

**Projet Recherche & Ingénierie**

Année 2022-2023 / Semestre 9

PRI-2022S9-SA307

**Etude et réalisation d'un drone quadricoptère**

Joyce SUDI

N° étudiant : 4022969

**7 février 2023**

**Site :** Villeurbanne

**Structure d'Accueil :** SA ISP - Ingénierie des Systèmes Pilotés

**Tuteur(s) scientifique(s) :** Eric Bideaux, Paolo Massioni

**Organisme commanditaire :** Laboratoire Ampère - CNRS UMR 5005

## Résumé

On souhaite à travers ce projet développer un nouveau support de travaux pratiques pour le cours de Dynamique et contrôle des systèmes aérospatiaux nouvellement introduit dans la maquette de formation du département de Génie Mécanique de l'INSA Lyon afin d'illustrer la dynamique du vol des véhicules autonomes. Il s'agit de modéliser et de simuler la dynamique de vol du quadricoptère, de réaliser et tester le prototype réel et de mettre en place les premières lois de commande.

Le véhicule est construit à partir des composants sélectionnés en fonction de leur compatibilité. Un banc d'essai à 4 degrés de liberté a également été réalisé afin de tester et de valider le fonctionnement du quadricoptère. Le modèle Simulink du quadricoptère est dérivé à partir des équations différentielles de la dynamique du vol du drone. Un contrôleur d'attitude basé sur le PID a été conçu et réglé pour le modèle Simulink du quadcoptère et pour l'implémentation sur le prototype du véhicule.

**Mots-clés :** quadricoptère, dynamique de vol, Matlab/Simulink, PID, Arduino, Modélisation multiphysique

## Abstract

Through this project, we aim to develop a new practical work support for the course of Dynamics and control of aerospace systems newly introduced in the curriculum of the Mechanical Engineering Department of INSA Lyon in order to illustrate the flight dynamics of autonomous vehicles. The project consists in modeling and simulating the flight dynamics of the quadricopter, in realizing and testing the real prototype and setting up the first control laws.

The vehicle is built from components selected according to their compatibility. A test bench with 4 degrees of freedom was also realized in order to test and validate the operation of the quadricopter. The Simulink model of the quadcopter was derived from the differential equations of the flight dynamics of the UAV. A PID-based attitude controller was designed and tuned for the Simulink model of the quadcopter and for implementation on the prototype vehicle.

**Keywords:** quadcopter, flight dynamics, Matlab/Simulink, PID, Arduino, Multiphysics modeling

# Table des matières

<b>1</b>	<b>Introduction</b>	<b>9</b>
1.1	Contexte et objectifs	9
1.2	Applications	9
1.3	État de l'art	10
<b>2</b>	<b>Dynamique du véhicule</b>	<b>11</b>
2.1	Orientation et repères de référence	11
2.2	Les angles d'Euler	13
2.3	Les équations du mouvement	14
<b>3</b>	<b>Réalisation du prototype</b>	<b>22</b>
3.1	Composants	22
3.1.1	Le cadre	23
3.1.2	Les hélices	23
3.1.3	Les moteurs à courant continu sans balais - Brushless DC Motors	24
3.1.4	Contrôleur électronique de vitesse - Electronic Speed Controllers (ESCs)	24
3.1.5	Batterie	26
3.1.6	Microcontrôleur - Arduino MKR WiFi 1010	26
3.1.7	Carte d'extension pour la commande des moteurs - Arduino MKR Motor Carrier	27
3.1.8	Centrale inertielle - Arduino MKR IMU Shield (BNO055 IMU)	28
3.2	Banc d'essai à 4 degrés de liberté pour quadricoptère	29
<b>4</b>	<b>Modélisation et contrôle avec Matlab Simulink</b>	<b>30</b>
4.1	Modélisation Simulink du quadricoptère	30
4.1.1	Le modèle physique	31
4.1.2	Vérification du modèle du quadricoptère	35
4.2	Contrôleur PID	38
4.2.1	Le contrôleur d'attitude	38
4.2.2	Réglage du contrôleur	39
4.2.3	Suivi de trajectoire	42
4.3	Implémentation	45
4.3.1	Modèle Simulink pour l'implémentation sur le drone réel	45
4.3.2	Réglage du contrôleur	48
<b>5</b>	<b>Conclusion et perspectives</b>	<b>49</b>
5.1	Conclusion	49
5.2	Perspectives	50
<b>A</b>	<b>Identification des paramètres du modèle du quadricoptère</b>	<b>52</b>
<b>B</b>	<b>Paramètres du modèle du quadricoptère - code MATLAB</b>	<b>54</b>
<b>C</b>	<b>Commande des moteurs BLDC avec des ESC sur MATLAB</b>	<b>55</b>

<b>D</b>	<b>Commande des moteurs BLDC avec des ESC sur IDE Arduino</b>	<b>56</b>
<b>E</b>	<b>Dessin d'ensemble - banc d'essai à 4 degrés de liberté pour quadricoptère</b>	<b>58</b>
<b>F</b>	<b>Dessin de définition - pied</b>	<b>59</b>
<b>G</b>	<b>Dessin de définition - coulisseau</b>	<b>60</b>
<b>H</b>	<b>Extrait du datasheet - embouts axiaux à rotule réglable M10</b>	<b>61</b>
<b>I</b>	<b>Dessin de définition - fixation drone</b>	<b>62</b>
<b>J</b>	<b>Dessin de définition - fixation coulisseau</b>	<b>63</b>
<b>K</b>	<b>Dessin d'ensemble - protection d'hélice</b>	<b>64</b>
<b>L</b>	<b>Dessin de définition - protection d'hélice part 1</b>	<b>65</b>
<b>M</b>	<b>Dessin de définition - protection d'hélice part 2</b>	<b>66</b>
<b>N</b>	<b>Dessin de définition - entretoise hexagonale M3x50mm</b>	<b>67</b>

