

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

UNIVERSITE MOHAMED BOUDIAF - M'SILA

FACULTE DE TECHNOLOGIE
DEPARTEMENT GENIE ELECTRIQUE



FILIERE : AUTOMATIQUE
OPTION : AUTOMATIQUE ET SYSTEMES

N° :

**Mémoire présenté pour l'obtention
Du diplôme de Master Académique**

Par: Dahoum Mehdi Abdessamad

Korichi Nabil Moustafa

Thème

**Approches De L'intelligence Artificielle Pour La
Commande Robuste Des Drones De Type Quadrotor**

Encadré par:

Dr. BENYOUNES Abdelhafid

Année universitaire : 2020 /2021

بِسْمِ اللَّهِ الرَّحْمَنِ الرَّحِيمِ

Dédicaces

Je dédie ce travail :

A mon cher père et ma chère mère

Mes sœurs

Mon frère

A toute ma famille

A tous mes amis spécialement I.Gendouz

M .Hadjeb, A.chatra et T.Bouzidi

Dahoum Mehdi Abdessamad

Dédicaces

Avant tous, je remercie dieu le tout puissant de m'avoir donné le courage et la patience pour réaliser ce travail malgré toutes les difficultés rencontrées.

Je dédie ce travail en signe de respect et de reconnaissance à :

Ma très chère mère et mon père

Mes sœurs

Mes frères

A toute ma famille

A tous mes amis spécialement R.Kahoul,

A.Safer Tabi et A .Touafek

Mon binôme M.Dahoum

Remerciements

A l'issue de cette fin de travail nous adressons nos remerciements premièrement à dieu tout puissant pour la volonté, la santé et la patience qu'il nous a donnée durant toutes ces longues années d'études.

Nos remerciements s'adressent à notre encadreur

Dr BENOÛNES ABDELHAFID pour avoir proposé et diriger ce modeste travail et pour son continuel suivi tout le long de la correction de ce mémoire. Il a présenté un excellent exemple de la compétence scientifique, discipline et générosité.

Nous remercions vivement Monsieur A .CHOUCHOU, Le chef de département de génie électrique pour sa disponibilité et sa générosité.

Nous remercions également tous les membres de jury d'avoir accepté de juger notre travail.

Nous tenons à remercier vivement toute personne qui nous a aidés de près ou de loin à accomplir ce travail.

Sommaire

INTRODUCTION GENERAL.....	1
Chapitre I : Description et Modélisation Dynamique du Quadrotor.....	3
I.1 Introduction.....	4
I.2 Définition d'UAV.....	4
I.3 Classification des drones.....	5
I.3.1 Selon la taille.....	5
I.3.2 Classification selon le mode de propulsion.....	6
I.3.3 Drones à voilures tournantes.....	7
I.4. Application des drones.....	8
I.5. Instrumentation des UAVs.....	9
I.6.Modélisation Dynamique du Quadrotor.....	11
I.6.1. Description générale du quadrirotor.....	11
I.6.2 Les possibilités de vol du quadrotor.....	12
I.6.3. Modélisation du Quadrotor.....	15
I.6.4. Matrice de rotation.....	16
I.6.5. Mouvement de translation et de rotation.....	17
I.6.5.1 Effets physiques agissants sur le quadrirotor.....	17
I.6.5.2 Effet gyroscopique.....	19
I.6.5.3 Equations de mouvement de translation.....	21
I.6.5.4 Equations de mouvement de rotation.....	21
I.6.6. Les contraintes non holonomes.....	22
I.6.7. Dynamique du rotor.....	22
I.6.8. Représentation d'état.....	23

I.7 Conclusion.....	24
Chapitre II : La Logique Floue Pour Le Contrôle du Quadrotor.....	26
II.1 Introduction.....	27
II.2. Concept de la logique floue.....	27
II.2.1 Sous-ensembles flous.....	28
II.2.2. Variables linguistiques.....	29
II.2.3 Fonctions d'appartenances.....	30
II.2.4 Operateurs flous.....	30
II.2.5 Règles d'inférences floues.....	30
II.3 Structure d'un contrôleur flou.....	31
II.3.1 La fuzzification.....	32
II.3.2 L'inférence et règles floue.....	33
II.3.3. Défuzzification.....	33
II.4. Commande par logique flou du quadrotor.....	34
II.4.1. Synthèse de contrôleur floue.....	36
II.4.2 Résultats de simulation.....	38
II.5. Commande PID adaptatif floue pour la stabilisation d'attitude du quadrotor.....	42
II.5.1. Principe de Régulateur PID.....	42
II.5.2. Synthèse de contrôleur PID adaptatif floue.....	43
II.5.3. Résultats de simulation.....	46
II.6. Commande par mode glissant-floue.....	47
II.6.1 la commande par mode de glissant.....	47
II.6.2 Phénomène de broutement (Chattering).....	48
II.6.3 Structure du contrôleur flou glissant.....	49
II.6.4 Synthèse de contrôleur floue par mode glissant.....	49
II.6.5 Résultats de simulation.....	53
II.7 Conclusions.....	57
Chapitre III : Commande Par Les Réseaux De Neurone Artificiel Pour La Stabilisation	
D'attitude Du Quadrotor.....	58
III.1 Introduction.....	59

III.2 Généralités.....	59
III.3 Neurone biologique.....	60
III.4. Eléments de base.....	62
III.4.1. Structure de base.....	62
III.4.2 Fonction d'activation.....	62
III.5. L'apprentissage des réseaux de neurones.....	63
III.6. Algorithme d'apprentissage.....	64
III.7. Identification et commande des systèmes par les réseaux de neurones.....	65
III.7.1. Identification des processus par réseaux de neurones.....	65
III.7.1.1. Identification directe.....	65
III.7.1.2 Identification inverse.....	66
III.7.2. Commande des processus par réseaux de neurones.....	67
III.7.2.1. Apprentissage d'un contrôleur conventionnel.....	67
III.7.2.2. Commande inverse avec apprentissage en ligne.....	68
III.8. Avantages et Inconvénients des réseaux de neurones.....	68
III.8.1. Avantages des réseaux de neurones.....	68
III.8.2. Inconvénients des réseaux de neurones.....	69
III.9. Commande par RNA directe pour la stabilisation d'attitude du quadrotor.....	69
III.9.1 Résultats de simulation.....	72
III.10 Commande PID-RNA pour la stabilisation d'attitude du quadrotor.....	73
III.10.1 Résultats de simulation.....	76
III.11 Conclusion.....	78
CONCLUSION GENERAL.....	79
Références bibliographiques.....	81

Liste des figures

Figure I.1 : Catégories des drones.....	6
Figure I.2 : Classification selon le mode de propulsion.....	6
Figure I.3. (a) Mono rotor; (b) Birotor.....	7
Figure I.4. (a) Tri rotor; (b) Quadrotor.....	8
Figure I.5 : Structure générale d'un quadrirotor.....	12
Figure I.6 : Illustration du mouvement vertical.....	13
Figure I.7 : Illustration du mouvement de roulis.....	14
Figure I.8 : Illustration du mouvement de tangage.....	14
Figure I.9 : Illustration du mouvement de lacet.....	15
Figure I.10 : Illustration du mouvement de translation.....	15
Figure I.11 : Géométrie du quadrirotor.....	16
Figure I.12 : les axes de rotation.....	17
Figure I.13 : schéma de rotor (mcc).....	23
Figure II.1 : Représentation de deux variables (grand et petit) par logique booléenne et par logique floue	28
Figure II.2 : Représentation floue de la variable vitesse.....	29
Figure II.3 : Exemple de fonctions d'appartenance.....	30
Figure II.4 Configuration générale d'un contrôleur flou.....	32
Figure II.5 : Schéma générale de commande du quadrotor.....	35
Figure II.6 : Schéma de la commande floue du quadrotor.....	36
Figure II.7 : Les fonctions d'appartenance de l'erreur et le dérivé de l'erreur.....	37
Figure II.8 : Les fonctions d'appartenance de la commande.....	37
Figure II.9 : Block de la commande floue.....	37
Figure II.10 : stabilisation d'attitude du quadrotor par le contrôleur floue.....	38
Figure II.11 : les signaux des commandes floue.....	38
Figure II.12 : Mouvement en X, Y, et Z par la commande floue.....	39
Figure II.13 : Trajectoire globale du quadrotor en 3D.....	39

Figure II.14 : variation des angles roulis, tangant, lacet par la commande floue.....	40
Figure II.15 : les commandes du contrôleur floue.....	41
Figure II.16 : Architecture système du PID flou pour le quadrirotor.....	43
Figure II.17 : Schéma de la commande PID floue du sous-système d'orientation.....	44
Figure II.18 : L'adaptation des gains par le régulateur floue.....	44
Figure II.19 : Les fonctions d'appartenance de l'erreur et le dérivé de l'erreur.....	45
Figure II.20 : Les fonctions d'appartenance de l'adaptation des gains.....	45
Figure II.21 : Block de la commande PID floue.....	45
Figure II.22 : stabilisation d'attitude par le contrôleur PID-floue.....	46
Figure II.23 : la commande du contrôleur PID-floue.....	46
Figure II.24 : l'adaptation des gains de contrôleur PID-floue.....	47
Figure II.25 Convergence du système glissant.....	48
Figure II.26 Phénomène de broutement (Chattering).....	49
Figure II.27 : Structure du régulateur flou-glissant.....	49
Figure II.28 : Schéma de la commande par mode glissant floue.....	50
Figure II.29 : Partition floue de l'espace autour de la surface de glissement.....	51
Figure II.30 : Les fonctions d'appartenances de S et de ΔU	52
Figure II.31 : Block de la commande par mode glissant floue.....	52
Figure II.32 : stabilisation d'attitude par le contrôleur mode glissant-floue.....	53
Figure II.33 : les signaux de commandes du contrôleur mode glissant-floue.....	53
Figure II.34 : réponse du système par le contrôleur mode glissant-floue.....	54
Figure II.35 : trajectoire de poursuite en 3D.....	54
Figure II.36 : variation des angles roulis, tangant, lacet par le contrôleur mode glissant-floue.....	55
Figure II.37 : les signaux de commande du contrôleur mode glissant-floue.....	56
Figure II.38 : phénomène de chattering.....	57
Figure III.1 Structure d'un réseau de neurones biologiques.....	60
Figure III.2 Schéma simplifié d'un neurone biologique.....	61
Figure III.3 le neurone formel.....	62
Figure III.4 Différentes fonctions d'activation.....	63
Figure III.5 Un réseau multicouche comportant 2 neurones d'entrée, 4 neurones cachés et un neurone de sortie.....	64
Figure III.6 Schéma d'identification directe d'un processus par réseau de neurones.....	66
Figure III.7 Schéma d'identification inverse d'un processus avec un réseau de neurones.....	67
Figure III.8 Schéma d'identification directe d'un contrôleur conventionnel avec un RNI.....	67

Figure III.9 Schéma de commande inverse avec un RNC.....	68
Figure III.10 : stratège de commande RNA du quadrotor proposée.....	69
Figure III.11 La structure de RNA utilisée.....	70
Figure III.12 la Stratégie de commande RNA du quadrotor sous Matlab Simulink.....	70
Figure III.13 Performance du modèle RNA.....	70
Figure III.14 Régression du modèle RNA.....	71
Figure III.15 stabilisation d'attitude par commande RNA directe.....	72
Figure III.16 les commande de RNA directe.....	72
Figure III.17 : stratège de commande PID-RNA du quadrotor proposée.	73
Figure III.18 blocks de commande PID-RNA.....	74
Figure III.19 : structure du modèle RNA utilisé.....	74
Figure III.20 : Performance du modèle PID-RNA.....	74
Figure III.21 : Régression du modèle PID-RNA.....	75
Figure III.22 stabilisation d'attitude par contrôleur PID par un modèle RNA.....	76
Figure III.23 : les commandes de contrôleur PID par un modèle RNA.....	76
Figure III.24 : l'adaptation des gains de contrôleur PID par un modèle RNA.....	77

Liste Des Tableau

Tableau II.1 : Récapitulatif des opérations floues.....	31
Tableau II.2 : Paramètres mécaniques et électriques du modèle du quadrotor.....	3

Introduction Générale.

Les engins volants autonomes ou semi autonomes sont des véhicules capables d'effectuer des missions en vol sans présence humaine à bord, ils sont aussi connus sous la dénomination « Unmanned Aerial Vehicles : UAV ». Le pilotage automatique d'avions et d'hélicoptères est né avec l'aviation moderne et a évolué au cours du temps pour satisfaire des besoins de plus en plus contraignants. Il peut être utilisé lorsque la tâche à réaliser est trop répétitive ou trop difficile pour le pilote.

Avec le développement et la commodité des drones, ces appareils sont de plus en plus utilisés dans un grand nombre de domaines. Nous ne prendrons comme exemples que la surveillance du trafic routier, l'exploration environnementale, l'arpentage et la cartographie et la photographie aérienne pour ne pas tous les énumérer. Utilitaires. Les capacités de décollage et d'atterrissage verticaux et la grande agilité des drones quadrirotors, ainsi que la capacité de vol stationnaire, les rendent très polyvalents, et leur aspect pratique est limité à l'imagination [1].

Pour la commande de telles classes de systèmes, les méthodes conventionnelles de l'automatique ont montré leurs limites en termes de stabilisation et de performances. Avec le développement considérable des calculateurs numériques, les automaticiens ont adopté de plus en plus de nouvelles approches telles que la commande prédictive et la commande robuste [2]. Néanmoins, cette solution présente l'inconvénient de nécessiter une estimation du modèle du système en temps réel. Les avancées récentes en informatique ont permis l'apparition d'outils intelligents comme les réseaux de neurones et la logique floue. Ces outils sont regroupés sous le concept de l'intelligence computationnelle. Cette dernière est un domaine de l'informatique qui étudie les mécanismes d'adaptation qui permettent un comportement intelligent dans des environnements changeants [3]. Elle se concentre sur les modèles informatiques des systèmes biologiques et d'intelligence naturelle. Elle recherche des méthodes de résolution de problèmes à forte complexité logique ou algorithmique. Par extension elle désigne, dans le langage courant, les dispositifs imitant ou remplaçant l'homme dans certaines mises en œuvre de ses fonctions cognitives [4].

La modélisation d'un quadrirotor n'est pas une tâche facile. Ceci est dû à la complexité et un grand nombre d'influences agissant sur le système, telles que le gyroscope, les influences aérodynamiques, la gravité, le frottement et les forces d'inertie. Toutes ces influences rendent notre système fortement non linéaire. Couplé à la forte interaction entre les états, on obtient un système difficile à modéliser [5].

Le contrôle intelligent est l'un des contrôleurs modernes qui n'a pas besoin d'un modèle mathématique comme les contrôleurs classiques, dans notre travail on veut appliquer ce contrôle sur un système sou actionné et non linéaire et en lisant les conclusions des chercheurs dans leurs recherches, on observe comment l'effet du contrôle intelligent sur ce système et si est efficace sur le contrôle classique.

Beaucoup de travail a été fait dans la modélisation et le contrôle des quadrirotors, et de nombreuses méthodes de l'intelligence artificielle ont été utilisées pour le contrôle, comme le régulateur flou, le PID flou, le mode glissant flou et le réseau neuronal [6] [7].

L'objectif commun de ces travaux est l'éducation et l'application des commandes intelligentes sur un système sous-actionné et l'utiliser comme adaptateur des contrôleurs classiques, atteindre un contrôle à haute efficacité pour stabiliser et d'asservir le quadrirotor et de lui faire suivre une trajectoire donnée.

Compte tenu de la difficulté de trouver un modèle mathématique fiable, les commandes floues constituent une méthode alternative de pilotage de ces dispositifs ce type de réglage est non linéaire et basé sur les connaissances de l'expert exécutant la commande. Le principe de ce règlement repose sur le choix des fonctions d'appartenance et des règles de comportement du système.

Ce mémoire s'articule autour de trois chapitres :

Le premier chapitre présentera l'état de l'Art et la modélisation mathématique du quadrirotor, qui consistera en une description de ce système et de son principe de vol puis nous développeront les différentes équations qui régissent ce système pour enfin proposer un modèle d'état en vue de réaliser une simulation de ce système sous Matlab-Simulink.

Le second chapitre introduira la logique floue et son principe en proposant une brève présentation de la régulation floue, puis proposera une commande à l'aide de régulateurs flous qui auront pour but de stabiliser la dynamique du quadrirotor par le développement des performances et des robustesses des contrôleurs conventionnels. Aussi, nous avons proposé deux techniques, la première est l'adaptation des gains des contrôleurs PID par un système expert flou, et la deuxième basée sur le choix d'une surface flou dans la structure de commande par mode glissant. Cette dernière prouvé son efficacité à travers les études théoriques rapportées, l'avantage que procure une telle commande et qui la rend aussi importante est sa robustesse vis-à-vis des perturbations et des incertitudes du modèle. Cependant, ces performances sont obtenues au prix de certains inconvénients : un phénomène de chattering qui peut causer l'instabilité et qui a un effet néfaste sur les actionneurs. On a combiné la logique floue avec le mode glissant afin d'obtenir une commande robuste et lisse. Enfin nous verrons les différents résultats obtenus lors de l'adaptation et lors des simulations effectuées grâce au modèle présenté dans le premier chapitre.

Le troisième chapitre portera des définitions et des généralités sur le réseau de neurones. Nous avons présenté deux techniques de commande, la première est une commande RNA directe, la deuxième basée sur l'adaptation des gains de la commande PID par un modèle RNA. On a appliqué ces deux techniques sur le système du quadrotor pour garantir la stabilisation d'attitude du quadrotor, et nous verrons les différents résultats obtenus lors des simulations effectuées.

Ce mémoire s'achèvera sur une conclusion générale.

CHAPITRE I

CHAPITRE I

DESCRIPTION ET MODÉLISATION DYNAMIQUE DU QUADROTOR

I.1 Introduction.

Les drones, qu'on appelle aussi UAVs (Unmanned Aerial Vehicles), sont des avions sans pilote. Ils sont adoptés par des organisations militaires et civiles pour accomplir des tâches difficiles dans des environnements très hostiles, sans aucun risque pour l'humain.

Les UAVs sont capables de transporter des caméras, des capteurs, des équipements de communication ou d'autres dispositifs. Ils sont utilisés pour réaliser des missions de reconnaissance, de recherche d'informations ou des opérations de combat [8] [9].

La commande des robots aériens nécessite un modèle dynamique afin de pouvoir tenir en compte les effets gyroscopiques ainsi que les forces aérodynamiques. Dans notre cas on s'intéresse de près au monde des drones et plus précisément à celui du quadrotor qui est en autre terme un avion à quatre rotors.

Dans ce chapitre, premièrement, nous présentons les UAVs, leurs classifications ainsi que leurs champs d'application, en suit nous intéressons principalement à la modélisation dynamique du quadrotor tenant en compte les différents paramètres qui peuvent affecter la dynamique d'une structure volante tel que les frottements dus aux moments aérodynamiques, les forces de traînée suivant les axes (X , Y , Z) et les effets gyroscopiques identifiés dans [10], pour un prototype expérimental existant au niveau du LISV de France (Laboratoire d'Ingénierie des Systèmes de Versailles), et aussi des contraintes non holonomes [11]. Et par conséquent cela nous a permis de mettre le système sous une nouvelle représentation d'état plus complète et plus réaliste.

I.2 Définition d'UAV.

Les drones sont des véhicules aériens sans pilote capables d'effectuer des tâches autonomes ou semi-autonomes [8]. Le drone est un appareil volant sans pilote, semi-automatique ou entièrement autonome qui peut transporter des charges utiles et être utilisé pour des missions spécifiques.

L'utilisation du mot « drone » pour décrivez que les véhicules ou robots autonomes terrestres, de surface ou sous-marins sont des cas particuliers en France. Classification des voitures autonomes sans conducteur aux États-Unis en UAV/UAS (aériens), USV/USS (de surface), UUV/UUS (sous-marins), UGV/UGS (robots terrestres) C'est le plus utilisé au monde.

Les drones à longue portée décrivent des drones et des hélicoptères volants. Si ça marche dans l'air, on l'appelle un UAV. Les drones amateurs les plus populaires sont les drones quadrotor. La caractéristique de base des drones est qu'ils sont récupérables, ce qui les rend réutilisables. Cela les sépare des missiles, qui peuvent essayer de les absorber.

I.3 Classification des drones.

Les drones peuvent être répartis selon plusieurs critères : taille, méthode de propulsion, système de contrôle, etc.

I.3.1 Selon la taille

On peut distinguer les catégories suivantes :

- HALE (Haute Altitude Longue Endurance) : Ce sont des aéronefs sans pilote, généralement à voilure fixe, capables de rester longtemps en vol et de collecter des informations pendant une longue période (12 à 48 heures).
- MALE (Moyenne Altitude Longue Endurance) : Ils ont une grande autonomie et peuvent être utilisés pour des vols longs distance à des altitudes de combat moyennes. Ces drones font partie d'un type d'avion de grande taille. Ils peuvent porter des armes, ce qui nécessite généralement une personne, cette personne doit conserver la décision de tirer et être en mesure d'annuler la mission à tout moment.
- Micro drones : Ce sont des drones dont la taille varie d'un centimètre à des dizaines de centimètres et se caractérisent par une faible charge. Habituellement, ils sont électriques et permettent le vol en intérieur.
- Mini drones : Ce sont des drones légers, de petite taille (pesant quelques kilogrammes, avec une envergure allant jusqu'à 1 à 2 mètres) et permettant d'atteindre une autonomie relativement faible (10 à 30 minutes). Généralement, ils sont utilisés pour observer les zones difficiles d'accès. La figure I.1 présente les différentes catégories des drones.

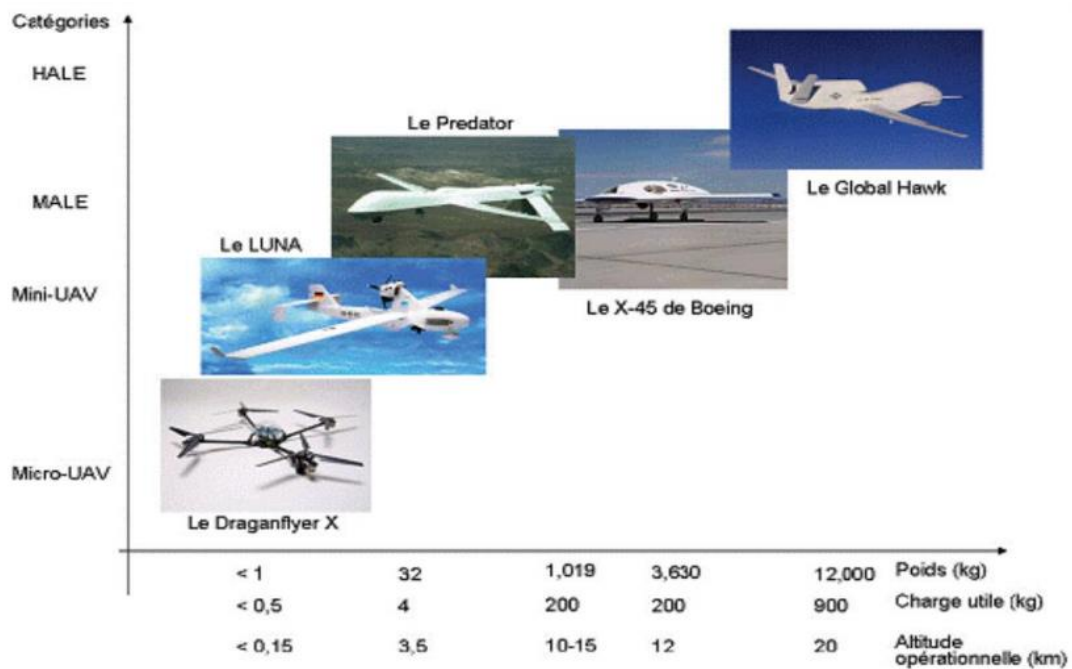


Figure I.1 : Catégories des drones.

I.3.2 Classification selon le mode de propulsion

Le fonctionnement offre une autre possibilité de classification. Par conséquent, les drones peuvent être divisés en trois familles principales :

- Drones à voilures fixes : utilisent des ailes fixes pendant le voyage.
- Drones à ailes battantes : utilisent des ailes comme un oiseau ou des insectes.
- Drones à voilures tournantes : Utilisent le même organe (rotor) pour la propulsion et le levage. Le drone peut atterrir et décoller verticalement, et planer à basse. La figure I.2 montre quelques exemples des drones à voilures fixes, voilures tournantes et ailes battantes.

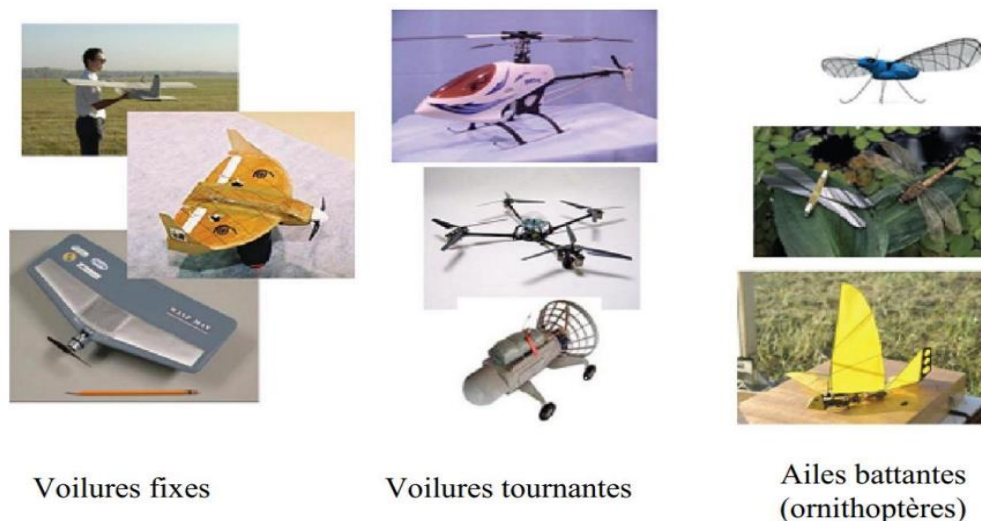


Figure I.2 : Classification selon le mode de propulsion.

I.3.3 Drones à voilures tournantes.

Les drones à voilure tournante se subdivisent, quant à eux, en quatre sous-classes qui sont comme suit [12]:

- **Les Mono-rotor** : Ils sont caractérisés par l'utilisation d'un seul rotor comme actionneur principal. Dans cette catégorie, nous trouvons essentiellement les convertibles.
- **Les birotors** : Il existe plusieurs types de configurations à deux rotors tels que l'hélicoptère classique composé d'un rotor principal et d'un rotor en queue. Les appareils sans plateaux cycliques (autre que les hélicoptères notamment) utilisent des ailerons pour faire pivoter les rotors. Il existe aussi des appareils possédant deux rotors sur le même axe tournant dans des sens opposés et des ailerons qui baignent dans le flux d'air de ces rotors.

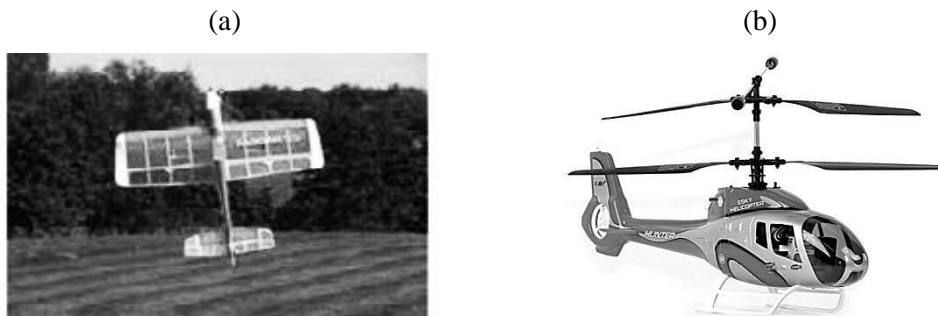


Figure I.3. (a) Mono rotor; (b) Birotor.

- **Les tris rotor** : Moins performant en vol que le quadrotor, le tri-rotor (figure I.4 a) est constitué de deux rotors à l'avant qui tournent dans des sens opposés pour modifier le tangage et d'un rotor en arrière pour régler le roulis.
- **Les Quadrotors** : Un quadrotor (figure I.4 b) est un engin volant doté de quatre rotors placés aux extrémités d'une armature en croix. Ces quatre rotors lui fournissent la force verticale (portance) qui lui permet de décoller.

Son principe de fonctionnement est comme suit : Deux rotors de même axe tournent dans le sens horaire tandis que les deux autres tournent dans le sens inverse. Ses mouvements possibles sont le Gaz (montée ou descente verticale), le Roulis ou Tangage qui est une orientation que le quadrotor prend, et le lacet qui est une rotation du quadrotor autour de lui-même.

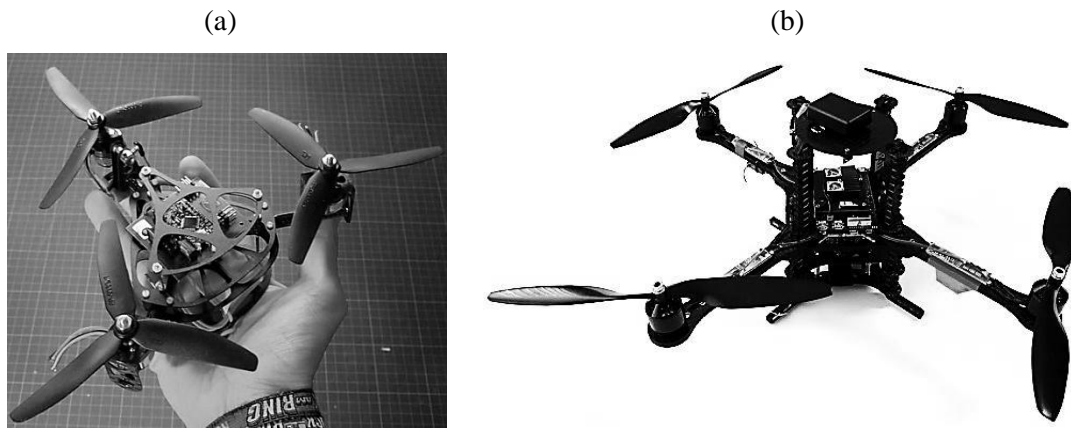


Figure I.4. (a) Tri rotor; (b) Quadrotor.

Tous ses mouvements possibles seront détaillés dans la partie de modélisation La conception du quadrotor offre des avantages par rapport à d'autres plateformes, à savoir :

- Sa simple mécanique.
- Sa manœuvrabilité élevée.
- Son décollage et atterrissage verticaux.

Une telle plateforme est commandée en variant seulement la vitesse de rotation des quatre moteurs. Son aptitude de voler près d'un obstacle, alors que pour d'autres configurations conventionnelles ce comportement est difficile.

L'agilité et l'interconnexion du quadrotor offrent une bonne dynamique et ses quatre rotors peuvent permettre une charge utile accrue.

Par contre, le quadrotor reste limité en termes de durée d'autonomie qui est due à la forte consommation de l'énergie électrique. En plus le décollage de cet engin nécessite de fortes considérations sur le poids et la répartition des charges.

I.4. Application des drones.

Pour les applications militaires, il est généralement secret et sensible. UAV est utilisé pour [13]:

- Maximiser la protection de la vie des soldats grâce à un accès en temps réel à l'information.
- Les décisions tactiques sont prises non seulement en apportant des changements majeurs dans la façon dont les opérations aériennes sont conçues, mais aussi en modifiant d'avantage les principes de base des opérations conjointes dans la conception.
- Tâches principales telles que la reconnaissance, l'espionnage et l'écoute.

Dans le domaine civil, ils sont largement utilisés dans diverses applications [14]:

- Recherche et sauvetage dans des endroits dangereux inaccessibles tels que les montagnes, les océans et les usines chimiques.
- Identifier les limites, évaluer les dommages, surveiller les incendies de forêt, les lignes électriques à haute tension, survoler les zones reculées, montagneuses et inaccessibles, les zones de pêche et les routes maritimes fréquentées et dangereuses, les niveaux de pollution atmosphérique ou marine, la surveillance du trafic routier, les cartes des itinéraires terrestres et la cartographie.
- Surveillance des oléoducs et des gazoducs, surveillance des marchandises dangereuses, des sources d'eau, des vestiges archéologiques, des matières premières ou des combustibles.

I.5. Instrumentation des UAVs.

Ces dernières années, le développement des aéronefs sans pilote a connu une croissance. Ceci est dû aux énormes progrès réalisés dans le domaine des capteurs et des cartes électroniques, devenues de plus en plus portables et efficaces, et le développement des batteries a également permis de réduire considérablement le poids. Pour une plus grande autonomie.

- **Autopilote**

Le pilote automatique forme le centre nerveux du drone, et il garantit au moins sa stabilité (dans ce cas, il est simplement appelé le «contrôleur») et même la navigation. Désormais, le pilote automatique peut être placé dans la paume de votre main, et il résume les progrès de la miniaturisation et de la fiabilité de la robotique au cours des 20 dernières années.

Le pilote automatique se présente sous la forme d'un ou deux circuits imprimés. Il contient le processeur principal, les capteurs, la mémoire et de nombreuses interfaces, y compris le module GPS nécessaire à la navigation. Il est situé entre le récepteur radio et l'actionneur, solidement fixé sur le châssis et le plus près possible du centre de gravité.

Il est programmé avec un logiciel contenant toutes les informations nécessaires à la gestion du vol [15].

- **Les batteries**

L'un des principaux problèmes rencontrés dans la navigation en drone est la durée du vol, la recherche énergétique et son économie sont donc très importantes, ce qui passe notamment par le choix des batteries et des moteurs électriques. Hélice. La batterie doit avoir la densité d'énergie la plus élevée possible et un taux de décharge élevé. Cette qualité rend les batteries au lithium polymère uniques.

- **La propulsion**

Les drones utilisent des outils de propulsion, tels que des moteurs électriques commerciaux

ou des moteurs thermiques de petite cylindrée. Ces aéronefs sont également issus de la fabrication de modèles d'avions.

- **Le moteur électrique**

En raison des mauvaises performances de l'accumulateur (conséquence : la qualité de l'accumulateur embarqué est trop élevée ou l'autonomie est réduite), la motorisation des gros véhicules (air ou terre) est quasi inutile ou inutile, et le moteur électrique a regagné son respect élevé. Propulsion de véhicule, véhicule modèle réduit.

