

**REPUBLIQUE ALGERIENNE DEMOCRATIQUE ET POPULAIRE**

**MINISTERE DE L'ENSEIGNEMENT SUPERIEUR**

**ET DE LA RECHERCHE SCIENTIFIQUE**

**UNIVERSITE SAAD DAHLEB DE BLIDA 1**

**Institut d'Aéronautique et des études spatiales**

**Département de construction Aéronautique**



**MEMOIRE DE FIN D'ETUDE**

**EN VUE DE L'OBTENTION DU DIPLOME MASTER EN AÉRONAUTIQUE**

**Option : Avionique**

**Thème :**

***Conception d'un autopilote pour  
un drone léger type quadri-rotor***

**Présenté par :**

- MOHAMMEDI Feriel
- IRBAH Nouria

**Encadré par :**

Mr CHEGRANI  
Mme BENNAOUI

**Promotion : 2018/2019**

## *Dédicace*

*À mes chers parents, aucune dédicace ne saurait exprimer mon respect, mon amour et ma considération pour les sacrifices que vous avez fait pour mon éducation et mon bien être. Je vous remercie pour tout le soutien et l'amour que vous me portez depuis mon enfance et j'espère que votre bénédiction m'accompagne toujours. Que Dieux puisse vous accorder santé, bonheur et longue vie.*

*À mon grand et unique frère Nazim, qui est mon exemple dans la vie et que malgré la distance, il est toujours là pour moi, que dieu te garde pour nous.*

*À la mémoire de mon oncle Lamine, qui m'encourageait toujours pour aller loin dans mes études, mais parti avant de me voir ingénieur, tu seras toujours dans mon cœur khali, que dieu t'accueille dans son vaste paradis.*

*À binôme Nouria, avec qui j'ai passé mes quatre ans et qu'on en a vu de toutes les couleurs, je te souhaite la réussite et le bonheur dans ta vie.*

*À mes ami(e)s que j'ai connu grâce à AIESEC.*

*À la meilleure promo Avionique 2019.*

*À ma familles, mes proches qui me donnent chaque jour du courage et l'amour, que dieu vous bénisse et vous protège.*

*À mes ami(e)s, qui m'ont soutenue tout au long de cette période et à qui je souhaite plus de succès.*

*À toutes les personnes qui ont contribué de près ou de loin.*

***Feriel***

## *Dédicace*

*Avec l'expression de ma reconnaissance, je dédie ce modeste travail à ceux qui, quels que soit les termes embrassés, je n'arriverais jamais à leur exprimer mon amour sincère.*

*À mes parents qui m'ont doté d'une éducation digne, leur amour a fait de moi ce que je suis aujourd'hui.*

*À mes frères et sœurs, qui n'ont pas cessé de me conseiller, encourager et soutenir tout au long de mes études. Que dieu les protège et leurs offre la chance et le bonheur.*

*À mes belles sœurs, pour leurs soutiens moral et mes très chers neveux et nièces qui savent toujours comment procurent la joie et le bonheur pour toute la famille.*

*À la mémoire de ma grand-mère, j'espère que, du monde qui sienne maintenant, elle apprécie cette humble geste. Puisse dieu, le tout puissant, l'avoir en sa sainte miséricorde !*

*À ma très chère Sabrina et toutes mes amies Dihia, Thiziri, Sabria, Faina, Numidia, Amira, Khaoula et Houda avec qui j'ai partagé des moments inoubliables.*

*À tous mes collègues.*

*Sans oublier mon binôme Ferial pour son soutien moral, sa patience et sa compréhension tout au long de ce projet.*

*Nouria*

# **Remerciement**

Nous tenons tout d'abord à remercier Dieu le tout puissant, qui nous a donné la force, la volonté et la patience d'accomplir ce modeste travail.

On adresse nos profonds remerciements et reconnaissances à notre promoteur Mr CHEGRANI Ahmed pour ses précieux conseils, ses orientations, sa confiance et sa patience tout au long de notre travail.

On tient à remercier tout particulièrement Mme BENOUI Nora notre Co-promotrice au Centre de Recherche en Technologie Industrielles, pour son accueil, son attention, sa confiance et sa gentillesse, qui ont fait de ces trois mois un moment très plaisant et intéressant. On est très reconnaissantes pour le temps qu'elle nous a consacré et son aide tout au long de notre expérience, sachant répondre à toutes nos questions et sans oublier son encadrement et sa participation au cheminement de notre travail.

On remercie également le directeur du Centre de Recherche en Technologie Industrielles Mr KECHIDA Ahmed pour l'opportunité et la chance qui grâce à lui nous a été donné pour faire notre stage au sein du centre, sans oublié également toute l'équipe : Billel, Hanna, Hichem, Akram, Madjid, Noureddine, Hafida.

Nous remercions les membres du jury pour l'honneur qu'ils nous ont fait en acceptant d'examiner notre travail.

Ainsi, tous nos remerciements à toutes les personnes qui ont contribué de près ou de loin à l'aboutissement de ce mémoire.

On termine par un grand remerciement à nos parents ainsi qu'à nos familles pour leurs soutiens et leurs aides durant nos études.

## Résumé

Ce mémoire se décompose en deux parties importantes : D'une part, la première partie est consacrée à l'établissement du modèle dynamique du quadrirotor pour étudier sa stabilité et son contrôle sous MATLAB. D'autre part, la deuxième partie est réservée à la réalisation d'une carte autopilote pour le quadrirotor en liaison avec autre carte de développement Arduino MEGA. Cette dernière doit être capable d'acquérir et de traiter les informations issues des capteurs MPU6050 et GPS en temps réel, de commander et de contrôler le quadrirotor. Les résultats obtenus par les capteurs doivent être affichés sur une interface graphique représentant la station sol.

**Mots-clés :** UAV, quadrirotor, autopilote, Arduino MEGA, GPS, MPU6050, station sol, temps réel, MATLAB.

## Abstract

This dissertation is divided into two important parts: the first part is dedicated to the establishment of the dynamic quadrotor model to study its stability and control under MATLAB. The second part is reserved for the realization of an autopilot card for the quadrotor that should be compatible with Arduino MEGA development board. This card must be able to acquire and process information from the MPU6050 and GPS sensors in real time, to order and control the quadrotor. The results obtained by the sensors must be displayed on a graphical interface representing the ground station.

**Keywords:** UAV, quadrotor, autopilot, Arduino MEGA, GPS, MPU6050, ground station, real time, MATLAB.

## ملخص

تنقسم هذه الأطروحة إلى قسمين مهمين: من ناحية، يخصص الجزء الأول لإنشاء نموذج رباعي ديناميكي لدراسة ثباته والتحكم به في MATLAB. من ناحية أخرى، يتم تخصيص الجزء الثاني لتحقيق بطاقة الطيار الآلي للرباعي بالتزامن مع بطاقة تطوير Arduino MEGA. يجب أن تكون البطاقة قادرة على الحصول عن المعلومات من أجهزة الاستشعار MPU6050 و GPS ومعالجتها في الوقت الفعلي، لقيادة وتحكم في الرباعي الديناميكي. يجب عرض النتائج التي حصلت عليها أجهزة الاستشعار على واجهة رسومية تمثل المحطة الأرضية.

**الكلمات المفتاحية:** الطائرات بدون طيار، رباعي ديناميكي، الطيار الآلي، Arduino MEGA، GPS، MPU6050، المحطة الأرضية، في الوقت الفعلي، MATLAB.

## Table des matières

Dédicace .....	02
Remerciement .....	04
Résumé .....	05
Table des matières .....	06
Table des figures .....	09
Liste des abréviations et tableaux .....	13
Nomenclature .....	15
Introduction générale .....	17
<b>CHAPITRE 1 : État d'art sur les drones et les autopilotes. ....</b>	<b>18</b>
1.1 Introduction .....	19
1.2 Classification des drones .....	19
1.1.1 Classification selon P.A.E.....	19
1.1.2 Classification selon la configuration aérodynamique .....	20
1.3 Définition d'un quadrirotor .....	22
1.3.1 Constitution d'un quadrirotor .....	22
1.3.2 Les applications d'un quadrirotor .....	25
1.4 Description du l'autopilote.....	26
1.4.1 Structure d'un autopilote : .....	26
1.4.2 Constitution d'un pilote automatique .....	26
1.4.3 Les Autopilotes existants.....	27
1.4.4 Les pilotes automatiques existants développés par les universités.....	29
<b>CHAPITRE 2 : Généralité, équation du mouvement et modélisation quadrirotor. 31</b>	
2.1 Introduction : .....	32
2.2 Description générale du mouvement d'un quadrirotor .....	32
2.3 Définitions des repères .....	35
2.4 Relation cinématique.....	36
2.4.1 Paramètre cinématique .....	36
2.4.2 Cinématiques de translation.....	37
2.4.3 Cinématique de rotation.....	37
2.5 Modélisation dynamique par la méthode Newton-Euler.....	38
2.5.1 Équation des forces.....	39
2.5.2 Équations des moments .....	39
2.6 Développement du modèle mathématique selon Newton-Euler .....	41
2.6.1 Equation de mouvement de translation .....	42
2.6.2 Equation de mouvement de rotation .....	42

2.7 La représentation d'état du système .....	44
2.8 Conclusion.....	45
<b>CHAPITRE 3 : Équipements et logiciels.....</b>	<b>46</b>
3.1 Introduction .....	47
3.2 Cahier des charges.....	47
3.3 Constitution d'une carte autopilote .....	48
3.3.1 Partie hard.....	48
3.3.2 Partie soft.....	54
3.4 Conclusion.....	57
<b>Chapitre 4 : Simulation sous Matlab .....</b>	<b>59</b>
4.1 Introduction :.....	60
4.2 Résolution numérique du modèle mathématique (équation non linéaire).....	60
4.2.1 Comparaison avec les autres méthodes .....	60
4.2.2 Méthode de Rung Kutta d'ordre 4.....	61
4.2.3 Algorithme du RK4 .....	62
4.2.4 Simulation sous MATLAB.....	62
4.2.5 Résultats de simulation :.....	63
4.3 Résolution numérique du modèle mathématique (équation linéaire).....	68
4.3.1 Linéarisation .....	68
4.3.2 Simulation MATLAB.....	71
4.4 Étude du système.....	72
4.4.1 La stabilité .....	72
4.4.2 La contrôlabilité.....	73
4.4.3 L'observabilité.....	73
4.4.4 Les résultats obtenus :.....	73
4.5 Control du système :.....	74
4.5.1 Placement des pôles :.....	74
4.5.2 Les résultats par MATLAB :.....	75
4.6 Contrôleur PID :.....	81
4.6.1 Définition :.....	81
4.6.2 Description : .....	82
4.6.3 Correcteur PID :.....	82
4.6.4 Présentation schématique : .....	83
4.6.5 La réponse d'un système : .....	84
4.6.6 Présentation du schéma réalisé .....	85

4.6.7 Simulation par MATLAB :	85
4.7 L'organigramme	92
4.8 Conclusion :	92
<b>Chapitre 5 : Test et réalisation</b>	<b>93</b>
5.1 Introduction	94
5.2 Test statique	94
5.2.1 Configuration de GPS	94
5.2.2 Configuration de l'MPU	96
5.2.3 Calibration des moteurs et ESC	97
5.2.4 Test du Bluetooth	98
5.3 Réalisation de la carte d'acquisition	99
5.3.1 Schéma de la carte d'acquisition sur Proteus ARES	99
5.3.2 Vue 3D de la carte	99
5.3.3 Imprimé sur du papier calques	100
5.3.4 Circuit imprimé	100
5.4 Schéma électrique de l'autopilote	101
5.4.1 Assemblage	101
5.5 L'organigramme	102
5.6 Test Dynamique	102
5.7 Test de vol	103
5.8 Conclusion	104
Conclusion générale	104
Bibliographie	105
Annexe	107

## Liste des figures

<b>Figure 1.1.</b> Drones à aile battante	20
<b>Figure 1.2.</b> Drone à voilure fixe	20
<b>Figure 1.3.</b> Drones mono-rotors	21
<b>Figure 1.4.</b> Les différents types de drone birotors	21
<b>Figure 1.5.</b> Drone tri-rotor	22
<b>Figure 1.6.</b> Exemples des drones à rotors multiples	22
<b>Figure 1.7.</b> Batterie utilisé pour l'alimentation d'un quadrirotor	23
<b>Figure 1.8.</b> La carte de contrôleur de vol	23
<b>Figure 1.9.</b> Moteur Brushless	24
<b>Figure 1.10.</b> Contrôleur de vitesse ESC	25
<b>Figure 1.11.</b> Les hélices pour un quadrirotor	25
<b>Figure 1.12.</b> Quadrirotor commercial	25
<b>Figure 1.13.</b> Structure d'un autopilot UAV	26
<b>Figure 1.14.</b> La constitution de l'autopilote	27
<b>Figure 1.15.</b> UNAV 3500	27
<b>Figure 1.16.</b> Procerus Kestrel autopilot	28
<b>Figure 1.17.</b> MP2028xp Autopilote	28
<b>Figure 1.18.</b> Pixhawk PX4	29
<b>Figure 1.19.</b> AggieAir	29
<b>Figure 1.20.</b> contrôleur de l'université Tsinghua	30
<b>Figure 2.1.</b> Description d'un quadrirotor	32
<b>Figure 2.2.</b> Illustration d'un mouvement vertical	33
<b>Figure 2.3.</b> Illustration d'un mouvement de roulis	33
<b>Figure 2.4.</b> Illustration d'un mouvement de tangage	34
<b>Figure 2.5.</b> Illustration d'un mouvement de lacet	34
<b>Figure 2.6.</b> Repères et systèmes d'axes du quadrirotor	35
<b>Figure 2.7.</b> Rotation suivant le roulis, le tangage et le lacet	35
<b>Figure 3.1.</b> Architecture de système à réaliser	47
<b>Figure 3.2.</b> Carte Arduino Méga	48
<b>Figure 3.3.</b> MPU6050	49
<b>Figure 3.4.</b> GPS NEO M6	50

<b>Figure 3.5.</b> Bluetooth HC-05	50
<b>Figure 3.6.</b> Le PCB de la carte d'acquisition	51
<b>Figure 3.7.</b> Le quadrirotor CRTI	51
<b>Figure 3.8.</b> Moteur Brushless	52
<b>Figure 3.9.</b> ESC	52
<b>Figure 3.10.</b> Hélice	52
<b>Figure 3.11.</b> Batterie Lipo 4S	53
<b>Figure 3.12.</b> Récepteur Futaba	53
<b>Figure 3.13.</b> Futaba T8J	53
<b>Figure 3.14.</b> Interface du logiciel IDE	54
<b>Figure 3.15.</b> Interface du logiciel Multiwii combiné avec Arduino	55
<b>Figure 3.16.</b> Interface graphique de Multiwii	55
<b>Figure 3.17.</b> Interface du logiciel u-center	56
<b>Figure 3.18.</b> Interface du logiciel Proteus	57
<b>Figure 4.1.</b> Résolution avec RK4 avec $h=0.1$	63
<b>Figure 4.2.</b> Résolution avec RK4 avec $h=0.5$	64
<b>Figure 4.3.</b> Résolution avec RK4 en changeant les angles	65
<b>Figure 4.4.</b> Résolution avec RK4 pour un mouvement de roulis	66
<b>Figure 4.5.</b> Réponse à un échelon	71
<b>Figure 4.6.</b> Les pôles de la matrice A	73
<b>Figure 4.7.</b> Réponse de la sortie y	74
<b>Figure 4.8.</b> Réponse de la sortie v	75
<b>Figure 4.9.</b> Réponse de la sortie phi	75
<b>Figure 4.10.</b> Réponse de la sortie p	76
<b>Figure 4.11.</b> Réponse de la sortie x	76
<b>Figure 4.12.</b> Réponse de la sortie u	77
<b>Figure 4.13.</b> Réponse de la sortie teta	77
<b>Figure 4.14.</b> Réponse de la sortie q	78
<b>Figure 4.15.</b> Réponse de la sortie psi	78

<b>Figure 4.16.</b> Réponse de la sortie r	79
<b>Figure 4.17.</b> Réponse de la sortie z	79
<b>Figure 4.18.</b> Réponse de la sortie w	80
<b>Figure 4.19.</b> Présentation du régulateur PID	82
<b>Figure 4.20.</b> Caractéristique de la réponse d'un système	83
<b>Figure 4.21.</b> Présentation du schéma réalisé	84
<b>Figure 4.22.</b> Réponse de la sortie y	84
<b>Figure 4.23.</b> Réponse de la sortie v	85
<b>Figure 4.24.</b> Réponse de la sortie phi	85
<b>Figure 4.25.</b> Réponse de la sortie p	86
<b>Figure 4.26.</b> Réponse de la sortie x	86
<b>Figure 4.27.</b> Réponse de la sortie u	87
<b>Figure 4.28.</b> Réponse de la sortie teta	87
<b>Figure 4.29.</b> Réponse de la sortie q	88
<b>Figure 4.30.</b> Réponse de la sortie psi	88
<b>Figure 4.31.</b> Réponse de la sortie r	89
<b>Figure 4.32.</b> Réponse de la sortie z	89
<b>Figure 4.33.</b> Réponse de la sortie w	90
<b>Figure 4.34.</b> Organigramme du programme MATLAB	91
<b>Figure 5.1.</b> Les différentes étapes du branchage du GPS	93
<b>Figure 5.2.</b> Interface u-center après configuration du GPS	94
<b>Figure 5.3.</b> Localisation TIPAZA	94
<b>Figure 5.4.</b> Branchage de MPU	95
<b>Figure 5.5.</b> Choix de la carte et port	95
<b>Figure 5.6.</b> Affichage des données accéléromètre et gyromètre	95
<b>Figure 5.7.</b> Batterie, ESC et moteur	96
<b>Figure 5.8.</b> Calibration des 4 moteurs	96
<b>Figure 5.9.</b> Branchage du Bluetooth	97
<b>Figure 5.10.</b> Choix de la carte et port	97
<b>Figure 5.11.</b> Transmission des messages entre PC et téléphone	97

<b>Figure 5.12.</b> Circuit de la carte d'acquisition sur ARES	98
<b>Figure 5.13.</b> Vue 3D de la carte d'acquisition	98
<b>Figure 5.14.</b> Imprimé du circuit ARES sur du papier calque	99
<b>Figure 5.15.</b> Circuit Imprimé de la carte d'acquisition	99
<b>Figure 5.16.</b> Schéma électrique de l'autopilote sur ISIS	100
<b>Figure 5.17.</b> Le résultat final de la carte autopilote	100
<b>Figure 5.18.</b> Organigramme du programme	101
<b>Figure 5.19.</b> Interface Multiwii	102
<b>Figure 5.20.</b> Calibration de centrale inertielle	102
<b>Figure 5.21.</b> Déverrouillé les moteurs avec la manette	103
<b>Figure 5.22.</b> Verrouillage des moteurs avec la manette	103

## Liste des Tableaux

<b>Tableau 1.1.</b> Classification des UAV	19
<b>Tableau 4.1.</b> Comparaison entre les méthodes	59
<b>Tableau 4.2.</b> Paramètres du quadrirotor	62
<b>Tableau 4.3.</b> Fonctions de transferts.	70
<b>Tableau 4.4.</b> l'influence de l'augmentation des paramètres du PID sur le comportement du système en boucle fermé	83

## Liste des Abréviations

AC	Alternative Current.
CR	Close Range.
CG	Centre de Gravité.
CRTI	Centre de Recherche en Technologie Industrielles.
DC	Direct Current.
DMP	Digital Motion Processor.
EEPROM	Electrically Erasable Programmable Read-Only Memory.
EN	Endurance.
EXO	Exo Start Spheritcal.
ESC	Electronic Speed Control.
EUROUVS	European Association of Unmanned Vehicles Systems.
FCS	Flight Control System.
FIFO	First-in First-out.
GCS	Ground Control Station.
GPS	Global Positioning System.
HALE	High Altitude Long Endurance.
IR	Infra-Red.
I/O	Inputs/Outputs.
ICSP	In Circuit Serial Programming.
IMU	Inertial Measurement Unit.
I2C	Inter Integrated Circuit.
LCD	Liquid Crystal Diode.

LED	Lighting Emitting Diode.
LR	Long Range.
LET	Lethal.
LIPO	Lithium Polymer.
MPU	MIDI Processing Unit.
MR	Medium Range.
MALE	Medium Altitude Long Endurance.
MEMS	Micro-Electro-Mechanical System.
ODE	Ordinary Differential Equation.
PCB	Printed Circuit Board.
PSD	Position Sensitive Device.
PID	Proportional Integral Derivative.
RC	Radio Commande.
RK4	Range Kutta ordre 4
SR	Short Range.
STARTO	Startospheric.
SONAR	Sound Navigation And Ranging.
SPP	Serial Port Protocol.
SFHSS	Slow Frequency Hopping Spread Spectrum.
TTL	Transistor Transistor Logic.
UART	Universal Asynchronous receiver transmitter.
USB	Universal Serial Bus.
UAV	Unmanned Aerial Vehicle.
UHF	Ultra High Frequency.
WIFI	Wireless Fidelity.

## Nomenclature

$R_0$	Repère terrestre.
$R_G$	Repère lié au quadrirotor.
$X_1$	Axe x du repère quadrirotor.
$Y_1$	Axe y du repère quadrirotor.
$Z_1$	Axe z du repère quadrirotor.
$F$	Force du moteur.
$\Phi$	Angle de roulis.
$\theta$	Angle de tangage.
$\Psi$	Angle de lacet.
$R_1$	Matrice de rotation orthogonale d'angle $\phi$ .
$R_2$	Matrice de rotation orthogonale d'angle $\theta$ .
$R_3$	Matrice de rotation orthogonale d'angle $\Psi$ .
$R$	Matrice de rotation.
$r$	Vecteur de position exprimé dans $R_0$ .
$\eta$	Vecteur des angles d'Euler représentant la position exprimé dans $R_0$ .
$v$	Vecteur de la vitesse linéaire exprimé dans $R_G$ .
$\Omega$	Vecteur de la vitesse de rotation instantané entre $R_0$ et $R_G$ exprimé dans $R_G$ .
$u$	Vitesse du quadrirotor selon l'axe $X_1$ .
$v$	Vitesse du quadrirotor selon l'axe $Y_1$ .
$w$	Vitesse du quadrirotor selon l'axe $Z_1$ .
$p$	Vitesse angulaire autour de l'axe $X_1$ .
$q$	Vitesse angulaire autour de l'axe $Y_1$ .
$r$	Vitesse angulaire autour de l'axe $Z_1$ .
$P$	Force de gravité.
$m$	Masse totale du quadrirotor.
$g$	Accélération gravitationnelle.
$F_i$	Force de poussé.
$b$	Coefficient de portance.
$\omega$	Vitesse de rotation de l'hélice.

$T_i$	Force de trainée réduite par l'hélice.
$d$	Coefficient de trainée.
$F_t$	Force de trainée.
$K_{ft}$	Coefficient de trainée de translation.
$M_x$	Rotation autour de l'axe x.
$M_y$	Rotation autour de l'axe y.
$M_z$	Rotation autour de l'axe z.
$l$	Longueur du bras entre rotor et CG.
$K_{fa}$	Coefficient des frottements aérodynamiques.
$M_{gh}$	Moment gyroscopique des hélices.
$J_r$	Inertie des rotors.
$J$	Inertie du système.
$\zeta$	Vecteur de position du quadrirotor.
$S$	Matrice antisymétriques.
$F_f$	Force totale générée par les quatre rotors.
$F_t$	Force de trainée selon les axe x, y, z.
$K_{ft}$	Coefficients de trainée de translation.
$F_g$	Force de gravité.
$M_f$	Moment provoqué par les forces de poussée et trainée.
$M_a$	Moment résultant des frottements aérodynamique.
$K_{fa}$	Coefficients de frottement aérodynamique.
$\Omega_r$	Vitesse résiduelle des moteurs.
$U$	Vecteur de commande.
$X$	Vecteur d'état.
$A$	Matrice d'état.
$B$	Matrice de commande.
$C$	Matrice d'observation.
$D$	Matrice d'action directe.

## Introduction générale

Les drones ou UAV (Unmanned Aerial Vehicle) sont actuellement très utiles dans plusieurs secteurs civils ou militaires, cette utilité encourage les chercheurs beaucoup plus à développer des outils d'aide pour servir les utilisateurs à contrôler et piloter leurs engins volants d'une manière robuste.

La conception d'un autopilote exige une configuration matérielle et logicielle minimale comprenant un ensemble de capteurs pour la détermination de l'attitude, longitude, l'accélération et la position du drone ainsi qu'un microprocesseur embarqué pour le calcul de la commande, le contrôle et la fusion des données. La qualité du vol, la stabilité du système, son autonomie, sa réactivité sont des facteurs déterminants pour le choix des capteurs et des calculateurs embarqués.

Ce projet de fin d'études a pour but principal de réaliser une carte autopilote pour un quadrirotor. Cette phase comprend le choix d'un microprocesseur qui est l'Arduino MEGA qui a pour rôle de commander, contrôler, acquérir et traiter les données des capteurs du central inertielle et GPS. Ces données seront affichées dans une interface graphique.

Pour d'atteindre l'objectif fixé, nous avons suivi un plan de travail composé de cinq chapitres structurés selon l'ordre chronologique suivant :

- Chapitre 1 : Dans ce chapitre, on présente les drones en général puis les quadrirotor en particulier avec leurs classifications et les différentes parties mécaniques et électroniques. Après, on fait une petite description sur le pilote automatique et sa constitution. En fin de chapitre, on va mentionner quelques autopilotes qui existent actuellement.
- Chapitre 2 : Il est consacré à la modélisation dynamique du quadrirotor en se basant sur les équations de mouvement de Newton-Euler. En premier lieu, on présente le fonctionnement du quadrirotor, puis nous définissons les repères de référence nécessaires pour exprimer l'orientation du quadrirotor. Dans un deuxième temps, nous exposons le modèle mathématique complet sous forme d'un système équations différentielles.
- Chapitre 3 : Dans ce chapitre, on s'intéresse à présenter le cahier de charge de notre projet fin d'étude. Ensuite, on définit tous les capteurs et logiciels utilisés pour la réalisation de la carte autopilote.
- Chapitre 4 : Ce chapitre consiste à faire la simulation du comportement du quadrirotor avec MATLAB. Premièrement, une résolution numérique est faite pour les équations non linéaires. Deuxièmement, une résolution et étude de stabilité pour les équations linéaire, une comparaison est faite entre les résultats obtenus.
- Chapitre 5 : Ce dernier chapitre contient tous les tests indispensables à faire pour chaque capteur. Ces capteurs sont assemblés sur une carte d'acquisition qui est compatible avec notre microprocesseur. Un test de vol est fait avec la carte autopilote embarquée sur le quadrirotor.

# CHAPITRE 1 : État d'art sur les drones et les autopilotes.

### 1.1 Introduction

Un drone ou UAV est un aéronef inhabité, inclus dans le domaine de la robotique aérienne, le pilotage de ce genre d'engin se fait d'une manière autonome ou télé piloté. Les drones peuvent effectuer des missions variées, pour des usages civils ou militaires selon son genre : cela repose sur les caractéristiques et les capacités de contrôle de leur vol dans les différents environnements. [1]

Les performances d'un drone se mesurent par sa manière de réagir face aux différents facteurs de l'environnement tel que : les phénomènes aérodynamiques, les perturbations (vents) ... etc. Avec l'évolution de la technologie dans le domaine électronique et les systèmes embarqués, la fabrication des engins de différentes tailles et l'embarquement de plusieurs types de capteurs simultanément sont devenu possibles. Ces derniers permettent de minimiser l'intervention humaine dans les missions complexes et dangereuses.

### 1.2 Classification des drones

Chaque catégorie de drone a un nombre de caractéristiques qui le distinguent, nous pouvons alors proposer une classification des drones en se basant sur différents critères.

#### 1.1.1 Classification selon P.A.E

Une classification intéressante fut proposée par l'European Association of Unmanned Vehicles Systems (EUROUVS). Cette classification est basée sur les paramètres suivants P.A.E c'est-à-dire : le poids maximum au décollage, l'altitude maximale et endurance. [2]

Catégorie	Acronyme	Poids maximum au décollage (kg)	Altitude de vol maximale (km)	Endurance (H)	Exemple
Micro/Mini drones	Mav	0.10	0.25	1	MicroStar
	Mini	<30	0.15-0.3	<2	Mikado
Drones tactiques	Close Range (CR)	150	3	2-4	Phantom
	Short Range (SR)	200	3	3-6	Luna
	Medium Range (MR)	150-500	3-5	6-10	Hunter B
	Long Range (LR)	-	5	6-13	Vigilante 502
	Endurance (EN)	500-1500	5-8	12-24	Aerosonde
	Medim Altitude Long Endurance (MALE)	1000-1500	5-8	24-48	Skyforce

Drones Stratégiques	High Altitude Long Endurance (HALE)	2500-125000	15-20	24-48	Condor
Drones spécialisés	Lethal (LET)	250	3-4	3- 4	Harpy
	Startospheric (STARTO)	A déterminer	20-30	>48	Pegasus
	Exo Start Sphertcal (EXO)	A déterminer	>30	A déterminer	Mac-1

**Tableau 1.1** : Classification des UAV.

### 1.1.2 Classification selon la configuration aérodynamique

La configuration aérodynamique permet une autre possibilité de classification. Les drones peuvent être classés en trois familles :

#### a) Aile battante :

Le guidage et le pilotage de ce genre de drones se font grâce à des ailes battantes. Ils imitent les insectes, ils peuvent aussi faire des vols stationnaires à basse vitesse et effectuées des missions dans des espaces très réduits. [1]



**Figure 1.1** : Drone à aile battante.

#### b) Drone à voilure fixe :

Ce genre de drone est constitué d'une paire d'ailes, un fuselage, un empennage et un ensemble d'hélices. Les configurations de ce genre donnent plus d'autonomie au drone.



**Figure 1.2** : Drone à voilure fixe.

#### c) Drones à voilure tournante

Ce genre de drone a la capacité de décoller, voler et atterrir verticalement, ils utilisent des voilures tournantes ou rotors multiples pour voler en état stationnaire, ils n'ont pas besoin d'une piste de décollage/atterrissage et ils peuvent réaliser une très grande variété de missions irréalisables par rapport aux drones à voilure fixe. [1]

Nous pouvons alors classer les drones à voilures tournantes en quatre catégories :

- Les mono-rotors.
- Les birotors.
- Les tri-rotors.
- les rotors multiples.

### ➤ Les mono-rotors

Les mono-rotors se composent d'un rotor principal qui permet le décollage et l'atterrissage. Ils se divisent en trois sous-catégories selon la configuration utilisée pour permettre au drone de se déplacer dans ses différents axes (en translation et en rotation), ainsi que pour l'empêcher de tourner sur lui-même sous l'action du couple de réaction du rotor principal.

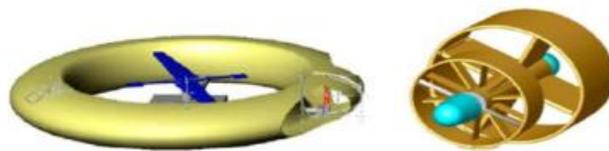


Figure 1.3 : Drones mono-rotors.

### ➤ Les birotors

Les birotors contrarotatifs se composent de deux rotors tournants en sens opposé et à la même vitesse de rotation. Ils sont divisés en 3 sous catégories selon la configuration utilisée pour permettre au drone de se déplacer suivant ses différents axes.

- Les birotors à pas cyclique et collectif : ils utilisent la variation cyclique et collective du pas pour se positionner.
- les birotors augmentés d'une ou plusieurs hélices.
- les birotors utilisant des volets.



Figure 1.4 : Les différents types de drone birotors.

### ➤ Les tri-rotos

Ils sont constitués de deux rotors à l'avant qui tournent dans les sens opposés pour modifier le tangage et d'un rotor en arrière pour régler le roulis. Le fonctionnement est similaire à celui d'un quadrirotor mais la performance en vol n'est pas aussi satisfaisante. [9]



Figure 1.5 : Drone tri-rotor.

### ➤ Les rotors multiples

Ces robots volants sont généralement équipés de quatre rotors, mais ils existent avec six ou même huit rotors



Figure 1.6 : Exemples des drones à rotors multiples.

## 1.3 Définition d'un quadrirotor

Un quadrirotor est un aéronef à voilure tournante comportant quatre rotors pour sa sustentation. Les rotors sont généralement placés aux extrémités d'une croix. Afin d'éviter à l'appareil de tourner sur lui-même sur son axe de lacet, il est nécessaire que deux hélices tournent dans un sens et les deux autres dans l'autre sens. Pour pouvoir diriger l'appareil, il est nécessaire que chaque couple d'hélices tournant dans le même sens soit placé aux extrémités opposées d'une branche de la croix. [4]

### 1.3.1 Constitution d'un quadrirotor

Le drone quadrirotor est constitué de deux parties, une partie électronique (l'électronique embarquée) et une partie mécanique (les propulseurs).

### 1.3.1.1 Partie électronique

#### ➤ Les batteries

Les batteries utilisées sur un quadrirotor sont essentiellement des « Lithium Polymère ». Elles sont issues d'une technologie qui permet d'avoir un très bon rapport poids/puissance. Un élément LiPo (1S) fournit une tension de 3,7v.



**Figure 1.7 :** Batterie utilisé pour l'alimentation d'un quadrirotor.

Sur un drone, on utilise en général des batteries à 3 ou 4 éléments (3S ou 4S). L'intensité est aussi un critère de choix. Un chargeur spécifique est à prévoir ainsi que des règles de sécurité très strictes car ces batteries peuvent exploser en cas de mauvaise manipulation. [5]

#### ➤ Contrôleur de vol (pilote automatique)

C'est une carte électronique, équipée de capteurs très précis, qui va traiter les consignes du pilote envoyées à l'émetteur ainsi que les informations envoyées par ses capteurs et va transmettre des impulsions électriques aux contrôleurs des moteurs pour faire varier leur vitesse. Ces cartes sont équipées de gyroscopes et d'accéléromètres pour mesurer et compenser les déplacements. Certaines cartes évoluées sont équipées de GPS et d'altimètre afin de maintenir un point fixe ou une altitude ou même de retourner au point de départ du drone en cas de perte de signal radio. [5]



**Figure 1.8 :** La carte de contrôleur de vol.

### ➤ Les capteurs

Les capteurs sont fondamentaux pour identifier et calculer l'altitude et la hauteur d'un corps. Grâce à ces capteurs il est possible d'accomplir l'évitement des obstacles et la planification de la trajectoire lors d'un contrôle du haut niveau.

