

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

UNIVERSITE MOULOUD MAMMARI DE TIZI-OUZOU



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

Mémoire de Fin d'Etudes de MASTER ACADEMIQUE

Domaine : Sciences et Technologies

Filière : Automatique

Spécialité : Automatique et Informatique
Industrielle

Présenté par

Said HABERA
Hamza DJOUZI

Thème

Modélisation, conception et réalisation d'un drone quadricoptère

Mémoire soutenu publiquement le 29/09/ 2024 devant le jury composé de :

M. Kamal HAMMOUCHE
Professeur, UMMTO, Président

Mme. Fatima OUENDI
MCB, UMMTO, Encadrant

M. Mohand Achour TOUAT
MCA, UMMTO, Examineur

M. Hocine KHATI
MAB, UMMTO, Examineur

Remerciements

Nous tenons à exprimer notre profonde gratitude à toutes les personnes qui ont contribué, de près ou de loin, à la réalisation de ce projet.

Tout d'abord, nous adressons nos sincères remerciements à notre promotrice, Madame F. OUENDI, pour son encadrement, ses conseils précieux et son soutien constant tout au long de ce travail.

Nous désirons également adresser nos plus sincères remerciements à Monsieur M. A. TOUAT, chef du département Automatique, pour son soutien précieux et son aide déterminante dans l'accomplissement de notre travail.

Nous tenons à exprimer notre profonde gratitude à Monsieur M. DAID pour son soutien inestimable et ses conseils éclairés, qui ont grandement contribué à la réalisation de ce projet.

Nous exprimons également toute notre reconnaissance à Monsieur A. MAIDI, notre professeur, pour son accompagnement dans la rédaction de ce mémoire et ses suggestions éclairées qui ont grandement enrichi notre travail.

Enfin, nous tenons à remercier nos chères familles et amis pour leur soutien inconditionnel, leur patience et leurs encouragements tout au long de cette aventure. Sans leur présence à nos côtés, cette réalisation n'aurait pas été possible.

Table des matières

Introduction générale	1
1 Modélisation du quadrirotor	3
1.1 Généralités sur les drones	3
1.1.1 Introduction	3
1.1.2 Histoire et état de l'art des drones	3
1.1.3 Classifications et applications des drones	4
1.2 Etude du quadri-rotor	6
1.2.1 Introduction	6
1.2.2 Description générale du quadrirotor	7
1.2.3 Mouvements du quadrirotor	8
1.2.4 Modélisation dynamique du quadrirotor	12
1.2.5 Effets physiques agissants sur le quadrirotor	15
1.2.6 Développement du Modèle mathématique selon Newton-Euler	17
1.2.7 Equations de mouvement de translation	19
1.2.8 Equations de mouvement de rotation	19
1.2.9 Représentation d'état du système	20
1.2.10 Etude de la dynamique des moteurs	21
1.3 Commande PID	22
1.3.1 Introduction	22
1.3.2 Proportionnelle (P)	22
1.3.3 Intégrale (I)	23
1.3.4 Dérivée (D)	23
1.3.5 Combinaison PID	23
1.3.6 Avantages et inconvénients	24
1.4 Simulation	26
1.5 Conclusion	28

2	Constitution du drone	29
2.1	Introduction	29
2.2	Châssis	29
2.3	Systeme de propulsion	30
2.3.1	Moteurs	30
2.3.2	Hélices	33
2.3.3	Contrôleurs électroniques de vitesse (ESC)	34
2.3.4	Batterie LIPO	34
2.3.5	Distributeur de puissance PDB-XT60	36
2.4	Contrôleur du vol	37
2.4.1	Microcontrôleur	37
2.4.2	Capteurs	38
2.5	Commande à distance	40
2.5.1	Radiocommande	40
2.5.2	Commande Bluetooth	42
2.6	Conclusion	44
3	Mise en œuvre et réalisation du quadricoptère	45
3.1	Introduction	45
3.2	Première méthode de programmation	45
3.3	Deuxième méthode de programmation (avec MULTIWII)	47
3.3.1	Application Android de pilotage de quadroptère réalisée avec App Inventor	48
3.3.2	Conception de la radio commande pour le quadricoptère	51
3.3.3	Définition de la plateforme MULTIWII	53
3.3.4	Procédure générale de configuration pour un quadrirotor utilisant Arduino Nano et MultiWii	54
3.3.5	Montage	54
3.3.6	Calibrage des ESC	56
3.3.7	Calibrage des capteurs	56
3.3.8	Ajustement des paramètres PID	57
3.3.9	Évaluation de la réponse du quadricoptère aux perturbations	60
3.3.10	Illustration finale du drone	61
3.3.11	Schéma récapitulatif des différentes parties du drone	63
3.3.12	Conclusion	63
	Conclusion générale	64

Table des figures

1.1	Types de drones multi-rotors [13].	5
1.2	Infographie sur l'utilisation des quadricoptères [32].	6
1.3	Description générale du quadrirotor.	8
1.4	Mouvement Vertical.	9
1.5	Mouvement de Roulis.	10
1.6	Mouvement de Tangage.	10
1.7	Mouvement de Lacet.	11
1.8	Translations horizontales.	12
1.9	Géométrie du quadrirotor.	13
1.10	Principe du contrôleur PID.	24
1.11	Suivi des trajectoires $X(t)$, $Y(t)$ et $Z(t)$ avec les consignes sinus, cosinus et constante.	26
1.12	Simulation d'une manœuvre en spirale en 3D et ses projections 2D. . .	27
1.13	Comparaison des mesures $(x, y, z, \phi, \theta, \psi)$ avec les consignes pour la ma- nœuvre en spirale.	27
1.14	Analyse des erreurs en position (x, y, z) et en orientation (ψ) pour la manœuvre en spirale.	28
2.1	DJI F450 [2].	30
2.2	Moteurs brushless 1000KV et 2200KV [31].	31
2.3	Mesure de la masse poussée.	32
2.4	Helice 1045R.	33
2.5	ESC 30A [2].	34
2.6	Batterie Lipo 3s 4200mah.	35
2.7	Chargeur Lipo 2S/3S.	36
2.8	PDB-TX60 [25].	37
2.9	Arduino NANO [15].	38
2.10	MPU6050 [28].	39
2.11	Capteur barométrique GY 63 [3].	39
2.12	HMC5883L [22].	40

2.13	FlySky FS i6X [29].	41
2.14	Récepteur FlySky FS-IA6B [16].	42
2.15	Module Bluetooth HC-05 [33].	42
2.16	Application Android App Inventor.	43
2.17	Application EZ-GUI.	44
3.1	Bicopter.	46
3.2	PID deux moteurs.	47
3.3	Design de l'application avec App Inventor.	49
3.4	Code de l'application avec App Inventor.	50
3.5	Conception de la Radio Commande.	52
3.6	Interface graphique (MultiWii GUI).	53
3.7	Schéma de montage.	56
3.8	Impact des paramètres PID optimisés sur le comportement du drone en vol.	58
3.9	Effet de l'application du filtre sur le drone lors de l'armement.	59
3.10	Effet de l'application du filtre sur le drone en vol.	60
3.11	Évaluation de la réponse du quadricoptère aux Perturbations	61
3.12	Stabilité atteinte par le drone lors de ses vols.	62
3.13	Schéma global.	63

Liste des tableaux

2.1	Comparaison entre les moteurs brushless 2200kv et 1000kv.	32
3.1	Paramètres PID optimaux.	58

Liste des symboles

R	Matrice de rotation décrivant l'orientation de l'objet mobile
ζ	Vecteur de position
Φ	Angle de roulis (rotation autour de l'axe x)
Θ	Angle de tangage (rotation autour de l'axe y)
Ψ	Angle de lacet (rotation autour de l'axe z)
Ω	Vitesse angulaire dans le repère fixe
$\Omega_{1,3}$	Composantes de la vitesse angulaire dans le repère fixe
$\dot{\Phi}, \dot{\Theta}, \dot{\Psi}$	Vitesses de rotation dans le repère mobile
v_m	Vitesse linéaire dans le repère mobile
P	Poids du quadrirotor
m	Masse du quadrirotor
g	Accélération de la pesanteur
F_i	Force de poussée générée par le moteur i
T_h	Force de traînée des hélices
F_t	Force de traînée selon les axes
M_x	Moment autour de l'axe x
M_y	Moment autour de l'axe y
M_z	Moment autour de l'axe z
M_{gh}	Moment gyroscopique des hélices
M_{gm}	Moment gyroscopique dû aux mouvements du quadrirotor
J	Matrice d'inertie symétrique
F_f	Force totale générée par les rotors
F_g	Force de gravité
K_{ft}	Coefficients de traînée de translation
K_{fa}	Coefficients des frottements aérodynamiques
$S(\Omega)$	Matrice antisymétrique associée à Ω
P	Composante proportionnelle du PID
I	Composante intégrale du PID
D	Composante dérivée du PID

$e(t)$	Erreur à l'instant t
K_p	Gain proportionnel
K_i	Gain intégral
K_d	Gain dérivé

Liste des abréviations

ESC	Electronic Speed Controller (Contrôleur de Vitesse Électronique)
6 DDL	Six Degrés De Liberté
VTOL	Vertical Take-Off and Landing (décollage vertical et atterrissage vertical)
UAV	Unmanned Aerial Vehicle (Véhicule Aérien Sans Pilote)
LiPo	Lithium-Polymer (Lithium-Polymère)
RAM	Random Access Memory (Mémoire Vive)
ROM	Read-Only Memory (Mémoire Non-Volatile)
MEMS	Micro-Electro-Mechanical Systems (Systèmes Micro-Électro-Mécaniques)
LCD	Liquid Crystal Display (Écran à Cristaux Liquides)
UART	Universal Asynchronous Receiver-Transmitter
URL	Asynchronous Receiver-Transmitter (Récepteur-Émetteur Asynchrone Universel)
PID	Proportional Integral Derivative (Contrôleur Proportionnel Intégral Dérivé)
DJI	Da-Jiang Innovations (Entreprise de fabrication de drones)
PDB	Power Distribution Board (Distributeur de Puissance)
MPU	Motion Processing Unit (Unité de Traitement de Mouvement)
BMP	Barometric Pressure Sensor (Capteur de Pression Barométrique)
HMC	High Magnetic Compass (Magnétomètre Haute Précision)
GND	Ground (Masse ou Terre)
VCC	Voltage Common Collector (Tension d’Alimentation)
HC	High Current (Module Bluetooth HC-05)
EZ-GUI	Easy Graphical User Interface (Interface Graphique Utilisateur Facile)
GPS	Global Positioning System (Système de Positionnement Global)
SDA	Serial Data Line (Ligne de Données Série)
SCL	Serial Clock Line (Ligne d’Horloge Série)
IDE	Integrated Development Environment (Environnement de Développement Intégré)
IP	Internet Protocol (Protocole Internet)
MAV	Micro Aerial Vehicle (Véhicule Aérien Microscopique)
MALE	Medium Altitude Long Endurance (Altitude Moyenne, Longue Endurance)
HALE	High Altitude Long Endurance (Haute Altitude, Longue Endurance)

LALE	Low Altitude Long Endurance (Basse Altitude, Longue Endurance)
ADC	Analog-to-Digital Converter (Convertisseur Analogique-Numérique)
SRAM	Static Random-Access Memory (Mémoire Vive Statique)
PWM	Pulse Width Modulation (Modulation de Largeur d'Impulsion)
AFHDS	Automatic Frequency Hopping Digital System
MIT	Massachusetts Institute of Technology (Institut de Technologie du Massachusetts)
ICSP	In-Circuit Serial Programming (Programmation Série en Circuit)
WiFi	Wireless Fidelity (Fidélité Sans Fil)

Résumé

Ce travail présente la conception, la modélisation, et la réalisation d'un quadrirotor. Dans un premier temps, une étude approfondie des drones est menée, incluant leur historique, leurs classifications, et leurs diverses applications dans les domaines civils, militaires et scientifiques. Une attention particulière est accordée au quadrirotor, avec une analyse détaillée de sa dynamique, de ses mouvements et des effets physiques auxquels il est soumis.

Ensuite, le projet se focalise sur la modélisation mathématique du quadrirotor en utilisant les équations de Newton-Euler, permettant de définir les équations de mouvement en translation et en rotation. La commande PID, essentielle pour la stabilité du quadrirotor, est également étudiée, en détaillant ses avantages et inconvénients.

Enfin, la partie pratique du projet présente la réalisation complète du drone, en passant par le choix des composants, l'intégration du système de commande MultiWii, la configuration du microcontrôleur, et le développement d'une application Android pour piloter le drone via Bluetooth. Des tests de stabilité et des ajustements des paramètres PID ont permis d'optimiser la réponse du drone face aux perturbations externes.

Ce travail met en évidence les défis techniques rencontrés ainsi que les solutions mises en place pour aboutir à un drone fonctionnel et performant.

Abstract

This work provides a comprehensive overview of the design, modeling, and development of a quadrirotor drone. It begins with an in-depth examination of drones, covering their history, classifications, and various applications across civil, military, and scientific domains. Special focus is given to the quadrirotor, including a detailed analysis of its dynamics, movements, and the physical forces it experiences.

The project then shifts to the mathematical modeling of the quadrirotor using Newton-Euler equations, which define the equations of motion for both translation and rotation. PID control, which is crucial for maintaining the stability of the quadrirotor, is also explored, highlighting its benefits and limitations.

In the practical section, the work details the complete implementation of the drone, from selecting components and integrating the MultiWii control system to configuring the microcontroller and developing an Android application for Bluetooth control. Stability testing and PID parameter tuning were carried out to refine the drone's performance in response to external disturbances.

Overall, this work emphasizes the technical challenges faced and the solutions devised to develop a functional and effective quadrirotor drone.

الملخص

يتناول هذا العمل تصميم وتنفيذ طائرة رباعية (درون) من خلال مراحل مختلفة تشمل الدراسة، النمذجة، والتطبيق العملي. في البداية، يتم استعراض شامل للطائرات بدون طيار، مع تسليط الضوء على تاريخها، تصنيفاتها، وتطبيقاتها في مجالات مثل الزراعة، المراقبة، والتصوير الجوي. كما يتم التركيز على الطائرة الرباعية من خلال تحليل ديناميكيتها، حركتها، والقوى المؤثرة عليها.

بعد ذلك، يتناول المشروع النمذجة الرياضية للطائرة باستخدام معادلات نيوتن-أويلر، والتي تتيح لنا صياغة معادلات الحركة الخاصة بالترجمة والدوران. يتم أيضاً دراسة التحكم PID باعتباره أساسياً لتحقيق استقرار الطائرة، مع مناقشة مزاياه وعيوبه.

أما في الجزء العملي، فيتم عرض كيفية تنفيذ الطائرة من الناحية الميكانيكية والإلكترونية، بدءاً من اختيار المكونات وحتى دمج نظام التحكم MultiWii. كما تم تطوير تطبيق أندرويد للتحكم في الطائرة عبر البلوتوث. وتم إجراء اختبارات لاستقرار الطائرة مع ضبط معايير PID لتكييف استجابتها مع الاضطرابات الخارجية.

يستعرض هذا العمل التحديات التقنية التي واجهت المشروع والحلول التي تم تطبيقها لتطوير طائرة رباعية تعمل بكفاءة وفعالية.

Introduction générale

L'évolution rapide des technologies aéronautiques, en particulier dans le domaine des drones, a profondément transformé les industries civiles et militaires, offrant une multitude de nouvelles opportunités et applications. Parmi les différentes configurations de drones, les quadricoptères se distinguent par leur maniabilité, leur simplicité de conception et leur polyvalence, ce qui les rend indispensables dans des domaines variés tels que la photographie aérienne, la surveillance, l'agriculture de précision, et bien d'autres encore.

Les drones, ou aéronefs sans pilote (UAV), ont évolué de simples dispositifs militaires à des outils polyvalents utilisés dans divers domaines comme la surveillance, la photographie aérienne, et la livraison de colis. Leur technologie avancée, combinant capteurs, systèmes de communication et algorithmes de contrôle, leur permet de s'adapter à une large gamme d'applications.

Pour le contrôle des quadricoptères, la commande PID (Proportional-Integral-Derivative) est la plus courante grâce à sa simplicité et son efficacité. D'autres méthodes incluent la commande LQR (Linear Quadratic Regulator) pour optimiser la performance, la commande MPC (Model Predictive Control) pour gérer des contraintes complexes, et la commande H_∞ (H-infinity) pour traiter les incertitudes. La commande adaptative ajuste les paramètres en fonction des variations du système, la commande de glissement est utile pour des environnements non linéaires, et le feedback control améliore la stabilité en ajustant les commandes basées sur les erreurs observées. Malgré ces options, la commande PID reste souvent préférée pour sa flexibilité et sa facilité d'implémentation [5].

Dans ce contexte, notre projet s'inscrit dans l'optique de développer un quadricoptère capable d'effectuer des vols stables et précis, tout en explorant les différents défis liés à sa conception et à sa modélisation. Ce mémoire présente l'ensemble du processus de conception, depuis la modélisation théorique jusqu'à la mise en œuvre pratique, en

passant par les différentes stratégies de contrôle adoptées pour assurer la stabilité et la performance de l'appareil.

Le choix de la plate-forme MultiWii, le développement d'un contrôleur PID optimisé, ainsi que l'intégration de divers capteurs et modules pour la stabilisation et le contrôle en temps réel, sont autant d'aspects techniques détaillés dans ce travail. Par ailleurs, nous avons également exploré des solutions innovantes pour améliorer l'expérience de pilotage, notamment par le biais d'une application mobile développée spécifiquement pour contrôler le drone via Bluetooth.

Ce mémoire se compose de plusieurs chapitres, chacun abordant une phase clé du projet. Le premier chapitre traite de la modélisation du quadrirotor, en présentant les généralités sur les drones, leur classification et les forces physiques qui agissent sur l'appareil, tout en développant un modèle mathématique basé sur les équations de Newton-Euler. Le deuxième chapitre est consacré à la constitution du drone, décrivant en détail les composants matériels tels que le châssis, les moteurs, les contrôleurs électroniques de vitesse (ESC), la batterie, ainsi que le rôle des capteurs et du contrôleur de vol. Le troisième chapitre se concentre sur la mise en œuvre pratique du projet, incluant le développement des logiciels de contrôle avec MultiWii, la création d'une application mobile pour piloter le drone via Bluetooth, ainsi que les différents tests et ajustements effectués pour assurer la stabilité et la performance du quadrirotor.

Ce travail est non seulement le fruit de connaissances théoriques acquises au cours de notre formation, mais aussi d'une démarche pratique visant à relever les défis inhérents à la construction et au pilotage d'un quadricoptère

Chapitre 1

Modélisation du quadrirotor

1.1 Généralités sur les drones

1.1.1 Introduction

Les UAV (Unmanned Aerial Vehicles), également appelés drones, sont des aéronefs sans pilote humain à bord. Ils peuvent être contrôlés à distance par un opérateur ou fonctionner de manière autonome grâce à des systèmes de contrôle embarqués. Utilisés dans divers contextes civils et militaires, les UAV répondent aux besoins variés des forces armées ou de sécurité. Leur masse peut varier de quelques grammes à plusieurs tonnes, selon les capacités requises. Les drones sont capables d'accomplir diverses missions, telles que la surveillance, la reconnaissance, la collecte d'informations, la cartographie, la photographie aérienne, la livraison de colis et même des opérations de combat. Ils sont équipés de caméras, de capteurs, d'équipements de communication et d'autres dispositifs nécessaires à l'exécution de leurs tâches [23].

1.1.2 Histoire et état de l'art des drones

L'histoire des drones débute à la fin du 19^e siècle avec les premières expériences de pilotage à distance. En 1898, Nikola Tesla présente son invention, le téléautomaton, au Madison Square Garden de New York. Ce bateau radiocommandé, pour lequel un brevet a été déposé, devient disponible quelques mois après le début de la guerre entre les États-Unis et l'Espagne (1898-1901). Tesla propose son invention au gouvernement américain en 1898 et en 1900 [21]. Par la suite, au début du 20^e siècle, d'autres inventeurs continuent les expérimentations dans le domaine du pilotage à distance. Durant les années 1910 et 1920, ces efforts se poursuivent, notamment avec le développement par les Allemands de missiles guidés télécommandés, tels que le V-1, pendant la seconde guerre mondiale, anticipant les drones modernes [19].

Pendant la guerre froide, les États-Unis intensifient leurs recherches sur les drones, en particulier pour des missions de reconnaissance et de surveillance. Des modèles tels que le "Ryan Firebee" deviennent des outils essentiels de la surveillance aérienne. Dans les années 1980, les drones sont utilisés dans des opérations de combat, notamment pendant la guerre Iran-Irak [34].

À partir des années 1990, les drones commencent à être utilisés pour des applications civiles telles que la surveillance environnementale, l'agriculture et la cartographie, grâce à des avancées technologiques rendant ces engins plus accessibles et polyvalents [5].

Au début du 21^e siècle, les progrès technologiques dans des domaines tels que la propulsion, les matériaux, l'électronique et l'informatique contribuent à l'émergence de nouvelles configurations de drones, dont les drones hélicoptères et à voilure tournante [23]. Ces drones trouvent des applications variées, tant civiles que militaires, allant de la surveillance à la livraison de fournitures médicales dans des zones reculées [4].

Ces dernières années, la popularité des drones a explosé avec l'introduction de modèles grand public dotés de fonctionnalités avancées. Parallèlement, les entreprises explorent de nouvelles applications commerciales telles que la livraison de colis et l'inspection des infrastructures. Les développements récents incluent l'intégration de l'intelligence artificielle pour des fonctions autonomes et l'expérimentation de la livraison de colis par des drones autonomes, ouvrant la voie à de nouvelles innovations dans ce domaine en constante évolution.

1.1.3 Classifications et applications des drones

Les drones, ou véhicules aériens sans pilote (UAV), sont devenus des outils indispensables dans de nombreux secteurs industriels en raison de leur polyvalence et de leur capacité à effectuer des tâches complexes sans intervention humaine directe. La classification des drones repose sur plusieurs critères, notamment la taille, le poids, la configuration de la voilure, et le nombre de rotors. Par exemple, les drones varient des micro ou mini-drones (MAVs) pesant moins de 5 kg aux super-lourds dépassant les 2000 kg. En fonction de l'altitude et de l'endurance, ils se divisent en catégories telles que HALE (High Altitude Long Endurance), MALE (Medium Altitude Long Endurance), et LALE (Low Altitude Long Endurance) [13].

En ce qui concerne la configuration de la voilure, on distingue principalement deux types : les drones à voilure fixe et les drones à voilure tournante. Les drones à voilure

fixe sont optimisés pour les vols longs et continus, offrant une grande autonomie et une consommation énergétique efficace, ce qui les rend idéaux pour des missions de surveillance de grande envergure. En revanche, les drones à voilure tournante, tels que les quadricoptères, se distinguent par leur maniabilité exceptionnelle et leur capacité de décollage et d'atterrissage verticaux (VTOL), ce qui les rend particulièrement adaptés à des applications nécessitant une précision accrue et une capacité de vol stationnaire [13].

Le nombre de rotors constitue un autre critère de classification important. Les drones peuvent être classés comme monocoptères (un rotor), bicoptères (deux rotors), tricoptères (trois rotors), quadricoptères (quatre rotors), hexacoïptères (six rotors), et octocoptères (huit rotors) (FIGURE 1.1). Chaque configuration présente des avantages spécifiques en termes de stabilité, de capacité de charge, et de simplicité de contrôle. Les quadricoptères, par exemple, offrent un bon équilibre entre une conception simple et une stabilité de vol élevée, ce qui explique leur popularité dans de nombreuses applications civiles et commerciales [13].