Un grand nombre de modèles réduits d'avions utilisant ces moteurs sont utilisés pour propulser des modèles réduits d'avions, qui sont équipés de pilotes automatiques et de dispositifs de surveillance pour devenir de véritables drones miniatures. Pour ce type d'application, il existe deux types de moteurs :

- le moteur à courant continu avec balais
- le moteur synchrone sans balais ou (brushless)

- **Les hélices**

Les multi-rotors peuvent être utilisés avec des paires d'hélices à pas normal (sens horaire) et à pas inverse (sens antihoraire). Vous devez suivre le type de pas, sinon le multi-rotor basculera pendant le décollage.

Les matériaux utilisés sont classés par ordre croissant de prix et de performances : le nylon (plastique), le bois (généralement le hêtre) et le carbone. L'avantage des hélices en bois est qu'elles sont moins dangereuses en cas de choc. Le métal réservé à l'aviation générale est utilisé pour les gros drones. L'hélice peut être fixée ou pliée pour éviter tout dommage. Il est très pratique de pouvoir plier ces éléments fragiles pendant le transport, ce qui économise beaucoup de place [16].

- **Les capteurs**

Les capteurs sont l'un des composants importants de la stabilité du véhicule et de l'autonomie de croisière, et leur rôle est de convertir des quantités physiques en signaux analogiques, numériques ou logiques. Ils ont quatre normes : plage de mesure, sensibilité, vitesse et précision. Généralement, les drones sont intéressés par les quantités suivantes :

- Position (coordonnées géographiques et altitude).
- Vitesse linéaire (selon les trois axes orthogonaux).
- Vitesse angulaire (vitesse de rotation autour des trois axes orthogonaux).
- Position angulaire ou attitude (roulis, tangage et lacet).

I.6.Modélisation Dynamique du Quadrotor.

Afin de concevoir un contrôleur de vol, il faut d'abord avoir une compréhension approfondie du mouvement de l'aéronef, de sa dynamique et donc des équations dynamiques. Cette compréhension est non seulement nécessaire pour la conception du contrôleur, mais également nécessaire pour s'assurer que la simulation du comportement du véhicule lors de l'application des commandes est aussi proche que possible de la situation réelle. En raison du grand nombre d'effets physiques qui affectent sa dynamique, les quadrotors sont classés comme le système de vol le plus complexe, à savoir les effets aérodynamiques, la gravité, les effets du gyroscope, le frottement et les moments d'inertie [17].

La cause première de cette complexité est que ces effets sont exprimés différemment dans chaque mode de vol. En effet, le modèle dynamique du quadrirotor est basé sur la mission prévue et l'environnement de navigation défini par l'opérateur au préalable [18].

I.6.1. Description générale du quadrirotor

Le quadrotor est un robot mobile aérien avec quatre rotors définis par 6 degrés de liberté dans l'espace [19].

Les quatre rotors sont généralement placés à l'extrémité de la croix, et l'électronique de commande est généralement placée au centre de la croix. Afin d'éviter que le dispositif ne s'ouvre sur son axe de lacet, il est nécessaire de faire tourner deux hélices dans un sens, tandis que les deux autres hélices tournent dans l'autre sens. Afin de pouvoir manipuler le dispositif, chaque paire d'hélices tournant dans le même sens doit être placée aux extrémités opposées d'une branche de la croix.

Le fonctionnement d'un quadrirotor est assez particulier.

En modifiant intelligemment la puissance du moteur, vous pouvez le faire tourner vers le haut / bas, l'incliner à gauche / à droite (rouler) ou basculer vers l'avant / l'arrière (tangage), et même tourner par lui-même (lacet) [20]. Le quadrirotor a six degrés de liberté, trois mouvements de rotation et trois mouvements de translation, seuls quatre déclencheurs doivent être utilisés pour contrôler ces six angles ; il s'agit donc d'un système sous actionné (le nombre d'entrées est inférieur au nombre de sorties).

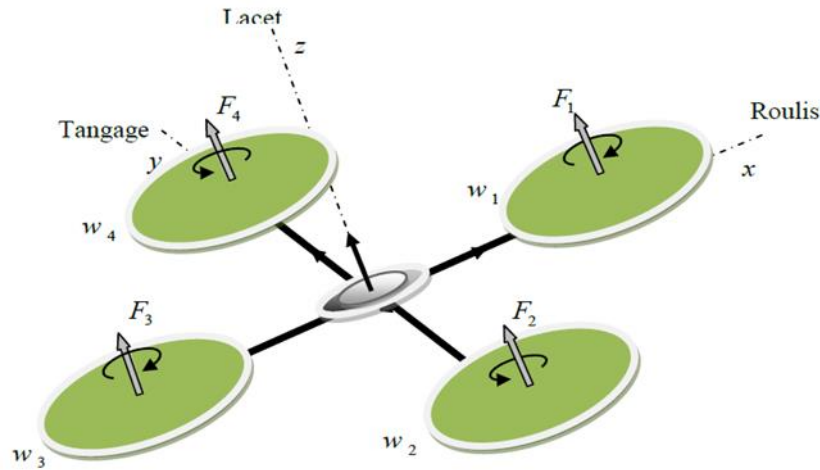


Figure I.5 : Structure générale d'un quadrotor.

I.6.2 Les possibilités de vol du quadrotor.

Dans un hélicoptère traditionnel, lorsque le rotor principal tourne, il génère un couple de réaction qui, si le couple n'est pas perturbé, fera tourner le corps de l'hélicoptère dans le sens opposé. Cela se fait généralement en ajoutant un rotor de queue qui générera une poussée dans la direction latérale. Cependant, le rotor et son alimentation associée ne contribuent pas à la poussée. D'autre part, dans le cas d'un quadrotor, les rotors droit et gauche tournent dans le sens des aiguilles d'une montre, tandis que les rotors avant et arrière tournent dans des sens opposés, ce qui compense efficacement l'excès de couple de réaction et permet au véhicule de planer sans caler. Contrôler. De plus, contrairement aux hélicoptères conventionnels, toute l'énergie utilisée pour contrer le mouvement de rotation contribue à la poussée [21].

En modifiant la vitesse de chaque rotor, modifiant ainsi la poussée générée, un mouvement de base à quatre rotors peut être obtenu. Le quadrotor s'incline en direction du rotor lent et permet alors la translation selon cet axe. Par conséquent, comme dans un hélicoptère traditionnel, le mouvement est couplé, ce qui signifie que le quadrotor ne peut pas translater sans roulis ni tangage, ce qui signifie que le changement de vitesse du rotor amène le mouvement à atteindre au moins trois degrés de liberté. Par exemple, l'augmentation de la vitesse du propulseur gauche provoquera un mouvement de roulis (le rotor du quadrotor s'incline lentement vers la droite) et un mouvement de lacet (l'équilibre entre la rotation horaire du rotor). La rotation inverse sera perturbée, entraînant un mouvement de rotation horizontale), et une translation (le mouvement de roulement entraîne l'inclinaison du cadre et la direction de la force de poussée qui l'accompagne). Ce couplage est la raison pour laquelle nous pouvons contrôler six degrés de liberté à quatre rotors avec seulement quatre entraînements (le couple appliqué par le moteur électrique à chaque hélice).

Le quadrotor a cinq mouvements principaux :

- Mouvement vertical

- Mouvement de roulis
- Mouvement de tangage
- Mouvement de lacet
- Translations horizontales

1-Mouvement vertical

Afin de planer, la portance totale ne doit être que le long de l'axe z , qui est complètement opposé à la gravité. De plus, la portance générée par chaque rotor doit être égale pour éviter que le véhicule ne se renverse davantage. Par conséquent, la poussée générée par chaque rotor doit être la même.

Le mouvement vers le haut et vers le bas est obtenu en changeant la vitesse de rotation du moteur électrique (donc la poussée générée), si la force de levage est supérieure au poids du quadrotor, alors le mouvement vers le haut, et si la force de levage est inférieure au poids du rotor quadripolaire. Mouvement quadrotor vers le bas.

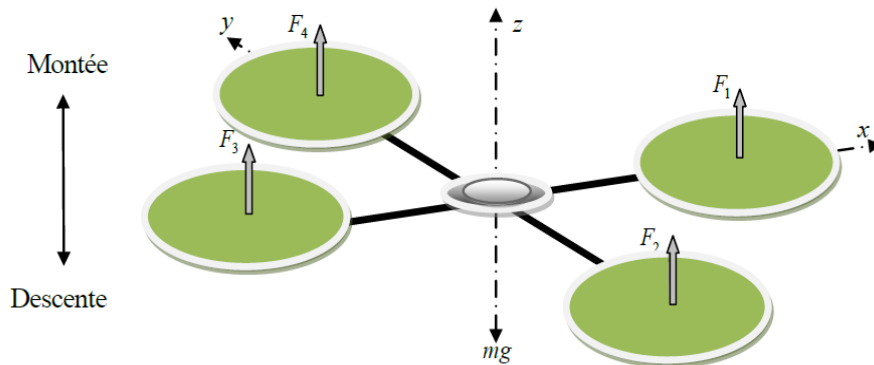


Figure I.6 : Illustration du mouvement vertical.

2-Mouvement de roulis

La figure (I.7) montre comment obtenir un mouvement de défilement. Dans ce cas, le couple est appliqué autour de l'axe x , c'est-à-dire en appliquant une différence de poussée entre le rotor 2 et le rotor 4.

Ce mouvement (rotation autour de l'axe x) est couplé au mouvement de translation le long de l'axe y .

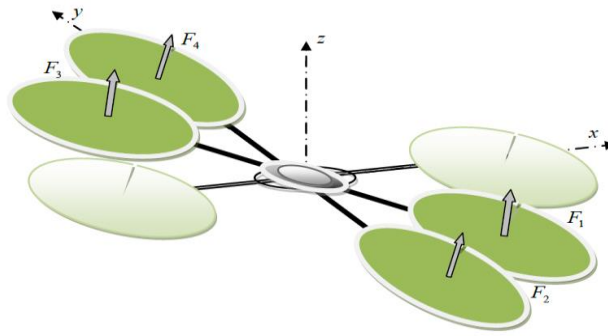


Figure I.7 : Illustration du mouvement de roulis.

3-Mouvement de tangage

La figure (I.8) montre comment obtenir un mouvement de défilement. Dans ce cas, le couple est appliqué autour de l'axe x, c'est-à-dire en appliquant une différence de poussée entre le rotor 1 et le rotor 3.

Ce mouvement (rotation autour de l'axe y) est couplé au mouvement de translation le long de l'axe x.

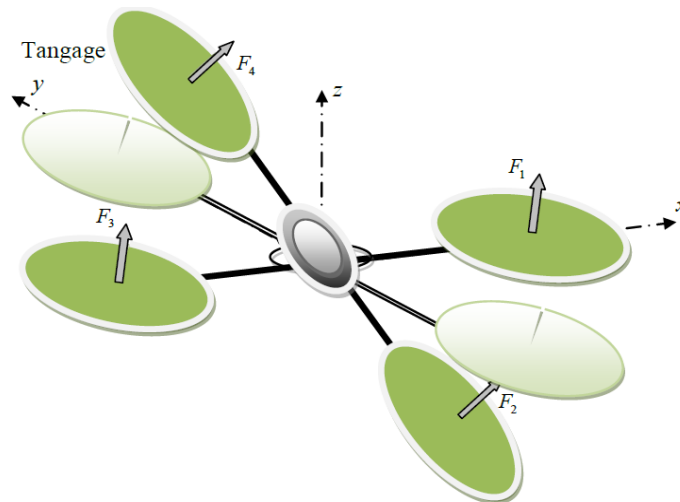


Figure I.8 : Illustration du mouvement de tangage.

4-Mouvement de lacet

La figure (I.9) montre comment obtenir le mouvement de lacet. Dans ce cas, nous voulons appliquer un couple autour de l'axe z, ce qui se fait en appliquant une différence de vitesse entre le rotor {1,3} et {2,4}. Ce mouvement n'est pas une conséquence directe de la poussée générée par l'hélice, mais une conséquence directe du couple de réaction généré par la rotation du rotor.

La direction de la poussée ne bougera pas pendant le mouvement, mais l'augmentation de la force de levage dans une paire de rotors doit être égale à la diminution des autres rotors pour garantir que toutes les poussées restent les mêmes.

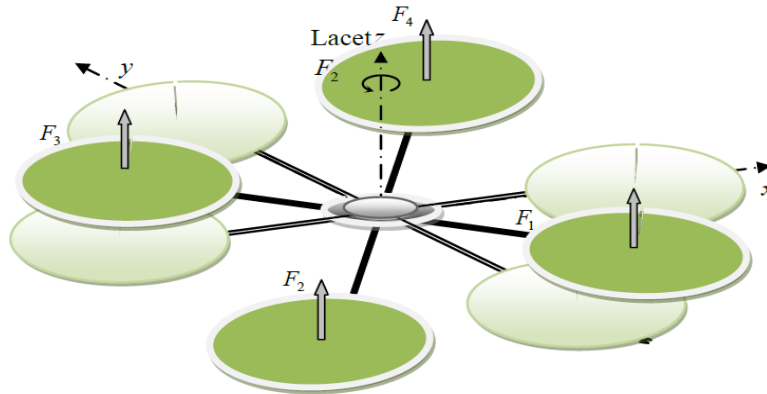


Figure I.9 : Illustration du mouvement de lacet.

5-Mouvements de translation

La figure (I.10) montre comment effectuer une translation horizontale. Dans ce cas, nous voulons appliquer une force le long de x ou y, ce qui se fait en inclinant l'objet (en tanguant ou en roulant) et en augmentant toute la poussée générée pour garder la composante z de la poussée d'égale importance.

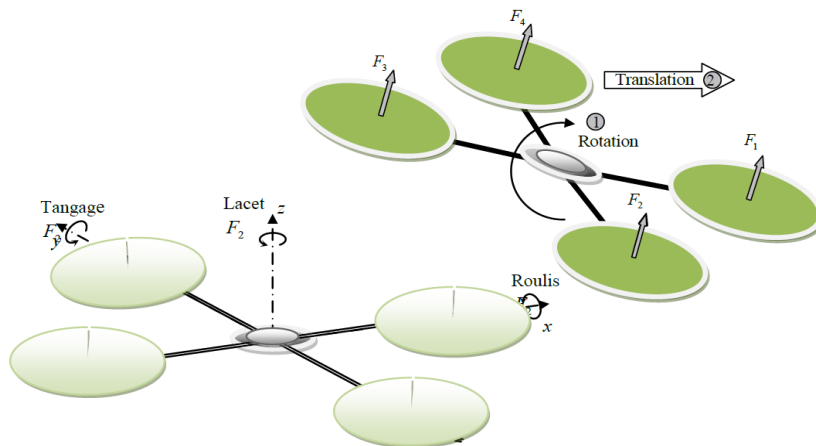


Figure I.10 : Illustration du mouvement de translation.

I.6.3. Modélisation du Quadrotor.

Modéliser un robot volant est une tâche difficile car la dynamique du système est hautement non linéaire et entièrement couplée. Afin de mieux comprendre le modèle dynamique développé ci-dessous, voici différentes hypothèses de travail :

- On suppose que la structure du quadcoptère est rigide et symétrique, ce qui signifie que la matrice d'inertie sera supposée diagonale
- L'hélice est considérée comme rigide de sorte que les effets de sa déformation lors de la rotation peuvent être ignorés.

- Le centre de gravité coïncide avec l'origine du système de coordonnées lié à la structure.
- La portance et la traînée sont proportionnelles au carré de la vitesse du rotor, qui est très proche du comportement aérodynamique.

Pour évaluer le modèle mathématique du quadricoptère, nous utilisons deux datums, un datum fixe connecté à la terre R^m et l'autre connecté à un appareil mobile R^b . Le passage entre le cadre mobile et le cadre fixe est donné par une matrice appelée matrice de transformation T, qui contient la direction et la position du cadre mobile par rapport au cadre fixe.

On choisit la convention d'axes suivants

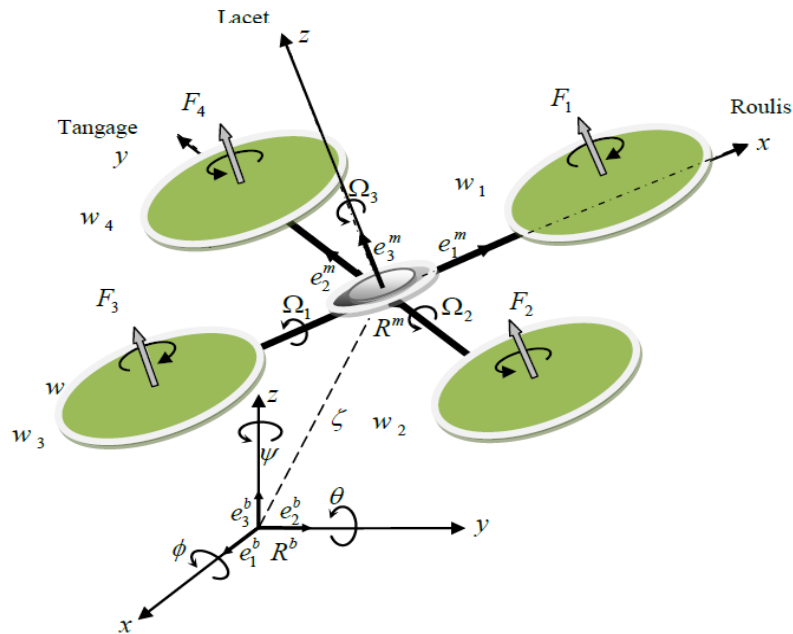


Figure I.11 : Géométrie du quadrirotor.

$$T = \begin{bmatrix} R & e \\ 0 & 1 \end{bmatrix} \quad (\text{I-1})$$

Avec R la matrice de rotation (décrit l'orientation de l'objet mobile), $e=[x \ y \ z]^T$ est le vecteur de position. Pour déterminer les éléments de la matrice de rotation R, on utilise les angles d'Euler.

I.6.4. Matrice de rotation.

La matrice de rotation est une matrice de déterminant égal 1 $\det(R)=1$ avec $RR^T=R^T R=I$, la matrice de rotation est le produit de 3 rotations roll, pitch et yaw (Euler angles)

- Roll est une rotation de l'axe x d'un angle ϕ avec : $-\pi/2 < \phi < \pi/2$.
- Pitch est une rotation de l'axe y d'un angle θ avec : $-\pi/2 < \theta < \pi/2$.

- Yaw est une rotation de l'axe z d'un angle ψ avec : $-\pi < \psi < \pi$.

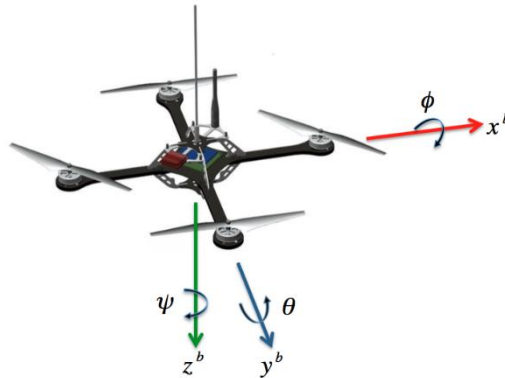


Figure I.12 : les axes de rotation

$$R(\psi, \theta, \varphi) = \text{rot}(z, \psi) \cdot \text{rot}(y, \theta) \cdot \text{rot}(x, \varphi)$$

- Rotation de l'axe x :
$$\text{rot}(x, \varphi) = \begin{bmatrix} 1 & 0 & 0 \\ 0 & \cos(\varphi) & -\sin(\varphi) \\ 0 & \sin(\varphi) & \cos(\varphi) \end{bmatrix} \quad (\text{I-2})$$

- Rotation de l'axe y :
$$\text{rot}(y, \theta) = \begin{bmatrix} \cos(\theta) & 0 & \sin(\theta) \\ 0 & 1 & 0 \\ -\sin(\theta) & 0 & \cos(\theta) \end{bmatrix} \quad (\text{I-3})$$

- Rotation de l'axe Z :
$$\text{rot}(z, \varphi) = \begin{bmatrix} \cos(\psi) & -\sin(\psi) & 0 \\ \sin(\psi) & \cos(\psi) & 0 \\ 0 & 0 & 1 \end{bmatrix} \quad (\text{I-4})$$

Alors la matrice de rotation est :

$$R(\psi, \theta, \varphi) = \begin{bmatrix} c\psi c\theta & s\psi s\theta c\varphi - s\psi c\varphi & c\psi s\theta c\varphi + s\psi s\varphi \\ s\psi c\theta & s\psi s\theta s\varphi + c\psi c\varphi & c\psi s\theta s\varphi - s\psi c\varphi \\ -s\theta & s\varphi c\theta & c\varphi c\theta \end{bmatrix} \quad (\text{I-5})$$

Avec c est cos et s est sin

I.6.5. Mouvement de translation et de rotation.

I.6.5.1 Effets physiques agissants sur le quadrotor.

Les forces :

Les forces agissant sur le système sont :

- **Le poids du quadrotor** : il est donné par $P = mg$, où : m est la masse totale et g la gravité.

- **Les forces de poussée** : qui sont des forces provoquées par la rotation des moteurs, elles sont perpendiculaires sur le plan des hélices. Ces forces sont proportionnelles au carré de la vitesse de rotation des moteurs :

$$F_i = b\omega_i^2 \quad (\text{I-6})$$

Avec $i = \overline{1:4}$, et b est le coefficient de portance, il dépend de la forme et le nombre des pales et la densité de l'air.

- **Les forces de traînée** : la force de traînée est le couplage entre une force de pression et la force de frottement visqueux, dans ce cas on a deux forces de traînée agissant sur le système qu'elles sont :
 - La traînée dans les hélices : elle agisse sur les pales, elle est proportionnelle à la densité de l'air, à la forme des pales et au carré de la vitesse de rotation de l'hélice, elle est donnée par la relation suivante :

$$T_h = d\omega^2 \quad (\text{I-7})$$

Avec d est le coefficient de drag il dépend de la fabrication de l'hélice.

- La traînée selon les axes (x, y, z) : elle est due au mouvement du corps du quadrirotor

$$F_t = K_{ft}v \quad (\text{I-8})$$

Avec : K_{ft} le coefficient de traînée de translation et v la vitesse linéaire.

Les moments :

Il y a plusieurs moments agissants sur le quadrirotor, ces moments sont dus aux forces de poussée et de traînée et aux effets gyroscopiques.

- **Moments dus aux forces de poussée** :
 - La rotation autour de l'axe x : elle est due au moment créé par la différence entre les forces de portance des rotors 2 et 4, ce moment est donné par la relation suivante :

$$M_x = l(F_4 - F_2) = lb(\omega_4^2 - \omega_2^2) \quad (\text{I-9})$$

Avec l est la longueur du bras entre le rotor et le centre de gravité du quadrirotor.

- La rotation autour de l'axe y : elle est due au moment créé par la différence entre les forces de portance des rotors 1 et 3, ce moment est donné par la relation suivante :
- **Moments dus aux forces de traînée** :

- La rotation autour de l'axe z : elle est due à un couple réactif provoqué par les couples de traînée dans chaque hélice, ce moment est donné par la relation suivante :

$$M_z = d(\omega_1^2 - \omega_2^2 + \omega_3^2 - \omega_4^2) \quad (\text{I-10})$$

- Moment résultant des frottements aérodynamiques, il est donné par :

$$M_a = K_{fa}\Omega^2 \quad (\text{I-11})$$

Avec, K_{fa} : Le coefficient des frottements aérodynamiques et Ω est la vitesse angulaire.

I.6.5.2 Effet gyroscopique.

L'effet gyroscopique est défini comme la difficulté de changer la position ou l'orientation du plan de rotation de la masse en rotation. L'effet gyroscopique est nommé en fonction du mode de fonctionnement du gyroscope. Le gyroscope est un dispositif de contrôle de mouvement utilisé dans l'aviation (du grec gyroscope, qui signifie tourner et observer).

Dans notre cas, il y a deux moments gyroscopiques, le premier est le moment gyroscopique de l'hélice, et l'autre est le moment gyroscopique généré par le mouvement du rotor à quatre axes.

- Moment gyroscopique des hélices : il est donné par la relation suivante :

$$M_{gh} = \sum_1^4 \Omega \wedge J_r \begin{bmatrix} 0 & 0 & (-1)^{i+1} & \omega_i \end{bmatrix}^T \quad (\text{I-12})$$

Avec J_r est l'inertie des rotors.

- Moment gyroscopique dû aux mouvements de quadrirotor : il est donné par la relation suivante :

$$M_{gm} = \Omega \wedge J \Omega \quad (\text{I-13})$$

Avec J est l'inertie de system.

D'après la formulation de Newton-Euler, on a :

$$\begin{cases} \dot{\zeta} = v \\ m\ddot{\zeta} = F_f + F_t + F_g \\ \dot{R} = RS(\Omega) \\ J\dot{\Omega} = -M_{gm} + M_f - M_a - M_{gh} \end{cases} \quad (\text{I-14})$$

Avec

ζ : Est le vecteur de position du quadrirotor

m : la masse totale du quadrirotor

Ω : La vitesse angulaire exprimée dans le repère fixe

R : La matrice de rotation

\wedge : Le produit vectoriel

J : matrice d'inertie symétrique de dimension (3×3) , elle est donnée par :

$$J = \begin{bmatrix} I_x & 0 & 0 \\ 0 & I_y & 0 \\ 0 & 0 & I_z \end{bmatrix} \quad (\text{I-15})$$

$S(\Omega)$: Est la matrice antisymétrique ; pour un vecteur de vitesse $\Omega = [\Omega_1 \quad \Omega_2 \quad \Omega_3]^T$, elle est donnée par :

$$S(\Omega) = \begin{bmatrix} 0 & -\Omega_3 & \Omega_2 \\ \Omega_3 & 0 & -\Omega_1 \\ -\Omega_2 & \Omega_1 & 0 \end{bmatrix} \quad (\text{I-16})$$

F_f : Est la force totale générée par les quatre rotors, elle est donnée par :

$$F_f = R \times \left[0 \quad 0 \quad \sum_{i=1}^4 F_i \right]^T \quad (\text{I-17})$$

$$F_i = b\omega_i^2 \quad (\text{I-18})$$

F_t : La force de traînée selon les axes (x, y, z) , elle est donnée par :

$$F_t = \begin{bmatrix} -K_{ftx} & 0 & 0 \\ 0 & -K_{fyy} & 0 \\ 0 & 0 & -K_{ftz} \end{bmatrix} \dot{\zeta} \quad (\text{I-19})$$

$K_{ftx}, K_{fyy}, K_{ftz}$: Les coefficients de traînée de translation,

F_g : Force de gravité, elle est donnée par :

$$F_g = \begin{bmatrix} 0 \\ 0 \\ -mg \end{bmatrix} \quad (\text{I-20})$$

M_f : Moment provoqué par les forces de poussée et de traînée.

$$M_f = \begin{bmatrix} l(F_4 - F_2) \\ l(F_3 - F_1) \\ d(\omega_1^2 - \omega_2^2 + \omega_3^2 - \omega_4^2) \end{bmatrix} \quad (\text{I-21})$$

M_a : Moment résultant des frottements aérodynamiques, il est donnée par :

$$M_a = \begin{bmatrix} K_{fax} \dot{\phi}^2 \\ K_{fay} \dot{\theta}^2 \\ K_{faz} \dot{\psi}^2 \end{bmatrix} \quad (\text{I-22})$$

$K_{fax}, K_{fay}, K_{faz}$: Les coefficients des frottements aérodynamiques.

I.6.5.3 Equations de mouvement de translation.

On a :

$$m\ddot{\zeta} = F_f + F_t + F_g \quad (\text{I-23})$$

On remplace chaque force par sa formule, on trouve :

$$m \begin{bmatrix} \ddot{x} \\ \ddot{y} \\ \ddot{z} \end{bmatrix} = \begin{bmatrix} c\phi c\psi s\theta + s\phi s\psi \\ c\phi s\theta s\psi - s\phi c\psi \\ c\phi c\theta \end{bmatrix} \sum_{i=1}^4 F_i - \begin{bmatrix} K_{fmx} \dot{x} \\ K_{fmy} \dot{y} \\ K_{fmz} \dot{z} \end{bmatrix} - \begin{bmatrix} 0 \\ 0 \\ mg \end{bmatrix} \quad (\text{I-24})$$

On obtient alors les équations différentielles qui définissent le mouvement de translation :

$$\begin{cases} \ddot{x} = \frac{1}{m} (c\phi c\psi s\theta + s\phi s\psi) \left(\sum_{i=1}^4 F_i \right) - \frac{K_{fmx}}{m} \dot{x} \\ \ddot{y} = \frac{1}{m} (c\phi s\theta s\psi - s\phi c\psi) \left(\sum_{i=1}^4 F_i \right) - \frac{K_{fmy}}{m} \dot{y} \\ \ddot{z} = \frac{1}{m} (c\phi c\theta) \left(\sum_{i=1}^4 F_i \right) - \frac{K_{fmz}}{m} \dot{z} - g \end{cases} \quad (\text{I-25})$$

I.6.5.4 Equations de mouvement de rotation.

On a :

$$J\dot{\Omega} = -M_{gh} - M_{gm} - M_a + M_f \quad (\text{I-26})$$

On remplace chaque moment par la formule correspondant, on trouve :

$$\begin{bmatrix} I_x & 0 & 0 \\ 0 & I_y & 0 \\ 0 & 0 & I_z \end{bmatrix} \begin{bmatrix} \ddot{\phi} \\ \ddot{\theta} \\ \ddot{\psi} \end{bmatrix} = - \begin{bmatrix} \dot{\phi} \\ \dot{\theta} \\ \dot{\psi} \end{bmatrix} \wedge \left(\begin{bmatrix} I_x & 0 & 0 \\ 0 & I_y & 0 \\ 0 & 0 & I_z \end{bmatrix} \begin{bmatrix} \dot{\phi} \\ \dot{\theta} \\ \dot{\psi} \end{bmatrix} \right) - \begin{bmatrix} J_r \Omega_r \dot{\theta} \\ -J_r \Omega_r \dot{\phi} \\ 0 \end{bmatrix} - \begin{bmatrix} K_{fax} \dot{\phi}^2 \\ K_{fay} \dot{\theta}^2 \\ K_{faz} \dot{\psi}^2 \end{bmatrix} + \begin{bmatrix} lb(\omega_4^2 - \omega_2^2) \\ lb(\omega_3^2 - \omega_1^2) \\ d(\omega_1^2 - \omega_2^2 + \omega_3^2 - \omega_4^2) \end{bmatrix} \quad (\text{I-27})$$

On obtient alors les équations différentielles définissant le mouvement de rotation

$$\begin{cases} I_x \ddot{\phi} = -\dot{\theta}\dot{\psi}(I_z - I_y) - J_r \bar{\Omega}_r \dot{\theta} - K_{fax} \dot{\phi}^2 + lb(\omega_4^2 - \omega_2^2) \\ I_y \ddot{\theta} = \dot{\phi}\dot{\psi}(I_z - I_x) + J_r \bar{\Omega}_r \dot{\phi} - K_{fay} \dot{\theta}^2 + lb(\omega_3^2 - \omega_1^2) \\ I_z \ddot{\psi} = -\dot{\phi}\dot{\theta}(I_y - I_x) - K_{faz} \dot{\psi}^2 + d(\omega_1^2 - \omega_2^2 + \omega_3^2 - \omega_4^2) \end{cases} \quad (\text{I-28})$$

Avec :

$$\bar{\Omega}_r = \omega_1 - \omega_2 + \omega_3 - \omega_4 \quad (\text{I-29})$$

En conséquence, le modèle dynamique complet qui régit le quadrirotor est le suivant :

$$\begin{cases} \ddot{\phi} = \frac{(I_y - I_z)}{I_x} \dot{\theta}\dot{\psi} - \frac{J_r}{I_x} \bar{\Omega}_r \dot{\theta} - \frac{K_{fax}}{I_x} \dot{\phi}^2 + \frac{l}{I_x} u_2 \\ \ddot{\theta} = \frac{(I_z - I_x)}{I_y} \dot{\phi}\dot{\psi} + \frac{J_r}{I_y} \bar{\Omega}_r \dot{\phi} - \frac{K_{fay}}{I_y} \dot{\theta}^2 + \frac{l}{I_y} u_3 \\ \ddot{\psi} = \frac{(I_x - I_y)}{I_z} \dot{\theta}\dot{\phi} - \frac{K_{fz}}{I_z} \dot{\psi}^2 + \frac{1}{I_z} u_4 \\ \ddot{x} = -\frac{K_{fx}}{m} x^i + \frac{1}{m} u_x u_1 \\ \ddot{y} = -\frac{K_{fy}}{m} y^j + \frac{1}{m} u_y u_1 \\ \ddot{z} = -\frac{K_{ft}}{m} z - g + \frac{\cos(\phi)\cos(\theta)}{m} u_1 \end{cases} \quad (\text{I-30})$$

Avec :

$$\begin{cases} u_x = (c\phi c\psi s\theta + s\phi s\psi) \\ u_y = (c\phi s\theta s\psi - s\phi c\psi) \end{cases} \quad (\text{I-31})$$

Et :

$$\begin{bmatrix} u_1 \\ u_2 \\ u_3 \\ u_4 \end{bmatrix} = \begin{bmatrix} b & b & b & b \\ 0 & -lb & 0 & lb \\ -lb & 0 & lb & 0 \\ d & -d & d & -d \end{bmatrix} \begin{bmatrix} \omega_1^2 \\ \omega_2^2 \\ \omega_3^2 \\ \omega_4^2 \end{bmatrix} \quad (\text{I-32})$$

I.6.6. Les contraintes non holonomes.

Le développement de contraintes non homonymiques permet de clarifier le couplage entre différents états du système. À partir des équations de la mécanique du mouvement transrationnel z , y et x , nous pouvons extraire l'expression de contraintes non homonymiques, comme indiqué ci-dessous :

$$\begin{cases} \phi_d = \arcsin(u_x \sin(\psi_d) - u_y \cos(\psi_d)) \\ \theta_d = \arcsin\left(\frac{(u_x \cos(\psi_d) + u_y \sin(\psi_d))}{\cos(\phi_d)}\right) \end{cases} \quad (\text{I-33})$$

I.6.7. Dynamique du rotor.