#### a) SONAR

SONAR est un capteur capable de mesurer la distance en utilisant des ondes ultra-son. Dans un quadrirotor on le combine avec le module IR pour estimer la hauteur du quadrirotor par rapport au sol. Il est monté sur la face de dessous et il pointe vers le bas. Pour calculer la distance ( $Z$ ), des données mesurées par ce capteur SONAR doivent être multipliées par une consigne de l'angle de roulis ( $\phi$ ) et l'angle de lacet ( $\theta$ ) mesuré par l'IMU présenté ci-après. [6]

#### b) IMU

IMU est un ensemble de capteurs capables de mesurer l'altitude, l'accélération et l'orientation par rapport un point fixe d'un corps. Il est composé de plusieurs capteurs Accéléromètre, Baromètre et Gyroscope. [7]

#### c) Module IR

IR sert à identifier l'altitude grâce à des ondes lumineuses Avec une certaine plage de fréquence. Ces ondes ne sont pas visibles par l'être humain. Le module ou le système IR est composé d'un IR LED chargé de la conversion d'un signal électrique en onde IR, et un PSD qui est chargé de la conversion inverse [6].

### 1.3.1.2 Partie mécanique

#### ➤ Les moteurs

Quatre moteurs sans balai (brushless) sont nécessaires pour le quadrirotor. Ils existent de différentes tailles et puissances. Ils sont caractérisés par le diamètre de leur cage tournante et par le nombre de tours/volt ou KV. Un moteur ayant un KV de 1000 tours/V fonctionnera à 12000 tours/min s'il est alimenté en 12 V. Sur les moteurs sans balai utilisés en modélisme, les bobinages en cuivre sont montés sur le stator et les aimants sur le rotor, à l'inverse des moteurs électriques conventionnels. [5]



Figure 1.9 : Moteur Brushless.

### ➤ Contrôleur de vitesse

Les ESC convertissent les DC en AC pour les sans balai (brushless) moteur, et ils permettent aussi d'alimenter les moteurs. Chaque moteur a besoin d'un ESC. On peut modifier le firmware d'un ESC pour créer un comportement différent d'un moteur. Par exemple, ESC est souvent configuré pour diminuer la vitesse d'un moteur plutôt de l'arrêter brusquement. [4]



Figure 1.10 : Contrôleur de vitesse ESC.

### ➤ Les hélices

Elles sont souvent en matière plastique mais aussi parfois en matériaux composites. Elles sont identifiées par leur diamètre et leur pas indiqués en pouces. Sur un drone quadrirotor, il faut 2 hélices à sens horaires et 2 hélices à sens antihoraire.

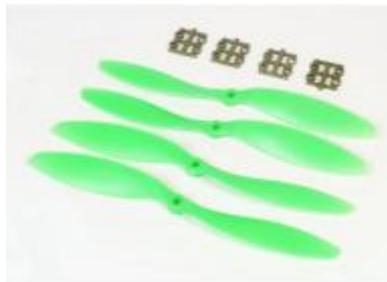


Figure 1.11 : Les hélices pour un quadrirotor.

## 1.3.2 Les applications d'un quadrirotor

Les quadrirotors sont utilisés dans différents domaines parmi eux [9] :

- **La recherche** : quadrirotor est un outil utile pour les chercheurs universitaires pour tester et évaluer de nouvelles idées dans un certain nombre de domaines différents.
- **Surveillance du territoire National** : quadrirotors sont utiles pour la surveillance, la reconnaissance, la recherche et le sauvetage des biens et des personnes.
- **Besoins commerciaux** : la plus grande utilisation de quadrirotor aux Etats-Unis a été dans le domaine de l'imagerie aérienne. Les Quadrirotors sont appropriés pour ce travail en raison de leur caractère autonome et leur coût économique.



Figure 1.12 : Quadrirotor commercial.

### 1.4 Description du l'autopilote

Un pilote automatique est un système utilisé pour guider automatiquement un véhicule sans l'aide d'opérateur humain. Ils ont été largement utilisés dans les véhicules aériens, bateaux, engins spatiaux, missiles et même sur des véhicules terrestres autonomes.

Un pilote automatique d'un UAV doit être capable de le guider constamment par des points de chemin de référence. C'est une partie intégrante dans tout système de contrôle de vol UAV. Le système de commande de vol communique avec le GCS en utilisant la télémétrie ou la station reçoit des données GPS pour la mise à jour de position et envoie des entrées de commande pour les servomoteurs de l'UAV. [10]

#### 1.4.1 Structure d'un autopilote :

Un système de pilote automatique des UAVs est un système de contrôle en boucle fermée qui comporte deux parties : le contrôleur et l'estimateur d'état.

Habituellement, l'estimateur d'état est un système de guidage inertiel qui inclut le gyroscope, l'accélération et les magnétomètres. Il existe d'autres dispositifs de détermination de l'attitude et de vitesse. Les mesures des capteurs et les données GPS sont transmises à un filtre de Kalman qui génère des positions actuelles du véhicule. Ces estimations sont ensuite transmises au contrôleur. Le schéma suivant décrit un système de pilote automatique [11].

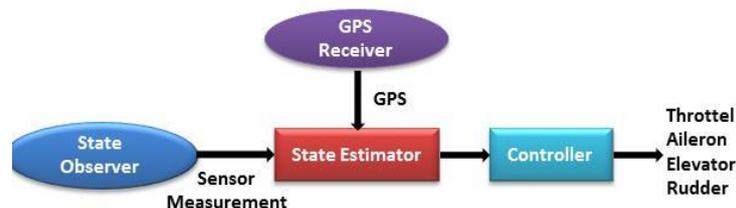


Figure 1.13 : Structure d'un autopilote UAV.

#### 1.4.2 Constitution d'un pilote automatique

En général un système de pilotage automatique pour un véhicule aérien sans pilote, comprend:

- Un processeur.
- Une mémoire en communication électronique avec le processeur.
- Des GPS en communication électronique avec le processeur.
- Trois accéléromètres en communication électronique avec le processeur.
- Trois gyroscopes de taux en communication électronique avec le processeur.
- Un capteur de pression absolue en communication électronique avec le processeur.
- Un capteur de pression différentielle en communication électronique avec le processeur.
- Un émetteur-récepteur en communication électronique avec le processeur pour recevoir et transmettre des signaux sans fil.

La figure suivante nous schématise la constitution de l'autopilote [12] :

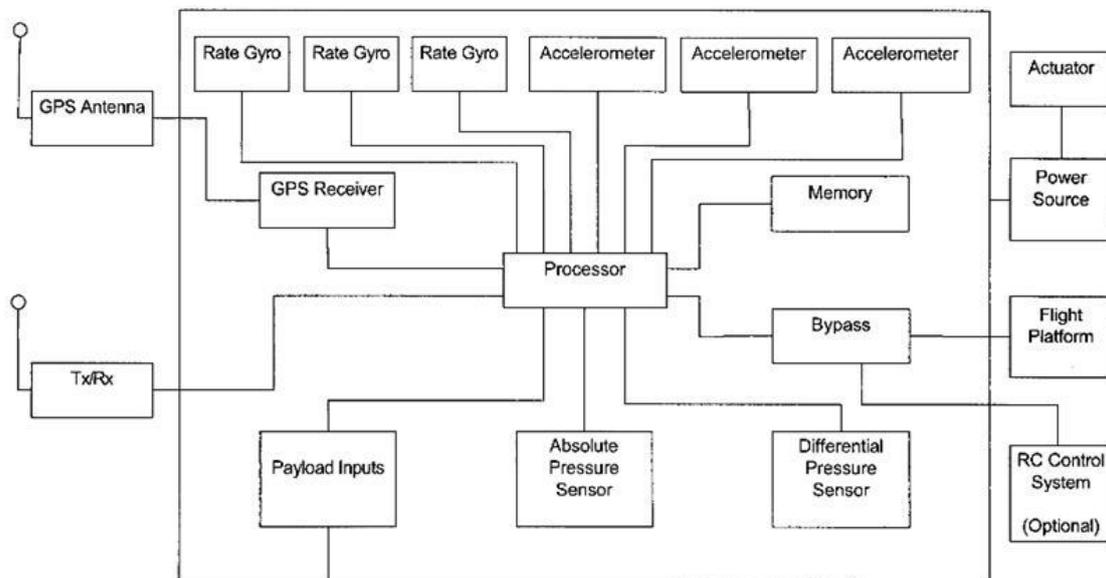


Figure 1.14 : La constitution de l'autopilote.

### 1.4.3 Les Autopilotes existants

#### a) UNAV 3500

Les pilotes automatiques de la société UNAV ont des types spéciaux de pilotes automatiques bon marché pour les débutants d'UAV. Le pilote automatique fournit la fonction basique de navigation autonome avec mode de programmation automatique. Donc le drone peut copier les mêmes waypoints après un vol manuel en mode enregistrement. Société Unav a également 3500 des pilotes automatiques avec un ensemble de capteurs plus complet comprenant Gyromètres à 3 axes, accéléromètres à 2 axes, capteur de pression baromètre et tube de Pitot. Unav 3500 communique avec le récepteur GPS externe via le port série par NMEA Protocole 0183. Unav utilise des gyroscopes électroniques, ce qui pourrait éviter la dérive de gyroscope. Un des avantages du pilote automatique Unav 3500 est sa capacité autonome, ce qui signifie qu'elle n'a pas besoin communication constante avec la station au sol mode autonome. Il prend en charge la navigation par points de cheminement GPS, mode de maintien de l'altitude et de la vitesse avec le gain possible et configuration de la vitesse pour le roulis et tangage. Unav 3500 a deux versions : 3500FW pour les drones à voilure fixe et 3500HL pour l'hélicoptère UAV. [11]



Figure 1.15 : UNAV 3500.

### b) Procerus Kestrel

Le pilote automatique Procerus Kestrel est spécialement conçu pour les petits ou micro-UAV ne pesant que 16,7 grammes (le modem et récepteur GPS non compris).

Kestrel comprend un ensemble complet de capteurs inertiels comprenant : des accéléromètres à 3 axes, Capteurs de vitesse angulaire à 3 axes, magnétomètres à 2 axes, des capteurs de pression absolus et différentiels pour la mesure de l'altitude et la vitesse et 3 capteurs de température combinés avec un algorithme de compensation de température lui d'estimer l'attitude des UAV et vitesse précise de vent. Kestrel a la capacité intégrée pour le décollage, l'atterrissage autonome, point de navigation, vitesse et altitude maintenues et plusieurs opérations d'UAV. [10]

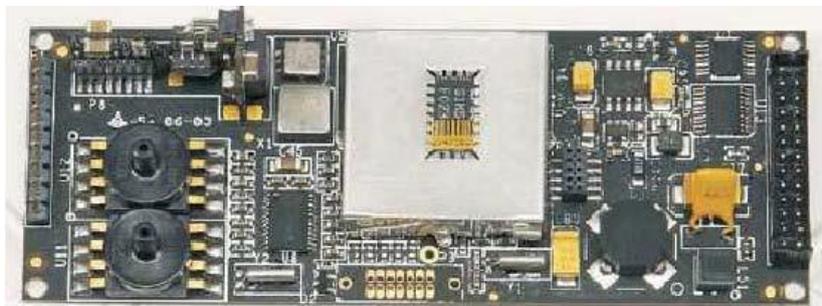


**Figure 1.16** : Procerus Kestrel autopilot

### c) MicroPilot MP 2x28 series

MicroPilot propose une série de pilotes automatiques pour les petits drones à voilure tournante ou à voilure fixe avec une fourchette de prix de 2000 \$ à 8000 \$. Le MP 2028g a un boîtier de capteur similaire à Kestrel sauf qu'il n'a pas de magnétomètre. Le récepteur GPS est intégré dans une seule carte pour MP 2028g, ce qui réduit la taille du pilote automatique, mais les interférences électromagnétiques provenant d'autres circuits peut causer des erreurs de position GPS légèrement plus grandes. Les données de GPS sont mises à jour à 1Hz par défaut.

Le pilote automatique MP 2028g prend en charge le maintien en altitude, maintien de la vitesse et navigation au point de cheminement. Il prend également en charge différents types de décollages et d'atterrissages autonomes y compris le lancement manuel et atterrissage. [11]



**Figure 1.17** : MP2028xp Autopilote.

### d) Pixhawk PX4

Le Pixhawk PX4 est un pilote automatique haut de gamme spécialement conçu par 3D Robotiques. Le microprocesseur avancé est accompagné du capteur de ST Microelectronics ainsi que NuttX, un système d'exploitation en temps réel le tout offrant une performance, une flexibilité et une fiabilité incroyable pour contrôler tout véhicule autonome.

Parmi les avantages de Pixhawk, le multithreading intégré. Cet environnement de programmation identique à Unix/Linux, propose des fonctions du pilote automatique au langage de script pour les missions et comportements de vol, et une couche conducteur PX4 personnalisée garantissant des délais très courts dans tous les processus pour des fonctionnalités avancées illimitées à votre drone. Pixhawk facilite les nouveaux utilisateurs du système à découvrir le pilotage automatique ainsi qu'aux opérateurs d'APM et de PX4. [10]



Figure 1.18 : Pixhawk PX4.

### 1.4.4 Les pilotes automatiques existants développés par les universités

#### a) AggieAir

Le Centre pour les systèmes auto-organisés et intelligents de l'Utah State Université présente une architecture matérielle et logicielle à faible coût, miniature, drones autonomes. Le système de pilote automatique AggieAir est composé de trois modules distincts mais interdépendants ; il s'agit des AggieCap, AggieNav et AggiePilot.

L'AggieCap est responsable de la gestion de la charge utile et du contrôle du système. Il fonctionne sur un ordinateur Gumstix connecté au tableau AggieNav. Il est capable de contrôler un système de caméra panoramique et inclinaison et relayer des images au sol avec une liaison Wifi. AggieNav est une suite de capteurs de navigation qui contient un module IMU 6-DoF, GPS et boussole, ainsi que deux capteurs de pression pour estimation de l'altitude et de la vitesse. AggiePilot est le système de pilote automatique Paparazzi avec l'ajout de l'utilisation du standard de messagerie de commande et de contrôle JAUS. [13]



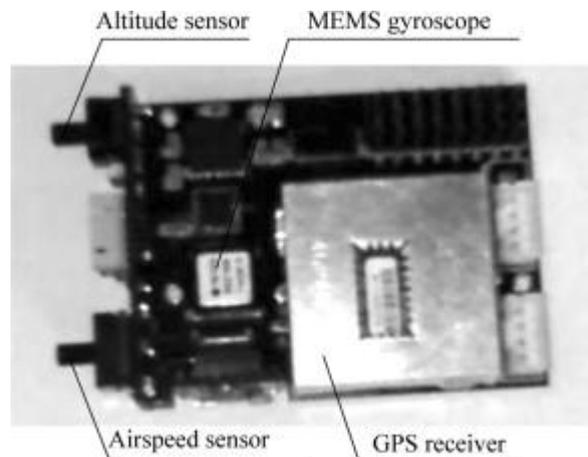
Figure 1.19 : AggieAir.

### b) Tsinghua University TUAUV1000

Le groupe de recherche sur les drones du département des instruments de précision et de la mécanique de l'université de Tsinghua a mis au point un système de contrôle de combat très petit et léger ne pesant que 24 g.

Le système intègre une pression absolue capteur sûr pour les mesures d'altitude et capteur de pression différentielle pour les mesures de vitesse. Un gyroscope à 3 axes est utilisé pour stabiliser le drone. Un récepteur GPS pour les informations de positionnement est directement intégré à la carte système principale.

Le contrôle manuel du véhicule est acheminé via la liaison de données radio fréquence via le serveur de contrôle au sol. Cela élimine le besoin d'une liaison pilote secondaire de sécurité et d'un commutateur, mais empêche le contrôle de l'aéronef en cas de défaillance du système FCS. Les contrôleurs latéraux et longitudinaux utilisent des contrôleurs PID conventionnels avec une planification du gain basée sur la vitesse. Pour le contrôle latéral, un contrôleur de taux de roulis est utilisé. Pour le contrôle longitudinal, des contrôleurs de vitesse de pas et d'accélérateur sont utilisés. [14]



**Figure 1.20 :** contrôleur de l'université Tsinghua.

# CHAPITRE 2 : Généralité, équation du mouvement et modélisation quadrirotor.

### 2.1 Introduction :

Il est nécessaire de comprendre les différents mouvements d'un quadrirotor, sa dynamique et par conséquent ses équations dynamiques pour s'assurer que les simulations du comportement des quadrirotors sont plus proches que possible de la réalité lorsque la commande est appliquée, et pour cela on doit faire une modélisation.

Réaliser une modélisation signifie avant tout chercher à comprendre ce qui se passe, ne pas se contenter d'une solution empirique. La modélisation regroupe un ensemble des techniques permettant de disposer d'une représentation mathématique du système à étudier. Elle requiert une connaissance précise des phénomènes intervenant dans le système et une aptitude à les représenter par des équations mathématiques.

### 2.2 Description générale du mouvement d'un quadrirotor

Le fonctionnement d'un quad-copter est assez particulier. En faisant varier astucieusement la puissance des moteurs, il est possible de le faire monter ou descendre, de l'incliner à gauche ou droite (roulis), en avant ou en arrière (tangage) ou encore de le faire pivoter sur lui-même (lacet), le quadrirotor 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. [1]

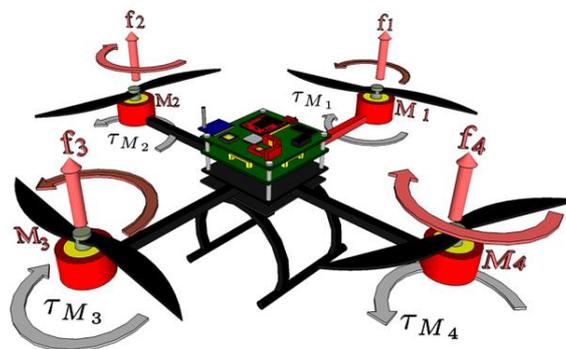


Figure 2.1 : Description d'un quadrirotor.

Les mouvements d'un quadrirotor sont réalisés en variant la vitesse de chaque rotor, de ce fait la poussée est produite. Le quadrirotor incline vers la direction du rotor plus lent, qui tient compte alors de la translation le long de cet axe. Par conséquent, le quadrirotor ne peut pas réaliser la translation sans roulis ou tangage, ce qui signifie qu'un changement de la vitesse d'un rotor se traduit dans un mouvement qui présente au moins trois degrés de liberté. Par exemple, l'augmentation de la vitesse du moteur gauche aura comme conséquence un mouvement de roulis. Nous pouvons commander les six degrés de liberté de quadrirotor avec seulement quatre commandes (le couple appliqué par les moteurs sur chaque propulseur). [1]

Le quadrirotor a quatre mouvements principaux :

- Mouvement vertical.
- Mouvement de roulis.
- Mouvement de tangage.
- Mouvement de lacet.

### a) Mouvement vertical

Afin de planer, toute la force de portance devrait seulement être le long de l'axe  $z$  avec une grandeur exactement opposée à la force de pesanteur. D'ailleurs, la force de portance créée par chaque rotor doit être égale pour empêcher le véhicule de s'incliner. Par conséquent, la poussée produite par chaque rotor doit être identique.

Le mouvement ascendant et descendant est obtenu par la variation de la vitesse de rotation des moteurs, si la force de portance est supérieure au poids du quadrirotor le mouvement est ascendant, et si la force de portance est inférieure au poids du quadrirotor le mouvement est descendant. [9]

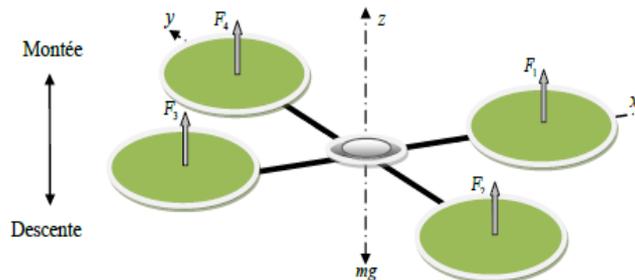


Figure 2.2 : Illustration d'un mouvement vertical.

### b) Mouvement de roulis

Le mouvement de roulis est obtenu en augmentant ou en diminuant la vitesse de l'hélice gauche et en diminuant ou en augmentant la vitesse de l'hélice droite. Elle conduit à un couple qui fait tourner le quadrirotor par rapport à l'axe  $x$ . L'ensemble des poussées verticales est le même que du vol stationnaire, donc cette commande conduit seulement à une accélération angulaire de roulis. [9]

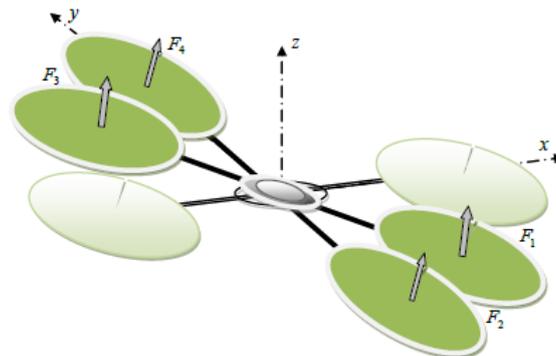


Figure 2.3 : Illustration d'un mouvement de roulis.

### c) Mouvement de tangage

Ce mouvement est très ressemblant à celui du roulis et il est obtenu en augmentant ou en diminuant la vitesse d'hélice arrière et en diminuant ou en augmentant la vitesse d'hélice d'avant. Ce qui conduit à un couple par rapport à l'axe  $y$  qui fait tourner le quadrirotor. [9]

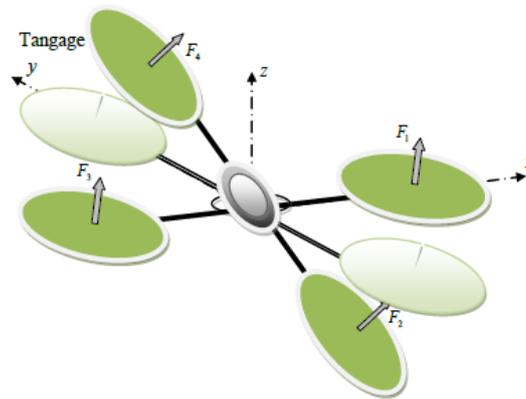


Figure 2.4 : Illustration d'un mouvement de tangage.

### d) Mouvement de lacet

Le mouvement de lacet est obtenu en augmentant ou en diminuant la vitesse des hélices avant-arrière et en diminuant ou en augmentant la vitesse des hélices gauche-droit. Ce qui conduit à un couple qui a un rapport avec l'axe  $z$  qui fait tourner le quadrirotor. Le mouvement de lacet est produit grâce au fait que les hélices gauche-droit tournent dans le sens horaire tandis que les hélices avant-arrière tournent dans le sens antihoraire. Ainsi, lorsque le couple global est déséquilibré, le quadrirotor tourne sur lui-même autour de  $z$ . Cette commande elle conduit seulement à une accélération de l'angle de lacet. [9]

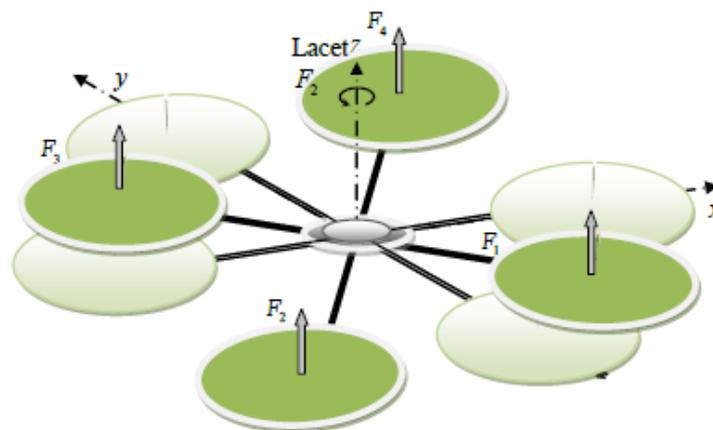
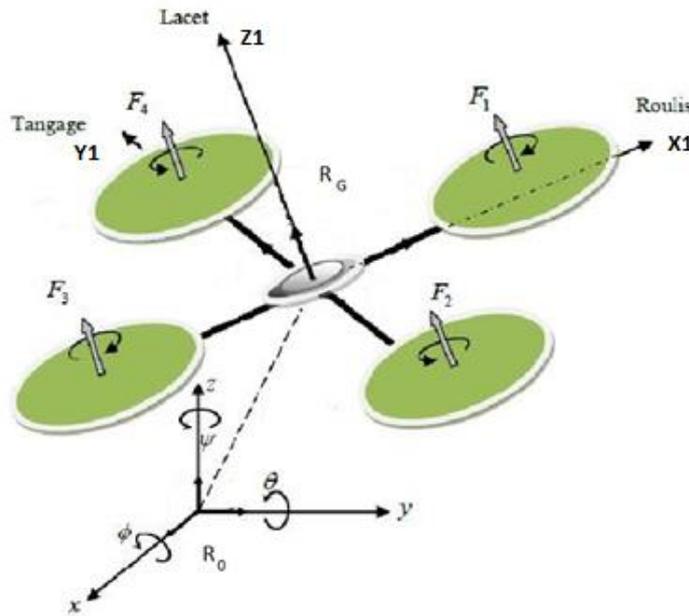


Figure 2.5 : Illustration d'un mouvement de lacet.

## 2.3 Définitions des repères

Un quadrirotor nécessite deux trièdres pour le repérer dans l'espace, ces repères sont [1] [17] :

- **Le repère terrestre** : Il est noté  $R_0(0, X, Y, Z)$ . C'est un repère lié à la terre.
- **Le repère lié au corps du quadrirotor** : Le repère lié au corps du quadrirotor, il est noté  $R_G(G, X_1, Y_1, Z_1)$ . C'est un repère dont l'origine  $G$  coïncide avec le centre de gravité du quadrirotor.

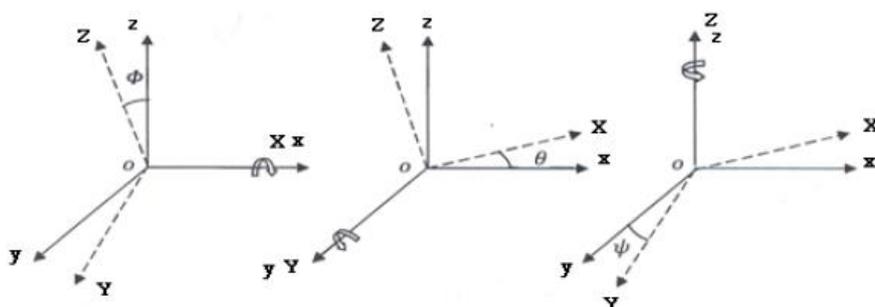


**Figure 2.6 :** Repère et systèmes d'axes du quadrirotor.

On utilise les angles d'Euler ( $\phi, \theta, \psi$ ) qui représentent respectivement le roulis, le tangage et le lacet, afin de déterminer l'orientation au cours de temps du repère quadrirotor par rapport au repère inertiel.

Les angles d'Euler sont définis par la façon suivante :

- Rotation autour de l'axe  $x$  d'un angle  $\phi$  avec :  $-\pi \leq \phi \leq \pi$
- Rotation autour de l'axe  $y$  d'un angle  $\theta$  avec :  $-\frac{\pi}{2} \leq \theta \leq \frac{\pi}{2}$
- Rotation autour de l'axe  $z$  d'un angle  $\psi$  avec :  $-\pi \leq \psi \leq \pi$



**Figure 2.7 :** Rotation suivant le roulis, le tangage et le lacet.

Les matrices relatives à ces transformations sont :

- $R_1$  est une matrice de rotation orthogonale d'angle  $\phi$  (roulis) autour de l'axe «  $x$  ».

$$R_1 = \begin{bmatrix} 1 & 0 & 0 \\ 0 & \cos\phi & -\sin\phi \\ 0 & \sin\phi & \cos\phi \end{bmatrix} \quad (2.1)$$

- $R_2$  est une matrice de rotation orthogonale d'angle  $\theta$  (tangage) autour de l'axe «  $y$  ».

$$R_2 = \begin{bmatrix} \cos\theta & 0 & \sin\theta \\ 0 & 1 & 0 \\ -\sin\theta & 0 & \cos\theta \end{bmatrix} \quad (2.2)$$

- $R_3$  est une matrice de rotation orthogonale d'angle  $\psi$  (lacet) autour de l'axe «  $z$  ».

$$R_3 = \begin{bmatrix} \cos\psi & -\sin\psi & 0 \\ \sin\psi & \cos\psi & 0 \\ 0 & 0 & 1 \end{bmatrix} \quad (2.3)$$

La multiplication de ces trois rotations nous donne la matrice de passage qui est exprimée par la façon suivante :

$$R(\phi, \theta, \psi) = R_1 \cdot R_2 \cdot R_3 \quad (2.4)$$

$$R(\phi, \theta, \psi) = \begin{bmatrix} 1 & 0 & 0 \\ 0 & \cos\phi & -\sin\phi \\ 0 & \sin\phi & \cos\phi \end{bmatrix} \cdot \begin{bmatrix} \cos\theta & 0 & \sin\theta \\ 0 & 1 & 0 \\ -\sin\theta & 0 & \cos\theta \end{bmatrix} \cdot \begin{bmatrix} \cos\psi & -\sin\psi & 0 \\ \sin\psi & \cos\psi & 0 \\ 0 & 0 & 1 \end{bmatrix} \quad (2.5)$$

$$R(\phi, \theta, \psi) = \begin{bmatrix} \cos\theta \cos\psi & \cos\psi \sin\theta \sin\phi - \sin\psi \cos\phi & \cos\psi \sin\theta \sin\phi + \sin\psi \cos\phi \\ \sin\psi \cos\theta & \sin\psi \sin\theta \sin\phi + \cos\psi \cos\phi & \sin\psi \sin\theta \sin\phi - \cos\psi \cos\phi \\ -\sin\theta & \cos\phi \sin\theta & \cos\phi \cos\theta \end{bmatrix} \quad (2.6)$$

## 2.4 Relation cinématique

### 2.4.1 Paramètre cinématique

Les paramètres cinématique constituent l'état du système. Ce dernier comprend quatre vecteur de trois éléments chacun définissant la position et la vitesse du centre de gravité ainsi que l'orientation et la vitesse angulaire de l'engin. Les paramètres sont donc définis comme suit [9] :

- $r(t) = [x \ y \ z]^T$  est la position du CG par rapport à  $R_0$  et exprimé dans  $R_0$ .
- $\eta(t) = [\phi \ \theta \ \psi]^T$  est le vecteur des angles d'Euler représentant la position angulaire du repère lié au corps du quadrirotor  $R_G$  par rapport à  $R_0$  et exprimé dans  $R_0$ .
- $v(t) = [u \ v \ w]^T$  est le vecteur de la vitesse linéaire du CG par rapport à  $R_0$  et exprimé dans  $R_G$ .
- $\Omega(t) = [p \ q \ r]^T$  est le vecteur vitesse de rotation instantané entre  $R_0$  et  $R_G$  et exprimé dans  $R_G$ .

## 2.4.2 Cinématiques de translation

En définissant la matrice de transformation, on peut trouver la vitesse de quadrirotor dans le trièdre inertiel à la place de trièdre lié au quadrirotor, cette matrice est donnée par :

$$v_1 = (\phi \ \theta \ \psi) v \quad (2.7)$$

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

$$\begin{bmatrix} \dot{x} \\ \dot{y} \\ \dot{z} \end{bmatrix} = R(\phi \ \theta \ \psi) \begin{bmatrix} u \\ v \\ w \end{bmatrix} \quad (2.8)$$

Par conséquent :

$$\begin{bmatrix} \dot{x} \\ \dot{y} \\ \dot{z} \end{bmatrix} = \begin{bmatrix} (\cos \theta \cos \psi) u & (\cos \psi \sin \theta \sin \phi - \sin \psi \cos \phi) v & (\cos \psi \sin \theta \sin \phi + \sin \psi \cos \phi) w \\ (\sin \psi \cos \theta) u & (\sin \psi \sin \theta \sin \phi + \cos \psi \cos \phi) v & (\sin \psi \sin \theta \sin \phi - \cos \psi \cos \phi) w \\ (-\sin \theta) u & (\cos \phi \sin \theta) v & (\cos \phi \cos \theta) w \end{bmatrix} \quad (2.9)$$

## 2.4.3 Cinématique de rotation

La cinématique de rotation exprime la relation entre les angles d'Euler  $\eta(t)$  et les composantes du vecteur de vitesse de rotation instantané  $\Omega(t)$ . Il s'agit donc de projeter les axes  $x_3$ ,  $y_2$  et  $z_1$  dans le repère lié au quadrirotor qui sont respectivement les vecteurs directeurs des angles  $\phi$ ,  $\theta$  et  $\psi$ .

Donc la vitesse de rotation instantanée s'écrit comme suit :

$$\Omega(t) = \dot{\phi} x_3 + \dot{\theta} y_2 + \dot{\psi} z_1 \quad (2.10)$$

Il s'ensuit que :

- Les composantes de  $x_3$  dans  $R_G$  sont les mêmes que dans  $R_G$ , c'est-à-dire :

$$x_3 = x_b = [1 \ 0 \ 0]^T \quad (2.11)$$

- Puisque les vecteurs  $y_2$  et  $y_3$  sont identiques, alors  $y_2 = [1 \ 0 \ 0]_{B_3}^T$  et nous pouvons donc exprimer  $y_2$  dans  $R_G$ , c'est-à-dire :

$$Y_2 = R_3^T \begin{bmatrix} 0 \\ 1 \\ 0 \end{bmatrix}_{B_3} = \begin{bmatrix} 0 \\ \cos \phi \\ -\sin \phi \end{bmatrix} \quad (2.12)$$

- Le vecteur  $z_1$  a pour composantes dans  $R_0$  :  $z_1 = \begin{bmatrix} 0 \\ 0 \\ 1 \end{bmatrix}_{R_0}$  (2.13)

Donc  $R_G$

$$Z_1 = R(\phi \theta \psi) \begin{bmatrix} 0 \\ 0 \\ 1 \end{bmatrix}_{R_0} = \begin{bmatrix} -\sin \theta \\ \sin \phi \cos \theta \\ \cos \phi \cos \theta \end{bmatrix} \quad (2.14)$$

Avec :

$$R(\phi \theta \psi)^{-1} = R(\phi \theta \psi)^T \quad (2.15)$$

On obtient :

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

On définit :

$$Q = \begin{bmatrix} 1 & 0 & -\sin \theta \\ 0 & \cos \phi & \sin \phi \cos \theta \\ 0 & -\sin \phi & \cos \phi \cos \theta \end{bmatrix} \quad (2.17)$$

Alors, la matrice inverse de  $Q$  :

$$Q^{-1} = \begin{bmatrix} 1 & \sin \phi \tan \theta & \cos \phi \tan \theta \\ 0 & \cos \phi & -\sin \phi \\ 0 & -\sin \phi \sec \phi & \cos \phi \sec \theta \end{bmatrix} \quad (2.18)$$

Par conséquent :

$$\begin{bmatrix} \dot{\phi} \\ \dot{\theta} \\ \dot{\psi} \end{bmatrix} = \begin{bmatrix} 1 & \sin \phi \tan \theta & \cos \phi \tan \theta \\ 0 & \cos \phi & -\sin \phi \\ 0 & -\sin \phi \sec \phi & \cos \phi \sec \theta \end{bmatrix} \begin{bmatrix} p \\ q \\ r \end{bmatrix} \quad (2.19)$$

## 2.5 Modélisation dynamique par la méthode Newton-Euler

La modélisation mathématique des robots volants de type quadrirotor est une tâche un peu complexe vu que la dynamique de ce type est non linéaire et pleinement couplée. Afin de développer un modèle dynamique complet du quadrirotor, on considère les hypothèses suivantes [18] :

- La structure mécanique du quadrirotor est supposée rigide et symétrique, ceci donne une matrice diagonale  $J$ .
- Pour pouvoir négliger l'effet de déformation des hélices lors de la rotation, on va les supposer comme étant rigides.

