

N° Ordre...../Faculté/UMBB/2022

REPUBLIQUE ALGERIENNE DEMOCRATIQUE ET POPULAIRE
MINISTERE DE L'ENSEIGNEMENT SUPERIEUR ET DE LA RECHERCHE SCIENTIFIQUE
UNIVERSITE M'HAMED BOUGARA-BOUMERDES



Faculté des Hydrocarbures et de la Chimie

Mémoire de Fin d'Etudes
En vue de l'obtention du diplôme :

MASTER 2

Présenté par

CHERRERI Soheib

KEBOUDJI Abderrahmen

Filière : Automatisation des procédés industriels

Option : Commande automatique

Thème

**Réalisation et commande d'un drone quadrirotor et
la reconstruction 3D**

Devant les jurys :

Mr. HAMADACHE Mohamed	MCB	FHC	Examineur
Mr. KHEBLI ABDELMALEK	MAA	FHC	Examineur
Mme. KAHOU L Fadhila	MCB	FHC	Encadreur

Année Universitaire : 2021/2022



Faculté des Hydrocarbures et de la Chimie

Département : Automatisation

Filière : Automatisation des procédés industriels

Option : Commande automatique

Mémoire de Fin d'Etudes

En vue de l'obtention du diplôme :

MASTER 2

Thème

**Réalisation et commande d'un drone quadrirotor et
la reconstruction 3D**

Présenté par :

CHERRERI Soheib

KEBOUDJI Abderrahmen

Avis favorable de l'encadreur :

F. KAHOU

signature

Avis favorable du Président du jury

Nom Prénom

Signature

Mr. HAMADACHE Mohamed

Cachet et signature

Remerciement

Tout d'abord nous remercions "ALLAH" et remercions pour la réalisation de cet humble travail.

*Nous remercions notre encadreur **madame** « F. KAHOU » pour sa contribution efficace les professeur superviseur en tant que comite' arbitrage et les autres professeurs du département.*

Nous remercions les honorables parents pour leurs appels sincères Et remercions tous ceux qui ont contribué de près ou de loin en leur Souhaitant bon chance et beaucoup de sucée.

Qui avons-nous accompagnés dans ce modeste travail avec tous patience et encouragement.

Nous espérons Enfin que cette offre sera acceptée et admirée

Dédicace

Je dédie ce modeste travail:

*Spécialement à la mémoire de **mon père** qui a toujours cru en moi, tu vivras toujours dans mon cœur. Je demande à Dieu d'avoir pitié de lui et d'habiter au paradis.*

*À **ma chère mère** à qui je voue mon respect et tous mes vœux de bonheur, prospérité et de bonne santé.*

*À **ma chère sœur, mes chers frères et ma famille** qui m'ont toujours soutenu.*

*Ainsi qu'à tous **mes proches** et à **mes amis** pour leurs encouragements et leurs soutiens qui se reconnaîtront.*

À Tous ceux qui m'aiment et que j'aime.

Et à toute la promo H17,

CHERRERI Soheib

Dédicace

Je dédie ce travail à :

A ma tendre Mère Soria et mon très cher père Elhani : Aucune dédicace ne saurait exprimer l'amour, l'estime, le dévouement et le respect que j'ai toujours pour vous. Vous représentez pour moi la source de tendresse et l'exemple de dévouement qui n'a pas cessé de m'encourager et de prier pour moi. Rien au monde ne vaut les efforts fournis jour et nuit pour mon éducation et mon bien être. Que Dieu vous garde et vous protège.

A mon frère et mes sœurs pour leurs soutien moral. Ce travail soit témoignage de ma reconnaissance et de mon amour sincère pour vous.

A tous mes camarades de groupe MACA17 et à tous les membres de club scientifique AEPI New Vision.

A tous mes enseignants depuis mes premières années d'études.

A tous ceux qui me sens chers et que j'ai omis de citer.

KEBOUDJI Abderrahmen

Résumé

Les drones sont des aéronefs sans pilote capables de faire des missions d'une façon plus ou moins autonome et automatique. Le travail présenté dans ce mémoire avait comme objectif la conception et la réalisation d'un prototype drone quadrirotor et l'application de la reconstruction 3D à partir des images. Notre drone se compose de quatre moteurs. Ces moteurs sont contrôlés par un contrôleur de vol basé sur l'Arduino via un contrôleur de vitesse électronique ESC, l'ensemble est alimenté par une batterie LI Po et piloté à l'aide d'une radiocommande Flysky FS-I6X. La stabilisation du drone a été assurée en implémentant une loi de commande de type PID.

La photogrammétrie est un processus qui peut être utilisé pour la reconstruction 3D précises à partir de images. Ce projet présente la mise en œuvre une description des méthodes et des données utilisées pour reconstruire les modèles 3D.

Abstract

Drones are unmanned aircraft capable of performing missions in a more or less autonomous and automatic way. The work presented in this thesis had as objective the design and realization of a prototype quadrotor drone and the application of 3D reconstruction from the images. Our drone consists of four motors. These motors are controlled by a flight controller based on Arduino via an electronic speed controller ESC, the whole is powered by a LI Po battery and controlled using a Flysky FS-I6X radio control. The stabilization of the UAV was ensured by implementing a PID control law.

Photogrammetry is a process that can be used for accurate 3D reconstruction from images. This project presents the implementation and description of the methods and data used to reconstruct the 3D models.

ملخص

الطائرات بدون طيار (الدرون) هي طائرات قادرة على أداء المهام بطريقة مستقلة وأتوماتيكية العمل المقدم في هذه المذكرة هو تصميم نموذج لطائرة رباعية بدون طيار وتطبيق إعادة البناء ثلاثي الأبعاد. تتكون طائرتنا بدون طيار من أربع محركات يتم التحكم في هذه المحركات بواسطة وحدة تحكم طيران تعتمد على أريدينو عبر وحدة تحكم إلكترونية للسرعة، ويتم تشغيل الكل بواسطة بطارية كما يتم التحكم فيها باستخدام جهاز تحكم لاسلكي. تم ضمان استقرار الطائرة بدون طيار من خلال متحكم تناسبي تكاملي تفاضلي.

المسح التصويري هو عملية يمكن استخدامها لإعادة البناء ثلاثي الأبعاد بدقة من الصور. يقدم هذا المشروع وصفاً للطريقة والبيانات المستخدمة في صنع النماذج ثلاثية الأبعاد باستخدام الصور.

Table des matières

Chapitre I : État de l'art sur les drones et la photogrammétrie

I.1 Introduction.....	3
I.2 Le drone	4
I.2.1 Définition et mise en contexte	4
I.2.2 Types de drones	4
I.2.2.1 Drones à voilure fixe.....	5
I.2.2.2 Drones à voilure tournante	5
I.2.3 Domaines d'applications.....	7
I.2.3.1 Drone militaires	7
I.2.3.2 Drone civil	8
I.3 Quadrirotor.....	9
I.3.1 Description.....	9
I.3.2 La technologie.....	9
I.3.3 Avantages de la configuration quadrirotor	10
I.4 La photogrammétrie.....	11
I.4.1 Introduction.....	11
I.4.2 Historique	12
I.4.3 Reconstruction 3D	14
I.4.4 Domaines d'application de la reconstruction 3D.....	15
I.4.4.1 Entraînement	16
I.4.4.2 Simulation	16
I.4.4.3 Compression de données.....	16
I.4.4.4 Imagerie médicale	16
I.4.4.5 Commerce	17
I.4.4.6 Conservations.....	17
I.4.4.7 Autres applications.....	17
I.4.5 Principe de fonctionnement	17
I.4.5.1 Acquisition des images et sélection des régions d'intérêt.....	19
I.4.5.2 Détection des points d'intérêt et extraction des caractéristiques.....	19

I.4.5.3 Correspondance des points d'intérêt et élimination des correspondances non fiables	24
I.4.5.4 Estimation de la Matrice essentielle et Rectification	25
I.4.5.5 Calcul de la carte de disparité	27
I.4.5.6 Triangulation et création du fichier PLY	27
I.4.5.7 Génération du Modèle 3D.....	28
I.5 Conclusion	28

Chapitre II : Anatomie d'un drone type quadrirotor et ses composants

II.1 Introduction	29
II.2 Les composants de notre réalisation.....	29
II.2.1 Le châssis.....	29
II.2.2 Les moteurs.....	32
II.2.3 les hélices.....	34
II.2.4 Les Contrôleurs de vitesse électroniques (ESC).....	36
II.2.5 MPU6050.....	39
II.2.6 Source et gestion d'énergie Batterie LiPo	43
II.2.7 La carte Arduino Mega 2560.....	44
II.2.8 Système de Communication (Le transmetteur FS-i6X et son récepteur FS-iA10B)....	45
II.3 Les mouvements du quadrirotor	47
II.3.1 Les mouvements de rotation du quadrirotor :.....	48
II.3.1.1 Mouvement de lacet.....	48
II.3.1.2 Mouvement de roulis :.....	49
II.3.1.3 Mouvement de tangage :.....	50
II.3.2 Les mouvements de translations du quadrirotor.....	50
II.3.2.1 Translation verticale	50
II.3.2.2 Translation horizontale	51
II.4. Conclusion.....	52

Chapitre III : Tests réels & discussions de résultats

III.1 Introduction.....	53
III.2 Montage et réalisation du quadrirotor	53
III.2.1 Expériences sur MPU-6050	53
III.2.2 Expériences sur Le transmetteur FS-i6X et son récepteur FS-iA10B.....	60

III.2.2.1 Protocoles TX.....	60
III.2.2.2 Protocoles RX	61
III.2.2.3 Première expérience	62
III.2.2.4 Deuxième expérience	66
III.2.3 Expériences sur l'équipement	68
III.2.3.1 Calibrage de l'ESC	71
III.3 Conclusion	74

Chapitre IV : Commande d'un drone type quadrirotor par régulateur PID

IV.1 Introduction.....	75
IV.2 Le régulateur PID du quadrirotor.....	75
IV.2.1 Saturation du régulateur PID et choix des gains [26]	78
IV.2.2 Algorithme PID du régulateur de vol	78
IV.2.3 Sous-programme de sortie PID.....	80
IV.2.4 Contrôle des ESC	82
IV.3.1 Conception du matériel	88
IV.3.2 Test du régulateur	89
IV.4 Conclusion	91

Chapitre V : La reconstruction 3D

V.1 Introduction	92
V.2 Le principe.....	92
V.3 Les étapes de la détermination du nuage de points	93
V.3.1 Étape 01 : L'initialisation de la Camera.....	93
V.3.2 Étape 02 : L'extraction des caractéristiques d'image.....	95
V.3.3 Étape 03 : Correspondance d'image.....	96
V.3.4 Étape 04 : Correspondance de caractéristiques d'image	97
V.3.5 Structure From Motion.....	97
V.4 Logiciels utilisés.....	100
V.4.1 MeshLab.....	100
V.4.2 Meshroom	101
V.4.3 3DF Zephyr	102
V.5 Exemples de reconstruction 3D	103
V.5.1 Boite de jus (49 photos)	104

V.5.2 Flysky (69 photos).....	105
V.5.3 Pumpjack (221 photos)	107
V.6 Critique des résultats obtenus.....	108
V.7 Conclusion.....	109

Table des figures

Figure I.1 : Drone dit ‘aile volante.....	4
Figure I.2 : L’hélicoptère est un sous-type de drone à voilure tournante. Le rotor de queue évite que l’appareil ne se mette à tourner sur lui-même.....	5
Figure I.3 : Configurations fréquentes des hélices des multicoptères. Le sens de rotation des hélices est indiqué par une flèche colorée.....	5
Figure I.4 : Drone militaire.	6
Figure I.5 : Drone civil.....	7
Figure I.6 : Numérisation 3D réalisée par photogrammétrie.....	11
Figure I.7 : Les techniques de relevé géométrique à haute définition produisent des fichiers 3D bruts sous forme soit de nuages de points, soit de maillages.....	12
Figure I.8 : Organigramme fonctionnel de la reconstruction 3D.....	14
Figure I.9 : Tableau des pixels de FAST.....	16
Figure I.10 : Pyramide La placienne générée par ORB.....	17
Figure I.11 : La sélection de pixel par ORB.....	18
Figure I.12: Coordonnées du monde et de la caméra.....	19
Figure II.1 : Différents châssis de drones.....	22
Figure II.2 : Châssis F450.....	24
Figure II.3 : Configuration des pins du moteur Brushless.....	25
Figure II.4 : Pair d’hélice 10×45 (CW et CCW).	28
Figure II.5 : Branchement d’un ESC avec moteur brushless.	28
Figure II.6 : Anatomie d'un ESC.	29
Figure II.7 : Le module MPU6050.....	30
Figure II.8 : Branchement du MPU6050 avec Arduino.	32

Figure II.9 : La batterie Li Po 3S 4200 mAh 25C.	33
Figure II.10 : l'Arduino Mega 2560.	34
Figure II.11 : Flysky FS-i6 et son récepteur FS-iA10B.	35
Figure II.12: Structure générale d'un quadrirotor.....	37
Figure II.13 : Illustration du mouvement de lacet.....	37
Figure II.14 : Mouvement de roulis de quadrirotor.....	38
Figure II.15 : Mouvement de tangage de quadrirotor.....	38
Figure II.16 : Translation verticale de quadrirotor.....	39
Figure II.17: Illustration du mouvement de translation.....	39
Figure I.1 : Branchement du MPU6050 avec Arduino Mega 2560.....	42
Figure II.2 : Programme pour obtenir des mesures d'accélération (x, y, z) sur MATLAB.....	42
Figure III.3 : Mesures d'accélération en fonction de temps.....	43
Figure III.4 : Programme pour obtenir des mesures de gyroscope (x, y, z) sur MATLAB.....	44
Figure III.5 : La variation de vitesse angulaire de gyroscope en fonction de temps.....	45
Figure III.6 : l'organigramme qui explique le processus.....	46
Figure III.7 : Programme pour tracer les courbes pitch, roll, yaw.....	47
Figure III.8 : Les mesures d'angle selon mouvement de gyroscope en fonction de temps.....	48
Figure III.9 : Les différents modes.....	49
Figure III.10: branchement du FS-iA10B avec Arduino Mega 2560.....	51
Figure III.11: Programme de communication avec l'Arduino et les canaux de transmetteur.....	51
Figure III.12 : Les valeurs des chaînes avant l'étalonnage.....	52
Figure III.13 : Les valeurs des chaînes après l'étalonnage.....	52
Figure III.14 : La variation des valeurs des chaînes selon les mouvements des sticks.....	52

Figure III.15 : Branchement du FS-iA10B (IBUS SERVO) avec Arduino Mega 2560.....	53
Figure III.16: Programme de communication avec l'Arduino et les canaux de transmetteur.....	54
Figure III.17 : Les valeurs des chaînes.....	54
Figure III.18 : La variation des valeurs des chaînes.....	55
Figure III.19 : Difference entre Servo moteur et BLDC moteur.....	56
Figure III.20 : Branchement Arduino, ESC, BLDC moteur et PWM signale.....	57
Figure III.21 : Calibration de l'ESC.....	58
Figure III.22 : Branchement Arduino mega 2560 avec ESC et FS-iA10B.....	59
Figure III.23 : Programme Arduino pour contrôler la vitesse de moteur avec FS-i6X.....	59
Figure III.24 : La vitesse de notre moteur et l'entrée que nous avons donnée.....	60
Figure IV.1 : Schéma de la boucle de régulateur PID.....	61
Figure IV.2: Schéma montre le principe du PID utilisé dans ce régulateur de vol.....	62
Figure IV.3: L'organigramme qui explique l'étape de conversion des variables du récepteur en degrés / seconde pour un axe de mouvement « Roll ».....	64
Figure IV.4 : L'algorithme utilisé dans le sous-programme PID du régulateur de vol pour un axe de mouvement.....	65
Figure IV.5: L'organigramme qui explique le calcul du sorties PID pour un axe de mouvement «Roll».....	66
La Figure IV.6: L'algorithme calculer les impulsions des ESC.....	67
La Figure IV.7: Les axes (Roll, Pitch et Yaw) et les mouvements du quadrirotor.....	68
Figure IV.8: L'algorithme du PID conduit le quadrirotor de revenir À l'état voulu en agissant sur les ESCs.....	69
Figure IV.9: Les impulsions des contrôleurs de vitesse esc avec le signe de régulateur.....	69

Figure IV.10: L’organigramme qui explique l’algorithme utilisé pour générer des impulsions PWM aux ESC.....	70
Figure IV.11: Conception du matériel du quadrirotor.....	71
Figure IV.12: Réponses d'attitude du quadrirotor avec le régulateur PID.....	72
Figure IV.13: Signal de commande.....	72
Figure IV.14: Le quadrirotor utilisé dans notre projet avec radiocommande Flysky.....	72
Figure V.1 : prise de vue différentes d’un damier.....	76
Figure V.2 : Détection des coins des damiers sur la mire de calibration.....	76
Figure V.3 : Organigramme de l’étape 5.....	80
Figure V.4 : Visualisation d'un nuage de point sous MeshLab.....	81
Figure V.5 : Modèle 3D de Flysky sur meshroom.....	82
Figure V.6 : logiciel 3DF Zephyr.....	82
Figure V.7 : Photos de la boite de jus.....	83
Figure V.8 : Nuage de points de la boite de jus.....	83
Figure V.9 : Modèle 3D de la boite de jus.....	84
Figure V.10 : Nuage de points de Flysky.....	84
Figure V.11 : Modèle 3D de Flysky sur 3D Zephyr.....	85
Figure V.12 : Photos structure 1.....	85
Figure V.13 : Photos structure 2.....	85
Figure V.14 : Nuage de points Structure.....	86
Figure V.15 : Modèle 3D Structure1.	86
Figure V.16 : Modèle 3D Structure2.	86
Figure V.17 : Modèle 3D Structure.....	86

Table des tableaux

Tableau II.1 : Tableau indique les Caractéristiques techniques de Châssis Quadcoptères F450...	23
Tableau II.2 : Caractéristique de moteur.....	24
Tableau II.3 : Configuration des pins du moteur Brushless.....	25
Tableau II.4 : Combinaisons hélices et moteurs.....	26
Tableau II.5 : caractéristiques d'une hélice.....	27
Tableau II.6 : La fiche technique d'hélice 10×45.....	27
Tableau II.7 : Configuration du brochage du MPU6050.....	31
Tableau II.8 : Branchement du MPU6050 avec Arduino.....	32
Tableau II.9 : La fiche technique de la batterie Li Po 3S 4200 mAh 25C.....	33
Tableau II.10 : Les caractéristiques d'Arduino Mega 2560.....	34
Tableau II.11 : Spécification de FlySky FS-i6.....	35
Tableau II.12 : Spécification de FS-iA10B.....	36
Tableau III.1 : Branchement du MPU6050 avec Arduino Mega 2560.....	41
Tableau III.2 : branchement du FS-iA10B avec Arduino Mega 2560.....	50
Tableau III.3 : Branchement du FS-iA10B (IBUS SERVO) avec Arduino Mega 2560.....	53
Tableau III.4 : Branchement Arduino Mega 2560 avec FS-iA10B.....	58
Tableau III.5 : Branchement Arduino Mega 2560 avec ESC.....	59
Tableau IV.1 : Appellation des rotations pour différents degrés de liberté.....	63
Tableau IV.2: Les constantes PID.....	7

Introduction générale

Introduction générale

Depuis un certain nombre d'années, on assiste à l'émergence de petits véhicules aériens sans pilote à bord, utilisés notamment pour des applications d'inspection ou de surveillance.

Avec les progrès constants en termes de miniaturisation et réduction des coûts des composants électroniques tels que centrales inertielles, capteurs de vision, ou microprocesseurs, de petits engins aériens pesant seulement quelques kilos peuvent maintenant bénéficier des avancées technologiques qui n'étaient jusqu'alors accessibles que pour des avions classiques. Ceci va permettre le développement d'engins beaucoup plus sophistiqués, dotés de fortes capacités de perception et d'action.

Le domaine de la robotique aérienne a beaucoup évolué au cours de ces dernières décennies, Les robots aériens ou drones (Remotely Piloted Aircraft Systems, en anglais RPAS) sont des engins volants sans pilote, capables de mener à bien une mission de façon plus ou moins autonome. Contrairement aux activités d'aéromodélisme, leur fonction principale est de réaliser une tâche « répétitive, sale ou dangereuse » (Dull, Dirty, Dangerous) de façon plus sûre ou plus efficace qu'un engin piloté à distance. La possibilité d'utiliser des systèmes autonomes tels que les drones aériens dans le secteur civil a été depuis quelques années largement testée grâce à de nombreux projets de recherche, certains financés par la Commission européenne, qui ont permis d'étudier des concepts d'application potentielle des drones comme vecteur expérimental.

La photogrammétrie est une méthode qui permet d'effectuer une modélisation en 3 dimensions et de prendre des cotes ou mesures dans un décor. Cette technique de mesure sans destruction et sans toucher l'objet à mesurer, utilise la parallaxe obtenue entre des prises des vues capturées avec des positions différentes. La méthode se base sur la vision stéréoscopique humaine. Actuellement, la photogrammétrie exploite majoritairement les calculs de corrélation entre des photographies numériques grâce à des logiciels de photogrammétrie très poussés. La photogrammétrie est utilisée depuis plus de 100 ans, mais a été démocratisée avec les drones civils, dans le but de simplifier les levés topographiques de grandes surfaces, ou de modéliser des bâtiments de manières précises afin d'obtenir des volumes précis ou pour la sauvegarde des monuments. Les applications de la photogrammétrie par drone sont très variées.

L'utilisation du drone nous permet de produire un grand volume de données 3D géo-référencées en peu de temps avec des possibilités de prises de vues complexe. Plus proches de l'industrie, les drones se sont surtout développés dans l'inspection de bâtiments et la surveillance d'infrastructures (pipeline, ligne, électrique, ligne de chemin de fer), L'installation d'oléoducs et de gazoducs nécessite une planification importante. Les enquêtes par drone aident ces entreprises à évaluer où les pipelines peuvent et ne peuvent pas passer, que ce soit en raison de la proximité de ressources naturelles et de sites de conservation, de constructions ou d'infrastructures préexistantes ou de propriétés privées.

Le but de ce projet est de faire la conception et la commande à base de PID d'un drone quadrirotor à base Arduino, ainsi que la réalisation d'un prototype. Ensuite une représentation tridimensionnelle des objets d'intérêt à partir d'images acquises par nos smartphones pour des premiers tests ainsi par notre drone quadrirotor.

Le mémoire est organisé en cinq chapitres :

Chapitre 01 : État de l'art sur les drones et la photogrammétrie : Dans ce chapitre, nous explorons les différents types de drones et leur historique de développement, avant d'aborder leurs diverses utilisations, ainsi que nous avons présenté la photogrammétrie et sa fonctionnalité principale.

Chapitre 02 : Anatomie d'un drone type quadrirotor et ses composants : la présentation des composants nécessaires à la construction d'un drone quadrirotor, et également une description détailler sur les mouvements du drone.

Chapitre 03 : Commande d'un drone type quadrirotor : dans ce chapitre nous avons présenté des expériences sur le fonctionnement des différentes parties de notre système, le branchement de matériel et une explication sur le logiciel utilisé pour ces expériences.

Chapitre 04 : Tests réels & discussions de résultats : la réalisation d'un véhicule aérien de type Quad X et d'un système de régulation de type PID, ce chapitre est dédié principalement à effectuer des tests réels et présenter les résultats de rotation du drone selon les trois axes x, y et z.

Chapitre 05 : La reconstruction 3D : Ce chapitre est dédié à tous les étapes de la reconstruction 3D, les résultats de nos tests et la critique des différents résultats obtenus.

Enfin nous avons conclus avec une conclusion générale.

Chapitre I :
État de l'art sur les drones et la
photogrammétrie

Chapitre I :

État de l'art sur les drones et la photogrammétrie

I.1 Introduction

Il y a quelques années encore, les drones étaient réservés à des applications de pointe, principalement dans le domaine militaire. Beaucoup ont découvert cette technologie en regardant les images des conflits armés du début des années 2000. A la même époque, on trouvait également des avions ou hélicoptères télécommandés dans les clubs d'aéromodélisme, sous forme de modèles réduits dont la construction et le pilotage demandaient de l'enthousiasme et de la persévérance.

Aujourd'hui, des drones 'intelligents' ont conquis le grand public. Les images d'un drone 'livreur de colis' imaginé par un célèbre distributeur en ligne ont, par exemple, fait le tour du monde. Les appareils modernes sont accessibles à tous, en témoignent les petits drones disponibles pour quelques dizaines d'euros sur Internet, qui permettent déjà de réaliser de petits films aériens sans requérir de compétences de pilotage pointues. Au niveau des secteurs professionnels, la grande diversité des capteurs et des outils que l'on peut embarquer permet d'envisager l'utilisation du drone pour de nombreuses tâches. L'industrie minière, le secteur énergétique, le secteur agricole et forestier, les médias ou encore la promotion immobilière s'approprient rapidement la technologie[1].

Le drone moderne est le fruit de nombreuses innovations technologiques mises au point au cours des dernières décennies, Le présent chapitre vise à comprendre ce contexte technologique, nous présentant des généralités sur les drones en donnant la définition, un historique domaines d'application, Ensuite la photogrammétrie et son principe de fonctionnement [1].

I.2 Le drone

I.2.1 Définition et mise en contexte

Historiquement, le concept n'est pas neuf et l'utilisation d'avions sans pilote était avancée¹. Déjà, pendant la première guerre mondiale, des prototypes d'avions sans pilote radio-commandés ont vu le jour. A titre d'exemple en France, le 2 juillet 1917, le pilote Max Boucher fera voler un avion sans intervention de l'homme sur 1km².

Au début de l'année 1918, Georges Clémenceau lui-même présentera à la commission sénatoriale de l'armée un projet d'avions sans pilote. Au cours des années 30', c'est principalement l'armée allemande qui développera des recherches sur des vecteurs guidés à distance et prenant la forme de bombes planantes anti-navires. Il faut toutefois attendre les guerres de Corée et du Vietnam pour qu'une réelle technologie des drones militaires se mette en place. C'est aussi au cours des années 70' que les Etats-Unis vont progressivement partager leur technologie avec Israël qui deviendra un des pays les plus en pointe au niveau mondial. Enfin, les années 90' verront émerger la doctrine « zéro mort » qui conduira à développer les projets de drones armés à travers le monde avec pour conséquence l'utilisation de cette technologie dans tous les conflits armés à partir des années 2000.

Pour rappel, le drone est également appelé UAV (Unmanned Aerial Vehicle) ou encore RPAS (Remotely Piloted Aircraft System)³. Il est télécommandé, autonome et peut emporter une charge utile destinée à des missions de type militaire. La taille du drone ainsi que sa masse dépendra des capacités recherchées. Enfin, le drone peut avoir une autonomie de vol de plusieurs dizaines d'heures contrairement aux avions classiques [2].

I.2.2 Types de drones

Il existe de nombreux types de drones, chacun avec une forme, une taille, un système de propulsion ou une source d'énergie différents. Les drones ressemblaient à des appareils d'aviation traditionnels : ils ressemblaient à des avions ou à des hélicoptères. Ensuite, les formulaires ont été spécialisés pour répondre au mieux aux missions spécifiques envisagées, en éliminant les éléments inutiles et en optimisant certains points clés pour le vol télécommandé. Si vous considérez un drone comme une machine volante sans pilote, distinguez-le de la charge utile qu'il transporte, qui est

généralement une caméra ou un autre appareil. Le drone n'est pas un outil de mesure, mais il est conçu de manière à tirer le meilleur parti des outils qui y sont attachés [1].

I.2.2.1 Drones à voilure fixe

Pour les drones à voilure fixe, la capacité à contrecarrer la gravité est assurée grâce à la présence d'une ou de plusieurs ailes rigides. C'est le profil spécifique des ailes qui produit la force portante lorsque l'appareil est soumis à un vent relatif (3). Les drones en forme d'avion standard appartiennent à cette catégorie, ainsi que toute une série d'appareils aux formes plus originales. Quand les ailes sont indiscernables du corps de l'avion, on parle généralement d'une 'aile volante' (voir **figure I.1**).

Les drones à voilure fixe sont capables de voler sur de grandes distances et sont très bien adaptés aux tâches de cartographie. La nécessité de maintenir une vitesse de déplacement minimale et la maîtrise limitée de leurs mouvements ne permettent cependant pas d'effectuer des manœuvres précises autour d'objets. De plus, leur décollage nécessite une vitesse horizontale initiale : il faudra donc les lancer afin d'initier leur vol [1].



Figure I.1 : Drone dit 'aile volante' [1].

I.2.2.2 Drones à voilure tournante

Dans le cas de drones à voilure tournante, le maintien de l'appareil dans les airs est assuré par un ou plusieurs rotors. Chaque hélice, dont le corps est parallèle au sol, exerce une force verticale sur l'air lorsqu'elle se met en rotation. On trouve dans cette catégorie des appareils similaires aux hélicoptères, mais aussi des drones de formes plus spécifiques. L'avantage majeur des drones à voilure tournante réside dans leur aptitude à maintenir un vol stationnaire, qui permet

de prendre des photographies dans de meilleures conditions de stabilité, et donc de multiplier les angles de vue sur un objet d'intérêt.

Les hélicoptères sont bien connus dans le monde de l'aviation traditionnelle. Dans ce type d'appareil, c'est un rotor unique (dit 'rotor principal') qui maintient l'appareil dans les airs. Afin d'éviter que l'appareil ne se mette à tourner sur lui-même lorsque le rotor principal tourne, par le principe de l'action et de la réaction, il est indispensable de prévoir un rotor de queue (ou rotor 'anticouple' – voir **figure I.2**) [1].



Figure I.2 : L'hélicoptère est un sous-type de drone à voilure tournante. Le rotor de queue évite que l'appareil ne se mette à tourner sur lui-même [1].

Les multicoptères, aussi appelés '*multirotors*' ou '*multicopters*' en anglais, sont très populaires. Ils sont munis de plusieurs bras au bout desquels sont fixées des hélices identiques. Le nombre d'hélices et leur configuration peuvent être très variés, comme indiqué à la figure 7. Cette figure montre également qu'il y a toujours une alternance du sens de rotation des hélices.

En effet, si toutes les hélices tournaient dans le même sens et à la même vitesse, l'appareil entrerait naturellement en rotation autour de son centre de masse [1].

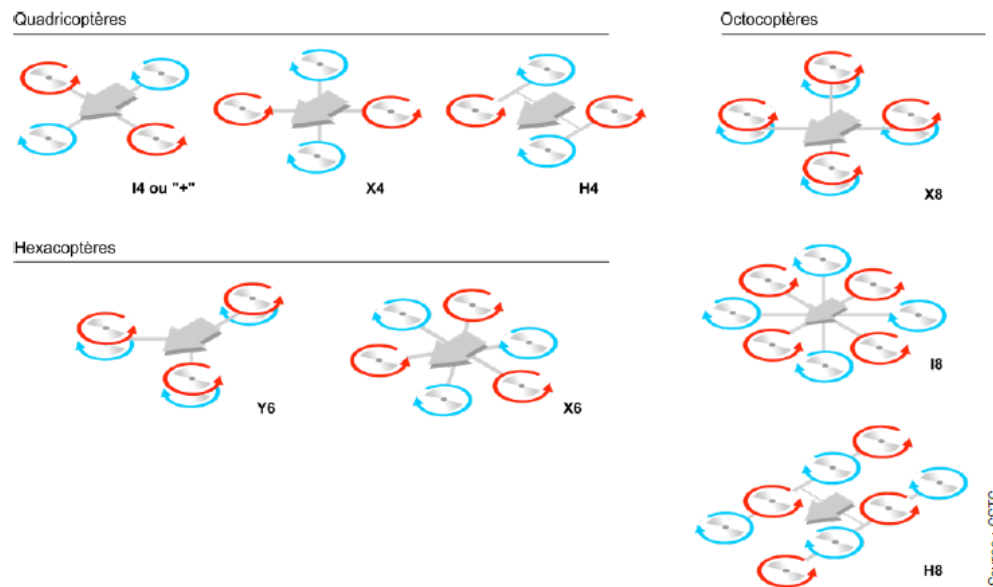


Figure I.3 : Configurations fréquentes des hélices des multicoptères. Le sens de rotation des hélices est indiqué par une flèche colorée.

I.2.3 Domaines d'applications

I.2.3.1 Drone militaires

La famille des drones militaires est très vaste⁴. Elle peut se subdiviser en plusieurs catégories dont nous résumerons ici l'essentiel. En règle générale, les drones militaires diffèrent selon leur vitesse, leur rayon d'action ainsi que leurs fonctions. Prenons à titre d'exemple les mini-drones qui seront utilisés pour des missions d'observation ; les drones tactiques appelés TUAV (Tactical Unmanned Air Vehicle) ; les drones de combat appelés UCAV (Unmanned Combat Air Vehicle) ou les drones de transport, en cours de préparation, qui seront utilisés pour le transport de troupes au combat.



Figure I.4 : Drone militaire.

Les drones militaires ont toujours présenté une alternative intéressante aux avions de combat qui nécessitent plus de moyens humains et financiers. Le drone est discret, rapide et sa masse moins importante. En outre, sa perte s'avère moins lourde qu'un avion et son pilote [2].

I.2.3.2 Drone civil

Le drone pour application civile est un aéronef autonome ou téléopéré. Il va se présenter sous deux formes. Le drone peut être multirotors ou de voiture fixe. Les drones multirotors sont capables de rester immobiles ou de voler à de très faible vitesse dans l'air, ce qui les rend particulièrement utiles pour effectuer par exemple des photos aériennes ou des vidéos. En revanche, les drones à voiture fixe sont utiles pour couvrir de longues distances ou atteindre des hautes altitudes. Ils sont par exemple utilisés pour des missions de sauvetage ou pour des applications topographiques.

Ce qui caractérise enfin le drone à application civile, c'est le capteur qu'il pourra emporter. Le capteur est très souvent une caméra. Cela peut aussi être un appareil photo, voire un compteur Geiger ou un sonomètre. La tendance actuelle est bien entendu de pouvoir embarquer des marchandises d'un point A à un point B ou un émetteur lumière ou wifi pour couvrir une zone particulièrement inaccessible. La possibilité de localiser avec précision (en utilisant un GPS) dans le ciel des points de mesure, est aussi une qualité très recherchée pour beaucoup d'applications [2].

Les applications des drones civils sont multiples et variées. Citons la cartographie, la logistique, la sécurité publique et privée, les prises de vues aériennes (photographies ou vidéos), l'inspection de sites (ex. éoliennes, centrales nucléaires, ...), l'observation d'environnements marins, les secours (ex. : détection de victimes d'avalanches), la thermographie, la recherche agronomique et bien d'autres encore [2].



Figure I.5 : Drone civil. [2]

I.3 Quadrirotor

I.3.1 Description

Le quadrirotor est un objet volant à quatre rotors placés aux extrémités d'un corps rigide en forme de croix. L'électronique de commande est généralement située au centre de la croix qui représente le centre de gravité de la machine. Pour empêcher le quadricoptère de tourner autour de l'axe de lacet, il faut que les hélices appartenant aux paires de moteurs avant et arrière tournent dans un sens et que les hélices appartenant aux paires gauche et droite dans l'autre sens. Pour garder le quadrirotor stable, il est nécessaire de développer des algorithmes qui permettent de contrôler chaque moteur séparément pour compenser l'inclinaison sur chaque axe et ainsi le stabiliser [3].

