

République Algérienne Démocratique et Populaire
Ministère de l'Enseignement Supérieur et de la Recherche Scientifique

UNIVERSITE MOULOUD MAMMARI DE TIZI-OUZOU



FACULTE DU GENIE ELECTRIQUE ET D'INFORMATIQUE
DEPARTEMENT D'AUTOMATIQUE

**Mémoire de Fin d'Etudes
De MASTER ACADEMIQUE**
Domaine : **Sciences et Technologies**
Filière : **Génie électrique**
Spécialité : **Commande des Systèmes**

Présenté par

Med Amine NEHMAR

Lounis HESSAS

Mémoire dirigé par **Mohand Achour TOUAT**

Thème

**Synthèse de Lois de Commande pour
un Drone à Voilures Tournantes de
Type Quadrirotor.**

Mémoire soutenu publiquement le 26 /09/ 2017 devant le jury composé de :

M Rabah MELAH

Professeur, UMMTO, Président

M Mohand Achour TOUAT

MCB, UMMTO, Encadreur

M Mohmed ALIBAIE

MCB, UMMTO, Examineur

M Fatiha LOUADJ

MAA, UMMTO, Examineur

Remerciements

Nous remercions **dieu** de nous avoir fourni la volonté nécessaire pour mener à terme ce modeste travail.

Ce mémoire de Master a été proposé et dirigé par Monsieur **Touat Mohand Achour**, maître de conférences à l'université Mouloud Mammeri de Tizi-Ouzou (UMMTO). Ce moment est pour nous l'occasion de vous témoigner notre plus grande estime, notre profond respect et notre immense reconnaissance.

Nous tenons à remercier tous les **membres de jury**, pour l'honneur qu'ils nous ont fait en acceptant d'examiner ce travail et de leur présence en ce jury.

Nos remerciements vont aussi à tous ceux qui ont contribué de près ou de loin à l'aboutissement de ce travail.

Enfin, nous tenons à exprimer nos sincères remerciements à l'ensemble des enseignants et tout le personnel de la faculté de Génie Electrique et d'informatique (FGEI).

Dédicaces

Je dédie ce modeste travail...

A la lumière de ma vie, ma **mère**. A mon très cher **père**...sans vos sacrifices et efforts, je ne serai guère là, à écrire ces mots qui décrivent et témoignent peu mes vrais sentiments pour vous, ma fierté de vous avoir à mes côtés durant toutes les étapes de ma modeste mais heureuse vie. J'espère que je saurai à la hauteur des valeurs que vous avez semées en moi...que vous serez fier de moi comme moi je le suis.

A ma petite sœur **Kaissa**.

A mon petit frère **Ghiles**.

A ma meilleure amie **Djidji**...

Djidji, tu as été l'époque la plus belle de ma vie. C'est pourquoi, non seulement je ne pourrai jamais t'oublier, mais même je t'aurai toujours constamment dans la mémoire la plus profonde, comme une raison de vie.

A toute ma grande famille...

A tous mes amis(es) et camarades...particulièrement: **Rezak.H, Mouloud.I, Nouara.M, Amirouche.H, Naima.R, Hocine.K...** ainsi que leurs familles. Tous et toutes ne peuvent pas figurer sur cette liste mais sont à jamais dans mon cœur.

Sans oublier tous les enseignants qui ont contribué à ma formation, tous cycles confondus: primaire, secondaire, lycée et université. Je leur serai éternellement reconnaissant.

✓ **Lounis**

Dédicaces

*À la plus belle perle du mondema tendre **mère**.*

*À celui qui m'a inspiré depuis l'enfancemon cher **père**.*

*À mon petit frère **Omar**.*

*À mes sœurs **Lydia et Katia**.*

*À ma tendre **Mélissa** et toute sa famille.*

*À tous mes amis et camarades en particulier **Koceila. L, Anis. B, Agamesthane .M, Sofiane. M, Hocine. K, Amirouche. H, Arezki. O, Amirouche. A, Ammar. H, Djamel. H, Mouloud. L, Katia. B, Louiza. K, LUNA, Nessrine. M**,... Tous et toutes ne peuvent pas être cité(e)s.*

M. Amine NEHMAR

Sommaire

Introduction générale.....	1
Chapitre I : Généralités et modélisation	
I.1. Introduction.....	3
I.2. Catégories et domaines d'application de Drones.....	3
I.2.1. Classification selon l'armée de l'Air des Etas Unis d'Amérique.....	3
I.2.2. Classification selon la voilure.....	5
I.2.2.1. Drones à voilures fixes.....	5
I.2.2.2. Drone à voilures tournantes.....	6
I.2.2.3. Drones à voilures mixtes.....	7
I.2.2.4. Drones à ailes battantes.....	8
I.2.3. Domaines d'application des drones.....	8
I.3. Fonctionnement d'un Quadrirotor.....	9
I.4. Modélisation.....	12
I.4.1. Angles d'Euler.....	13
I.4.2. Vitesse angulaire.....	14
I.4.3. Vitesse linéaire.....	15
I.4.4. Effets physiques agissant sur le quadrirotor.....	15
I.4.4.1. Forces.....	15
I.4.4.2. Moments.....	16
I.4.4.3. Effets gyroscopiques.....	17
I.4.5. Modèle mathématique du quadrirotor selon EULER-NEWTON.....	17
I.4.5.1. Equation de mouvement de translation.....	19
I.4.5.2. Equation de mouvement de rotation.....	19
I.4.6. La représentation d'état du système.....	21
I.4.7. Dynamique des rotors.....	22
I.5. Conclusion.....	23

Chapitre II : Généralités sur la commande

II.1. Introduction.....	24
II.2. Commande Classique.....	24
II.3. Commande Intelligente.....	25
II.3.1. Logique floue.....	25
II.3.1.1. Sous-ensemble flou.....	26
II.3.1.2 Univers de discours.....	26
II.3.1.3. Variable linguistique.....	27
II.3.1.4. Fonctions d'appartenances.....	28
II.3.1.5. Règles floues.....	29
II.3.2. Commande floue.....	30
II.3.2.1. Structure générale d'un système floue.....	30
II.3.2.1.1. Interface de fuzzification.....	31
II.3.2.1.2. base de connaissance.....	31
II.3.2.1.3. Moteur d'inférence floue.....	31
II.3.2.1.4. Interface de défuzzification.....	32
II.3.3. Réseaux de neurones.....	34
II.3.3.1. Structure des neurones artificiels.....	34
II.3.3.1.1. Du neurone biologique au neurone formel (artificiel).....	34
II.3.3.1.2 Le neurone biologique.....	35
II.3.3.1.3. Neurone formel (artificiel).....	36
II.3.3.2. Les réseaux de neurones artificiels.....	37
II.3.3.2.1. Les réseaux non bouclés.....	38
II.3.3.2.2. Réseaux bouclés (récursifs).....	38
II.3.3.3. Algorithme d'apprentissage.....	39
II.3.3.3.1. La règle d'apprentissage du perceptron.....	40
II.3.3.3.2. Algorithme de retro propagation (back-propagation).....	41

II.3.4. Systèmes neuro-flous.....	43
II.3.4.1. Définition.....	45
II.3.4.2. Méthodes de combinaisons neuro-floues.....	45
II.3.4.2.1. Réseau flou neuronal.....	45
II.3.4.2.2. Système neuronal/flou simultanément.....	45
II.3.4.2.3. Modèle neuro-flou coopératif.....	45
II.3.4.2.4. Modèle neuro-flou hybride.....	46
II.3.4.3. Implémentation des réseaux neuro-flous.....	46
II.3.4.3.1. systèmes d'inférences neuro-flous adaptatifs (ANFIS).....	46
II.4. Conclusion.....	50

Chapitre III : Simulations et résultats

III.1. Introduction.....	51
II.2. Application de la commande PID à l'UAV.....	51
II.2.1. Choix des paramètres.....	51
II.2.2. Schéma de régulation.....	52
III.3. Application d'une commande floue à l'UAV.....	68
III.4. Réalisation d'une Neuro-floue (ANFIS) pour le contrôle D'altitude de l'UAV.....	85
III.5 Conclusion.....	91
Conclusion générale.....	92

Liste des figures

Figure I.1: Global Hawk RQ-4B.....	3
Figure I.2: RQ 170 sentinel.....	3
Figure I.3: The REAPER MQ-9.....	4
Figure I.4: The Gray EAGLE.....	4
Figure I.5: The SHADOW RQ -7.....	4
Figure I.6: The HERMES 450.....	4
Figure I.7: The BAYRAKTAR.....	4
Figure I.8: The HOVEREYE.....	4
Figure I.9: WASP.....	5
Figure I.10: CYPHER II.....	5
Figure I.11: principales composantes d'un aéronef.....	5
Figure I.12: Exemple de multicoptère.....	6
Figure I.13 : Drones à voilures mixtes.....	7
Figure I.14: Exemple d'aile Battante.....	8
Figure I.15 : Mouvement de translation, changement d'altitude.....	10
Figure I.16 : Mouvement de lacet.....	10
Figure I.17 : Mouvement de roulis.....	11
Figure I.18 : Mouvement de tangage.....	11
Figure I.19 : Géométrie du quadrirotor.	12
Figure II.1: univers de discours.	26
Figure II.2: Variable linguistique.	27
Figure II.3: Fonction d'appartenance.	28
Figure II.4: structure générale d'un système flou.	29
Figure II.5: défuzzification par centre de gravité.	32
Figure II.6: le neurone biologique.	34
Figure II.7: neurone formel.	35
Figure II.8: les fonctions d'activation les plus utilisées.	36

Figure II.9: réseaux de neurones non bouclés.	37
Figure II.10: réseaux de neurones bouclés.	37
Figure II.11: structure générale d'un réseau neuro-flou.	43
Figure II.12: principe du système neuro-flou.	43
Figure II.13: architecture ANFIS.	45
Figure III.1: schéma de régulation de vitesse des moteurs.	53
Figure III.2: réponse à un échelon de la régulation PI des moteurs.	53
Figure III.3: réaction du système pour une vitesse de 150 rad/s pour les quatre moteurs.....	54
Figure III.4: réaction du système pour une vitesse de 213 rad/s pour les quatre moteurs.....	55
Figure III.5: réponse des quatre moteurs.	55
Figure III.6: réaction du système en appliquant une différence entre les moteurs 1,3 et 2,4.....	56
Figure III.7: réaction du système En appliquant la différence entre les moteurs 2 et 4.....	56
Figure III.8: réaction du système En appliquant la différence entre les moteurs 1 et 3.....	57
Figure III.9: schéma de régulation des angles.....	58
Figure III.10: Réponse à un échelon de la régulation PID sur tous les angles.....	59
Figure III.11: réponse à un signal sinusoïdal de la régulation PID sur tous les angles.....	60
Figure III.12: Réponse face aux perturbations de la régulation PID de tous les angles	60
Figure III.13: schéma de régulation des positions.....	61
Figure III.14: Première trajectoire désirée.....	62
Figure III.15: deuxième trajectoire désirée.....	62
Figure III.16: Réponse des variables x, y, z à la première trajectoire désirée.....	63
Figure III.17: Trajectoire de l'UAV en 3D.....	64
Figure III.18: Réponse des variables x, y, z à la deuxième trajectoire désirée.....	65
Figure III.19: Trajectoire de l'UAV en 3D.....	66
Figure III.20: Réponse face aux perturbations de la régulation PID des états x, y, z.....	67
Figure III.21: Entrées du régulateur flou de l'angle phi.....	68

Figure III.22: Entrées du régulateur flou de l'angle θ	69
Figure III.23: Entrées du régulateur flou de l'angle ψ	69
Figure III.24: Sortie du régulateur flou de l'angle ϕ	70
Figure III.25: Sortie du régulateur flou de l'angle θ	70
Figure III.26: Sortie du régulateur flou de l'angle ψ	71
Figure III.27: schéma de régulation floue des angles	71
Figure III.28: réponse à un échelon de la régulation floue sur tous les angles.....	72
Figure III.29: réponse à un signal sinusoïdal de la régulation floue sur tous les angles.....	73
Figure III.30: Réponse face aux perturbations de la régulation floue des états (ϕ, θ, ψ).....	74
Figure III.31: Schéma de régulation floue des positions.....	75
Figure III.32: Entrées du régulateur flou de la position x	76
Figure III.33: Entrées du régulateur flou de la position y	76
Figure III.34: Entrées du régulateur flou de la position z	77
Figure III.35: Sortie du régulateur flou de la position z	77
Figure III.36: Sortie du régulateur flou de la position x	78
Figure III.37: Sortie du régulateur flou de la position y	78
Figure III.38 : Première trajectoire désirée.....	79
Figure III.39: Deuxième trajectoire désirée.....	79
Figure III.40: Réponse des variables x, y, z à la Première trajectoire désirée.....	80
Figure III.41: Trajectoire de l'UAV en 3D.....	81
Figure III.42: Réponse des variables x, y, z à la Deuxième trajectoire désirée.....	82
Figure III.43: trajectoire de l'UAV en 3D.....	83
Figure III.44: réponse face aux perturbations de la régulation floue des états x, y, z	84
Figure III.45: présentation de l'interface neuro –fuzzy designer.....	85
Figure III.46: Extraction des données d'apprentissages.....	86
Figure III.47: Chargement des données d'apprentissages.....	86

Figure III.48: Génération du système flou.....	87
Figure III.49: Apprentissage.....	87
Figure III.50: Test du système flou.....	88
Figure III.51: Exportation du fichier.....	88
Figure III.52: Architecture du réseau.....	89
Figure III.53: Surface du régulateur.....	89
Figure III.54: Réponse de z à un signal échelon(mamdani).....	90
Figure III.55: Réponse de z à un signal échelon(ANFIS).....	90

Liste des abréviations

Liste des abréviations

UAV: Unmanned Aerial Vehicle.

TUAV: Tactical unmanned aerial vehicle.

VTOL: Vertical Take Off and Landing.

DDL: degré de liberté.

R^T : repère fixe lié à la terre.

R^m : repère mobile.

T: matrice de transformation.

R : matrice de rotation de l'objet mobile.

φ : angle de roulis.

θ : angle de tangage.

ψ : angle de lacet.

Ω : vitesse angulaire.

C: fonction cosinus.

S: fonction sinus.

V: vitesse linéaire.

P: poids du quadrirotor.

m: masse totale du quadrirotor.

g : l'accélération de pesanteur.

F: force de poussée.

b : coefficient de portance.

Th: force de trainée.

d : coefficient de drag (trainée).

k_{ft} : Coefficient de trainée de translation.

M: moments qui agissent sur le quadrirotor.

k_{fa} : Coefficient des frottements aérodynamiques.

ζ : le vecteur de position du quadrirotor.

Liste des abréviations

J : matrice d'inertie.

F_f : force totale générée par le quadrirotor.

F_t : force de traînée selon les axes.

F_g : Force de gravité.

V : tension d'entrée du moteur.

w_i : la vitesse angulaire du rotor.

k_e : constante du couple électrique.

k_m : constante du couple mécanique.

r : résistance du moteur.

L : l'inductance du moteur.

J_r : L'inertie du rotor.

C_s : représente le frottement sec.

TOR : tout ou rien.

$u_A(x)$: fonction d'appartenance.

FA: fonctions d'appartenances.

x^* : sortie defuzzifiée.

$u_{Res}(x_R)$: fonction d'appartenance résultante.

RNA: réseaux de neurones artificiels.

z : la somme pondérée.

w_i : poids synaptiques.

x_i : entrées du neurone.

$e(t)$: l'écart entre la sortie désirée et la sortie réelle.

η : Le coefficient d'apprentissage.

ANFIS: systèmes d'inférences neuro-flous adaptatifs

Introduction générale

Introduction Générale

Depuis la nuit des temps les hommes rêvent de voler. Ce rêve sera réalisé par Icare dans la légende, puis sera autour de Clément Ader et les frères Wright de réaliser réellement cet exploit. Depuis, l'aviation n'a cessé d'évoluer et de surmonter des défis de plus en plus complexes à des fins civiles, commerciales et militaires, ces dernières ont tant préoccupé les chercheurs du domaine à cause de l'implication humaine dans des missions de surveillance aériennes ou d'explorations de zones inconnues qui s'avèrent être dangereuses, pour y remédier il fallait se dispenser de la présence de l'homme dans ce genre de missions. Ce qui a donné naissance à des véhicules volants pilotés à distance ou programmés pour exécuter des tâches précises. La littérature leur a attribué beaucoup de noms, les plus courants sont UAV (Unmanned Aerial Vehicle), aéronefs ou Drones.

La modélisation d'un drone type quadrirotor n'est pas une tâche aisée, cela s'explique par la complexité et le nombre importants d'effets agissant sur le système comme les effets gyroscopique, aérodynamiques, la force de gravité, les frottements et les forces d'inertie. Tous ces effets font que notre système est fortement non linéaire, rajoutée à ça une forte interaction entre les états et nous obtenons un système difficile à modéliser [1].

De nombreux travaux ont été fait sur la modélisation et la commande d'un quadrirotor, de nombreuses méthodes ont été utilisé pour la commande tel les régulateur PI et PID, la commande à base de backstepping [2] [3] ou encore la commande par mode glissant. Ces travaux en tous le même objectif qui est de stabiliser et d'asservir le quadrirotor ou de lui faire suivre une trajectoire donnée.

Vue la difficulté de trouver un modèle mathématique fiable, la commande floue et neuro-floue constituent une alternative au contrôle de ces appareils, ce type de régulation est non linéaire et se base sur les connaissances de l'expert qui réalise la commande.

Ce mémoire s'articule autour de trois chapitres :

Le premier chapitre présentera un état de l'art sur les drone, ainsi qu'une 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 son comportement pour enfin proposer un modèle d'état en vue de réaliser une simulation de ce système sous Matlab-Simulink.

Introduction Générale

Dans le deuxième Chapitre, nous étudierons les différentes techniques de l'intelligence artificielle à savoir : les bases théoriques de la logique floue, les réseaux de neurones, les structures neuro-floues qui nous serviront lors de la phase de stabilisation du quadrirotor, le concept de variable linguistique ainsi que les notions de signification et de description floue. Ensuite nous étudierons la structure multicouche du réseau de neurones en tenant compte des différentes méthodes d'apprentissage. Nous nous concentrerons en particulier sur la projection des systèmes flous dans un réseau de neurones afin de former un système neuro-flou de type ANFIS en tirant profit de la logique floue et des réseaux de neurones.

Dans le troisième chapitre nous implémenterons une commande PID pour le quadrirotor et nous discuterons les résultats obtenus lors de la simulation. Ensuite nous synthétiserons une commande floue de type Mamdani pour l'implémenter sur notre système afin de le stabiliser et nous discuterons les résultats obtenus lors de la simulation. Enfin nous verrons les différentes étapes à suivre pour concevoir un régulateur neuro-flou type ANFIS afin de stabiliser l'altitude du quadrirotor.

Enfin nous terminerons notre travail par une conclusion générale résumant les différents résultats obtenus, quelque perspective et les difficultés rencontrées.

Chapitre I:
Généralités et
Modélisation

I.1. Introduction

Afin de concevoir un régulateur pour l'UAV, on doit d'abord comprendre profondément les mouvements du quadricopter, sa dynamique et par conséquent ses équations dynamiques. Cette compréhension est nécessaire non simplement pour la conception du contrôleur, mais aussi pour s'assurer que les simulations du comportement du véhicule sont plus proche que possible de la réalité lorsque la commande est appliquée.

I.2. Catégories et domaines d'application de Drones

Dans cette partie nous exposerons les membres de la grande famille des drones, leur classification est très difficile car elle dépend de nombreux paramètres comme l'autonomie de vol, l'altitude de vol, la charge utile embarquée, la dimension ou encore leurs masses ... etc.

I.2.1. Classification selon l'armée de l'Air des Etats Unis d'Amérique

La classification qui suit repose sur celle proposée par l'armée de l'air des états unis d'Amérique, qui répartit les drones en cinq catégories [4], [5] :

- **Les Drone HALE (de l'anglais HIGH ALTITUD, LONG ENDURENCE) :**
Ayant une grande autonomie de vol pouvant aller jusqu'à plusieurs jours et évoluent à plus de 20 000 m d'altitude.



Figure I.1: Global Hawk RQ-4B.



Figure I.2- RQ 170 sentinel .

- **Les Drones MALE (de l'Anglais Medium Altitude, Long Endurance)** : leurs autonomies est de trentaines d'heures et évoluent entre 5000 et 15000 m d'altitude.



Figure I.3: The REAPER MQ-9.



Figure I.4: The Gray EAGLE.

- **Les Drones TUAV (Tactical Unmanned Aerial Vehicle)** : ces derniers peuvent avoir une autonomie d'une dizaine d'heures et évoluent à moins de 5000 m d'altitude



Figure I.5: The SHADOW RQ -7.



Figure I.6: The HERMES 450.

- **Les MINI UAV** : ces Aéronefs ont une autonomie de quelques heures seulement et de dimensionnement de l'ordre du mètre.



Figure I.7: The BAYRAKTAR.



Figure I.8: The HOVEREYE.

- **Les Micro UAV** : ils ont une autonomie de quelques minutes et une masse inférieure à 2Kg.



Figure I.9: WASP.



Figure I.10: CYPHER II.

I.2.2. Classification selon la voilure

I.2.2.1. Drones à voilures fixes

Un drone à voilures fixes utilise ses ailes pour générer une force de levage pour compenser le véhicule. La force de poussée générée par les moteurs compense la force de traînée sur le corps de l'aéronef. La figure (1.11) montre les principales composantes de l'aéronef [6].

- **Fuselage** : est le corps principal central de l'aéronef.
- **Ailes** : fournissent principalement la force de levage. Les surfaces mobiles sur les ailes sont des ailerons.
- **Empennage** : est le groupe de queues qui se compose des stabilisateurs, gouverne de profondeur et gouverne de direction.
- **Groupe motopropulseur** : est un groupe d'assemblage de turbines (moteurs) et hélices.



Figure I.11: Principales composantes d'un aéronef.

I.2.2.2. Drone à voilures tournantes

Les drones à voilures tournantes appartiennent tous au groupe de véhicule VTOL (Vertical Take Off and Landing), qui définit les appareils à décollages et atterrissages Verticaux et sont capables de faire du vol stationnaire. Ils sont principalement utilisés comme plateforme expérimentale. Ils servent ainsi à mettre en pratique et à illustrer des travaux de recherche. Ils utilisent un ou plusieurs rotor(s) pour se sustenter et se propulser ; on distingue [4], [5] :

- Les monorotors.
- Les birotors contrarotatifs.
- Les trirotors.
- Les Quadrirotors.
- Les Hexarotors.
- Les Octorotors.



Figure I.12: Exemple de multicoptère.

I.2.2.3. Drones à voilures mixtes

Les drones à voilure fixe ont l'avantage de pouvoir voler à vitesses élevées pour une longue durée avec des structures relativement simples. Ils constituent donc la plus grande famille de drones, allant de Le RQ-4 Global Hawk avec une envergure de 39,8 m à l'Aero Vironment Guêpe avec une envergure de 72 cm. Cependant, ils exigent une Piste de décollage/atterrissage ou des équipements de lancement et de récupération supplémentaires.

D'un autre côté, les UAV à voilures tournantes sont capables d'effectuer un décollage et atterrissage vertical avec des simplicités de maniabilité, même à très basse vitesse. Cela est très utile, surtout dans les zones urbaines et à l'intérieur.

Les conceptions hybrides combinent les avantages des UAV à voilures tournantes où la poussée du rotor est utilisée comme source de vol vertical et la poussée du rotor et les ailes sont utilisées comme source de propulsion et de levage en vol horizontal. Cette stratégie ajoute généralement une certaine complexité mécanique et des problèmes de contrôle, notamment en transition entre deux modes de vol. Le Quad tilt-rotor UM QTR de l'université de Maryland des Etats Unis d'Amérique et le quad tilt-wings QTW de l'université de Chiba du JAPON, ainsi que le VertiKul sont des exemples de ce type de mixage illustrés ci-dessous [7], [8]:

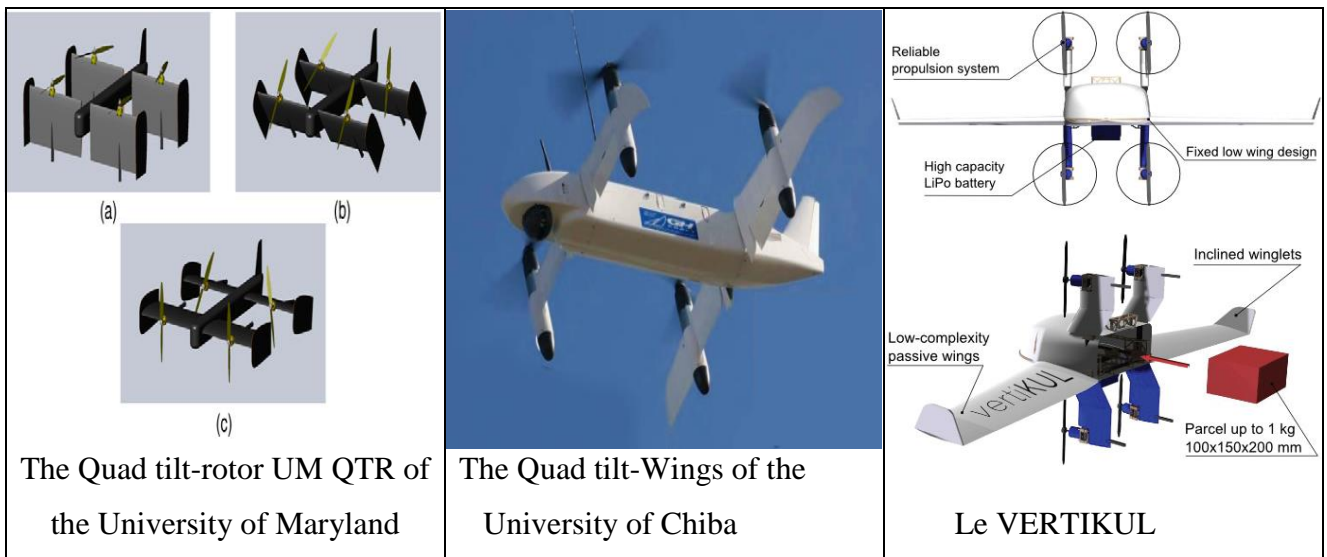


Figure I.13: Drones à voilures mixtes.

I.2.2.4. Drones à ailes battantes

La tendance actuelle est une course à la miniaturisation et on projette dorénavant la mise au point de Micro-Drone voire même des Nano-Drones (de l'ordre du Cm). C'est dans cette évolution vers plus de miniaturisation, de discrétion et d'agilité qu'est né le concept d'ailes battantes. Outre la capacité de vol sur place, les ailes battantes présentent également une

discrétion acoustique supérieure à celle des voilures tournantes. De plus, comme le démontre les insectes, ce genre de drones sont dotés d'une très grande agilité face aux rafales de vent [9].

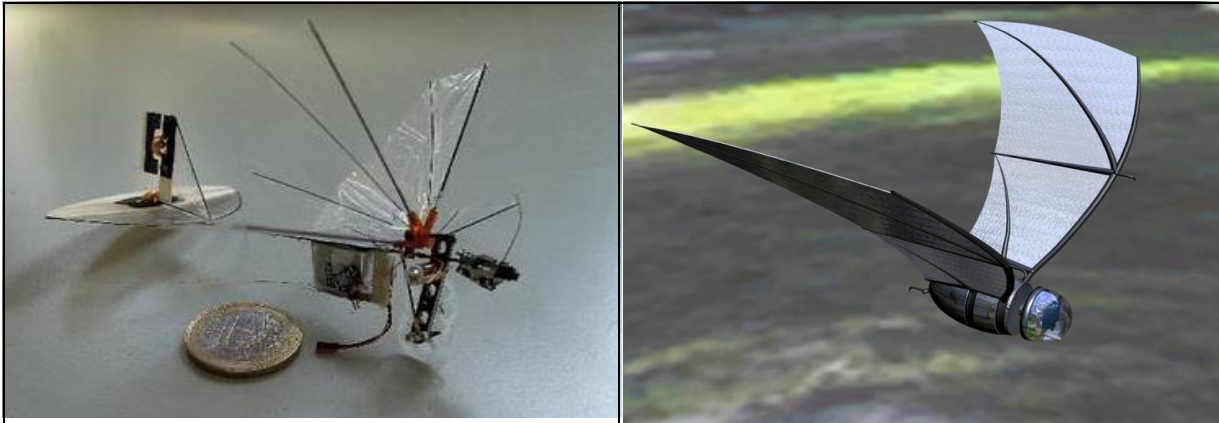


Figure I.14: Exemple d'aile Battante

I.2.3. Domaines d'application des drones

L'utilisation des drones est de plus en plus répandue, professionnelle ou non, militaire ou civile, dans différents domaines ou secteur d'activités, nous essayerons de citer certains domaines où l'utilisation est plus accrue [10], [11].

<p style="text-align: center;">Militaire</p> <p>Malheureusement l'usage premier des drones est militaire. Ces drones sont employés dans la guerre contemporaine servent à éliminer des cibles stratégiques sans mettre en danger les pilotes, ou bien pour la surveillance et la reconnaissance du terrain.</p>	<p style="text-align: center;">Agriculture</p> <p>Au début de l'année 2016, le drone agricole avait une place de choix dans les allées du pôle innovation, ces drones peuvent :</p> <ul style="list-style-type: none"> ✓ Repérer et reconnaître les hétérogénéités du sol. ✓ Permettre de créer des zones sur une parcelle pour varier la densité des semis. ✓ Etablir avec précision les besoins des plantations en examinant la densité végétale.
<p style="text-align: center;">Multimédia</p> <ul style="list-style-type: none"> ✓ Prise de vue / cinéma. ✓ Communication et marketing. ✓ Mise en valeur des régions touristiques, immobilier/monuments. 	<p style="text-align: center;">Cartographie</p> <ul style="list-style-type: none"> ✓ Travaux publics. ✓ Géo-référencement

<p style="text-align: center;">Environnement</p> <ul style="list-style-type: none"> ✓ Gestion de l'eau. ✓ Détection de flore et faune. 	<p style="text-align: center;">Inspection</p> <ul style="list-style-type: none"> ✓ Ouvrages d'art. ✓ Réseaux de communication. ✓ Thermographie. ✓ Eolienne / photovoltaïque. ✓
<p style="text-align: center;">Surveillance</p> <ul style="list-style-type: none"> ✓ Infrastructure. ✓ Entrepôt /bâtiment. ✓ Trafic routier. ✓ Frontières. 	<p style="text-align: center;">Sante / urgence</p> <ul style="list-style-type: none"> ✓ Transport de médicaments. ✓ Transport de matériels d'urgence
<p style="text-align: center;">Sport et loisirs</p> <ul style="list-style-type: none"> ✓ C'est avant tout un sport mécanique. ✓ Suivi d'évènements sportifs/concert ... etc. 	<p style="text-align: center;">Transport et livraison</p> <ul style="list-style-type: none"> ✓ Amazon a réalisé la première livraison par drone au profit d'un particulier en Grande-Bretagne le 07 Décembre 2016. [12].

I.3. Fonctionnement d'un Quadrirotor

Un quadrirotor est un mobile aérien à quatre rotors définis dans l'espace par 6DDL [13], [14], cet appareil est mécaniquement simple et possède certains avantages par rapport aux autres configurations, en raison de sa symétrie puisqu'il vole grâce à des pôles de dimension équivalente, de localiser les capteurs inertiels au centre de masse du véhicule. La somme des poussées générées par les quatre rotors permet la sustentation du véhicule. Les deux rotors situés sur l'axe latéral de l'engin ont un sens de rotation opposé à celui des deux autres rotors situés sur l'axe longitudinal afin de pouvoir réguler la somme des couples de trainée et ainsi réguler la vitesse de rotation de lacet [15].

Le fonctionnement d'un quadrirotor est assez particulier en faisant varier astucieusement la puissance des moteurs, il est possible de le faire monter /descendre, de l'incliner à gauche/droite (roulis) ou en avant/arrière (tangage) ou encore de le faire pivoter sur lui-même (lacet) [16]. Ces six degrés de libertés sont composés de trois mouvements de rotation et trois

mouvements de translation, ils doivent être contrôlés grâce à quatre actionneurs seulement; donc c'est un système sous actionné (le nombre d'entrées est inférieure au nombre de sorties).

➤ **Mouvement de translation, changement d'altitude**

Lorsque tous les moteurs tournent à la même vitesse (généralisant une poussée égale), le corps fera un vol stationnaire (cas sans perturbations) ou gagne/perd de l'altitude en augmentant /diminuant la poussée générée (en variant la vitesse des 4 moteurs).

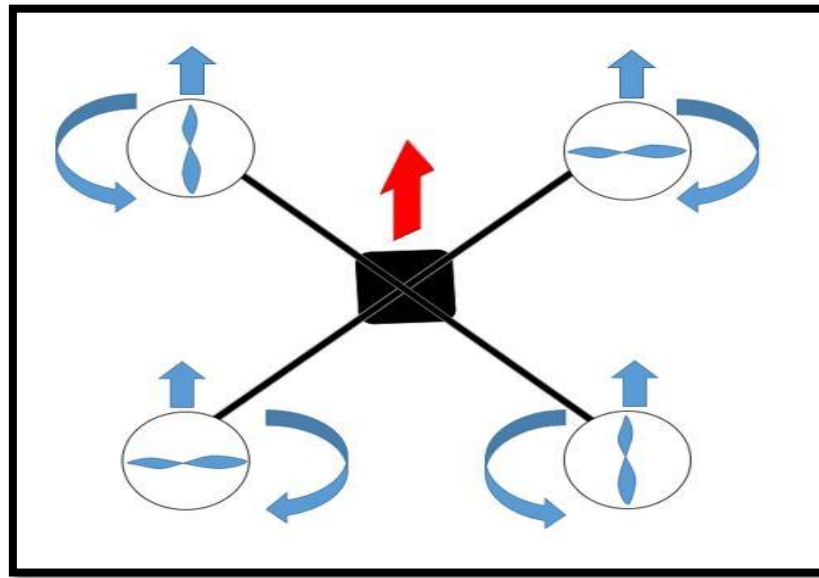


Figure I.15: Mouvement de translation, changement d'altitude.

➤ **Mouvement de lacet**

Si la vitesse des moteurs qui tournent dans le sens horaire est diminuée (ou celle des moteurs qui tournent de le sens antihoraire est augmentée), un couple net sera induit sur l'engin, ce qui résultera un changement d'angle de lacet (rotation selon l'axe vertical du corps).

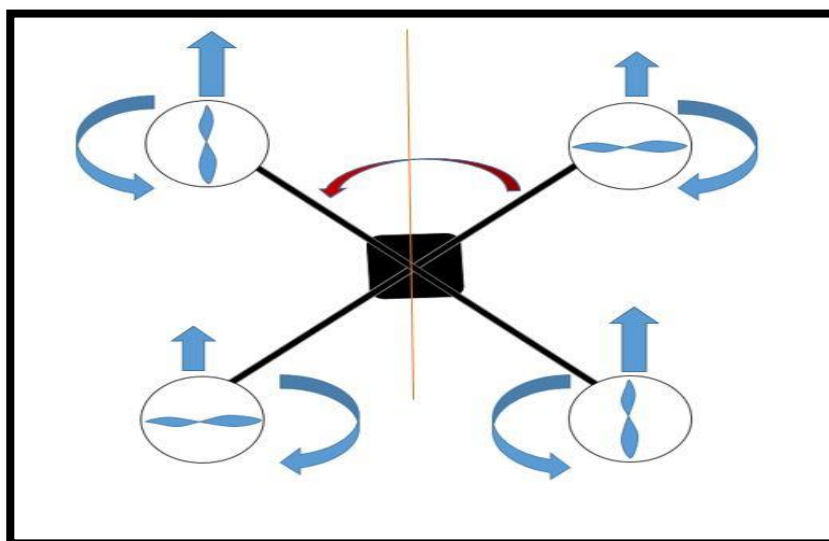


Figure I.16: Mouvement de lacet.

➤ **Mouvement de roulis (droite/gauche)**

Si la vitesse d'un moteur situé sur l'axe de roulis est diminuée ou augmentée par rapport à l'autre moteur du même axe, un mouvement de roulis sera généré et le corps roulera vers le sens du moteur auquel on a diminué la vitesse.