- Le centre de masse du quadrirotor et l'origine du repère lié à la terre se coïncident.
- La force de poussée et la force de traînée sont proportionnelles aux carrés des vitesses de rotation des moteurs.

### 2.5.1 Équation des forces

Les forces agissantes sur le quadrirotor sont [1] :

- **Force de gravité** : il est défini par

$$P = m g \quad (2.20)$$

- **Force de poussé** : Ce sont des forces provoquées par la rotation des moteurs, elles sont perpendiculaires au plan des hélices.

$$F_i = b \omega_i^2 \quad (2.21)$$

Avec :

$i = 1 : 4$  et  $b$  le coefficient de portance.

- **Force de traînée** : c'est la résultante entre la force de pression et la force de frottement visqueux. Dans ce cas, on a deux forces de traînée procédant sur le système :

- La traînée dans les hélices : elle agit sur les pales, en fonction de la densité de l'air, la forme des pales est proportionnelle au carré de la vitesse de rotation de l'hélice. Elle est présentée par la relation suivante :

$$T_i = d \omega_i^2 \quad (2.22)$$

Avec  $d$  coefficient de proportionnalité de traînée.

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

$$F_t = K_{ft} v \quad (2.23)$$

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

### 2.5.2 Équations des moments

Il y a plusieurs moments agissants sur le quadrirotor, ces moments sont dus aux forces de poussée et de traînée et aux effets gyroscopiques. [19]

- **Moment due aux forces de poussée**

La rotation autour de l'axe  $x$  : elle est due au moment créé par la différence entre les forces de portance des rotors 2 et 4.

Ce moment est donné par la relation suivante :

$$M_x = l (F_4 - F_2) = lb (\omega_4^2 - \omega_2^2) \quad (2.24)$$

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 l'expression suivante :

$$M_y = l (F_3 - F_1) = lb (\omega_3^2 - \omega_1^2) \quad (2.25)$$

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

### ➤ Moment due aux forces de traînée des hélices

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

$$M_z = d (\omega_1^2 - \omega_2^2 + \omega_3^2 - \omega_4^2) \quad (2.26)$$

Moment résultant des frottements aérodynamiques, il est donné par :

$$M_a = K_{fa} \Omega^2 \quad (2.27)$$

Avec :

$K_{fa}$  est le coefficient des frottements aérodynamiques.

$\Omega$  est la vitesse angulaire.

### ➤ Effets gyroscopiques

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

Dans notre cas il y a deux moments gyroscopiques, le premier est le moment gyroscopique des hélices, l'autre est le moment gyroscopique dû aux mouvements de quadrirotor.

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

$$M_g = \sum_1^4 \Omega \Lambda J_r [0 \ 0 \ (-1)^{i+1} \ \omega_i]^T \quad (2.28)$$

Avec  $J_r$  est l'inertie des rotors.

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

$$M_g = \Omega \Lambda J \Omega \quad (2.29)$$

Avec  $J$  est l'inertie du système.

## 2.6 Développement du modèle mathématique selon Newton-Euler

En utilisant la formulation de Newton-Euler, les équations sont écrites sous la forme suivante [1], [17] et [19] :

$$\begin{cases} \dot{\zeta} = v \\ m \ddot{\zeta} = F_f + F_t + F_g \\ \dot{R} = RS(\Omega) \\ j\dot{\Omega} = -\Omega \wedge J\Omega + M_f - M_a - M_{gh} \end{cases} \quad (2.30)$$

Avec :

$\zeta$  : le vecteur de position du quadrirotor.

$m$  : la masse totale du quadrirotor.

$\Omega$  : la vitesse angulaire exprimée dans le repère fixe.

$R$  : la matrice de rotation.

$\wedge$  : le produit vectoriel.

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

$$J = \begin{bmatrix} I_x & 0 & 0 \\ 0 & I_y & 0 \\ 0 & 0 & I_z \end{bmatrix} \quad (2.31)$$

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

$$S(\Omega) = \begin{bmatrix} 0 & -\Omega_3 & \Omega_2 \\ \Omega_3 & 0 & -\Omega_1 \\ -\Omega_2 & \Omega_1 & 0 \end{bmatrix} \quad (2.32)$$

$F_f$  : La force totale générée par les quatre rotors, elle est donnée par :

$$F_f = R \times [0 \ 0 \ \sum_{i=1}^4 F_i]^T \quad (2.33)$$

$$F_i = b \omega_i^2 \quad (2.34)$$

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

$$F_t = \begin{bmatrix} -K_{ftx} & 0 & 0 \\ 0 & -k_{fty} & 0 \\ 0 & 0 & -k_{ftz} \end{bmatrix} \dot{\zeta} \quad (2.35)$$

Avec :

$K_{ftx}$ ,  $K_{fty}$  et  $K_{ftz}$  sont les coefficients de traînée de translation.

$F_g$  : La force de gravité, elle est donnée par :

$$F_g = \begin{bmatrix} 0 \\ 0 \\ -mg \end{bmatrix} \quad (2.36)$$

$M_f$  : Le moment provoqué par les forces de poussée et de traînée.

$$M_f = \begin{bmatrix} l(F_4 - F_2) \\ l(F_3 - F_1) \\ d(\omega_1^2 - \omega_2^2 + \omega_3^2 - \omega_4^2) \end{bmatrix} = \begin{bmatrix} lb(\omega_4^2 - \omega_2^2) \\ lb(\omega_3^2 - \omega_1^2) \\ d(\omega_1^2 - \omega_2^2 + \omega_3^2 - \omega_4^2) \end{bmatrix} \quad (2.37)$$

$M_a$  : Le moment résultant des frottements aérodynamiques, il est donnée par :

$$M_a = \begin{bmatrix} K_{fax} \dot{\phi}^2 \\ K_{fay} \dot{\theta}^2 \\ K_{faz} \dot{\psi}^2 \end{bmatrix} \quad (2.38)$$

$K_{fax}$ ,  $K_{fay}$  et  $K_{faz}$  sont les coefficients des frottements aérodynamiques.

### 2.6.1 Equation de mouvement de translation

L'application de la loi de Newton en translation :

$$m \ddot{\zeta} = F_f + F_t + F_g \quad (2.39)$$

On remplace chaque force par sa formule, on trouve :

$$m \begin{bmatrix} \ddot{x} \\ \ddot{y} \\ \ddot{z} \end{bmatrix} = \begin{bmatrix} c\phi c\psi s\theta + s\phi s\psi \\ c\phi s\psi s\theta - s\phi c\psi \\ c\phi c\theta \end{bmatrix} \sum_{i=1}^4 F_i - \begin{bmatrix} K_{ftx} \dot{x} \\ K_{fty} \dot{y} \\ K_{ftz} \dot{z} \end{bmatrix} - \begin{bmatrix} 0 \\ 0 \\ -mg \end{bmatrix} \quad (2.40)$$

Avec :  $c = \cos$  et  $s = \sin$

La projection le long des axes permet d'obtenir les équations différentielles qui définissent le mouvement de translation :

$$\begin{cases} \ddot{x} = \frac{1}{m} (c\phi c\psi s\theta + s\phi s\psi) (\sum_{i=1}^4 F_i) - \frac{K_{ftx}}{m} \dot{x} \\ \ddot{y} = \frac{1}{m} (c\phi s\psi s\theta - s\phi c\psi) (\sum_{i=1}^4 F_i) - \frac{K_{fty}}{m} \dot{y} \\ \ddot{z} = c\phi c\theta (\sum_{i=1}^4 F_i) - \frac{K_{ftz}}{m} \dot{z} - g \end{cases} \quad (2.41)$$

### 2.6.2 Equation de mouvement de rotation

L'application de la loi de Newton en rotation :

$$J\dot{\Omega} = -\Omega \Lambda J\Omega + M_f - M_a - M_{gh} \quad (2.42)$$

On remplace chaque variable par son expression, on obtient la forme suivante :

$$\begin{aligned} \begin{bmatrix} I_x & 0 & 0 \\ 0 & I_y & 0 \\ 0 & 0 & I_z \end{bmatrix} \begin{bmatrix} \ddot{\phi} \\ \ddot{\theta} \\ \ddot{\psi} \end{bmatrix} &= - \begin{bmatrix} \dot{\phi} \\ \dot{\theta} \\ \dot{\psi} \end{bmatrix} \Lambda \begin{bmatrix} I_x & 0 & 0 \\ 0 & I_y & 0 \\ 0 & 0 & I_z \end{bmatrix} \begin{bmatrix} \dot{\phi} \\ \dot{\theta} \\ \dot{\psi} \end{bmatrix} - \begin{bmatrix} K_{fax} \dot{\phi}^2 \\ K_{fay} \dot{\theta}^2 \\ K_{faz} \dot{\psi}^2 \end{bmatrix} \\ &+ \begin{bmatrix} lb(\omega_4^2 - \omega_2^2) \\ lb(\omega_3^2 - \omega_1^2) \\ d(\omega_1^2 - \omega_2^2 + \omega_3^2 - \omega_4^2) \end{bmatrix} - \begin{bmatrix} J_r \Omega_r \dot{\theta} \\ -J_r \Omega_r \dot{\phi} \\ 0 \end{bmatrix} \quad (2.42) \end{aligned}$$

Ce qui conduit aux équations différentielles définissant le mouvement de rotation :

$$\begin{cases} I_x \ddot{\phi} = -\dot{\theta}\dot{\psi}(I_z - I_y) - K_{fax} \dot{\phi}^2 - J_r \Omega_r \dot{\theta} + lb(\omega_4^2 - \omega_2^2) \\ I_y \ddot{\theta} = \dot{\phi}\dot{\psi}(I_z - I_x) - K_{fay} \dot{\theta}^2 + J_r \Omega_r \dot{\phi} + lb(\omega_3^2 - \omega_1^2) \\ I_z \ddot{\psi} = \dot{\phi}\dot{\psi}(I_x - I_y) - K_{faz} \dot{\psi}^2 + d(\omega_1^2 - \omega_2^2 + \omega_3^2 - \omega_4^2) \end{cases} \quad (2.43)$$

Avec :

$$\Omega_r = \omega_1 - \omega_2 + \omega_3 - \omega_4 \quad (2.44)$$

Notant que  $u_1, u_2, u_3$  et  $u_4$  sont les entrées de commande de translation et de rotation selon les axes «x, y, z» du système et qui s'écrivent en fonction des vitesses angulaires du quadrirotor :

$$U = [u_1 \ u_2 \ u_3 \ u_4] \quad (2.45)$$

Où les entrées sont choisis tels que :

- $u_1$  : Commande de puissance.
- $u_2$  : Commande de roulis.
- $u_3$  : Commande de tangage.
- $u_4$  : Commande de lacet.

Avec :

$$\begin{cases} u_1 = b(\omega_1^2 + \omega_2^2 + \omega_3^2 + \omega_4^2) \\ u_2 = lb(\omega_4^2 - \omega_2^2) \\ u_3 = lb(\omega_3^2 - \omega_1^2) \\ u_4 = d(\omega_1^2 - \omega_2^2 + \omega_3^2 - \omega_4^2) \end{cases} \quad (2.46)$$

En regroupant ces équations dans une matrice on aura :

$$\begin{bmatrix} u_1 \\ u_2 \\ u_3 \\ u_4 \end{bmatrix} = \begin{bmatrix} b & b & b & b \\ 0 & -lb & 0 & lb \\ -lb & 0 & lb & 0 \\ d & -d & d & -d \end{bmatrix} \begin{bmatrix} \omega_1^2 \\ \omega_2^2 \\ \omega_3^2 \\ \omega_4^2 \end{bmatrix} \quad (2.47)$$

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

$$\begin{cases} \ddot{\phi} = \frac{(I_y - I_z)}{I_x} \dot{\theta} \dot{\psi} - \frac{J_r}{I_x} \Omega_r \dot{\theta} - \frac{K_{fax}}{I_x} \dot{\phi}^2 + \frac{1}{I_x} u_2 \\ \ddot{\theta} = \frac{(I_z - I_x)}{I_y} \dot{\phi} \dot{\psi} - \frac{J_r}{I_y} \Omega_r \dot{\phi} - \frac{K_{fay}}{I_y} \dot{\theta}^2 + \frac{1}{I_y} u_3 \\ \ddot{\psi} = \frac{(I_x - I_y)}{I_z} \dot{\phi} \dot{\psi} - \frac{K_{faz}}{I_z} \dot{\psi}^2 + \frac{1}{I_z} u_4 \\ \ddot{x} = -\frac{K_{ftx}}{m} \dot{x} + \frac{1}{m} u_x u_1 \\ \ddot{y} = -\frac{K_{fty}}{m} \dot{y} + \frac{1}{m} u_y u_1 \\ \ddot{z} = -\frac{K_{ftz}}{m} \dot{z} - g + \frac{(c\phi c\theta)}{m} u_1 \end{cases} \quad (2.48)$$

Avec :

$$\begin{cases} u_x = (c\phi c\psi s\theta + s\phi s\psi) \\ u_y = (c\phi s\psi s\theta - s\phi c\psi) \end{cases} \quad (2.49)$$

## 2.7 La représentation d'état du système

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

$$\begin{aligned} X &= [\phi \ \dot{\phi} \ \theta \ \dot{\theta} \ \psi \ \dot{\psi} \ x \ \dot{x} \ y \ \dot{y} \ z \ \dot{z}]^T \\ X &= [x_1 \ x_2 \ x_3 \ x_4 \ x_5 \ x_6 \ x_7 \ x_8 \ x_9 \ x_{10} \ x_{11} \ x_{12}]^T \\ X &= [\phi \ p \ \theta \ q \ \psi \ r \ x \ u \ y \ v \ z \ w]^T \end{aligned} \quad (2.50)$$

On obtient la représentation d'état suivante :

$$\left\{ \begin{array}{l} \dot{x}_1 = x_2 \\ \dot{x}_2 = a_1 x_4 x_6 + a_2 x_2^2 + a_3 \Omega_r x_4 + b_1 u_2 \\ \dot{x}_3 = x_4 \\ \dot{x}_4 = a_4 x_2 x_6 + a_5 x_4^2 + a_6 \Omega_r x_2 + b_2 u_3 \\ \dot{x}_5 = x_6 \\ \dot{x}_6 = a_7 x_2 x_4 + a_8 x_6^2 + b_3 u_4 \\ \dot{x}_7 = x_8 \\ \dot{x}_8 = a_9 x_8 + \frac{u_1}{m} u_x u_1 \\ \dot{x}_9 = x_{10} \\ \dot{x}_{10} = a_{10} x_{10} + \frac{u_1}{m} u_y u_1 \\ \dot{x}_{11} = x_{12} \\ \dot{x}_{12} = a_{11} x_{12} + \frac{u_1}{m} (\cos \phi \cos \theta) - g \end{array} \right. \quad (2.51)$$

Avec :

$$\left\{ \begin{array}{l} a_1 = \frac{(I_y - I_z)}{I_x}; a_2 = -\frac{K_{fax}}{I_x}; a_3 = \frac{J_r}{I_x}; a_4 = \frac{(I_z - I_x)}{I_y} \\ a_5 = -\frac{K_{fay}}{I_y}; a_6 = \frac{J_r}{I_y}; a_7 = \frac{(I_x - I_y)}{I_z}; a_8 = -\frac{K_{faz}}{I_z}; \\ a_9 = -\frac{K_{ftx}}{m}; a_{10} = -\frac{K_{fty}}{m}; a_{11} = -\frac{K_{ftz}}{m}. \end{array} \right. \quad (2.52)$$

Et :

$$\left\{ \begin{array}{l} b_1 = \frac{1}{I_x} \\ b_2 = \frac{1}{I_y} \\ b_3 = \frac{1}{I_z} \end{array} \right. \quad (2.53)$$

## 2.8 Conclusion

Dans ce chapitre, nous avons donné une description générale sur le quadrirotor et ses mouvements possibles.

En se basant sur les lois de Newton-Euler, ce chapitre présente une modélisation dynamique du drone quadrirotor. La complexité du modèle est due au fait qu'il est non linéaire et l'interaction entre les états du système sont bien distingués à travers le modèle proposé. En conséquence, le quadrirotor est un système sous actionné.

# CHAPITRE 3 :

## Équipements et logiciels.

### 3.1 Introduction

Dans ce chapitre, on définit le cahier des charges qui se divise en deux parties. La première partie est consacrée à la réalisation d'une carte autopilote qui est subdivisée en deux sections différentes : hard et soft qui sont exploitées dans la phase de réalisation. La deuxième partie est consacrée à la modélisation de quadricoptère qui sera détaillée dans le prochain chapitre.

### 3.2 Cahier des charges

Notre projet consiste à concevoir premièrement une carte autopilote destinée pour les petits UAV à voilure tournante (quadricoptère) en utilisant une carte de développement Arduino MEGA, qui sera le cerveau de notre autopilote (processeur). Cette carte doit être capable d'acquérir et de traiter les informations issues des capteurs : accéléromètre, gyromètre, compas et GPS en temps réel, ainsi de commander et de contrôler le quadricoptère utilisé pour tester la carte autopilote réalisée. Les résultats obtenus par les capteurs doivent être affichés via une interface graphique représentant la station sol et qui seront transmis par le module Bluetooth.

Les capteurs utilisés seront regroupés et implantés dans une carte à réaliser et que sera compatible avec la carte de développement

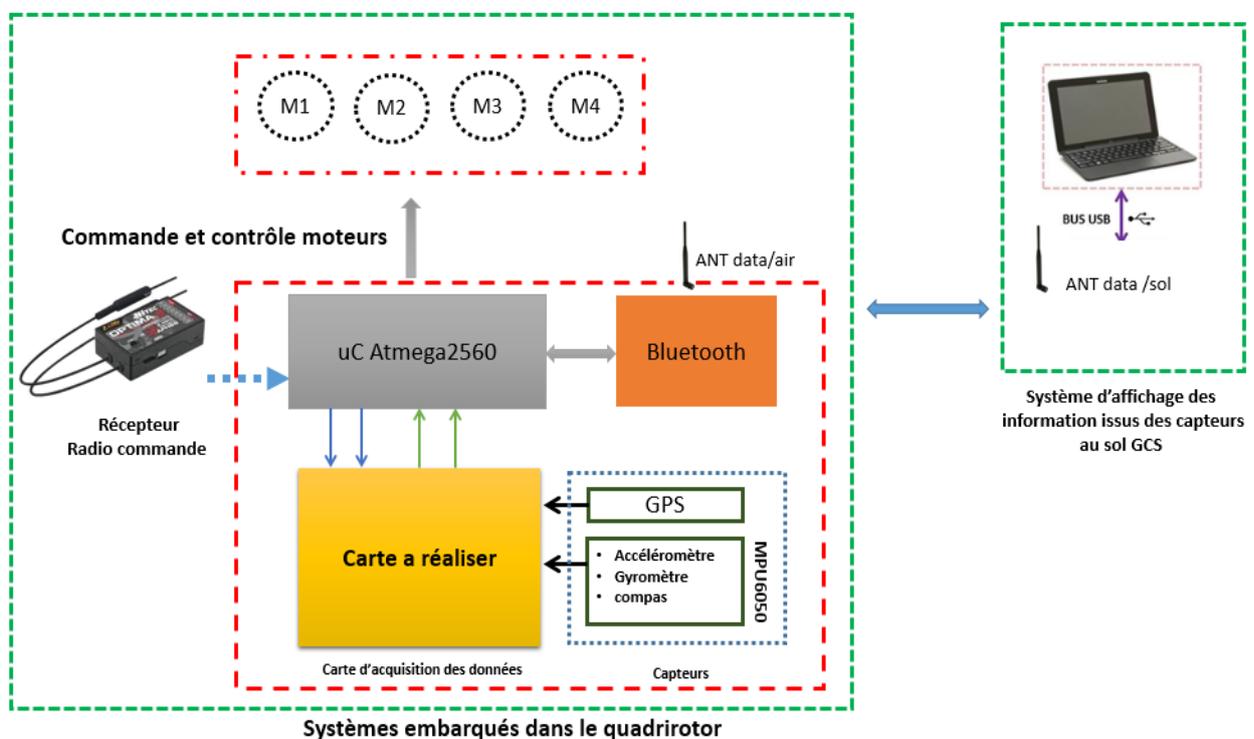


Figure 3.1 : Architecture de système à réaliser.

La deuxième partie du cahier des charges consiste à modéliser le quadricoptère utilisé pour le test de la carte autopilote sous Matlab.

### 3.3 Constitution d'une carte autopilote

#### 3.3.1 Partie hard

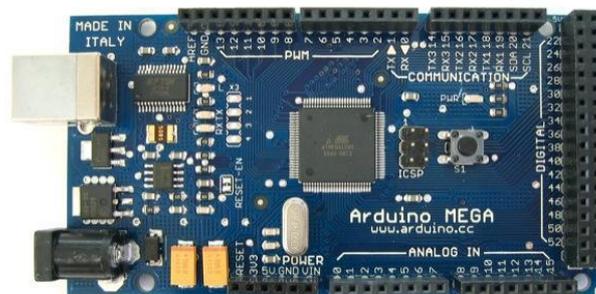
##### a) La carte Arduino

La carte Arduino est le microprocesseur principal de la carte autopilote. Elle fait partie de catégorie des microcontrôleurs, c'est-à-dire d'ordinateur miniature capable de stocker des données, de recevoir et envoyer des informations. Le modèle de carte Arduino utilisée est le MEGA 2560 et elle est basée sur un Atmega2560. [20]

Cette carte dispose :

- 54 broches numériques I/O
- 16 entrées analogiques,
- UART (port série matériel),
- Un quartz 16Mhz,
- Une connexion USB,
- Un connecteur d'alimentation jack,
- Un connecteur ICSP,
- Un bouton de réinitialisation.

Elle contient tout ce qui est nécessaire pour le fonctionnement du microcontrôleur. Elle dispose de toute une série de facilités pour communiquer avec un ordinateur à l'aide d'un câble USB, l'alimentation étant fournie par le port USB, une autre carte Arduino, ou avec d'autres microcontrôleurs.



**Figure 3.2 :** Carte Arduino Méga.

Un circuit intégré Atmega8U2 sur la carte assure la connexion entre cette communication série de l'un des ports série de l'Atmega 2560 vers le port USB de l'ordinateur qui apparaît comme un port COM virtuel pour les logiciels de l'ordinateur. La communication en série de niveau TTL (5V) est disponible sur les broches 0 (RX) et 1 (TX). Le code de programmation de l'Atmega8U2 utilise le driver standard USB.

### b) Les capteurs

La qualité et la précision des capteurs sont très importantes pour assurer une bonne connaissance des paramètres du drone dans l'espace. Dans notre projet, on va utiliser deux capteurs MPU et GPS qui sont bien sûr les éléments les plus importants et les plus complexes. La centrale inertielle a été réalisée à l'aide d'un capteur MPU6050, le GPS à l'aide d'un capteur NEO 6M et un Bluetooth H05 pour la transmission.

Le choix des capteurs résulte d'un compromis entre le prix, la qualité des mesures et la disponibilité.

#### 1. MPU6050

Le capteur utilisé porte la référence GY-521 architecturé autour d'un MPU605 et composé de deux capteurs et d'un processeur :

- Un capteur accéléromètre 3 axes (x, y, z) qui mesure l'accélération.
- Un capteur gyroscope 3 axes qui mesure la vitesse angulaire.
- Un processeur de mouvement digital (DMP) capable de stocker des données et les restituer.

Le MPU 6050 possède une puce MEMS très précise avec une conversion analogique-digitale sur 16 bits simultanée sur chaque canal et une interface I2C (400 kHz). Le capteur contient un registre FIFO de 1024 octets que le microcontrôleur Arduino peut lire si un signal d'interruption est donné.

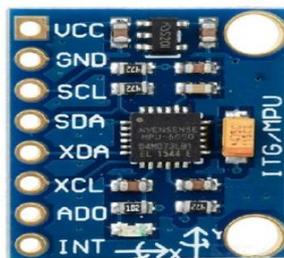


Figure 3.3 : MPU6050.

Le module fonctionne en esclave sur le bus I2C vis à vis de l'Arduino (pins SDA, SLC) mais il peut aussi contrôler un autre dispositif en aval avec AUX-DA et AUX-CL. Le DMP peut faire des calculs rapides directement sur la puce à partir des mesures brutes du capteur et cela réduit la charge sur le microcontrôleur. [15]

#### 2. GPS NEO 6M

Le module GPS NEO-6M est un récepteur GPS complet très performant avec une antenne en céramique intégrée, qui offre une capacité de recherche par satellite puissante, il peut traquer jusqu'à 22 satellites sur 50 canaux, tout on ne consommant qu'un courant d'alimentation de 45mA. Grâce à la pile de sauvegarde des données, le module peut sauvegarder les données en cas de coupure accidentelle de l'alimentation principale.

Les caractéristiques de GPS sont :

- Alimentation : 3V/5V
- Batterie : batterie de secours rechargeable
- Signal lumineux : lumière à DEL
- Dimensions de l'antenne : 25 x 25 mm
- Dimensions du module : 25 mm x 35 mm
- Poids : 19g
- Débit en bauds par défaut : 9600
- Taux de mise à jour de la position : 5Hz



Figure 3.4 : GPS NEO M6.

### 3. Bluetooth HC-05

Le module HC-05 est un module Bluetooth SPP facile à utiliser, conçu pour une configuration de connexion série bidirectionnelle sans fil sur de très courte distance en utilisant des ondes radios UHF sur une bande de fréquence de 2.4 GHz. Le module HC-05 peut être utilisé en configuration maître ou esclave, ce qui en fait une excellente solution de communication sans fil. [16]

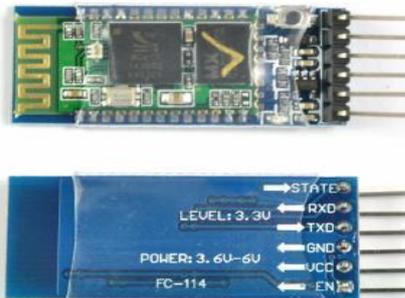


Figure 3.5 : Bluetooth HC-05.

Les caractéristiques :

- Emetteur-récepteur sans fil
- Sensibilité peut atteindre : -80dBm.
- Possède une antenne 2.4GHz intégrée.
- Peut travailler à basse tension (3.1V ~ 4.2V).
- Interface UART avec débit binaire programmable (programmable Baud rate).
- Compact (27mm x 13 mm x 2mm)

### c) Carte d'acquisition à réaliser

La carte d'acquisition à réaliser est une carte imprimée qui est un support plan (Shield), qu'on déposera au-dessus de notre microprocesseur MEGA. Cette carte regroupe les différents capteurs utilisés (MPU6050, GPS Neo M6, Bluetooth HC-05) ainsi que le récepteur RC.

Le PCB de cette carte est réalisé par le logiciel ARES professionnel (Proteus7), qui est un circuit à double face à trous métallisés : c'est-à-dire un circuit contenant deux couches et une métallisation par bain chimique qui est utilisé pour déposer une couche de 24 micromètre dans les trous et qui servira de liaison entre la couche 01 et la couche 02

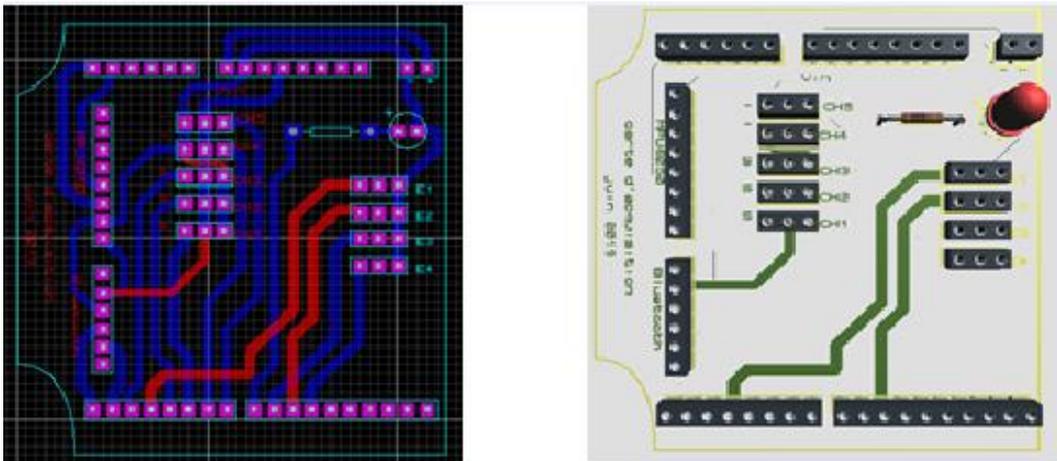


Figure 3.6 : Le PCB de la carte d'acquisition.

### d) Quadrirotor du test

La plate-forme utilisée pour tester la carte autopilote est un quadrirotor fabriqué au niveau du centre de recherche en technologie industrielle CRTI avec les caractéristiques suivantes :



Figure 3.7 : Le quadrirotor CRTI.

### 1. Moteur

Les quatre moteurs utilisés sont des moteurs brushless SunnySky V2216-11, ils sont caractérisés par :

- Dimensions : 48 × 28mm.
- Poids : 39 g.
- Puissance : 210 W.
- KV : 900 tr/min.
- Tension : 10v.
- Courant : 0.4 A



Figure 3.8 : Moteur Brushless.

### 2. Les contrôleurs de vitesse des moteurs ESC

Le contrôleur de vitesse des moteurs doit être compatible avec le moteur et la batterie. Chaque moteur a besoin d'un ESC. Les caractéristiques de l'ESC sont :

- Vitesse Maximum : 210000 t/min pour 2 pôles, 70000 t/min pour 6 pôles, 35000 t/min pour 12 pôles.
- Courant de sortie : 40A continue.
- BEC : 3A / 5V.
- Dimensions : 55mm x 28mm x 12mm.
- Résistance de sortie extrêmement basse, très bonne résistance au courant.
- Multiple protections : Coupure basse tension / Température / Perte de signal.
- 3 modes de démarrage : Normal / Soft / Super-Soft.



Figure 3.9 : ESC.

### 3. Les hélices

L'hélice utilisée est en matière plastique qui présente les caractéristiques suivantes :

- Le diamètre (généralement en pouce) : 9"
- Le pas (en pouce aussi) : 5"
- Poids : 9.6g



Figure 3.10 : L'hélice.

### 4. La batterie

La batterie utilisée sur le quadrirotor est du type Lipo 4S. Elle a les caractéristiques suivantes :

- Capacité : 6300 mAh
- Tension : 14.8v
- Courant de décharge continu : 30 C
- Energie : 93.2 Wh
- Dimension : 155 × 48 × 24mm
- Poids : 375 g
- Prise principale : EC5 femelle
- Prise d'équilibrage : JST-XH



Figure 3.11 : Batterie Lipo 4S.

### 5. Récepteur Futaba

Le récepteur Futaba R2008SB est compatible avec la radio commande T8J, il utilise la technologie S-FHSS mais peut également fonctionner avec un émetteur en FHSS. Ces caractéristiques :

- Dimensions : 37 × 25 × 15 mm
- Masse : 8.1g.
- Nombre de voies : 8.
- Deux antennes.
- Modulation : 2.4Ghz.
- Tension d'alimentation : 4.8 - 7.4V.
- Compatible avec système S-Bus.



Figure 3.12 : Récepteur Futaba.

### 6. Radiocommande Futaba T8J

La radio commande Futaba T8J est un type de télécommande qui permet de contrôler à distance. Elle est compatible avec les avions, drones et hélicoptères. Elle utilise principalement la fréquence 2.4 GHz offrant une portée moyenne d'environ 1500m.

Ces caractéristiques sont :

- Gestion du mode S-FHSS ou FHSS.
- Mode avions et hélicoptères.
- 20 modèles en mémoire.
- Écran LCD.
- Transfer de données des modèles entre 8J.
- 6 mixages programmables.
- Double antennes interne.
- 8 voies.



Figure 3.13 : Futaba T8J.

### 3.3.2 Partie soft

#### a) Logiciel Arduino

Le logiciel de programmation des modules Arduino est une application Java libre et multiplateformes. Il sert d'éditeur de code et de compilateur qui peut transférer le *firmware* et le programme à travers la liaison série. Il est également possible de se passer de l'interface Arduino, de compiler et uploader les programmes via l'interface en ligne de commande. Le langage de programmation utilisé est le C++ compilé avec `avr-g++3`, lié à la bibliothèque de développement Arduino et permettant l'utilisation de la carte et de ses entrées/sorties. Le logiciel Arduino intègre également un terminal série (fenêtre séparée) qui permet d'afficher des messages textes reçus de la carte Arduino et d'envoyer des caractères vers la carte Arduino. Cette fonctionnalité permet une mise au point très facile des programmes, permettant d'afficher sur l'ordinateur l'état des variables, des résultats de calculs : un élément essentiel pour améliorer, tester et corriger ses programmes. [20]

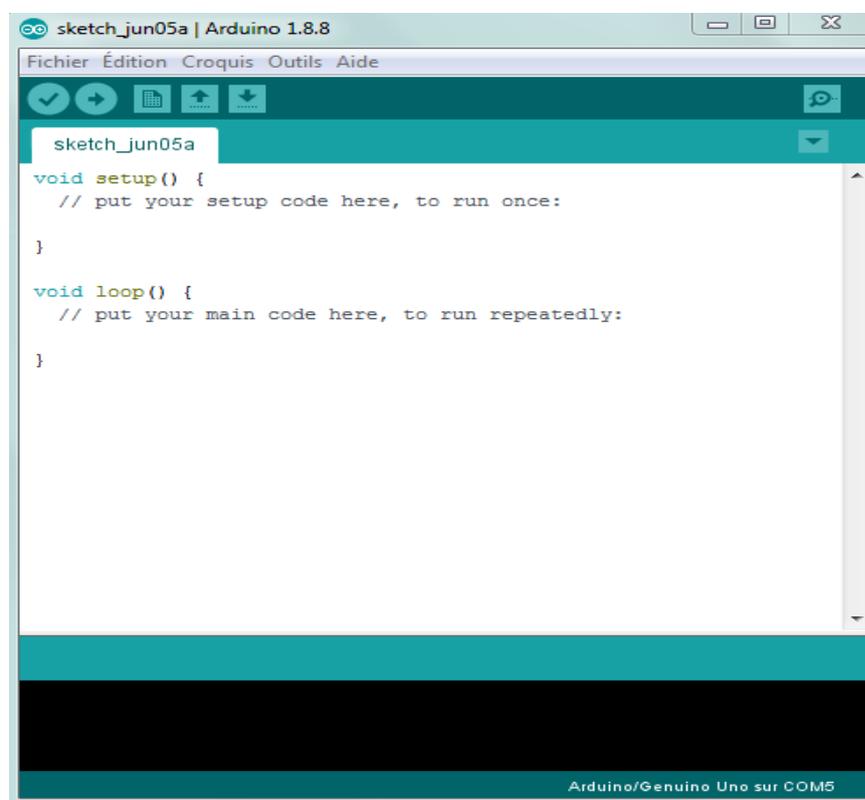


Figure 3.14 : Interface du logiciel IDE.