# Table des figures

2.1	Les six degrés de liberté du véhicule. . . . .	11
2.2	Repère inertiel et repère lié au solide (Tytler, 2017) . . . . .	12
2.3	Différentes configurations de l'orientation du quadricoptère : configuration en "+" à gauche et configuration en "x" à droite. (Ferry, 2017) . . . . .	13
2.4	Les angles d'Euler (Usman, 2020) . . . . .	13
3.1	Composants du quadcopter . . . . .	22
3.2	Cadre du quadricoptère (Lynxmotion Multirotor Erector Set) . . . . .	23
3.3	Les hélices (Lynxmotion 9x4.5R Propeller) . . . . .	23
3.4	Les moteurs à courant continu sans balais (Lynxmotion Brushless Motor 28x30 1000kv) . . . . .	24
3.5	Contrôleur électronique de vitesse (Lynxmotion 12A ESC SimonK) . . . . .	25
3.6	Branchements des moteurs sans balais aux ESCs ( <a href="https://www.dronetrest.com/">https://www.dronetrest.com/</a> ) . . . . .	25
3.7	Batterie 3S LiPo 11.1 V 3800 mAh Conrad energy . . . . .	26
3.8	Microcontrôleur - Arduino MKR WiFi 1010 . . . . .	27
3.9	Carte d'extension pour la commande des moteurs - Arduino MKR Motor Carrier . . . . .	27
3.10	Centrale inertielle - Arduino MKR IMU Shield (BNO055 IMU) . . . . .	28
3.11	Schéma cinématique du banc d'essai à 4 degrés de liberté . . . . .	29
3.12	Banc d'essai à 4 degrés de liberté pour quadricoptère . . . . .	29
4.1	Modèle de simulation du quadricoptère . . . . .	30
4.2	Le modèle physique du quadricoptère avec ses entrées et ses sorties . . . . .	31
4.3	Les différents sous-systèmes du modèle physique . . . . .	32
4.4	Le sous-système - Orientation du véhicule . . . . .	32
4.5	Les équations donnant la poussée et les moments en fonction des vitesses de rotation de rotors . . . . .	33
4.6	Le sous-système des accélérations angulaires . . . . .	33
4.7	Les équations des accélérations angulaires du quadricoptère . . . . .	34
4.8	Le sous-système des accélérations linéaires . . . . .	34
4.9	Les équations des accélérations linéaires du quadricoptère . . . . .	35
4.10	Vérification du modèle du quadricoptère comportements souhaités (Ferry, 2017) . . . . .	36
4.11	Vérification de la prise d'altitude . . . . .	36
4.12	Vérification du mouvement de roulis . . . . .	37
4.13	Vérification du mouvement de lacet . . . . .	37
4.14	Schéma-bloc du contrôleur PID . . . . .	38
4.15	Le système de contrôle en boucle fermée . . . . .	38
4.16	Le contrôleur d'Attitude pour la simulation . . . . .	39
4.17	Réponse du système à une rampe - réglage du PID pour le contrôle d'altitude . . . . .	40
4.18	Réponse du système à un angle échelon de 12 degrés en roulis - réglage du PID pour le contrôle de l'angle de roulis . . . . .	41
4.19	Réponse du système à un angle échelon de 12 degrés en lacet - réglage du PID pour le contrôle de l'angle de lacet . . . . .	42
4.20	Signal de consigne et réponse du système pour l'angle de roulis . . . . .	42
4.21	Signal de consigne et réponse du système pour l'angle de tangage . . . . .	43

4.22	Signal de consigne et réponse du système pour l'angle de lacet . . . . .	43
4.23	La réponse du système aux trois signaux de consigne d'angle d'orientation . . . . .	44
4.24	Trajectoire du véhicule en coordonnées inertielles . . . . .	44
4.25	Modèle Simulink pour l'implémentation sur le drone . . . . .	45
4.26	Le sous-système IMU SENSOR DAQ pour l'acquisition de données du capteur inertiel . . . . .	46
4.27	Le contrôleur d'Attitude pour l'implémentation sur le drone . . . . .	46
4.28	Le sous-système Quadcopter Motors Controller pour le pilotage des moteurs BLDC . . . . .	47
A.1	Modélisation du quadricoptère sur le logiciel Onshape . . . . .	52

# Liste des tableaux

2.1	Les différentes variables du modèle . . . . .	15
3.1	Liste des composants du drone . . . . .	22
3.2	Spécifications du moteur à courant continu sans balais Lynxmotion Brushless Motor 28x30 1000kv . . . . .	24
3.3	Spécifications du contrôleur électronique de vitesse (Lynxmotion 12A ESC SimonK) . . . . .	25
3.4	Branchement des ESCs à la carte d'extension pour la commande des moteurs - Arduino MKR Motor Carrier . . . . .	28
4.1	Paramètres du modèle du quadricoptère . . . . .	31
4.2	Les valeurs des gains du PID pour le contrôle de l'altitude . . . . .	40
4.3	Les valeurs des gains du PID pour le contrôle de l'angle de roulis et de tangage . . . . .	40
4.4	Les valeurs des gains du PID pour le contrôle de l'angle de lacet . . . . .	41
A.1	Masse et volume de chaque pièce de l'assemblage du drone . . . . .	52

# Liste des abréviations

CG	Centre de gravité
CW	Clockwise
CCW	Counterclockwise
NED	North-East-Down
ENU	East-North-Up
IMU	Inertial Measurement Unit
DC	Direct Current
LiPo	Lithium Polymer
BLDC	Brushless DC
PID	Proportional, Integral, Derivative
MPC	Model Predictive Control
RPM	Rotations Per Minute
PWM	Pulse Width Modulation
ESC	Electronic Speed Controller
LQR	Linear Quadratic Regulator
ZOH	Zero Order Hold
VTOL	Vertical Take-off and Landing Aircraft
UAV	Unmanned Aerial Vehicle

