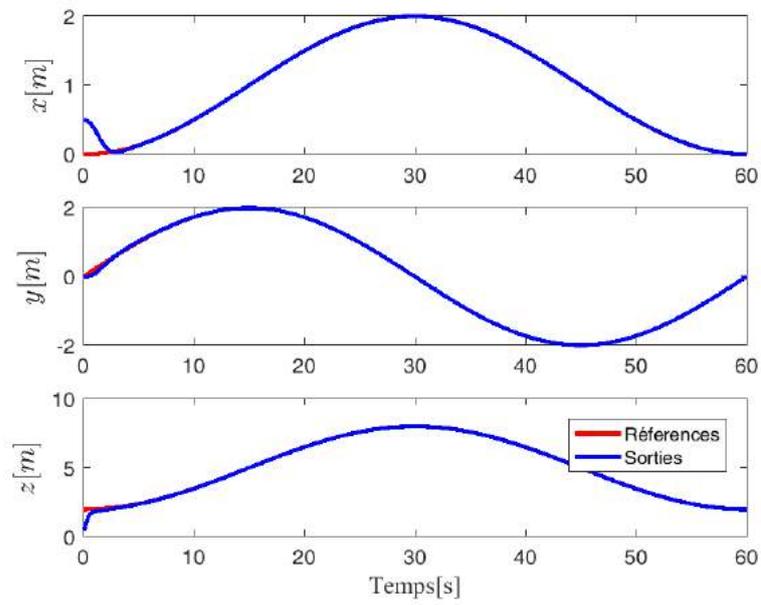


Figure 4.6: Suivi de position sans perturbations externes, CAFD

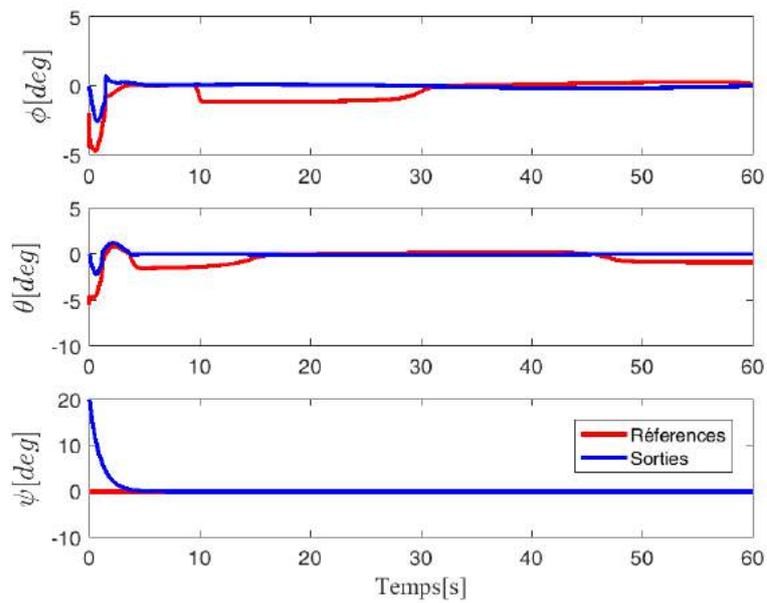


Figure 4.7: Suivi d'attitudes sans perturbations externes, CAFD

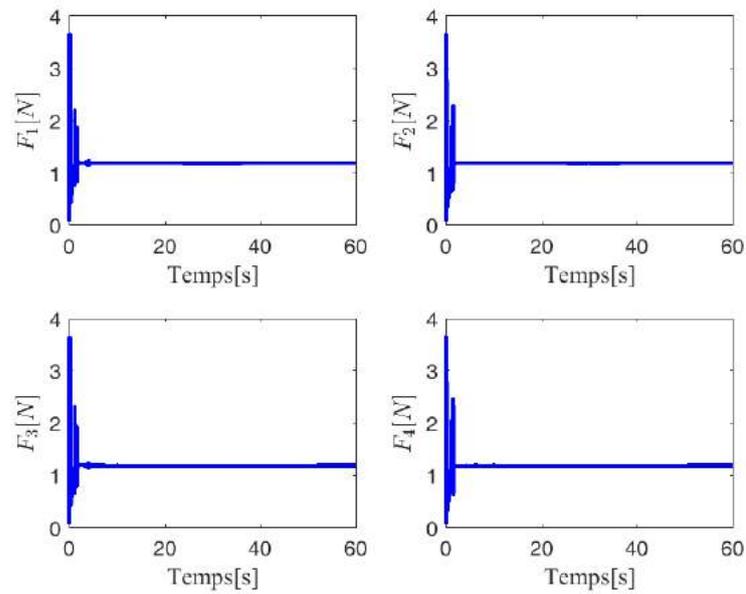


Figure 4.8: Entrées de commande sans perturbations externes, CAFD

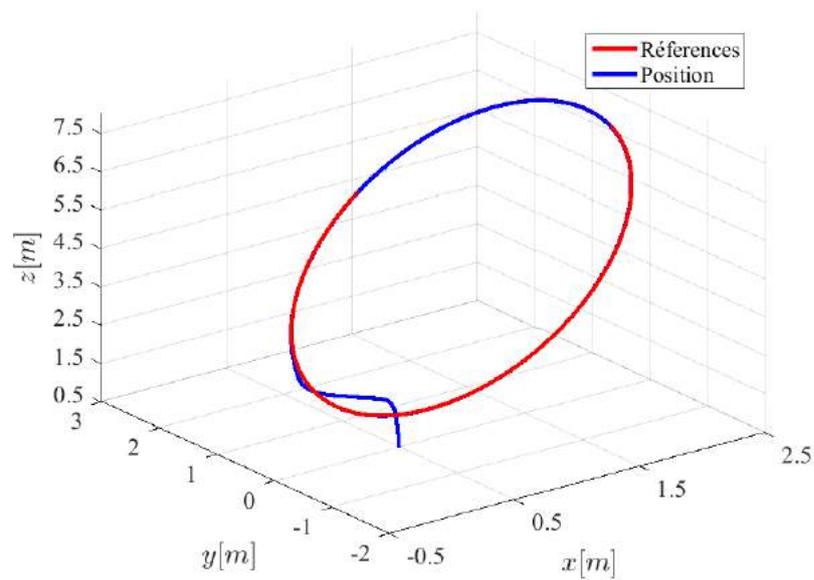


Figure 4.9: Position en 3D sans perturbations externes, CAFD

Suivi de trajectoire avec perturbations externes

Les résultats de simulation pour ce scénario sont présentés sur les figures 4.10-4.14. Les vitesses du vent calculées et appliquées sur la vitesse de translation du Quadrotor

sont indiquées dans la figure 4.10. Les figures 4.11, 4.12 et 4.14 montre que la commande CAFD arrive a compensé les effets des perturbation externes (rafale de vents) et permet au Quadrotor de suivre la trajectoire de référence avec des entrées de commande acceptables (figure 4.13).

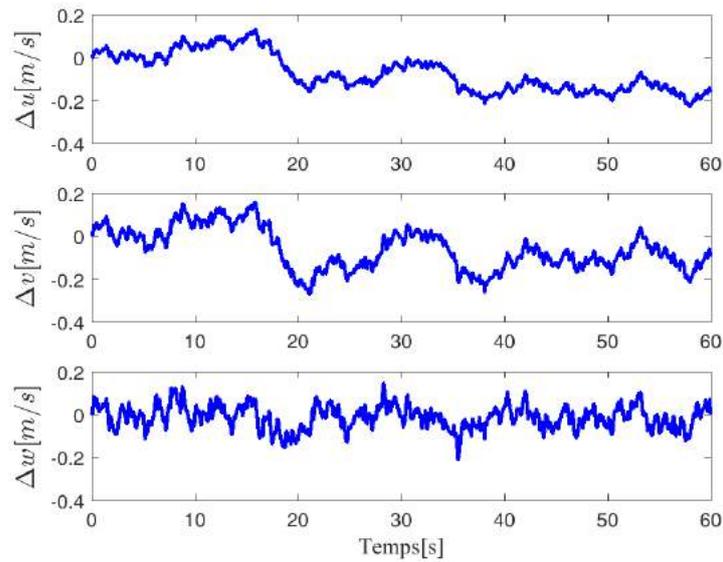


Figure 4.10: Vitesse du vent

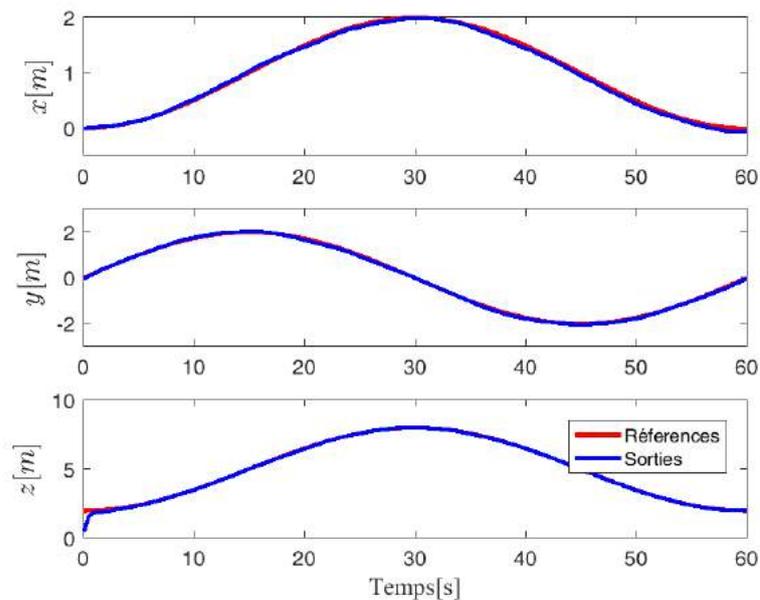


Figure 4.11: Suivi de position avec perturbations externes, CAFD

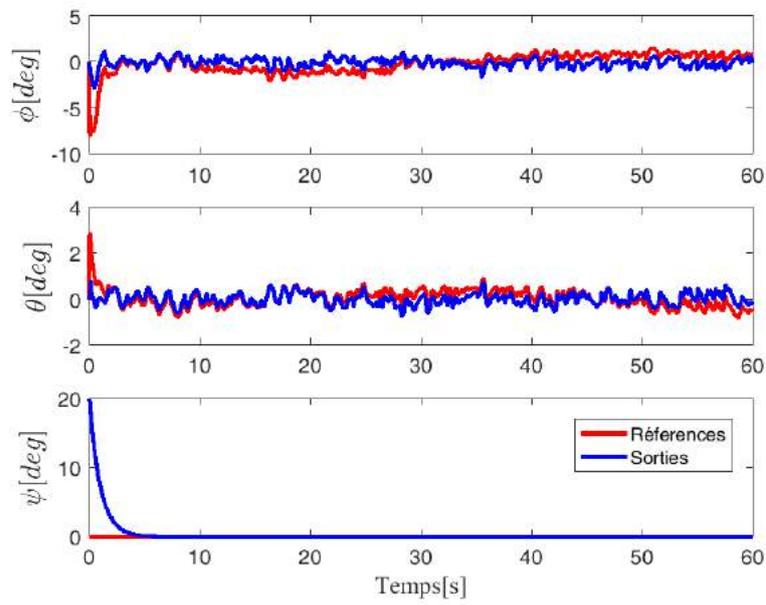


Figure 4.12: Suivi d'attitudes avec perturbations externes, CAFD

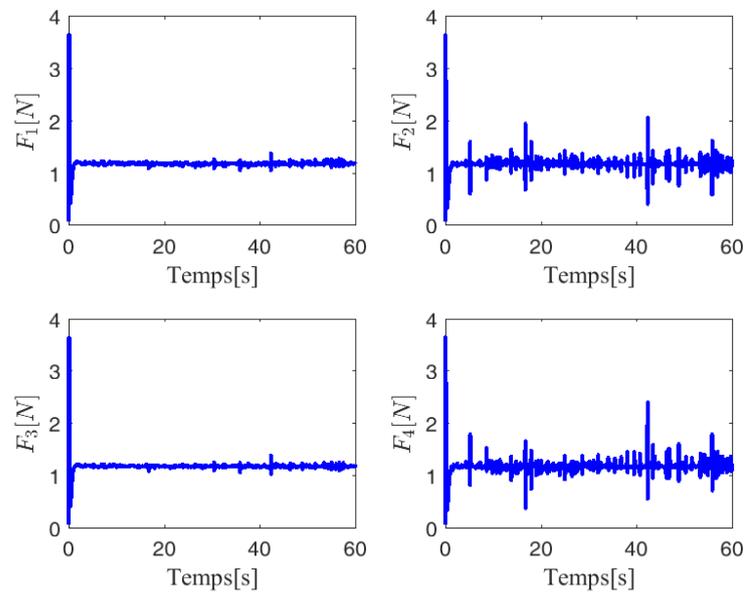


Figure 4.13: Entrées de commande avec perturbations externes, CAFD

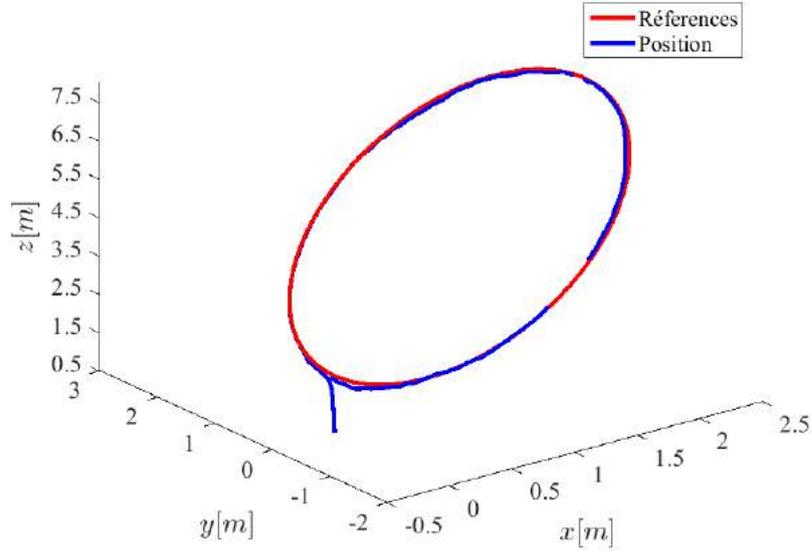


Figure 4.14: Position en 3D avec perturbations externes, CAFD

4.6.2 Simulation de la commande adaptative floue directe avec observateur

Les paramètres de conception pour la commande adaptative floue directe avec observateur CAFDO sont choisis comme suit $\kappa_{11} = 0.3, c_{12} = 0.4, c_{13} = 1, c_{14} = 3, c_{15} = 5, c_{16} = 3, k_{11} = 8, k_{12} = 8, k_{13} = 5, k_{14} = 8, k_{15} = 8, k_{16} = 8, c_{21} = 0.6, c_{22} = 0.8, c_{23} = 2, c_{24} = 6, c_{25} = 10, c_{26} = 6, k_{21} = 16, k_{22} = 16, k_{23} = 10, k_{24} = 16, k_{25} = 16, k_{26} = 16$. Les gains de la loi d'adaptation floue sont choisis comme suit $\gamma_i = 0.9$ ($i = 1, 2, 3, 5, 6$), $\gamma_4 = 1.8, \sigma_{1i} = 0.01$ et les gains du terme robuste $\epsilon_i = 0.001, \eta_i = 0.03$ et $\sigma_{2i} = 0.01$. Les valeurs initiales pour la position et les angles d'Euler sont $x_{11}(0) = 0, x_{12}(0) = 0, x_{13}(0) = 0, x_{14}(0) = 1, x_{15}(0) = 0, x_{16}(0) = 0, \hat{x}_{11}(0) = 0, \hat{x}_{12}(0) = 0, \hat{x}_{13}(0) = 0.179, \hat{x}_{14}(0) = 2, \hat{x}_{15}(0) = 0, \hat{x}_{16}(0) = 0$, et les valeurs initiales des lois d'adaptation paramétrique $\hat{\epsilon}_i(0) = 0, \Theta_i(0) = [0, 0, 0, 0, 0, 0, 0, 0, 0]$. Les vitesses linéaires et angulaires initiales sont choisies égales à zéro.

