

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

UNIVERSITE MOULOUD MAMMARI DE TIZI-OUZOU



FACULTE DE GENIE DE LA CONSTRUCTION
DEPARTEMENT DE GENIE MECANIQUE – CAMPUS TAMDA 2

Mémoire de Fin d'Etudes De MASTER ACADEMIQUE

Domaine : Sciences et Technologies

Filière : Electromécanique

Spécialité : **MAINTENANCE INDUSTRIELLE**

Présenté par

Mlle FERRAT Kenza

Mlle SADOUK Ouarda

Thème

**Réalisation d'un banc d'essai pour un
drone à un seul axe**

Encadré par : Dr Si Ahmed

Mr YAMANI Nourdine
M C B, UMMTO, Président

Mr BELGAID Houcine
M C B, UMMTO, Examineur

Mr SI AHMED Hamid
M A A, UMMTO, Encadreur

Promotion : 2020/2021



REMERCIEMENTS

On dit souvent que le trajet est aussi important que la destination. Ces cinq années nous ont permis de bien comprendre la signification de cette phrase toute simple. Ce parcours en effet, ne s'est pas réalisé sans défis et sans soulever de nombreuses questions pour lesquelles les réponses nécessitent de longues heures de travail.

Nous remercions Dieu « ALLAH » tous puissants qui nous a donné le courage de confronter ce nouveau domaine, et la force de réaliser ce travail.

On tiens à remercier nos parents de nous avoir soutenu et croient en nous lorsque nous nous savions rien et L'orsque on saura rien un jour , on serai jamais assez reconnaissant envers eux

On adresse nos remerciement , tout particulièrement, à notre encadreur Mr. SI AHMED Hamid qui nous a aidé beaucoup dans la réalisation de notre drone et ses précieux conseils .

Veillez monsieur croire à l'expression de notre profond respect .

On remercie égalemnt tous les membres du jury qui on accepté de nous honorer de leur présence et de juger notre travail.

On remercie aussi toute personne ayant contribue à notre soutiene moral ainsi que toutes nos familles .

Merci a tous

Mme FERRAT KENZA

Mme SADOUK Ouarda

Dédicaces

Je dédie ce mémoire

Ames chers parents. Ma mère, qui m'a encouragé à aller de l'avant et

Qui m'a donné tout son amour

Mon très cher père est la plus belle perle du monde

Pour leur patience, leur amour, leur soutien et leurs

Encouragements.

A tout ma famille pour l'amour et le respect qu'ils m'ont toujours accordé

A mes frères

KAMEL, AREZKI, MOUH et IDIR

A mes sœurs

DJAMILA, NOUARA et SADIA

Je leur souhaite tout le succès... tout le bonheur

A mes beaux frères

SAID et DAOUDI

A mes belles sœurs

FADILA ET ZAINA

A mes adorables

TIKINAS, RAYANE, TINA, MELISSA, MELINA, EMMA, AYLINE et AHMED

A mon mari RYADH que dieu le protège

Une spéciale dédicace pour ma chère binôme Dadi (Ouarda) qui m'a supporté toute cette période de stress qui m'a rendu un peu folle mais que j'aime d'amour.

A mes amis et mes camarades : SARA, SAMIRA, AMEL, OUARDA, MOUH, YUCEF, OURAMDHANE, HAMZA, JUGO et YACINE.

A tout personne

Qui m'a aidé à franchir un horizon dans ma vie

F. KENZA

Dédicaces

Je dédie ce mémoire

Ames chers parents. Ma mère, qui m'a encouragé à aller de l'avant et

Qui m'a donné tout son amour

Mon très cher père est la plus belle perle du monde

Pour leur patience, leur amour, leur soutien et leurs

Encouragements.

A tout ma famille pour l'amour et le respect qu'ils m'ont toujours accordé

A mes frères

AMINE ET AMAR

A mes sœurs

HAYAT, GHANIA, HANNANE

A mes beaux frères

RABAH et SAID

A ma belle sœur

KATIA

A mes adorables

DANNA, ANELIA et SILAS,

A mes cousines

SONIA, SAMIRA et son adorable YANIS, MRIEME, SARAH

A toutes mes tantes et mes oncles et leur fils

Une spéciale dédicace pour ma chère binôme KIKI (Kenza) qui m'a supporté toute cette période de stresse qui m'a rendu un peu folle mais que j'aime d'amour.

A mes amis et mes camarades : YOUCEF, MOUH, SONIA, WARDIA, DIHIA, NADIA, KAHINA, NADIA, OURMDHANE, HAMZA, JUGO, et YACINE.

A tout personne

Qui m'a aidé à franchir un horizon dans ma vie

S. OUARDA

SOMMAIRE

Remerciements	II
Dédicaces	III
Dédicaces	IV
Liste des Figures	A
Liste des Tableaux	B
Liste des Symboles.....	C
Introduction Générale	1

Chapitre I : Recherche bibliographique

I.1 Introduction	3
I.2 Généralités sur les UAV	3
I.2.1 Définition d'UAV	3
I.2.2 Historique.....	3
I.2.3 Classification des drones.....	4
I.2.3.1 Selon la taille.....	5
I.2.3.2 Selon la propulsion.....	8
I.2.3.3 Selon le nombre de rotor.....	8
I.3 Utilisations des drones.....	8
I.3.1 Applications militaires.....	8
I.3.2 Applications civiles.....	8
I.4 Les capteurs pour la navigation et la localisation.....	13
I.4.1 Les capteurs proprioceptifs.....	13
I.4.2 Les capteurs extéroceptifs.....	14
I.4.3 Les capteurs exproprioceptifs.....	14
I.5 Les Architectures de commande des drones.....	14
I.5.1 Les méthodes de pilotage et de guidage.....	15
I.5.1.1 Les Techniques de commande linéaires.....	15
I.5.1.2 Les Techniques de commande non linéaire.....	16
I.6 Conclusion.....	17

Chapitre II : Principe de fonctionnement et modélisation d'un quadrirotor

II.1 Introduction	18
II.2 Description d'un Quadrirotor... ..	18
II.2.1 Dynamique de vol.....	20
II.2.2 Mouvement principaux.....	21

SOMMAIRE

II.3 Repérage du quadrirotor dans l'espace.....	26
II.3.1 Repère utilisés	26
II.3.2 Matrice de rotation.....	27
II.3.2.1 Paramètres cinématique.....	29
II.3.2.2 Cinématique de translation.....	30
II.3.2.3 Cinématique de rotation.....	30
II.4 Modélisation dynamique avec le formalisme Newton-Euler.....	32
II.4.1 Equations des forces.....	32
II.4.2 Equations des moments.....	36
II.5 La représentation d'état du système.....	41
II.5.1 Vecteur d'état.....	41
II.5.2 vecteur de commande.....	42
II.5.3 Contraintes de sous actionnement.....	45
II.6 Conclusion.....	47

Chapitre III : Commande et Simulation

III.1 Introduction	48
III.2 Principe de régulation à base d'un régulateur PID	48
III.2.1 La composant proportionnelle.....	49
III.2.2 La composant intégrale	50
III.2.3 La composant dérivée	50
III.3 Régulateur proportionnel-intégral PI.....	50
III.3.1 Le correcteur PID.....	51
III.3.2 Définition de la réponse d'un systeme.....	52
III.4 Modélisation sous Simulink de quadrirotor	53
III.4.1 La stabilité de quadrirotor	56
III.5 Conclusion	59

Chapitre IV : Réalisation d'un banc d'essai

IV.1 Introduction	60
IV.2 Partie 1 : les éléments de banc d'essai et sa réalisation	60
IV .2.1 Composant mecanique	61
IV..2.2 Composant electrique.....	62
IV.2.3 Câblage des ESC	68

SOMMAIRE

IV.2.3.1 Calibrage	70
IV.2.3.2 Câblage de capteur MPU 6050	70
IV. 3 Partie 2 : commande de banc d'essai	71
IV.3.1 Programme	72
IV.3.2 Procédure de calibration	74
IV.4 Conclusion.....	75
Conclusion générale.....	76
Annexe A :Programmation de capteur	76
Annexe B:Progamation de la radiocommande	80
Annexe C : Programme final.....	85

Liste des Figures.

Figure I.1 : Segmentation des systèmes de drones aériens.....	5
Figure I.2 : Exemple d'un drone de type MALE.....	6
Figure I.3 : Exemple d'un drone de type HALE.....	7
Figure I.4 : Sperwer (SAGEM).....	7
Figure I.5 : Mini drone.....	8
Figure I.6 : Micro drone.....	8
Figure I.7 : Drone à voilures fixes.....	9
Figure I.8 : Drone à ailes battantes.....	9
Figure I.9 : Drones à voilure tournantes.....	10
Figure I.10 : Avion 3D.....	10
Figure I.11 : Hélicoptère classique (à gauche) Hélicoptère entandem (à droite).....	11
Figure I.12 : Drones multi rotors.....	11
Figure I.13 : Drone militaires de surveillance.....	12
Figure I.14 : Drones civils.....	13
Figure II.1 : Les deux types de quad-copter.....	19
Figure II.2 : Système d'axes utilisés comme référentiels.....	20
Figure II.3 : Principe de fonctionnement du Quadrirotor.....	21
Figure II.4 : Mouvement de lacet.....	22
Figure II.5 : Force aérodynamique.....	23
Figure II.6 : mouvement tangage et le roulis.....	23
Figure II.7 : mouvement tangage.....	24
Figure II.8 : Mouvement de roulis.....	24
Figure II.9 : vol vertical.....	25
Figure II.10 : Repérage de quadrirotor.....	26
Figure II.11 : Représentation des Angles d'Euler.....	27
Figure II.12 : Rotation au tour de l'axe X (Roulis).....	28
Figure II.13 : Rotation au tour de l'axe Y (Tangage).....	28
Figure II.14 : Rotation au tour de l'axe Z (Lacet).....	29
Figure II.15 : commande des systèmes.....	41
Figure III.1 : structure de la commande par PID.....	48
Figure III.2 : Régulateur proportionnel.....	50
Figure III.3 : Régulateur proportionnel-intégrale.....	51
Figure III.4 : Caractéristiques de la réponse d'un système.....	53

Liste des Figures.

Figure III.5 : simulation des accélérations angulaires.....	54
Figure III.6 : simulation de déplacement.....	55
Figure III.7 : le modèle du quadrirotor.....	56
Figure III.8 : le signal de reference de la position «z».....	57
Figure III.9 : schéma simulink du régulateur PID.....	57
Figure III.10 : le bloc PID.....	58
Figure III.11 : la sortie de la commande par PID (etat1).....	58
Figure III.12 : Les résultats de la commande par PID (etat1).....	58
Figure III.13 : Les résultats de la commande par PID (etat2)	59
Figure IV.1 : les châssis.....	61
Figure IV.2 : caractéristique de l'hélice.....	62
Figure IV.3 : Hélices 5045 de quadrirotor.....	63
Figure IV.4 : contrôleur électronique de vitesse.....	65
Figure IV.5 : Alimentation découpage	66
Figure IV.6 : MPU6050.....	67
Figure IV.7 : FlySky FS-i6 et son récepteur.....	68
Figure IV.8 : réalisation pratique.....	69
Figure IV.9 : câblage des systèmes avec arduino	69
Figure IV.10 : câblage des ESC.....	70
Figure IV.11 : câblage de capteur avec arduino	71
Figure IV.12 : notre banc d'essai.....	72
Figure IV.13 : la stabilisation souhaiter.....	73
Figure IV.14 : angle d'inclinations.....	73
Figure B.1 : Appairer émetteur et récepteur.....	85
Figure B.2 : Câblage de l'arduino avec le récepteur	89

Liste des Tableaux

Tableau I.1 : historique des quadrotors.....	4
Tableau III.1 : les paramètres constant de quadrirotor.....	52
Tableau III.2 : les conditions initiales pour les angles et z.....	55
Tableau III.3 : les conditions initiales pour les angles et z	56
Tableau IV.1 . Le choix des moteurs et les hélices adéquates.....	64
Tableau IV.2 : choix des moteurs d'un quadrirotor.....	64
Tableau IV.3 : caractéristique de module MPU 6050.....	67

Liste des symboles

P : Pesanteur

m : la masse

g : Accélération de la pesanteur (9,81 m/s²)

F : Force de poussée

F_g : Force de gravité

F_f : Force de portance

C_t : terme aéronautique

V : Vitesse linéaire

Ω : Vitesse angulaire dans le repère fixe

W : Vitesse angulaire

F_t : Traînée selon les axes

K_{f t} : Coefficient de traînée

d : Coefficient de trainée

b : Coefficient de portance

l : Longueur du segment

M_f : Matrice de moment de portance

M_x, M_y, M_z : Moment de rotation selon les axes X, Y et Z

M_a : Moment de frottement aérodynamique

K_{f a} : Coefficient de frottement aérodynamique

J : Inertie du système

J_r : Inertie des rotors

∧ : Produit vectoriel

M_{g h} : Moment gyroscopique selon les hélices

Liste des symboles

Mg m : Moment gyroscopique selon les mvt du drone

Φ : Angle de rotation du couple créé selon X

Θ : Angle de rotation du couple créé selon Y

Ψ : Angle de rotation du couple créé selon Z

R : Matrice de rotation

R0 : Le repère terrestre

R1 : Le repère lié au corps du Quadrirotor

Rotx, Roty, Rotz : Matrice de rotation autour de X, Y et Z

c, s, tan, atan : Cos, sin, tangente, arc-tangente

Ω : vecteur des angles d'Euler

ρ : la densité de l'air

A : l'aire de disque généré par l'hélice

Ti : la force de la poussé

PID : Proportionnel Integral Dérivée

DDL : degré de liberté

UAV : Unmanned Aerial Vehicles

MCC : Machine à Courant Continue

Introduction Générale

Introduction Générale :

Le monde a connu des avancées technologiques dans différents domaines tels que la robotique, l'automatisation et l'aéronautique.

De nombreuses applications potentielles qui ont suscité un intérêt croissant sont portées aux engins volants sans pilotes humain à bord que l'on appelle UAV (Unmanned Aerial Vehicle) ou bien drone. Un drone est un véhicule aérien robotique sans pilotes capables de mener une mission de façon autonome [1].

Au début, les applications du drone ont été orientées uniquement vers le domaine militaire. Aujourd'hui, ils sont aussi utilisés dans divers domaines tels que la recherche, le sauvetage, la cartographie, etc.

Dans le domaine civil, l'utilisation des drones est envisageable pour un grand nombre de missions délicates dangereuses ou coûteuses telles que l'exploration d'un environnement inconnu, la surveillance ou l'intervention sur des zones potentiellement dangereuses, l'évaluation des dommages, la surveillance des feux de forêts, des lignes électriques à haute tension, du trafic routier, le survol des régions montagneuses et peu accessibles [2].

La recherche dans le domaine des drones fait appel à plusieurs disciplines scientifiques à savoir l'aérodynamique, la mécanique, l'électronique, l'automatique, la communication...etc.

Dans ce mémoire, nous nous intéressons en particulier aux UAV (Unmanned Aerial Vehicle) et plus particulièrement à un quadri-rotor. Les quadri-rotors sont parmi les plus complexes des objets volants, parce que leur dynamique de vol est intrinsèquement non linéaire, et les variables sont fortement couplées.

De nombreux travaux ont été réalisés sur la modélisation et la commande d'un quadri-rotor, dans le but de maîtriser la déstabilisation et la stabilisation du quad-rotor selon les valeurs du régulateur PID [3]. Pour contrôler les différents mouvements d'un drone, il faut considérer le mouvement vertical, le roulis, tangage et le lacet.

Ce mémoire s'articule autour de quatre chapitres :

Introduction Générale

Dans le premier chapitre, nous présentons en 1er lieu une recherche bibliographique sur les drones, leurs classifications, et leurs domaines d'applications. Ensuite, une revue de littérature sur les diverses approches de commande linéaires et non-linéaires.

Le deuxième chapitre est consacré à la description détaillée du principe de fonctionnement du drone. Par la suite les modes de vol ont été donnés et un ensemble de repères sont présentés avant d'essayer de modéliser la dynamique ou la cinématique du quadrirotor. En se basant sur les lois de la cinématique (Euler / Lagrange), ce chapitre présente aussi une modélisation dynamique du drone quadrirotor. Connaissant les relations entre les poussées des différents moteurs et les accélérations angulaires.

Dans le troisième chapitre, nous exposons une technique de régulation PID pour avoir un bon suivi des consignes. Puis nous discuterons les résultats obtenus lors des simulations.

Dans le quatrième chapitre, nous présentons le matériel utilisé pour la construction mécanique, et la réalisation d'un banc d'essai et sa stabilisation.

Enfin nous concluons notre travail avec une conclusion générale résumant les différents résultats obtenus ainsi que quelques perspectives et les difficultés rencontrées dans la réalisation pratique.

Le travail présenté à pour objectif de concevoir et de réaliser d'un banc d'essai a un seul axe, pour permettre aux étudiants de faire des TP pédagogiques et comprendre comment passer de la modélisation à l'implémentation d'une commande sur un système physique.

CHAPITRE I

Recherche bibliographique

I.1 Introduction :

La robotique aérienne est un champ vaste et interdisciplinaire. Les drones ou UAVs (Unmanned Aerial Vehicles) sont des aéronefs sans pilote capables de mener à bien des missions de façon plus au moins autonome. La fonction principale de ces véhicules aériens est d'étendre la vision humaine au-delà de l'horizon, afin d'accomplir des travaux à risques ou dans des environnements hostiles. Ils ont joué un rôle primordial dans la réalisation des missions de reconnaissances, de recherche d'informations voire des opérations de combat.

Dans ce chapitre, nous présentons : dans un premier temps, des généralités sur les UAV et leur historique, puis nous exposons les différentes catégories de drones, ensuite nous présentons ses avantages techniques et pratiques à travers différents projets et applications existantes.

I.2 Généralités sur les UAV :**I.2.1 Définition d'UAV :**

Un UAV (Unmanned Aerial Vehicle) ou drone est un véhicule aérien motorisé, qui ne transporte pas d'opérateur humain, utilise la force aérodynamique pour assurer sa portance, peut voler de façon autonome ou être piloté à distance, être réutilisable ou récupérable et qui emporte une charge utile létale ou non létale [4].

I.2.2 Historique :

Bien que la configuration "Quadrirotor" n'a pas obtenu beaucoup d'attention jusqu'au début des années 80. Depuis, plusieurs chercheurs ont commencé à s'intéresser à cette configuration pour des raisons de simplicité, de capacité à supporter une charge utile et son coût réduit.

Depuis 2001, plusieurs centres de recherche et de groupes de spécialistes en aéronautique ont commencé à publier les premiers résultats à propos de la modélisation et de la commande de cet hélicoptère à quatre rotors tels que : les travaux de l'université de Compiègne Et le projet 'Robovolint' en France, ceux de l'université de Lakehead au Canada et beaucoup d'autres.

Actuellement, de nombreux projets de recherche sont basés sur des jouets comme le draganflyer [5], ensuite les chercheurs les ont modifiés en embarquant plus de capteurs et des systèmes de communication. Peut de groupes s'intéressent à la conception et la réalisation des quadrotors, et la minorité de ces groupes le fait de manière optimale (considération de la conception et le contrôle) [6].

Drones	QH-50C DASH	RQ-2 PIONEER	RQ-4 GLOBAL HAWK	Rq-11 RAVEN	NEURON
Constructeurs	Gyrodyne Company of America	Israel Aerospace Industries	Nor throp	AeroViron ment	Dassault Aviation
Premier vol	1959	1986	1998	2001	2012
propulsion	1 turbine Boeing de 300 ch	1 moteur, 2 cylindres de 26 ch	1 turboréacteur de 31,4 kN	Moteur electrique	1 turboréacteur de 29 kN
longueur	3,94m	4,26m	14,50m	1,1m	9,20m
envergure		5,15m	39,80m	1,3m	12,50m
Masse à vide	524kg	177kg	510kg	6,7kg	49000kg
Masse maximum	1036kg	204kg	1020kg		7000kg
Vitesse maximum	148 km /h	203km /h	635km /h	95km /h	980km/h
Altitude maximum	5000m	46000m	18300m	5000m	14000m
Autonomie		4h	36h	60-90m	3h
armement	2 torpilles MK 44 ou 46				2 bombes guidées laser de 205kg

Tableau I.1: Historique des quadrotors.

I.2.3 Classification des drones :

Les drones peuvent être répartis selon plusieurs critères : la taille, l’altitude, les systèmes de contrôle, etc.

I.2.3.1 Selon la taille :

Il n’est pas facile de classer les systèmes aériens, car il existe plusieurs catégories de drones aériens par tant du nano-drone de quelques grammes jusqu’au drone lourd capable d’effectuer des missions de plus de 24 heures à plusieurs milliers de kilomètres de sa base. La figure ci-dessous illustre les différents types de drones selon les principaux critères: la taille et les performances (autonomie en vol ,portée ou altitude d’opération).

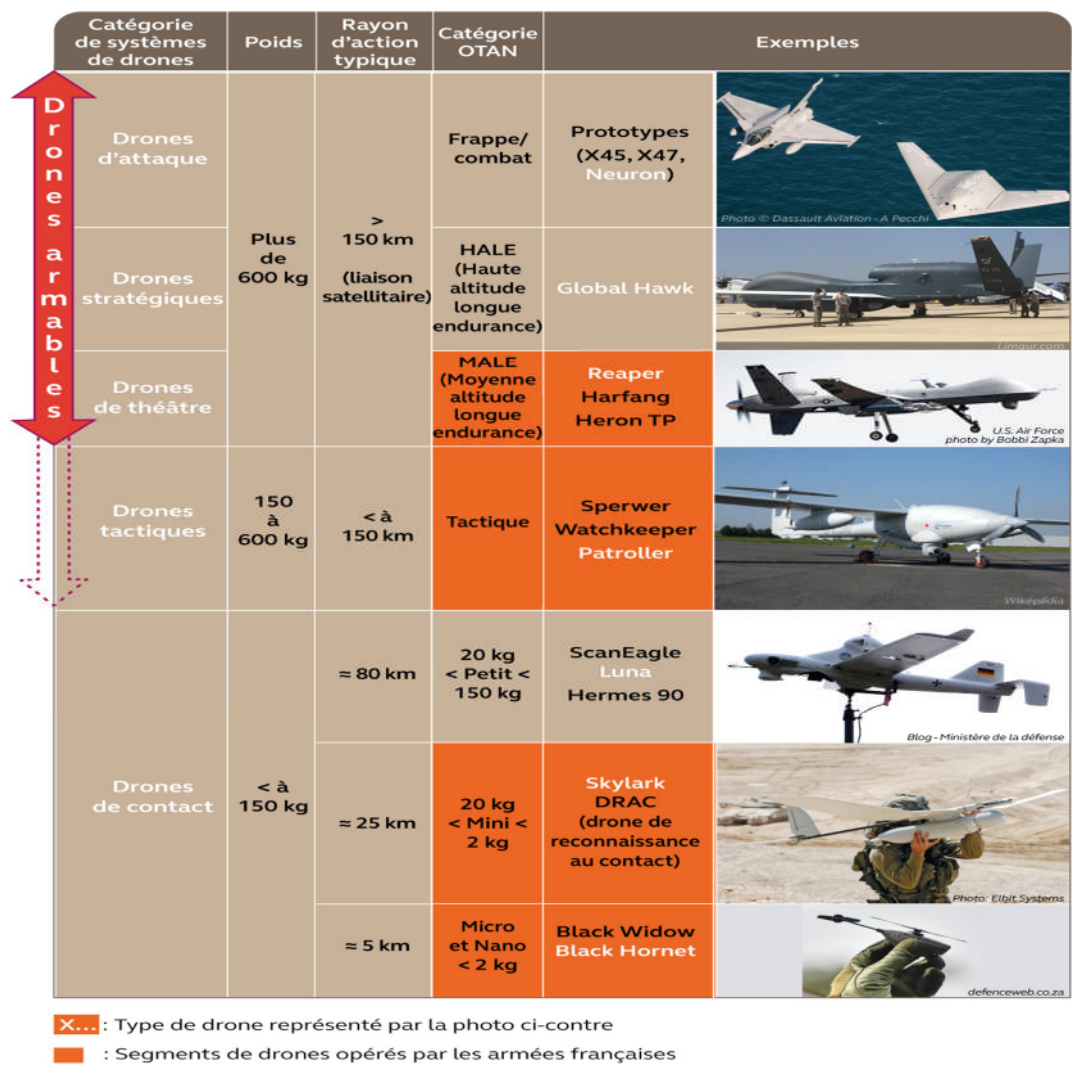


Figure I.1: Segmentation des systèmes de drones aériens

Trois grandes familles de drones se détachent alors en fonction du type de mission :

1. Les drones de Moyenne Altitude et Longue Endurance (MALE) .
2. Les drones de Haute Altitude et Longues Endurances (HALE)
3. Les drones tactiques.

- **MALE (Moyenne Altitude Longue Endurance):**

Sont utilisés pour des vols de longue durée à moyenne altitude opérationnelle, ayant une grande autonomie.

Ces deux types de drones font partie de la classe de grande taille. Ils peuvent embarquer des armes, ce qui nécessite généralement d'avoir un humain dans la boucle, ce dernier doit garder la décision de tir et pouvoir à tout moment annuler la mission.



Figure I.2 : Exemple d'un drone de type MALE

- **HALE (Haute Altitude Longue Endurance) :** Ce sont des drones de grande taille, le plus souvent à voilure fixe. Ils sont capables de rester très longtemps en vol et de collecter des informations sur de très longues périodes (entre 12 et 48 heures)



Figure I.3 : Exemple d'un drone de type HALE

Les drones tactiques, tels que le Sperwer (SAGEM) ou l'Aérostar (Aéronautiques Défense Systèmes), utilisés pour des missions de reconnaissance ou de supervision du champ de bataille et volant à une altitude comprise entre 200 m et 5 km, pour un rayon d'action de 30 à 500km.



Figure I.4 : Sperwer (SAGEM)

- **Mini drones :** Ce sont des drones légers et de taille réduite (jusqu'à quelques kilogrammes et d'une envergure jusqu'à 1 à 2 mètres) ils ont une autonomie relativement faible (de 10 à 30 minutes) et généralement utilisés pour l'observation de zones difficiles à l'accès.



Figure I.5 : Mini drone

- **Micro drones :** Ce sont des drones ayant des tailles variant du centimètre à quelques dizaines de centimètres. Généralement propulsés électriquement. Ils permettent de faire des vols à l'intérieur. ils s'emportent de faibles charges.



