

REPUBLIQUE ALGERIENNE DEMOCRATIQUE ET POPULAIRE

Ministère de l'Enseignement Supérieur et de la Recherche Scientifique

Ecole Nationale Polytechnique



Département d'Automatique

Mémoire de projet de fin d'études
pour l'obtention du diplôme d'ingénieur d'état en
Automatique

Intitulé

**Commande et simulation par la
méthode du Hardware In the Loop
d'un quadri-rotor**

Réalisé par : **Maroua BAKOUR**

Présentée et soutenue publiquement le 20/06/2017

Composition du Jury :

Président	Mr. Omar STIHI,	Chargé de cours	ENP
Promoters	Mr. Mohamed TADJINE,	Professeur	ENP
	Mr. Messaoud CHAKIR ,	MCB ENP	ENP
Examineur	Mr. Lahcene ABDELOUEL,	Chargé de cours	ENP

REPUBLIQUE ALGERIENNE DEMOCRATIQUE ET POPULAIRE

Ministère de l'Enseignement Supérieur et de la Recherche Scientifique

Ecole Nationale Polytechnique



Département d'Automatique

Mémoire de projet de fin d'études
pour l'obtention du diplôme d'ingénieur d'état en
Automatique

Intitulé

**Commande et simulation par la
méthode du Hardware In the Loop
d'un quadri-rotor**

Réalisé par : **Maroua BAKOUR**

Présentée et soutenue publiquement le 20/06/2017

Composition du Jury :

Président	Mr. Omar STIHI,	Chargé de cours	ENP
Promoters	Mr. Mohamed TADJINE,	Professeur	ENP
	Mr. Messaoud CHAKIR ,	MCB ENP	ENP
Examineur	Mr. Lahcene ABDELOUEL,	Chargé de cours	ENP

ملخص

ملخص هذه الأطروحة يهتم بتصميم خوارزميات التحكم و التوجيه بالطائرة الموجهة رباعية المحرك. والغرض من وراء هذه الخوارزميات هو تطوير قدرة الطائرة على تتبع المسارات المحددة بشرط ضمان التوازن و الدقة في الاداء. على أساس نموذج ديناميكي يتم إجراء محاكاة الأجهزة في حلقة قبل أي اختبارات فعلية لتقييم الأداء الصحيح و السليم. واجهة رسومية تسمح بعرض مختلف معطيات تجارب التحليق

الكلمات المفتاحية: التوجيه،التصميم ، التحكم، طائرة بدون طيار، محاكاة الأجهزة في حلقة.

Abstract

This thesis focuses on the design and implementation of Guidance, Control and modelisation algorithms in a quadcopter. The purpose behind those algorithms is that the quadcopters can track a pre-determined path, while ensuring attitude and position stability also good accuracy of monitored points. A simulation of the hardware in the loop based of the dynamic model is performed before any tests to assess the correct functioning of the controller and the embedded module. A graphical interface allows the display of real-time flight parameters.

Key words: Guidance, Control, Modelisation ,Quadcopter, hardware in the loop...

Résumé

Ce mémoire s'intéresse à la conception et l'implémentation des algorithmes de guidage, contrôle et modélisation dans un drone de type quadri-rotor. Le but derrière ces algorithmes est que le quadri-rotors puisse suivre une trajectoire déterminée, tout en assurant la stabilité en attitude aussi en position et la bonne précision des points suivies. On se basant sur le modèle dynamique, une simulation du matériel dans la boucle (Hardware In the Loop) est effectuée avant le test réel afin d'évaluer le bon fonctionnement du contrôleur ainsi que le module embarqué. Une interface graphique assure l'affichage des paramètres de vol en temps réelle.

Mot clés : Guidage, Control, Modélisation quadri-rotor, matériel dans la boucle

REMERCIEMENT

Je tiens à remercier toutes les personnes qui ont contribué au succès de mon projet.

Tout d'abord, j'adresse mes remerciements à mes parents, sœurs, unique frère, mes grandes tantes Bakour Tlidja, et Bakour Kheira, ainsi que mon grand père Houcine Bakour qui m'ont beaucoup aidées avec leurs encouragements pendant toutes mes années d'études.

Je remercie vivement Kheiredinne Choutri pour son aide, son assistance et ses conseils m'ont permis de cibler mon objectif et d'assurer un bon déroulement du projet.

Je tiens à remercier vivement mes encadreurs du projet, Mr Chakir Messaoud et Mr Tadjine Mohamed, pour leurs temps et le partage de leurs expertises au quotidien. Grâce aussi à leurs confiances j'ai pu accomplir un bon peu de mon projet.

Enfin, je tiens à remercier tous les enseignants du département d'Automatique de l'Ecole Nationale Polytechnique d'Alger et mes camarades de la promotion 2017.

Maroua bakour

TABLE DES MATIERES

NOMENCLATURES

LISTE DES FIGURES

LISTE DES TABLEAUX

INTRODUCTION GENERALE	14
CHAPITRE I : MODELISATION DYNAMIQUE DU QUADRIROTO	
I.1.INTRODUCTION	16
I.2. DESCRIPTION GENERALE DU QUADRIROTOR	17
I.2.1. Description Architecturale	17
I.2.2. Objectif de l'étude	17
I.3.LES MOUVEMENT DU QUADRIROTOR	19
I.3.1. Mouvement du roulis	19
I.3.2. Mouvement de tangage	19
I.3.3. Mouvement de lacet	20
I.4. RELATION ENTRE LES ANGLES ET LES VITESSES ANGULAIRES	
I.4.1. Vol vertical	21
I.4.2. Vol stationnaire	21
I.4.3. Vol de translation	21
I.5. MODELE DYNAMIQUE DU QUADRIROTOR	22
I.5.1. Angles d'EULER	23

I.5.1.1. La matrice de passage	23
I.5.1.2. Vitesses linéaires	24
I.5.1.3. Relation entre les angles et les vitesses angulaire d'Euler	24
I.5.2. Forces et moments principaux	25
I.5.2.1. Les forces (dynamique de translation).....	25
I.5.2.2. Les moments (dynamique de rotation)	27
I.5.3. Développement du modèle mathématique selon Newton-Euler	29
I.5.3.1. Equations de mouvement de translation.....	30
I.5.3.2. Equations de mouvement de rotation	31
I.5.3.3. Liaisons non holonomes	32
I.6. MODILISATION AVEC LES QUATERNIONS	34
I.6.1. Modèle dynamique de translation	34
I.6.2. Modèle dynamique de rotation	34
I.6.3. Relation entre les quaternions et les vitesses angulaires.....	35
I.6.4. Avantages d'utilisation des quaternions	35
I.7. LA DYNAMIQUE DES MOTEURS	36
I.8. CONCLUSION	39
CHAPITRE II : COMMANDE DU QUADRIROTOR	
II.1. INTRODUCTION.....	40
II.2. MODELE SIMULINK DU QUADRIROTOR	41
II.2.1. Simulation en boucle ouverte	43
II.2.2. Résultats de simulation	44

II.3. CAHIER DE CHARGES	46
II.4. COMMANDE D'ATTITUDE ET DE POSITION DU QUADRIROTOR	
II.4.1. Linéarisation	47
II.4.2. Les commandes	48
II.4.3. Les signaux de commande	49
II.5. PRINCIPE DE REGULATION PAR « PID ».....	50
II.6. PRINCIPE DE LA COMMANDE PAR LA METHODE DU « LOOP-SHAPING ».....	53
II.7. SYNTHESE DES LOIS DE COMMANDE POUR LE QUADRIROTOR	55
II.7.1. Commande par le régulateur PID avec la methode du Loop-Shaping	56
II.7.2. Résultats de simulation	60
II.8. CONCLUSION	64

CHAPITRE III : HARDWARE IN THE LOOP

III.1. INTRODUCTION	65
III.2. MATERIEL UTILISE	66
III.2.1. Présentation du régulateur de vol	66
III.2.1.1. Gyromètre.....	67
III.2.1.2. Accéléromètre	67
III.2.1.3. Baromètre	67
III.2.1.4. GPS	67
III.2.2. Architecture de l'Ardu-pilot utilisé.....	68
III.2.3. APM2 Simulink Blockset	67

III.3. GENERATION DE TRAJECTOIRE	69
III.3.1. Guidage par champ de vecteurs	70
III.3.2. Résultats de simulation de la trajectoire sur SIMULINK/Matlab.....	72
III.4. LA SIMULATION AVEC HARDWARE IN THE LOOP	73
III.4.1. Simulation sur ordinateur et simulation Hardware In the Loop	74
III.4.1.1. L'architecture du Harware In the Loop.....	74
III.4.1.2. Chemin de Simulation "HIL".....	76
III.5. INTARFACE GRAPHIQUE	76
III.6. CONCLUSION	78
CONCLUSION GENERALE.....	79
BIBLIOGRAPHIE.....	81
ANNEXE 1.....	84
ANNEXE 2	92
ANNEXE 3	93
ANNEXE 4	95

NOMENCLATURES

e : Un repère fixe lié à la terre.

b Un repère mobile.

F : Force de poussée

Ω : Vitesse de rotation dans le repère fixe

V : *Vitesse* linéaire dans le repère fixe

R : Matrice de rotation

T : Matrice de transformation

ξ : Vecteur de position

x : Coordonnée du centre de gravité G du quadri-rotor suivant X_e .

y : Coordonnée du centre de gravité G du quadri-rotor suivant Y_e .

z : Coordonnée du centre de gravité G du quadri-rotor suivant Z_e .

ϕ : Angle de roulis

θ : Angle de tangage

ψ : Angle de lacet

M : La masse totale.

g : La gravité

ω : Vitesse de rotation de moteur

ω_d : Vitesse de rotation désirée des moteurs

τ : Couple d'entrée des moteurs

u : La commande

x : Variable d'état

x_d : Etat désiré

e : Erreur de poursuite

t : Variable de temps

P : Le poids du quadri-rotor.

b : Le coefficient de portance.

F : La force de portance.

F_T : La force de trainée.

C_d : Le coefficient de trainée.

ρ : Densité de l'air.

$F_{\text{perturbation}}$: La force de perturbation

\dot{V} : La première dérivée de la vitesse linéaire.

J : Matrice d'inertie symétrique de dimension (3x3).

$M_{pousé}$: Le Moment de la poussée.

k : Le coefficient de traînée.

M_{gyr} : Le moment gyroscopique.

M_d : Moment du aux forces de traînée.

Ω : Le vecteur des vitesses angulaires.

L : L'inductance.

fem : Force électromotrice.

V_s : Source de tension.

i : Courant d'armature.

T_e : Le couple électrique.

K_f : La constante de friction.

J : Inertie du rotor.

ω_i : La vitesse angulaire.

K_e : La constante de la fem de retour.

K_t : La constante du couple électrique.

a, b, c, d : Eléments du quaternion unitaire

\vec{g} : Le vecteur de gravité..

x_A, y_A, z_A : Coordonnées connues du point A.

x_B, y_B, z_B : Coordonnées connues du point B.

x_C, y_C, z_C : Coordonnées connues du point C.

x_P, y_P, z_P : Coordonnées inconnues du point P.

d : Mesure de distance entre le récepteur et le satellite.

K_p : Le coefficient proportionnel.

K_i : Le coefficient intégral.

K_d : Le coefficient dérivé.

X^f : Le heading.

$W1X, 1Y$: Les coordonnées X, Y respectivement du waypoint 1.

$W2X, 2Y$: Les coordonnées X, Y respectivement du waypoint 2.

x, y, z : Coordonnées cartésiennes.

LISTE DES FIGURES :

Chapitre I

- Figure I.2.1 : Prototype étudié dans ce sujet [6]
Figure I.2.2 : AR.Drone de Parrot
Figure I.2.1 : Structure générale d'un quadri-rotor [1]
Figure I.2.2 : Présentation de la démarche de modélisation. [6]
Figure I.3.1 : Illustration du mouvement de roulis. [1]
Figure I.3.2 : Illustration du mouvement de tangage. [1]
Figure I.3.3 : Illustration du mouvement de lacet. [1]
Figure I.4.1 : Illustration du mode vertical. [1]
Figure I.4.2 : Illustration du mode stationnaire. [1]
Figure I.4.3 : Illustration du mode de translation. [1]
Figure I.5 : Géométrie du quadri-rotor [1]
Figure I.7.1: Modèle électrique équivalent du moteur DC. [6]
Figure I.7.2: Circuit équivalent du moteur DC. [6]

Chapitre II

- Figure.II.1 : Modèle SIMULINK du quadri-rotor.
Figure II.2 : Mouvement vertical selon z.
Figure II.3 : Mouvement du lacet.
Figure II.4 : Mouvement de roulis.
Figure II.5 : Mouvement de tangage
Figure II.6 : Variation d'altitude dans le cas des mouvements de tangage, roulis et lacet.
Figure II.7 : choix des références du réglage
Figure.II.8: block 'Motor Mixer'
Figure.II.9 : Le principe de commande par PID [10]
Figure.II.10 : Le principe de commande par PID. [10]
Figure.II.11 : La structure améliorée du PID. [9]
Figure.II.12 : Exemple de tracé de Bode avec PD. [13]
Figure.II.13 : Illustration de la structure de commande du quadri-rotor. [1]
Figure.II.14 : Tracer de Bode de roulis (ϕ).

Figure.II.15 : Tracer de Bode de tangage (θ)

Figure.II.16 : Tracer de Bode de lacet (ϕ).

Figure.II.17 : Tracer de Bode pour le Z.

Figure.II.18 : Réponse de la position d'un contrôleur PID.

Figure.II.19 : Réponse d'attitude d'un contrôleur PID.

Figure.III.20 : Evolution des commandes U_1 ; U_2 ; U_3 et U_4 .

Figure.II.21 : Réponses des signaux de commande PWM d'un contrôleur PID

Chapitre III

Figure.III.1 : Régulateur du vol (Ardu-pilot).[28]

Figure. III.2 : l'Ardu-pilot [28]

Figure. III.3 : la librairie de l'Ardu-pilot [30]

Figure. III.3 : Différentes étages de commande [25]

Figure. III.4 : Exemple de segment de trajectoire linéaire.[25]

Figure. III.5 : Le modèle du contrôleur associée à l'algorithme de guidage.[27]

Figure. III.6: Le modèle SIMULINK/Matlab de la génération de la trajectoire ainsi que le contrôleur.

Figure. III.7: Simulation de la trajectoire rectangulaire

Figure. III.8: Simulation de la trajectoire rectangulaire entre X et Y

Figure. III.9: Schéma du principe de la simulation Hardware In the Loop.[31]

Figure. III.9: Architecture globale de la plateforme expérimentale [31]

Figure. III.10: Illustration de la structure de commande par Hardware In the Loop

Figure.III.18 : Mission Planner interface graphique

ANNEXES

Figure I.1 : Moteur brushless[17]

Figure I.2 : Batterie LIPO [17]

Figure I.3 : Moteur brushless[17]

Figure I.4 : Capteur Altimètre Ultrason [17]

Figure I.5: Centrale Inertielle [17]

Figure I.6 : l'hélice du quadri-rotor [17]

Figure I.7 : le support du quadri-rotor [17]

Figure I.8 : Composants constituant le Kit DJI F250 [28]

Figure I.9 : assemblage DJI 250 [28]

Figure II.1 : Repère NED (en rouge) et ENU (en bleu)

Figure II.2 : Repère du mobile

Figure VI.1 : fenêtre du tuning d'un PID sur SIMULINK /Matlab

LISTE DES TABLEAUX :

Chapitre II

Tableau II.1 : Paramètre des moteurs et de la batterie LIPO [annexe1]

Tableau II.2 : Paramètre du quadri-rotor [annexe 1]

Tableau II.3 : Critère du cahier des charges

Tableau II.4 : les fréquences de coupure et désirées

Tableau II.5 : Gains des PID de la commande.

Chapitre III

Tableau.III.1 : Acteurs majeurs des régulateurs de vol. [26]

Annexes

Tableau I.1 : Paramètres du moteur brushless utilisé

Tableau I.2 : Paramètres de la batterie LiPo utilisé

Tableau I.3 : Paramètres du driver utilisé

Tableau I.4 : Paramètres de l'Altimètre Ultrason utilisé

Tableau I.5 : Paramètres de la Centrale Inertielle utilisé.

Tableau I.6 : Paramètres des hélices utilisé

Tableau I.7 : Paramètres de la structure du quadri-rotor utilisé

INTRODUCTION GENERALE

Les drones sont devenus plus accessibles et moins chers. Aussi du à l'avancement de la technologie, les microprocesseurs sont plus réduits dans la dimension et le prix, avec suffisamment d'énergie pour exécuter différentes tâches. Dû à cette évolution, les UAV (autonomous aircraft vehicles) se sont développés, citant les multi-rotors. L'application de ces engins paraît illimitée ; ils peuvent être utilisés pour la géo-archéologie (Oczipka et al, 2009), les opérations de sauvetage durant les catastrophes naturelles (Apvrille, Tanzi, & Dugelay, 2014), le domaine de l'agriculture (Tripicchio, Satler, Dabisias, Ruffaldi, & Avizzano, 2015), les missions de surveillance contre le braconnage en Afrique, et plusieurs d'autres applications.

En occurrence du développement moderne dans la technologie des drones, chaque chercheur, étudiante ou personne ému par le savoir doit prendre et comprendre en vigueur ce développement. Moi tant que étudiante en dernière année d'étude d'ingénieur à l'école Nationale Polytechnique d'Alger et mener par la curiosité de cette technologie je voulais traiter une partie d'un grand projet ouvert de recherche qui sera travaillé le long de cette thèse.

Pour qu'un drone puisse accomplir sa mission, il doit être équipé d'un matériel embarqué capable d'exécuter les algorithmes (Guidage, contrôle et modélisation), c'est à base de ses algorithmes que les commandes d'attitude et de position sont générés afin de suivre les 'waypoints' de la trajectoire désirée.

Une fois le quadri-rotor est choisi, les paramètres du vol seront définis et ils seront utilisés comme des entrées pour les commandes de contrôle. Dans le cadre de notre projet en va se limiter aux commandes linéaires à cause de nos matériels embarqués qui ne peut pas traiter des algorithmes complexes. Les sorties des commandes de contrôle sont des signaux PWM (Pulse Width Modulation) qui commandent les moteurs afin d'assurer la stabilité d'attitude d'un côté, et de suivre les positions imposées par l'algorithme de guidage.

Il est important de choisir judicieusement les instruments embarqués afin d'obtenir une intégration optimale de l'ensemble. Pour ce fait plusieurs modèles de quadri-rotor ainsi que le matériel embarqué disponible dans ont été présentés avec justification de ceux choisis.

On note que la modélisation du quadri-rotor est le point le plus critique dans l'implémentation des algorithmes de commande et de guidage, puisque ces derniers sont tous basé sur le modèle dynamique choisis, de plus les modèle des actuateurs ainsi que les effets non linéaire sont prisent en considération de plus les algorithmes de commande et de guidage se rapprochent de la réalité.

La simulation Hardware In the Loop est l'étape de transition entre la simulation et le teste réelle, elle permet non seulement d'évaluer les capacités du contrôleur mais aussi de diagnostiquer à l'avance des problèmes d'intégration et d'explorer des scénarios d'anomalies matérielles et logicielles.

Ce mémoire s'articule autour de trois chapitres :

- Le premier chapitre est consacré à la modélisation dynamique du quadri-rotor, d'abord on fait une description de ce système, structure générale et principe de vol. Puis, nous mettons en évidence la modélisation dynamique du quadri-rotor, enfin, on donne le modèle d'état en vue de la commande de ce système.
- Dans le deuxième chapitre, on entame les fondements théoriques de la technique de commande ou on établie une nouvelle méthode de modélisation sur le logiciel SIMULINK/ Matlab de notre quadri-rotor Puis, on applique les lois de commandes pour la stabilisation de la position et l'attitude du quadri-rotor. Pour se faire, une méthode a était étudiées qui est la méthode classique basée sur un contrôleur PID en utilisant le concept du Loop-shaping afin de définir les gains du régulateur.
- Et le troisième chapitre s'intéresse à la méthode de génération de la trajectoire et la simulation du matérielle dans la boucle.

Enfin nous concluons par un bilan et des perspectives de notre travail.

I.1.Introduction :

Un système est un ensemble d'objets ou de phénomènes liés entre eux et isolés artificiellement du monde extérieur. Sa modélisation regroupe l'ensemble des techniques permettant de disposer d'une représentation mathématique de celui-ci. [1]

La modélisation théorique nécessite une connaissance précise des phénomènes intervenant dans le système et une aptitude de les représenter mathématiquement. Les systèmes physiques sont souvent complexes et difficilement exploitables, notamment pour la commande et l'observation. [1]

Les grands soucis de la modélisation sont : la précision et la fidélité du modèle vis-à-vis le processus réel, ainsi que l'adéquation de ce modèle à une forme mathématiquement exploitable. [1]

Et afin de concevoir un contrôleur de vol, on doit d'abord comprendre profondément les mouvements de l'avion, sa dynamique et par conséquent ses équations dynamiques. Cette compréhension est nécessaire pas simplement pour la conception du contrôleur, mais aussi afin de s'assurer que les simulations du véhicule dépeindront un comportement aussi proche que possible de la réalité quand la commande est appliquée.[2]

Beaucoup de tentatives de modélisation d'un drone (Quadri-rotor) sont enregistrées dans la littérature. On cite parmi-elles, celle établie par Bouabdallah [3] via la méthode d'Euler-Lagrange, le modèle présenté par KHEBBACHE [4] basé sur le formalisme de Newton a été obtenu à partir de la dynamique d'un corps rigide associé au fuselage auquel sont ajoutées les forces aérodynamiques générées par les rotors, ce modèle intègre en plus la dynamique des actionneurs...

Dans le cadre de notre travail, l'approche de modélisation adoptée est basée sur le formalisme de Newton, aussi bien l'implémentation des Quaternion afin d'éliminé les singularités trouvées.

I.2. DESCRIPTION GENERALE DU QUADRIROTOR :

I.1.Introduction :

La configuration de mini-drone en particulier a connu un succès grandissant depuis plus de 10 ans. Il s'agit d'un drone possédant 4 hélices, d'où son nom de « quadri-rotor » ou « quadri-rotor ». Ce type de mini-drone est étudié dans de nombreuses universités ou écoles d'ingénieurs (figure I.2.1) et est même vendu au grand public comme un jeu pour l'extérieur ou l'intérieur (un exemple est donné sur la figure I.2.2).



Figure I.2.1 : Prototype étudié dans ce sujet [6]

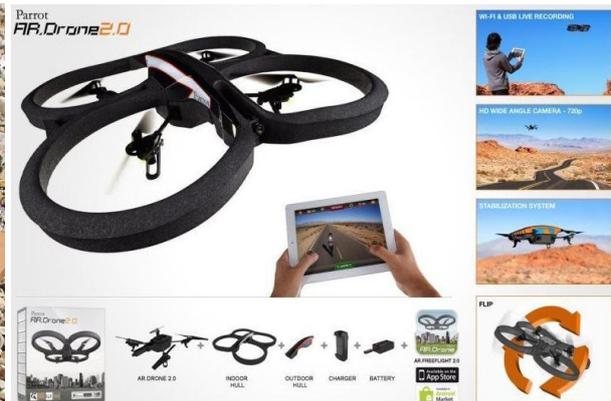


Figure I.2.2 : AR.Drone de Parrot

I.2.1. Description architecturale :

Un quadri-rotor est constitué d'un corps principal mis en mouvement par 4 hélices entraînées chacune par un moteur électrique. Tous les composants électroniques du drone, dont les 4 moteurs, sont alimentés par une batterie Lithium Polymère, 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.

Les 4 rotors sont généralement placés aux extrémités d'une croix, et l'électronique du contrôle est habituellement placée au centre de la croix. Afin d'éviter que l'appareil tourne sur lui-même sur son axe du lacet, il est nécessaire que deux hélices tournent dans un sens, et les deux autres dans un autre sens. Pour pouvoir diriger l'appareil, il est nécessaire que chaque couple d'hélice tournant dans le même sens soit placé aux extrémités opposées d'une branche de la croix. (Figure I.2.1)

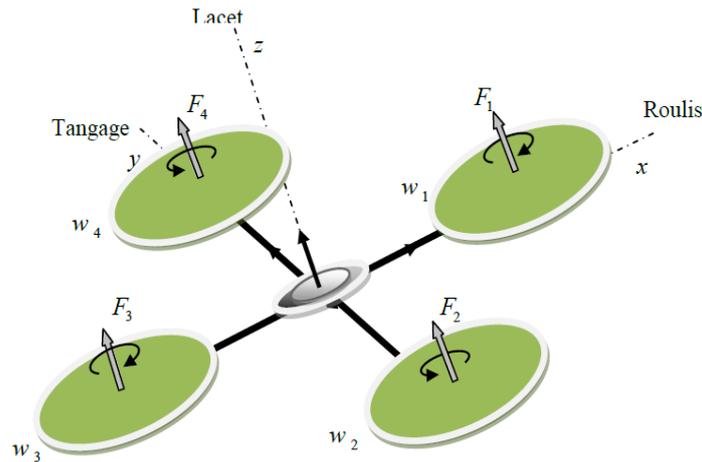


Figure I.2.1 : Structure générale d'un quadri-rotor [1]

Notre modèle du quadri-rotor est équipé de plusieurs capteurs, à fonctionnalités bien différentes : un capteur à ultrasons qui est réellement 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, on peut aussi lui ajouter à l'aide de l'arduino-pilot (chapitre III) une antenne GPS et deux caméras pointant l'une vers l'avant du drone et l'autre vers le sol.

Le fonctionnement du quadri-rotor est assez particulier. En faisant varier astucieusement la puissance des moteurs, il est possible de le faire monter/descendre, de l'incliner à gauche / à droite (roulis) ou en avant / en arrière (tangage), ou encore de le faire pivoter sur lui-même (lacet) le quadri-rotor a six degrés de liberté, trois mouvements de rotation et trois mouvements de translation, ces six degrés doivent être commandés à l'aide de quatre déclencheurs seulement, donc c'est un système sous actionné (le nombre des entrées inférieure au nombre des sorties).[1]

I.2.2. Objectif de l'étude :

L'objectif de l'étude est déterminer les paramètres de notre quadri-rotor et établir un modèle d'état en basant sur les forces et les moments appliqués, Pour ce faire, il est essentiel de modéliser le système étudié selon la démarche décrite ci-après et illustrée sur la figure I.2.2

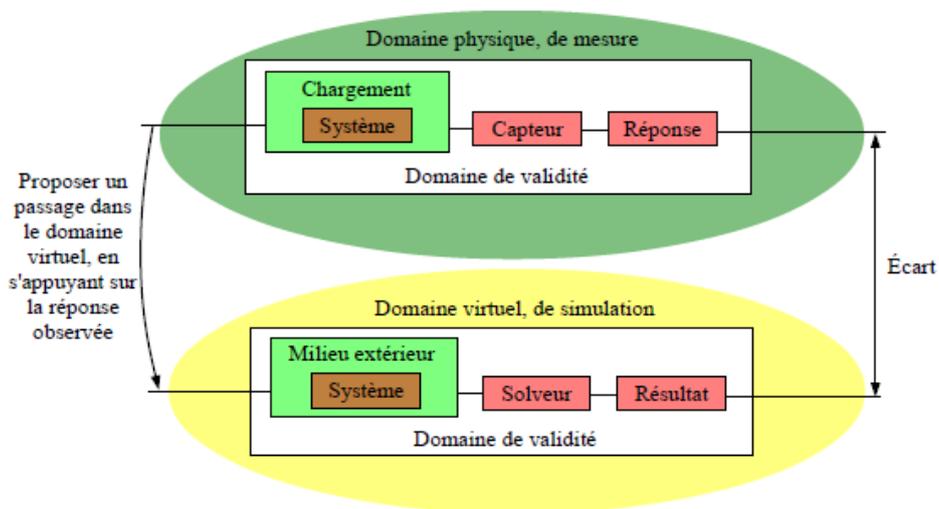


Figure I.2.2 : Présentation de la démarche de modélisation. [6]

La démarche de modélisation suivie dans cette étude se compose symboliquement de deux domaines: le domaine physique où l'on fait des mesures sur un système réel, et le domaine virtuel où l'on réalise des simulations sur un système virtuel (analytique, numérique,...) censé représenter le comportement du système réel. Le domaine physique est nécessaire car il est indispensable de s'appuyer sur l'observation des phénomènes réels pour proposer et valider un modèle de comportement et une modélisation du système et du milieu extérieur. Le solveur est un outil qui permet de résoudre les équations et qui donne le résultat de la simulation. La modélisation est pertinente si l'écart entre les réponses expérimentales et les résultats issus de la simulation est faible [6]. Dans notre cas on va utiliser des résultats du domaine physique (équations physiques, mathématique, identification des paramètres, objectifs de régulation du système...) et l'implémenter dans le domaine virtuel qui permet de tester le bon fonctionnement de l'algorithme en simulation avant de l'implémenter et de le tester sur le véritable système. L'étude présentée ici s'intéresse principalement au domaine virtuel de simulation.

I.3. LES MOUVEMENTS DU QUADRIROTOR :

Comme est défini dans la partie précédente, on rappelle que le quadri-rotor est un engin volant doté de quatre rotors placés aux extrémités d'une croix. Ce sont ces quatre rotors qui fournissent la force verticale (portance) qui permet à l'appareil de s'élever. Les mouvements possibles du quadri-rotor sont: [1]

I.3.1. Mouvement de roulis :

La figure (I.3.1) montre comment un mouvement de roulis est obtenu. Dans ce cas, on applique un couple autour de l'axe x , c'est-à-dire en appliquant une différence de poussée entre le rotor 2 et le rotor 4. Ce mouvement (rotation autour de l'axe x est couplé avec un mouvement de translation selon l'axe y .

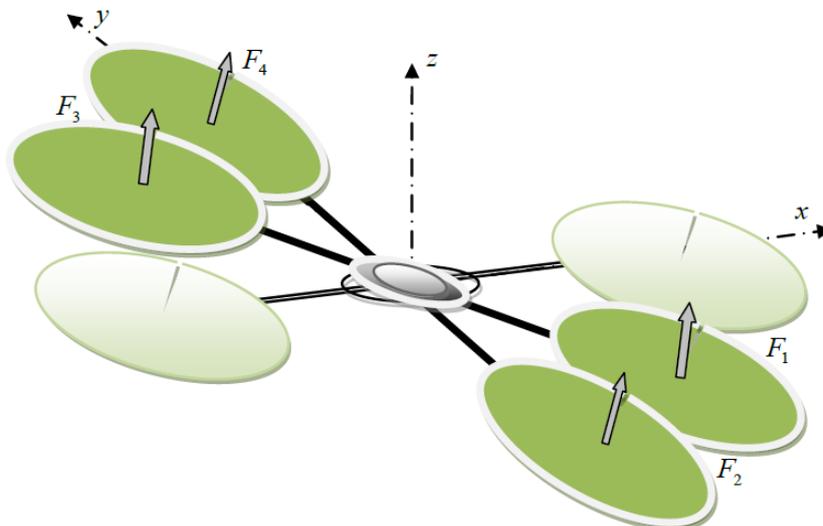


Figure I.3.1 : Illustration du mouvement de roulis. [1]

I.3.2. Mouvement de tangage :

La figure (I.3.2) montre comment un mouvement de tangage est obtenu. Dans ce cas, on applique un couple autour de l'axe y , c'est-à-dire en appliquant une différence de poussée entre le rotor 1 et le rotor 3. Ce mouvement (rotation autour de y) est couplé avec un mouvement de translation selon l'axe x .