Suivi de trajectoire sans perturbations externes

Les résultats de la simulation pour le suivi de cette trajectoire de référence sont présentés par les figures 4.15-4.18. La trajectoire désirée en 3D et la trajectoire de Quadrotor générée par la commande adaptative floue directe sont illustrées sur la figure 4.18, tandis que les réponses de suivi de la position et de l'attitude du Quadrotor sont présentées par la figure 4.15 et à la figure 4.16 respectivement. On peut remarquer que le suivi est

réalisé avec une erreur nulle pour toutes les sorties de Quadrotor pour l'approche de commande proposée. La Figure 4.16 montre comment les angles désirés de roulis et de tangage changent leurs valeurs dans un intervalle de 20 deg entre -10 deg et $+10 \text{ deg}$ pour assurer les performances souhaité en terme de suivi pour le mouvement en x et en y . Comme représenté sur la figure 4.17, la stratégie de commande adaptative floue génère des signaux d'entrée de commande lisses et bornés.

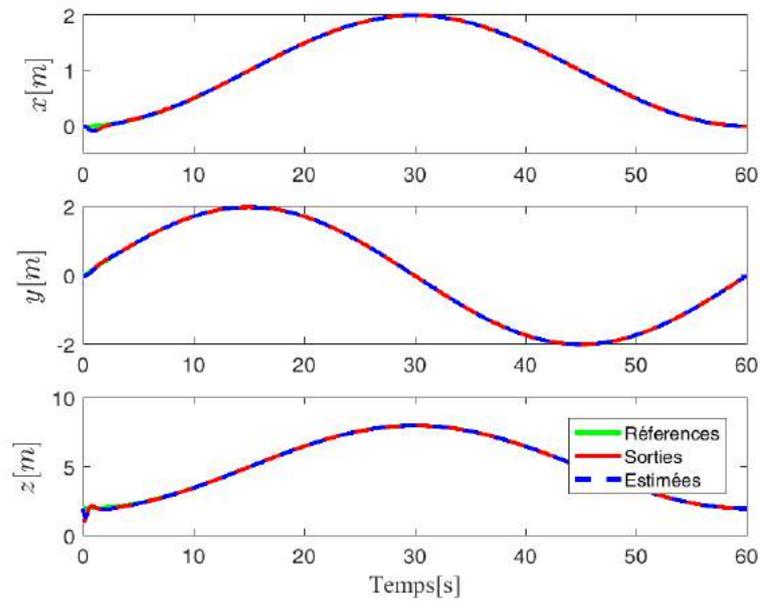


Figure 4.15: Suivi de position sans perturbations externes, CAFDO

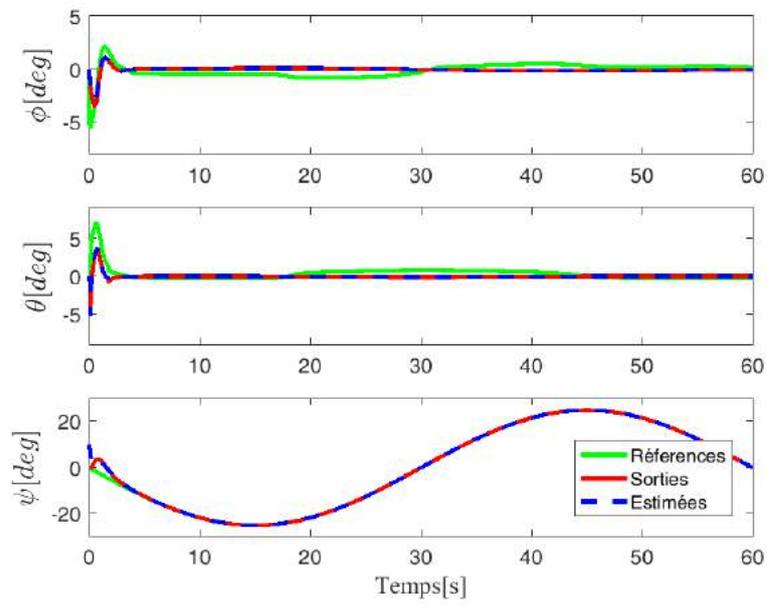


Figure 4.16: Suivi d'attitudes sans perturbations externes, CAFDO

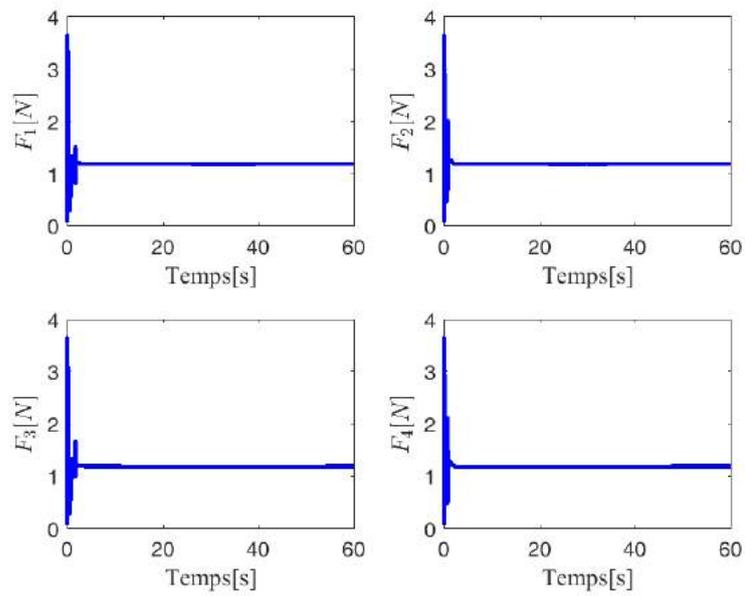


Figure 4.17: Entrées de commande sans perturbations externes, CAFDO

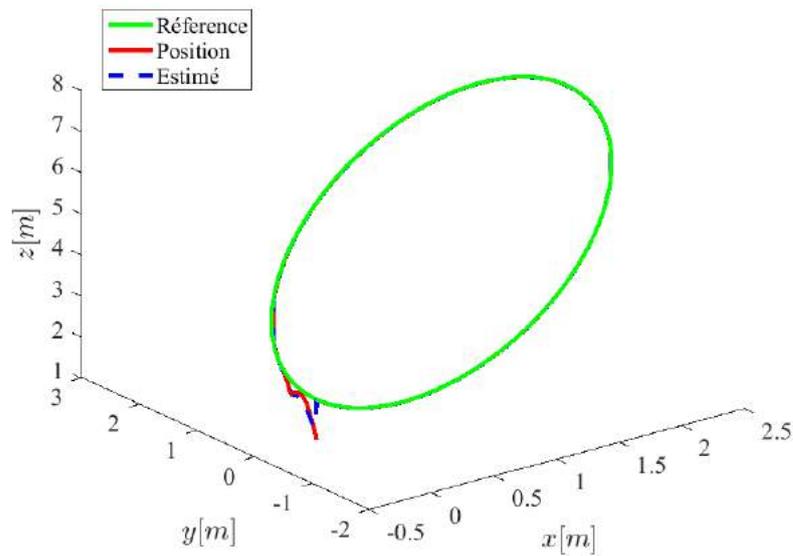


Figure 4.18: Position en 3D sans perturbations externes, CAFDO

Suivi de trajectoire avec perturbations externes

Les résultats de simulation pour ce scénario sont présentés sur les figures 4.19-4.22. On peut voir clairement comment à partir d'une position initiale loin de la position désirée, la stratégie de commande adaptative floue proposée est capable d'assurer le suivi des trajectoires de références par le Quadrotor malgré la présence des perturbations externes (effet de vent). En plus, les figures 4.19 et 4.20 montrent les bonnes performances de suivi pour les sorties du Quadrotor obtenues par le contrôleur proposé. On peut voir clairement l'effet de la rafale de vent sur le mouvement du Quadrotor dans les angles de roulis et de tangage (figure 4.19), les entrées de commande (figure 4.21) et la position en 3D (figure 4.22). Il est à noter que les entrées de commande changent leurs valeurs pour assurer les performances désirées.

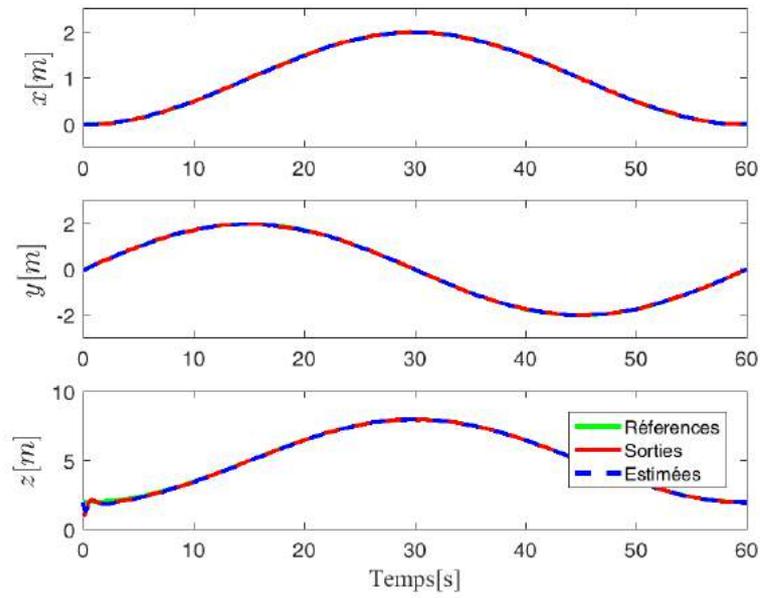


Figure 4.19: Suivi de position avec perturbations externes, CAFDO

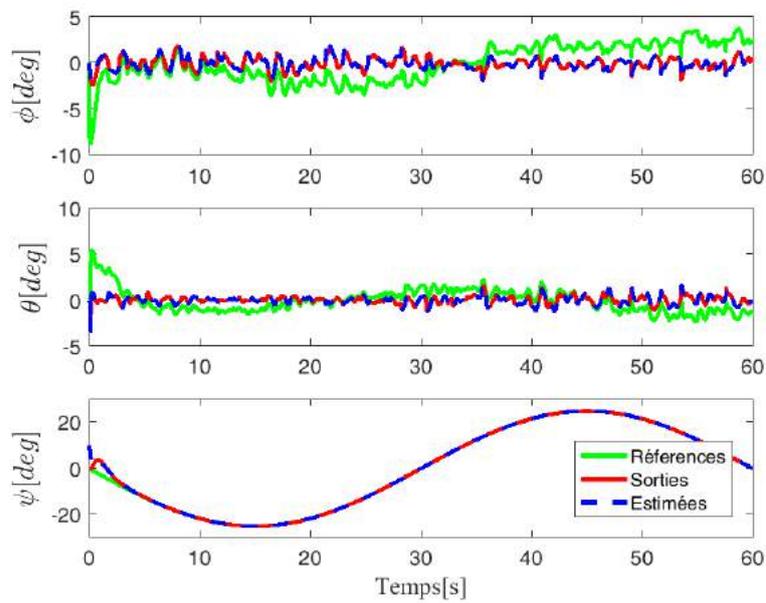


Figure 4.20: Suivi d'attitudes avec perturbations externes, CAFDO

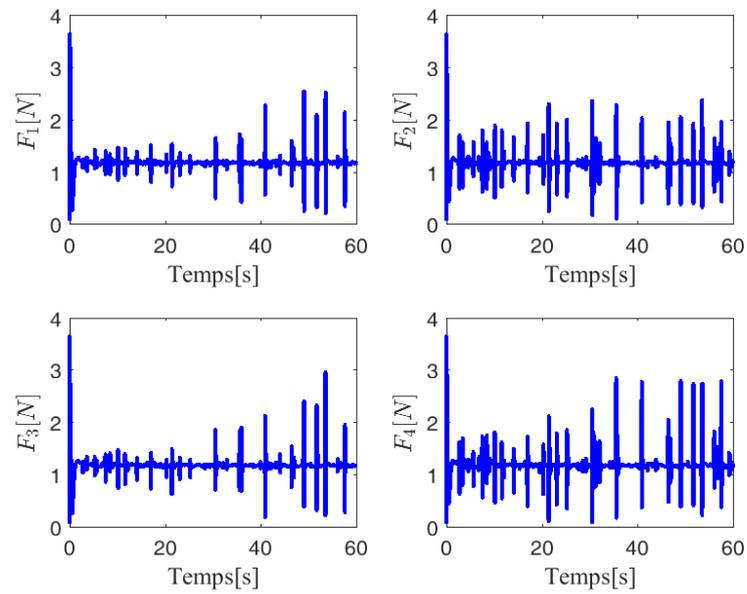


Figure 4.21: Entrées de commande avec perturbations externes, CAFDO

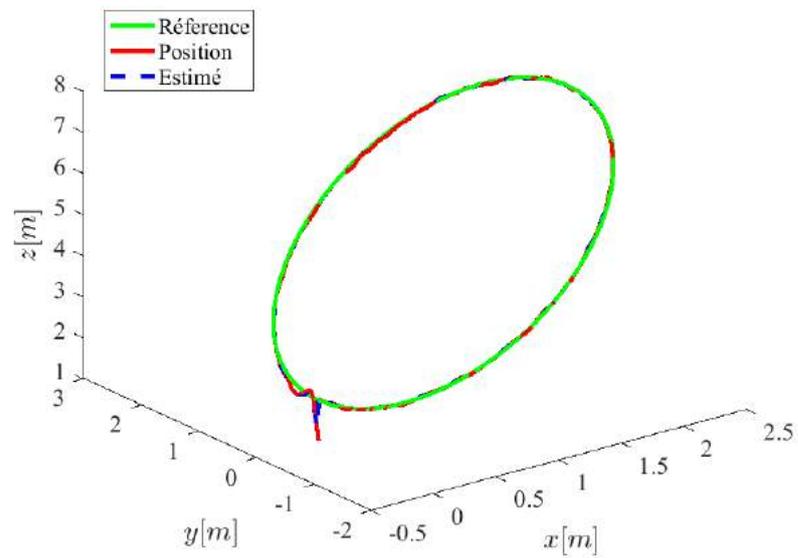


Figure 4.22: Position en 3D avec perturbations externes, CAFDO

4.7 Conclusion

Dans ce chapitre, une loi de commande adaptative floue directe a été proposée et développée pour une classe des systèmes robotiques sous actionnées. Après avoir montré l'existence d'une commande explicite basée sur le modèle dynamique du système capable d'assurer les objectifs de commande, ensuite nous nous sommes tournés vers l'approximation de cette commande idéale inconnue par une loi de commande adaptative floue directe avec l'aide d'un terme robuste et une action proportionnelle dérivée. Le mécanisme d'adaptation flou est synthétisé de telle sorte que la fonction de Lyapunov soit négative. Ensuite, nous avons élargi notre étude au cas où le vecteur d'état est reconstruit par un observateur ; ainsi la structure de commande à base d'observateur a été dégagée aussi.

En effet, le développement des calculateurs et des capteurs embarqués rendent l'implémentation de contrôleur adaptatif flou simple et réalisable, avec un grand nombre d'application émergée par les drones miniatures à voilure tournante et la nécessité d'un contrôleur performant sont les raisons qui nous ont poussées de développer ce contrôleur pour un robot volant de type Quadrotor. Il est important de noter ici que les hypothèses faites lors de la conception de cette approche de commande sont communément utilisées dans la littérature et vérifiées par plusieurs systèmes physiques. Le contrôleur et les mécanismes d'adaptation, outre leur indépendance du modèle dynamique, ils assurent la stabilité de la boucle fermée ainsi que la convergence des erreurs de poursuite vers un petit voisinage de zéro. Les résultats de simulation issus de l'application de cette approche dans la commande du Quadrotor montrent bien la faisabilité et l'efficacité de cette approche dans le rejet des perturbations externes et les perturbations internes.