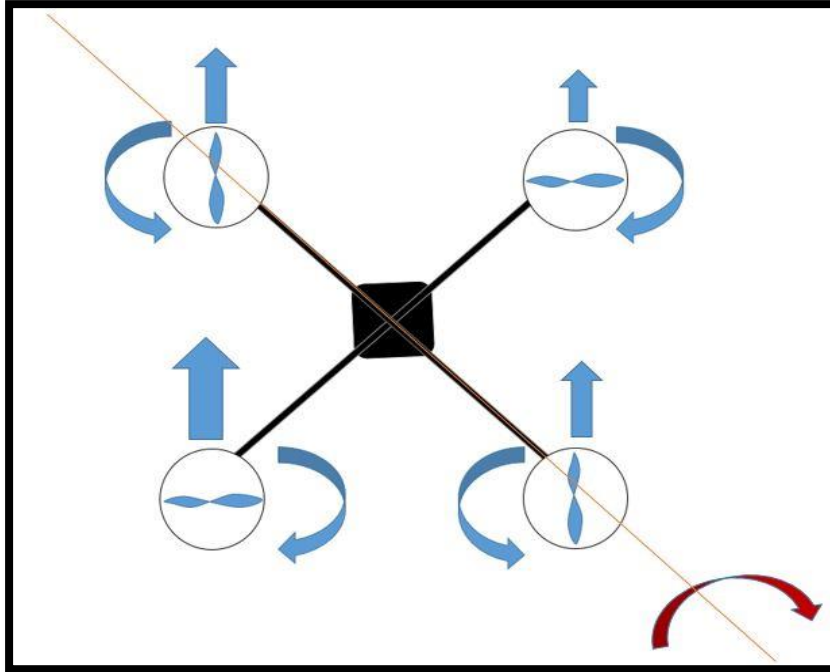


Figure I.17: Mouvement de roulis.

➤ **Mouvement de tangage (avant/arrière)**

Similairement à l'axe de roulis, si on change la vitesse d'un moteur situé sur l'axe de tangage, le corps tourne en direction de la poussée la plus petite (moteur à faible vitesse) et génère un mouvement de translation vers cette même direction.

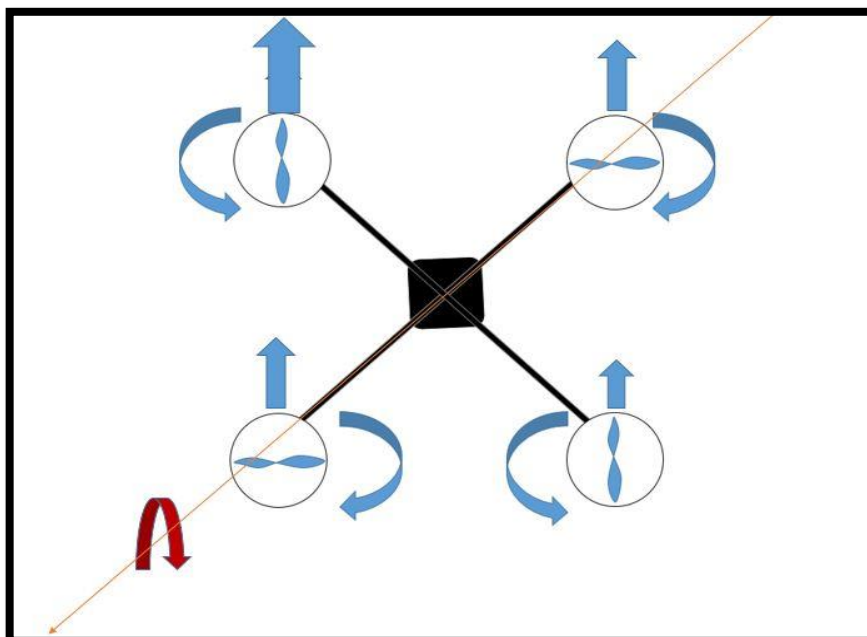


Figure I.18: Mouvement de tangage.

I.4. Modélisation

La modélisation des aéronefs est une tâche assez délicate puisque la dynamique du système est fortement non linéaire et fortement couplée, le nombre des effets qui affectent cette dernière est importante (les effets aérodynamiques, la gravité, les effets gyroscopiques, les frottements et les moments d'inertie). Après avoir vu précédemment les différents mouvements et le fonctionnement d'un quadrirotor dans (I.3), on va entamer dans ce point la modélisation dynamique du quadrirotor. Pour réaliser cette tâche, plusieurs hypothèses ont été émises dans la littérature, nous posons celles-ci comme suit [17], [18] :

- La structure du quadrirotor est supposée rigide et symétrique.
- Les hélices sont supposées rigides afin d'éviter l'effet de leurs déformations lors de la rotation.
- Le centre de masse du corps est considéré confondu avec le repère lié à la structure.
- Les forces de portance et de traînée sont proportionnelles aux carrés de la vitesse de rotation des rotors.

Afin de pouvoir modéliser mathématiquement notre quadrirotor qui est un corps mobile, on utilise deux repères, un repère fixe lié à la terre qu'on nommera R^T et un autre repère mobile R^m .

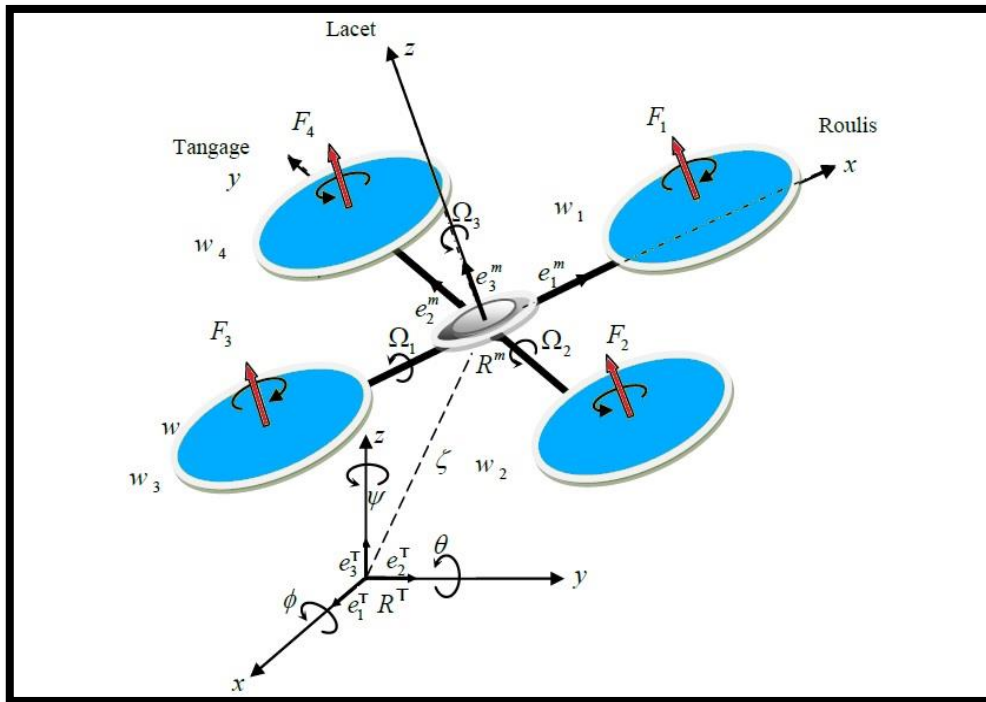


Figure I.19: Géométrie du quadrirotor.

Le passage entre le repère fixe et le repère mobile est donné par une matrice dite matrice de transformation T qui comporte l'orientation et la position du repère mobile par rapport au repère fixe.

$$T = \begin{pmatrix} R & \zeta \\ 0 & 1 \end{pmatrix} \quad (\text{I.1})$$

Avec :

R : matrice de rotation de l'objet mobile.

$\zeta = [x \ y \ z]$: le vecteur position.

Pour déterminer les éléments de la matrice R, on utilise les angles d'EULER.

I.4.1. Angles d'Euler

Au début, les deux repères coïncident, en suite le repère mobile exécute trois mouvements de rotation, une rotation autour de l'axe X avec un angle de roulis(φ), une autre rotation autour de l'axe Y avec un angle de tangage(θ), et puis une rotation autour de l'axe Z avec un angle de lacet(Ψ). Avec : $(-\frac{\pi}{2} \leq \varphi \leq \frac{\pi}{2})$, $(-\frac{\pi}{2} \leq \theta \leq \frac{\pi}{2})$ et $(-\pi \leq \psi \leq \pi)$.

Ces matrices de rotation sont données par :

$$R(x, \varphi) = \begin{pmatrix} 1 & 0 & 0 \\ 0 & c\varphi & -s\varphi \\ 0 & s\varphi & c\varphi \end{pmatrix}; \quad R(y, \theta) = \begin{pmatrix} c\theta & 0 & s\theta \\ 0 & 1 & 0 \\ -s\theta & 0 & c\theta \end{pmatrix}; \quad R(z, \psi) = \begin{pmatrix} c\psi & -s\psi & 0 \\ s\psi & c\psi & 0 \\ 0 & 0 & 1 \end{pmatrix} \quad (\text{I.2})$$

La matrice de rotation complète est le produit des trois matrices de rotation successives.

$$R(\varphi, \theta, \Psi) = R(z, \Psi) R(y, \theta) R(x, \varphi) \quad (\text{I.3.a})$$

Ce qui nous donne [1], [18]:

$$R = \begin{pmatrix} c\psi c\theta & c\psi s\theta s\varphi - s\psi c\varphi & c\psi s\theta c\varphi + s\psi s\varphi \\ s\psi c\theta & s\psi s\theta s\varphi + c\psi c\varphi & s\psi s\theta c\varphi - s\psi s\varphi \\ -s\theta & c\theta s\varphi & c\theta c\varphi \end{pmatrix} \quad (\text{I.3.b})$$

I.4.2. Vitesse angulaire

Les vitesses $\Omega_1, \Omega_2, \Omega_3$ sont les vitesses angulaires exprimées dans le repère fixe tel que :

$$\Omega = \begin{bmatrix} \Omega_1 \\ \Omega_2 \\ \Omega_3 \end{bmatrix} = \begin{bmatrix} \dot{\phi} \\ 0 \\ 0 \end{bmatrix} + Rot(x, \varphi)^{-1} \begin{bmatrix} 0 \\ \dot{\theta} \\ 0 \end{bmatrix} + ((Rot(y, \theta)Rot(x, \varphi))^{-1} \begin{bmatrix} 0 \\ 0 \\ \dot{\psi} \end{bmatrix} \quad (I.4)$$

En effet, la rotation en roulis a lieu lorsque les repères sont encore confondus. Puis en ce qui concerne le tangage, le vecteur représentant la rotation doit être exprimé dans le repère fixe, il est donc multiplié par le terme $Rot(x, \varphi)^{-1}$. De même, le vecteur représentant la rotation en lacet doit être exprimé dans le repère fixe qui a déjà subi deux rotations.

On arrive ainsi à [19] :

$$\Omega = \begin{bmatrix} \Omega_1 \\ \Omega_2 \\ \Omega_3 \end{bmatrix} = \begin{bmatrix} \dot{\phi} \\ 0 \\ 0 \end{bmatrix} + \begin{bmatrix} 0 \\ \dot{\theta}c\varphi \\ -\dot{\theta}s\varphi \end{bmatrix} + \begin{bmatrix} -\dot{\psi}s\theta \\ \dot{\psi}s\varphi c\theta \\ \dot{\psi}c\varphi c\theta \end{bmatrix} = \begin{bmatrix} \dot{\phi} - \dot{\psi}s\theta \\ \dot{\theta}c\varphi + \dot{\psi}s\varphi c\theta \\ \dot{\psi}c\varphi c\theta - \dot{\theta}s\varphi \end{bmatrix} \quad (I.5)$$

$$\Omega = \begin{pmatrix} 1 & 0 & -s\theta \\ 0 & c\varphi & s\varphi c\theta \\ 0 & -s\varphi & c\varphi c\theta \end{pmatrix} \times \begin{bmatrix} \dot{\phi} \\ \dot{\theta} \\ \dot{\psi} \end{bmatrix} \quad (I.6)$$

Dans le cas où le quadrirotor réalise des mouvements angulaires de faibles amplitudes :

$$\begin{cases} c\varphi = c\theta = c\psi = 1 \\ s\varphi = s\theta = s\psi = 0 \end{cases}$$

La vitesse angulaire s'écrit donc comme suit :

$$\Omega = \begin{bmatrix} \dot{\phi} \\ \dot{\theta} \\ \dot{\psi} \end{bmatrix} \quad (I.7)$$

I.4.3. Vitesse linéaire

Les vitesses linéaires V_x^T, V_y^T, V_z^T dans le repère fixe sont en fonction des vitesses linéaires V_x^m, V_y^m, V_z^m dans le repère mobile tel que :

$$V = \begin{bmatrix} V_x^T \\ V_y^T \\ V_z^T \end{bmatrix} = R \times \begin{bmatrix} V_x^m \\ V_y^m \\ V_z^m \end{bmatrix} \quad (\text{I.8})$$

R : la matrice de rotation définie précédemment.

I.4.4. Effets physiques agissant sur le quadrirotor

I.4.4.1. Forces

Les forces qui agissent sur le système sont :

- **Le poids** du quadrirotor : est donné par :

$$P = m.g \quad (\text{I.9})$$

m: masse totale.

g : l'accélération de pesanteur ($g=9,81 \text{ m/s}^2$).

- **Les forces de poussée (portance)** : qui sont les forces causées par rotation des moteurs, elles sont perpendiculaires sur le plan des hélices. Ces forces sont proportionnelles au carrée de la vitesse de rotation des moteurs.

$$F_i = b\omega_i^2 \quad (\text{I.10})$$

Avec : $i = \overline{1:4}$

b : coefficient de portance, il dépend de la forme, de nombre de pales et de la densité de l'aire.

- **Les forces de trainée (résistance)** : la force de trainée est le couplage entre une force de pression et la force de frottements visqueux, dans ce cas on a deux forces agissant sur le système :

-la trainée des hélices : elle agit sur les pales, elle est proportionnelle à la densité de l'aire, à la forme des pales et au carré de la vitesse de rotation de l'hélice, elle est donnée par la relation suivante :

$$Th_i = d\omega_i^2 \quad (\text{I.11})$$

Avec : $i = \overline{1:4}$

d : coefficient de drag (trainée), il dépend de la fabrication de l'hélice.

- ✓ la traînée selon les axes (x, y, z) : elle est due aux mouvements du corps du quadrirotor :

$$F_t = k_{ft} v = \begin{pmatrix} -k_{ftx} & 0 & 0 \\ 0 & -k_{fty} & 0 \\ 0 & 0 & -k_{ftz} \end{pmatrix} \quad (\text{I.12})$$

Avec :

k_{ft} : Le coefficient de traînée de translation.

v : la vitesse linéaire.

I.4.4.2. Moments

Il existe en effet plusieurs moments qui agissent sur le quadrirotor, ces moments causés par les différentes forces de poussée, de traînée et par les effets gyroscopiques.

- **Moments dus aux forces de poussée des hélices :**

La différence entre les forces de portance des rotors 2 et 4 engendre une rotation autour de l'axe x, ce moment est donné par la relation suivante :

$$M_x = l(f_4 - f_2) = lb(\omega_4^2 - \omega_2^2) \quad (\text{I.13})$$

Avec :

l : la longueur des bras entre le rotor et le centre de masse du quadrirotor.

Lors d'une rotation autour de l'axe y, qui est due à la différence entre les forces de portance des rotors 1 et 3, nous obtenons un moment qui est donné par :

$$M_y = l(f_3 - f_1) = lb(\omega_3^2 - \omega_1^2) \quad (\text{I.14})$$

- **Moments dus aux forces de traînée des hélices :**

La rotation autour de l'axe z est due à un couple réactif provoqué par les couples de traînée dans chaque hélice. Ce moment est donné par :

$$M_z = d(\omega_1^2 - \omega_2^2 + \omega_3^2 - \omega_4^2) \quad (\text{I.15})$$

- **Moments résultants des frottements aérodynamiques :**

$$M_a = K_{fa} \Omega^2 = \begin{bmatrix} K_{fax} \dot{\phi}^2 \\ K_{fay} \dot{\theta}^2 \\ K_{faz} \dot{\psi}^2 \end{bmatrix} \quad (\text{I.16})$$

Avec :

K_{fa} : Coefficient des frottements aérodynamiques.

Ω : La vitesse angulaire.

I.4.4.3. Effets gyroscopiques

L'effet gyroscopique se définit comme la difficulté de modifier la position ou l'orientation du plan de rotation d'une masse tournante, l'effet gyroscopique est ainsi nommé en référence au mode de fonctionnement du gyroscope, appareil de contrôle de mouvement utilisé dans l'aviation (du grec gyro: rotation et scop : observation)

Un gyroscope est donc un appareil qui, fixé à un mobile ou à tout autres objets, permet d'effectuer une mesure de rotation absolue de ce dernier autour d'un axe, par conséquence, trois gyroscopes sont nécessaires pour déterminer les trois rotations selon les trois axes orthogonaux de l'espace. Le fonctionnement de ces capteurs repose sur des effets mécaniques, piézoélectriques ou encore optiques.

Dans notre cas, il y a deux mouvements gyroscopiques, le premier est le mouvement des hélices, le deuxième est celui due aux mouvements du quadrirotor.

- **Moment gyroscopique des hélices** : il est donné par [20]:

$$M_{gh} = \sum_1^4 \Omega \wedge J_r [0 \quad 0 \quad (-1)^{i+1} \omega_i]^T = \begin{bmatrix} J_r \dot{\theta} \bar{\Omega}_r \\ -J_r \dot{\phi} \bar{\Omega}_r \\ 0 \end{bmatrix} \tag{I.17}$$

Avec :

J_r : L'inertie des rotors.

$\bar{\Omega}_r = (\omega_1 - \omega_2 + \omega_3 - \omega_4)$.

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

$$M_{gm} = \Omega \wedge J \Omega \tag{I.18}$$

J est l'inertie du système.

I.4.5. Modèle mathématique du quadrirotor selon EULER-NEWTON

Après avoir décrit les différentes équations, on peut élaborer le modèle mathématique en utilisant le formalisme de NEWTON-EULER, les équations sont écrites sous la forme suivante [19], [20], [21], [22]:

$$\begin{cases} \dot{\zeta} = v \\ m\ddot{\zeta} = F_f + F_t + F_g \\ \dot{R} = RS(\Omega) \\ J\dot{\Omega} = -\Omega \wedge J\Omega + M_f - M_a - M_{gh} \end{cases} \tag{I.19}$$

Avec :

ζ : est le vecteur de position du quadrirotor $\zeta = [x \ y \ z]^T$.

m : la masse totale du quadrirotor.

J : matrice d'inertie symétrique de dimension (3x3), elle est donnée par :

$$J = \begin{pmatrix} I_m & 0 & 0 \\ 0 & I_y & 0 \\ 0 & 0 & I_z \end{pmatrix} \quad (\text{I.19.a})$$

Ω : La vitesse angulaire exprimée dans le repère fixe.

R : La matrice de rotation.

$S(\Omega)$: la matrice antisymétrique pour un vecteur de vitesse $\Omega = [\Omega_1 \ \Omega_2 \ \Omega_3]^T$

.elle est donnée par :

$$S(\Omega) = \begin{pmatrix} 0 & -\Omega_3 & \Omega_2 \\ \Omega_3 & 0 & -\Omega_1 \\ -\Omega_2 & \Omega_1 & 0 \end{pmatrix} \quad (\text{I.19.b})$$

F_f : La force totale générée par le quadrirotor, elle est donnée par :

$$F_f = R \times \left[0 \ 0 \ \sum_{i=1}^4 F_i \right]^T \quad (\text{I.19.c})$$

$$F_i = b\omega_i^2 \quad (\text{I.19.d})$$

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

$$F_t = \begin{pmatrix} -K_{f_{tx}} & 0 & 0 \\ 0 & -K_{f_{ty}} & 0 \\ 0 & 0 & -K_{f_{tz}} \end{pmatrix} \dot{\zeta} \quad (\text{I.19.e})$$

$K_{f_{tx}}, K_{f_{ty}}, K_{f_{tz}}$: Les coefficients de traînée de translation.

F_g : Force de gravité.

$$F_g = \begin{bmatrix} 0 \\ 0 \\ -mg \end{bmatrix} \quad (\text{I.19.f})$$

M_f : Moments provoqué par les forces de poussée et de traînée. Il est donné par:

$$M_f = \begin{bmatrix} l(F_4 - F_2) \\ l(F_3 - F_1) \\ d(\omega_1^2 - \omega_2^2 + \omega_3^2 - \omega_4^2) \end{bmatrix} \quad (\text{I.19.g})$$

M_a : Moment résultant des frottements aérodynamiques. Il est donné par :

$$M_a = \begin{pmatrix} K_{fax} & 0 & 0 \\ 0 & K_{fay} & 0 \\ 0 & 0 & K_{faz} \end{pmatrix} \begin{bmatrix} \Omega_1^2 \\ \Omega_2^2 \\ \Omega_3^2 \end{bmatrix} = \begin{bmatrix} K_{fax} \cdot \dot{\phi} \\ K_{fay} \cdot \dot{\theta} \\ K_{faz} \cdot \dot{\psi} \end{bmatrix} \quad (\text{I.19.h})$$

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

I.4.5.1. Equation de mouvement de translation

On a : $m\ddot{\xi} = F_f + F_t + F_g$. (I.20)

En remplaçant chaque force par sa formule, on trouve :

$$m \begin{bmatrix} \ddot{x} \\ \ddot{y} \\ \ddot{z} \end{bmatrix} = \begin{bmatrix} c\varphi c\psi s\theta + s\varphi s\psi \\ c\varphi s\theta s\psi - s\varphi c\psi \\ c\varphi c\theta \end{bmatrix} \sum_{i=1}^4 F_i - \begin{bmatrix} k_{f_{ix}} \dot{x} \\ k_{f_{iy}} \dot{y} \\ k_{f_{iz}} \dot{z} \end{bmatrix} - \begin{bmatrix} 0 \\ 0 \\ mg \end{bmatrix} \quad (\text{I.21})$$

On obtient alors les équations différentielles qui définissent le mouvement de translation, il est donné comme suit:

$$\begin{cases} \ddot{x} = \frac{1}{m} (c\varphi c\psi s\theta + s\varphi s\psi) \left(\sum_{i=1}^4 F_i \right) - \frac{k_{f_{ix}}}{m} \dot{x} \\ \ddot{y} = \frac{1}{m} (c\varphi s\theta s\psi - s\varphi c\psi) \left(\sum_{i=1}^4 F_i \right) - \frac{k_{f_{iy}}}{m} \dot{y} \\ \ddot{z} = \frac{1}{m} (c\varphi c\theta) \left(\sum_{i=1}^4 F_i \right) - \frac{k_{f_{iz}}}{m} \dot{z} - g \end{cases} \quad (\text{I.22})$$

I.4.5.2. Equation de mouvement de rotation

On a : $J\dot{\Omega} = -M_{gh} - M_{gm} - M_a + M_f$ (I.23)

En remplaçant chaque moment par sa formule correspondante, on obtient alors:

$$\begin{pmatrix} I_x & 0 & 0 \\ 0 & I_y & 0 \\ 0 & 0 & I_z \end{pmatrix} \begin{pmatrix} \ddot{\phi} \\ \ddot{\theta} \\ \ddot{\psi} \end{pmatrix} = \begin{bmatrix} J_r \overline{\Omega_r} \dot{\theta} \\ -J_r \overline{\Omega_r} \dot{\phi} \\ 0 \end{bmatrix} - \begin{bmatrix} \dot{\phi} \\ \dot{\theta} \\ \dot{\psi} \end{bmatrix} \wedge \left(\begin{pmatrix} I_x & 0 & 0 \\ 0 & I_y & 0 \\ 0 & 0 & I_z \end{pmatrix} \begin{bmatrix} \dot{\phi} \\ \dot{\theta} \\ \dot{\psi} \end{bmatrix} \right) - \begin{bmatrix} k_{fax} \dot{\phi}^2 \\ k_{fay} \dot{\theta}^2 \\ k_{faz} \dot{\psi}^2 \end{bmatrix} + \begin{bmatrix} lb(\omega_4^2 - \omega_2^2) \\ lb(\omega_3^2 - \omega_1^2) \\ d(\omega_1^2 - \omega_2^2 + \omega_3^2 - \omega_4^2) \end{bmatrix} \quad (\text{I.24})$$

On obtient alors les équations différentielles qui définissent le mouvement de rotation comme suit:

$$\begin{cases} I_x \ddot{\phi} = -\dot{\theta} \dot{\psi} (I_z - I_y) - J_r \overline{\Omega_r} \dot{\theta} - k_{fax} \dot{\phi}^2 + lb(\omega_4^2 - \omega_2^2) \\ I_y \ddot{\theta} = \dot{\phi} \dot{\psi} (I_z - I_x) - J_r \overline{\Omega_r} \dot{\phi} - k_{fay} \dot{\theta}^2 + lb(\omega_3^2 - \omega_1^2) \\ I_z \ddot{\psi} = -\dot{\phi} \dot{\theta} (I_y - I_x) - k_{faz} \dot{\psi}^2 + d(\omega_1^2 - \omega_2^2 + \omega_3^2 - \omega_4^2) \end{cases} \quad (I.25)$$

Avec : $\overline{\Omega_r} = \omega_1 - \omega_2 + \omega_3 - \omega_4$

Par conséquent, le modèle dynamique complet est le suivant:

$$\begin{cases} \ddot{\phi} = \frac{(I_y - I_z)}{I_x} \dot{\theta} \dot{\psi} - \frac{J_r \overline{\Omega_r}}{I_x} \dot{\theta} - \frac{k_{fax}}{I_x} \dot{\phi}^2 + \frac{1}{I_x} u_2 \\ \ddot{\theta} = \frac{(I_z - I_x)}{I_y} \dot{\phi} \dot{\psi} - \frac{J_r \overline{\Omega_r}}{I_y} \dot{\phi} - \frac{k_{fay}}{I_y} \dot{\theta}^2 + \frac{1}{I_y} u_3 \\ \ddot{\psi} = \frac{(I_x - I_y)}{I_z} \dot{\theta} \dot{\phi} - \frac{k_{faz}}{I_z} \dot{\psi}^2 + \frac{1}{I_z} u_4 \\ \ddot{x} = \frac{-k_{fx}}{m} \dot{x} + \frac{1}{m} u_x u_1 \\ \ddot{y} = \frac{-k_{fy}}{m} \dot{y} + \frac{1}{m} u_y u_1 \\ \ddot{z} = \frac{-k_{fz}}{m} \dot{z} - g + \frac{\cos(\varphi) \cos(\theta)}{m} u_1 \end{cases} \quad (I.26)$$

Avec :
$$\begin{cases} u_x = (c\varphi c\psi s\theta + s\varphi s\psi) \\ u_y = (c\varphi s\theta s\psi + s\varphi c\psi) \end{cases} \quad (I.26.a)$$

Et :
$$\begin{bmatrix} u_1 \\ u_2 \\ u_3 \\ u_4 \end{bmatrix} = \begin{bmatrix} b & b & b & b \\ 0 & -lb & 0 & lb \\ -lb & 0 & lb & 0 \\ d & -d & d & -d \end{bmatrix} \begin{bmatrix} \omega_1^2 \\ \omega_2^2 \\ \omega_3^2 \\ \omega_4^2 \end{bmatrix} \quad (I.26.b)$$

I.4.6. La représentation d'état du système

Pour écrire le modèle mathématique sous forme d'un modèle d'état on pose :

$$\begin{aligned} X &= [x_1 \ x_2 \ x_3 \ x_4 \ x_5 \ x_6 \ x_7 \ x_8 \ x_9 \ x_{10} \ x_{11} \ x_{12}]^T \\ &= [\varphi \ \dot{\phi} \ \theta \ \dot{\theta} \ \psi \ \dot{\psi} \ x \ \dot{x} \ y \ \dot{y} \ z \ \dot{z}]^T \end{aligned} \quad (I.27)$$

Et on obtient la représentation suivante:

$$\begin{cases} \dot{x}_1 = x_2 \\ \dot{x}_2 = a_1 x_4 x_6 + a_2 x_2^2 + a_3 \bar{\Omega} x_4 + b_1 u_2 \\ \dot{x}_3 = x_4 \\ \dot{x}_4 = a_4 x_2 x_6 + a_5 x_4^2 + a_6 \bar{\Omega} x_2 + b_2 u_3 \\ \dot{x}_5 = x_6 \\ \dot{x}_6 = a_7 x_2 x_4 + a_8 x_6^2 + b_3 u_4 \\ \dot{x}_7 = x_8 \\ \dot{x}_8 = a_9 x_8 + u_x \frac{u_1}{m} \\ \dot{x}_9 = x_{10} \\ \dot{x}_{10} = a_{10} x_{10} + u_y \frac{u_1}{m} \\ \dot{x}_{11} = x_{12} \\ \dot{x}_{12} = a_{11} x_{12} + \frac{c x_1 c x_3}{m} u_1 - g \end{cases} \quad (\text{I.27.a})$$

$$\text{Avec : } \begin{cases} a_1 = \left(\frac{I_y - I_z}{I_x} \right), a_2 = \frac{-k_{fax}}{I_x}, a_3 = \frac{-J_r}{I_x} \\ a_4 = \left(\frac{I_z - I_x}{I_y} \right), a_5 = \frac{-k_{fay}}{I_y}, a_6 = \frac{J_r}{I_y} \\ a_7 = \left(\frac{I_x - I_y}{I_z} \right), a_8 = \frac{-k_{faz}}{I_z}, a_9 = \frac{-k_{fzx}}{m} \\ a_{10} = \frac{-k_{fyy}}{m}, a_{11} = \frac{-k_{fzz}}{m} \\ b_1 = \frac{1}{I_x}, b_2 = \frac{1}{I_y}, b_3 = \frac{1}{I_z} \end{cases} \quad (\text{I.27.b})$$

$$\text{Et : } \begin{cases} u_x = c x_1 s x_3 c x_5 + s x_1 s x_5 \\ u_y = c x_1 s x_3 s x_5 + s x_1 c x_5 \end{cases} \quad (\text{I.27.c})$$

I.4.7. Dynamique des rotors

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 les équations dynamiques suivantes :

$$\begin{cases} V = r i + L \frac{di}{dt} + k_e \omega \\ k_m i = J_r \frac{d\omega}{dt} + C_s + k_r \omega^2 \end{cases} \quad (\text{I.28})$$

Les différents paramètres du moteur sont définis comme suit:

V : est la tension d'entrée du moteur.

ω_i : est la vitesse angulaire du rotor.

k_e, k_m : Représentent respectivement les constantes des couples électrique et mécanique.

k_r : La constante du couple de charge.

r, L : Sont la résistance du moteur et l'inductance.

J_r : L'inertie du rotor.

C_s : Représente le frottement sec.

Sur ce ; le modèle choisi pour le rotor est :

$$\dot{\omega}_i = \beta V_i - \beta_0 - \beta_1 \omega_i - \beta_2 \omega_i^2 \quad , i \in [1 : 4] \quad (I.29)$$

Avec : $\beta_0 = \frac{C_s}{J_r}$, $\beta_1 = \frac{k_e k_m}{r J_r}$, $\beta_2 = \frac{k_r}{J_r}$ et $\beta = \frac{k_m}{r J_r}$.

Paramètres	Valeurs	Paramètres	Valeurs
m	1 [kg]	l	0.225 [m]
g	9.81 [$m.s^{-2}$]	k_{fax}	5.56×10^{-4} [N/rad.s ⁻¹]
b	5.42×10^{-5} [N/rad.s ⁻¹]	k_{fay}	5.56×10^{-4} [N/rad.s ⁻¹]
d	1.1×10^{-6} [N.m/rad.s ⁻¹]	k_{faz}	6.354×10^{-4} [N/rad.s ⁻¹]
k_{fxx}	5.56×10^{-4} [N/m.s ⁻¹]	J_r	1.04×10^{-4} [kg.m ²]
k_{fyy}	5.56×10^{-4} [N/m.s ⁻¹]	I_x	8.1×10^{-3} [kg.m ²]
k_{fzz}	6.354×10^{-4} [N/m.s ⁻¹]	I_y	8.1×10^{-3} [kg.m ²]
k_r	5.12×10^{-2}	I_z	14.2×10^{-3} [kg.m ²]
k_e	6.3×10^{-3}	k_m	6.3×10^{-3}
r	0.6 [Ohm]		

Tableau I.1 : Paramètres du modèle simulé.

I.5. Conclusion

En conclusion à ce chapitre, on constate que le monde des drones a connu ces dernières années un intérêt croissant dans la communauté scientifique vu les développements et les avancées technologique, ces derniers ont rendus possible la conception de systèmes performants dotés de réelle capacités de navigation autonome.

Tous les drones nécessitent une modélisation du système, l'utilisation du formalisme de Newton-Euler nous a permis d'établir le modèle dynamique, où nous avons constaté que le quadrirotor est un système sous actionné. De plus la complexité du modèle, la non linéarité, l'interaction entre les états du système peuvent se voir clairement.

Chapitre II:
Généralités sur la
commande

II.1. Introduction

La plupart des techniques de commande des systèmes sont basées sur la connaissance du processus à commander et de son environnement. Or, les phénomènes mis en cause sont généralement trop complexes pour que l'on puisse établir rigoureusement un modèle mathématique de leur comportement. En conséquent, la plupart des systèmes non linéaires sont caractérisés par des paramètres incertains et/ou variable en fonction du temps, ce complique leur commande en vue d'améliorer leurs performances.

L'absence totale de modèle ou celle d'un modèle précis du système, rend dans certains cas, l'application de l'automatique classique difficile, voire impossible pour la synthèse d'une loi de commande performante.

II.2. Commande Classique

➤ Commande PID

La régulation par contrôleur PID exploite trois actions principales, l'action proportionnelle, intégrale et dérivée. Ces actions ont chacune leurs avantages et inconvénients, en combinons ces trois actions nous permet d'avoir une certaine compensation des inconvénients tout en gardant les avantages afin d'aboutir à des performances satisfaisantes.

1. Action proportionnelle

L'effet de cette action consiste à amplifier l'erreur d'un gain constant dans le but d'augmenter la rapidité d'un procédé. Cette action est représentée comme suit:

$$C(t) = K_p \cdot e(t) \quad (\text{II.1})$$

Plus la valeur de K_p est grande plus la réponse est plus rapide mais au détriment de la stabilité du système.

2. Action intégrale

Cette action a pour objectif de réduire voire même éliminer l'erreur statique en régime permanent. pour ce faire, le régulateur intègre l'erreur par rapport au temps et multiplie le résultat par une constante K_i comme tel:

$$C(t) = K_i \cdot \int_0^t e(t) dt \quad (\text{II.2})$$

Plus la valeur de K_i est grande plus l'erreur statique n'est corrigée mais nous perdons un peu en stabilité avec un risque de dépassement qui subvient.

3. 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 :

$$C(t) = K_d \cdot \frac{de(t)}{dt} \quad (\text{II.3})$$

Plus la valeur de K_d est grande plus le dépassement est atténué mais au détriment de la rapidité du système avec un risque d'instabilité pour des grandes valeurs.

4. Action PID

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

$$C(t) = K_p \cdot e(t) + K_i \cdot \int_0^t e(t) dt + K_d \cdot \frac{de(t)}{dt} \quad (\text{II.4})$$

II.3. Commande Intelligente

II.3.1. Logique floue

La logique floue est basée sur la manière dont le cerveau traite l'information, elle permet d'étendre la notion de la logique classique, associée aux variables booléennes ne prenant que deux valeurs 0 et 1.

Il est alors possible d'associer à des variables des coefficients d'appartenance à des sous-ensembles flous prenant des valeurs dans l'intervalle $[0, 1]$ et quantifiant l'incertitude sur la variable contrairement à la logique TOR (tout ou rien). Dans la logique floue un événement certain pour la variable se traduira par un degré d'appartenance au sous-ensemble flou égale à 1 alors que la valeur sera inférieure à 1 en présence d'incertitude alors qu'une

valeur nulle pour un degré d'appartenance indique que la possibilité d'appartenance au sous-ensemble sélectionné est complètement rejeté [23].

II.3.1.1. Sous-ensemble flou

La notion d'ensemble flou à été proposée par ZADEH en introduisant un caractère graduel de l'appartenance d'un élément à un ensemble donné. Dans la théorie classique des ensembles, un ensemble A de B est défini par une fonction d'appartenance $u_A(x)$ qui caractérise tout élément x appartenant à B. cette fonction prend la valeur 1 si x appartient à A et la valeur 0 dans le cas contraire [24]:

$$u_A(x) = \begin{cases} 1 & \text{si } x \in A \\ 0 & \text{si } x \notin A \end{cases} \quad (\text{II.5})$$

En logique floue, un sous-ensemble A de B est défini par une fonction d'appartenance $u_A(x)$ pouvant prendre différentes valeurs comprises entre 0 et 1, selon le degré d'appartenance de l'élément x au sous-ensemble A.

$$u_A(x) \in [0 \ 1]$$

II.3.1.2 Univers de discours

Il représente l'ensemble de référence ou le domaine de variation de la variable linguistique, ou le domaine de fonctionnement du processus dans le cas de réglage, comme illustré sur la figure (II.1) dont plusieurs sous-ensembles sont décrits par des fonctions d'appartenances (FA) triangulaires et trapézoïdales.