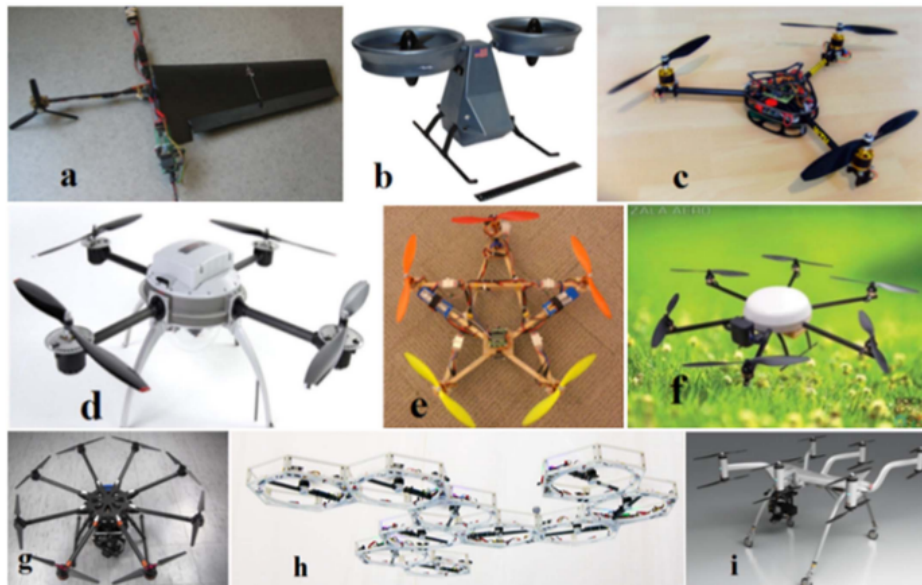


FIGURE 1.1 – Types de drones multi-rotors [13].

Les applications des drones sont multiples et couvrent une vaste gamme de domaines civils, militaires, et scientifiques (FIGURE 1.2). Dans le secteur civil, les drones sont largement utilisés pour l'agriculture de précision, la livraison de colis, la surveillance des infrastructures, et même la réalisation de films et de photographie aérienne. Dans le domaine militaire, ils servent à la reconnaissance, à la surveillance, et aux opérations de combat, offrant des capacités inégalées pour les missions à haut risque sans mettre en danger des vies humaines. Enfin, dans le domaine scientifique, les drones sont employés

pour l'exploration planétaire, la recherche environnementale, et le suivi de la faune, permettant aux chercheurs d'accéder à des zones difficiles d'accès et de recueillir des données précieuses [13].

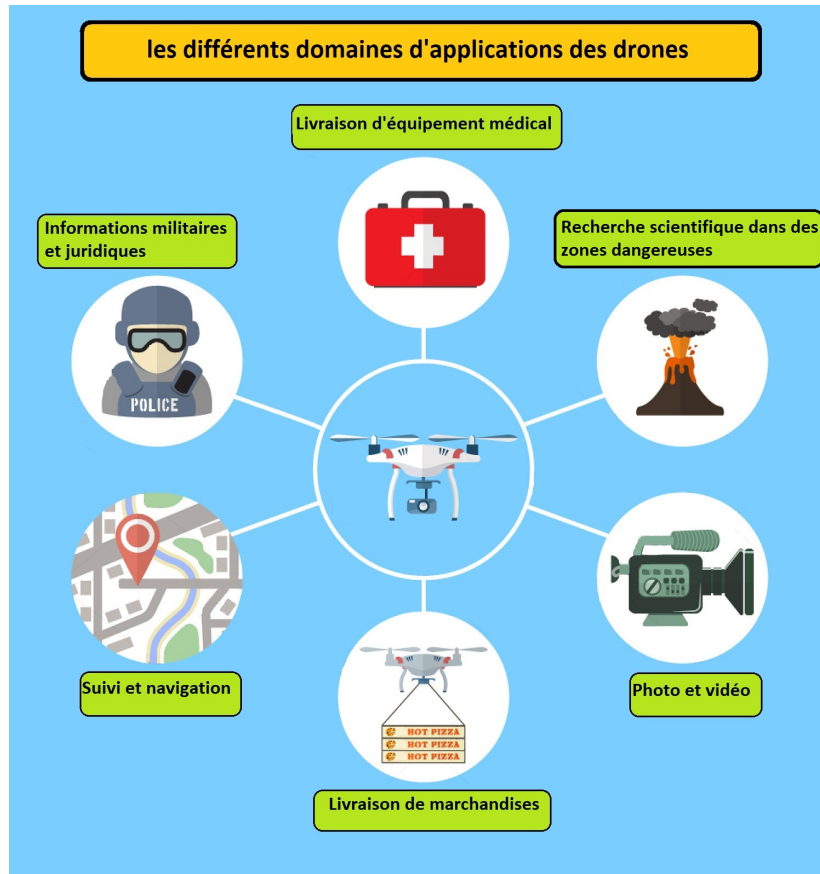


FIGURE 1.2 – Infographie sur l'utilisation des quadricoptères [32].

En résumé, les drones se distinguent par leur flexibilité et leur adaptabilité à diverses missions grâce à leurs configurations variées et à leurs capacités spécifiques. La combinaison de ces caractéristiques leur permet de répondre à des besoins variés, allant de simples tâches d'observation à des missions complexes et spécialisées [13].

1.2 Etude du quadri-rotor

1.2.1 Introduction

Pour concevoir un contrôleur de vol, il est essentiel de comprendre en profondeur les mouvements de l'avion, sa dynamique et les équations dynamiques associées. Cette compréhension est cruciale non seulement pour la conception du contrôleur, mais aussi pour garantir que les simulations du comportement du véhicule soient aussi proches

que possible de la réalité lorsque la commande est appliquée [18].

Le quadrirotor est classé parmi les systèmes volants les plus complexes en raison des nombreux effets physiques influençant sa dynamique, tels que les effets aérodynamiques, la gravité, les effets gyroscopiques, les frottements et le moment d'inertie. Cette complexité découle principalement du fait que l'expression de ces effets varie selon le mode de vol. En effet, les modèles dynamiques du quadrirotor évoluent en fonction des tâches planifiées et des environnements de navigation définis à l'avance par l'opérateur [18].

Un quadrirotor est un hélicoptère composé de quatre rotors à voilure tournante. Il s'agit d'un engin omnidirectionnel (se propageant de manière homogène dans toutes les directions) capable de décoller et d'atterrir verticalement "Vertical Take-Off and Landing" (VTOL) et de réaliser des vols stationnaires. Il est constitué de quatre pales à pas fixe accouplées à des moteurs brushless [23].

1.2.2 Description générale du quadrirotor

Un quadrirotor est un drone à quatre rotors (FIGURE 1.3), caractérisé par six degrés de liberté (6DDL). Les rotors sont placés aux extrémités d'une structure en croix, avec l'électronique de contrôle au centre. Pour stabiliser l'appareil et éviter qu'il ne tourne sur son axe de lacet, deux hélices tournent dans un sens et les deux autres dans le sens opposé. Chaque paire d'hélices tournant dans le même sens est placée à des positions opposées sur la croix [18].

Le fonctionnement d'un quadrirotor repose sur la variation de la puissance des moteurs, permettant ainsi de le faire monter, descendre, l'incliner à gauche ou à droite (roulis), en avant ou en arrière (tangage), ou encore pivoter sur lui-même (lacet). Avec ses six degrés de liberté (trois mouvements de rotation et trois de translation), le quadrirotor est un système sous-actionné, utilisant seulement quatre moteurs pour contrôler ses mouvements [18].

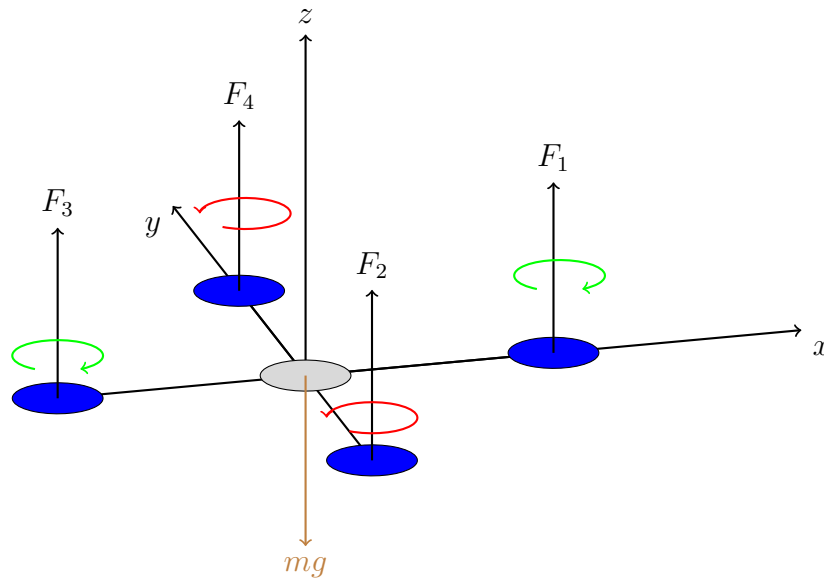


FIGURE 1.3 – Description générale du quadrirotor.

Avec : F_i étant les forces de poussée générées par les moteurs et mg représentant le poids du quadrirotor.

1.2.3 Mouvements du quadrirotor

Les mouvements de base d'un quadrirotor sont réalisés en faisant varier la vitesse de chaque rotor, modifiant ainsi la poussée produite. Le quadrirotor s'incline vers le rotor le plus lent, ce qui entraîne une translation dans cette direction. Comme pour un hélicoptère classique, les mouvements sont couplés, ce qui signifie que le quadrirotor ne peut pas se déplacer sans inclinaison. Ainsi, toute variation de la vitesse d'un rotor entraîne un mouvement dans au moins trois degrés de liberté [18].

Par exemple, augmenter la vitesse du rotor gauche entraîne un mouvement de roulis (le quadrirotor s'incline vers la droite, vers le rotor plus lent), un mouvement de lacet (l'équilibre entre les rotors tournant dans le sens des aiguilles d'une montre et ceux tournant dans le sens inverse est perturbé, provoquant une rotation horizontale), et une translation (l'inclinaison de la structure modifie l'orientation de la force de poussée). Cet accouplement explique pourquoi il est possible de contrôler les six degrés de liberté du quadrirotor avec seulement quatre moteurs [18].

Le quadrirotor peut réaliser cinq mouvements principaux :

- Mouvement de translations vertical.
- Mouvement de roulis.

- Mouvement de tangage.
- Mouvement de lacet.
- Mouvement de translations horizontales.

Mouvement vertical

Pour que le quadrirotor puisse planer, la force de portance doit être alignée le long de l'axe z et être exactement égale à la force de gravité. De plus, la force de portance générée par chaque rotor doit être identique afin d'empêcher le véhicule de se renverser. Ainsi, la poussée produite par chaque rotor doit être égale [18].

Le mouvement vertical (ascendant et descendant) (FIGURE 1.4) est contrôlé en modifiant la vitesse de rotation des moteurs, et par conséquent la poussée produite. Si la force de portance dépasse le poids du quadrirotor, celui-ci monte. Inversement, si la force de portance est inférieure au poids du quadrirotor, celui-ci descend [18].

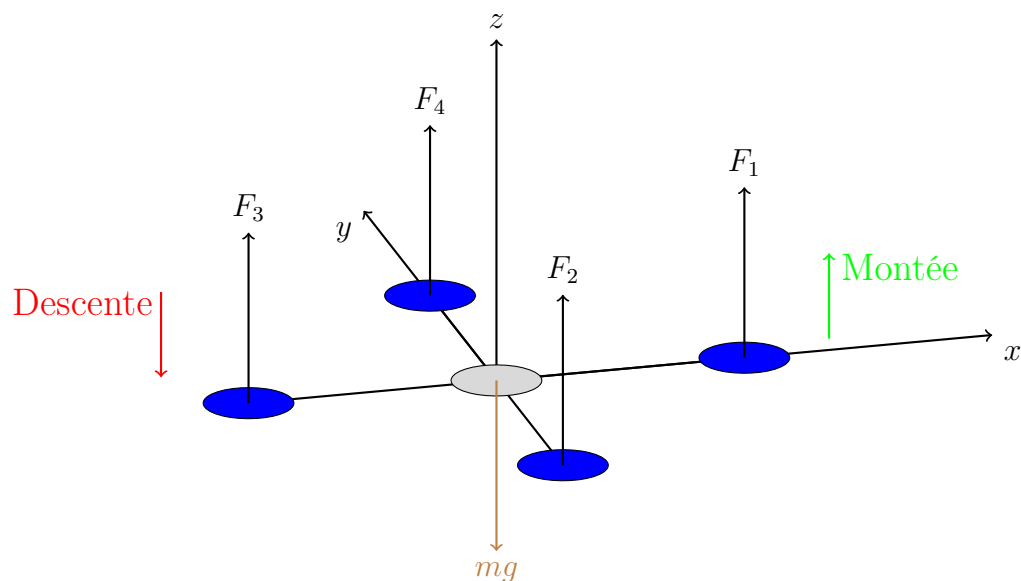


FIGURE 1.4 – Mouvement Vertical.

Mouvement de roulis

La (FIGURE 1.5) illustre la manière dont un mouvement de roulis est obtenu. Pour ce faire, un couple est appliqué autour de l'axe x en créant une différence de poussée entre les rotors 2 et 4. Ce mouvement de rotation autour de l'axe x est associé à une translation le long de l'axe y [18].

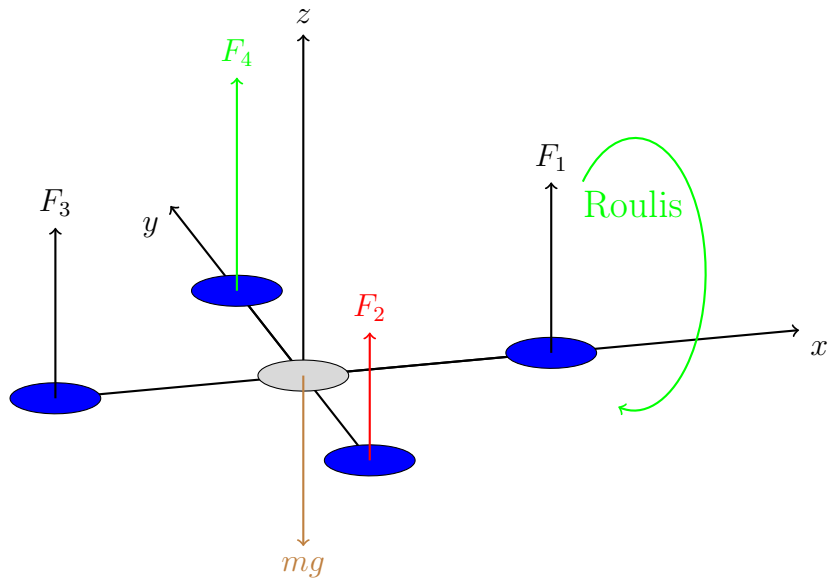


FIGURE 1.5 – Mouvement de Roulis.

Mouvement de tangage

La (Figure 1.6) montre comment un mouvement de tangage est obtenu. Dans ce cas, on applique un couple autour de l'axe y , c'est-à-dire en appliquant une différence de poussée entre le rotor 1 et le rotor 3. Ce mouvement (rotation autour de y) est couplé avec un mouvement de translation selon l'axe x [18].

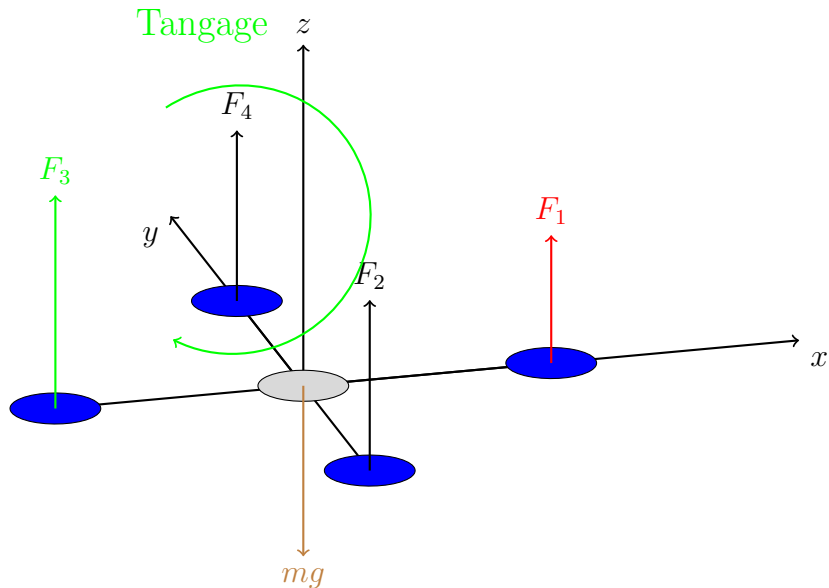


FIGURE 1.6 – Mouvement de Tangage.

Mouvement de lacet

La (FIGURE 1.7) montre comment le mouvement de lacet est obtenu. Dans ce cas, nous voulons appliquer un couple autour de l'axe z , qui est fait en appliquant une différence de vitesse entre les rotors 1,3 et 2,4. Ce mouvement n'est pas un résultat direct de la poussée produit par les propulseurs mais par les couples réactifs produits par la rotation des rotors. La direction de la force de poussée ne décale pas pendant le mouvement, mais l'augmentation de la force de portance dans une paire de rotors doit être égale à la diminution de l'autres paires pour s'assurer que toute la force de poussée demeure la même [18].

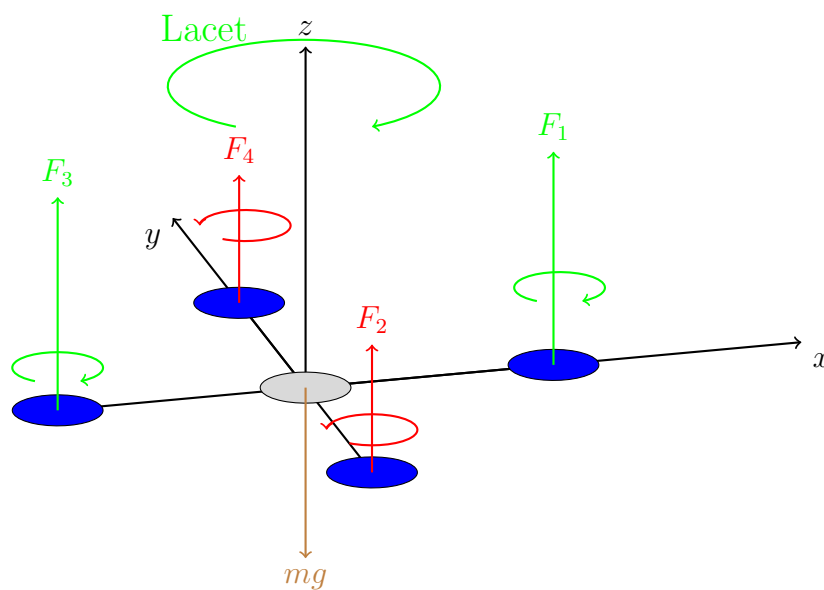


FIGURE 1.7 – Mouvement de Lacet.

Mouvements de translation

La (FIGURE 1.8) montre comment la translation horizontale est réalisée. Dans ce cas, nous voulons appliquer une force le long de x ou de y qui est fait en inclinant le corps (par le tangage ou le roulement) et en augmentant toute la poussée produite pour garder l'importance du composant de z de la poussée égale à la force de pesanteur [18].

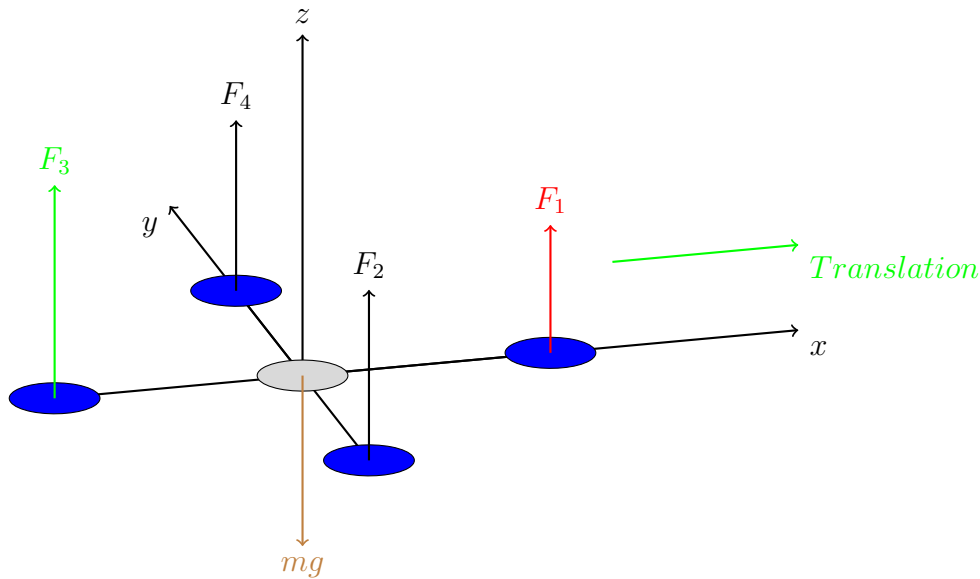


FIGURE 1.8 – Translations horizontales.

1.2.4 Modélisation dynamique du quadrirotor

De nombreuses tentatives de modélisation du drone sont documentées dans la littérature, telles que celle développée par Lozano en utilisant la méthode Euler-Lagrange. De même, le modèle proposé par Hamel, qui repose sur le formalisme de Newton, a été dérivé à partir de la dynamique d'un corps rigide associé au fuselage, avec l'ajout des forces aérodynamiques générées par les rotors. Ce modèle intègre également la dynamique des actionneurs. Notre propre approche de modélisation est également fondée sur le formalisme de Newton [23].

Modèle dynamique du quadrirotor

La modélisation des quadrirotors est particulièrement complexe en raison de la nature fortement non linéaire et couplée de la dynamique du système. Pour faciliter la compréhension du modèle dynamique que nous allons développer, nous adoptons les hypothèses suivantes [18] :

- La structure du quadrirotor est supposée rigide et symétrique, ce qui implique que la matrice d'inertie sera considérée comme diagonale.
- Les hélices sont supposées rigides, permettant ainsi de négliger l'effet de leur déformation pendant la rotation.
- Le centre de masse est supposé coïncider avec l'origine du repère lié à la structure.
- Les forces de portance et de traînée sont considérées comme proportionnelles

aux carrés de la vitesse de rotation des rotors, une approximation très proche du comportement aérodynamique réel du système.

Pour analyser le modèle mathématique du quadrirotor, deux repères sont utilisés : un repère fixe lié à la terre, R_b , et un autre repère mobile, R_m , associé au centre de gravité du quadrirotor (FIGURE 1.9). La transformation entre le repère mobile et le repère fixe est décrite par une matrice, appelée matrice de transformation T , qui intègre l'orientation et la position du repère mobile par rapport au repère fixe. Le placement des axes est choisi comme suit [18] :

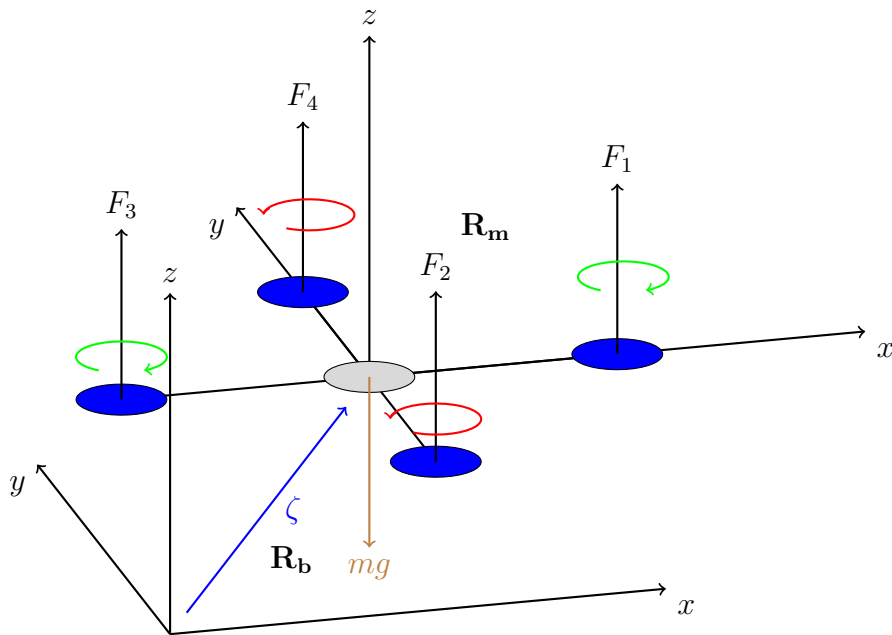


FIGURE 1.9 – Géométrie du quadrirotor.

$$T = \begin{bmatrix} R & \zeta \\ 0 & 1 \end{bmatrix} \quad (1.1)$$

Soit R la matrice de rotation décrivant l'orientation de l'objet mobile, et $\zeta = \begin{bmatrix} x & y & z \end{bmatrix}^T$ le vecteur de position. Pour déterminer les éléments de la matrice de rotation R , nous utilisons les angles d'Euler [18].