I.3.2 La technologie

Avant de commander n'importe quel système il est nécessaire de bien définir ces composants afin de pouvoir bien le modéliser. Comme tout autre robot, quadrirotor se compose

principalement d'une partie mécanique qui forme son squelette et ses muscles, ainsi que d'une partie électronique qui rassemblant les capteurs, le calculateur et les modules de communications [3].

I.3.3 Avantages de la configuration quadrirotor

La conception du quadrirotor offre de réels avantages par rapport à d'autres configurations[3] :

- Leurs tailles réduites et leur manœuvrabilité leur permettent de voler dans des environnements fermés (Indoor) ou ouverts (Outdoor) et près des obstacles à l'opposition des hélicoptères classiques.

- La simplicité de sa mécanique facilite sa maintenance.

- Aucun embrayage n'est exigé entre le moteur et le rotor et aucune exigence n'est donnée sur l'angle d'attaque des rotors.

- Quatre petits rotors remplacent le grand rotor de l'hélicoptère ce qui réduit énormément l'énergie cinétique stockée et minimise les dégâts en cas d'accidents.

- Son décollage et atterrissage verticaux.

- Cette configuration est commandée en variant seulement la vitesse de rotation des quatre moteurs.

- Leur capacité de portance à cause de la présence de quatre rotors au lieu d'un qui peut être augmenté en rallongeant les pales d'un rotor ou en augmentant leur nombre, mais à cause de phénomènes aérodynamiques et d'encombrement, cela à des limites.

- Réduction de l'effet gyroscopique.

- Sa dynamique est plus faible que celle de l'hélicoptère ce qui ne nécessite pas un temps de réaction rapide.

Toutefois, le quadrirotor possède pas mal d'inconvénients qui l'empêchent encore d'être en tête du peloton des UAV's, on peut citer :

- Malgré ses quatre rotors et son équilibre stationnaire, le X4 reste un système sous actionné et dynamiquement instable.

- Le rapport portance-poids reste faible, bien que la portance est avantagée par les quatre rotors, le poids du quadrirotor à vide augmente forcément avec, ce qui résulte en le

fait qu'un grand pourcentage de la portance gagnée est utilisée pour soulever le poids du véhicule même.

- Leur technologie reste émergente, ce qui n'encourage pas les industriels pour investir dessus.

La modélisation 3D est une technique qui consiste à reconstituer en 3 dimensions les d'objets, pour produire ces modélisations 3D, nous combinons différentes techniques et outils. Le drone est un outil particulièrement apprécié pour les modélisations 3D car il permet facilement d'aller capturer la donnée essentielle pour une parfaite reconstruction. [4].

Dans notre travail, on va utiliser le drone quadrirotor pour prendre des images à l'aide d'un camera intégrer ainsi relever des points d'intérêts utilisant la méthode SIFT et relever le nuage des points et puis les reconstruire avec un logiciel de reconstruction 3D pour obtenir des modèles 3D à partir de ces images.

I.4 La photogrammétrie

I.4.1 Introduction

La photogrammétrie est utilisée depuis plus d'un siècle comme instrument de mesure géométrique, essentiellement dans le domaine de la cartographie, ouaille est la base de la quasi-totalité des cartes topographiques du monde. Pendant tout le vingtième siècle, les technologies ont relativement peu évolué. Elles sont restées très onéreuses et confinées dans des grands organismes, instituts géographiques civils ou militaires, ou sociétés privées de topographie. Depuis l'an 2000, trois facteurs ont complètement modifié les possibilités d'usage de la photogrammétrie [6]:

- Le plus important a été la généralisation de l'image numérique de bonne qualité, à des coûts toujours plus bas. Le handicap principal des techniques analogiques était en effet le support des images, plaques de verre ou film de grand format, qui rendait indispensable un système de visualisation optico-mécanique complexe et délicat. Même l'apparition des restituteurs numériques dans les années 80 n'a guère amélioré cette situation ;

- L'augmentation constante de la puissance de calcul des ordinateurs, qui permet de réaliser des opérations complexes en des temps très courts sur des grandes quantités de

données-la généralisation dans le domaine universitaire de logiciels libres a mis à la disposition de tout chercheur des outils extrêmement performants.

I.4.2 Historique

La photogrammétrie a fortement évolué depuis sa première application réalisée par un officier de l'armée française, Aimé Laussedat en 1849 sur la façade de l'Hôtel des Invalides : il a eu en effet l'idée d'utiliser des photographies de paysages non seulement pour observer le terrain mais aussi pour le mesurer. Il met ainsi au point une technique qui s'appellera plus tard la photogrammétrie. Laussedat devient professeur au CNAM en 1873, titulaire de la chaire de géométrie appliquée aux arts où il est souvent question de topographie, et celui que l'on peut considérer comme l'inventeur de la photogrammétrie sera même directeur du Conservatoire de 1881 à 1900.

L'autre personnage-clé est Félix Tournachon, plus connu sous le nom de Nadar, qui dans les années 1860 monte en ballon pour photographier Paris et bien d'autres villes, et qui a très bien compris, à en juger par les brevets qu'il a déposés, l'intérêt de la photographie aérienne pour des finalités tant civiles que militaires.

Inventée en France, la photogrammétrie a ensuite été développée et industrialisée de façon très complète en Allemagne. En pratique, il faut attendre l'entre-deux-guerres pour que la photographie aérienne se généralise, avec le développement de l'aviation.

Après-guerre, on observe un emploi de plus en plus systématique de la photogrammétrie pour réaliser les cartes de base de pays entiers.

Les développements de l'imagerie spatiale à haute résolution et de la puissance de l'informatique grand public ont donné depuis peu de nouvelles impulsions à ce domaine. Mais en parallèle, les développements au sein de la communauté de vision par ordinateur, destinés essentiellement à des applications de robotique, ont capitalisé depuis une décennie l'essentiel des efforts de recherche dans ce même domaine. C'est cette communauté qui, actuellement, est principalement porteuse de l'avenir de la photogrammétrie.

Le principe général est basé sur la perception humaine du relief par observation stéréoscopique. Pour le cas de la photogrammétrie aérienne, un avion équipé d'une chambre de prise de vues vole au-dessus d'une région, de façon qu'une partie du terrain figure sur deux clichés correspondant à deux positions différentes de l'avion.

Si on observe simultanément un cliché avec un œil et le second avec l'autre œil grâce à un outil optique approprié (stéréoscope à miroirs, appareil de restitution, ordinateur équipé de lunettes spéciales, etc.), on voit en relief la zone de terrain vue sur les deux images.

La vision humaine permet en effet de voir en relief dans une large gamme de dispositions relatives de ces deux images. Mais si nous disposons ces dernières dans une position relative exactement semblable à celle qu'elles avaient au moment de la prise de vue, alors l'image stéréoscopique observée est une exacte homothétie du terrain réel photographié, pour autant que la chambre de prise de vue soit parfaite (c'est-à-dire n'apporte aucune distorsion à l'image, on l'appelle alors chambre "métrique"), ou que l'image ait été corrigée de sa distorsion. Pour exploiter alors cette scène stéréoscopique, l'appareil de restitution superpose à chaque image un point (le "ballonnet"), que la vision humaine comprendra comme un petit objet dont la position est déplaçable à volonté en hauteur au-dessus de l'image du terrain grâce à des commandes appropriées. L'opérateur aura donc pour travail de promener ce ballonnet dans l'image sur tous les objets à mesurer, pendant que l'appareil archivera toutes les informations numériques produites.

Pour que l'image observée soit une copie exacte de l'objet mesuré, il faut contraindre un certain nombre de points dans l'image en les obligeant à être à des positions relatives similaires aux leurs sur l'objet. Pour un couple stéréoscopique donné, on montre qu'il faut 6 points connus pour que l'image soit fidèle. Ces points seront mesurés : cette opération est appelée stéréopréparation. Lorsque de nombreux couples stéréoscopiques sont enchaînés (bande de clichés aériens), on peut limiter le nombre de points terrain à mesurer en analysant toutes les contraintes géométriques qui se transmettent de cliché à cliché. Le processus de calcul, très complexe, s'appelle aérotriangulation. Par ailleurs, la manipulation des grandes quantités de données numériques extraites est résolue par des logiciels spécialisés, outils permettant la mise en forme finale des données sorties de l'appareil, d'entrée des corrections en provenance des équipes de terrain (qui complètent les levés de toutes les informations non visibles sur les clichés et corrigent

les points douteux, phase dite de complètement), et enfin de formatage et d'édition des données selon les besoins du client [7].

I.4.3 Reconstruction 3D

Les technologies de relevé géométrique ont fortement évolué ces dernières années. Elles permettent de numériser automatiquement des objets existants et d'en produire une représentation en très haute résolution (reproduction de détails fins et de couleurs). Le travail de modélisation s'en trouve grandement simplifié ; il n'est même parfois plus nécessaire.

On distingue deux grandes familles de techniques de relevé 3D en haute définition utilisables en combinaison avec un drone : les techniques de scanning 3D à proprement parler, où la géométrie d'un objet est mesurée en temps réel par des systèmes très performants comme les scanners laser, et les techniques de reconstruction photogrammétrique, consistant à déterminer la géométrie de l'objet à partir de photos grâce à un logiciel. Le lecteur intéressé trouvera un aperçu des évolutions récentes en la matière dans la monographie du CSTC intitulée 'Le relevé 3D à l'heure du BIM. Capturer la réalité en haute définition [1].



Figure I.6 : Numérisation 3D réalisée par photogrammétrie [1].

Un nuage de points est un type de fichier fréquemment obtenu lorsque l'on fait appel à une méthode de relevé 3D à haute définition. Les objets scannés y sont reproduits sous la forme d'une multitude de points localisés dans l'espace et suffisamment rapprochés pour que l'on puisse distinguer des détails très fins. Un autre type courant de fichier 3D brut est le maillage triangulé ou mesh. L'objet y est représenté par un ensemble de faces triangulaires interconnectées. Au regard de la nature très différente de ces deux types de modèles, l'information de couleur y sera encodée de deux manières distinctes. Dans un nuage de points, une couleur (une valeur RVB) est attribuée à chaque point composant le modèle. Dans le cas d'un maillage, la surface de chaque face triangulaire doit être colorée ; cette opération est réalisée au moyen d'une texture. La texture consiste en une image que l'on colle sur le modèle. La figure 2 montre les distinctions entre ces deux types courants de fichiers 3D bruts issus de relevés en haute définition [1].

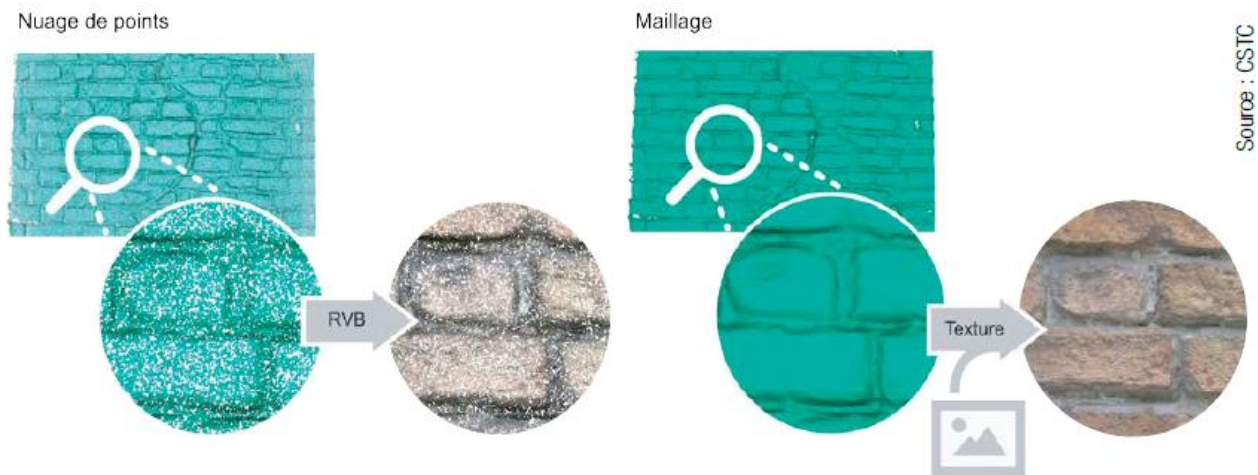


Figure I.7 : Les techniques de relevé géométrique à haute définition produisent des fichiers 3D bruts sous forme soit de nuages de points, soit de maillages [1].

I.4.4 Domaines d'application de la reconstruction 3D

Les applications pouvant bénéficier de la reconstruction 3D, sont toutes les applications usuelles de la synthèse d'images : CAO1, Réalité Virtuelle2, Réalité Augmentée, entraînement, simulation, commerce, loisirs, compression de données, et beaucoup de recherches sont en cours dans ces sens [8].

I.4.4.1 Entraînement

Pour toutes sortes d'interventions en milieux hostiles : nucléaire, militaire, spatial ; il est nécessaire de préparer et d'entraîner les hommes à évoluer dans leur futur milieu d'action ; en les plongeant dans une situation simulée le plus fidèlement possible, ils apprennent donc à repérer la topographie du site et de se familiariser à la géographie du lieu à l'aide d'une représentation graphique tridimensionnelle réaliste.

I.4.4.2 Simulation

L'intérêt des simulations a été démontré dans des contextes industriels, parfois de façon spectaculaire. Et comme il est fastidieux de produire de tels modèles manuellement, des systèmes automatiques de génération à partir d'images ont ici grand intérêt.

I.4.4.3 Compression de données

Calculer de nouvelles vues à partir de quelques images d'une scène, a une application immédiate en compression de données. Le principe est de calculer une représentation géométrique grossière des objets filmés, ainsi que les transformations qu'ils subissent : changements de position, ou de point de vue.

I.4.4.4 Imagerie médicale

Depuis de nombreuses années, la manipulation d'images médicales a véritablement aidé le chirurgien dans la pratique de son art. L'utilisation de caméras vidéo, d'ordinateurs, et d'appareils de mesure sophistiqués pour préparer et réaliser les interventions, a permis d'améliorer le confort des patients, de diminuer les complications et de faire d'importantes économies de santé publique. La complexité des données 3D, en imagerie médicale nécessite le développement de modèles intégrant de multiples connaissances et de grandes quantités d'informations, et la synthèse d'images à partir d'images est donc complètement adéquate.

I.4.4.5 Commerce

Avec Internet s'est développée la possibilité d'effectuer des transactions financières à distance, et de pratiquer le commerce électronique. Alors, les catalogues agrémentés de photos et de visualisations plus ou moins animées ou interactives, permettent de bien visualiser puis de choisir les produits.

I.4.4.6 Conservations

La capture automatique de modèles à partir d'images, peut être utilisée pour la restitution de maquettes numériques d'édifices ou d'objets d'antiquité. Ceci peut simplifier la navigation et les interactions avec les sources documentaires qui lui sont associées. Ces informations et les représentations géométriques multiples des objets constituant l'édifice peuvent être restituées de manière dynamique en fonction de différents points de vue utilisateurs.

I.4.4.7 Autres applications

On peut encore citer le domaine des loisirs, et de toutes les applications exigeant un grand nombre d'images : visite de musées virtuels, visualisation de maisons ou d'équipements ménagers pour la vente, de lieux de vacances dans une agence de voyage, jeux vidéo.

I.4.5 Principe de fonctionnement

Le principe général est basé sur la perception humaine du relief par observation stéréoscopique. Pour le cas de la photogrammétrie aérienne, un avion équipé d'une chambre de prise de vues vole au-dessus d'une région, de façon qu'une partie du terrain figure sur deux clichés correspondant à deux positions différentes de l'avion.

Si on observe simultanément un cliché avec un oeil et le second avec l'autre oeil grâce à un outil optique approprié (stéréoscope à miroirs, appareil de restitution, ordinateur équipé de lunettes spéciales, etc.), on voit en relief la zone de terrain vue sur les deux images. La vision humaine permet en effet de voir en relief une large gamme de dispositions relatives de ces deux images. Mais si nous disposons ces dernières dans une position relative exactement semblable à celle qu'elles avaient au moment de la prise de vue, alors l'image stéréoscopique observée est une exacte homothétie du terrain réel photographié, pour autant que la chambre de prise de vue soit parfaite

(c'est-à-dire n'apporte aucune distorsion à l'image, on l'appelle alors chambre "métrique"), ou que l'image ait été corrigée de sa distorsion. Pour exploiter cette scène stéréoscopique, l'appareil de restitution superpose à chaque image un point (le "ballonnet"), que la vision humaine comprendra comme un petit objet dont la position est déplaçable à volonté en hauteur au-dessus de l'image du terrain grâce à des commandes appropriées. L'opérateur aura donc pour travail de promener ce ballonnet dans l'image sur tous les objets à mesurer, pendant que l'appareil archivera toutes les informations numériques produites.

Pour que l'image observée soit une copie exacte de l'objet mesurée, il faut contraindre un certain nombre de points dans l'image en les obligeant à être à des positions relatives similaires à celles de l'objet. Pour un couple stéréoscopique donné, on montre qu'il faut 6 points connus pour que l'image soit fidèle. Ces points seront mesurés : cette opération est appelée stéréopréparation. Lorsque de nombreux couples stéréoscopiques sont enchaînés (bande de clichés aériens), on peut limiter le nombre de points terrain à mesurer en analysant toutes les contraintes géométriques qui se transmettent de cliché à cliché. Le processus de calcul, très complexe, s'appelle triangulation. Par ailleurs, la manipulation de grandes quantités de données numériques extraites est résolue grâce à divers algorithmes et méthodes de traitement d'images. Ceux-ci permettent la mise en forme finale des données en sorties de l'appareil [10].

Le schéma fonctionnel de l'algorithme de reconstruction 3D est illustré par la figure I.9 [9] :

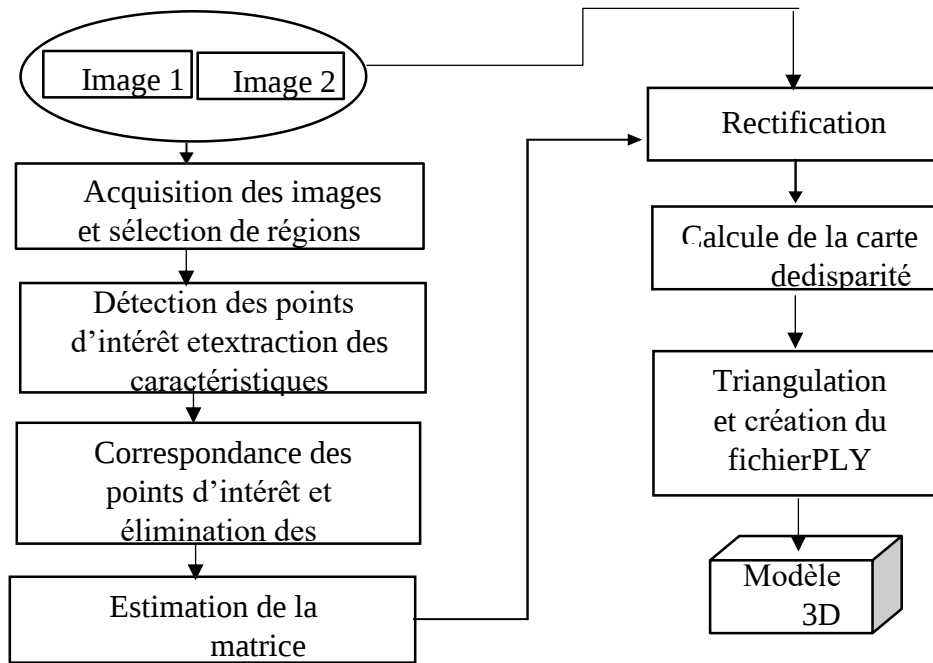


Figure I.8 : Organigramme fonctionnel de la reconstruction 3D

I.4.5.1 Acquisition des images et sélection des régions d'intérêt

Après avoir acquis le couple d'images, l'opérateur sélectionne l'objet ou la région que nous souhaitons reconstruire à partir des deux images. Une partie de chaque image est isolée centrée sur l'objet/région d'intérêt respectif (nous appelons chaque partie P1 et P2, respectivement.). P1 et P2 doivent être riches en caractéristiques facilement détectables (comme les coins et les bords). Par conséquent, dans le cas d'images très texturées, elles peuvent être de petite taille. Dans le cas d'images à texture fine, la taille de la région d'intérêt doit être grande. En P1 et P2, diverses transformations sont utilisées pour détecter les points les plus informatifs (on parlera des points clés par KP).

I.4.5.2 Détection des points d'intérêt et extraction des caractéristiques

Les caractéristiques de chaque KP sont extraites et paramétrées sous la forme d'un vecteur, via les différents descripteurs tels que SIFT, SURF ou ORB. La détection des points clés KP influence de façon considérable les étapes suivantes (l'appariement des KP affecte directement le calcul de la matrice essentielle qui est l'opération pivot de l'ensemble de l'algorithme de reconstruction).

- **SIFT (Scale Invariant Feature Transform)**

SIFT effectue l'extraction des patches (régions d'intérêt) dans une première image qui peut être comparée aux autres patches d'une seconde image indépendamment de la rotation, de la translation et de l'échelle. Comme un détail pertinent n'existe qu'à une certaine échelle, les patches extraits sont centrés sur des points d'intérêt stables. L'idée clé est que, dans une certaine mesure, on peut utiliser l'invariance SIFT pour traiter les transformations d'image qui se produisent lorsque les points de vue changent pendant l'acquisition d'image. L'image est représentée à différentes échelles. SIFT calcule les maxima de la représentation laplacienne à chaque échelle, en utilisant un filtrage par Gaussiennes pour supprimer le bruit. Ces maxima correspondent à des points d'intérêt. Puis pour chaque maximum, un patch est échantillonné. L'origine est le maximum et la direction est le gradient dominant à l'origine.

Pour chaque point clé, une description de ces patches est associée.

Les données fournies initialement par le détecteur sont les coordonnées des points d'intérêt ainsi que leur échelle caractéristique. Une étape préliminaire consiste à construire l'histogramme des orientations locales définies en chaque point (x) par :

$$\theta(x) = \arctg \left(\frac{L(x, y + 1) - L(x, y - 1)}{L(x + 1, y) - L(x - 1, y)} \right)$$

Cet histogramme se compose de trente-six intervalles (36 classes), couvrant chacun un angle de dix degrés. Ce dernier est pondéré d'une part par un filtre gaussien d'écart type égal à une fois et demie la valeur de l'échelle locale et d'autre part par l'amplitude m de chaque point défini par :

$$m(x, y) = \sqrt{((L(x + 1, y) - L(x - 1, y))^2 + ((L(x, y + 1) - L(x, y - 1))^2}$$

La description, qui est généralement stockée sur 128 bits, consiste en une statistique de gradients calculés dans des régions autour du point clé. La taille de la région est déterminée par l'échelle des points clés et l'orientation est déterminée par l'axe dominant. Comme le nombre d'entités extraites peut varier beaucoup en raison de la variabilité de la complexité des textures (d'une image à une autre ou dans différentes parties de l'image), une étape de post-filtrage permet de contrôler le nombre de caractéristiques extraites dans des limites raisonnables (par exemple

entre un et dix milliers par image). Nous utilisons un filtrage pour assurer une bonne répartition dans l'image [31, 32, 33].

- **SURF (Speeded-Up Robust Features)**

Surf s'appuie sur le déterminant de la matrice Hessienne pour sélectionner l'échelle ainsi que l'emplacement des points d'intérêt. Étant donné un pixel p , l'Hessienne de p est :

$$H(f(x, y)) = \begin{bmatrix} \frac{\partial^2 f}{\partial x^2} & \frac{\partial^2 f}{\partial x \partial y} \\ \frac{\partial^2 f}{\partial x \partial y} & \frac{\partial^2 f}{\partial y^2} \end{bmatrix}$$

SURF est basé sur une approximation du filtrage gaussien, permettant de diminuer considérablement les temps de calculs. Il est donné par l'équation (Eq-I.4), que l'on obtient pour un point x sur l'échelle σ :

$$\mathcal{H}(x, \sigma) = \begin{bmatrix} L_{xx}(x, \sigma) & L_{xy}(x, \sigma) \\ L_{xy}(x, \sigma) & L_{yy}(x, \sigma) \end{bmatrix}$$

Où $L_{xx}(x, \sigma)$ est la convolution avec la dérivée gaussienne du second ordre avec l'image I au point x , et de même pour $L_{xy}(x, \sigma)$ et $L_{yy}(x, \sigma)$. Les gaussiennes sont optimales pour l'analyse échelle-espace, mais en pratique, elles doivent être rognées et discrétisées.

Afin de garder la cohérence dans le filtrage, un filtre approximé initial de taille 9×9 correspondant à un filtre gaussien d'écart type $\sigma = 1,2$ est utilisé. Enfin, afin de gérer la multi-échelle, SURF s'appuie sur un ensemble de masques de tailles croissantes (9×9 , 15×15 , 21×21 , 27×27 , ...) dépendant de l'écart type de la gaussienne à approximer.

- **ORB (Oriented Fast and Rotated Brief)**

ORB fonctionne aussi bien que SIFT dans la tâche de détection de caractéristiques. Il est deux fois plus rapide que SIFT. Il est deux fois plus rapide que SIFT. ORB est basé sur le célèbre détecteur de points clés FAST et le descripteur Brief. Ces deux techniques sont intéressantes en raison de leurs bonnes performances et de leur faible coût. Les principales contributions d'ORB sont les suivantes :

- L'ajout d'un composant d'orientation rapide et précis à FAST
- Le calcul efficace des caractéristiques BRIEF orientées.
- Analyse de variance et corrélation de caractéristiques BRIEF orientées.

- **Fast (Features from Accelerated and Segments Test)**

Étant donné un pixel p , la luminosité de p est comparée à celle des 16 pixels environnants qui sont dans un petit cercle autour de p . Les pixels du cercle sont ensuite triés en trois classes (plus clair que p , plus foncé que p ou similaire à p). Si plus de 8 pixels sont plus sombres ou plus clairs que p , ils sont sélectionnés comme point clé. Ainsi, les points clés trouvés par FAST nous donnent des informations sur l'emplacement des contours déterminants d'une image. (Détecteur de coins)

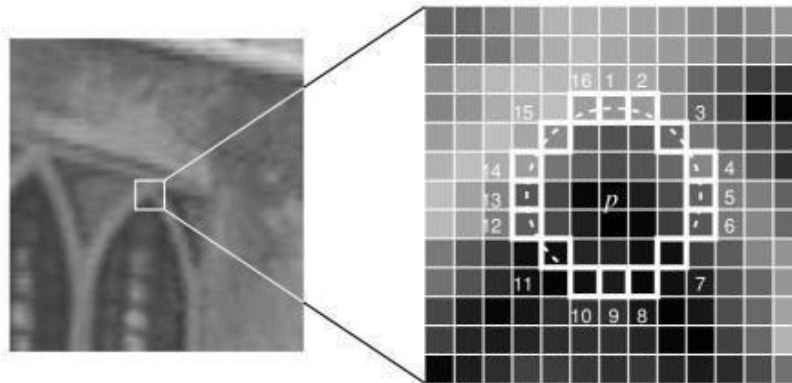


Figure I.9 : Tableau des pixels de FAST

Cependant, les entités FAST n'ont pas de composant d'orientation et d'entités multi-échelles. L'algorithme ORB utilise donc une pyramide d'images à plusieurs échelles. Une pyramide d'images (Pyramide Laplacienne). Chaque niveau de la pyramide contient la version sous-échantillonnée de l'image par rapport au niveau précédent. Une fois que la pyramide d'images a été créée, ORB utilise l'algorithme FAST pour détecter les points clés dans l'image. En détectant les points clés à chaque niveau, l'ORB localise **efficacement les points clés à une échelle différente.**

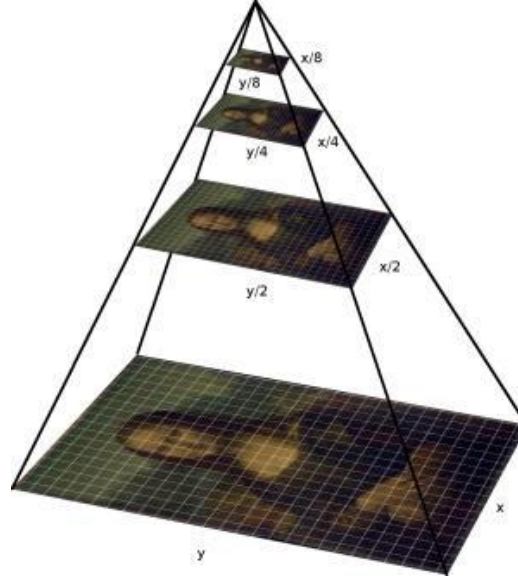


Figure I.10 : Pyramide La placienne générée par ORB

Après avoir localisé les points clés, une orientation à chaque point clé est attribuée. Pour détecter le changement d'intensité, ORB utilise le centre de gravité. Le centroïde d'intensité suppose que l'intensité d'un coin est décalée de son centre, et ce vecteur peut être utilisé pour attribuer une orientation.

Le centroid ou centre de masse est définis par l'équation suivante :

$$C = \left(\frac{m_{10}}{m_{00}}, \frac{m_{01}}{m_{00}} \right)$$

Avec m les moments des patchs de points clés calculés par :

$$m_{pq} = \sum_{x,y} x^p y^q I(x,y)$$

L'orientation du patch est donnée par l'expression suivante :

$$\theta = \text{atan2}(m_{01}, m_{10})$$

Puis le descripteur est calculé par FAST.

- **Brief (fonction élémentaire indépendante robuste binaire) :**

Brief prend tous les points clés trouvés par FAST et les convertit en un vecteur de caractéristiques binaires représentant un objet. Le vecteur de caractéristiques binaires, également appelé descripteur de caractéristiques binaires, est un vecteur de caractéristiques qui ne contient que des 1 et 0. En bref, chaque point clé est décrit par un vecteur de caractéristiques constitué d'une chaîne de 128 à 512 bits.

Brief débute par le lissage de l'image à l'aide d'un noyau gaussien afin d'éviter que le descripteur soit affecté par le bruit. Ensuite, une paire aléatoire de pixels dans un voisinage défini autour de ce point clé est sélectionnée. Le voisinage défini autour du pixel constitue la région d'intérêt. Le premier pixel de la paire aléatoire est tiré d'une distribution gaussienne centrée autour du point clé avec une déviation sigma. Le deuxième pixel de la paire aléatoire est tiré d'une distribution gaussienne centrée autour du premier pixel avec un écart type ou un étalement de sigma par deux. Maintenant, si le premier pixel est plus lumineux que le second, il attribue la valeur 1 au bit correspondant sinon 0.



Figure I.11 : La sélection de pixel par ORB

I.4.5.3 Correspondance des points d'intérêt et élimination des correspondances non fiables

Les Points d'intérêt analogues dans les deux portions d'image sont mis en correspondance, en recherchant, pour chaque vecteur de caractéristiques de la première image, le vecteur de caractéristiques le plus proche de la deuxième image. Les points d'intérêt appariés dont les vecteurs de caractéristiques sont plus éloignés qu'un certain seuil sont supprimés. Le seuil est fixé entre 0,9

et 1,5 fois la distance moyenne sur l'ensemble des couples des vecteurs de caractéristiques des points d'intérêt analogues, en fonction du nombre total de points appariés. Cette étape nous permet non seulement d'affiner les correspondances trouvées, mais aussi de réduire le temps de calcul de l'étape suivante, c'est-à-dire l'élimination des valeurs aberrantes, puisqu'il sera réalisé sur moins de couples de points. Pour supprimer les valeurs aberrantes, le test randomisé RANSAC avec rapport de probabilité séquentielle est utilisé, car il est l'une des méthodes plus rapides.

I.4.5.4 Estimation de la Matrice essentielle et Rectification

Avec les points appariés restants en entrée, l'algorithme donne en sortie une estimation de la matrice fondamentale (E) basée sur le principe suivant :

Supposons qu'il existe un point p dans le système de coordonnées du monde, la coordonnée est X , son image dans la caméra 1 est x_1 et son image dans la caméra 2 est x_2 , comme suit :

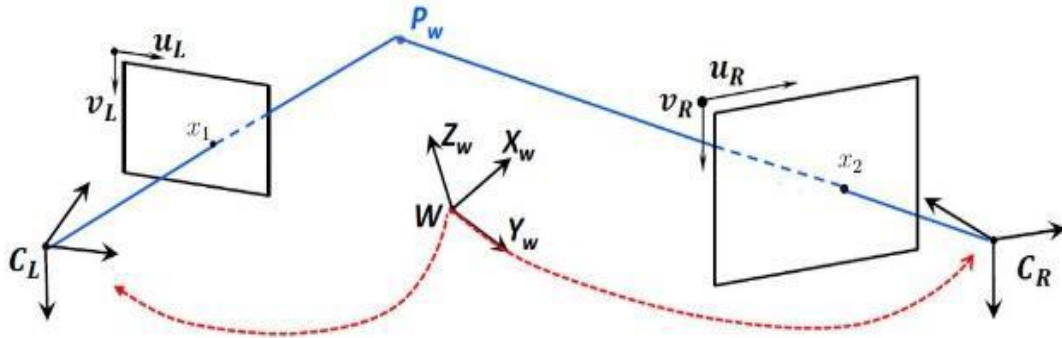


Figure I.12: Coordonnées du monde et de la camera

Supposons que les distances verticales de X aux plans d'image des deux caméras soient respectivement s_1 et s_2 , et que les deux caméras aient la même matrice de paramètres internes K , et que la relation de transformation avec le système de coordonnées mondiales soit $[R_1, T_1]$ et $[R_2, T_2]$, on peut alors obtenir les deux équations suivantes :

$$s_1 x_1 = K(R_1 X + T_1)$$

$$s_2 x_2 = K(R_2 X + T_2)$$

Puisque K est une matrice inversible, on multiplie les deux équations par l'inverse de K pour obtenir :

$$s_1 K^{-1} x_1 = (R_1 X + T_1)$$

$$s_2 K^{-1} x_2 = (R_2 X + T_2)$$

Supposons que :

$$K^{-1} x_1 = x'_1, K^{-1} x_2 = x'_2$$

Puis :

$$s_1 x'_1 = (R_1 X + T_1)$$

$$s_2 x'_2 = (R_2 X + T_2)$$

x'_1 et x'_2 sont les coordonnées normalisées de l'image.

Étant donné que le système de coordonnées mondial peut être sélectionné arbitrairement, nous choisirons le système de coordonnées mondial comme système de coordonnées de la première caméra. La formule ci-dessus devient :

$$s_1 x'_1 = X$$

$$s_2 x'_2 = (R_1 X + T_2)$$

En l'écrivant sous la deuxième forme :

$$s_2 x'_2 = s_1 R_2 x'_1 + T_2$$

Le vecteur tridimensionnel est défini par :

$$\hat{T}_2 x'_1 = [T_2]_x x'_2 = \begin{bmatrix} 0 & -t_3 & t_2 \\ t_3 & 0 & -t_1 \\ -t_2 & t_1 & 0 \end{bmatrix}$$

Le Vecteur est perpendiculaire à x'_2 et T_2

En utilisant le vecteur pour faire le produit scalaire des deux côtés de l'équation on obtient :

$$x'_2 \hat{T}_2 R_2 x'_1 = 0$$

D'où la déduction de l'expression de la matrice essentielle à travers l'équation des contraintes épi polaire comme suit :

$$E = \hat{T}_2 R_2$$

La valeur estimée de E est utilisée pour rectifier les deux images, c'est-à-dire de les transformer, par une homographie projective, de sorte que les rangées de la première image soient alignées avec les rangées analogues de la deuxième image. A noter que la rectification peut être précise (jusqu'à un certain degré) uniquement si la taille des photos détecteurs et le décalage entre l'axe optique et le centre de l'image, qui composent les paramètres intrinsèques de la caméra ainsi que la distance focale, soient connus. Pour limiter le temps de calcul, la rectification est effectuée uniquement pour les régions d'intérêt.