Le rotor est un ensemble constitué d'un moteur à courant continu entraînant une hélice via un réducteur. Le moteur à courant continu est régi par la dynamique suivante :

$$\left\{ \begin{array}{l} u(t) = e(t) + R.i(t) + L.\frac{di(t)}{dt} \\ e(t) = Ke.\omega_m(t) \\ J.\frac{d\omega_m}{dt} = C_m(t) - C_r(t) - f.\omega_m(t) \\ C_m(t) = K_t.i(t) \end{array} \right. \quad (\text{I-34})$$

Avec :

$u(t)$ = Tension du moteur [V]

$e(t)$ = Force contre électromotrice du moteur [V]

$i(t)$ = Intensité dans le moteur [A]

$C_m(t)$ = Couple exercé par le moteur [N.m]

$C_r(t)$ = Couple résistant sur l'axe moteur [N.m]

$\omega_m(t)$ = Vitesse angulaire du moteur [rad/s]

R = Valeur de la résistance [Ω]

L = Valeur de l'inductance [H]

Ke = Coefficient de la force contre électromotrice [V/(rad/s)]

J = Inertie équivalente ramenée sur l'arbre moteur [kg.m²]

$f = 0,01$ = Paramètre de « frottement fluide » total [N.m.s]

K_t = Constante de couple [N.m/A]

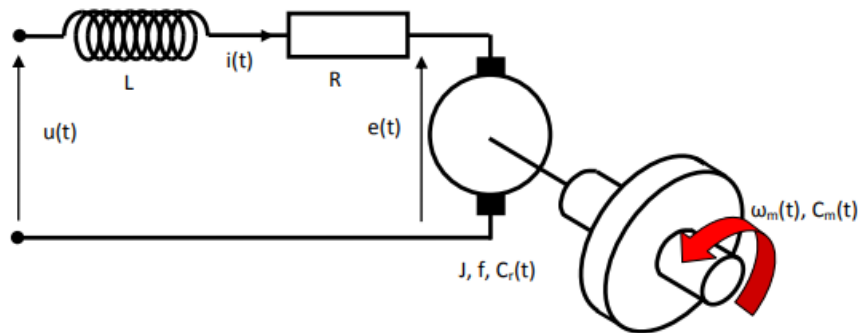


Figure I.13 : schéma de rotor (mcc)

I.6.8. Représentation d'état.

Pour un système physique il existe une multitude de représentations d'état, dans notre cas on choisit le vecteur d'état comme suit [22] :

$$\begin{aligned} X &= [\phi \quad \dot{\phi} \quad \theta \quad \dot{\theta} \quad \psi \quad \dot{\psi} \quad x \quad \dot{x} \quad y \quad \dot{y} \quad z \quad \dot{z}]^T \\ &= [x_1 \quad x_2 \quad x_3 \quad x_4 \quad x_5 \quad x_6 \quad x_7 \quad x_8 \quad x_9 \quad x_{10} \quad x_{11} \quad x_{12}]^T \end{aligned}$$

On obtient la représentation d'état suivante :

$$\left\{ \begin{array}{l}
 \dot{x}_1 = x_2 \\
 \dot{x}_2 = a_1 x_4 x_6 + a_2 x_2^2 + a_3 \Omega_r x_4 + b_1 u_2 \\
 \dot{x}_3 = x_4 \\
 \dot{x}_4 = a_4 x_2 x_6 + a_5 x_4^2 + a_6 \Omega_r x_2 + b_2 u_3 \\
 \dot{x}_5 = x_6 \\
 \dot{x}_6 = a_7 x_2 x_4 + a_8 x_6^2 + b_3 u_4 \\
 \dot{x}_7 = x_8 \\
 \dot{x}_8 = a_9 x_8 + \frac{1}{m} u_x u_1 \\
 \dot{x}_9 = x_{10} \\
 \dot{x}_{10} = a_{10} x_{10} + \frac{1}{m} u_y u_1 \\
 \dot{x}_{11} = x_{12} \\
 \dot{x}_{12} = a_{11} x_{12} + \frac{\cos(\phi)\cos(\theta)}{m} u_1 - g
 \end{array} \right. \quad (\text{I-35})$$

Avec :

$$\left\{ \begin{array}{l}
 a_1 = \frac{(I_y - I_z)}{I_x}, a_2 = -\frac{K_{fax}}{I_x}, a_3 = -\frac{J_r}{I_x}, a_4 = \frac{(I_z - I_x)}{I_y}, a_5 = -\frac{K_{fay}}{I_y}, a_6 = \frac{J_r}{I_y} \\
 a_7 = \frac{(I_x - I_y)}{I_z}, a_8 = -\frac{K_{faz}}{I_z}, a_9 = -\frac{K_{ftx}}{m}, a_{10} = -\frac{K_{fty}}{m}, a_{11} = -\frac{K_{ftz}}{m}, b_1 = \frac{l}{I_x} \\
 b_2 = \frac{l}{I_y}, b_3 = \frac{l}{I_z}.
 \end{array} \right.$$

I.7 Conclusion.

À la fin de ce chapitre, nous pouvons conclure qu'un modèle dynamique universel pour le quadrotor n'existe pas. Cela peut être justifié, d'une part, par la complexité du système en lui-même (modèle non linéaire et sous actionné). D'autre part, une représentation simple de la dynamique du comportement complet du quadrotor, dans tous les modes de vol permettant la synthèse de loi de commande, n'existe pas. Ceci est due au fait que la dynamique résulte principalement de la nature différente des diverses forces aérodynamiques dans les différentes conditions de vol.

En outre, si l'on prend le cas par exemple, de la résultante aérodynamique, cette force dépend des paramètres qui sont en général difficilement quantifiables et qui dépendent fortement des conditions de vol. En d'autres termes, leurs valeurs changent en fonction du mode de vol et du milieu dans lequel évolue le quadrotor. Par conséquent, il est difficile d'obtenir un modèle dynamique sans avoir de bonnes estimations de ces paramètres.

Une voie directe pour y remédier est l'utilisation d'un système d'identification. Dans cette approche, une approximation du modèle est choisie et les coefficients aérodynamiques sont évalués en se basant sur des données expérimentales. La difficulté d'utiliser une telle approche est qu'il faut disposer de matériels appropriés pour réaliser ces mesures. Dans notre cas, la modélisation dynamique présentée est

réalisée sur la base des données d'identification [9] faite sur un prototype expérimental proposé par le LISV de France.

CHAPITRE II

CHAPITRE II

LA LOGIQUE FLOUE POUR LE CONTRÔLE DU QUADROTOR

II.1 Introduction

La logique floue suscite actuellement l'intérêt des chercheurs, des ingénieurs et des industriels. Elle repose sur la théorie des ensembles flous, une évolution de la théorie des ensembles qui remplace le concept booléen selon lequel les entités sont membres ou non d'un ensemble bien défini par la notion d'appartenance "floue" [23], c'est-à-dire que les entités peuvent avoir des "degrés d'appartenance" à un ensemble.

Au cours des trois dernières décennies. Le nombre de travaux de recherche sur le développement des contrôleurs floue ainsi que le nombre d'applications industrielles de ce type de contrôle ont considérablement augmenté.

Dans ce chapitre, nous commencerons par un examen rapide de la théorie des ensembles flous et la structure d'un contrôleur flou. Puis une stratégie de commande basée sur cette technique pour contrôler la dynamique du quadrotor sera proposée. Après cela afin de présenter l'utilité et les avantages de la logique floue dans l'amélioration des performances et des robustesses des contrôleurs conventionnels, deux autres stratégies sera proposée, la première basée sur l'adaptation des gains des contrôleurs PID avec un système expert flou, et le deuxième repos sur le choix d'une surface floue dans la structure de la commande par mode glissante d'un quadrotor.

II.2. Concept de la logique floue

Les bases de la logique floue ont été établies en 1965 par le professeur Lotfi Zadeh de l'université de Berkeley en Californie, mais ce n'est qu'en 1974 que le professeur Mamdani propose la première application industrielle en définissant la structure de base des régulateurs flous et les principes de la

commande floue. Elle a été créée pour répondre au besoin de formaliser la représentation et le traitement de connaissances imprécises ou approximatives [24]. Cette méthode intervient donc dans la manipulation de connaissances imparfaites. Par exemple ; Soit E l'ensemble des tailles possible et A l'ensemble des grandes tailles, comme il est montré sur la Figure II.1. En logique booléenne, on est soit de petite taille, soit de grande taille. Si 170 cm est la frontière entre les deux, on est petit pour $x < 170$ cm et grand pour $x > 170$ cm. Mais si l'on mesure 169 cm, on est petit ! Cette discontinuité est totalement absurde.

Donc, le plus correcte sera de représenter A par un ensemble flou permettant un passage progressif des tailles petites aux tailles grandes.

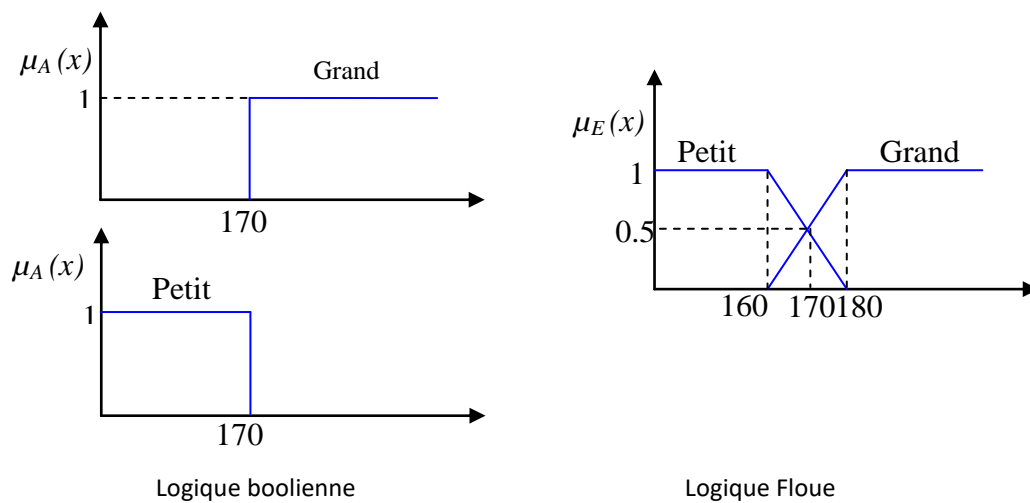


Figure II.1: Représentation de deux variables (grand et petit) par logique booléenne et par logique floue.

Les avantages de la logique floue sont nombreux :

- La robustesse de la commande.
- La possibilité de commande non linéaire.
- La réduction du temps de développement et de maintenance.
- La non nécessité d'une modélisation mathématique rigoureuse du processus.
- La simplicité de définition et de conception.

II.2. 1 Sous-ensembles flous

Dans la logique conventionnelle, l'appartenance d'un élément à un objet est décrite de manière binaire, c'est-à-dire que l'élément peut soit appartenir entièrement ou ne pas appartenir du tout à l'ensemble. On a donc deux possibilités $\{0,1\}$. Dans le cas de la logique floue, un élément appartient à un ensemble avec un degré d'appartenance, qui est compris dans l'intervalle $[0,1]$. Ces deux logiques ne s'opposent pas, au contraire la logique floue est une extension de la logique booléenne pour laquelle les

degrés de vérité (degrés d'appartenance que l'on note μ), au lieu d'être vrai ou faux peuvent prendre des valeurs comprises entre 0 et 1 [25].

La logique floue s'appuie sur la théorie des sous-ensembles flous, permet ainsi d'intégrer et de traiter le caractère approximatif ou vague de la connaissance humaine en utilisant des catégories aux limites mal définies comme celui de 'grande vitesse' ou 'petite taille', des situations intermédiaires entre le vrai et le faux et d'introduire un passage progressif d'une propriété à une autre. Ce qui autorise à des éléments de n'appartenir complètement ni à un ensemble ni à l'autre, ou encore à appartenir partiellement à chacun.

Définition 1 Soit X , un ensemble dénombrable ou non. Un sous-ensemble flou A de X est caractérisé par sa fonction d'appartenance, telle que:

$$\begin{aligned} \mu_A : X &\rightarrow [0,1] \\ x &\rightarrow \mu_A(x) \end{aligned} \tag{II.1}$$

Où : $\mu_A(x)$ représente le degré d'appartenance de x à l'ensemble flou A .

II.2.2. Variables linguistiques

La notion de variable linguistique permet de modéliser les connaissances imprécises ou vagues sur une variable dont la valeur précise est inconnue. Une variable linguistique, ou variable floue, est donc une variable dont les valeurs appartiennent à des ensembles flous pouvant représenter des mots du langage naturel. Ainsi une variable floue peut prendre simultanément plusieurs valeurs linguistiques. Par exemple la variable «Vitesse» peut appartenir aux ensembles flous, comme il est montré sur la Figure II.2 ; Lente, Moyen et Rapide.

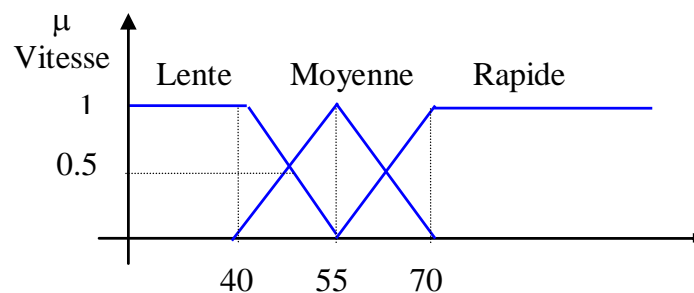


Figure II.2 : Représentation floue de la variable vitesse.

La création du système d'aide à la décision suppose la définition de variables floues ainsi que la détermination des ensembles flous correspondants, un ensemble flou A défini sur l'univers de discours X est noté : Dans le cas discret Dans le cas continu

$$A = \sum_{i=1}^n \frac{\mu_A(x_i)}{x_i} = \frac{\mu_A(x_1)}{x_1} + \dots + \frac{\mu_A(x_n)}{x_n} \quad \text{Dans le cas discret} \quad (\text{II.2})$$

$$A = \int \mu_A(x)/x \quad \text{Dans le cas continu}$$

II.2.3 Fonctions d'appartenances

On attribue à chaque valeur de la variable linguistique des fonctions d'appartenance, dont la valeur varie entre 0 et 1, en tenant compte de la classification en un certain nombre d'ensembles flous. Le plus souvent, on utilise pour les fonctions d'appartenance de forme trapézoïdales ou triangulaires, rectangulaires ou de type singleton. Les fonctions d'appartenance donnée par la **Figure II.3** sont les plus souvent utilisées pour définir les ensembles flous [25] [26]:

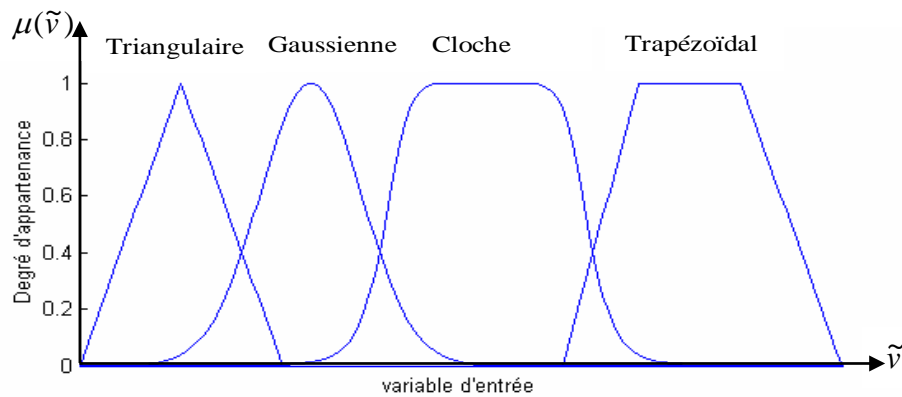


Figure II.3 : Exemple de fonctions d'appartenance.

II.2.4 Opérateurs flous

Les opérateurs de la logique floue sont au nombre de trois, le ET le OU et le NON. Le ET peut être exprimé par la fonction Minimum ou la fonction multiplication. Le OU peut être exprimé par la fonction Maximum ou la fonction addition. Le NON est exprimé par l'expression :

$$\mu_{E\bar{\square}}(x) = 1 - \mu_E(x) \quad (\text{II.3})$$

Avec $\mu_{E\bar{\square}}(x)$ la fonction d'appartenance de l'opérateur NON

II.2.5 Règles d'inférences floues

La prise de décision est basée sur les connaissances humaine ou de l'expert et est présentée sous forme de règles d'inférence floue [27, 28] qui s'exprimes comme suis :

SI (prémisse) **ALORS** (conclusion)

Où

SI (prémisse) **ET** (prémisse) **ALORS** (conclusion)

Les prémisses sont des conditions (entrées) et la conclusion est l'action à effectuer (sorties).

De façon générale, dans la théorie des ensembles classiques, la notion d'appartenance est fondamental, mais elle est de type tout ou rien, un élément appartient ou n'appartient pas à un ensemble. Un tel outil s'avère alors difficilement utilisable lorsqu'il s'agit de manipuler des données vagues, imprécises, contradictoires ou lorsqu'il s'agit de classer des informations suivant des catégories aux frontières mal définies.

Malheureusement aussi, lorsque le système est difficilement modélisable, la conception du contrôleur peut s'avérer très complexe, sinon impossible [29]. La théorie des ensembles flous, par un assouplissement de la notion d'appartenance, permet d'atteindre de tels objectifs. Elle s'avère alors plus adaptée pour la représentation des connaissances qualitatives [29]. Lorsqu'on dispose d'un modèle plus ou moins précis du système à commander, on peut utiliser un contrôleur de structure standard, fixe ou adaptatif, dont les paramètres seront évalués à partir du modèle. Nous présenterons dans la Figure II.8 quelques opérations sur les ensembles flous.

Tableau II.1 : Récapitulatif des opérations flous.

Appellation	ET	OU	NON
Zadeh	$\min(x, y)$	$\max(x, y)$	$1 - x$
Probalistique	$x y$	$x + y - x y$	$1 - x$
Lukasiewicz	$\max(x + y - 1, 0)$	$\min(x + y, 1)$	$1 - x$
Hanacher ($\beta > 0$)	$x y / (\beta + (1 - \beta)(x + y - x y))$	$(x + y + x y - (1 - \beta) x y) / (1 - (1 - \beta) x y)$	$1 - x$
Weber	x si $y = 1$, y si $x = 1$ 0 sinon	x si $y = 0$, y si $x = 0$ 1 sinon	$1 - x$

II.3 Structure d'un contrôleur flou

La structure conventionnelle d'un contrôleur flou est représentée sur la **figure (II.4)**, elle est proposée par Mamdani et se base sur une interface de fuzzification, inférence floue et défuzzification.

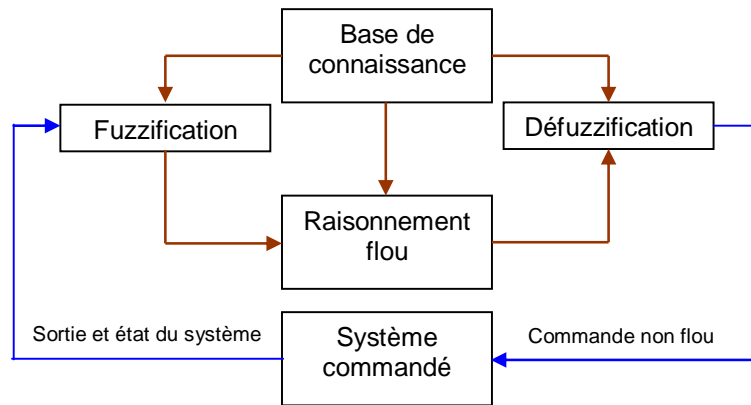


Figure II.4 Configuration générale d'un contrôleur flou.

En résumé, un contrôleur flou comporte les différents éléments suivants [30]:

- Un sous-système d'interface avec le flou, composé en général d'un ensemble de variables linguistiques,
- Une base de connaissances : «base de données» et «base de règles linguistiques de commande»,
- Un sous-système réalisant un raisonnement en utilisant des méthodes issues de la logique floue,
- Un sous-système d'interface avec le non flou, qui fournit la ou les commandes envoyées au système.

le fonctionnement précis de chacun des sous-ensembles est décrit dans la suite dans ce chapitre.

II.3.1 La fuzzification

Le but du module de fuzzification est la transformation des variables d'entrée en variable linguistique. Cette action s'effectue en définissant des fonctions d'appartenances pour les variables d'entrée.

La fuzzification est une étape clé dans tout contrôleur flou, dont la variable linguistique est définie par une expertise, doit respecter un certain nombre de critères afin d'être efficace [31] :

- Chaque terme linguistique est un nombre flou, de noyau non nul et dont la fonction d'appartenance est convexe,
- Les fonctions d'appartenance doivent respecter l'ordre linguistique,
- Les fonctions d'appartenance ne doivent pas trop se superposer. On tolère en général un chevauchement qui ne dépasse pas la mi-hauteur des termes linguistiques consécutifs

II.3.2 L'inférence et règles floue

C'est là que sont appliquées les règles floues définies par le concepteur, elles se basent sur les connaissances et l'expérience du concepteur et sont la partie la plus importante. Le concepteur choisit aussi une méthode d'inférences parmi :

- La méthode max-min.
- La méthode max-produit.
- La méthode somme-produit.

Cette inférence sera appliquée aux différentes règles floues que le concepteur aura posées pour aboutir à une conclusion qui sera transmise au module de défuzzification.

II.3.3. Défuzzification

La défuzzification est l'étape de déduction d'une valeur numérique exploitable à partir de la conclusion obtenue grâce à la règle floue [32]. Il y a trois méthodes de défuzzification :

- Défuzzification par centre de gravité
- Défuzzification par moyenne de maximum
- Défuzzification par valeur maximum

- **Méthode de Centre gravité (centroïde)**

La défuzzification par centre de gravité consiste à calculer l'abscisse du centre de gravité de la fonction d'appartenance donnée par [32]:

$$y_{cg} = \frac{\int y \cdot \mu_{B_{res}}(y) dy}{\int y \cdot B_{res}(y) dy} \quad (\text{II.4})$$

En pratique, on estime le centre de gravité en calculant la moyenne d'un certain nombre de points échantillonnés sur la fonction [26,29]:

$$y_{cg} = \frac{\sum y_i \cdot \mu_{B_{res}}(y_i)}{\mu_{B_{res}}(y_i)} \quad (\text{II.5})$$

Le temps nécessaire au traitement est directement proportionnel au nombre de points retenus pour le calcul de la moyenne. Selon les contraintes fixées par l'application, il y a un compromis à réaliser entre la précision souhaitée et le temps de calcul disponible.

- **Méthode de Moyenne de maxima**

Dans cette méthode, la valeur de sortie est estimée par l'abscisse du point correspondant au centre de l'intervalle pour lequel la fonction d'appartenance est maximale, cette valeur est donnée par l'expression suivante :

$$y_{cm} = \frac{\inf M + \sup M}{2} \quad (\text{II.6})$$

Dans laquelle M est l'ensemble des points pour lesquels la fonction d'appartenance est maximale :

$$M = \{ y \in [-c, c] \mid \mu_{B_{res}}(y) = H(B_{res}) \} \quad (\text{II.7})$$

Dans le cas discret, on explore en fait la liste de tous les points pour lesquels la fonction d'appartenance est maximale afin de trouver le plus petit et le plus grand [32, 38, 64].

- **Méthode de Maximum**

Cette méthode ne s'utilise que dans le cas discret, on choisit comme sortie y_m l'abscisse de la valeur maximale de la fonction d'appartenance résultante μ , lorsque cette fonction d'appartenance est échantillonnée, on prend la moyenne des abscisses du maximum, donnée par :

$$y_m = \frac{\sum_{y_i \in M} y_i}{|M|} \quad (\text{II.8})$$

Pour cela nous emploierons la méthode du centre de gravité ou méthode du centroid qui est donné par la formule suivante [25] :

$$x_R = \frac{\int x \cdot \mu(x) dx}{\int \mu(x) dx} \quad (\text{II.9})$$

II.4. Commande par logique floue du quadrotor

Après la modélisation faite dans le premier chapitre, nous allons adopter une stratégie de commande basée principalement sur la décomposition du système d'origine en deux sous-systèmes ; le premier concerne la commande en position tandis que le second est celui de la commande en orientation ainsi que l'introduction des contraintes non holonomes dans le schéma général de commande. Le schéma ci-dessous présente la stratégie de commande adoptée :

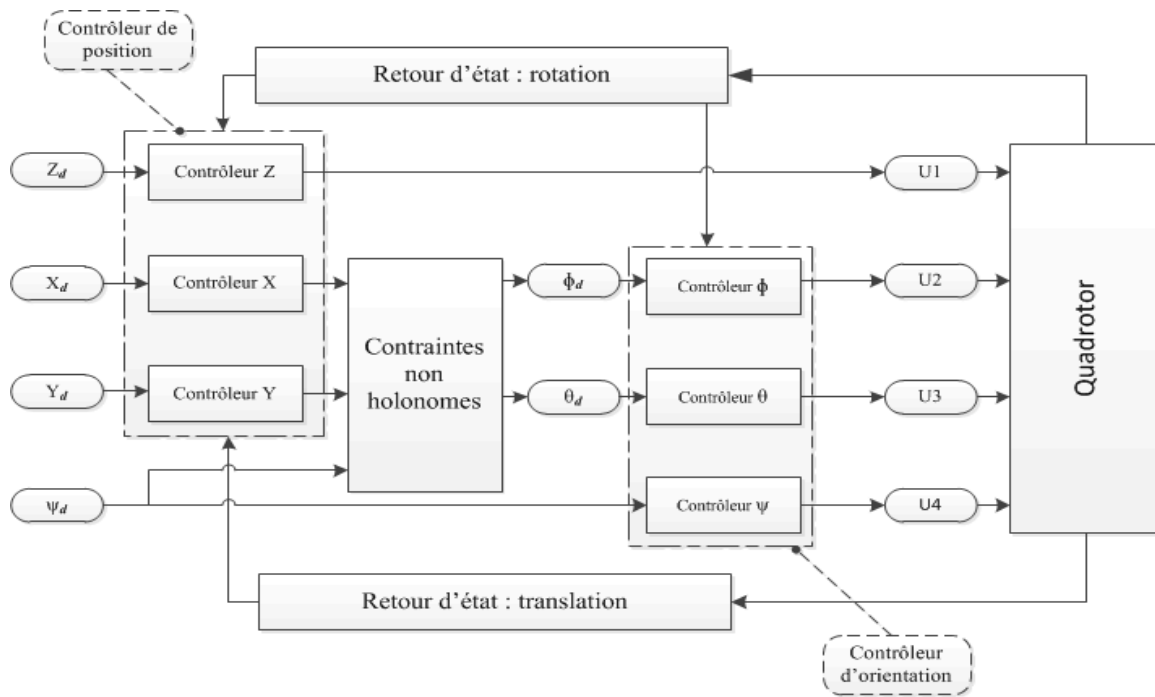


Figure II.5 : Schéma générale de commande du quadrotor.

Tableau II.2 : Paramètres mécaniques et électriques du modèle du quadrotor

Paramètre	Désignation	Valeur
K_p	Coefficient de portance	$2.9842 \times 10^{-5} \text{ N. rad}^2 / \text{s}^2$
K_d	Coefficient de traînée	$3.2320 \times 10^{-7} \text{ N. rad}^2 / \text{s}^2$
m	Masse du quadrotor	486g
d	Distance entre le centre de masse du système et l'axe de rotation des moteurs	25cm
J	Matrice d'inertie du quadrotor	$\text{diag}(3.8278; 3.828; 7.656) \times 10^{-3} \text{ N.m} / \text{rad} / \text{s}^2$
K_{fa}	Coefficients des frottements aérodynamiques	$\text{diag}(5.5670; 5.5670; 6.3540) \times 10^{-4} \text{ N} / \text{rad} / \text{s}$
K_{ft}	Coefficients des forces de traînées selon (X , Y , Z)	$\text{diag}(5.5670; 5.5670; 6.3540) \times 10^{-4} \text{ N} / \text{rad} / \text{s}$
J_r	L'inertie du rotor	$2.8385 \times 10^{-5} \text{ N.m} / \text{rad} / \text{s}^2$
K_e	Constante du couple électrique	0.0216
C_s	Frottement sec	5.3826×10^{-3}
K_r	Constante du couple de charge	3.4629×10^{-7}
$\beta_0, \beta_1, \beta_2, b$	Paramètres du moteur	189.63, 6.0612, 0.0122, 280.19

P	Coefficients proportionnel et	22.8257
I	intégral de correcteur PI pour les quatre moteurs	95.0159

II.4.1. Synthèse de contrôleur floue

Nous utilisons d'abord l'outil "fuzzy" de Matlab pour créer un ajusteur de flou, puis nous avons dessiné le graphique sur Simulink. Par conséquent, nous avons intégré le système dynamique on met le vol dans le bloc système, à l'entrée du système, on a sortie régulateur floue, il y a douze variables représentées par l'état à la sortie, regardons l'angle et la position. Nous injectons l'erreur statique et la dérivée de l'erreur dans l'entrée de régulateurs.

Afin de générer des instructions, nous utilisons un générateur de signaux aléatoires au besoin ou générateur d'échelon (step). Ces signaux sont utilisés pour calculer les erreurs statiques. (**Figure II.6**).

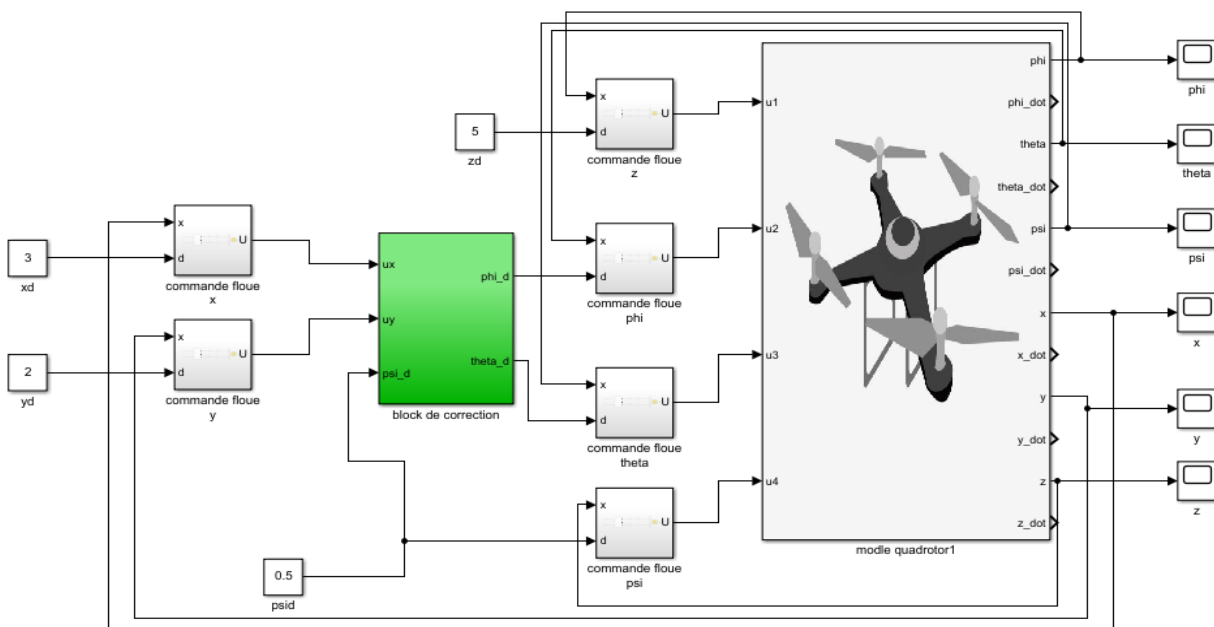


Figure II.6 : Schéma de la commande floue du quadrotor.

En exploitant les informations précédentes, nous pouvons dès à présent concevoir un régulateur flou de type Mamdani.

Nous avons opté pour trois fonctions d'appartenance triangulaires par variable d'entrée. Chaque régulateur reçoit en entrée l'erreur et le dérivé de l'erreur (**figure II.7**) et grâce à neuf règles d'inférences nous obtenons (**figure II.8**), en appliquant une inférence de type max-min, une sortie du régulateur à laquelle nous appliquons une défuzzification utilisant la méthode de centre de gravité.

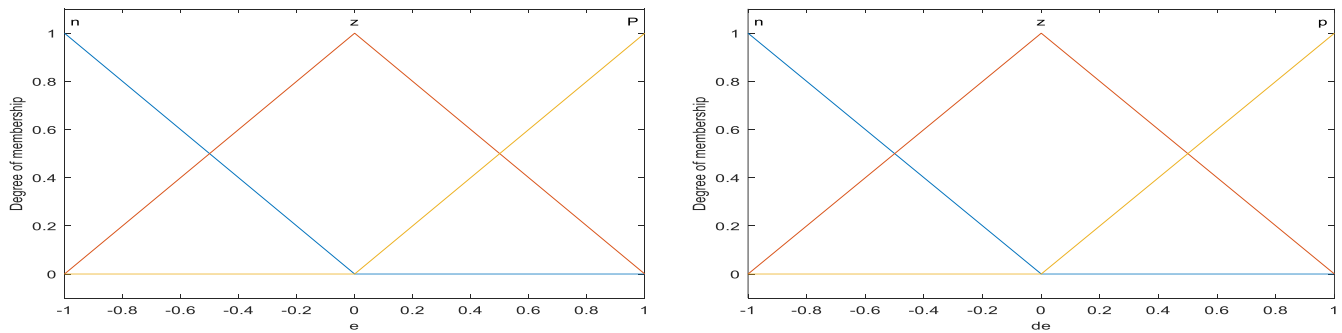


Figure II.7 : Les fonctions d'appartenance de l'erreur et le dérivé de l'erreur.

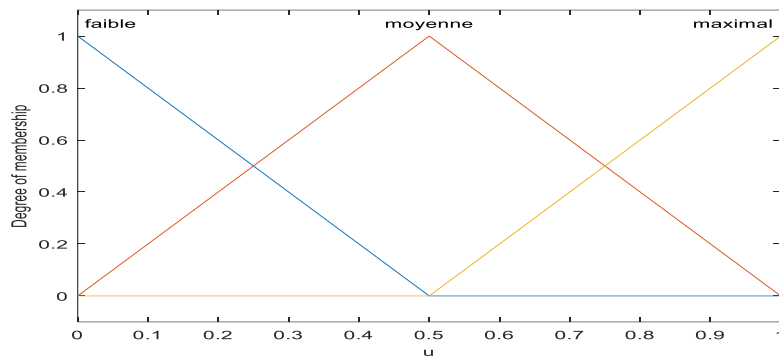


Figure II.8 : Les fonctions d'appartenance de la commande.