Dans le prochain chapitre, nous allons introduire un algorithme de commande adaptative floue indirecte avec surface dynamique à base d'un observateur de perturbation, validé en simulation pour un Quadrotor.

Chapitre 5

Commande adaptative floue robuste à base d'observateur de perturbation

5.1 Introduction

La conception d'une stratégie de commande adaptative floue robuste à base d'un observateur de perturbation pour une classe des robots volants à voilure tournante non linéaires fait l'objet de ce chapitre. Dans la commande à base d'observateur de perturbation, au lieu de considérer le problème de contrôle pour un système non linéaire avec des perturbations comme un seul problème, il est divisé en deux sous-problèmes, chacun avec ses propres objectifs de conception [140]. Le premier sous-problème est le même que le problème de commande pour un système non linéaire sans perturbations et son objectif est de stabiliser le système et d'obtenir les spécifications de performance de suivi et/ou de régulation souhaitées. Le second sous-problème est l'atténuation des perturbations. Un observateur de perturbation non linéaire est conçu pour estimer les perturbations externes et ensuite utilisée cet estimé pour compenser l'influence des perturbations.

Les drones miniatures à voilure tournante sont généralement affectés par des perturbations externes, telles que les rafales de vent et les forces aérodynamiques. Pour surmonter ce problème de nombreuses approches ont été proposées, on peut citer quelques études sur le Quadrotor. Dans [110], les auteurs ont proposé une nouvelle application d'une approche de commande hiérarchique, à l'aide d'un observateur de perturbation, pour la commande de vol stationnaire d'un véhicule aérien sans pilote à voilure tournante. Alors que dans [141] le problème de la conception et la validation expérimentale d'un contrôleur pour guider un véhicule Quadrotor le long d'une trajectoire, tout en rejetant les perturbations constantes a été adressée. Une stratégie de commande avec rejet des perturbations pour le suivi d'attitude d'un aéronef à la fois avec des incertitudes internes et des perturbations externes a été étudiée dans [142]. Les auteurs de [143] proposent un schéma de commande non linéaire hiérarchique pour le Quadrotor soumis à des variations de la charge utile et des rafales de vent variant rapidement dans le temps afin de suivre une trajectoire 3D.

La loi de commande que nous allons proposer ici se base principalement sur la commande développée dans le chapitre précédent. La particularité de cette commande est la présence de l'estimé des perturbations externe telle que les rafales de vent. Dans un premier temps, nous allons synthétiser un observateur adaptatif de perturbation pour estimer les effets de ces derniers sur le système. Par la suite nous allons employer cet estimé dans la loi de commande avec le terme adaptatif flou et le terme robuste. La stabilité de la boucle de commande sera étudiée par l'approche de Lyapunov.

Ce chapitre est construit ainsi : la section 5.2 expose la formulation du problème. Quant à la section 5.3, elle est consacrée à la conception de la loi de commande. La synthèse de l'observateur de perturbation fait l'objet de la section 5.4. L'analyse de stabilité du système en boucle fermé est illustré dans la section 5.5. Les résultat de simulation de l'approche de commande proposée dans ce chapitre sont présentés dans la section 5.6.

5.2 Formulation de problème

Soit le le système de Quadrotor soumis à des perturbations externes composé de n sous systèmes. Chaque sous système $\sum_i, i = 1, \dots, n$, est représenté comme suit

$$\begin{cases} \dot{x}_{1i} = x_{2i} \\ \dot{x}_{2i} = f_i(x_s) + g_i(x_s)u_i(t) + \alpha_i w_i \\ y_i = x_{1i} \end{cases} \quad (5.1)$$

avec $i = 1, \dots, n$, $n = 6$ est le nombre de sous-système, $f_i(x_s)$ et $g_i(x_s)$, sont des fonction non linéaires lisses, $x_s = [x_{11}, x_{21}, \dots, x_{16}, x_{26}]^T \in \mathfrak{R}^{12}$ et $x_{11} = x$, $x_{21} = \dot{x}$, $x_{12} = y$, $x_{22} = \dot{y}$, $x_{13} = z$, $x_{23} = \dot{z}$, $x_{14} = \phi$, $x_{24} = \dot{\phi}$, $x_{15} = \theta$, $x_{25} = \dot{\theta}$, $x_{16} = \psi$, $x_{26} = \dot{\psi}$, $u = [u_1, \dots, u_6]^T \in \mathfrak{R}^6$ est le vecteur des entrées de commande, $Y = [y_1, \dots, y_6]^T \in \mathfrak{R}^6$ est le vecteur des sorties et $W = [0, w_1, \dots, 0, w_6]^T$ est le vecteur de rafales du vent.

$$\begin{aligned} f_1(x_s) &= 0, \quad g_1(x_s) = 1 \\ f_2(x_s) &= 0, \quad g_2(x_s) = 1 \\ f_3(x_s) &= -g, \quad g_3(x_s) = (\cos \phi \cos \theta)/m \\ f_4(x_s) &= \dot{\theta}\dot{\psi}(I_y - I_z)/I_x + J\dot{\theta}\varpi/I_x, \quad g_4(x_s) = l/I_x \\ f_5(x_s) &= \dot{\phi}\dot{\psi}(I_z - I_x)/I_y - J\dot{\phi}\varpi/I_y, \quad g_5(x_s) = l/I_y \\ f_6(x_s) &= \dot{\phi}\dot{\theta}(I_x - I_y)/I_z, \quad g_6(x_s) = 1/I_z \end{aligned}$$

L'objectif principal de la conception de commande est de développer un contrôleur de suivi de trajectoire adaptatif flou robuste basé sur l'observateur de perturbations. L'observateur de perturbations fournit une estimation des signaux de perturbation utilisés par le contrôleur pour compenser l'effet des perturbations. Le système global contrôleur/observateur génère les signaux d'entrée pour les moteurs de Quadrotor en s'assurant que les sorties $(x(t), y(t), z(t), \psi(t))$ suivent les trajectoires de références $(x_r(t),$

$y_r(t)$, $z_r(t)$, $\psi_r(t)$), et garantir la bornitude de tous les signaux de la boucle fermée.

5.3 Conception de la loi de commande

Dans cette section, nous allons concevoir une commande adaptative floue robuste, les lois d'adaptation des paramètres, ainsi que un observateur de perturbations de telle sorte que tous les signaux de la boucle fermée soient bornés et que les erreurs de poursuite converge vers un petite voisinage de zéro. La loi de commande adaptative floue est composée de trois termes [139] [134], le terme adaptatif flou $u_{f,i} = \Theta_i^T \varphi_i(\hat{x}_s)$ conçu pour approximer une commande idéale inconnue basé sur le modèle dynamique du système $u_i^*(t)$, avec $\varphi_i(\hat{x}_s)$ le vecteur des fonctions de base floue, Θ_i^T est le vecteur des paramètres ajustable et \hat{x}_s est l'estimé du vecteur d'état. Le terme robuste borné $u_{r,i} = \hat{\delta}_i \tanh\left(\frac{\hat{e}_{2i}}{\epsilon_i}\right)$ conçu pour compenser les erreurs d'approximation floues $\delta^* = u_i^*(t) - \Theta_i^{*T} \varphi_i(\hat{x}_s)$. Finalement $u_{p,i} = \kappa_{2i} \hat{e}_{2i}$ le terme proportionnel dérivé.

Avant de procéder au contrôle robuste, nous définissons les erreurs de suivi comme suit:

$$\hat{e}_{1i} = x_{1i,r} - \hat{x}_{1i}, \quad i = 1, \dots, 6 \quad (5.2)$$

avec $x_{1i,r}$ le signal de référence pour l'attitude et la position du Quadrotor. Utilisation le contrôleur de position u_5 and u_6 (5.6), Les signaux de roulis et de tangage désirées peuvent être calculés comme suite

$$\theta_r = \text{atan}\left(\frac{u_5 \cos \hat{\psi} + u_6 \sin \hat{\psi}}{g}\right) \quad (5.3)$$

$$\phi_r = \text{atan}\left(\frac{u_5 \sin \hat{\psi} - u_6 \cos \hat{\psi}}{g} \cos \theta_r\right) \quad (5.4)$$

Les deuxièmes signaux d'erreurs de poursuite du backstepping sont définis comme

$$\hat{e}_{2i} = \hat{v}_i - \hat{x}_{2i} \quad (5.5)$$

avec $\hat{v}_i = \dot{x}_{1i,r} + \kappa_{1i} \hat{e}_{1i}$ et $\kappa_{1i} > 0$. Ensuite, nous introduisons l'algorithme de commande suivant:

$$u_i(t) = \Theta_i^T \varphi_i(\hat{x}_s) + \hat{\delta}_i \tanh\left(\frac{\hat{e}_{2i}}{\epsilon_i}\right) + \kappa_{2i} \hat{e}_{2i} \quad (5.6)$$

$$\dot{\Theta}_i = \gamma_i \hat{e}_{2i} \varphi_i(\hat{x}_s) - \sigma_{1i} \Theta_i \quad (5.7)$$

$$\dot{\hat{\delta}}_i = \eta_i \hat{e}_{2i} \tanh\left(\frac{\hat{e}_{2i}}{\epsilon_i}\right) - \sigma_{2i} \hat{\delta}_i \quad (5.8)$$

avec $\kappa_{2i} > 0$, $\epsilon_j > 0$, $\gamma_j > 0$, $\eta_j > 0$, $\sigma_{1i} > 0$ et $\sigma_{2i} > 0$ sont des paramètres de conception, et $\Theta_i(0) = 0$, $\hat{\delta}_i(0) = 0$.

Afin de compenser les perturbations extérieures (rafales de vent), un quatrième terme est ajouté à la loi de commande (5.6) $u_{w,i} = \beta_i \hat{w}_i$. Donc, la loi de commande (5.6) s'écrit

$$u_i(t) = \Theta_i^T \varphi_i(\hat{x}_s) + \hat{\delta}_i \tanh\left(\frac{\hat{e}_{2i}}{\epsilon_i}\right) + \kappa_{2i} \hat{e}_{2i} + \beta_i \hat{w}_i \quad (5.9)$$

avec: β_i une constante positive.

5.4 Synthèse de l'observateur de perturbation

L'observateur de perturbation étendu proposé est donné comme suit

$$\begin{cases} \dot{\hat{x}}_{1i} = \hat{x}_{2i} + k_{1i}(y_i - \hat{y}_i) \\ \dot{\hat{x}}_{2i} = f_i(\hat{x}_s) + g_i(\hat{x}_s)u_i(t) + \alpha_i \hat{w}_i + k_{2i}(y_i - \hat{y}_i) \\ \dot{\hat{w}}_i = \tau_i \chi(y_i - \hat{y}_i) - \sigma_{3i} \hat{w}_i \\ \hat{y}_i = \hat{x}_{1i} \end{cases} \quad (5.10)$$

avec k_{1i} , α_i , k_{2i} , τ_i , σ_{3i} des constantes positives. l'équation(5.10) peut être réécrite sous la forme suivante

$$\begin{cases} \dot{\hat{x}}_i = A_i \hat{x}_i + K_i y_i + B_i + G_i u_i(t) + E_i \hat{w}_i \\ \dot{\hat{w}}_i = \tau_i \chi(y_i - \hat{y}_i) - \sigma_{3i} \hat{w}_i \\ \hat{y}_i = C^T \hat{x}_i \end{cases} \quad (5.11)$$

avec

$$A_i = \begin{bmatrix} -k_{1i} & 1 \\ -k_{2i} & 0 \end{bmatrix}, \quad K_i = \begin{bmatrix} k_{1i} \\ k_{2i} \end{bmatrix}, \quad \hat{x}_i = \begin{bmatrix} \hat{x}_{1i} \\ \hat{x}_{2i} \end{bmatrix}$$

$$B_i = \begin{bmatrix} 0 \\ f_i(\hat{x}_s) \end{bmatrix}, \quad G_i = \begin{bmatrix} 0 \\ g_i(\hat{x}_s) \end{bmatrix}$$

$$E_i = \begin{bmatrix} 0 \\ \alpha_i \end{bmatrix}, \quad C_i = \begin{bmatrix} 1 \\ 0 \end{bmatrix}$$

Les vecteurs K_i sont choisies de tel sorte que A_i soit une matrice Hurwitz. Notons $Q_i^T = Q_i > 0$, il existe des matrices $P_i^T = P_i > 0$ symétriques et définies positives

solutions du système d'équations suivant

$$A_i^T P_i + P_i A_i = -2Q_i \quad (5.12)$$

Définissons l'erreur d'observation $s_i = x_i - \hat{x}_i$, l'erreur d'estimation $e_{yi} = y_i - \hat{y}_i$ et $\tilde{w}_i = w_i - \hat{w}_i$; donc à partir de (5.1) et (5.10), la dynamique de l'erreur d'observation s'écrit comme suit

$$\begin{cases} \dot{s}_i = A_i e_{xi} + \varepsilon_i + E_i \tilde{w}_i \\ \dot{\tilde{w}}_i = \tau_i \chi e_{yi} - \sigma_{3i} \hat{w}_i \\ e_{yi} = C^T s_i \end{cases} \quad (5.13)$$

avec $x_i = [x_{1i}, x_{2i}]^T$, $\varepsilon_i = [0, \tilde{f}_i + \tilde{g}_i]^T$, $\tilde{f}_i = f_i(x_s) - f_i(\hat{x}_s)$ et $\tilde{g}_i = g_i(x_s) - g_i(\hat{x}_s)$.

Hypothèse 5.7. Les erreurs d'estimation non linéaires \tilde{f}_i, \tilde{g}_i sont lipschitziennes et satisfait $|\tilde{f}_i(x_1) - \tilde{f}_i(x_2)| \leq \bar{f}_i |x_1 - x_2|$, $|\tilde{g}_i(x_1) - \tilde{g}_i(x_2)| \leq \bar{g}_i |x_1 - x_2|$, avec \bar{f}_i et \bar{g}_i sont deux constantes positive.

5.5 Analyse de stabilité

Considérons la fonction de Lyapunov candidate suivante

$$V = \frac{1}{2} \sum_{i=1}^6 \left(s_i^T P_i s_i + \hat{e}_{1i}^2 + \frac{1}{g_i(\hat{x}_s)} \hat{e}_{2i}^2 + \frac{1}{\gamma_i} \tilde{\Theta}_i^T \tilde{\Theta}_i + \frac{1}{\eta_i} \tilde{\delta}_i^2 + \frac{1}{\tau_i} \tilde{w}_i^2 \right) \quad (5.14)$$

avec $\tilde{\delta}_i = \delta_i^* - \hat{\delta}_i$, $\hat{\delta}_i$ est l'estimé de δ_i^* and $\tilde{\Theta}_i = \Theta_i^* - \Theta_i$ est l'erreur d'estimation, avec $\hat{e}_{1i} = x_{1i,r} - \hat{x}_{1i}$ et $\hat{e}_{2i} = \hat{v}_i - \hat{x}_{2i}$ avec $\hat{v}_i = \dot{x}_{1i,r} + \kappa_{1i} \hat{e}_{1i}$.