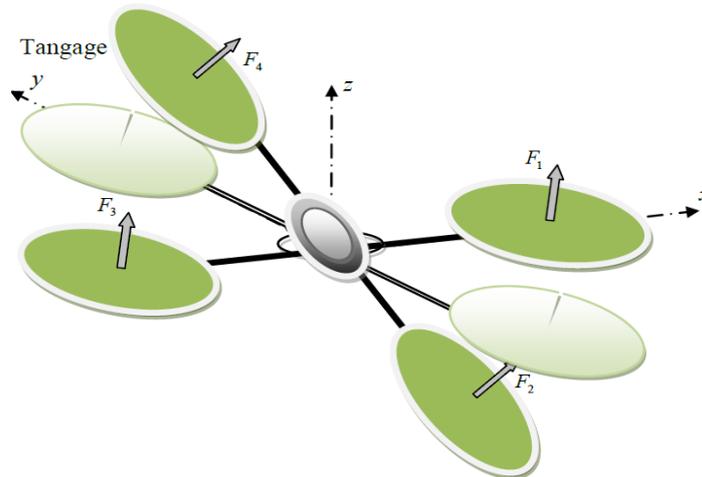


Figure I.3.2 : Illustration du mouvement de tangage. [1]

I.3.3. Mouvement de lacet :

La figure (I.3.3) montre comment le mouvement de lacet est obtenu. Dans ce cas, nous voulons appliquer un couple autour de l'axe z , qui est fait en appliquant une différence de vitesse entre les rotors $\{1,3\}$ et $\{2,4\}$. Ce mouvement n'est pas un résultat direct de la poussée produit par les propulseurs mais par les couples réactifs produits par la rotation des rotors. La direction de la force de poussée ne décale pas pendant le mouvement, mais l'augmentation de la force de portance dans une paire de rotors doit être égale à la diminution des autres paires pour s'assurer que toute la force de poussée demeure la même.

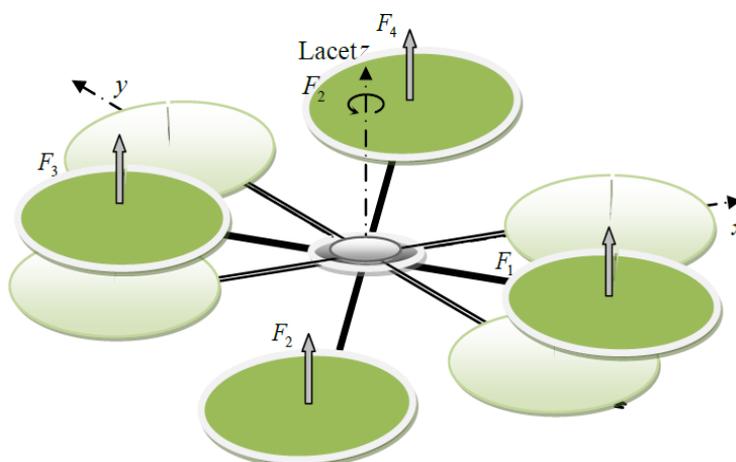


Figure I.3.3 : Illustration du mouvement de lacet. [1]

I.4. LES MODES DE VOL D'UN QUADRIROTOR :

En se basant sur les mouvements possibles, le drone peut effectuer trois modes de vol [4] :

I.4.1. Vol vertical :

La figure (I.4.1) montre comment le mode vertical est obtenu. Dans le vol vertical, la résultante aérodynamique et le poids total sont deux forces ayant la même direction mais de sens opposés. L'hélicoptère peut monter ou descendre, suivant l'effet aérodynamique soit supérieur ou inférieur au poids de l'appareil.

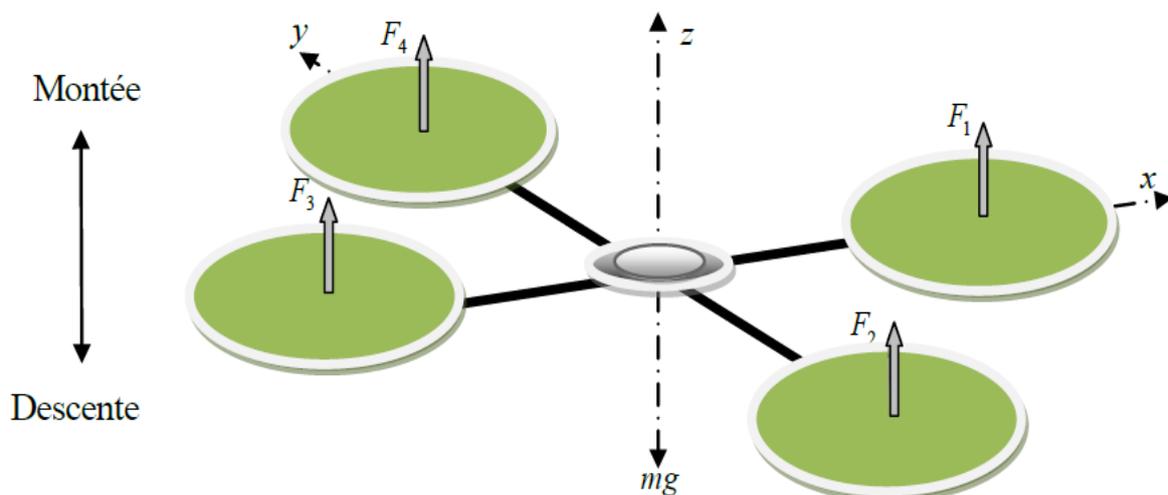


Figure I.4.1 : Illustration du mode vertical. [1]

I.4.2. Vol stationnaire :

La figure (I.4.2) montre comment le mode stationnaire est obtenu. Quand la force de Portance, et celle de pesanteur sont égales et opposées, l'hélicoptère reste immobile. On parle de vol stationnaire.

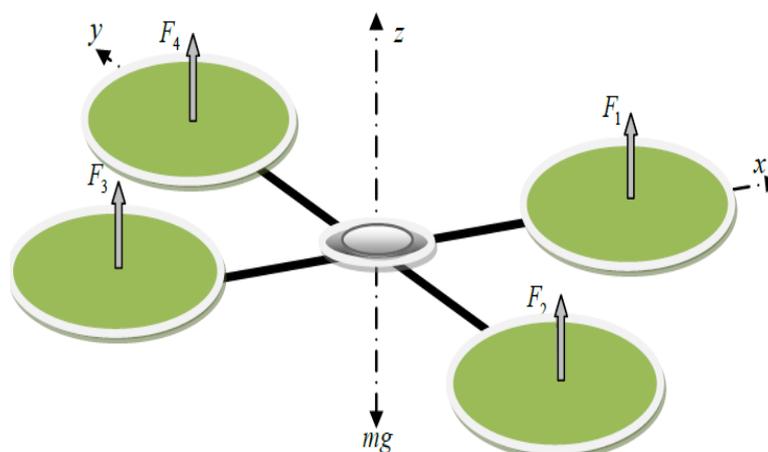


Figure I.4.2 : Illustration du mode stationnaire. [1]

I.4.3. Vol de translation :

La figure (I.4.3) montre comment le mode vertical est obtenu. Le vol de translation correspond à la navigation de l'hélicoptère sur un plan horizontal. Il est assuré en se basant sur les mouvements d'inclinaison tangage, et roulis.

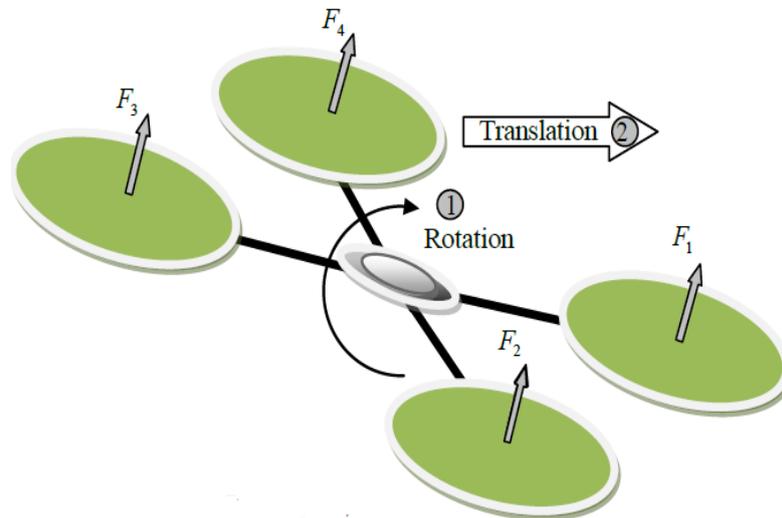


Figure I.4.3 : Illustration du mode de translation. [1]

I.5. MODELE DYNAMIQUE DU QUADRIROTOR :

La modélisation des quadri-rotors est une tâche délicate puisque la dynamique du système est fortement non linéaire et pleinement couplée. En réalité, un grand nombre d'effets physiques influent sur notre système, à cet effet, il est nécessaire dans un premier lieu d'imposer quelques hypothèses de travail afin d'éliminer ceux de plus faible importance, énoncées comme suite : [2]

- La structure du quadri-rotor est supposée rigide et symétrique, ce qui induit que la matrice d'inertie sera supposée diagonale.
- Les hélices sont supposées rigides pour pouvoir négliger l'effet de leur déformation lors de la rotation.
- Les forces de portance et de traînée sont proportionnelles aux carrés de la vitesse de rotation des moteurs, ce qui est une approximation très proche du comportement aérodynamique.

Pour définir l'orientation des forces et des moments qui agissent sur le quadri-rotor on a choisit :

- La structure type croix du quadri-rotor.
- L'orientation Est pour le X, Nord pour le Y et UP pour le Z. (Orientation NED).
- La numération des W_i comme la figure I.5 le montre. ($i=0\dots 4$)

Pour évaluer le modèle mathématique du quadri-rotor on utilise deux repères, un repère fixe lié à la terre « e » et un autre mobile « b » fixé au centre de masse du quadri-rotor

(figure.II.5). Le passage entre le repère mobile et le repère fixe est donné par une matrice dite matrice de transformation T qui contient l'orientation et la position du repère mobile par rapport au repère fixe (voir ANNEXE 2). [5]

$$T = \begin{bmatrix} \mathbf{R} & \mathbf{X} \\ \mathbf{0} & \mathbf{1} \end{bmatrix} \quad \text{Eq.I.5.1}$$

Avec

$$\mathbf{R} = \text{Rot}_x(\phi) \times \text{Rot}_y(\theta) \times \text{Rot}_z(\psi) = \begin{bmatrix} 1 & 0 & 0 \\ 0 & C\phi & -S\phi \\ 0 & S\phi & C\phi \end{bmatrix} \times \begin{bmatrix} C\theta & 0 & S\theta \\ 0 & 1 & 0 \\ -S\theta & 0 & C\theta \end{bmatrix} \times \begin{bmatrix} C\psi & -S\psi & 0 \\ S\psi & C\psi & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

Où \mathbf{R} est la matrice de rotation (décrit l'orientation de l'objet mobile), $\mathbf{X} = [x \ y \ z]$ est le vecteur de position. Pour déterminer les éléments de la matrice de rotation \mathbf{R} , on utilise les angles d'Euler.

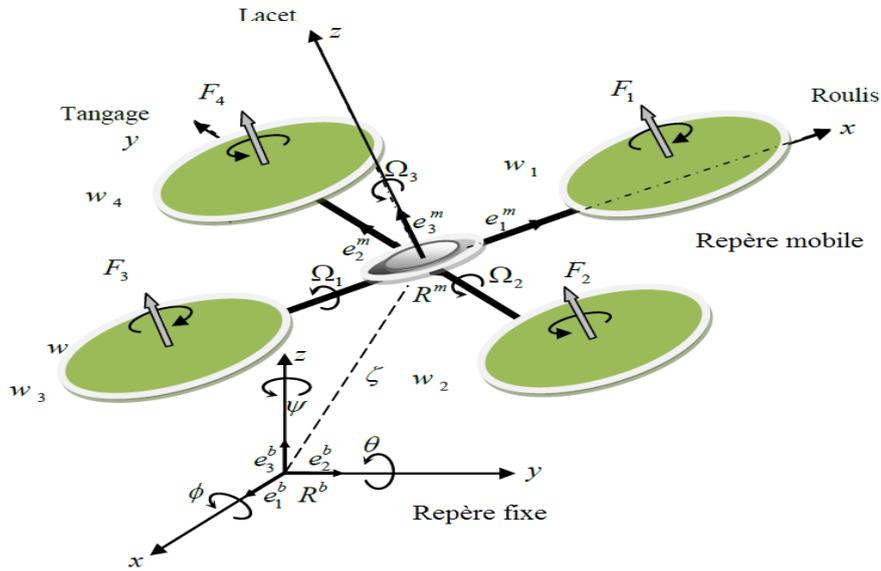


Figure I.5 : Géométrie du quadri-rotor [1]

Donc les paramètres qui permettent de décrire le mouvement du quadri-rotor sont $(x, y, z, \phi, \theta, \psi, v, \Omega)$ avec:

x : Coordonnée du centre de gravité G du quadri-rotor suivant X_n .

y : Coordonnée du centre de gravité G du quadri-rotor suivant Y_n .

z : Coordonnée du centre de gravité G du quadri-rotor suivant Z_n .

ϕ : L'angle de roulis, sa rotation est autour X_b ($-\frac{\pi}{2} < \phi < \frac{\pi}{2}$).

θ : L'angle de tangage, sa rotation est autour Y_b ($-\frac{\pi}{2} < \theta < \frac{\pi}{2}$).

ψ : L'angle de lacet, sa rotation est autour Z_b ($-\pi < \psi < \pi$).

$v = [u, v, w]^T$: La vitesse linéaire liée au quadri-rotor.

$\Omega = [p, q, r]$: La vitesse de rotation liée au même corps.

I.5.1. ANGLES d'Euler :

I.5.1.1. la matrice de passage R :

Au début le repère mobile coïncide avec le repère fixe, après le repère mobile fait un mouvement de rotation autour de l'axe x d'un angle de roulis ($-\frac{\pi}{2} < \phi < \frac{\pi}{2}$) suivi d'une rotation autour de l'axe y d'un angle de tangage ($-\frac{\pi}{2} < \theta < \frac{\pi}{2}$), suivi d'une rotation autour de l'axe z d'angle de lacet ($-\pi < \psi < \pi$). Donc on a la formule de la matrice de rotation R suivante :

$$R = Rot_x(\phi) \times Rot_y(\theta) \times Rot_z(\psi) = \begin{bmatrix} 1 & 0 & 0 \\ 0 & C\phi & -S\phi \\ 0 & S\phi & C\phi \end{bmatrix} \times \begin{bmatrix} C\theta & 0 & S\theta \\ 0 & 1 & 0 \\ -S\theta & 0 & C\theta \end{bmatrix} \times \begin{bmatrix} C\psi & -S\psi & 0 \\ S\psi & C\psi & 0 \\ 0 & 0 & 1 \end{bmatrix} \quad \text{Eq.I.5.2}$$

$$R = \begin{bmatrix} C\psi C\theta & S\phi S\theta C\psi - S\psi C\phi & C\phi S\theta C\psi + S\psi S\phi \\ S\psi C\theta & S\phi S\theta S\psi + C\psi C\theta & C\phi S\theta S\psi - S\psi C\phi \\ -S\theta & S\phi C\theta & C\phi C\theta \end{bmatrix} \quad \text{Eq.I.5.3}$$

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

I.5.1.2. Vitesses linéaires :

Les vitesses linéaires, v_x^e, v_y^e, v_z^e dans le repère fixe en fonction des vitesses linéaires v_x^b, v_y^b, v_z^b dans le repère mobile sont données par :

$$v = \begin{bmatrix} v_x^e \\ v_y^e \\ v_z^e \end{bmatrix} = R \times \begin{bmatrix} v_x^b \\ v_y^b \\ v_z^b \end{bmatrix} \quad \text{Eq.I.5.4}$$

I.5.1.3. Relation entre les angles et les vitesses angulaires d'Euler :

Si un solide tourne à une vitesse constante, sa vitesse angulaire Ω est constante. Par contre, les variations des angles d'Euler seront variables car elles dépendent des angles instantanés entre les axes des deux repères. La séquence des angles d'Euler est obtenue à partir de trois rotations successives : roulis, tangage et lacet.

Les vitesses de rotations $\Omega_1, \Omega_2, \Omega_3$ dans le repère fixe sont exprimées en fonction des vitesses de rotations p, q, r dans le repère fixe, et par $\dot{\phi}, \dot{\theta}, \dot{\psi}$ dans le repère mobile on a : (Plus de détails sur la transformation du repère fixe au repère mobile voir ANNEXE 2)

$$\Omega = R(X_0, \phi) = \begin{bmatrix} \Omega_1 \\ \Omega_2 \\ \Omega_3 \end{bmatrix} = \begin{bmatrix} \dot{\phi} \\ 0 \\ 0 \end{bmatrix} + Rot_x(\phi)^{-1} \begin{bmatrix} 0 \\ \dot{\theta} \\ 0 \end{bmatrix} + (Rot_y(\theta) Rot_x(\phi))^{-1} \begin{bmatrix} 0 \\ 0 \\ \dot{\psi} \end{bmatrix} \quad \text{Eq.I.5.5}$$

$$\begin{bmatrix} p \\ q \\ r \end{bmatrix} = \begin{bmatrix} 1 & 0 & -S\theta \\ 0 & C\phi & S\phi C\phi \\ 0 & -S\phi & C\phi C\phi \end{bmatrix} \begin{bmatrix} \dot{\phi} \\ \dot{\theta} \\ \dot{\psi} \end{bmatrix} \quad \text{Eq.I.5.6}$$

On aura:

$$\begin{bmatrix} \dot{\phi} \\ \dot{\theta} \\ \dot{\psi} \end{bmatrix} = [\zeta]^{-1} \begin{bmatrix} p \\ q \\ r \end{bmatrix} \quad \text{Eq.I.5.7}$$

Avec :

$$[\zeta]^{-1} = \begin{bmatrix} 1 & S\phi t(\theta) & C\phi t(\theta) \\ 0 & C\phi & -S\phi \\ 0 & \frac{S\phi}{C\theta} & \frac{C\phi}{C\theta} \end{bmatrix} \quad \text{Eq.I.5.8}$$

On remarque bien les fractions dans la dernière ligne de la matrice $[\zeta]^{-1}$ qui représentent des singularités pour le modèle d'étude pour cela on fait appel à une nouvelle méthode afin d'éliminer ces singularité.

I.5.2. Forces et moments principaux :

Les forces et les moments appliqués sur le quadri-rotor proviennent des forces générées par la rotation des hélices, de la précession gyroscopique ainsi que des effets aérodynamiques.

Nous présenterons ici chacune des forces et des moments séparément, ensuite nous les appliquerons au modèle du quadri-rotor et, pour finir, nous les ajouterons aux équations dynamiques.

Puisque plusieurs équations des sections suivantes demandent une analyse utilisant des notions de mécanique des fluides et d'analyse d'hélice qui dépassent le cadre de ce mémoire, nous ne présenterons ici qu'une vulgarisation des principes ainsi que les résultats nécessaires pour notre modélisation. Pour plus détails, le lecteur peut consulter les références suivantes.[3]

I.5.2.1. Les forces : (dynamique de translation) :

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

$$\sum F_{ext} = \frac{d(mV)}{dt} = m\dot{V} \quad \text{Eq.I.5.9}$$

Les forces extérieures appliquées au quadri-rotor sont :

❖ **La force du poids** : elle est donnée par :

$$\mathbf{P} = \mathbf{m.g} \quad \text{Eq.I.5.10}$$

Avec : '**m**' la masse totale du quadri-rotor et '**g**' la gravité.

❖ **La force de portance** : Elle est perpendiculaire à l'écoulement d'air et dirigée vers le haut, c'est-à-dire qu'elle a tendance à faire élever le quadri-rotor. Elle représente la force totale produite par les quatre hélices. Elle est donnée par

$$F_p = b(w_1^2 + w_2^2 + w_3^2 + w_4^2) \quad \text{Eq.I.5.11}$$

Avec : '**b**' est le coefficient de portance, il dépend de la forme et le nombre des pales et la densité de l'air.

On doit exprimer cette force dans le repère inertiel. Pour cela, on utilise la matrice de rotation R.

$$F_p = b(w_1^2 + w_2^2 + w_3^2 + w_4^2).R \quad \text{Eq.I.5.12}$$

❖ **La force de traînée** : C'est la résultante des forces qui s'opposent au mouvement du quadri-rotor dans l'air. Elle s'exerce parallèlement au sol, de même direction que le mouvement du quadri-rotor, mais dans le sens opposé. Elle représente en quelque sorte les forces de frottement sur l'objet. Elle est donnée par: [6]

$$F_t = \frac{1}{2} \cdot \rho \cdot V^2 \cdot S_w \cdot C_d \quad \text{Eq.I.5.13}$$

Où :

C_d : le coefficient de traînée.

ρ : Densité de l'air

V : La vitesse linéaire.

S_w : La surface de référence.

Où encore plus :

$$F_t = K_{ft} \cdot V \quad \text{Eq.I.5.14}$$

Avec :

$$K_{ft} = \frac{1}{2} \cdot \rho \cdot V \cdot S_w \cdot C_d \quad \text{Eq.I.5.15}$$

Qui représente le coefficient de traînée de translation

❖ **La force de perturbation** : D'autres forces comme la force de Coriolis de la terre, le vent et les forces d'Euler sont considérées comme une perturbation résumée comme **F_{perturbation}**.

→ Les forces expliquées précédemment se résument ensemble pour trouver la somme des forces totales agissant sur le quadri-rotor.

En remplaçant dans **Eq.I.5.9** :

$$\mathbf{m}\dot{\mathbf{V}} = \mathbf{P} + \mathbf{F}_p + \mathbf{F}_t + \mathbf{F}_{perturbation} \quad \text{Eq.I.5.16}$$

I.5.2.2. Les moments : (dynamique de rotation) :

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

$$\sum \mathbf{M}_{ext} = \frac{d(\mathbf{J}\boldsymbol{\Omega})}{dt} \quad \text{Eq.I.5.17}$$

Et comme la vitesse angulaire est exprimé dans le repère lié au quadri-rotor, alors :

$$\frac{d(\mathbf{J}\boldsymbol{\Omega})}{dt} = \mathbf{J}\dot{\boldsymbol{\Omega}} + \boldsymbol{\Omega} \wedge \boldsymbol{\Omega}\mathbf{J} \quad \text{Eq.I.5.18}$$

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

$$\mathbf{J} = \begin{bmatrix} I_x & 0 & 0 \\ 0 & I_y & 0 \\ 0 & 0 & I_z \end{bmatrix} \quad \text{Eq.I.5.19}$$

$\boldsymbol{\Omega} \wedge \boldsymbol{\Omega}\mathbf{J}$: représente les moments de Coriolis qui seront négligés par la suite.

Il y a plusieurs moments agissant sur le quadri-rotor, ces moments sont dus aux forces de poussée et de traînée et aux effets gyroscopiques. [1]

❖ Le Moment de la poussée :

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

$$\mathbf{M}_x = l (\mathbf{F}_4 - \mathbf{F}_2) = l\mathbf{b} (\omega_4^2 - \omega_2^2) \quad \text{Eq.I.5.20}$$

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

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

$$M_y = l (F_3 - F_1) = lb (\omega_3^2 - \omega_1^2) \quad \text{Eq.I.5.21}$$

❖ **Moments dus aux forces de traînée :**

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

$$M_z = d (\omega_1^2 - \omega_2^2 + \omega_3^2 - \omega_4^2) \quad \text{Eq.I.5.22}$$

Avec : d est le coefficient de traînée.

→ Moment résultant des frottements aérodynamiques : il est donné par

$$M_f = K_f \Omega^2 \quad \text{Eq.I.5.23}$$

Avec :

K_f : Le coefficient des frottements aérodynamiques et Ω est la vitesse angulaire.

❖ **Le moment gyroscopique :**

L'effet gyroscopique se définit comme la difficulté de modifier la position ou l'orientation du plan de rotation d'une masse tournante. L'effet gyroscopique est ainsi nommé en référence au mode de fonctionnement du gyroscope, appareil de contrôle de mouvement utilisé dans l'aviation. [1]. Dans le cas du quadri-rotor il y a deux moments gyroscopiques présenter comme suite :

→ Moment gyroscopique des hélices : donné par la relation suivante :

$$M_{gh} = \sum_{i=1}^4 \Omega \wedge J_r \begin{bmatrix} 0 \\ 0 \\ (-1)^{i+1} \omega_i \end{bmatrix} \quad \text{Eq.I.5.24}$$

Avec : J_r est l'inertie des rotors qui est supposée diagonale.

$$J_r = \begin{bmatrix} J_{rx} & 0 & 0 \\ 0 & J_{ry} & 0 \\ 0 & 0 & J_{rz} \end{bmatrix} \quad \text{Eq.I.5.25}$$

→ Moment gyroscopique dû aux mouvements de quadri-rotor : donné par la relation suivante :

$$M_{gm} = \Omega \wedge J \Omega \quad \text{Eq.I.5.26}$$

Avec : J est l'inertie du système.

→ Les moments expliqués précédemment se résument ensemble pour trouver la somme des moments extérieurs totaux agissant sur le quadri-rotor.

En remplaçant dans **Eq.I.5.17** :

$$J\dot{\Omega} + \Omega \wedge J\Omega = \begin{bmatrix} M_x \\ M_y \\ M_z \end{bmatrix} - M_{gh} - M_{gm} \quad \text{Eq.I.5.27}$$

I.5.3. Développement du Modèle mathématique selon Newton-Euler :[1]

En utilisant la formulation de Newton-Euler, les équations sont écrites sous la forme suivante:

$$\begin{cases} \dot{X} = V \\ m\dot{V} = P + F_p + F_t + F_{\text{perturbation}} \\ \dot{R} = RS(\Omega) \\ J\dot{\Omega} = -\Omega \wedge J\Omega + M_a - M_{gh} - M_{gm} \end{cases} \quad \text{Eq.I.5.28}$$

Avec:

X : est le vecteur de position du quadri-rotor.

m : la masse totale du quadri-rotor.

Ω : La vitesse angulaire exprimée dans le repère fixe.

R : La matrice de rotation.

\wedge : Le produit vectoriel.

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

$S(\Omega)$: est la matrice antisymétrique pour un vecteur de vélocité: $\Omega = \begin{bmatrix} \Omega_1 \\ \Omega_2 \\ \Omega_3 \end{bmatrix}$ donnée par :

$$S(\Omega) = \begin{bmatrix} 0 & -\Omega_3 & \Omega_2 \\ \Omega_3 & 0 & -\Omega_1 \\ -\Omega_2 & \Omega_1 & 0 \end{bmatrix} \quad \text{Eq.I.5.29}$$

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

$$F_p = R \times \begin{bmatrix} 0 \\ 0 \\ \sum_{i=1}^4 b\omega_i^2 \end{bmatrix} \quad \text{Eq.I.5.30}$$

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

$$F_t = \begin{bmatrix} -K_{ftx} & 0 & 0 \\ 0 & -K_{fty} & 0 \\ 0 & 0 & -K_{ftz} \end{bmatrix} \dot{X} \quad \text{Eq.I.5.31}$$

K_{ftx} , K_{fty} , K_{ftz} : Les coefficients de traînée de translation.

P : force de gravité, elle est donnée par :

$$P = \begin{bmatrix} 0 \\ 0 \\ -mg \end{bmatrix} \quad \text{Eq.I.5.32}$$

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

$$M_a = \begin{bmatrix} lb(\omega_4^2 - \omega_2^2) \\ lb(\omega_3^2 - \omega_1^2) \\ d(\omega_1^2 - \omega_2^2 + \omega_3^2 - \omega_4^2) \end{bmatrix} \quad \text{Eq.I.5.33}$$

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

$$M_f = \begin{bmatrix} K_{fx}\Omega_1 \\ K_{fy}\Omega_2 \\ K_{fz}\Omega_3 \end{bmatrix} \quad \text{Eq.I.5.34}$$

Avec : K_{fx} , K_{fy} , K_{fz} Les coefficients des frottements aérodynamiques.