Base des règles :

- R1 : SI e est N ET de est N Alors u est faible
- R2 : SI e est N ET de est Z Alors u est faible
- R3 : SI e est N ET de est P Alors u est moyenne
- R4 : SI e est Z ET de est N Alors u est faible
- R5 : SI e est Z ET de est Z Alors u est moyenne

- R6 : SI e est Z ET de est P Alors u est maximal
- R7 : SI e est P ET de est N Alors u est moyenne
- R8 : SI e est P ET de est Z Alors u est maximal
- R9 : SI e est P ET de est P Alors u est maximal

La figure II.9 représente la structure du contrôleur flou sous matlab

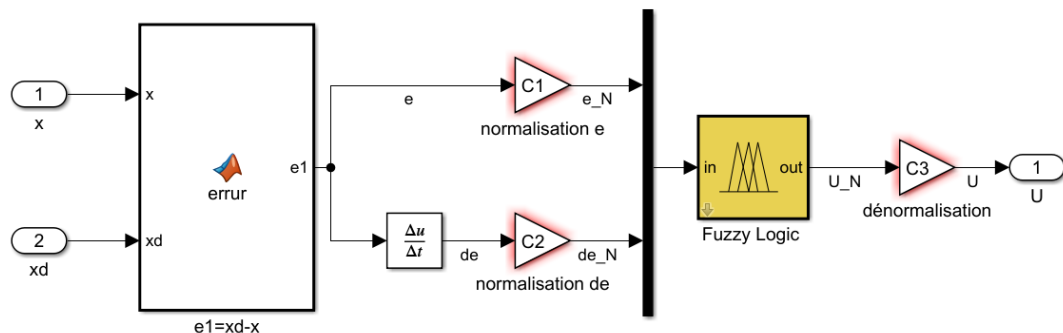


Figure II.9 : Block de la commande floue.

II.4.2 Résultats de simulation

- Sous système d'orientation

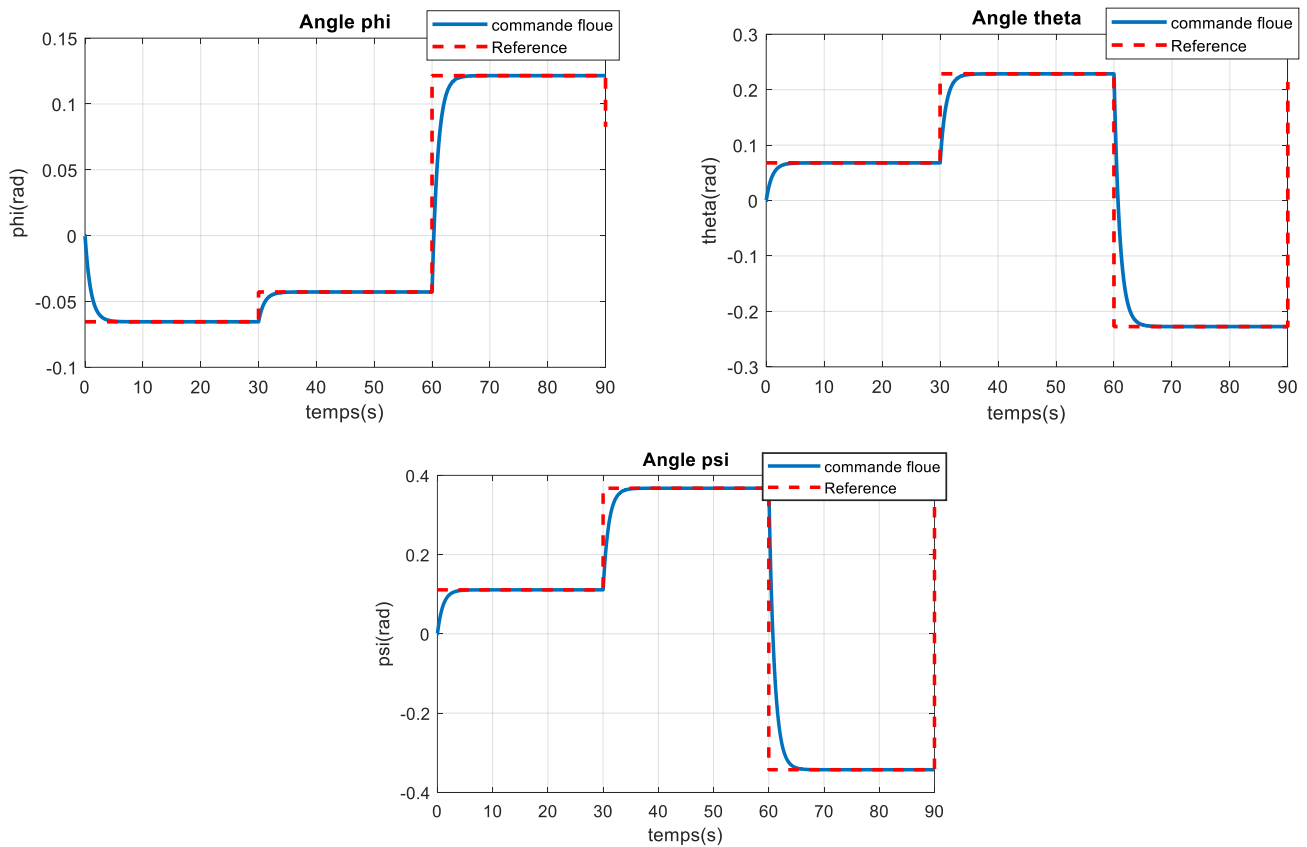


Figure II.10 : stabilisation d'attitude du quadrotor par le contrôleur floue.

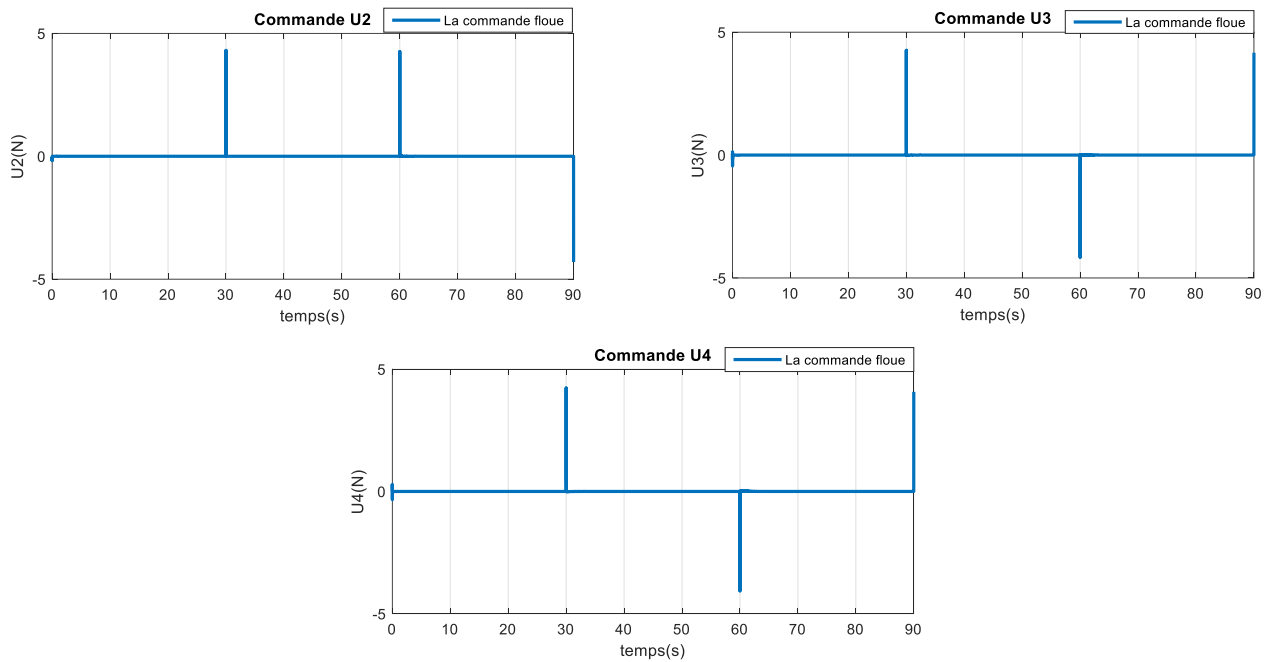


Figure II.11 : les signaux des commandes floue.

- Le système complet (6DOF)

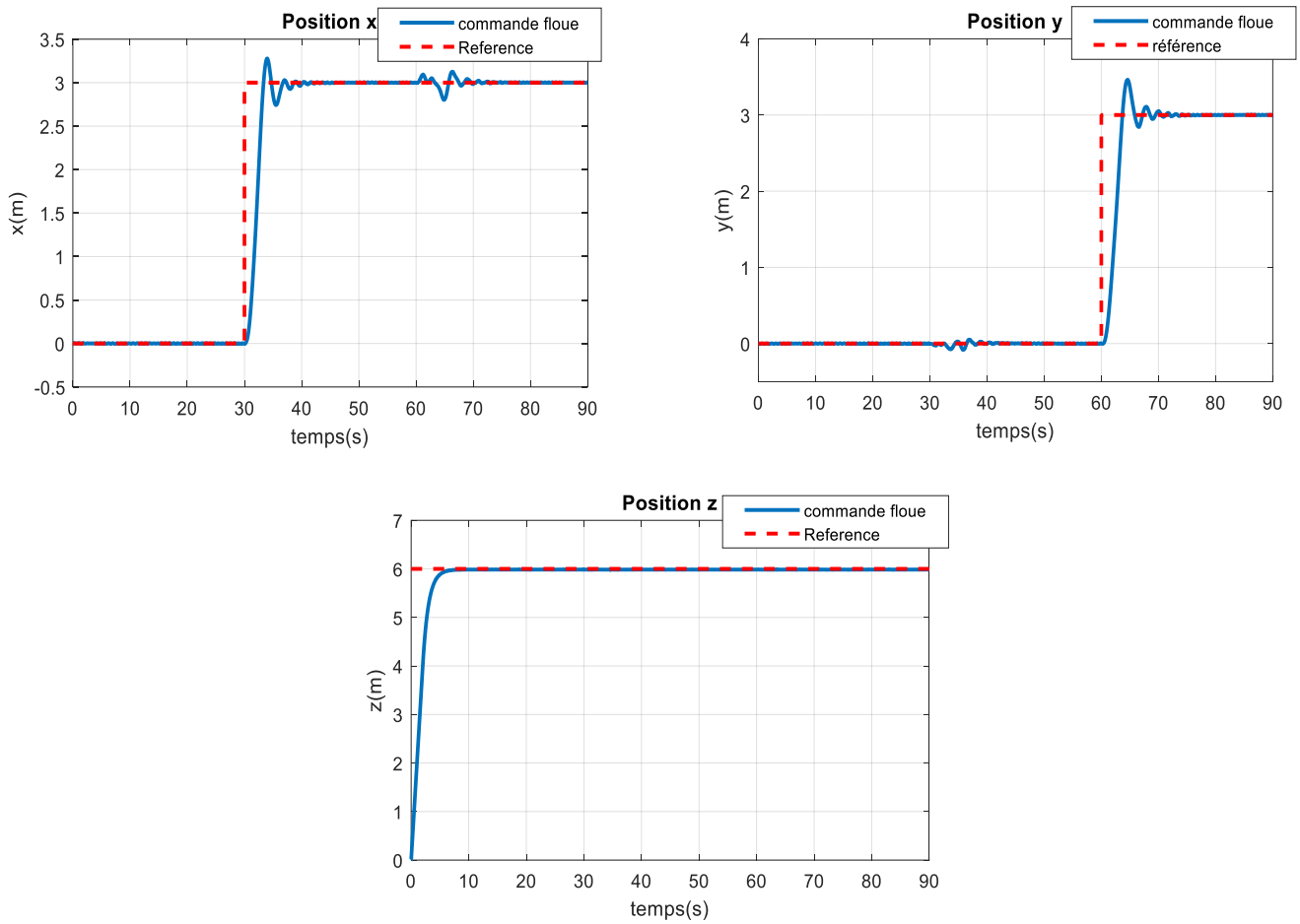


Figure II.12 : Mouvement en X, Y, et Z par la commande floue

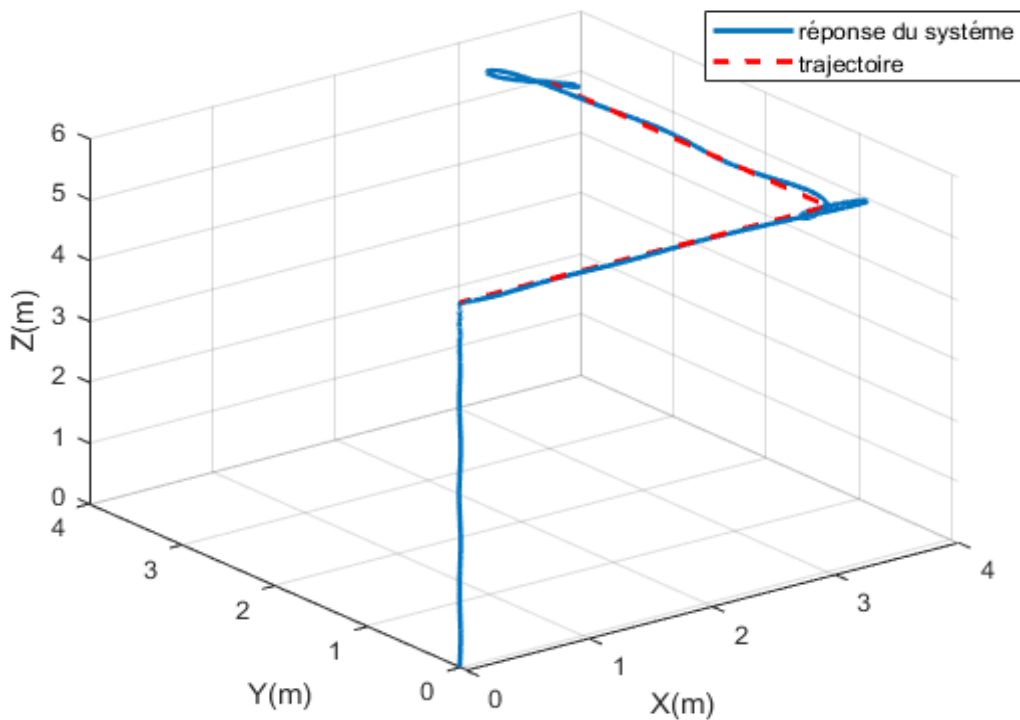


Figure II.13 : Trajectoire globale du quadrotor en 3D.

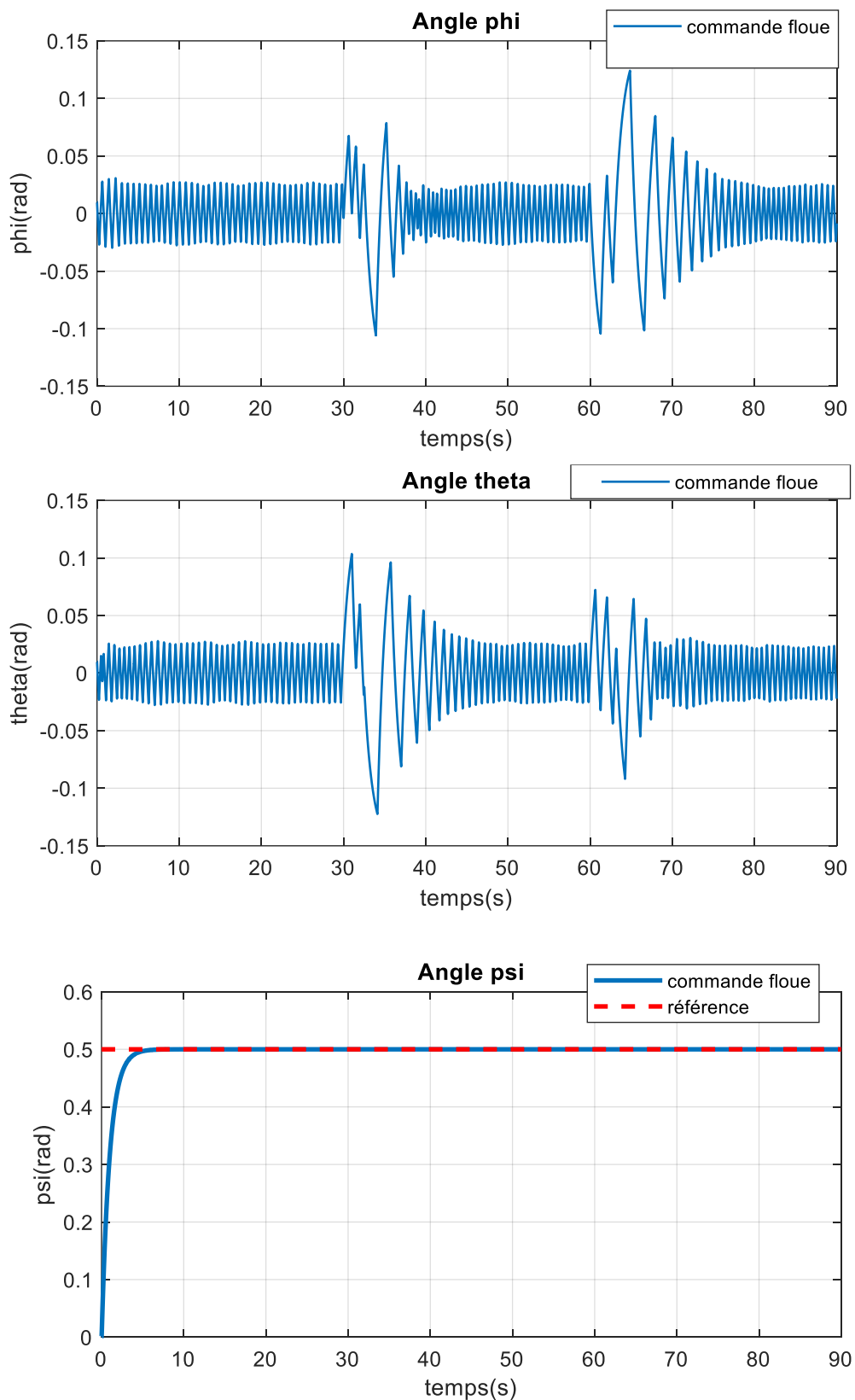


Figure II.14 : variation des angles roulis, tangant, lacet par la commande floue.

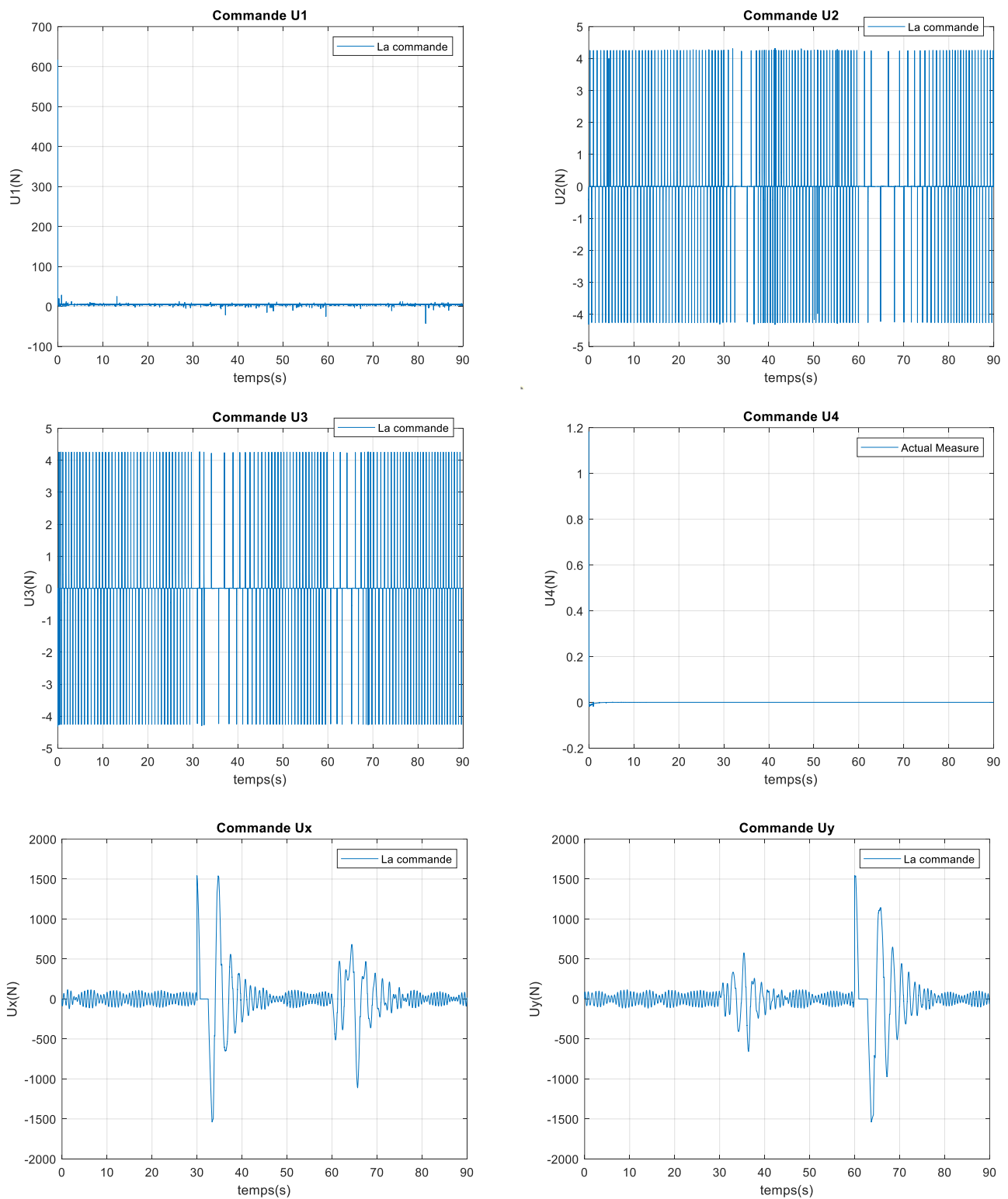


Figure II.15 : les commandes du contrôleur floue.

Les résultats du contrôle du système complet du quadrotor à l'aide de la logique floue sont montrés, dans les figures (II.10 - II.15), nous avons remarqué à partir de ces résultats que le contrôle proposé, généralement, fonctionne bien pour le système complet, avec un bon temps de stabilisation, environ 5 sec,

et un très bon suivi de référence. Toutes les erreurs vont rapidement à zéro, de plus, il n'y a pas de dépassement ou de sous-dépassement, et a une réponse satisfaisante au test de robustesse. Cependant, les résultats de position (mouvements de longitude et de latitude) montrent le problème des oscillations de fréquence, qui se répercutent négativement sur la trajectoire de roulis et de tangage. Après ces observations, nous concluons : Le contrôle à logique floue est une technique robuste, parfaitement adaptée au contrôle de l'ensemble du système du quadrotor, avec un bon temps de réponse et une consommation d'énergie efficace.

II.5. Commande PID adaptatif floue pour la stabilisation d'attitude du quadrotor

II.5.1. Principe de Régulateur PID

La régulation par contrôleur PID exploite trois actions principales, l'action proportionnelle, l'action intégrale et l'action dérivée. Ces actions ont chacune leurs avantages et inconvénients et leur action combinée permet d'avoir une certaine compensation des inconvénients tout en gardant les avantages et ainsi aboutir à des performances intéressantes [33] [34].

✓ Action proportionnelle

L'effet de l'action proportionnelle consiste à amplifier l'erreur d'un gain constant afin que le système réagisse plus rapidement aux changements de consigne. Cette action proportionnelle est représentée comme suit :

$$C(t) = K_p \cdot \varepsilon(t) \quad (\text{II.10})$$

Plus la valeur de K_p est grande plus la réponse est rapide mais au détriment d'une détérioration de la stabilité du système allant jusqu'à l'instabilité pour de grandes valeurs.

✓ Action intégrale

L'action intégrale a pour but de réduire voire d'éliminer l'erreur statique en régime permanent pour réaliser cela le régulateur intègre l'erreur par rapport au temps et multiplie le résultat par une constante K_i comme suit :

$$C(t) = K_i \cdot \int_0^t \varepsilon(t) dt \quad (\text{II.11})$$

Plus la valeur de K_i est grande plus l'erreur statique sera vite corrigée mais nous perdons un peu en stabilité et il y a un risque de dépassement qui subvient.

✓ Action dérivée

Pour obtenir une action dérivée nous multiplions la dérivée de l'erreur par un coefficient K_d cette action permet d'éliminer le dépassement de la réponse et d'améliorer la stabilité du système. Sa relation est donnée comme suit :

$$C(t) = K_d \cdot \frac{d\varepsilon(t)}{dt} \tag{II.12}$$

Plus la valeur de K_d est grande plus le dépassement est atténué mais si elle est trop grande le système est ralenti jusqu'à risquer de devenir instable pour de très grandes valeurs.

✓ **Action PID**

Le régulateur PID combine les trois actions vues précédemment et permet ainsi d'avoir de bonnes performances aussi bien en vitesse en stabilité qu'en précision. L'expression d'un régulateur PID est donnée comme suit :

$$C(t) = K_p \cdot \varepsilon(t) + K_i \cdot \int_0^t \varepsilon(t)dt + K_d \cdot \frac{d\varepsilon(t)}{dt} \tag{II.13}$$

Ainsi ce régulateur permet d'atteindre les objectifs fixés en termes de vitesse de stabilité et de précision et ce en trouvant la configuration optimale des valeurs des différents gains K_p , K_i et K_d .

II.5.2. Synthèse de contrôleur PID adaptatif floue

On applique la commande floue PID sur le sous-système (**Figure II.16**), nous ne prenons pas en compte les frottements dus aux moments aérodynamique ni les couples dus aux effets gyroscopiques, on considère que les angles très petites, pour découpler le système et rendre modèle d'état linéarisé :

$$\begin{cases} \dot{x}_1 = x_2 \\ \dot{x}_2 = b_1 U_2 \\ \dot{x}_3 = x_4 \\ \dot{x}_4 = b_2 U_3 \\ \dot{x}_5 = x_6 \\ \dot{x}_6 = b_3 U_4 \\ \dot{x}_7 = x_8 \\ \dot{x}_8 = \frac{U_1}{m} x_3 \\ \dot{x}_9 = x_{10} \\ \dot{x}_{10} = -\frac{U_1}{m} x_1 \\ \dot{x}_{11} = x_{12} \\ \dot{x}_{12} = \frac{1}{m} U_1 - g \end{cases} \tag{II.14}$$

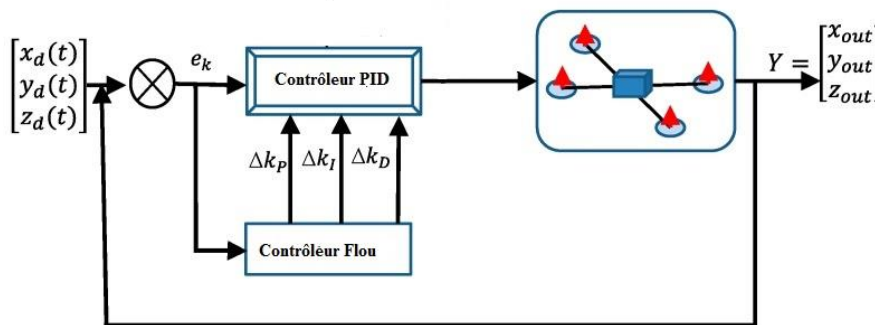


Figure II.16 : Architecture système du PID flou pour le quadrotor.

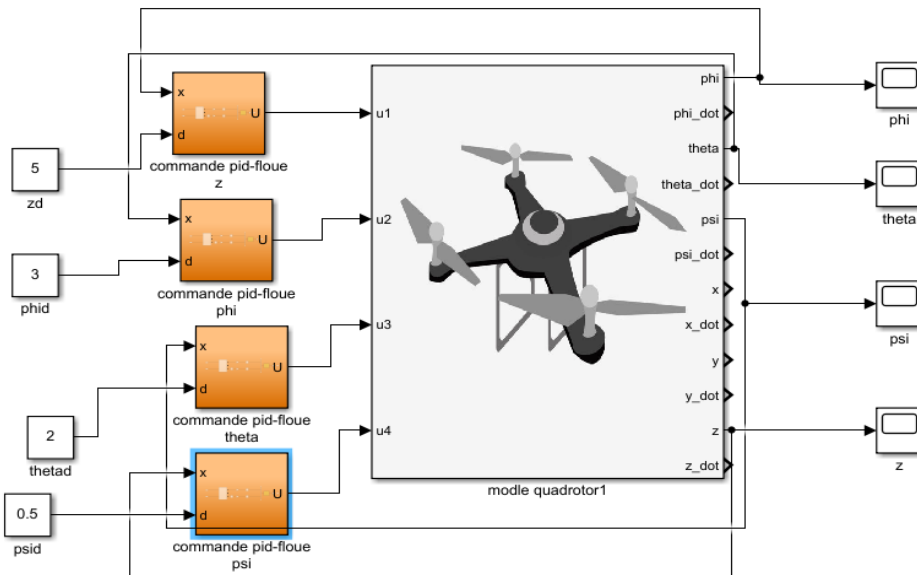


Figure II.17 : Schéma de la commande PID floue du sous-système d'orientation.

Nous avons utilisé un système expert floue pour ajuster les gains de la commande PID (**figure II.18**), on a opté pour sept fonctions d'appartenance triangulaires par variable d'entrée. Chaque régulateur reçoit en entrée l'erreur et le dérivé de l'erreur (**figure II.19**) et grâce à quarante-neuf règles d'inférences nous obtenons (**figure II.20**), en appliquant une inférence de type max-min, une sortie du régulateur à laquelle nous appliquons une défuzzification utilisant La méthode de centre de gravité.

Dans la sortie on a trois adaptations des gains du régulateur PID :

- Adaptation sur le gain K_p .
- Adaptation sur le gain K_i .
- Adaptation sur le gain K_d .
-

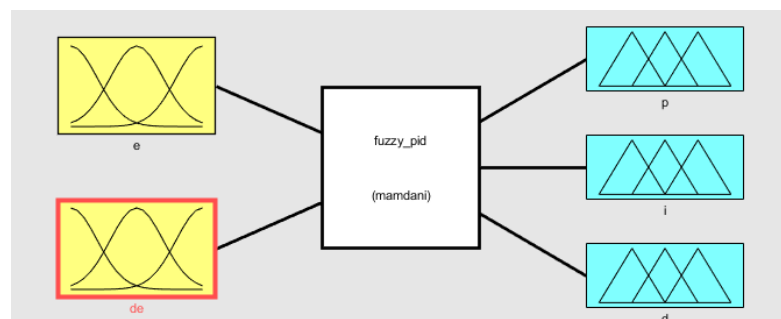


Figure II.18 : L'adaptation des gains par le régulateur floue.

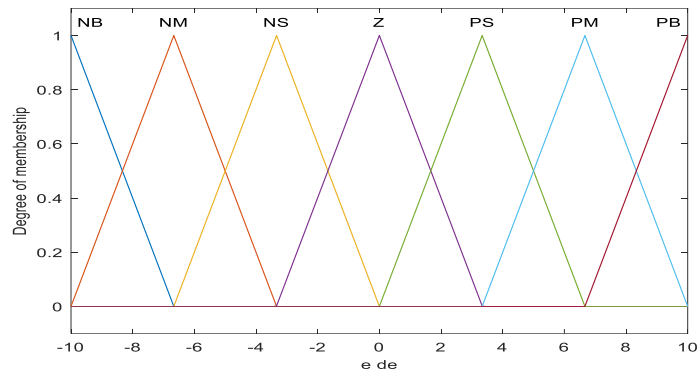


Figure II.19 : Les fonctions d'appartenance de l'erreur et le dérivé de l'erreur.

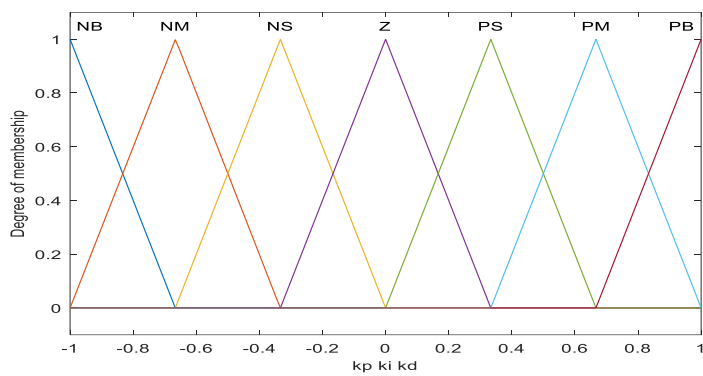


Figure II.20 : Les fonctions d'appartenance de l'adaptation des gains.

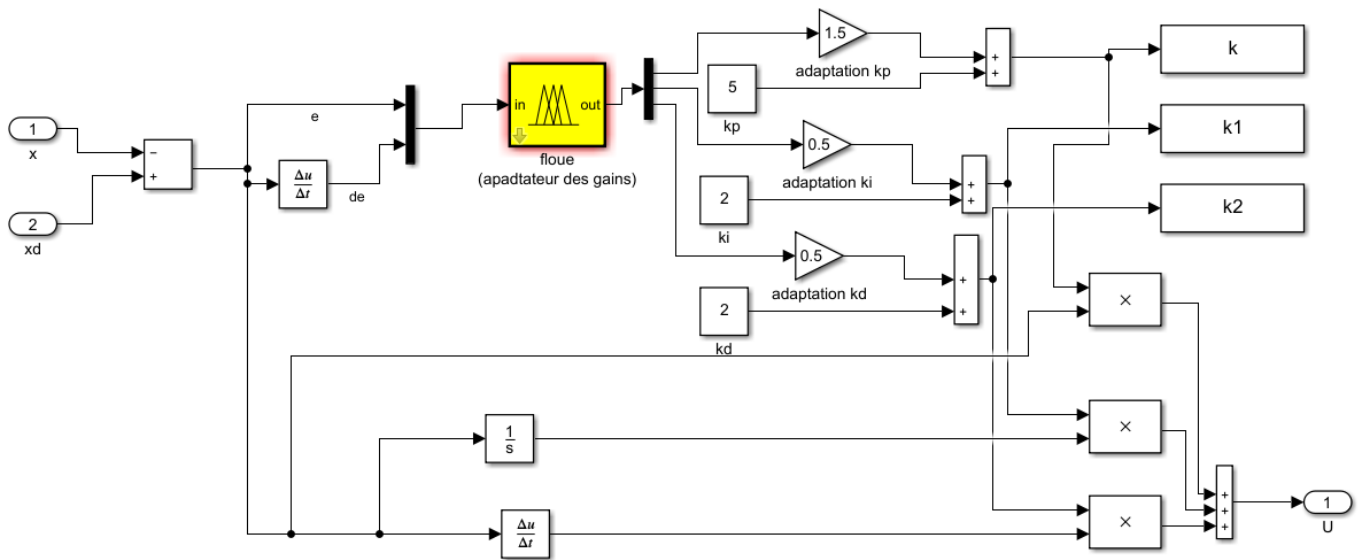


Figure II.21 : Block de la commande PID floue.