I.4.5.5 Calcul de la carte de disparité

La carte de disparité est calculée sur les deux sous-images rectifiées.

La disparité fait référence au décalage horizontal entre les points analogues des deux images considérées. La reconstruction doit être la plus réaliste que possible. Donc, cette étape doit être effectuée pour tous les points des deux images afin d'éviter des « trous » dans le modèle 3D et, ainsi, obtenir une reconstruction fidèle à la réalité.

La correspondance des points est exécutée au moyen de l'algorithme de correspondance de blocs semi-global.

I.4.5.6 Triangulation et création du fichier PLY

Enfin, la reconstruction 3D de la zone d'intérêt est obtenue, à partir de la carte de disparité, par triangulation.

La triangulation fait référence au processus de détermination d'un point dans l'espace 3D compte tenu de ses projections sur deux ou plusieurs images. Afin de résoudre ce problème, il est nécessaire de connaître les paramètres de la fonction de projection des caméras de 3D à 2D pour les caméras concernées, dans le cas le plus simple représenté par les matrices de caméras. La triangulation est parfois aussi appelée reconstruction ou intersection.

L'équation de triangulation est la suivante :

$$x = \frac{b \cdot u}{d}, y = \frac{b \cdot v}{d}, z = \frac{b \cdot f}{d}$$

Où u, v = coordonnées 2D du point à reconstruire, exprimé dans le système d'images.

x, y, z = Coordonnées du point reconstruit, exprimé dans le système de référence 3D choisi.

f = La focale de la camera

b = La distance entre chaque prise

d = La disparité entre chaque prise

L'algorithme crée également un fichier PLY du nuage de points. Ce format permet d'associer les coordonnées de chaque point reconstruit à ses valeurs d'intensité de couleur (RGB). Ce qui rend le modèle 3D plus réaliste. Le format PLY peut être lu par tous les logiciels de visualisation 3D les plus courants, de sorte que le nuage de points peut être facilement présenté et utilisé par l'opérateur.

I.4.5.7 Génération du Modèle 3D

Cette partie a été effectuée en faisant l'importation du nuage de points ainsi que leurs photos correspondantes vers des logiciels de traitement d'images dont on parlera plus en détails dans le chapitre 5.

I.5 Conclusion

Dans ce chapitre nous avons exposé des généralités sur les drones et son domaine d'applications ainsi que le drone quadrirotor, le type qui nous intéresse dans ce travail. Nous avons donné également les différents éléments nécessaires à notre projet. A savoir le principe de base de la photogrammétrie domaines d'application et son fonctionnement.

Le chapitre suivant est consacré à la présentation des différents composants constituant un quadrirotor.

Chapitre II :
Anatomie d'un drone type quadrirotor
et ses composants

Chapitre II :

Anatomie d'un drone type quadrirotor et ses composants

II.1 Introduction

Dans ce chapitre on va présenter les différentes pièces, les composants nécessaires à la construction d'un drone quadrirotor, ceux-ci incluent le système d'alimentation, le contrôleur de vol, les hélices, les capteurs, le transmetteur et son récepteur. Ensuite, on va détailler les mouvements de drone.

II.2 Les composants de notre réalisation

II.2.1 Le châssis

Le châssis est la structure du drone sur laquelle viennent se fixer tous les autres composants, à savoir les pièces électroniques, les moteurs ou encore les différents capteurs. Cette structure peut prendre des formes très différentes selon le type de drone. A puissance égale, un drone constitué de composants plus légers pourra soulever une charge plus importante.

Cela explique la recherche constante de matériaux légers et résistants pour fabriquer le châssis.

Un drone comporte en général des bras sur lesquels sont fixés les moteurs. Ces bras peuvent être attachés mécaniquement à un élément central du châssis ou moulés directement dans un châssis monobloc. La longueur des bras dépendra de l'importance accordée à la stabilité et à la manœuvrabilité du drone. Si l'on privilégie la stabilité, on optera pour des bras longs, alors que des bras courts rendront le drone plus réactif. Pour plus de facilité lors du transport, certains châssis disposent d'éléments repliables (voir figure). Les trains d'atterrissage peuvent être considérés comme faisant partie du châssis. Ils permettent au drone de se poser sans endommager

l'équipement embarqué. Il est également possible de monter un système de train rétractible à distance, afin de ne pas gêner la prise de vue [1].



Figure II.1 : Différents châssis de drones [11].

Dans notre réalisation on a choisi le type des drones appeler 'prise en vue' ou FPV (first person View) ce type offre l'avantage de la stabilité par rapport à d'autre type de véhicule aérien.

Pour construire notre propre multirotor de A à Z. Son assemblage est simple et rapide. Les 4 bras sont vissés entre la plaque inférieure et la plaque supérieure. Ce châssis a une envergure de 450 mm et pèse 272 grammes.

Cette structure permettra d'accueillir toute l'électronique : ESCs, moteurs, contrôleur de vol... Quelques soudures seront à prévoir sur la PCB intégrée dans la plaque inférieure. La propulsion recommandée repose sur des moteurs 2212, des hélices 10x45 en 3S et 8x45 en 4S pour un poids total en vol situé entre 800 et 1600 grammes maximum.

Points forts [12] :

- Châssis F450 polyvalent facile à monter
- Robuste et léger : 272 g seulement
- Propulsion modulable, de 3 à 4S
- 800 à 1600 g en vol

Caractéristiques techniques [12] :

Référence	F450
Dimensions	450 mm
Poids	272 grammes
Couleur	Frame : noir et or/ bras : rouge et noir
Matériau(x)	Bras en ABS
Hélices	10x4.5 en 3S ou 8x4.5 en 4S
Moteurs	2212 et ESCs 30A
Batterie	3 et 4S
Autres spécifications	En kit à monter

Tableau II.1 : Tableau indique les Caractéristiques techniques de Châssis Quadcoptère F450.



Figure II.2 : Châssis F450 [12].

II.2.2 Les moteurs

Le 2212 (BLDC) BRUSHLESS DC MOTOR est un moteur brushless à haute vitesse conçu spécifiquement pour les quadcoptères, les drones ou les avions jouets. Le moteur est de type OUTRUNNER où le boîtier extérieur tourne tandis que l'intérieur reste fixe. C'est l'un des modèles les plus populaires sur le marché en raison de son faible coût. Il est préféré pour les petits drones et avions. Ce modèle est disponible en différentes vitesses et doit être choisi en conséquence [13].

Le nom du moteur sans balais est le suivant : a2212/ 1000KV.

Information	Représente	Informations complémentaires
22	Diamètre du moteur (22mm dans ce cas)	Avec un plus grand diamètre, nous aurons plus de couple. *Choisissez le plus grand
12	Hauteur du moteur (12mm dans ce cas)	*Choisissez le plus grand
1000KV	Nombre de tours effectués par le moteur par volt d'alimentation (1000 tours par volt ici. Si la tension d'alimentation est de 10 volts, le moteur tournera à 10000 tours par minute).	*Choisissez un moteur avec un KV plus élevé pour une meilleure poussée.

Tableau II.2 : Caractéristique de moteur [13].

Un nombre de tours inférieur signifie une vitesse élevée mais un couple inférieur. De même, un nombre plus élevé de tours signifie une vitesse moindre mais un couple plus élevé.

Configuration des pins du moteur Brushless :

Le moteur a trois pins comme indiqué sur la figure. La connexion interne entre trois d'entre elles est indiquée ci-dessous.

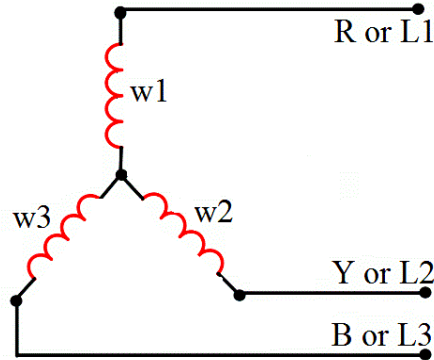


Figure II.3 : Configuration des pins du moteur Brushless

Nom de la pin	Description
R ou L1	Connecté à l'enroulement 1 ou à la phase 1
Y ou L2	Connecté à l'enroulement 2 ou à la phase 2
B ou L3	Connecté à l'enroulement 3 ou à la phase 3

Tableau II.3 : Configuration des pins du moteur Brushless

Caractéristiques et spécifications [14] :

- Tension de fonctionnement : 7,2V à 12V (2 à 3Li-poly ou 6 à 10 NiCad)
- Courant à vide : 0.5Amp
- Courant maximum : 13Amp pour 60Sec
- Puissance maximale : 150 watts
- Poids du moteur : 50-60 grammes

- Température maximale de fonctionnement : + 80°C
- Contrôle complet de la vitesse grâce à la connexion triphasée.
- Longue durée de vie car il n'y a pas de contacts mobiles.

Il n'existe pas de norme industrielle pour ces moteurs sans balais. Les appareils des différentes sociétés ont des valeurs nominales légèrement différentes. Il faut consulter la fiche technique du moteur BLDC pour plus d'informations sur un modèle particulier.

Voici quelques raisons pour lesquelles le moteur 2212 BLDC est préféré aux autres moteurs:

- Le 2212 est utilisé lorsque le coût du projet est faible. Le 2212 est l'un des moteurs les moins chers et les plus robustes du marché. Il est disponible sur tous les marchés électroniques.
- Le 2212 est facile à utiliser, étant donné qu'il existe de nombreux modules de commande pour ce moteur. Avec ces modules, nous pouvons facilement contrôler la vitesse du moteur sans balais.
- Grâce aux connexions triphasées, nous pouvons contrôler la vitesse du moteur de manière plus précise et plus régulière.
- L'efficacité est également l'une des raisons de choisir ce moteur. Le rendement de ce moteur est bon compte tenu de son coût d'achat.

Applications :

- Drones
- QUADCOPTERS
- Robotique
- Avions jouets
- Projets de loisirs

II.2.3 les hélices

Elles sont souvent en matière plastique mais aussi parfois en matériaux composites. Elles sont identifiées par leur diamètre et leur pas indiqués en pouces (ex : 10x4,5). Sur un drone quadrirotor, il faut 2 hélices à sens horaires et 2 hélices à sens anti-horaire [15].

Voici quelques exemples de combinaisons d'hélices et de moteurs :

Moteurs	Hélices	Résultats
KV élevé	Grandes hélices	NON → tire fort sur la batterie et risque de cramer les moteurs.
KV élevé	Petites hélices	OUI → pour du vol acrobatique.
KV faible	Grandes hélices	OUI → pour du vol stable.
KV faible	Petites hélices	NON → pas assez de portance, le drone ne décollera pas.

Tableau II.4 : Combinaisons hélices et moteurs [15].

Structure	Conséquence
Grande hélice	Beaucoup de portance, vol stable mais a besoin de puissance pour faire 1 tour entier → il faut un moteur puissant.
Petite hélice	Moins de portance, vol réactif et a besoin de moins de puissance pour faire 1 tour entier.
Petit pas	Plus grande traction à faible vitesse, mais vitesse maxi limitée.
Grand pas	Plus petite traction à faible vitesse, mais vitesse maxi élevée.

Tableau II.5 : caractéristiques d'une hélice [15].

Bien souvent, les caractéristiques d'une hélice sont présentées sous cette forme : 1045 dans cet exemple, ça signifie simplement une longueur de 10 pouces et un pas 4,5 pouces. Pour du vol acrobatique, on préférera des hélices courtes apportant de la réactivité au drone. Pour du vol stable, on préférera des hélices longues apportant de la stabilité. Ça permettra également de transporter des charges (une caméra par exemple) [15].

Selon la fiche technique précédente (Tableau II.5) du moteur que nous avons choisi, l'hélice parfaite pour notre propulseur est :

Type d'hélice (catégorisation)	Hélice d'avion
Nombre de pales	2 pales
Taille d'hélice	10×45 pouces (25.4cm)
Matière de fabrication	Plastique

Tableau II.6 : La fiche technique d'hélice 10×45.

Les hélices que on a utilisées sont en plastique de taille 10 pouces et un pas de 45 (10×45).



Figure II.4 : Pair d'hélice 10×45 (CW et CCW).

Applications :

- Drones
- QUADCOPTERS

II.2.4 Les Contrôleurs de vitesse électroniques (ESC)

Un contrôleur de vitesse électronique, ou ESC, prend la puissance de la batterie et l'envoie aux bobines du moteur sans balai de manière à le faire tourner à la vitesse commandée à son entrée. Chaque contrôleur contient plusieurs transistors à effet de champ (FET) pour contrôler

l'alimentation et des capteurs pour déterminer la position du moteur pour déterminer la position du moteur, ainsi qu'un contrôleur intégré pour relier le tout oui, chaque ESC a son propre petit processeur, et dans certains cas, vous pouvez réellement mettre à jour le code, bien que nous n'en ayons pas besoin [16].

Un ESC sert à contrôler la vitesse de rotation d'un moteur brushless.

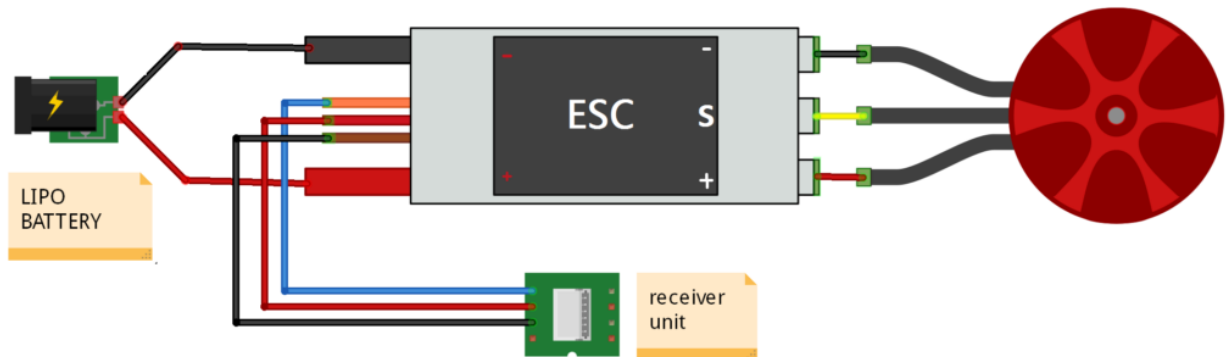


Figure II.5 : Branchement d'un ESC avec moteur brushless.

Avantages :

- Programmable à l'aide d'une carte de programmation ou d'un émetteur
- Réglage linéaire de l'accélérateur très doux et précis
- Supporte les moteurs à haut régime.
- Utilise des MOSFETs plus petits pour minimiser le poids.
- Perte de chaleur minimale

Anatomie d'un ESC :



Figure II.6 : Anatomie d'un ESC.

Comme on peut le voir sur la figure ci-dessus, un ESC possède un connecteur 3 broches (parfois, seulement deux) sur lequel on retrouve

- Une broche de masse, ici le fil noir : à mettre en commun avec les masses des autres composants
- Une broche signal, ici le fil blanc : c'est une broche d'entrée sur laquelle on envoie le signal de commande de l'ESC
- Une broche d'alimentation (optionnelle), ici le fil rouge : cette broche fournit une alimentation +5V qui peut servir à alimenter d'autres composants (un récepteur RF par exemple). Ça implique que l'ESC embarque un régulateur de tension en interne, ce qui n'est pas toujours le cas.

Du même côté que ce connecteur, on retrouve deux fils plus épais, un rouge et un noir. On devine rapidement qu'il s'agit de l'alimentation de l'ESC, à brancher directement sur l'alimentation générale du drone.

Enfin, de l'autre côté, trois fils, de la même couleur. Ces fils sont à brancher sur le moteur à contrôler.

Le choix des ESC est assez simple et dépend beaucoup de votre budget et de la qualité désirée. Les variateurs ont deux valeurs importantes : l'ampérage et le voltage.

Exemple : ESC 30A 3-4S

- 30A veut dire qu'il peut fournir jusqu'à 30A
- 3-4S définit le nombre de cellules de la batterie qui l'alimentera et donc son voltage (pour une batterie LiPo, on compte en général 3.7V / cellule)

Sur la fiche technique des moteurs est indiqué leur consommation maximale. Si un moteur a une consommation maximale de 25A, il faut prendre des ESC d'au moins 30A.

Applications :

- Multirotors
- Avions RC
- Quadcopter
- Hexacopter etc.

II.2.5 MPU6050

Le module MPU6050 est un microsystème électromécanique (MEMS) qui comprend un accéléromètre à 3 axes et un gyroscope à 3 axes. Cela nous permet de mesurer l'accélération, la vitesse, l'orientation, le déplacement et de nombreux autres paramètres liés au mouvement d'un système ou d'un objet.

Configuration du brochage du MPU6050 :



Figure II.7 : Le module MPU6050. [18]

Numéro de la broche	Nom de la broche	Description
1	Vcc	Fournit l'alimentation du module, peut être de +3V à +5V. Généralement, le +5V est utilisé
2	Ground	Connecté à la terre du système
3	Serial Clock (SCL)	Utilisé pour fournir une impulsion d'horloge pour la communication I2C.
4	Serial Data (SDA)	Utilisé pour le transfert de données par la communication I2C
5	Auxiliary Serial Data (XDA)	Peut être utilisé pour interfacier d'autres modules I2C avec MPU6050. Il est optionnel
6	Auxiliary Serial Clock (XCL)	Peut être utilisé pour interfacier d'autres modules I2C avec MPU6050. Il est optionnel
7	AD0	Si plus d'un MPU6050 est utilisé par un seul MCU, cette broche peut être utilisée pour faire varier l'adresse.
8	Interrupt (INT)	Broche d'interruption pour indiquer que les données sont disponibles pour la lecture par le MCU.

Tableau II.7 : Configuration du brochage du MPU6050 [18]

Caractéristiques du MPU6050[17] :

- Valeurs combinées de l'accéléromètre MEMS 3-axes et du gyroscope 3 axes
- Alimentation électrique : 3-5V
- Communication : protocole I2C
- ADC 16 bits intégré pour une grande précision
- Le DMP intégré fournit une puissance de calcul élevée.
- Peut être utilisé pour s'interfacer avec d'autres dispositifs IIC comme le magnétomètre.
- Adresse IIC configurable
- Capteur de température intégré

Accéléromètre [19]

- L'un des composants clés du MPU-6050 est un accéléromètre. Comme son nom l'indique, il s'agit d'un dispositif capable de mesurer l'accélération.
- L'accélération est décrite comme étant le taux de changement de la vitesse d'un objet. Un objet se déplaçant à une vitesse constante n'a pas d'accélération.
- Une unité courante d'accélération est la force G. Sur Terre, une force G équivaut à 9,8 mètres par seconde au carré, ce qui correspond à l'accélération de la gravité. Sur d'autres planètes, le chiffre est différent.
- Un accéléromètre doit prendre en compte l'accélération statique comme la force de gravité lorsqu'il effectue ses mesures.
- L'accéléromètre utilisé dans le MPU-6050 est un accéléromètre à trois axes, ce qui signifie qu'il détecte l'accélération sur les axes X, Y et Z.

Gyroscope [19]

- Un autre composant clé du MPU-6050 est le gyroscope.
- Un gyroscope peut mesurer le moment angulaire ou la rotation le long des axes X, Y et Z. Ces composants sont essentiels pour maintenir l'équilibre des avions et des engins spatiaux.

- Un gyroscope mécanique est constitué d'une roue ou d'un disque monté de manière à pouvoir tourner rapidement autour d'un axe libre de se déplacer dans n'importe quelle direction.
- Le type de gyroscope utilisé dans le MPU-6050 est un gyroscope "Micro Electro Mechanical System" ou MEMS. Il se compose de trois capteurs, un par axe, qui produisent une tension lorsqu'ils tournent. Cette tension est échantillonnée en interne à l'aide d'un convertisseur analogique-numérique 16 bits.

Branchement du MPU6050 avec Arduino :

Il est très facile de brancher le MPU6050 avec Arduino.

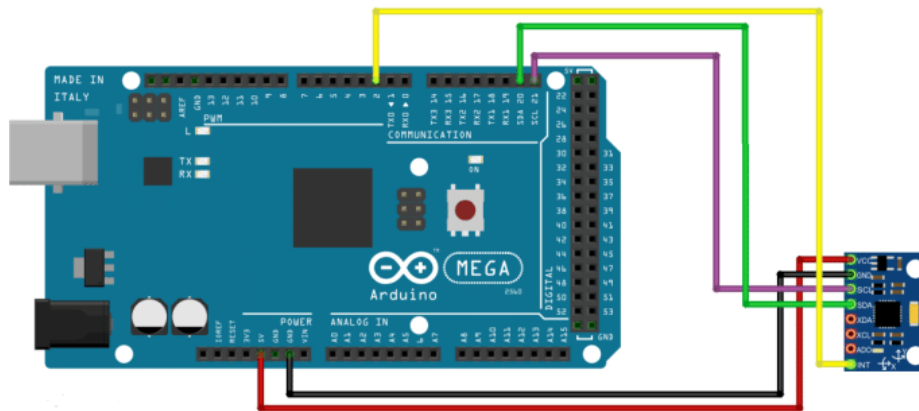


Figure II.8 : Branchement du MPU6050 avec Arduino.

Arduino	MPU6050
5v/3v	VCC
GND	GND
A5/ SCL pin	SCL
A4/SDA pin	SDA

Tableau II.8 : Branchement du MPU6050 avec Arduino.

Applications :

- Utilisé pour la mesure de l'IMU
- Drones / Quadcopters
- Robots auto-équilibrés

- Contrôle des bras robotiques
- Robots humanoïdes
- Capteur d'inclinaison
- Détecteur d'orientation / de rotation

II.2.6 Source et gestion d'énergie Batterie LiPo

La plupart des multicoptères électriques sont équipés de batteries de type lithium-ion-polymer ou LiPo. Ces modèles de batteries permettent d'emmagasiner suffisamment d'énergie et leur poids est moindre que les batteries lithium-ion, à capacité égale. En d'autres termes, elles possèdent le meilleur rapport puissance/masse. De plus, leur capacité à offrir une décharge rapide, grâce à leur faible résistance interne, se révèle très utile. Ce type de batterie est toutefois hautement inflammable, et il est donc essentiel de prendre ce risque en compte lors de leur manipulation, leur chargement et leur stockage [1].

Le tableau (II.2) montre les paramètres de l'accumulateur Batterie LiPo 3S 4200mAh choisi dans notre réalisation.

Batterie	Li Po HJPOWER
Le voltage(v)	3×3.7=11.1
La capacité(mAh)	4000
Le taux de décharge	25C
Configuration (nombre de cellules)	3s
Poids(g)	275
Type de connecteur	XT60
La taille(cm)	4×2×13

Tableau II.9 : La fiche technique de la batterie Li Po 3S 4200 mAh 25C.



Figure II.9 : La batterie Li Po 3S 4200 mAh 25C.

Applications :

- Drones
- QUADCOPTERS
- Robotique

II.2.7 La carte Arduino Mega 2560

L'Arduino Mega 2560 est une carte microcontrôleur basée sur l'ATmega2560 (datasheet). Elle possède 54 broches d'entrée/sortie numériques (dont 14 peuvent être utilisées comme sorties PWM), 16 entrées analogiques, 4 UART (ports série matériels), un oscillateur à quartz de 16 MHz, une connexion USB, une prise d'alimentation, un connecteur ICSP et un bouton de réinitialisation. Il contient tout ce qui est nécessaire pour prendre en charge le microcontrôleur ; il suffit de le connecter à un ordinateur avec un câble USB ou de l'alimenter avec un adaptateur CA/CC ou une batterie pour commencer. Le Mega est compatible avec la plupart des shields conçus pour l'Arduino Duemilanove ou Diecimila [21].



Figure II.10 : l'Arduino Mega 2560 [21].

Les caractéristiques d'Arduino Mega 2560 :

Microcontrôleur	ATmega2560
Tension de fonctionnement	5V
Tension d'entrée (recommandée)	7-12V
Tension d'entrée (limites)	6-20V
Broches d'E/S numériques	54 (dont 14 fournissent une sortie PWM)
Broches d'entrée analogique	16
Courant continu par broche E/S	40mA
Courant continu pour la broche 3.3V	50mA
Mémoire flash	256 KB dont 8 KB utilisés par le bootloader
SRAM	8KB
EEPROM	4 KB
Vitesse d'horloge	16 MHz

Tableau II.10 : Les caractéristiques d'Arduino Mega 2560 [21].

II.2.8 Système de Communication (Le transmetteur FS-i6X et son récepteur FS-iA10B)

Le système de communication radio est l'équipement qui assure l'interface entre l'humain et la machine. Le télépilote actionne les différents leviers et boutons sur la télécommande et les ordres correspondants sont communiqués par ondes radio via un émetteur vers un récepteur installé sur le drone, on utilise principalement les bandes de 2,4 GHz et 5,8 GHz pour la communication avec les drone [1].

Pour notre part, on 'a investi dans la FlySky FS-i6 et son récepteur, livrée avec son récepteur. Cette radiocommande dispose de 10 canaux, de 4 switches, d'un écran LCD et de boutons permettant de naviguer dans les menus de configuration.



Figure II.11 : FlySky FS-i6 et son récepteur FS-iA10B.

Spécification [21] :

- FlySky FS-i6 :

Canaux	10
Type de protection	CONTRÔLEUR RADIO
Type de modèle	Quadcopter
Gamme RF	2.408 - 2.475GHz
Bande passante	500 KHz
Alarme de basse tension	Oui (inférieur à 4,2V)
PS2/USB Port	Oui (Micro-USB)
Alimentation électrique	6V DC 1.4AA*4
Poids	392g
Taille	174*89*190 mm

Tableau II.11 : Spécification de FlySky FS-i6.

• FS-iA10B :

Canaux PWM	10
Fréquence sans fil	2.4GHz
Protocole sans fil	AFHDS 2A
Gamme	500 ~ 1500m (en l'air)
Type d'antenne	Antenne double tube en cuivre (150mm * 2)
Puissance	4.0-8.4V
RSSI	Supporté
Port de données	PWM / PPM / i.bus / s.bus
Poids	19.3 g
Taille	47 * 33.1 * 14.7 mm

Tableau II.12 : Spécification de FS-iA10B.

II.3 Les mouvements du quadrirotor

Pour réaliser les six mouvements du quadrirotor nous n'avons à notre disposition que quatre actionneurs, par conséquent les mouvements de l'appareil sont couplés ce qui signifie que nous ne pouvant pas effectuer un des mouvements sans en impliquer d'autres. Par exemple en réduisant la vitesse du rotor droit, le quadrirotor s'incline de ce côté et l'équilibre entre les rotors qui tourne dans le sens des aiguilles d'une montre et les rotors qui tourne dans le sens inverse est perturbé, ce qui implique deux mouvements de rotation appelés roulis et lacet. Le mouvement de rotation de roulis implique un mouvement de translation sur l'axe Y. C'est cette particularité qui permet de commander le quadrirotor sur six degrés de libertés avec seulement quatre actionneurs. [22]

Ainsi le quadrirotor a les cinq mouvements suivants:

• Les mouvements de rotation

Rotation de roulis

Rotation de tangage

Rotation de lacet

- Les mouvements de translations

Translation verticale

Translation horizontale

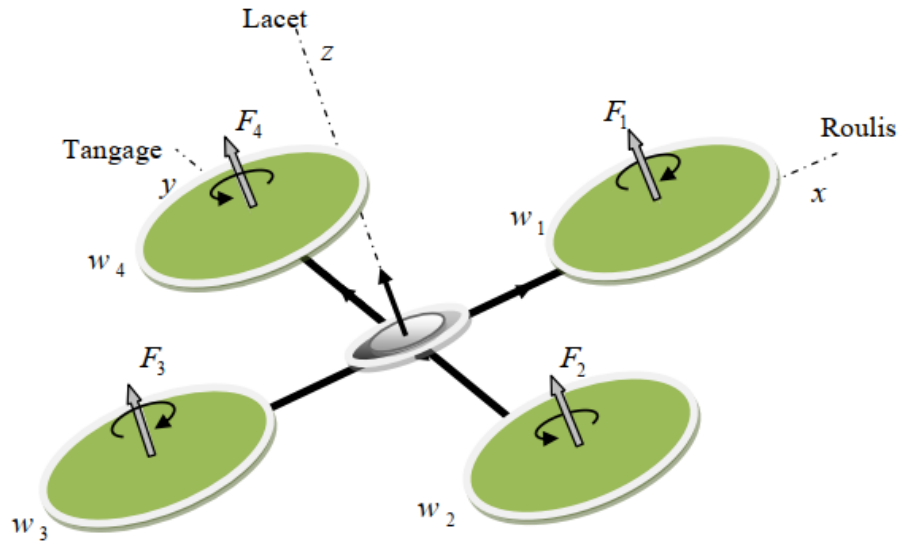


Figure II.12: Structure générale d'un quadrirotor

II.3.1 Les mouvements de rotation du quadrirotor :

La combinaison des forces $\vec{F}_i$ et des couples résistants C_i produit les mouvements angulaires autour des axes du quadrirotor. Ces axes représentent les axes d'attitude du quadrirotor. Les mouvements angulaires sont définis selon l'axe. On a alors **le mouvement de lacet, le mouvement de roulis et le mouvement de tangage**.

II.3.1.1 Mouvement de lacet

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

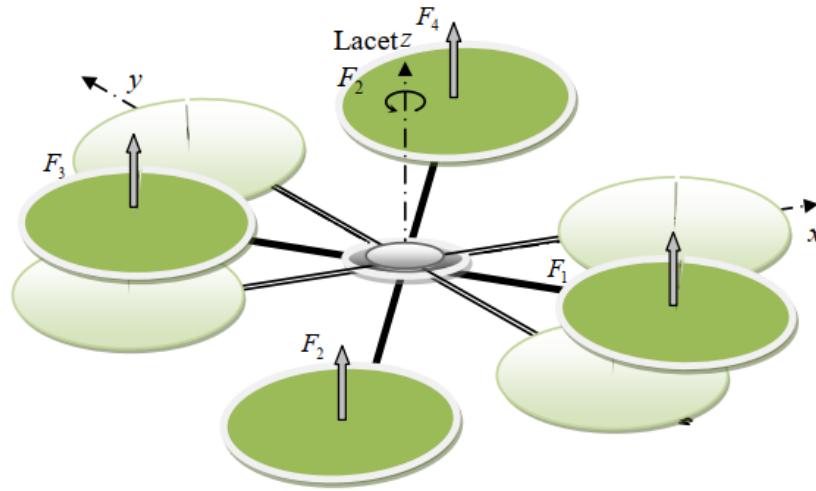


Figure II.13 : Illustration du mouvement de lacet

II.3.1.2 Mouvement de roulis :

Ce mouvement est assuré par la différence des forces (F_1 , F_3) produites par les moteurs avant et arrière. C'est le même principe que pour le mouvement de tangage mais le couple créé par la différence des forces sera autour de l'axe de roulis. La figure (IV.3) présente une illustration d'un mouvement de roulis pour un quadrirotor.

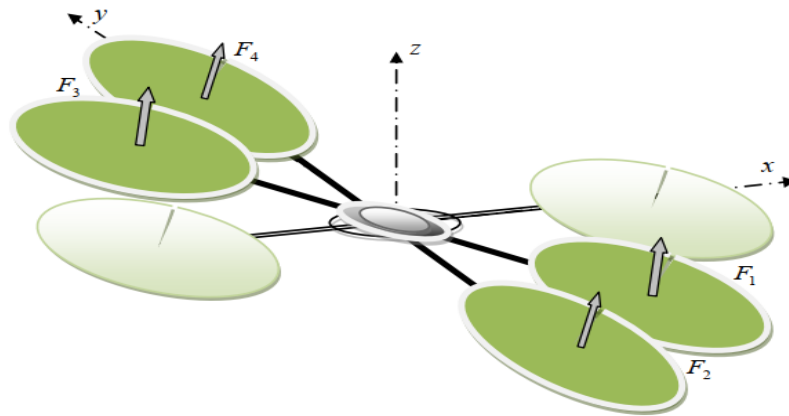


Figure II.14 : Mouvement de roulis de quadrirotor

II.3.1.3 Mouvement de tangage :

Ce mouvement est assuré par la différence des forces (F_2 , F_4) produites par les moteurs droite et gauche c'est à dire que les moteurs M_1 et M_3 vont tourner avec la même vitesse de rotation tandis que M_2 et M_4 vont tourner avec deux vitesses différentes. Par conséquent cette différence de forces produit un couple autour de l'axe y . La figure 2.5 présente une illustration d'un mouvement de tangage pour un quadrirotor.

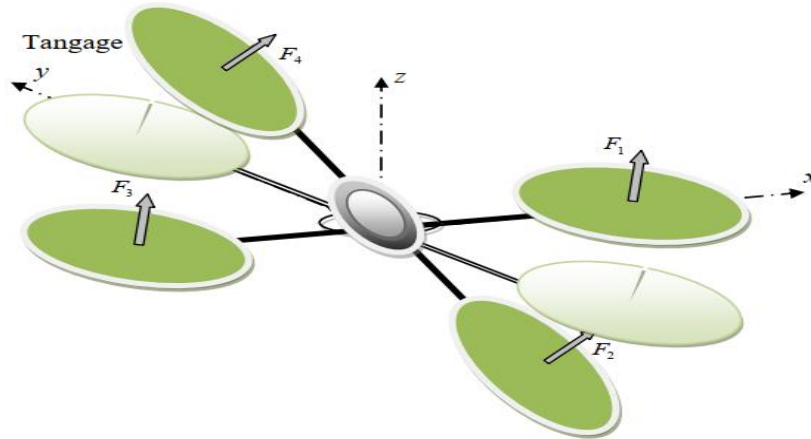


Figure II.15 : Mouvement de tangage de quadrirotor

Les commandes de mouvements angulaires sont envoyées au quadrirotor afin de permettre des déplacements dans les différentes directions : gauche/droite, avant/arrière, haut/bas.

II.3.2 Les mouvements de translations du quadrirotor

Les mouvements de translations sont classés en deux : les translations verticales et les translations horizontales.