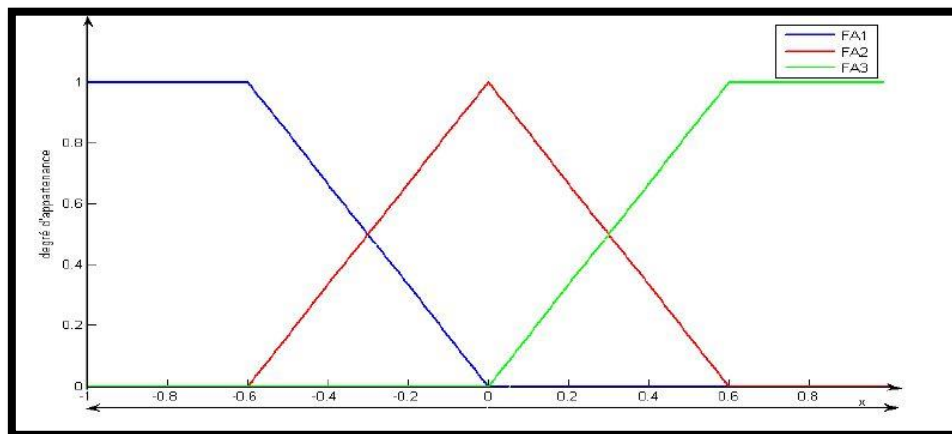


Figure II.1: Univers de discours.

II.3.1.3. Variable linguistique

Une variable linguistique est une variable dont les valeurs sont des mots ou des phrases exprimées dans une langue naturelle ou dans un langage artificiel [25]. Elle est représentée par un triplet (v, x, T_v) où v est la variable du problème, x l'univers de discours et $T_v = \{A_1, A_2, A_3, \dots, A_n, \dots\}$ un ensemble fini ou non de sous-ensemble flou de x , qui caractérise v , par exemple : $\{T, [0^0, 100^0], ("basse", "normal", "élevée")\}$ est une variable linguistique figure(II.2).les valeurs de T "basse", "normal" et "élevé" sont appelées valeurs linguistiques ou termes linguistiques ou encore termes flous [26].

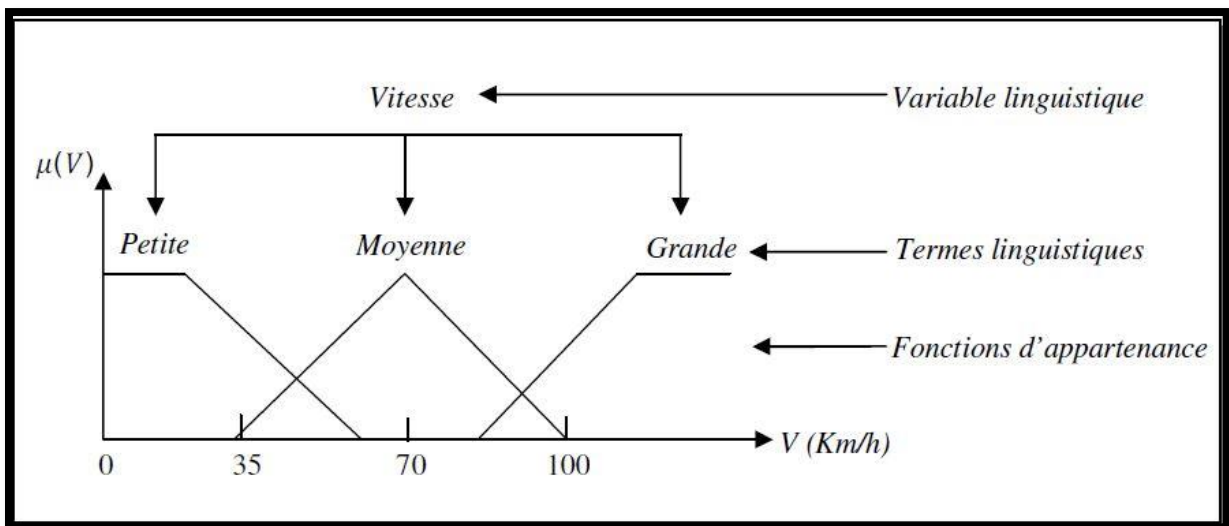


Figure II.2: Variable linguistique

II.3.1.4. Fonctions d'appartenances

On représente les variables linguistiques par leurs fonctions d'appartenances donc à chaque sous-ensemble floue A est associé une fonction d'appartenance $u_A(x)$ où x est la variable linguistique. tel que, à chaque point x est associée une valeur précise $u_A(x)$, qui désigne le degré d'appartenance de x à A [23][27].

Le plus souvent, la fonction d'appartenance est déterminée par l'une des fonctions suivantes (figure II.3).

- Fonction triangulaire: elle est définie par trois paramètres a, b, c qui déterminent les coordonnées des trois sommets.

$$u_A(x) = \max\left(\min\left(\frac{x-a}{b-a}, \frac{c-x}{c-b}\right), 0\right) \tag{II.6}$$

- Fonction trapézoïdale: elle est définie par quatre paramètres a,b,c,d.

$$u_A(x) = \max\left(\min\left(\frac{x-a}{b-a}, 1, \frac{c-x}{d-c}\right), 0\right) \tag{II.7}$$

- Fonction gaussienne: elle est définie par deux paramètres δ et m.

$$u_A(x) = \exp\left\{-\frac{(x-m)^2}{2\delta^2}\right\} \tag{II.8}$$

- Fonction sigmoïde: une fonction sigmoïde est définie par deux paramètres a et c.

$$u_A(x) = \left\{ \frac{1}{1 + \exp(-a(x-c))} \right\} \tag{II.9}$$

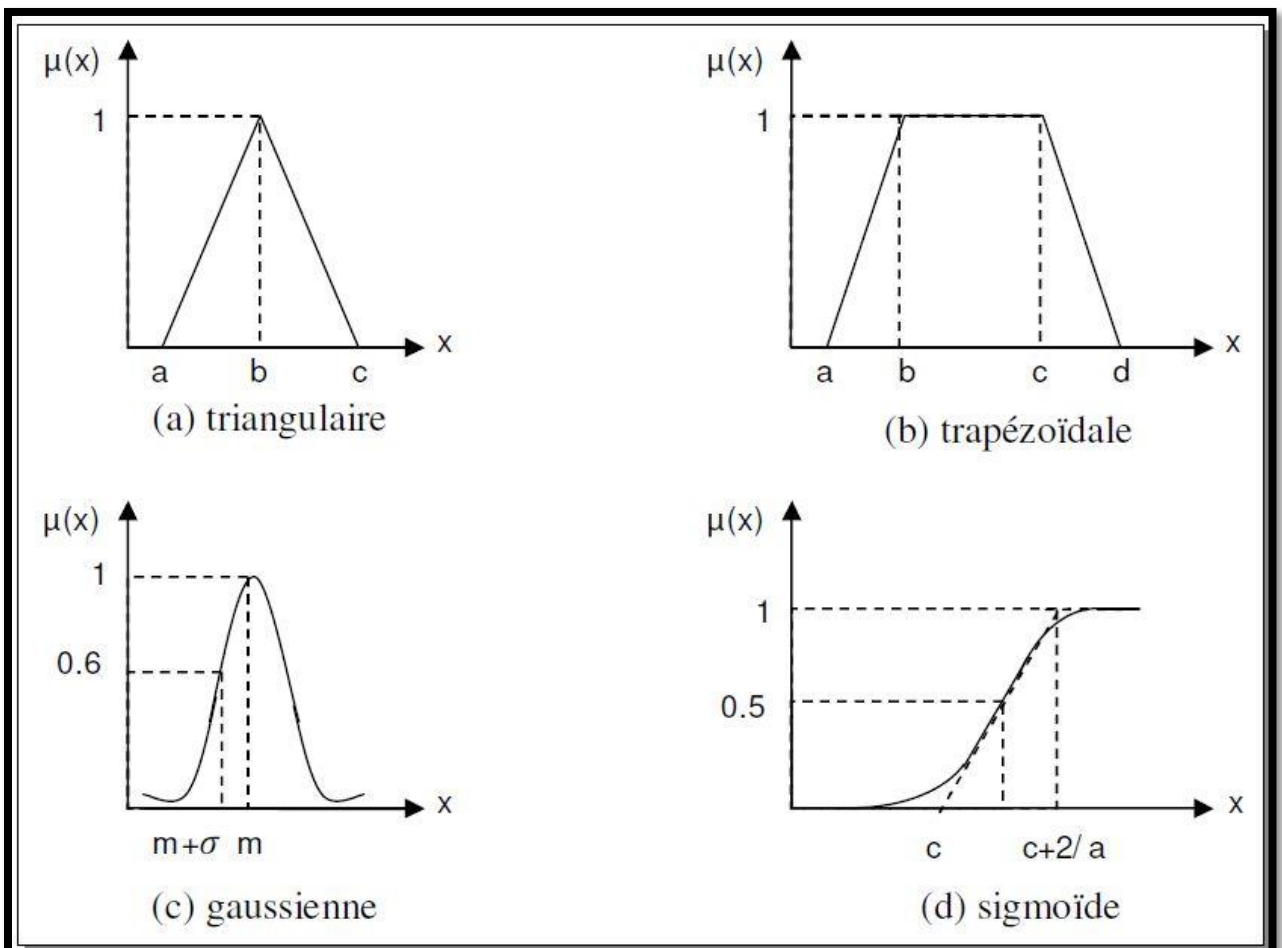


Figure II.3: Fonction d'appartenance.

II.3.1.5. Règles floues

On définit une règle floue comme une proposition floue correspondante à la mise en relation de deux propositions floues par une implication [28].

Une proposition floue est dite élémentaire, si elle n'est constitué que d'un prédicat de la forme "X est A". La composition de deux ou plusieurs variables linguistiques constitue une proposition floue. Elle se présente sous la forme:

- Si condition1 et/ou condition2 et/ou.....alors actions sorties.

II.3.2. Commande floue

La commande floue est une application de la logique floue au contrôle des systèmes dynamiques pour lesquels on ne possède pas de modèle satisfaisant. son principe est simple; il s'agit dans la plus par des cas d'imiter le comportement d'un operateur humain dans le régulateur d'un processus complexe, à l'aide des règles floues [29].

II.3.2.1. Structure générale d'un système floue

De manière classique, le fonctionnement interne d'un système flou repose sur la structure présentée par la figure suivante [30]:

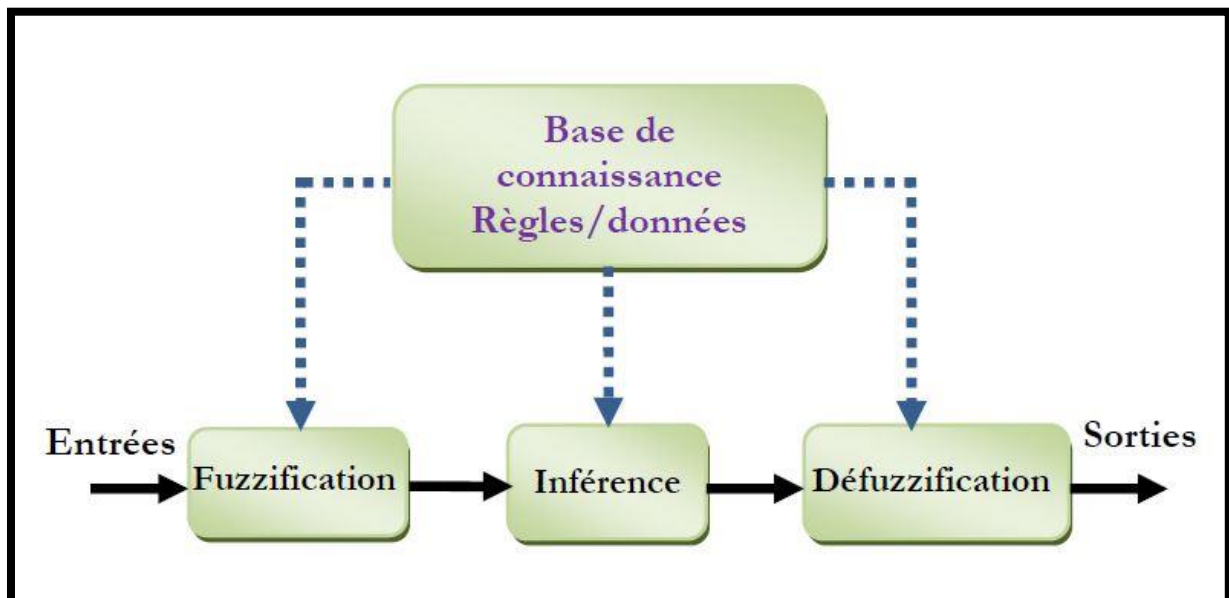


Figure II.4: structure générale d'un système flou.

La configuration de base d'un contrôleur flou comprend quatre parties :

- Interface de fuzzification
- Base de connaissance
- Moteur d'inférence
- Interface de défuzzification

II.3.2.1.1. Interface de fuzzification

La fuzzification de la valeur précise d'une variable consiste à caractériser le degré avec lequel cette mesure appartient à un sous-ensemble flou donné, c'est-à-dire le passage d'une grandeur précise à une variable floue.

Le choix des formes des fonctions d'appartenances est arbitraire. La forme la plus fréquemment utilisée est la forme triangulaire. Le nombre de fonctions d'appartenances est généralement impaire et se répartissent autour de zéro. Le choix du nombre dépend de la précision souhaitée. Elles peuvent être symétriques ou non symétriques et équidistantes ou non équidistantes.

II.3.2.1.2. base de connaissance

La conception d'une base de connaissance représente la phase importante dans la conception des systèmes experts. Elle comprend la base de données et la base des règles floues.

- La base de données: contient la définition des ensembles flous, les facteurs d'échelle pour la normalisation des ensembles de référence et la répartition de l'espace flou d'entrée et de sortie.
- La base des règles floues: elle rassemble l'ensemble des règles floues de type "si-alors" décrivant en termes linguistiques basés sur la connaissance d'un expert le comportement dynamique de système.

II.3.2.1.3. Moteur d'inférence floue

C'est un mécanisme de décision. Il permet la réalisation des différents opérateurs ET, OU,... intervenant dans les règles d'inférence et s'appliquant aux fonctions d'appartenances.

A cause de l'empiètement des fonctions d'appartenances, en générale deux ou plusieurs règles sont activées en même temps. Ce fait doit être pris en considération lors de la

réalisation de l'opérateur OU. Il existe plusieurs possibilités pour réaliser ces opérateurs qui s'appliquent aux fonctions d'appartenances. On introduit alors la notion de la méthode d'inférence, elle détermine la réalisation des différents opérateurs dans une inférence, permettant ainsi un traitement numérique de cette dernière [30].

Il existe plusieurs méthodes pour réaliser ces opérateurs à savoir :

- **Méthode d'inférence MAX-MIN (mamdani)**

Cette méthode réalise, au niveau de la condition, l'opérateur OU par la formation maximum et l'opérateur ET par la formation du minimum. La conclusion dans chaque règle, introduite par alors, lie le facteur d'appartenance de la variable de sortie par l'opérateur ET, réalisé dans le cas présent par la formation du minimum. Enfin, l'opérateur OU qui lie les différentes règles est réalisé par la formation du maximum [31].

C'est la méthode la plus universelle mais elle a un temps de calcul très long.

- **Méthode d'inférence MAX-PROD (larsen)**

Cette méthode réalise en générale, au niveau de la condition, l'opérateur OU par la formation maximum et l'opérateur ET par la formation minimum. Par contre, la conclusion dans chaque règle, introduite par alors, lie le facteur d'appartenance de la condition avec la fonction d'appartenance de la variable de sortie par l'opérateur ET, est réalisé cette fois-ci par la formation du produit. Enfin l'opérateur OU qui lie les différentes règles est réalisé par la formation du maximum [31].

- **Méthode d'inférence somme-produit (sugeno)**

Par opposition aux méthodes d'inférences précédentes, cette méthode réalise, au niveau de la condition, l'opérateur OU par la formation de la somme, plus précisément par la valeur moyenne, tandis que l'opérateur ET est réalisé par la formation du produit. La conclusion de chaque règle, introduite par alors, liant le facteur d'appartenance de la condition avec la fonction d'appartenance de la variable de sortie par l'opérateur ET, est réalisé cette fois-ci par la formation du produit. Enfin l'opérateur OU qui lie les différentes règles est réalisé par la formation de la somme, donc de la valeur moyenne [31].

II.3.2.1.4. Interface de défuzzification

C'est une interface "linguistique-numérique" qui transforme la partie floue de l'inférence en valeurs numériques directement exploitables par le processus. Les méthodes d'inférences fournissent une fonction d'appartenance résultante, $u_{Res}(x_R)$ de la variable de sortie x_R , il s'agit donc d'une information floue. Par cette étape, se fait alors le retour aux grandeurs de sortie réelles. Il s'agit à cet effet, de calculer à partir des degrés d'appartenances à tous les ensembles flous de la variable de sortie, l'abscisse qui correspond à la valeur de cette sortie. Plusieurs stratégies sont utilisées pour réaliser l'étape de défuzzification, les plus répandues sont [30]:

- La méthode du centre de gravité: la sortie defuzzifiée x^* correspond à l'abscisse du centre de gravité de la surface de la fonction d'appartenance résultante $u_{Res}(x_R)$ (figure...).mathématiquement, cela se traduit par:

$$x^* = \frac{\sum_{i=1}^n u(x_i) \cdot x_i}{\sum_{i=1}^n u(x_i)} \quad (\text{II.10})$$

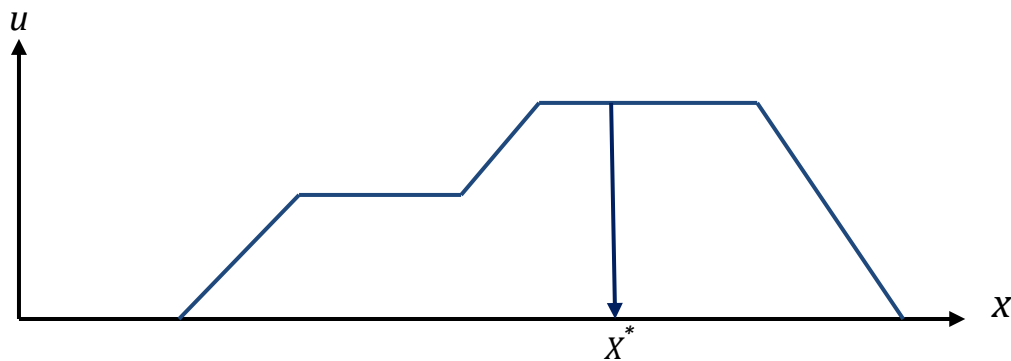


Figure II.5: défuzzification par centre de gravité

- La méthode de la moyenne des maximas: elle a été introduite en raison de la lourdeur des calculs de la méthode du centre de gravité. La sortie défuzzifiée est obtenue par le calcul de la moyenne de abscisses pour lesquelles la fonction d'appartenance est maximale.

$$x^* = \frac{\sum_{i=1}^n u(x_i)}{n} \quad (\text{II.11})$$

Où les $u(x_i)$ sont définis par : $u(x_i) = \max(u(x))$.

- Méthode du maximum: cette méthode consiste à choisir comme valeur de sortie celle correspondante à l'abscisse du maximum de la fonction d'appartenance. Cette méthode s'applique uniquement dans le cas où la fonction d'appartenance associée à l'ensemble de sortie n'admet qu'un seul maximum.

$$x^* = \max(u_A(x)).$$

- Méthode de la somme pondérée: cette méthode est un compromis entre la méthode du maximum et la méthode du centre de gravité. On calcule individuellement les sorties relatives à chaque règle selon le principe de la moyenne de maximas. Puis on réalise la moyenne pondérée de toutes ces sorties relatives.

II.3.3. Réseaux de neurones

Les réseaux de neurones artificiels (RNA) sont des réseaux fortement connectés de processeurs élémentaires fonctionnant en parallèle. Chaque processeur élémentaire calcule une sortie unique sur la base de l'information qu'il reçoit. Ils infèrent des propriétés émergentes permettant de solutionner les problèmes jadis qualifiés et complexes.

Dans un réseau, chaque sous-groupe fait un traitement indépendant des autres et transmet le résultat de son analyse au sous-groupe suivant. L'information donnée au réseau va donc se propager de la couche d'entrée à la couche de sortie en passant soit par aucune, une ou plusieurs couches intermédiaires. Il est à noter qu'en fonction de l'algorithme d'apprentissage, il est possible d'avoir une propagation de l'information à reculons [32].

Dans cette partie nous donnons un petit aperçu de ces outils.

II.3.3.1. Structure des neurones artificiels

II.3.3.1.1. Du neurone biologique au neurone formel (artificiel)

La première conceptualisation d'un neurone artificiel revient à MC CULLOCH et W.PITTS (1943). Le principe d'apprentissage en lui-même a été proposé par HEBB(1949). Quelques années plus tard, des modèles biologiques des neurones ont été complètement formalisés. Deux approches ont été développées en même temps, sans lien entre elles. Ces travaux ont conduit à deux modèles de neurones artificiels, deux neurones formels, chacun

associé à son principe d'apprentissage: le perceptron et l'Adaline (Adaptive Linear Neuron) [33].

Le perceptron a été formalisé par ROSENBLATT(1958,1962) pour sa structure et reprend le principe d'apprentissage proposé par HEBB(1949). Le principe de ce modèle de neurone repose sur un postulat biologique. Il cherche essentiellement à reproduire de la meilleure manière possible les observations biologiques.

L'Adaline a été proposée par WIDROW et HOFF(1960). Ce modèle de neurone découle d'une approche mathématique, son fonctionnement est rigoureux, son application est des plus directes et des plus pratiques (WIDROW et LEHR, 1990;WIDROW et WALLACHE,1996).l'efficacité de son apprentissage a été prouvée (WIDROW et KAMENEKSKY,2003).

II.3.3.1.2 Le neurone biologique

Le cerveau humain est un système très complexe capable de penser, de se souvenir et de résoudre des problèmes. Il y a eu de nombreuses tentatives pour émuler les fonctions cérébrales avec un modèle informatique et bien qu'il ait eu quelques réalisations plutôt spectaculaires provenant de ces efforts, tous les modèles développés à ce jour pâlissent dans l'oubli par rapport au fonctionnement complexe du cerveau humain [34].

Un neurone est l'unité cellulaire fondamentale du système nerveux du cerveau. C'est un élément de traitement simple qui reçoit et combine des signaux d'autres neurones à travers des chemins d'entrée appelés dendrites. Si les signaux d'entrée combinés sont assez forts, le neurone se déclenche produisant un signal de sortie le long de l'axone qui est connecté aux dendrites de nombreux autres neurones, la figure (...) est un croquis d'un neurone montrant ses différents composants. Chaque signal entrant dans un neurone le long d'une dendrite passe par une synapse ou une jonction synaptique. Cette jonction est un espace infinitésimal dans la dendrite qui est rempli de fluide neurotransmetteur qui accélère ou retarde l'écoulement des charges électriques. Les actions fondamentales du neurone chimique dans la nature, et ce fluide neurotransmetteur produit des signaux électriques qui vont au noyau ou au soma du neurone. L'ajustement de l'impédance ou de la conductance de l'espace synaptique est un processus extrêmement important. En effet, ces ajustements mènent à la mémoire et à l'apprentissage. Comme les forces synaptiques des neurones sont ajustées, le cerveau apprend et stocke l'information.

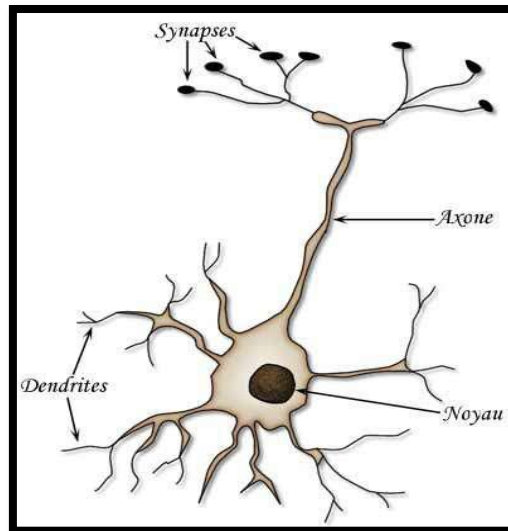


Figure II.6: le neurone biologique

II.3.3.1.3. Neurone formel (artificiel)

Dans la littérature un neurone formel est un automate très simple imitant grossièrement la structure et le fonctionnement d'un neurone biologique. Il représente une fonction algébrique non linéaire et bornée, dont la valeur dépend des paramètres appelés coefficients ou poids. Les variables de cette fonction sont habituellement appelées "entrées" du neurone et la valeur de la fonction est appelée sa sortie. Un neurone est avant tout un opérateur mathématique numérique qui réalise deux opérations. La première est la somme pondérée de ses entrées par les poids synaptiques; cette somme est appelée potentiel neuronal. La seconde génère la sortie du neurone, image par une fonction f appelée généralement fonction d'activation (figure II.7).

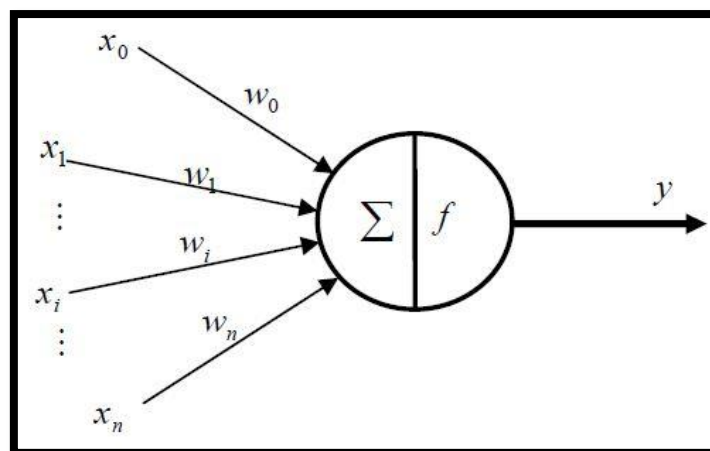


Figure II.7: neurone formel

Avec:

$z = \sum_{i=0}^n w_i x_i$ Est la somme pondérée.

$w_i, i = 0, 1, \dots, n$ S'appellent les poids synaptiques.

x_i Représente les entrées du neurone.

Y: est la sortie du neurone après son activation ($y = f(\sum_{i=0}^n w_i x_i)$).

F: est une fonction non linéaire habituellement appelée fonction d'activation. La figure (.....) montre les fonctions d'activations les plus utilisées.

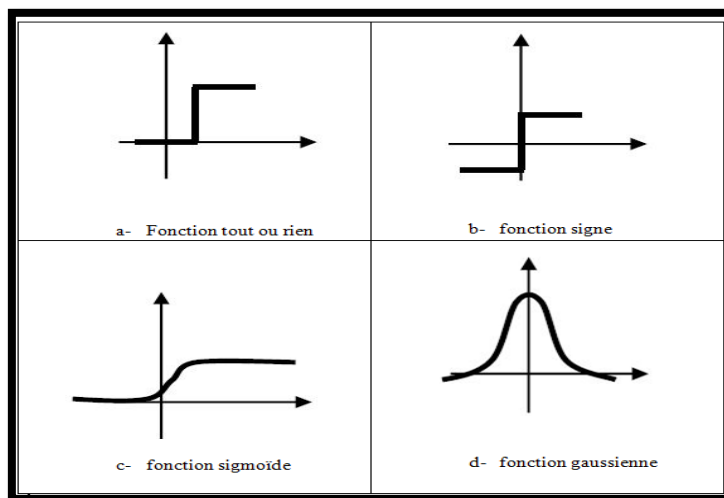


Figure II.8: les fonctions d'activation les plus utilisées

II.3.3.2. Les réseaux de neurones artificiels

Un réseau de neurone artificiel peut être défini comme un système de traitement de données composé d'un grand nombre d'éléments (neurones artificiels) de traitement simple et hautement interconnectés dans une architecture inspirée de la structure du cortex cérébrale du cerveau. Il est constitué de plusieurs unités (neurones) organisées sous forme de niveaux différents, appelés couches du réseau. Les neurones appartenant à la même couche possèdent les mêmes caractéristiques et utilisent le même type de fonction d'activation. Entre deux couches voisines les connections se font par l'intermédiaire des poids qui jouent le rôle des synapses. L'information est portée par la valeur de ses poids, tandis que la structure du réseau de neurones ne sert qu'à traiter l'information et l'acheminer vers la sortie. La structure ou la topologie de ce dernier est la manière dont les neurones sont connectés. Les structures

résultantes sont très variées mais elles sont souvent réparties en deux grandes familles à savoir: les réseaux de neurones non bouclés et les réseaux de neurones bouclés [34].

II.3.3.2.1. Les réseaux non bouclés

Un réseau de neurones non bouclé ou réseau unidirectionnel (propagation avant) est représenté par un ensemble neurones connectés entre eux tel que l'information circulant des couches entrées vers les couches sorties au travers d'éventuelles couches intermédiaires (couche cachées) sans retour en arrière. Ce type de réseau comprend deux groupes d'architecture: les réseaux monocouches et les réseaux multicouches (Figure II.9).

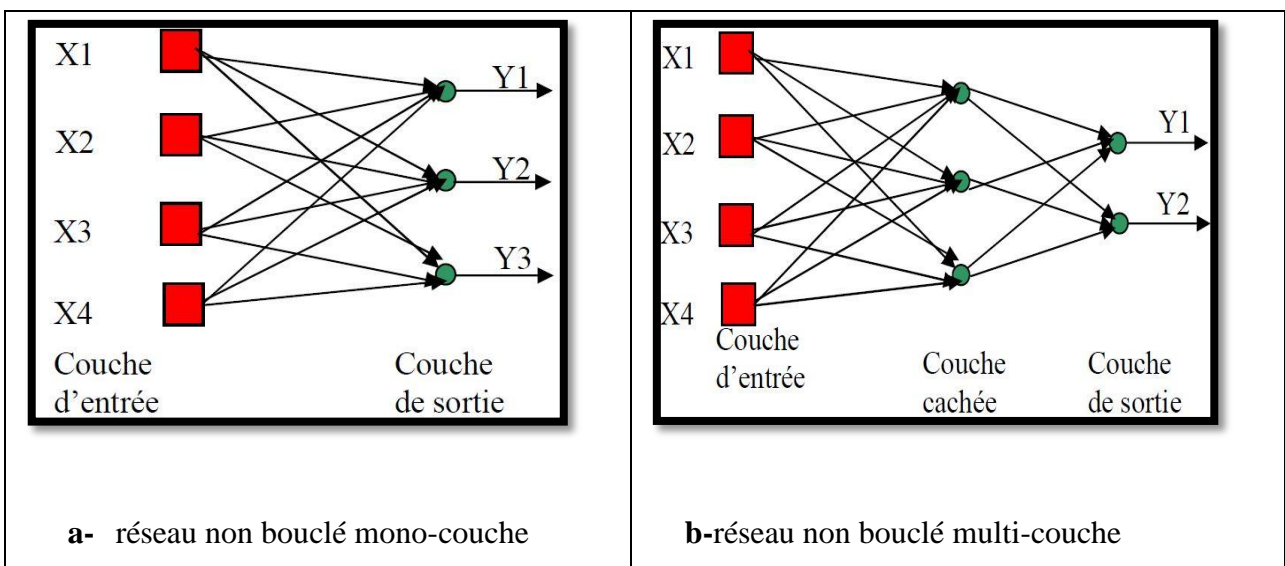


Figure II.9: Réseaux de neurones non bouclés

II.3.3.2.2. Réseaux bouclés (récursifs)

Les réseaux récurrents possèdent une structure similaire à celle des réseaux unidirectionnels mais complétée par des connexions entre éléments de la même couche ou vers des couches amants, ces réseaux sont assez puissants, car leurs fonctionnement est séquentiel et adopte un comportement dynamique [23].

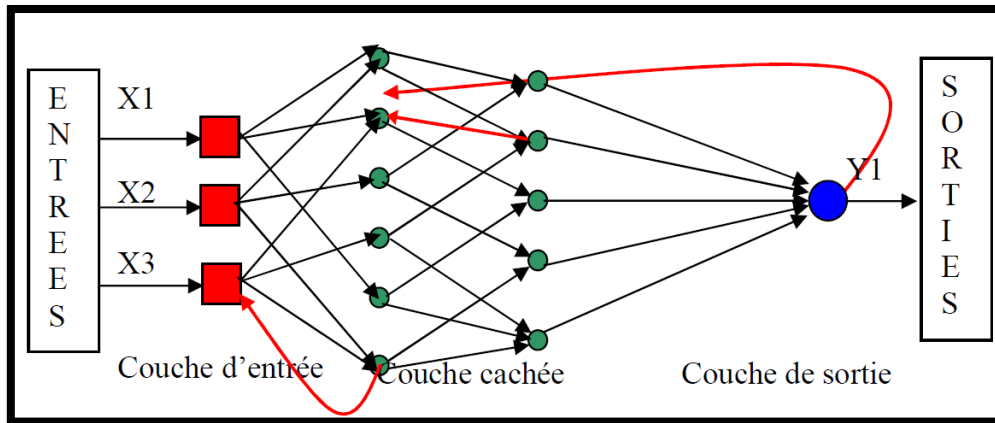


Figure II.10: réseaux de neurones bouclés

II.3.3.3. Algorithme d'apprentissage

Comme le cerveau humain, les réseaux de neurones artificiels (RNA) peuvent apprendre par expérience. Ainsi, suite à l'application séquentielle de plusieurs entrées à apprendre, les algorithmes d'apprentissage modifient la valeur des poids entre les neurones ainsi que la valeur des biais de façon à améliorer les performances du RNA [34], [35].

En fait, l'ajustement des poids est plus au moins efficace tout dépendamment de la connaissance que nous avons sur le système à modéliser ou sur le problème à résoudre. Il existe trois grandes classes d'apprentissage à savoir:

- A. Apprentissage non supervisé:** ce type d'apprentissage est choisi lorsqu'il n'y a pas de connaissances a priori des sorties désirées pour des entrées données. En fait, c'est de l'apprentissage par exploration où l'algorithme d'apprentissage ajuste les poids des liens entre neurones de façon à maximiser qualité de classification des entées.
- B. Apprentissage par renforcement (semi-supervisé):** dans ce cas, bien que les sorties idéales ne soient pas connues directement, il y a un moyen quelconque de connaître si les sorties du RNA s'approchent ou s'éloignent du but visé. Ainsi les poids sont ajustés de façon plus au moins aléatoire et la modification est conservée si l'impact est positif ou rejeté sinon.
- C. Apprentissage supervisé (retro-propagation):** cet algorithme d'apprentissage ne peut être utilisé que lorsque les combinaisons d'entrées-sorties désirées sont connues. L'apprentissage est alors facilité et par là, beaucoup plus rapide que pour les deux autres algorithmes puisque l'justement des poids est fait directement à partir de l'erreur (la différence entre la sortie obtenue par le RNA et la sortie désirée.

Pour ces trois types d'apprentissage, il y'a un choix traditionnel entre:

- ✓ **l'apprentissage dit préapprentissage (offline):** toutes les données sont dans une base d'exemples d'apprentissage qui sont traitée simultanément.
- ✓ **l'apprentissage continu (on-line):** les exemples sont présentés les uns après les autres au fur et à mesure de leurs disponibilités.

L'ajustement des poids des liens entre les neurones peut s'effectuer selon diverses équations mathématiques. On se contentera de citer deux méthodes des plus utilisées.

II.3.3.3.1. La règle d'apprentissage du perceptron

L'apprentissage du perceptron peut être décrit par les étapes ci-dessous :

1. Initialiser les poids w_i avec des valeurs aléatoires
2. Présenter une entrée \vec{x} de la base d'apprentissage
3. Calculer la valeur d'activation du neurone (la sortie).

$$O = F\left(\sum_{i=1}^n w_i x_i\right) \quad (\text{II.12})$$

4. Calculer l'erreur sur la sortie:

$$E = D - O \quad (\text{II.13})$$

5. Modifier les poids selon la relation:

$$w_i(t+1) = w_i(t) + k \cdot x_i \cdot E \quad (\text{II.14})$$

Où k est le pas d'apprentissage ($k > 0$).