I.5.3.1. Equations de mouvement de translation :

En négligeant la force de perturbation on aura :

$$\mathbf{m}\dot{\mathbf{V}} = \mathbf{P} + \mathbf{F}_p + \mathbf{F}_t \quad \text{Eq.I.5.35}$$

On remplace chaque force par sa formule, on trouve :

$$m \begin{bmatrix} \ddot{x} \\ \ddot{y} \\ \ddot{z} \end{bmatrix} = \begin{bmatrix} C\phi S\theta C\psi + S\psi S\phi \\ C\phi S\theta S\psi - C\psi S\phi \\ C\phi C\theta \end{bmatrix} \sum_{i=1}^4 \mathbf{F}_i - \begin{bmatrix} K_{ftx}\dot{x} \\ K_{fity}\dot{y} \\ K_{ftz}\dot{z} \end{bmatrix} - \begin{bmatrix} \mathbf{0} \\ \mathbf{0} \\ mg \end{bmatrix} \quad \text{Eq. I.5.36}$$

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

$$\begin{cases} \ddot{x} = \frac{1}{m} (C\phi S\theta C\psi + S\psi S\phi) \sum_{i=1}^4 \mathbf{F}_i - \frac{K_{ftx}}{m} \dot{x} \\ \ddot{y} = \frac{1}{m} (C\phi S\theta S\psi - C\psi S\phi) \sum_{i=1}^4 \mathbf{F}_i - \frac{K_{fity}}{m} \dot{y} \\ \ddot{z} = \frac{1}{m} (C\phi C\theta) \sum_{i=1}^4 \mathbf{F}_i - \frac{K_{ftz}}{m} \dot{z} - g \end{cases} \quad \text{Eq.I.5.37}$$

I.5.3.2. Equations de mouvement de rotation :

On a:

$$\mathbf{J}\dot{\boldsymbol{\Omega}} = -\mathbf{M}_{gh} + \mathbf{M}_a - \mathbf{M}_{gh} - \mathbf{M}_{gm} \quad \text{Eq.I.5.38}$$

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

$$\begin{bmatrix} I_x & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & I_y & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & I_z \end{bmatrix} \begin{bmatrix} \ddot{\phi} \\ \ddot{\theta} \\ \ddot{\psi} \end{bmatrix} = - \begin{bmatrix} \dot{\phi} \\ \dot{\theta} \\ \dot{\psi} \end{bmatrix} \wedge \begin{bmatrix} I_x & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & I_y & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & I_z \end{bmatrix} \begin{bmatrix} \dot{\phi} \\ \dot{\theta} \\ \dot{\psi} \end{bmatrix} - \begin{bmatrix} J_r \overline{\Omega_r} \dot{\theta} \\ J_r \overline{\Omega_r} \dot{\phi} \\ \mathbf{0} \end{bmatrix} - \begin{bmatrix} K_{fx}\dot{\phi}^2 \\ K_{fy}\dot{\theta}^2 \\ K_{fz}\dot{\psi}^2 \end{bmatrix} + \begin{bmatrix} lb(\omega_4^2 - \omega_2^2) \\ lb(\omega_3^2 - \omega_1^2) \\ d(\omega_1^2 - \omega_2^2 + \omega_3^2 - \omega_4^2) \end{bmatrix} \quad \text{Eq.I.5.39}$$

On obtient alors les équations différentielles définissant le mouvement de rotation :

$$\begin{cases} I_x \ddot{\phi} = -\dot{\phi}\dot{\psi}(I_z - I_y) - J_r \overline{\Omega_r} \dot{\theta} - K_{fx}\dot{\phi}^2 + lb(\omega_4^2 - \omega_2^2) \\ I_y \ddot{\theta} = -\dot{\phi}\dot{\psi}(I_z - I_x) - J_r \overline{\Omega_r} \dot{\phi} - K_{fy}\dot{\theta}^2 + lb(\omega_3^2 - \omega_1^2) \\ I_z \ddot{\psi} = -\dot{\phi}\dot{\psi}(I_x - I_y) - K_{fz}\dot{\psi}^2 + d(\omega_1^2 - \omega_2^2 + \omega_3^2 - \omega_4^2) \end{cases} \quad \text{Eq.I.5.40}$$

Avec:

$$\overline{\Omega}_r = \omega_1 - \omega_2 + \omega_3 - \omega_4 \quad \text{Eq.I.5.41}$$

En conséquence, le modèle dynamique complet qui régit le quadri-rotor est le suivant :

$$\left\{ \begin{array}{l} \ddot{\phi} = -\dot{\phi}\dot{\psi} \frac{(I_z - I_y)}{I_x} - \frac{J_r}{I_x} \overline{\Omega}_r \dot{\theta} - \frac{K_{fx}}{I_x} \dot{\phi}^2 + \frac{l}{I_x} \mathbf{u}_2 \\ \ddot{\theta} = -\dot{\phi}\dot{\psi} \left(\frac{I_z - I_x}{I_y} \right) - \frac{J_r}{I_y} \overline{\Omega}_r \dot{\phi} - \frac{K_{fy}}{I_y} \dot{\theta}^2 + \frac{l}{I_y} \mathbf{u}_3 \\ \ddot{\psi} = -\dot{\phi}\dot{\psi} \left(\frac{I_x - I_y}{I_z} \right) - \frac{K_{fz}}{I_z} \dot{\psi}^2 + \frac{1}{I_z} \mathbf{u}_4 \\ \ddot{x} = -\frac{K_{ftx}}{m} \dot{x} + \frac{1}{m} \mathbf{u}_x \mathbf{u}_1 \\ \ddot{y} = -\frac{K_{fty}}{m} \dot{y} + \frac{1}{m} \mathbf{u}_y \mathbf{u}_1 \\ \ddot{z} = -\frac{K_{ftz}}{m} \dot{z} - g + \frac{\cos(\phi)\cos(\theta)}{m} \mathbf{u}_1 \end{array} \right. \quad \text{Eq.I.5.42}$$

Avec:

$$\begin{cases} \mathbf{u}_x = C\phi S\theta C\psi + S\psi S\phi \\ \mathbf{u}_y = C\phi S\theta S\psi - C\psi S\phi \end{cases} \quad \text{Eq.I.5.43}$$

Et:

$$\begin{bmatrix} \mathbf{u}_1 \\ \mathbf{u}_2 \\ \mathbf{u}_3 \\ \mathbf{u}_4 \end{bmatrix} = \begin{bmatrix} b & b & b & b \\ 0 & -lb & 0 & lb \\ -lb & 0 & lb & 0 \\ d & -d & d & -d \end{bmatrix} \begin{bmatrix} \omega_1 \\ \omega_2 \\ \omega_3 \\ \omega_4 \end{bmatrix} \quad \text{Eq.I.5.44}$$

I.5.3.3. Liaisons non holonomes :

Les liaisons non holonomes sont des relations liant les deux angles ϕ et θ avec ψ et les composantes du vecteur d'accélération. Ces relations sont obtenues en manipulant les trois équations du modèle dynamique dans l'Eq.I.5.42 (En négligeant les effets gyroscopiques en x, y, z et en remplaçant les relations de \mathbf{u}_x et \mathbf{u}_y dans Eq.I.5.43).

$$\ddot{x} = \frac{u_1}{m} (C\phi S\theta C\psi + S\psi S\phi) \quad \text{Eq.I.5.45}$$

$$\ddot{y} = \frac{u_1}{m} (C\phi S\theta S\psi - C\psi S\phi) \quad \text{Eq.I.5.46}$$

$$\ddot{z} = \frac{C\phi C\theta}{m} u_1 - g \quad \text{Eq.I.5.47}$$

$$(\text{Eq.I.5.45}) C\psi + (\text{Eq.I.5.45}) S\psi \Rightarrow C\phi S\theta = \frac{m}{u_1} (\ddot{x}C\psi + \ddot{y}S\psi) \quad \text{Eq.I.5.48}$$

$$(\text{Eq.I.5.45}) S\psi + (\text{Eq.I.5.45}) C\psi \Rightarrow C\phi S\theta = \frac{m}{u_1} (-\ddot{x}S\psi + \ddot{y}C\psi) \quad \text{Eq.I.5.49}$$

$$\text{Eq.I.5.47} \Rightarrow C\phi C\theta = \frac{m}{u_1} (\ddot{z} + g) \quad \text{Eq.I.5.50}$$

$$\text{Eq.I.5.48/ Eq.I.5.50} \Rightarrow \tan\theta = \frac{\ddot{x}C\psi + \ddot{y}S\psi}{\ddot{z} + g} \quad \text{Eq.I.5.51}$$

$$\text{Eq.I.5.49} \Rightarrow \sin\phi = \frac{m}{u_1} (\ddot{x}S\psi - \ddot{y}C\psi) \quad \text{Eq.I.5.52}$$

Or:

$$\ddot{x}^2 + \ddot{y}^2 + (\ddot{z} + g)^2 = \frac{u_1^2}{m^2} \quad \text{Eq.I.5.53}$$

Donc:

$$\frac{m}{u_1} = \frac{1}{\sqrt{\ddot{x}^2 + \ddot{y}^2 + (\ddot{z} + g)^2}} \quad \text{Eq.I.5.54}$$

En remplaçant dans l'Eq.I.5.53 on obtient :

$$\sin\phi = \frac{\ddot{x}S\psi - \ddot{y}C\psi}{\sqrt{\ddot{x}^2 + \ddot{y}^2 + (\ddot{z} + g)^2}} \quad \text{Eq.I.5.55}$$

Les équations Eq.I.5.51 et Eq.I.5.55 sont appelées liaisons non holonomes.

On va appliquer cette présentation avec une autre sur les équations des mouvements de translation dans le paragraphe suivant afin d'énoncer le principe des Quaternions pour éliminer les singularités des sinus et cosinus figurent dans la matrice de passage entre les vitesses angulaire et les vitesses angulaire d'Euler « Eq.I.5.8 ».

I.6. MODILISATION AVEC LES QUATERNIONS :

L'utilisation des angles d'Euler (lacet, tangage, roulis) est devenue très répandue dans le domaine aérospatial. Cependant, elles souffrent des problèmes de singularités connues par le « gimballock ». Ce problème se produit lorsque l'angle de roulis perd sa signification. En d'autre terme, si l'appareille fait un tangage de 90° , son axe de roulis devient parallèle à l'axe de lacet, et il n'y a aucun axe disponible pour adapter la rotation de lacet (un degré de liberté est perdu). Pour éviter ce problème on utilise les quaternions [2].

Une vue d'ensemble sur les quaternions est décrite dans l'ANNEXE 3.

I.6.1. Modèle dynamique de translation :

La détermination de modèle dynamique de translation avec les quaternions ne diffère pas à celle de Newton-Euler. Seulement, on doit représenter la force de portance dans le repère en R_n utilisant la rotation en quaternion. [2]

On a:

$$F_{p0} = q \times F_{portance} \times \dot{q} \quad \text{Eq.I.5.56}$$

Après le développement de cette équation, on obtient :

$$\Sigma F_{exterieur} = \begin{bmatrix} (2q_1q_3 + 2q_2q_4) \Sigma_1^4 F_i - K_x \dot{x}^2 \\ (2q_3q_4 - 2q_1q_2) \Sigma_1^4 F_i - K_y \dot{y}^2 \\ (q_1^2 - q_2^2 - q_3^2 + q_4^2) \Sigma_1^4 F_i - K_z \dot{z}^2 - mg \end{bmatrix} \quad \text{Eq.I.5.57}$$

D'où :

$$\begin{bmatrix} \ddot{x} = \frac{1}{m} ((2q_1q_3 + 2q_2q_4) \Sigma_1^4 F_i - K_x \dot{x}^2) \\ \ddot{y} = \frac{1}{m} ((2q_3q_4 - 2q_1q_2) \Sigma_1^4 F_i - K_y \dot{y}^2) \\ \ddot{z} = \frac{1}{m} ((q_1^2 - q_2^2 - q_3^2 + q_4^2) \Sigma_1^4 F_i - K_z \dot{z}^2 - mg) \end{bmatrix} \quad \text{Eq.I.5.58}$$

Avec :

$$K_{x,y,z} = \frac{1}{2} \rho \cdot S_w \cdot C_d \quad \text{Eq.I.5.59}$$

I.6.2. Modèle dynamique de rotation :

Le modèle de rotation ne change pas car il est établi dans le repère lié au quadri-rotor (Pas de passage entre les repères). [2]

$$\begin{bmatrix} \dot{p} \\ \dot{q} \\ \dot{r} \end{bmatrix} = \begin{bmatrix} 1 & 0 & -S\theta \\ 0 & C\phi & S\phi C\phi \\ 0 & -S\phi & C\phi C\phi \end{bmatrix} \begin{bmatrix} \dot{\phi} \\ \dot{\theta} \\ \dot{\psi} \end{bmatrix} \quad \text{Eq.I.5.60}$$

On aura:

$$\begin{cases} \dot{p} = -\frac{(I_z - I_y)}{I_x} qr - \frac{J_{rz}}{I_x} \overline{\Omega_r} q + \frac{M_x}{I_x} \\ \dot{q} = -\frac{(I_z - I_x)}{I_y} pr - \frac{J_{rz}}{I_y} \overline{\Omega_r} p + \frac{M_y}{I_y} \\ \dot{r} = -\frac{(I_x - I_y)}{I_z} pq + \frac{M_z}{I_z} \end{cases} \quad \text{Eq.I.5.61}$$

I.6.3. Relation entre les quaternions et les vitesses angulaires :

Pour établir la relation entre les quaternions et les vitesses angulaires, on utilise la relation suivante : [2]

$$\dot{q} = \frac{1}{2} q \times \Omega \quad \text{Eq. I.5.62}$$

En développant cette équation, on obtient le résultat final suivant :

$$\begin{cases} \dot{q}_1 = \frac{-1}{2} (q_2 p + q_3 q + q_4 r) \\ \dot{q}_2 = \frac{1}{2} (q_1 p + q_3 r + q_4 q) \\ \dot{q}_3 = \frac{1}{2} (q_1 q + q_4 p + q_2 r) \\ \dot{q}_4 = \frac{1}{2} (q_1 r + q_2 q + q_3 p) \end{cases} \quad \text{Eq. I.5.63}$$

I.6.4. Avantages d'utilisation des quaternions :

L'utilisation des quaternions offre plusieurs avantages : [2]

- Pas de singularité.
- Pas de fonction trigonométrique.
- Représentation facile pour des rotations successives.
- Simplification de calcul.

Dans notre programme de modélisation on a utilisé le block quaternion de SIMULINK présenté en détail dans le chapitre suivant.

I.7.LA DYNAMIQUE DES MOTEURS :

Un circuit équivalent au moteur à courant continu qui a un modèle bien connu qui relie des quantités électriques et mécaniques. Ce modèle est composé de la série d'une résistance R [Ohm], Une inductance L [H] et un générateur de tension e [V]. La résistance représente les pertes de Joule due Au flux de courant dans le conducteur de cuivre. Sa valeur dépend de la géométrie Et les caractéristiques matérielles telles que la résistivité du fil, la longueur et la section. Le comportement de l'inducteur dérive de la forme des fils du moteur qui sont enroulés Au milieu du rotor. Enfin, le générateur e fournit une Tension appelée la force électromotrice (FEM) de retour, proportionnelle à la vitesse du moteur. Le modèle est représenté dans la figure I.7.1 et la figure I.7.2 [6]

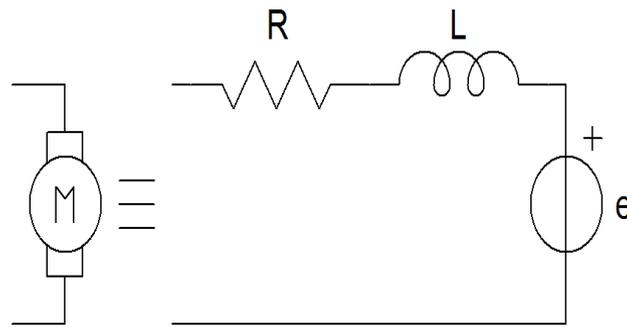


Figure I.7.1: Modèle électrique équivalent du moteur DC. [6]

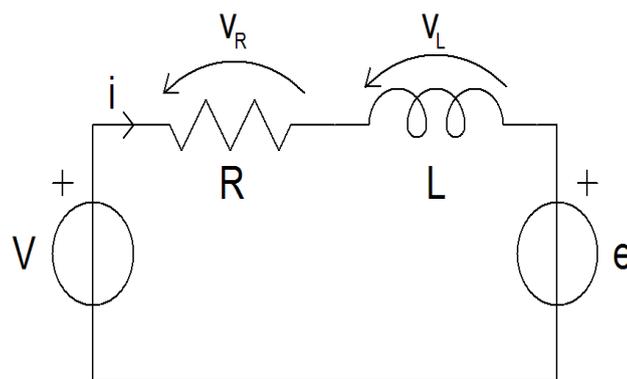


Figure I.7.2: Circuit équivalent du moteur DC. [6]

En appliquant la loi de tension de Kirchhoff, l'équation suivante est obtenue :

$$V_s = Ri + L \frac{di}{dt} + e \quad \text{Eq I.7.1}$$

À l'état d'équilibre (état dont le courant continu est de fréquence zéro)

$$V_s = Ri + e \quad \text{Eq I.7.2}$$

Par conséquent, pour l'état non stationnaire, **Eq.I.34** est réarrangée pour avoir la fem, comme est indiqué dans l'équation ci-dessous: [7]

$$e = -Ri - L \frac{di}{dt} + V_s \quad \text{Eq I.7.3}$$

Avec :

V_s : Source de tension.
 i : courant d'armature.

De même, on tenant compte des propriétés mécaniques du moteur à courant continu, et à partir de la deuxième loi de Newton sur le mouvement :

$$J \frac{d\omega_m}{dt} = \sum T_i \quad \text{Eq I.7.4}$$

$$T_e = k_f \omega_m + J \frac{d\omega_m}{dt} \quad \text{Eq I.7.5}$$

Où

T_e : Le couple électrique.
 K_f : La constante de friction.
 J : Inertie du rotor.
 ω_m : La vitesse angulaire.

Le couple électrique et la f_{em} pourraient être écrits comme : [7]

$$e = k_e \omega_m \quad \text{Et} \quad T_e = k_f \omega_m \quad \text{Eq I.7.6}$$

Avec :

K_e : La constante de la f_{em} de retour.
 K_t : La constante du couple électrique.

En remplaçant dans **Eq I.7.3** et **Eq I.7.4** on trouve les équations suivantes :

$$\frac{di}{dt} = -i \frac{R}{L} - \frac{k_e}{L} \omega_m + \frac{1}{L} V_s \quad \text{Eq I.7.7}$$

$$\frac{d\omega_m}{dt} = i \frac{k_t}{J} - \frac{k_f}{L} \omega_m \quad \text{Eq I.7.8}$$

Dans la suite de ce travail, on fait appel à la transformée de Laplace pour évaluer les deux équations **Eq.I.7.7** et **Eq.I.7.8** Et en ce qui suit, les expressions abouties sont obtenus de façon appropriée (toutes les conditions initiales sont supposées être zéro). [7]

En appliquant la transformer de Laplace à l'**Eq I.7.7** et l'**Eq I.7.8** :

$$s i = -i \frac{R}{L} - \frac{k_e}{L} \omega_m + \frac{1}{L} V_s \quad \text{Eq I.7.9}$$

$$s \omega_m = i \frac{k_t}{J} - \frac{k_f}{L} \omega_m \quad \text{Eq I.7.10}$$

De l'**Eq I.7.10** on aura l'équation suivante :

$$i = \frac{s \omega_m + \frac{k_f}{L} \omega_m}{\frac{k_t}{J}} \quad \text{Eq I.7.11}$$

On remplace le i dans l'**Eq I.7.9** avec le la relation trouver dans l'**Eq I.7.11** on trouve :

$$\left(\frac{s \omega_m + \frac{k_f}{L} \omega_m}{\frac{k_t}{J}} \right) \left(s + \frac{R}{L} \right) = -\frac{k_e}{L} \omega_m + \frac{1}{L} V_s \quad \text{Eq I.7.12}$$

Finalement, **Eq.I.7.12** devient :

$$V_s = \left(\frac{s^2 J L + s k_f L + s R J + k_f R + k_e k_t}{k_t} \right) \omega_m \quad \text{Eq I.7.13}$$

La fonction de transfert est donc obtenue en utilisant le rapport entre la vitesse angulaire et tension de source.

$$G(s) = \frac{\omega_m}{V_s} = \left(\frac{k_t}{s^2 J L + s k_f L + s R J + k_f R + k_e k_t} \right) \quad \text{Eq I.7.14}$$

Considérant les hypothèses suivantes : [7]

- La constante de frottement k_f est faible et tend vers zéro, cela implique

$$\rightarrow R J \gg k_f L$$

$$\rightarrow k_e k_t \gg k_f R$$

La fonction de transfert peut être écrite :

$$G(s) = \left(\frac{\frac{1}{k_e}}{\frac{RJ}{k_e k_t R} s^2 + \frac{RJ}{k_e k_t} s + 1} \right) \quad \text{Eq I.7.15}$$

De l'Eq I.7.15, les constantes de temps suivantes sont acquises :

→ La constante de temps magnétique :

$$\tau_m = \frac{RJ}{k_e k_t} \quad \text{Eq I.7.16}$$

→ La constante de temps électrique :

$$\tau_e = \frac{L}{R} \quad \text{Eq I.7.17}$$

Finalement en remplaçant les constant de temps dans l'Eq I.7.15, on obtient :

$$G(s) = \left(\frac{\frac{1}{k_e}}{\tau_m \tau_e s^2 + \tau_m s + 1} \right) \quad \text{Eq I.7.18}$$

L'Eq I.7.18 représente la fonction de Transfer entre l'entrée du moteur qui est la fem et sa sortie qui représente la vitesse. Dans le chapitre suivant on va utiliser cette équations plus d'autres, afin de construire le model SIMULINK du quadri-rotor.

I.8.CONCLUSION :

L'utilisation du formalisme de Newton-Euler, nous a permet d'établir le modèle dynamique du quadri-rotor. Apres on a utilisé les quaternions afin d'éliminer les singularités dans la matrice de passage inverse entre les vitesses angulaires d'Euler et les vitesses angulaires (ce problème représente le problème d'utilisation du formalisme de Newton-Euler). A partir du modèle obtenu, nous concluons que le quadri-rotor est un système fortement couplé. De plus, la complexité du modèle et la non-linéarité, peuvent se voir clairement.

Dans les sections suivantes, on va faire des simulations en boucle ouverte sur le modèle du quadri-rotor afin de mètre en évidence l'instabilité du système en boucle ouverte et la nécessité d'un régulateur performant en boucle fermée qui est basé sur une estimation précise d'état [1].

III.1.Introduction :

Les UAVs et les aéronefs autonomes de n'importe quelle famille doivent être capables de stabiliser leurs attitudes et leurs positions avant qu'une commande de vol n'intervienne. La stabilisation d'attitude et de la position est la partie la plus importante dans la commande des robots volants qui sont de nature instable, et les rendre manœuvrables reste toujours un défi, vu la complexité des états dépendants et la non linéarité du model.

La configuration à quatre rotors a attiré l'attention des chercheurs en aéronautique, robotique et automatique. Ainsi la recherche des lois de commande simple, robuste et facilement à embarqué reste aussi un domaine très vaste. Plusieurs laboratoires travaillent sur la commande des drones et plus particulièrement les quadri-rotors et plusieurs lois de commande ont été proposées, parmi lesquelles je vais travailler sur la commande PID en utilisant la méthode du loop-shaping.

Dans ce chapitre nous allons tout d'abord présenter les blocs du programme SIMILINK qui constitue le quadri-rotor avec les méthodes de commande utilisés, ensuite étudier le comportement du quadri-rotors en boucle ouverte, et présenter la synthèse des lois de commande, et enfin donner les résultats de simulation associés.

II.2.MODELE SIMULINK DU QUADRI-ROTOR :

Dans le cadre de ce projet, le modèle dynamique du quadri-rotor a été construit sous le logiciel Matlab/Simulink, où on a utilisé une nouvelle méthode de modélisation qui est élaborée à base des équations de la physique et pas sur la représentation d'état du quadri-rotor. On a fait appelle à quelques blocks de Mathwork / Matlab et on a construit des fonctions des différents blocks afin de modéliser chaque composant du quadri-rotor.

Notre model se compose de :

- **Le block « multicopter »** : qui regroupe les équations mathématiques reliées à la dynamique de rotation et de translation, il a comme rentrées les vitesses angulaires et linéaires du quadri-rotor dans le repère mobile, la densité de l'air ' ρ ' produit par le block sur SIMULINK « ISA Atmosphere Model » qui a comme entrée le ' Z ' du système, la matrice de transfère du repère fixe au repère mobile et enfin la somme des vitesses des 4 rotors, Et comme sortie la somme des moments et les forces.
- **Le block « brushless motors »** : qui comporte le modèle mathématique de la dynamique des rotors, il a une entrée PWM produite par la régulation et une autre entrée qui est celle de l'alimentation en volts.
- **Le block « Quaternions »** : une fois les forces et les moments sont calculés (par le block multicopter), Matlab/Simulink comporte un block qui modélise la dynamique du corps rigide à 6 DDL de masse fixe avec la méthode des quaternions. Il a comme entrées les forces et les moments et il produit comme sorties : les vitesses linéaire et les positions reliées au repère fixe, les angles d'Euler, la matrice de passage entre le repère fixe et mobile et enfin les vitesses et les accélérations linéaires et angulaires reliées au repère mobile.
- **Le block « LiPo »** : la diminution de puissance délivrée aux moteurs est prise en considération à l'aide d'un block qui modélise la batterie « LiPo » (voir **Annexe.I**), il a une entrée PWM et deux sorties qui sont : la tension délivrée aux moteurs et le voltage de chaque cellule.

On a aussi modélisé deux types de capteur l'ultrasonique et la centrale inertielle (gyroscope et l'accéléromètre) afin de produire les entrées comme la densité et les sortie comme les accélérations ... (voir ANNEXE1)

La **Figure. II.1** illustre le modèle développé.

Tableau II.1 : contient les paramètres des moteurs et de la batterie LIPO utilisés dans notre modélisation des blocks

Tableau II.2 : contient les paramètres du quadri-rotor choisit présenté dans l'annexe 1

Les paramètres du modèle du quadri-rotor réel utilisé sont :

❖ Les valeurs des paramètres de la batterie et des moteurs :

Paramètre	Valeur
I_{max}	5.5 (A)
$I_{des\ composants\ auxiliaires}$	0.1
Nombre de cellules	2
V	10 V (chaque cellule a un $v_{max} = 5\ V$)
Capacité	1 (A.h)
τ_e	$2.5\ 10^{-6}$
τ_m	0.01
K_e	$3.08\ 10^{-3}$

Tableau II.1 : Paramètre des moteurs et de la batterie LIPO [annexe 1]

❖ Les valeurs des paramètres du quadri-rotor :

Paramètre	Valeur
m	0.26 (kg)
l	0.25 (mètres)
Diamètre des hélices	0.1397 (mètres)
g	9.81 (m/s^2)
Coefficient de traînée « d »	15
Coefficient de portance « b »	10^{-3}
I_x	0.00080 ($kg.m^2$)
I_y	0.00065 ($kg.m^2$)
I_z	0.0014 ($kg.m^2$)
J_r	0.0000025 ($kg.m^2$)

Tableau II.2 : Paramètre du quadri-rotor [annexe 1]

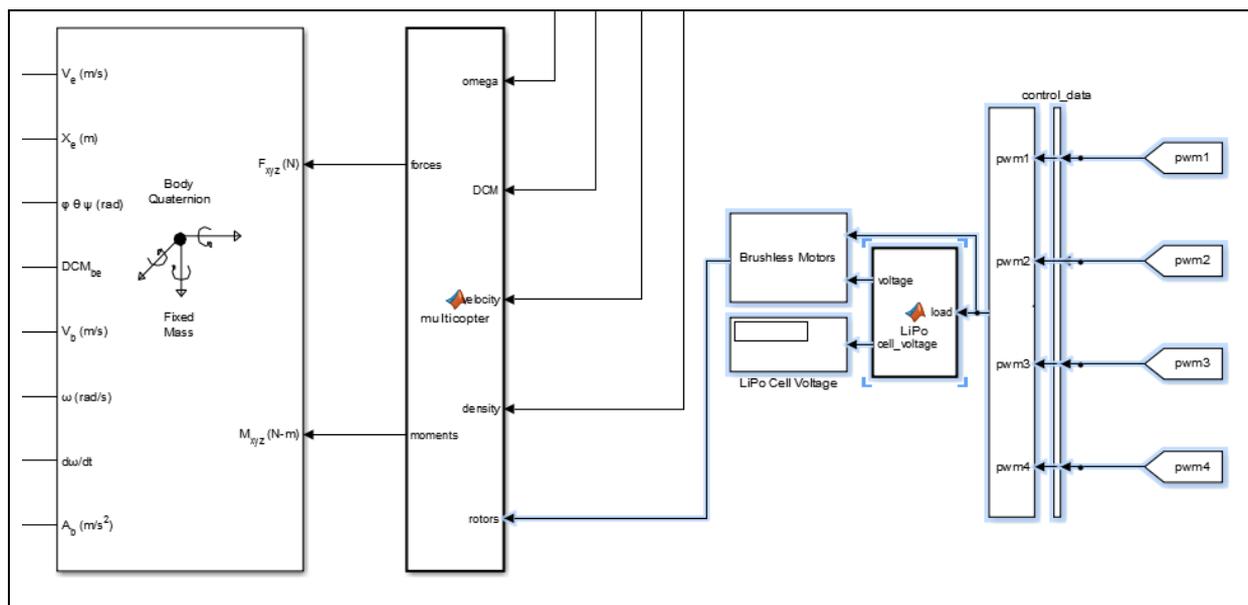


Figure.II.1 : Modèle SIMULINK du quadri-rotor.

II.2.1.Simulation en boucle ouverte :

Dans cette partie on va faire une simulation en boucle ouverte sur le quadri-rotor, en injectant directement aux entrées du système les vitesses de rotation. Pour illustrer le comportement dynamique du système et l'effet des vitesses de rotations sur les mouvements du quadri-rotor, on prend les quatre cas suivants :

1^{ier} cas : pour $\omega_1^2 = \omega_2^2 = \omega_3^2 = \omega_4^2 = 670 \text{ (tr/sec)}^2$

2^{ème} cas : pour $\omega_1^2 = \omega_3^2 = 556 \text{ (tr/sec)}^2$ et $\omega_2^2 = \omega_4^2 = 819 \text{ (tr/sec)}^2$

3^{ème} cas : pour $\omega_2^2 = \omega_3^2 = 662 \text{ (tr/sec)}^2$ et $\omega_1^2 = \omega_4^2 = 702 \text{ (tr/sec)}^2$

4^{ème} cas : pour $\omega_1^2 = \omega_2^2 = 649 \text{ (tr/sec)}^2$ et $\omega_3^2 = \omega_4^2 = 714 \text{ (tr/sec)}^2$

Les paramètres du modèle du quadri-rotor réel utilisé (dji F450) sont obtenus à partir d'un modèle développé sous le logiciel « SolidWorks » (voir Annexe 1)

Avec : U_1, U_2, U_3 et U_4 sont les entrées de commande ou :

$$\begin{cases} M_x = dU_2 \\ M_y = dU_3 \\ M_z = dU_4 \\ F_p = U_1 \end{cases} \quad \text{Eq. II.1}$$

$$\begin{bmatrix} u_1 \\ u_2 \\ u_3 \\ u_4 \end{bmatrix} = \begin{bmatrix} b & b & b & b \\ 0 & -lb & 0 & lb \\ -lb & 0 & lb & 0 \\ d & -d & d & -d \end{bmatrix} \begin{bmatrix} \omega_1 \\ \omega_2 \\ \omega_3 \\ \omega_4 \end{bmatrix} \quad \text{Eq.II.2}$$

II.2.2.Résultats de simulation :

→ **1^{ier} cas :**

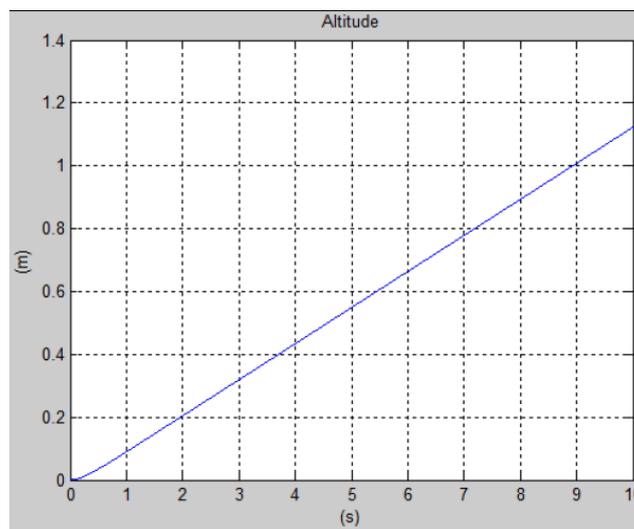


Figure II.2 : Mouvement vertical selon z.