II.3.2.1 Translation verticale

Ce mouvement est un déplacement suivant l'axe z . Il peut être effectué en faisant augmenter la vitesse de rotation des 4 moteurs de façon à avoir une force de poussée $F_T > mg$ tandis que si $F_T < mg$ donc nous allons avoir un mouvement vertical vers le bas suivant l'axe z . La figure (IV.5) présente une illustration de translation verticale pour un quadrirotor

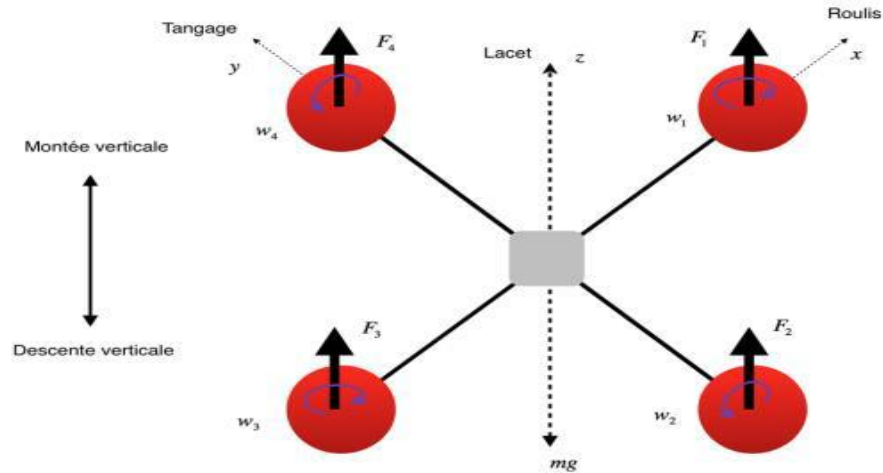


Figure II.16 : Translation verticale de quadrirotor

II.3.2.2 Translation horizontale

Il a le même principe que pour la translation verticale mais cette fois-ci dans le plan $x y$. Il faut augmenter les vitesses de rotation de façon à ce que la force de poussée maintient le quadrirotor à une hauteur constante par rapport au sol et en appliquant une force le long de x ou y , le système effectue un vol connu dans la littérature sous le nom de « vol en palier ».

La figure (IV.6) présente une illustration de translation horizontale pour un quadrirotor.

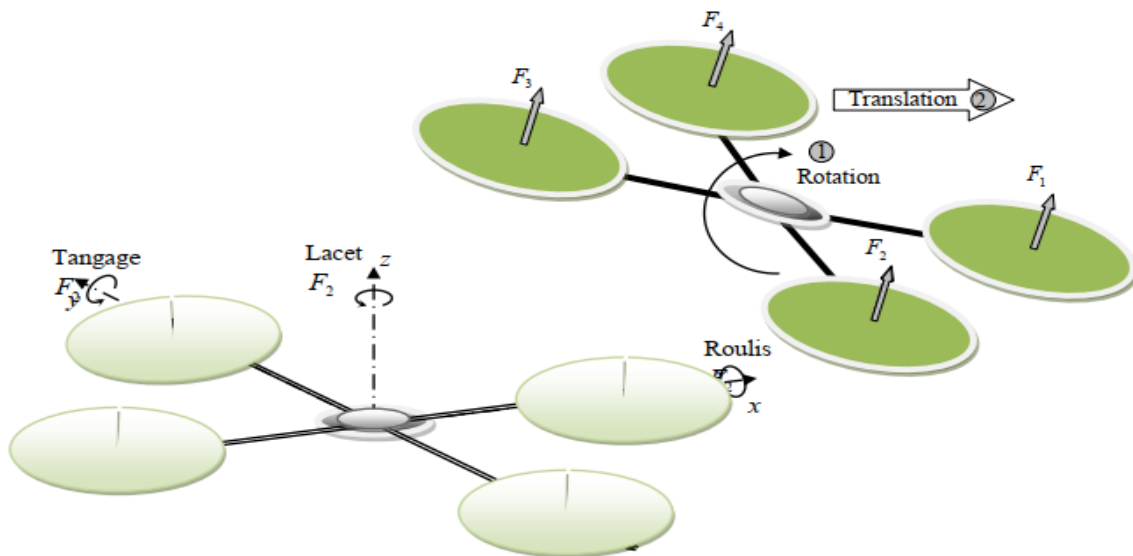


Figure II.17: Illustration du mouvement de translation

II.4. Conclusion

Dans ce chapitre, une description détaillée des composants utilisés dans le montage de notre drone quadrirotor et ses mouvements est présentée, également les logiciels utilisés pour réaliser notre travail.

Dans le chapitre suivant, nous présenterons les opérations d'assemblage des composants et réalisation de quadrirotor. Ensuite, nous réaliserons des tests sur le calibrage des capteurs, les contrôleurs de vitesse électronique ESC, les moteurs et la radiocommande FLYSKY.

Chapitre III :
Tests réels & discussions de résultats

Chapitre III :

Tests réels & discussions de résultats

III.1 Introduction

Le but de ce chapitre est de donner des expériences complètes sur le fonctionnement des différentes parties de notre système. Nous expliquerons le branchement de matériel que nous avons utilisé pour ces expériences et comment assembler les outils et gadgets électroniques de base pour construire un drone. Nous donnerons également une explication détaillée du logiciel utilisé pour ces expériences.

III.2 Montage et réalisation du quadrirotor

Dans les parties suivantes nous allons faire plusieurs montages et différents tests nécessaires nous irons pas à pas pour arriver à la réalisation finale de notre quadrirotor, dans chaque montage nous allons apprendre de nouvelles choses sur l'électronique, la programmation, les effets aéronautiques ...etc., et nous connaissons les détails de chaque composant : moteurs, ECS, Flysky FS-I6X, MPU-6050.

III.2.1 Expériences sur MPU-6050

Après avoir identifié le MPU-6050 dans le chapitre précédent, nous allons maintenant nous intéresser au module MPU-6050 pour voir comment l'utiliser avec un Arduino.

III.2.1.1 Première expérience

En fait, dans cet exemple, nous allons voir si notre capteur fonctionne et nous allons afficher les mesures du capteur sur MATLAB.

Dans cette expérience, nous allons tester le gyroscope et l'accélération. Nous utiliserons :

- **Matériel :**

- Carte Arduino Mega 2560.
- MPU 6050
- Fil de liaison
- **Logiciel :**
 - MATLAB

Pour Branchement du MPU6050 avec Arduino dans notre expérience voire le tableau et la Figure ci-dessous.

Arduino Mega 2560	MPU6050
5v/3v	VCC
GND	GND
SCL pin	SCL
SDA pin	SDA

Tableau III.1 : Branchement du MPU6050 avec Arduino Mega 2560.

Figure III.1 : Branchement du MPU6050 avec Arduino Mega 2560.

- **Testez l'accélération :**

Nous avons extrait les mesures d'accélération des registres 59 à 64, tout cela par programmation sur MATLAB.

La figure suivante donne le programme sur MATLAB qui nous permet d'obtenir des mesures d'accélération (x, y, z).

```

1 - hold all
2 - a = arduino('COM3','Mega2560');
3 - mpu = i2cdev(a,'0x68'); %L'adresse du mpu est normalement 0x68
4 - writeRegister(mpu, hex2dec('B6'), hex2dec('00'), 'int16'); %réinitialiser
5 - data = zeros(10000,14,'int8'); %la prélocalisation pour la vitesse
6 - j = 1;
7 - a1 = animatedline('Color',[1 0 0]);
8 - a2 = animatedline('Color',[0 1 0]);
9 - a3 = animatedline('Color',[0 0 1]);
10 - legend('Accel_x','Accel_y','Accel_z')
11 - grid on
12
13 %% loop
14 while(true)
15     x=1;
16     for i=59:64
17         data(j,x) = readRegister(mpu, i, 'int8');
18         x = x + 1;
19     end
20     y = swapbytes(typecast(data(j,:), 'int16'))
21     addpoints(a1,j,double(y(1)));
22     addpoints(a2,j,double(y(2)));
23     addpoints(a3,j,double(y(3)));
24     j = j+1;
25     drawnow limitrate
26 end

```

Figure IV.2 : Programme pour obtenir des mesures d'accélération (x, y, z) sur MATLAB.

La figure III.3 montre l'accélération des registres 59 à 64 selon les axes x, y, z en fonction de temps. L'accélération selon l'axe x représente la courbe rouge, l'accélération selon l'axe y représente la courbe vert et L'accélération selon l'axe z est blue,

Les variations brusques des courbes dans l'intervalle de 175s à 1300s sont dû au déplacement de MPU6050 dans différentes vitesses sur les axes x, y, z.

Les mesures de valeurs d'accélération de MPU6050 en fonction de temps stabilisants si on pause les déplacements, et les courbes restent stable aux dernières valeurs d'accélération mesuré.

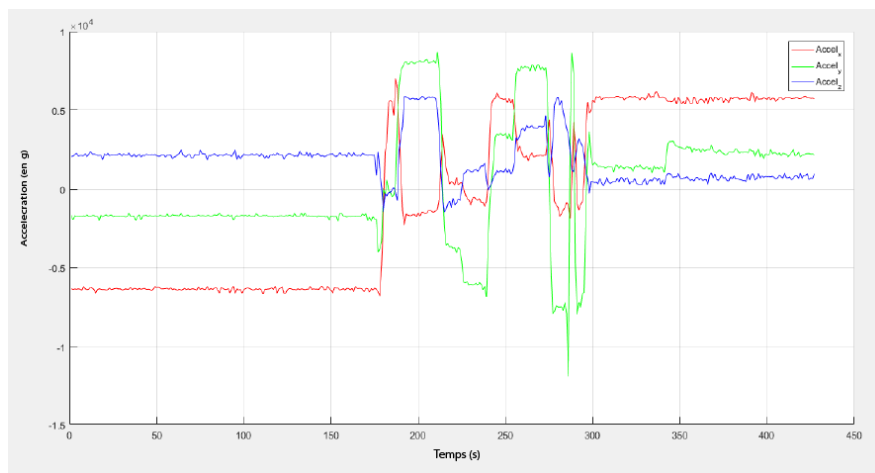


Figure III.3 : Mesures d'accélération en fonction de temps.

• Testez le gyroscope

Nous avons extrait les mesures de gyroscope des registres 67 à 72, tout cela par programmation sur MATLAB.

La figure suivante donne le programme sur MATLAB qui nous permet d'obtenir des mesures de gyroscope (x, y, z).

```

1 - hold all
2 - a = arduino('COM3','Mega2560');
3 - mpu = i2cdev(a,'0x68'); %L'adresse du mpu est normalement 0x68
4 - writeRegister(mpu, hex2dec('B6'), hex2dec('00'),'int16'); %réinitialiser
5 - data = zeros(10000,14,'int8'); %la prélocalisation pour la vitesse
6 - j = 1;S
7 - gyro1 = animatedline('Color',[1 0 0]);
8 - gyro2 = animatedline('Color',[0 1 0]);
9 - gyro3 = animatedline('Color',[0 0 1]);
10 - legend('gyro_x','gyro_y','gyro_z')
11 - grid on
12
13 %% loop
14 - while(true)
15 -     x=1;
16 -     for i=67:72
17 -         data(j,x) = readRegister(mpu, i, 'int8');
18 -         x = x + 1;
19 -     end
20 -     gyro = swapbytes(typecast(data(j,:), 'int16'))
21 -     addpoints(gyro1,j,double(gyro(1)));
22 -     addpoints(gyro2,j,double(gyro(2)));
23 -     addpoints(gyro3,j,double(gyro(3)));
24 -     j = j+1;
25 -     drawnow limitrate
26 - end

```

Figure III.4 : Programme pour obtenir des mesures de gyroscope (x, y, z) sur MATLAB.

La figure III.5 montre la variation de vitesse angulaire selon les axes x, y, z en fonction de temps. Gyro selon l'axe x représente la courbe rouge, gyro selon l'axe y représente la courbe vert et gyro selon l'axe z est blue.

D'après la figure ci-dessous, Les changements de courbes présentées les mesures de vitesse angulaire dans l'intervalle de 150s à 175s à cause du déplacement de gyroscope dans des différents vitesses et angles.

Les mesures de vitesse angulaires de gyroscope en fonction de temps stabilisent à 0 si on pause les changements d'angles et de vitesse, alors les allures des mesures reviennent à l'état initiale.

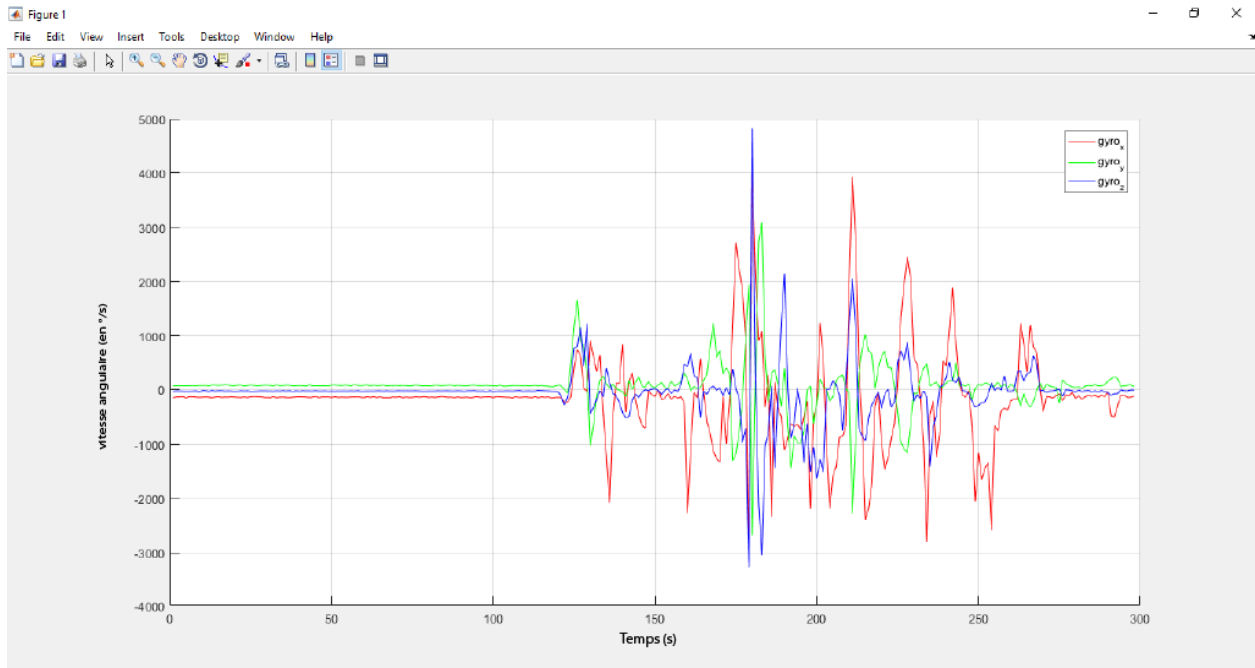


Figure III.5 : La variation de vitesse angulaire de gyroscope en fonction de temps.

III.2.1.2 Deuxième expérience

Dans cette expérience, nous allons obtenir les angles de tangage, de roulis et de lacet à partir des données du gyroscope.

L'Arduino Mega 2560 du contrôleur de vol lit les données des angles parcouru par le Gyro à partir des registres de sortie d'adresses 0x69 et 0xA8 qui sont à 16 bits en utilisant le protocole de communication I²C.

Par exemple pour lire l'axe « Yaw » on lit les 8 bits du poids fort du registre 16 bit à part, puis les 8 bits du poids faible et on fait un décalage pour avoir les 16 bits puis on affecte le résultat au variable « gyro_yaw ».

Toutes les autres parties intéressantes pour la communication I²C avec le Gyro sont gérées par la librairie « wire » qui nous facilitent la tâche.

La figure ci-dessous montre l'organigramme qui explique le processus :

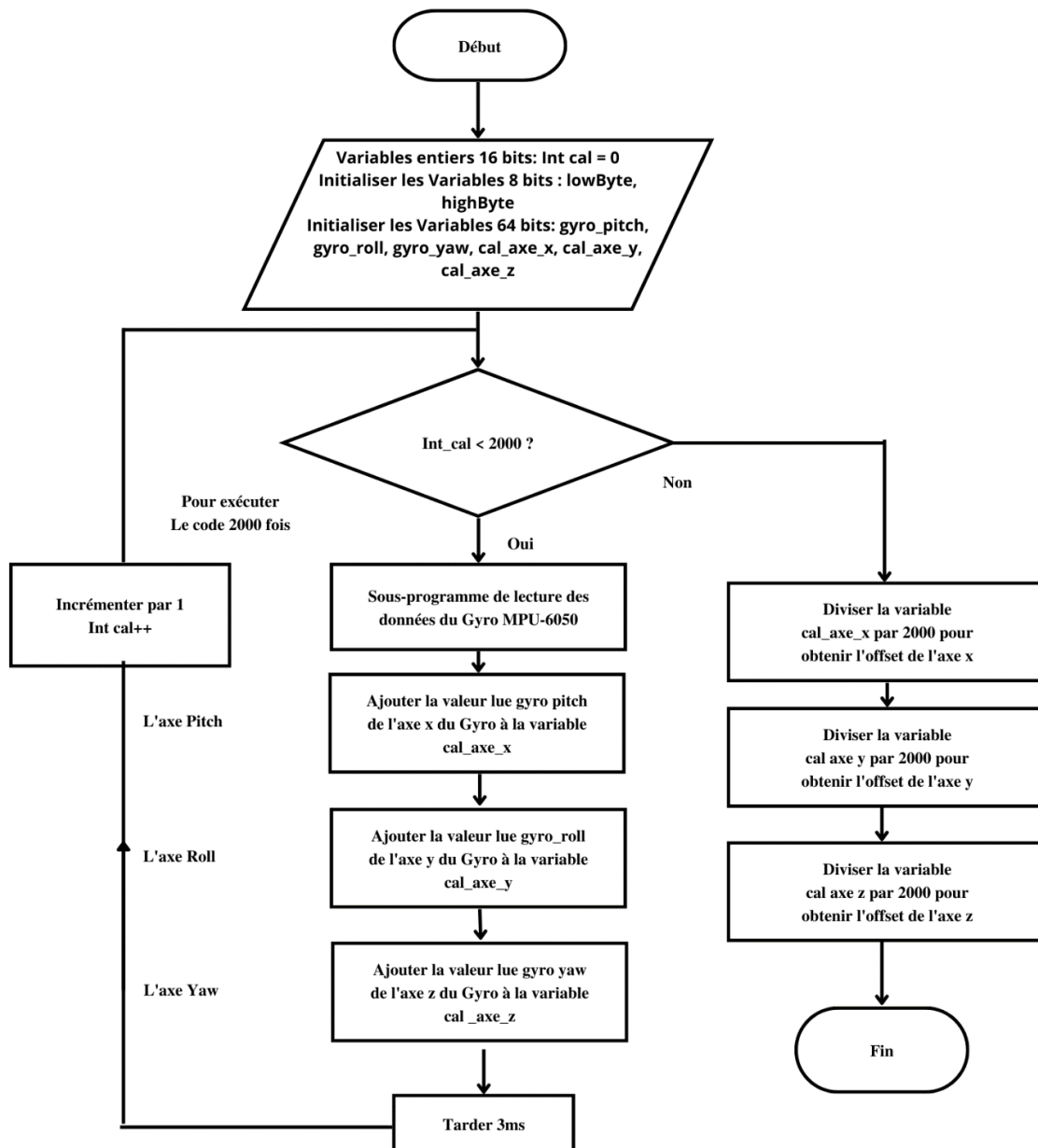


Figure III.6 : l'organigramme qui explique le processus.

Nous utiliserons :

- **Matériel :**
 - Carte Arduino Mega 2560.
 - MPU 6050
 - Fil de liaison

- **Logiciel :**
 - MATLAB
 - IDE Arduino

Pour Le branchement voir la **Figure V.1**.

Nous avons programmé ce code dans logiciel IDE Arduino pour obtenir des données et nous avons utilisé MATLAB pour dessiner les courbes.

```

MPU6050_gyro_pitch_roll_yaw $
1
2
3 #include <Wire.h>
4 #include <MPU6050.h>
5
6 MPU6050 mpu;
7
8 // Timers
9 unsigned long timer = 0;
10 float timeStep = 0.01;
11
12 // Pitch, Roll and Yaw values
13 float pitch = 0;
14 float roll = 0;
15 float yaw = 0;
16
17 void setup()
18 {
19   Serial.begin(115200);
20
21   // Initialize MPU6050
22   while(!mpu.begin(MPU6050_SCALE_2000DPS, MPU6050_RANGE_2G))
23   {
24     Serial.println("Could not find a valid MPU6050 sensor, check wiring!");
25     delay(500);
26   }
27
28   // Calibrate gyroscope. The calibration must be at rest.
29   // If you don't want calibrate, comment this line.

```

Impression terminée.

Figure III.7 : Programme pour tracer les courbes pitch, roll, yaw.

La figure III.7 montre les mesures d'angle selon mouvement de gyroscope en fonction de temps. Pitch représente la courbe blue, roll selon y représente la courbe rouge et yaw selon z est orange.

Entre l'intervalle de 0 à 700ms les courbes sont stable, au moment de 700ms on a déplacé gyroscope dans des différents angles on remarque qu'entre 700ms et 2800ms des changements des courbes selon x, y, z en fonction de temps, après ces courbes stabilisants le temps on a fixé l'angle du gyroscope.

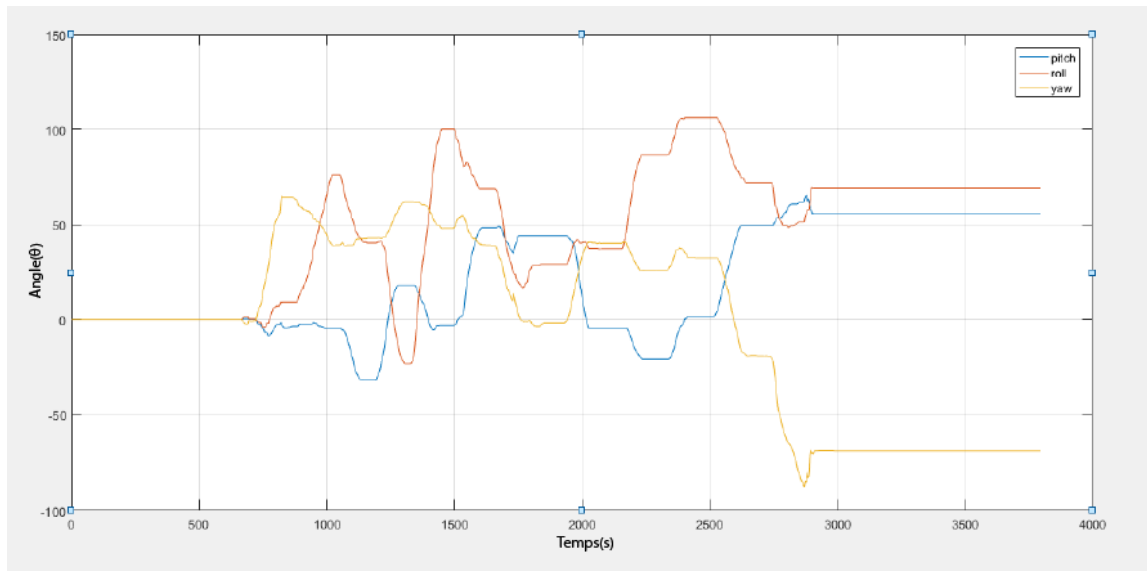


Figure III.8 : Les mesures d'angle selon mouvement de gyroscope en fonction de temps.

III.2.2 Expériences sur Le transmetteur FS-i6X et son récepteur FS-iA10B

III.2.2.1 Protocoles TX

Nous verrons ce terme utilisé pour décrire un émetteur RC. Le terme "mode" s'applique généralement à l'utilisation de l'émetteur avec un avion et fait référence à la façon dont l'émetteur est configuré pour contrôler l'avion, c'est-à-dire quels manches actionnent quelles commandes sur l'avion.

Un émetteur standard possède deux manches, chacun pouvant effectuer des mouvements horizontaux et verticaux. Chaque manche possède donc deux canaux, un pour les mouvements horizontaux et un autre pour les mouvements verticaux.

Il existe quatre modes TX au total :

- Mode 1 - Le manche gauche actionne la gouverne de profondeur et la gouverne de direction, le manche droit actionne les gaz et les ailerons.
- Mode 2 - Le manche gauche actionne les gaz et la direction, le manche droit actionne la profondeur et les ailerons.
- Mode 3 - Le manche gauche actionne la profondeur et les ailerons, le manche droit actionne les gaz et le gouvernail.

- Mode 4 - Le manche gauche actionne les gaz et les ailerons, le manche droit actionne la profondeur et le gouvernail.

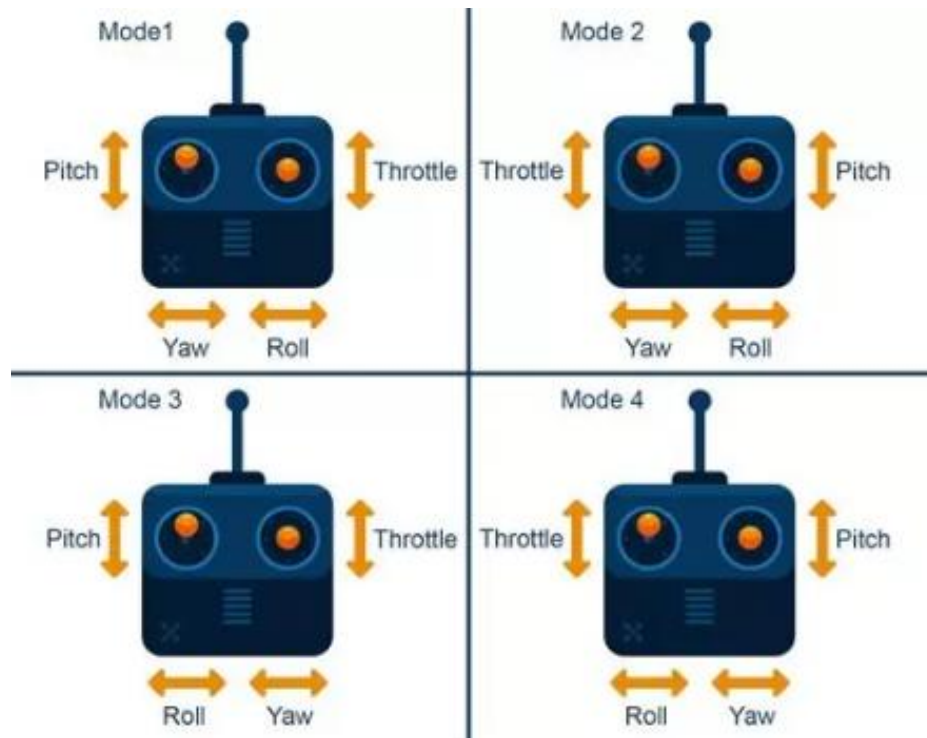


Figure III.9 : Les différents modes.

III.2.2.2 Protocoles RX

Une fois de plus, nous avons un terme qui peut être utilisé pour deux aspects différents du système de radiocommande. Alors qu'un "protocole" peut se référer à la méthode utilisée pour échanger des données entre l'émetteur et le récepteur, dans ce contexte, il se réfère aux ports sur le récepteur.

Le protocole le plus communément utilisé du côté du récepteur est le PWM (Pulse Width Modulation), et il est utilisé presque universellement pour contrôler les servomoteurs et les ESC.

Nous montrerons deux manières d'utiliser le récepteur avec un Arduino Mega 2560 :

- Utilisation des sorties PWM comme entrées de l'Arduino Mega 2560.
- Utilisation de la sortie iBus du Flysky.

Dans les deux expériences, nous utiliserons :

• **Matériel :**

- Carte Arduino Mega 2560.
- Transmetteur FS-i6X et le récepteur FS-iA10B
- Fil de liaison

• **Logiciel :**

- IDE Arduino

III.2.2.3 Première expérience

Une façon de travailler avec le récepteur Flysky est de lire chaque canal individuel de sortie du connecteur "servo".

Par défaut, les sorties de l'émetteur sont réglées comme suit

- Canal 1 - Manche droit, gauche/droite
- Canal 2 - Stick droit, haut/bas
- Canal 3 - Stick gauche, haut/bas
- Canal 4 - Stick gauche, gauche/droite
- Canal 5 - Contrôle VRA
- Canal 6 - Contrôle VRB

Ainsi, en connectant des dispositifs à ces sorties de canal du récepteur, vous pouvez les contrôler avec leurs fonctions associées.

Notre contrôleur émet un signal PWM, qui par sa propre nature communique en manipulant la largeur d'une impulsion. Nous utilisons Arduino Mega 2560, pour mesurer la largeur de l'impulsion et interpréter ces données.

Le branchement est assez simple, les six sorties du récepteur Flysky sont connectées aux six entrées de l'Arduino Mega 2560.

Nous connectons également la sortie 5 volts de l'Arduino Mega 2560 au VCC du récepteur pour l'alimenter. Et, bien sûr, nous connectons également une masse.

Pour interfacer le récepteur FS-iA10B avec l'Arduino Mega 2560 dans notre expérience, voir le tableau et la figure ci-dessous.

Arduino Mega 2560	FS-iA10B
GND	GND
5V	VCC
Pin numérique 3 (PWM)	C1
Pin numérique 4 (PWM)	C2
Pin numérique 5 (PWM)	C3
Pin numérique 6 (PWM)	C4
Pin numérique 7 (PWM)	C5
Pin numérique 8 (PWM)	C6

Tableau III.2 : branchement du FS-iA10B avec Arduino Mega 2560.

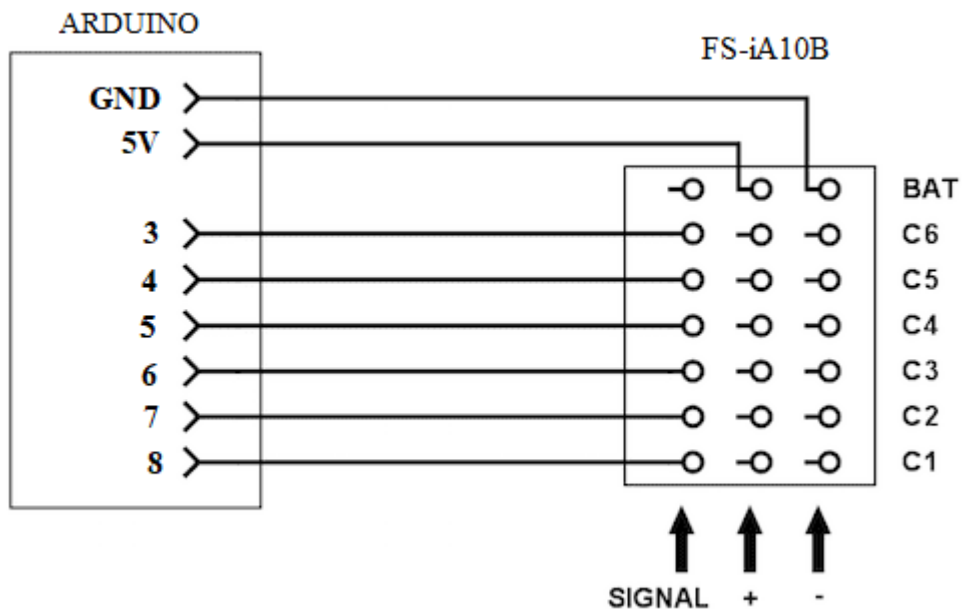


Figure III.10: branchement du FS-iA10B avec Arduino Mega 2560.

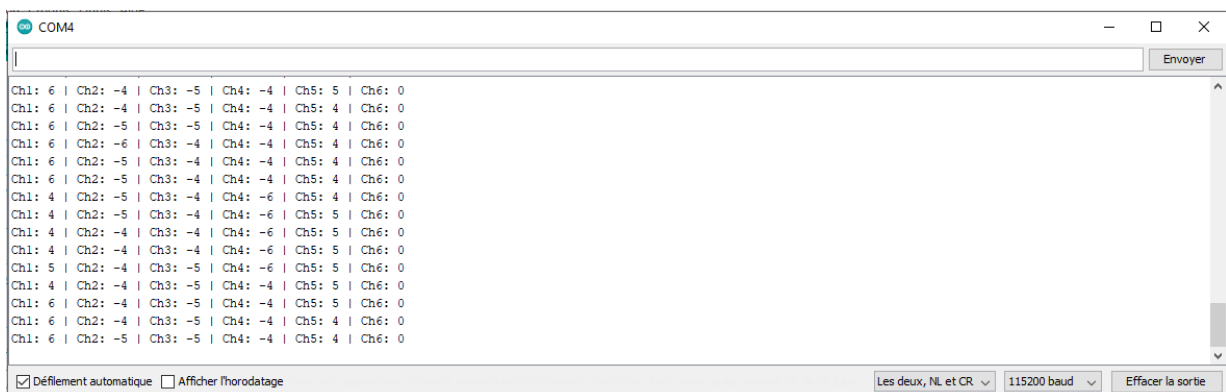
La figure ci-dessous montre le programme de communication avec l'Arduino et les canaux de notre transmetteur pour obtenir les données.

```
FLYSKY$
1 |
2 |
3 // Define Input Connections
4 #define CH1 3
5 #define CH2 4
6 #define CH3 5
7 #define CH4 6
8 #define CH5 7
9 #define CH6 8
10
11 // Integers to represent values from sticks and pots
12 int ch1Value;
13 int ch2Value;
14 int ch3Value;
15 int ch4Value;
16 int ch5Value;
17
18 // Boolean to represent switch value
19 bool ch6Value;
20
21 // Read the number of a specified channel and convert to the range provided.
22 // If the channel is off, return the default value
23 int readChannel(int channelInput, int minLimit, int maxLimit, int defaultValue){
24     int ch = pulseIn(channelInput, HIGH, 30000);
25     if (ch < 100) return defaultValue;
26     return map(ch, 1000, 2000, minLimit, maxLimit);
27 }
28
29 // Read the switch channel and return a boolean value
```

Figure III.11: Programme de communication avec l'Arduino et les canaux de transmetteur.

On remarque qu'il y a des perturbations dans des données des canaux, voire La figure ci-dessous

Dans la figure III.11 On trouve que les valeurs des chaînes ne sont pas proches 0, alors nous avons besoin de faire un étalonnage.



```
COM4
Ch1: 6 | Ch2: -4 | Ch3: -5 | Ch4: -4 | Ch5: 5 | Ch6: 0
Ch1: 6 | Ch2: -4 | Ch3: -5 | Ch4: -4 | Ch5: 4 | Ch6: 0
Ch1: 6 | Ch2: -5 | Ch3: -5 | Ch4: -4 | Ch5: 4 | Ch6: 0
Ch1: 6 | Ch2: -6 | Ch3: -4 | Ch4: -4 | Ch5: 4 | Ch6: 0
Ch1: 6 | Ch2: -5 | Ch3: -4 | Ch4: -4 | Ch5: 4 | Ch6: 0
Ch1: 6 | Ch2: -5 | Ch3: -4 | Ch4: -4 | Ch5: 4 | Ch6: 0
Ch1: 4 | Ch2: -5 | Ch3: -4 | Ch4: -6 | Ch5: 4 | Ch6: 0
Ch1: 4 | Ch2: -5 | Ch3: -4 | Ch4: -6 | Ch5: 5 | Ch6: 0
Ch1: 4 | Ch2: -4 | Ch3: -4 | Ch4: -6 | Ch5: 5 | Ch6: 0
Ch1: 4 | Ch2: -4 | Ch3: -4 | Ch4: -6 | Ch5: 5 | Ch6: 0
Ch1: 5 | Ch2: -4 | Ch3: -5 | Ch4: -6 | Ch5: 5 | Ch6: 0
Ch1: 4 | Ch2: -4 | Ch3: -5 | Ch4: -4 | Ch5: 5 | Ch6: 0
Ch1: 6 | Ch2: -4 | Ch3: -5 | Ch4: -4 | Ch5: 5 | Ch6: 0
Ch1: 6 | Ch2: -4 | Ch3: -5 | Ch4: -4 | Ch5: 4 | Ch6: 0
Ch1: 6 | Ch2: -5 | Ch3: -5 | Ch4: -4 | Ch5: 4 | Ch6: 0
```

Figure III.12 : Les valeurs des chaînes avant l'étalonnage.