6. Retourner à l'étape 2 jusqu'à ce que $E=0$ pour tous les exemples de la base d'apprentissage.

Avec:

x: l'entrée.

O: la sortie.

D: la sortie désirée.

K: la contrainte d'apprentissage.

❖ **Algorithme:**

Entrée: un échantillon de S.

Initialisation aléatoire des poids w_i pour i entre 1 et n .

- répéter

Prendre un exemple (x D) dans S

Calculer la sortie O du perceptron pour l'entrée x

% mise à jour des poids %

- Si $D \neq O$
 - Pour i allant de 1 à n

$$w_i \leftarrow w_i + (D - O) * k$$

Fin pour.

Fin Si.

Fin répéter.

Sortie: un perceptron définit par (w_1, w_2, \dots, w_n) .

II.3.3.3.2. Algorithme de retro propagation (back-propagation)

L'algorithme de retro propagation est l'un des algorithmes supervisés les plus utilisés pour l'apprentissage des réseaux de neurones. L'algorithme de retro propagation est basé sur la généralisation de la règle de WINDROW HOFF (la règle delta), en utilisant une fonction d'activation sigmoïde. Le principe de cet algorithme est la propagation d'un signal provenant des nœuds d'entrées vers la sortie et ensuite on propage l'erreur de la sortie vers les couches internes (cachées) jusqu'à l'entrée. Le critère de performance à minimiser peut être exprimé

par:
$$j(t) = \frac{1}{2} e^2(t) = \frac{1}{2} [y_d(t) - y(t)] \quad (\text{II.15})$$

$$\Delta w_{ij} = \eta \frac{\delta j(t)}{\delta y(t)} \quad (\text{II.16})$$

Avec:

$e(t)$: l'écart entre la sortie désirée $y_d(t)$ et la sortie réelle $y(t)$.

η : Le coefficient d'apprentissage.

- L'ajustement des coefficients (poids) synaptiques se fait par la méthode du gradient basée sur la formule itérative suivante:
- Pour la couche de sortie: actualisation des poids

$$w_{hj}(t+1) = w_{hj}(t) + \Delta w_{hj} \quad (\text{II.17})$$

$$\Delta w_{hj} = \eta \delta_j O_h \quad (\text{II.18})$$

Avec:

$$\delta_j = f'(O) (O_{jd} - O_j).$$

j: l'indice des neurones de la couche de sortie.

h: l'indice des neurones de la couche cachée.

i: l'indice des neurones de la couche d'entrée.

- L'actualisation des biais:

$$\theta_j(t+1) = \theta_j(t) + \Delta \theta_j \quad (\text{II.19})$$

Avec: $\Delta \theta_j = \eta \delta_j . 1$.

- Pour la couche cachée:
- Actualisation des poids:

$$w_{ih}(t+1) = w_{ih}(t) + \Delta w_{ih} \quad (\text{II.20})$$

$$\Delta w_{ih} = \eta \delta_h O_i \quad (\text{II.21})$$

Avec: $\delta_h = f'(h) . \sum \delta_j w_{hj}(t+1)$.

- Actualisation de biais:

$$\theta_h(t+1) = \theta_h(t) + \Delta \theta_h \quad (\text{II.22})$$

$$\Delta \theta_h = \eta \delta_h . 1 \quad (\text{II.23})$$

❖ Algorithme de retro propagation

Entrée: un échantillon de S.

- Initialisation des poids:
- Répéter:

Prendre un exemple de S.

% calcul de δ_j , actualisation des w_{hj} et θ_j %

- Pour: toutes les cellules de sortie j (j allant de 1 à n).

$$\delta_j = f'(O_j) \cdot (O_{jd} - O_j)$$

$$b_j = b_j + k \cdot \delta_j \cdot 1$$

- Fin pour

- Pour i allant de 1 à p

- Pour j allant de 1 à n

$$w_{hj} = k \cdot \delta_j \cdot O_h + w_{hj}$$

Fin pour

Fin pour

% calcule des δ_h , actualisation des w_{ih} et b_{ih} %

Pour toutes les cellules cachées h (h allant de 1 à p)

Pour j allant de 1 à m

$$z = z + \delta_j \cdot w_{hj}$$

Fin pour

$$\delta_h = f'(O_h) \cdot z$$

Fin pour

Pour i allant de 1 à n

Pour h allant de 1 à p

$$\theta_h = \theta_h + k \cdot \delta_h \cdot 1$$

$$w_{ih} = w_{ih} + k \cdot \delta_h \cdot x_i$$

Fin pour

Fin pour

Fin répéter

II.3.4. Systèmes neuro-flous

La logique floue et les réseaux de neurones sont des technologies complémentaires. Ces techniques ont chacune leurs forces et leurs faiblesses. La combinaison de ces deux approches dans un système intégré semble plus prometteuse pour le développement des systèmes intelligents [36].

Les réseaux de neurones multicouches constituent des approximateurs universels grâce à leur capacité d'apprentissage, mais leurs structures et leurs paramètres n'ont pas toujours d'interprétation physique. De plus, la connaissance humaine ne peut être exploitée pour les construire.

En revanche, les systèmes d'inférences flous possèdent deux points forts par rapport aux réseaux de neurones. D'une part, ils sont généralement construits à partir de la connaissance humaine, d'autre part, ils ont une capacité descriptive et élevée due à l'utilisation des variables linguistiques. L'approche traditionnelle pour la conception des systèmes d'inférences flous est basée sur les connaissances acquises par des opérateurs experts. Cette approche a prouvée son efficacité dans plusieurs applications.

Depuis les années 1990, de nombreux travaux ont été orientés vers la combinaison des deux techniques, ce qui a donné naissance aux approches neuro-floues. Un réseau de neurone et un système flou sont combinés dans une architecture homogène. Le système peut être interprété comme un réseau neuronal spécial avec des paramètres flous ou comme un système flou mis en application sous une forme distribuée parallèle. Le schéma générale d'une structure neuro-floue se présente sous cette forme:

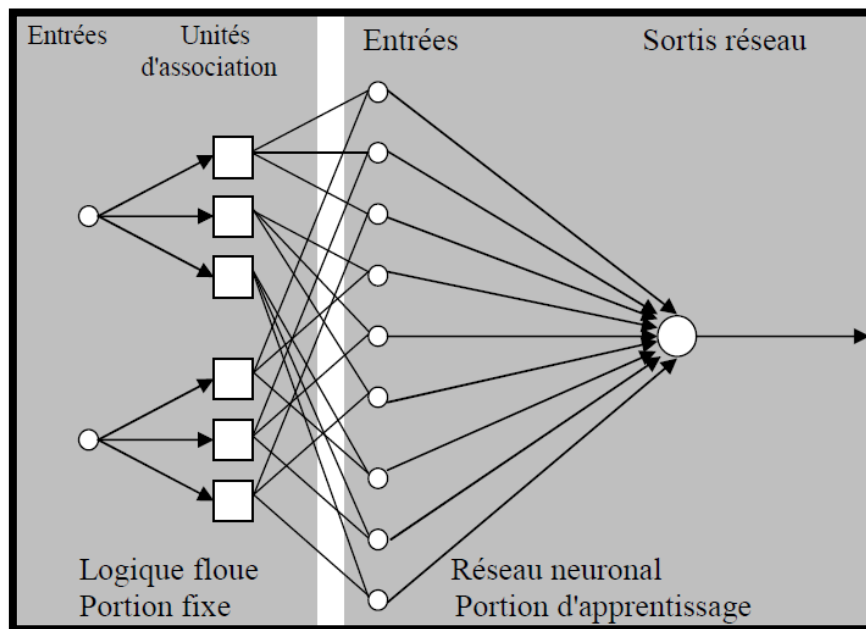


Figure II.11: structure générale d'un réseau neuro-flou.

II.3.4.1. Définition

George Lee définit un système neuro-flou comme étant un réseau neuronal multicouche avec des paramètres flous, ou comme un système flou mis en application sous une forme distribuée parallèle.

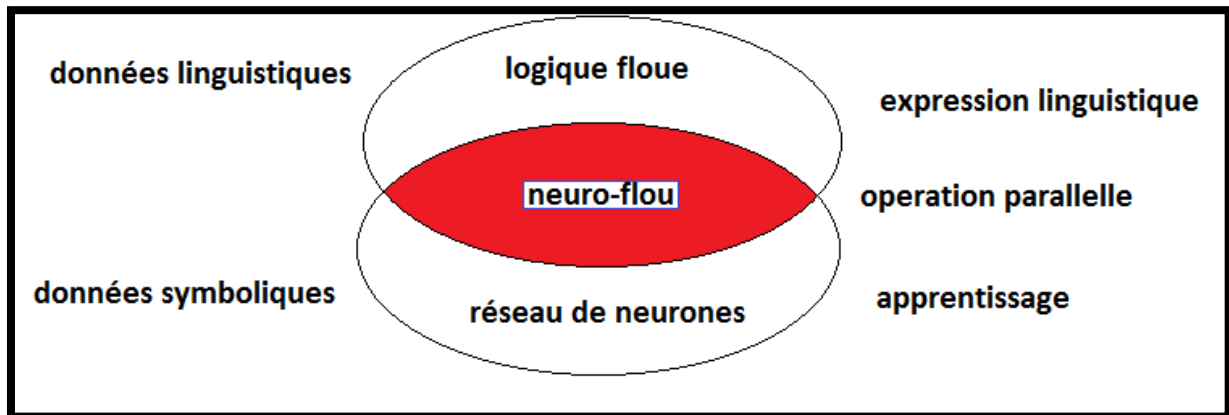


Figure II.12: principe du système neuro-flou

II.3.4.2. Méthodes de combinaisons neuro-floues

Il existe quatre grandes catégories de combinaisons des réseaux de neurones avec la logique floue [31] [37].

II.3.4.2.1. Réseau flou neuronal

Dans ces réseaux, les techniques floues sont employées pour augmenter les possibilités du processus d'apprentissage et d'exécution des réseaux de neurones.

II.3.4.2.2. Système neuronal/flou simultanément

Le réseau de neurones et le système flou fonctionnent ensemble sur la même tâche, mais sans s'influencer, ce qui signifie que ni l'un ni l'autre n'est employé pour déterminer les paramètres de l'autre. En générale le réseau neuronale traite les entrées ou poste-traite les sorties du système flou.

II.3.4.2.3. Modèle neuro-flou coopératif

Dans ce modèle le réseau de neurones est employé pour déterminer les paramètres (règles et ensembles flous) d'un système flou. Après la phase d'apprentissage, le système flou fonctionne sans le réseau de neurones. C'est une forme simple des systèmes neuro-flous.

II.3.4.2.4. Modèle neuro-flou hybride

Les approches neuro-flou modernes sont de cette forme; un réseau neuronal et un système flou sont combinés dans une architecture homogène. Le système peut être interprété comme un réseau neuronal spécial avec des paramètres flous ou comme un système flou mis en application sous une forme distribuée parallèle.

II.3.4.3. Implémentation des réseaux neuro-flous

Dans la littérature il existe une multitude de systèmes neuro-flous: les systèmes neuro-flous coopératifs, concurrents, les systèmes dit hybrides ou neuro-flous adaptatifs, le modèle NEFCLASS, ANFIS, FALCON et GARIC etc...

Compte tenu de nos objectifs, nous avons retenu un seul type des systèmes neuro-flous à savoir: systèmes d'inférences neuro-flous adaptatifs (ANFIS)

II.3.4.3.1. systèmes d'inférences neuro-flous adaptatifs (ANFIS)

Les systèmes d'inférences neuro-flous adaptatifs (ANFIS), sont des systèmes hybrides utilisant l'inférence floue de takagi-sugeno. La structure ANFIS est constituée de cinq couches comme est illustré dans la figure (II.13). La première couche cachée fuzzifie les variables d'entrées et les operateurs type T-norme calculent la partie prémisse des règles dans la deuxième couche cachée. La troisième couche cachée normalise les poids des règles suivit par la quatrième couche cachée où les paramètres des parties conclusions des règles sont déterminés. La couche de sortie calcule la somme de tous les signaux venant de la quatrième couche [38].

Son entrainement est accomplit par l'algorithme du gradient de descente (retro propagation) pour l'optimisation des paramètres des parties prémisses, et l'algorithme des moindres carrée pour la résolution des paramètres conclusion afin de minimiser l'erreur E.

Nous considérons un système à deux entrées x_1 et x_2 et une sortie y , modélisé par un système flou de type TSK, composé des deux règles suivantes:

Si x_1 est A_1 et x_2 est B_1 alors $y_1 = f_1(x_1, x_2) = a_1x_1 + b_1x_2 + c_1$.

Si x_1 est A_2 et x_2 est B_2 alors $y_2 = f_2(x_1, x_2) = a_2x_1 + b_2x_2 + c_2$.

Avec:

x, y : Entrées.

A_j, B_j : Sous-ensembles flous (variables linguistiques).

a, b et c: des paramètres ajustables qui sont déterminés pendant le processus d'apprentissage [34].

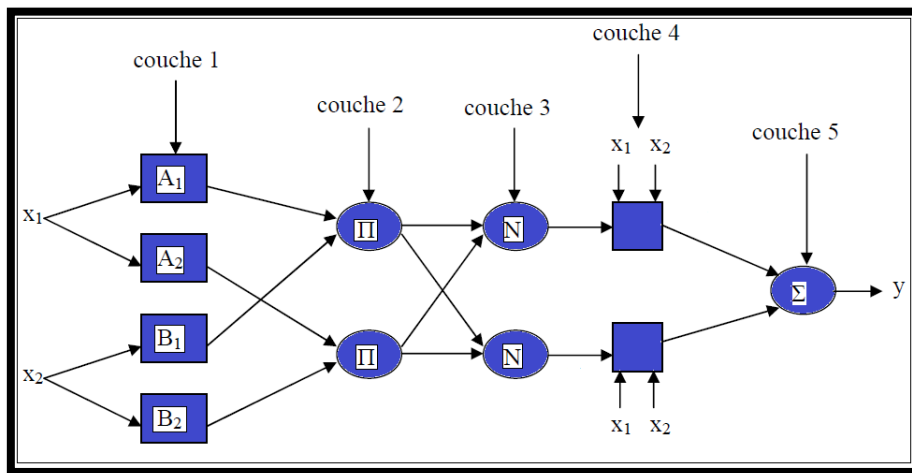


Figure II.13: architecture ANFIS.

On constate à partir de la figure (II.13) que les nœuds sont de deux types différents selon leurs fonctionnalités: les nœuds carrés, dit adaptatifs, contiennent des paramètres, et les nœuds circulaires, dit fixes, n'ont pas de paramètres. Cependant chaque nœud applique une fonction sur ses signaux d'entrées.

La sortie $O_{k,i}$ du nœud i de la couche k , dépend des signaux provenant de la couche $(k-1)$ et des paramètres du nœud (k, i) .

Dans le réseau de la figure (II.13), les nœuds d'une même couche ont des fonctions issues d'une même famille :

- **Couche 1:** génération du degré d'appartenance

Les nœuds de cette couche possèdent des paramètres ajustables. La fonction du nœud est semblable à la fonction d'appartenance à un sous-ensemble flou de l'univers de discours des entrées

$$O_{1,i} = \mu A_i(x_1) \quad \text{Pour } i=1,2 \quad (\text{II.24})$$

Où:
$$O_{1,i} = \mu B_{i-2}(x_2) \quad \text{Pour } i=3,4 \quad (\text{II.25})$$

Où x est l'entrée du nœud i , A_i le terme linguistique associé à sa fonction d'appartenance μA_i .

- **Couche 2:** génération du poids de la règle i :

Dans cette couche, les nœuds sont des nœuds fixes, leurs fonction est la multiplication (l'inférence prod). La sortie des nœuds de cette couche peut être représentée comme une équation de cette forme $O_{2,i} = w_i = \mu A_i(x_1) \cdot \mu B_i(x_2)$ pour $i=1,2$.

Chaque nœud de cette couche représente une règle floue de sugeno. Il reçoit les sorties des nœuds de fuzzification et calcule son activation.

- **Couche 3:**normalisation des poids de la règle i :

Les nœuds de cette couche sont des nœuds circulaires appelés N (norm). Le nœud calcule le degré d'activation normalisé de la règle floue. La sortie des nœuds de cette couche représente la contribution de la règle floue au résultat final et est appelée poids de la règle normalisée. Elle est donnée par :

$$O_{3,i} = \bar{w}_i = \frac{w_i}{w_1 + w_2} \quad \text{pour } i=1,2 \quad (\text{II.26})$$

- **Couche 4:** calcule de la sortie des règles

Chaque nœud i de cette couche et un nœud adaptatif avec une fonction qui est représentée tel:

$$O_{4,i} = \bar{w}_i f_i = \bar{w}_i (a_i x_1 + b_i x_2 + c_i) \quad \text{pour } i=1,2 \quad (\text{II.27})$$

Où: \bar{w}_i est la sortie de la couche 3.

$\{a_i, b_i, c_i\}$: sont des paramètres ajustables de sortie de la règle i . ils font référence au paramètres de la conséquence.

- **Couche5:** l'agrégation, la somme de toutes les entrées à poster de la couche 4.

Cette couche comprend un seul nœud fixe avec la fonction somme (sum), elle calcule la sortie de l'ANFIS en faisant la somme de toutes les sorties de tous les nœuds de défuzzification, c'est-à-dire:

$$O_{5,1} = y = \sum_i \bar{w}_i f_i \quad (\text{II.28})$$

❖ **Algorithme d'apprentissage**

Le système ANFIS applique le mécanisme d'apprentissage des réseaux de neurones sur des techniques d'inférences floues. D'un autre terme l'ANFIS est un système d'inférence flou (SIF) dont les paramètres des fonctions d'appartenance sont ajustés en utilisant l'algorithme de rétro propagation, ou en combinaison avec un autre type d'algorithme comme les moindres carrés.

Dans la première phase (forward) les paramètres de la prémisse sont supposés fixes et les valeurs optimales des paramètres en conséquence sont estimées en appliquant la méthode des moindres carrés. Puis, la sortie du système est calculée et l'erreur observée est utilisée pour ajuster les paramètres de la prémisse par le biais de l'algorithme de rétro propagation standard.

	Propagation avant	Rétro propagation
Paramètres de la prémisse	Fixe	Descente du gradient
Paramètres de la conséquence	Estimateur des moindres carrés	Fixe
Signal	La sortie des nœuds	Erreurs de sortie

Deux propagations dans l'algorithme hybride d'apprentissage pour ANFIS.

a) La propagation avant

Dans la propagation avant de l'algorithme hybride d'apprentissage, la sortie des nœuds se propage en avant jusqu'à la quatrième couche et les paramètres de la conséquence sont identifiés par la méthode des moindres carrés.

Lorsque les valeurs des paramètres des prémisses sont fixées, la sortie du réseau peut être exprimée par une combinaison linéaire des paramètres de la conséquence.

$$f = \frac{w_1}{w_1 + w_2} f_1 + \frac{w_2}{w_1 + w_2} f_2 = \bar{w}_1 f_1 + \bar{w}_2 f_2 = (\bar{w}_1 x_1) a_1 + (\bar{w}_1 x_2) b_1 + (\bar{w}_1) c_1 + (\bar{w}_2 x_1) a_2 + (\bar{w}_2 x_2) b_2 + (\bar{w}_2) c_2$$

Où $a_1, a_2, b_1, b_2, c_1, c_2$ sont les paramètres linéaires de la conséquence.

$$f = A.X \tag{II.29}$$

Où X est un vecteur colonne de paramètres $\{a_1, b_1, c_1, a_2, b_2, c_2\}$ et A est une matrice de taille $p \times m$ où p est le nombre des N données d'entraînement. F est le vecteur de sortie de taille $p \times 1$.

En général les données d'entraînement ont un nombre plus grand que les paramètres de la conséquence, ce qui implique que la meilleure structure pour w est le minimum de l'erreur quadratique ($\min \|AX - f\|^2$).

La meilleure estimation par l'algorithme d'estimation des moindres carrés (LSE) de X est X^* en utilisant la matrice pseudo-inversible de X :

$$X^* = (A^T A)^{-1} A^T f \quad (\text{II.30})$$

Où : A^T est le transposé de A

$(A^T A)^{-1} A^T$ est la matrice pseudo-inversible de A .

b) Rétro-propagation

Dans la rétro propagation, le signal de l'erreur est propagé en arrière et les paramètres de la prémisse sont mis à jour par la descente du gradient.

$$\alpha_i(t+1) = \alpha_i(t) - \eta \frac{\delta E^k}{\delta \alpha_i} \quad (\text{II.31})$$

$$\beta_i(t+1) = \beta_i(t) - \eta \frac{\delta E^k}{\delta \beta_i} \quad (\text{II.32})$$

Où $\eta > 1$ est la constante d'apprentissage.

$$a_i(t+1) = a_i(t) - \eta \frac{\delta E^k}{\delta a_i} \quad (\text{II.33})$$

$$b_i(t+1) = b_i(t) - \eta \frac{\delta E^k}{\delta b_i} \quad (\text{II.34})$$

$$c_i(t+1) = c_i(t) - \eta \frac{\delta E^k}{\delta c_i} \quad (\text{II.35})$$

II.4. Conclusion

Dans ce chapitre, nous avons présenté les concepts fondamentaux de la commande classique (PID), commande intelligente basé sur la logique floue, les réseaux de neurones ainsi que les systèmes neuro-flous issus de la combinaison de la logique floue et les réseaux de neurones.

Chapitre III:
Simulations et
résultats

III.1. Introduction

L'objectif de notre travail dans ce chapitre consiste à synthétiser trois commandes, la première est une commande PID, la deuxième c'est une commande floue afin d'assurer un suivi de trajectoire et une stabilité pour notre système, et enfin nous synthétiserons une commande neuro-floue (ANFIS) pour une stabilisation d'altitude.

II.2. Application de la commande PID à l'UAV

À partir du modèle d'état, nous constatons que certains états dépendent de certaines commandes. Par exemple:

\dot{x}_2 Dépend de u_2 ;

\dot{x}_4 Dépend de u_3 ;

\dot{x}_6 Dépend de u_4 ;

$\dot{x}_8, \dot{x}_{10}, \dot{x}_{12}$ dépendent tous de u_1 .

Tel que :

- u_1 Représente la commande de puissance totale des moteurs (commande d'altitude).
- u_2 Représente la différence de puissance entre le moteur 2 et le moteur 4 (commande de roulis).
- u_3 Représente la différence de puissance entre le moteur 1 et le moteur 3 (commande de tangage).
- u_4 Représente la différence de puissance entre les moteurs tournant dans le sens horaire (moteur 1 et 3) et les moteurs tournant dans le sens antihoraire (moteurs 2 et 4), commande de lacet.

II.2.1. Choix des paramètres

Afin d'assurer un bon fonctionnement de notre système, le choix des paramètres est une étape pertinente. Nous avons donc commencé par chercher des paramètres en effectuant plusieurs essais pour amener notre système à un état stable.

Paramètres des Régulateurs x, y, z	K_p	5
	K_i	0
	K_d	10
Paramètres des Régulateurs φ, θ, ψ	K_p	10
	K_i	0
	K_d	20
Paramètres des Régulateurs de Vitesses pour les moteurs	K_p	1
	K_i	10

Tableau III.1: paramètres PID.

II.2.2. Schéma de régulation

Nous avons choisi une régulation à trois boucles:

- **Boucle01** : La première est une boucle de régulation de quatre contrôleurs PI pour une régulation de vitesse des quatre moteurs.

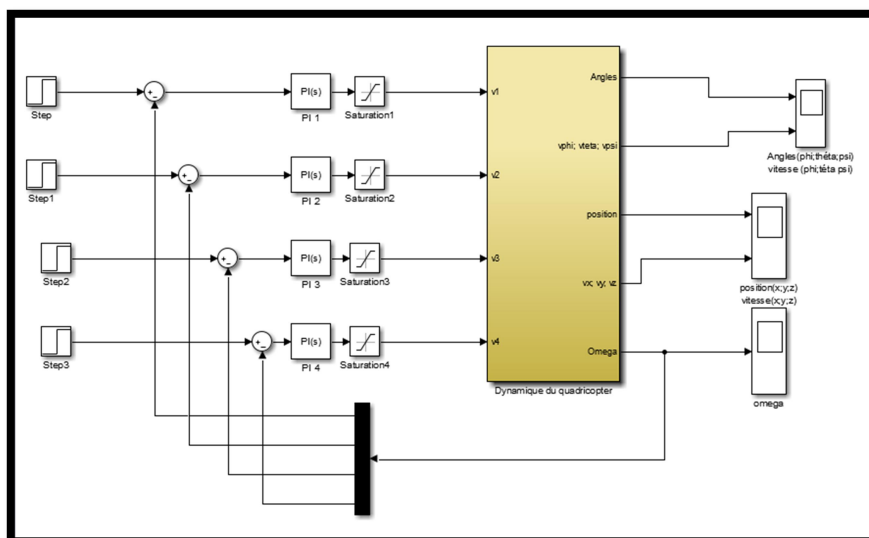


Figure III.1: schéma de régulation de vitesse des moteurs.

Résultat de simulation pour une référence ω désirée des quatre moteurs de 150 rad/s (1432 tr/min):

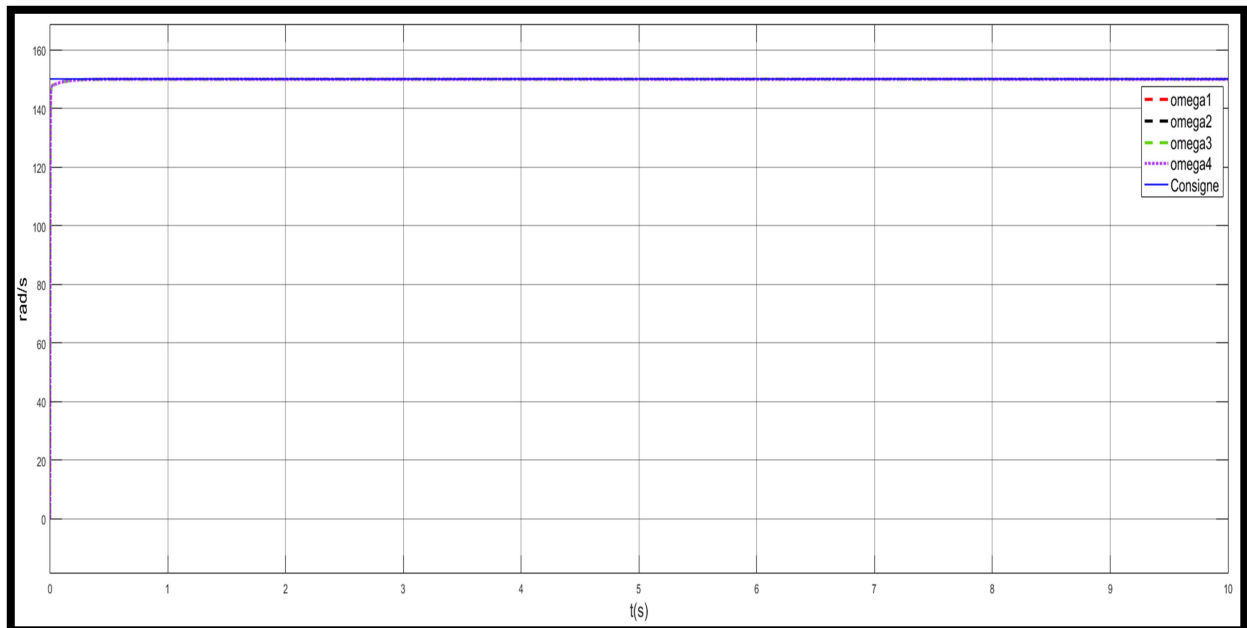


Figure III.2: réponse à un échelon de la régulation PI des moteurs.

On constate que la vitesse désirée est atteinte après 0.45s et ce pour les quatre moteurs vu que leur dynamique est identique avec des légères oscillations au régime permanent.

La réaction du système pour cette référence:

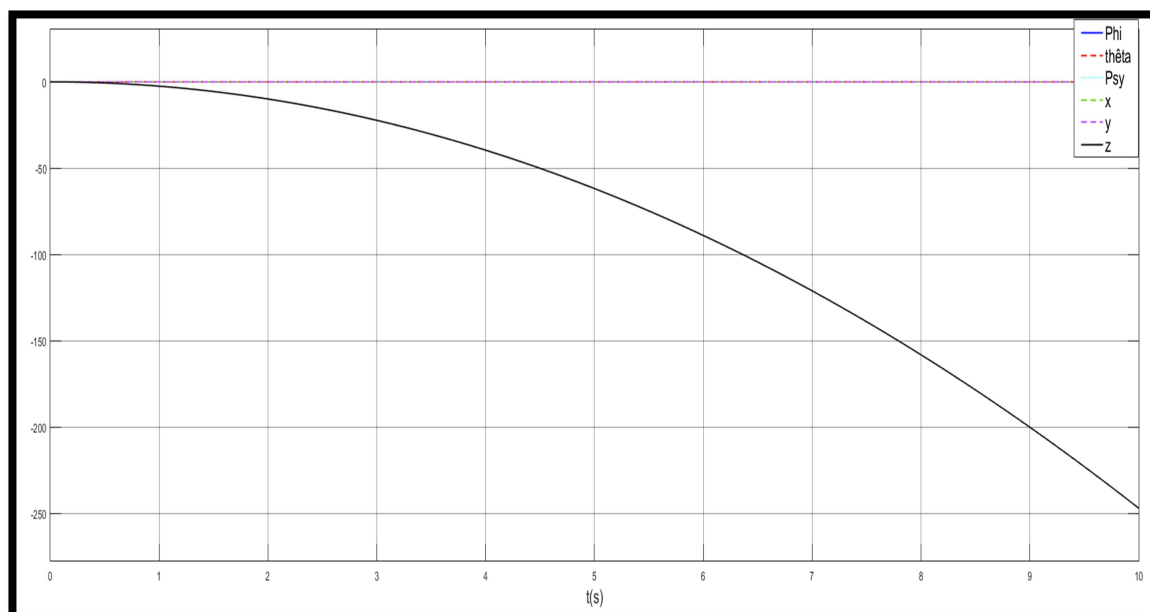


Figure III.3: réaction du système pour une vitesse de 150 rad/s pour les quatre moteurs.

On constate que les états $x, y, \varphi, \theta, \psi$ ne réagissent pas comme la consigne de référence est la même pour les quatre moteurs ce qui induit l'absence des moments de roulis, tangage ainsi que le lacet. Pour l'état z , On constate qu'il diverge vers des valeurs négatives car la force de portance générée par les quatre rotors est inférieure à celle du poids.

$$P = m.g = 9.81N$$

La force de portance:

$$F = b \cdot \sum_{i=1}^4 \omega_i^2$$

Sachant que :

$$\omega_1 = \omega_2 = \omega_3 = \omega_4 = 150 \text{ rad / s}$$

Ce qui nous donne :

$$F = 4.b.(150)^2 = 4.878N$$

Pour que l'UAV décolle, la force de portance doit être supérieure à celle du poids :

$$F = 4.b.\omega^2 > m.g$$

$$\omega > \sqrt{\frac{m.g}{4.b}}$$

$$\omega > 212.71 \text{ rad / s}$$

Résultat obtenu pour $\omega = 213 \text{ rad / s}$:

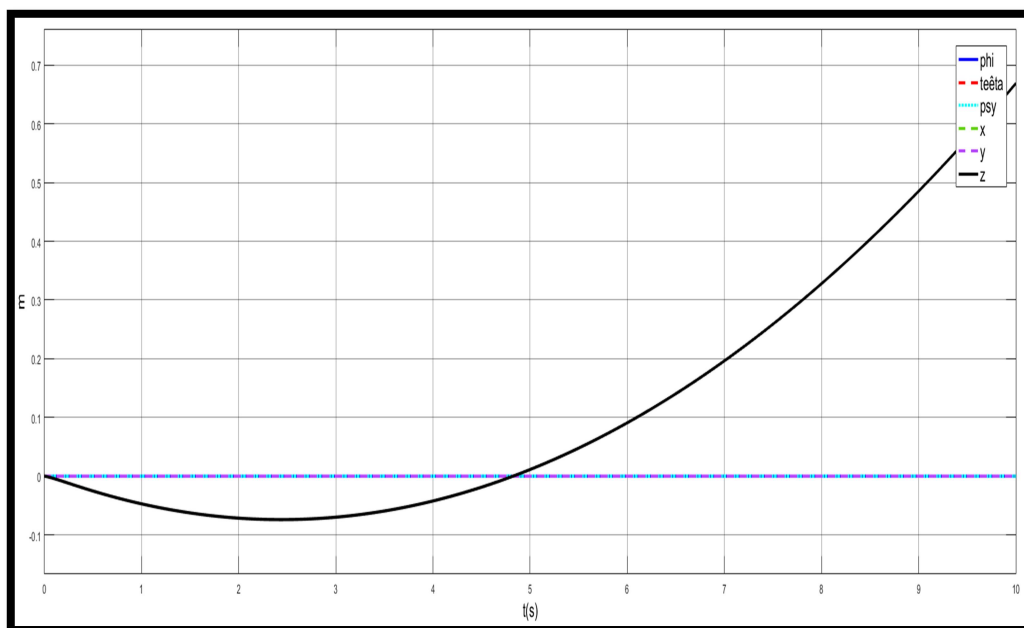


Figure III.4: réaction du système pour une vitesse de 213 rad/s pour les quatre moteurs.

En appliquant la différence entre les moteurs 1,3 et les moteurs 2,4.

$$\omega_1 = \omega_3 = 240 ; \omega_2 = \omega_4 = 200 :$$

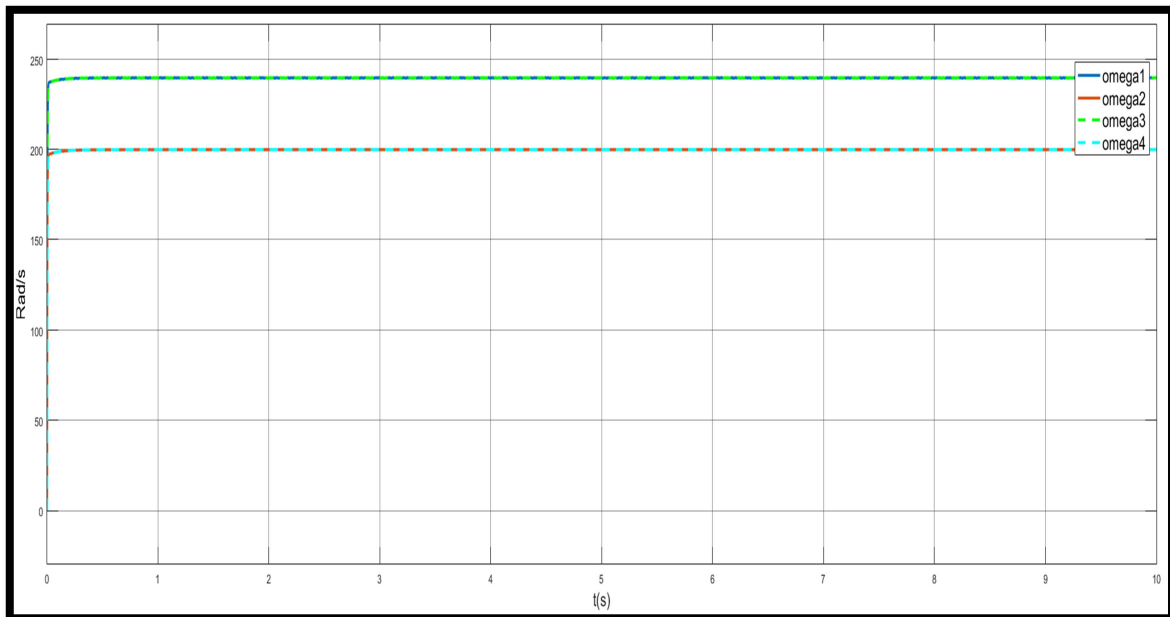


Figure III.5: réponse des quatre moteurs.