Angles d'Euler

Initialement, le repère mobile coïncide avec le repère fixe. Ensuite, le repère mobile effectue une rotation autour de l'axe x d'un angle de roulis ($-\frac{\pi}{2} < \Phi < \frac{\pi}{2}$), suivie d'une rotation autour de l'axe y d'un angle de tangage ($-\frac{\pi}{2} < \Theta < \frac{\pi}{2}$), puis d'une rotation

autour de l'axe z d'un angle de lacet ($-\pi < \Psi < \pi$). Ainsi, on obtient la formule de la matrice de rotation R [18] :

$$R = Rot_z(\Psi) \times Rot_y(\Theta) \times Rot_x(\Phi) = \begin{bmatrix} c\Psi & -s\Psi & 0 \\ s\Psi & c\Psi & 0 \\ 0 & 0 & 1 \end{bmatrix} \times \begin{bmatrix} c\Theta & 0 & s\Theta \\ 0 & 1 & 0 \\ -s\Theta & 0 & c\Theta \end{bmatrix} \times \begin{bmatrix} 1 & 0 & 0 \\ 0 & c\Phi & -s\Phi \\ 0 & s\Phi & c\Phi \end{bmatrix} \quad (1.2)$$

$$R = \begin{bmatrix} c\Psi c\Theta & s\Phi s\Theta c\Psi - s\Psi c\Phi & c\Phi s\Theta c\Psi + s\Psi s\Phi \\ s\Psi c\Theta & s\Phi s\Theta s\Psi + c\Psi c\Theta & c\Phi s\Theta s\Psi - s\Phi c\Psi \\ -s\Theta & s\Phi c\Theta & c\Phi c\Theta \end{bmatrix} \quad (1.3)$$

avec :

$$c = \cos$$

$$s = \sin$$

Vitesses angulaires

Les vitesses de rotation $\Omega_1, \Omega_2, \Omega_3$ dans le repère fixe sont exprimées en fonction des vitesses de rotation $\dot{\Phi}, \dot{\Theta}, \dot{\Psi}$ dans le repère mobile, comme suit :

$$\Omega = \begin{bmatrix} \Omega_1 \\ \Omega_2 \\ \Omega_3 \end{bmatrix} = \begin{bmatrix} \dot{\Phi} \\ 0 \\ 0 \end{bmatrix} + Rot_x(\Phi)^{-1} \begin{bmatrix} 0 \\ \dot{\Theta} \\ 0 \end{bmatrix} + (Rot_y(\Theta)Rot_x(\Phi))^{-1} \begin{bmatrix} 0 \\ 0 \\ \dot{\Psi} \end{bmatrix} \quad (1.4)$$

En effet, la rotation en roulis a lieu lorsque les repères sont encore confondus. Ensuite, pour le tangage, le vecteur représentant la rotation doit être exprimé dans le repère fixe, il est donc multiplié par $Rot_x(\Phi)^{-1}$. De même, le vecteur représentant la rotation en lacet doit être exprimé dans le repère fixe, qui a déjà subi deux rotations. On obtient ainsi [18] :

$$\Omega = \begin{bmatrix} \Omega_x \\ \Omega_y \\ \Omega_z \end{bmatrix} = \begin{bmatrix} \dot{\Phi} \\ 0 \\ 0 \end{bmatrix} + \begin{bmatrix} 0 \\ \dot{\Theta}c\Phi \\ -\dot{\Theta}s\Phi \end{bmatrix} + \begin{bmatrix} -\dot{\Psi}s\Theta \\ \dot{\Psi}s\Phi c\Theta \\ \dot{\Psi}c\Phi c\Theta \end{bmatrix} = \begin{bmatrix} \dot{\Phi} - \dot{\Psi}s\Theta \\ \dot{\Theta}c\Phi + \dot{\Psi}s\Phi c\Theta \\ \dot{\Psi}c\Phi c\Theta - \dot{\Theta}s\Phi \end{bmatrix} \quad (1.5)$$

$$\Omega = \begin{bmatrix} 1 & 0 & -s\Theta \\ 0 & c\Phi & s\Phi s\Theta \\ 0 & -s\Phi & c\Phi c\Theta \end{bmatrix} \times \begin{bmatrix} \dot{\Phi} \\ \dot{\Theta} \\ \dot{\Psi} \end{bmatrix} \quad (1.6)$$

Quand le quadrirotor fait des petites rotations, on peut faire les approximations suivantes :

$$c\Phi = c\Theta = c\Psi = 1, \text{ et } s\Phi = s\Theta = s\Psi = 0$$

Donc la vitesse angulaire sera :

$$\Omega = \begin{bmatrix} \dot{\Phi} \\ \dot{\Theta} \\ \dot{\Psi} \end{bmatrix} = \begin{bmatrix} \dot{\Phi} & \dot{\Theta} & \dot{\Psi} \end{bmatrix}^T \quad (1.7)$$

Vitesses linéaires

Les vitesses linéaires v_x^b, v_y^b, v_z^b dans le repère fixe en fonction des vitesses linéaires v_x^m, v_y^m, v_z^m dans le repère mobile sont données par [18] :

$$v = \begin{bmatrix} v_x^b \\ v_y^b \\ v_z^b \end{bmatrix} = R \times \begin{bmatrix} v_x^m \\ v_y^m \\ v_z^m \end{bmatrix} \quad (1.8)$$

1.2.5 Effets physiques agissants sur le quadrirotor

Les forces

Les forces agissant sur le système sont [18] :

- **Le poids du quadrirotor** : Il est donné par $P = mg$, où m est la masse totale et g la gravité.
- **Les forces de poussée** : Ce sont des forces générées par la rotation des moteurs, perpendiculaires au plan des hélices. Ces forces sont proportionnelles au carré de la vitesse de rotation des moteurs :

$$F_i = b\omega_i^2 \quad (1.9)$$

Avec $i = 1 : 4$, et b étant le coefficient de portance, dépendant de la forme et du nombre des pales, ainsi que de la densité de l'air.

Les forces de traînée : La force de traînée résulte du couplage entre une force de pression et une force de frottement visqueux. Dans ce cas, deux forces de traînée agissent sur le système :

- **La traînée dans les hélices** : Elle agit sur les pales, est proportionnelle à la densité de l'air, à la forme des pales, et au carré de la vitesse de rotation de l'hélice. Elle est donnée par la relation suivante :

$$T_h = d\omega^2 \quad (1.10)$$

Avec d étant le coefficient de traînée, qui dépend de la fabrication de l'hélice.

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

$$F_t = K_{ft}v \quad (1.11)$$

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

Les moments

Il y a plusieurs moments agissants sur le quadrirotor, ces moments sont dus aux forces de poussée et de traînée et aux effets gyroscopiques [18].

Moments dus aux forces de poussée

- **La rotation autour de l'axe x** : Elle est due au moment créé par la différence entre les forces de portance des rotors 2 et 4. Ce moment est donné par la relation suivante :

$$M_x = l(F_4 - F_2) = lb(\omega_4^2 - \omega_2^2) \quad (1.12)$$

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

- **La rotation autour de l'axe y** : Elle est due au moment créé par la différence entre les forces de portance des rotors 1 et 3. Ce moment est donné par la relation suivante :

$$M_y = l(F_3 - F_1) = lb(\omega_3^2 - \omega_1^2) \quad (1.13)$$

Moments dus aux forces de traînée

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

$$M_z = d(\omega_1^2 - \omega_2^2 + \omega_3^2 - \omega_4^2) \quad (1.14)$$

- **Le moment résultant des frottements aérodynamiques** : Il est donné par la relation suivante :

$$M_a = K_{fa}\Omega^2 \quad (1.15)$$

avec : K_{fa} le coefficient des frottements aérodynamiques et Ω la vitesse angulaire.

Effet gyroscopique

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

Dans notre cas, il y a deux moments gyroscopiques : le premier est le moment gyroscopique des hélices, et le second est dû aux mouvements du quadrirotor [18].

— **Moment gyroscopique des hélices** : Il est donné par la relation suivante :

$$M_{gh} = \sum_{i=1}^4 \Omega \wedge J_r \left[0 \quad 0 \quad (-1)^{i+1} \omega_i \right]^T \quad (1.16)$$

avec J_r étant l'inertie des rotors.

— **Moment gyroscopique dû aux mouvements du quadrirotor** : Il est donné par la relation suivante :

$$M_{gm} = \Omega \wedge J \Omega \quad (1.17)$$

avec J est l'inertie du système.

1.2.6 Développement du Modèle mathématique selon Newton-Euler

En utilisant la formulation de Newton-Euler, les équations sont écrites sous la forme suivante [18] :

$$\begin{cases} \dot{\zeta} = v \\ m\ddot{\zeta} = F_f + F_t + F_g \\ \dot{R} = RS(\Omega) \\ J\dot{\Omega} = -\Omega \wedge J\Omega + M_f - M_a - M_{gh} \end{cases} \quad (1.18)$$

avec :

ζ : Vecteur de position du quadrirotor

m : Masse totale du quadrirotor

Ω : Vitesse angulaire exprimée dans le repère fixe

R : Matrice de rotation

\wedge : Produit vectoriel

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

$$J = \begin{bmatrix} I_x & 0 & 0 \\ 0 & I_y & 0 \\ 0 & 0 & I_z \end{bmatrix} \quad (1.19)$$

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

$$S(\Omega) = \begin{bmatrix} 0 & -\Omega_3 & \Omega_2 \\ \Omega_3 & 0 & -\Omega_1 \\ -\Omega_2 & \Omega_1 & 0 \end{bmatrix} \quad (1.20)$$

\mathbf{F}_f : Force totale générée par les quatre rotors, elle est donnée par :

$$F_f = R \times \left[0 \ 0 \ \sum_{i=1}^4 F_i \right]^T \quad (1.21)$$

$$F_i = b\omega_i^2 \quad (1.22)$$

\mathbf{F}_t : Force de traînée selon les axes (x, y, z) , elle est donnée par :

$$F_t = \begin{bmatrix} -K_{ftx} & 0 & 0 \\ 0 & -K_{fity} & 0 \\ 0 & 0 & -K_{ftz} \end{bmatrix} \dot{\zeta} \quad (1.23)$$

$K_{ftx}, K_{fity}, K_{ftz}$: Coefficients de traînée de translation,

\mathbf{F}_g : Force de gravité, elle est donnée par :

$$F_g = \begin{bmatrix} 0 \\ 0 \\ -mg \end{bmatrix} \quad (1.24)$$

\mathbf{M}_f : Moment provoqué par les forces de poussée et de traînée :

$$M_f = \begin{bmatrix} l(F_4 - F_2) \\ l(F_3 - F_1) \\ d(\omega_1^2 - \omega_2^2 + \omega_3^2 - \omega_4^2) \end{bmatrix} \quad (1.25)$$

\mathbf{M}_a : Moment résultant des frottements aérodynamiques, il est donnée par :

$$M_a = \begin{bmatrix} K_{fax} \dot{\Phi}^2 \\ K_{fay} \dot{\Theta}^2 \\ K_{faz} \dot{\Psi}^2 \end{bmatrix} \quad (1.26)$$

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

1.2.7 Equations de mouvement de translation

On a :

$$m\ddot{\zeta} = F_f + F_t + F_g \quad (1.27)$$

On remplace chaque force par sa formule, on trouve [18] :

$$m \begin{bmatrix} \ddot{x} \\ \ddot{y} \\ \ddot{z} \end{bmatrix} = \begin{bmatrix} c\Phi c\Psi s\Theta + s\Phi s\Psi \\ c\Phi s\Theta s\Psi - s\Phi c\Psi \\ c\Phi c\Theta \end{bmatrix} \sum_{i=1}^4 F_i - \begin{bmatrix} K_{ftx} \dot{x} \\ K_{fity} \dot{y} \\ K_{ftz} \dot{z} \end{bmatrix} - \begin{bmatrix} 0 \\ 0 \\ mg \end{bmatrix} \quad (1.28)$$

On obtient alors les équations différentielles qui définissent le mouvement de translation [18] :

$$\begin{cases} \ddot{x} = \frac{1}{m} (c\Phi c\Psi s\Theta + s\Phi s\Psi) \left(\sum_{i=1}^4 F_i \right) - \frac{K_{ftx}}{m} \dot{x} \\ \ddot{y} = \frac{1}{m} (c\Phi s\Theta s\Psi - s\Phi c\Psi) \left(\sum_{i=1}^4 F_i \right) - \frac{K_{fity}}{m} \dot{y} \\ \ddot{z} = \frac{1}{m} (c\Phi c\Theta) \left(\sum_{i=1}^4 F_i \right) - \frac{K_{ftz}}{m} \dot{z} - g \end{cases} \quad (1.29)$$

1.2.8 Equations de mouvement de rotation

On a :

$$J\dot{\Omega} = -M_{gm} - M_{gh} - M_a + M_f \quad (1.30)$$

On remplace chaque moment par la formule correspondant, on trouve [18] :

$$\begin{bmatrix} I_x & 0 & 0 \\ 0 & I_y & 0 \\ 0 & 0 & I_z \end{bmatrix} \begin{bmatrix} \ddot{\Phi} \\ \ddot{\Theta} \\ \ddot{\Psi} \end{bmatrix} = - \begin{bmatrix} \dot{\Phi} \\ \dot{\Theta} \\ \dot{\Psi} \end{bmatrix} \wedge \left(\begin{bmatrix} I_x & 0 & 0 \\ 0 & I_y & 0 \\ 0 & 0 & I_z \end{bmatrix} \begin{bmatrix} \dot{\Phi} \\ \dot{\Theta} \\ \dot{\Psi} \end{bmatrix} \right) - \begin{bmatrix} -J_r \bar{\Omega}_r \dot{\Theta} \\ -J_r \bar{\Omega}_r \dot{\Phi} \\ 0 \end{bmatrix} - \begin{bmatrix} K_{fax} \dot{\Phi}^2 \\ K_{fay} \dot{\Theta}^2 \\ K_{faz} \dot{\Psi}^2 \end{bmatrix} + \begin{bmatrix} lb(\omega_4^2 - \omega_2^2) \\ lb(\omega_3^2 - \omega_1^2) \\ d(\omega_1^2 - \omega_2^2 + \omega_3^2 - \omega_4^2) \end{bmatrix} \quad (1.31)$$

On obtient alors les équations différentielles définissant le mouvement de rotation [18] :

$$\begin{cases} I_x \ddot{\Phi} = -\dot{\Theta} \dot{\Psi} (I_z - I_y) - J_r \bar{\Omega}_r \dot{\Theta} - K_{fax} \dot{\Phi}^2 + lb(\omega_4^2 - \omega_2^2) \\ I_y \ddot{\Theta} = \dot{\Phi} \dot{\Psi} (I_z - I_x) + J_r \bar{\Omega}_r \dot{\Phi} - K_{fay} \dot{\Theta}^2 + lb(\omega_3^2 - \omega_1^2) \\ I_z \ddot{\Psi} = -\dot{\Phi} \dot{\Theta} (I_y - I_x) - K_{faz} \dot{\Psi}^2 + d(\omega_1^2 - \omega_2^2 + \omega_3^2 - \omega_4^2) \end{cases} \quad (1.32)$$

Avec :

$$\bar{\Omega}_r = \omega_1 - \omega_2 + \omega_3 - \omega_4 \quad (1.33)$$

En conséquence, le modèle dynamique complet qui régit le quadrirotor est le suivant [18] :

$$\begin{cases} \ddot{\Phi} = \frac{(I_y - I_z)}{I_x} \dot{\Theta} \dot{\Psi} - \frac{J_r \bar{\Omega}_r \dot{\Theta}}{I_x} - \frac{K_{fax} \dot{\Phi}^2}{I_x} + \frac{l}{I_x} u_2 \\ \ddot{\Theta} = \frac{(I_z - I_x)}{I_y} \dot{\Phi} \dot{\Psi} + \frac{J_r \bar{\Omega}_r \dot{\Phi}}{I_y} - \frac{K_{fay} \dot{\Theta}^2}{I_y} + \frac{l}{I_y} u_3 \\ \ddot{\Psi} = \frac{(I_x - I_y)}{I_z} \dot{\Theta} \dot{\Phi} - \frac{K_{faz} \dot{\Psi}^2}{I_z} + \frac{1}{I_z} u_4 \\ \ddot{x} = -\frac{K_{ftx}}{m} \dot{x} + \frac{1}{m} u_x u_1 \\ \ddot{y} = -\frac{K_{fty}}{m} \dot{y} + \frac{1}{m} u_y u_1 \\ \ddot{z} = -\frac{K_{ftz}}{m} \dot{z} - g + \frac{\cos(\Phi) \cos(\Theta)}{m} u_1 \end{cases} \quad (1.34)$$

$$\text{Avec : } \begin{cases} u_x = (c\Phi c\Psi s\Theta + s\Phi s\Psi) \\ u_y = (c\Phi s\Theta s\Psi - s\Phi c\Psi) \end{cases} \quad (1.35)$$

$$\text{Et : } \begin{bmatrix} u_1 \\ u_2 \\ u_3 \\ u_4 \end{bmatrix} = \begin{bmatrix} b & b & b & b \\ 0 & -lb & 0 & lb \\ -lb & 0 & lb & 0 \\ d & -d & d & -d \end{bmatrix} \begin{bmatrix} \omega_1^2 \\ \omega_2^2 \\ \omega_3^2 \\ \omega_4^2 \end{bmatrix} \quad (1.36)$$

À partir de (1.35), on trouve :

$$\begin{cases} \Phi_d = \sin^{-1}(u_x \sin(\Psi_d) - u_y \cos(\Psi_d)) \\ \Theta_d = \sin^{-1} \left(\frac{u_x \cos(\Psi_d) + u_y \sin(\Psi_d)}{\cos(\Phi_d)} \right) \end{cases} \quad (1.37)$$

1.2.9 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 [18] :

$$X = \left[\Phi \quad \dot{\Phi} \quad \Theta \quad \dot{\Theta} \quad \Psi \quad \dot{\Psi} \quad x \quad \dot{x} \quad y \quad \dot{y} \quad z \quad \dot{z} \right]^T \quad (1.38)$$

On obtient le vecteur d'état suivante :

$$X = \left[x_1 \quad x_2 \quad x_3 \quad x_4 \quad x_5 \quad x_6 \quad x_7 \quad x_8 \quad x_9 \quad x_{10} \quad x_{11} \quad x_{12} \right]^T \quad (1.39)$$

On obtient la représentation d'état suivante :

$$\begin{cases} \dot{x}_1 = x_2 \\ \dot{x}_2 = a_1 x_4 x_6 + a_2 x_2^2 + a_3 \bar{\Omega}_r x_4 + b_1 u_2 \\ \dot{x}_3 = x_4 \\ \dot{x}_4 = a_4 x_2 x_6 + a_5 x_4^2 + a_6 \bar{\Omega}_r x_2 + b_2 u_3 \\ \dot{x}_5 = x_6 \\ \dot{x}_6 = a_7 x_2 x_4 + a_8 x_6^2 + b_3 u_4 \\ \dot{x}_7 = x_8 \\ \dot{x}_8 = a_9 x_8 + \frac{1}{m} u_x u_1 \\ \dot{x}_9 = x_{10} \\ \dot{x}_{10} = a_{10} x_{10} + \frac{1}{m} u_y u_1 \\ \dot{x}_{11} = x_{12} \\ \dot{x}_{12} = a_{11} x_{12} + \frac{\cos(\Phi) \cos(\Theta)}{m} u_1 - g \end{cases} \quad (1.40)$$

$$\text{Avec : } \begin{cases} a_1 = \frac{I_y - I_z}{I_x}, & a_2 = -\frac{K_{fax}}{I_x}, & a_3 = -\frac{J_r}{I_x}, \\ a_4 = \frac{I_z - I_x}{I_y}, & a_5 = -\frac{K_{fay}}{I_y}, & a_6 = \frac{J_r}{I_y}, \\ a_7 = \frac{I_x - I_y}{I_z}, & a_8 = -\frac{K_{faz}}{I_z}, & a_9 = -\frac{K_{ftx}}{m}, \\ a_{10} = -\frac{K_{f ty}}{m}, & a_{11} = -\frac{K_{ftz}}{m}, & b_1 = \frac{l}{I_x}, \\ b_2 = \frac{l}{I_y}, & b_3 = \frac{1}{I_z} \end{cases} \quad (1.41)$$

1.2.10 Etude de la dynamique des moteurs

La fonction de transfert d'un moteur électrique est du second ordre et s'écrit sous la forme suivante [23] :

$$H(p) = \frac{K}{K^2 + Rf + (Rj + Lf)p + LJp^2} \quad (1.42)$$

K : Gain du moteur en V.s/rad

R : Résistance interne du moteur en Ω

L : Inductance en H

f : Frottements

J : Inertie du rotor en g.cm²

Ainsi, cette fonction de transfert peut être approximée par un premier ordre en négligeant certains éléments. En effet, l'inductance est de l'ordre du millihenry, elle est donc négligeable devant la résistance du moteur qui est de l'ordre de la dizaine d'ohms. De même, les frottements d'un moteur sont négligeables devant l'inertie du rotor [23].

On obtient donc :

$$H(p) = \frac{K}{K^2 + RJp} \Rightarrow H(p) = \frac{k}{1 + \tau p} \quad (1.43)$$

1.3 Commande PID

1.3.1 Introduction

La régulation PID (Proportionnelle-Intégrale-Dérivée) est une méthode de contrôle utilisée dans les systèmes automatisés pour maintenir une variable de processus (comme la température, la vitesse, ou la position) à une valeur désirée [17].

Voici une définition détaillée de chacun des termes :

1.3.2 Proportionnelle (P)

La composante proportionnelle est directement liée à l'erreur actuelle, qui est la différence entre la valeur mesurée et la valeur souhaitée (setpoint). La formule générale pour la composante proportionnelle est [17] :

$$P = K_p e(t) \quad (1.44)$$

où :

K_p : Gain proportionnel.

$e(t)$: Erreur à l'instant t .

L'effet de la composante proportionnelle est de fournir une réponse corrective immé-

diante en proportion directe à l'erreur. Un grand K_p entraîne une réponse rapide, mais peut provoquer des oscillations et de l'instabilité.

1.3.3 Intégrale (I)

La composante intégrale tient compte de l'erreur cumulée au fil du temps. Elle est utilisée pour éliminer l'erreur résiduelle (offset) qui peut persister lorsque seule la composante proportionnelle est utilisée. La formule pour la composante intégrale est [17] :

$$I = K_i \int_0^t e(\tau) d\tau \quad (1.45)$$

où :

K_i : Gain intégral.

$\int_0^t e(\tau) d\tau$: Intégrale de l'erreur de 0 à t .

La composante intégrale augmente la sortie jusqu'à ce que l'erreur soit nulle, mais une trop grande intégration peut entraîner un dépassement et des oscillations.

1.3.4 Dérivée (D)

La composante dérivée est liée à la vitesse de changement de l'erreur. Elle anticipe l'avenir en réagissant à la variation de l'erreur. La formule pour la composante dérivée est [17] :

$$D = K_d \frac{de(t)}{dt} \quad (1.46)$$

où :

K_d : Gain dérivé.

$\frac{de(t)}{dt}$: Dérivée de l'erreur par rapport au temps.

La composante dérivée aide à réduire la tendance du système à osciller en amortissant les variations rapides de l'erreur.