• Etalonnage de FS-i6X :

Nous utilisons FS-i6X lui-même pour résoudre le problème de perturbation, voire La figure ci-dessous.

Donc, d'après l'étalonnage les valeurs des chaînes devient proche de 0.

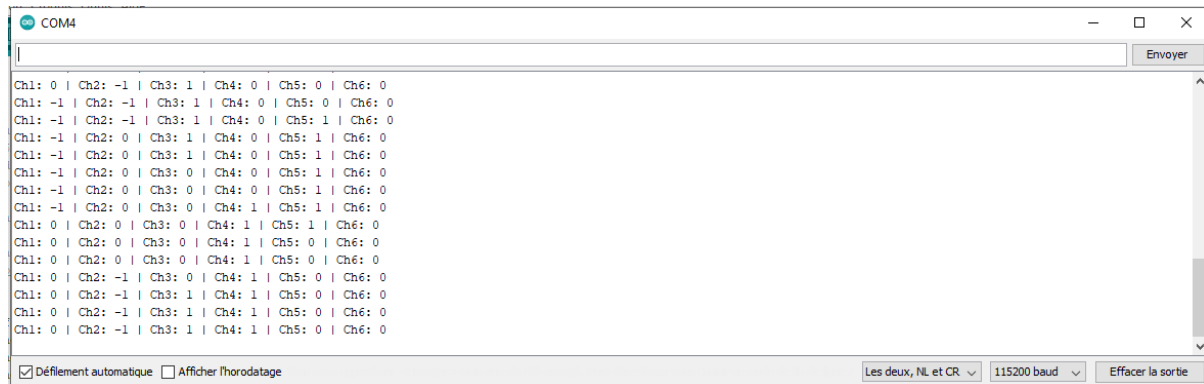


Figure III.13 : Les valeurs des chaînes après l'étalonnage.

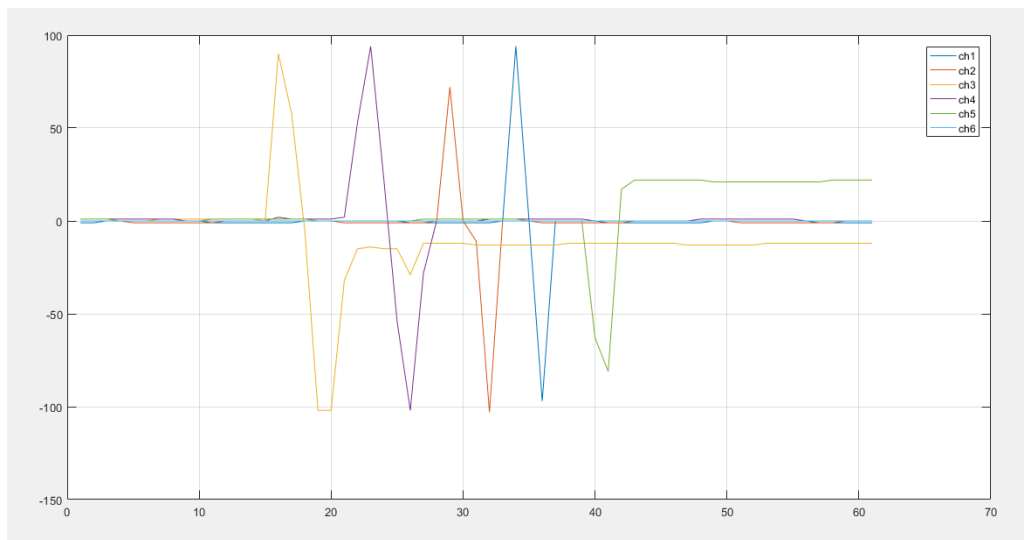


Figure III.14 : La variation des valeurs des chaînes selon les mouvements des sticks.

La figure III.13 montre la variation des valeurs des chaînes selon les mouvements des sticks en fonction de temps(s).

Donc, d'après les résultats de cet essai, nous constatons que les courbes des chaînes présentent des petits perturbations le temps on a déplacé les sticks de commande entre 15 et 42ms.

III.2.2.4 Deuxième expérience

Une autre façon d'obtenir des données du récepteur Flysky est de passer par la connexion série iBus. Cela présente quelques avantages, notamment celui de ne nécessiter qu'un seul fil (plus l'alimentation et la masse).

Nous allons avoir besoin d'un deuxième UART, ou entrée série, sur notre microcontrôleur. Nous puissions émuler un port série avec SoftwareSerial, dans ce cas, nous utilisons une bibliothèque qui nécessite un UART physique.

Pour cette raison, nous passons à un Arduino Mega 2560, il a plusieurs UARTs matériels.

Pour le branchement du FS-iA10B avec Arduino Mega 2560 dans cette expérience voir le tableau et la Figure ci-dessous.

Arduino Mega 2560	FS-iA10B (IBUS SERVO)
GND	GND
5V	VCC
19 (RX1)	DATA

Tableau III.3 : Branchement du FS-iA10B (IBUS SERVO) avec Arduino Mega 2560.

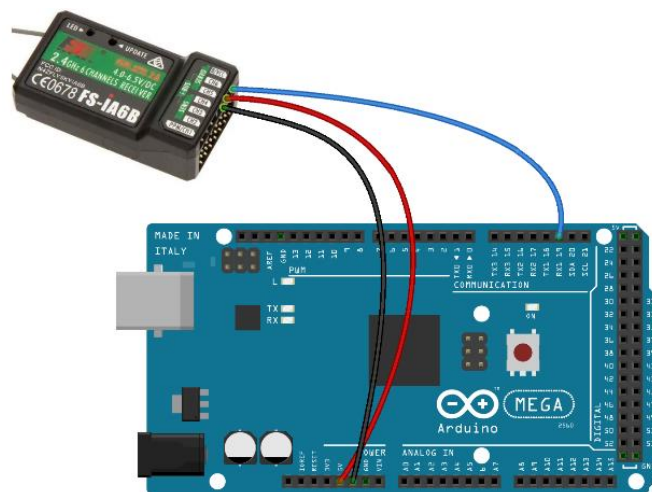


Figure III.15 : Branchement du FS-iA10B (IBUS SERVO) avec Arduino Mega 2560.

La figure ci-dessous montre le programme de communication avec l'Arduino et les canaux de notre transmetteur pour obtenir les données.

```

flysky2
1 #include <IBusBM.h>
2 // Create iBus Object
3 IBusBM ibus;
4
5 // Read the number of a given channel and convert to the range provided.
6 // If the channel is off, return the default value
7 int readChannel(byte channelInput, int minLimit, int maxLimit, int defaultValue) {
8     uint16_t ch = ibus.readChannel(channelInput);
9     if (ch < 100) return defaultValue;
10    return map(ch, 1000, 2000, minLimit, maxLimit);
11 }
12
13 // Read the channel and return a boolean value
14 bool readSwitch(byte channelInput, bool defaultValue) {
15     int intDefaultValue = (defaultValue) ? 100 : 0;
16     int ch = readChannel(channelInput, 0, 100, intDefaultValue);
17     return (ch > 50);
18 }
19
20 void setup() {
21     // Start serial monitor
22     Serial.begin(115200);
23
24     // Attach iBus object to serial port
25     ibus.begin(Serial1);
26 }
27
28 void loop() {
29
30
31
32
33
34
35
36
37
38
39
40
41
42
43
44
45
46
47
48
49
50
51
52
53
54
55
56
57
58
59
60
61
62
63
64
65
66
67
68
69
70
71
72
73
74
75
76
77
78
79
80
81
82
83
84
85
86
87
88
89
90
91
92
93
94
95
96
97
98
99
100
101
102
103
104
105
106
107
108
109
110
111
112
113
114
115
116
117
118
119
120
121
122
123
124
125
126
127
128
129
130
131
132
133
134
135
136
137
138
139
140
141
142
143
144
145
146
147
148
149
150
151
152
153
154
155
156
157
158
159
160
161
162
163
164
165
166
167
168
169
170
171
172
173
174
175
176
177
178
179
180
181
182
183
184
185
186
187
188
189
190
191
192
193
194
195
196
197
198
199
200
201
202
203
204
205
206
207
208
209
210
211
212
213
214
215
216
217
218
219
220
221
222
223
224
225
226
227
228
229
230
231
232
233
234
235
236
237
238
239
240
241
242
243
244
245
246
247
248
249
250
251
252
253
254
255
256
257
258
259
260
261
262
263
264
265
266
267
268
269
270
271
272
273
274
275
276
277
278
279
280
281
282
283
284
285
286
287
288
289
290
291
292
293
294
295
296
297
298
299
300
301
302
303
304
305
306
307
308
309
310
311
312
313
314
315
316
317
318
319
320
321
322
323
324
325
326
327
328
329
330
331
332
333
334
335
336
337
338
339
340
341
342
343
344
345
346
347
348
349
350
351
352
353
354
355
356
357
358
359
360
361
362
363
364
365
366
367
368
369
370
371
372
373
374
375
376
377
378
379
380
381
382
383
384
385
386
387
388
389
390
391
392
393
394
395
396
397
398
399
400
401
402
403
404
405
406
407
408
409
410
411
412
413
414
415
416
417
418
419
420
421
422
423
424
425
426
427
428
429
430
431
432
433
434
435
436
437
438
439
440
441
442
443
444
445
446
447
448
449
450
451
452
453
454
455
456
457
458
459
460
461
462
463
464
465
466
467
468
469
470
471
472
473
474
475
476
477
478
479
480
481
482
483
484
485
486
487
488
489
490
491
492
493
494
495
496
497
498
499
500
501
502
503
504
505
506
507
508
509
510
511
512
513
514
515
516
517
518
519
520
521
522
523
524
525
526
527
528
529
530
531
532
533
534
535
536
537
538
539
540
541
542
543
544
545
546
547
548
549
550
551
552
553
554
555
556
557
558
559
560
561
562
563
564
565
566
567
568
569
570
571
572
573
574
575
576
577
578
579
580
581
582
583
584
585
586
587
588
589
590
591
592
593
594
595
596
597
598
599
600
601
602
603
604
605
606
607
608
609
610
611
612
613
614
615
616
617
618
619
620
621
622
623
624
625
626
627
628
629
630
631
632
633
634
635
636
637
638
639
640
641
642
643
644
645
646
647
648
649
650
651
652
653
654
655
656
657
658
659
660
661
662
663
664
665
666
667
668
669
670
671
672
673
674
675
676
677
678
679
680
681
682
683
684
685
686
687
688
689
690
691
692
693
694
695
696
697
698
699
700
701
702
703
704
705
706
707
708
709
710
711
712
713
714
715
716
717
718
719
720
721
722
723
724
725
726
727
728
729
730
731
732
733
734
735
736
737
738
739
740
741
742
743
744
745
746
747
748
749
750
751
752
753
754
755
756
757
758
759
760
761
762
763
764
765
766
767
768
769
770
771
772
773
774
775
776
777
778
779
780
781
782
783
784
785
786
787
788
789
790
791
792
793
794
795
796
797
798
799
800
801
802
803
804
805
806
807
808
809
810
811
812
813
814
815
816
817
818
819
820
821
822
823
824
825
826
827
828
829
830
831
832
833
834
835
836
837
838
839
840
841
842
843
844
845
846
847
848
849
850
851
852
853
854
855
856
857
858
859
860
861
862
863
864
865
866
867
868
869
870
871
872
873
874
875
876
877
878
879
880
881
882
883
884
885
886
887
888
889
890
891
892
893
894
895
896
897
898
899
900
901
902
903
904
905
906
907
908
909
910
911
912
913
914
915
916
917
918
919
920
921
922
923
924
925
926
927
928
929
930
931
932
933
934
935
936
937
938
939
940
941
942
943
944
945
946
947
948
949
950
951
952
953
954
955
956
957
958
959
960
961
962
963
964
965
966
967
968
969
970
971
972
973
974
975
976
977
978
979
980
981
982
983
984
985
986
987
988
989
990
991
992
993
994
995
996
997
998
999
1000
1001
1002
1003
1004
1005
1006
1007
1008
1009
1010
1011
1012
1013
1014
1015
1016
1017
1018
1019
1020
1021
1022
1023
1024
1025
1026
1027
1028
1029
1030
1031
1032
1033
1034
1035
1036
1037
1038
1039
1040
1041
1042
1043
1044
1045
1046
1047
1048
1049
1050
1051
1052
1053
1054
1055
1056
1057
1058
1059
1060
1061
1062
1063
1064
1065
1066
1067
1068
1069
1070
1071
1072
1073
1074
1075
1076
1077
1078
1079
1080
1081
1082
1083
1084
1085
1086
1087
1088
1089
1090
1091
1092
1093
1094
1095
1096
1097
1098
1099
1100
1101
1102
1103
1104
1105
1106
1107
1108
1109
1110
1111
1112
1113
1114
1115
1116
1117
1118
1119
1120
1121
1122
1123
1124
1125
1126
1127
1128
1129
1130
1131
1132
1133
1134
1135
1136
1137
1138
1139
1140
1141
1142
1143
1144
1145
1146
1147
1148
1149
1150
1151
1152
1153
1154
1155
1156
1157
1158
1159
1160
1161
1162
1163
1164
1165
1166
1167
1168
1169
1170
1171
1172
1173
1174
1175
1176
1177
1178
1179
1180
1181
1182
1183
1184
1185
1186
1187
1188
1189
1190
1191
1192
1193
1194
1195
1196
1197
1198
1199
1200
1201
1202
1203
1204
1205
1206
1207
1208
1209
1210
1211
1212
1213
1214
1215
1216
1217
1218
1219
1220
1221
1222
1223
1224
1225
1226
1227
1228
1229
1230
1231
1232
1233
1234
1235
1236
1237
1238
1239
1240
1241
1242
1243
1244
1245
1246
1247
1248
1249
1250
1251
1252
1253
1254
1255
1256
1257
1258
1259
1260
1261
1262
1263
1264
1265
1266
1267
1268
1269
1270
1271
1272
1273
1274
1275
1276
1277
1278
1279
1280
1281
1282
1283
1284
1285
1286
1287
1288
1289
1290
1291
1292
1293
1294
1295
1296
1297
1298
1299
1300
1301
1302
1303
1304
1305
1306
1307
1308
1309
1310
1311
1312
1313
1314
1315
1316
1317
1318
1319
1320
1321
1322
1323
1324
1325
1326
1327
1328
1329
1330
1331
1332
1333
1334
1335
1336
1337
1338
1339
1340
1341
1342
1343
1344
1345
1346
1347
1348
1349
1350
1351
1352
1353
1354
1355
1356
1357
1358
1359
1360
1361
1362
1363
1364
1365
1366
1367
1368
1369
1370
1371
1372
1373
1374
1375
1376
1377
1378
1379
1380
1381
1382
1383
1384
1385
1386
1387
1388
1389
1390
1391
1392
1393
1394
1395
1396
1397
1398
1399
1400
1401
1402
1403
1404
1405
1406
1407
1408
1409
1410
1411
1412
1413
1414
1415
1416
1417
1418
1419
1420
1421
1422
1423
1424
1425
1426
1427
1428
1429
1430
1431
1432
1433
1434
1435
1436
1437
1438
1439
1440
1441
1442
1443
1444
1445
1446
1447
1448
1449
1450
1451
1452
1453
1454
1455
1456
1457
1458
1459
1460
1461
1462
1463
1464
1465
1466
1467
1468
1469
1470
1471
1472
1473
1474
1475
1476
1477
1478
1479
1480
1481
1482
1483
1484
1485
1486
1487
1488
1489
1490
1491
1492
1493
1494
1495
1496
1497
1498
1499
1500
1501
1502
1503
1504
1505
1506
1507
1508
1509
1510
1511
1512
1513
1514
1515
1516
1517
1518
1519
1520
1521
1522
1523
1524
1525
1526
1527
1528
1529
1530
1531
1532
1533
1534
1535
1536
1537
1538
1539
1540
1541
1542
1543
1544
1545
1546
1547
1548
1549
1550
1551
1552
1553
1554
1555
1556
1557
1558
1559
1560
1561
1562
1563
1564
1565
1566
1567
1568
1569
1570
1571
1572
1573
1574
1575
1576
1577
1578
1579
1580
1581
1582
1583
1584
1585
1586
1587
1588
1589
1590
1591
1592
1593
1594
1595
1596
1597
1598
1599
1600
1601
1602
1603
1604
1605
1606
1607
1608
1609
1610
1611
1612
1613
1614
1615
1616
1617
1618
1619
1620
1621
1622
1623
1624
1625
1626
1627
1628
1629
1630
1631
1632
1633
1634
1635
1636
1637
1638
1639
1640
1641
1642
1643
1644
1645
1646
1647
1648
1649
1650
1651
1652
1653
1654
1655
1656
1657
1658
1659
1660
1661
1662
1663
1664
1665
1666
1667
1668
1669
1670
1671
1672
1673
1674
1675
1676
1677
1678
1679
1680
1681
1682
1683
1684
1685
1686
1687
1688
1689
1690
1691
1692
1693
1694
1695
1696
1697
1698
1699
1700
1701
1702
1703
1704
1705
1706
1707
1708
1709
1710
1711
1712
1713
1714
1715
1716
1717
1718
1719
1720
1721
1722
1723
1724
1725
1726
1727
1728
1729
1730
1731
1732
1733
1734
1735
1736
1737
1738
1739
1740
1741
1742
1743
1744
1745
1746
1747
1748
1749
1750
1751
1752
1753
1754
1755
1756
1757
1758
1759
1760
1761
1762
1763
1764
1765
1766
1767
1768
1769
1770
1771
1772
1773
1774
1775
1776
1777
1778
1779
1780
1781
1782
1783
1784
1785
1786
1787
1788
1789
1790
1791
1792
1793
1794
1795
1796
1797
1798
1799
1800
1801
1802
1803
1804
1805
1806
1807
1808
1809
1810
1811
1812
1813
1814
1815
1816
1817
1818
1819
1820
1821
1822
1823
1824
1825
1826
1827
1828
1829
1830
1831
1832
1833
1834
1835
1836
1837
1838
1839
1840
1841
1842
1843
1844
1845
1846
1847
1848
1849
1850
1851
1852
1853
1854
1855
1856
1857
1858
1859
1860
1861
1862
1863
1864
1865
1866
1867
1868
1869
1870
1871
1872
1873
1874
1875
1876
1877
1878
1879
1880
1881
1882
1883
1884
1885
1886
1887
1888
1889
1890
1891
1892
1893
1894
1895
1896
1897
1898
1899
1900
1901
1902
1903
1904
1905
1906
1907
1908
1909
1910
1911
1912
1913
1914
1915
1916
1917
1918
1919
1920
1921
1922
1923
1924
1925
1926
1927
1928
1929
1930
1931
1932
1933
1934
1935
1936
1937
1938
1939
1940
1941
1942
1943
1944
1945
1946
1947
1948
1949
1950
1951
1952
1953
1954
1955
1956
1957
1958
1959
1960
1961
1962
1963
1964
1965
1966
1967
1968
1969
1970
1971
1972
1973
1974
1975
1976
1977
1978
1979
1980
1981
1982
1983
1984
1985
1986
1987
1988
1989
1990
1991
1992
1993
1994
1995
1996
1997
1998
1999
2000
2001
2002
2003
2004
2005
2006
2007
2008
2009
2010
2011
2012
2013
2014
2015
2016
2017
2018
2019
2020
2021
2022
2023
2024
2025
2026
2027
2028
2029
2030
2031
2032
2033
2034
2035
2036
2037
2038
2039
2040
2041
2042
2043
2044
2045
2046
2047
2048
2049
2050
2051
2052
2053
2054
2055
2056
2057
2058
2059
2060
2061
2062
2063
2064
2065
2066
2067
2068
2069
2070
2071
2072
2073
2074
2075
2076
2077
2078
2079
2080
2081
2082
2083
2084
2085
2086
2087
2088
2089
2090
2091
2092
2093
2094
2095
2096
2097
2098
2099
2100
2101
2102
2103
2104
2105
2106
2107
2108
2109
2110
2111
2112
2113
2114
2115
2116
2117
2118
2119
2120
2121
2122
2123
2124
2125
2126
2127
2128
2129
2130
2131
2132
2133
2134
2135
2136
2137
2138
2139
2140
2141
2142
2143
2144
2145
2146
2147
2148
2149
2150
2151
2152
2153
2154
2155
2156
2157
2158
2159
2160
2161
2162
2163
2164
2165
2166
2167
2168
2169
2170
2171
2172
2173
2174
2175
2176
2177
2178
2179
2180
2181
2182
2183
2184
2185
2186
2187
2188
2189
2190
2191
2192
2193
2194
2195
2196
2197
2198
2199
2200
2201
2202
2203
2204
2205
2206
2207
2208
2209
2210
2211
2212
2213
2214
2215
2216
2217
2218
2219
2220
2221
2222
2223
2224
2225
2226
2227
2228
2229
2230
2231
2232
2233
2234
2235
2236
2237
2238
2239
2240
2241
2242
2243
2244
2245
2246
2247
2248
2249
2250
2251
2252
2253
2254
2255
2256
2257
2258
2259
2260
2261
2262
2263
2264
2265
2266
2267
2268
2269
2270
2271
2272
2273
2274
2275
2276
2277
2278
2279
2280
2281
2282
2283
2284
2285
2286
2287
2288
2289
2290
2291
2292
2293
2294
2295
2296
2297
2298
2299
2300
2301
2302
2303
2304
2305
2306
2307
2308
2309
2310
2311
2312
2313
2314
2315
2316
2317
2318
2319
2320
2321
2322
2323
2324
2325
2326
2327
2328
2329
2330
2331
2332
2333
2334
2335
2336
2337
2338
2339
2340
2341
2342
2343
2344
2345
2346
2347
2348
2349
2350
2351
2352
2353
2354
2355
2356
2357
2358
2359
2360
2361
2362
2363
2364
2365
2366
2367
2368
2369
2370
2371
2372
2373
2374
2375
2376
2377
2378
2379
2380
2381
2382
2383
2384
2385
2386
2387
2388
2389
2390
2391
2392
2393
2394
2395
2396
2397
2398
2399
2400
2401
2402
2403
2404
2405
2406
2407
2408
2409
2410
2411
2412
2413
2414
2415
2416
2417
2418
2419
2420
2421
2422
2423
2424
2425
2426
2427
2428
2429
2430
2431
2432
2433
2434
2435
2436
2437
2438
2439
2440
2441
2442
2443
2444
2445
2446
2447
2448
2449
2450
2451
2452
2453
2454
2455
2456
2457
2458
2459
2460
2461
2462
2463
2464
2465
2466
2467
2468
2469
2470
2471
2472
2473
2474
2475
2476
2477
2478
2479
2480
2481
2482
2483
2484
2485
2486
2487
2488
2489
2490
2491
2492
2493
2494
2495
2496
2497
2498
2499
2500
2501
2502
2503
2504
2505
2506
2507
2508
2509
2510
2511
2512
2513
2514
2515
2516
2517
2518
2519
2520
2521
2522
2523
2524
2525
2526
2527
2528
2529
2530
2531
2532
2533
2534
2535
2536
2537
2538
2539
2540
2541
2542
2543
2544
2545
2546
2547
2548
2549
2550
2551
2552
2553
2554
2555
2556
2557
2558
2559
2560
2561
2562
2563
2564
2565
2566
2567
2568
2569
2570
2571
2572
2573
2574
2575
2576
2577
2578
2579
2580
2581
2582
2583
2584
2585
2586
2587
2588
2589
2590
2591
2592
2593
2594
2595
2596
2597
2598
2599
2600
2601
2602
2603
2604
2605
2606
2607
2608
2609
2610
2611
2612
2613
2614
2615
2616
2617
2618
2619
2620
2621
2622

```

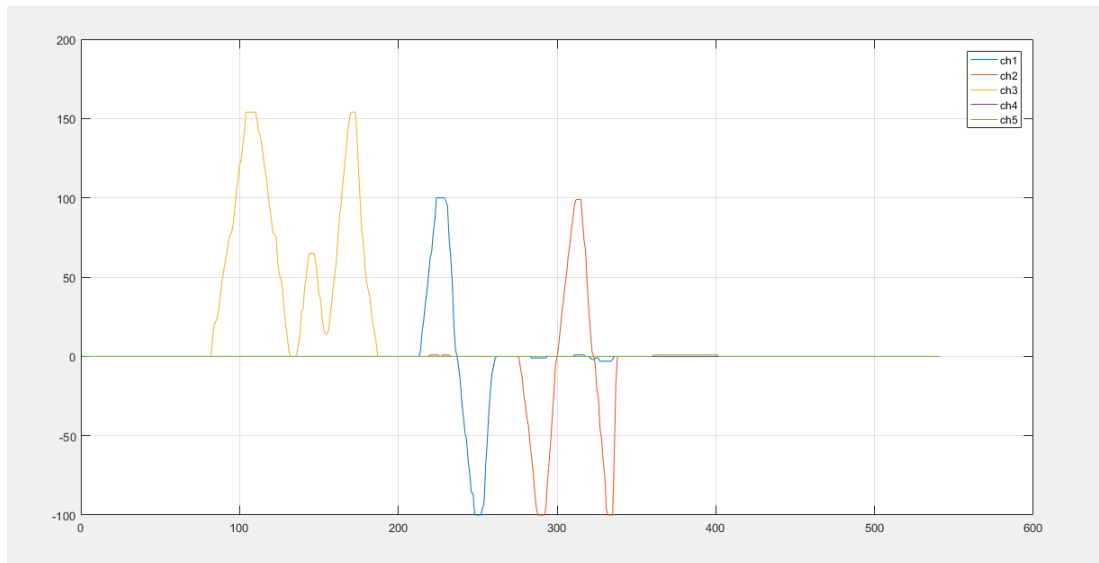


Figure III.18 : La variation des valeurs des chaines.

III.2.3 Expériences sur l'équipement

Dans cette expérience nous allons utiliser un seul moteur, donc nous allons utiliser la batterie Li -Po avec un contrôleur de vitesse et une hélice, maintenant que nous avons acheté tout l'équipement il est temps de le tester. Cependant, nous devons d'abord calibrer les ESC afin de pouvoir les utiliser. Le but de cette expérience est de :

- Comprendre le fonctionnement du contrôleur de vitesse électrique (ESC).
- Observer la vitesse du moteur.
- Mesurer le poids maximum que le moteur peut soulever.

Dans cette expérience, nous utiliserons :

L'expérience précédent :

- Utilisation des sorties PWM comme entrées de l'Arduino Mega 2560.
- **Matériel :**
 - Carte Arduino Mega 2560.
 - Transmetteur FS-i6X et le récepteur FS-iA10B
 - Batterie Li Po 3300mAh 11.1V 25C.
 - Moteur brushless 2212 1000KV.

- Contrôleur de vitesse électrique 30A
- Fil de liaison
- Hélices 10×45
- **Logiciel :**
 - IDE Arduino
 - MATLAB

Pour cette expérience, nous allons utiliser une batterie LI Po 3s qui comporte 3 éléments et 11,1V. Nous pouvons donc nous attendre à ce que mon moteur atteigne un RPM maximum de 11100.

L'ESC 30A que nous allons utiliser pour cette expérience et qui correspond aux exigences du moteur. D'un côté, l'ESC a trois fils qui contrôlent les trois phases du moteur.

D'un autre côté, il y a deux fils, VCC et GND, pour l'alimentation.

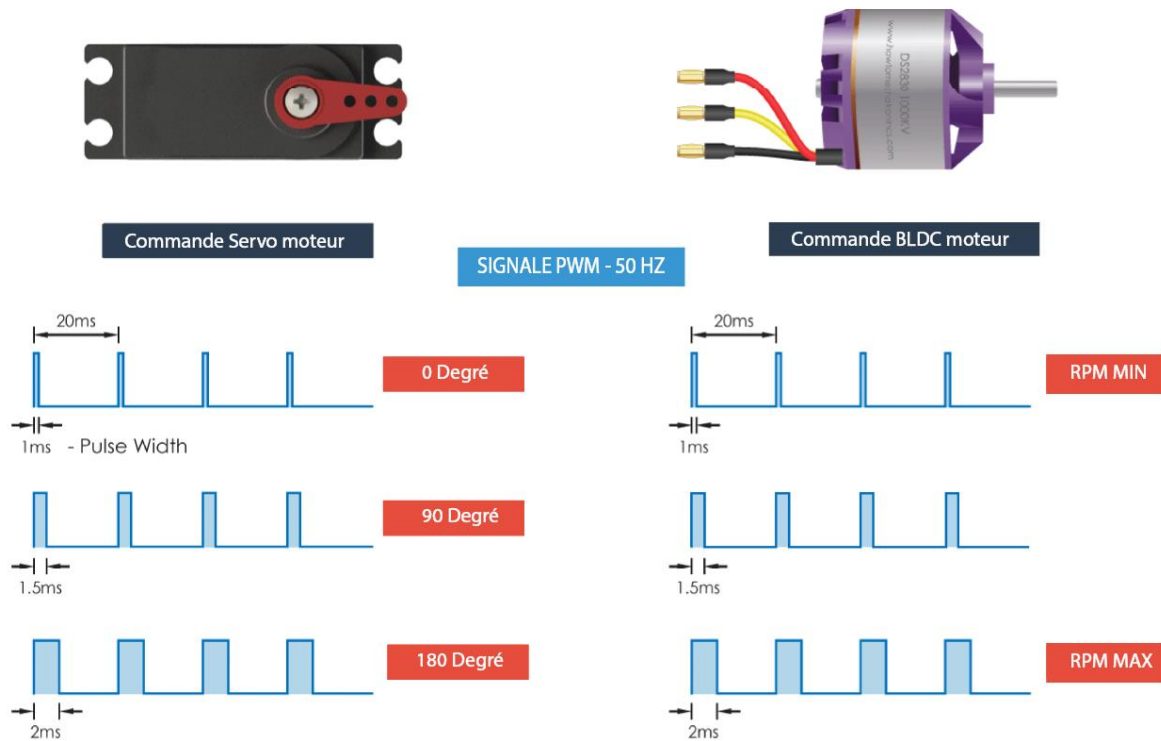
Il y a aussi un autre ensemble de trois fils qui sortent de l'ESC et c'est la ligne de signal, +5V et la masse. Cette caractéristique de l'ESC est appelée circuit éliminateur de batterie.

Comme son nom l'indique, ce circuit élimine le besoin d'une batterie séparée pour le microcontrôleur. Avec cela, l'ESC fournit 5V régulé qui peut être utilisé pour alimenter notre Arduino.

Les ESC se comportent comme des servomoteurs. Aussi, nous allons devoir générer un signal PWM bien particulier pour les contrôleurs

Nous pouvons remarquer ici que cette connexion est en fait la même que celle que nous voyons sur les servomoteurs.

Ainsi, contrôler un moteur brushless en utilisant un ESC et Arduino est aussi simple que de contrôler un servo en utilisant Arduino. Les ESC utilisent le même type de signal de contrôle que



les servos, à savoir le signal PWM 50Hz standard.

Figure III.19 : Difference entre Servo moteur et BLDC moteur.

Ceci est très pratique, car par exemple, lors de la construction d'un drone RC, nous avons généralement besoin de servos et de moteurs brushless et, de cette façon, nous pouvons les contrôler facilement avec le même type de contrôleur.

Ainsi, en utilisant l'Arduino, il suffit de générer le signal PWM 50Hz et en fonction de la largeur des impulsions ou de la durée de l'état haut, qui doit varier entre 1 milliseconde et 2 millisecondes, l'ESC fera passer le moteur de la vitesse minimale à la vitesse maximale.

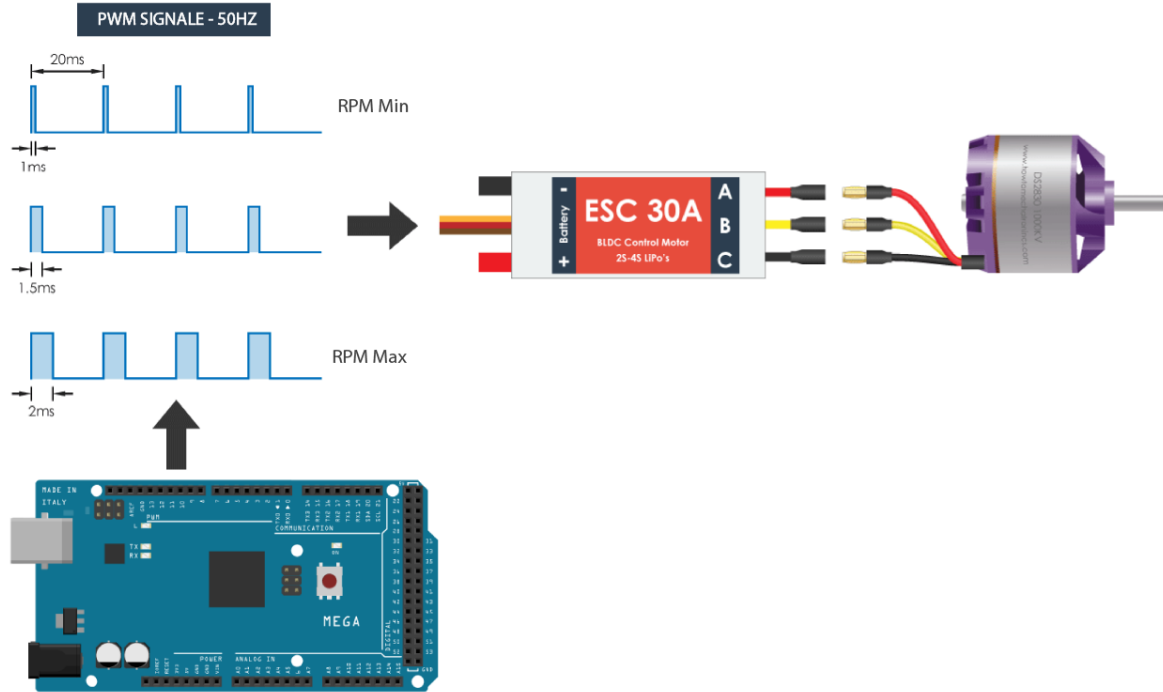


Figure III.20 : Branchement Arduino, ESC, BLDC moteur et PWM signale.

III.2.3.1 Calibrage de l'ESC

Les ESC se comportent comme des servomoteurs. Aussi, nous allons devoir générer un signal PWM bien particulier pour les contrôleurs

Enfin, expliquons comment fonctionne la calibration de l'ESC. Chaque ESC a ses propres points hauts et bas, et ils peuvent varier légèrement. Par exemple, le point bas peut être de 1,2 millisecondes et le point haut de 1,9 millisecondes. Dans ce cas, notre accélérateur ne fera rien dans les premiers 20 % jusqu'à ce qu'il atteigne le point bas de 1,2 millisecondes. Ce point bas de 1,2 milliseconde. Voir la Figure ci-dessous.