# Chapitre 1

## Introduction

Le quadricoptère constitue encore aujourd'hui une technologie émergente dont le champ d'application continue de s'élargir au fil des innovations. Un quadrirotor ou quadricoptère est un véhicule aérien comportant quatre rotors lui permettant de voler. Les quatre rotors sont souvent disposés à égale distance du centre de masse du véhicule et forment une structure en croix. Les quadricoptères font partie de la catégorie des véhicules à décollage et atterrissage verticaux (VTOL - Vertical Take-off and Landing Aircraft). La rotation de chaque rotor permet de créer une poussée dans la direction perpendiculaire au plan de rotation de l'hélice, la combinaison méthodique des différentes poussées créées par les rotors permet d'effectuer différentes manœuvres (vol stationnaire, décollage, atterrissage, voire des mouvements plus acrobatiques).

Avec l'avènement du GPS et de l'internet, les drones ont connu une grande croissance de leur adoption dans le secteur civil. En 2019, le marché de drones civils était évalué à 4,9 milliards de dollars US avec un taux de croissance annuel composé de plus de 15% au cours de la période de prévision 2022 - 2027 (source : Teal Group).

### 1.1 Contexte et objectifs

Dans le cadre de la formation Mécatronique et Systèmes du département de Génie Mécanique de l'INSA Lyon, un nouveau cours « Dynamique et contrôle des systèmes aérospatiaux » a été introduit. On souhaite à travers ce projet développer un nouveau support de travaux pratiques pour ce cours afin d'illustrer la dynamique du vol des véhicules autonomes.

Les objectifs du projet sont la modélisation et la simulation de la dynamique de vol du quadricoptère, la réalisation et le test du prototype réel ainsi que le développement des premières lois de commande.

### 1.2 Applications

Les progrès technique et l'innovation dans le développement des drones ont permis d'élargir leur champ d'applications. Aujourd'hui, les drones ont des applications dans plusieurs domaines avec des objectifs différents. Ci-dessous, quelques applications des drones sont énumérées.

- Le transport (Livraison de marchandises)
- L'inspection des sites dangereux (mines, centrales nucléaires, chantiers dangereux)
- Les missions de reconnaissance (acquisition des renseignements militaires sur le territoire ennemi en situation de guerre)
- La recherche et sauvetage (lors des catastrophes naturelles, incendies et sinistres)
- La cartographie 3D des bâtiments ou des sites (imagerie thermique)

- La surveillance aérienne (patrouille routière, maintien de l'ordre)
- La prise de vue aérienne (photographie, cinéma, concerts et événements sportifs)
- L'agriculture, etc...

### 1.3 État de l'art

Le drone quadricoptère présente un intérêt grandissant pour les équipes de recherche en ingénierie de contrôle notamment en tant que plateforme d'essai pour différentes stratégies de contrôle. Plusieurs recherches sont menées afin d'améliorer la manoeuvrabilité et les performances de vol des drones tout en utilisant diverses stratégies de contrôle telles que les contrôleurs à mode glissant, la logique floue linéaire, le régulateur quadratique (LQR), le contrôle adaptatif, le contrôle prédictif (MPC), la commande par linéarisation entrée-sortie, etc. ([Fiaz et al., 2018](#)).

L'utilisation des drones dans des applications complexes visant à effectuer des manoeuvres agressives ou de suivi de trajectoire complexe (inspection ou exploration rapide en intérieure) exige une manipulation précise des mouvements du drone. De telles applications nécessitent une bonne compréhension de la dynamique du système et une élaboration d'un contrôleur adapté.

La modélisation du système dynamique est une étape importante pour bien comprendre le fonctionnement du quadricoptère. Les équations différentielles régissant la dynamique du quadcoptère peuvent être dérivées à l'aide des équations de Newton-Euler et des équations d'Euler-Lagrange, les deux approches sont présentées dans cet article [Luukkonen \(2011\)](#). Cet article inclut la modélisation de la force de traînée causée par la résistance de l'air et néglige les interactions dynamiques complexes comme les effets aérodynamiques et le battement des pales. Une approche matricielle pour la dérivation des équations de la dynamique du véhicule est présentée dans [Hossain et al. \(2010\)](#). [Mahony et al. \(2012\)](#) élabore plus en détails les effets aérodynamiques dominants sur le quadcoptère.

Différentes stratégies de contrôle pour la stabilisation de l'attitude des quadricoptères ont été élaborées et étudiées, notamment les contrôleurs PID ([Luukkonen \(2011\)](#), [Mahony et al. \(2012\)](#), [Ferry \(2017\)](#) et [Usman \(2020\)](#)), les contrôleurs à mode glissant ([Lee et al., 2009](#)), le contrôleur LQR ([Lee et al., 2009](#)) et le contrôle de linéarisation par rétroaction ([Lee et al., 2009](#)).