On remarque que le quadri-rotor fait une translation verticale due aux forces de portance créées par la commande $U1$ (On remarque aussi, que la translation suivant l'axe x et y est nulle car la rotation autour de ces deux axes est nulle. Donc on peut dire que la translation selon l'axe z est provoquée par l'action $U1$).

→ 2^{ème} cas :

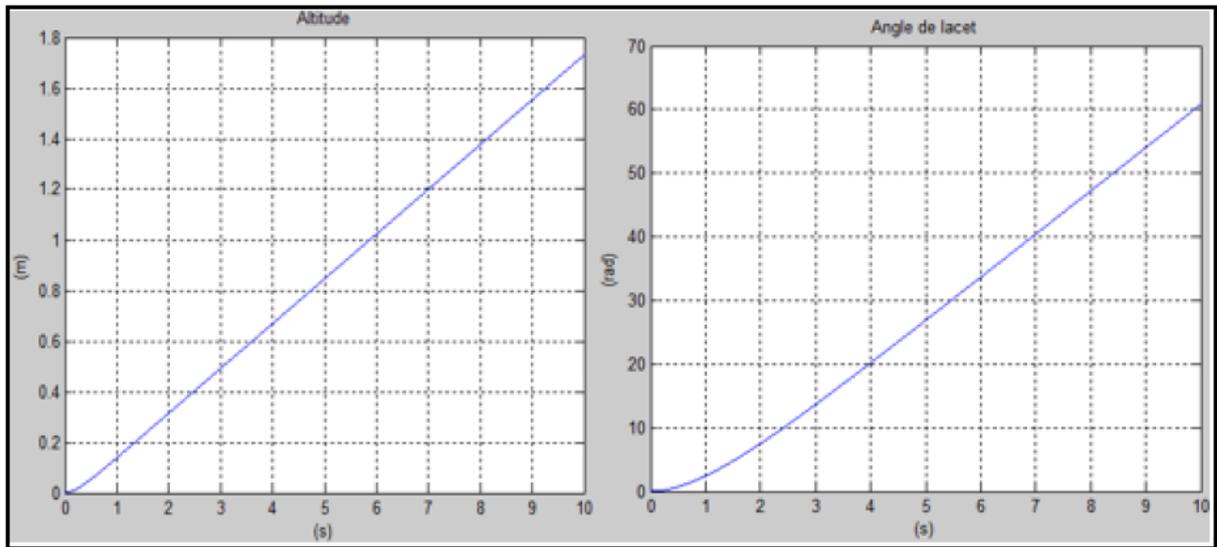


Figure II.3 : Mouvement du lacet.

Selon la **Fig.II.3**, on peut remarquer que le quadri-rotor fait une rotation autour de l'axe z (lacet), ce mouvement est provoqué par la variation des vitesses de rotation des rotors avant et arrière par rapport aux rotors latéraux. Alors la rotation autour de l'axe z est due à l'action $U4$.

→ 3^{ème} cas :

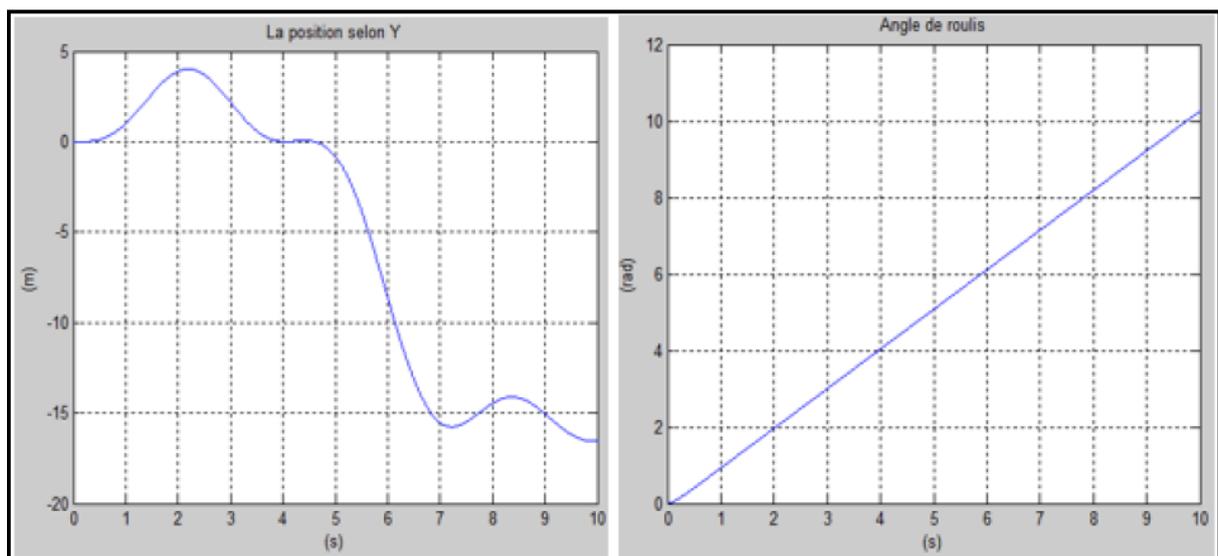


Figure II.4 : Mouvement de roulis.

D'après la **Fig.II.4**, le quadri-rotor fait une rotation autour de l'axe y (tangage), cette rotation provoque une translation selon x . il existe aussi un mouvement de rotation autour de l'axe x dû aux différents effets agissants sur le quadri-rotor et une translation verticale parce que la force de portance est supérieure au poids du système. Donc la translation selon l'axe x est due à la rotation autour de l'axe y qui est provoquée par l'action U_3 .

→ 4^{ème} Cas :

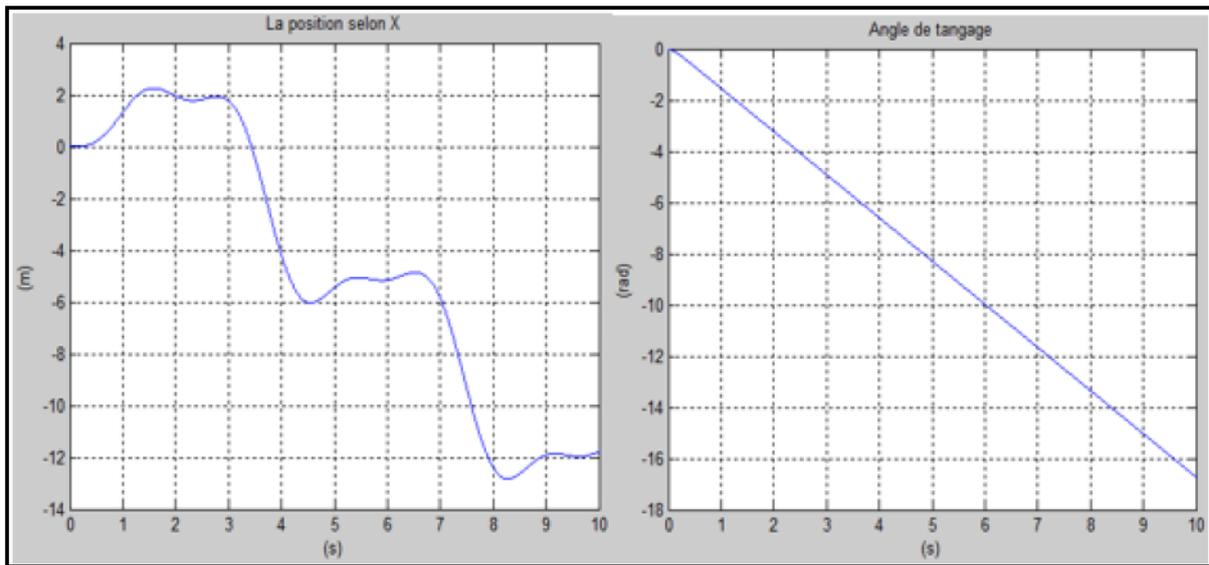


Figure II.5 : Mouvement de tangage

Selon la **Fig.II.5**, on remarque que le quadri-rotor effectue une rotation autour de l'axe x (roulis), ce mouvement donne une translation selon l'axe y . On a aussi un mouvement de rotation autour de l'axe y et une translation verticale. Donc la translation selon l'axe y est due à la rotation autour de l'axe x qui est provoquée par l'action U_2 .

La **Fig.II.6** montre la variation d'altitude dans le cas des mouvements de tangage, roulis et lacet.

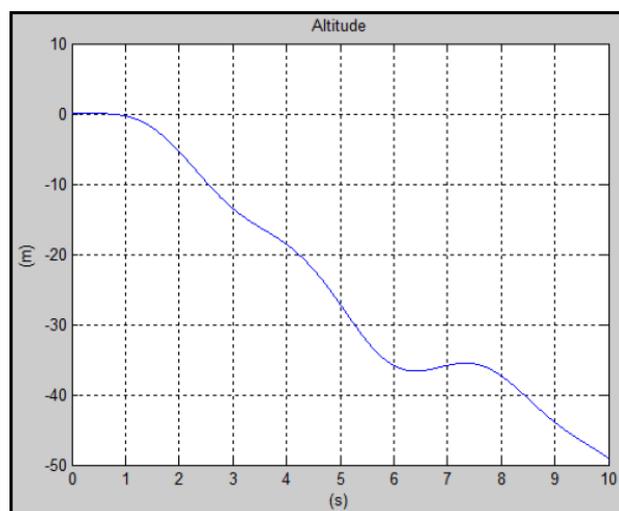


Figure II.6 : Variation d'altitude dans le cas des mouvements de tangage, roulis et lacet.

II.3. CAHIER DE CHARGES :

L'objectif de l'étude est de déterminer les paramètres de réglage des commandes permettant de satisfaire les contraintes du cahier des charges suivantes :

- 1- **Rapidité** : temps de réponse inférieur à 5s.
- 2- **Amortissement** : pas de dépassement de l'asymptote.
- 3- **Précision** : erreur indicielle nulle.
- 4- **Consigne** on prend les références x_r, y_r, z_r égaux à '1' pour avoir l'angle entre x et z de référence ϕ_d et θ_d sont définis à partir du principe non holonome ϕ_d égaux à ' $\frac{\pi}{4}$ ', suivant le schéma suivant :

Si on prend $x_r = y_r = z_r = 1$: on aura l'angle de $\frac{\pi}{2}$ entre z et x qui se divise en deux et qui sera égale à $\frac{\pi}{4}$.

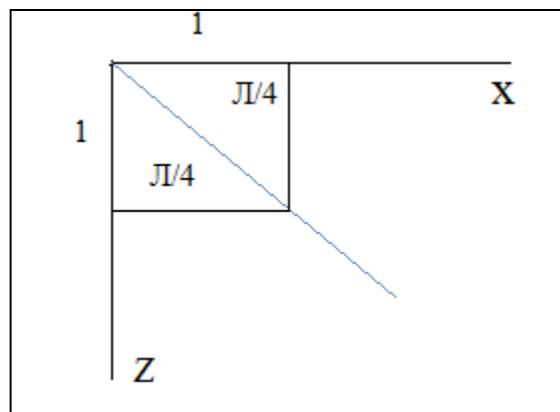


Figure II.7 : choix des références du réglage

On considère que les commandes sont correctement réglées si les critères du tableau ci dessous sont respectés.

Exigence	Critère	Niveau
Obtenir un bon comportement de l'attitude	Rapidité	$t_r < 5$ s
	Dépassement	$< 5\%$
	Consignes	$x_r = y_r = z_r = 1$ $\phi_d = 0$

Tableau II.3 : Critère du cahier des charges

Ces critères ont été choisis à base expérimentales car si nous obtenons des valeurs qui sont hors les valeurs proposées nous tombons dans un système trop vibratoire ou bien mal contrôlé (trop lents) surtout en attitude qui peut influencer le bon guidage de l'aéronef, à cause de la dynamique forte couplée du quadri-rotor.

Le choix des valeurs est basé sur la connaissance du point critique du processus. Expérimentalement, on boucle le processus sur un simple régulateur proportionnel dont on augmente le gain jusqu'à amener le système à osciller de manière permanente ; on se trouve ainsi à la limite de stabilité. Les premières valeurs proposées conduisent à un temps de montée relativement court malheureusement assorti d'un dépassement élevé.

II.4.COMMANDE D'ATTITUDE ET DE POSITION DU QUADRI-ROTOR :

L'objectif de cette partie est de modifier le modèle de la boucle ouverte afin de contrôler l'attitude (φ , θ et ψ) et la position z . En particulier, on souhaite pouvoir stabiliser le quadri-rotor à l'horizontale pour obtenir un vol stationnaire (chapitre I.4.2). Cette commande d'attitude représente une étape essentielle pour développer une commande complète en position du quadri-rotor.

Un type de commande linéaire est utilisé pour commander l'attitude et la position z , le PID (Proportionnelle-Intégrale-Dérivée) en utilisant le principe de Loop-shaping afin de définir les coefficients du régulateur.

II.4.1.Linéarisation :

Afin de pouvoir appliquer les commandes motionnées précédemment, une linéarisation autour d'un point de fonctionnement et nécessaire.

On prend le vecteur suivant comme point de fonctionnement :

$$[\varphi \ \theta \ \psi \ \dot{\varphi} \ \dot{\theta} \ \dot{\psi} \ x \ y \ z \ \dot{x} \ \dot{y} \ \dot{z}] = [0 \ 0 \ 0 \ 0 \ 0 \ 0 \ 0 \ 0 \ 0 \ 0 \ 0 \ 0]. \quad \text{Eq.II.3}$$

Le modèle linéaire associé est décrit par l'équation suivante :

$$\left\{ \begin{array}{l} \ddot{\varphi} = \frac{l}{I_x} \mathbf{u}_2 \\ \ddot{\theta} = \frac{l}{I_y} \mathbf{u}_3 \\ \ddot{\psi} = \frac{1}{I_z} \mathbf{u}_4 \\ \ddot{x} = \frac{1}{m} \mathbf{u}_x \mathbf{u}_1 \\ \ddot{y} = \frac{1}{m} \mathbf{u}_y \mathbf{u}_1 \\ \ddot{z} = -\mathbf{g} + \frac{1}{m} \mathbf{u}_1 \end{array} \right. \quad \text{Eq.II.4}$$

En appliquant la transformée de Laplace on aura Eq.II.4 qui devient :

$$\left\{ \begin{array}{l} \Phi(s) = \frac{l}{I_x s^2} \mathbf{u}_2 \\ \Theta(s) = \frac{l}{I_y s^2} \mathbf{u}_3 \\ \Psi(s) = \frac{1}{I_z s^2} \mathbf{u}_4 \\ X(s) = \frac{1}{m s^2} \mathbf{u}_x \mathbf{u}_1 \\ Y(s) = \frac{1}{m s^2} \mathbf{u}_y \mathbf{u}_1 \\ (Z + \mathbf{g})(s) = \frac{1}{m s^2} \mathbf{u}_1 \end{array} \right. \quad \text{Eq.II.5}$$

En Remarque que chaque fonction de transfère d'un état est un double intégrateur multiplier par un gain.

II.2.2. Les commande :

On va commencer la synthèse de la commande par le système découplé parce qu'il est le plus simple, puis on va appliquer les régulateurs trouvés sur le système couplé.

La synthèse va se faire sans tenir compte des dynamiques des actionneurs car elles sont sensées d'être négligeables devant les dynamiques du système, mais on va tenir compte de l'effet de saturation des moteurs car c'est un facteur pratique primordial.

Pour cela on va contrôler chaque commande avec l'état qu'il la convient (par exemple \mathbf{u}_2 avec ϕ ont un régulateur ...).

Par la suite on ajuste les gains des commandes suivant le modèle non linéaire complet décrit par les équations suivantes :

→ Pour les Angles :

$$\begin{cases} \ddot{\phi} = -\dot{\phi}\dot{\psi} \frac{(I_z - I_y)}{I_x} - \frac{I_r}{I_x} \overline{\Omega_r} \dot{\theta} - \frac{K_{fx}}{I_x} \dot{\phi}^2 + \frac{l}{I_x} \mathbf{u}_2 \\ \ddot{\theta} = -\dot{\phi}\dot{\psi} \left(\frac{I_z - I_x}{I_y} \right) - \frac{I_r}{I_y} \overline{\Omega_r} \dot{\phi} - \frac{K_{fy}}{I_y} \dot{\theta}^2 + \frac{l}{I_y} \mathbf{u}_3 \\ \ddot{\psi} = -\dot{\phi}\dot{\psi} \left(\frac{I_x - I_y}{I_z} \right) - \frac{K_{fz}}{I_z} \dot{\psi}^2 + \frac{1}{I_z} \mathbf{u}_4 \end{cases} \quad \text{Eq.II.6}$$

→ Pour le z :

On l'obtient à partir de la relation entre l'asservissement en vitesse des quatre moteurs de fonction de transfère suivante trouvé dans le **chapitre I**, la représentation de la commande \mathbf{u}_1 dans l'**Eq.II.2** et la fonction de transfère linéaire du dans l'**Eq.II.5**:

$$\mathbf{G}_T(s) = \frac{\sum \omega_m}{V_s} = \sum \left(\frac{\frac{1}{k_e}}{\tau_m \tau_e s^2 + \tau_m s + 1} \right) \quad \text{Eq.II.7}$$

De l'**Eq.II.2**

$$\mathbf{u}_1 = \mathbf{b} (\omega_1 + \omega_2 + \omega_3 + \omega_4) = \mathbf{b} \sum \omega_m \quad \text{Eq.II.8}$$

De l'**Eq.II.5**

$$(\mathbf{z} + \mathbf{g})(s) = \frac{1}{m s^2} \mathbf{u}_1 \quad \text{Eq.II.9}$$

On aura

$$(\mathbf{z} + \mathbf{g})(s) = \frac{\mathbf{b}}{m s^2} \sum \omega_m \quad \text{Eq.II.10}$$

Ou ω_m est la sortie d'asservissement d'un seul moteur et $\sum \omega_m$ représente réellement la sortie d'asservissement des quatre moteurs qui représente la sortie du bloc « brushless motors » dans notre programme SIMULINK représenter dans la figure **Figure.II.1** .

On voit bien qu'en commandant les vitesses des moteurs on aura bien le u_1 qui nous permet de commander le z, il suffit juste de divisé par la masse « m » et multiplier par un $-b$ (configuration des axes ou le z est orienté ver le bas) le retour d'état de la boucle d'asservissement en vitesse des moteurs.

→ Pour le x, y, ϕ_d, θ_d :

On a utilisé l'effet non holonome présenté dans le **chapitre I**.

III.4.3. Les signaux de commande :

Les signaux de commande sont interprétés a partir des signales radio **PWM** (Pulse Width Modulation), dans un intervalle de magnitude de -1 à 1. Ses signaux sont par la suite comparés avec les angles d'Euler obtenues par la centrale inertielle (appeler limitation de la commande). [8]

Une fois la commande est appliquée, l'interprétation des signaux de commande se fait à l'aide du block 'Motor Mixer' (**Fig II.8**). Ce block mixe les signaux d'attitude et de poussé (z) de tel façon que le moteur approprié délivre une augmentation ou une diminution de poussé et il représente le passage entre u_i, ω_i comme le montre la relation **Eq.II.2** et entre ω_i et **PWM**. La sortie du 'Motor Mixer' est un signal PWM pour chaque moteur. Il est important de garder la largeur d'impulsion entre $1000\mu s$ et $2000\mu s$. [8]

Après la combinaison des signaux, la valeur obtenue doit être multipliée par 1000 et ajouter à la valeur du idle_PWM qui est fixée à $1000\mu s$ afin d'assurer un minimum de vitesse de rotation des moteurs. [8]

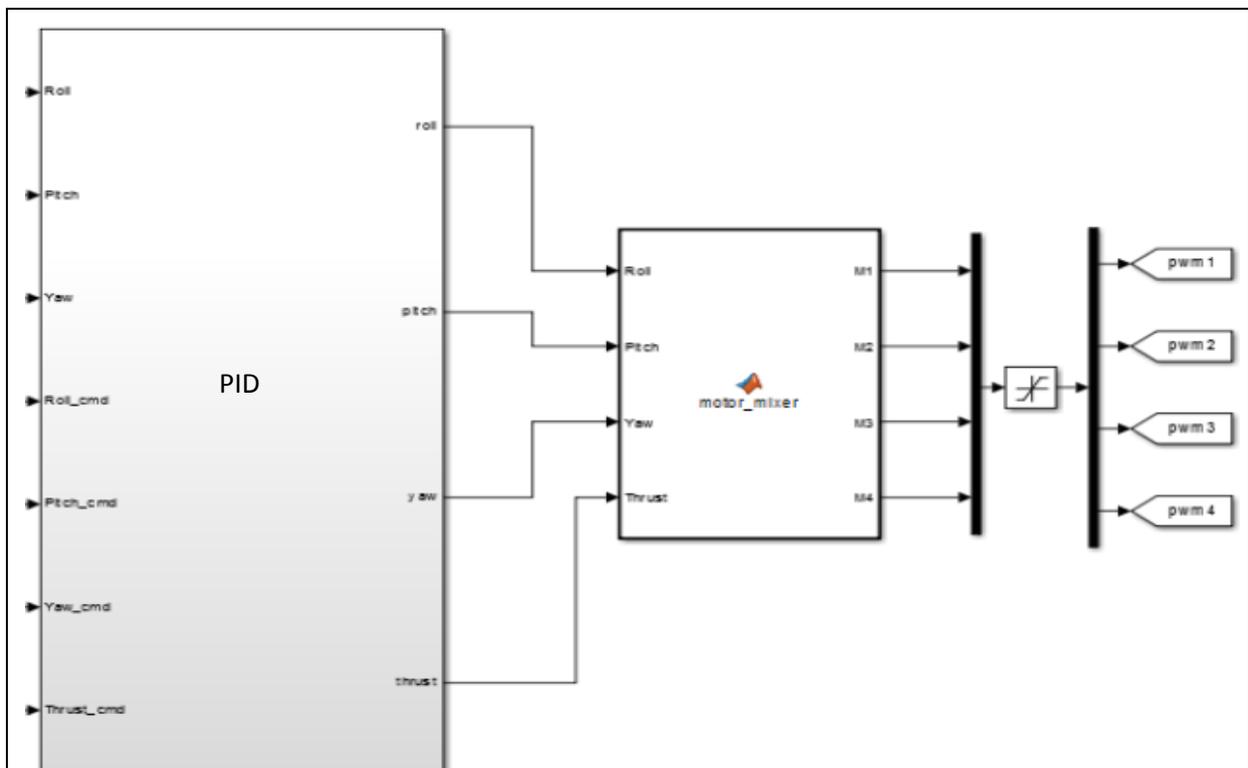


Figure.II.8: block 'Motor Mixer'.

La partie la plus importante du 'Motor Mixer' est l'équation qui combine les angles d'attitude et la poussée $b \sum \omega_m$ pour chaque servocommande (actuateur). Pour notre configuration, les équations des quatre moteurs sont choisies comme suit : [8]

$$M1 = ((\text{Roll} + \text{Pitch} - \text{Yaw}) / 2 * \text{Thrust} + b \sum \omega_m) * 1000 + \text{idle_PWM}; \quad \text{Eq.II.11}$$

$$M2 = ((-\text{Roll} + \text{Pitch} + \text{Yaw}) / 2 * \text{Thrust} + b \sum \omega_m) * 1000 + \text{idle_PWM}; \quad \text{Eq.II.12}$$

$$M3 = ((-\text{Roll} - \text{Pitch} - \text{Yaw}) / 2 * \text{Thrust} + b \sum \omega_m) * 1000 + \text{idle_PWM}; \quad \text{Eq.II.13}$$

$$M4 = ((\text{Roll} - \text{Pitch} + \text{Yaw}) / 2 * \text{Thrust} + b \sum \omega_m) * 1000 + \text{idle_PWM}; \quad \text{Eq.II.14}$$

II.5.PRINCIPE DE REGULATION PAR « PID » :

La commande par régulation PID est certainement la structure de commande la plus couramment utilisée dans l'industrie. Elle s'est, en effet, imposée dans plus de 90 % des architectures de commande, dont la plupart ne sont constituées que des actions proportionnelle et intégrale. [9]

La prévalence de cette approche vient, au-delà de sa simplicité, des performances qu'elle offre aux systèmes pilotés en boucle fermée, quel que soit leur domaine. Les raisons de ce succès sont principalement de nombre trois (3) : [9]

- Structure simple.
- Bonnes performances pour plusieurs processus.
- Réglable même sans un modèle spécifique du système contrôlé.

En robotique, la technique PID représente les bases du contrôle. Même si un grand nombre de différents algorithmes peuvent fournir de meilleures performances que PID, cette dernière structure est souvent choisie pour les raisons exprimées ci-dessus. Le principe de PID repose sur le fait de délivrer un signal de commande $e(t)$ à partir de la différence entre la consigne $r(t)$ et la mesure $y(t)$ (**Figure.II.9**), d'où: [10]

$$E(t) = r(t) - y(t) \quad \text{Eq.II.15}$$

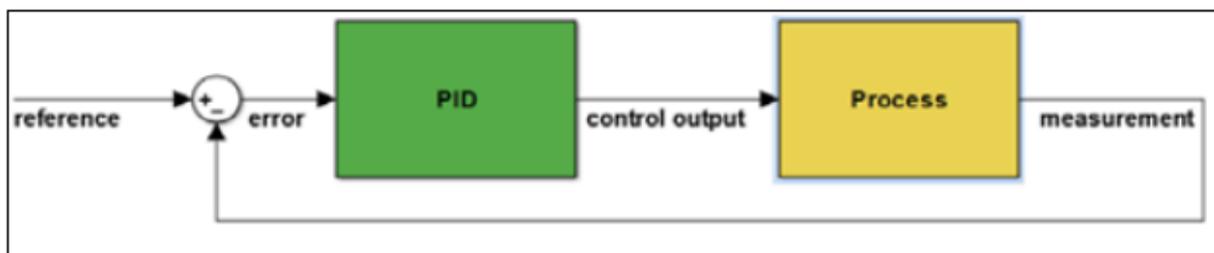


Figure.II.9 : Le principe de commande par PID [10]

Afin de minimiser $e(t)$ trois contribution sont additionnées à la sortie du contrôleur .La **Fig.II.10** présente la structure principale du PID: [9]

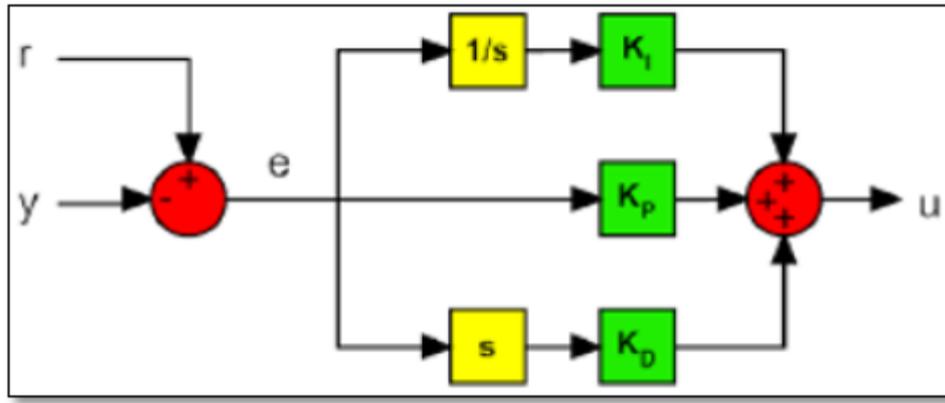


Figure.II.10 : Le principe de commande par PID. [10]

La sortie $u(t)$ du PID peut être donc calculée comme suit : [9]

$$R(t) = K_p e(t) + K_i \int_0^t e(\tau) d\tau + K_d \frac{de(t)}{dt} \quad \text{Eq.II.15}$$

Les blocs " $\frac{1}{s}$ " et " s " représentent les opérations d'intégration et de dérivation.

Où :

K_p : est le coefficient proportionnel.

K_i : est le coefficient d'intégrale.

K_d : est le coefficient dérivé.

Dans la première contribution (P) l'erreur est virtuellement amplifiée d'un certain gain constant qu'il conviendra de déterminer en fonction du système.

$$P = K_p(t) \quad \text{Eq.II.16}$$

L'idée étant d'augmenter l'effet de l'erreur $e(t)$ sur le système afin qu'il 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 K_p démesuré, le système peut même diverger. [9]

La contribution de I varie en fonction de l'intégrale de l'erreur.

$$I = K_i \int_0^t e(\tau) d\tau \quad \text{Eq.II.17}$$

Même si cette composante augmente le dépassement et le temps de stabilisation, elle a une propriété unique: elle élimine l'erreur d'état stationnaire.

La contribution de D varie en fonction de la dérivée de l'erreur. [9]

$$D = K_d \frac{de(t)}{dt} \quad \text{Eq.II.18}$$

Ceci aide à diminuer la composante de la remise des attitudes et le temps de stabilisation.

Dans le domaine de Laplace, la structure traditionnelle PID peut être réécrite conformément à l'équation II.14 [9]

$$R(s) = \left(K_p + K_i \frac{1}{s} + K_d s \right) \quad \text{Eq.II.19}$$

Comme cette fonction est impropre, il n'est pas physiquement possible (en raison de la composante dérivée). Après une certaine fréquence, la contribution dérivative doit être atténuée pour filtrer le bruit, alors il faut inclure un filtre passe bas. Pour cette raison, dans le dérivateur un véritable pôle est ajouté comme indiqué dans l'équation II.20 [9]

$$R(s) = \left(K_p + K_i \frac{1}{s} + K_d \frac{N}{1+s} \right) \quad \text{Eq.II.20}$$

Avec :

N: le coefficient du filtre passe bas

❖ **La structure traditionnelle PID présente deux inconvénients principaux:** [9]

- L'action dérivative est calculée à partir de l'erreur. Si on excite l'entrée par un signal, la sortie du dérivé présenterait une impulsion. Ce mouvement brusque peut saturer les actionneurs et repousser le système de la zone linéaire. Pour cette raison, la plupart de l'architecture présente seulement le PID action dérivée à la sortie du processus.
- L'action intégrale combinée avec une saturation de l'actionneur peut fournir un effet non linéaire qui peut diminuer les performances du système de contrôle. Lorsque la valeur intégrale et les changements d'erreur sont grands, il est nécessaire d'attendre beaucoup de temps avant que le système restaure son comportement linéaire (après la "décharge" de l'action intégrale). Ce phénomène est appelé 'intégrante liquidation'. Pour éviter cela, un saturateur est ajouté après l'intégrale pour limiter ses valeurs maximales et minimales.