II.5.3. Résultats de simulation

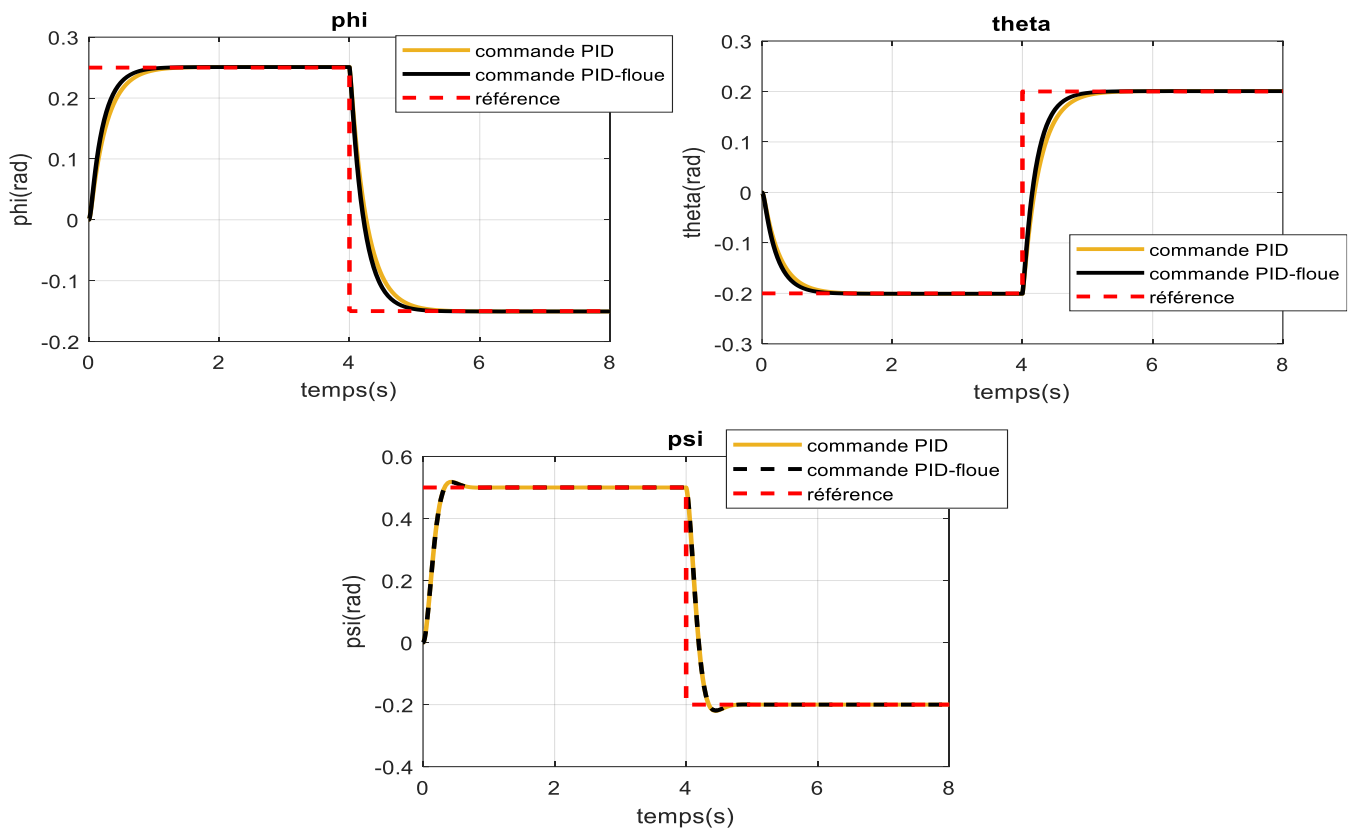


Figure II.22 : stabilisation d'attitude par le contrôleur PID-floue.

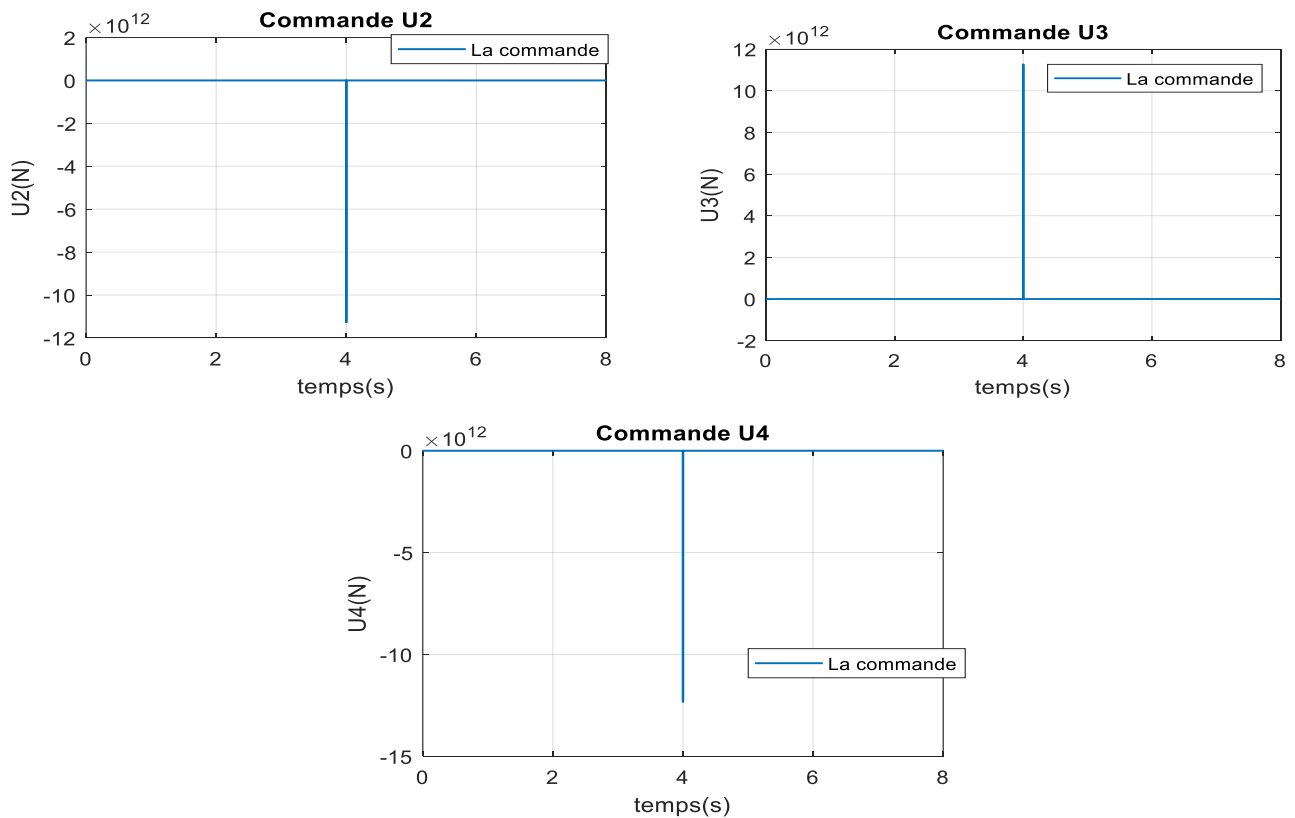


Figure II.23 : la commande du contrôleur PID-floue.

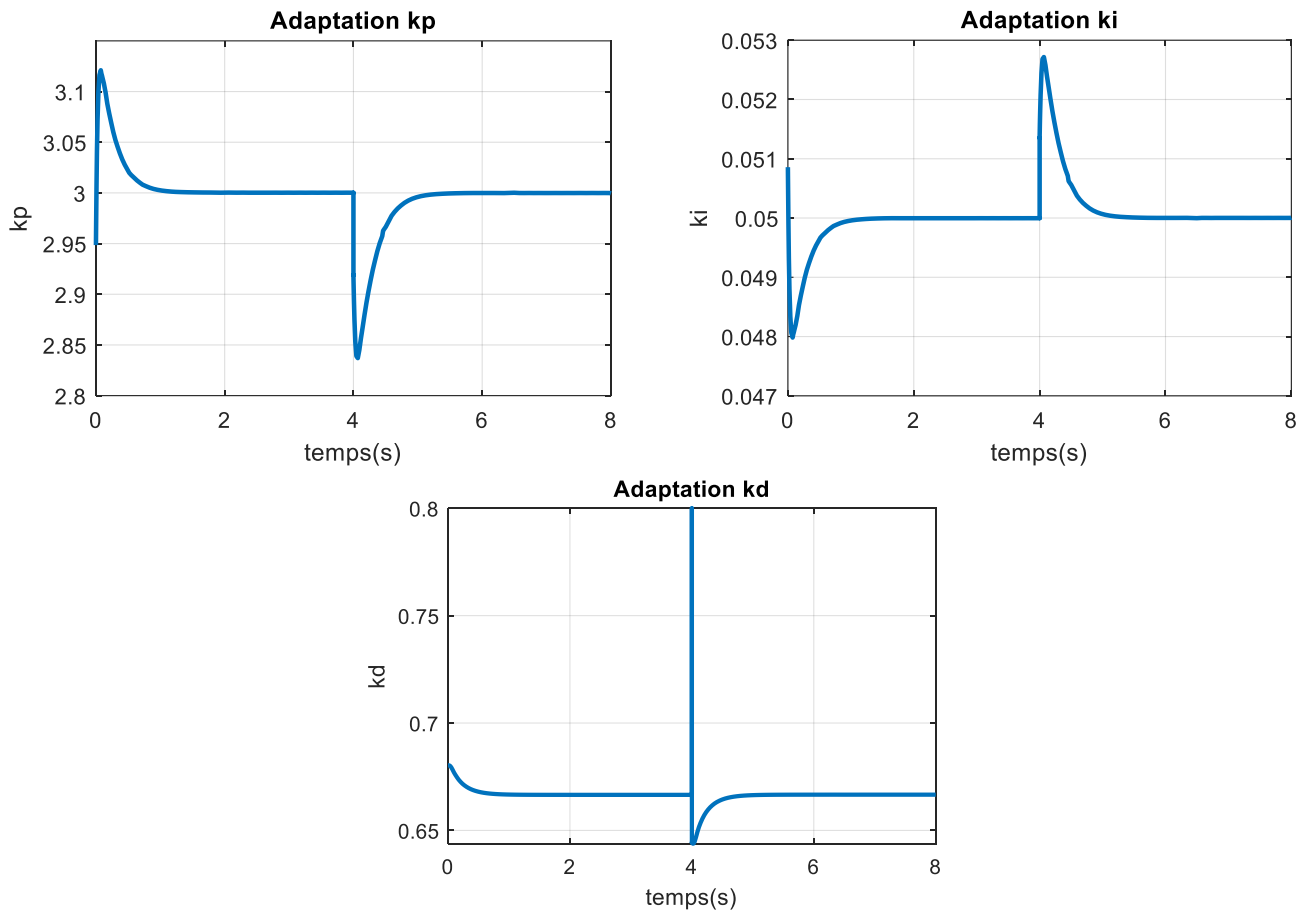


Figure II.24 : l'adaptation des gains de contrôleur PID-floue.

Les résultats obtenus montrent que l'adaptation des gains de la commande PID par un système expert flou permet de garantir la stabilité et de réaliser une poursuite des trajectoires désirées avec une précision meilleure que celle obtenue dans le cas de la commande PID classique.

- Le bon choix des fonctions d'appartenances du système flou assure de très bons résultats de poursuite.
- Pour le régulateur PID adaptatif flou, les positions X, Y, Z et l'angle Ψ convergent, sans oscillation, à leurs valeurs désirées.

II.6. Commande par mode glissant-floue

II.6.1 la commande par mode de glissant

L'idée de base de la commande par mode glissant est premièrement d'attirer les états du système vers une surface de glissement convenablement choisi, puis de concevoir une loi de commande stabilisante qui pourra maintenir le système sur cette surface. En résumé, une commande par mode glissant est divisée en trois parties (voir Figure II.25) :

- Choix des surfaces de glissement.
- Définition des conditions d'existence et de convergence du régime glissant.
- Détermination de la loi de commande [35].

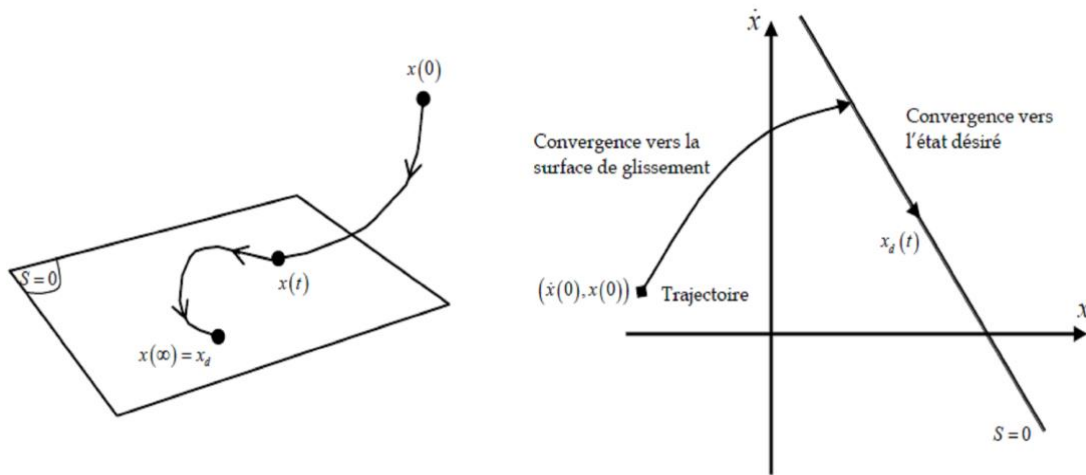


Figure II.25 Convergence du système glissant.

Les deux composantes de la commande sont [36]:

$$U = U_{\text{equivalente}} + U_{\text{attractive}}$$

$U_{\text{equivalente}}$: La commande équivalente ou nominale est déterminée par le modèle du système, dans ce cas il s'agit d'un modèle en immersion linéaire ou non linéaire. Cette partie est conçue avec la méthode de la commande équivalente, dont le principe est basé sur la détermination du comportement du système lorsqu'il est sur la surface de glissement S soit ($S = 0$).

$U_{\text{attractive}}$: Le glissement est utile pour compenser les incertitudes du modèle. Il est constitué de la fonction signe "sign" de la surface de glissement, S multipliée par une constante K . La surface de glissement est définie dans l'espace d'état des erreurs afin de garantir la convergence des états du système.

II.6.2 Phénomène de broutement (Chattering)

Dans la pratique, un régime glissant idéal n'existe pas car la fréquence de commutation des organes de commande a une limite finie. Autrement dit, il n'existe aucun organe de commutation pouvant commuter à une fréquence infinie (en effet cet organe devrait délivrer une énergie infinie). Le caractère discontinu de la commande engendre un comportement dynamique particulier autour d'une couche limite de la surface de glissement qui est communément appelé chattering ou phénomène de broutement ou réticence figure (II.26). Cette oscillation au voisinage de la surface est due à l'imperfection des éléments de commutation ou des limites technologiques et physiques, telles que les retards au niveau des

commutations ou des comportements avec hystérésis, qui peuvent exciter les dynamiques négligées (non modélisées) en haute fréquence.

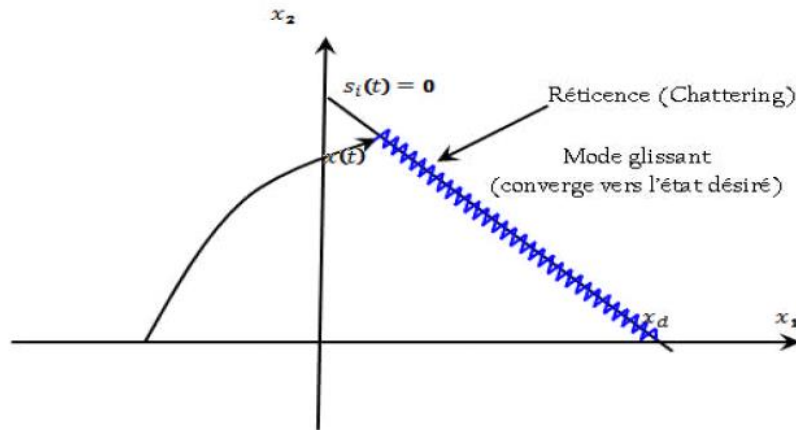


Figure II.26 Phénomène de broutement (Chattering)

II.6.3 Structure du contrôleur flou glissant

Le contrôleur flou-glissant est composé de deux blocs (**figure II.27**). Un bloc pour le calcul de la commande équivalente et la surface de glissement. Le deuxième bloc est un régulateur flou, dont l'entrée est la surface et la sortie est la commande attractante.

La sortie du contrôleur flou-glissant est la somme de la commande équivalente et la commande attractante.

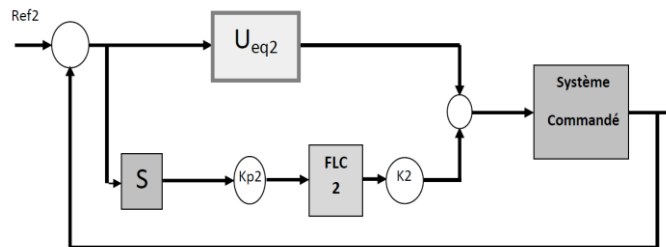


Figure II.27: Structure du régulateur flou-glissant.

II.6.4 Synthèse de contrôleur flou par mode glissant

Soit l'erreur de poursuite suivante :

$$\begin{cases} e_i = x_i - x_{id} \\ e_{i+1} = \dot{e}_i \end{cases}$$

$$\begin{cases} e_1 = x_1 - x_{1d} \\ e_2 = \dot{e}_1 \end{cases} \quad \text{(II.15)}$$

La surface de glissement est choisie comme suite :

$$S_\phi = e_2 + \lambda_2 e_1 \quad (\text{II.16})$$

$$S_\phi = \dot{e}_1(\theta) + \lambda_2 e_2(\theta)$$

Soit la fonction de Lyapunov suivante :

$$V_{S_\phi} = \frac{1}{2} S_\phi^2 \quad (\text{II.17})$$

$$S_\phi = x_2 - \dot{x}_{1d} + \lambda_2 e_1 \quad (\text{II.18})$$

$$\dot{S}_\phi = \dot{x}_2 - \ddot{x}_{1d} + \lambda_2 \dot{e}_1 \quad (\text{II.19})$$

Si $\dot{S}_\phi S_\phi < 0$, on peut dire que la condition nécessaire de glissement est vérifiée et la stabilité au sens Lyapunov est garantie.

Soit :

$$\dot{S}_\phi = -K_2 \text{sign}(S_\phi) \quad (\text{II.20})$$

$$\dot{S}_\phi = \dot{x}_2 - \ddot{x}_{1d} + \lambda_2 \dot{e}_1$$

$$\dot{S}_\phi = a_1 x_4 x_6 + a_3 x_4 q + a_2 x_2^2 b_1 U_2 - \ddot{x}_{1d} + \lambda_2 \dot{e}_1 = -K_2 \text{sign}(S_\phi) \quad (\text{II.21})$$

$$U_2 = \frac{1}{b_1} \{-a_1 x_4 x_6 - a_3 x_4 q - a_2 x_2^2 + \ddot{x}_{1d} - \lambda_2 e_2 - K_2 \text{sign}(S_\phi)\} \quad (\text{II.22})$$

$$\begin{cases} U_{2d} = -\frac{K_2}{b_1} \text{sign}(S_\phi) \\ U_{2eq} = \frac{1}{b_1} \{-a_1 x_4 x_6 - a_3 x_4 q - a_2 x_2^2 + \ddot{x}_{1d} - \lambda_2 e_2\} \end{cases} \quad (\text{II.23})$$

Les mêmes étapes sont reprises à fin d'extraire :

$$U_1 = \frac{1}{n} \{-a_{11} x_{12} + \ddot{x}_{11d} - \lambda_1 e_{12} - K_1 \text{sign}(S_z)\}$$

$$U_2 = \frac{1}{b_1} \{-a_1 x_4 x_6 - a_3 x_4 q - a_2 x_2^2 + \ddot{x}_{1d} - \lambda_2 e_2 - K_2 \text{sign}(S_\phi)\}$$

$$U_3 = \frac{1}{b_2} \{-a_4 x_2 x_6 - a_6 x_2 q - a_5 x_4^2 + \ddot{x}_{3d} - \lambda_3 e_4 - K_3 \text{sign}(S_\theta)\}$$

$$U_4 = \frac{1}{b_3} \{-a_7 x_2 x_4 - a_8 x_6^2 + \ddot{x}_{5d} - \lambda_4 e_6 - K_4 \text{sign}(S_\psi)\}$$

$$U_x = \frac{m}{U_1} \{-a_9 x_8 + \ddot{x}_{7d} - \lambda_5 e_8 - K_5 \text{sign}(S_x)\}$$

$$U_y = \frac{m}{U_1} \{-a_{10} x_{10} + \ddot{x}_{9d} - \lambda_6 e_8 - K_6 \text{sign}(S_y)\}$$

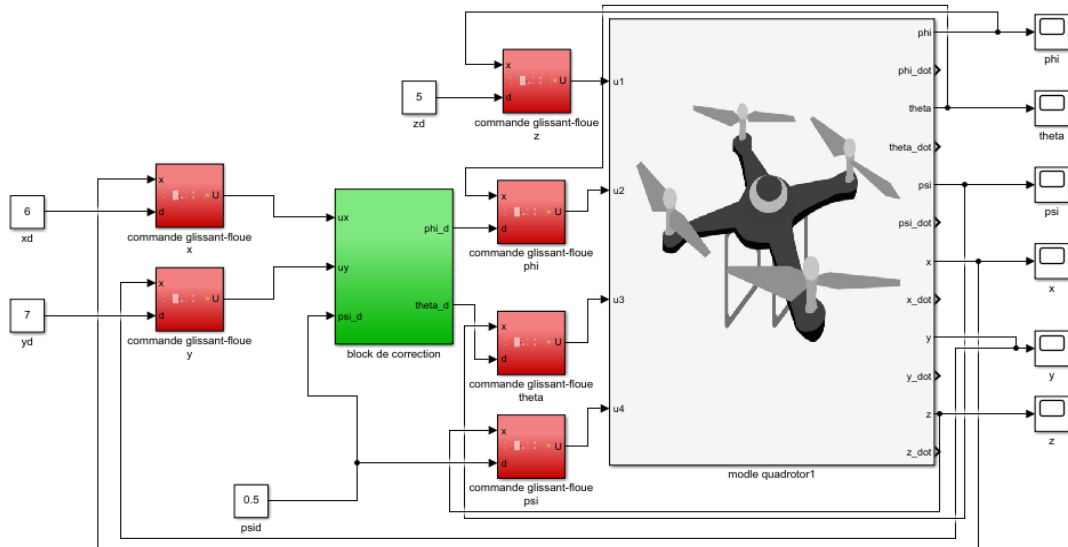


Figure II.28 : Schéma de la commande par mode glissant floue.

✓ **Commande d'attractivité**

L'attractivité est réalisée par un contrôleur flou du type Takagi-Sugeno se ayant une seule entrée (surface) qui a cinq fonctions d'appartenance, quant à la sortie (la commande d'attractivité).

Les fonctions d'appartenance des deux variables s et Δu , sont illustrées par la **figure (II.29)**.

On définit des fonctions d'appartenance de forme triangulaires pour la surface de glissement s et des singletons pour la commande Δu .

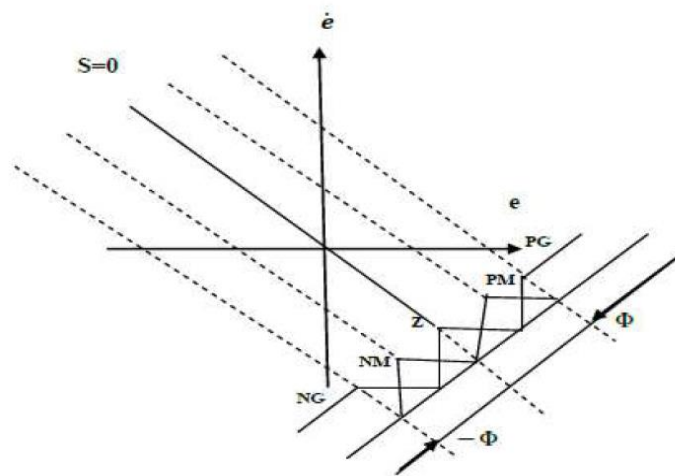


Figure II.29 : Partition floue de l'espace autour de la surface de glissement.

Elle en possède cinq règles sont les suivant :

R1 : SI s est NG Alors Δu est PG

R2 : SI s est NM Alors Δu est PM

R3 : SI s est Z Alors Δu est Z

R4 : SI s est PM Alors Δu est NM

R5 : SI s est PG Alors Δu est NG

Ou :

NG : négatif grand ;

NM : négatif moyen ;

Z : zéro ;

PM : positif moyen ;

PG : positif grand.

Les fonctions d'appartenance sont choisies de la forme :

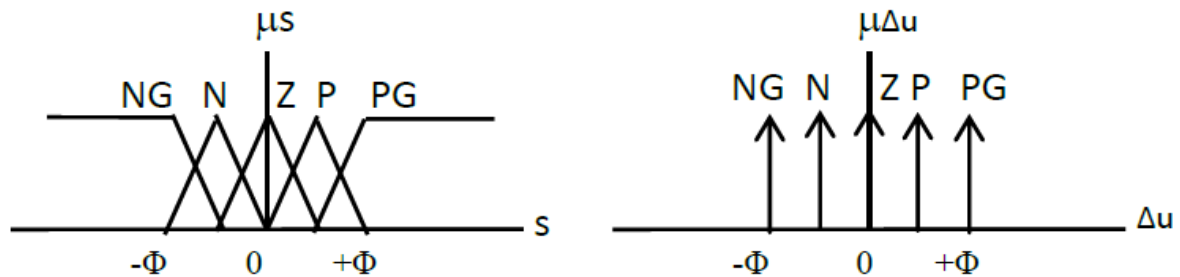


Figure II.30 : Les fonctions d'appartenances de s et de Δu .

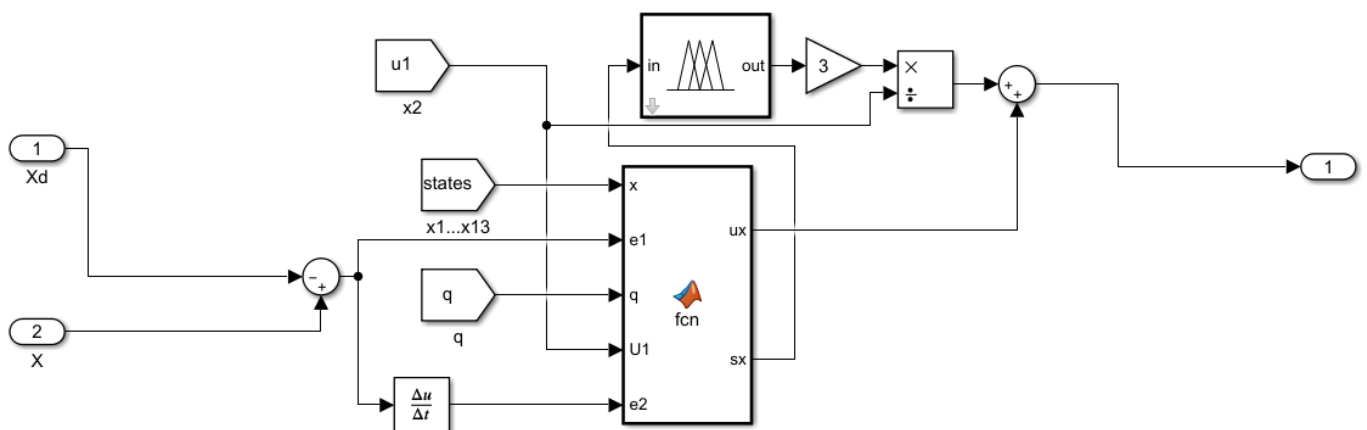


Figure II.31 : Block de la commande par mode glissant floue.

II.6.5 Résultats de simulation

-sous système d'orientation

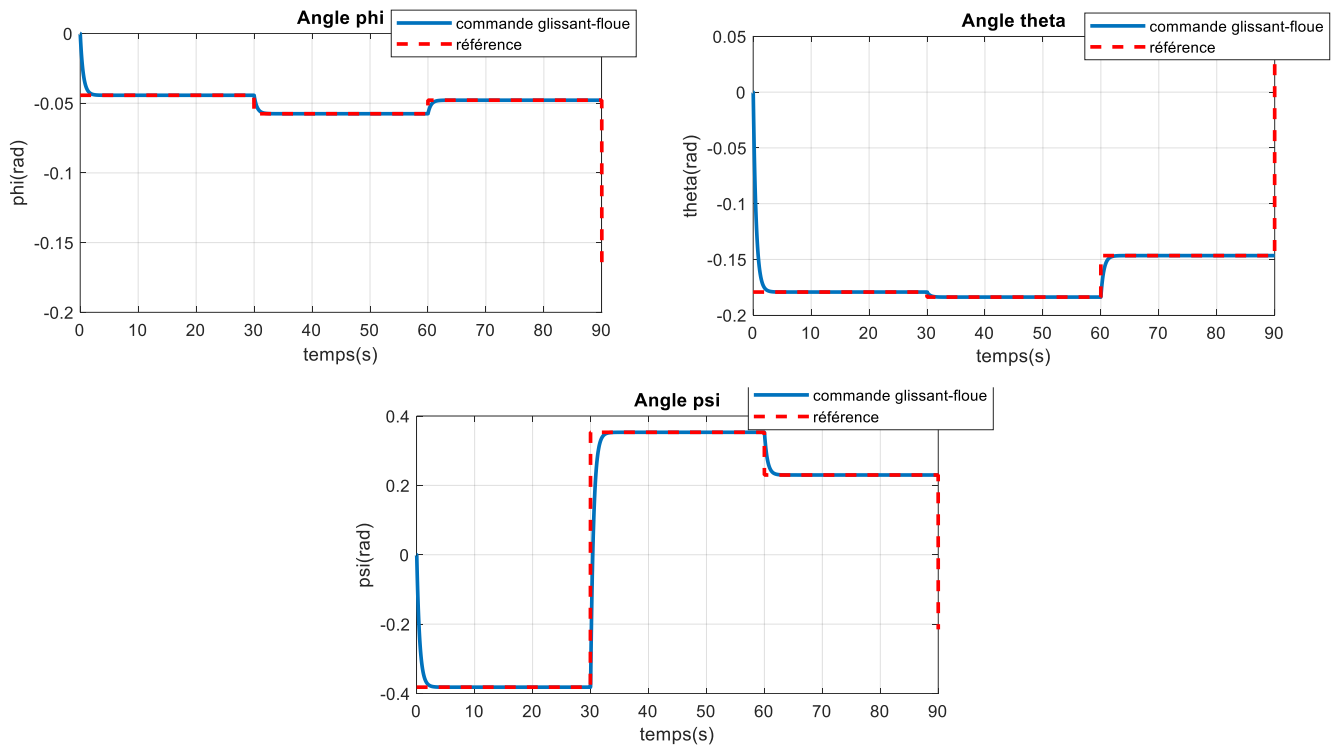


Figure II.32 : stabilisation d'attitude par le contrôleur mode glissant-floue.

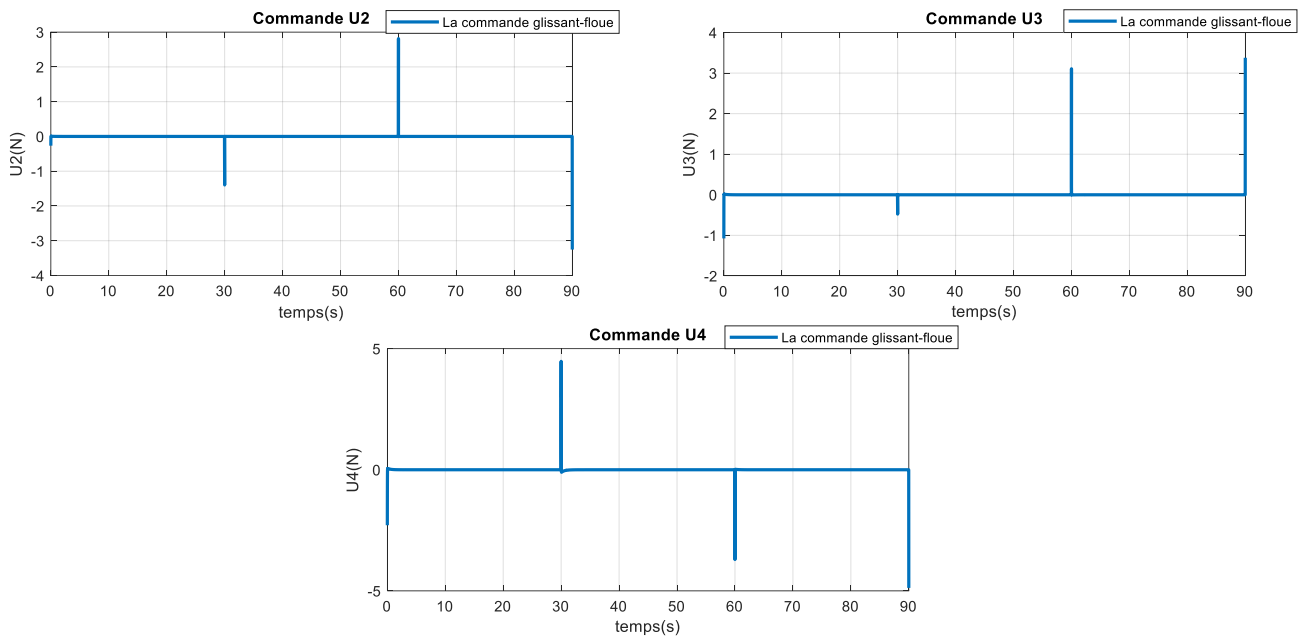


Figure II.33 : les signaux de commandes du contrôleur mode glissant-floue.