#### b) Multiwii

Multiwii est un logiciel qui a été créé à l'aide du logiciel Arduino, il permet de contrôler les multi-rotors par une radio contrôle. Il est disponible en tant que logiciel open source visant à fournir le cerveau d'une plate-forme volante. Il est compatible avec plusieurs cartes et capteurs matériels (gyroscopes, accéléromètres, GPS... etc.). [23]

## Chapitre 3 : Équipements et logiciels

Le logiciel est composé de 11 fichiers.cpp et de leurs fichiers d'en-tête (Alarms, EEPROM, GPS, IMU, LCD, Output, Protocol, Multiwii, Rx, Sensors et Serial) et de trois fichiers.hseuls (Config, Def et Types).

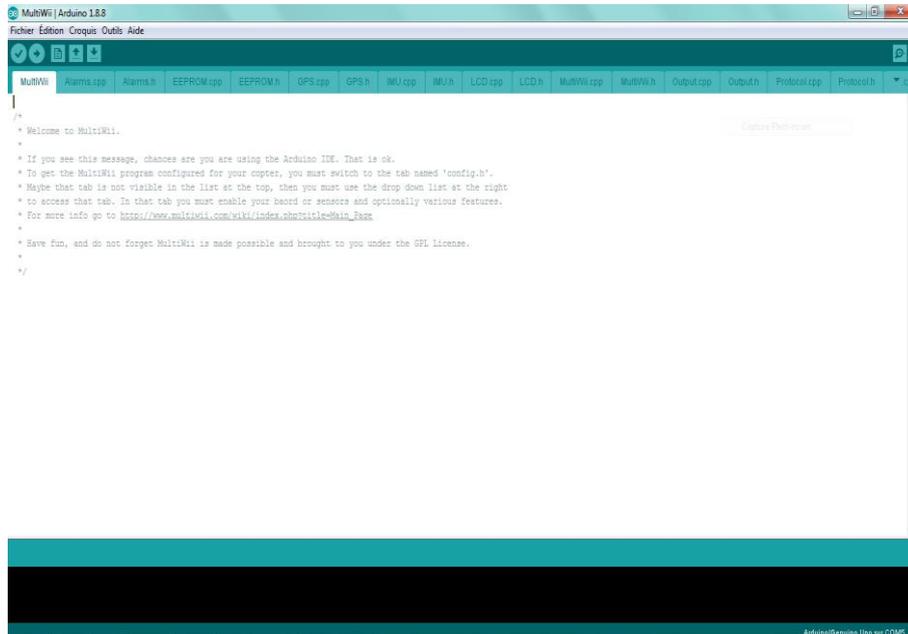


Figure 3.15 : Interface du logiciel Multiwii combiné avec Arduino.

Le réglage des paramètres principaux peut être effectué par l'intermédiaire d'une fenêtre graphique. Ce programme permet d'interagir avec le quadrirotor par l'intermédiaire d'un câble USB ou par Bluetooth. Une fois connectée, la fenêtre affiche les paramètres pouvant être modifiés ainsi que diverses informations fournies par la carte comme : l'inclinaison des axes de roulis et de tangage, l'orientation magnétique du prototype, les données GPS, une visualisation 3D de l'orientation du modèle dans l'espace, zone d'affichage des informations des capteurs... etc.

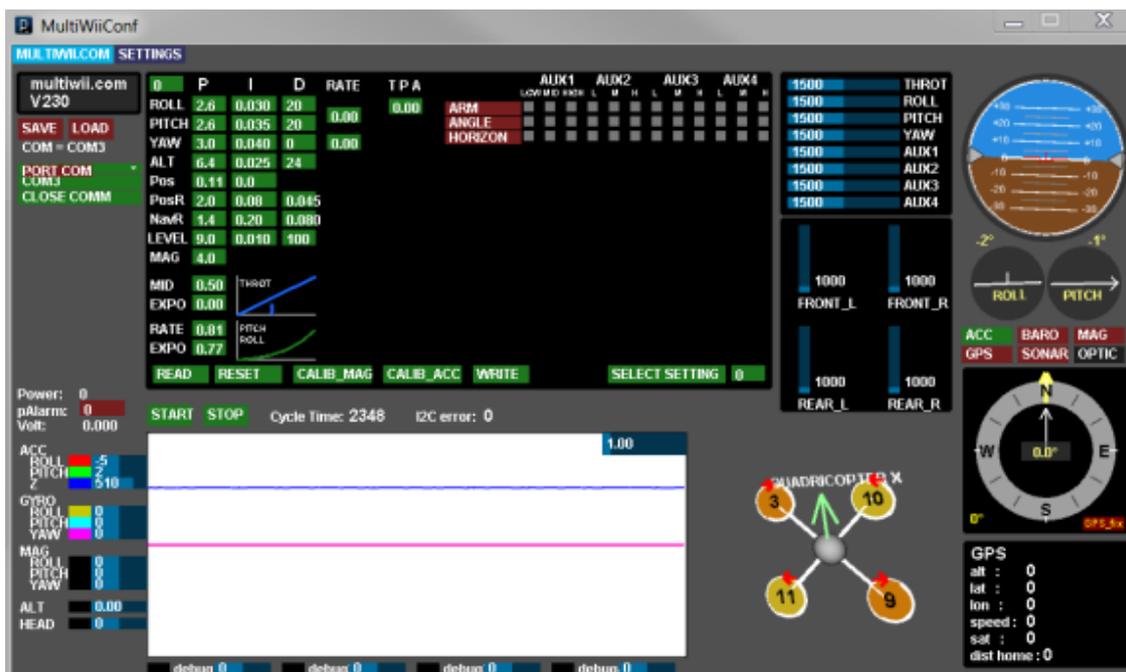


Figure 3.16 : Interface graphique de Multiwii.

### c) U-center

Le logiciel d'évaluation u-center fournit aux intégrateurs des systèmes et aux utilisateurs un outil simple et rapide, c'est un outil d'évaluation et de visualisation GPS très puissant. Le but d'u-center est de permettre aux utilisateurs de [22] :

- Effectuer des tests de performance sur les périphériques u-blox et autres systèmes GPS.
- Configurer les puces et modules de positionnement u-blox GPS.
- Mettre à jour le firmware sur les modules GPS.
- Tester les performances supplémentaires fournies par le service Assist Now de u-blox.

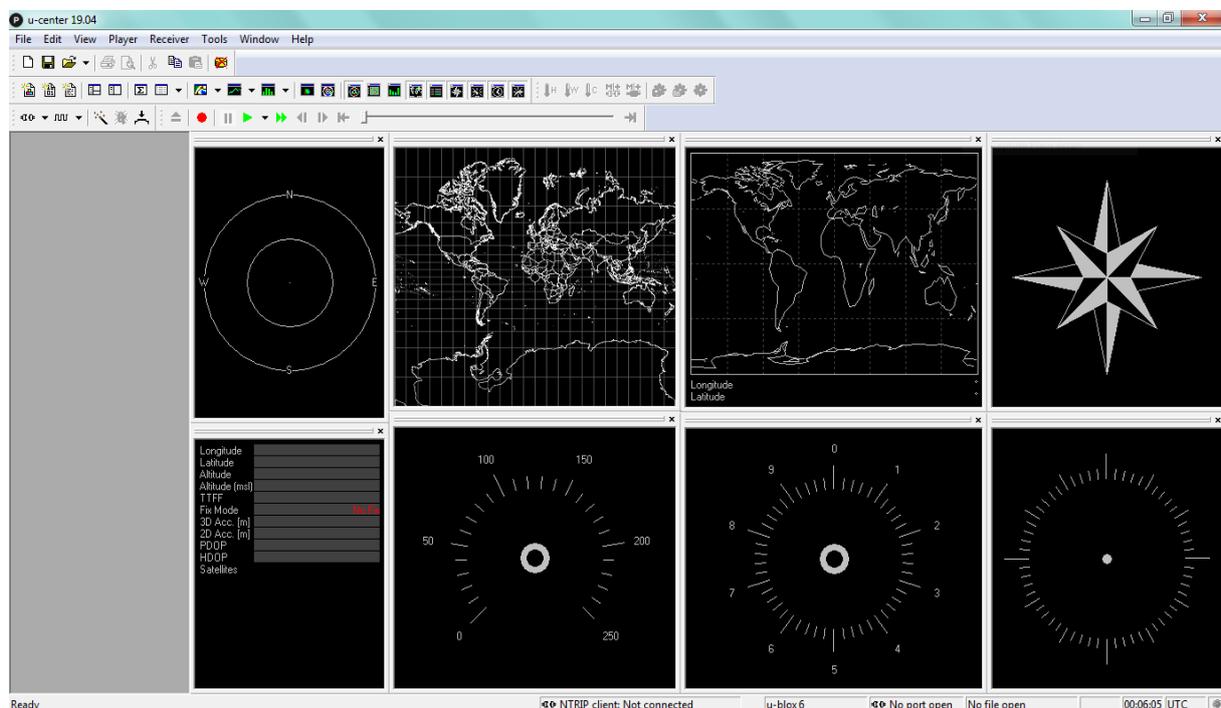


Figure 3.17 : Interface du logiciel u-center.

### d) Proteus (ISIS & ARES)

Proteus est un logiciel permettant la conception de la carte électronique assistée par ordinateur qui est édité par la société Lab-center Electronics. Proteus est composé de deux logiciels principaux : ISIS, permettant entre autres la création de schémas et la simulation électrique. ARES est dédié à la création de circuits imprimés. [21]

#### ➤ ISIS

Le logiciel ISIS de Proteus est principalement connu pour éditer des schémas électriques. Par ailleurs, le logiciel permet également de simuler ces schémas, ce qui permet de déceler certaines erreurs pendant l'étape de conception. Indirectement, les circuits électriques conçus grâce à ce logiciel peuvent être utilisés dans des documentations car le logiciel permet de contrôler la majorité de l'aspect graphique des circuits.

### ➤ ARES

Le logiciel ARES est un outil d'édition et de routage qui complète parfaitement ISIS. Un schéma électrique réalisé sur ISIS peut être importé facilement sur ARES pour réaliser le PCB de la carte électronique. Bien que l'édition d'un circuit imprimé soit plus efficace lorsqu'elle est réalisée manuellement, ce logiciel permet de placer automatiquement les composants et de réaliser le routage automatiquement.

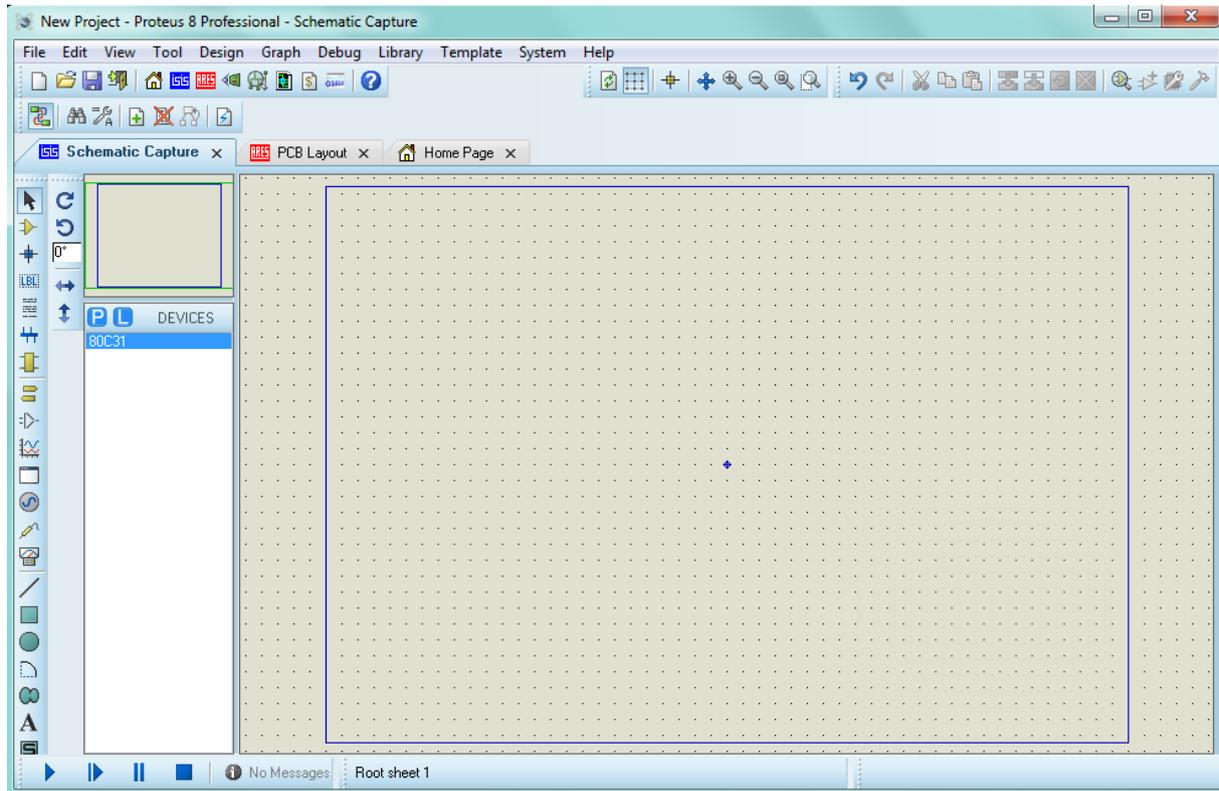


Figure 3.18 : Interface du logiciel Proteus.

## 3.4 Conclusion

Dans ce chapitre nous avons présenté le cahier des charges de notre carte autopilote, ainsi que les capteurs et logiciel utilisés pour sa réalisation.



# Chapitre 4 : Simulation sous Matlab

## 4.1 Introduction :

Dans ce chapitre, le logiciel de simulation MATLAB permet de simuler le comportement dynamique du quadricoptère représenté par un modèle mathématique établi dans le chapitre 2. À chaque étape de la simulation, l'état de chaque partie du système est calculé à l'aide de solveurs basés sur le temps ou sur l'événement. En règle générale, les logiciels de simulation intègrent également des outils de visualisation, tels que des systèmes d'affichage des données (graphes), qui permettent de surveiller la simulation pendant son exécution.

## 4.2 Résolution numérique du modèle mathématique (équation non linéaire)

Il existe plusieurs méthodes pour résoudre les équations non linéaires (Euler, Newton, Adams Moulton, Runge Kutta ... etc.). On a opté pour la méthode de Runge Kutta car ce choix se base sur [26] :

- La facilité.
- L'approximation la plus précise de la solution.
- L'avantage d'être simples à mettre en œuvre.
- La stabilité pour les fonctions courantes rencontrées en physique.

### 4.2.1 Comparaison avec les autres méthodes

Le tableau 4.1 montre la comparaison entre quelques méthodes de résolutions numériques selon plusieurs paramètres pour faire le choix :

	<b>Euler</b>	<b>Runge kutta d'ordre 4</b>	<b>Adams Moulton</b>
<b>Type de méthode</b>	Pas unique	Pas unique	Pas multiple
<b>Erreur locale</b>	$O(h^3)$	$O(h^5)$	$O(h^5)$
<b>Erreur globale</b>	$O(h^2)$	$O(h^4)$	$O(h^4)$
<b>Nb d'évaluation de fonction</b>	2	4	2
<b>Facilité pour changer le pas</b>	oui	Oui	Non
<b>Stabilité</b>	Bonne	Bonne	Bonne
<b>Recommandée</b>	non	Oui	Oui

**Tableau 4.1 :** Comparaison entre les méthodes.

Où : **h** est le pas et **O** est l'ordre de l'erreur

### 4.2.2 Méthode de Rung Kutta d'ordre 4

La technique de Runge-Kutta d'ordre 4 est un schéma numérique à un pas ( $h$ ) basé sur la discrétisation de la variable  $t$  qui permet de résoudre les équations différentielles ordinaires (ODE) à une seule variable. Elles font parties des méthodes les plus populaires. C'est Carl Runge et Martin Kutta qui, au début du 20e siècle, ont inventé cette méthode. [26]

L'idée fondamentale des méthodes de Runge-Kutta est d'intégrer l'équation suivante sur  $[t_n, t_{n+1}]$  et d'approcher cette intégrale de façon plus précise :

$$\begin{cases} \frac{dy}{dt} = f(t, y(t)) \\ \text{ou:} \\ 0 \leq t \leq T \\ y(0) = y_0 \end{cases} \quad (4.1)$$

Avec :

$y(t)$  : La fonction recherchée (scalaire ou vecteur).

$y_0$  : La valeur initiale.

$f$  : Une fonction connue suffisamment régulière pour que l'existence et l'unicité de la solution ne pose pas de problème.

$h$  : Le pas

$T$  : La durée.

En intégrant l'équation différentielle entre  $t_n$  et  $t_{n+1}$ , on obtient la relation suivante :

$$y(t_{n+1}) - y(t_n) = \int_{t_n}^{t_{n+1}} f(t, y(t)) dt \quad (4.2)$$

$y_n$  est la valeur approchée de  $y(t_n)$  pour les différents instants  $t_n = nh$ . On obtient :

$$\begin{cases} k_1 = hf(t_n, y(t_n)) \\ k_2 = hf\left(t_n + \frac{1}{2}h, y(t_n) + \frac{1}{2}k_1\right) \\ k_3 = hf\left(t_n + \frac{1}{2}h, y(t_n) + \frac{1}{2}k_2\right) \\ k_4 = hf(t_n + h, y(t_n) + k_3) \end{cases}$$

$$y(t_{n+1}) = y(t_n) + \frac{1}{6}(k_1 + 2k_2 + 2k_3 + k_4) \quad (4.3)$$

**Remarque :**

Même si l'erreur produite par la méthode Runge Kutta est de l'ordre de  $O(h^5)$ , il reste la question du choix du pas  $h$  afin d'avoir le meilleur compromis entre précision et temps de calcul. Il est intuitivement clair que l'on peut choisir un  $h$  relativement grand dans une région où  $y(t)$  varie plutôt lentement, mais un petit  $h$  est nécessaire quand il y a de fortes variations de  $y(t)$ .

### 4.2.3 Algorithme du RK4

1. Initialisation du pas  $h$  et de la durée  $T$ .
2. Initialisation des conditions initiales :  $t = 0$  et  $y = y(0)$ .
3. Définition de la fonction  $f(t, y)$ .
4. Tant que  $t \leq T$  faire :
  - (a) Calcul de  $k_1 = f(t, y)$ .
  - (b) Calcul de  $k_2 = f\left(t + \frac{h}{2}, y + h\frac{k_1}{2}\right)$ .
  - (c) Calcul de  $k_3 = f\left(t + \frac{h}{2}, y + h\frac{k_2}{2}\right)$ .
  - (d) Calcul de  $k_4 = f(t + h, y + hk_3)$ .
  - (e)  $y = y + \frac{1}{6}(k_1 + 2k_2 + 2k_3 + k_4)$  ;  $t = t + h$
  - (f) Enregistrement des résultats calculés.

### 4.2.4 Simulation sous MATLAB

- 1) On utilise le système d'équation obtenu dans le chapitre 2 (représentation d'état) :

$$\begin{cases} \dot{\phi} = p \\ \dot{p} = a_1qr + a_2p^2 + a_3\Omega_rq + b_1u_2 \\ \dot{\theta} = q \\ \dot{q} = a_4pr + a_5q^2 + a_6\Omega_rp + b_2u_3 \\ \dot{\psi} = r \\ \dot{r} = a_7pq + a_8r^2 + b_3u_4 \\ \dot{x} = u \\ \dot{u} = a_9u + \frac{u_1}{m}(c\phi c\psi s\theta + s\phi s\psi) \\ \dot{y} = v \\ \dot{v} = a_{10}v + \frac{u_1}{m}(c\phi s\psi s\theta - s\phi c\psi) \\ \dot{z} = w \\ \dot{w} = a_{11}w + \frac{u_1}{m}(\cos\phi \cos\theta) - g \end{cases} \quad (4.4)$$

- 2) On prend les conditions initiales, dans notre cas, c'est l'état du repos :

$$\begin{cases} \phi_0 = 0 \\ p_0 = 0 \\ \theta_0 = 0 \\ q_0 = 0 \\ \psi_0 = 0 \\ r_0 = 0 \\ x_0 = 0 \\ u_0 = 0 \\ y_0 = 0 \\ v_0 = 0 \\ z_0 = 0 \\ w_0 = 0 \end{cases} \quad (4.5)$$

- 3) On insère les paramètres du drone CRTI :

Symbole	Désignation	Valeur	unité
g	L'accélération gravitationnelle	9.81	$m/s^2$
b	Le coefficient de portance	$2.9842 * 10^{-5}$	$N/rad/s$
m	Masse	0.486	$kg$
$I_x$	Moment d'inertie selon l'axe x	$3.8278 * 10^{-3}$	$kg * m^2$
$I_y$	Moment d'inertie selon l'axe y	$3.8288 * 10^{-3}$	$kg * m^2$
$I_z$	Moment d'inertie selon l'axe z	$7.6566 * 10^{-3}$	$kg * m^2$
$J_r$	L'inertie du rotor	$2.8385 * 10^{-5}$	$kg * m^2$
L	Distance entre le moteur et le C.G	0.25	$m$
$w_r$	La vitesse résiduelle des moteurs	0	$rad/s$
$K_{fax}$	Coefficient de fortement aérodynamique selon x	$5.5670 * 10^{-4}$	$N/rad/s$
$K_{fay}$	Coefficient de fortement aérodynamique selon y	$5.5670 * 10^{-4}$	$N/rad/s$
$K_{faz}$	Coefficient de fortement aérodynamique selon z	$6.3540 * 10^{-4}$	$N/rad/s$
$K_{ftx}$	Coefficient de trainée de translation selon x	$5.5670 * 10^{-4}$	$N/m/s$
$K_{fty}$	Coefficient de trainée de translation selon y	$5.5670 * 10^{-4}$	$N/m/s$
$K_{ftz}$	Coefficient de trainée de translation selon z	$6.3540 * 10^{-4}$	$N/m/s$
d	Le coefficient de drag	$3.2320 * 10^{-7}$	$N * m/rad/s$
$w_1, w_2, w_3, w_4$	Vitesse de rotation des hélices 1, 2, 3,4 respectivement	0	$rad/s$

Tableau 4.2 : Paramètres du quadrirotor.

#### 4.2.5 Résultats de simulation :

➤ 1<sup>er</sup> cas : changement du pas ( $h = 0.1$  et  $h = 0.5$ )

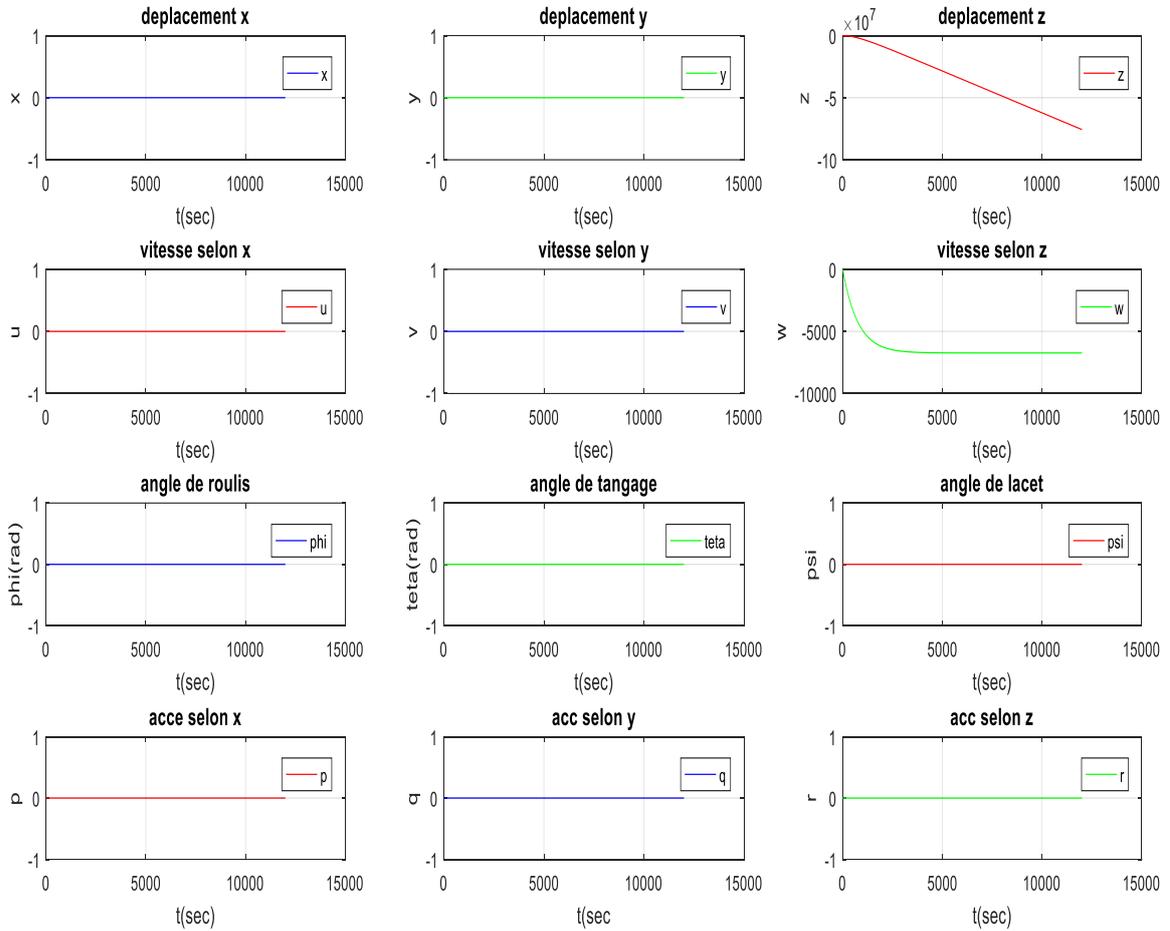
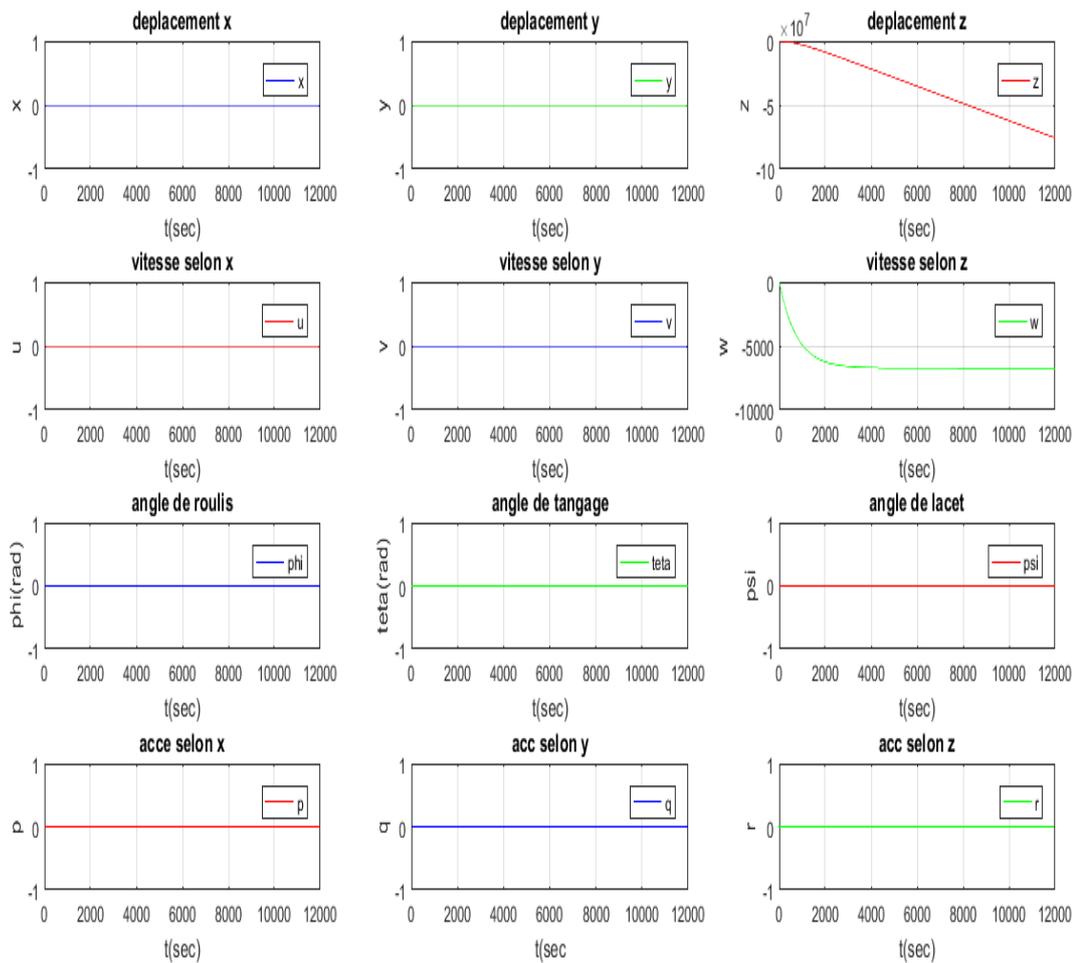


Figure 4.1 : Résolution avec RK4 avec  $h=0.1$ .



**Figure 4.2 :** Résolution avec RK4 avec  $h=0.5$ .

### Discussion des résultats :

D'après les résultats obtenus à partir de la simulation du premier cas, on remarque que :

1. Le choix de pas  $h$  revient à la précision voulu.
2. En diminuant le pas la précision augmente.

On remarque aussi que tous les graphes commencent du 0 et leurs valeurs finale est 0 sauf le déplacement  $z$  et la vitesse linéaire  $w$  et ça revient aux conditions initiales, comme la dérivé de la vitesse de lacet qui dépend de l'accélération gravitationnelle.

- 2<sup>ème</sup> cas : pour ce cas on a choisi les angles d'Euler à  $\phi = \theta = \psi = 12^\circ = 0.209$  rad

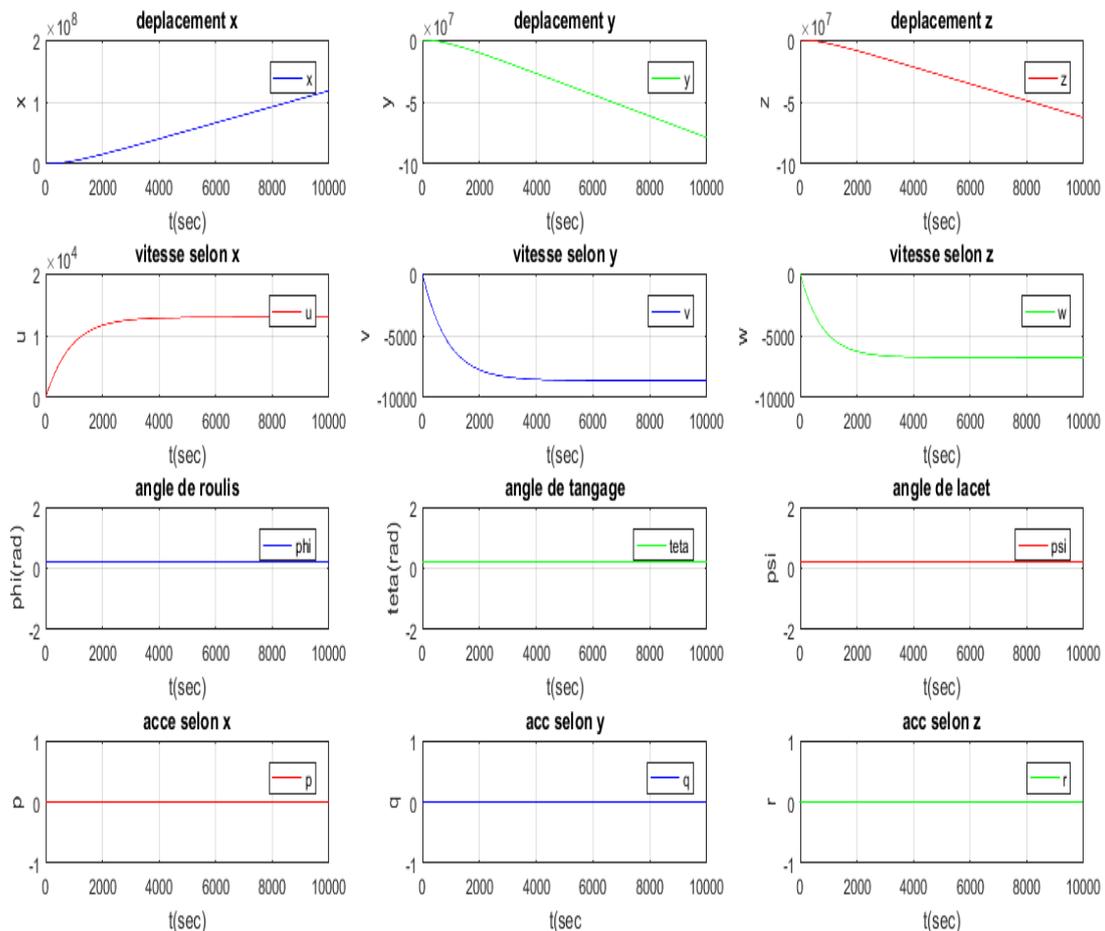


Figure 4.3 : Résolution avec RK4 en changeant les angles.

### Discussion des résultats :

D'après les résultats trouvés, on remarque le changement dans les graphes de  $x, y, z, u, v, w$  alors que les graphes  $p, q, r, \phi, \theta, \psi$  ne changent pas, et ça c'est due aux paramètres liés aux équations et aussi aux conditions initiales.

On remarque également que tous les graphes qui ont changé commencent du 0 car leurs conditions initiales sont nulles.

D'après les graphes  $u, v, w$  tant vers une valeur finale.