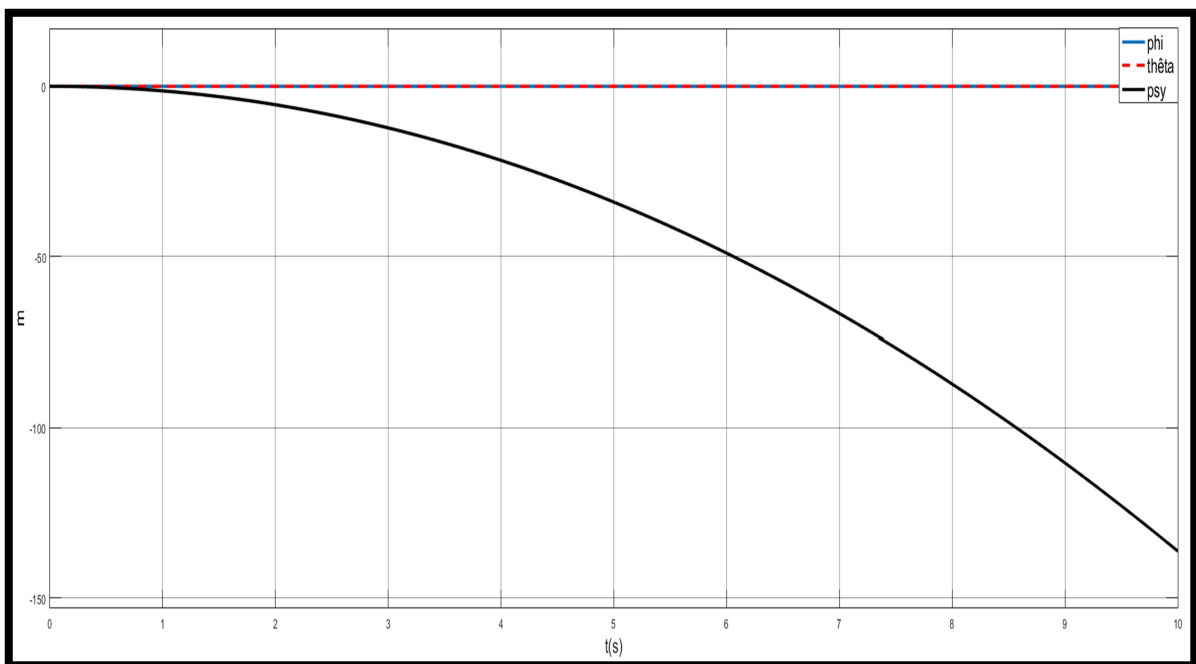


Figure III.6: réaction du système en appliquant une différence entre les moteurs 1,3 et 2,4.

On constate un changement de l'angle ψ .

En appliquant la différence entre les moteurs 2 et 4

$$\omega_1 = \omega_3 = 240 ; \omega_2 = 280 ; \omega_4 = 200 :$$

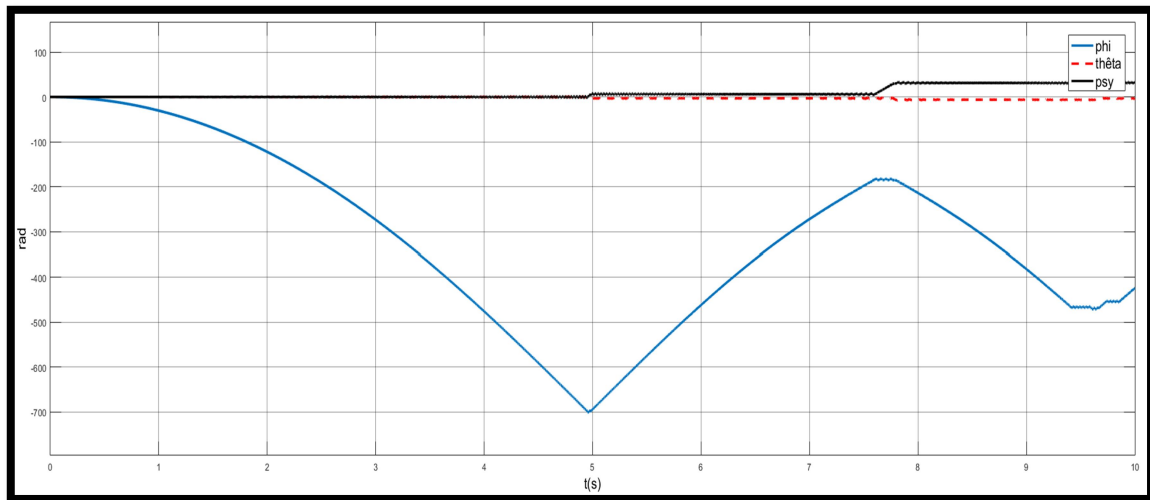


Figure III.7: réaction du système En appliquant la différence entre les moteurs 2 et 4.

On constate un changement de l'angle φ .

En appliquant la différence entre les moteurs 1 et 3.

$$\omega_2 = \omega_4 = 240 ; \omega_1 = 280 ; \omega_3 = 200 :$$

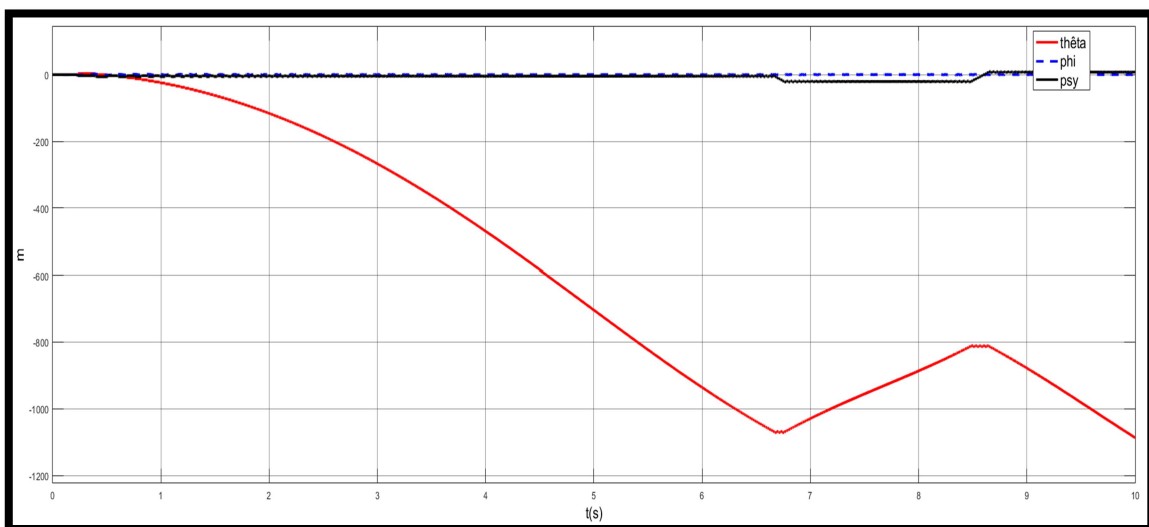


Figure III.8: réaction du système En appliquant la différence entre les moteurs 1 et 3.

On constate un changement de l'angle θ .

- Boucle2:** La deuxième est une boucle de régulation de trois contrôleurs PID pour une régulation de position des angles (φ, θ, ψ) , cette boucle a pour objectif de générer les vitesses des moteurs désirées à partir des commandes et ce en utilisant l'inverse de la matrice de commande $\omega_d = M^{-1}U$:

$$\begin{bmatrix} \omega_{1d}^2 \\ \omega_{2d}^2 \\ \omega_{3d}^2 \\ \omega_{4d}^2 \end{bmatrix} = \begin{pmatrix} b & b & b & b \\ 0 & -lb & 0 & lb \\ -lb & 0 & lb & 0 \\ d & -d & d & -d \end{pmatrix} \begin{bmatrix} U1 \\ U2 \\ U3 \\ U4 \end{bmatrix}$$

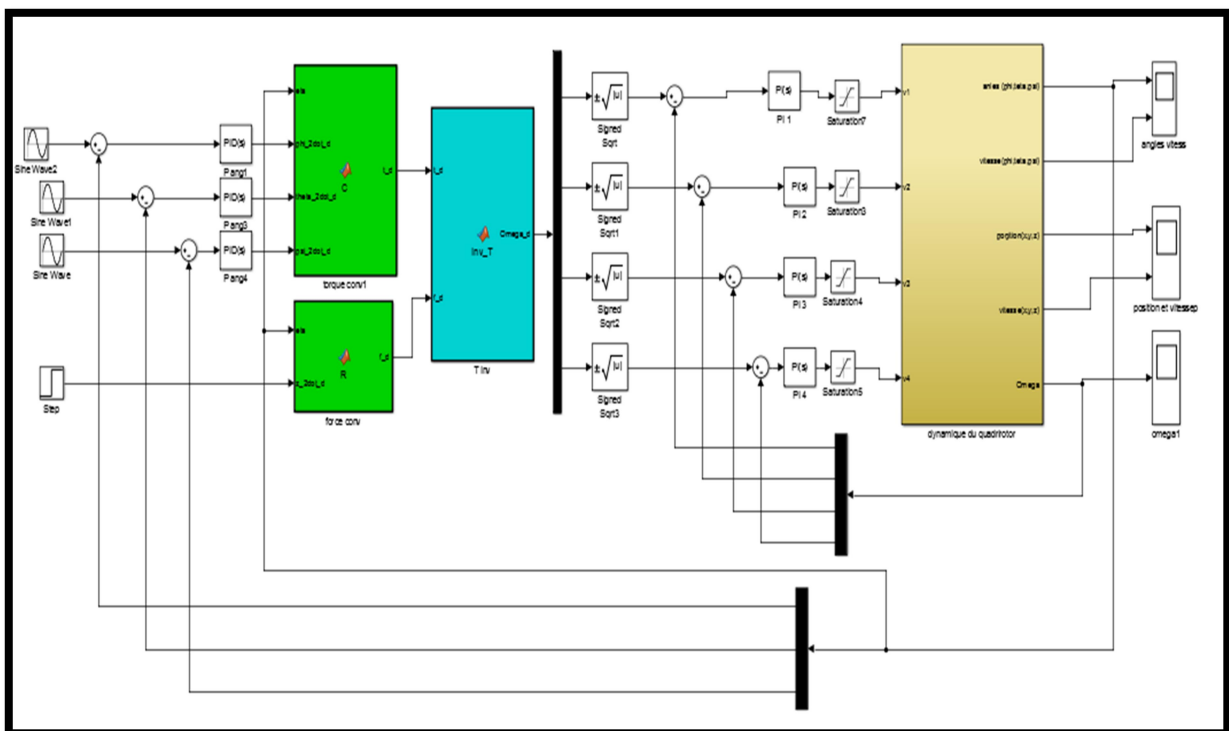


Figure III.9: schéma de régulation des angles.

Les résultats obtenus sont les suivant:

✓ Réponse à un signal échelon :

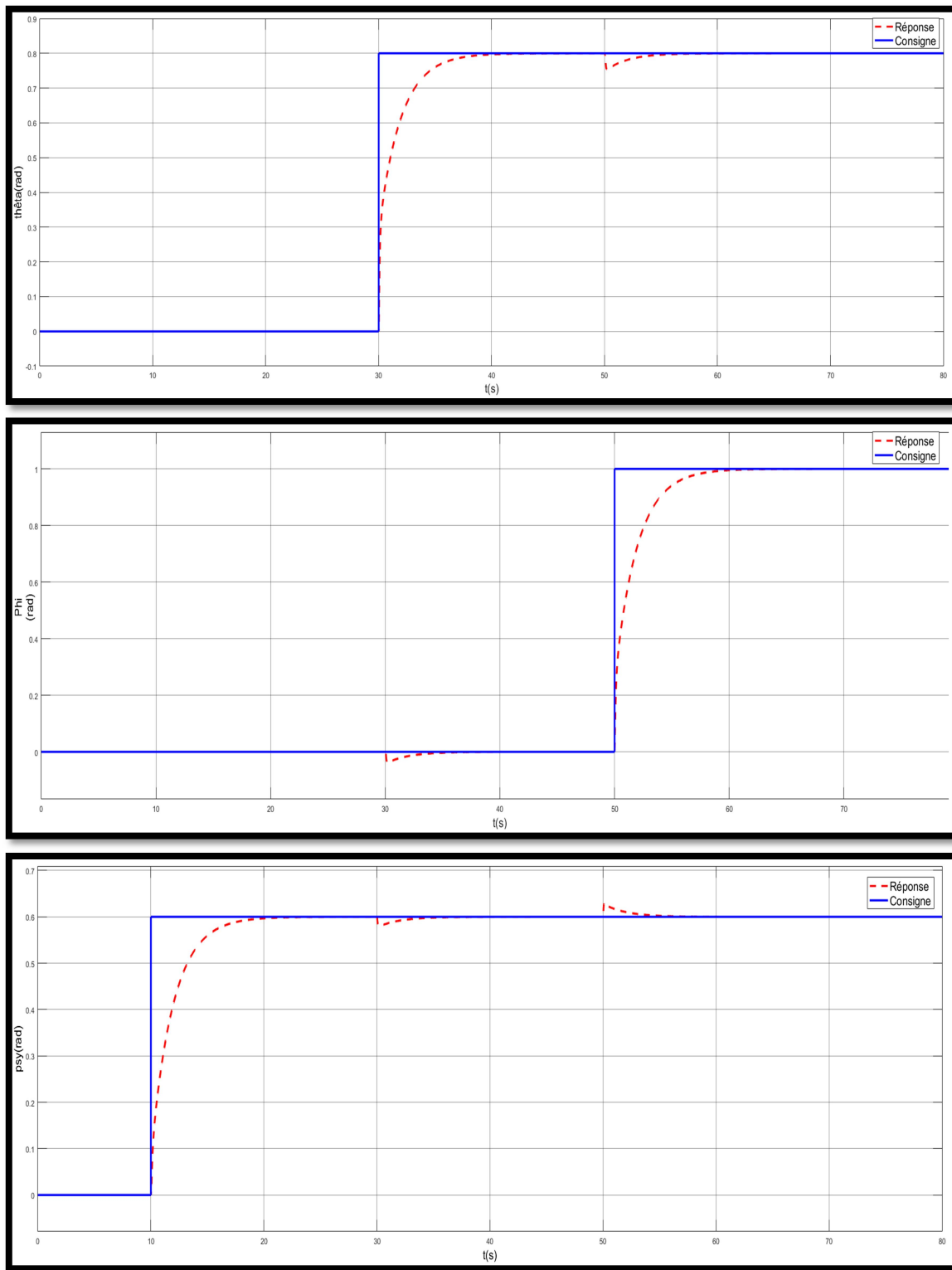


Figure III.10: Réponse à un échelon de la régulation PID sur tous les angles.

En étudiant ces réponses, nous avons extraits les temps de réponse à 5% de chaque variable:

- Le temps de réponse en phi est de 5.31s.
- Le temps de réponse en theta est de 5.00s.
- le temps de réponse en psy est de 5.58.

✓ Réponse à un signal sinusoïdal :

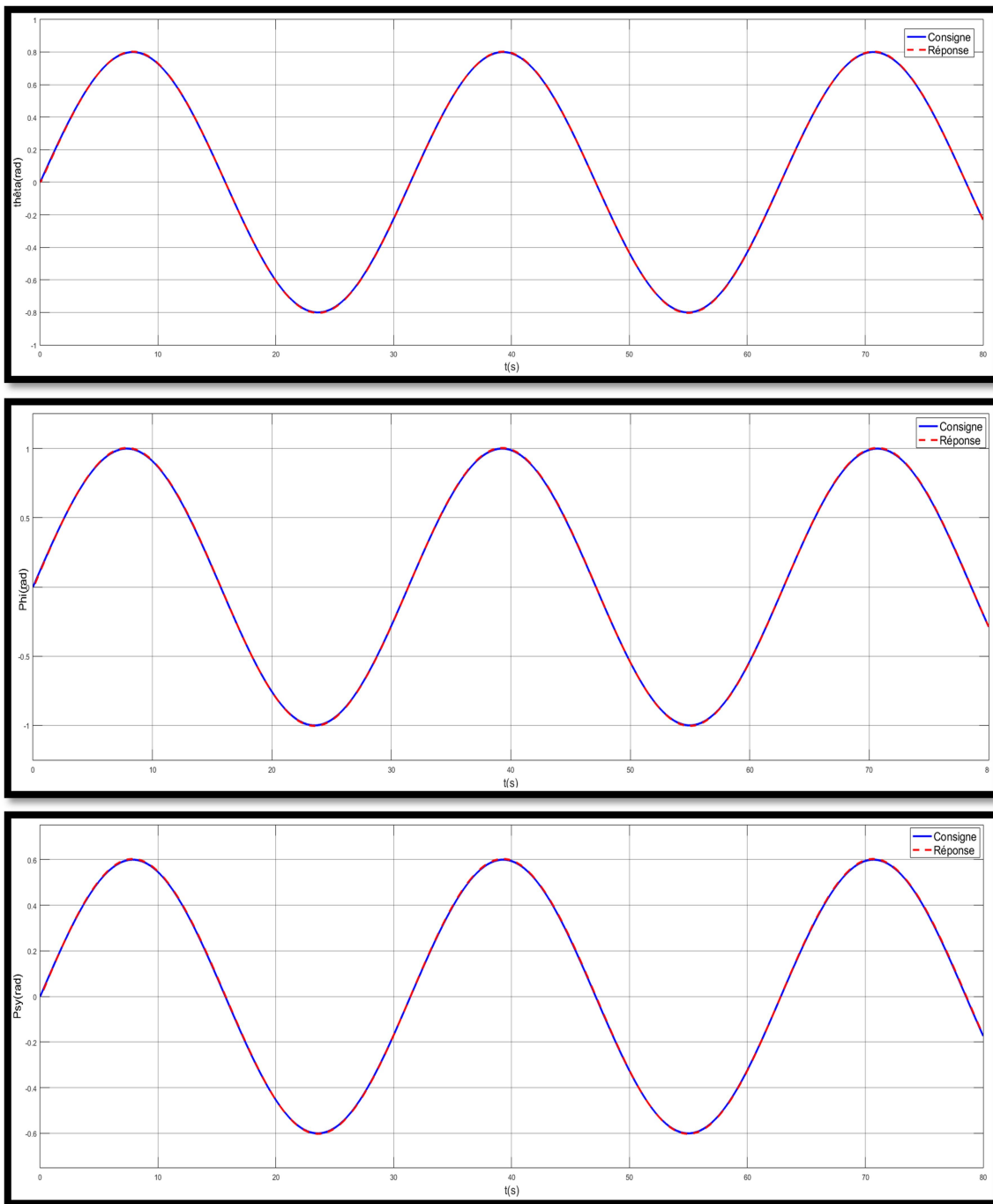


Figure III.11: réponse à un signal sinusoïdal de la régulation PID sur tous les angles.

Nous observons une bonne poursuite de la consigne.

✓ Régulation face aux perturbations

Pour réaliser ce dernier test nous allons fixer la consigne a une valeur constante et nous allons injecter des perturbations instantanées et a valeurs aléatoires. Les résultats sont présentés sur la figure suivante :

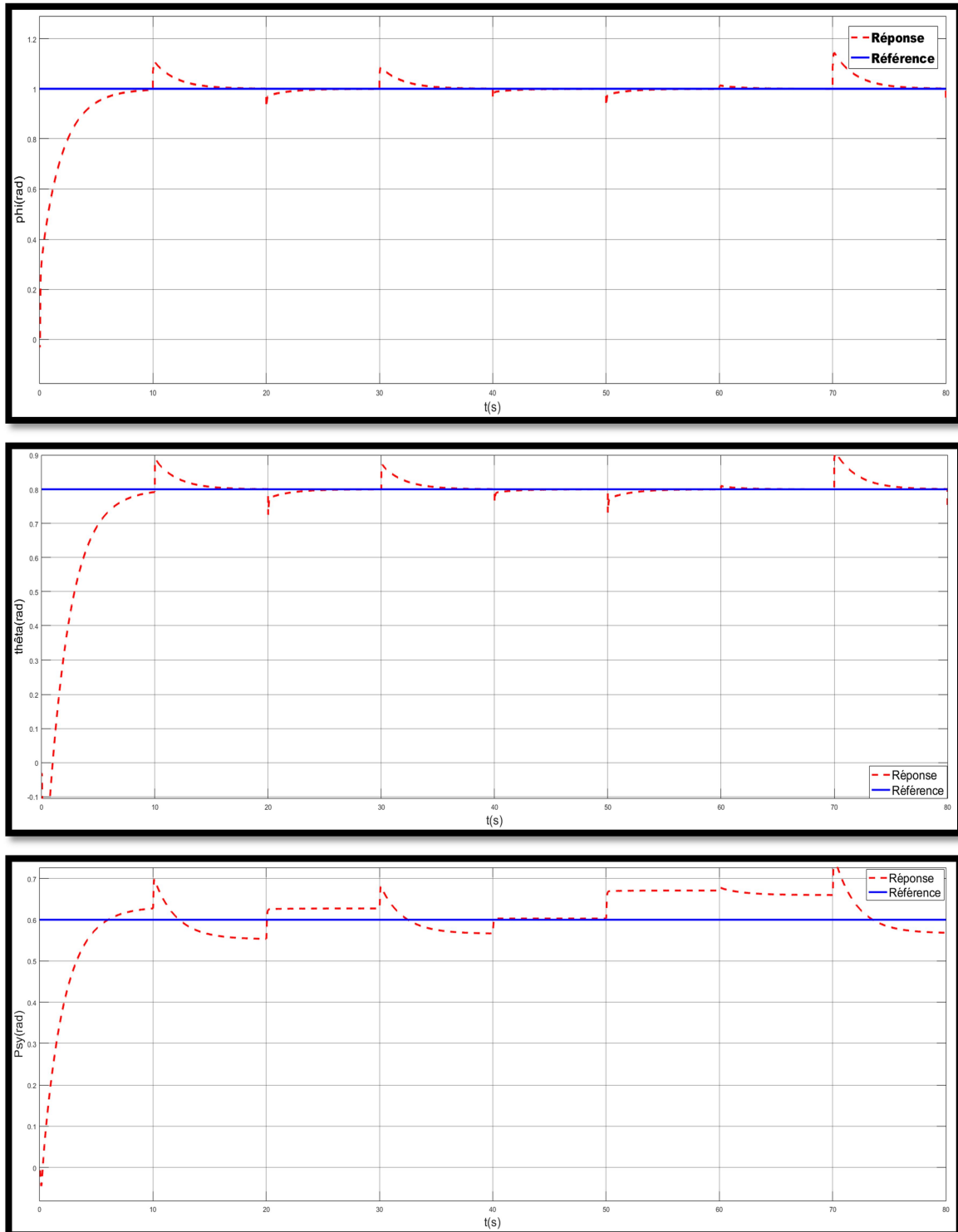


Figure III.12 : Réponse face aux perturbations de la régulation PID de tous les angles.

Nous observons que notre système rejette les perturbations.

- Boucle3:** la troisième boucle contient trois régulateurs PID pour une régulation de position (x, y, z), Cette boucle génère les angles (φ, θ) à partir des positions x et y désirées. Le schéma de régulation complet de notre système est donc le suivant:

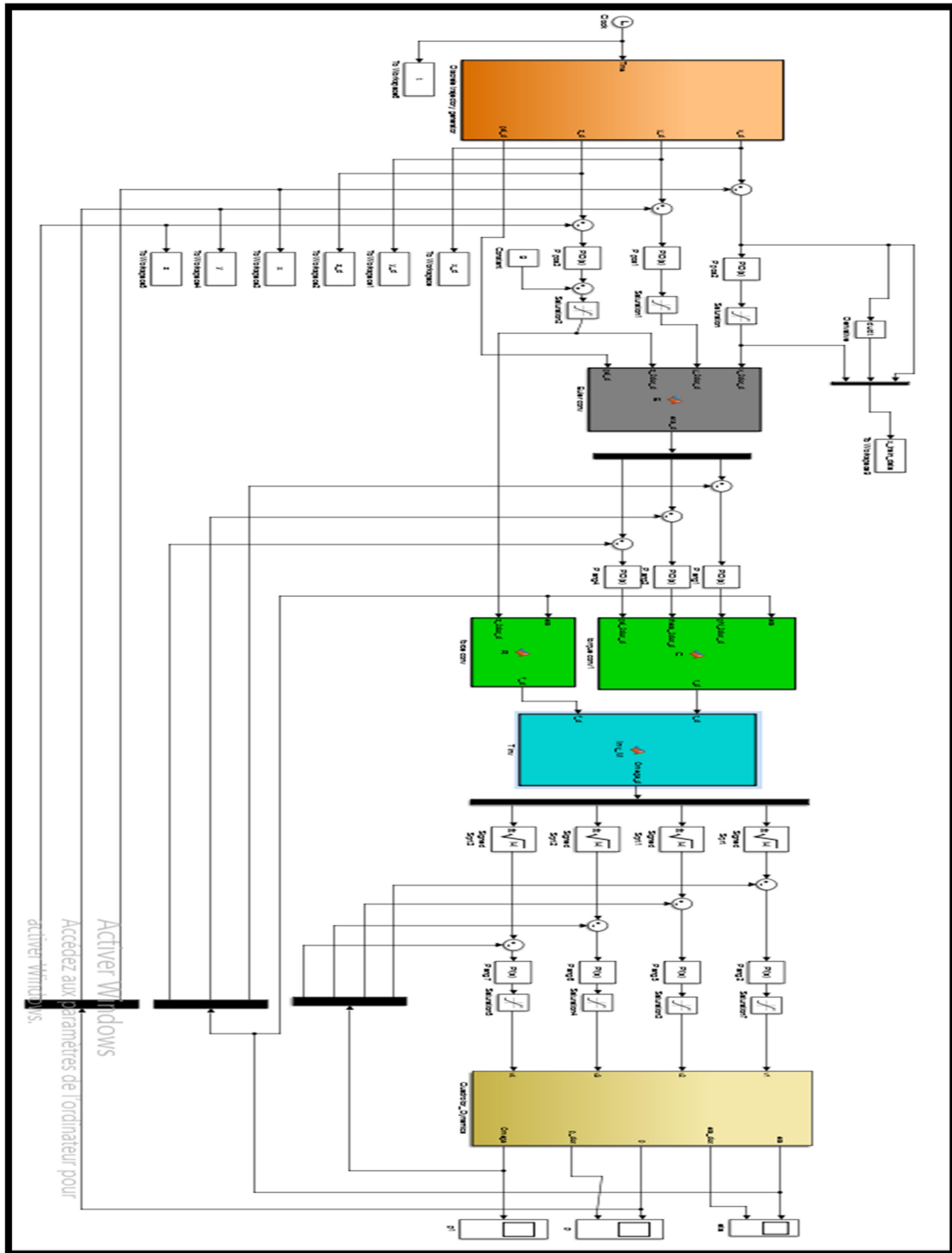


Figure III.13: schéma de régulation des positions.

Nous avons simulé notre système avec deux trajectoires différentes et qui sont illustrées dans les deux figures ci dessous:

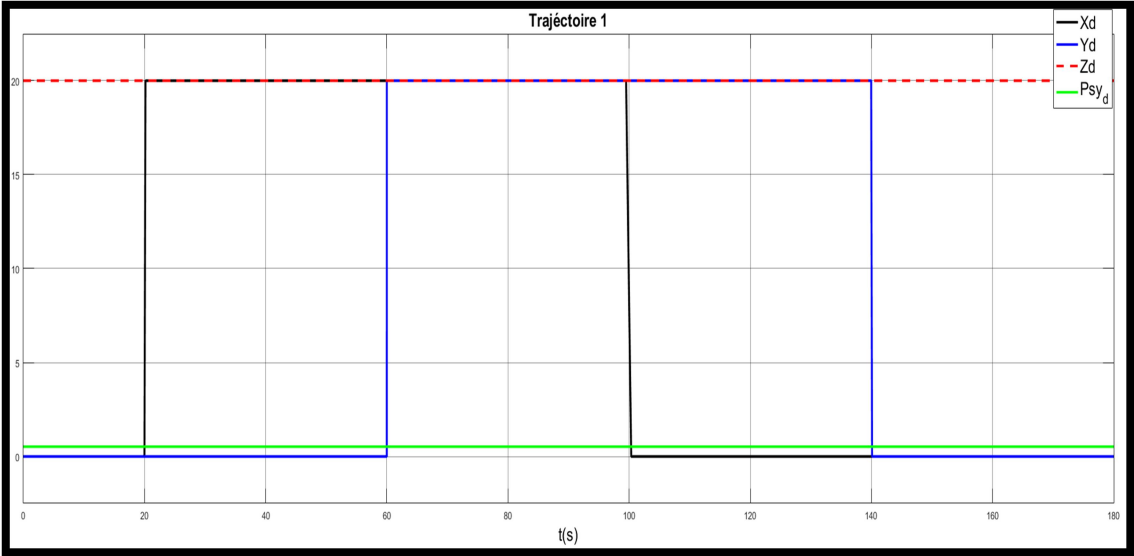


Figure III.14: Première trajectoire désirée.

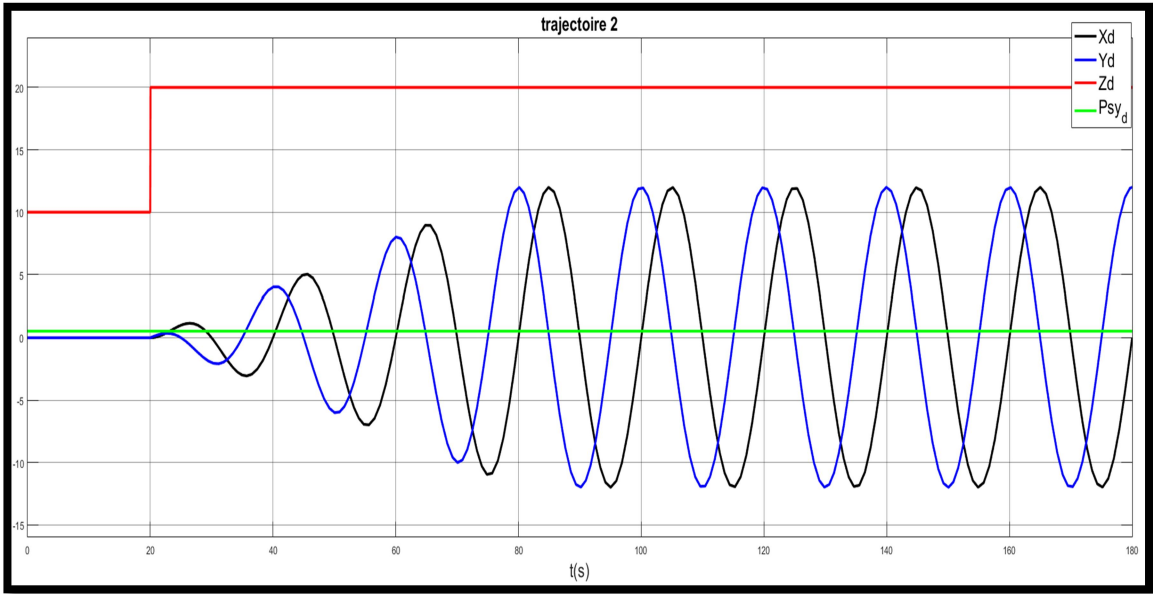


Figure III.15: Deuxième trajectoire désirée.

Les résultats de simulation obtenus sont :

✓ Pour la trajectoire 1:

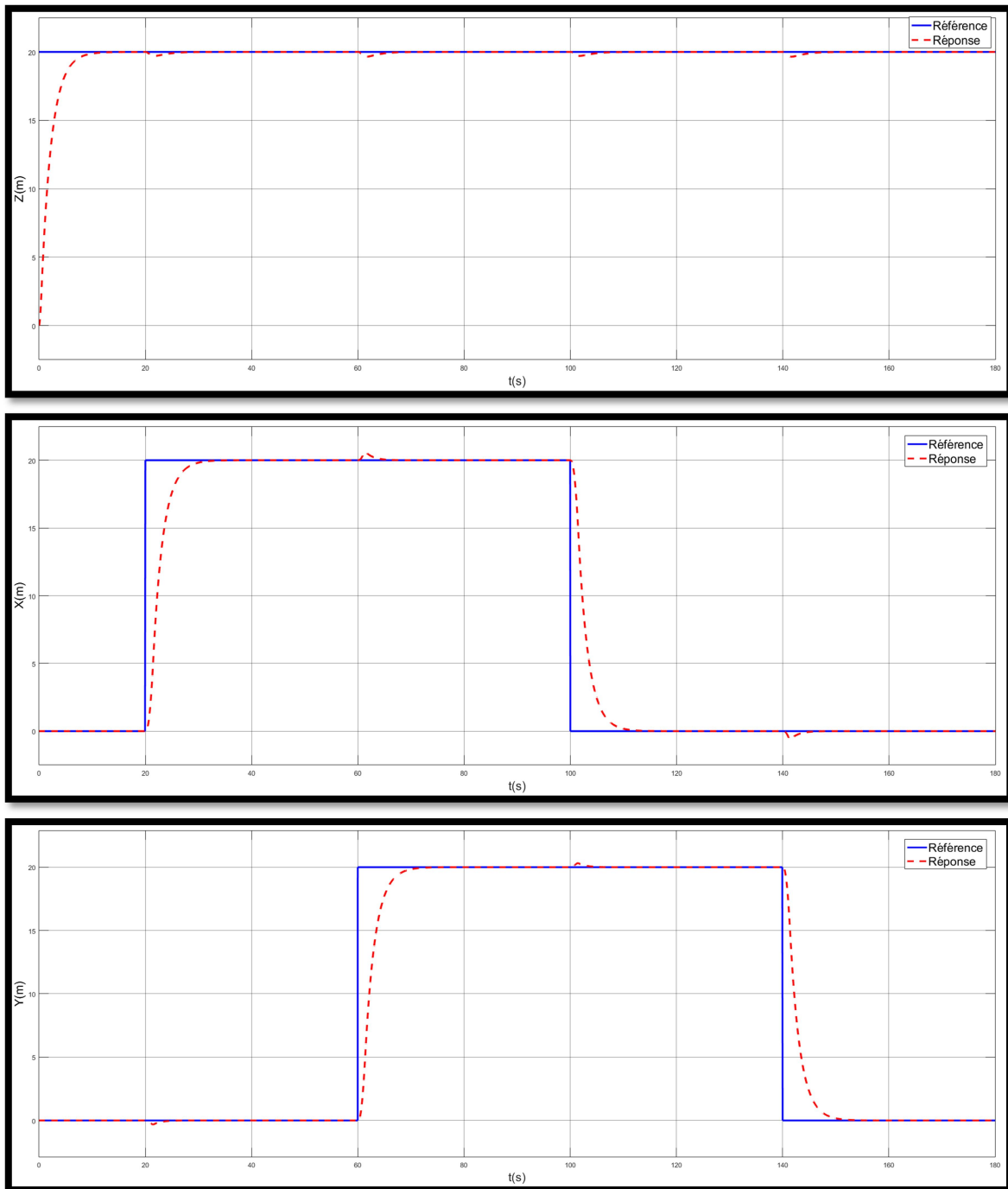


Figure III.16: Réponse des variables x , y , z à la première trajectoire désirée.

En étudiant ces réponses, nous avons extraits les temps de réponse à 5% de chaque variable:

- Le temps de réponse en z est de 5.72s.
- Le temps de réponse en x est de 6.45s.
- le temps de réponse en y est de 6.28.

✓ La trajectoire effectuée donc par l'UAV en 3D :

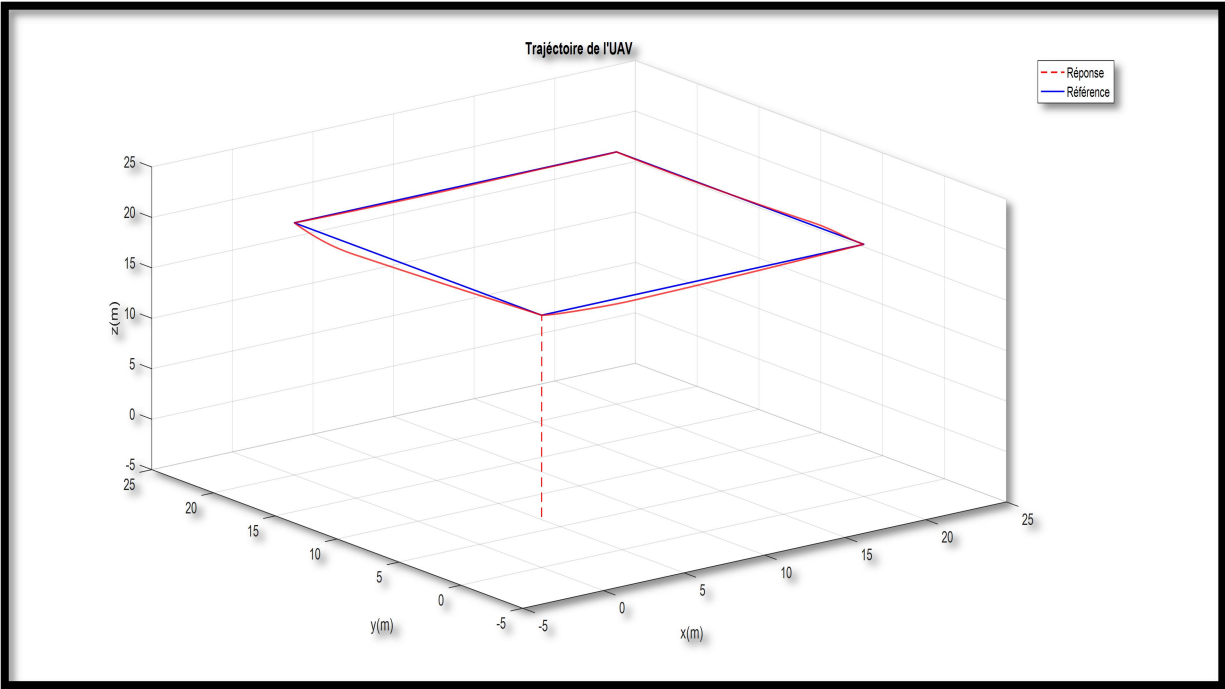


Figure III.17: Trajectoire de l'UAV en 3D.