-le système complet (6DOF)

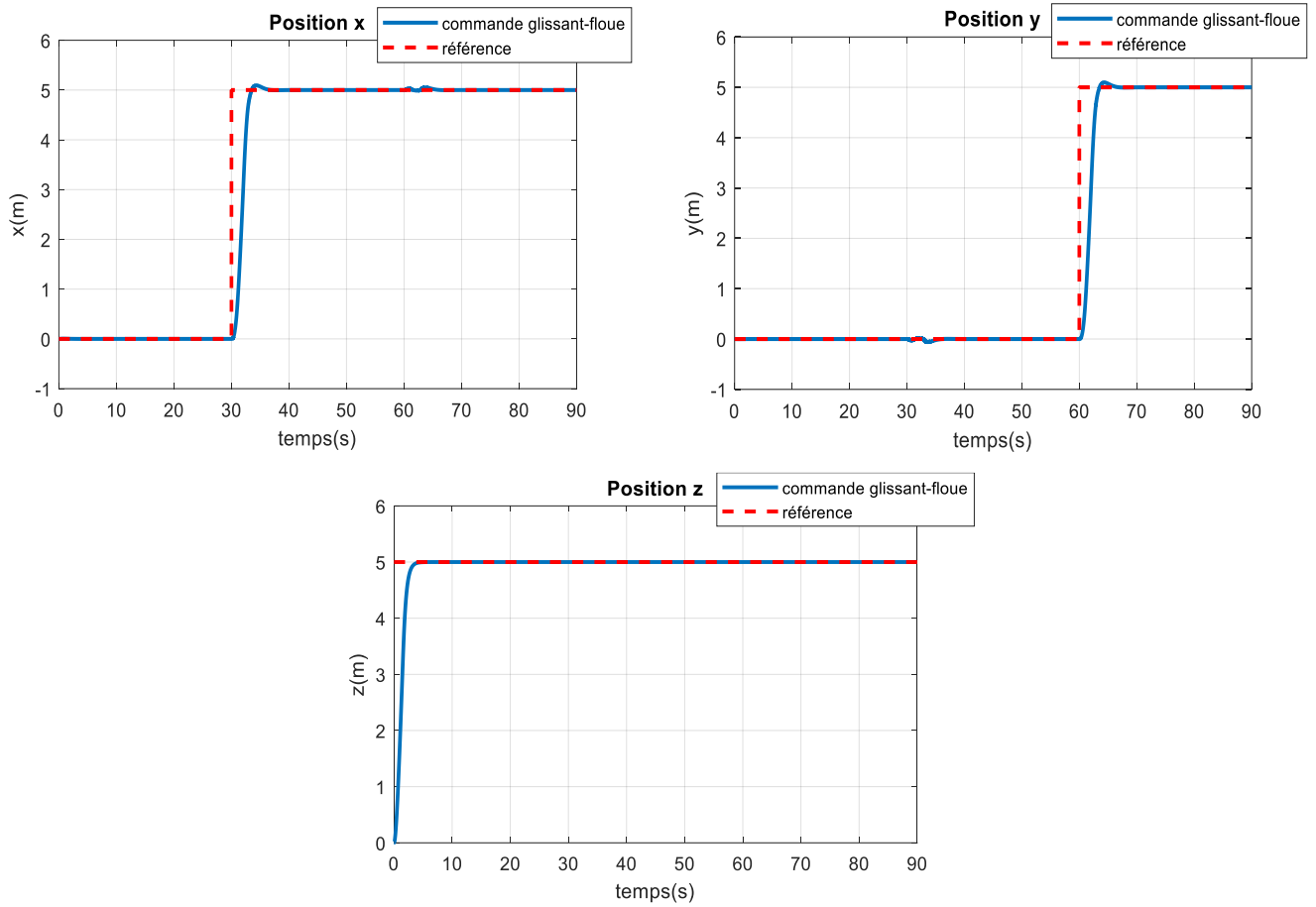


Figure II.34 : réponse du système par le contrôleur mode glissant-floue.

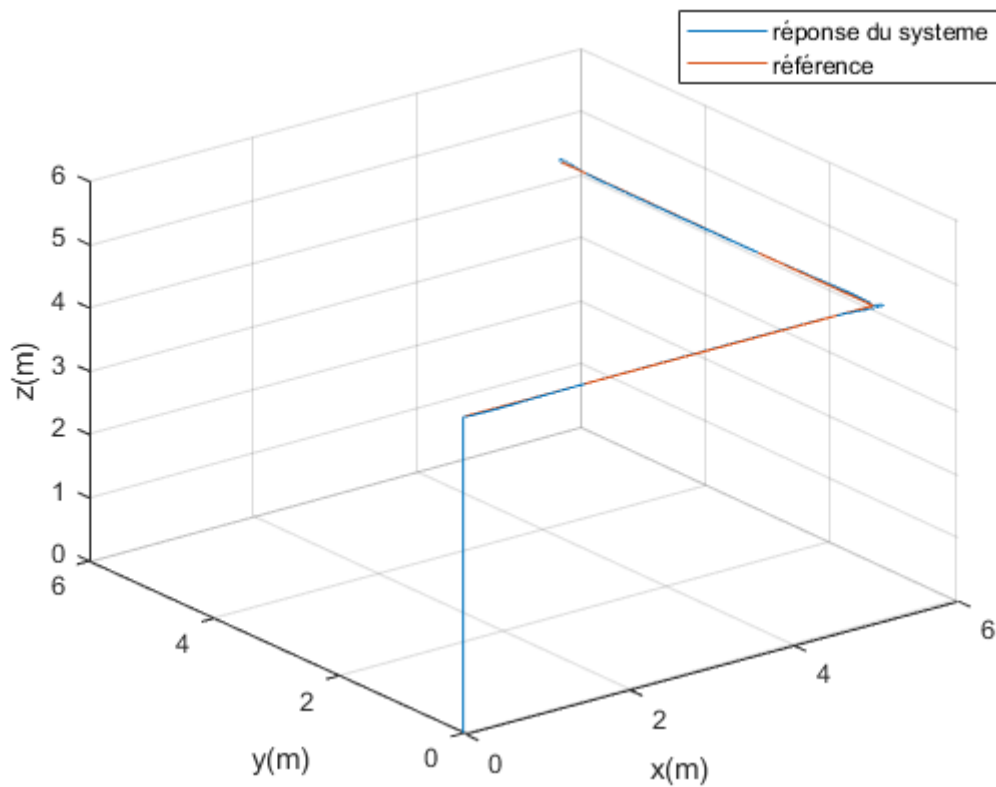


Figure II.35 : trajectoire de poursuite en 3D.

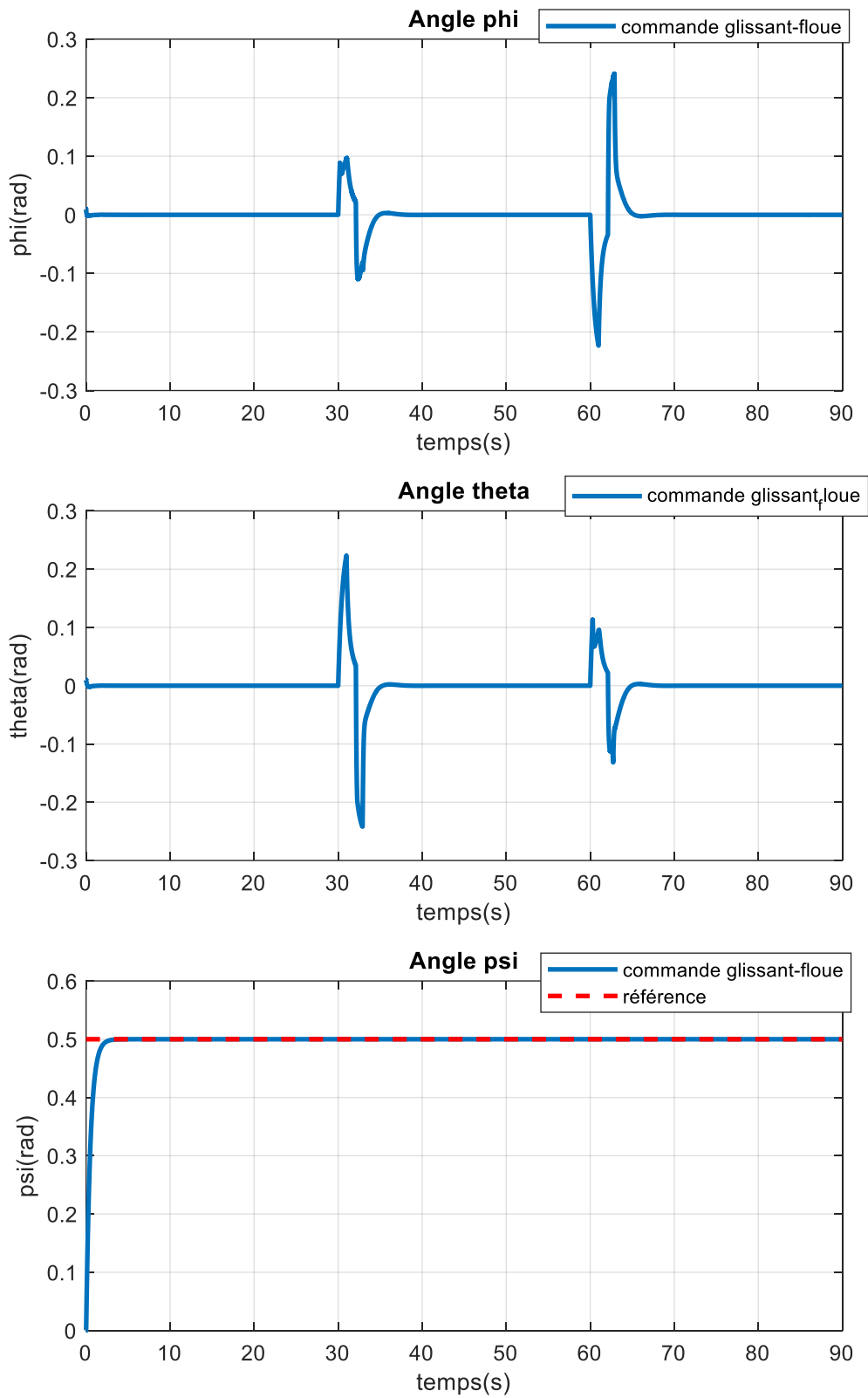


Figure II.36 : variation des angles roulis, tangant, lacet par le contrôleur mode glissant-floue.

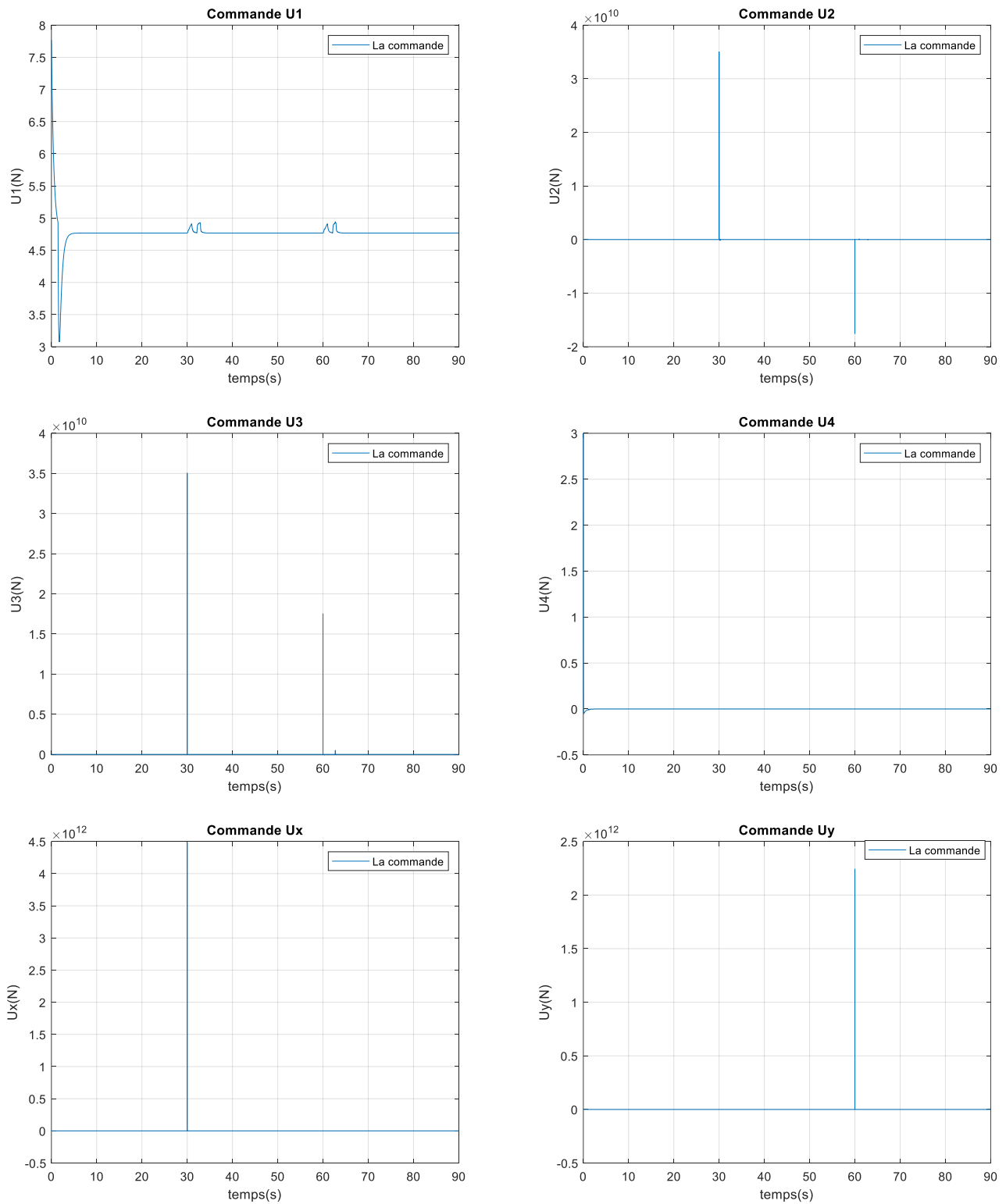


Figure II.37 : les signaux de commande du contrôleur mode glissant-floue.

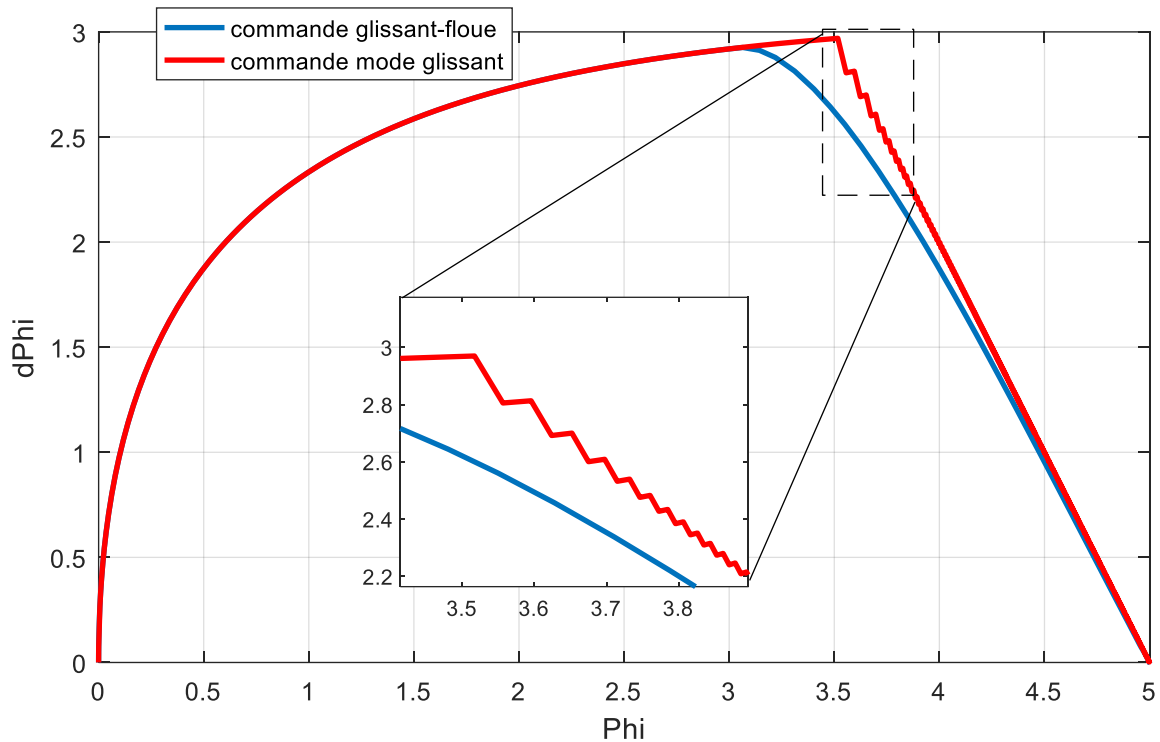


Figure II.38 : phénomène de chattering.

Les différents résultats de la simulation obtenue montrent l'amélioration des performances et la robustesse du contrôleur par mode glissant, avec le choix d'une surface floue dans la structure de la commande.

II.7 Conclusions

Dans ce chapitre, différentes techniques de commande basées sur la logique floue ont été développées (commande floue, PID floue et floue-glissante) et testées aussi bien en simulation. Pour garantir la stabilité en attitude et en altitude du quadrotor, et aussi la robustesse en poursuite de la trajectoire désirée.

La stratégie de la commande par la logique floue exige moins d'informations sur le modèle du système et elle est simple et facile pour la conception. L'adaptation des gains de la commande PID par un système expert flou permet de garantir la stabilité et de réaliser une poursuite des trajectoires désirées avec une précision meilleure que celle obtenue dans le cas de la commande PID classique, aussi. En résolvant le problème du chattering de la commande par mode de glissement par l'hybridation de cette technique par la logique floue dans le choix de la surface de glissement. Les différents résultats de la simulation obtenue montrent les performances et la robustesse des contrôleurs proposés.

CHAPITRE III

CHAPITRE III

COMMANDE PAR LES RÉSEAUX DE NEURONE ARTIFICIEL POUR LA STABILISATION D'ATTITUDE DU QUADROTOR

III.1 Introduction

L'un des défis de l'homme aujourd'hui est de copier la nature et de reproduire des modes de raisonnement et de comportement qui lui sont propre. Les réseaux de neurones, sont nés de cette envie, ils constituent une famille de fonctions non linéaires paramétrées, utilisées dans de nombreux domaines (physique, chimie, biologie, finance, etc.), notamment pour la modélisation de processus et la synthèse de lois de commandes, L'utilisation de réseaux de neurones artificiels (RNA) pour la réalisation des systèmes de commande a connu un essor important au cours des dernières années.

Dans ce chapitre nous avons fait d'abord une brève présentation de quelques notions générales sur les RNA et les modalités de leur utilisation, dans ce domaine, ensuite nous avons appliqué deux technique de commande sur le quadrotor basé sur RNA, la première est une commande RNA directe, la deuxième est une commande RNA-PID. Nous présentons les modèles de réseaux utilisés dans ce mémoire pour garantir la stabilisation d'attitude du quadrotor.

III.2 Généralités

L'origine des réseaux de neurones vient de l'essai de modélisation mathématique du cerveau humain. Les premiers travaux datent de 1943 et sont l'oeuvre de MM. Mac Culloch et Pitts [37]. Ils supposent que l'impulsion nerveuse est le résultat d'un calcul simple effectué par chaque neurone et que la pensée née grâce à l'effet collectif d'un réseau de neurones interconnectés (**figure III.1**). Ils ont connu des débuts prometteurs vers la fin des années 50, mais le manque d'approfondissement de la théorie a gelé ces travaux jusqu'aux années 80.

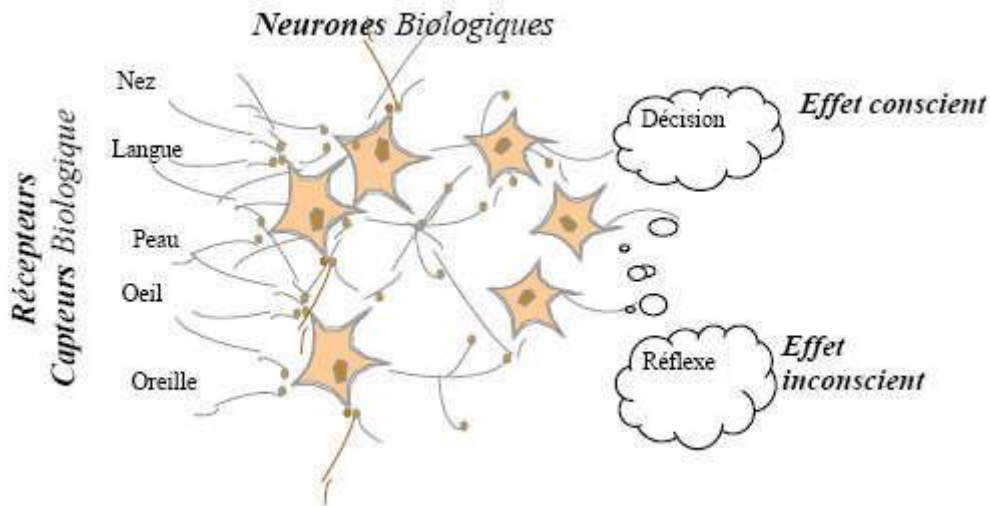


Figure III.1 Structure d'un réseau de neurones biologiques.

Les réseaux de neurones forment une famille de fonctions non linéaires, permettant de construire, par apprentissage, une très large classe de modèles et de contrôleurs. Un réseau de neurones est un système d'opérateurs non linéaires interconnectés, recevant des signaux de l'extérieur par ses entrées, et délivrant des signaux de sortie, qui sont en fait les activités de certains neurones.

III.3 Neurone biologique

Le neurone est une cellule composée d'un corps cellulaire et d'un noyau. Le corps cellulaire se ramifie pour former ce que l'on nomme les dendrites. Celles-ci sont parfois si nombreuses que l'on parle alors de chevelure dendritique ou d'arborisation dendritique. C'est par les dendrites que l'information est acheminée de l'extérieur vers le soma, corps du neurone. L'information traitée par le neurone chemine ensuite le long de l'axone (unique) pour être transmise aux autres neurones. La transmission entre deux neurones n'est pas directe. En fait, il existe un espace intercellulaire de quelques dizaines d'Angstroms (10- 9 m) entre l'axone du neurone afférent et les dendrites (on dit une dendrite) du neurone efférent. La jonction entre deux neurones est appelée la synapse (**figure III.2**).

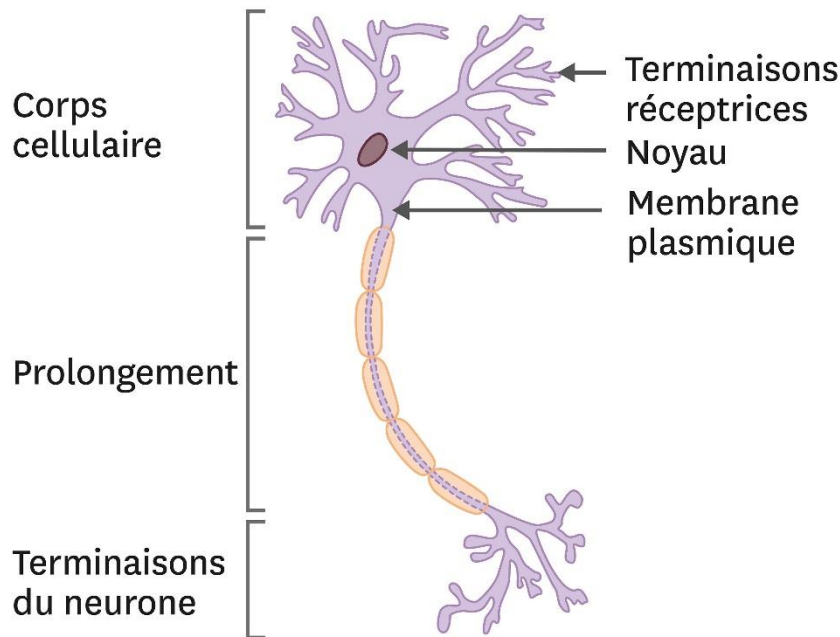


Figure III.2 Schéma simplifié d'un neurone biologique.

Les réseaux de neurones artificiels (Artificial Neural Networks – ANN) constituent une approche fondamentalement nouvelle dans le traitement de l'information. Ce sont des systèmes parallèles, adaptatifs et distribués dont le fonctionnement imite celui des réseaux de neurones biologiques tout en reproduisant leurs caractéristiques de base :

1. La connaissance est acquise par le réseau à travers un processus d'apprentissage ;
2. Les connexions entre neurones, appelées poids synaptiques, sont utilisées pour le stockage de la connaissance.

Du point de vue structurel, un réseau de neurones est d'un certain nombre d'unités de traitement simples appelées neurones formels ou artificiels. Ces derniers sont connectés entre eux de façon à produire la réponse correspondant aux entrées reçues par le réseau. Plusieurs modèles de neurones artificiels ont été développés [38], s'inspirant du principe de fonctionnement du neurone biologique qui assure essentiellement les fonctions suivantes :

- Réception des signaux provenant des neurones voisins.
- Intégration de ces signaux.
- Génération d'une réponse.
- Transmission de celle-ci à d'autres neurones.

III.4. Eléments de base

III.4.1. Structure de base

Le premier modèle du neurone formel date des années quarante. Il été présenté par Mc Culloch et Pitts [37]. S'inspirant de leurs travaux sur les neurones biologiques ils ont proposés le modèle suivant:

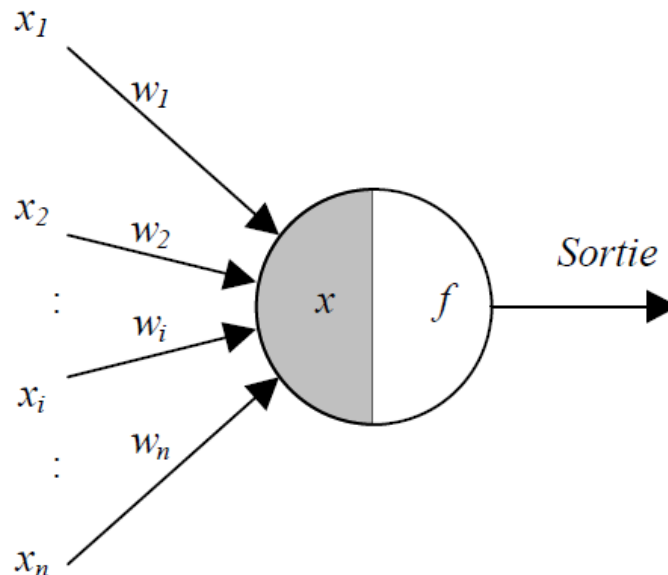


Figure III.3 le neurone formel.

Un neurone formel est une fonction algébrique non linéaire et bornée, dont la valeur dépend des paramètres appelés poids synaptiques ou poids des connexions. D'une façon plus générale, un neurone formel est un élément de traitement (opérateur mathématique) possédant n entrées (qui sont les neurones externes ou les sorties des autres neurones), et une seule sortie [39]. Ce modèle est décrit mathématiquement par les équations suivantes:

$$\begin{aligned} s &= \sum_{i=1}^n w_i x_i \\ y_n &= \varphi(s) \end{aligned} \quad \text{(III-1)}$$

Où w_i, x_i, φ et y_n sont respectivement, les entrées, les poids synaptiques, la fonction d'activation et la sortie du neurone.

III.4.2 Fonction d'activation

Les fonctions d'activations représentent généralement certaines formes de non linéarité. L'une des formes de non linéarité la plus simple, et qui est appropriée aux réseaux discret, est la fonction signe [40], **figure III.4.a**. Une autre variante de ce type des non linéarités est la fonction de Heaviside, **figure III.4.b**. Pour la majorité des algorithmes d'apprentissage il est nécessaire d'utiliser des fonctions sigmoïdes différentiables, telles que la fonction sigmoïde unipolaire, **figure III.4.c** et la

fonction sigmoïde bipolaire, **figure III.4.d**. La classe, la plus utilisée des fonctions d'activation, dans le domaine de la modélisation et de la commande des systèmes non linéaires est la fonction sigmoïde bipolaire.

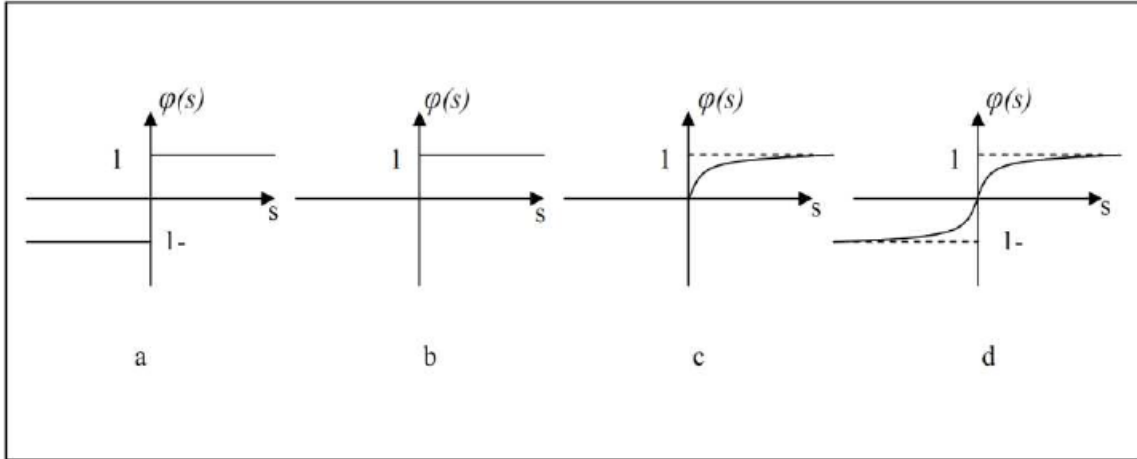


Figure III.4 Différentes fonctions d'activation.

Donc un réseau de neurone est un ensemble d'éléments de traitement de l'information, avec une topologie spécifique d'interconnexions (architecteur du réseau) entre ces éléments et une loi d'apprentissage pour adapter les poids de connexions (poids synaptiques), il est caractérisé par un parallélisme à gain très fin et à forte connectivité [38]. Nous entendons par là que dans un réseau de neurones donné, l'information est traitée par grand nombre de processeurs élémentaires très simples, chacun étant relié à d'autres processus. Ce processus très simple est un neurone formel désigné ainsi son fonctionnement s'inspire d'une modélisation des cellules biologiques. Ils sont dotés de deux propriétés importantes qui sont à l'origine de leur intérêt pratique des domaines très divers:

- Capacité d'adaptation ou d'apprentissage qui permet au réseau de tenir compte des nouvelles contraintes ou de nouvelles données du monde extérieur.
- Capacité de généralisation qui est son aptitude de donner une réponse satisfaisante à une entrée qui ne fait pas partie des exemples à partir desquels il apprend.

III.5. L'apprentissage des réseaux de neurones

L'information que peut acquérir un réseau de neurones est représentée dans les poids des connexions entre les neurones. L'apprentissage consiste donc à ajuster ces poids de telle façon que le réseau présente certains comportements désirés. En d'autres termes, l'apprentissage des réseaux de

neurones consiste à ajuster les poids synaptiques de telle manière que les sorties du réseau soient aussi proches que possible des sorties désirées. Il existe deux types d'apprentissage:

- L'apprentissage supervisé: pour lequel on dispose de la sortie désirée et qui consiste à ajuster les poids synaptiques de telle sorte à minimiser l'écart entre la sortie désirée et la sortie du réseau,
- L'apprentissage non supervisé: pour lequel le réseau de neurones organise lui-même les entrées qui lui sont présentées de façon à optimiser un critère de performances donné.

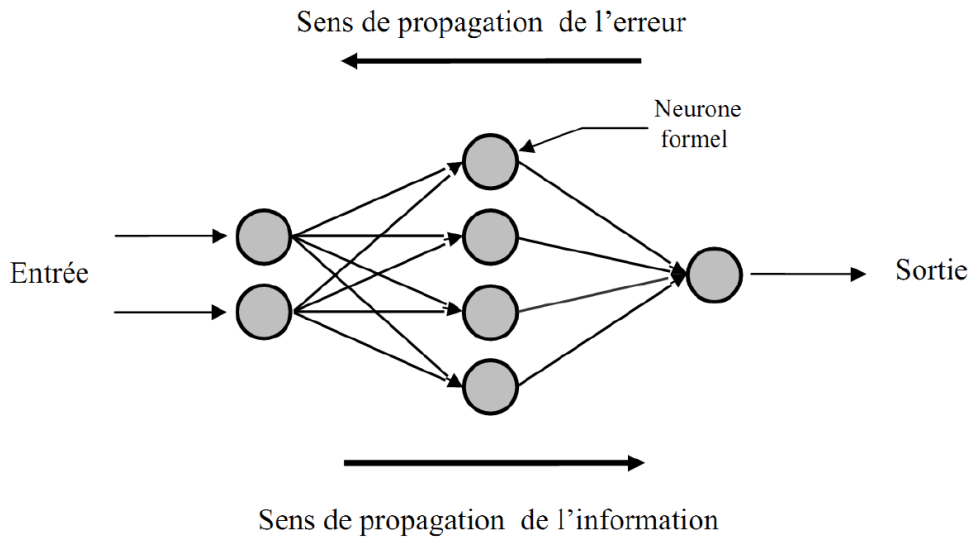


Figure III.5 Un réseau multicouche comportant 2 neurones d'entrée, 4 neurones cachés et un neurone de sortie

III.6. Algorithme d'apprentissage

L'apprentissage des réseaux de neurones, consiste à adapter les poids synaptiques de telle manière que l'erreur entre la sortie du réseau et la sortie désirée soit aussi petite que possible. La plupart des algorithmes d'apprentissage des réseaux de neurones est basée sur les méthodes du gradient : ils cherchent à minimiser un critère de la forme suivante:

$$J = \frac{1}{2} \sum_{q=1}^m \left(y_{rq}(X_p) - y_q^d(X_p) \right)^2 \quad \text{(III-2)}$$

Où y_{rq} , y_q^d sont la sortie du réseau et la sortie désirée pour le vecteur d'entrée X_p . Cette optimisation se fait d'une manière itérative [41], en modifiant les poids en fonction du gradient de la fonction de coût selon la loi d'adaptation suivante:

$$w_{ji}^l(k+1) = w_{ji}^l(k) - \eta \frac{\partial J}{\partial w_{ji}^l} \quad \text{(III-3)}$$

Où w_{ji}^l est le poids de la connexion entre le j ème neurone de la couche l et le i ème neurone de la couche $l-1$

$i=1, \dots, n+1$ la i ème composante du vecteur d'entrée.