## Chapitre 2

# Dynamique du véhicule

Afin de développer les équations de la dynamique du véhicule, nous commencerons par décrire l'espace dans lequel évolue notre quadricoptère. Un corps rigide dans l'espace à 3 dimensions peut être décrit à l'aide de 6 coordonnées. Les trois premières coordonnées ( $x, y, z$ ) représentent la distance du centre de masse de l'objet par rapport à une origine dans l'espace tridimensionnel. Les 3 dernières coordonnées représentent l'orientation du véhicule données par trois angles :  $(\phi, \theta, \psi)$ . Si nous connaissons les 6 coordonnées, nous pouvons déterminer l'emplacement du centre de gravité à partir de la position ( $x, y, z$ ) et de l'orientation, ou de "l'attitude" du véhicule  $(\phi, \theta, \psi)$ , et nous pouvons déterminer l'emplacement du reste du véhicule par rapport au centre de gravité. (Usman, 2020)

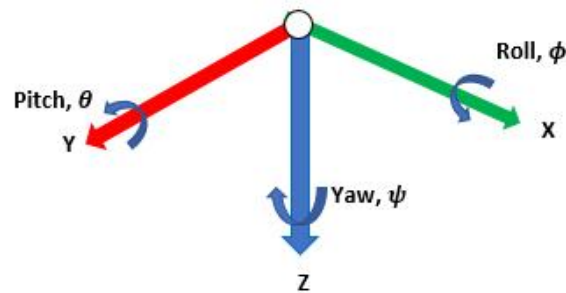


Figure 2.1 – Les six degrés de liberté du véhicule.

Les équations de dynamique du véhicule représentent simplement le changement d'orientation et de position du véhicule au cours du temps.

Dans ce travail, nous supposons que la terre est plate (repère local), la structure du quadricoptère est rigide, la force de gravité est constante, ce qui permet de faire coïncider le centre de masse au centre de gravité, nous traiterons donc ces termes comme interchangeables.

### 2.1 Orientation et repères de référence

Les trois coordonnées du vecteur de position ( $x, y, z$ ) sont, je pense, les plus simples à décrire. Ce repère de coordonnées pourrait être aligné avec le nord, l'est et la hauteur verticale et les composantes seraient à une certaine distance de l'origine dans chacune de ces directions respectives.

La description de l'attitude du véhicule n'est pas aussi intuitive à saisir. Il existe plusieurs moyens de représenter l'orientation d'un solide dans l'espace, les deux représentations courantes sont les angles d'Euler

et les quaternions.

Les angles d'Euler au nombre de trois ont été introduits par le mathématicien et physicien suisse Leonhard Euler, c'est la manière la plus simple de décrire l'orientation d'un solide en analysant directement son mouvement. Il s'agit du roulis, du tangage et du lacet. Le roulis peut être défini comme la rotation autour de l'axe X, le tangage peut être défini comme la rotation autour de l'axe Y et le lacet peut être défini comme la rotation autour de l'axe Z.

C'est la méthode que nous utiliserons pour dériver les équations du mouvement de notre véhicule. Cette méthode a cependant une limitation car pour certaines orientations, une singularité apparaît lorsque les trois coordonnées ne définissent pas une orientation unique. Cette singularité est connue sous le nom de "blocage de cardan" et peut être atténuée en utilisant des quaternions à la place. Bien souvent, les algorithmes de simulations et les contrôleurs de véhicules effectuent les calculs en quaternions et les convertissent ensuite en angles d'Euler en sortie en raison de leur facilité d'utilisation.

Pour la description de notre modèle, nous aurons deux systèmes de coordonnées :

1. Le repère inertiel

Il est considéré fixe par rapport à la Terre et les coordonnées sont souvent basées sur les directions cardinales (Nord, Est, bas souvent désigné par NED en anglais pour *North, East and Down*), nous définissons le cadre inertiel avec les coordonnées  $(x, y, z)$ .

2. Le repère lié au corps solide

Son origine est le centre de gravité du solide, il est fixe par rapport au corps avec ses axes de coordonnées  $(b_1, b_2, b_3)$  alignés avec la structure de telle sorte que  $b_1$  et  $b_2$  soient alignés avec les axes symétriques du corps et que  $b_3$  soit dans la direction verticale (parallèle aux axes des moteurs des hélices). Nous disposons maintenant d'un moyen pratique pour décrire la distance des hélices et leur direction par rapport au centre de gravité. Pour représenter l'orientation ou l'attitude du véhicule, les angles  $\phi$ ,  $\theta$  et  $\psi$  sont utilisés par rapport au référentiel du corps.

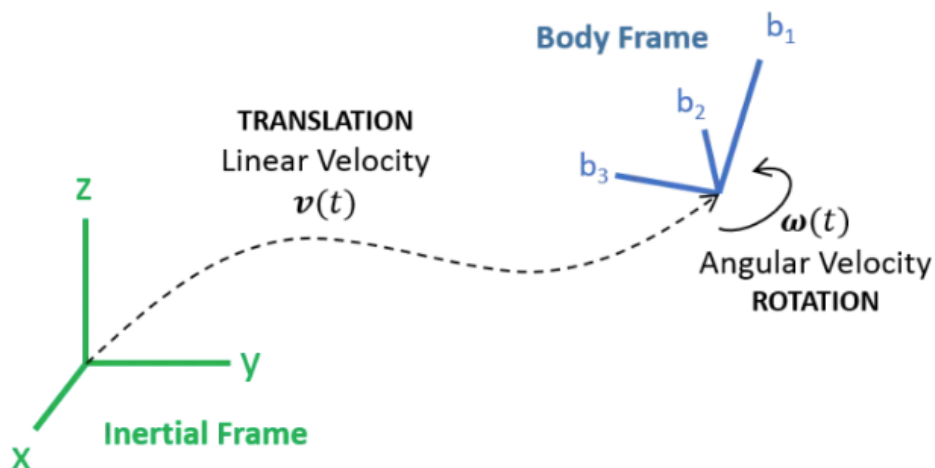


Figure 2.2 – Repère inertiel et repère lié au solide (Tytler, 2017)

Les axes de rotation du repère fixé au corps peuvent être orientés sous deux configurations, l'une étant la configuration en "x" et l'autre configuration en "+". La différence entre les deux provient de la direction des axes X et Y, ainsi que des moteurs utilisés pour changer de direction. Les moteurs 1 et 3 tournent dans le sens de rotation horaire (CW - *clockwise*) tandis que les moteurs 2 et 4 tournent dans le sens anti-horaire (CCW - *counterclockwise*), ceci permet de supprimer le couple de lacet et de découpler un mouvement à l'autre.