Figure I.6 : Micro drone

I.2.3.2 Selon le mode de propulsion :

On peut aussi classer les drones selon le fonctionnement aérodynamique dont on trouve :

- **Les drones à voilures fixes :**

Ces configurations telle d'une configuration d'un avion conventionnel doté d'une paire d'ailes assurant la sustentation, un fuselage, une dérive et un empennage, la propulsion de ce

dernier est assurée par une ou plusieurs hélices [7]. Ce type de drone offrent plus d'autonomie mais nécessitent de l'espace pour le décollage et l'atterrissage [8].



Figure I.7 : Drone à voilures fixes

- **Les drones à ailes battantes :**

Pour ces drones, la puissance et le contrôle sont assurés grâce à des ailes battantes doté d'une capacité de vol stationnaire comme les voilures tournantes [9]. Et de vol avec des trajectoires similaires à celle des insectes ce qui est intéressant pour des missions dans des espaces réduits, ces configurations sont d'une telle agilité tout en étant discrètes ce qui présente un autre avantage pour des missions de reconnaissance ou de surveillance [10].

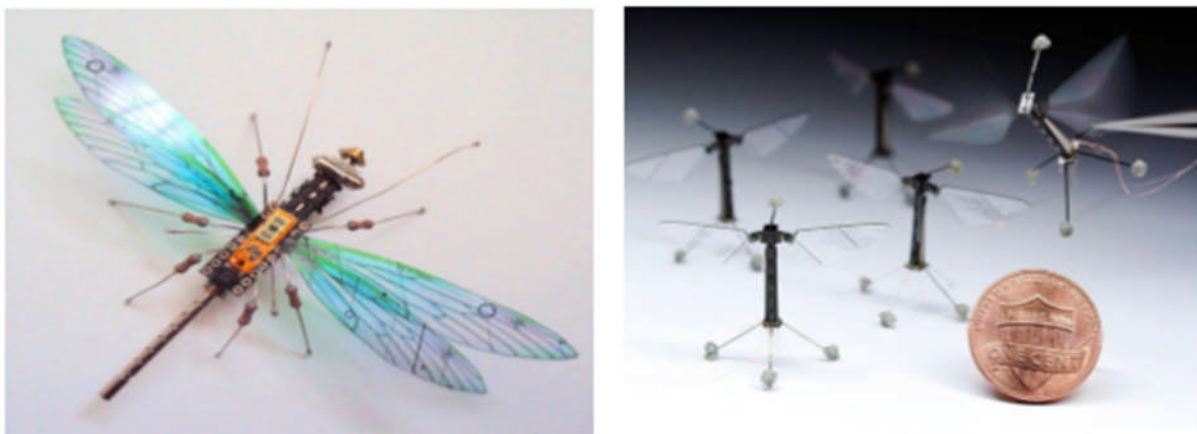


Figure I.8 : Drone à ailes battantes

- **Drones à voilures tournantes :**

Les cellules à voilures tournantes sont des aéronefs ayant la capacité de décoller, voler et atterrir verticalement (ADAV, en anglais vertical take-off and landing aircraft ou autrement dit VTOL) et donc n'ont pas besoin de piste de décollage au d'atterrissage [11]. Ils utilisent des voilures tournantes produisant la portance en soufflant de l'air vers le bas grâce à leur(s) rotor(s). Il en existe deux types : les hélicoptères traditionnels a pas variable et les récent (et très populaires) multi-rotors a pas fixe.

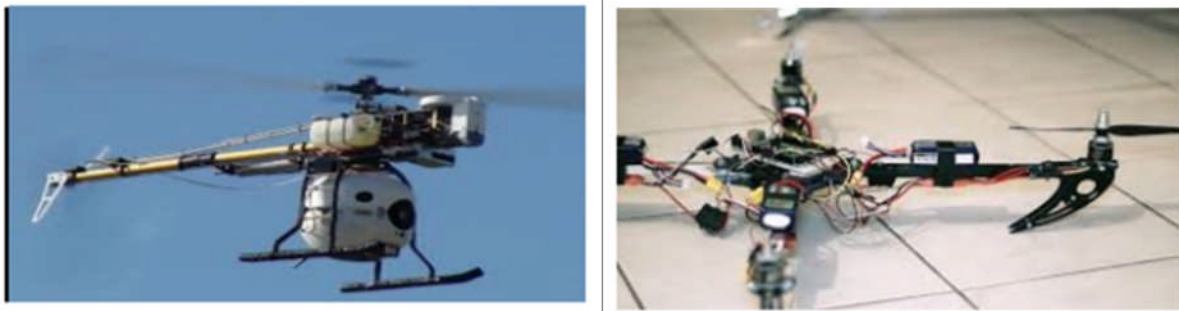


Figure I.9 : Drones à voilure tournantes

I.2.3.3 Selon le nombre de rotors :

Les types de drones les plus courants dans cette catégorie sont :

- **Mono-rotors :**

L'exemple le plus commun dans cette configuration est l'avion Appelés 3D, qui a un seul moteur comme actionneur principal, et qui doit être suffisamment puissant Pour décoller verticalement, et des ailerons d'une grande surface Qui assurent des couples de commande suffisamment grands afin de piloter l'appareil facilement.



DHgate.com alongrc

Figure I.10 : Avion 3D

- **Birotors :**

Ce type de configuration incluse deux types de drones, ceux avec un Ou deux plateaux cycliques comme l'hélicoptère classique et ceux avec des pales à pas fixe. Le cas de l'hélicoptère nous avons un rotor principal et un rotor de queue Il existe aussi l'hélicoptère entandem qui possède deux rotors qui tournent en contre sens mais dans des axes différents.



. **Figure I.11 :** Hélicoptère classique (à gauche) Hélicoptère entandem (à droite)

Nous pouvons aussi distinguer la configuration à deux rotors, coaxiales, qui participent les deux à la poussée. En tournant dans des sens opposés les effets de couple générés par la rotation des rotors, sont annulés.

- **Multi-rotors :**

Ce type de véhicules aériens se compose de plusieurs rotors avec un sens de rotation inversé deux à deux pour compenser le couple de réaction, le changement des vitesses des moteurs convenablement assure les déplacements du Drone.



Figure I.12 : Drones multi rotors

I.3 Utilisations des drones :

Les drones sont conçus à la base pour remplacer l'homme dans des environnements ou des situations dangereuses, Le domaine d'application des aéronefs qui ne cesse de s'élargir [12]. Principalement on peut distinguer deux types d'applications : Applications militaires et application civiles.

I.3.1 Applications militaires :

Les drones jouent un rôle toujours plus important dans les considérations militaires. de manière générale à un triple usage dans les armées contemporaines :

- La surveillance et le renseignement.
- La reconnaissance du terrain pour les troupes terrestres et aériennes
- Le combat.



Figure I.13 : Drone militaires de surveillance

I.3.2 Applications civiles :

L'utilisation des drones civils représente une petite révolution dans l'exploitation de l'espace aérien, il a évolué énormément et permis de répondre à une multitude de besoins spécifique bien plus qu'un loisir.

On le retrouve aujourd'hui dans de nombreux domaines d'activités tel-que :

- Etude de l'atmosphère, des sols (géologie) et des océans.
- Incendie de forêts, avalanches
- Surveillance des cultures et épandage agricole.

- Recherche et sauvetage (mer, montagnes, désert...).
- Largage de vivres et d'équipements de sauvetage en zone hostiles
- Largage de vivres et d'équipements de sauvetage en zone hostiles
- Surveillance du trafic routier et du transport de matières dangereuses



Figure I.14 : Drones civils

I.4 Les capteurs pour la navigation et la localisation :

Dans cette section nous décrivons les différents types de capteurs qui équipent les drones et sont essentiels pour les doter de capacité de navigation et de localisation. Ces capteurs peuvent être divisés en trois catégories :

I.4.1 Les capteurs proprioceptifs :

Les informations proprioceptives renseignent sur le déplacement d'un véhicule dans l'espace. Elles sont obtenues par la mesure d'informations relatives aux mouvements de l'appareil par rapport à son état de référence.

Elles constituent une source d'informations simples de manipulation ne dépendant pas des conditions environnementales dans lesquelles évolue l'engin.

Elles sont donc d'une importance capitale pour la navigation autonome des drones. Cependant, la précision de ces informations se dégrade continuellement au cours du temps, les rendant inutilisables comme seules références à long terme.

Il est alors indispensable de leur associer un système permettant de recalibrer périodiquement la position absolue du véhicule [13].

Les capteurs proprioceptifs les plus utilisées sur les drones sont présentés ci-dessous :

- Accéléromètres
- Gyroscopes et gyromètres
- Centrales inertielle (IMU : Inertiel Measurement Unit)

I.4.2 Les capteurs extéroceptifs :

Les informations extéroceptives, quant à elles, renseignent sur la position absolue d'un véhicule dans l'environnement.

Elles sont obtenues par l'observation de points de repère naturels ou artificiels dont la situation est connue dans un référentiel attaché à l'environnement.

Ces informations sont indépendantes des déplacements du robot et pourront donc être acquises quelle que soit l'erreur accumulée par les données proprioceptives.

De plus, la reconnaissance des amers n'est pas soumise à une erreur cumulative, ce qui rend les informations extéroceptives utilisables comme références à long terme [14].

Les capteurs extéroceptifs les plus utilisées sur les drones sont présenté ci-dessous :

- Compas magnétique
- Gyrocompas
- Localisation sur balise (GPS : Global Positioning System)
- Capteurs télémétrique
- Caméra

I.4.3 Les capteurs exproprioceptifs :

C'est une combinaison entre les deux types de capteurs proprioceptifs et extéroceptifs.

I.5 Les Architectures de commande des drones :

Les différentes architectures de commande se résument au pilotage, au guidage et à la navigation autonome, ce qui correspond chacun a une fonction bien précise de la commande des aéronefs [15].

En effet dans le domaine aéronautique le pilotage concerne le contrôle de l'attitude, le guidage le contrôle de la vitesse et enfin la navigation le contrôle en position.

Les stratégies de commande des véhicules aériens autonome a fait l'objet de plusieurs publications, cette section est destinée à présenter quelques architectures de commande les plus employées dans les applications de la robotique.

I.5.1 Les méthodes de pilotage et de guidage :

Durant ces dernières années la conception et la mise au point des algorithmes de commande pour les drones aériens a fait l'objet d'un certain nombre d'études.

Ceci est dû au besoin de produire des véhicules aériens autonomes [16], En effets, ces appareils sont des systèmes sous-actionnés sensible aux perturbations aérodynamique et dont la dynamique est fortement non-linéaire et fortement couplée.

Plusieurs méthodologies de la conception de la loi de commande pour les drones ont été développées afin de stabiliser leurs mouvements, certaines de ces lois sont basé sur le modèle linéaire alors que d'autre considèrent le modèle en entiers (modèle non-linéaire) [17].

I.5.1.1 Les Techniques de commande linéaires :

Dans le cadre de la théorie des systèmes linéaires, la synthèse de la commande est basée sur une approximation linéaire du modèle dynamique du véhicule. Cette approche permet de découpler cette dynamique en quatre boucles de type SISO (de l'anglais Single Input - Single Output), associée à une seule entrée de commande [18].

L'élaboration de la loi de commande linéaire pour contrôler le mouvement d'un drone pose de nombreux problèmes car les modèles linéaires simplifiés sont généralement loin de la réalité du système physique.

Le modèle dynamique complet d'un drone engendre en réalité des incertitudes qui constituent des erreurs de dynamique par rapport au modèle linéaire, et par conséquent rend la conception de l'algorithme de commande très complexe, en plus de ca ils doivent présenter des propriétés de robustesse et de performance [19].

Les commandes linéaires les plus utilisées pour le pilotage et le guidage des drones sont présenté ci-dessous :

- La commande par PID ;
- La commande par placement de pôles ;
- La commande linéaire quadratique ;

- La commande linéaire quadratique gaussienne ;
- La commande prédictive ;
- La commande par échelonnement de gains ;
- La commande H_∞ .

I.5.1.2 Les Techniques de commande non linéaires :

Dans la plupart des applications, la commande embarquée est basée sur modèle linéarisé de la dynamique du véhicule considéré. Les techniques de commande non linéaire permettent d'améliorer les performances et la robustesse par rapport aux techniques linéaires [3].

Cependant ces dernières années, de plus en plus les recherches se sont orientées vers des commandes basées sur une représentation non-linéaire de la dynamique des véhicules aériens ces approches non-linéaires apportent une contribution théorique certaine mais leur application, de par la complexité des modèles et des algorithmes de commande restent toutefois limitée [20].

Les commandes non-linéaires les plus utilisées pour le pilotage et le guidage des drones sont présenté ci-dessous :

- La commande par linéarisation entrée-sortie ;
- La commande par backstepping (Dynamics inversion) ;
- La commande hiérarchique ;
- La commande par saturations imbriquées ;
- La commande par mode glissants (sliding mode control) ;
- La commande par la logique floue ;
- La commande adaptative ;
- La commande par asservissement visuel.

I.6 Conclusion :

Il existe une grande diversité de famille des drones, qui possède chacune ses spécificités techniques. Le choix d'une architecture de drone n'est pas anodin et doit notamment correspondre aux missions pour lesquelles il sera employé, ainsi qu'aux degrés de performances recherchés

Afin de permettre l'exploitation des drones sans intervention humaine, un système de commande et de décision doit être mis en place, ce dernier doit assurer soigneusement le contrôle du drone d'une manière cohérente, et dans les limites spécifiées afin d'assurer la qualité de l'application et la sécurité des équipements, Cet objectif de contrôle efficace et performant ne peut se réaliser sans l'appréhension parfaite du comportement dynamique de cet engin, C'est le rôle de l'étude de modélisation qui devrait précéder toute commande de systèmes.

Dans les chapitres suivants nous allons essayer d'établir un modèle dynamique décrivant les mouvements de notre véhicule aérien, et synthétiser une loi pour le commander.

CHAPITRE II

Principe de fonctionnement et modélisation d'un quadrirotor

II.1 Introduction :

Un système de drone est l'ensemble des éléments permettant le vol d'un aéronef sans pilote. Il est donc constitué de la cellule de l'aéronef mais aussi de la station sol, des systèmes de communication et de positionnement. Ce chapitre est consacré à la description des systèmes utilisés pour la mise en œuvre des quadri rotor, mais surtout à la modélisation physique.

L'objectif de la modélisation présentée n'est pas d'atteindre un niveau de finesse extrême mais de dégager les principaux phénomènes physiques qui régissent le comportement des quadrirotor.

Cela facilite le choix de paramètres pertinents à estimer pour la commande adaptative et met en valeur les dynamiques auxquelles le correcteur doit être robuste [21]

Afin de réaliser un contrôleur de vol robuste, nous devons d'abord comprendre les mouvements du système et sa dynamique. Cette compréhension est nécessaire pour assurer que les simulations de l'engin dépendront un comportement aussi proche que possible de la réalité

II.2 Description du quadri rotor :

Dans cette section, nous présentons certaines caractéristiques physiques des quadri rotors. Il est Constitué d'un corps principal mis en mouvement par 4 hélices entraînées chacune par un rotor fixé aux extrémités en forme de croix. Tous les composants électroniques du drone, dont les 4 moteurs, sont alimentés par une batterie Lithium polymère [22].

La carte électronique embarquée à bord du Quadri rotor traite les informations des différents capteurs et élabore la commande des 4 variateurs des moteurs en fonction de la consigne à suivre, définie par l'utilisateur et transmise au drone par liaison Wi-Fi. Le drone est équipé de plusieurs capteurs, à fonctionnalités bien différentes :

Chapitre II : *Principe de fonctionnement et modélisation d'un quadrirotor.*

- Un capteur à ultrasons situé sous le drone et pointant vers le sol,
- Un accéléromètre tri-axe (mesurant suivant trois directions orthogonales de l'espace),
- Un gyromètre tri-axe, une antenne GPS,
- Un magnétomètre tri-axe et deux caméras pointant l'une vers l'avant du drone et l'autre vers le sol. [23]

On distingue deux types de quadrirotor, les modèles dit "+" ou "plus" et ceux appelés "X". La structure en elle-même ne change pas, c'est l'orientation de l'avant de l'appareil qui varie.

Sur un Quad+, l'avant de l'appareil pointe dans la direction d'une des branches. Alors que pour un QuadX la structure est pivotée de 45°. L'avant se situe donc entre deux bras.

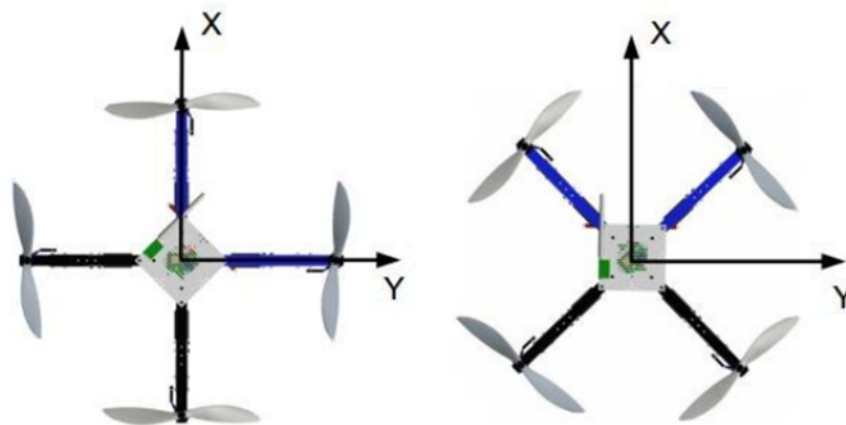


Figure II.1 : Les deux types de quad-copter

Une même structure peut potentiellement voler dans les deux dispositions. Cependant la gestion des mouvements ne se fait pas de la même manière. Il faut donc que l'attribution des moteurs puisse être modifiée. Le mode "X" implique une réaction de chaque moteur pour tous les déplacements alors que pour certaines rotations le Quad+ n'en modifie que deux. Les performances de vol des deux modes ne présentent pas de grosses différences.

Le mode "+" facilite le pilotage du quadrirotor. Le bras avant peut être facilement mis en évidence par marquage distinctif (couleur différente ou LED par exemple) et permet une identification rapide de la position du quadrirotor.

Chapitre II : Principe de fonctionnement et modélisation d'un quadrirotor.

Le mode "X" est presque toujours utilisé pour les modèles possédant une caméra fixe. Il est possible de filmer à l'avant du quadrirotor sans qu'un des bras ou qu'une hélice n'obstrue le champ de vision. Cela améliore la visibilité pour un vol en FPV (*First Person View*). [24]

II.2.1 Dynamique de vol :

Le quadrirotor répond à une dynamique de vol bien particulière. Si sa mobilité dans l'espace est équivalente à celle d'un hélicoptère, la réalisation des déplacements est fondamentalement différente. En effet, c'est l'effet combiné des quatre moteurs qui permet aux quadrirotor de s'orienter et de se mouvoir. Il est donc primordial de comprendre le fonctionnement de ce type de drone afin d'espérer le contrôler. [25]

Degré de liberté :

Le quadrirotor possède quatre degrés de liberté, une translation et trois rotations. On peut constituer un système d'axes orthogonaux ayant leur origine au centre de gravité de l'appareil.

L'axe X pointe vers l'avant du quadrirotor (axe longitudinal), l'axe Y est perpendiculaire à X sur un plan horizontal (axe transversal) tandis que l'axe Z représente l'axe vertical. [23]

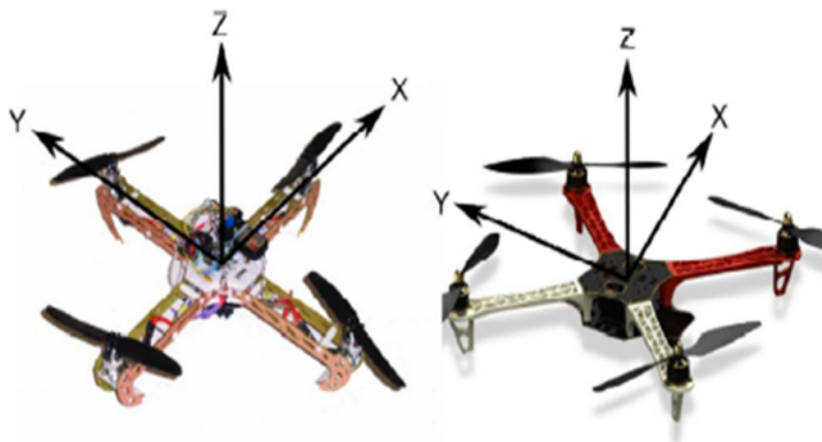


Figure II.2 : Système d'axes utilisés comme référentiels

II.2.2 Mouvement principaux :

Les mouvements de base de quadrirotor sont réalisés en variant la vitesse de chaque rotor changeant de ce fait la poussée produite. Le quadrirotor incline vers la direction du rotor plus lent, qui tient compte alors de la translation le long de cet axe [26]. Par conséquent, c'est l'inclinaison qui permet au quad-copter de réaliser une translation, ce qui signifie qu'un changement de la vitesse d'un rotor se traduit dans un mouvement en au moins trois DDL [27].

Les principaux mouvements et vol possibles pour un quadrirotor :

- a. Le mouvement de roulis.
- b. Le mouvement de tangage.
- c. Le mouvement de lacet.
- d. Vol vertical.
- e. Vol stationnaire.

Il est important qu'on puisse varier rapidement la vitesse de rotation des moteurs. Pour cela, il convient d'utiliser des pales très légères.

Le Quadrirotor ne possède pas intrinsèquement les degrés de libertés de translation dans le plan horizontal, mais ils peuvent être obtenus par l'addition d'une rotation et de la composante verticale.

Lorsque le quadrirotor est incliné grâce à une rotation selon X ou Y,

La poussée selon Z comprend une composante horizontale. En augmentant la poussée pour que la composante verticale soit égale à la force de gravité, le déplacement est purement horizontal.

Chapitre II : Principe de fonctionnement et modélisation d'un quadrirotor.

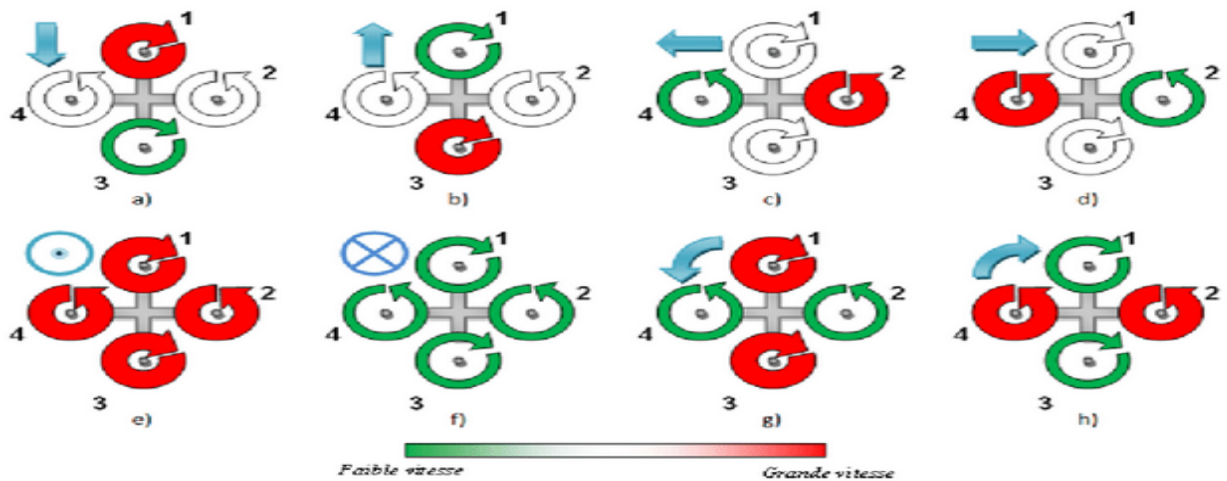


Figure II.3 : Principe de fonctionnement du Quadrirotor

a) Le mouvement de Lacet : (ψ)

La rotation autour de l'axe z est appelée lacet en aéronautique. Lorsque les moteurs tournent à vitesse égale le couple d'anti-rotation est nul et le drone ne tourne pas.

Par contre, si nous provoquons une différence de vitesse entre les moteurs tournant en sens opposé, les couples résistants provoquent un couple τ_{ψ}^a autour de l'axe z, provoquant ainsi la rotation de l'engin.[30] .

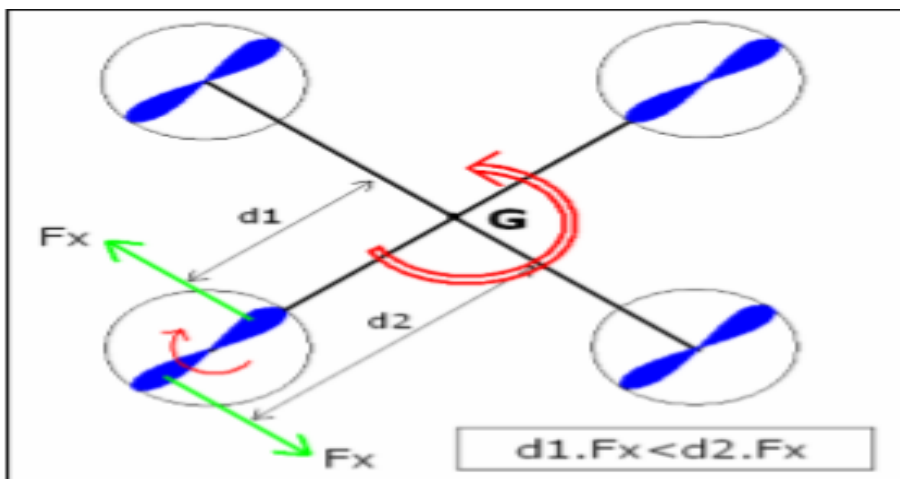


Figure II.4 : Mouvement de lacet.

Chapitre II : Principe de fonctionnement et modélisation d'un quadrirotor.

En effet, lorsque l'on projette les forces aérodynamiques exercées par l'air sur la pale, on s'aperçoit qu'un rotor a toujours tendance à faire tourner le micro hélicoptère dans le sens inverse de sa rotation.