$j=1, \dots, m+1$ la j ème composante du vecteur de sortie.

$l=1, \dots, L$ l'ordre d'une couche dans le réseau de neurone.

η est une constante positive ($\eta > 0$) appelée taux d'apprentissage.

Le gradient est calculé par une méthode spécifique aux réseaux de neurones, dites méthode de rétro-propagation [42]. Dans cette méthode il est possible de calculer le gradient de la fonction coût en retropropageant l'erreur commise en sortie vers les couches cachées.

III.7. Identification et commande des systèmes par les réseaux de neurones

Par leur capacité d'approximation universelle, les réseaux de neurones sont bien adaptés pour l'identification et commande des systèmes non linéaires. En effet dans ce cas la fonction commande est une fonction non linéaire, l'objectif est alors d'approximer cette fonction par les RNA (réseaux de neurones artificiels). Cette approximation est réalisée par apprentissage des poids du réseau, l'apprentissage peut se faire hors ligne ou en ligne :

- Dans le cas de hors ligne, l'apprentissage est basé sur un ensemble de données définissant la fonction commande,
- Dans le cas d'en ligne, la mise à jour des poids est essentiellement adaptative.

III.7.1. Identification des processus par réseaux de neurones

Le principe de l'identification par réseau neuronaux consiste à substituer aux modèles paramétriques classiques des modèles neuronaux [43], c'est-à-dire proposer un modèle établissant une relation entre son entrée et sa sortie et à déterminer, à partir du couple des signaux d'entrée-sortie, le comportement du modèle. Deux raisons importantes nous motivent:

- Prédire le comportement d'un système pour différentes conditions de fonctionnement ;
- Elaborer une loi de commande à appliquer au processus, pour qu'il réalise l'objectif assigné.

Nous citerons deux techniques d'identification à base de réseaux de neurones multicouches : la méthode d'identification directe et la méthode d'identification inverse.

III.7.1.1. Identification directe

La **figure III.6** montre le schéma général d'identification directe d'un processus. Sur cette figure, le réseau de neurones identificateur RNI est utilisé en parallèle avec un processus de type boîte noire. La sortie du processus, y , est comparée avec la sortie du réseau de neurones, \hat{y} , puis l'erreur $y - \hat{y}$ est utilisée afin d'affiner les paramètres du système neuronal.

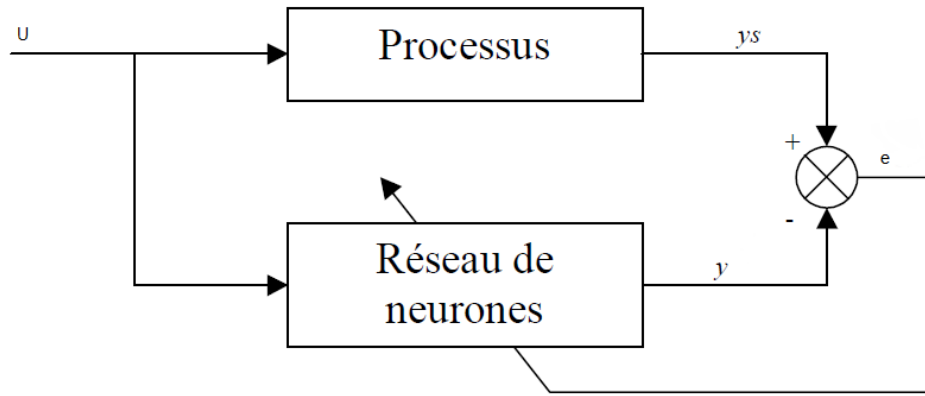


Figure III.6 Schéma d'identification directe d'un processus par réseau de neurones.

Considérons un système non linéaire avec une entrée $u(k)$ et une sortie $y(k)$

$Y(k)$ peut dépendre de $u(k)$ seulement, ou de $u(k)$ et les états précédents de y et/ou u c'est à dire

$$Y(k)=F(u(k))$$

Ou

$$y(k)=F(u(k-1), \dots, u(k-m), y(k-1), y(k-2), \dots, y(k-n))$$

Le principe de l'identification directe par réseau de neurones consiste à construire le réseau (en générale MLP) qui approxime la fonction F , cette approximation se fait par apprentissage hors ligne [44]. Etant donné la fonction F , il faut construire un RNA pour réaliser l'approximation, on génère une base de données de N échantillon $y(k-1), y(k-2), \dots, y(k-n), u(k-1), \dots, u(k-m)$ (les entrés) et on obtient $y(k)$ (la sortie) à partir de F . L'objectif de l'apprentissage est de déterminer la fonction F à partir de ces données. A l'instant t , on donne au RNA :

$Y(k-1), y(k-2), \dots, y(k-n), u(k-1), \dots, u(k-m)$ et on obtient à la sortie $\hat{y}(k)$. La mise à jour des poids du réseau est basée sur la minimisation de l'erreur entre $y(k)$ et $\hat{y}(k)$.

$$J = \frac{1}{2} \sum_{i=1}^m y_i - \hat{y}_i \tag{III-4}$$

III.7.1.2 Identification inverse

Dans cette méthode, l'entrée du processus est comparée avec la sortie de l'identificateur neuronal RNI est la sortie du processus est injectée comme entrée du réseau de neurones (figure II.7).

Après un apprentissage hors-ligne du modèle inverse, le RNI peut être configuré afin d'assurer un contrôle directe du processus.

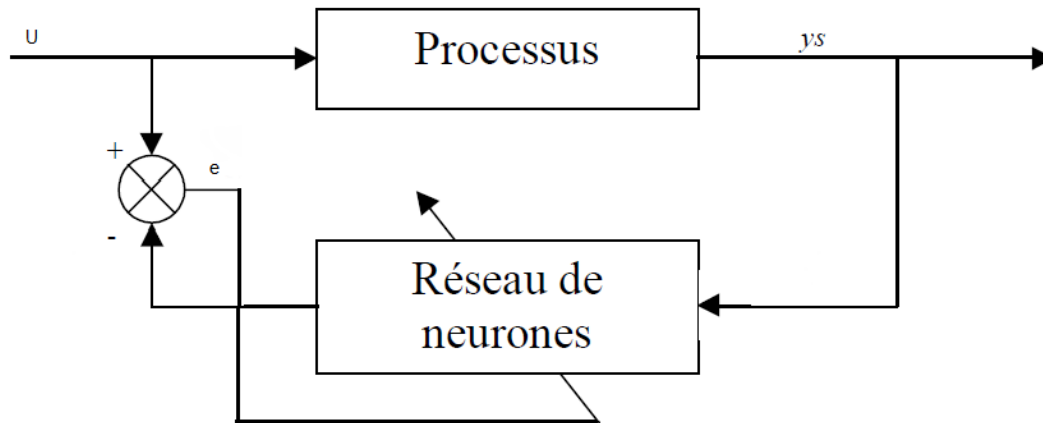


Figure III.7 Schéma d'identification inverse d'un processus avec un réseau de neurones.

III.7.2. Commande des processus par réseaux de neurones

La littérature scientifique fait mention de différentes architectures de commande. Les plus simples se basent sur l'apprentissage d'un contrôleur conventionnel déjà existant [45], d'autres opèrent un apprentissage hors-ligne du modèle inverse du processus ou d'un modèle de référence et enfin, d'autres travaillent complètement en ligne.

III.7.2.1. Apprentissage d'un contrôleur conventionnel

Un réseau de neurones peut reproduire le comportement d'un contrôleur conventionnel déjà existant (PI, PID, RST, ...) grâce à ses facultés d'apprentissage et d'approximation. Il suffit de le soumettre à un apprentissage hors ligne pendant une phase d'identification directe en considérant que le contrôleur est lui-même un processus. La figure III.8 montre le principe de l'identification directe d'un contrôleur conventionnel.

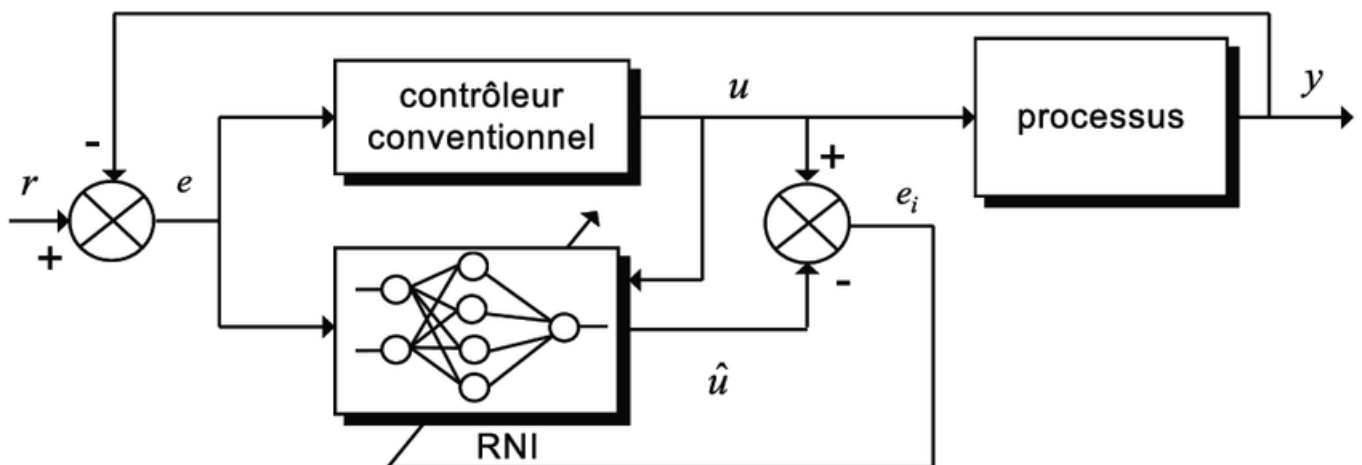


Figure III.8 Schéma d'identification directe d'un contrôleur conventionnel avec un RNI.

Le but de cette architecture n'est pas de perfectionner les performances du contrôleur conventionnel déjà existant, mais de s'affranchir des contraintes d'implémentations matérielles que peuvent nécessiter certains régulateurs. La méthode de régulation de type RST par exemple est reconnue pour ses bonnes performances en commande mais elle pose de sérieux problèmes en intégration numérique.

III.7.2.2. Commande inverse avec apprentissage en ligne

Le principe de cette commande repose sur une d'identification par modèle inverse [46]. La **figure III.9** représente le schéma de commande inverse avec un (RNC). Cette architecture reprend le même principe que celui de l'identification inverse montrée dans la figure II.8. En effet, L'entrée de référence r est comparée à la sortie y du processus pour former l'erreur de poursuite, $e = r - y$ qui sert à modifier les paramètres du réseau en ligne. Après avoir appris le modèle inverse, le neuro-contrôleur délivre la sortie u du RNC qui est la commande injectée en entrée du processus, l'erreur est alors nulle et la sortie y est égale à la référence r . Ce principe est identique au RNI de la figure II.6 ou lorsque l'apprentissage du modèle inverse est accompli, la sortie du RNI est égale à l'entrée du processus.

L'avantage de la commande inverse avec un RNC est le suivi en temps réel de l'évolution du processus, car l'apprentissage est réalisé en ligne.

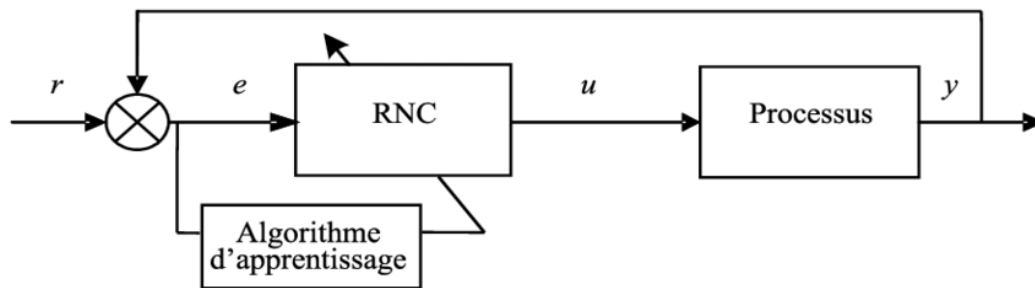


Figure III.9 Schéma de commande inverse avec un RNC.

Comme les performances de cet organe de commande dépendent étroitement de la fidélité du modèle inverse, la stabilité et le niveau de performance ne sera pas garanti dans le cas où le modèle inverse n'existe pas ou s'il est difficile à trouver.

III.8. Avantages et Inconvénients des réseaux de neurones

III.8.1. Avantages des réseaux de neurones

- Capacité de représenter n'importe quelle fonction, linéaire ou pas, simple ou complexe ;
- Faculté d'apprentissage à partir d'exemples représentatifs, par "rétropropagation des erreurs".

L'apprentissage (ou construction du modèle) est automatique ;

- Résistance au bruit ou au manque de fiabilité des données ;
- Simple à manier, beaucoup moins de travail personnel à fournir que dans l'analyse statistique classique. Aucune compétence en mathématiques, informatique statistique requise ;
- Comportement moins mauvais en cas de faible quantité de données ;
- Pour l'utilisateur novice, l'idée d'apprentissage est plus simple à comprendre que les complexités des statistiques multivariées.

III.8.2. Inconvénients des réseaux de neurones

- L'absence de méthode systématique permettant de définir la meilleure topologie du réseau et le nombre de neurones à placer dans la (ou les) couche(s) cachée(s) ;
- Le choix des valeurs initiales des poids du réseau et le réglage du pas d'apprentissage, qui jouent un rôle important dans la vitesse de convergence ;
- Le problème du sur-apprentissage (apprentissage au détriment de la généralisation) ;
- La connaissance acquise par un réseau de neurone est codée par les valeurs des poids synaptiques, les réseaux de neurones sont donc des boîtes noires où les connaissances sont inintelligibles pour l'utilisateur.

III.9. Commande par RNA directe pour la stabilisation d'attitude du quadrotor

L'architecture générale est représentée sur la figure III.10. Elle consiste à apprendre au réseau à reproduire les commandes U_2, U_3, U_4 , préconisée par le premier contrôleur à partir des erreurs entre les sorties désirées $[\varphi_d, \theta_d, \psi_d]$ et les sorties mesurées ou précédentes $[\varphi_m, \theta_m, \psi_m]$ et les variations des erreurs et éventuellement les variations des sorties actuelles $[d\varphi_m, d\theta_m, d\psi_m]$.

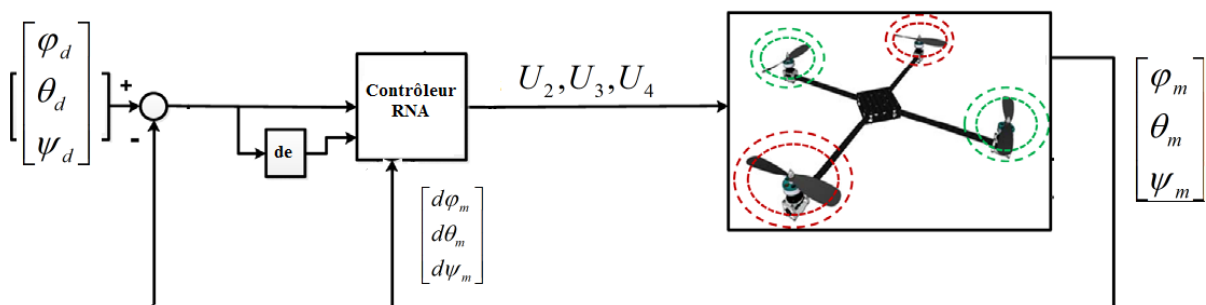


Figure III.10 : stratégie de commande RNA du quadrotor proposée.

Nous présentons dans un premier temps les résultats obtenus, à partir d'un échantillon de données recueillies sur le quadrotor simulé supposé parfaitement conforme à notre modèle, en réalisant un apprentissage hors-ligne. Nous utilisons 7005 échantillons correspondant aux paramètres utilisés pour

chaque contrôleur. L'architecture de réseau utilisé est présentée dans la figure III.11, on utilise dix neurones dans la couche cachée avec une fonction d'activation de type sigmoïde et un neurone dans la couche de sortie et une fonction d'activation de type linéaire. On choisit l'algorithme (Levenberg-Marquardt) dans la phase de rétropropagation d'erreur et l'adaptation des poids et biais.

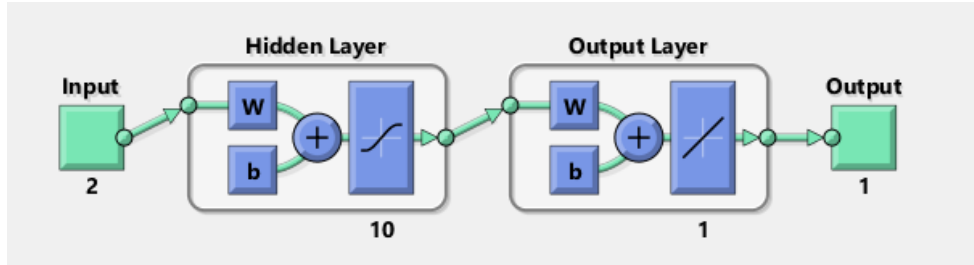


Figure III.11 La structure de RNA utilisée.

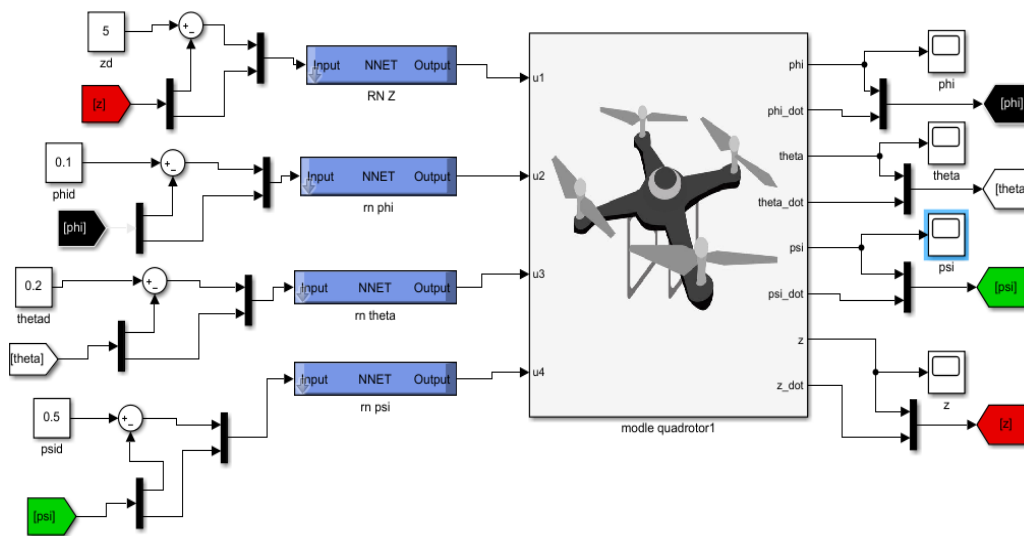


Figure III.12 la Stratégie de commande RNA du quadrotor sous Matlab Simulink.

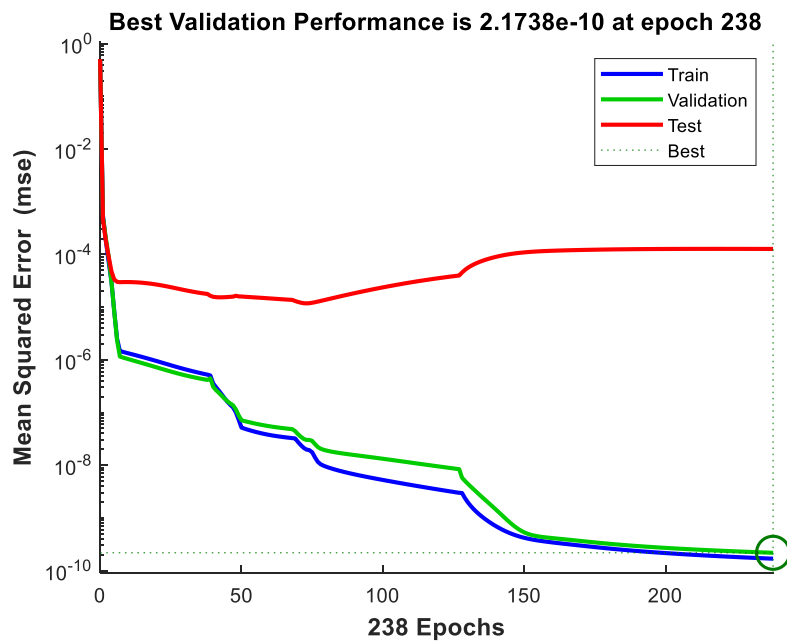


Figure III.13 Performance du modèle RNA.

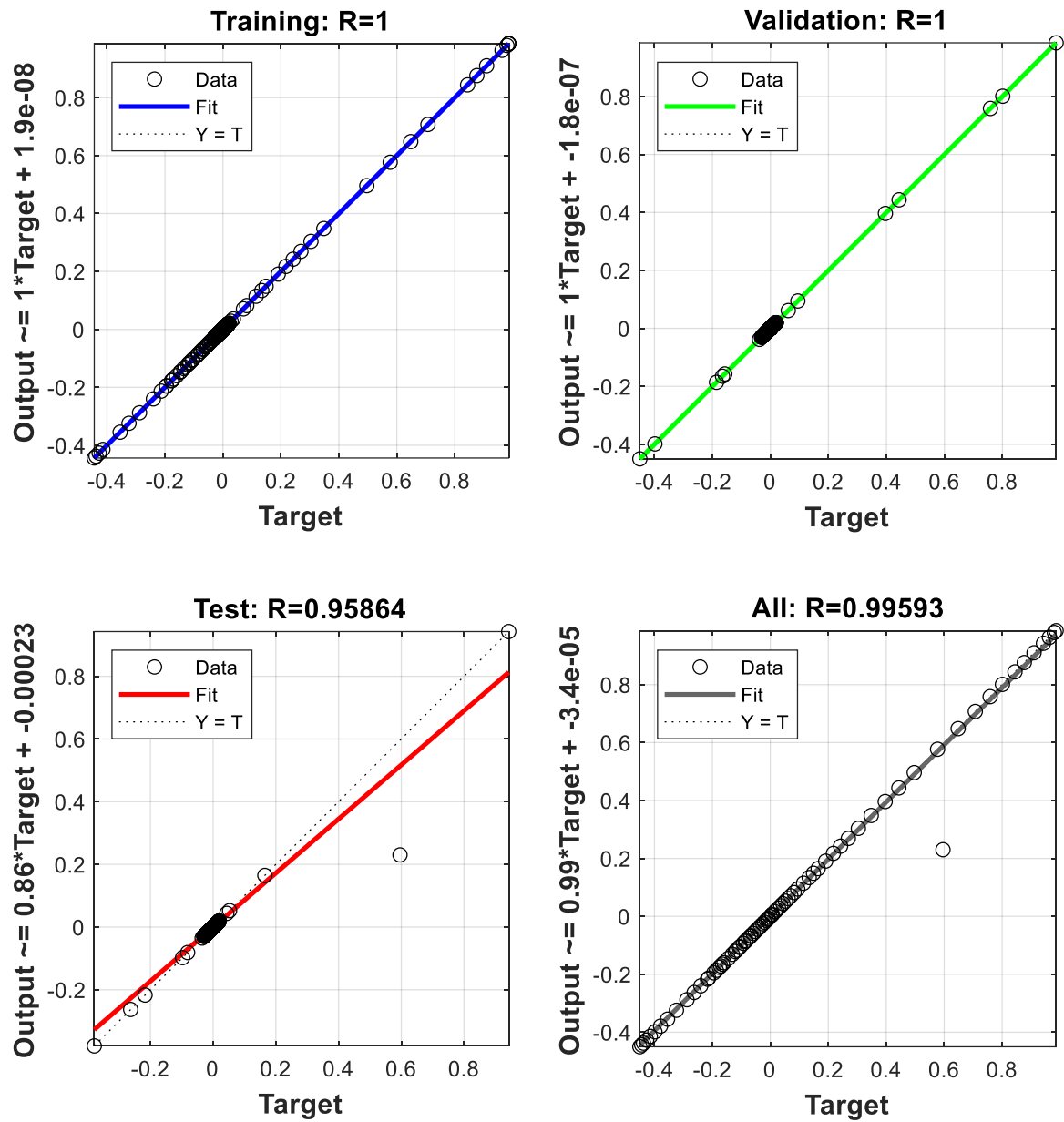


Figure III.14 Régression du modèle RNA

III.9.1 Résultats de simulation

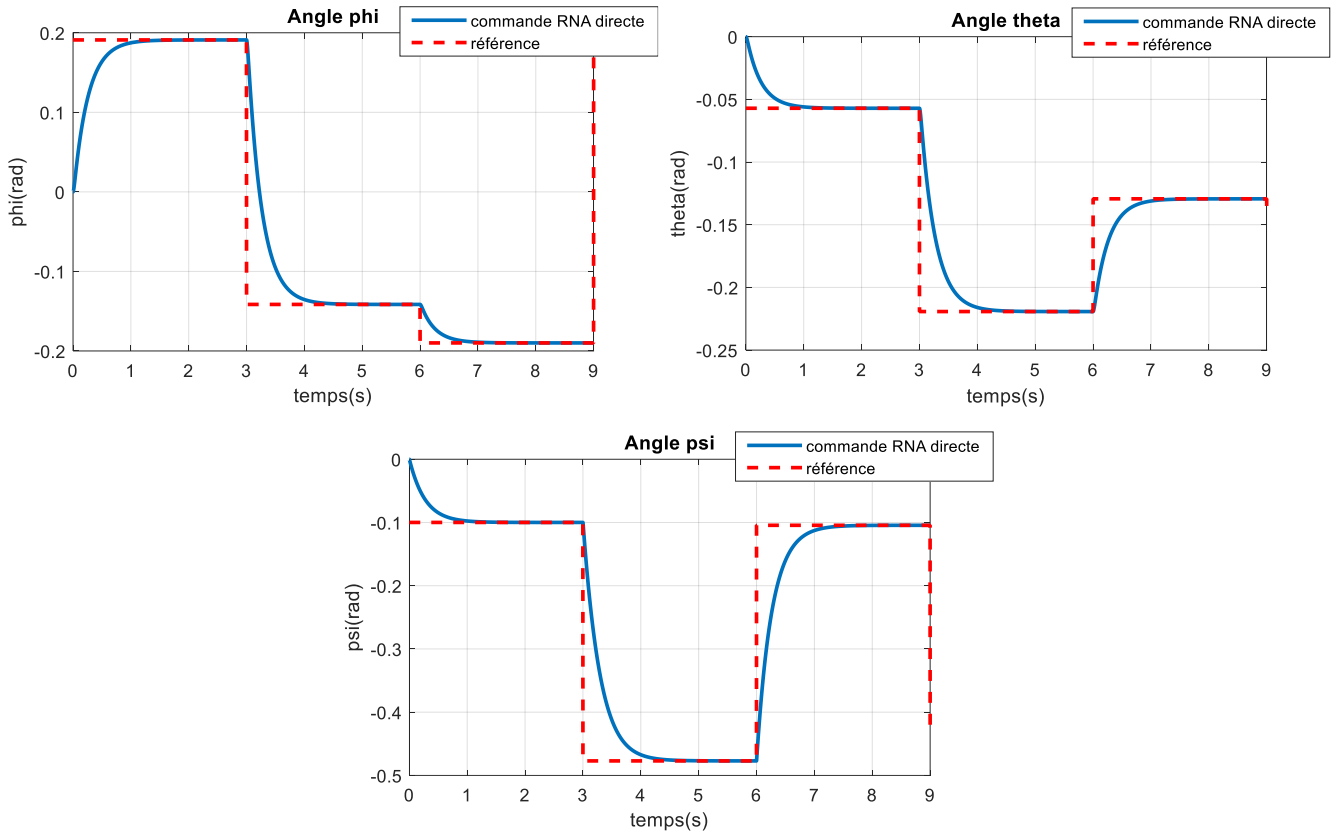


Figure III.15 stabilisation d'attitude par commande RNA directe.

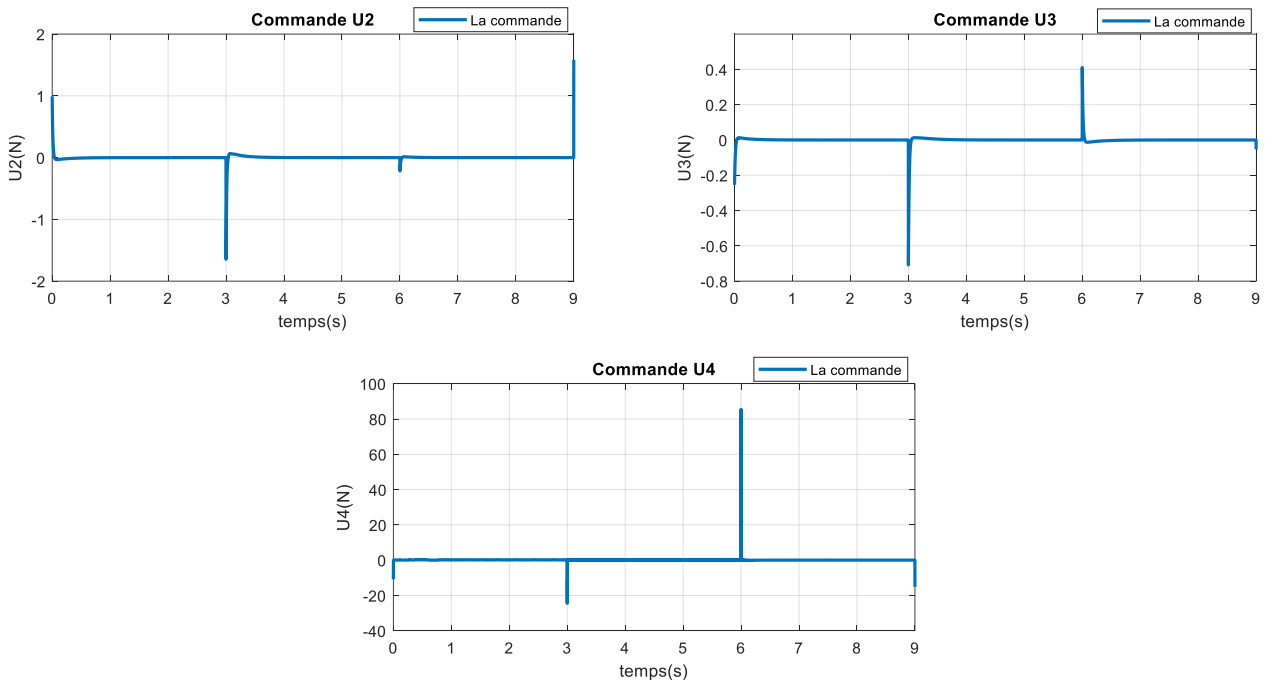


Figure III.16 les commande de RNA directe.

La première méthode utilisée pour la réalisation d'un système de commande neuronal consiste à reproduire le fonctionnement d'un contrôleur existant. Même si cette approche semble au premier abord

peu intéressant puisqu'elle nécessite l'existence d'un autre contrôleur, elle peut s'avérer utile si ce dernier est trop complexe ou trop lent pour être utilisé en temps réel, ou encore s'il utilise des données qui ne sont pas disponibles en permanence. Il est également possible de classer dans cette catégorie les cas où le système de commande reproduit est un opérateur humain.

Les résultats du contrôle du quadrotor à l'aide de la commande RNA sont montrés, dans la figure (III.14) nous avons remarqué que ce contrôleur proposé donne des bons résultats, avec un bon temps de stabilisation, environ 1 sec, et un très bon suivi de référence. Toutes les erreurs vont rapidement à zéro, de plus, il n'y a pas de dépassement ou de sous-dépassement, et a une réponse satisfaisante au test de robustesse

III.10 Commande PID-RNA pour la stabilisation d'attitude du quadrotor

Nous avons utilisé un modèle de réseaux de neurone pour ajuster les gains de la commande PID. L'architecture générale est représentée sur la figure III.16. Elle consiste à apprendre au réseau à reproduire les commandes U_2, U_3, U_4 . À partir des erreurs entre les sorties désirées $[\varphi_d, \theta_d, \psi_d]$ et les sorties mesurer ou précédentes $[\varphi_m, \theta_m, \psi_m]$ et les variations des erreurs et éventuellement les variations des sorties actuelles $[d\varphi_m, d\theta_m, d\psi_m]$.

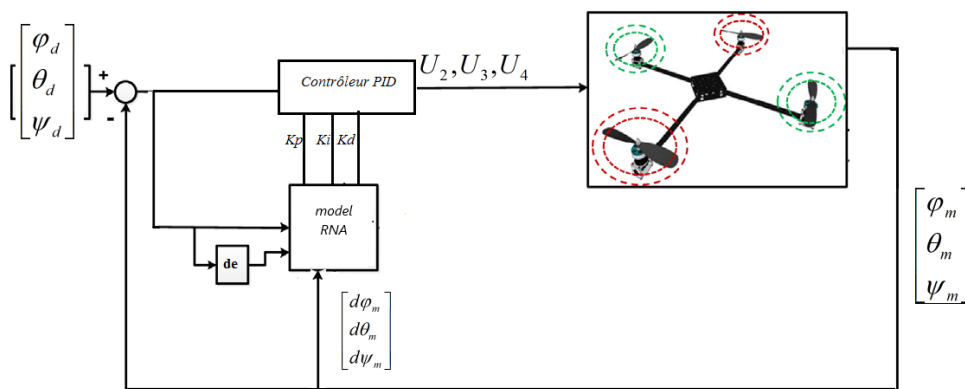


Figure III.17 : stratégie de commande PID-RNA du quadrotor proposée.