1.3.5 Combinaison PID

La sortie globale du contrôleur PID est la somme des trois composantes (FIGURE 1.10) [17] :

$$sortie = P + I + D = K_p e(t) + K_i \int_0^t e(\tau) d\tau + K_d \frac{de(t)}{dt} \quad (1.47)$$

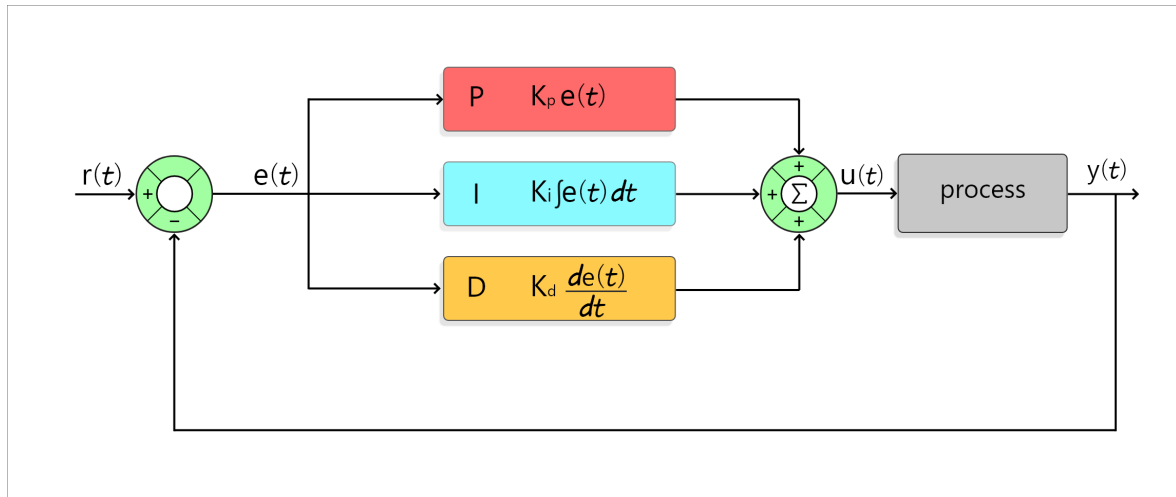


FIGURE 1.10 – Principe du contrôleur PID.

1.3.6 Avantages et inconvénients

Avantages

La commande PID est largement utilisée pour les drones en raison de plusieurs avantages et caractéristiques qui la rendent particulièrement adaptée à ce type de système de contrôle. Voici les raisons principales :

Simplicité et efficacité

- **Facilité de compréhension et de mise en œuvre** : Les algorithmes PID sont relativement simples à comprendre et à implémenter. Cela les rend accessibles pour une large gamme d'applications, y compris les drones.
- **Adaptabilité** : Le PID peut être ajusté pour une grande variété de systèmes, ce qui en fait une solution polyvalente.

Performance en temps réel

- **Réponse rapide** : Le terme proportionnel permet de réagir rapidement aux erreurs en ajustant la commande en fonction de l'écart instantané entre la consigne et la sortie.
- **Élimination de l'erreur statique** : Le terme intégral accumule les erreurs passées et ajuste la commande pour éliminer les erreurs statiques à long terme.
- **Amortissement des oscillations** : Le terme dérivé prédit les tendances futures de l'erreur et ajuste la commande pour amortir les oscillations, assurant ainsi une réponse plus stable.

Robustesse et stabilité

- **Robustesse** : Les contrôleurs PID peuvent maintenir la performance même en présence de perturbations et de variations dans les paramètres du système.
- **Stabilité** : Avec un bon réglage des gains (K_p , K_i , K_d), un contrôleur PID peut assurer une stabilité robuste, ce qui est essentiel pour les drones en vol.

Flexibilité et adaptabilité

- **Réglage des paramètres** : Les gains du PID (proportionnel, intégral, dérivé) peuvent être ajustés pour répondre à des conditions spécifiques et des dynamiques variées des drones.
- **Compensation des perturbations** : Les contrôleurs PID peuvent s'ajuster pour compenser les perturbations externes, comme le vent ou les changements de charge.

Inconvénients

Bien que la commande PID présente de nombreux avantages, elle comporte également quelques inconvénients :

- **Nécessité d'un bon réglage** : Les performances optimales dépendent d'un réglage précis des gains (K_p , K_i , K_d). Trouver les valeurs correctes peut nécessiter du temps et de l'expérimentation.
- **Sensibilité aux perturbations et bruit** : La composante dérivée peut amplifier le bruit dans le signal de l'erreur, ce qui peut entraîner des oscillations indésirables.
- **Complexité avec systèmes non linéaires** : Les systèmes de drones peuvent présenter des comportements non linéaires, et un contrôleur PID peut avoir des difficultés à s'adapter efficacement à ces conditions sans ajustements supplémentaires.
- **Limitation des performances dynamiques** : En présence de changements rapides ou de conditions dynamiques complexes, un contrôleur PID seul peut ne pas être suffisant pour maintenir une performance optimale sans l'ajout de méthodes de contrôle avancées.

En dépit de ces inconvénients, la commande PID reste une méthode de contrôle puissante et largement utilisée pour les drones en raison de sa simplicité, de sa robustesse et de sa capacité à être ajustée pour diverses conditions de vol.

1.4 Simulation

Pour évaluer les performances du correcteur PID, plusieurs simulations ont été effectuées. Celles-ci incluent le suivi de trajectoires simples (consignes sinus et cosinus) ainsi que la réalisation d'une manœuvre complexe en spirale 3D. Les résultats obtenus mettent en évidence l'efficacité du correcteur PID pour assurer un contrôle précis des positions et des orientations, tout en minimisant les erreurs.

La figure suivante (FIGURE 1.11) illustre le suivi des trajectoires $X(t)$, $Y(t)$ et $Z(t)$ mesurées par le système par rapport aux consignes imposées ($\sin(t)$, $\cos(t)$ et une constante égale à 1). Le correcteur PID a permis de réduire l'écart entre les consignes et les valeurs mesurées, démontrant un suivi précis et une stabilité en position.

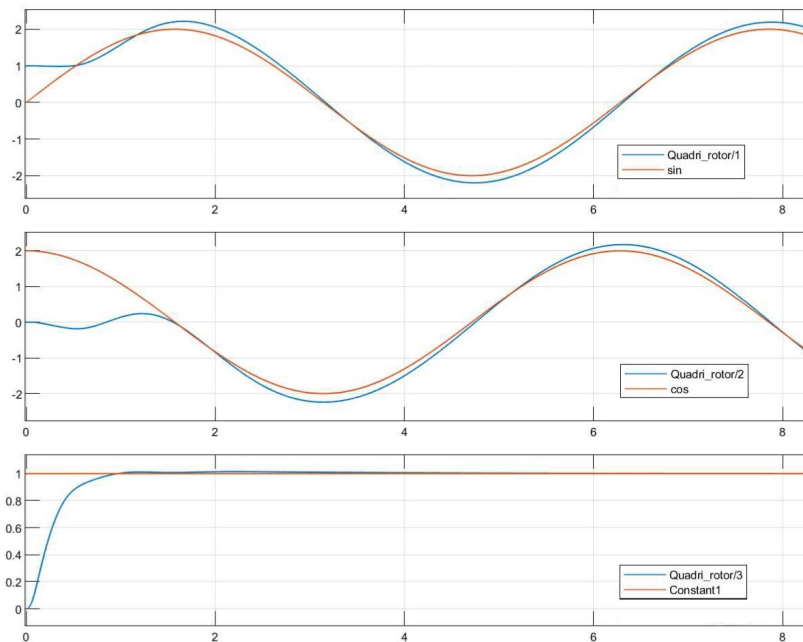


FIGURE 1.11 – Suivi des trajectoires $X(t)$, $Y(t)$ et $Z(t)$ avec les consignes sinus, cosinus et constante.

Cette figure (FIGURE 1.12) représente une manœuvre en spirale visualisée en 3D et dans des projections 2D (plans XY , XZ , et YZ). Elle montre la trajectoire réalisée par le quadrirotor sous l'effet du correcteur PID, qui maintient le drone sur une trajectoire préprogrammée en spirale, même dans un espace tridimensionnel.

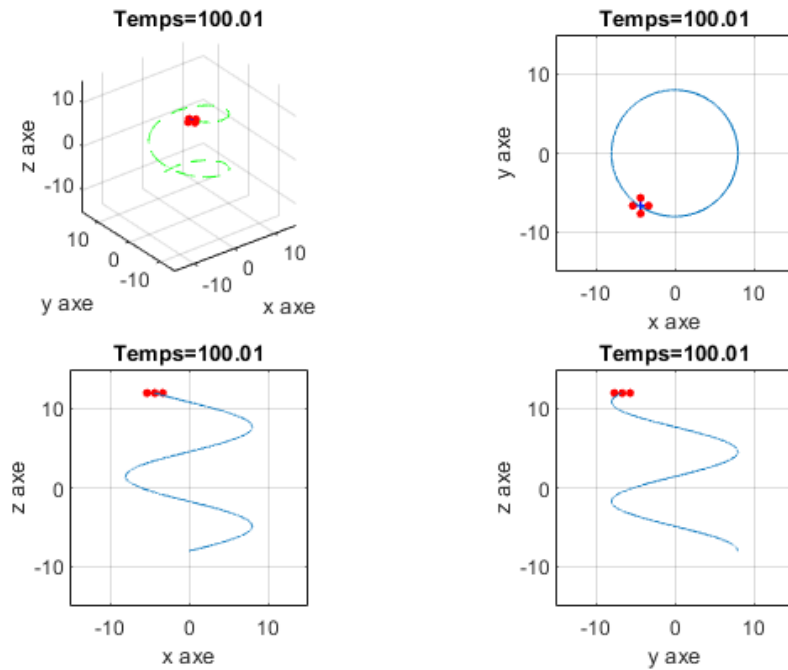


FIGURE 1.12 – Simulation d’une manœuvre en spirale en 3D et ses projections 2D.

La figure (FIGURE 1.13) compare les positions mesurées (x, y, z) et les angles d’orientation (ϕ, θ, ψ) (roll, pitch, yaw) du quadrirotor avec les consignes imposées pendant la manœuvre en spirale. La superposition entre les mesures et les références montre une précision remarquable du correcteur PID dans le suivi des trajectoires et des attitudes.

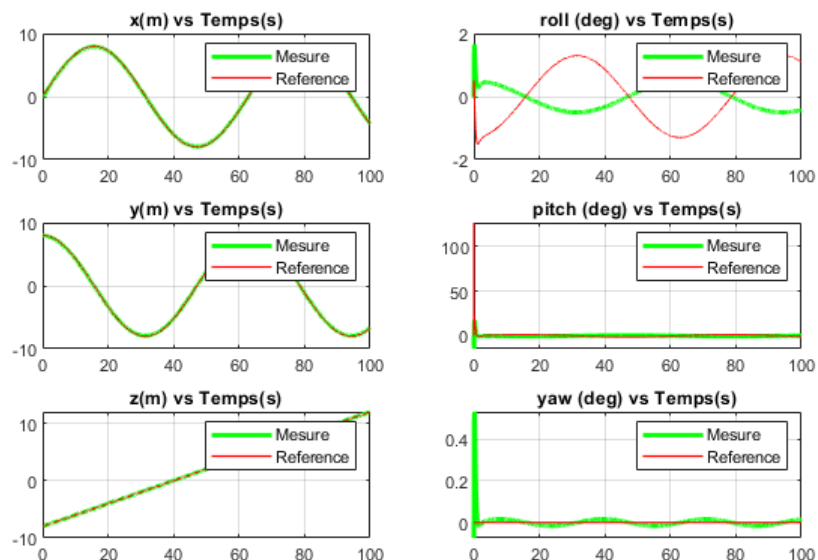


FIGURE 1.13 – Comparaison des mesures $(x, y, z, \phi, \theta, \psi)$ avec les consignes pour la manœuvre en spirale.

Cette figure (FIGURE 1.14) analyse les erreurs en position (x, y, z) et en orienta-

tion (ψ) pendant la manœuvre en spirale. Les résultats montrent des erreurs faibles et contrôlées, ce qui reflète l'efficacité du correcteur PID dans la stabilisation et la réduction des écarts.

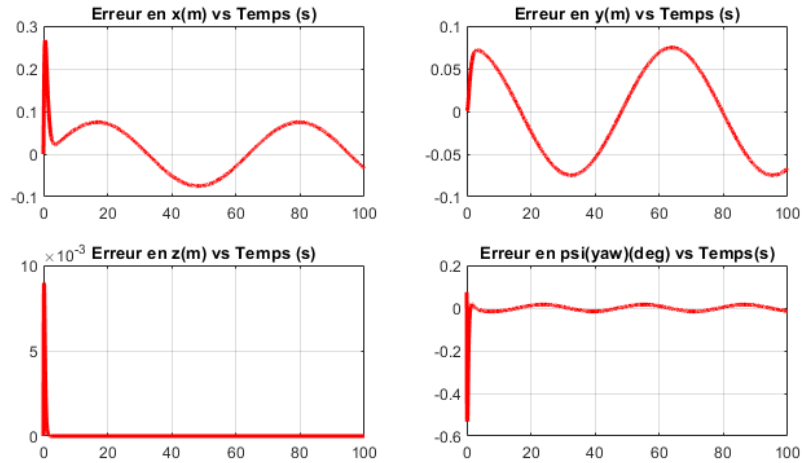


FIGURE 1.14 – Analyse des erreurs en position (x, y, z) et en orientation (ψ) pour la manœuvre en spirale.

Les simulations réalisées mettent en évidence l'efficacité du correcteur PID pour le contrôle du quadrirotor, que ce soit pour des trajectoires simples ou des manœuvres complexes comme la spirale en 3D. Les analyses des positions, des orientations et des erreurs montrent une stabilité et une précision élevées, confirmant la pertinence de cette méthode pour ce type de système.

1.5 Conclusion

En conclusion de ce chapitre, nous avons exploré la modélisation du quadrirotor à travers une étude générale des drones, leur classification, et leurs applications. L'analyse des dynamiques de vol et la modélisation mathématique selon les approches de Newton-Euler ont permis de comprendre les comportements complexes du système. La commande PID, introduite comme méthode clé de régulation, s'est révélée efficace pour stabiliser le système malgré ses limitations face à des perturbations ou des conditions non linéaires. Les simulations effectuées, incluant des trajectoires simples et une manœuvre complexe en spirale, ont démontré la capacité du PID à assurer un suivi précis avec des erreurs faibles et contrôlées.

Dans le chapitre suivant, nous explorerons la composition matérielle du quadrirotor en détaillant ses principaux composants, notamment la structure, les moteurs, ainsi que les systèmes de propulsion et de contrôle.

Chapitre 2

Constitution du drone

2.1 Introduction

Dans ce chapitre, nous allons détailler les différents composants essentiels qui constituent un drone, en particulier un quadrirotor. L'analyse commencera par le châssis, qui forme la structure de base, avant de se concentrer sur le système de propulsion, comprenant les moteurs, les hélices, les contrôleurs électroniques de vitesse (ESC), la batterie LIPO, et le distributeur de puissance PDB-XT60. Par la suite, nous aborderons le contrôleur de vol, composé du microcontrôleur et des capteurs, qui joue un rôle crucial dans la gestion et la stabilisation du drone en vol. Enfin, nous conclurons par une discussion sur les systèmes de commande à distance, que ce soit via une radiocommande traditionnelle ou une commande Bluetooth, permettant à l'utilisateur de piloter le drone de manière efficace et précise.

2.2 Châssis

Le châssis d'un drone est la structure de base qui soutient et maintient tous les autres composants ensemble. Conçu pour être à la fois léger et robuste, il est capable de résister aux forces générées pendant le vol et les atterrissages. Le châssis inclut généralement les bras sur lesquels sont montés les moteurs et les hélices, ainsi que le cadre central qui héberge les éléments essentiels tels que la batterie, les contrôleurs de vol et d'autres équipements électroniques. Un bon châssis est crucial pour la stabilité, la maniabilité et la durabilité du drone [30].

Dans notre cas, nous avons choisi d'utiliser un châssis DJI F450 (FIGURE 2.1) pour ses nombreuses qualités. Il est facilement disponible et présente des propriétés intéressantes, notamment :

- Facilité d'assemblage.
- Robustesse et durabilité.
- Polyvalence.
- Stabilité en vol.
- Excellent rapport qualité-prix.

Ces caractéristiques font du DJI F450 un choix idéal pour notre projet, garantissant des performances fiables et une grande adaptabilité aux besoins spécifiques de notre drone.



FIGURE 2.1 – DJI F450 [2].

2.3 Systeme de propulsion

2.3.1 Moteurs

Les moteurs d'un drone sont des dispositifs électromécaniques qui convertissent l'énergie électrique en énergie mécanique pour faire tourner les hélices, permettant ainsi la propulsion et le contrôle du vol. Principalement, les drones utilisent des moteurs à courant continu sans balais (brushless) en raison de leur efficacité, durabilité et puissance supérieure. Ces moteurs se composent d'un stator et d'un rotor, fonctionnant sans contact direct pour réduire l'usure. Les caractéristiques clés des moteurs de drone incluent la vitesse (KV Rating), la puissance, l'efficacité et la durabilité, influençant fortement les performances globales du drone [8][6].

Pour notre projet, nous avons considéré deux types de moteurs brushless : le 2200kv et le 1000kv (FIGURE 2.2).



FIGURE 2.2 – Moteurs brushless 1000KV et 2200KV [31].

Afin de déterminer lequel des deux était le plus adapté, nous avons effectué plusieurs tests :

- Les quatre moteurs ensemble produisent une poussée de 2 à 3 fois celle d'un seul moteur.
- Notre drone pèse environ 1 kg, et pour le faire décoller, un seul moteur doit fournir une poussée supérieure à 500 g. Nous avons décidé d'ajouter une marge de sécurité de 200 g.
- Pour mesurer la poussée de manière pratique, nous avons fixé le moteur sur une balance en plaçant l'hélice à l'envers pour que la poussée soit dirigée vers le bas. Ensuite, nous avons connecté l'ESC au moteur et à l'Arduino, et alimenté l'ensemble avec une batterie LiPo (FIGURE 2.3). À l'aide d'un potentiomètre, nous avons ajusté la vitesse en fonction de la tension et du courant consommé, puis nous avons mesuré la masse poussée.

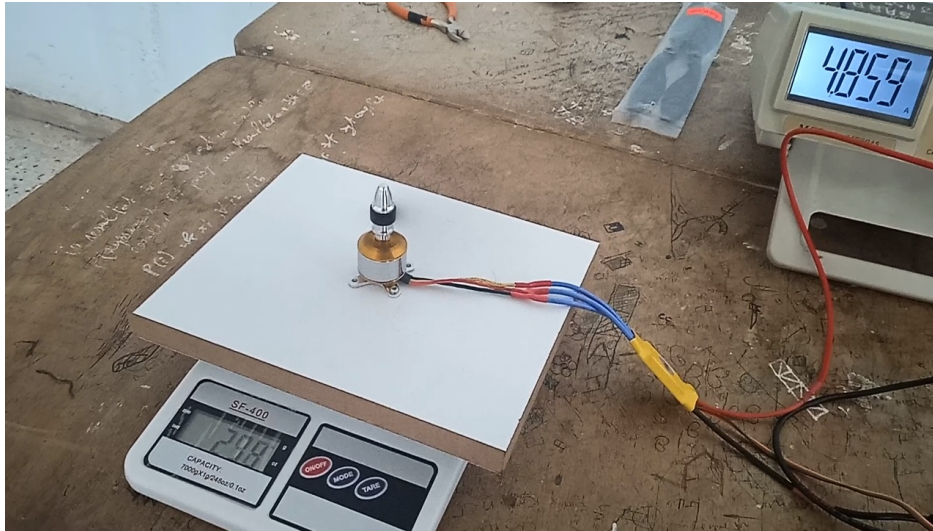


FIGURE 2.3 – Mesure de la masse poussée.

Les résultats de nos mesures sont présentés dans le tableau suivant :

Type de moteur	Mesures	Courant (A)	PWM	Tension (V)
2200kv		15	175	7.6
1000kv		8	255	11.1

Puissance (W)	Autonomie de la batterie (min)	Poussée d'un moteur (kg)
114	4.2	0.7
88.8	7.8	0.7

Poussée de 4 moteurs (kg)
1.4 - 2.1
1.4 - 2.1

TABLE 2.1 – Comparaison entre les moteurs brushless 2200kv et 1000kv.

Sur la base des résultats obtenus, nous avons opté pour le moteur brushless 1000kv. Pour vérifier l'exactitude de nos calculs, nous avons construit un prototype de drone sans gyroscope ni accéléromètre, équipé d'un capteur Bluetooth HC-05. Nous avons graduellement augmenté la vitesse des quatre moteurs de manière synchronisée à l'aide de l'application que nous avons développée (FIGURE 2.16), jusqu'à ce que le drone décolle. Cette expérience visait à confirmer que le drone était capable de voler.

Remarque : Ces moteurs brushless contiennent trois fils car ils fonctionnent de manière similaire aux moteurs asynchrones. Ces fils sont utilisés pour contrôler les bobines du stator à travers un contrôleur électronique de vitesse (ESC), qui gère le courant et

la séquence des phases pour produire un champ magnétique tournant. Cette configuration permet une plus grande efficacité, une meilleure performance et une réponse plus rapide aux commandes de vol, contribuant ainsi à une expérience de vol stable et précise.

2.3.2 Hélices

Les hélices sont des composants essentiels d'un drone, responsables de générer la poussée nécessaire au vol. Elles convertissent la puissance des moteurs en force de propulsion, permettant au drone de décoller, de se stabiliser et de se déplacer dans différentes directions. Les hélices sont disponibles en différentes tailles, formes et matériaux, et leur choix impacte directement les performances et la maniabilité du drone [12].

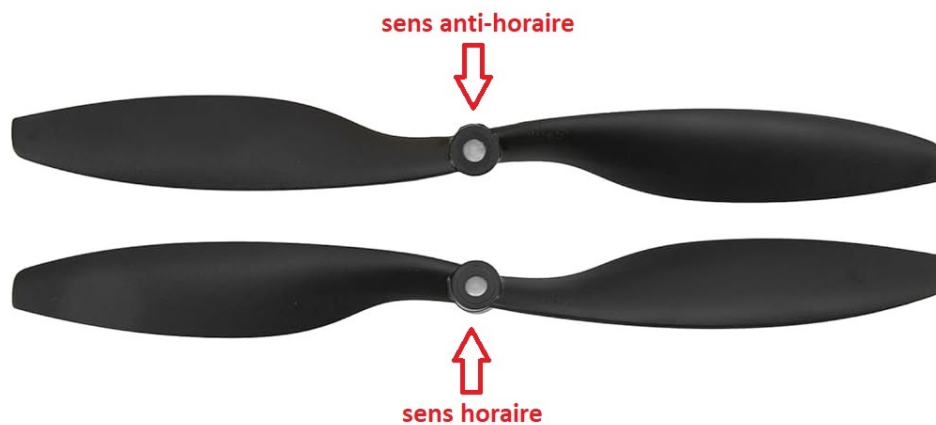


FIGURE 2.4 – Helice 1045R.

Nous avons choisi les hélices 1045R (FIGURE 2.4) pour notre drone, après avoir effectué des tests comparatifs avec d'autres modèles comme les EP-9050, en raison de leurs caractéristiques optimales pour notre configuration spécifique. Les hélices 1045R, avec un diamètre de 10 pouces et un pas de 4,5 pouces, offrent un excellent équilibre entre poussée et efficacité énergétique. Elles permettent une stabilité remarquable et un contrôle précis, essentiels pour les manœuvres complexes et le maintien d'une position stable en vol. De plus, ces hélices sont parfaitement compatibles avec les moteurs que nous utilisons, garantissant une performance fiable et une intégration sans problème dans notre système [12].

2.3.3 Contrôleurs électroniques de vitesse (ESC)

Un contrôleur électronique de vitesse (ESC) (FIGURE 2.5) est un dispositif essentiel pour les drones, utilisé pour réguler la vitesse et le sens des moteurs électriques. Il reçoit des signaux de commande du contrôleur de vol et ajuste la puissance envoyée aux moteurs, permettant ainsi des mouvements précis comme la montée, la descente, la rotation et le déplacement latéral. Les ESC offrent également des fonctionnalités de protection contre les surtensions, les surintensités et les surchauffes, assurant la sécurité et la longévité des composants du drone [10].

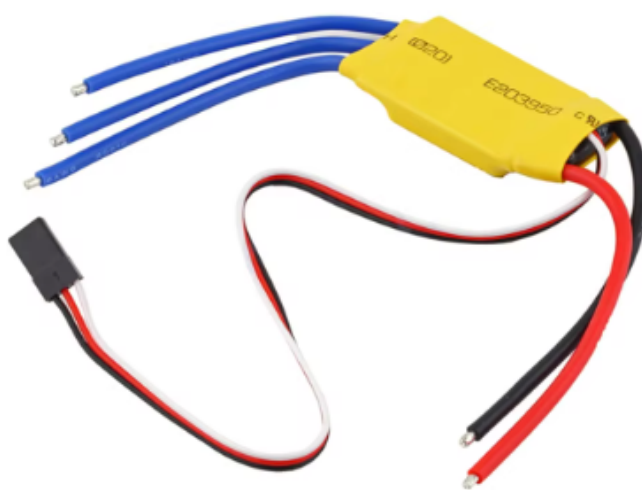


FIGURE 2.5 – ESC 30A [2].