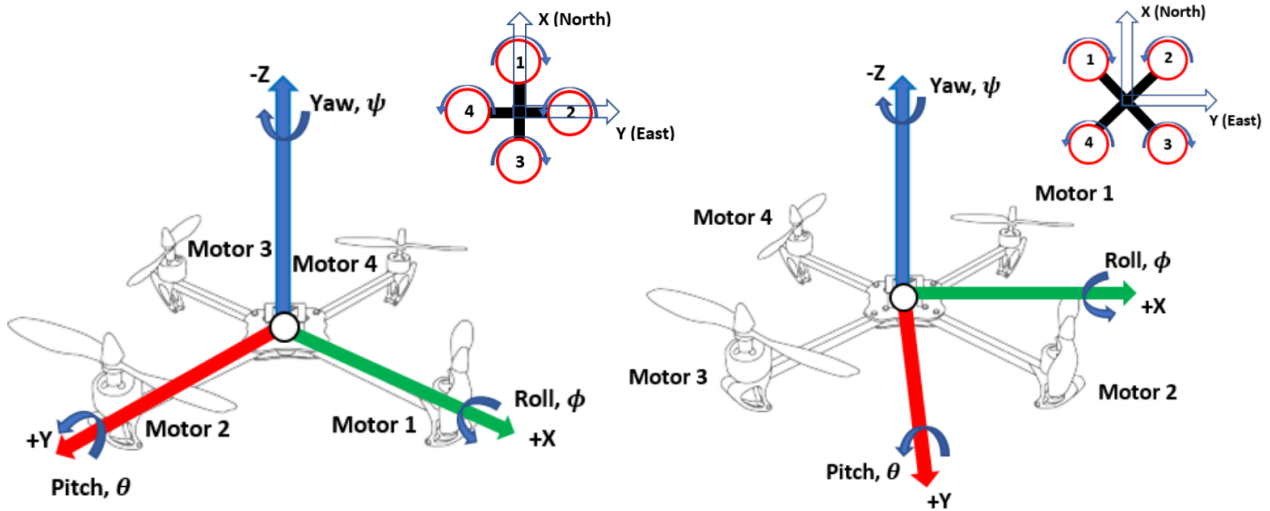


Figure 2.3 – Différentes configurations de l'orientation du quadcoptère : configuration en "+" à gauche et configuration en "x" à droite. (Ferry, 2017)

Maintenant que nous avons deux systèmes de coordonnées, il nous faut un moyen de passer de l'un à l'autre. Si nous pouvons établir une relation générale qui nous permet de passer d'un ensemble de coordonnées à l'autre, nous pouvons alors effectuer des calculs dans le cadre qui nous convient le mieux.

## 2.2 Les angles d'Euler

Les angles d'Euler permettent de définir l'orientation finale du véhicule par rapport au référentiel du corps en utilisant une matrice de transformation du référentiel inertiel en référentiel lié au corps.

Il est pratique d'utiliser la multiplication matricielle pour les transformations vectorielles.

La figure 2.4 montre la rotation des trois axes du référentiel lié au solide b1, b2 et b3 par rapport aux axes correspondants du cadre inertiel, les équations trigonométriques dans le repère inertiel et leurs représentations matricielles.

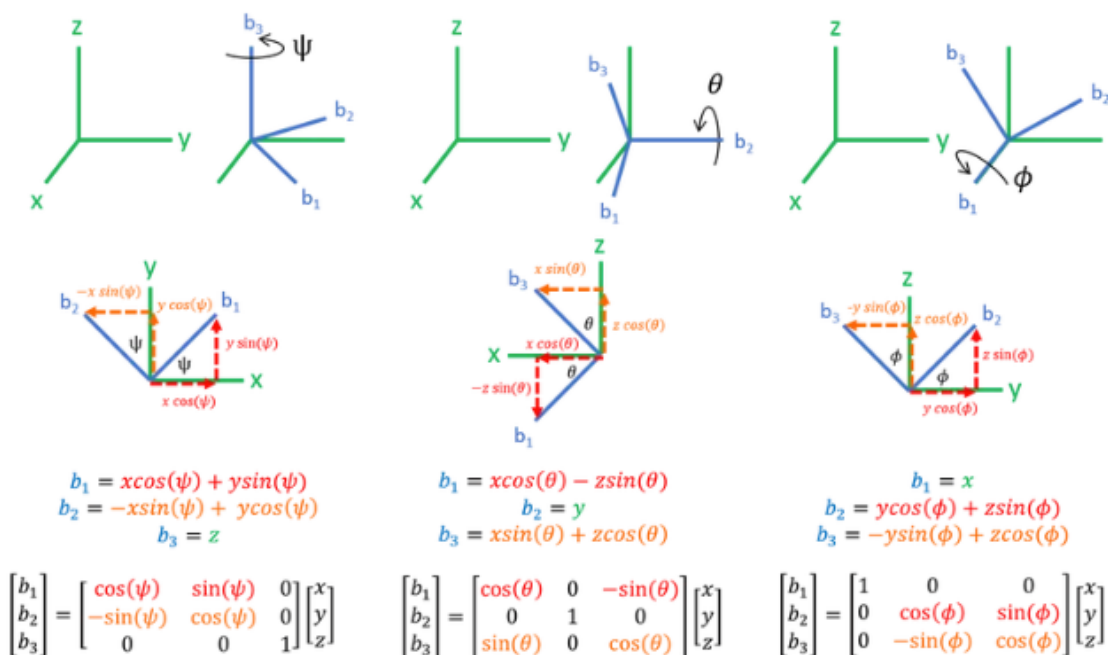


Figure 2.4 – Les angles d'Euler (Usman, 2020)