✓ Pour la trajectoire2:

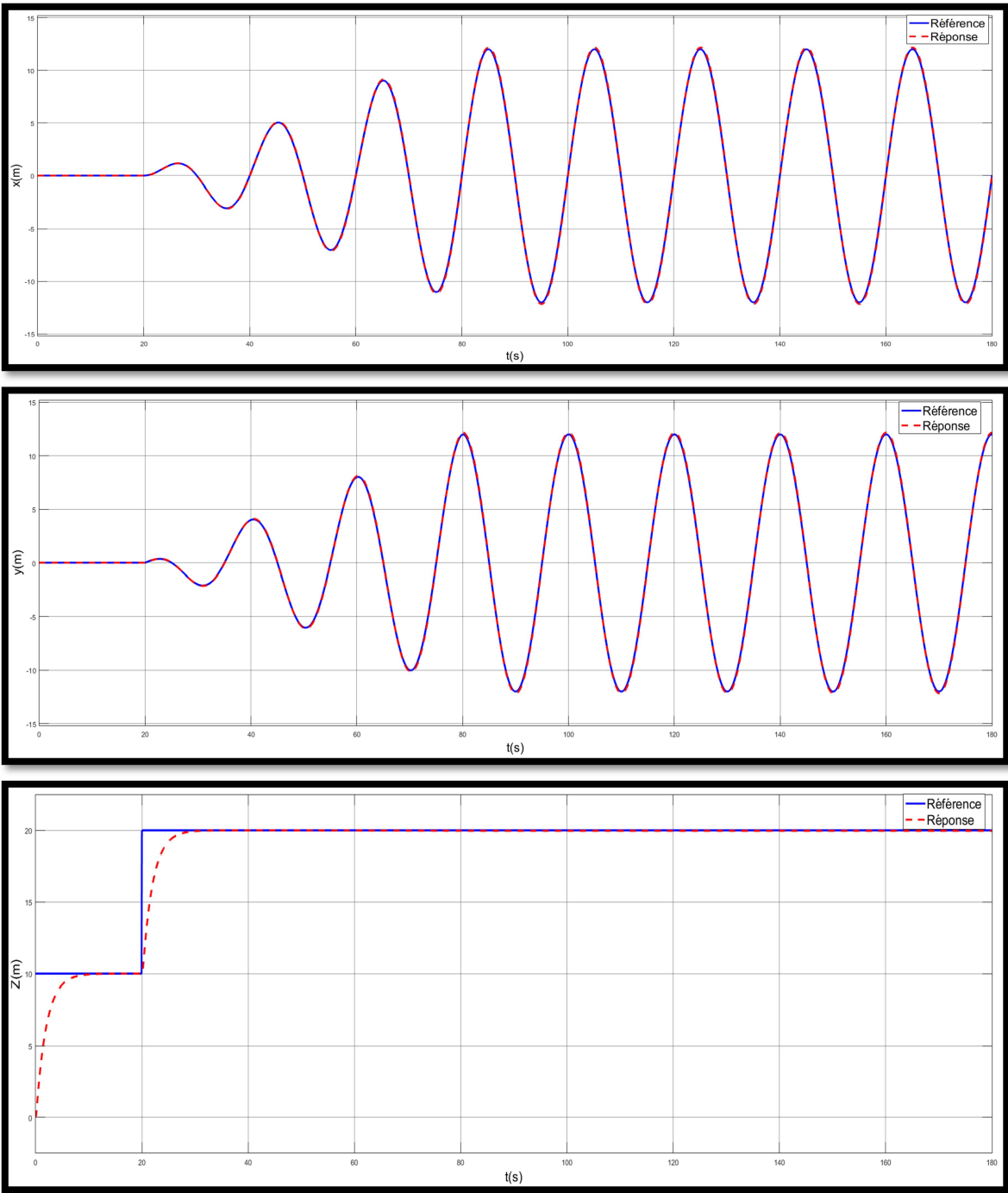


Figure III.18: Réponse des variables x, y, z à la deuxième trajectoire désirée.

Nous observons une bonne poursuite de la consigne.

- ✓ Trajectoire effectuée donc par l'UAV en 3D:

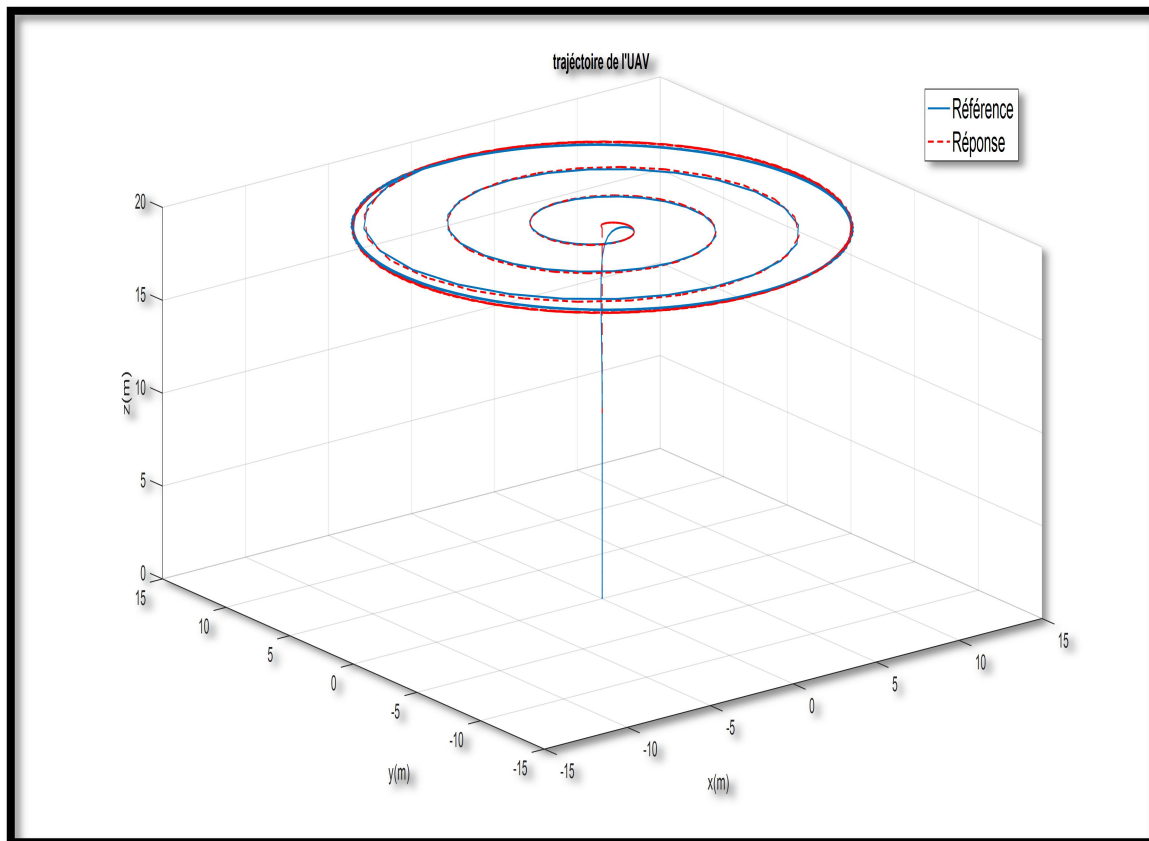


Figure III.19: Trajectoire de l'UAV en 3D.

- ✓ Régulation face aux perturbations

Pour réaliser ce dernier test nous allons fixer la consigne à une valeur constante et injecter des perturbations instantanées et à valeurs aléatoires. Les résultats sont présentés sur la figure suivante :

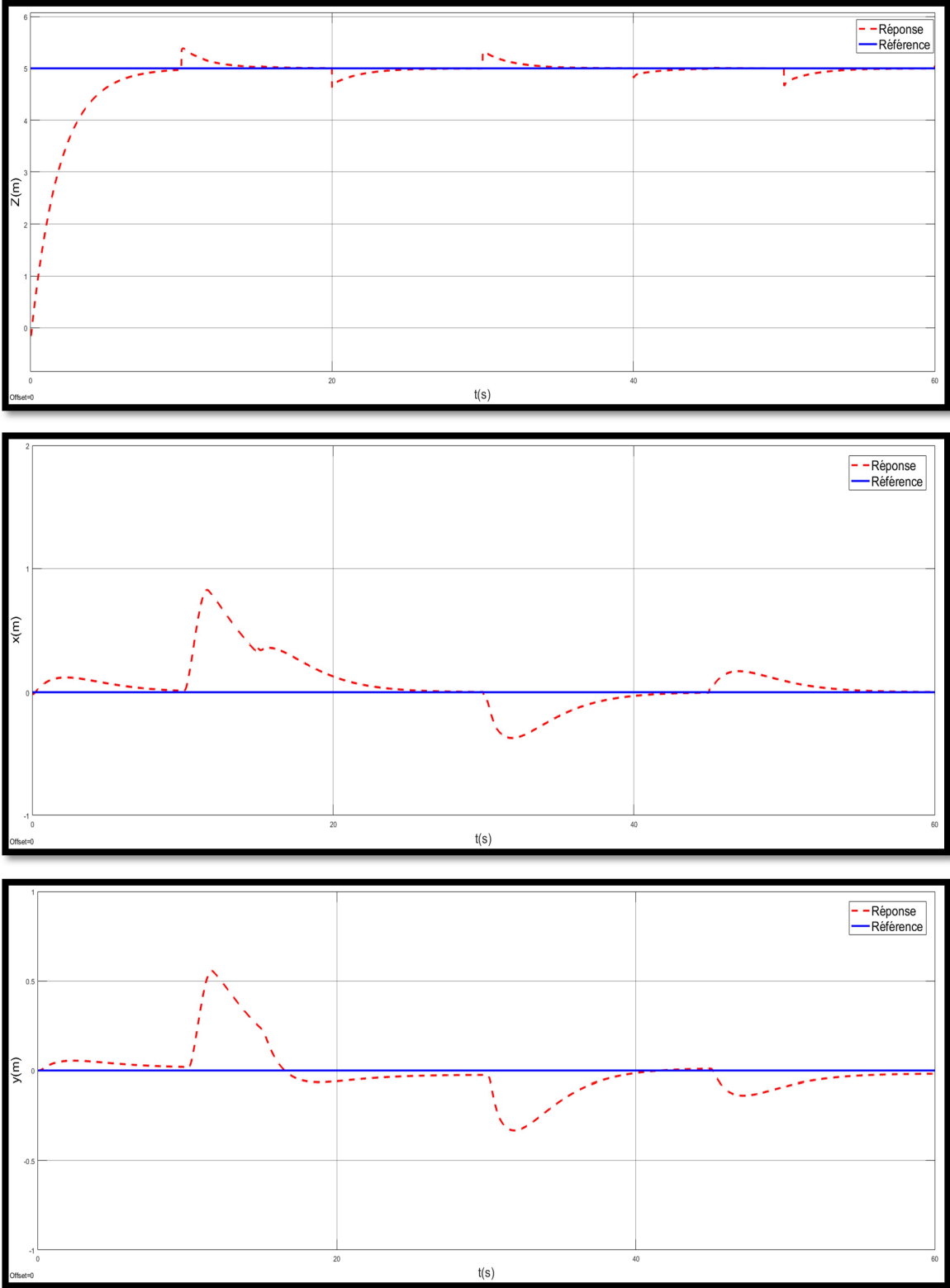


Figure III.20 : Réponse face aux perturbations de la régulation PID des états x, y, z .

Nous pouvons constater que les perturbations sont rejetées.

III.3. Application d'une commande floue à l'UAV

Dans la conception du régulateur flou, nous avons opté pour trois fonctions d'appartenance triangulaires pour chaque variable d'entrée et de sortie.

Chaque régulateur reçoit en entrée l'erreur de position et la vitesse de déplacement (ou de rotation) et grâce à neuf règles d'inférences nous obtenons, 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.

- Régulation des angles (φ, θ, ψ)

Pour la régulation sur les axes de rotation(φ, θ, ψ) , nous avons pris en entrée la différence entre la consigne et l'angle sur chaque axe (en radian) et la vitesse de déplacement (en radian par seconde). Après plusieurs tests et ajustements nous avons abouti aux fonctions d'appartenances pour les entrées représentées sur les figures suivantes.

✓ L'axe φ :

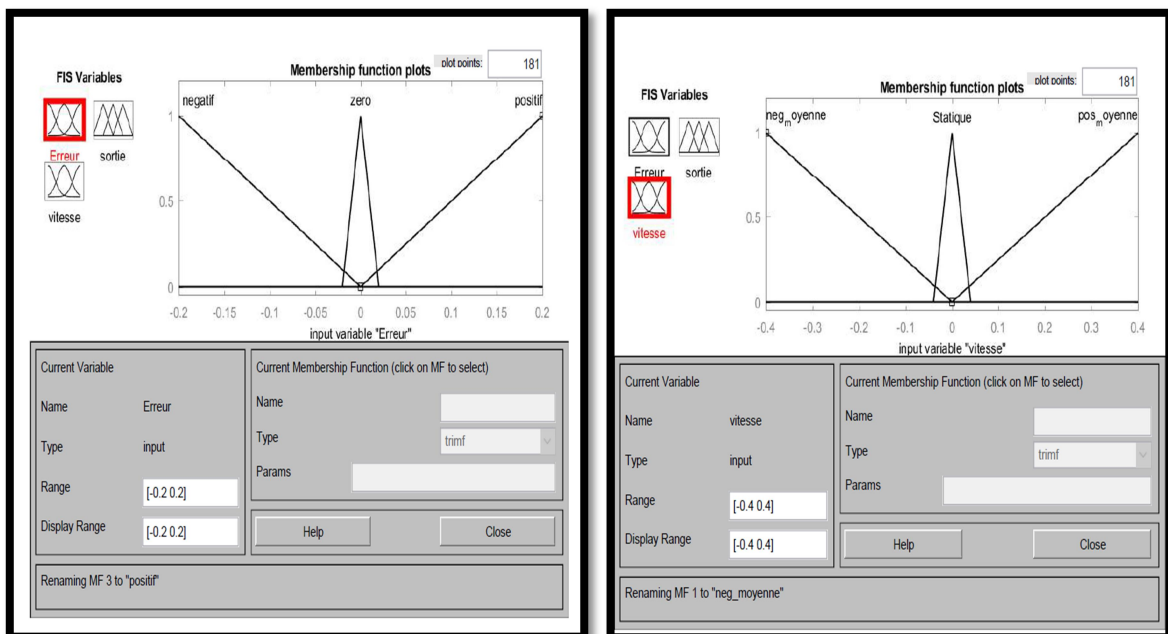


Figure III.21: Entrées du régulateur flou de l'angle phi.

✓ L'axe θ :



Figure III.22: Entrées du régulateur flou de l'angle θ .

✓ L'axe ψ :

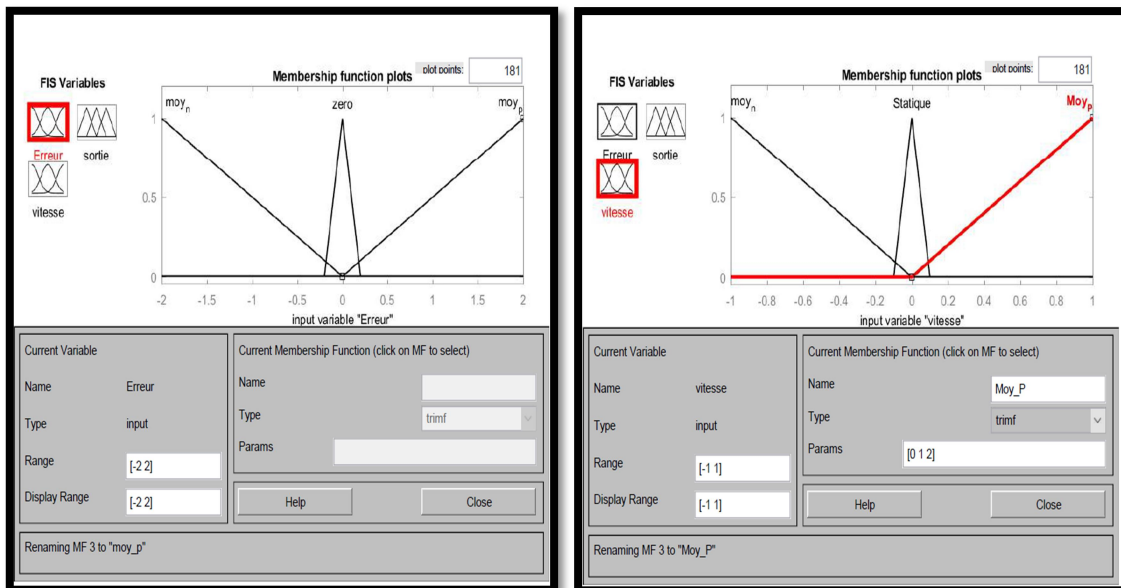


Figure III.23: Entrées du régulateur flou de l'angle ψ .

Une fois les lois de commande floues appliquées et l'inférence appliquée nous obtenons une valeur floue en sortie de chaque régulateur, nous effectuons donc une défuzzification. les figures suivantes représente les fonctions d'appartenances de sortie:

✓ L'axe φ :

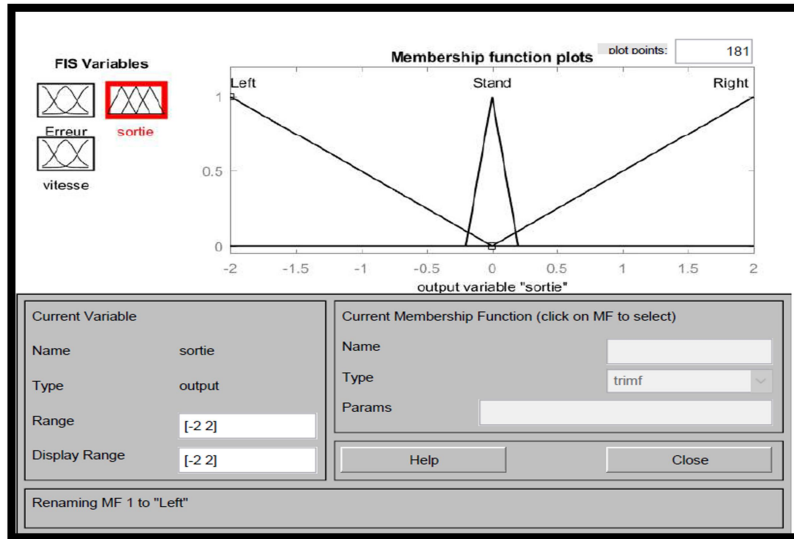


Figure III.24:Sortie du régulateur flou de l'angle phi.

✓ L'axe θ :

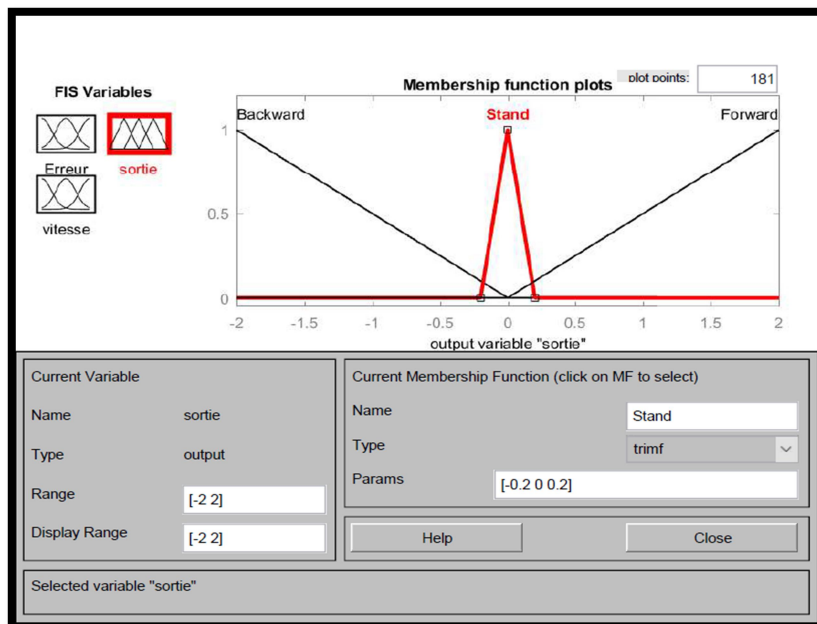


Figure III.25:Sortie du régulateur flou de l'angle theta.

✓ L'axe ψ :

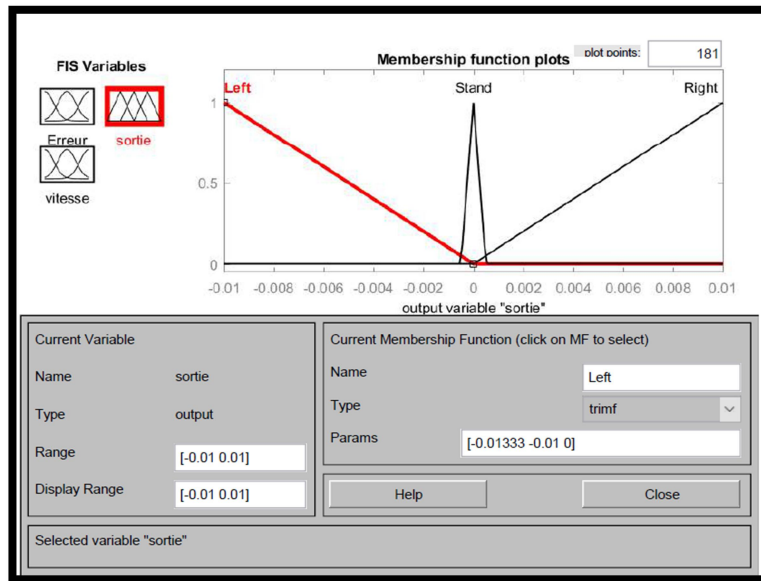


Figure III.26:Sortie du régulateur flou de l'angle psy.

✓ Schéma de régulation

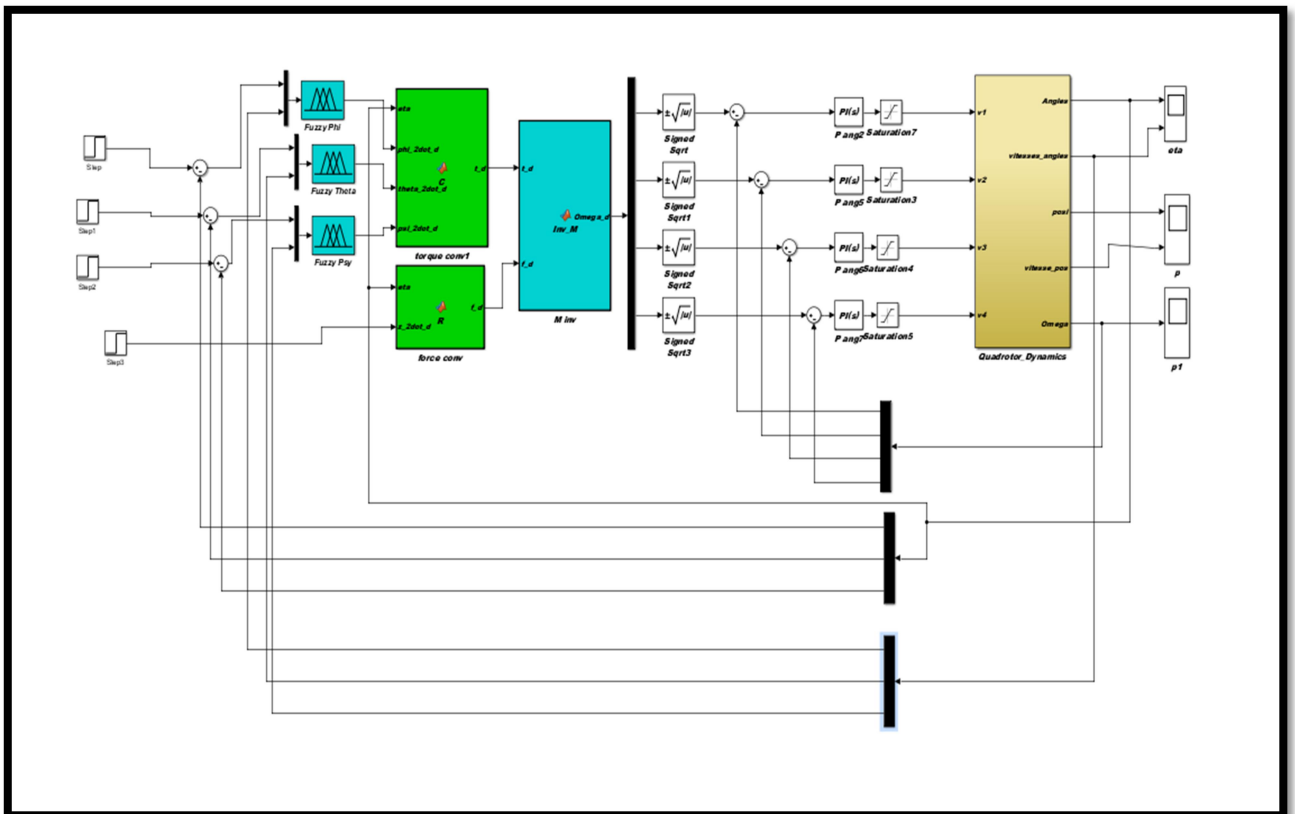


Figure III.27 : Schéma de régulation floue des angles.

Les résultats obtenus sont les suivants:

✓ Réponse à signal échelon:

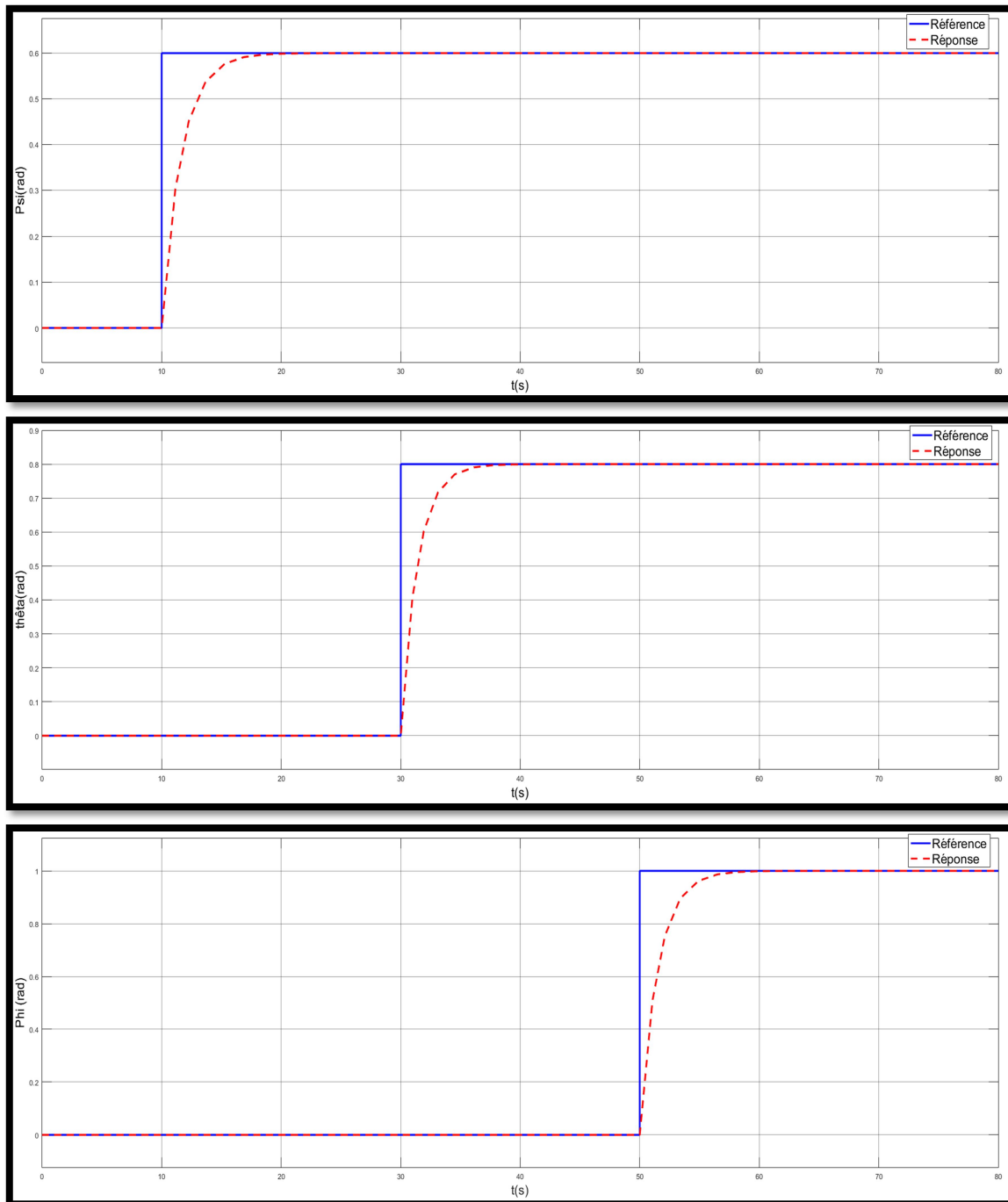


Figure III.28: réponse à un échelon de la régulation PID sur tous les angles.

En étudiant ces réponses, nous avons extrait les temps de réponses à 5% de chaque variable:

- Le temps de réponse en ϕ est de 4.63s.
- Le temps de réponse en θ est de 4.25s.
- Le temps de réponse en ψ est de 5.01s.

✓ Réponse à un signal sinusoïdal

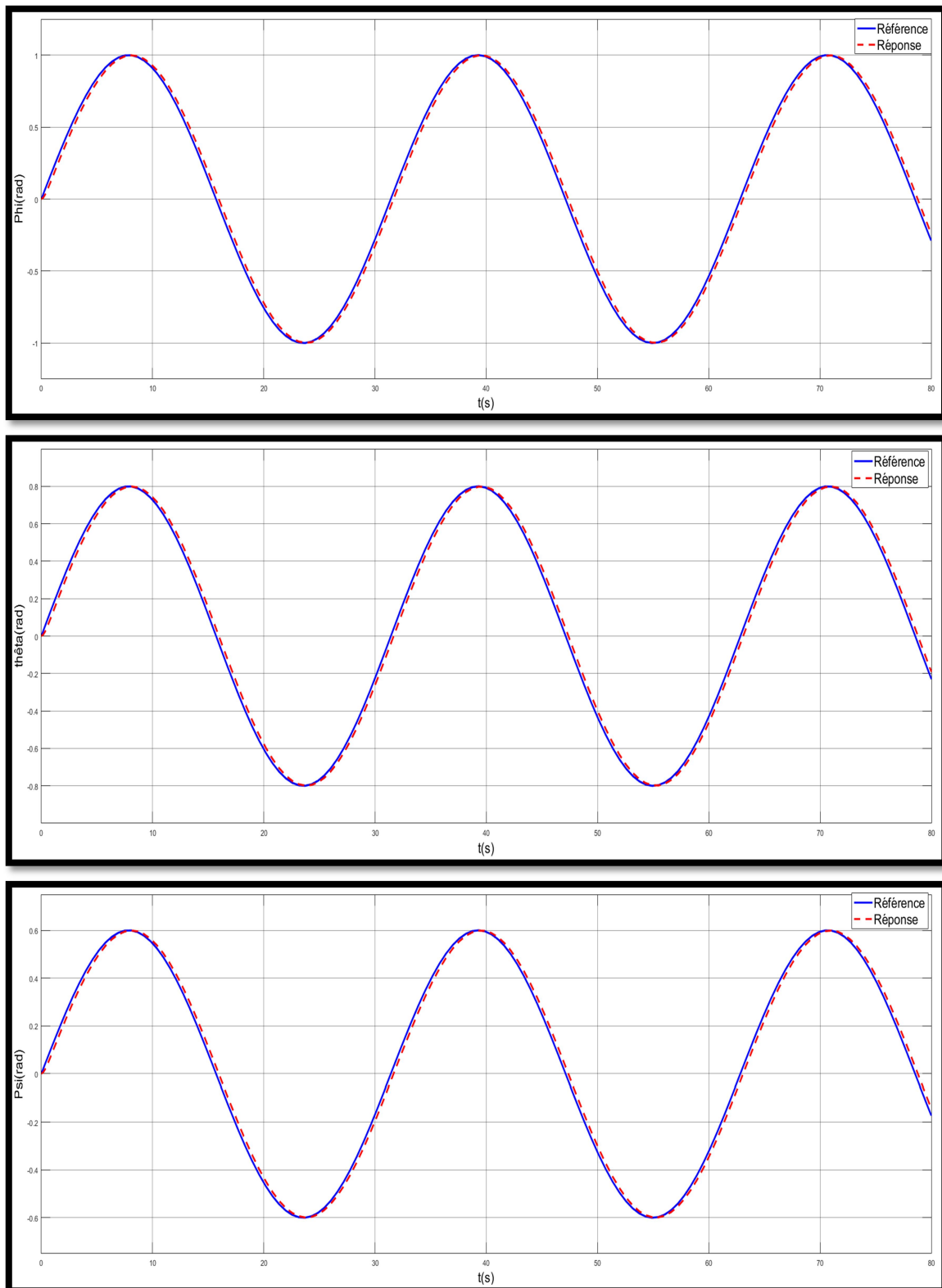


Figure III.29: réponse à un signal sinusoïdal de la régulation PID sur tous les angles.

Nous observons une bonne poursuite de trajectoire avec un léger déphasage.