La Fig.II.11 Présente la structure améliorée du PID : [9]

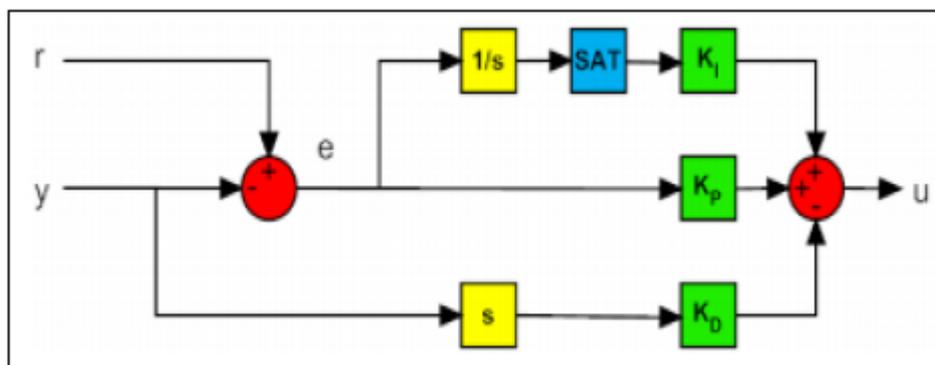


Figure.II.11 : La structure améliorée du PID. [9]

Remarque :

Comme on a Remarqué que chaque fonction de transfère d'un état est un double intégrateur multiplier par un gain dans l'**Eq.II.5**, on va implémenter que des régulateurs PD avec un filtre passe bas de coefficient égale à N (pas d'influence du terme avec une fonction égale à :

$$R(s) = \left(K_p + K_d \frac{N}{1 + \frac{N}{s}} \right) \quad \text{Eq.II.21}$$

Aucun effet du nouveau pôle donné par le filtre aux basses fréquences mais le gain à haute fréquence est limité à N.

II.6.PRINCIPE DE LA COMMANDE PAR LA METHODE DU**« LOOP-SHAPING » :**

Une méthode alternative, celle du loop-shaping, se focalise sur le transfert en boucle ouverte de l'ensemble système-correcteur [12].

$$G_{Bo}(s) = R(s).G(s) \quad \text{Eq.II.22}$$

Ou :

R(s) est la fonction de transfère du régulateur PID.

G(s) est la fonction de transfère du système.

Dans cette approche, une version de départ du correcteur est d'abord établie par le concepteur, dans notre cas on va utiliser des correcteurs PID (principe élaboré précédemment), (voir Figure.II.9).

❖ Les objectifs d'utilisation de la méthode du Loop-shaping:

- Fixer la bande passante
- Augmenter le gain en basse fréquence afin de rejeter les perturbations
- Faire chuter le gain en haute fréquence pour des raisons de robustesse face aux incertitudes mal modélisées.

Ce premier correcteur n'a pas besoin d'être stabilisant. Dans un second temps, un correcteur R(s) est synthétisé de manière à assurer la stabilité et à minimiser la norme du transfert entre le vecteur d'entrée e(s) et le vecteur des sorties y(s), afin d'apporter la robustesse suffisante.

Le correcteur final a un ordre égal à celui du système G (s) additionné de deux fois de celui des pondérations obtenus des coefficients du correcteur. En effet, contrairement au cas précédent, les pondérations font partie intégrante du correcteur à implanter. [11]

❖ Les étapes d'implémentation du Loop-shaping :

1^{ère} étape : Calculer la fonction de Transfer correcteur-système en boucle ouverte :

Comme notre système représente un double intégrateur multiplié par un gain de la forme $G(s) = \frac{K}{s^2}$ et le régulateur PD et de la forme $R(s) = (K_p + K_d s)$ on écrit la fonction de transfère correcteur-système en boucle ouverte de la forma suivante :

$$G_{Bo}(s) = \frac{K K_p}{s^2} \left(1 + \frac{s}{\frac{K_p}{K_d}} \right) \quad \text{Eq.II.23}$$

2^{ème} étape : Tracé de Bode de la fonction du système G(s) :

On trace Bode à l'aide de la fonction **margin(G(s))** sur MATLAB de la fonction **G(s)** pour connaitre la fréquence de coupure à $|G(\omega_c)|=0$, la marge de gain et la marge de phase.

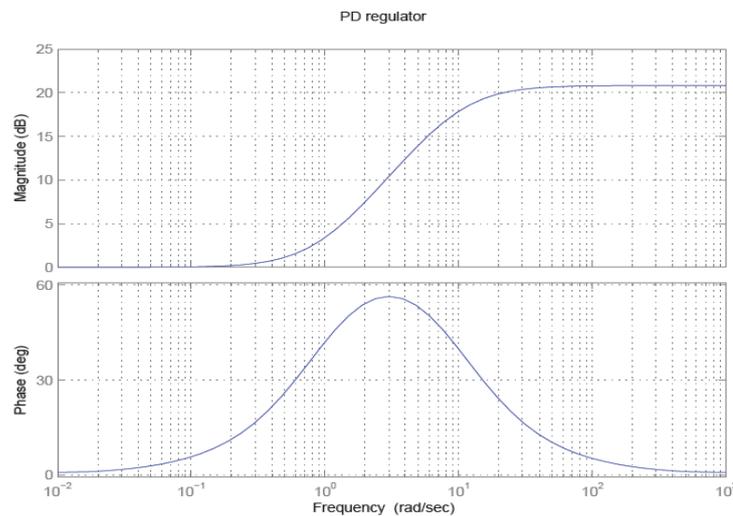


Figure.II.12 : Exemple de tracé de Bode avec PD. [13]

3^{ème} étape : Appliquer le principe de loop-shaping soit à un régulateur à avance de phase ou à un retard de phase (dans notre cas on va utiliser à avance de phase car on a choisit un régulateur PD)

On va choisir le gain et la fréquence de coupure désirés en respectant la règle d'or suivante : [13]

$$\omega_{cd} \leq 2\omega_c \quad \text{Eq.III.24}$$

Où : ω_{cd} est la fréquence de coupure désirer.

On impose la règle de loop-shaping pour un régulateur à avance de phase à partir De l'Eq.II.23 :

$$\frac{K_p}{K_d} \geq \omega_{cd} \quad \text{Eq.II.25}$$

Un déplacement d'une décade sur l'axe des fréquences équivalent à un déplacement de 40 dB au niveau de la marge de gain ou K_p sera égale au déplacement. Car le dénominateur de $G_{Bo}(s)$ et d'ordre 2.

4^{ème} étape : Tracé de Bode de la fonction du système $G_{Bo}(s)$:

On trace Bode encore une fois à l'aide de la fonction **margin**($G_{Bo}(s)$) sur MATLAB pour connaître le nouveau comportement imposé par la méthode de loop-shaping.

Remarque : [13]

- Les techniques du Loop-shaping sont les plus adéquates pour les systèmes SISO (single-input single-output) pour cela on a mit chaque sortie avec un régulateur PD.
- Ils fournissent une vue sur les propriétés du domaine fréquentiel et de la boucle fermée (bande passante, filtrage du bruit, robustesse à l'incertitude, etc.)
- La plupart des contrôleurs industriels à une seule boucle sont des PID et plus de 90% des PID sont des PD.

II. 7. SYNTHÈSE DES LOIS DE COMMANDE POUR LE QUADRI-ROTOR :

On propose, dans cette section, une stratégie de commande développer dans [1] et qui est basée principalement sur deux boucles (boucle interne et boucle externe). La boucle interne contient quatre lois de commande :

- Commande de roulis (ϕ)
- Commande de tangage (θ).
- Commande de lacet (ψ).
- Commande d'altitude Z.

La boucle externe inclut deux lois de commande de positions x et y . La boucle de commande externe génère une désirée pour le mouvement de roulis (ϕ_d) et de tangage (θ_d) (voir paragraphe **I.5.2. Liaisons non holonomes** dans le chapitre II) en passant par le bloc de correction. Ce bloc corrige la rotation de roulis et de tangage en fonction du désiré de lacet (ψ_d).

Le schéma ci-dessous représente la stratégie de commande adoptée: [1]

On va utiliser des régulateur PD élaborer par la méthode du Loop-shaping afin de déterminer les constantes de régulation pour les 6 commande roulis (ϕ), tangage(θ), lacet, X, Y,Z avec :

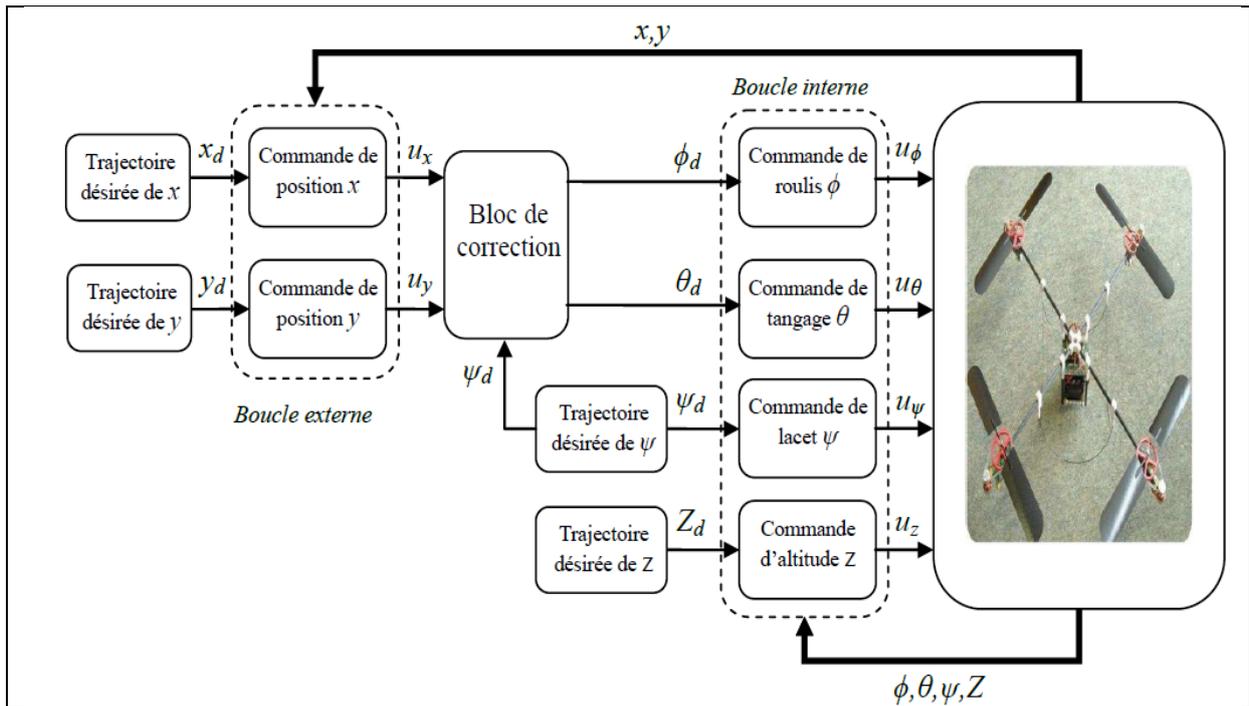


Figure.II.13 : Illustration de la structure de commande du quadri-rotor. [1]

II.7.1. Commande par le régulateur PID avec la méthode du Loop-shaping :

Les performances de la commande par régulation PID sont maintenant évaluées en simulation sur le modèle dynamique du quadri-rotor donné par **Eq.II.5**. L'évaluation est réalisée sur la capacité de la commande à garantir la bonne exécution d'une manœuvre désirée. Les valeurs des fréquences de coupure et désirer des différents états ainsi que leurs tracer de Bode sont présentées dans le Tableau.II.4 et les figures II.14, II.15, II.16, II.17, II.18 et II.19 aussi bien les valeurs des paramètres calculées des différents régulateurs PID sont mentionnées dans le Tableau.II.5, ces valeurs ont été sélectionnées sur la base des travaux de [1] et [3] puis adaptées aux besoins de notre modèle avec la méthode de réglage par Loop-shaping.

Les états	ω_c	ω_{cd}
roulis (ϕ)	12.5 rd/s	47 rd/s
tangage (θ)	13.9 rd/s	57 rd/s
lacet (ψ)	26.7 rd/s	57 rd/s
Z	1.96 rd/s	5 rd/s

Tableau II.4 : les fréquences de coupure et désirées

Gains	Kp	Ki	Kd	N
Roulis	9.5	0	0.3	350
Tangage	9.5	0	0.3	350
Lacet	12	0	0.8	300

Gains	Kp	Ki	Kd	N
X	0.08	0.001	0.001	77
Y	0.08	0.001	0.001	77
Z	6.25	0	0.25	350

Tableau II.5 : Gains des PID de la commande.

❖ Tracer de Bode des différents états :

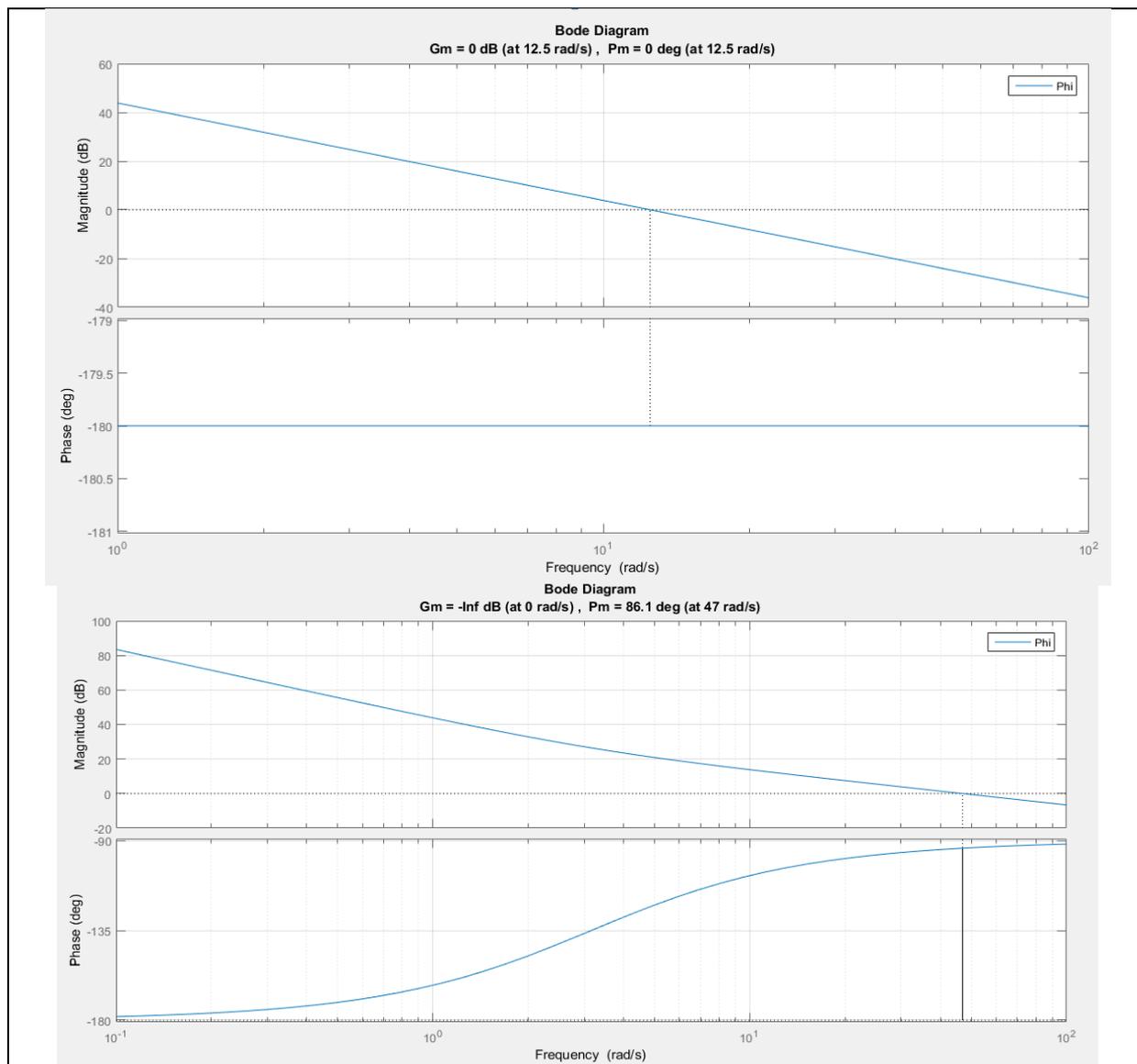


Figure.II.14 : Tracer de Bode de roulis (ϕ).

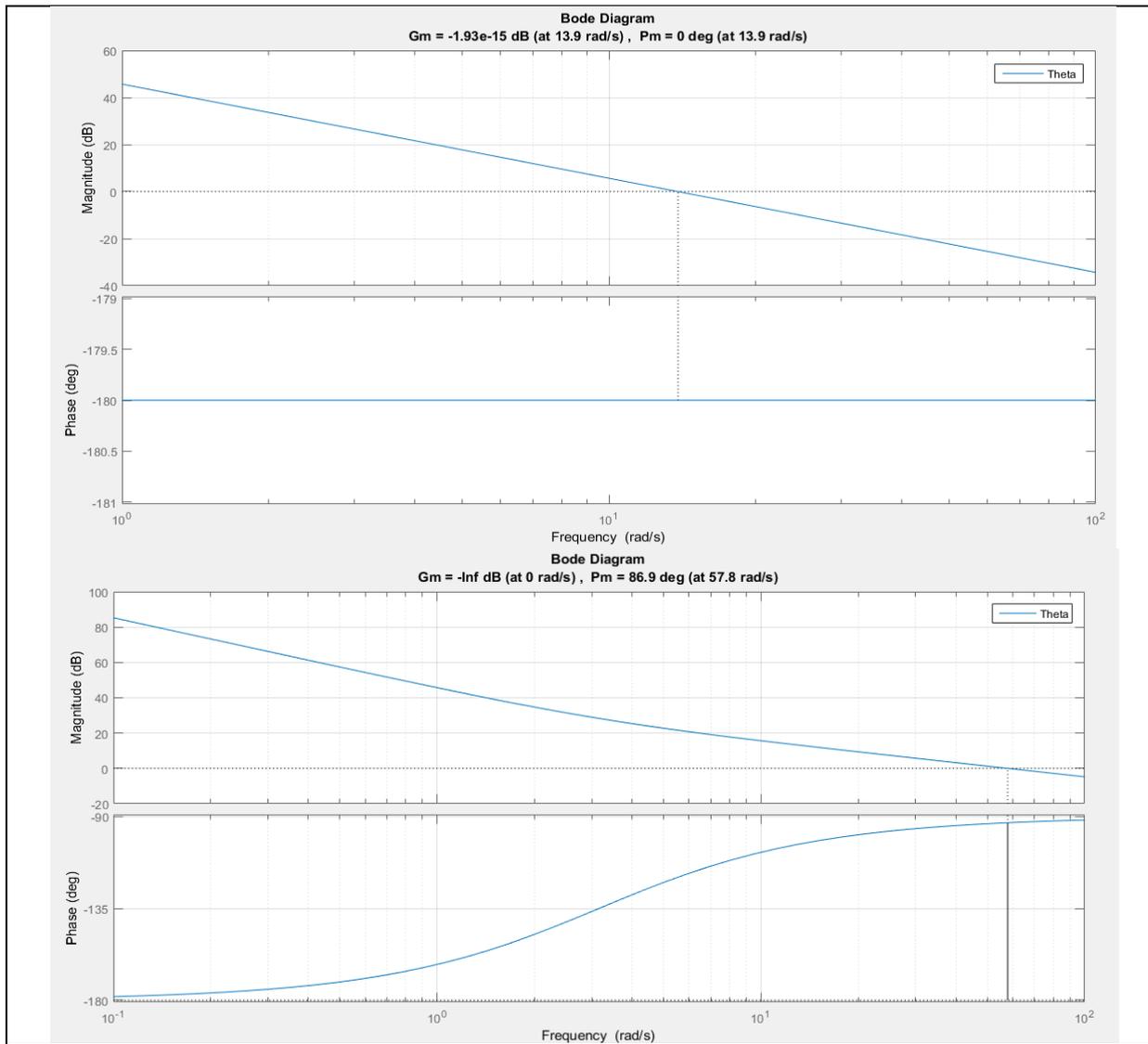
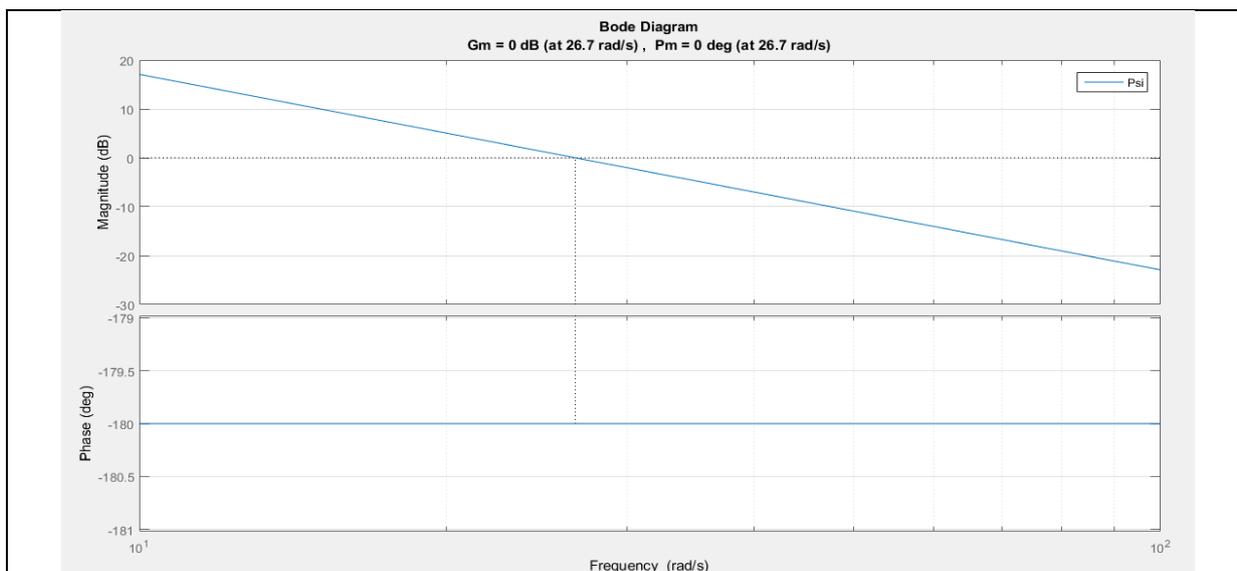


Figure.II.15 : Tracer de Bode de tangage (θ)



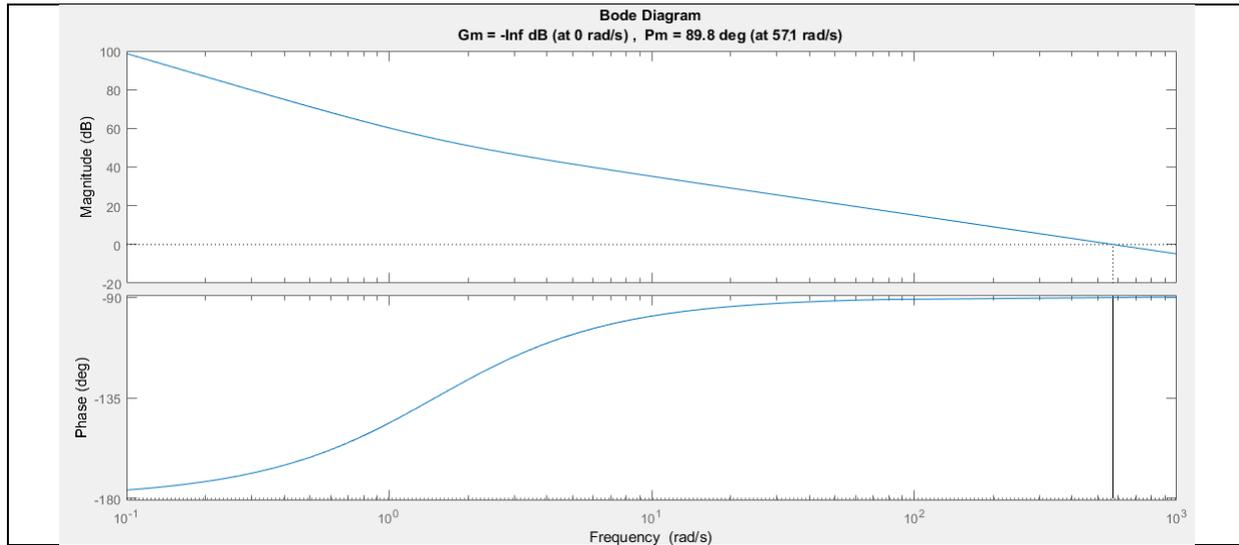


Figure.II.16 : Tracer de Bode de lacet (ϕ).

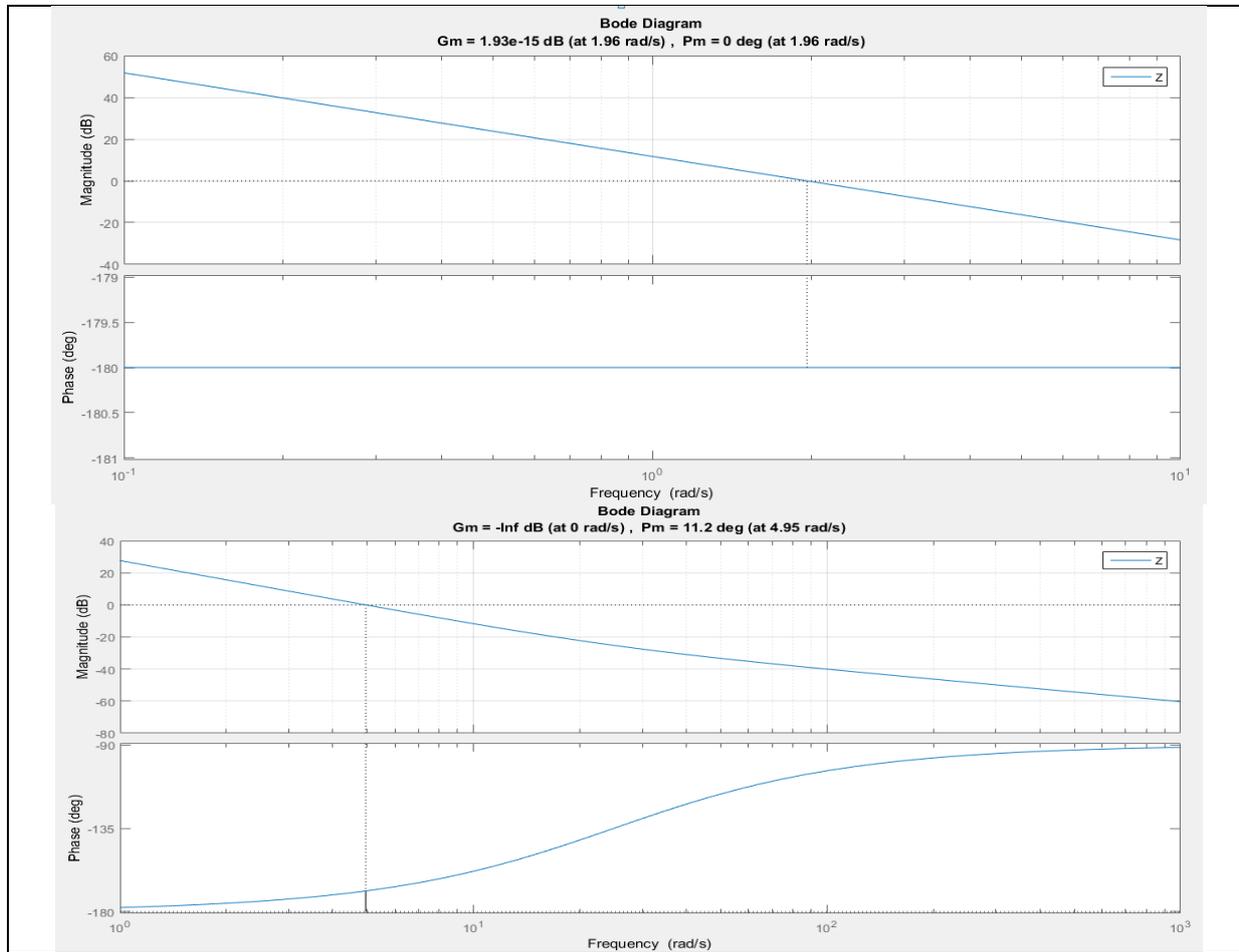


Figure.II.17 : Tracer de Bode pour le Z.

→ Pour les paramètres du block de correction (**Figure.III.13**) on a utilisé la méthode de tuning avec SIMULINK MATLAB (plus de détail sur le concept de tuning (Voir **Annexe 4**))

On remarque que les différents états du système répondent aux fréquences imposées par la méthode du loop-shaping c'est ce que les figures précédentes montrent.

II.7.2. Résultats de simulation :

Les valeurs initiales des angles de roulis et de tangage sont définies par les relations non holonome sinon l'angle de lacet on l'a pris égale à '45°', et ce qui concerne les valeurs initiales des position X, Y et Z on les a pris égales toutes à '1'.

→ Les résultats obtenus pour la stabilisation de position du Quadri-rotor par la commande PID en passant par le concept du Loop-shaping sont donnés sur la **Figure.II.18**.