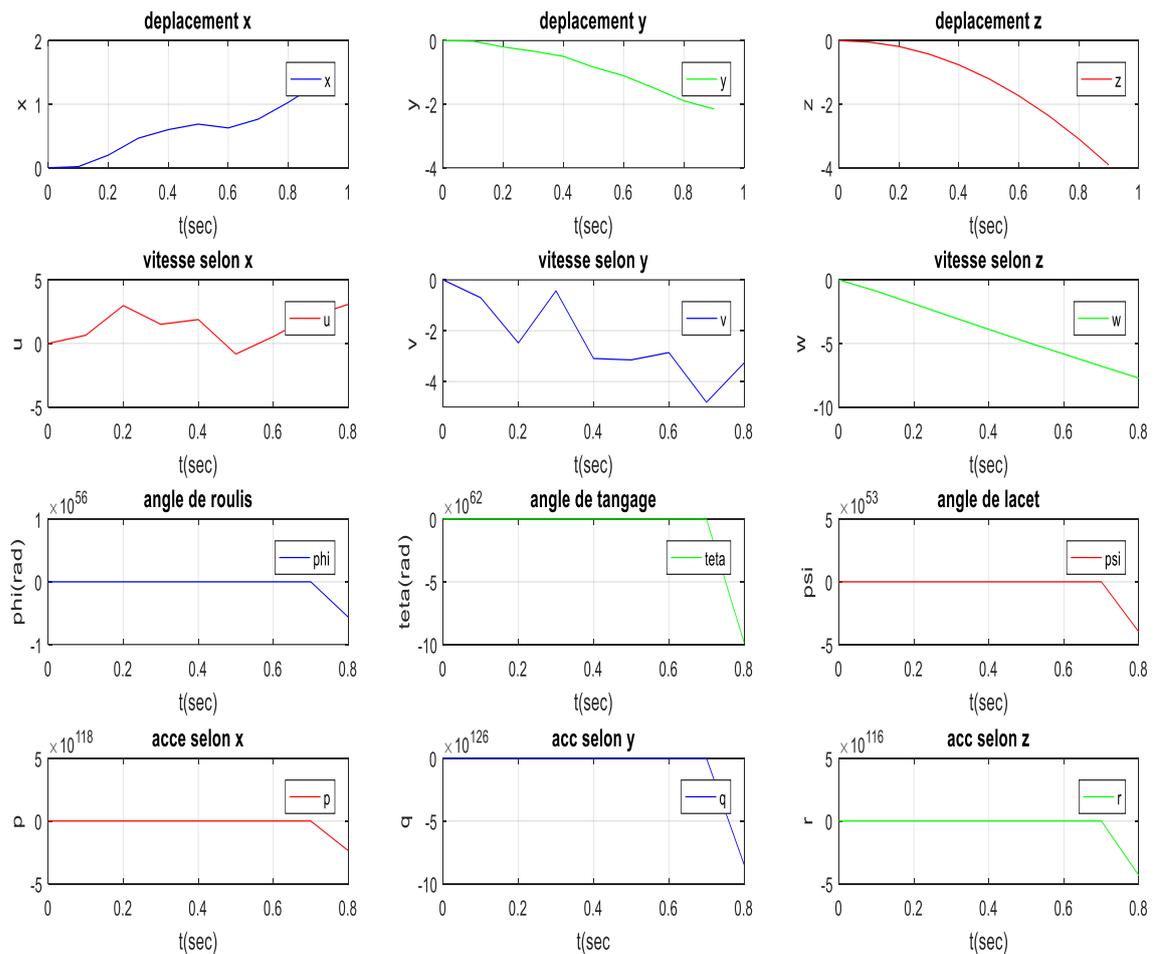
On déduit d'après les 12 graphes que notre système est n'est pas stable donc on doit rajouter un commande.

### ➤ 3<sup>eme</sup> cas : Mouvement de roulis

Dans ce cas on a réalisé un mouvement de roulis en :

- Augmentant la vitesse des rotors 1et 3 (la même valeur).
- Différencier les vitesses des rotors 2 et 4.

$$\Psi = 0.785 \text{ rad} ; w_1 = w_3 = 300 \frac{\text{rad}}{\text{s}} ; w_2 = 400 \frac{\text{rad}}{\text{s}} ; w_4 = 500 \frac{\text{rad}}{\text{s}}$$



**Figure 4.4 :** résolution avec RK4 pour un mouvement de roulis.

### Discussion des résultats :

Cette simulation montre la variation de tous les paramètres du quadrirotor en choisissant un mouvement de roulis.

Le bon choix du pas h induit la stabilité du paramètre, alors que dans ce cas on a utilisé le même h pour tous les paramètres car les équations sont couplées.

D'après les graphes le système a besoin d'une commande pour contrôler la stabilité.

## 4.3 Résolution numérique du modèle mathématique (équation linéaire)

Tout d'abord, on doit entamer la résolution numérique des fonctions linéaires obtenues à partir du modèle non linéaire. Ensuite, on passe à la représentation d'état permettant d'extraire les fonctions de transferts pour chaque variable d'état (x, y, z,...). À la fin, une étude de stabilité est faite, pour pouvoir appliquer une commande « PID » ce qui permet d'assurer la stabilité du quadrirotor.

### 4.3.1 Linéarisation

En utilisant les faibles angles et l'hypothèse de la force de trainée négligeable, les équations peuvent être linéarisées par la méthode jacobéenne. Les fonctions sont décrites de façon qu'elles soient en relation avec les variables d'état et en relation avec les variables de contrôle autour d'un point d'équilibre. Le système linéaire du modèle dynamique du quadrirotor est :

- 1) Equations cinématiques du mouvement de translation :

$$\begin{cases} \dot{x} = u + \psi v - \theta w \\ \dot{y} = (\psi + \varphi\theta)u + (1 + \varphi\theta\psi)v + \varphi w \\ \dot{z} = (\varphi\psi + \theta)u + (-\varphi + \theta\psi)v + w \end{cases} \quad (4.6)$$

- 2) Équations dynamique du mouvement de translation :

$$\begin{cases} \dot{u} = -\frac{1}{m_t}\rho AC_D u^2 - g\theta + (rv - qw) \\ \dot{v} = -\frac{1}{m_t}\rho AC_D v^2 + g\varphi + (pw - ru) \\ \dot{w} = -\frac{1}{m_t}\rho AC_D w^2 + g + \frac{F}{m_t} + (qu - pv) \end{cases} \quad (4.7)$$

- 3) Equations cinématiques du mouvement de rotation :

$$\begin{cases} \dot{\varphi} = p + (q\varphi + r)\theta \\ \dot{\theta} = q - r\varphi \\ \dot{\psi} = (q\varphi + r) \end{cases} \quad (4.8)$$

- 4) Equations dynamiques du mouvement de rotation :

$$\begin{cases} \dot{p} = \frac{1}{I_{xx}} [T_\varphi + (I_{yy} - I_{zz})q r] \\ \dot{q} = \frac{1}{I_{yy}} [T_\theta + (I_{zz} - I_{xx})p r] \\ \dot{r} = \frac{1}{I_{zz}} [T_\psi + (I_{xx} - I_{yy})p q] \end{cases} \quad (4.9)$$

## Chapitre 4 : Simulation sous MATLAB

La représentation d'état s'écrit sous forme suivante :

$$\begin{cases} \dot{X} = A.X + B.U \\ Y = C.X + D.U \end{cases} \quad (4.10)$$

Avec :

$X \in R^n$  : Vecteur qui représente les  $n$  variables d'état.

$Y \in R^q$  : Vecteur qui représente les  $q$  mesures.

$U \in R^p$  : Vecteur qui représente les  $p$  commandes.

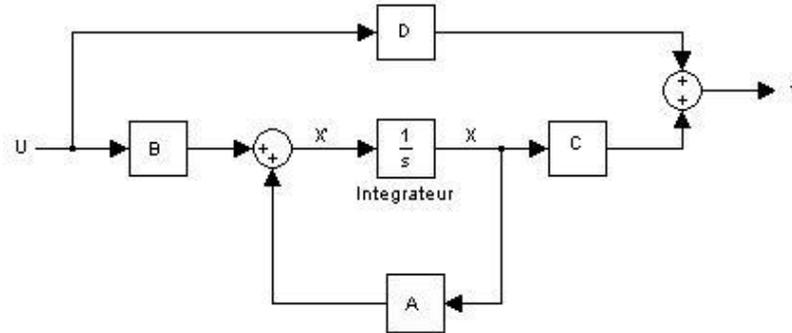
$A \in R^{n \times n}$  : Matrice d'état dynamique.

$B \in R^{n \times p}$  : Matrice de commande.

$C \in R^{q \times n}$  : Matrice d'observation.

$D \in R^{q \times p}$  : Matrice d'action directe

La présentation sous forme d'un schéma en bloc est :



Dans notre cas, on a :

➤ Variables d'état :

$$X^T = [x \ y \ z \ u \ v \ w \ \varphi \ \theta \ \psi \ p \ q \ r] \quad (4.11)$$

➤ Variables de Contrôle :

$$U^T = [T_\varphi \ T_\theta \ T_\psi \ F] = [u_1 \ u_2 \ u_3 \ u_4] \quad (4.12)$$

$$U = \begin{bmatrix} u_1 \\ u_2 \\ u_3 \\ u_4 \end{bmatrix} = \begin{bmatrix} T_\varphi \\ T_\theta \\ T_\psi \\ F \end{bmatrix} = \begin{bmatrix} l(F_2 - F_4) \\ l(F_3 - F_1) \\ \tau_1 - \tau_2 + \tau_3 - \tau_4 \\ F_1 + F_2 + F_3 + F_4 \end{bmatrix} = \begin{bmatrix} bl(\omega_2^2 - \omega_4^2) \\ bl(\omega_3^2 - \omega_1^2) \\ d(\omega_1^2 - \omega_2^2 + \omega_3^2 - \omega_4^2) \\ b(\omega_1^2 + \omega_2^2 + \omega_3^2 + \omega_4^2) \end{bmatrix} \quad (4.13)$$

Avec :

$$F = \begin{bmatrix} F_{arr,x} \\ F_{arr,y} \\ F_{arr,z} \end{bmatrix} = \begin{bmatrix} \frac{1}{2}\rho AC_D u^2 \\ \frac{1}{2}\rho AC_D v^2 \\ \frac{1}{2}\rho AC_D w^2 \end{bmatrix} \quad (4.14)$$

$T_\varphi$  : Couple moteur selon x.

$T_\psi$  : Couple moteur selon z.

$F$  : Force de portance.

$T_\theta$  : Couple moteur selon y.

Matrice d'état A et de contrôle B sont :

$$A = \begin{bmatrix} \frac{\partial f_1}{\partial x_1} & \dots & \frac{\partial f_1}{\partial x_n} \\ \vdots & \ddots & \vdots \\ \frac{\partial f_n}{\partial x_1} & \dots & \frac{\partial f_n}{\partial x_n} \end{bmatrix}_{x_0, u_0}, \quad B = \begin{bmatrix} \frac{\partial f_1}{\partial u_1} & \dots & \frac{\partial f_1}{\partial u_n} \\ \vdots & \ddots & \vdots \\ \frac{\partial f_n}{\partial u_1} & \dots & \frac{\partial f_n}{\partial u_n} \end{bmatrix}_{x_0, u_0} \quad (4.15)$$

Par conséquent, la matrice d'état A est :

$$A = \begin{bmatrix} 0 & 0 & 0 & 1 & \psi & -\theta & 0 & -w & v & 0 & 0 & 0 \\ 0 & 0 & 0 & \psi - \varphi\theta & 1 + \varphi\theta\psi & \varphi & \theta u + \theta\psi v + w & \varphi u + \varphi\psi v & -u + \varphi\theta v & 0 & 0 & 0 \\ 0 & 0 & 0 & \theta + \varphi\psi & \theta\psi - \varphi & 1 & \psi u - v & u + \psi v & \varphi u + \theta v & 0 & 0 & 0 \\ 0 & 0 & 0 & -\frac{2}{m_t}\rho AC_D u & r & -q & 0 & -g & 0 & 0 & -w & v \\ 0 & 0 & 0 & -r & -\frac{2}{m_t}\rho AC_D v & p & g & 0 & 0 & w & 0 & -u \\ 0 & 0 & 0 & q & -p & -\frac{2}{m_t}\rho AC_D w & 0 & 0 & 0 & -v & u & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & q\theta & q\varphi + r & 0 & 1 & \varphi\theta & \theta \\ 0 & 0 & 0 & 0 & 0 & 0 & -r & 0 & 0 & 0 & 1 & -\varphi \\ 0 & 0 & 0 & 0 & 0 & 0 & q & 0 & 0 & 0 & \varphi & 1 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & \frac{(I_{yy} - I_{zz})r}{I_{xx}} & \frac{(I_{yy} - I_{zz})q}{I_{xx}} \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & \frac{(I_{zz} - I_{xx})r}{I_{yy}} & 0 & \frac{(I_{zz} - I_{xx})p}{I_{yy}} \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & \frac{(I_{xx} - I_{yy})q}{I_{zz}} & \frac{(I_{xx} - I_{yy})p}{I_{zz}} & 0 \end{bmatrix}_{x_0, u_0} \quad (4.16)$$

$$B = \begin{bmatrix} 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & \frac{1}{m_t} \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ \frac{1}{I_{xx}} & 0 & 0 & 0 \\ 0 & \frac{1}{I_{yy}} & 0 & 0 \\ 0 & 0 & \frac{1}{I_{zz}} & 0 \end{bmatrix} \quad (4.17)$$

Il convient de noter que la linéarisation des forces de traînée a été effectuée. Cette approximation n'est valable que pour les régions proches du point d'équilibre car il s'agit des termes quadratiques. Dans les régions situées en dehors de ce point, l'approximation n'est plus valide et conduit à des erreurs significatives. Nous avons également obtenu la matrice B, en supposant que la réponse de l'actionneur est plus rapide que celle du système.

4.3.2 Simulation MATLAB

On prend toutes les conditions initiales nulles, on obtient les résultats de la matrices A et B :

$$A = \begin{bmatrix} 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & -9.8 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 9.8 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix} ; B = \begin{bmatrix} 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 2.0576 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 261.2467 & 0 & 0 & 0 \\ 0 & 261.1784 & 0 & 0 \\ 0 & 0 & 130.6063 & 0 \end{bmatrix} \tag{4.18}$$

Les fonctions de transferts obtenues sont représentées dans le tableau suivant :

Sortie \ Entrée	entrée $T_\varphi$	entrée $T_\theta$	entrée $T_\psi$	entrée $f$
Sortie x	0	$\frac{-2560}{s^4}$	0	0
Sortie y	$\frac{2560}{s^4}$	0	0	0
Sortie z	0	0	0	$\frac{2.058}{s^2}$
Sortie u	0	$\frac{-2560}{s^3}$	0	0
Sortie v	$\frac{2560}{s^3}$	0	0	0
Sortie w	0	0	0	$\frac{2.058}{s}$
Sortie $\varphi$	$\frac{261.2}{s^2}$	0	0	0
Sortie $\theta$	0	$\frac{261.2}{s^2}$	0	0
Sortie $\psi$	0	0	$\frac{130.6}{s^2}$	0
Sortie p	$\frac{261.2}{s}$	0	0	0
Sortie q	0	$\frac{261.2}{s}$	0	0
Sortier	0	0	$\frac{130.6}{s}$	0

Tableau 4.3 : Fonctions de transferts.

### ➤ Les graphes

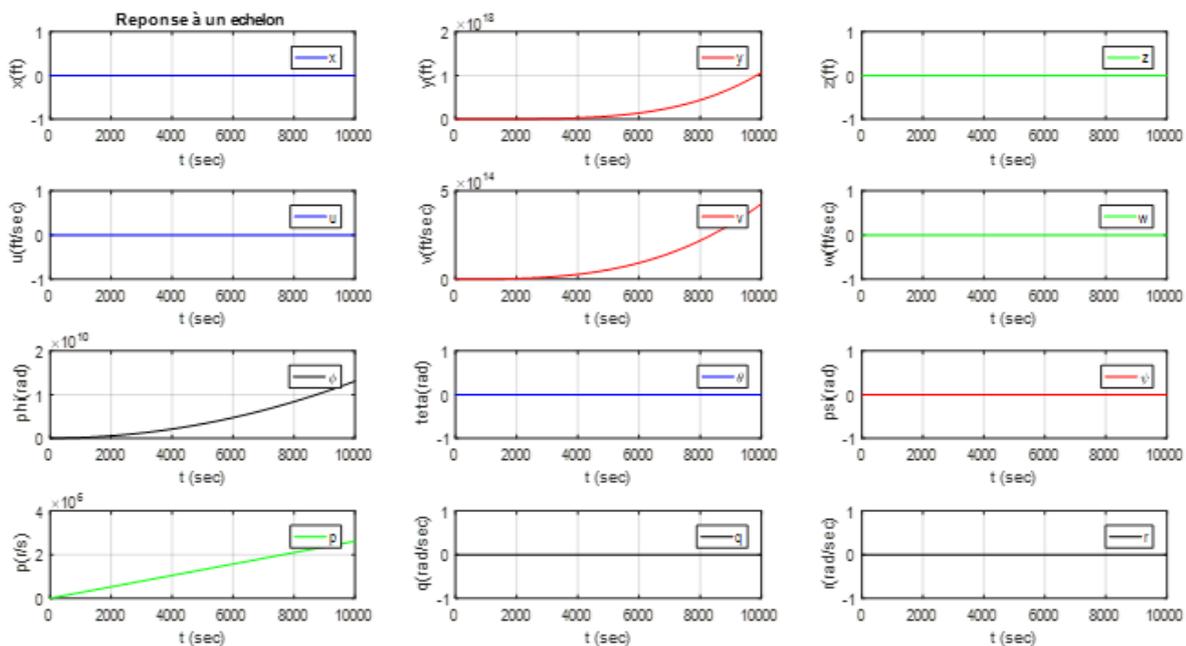


Figure 4.5 : Réponse à un Echelon.

### Discussion des résultats :

Ces derniers résultats montrent la réaction du système à une excitation échelon ou

- Avec une entrée  $T_\phi$  on aura  $y, v, \phi, p$  comme fonctions de transferts
- Avec une entrée  $T_\theta$  on aura  $x, u, \theta, q$
- Avec une entrée  $T_\psi$  on aura  $\psi, r$  et finalement l'entrée  $F$  on aura  $z, w$

Les graphes montrent le changement des paramètres  $y, v, \phi, p$  alors que les autres paramètres  $u, \theta, q, \psi, r, z, w$  ne changent pas

## 4.4 Étude du système

### 4.4.1 La stabilité

La stabilité est l'une des caractéristiques du système qui signifie l'aptitude qu'un système évolue vers une sortie constante lorsqu'on lui applique une entrée constante (échelon dans notre cas).

Un système est dit stable lorsque les pôles de sa fonction de transfert sont à partie réelle négatives ou bien les valeurs propres de la matrice  $A$  ont une partie réelle négative.

Les pôles sont trouvés à partir des racines de l'équation caractéristique suivante [24] :

$$|sI - A| = 0 \quad (4.19)$$

Avec  $I$  : Matrice identité.

### 4.4.2 La contrôlabilité

La contrôlabilité fait partie des propriétés dites structurelles qui caractérisent les systèmes et éventuellement permettent de les classifier, par leurs propriétés algébriques et géométriques. Elle est indispensable dans les applications pour qu'un système puisse être convenablement commandé et permet de construire des lois de commande de façon effective. Cependant, elle sert d'introduction à de nombreuses questions d'une grande importance pratique, comme la planification de trajectoires. [24]

Un système est dit contrôlable en temps T si et seulement si la matrice C est de rang égal à n.

$$\text{Cont} = [ B \ AB \ A^2B \ \dots \ A^{N-1}B ] \quad (4.20)$$

La matrice Cont est appelée matrice de Kalman.

La condition rang Cont = n est appelée condition de Kalman.

### 4.4.3 L'observabilité

On appelle observabilité d'un système la possibilité d'évaluer l'ensemble des grandeurs constitutives du vecteur d'état à partir des mesures effectuées sur le système.

Un système est dit observable (en temps T quelconque) si et seulement si la matrice d'observabilité de Kalman est de rang n (rang Obs=n), on dit alors que Obs est observable. [24]

$$\text{Obs} = [ C \ CA \ \dots \ CA^{n-1} ]^T \quad (4.21)$$

### 4.4.4 Les résultats obtenus :

Les valeurs propres obtenues pour le cas où les conditions initiales nulles

$$\text{valprp} = [0 \ 0 \ 0 \ 0 \ 0 \ 0 \ 0 \ 0 \ 0 \ 0 \ 0 \ 0] \quad (4.22)$$

#### Remarque :

La matrice C (matrice d'observation) est une matrice diagonale 12\*12.

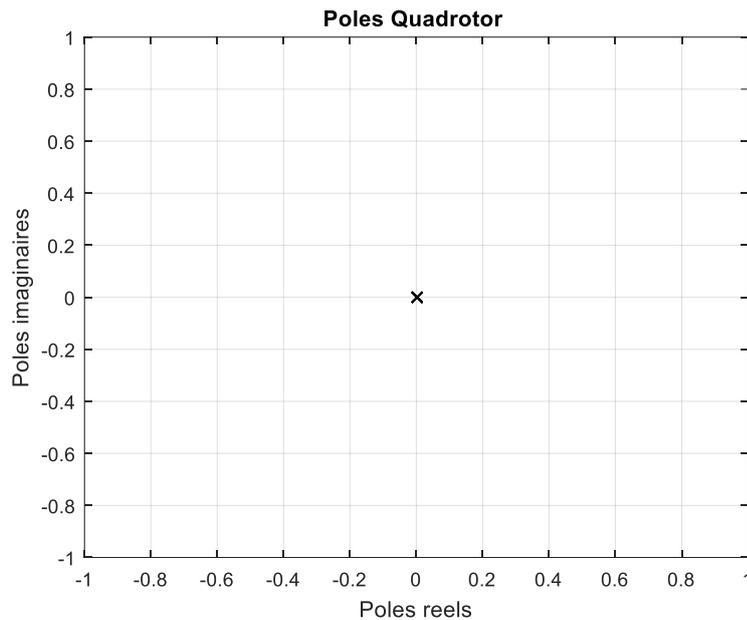


Figure 4.6 : Les pôles de la matrice A.

D'après ces résultats, le système est marginalement stable. Il est également contrôlable et observable.

## 4.5 Control du système :

La conception des contrôleurs dans l'espace d'état permet de spécifier complètement les pôles d'un système, pour obtenir la réponse temporelle voulue.

### 4.5.1 Placement des pôles :

C'est une méthode par laquelle nous pouvons calculer la matrice de gain appropriée pour garantir la stabilité du système.

La stabilité et la dynamique du système sont fixées par  $\sigma(A) = \{\lambda_1 \lambda_2 \dots \lambda_n\}$ .

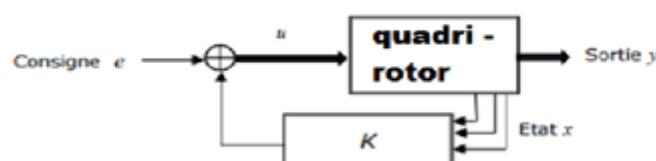
On souhaite commander ce système dans le but d'améliorer les performances par une commande appelée commande par retour d'état :

$$u(t) = e(t) - kx(t) = e(t) - k_1x_1(t) - k_2x_2(t) \dots k_nx_n(t) \quad (4.22)$$

Où

$K = [k_1 \ k_2 \ \dots \ k_n]$  : c'est gain du retour d'état.

Symboliquement, le système en boucle fermée peut être représenté par le schéma suivant :



## Chapitre 4 : Simulation sous MATLAB

Avec cette commande, les équations en boucle fermée s'écrivent :

$$S_{boucle\ fermé} \begin{cases} (A - B.K)x(t) + Be(t) \\ y(t) = Cx(t) \end{cases} \quad (4.23)$$

La matrice d'état du système en boucle fermée est  $-B.K$ , la matrice de commande B et la matrice de sortie C sont inchangées. [25]

### 4.5.2 Les résultats par MATLAB :

En exploitant les résultats dans la partie précédente :

➤ **La commande  $T_\varphi$**

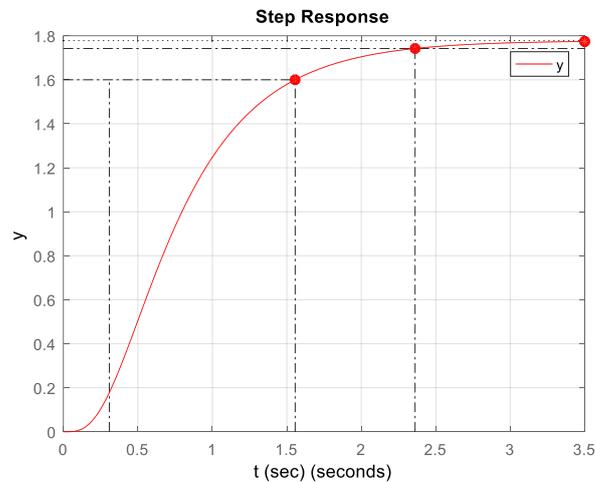
La sortie y :

Les pôles utilisés  $P = [-2 -6 -8 12]$

La fonction de transfert :

$$y = \frac{2560}{s^4 + 30s^3 + 308s^2 + 1224s + 1440}$$

Le graphe :



**Figure 4.7 :** Réponse de la sortie y.

## Chapitre 4 : Simulation sous MATLAB

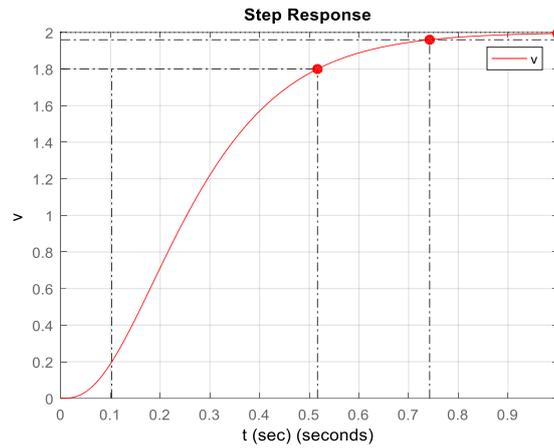
La sortie v :

Les pôles utilisés  $P = [-8 \ -10 \ -16]$

La fonction de transfert :

$$v = \frac{2560}{s^3 + 34s^2 + 368s + 1280}$$

Le graphe :



**Figure 4.8 :** Réponse de la sortie v.

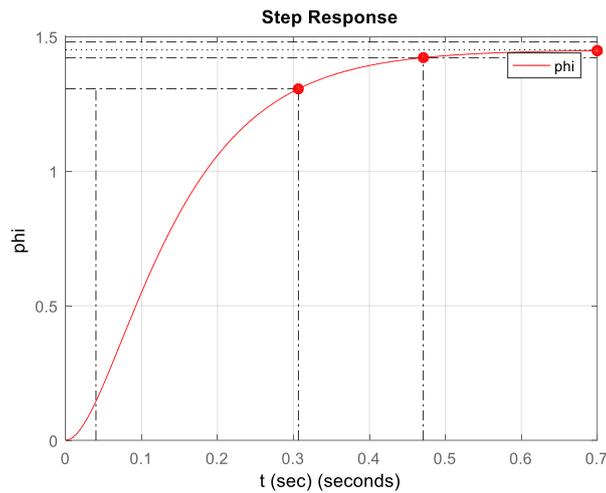
La sortie phi :

Les pôles utilisés  $P = [-10 \ -18]$

La fonction de transfert :

$$phi = \frac{261.2}{s^2 + 28s + 180}$$

Le graphe :



**Figure 4.9 :** Réponse de la sortie phi.

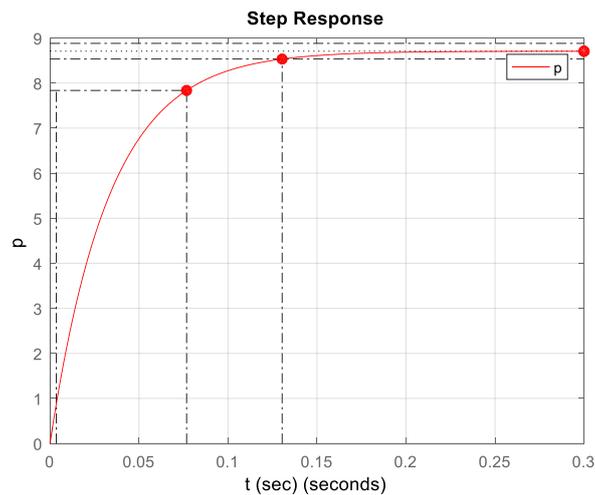
La sortie p :

Le pôle utilisé  $P = [-30]$

La fonction de transfert :

$$p = \frac{261.2}{s + 30}$$

Le graphe :



**Figure 4.10 :** Réponse de la sortie p.

➤ **La commande  $T_\theta$  :**

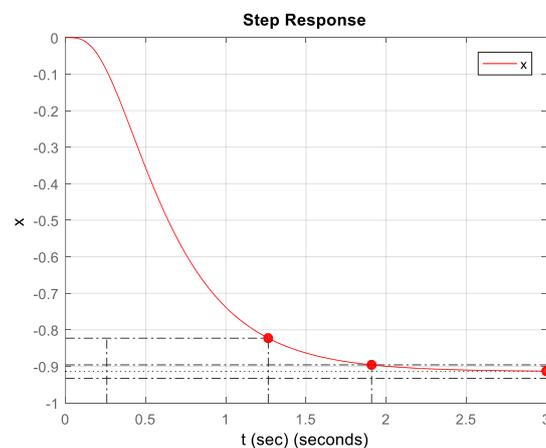
La sortie x :

Les pôles utilisés  $P = [-2,5 -8 -10 14]$

La fonction de transfert :

$$x = \frac{-2560}{s^4 + 34.5s^3 + 412s^2 + 1950s + 2800}$$

Le graphe :



**Figure 4.11 :** Réponse de la réponse x.

## Chapitre 4 : Simulation sous MATLAB

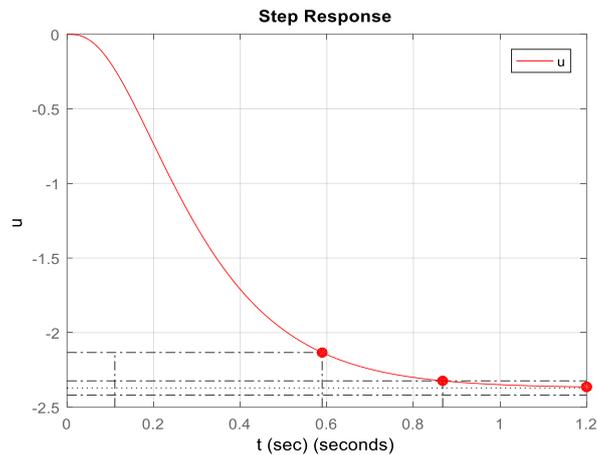
La sortie u :

Les pôles utilisés  $P = [-2 \ -10 \ -18]$

La fonction de transfert :

$$u = \frac{-2560}{s^3 + 34s^2 + 348s + 1080}$$

Le graphe :



**Figure 4.12 :** Réponse de la sortie u.

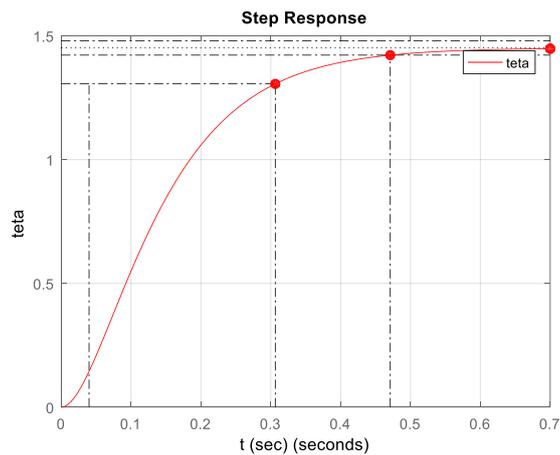
La sortie teta :

Les pôles utilisés  $P = [-10 \ -18]$

La fonction de transfert :

$$teta = \frac{261.2}{s^2 + 28s + 180}$$

Le graphe :



**Figure 4.13 :** Réponse de la sortie teta.

## Chapitre 4 : Simulation sous MATLAB

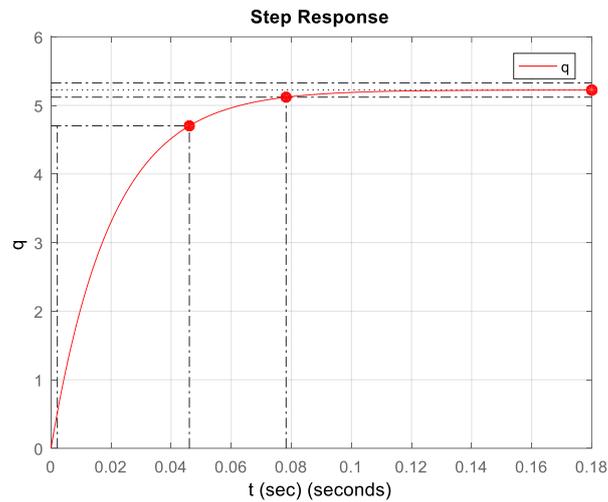
La sortie q :

Le pôle utilisé  $P = [-50]$

La fonction de transfert :

$$q = \frac{261.2}{s + 50}$$

Le graphe :



**Figure 4.14 :** Réponse de la sortie q.

➤ **La commande  $T_\psi$  :**

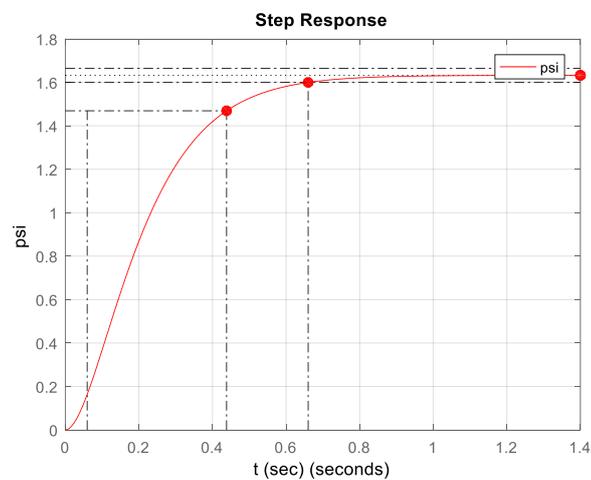
La sortie psi :

Les pôles utilisés  $P = [-8 \ -10]$

La fonction de transfert :

$$psi = \frac{130.6}{s^2 + 18s + 80}$$

Le graphe :



**Figure 4.15 :** Réponse de la sortie psi.

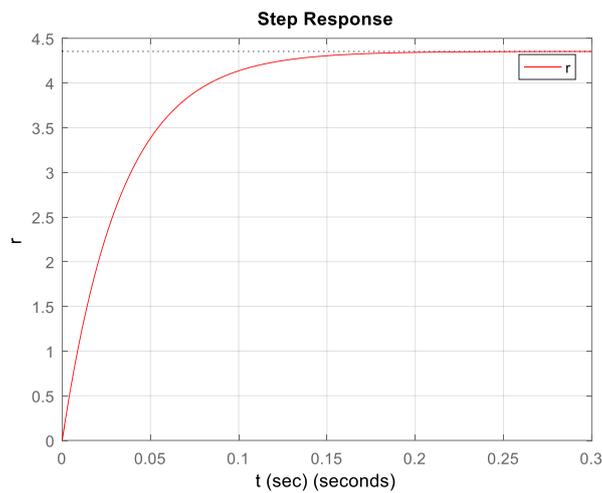
La sortie r :

Le pôle utilisé  $P = [-30]$

La fonction de transfert :

$$r = \frac{130.6}{s + 30}$$

Le graphe :



**Figure 4.16 :** Réponse de la sortie r.

➤ **La commande F :**

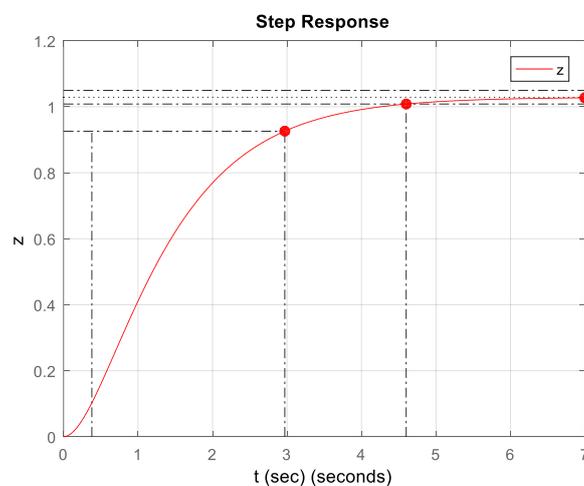
La sortie z :

Les pôles utilisés  $P = [-1 -2]$

La fonction de transfert :

$$z = \frac{2.058}{s^2 + 3s + 2}$$

Le graphe :



**Figure 4.17 :** Réponse de la sortie z.

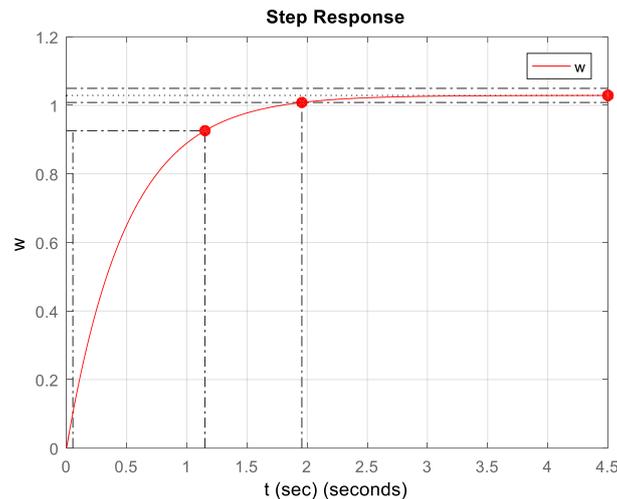
La sortie de w :

Le pôle utilisé  $P = [-2]$

La fonction de transfert :

$$w = \frac{2.058}{s + 2}$$

Le graphe :



**Figure 4.18 :** Réponse de la sortie w.

### Discussion des résultats :

La technique du placement des pôles (ou commande par retour d'état) permet la stabilité au système et mène les différents paramètres à la convergence tout en choisissant les gains pour chaque paramètres.