En combinant successivement les trois matrices de rotation (de la figure 2.4), on obtient la matrice de passage du repère inertiel au repère lié au corps. L'ordre des opérations est très important, ici la convention des angles d'Euler ZYX est appliquée, on "défait" successivement le lacet, le tangage et le roulis.

$$\begin{bmatrix} b_1 \\ b_2 \\ b_3 \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 \\ 0 & \cos(\phi) & \sin(\phi) \\ 0 & -\sin(\phi) & \cos(\phi) \end{bmatrix} \begin{bmatrix} \cos(\theta) & 0 & -\sin(\theta) \\ 0 & 1 & 0 \\ \sin(\theta) & 0 & \cos(\theta) \end{bmatrix} \begin{bmatrix} \cos(\Psi) & \sin(\Psi) & 0 \\ -\sin(\Psi) & \cos(\Psi) & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x \\ y \\ z \end{bmatrix} \quad (2.1)$$

Les trois matrices de rotation sont multipliées pour obtenir une seule matrice de transformation  $C_n^b$  donnée par l'équation [2.3] qui sera utilisée pour la transformation du repère inertiel au repère lié au corps.

$$\begin{bmatrix} b_1 \\ b_2 \\ b_3 \end{bmatrix} = \mathbf{C}_n^b \begin{bmatrix} x \\ y \\ z \end{bmatrix} \quad (2.2)$$

$$\begin{bmatrix} b_1 \\ b_2 \\ b_3 \end{bmatrix} = \begin{bmatrix} \cos(\theta) \cos(\Psi) & \cos(\theta) \sin(\Psi) & -\sin(\theta) \\ -\cos(\phi) \sin(\Psi) + \cos(\Psi) \sin(\theta) \sin(\phi) & \cos(\Psi) \cos(\phi) + \sin(\theta) \sin(\phi) \sin(\Psi) & \cos(\theta) \sin(\phi) \\ \sin(\Psi) \sin(\phi) + \cos(\Psi) \cos(\phi) \sin(\theta) & -\sin(\phi) \cos(\Psi) + \cos(\phi) \sin(\theta) \sin(\Psi) & \cos(\theta) \cos(\phi) \end{bmatrix} \begin{bmatrix} x \\ y \\ z \end{bmatrix} \quad (2.3)$$

Pour le passage du repère lié au corps au repère inertiel, on inverse théoriquement la matrice de transformation  $C_n^b$ . Les matrices de transformation étant orthonormées, nous pouvons simplement utiliser ici la transposée de la matrice de transformation  $R_t = R^{-1}$ .

$$\begin{bmatrix} x \\ y \\ z \end{bmatrix} = \mathbf{C}_b^n \begin{bmatrix} b_1 \\ b_2 \\ b_3 \end{bmatrix} = [\mathbf{C}_n^b]^T \begin{bmatrix} b_1 \\ b_2 \\ b_3 \end{bmatrix} \quad (2.4)$$

$$\begin{bmatrix} x \\ y \\ z \end{bmatrix} = \begin{bmatrix} \cos(\theta) \cos(\psi) & -\cos(\phi) \sin(\psi) + \sin(\phi) \sin(\theta) \cos(\psi) & \sin(\phi) \sin(\psi) + \cos(\phi) \sin(\theta) \cos(\psi) \\ \cos(\theta) \sin(\psi) & \cos(\phi) \cos(\psi) + \sin(\phi) \sin(\theta) \sin(\psi) & -\sin(\phi) \cos(\psi) + \cos(\phi) \sin(\theta) \sin(\psi) \\ -\sin(\theta) & \sin(\phi) \cos(\theta) & \cos(\phi) \cos(\theta) \end{bmatrix} \begin{bmatrix} b_1 \\ b_2 \\ b_3 \end{bmatrix} \quad (2.5)$$

Les équations [2.3] et [2.5] permettent d'obtenir l'orientation du véhicule par rapport au repère lié au solide ou au repère inertiel selon le besoin. (Usman, 2020)

## 2.3 Les équations du mouvement

Avant de plonger dans les détails des équations du mouvement de notre quadricoptère, il convient d'identifier quelles sont les variables qui nous permettent de décrire le mouvement du drone.

Nous utilisons la règle de la main droite pour la convention de signe des différents repères.

Afin de dériver les équations de la dynamique de notre système quelques hypothèses ont été formulées.

- Le drone est une structure rigide
- La structure du drone et les composants sont symétriques.
- Le centre de gravité et l'origine du cadre du drone coïncident.
- La poussée et la traînée sont proportionnelles au carré de la vitesse de l'hélice.
- Il n'y a pas de perturbations externes au quadricoptère telles que le vent, la température, etc.

Symbole	Désignation	Unités
$V^b$ Vitesses linéaires dans le repère lié au solide		
u	vitesse longitudinale	$m/s$
v	vitesse latérale	$m/s$
w	vitesse normale	$m/s$
$X$ Vitesses linéaires dans le repère inertiel		
$\dot{x}$	vitesse suivant x	$m/s$
$\dot{y}$	vitesse suivant y	$m/s$
$\dot{z}$	vitesse suivant z	$m/s$
$W^b$ Vitesses angulaires dans le repère lié au solide		
p	vitesse angulaire de roulis	$rad/s$
q	vitesse ang. de tangage	$rad/s$
r	vitesse angulaire de lacet	$rad/s$
$X$ Position dans le repère inertiel		
x	position sur l'axe x	$m$
y	position sur l'axe y	$m$
z	position sur l'axe z	$m$
Les angles d'Euler (orientation dans le repère lié au corps)		
$\phi$	angle de roulis	$rad$
$\theta$	angle de tangage	$rad$
$\psi$	angle de lacet	$rad$
Vitesses de rotation des moteurs		
$\omega_i$	vitesse de rotation du moteur i	$rad/s$
Forces		
$F_x$	Force suivant la direction x	$N$
$F_y$	Force suivant la direction y	$N$
$F_z$	Force suivant la direction z	$N$
Moments		
L	moment angulaire autour de x	$Nm$
M	moment angulaire autour de y	$Nm$
N	moment angulaire autour de z	$Nm$