On utilisant l'équation (4.10), (5.12) et (5.13), la dérivée temporelle de V est

$$\begin{aligned} \dot{V} = \sum_{i=1}^6 \left(-s_i^T Q_i s_i + e_{xi}^T P_i (\varepsilon_i + E_i e_{wi}) - \kappa_{1i} \hat{e}_{1i}^2 + \hat{e}_{2i} [u_i^*(t) - u_i(t)] \right. \\ \left. + \frac{1}{\gamma_i} \tilde{\Theta}_i^T \dot{\tilde{\Theta}}_i + \frac{1}{\eta_i} \tilde{\delta}_i \dot{\tilde{\delta}}_i + \frac{1}{\tau_i} \tilde{w}_i \dot{\tilde{w}}_i \right) \end{aligned} \quad (5.15)$$

avec les inégalités suivantes

$$\begin{aligned} s_i^T P_i \varepsilon_i &\leq \frac{1}{2} \|s_i\|^2 + \frac{1}{2} \|P_i\|^2 (\tilde{f}_i^2 + \tilde{g}_i^2) \\ &\leq \frac{1}{2} \|s_i\|^2 + \frac{1}{2} \|P_i\|^2 (\bar{f}_i^2 + \bar{g}_i^2) \end{aligned} \quad (5.16)$$

L'équation (5.16) peut être bornée de la manière suivante

$$\begin{aligned} \dot{V} &\leq \sum_{i=1}^6 -s_i^T \left(Q_i - \frac{1}{2} I \right) s_i + \frac{1}{2} \|P_i\|^2 (\bar{f}_i^2 + \bar{g}_i^2) - \kappa_{1i} \hat{e}_{1i}^2 \\ &\quad + \hat{e}_{2i} \left(\Theta_i^{*T} \varphi_i(\hat{x}_s) + \delta_i^* - \Theta_i^T \varphi_i(\hat{x}_s) - \hat{\delta}_i \tanh \left(\frac{\hat{e}_{2i}}{\epsilon_i} \right) - \kappa_{2i} \hat{e}_{2i} - \beta_i \hat{w}_i \right) \\ &\quad + s_i^T P_i E_i \tilde{w}_i + \frac{1}{\gamma_i} \tilde{\Theta}_i^T \dot{\Theta}_i + \frac{1}{\eta_i} \tilde{\delta}_i \dot{\delta}_i + \frac{1}{\tau_i} \tilde{w}_i \dot{w}_i \end{aligned} \quad (5.17)$$

Les vecteurs des paramètres optimaux Θ_i^* , δ_i^* and w_i sont lentement variable dans le temps, par conséquent la dérivée de l'erreur d'estimation sera $\dot{\Theta}_i = -\dot{\Theta}_i$, $\dot{\delta}_i = -\dot{\delta}_i$ and $\dot{w}_i = -\dot{w}_i$. One have

$$\begin{aligned} \dot{V} &\leq \sum_{i=1}^6 -s_i^T \left(Q_i - \frac{1}{2} I \right) s_i + \frac{1}{2} \|P_i\|^2 (\bar{f}_i^2 + \bar{g}_i^2) - \kappa_{1i} \hat{e}_{1i}^2 - \kappa_{2i} \hat{e}_{2i}^2 \\ &\quad + \hat{e}_{2i} \tilde{\Theta}_i^T \varphi_i(\hat{x}_s) + s_i^T P_i E_i \tilde{w}_i + \hat{e}_{2i} \left(\delta_i^* - \hat{\delta}_i \tanh \left(\frac{\hat{e}_{2i}}{\epsilon_i} \right) - \beta_i \hat{w}_i \right) \\ &\quad - \frac{1}{\gamma_i} \tilde{\Theta}_i^T \dot{\Theta}_i - \frac{1}{\eta_i} \tilde{\delta}_i \dot{\delta}_i - \frac{1}{\tau_i} \tilde{w}_i \dot{w}_i \end{aligned} \quad (5.18)$$

En ajoutant et en substituant $\hat{e}_{2i} \tilde{\delta}_i \tanh \left(\frac{\hat{e}_{2i}}{\epsilon_i} \right)$ de (5.18), on obtient

$$\begin{aligned} \dot{V} &\leq \sum_{i=1}^6 -s_i^T \left(Q_i - \frac{1}{2} I \right) s_i + \frac{1}{2} \|P_i\|^2 (\bar{f}_i^2 + \bar{g}_i^2) - \kappa_{1i} \hat{e}_{1i}^2 - \kappa_{2i} \hat{e}_{2i}^2 \\ &\quad + \frac{1}{\gamma_i} \tilde{\Theta}_i^T \left(\gamma_i \hat{e}_{2i} \varphi_i(\hat{x}_s) - \dot{\Theta}_i \right) - \hat{e}_{2i} \hat{\delta}_i \tanh \left(\frac{\hat{e}_{2i}}{\epsilon_i} \right) + \frac{1}{\eta_i} \tilde{\delta}_i \left(\eta_i \hat{e}_{2i} \tanh \left(\frac{\hat{e}_{2i}}{\epsilon_i} \right) - \dot{\delta}_i \right) \\ &\quad - \delta_i^* \hat{e}_{2i} \tanh \left(\frac{\hat{e}_{2i}}{\epsilon_i} \right) + \hat{e}_{2i} \hat{\delta}_i \tanh \left(\frac{\hat{e}_{2i}}{\epsilon_i} \right) + |\hat{e}_{2i}| \delta_i^* - \hat{e}_{2i} \beta_i \hat{w}_i + \frac{1}{\tau_i} \tilde{w}_i \left(\tau_i s_i^T \underbrace{P_i E_i}_{C_X} - \dot{w}_i \right) \end{aligned} \quad (5.19)$$

On utilisant l'inégalité suivante

$$|\hat{e}_{2i}| - \hat{e}_{2i} \tanh\left(\frac{\hat{e}_{2i}}{\epsilon_i}\right) \leq \zeta \epsilon_i = \varsigma \quad (5.20)$$

et en remplaçant par les lois d'adaptations $\dot{\Theta}_i$ (5.7), $\dot{\delta}_i$ (5.8) ET \dot{w}_i (5.10) dans (5.19) avec la condition $P_i E_i = C \chi$, on obtient

$$\begin{aligned} \dot{V} \leq & \sum_{i=1}^6 -s_i^T \left(Q_i - \frac{1}{2} I \right) s_i - \kappa_{1i} \hat{e}_{1i}^2 - \kappa_{2i} \hat{e}_{2i}^2 + \frac{1}{2} \|P_i\|^2 (\bar{f}_i^2 + \bar{g}_i^2) \\ & + \frac{\sigma_{1i}}{\gamma_i} \tilde{\Theta}_i^T \Theta_i + \frac{\sigma_{2i}}{\eta_i} \tilde{\delta}_i \hat{\delta}_i + \frac{\sigma_{3i}}{\tau_i} \tilde{w}_i \hat{w}_i + \bar{\delta}_i \varsigma - \hat{e}_{2i} \beta_i \hat{w}_i \end{aligned} \quad (5.21)$$

avec les inégalités suivantes

$$\frac{\sigma_{1i}}{\gamma_i} \tilde{\Theta}_i^T \Theta_i \leq -\frac{\sigma_{1i}}{2\gamma_i} \tilde{\Theta}_i^T \tilde{\Theta}_i + \frac{\sigma_{1i}}{2\gamma_i} \Theta_i^{*T} \Theta_i^* \quad (5.22)$$

$$\frac{\sigma_{2i}}{\eta_i} \tilde{\delta}_i \hat{\delta}_i \leq -\frac{\sigma_{2i}}{2\eta_i} \tilde{\delta}_i^2 + \frac{\sigma_{2i}}{2\eta_i} |\delta_i^*|^2 \quad (5.23)$$

$$\frac{\sigma_{3i}}{\tau_i} e_{wi} \hat{w}_i \leq -\frac{\sigma_{3i}}{2\tau_i} e_{wi}^2 + \frac{\sigma_{3i}}{2\tau_i} |w_i^*|^2 \quad (5.24)$$

$$-\beta_i \hat{e}_{2i} \hat{w}_i \leq -\frac{\beta_i}{2} \hat{e}_{2i}^2 + \frac{\beta_i}{2} |w_i^*|^2 \quad (5.25)$$

Par conséquent, (5.21) peut être réécrit sous la forme suivante

$$\begin{aligned} \dot{V} \leq & \sum_{i=1}^6 -s_i^T \left(Q_i - \frac{1}{2} I \right) s_i - \kappa_{1i} \hat{e}_{1i}^2 - \kappa_{2i} \hat{e}_{2i}^2 + \frac{1}{2} \|P_i\|^2 (\bar{f}_i^2 + \bar{g}_i^2) \\ & - \frac{\sigma_{1i}}{2\gamma_i} \tilde{\Theta}_i^T \tilde{\Theta}_i + \frac{\sigma_{1i}}{2\gamma_i} \Theta_i^{*T} \Theta_i^* - \frac{\sigma_{2i}}{2\eta_i} \tilde{\delta}_i^2 + \frac{\sigma_{2i}}{2\eta_i} |\delta_i^*|^2 \\ & + \bar{\delta}_i \varsigma - \frac{\beta_i}{2} \hat{e}_{2i}^2 + \frac{\beta_i}{2} |w_i^*|^2 - \frac{\sigma_{3i}}{2\tau_i} e_{wi}^2 + \frac{\sigma_{3i}}{2\tau_i} |w_i^*|^2 \end{aligned} \quad (5.26)$$

Définissant

$$c = \min \{ 2 \min(\lambda_{\min}(Q_i) - 1/2), \sigma_{1i}, \sigma_{2i}, \sigma_{2i}, 2\kappa_{1i}, 2\kappa_{2i}, \beta_i \}$$

donc, (5.26) s'écrit

$$\dot{V} \leq -cV + \frac{\sigma_{1i}}{2\gamma_i} \Theta_i^{*T} \Theta_i^* + \frac{\sigma_{2i}}{2\eta_i} \tilde{\delta}_i^2 + \frac{\sigma_{3i}}{2\tau_i} |w_i^*|^2 + \frac{1}{2} \|P_i\|^2 (\bar{f}_i^2 + \bar{g}_i^2) + \frac{\beta_i}{2} |w_i^*|^2 + \bar{\delta}_i \varsigma \quad (5.27)$$

Comme les vecteurs des paramètres Θ_i^* , δ_i^* et \hat{w}_i sont supposés bornés, nous pouvons donc définir une borne constante et positive $\bar{\rho}$, telle que

$$\bar{\rho} = \sup_t \left(\sum_{i=1}^6 + \frac{\sigma_{1i}}{2\gamma_i} \Theta_i^{*T} \Theta_i^* + \frac{\sigma_{2i}}{2\eta_i} \bar{\delta}_i^2 + \frac{\sigma_{3i}}{2\tau_i} |w_i^*|^2 + \frac{1}{2} \|P_i\|^2 (\bar{f}_i^2 + \bar{g}_i^2) + \frac{\beta_i}{2} |w_i^*|^2 + \bar{\delta}_i \varsigma \right) \quad (5.28)$$

avec ce choix, (5.27) devient

$$\dot{V} \leq -cV + \bar{\rho} \quad (5.29)$$

Ainsi la stabilité du système global contrôleur/observateur peut être prouvée à travers le théorème suivant

Théorème 5.3. *Considérons le système non linéaire du Quadrotor (5.1). Supposons que l'hypothèse 5.7 est satisfaites. Alors, la loi de commande (5.9) avec l'observateur de perturbation (5.10) garantissent la la bornitude de tous les signaux de la boucle fermée ainsi que la convergence des erreurs \hat{e}_{1i} et s_i vers un petit voisinage autour de zéro.*

Prouve. A partir de (5.29), on déduit

$$V(t) \leq V(0)e^{-ct} + \frac{\bar{\rho}}{c} \quad (5.30)$$

A partir de (5.30), on conclut que pour $V \geq \bar{\rho}/c$, nous avons $\dot{V} \leq 0$. Donc, en se basant sur le théorème de Lyapunov, les signaux en boucle fermée $\hat{e}_{1i}(t)$, $\hat{e}_{2i}(t)$, $s_i(t)$, $\tilde{\Theta}_i(t)$, $\tilde{\delta}_i$, \tilde{w}_i et $u_i(t)$, ainsi que les états du système x_{1i} , x_{2i} , \hat{x}_{1i} et \hat{x}_{2i} ; en plus, pour $\mu \geq \sqrt{\bar{\rho}/c}$, il existe un temps $T > 0$, tel que $|\hat{e}_{2i}(t)| \leq \bar{\mu}$ pour tout $t \geq T$. Afin d'assurer la convergence de l'erreur de poursuite vers un petit voisinage de zéro, il suffit de choisir des valeurs appropriées pour les paramètres γ_i , η_i , ϵ_i , σ_{1i} , σ_{2i} , σ_{3i} , β_i , κ_{1i} et κ_{2i} . Par conséquent, il est possible de rendre $\sqrt{\bar{\rho}/c}$ aussi petit que souhaité. Donc, les erreurs de poursuite convergent vers des ensembles résiduels tels que $|\hat{e}_{1i}(t)| \leq 2\alpha_i \bar{\mu}$. ▲

5.6 Résultat de simulation

La commande adaptative floue basé sur l'observateur de perturbation proposée, est appliqué sur le Quadrotor et validé par simulations sous *Matlab* pour vérifier la robustesse et les performances atteintes par cette approche de commande pour le suivi de trajectoire et le rejet des perturbations externes. Le modèle de rafale de vent utilisé pour générer la vitesse du vent le long des trois axes est le même modèle utilisé dans le chapitre précédent ainsi que les paramètres de ce modèle.

Les paramètres de conception sont choisis tel que $\kappa_{11} = 0.3, \kappa_{12} = 0.4, \kappa_{13} = 1, \kappa_{14} = 3, \kappa_{15} = 4, \kappa_{16} = 4, k_{11} = 10, k_{12} = 10, k_{13} = 8, k_{14} = 10, k_{15} = 10, k_{16} = 10, \kappa_{21} = 0.3, \kappa_{22} = 0.4, \kappa_{23} = 2, \kappa_{24} = 6, \kappa_{25} = 8, \kappa_{26} = 8, k_{21} = 18, k_{22} = 18, k_{23} = 16, k_{24} = 18, k_{25} = 18, k_{26} = 18$. Les gains de la loi d'adaptation floue sont choisis comme suit $\gamma_i = 0.9$ ($i = 1, 2, 3, 5, 6$), $\gamma_4 = 1.8$, $\sigma_{1i} = 0.01$ et les gains du terme robuste $\epsilon_i = 0.001, \eta_i = 0.03$ avec $\sigma_{2i} = 0.01$, et $\tau_i = 35, \sigma_{3i} = 0.001, \beta_i = 1$.

Les valeurs initiales pour la position et les angles d'Euler sont $x_{11}(0) = 0, x_{12}(0) = 0, x_{13}(0) = 0, x_{14}(0) = 0, x_{15}(0) = 0, x_{16}(0) = 0, \hat{x}_{11}(0) = 0, \hat{x}_{12}(0) = 0, \hat{x}_{13}(0) = 0.179, \hat{x}_{14}(0) = 0, \hat{x}_{15}(0) = 0, \hat{x}_{16}(0) = 0$, et les valeurs initiales des lois d'adaptation paramétrique $\hat{\epsilon}_i(0) = 0, \Theta_i(0) = [0_{9 \times 1}]$. Les vitesses linéaires et angulaires initiales sont choisies égales à zéro.