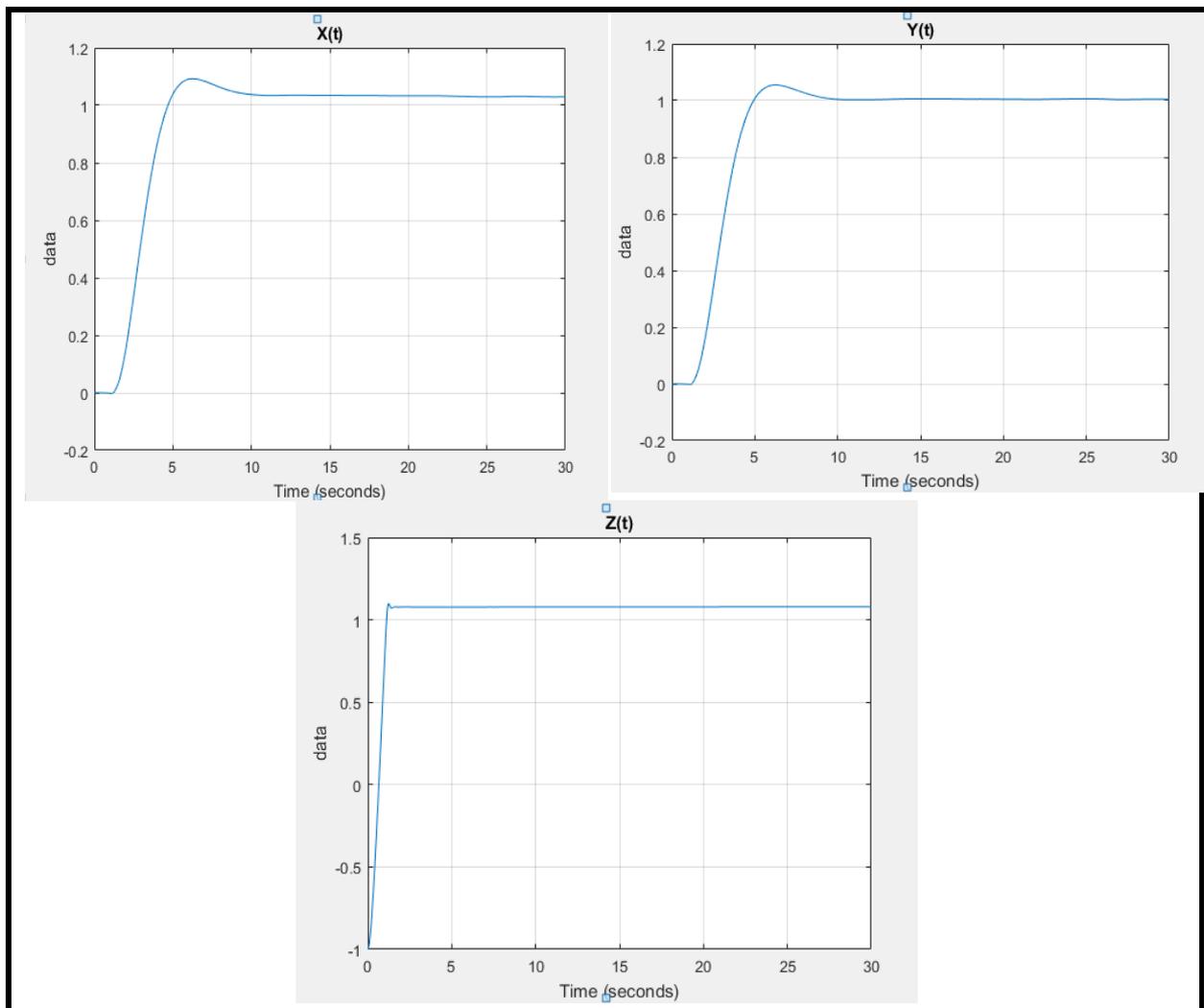


Figure.II.18 : Réponse de la position d'un contrôleur PID.

On remarque que les réponses des positions sont beaucoup plus meilleures que les réponses précédentes (une meilleure stabilité), aussi on voit que le Z répond avant le X et le Y et cela est dû aux équations d'états du système reliant les positions.

→ Les résultats obtenus pour la stabilisation d'attitude du Quadri-rotor par la commande PID sont donnés sur la **Figure.II.19**.

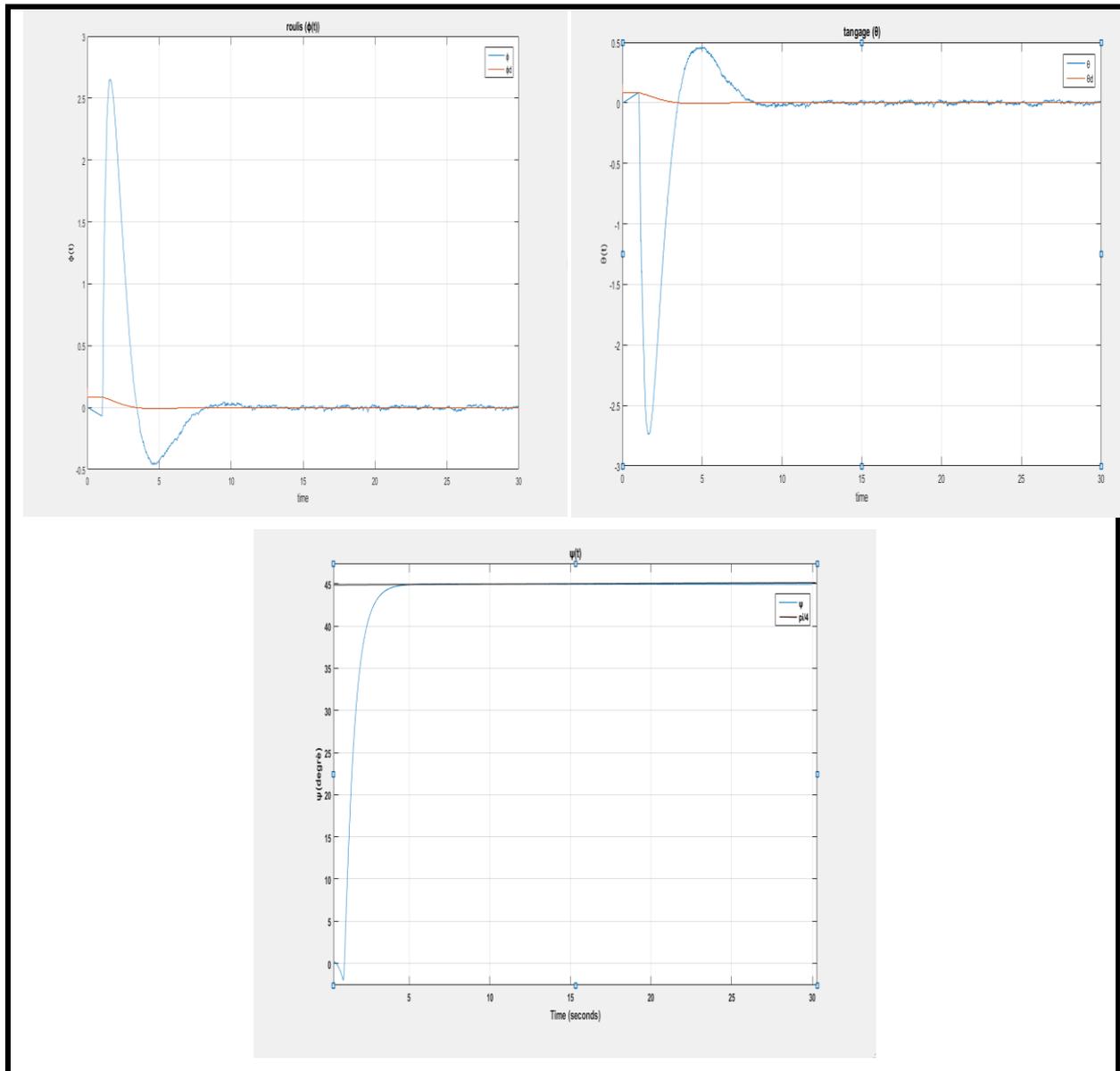


Figure.II.19 : Réponse d'attitude d'un contrôleur PID.

Sur la base de ces résultats, et malgré que la commande par régulation PID avec le concept du Loop-shaping ne prenne pas en compte le couplage entre la dynamique de

translation et celle de rotation, les performances de l'algorithme de commande sont satisfaisantes. Les réponses obtenues présentent une amélioration remarquable dans le temps de réponse et l'erreur statique (de l'ordre de 10^{-1} à la moyenne des trois angles) par rapport aux travaux de [3] et [24].

→ Les résultats obtenus pour la commande du roulis ' U_2 ', tangage ' U_3 ', lacet ' U_4 ' et le Z ' U_1 ' du Quadri-rotor par la commande PID sont donnés sur la **Figure.II.20**.

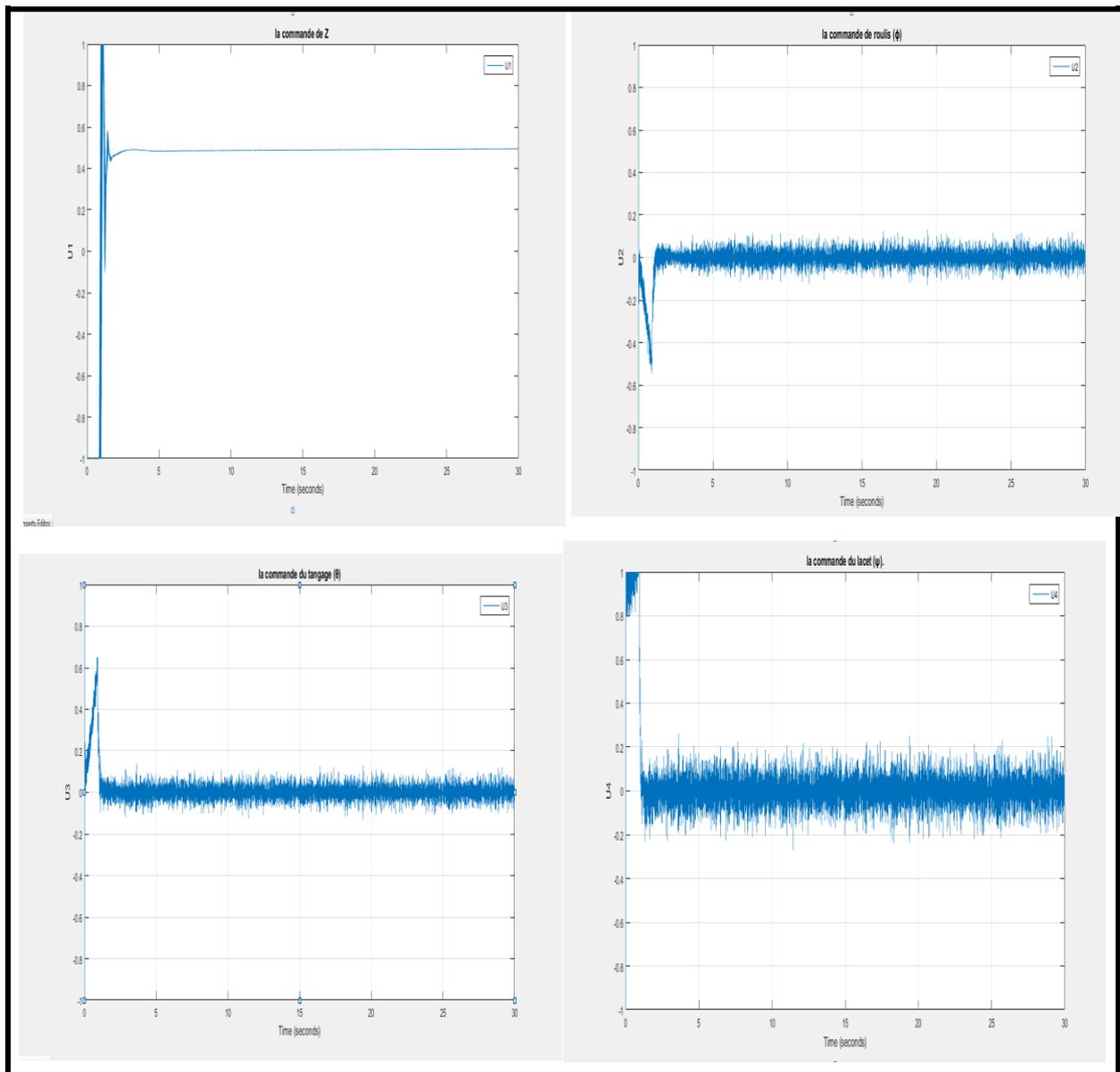


Figure.III.20 : Evolution des commandes U_1 ; U_2 ; U_3 et U_4 .

→ Les résultats obtenus pour La variation des signaux de commande en PWM sont donnés sur la **Figure.II.21**.

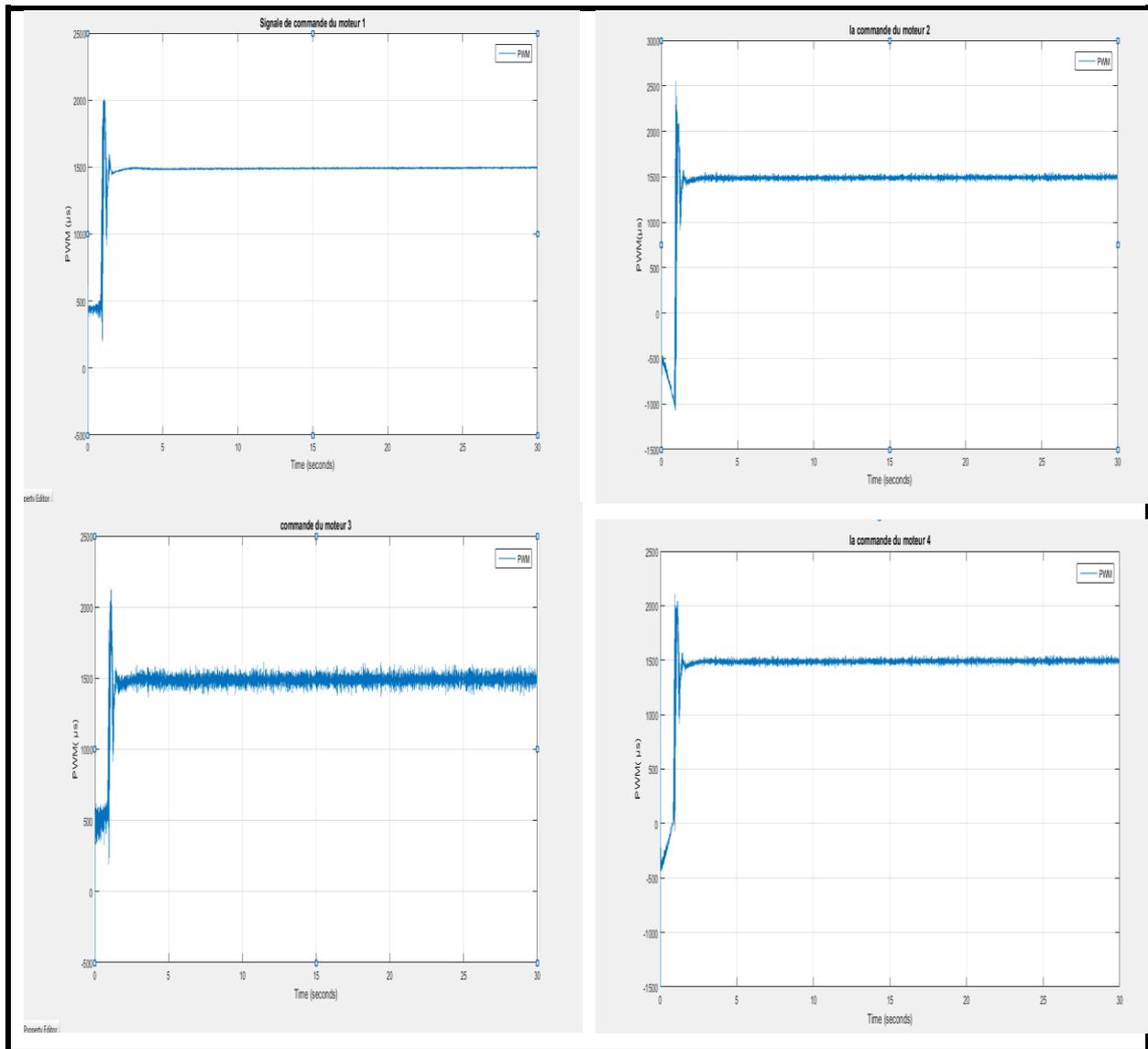


Figure.II.21 : Réponses des signaux de commande PWM d'un contrôleur PID

Il est remarquable de la **Figure.II.21** que les signaux de commandes satisfont les limitations imposées en termes d'intervalle de fonctionnement et largeur d'impulsion, ce qui assure le bon fonctionnement des ESC (Engine Speed Controller) ou aussi appelé le driver dans une application réelle.

L'objectif dernier des commandes est que le système soit robuste, rapide et précis tout en limitant les dépassements et la commande PID à suffisamment répondu au cahier des charges car Ensuite, on peut légèrement faire varier les valeurs de K_p , K_i et K_d selon le compromis précision, rapidité et stabilité recherché pour corriger la différence qui existe entre la réalité et la théorie (voir **chapitre III**).

II.8.CONCLUSION :

Le travail présenté dans ce chapitre porte sur la stabilisation de l'attitude et de la position du Quadri-rotor. La commande a été élaborée en utilisant le régulateur PID avec la méthode du loop-shaping afin de déterminer les constantes des régulateurs. L'objectif de la loi de commande est de maximiser l'efficacité des actionneurs sans prendre de risque dans la stabilité du système. Nous avons montré de plus que cette approche permet d'obtenir de bonnes performances en respectant les contraintes de vitesse angulaire.

Les résultats de simulation effectuée sur un modèle du Quadri-rotor, montrent que cette structure de commande donne des résultats acceptables en termes de stabilisation et performance. Mais cette commande reste limitée en termes de simplification adopté ou modèle et face aux perturbations, l'avantage principal de la commande appliquée est la simplicité.

III.1.Introduction :

Après avoir présenté le modèle du quadri-rotor et les algorithmes de contrôle, on arrive dans ce chapitre à la réalisation d'un système MCG (Modélisation, Contrôle et Guidage) complet qui permet au quadri-rotors de suivre une trajectoire bien déterminée en utilisant le logiciel SIMULINK sur Matlab. L'algorithme de guidage qui permet de générer la trajectoire et la commande position associées est conçu de telle façon à assurer que le quadri-rotor compile sa mission en toute sécurité.

Une simulation hardware in the loop (matériel dans la boucle) représente une étape primordiale avant le teste réel afin de confirmer que la carte de commande est adéquate avec les algorithmes programmés, la méthode ainsi que le matériel utilisé seront présenté en détail.

III.2. MATERIEL UTILISE :

III.2.1. Présentation du Régulateurs de vol :

On trouve sur le marché deux types de régulateurs de vol. Les kits « fermé » qui sont composés de plusieurs composants connectés entre eux, produits par la même marque et se comportant comme une « boîte noire ». Seule l'entrée et la sortie sont accessibles, les composants ne sont pas visibles et la régulation est faite par un logiciel propriétaire. La deuxième catégorie englobe les cartes 'open source'. Le code est libre, même s'il n'est souvent disponible qu'après l'achat de la carte correspondante. Ce type de régulateurs est particulièrement intéressant car il est possible de modifier le code, bien que ce dernier soit tout à fait fonctionnel d'origine, pour améliorer les performances de stabilisation. Ces cartes 'open source' se basent majoritairement sur un Arduino mais également sur RaspberryPI par exemple. [26]

Marque	Type	Spécificité	Prix [\$]
DJI	Fermé	Leader de marché	150-1300
Hoverfly	Fermé	Destiné aux lobbyistes et aux professionnels	225-450
xaircraft	Fermé	Destiné aux professionnels	400-600
Feyu Tech	Fermé	Milieu de gamme	100-200
Optimus	Fermé	Bas gamme	45
Multiwii	Open source	Précurseurs du quadri-rotor open source	20-65
ArduiPilot	Open source	Principal concurrent de Multiwii	150-250
OPenPilot	Open source	Processeur 32 bits	60-200
AeroQuad	Open source	Orienté vers DIY	100-150

Tableau.III.1 : Acteurs majeurs des régulateurs de vol. [26]

Dans ces deux classes de régulateurs on retrouve un certain nombre de capteurs permettant la stabilisation du quadri-rotor. Chaque capteur permet une meilleure stabilisation ou une fonctionnalité en plus dans le contrôle du drone. Voici la liste des composants qu'on retrouve fréquemment dans les régulateurs actuels. D'autres capteurs peuvent être ajoutés mais ne figurent pas par défaut dans les contrôleurs de vol. [26]



Figure.III.1 : Régulateur du vol (Ardu-pilot).[28]

Par exemple on cite :

III.2.1.1. Gyromètre :

Le gyromètre est un capteur qui mesure la vitesse angulaire. Sur les quadri-rotors, il mesure la vitesse de rotation autour d'un des axes de direction qui peut être ensuite comparée à la consigne souhaitée. C'est le seul capteur indispensable pour la stabilisation. [26]

III.2.1.2. Accéléromètre :

L'accéléromètre mesure l'accélération de l'objet sur lequel il est placé. Les quadri-rotors sont équipés d'un accéléromètre 3 axes qui permettent de déterminer l'accélération du modèle selon les 3 degrés de liberté de translation. L'accéléromètre est également capable de mesurer l'inclinaison d'un objet car il est influencé par la force de gravité.

Combiné aux gyromètres et après application d'un filtre, ces capteurs permettent d'obtenir la position du quadri-rotor dans l'espace à l'exception de la dimension verticale. [26]

III.2.1.3. Baromètre :

Le baromètre mesure la pression atmosphérique. En mesurant la différence de pression par rapport à une référence, il permet au régulateur de vol de connaître la composante verticale du quadri-rotor. Ce capteur est utilisé afin de maintenir la plateforme à une hauteur définie (mode *Altitude Hold*). Pour les déplacements proches du sol, un capteur à ultrasons peut également être utilisé afin de maintenir une altitude fixe. [26]

III.2.1.4. Magnétomètre :

Le magnétomètre mesure le champ magnétique. Il est utilisé dans des modes de navigation un peu plus avancés. Il permet par exemple de définir une direction de pilotage fixe quel que soit la position du quadri-rotor. L'avant de l'appareil n'est plus défini par rapport à la structure mais par rapport à une direction géographique. Cela facilite le pilotage puisque la rotation du quadri-rotor sur l'axe vertical n'a plus d'influence. Le magnétomètre est également utilisé dans les modes de pilotages utilisant le GPS. [26]

III.2.1.5. GPS :

Le GPS n'est pas utilisé pour stabiliser l'appareil à proprement parler. Il est utilisé pour programmer des déplacements à l'avance (*Waypoint*) et pour revenir, automatiquement au point de lancement (*Return to Home*). Le GPS permet également de maintenir l'appareil à un endroit précis (*Position Hold*). Certains modèles utilisent également un capteur à flux optique (même principe que sur les souris d'ordinateur à fonctionnement optique) afin de garantir un positionnement fixe de l'appareil, mais il ne marche qu'à des hauteurs relativement faibles. [26]

III.2.2. Architecture de L'Ardu-pilot utilisé :

L'Ardu-pilot 2.6 est un système de pilotage automatique complet et Open-Source qui permet de transformer n'importe quelle machine Radio Commandée en véhicule totalement autonome et donc capable d'effectuer des missions programmées avec des points de passage.

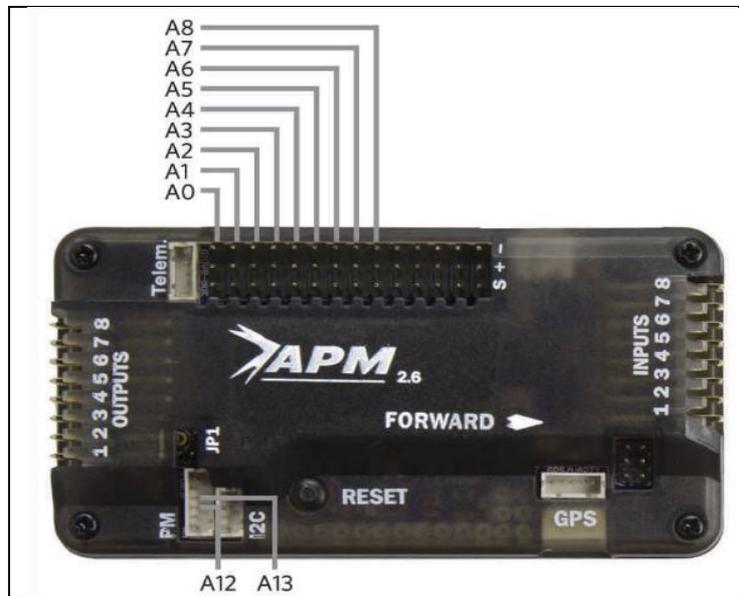


Figure. III.2 : l'Ardu-pilot [28]

Il dispose de 8 entrées et 8 sorties digitales et 10 Pins analogiques qui sont les suivantes : [28]

- **Pin 0 à 8:** L'APM2 possède une rangée de broches d'entrée analogiques, étiquetées de A0 à A8 en dessous de la carte.

Toutes ces broches peuvent supporter jusqu'à 5V et peuvent être utilisées pour toute entrée analogique générale. Ils sont couramment utilisés pour les entrées de vitesse..

- **Pin 12:** c'est le connecteur de gestion d'alimentation en courant, il accepte jusqu'à 5V, habituellement attachée à la brique de puissance.
- **Pin 13:** c'est le connecteur de gestion de l'alimentation en tension, il accepte jusqu'à 5V, généralement fixé à la brique d'alimentation.

III.2.3. APM2 Simulink Blockset : (l'ardu-pilot toolbox sur SIMULINK) [30]

Dans le cadre de l'élaboration d'un nouveau cours d'orientation, de navigation et de contrôle à l'Université d'aéronautique Embry-Riddle, une toolbox Simulink a été développée et permettra de travailler uniquement avec Matlab / Simulink pour le développement du code MCG.

Cette toolbox utilise la nouvelle fonctionnalité de Run-On-Target-Hardware qui est apparue pour la première fois dans Matlab 2012a, ce qui permet aux utilisateurs de Simulink

de générer automatiquement du code à partir d'un modèle Simulink pour certaines cibles matérielles. Cette fonctionnalité n'était précédemment disponible que pour les propriétaires de Simulink Coder et d'autres boîtes à outils similaires.

Cet « APM2 Simulink Blockset » présente le package de notre Ardu-pilot utilisé et on peut le télécharger facilement sur [30] avant d'avoir bien sur établie la connexion « PC.(Simulink/Matlab)_Ardu-pilot »

On trouve dans la librairie de l'APM2 Simulink Blockset de notre Ardu-pilot utilisé, les différents blocks des composants présenté précédemment plus deux blocks d'entrées/sorties analogique et celui du périphérique (pilot), sous forme de S-function Simulink/Matlab.

La **figure.III.3** représente les blocks de l'ardu-pilot qu'on va utiliser par la suite

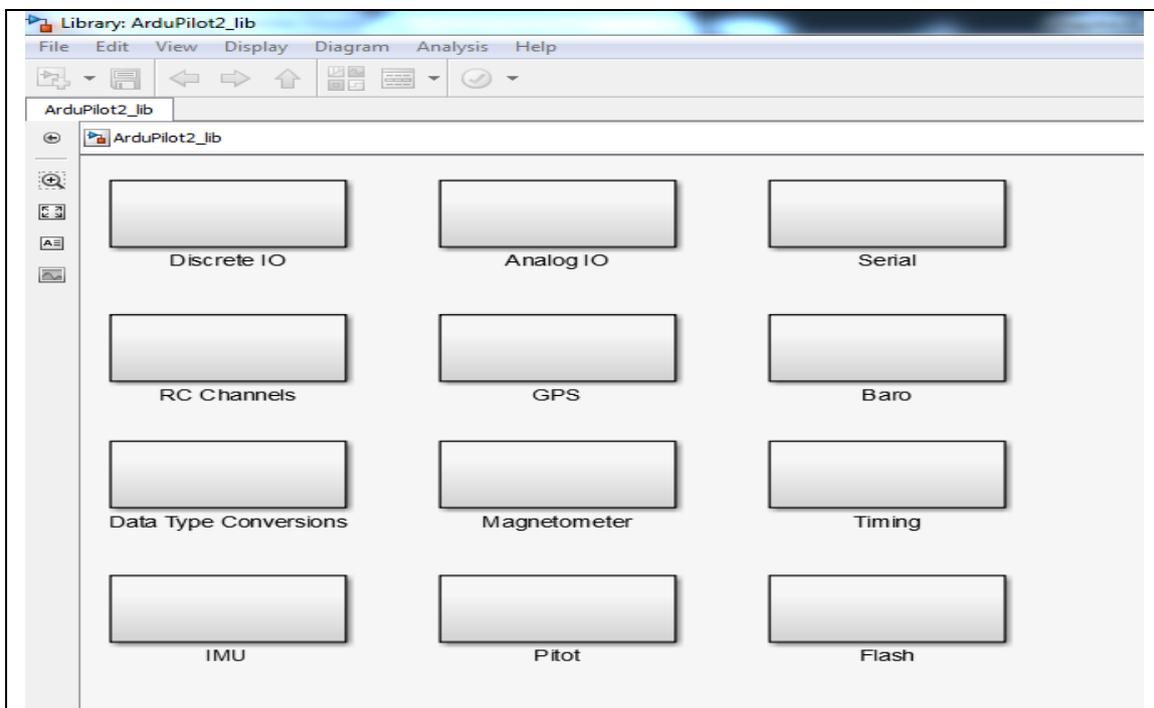


Figure. III.3 : la librairie de l'Ardu-pilot [30]

III.3.GENERATION DE LA TRAJECTOIRE :

Dans un système de guidage, les commandes d'attitude sont automatiquement générées à partir d'un système de commande de position qui permet de suivre une trajectoire désirée formée de plusieurs 'waypoints'. Plusieurs algorithmes de guidage sont proposés dans la littérature. Dans le cadre de notre projet, on a choisis un guidage par champ de vecteur. [25]

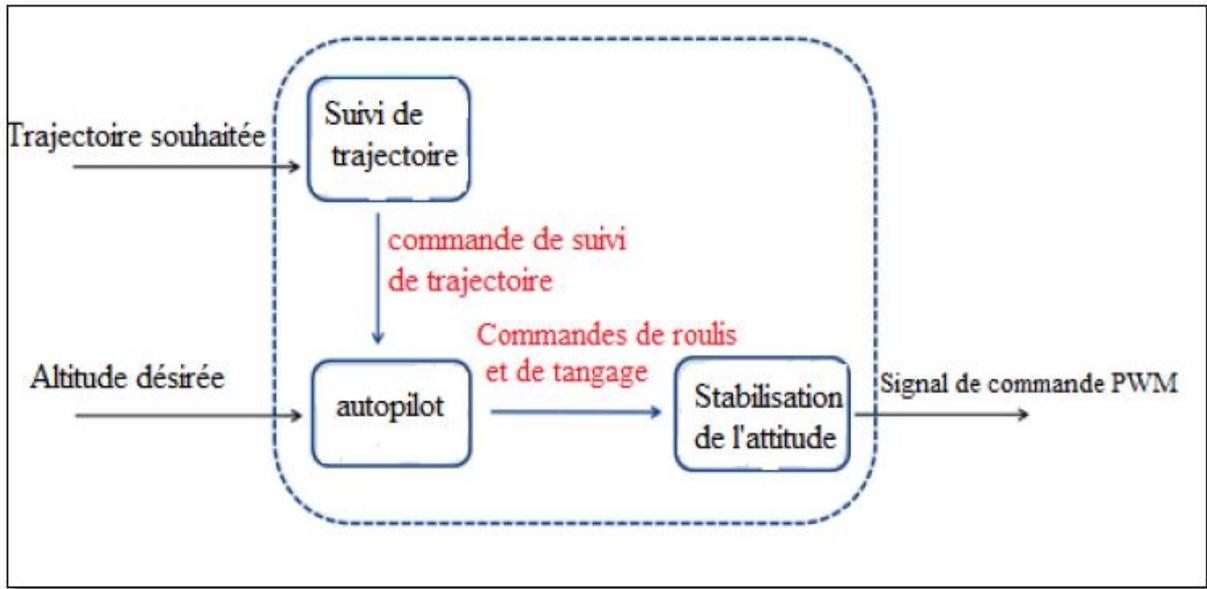


Figure. III.3 : Différentes étages de commande [25]

III.3.1 Guidage par champ de vecteurs :

Dans un guidage par champ de vecteurs, les trajectoires choisies sont construites par des segments linéaires définis par un ensemble des ‘waypoints’ de coordonnées X, Y et Z dans le repère terrestre. Afin de simplifier l’algorithme, l’altitude est maintenue constante pendant le vol, ce qui limite le guidage juste en mode latéral de deux dimensions. [25]

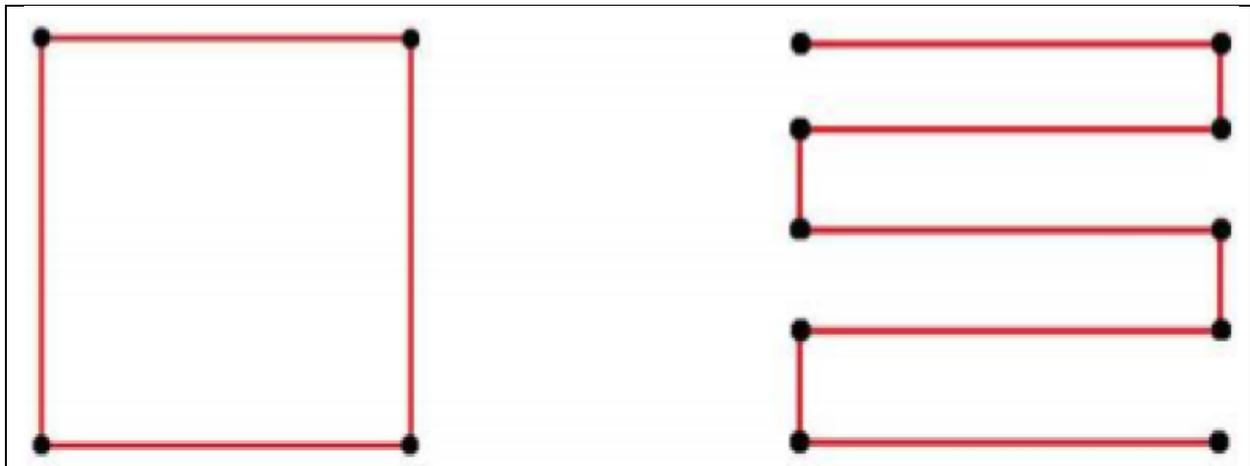


Figure. III.4 : Exemple de segment de trajectoire linéaire.[25]

Pour passer d’un ‘waypoint’ à un autre, le système de guidage impose un angle de heading désiré calculer par la relation suivante : [25]

$$X^f = \tan^{-1} \left(\frac{w_{2y} - w_{1y}}{w_{2x} - w_{1x}} \right) \tag{Eq.III.2}$$

Avec:

X^f : Le heading

$w_{1x,1y}$: Sont les coordonnées X, Y respectivement du waypoint 1.