Pour inverser le sens de rotation d'un moteur, il suffit d'inverser deux des fils qui le relie à l'ESC. Il est important que l'ESC puisse supporter le courant absorbé par le moteur. C'est pourquoi nous avons opté pour des ESC de 30A, offrant une marge de sécurité confortable, car nos tests ont montré que les moteurs ne consomment pas plus de 12A chacun.

2.3.4 Batterie LIPO

Une batterie LiPo (Lithium-Polymère) est une batterie rechargeable largement utilisée dans les drones et les appareils électroniques portables. Elle offre une densité énergétique élevée, permettant de stocker plus d'énergie tout en étant légère. Chaque cellule a une tension nominale de 3,7 volts, et les packs peuvent être composés de plusieurs cellules pour obtenir la tension désirée. Les batteries LiPo peuvent fournir des courants de décharge élevés, essentiels pour des applications nécessitant beaucoup de

puissance instantanée. Cependant, elles nécessitent une gestion prudente pour éviter les risques de surcharge, de décharge excessive et de dommages physiques [9].



FIGURE 2.6 – Batterie Lipo 3s 4200mah.

Dans notre cas, nous avons utilisé une batterie LiPo 3S 4200mAh (FIGURE 2.6) car elle fournit une tension de 11,1V, ce qui est largement suffisant pour obtenir une poussée efficace pour notre drone. Nous aurions pu utiliser une batterie 4S 14.8v, mais celle-ci serait plus lourde et la tension supplémentaire ne serait pas nécessaire. Par conséquent, nous avons préféré augmenter la capacité en mAh pour prolonger l'autonomie de la batterie, même si cela ajoute un peu de poids au drone.

Une batterie LiPo 3S est constituée de trois cellules en série. Chacune de ces cellules fournit une tension nominale d'environ 3,7 volts, conférant ainsi à la batterie LiPo 3S une tension globale d'environ 11,1 volts. En ce qui concerne la charge, l'utilisation d'un chargeur compatible est essentielle. Dans notre cas, nous avons choisi le "imaxb3 ac" (FIGURE 2.7) en raison de sa disponibilité sur le marché et de son excellent rapport qualité-prix.



FIGURE 2.7 – Chargeur Lipo 2S/3S.

Remarque : Pour estimer l'autonomie minimale de la batterie, il suffit d'appliquer la loi suivante :

$$T = \frac{C}{I} \quad (2.1)$$

où :

T : Temps de vol en heures.

C : Capacité de la batterie en ampères-heures (Ah).

I : Consommation de courant en ampères (A).

Dans notre étude, nous avons évalué l'autonomie de la batterie pour des moteurs Brushless de 2200 KV et 1000 KV :

— **Pour un moteur de 2200 KV :**

$$T_1 = \frac{4.2Ah}{15A} = 0.28h = 16.8minutes$$

Ce temps de vol pour une seule hélice (T_1) est ensuite divisé par 4 pour obtenir l'autonomie totale, soit : $T_4 = \frac{T_1}{4} = 4.2minutes$

— **Pour un moteur de 1000 KV :**

$$T_1 = \frac{4.2Ah}{8A} = 0.525h = 31.5minutes$$

De même, le temps total est : $T_4 = \frac{T_1}{4} = 7.875minutes$

2.3.5 Distributeur de puissance PDB-XT60

Un distributeur de puissance PDB-XT60 (FIGURE 2.8) est un élément électronique essentiel dans les drones et d'autres applications similaires, permettant de répartir efficacement l'alimentation électrique de la batterie vers divers composants du système, tels que les moteurs, les contrôleurs de vol, les caméras et les émetteurs vidéo. Le terme "PDB" désigne une "Plaque de Distribution de Puissance", tandis que "XT60" fait ré-

férence au type de connecteur souvent utilisé pour se brancher à la batterie. Son rôle est d'optimiser le câblage et d'assurer une alimentation électrique propre et fiable à tous les éléments du drone [25].

Il est à noter que dans certaines configurations, il serait possible de connecter les moteurs directement à la batterie et d'alimenter le microcontrôleur via les ESC, ces derniers fournissant une tension stable. Toutefois, nous avons décidé d'incorporer le distributeur de puissance pour garantir une répartition équitable et sûre du courant entre les quatre moteurs, améliorant ainsi les performances globales du drone.

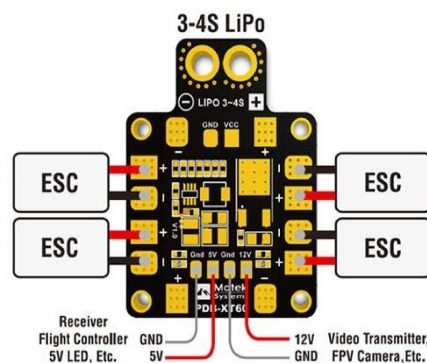


FIGURE 2.8 – PDB-TX60 [25].

2.4 Contrôleur du vol

2.4.1 Microcontrôleur

Un microcontrôleur est une puce électronique intégrant un processeur central, de la mémoire (RAM, ROM, EEPROM), des ports d'entrée/sortie (I/O) et divers périphériques intégrés comme les convertisseurs analogiques-numériques (ADC). Il est conçu pour exécuter des tâches spécifiques dans des systèmes embarqués, tels que les appareils électroniques grand public, les systèmes de contrôle industriels, et les équipements médicaux, grâce à sa faible consommation d'énergie, sa compacité et son coût réduit. Nous avons choisi l'Arduino Nano, qui utilise le microcontrôleur ATmega328, en raison de sa facilité d'utilisation et de sa petite taille. L'ATmega328, également utilisé dans l'Arduino Uno, offre 32 KB de mémoire Flash pour le stockage des programmes, 2 KB de SRAM pour les données temporaires, et 1 KB d'EEPROM pour les données non volatiles. L'Arduino Nano dispose de 14 broches numériques I/O, dont 6 peuvent être

utilisées comme sorties PWM pour le contrôle de moteurs ou l'émission de signaux analogiques. Il offre également 8 entrées analogiques, incluant les ports supplémentaires A6 et A7, permettant une lecture précise des capteurs analogiques (FIGURE 2.9) [24].

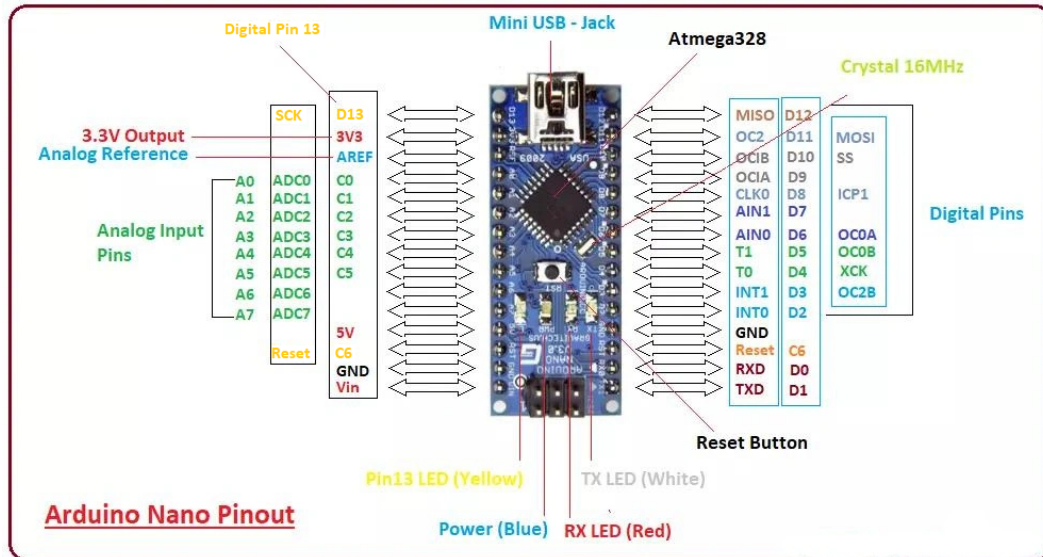


FIGURE 2.9 – Arduino NANO [15].

2.4.2 Capteurs

Accéléromètre-gyroscope

Un capteur accéléromètre-gyroscope est un dispositif électronique combinant un accéléromètre, qui mesure l'accélération linéaire, et un gyroscope, qui détecte l'orientation angulaire ou la vitesse de rotation. Cette combinaison permet de fournir des données précises sur les mouvements tridimensionnels d'un objet ou d'un système. Ces capteurs sont largement utilisés dans des domaines tels que la navigation, le contrôle d'orientation, la réalité virtuelle et la stabilisation d'images. Nous avons choisi le MPU6050 (FIGURE 2.10) pour ses qualités techniques et sa simplicité d'utilisation. Ce capteur à six axes intègre un accéléromètre trois axes et un gyroscope trois axes, avec une plage de mesure allant de $\pm 2g$ à $\pm 16g$ pour l'accéléromètre et de $\pm 250^\circ/s$ à $\pm 2000^\circ/s$ pour le gyroscope. Le MPU6050 communique via l'interface I2C, permettant une connexion facile avec des microcontrôleurs comme l'Arduino Nano. Il fonctionne avec une tension d'alimentation de 2.375V à 3.46V et inclut un capteur de température intégré pour des mesures précises et une compensation des dérives thermiques [14].

La capacité du MPU6050 à fusionner les données des deux capteurs permet de fournir des informations précises sur l'orientation et le mouvement, ce qui est crucial pour des applications comme la stabilisation de drones, les systèmes de réalité virtuelle, et les

robots. Son coût abordable et sa large disponibilité sur le marché en font un choix populaire pour les projets nécessitant un suivi précis des mouvements. De plus, la grande communauté d'utilisateurs de ce capteur offre un support précieux, avec de nombreuses ressources en ligne pour faciliter son intégration et son utilisation.

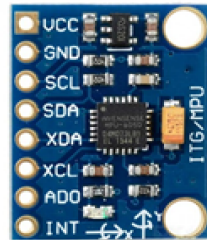


FIGURE 2.10 – MPU6050 [28].

Baromètre

Un capteur barométrique est un dispositif électronique utilisé pour mesurer la pression atmosphérique. Cette mesure permet de déterminer des informations telles que l'altitude, les variations météorologiques et les conditions locales. Les capteurs barométriques sont couramment utilisés dans des domaines variés, notamment la météorologie, la navigation, l'aviation, les drones et les appareils mobiles. Le GY-63 (FIGURE 2.11), qui intègre le capteur MS5611, est un exemple de capteur barométrique précis. Il offre une excellente résolution et fiabilité, avec une faible consommation d'énergie. Grâce à sa capacité à mesurer avec précision la pression et la température, le GY-63 est utilisé dans les drones, les systèmes de navigation et les instruments météorologiques, offrant une grande stabilité et robustesse dans diverses applications [27].

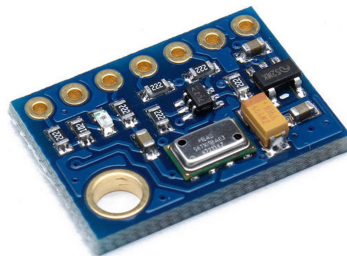


FIGURE 2.11 – Capteur barométrique GY 63 [3].

Magnétomètre

Un capteur magnétomètre mesure le champ magnétique ambiant généré par des sources telles que la Terre, les aimants permanents ou les courants électriques. Il détecte la direction et l'intensité du champ dans les trois dimensions, utilisé dans diverses applications comme la navigation, la détection de métaux et la réalité augmentée. Le HMC5883L (FIGURE 2.12), un modèle à trois axes de Honeywell, est connu pour sa haute précision, sa faible consommation d'énergie et sa compacité, intégré dans des dispositifs comme les smartphones, les drones et les systèmes de navigation [20].

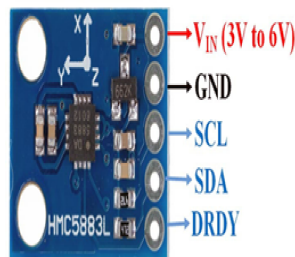


FIGURE 2.12 – HMC5883L [22].

2.5 Commande à distance

2.5.1 Radiocommande

Une radiocommande, ou commande radio, pour un drone est un dispositif électronique qui permet de contrôler le drone à distance en utilisant des signaux radio. Elle est composée d'un émetteur (tenu par l'utilisateur) et d'un récepteur (installé sur le drone). L'émetteur envoie des commandes au récepteur pour piloter différentes fonctions du drone, telles que le vol, l'altitude, la direction et les mouvements spécifiques (comme les flips ou les rotations). La radiocommande fonctionne généralement sur des fréquences spécifiques pour assurer une communication stable et sans interférence avec le drone, offrant ainsi un contrôle précis et réactif [29].

Émetteur

Nous avons opté pour la radiocommande FlySky FS-i6X (FIGURE 2.13) en raison de ses nombreux avantages. Ce modèle est particulièrement apprécié pour sa fiabilité et sa portée étendue, assurées par son système de transmission en 2,4 GHz, qui minimise les risques d'interférences. La FS-i6X se distingue par ses 10 canaux programmables, permettant une personnalisation poussée des contrôles pour les différentes fonctions du drone, ce qui est essentiel pour des applications complexes.

En termes d'ergonomie, la FS-i6X est équipée d'un écran LCD rétroéclairé de 2,8 pouces, offrant une interface claire et intuitive pour la navigation dans les menus et les réglages. Sa conception légère, combinée à une forme ergonomique, assure un confort de pilotage optimal, même lors de longues sessions. La radiocommande est également dotée de la technologie AFHDS 2A (Automatic Frequency Hopping Digital System), qui améliore la sécurité et la stabilité de la connexion en utilisant des fréquences multiples pour éviter les interférences.

Enfin, son rapport qualité-prix exceptionnel la rend attrayante tant pour les amateurs que pour les professionnels, offrant des caractéristiques avancées à un coût abordable [7].



FIGURE 2.13 – FlySky FS i6X [29].

Récepteur

Nous avons choisi le récepteur FS-IA6B (FIGURE 2.14) pour ses nombreux avantages. Il offre une excellente portée de signal et fonctionne sur la fréquence de 2,4 GHz, réduisant les interférences. Avec six canaux, il permet un contrôle flexible des fonctions du drone et est compatible avec de nombreux émetteurs Flysky. Ses capacités de télémétrie fournissent des informations en temps réel, comme la tension de la batterie et la qualité du signal. Compact et léger, il est facile à installer et fiable en vol, ce qui en fait un choix idéal pour les pilotes de tous niveaux [29].

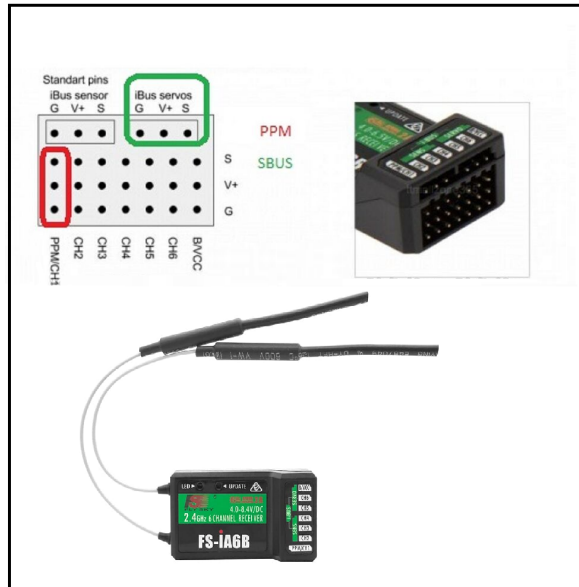


FIGURE 2.14 – Récepteur FlySky FS-IA6B [16].

2.5.2 Commande Bluetooth

Module Bluetooth HC05

Le module Bluetooth HC-05 (FIGURE 2.15) est un dispositif de communication sans fil qui permet d'établir une connexion Bluetooth entre différents appareils. Il est couramment utilisé pour des projets de prototypage et d'intégration avec des microcontrôleurs comme Arduino et Raspberry Pi. Ce module prend en charge la communication série (UART), facilitant l'échange de données entre appareils à courte distance. Le HC-05 est apprécié pour sa simplicité d'utilisation, sa faible consommation d'énergie et sa capacité à passer en mode maître ou esclave, offrant ainsi une grande flexibilité pour diverses applications de communication sans fil [16].

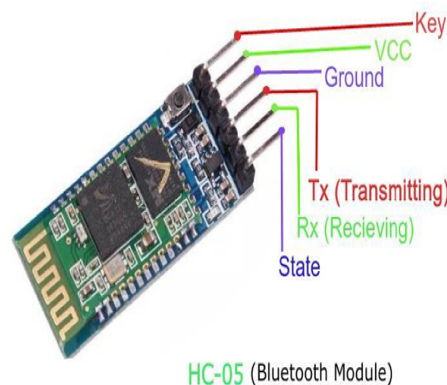


FIGURE 2.15 – Module Bluetooth HC-05 [33].

Dans notre projet, nous avons utilisé le module Bluetooth HC-05 dans différentes applications. Initialement, nous l'avons intégré avec une application Android que nous avons créée à l'aide du logiciel App Inventor (FIGURE 2.16). Cette application permet de piloter le drone à courte distance en envoyant un signal PWM via le module Bluetooth et d'afficher la caméra en temps réel sur l'écran du téléphone.

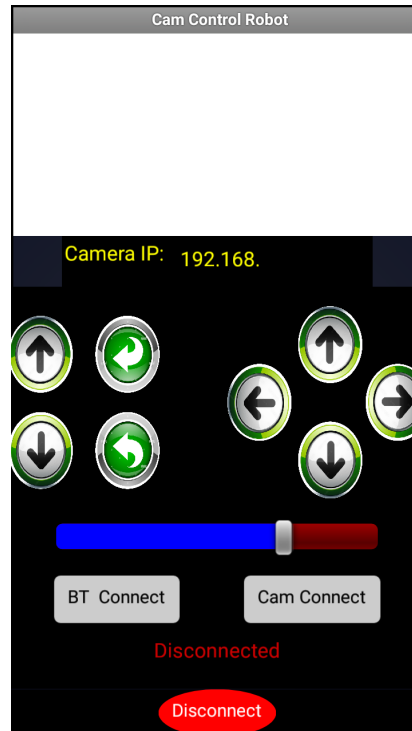


FIGURE 2.16 – Application Android App Inventor.

Ensuite, nous avons intégré la radiocommande avec l'interface MultiWii pour un contrôle plus avancé. Pour modifier les paramètres PID et surveiller notre drone, nous avons utilisé l'application EZ-GUI (FIGURE 2.17) en conjonction avec le module Bluetooth, ce qui nous a permis d'ajuster facilement les réglages et de maintenir une communication stable avec le drone.

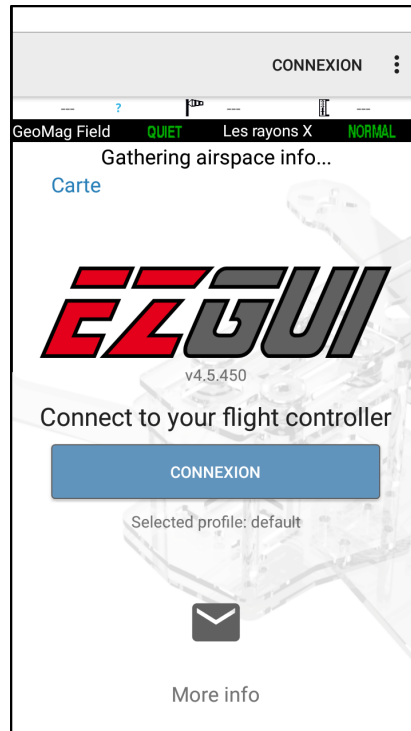


FIGURE 2.17 – Application EZ-GUI.

2.6 Conclusion

Ce chapitre a détaillé les éléments constitutifs fondamentaux d'un quadrirotor, en couvrant le châssis, le système de propulsion, le contrôleur de vol, et les systèmes de commande à distance. L'analyse approfondie de chaque composant a permis de comprendre leur rôle spécifique dans le fonctionnement global du drone et leur interaction pour assurer stabilité et performance en vol. La maîtrise de ces éléments est cruciale pour la phase suivante, qui se concentrera sur la mise en œuvre pratique et la réalisation concrète du quadrirotor. Le prochain chapitre détaillera les étapes de l'assemblage, des tests, et de l'optimisation, mettant en application les connaissances acquises pour développer un drone pleinement opérationnel.

Chapitre 3

Mise en œuvre et réalisation du quadricoptère

3.1 Introduction

Dans ce chapitre, nous présenterons les résultats obtenus au cours de notre travail. Nous détaillerons les deux méthodes de programmation que nous avons mises en œuvre, ainsi que les essais réalisés avant d'atteindre des résultats satisfaisants. Nous illustrerons le comportement en temps réel de notre quadrirotor en appliquant un correcteur de type PID, puis nous analyserons les effets de l'ajout d'un filtre passe-bas à l'accéléromètre-gyroscopie. Enfin, nous discuterons en profondeur ces différents résultats.

3.2 Première méthode de programmation

Notre stratégie initiale consistait à maintenir la position horizontale de notre quadrirotor d'abord selon l'axe X, puis selon l'axe Y, et enfin selon les deux axes simultanément. Cette approche visait à simplifier la programmation.

Nous avons d'abord commencé par fabriquer un châssis en forme de croix ("+"). Deux moteurs brushless ont été placés aux extrémités opposées du châssis, tandis que les deux autres extrémités ont été montées de manière à pouvoir tourner librement autour de l'axe X (FIGURE 3.1). Les ESC (contrôleurs de vitesse électronique) ont ensuite été montés sur le châssis et reliés aux moteurs. L'Arduino et l'accéléromètre-gyroscopie ont été placés au centre du châssis. Nous avons ensuite téléversé le code Arduino d'un birotor et configuré les paramètres PID sur leurs valeurs par défaut, que nous avons ajustés progressivement jusqu'à obtenir les meilleurs résultats. Cette procédure a été répétée pour l'axe Y, puis pour les deux axes simultanément.

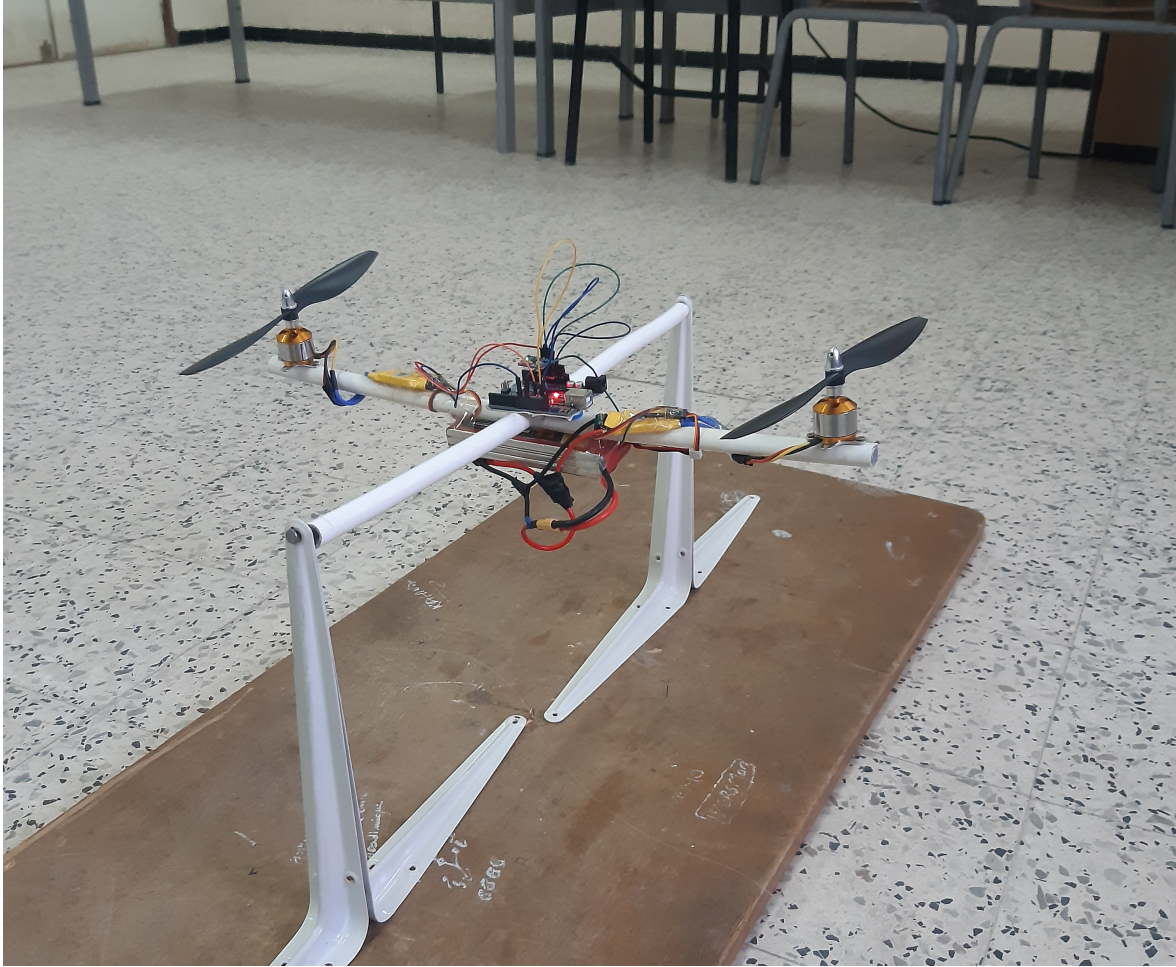


FIGURE 3.1 – Bicopteur.