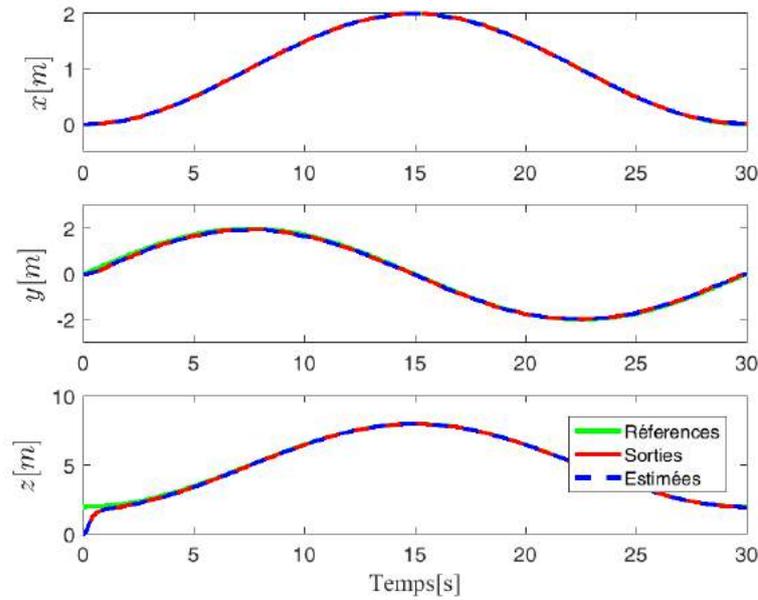


Figure 5.1: Suivi de position avec estimation de perturbations externes

Les résultats de la simulation pour le suivi des trajectoires de cette trajectoire désirée sont présentés sur les figures 5.1-5.4. La trajectoire désirée en 3D et la trajectoire de Quadrotor générée par la commande adaptative floue sont illustrées sur la figure 1, tandis que les réponses de suivi de la position et de l'attitude du Quadrotor sont présentées par la figure 5.2 et figure 5.3 respectivement. On peut remarquer que le suivi est réalisé avec une erreur nulle et le rejet de l'effet de rafale de vent pour toutes les sorties de Quadrotor. Figure 5.3 montre comment les angles de roulis et de tangage changent leurs valeurs afin d'assurer les performances de suivi appropriées pour le mouvement x et y . Comme

représenté sur la figure 5.4, la stratégie de commande proposée génère des signaux de d'entrée lisses et bornées.

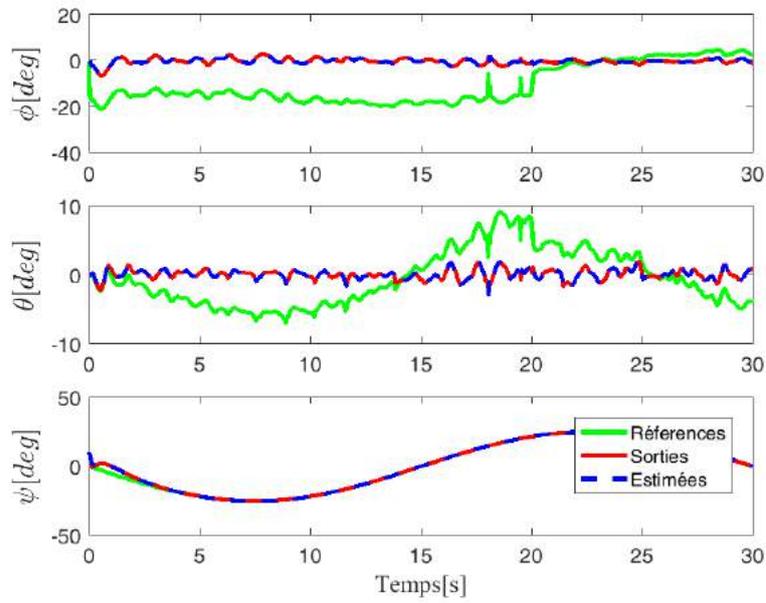


Figure 5.2: Suivi d'attitudes avec estimation de perturbations externes

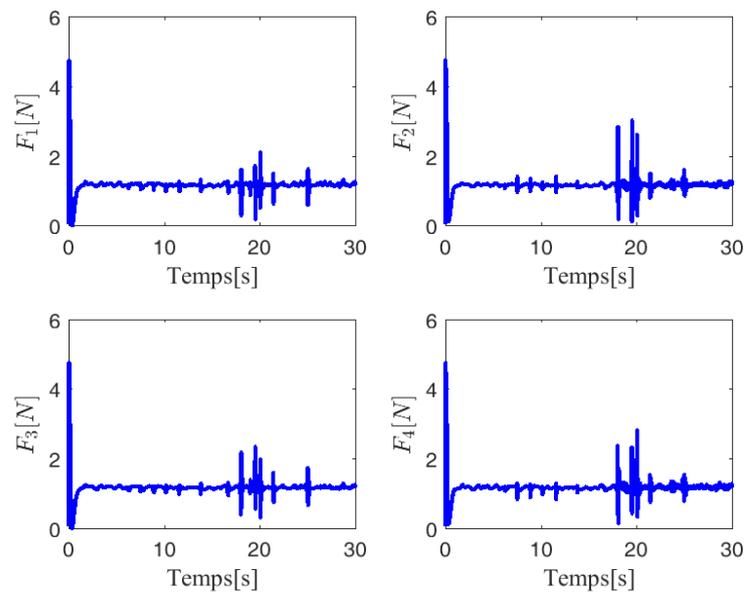


Figure 5.3: Entrées de commande

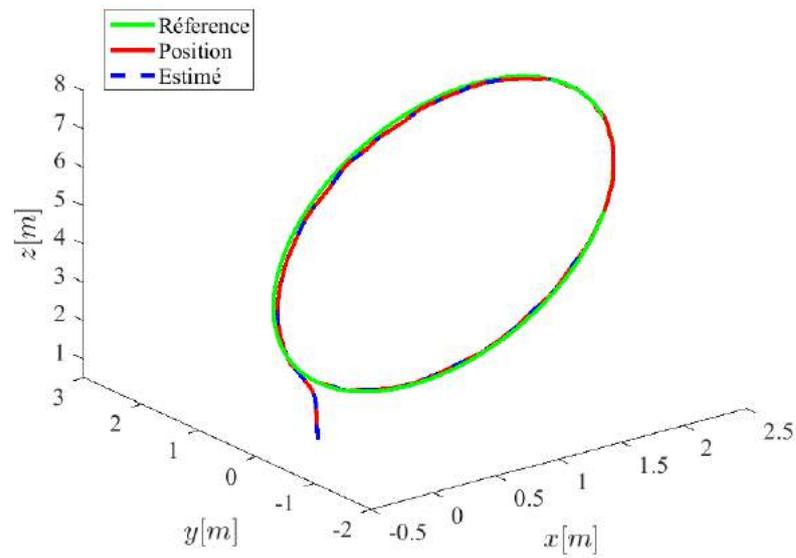
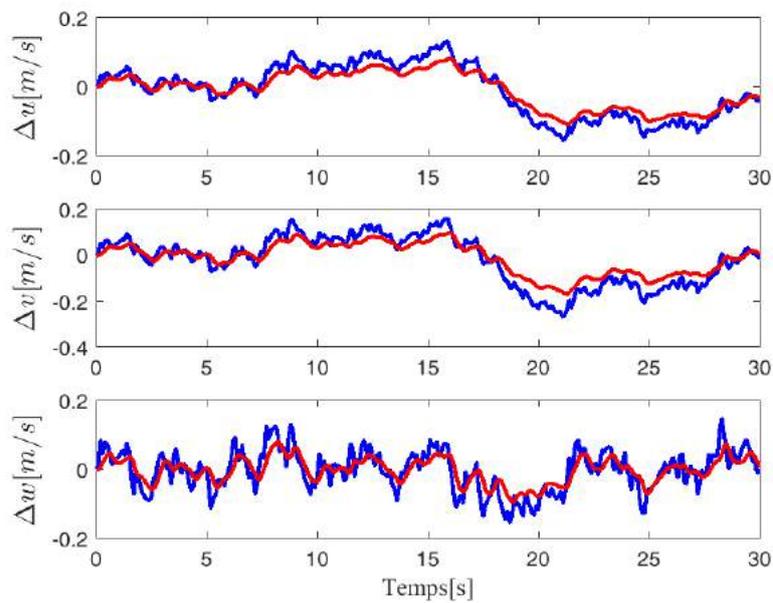


Figure 5.4: Position en 3D avec estimation de perturbations externes

Figure 5.5: Perturbations externes, \hat{w}_i (—), w_i (—)

La figure 5.5 représente la vitesse du vent réelle et estimée le long des trois axes (x, y, z) , comme le montrent ces figures nous avons obtenu de bonnes performances d'estimation, ce qui permet au contrôleur de compenser les perturbations externes.

5.7 Conclusion

Le chapitre 5 a proposé une commande adaptative floue basée sur un observateur de perturbation adaptative pour une classe des systèmes non linéaires sous actionnées. En premier temps, nous avons introduit un observateur de perturbation adaptatif, permettant d'estimer les perturbations externes telles que les rafales de vent et les frottements aérodynamiques. Ensuite, nous sommes intéressés ensuite à la conception d'une loi de commande robuste adaptative floue basée sur l'observateur de perturbation, l'estimé fourni par l'observateur est employé afin de compenser les effets des rafales de vent sur le système volant le Quadrotor. Il est important de noter que cette approche n'exige pas la connaissance du modèle dynamique du système et que la loi de commande et ses mécanismes d'adaptation sont simples à implémenter. Cette approche garantit, sous certaines hypothèses physiquement acceptables, la bornitude de tous les signaux de la boucle fermée et la convergence de l'erreur de poursuite à un petit voisinage autour de zéro.

Les performances de cette stratégie de commande ont été étudiées et examinées à travers l'application de cette approche sur le Quadrotor sous *Matlab*.

Conclusion générale et perspectives

Dans la dernière décennie, le Quadrotor est devenu une plate-forme populaire dans la recherche scientifique. En raison de sa simplicité de conception et de manœuvre, les Quadrotors ont de nombreuses applications civiles. Les derniers sont des robots volants qui utilisent la portance générée par quatre moteurs, et pour cette raison, une grande partie de leur énergie disponible est consommée par les moteurs afin de maintenir le véhicule dans l'espace. Les applications potentielles de Quadrotor tels que la photographie et la surveillance des agricultures nécessite une grande autonomie, mais la technologie des batteries LiPo disponible actuellement ne donne pas l'autonomie nécessaire pour la mission du drone.

Le problème fondamental des drones miniatures à voilure tournante est le vol sécurisé, robuste et fiable du véhicule dans diverses circonstances. La petite taille de ces véhicules pose un ensemble de défis pour les utilisateurs potentiels et les concepteurs de ces véhicules. La taille décroissante de système des capteurs embarqués injecte plus de bruit dans les algorithmes de commande. La petite taille du châssis du véhicule rend le système plus sensible aux perturbations de l'environnement. Ces facteurs doivent être considérés et atténués lors de la conception de ces plate-formes pour un vol robuste et fiable. Dans la littérature au cours de la dernière décennie, il y a eu de nombreux travaux dans le développement de méthodes de commande linéaire pour les drones miniatures. Cependant, les progrès récents dans la technologie des calculateurs embarqués ont permis de mettre en œuvre des algorithmes non linéaires plus avancés à bord de ces systèmes. Ces méthodes non linéaires donnent de meilleurs résultats mais supposent une connaissance parfaite de la dynamique du véhicule.

Cette thèse a présenté des contributions à l'état de l'art dans la génération de trajectoire optimale en terme d'énergie du Quadrotor, l'optimisation d'énergie et la commande du Quadrotor. Dans le chapitre 3, nous avons traité le problème de limitation de l'endurance de vol d'un Quadrotor. Nous avons décrit un modèle énergétique et deux problèmes de commande optimale pour calculer l'énergie minimale ainsi que le temps. La résolution des deux problèmes permet de trouver les entrées de commande et la trajectoire du véhicule qui minimisent l'énergie consommée et le temps durant la mission du drone.

Dans le chapitre 4, nous avons présenté deux approche de commande avec et sans observateur d'état. Les lois de commande proposées permettent d'améliorer les performances de poursuite de trajectoire pour le Quadrotor et atténuer les perturbation internes et externes. Dans le chapitre 5, un autre algorithme de commande robuste basé sur l'observateur de perturbation est introduit. Il permet l'estimation des perturbations

externes variable dans le temps dans le but de les compenser pour atténuer leurs effets.

En extension des travaux présentés dans cette thèse, plusieurs pistes peuvent être explorées et développées. On peut révéler quelques idées à développer dans le futur.

La première piste à explorer est d'intégrer l'aérodynamique des hélices dans le modèle énergétique pour avoir un modèle proche de la réalité. L'optimisation de taux de vieillissement de la batterie pour augmenter la durée de vie de la batterie est une piste importante.

Une deuxième idée à développer est de formuler un problème d'optimisation multi-objectives pour minimiser l'énergie et le taux de vieillissement de la batterie en même temps et augmenter l'énergie économisée.

L'amélioration de la stabilité obtenue par les approches de commande présentées dans cette thèse peuvent être envisageables et ce, pour avoir une stabilité asymptotique au lieu d'une stabilité bornée.

Il serait très intéressant d'implémenter les algorithmes de génération de trajectoire et les algorithmes de commande sur une plate-forme de Quadrotor afin de les valider.

Annexe A

Codes Matlab

Dans cette section nous présentons les codes Matlab utilisés dans la simulation pour les chapitres 1 et 2.

A.1 Codes Matlab chapitre 1

Exemple commande adaptative floue indirecte (section 1.5.1).

Le fichier principale main.m.

```
1 close all , clear all ; clc
2 global l m g mc gamma1
3 global c1 c2 Sigma gamma2
4 %% Paramètres %%
5     l = 0.5 ;
6     m = 0.1 ;
7     g = 9.8 ;
8     mc = 1 ;
9     c1=pi/6;
10    c2=pi/12;
11    Sigma=pi/24;
12    gamma1=50;
13    gamma2=1;
14    tf=20;
15    pas=0.001 ;
16 %% Initialisation %%
17 x=[pi/60 0]'; u=5; ea=0;
18 tt=[]; xx1=[]; yy=[]; yyr=[];
19 K=[1 2]; thetag=zeros(1,25);
20 thetag=zeros(1,25);
21 for t=0:pas:tf
22     x1=x(1);
23     x2=x(2);
24     xx1=[xx1 x1];
25     tt=[tt t];
26     yr=pi/30*sin(t);
27     yrp=pi/30*cos(t);
28     yrpp=-(pi/30)*sin(t);
29     yyr=[yyr yr];
30     e=yr-x1;
31     ep=(e-ea)/pas;
32     ea=e;
33     E=[e;ep];
```

```

34 %% calcul de la commande %%
35 omega1=exp(-((x1-c2)/Sigma)^2)*exp(-((x2-c2)/Sigma)^2);
36 omega2=exp(-((x1-c2)/Sigma)^2)*exp(-((x2-c1)/Sigma)^2);
37 omega3=exp(-((x1-c2)/Sigma)^2)*exp(-((x2)/Sigma)^2);
38 omega4=exp(-((x1-c2)/Sigma)^2)*exp(-((x2+c2)/Sigma)^2);
39 omega5=exp(-((x1-c2)/Sigma)^2)*exp(-((x2+c1)/Sigma)^2);
40 omega6=exp(-((x1-c1)/Sigma)^2)*exp(-((x2-c2)/Sigma)^2);
41 omega7=exp(-((x1-c1)/Sigma)^2)*exp(-((x2-c1)/Sigma)^2);
42 omega8=exp(-((x1-c1)/Sigma)^2)*exp(-((x2)/Sigma)^2);
43 omega9=exp(-((x1-c1)/Sigma)^2)*exp(-((x2+c2)/Sigma)^2);
44 omega10=exp(-((x1-c1)/Sigma)^2)*exp(-((x2+c1)/Sigma)^2);
45 omega11=exp(-((x1)/Sigma)^2)*exp(-((x2-c2)/Sigma)^2);
46 omega12=exp(-((x1)/Sigma)^2)*exp(-((x2-c1)/Sigma)^2);
47 omega13=exp(-((x1)/Sigma)^2)*exp(-((x2)/Sigma)^2);
48 omega14=exp(-((x1)/Sigma)^2)*exp(-((x2+c2)/Sigma)^2);
49 omega15=exp(-((x1)/Sigma)^2)*exp(-((x2+c1)/Sigma)^2);
50 omega16=exp(-((x1+c1)/Sigma)^2)*exp(-((x2-c2)/Sigma)^2);
51 omega17=exp(-((x1+c1)/Sigma)^2)*exp(-((x2-c1)/Sigma)^2);
52 omega18=exp(-((x1+c1)/Sigma)^2)*exp(-((x2)/Sigma)^2);
53 omega19=exp(-((x1+c1)/Sigma)^2)*exp(-((x2+c2)/Sigma)^2);
54 omega20=exp(-((x1+c1)/Sigma)^2)*exp(-((x2+c1)/Sigma)^2);
55 omega21=exp(-((x1+c2)/Sigma)^2)*exp(-((x2-c2)/Sigma)^2);
56 omega22=exp(-((x1+c2)/Sigma)^2)*exp(-((x2-c1)/Sigma)^2);
57 omega23=exp(-((x1+c2)/Sigma)^2)*exp(-((x2)/Sigma)^2);
58 omega24=exp(-((x1+c2)/Sigma)^2)*exp(-((x2+c2)/Sigma)^2);
59 omega25=exp(-((x1+c2)/Sigma)^2)*exp(-((x2+c1)/Sigma)^2);
60 sumomega=omega1+omega2+omega3+omega4+omega5+omega6+omega7+...
61         omega8+omega9+omega10+omega11+omega12+omega13+...
62         omega14+omega15+omega16+omega17+omega18+omega19+...
63         omega20+omega21+omega22+omega23+omega24+omega25;
64 mu1=omega1/sumomega; mu6=omega6/sumomega;
65 mu2=omega2/sumomega; mu7=omega7/sumomega;
66 mu3=omega3/sumomega; mu8=omega8/sumomega;
67 mu4=omega4/sumomega; mu9=omega9/sumomega;
68 mu5=omega5/sumomega; mu10=omega10/sumomega;
69 mu11=omega11/sumomega; mu16=omega16/sumomega;
70 mu12=omega12/sumomega; mu17=omega17/sumomega;
71 mu13=omega13/sumomega; mu18=omega18/sumomega;
72 mu14=omega14/sumomega; mu19=omega19/sumomega;
73 mu15=omega15/sumomega; mu20=omega20/sumomega;
74 mu21=omega21/sumomega;
75 mu22=omega22/sumomega;
76 mu23=omega23/sumomega;