$w_{2x,2y}$: Sont les coordonnées X, Y respectivement du waypoint 1.

Les coordonnées des waypoints désirés ainsi que les angles de heading sont fournis à l’algorithme de contrôle de position qui génère les angles de roulis et de tangage adéquats en se basant sur le modèle linéaire suivant:

$$\begin{cases} \ddot{x} \approx -\theta \frac{T}{m} \\ \ddot{y} \approx \phi \frac{T}{m} \\ \ddot{z} \approx g - \frac{T}{m} \end{cases} \tag{Eq.III.3}$$

Le contrôleur PID est celui choisi pour la boucle de contrôle de position. La **figure III.5** montre le modèle du contrôleur associé à l’algorithme de guidage. [27]

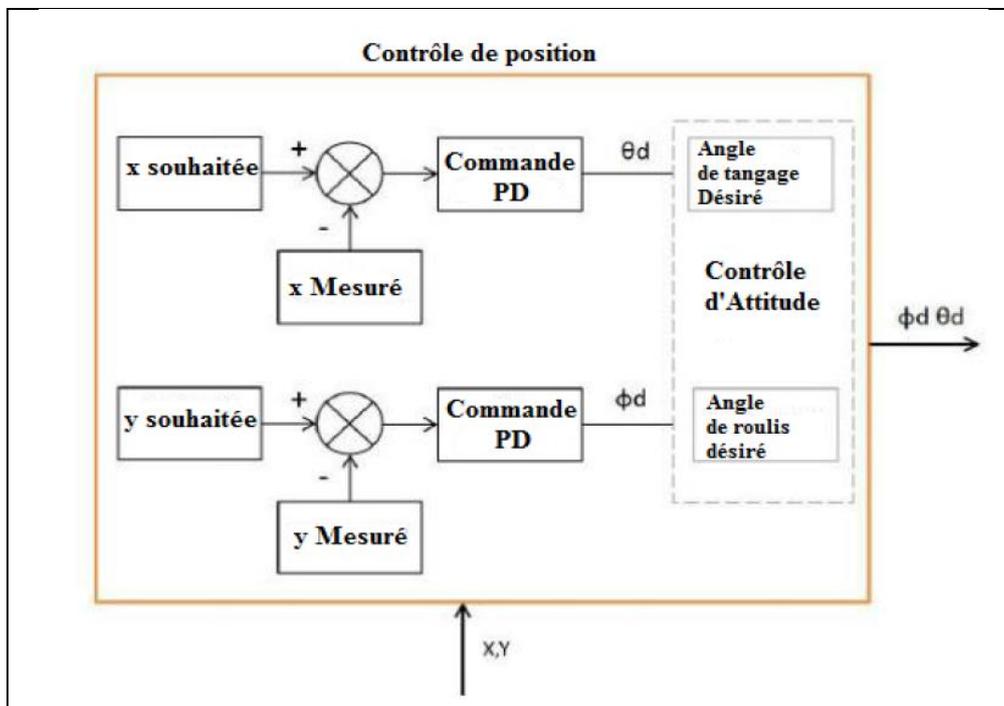


Figure. III.5 : Le modèle du contrôleur associée à l’algorithme de guidage.[27]

❖ Le model Simulink de génération de la trajectoire ainsi que le contrôleur sont présentés dans la Figure.III.6 ou le block « **Path command** » représente la trajectoire de référence créée à travers la fonction sur MATLAB « **timeseries(la position ,t)** »

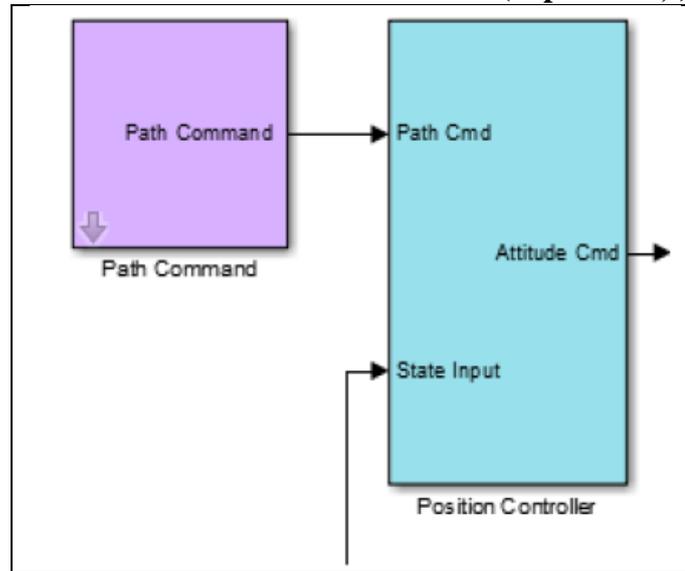


Figure. III.6: Le modèle SIMULINK/Matlab de la génération de la trajectoire ainsi que le contrôleur.

III.3.2: Résultats de simulation de la trajectoire sur SIMULINK/ Matlab

Afin de tester le block « Path command » présenté dans la figure.III.6 et construit par la méthode du guidage par champ de vecteurs, on va simuler notre modèle du quadri-rotor sur Simulink/Matlab et voir si notre modèle suit la trajectoire de référence choisit ou non.

La trajectoire choisie est une trajectoire rectangulaire de dimensions (2*1 m), l'altitude est maintenue à 1 m .la commande avec régulateur PID varie chaque 10s sur un intervalle de simulation égale à 65 s, ce qui donne le temps au quadri-rotor pour se stabiliser.

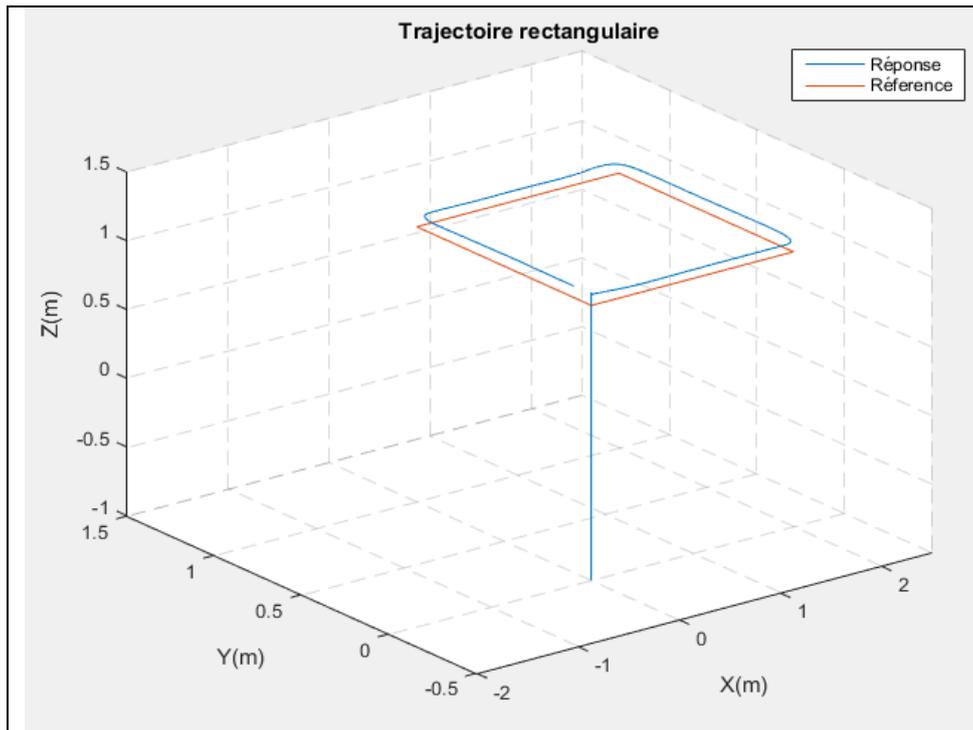


Figure. III.7: Simulation de la trajectoire rectangulaire

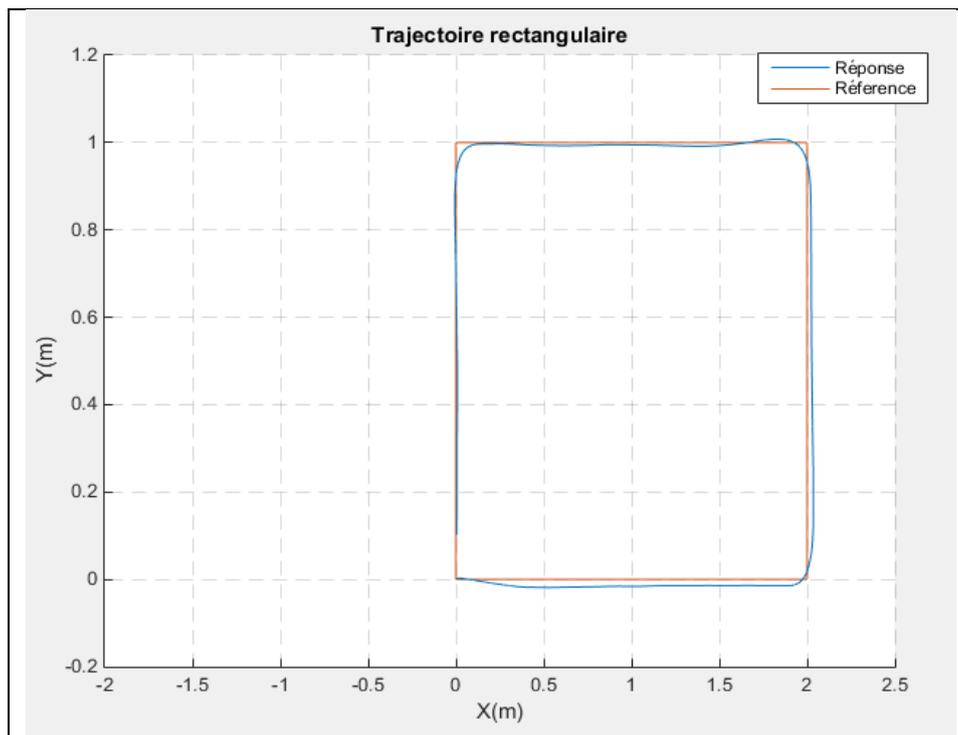


Figure. III.8: Simulation de la trajectoire rectangulaire entre X et Y

D'après la **figure.III.7** et la **figure.III.8** on remarque que le quadri-rotor a un assez bon suivi de la trajectoire imposée, accompagner d'un petit retard à la fin cela peut être justifié par la différence du temps de réponse de l'état Z avec les deux autres états X et Y.

III.4. LA SIMULATION AVEC HARDWARE IN THE LOOP:

La simulation avec matériel dans la boucle, ou souvent appelée « hardware-in-the loop », est un terme couramment utilisé dans le langage scientifique pour définir l'incorporation d'une partie physique dans une boucle de simulation.

Elle consiste soit à remplacer certaines parties de la boucle de simulation par des composants matérielles ou implémenter le contrôleur de vol dans un ordinateur réel. Ainsi, la qualité de la simulation est non seulement améliorée en remplaçant les sous-systèmes dont les modèles sont imprécis ou inconnus par leur équivalent physique, mais aussi cette méthode permet de diagnostiquer à l'avance les problèmes d'intégration et d'explorer les scénarios d'anomalies matérielles et logicielles. [31]

La Figure. III.9 représente le schéma du principe de la simulation Hardware In the Loop entre deux environnements logiciels qui représentent notre modèle Simulink/ Matlab établi précédemment et un environnement matériel qui représente l'ardu-pilot et la simulation en temps réel.

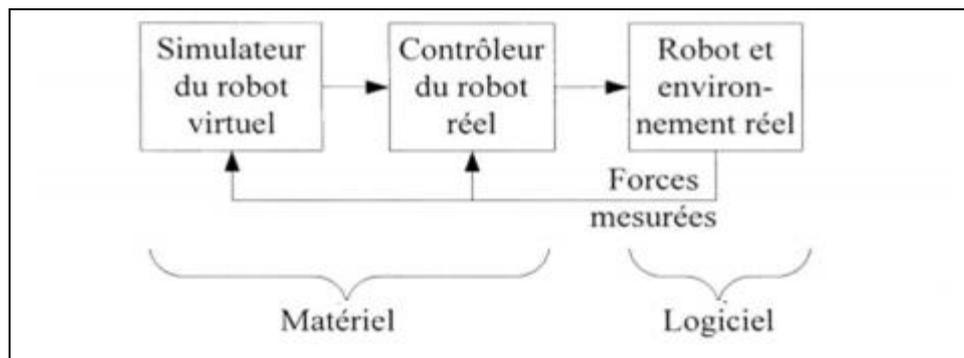


Figure. III.9: Schéma du principe de la simulation Hardware In the Loop.[31]

III.4.1.Simulation sur ordinateur et simulation «Hardware In the Loop»:

Une simulation sur ordinateur est utilisée pour tester les dynamiques principales du système étudié et de sa commande associée. Le système étudié et maintenu dans l'environnement de simulation ainsi que sa commande sont implémentés dans le ordinateur. Mais peuvent fonctionner à des fréquences de calcul différentes. La simulation n'étant composée que du modèle dynamique, toutes les variables d'état transitant entre les deux systèmes ne sont que des signaux numériques gérés par un protocole de communication. L'algorithme de commande peut fonctionner à la même fréquence que l'environnement de simulation ou à une fréquence différente. [31]

III.4.1.1 L'architecture du Hardware In the Loop « HIL »:

Dans le cadre de ce présent travail, la simulation avec le matériel dans la boucle est appliquée à la problématique du prototypage du quadri-rotor. La description de l'architecture de contrôle du quadri-rotor représente la configuration logicielle et matérielle qui permet d'asservir le système. [31]

Dans la simulation avec matériel dans la boucle, deux environnements sont créés : un environnement de simulation qui représente le modèle dynamique du quadri-rotor décrit dans le chapitre1 développé sous le logiciel Matlab/Simulink, et un environnement de lois de commande implémenté dans la carte APM. La liaison entre les deux environnements est assurée à l'aide de protocole de communication RS232.

Afin de faire une simulation Hardware In the Loop et connecter les deux environnements, il faut changer la configuration des paramètres de notre modèle SIMULINK/matlab, il faut :

- Passer de la simulation en pas variable à la simulation en pas fixe [28]. Afin d'éviter les problèmes de synchronisation, la fréquence de fonctionnement du modèle de simulation et du modèle embarqué est fixée à 450 Hz. Ce choix provient du fait que le modèle embarqué est limité par la fréquence de fonctionnement des drivers. Celle-ci est limitée par 450 Hz d'après le **Tableau I.3** dans l'**Annexe 1**.

Le pas est calculé comme suite :

$$T = \frac{1}{f} \quad \text{Eq.III.4}$$

Avec :

f : la fréquence du driver choisit.

T : le pas fixe du modèle SIMULINK.

- Le temps de Simulation sur Simulink/ Matlab [28] : il faut le déterminer de sorte qu'il soit suffisant pour que le quadri-rotor peut suivre la trajectoire demander.

De plus, la mise en œuvre du logiciel de contrôle du quadri-rotor réel nécessite la traduction de ces équations en langage de programmation C/C++ afin que le modèle conçu par le logiciel Simulink/Matlab soit compréhensible par la carte APM. Cette traduction manuelle est souvent longue et devient une source d'erreur non négligeable. Pour éviter ce genre de problèmes, le Toolbox APM conçu par [28] est utilisé. Ce Toolbox présenté précédemment propose des blocks créés sous forme des fonctions embarquées S-fonctions adéquates avec l'encodeur C/C++ de Matlab/Simulink. Une procédure de prototypage rapide a été développée de façon à faciliter l'implémentation des algorithmes GNC.

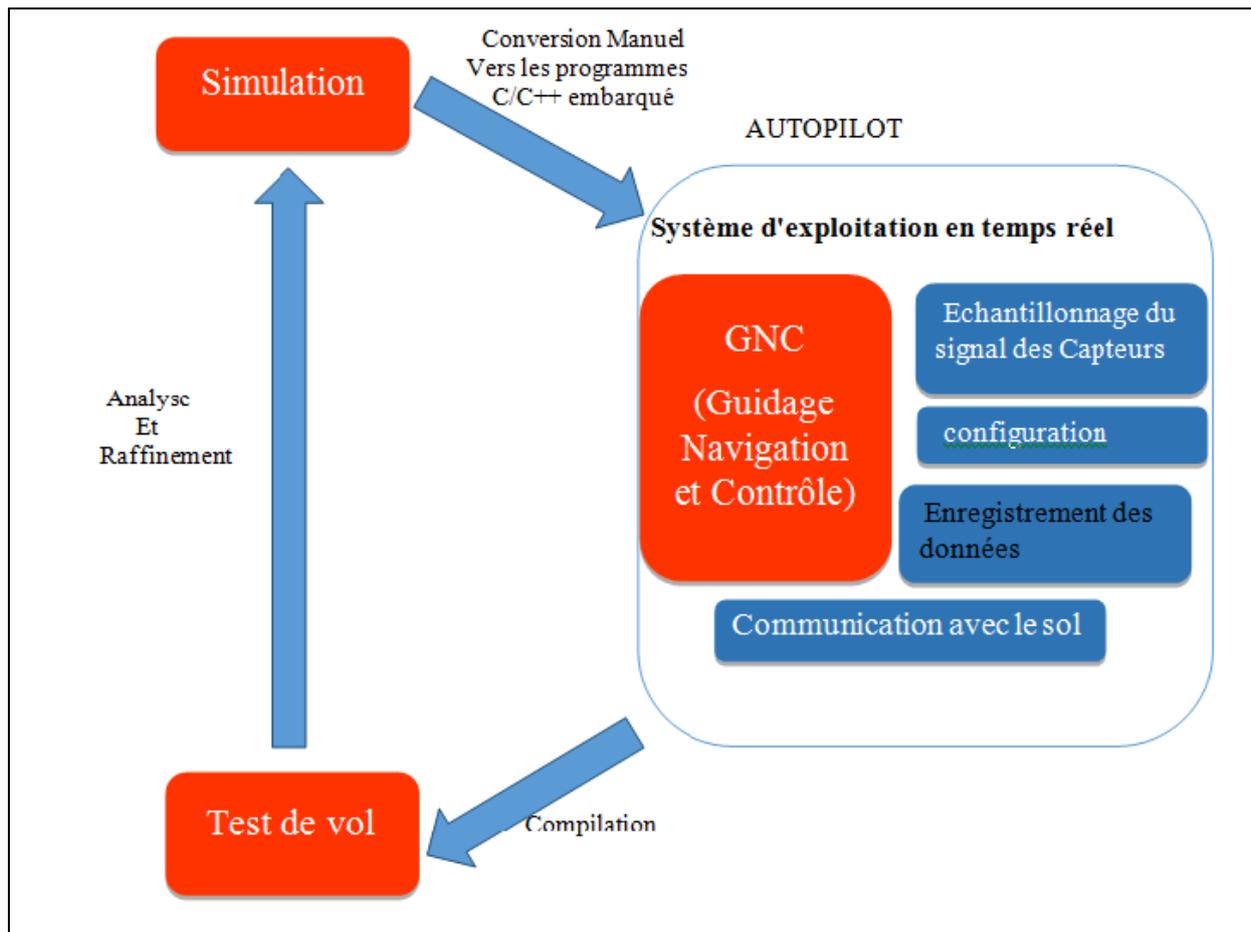


Figure. III.9: Architecture globale de la plateforme expérimentale [31]

III.4.1.2. Chemin de Simulation «HIL»:

Après avoir changé la configuration des paramètres du modèle SIMULINK/Matlab du quadri-rotor, il est temps de passer à l'étape de la simulation en temps réel, le principe de cette simulation est de remplacer la boucle de réglage de position par l'ardu-pilot, ou on va passer d'un régulateur logique à un régulateur numérique afin de suivre la trajectoire de référence.

On aura comme entrée de l'ardu-pilot la trajectoire de position désirée et les positions réelles qui sont les sorties du système et comme sortie les angles roulis et tangage désirés comme le montre la Figure.III.10 qui vont être les entrées du quadri-rotor

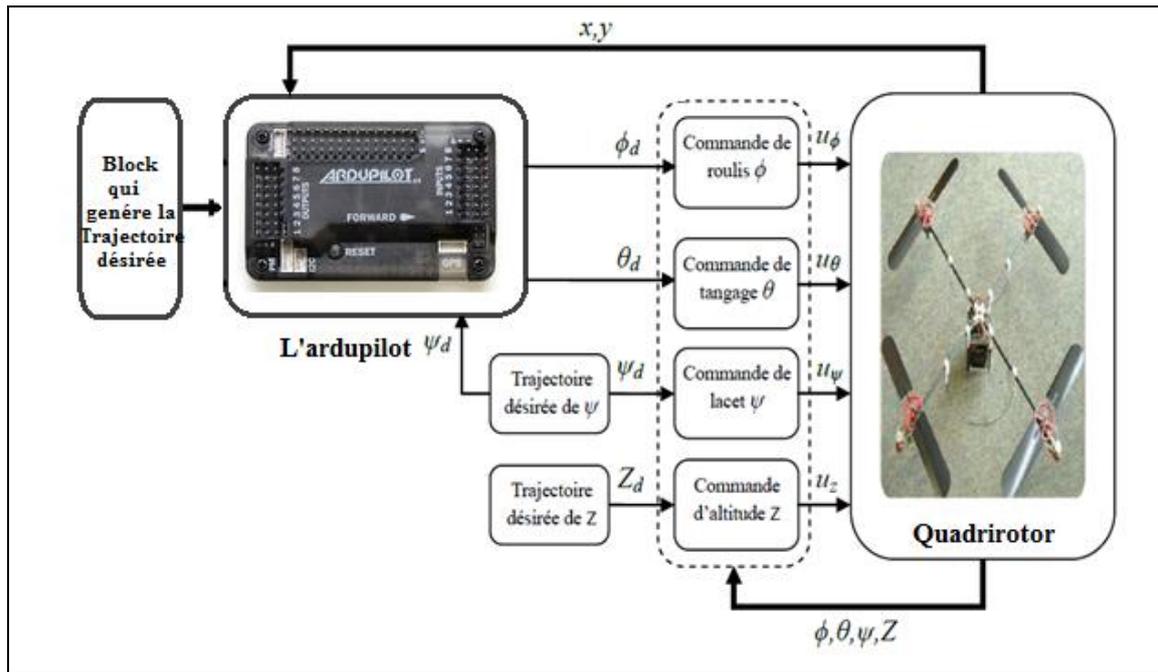


Figure. III.10: Illustration de la structure de commande par Hardware In the Loop

III.5. INTERFACE GRAPHIQUE :

Mission Planner est un logiciel à code source ouvert qui nous donnent la possibilité de créer les WayPoints ainsi que le l'affichage des paramètres de vol. Complet pour le projet de simulation avec l'Ardu-Pilot open source.

La simulation dans mission planner nous permet de visualiser la trajectoire établie précédemment « la trajectoire rectangle » mais dans une station terrestre.



Figure.III.18 : Mission Planner interface graphique

III.6.CONCLUSION:

Dans ce chapitre, une contribution expérimentale indirecte, mais d'une grande utilité, est le développement d'une méthode de prototypage rapide pour la simulation Hardware In the Loop. Cette procédure a permis de générer une trajectoire pour notre quadri-rotor étudié à partir de notre modèle sur Simulink (chapitres I & II) et la méthode du guidage par champ de vecteurs dans un environnement réel de simulation qui peut être testé à l'aide du contrôleur de vol l'Ardu-pilot 2.6.

CONCLUSION GENERALE

L'objectif de ce projet était de concevoir un modèle de simulation qui englobe les algorithmes de guidage, contrôle et de modélisation afin que le quadri-rotor puisse suivre une trajectoire choisie.

BILAN :

On premier lieu le modèle dynamique du quadri-rotor était développé en se basant sur les lois des forces et des moments agissant sur le quadri-rotor qui est traité comme étant un corps rigide à 6 degrés de liberté deux approches pour ce fait ont été traitées : une approche avec les angles d'Euler et une approche avec les quaternions.

Après la modélisation, on a traité la partie la plus importante qui est les lois de commande en assurant la stabilité du quadri-rotor. La commande élaborée est la commande avec régulateurs PID ou leurs gains sont déterminés par la méthode du Loop-shaping. Cette boucle de régulation a été choisie aussi pour l'implémentation réelle à cause de ses réponses adéquates avec le cahier de charge et la simplicité du modèle.

Afin que le quadri-rotor puisse suivre une trajectoire bien déterminée, un algorithme de guidage basé sur le guidage par champ de vecteurs est conçu de telle façon à générer les commandes en position basées sur le régulateur PID.

Les algorithmes de guidage, contrôle et modélisation développés ont été réunis dans le même modèle. Le modèle finale obtenue est subdivisé en deux sous-systèmes : un système qui représente le modèle dynamique dans l'environnement de simulation et un sous système de contrôle embarquée dans une carte électronique. Les deux environnements sont mis dans une simulation Hardware In the Loop.

PERSPECTIVES :

On peut dire que les objectifs posés au début de ce projet sont atteints, mais cela ne nous empêche pas de proposer quelques idées de développement :

- 1- Améliorée le modèle dynamique du quadri-rotor ou les paramètres dynamiques du robot réel pourraient être identifiés de façon plus précise afin de réduire les incertitudes de la loi de commande linéarisante, cela permet de ce rapprocher vers le modèle réelle du quadri-rotor.
- 2- Ajouter un algorithme de navigation, cela permet d'estimer les paramètres de vol avec plus de précision.
- 3- Dans l'algorithme de contrôle on peut proposer l'utilisation des commandes non linéaire qui vont améliorer les réponses et assurer la stabilité et la robustesse. Les commandes non linéaires on éprouver leurs puissance dans plusieurs littératures.
- 4- La limitation major de notre implémentation c'est bien la limitation de la carte APM ; l'utilisation d'une carte plus puissant tel que la PX4 va permit d'implémenter des algorithmes de control plus complexe mais plus aussi précise.

La suite logique de ce travail serait de valider le programme SIMULINK/Matlab établie avec la méthode du Hardware In the Loop étudier théoriquement dans la dernière partie du projet

BIBLIOGRAPHIE :

- [1] KHEBBACHE Hicham, «**Tolérance aux défauts via la méthode backstepping des systèmes non linéaires**», Thèse Magister, UNIVERSITE FERHAT ABBAS DE SETIF, 2012.
- [2] SAM Zine laabidine et BOUKKEBEL Abed, « **commande d'un quadrotor par reseaux de neurones**», Projet fin d'études, Ecole Militaire Polytechnique, 2014.
- [3] S.Bouabdallah, « **Design and Control of Quadrotor with Application to Autonomous Flight** », PHD thesis, EPFL, 2007.
- [4] Mr. AYAD Redouane « *Conception et réalisation d'un drone quadrirotor* » **Thèse de Magister en automatique**, *Université des Sciences et de la Technologie d'Oran*, 2009-2010.
- [5] Øyvind Magnussen et Kjell Eivind Skjønhaug «**Modeling, Design and Experimental Study for a Quadcopter System Construction**», These Master, University of Agder, 2011.
- [6] Tommaso Bresciani« **Modelling, Identification and Control of a Quadrotor Helicopter** » Department of Automatic Control, Lund University, October 2008.
- [7] Oludayo John Oguntoyinbo «**pid control of brushless dc motor and tobot trajectory planning and simulation with matlab/simulink**», Honor Thesis, *VASA University*, 2009.
- [8] Adam Polak « **APM Multicopter Development Kit for Simulink** », Publication, *Arizona State University*, 2013
- [9] Tommaso Bresciani « **Modelling, Identification and Control of a Quadrotor Helicopter** », master thesis, Department of Automatic Control Lund University, 2008.
- [10] Johan Fogelberg « **Navigation and Autonomous Control of a Hexacopter in Indoor Environments** », Msc Thesis , Department of Automatic Control Lund University, 2013.
- [11] Edouard Laroche. « **Identification et Commande Robuste de Systèmes Electromécaniques** ». Automatique/Robotique. Université Louis Pasteur , Strasbourg I, 2007.
- [12] <https://www.mathworks.com/help/robust/ref/ncfsyn.html>
- [13] Alberto Bemporad «**Automatic Control 2 Loop shaping** »University of Trento Academic year 2010-2011
- [14] LAIB Khaled, MAAMRIA Djamaledine,
- [15] Nizar Zein Eddine «**Modélisation et contrôle d'un Quadrirotor "Biritos"**» Projet fin d'études. Université libanaise année académique , Faculté de genie, Département mécanique 2011-2012

- [16] M. Orsag and S. Bogdan. « **Hybrid control of Quadrotor** ». Proceedings of IEEE, 17th Mediterranean Conference on Control and Automation, pp. 1239-1244, Makedonia Palace, Thessaloniki, Greece, 2009.
- [17] **Société DMS - Département STI « Dossier Technique, AR.Drone de Parrot »** Site internet : www.dmseducation.com Email: info@dmseducation.com
- [18] <http://www.moteurindustrie.com/brushless/technique.html>
- [19] Gilles Brocard « Moteur Brushless à rotor externe, (Brushless outrunner) » site web <http://claud.gueniffey.free.fr>
- [20] <http://blog.patrickmodelisme.com/post/qu-est-ce-qu-une-batterie-lipo>
- [21] <https://www.absolu-modelisme.com/variateurs-de-vitesse-et-contrôleurs-brushless>
- [22] <https://fr.wikipedia.org>
- [23] R. Lozano, P. Castillo, S. Salazar, D.Lara « **Stabilisation de véhicules aériens à décollage vertical : Théorie et application** »
- [24] CHERIET Reda, « **Commande et Stabilisation d'Attitude d'un Drone de Type Quadrirotor** », Projet fin d'études, IAB, 2013.
- [25] Adam Ufford « **Development and Implementation of Guidance, Navigation and Control Systems for an Autonomous Air Vehicle** » MASTER OF SCIENCE IN MECHANICAL ENGINEERING, Texas Tech University, 2009.
- [26] Alain Vuille « **Etude et expérimentation de petits quadricoptères destinés à des missions d'observation** », travail de bachelor , Haute école d'ingénierie et de gestion du canton de vaud , 2014.
- [27] Alvin Ng « **Design and Build of Swarm Quadrotor UAVs at UGS** », IMAV2014, University of Glasgow Singapore, 2014.
- [28] Robert Hartley, « **APM2 Simulink Blockset** » MATLAB Central, 13 November, 2012.
- [28] Recherche sur Google : <http://ardupilot.org/copter/docs/common-powering-the-apm2.html>
- [30] <https://www.mathworks.com/matlabcentral/fileexchange/39037-apm2-simulink-blockset>
- [31] HAMELIN, Philippe « **simulation avec materiel dans la boucle d'un robot en interaction avec un environnement réel** », Maitrise en génie de la production automatisée, école de technologie supérieure université du québec, 2008.