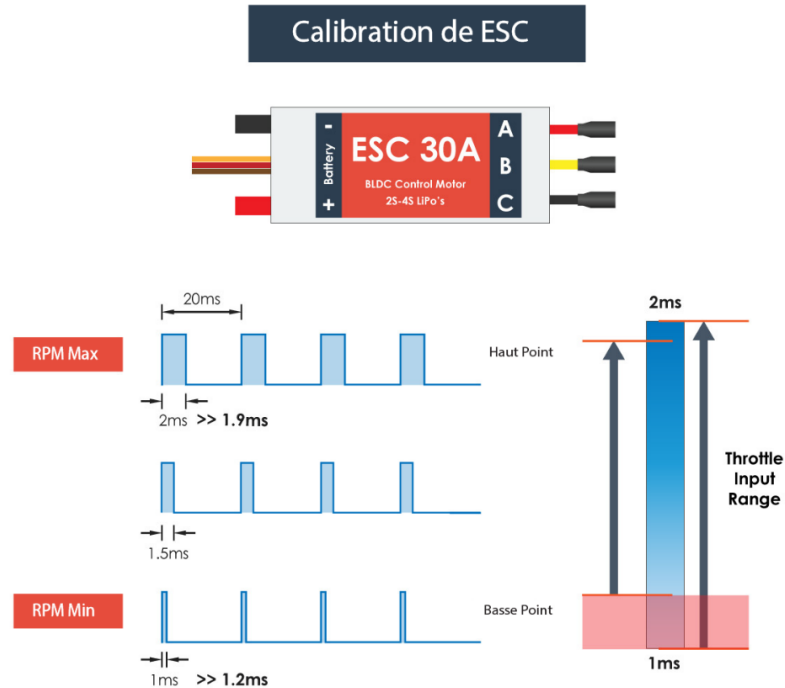


Figure III.21 : Calibration de l'ESC.

Pour Branchement dans cette expérience voir le tableau et la Figure ci-dessous.

Arduino 2560	Mega	FS-iA10B
GND		GND
5V		VCC
Pin analogique A0		C3

Tableau III.4 : Branchement Arduino Mega 2560 avec FS-iA10B.

Arduino Mega 2560	ESC
GND	GND
5V	VCC
Pin numérique 9	La ligne de signal

Tableau III.5 : Branchement Arduino Mega 2560 avec ESC.

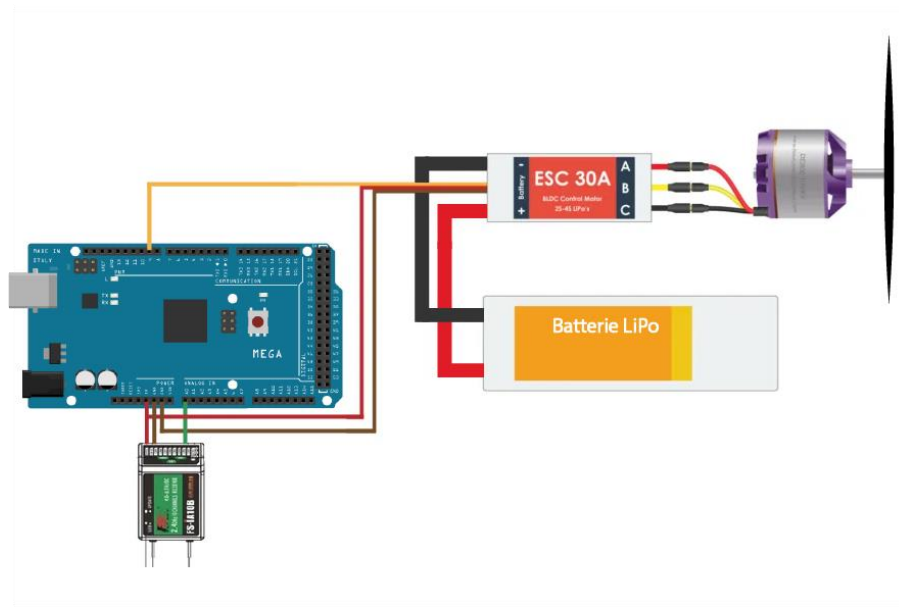


Figure III.22 : Branchement Arduino mega 2560 avec ESC et FS-iA10B

La figure suivante donne le programme Arduino qui nous permet de contrôler la vitesse de notre moteur avec le transmetteur FS-i6X.

```

test
1 #include<Servo.h>
2 #define CH3 5
3 Servo esc;
4
5 // Integers to represent values from sticks and pots
6 int ch3Value;
7
8 // Read the number of a specified channel and convert to the range provided.
9 // If the channel is off, return the default value
10 int readChannel(int channelInput, int minLimit, int maxLimit, int defaultValue){
11     int ch = pulseIn(channelInput, HIGH, 30000);
12     if (ch < 100) return defaultValue;
13     return map(ch, 1000, 2000, minLimit, maxLimit);
14 }
15
16
17 void setup(){
18     esc.attach(9); //Specify the esc signal pin,Here as D8
19
20     esc.writeMicroseconds(1000); //initialize the signal to 1000
21
22     // Set up serial monitor
23     Serial.begin(115200);
24
25     // Set pin as input
26
27     pinMode(CH3, INPUT);
28
29 }
Enregistrement terminé.

```

Figure III.23 : Programme Arduino pour contrôler la vitesse de moteur avec FS-i6X

La figure ci-dessous montre la vitesse de notre moteur et la consigne que nous avons donnée.

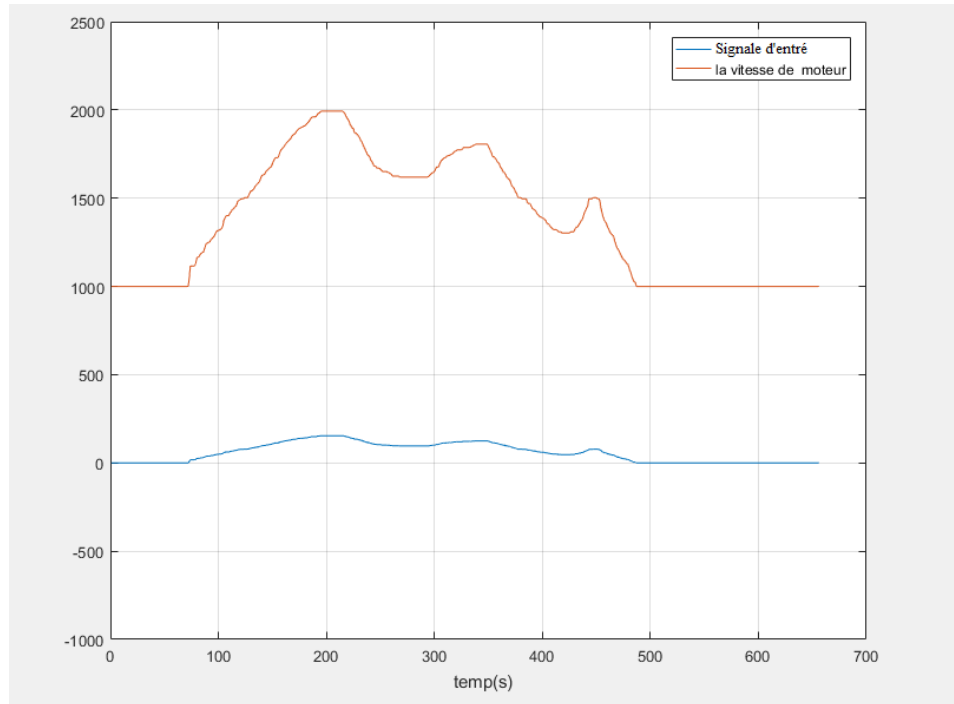


Figure III.24 : La vitesse de notre moteur et l'entrée que nous avons donnée.

III.3 Conclusion

Ce chapitre a été consacré à l'explication des opérations et des tests des composants qui composent notre drone. Nous avons programmé avec l'Arduino pour tester nos composants, nous avons également programmé avec Matlab dans l'une de nos expériences, nous avons également implémenté un code Matlab pour tracer les courbes des valeurs que nous avons mesurées avec ces composants. Nous avons trouvé que l'équipement que nous avons utilisé pour notre projet a très bien fonctionné et nous pouvons l'utiliser pour réaliser le drone.

Chapitre IV :

Commande d'un drone type quadrirotor par régulateur PID

Chapitre IV :

Commande d'un drone type quadrirotor par régulateur PID

IV.1 Introduction

Afin de concevoir un contrôleur de vol, il faut bien comprendre la dynamique des mouvements de l'avion. Cette compréhension est non seulement nécessaire pour la conception du contrôleur, mais également pour garantir que les simulations du comportement du véhicule lors de l'application du contrôleur sont aussi réalistes que possible. Le quadrirotor est classé dans la catégorie des systèmes de vol les plus complexes en raison du nombre d'effets physiques affectant sa dynamique, à savoir les effets aérodynamiques, la gravité, les effets gyroscopiques, les frottements et le moment d'inertie.

Notre but dans ce chapitre est de réaliser un véhicule aérien de type Quad X et un système de régulation de type PID, ce chapitre est dédié principalement à effectuer des tests réels et présenter les résultats de rotation du drone selon les trois axes x , y et z . Nous allons également visualiser les réponses de variation des angles de rotation en déduisant sur la stabilité du système.

IV.2 Le régulateur PID du quadrirotor

Le contrôle par PID est la méthode la plus simple à programmer sur un processeur et elle est souvent employée pour les asservissements [24].

Cette méthode est basée sur une structure qui délivre un signal de commande à partir de l'erreur existante entre la référence et la mesure de son intégrale et de sa dérivée **figure IV.1**

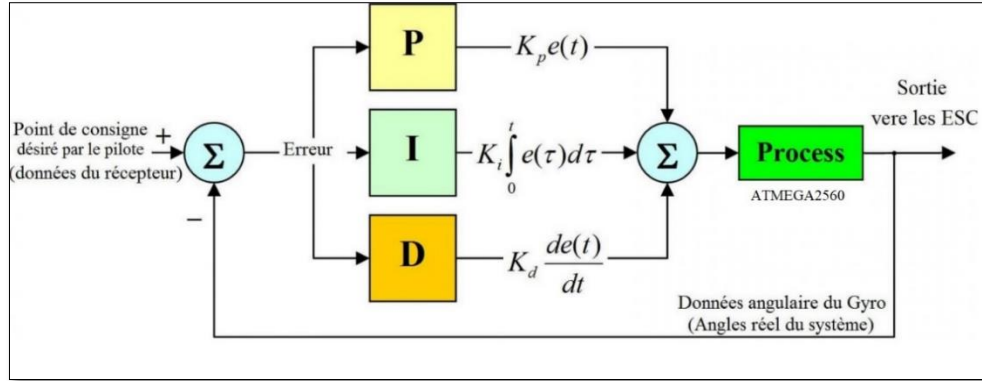


Figure IV.1 : Schéma de la boucle de régulateur PID.

Le contrôle par PID est souvent utilisé dans l'industrie car il est très simple à mettre en place et s'avère efficace à réduire les erreurs pour la plupart des systèmes réels non linéaires.

Dans notre cas le principe de base du régulateur PID est simple, lorsque le drone se trouve déséquilibré à cause du vent ou de la turbulence.

L'algorithme envoie une commande vers les ESC qui provoque une réaction dont le sens est opposé à celui qui a conduit au déséquilibre, afin que le drone retrouve l'état souhaité.

Plus précisément, il faut garder les valeurs mesurées du MPU6050 (données angulaires réel du système) identique au point de consigne désiré par le pilote (données d'entrée du récepteur de la routine d'interruption) voire la **Figure IV.2**.

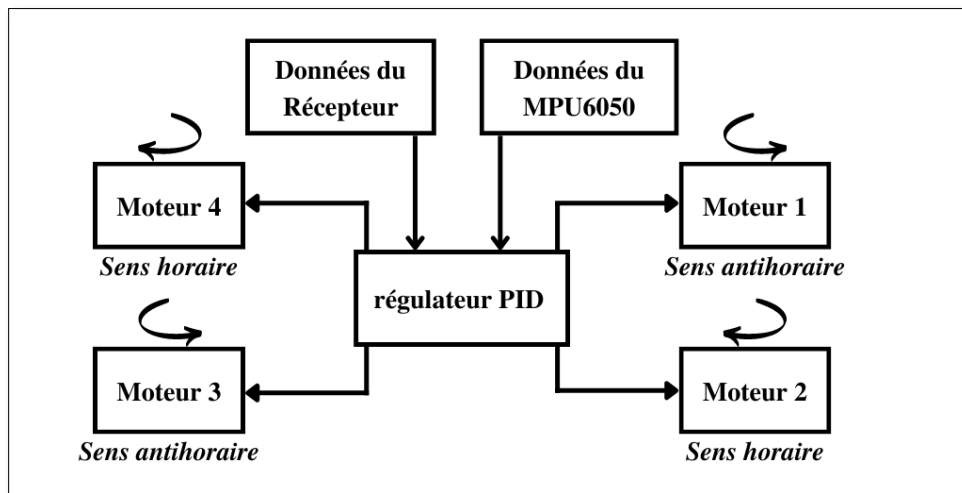


Figure IV.2: Schéma montre le principe du PID utilisé dans ce régulateur de vol.

Par exemple si le pilote ne veut aucun mouvement, le taux angulaire Gyro doit être également nul, donc l'erreur du système « ε » pour un axe de mouvement est la différence :

$$\varepsilon = (\text{Données d'entrée du MPU6050} - \text{Données d'entrées du récepteur}).$$

Le régulateur PID calcule les corrections nécessaires pour les trois axes (Pitch, Roll et Yaw).

Les consignes pour le régulateur PID sont les signaux d'entrée du récepteur et les variables des mouvements angulaires mesurées par le gyroscope.

Les sorties du régulateur PID sont destinées pour contrôler la vitesse des moteurs du quadrirotor afin d'assurer l'équilibre.

L'utilisation des opérateurs mathématiques (Proportionnel, Intégrateur et Dérivateur) permet de produire une sortie de commande adéquate pour le système.

Le modèle mathématique du régulateur PID est décrit comme suit :

$$u(t) = k_p(t) + k_i \int_0^t e(\tau) d\tau + k_d \frac{de(t)}{dt} \quad , \text{ tel que :}$$

- $e(\tau)$: L'erreur du système.
- $u(t)$: Variable de sortie du PID.
- k_p : Gain proportionnel.
- k_i : Gain d'intégration.
- k_d : Gain de dérivation.

Effet proportionnel $k_p(t)$

Ce terme est proportionnel à l'erreur, il contribue à la stabilité et à la réactivité et permet une augmentation du temps de réponse du système, en fait le régulateur seul n'est pas tout à fait suffisant pour stabiliser le drone [25].

Effet d'intégral $k_i \int_0^t e(\tau) d\tau$

Ce terme est proportionnel à l'intégrale de l'erreur, il permet au système d'atteindre le point de consigne désiré, permet aussi d'éliminer l'erreur statique du système et de diminuer l'impact des perturbations [25], il en résulte un système précis.

Effet de la dérivée $k_d \frac{de(t)}{dt}$

Ce terme est proportionnel à la dérivée de l'erreur, il permet d'accélérer la réponse du système et d'éliminer les oscillations autour du point de consigne [25].

IV.2.1 Saturation du régulateur PID et choix des gains [26]

Pour empêcher le PID de devenir incontrôlable, il est nécessaire de limiter la sortie PID à une valeur critique qui est le point de saturation du système et ça se détermine selon les caractéristiques du système et l'algorithme du régulateur PID utilisé.

Cette limitation est souvent appliquée pour la sortie globale du régulateur PID.

Chaque quadrirotor est caractérisé par ces propres gains PID, généralement les gains PID varient d'un drone à l'autre en fonction du poids et du hardware utilisé.

La mise en œuvre de la commande PID exige d'abord une sélection adéquate des gains, k_i et k_d car la réponse de la sortie PID sera influencée en agissant sur ces trois gains.

Pour notre système les gains seront déterminés expérimentalement.

IV.2.2 Algorithme PID du régulateur de vol

Dans le programme du régulateur de vol, trois régulateurs PID sont utilisés pour les trois axes de mouvement essentiels (Pitch, Roll et Yaw).

Les trois axes de mouvement essentiels	Angle de rotation Selon l'axe	Le nom de l'angle
Roll	X	φ
Pitch	Y	θ
Yaw	Z	ψ

Tableau IV.1 : Appellation des rotations pour différents degrés de liberté.

Au début, il est nécessaire de transférer les variables des canaux d'entrée du récepteur (de la routine d'interruption) en degrés / seconde pour avoir les trois points de consigne de mouvement, identiques en grandeurs que les données d'entrée du MPU6050.

L'organigramme suivant explique l'étape de conversion des variables du récepteur en degrés / seconde pour un axe de mouvement « Roll », la même routine se répète pour les deux autres axes « Pitch et Yaw ».

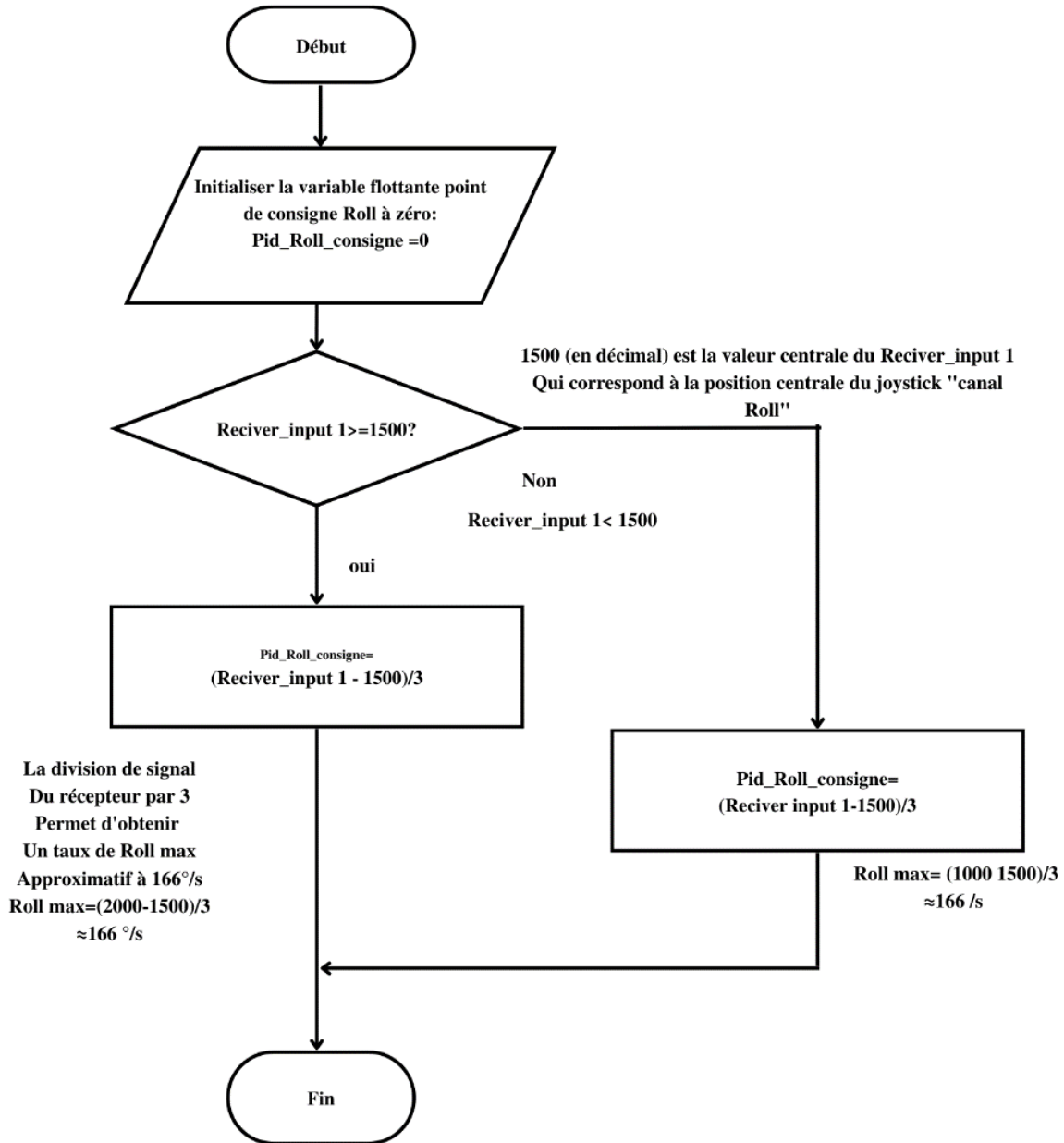


Figure IV.3: L'organigramme qui explique l'étape de conversion des variables du récepteur en degrés / seconde pour un axe de mouvement « Roll »

Les entrées du régulateur PID {données du récepteur : (Pid_Roll_consigne, Pid_Pitch_consigne, Pid_Yaw_consigne) et les données angulaires du MPU6050 :

(angle_roll, angle_pitch, angle_yaw)} sont connues, reste à déterminer les variables de sortie PID pour les mouvements (Pitch, Roll et Yaw).

Dans le sous-programme PID les trois parties (proportionnel, intégral et dérivé) sont calculées simultanément et combinées à un seul signal de sortie pour chaque axe de mouvement voire la **Figure IV.4**

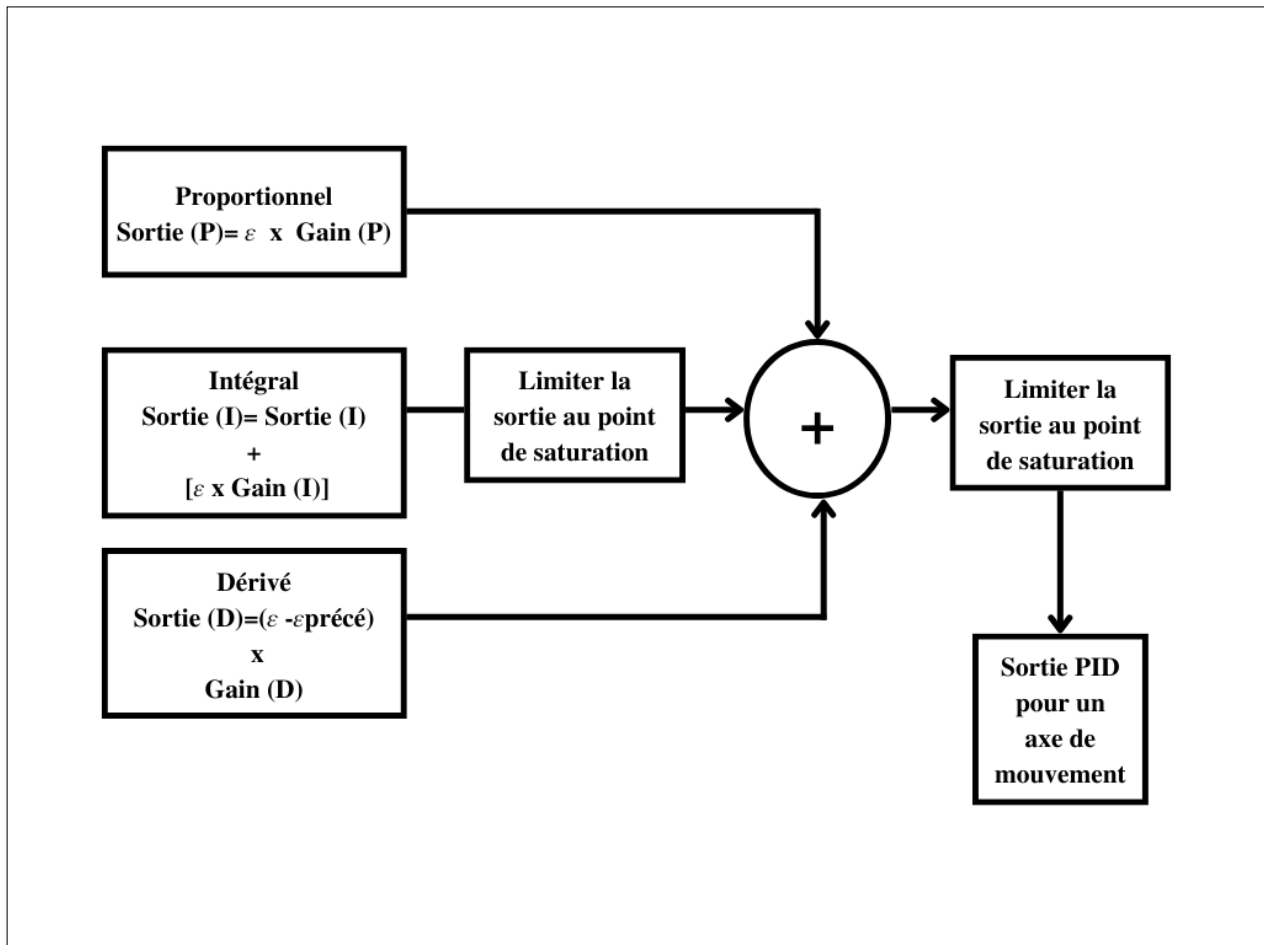


Figure IV.4 : L'algorithme utilisé dans le sous-programme PID du régulateur de vol pour un axe de mouvement.

IV.2.3 Sous-programme de sortie PID

L'organigramme ci-dessous explique le calcul de la sortie PID pour un axe de mouvement « Roll ».

La même routine est répétée dans la sous-routine PID du régulateur de vol pour les deux autres axes «Pitch et Yaw ».

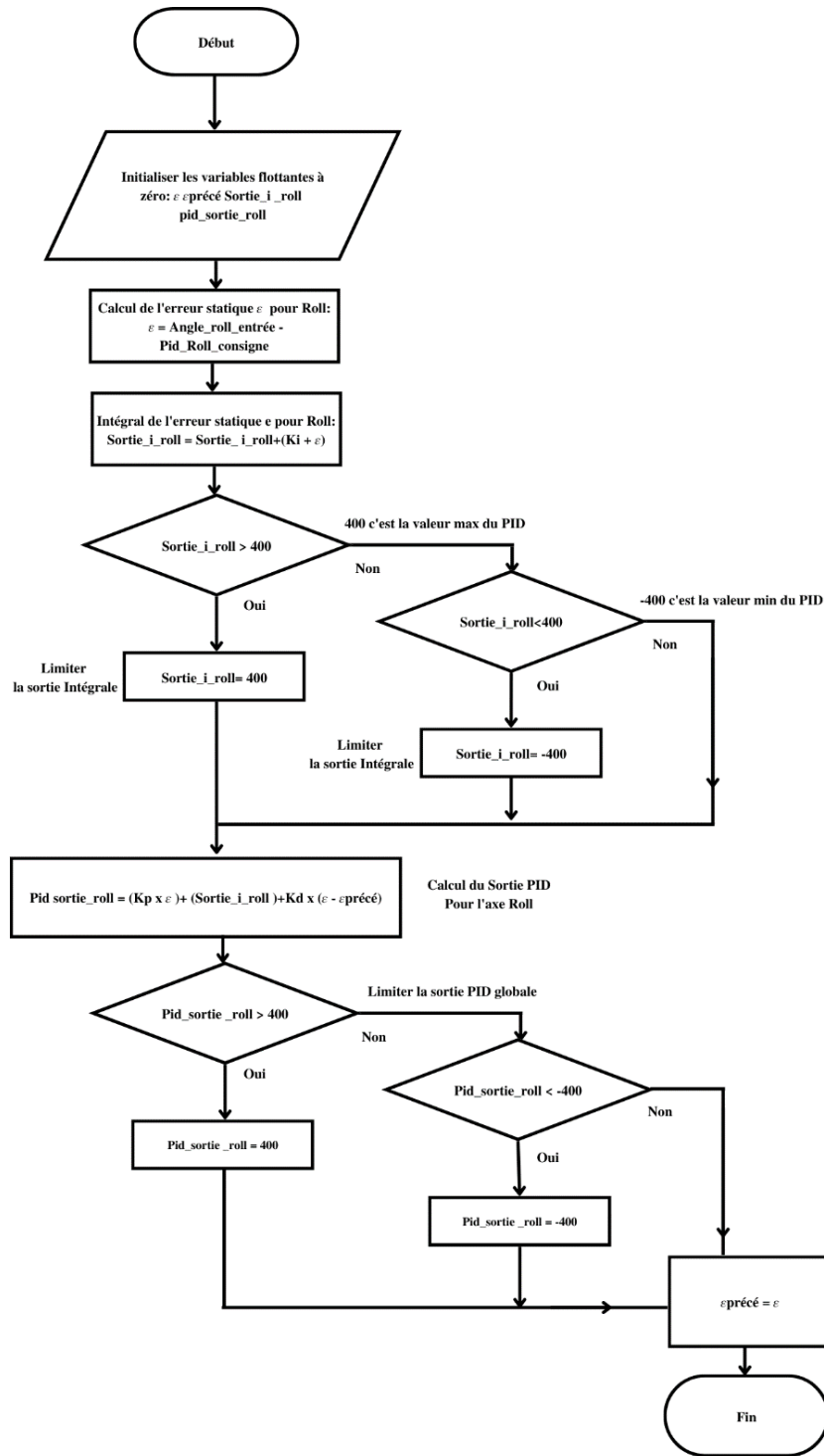


Figure IV.5: L'organigramme qui explique le calcul du sorties PID pour un axe de mouvement « Roll ».

IV.2.4 Contrôle des ESC

Dans cette partie, le but est de générer les impulsions de commande à envoyer pour les quatre ESCs en fonction des variables de sorties PID, afin de contrôler la vitesse des moteurs du quadrirotor.

Les ESC sont connectés à l'Arduino Uno du régulateur de vol via les broches digitales (B4, B5, B6 et B7) du registre PORTB.

Au début du programme principal, ces broches numériques doivent être déclarés en tant que sorties.

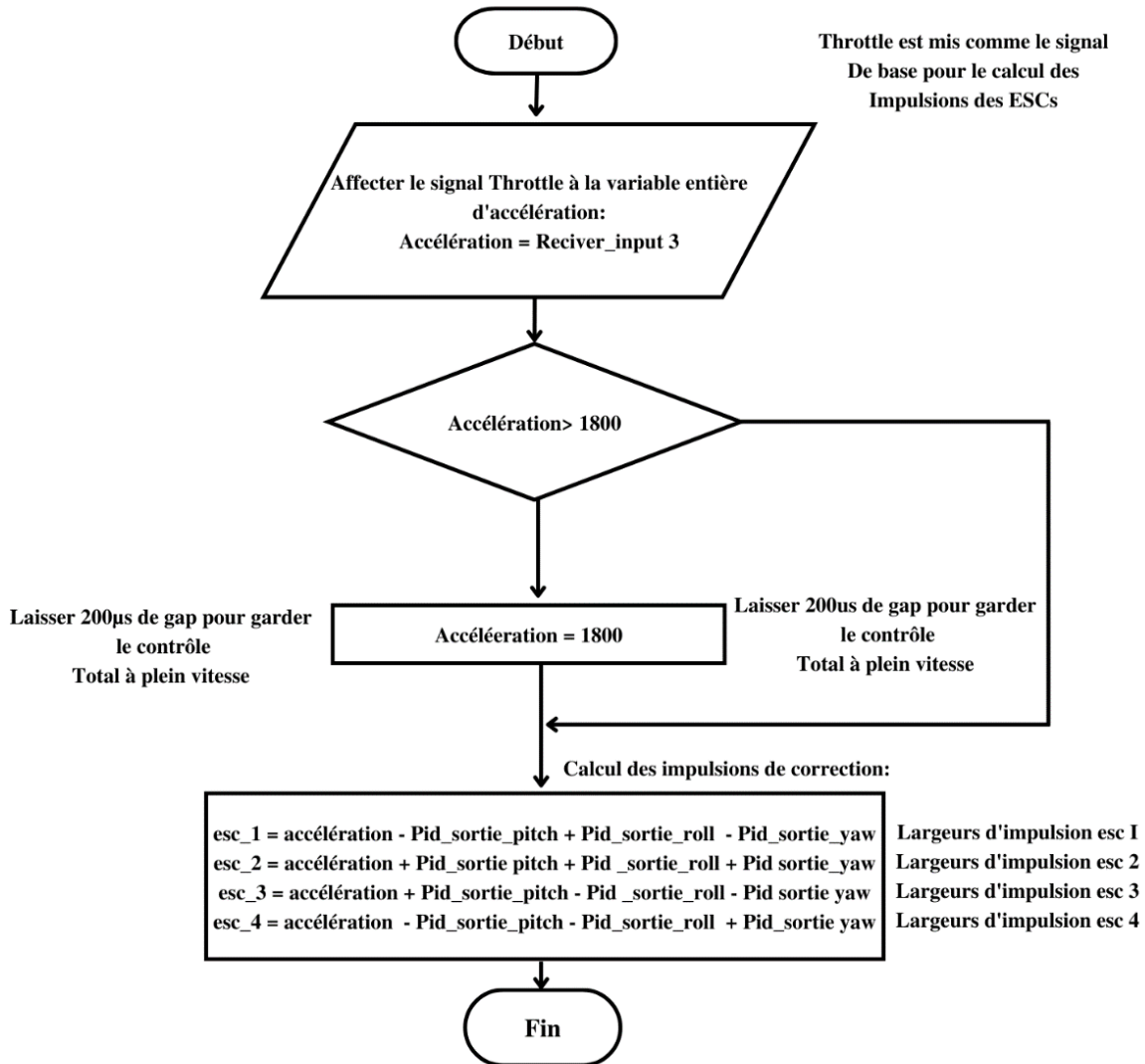
Après avoir calculé les variables de sortie du régulateur PID pour les axes de commande (Pitch, Roll et Yaw), il devient possible de calculer les largeurs d'impulsion à fournir pour les ESC toutes les 4 ms afin de corriger la position du quadrirotor.

Au début, il est nécessaire de déclarer le signal d'entrée du récepteur Throttle « Reciver_input 3 » comme la variable de base d'accélération.

Pour assurer que les trois régulateurs PID ont suffisamment d'espace pour corriger le quadrirotor à pleine vitesse, la variable d'accélération est limitée à 1800 μ s au lieu de 2000 μ s.