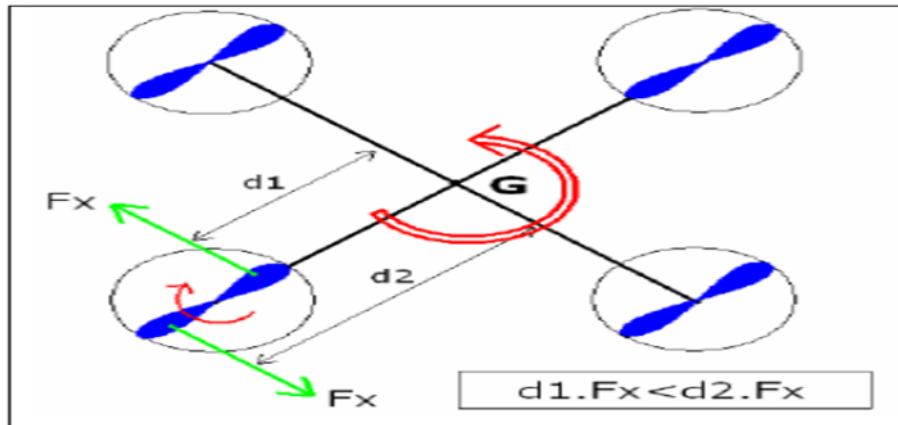


Figure II.5 : Force aérodynamique

b) Le Tangage (θ) et le Roulis (φ) :

Le roulis et le tangage sont des mouvements qui provoquent l'inclinaison du quadrirotor selon les axes horizontaux. Ces mouvements peuvent être assurés par le drone, en agissant sur la vitesse de l'un des quatre moteurs, est par conséquent sur la force des rotors.



Figure II.6 : mouvement tangage et le roulis.

La rotation autour de l'axe y est appelée tangage en aéronautique. Une variation de l'angle de tangage est obtenue grâce à une différence de vitesse entre le moteur un, et trois. Cette différence de forces produit un couple τ_{θ}^a autour de l'axe y

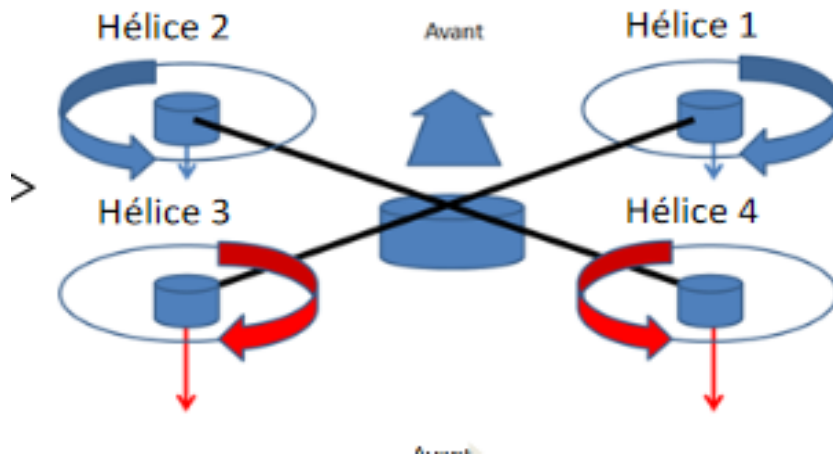


Figure II.7 : mouvement tangage.

En aéronautique, la rotation autour de l'axe x est appelée roulis. Une variation de l'angle de roulis est obtenue grâce à une différence de vitesse entre le moteur deux, et quatre. Cette différence de forces produit un couple $\tau \frac{a}{\phi}$ autour de l'axe x.

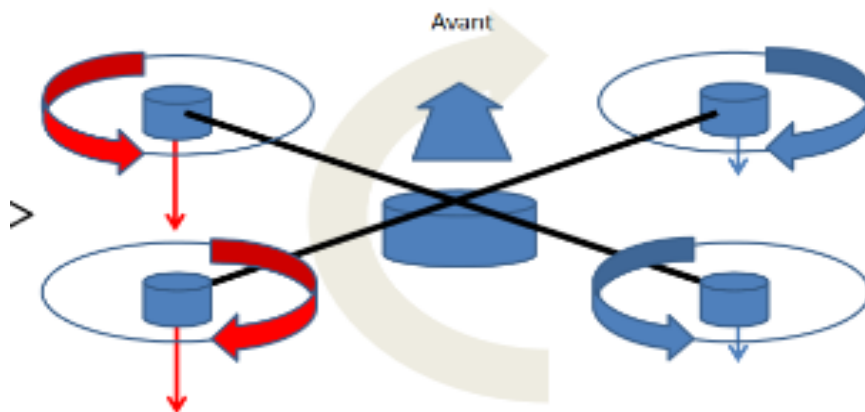


Figure II.8 : Mouvement de roulis.

Mode de vol :

c) Vol vertical :

À une vitesse égale des quatre hélices, les couples de rotation générés par une paire de moteurs du même axe s'annulent. Le Quadrirotor vole à plat et ne tourne pas sur son axe central.

Ainsi, la commande des quatre moteurs est nécessaire pour le vol vertical ou la commande en puissance, en augmentant la vitesse des quatre moteurs en même temps (tous ensemble) et avec la même valeur pour maintenir l'équilibre de l'ensemble, ainsi le décollage et l'inverse pour l'atterrissage, la force résultante due à l'application des forces F_1 , F_2 , F_3 , F_4 doivent être supérieur à la force de pesanteur pour que l'engin puisse décoller.

Le mouvement ascendant et descendant est obtenu par la variation de la vitesse de rotation des moteurs (par conséquent la poussée produite), si la force de portance est supérieure au poids du quadrirotor le mouvement est ascendant, et si la force de portance est inférieure au poids du quadrirotor le mouvement est descendant. [28]

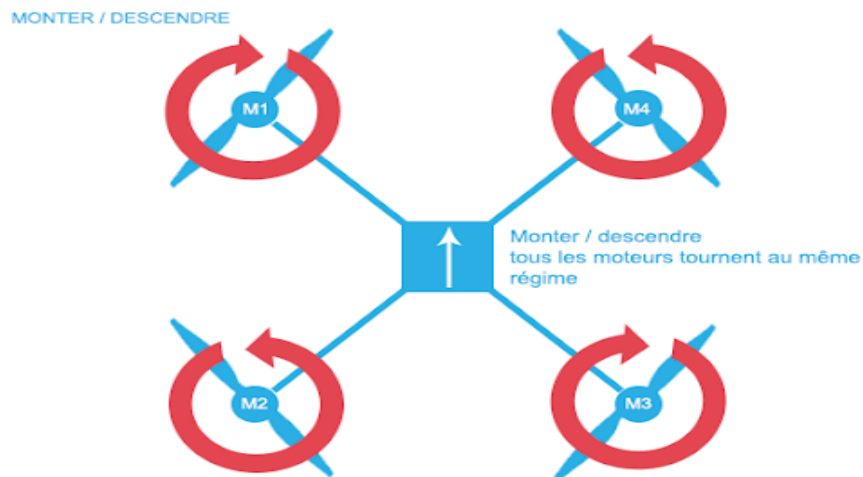


Figure II.9 : vol vertical

d) Le vol stationnaire :

En montée verticale et après avoir franchi le seuil définissant l'effet du sol, le quadrirotor peut rester en vol stationnaire à une certaine hauteur constante par rapport au

Chapitre II : Principe de fonctionnement et modélisation d'un quadrirotor.

sol en ayant une vitesse de translation nulle. La force de sustentation doit alors équilibrer le poids mg du Quadrirotor. Cette force de sustentation correspond à une force de poussée F_T qu'est orientée (en l'absence de perturbation) dans la direction de l'axe Z . Dans ce mode de vol, le quadrirotor a la liberté de faire des rotations autour de l'axe Z qui, dans ce cas, coïncide avec l'axe Z_0 du système de coordonnées inertiel.

II.3 Repérage du Quadrirotor dans l'espace :

II.3.1 Repères utilisés :

Pour évaluer le modèle mathématique du quadrirotor on utilise deux repères (**figure I7**), un repère fixe terrestre lié à la terre Il est noté : $R_0(O_0, X_0, Y_0, Z_0)$ et un autre mobile est un repère lié au corps du quadrirotor Il est noté : $R_1(O_1, X_1, Y_1, Z_1)$. C'est un repère dont l'origine O_1 coïncide avec le centre de gravité G du quadrirotor. [15]

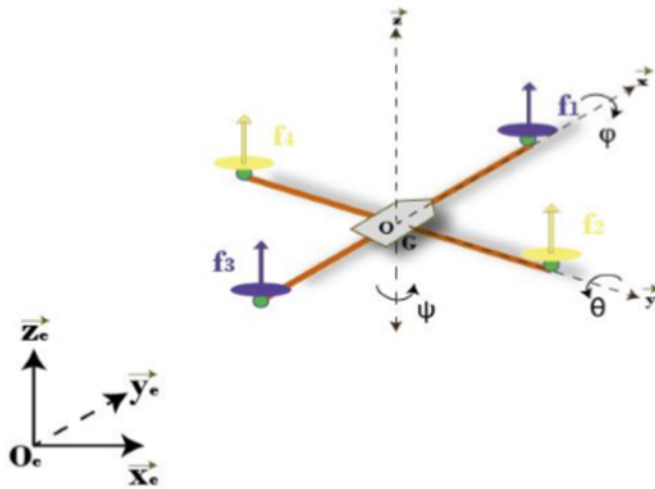


Figure II.10 : Repérage de quadrirotor.

Donc les paramètres qui permettent de décrire le mouvement du quadrirotor sont : [27, 29]

$(\Phi, \theta, \psi, x, y, z, \Omega, V)$ avec :

- Φ (angle de roulis) : rotation autour de X_1 ($-\pi < \varphi < \pi$).
- θ (angle de tangage) : rotation autour de Y_1 ($-2\pi < \theta < 2\pi$).

Chapitre II : Principe de fonctionnement et modélisation d'un quadrirotor.

- Ψ (**angle de lacet**) : rotation autour de Z_1 ($-\pi < \psi < \pi$).
- X : coordonnée du centre de gravité G du Quadrirotor suivant X_0 .
- Y : coordonnée du centre de gravité G du Quadrirotor suivant Y_0 .
- Z : coordonnée du centre de gravité G du Quadrirotor suivant Z_0 .
- $\Omega : [p, q, r]^T \in R_0$: la vitesse de rotation du quadrirotor par rapport au repère inertiel.
- $V : [u, v, w]^T \in R_0$: la vitesse linéaire liée du quadrirotor par rapport au repère inertiel.

Les angles d'Euler (φ, θ, ψ) sont représentés dans la **Figure II.10**

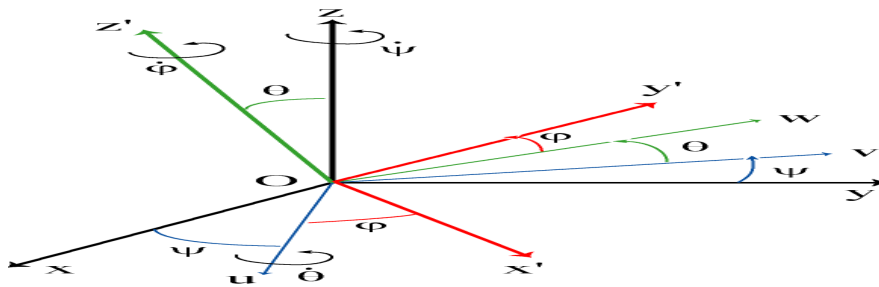


Figure II.11 : Représentation des Angles d'Euler.

II.3.2 Matrice de rotation :

On considère que les centres O_0 et O_1 des deux repères sont confondus, ce qui signifie que le repère R_1 ne fait que des rotations par rapport au repère R_0 . Trois paramètres indépendants sont nécessaires pour décrire complètement l'orientation du repère R_1 par rapport à celle de R_0 . Le passage du repère R_1 vers le repère R_0 se fera par trois rotations en utilisant deux repères intermédiaires R_i et J_r . [31].

a. “ Passage du repère R_0 vers le repère R_i

La rotation se fait autour de l'axe $x_i = x_0$. On passe du repère **R_0** vers **R_i** en faisant une rotation d'angle φ appelé angle de roulis.

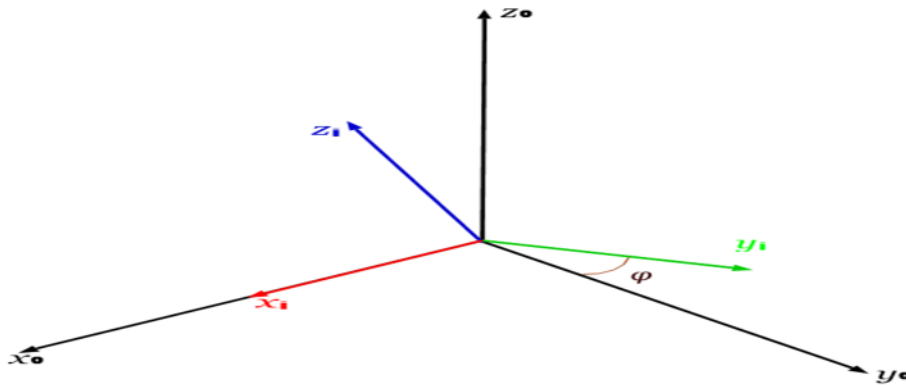


Figure II.12 : Rotation au tour de l'axe X (Roulis).

La représentation se fait par des figures planes, à partir desquelles nous construisons les matrices de passage, nous avons ainsi la matrice :

$$R(X_0, \varphi) = \begin{bmatrix} 1 & 0 & 0 \\ 0 & \cos(\varphi) & -\sin(\varphi) \\ 0 & \sin(\varphi) & \cos(\varphi) \end{bmatrix} \quad (2.1)$$

b. “ Passage du repère R_i vers le repère R_j

La rotation se fait autour de l'axe $Y_j = Y_i$. On passe du repère R_i vers le repère R_j en faisant une rotation d'angle θ appelé angle de tangage

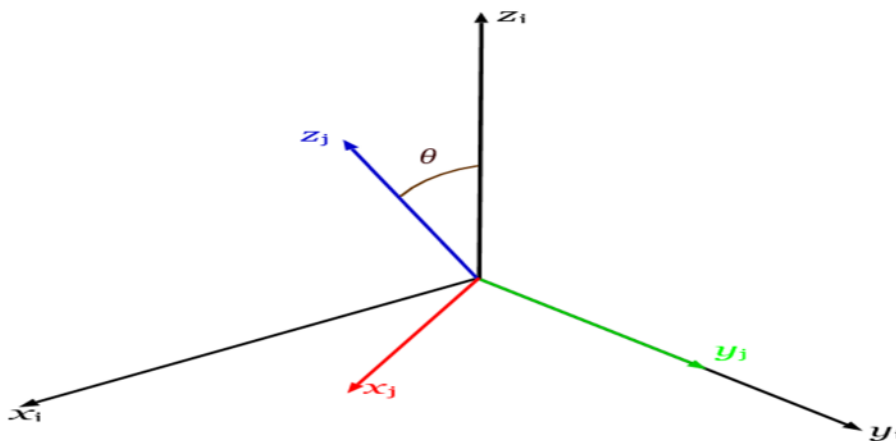


Figure II.13 : Rotation au tour de l'axe Y (Tangage)

Chapitre II : Principe de fonctionnement et modélisation d'un quadrirotor.

Nous avons ainsi la matrice :

$$R(Y_0, \theta) = \begin{bmatrix} \cos(\theta) & 0 & \sin(\theta) \\ 0 & 1 & 0 \\ -\sin(\theta) & 0 & \cos(\theta) \end{bmatrix} \quad (2.2)$$

c. Passage du repère R_j vers le repère R_1

La rotation se fait autour de l'axe $z_1 = z_j$. On passe du repère R_j vers le repère R_1 en faisant une rotation d'angle ψ appelé angle du lacet.

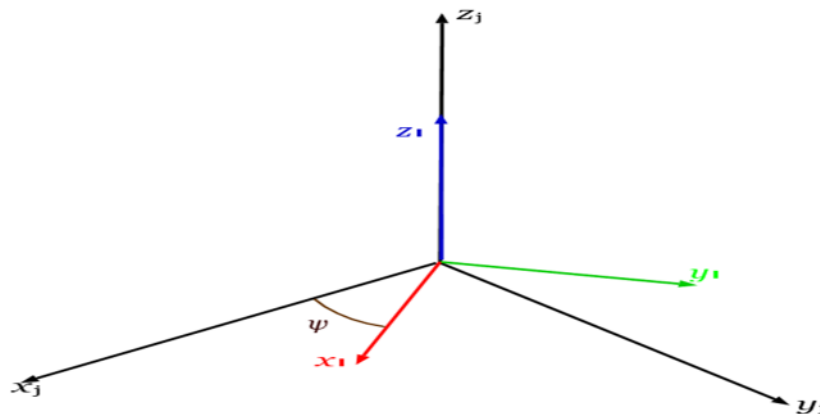


Figure II.14 : Rotation au tour de l'axe Z (Lacet)

Nous avons ainsi la matrice :

$$R(Z_0, \psi) = \begin{bmatrix} \cos(\psi) & -\sin(\psi) & 0 \\ \sin(\psi) & \cos(\psi) & 0 \\ 0 & 0 & 1 \end{bmatrix} \quad (2.3)$$

Le passage du repère R_1 vers le repère R_0 ou inversement se fait par trois rotations successives de telle sorte que tous les axes de R_1 occupent des positions différentes de celle de R_0 .

La matrice de passage de R_1 vers R_0 est donnée par le produit des trois matrices successives, on obtient :

$$\mathbf{R} = \mathbf{R}(\varphi, \theta, \psi) = R(Z_0, \psi) * R(Y_0, \theta) * R(X_0, \varphi)$$

Chapitre II : Principe de fonctionnement et modélisation d'un quadrirotor.

$$\mathbf{R} = \begin{bmatrix} C\theta C\psi & -S\psi C\varphi + S\varphi S\theta C\psi & S\varphi S\psi + C\varphi S\theta C\psi \\ C\theta S\psi & C\varphi C\psi + S\varphi S\theta S\psi & -S\varphi C\psi + C\varphi S\theta S\psi \\ -S\theta & S\varphi C\theta & C\varphi C\theta \end{bmatrix} \quad (2.4)$$

II.3.3 Paramètres cinématique :

Les paramètres cinématiques constituent l'état du système. L'état comprend quatre vecteurs de 3 éléments chacun définissant la position et la vitesse du centre de gravité (CG) ainsi que l'orientation et la vitesse angulaire de l'engin. Le CG est le centre instantané de rotation. Les paramètres sont donc définis comme suit :

- * $\mathbf{r}(t) = [x \ y \ z]$ est la position du CG par rapport à R0 et exprimé dans R0 .
- * $\mathbf{\Pi}(t) = [\varphi \ \theta \ \psi]_t$ le vecteur des angles d'Euler représentant la position angulaire du repère mobile R1 par rapport à R0 et exprimé dans R0;
- * $\mathbf{v}(t) = [u \ v \ w]_t$ est le vecteur de la vitesse linéaire du CG par rapport à R0 et

Exprimé dans R1.

- * $\mathbf{\Omega}(t) = [p \ q \ r]_t$ le vecteur vitesse de rotation instantané entre R0 et R1 et exprimé dans R1.

II.3.4 Cinématiques de translation :

On a défini la matrice de transformation, donc on peut trouver la vitesse de véhicule dans le trièdre inertiel à la place de trièdre lié au corps, cette matrice est donné comme suit :

$$\mathbf{V}_I(t) = \mathbf{R}(\psi, \theta, \varphi) \mathbf{V} \quad (2.5)$$

En particulier, la cinématique de translation de l'engin s'écrit :

$$\begin{bmatrix} \dot{x} \\ \dot{y} \\ \dot{z} \end{bmatrix} = \mathbf{R}(\psi, \theta, \varphi) \begin{bmatrix} u \\ v \\ w \end{bmatrix}$$

$$= \begin{bmatrix} (C\theta C\psi)u & (-S\psi C\varphi + S\varphi S\theta C\psi)v & (S\varphi S\psi + C\varphi S\theta C\psi)w \\ (C\theta S\psi)u & (C\varphi C\psi + S\varphi S\theta S\psi)v & (-S\varphi C\psi + C\varphi S\theta S\psi)w \\ (-S\theta)u & (S\varphi C\theta)v & (C\varphi C\theta)w \end{bmatrix} \quad (2.6)$$

II.3.5 Cinématique de rotation :

La cinématique de rotation exprime la relation entre les angles d'Euler $\Pi(t)$ et les composantes de la vectrice vitesse de rotation instantanée $\Omega(t)$. Il s'agit donc de projeter les axes x_i, y_j, z_1 dans le repère mobile R_1 , qui sont respectivement les vecteurs directeurs des angles φ, θ et ψ . Puisque $\Omega(t)$ s'écrit :

$$\Omega(t) = \dot{\varphi}x_i + \dot{\theta}y_j + \dot{\psi}z_1 \quad (2.8)$$

Il s'ensuit que :

Les composantes de x_i dans R_i sont les mêmes que dans R_1 , c'est à dire que :

$$X_i = x_1 = [0 \quad 0 \quad 1]^T \quad (2.9)$$

puisque les vecteurs $y_j = y_i$ sont identiques, alors $y_j = [0 \quad 1 \quad 0]^T_i$ et nous pouvons donc exprimer y_j dans R_1 c'est-à-dire :

$$y_j = R_i^T \begin{bmatrix} 0 \\ 1 \\ 0 \end{bmatrix}_i = \begin{bmatrix} 0 \\ \cos\varphi \\ -\sin\varphi \end{bmatrix} \quad (2.10)$$

le vecteur Z_1 a pour composant dans R_0 : $Z_1 = \begin{bmatrix} 0 \\ 0 \\ 1 \end{bmatrix}_0$ et donc dans R_1 :

$$Z_1 = R(\psi, \theta, \varphi)^{-1} \begin{bmatrix} 0 \\ 0 \\ 1 \end{bmatrix}_0 = \begin{bmatrix} -\sin\theta \\ \sin\varphi\cos\theta \\ \cos\varphi\cos\theta \end{bmatrix} \quad (2.11)$$

φ, θ et ψ

$$\text{avec :} \quad R(\psi, \theta, \varphi)^{-1} = R(\psi, \theta, \varphi)^T \quad (2.12)$$

On obtient :

$$\begin{bmatrix} p \\ q \\ r \end{bmatrix} = \begin{bmatrix} 1 & 0 & -\sin\theta \\ 0 & \cos\varphi & \sin\varphi\cos\theta \\ 0 & -\sin\varphi & \cos\varphi\cos\theta \end{bmatrix} \begin{bmatrix} \dot{\varphi} \\ \dot{\theta} \\ \dot{\psi} \end{bmatrix} \quad (2.13)$$

Chapitre II : Principe de fonctionnement et modélisation d'un quadrirotor.

On prend

$$Q = \begin{bmatrix} 1 & 0 & -\sin\theta \\ 0 & \cos\varphi & \sin\varphi\cos\theta \\ 0 & -\sin\varphi & \cos\varphi\cos\theta \end{bmatrix} \quad (2.14)$$

Donc :

$$Q^{-1} = \begin{bmatrix} 1 & \sin\varphi\tan\theta & \cos\theta\tan\theta \\ 0 & \cos\varphi & -\sin\varphi \\ 0 & \sin\varphi\sec\varphi & \cos\varphi\sec\theta \end{bmatrix} \quad (2.15)$$

$$\begin{bmatrix} \dot{\varphi} \\ \dot{\theta} \\ \dot{\psi} \end{bmatrix} = \begin{bmatrix} 1 & \sin\varphi\tan\theta & \cos\theta\tan\theta \\ 0 & \cos\varphi & -\sin\varphi \\ 0 & \sin\varphi\sec\varphi & \cos\varphi\sec\theta \end{bmatrix} \begin{bmatrix} p \\ q \\ r \end{bmatrix} \quad (2.16)$$

II.4 Modélisation dynamique avec le formalisme Newton-Euler

Les hypothèses suivantes sont prises en compte pour établir le modèle dynamique du Quadrirotor [32, 33] :

- La structure du Quadrirotor est supposée rigide et symétrique.
- Les hélices sont supposées rigides pour pouvoir négliger l'effet de déformation lors de la rotation (battement).
- La matrice d'inertie J est supposée constante (il n'ya pas de changement de poids).
- Les forces de portance et de trainée sont supposées proportionnelles au carré de la vitesse de rotation des rotors.
- Le repère lié au corps du Quadrirotor est supposé confondu avec son centre de gravité.

Sous ces hypothèses nous avons la possibilité de décrire le modèle dynamique du Quadrirotor en utilisant le formalisme de Newton-Euler.

II.4.1 Equations des forces :

D'après la première loi de la dynamique de Newton :

Chapitre II : Principe de fonctionnement et modélisation d'un quadrirotor.

$$\frac{d(mv)}{dt} = \sum F_{ext} \rightarrow \text{repere} \quad (2.17)$$

- F_{ext} = les forces appliquées aux corps du Quadrirotor
- m = la masse totale du Quadrirotor
- v = la vitesse linéaire de centre de gravité du Quadrirotor.

La vitesse exprimée dans le repère inertiel est :

$$\frac{d(mv)}{dt} = m\dot{v}$$

Les forces extérieures appliquées au Quadrirotor sont : [29,34]

a. Le poids :

La force de gravité est donnée par :

$$F_g = - m g z_0 \quad (2.18)$$

b. La force de portance :

Cette force résulte de la force de réaction, au sens de la loi de Newton. Elle est générée par la perturbation du flux d'air suite à la rotation de l'hélice. La force de poussée générée par l'hélice est donnée par :

$$T = \frac{1}{2} C_T \rho A (\omega r_b)^2 \quad (2.19)$$

D'où :

- C_T : terme aéronautique qui dépend directement de la géométrie de l'hélice.
- ρ : la densité de l'air.
- A : l'air de disque généré par l'hélice.
- ω : la vitesse de rotation de l'hélice.
- r_b : le rayon de l'hélice.

Chapitre II : Principe de fonctionnement et modélisation d'un quadrirotor.

Toutefois, comme on suppose que le Quadrirotor est un corps rigide, plusieurs termes sont considérés constants, ainsi on peut réécrire l'équation précédente :

$$T_i = b (\omega_i)^2 \quad (2.20)$$

D'où :

b : le coefficient de la poussée.