Le code Arduino développé pour le bicoptère (voir l'annexe A), utilise le capteur accéléromètre-gyroscopie pour mesurer en temps réel l'angle du bicoptère. Cette mesure est ensuite comparée à une consigne d'angle, fixée à 0 degré pour maintenir l'horizontalité. Le contrôleur PID, paramétré avec des gains proportionnel (K_p), intégral (K_i), et dérivé (K_d), calcule la correction nécessaire afin de réduire au maximum l'écart entre l'angle mesuré et la consigne. La sortie du PID est alors utilisée pour ajuster les vitesses des deux moteurs (M1 et M2), garantissant ainsi la stabilisation du bicoptère à l'angle souhaité. Ce système de contrôle en boucle fermée assure la stabilité du bicoptère, comme récapitulé dans la figure suivante (FIGURE 3.2).

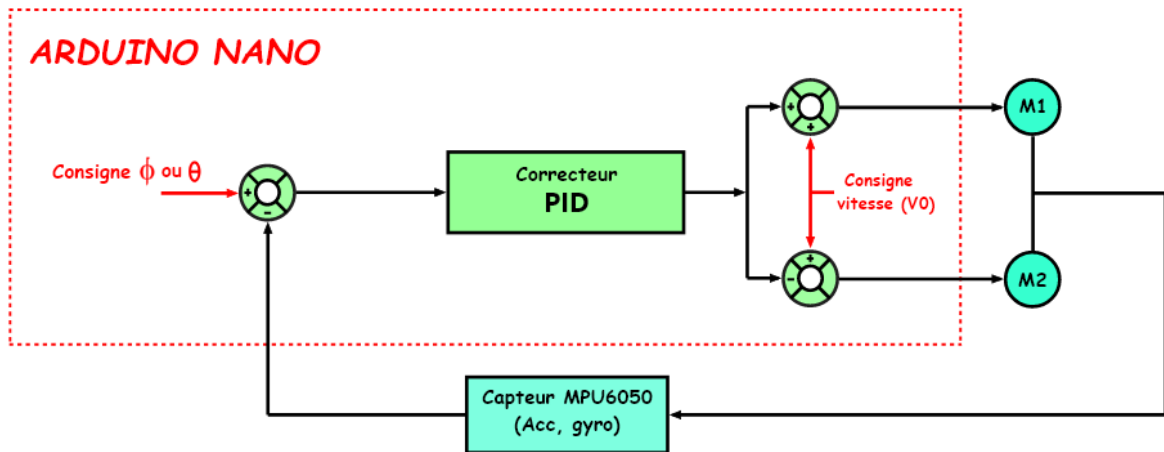


FIGURE 3.2 – PID deux moteurs.

Cependant, cette méthode s'est révélée non seulement laborieuse, mais elle comportait également de nombreuses insuffisances et lacunes.

C'est pour ces raisons que nous avons choisi de changer de stratégie en optant pour le système MultiWii, qui s'est avéré plus efficace et mieux adapté à nos besoins.

3.3 Deuxième méthode de programmation (avec MULTIWI)

En raison de la complexité et du temps considérable requis par notre méthode de programmation initiale, nous avons décidé de changer de stratégie. Nous avons opté pour une approche qui nous permettrait de développer notre quadricoptère plus rapidement et avec une plus grande robustesse. Cette nouvelle méthode utilise une interface graphique MultiWii, nous offrant la possibilité d'intégrer plusieurs types de capteurs. Grâce à cette interface, nous avons pu simplifier le processus de développement tout en améliorant la performance et la stabilité de notre quadricoptère.

Pour piloter notre drone, nous avons besoin d'une télécommande. Plutôt que d'en acheter une, nous avons choisi de la fabriquer nous-mêmes. La solution la plus économique a été de créer une application Android pour piloter notre drone via Bluetooth. Nous avons donc opté pour App Inventor pour développer cette application.

3.3.1 Application Android de pilotage de quadrotère réalisée avec App Inventor

App Inventor est une plateforme en ligne de développement visuel créée par Google, désormais maintenue par le Massachusetts Institute of Technology (MIT), qui permet de concevoir des applications mobiles pour Android. Grâce à une interface graphique, les utilisateurs peuvent assembler des blocs de code visuel par glisser-déposer pour définir le comportement de leurs applications. Cette plateforme, destinée aux appareils Android, bénéficie d'une vaste communauté et de nombreuses ressources éducatives [26].

Dans notre projet, nous avons utilisé une application existante de (Muhammad Ansar) pour piloter des robots. Nous l'avons entièrement modifiée et améliorée, tant au niveau du design (FIGURE 3.3) que du code (FIGURE 3.4), afin de l'adapter à notre quadrirotor. Les modifications apportées incluent des améliorations de l'interface utilisateur et des optimisations du code pour mieux répondre aux besoins spécifiques de notre projet.

Ce code App Inventor (FIGURE 3.4) permet de contrôler un quadrirotor via Bluetooth en utilisant une application mobile. Les différents blocs de code gèrent les événements de connexion Bluetooth, les commandes de mouvement, les interactions avec la caméra, et d'autres fonctionnalités. Lors de la sélection d'un périphérique Bluetooth, le code récupère les noms et adresses des périphériques disponibles et tente de se connecter à celui sélectionné. Un timer vérifie constamment l'état de la connexion Bluetooth, affichant "Connected" en vert si la connexion est réussie, ou "Disconnected" en rouge dans le cas contraire. Les commandes de mouvement sont gérées par des boutons qui envoient des valeurs spécifiques via Bluetooth lorsqu'ils sont pressés et relâchés, permettant ainsi de contrôler les directions du quadrirotor. Un curseur de glissière envoie la position actuelle arrondie via Bluetooth pour des contrôles plus fins. En ce qui concerne la caméra, un clic sur le bouton de la caméra ouvre une URL dans un visualiseur web "WebView1" en combinant une URL de base avec une adresse IP ou un lien saisi par l'utilisateur. Enfin, un bouton de déconnexion permet de couper la connexion Bluetooth et d'afficher un message de déconnexion.

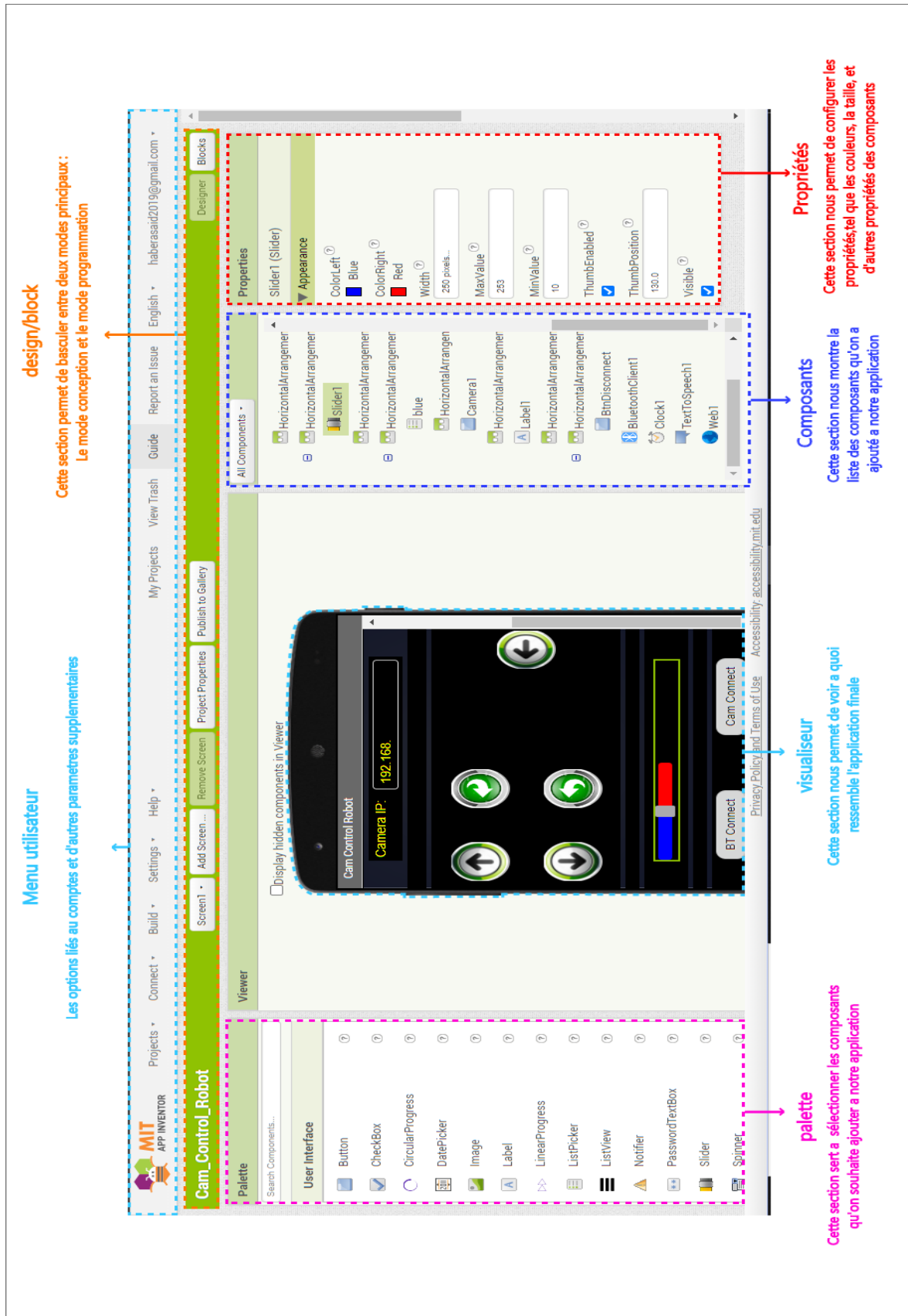
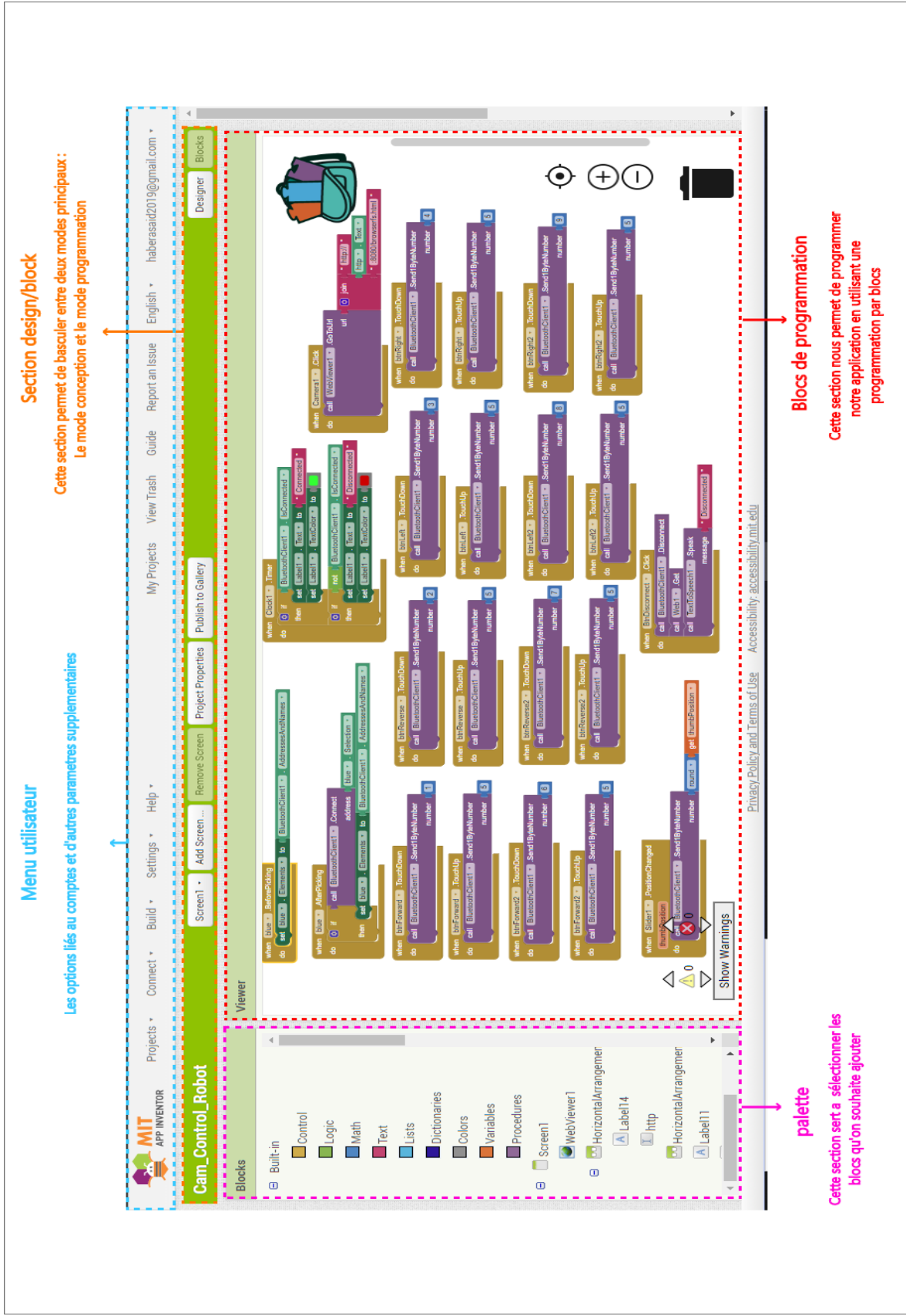


FIGURE 3.3 – Design de l'application avec App Inventor.



Section design/block

Cette section permet de basculer entre deux modes principaux:
Le mode conception et le mode programmation

Menu utilisateur

Les options liés au comptes et d'autres paramètres supplémentaires

Blocs de programmation

Cette section nous permet de programmer
notre application en utilisant une
programmation par blocs

palette

Cette section sert à sélectionner les
blocs qu'on souhaite ajouter

FIGURE 3.4 – Code de l'application avec App Inventor.

Remarque : En raison de la réponse lente du module Bluetooth, nous avons préféré effectuer seulement quelques tests simples avec cette application modifiée. Ces tests nous ont permis de vérifier les fonctionnalités de base et d'assurer que les commandes étaient correctement transmises. Après avoir constaté les limitations du Bluetooth en termes de latence, nous avons décidé de passer à l'utilisation d'une radiocommande, qui offre une réponse plus rapide et plus fiable, essentielle pour le contrôle précis et sophistiqué de notre quadricoptère.

3.3.2 Conception de la radio commande pour le quadricoptère

Après avoir abandonné l'idée d'utiliser l'application AppInventor pour contrôler notre quadricoptère, nous avons décidé de nous orienter vers une solution de radio commande. Cependant, en raison du coût relativement élevé des systèmes de radio commande commerciaux, nous avons choisi de concevoir notre propre radio commande (FIGURE 3.5). Pour ce faire, nous avons utilisé les composants suivants :

- **Breadboard :** Une breadboard, ou plaque de prototypage, permet de construire des circuits électroniques sans soudure. Elle facilite les tests et les expérimentations rapides de différentes configurations de circuits sans besoin de soudures permanentes. Cela rend le processus de prototypage plus flexible et rapide.
- **Deux modules radio HC-12 :** Ces modules permettent une communication série sans fil entre deux microcontrôleurs. Le HC-12 fonctionne dans la bande de 433 MHz et peut transmettre des données sur une distance allant jusqu'à 1 km en extérieur avec une ligne de vue directe. Un module est utilisé pour la transmission et l'autre pour la réception, assurant une communication bidirectionnelle fiable.
- **Deux joysticks :** Les joysticks sont des dispositifs d'entrée permettant de contrôler les mouvements du quadricoptère. Un joystick est dédié au contrôle du pitch (inclinaison avant/arrière) et du roll (inclinaison latérale), tandis que l'autre contrôle le yaw (rotation autour de l'axe vertical) et le throttle (puissance des moteurs). Ces joysticks convertissent les mouvements physiques en signaux électriques précis, essentiels pour un contrôle précis du quadricoptère.
- **Une carte Arduino UNO :** L'Arduino UNO est une carte électronique basée sur le microcontrôleur ATmega328P. Elle comprend 14 broches d'entrée/sortie numériques (dont 6 peuvent être utilisées comme sorties PWM), 6 entrées analogiques, un oscillateur à quartz de 16 MHz, une connexion USB, un jack d'alimentation, un header ICSP et un bouton de réinitialisation. L'Arduino UNO est largement utilisé en raison de sa simplicité d'utilisation et de sa flexibilité pour divers projets électroniques.

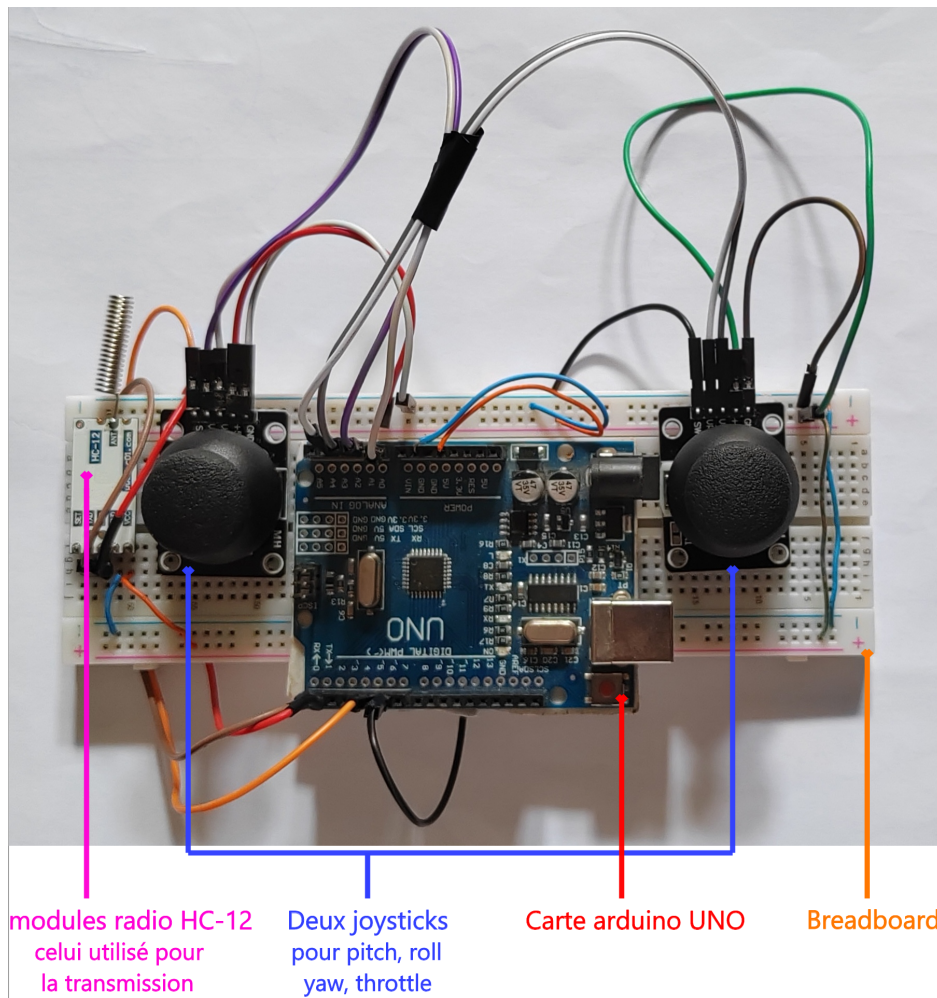


FIGURE 3.5 – Conception de la Radio Commande.

Nous avons programmé notre carte Arduino et effectué des tests avec l'interface MultiWii. Malheureusement, nous avons rencontré des problèmes de perturbation des signaux, principalement dus à la sensibilité excessive des joysticks. Bien que cette configuration puisse être suffisante pour contrôler un robot nécessitant une sensibilité moyenne, elle s'est avérée inadaptée pour un quadricoptère, où une précision et une stabilité extrêmes sont nécessaires pour maintenir un vol stable et contrôlé.

Face à ces limitations, nous avons décidé d'utiliser la radio commande FlySky i6X, une solution plus sophistiquée et adaptée. La FlySky i6X est une télécommande programmable offrant une portée allant jusqu'à 500 mètres en conditions optimales, des options de personnalisation pour différents modes de vol, et une interface utilisateur intuitive. Son adoption a considérablement amélioré la stabilité et la maniabilité de notre quadricoptère, permettant un contrôle précis et

fiable des mouvements en vol.

Grâce à cette radio commande, nous avons pu surmonter les défis techniques posés par notre solution initiale et garantir une meilleure performance de notre quadricoptère, tant en termes de stabilité que de réactivité des commandes.

3.3.3 Définition de la plateforme MULTIWII

MultiWii est une plateforme de contrôle de vol open-source, offrant une grande flexibilité et permettant aux développeurs et amateurs de l'adapter à leurs besoins spécifiques. Elle prend en charge une large gamme de capteurs, tels que les accéléromètres, gyroscopes, magnétomètres, baromètres et GPS, ce qui permet une stabilisation et une navigation précises. Grâce à son interface graphique conviviale (MultiWii GUI) (FIGURE 3.6), la configuration, le réglage et la surveillance des paramètres de vol sont facilités en temps réel. De plus, la modularité de MultiWii permet son utilisation pour différents types de drones et configurations de vol. Enfin, une communauté active de développeurs et d'utilisateurs enrichit continuellement la plateforme en partageant des améliorations et des conseils, faisant de MultiWii un outil précieux pour personnaliser et optimiser les véhicules aériens [11].

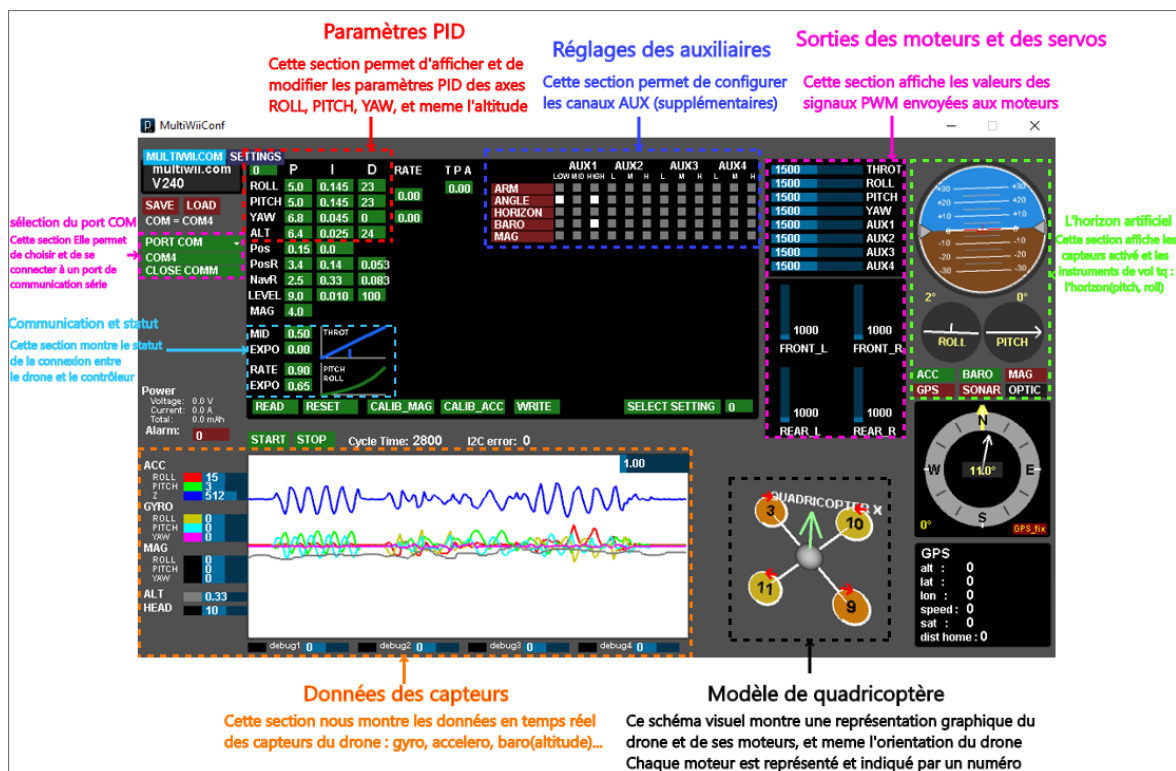


FIGURE 3.6 – Interface graphique (MultiWii GUI).