Table 2.1 – Les différentes variables du modèle

### Composition des mouvements et dérivation vectorielle

Nous voulons exprimer le mouvement du repère inertiel dans le repère lié au solide. Cela entraîne quelques considérations mathématiques pour le calcul de la dérivée des vecteurs qui sont exprimés dans le repère inertiel. La règle de dérivation en chaîne donne la dérivée à la fois du changement dû à la dérivée temporelle du vecteur dans le repère de coordonnées et de la dérivée temporelle de la rotation du repère de coordonnées.

$$V^b = \begin{bmatrix} u \\ v \\ w \end{bmatrix}^b = u\vec{b}_1 + v\vec{b}_2 + w\vec{b}_3 \quad (2.6)$$

Où  $\vec{b}_1$ ,  $\vec{b}_2$  et  $\vec{b}_3$  dans l'équation 2.6 sont des vecteurs unitaires associés au repère du corps pour représenter la vitesse du repère inertiel dans le repère lié au corps.

En appliquant la règle de dérivation en chaîne à l'équation 2.6, on obtient :

$$\left(\frac{dV}{dx}\right)_{\text{inertial}} = \left(\frac{du}{dt}\vec{b}_1 + \frac{dv}{dt}\vec{b}_2 + \frac{dw}{dt}\vec{b}_3\right) + \left(u\frac{d\vec{b}_1}{dt} + v\frac{d\vec{b}_2}{dt} + w\frac{d\vec{b}_3}{dt}\right) \quad (2.7)$$

Dans l'équation 2.7 ci-dessus, la première série de parenthèses représente la vitesse inertielle dans les coordonnées du repère du corps. La deuxième série représente le changement de vitesse dû à la rotation du repère de coordonnées. L'équation 2.7 ci-dessus peut être écrite comme l'équation 2.8 en utilisant la formule de Bour, qui est le produit croisé de la vitesse angulaire par le vecteur vitesse représentant la rotation du cadre. (Usman, 2020).

$$\dot{V}_{\text{inertial}} = \dot{V}^b + \mathcal{W}_n^b \times V^b \quad (2.8)$$

La multiplication vectorielle de la dernière équation peut être représentée comme une multiplication par une matrice asymétrique.

$$\mathcal{W} \times v = [\mathcal{W} \times] v = \begin{bmatrix} 0 & -\omega_z & \omega_y \\ \omega_z & 0 & -\omega_x \\ -\omega_y & \omega_x & 0 \end{bmatrix} v \quad (2.9)$$

## La seconde loi du mouvement de Newton

Pour dériver les équations de mouvement linéaire et de rotation, nous utilisons la deuxième loi du mouvement de Newton, qu'on peut resumer comme suit : la force (F) égale la variation de la quantité de mouvement (P) ou de la masse (m) multipliée par l'accélération (a) ou la dérivée de la vitesse (v).

$$\mathbf{F} = \frac{d\mathbf{p}}{dt} = m \frac{d\mathbf{v}}{dt} = m\mathbf{a} \quad (2.10)$$

Cette équation définit la variation du moment cinétique linéaire, l'analogue en rotation peut être également exprimer en multipliant les deux côtés de l'équation avec un vecteur position (r) à partir d'une origine O, nous obtenons la loi de rotation du mouvement autour de l'origine O. Ainsi le moment (M) est égal à la variation du moment angulaire (H) ou, pour les systèmes à distribution de masse constante, le moment est égal au moment d'inertie (I) multiplié par l'accélération angulaire ( $\omega$ ) - la dérivée de la vitesse angulaire ( $\omega$ ).

$$\vec{r} \times \mathbf{F} = \vec{r} \times m \frac{d\mathbf{v}}{dt} \quad (2.11)$$

$$\mathbf{M} = \frac{d\mathbf{H}}{dt} = \mathbf{I} \frac{d\omega}{dt} = \mathbf{I}\Omega \quad (2.12)$$

Nous allons dériver séparément les équations de mouvement en translation et en rotation.

## Forces et moments extérieurs

### 1. Les moteurs

Les moteurs sans balais sont utilisés dans notre quadricoptère. Pour notre moteur électrique, le couple produit est donné par :

$$\tau = K_t (I - I_0) \quad (2.13)$$

Où  $\tau$  est le couple moteur, I est le courant d'entrée,  $I_0$  est le courant lorsqu'il n'y a pas de charge sur le moteur, et  $K_t$  est la constante de proportionnalité du couple.

La tension aux bornes du moteur est donnée par :

$$V = IR_m + K_v \omega \quad (2.14)$$

Où V est la chute de tension aux bornes du moteur,  $R_m$  est la résistance du moteur,  $\omega$  est la vitesse angulaire du moteur, et  $K_v$  est une constante de proportionnalité (indiquant la force contre-électromotrice produite par tr/min).