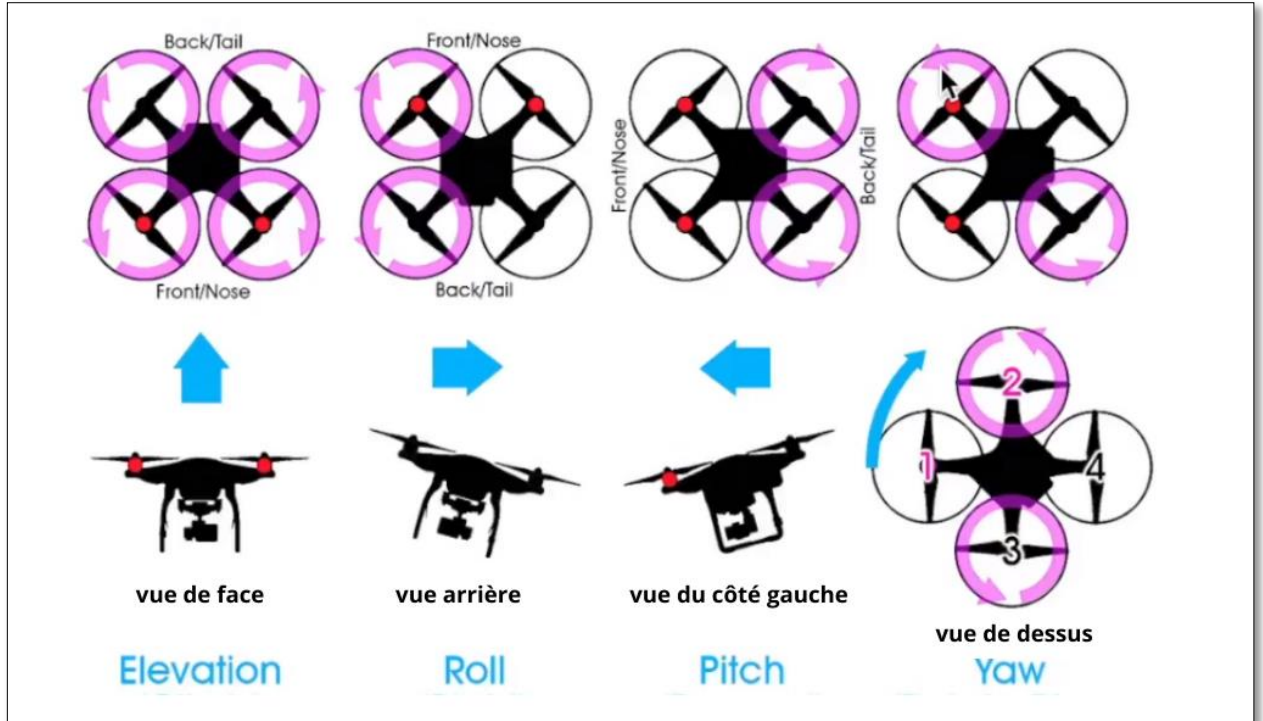
De cette façon, chaque régulateur PID peut ajouter 200 μ s supplémentaires en cas de besoin.

Pour calculer les impulsions des ESC, il suffit de combiner les variables de sorties PID séparées (Pitch, Roll et Yaw) et de les ajouter ou les soustraire à la variable d'accélération suivant cet algorithme :



La Figure IV.6: L'algorithme calculer les impulsions des ESC.

La Figure représente les axes (Roll, Pitch et Yaw) et les mouvements du quadrirotor.



La Figure IV.7: Les axes (Roll, Pitch et Yaw) et les mouvements du quadrirotor.

Pour l'axe de Roulis, si le quadrirotor penche vers la droite en raison du vent ou de la turbulence, cela conduit le Gyro d'envoyer un angle de roulis positif (Données d'entrée Roll du Gyro > 0).

Lorsque le pilote ne veut aucun mouvement (le joystick de l'émetteur est en position centrale).

C'est à dire la variable roulis du récepteur est de $0^\circ / \text{sec}$ ($\text{Pid_Roll_consigne} = 0^\circ$) implique que la sortie roulis du régulateur PID est positive ($\text{Pid_sortie_roll} > 0$) car l'erreur $\varepsilon > 0$.

Donc pour que le quadrirotor puisse revient à l'état stable 0° , la vitesse du moteur avant droit doit augmenter et ceci est fait en ajoutant la sortie du régulateur PID Roll (Pid_sortie_roll) à l'impulsion de l'ESC 1.

Le même principe pour le tangage (Pitch), lorsque le front du drone est incliné en bas à cause d'une perturbation ou du vent, le Gyro envoie un angle de tangage négatif et la sortie du régulateur PID Pitch devienne négative ($\text{Pid_sortie_pitch} < 0$).

En soustrayant la sortie du PID Pitch de l'impulsion ESC 1 implique que la vitesse du moteur 1 s'augmente et le quadrirotor retrouve l'état stable 0° .

Cela est illustré dans la Figure IV.8 pour l'axe de roulis (Roll) et la même chose pour le tangage (Pitch).

Le même principe se répète pour chaque combinaison moteur / ESC.

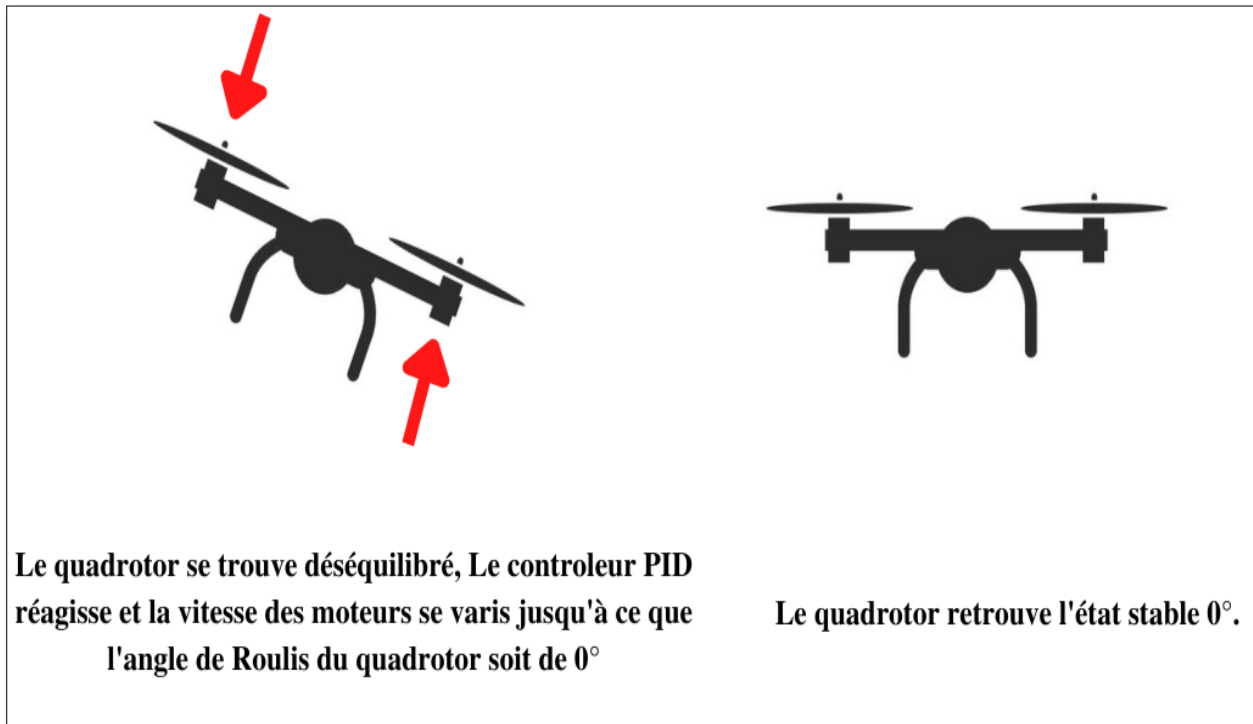


Figure IV.8: L'algorithme du PID conduit le quadrirotor de revenir À l'état voulu en agissant sur les ESCs.

Selon les dernières instructions citées dans l'algorithme en haut :

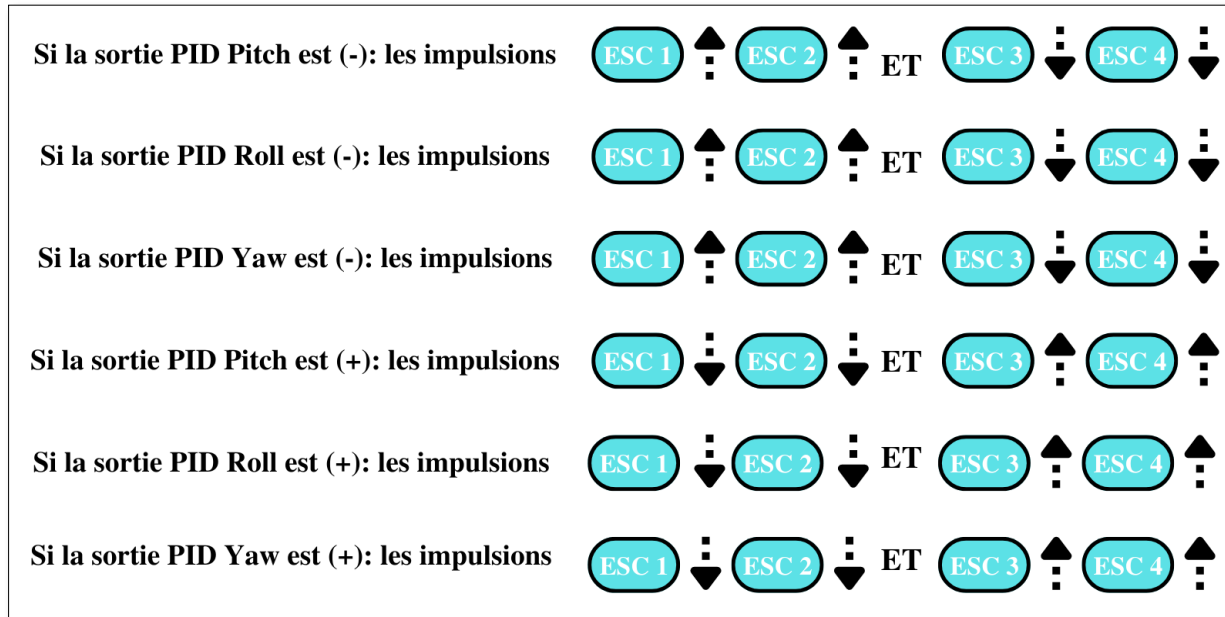


Figure IV.9: Les impulsions des contrôleurs de vitesse esc avec le signe de régulateur.

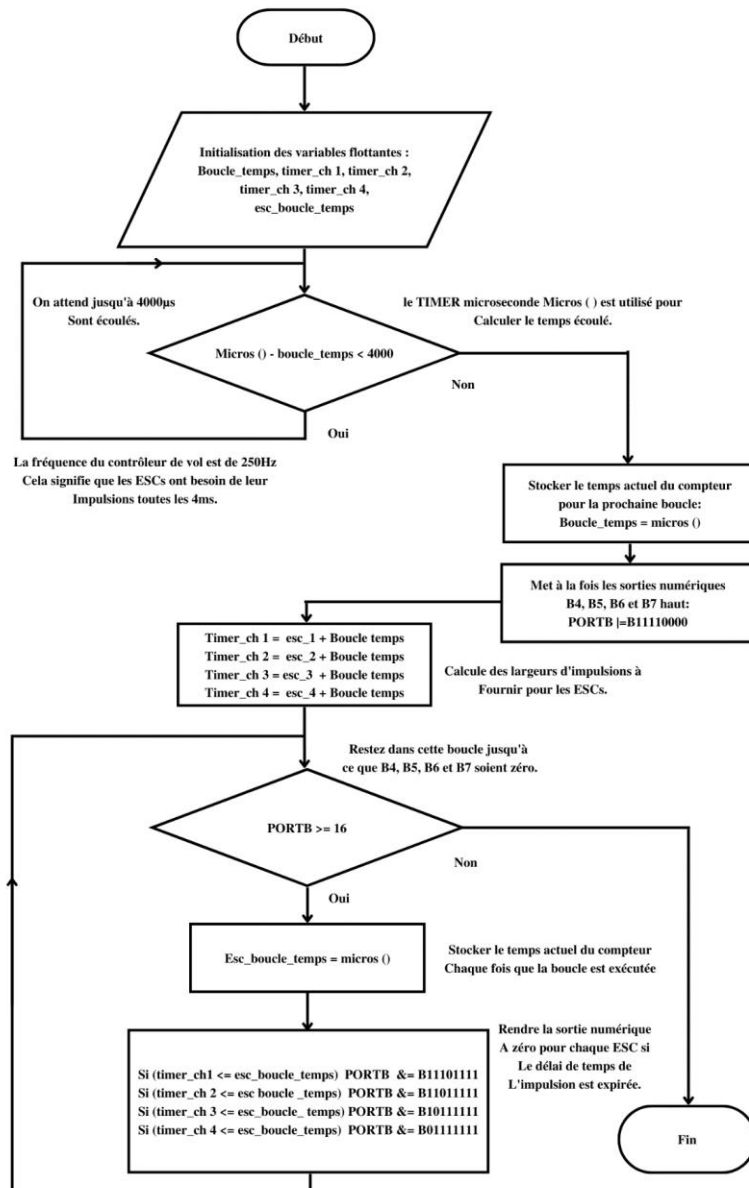
L'étape suivante consiste à envoyer les impulsions calculées aux ESC, ceci est fait en maintenant les sorties numériques du PORTB (B4, B5, B6 et B7) à un niveau logique haut avec un délai de temps correspondant aux largeurs d'impulsions ESC calculées précédemment.

Pour avoir le front descendant des impulsions, il suffit d'exécuter une boucle « while » jusqu'à ce que les quatre bits de poids fort du registre PORTB soient à zéro, ça signifie la fin de l'impulsion ESC.

Pour s'assurer que la fréquence des signaux PWM est de 250 Hz, un temporisateur de boucle vide est employé pour s'exécuter jusqu'à ce que les 4 ms soient expirées.

Finalement, il en résulte quatre trains d'impulsions PWM synchronisé avec une fréquence de 250 Hz.

Toutes les données pour contrôler les moteurs sont disponibles. L'organigramme suivant



explique l'algorithme utilisé pour générer des impulsions PWM aux ESC:

Figure IV.10: . L'organigramme qui explique l'algorithme utilisé pour générer des impulsions PWM aux ESC.

IV.3 Résultats et analyse

IV.3.1 Conception du matériel

Dans cette étude, le quadrirotor est assemblé à l'aide de composants relativement peu coûteux qui sont les suivants qui sont le microcontrôleur, le capteur IMU MPU6050, le moteur BLDC, l'ESC, l'unité de contrôle à distance, la batterie et le cadre. Ces composants sont intégrés dans un matériel de quadrirotor dont le poids total est de 3.75 Kg. Le microcontrôleur utilisé est ATmega2560 avec une vitesse d'horloge de 16 MHz, intégré sur la carte Arduino Uno. Carte Arduino Uno. Nous utilisons le capteur IMU MPU-6050 avec 6 degrés de liberté (DOF) qui est une combinaison de gyroscope et d'acétomètre. Une combinaison de gyroscope et d'accéléromètre. Quatre unités de A2212 BLDC moteur 1000 KV est également utilisé comme l'actionneur combiné avec 4 unités d'hélice avec un rayon de 4,5 pouces et Simonk ESC 30 A comme contrôleur de vitesse. Le transmetteur Flysky FS-i6 avec une fréquence radio de 2.4 GHz est communiquée avec son récepteur sur le quadrirotor comme régulateur pilote. Nous utilisons 3 cellules Lithium-Polymère (LiPo) qui ont une tension de 3.7 V dans chaque cellule. Enfin, le châssis utilisé a une configuration en "x" avec une longueur de chaque bras de 0,21 m. Ces composants ci-dessus sont configurés et assemblés pour former un quadrirotor. Configurés et assemblés pour former un quadrirotor comme le montre la figure IV.11.

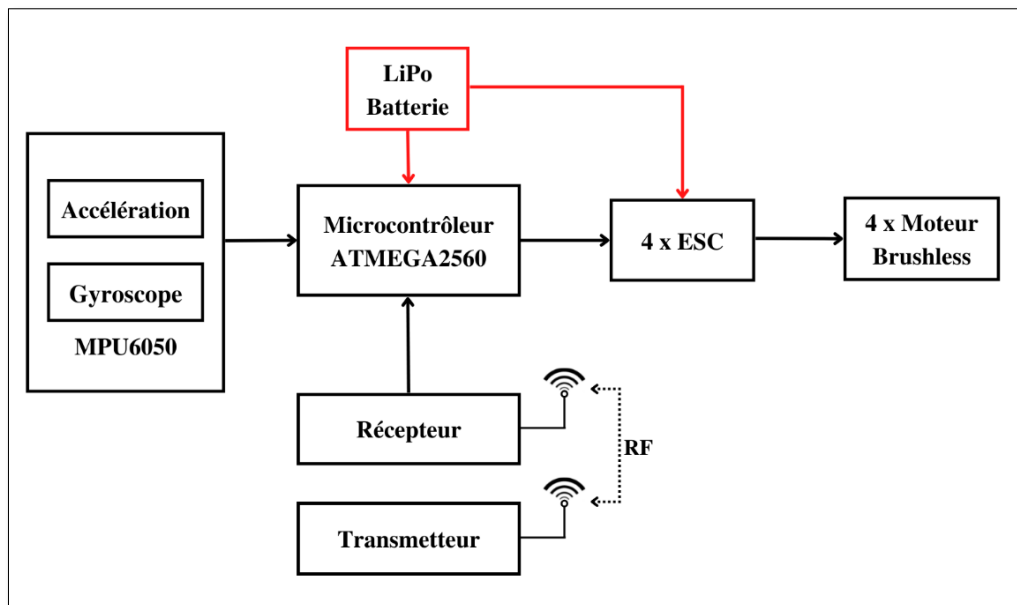


Figure IV.11: Conception du matériel du quadrirotor.

IV.3.2 Test du régulateur

Le test expérimental est réalisé pour voir la performance de la commande PID pendant le vol du quadrirotor. Les constantes PID du **tableau IV.2** sont utilisées pour contrôler le quadrirotor conformément à l'équation du signal de commande PID. Le résultat du test montre que le régulateur PID est capable de stabiliser l'attitude du quadrirotor. Ceci est démontré par les réponses de la troisième attitude qui sont décrites dans **la figure IV.12**. D'après les résultats des tests ci-dessus, les réponses de l'attitude sont bien contrôlées lorsque le régulateur PID est utilisé. Le régulateur est capable de produire des réponses stables avec un temps de stabilisation estimé pour le roulis, le tangage et le lacet de 2,78 s, 5,37 s et 0,94 s respectivement. Les dépassements maximaux obtenus sont de $-25,70^\circ$, $24,01^\circ$ et $-17,09^\circ$ pour le roulis, le tangage et le lacet respectivement. Cependant, le régulateur produit toujours un dépassement dans les conditions transitoires. Les meilleures performances du régulateur peuvent être obtenues en augmentant la robustesse de la conception du régulateur et en optimisant la constante PID. **La figure IV.13** montre le signal de commande produit par le régulateur PID. Ce signal est obtenu à partir de l'équation PID et tracé en fonction des réponses d'attitude sous forme de valeur numérique.

Pour être complet, le quadrirotor utilisé dans cette étude est montré dans **la figure IV.14**.

Paramètre	Pitch Φ	Roll Θ	Yaw Ψ
Kp	1.8	1.8	3.6
Ki	0.04	0.04	0.02
Kd	32	32	0

Tableau IV.2: Les constantes PID.

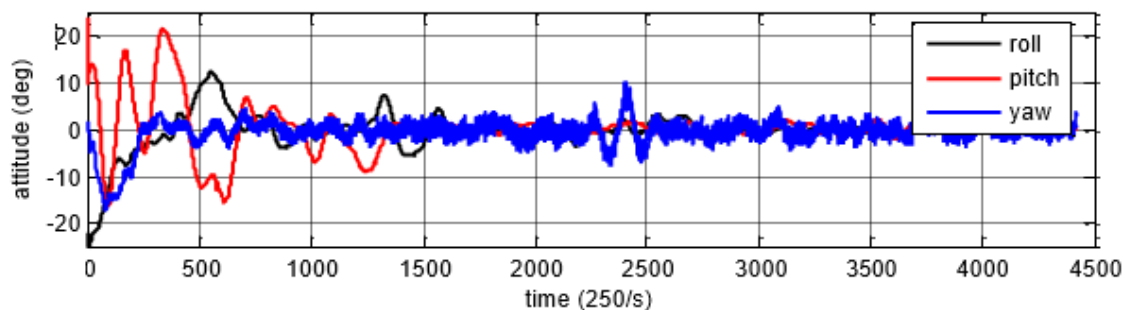


Figure IV.12: Réponses d'attitude du quadrirotor avec le régulateur PID.

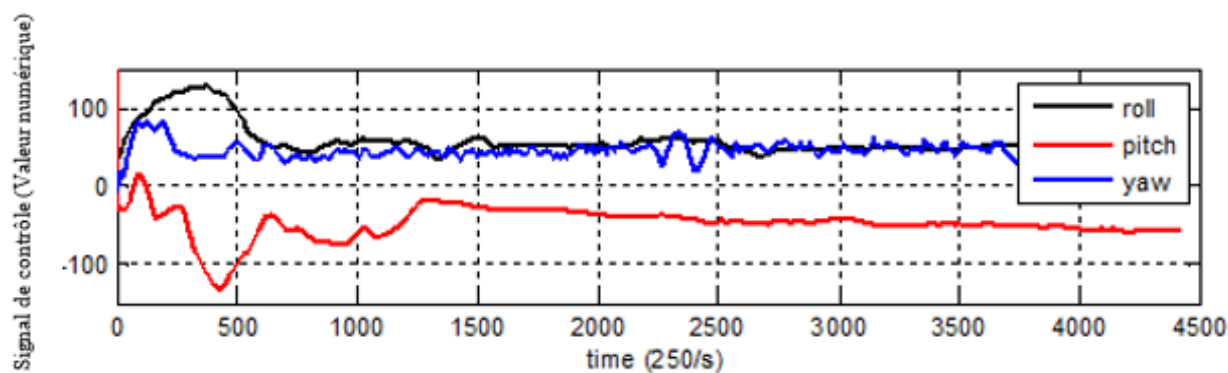


Figure IV.13: Signal de commande.



Figure IV.14: Le quadrirotor utilisé dans notre projet avec radiocommande Flysky.

IV.4 Conclusion

Le software utilisé en transmission et en réception est présenté avec détail et les diverses étapes de programme du régulateur de vol sont expliquées.

Le quadrirotor peut être contrôlé à l'aide d'un système de contrôle simple comme le régulateur PID. De plus, l'attitude du quadrirotor semble stable, même s'il présente un dépassement. Néanmoins, ce résultat est acceptable et conforme aux attentes lorsque nous concevons le régulateur en utilisant des méthodes simples et rapides.

L'amélioration de ce quadrirotor à faible coût nécessite des recherches supplémentaires, ainsi qu'une conception robuste du régulateur, des améliorations matérielles et le développement de son utilisation.

Chapitre V :
La reconstruction 3D

Chapitre V :

La reconstruction 3D

V.1 Introduction

Reconstruire des formes 3D à partir d'une séquence d'images est un problème difficile. L'interprétation des points 3D dépend de la connaissance des paramètres intrinsèques et extrinsèques de la caméra. Les paramètres intrinsèques et extrinsèques définissent la localisation et l'orientation de la caméra par rapport au monde. Les nuages de points sont un type de données spatiales qui peuvent être utilisées pour représenter l'emplacement et la forme des objets. Plus la résolution et le nombre de vues d'un objet dans une photo sont élevés, plus le nuage de points est détaillé et précis.

Dans ce chapitre, nous allons décrire l'outil développé, le tester et vérifier les résultats obtenus pour évaluer la qualité du projet. Nous énumérerons également les diverses limitations que nous avons rencontrées lors de l'élaboration de ce travail.

V.2 Le principe

La photogrammétrie est une technique qui consiste à reconstituer en 3 D une scène en utilisant la parallaxe obtenue entre des images acquises selon des points de vue différents

- Elle exploite de plus en plus les calculs de corrélation entre des images désormais numérique.
- Cette technique repose entièrement sur une modélisation rigoureuse de la géométrie des images et de leur acquisition afin de reconstituer une copie 3 D exacte de la réalité

Les outils numériques permettent aujourd'hui un traitement rapide et aisé des photographies afin d'obtenir des reconstructions 3D.

V.3 Les étapes de la détermination du nuage de points

V.3.1 Étape 01 : L'initialisation de la Camera

L'initialisation de la caméra consiste à charger les images de l'objet en vue d'être reconstruit en 3D et d'approximer à partir de ces dernières la focale de la caméra. Puis, la comparer à une base de données contenant différentes focales de différents modèles de caméra et faire la correspondance pour en déduire approximativement les paramètres de la caméra utilisée (Les différents points de vues, les intrinsèques et extrinsèques de la caméra, la longueur de focale et le champ de vision), il est préférable d'utiliser une focale fixe et des angles fixes afin d'avoir un meilleur rendu.

Cette étape est basée sur le principe de calibration de caméra sous python OpenCV :

Pour la distorsion, les facteurs radiaux et tangentiels sont pris en compte par OpenCV. Pour le facteur radial, on utilise la formule suivante :

$$x_{\text{distorted}} = x(1 + k_1r^2 + k_2r^4 + k_3r^6)$$

$$y_{\text{distorted}} = y(1 + k_1r^2 + k_2r^4 + k_3r^6)$$

La distorsion tangentielle se produit parce que les objectifs de prise d'image ne sont pas parfaitement parallèles au plan d'imagerie. Il peut être représenté via les formules :

$$x_{\text{distorted}} = x + [2p_1xy + p_2(r^2 + 2x^2)]$$

$$y_{\text{distorted}} = y + [2p_1xy + p_2(r^2 + 2x^2)]$$

Où les k sont les coefficients de distorsion radial et p les tangentiels qui dans OpenCV sont présentés sous la forme d'une matrice à une ligne avec 5 colonnes :

$$\text{coefficients de distorsion} = (k_1 \quad k_2 \quad p_1 \quad p_2 \quad k_3)$$

Maintenant, pour la conversion d'unité, la formule suivante est utilisée :

$$\begin{bmatrix} x \\ y \\ w \end{bmatrix} = \begin{bmatrix} f_x & 0 & c_x \\ 0 & f_y & c_y \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x \\ y \\ z \end{bmatrix}$$

Ici, la présence de w est expliquée par l'utilisation du système de coordonnées homographique ($w = Z$). Les paramètres inconnus sont f_x et f_y (focales de la caméra) et (c_x, c_y)

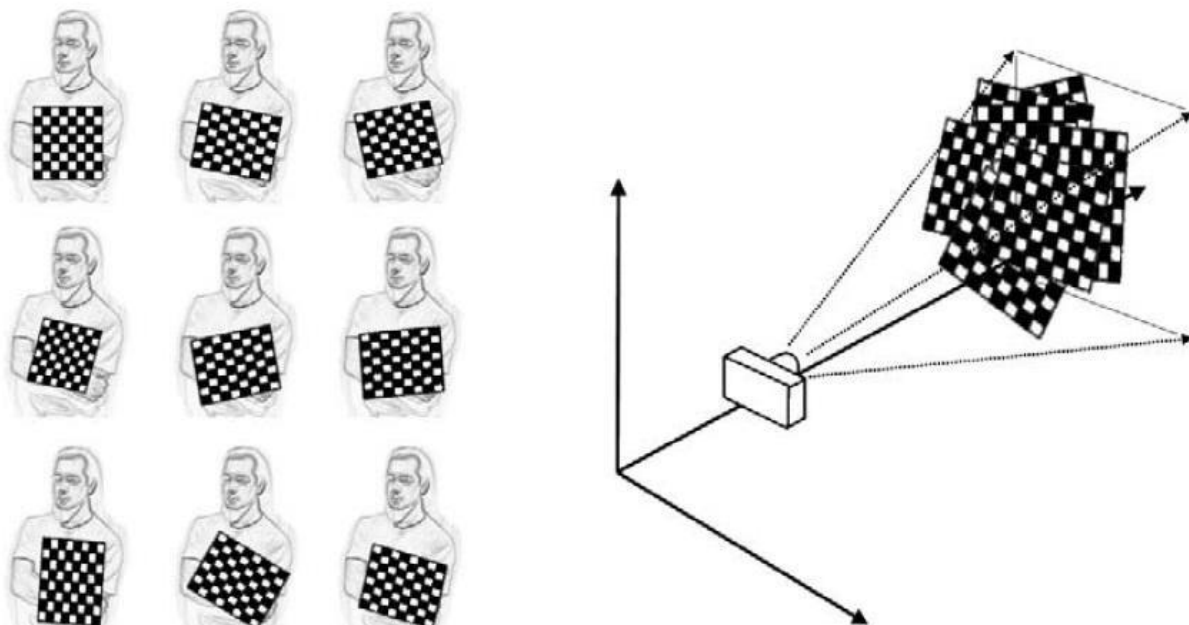
qui sont les centres optiques exprimés en coordonnées pixels. Si pour les deux axes une distance focale commune est utilisée avec un rapport hauteur / largeur donné (généralement 1), alors $f_y = f_x * a$ et dans la formule supérieure, nous aurons une seule distance focale f . La matrice contenant ces quatre paramètres est appelée matrice de caméra. Bien que les coefficients de distorsion soient les mêmes quelles que soient les résolutions de caméra utilisées, ceux-ci doivent être mis à l'échelle avec la résolution actuelle de la résolution calibrée. [28, 29]

Protocole de calibration :

La calibration se fait en quatre étapes grâce aux fonctions fournis par la bibliothèque OpenCV:

- Tout d'abord il faut extraire et suivre les points situés aux intersections des cases blanches et noires du damier.
- Ensuite il faut réordonner les points trouvés, pour qu'ils soient dans un certain ordre
- Puis connaissant les coordonnées de ces points sur l'image, il est possible de déterminer les paramètres internes de la caméra.
- Enfin il est possible de déterminer les paramètres externes de la caméra.

On a utilisé un Damier, parce que ses coins sont très faciles à trouver en utilisant des algorithmes de vision par ordinateur et sa forme est très simple. Afin de connaître la position de



n'importe quel coin nous avons seulement besoin de savoir combien de carrés horizontaux et verticaux il y a dans le damier et la taille d'un carré. Le damier de l'image est un damier 9x6.

Figure V.1 : prise de vue différentes d'un damier [30].

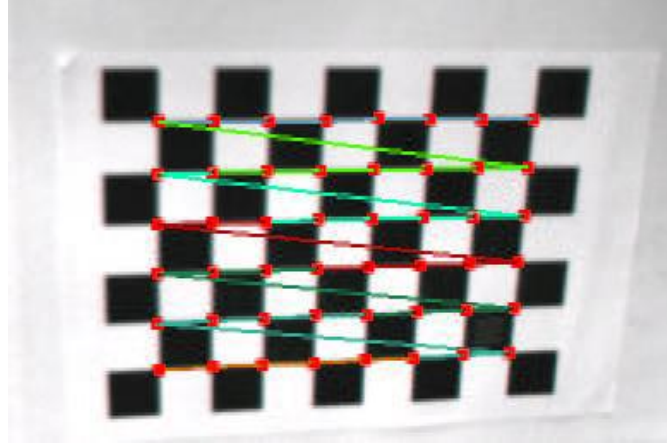


Figure V.2 : Détection des coins des damiers sur la mire de calibration [30].

Afin de déterminer les coefficients de la matrice intrinsèque, il suffit de capturer au minimum 3 images du damier sous différents angles de vues et de stocker les intersections trouvées afin de pouvoir déterminer l'homographie dans le plan projectif de l'image. Ce sont les coefficients de la matrice d'homographie qui sont les coefficients de la matrice intrinsèque. Cette étape est réalisée de façon indépendante pour chaque caméra. Les coefficients de distorsion sont déterminés à posteriori en corrigeant par itération successives les paramètres précédents [30].

V.3.2 Étape 02 : L'extraction des caractéristiques d'image

Les données fournies initialement par le détecteur sont les coordonnées des points d'intérêt ainsi que leur échelle caractéristique. Une étape préliminaire consiste à construire l'histogramme des orientations locales définies en chaque point (x) par :

$$\theta(x, y) = \tan^{-1} \left(\frac{L(x, y + 1) - L(x, y - 1)}{L(x + 1, y) - L(x - 1, y)} \right)$$

$$\forall (x, y) \text{ dans un voisinage de } (x_0, y_0)$$

Cet histogramme se compose de trente-six intervalles (36 classes), couvrant chacun un angle de dix degrés. Ce dernier est pondéré d'une part par un filtre gaussien d'écart type égal a une fois et demie la valeur de l'échelle locale et d'autre part par l'amplitude m de chaque point défini par :

$$m(x, y) = \sqrt{(L(x + 1, y) - L(x - 1, y))^2 + (L(x, y + 1) - L(x, y - 1))^2}$$

Cette méthode a été détaillée dans le chapitre 1 [31, 32, 33].

V.3.3 Étape 03 : Correspondance d'image

L'objectif de cette partie est de trouver des images qui capturent les mêmes zones de la scène. Pour cela, nous utilisons les techniques de récupération d'images pour trouver des images qui partagent du contenu sans le coût de l'extraction de toutes les caractéristiques des correspondances dans les détails. L'idée est de simplifier l'image dans un descripteur d'image compact qui permet de calculer efficacement la distance entre tous les descripteurs d'images.

La génération de ce descripteur d'image est faite à travers l'approche par arbre de vocabulaire. En passant tous les descripteurs de caractéristiques extraits à travers l'arbre vocabulaire, on effectue une classification en comparant leurs descripteurs à ceux de chaque noeud de cet arbre comme suit :

$$q_i = n_i w_i$$

$$d_i = m_i w_i$$

$$w_i = \ln N / N_i$$

Avec q_i les descripteurs et d_i les distances et w_i leurs poids dans l'arbre a vocabulaire

N est le nombre d'image et N_i le nombre d'image avec au moins une correspondance de descripteurs dans le nœud i

Chaque descripteur de fonction se termine dans une feuille, qui peut être stockée par un index simple : l'index de cette feuille dans l'arbre. Le descripteur d'image est alors représenté par cette collection d'indices de feuilles utilisées. Donc, il sera possible de voir si différentes images partagent le même contenu en comparant ces descripteurs d'images [34].

V.3.4 Étape 04 : Correspondance de caractéristiques d'image

L'objectif de cette étape est de faire correspondre toutes les caractéristiques entre les paires d'images. Tout d'abord, une correspondance photométrique entre l'ensemble des descripteurs des 2 images d'entrée est effectuée. Pour chaque caractéristique dans l'image A, une liste de caractéristiques candidates dans l'image B est obtenue. Comme l'espace descripteur n'est ni linéaire ni un espace bien défini, il est impossible de s'appuyer sur des valeurs de distance absolues pour savoir si la correspondance est valide ou non (nous ne pouvons avoir qu'une distance absolue limitée en borne supérieure). Pour supprimer les mauvais candidats, une seule correspondance valide dans l'autre image est supposée. Donc, pour chaque descripteur de fonctionnalité sur la première image, une recherche est effectuée pour les 2 descripteurs les plus proches avec un seuil relatif entre eux.

Cela fournit une liste de caractéristiques correspondantes sur la seule base d'un critère photométrique.

Trouver les 2 descripteurs les plus proches dans la deuxième image pour chaque caractéristique est intensif en calcul avec une approche de force brute, mais de nombreux algorithmes optimisés existent. Le plus courant est le voisin le plus proche approximatif (« Approximate nearest neighbor algorithm »).

Ensuite, les positions des caractéristiques dans les images sont utilisées pour faire un filtrage géométrique en utilisant la géométrie épipolaire dans un cadre de détection des valeurs aberrantes appelé RANSAC (RANDOM SAMple Consensus). Un petit ensemble de correspondances de caractéristiques est sélectionné et ainsi la matrice essentielle est calculée [35, 36].

V.3.5 Structure From Motion

L'objectif de cette étape est de comprendre la relation géométrique derrière toutes les observations fournies par les images d'entrée, et déduire la structure de scène rigide (nuage de points 3D) avec la pose (position et orientation) et l'étalonnage interne de toutes les caméras.