Le choix des gains se fait d'une manière à diminuer l'amplitude du signal et l'approximer à l'amplitude de l'échelon.

## 4.6 Contrôleur PID :

L'intérêt de cette approche est de présenter une stratégie de commande linéaire pouvant être appliquée au quadricopter pour permettre à ce dernier de se stabiliser. [9]

### 4.6.1 Définition :

La commande par régulation PID (P : proportionnelle, I : intégrale et D : dérivée) est certainement la structure de commande la plus couramment utilisée dans l'industrie. Elle est en effet imposée dans plus de 90% des architectures de la commande. 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 d'application.

Les contrôleurs PID permettent d'engendrer à partir de la sortie du comparateur (c'est-à-dire l'écart existant entre la consigne et la grandeur à réguler) un signal proportionnel à l'erreur et sa dérivée d'une part, et à son intégrale d'autre part.

### 4.6.2 Description :

Un régulateur PID se décompose essentiellement en trois actions qui sont les suivantes [9] :

➤ **Action Proportionnelle :**

L'action proportionnelle corrige de manière instantanée, donc rapide, tout écart de la grandeur à réguler, elle permet d'éliminer les grandes inerties du système. Pour but de diminuer l'écart de réglage et de rendre le système plus rapide, on augmente le gain, mais est limité car ça risque de déstabilisé le système. Cette action est appliquée lorsqu'on n'a pas besoin d'une grande précision.

➤ **Action Intégrale :**

Elle complète l'action proportionnelle. Elle permet d'éliminer l'erreur résiduelle en régime permanent. Afin de rendre le système plus dynamique (diminuer le temps de réponse), on diminue l'action intégrale mais, ceci induit l'augmentation du déphasage ce qui provoque l'instabilité en boucle fermée. L'action intégrale est utilisée lorsqu'on désire avoir en régime permanent, une précision parfaite, en outre, elle permet de filtrer la variable à régler d'où l'utilité pour le réglage des variables bruitées.

➤ **Action Dérivée**

L'action dérivée, en compensant les inerties dues au temps mort, accélère la réponse du système et améliore la stabilité, en permettant notamment un amortissement rapide des oscillations dues à l'apparition d'une perturbation ou à une variation subite de la consigne.

Cette action est utilisée dans l'industrie pour le réglage des variables lentes, elle n'est pas recommandée pour le réglage d'une variable bruitée ou trop dynamique. En dérivant un bruit, son amplitude risque de devenir plus importante que celle du signal utile.

### 4.6.3 Correcteur PID :

L'intérêt du correcteur PID est d'intégrer les effets positifs des trois correcteurs, La détermination des coefficients  $k_p$ ,  $T_i$ ,  $T_d$ , du correcteur PID permet d'améliorer à la fois la précision ( $T_D$  et  $k_p$ ) la stabilité ( $T_d$ ) et la rapidité ( $T_d$  et  $k_p$ ) . Le réglage d'un correcteur PID est en général assez complexe mais des méthodes pratiques de réglages permettent d'obtenir des bons résultats.

On note que ces corrections peuvent ne pas être exactement précise, parce que  $k_p$  ,  $k_I$  et  $k_d$  dépendent l'un de l'autre. En fait, changer un de ces variables peut changer l'effet des deux autres.

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

$$U(t) = k_p \varepsilon(t) + k_I \int_0^t \varepsilon(t) d\tau + k_d \frac{d\varepsilon(t)}{dt} \quad (4.24)$$

Où :

$k_p$  : Gain proportionnelle

$k_I$  : Gain intégral

$k_d$  : Gain dérivé.

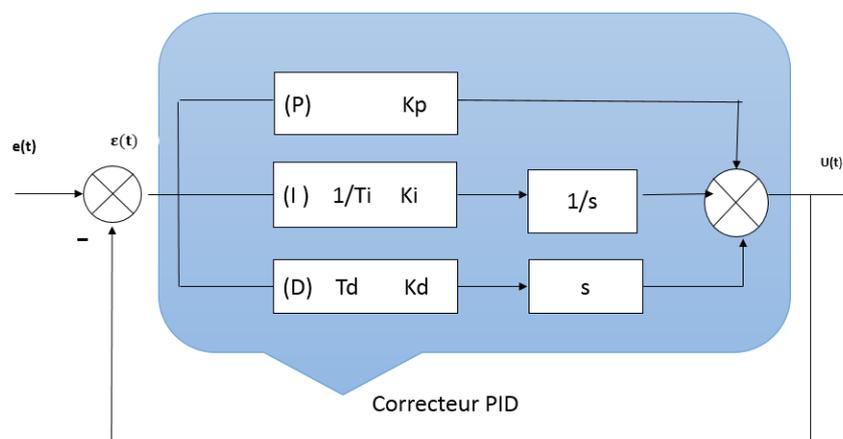
$\varepsilon(t)$  : Signal d'entrée du correcteur (ou signal d'erreur).

$U(t)$  : Signal de sortie du correcteur (ou signal de commande).

Avec :

$$T_i = \frac{k_p}{k_I}, \quad T_d = \frac{k_d}{k_p}$$

### 4.6.4 Présentation schématique :



**Figure 4.19** : Présentation du régulateur PID.

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

$$H_{PID}(s) = \frac{U(s)}{\varepsilon(s)} = k_p + \frac{k_I}{s} + k_d s \quad (4.25)$$

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

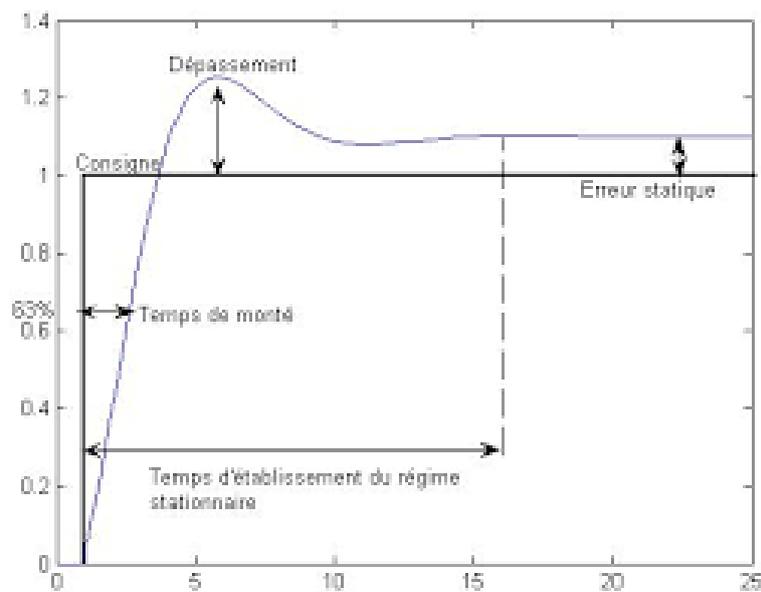
paramètres	Temps de montée	Temps de stabilisation	dépassement	Erreur statique
$k_p$	Diminue	Augmente	augmente	Diminue
$k_I$	Diminue	Augmente	augmente	Annule
$k_d$	-	Diminue	Diminue	-

**Tableau 4.4 :** l'influence de l'augmentation des paramètres du PID sur le comportement du système en boucle fermé.

#### 4.6.5 La réponse d'un système :

La réponse d'un système de fonction de transfert  $G(p)$  correspond à la forme du signal de sortie  $y(t)$  lorsque le système est excité par un signal d'entrée connu  $u(t)$ . On distingue plusieurs types de réponses selon le signal d'entrée. Certaines réponses sont très couramment étudiées pour comprendre le fonctionnement d'un système.

La réponse d'un système dépend de la valeur de la sortie à l'instant initial considéré. Cependant, pour des fins de simplification ; on s'intéresse à des réponses pour lesquelles les conditions initiales sont nulles.



**Figure 4.20 :** Caractéristiques de la réponse d'un système

### 4.6.6 Présentation du schéma réalisé

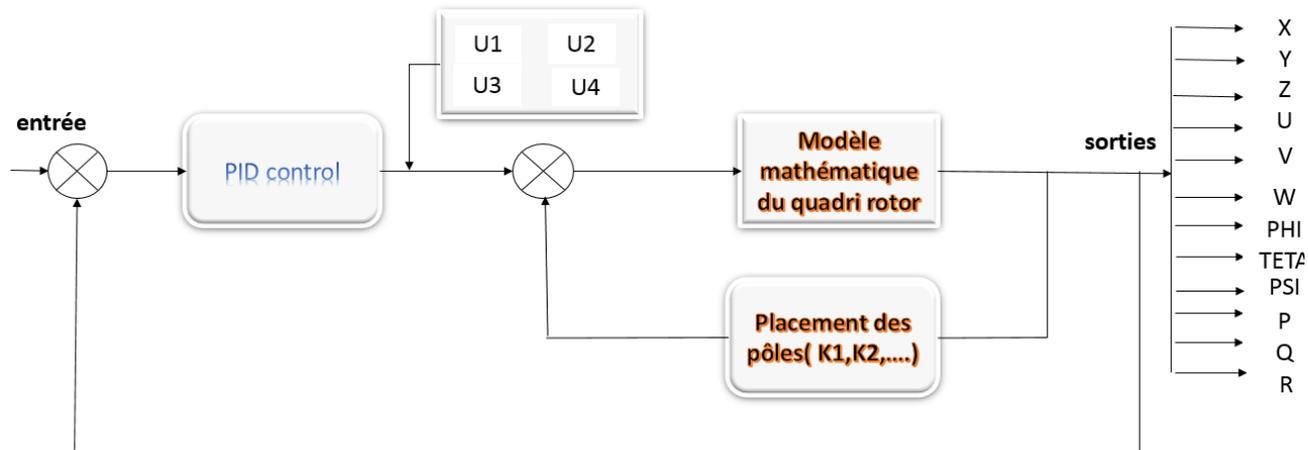


Figure 4.21 : Présentation du schéma réalisé.

### 4.6.7 Simulation par MATLAB :

➤ La commande  $T_\varphi$  :

La sortie  $y$  :

Les pôles utilisés  $k_p = 2$ ,  $k_I = 1$  et  $k_d = 0.5$

La fonction de transfert :

$$y = \frac{1280s^2 + 5120s + 2560}{s^5 + 30s^4 + 308s^3 + 2504s^2 + 6560s + 2560}$$

Le graphe :

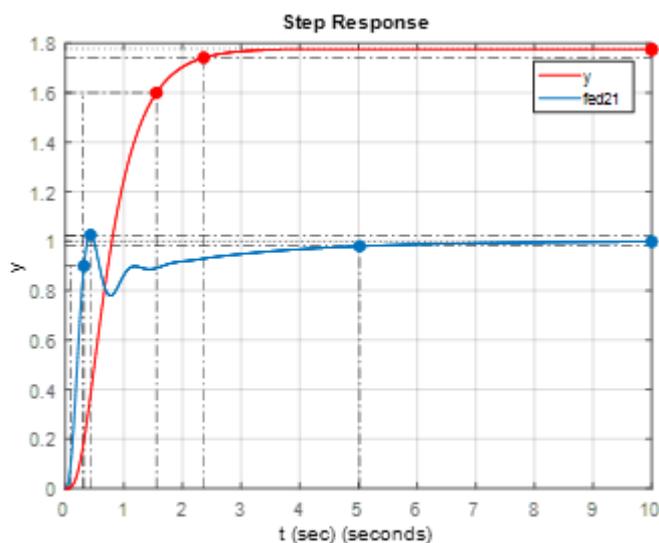


Figure 4.22 : Réponse de la sortie  $y$ .

## Chapitre 4 : Simulation sous MATLAB

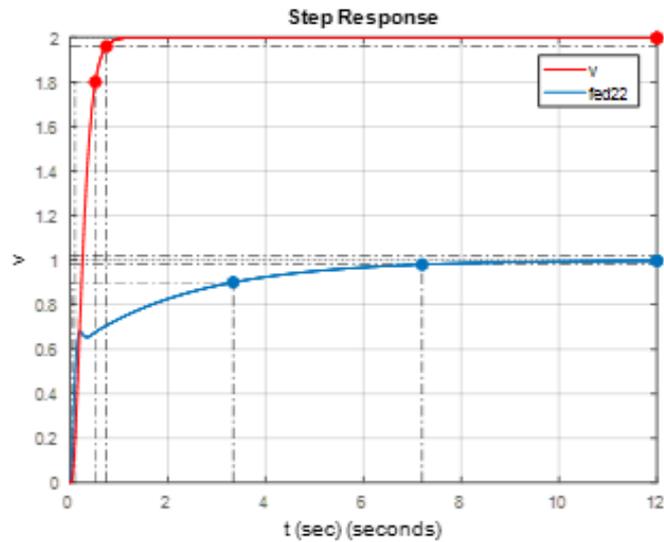
La sortie v :

Les pôles utilisés  $k_p = 0.8$  ,  $k_I = 0.5$  et  $k_d = 0.1$

La fonction de transfert :

$$v = \frac{256s^2 + 2048s + 1280}{s^4 + 34s^3 + 624s^2 + 3328s + 1280}$$

Le graphe :



**Figure 4.23 :** Réponse de la sortie v.

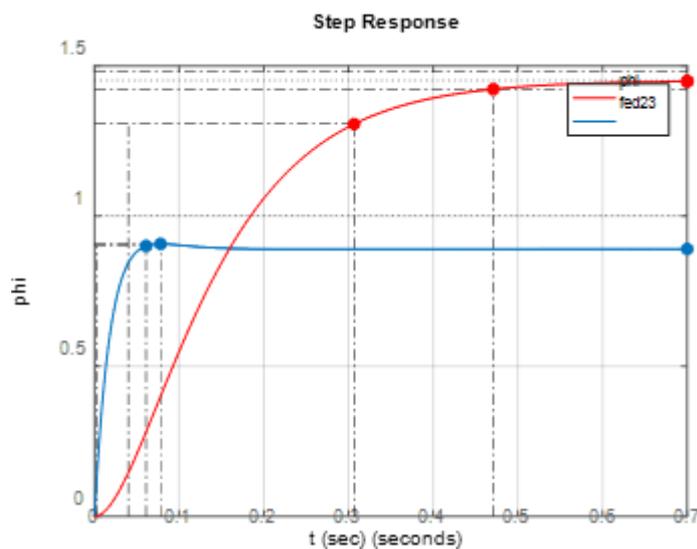
La sortie phi :

Les pôles utilisés  $k_p = 5.5$  ,  $k_I = 0.1$  et  $k_d = 0.2$

La fonction de transfert :

$$phi = \frac{52.24s^2 + 21437s + 126.12}{s^3 + 80.24s^2 + 1617s + 26.12}$$

Le graphe :



**Figure 4.24 :** Réponse de la sortie phi.

## Chapitre 4 : Simulation sous MATLAB

La sortie q :

Les pôles utilisés  $k_p = 3$  ,  $k_I = 0.01$  et  $k_d = 0$

La fonction de transfert :

$$q = \frac{783.6s + 2.612}{s^2 + 813.6s + 2.612}$$

Le graphe :

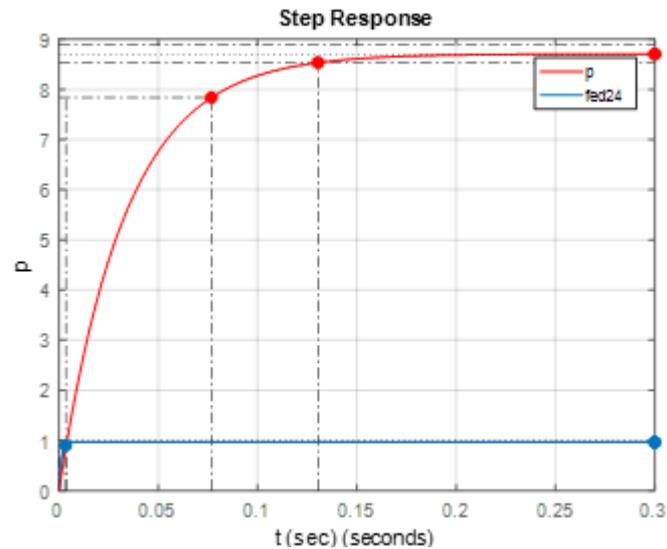


Figure 4.25 : Réponse de la sortie q.

➤ **La commande  $T_\theta$  :**

La sortie x :

Les pôles utilisés  $k_p = -3$  ,  $k_I = 0$  et  $k_d = 0$

La fonction de transfert :

$$X = \frac{7680}{s^4 + 34.5s^3 + 412s^2 + 1950s + 10480}$$

Le graphe :

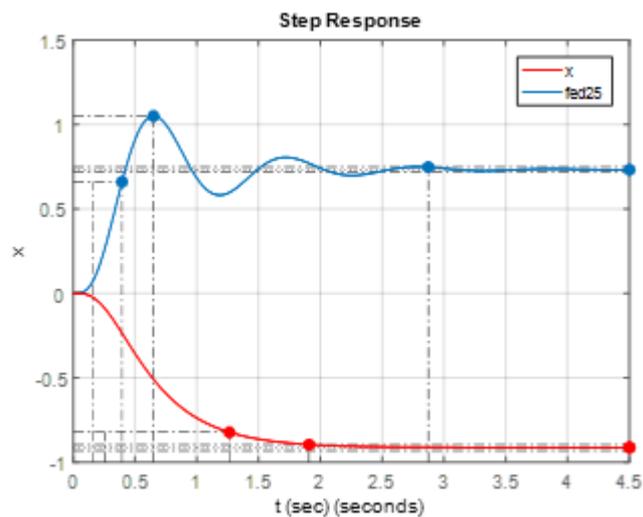


Figure 4.26 : Réponse de la sortie x.

## Chapitre 4 : Simulation sous MATLAB

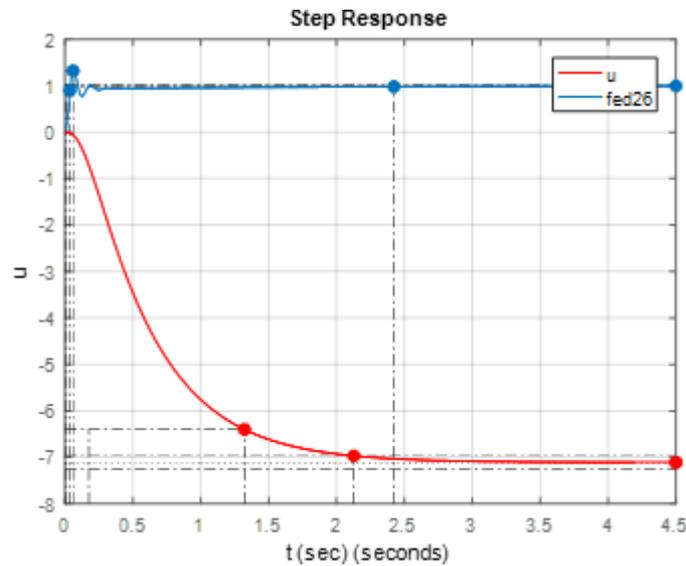
La sortie u :

Les pôles utilisés  $k_p = -2$  ,  $k_I = -1$  et  $k_d = -1$

La fonction de transfert :

$$U = \frac{2560s^3 + 5120s + 2560}{s^4 + 30s^3 + 2496s^2 + 5480s + 2560}$$

Le graphe :



**Figure 4.27 :** Réponse de la sortie u.

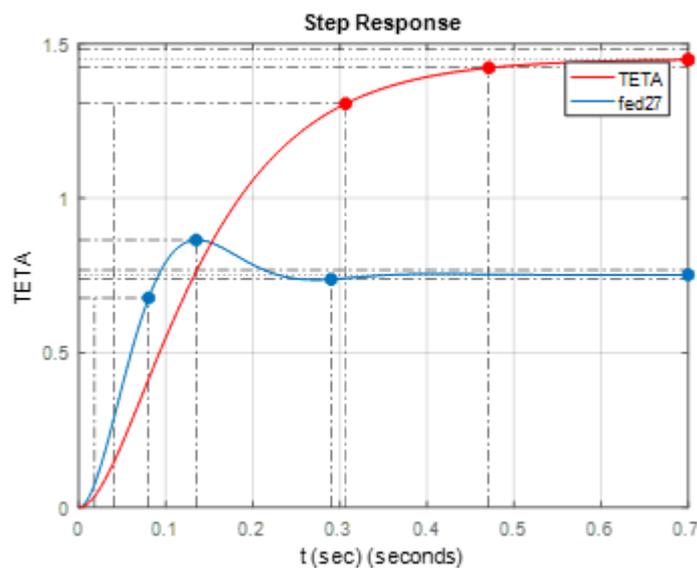
La sortie teta :

Les pôles utilisés  $k_p = 2.1$  ,  $k_I = 0$  et  $k_d = 0$

La fonction de transfert :

$$TETA = \frac{548.5}{s^2 + 28s + 728.5}$$

Le graphe :



**Figure 4.28 :** Réponse de la sortie teta.

## Chapitre 4 : Simulation sous MATLAB

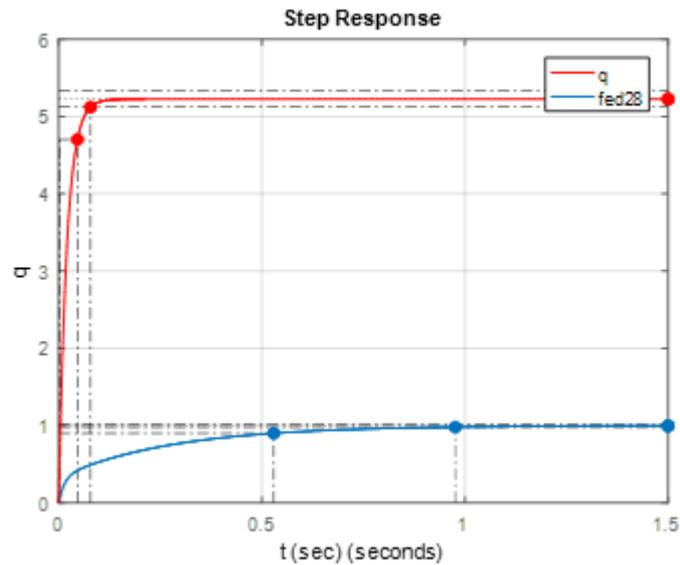
La sortie q :

Les pôles utilisés  $k_p = 0.1$  ,  $k_I = 1$  et  $k_d = 0$

La fonction de transfert :

$$q = \frac{26.12s + 261.2}{s^2 + 76.12s + 261.2}$$

Le graphe :



**Figure 4.29 :** Réponse de la sortie q.

➤ **La commande  $T_\psi$  :**

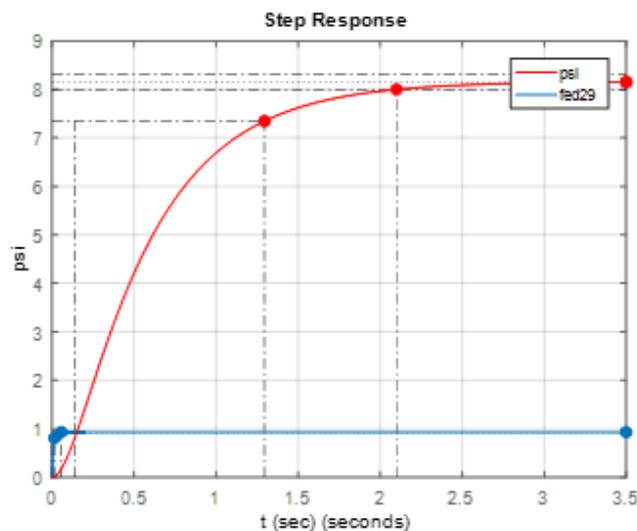
La sortie psi :

Les pôles utilisés  $k_p = 0.1$  ,  $k_I = 1$  et  $k_d = 0$

La fonction de transfert :

$$psi = \frac{130.6s + 143.7}{s^2 + 140.6s + 159.7}$$

Le graphe :



**Figure 4.30 :** Réponse de la sortie psi.

## Chapitre 4 : Simulation sous MATLAB

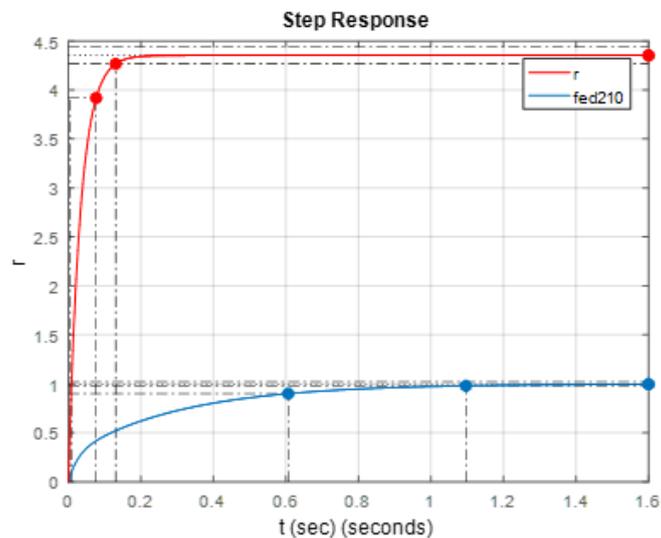
La sortie r :

Les pôles utilisés  $k_p = 0.1$  ,  $k_I = 1$  et  $k_d = 0$

La fonction de transfert :

$$r = \frac{13.06s + 130.6}{s^2 + 43.06s + 130.6}$$

Le graphe :



**Figure 4.31 :** Réponse de la sortie r.

➤ **La commande F :**

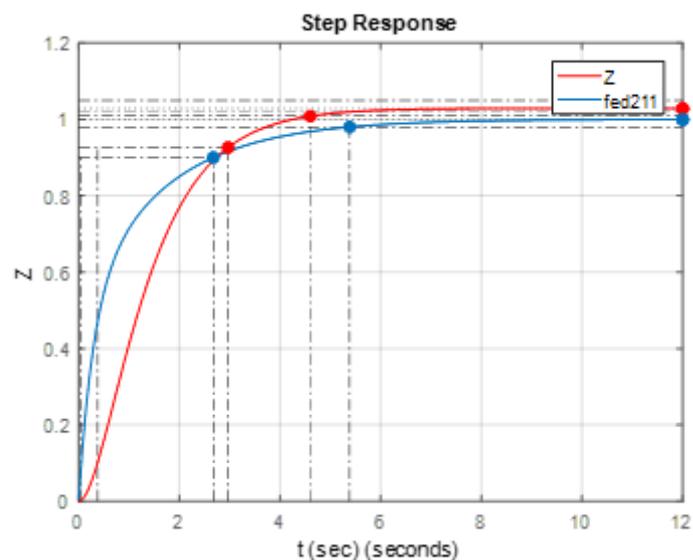
La sortie z :

Les pôles utilisés  $k_p = 2$  ,  $k_I = 1$  et  $k_d = 1$

La fonction de transfert :

$$z = \frac{2.058s^2 + 4.116s + 2.058}{s^3 + 5.058s^2 + 6.116s + 2.058}$$

Le graphe :



**Figure 4.32 :** Réponse de la sortie z

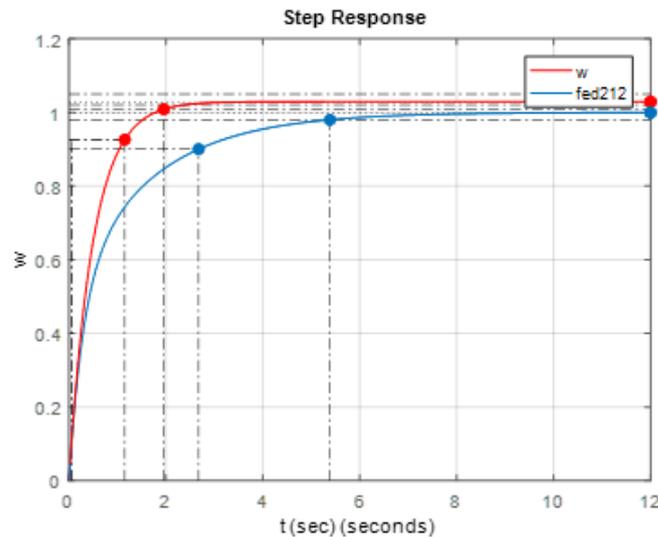
La sortie w :

Les pôles utilisés  $k_p = 1$  ,  $k_I = 1$  et  $k_d = 0$

La fonction de transfert :

$$w = \frac{2.058s + 2.058}{s^2 + 4.058s + 2.058}$$

Le graphe :



**Figure 4.33 :** Réponse de la sortie w.

### Discussion des résultats :

Les graphes présentés la dessus montrent la réponse du système à la commande PID.

Le choix des gains s'est fait pour optimiser entre un bon temps de réponse du système et une erreur statique presque nulle, également pour diminuer le dépassement et le temps de stabilisation. Toute ces paramètres qui dépendent des gains des régulateur ( $k_i$ ,  $k_d$ ,  $k_p$ ). On peut aussi choisir les gains d'une manière à imposer une valeur finale au paramètre.

## 4.7 L'organigramme

Pour qu'on puisse résumer la partie de la modélisation, le programme réalisé avec MATLAB est illustré dans l'organigramme suivant :

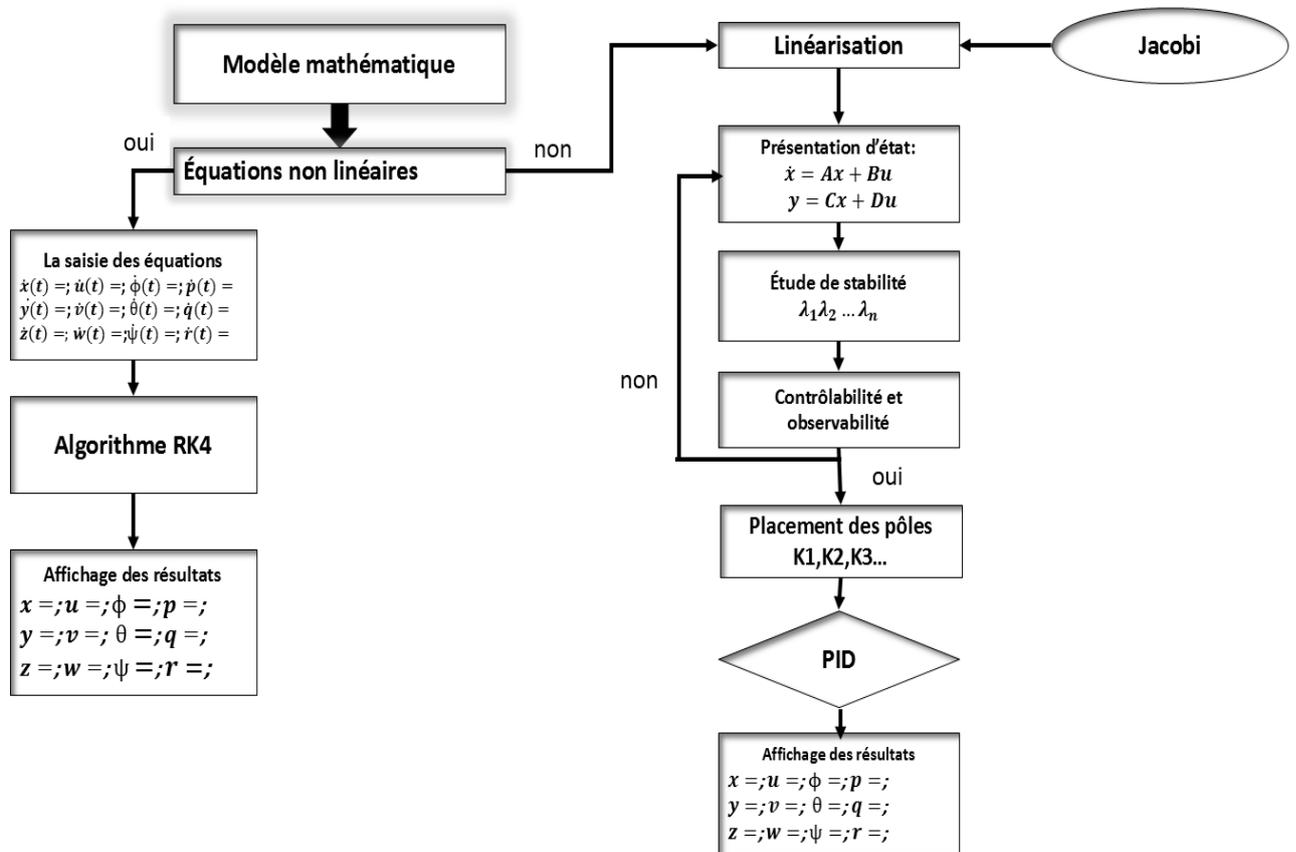


Figure 4.34 : Organigramme du programme MATLAB.