Vidéos:

- 1) Loop shaping Control Bootcamp: Introduction to Robust Control

https://www.youtube.com/watch?v=Y6MRgg_TGy0

- 2) Quadcopter Simulation and Control Made Easy - MATLAB and Simulink

<https://www.youtube.com/watch?v=fpJZSQmqDVk>

- 3) Quadcopter/Drone/UAV - Dynamic modelling and trajectory control in MATLAB/Simulink

<https://www.youtube.com/watch?v=72PF4xiRkek>

Logiciels:

- 1) Matlab 2015b.
- 2) Matlab 2014b.
- 3) Mission planner.
- 4) Le périphérique de l'Ardu-pilot.

ANNEXE 1 :

I.PRESENTATION TECHNIQUE DE NOTRE QUADRIROTOR

DJI.450:

La section suivante présente le matériel utilisé dans l'implémentation des algorithmes de commande et de guidage étudié dans les chapitres suivants. Plusieurs littératures telles que [1], [3], [18] ont proposés leurs propres modèles de quadri-rotor, et ils ont détaillé la partie qui concerne la construction. Dans notre projet, la construction du drone est faite sur un logiciel appelée SIMULINK sur matlab à partir de blocks réels de commande représentent les plus indispensables composants du quadri-rotor. Les composants indispensables pour le vol d'un quadri-rotor sont modélisés.

Chaque composant doit être choisi de manière soignée et comporte un certain nombre de défis. La liste qui suit présente les composants indispensables qui se trouvent dans notre programme du quadri-rotor sur SIMULINK.

I.1. Moteur et énergie :

La propulsion de notre quadri-rotor est assurée par quatre moteurs alimentés par une batterie Lithium polymère.

I.1.1.Moteurs brushless :[18]

Un moteur brushless comporte les mêmes éléments qu'un moteur à courant continu, excepté le collecteur, mais l'emplacement des bobines et des aimants permanents sont inversés. Le rotor est composé d'un ou plusieurs aimants permanents, et le stator de plusieurs bobinages.

❖ Avantage : [19]

- Bon rendement : >70% pour les petits, >90% pour les plus gros.
- A puissance égale il tourne moins vite et donne plus de couple qu'un autre moteur électrique. l'hélice peut fonctionner au mieux sans réducteur.
- Absence de balais :
 - pas d'étincelles, donc moins de parasites électriques.
 - Souvent les balais sont les éléments qui se détériorent le plus vite.

❖ Désavantage : [19]

- Absence des balais oblige à intercaler un contrôleur entre la batterie et le moteur. Dispositif complexe, génère une tension triphasé synchrone avec le rotor. Dispositif devenu fiable mais plus onéreux qu'un variateur de moteur classique.

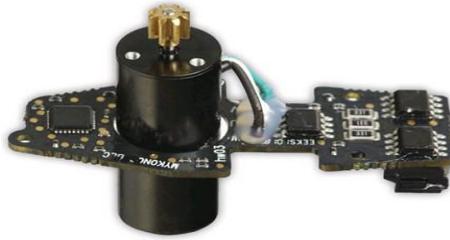


Figure I.1 : Moteur brushless[17]

Paramètre	Valeur
Nombre de moteur brushless	4 moteurs
Vitesse	35 000tr/min
Puissance	15W

Tableau I.1 : Paramètres du moteur brushless utilisé

I.1.2. Batterie Lithium polymère : [20]

Les [batteries LiPo](#) demeurent des accumulateurs électrochimiques dont la réaction est basée sur le lithium à l'état de polymère (l'électrolyte est sous forme de gel) qui sont beaucoup mieux, Comparativement aux "anciens" accumulateurs Ni-Mh, les avantages sont nombreux.

❖ **Avantage :**

- Pas d'effet de PEUKERT qui est lié à la résistance interne de la batterie (plus on décharge fort et plus la chute de tension est importante et là on constate une diminution de puissance) les batteries LiPo n'ont pas de paliers de décharger et conservent ainsi tout leur "Punch" (donc la puissance) au cours de l'utilisation.
- le poids. En effet, ces batteries sont dénuées de tout emballage de métal qui alourdissait fortement les autres type de batteries par exemple celles du Nickel.
- La charge peut atteindre de plus hautes intensités. Il était, en effet, conseillé auparavant de charger une batterie Ni-Mh avec une intensité de charge égale à sa capacité (ce qui donne pour un accumulateur de 1000 mAh à une intensité de charge maximale de 1 A/heure. A présent, les fabricants rapportent qu'on peut charger un accumulateur LiPo jusque 5 fois sa capacité (un accumulateur de 1000 mAh pourra donc être chargé à 5 A/heure!).

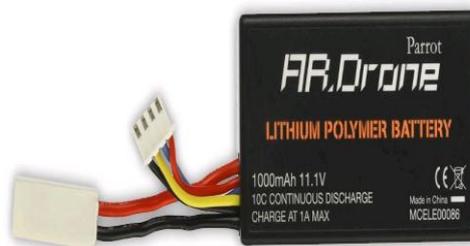


Figure I.2 : Batterie LIPO [17]

Paramètre	Valeur
Nombre de cellule	2 cellules
Tension de charge par cellule	4.2 Voltes
Capacité	1000 mAh
Courant max	5.5 A
Courant des dispositifs auxiliaires	0.1 A
Temps de charge	90 min

Tableau I.2 : Paramètres de la batterie LiPo utilisé

I.1.3.Driver: [21]

Les moteurs brushless fonctionnent en courant triphasé et ont besoins d'être pilotés par un contrôleur pour transformer le courant continu des batteries et les faire tourner. Chaque bobine va être alimentée alternativement pour entraîner la rotation du moteur.

Un contrôleur brushless (tout comme un variateur pour moteur à balais) se choisira en fonction de la consommation du courant (A) et de la tension (V) du moteur qu'il devra faire fonctionner.

❖ Pourquoi utiliser un contrôleur (Driver):

- **Seuil de coupure moteur** : Permet de paramétrer à quelle tension de la batterie le contrôleur va couper l'alimentation du moteur. Cette option permet de préserver les batteries LiPo qui n'aiment pas descendre sous une tension trop basse
- **Coupure moteur** : Soft, médium ou hard sont les valeurs couramment proposées. Un réglage soft ou médium permettra de ne pas ce faire surprendre lorsque la batterie sera vide et de poser son modèle dans de bonnes conditions.
- **Sens de rotation** : Normal ou inversé. Notez qu'on peut inverser le sens de rotation d'un moteur brushless en intervertissant deux des trois fils (n'importe lesquels) qui le raccordent au contrôleur.
- **Fréquence de découpage PWM (Pulse Width Modulation)** : c'est la Fréquence de découpage à laquelle le contrôleur pilote le moteur. On règle donc cette option sur la plus faible valeur permettant un fonctionnement fluide du moteur. Généralement 8kHz. *(cette valeur va nous servir à contrôler en pas fixe notre programme SIMULINK dans la chapitre III).*
- **Régime constant** : Le contrôleur maintient un régime constant quel que soit la résistance appliquée à l'hélice. Surtout utilisé sur les hélicoptères.



Figure I.3 : Moteur brushless[17]

Paramètre	Valeur
30 Hz ~ 450	30 Hz ~ 450Hz

Tableau I.3 : Paramètres du driver utilisé

I.2. Centrale Inertielle :

Le quadri-rotor utilisé embarque un altimètre ultrason et une centrale inertielle basée sur des accéléromètres et des gyroscopes qui lui permet de se stabiliser (vol stationnaire) et de contrôler finement ses déplacements.

I.2.1. Altimètre Ultrason : [22]

Un altimètre ultrason est un [instrument de mesure](#) permettant de déterminer la [distance](#) verticale entre un point et une surface de référence, à l'aide de l'imitation et la réflexion avec le sol d'une onde de même nature que les ondes sonores, mais dont la gamme de fréquence se situe entre 20 kilohertz et plusieurs centaines de mégahertz. Cette gamme est trop élevée pour que l'oreille humaine puisse la percevoir.

Paramètre	Valeur
La fréquence d'émission	40khz
Portée	6m

Tableau I.4 : Paramètres de l'Altimètre Ultrason utilisé



Figure I.4 : Capteur Altimètre Ultrason [17]

I.2.2. Centrale Inertielle : [22]

Une centrale à inertie ou centrale inertielle est un instrument utilisé en navigation, capable d'intégrer les mouvements d'un mobile ([accélération](#) et [vitesse angulaire](#)) pour estimer son orientation (angles de roulis, de tangage et lacet), sa vitesse linéaire et sa position. L'estimation de position est relative au point de départ ou au dernier point de recalage.

Une centrale à inertie comporte six capteurs:

- trois gyromètres mesurant les trois composantes du vecteur vitesse angulaire (vitesses de variation des angles de roulis, de tangage et de lacet).
- trois accéléromètres mesurant les trois composantes du vecteur force spécifique. La force spécifique est la somme des forces extérieures autres que gravitationnelles divisée par la masse. Cette quantité a donc la dimension d'une accélération mais, contrairement à ce que suggère le nom de l'instrument de mesure, il ne s'agit pas exactement d'une accélération.

Le calculateur de la centrale à inertie réalise l'intégration en temps réel, uniquement à partir des mesures de ces six capteurs :

- Des angles d'attitude (roulis, tangage et lacet) ;
- Du vecteur vitesse et de la position.

Paramètre	Valeur
Accéléromètre	3 axes
Gyroscope	2 axes

Tableau I.5 : Paramètres de la Centrale Inertielle utilisé.

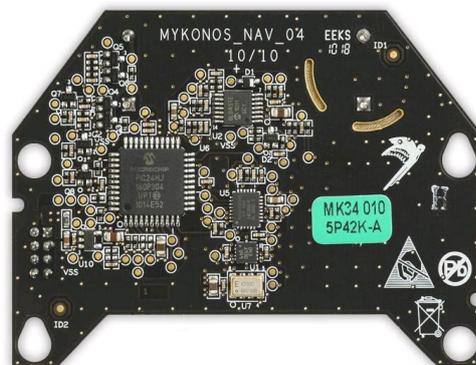


Figure I.5: Centrale Inertielle [17]

I.2. Autres caractéristiques générales :

I.2.1. Les hélices : [22]

L'hélice est un dispositif formé de plusieurs [pales](#) disposées régulièrement autour d'un [axe](#). Lorsque son axe entre en rotation, ce système décrit des hélices dans un fluide en avançant perpendiculairement au plan de rotation, grâce aux pales orientées suivant un certain angle de calage qui prennent appui sur le fluide. Une hélice peut donc être motrice (transfert d'énergie vers le fluide : avion) ou réceptrice (récupération d'énergie depuis le fluide: turbine,

éolienne ou aérogénérateur). Le profil et le calage des pales sont optimisés pour l'utilisation prévue.

Paramètre	Valeur
Diamètre des hélices	0.1397 mètres

Tableau I.6 : Paramètres des hélices utilisé



Figure I.6 : l'hélice du quadri-rotor [17]

I.2.2.Structure :

Les structures disponibles sur le marché résultent d'une série de mutations des premiers quadri-rotors. Elles sont donc plus performantes que les structures maison, notamment quant aux problèmes de vibrations, de résistance et de dynamique de vol.

D'abord en bois ou en aluminium, les modèles sont à présents fabriqués en plastique moulé, permettant des prix très avantageux, ou en fibres de carbone, matériaux léger et résistant. Il existe plusieurs copies des modèles les plus connus. Ces génériques sont sensiblement identiques aux modèles d'origine mais sont constitués de matériaux moins résistants afin de diminuer le coût. [18]

Paramètre	Valeur
Distance entre deux moteurs	0.25 mètres
Matière de fabrication	Fibre de carbone

Tableau I.7 : Paramètres de la structure du quadri-rotor utilisé



Figure I.7 : le support du quadri-rotor [17]



Figure I.8 : Composants constituant le Kit DJI F250 [28]

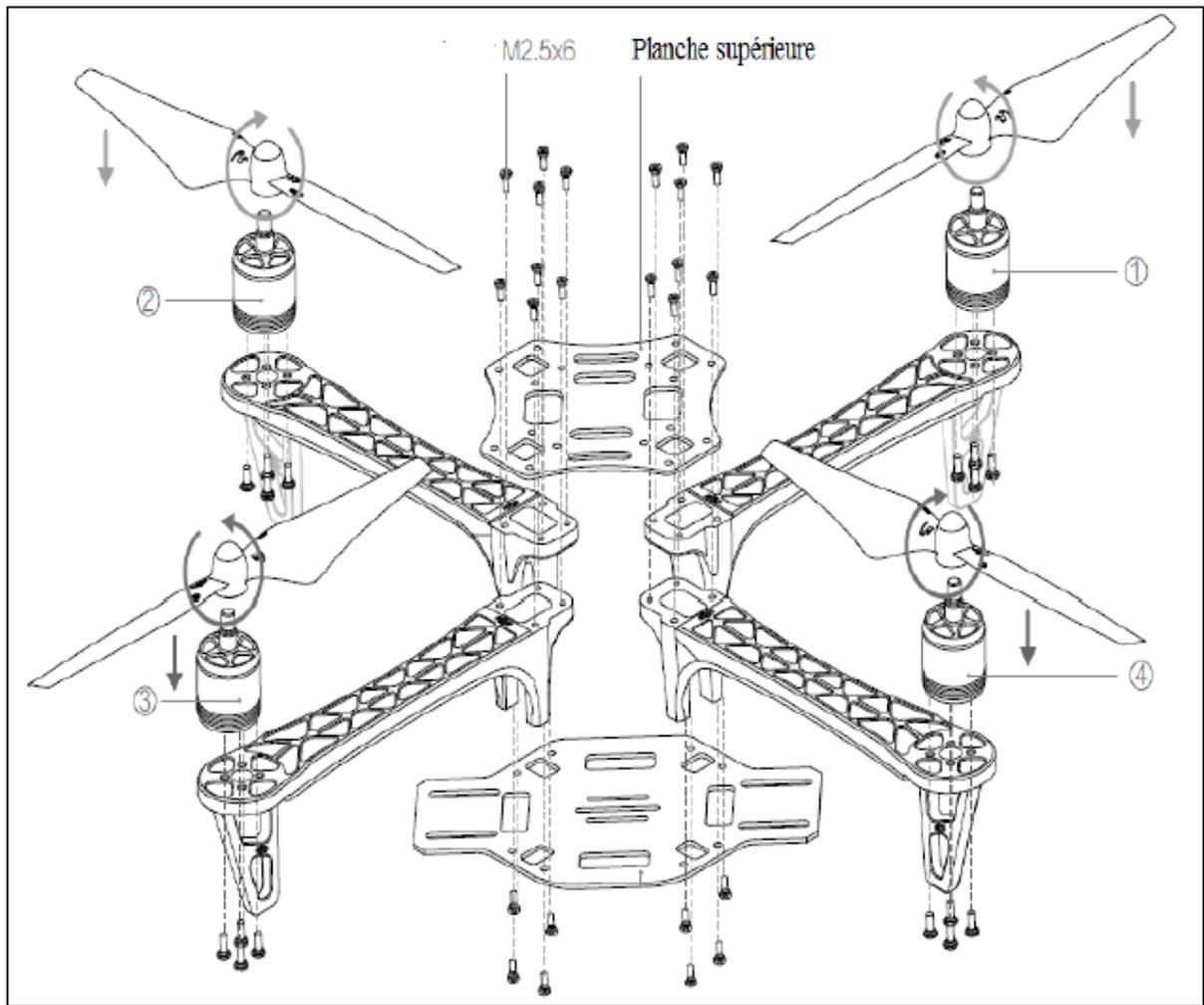


Figure I.9 : assemblage DJI 250 [28]

ANNEXE 2 : Repère fixe et mobile

I.1. Repère local (L) :

Le repère local est un repère orthogonal centré sur le mobile et dont le plan X-Y est tangent à l'ellipsoïde de référence d'où son surnom de plan tangent. Tel qu'illustré à la Figure II.1, les axes X, Y et Z de ce repère sont orientés respectivement selon le nord vrai, l'est et le bas, ce qui explique pourquoi il est également appelé repère dans la littérature. C'est par rapport à ce repère local que l'attitude du mobile est généralement représentée. [8]

ECI: Earth Centered Inertial

NED : North East Down

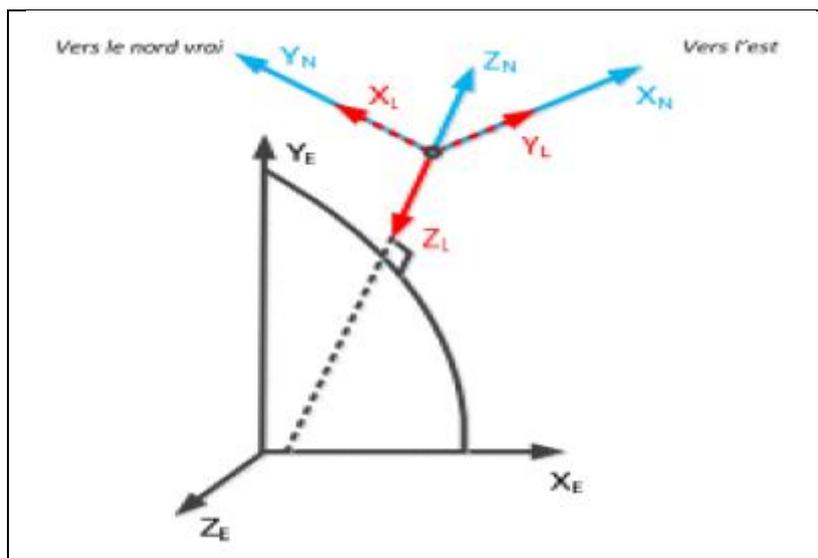


Figure II.1 : Repère NED (en rouge) et ENU (en bleu)

Repère du mobile (B) :

Le repère du mobile (en anglais : body frame) est un repère orthogonal fixé au cadre rigide du véhicule et dont l'origine coïncide généralement avec le centre de masse de ce dernier. Les axes X, Y et Z de ce repère sont orientés respectivement vers l'avant (selon l'axe longitudinal), la droite et le bas du véhicule tel qu'illustré sur la Figure II.2. Les mesures inertielles utilisées par l'algorithme INS sont généralement exprimées selon ce repère. [8]

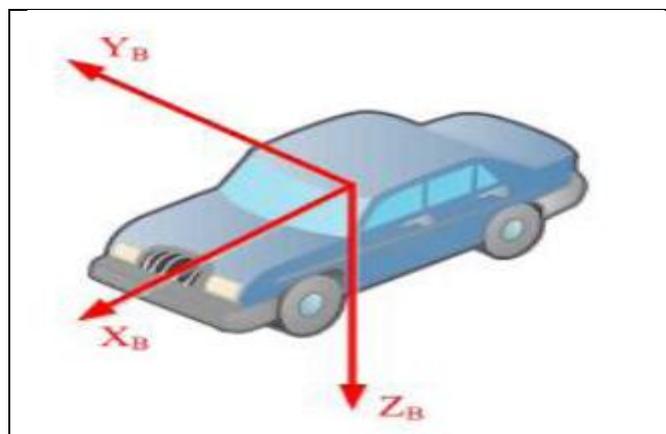


Figure II.2 : Repère du mobile

ANNEXE 3 :

III.1. Vue d'ensemble sur les quaternions

Le quaternion a été découvert par Hamilton, un physicien qui recherchait une prolongation des nombres complexe. La méthode des quaternions est très répandue dans l'infographie et la robotique parce qu'ils peuvent être employés pour commander des rotations dans l'espace tridimensionnel. [2]

Un quaternion peut être considéré comme un nombre complexe avec une partie réel (scalaire) et trois parties imaginaires qui forment un vecteur [49].

$$q = a + bi + cj + dk \quad \text{Eq.III.1}$$

Avec a, b, c, d sont des nombres réels.

On peut écrire un quaternion comme :

$$q = (a, \overline{v}) \quad \text{Eq.III.2}$$

Avec : $\overline{v} = [b \ c \ d]$

III.2. Opérations sur les quaternions

On considère les deux quaternions $q_1 = (a_1, \overline{v_1})$, $q_2 = (a_2, \overline{v_2})$, Les différentes opérations mathématiques dans l'espace des quaternions sont données par : [2]

II.2.1. Addition (soustraction) :

$$q_1 \pm q_2 = (a_1 \pm a_2, \overline{v_1} \pm \overline{v_2}) \quad \text{Eq.III.3}$$

II.2.2. Multiplication :

Le produit en quaternion est noté par:

$$q_1 \times q_2 = (a_1 a_2 - \overline{v_1} \cdot \overline{v_2}), (a_1 \overline{v_2} + a_2 \overline{v_1} + \overline{v_1} \wedge \overline{v_2}) \quad \text{Eq.III.4}$$

II.2.3. multiplication non commutative :

$$q_1 \times q_2 \neq q_2 \times q_1 \quad \text{Eq.III.5}$$

III.3. Rotation dans les quaternions :

On peut aussi écrire un quaternion comme suit : [2]

$$\mathbf{q} = \cos \theta + \hat{\mathbf{U}} \sin \theta \quad \text{Eq.III.6}$$

Ce qui représente une rotation avec angle 2θ autour de l'axe $\hat{\mathbf{U}}$

Soit $\vec{\mathbf{V}}$ un vecteur, la rotation. $\vec{\mathbf{V}}$ Avec un angle 2θ autour de l'axe $\hat{\mathbf{U}}$ est donnée par :

$$\mathbf{R}(\vec{\mathbf{V}}) = \mathbf{q} \times \mathbf{V} \times \mathbf{q}^* \quad \text{Eq.III.7}$$

Avec : q^* Représente le conjugué de q .

La rotation successive par q est donnée par :

$$\hat{\mathbf{q}} = \mathbf{q} \times \mathbf{q}^* \quad \text{Eq.III.8}$$

Avec:

Les angles d'Euler peuvent être calculés à partir de q par la relation suivante :

$$\begin{cases} q_1 = \cos\left(\frac{\phi}{2}\right) \cos\left(\frac{\theta}{2}\right) \cos\left(\frac{\psi}{2}\right) + \sin\left(\frac{\phi}{2}\right) \sin\left(\frac{\theta}{2}\right) \sin\left(\frac{\psi}{2}\right) \\ q_2 = \cos\left(\frac{\phi}{2}\right) \cos\left(\frac{\theta}{2}\right) \cos\left(\frac{\psi}{2}\right) - \sin\left(\frac{\phi}{2}\right) \sin\left(\frac{\theta}{2}\right) \sin\left(\frac{\psi}{2}\right) \\ q_3 = \cos\left(\frac{\phi}{2}\right) \sin\left(\frac{\theta}{2}\right) \cos\left(\frac{\psi}{2}\right) + \sin\left(\frac{\phi}{2}\right) \cos\left(\frac{\theta}{2}\right) \sin\left(\frac{\psi}{2}\right) \\ q_4 = \sin\left(\frac{\phi}{2}\right) \cos\left(\frac{\theta}{2}\right) \cos\left(\frac{\psi}{2}\right) + \cos\left(\frac{\phi}{2}\right) \sin\left(\frac{\theta}{2}\right) \sin\left(\frac{\psi}{2}\right) \end{cases} \quad \text{Eq.III.9}$$

$$\begin{bmatrix} \phi \\ \theta \\ \psi \end{bmatrix} = \begin{bmatrix} \arctg\left(\frac{2(q_3q_4 + q_1q_2)}{q_1^2 - q_2^2 - q_3^2 - q_4^2}\right) \\ \arcsin(-2(q_2q_4 + q_1q_3)) \\ \arctg\left(\frac{2(q_3q_4 + q_1q_2)}{q_1^2 + q_2^2 - q_3^2 - q_4^2}\right) \end{bmatrix} \quad \text{Eq.III.10}$$

ANNEXE 4 :

Le tuning d'un PID est un processus qui consiste à trouver les valeurs des gains proportionnels, intégrales et dérivés d'un contrôleur PID pour atteindre les performances souhaitées et répondre aux exigences de conception.

Le réglage du contrôleur PID semble facile, mais trouver un ensemble de gains qui garantit la meilleure performance de notre système de contrôle est une tâche complexe. Traditionnellement, les contrôleurs PID sont réglés manuellement ou en utilisant des méthodes basées sur des règles. Les méthodes de réglage manuel sont itératives et nécessitent beaucoup de temps et, si elles sont utilisées sur du matériel, elles peuvent causer des dégâts. Les méthodes basées sur les règles ont également de sérieuses limites: elles ne prennent pas en charge certains types de modèles, tels que les tracés instables, les tracés d'un ordre supérieur ou les tracés qui n'ont pas de temporisation.

Vous pouvez synchroniser automatiquement les contrôleurs PID pour obtenir la conception optimale du système et pour répondre aux exigences de conception, même pour les modèles d'usine, selon lesquels les méthodes traditionnelles basées sur les règles ne peuvent pas bien gérer.

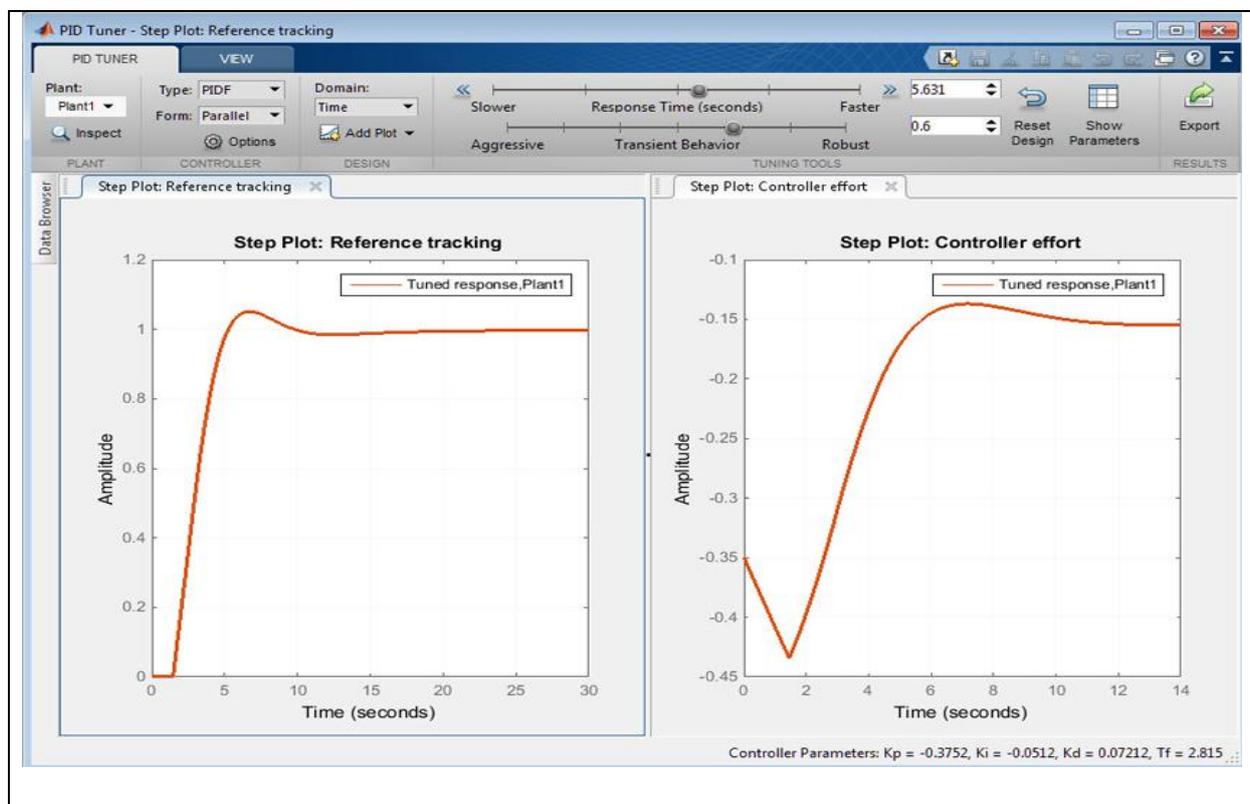


Figure VI.1 : fenêtre du tuning d'un PID sur SIMULINK /Matlab

Afin d'utilisé le tuning MATLAB d'un PID il faut :

- Identification du tracé du modèle à partir des données d'essai entrée-sortie.
- Modélisation des contrôleurs PID dans MATLAB à l'aide d'objets PID ou dans Simulink en utilisant des blocs de contrôleur PID
- Réglage automatique des gains du contrôleur PID et réglage précis de votre conception de manière interactive
- Réglage de plusieurs contrôleurs en mode batch
- Réglage des contrôleurs PID à une seule entrée à une seule sortie ainsi que des architectures PID multi-boucles.