D'abord une reconstruction initiale à deux vues est calculée puis elle est étendue de manière itérative en ajoutant de nouvelles vues. Toutes les correspondances de caractéristiques entre les paires d'images sont fusionnées. Chaque résultat de cette fusion est censé représenter un point dans

l'espace, visible depuis plusieurs caméras. On utilise un système de conversion de coordonnées depuis les coordonnées réelles vers les coordonnées des pixels selon :

$$Z_c \begin{bmatrix} u \\ v \\ 1 \end{bmatrix} = \begin{bmatrix} \frac{1}{dx} & 0 & u_0 \\ 0 & \frac{1}{dy} & v_0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} f & 0 & 0 \\ 0 & 0 & f \\ 0 & 0 & 0 \\ 0 & 1 & 0 \end{bmatrix} \begin{bmatrix} R & t \\ 0 & 1 \end{bmatrix} \begin{bmatrix} X_w \\ Y_w \\ Z_w \\ 1 \end{bmatrix} = \begin{bmatrix} f_x & 0 & u_0 \\ 0 & 0 & f_y \\ v_0 & 0 & 0 \\ 0 & 1 & 0 \end{bmatrix} \begin{bmatrix} R & t \\ 0 & 1 \end{bmatrix} \begin{bmatrix} X_w \\ Y_w \\ Z_w \\ 1 \end{bmatrix}$$

Avec (u,v) le plan des coordonnées pixel et (x, y,z) le plan des coordonnées du monde

Cependant, à cette étape du processus SFM, ce dernier contient encore de nombreuses valeurs aberrantes. Lors de cette fusion des valeurs obtenues à partir des étapes de correspondance, on supprime les pistes incohérentes.

Ensuite, l'algorithme incrémental doit choisir la meilleure paire d'images initiale. Ce choix est essentiel pour la qualité de la reconstruction finale. Il fournit des correspondances robustes et contient des informations géométriques fiables. Ainsi, cette paire d'images choisie par l'algorithme doit maximiser le nombre de correspondances et la répartition des caractéristiques de chaque image. Mais en même temps, l'angle entre les caméras doit également être suffisamment large pour fournir des informations géométriques fiables. Ensuite, la matrice essentielle est calculée entre ces images grâce à l'équation des contraintes épipolaire :

$$E = \hat{T}_2 R_2$$

Ensuite, toutes les images qui ont suffisamment d'associations avec les caractéristiques déjà reconstruites en 3D sont sélectionnées grâce à l'algorithme de sélection des meilleures vues successives (« Next best views selection »). Basé sur ces associations 2D-3D, il effectue la résection de chacune de ces nouvelles caméras. La résection est un algorithme « Perspective-n-Point » (PnP) dans RANSAC pour trouver la pose de la caméra qui valide la plupart des correspondances de caractéristiques effectuées selon la définition suivante :

Étant donné un ensemble de n points 3D dans un référentiel mondial et leurs projections d'images 2D correspondantes ainsi que les paramètres intrinsèques calibrés de la caméra, la pose à 6 degrés de liberté de la caméra sous la forme de sa rotation et de sa translation par rapport au monde est déterminée. Cela suit le modèle de projection en perspective pour les caméras :

$$sp_c = K[R \mid T]p_w$$

Ou $p_w = [x \ y \ z \ 1]^T$ représente un point homogène du monde, $p_c = [u \ v \ 1]^T$ représente un point homogène dans l'image et K est la matrice des paramètres intrinsèques de la caméra ce qui donne la forme suivante :

$$s \begin{bmatrix} u \\ v \\ 1 \end{bmatrix} = \begin{bmatrix} f_x \gamma_u 0 \\ 0 f_y \gamma_v 0 \\ 0 0 1 \end{bmatrix} \begin{bmatrix} r_{11} & r_{12} & r_{13} & t_1 \\ r_{21} & r_{22} & r_{23} & t_2 \\ r_{31} & r_{32} & r_{33} & t_3 \end{bmatrix} \begin{bmatrix} x \\ y \\ z \\ 1 \end{bmatrix}$$

À partir de ces nouvelles poses de caméras, certaines pistes deviennent visibles par 2 caméras réséquées ou plus, puis l'algorithme fait la triangulation. Après, un « Bundle Adjustment » est effectué pour tout affiner : paramètres extrinsèques et intrinsèques de tous les Points de vue de la caméra ainsi que la position de tous les points 3D selon l'équation :

$$(P_i, C_2) = \arg \min_{P_i, C_1, C_2} \sum_{N_i=1}^N ||p_1^i - \pi_1(P_i, C_1)||^2 + ||p_2^i - \pi_2(P_i, C_2)||^2$$

Les résultats du Bundle ajustement sont filtrés en supprimant toutes les observations qui ont une erreur de reprojection élevée ou des angles insuffisants entre les points de vue de la caméra.

Au fur et à mesure que la triangulation de nouveaux points se fait, plus d'images candidates pour la prochaine sélection des meilleures vues sont obtenue. C'est de manière itérative comme cela, en ajoutant des points de vue et trianguler de nouvelles entités 2D en points 3D et en supprimant les points 3D invalides jusqu'à ce que la localisation de nouvelles vues est impossible, qu'un nuage de point général à la fin du processus SFM est généré [35, 37,38].

• Organigramme de la génération du nuage de points

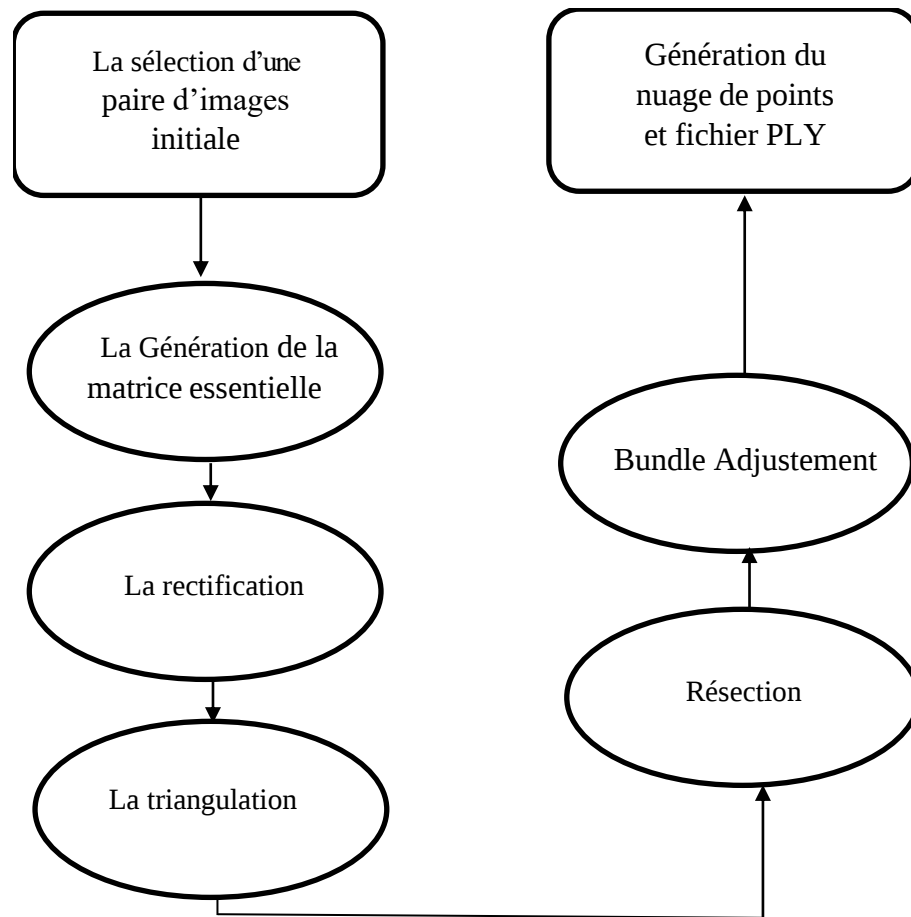


Figure V.3 : Organigramme de l'étape 5.

V.4 Logiciels utilisés

Nous avons utilisé le logiciel MeshLab pour faire la visualisation du nuage de points généré par notre code, d'abord nous avons importé notre nuage de points dans le logiciel Meshroom, puis après quelques tests, le résultat est que ce dernier ne génère pas de modèles 3D d'assez bonne qualité, alors soit on y va au logiciel 3D Zyper ou une fois que nous avons importé notre nuage de points, le modèle 3D est plus détaillé.

V.4.1 MeshLab

MeshLab est un logiciel open source de traitement de maillages 3D. Ce programme est développé par l'ISTI et le CNR, dans le but de fournir un outil généraliste pour la visualisation du

nuage de point, manipulation et l'édition de modèles 3D importants, provenant notamment de scans 3D.

Il propose différents filtres de nettoyage du maillage (suppression des vertex non référencés par exemple), des outils de remaillage (simplification, subdivision, reconstruction de surfaces...) et des fonctions de suppression de bruit (adoucissement, etc.).

MeshLab supporte les formats de stockage suivants : PLY, STL, OFF, 3DS, VRML 2.0, X3D et COLLADA.

MeshLab est utilisé dans de nombreux domaines de recherche et universitaires, tels que la microbiologie, la gestion du patrimoine culturel, la reconstruction de surfaces et l'impression tridimensionnelle.

Ci-dessous le point Cloud généré par notre code Python opencv :

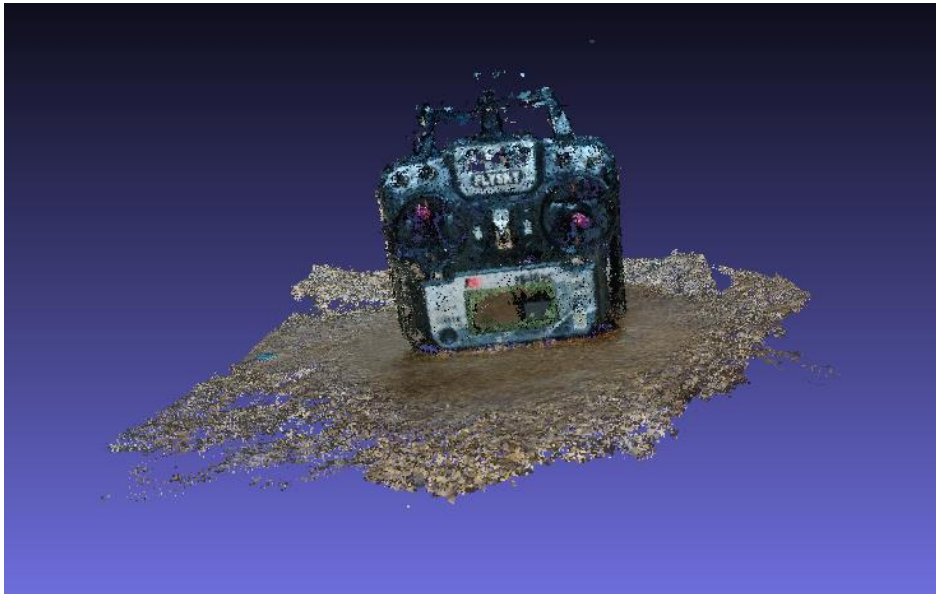


Figure V.4 : Visualisation d'un nuage de point sous MeshLab.

V.4.2 Meshroom

Meshroom est une application qui permet de créer des modèles en 3D à partir d'une série de d'images. Cela est très utile si l'on veut exporter des modèles sur d'autres programmes dans lesquels on pourra continuer à traiter ou à analyser les informations.

L'interface de Meshroom est conçue pour aider à réaliser des modèles sans trop de difficulté. Meshroom propose plusieurs formats qui permettent de voir le modèle qui en résulte. Peu importe celui choisis, on peut facilement voir les compositions extraites sur un PC. Également, il est possible d'exporter ses créations et dans un format qui convient le mieux aux besoins. Il suffit d'importer les images sur lesquelles on a appliquées notre code ainsi que le nuage de points et laisser Meshroom tourner jusqu'à ce qu'il génère le modèle 3D comme illustré dans la figure suivante :

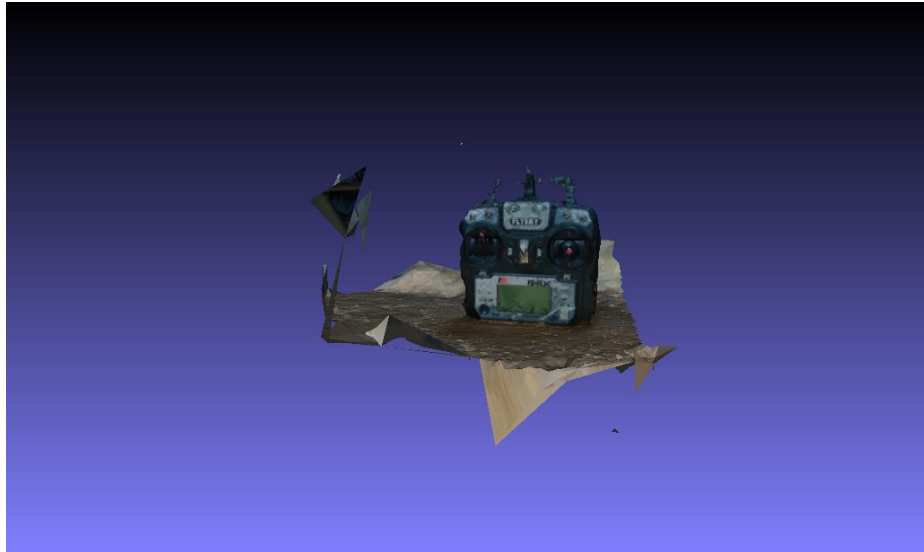


Figure V.5 : Modèle 3D de Flysky sur meshroom.

V.4.3 3DF Zephyr

3Dflow, éditeur du logiciel de scan 3D à partir de photos 3DF Zephyr, a entamé une série de tutoriels autour de son outil. A partir du workflow de base, on découvrira progressivement les fonctions plus avancées : méthodes de prise de vue, masques, nettoyage du scan, reconstruction à partir de vidéos, filtrage, etc.

A noter : 3DF Zephyr est un des rares outils qui soit disponible en version gratuite. 3DF Zephyr Free permet donc de s'initier à moindre coût, mais ne permet pas d'utiliser plus de 50 photos par projet.

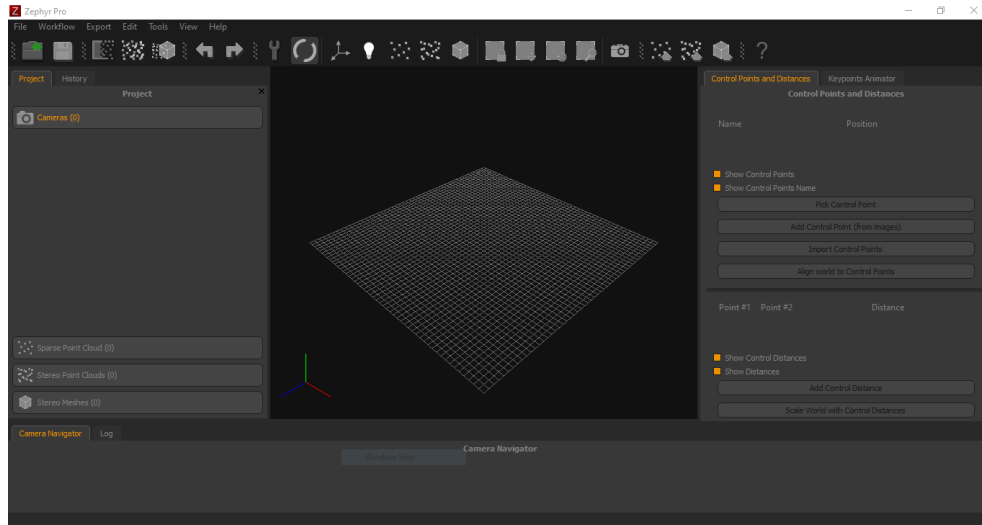


Figure V.6 : logiciel 3DF Zephyr.

Caractéristiques principales de 3DF Zephyr :

- Interface conviviale : le menu du logiciel est facile à utiliser, avec un menu pratique de flux de photos en bas.
- Processus automatique : pas besoin de codage ou d'édition manuelle lors de la création du scan 3D final.
- Outils d'édition : le logiciel permet aux utilisateurs de modifier leur résultat final à l'aide de divers outils.

- **Remarque :**

À partir des résultats illustrés ci-dessus, on peut déduire que le logiciel 3DF Zephyr produit des modèles 3D plus précis et détaillés que ceux produits par le logiciel Meshroom.

V.5 Exemples de reconstruction 3D

Nos tests ont été effectués avec des photos prises avec nos Smartphone. Les figures ci-dessous représentent les résultats obtenus avec ces photos :

V.5.1 Boite de jus (49 photos)

Pour les premiers tests de notre code on a exécuté notre code sur les 49 photos de la boîte de jus dont 2 sont présentées à titre d'exemple, nous obtenons le nuage de points comme le montre la figure ci-dessous



Figure V.7 : Photos de la boîte de jus.



Figure V.8 : Nuage de points de la boîte de jus.

Pour la génération du modèle 3D, on importe le nuage de points vers les logiciels mentionnés précédemment afin d’obtenir les résultats de la figure ci-dessous :



Figure V.9 : Modèle 3D de la boîte de jus.

• **Remarques :**

On voit que le modèle de la boîte de jus est assez fidèle à la réalité malgré une petite distorsion du a plusieurs facteurs dont l’éclairage et les angles de prise de photo.

V.5.2 Flysky (69 photos)

La figure ci-dessous montre un exemple de photos de la base d’images constituées de 69 photos sur lesquelles, on a appliqué notre code :

Le nuage de points obtenu est comme illustré dans la figure ci-dessous :



Figure V.10 : Nuage de points de Flysky.

Le nuage de points est ensuite, exporté vers les logiciels mentionnés précédemment pour faire la génération du modèle 3D. Le résultat est celui de la figure ci-dessous :

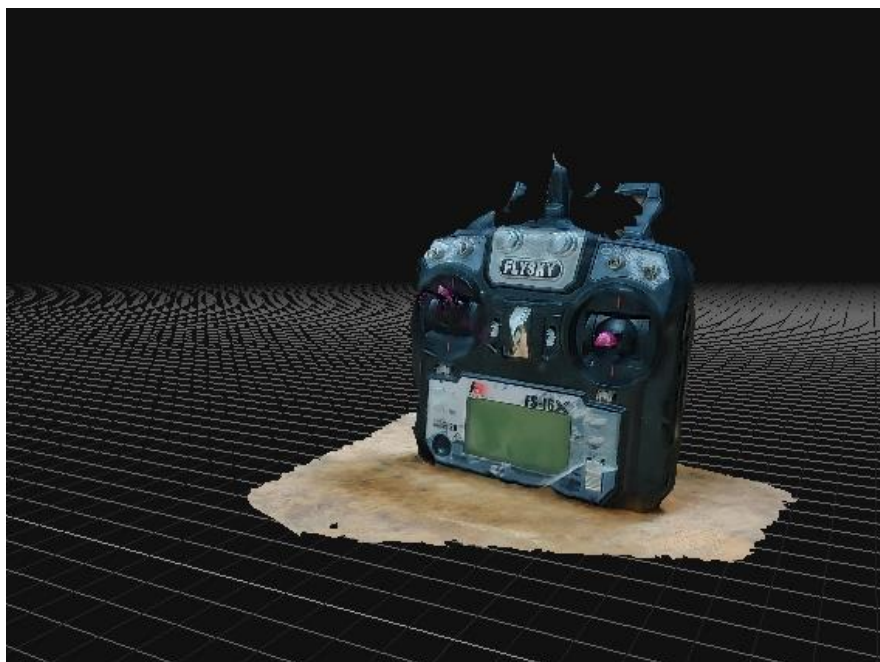


Figure V.11 : Modèle 3D de Flysky sur 3D Zephyr.

• **Remarques :**

On voit que le modèle Flysky est très proche d'être parfait sur 3D Zephyr, mais il y a encore une certaine distorsion due à un certain nombre de facteurs, notamment l'éclairage et les angles de prise de vue.

V.5.3 Pumpjack (221 photos)

La figure ci-dessous montre un exemple de photos de la base d'images constituées de 69 photos sur lesquelles, on a appliqué notre code :



Figure V.12 : Photo 1



Figure V.13 : Photo 2

Le nuage de points obtenu est comme illustré dans la figure ci-dessous :

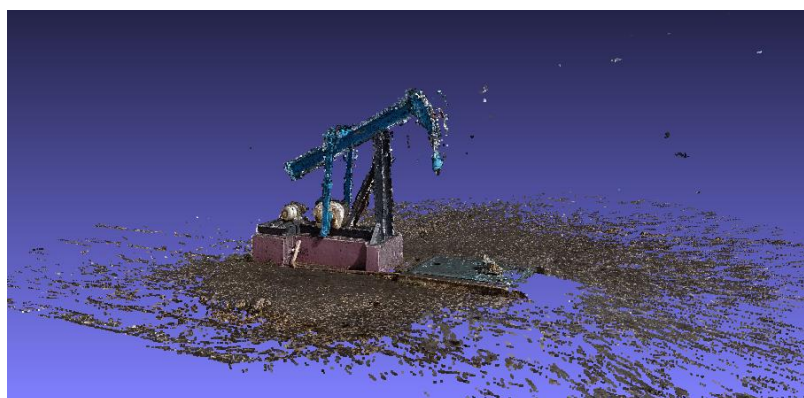


Figure V.14 : Nuage de points de la pompe.

Pour la génération du modèle 3D, on exporte le nuage de points vers les logiciels mentionnés précédemment et on obtient le résultat de la figure ci-dessous :



Figure V.15 : Modèle 3D de pompe 1



Figure V.16 : Modèle 3D de pompe 2.

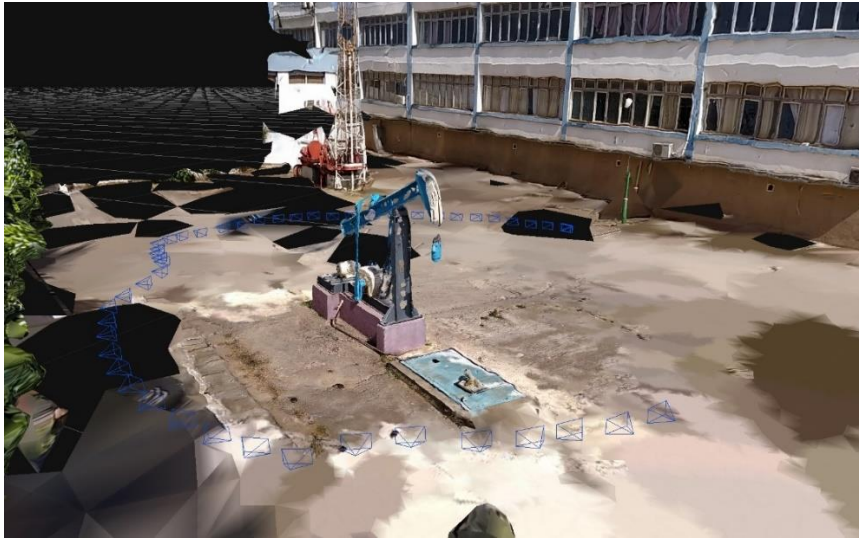


Figure V.17 : Modèle 3D de pompe.

- **Remarques :**

On voit que le modèle de la structure est presque parfait, on remarque seulement des zones vides qui sont le résultat de certaines zones d'ombre dans les photos et du nombre insuffisant de photos de la partie basse.

V.6 Critique des résultats obtenus

Nous pouvons voir que la qualité des caméras de nos smartphones est un facteur important dans notre prise de décision. Il peut y avoir une certaine distorsion dans le modèle 3D en raison du

petit nombre de photos utilisées. Parce que plus il y a de photos de l'objet à reconstruire disponibles en 3D, plus les modèles sont réalistes.

Il y a aussi le fait que, dans le premier test, nous n'avons pas encore clairement identifié le principe de la photographie, car les photos doivent être prises à des angles précis avec un recouvrement de 60% entre chaque paire d'images traitées. L'éclairage change également, ainsi que les couleurs de l'objet, un objet trop éclairé peut provoquer une erreur d'algorithme et vice versa, il faut aussi tenir compte de la texture de l'objet, un objet constitué de surfaces réfléchissantes ou une partie trop sombre ou noire ne sera pas prise en compte par l'algorithme. Enfin, si l'objet est creux ou constitué de plusieurs cavités, il est possible qu'il ne soit pas reconstitué en détail car il y a trop peu de détails dans les cavités.

V.7 Conclusion

A travers ce chapitre, nous présentons nos résultats de simulation. Cela comprend une description des méthodes et des données utilisées, ainsi que les résultats des simulations.

La photogrammétrie est un processus d'acquisition de données 3D précises à partir de photographies. Quand on a clairement identifié le principe de la photographie, notre code a généralement bien fonctionné, générant des nuages de points réalistes et des modèles 3D proches de l'objet sur la photo.

Sur la base des données de ce chapitre, nous pouvons dire que l'algorithme utilisé pour la photogrammétrie (SIFT et SFM) est efficace.

Conclusion Générale

Conclusion Générale

Avec la croissance de l'intérêt pour les UAV, celui des avions à atterrissage et à décollage verticaux et le besoin d'une instrumentation discrète et surtout légère, le quadrirotor a connu une grande popularité ces dernières années. Cela en fait, la principale motivation pour ce travail de recherche. Le travail présenté dans ce mémoire était consacré à la réalisation d'un drone de type quadrirotor et la réalisation de la reconstruction 3D.

Notre travail a débuté par des généralités sur les drones, avant d'aborder leurs diverses utilisations, ainsi que nous avons présenté la photogrammétrie et sa fonctionnalité principale. Par la suite, on a présenté les composants nécessaires à la construction de notre drone quadrirotor et leur mouvement. Notre étude s'est poursuivie par des tests sur le fonctionnement des différentes parties de notre système et la réalisation d'un prototype drone quadrirotor. La stabilisation du drone a été assurée en implémentant une loi de commande de type PID. Par la suite, une description des méthodes et des données utilisées pour la reconstruction 3D été présenté, après l'identification du principe de la photographie, l'algorithme de la photogrammétrie a généralement bien fonctionné, générant des nuages de points réalistes et des modèles 3D proches de l'objet sur la photo.

À l'issue de ce projet, nous avons approfondi nos connaissances et appliqué ce que nous avons appris lors de formations dans le domaine de l'automatisation.

Au terme de ce projet, nous avons couvrez 3 cotés :

- Le coté montage : de ce travail, nous a permis à apprendre beaucoup sur les composants électroniques, leur montage et leur programmation, surtout l'importance des microcontrôleurs pour la commande des systèmes. Nous avons utilisé une carte de contrôle Arduino pour contrôler 4 moteurs brushless de 1000 KV chacun utilisant des variateurs de vitesse (ESC). Cette configuration a été installée sur le châssis du quadricoptère.

- Le coté programmation : nous avons utilisé plusieurs langages comme l'Arduino pour la conception de système de notre drone, Matlab pour tracer les courbes et Python pour générer et détecté les points d'intérêt ainsi le nuage des points pour modéliser les objets 3D.

- Le côté modélisation 3D : On a identifié après des multiples tests le principe de la photographie dans le but d'éviter les zones d'ombres qui produisent les distorsions. Par la suite, on a généré des nuages de points pour réaliser des modèles 3D et obtenu des résultats assez fidèles à la reality.

Difficultés rencontrées :

- Au départ, étant donné la nature du projet, il était difficile de déterminer l'ampleur de la tâche. La réalisation d'un drone quadrirotor offre de nombreuses possibilités en termes d'applications.
- La plus grande difficulté rencontrée lors de ce projet fut la stabilisation du drone.
- Nous avons rencontré la contrainte de la batterie de drone, la durée de consommation c'était l'environnement de 15min, donc la durée des tests était vraiment courte.
- La difficulté rencontrée dans la partie de la photogrammétrie c'était la compréhension du principe de fonctionnement de la reconstruction 3D, la détection de nuage des points et maîtrise du principe de la photographie.
- Enfin, en général, nous avons mal estimé la durée comme étant attribuable certaines tâches au final ne nous ont pas permis de finir tout ce que nous voulions faire.

Notre objectif ne s'est que réalisé 80 %, on est vis-à-vis l'effectuer après la date de soutenance pour partir plus loin que « réaliser de la reconstruction 3D à partir d'images acquises par notre quadrirotor ».

Perspectives :

- Nous pourrions intégrer un algorithme de guidage à l'aide du GPS et des différents capteurs (ultrasons, baromètre...). Couplée aux données capteurs, l'implémentation de traitement d'images embarquées offrirait toute une palette d'activités que l'on pourrait imaginer (ex : détection de formes spécifiques, tracking d'objets, etc.).
- Remplacer la carte Arduino par Raspberry pi.
- L'intégration d'un algorithme décisionnel lié à de l'intelligence artificielle rendrait complètement autonome pour les missions de la photogrammétrie.
- A l'aide d'une application Android, pour voir ce que la caméra capte par le smartphone, ainsi suivre la trajectoire d'un objet ou d'une personne de façon autonome sans intervention humaine.
- Sauvegarder les données de trajectoire et les images de l'environnement extérieur sur une grande base de données pour l'auto-apprentissage.

Références bibliographiques

- [1]. Samuel Dubois, Yves Vanhellemont, Michael de Bouw (CSTC) : « Les drones au service de la construction, Technologies, enjeux et perspectives ». CENTRE SCIENTIFIQUE ET TECHNIQUE DE LA CONSTRUCTION, Août 2019
- [2]. AMAURY DE SAINT MARTIN : « LES DRONES, LA NOUVELLE RÉVOLUTION TECHNOLOGIQUE ». Centre Jean Gol, 2019.
- [3]. Chekakta Zakaria, Zoubiri Abdellah : « Conception, Modélisation et Commande d'un UAV de type Quadrirotor ». École Nationale Polytechnique d'Oran, Mémoire Pour l'Obtention du Diplôme d'Ingénieur d'Etat, 2016.
- [4]. M. Pierre GRUSSENMEYER : « Du nuage de points à la maquette numérique de bâtiment : Reconstruction 3D semi-automatique de bâtiments existants La photogrammétrie par drone : les domaines et applications ». L'Université de Strasbourg, Thèse, 2017.
- [5]. La photogrammétrie par drone : les domaines et applications :
<https://www.studioflytechnologie.fr/photogrammetrie-drone-volumetrie-domaines-applications/>
- [6]. Egels Yves : « La photogrammétrie, principes généraux et terminologie ». In : Collection EDYTEM. Cahiers de géographie, numéro 12, 2011.
- [7]. Photogrammétrie :
<https://www.apgtp.fr/la-branche/les-metiers/photogrammetre/>
- [8]. M. LAURENT : « Acquisition 3D au service de l'industrie ». Magazine électronique CiMax, 1997.
- [9]. A. Zingoni a, M. Diani a, G. Corsini a, A. Masini : «REAL-TIME 3D RECONSTRUCTION FROM IMAGES TAKEN FROM AN UAV». Dept. of Information Engineering, University of Pisa, 2015.
- [10]. K. Kraus, P. Waldhäusl : « Manuel de photogrammétrie, principes et procédés fondamentaux ». Traduction de Pierre Grussenmeyer et O. Reis, Hermes, 1998
- [11]. Réalisation d'un drone à base d'Arduino - Chapitre 2 :
<https://www.firediy.fr/article/realisation-d-un-drone-a-base-d-arduino-chapitre>
- [12]. Studiosport : « Châssis Quadcoptère F450 » :
<https://www.firediy.fr/article/realisation-d-un-drone-a-base-d-arduino-chapitre-2>
- [13]. Components101 : « 2212 Brushless Motor », 2018 :

- <https://components101.com/motors/2212-brushless-motor>
- [14]. A2212 Brushless Motor Datasheet: « A2212/13T TECHNICALDATA ». Components101, 2018.
- [15]. Lobodol : « Réalisation d'un drone à base d'Arduino - Chapitre 2 ». Fire-DIY,2017.
- [16]. David McGriffy: «Make drones: Teach an Arduino to Fly». Maker Media Year, 2016.
- [17]. MPU-6000-Datasheet: «MPU-6000 and MPU-6050 Product Specification Revision 3.4».
- [18]. InvenSense Inc. Numéro de document : PS-MPU-6000A-00,2013.
- [19]. MPU6050 Datasheet: «MPU6050 Accelerometer and Gyroscope Module». Components101,2021.
- [20]. DroneBot Workshop: Build a Digital Level with MPU-6050 and Arduino. DroneBot Workshop, 2019.
- [21]. Arduino Mega 2560 Datasheet: «Arduino Mega 2560 Datasheet».Manualzz, Numéro de document : 14316422.
- [22]. FlySky FS-i6 et FlySky FS- iA10B
<https://www.flysky-cn.com/>
- [23]. T. Hamel, R. Mahony, R. Lozano, J. Ostrowski “Dynamic modelling and configuration stabilization for an x4-flyer.”, 15th Triennial World Congress, Barcelona, Spain, 2002.
- [24]. L’IDE Arduino :
<https://arduino.blaisepascal.fr/presentation/logiciel/>
- [25]. Henri Bourlés : « Systèmes linéaires De la modélisation á la commande ». Hermes Science, 2006, 510p.
- [26]. Henri Bourlés, Hervé Guillard : « Commande des systèmes. Performance et robustesse ». Ellipses,2012.
- [27]. Paul Avery: « Introduction to PID control ». Yaskawa Electric America, 2019.
- [28]. Camera sous OpenCV :
https://docs.opencv.org/master/dc/dbb/tutorial_py_calibration.html
- [29]. Les Distorsions en calibration de caméra
<https://ori.codes/artificial-intelligence/camera-calibration/camera-distortions/>
- [30]. NAAS NASREDDINE, RIAD SIDAHMED : « Détermination de la position d’un objet à l’aide d’un couple de caméras ». Université Abdelhamid Ibn Badis Mostaganem, Mémoire de master, 2012.

- [31]. D.Lowe : « Distinctive Image Features from Scale-Invariant Keypoints ». Université de la Colombie-Britannique, 2004.
- [32]. Deepanshu Tyagi: « Introduction to SIFT ». Data Breach, Article, 2019.
- [33]. Manuel Grand-Brochier : « Descripteurs 2D et 2D+t de points d'intérêt pour des appariements robustes ». Université Blaise Pascal - Clermont-Ferrand II, Thèse de doctorat, 2011.
- [34]. D.Nistér, H.Stewénius: « Scalable Recognition with a Vocabulary Tree ». CVPR, 2006.
- [35]. Yan Shen and Yuxing Dai, "Structure from motion with efficient homography-based line matching," J. Opt. Soc. Am. A 35, 200-209, 2018.
- [36]. S. Arya et.al: « An Optimal Algorithm for Approximate Nearest Neighbor Searching in Fixed Dimensions ». In the Proceedings of the Fifth Annual ACM-SIAM Symposium on Discrete Algorithms, 1994, pp. 573-582.
- [37]. Pierre Moulon & Alessandro Bezzi: « Python Photogrammetry Toolbox: A free solution for Three-Dimensional Documentation ». Article, 2011.
- [38]. Pierre Moulon, Pascal Monasse, Renaud Marlet : « Global Fusion of Relative Motions for Robust, Accurate and Scalable Structure from Motion ». Université Paris-Est, 2013.