```

```

77     mu24=omega24/sumomega;
78     mu25=omega25/sumomega;
79     input1=[e ep mu1 mu2 mu3 mu4 mu5 mu6 mu7 mu8 mu9 mu10 ...
80             mu11 mu12 mu13mu14 mu15 mu16 mu17 mu18 mu19 ...
81             mu20 mu21 mu22 mu23 mu24 mu25];
82     input2=[e ep mu1 mu2 mu3 mu4 mu5 mu6 mu7 mu8 mu9 mu10 ...
83             mu11 mu12 mu13 mu14 mu15 mu16 mu17 mu18 mu19 ...
84             mu20 mu21 mu22 mu23 mu24 mu25 u];
85     thetAFP=adaptation_f(input1);
86     thetAF=thetAF+pas*thetAFP;
87     thetAG=adaptation_g(input2);
88     thetAG=thetAG+pas*thetAG;
89     phi=[mu1 mu2 mu3 mu4 mu5 mu6 mu7 mu8 mu9 mu10 mu11 mu12 ...
90          mu13 mu14 mu15 mu16 mu17 mu18 mu19 mu20 mu21 mu22 ...
91          mu23 mu24 mu25];
92     f=thetAF*phi';
93     G=thetAG*phi';
94     u=(1/G)*(-f+K*E+yrpp);
95
96     %% itegration du modele%%
97     input3=[x1 x2 u];
98     F=pendule(input3);
99     x=x+pas*F';
100 end
101 figure
102 plot(tt,yyr,'r',tt,xx1,'b');

```

Les trois fonctions utilisés dans le main.m sont pendule.m représente le modèle du système, adaptation_f.m donne l'estimation des paramètres θ_f et adaptation_g.m représente l'estimation des paramètres θ_g .

```

1  function [output]=pendule(input)
2  global l m g mc
3  %% les entrées %%
4  x1 = input(1);
5  x2 = input(2);
6  u = input(3);
7  %% les sorties %%
8  output(1) = x2;
9  output(2) = (g*sin(x1)-(m*l*x2^2*cos(x1)*sin(x1)/(mc+m))/...
10              (1*(4/3-(m*(cos(x1))^2)/(mc+m)))+(((cos(x1)/...
11              (mc+m)))/(1*(4/3-(m*(cos(x1))^2)/(mc+m))))*u;

```

Exemple commande adaptative floue directe (section 1.5.2)

```
1 function output=adaptation_f(input)
2 global gamma1
3 e=input(1);
4 ep=input(2);
5 mu1=input(3);
6 mu2=input(4);
7 mu3=input(5);
8 mu4=input(6);
9 mu5=input(7);
10 mu6=input(8);
11 mu7=input(9);
12 mu8=input(10);
13 mu9=input(11);
14 mu10=input(12);
15 mu11=input(13);
16 mu12=input(14);
17 mu13=input(15);
18 mu14=input(16);
19 mu15=input(17);
20 mu16=input(18);
21 mu17=input(19);
22 mu18=input(20);
23 mu19=input(21);
24 mu20=input(22);
25 mu21=input(23);
26 mu22=input(24);
27 mu23=input(25);
28 mu24=input(26);
29 mu25=input(27);
30 E=[e ep]; b=[0;1];
31 P=[15 5;5 5];
32 output(1) = -gamma1*E*P*b*mu1;
33 output(2) = -gamma1*E*P*b*mu2;
34 output(3) = -gamma1*E*P*b*mu3;
35 output(4) = -gamma1*E*P*b*mu4;
36 output(5) = -gamma1*E*P*b*mu5;
37 output(6) = -gamma1*E*P*b*mu6;
38 output(7) = -gamma1*E*P*b*mu7;
39 output(8) = -gamma1*E*P*b*mu8;
40 output(9) = -gamma1*E*P*b*mu9;
41 output(10) = -gamma1*E*P*b*mu10;
```

```
42 output(11) = -gamma1*E*P*b*mu11;
43 output(12) = -gamma1*E*P*b*mu12;
44 output(13) = -gamma1*E*P*b*mu13;
45 output(14) = -gamma1*E*P*b*mu14;
46 output(15) = -gamma1*E*P*b*mu15;
47 output(16) = -gamma1*E*P*b*mu16;
48 output(17) = -gamma1*E*P*b*mu17;
49 output(18) = -gamma1*E*P*b*mu18;
50 output(19) = -gamma1*E*P*b*mu19;
51 output(20) = -gamma1*E*P*b*mu20;
52 output(21) = -gamma1*E*P*b*mu21;
53 output(22) = -gamma1*E*P*b*mu22;
54 output(23) = -gamma1*E*P*b*mu23;
55 output(24) = -gamma1*E*P*b*mu24;
56 output(25) = -gamma1*E*P*b*mu25;
```

```
1 function output=adaptation_g(input)
2 global gamma2
3 e=input(1);
4 ep=input(2);
5 mu1=input(3);
6 mu2=input(4);
7 mu3=input(5);
8 mu4=input(6);
9 mu5=input(7);
10 mu6=input(8);
11 mu7=input(9);
12 mu8=input(10);
13 mu9=input(11);
14 mu10=input(12);
15 mu11=input(13);
16 mu12=input(14);
17 mu13=input(15);
18 mu14=input(16);
19 mu15=input(17);
20 mu16=input(18);
21 mu17=input(19);
22 mu18=input(20);
23 mu19=input(21);
24 mu20=input(22);
25 mu21=input(23);
26 mu22=input(24);
```

```
27 mu23=input (25) ;
28 mu24=input (26) ;
29 mu25=input (27) ;
30 u=input (28) ;
31 E=[e ep];b=[0;1];
32 P=[15 5;5 5];
33 output (1) = -gamma2*E*P*b*mu1*u;
34 output (2) = -gamma2*E*P*b*mu2*u;
35 output (3) = -gamma2*E*P*b*mu3*u;
36 output (4) = -gamma2*E*P*b*mu4*u;
37 output (5) = -gamma2*E*P*b*mu5*u;
38 output (6) = -gamma2*E*P*b*mu6*u;
39 output (7) = -gamma2*E*P*b*mu7*u;
40 output (8) = -gamma2*E*P*b*mu8*u;
41 output (9) = -gamma2*E*P*b*mu9*u;
42 output (10) = -gamma2*E*P*b*mu10*u;
43 output (11) = -gamma2*E*P*b*mu11*u;
44 output (12) = -gamma2*E*P*b*mu12*u;
45 output (13) = -gamma2*E*P*b*mu13*u;
46 output (14) = -gamma2*E*P*b*mu14*u;
47 output (15) = -gamma2*E*P*b*mu15*u;
48 output (16) = -gamma2*E*P*b*mu16*u;
49 output (17) = -gamma2*E*P*b*mu17*u;
50 output (18) = -gamma2*E*P*b*mu18*u;
51 output (19) = -gamma2*E*P*b*mu19*u;
52 output (20) = -gamma2*E*P*b*mu20*u;
53 output (21) = -gamma2*E*P*b*mu21*u;
54 output (22) = -gamma2*E*P*b*mu22*u;
55 output (23) = -gamma2*E*P*b*mu23*u;
56 output (24) = -gamma2*E*P*b*mu24*u;
57 output (25) = -gamma2*E*P*b*mu25*u;
```

A.2 Codes Matlab chapitre 2

Simulation en boucle ouverte du Quadrotor.

Le fichier principale main.m.

```
1 close all , clear all ; clc
2 %% Paramètres %%
3 l = 0.175 ;
4 m = 1.3 ;
```

```

5  g = 9.8066;
6  Ix = 0.081;
7  Iy = 0.081;
8  Iz = 0.142;
9  nB = 2;
10 mB = 0.0055;
11  r = 0.12;
12  e = 0.004;
13  Jm = 4.9*(10^-6);
14  J1 = (1/4)*nB*mB*(r-e)^2;
15  Ir = J1+Jm ;
16  CT = 0.0048;
17  CQ = 2.3515*(10^-4);
18  Rho = 1.225;
19  A = pi*r^2;
20  b = CT*Rho*A*r^2;
21  k = CQ*Rho*A*r^3;
22  tf=5;
23  pas=0.001;
24  %% initialisation %%
25  x=zeros(12,1);u=zeros(4,1);
26  tt=[];xx=[];uu=[];
27  for t=0:pas:tf
28      xx=[xx x];
29      tt=[tt t];
30      uu=[uu u];
31      %% entrées de commande pour décollage vertical
32      w1=50*sin(0.9*t)+912;%712
33      w2=50*sin(0.9*t)+912;
34      w3=50*sin(0.9*t)+912;%1112
35      w4=50*sin(0.9*t)+912;
36      u=[w1 w2 w3 w4]';
37      %% entrées de commande pour translation horizontale
38      %w1=50*sin(0.9*t)+712;
39      %w2=50*sin(0.9*t)+912;
40      %w3=50*sin(0.9*t)+1112;
41      %w4=50*sin(0.9*t)+912;
42      %u=[w1 w2 w3 w4]';
43      %% itegration du modele%%
44      input=[x;u];
45      F=quadrotor(input);
46      x=x+pas*F';
47      x(1) = x(1)*(abs(x(1))<=pi/2)+pi/2*(abs(x(1))>pi/2)*sign(x(1));

```

```

48     x(3) = x(3)*(abs(x(3))≤pi/2)+pi/2*(abs(x(3))>pi/2)*sign(x(3));
49     x(5) = x(5)*(abs(x(5))≤pi)+pi*(abs(x(5))>pi)*sign(x(5));
50 end
51 figure
52 plot(tt,xx(1,:), 'b',tt,xx(3,:), 'r',tt,xx(5,:), 'k', 'LineWidth',2);
53 xlabel( '\fontsize{16} Temps [s] ')
54 ylabel( '\fontsize{16} Attitude [rad] ')
55 legend( '\fontsize{14}\phi', '\fontsize{14}\theta', '\fontsize{14}\psi ')
56 figure
57 plot(tt,xx(7,:), 'b',tt,xx(9,:), 'r',tt,xx(11,:), 'k', 'LineWidth',2);
58 xlabel( '\fontsize{16} Temps [s] ')
59 ylabel( '\fontsize{16} Position [rad] ')
60 legend( '\fontsize{14} x', '\fontsize{14} y', '\fontsize{14} z ')

```

Le modèle dynamique est donné par la fonction quadrotor.m.

```

1 function [output]=quadrotor(input)
2 global l m g k b Ix Iy Iz Ir
3 a1 = (Iy-Iz)/Ix;
4 a2 = -Ir/Ix;
5 a3 = (Iz-Ix)/Iy;
6 a4 = Ir/Iy;
7 a5 = (Ix-Iy)/Iz;
8 b1 = 1/Ix;
9 b2 = 1/Iy;
10 b3 = 1/Iz;
11 %% les entrees de la fonction %%
12 phi=input(1);
13 phip=input(2);
14 theta=input(3);
15 thetap=input(4);
16 psi=input(5);
17 psip=input(6);
18 x=input(7);
19 xp=input(8);
20 y=input(9);
21 yp=input(10);
22 z=input(11);
23 zp=input(12);
24 w1=input(13);
25 w2=input(14);
26 w3=input(15);

```

```
27 w4=input(16);
28 %% les entrées de commande %%
29 ux=cos(psi)*sin(theta)*cos(phi)+sin(psi)*sin(phi);
30 uy=sin(psi)*sin(theta)*cos(phi)-cos(psi)*sin(phi);
31 u1=b*(w1^2+w2^2+w3^2+w4^2);
32 u2=b*1*(w4^2-w2^2);
33 u3=b*1*(w3^2-w1^2);
34 u4=k*(w1^2-w2^2+w3^2-w4^2);
35 Wr=w1-w2+w3-w4;
36 %% les sorties de la fonction %%
37 output(1) = phip;
38 output(2) = b1*u2+a1*thetap*psip+a2*Wr*thetap;
39 output(3) = thetap;
40 output(4) = b2*u3+a3*phip*psip+a4*Wr*phip;
41 output(5) = psip;
42 output(6) = b3*u4+a5*phip*thetap;
43 output(7) = xp;
44 output(8) = (ux/m)*u1;
45 output(9) = yp;
46 output(10) = (uy/m)*u1;
47 output(11) = zp;
48 output(12) = (cos(phi)*cos(theta)/m)*u1-g;
```

Références

- [1] “Ascending technologies.” <http://www.asctec.de>. Accessed: 2017-05-11.
- [2] “Draganfly innovations inc.” <http://www.draganfly.com>. Accessed: 2017-05-17.
- [3] “Mikrokopter.” <http://http://www.mikrokopter.de>. Accessed: 2017-05-28.
- [4] “Dji phantom 2.” <http://www.dji.com/fr/phantom-2>. Accessed: 2016-02-25.
- [5] B. Samir, *Design and Control of Quadrotors with Application to Autonomous Flying*. PhD thesis, Ecole Polytechnique Federale de Lausanne, 2007.
- [6] G. M. Hoffmann, S. L. Waslander, M. P. Vitus, H. Huang, J. Gillula, V. Pradeep, and C. J. Tomlin, “Stanford testbed of autonomous rotorcraft for multi-agent control,” in *Intelligent Robots and Systems (IROS), IEEE/RSJ International Conference on*, pp. 404–405, Oct 2009.
- [7] P. Pounds, R. Mahony, and P. Corke, “Modelling and control of a quad-rotor robot,” in *Proceedings of the Australasian Conference on Robotics and Automation 2006*, 2006.
- [8] B. R. Hein and I. Chopra, “Hover performance of a micro air vehicle: Rotors at low reynolds number,” *Journal of the American Helicopter Society*, vol. 52, no. 3, pp. 254–262, 2007.
- [9] W. Giernacki, M. Skwierczyński, W. Witwicki, P. Wroński, and P. Koziński, “Crazyflie 2.0 quadrotor as a platform for research and education in robotics and control engineering,” in *2017 22nd International Conference on Methods and Models in Automation and Robotics (MMAR)*, (Miedzyzdroje, Poland), pp. 37–42, 2017.
- [10] J. H. Gillula, H. Huang, M. P. Vitus, and C. J. Tomlin, “Design of guaranteed safe maneuvers using reachable sets: Autonomous quadrotor aerobatics in theory and