✓ Régulation face aux perturbations

Pour réaliser ce dernier test nous allons fixer la consigne a une valeur constante et nous allons injecter des perturbations instantanées et a valeurs aléatoires. Les résultats sont présentés sur la figure suivante :

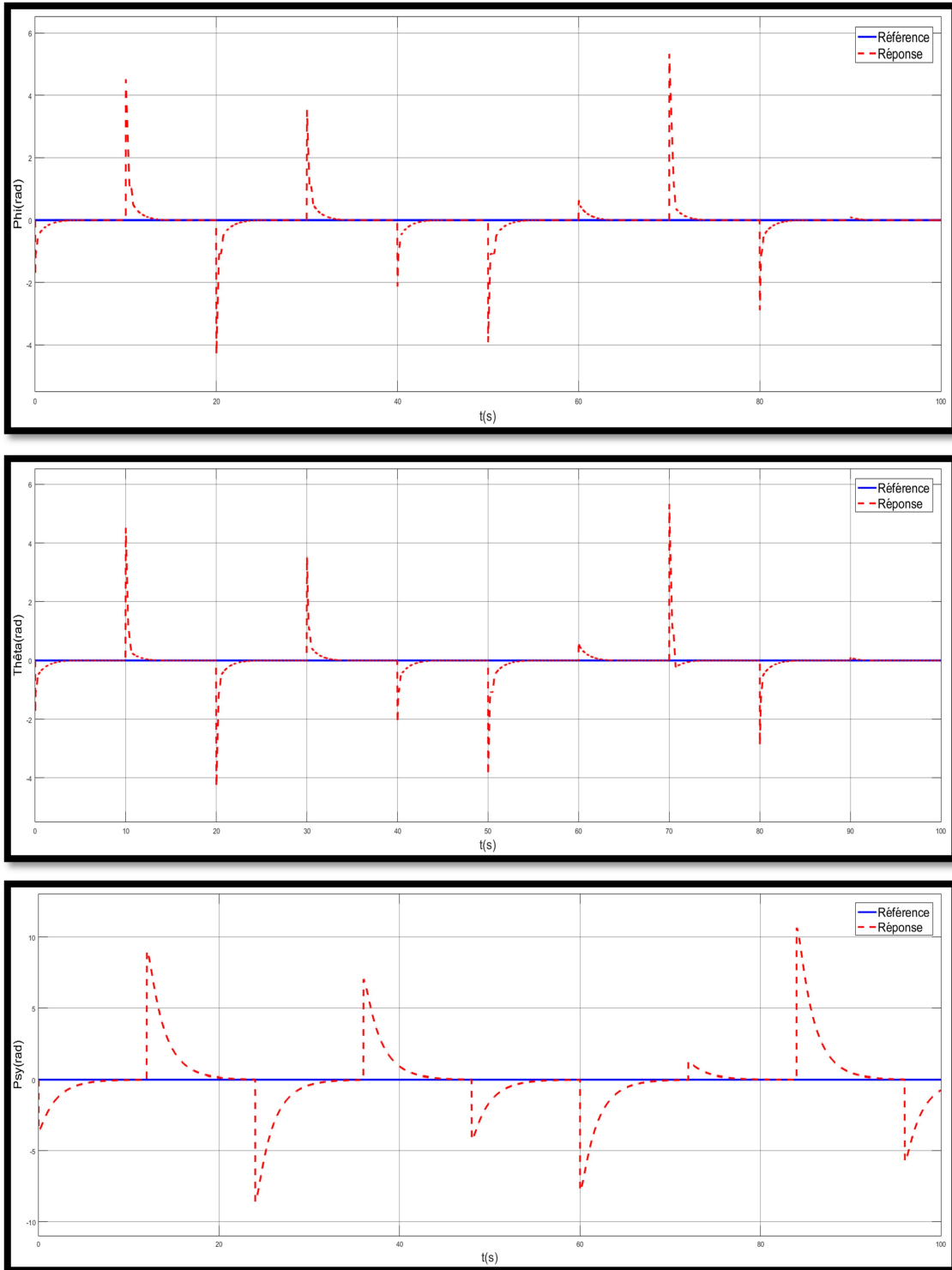


Figure III.30 : Réponse face aux perturbations de la régulation flou des états (φ θ ψ).

A partir de ses résultats nous constatons que notre système rejette les perturbations.

- Régulation des positions (x, y, z) :
- ✓ Schéma de régulation des positions

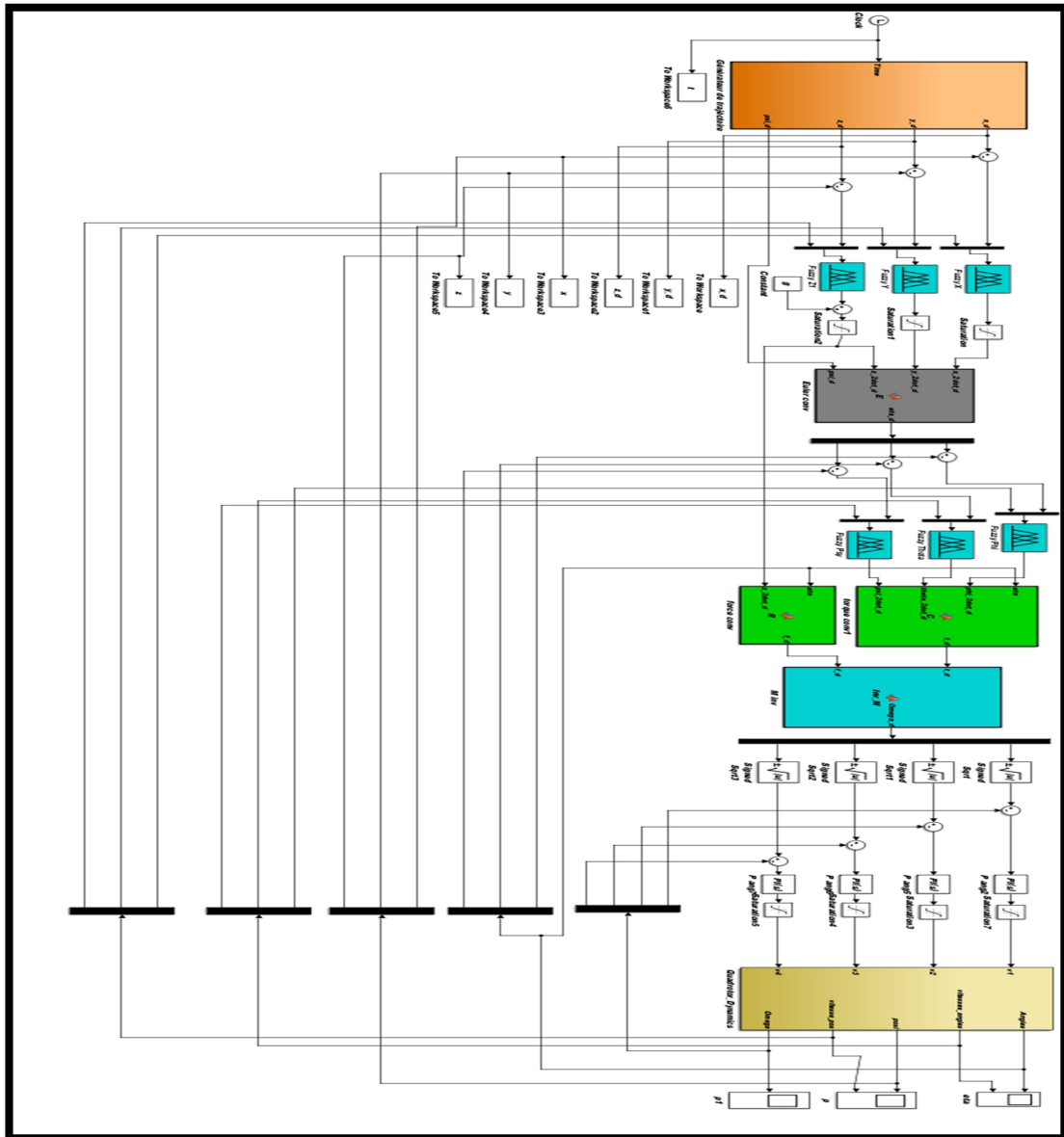


Figure III.31 : Schéma de régulation floue des positions.

Pour la régulation des positions, nous avons pris en entrée la différence entre la consigne et la position (en mètre) sur chaque axe et la vitesse de déplacement (en mètre par seconde). Après plusieurs tests et ajustage des paramètres de fonctions d'appartenance. Nous avons abouti aux fonctions d'appartenances pour les entrées représentées sur les figures suivantes:

✓ L'axe x:

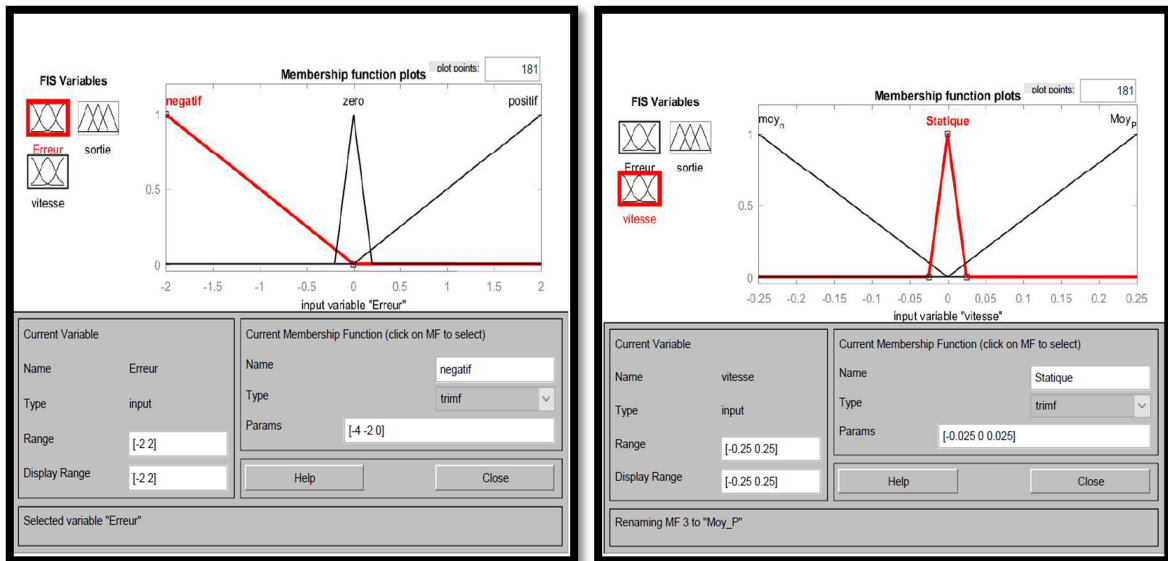


Figure III.32: Entrées du régulateur flou de la position x.

✓ L'axe y:

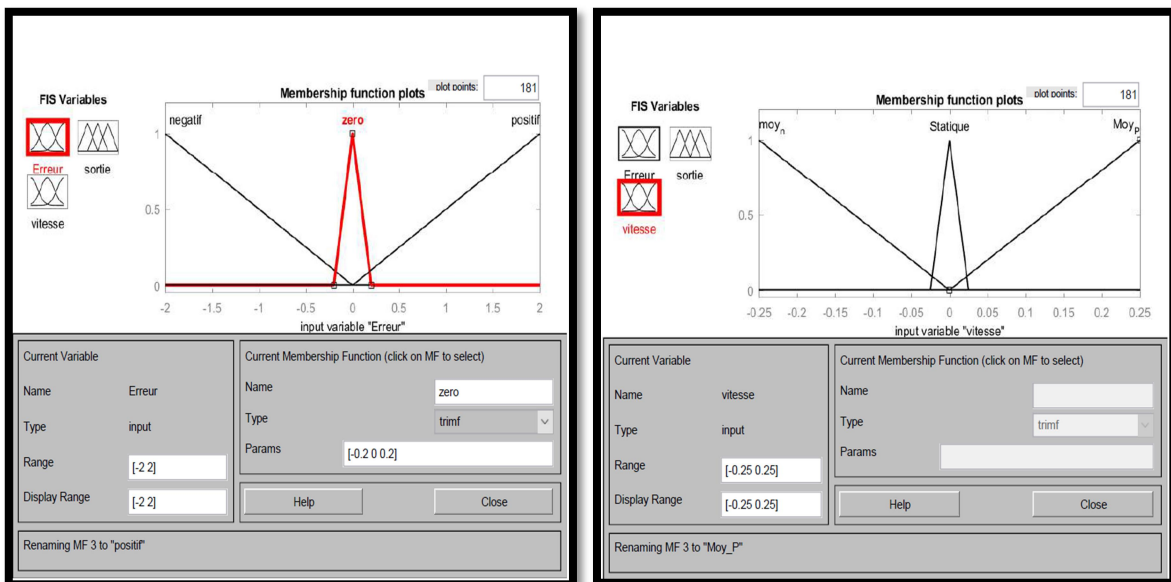


Figure III.33: Entrées du régulateur flou de la position y.

✓ L'axe z:

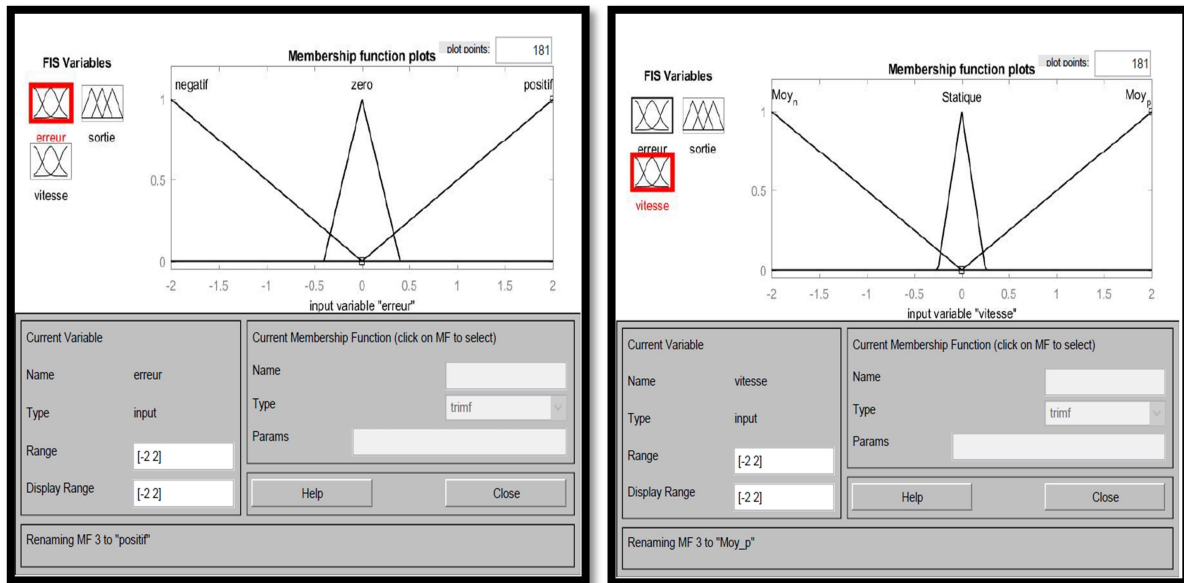


Figure III.34: Entrées du régulateur flou de la position z.

Une fois les lois de commande floues appliquées et l'inférence appliquée nous obtenons une valeur floue en sortie de chaque régulateur, nous effectuons donc une défuzzification. Les figures suivantes représentent les fonctions d'appartenances de sortie:

✓ L'axe z:

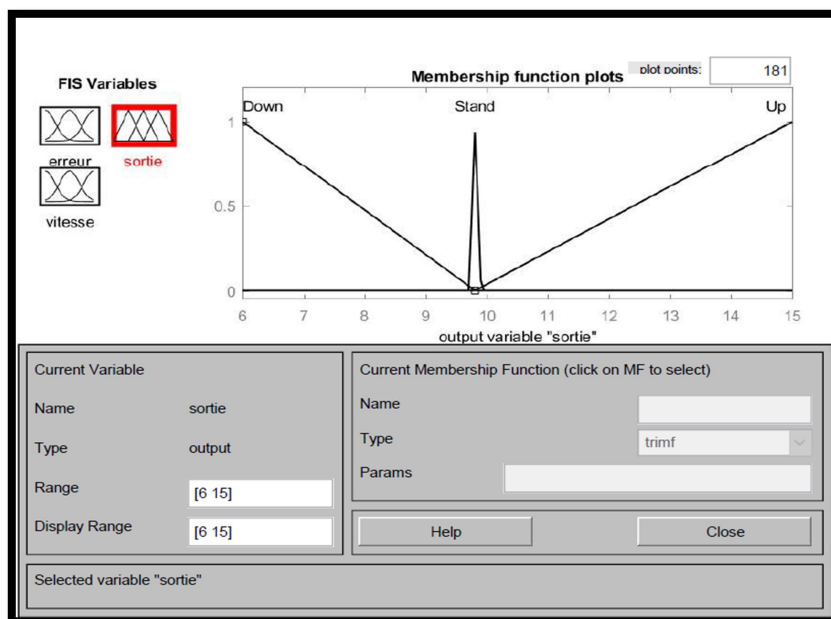


Figure III.35: Sortie du régulateur flou de la position z.

✓ L'axe x:

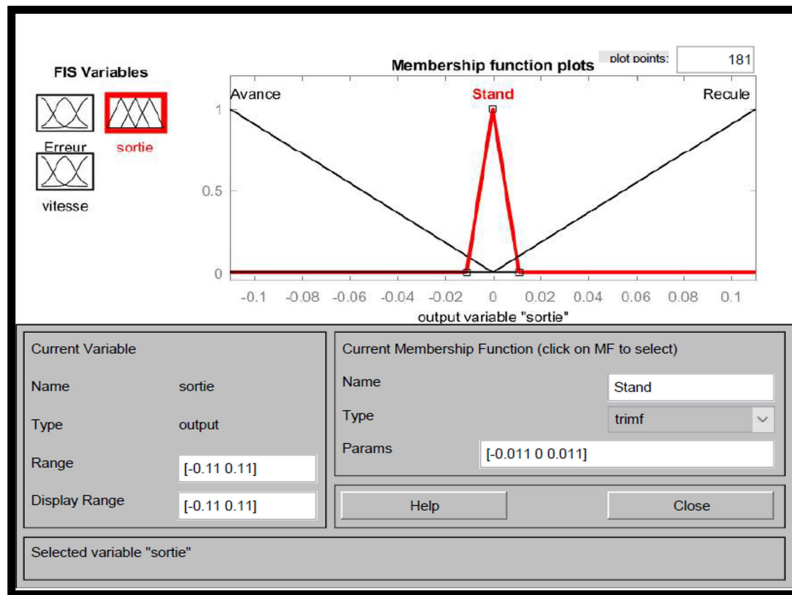


Figure III.36: Sortie du régulateur flou de la position x.

✓ L'axe y:

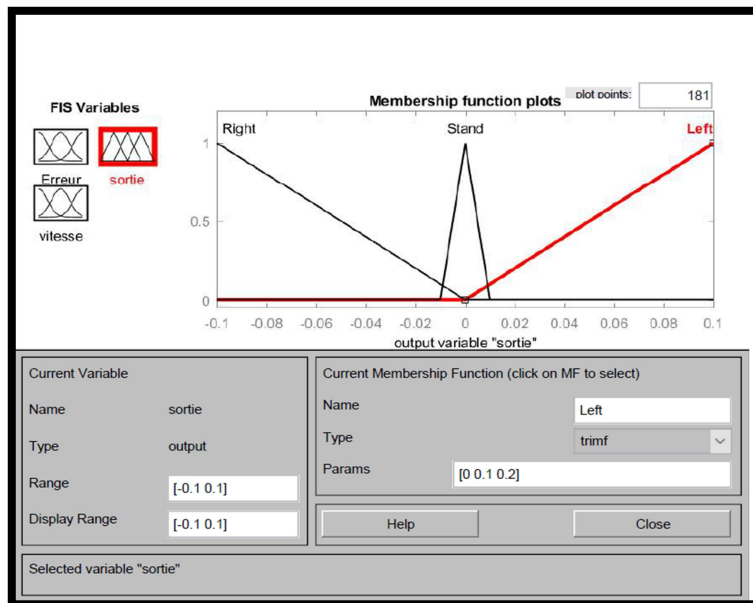


Figure III.37: Sortie du régulateur flou de la position y.

Nous avons simulé notre système avec deux trajectoires différentes et qui sont illustrées dans les figures ci-dessous:

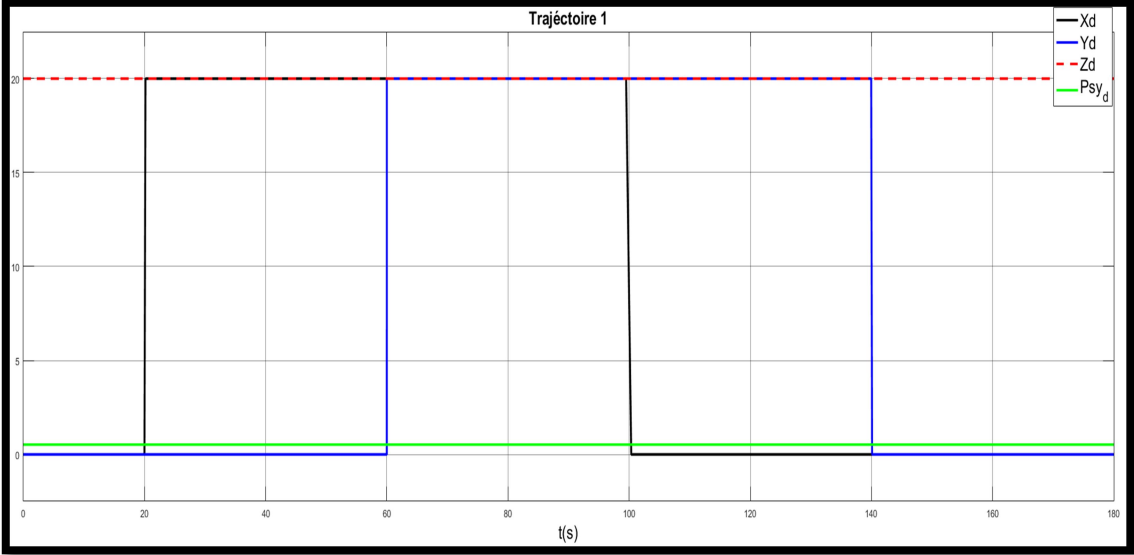


Figure III.38 : Première trajectoire désirée.

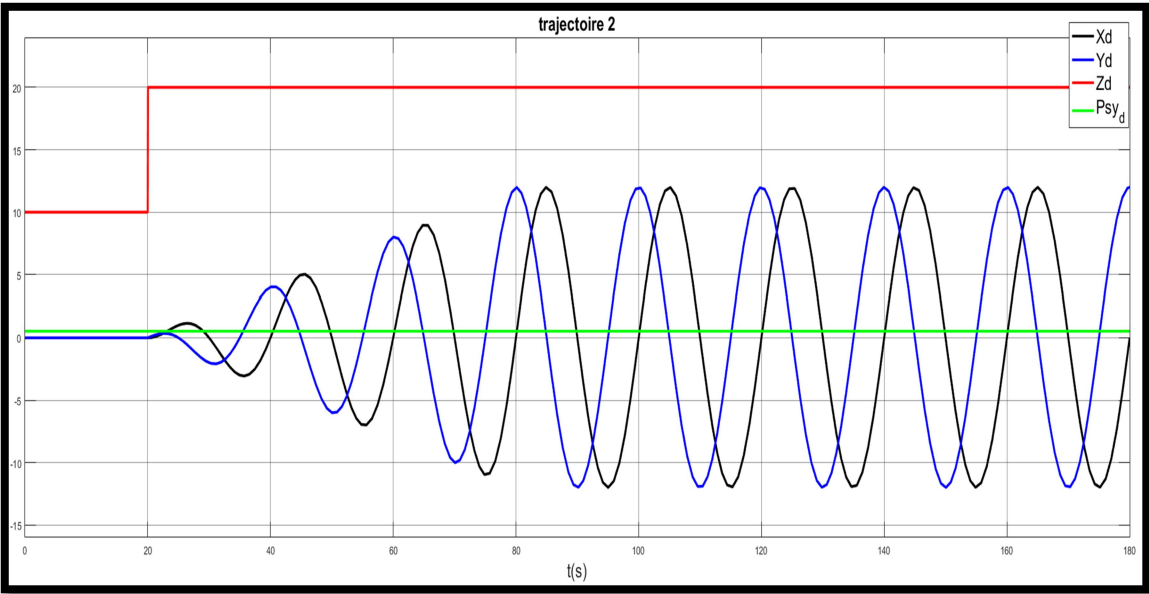


Figure III.39: Deuxième trajectoire désirée.

Les résultats de simulation obtenus sont:

✓ Pour la trajectoire 1

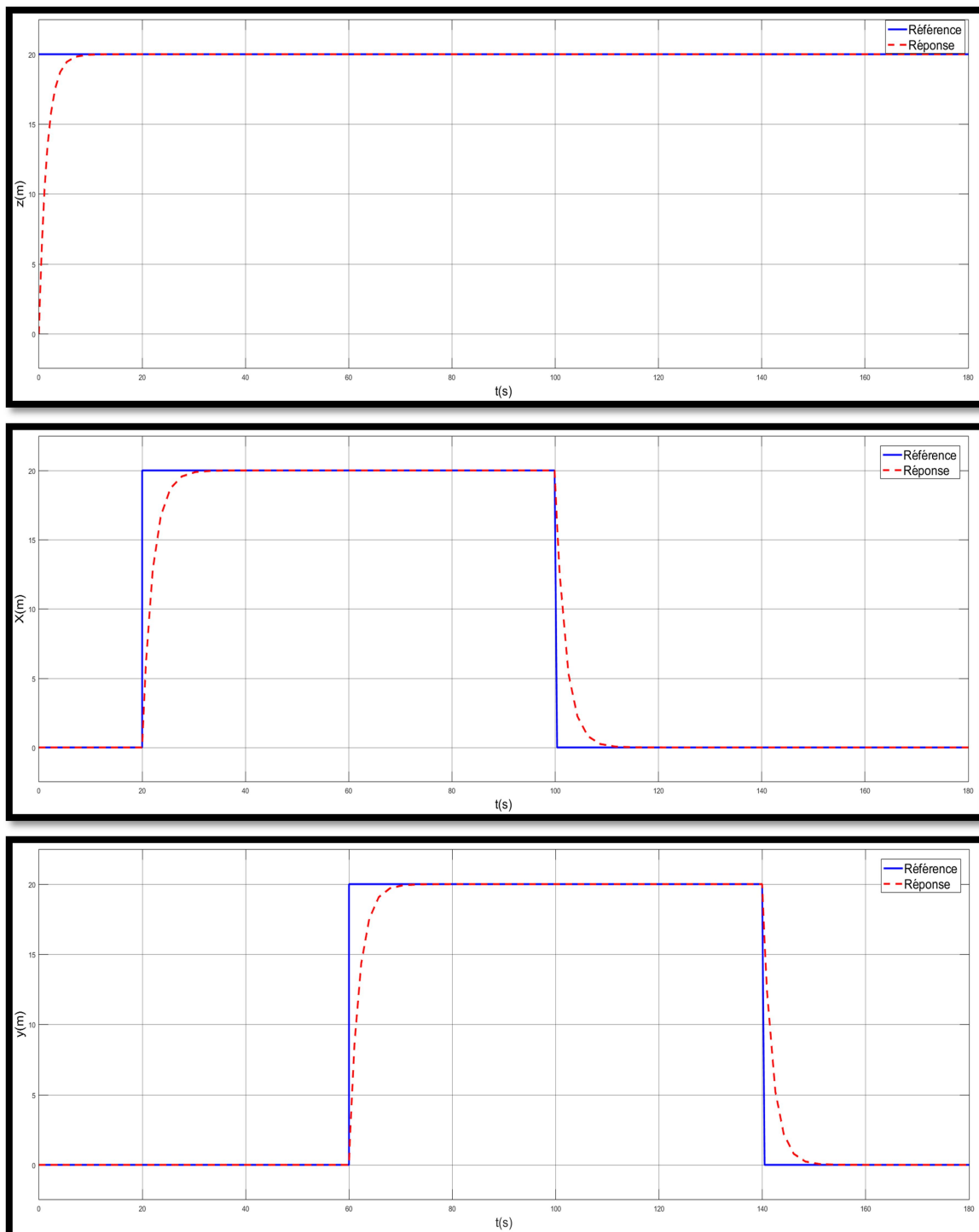


Figure III.40: Réponse des variables x , y , z à la Première trajectoire désirée.

En étudiant ces réponses, nous avons extrait les temps de réponses à 5% de chaque variable:

- Le temps de réponse en z est de 4.92s.
- Le temps de réponse en x est de 5.67s.
- Le temps de réponse en y est de 6.15s.

✓ Trajectoire effectuée donc par l'UAV en 3D:

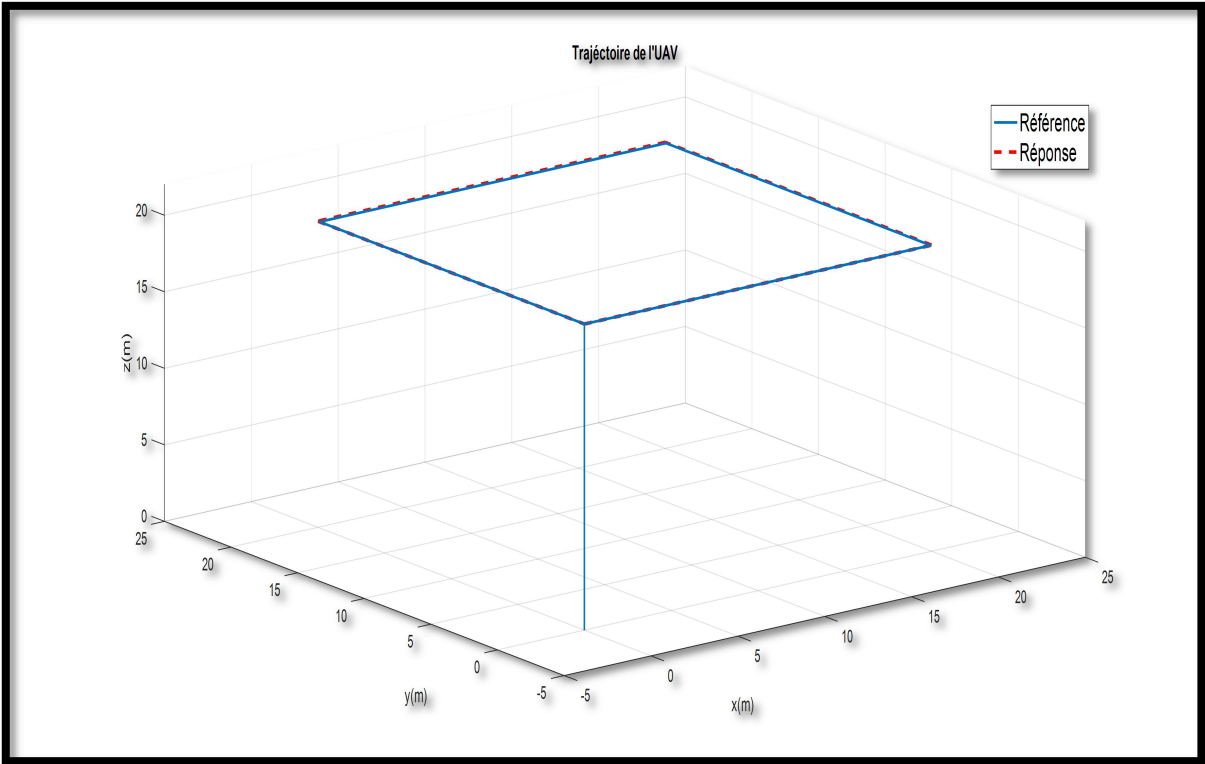


Figure III.41: Trajectoire de l'UAV en 3D.

✓ Pour la trajectoire 2

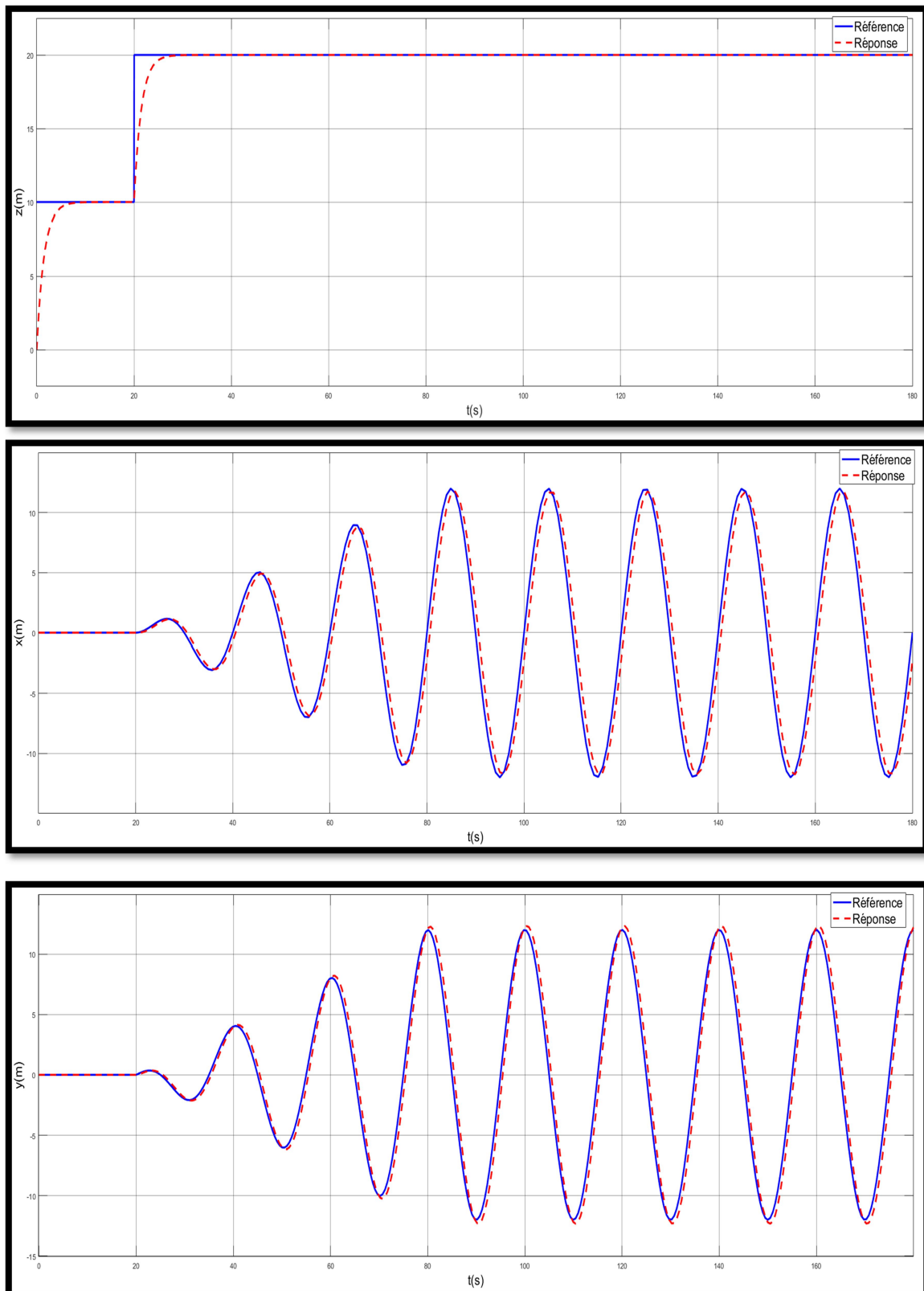


Figure III.42 : Réponse des variables x , y , z à la Deuxième trajectoire désirée.

Nous observons une bonne poursuite de la consigne avec un léger déphasage pour les réponses x et y .

- ✓ La trajectoire effectuée donc par l'UAV en 3D:

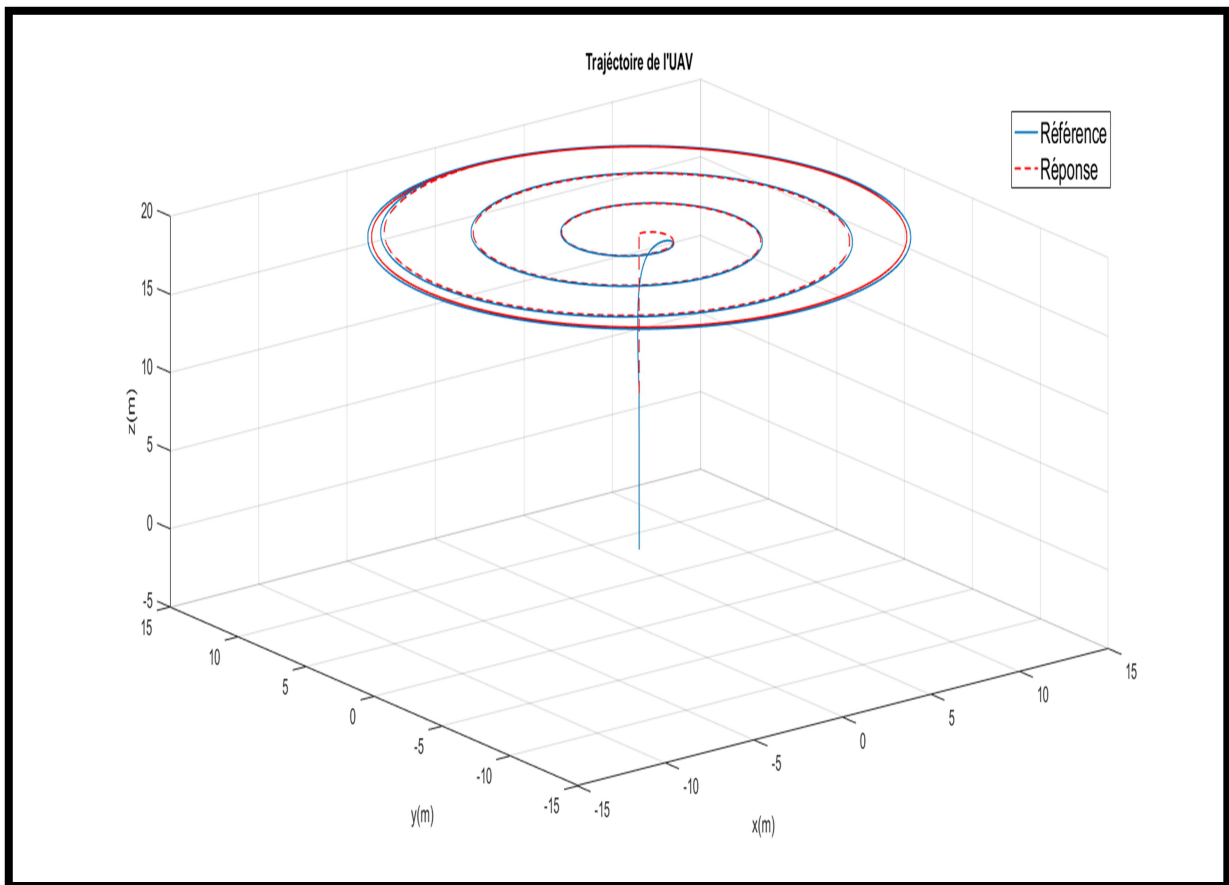


Figure III.43 : trajectoire de l'UAV en 3D.