T_i : la force de la poussée produite par l'hélice i

c. La force de trainée :

C'est la résultante des forces qui s'opposent au mouvement du Quadrirotor dans l'air, de même direction que le mouvement du Quadrirotor mais de sens opposé. Elle représente en quelque sorte les forces de frottement visqueux sur l'objet, elle est donnée par :

$$F_t = K_{ft} v \quad (2.21)$$

- K_{ft} : le coefficient de trainée de translation .
- v : la vitesse linéaire.

À partir de l'équation (2.17) on obtient :

$$m \ddot{r} = F_f + F_t + F_g \quad (2.22)$$

D'où :

- m : la masse du système.
- r : le vecteur de position du Quadrirotor.
- F_f : est la force totale générée par les quatre rotors dans le repère terrestre.

$$F_f = R(\psi, \theta, \varphi) \cdot \begin{bmatrix} 0 \\ 0 \\ T \end{bmatrix} \quad (2.23)$$

D'où :

- T : est la force de poussée totale tel que :

Chapitre II : Principe de fonctionnement et modélisation d'un quadrirotor.

$$T = b (\omega_1^2 + \omega_2^2 + \omega_3^2 + \omega_4^2) \quad (2.24)$$

- F_t : la force de trainée selon $[x \ y \ z]$ est donnée par :

$$F_t = \begin{bmatrix} -K_{ftx} & 0 & 0 \\ 0 & -K_{fty} & 0 \\ 0 & 0 & -K_{ftz} \end{bmatrix} \dot{r} \quad (2.25)$$

- K_{ftx} , K_{fty} , K_{ftz} : les coefficients de trainée de translation .
- F_g : la force de gravité est donnée par :

$$F_g = \begin{bmatrix} 0 \\ 0 \\ -mg \end{bmatrix} \quad (2.26)$$

On remplace (2.23), (2.25), (2.26) dans l'équation (2.22), on aura :

$$m \begin{bmatrix} \ddot{x} \\ \ddot{y} \\ \ddot{z} \end{bmatrix} = R(\psi, \theta, \varphi) \cdot \begin{bmatrix} 0 \\ 0 \\ T \end{bmatrix} - \begin{bmatrix} K_{ftx} & \dot{x} \\ K_{fty} & \dot{y} \\ K_{ftz} & \dot{z} \end{bmatrix} + \begin{bmatrix} 0 \\ 0 \\ -mg \end{bmatrix} \quad (2.27)$$

$$m \begin{bmatrix} \ddot{x} \\ \ddot{y} \\ \ddot{z} \end{bmatrix} = \begin{bmatrix} C\varphi C\psi S\theta + S\varphi S\psi \\ C\varphi S\psi S\theta - S\varphi C\psi \\ C\varphi C\theta \end{bmatrix} T - \begin{bmatrix} K_{ftx} & \dot{x} \\ K_{fty} & \dot{y} \\ K_{ftz} & \dot{z} \end{bmatrix} + \begin{bmatrix} 0 \\ 0 \\ -mg \end{bmatrix} \quad (2.28)$$

- Avec : $C = \cos$, et $S = \sin$.

Les équations de mouvements de translation

$$\begin{cases} \ddot{x} = \frac{T}{m} (C\varphi C\psi S\theta + S\varphi S\psi) - \frac{K_{ftx}}{m} \dot{x} \\ \ddot{y} = \frac{T}{m} (C\varphi S\psi S\theta - S\varphi C\psi) - \frac{K_{fty}}{m} \dot{y} \\ \ddot{z} = \frac{T}{m} C\varphi C\theta - \frac{K_{ftz}}{m} \dot{z} - g \end{cases} \quad (2.29)$$

II.4.2 Equation des moments :

- L'équation du mouvement de rotation est dérivée du repère mobile, en utilisant le formalisme de Newton-Euler on obtient :

$$\bullet \sum_{ext} \tau = \frac{d(J\Omega)}{dt} \quad (2.30)$$

Et comme la vitesse angulaire est exprimé dans le repère lié au Quadrirotor, alors :

$$\frac{d(J\Omega)}{dt} = J \dot{\Omega} + \Omega \times J\Omega \quad (2.31)$$

Avec :

- $J \dot{\Omega} + \Omega \times J\Omega$: représente la dynamique des mouvements rotatifs elle est obtenue à partir de la variation du moment angulaire dans le repère du solide.
- J : la matrice d'inertie exprimée dans le repère R1 est une diagonale, car la structure du Quadrirotor est supposée symétrique :

$$J = \begin{bmatrix} J_x & 0 & 0 \\ 0 & J_y & 0 \\ 0 & 0 & J_z \end{bmatrix} \quad (2.32)$$

Les moments extérieurs appliqués sur le Quadrirotor sont :

a. Les couples aérodynamiques

L'étude des portances et trainées de chaque rotor nous aide à identifier les couples actifs et par conséquent, les mouvements de rotation du Quadrirotor dans le repère mobile. La portance générée par les quatre moteurs en rotation permet à l'engin de s'élever si elle compense au moins la trainée.

1. La différence entre les portances gauche et droite produisent le couple τ_{φ}^a qui est responsable du mouvement de roulis il s'écrit :

Chapitre II : Principe de fonctionnement et modélisation d'un quadrirotor.

$$\begin{aligned}
 \tau_{\varphi}^a &= F_4 L - F_2 L \\
 &= b (\omega_4^2) L - b (\omega_2^2) L \\
 &= b L (\omega_4^2 - \omega_2^2)
 \end{aligned} \tag{2.33}$$

Avec :

- L : La distance entre le centre de gravité du Quadrirotor et l'axe du rotor.
2. Le mouvement de tangage est produit par le couple τ_{θ}^a causé par la dissymétrie entre les portances produites par le rotor avant et arrière est :

$$\begin{aligned}
 \tau_{\theta}^a &= F_3 L - F_1 L \\
 &= b (\omega_3^2) L - b (\omega_1^2) L \\
 &= b L (\omega_3^2 - \omega_1^2)
 \end{aligned} \tag{2.34}$$

3. Le mouvement de lacet est le résultat de la somme des couples résistants générés par les forces F1, F2, F3 et F4 autour du centre de l'engin. Ce mouvement crée un effet de traînée qui se traduit par les frottements de l'air sur le Quadrirotor. La traînée qui est parallèle et opposée à la trajectoire du Quadrirotor.

$$\begin{aligned}
 \tau_{\psi}^a &= M1 - M2 + M3 - M4 \\
 &= d (\omega_1^2) - d (\omega_2^2) + d (\omega_3^2) - d (\omega_4^2) \\
 &= d (\omega_1^2 - \omega_2^2 + \omega_3^2 - \omega_4^2)
 \end{aligned} \tag{2.35}$$

- d : coefficient de traînée.

En regroupant les équations (2.33), (2.34), (2.35) dans le même vecteur on obtient :

$$\tau^a = \begin{bmatrix} \tau_{\varphi}^a \\ \tau_{\theta}^a \\ \tau_{\psi}^a \end{bmatrix} = \begin{bmatrix} bL(\omega_4^2 - \omega_2^2) \\ bL(\omega_3^2 - \omega_1^2) \\ d(\omega_1^2 - \omega_2^2 + \omega_3^2 - \omega_4^2) \end{bmatrix} \tag{2.36}$$

Chapitre II : Principe de fonctionnement et modélisation d'un quadrirotor.

b. Le Moment Gyroscopique :

Moment gyroscopique se définit comme la difficulté de modifier la position ou l'orientation du plan de rotation d'une masse tournante (sens négatif). Physiquement, l'effet gyroscopique est une conséquence directe de l'énergie cinétique de rotation, cette grandeur étant proportionnelle à la masse, au carré de son rayon et au carré de sa vitesse de rotation [20].

Il s'agit du moment résultant de l'application d'un couple perpendiculairement à l'axe de rotation d'un objet en mouvement. Le moment gyroscopique engendré est orienté suivant la perpendiculaire au couple appliqué et à l'axe de rotation de l'objet, de sorte à former le trièdre des vecteurs. (« Il se crée dans les systèmes physiques en mouvement avec des parties rotatoires et tend à résister aux mouvements du Quadrirotor »).

Pour notre cas d'étude, ce couple s'applique suite à la perturbation de vitesse de rotation de l'hélice par les mouvements rotatifs du Quadrirotor :

$$M_G = \Omega \times J_r [0 \quad 0 \quad \omega_r]^T \quad (2.37)$$

Avec :

- ω_r : la vitesse résiduelle des moteurs définie par :

$$\omega_r = \omega_1 - \omega_2 + \omega_3 - \omega_4 \quad (2.38)$$

- J_r : représente l'inertie du rotor.
- M_G : le moment gyroscopique généré par l'hélice.

c. Moment de trainée :

La même chose que la force de trainée, en raison de la friction de l'air il y a un moment de trainée résultant des frottements aérodynamiques qui agit sur le corps du Quadrirotor et qui est donné par :

Chapitre II : Principe de fonctionnement et modélisation d'un quadrirotor.

$$M_a = K_{fa} \Omega^2 = \begin{bmatrix} K_{fax} & \dot{\phi}^2 \\ K_{fay} & \dot{\theta}^2 \\ K_{faz} & \dot{\psi}^2 \end{bmatrix} \quad (2.39)$$

Avec :

- K_{fa} : le coefficient de frottement aérodynamique
- Ω : la Vitesse angulaire
- K_{fax} , K_{fay} , K_{faz} : les coefficients de frottement aérodynamiques.

Dans ce cas l'équation (2.31) peut être réécrite de la façon suivante :

$$J \dot{\Omega} = \tau^a - M_G - M_a - \Omega \times J \Omega \quad (2.40)$$

On remplace chaque moment par sa formule dans l'équation (2.40), on obtient :

$$\begin{bmatrix} J_x & 0 & 0 \\ 0 & J_y & 0 \\ 0 & 0 & J_z \end{bmatrix} \begin{bmatrix} \ddot{\phi} \\ \ddot{\theta} \\ \ddot{\psi} \end{bmatrix} =$$

$$\begin{bmatrix} bL(\omega_4^2 - \omega_2^2) \\ bL(\omega_3^2 - \omega_1^2) \\ d(\omega_1^2 - \omega_2^2 + \omega_3^2 - \omega_4^2) \end{bmatrix} - \begin{bmatrix} \dot{\phi} \\ \dot{\theta} \\ \dot{\psi} \end{bmatrix} \times \begin{bmatrix} 0 \\ 0 \\ J_r \omega_r \end{bmatrix} - \begin{bmatrix} K_{fax} & \dot{\phi}^2 \\ K_{fay} & \dot{\theta}^2 \\ K_{faz} & \dot{\psi}^2 \end{bmatrix} - \begin{bmatrix} \dot{\phi} \\ \dot{\theta} \\ \dot{\psi} \end{bmatrix} \times \begin{bmatrix} J_x & 0 & 0 \\ 0 & J_y & 0 \\ 0 & 0 & J_z \end{bmatrix} \begin{bmatrix} \dot{\phi} \\ \dot{\theta} \\ \dot{\psi} \end{bmatrix}$$

$$\begin{bmatrix} J_x \ddot{\phi} \\ J_y \ddot{\theta} \\ J_z \ddot{\psi} \end{bmatrix} = \begin{bmatrix} bL(\omega_4^2 - \omega_2^2) \\ bL(\omega_3^2 - \omega_1^2) \\ d(\omega_1^2 - \omega_2^2 + \omega_3^2 - \omega_4^2) \end{bmatrix} - \begin{bmatrix} \dot{\theta} J_r \omega_r \\ -\dot{\phi} J_r \omega_r \\ 0 \end{bmatrix} - \begin{bmatrix} K_{fax} & \dot{\phi}^2 \\ K_{fay} & \dot{\theta}^2 \\ K_{faz} & \dot{\psi}^2 \end{bmatrix} - \begin{bmatrix} \dot{\theta} \dot{\psi} (J_z - J_y) \\ \dot{\psi} \dot{\phi} (J_x - J_z) \\ \dot{\phi} \dot{\theta} (J_y - J_x) \end{bmatrix} \quad (2.41)$$

Les équations de mouvement de rotation

Chapitre II : Principe de fonctionnement et modélisation d'un quadrirotor.

$$\begin{cases} \ddot{\varphi} = \left(\frac{J_{yy}-J_{zz}}{J_x}\right)\dot{\theta}\dot{\psi} - \frac{J_r}{J_{xx}}\omega_r\dot{\theta} - \frac{K_{fax}}{J_{xx}}\dot{\varphi}^2 + \frac{L}{J_{xx}}b(\omega_4^2 - \omega_2^2) \\ \ddot{\theta} = \left(\frac{J_{zz}-J_{xx}}{J_y}\right)\dot{\varphi}\dot{\psi} + \frac{J_r}{J_{yy}}\omega_r\dot{\varphi} - \frac{K_{fay}}{J_{yy}}\dot{\theta}^2 + \frac{L}{J_{yy}}b(\omega_3^2 - \omega_1^2) \\ \ddot{\psi} = \left(\frac{J_{xx}-J_{yy}}{J_{zz}}\right)\dot{\varphi}\dot{\theta} - \frac{K_{faz}}{J_{zz}}\dot{\psi}^2 + \frac{d}{J_{zz}}(\omega_1^2 - \omega_2^2 + \omega_3^2 - \omega_4^2) \end{cases} \quad (2.42)$$

Au final le modèle dynamique de notre système représenté par les accélérations linéaires (2.29) et angulaires (2.42) est donné par :

$$\begin{cases} \ddot{x} = \frac{T}{m}(C\varphi C\psi S\theta + S\varphi S\psi) - \frac{K_{ftx}}{m}\dot{x} \\ \ddot{y} = \frac{T}{m}(C\varphi S\psi S\theta - S\varphi C\psi) - \frac{K_{fty}}{m}\dot{y} \\ \ddot{z} = \frac{T}{m}C\varphi C\theta - \frac{K_{ftz}}{m}\dot{z} - g \\ \ddot{\varphi} = \left(\frac{J_{yy}-J_{zz}}{J_{xx}}\right)\dot{\theta}\dot{\psi} - \frac{J_r}{J_{xx}}\omega_r\dot{\theta} - \frac{K_{fax}}{J_{xx}}\dot{\varphi}^2 + \frac{L}{J_{xx}}b(\omega_4^2 - \omega_2^2) \\ \ddot{\theta} = \left(\frac{J_{zz}-J_{xx}}{J_y}\right)\dot{\varphi}\dot{\psi} + \frac{J_r}{J_{yy}}\omega_r\dot{\varphi} - \frac{K_{fay}}{J_{yy}}\dot{\theta}^2 + \frac{L}{J_{yy}}b(\omega_3^2 - \omega_1^2) \\ \ddot{\psi} = \left(\frac{J_{xx}-J_{yy}}{J_{zz}}\right)\dot{\varphi}\dot{\theta} - \frac{K_{faz}}{J_{zz}}\dot{\psi}^2 + \frac{d}{J_{zz}}(\omega_1^2 - \omega_2^2 + \omega_3^2 - \omega_4^2) \end{cases} \quad (2.43)$$

II.5 La Représentation d'état du Système :

Lorsque on l'on envisage la commande d'un système, la première étape consiste à le modéliser. Modéliser un système consiste à élaborer une présentation mathématique qui permette de décrire et prédire son comportement dynamique et permanent lorsqu'il est soumis à des influences extérieures (entrées de commande, perturbation...)

Chapitre II : Principe de fonctionnement et modélisation d'un quadrirotor.

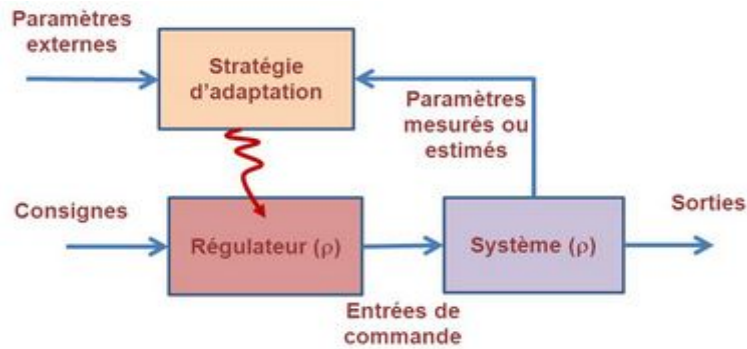


Figure II.15 : commande des systèmes.

Il s'agit de décrire un système en considérant sa dynamique interne et pas seulement une relation entre son entrée et sa sortie. Ainsi, il convient de « redonner de l'importance » à des grandeurs qui ne sont ni l'entrée, ni la sortie, tout en prenant en compte l'ensemble des phénomènes dynamiques et statiques qui confèrent au système son comportement.

Résoudre les équations d'état consiste à déterminer l'expression du vecteur d'état en fonction du temps [36, 37]

Dans notre cas on considère : [29, 35, 38, 39]

II.5.1 Vecteur d'état :

On définit le vecteur d'état du Quadrirotor comme suit :

$$X = [x_1 \ x_2 \ x_3 \ x_4 \ x_5 \ x_6 \ x_7 \ x_8 \ x_9 \ x_{10} \ x_{11} \ x_{12}]^T$$

Ces vecteurs définissent les degrés de liberté de Quadrirotor :

$$X = [\varphi \ \dot{\varphi} \ \theta \ \dot{\theta} \ \psi \ \dot{\psi} \ x \ \dot{x} \ y \ \dot{y} \ z \ \dot{z}]^T$$

Le vecteur d'état représente la position du Quadrirotor dans l'espace ainsi que la vitesse angulaire et linéaire.

Chapitre II : Principe de fonctionnement et modélisation d'un quadrirotor.

II.5.2 Vecteur de commande :

Le vecteur de commande est constitué de 4 entrées qui sont définies comme suit : la poussée (U1), le moment de roulis (U2), le moment de tangage (U3) et le moment de lacet (U4).

$$U = [U_1 \quad U_2 \quad U_3 \quad U_4]$$

Avec :

$$U_1 = b (\omega_1^2 + \omega_2^2 + \omega_3^2 + \omega_4^2)$$

$$U_2 = b (\omega_4^2 - \omega_2^2)$$

$$U_3 = b (\omega_3^2 - \omega_1^2)$$

$$U_4 = b (\omega_1^2 - \omega_2^2 + \omega_3^2 - \omega_4^2)$$

En regroupant ces équations dans une matrice on aura :

$$\begin{bmatrix} U_1 \\ U_2 \\ U_3 \\ U_4 \end{bmatrix} = \begin{bmatrix} b & b & b & b \\ 0 & -lb & 0 & lb \\ -lb & 0 & lb & 0 \\ d & -d & d & -d \end{bmatrix} \begin{bmatrix} \omega_1^2 \\ \omega_2^2 \\ \omega_3^2 \\ \omega_4^2 \end{bmatrix} \quad (2.44)$$

En utilisant les équations du mouvement de translation et de rotation on peut obtenir le modèle mathématique complet du Quadrirotor sous forme d'espace d'état :

$$\left\{ \begin{array}{l} \dot{x}_1 = \varphi = x_2 \\ \dot{x}_2 = \dot{\varphi} = a_1 x_4 x_6 + a_2 x_2^2 + a_3 x_4 \omega_r + b_1 U_2 \\ \dot{x}_3 = \theta = x_4 \\ \dot{x}_4 = \dot{\theta} = a_4 x_2 x_6 + a_5 x_4^2 + a_6 x_2 \omega_r + b_2 U_3 \\ \dot{x}_5 = \psi = x_6 \\ \dot{x}_6 = \dot{\psi} = a_7 x_2 x_4 + a_8 x_6^2 + b_3 U_4 \\ \dot{x}_7 = x = x_8 \\ \dot{x}_8 = \dot{x} = a_9 x_8 + \frac{U_1}{m} (\sin x_1 \sin x_5 + \cos x_1 \sin x_3 \cos x_5) \\ \dot{x}_9 = y = x_{10} \\ \dot{x}_{10} = \dot{y} = a_{10} x_{10} + \frac{U_1}{m} (\cos x_1 \sin x_3 \sin x_5 - \sin x_1 \cos x_5) \\ \dot{x}_{11} = z = x_{12} \\ \dot{x}_{12} = \dot{z} = a_{11} x_{12} + \frac{U_1}{m} (\cos x_1 \cos x_3) - g \end{array} \right. \quad (2.45)$$

Chapitre II : Principe de fonctionnement et modélisation d'un quadrirotor.

Avec :

$$\left\{ \begin{array}{l} a_1 = \left(\frac{J_{yy} - J_{zz}}{J_{xx}} \right) ; a_2 = \frac{-K_{fax}}{J_{xx}} ; a_3 = \frac{-J_r}{J_{xx}} ; a_4 = \left(\frac{J_{zz} - J_{xx}}{J_{yy}} \right) \\ a_5 = \left(\frac{-K_{fay}}{J_{yy}} \right) ; a_6 = \frac{J_r}{J_{yy}} ; a_7 = \left(\frac{J_{xx} - J_{yy}}{J_{zz}} \right) ; a_8 = \frac{-K_{faz}}{J_{zz}} \\ a_9 = \frac{-K_{ftx}}{m} ; a_{10} = \frac{-K_{f ty}}{m} ; a_{11} = \frac{-K_{ftz}}{m} \end{array} \right.$$

(2.46)

Et :

$$\left\{ \begin{array}{l} b_1 = \frac{L}{J_{xx}} \\ b_2 = \frac{L}{J_{yy}} \\ b_3 = \frac{1}{J_{zz}} \end{array} \right. \quad (2.47)$$

On pose :

$$C_x = (\sin x_1 \sin x_5 + \cos x_1 \sin x_3 \cos x_5) .$$

$$C_y = (\cos x_1 \sin x_3 \sin x_5 - \sin x_1 \cos x_5) . \quad (2.48)$$

$$C_z = (\cos x_1 \cos x_3) .$$

On aura le système suivant :

Chapitre II : Principe de fonctionnement et modélisation d'un quadrirotor.

$$f(X,U) = \begin{bmatrix} x_2 \\ a_1 x_4 x_6 + a_2 x_2^2 + a_3 x_4 \omega_r + b_1 U_2 \\ x_4 \\ a_4 x_2 x_6 + a_5 x_4^2 + a_6 x_2 \omega_r + b_2 U_3 \\ x_6 \\ a_7 x_2 x_4 + a_8 x_6^2 + b_3 U_4 \\ x_8 \\ a_9 x_8 + \frac{U_1}{m} C_x \\ x_{10} \\ a_{10} x_{10} + \frac{U_1}{m} C_y \\ x_{12} \\ a_{11} x_{12} + \frac{U_1}{m} C_z \end{bmatrix} \quad (2.49)$$

Le modèle dynamique de notre Quadricoptère contient deux effets gyroscopiques, l'influence de ces effets est dans notre cas moins important que ceux de la rotation des moteurs, ce qui va simplifier notre modèle. Donc les commandes U2, U3, U4 sont exprimées comme suit :

$$\begin{cases} U_2 = \ddot{\phi} \times J_{xx} \\ U_3 = \ddot{\theta} \times J_{yy} \\ U_4 = \ddot{\psi} \times J_{zz} \end{cases} \quad (2.50)$$

L'expression de U1 est déterminée depuis l'équation dynamique de translation \mathbf{Z}'' :

$$U_1 = \frac{m(\ddot{z}+g)}{\cos\varphi+\cos\theta} \quad (2.51)$$

Après avoir exprimé les commandes nous allons les utiliser pour déterminer les expressions des vitesses de rotation des moteurs : A partir de l'équation (2.44) nous obtenons :

Chapitre II : Principe de fonctionnement et modélisation d'un quadrirotor.

$$\begin{cases} \omega_1 = \sqrt{\left| \frac{U_1}{4b} - \frac{U_3}{2lb} - \frac{U_4}{4d} \right|} \\ \omega_2 = \sqrt{\left| \frac{U_1}{4b} - \frac{U_2}{2lb} + \frac{U_4}{4d} \right|} \\ \omega_3 = \sqrt{\left| \frac{U_1}{4b} + \frac{U_3}{2lb} - \frac{U_4}{4d} \right|} \\ \omega_4 = \sqrt{\left| \frac{U_1}{4b} + \frac{U_3}{2lb} + \frac{U_4}{4d} \right|} \end{cases} \quad (2.52)$$

Les commandes U1, U2, U3, U4 sont reconstruites à partir des expressions des vitesses des moteurs par l'équation (2.44).

II.5.3 Contraintes de sous actionnement :

Les contraintes de sous actionnement consistent à exprimer les angles φ d et θ d en fonction de ψ d et le vecteur d'accélération $(\mathbf{x}'' , \mathbf{y}'' , \mathbf{z}'')$ pour cela nous allons exploiter les équations dynamiques de translation :

Avec $\mathbf{U} = (\mathbf{x}'' , \mathbf{y}'' , \mathbf{z}'')$ l'équation (2.17) devient :

$$\mathbf{U} = -g_z + \frac{1}{m} \mathbf{R} \times \mathbf{T} \quad (2.53)$$

Avec :

$$\mathbf{T} = [0 \quad 0 \quad \sum_{i=1}^4 F_i]^T \quad \text{et} \quad \mathbf{g}_z = [0 \quad 0 \quad g]^T$$

En déplaçant le vecteur de gravité \mathbf{g}_z à gauche et en multipliant l'équation par \mathbf{R}^T on obtient :

$$\mathbf{R}^T (\mathbf{U} + \mathbf{g}_z) = \frac{1}{m} \mathbf{T} \quad (2.54)$$

Qui donne :

$$\begin{bmatrix} C\theta C\psi & -C\varphi S\psi + S\varphi S\theta C\psi & S\varphi S\psi + C\varphi S\theta C\psi \\ C\theta S\psi & C\varphi C\psi + S\varphi S\theta S\psi & -S\varphi C\psi + C\varphi S\theta S\psi \\ -S\theta & S\varphi C\theta & C\varphi C\theta \end{bmatrix}^T \begin{bmatrix} \ddot{x} \\ \ddot{y} \\ \ddot{z} + g \end{bmatrix} = \frac{1}{m} \begin{bmatrix} 0 \\ 0 \\ T \end{bmatrix} \quad (2.55)$$

$$\ddot{x} C\theta C\psi + \ddot{y} C\theta S\psi - (\ddot{z} + g)S\theta = 0 \quad (2.56)$$

$$\ddot{x} (-C\varphi S\psi + S\varphi S\theta C\psi) + \ddot{y} (C\varphi C\psi + S\varphi S\theta S\psi) + (\ddot{z} + g)S\varphi C\theta = 0 \quad (2.57)$$

Chapitre II : Principe de fonctionnement et modélisation d'un quadrirotor.

$$\ddot{x} (S\varphi S\psi + C\varphi S\theta C\psi) + \ddot{y}(-S\varphi C\psi + C\varphi S\theta S\psi) + (\ddot{z} + g)C\varphi C\theta \quad (2.58)$$

En considérant que $C\theta$ est différent de 0 on divise l'équation (2.56) par $C\theta$, l'expression de θ_d est la suivante :

$$\theta_d = \arctan \left(\frac{\ddot{x}C\psi + \ddot{y}S\psi}{(\ddot{z} + g)} \right) \quad (2.59)$$

On soustrait (2.57). $\cos \varphi$ de (2.58). $\sin \varphi$ pour obtenir :

$$\frac{1}{m} T \sin \varphi = \ddot{x} \sin \psi - \ddot{y} \cos \psi \quad (2.60)$$

En manipulant l'équation (2.54) nous obtenons la relation suivante :

$$(U + g_z)^T (U + g_z) = \left(\frac{1}{m}T\right)^T \left(\frac{1}{m}T\right) \quad (2.61)$$

Qui est équivalente à :

$$\ddot{x}^2 + \ddot{y}^2 + \ddot{z}^2 + g^2 = \left(\frac{1}{m}T\right)^2 \quad (2.62)$$

L'expression de φ_d est obtenue en combinant les équations (2.60) et (2.62) :

$$\varphi_d = \arcsin \left(\frac{\ddot{x} \sin \psi - \ddot{y} \cos \psi}{\sqrt{\ddot{x}^2 + \ddot{y}^2 + (\ddot{z} + g)^2}} \right) \quad (2.63)$$

II.6 Conclusion :

Dans ce chapitre nous avons fait une description détaillée sur les mouvements de base du quadri rotor, nous avons fait la modélisation de ce système selon le formalisme de newton-Euler afin de s'approcher le plus possible de la dynamique réelle du quadri rotor. A partir du modèle obtenu nous concluons que le quadri rotor est un système sous actionné. De plus, la complexité du modèle, la non linéarité et l'interaction entre les états du système, peuvent se voir clairement. Les algorithmes de contrôle pour stabiliser ce type d'engin continuent à se développer.