- practice,” in *Robotics and Automation (ICRA), IEEE International Conference on*, (Anchorage, AK, USA), pp. 1649–1654, 2010.
- [11] G. M. Hoffmann, H. Huang, S. L. Waslander, and C. J. Tomlin, “Quadrotor helicopter flight dynamics and control: Theory and experiment,” in *AIAA Guidance, Navigation and Control Conference*, pp. 1–20, 2007.
- [12] T. Lee, “Robust adaptive attitude tracking on $rmSO(3)$ with an application to a quadrotor uav,” *Control Systems Technology, IEEE Transactions on*, vol. 21, no. 5, pp. 1924–1930, 2013.
- [13] R. He, A. Bachrach, M. Achtelik, A. Geramifard, D. Gurdan, S. Prentice, J. Stumpf, and N. Roy, “On the design and use of a micro air vehicle to track and avoid adversaries,” *The International Journal of Robotics Research*, vol. 29, no. 5, pp. 529–546, 2010.
- [14] A. Kushleyev, B. MacAllister, and M. Likhachev, “Planning for landing site selection in the aerial supply delivery,” in *Intelligent Robots and Systems (IROS), IEEE/RSJ International Conference on*, (San Francisco, CA, USA), pp. 1146–1153, 2011.
- [15] R. He, S. Prentice, and N. Roy, “Planning in information space for a quadrotor helicopter in a gps-denied environment,” in *Robotics and Automation (ICRA), IEEE International Conference on*, (Pasadena, CA, USA), pp. 1814–1820, 2008.
- [16] D. Mellinger and V. Kumar, “Minimum snap trajectory generation and control for quadrotors,” in *Robotics and Automation (ICRA), IEEE International Conference on*, (Shanghai, China), pp. 2520–2525, 2011.
- [17] M. Hehn and R. D’Andrea, “Quadcopter trajectory generation and control,” *IFAC Proceedings Volumes*, vol. 44, no. 1, pp. 1485–1491, 2011. 18th IFAC World Congress.
- [18] S. Lupashin, A. Schöllig, M. Sherback, and R. D’Andrea, “A simple learning strategy for high-speed quadcopter multi-flips,” in *Robotics and Automation (ICRA), IEEE International Conference on*, (Anchorage, AK, USA), pp. 1642–1648, 2010.
- [19] D. Mellinger, N. Michael, and V. Kumar, *Trajectory Generation and Control for Precise Aggressive Maneuvers with Quadrotors*, pp. 361–373. Berlin, Heidelberg: Springer Berlin Heidelberg, 2014.

- [20] Q. Lindsey, D. Mellinger, and V. Kumar, *Construction of cubic structures with quadrotor teams*. Robotics: Science & Systems, 2011.
- [21] F. Augugliaro, S. Lupashin, M. Hamer, C. Male, M. Hehn, M. W. Mueller, J. S. Willmann, F. Gramazio, M. Kohler, and R. D’Andrea, “The flight assembled architecture installation: Cooperative construction with flying machines,” *IEEE Control Systems*, vol. 34, pp. 46–64, Aug 2014.
- [22] V. Ghadiok, J. Goldin, and W. Ren, “Autonomous indoor aerial gripping using a quadrotor,” in *Intelligent Robots and Systems (IROS), IEEE/RSJ International Conference on*, (San Francisco, CA, USA), pp. 4645–4651, 2011.
- [23] L. Heng, L. Meier, P. Tanskanen, F. Fraundorfer, and M. Pollefeys, “Autonomous obstacle avoidance and maneuvering on a vision-guided mav using on-board processing,” in *Robotics and Automation (ICRA), IEEE International Conference on*, (Shanghai, China), pp. 2472–2477, 2011.
- [24] S. Shen, Y. Mulgaonkar, N. Michael, and V. Kumar, “Vision-based state estimation for autonomous rotorcraft mavs in complex environments,” in *Robotics and Automation (ICRA), IEEE International Conference on*, (Karlsruhe, Germany), pp. 1758–1764, 2013.
- [25] A. S. Huang, A. Bachrach, P. Henry, M. Krainin, D. Maturana, D. Fox, and N. Roy, *Visual Odometry and Mapping for Autonomous Flight Using an RGB-D Camera*, pp. 235–252. Cham: Springer International Publishing, 2017.
- [26] Z. T. Dydek, A. M. Annaswamy, and E. Lavretsky, “Adaptive control and the nasa x-15-3 flight revisited,” *IEEE Control Systems Magazine*, vol. 30, no. 3, pp. 32–48, 2010.
- [27] J. G. Leishman, *Principles of helicopter aerodynamics*. Cambridge University Press, 2006.
- [28] “Convertawings model a.” http://www.aviastar.org/helicopters_eng/convertawings.php. Accessed: 2017-05-25.
- [29] “Curtiss-wright vz-7.” http://www.aviastar.org/helicopters_eng/curtiss_vz-7.php. Accessed: 2017-05-25.
- [30] “Curtiss x-19.” http://www.aviastar.org/helicopters_eng/curtiss_x-19.php. Accessed: 2017-05-25.

- [31] “Ar. drone.” <https://www.parrot.com/fr>. Accessed: 2017-05-28.
- [32] B. Pierre-Jean, C. François, V. David, and P. Nicolas, “The navigation and control technology inside the ar.drone micro uav,” *IFAC Proceedings Volumes*, vol. 44, no. 1, pp. 1477–1484, 2011.
- [33] “Arducopter.” <http://code.google.com/p/arducopter>. Accessed: 2017-05-28.
- [34] “Paparazzi.” <http://http://paparazzi.enac.fr>. Accessed: 2017-05-28.
- [35] “Pixhawk.” <http://http://Pixhawk.ethz.ch>. Accessed: 2017-05-28.
- [36] T. Hamel and P. Souères, “Modélisation, estimation et contrôle des drones à voilures tournantes; un aperçu des projets de recherche français,” *5ème Journées Nationales de la Recherche en Robotique*, 2005.
- [37] P. Pounds, *Design, construction and control of a large quadrotor micro air vehicle*. PhD thesis, Australian National University, Australia, 2007.
- [38] P. Pounds, R. Mahony, P. Hynes, and J. Roberts, “Design of a four-rotor aerial robot,” in *Proceedings of the Australasian Conference on Robotics and Automation*, pp. 145–150, 2002.
- [39] P. Pounds, R. Mahony, and J. Gresham, “Towards dynamically favourable quadrotor aerial robots,” in *Proceedings of the Australasian Conference on Robotics and Automation*, pp. 145–150, 2004.
- [40] P. Pounds, R. Mahony, and P. Corke, “Modelling and control of a large quadrotor robot,” *Control Engineering Practice*, vol. 18, no. 7, pp. 691–699, 2010.
- [41] S. Bouabdallah, P. Murrieri, and R. Siegwart, “Design and control of an indoor micro quadrotor,” in *Robotics and Automation (ICRA), IEEE International Conference on*, pp. 4393–4398, 2004.
- [42] S. Bouabdallah, A. Noth, and R. Siegwart, “Pid vs lq control techniques applied to an indoor micro quadrotor,” in *Intelligent Robots and Systems (IROS), IEEE/RSJ International Conference on*, vol. 3, pp. 2451–2456, 2004.
- [43] S. Bouabdallah and R. Siegwart, “Backstepping and sliding-mode techniques applied to an indoor micro quadrotor,” in *Robotics and Automation, IEEE International Conference on*, pp. 2247–2252, IEEE, 2005.

- [44] P. Castillo, A. Dzul, and R. Lozano, “Real-time stabilization and tracking of a four-rotor mini rotorcraft,” *Control Systems Technology, IEEE Transactions on*, vol. 12, no. 4, pp. 510–516, 2004.
- [45] A. R. Teel, “Global stabilization and restricted tracking for multiple integrators with bounded controls,” *Systems & control letters*, vol. 18, no. 3, pp. 165–171, 1992.
- [46] P. Castillo, P. Albertos, P. Garcia, and R. Lozano, “Simple real-time attitude stabilization of a quad-rotor aircraft with bounded signals,” in *Decision and Control, 2006 45th IEEE Conference on*, pp. 1533–1538, IEEE, 2006.
- [47] A. Tayebi and S. McGilvray, “Attitude stabilization of a vtol quadrotor aircraft,” *Control systems technology, IEEE Transactions on*, vol. 14, no. 3, pp. 562–571, 2006.
- [48] S. Berkane, A. Abdessameud, and A. Tayebi, “Hybrid global exponential stabilization on $so(3)$,” *Automatica*, vol. 81, no. Supplement C, pp. 279–285, 2017.
- [49] S. Berkane, A. Abdessameud, and A. Tayebi, “Hybrid attitude and gyro-bias observer design on $so(3)$,” *Automatic Control, IEEE Transactions on*, vol. 62, no. 11, pp. 6044–6050, 2017.
- [50] S. Berkane and A. Tayebi, “On the design of attitude complementary filters on $so(3)$,” *Automatic Control, IEEE Transactions on*, vol. PP, no. 99, pp. 1–1, 2017.
- [51] G. M. Hoffmann, *Autonomy for sensor-rich vehicles: Interaction between sensing and control actions*. PhD thesis, Stanford University, CA, USA, 2008.
- [52] G. M. Hoffmann and C. J. Tomlin, “Decentralized cooperative collision avoidance for acceleration constrained vehicles,” in *Decision and Control (CDC), 47th IEEE Conference on*, (Cancun, Mexico), pp. 4357–4363, 2008.
- [53] D. Mellinger, M. Shomin, and V. Kumar, “Control of quadrotors for robust perching and landing,” in *Proceedings of the International Powered Lift Conference*, pp. 205–225, 2010.
- [54] D. Mellinger, M. Shomin, N. Michael, and V. Kumar, “Cooperative grasping and transport using multiple quadrotors,” in *Distributed autonomous robotic systems*, pp. 545–558, Springer, 2013.
- [55] D. Mellinger, N. Michael, and V. Kumar, “Trajectory generation and control for precise aggressive maneuvers with quadrotors,” *The International Journal of Robotics Research*, vol. 31, no. 5, pp. 664–674, 2012.

- [56] S. Lupashin, A. Schoellig, M. Sherback, and R. D’Andrea, “A simple learning strategy for high-speed quadrocopter multi-flips,” in *Robotics and Automation (ICRA), IEEE International Conference on*, pp. 1642–1648, IEEE, 2010.
- [57] A. Schöllig, J. Alonso-Mora, and R. D’Andrea, “Independent vs. joint estimation in multi-agent iterative learning control,” in *Decision and Control (CDC), 49th IEEE Conference on*, pp. 6949–6954, IEEE, 2010.
- [58] M. Hehn and R. D’Andrea, “Real-time trajectory generation for interception maneuvers with quadrocopters,” in *Intelligent Robots and Systems (IROS), IEEE/RSJ International Conference on*, pp. 4979–4984, IEEE, 2012.
- [59] M. Hehn, R. Ritz, and R. D’Andrea, “Performance benchmarking of quadrotor systems using time-optimal control,” *Autonomous Robots*, vol. 33, pp. 69–88, 2012. 10.1007/s10514-012-9282-3.
- [60] R. Ritz, M. Mueller, and R. D’Andrea, “Cooperative quadrocopter ball throwing and catching,” in *Intelligent Robots and Systems (IROS), IEEE/RSJ International Conference on*, pp. 4972–4978, IEEE, 2012.
- [61] F. Augugliaro, A. Mirjan, F. Gramazio, M. Kohler, and R. D’Andrea, “Building tensile structures with flying machines,” in *Intelligent Robots and Systems (IROS), IEEE/RSJ International Conference on*, pp. 3487–3492, IEEE, 2013.
- [62] C. Coza and C. Macnab, “A new robust adaptive-fuzzy control method applied to quadrotor helicopter stabilization,” in *Fuzzy Information Processing Society, 2006. NAFIPS 2006. Annual meeting of the North American*, pp. 454–458, IEEE, 2006.
- [63] A. Rabhi, M. Chadli, and C. Pegard, “Robust fuzzy control for stabilization of a quadrotor,” in *Advanced Robotics (ICAR), 15th International Conference on*, pp. 471–475, IEEE, 2011.
- [64] Z. Lendek, A. Berna, J. Guzmán-Giménez, A. Sala, and P. García, “Application of takagi-sugeno observers for state estimation in a quadrotor,” in *Decision and Control and European Control Conference (CDC-ECC), 50th IEEE Conference on*, pp. 7530–7535, IEEE, 2011.
- [65] B. Castillo-Toledo, S. Di Gennaro, and F. Jurado, “Trajectory tracking for a quadrotor via fuzzy regulation,” in *World Automation Congress (WAC), 2012*, pp. 1–6, IEEE, 2012.

- [66] C.-L. Hwang and C. Jan, “Fuzzy decentralized sliding-mode under-actuated trajectory-tracking control for quadrotor unmanned aerial vehicle,” in *Fuzzy Systems (FUZZ-IEEE), IEEE International Conference on*, pp. 1–10, IEEE, 2012.
- [67] H. Voos, “Nonlinear and neural network-based control of a small four-rotor aerial robot,” in *Advanced intelligent mechatronics, IEEE/ASME international conference on*, pp. 1–6, IEEE, 2007.
- [68] C. Nicol, C. Macnab, and A. Ramirez-Serrano, “Robust neural network control of a quadrotor helicopter,” in *Electrical and Computer Engineering (CCECE), Canadian Conference on*, pp. 001233–001238, IEEE, 2008.
- [69] H. Boudjedir, O. Bouhali, and N. Rizoug, “Adaptive neural network control based on neural observer for quadrotor unmanned aerial vehicle,” *Advanced Robotics*, vol. 28, no. 17, pp. 1151–1164, 2014.
- [70] T. Madani and A. Benallegue, “Adaptive control via backstepping technique and neural networks of a quadrotor helicopter,” *IFAC Proceedings Volumes*, vol. 41, no. 2, pp. 6513–6518, 2008.
- [71] A. Mokhtari, A. Benallegue, and B. Daachi, “Robust feedback linearization and g_h^∞ controller for a quadrotor unmanned aerial vehicle,” in *Intelligent Robots and Systems (IROS), IEEE/RSJ International Conference on*, pp. 1198–1203, IEEE, 2005.
- [72] K. Alexis, G. Nikolakopoulos, and A. Tzes, “On trajectory tracking model predictive control of an unmanned quadrotor helicopter subject to aerodynamic disturbances,” *Asian Journal of Control*, vol. 16, no. 1, pp. 209–224, 2014.
- [73] S. L. Waslander, G. M. Hoffmann, J. S. Jang, and C. J. Tomlin, “Multi-agent quadrotor testbed control design: Integral sliding mode vs. reinforcement learning,” in *Intelligent Robots and Systems (IROS), IEEE/RSJ International Conference on*, pp. 3712–3717, IEEE, 2005.
- [74] K. Alexis, G. Nikolakopoulos, and A. Tzes, “Constrained optimal attitude control of a quadrotor helicopter subject to wind-gusts: Experimental studies,” in *American Control Conference (ACC)*, pp. 4451–4455, IEEE, 2010.
- [75] F. Hoffmann, N. Goddemeier, and T. Bertram, “Attitude estimation and control of a quadcopter,” in *Intelligent Robots and Systems (IROS), IEEE/RSJ International Conference on*, pp. 1072–1077, IEEE, 2010.