- ✓ Régulation face aux perturbations

Pour réaliser ce dernier test nous allons fixer la consigne a une valeur constante et nous allons injecter des perturbations instantanées et a valeurs aléatoires. Les résultats sont présentés sur la figure suivante :

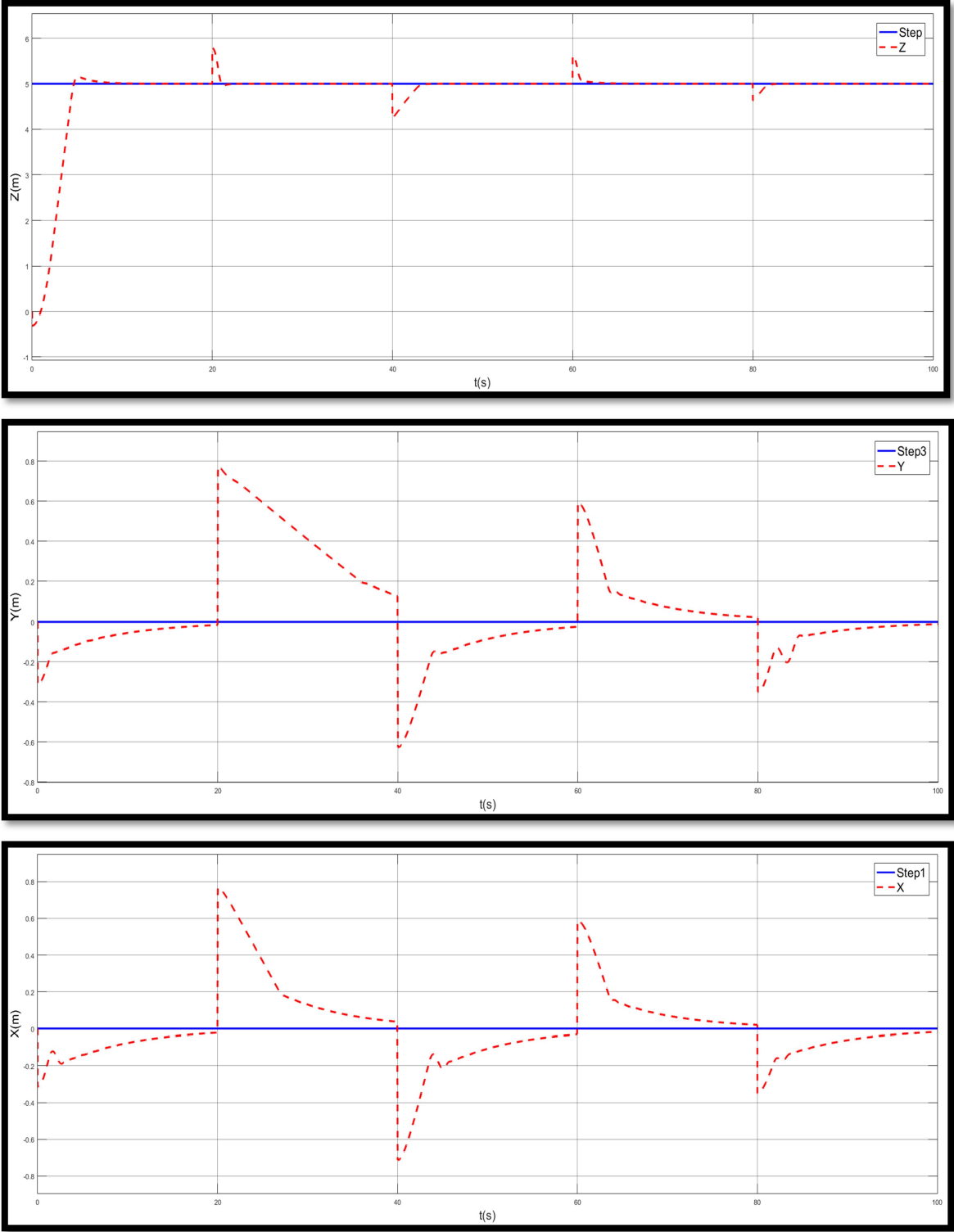


Figure III.44. : Réponse face aux perturbations de la régulation floue des etats x y z.

Nous pouvons voir que notre système rejette clairement les perturbations.

III.4. Réalisation d'une Neuro-floue (ANFIS) pour le contrôle d'altitude de l'UAV

Dans cette section nous allons synthétiser et appliqué une commande neuro-floue type ANFIS pour Réguler l'état Z de notre système en utilisant la toolbox de Matlab Neuro-Fuzzy Designer accessible en saisissant la commande *anfisedit* dans le Workspace.

✓ Présentation de l'Interface Graphique :

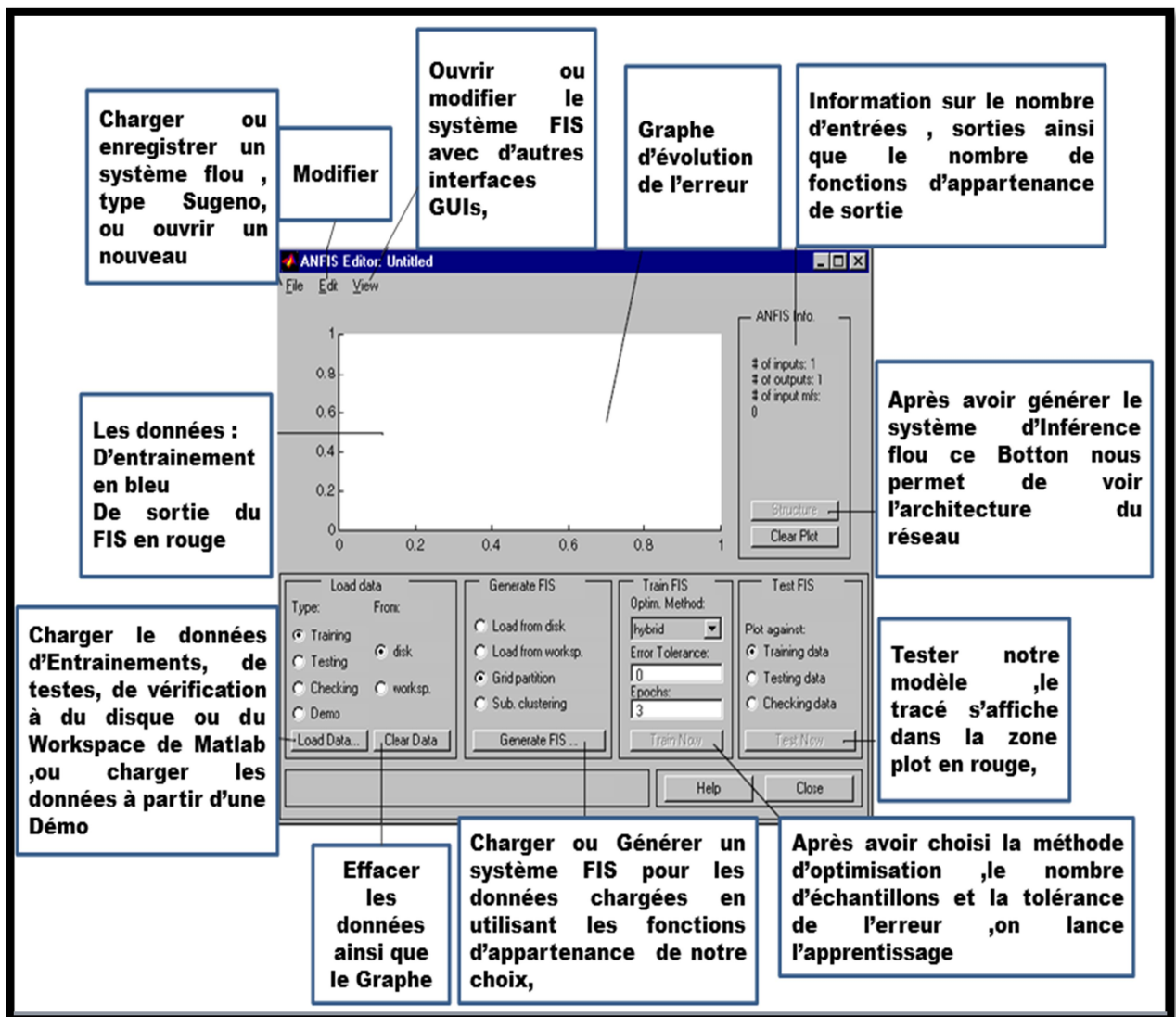


Figure III.45 : Présentation de l'interface Neuro-Fuzzy Designer.

En premier lieu, nous devons récupérer les données d'apprentissage avec lesquelles nous entrainerons notre réseau, ces données seront générées par un expert, ce rôle sera joué par le régulateur flou de l'état Z conçu précédemment, ces données seront envoyées vers le Workspace sous forme d'une matrice de valeurs de $n+1$ colonnes (n : nombre d'entrées de notre régulateur) et la dernière colonne représente les valeurs de sortie de notre régulateur.

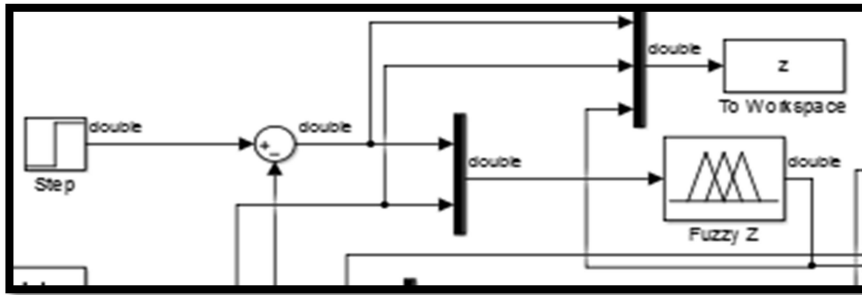


Figure III.46 : Extraction des données d'apprentissage.

Puis, charger ces données dans le Neuro-Fuzzy Designer en appuyant sur le bouton 'Load Data...' et en choisissant type 'training' et from 'Workspace'.

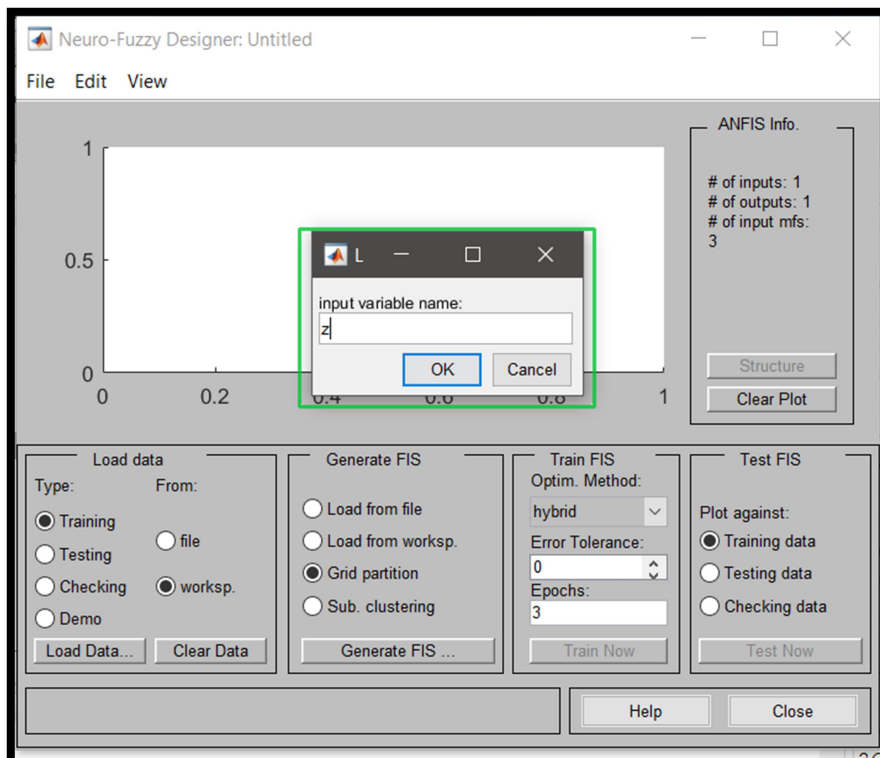


Figure III.47 : Chargement des données d'apprentissage.

Ensuite, générer le Système d'Inférence Flou initial en appuyant sur le bouton 'Generate FIS.'. Nous avons choisi sept fonctions d'appartenance pour chaque entrée de type Triangulaire, et opté pour une fonction d'appartenance de la sortie de type linéaire.

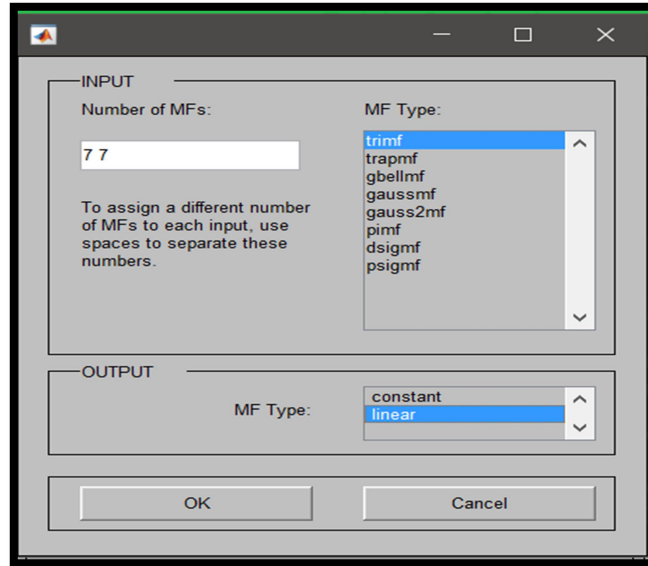


Figure III.48 : Génération du Système flou.

Et enfin lancer l'apprentissage en appuyant sur 'train Now' tout en choisissant la méthode d'optimisation 'Hybride', une tolérance à l'erreur de 0 , ainsi que le nombre d'Epochs à 70% .

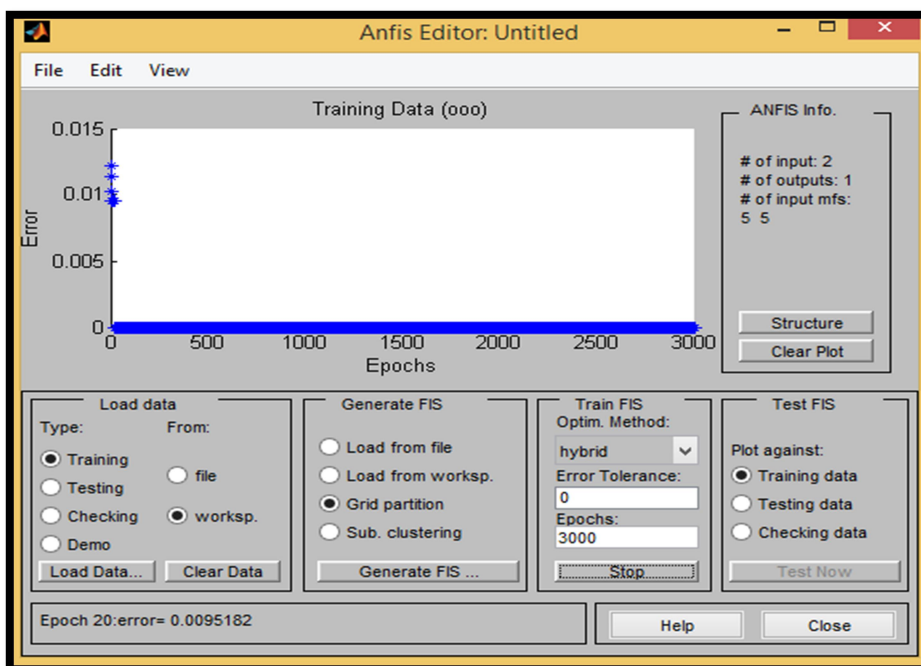


Figure III.49 : Apprentissage.

Une fois l'apprentissage terminé nous testons notre Système en Appuyant sur 'Test Now'.

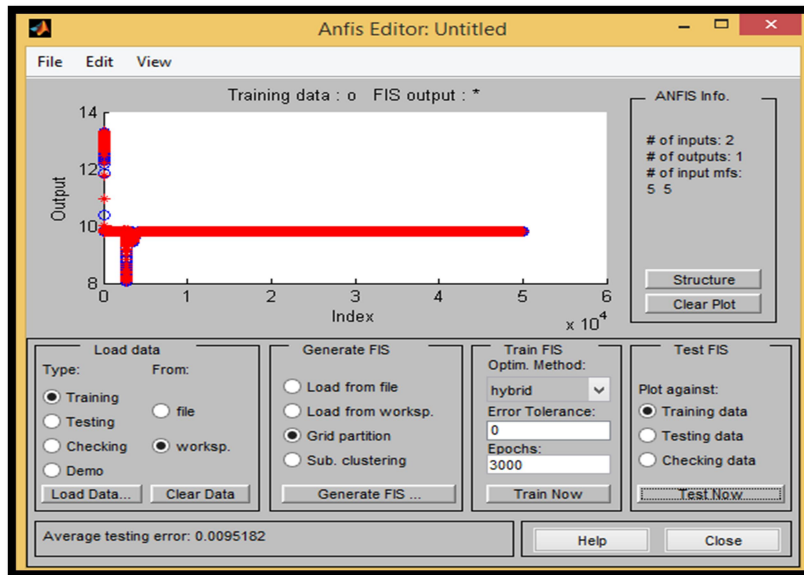


Figure III.50 : Test du Système flou.

Une fois le système est testé, on enregistre en exportant le fichier comme illustre la figure ci-dessous :

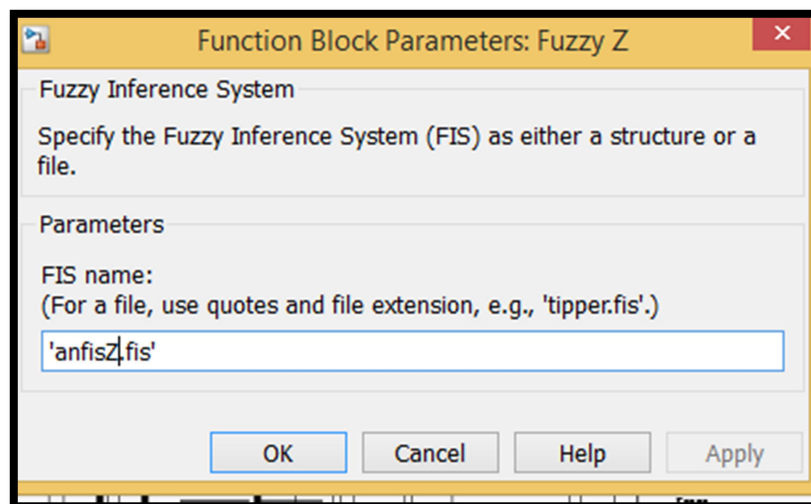


Figure III.51 : Exportation du fichier.

- Informations sur le système obtenu :
 - ✓ Nombre de nœuds : 131.
 - ✓ Nombre des paramètres linéaires : 147.
 - ✓ Nombre des paramètres non linéaires : 42.
 - ✓ Total nombre des paramètres : 189.
 - ✓ Nombre de paires de données d'entraînement : 4500.
 - ✓ Nombre de règles floues : 49.
- Architecture du Réseau :

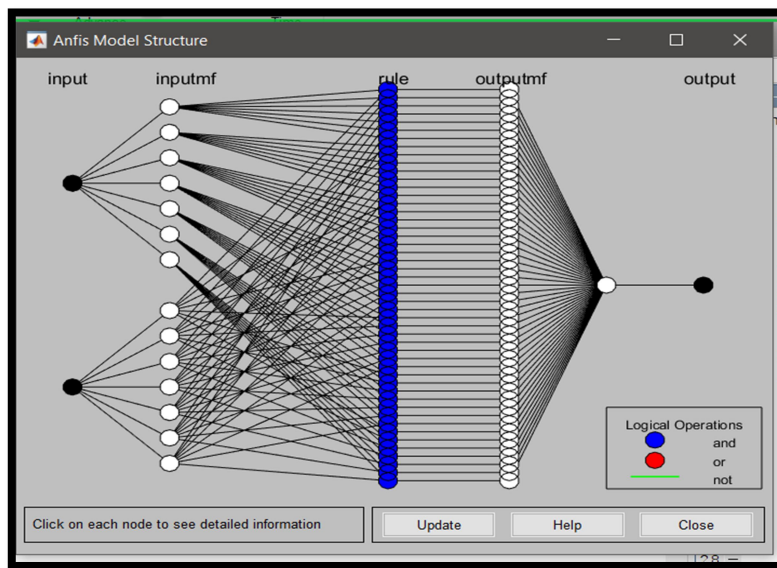


Figure III.52 : Architecture du Réseau.

- Surface du Régulateur :

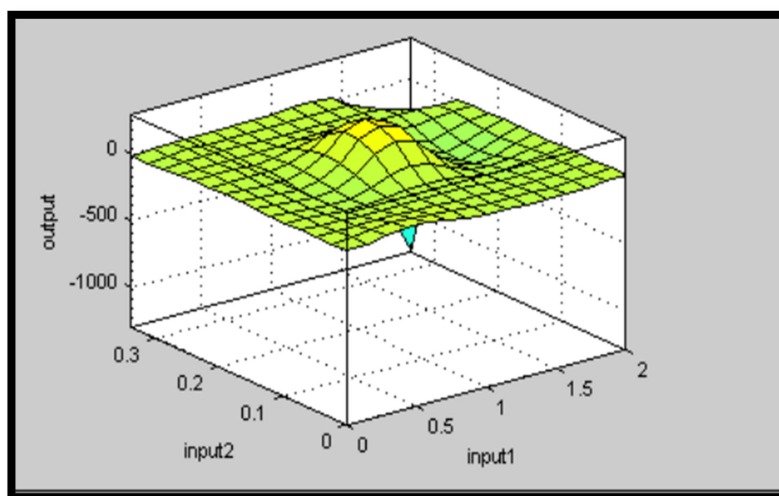


Figure III.53 : Surface du Régulateur.

- **Résultat obtenu :**

Le Régulateur de référence :

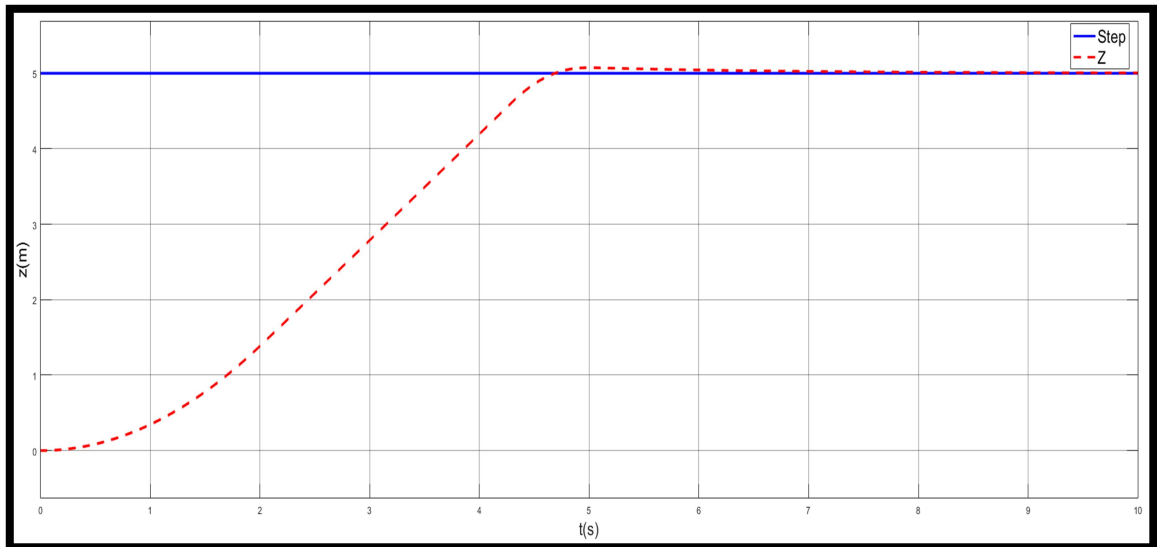


Figure III.54 : Réponse de Z à un signal échelon (Mamdani).

- ✓ Temps de réponse à 5% est de 4.42 s.
- ✓ Présente un léger dépassement.

Le régulateur ANFIS :

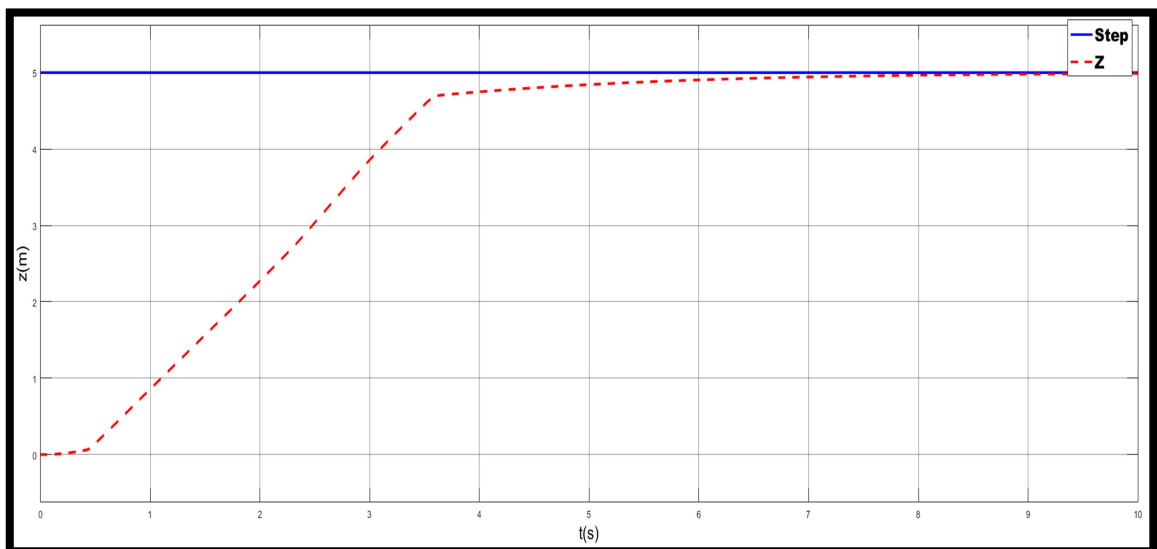


Figure III.55 : Réponse de Z à un signal échelon (ANFIS).

- ✓ Temps de réponse à 5% est de 3.80 s.
- ✓ Ne présente aucun dépassement.

III.5 Conclusion

En premier lieu, nous avons synthétisé une commande PID pour réguler notre système, nous avons opté pour une régulation à trois boucles. La première comporte quatre régulateurs PI pour la régulation de vitesse des quatre moteurs ; nous avons présenté les résultats obtenus et nous avons testé le comportement du modèle. La deuxième contient trois régulateurs PID des angles et les résultats obtenus sont jugés acceptables pour la poursuite d'un signal échelon, sinusoïdal et même pour le rejet de perturbations. La troisième comporte trois régulateurs PID de positions, les résultats obtenus sont jugés acceptables pour la poursuite d'un signal échelon, sinusoïdal et même pour le rejet de perturbations.

En second lieu, nous avons synthétisé une loi de commande floue de type Mamdani pour réguler notre système. Nous avons d'abord réalisé des commandes pour les angles qui ont donné des résultats amplement satisfaisants, autant en asservissement et poursuite de consigne et même en matière de robustesse. Puis nous avons réalisé des commandes pour les positions qui ont fourni des résultats meilleurs que ceux obtenus avec la commande PID en matière de rapidité et robustesse, cependant en matière de précision la commande floue présente de meilleurs résultats lors de la poursuite d'un signal échelon en outre pour ce qui est de la poursuite d'un signal sinusoïdal, la commande PID montre une meilleure poursuite de la consigne.

Enfin, Nous avons conçu un régulateur neuro-flou pour le control de l'altitude pour notre système, nous avons constaté qu'avec cette commande le temps de réponse est plus rapide que les deux commandes citées au précédemment sans aucun dépassement. Cependant, le régulateur neuro-flou donne des résultats uniquement pour la consigne avec laquelle nous avons fait l'apprentissage.

Conclusion générale

Le travail présenté dans ce mémoire concerne la modélisation et la commande d'un drone de type quadrirotor dans le but de le stabiliser.

Dans ce travail, nous avons donné un aperçu général sur les drones, leurs classifications ainsi que leurs domaines d'application. Ensuite, nous avons décrit leur principe de fonctionnement.

Par la suite, nous avons pu aboutir à un modèle dynamique du quadrirotor grâce à l'étude des équations agissant sur le système et ce en utilisant le formalisme de Newton-Euler qui décrit la dynamique de rotation et de translation d'un corps rigide. Nous avons constaté que le modèle obtenu présente de fortes non linéarités.

Puis, nous avons présenté le principe des commandes classique et intelligentes dans le but de stabiliser et réguler notre système. Le modèle du quadrirotor a servi alors à la synthèse de trois lois de commande.

Ces commandes ont été implémentées et simulées sous Matlab-Simulink. Les résultats de simulation du système piloté par ces lois de commande montrent des réponses satisfaisantes en termes de rapidité, précision et robustesse. Cependant, la commande floue s'avère plus performante en termes de rapidité et de robustesse quant à la précision, la commande par PID montre des résultats meilleurs.

En guise de perspectives à ce travail, il serait intéressant de faire un apprentissage assez minutieux en utilisant plusieurs consignes de références pour tous les états du système ou bien avoir des données expérimentales en réalisant un prototype. Le choix de l'expert doit être un contrôleur non linéaire, robuste présentant les meilleurs résultats possibles (commande par backstepping, retour d'état linéarisant...etc.).

Bibliographie

- [1] Samir Bouabdallah, thèse de doctorat de l'école polytechnique fédérale de LAUSANNE "design and control of quadrotors with application to autonomous flying" 2007.
- [2] 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.
- [3] T. Hamel, R. Mahony, R. Lozano, J. Ostrowski "Dynamic modelling and configuration
stabilization for an x4-flyer.", 15th Triennial World Congress, Barcelona, Spain, 2002.
- [4] ADRIEN DROUOT, thèse doctorat de l'université de lorraine en automatique 'stratégies de commande pour la navigation autonome d'un drone projectile miniature'.2013
- [5] DAMIEN POINSOT, thèse de doctorat de l'ONERA "commande d'un drone en vue de la conversion vol rapide-vol stationnaire"2008.
- [6] DUC KIEN PHUNG, thèse de doctorat de l'université de pierre et marie curie en robotique "conception, modélisation et commande d'un mini drone convertible".2015
- [7] Ertugrul Cetinsoy, Cevdet hancer, Kaan T.oner, Efe sirimoglu, and Mustafa Unel 'Aerodynamic design and characterization of a quad Tilt-Wing UAV via Wind tunnel tests" 2012.
- [8] Menno HOCHSTEN bach, cyriel notteboom, bart theys and joris de schutter 'design and control of unmanned aerial vehicle for autonomous parcel delivery

with transition from vertical take-off to forward flight-vertikul, a quadcopter tailsitter' KU leuven, belgium.2015.

[9] JEAN-YVES ANDRO, thèse de doctorat de l'université de Poitiers en mécanique des milieux fluides 'aérodynamique d'un profil d'aile battante a bas nombre de Reynolds'.2008

[10] www.locadron.com/champ-daplication-drone/

[11] www.dronedecole.fr/différentes-utilisation-drones/

[12] www.lefigaro.fr/secteur/high-tech/2016/12/14/ .

[13] A.TAYBI, 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.

[14] A .TAYBI, and S MCGILVRAY 'Attitude stabilisation of a VTOL Quadrotor Aircraft', IEEE Transactions on control systems Technology, Vol 14,No 3, pp. 562-571, mary 2006.

[15] MOKHTARI MOHAMMED RIDA , these de doctorat de l'universite d' A BOUBEKR BELKAID TLEMCEN en automatique 'observation et commande de drones miniatures a voilures tournantes .2015

[16] L.GAUTIER , B.Hamilton, J. Hazebrouck, and T.Tourrette 'pilotage d'un quadri -rotor via un FPGA',tuteur ESIEE engineering, journee pedagogique, 2008.

[17] H.Bouadi, M.Bouclouche, M.Tadjine " modeling and stabilizing control laws designe based on sliding Mode for an UAV type quadrotor.engineering letters,15 :2,el_15_2_24 .november 2007.

[18] Khebbache Hicham, thèse de magister en automatique de l'université de FERHAT ABBAS DE SETIF « Tolérance aux défauts via la méthode de

backstepping des systèmes non linéaires sur un système UAV de type quadrirotor ».2012.

[19] Bellahcene Zakaria, thèse de magister en automatique de l'université des sciences et de la technologie d'Oran Mohamed Boudiaf "synthèse de loi de commande robuste pour un hélicoptère à quatre hélices"2013.

[20] Asma. Ammar Boudjellal , Farés.Boudjema « commande par backstepping basée sur un Observateur mode glissant pour un drone de type Quadri-rotor »

[21] H.Bouadi, M.Bouclouche, M.Tadjine 'modeling and stabilizing control laws design based on backstepping for an UAV type quadrotor.

[22] P.Pounds, R.Mahony&J.Roberts "design of four-rotor Aerial robot" Australasian conference on robotics and automation auckland, 27-29 novembre 2002.

[23] Ferhat Lehouazi, mémoire de magister en automatique de l'université de tizi-ouzou" mise en œuvre d'une stratégie de commande neuro floue: application à un pendule inversé."

[24] Laurant F. sylvie G. Anfre T. Hermès sciences publication, paris 2003"commande floue 1 et 2".

[25] L.A ZADEH "quantitative fuzzy semantics".information science.

[26] El-Hadi Rebaa, thèse de doctorat de l'université paris XII Val de Marne: "Génération automatique et optimisation de systèmes à inférence flou".

[27] Kevin PASSINO & Stephen YURKOVICH, Livre: "Fuzzy Control".

- [28] Kadda ZEMALACHE MEGUENNI, Thèse de doctorat en robotique de l'université d'Every val D'essonne: "commande d'un système sous actionné ; application à un drone à quatre Hélices".
- [29] Abdallah BEZZINI, Thèse de magister en automatique de l'université Mohamed khider de biskra: "commande prédictive non linéaire en utilisant les systèmes neuro flous et les algorithmes génétiques 2013"
- [30] Talbi Nesrine, thèse de doctorat en électronique de l'université Constantine 1: "Conception des systèmes d'inférence flou par des approches hybrides".
- [31] CHAPMAN & HALL, Livre: "A First Course In Fuzzy and Neural Control".
- [32] Lefteri H. Tsoukalas et Robert E.Uhrig, livre: "Fuzzy and Neural approaches in Engineering".
- [33] PATRICE WIRA, Rapport d'habilitation de l'université de Haute Alsace: "approches neuro mimétiques pour l'identification de la commande 2009".
- [34] Yung c. Shin et Chengying Xu, livre: "Intelligent systems; Modeling, Optimisation and control".
- [35] Bendaas Ismail, thèse de doctorat en génie électrique de l'université de Batna2: "Contribution à la commande hybride par mode glissant floue appliquée à un moteur à induction".
- [36] MAGALY DE PAULA CANUTO, (2001), combining neural networks and fuzzy logic for applications in character recognition. A thesis submitted to the university of kent at canterbury for the degree of doctor philosophiy in the subject of Electronic engineering.

[37] Mordjaoui M Thèse de doctorat de l'université de savoie,France 1997.

"Modelisation des effets électromagnetiques Apport de la logique floue et neuro-floue".

[38] J.jang C.T sun."fonctional equivalence between radial basis function network and fuzzy inference system iee transaction on neural network.1993".