CHAPITRE III

Commande et simulation

III.1 Introduction :

A fin de pouvoir piloter et gérer un quadri rotor, plusieurs algorithmes de contrôle peuvent être appliqués pour construire des lois de commande.

Au cours de ce chapitre nous nous intéressons au système de contrôle d'un drone quadri rotor. Nous allons voir comment contrôler les différents mouvements en utilisant la technique PID tout en testant le modèle de simulation réalisé sous MATLAB qui permet de générer des lois de commande pour la stabilisation, cela facile d'évaluer les performances du système avec une approche mathématique.

Nous allons simuler le modèle expliqué dans le chapitre précédent, et discuter les différents résultats que nous avons obtenus avec les différentes valeurs. Ce chapitre a pour but d'illustrer la déstabilisation et la stabilisation du quadri rotor selon les valeurs du régulateur PID. Les résultats sont présentés à l'aide de l'outil scope sous SIMULIK ou la commande plot dans un script MATLAB.

III.2 Principe d'un régulateur PID :

Le régulateur PID (Proportionnelle, Intégrale, Dérivée) est le régulateur le plus utilisé dans l'industrie en raison de sa simplicité, sa robustesse est certainement l'approche la plus intuitive et la plus simple à embarquer sur un processeur. Elle permet de comprendre facilement le rôle physique de chacun des termes de la commande, ce qui permet de régler les gains en conséquence. [40]

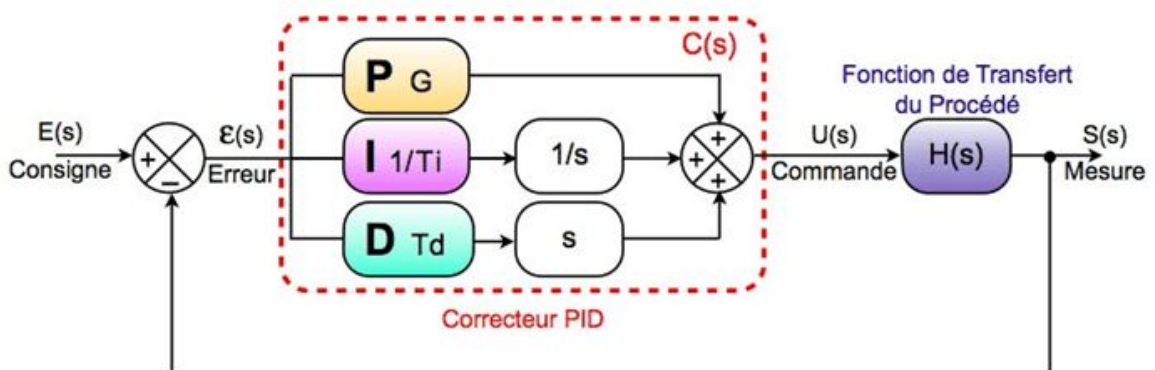


Figure III.1 : structure de la commande par PID

III.2.1 La composante proportionnelle :

L'action proportionnelle applique une correction instantanée pour tout écart entre la mesure et la consigne, plus la perturbation est grande, plus la correction apportée est grande. Cette composante seule ne permet pas une grande précision surtout dans les systèmes à faible inertie, comme dans le traitement de l'air, cette rapidité d'action engendre un phénomène appelé le pompage. [22]

Cette action est représentée comme suit :

$$C(t)=K_p \varepsilon(t) \tag{3.1}$$

K_p : gain proportionnel

$\varepsilon(t)$: erreur entre consigne et mesure ($\varepsilon(t)=\text{consigne} - \text{mesure}$)

$C(t)$: la commande (consigne)

Ce qui en Laplace donne

$$C(p)=K_p \varepsilon(p) \tag{3.2}$$

L'idée étant d'augmenter l'effet de l'erreur sur le système afin que celui-ci réagisse plus rapidement aux changements de consignes.

Plus la valeur de K_p est grande, plus la réponse l'est aussi. En revanche, la stabilité du système s'en trouve détériorée et dans le cas d'un démesuré le système peut même diverger.

L'action proportionnelle applique une correction instantanée pour tout écart entre la mesure et la consigne, plus la perturbation est grande, plus la correction apportée est grande. Cette composante seule ne permet pas une grande précision surtout dans les systèmes à faible inertie [41].

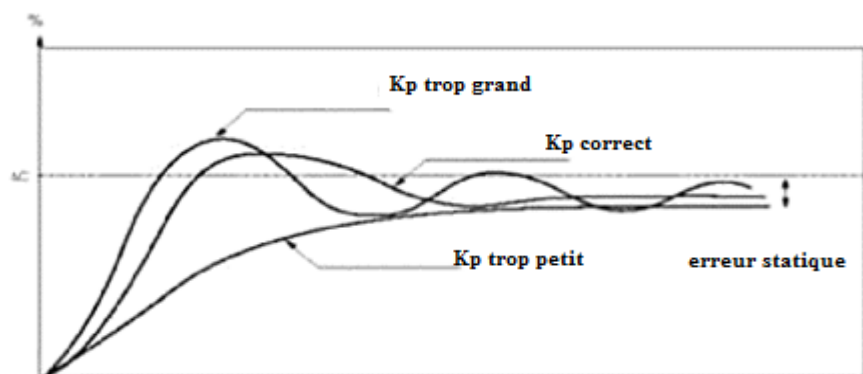


Figure III.2 : Régulateur proportionnel

III.2.2 La composante intégrale :

Cette composante apporte une notion de temps d'intégration à la correction, cette notion de temps s'exprime généralement en seconde. Cette action est complémentaire à l'action proportionnelle, elle permet de stabiliser dans le temps l'action proportionnelle et réduire voire éliminer l'erreur statique en régime permanent, plus l'erreur mesurée est constante plus la correction est constante. Sa relation est donnée comme suit :

$$C(t) = K_i \int_0^t \varepsilon(t) dt \quad (3.3)$$

III.2.3 La composante dérivée :

Cette action permet d'anticiper la réponse de la régulation en cas de perturbation rapide ou de modification de consigne ce qui améliore la stabilité du système. On peut donc dire que cette composante permet de compenser tout dépassement excessif de la consigne. Sa relation est donnée comme suit :

$$C(t) = K_d \frac{d\varepsilon(t)}{dt} \quad (3.4)$$

III.3 Régulateur proportionnel-intégrale PI :

Au contrôle proportionnel, nous pouvons ajouter l'intégration de l'erreur. Dans ce cas nous obtenons une régulation PI.

L'erreur entre la consigne et la mesure est ici intégrée par rapport au temps et multipliée par une constante qu'il faudra aussi déterminer en fonction du système.

$$U(t) = K_p \varepsilon(t) + K_i \int_0^t \varepsilon(t) dt \quad (3.5)$$

Ce qui en Laplace donne :

$$U(p) = K_p \varepsilon(p) + K_i \frac{\varepsilon(p)}{p} \quad (3.6)$$

Pourquoi a-t-on besoin de rajouter cette fonctionnalité à notre organe de contrôle ?

Parce que dans un simple contrôle proportionnel, il subsiste une erreur statique. Lorsque le système s'approche de sa consigne, l'erreur n'est plus assez grande pour faire avancer le moteur. Le terme intégral permet ainsi de compenser l'erreur statique et fournit, par conséquent, un système plus stable en régime permanent. Plus K_i est élevé l'erreur statique est élevée. Cette composante apporte une notion de temps d'intégration à la correction,

cette notion de temps s'exprime généralement en seconde. Cette action est complémentaire à l'action proportionnelle, elle permet de stabiliser dans le temps l'action proportionnelle, plus l'erreur mesurée est constante plus la correction est constante [41].

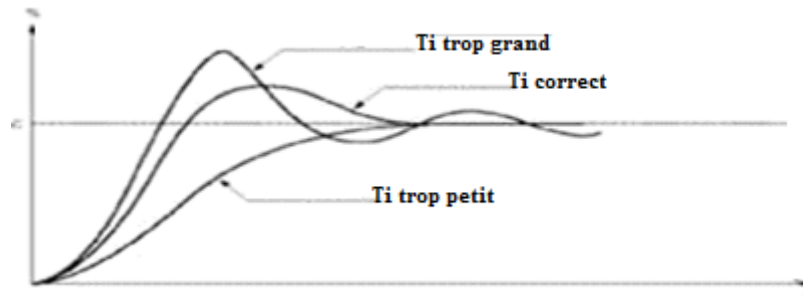


Figure III.3 : Régulateur proportionnel-intégrale

III.3.1 Le correcteur PID :

L'intérêt du correcteur PID est d'intégrer les effets positifs des trois correcteurs précédents.

La détermination des coefficients K_p , T_i , T_d du correcteur PID permet d'améliorer à la fois la précision (T_d et K_p) la stabilité (T_d) et la rapidité (T_d , K_p). Le réglage d'un correcteur PID est en général assez complexe, mais des méthodes pratiques de réglages permettent d'obtenir des bons résultats. Notez que ces corrélations peuvent ne pas être exactement précises, parce que K_p , K_i et K_d interdépendantes. En fait, changer un de ces variables peut changer l'effet des deux autres.

La loi de commande générale d'un correcteur PID est de la forme :

$$U(t) = K_p \epsilon(t) + K_I \int_0^t \epsilon(\tau) d\tau + K_d \frac{d\epsilon(t)}{dt} \quad (3.7)$$

En automatique, l'action du correcteur PID est plus généralement écrite dans le domaine de Laplace sous la forme d'une fonction de transfert entre l'écart $\epsilon(p)$ et le signal de commande $U(p)$. En effet, à partir de cette représentation, le comportement du système peut être étudié sans passer par une fonction temporelle souvent plus compliquée. La fonction de transfert classique d'un correcteur PID se met alors sous la forme :

$$H_{PID}(p) = \frac{U(p)}{\epsilon(p)} = K_p + \frac{K_I}{p} + K_d p \quad (3.8)$$

Cependant, il n'existe pas de méthode unique pour déterminer ces paramètres et il est alors nécessaire de réaliser des compromis entre durée du régime transitoire, dépassement, erreur statique.

Le tableau suivant présente l'influence des paramètres du PID sur le comportement du système en boucle fermée

Paramètre	Temps de montée	Temps de stabilisation	Dépassement	Erreur statique
K_p	diminue	augmente	augmente	diminue
k_I	diminue	augmente	augmente	annule
k_d		diminue	diminue	

Tableau III.1 : Influence d'augmenté les paramètres du PID sur le comportement du système en boucle fermée.

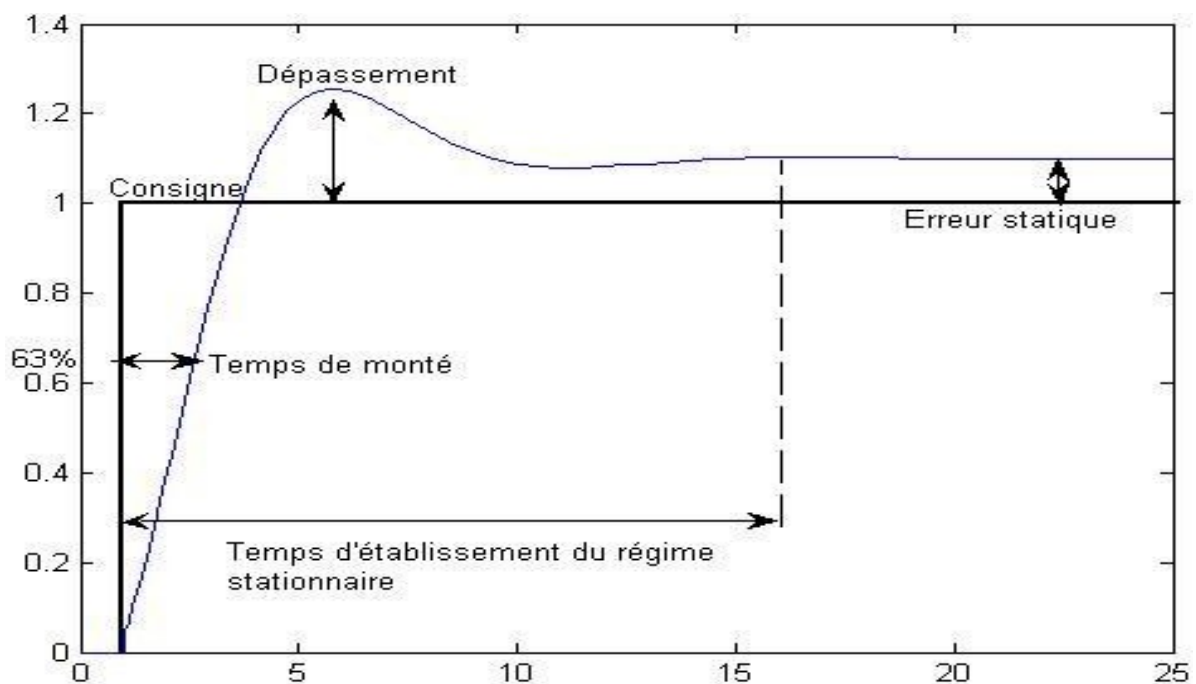
III.3.2 Définition de la réponse d'un système :

La réponse d'un système de fonction de transfert $G(p)$ correspond à la forme du signal de sortie $y(t)$ lorsque le système est excité par un signal d'entrée connu $u(t)$.

On distingue plusieurs types de réponse selon le signal d'entrée. Certaines réponses sont très couramment étudiées pour comprendre le fonctionnement d'un système. En toute logique, la réponse d'un système dépend de la valeur de la sortie à l'instant initial considéré.

Cependant, à des fins de simplification nous n'intéresserons qu'à des réponses pour lesquelles les conditions initiales sont nulles.

Figure III.4 : Caractéristiques de la réponse d'un système.



III.4 Modélisation sous SIMULINK du quadri rotor :

Après avoir modélisé le comportement du quadrirotor, nous passons à l'étape de la Simulation afin de valider notre modèle. Pour cela, on a opté d'utiliser MATLAB/Simulink.

A partir de l'expression des accélérations angulaire on établit un modèle sous Simulink : Modélisation de la poussée et la trainée + l'effet gyroscopique + l'effet d'inertie.

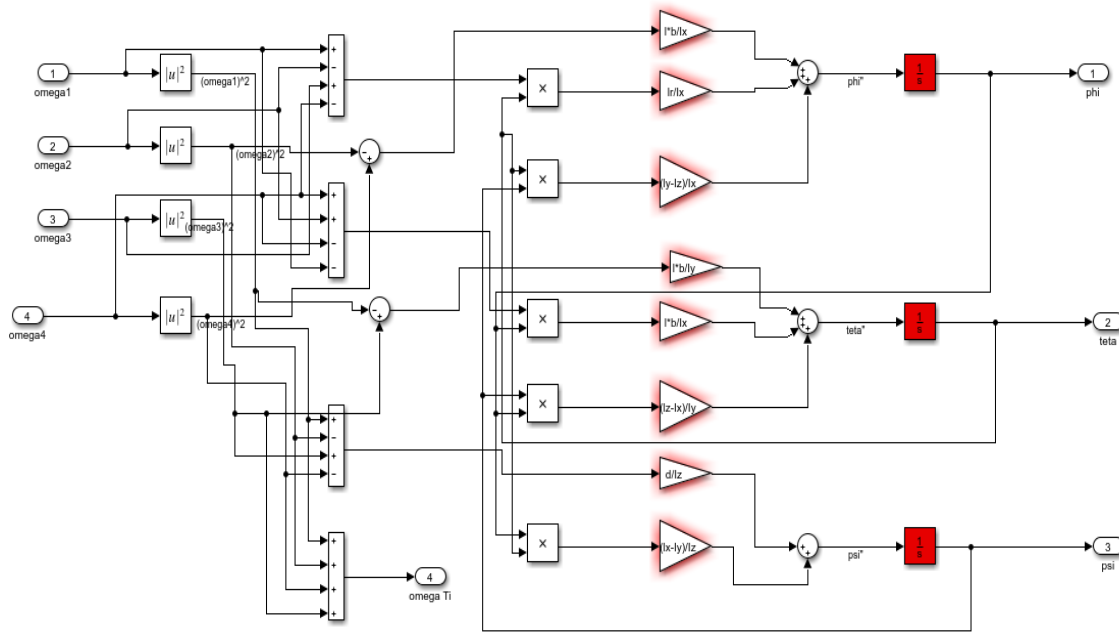


Figure III.5 : simulation des accélérations angulaires

Model mathématique des accélérations angulaires:

$$\begin{cases} \ddot{\phi} = \left(\frac{J_{yy} - J_{zz}}{J_x}\right)\dot{\theta}\dot{\psi} - \frac{J_r}{J_{xx}} \omega_r \dot{\theta} - \frac{K_{fax}}{J_{xx}} \dot{\phi}^2 + \frac{L}{J_{xx}} b(\omega_4^2 - \omega_2^2) \\ \ddot{\theta} = \left(\frac{J_{zz} - J_{xx}}{J_y}\right)\dot{\phi}\dot{\psi} + \frac{J_r}{J_{yy}} \omega_r \dot{\phi} - \frac{K_{fay}}{J_{yy}} \dot{\theta}^2 + \frac{L}{J_{yy}} b(\omega_3^2 - \omega_1^2) \\ \ddot{\psi} = \left(\frac{J_{xx} - J_{yy}}{J_{zz}}\right)\dot{\phi}\dot{\theta} - \frac{K_{faz}}{J_{zz}} \dot{\psi}^2 + \frac{d}{J_{zz}} (\omega_1^2 - \omega_2^2 + \omega_3^2 - \omega_4^2) \end{cases}$$

Et a partir les équations des accélérations suivant les axes (X, Y) et la position (Z) on établit le modèle sous Simulink suivant :

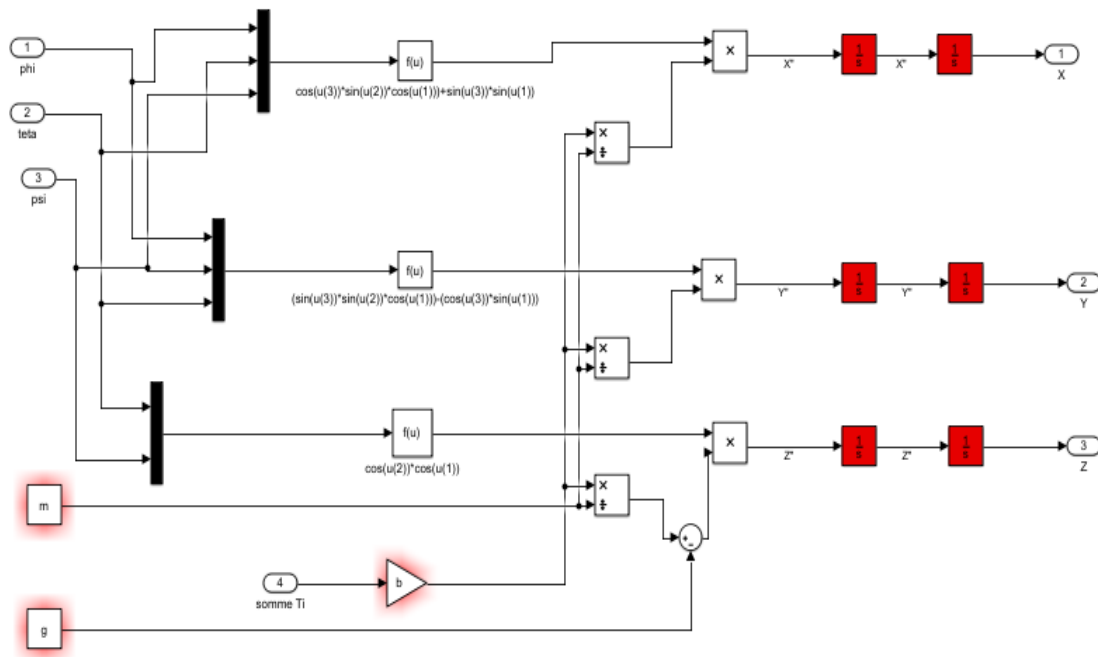


Figure III.6 : simulation de déplacement

Connaissant les relations entre les poussées des différents moteurs et les accélérations angulaires. On peut maintenant réaliser le modèle sur Simulink - Matlab, comme le montre la figure :

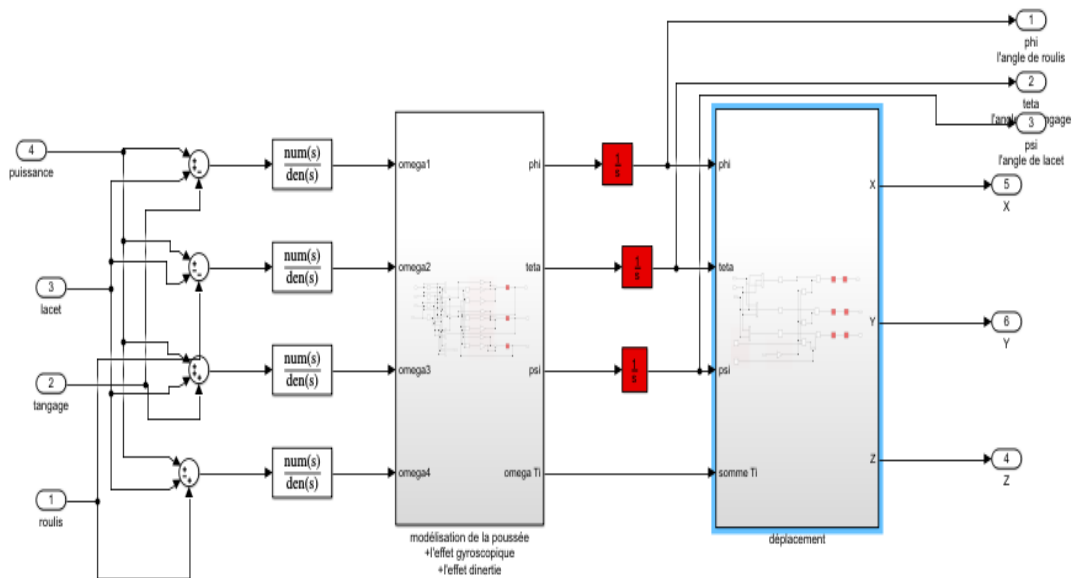


Figure III.7 : le modèle du quadri rotor.

Les paramètres mécaniques du Quadri rotor sont donnés dans le tableau suivant :

<i>Symbol</i>	<i>Désignation</i>	<i>Valeur et Unité</i>
g	l'accélération gravitationnelle	9.81 m/s ²
b	Le coefficient de portance	3.13 x10 ⁻⁵ N s / r a d
m	Masse du band d'essai	0.96 kg
J_x	Moment d'inertie suivant l'axe X	0.0086 kg.m ²
J_y	Moment d'inertie suivant l'axe Y	0.0086 kg. m ²
J_z	Moment d'inertie suivant l'axe Z	0.0172 kg. m ²
J_r	L'inertie du rotor	3.7404x10 ⁻⁵ kg. m ²
L	Distance entre le moteur et le C.G	0.0348 m

Tableau III.2 : les paramètres constant de quadri rotor

Les conditions initiales pour les angles et z :

Etat	Etat 1	Etat 2
angle		
Phi	0.4	-0.1
Teta	0.5	0.3
Psi	0.6	-0.2
z	0.8	0.6

Tableau III.3 : les conditions initiales pour les angles et z

Le signal de référence est donné comme suit :

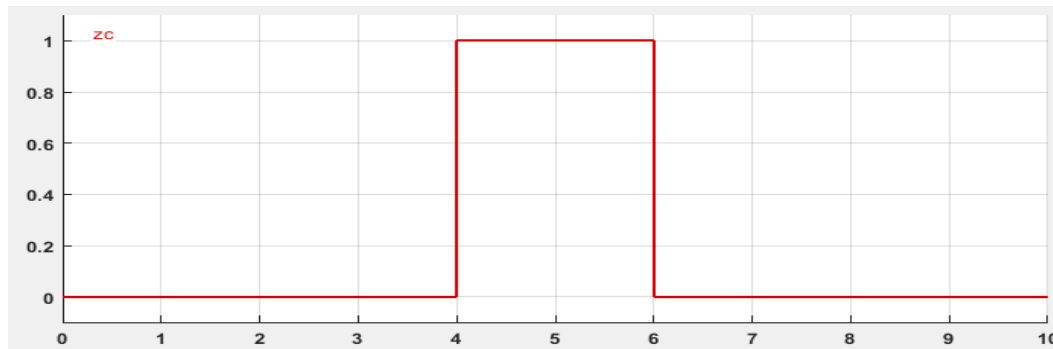


Figure III.8 : le signal de référence de la position «z»

III.4.1 La stabilité de quadri rotor

L'objectif du travail est d'assurer la stabilisation du banc d'essai en l'absence de commande extérieure : le banc d'essai doit se remettre en position horizontale.