- [76] G. V. Raffo, M. G. Ortega, and F. R. Rubio, “Backstepping/nonlinear h_∞ control for path tracking of a quadrotor unmanned aerial vehicle,” in *American Control Conference (ACC)*, (Seattle, Washington), pp. 3356–3361, 2008.
- [77] M. Kerma, A. Mokhtari, B. Abdelaziz, and Y. Orlov, “Nonlinear h_∞ control of a quadrotor (uav), using high order sliding mode disturbance estimator,” *International Journal of Control*, vol. 85, no. 12, pp. 1876–1885, 2012.
- [78] M. Huang, B. Xian, C. Diao, K. Yang, and Y. Feng, “Adaptive tracking control of underactuated quadrotor unmanned aerial vehicles via backstepping,” in *American Control Conference (ACC)*, pp. 2076–2081, 2010.
- [79] H. Bouadi, S. S. Cunha, A. Drouin, and F. Mora-Camino, “Adaptive sliding mode control for quadrotor attitude stabilization and altitude tracking,” in *Computational Intelligence and Informatics (CINTI), IEEE 12th International Symposium on*, pp. 449–455, IEEE, 2011.
- [80] D. Lee, C. Nataraj, T. C. Burg, and D. M. Dawson, “Adaptive tracking control of an underactuated aerial vehicle,” in *American Control Conference (ACC)*, pp. 2326–2331, IEEE, 2011.
- [81] T. Fernando, J. Chandiramani, T. Lee, and H. Gutierrez, “Robust adaptive geometric tracking controls on $so(3)$ with an application to the attitude dynamics of a quadrotor uav,” in *Decision and Control and European Control Conference (CDC-ECC), 50th IEEE Conference on*, pp. 7380–7385, IEEE, 2011.
- [82] A. Zavala-Río, I. Fantoni, and R. Lozano, “Global stabilization of a pvtol aircraft model with bounded inputs,” *International Journal of Control*, vol. 76, no. 18, pp. 1833–1844, 2003.
- [83] R. Lozano, P. Castillo, and A. Dzul, “Global stabilization of the pvtol: real-time application to a mini-aircraft,” *International Journal of Control*, vol. 77, no. 8, pp. 735–740, 2004.
- [84] T. Madani and A. Benallegue, “Backstepping control for a quadrotor helicopter,” in *Intelligent Robots and Systems (IROS), IEEE/RSJ International Conference on*, pp. 3255–3260, IEEE, 2006.
- [85] M. Bouchoucha, M. Tadjine, A. Tayebi, and P. Mullhaupt, “Backstepping based nonlinear pi for attitude stabilisation of a quadrotor: From theory to experi-

- ment,” in *Intelligent Robots and Systems (IROS), IEEE/RSJ International Conference on*, pp. 4183–4183, IEEE, 2008.
- [86] R. Xu and U. Ozguner, “Sliding mode control of a quadrotor helicopter,” in *Decision and Control, 45th IEEE Conference on*, pp. 4957–4962, IEEE, 2006.
- [87] L. Derafa, L. Fridman, A. Benallegue, and A. Ouldali, “Super twisting control algorithm for the four rotors helicopter attitude tracking problem,” in *Variable Structure Systems (VSS), 11th International Workshop on*, pp. 62–67, IEEE, 2010.
- [88] L.-X. Wang, *Adaptive fuzzy systems and control: design and stability analysis*. PTR Prentice Hall, Englewood Cliffs, 1994.
- [89] L.-X. Wang, “Fuzzy systems are universal approximators,” in *Fuzzy Systems, IEEE International Conference on*, pp. 1163–1170, IEEE, 1992.
- [90] T. Takagi and M. Sugeno, “Fuzzy identification of systems and its applications to modeling and control,” *IEEE Transactions on Systems, Man, and Cybernetics*, vol. SMC-15, no. 1, pp. 116–132, 1985.
- [91] L. X. Wang and J. M. Mendel, “Fuzzy basis functions, universal approximation, and orthogonal least-squares learning,” *IEEE Transactions on Neural Networks*, vol. 3, no. 5, pp. 807–814, 1992.
- [92] L.-X. Wang, *A course in fuzzy system and control*. Prentice-Hall International, Inc, 1997.
- [93] K. M. Passino and S. Yurkovich, *Fuzzy Control*. Addison Wesley Longman, Inc., 1998.
- [94] P. Castillo, R. Lozano, and A. Dzul, *Modelling and control of mini-flying machines*. Springer Science & Business Media, 2006.
- [95] J. Escareño, S. Salazar-Cruz, and R. Lozano, “Embedded control of a four-rotor uav,” in *American Control Conference (ACC), (Minneapolis, MN)*, pp. 3936–3941, 2006.
- [96] F. Kendoul, D. Lara, I. Fantoni, and R. Lozano, “Nonlinear control for systems with bounded inputs: Real-time embedded control applied to uavs,” in *Decision and Control (CDC), 45th IEEE Conference on, (San Diego, CA, USA)*, pp. 5888–5893, 2006.

- [97] G. V. Raffo, M. G. Ortega, and F. R. Rubio, “An integral predictive/nonlinear ∞ control structure for a quadrotor helicopter,” *Automatica*, vol. 46, no. 1, pp. 29–39, 2010.
- [98] T. Madani and A. Benallegue, “Control of a quadrotor mini-helicopter via full state backstepping technique,” in *Decision and Control (CDC), 45th IEEE Conference on*, (Minneapolis, MN), pp. 1515–1520, 2006.
- [99] T. Madani and A. Benallegue, “Backstepping control with exact 2-sliding mode estimation for a quadrotor unmanned aerial vehicle,” in *Intelligent Robots and Systems (IROS), IEEE/RSJ International Conference on*, pp. 141–146, 2007.
- [100] S. Bouabdallah and R. Siegwart, “Full control of a quadrotor,” in *Intelligent Robots and Systems (IROS), IEEE/RSJ International Conference on*, pp. 153–158, 2007.
- [101] T. Madani and A. Benallegue, “Sliding mode observer and backstepping control for a quadrotor unmanned aerial vehicles,” in *American Control Conference (ACC)*, pp. 5887–5892, 2007.
- [102] J. R. Taylor, *Classical Mechanics*. U.S: University Science Books, 2005.
- [103] H. Goldstein, C. Poole, and J. Safko, *Classical Mechanics*. Addison Wesley, 2002.
- [104] G. Fay, *Derivation of the aerodynamic forces for the mesicopter simulation*. Stanford University, USA, 2001.
- [105] H. A. Izadi, Y. Zhang, and B. W. Gordon, “Fault tolerant model predictive control of quad-rotor helicopters with actuator fault estimation,” *IFAC Proceedings Volumes*, vol. 44, no. 1, pp. 6343–6348, 2011.
- [106] H. Voos, “Nonlinear control of a quadrotor micro-uav using feedback-linearization,” in *Mechatronics, ICM 2009. IEEE International Conference on*, pp. 1–6, IEEE, 2009.
- [107] L. Mollov, J. Kralev, T. Slavov, and P. Petkov, “ μ -synthesis and hardware-in-the-loop simulation of miniature helicopter control system,” *Journal of Intelligent & Robotic Systems*, vol. 76, no. 2, pp. 315–351, 2014.
- [108] J. Etele, *Overview of Wind Gust Modelling with Application to Autonomous Low-Level UAV Control*. Defence Research and Development Canada, 2006.

- [109] K. Alexis, G. Nikolakopoulos, and A. Tzes, “Constrained-control of a quadrotor helicopter for trajectory tracking under wind-gust disturbances,” in *Mediterranean Electrotechnical MELECON, 15th IEEE Conference on*, pp. 1411–1416, 2010.
- [110] B. Ahmed and F. Kendoul, “Flight control of a small helicopter in unknown wind conditions,” in *Decision and Control (CDC), 49th IEEE Conference on*, pp. 3536–3541, 2010.
- [111] K. A. Swieringa, C. B. Hanson, J. R. Richardson, J. D. White, Z. Hasan, E. Qian, and A. Girard, “Autonomous battery swapping system for small-scale helicopters,” in *Robotics and Automation (ICRA), IEEE International Conference on*, pp. 3335–3340, 2010.
- [112] J. Leonard, A. Savvaris, and A. Tsourdos, “Energy management in swarm of unmanned aerial vehicles,” *Journal of Intelligent & Robotic Systems*, vol. 74, no. 1-2, pp. 233–250, 2014.
- [113] N. K. Ure, G. Chowdhary, T. Toksoz, J. P. How, M. A. Vavrina, and J. Vian, “An automated battery management system to enable persistent missions with multiple aerial vehicles,” *Mechatronics, IEEE/ASME Transactions on*, vol. 20, no. 1, pp. 275–286, 2015.
- [114] J. F. Roberts, J.-C. Zufferey, and D. Floreano, “Energy management for indoor hovering robots,” in *Intelligent Robots and Systems (IROS), IEEE/RSJ International Conference on*, pp. 1242–1247, 2008.
- [115] A. Abdilla, A. Richards, and S. Burrow, “Power and endurance modelling of battery-powered rotorcraft,” in *Intelligent Robots and Systems (IROS), IEEE/RSJ International Conference on*, pp. 675–680, 2015.
- [116] M. Podhradský, C. Coopmans, and A. Jensen, “Battery state-of-charge based altitude controller for small, low cost multirotor unmanned aerial vehicles,” *Journal of Intelligent & Robotic Systems*, vol. 74, no. 1-2, pp. 193–207, 2014.
- [117] D. C. Gandolfo, L. R. Salinas, A. Brandão, and J. M. Toibero, “Stable path-following control for a quadrotor helicopter considering energy consumption,” *Control Systems Technology, IEEE Transactions on*, vol. 25, no. 4, pp. 1423–1430, 2016.
- [118] K. Vicencio, T. Korras, K. A. Bordignon, and I. Gentilini, “Energy-optimal path planning for six-rotors on multi-target missions,” in *Intelligent Robots and Systems (IROS), IEEE/RSJ International Conference on*, pp. 2481–2487, 2015.

- [119] F. Morbidi, R. Cano, and D. Lara, “Minimum-energy path generation for a quadrotor uav,” in *Robotics and Automation (ICRA), IEEE International Conference on*, 2016.
- [120] J. T. Betts, “Survey of numerical methods for trajectory optimization,” *Journal of Guidance Control and Dynamics*, vol. 21, no. 2, pp. 193–207, 1998.
- [121] S. Hosseini, R. Dai, and M. Mesbahi, “Optimal path planning and power allocation for a long endurance solar-powered uav,” in *American Control Conference (ACC)*, pp. 2588–2593, 2013.
- [122] M. Kamgarpour, M. Soler, C. J. Tomlin, A. Olivares, and J. Lygeros, “Hybrid optimal control for aircraft trajectory design with a variable sequence of modes,” *IFAC Proceedings Volumes*, vol. 44, no. 1, pp. 7238–7243, 2011. 18th IFAC World Congress.
- [123] F. Yacef, N. Rizoug, O. Bouhali, and M. Hamerlain, “Optimization of energy consumption for quadrotor uav,” in *International Micro Air Vehicle Conference and Flight Competition (IMAV)*, (Toulouse, France), pp. 215–222, 2017.
- [124] M. Cutler, N. K. Ure, B. Michini, and J. P. How, “Comparison of fixed and variable pitch actuators for agile quadrotors,” in *AIAA Conference on Guidance, Navigation and Control*, 2011.
- [125] T. Mesbahi, N. Rizoug, P. Bartholomeus, and P. Le Moigne, “Li-ion battery emulator for electric vehicle applications,” in *Vehicle Power and Propulsion (VPPC), IEEE Conference on*, pp. 1–8, IEEE, 2013.
- [126] K. Divya and J. Østergaard, “Battery energy storage technology for power systems—an overview,” *Electric Power Systems Research*, vol. 79, no. 4, pp. 511–520, 2009.
- [127] F. Yacef, N. Rizoug, L. Degaa, O. Bouhali, and M. Hamerlain, “Trajectory optimisation for a quadrotor helicopter considering energy consumption,” in *Control Decision and Information Technologies (CoDIT), IEEE Conference on*, (Barcelona, Spain), pp. 1030–1035, 2017.
- [128] M. A. Patterson and A. V. Rao, “Gpops-ii: A matlab software for solving multiple-phase optimal control problems using hp-adaptive gaussian quadrature collocation methods and sparse nonlinear programming,” *ACM Trans. Math. Softw.*, vol. 41, no. 1, pp. 1–37, 2014.

- [129] D. Garg, W. W. Hager, and A. V. Rao, “Pseudospectral methods for solving infinite-horizon optimal control problems,” *Automatica*, vol. 47, no. 4, pp. 829–837, 2011.
- [130] M. A. Patterson and A. Rao, “Exploiting sparsity in direct collocation pseudospectral methods for solving optimal control problems,” *Journal of Spacecraft and Rockets*, vol. 49, no. 2, pp. 354–377, 2012.
- [131] L. T. Biegler, O. Ghattas, M. Heinkenschloss, and B. van Bloemen Waanders, “Large-scale pde-constrained optimization: an introduction,” in *Large-Scale PDE-Constrained Optimization*, pp. 3–13, Springer, 2003.
- [132] P. E. Gill, W. Murray, and M. A. Saunders, “Snopt: An sqp algorithm for large-scale constrained optimization,” *SIAM review*, vol. 47, no. 1, pp. 99–131, 2005.
- [133] P. R. Amestoy, I. S. Duff, J. Koster, and J.-Y. L’Excellent, “A fully asynchronous multifrontal solver using distributed dynamic scheduling,” *SIAM Journal on Matrix Analysis and Applications*, vol. 23, no. 1, pp. 15–41, 2001.
- [134] F. Yacef, O. Bouhali, M. Hamerlain, and N. Rizoug, “Observer-based adaptive fuzzy backstepping tracking control of quadrotor unmanned aerial vehicle powered by li-ion battery,” *Journal of Intelligent & Robotic Systems*, vol. 84, no. 1, pp. 179–197, 2016.
- [135] J.-J. Slotine, W. Li, *et al.*, *Applied nonlinear control*, vol. 199. Prentice Hall Englewood Cliffs, NJ, 1991.
- [136] S.-C. Tong, Y.-M. Li, G. Feng, and T.-S. Li, “Observer-based adaptive fuzzy backstepping dynamic surface control for a class of mimo nonlinear systems,” *Systems, Man, and Cybernetics, Part B: Cybernetics, IEEE Transactions on*, vol. 41, no. 4, pp. 1124–1135, 2011.
- [137] L.-X. Wang, *Adaptive Fuzzy Systems and Control: Design and Stability Analysis*. Upper Saddle River, NJ, USA: Prentice-Hall, Inc., 1994.
- [138] P. Ioannou and J. Sun, *Robust Adaptive Control*. Prentice Hall, Inc, 1996.
- [139] F. Yacef, O. Bouhali, and M. Hamerlain, “Adaptive fuzzy backstepping control for trajectory tracking of unmanned aerial quadrotor,” in *Unmanned Aircraft Systems (ICUAS), IEEE International Conference on*, pp. 920–927, 2014.
- [140] L. Shihua, Y. Jun, C. Wen-Hua, and C. Xisong, *Disturbance Observer-Based Control*. CRC Press, Taylor & Francis Group, 2014.

-
- [141] D. Cabecinhas, R. Cunha, and C. Silvestre, “A nonlinear quadrotor trajectory tracking controller with disturbance rejection,” *Control Engineering Practice*, vol. 26, pp. 1–10, 2014.
- [142] L. Wang and J. Su, “Robust disturbance rejection control for attitude tracking of an aircraft,” *Control Systems Technology, IEEE Transactions on*, vol. 23, no. 6, pp. 2361–2368, 2015.
- [143] C. Wang, B. Song, P. Huang, and C. Tang, “Trajectory tracking control for quadrotor robot subject to payload variation and wind gust disturbance,” *Journal of Intelligent & Robotic Systems*, vol. 83, no. 2, pp. 315–333, 2016.