## 4.8 Conclusion :

Dans ce chapitre, on a pu faire la résolution linéaire et non linéaire du modèle mathématique du quadrirotor.

La résolution non linéaire a montré que le choix du pas h est très important pour garantir la convergence des courbes des équations non linéaire et l'approche de la solution exacte vers la solution numérique

La résolution linéaire nous a permis de tracer les réponses du système ainsi de contrôler la stabilité de ce dernier en appliquant une commande par retour d'état et une commande PID.

# Chapitre 5 : Test et réalisation

### 5.1 Introduction

Ce chapitre est consacré à la programmation et aux tests matérielles et logicielles de notre système. En premier lieu, un test individuel est fait pour chaque capteur pour vérifier son fonctionnement. En second lieu, en termes de programmation, on commence par le schéma électrique et l'organigramme suivi pour atteindre notre objectif. Par la suite, un test de vol est fait pour valider notre carte autopilote.

### 5.2 Test statique

Dans ce cas, on a procédé à des tests unitaires, c'est à dire tester chaque module isolé pour visualiser de près son fonctionnement, ainsi que la bonne méthode d'utilisation.

#### 5.2.1 Configuration de GPS

Pour configurer et tester le bon fonctionnement du GPS, on utilise le logiciel u-center. On suit les étapes suivantes :

- On réalise le montage en branchant le GPS avec MEGA c'est à dire le VCC, GND, TX et RX. On connecte la carte avec l'ordinateur via un câble USB et on lance l'application u-center.
- On choisit le port de connexion (COM 3) et le 9600 qui est pris comme baud. La configuration de GPS est faite grâce à un fichier téléchargé qu'on l'extrait vers le logiciel, lorsque c'est terminé, on doit la sauvegarder dans la mémoire flash du module pour qu'au prochain démarrage les paramètres restent inchangés.

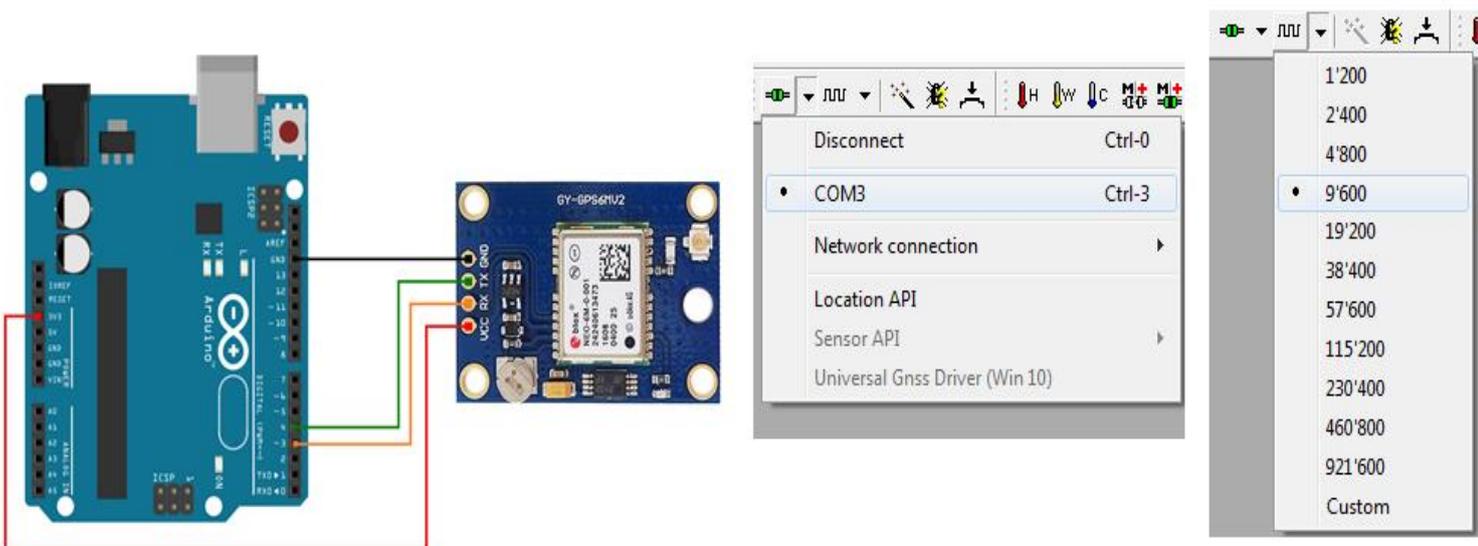


Figure 5.1 : Les différentes étapes du branchage du GPS.

## Chapitre 5 : Test et réalisation

Après avoir terminé toutes ces étapes, l'interface du logiciel u-center s'affichera comme la figure 5.2. Par conséquent, on peut lire les différentes informations comme : longitude, latitude, altitude nombre de satellite, localisation sur la carte du monde ... etc.

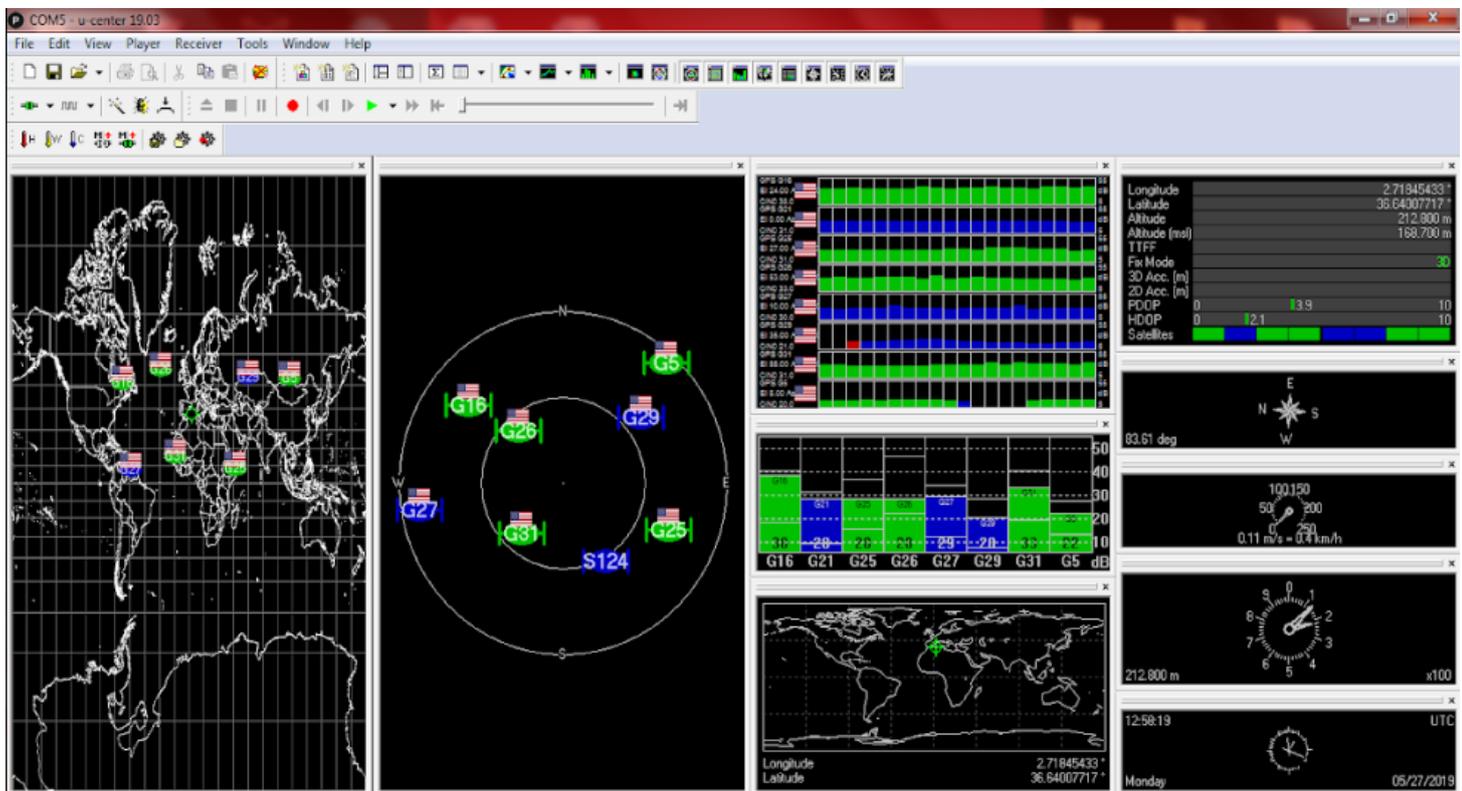


Figure 5.2 : Interface u-center après configuration du GPS.

Pour vérifier le bon fonctionnement du GPS, on a cherché les coordonnées réelles de notre emplacement à TIPAZA. En comparant les résultats des figures 5.2 et 5.3, on remarque que les mêmes informations qui s'affichent. En conséquence, notre GPS fonctionne parfaitement.



Figure 5.3 : Localisation TIPAZA.

### 5.2.2 Configuration de l'MPU

Pour tester le bon fonctionnement de l'MPU, le logiciel Arduino est utilisé en suivant les étapes suivantes :

- Le montage est réalisé en branchant MPU avec la carte Arduino MEGA c'est à dire le VCC, GND, SCL et SDA. On connecte la carte avec l'ordinateur via un câble USB et on lance l'application Arduino.
- Après avoir écrit le programme et insérer dans le logiciel, on doit choisir le type de la carte et le port de connexion (COM 4).
- On flashe le programme dans la carte MEGA, et on lance le moniteur série.

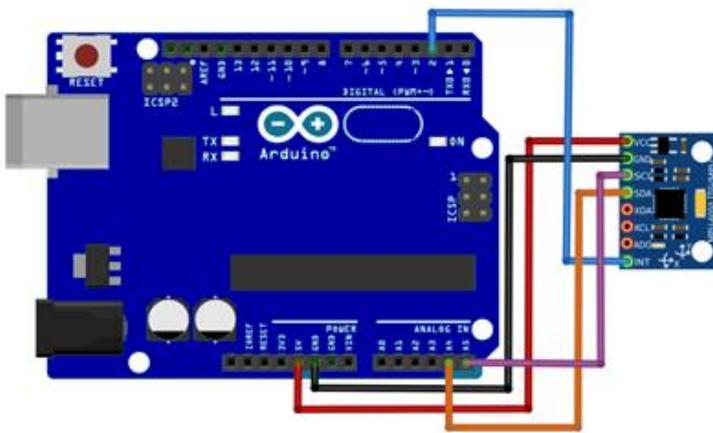


Figure 5.4 : Branchage de MPU.

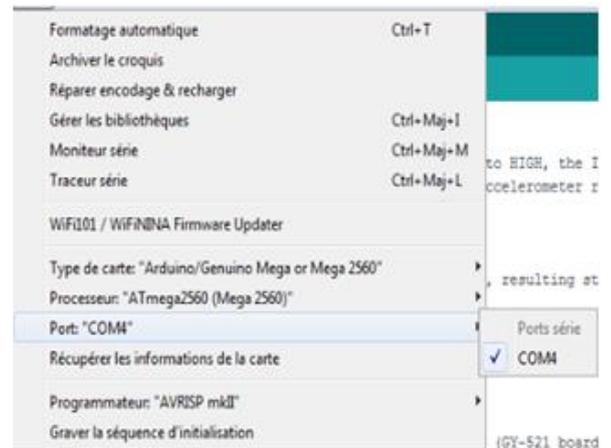


Figure 5.5 : Choix de la carte et port.

Après avoir terminé toutes ces étapes, le moniteur série affichera les données de l'accéléromètre et le gyromètre. Lorsqu'on fait bouger le capteur, le changement des données, se fait immédiatement ce qui indique le bon fonctionnement de l'MPU.

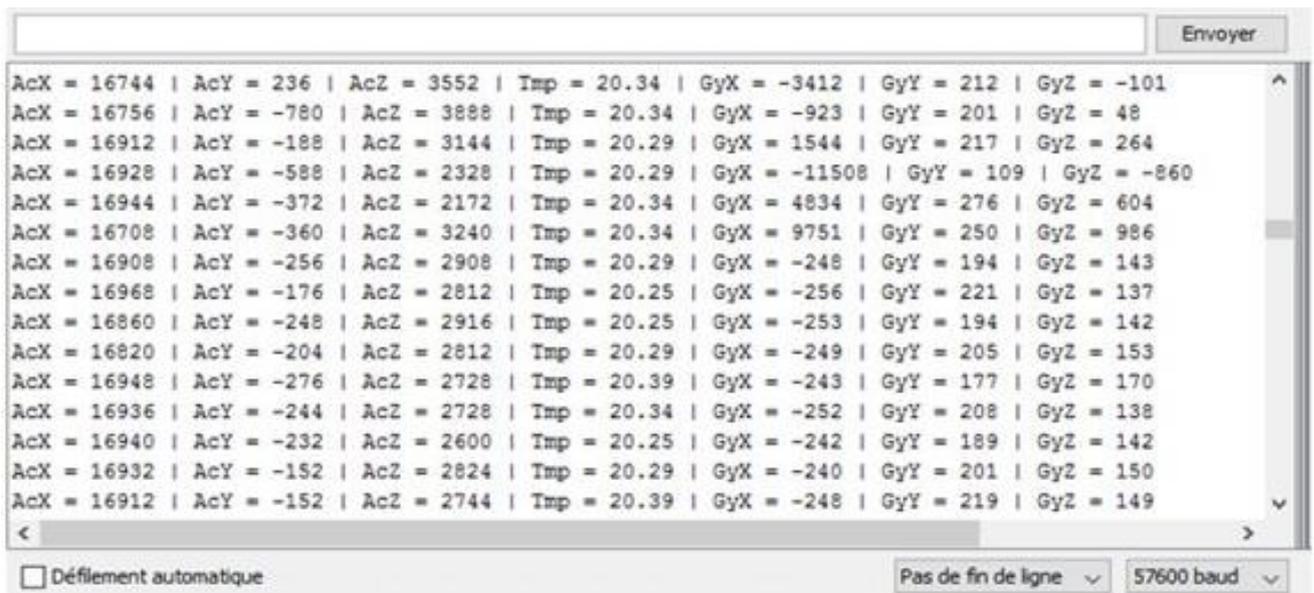


Figure 5.6 : Affichage des données accéléromètre et gyromètre.

### 5.2.3 Calibration des moteurs et ESC

Pour faire voler un drone, une calibration des moteurs est impérative, pour cela, on doit vérifier le fonctionnement de chaque moteur avec l'ESC individuellement sans les hélices par mesure de sécurité. Les moteurs ne peuvent pas être directement alimentés en courant continu, nous devons passer par un ESC, qui va recevoir un courant continu directement de la batterie, puis va le transformer en signal adapté au moteur. On peut schématiser le branchement à effectuer de cette manière :

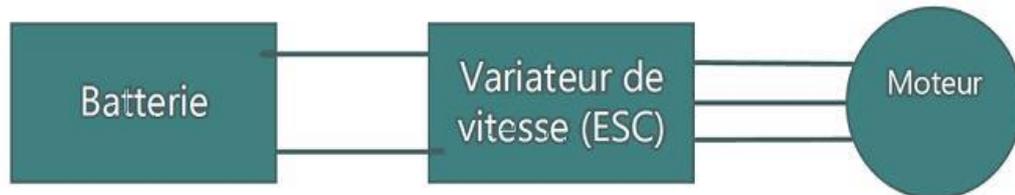


Figure 5.7 : Batterie, ESC et moteur.

Avec le branchement précédent, on rajoute un récepteur Futaba pour recevoir le signal depuis la radio commande. A l'aide de la manette, on actionne vers le haut le joystick gauche, qui est le contrôleur des gaz (Throttle), le moteur commence à tourner, pour l'arrêter on fait descendre le joystick. On doit faire pour les quatre moteurs le même test, et on constate le bon fonctionnement des moteurs.

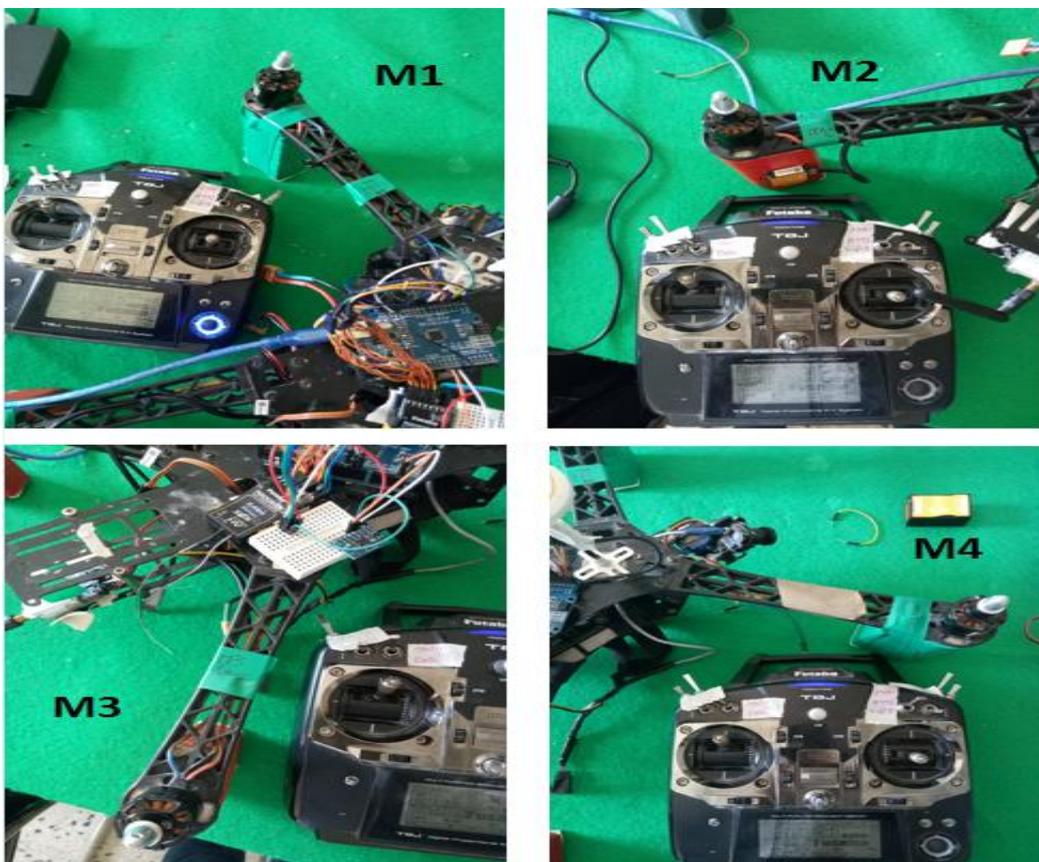


Figure 5.8 : Calibration des 4 moteurs.

### 5.2.4 Test du Bluetooth

Pour tester le bon fonctionnement de Bluetooth, on utilise le logiciel Arduino en respectant les étapes suivantes :

- Le montage est réalisé en branchant le Bluetooth avec la carte Arduino MEGA c'est à dire le VCC, GND, TX et RX. On connecte la carte avec l'ordinateur via un câble USB et on lance l'application Arduino.
- Après avoir écrit le programme et l'insérer dans le logiciel, on doit choisir le type de la carte et le port de connexion (COM 3).
- On flashe le programme dans la carte MEGA, on lance le moniteur série et l'application « Bluetooth Terminal HC-05 » qui est téléchargé par le Play Stor (Téléphone).

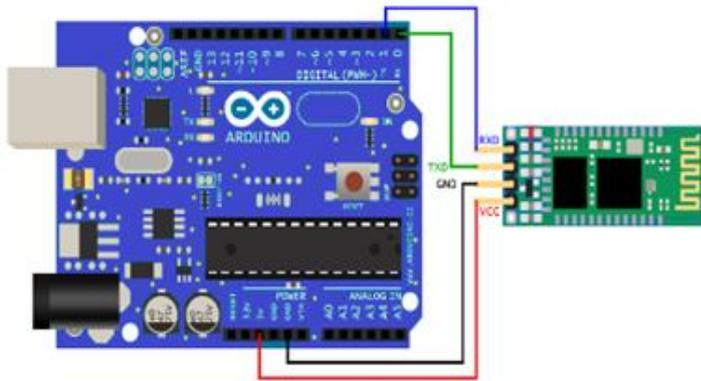


Figure 5.9 : Branchage de Bluetooth.

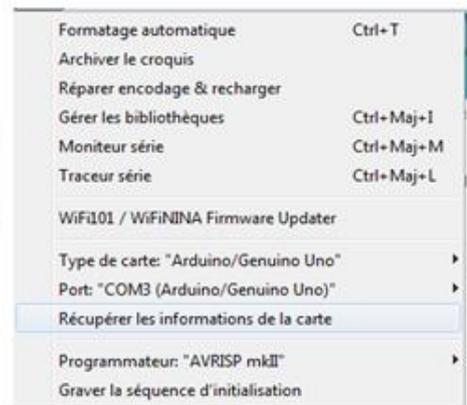


Figure 5.10 : Choix de la carte et port.

Après avoir terminé toutes ces étapes, le moniteur série affichera une fenêtre ou on peut communiquer avec l'application comme indiqué sur la figure 5.11. La transmission des messages se fait facilement et le Bluetooth fonctionne parfaitement.

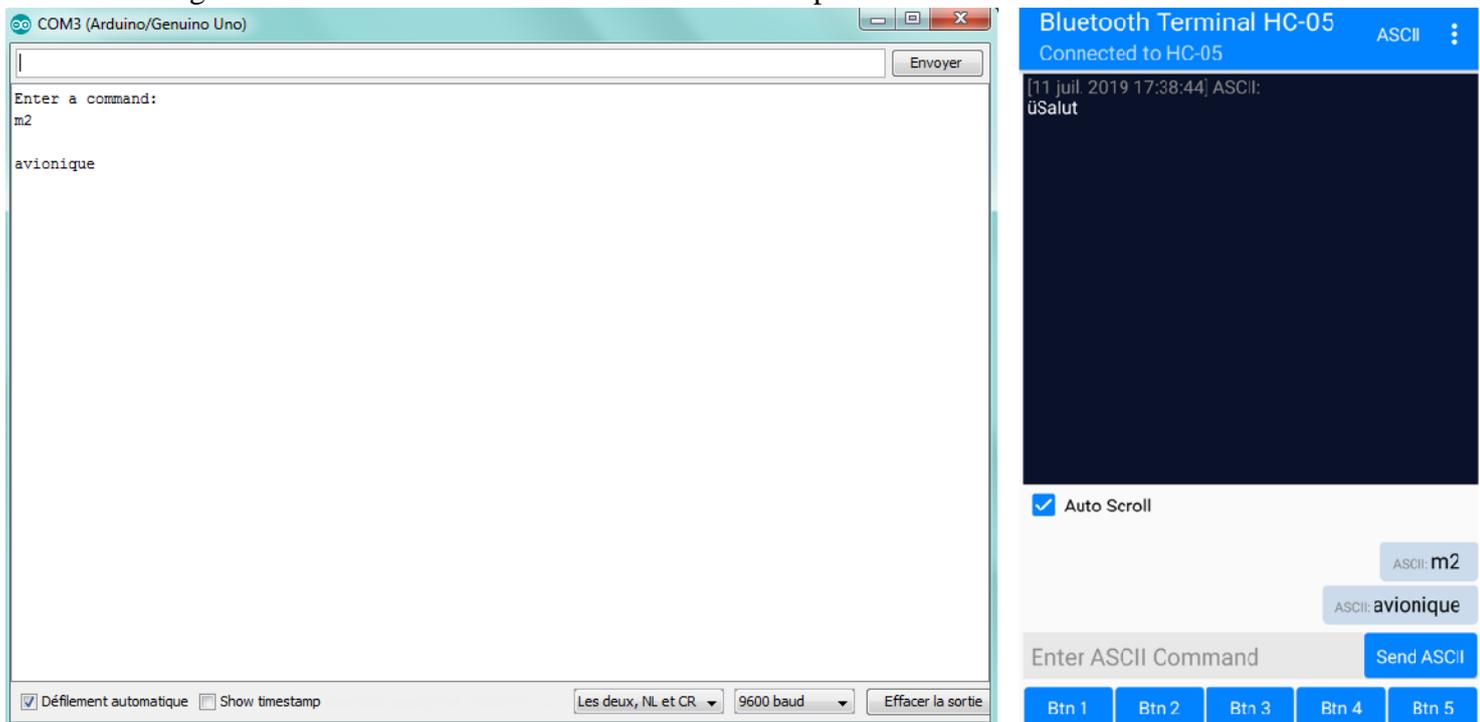


Figure 5.11 : Transmission des messages entre PC et téléphone.

### 5.3 Réalisation de la carte d'acquisition

Avant de réaliser la carte de l'autopilote, on a besoin d'une carte d'acquisition qui doit être fixé au-dessus du microprocesseur MEGA.

#### 5.3.1 Schéma de la carte d'acquisition sur Proteus ARES

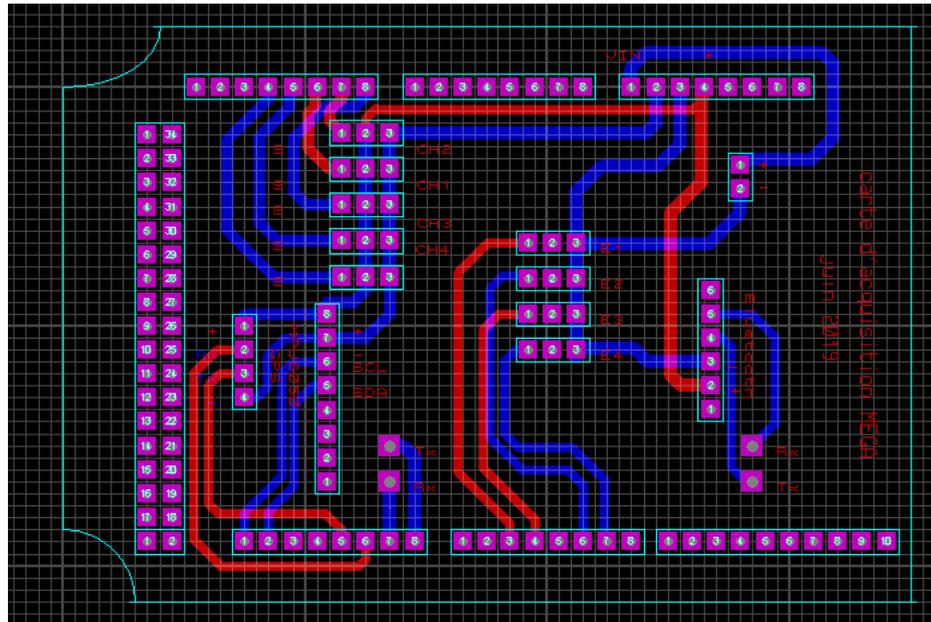


Figure 5.12 : Circuit de la carte d'acquisition sur ARES.

#### 5.3.2 Vue 3D de la carte

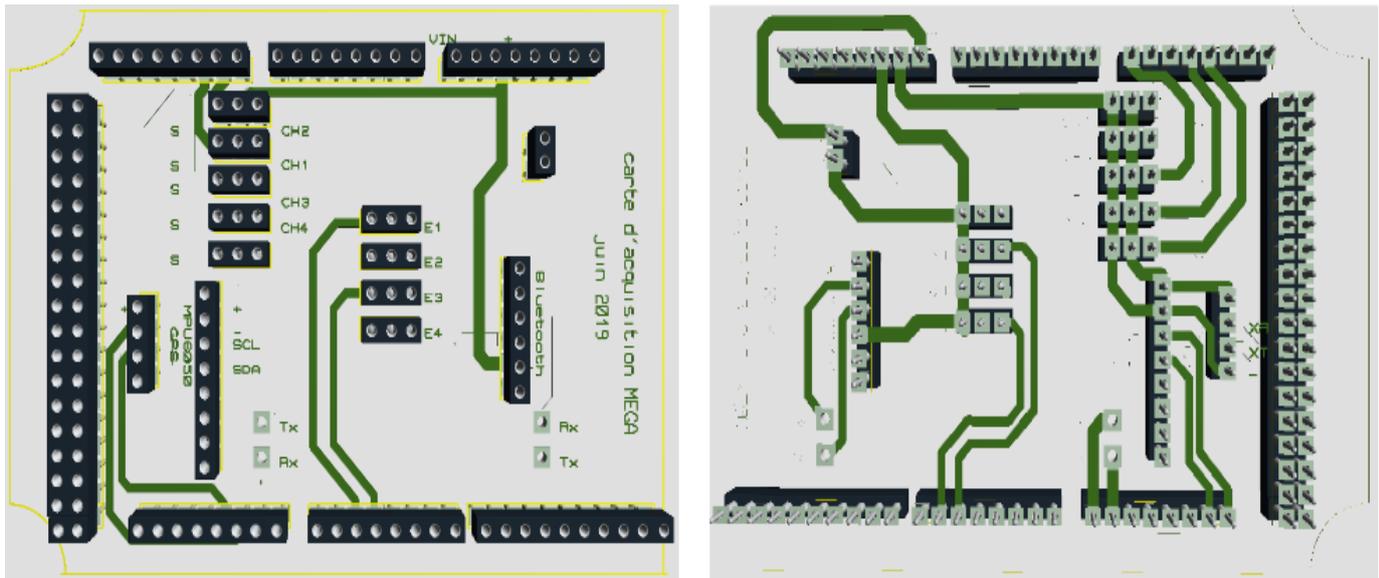


Figure 5.13 : Vue 3D de la carte d'acquisition.

### 5.3.3 Imprimé sur du papier calques

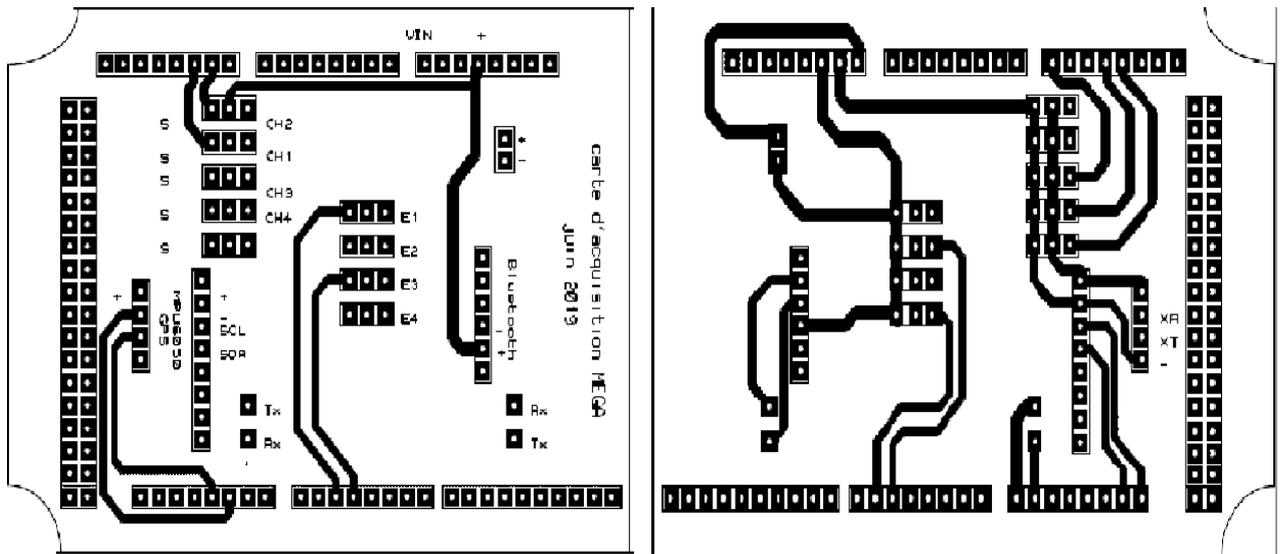


Figure 5.14 : Imprimé du circuit ARES sur du papier calques.

### 5.3.4 Circuit imprimé

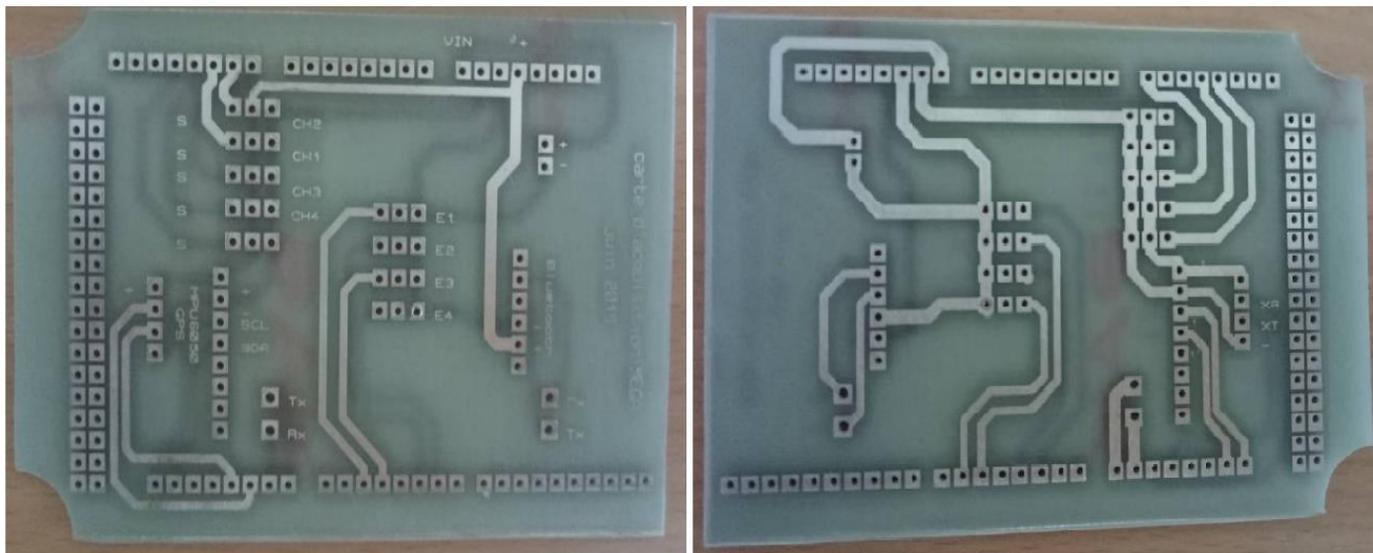


Figure 5.15 : Circuit Imprimé de la carte d'acquisition.

## 5.4 Schéma électrique de l'autopilote

Le logiciel Proteus ISIS est utilisé pour réaliser le schéma électrique. On définit tout le branchage des capteurs, les ESC, le récepteur et la batterie avec la carte Arduino MEGA.