Et le schéma suivant illustre le modèle de quadri rotor régulé avec la régulateur PID par une boucle fermée. Dans ce cas l'association procédée et instruments constitue un système asservi :

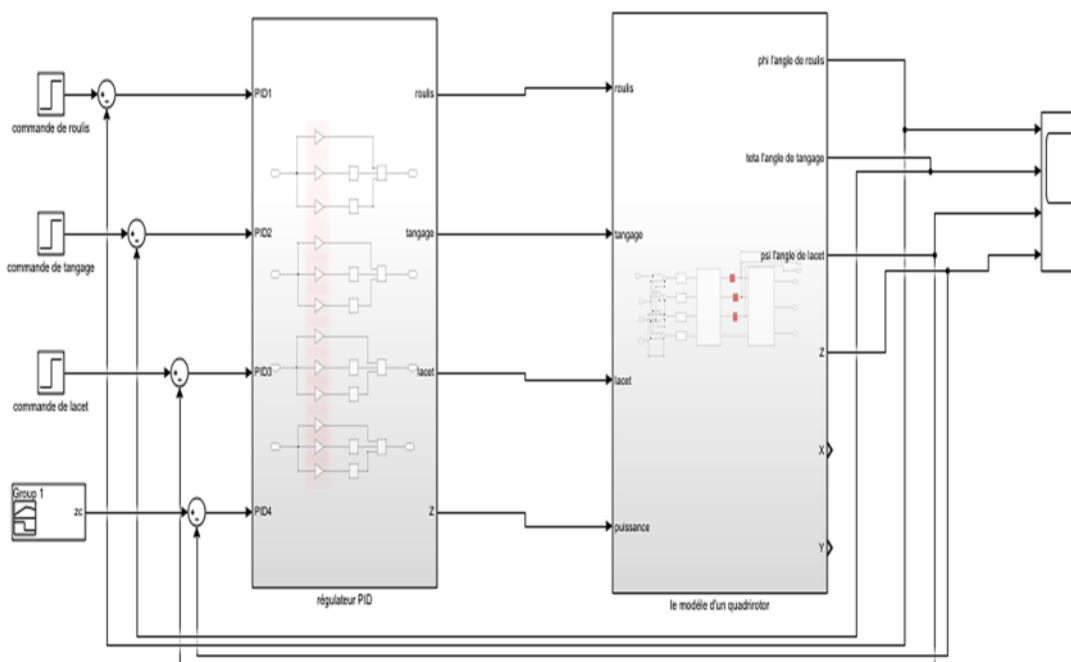


Figure III.9 : schéma simulink du régulateur PID

La figure (III.10) présente l'intérieur de la régulateur PID

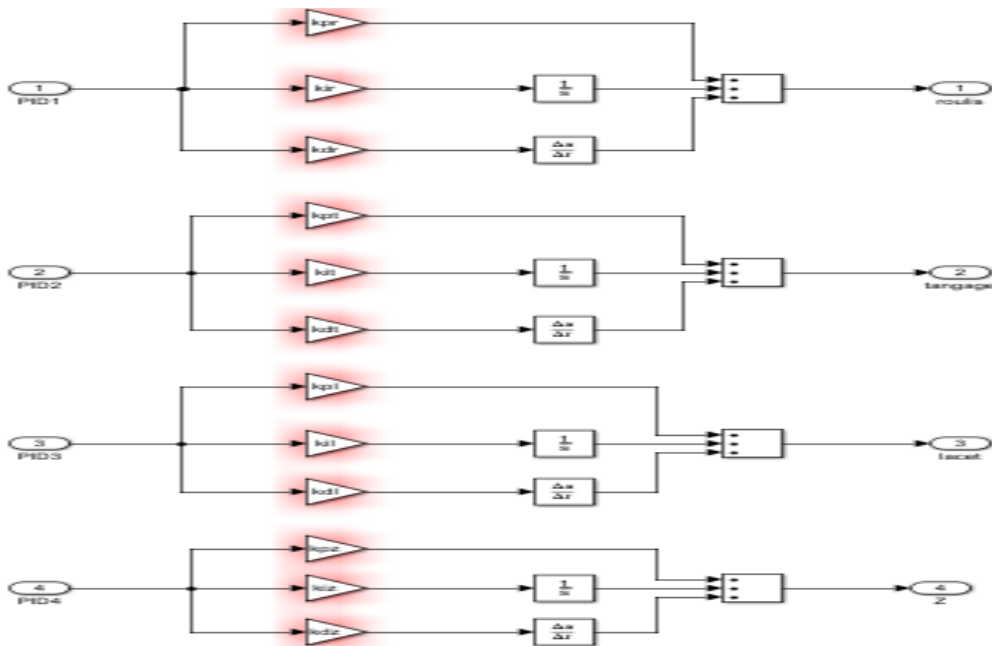


Figure III.10 : le bloc PID

Avec les paramètres : $k_p=5, k_i=0.006, k_d= 2$ pour lacet, tangage, roulis, et le mouvement verticale sur l'axe z , les Figures suivantes montres les résultats de la simulation du contrôleur PID.

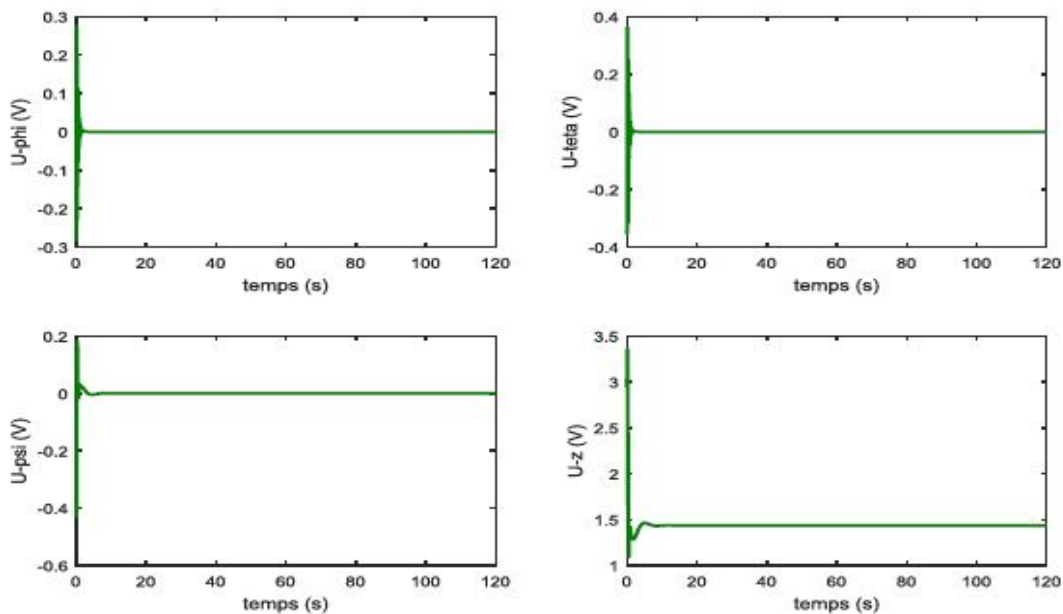


Figure III.11 : la sortie de la commande

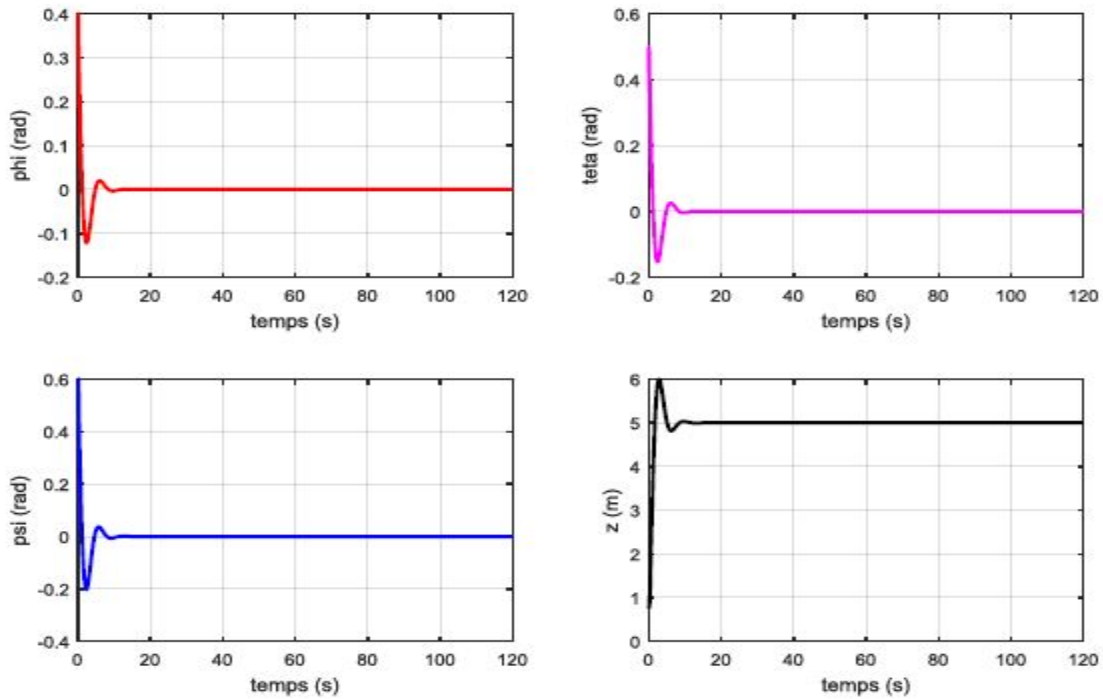


Figure III.12 : La sortie du système (etat1)

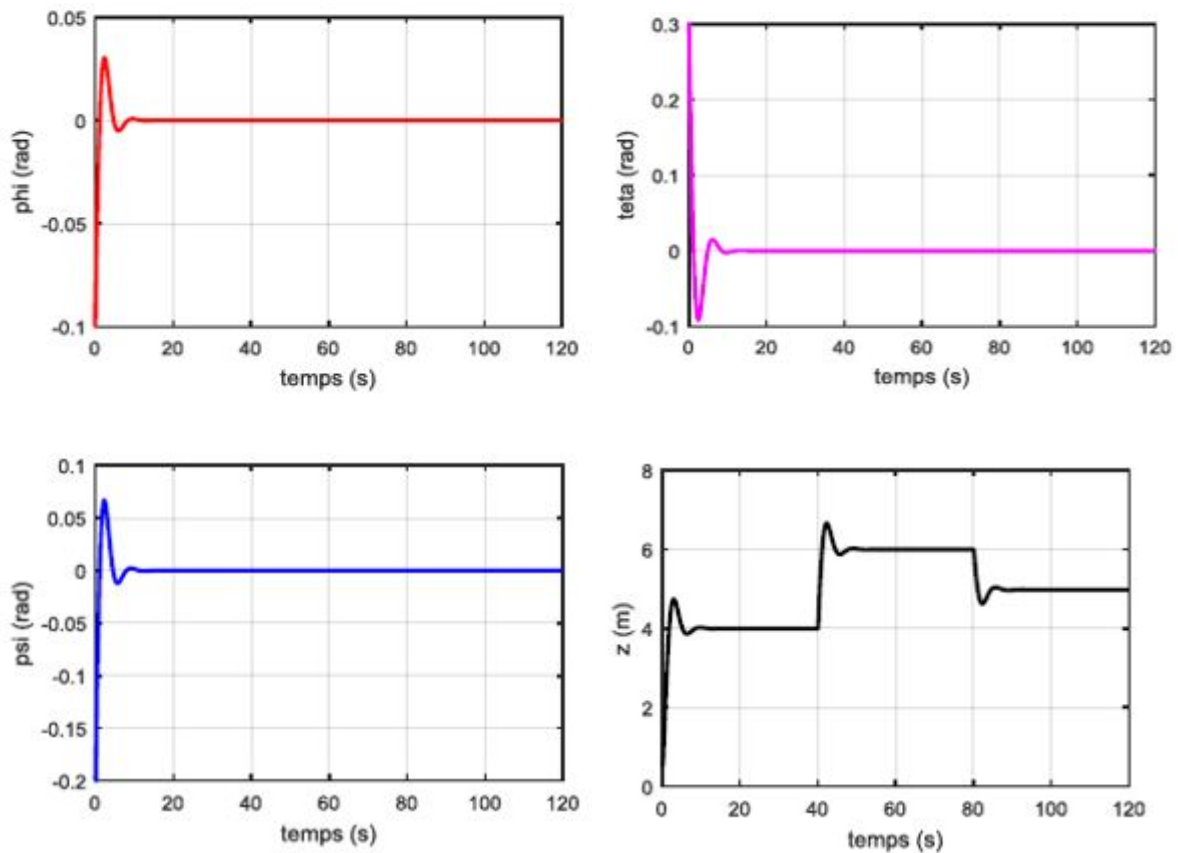


Figure III.13 : La sortie du système (etat2)

Interprétation :

Les figures III.13 montrent que la régulation PID est suffisante pour la stabilisation horizontale (en zéro) d'attitude du Quadri rotor et en même temps assure un temps de réponse très rapide et ne consomme pas beaucoup d'énergie autour d'un seul point de fonctionnement.

III.5 Conclusion :

Dans ce chapitre, nous avons étudié et présenté une technique de commande linéaire permettant le pilotage d'un véhicule aérien semi autonome. Cette technique est la commande par régulation PID.

Les résultats de simulation obtenus pour la stabilisation en altitude du quadri-rotor, montrent que l'utilisation de cette commande conduit à un meilleur comportement et donne des résultats intéressants en termes de performance.

A partir de ces résultats on peut réaliser notre banc d'essai et faire l'implémentation des paramètres de notre PID

CHAPITRE IV

Réalisation d'un banc d'essai

IV.1 Introduction :

Dans ce chapitre nous présentons les étapes de la réalisation du banc d'essai, notre objectif principal est de construire un banc d'essai facile à manipuler par des étudiants pour des projets pédagogiques. Ce travail est réalisé en plusieurs étapes. Nous commencerons par la présentation des éléments indispensables à sa réalisation par la suite nous passerons au calibrage du capteur et des ECS, à la fin nous allons faire l'implémentation des paramètres du PID pour stabiliser notre système.

IV.2 Eléments constituant le banc d'essai**IV.2.1 Composant mécanique :**

Les composants mécaniques ne nécessitent pas d'alimentation électrique.

a. Châssis :

C'est la structure qui va comporter tous les éléments du banc d'essai, c'est de ce châssis que dépend le comportement du drone. Un châssis ultraléger permet de gagner en autonomie. Des bras longs amélioreront la stabilité alors que des bras plus courts permettront des figures plus acrobatiques

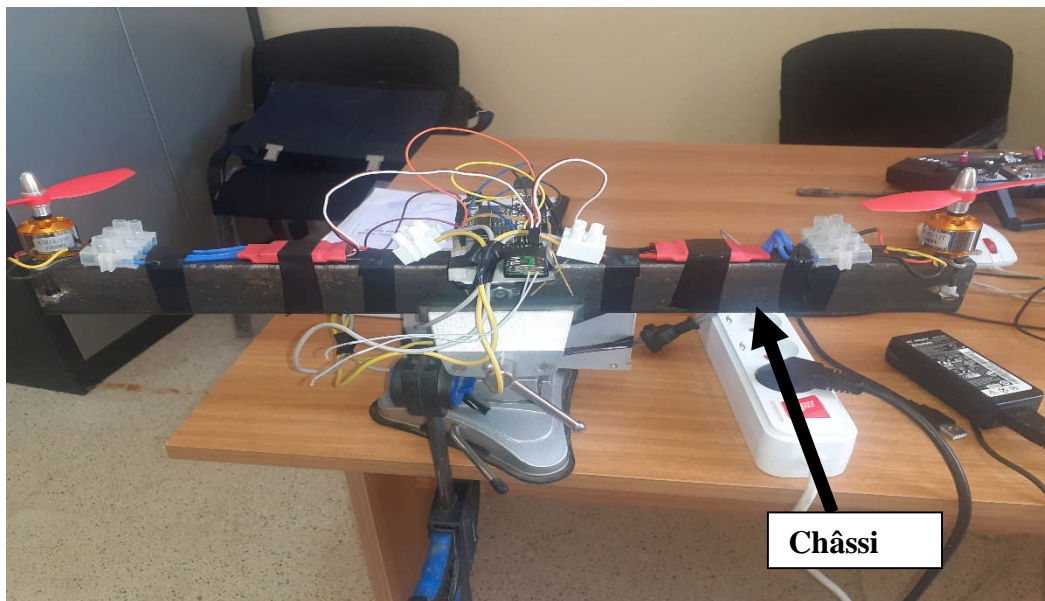


Figure IV.1 : les châssis

b. Hélices :

L'hélice est un organe mécanique entraîné par le moteur et assure la propulsion de l'aéronef. Lorsqu'elle se met en rotation, les pales forment un disque rotor. C'est alors leur forme aérodynamique qui produit entre les deux faces du disque une différence de pression perpendiculaire au disque et centrée sur son axe [42].

La poussée produite par une hélice dépend de la densité de l'air, de son diamètre, de la forme et de la surface des pales et de son pas. C'est pourquoi il est très judicieux de choisir une hélice adéquate à notre quadri-rotor selon le type de vol que nous voulons réaliser.

La figure ci-dessous (**Figure IV.1**) illustre les deux caractéristiques de l'hélice et leurs effets sur le comportement du quadri-rotor

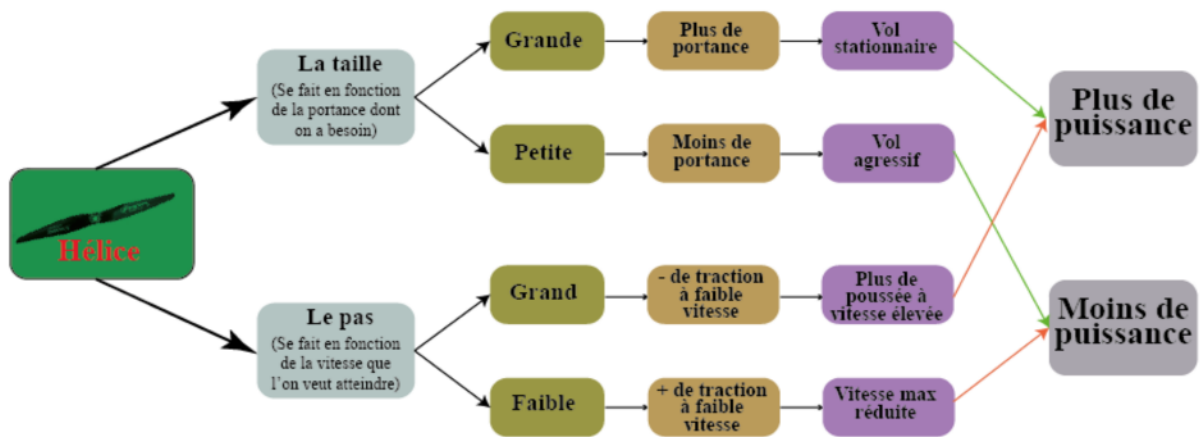


Figure IV.2 : caractéristique de l'hélice

Un autre critère sur le choix des hélices est leur composition. Il existe plusieurs matériaux :

- ✓ Le plastique : Peu coûteux et durable mais moins rigide.
- ✓ Le bois : Robuste mais très cher et difficile à trouver.
- ✓ La fibre de carbone : Plus flexible et très disponible mais coûteuse

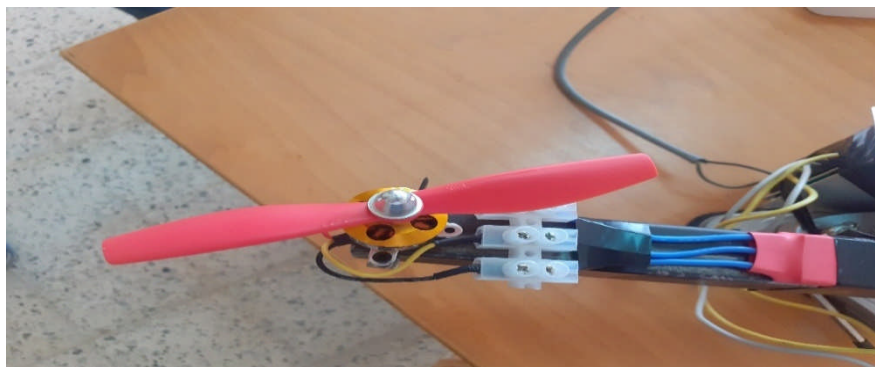


Figure IV.3 : Hélices 5045 de quadri rotor

- ✓ Hélices à sens de rotation horaire (Sérigraphies «5045R» sur l'hélice) sur le moteur avant gauche et sur le moteur arrière droit (pas de vis inverse). 2.
- ✓ Hélices à sens de rotation antihoraire (Sérigraphies «5045» sur l'hélice) sur le moteur avant droit et sur le moteur arrière gauche (pas de vis standard).

IV.2.2 Composants électriques :

a. Les moteurs brushless :

- Les moteurs brushless sont des moteurs a courant continue qui n'utilisent pas de balais collecteurs car les bobines sont alimentées directement et que les aimants permanent sont placés sur le rotor [43, 44].
- Les moteurs brushless sont caractérisé par leur ratio KV, il représente le nombre de tour par minute que réalise le moteur pour un volt 1V et est proportionnel au voltage d'entrée. [45]
- Plus le ratio KV d'un moteur est grand plus ce dernier sera rapide mais moins il aura de couple moteur qu'un moteur a ratio KV plus petit **tableau IV.1.**

	Vol nerveux	Vol polyvalent	Vol stable
Hélice (longueur en pouce)	8	8-10	8-11
Pas géométrique (en pouce)	5	4.5-5	4-4.5
RPM ou tr/min max	12000	10000	8000
KV	1100	900-1100	700-900

Tableau IV.1. Le choix des moteurs et les hélices adéquates.

En résumé, pour choisir un moteur, il faut prendre en compte à la fois le nombre de KV, la puissance et le poids. Le tableau suivant illustre quelques paramètres sur le choix du matériel selon le cahier de charge spécifié.

Spécification	Vol en immersion (de portée)	Vol nerveux	Vol stable
Matériel	Avoir plus d'autonomie	Avoir plus de vitesse	Avoir plus de couple
Moteurs	Faible KV	- KV important (1000 - 1500 KV) - Grande puissance des moteurs	Faible KV (<1000 KV)
Batterie	Batterie plus importante (plus de mAh)	Le "C" des Lipos plus élevé	
Hélice	Grande taille	- Petite taille (8-9") - grand pas (>4.5)	- Grande taille (>9") - petit pas (4-4.5")

Tableau IV.2 : choix des moteurs d'un quadrirotor

b. ESC :



Figure IV.4 : contrôleur électronique de vitesse

« Electronic Speed Controller » ou « *Contrôleur électronique de vitesse* en français»: C'est un dispositif qui permet de commander la vitesse des moteurs brushless en transformant le courant continu de la batterie en impulsion précises pour faire fonctionner les moteurs brushless.

Pour changer le sens de rotation, il suffit de permuter deux quelconques des trois connecteurs entre l'ESC et le moteur sans balais. Pratiquement, on ne devra pas aussi oublier de configurer les ESC avant leur utilisation, cette opération s'appelle "Calibration des ESC".

Elle permet de faire travailler les 4 moteurs dans une plage commune à bas régime comme à haut régime [46].

Le choix des contrôleurs électroniques de vitesse se rapporte sur sa caractéristique qui est l'intensité qu'il peut fournir (10A, 12A, 20A, 30A etc.), une valeur choisie en fonction du moteur avec une marge pour que l'ESC ne risque pas de bruler. [47]

c. Batterie : Alimentation à découpage :

Une alimentation à découpage est une alimentation électrique dont la régulation est assurée par des composants électroniques de puissance utilisés en commutation (généralement des transistors). Ce mode de fonctionnement diffère de celui des alimentations linéaires dans lesquelles les composants électroniques sont utilisés en mode linéaire. Une alimentation à découpage de type forward est une alimentation qui transmet instantanément la puissance, alors que celle de type flyback stocke cette énergie sous forme d'énergie magnétique dans une inductance (bobine) et libère ensuite cette énergie dans un circuit dit secondaire.

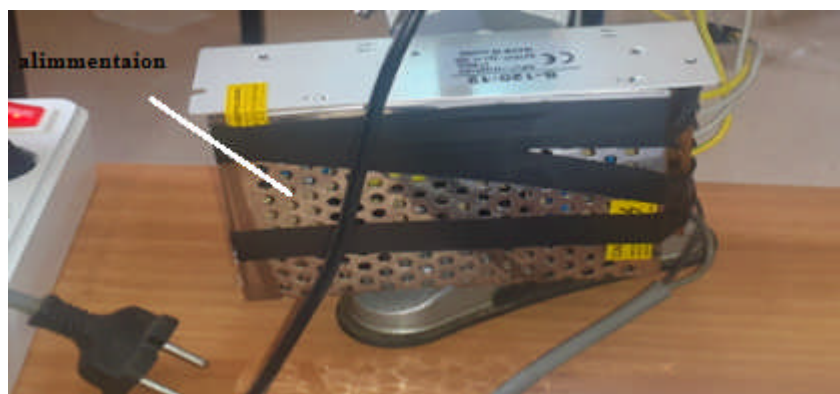


Figure IV.5 : Alimentation découpage

d. Le capteur MPU6050 :

Le capteur que nous allons utiliser est le capteur MPU6050 [48, 49]. D'abord, qu'est-ce qu'un IMU ? C'est un appareil capable de mesurer la force (accélération) et la vitesse. Généralement, il se compose d'un accéléromètre et d'un gyroscope. Par conséquent : une IMU ne mesure pas les angles.

Du moins pas directement, cela nécessite quelques calculs afin d'obtenir les angles. Le MPU-6050 est une IMU de 6DOF (lit "6 degrés de liberté"). Cela signifie qu'il dispose d'un accéléromètre et d'un gyroscope, tous deux sur 3 axes ($3 + 3 = 6DOF$). Il y a 9DOF IMU,

dans ce cas ils portent également un magnétomètre. D'autres peuvent avoir 5DOF, auquel cas le gyroscope ne mesure que deux axes.