3.3.4 Procédure générale de configuration pour un quadrirotor utilisant Arduino Nano et MultiWii

La configuration d'un quadrirotor utilisant la plateforme MultiWii commence par la préparation du matériel nécessaire, comprenant une carte de contrôle Arduino Nano, des capteurs comme le MPU6050 pour le gyroscope et l'accéléromètre, le HMC5883L pour le magnétomètre, un module Bluetooth HC-05, et un baromètre GY63, en plus des moteurs, ESC, châssis, hélices, batterie et émetteur-récepteur radio. Ensuite, le firmware MultiWii est téléchargé, configuré en éditant le fichier config.h pour activer les capteurs utilisés, puis téléversé sur l'Arduino Nano via l'IDE Arduino. Une fois le firmware installé, les capteurs sont connectés à la carte et calibrés à l'aide du logiciel MultiWii Configurator. Les ESC sont ensuite connectés et calibrés, suivis du montage des hélices sur les moteurs en respectant leur sens de rotation. La communication via Bluetooth est établie en associant le module HC-05, et, si utilisé, le récepteur radio est connecté et configuré. Des tests initiaux au sol sont effectués pour vérifier la réponse des moteurs, suivis d'ajustements des paramètres PID pour optimiser la stabilité et les performances de vol. Des vérifications régulières sont effectuées pour garantir des connexions sécurisées et le bon état des composants, assurant ainsi un fonctionnement sûr et efficace du quadrirotor [11].

3.3.5 Montage

Nous avons réalisé un quadrirotor en utilisant un Arduino Nano comme contrôleur principal, connecté à un module MPU-6050 pour la mesure de l'accélération et de la rotation. Le montage comprend quatre moteurs brushless, chacun contrôlé par un régulateur de vitesse électronique (ESC) connecté aux sorties PWM de l'Arduino. Les moteurs sont montés sur un cadre de drone robuste et équilibré de forme "X" DJI F450, équipé d'hélices 1045R. L'alimentation est fournie par une batterie LiPo 11.1V, connectée à un distributeur de puissance PDB-XT60, qui alimente l'Arduino et les ESCs. Les données du MPU-6050 sont lues via les broches SDA et SCL (A4 et A5 de l'Arduino), permettant ainsi une stabilisation précise du drone en temps réel grâce à un code de contrôle personnalisé. Ce montage permet de contrôler la vitesse et la direction de chaque moteur, assurant ainsi une stabilité et une maniabilité optimales du quadrirotor.

Les composants supplémentaires incluent un capteur GY63 pour la mesure de la pression barométrique et un capteur HMC5883L pour la mesure du champ magnétique, tous deux montés de la même manière que le MPU-6050.

Les composants sont placés et branchés comme suit :

- **Arduino Nano** : Fixé sur le cadre central du drone. Le Vin est connecté au +11.1V du PDB-XT60 et le GND au GND du PDB.
- **Module MPU-6050** : Monté près de l'Arduino Nano. Le VCC est connecté au 5V de l'Arduino, le GND au GND de l'Arduino, le SDA à la broche A4 de l'Arduino et le SCL à la broche A5 de l'Arduino.
- **Capteur GY63** : Monté près de l'Arduino Nano. Le VCC est connecté au 3.3V de l'Arduino, le GND au GND de l'Arduino, le SDA à la broche A4 de l'Arduino et le SCL à la broche A5 de l'Arduino.
- **Capteur HMC5883L** : Monté près de l'Arduino Nano. Le VCC est connecté au 3.3V de l'Arduino, le GND au GND de l'Arduino, le SDA à la broche A4 de l'Arduino et le SCL à la broche A5 de l'Arduino.
- **Régulateurs de vitesse électroniques (ESC)** : Fixés sur les bras du cadre du drone, proches des moteurs correspondants. Les fils de signal des ESCs sont connectés aux broches D3, D9, D10, et D11 de l'Arduino Nano. Les fils d'alimentation des ESCs sont connectés aux bornes correspondantes du PDB-XT60.
- **Moteurs brushless** : Montés aux extrémités des bras du cadre du drone. Les trois fils des moteurs brushless sont connectés aux fils de puissance correspondants des ESCs.
- **Distributeur de puissance PDB-XT60** : Fixé au centre du cadre du drone. Le +11.1V de la batterie LiPo est connecté à l'entrée du PDB, qui distribue ensuite l'alimentation à l'Arduino Nano et aux ESCs.
- **Batterie LiPo 11.1V** : Fixée au centre en dessous du cadre du drone pour un meilleur équilibre. Le +11.1V est connecté à l'entrée du PDB-XT60, et le GND est connecté au GND du PDB.
- **Récepteur FlySky FS-IA6B** : Monté sur le cadre central du drone, à proximité de l'Arduino Nano. Le canal de Throttle est connecté à la broche D2 de l'Arduino, Roll à la broche D4, Pitch à D5, Yaw à D6, Aux1 à D7, et Aux2 à D8.
- **Module Bluetooth HC-05** : Monté près de l'Arduino Nano. Le VCC est connecté au 5V de l'Arduino, le GND au GND de l'Arduino, le TX à la broche RX (D0) de l'Arduino et le RX à la broche TX (D1) de l'Arduino.

Après avoir détaillé les branchements de chaque composant, le schéma de montage (FIGURE 3.7) suivant illustre l'ensemble des connexions effectuées sur le drone :

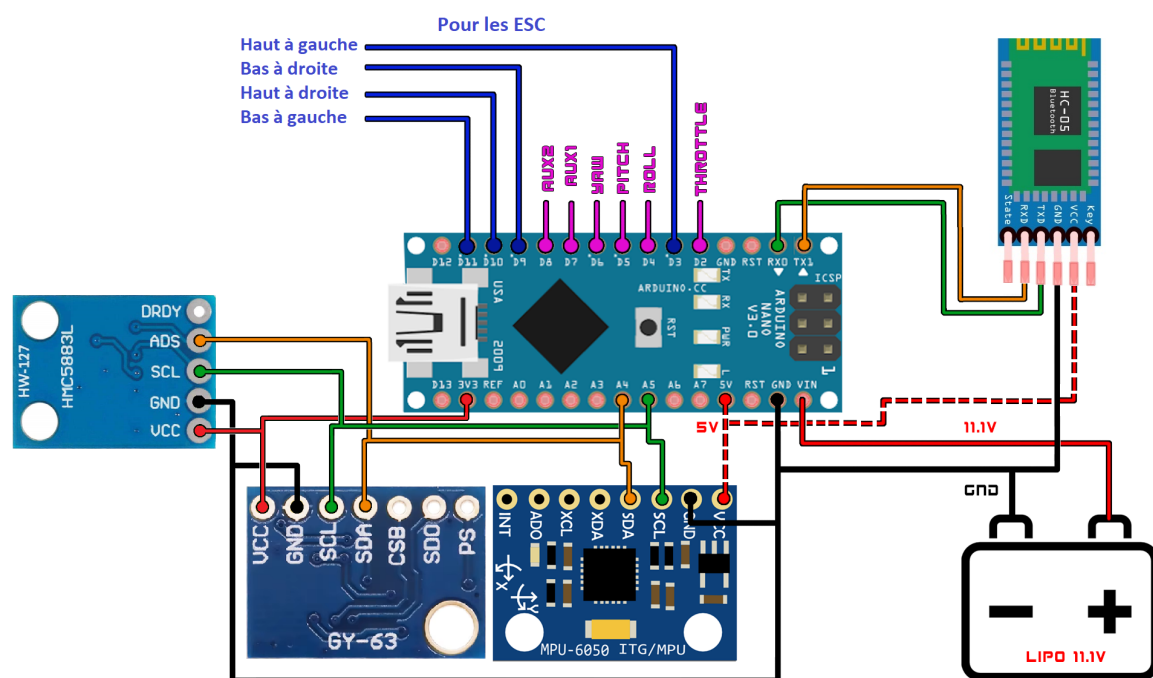


FIGURE 3.7 – Schéma de montage.

3.3.6 Calibrage des ESC

Le calibrage des ESC (Electronic Speed Controllers) est une étape cruciale pour assurer le bon fonctionnement de notre quadrirotor. Pour commencer, nous retirons les hélices des moteurs pour des raisons de sécurité. Ensuite, nous connectons l'Arduino Nano à notre ordinateur et ouvrons l'IDE Arduino. Nous chargeons un programme Arduino spécialement conçu pour donner des instructions aux ESC afin que les moteurs tournent avec la vitesse max pendant 10 secondes puis s'arrêtent. Nous téléversons ce programme sur l'Arduino.

Nous alimentons ensuite le quadrirotor et connectons chaque ESC à la carte de contrôle. Les moteurs émettent un bip indiquant que les ESC ont enregistré le signal de la position maximale. Après cela, les moteurs tournent pendant 10 secondes puis s'arrêtent, confirmant ainsi la calibration [1].

3.3.7 Calibrage des capteurs

Le calibrage des capteurs utilisés sur notre quadrirotor est essentiel pour garantir des performances optimales et une stabilité en vol. Pour commencer, nous connectons l'Arduino Nano à notre ordinateur et ouvrons l'IDE Arduino. Nous chargeons le firmware MultiWii configuré pour notre matériel, incluant les capteurs MPU6050 pour le

gyroscope et l'accéléromètre, le HMC5883L pour le magnétomètre, et le GY63 pour le baromètre, puis téléversons le firmware sur l'Arduino. Une fois le firmware installé, nous utilisons le logiciel MultiWii Configurator pour effectuer le calibrage des capteurs. Pour calibrer le gyroscope, nous plaçons le quadrirotor sur une surface plane et stable, puis lançons la commande de calibration dans le logiciel. Ensuite, nous calibrons l'accéléromètre en suivant les instructions à l'écran, ce qui implique de positionner le quadrirotor dans plusieurs orientations spécifiques. Pour le magnétomètre, nous effectuons une calibration en rotation, en faisant pivoter le quadrirotor lentement autour de ses axes pour capturer les données magnétiques. Enfin, nous vérifions les lectures du baromètre et ajustons si nécessaire. Une fois le calibrage terminé, nous effectuons des tests de vérification pour s'assurer que tous les capteurs fonctionnent correctement.

Remarque : Pour accroître la flexibilité et le contrôle, nous avons intégré un module Bluetooth HC-05. Cela nous permet de régler les paramètres PID à distance et de surveiller en temps réel l'état des moteurs, les graphiques des angles et de l'attitude, ainsi que divers autres paramètres via l'application EZ-GUI.

3.3.8 Ajustement des paramètres PID

Nous avons téléversé les paramètres PID par défaut sur notre carte et activé uniquement le MPU6050 pour assurer son bon fonctionnement. Lors des premiers tests avec ces paramètres, nous avons remarqué que le drone oscillait considérablement dans toutes les directions. Ce comportement était attribué à une réactivité insuffisante pour corriger rapidement les erreurs. Pour résoudre ce problème, nous avons progressivement augmenté l'action proportionnelle pour le tangage et le roulis jusqu'à ce que les oscillations diminuent. Cependant, en augmentant davantage cette valeur, le drone a commencé à montrer des signes de déstabilisation. Nous avons alors déterminé la valeur maximale admissible pour l'action proportionnelle et l'avons légèrement réduite pour restaurer la stabilité [11][1].

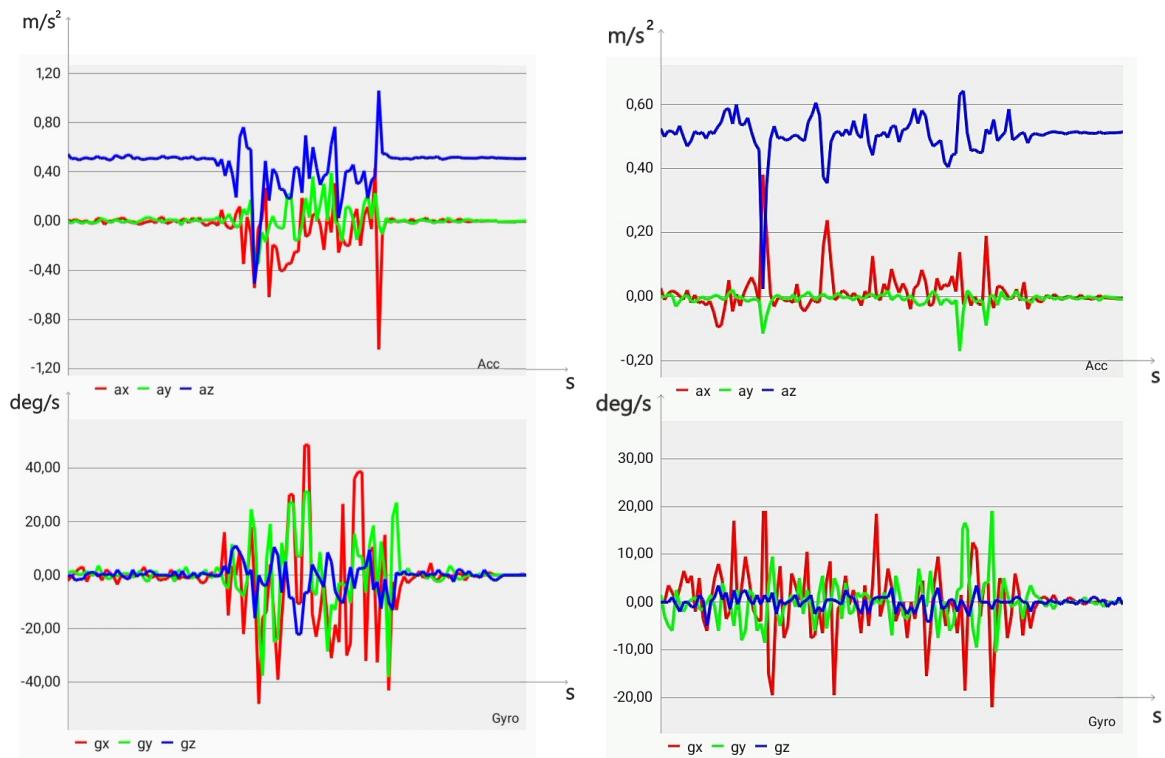
Ensuite, nous avons ajusté les paramètres intégrale pour le tangage et le roulis afin de maintenir le drone parfaitement horizontal, ce qui a permis de corriger les disparités de poids. Après avoir stabilisé horizontalement notre quadrirotor, nous avons intégré le capteur barométrique GY63 pour assurer une stabilisation verticale efficace [11][1]. Les paramètres PID optimaux sont donnés dans le tableau (3.1) :

Les graphes suivants illustrent en temps réel les rotations et les translations prélevées avec l'application EZ-GUI, en comparant le comportement du drone avec les paramètres PID par défaut et après l'intégration des meilleurs paramètres PID opti-

Angle \ Action	P	I	D
ROLL	5.0	0.145	23
PITCH	5.0	0.145	23
YAW	6.8	0.045	0

TABLE 3.1 – Paramètres PID optimaux.

misés. Ces courbes permettent de visualiser l'impact direct de ces paramètres sur le comportement du drone.



(a) Vol avec PID par défaut.

(b) Vol avec paramètres PID optimisés.

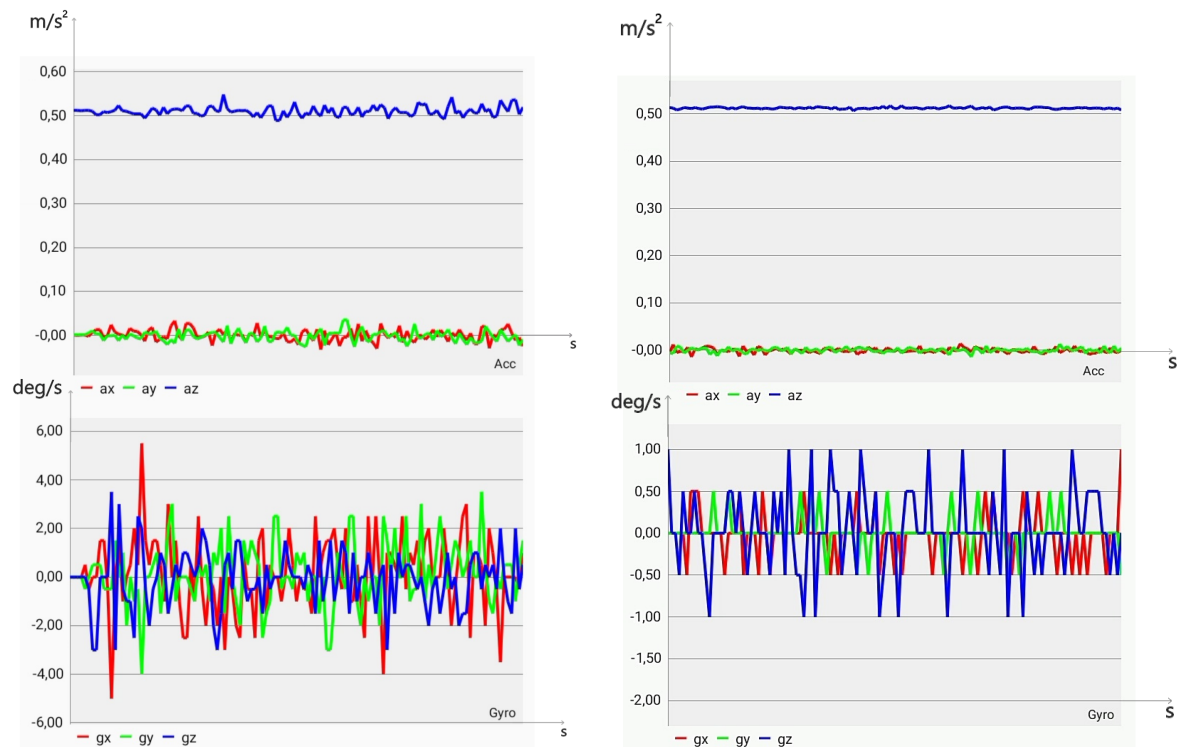
FIGURE 3.8 – Impact des paramètres PID optimisés sur le comportement du drone en vol.

Lors de nos essais avec ces meilleurs paramètres PID, nous avons observé que bien que la correction apportée par le PID soit efficace, elle ne parvenait pas à assurer la stabilité du drone. En approfondissant notre analyse, nous avons remarqué que les perturbations étaient particulièrement prononcées lors de l'armement du drone, ce qui a soulevé des doutes quant à l'intégrité des données fournies par l'accéléromètre-gyroscopie.

Nous avons alors entrepris de résoudre ce problème en intégrant un filtre passe-bas avec

une bande passante de 20 Hz, appliqué directement aux données de l'accéléromètre-gyroscope. Ce filtrage a permis de réduire de manière significative les bruits et les perturbations parasites, laissant passer uniquement les signaux pertinents pour le contrôle du drone. L'effet de ce filtre a été immédiatement perceptible, la stabilité du drone s'est nettement améliorée, rendant son comportement en vol beaucoup plus prévisible et contrôlable.

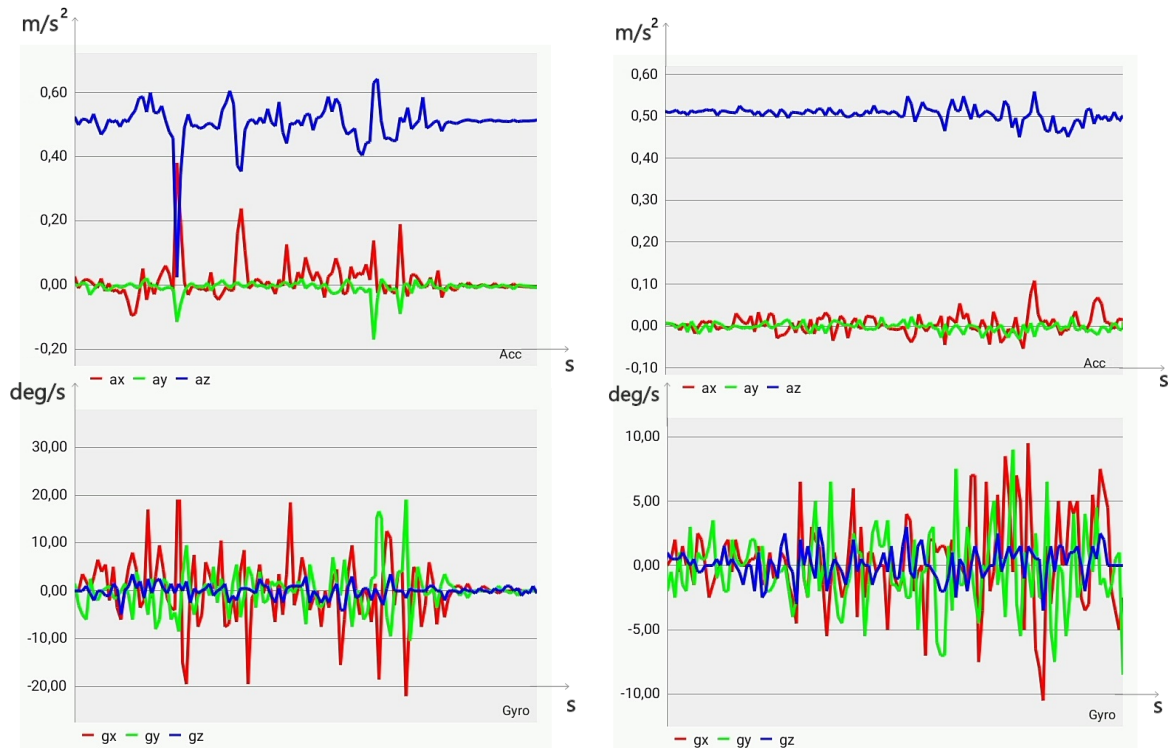
Les graphes suivants comparent les résultats obtenus avant et après l'application du filtrage. Ils mettent en évidence l'efficacité du filtre passe-bas, en montrant comment la réduction des perturbations a conduit à une amélioration substantielle de la stabilité du drone. Ces résultats confirment l'importance de traiter les données des capteurs avec soin pour optimiser les performances d'un système de contrôle tel qu'un quadcoptère.



(a) Armement du drone avec paramètres PID optimisés sans filtre appliqué.

(b) Armement du drone avec paramètres PID optimisés et filtre appliqué.

FIGURE 3.9 – Effet de l'application du filtre sur le drone lors de l'armement.



(a) Vol du drone avec paramètres PID optimisés sans filtre appliqué.