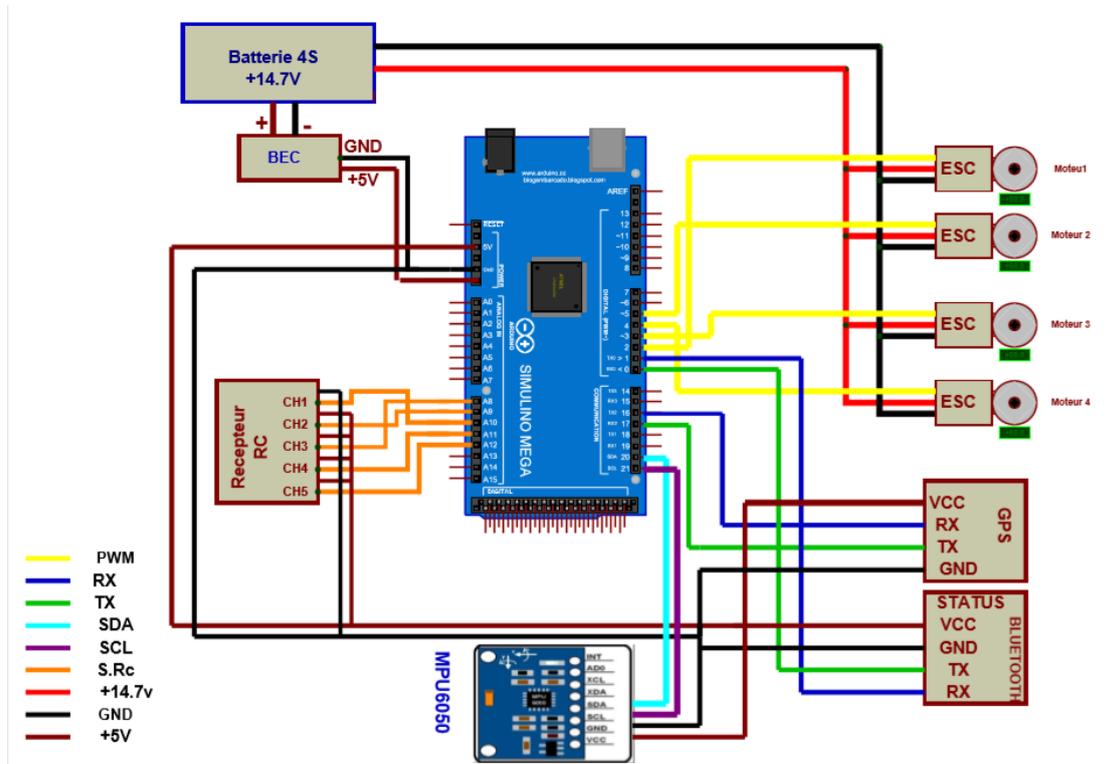


Figure 5.16 : Schéma électrique de l'autopilote sur ISIS.

### 5.4.1 Assemblage

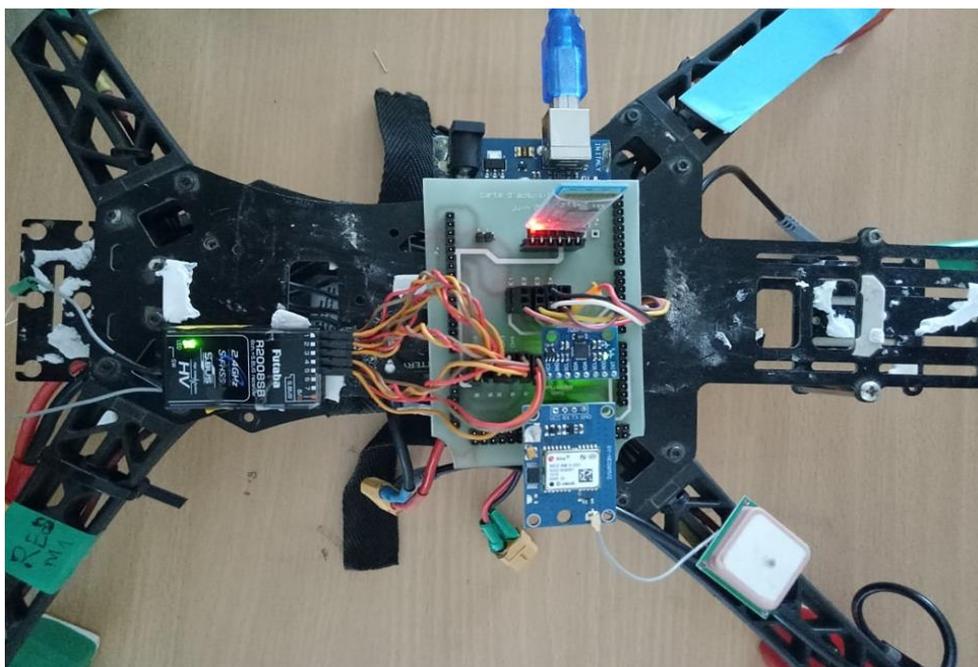


Figure 5.17 : Le résultat final de la carte autopilote.

## 5.5 L'organigramme

Pour qu'on puisse assurer les fonctionnalités de l'autopilote, le programme réalisé au niveau du microprocesseur Arduino MEGA est illustré selon l'organigramme suivant :

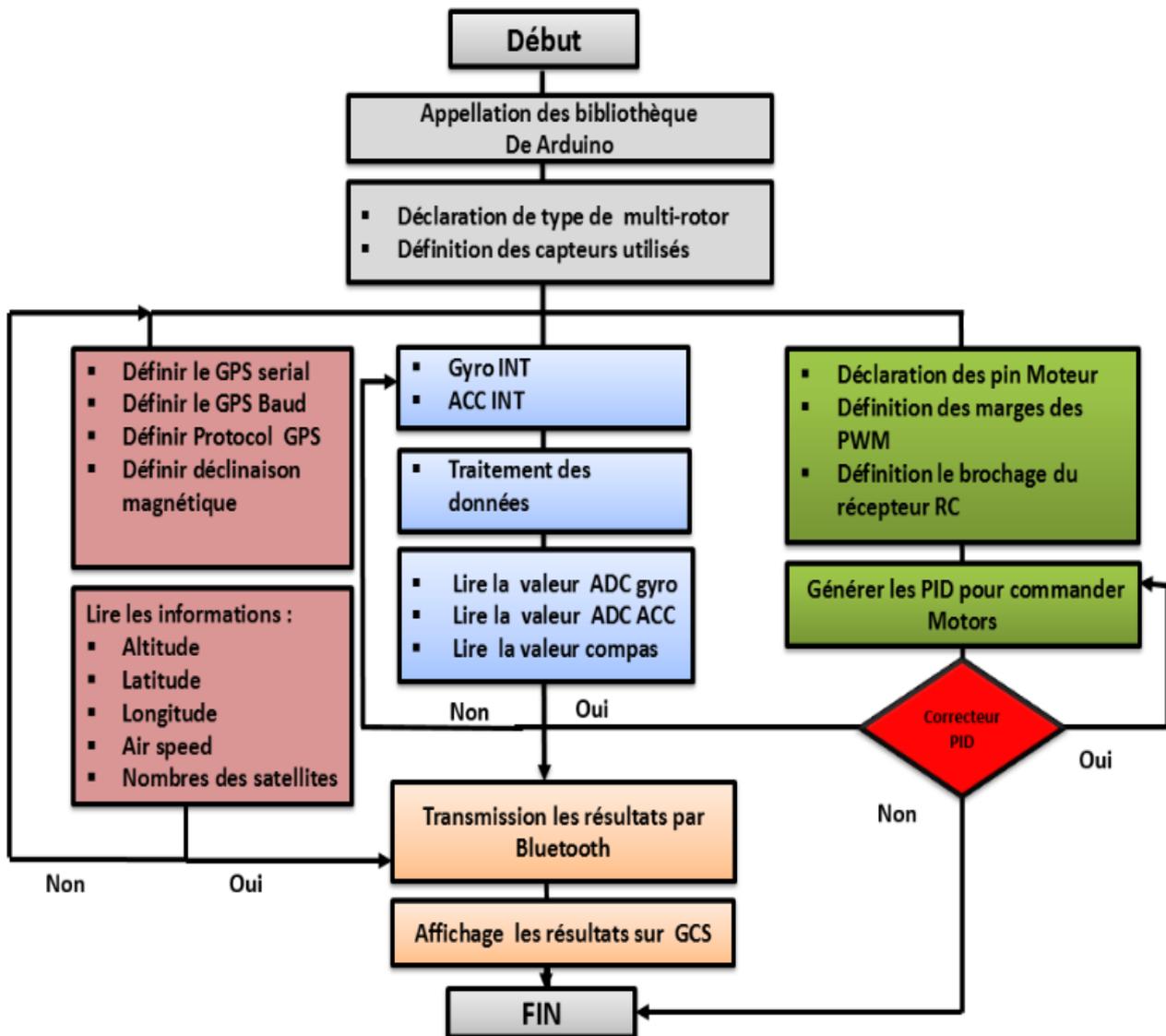


Figure 5.18 : Organigramme du programme.

## 5.6 Test Dynamique

Dans cette partie, on a procédé au test globale de la carte autopilote et cela pour visualiser les données issus des capteurs en temps réel et valider notre réalisation.

Les étapes sont structurées de la manière suivante :

- On branche la carte autopilote avec l'ordinateur.

## Chapitre 5 : Test et réalisation

- Le logiciel Arduino est lancé, on choisit les types de la carte et COM puis on flashe le programme vers la carte après avoir fait les modifications nécessaires pour son fonctionnement.
- On lance le logiciel Multiwii, on choisit COM de la carte et on clique sur START.

Après avoir respecté toutes les étapes précédentes, la visualisation montre qu'il y a un changement sur l'interface de la station au sol indiqué par la figure 5.19.

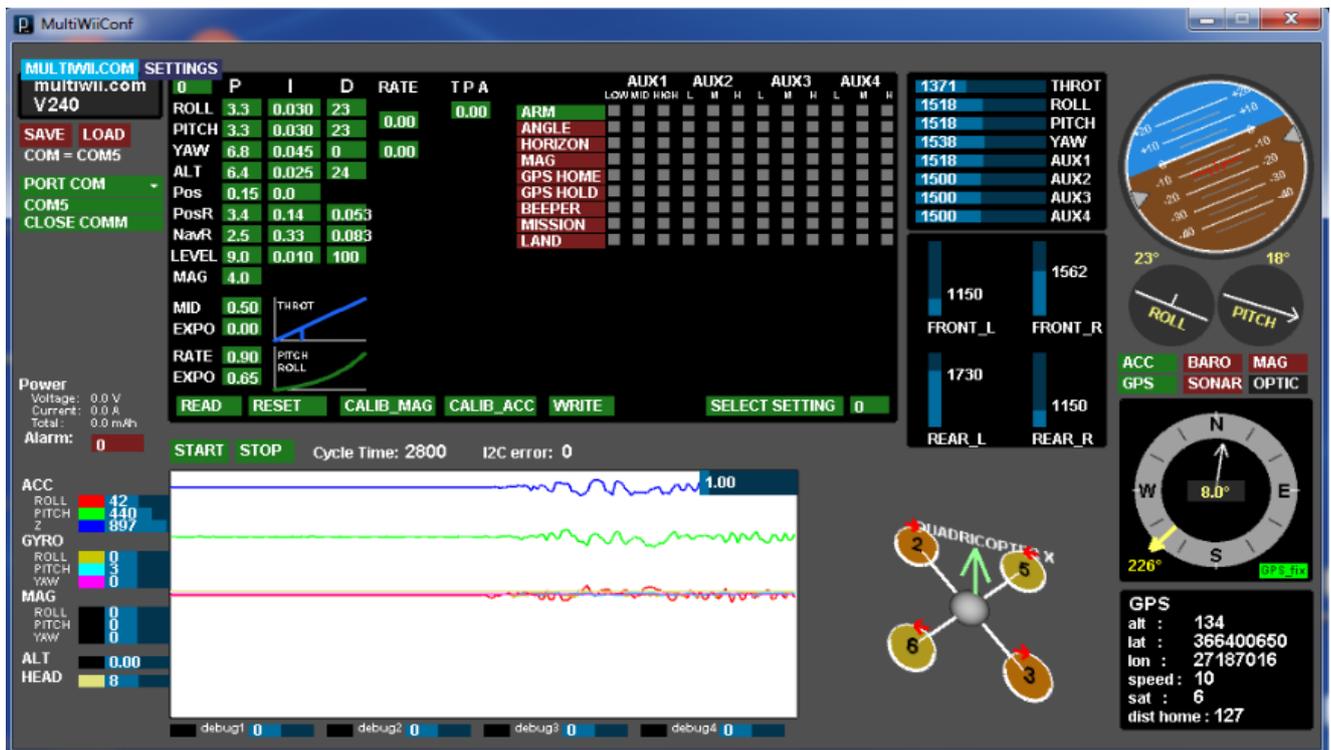


Figure 5.19 : Interface Multiwii.

### 5.7 Test de vol

Les capteurs de l'accéléromètre, gyroscope et la boussole doivent impérativement être calibrés avant de voler dans un nouvel emplacement. Cette opération vise à étalonner la stabilisation centrale du drone et pour les calibrer, il faut actionner sur la manette comme indiqué par la figure 5.20.

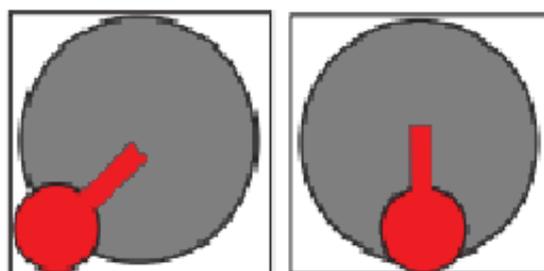
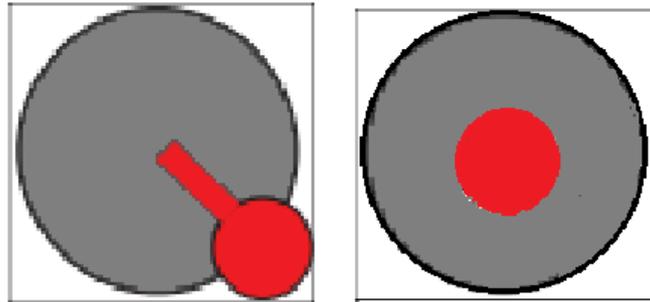


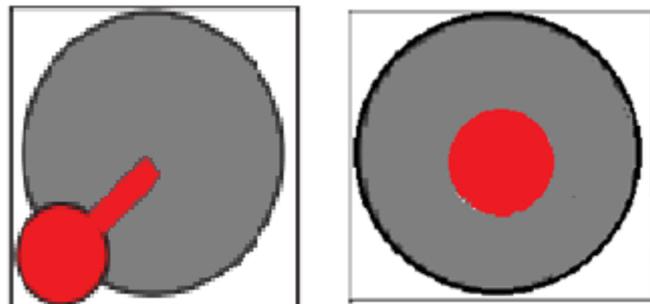
Figure 5.20 : Calibration de centrale inertielle.

Pour faire voler le drone, on doit le placé sur une surface plate et parfaitement stable. D'abord, on doit déverrouiller les moteurs en actionnant sur la manette comme le montre la figure 5.21.



**Figure 5.21** : Déverrouillé moteur avec la manette.

Une fois que les capteurs son calibrés et les moteurs sont déverrouillés, le drone est prêt à voler. Après que le vol est terminé et que le drone soit atterré sur une surface choisie, il faut verrouiller les moteurs par mesure de sécurité en actionnant sur la manette montré par la figure 5.22.



**Figure 5.22** : Verrouillage du moteur avec la manette.

## 5.8 Conclusion

Dans ce chapitre, nous avons détaillé les différents tests sur les capteurs en mettant en place la structure des étapes à suivre pour la réalisation de la carte autopilote. Ensuite, une présentation est faite sur la manière de commande des quatre moteurs et la façon de calibrage des capteurs (accéléromètre, gyroscopes et boussole). A la fin, on a fait le test de vol du quadrirotor avec la carte autopilote réalisée.

## Conclusion générale

Le quadrirotor est un engin volant à quatre hélices appartenant au domaine de la robotique aérienne. Le projet de pilotage d'un quadrirotor se compose de deux phases essentielles : la phase de modélisation dynamique qui concerne le coté physique de notre étude et la phase de réalisation d'une carte autopilote.

A cet effet nous avons effectué une vision générale sur les différentes catégories des drones, les différents autopilotes des mini UAV à voilure tournante, les différents mouvements qu'un quadrirotor peut faire, les lois physiques qu'un quadrirotor subit et une présentation de la loi de commande.

Cette étude nous a permis de bien comprendre les exigences de notre système ainsi le choix du microprocesseur et les capteurs qu'on doit utiliser. Ce projet est composé de trois sous-systèmes :

- Un sous-système qui mesure longitude, altitude, latitude, position, l'accélération et la vitesse angulaire.
- Un sous-système de traitement des données.
- Un sous-système de transmission et d'affichage.

Le modèle mathématique établi pour le drone nous a aidé à faire la résolution linéaire et non linéaire ce qui a permis de comprendre et de voir les réponses des différents paramètres selon des conditions imposées afin de mener ce drone à un état d'équilibre et de stabilité.

Ce travail nous a permis de définir l'architecture sur la quelle sera bâti notre futur autopilote, nous avons opté pour un microprocesseur très basse consommation disposant de toutes les ressources matérielles nécessaires et disposant de la capacité de puissance de calcul.

Les différents programmes implémentés sur notre système ont été validés par des tests au niveau du laboratoire, les résultats obtenus sont très concluants et encourageant pour la continuité du travail.

Ce projet, nous a permis d'acquérir des connaissances et un savoir-faire dans plusieurs domaines, notamment dans le domaine des autopilotes des drones, des processeurs embarqués.

Le domaine de l'autopilote est très vaste, nous suggérons comme perspectives, d'ajouter d'autres capteurs, pour élargir la base de donnée et exploiter l'information reçue pour mieux perfectionner le vol du quadrirotor.

## Bibliographie

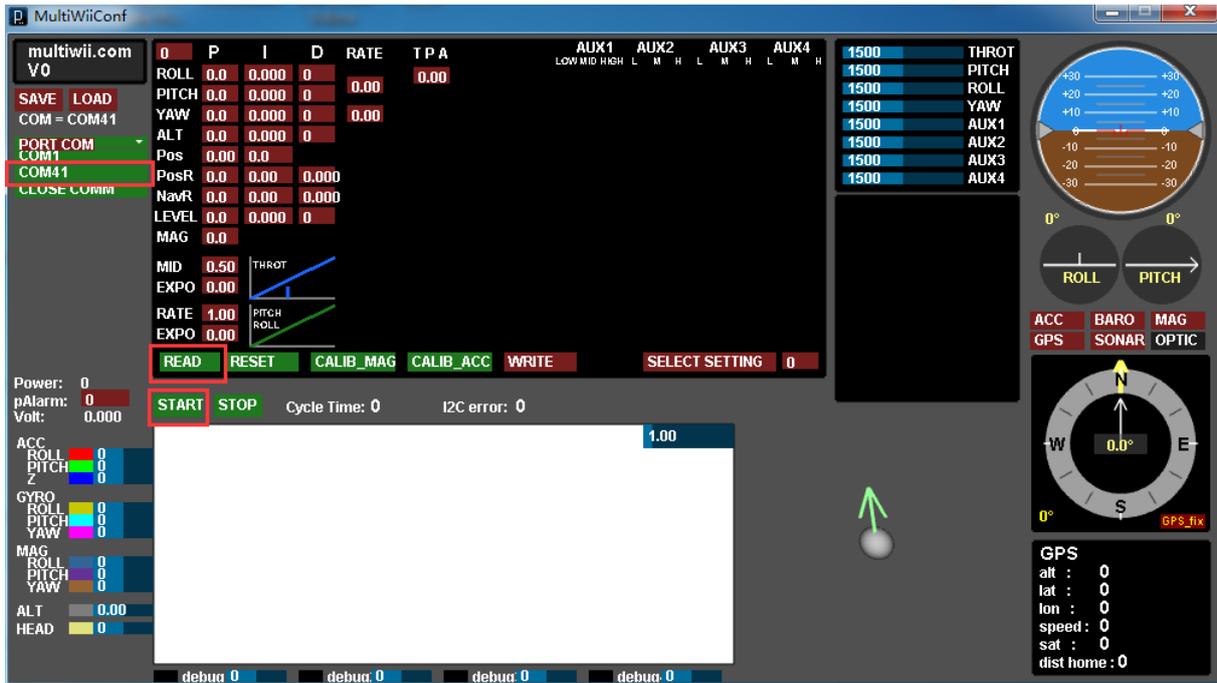
- [1] BENAÏSSA Med Rida, «Étude & implémentation d'une loi de commande pour le pilotage d'un drone quadrirotor». Université de Telemcen, 2017.
- [2] MARIA DE FATIMA BENTO, « Unmanned Aerial Vehicles: An overview ». 2010.
- [3] BOULANOUAR Khedidja. «Dynamique et control d'un drone (aile volante) ».Université Saad Dahleb Blida, 2016.
- [4] SACI Nassim, « Simulation d'un drone sous MATLAB cas d'étude quad-copter ». Université de Bejaia, 2016.
- [5] afcadillac, «Généralités». [www.afcadillac.net](http://www.afcadillac.net) . Février 2016.
- [6] T. Bresciani, «Modelling, Identification and Control of a Quadrotor Helicopter». Lund University, October 2008.
- [7] Yvind Magnussen & K. Skjnhauug «Modeling, Design and Experimental Study for a Quadcopter System Construction». Université d'Agder, 2011.
- [8] Samir Bouabdallah and Roland Siegwart «Backstepping and Sliding mode Techniques Applied to an Indoor Micro Quadrotor» IEEE international conference on robotics and automation, Barcelona, spain, april, 2005.
- [9] KEDDAR Manel et MERAZKA Sarah. «Etude dynamique, simulation et réalisation d'un quadrirotor». Université Saad Dahleb Blida, 2015.
- [10] BENNAOUI Noura. «Etude et réalisation d'un système de mesure d'altitude et de vitesse d'un mini drone». Ecole national supérieure technologie, 2017.
- [11] H. Chao, Y. Cao, and Y. Chen. «Autopilot for small-unmanned aerial vehicles: A survey». International Journal of Control, Automation and Systems, 2010.
- [12] Randal Beard, Walter Johnson, Reed Christiansen, Joshua Hintze, Timo-thy McLain. «Programmable autopilot system for autonomous flight of unmanned aerial vehicles». Publication, 2006.
- [13] Jose Ortiz. «Development of a Low Cost Autopilot System for Unmanned Aerial Vehicles». University of Virginia Commonwealth, 2010.
- [14] X. Fu, Z. Zhou, W. Xionga, and Q. Guoa «Mems-based low-cost flight control system for small uavs». Université de Tsinghua de science et Technologie, 2008.
- [15] EL KHAOUDI Samir, OUARGA Mohammed, « ESETDrone ». 2016.
- [16] Dhif Amin Rachid, « Conception d'un système hépatique pour le contrôle d'un bras manipulateur ».Université Annaba, 2018.

- [17] KHLEDJ Abdennour et BOUGAR Oussama. «Etude et réalisation d'un drone de type quadrirotor». Université Djilali Bounaama Khemis Miliana, 2017.
- [18] BOUDGUIGA Oussama «Commande à saturation pour le contrôle de la position d'un robot volant de type quadrirotor». École de technologie supérieure université du Québec, 2016.
- [19] K. Hicham, «Tolérance aux défauts via la méthode backstepping des systèmes non linéaires». Université Ferhat Abbas de Sétif, 2012.
- [20] CHABBAR Houria, « Innovation et réalisation d'une solution de géolocalisation combinée à la télécommunication ». Université Sidi Mohamed Ben Abdellah, 2014
- [21] DJAFRI Menad & CHELOUCHE Djalal, « Etude et Réalisation d'une Carte Arduino». Université de Bejaia, 2016.
- [22] Production Information, « GPS evaluation software for Windows User Guide», 2012.
- [23] Lynxmotion, « Multiwii Software Configuration », 2014.
- [24] BERRAHO Rahmouna, « Commandabilité et Observabilité des systèmes non linéaires Présenté », Université de Telemcen, 2016.
- [25] BENNIS Najib, « Représentation d'état des systèmes linéaires continus commande par placement des pôles », 2015
- [26] CUVELIER Francois, « Méthode mathématique II ». Université Paris 13, 2016.

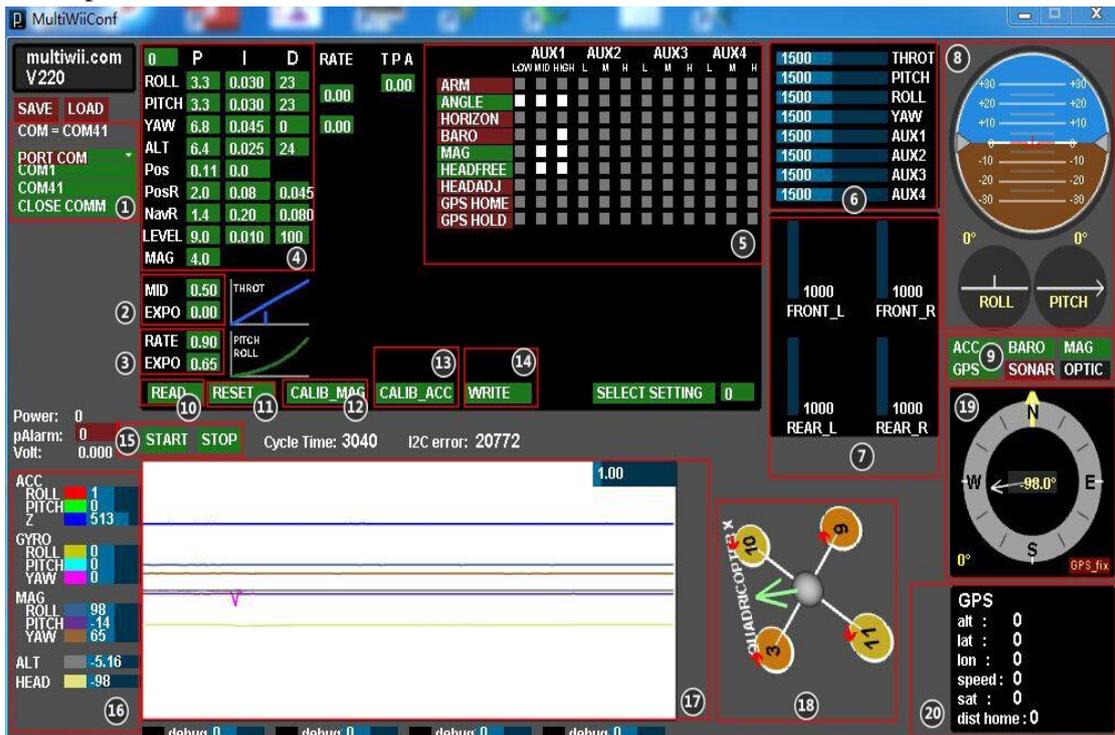
# Annexe A

## Multiwii

Sélectionnez une option sous PORT COM pour sélectionner un port série approprié (ici, il s'agit du COM41), puis les boutons START et READ vireront au vert. Cliquez sur DÉMARRER pour afficher la courbe de données. Enfin, cliquez sur READ pour charger les paramètres PID par défaut.



Si tout se passe bien, vous verrez l'interface suivante :



On va donner la signification de chaque zone de l'interface graphique, comme illustré à la figure précédente :

- 1) Sélection du port série.
- 2) Courbe d'accélérateur.
- 3) Réglage de la courbe de commande à distance.
- 4) Réglage des paramètres PID.
- 5) Réglage du mode de vol.
- 6) Canaux de sortie pour la télécommande.
- 7) Puissance moteur.
- 8) Angles d'attitude.
- 9) Etat d'activation des capteurs.
- 10) Paramètres de lecture.
- 11) Réinitialiser les paramètres.
- 12) Calibrer le magnétomètre.
- 13) Calibrer l'accéléromètre.
- 14) Ecrire les paramètres.
- 15) Commencer à sortir ou arrêter la sortie des données du capteur.
- 16) Données du capteur.
- 17) Courbe de sortie des capteurs.
- 18) Schéma 3D pour quadrirotor.
- 19) Angle de direction 20. Données de sortie GPS.

### **Sélectionner le mode de vol :**

ARM : Sélection du mode verrouillage / déverrouillage de la commande de vol

ANGLE : Mode auto-stabilisation.

HORIZON : mode de vol horizontal.

BARO : Maintien de la hauteur par pression barométrique.

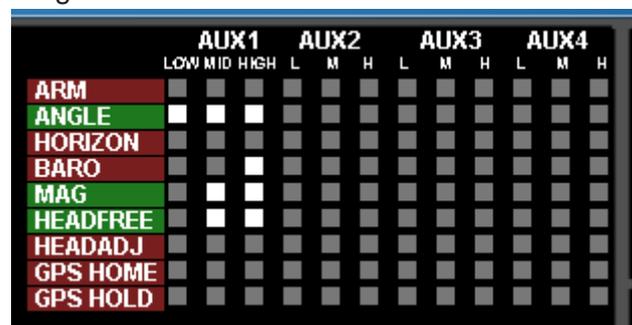
MAG : Maintien de la direction.

HEADFREE : Mode sans tête.

HEADADJ : Mode de réglage de la tête.

GPS HOME : une clé à retourner.

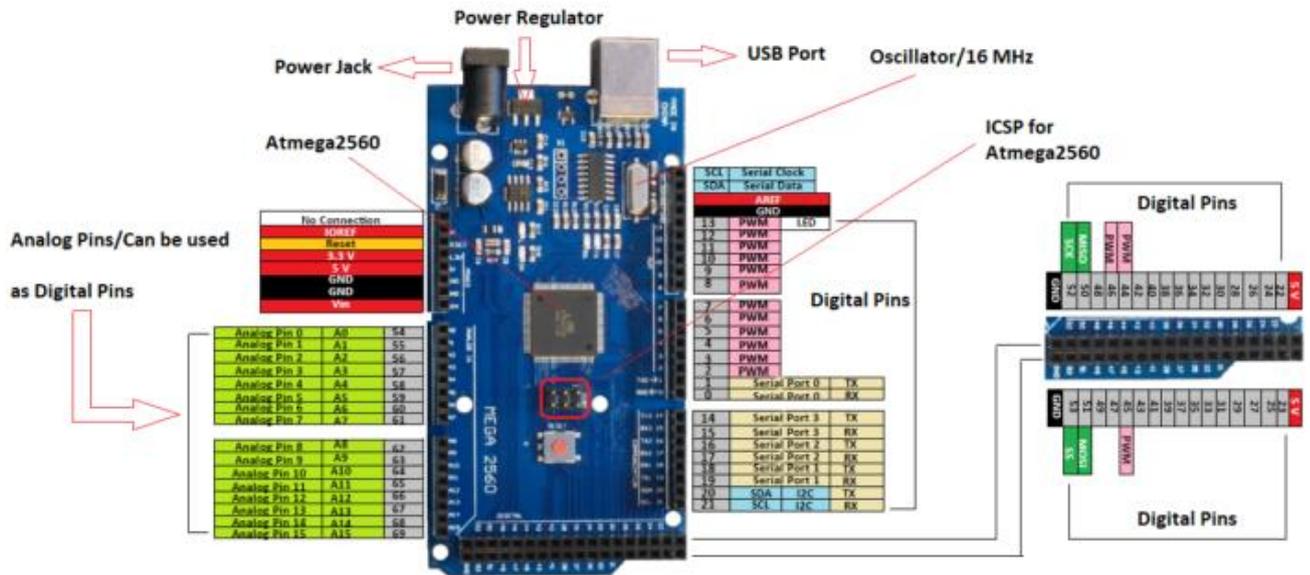
GPS HOLD : Point de fixation.



Cette zone permet d'assigner les modes de vol en fonction de la valeur d'une des quatre entrées auxiliaire. Chaque AUX est divisé en trois états, lorsque l'utilisateur modifie l'état d'un des AUX, le capteur présélectionné est activé. Plusieurs capteurs peuvent être activés en même temps. Le mode de vol résultant dépend des capteurs activés. Dans ce cas, l'un des potentiomètres de la télécommande a été relié à l'entrée AUX1. Cela permet de sélectionner jusqu'à trois modes de vols différents et d'effectuer des changements de mode lorsque le quadrirotor est en l'air.

## Annexe B

Description de la carte Arduino MEGA :



**Arduino Mega 2560 Pinout**

- **5V et 3,3V** : Cette broche est utilisée pour fournir une tension de sortie régulée autour de 5V. Cette alimentation régulée alimente le contrôleur et les autres composants de la carte.
- **GND** : Le tableau comporte 5 broches de masse, ce qui le rend utile lorsque plusieurs broches de terre sont nécessaires pour le projet.
- **Reset** : Cette broche est utilisée pour réinitialiser la carte.
- **Vin** : C'est la tension d'entrée fournie à la carte qui varie de 7V à 20V. La tension fournie par la prise d'alimentation est accessible via cette broche. Cependant, la tension de sortie via cette broche vers la carte sera automatiquement réglée sur 5V.
- **Communication série RXD & TXD** : sont les broches série utilisées pour transmettre et recevoir des données série, c'est-à-dire que Rx représente la transmission de données alors que Tx est utilisé pour recevoir des données. Il existe quatre combinaisons de ces broches série utilisées lorsque Serial 0 contient RX (0) et TX (1), Serial 1 contient TX (18) et RX (19), Serial 2 contient TX (16) et RX (17). et la série 3 contient TX (14) et RX (15).
- **Interruptions externes** : Six broches sont utilisées pour créer des interruptions externes, à savoir les interruptions 0 (0), 1 (3), 2 (21), 3 (20), 4 (19) et 5 (18). Ces broches produisent des interruptions de différentes manières, à savoir fournir une valeur basse, un front montant ou descendant ou une valeur variable aux broches d'interruption.

- **LED** : Cette carte est livrée avec un LED intégrée connectée à la broche numérique 13. Une valeur HIGH à cette broche allume la LED et une valeur LOW l'éteint. Cela vous donne le changement d'allaitement de vos compétences en programmation en temps réel.
- **Broches analogiques** : Il y a 16 broches analogiques intégrées sur la carte étiquetées d'A0 à A15. Il est important de noter que toutes ces broches analogiques peuvent être utilisées en tant que broches d'E / S numériques. Chaque broche analogique est fournie avec une résolution de 10 bits. Ces broches peuvent mesurer de la terre à 5V.
- **I2C** : Deux broches 20 et 21 prennent en charge la communication I2C, 20 représentant SDA (ligne de données série principalement utilisée pour conserver les données) et 21 représentant SCL (ligne d'horloge série principalement utilisée pour la synchronisation des données entre les périphériques).
- **Communication SPI** : SPI signifie interface périphérique série utilisée pour la transmission de données entre le contrôleur et les autres composants périphériques. Quatre broches, à savoir 50 (MISO), 51 (MOSI), 52 (SCK), 53 (SS) sont utilisées pour la communication SPI.