Le MPU-6050 fonctionne en 3,3 volts, bien que certaines versions disposent d'un régulateur qui permet de le brancher en 5V.

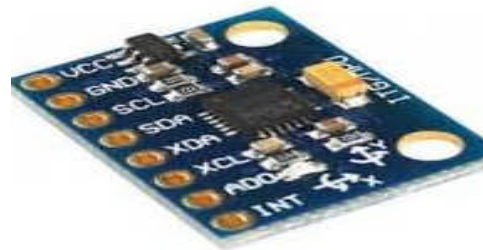


Figure IV.6 : Capteur MPU6050

Les caractéristiques techniques de ce module sont données dans le tableau suivant

Alimentation	2,375V - 3,46V
Consommation	3,9 mA
Interface	I2C
Accéléromètre	
Plages de mesures (g)	Sensibilité (LSB/g)
± 2	16384
± 4	8192
± 8	4096
± 16	2048
Accelerometer	
Plages de mesures (°/s)	Sensibilité (LSB/°/s)
± 250	131
± 500	65.5
± 1000	38.8
± 2000	16.4

Tableau IV.3 : caractéristique de module MPU 6050.

e. La radiocommande : [ANNEXE B]

La première fois qu'on manipule une vraie radiocommande, ça impressionne un peu. Beaucoup de boutons, un menu assez complexe, pas simple de s'y retrouver au début

✚ Exemple de la FlySky FS-i6 :

La FlySky FS-i6, livrée avec son récepteur. Cette radiocommande dispose de 6 canaux, de 4 switches, d'un écran LCD et de boutons permettant de naviguer dans les menus de configuration.



Figure IV.7 : Fly Sky FS-i6 et son récepteur

Le manuel utilisateur est fourni sur CD. Après avoir fouillé dans les menus, expérimenté et cherché sur le net, voici un rapide résumé :

- Les 4 switches permettent de passer d'une configuration préenregistrée à une autre rapidement. Par exemple, passer d'un mode standard à un mode sport, ou encore mode vol et mode atterrissage.
- Lorsqu'on modifie la configuration, il faut maintenir enfoncé le bouton "cancel" pour sauvegarder.
- On peut visualiser la commande de chaque canal sous forme d'histogrammes dans un menu dédié, ce qui est assez pratique pour identifier l'association stick/canal.
- On peut également configurer le **fail-safe**. Pour faire court, il s'agit des consignes par défaut que le récepteur doit appliquer en cas de perte de signal.

- On peut régler par exemple le **fail-safe** sur "**throttle** = 0", ce qui coupe immédiatement les moteurs. Le drone se crash immédiatement mais au moins ne s'en va pas dans la nature, au risque de s'écraser sur une voiture ou un piéton.

IV.2.2 Câblage des ESC :

Commençons par relier nos moteurs à nos ESC. Pour rappel, les moteurs brushless se comportent comme des moteurs triphasés, aussi le sens de branchement du trio de fils n'affectera que le sens de rotation. Ensuite, relierons nos ESC à l'Arduino comme suit :

- ESC 1: pin #4 de l'Arduino
- ESC 2: pin #5 de l'Arduino

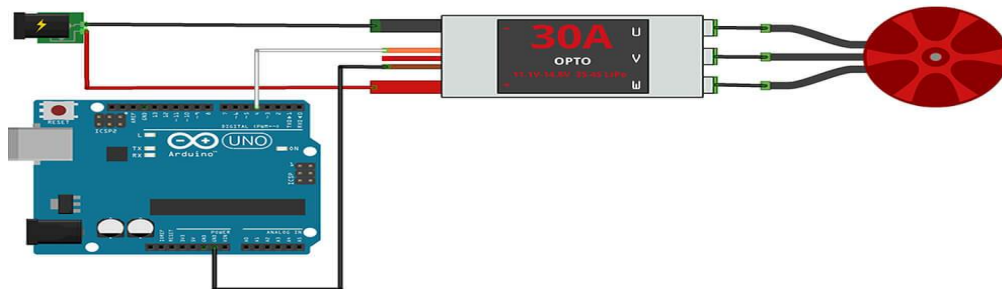


Figure IV.10 : câblage des ESC

- Attention à ne **PAS** alimenter l'Arduino avec la sortie BEC d'un ESC. L'Arduino sera alimenté en USB.

a-Calibrage:

Le calibrage est une étape indispensable sans laquelle on ne pourra pas faire tourner nos moteurs.

Bien évidemment ce calibrage se fait avec un arduino. Il s'agit d'envoyer aux ESC les extrêmes des commandes de gaz soit, en clair, le signal correspondant à la commande "arrêt moteur" et le signal correspondant à la commande "plein gaz".

Ainsi, les ESC connaissent leur plage de fonctionnement. Nous avons vu que les ESC se comportent comme des servomoteurs.

Un sketch Arduino qui nous permettra de calibrer nos ESC facilement est donné dans l'annexe 2. Une fois le programme téléversé, nous devrions voir nos moteurs commencer à tourner de plus en plus vite jusqu'à atteindre leur vitesse de rotation maximale.

IV.2.3 Câblage de capteur MPU 6050:

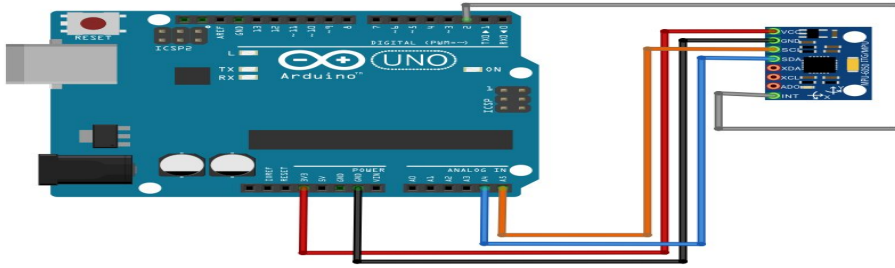


Figure IV.11 : câblage de capteur avec arduino

Placez le module MPU6050 aussi centré que possible sur la balance. Nous avons utilisés une planche à pain pour faire toutes les connexions, nous pouvons commencer par son calibrage voir [ANNEXE A].

IV.3 commande du banc d'essai :

Une fois qu'on a terminé le calibrage on passe au câblage de tout le système , Maintenant, nous devrions avoir le schéma ci-dessus :

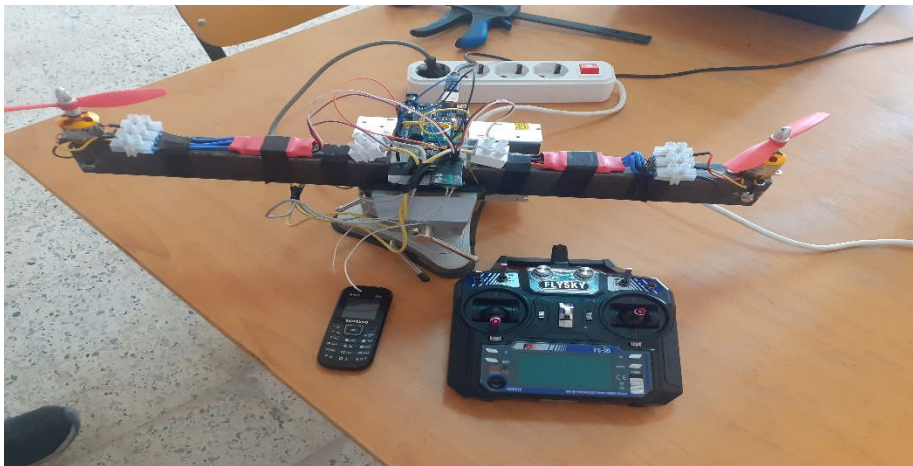


Figure IV.12 : réalisation pratique notre banc d'essai

Schéma de circuit : voici le câblage du système sur ARDUINO

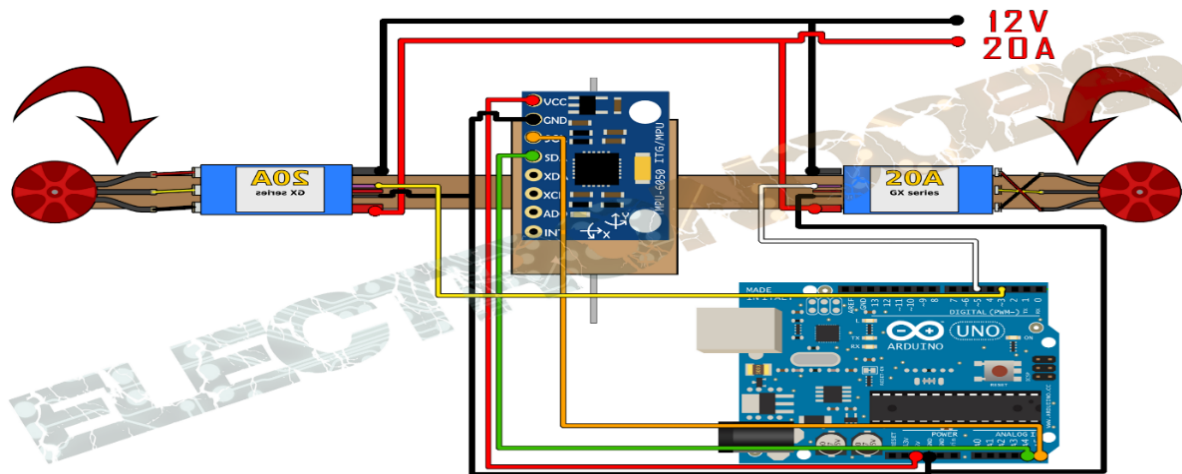


Figure IV.9 : câblage des systèmes avec arduino

IV.3.1 Implémentation des paramètres du PID

Le programme commence par la déclaration des bibliothèques des deux moteurs gauche et droit :

```
#include <Wire.h>
#include <Servo.h>
Servo right_prop;
Servo left_prop;
```

Ensuite la carte arduino doit recevoir des données à partir du module capteur IMU6050 (unité de mouvement inertiel) :

```
int16_t Acc_rawX, Acc_rawY, Acc_rawZ, Gyr_rawX, Gyr_rawY, Gyr_rawZ;
float Acceleration_angle[2];
float Gyro_angle[2];
float Total_angle[2];
```

Ensuite, En utilisant ces données, nous calculons l'angle d'inclinaison réel de l'axe :

```
Acceleration_angle[0] = atan((Acc_rawY/16384.0)/sqrt(pow((Acc_rawX/16384.0),2) +
pow((Acc_rawZ/16384.0),2)))*rad_to_deg;
Acceleration_angle[1] = atan(-1*(Acc_rawX/16384.0)/sqrt(pow((Acc_rawY/16384.0),2) +
pow((Acc_rawZ/16384.0),2)))*rad_to_deg;
```

Nous comparons cet angle avec celui souhaité qui est de 0° car nous voulons que le drone soit horizontal.

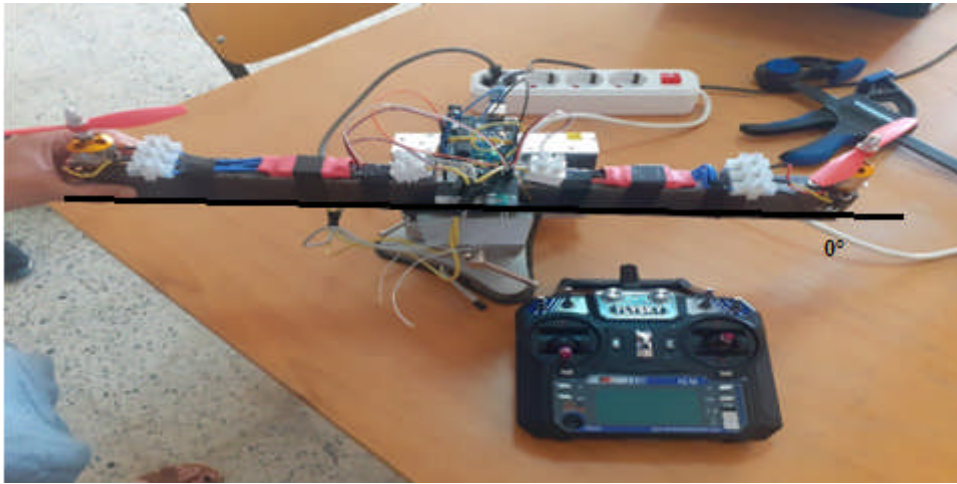


Figure IV.13 : la stabilisation souhaiter.

Nous appliquons quelque filtre, nous obtenons l'angle d'inclinaison réel de la barre, et nous voulons qu'elle reste toujours à 0 degré ce qui est parfaitement horizontal, l'angle maximum vers la droite dans ce cas environs 45 degré.

```
Total_angle[0] = 0.98 *(Total_angle[0] + Gyro_angle[0]*elapsedTime) +  
0.02*Acceleration_angle[0];  
Total_angle[1] = 0.98 *(Total_angle[1] + Gyro_angle[1]*elapsedTime) +  
0.02*Acceleration_angle[1];
```

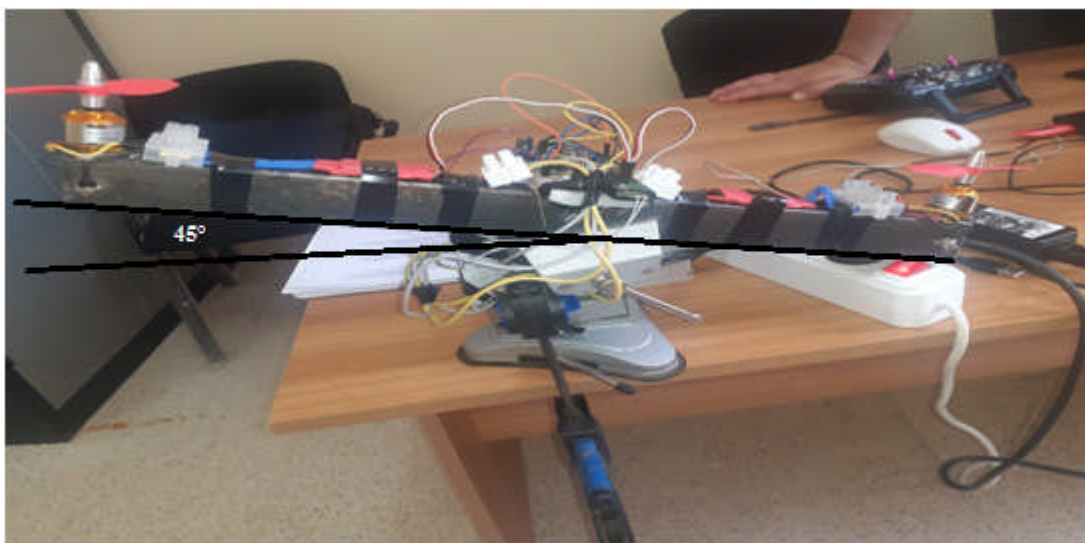


Figure IV.14 : angle d'inclinations

IV.3.2 Procédure de configuration des paramètres PID :

1. Commençons par initialiser tous les coefficients du PID à 0 comme ceci :

- $K_p = 0$
- $K_i = 0$
- $K_d = 0$

Augmentons K_p jusqu'à ce que le banc d'essai commence à osciller.

- $K_p = 3$
- $K_i = 0$
- $K_d = 0$

2. Ensuite, augmentons la valeur de K_d (roll & pitch) jusqu'à ce que le drone commence tout juste à ce stabilisé.

3. Augmentons le gain K_i par pas de 0.001. Le banc d'essai est censé devenir de plus en plus stable.

Nous pouvons continuer à ajuster les coefficients jusqu'à trouver la config qui nous convient le mieux.

L'étape suivante de contrôle PID consiste à utiliser les constantes K_p , K_i , K_d . Et faire des tests jusqu'à l'obtention de la stabilité. Après plusieurs tests ont pris les paramètres

Double $k_p=5$;

Double $k_i=0.006$;

Double $k_d=2$;

N.B : pour les détails du programmer voir [ANNEXE C]

IV.4 Conclusion :

Dans ce chapitre nous avons détaillé les différentes étapes de la construction mécanique d'un banc d'essai et nous avons vu les éléments qui le constitue. Ensuite, nous avons calibré les ESC et le capteur. A la fin nous sommes passé à la configuration des paramètres du PID ont ce basant sur une méthode purement expérimentale.

Conclusion générale

Conclusion Générale

Au terme de ce projet, nous pouvons dire que nous avons touché à un nouveau domaine de l'électromécanique, c'est le domaine des drones.

Nous avons étudié les drones en général et les quadri-rotors en particulier qui sont des systèmes complexes multi variables non linéaires, instables en mode de vol et représente une dynamique fortement couplée.

Puis nous avons réalisé une simulation sous MATLAB pour pouvoir tester le régulateur synthétisé, ces tests ont été visualisés et observé, nous avons donc conclu que la régulation PID donne des bons résultats en termes de rapidité et dépassement.

Enfin nous avons passé à la partie expérimentale ou nous avons pu réaliser un banc d'essai pour faire l'implémentation de notre PID dans l'objectif de faire le poids entre la théorie a la pratique.

Comme perspective :

Dans le but d'optimiser ce travail, nous suggérons d'introduire les taches suivantes :

- Prendre en considération des non linéarités dans la plate-forme Simulink pour avoir des résultats plus proches de la réalité.
- Ajouter l'interface 3D à la plateforme sous Simulink pour pouvoir visualiser et étudier la trajectoire du quadri-rotor.
- Implémenter d'autres capteurs pour améliorer les performances de notre banc d'essai.

La stabilisation de notre prototype n'est pas encore achevée. Pour ces différentes raisons :

Dans un premier temps, compte-tenu de la nature du projet, il nous a été difficile de déterminer l'ampleur de la tâche. La réalisation d'un drone quadrirotor offre de nombreuses possibilités en matière d'applications.

Conclusion Générale

La plus grande difficulté rencontrée lors de ce projet fut la stabilisation du drone et la stabilisation du quadri rotor. Nous nous ne sommes pas penchés beaucoup sur le modèle théorique réalisé. Car, d'une manière générale, nous avons mal estimé la durée à attribuer à certaines tâches, ce qui au final ne nous a pas permis de finir tout ce que nous souhaitions entreprendre.

Le quadri rotor n'est pas encore contrôlable sur cet axe. La réalisation électronique est aboutie, mais tous les éléments n'ont pas été intégrés sur le centre final, en raison de nos difficultés à stabiliser le ban d'essai.

Enfin nous proposons aux étudiantes qui vont travailler sur la suite de ce mémoire de se pencher plus sur la partie :

- Modalisation : minimiser les hypothèses simplificatrices.
- Commande : trouver une bonne méthode pour optimiser les paramètres PID ou chercher une autre stratégie de commande.

ANNEXE A

Programmation de capteur MPU 6050

Annexe A : Programmation de capteur MPU dans arduino

Introduction :

Dans cette annexe nous allons voir la programmation de capteur MPU 6050 [50]

Implémentation du Code

```
                // I2Cdev and MPU6050 must be installed as libraries
#include "I2Cdev.h"
#include "MPU6050.h"
#include "Wire.h"
                /////////////////////////////////////////////////// CONFIGURATION ///////////////////////////////////
//Change this 3 variables if you want to fine tune the sketch to your needs.
int buffersize=1000; //Amount of readings used to average, make it higher to get more
precision but sketch will be slower (default:1000)
int acel_deadzone=8; //Acelerometer error allowed, make it lower to get more precision,
but sketch may not converge (default:8)
int giro_deadzone=1; //Giro error allowed, make it lower to get more precision, but sketch
may not converge (default:1)
// default I2C address is 0x68
// specific I2C addresses may be passed as a parameter here
// AD0 low = 0x68 (default for InvenSense evaluation board)
// AD0 high = 0x69
//MPU6050 accelgyro;
MPU6050 accelgyro(0x68); // <-- use for AD0 high
int16_t ax, ay, az,gx, gy, gz;
int mean_ax,mean_ay,mean_az,mean_gx,mean_gy,mean_gz,state=0;
int ax_offset,ay_offset,az_offset,gx_offset,gy_offset,gz_offset;
                /////////////////////////////////////////////////// SETUP ///////////////////////////////////
void setup() {
  // join I2C bus (I2Cdev library doesn't do this automatically)
  Wire.begin();
  // COMMENT NEXT LINE IF YOU ARE USING ARDUINO DUE
  TWBR = 24; // 400kHz I2C clock (200kHz if CPU is 8MHz). Leonardo measured 250kHz.
  // initialize serial communication
  Serial.begin(115200);
  // initialize device
  accelgyro.initialize();
  // wait for ready
  while (Serial.available() && Serial.read()); // empty buffer
  while (!Serial.available()){
    Serial.println(F("Send any character to start sketch.\n"));
    delay(1500);
  }
  while (Serial.available() && Serial.read()); // empty buffer again
  // start message
  Serial.println("\nMPU6050 Calibration Sketch");
  delay(2000);
}
```

Annexe A : Programmation de capteur MPU dans arduino

```
Serial.println("\nYour MPU6050 should be placed in horizontal position, with package
letters facing up. \nDon't touch it until you see a finish message.\n");
delay(3000);
// verify connection
Serial.println(accelgyro.testConnection() ? "MPU6050 connection successful" : "MPU6050
connection failed");
delay(1000);
// reset offsets
accelgyro.setXAccelOffset(0);
accelgyro.setYAccelOffset(0);
accelgyro.setZAccelOffset(0);
accelgyro.setXGyroOffset(0);
accelgyro.setYGyroOffset(0);
accelgyro.setZGyroOffset(0);
}
////////////////////////////////////////////////// LOOP ////////////////////////////////////////////
void loop() {
if (state==0){
Serial.println("\nReading sensors for first time...");
meansensors();
state++;
delay(1000);
}
if (state==1) {
Serial.println("\nCalculating offsets...");
calibration();
state++;
delay(1000);
}
if (state==2) {
meansensors();
Serial.println("\nFINISHED!");
Serial.print("\nSensor readings with offsets:\t");
Serial.print(mean_ax);
Serial.print("\t");
Serial.print(mean_ay);
Serial.print("\t");
Serial.print(mean_az);
Serial.print("\t");
Serial.print(mean_gx);
Serial.print("\t");
Serial.print(mean_gy);
Serial.print("\t");
Serial.println(mean_gz);
Serial.print("Your offsets:\t");
Serial.print(ax_offset);
```

Annexe A : Programmation de capteur MPU dans arduino

```
Serial.print("\t");
Serial.print(ay_offset);
Serial.print("\t");
Serial.print(az_offset);
Serial.print("\t");
Serial.print(gx_offset);
Serial.print("\t");
Serial.print(gy_offset);
Serial.print("\t");
Serial.println(gz_offset);
Serial.println("\nData is printed as: accelX accelY accelZ giroX giroY giroZ");
Serial.println("Check that your sensor readings are close to 0 0 16384 0 0 0");
Serial.println("If calibration was succesful write down your offsets so you can set them in
your projects using something similar to mpu.setXAccelOffset(youroffset)");
while (1);
}
}

////////////////////////////////////// FUNCTIONS ////////////////////////////////////////
void meansensors(){
long i=0,buff_ax=0,buff_ay=0,buff_az=0,buff_gx=0,buff_gy=0,buff_gz=0;
while (i<(buffersize+101)){
// read raw accel/gyro measurements from device
accelgyro.getMotion6(&ax, &ay, &az, &gx, &gy, &gz);
if (i>100 && i<=(buffersize+100)){ //First 100 measures are discarded
buff_ax=buff_ax+ax;
buff_ay=buff_ay+ay;
buff_az=buff_az+az;
buff_gx=buff_gx+gx;
buff_gy=buff_gy+gy;
buff_gz=buff_gz+gz;
}
if (i==(buffersize+100)){
mean_ax=buff_ax/buffersize;
mean_ay=buff_ay/buffersize;
mean_az=buff_az/buffersize;
mean_gx=buff_gx/buffersize;
mean_gy=buff_gy/buffersize;
mean_gz=buff_gz/buffersize;
}
i++;
delay(2); //Needed so we don't get repeated measures
}
}
}

void calibration(){
ax_offset=-mean_ax/8;
ay_offset=-mean_ay/8;
```

Annexe A : Programmation de capteur MPU dans arduino

```
az_offset=(16384-mean_az)/8;
gx_offset=-mean_gx/4;
gy_offset=-mean_gy/4;
gz_offset=-mean_gz/4;
while (1){
  int ready=0;
  accelgyro.setXAccelOffset(ax_offset);
  accelgyro.setYAccelOffset(ay_offset);
  accelgyro.setZAccelOffset(az_offset);
  accelgyro.setXGyroOffset(gx_offset);
  accelgyro.setYGyroOffset(gy_offset);
  accelgyro.setZGyroOffset(gz_offset);
  meansensors();
  Serial.println("...");
  if (abs(mean_ax)<=acel_deadzone) ready++;
  else ax_offset=ax_offset-mean_ax/acel_deadzone;
  if (abs(mean_ay)<=acel_deadzone) ready++;
  else ay_offset=ay_offset-mean_ay/acel_deadzone;
  if (abs(16384-mean_az)<=acel_deadzone) ready++;
  else az_offset=az_offset+(16384-mean_az)/acel_deadzone;
  if (abs(mean_gx)<=giro_deadzone) ready++;
  else gx_offset=gx_offset-mean_gx/(giro_deadzone+1);
  if (abs(mean_gy)<=giro_deadzone) ready++;
  else gy_offset=gy_offset-mean_gy/(giro_deadzone+1);
  if (abs(mean_gz)<=giro_deadzone) ready++;
  else gz_offset=gz_offset-mean_gz/(giro_deadzone+1);

  if (ready==6) break;
}
}
```

ANNEXE B

Programmation de la radiocommande

Introduction :

Dans cette annexe nous allons bien décrire la radio commande et son programme.

Appairer émetteur et récepteur (optionnel)

Pour que l'émetteur et le récepteur communiquent, ils doivent d'abord être appairés. Lorsque nous achetons une radiocommande livrée avec son récepteur, ils le sont déjà (normalement). Si par contre Nous avons acheté la RC et le récepteur séparément, il est nécessaire de les appairer. Voici comment procéder d'après la doc de la FS-i6: [50]

1. Connectons le jumper d'appariement sur le port batterie du récepteur
2. Alimentons le récepteur via la broche d'alim de n'importe quel canal. La LED rouge clignote, indiquant que le récepteur est en mode d'appariement.
3. Maintenons appuyé le bouton d'appariement de la radiocommande et allumons-la
4. Le processus d'appariement est terminé lorsque la LED du récepteur clignote plus lentement. Retirons le jumper, la LED doit se figer.