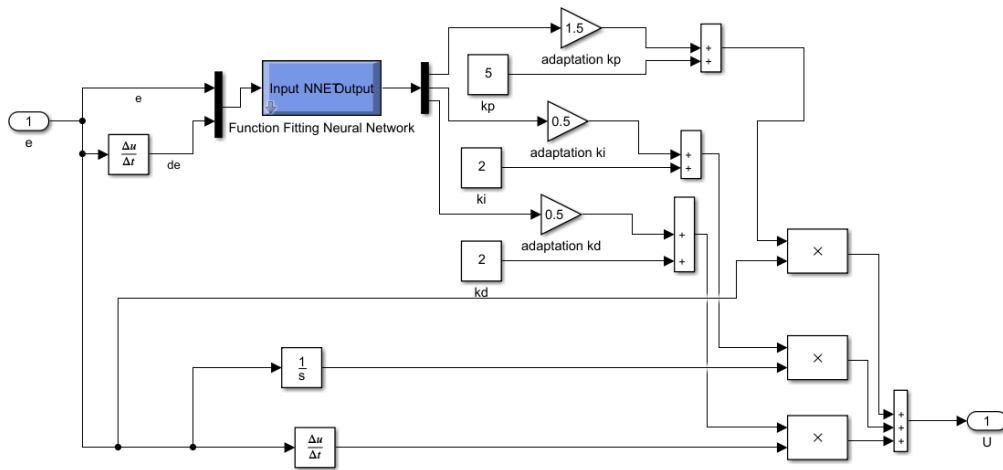


Figure III.18 blocks de commande PID-RNA.

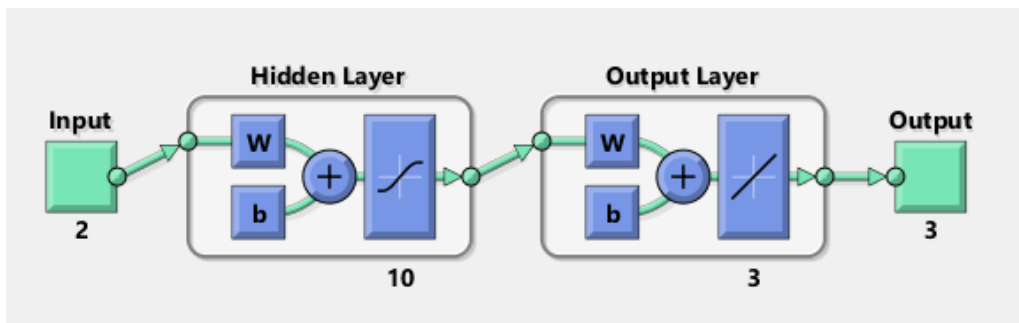


Figure III.19 : structure du modèle RNA utilisé.

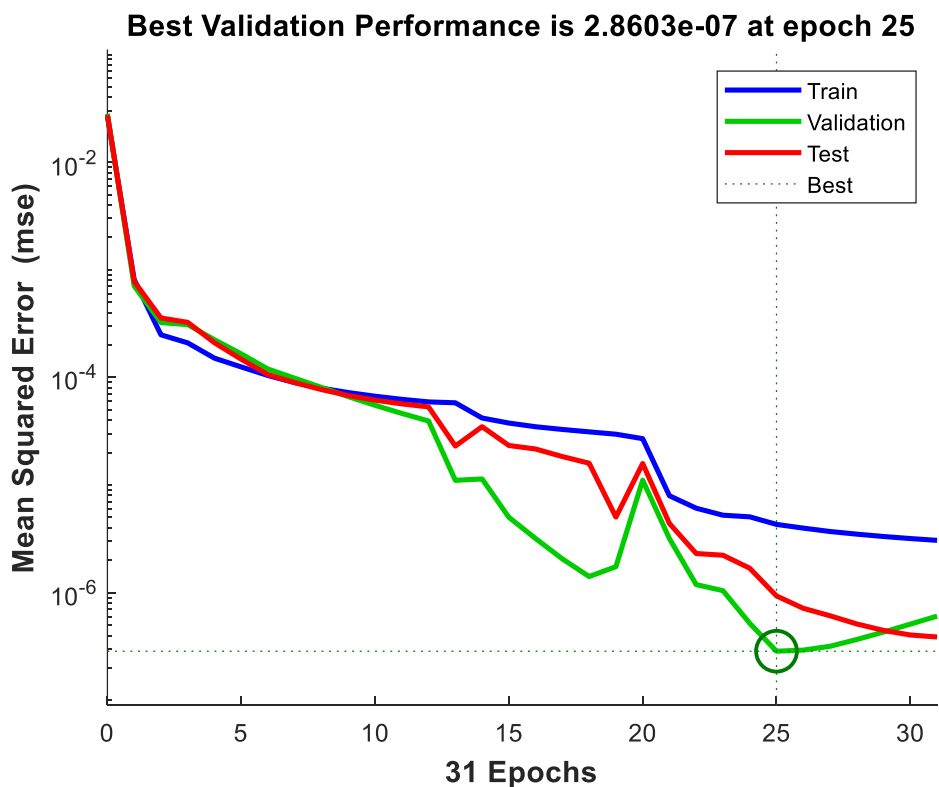


Figure III.20 : Performance du modèle PID-RNA.

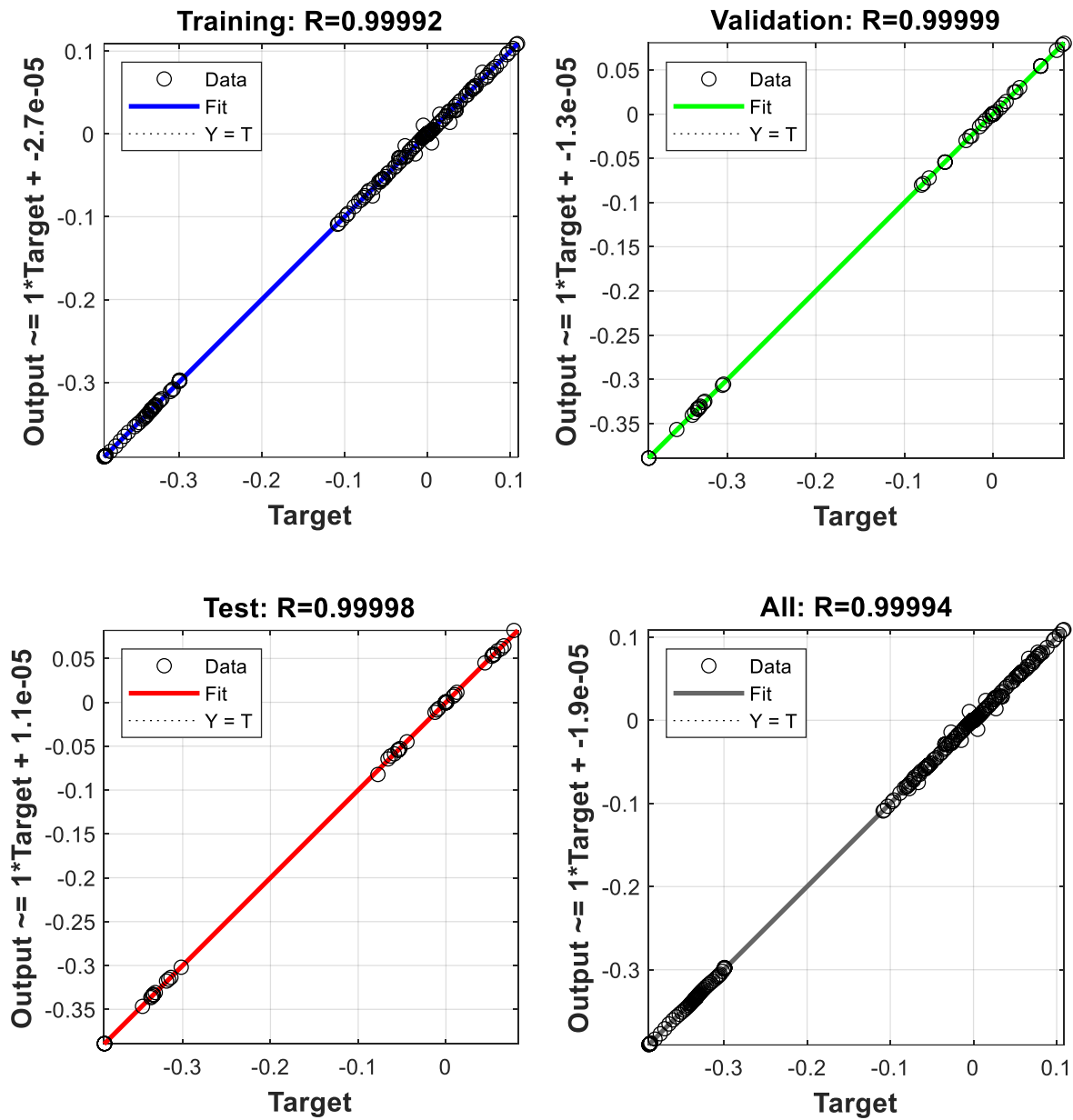


Figure III.21 : Régression du modèle PID-RNA.

III.10.1 Résultats de simulation

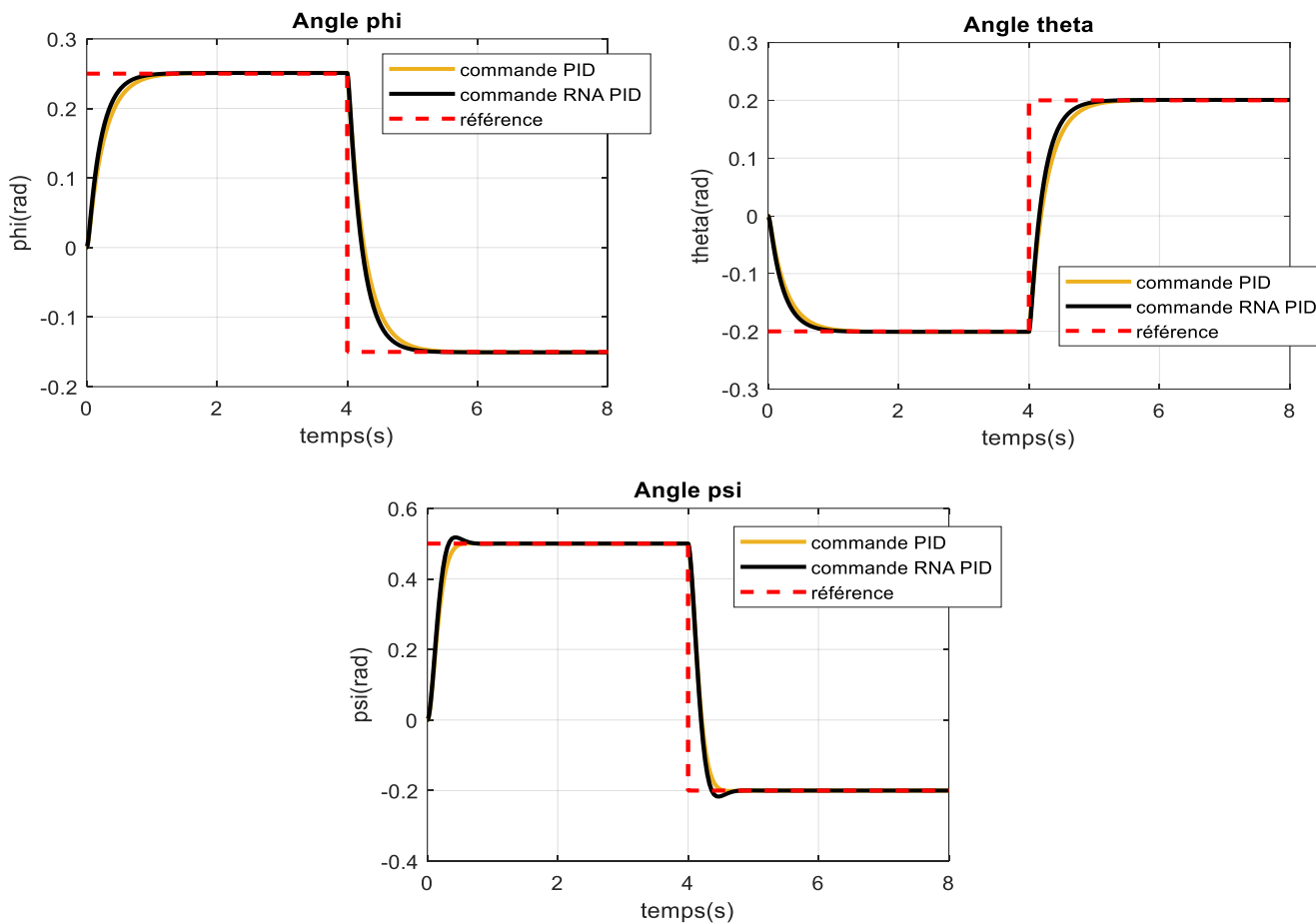


Figure III.22 stabilisation d'attitude par contrôleur PID par un modèle RNA.

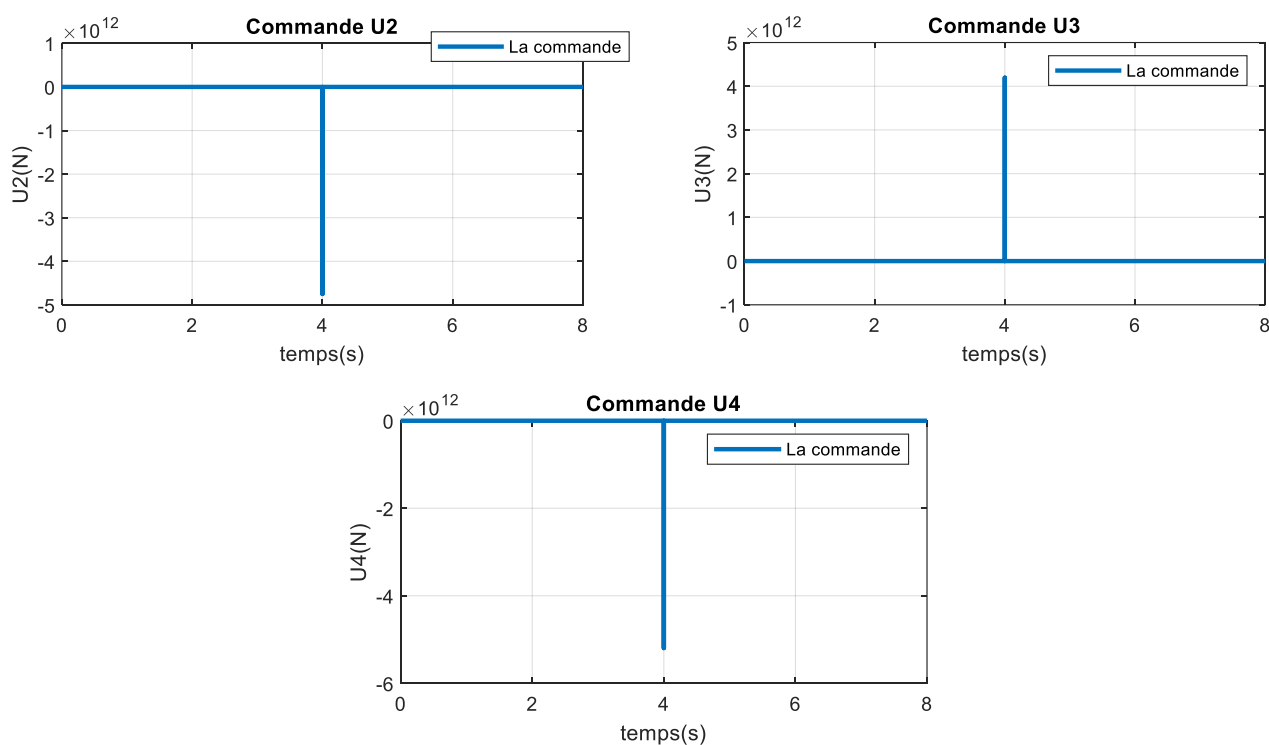


Figure III.23 les commande de contrôleur PID par un modèle RNA.

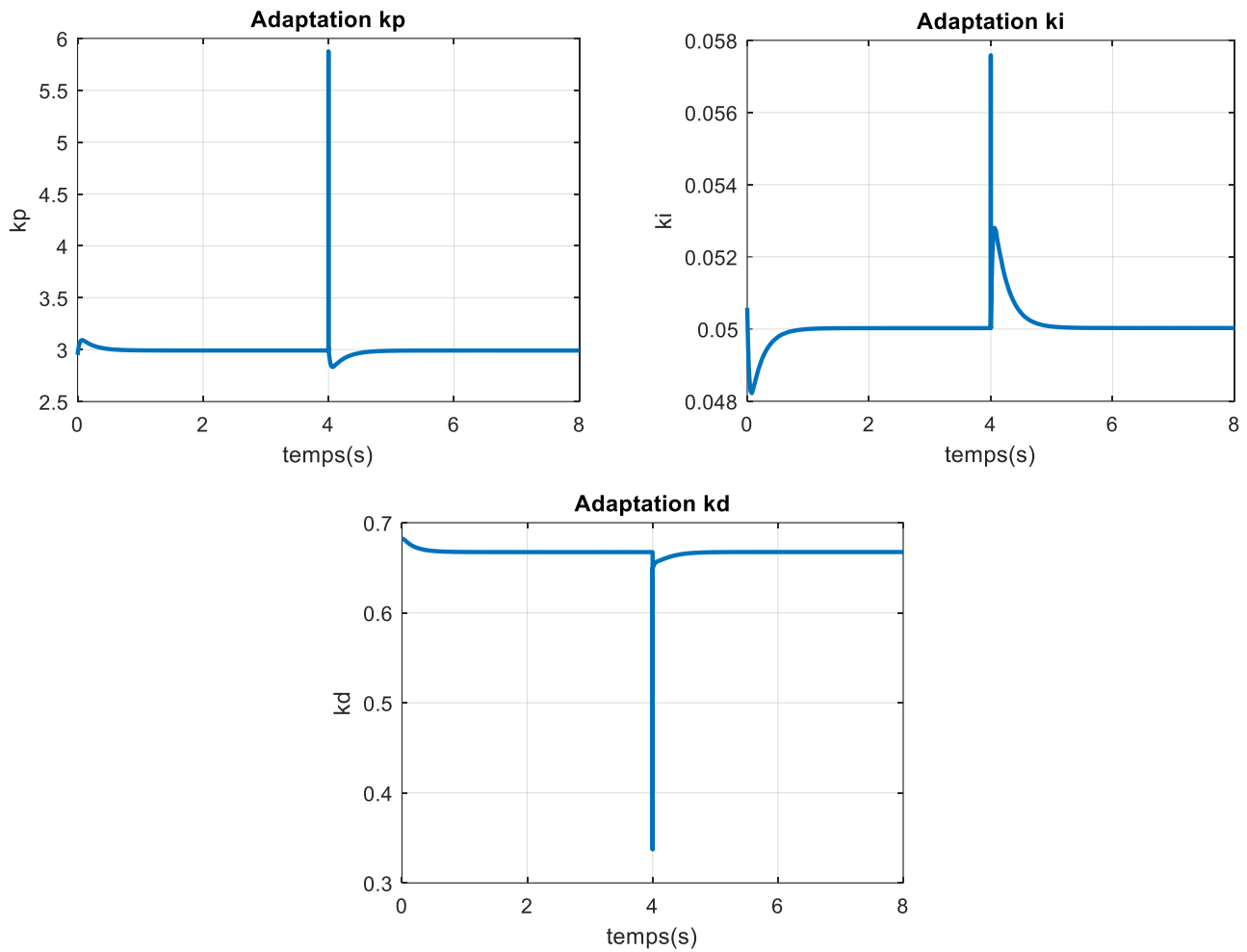


Figure III.24 l'adaptation des gains de contrôleur PID par un modèle RNA.

Les résultats obtenus montrent que l'adaptation des gains de la commande PID par un système expert RNA permet de garantir la stabilité et de réaliser une poursuite des trajectoires désirées avec une précision meilleure que celle obtenue dans le cas de la commande PID classique.

III.11 Conclusion.

Au cours de ce chapitre nous avons présenté une synthèse des différentes approches de commande utilisant des réseaux de neurones artificiels. Le choix de l'un ou l'autre de ces schémas d'utilisation sera discuté pour chaque problème abordé

La stratégie de la commande par la RNA exige moins d'informations sur le modèle du système et elle est simple et facile pour la conception. L'adaptation des gains de la commande PID par un modèle RNA permet de garantir la stabilisation de l'attitude avec une précision meilleure que celle obtenue dans le cas de la commande PID classique. Les différents résultats de la simulation obtenue montrent les performances et la robustesse des contrôleurs proposés.

Conclusion générale

Avec la croissance de l'intérêt des UAV, celui des avions à atterrissage et à décollage vertical (VTOL) et le besoin d'une instrumentation discrète et surtout légère, le quadrotor a connu une grande popularité ces dernières années. Ceci dit la principale motivation pour ce travail de recherche était la synthèse de lois de commande stabilisant pour ce type d'engin chose indispensable pour la navigation.

Le travail effectué dans ce mémoire comprend la modélisation et la synthèse des commandes intelligentes robuste du drone de type quadrirotor dont le but est d'assurer la stabilité du système ainsi que la poursuite (suivi) de trajectoire.

À la fin du premier chapitre, on conclure que le quadrotor n'a pas un modèle dynamique universel. Cela peut s'exprimer par la complexité du système (modèle non linéaire et sou actionné) et l'inexistence d'une représentation simple de la dynamique du quadrotor dans tous les modes de vols. Grâce à l'étude des équations agissant sur le système et à l'utilisation des équations de Newton-Euler décrivant la dynamique de la translation et de la rotation des corps rigides, nous avons pu obtenir le modèle dynamique du système à travers le corps étudié. Enfin nous avons obtenu un modèle d'état non linéaire et avons remarqué la forte interaction entre les états.

Dans le deuxième chapitre, nous avons présenté différentes techniques de commande repos sur la logique floue (commande floue, floue PID et floue-glissante). La commande par logique floue est facile et simple pour la conception et exige moins d'information sur le modèle mathématique du système, la commande floue PID pour s'adapter aux paramètres de la commande PID classique permet de garantir la stabilité et de réaliser une poursuite des trajectoires désirées avec une précision meilleure que celle obtenue dans le cas de la commande PID classique. Ensuite on a hybridé la commande par mode de glissement avec la logique floue dans le choix de la surface de glissement pour éliminer l'effet de broutement (chattering) du «contrôle quadrotor». Notre système est enfin simulé pour tester les régulateurs synthétisés, pour assure la stabilité et la robustesse du quadrotor et nous proposons un modèle Simulink. Visualisez, observez et commentez ces tests.

Le troisième chapitre se concentre sur la commande par RNA, deux techniques sont utilisées (RNA directe, RNA-PID). La commande par RNA directe simple et facile la conception et exige moins d'informations sur le modèle mathématique du quadrotor, la commande RNA-PID est plus précisée que celle obtenu dans le cas PID classique, assure la stabilité du quadrotor en attitude et en altitude et améliore les performances et la robustesse du système. Nous avons effectué des simulations et effectué des tests. Nous avons introduit le modèle Simulink. Visualisez, observez et commentez ces tests.

Les résultats de simulation des lois de commande proposées sont très encourageants. Nous projetons en perspectives l'implémentation d'au moins une technique de commande sur un prototype réel afin de voir le véritable comportement de ce système dans la réalité.

Références bibliographiques

- [1] S. Bouabdallah “Design and control of quadrotors with application to autonomous flying”, école Polytechnique Fédérale de Lausanne, 2007.
- [2] Slotine, J. J. E., & Li, W. (1991). Applied nonlinear control (Vol. 199, No. 1). Englewood Cliffs, NJ: Prentice hall.
- [3] Kusiak, A. (2000). *Computational intelligence in design and manufacturing*. John Wiley & Sons.
- [4] Jack Challoner (trad. de l'anglais), L'Intelligence artificielle : Un guide d'initiation au futur de l'informatique et de la robotique, Paris, Pearson Education, 2003, 72 p. (ISBN 2-7440-1600-4).
- [5] L. Derafa, T. Madani, A. Benallegue “Dynamnic Modelling and Experirmental Identification of Four Rotors Helicopter Paramters”, IEEE 1-4244-0726-5/06, 2006, pp 1834-1839.
- [6] P. Pounds, R. Mahony, P. Hynes, J. Roberts “Design of a four rotor aerial robot”, Australasian conference on robotics and automation, 2002, pp 145-150.
- [7] T. Hamel, R. Mahony, R. Lozano, J. Ostrowski “Dynamic modelling and configuration stabilization for an x4-flyer.” 15th Triennial World Congress, Barcelona, Spain, 2002.
- [8] C.A. Patel, «Building a Test Bed for mini Quadrotor Unmanned Aerial Vehicle with protective Shroud », Master Thesis of Science, Graduate School of Wichita, USA July 2006.
- [9] Rami AbouSleiman, Dan Korff, Ermal Gjioni, and Hong Chul Yang, “The Oakland University Unmanned Aerial Quadrotor System”, for the 2008 AUVSI UAS Student Compétition.
- [10] Akira Sato. Research, development and civil application of an autonomous, unmanned helicopter. Unmanned Vehicle Systems International, www.uvs-international.org.
- [11] RFI. Le monde change, 02/07/04, 2004.
- [12] Abdellah Mokhtari and A. Benallegue,” Dynamic Feedback Controller of Euler Angles and Wind parameters estimation for a Quadrotor Unmanned Aerial Vehicle “International Conference on Robotics and Automation, USA, April 2004.
- [13] European Aeronautic Defence and Space Company, EADS N.V.
- [14] I. Kroo, F. Prinz, “The Mesicopter: A miniature rotorcraft concept – phase ii interimreport”, Stanford university, USA, 2000.
- [15] All the world’s Rotorcraft, www.aviastar.org.
- [16] S. L. Waslander, «Multi-Agent systems design for aerospace applications», the degree of doctor of philosophy, Stanford University, USA, June 2007.

- [17] H. Bouadi, and M. Tadjine “Nonlinear observer design and sliding mode control for four rotors helicopter” Proceedings of World Academy of Science, Engineering and Technology, Venise, Italy, Vol. 25, pp. 225-230, 2007.
- [18] M. Bouchoucha, M. Tadjine, A. Tayebi, and P. Müllhaupt “Step by Step Robust Nonlinear PI for Attitude Stabilisation of a Four-Rotor Mini-Aircraft”, 16th Mediterranean Conference on Control and Automation Congress Centre, Ajaccio, France, June 2008
- [19] A. Tayebi, and S. Mcgilvray “Attitude stabilisation of a four rotor aerial robot”, Proceedings of the 43rd IEEE Conference on Decision and Control, Atlantis, Paradise Island, Bahamas, December 2004.
- [20] L. Gautier, B. Hamilton, J. Hazebrouck, and T. Tournette “Piloteage d’un quadri-rotor via un FPGA”, Tuteur ESIEE engineering, Journées pédagogiques, 2008.
- [21] L. Besnard “Control of a quadrotor vehicle using sliding mode disturbance observer”, Master Thesis, Alabama university, 2006.
- [22] H. Bouadi, M. Bouchoucha, and M. Tadjine “Modelling and Stabilizing Control Laws Design Based on Sliding Mode for an UAV Type-Quadrotor” Engineering Letters, London, England, Vol. 15, No. 2, pp. 15-24, 2007.
- [23] T. Madani, and A. Benallegue “*Backstepping control for a quadrotor helicopter*” Proceedings of 2006 IEEE RSJ International Conference on Intelligent Robots and Systems, 3255 – 3260, 2006.
- [24] B. Toual “Modélisation et Commande Floue Optimisée d’une Génératrice à Double Alimentation, Application à un Système Eolien à Vitesse Variable”, Université de Batna, 2010
- [25] T. Laamayad “Commande Optimale d’une Machine Asynchrone Apport de La Logique Floue“, université de Batna, 2008.
- [26] Zadeh, L. A., Klir, G. J., & Yuan, B. (1996). Fuzzy sets, fuzzy logic, and fuzzy systems: selected papers (Vol. 6). World Scientific.
- [27] Buckley, J. J. (1992). Universal fuzzy controllers. *Automatica*, 28(6), 1245-1248.
- [28] Benyounes, A., Hafaifa, A., & Guemana, M. (2016). Gas turbine modeling based on fuzzy clustering algorithm using experimental data. *Applied Artificial Intelligence*, 30(1), 29-51.
- [29] Mamdani, E. H. (1974, December). Application of fuzzy algorithms for control of simple dynamic plant. In Proceedings of the institution of electrical engineers (Vol. 121, No. 12, pp. 1585-1588). IET.
- [30] Singh, H., Gupta, M. M., Meitzler, T., Hou, Z. G., Garg, K. K., Solo, A. M., & Zadeh, L. A. (2013). Real-life applications of fuzzy logic.
- [31] Bühler, H., Réglages par logique floue, Presses Polytechniques Romandes, 1994, 181p.
- [32] Mamdani, E. H., "Advances in the linguistic synthesis of fuzzy controllers," *International Journal of Man-Machine Studies*, vol. 8, pp. 325-334, 1976.

- [33] M. Kunz, «Helicopter indoor», projet de semestre, Ecole Polytechnique Fédéral de Lausanne, Suisse, Février 2003.
- [34] S. Bouabdallah, A. Noth and R. Siegwart, «PID vs LQ Control Techniques Applied to an Indoor Micro Quadrotor », Autonomous Systems Laboratory Swiss Federal Institute of Technology Lausanne, Switzerland, 2004.
- [35] GmbH Microdrones <http://www.microdrones.com>.
- [36] Ahcéne Boubakir “Méthodes de commande par mode de glissement appliquées au bras manipulateur et au système hydrolique CE105 à réservoirs couplés ”, Thèse Magister, Ecole Militaire Polytechnique, février 2007.
- [37] Jodouin, J-F., *Les réseaux de neurones; principes et définition*, Hermes, 1994, 124p.
- [38] Burton, B.; Kamran, F.; Harley, R. G.; Habetler, T. G.; Brooke, M. A.; Poddar, R., "Identification and control of induction motor stator currents using fast online random training of a neural network," *IEEE Trans. Ind. Applicat.*, vol. 33, pp. 697-704, May/June 1997.
- [39] Thiria, S.; Lechevallier, Y.; Gascuel, O.; Canu, S., *Statistique et méthodes neuronales*, Ed. Dunod, 1997.
- [40] Picton, P., *Introduction to neural networks*, Ed. Macmillan, 1994, 168p.
- [41] Lutaud-Brunet, M., *Identification et contrôle de processus par réseaux neuroflous*, Thèse de Doctorat de l'Université de d'Evry Val d'Essonne, 1996.
- [42] De Villier, J.; Barnard, E., "Backpropagation neural nets with one and two hidden layers," *IEEE Trans. on Neural Networks*, vol. 4, n°1, pp. 136-141, January. 1992.
- [43] Chen, C. H., *Fuzzy logic and neural network handbook*, IEEE Press, 1996.
- [44] XBee, <http://www.selmaware.com/appbee>
- [45] H. Bouadi, «Synthèse de lois de commande stabilisantes pour un UAV type Quadrotor étude et application », Thèse de Magister, Ecole Militaire Polytechnique, ALGERIE 2007.
- [46] Derafa L, Madani t. and Benallegue A « Dynamic modelling and experimental identification of four rotor helicopter parameters » CIT Mumbai, India, 2006.

Résumé : Ce travail présente l'application des techniques de l'intelligence artificielle pour contrôler les drones de type Quadrotor. Avant la discussion des méthodologies de contrôle, une description détaillée de la modélisation dynamique du Quadrotor est présentée, tenant en compte les divers phénomènes aérodynamiques, qui peuvent influencer à la structure de drone ainsi que les contraintes non holonomes. En suite diverses stratégies de contrôle intelligent robuste basées sur la logique floue et le réseau de neurones artificiel A été développé pour garantir la stabilité et la poursuite de trajectoire de quadrotor . et afin de pallier aux limites des contrôleurs conventionnels et améliorer sa robustesse, telles que la commande PID et mode glissant. Une stratégie de commande hybride a été proposée.

Mots clés: Quadrotor, commande intelligente, commande floue, RNA, PID, mode glissant.

Abstract: This work presents the application of artificial intelligence techniques to control Quadrotor UAVs. Before the discussion of the control methodologies, a detailed description of the dynamic modeling of the Quadrotor is presented, taking into account the various aerodynamic phenomena, which can influence to the UAV structure as well as the non-holonomic constraints. Subsequently various robust intelligent control strategies based on fuzzy logic and artificial neural network has been developed to ensure the stability and trajectory tracking of quadrotor. And to overcome the limitations of conventional controllers and improve its robustness, such as PID control and sliding mode. A hybrid control strategy has been proposed.

Keywords: Quadrotor, intelligent control, fuzzy control, RNA, PID, sliding mode.

ملخص: يقدم هذا العمل تطبيق تقنيات الذكاء الاصطناعي للتحكم في الطائرات بدون طيار رباعية المرواح. قبل مناقشة منهجيات التحكم، يتم تقديم وصف تفصيلي للنموذج الديناميكية لـ رباعية المرواح، مع مراعاة الظواهر الديناميكية الهوائية المختلفة، والتي يمكن أن تؤثر على هيكل الطائرات بدون طيار وكذلك القيود غير الشاملة. بعد ذلك، تم تطوير العديد من استراتيجيات التحكم الذكي القوية القائمة على المنطق الضبابي والشبكة العصبية الاصطناعية لضمان الاستقرار وتتبع المسار للرباعي. وللتغلب على قيود أجهزة التحكم التقليدية وتحسين متانتها، مثل التحكم التناسبي التكاملية التفاضلي ووضع الانزلاق. تم اقتراح استراتيجية تحكم هجينة.

الكلمات المفتاحية: رباعية المرواح، التحكم الذكي، التحكم الضبابي، الشبكة العصبية الاصطناعية، التحكم التناسبي التكاملية التفاضلي، وضع الانزلاق.