(b) Vol du drone avec paramètres PID optimisés et filtre appliqué

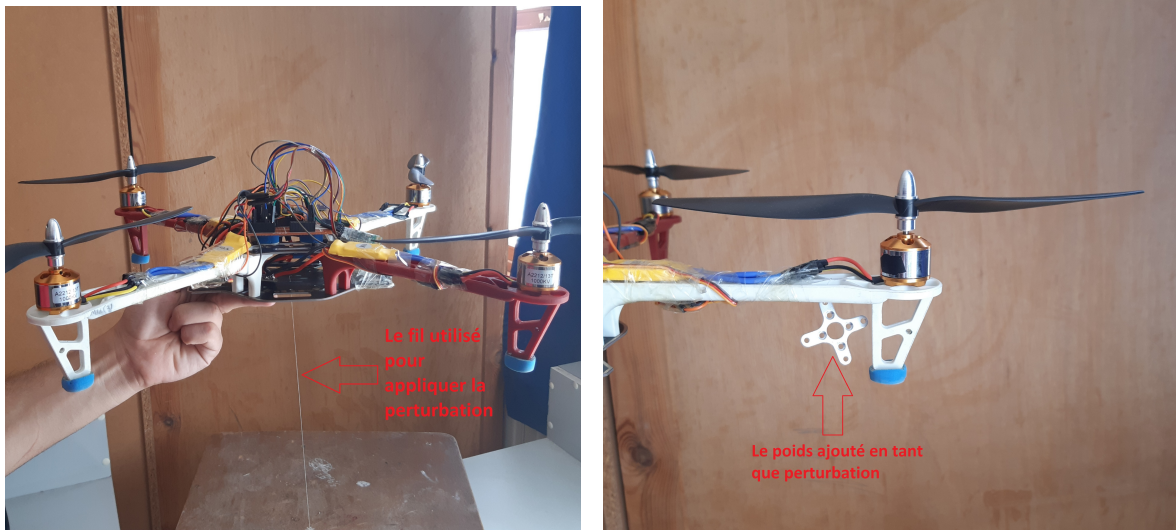
FIGURE 3.10 – Effet de l’application du filtre sur le drone en vol.

Remarque : Nous avons laissé tous les autres paramètres sur leurs valeurs par défaut, sans apporter de modifications. Nous avons préféré nous concentrer uniquement sur l’ajustement des paramètres PID et la calibration des capteurs principaux pour obtenir une stabilité de base. Cette approche nous permet de mieux comprendre l’impact des réglages PID sur le comportement du drone avant d’effectuer des ajustements supplémentaires sur d’autres paramètres.

3.3.9 Évaluation de la réponse du quadricoptère aux perturbations

Après avoir testé notre quadricoptère dans des conditions optimales et déterminé les paramètres idéaux pour le stabiliser, ainsi que pour rejeter les perturbations internes et celles causées par le vent généré par les hélices, nous avons entrepris de l’évaluer face à des perturbations plus significatives. Dans un premier temps, le drone a été attaché par son centre à l’aide d’un fil, que nous avons ensuite tiré brusquement vers le bas (FIGURE 3.11a). Cette manœuvre a permis d’observer la réponse immédiate du drone, qui a réagi rapidement et efficacement en corrigeant l’erreur. Par la suite,

un autre type de perturbation, prolongée dans le temps, a été introduit pour tester l'efficacité de l'action intégrale du contrôleur. Pour ce faire, un petit poids a été fixé à l'une des armatures du drone afin de provoquer un déséquilibre (FIGURE 3.11b). Là encore, le drone a démontré une capacité satisfaisante à rétablir son équilibre.



(a) Perturbation par fil.

(b) Perturbation par poids.

FIGURE 3.11 – Évaluation de la réponse du quadricoptère aux Perturbations

3.3.10 Illustration finale du drone

Après avoir construit notre drone et réalisé tous les ajustements nécessaires, nous avons réussi à le faire survoler de manière presque stable. Cela démontre l'efficacité de notre méthode de programmation et de calibration. Cependant, nous avons identifié quelques réglages supplémentaires qui pourraient encore améliorer sa performance et sa maniabilité. La (FIGURE 3.12) illustre la stabilité atteinte par le drone lors de ses vols, mettant en évidence les progrès réalisés dans le contrôle et la précision de vol. Ces ajustements fins permettront d'optimiser davantage l'expérience de vol, rendant le drone plus réactif et agréable à piloter.

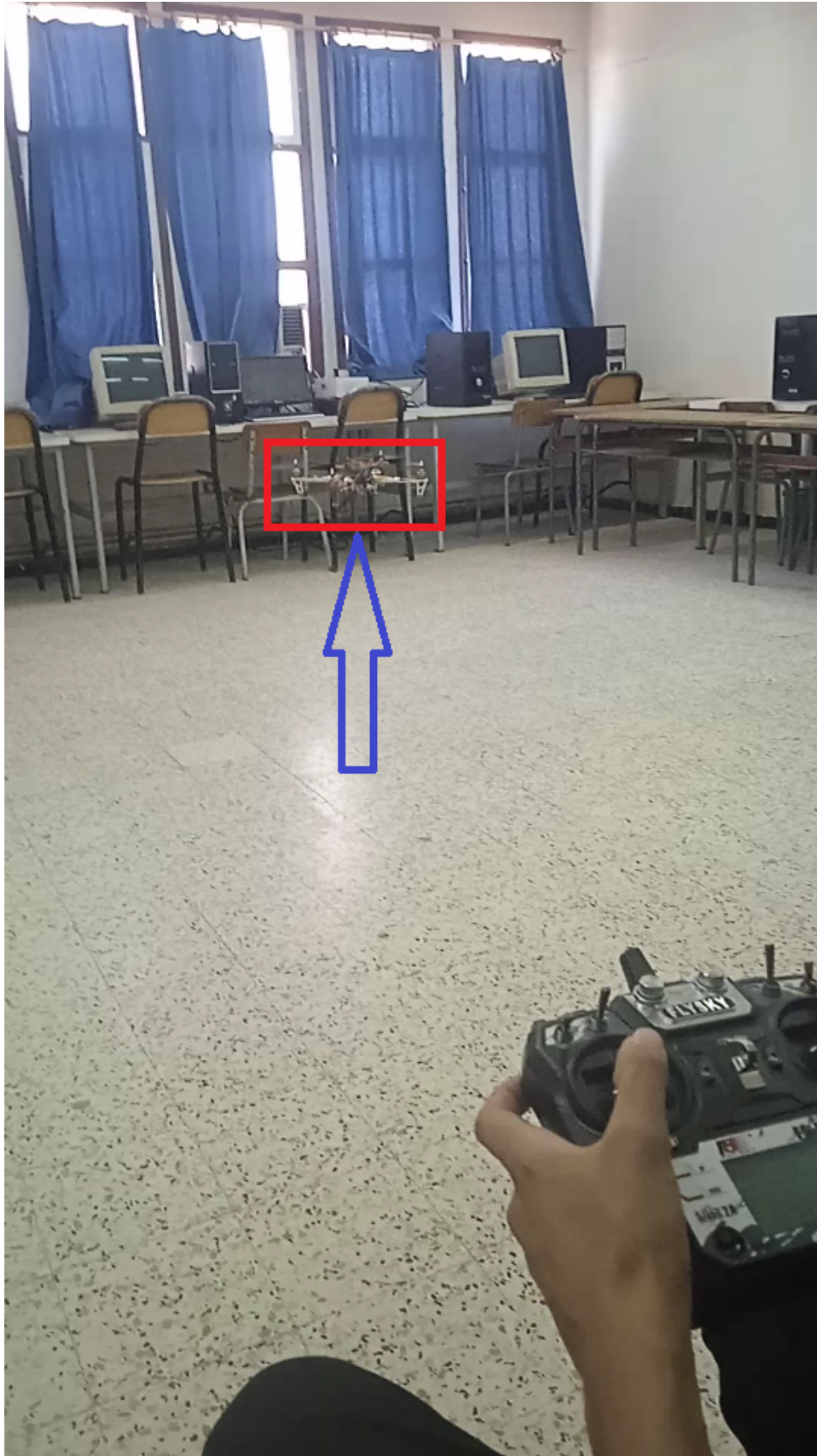


FIGURE 3.12 – Stabilité atteinte par le drone lors de ses vols.

3.3.11 Schéma récapitulatif des différentes parties du drone

Ce schéma résume toutes les parties du drone.

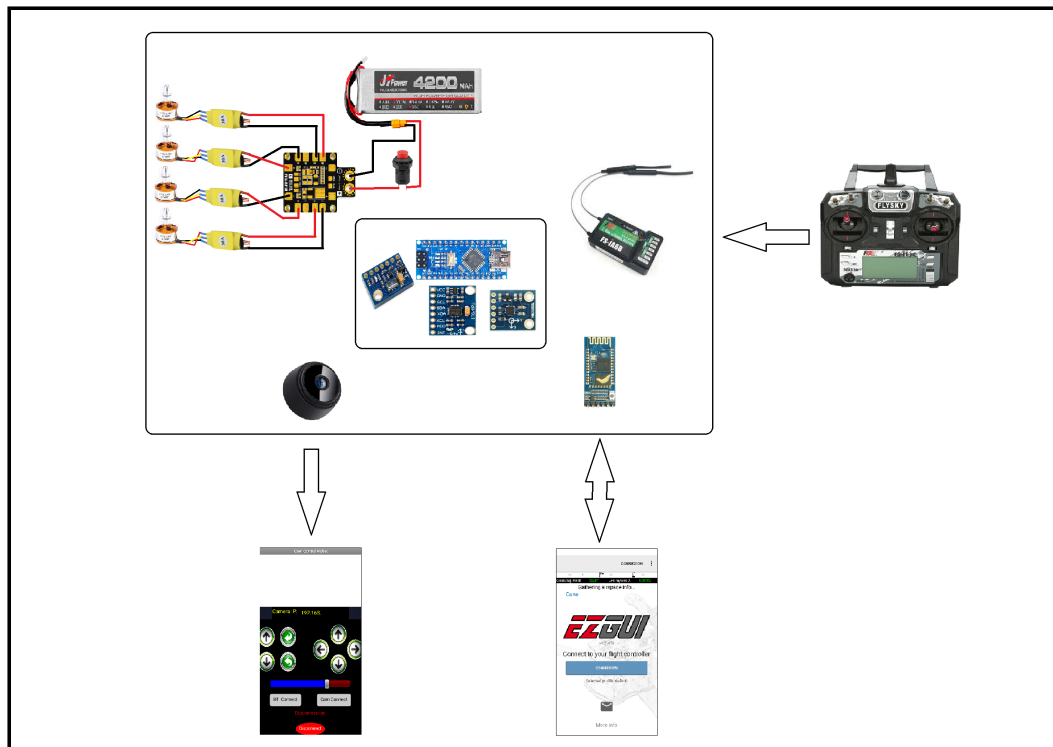


FIGURE 3.13 – Schéma global.

3.3.12 Conclusion

En adoptant la méthode de programmation MultiWii pour notre quadrirotor, nous avons réussi à accélérer le processus de développement tout en améliorant la performance et la stabilité de l'appareil. Grâce à une interface pratique et à la possibilité d'intégrer divers capteurs, nous avons pu optimiser notre drone de manière significative.

La configuration rigoureuse du matériel, le calibrage des ESC et des capteurs, ainsi que l'ajustement précis des paramètres PID ont été essentiels pour garantir un vol stable et réactif. L'intégration du module Bluetooth HC-05 a également ajouté une dimension de contrôle à distance, facilitant l'ajustement en temps réel des paramètres de vol.

Ces avancées, soutenues par une communauté active autour de MultiWii, offrent une base solide pour des projets futurs, permettant d'explorer des configurations plus complexes et des applications variées dans le domaine des drones. En somme, cette expérience a démontré que la flexibilité et l'accessibilité de MultiWii en font un choix judicieux pour les passionnés et les professionnels du développement de drones.

Conclusion générale

La réalisation de ce projet de quadricoptère nous a permis de mettre en pratique un large éventail de compétences techniques, théoriques et pratiques acquises au cours de notre formation. Depuis la modélisation mathématique jusqu'à la mise en œuvre du système de contrôle, en passant par l'intégration des divers composants électroniques, chaque étape de ce projet a représenté un défi unique, nécessitant une approche méthodique et rigoureuse.

L'objectif initial de concevoir un drone capable de voler de manière stable et précise a été atteint grâce à une série de choix techniques bien fondés. L'adoption de la plate-forme MultiWii s'est révélée judicieuse pour la gestion des capteurs et le contrôle du vol. Le développement et l'optimisation d'un contrôleur PID ont été essentiels pour assurer la stabilité de l'appareil en vol. Par ailleurs, l'intégration d'une application mobile pour le contrôle du drone via Bluetooth a non seulement enrichi le projet, mais a également démontré la faisabilité d'une solution de pilotage plus intuitive et accessible.

Au-delà des résultats techniques obtenus, ce projet a été l'occasion de développer une réelle capacité d'analyse et de résolution de problèmes complexes, en tenant compte des contraintes inhérentes à la conception d'un système aérien. Les multiples tests et ajustements effectués ont permis de valider les choix effectués, tout en mettant en lumière certaines limites du système, notamment en ce qui concerne la gestion des perturbations extérieures et l'optimisation des paramètres de contrôle.

En conclusion, ce travail constitue une contribution modeste mais significative à la compréhension des défis liés à la conception et au pilotage des quadricoptères. Il ouvre également la voie à des améliorations futures, telles que l'intégration de nouvelles technologies pour améliorer la stabilité, l'autonomie et les capacités de vol du drone. Nous espérons que ce projet servira de base pour des développements ultérieurs et qu'il pourra inspirer d'autres étudiants ou chercheurs à explorer davantage ce domaine passionnant et en pleine expansion.

Les codes sources, les schémas, les montages ainsi que l'application mobile développés dans le cadre de ce projet sont disponibles sur notre page GitHub : https://github.com/jetix1234/notre_projet_drone_1. Nous espérons que ce partage pourra bénéficier à d'autres personnes travaillant sur des projets similaires.

Perspectives et travaux futurs

Comme perspectives de notre travail, on suggère de concevoir un drone avec une meilleure autonomie de batterie, particulièrement pour les drones de surveillance qui nécessitent une grande autonomie énergétique. Nous proposons de remplacer les moteurs par un ballon rempli de gaz (hélium) pour réduire la consommation d'énergie. Pour diriger le drone, nous utiliserons de petits moteurs à faible consommation d'énergie, et pour alimenter le circuit, nous ajouterons un panneau solaire efficace et léger.

En résumé, nous cherchons à optimiser la stabilité, la sécurité et l'autonomie énergétique de nos drones grâce à des innovations dans le contrôle, les options de sécurité et les sources d'énergie alternatives.

Bibliographie

- [1] Amrani Mohamed, O. N. (2020). Etude et réalisation d'un drone quadrirotor. Mémoire de Master, Université 08 Mai 1945 de Guelma.
- [2] ARCHIBONG, E. A., NJOK, A. O., FISCHER, G., and ONOYOM-ITA, E. (2023). Design and development of a quadcopter with ground station monitoring unit.
- [3] BELOUFI, M. O., SEFAOUI, M. A. S., and BADIR, L. (2022). Détection de la corrosion à l'aide d'une nouvelle méthode basée sur des capteurs et antennes inspirée de la technologie rfid. Mémoire de Master, Université –Ain Temouchent- Belhadj Bouchaib.
- [4] Bin, H. and Justice, A. (2009). The design of an unmanned aerial vehicle based on the ardupilot. Indian Journal of Science and Technology, Beijing University of Chemical Technology.
- [5] Bouabdallah, S. (2007). Design and control of quadrotors with application to autonomous flying. Thèse de doctorat, École Polytechnique Fédérale de Lausanne (EPFL).
- [6] Collège Anatole France - Cadillac (2024). Composition d'un drone. http://afcadillac.net/_serveurs/drone/composition_dun_drone.html. Consulté le 13 septembre 2024.
- [7] FlySky (n.d.). Flysky fs-i6x user manual. <https://flysky.net/downloads/fs-i6x/>. Accessed : 2024-08-03.
- [8] Gamazo-Real, J. C., Vázquez-Sánchez, E., and Gómez-Gil, J. (2010). Position and speed control of brushless dc motors using sensorless techniques and application trends. *Sensors*. MDPI.
- [9] Gibbs, A. (2009). Lithium polymer batteries. *Guibbs Guides*, 328.

- [10] Gong, A. and Verstraete, D. (2017). Experimental testing of electronic speed controllers for uavs. In *53rd AIAA/SAE/ASEE joint propulsion conference*, page 4955.
- [11] Guliaev, N. (2017). The design, construction and implementation of an autonomous outdoor quadcopter using an rpi microcomputer and a multiwii flight controller.
- [12] Gupta, G. and Abdallah, S. (2018). Propeller force-constant modeling for multi-rotor uavs from experimental estimation of inflow velocity. *International Journal of Aerospace Engineering*.
- [13] Hassanalian, M. and Abdelkefi, A. (2017). Classifications, applications, and design challenges of drones : A review. *Progress in Aerospace Sciences*, 91 :99–131.
- [14] InvenSense Inc. (2013). Mpu-6000 and mpu-6050 product specification revision 3.4. <https://invensense.tdk.com/wp-content/uploads/2015/02/MPU-6000-Datasheet1.pdf>. Fiche technique du MPU-6000/6050, accédé en août 2024.
- [15] Islam, B. and Borah, M. J. (2023). Design and simulation of micro-controller based electric vehicle controller using proteus design suite. *Annals of Multidisciplinary Research, Innovation and Technology (AMRIT)*. Research article.
- [16] Janco, N., Long, X., Walling, E., and Mughal, M. (2020). Fire detection and suppression drone. Major Qualifying Project, Worcester Polytechnic Institute.
- [17] Johnson, M. A. and Moradi, M. H. (2005). *PID control*. Springer.
- [18] Khebbache, H. (2012). Tolérance aux défauts via la méthode backstepping des systèmes non linéaires : application système uav de type quadrirotor. Mémoire de Magister en Automatique, UNIVERSITE FERHAT ABBAS DE SETIF.
- [19] Koceila, M. and Hocine, M. (2015). Conception d’une loi de commande à l’aide des systèmes flous tsk «application à la commande du canal longitudinal d’un drone». Mémoire de Master Académique, UMMTO.
- [20] Lubis, S., Harahap, P., Pasaribu, F. I., Damanik, W. S., Siregar, M. A., Siregar, I., and Hasibuan, E. S. (2020). The design and qibla direction by using the hmc 5883l sensor as a compass rhi in the umsu science laboratory (oif). *Budapest International Research in Exact Sciences (BirEx) Journal*.

- [21] Molinaro, H. H. and Mazaleyrat, F. (2022). Mais qui est nikola tesla? Article pédagogique publié dans l'archive ouverte HAL, ENS Paris-Saclay.
- [22] MoussaSoft (2023). Hmc5883l 3-axis module avec arduino. <https://www.moussasoft.com/hmc5883l-3-axis-module-avec-arduino/>. Consulté le 13 septembre 2024.
- [23] Nana, V. B. T. (2020). Conception et réalisation d'un drone de transport de médicaments urgents. Mémoire de fin d'étude, University of Yaoundé I, Higher Technical Teacher Training College of Ebolowa, Ebolowa, Cameroon. En vue de l'obtention du Diplôme de Professeur d'Enseignement Technique et Professionnel de 2e grade (DIPET II).
- [24] Nano, A. (2018). Arduino nano. *A MOBICON Company*.
- [25] Ouacel, S. (2022). Conception et réalisation d'un drone quadri-rotor personnalisé. Mémoire de Master Académique, UMMTO.
- [26] Patton, E. W., Tissenbaum, M., and Harunani, F. (2019). Mit app inventor : Objectives, design, and development. *Computational thinking education*. Springer Singapore.
- [27] ProtoSupplies (2023). Gy-63 ms5611 pressure temperature sensor module. <https://protosupplies.com/product/gy-63-ms5611-pressure-temperature-sensor-module/>. Accessed : 2024-09-11.
- [28] Rafiq, A. A., Rohman, W. N., and Riyanto, S. D. (2020). Development of a simple and low-cost smartphone gimbal with mpu-6050 sensor. *Journal of Robotics and Control (JRC)*, 1(4) :136–140.
- [29] SAILESH, A. S. (2020). Design and fabrication of long endurance uav. Project Report, AMITY UNIVERSITY UTTAR PRADESH.
- [30] SELLAOUI, D. E. (2023). Etude et réalisation d'un drone quadrirotor. Mémoire de Master, Université 08 Mai 1945 de Guelma.
- [31] Sheikh, H. S., Sheikh, A. S., Sheikh, U. R., Hiware, U. V., Dhakate, S. A., and Sagore, S. S. (2020). Renewable energy based interleaved boost converter for the application of bldc motor. *IRJET*.
- [32] Shutterstock (n.d.). Quadrocopter/uav usage infographic vector icons for science, medicine, and technology. <https://www.shutterstock.com/image-vector/>

quadrocopteruav-usage-infographic-vector-icons-science-281575166.

Consulted on 2024-07-29.

- [33] Singh, T. D. and Kumar, M. (2021). Design and development of bluetooth based home automation system using fpga. *International Journal of Applied Engineering Research*.
- [34] Wodka-Gallien, P. (2001). Les drones, un siècle de développement et d'utilisation militaire. *Revue historique des Armées*.

Code Arduino développé pour le bicoptère

```
#include <PID_v1_bc.h>
#include <Wire.h>
#include <MPU6050.h>
// #include <PID_v1.h>
#include <Servo.h>

// Initialisation du MPU6050
MPU6050 mpu;

// PID constants
double Kp = 100, Ki = 5.0, Kd = 1.0;

// PID variables
double setpoint, input, output;

// PID controller
PID myPID(&input, &output, &setpoint, Kp, Ki, Kd, DIRECT);

// Variables de contrôle du bicoptère
Servo motor1;
Servo motor2;

// Paramètres de la fonction de transfert
double K_system = 1.0; // Gain statique
double tau_system = 0.5; // Constante de temps
double delta_t = 0.02; // Intervalle de temps (20 ms)
double system_output = 0; // Sortie du système (angle)
```

```

// Variables pour l'angle
double angle = 0;

void setup() {
  // Initialisation des moteurs
  motor1.attach(9); // Attache le moteur 1 au pin 9
  motor2.attach(10); // Attache le moteur 2 au pin 10

  // Initialisation de la communication série
  Serial.begin(115200);

  // Initialisation du MPU6050
  Wire.begin();
  mpu.initialize();
  if (!mpu.testConnection()) {
    Serial.println("Erreur de connexion MPU6050");
    while (1);
  }

  // Ajouter un délai après l'attachement des servos
  delay(1000);

  // Initialisation des valeurs du PID
  setpoint = 0; // consigne = Pid = 0
  myPID.SetMode(AUTOMATIC);
  myPID.SetOutputLimits(-150, 150); // limites des valeurs de sortie pour les moteurs

  // Initialisation des moteurs
  for (int i = 1000; i <= 1150; i++) {
    motor1.writeMicroseconds(i);
    motor2.writeMicroseconds(i);
    Serial.print("i= "); Serial.println(i);
    delay(40);
  }
}

void loop() {
  for (int j = 0; j <= 500; j++) {
    // Lire l'angle actuel du MPU6050
    int16_t ax, ay, az, gx, gy, gz;
    mpu.getMotion6(&ax, &ay, &az, &gx, &gy, &gz);

    // Calculer l'angle du bicoptère (simplifié)
    angle = atan2(ay, az) * 180 / PI;

    // Fonction de transfert discrétisée
    system_output = (1 - delta_t / tau_system) * system_output + (K_system * delta_t / tau_system) * angle;

    // Mettre à jour l'entrée du PID avec la sortie du système
    input = system_output;

    // Calculer la correction PID
    myPID.Compute();
  }
}

```

```

// Ajuster les moteurs en fonction de la sortie du PID
int motor1Speed = constrain(1150 + output, 1000, 1250); // Ajuster les valeurs selon les besoins
int motor2Speed = constrain(1150 - output, 1000, 1250);

motor1.writeMicroseconds(motor1Speed);
motor2.writeMicroseconds(motor2Speed);

// Afficher les valeurs pour le débogage
Serial.print("Angle: "); Serial.print(angle);
Serial.print(" System Output: "); Serial.print(system_output);
Serial.print(" Input: "); Serial.print(input);
Serial.print(" Output: "); Serial.print(output);
Serial.print(" Motor1: "); Serial.print(motor1Speed);
Serial.print(" Motor2: "); Serial.println(motor2Speed);

// Attendre un court instant avant la prochaine boucle
delay(20);
}

motor1.writeMicroseconds(1150);
motor2.writeMicroseconds(1150);
delay(2000);

for (int z = 1150; z >= 1000; z--) {
  motor1.writeMicroseconds(z);
  motor2.writeMicroseconds(z);
  Serial.print("z= "); Serial.println(z);
  delay(20);
}

while (1) {
  motor1.writeMicroseconds(1000);
  motor2.writeMicroseconds(1000);
  Serial.print("z= 1000");
  delay(10);
}
}

```