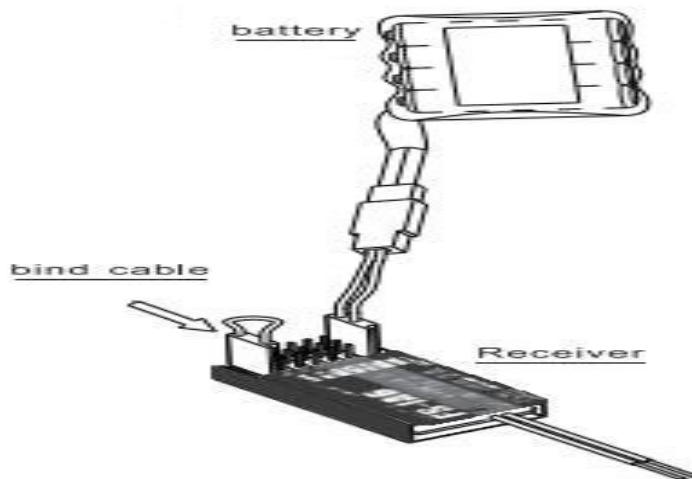


Figure B.1 : Appairer émetteur et récepteur

Notre récepteur est maintenant apparié avec Notre radiocommande.

Programme :

```
#define CHANNEL1 0
#define CHANNEL2 1
#define CHANNEL3 2
#define CHANNEL4 3
#define YAW 0
#define PITCH 1
#define ROLL 2
#define THROTTLE 3
```

```
volatile unsigned long current_time;
volatile unsigned long timer[6];
```

```
// Previous state of each channel (HIGH or LOW)
```

```

volatile byte previous_state[6];

// Duration of the pulse on each channel in µs (must be within 1000µs & 2000µs)
volatile unsigned int pulse_duration[6] = {1500, 1500, 1000, 1500};

int mode_mapping[6];

/**
 * Setup routine.
 *
 * @return void
 */
void setup()
{
    Serial.begin(9600);

    // Customize mapping of controls: set here which command is on wich channel.
    mode_mapping[YAW]    = CHANNEL4;
    mode_mapping[PITCH]  = CHANNEL2;
    mode_mapping[ROLL]   = CHANNEL1;
    mode_mapping[THROTTLE] = CHANNEL3;

    PCICR |= (1 << PCIE0); //Set PCIE0 to enable PCMSK0 scan.
    PCMSK0 |= (1 << PCINT0); //Set PCINT0 (digital input 8) to trigger an interrupt on state
change.
    PCMSK0 |= (1 << PCINT1); //Set PCINT1 (digital input 9)to trigger an interrupt on state
change.
    PCMSK0 |= (1 << PCINT2); //Set PCINT2 (digital input 10)to trigger an interrupt on state
change.
    PCMSK0 |= (1 << PCINT3); //Set PCINT3 (digital input 11)to trigger an interrupt on state
change.
}
/**
 * Main loop routine.
 *
 * @return void
 */
void loop()
{
    // Simply dump received data.
    dumpChannels();
}

/**
 * Print received instruction for each channel in the serial port for debug.
 *
 * @return void
 */
void dumpChannels()
{

```

```

for (int i = CHANNEL1; i <= CHANNEL4; i++) {
  Serial.print("Channel ");
  Serial.print(i+1);
  Serial.print(" ");
  Serial.print(pulse_duration[i]);
  Serial.print("\t");
}
Serial.print("\n");
}
/**
 * This Interrupt Sub Routine is called each time input 8, 9, 10 or 11 changed state.
 * Read the receiver signals in order to get flight instructions.
 *
 * This routine must be as fast as possible to prevent main program to be messed up.
 * The trick here is to use port registers to read pin state.
 * Doing (PINB & B00000001) is the same as digitalRead(8) with the advantage of using less
CPU loops.
 * It is less convenient but more efficient, which is the most important here.
 *
 * @see https://www.arduino.cc/en/Reference/PortManipulation
 */
ISR(PCINT0_vect)
{
  current_time = micros();

  // Channel 1 -----
  if (PINB & B00000001) { // Is input 8 high ?
    if (previous_state[CHANNEL1] == LOW) { // Input 8 changed from 0 to 1
(rising edge)
      previous_state[CHANNEL1] = HIGH; // Save current state
      timer[CHANNEL1] = current_time; // Start timer
    }
    } else if (previous_state[CHANNEL1] == HIGH) { // Input 8 changed from 1 to
0 (falling edge)
      previous_state[CHANNEL1] = LOW; // Save current state
      pulse_duration[CHANNEL1] = current_time - timer[CHANNEL1]; // Stop timer &
calculate pulse duration
    }
  // Channel 2 -----
  if (PINB & B00000010) { // Is input 9 high ?
    if (previous_state[CHANNEL2] == LOW) { // Input 9 changed from 0 to 1
(rising edge)
      previous_state[CHANNEL2] = HIGH; // Save current state
      timer[CHANNEL2] = current_time; // Start timer
    }
    } else if (previous_state[CHANNEL2] == HIGH) { // Input 9 changed from 1 to
0 (falling edge)
      previous_state[CHANNEL2] = LOW; // Save current state
      pulse_duration[CHANNEL2] = current_time - timer[CHANNEL2]; // Stop timer &
calculate pulse duration
    }
}

```

```

}
// Channel 3 -----
if (PINB & B00000100) { // Is input 10 high ?
  if (previous_state[CHANNEL3] == LOW) { // Input 10 changed from 0 to 1
(rising edge)
    previous_state[CHANNEL3] = HIGH; // Save current state
    timer[CHANNEL3] = current_time; // Start timer
  }
} else if (previous_state[CHANNEL3] == HIGH) { // Input 10 changed from 1 to
0 (falling edge)
  previous_state[CHANNEL3] = LOW; // Save current state
  pulse_duration[CHANNEL3] = current_time - timer[CHANNEL3]; // Stop timer &
calculate pulse duration
}

// Channel 4 -----
if (PINB & B00001000) { // Is input 11 high ?
  if (previous_state[CHANNEL4] == LOW) { // Input 11 changed from 0 to 1
(rising edge)
    previous_state[CHANNEL4] = HIGH; // Save current state
    timer[CHANNEL4] = current_time; // Start timer
  }
} else if (previous_state[CHANNEL4] == HIGH) { // Input 11 changed from 1 to
0 (falling edge)
  previous_state[CHANNEL4] = LOW; // Save current state
  pulse_duration[CHANNEL4] = current_time - timer[CHANNEL4]; // Stop timer &
calculate pulse duration
}
}

```

Comme chacun a ses préférences en ce qui concerne le mode de sa radiocommande, on a prévu dans le code un mapping qui nous permet d'associer chaque commande au canal de notre choix. À nous de modifier en conséquence cette portion du `setup()`:

```

mode_mapping[YAW] = CHANNEL4;
mode_mapping[PITCH] = CHANNEL2;
mode_mapping[ROLL] = CHANNEL1;
mode_mapping[THROTTLE] = CHANNEL3;

```

- Le mapping par défaut correspond au mode 2.

Câblage et test:

Il est maintenant temps de tester notre code. Commençons par relier la broche "signal" de chaque canal à notre Arduino en respectant le mapping suivant:

- Canal 1 → Broche 8
- Canal 2 → Broche 9
- Canal 3 → Broche 10
- Canal 4 → Broche 11

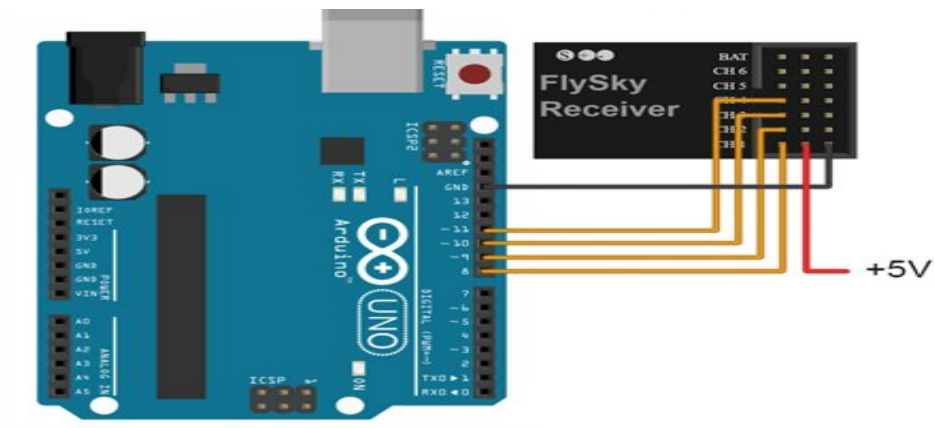


Figure B.2 : Câblage de l'arduino avec le récepteur.

- N'oublions pas la masse commune.

ANNEXE C

Programme finale

Introduction :

Dans cette annexe nous allons voir le programme complet sur arduino Nous pouvons continuer à ajuster les coefficients jusqu'à trouver la config qui nous convient le mieux. [52]

```
#include <Wire.h>

#include <Servo.h>

Servo right_prop;

Servo left_prop;

/*MPU-6050 gives you 16 bits data so you have to create some 16int constants
 * to store the data for accelerations and gyro*/

int16_t Acc_rawX, Acc_rawY, Acc_rawZ,Gyr_rawX, Gyr_rawY, Gyr_rawZ;

float Acceleration_angle[2];

float Gyro_angle[2];

float Total_angle[2];

float elapsedTime, time, timePrev;

int i;

float rad_to_deg = 180/3.141592654;

float PID, pwmLeft, pwmRight, error, previous_error;

float pid_p=0;

float pid_i=0;

float pid_d=0;

//////////PID CONSTANTS//////////

double kp=3;//3.55

double ki=0;//0.003

double kd=2;//2.05

//////////

double throttle=1000; //initial value of throttle to the motors
```

```
float desired_angle = 0; //This is the angle in which we want the
```

```
    //balance to stay steady
```

```
void setup() {
```

```
    Wire.begin(); //begin the wire communication
```

```
    Wire.beginTransmission(0x68);
```

```
    Wire.write(0x6B);
```

```
    Wire.write(0);
```

```
    Wire.endTransmission(true);
```

```
    Serial.begin(250000);
```

```
    right_prop.attach(3); //attach the right motor to pin 3
```

```
    left_prop.attach(5); //attach the left motor to pin 5
```

```
    time = millis(); //Start counting time in milliseconds
```

```
    /*In order to start up the ESCs we have to send a min value
```

```
    * of PWM to them before connecting the battery. Otherwise
```

```
    * the ESCs won't start up or enter in the configure mode.
```

```
    * The min value is 1000us and max is 2000us, REMEMBER!*/
```

```
    left_prop.writeMicroseconds(1000);
```

```
    right_prop.writeMicroseconds(1000);
```

```
    delay(7000); /*Give some delay, 7s, to have time to connect
```

```
        *the propellers and let everything start up*/
```

```
}//end of setup void
```

```
void loop() {
```

```
//////////////////////////////////I M U//////////////////////////////////
```

```
timePrev = time; // the previous time is stored before the actual time read

time = millis(); // actual time read

elapsedTime = (time - timePrev) / 1000;

/*The tiemStep is the time that elapsed since the previous loop.

* This is the value that we will use in the formulas as "elapsedTime"

* in seconds. We work in ms so we haveto divide the value by 1000

to obtain seconds*/

/*Reed the values that the accelerometre gives.

* We know that the slave adress for this IMU is 0x68 in

* hexadecimal. For that in the RequestFrom and the

* begin functions we have to put this value.*/

Wire.beginTransmission(0x68);

Wire.write(0x3B); //Ask for the 0x3B register- correspond to AcX

Wire.endTransmission(false);

Wire.requestFrom(0x68,6,true);

/*We have asked for the 0x3B register. The IMU will send a brust of register.

* The amount of register to read is specify in the requestFrom function.

* In this case we request 6 registers. Each value of acceleration is made out of

* two 8bits registers, low values and high values. For that we request the 6 of them

* and just make then sum of each pair. For that we shift to the left the high values

* register (<<) and make an or (|) operation to add the low values.*/

Acc_rawX=Wire.read()<<8|Wire.read(); //each value needs two registres

Acc_rawY=Wire.read()<<8|Wire.read();
```

```
Acc_rawZ=Wire.read()<<8|Wire.read();

//////This is the part where you need to calculate the angles using Euler equations///*/

/* - Now, to obtain the values of acceleration in "g" units we first have to divide the raw
 * values that we have just read by 16384.0 because that is the value that the MPU6050
 * datasheet gives us.*/

/* - Next we have to calculate the radian to degree value by dividing 180° by the PI number
 * which is 3.141592654 and store this value in the rad_to_deg variable. In order to not have
 * to calculate this value in each loop we have done that just once before the setup void.
 */

/* Now we can apply the Euler formula. The atan will calculate the arctangent. The
 * pow(a,b) will elevate the a value to the b power. And finally sqrt function
 * will calculate the root square.*/

/*---X---*/

Acceleration_angle[0] = atan((Acc_rawY/16384.0)/sqrt(pow((Acc_rawX/16384.0),2) +
pow((Acc_rawZ/16384.0),2)))*rad_to_deg;

/*---Y---*/

Acceleration_angle[1] = atan(-1*(Acc_rawX/16384.0)/sqrt(pow((Acc_rawY/16384.0),2) +
pow((Acc_rawZ/16384.0),2)))*rad_to_deg;

/*Now we read the Gyro data in the same way as the Acc data. The address for the
 * gyro data starts at 0x43. We can see this addresses if we look at the register map
 * of the MPU6050. In this case we request just 4 values. We don't want the gyro for
 * the Z axis (YAW).*/

Wire.beginTransaction(0x68);
```

```
Wire.write(0x43); //Gyro data first adress

Wire.endTransmission(false);

Wire.requestFrom(0x68,4,true); //Just 4 registers

  Gyr_rawX=Wire.read()<<8|Wire.read(); //Once again we shif and sum

Gyr_rawY=Wire.read()<<8|Wire.read();

/*Now in order to obtain the gyro data in degrees/seconda we have to divide first
the raw value by 131 because that's the value that the datasheet gives us*/

/*---X---*/

Gyro_angle[0] = Gyr_rawX/131.0;

/*---Y---*/

Gyro_angle[1] = Gyr_rawY/131.0;

/*Now in order to obtain degrees we have to multiply the degree/seconds
*value by the elapsedTime.*/

/*Finnaly we can apply the final filter where we add the acceleration
*part that affects the angles and ofcourse multiply by 0.98 */

/*---X axis angle---*/

Total_angle[0]  =  0.98  *(Total_angle[0]  +  Gyro_angle[0]*elapsedTime)  +
0.02*Acceleration_angle[0];

/*---Y axis angle---*/

Total_angle[1]  =  0.98  *(Total_angle[1]  +  Gyro_angle[1]*elapsedTime)  +
0.02*Acceleration_angle[1];

/*Now we have our angles in degree and values from -10°0 to 100° approx*/

//Serial.println(Total_angle[1]);

/*////////////////////////////////////P I D////////////////////////////////////*/

/*Remember that for the balance we will use just one axis. I've choose the x angle
```

to implement the PID with. That means that the x axis of the IMU has to be paralel to the balance*/

/*First calculate the error between the desired angle and

the real measured angle/

error = Total_angle[1] - desired_angle;

/*Next the proportional value of the PID is just a proportional constant

multiplied by the error/

pid_p = kp*error;

/*The integral part should only act if we are close to the

desired position but we want to fine tune the error. That's

why I've made a if operation for an error between -2 and 2 degree.

To integrate we just sum the previous integral value with the

error multiplied by the integral constant. This will integrate (increase)

the value each loop till we reach the 0 point*/

if(-3 <error <3)

{

pid_i = pid_i+(ki*error);

}

/*The last part is the derivate. The derivate acts upon the speed of the error.

As we know the speed is the amount of error that produced in a certain amount of time divided by that time. For taht we will use a variable called previous_error.

We substract that value from the actual error and divide all by the elapsed time.

Finnaly we multiply the result by the derivate constant*/

```
pid_d = kd*((error - previous_error)/elapsedTime);

/*The final PID values is the sum of each of this 3 parts*/

PID = pid_p + pid_i + pid_d;

/*We know taht the min value of PWM signal is 1000us and the max is 2000. So that
tells us that the PID value can/s oscilate more than -1000 and 1000 because when we
have a value of 2000us the maximum value taht we could sybstract is 1000 and when
we have a value of 1000us for the PWM sihnal, the maximum value that we could add is 1000
to reach the maximum 2000us*/

if(PID < -1000)
{
    PID=-1000;
}

if(PID > 1000)
{
    PID=1000;
}

/*Finnaly we calculate the PWM width. We sum the desired throttle and the PID value*/

pwmLeft = throttle + PID;

pwmRight = throttle - PID;

/*Once again we map the PWM values to be sure that we won't pass the min
and max values. Yes, we've already maped the PID values. But for example, for
throttle value of 1300, if we sum the max PID value we would have 2300us and
that will mess up the ESC.*/

//Right

if(pwmRight < 1000)
```

```
{
  pwmRight= 1000;
}
if(pwmRight > 2000)
{
  pwmRight=2000;
}
//Left
if(pwmLeft < 1000)
{
  pwmLeft= 1000;
}
if(pwmLeft > 2000)
{
  pwmLeft=2000;
}
/*Finnaly using the servo function we create the PWM pulses with the calculated
width for each pulse*/
left_prop.writeMicroseconds(pwmLeft);
right_prop.writeMicroseconds(pwmRight);
previous_error = error; //Remember to store the previous error.
} //end of loop void
```

Bibliographie

Bibliographie

- [1] <https://www.futura-sciences.com/sciences/definitions/aeronautique-drone-6174>
- [2] https://www.onera.fr/sites/default/files/ressources_documentaires/cours-exposes_conf/mieux-connaître-les-drones.pdf
- [3] T. Hamel, R. Mahony, R. Lozano, J. Ostrowski “Dynamic modelling And configuration
- [4] Michel Asensio, les drones et les opérations en réseau, segmentation mission, fondation pour la recherche stratégique ,2008
- [5] _RCtoys, <http://www.rctoys.com/>.
- [6] S. Bouabdallah, DESIGN AND CONTROL OF QUADROTORS WITH APPLICATION TO AUTONOMOUS FLYING, Lausanne: ÉCOLE POLYTECHNIQUE FÉDÉRALE DE LASUSANNE, 23 FÉVRIER 2007.
- [7] Rodrigo Martinez-Val, Carlos Hernandez, Preliminary design of a low speed, long endurance remote piloted vehicles (RPV) for civil applications, Aircraft Design2(1999),167-182
- [8] G, Romeo, G, Frulla, and E, Cestino, Design of a high-altitude long-endurance solar-powered unmanned air vehicle formulti-payloadandoperationsProc.IMEchE Vol.221Part G: J. Aerospace Engineering 2007
- [9] Air and Space europe, Current and Future UAV Military Users and Applications,VOL.1, NO 5/6, 1999
- [10] Michel Asensio, Les drones tactiques à voilure tournante dans les engagements contemporains, Recherches & Documents° 08, 2010
- [11] Adnan MARTINI << modélisation et commande de vol d’un hélicoptère drone soumis à une rafale de vent >>, thèse de doctorat l’université Paul Verlaine –METS. 2008.
- [12] Direction Cosine Matrix IMU:Theory.William Premerlani and Paul Bizard.Mai2009flying”
- [13] KHLEDJ Abdennour, BOUGAR Oussama, « Etude et réalisation d’un drone de type quadrotor », Mémoire du Projet de Fin d’Etudes Pour l’obtention de diplôme Master En « Génie électrique », Option : « Automatique des systèmes de production », Khemis Miliana, 2016 -2017

Bibliographie

- [14] Ahmad hably, << approches bornées pour la commande drones>> docteur de LINP Grenoble. 2007.
- [15] ADRIEN DROUOT, thèse doctorat de l'université de lorraine en automatique 'stratégies de commande pour la navigation autonome d'un drone projectile miniature'.2013
- [16] I. Rivals << modélisation et commande de processus par réseaux de neurones : application au pilotage d'un véhicule autonome >>, thèse de doctorat –universalité de paris VI – 1995.
- [17] V. parenti –castelli, W. schiehleu << ramansy 18 robot design, dynamique and control>> - livre springer 2010.
- [18] A. De Luca, << dynamique mode 10 F robot >> cour pédagogique - universalité de Rome – 2015.
- [19] PS. Neetata nata, D. De groff, << neural network modeling: statistical mechanics and cybernatic perspectives >> CRC press – Boca katon Florida, 1994.
- [20] 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.
- [21] Yann AMEHO, thèse de doctorat de l'université de TOULOUSE en automatique 'Du pilotage d'une famille de drones à celui d'un drone hybride via la commande adaptative', 2013.
- [22] HAMDAD GHILES, IDER ABELLAH FATAH, KLOUL CHABANE, mémoire master de l'Université MOULOUD MAMMERIE en Génie électrique spécialité commande des systèmes, 'Réalisation et Synthèse de lois de Commande PID d'un Drone à Voilures Tournantes de Type Quadrirotor' ,2017.
- [23] Modélisation et Contrôle d'un mini Quadrirotor .Epreuve de Modélisation (session 2013) Ecole Polytechnique – ENS de cachan.
- [24] Alain Vuille, Travail de Bachelor, Filière Microtechnique-MI, 'Etude et expérimentation de petits quadricoptères destinés à des missions d'observation', 2014.
- [25] **A.Mokhtari,A.Benallegue.**—Dynamic feedback controller of Euler angles and windparameters estimation for a quadrotorunmannedaerialvehicle.∥IEEE International Conference on Robotics and Automation,ICRA'04 . 1245—1250.2004.
- [26] LAMRI Sofiane, MOHAMMEDI Achref, mémoire master a l'école supérieure en science appliqué spécialité é automatique, 'Modélisation d'un Système UAV de type Quadrotor hybride'

Bibliographie

- [27] SEDINI Chahrazed, CHERIGUI Nasre-Eddine a l'Université Abou Bekr Belkaïd de Tlemcen, spécialité automatique,' Conception et commande d'un quadrotor UAV à base d'Arduino', 2019.
- [28] AYAD Redouane, Mémoire magister, Université des Sciences et de la Technologie d'Oran MOHAMED BOUDIAF, spécialité automatique,' Conception et réalisation d'un drone quadrirotor', 2010.
- [29] KHEBBACHE Hicham, Tolérance aux défauts via la méthode backstepping des systèmes non linéaires, phd, Université FERHAT ABBAS de Setif, 2012
- [30] L. Laïb and D.E. Maamria, "Commande d'un Quadrirotor ", Mémoire d'ingénieur, École Nationale Polytechnique, Alger,2011.
- [31] A. Kadi, "Mécanique Rationnelle Cours et Exercices Résolus", (pages 222-225), Université M'hamedBougara, Boumerdés.
- [32] D. RachediNedjmi, "Commande hybride avec observation d'un UAV de type quadrotor",Thèse de Magistère, École Militaire Polytechnique, Alger,Algérie,2010.
- [33] A.CHRIETTE. DRMMC : Drones autonomes : Définition et Classification. Ecole Centrale de Nantes à Département Auto-Robot, 2012.
- [34] MOKHTARI MOHAMMED RIDA, thèse de doctorat de l'université d'ABOUBEKR BELKAID TLEMEN en automatique observation et commande.
- [35] Hebatalla Mohamed Nabil EIKholy, « Dynamic Modeling and Control of a Quadrotor Using Linear and Nonlinear Approaches », The American university in Cairo,2014
- [36] AHMED CHEMONI, Cour d'automatique des systèmes actionnés.
- [37] Cours d'automatique : Représentations d'état linéaires des systèmes mono-entrée, mono-sortie, Éditions Lily et Aksel, 2021.
- [38] Monia MECHIRGUI, COMMANDE OPTIMALE MINIMISANT LA CONSOMMATION D'ÉNERGIE D'UN DRONE UTILISÉ COMME RELAI DE COMMUNICATION.
- [39] I. K. Peddle, T. Jones et J. Treurnicht, PracticalNearHover Flight Control of a Ducted Fan (SLADe) , Control Engineering Practice, vol. 17, no. 1, p. 48 58, 2009.
- [40] AIT RAMDANE. N, AIT KACI D « Réalisation d'un capteur de position en vue d'autopiloter une machine synchrone a aimants permanents », PFE DEUA, UMMTO, 2005.
- [41] D. Ross, E. Deguine, and M. Camus, Asservissement par PID. Université de lille 1,2010.

Bibliographie

- [42] Calcul et construction d'un avion [en ligne]. Disponible sur: http://www.gilbert-pernot.fr/helice_calcul_fabrication.html
- [43] R. Longchamp, << Commande Sous-optimale Adaptative de Systèmes Nonlineaire >>. Thèse PhD, EPFL, 1978.
- [44] UAV forum: <http://www.uavforum.com/library/photo.htm>. Juillet 2004
- [45] Moteur brushless, le principe [en ligne]. Disponible sur: <http://www.dronekas.com/drone-principe-du-moteur-brushless>
- [46] Rodolphe Jobard, Les drones fonctionnement, télépilotage, applications, réglementation. Paris, Eyrolles (2016), 2ème édition. p48-53. ISBN: 978-2-212--14189-4
- [47] Comment choisir les composants d'un quadricoptère [en ligne]. Disponible sur: https://projetdroneamateur.wordpress.com/choix_composant/
- [48] MEMS Accelerometer Gyroscope Magnetometer & Arduino . Forum « How to Mechatronics » . Disponible en ligne sur : <http://howtomechatronics.com/how-it-works/electrical-engineering/mems-accelerometer-gyroscope-magnetometer-arduino/>
- [49] [Comment fonctionne un accéléromètre de smartphone ?](https://couleur-science.eu/?d=669308--comment-fonctionne-un-accelerometre-de-smartphone) Forum « Couleur science » Disponible en ligne sur : <https://couleur-science.eu/?d=669308--comment-fonctionne-un-accelerometre-de-smartphone>
- [50] Drone Icon Pack. FlatIcon. Disponible en ligne sur : <https://www.flaticon.com/packs/drone>