La puissance consommée s'écrit alors :

$$P = IV = \frac{(\tau + K_t l_0)(K_t l_0 R_m + \tau R_m + K_t K_v \omega)}{K_t^2} \quad (2.15)$$

Nous supposons que la résistance du moteur est négligeable et que  $K_t l_0 \ll \tau$ . Nous obtenons ainsi une expression simplifiée de la puissance consommée. (Gibiansky, 2012)

$$P \approx \frac{K_v}{K_t} \tau \omega \quad (2.16)$$

## 2. La poussée

La poussée produite par les hélices du quadcopter agit perpendiculairement au véhicule. Le théorème de la conservation de l'énergie permet d'établir que l'énergie dépensée par le moteur pendant un temps donné est égale à la force générée sur l'hélice multipliée par la distance parcourue par l'air qu'elle déplace ( $Pdt = Fdx$ ). Ainsi, la puissance est égale à la poussée multipliée par la vitesse de l'air ( $P = F \frac{dx}{dt}$ ).

$$P = T v_h \quad (2.17)$$

On suppose que la vitesse du véhicule est faible, donc  $v_h$  est la vitesse de l'air en vol stationnaire. On suppose également que la vitesse du flux libre,  $v_\infty$ , est nulle (l'air dans l'environnement est stationnaire par rapport au quadricoptère). Le théorème de la quantité de mouvement nous donne l'équation de la vitesse de vol stationnaire en fonction de la poussée.

$$v_h = \sqrt{\frac{T}{2\rho A}} \quad (2.18)$$

Où  $\rho$  est la densité de l'air environnant et  $A$  est la surface balayée par le rotor. En utilisant notre équation simplifiée pour la puissance 2.16, nous pouvons alors écrire :

$$P = \frac{K_v}{K_t} \tau \omega = \frac{K_v K_\tau}{K_t} T \omega = \frac{T^{\frac{3}{2}}}{\sqrt{2\rho A}}. \quad (2.19)$$

En général,  $\tau = \vec{r} \times \vec{F}$ ; dans ce cas, le couple est proportionnel à la poussée  $T$  par un certain rapport constant  $K_\tau$  déterminé par la configuration et les paramètres de la pale.

En résolvant pour de la poussée  $T$ , on obtient que la poussée est proportionnelle au carré de la vitesse angulaire du moteur. (Gibiansky, 2012)

$$T = \left( \frac{K_v K_\tau \sqrt{2\rho A}}{K_t} \omega \right)^2 = b \omega^2 \quad (2.20)$$

Où  $b$  est une constante de dimension appropriée, par souci de simplification nous l'appellerons coefficient de poussée (équivalence à un facteur dimensionnel près, normalement le coefficient de poussée est sans dimension). En additionnant pour tous les moteurs, on trouve que la poussée totale sur le quadricoptère (dans le repère lié au corps) est donnée par :

$$\begin{bmatrix} F_x \\ F_y \\ F_z \end{bmatrix} = b \begin{bmatrix} 0 \\ 0 \\ \sum \omega_i^2 \end{bmatrix} \quad (2.21)$$

## 3. Les couples

### 3.1 Moments de roulis et de tangage

Les moments sont simplement le produit des forces par la distance au centre de gravité suivant les trois axes du repère du corps.

L'équation 2.22 donne le moment du véhicule autour de l'axe x.

$$L = bl(F_1 - F_2 - F_3 + F_4) \quad (2.22)$$

L'équation 2.23 donne le moment du véhicule autour de l'axe y.

$$M = bl(-F_1 + F_2 - F_3 + F_4) \quad (2.23)$$

Où  $F_i$  est la poussée développée par le moteur  $i$  et  $l$  est la distance entre le centre de masse du quadcopter et le centre d'une hélice (rotor).

### 3.2 Moment de lacet

On rappelle que deux hélices tournent dans le sens des aiguilles d'une montre et les deux autres tournent dans le sens inverse pour équilibrer le mouvement de lacet.

En dynamique des fluides, la traînée est donnée par l'équation :

$$F_D = \frac{1}{2} \rho C_D A v^2 \quad (2.24)$$

Où  $\rho$  est la densité du fluide environnant,  $A$  est la surface de référence (section transversale de l'hélice) et  $C_D$  est une constante sans dimension. Cette méthode n'est précise que dans certains cas, elle est suffisamment bonne pour notre modélisation. Ainsi, le couple dû à la traînée s'écrit alors :

$$\tau_D = \frac{1}{2} R \rho C_D A v^2 = \frac{1}{2} R \rho C_D A (\omega R)^2 = d \omega^2 \quad (2.25)$$

Avec  $\omega$  la vitesse angulaire de l'hélice,  $R$  le rayon de l'hélice, et  $d$  est une constante de dimension appropriée, par souci de simplification nous l'appellerons coefficient de traînée (équivalence à un facteur dimensionnel près, normalement le coefficient de traînée est sans dimension).

Nous avons supposé dans la formule ci-dessus que toute la force est appliquée à l'extrémité de l'hélice, ce qui est inexact ; cependant, le seul résultat qui importe pour notre propos est que le couple de traînée est proportionnel au carré de la vitesse angulaire. Nous pouvons alors écrire le couple complet autour de l'axe z pour le  $i$ ème moteur :

$$\tau_z = d \omega^2 + I_M \dot{\omega} \quad (2.26)$$

Où  $I_M$  est le moment d'inertie sur l'axe z du moteur,  $\dot{\omega}$  est l'accélération angulaire de l'hélice l'hélice, et  $d$  est notre coefficient de traînée. Notons qu'en vol stationnaire  $\dot{\omega}$ , puisque la plupart du temps, les hélices maintiendront une poussée constante (ou presque constante) et ne seront pas en train d'accélérer, nous ignorons ce terme en simplifiant l'expression entière à :

$$\tau_z = (-1)^{i+1} d \omega_i^2 \quad (2.27)$$

Où le terme  $(-1)^{i+1}$  est positif pour la  $i$ ème hélice si l'hélice tourne dans le sens horaire et négatif si elle tourne dans le sens inverse des aiguilles d'une montre.

Le moment total du véhicule autour de l'axe z est donné par la somme de tous les couples de chaque hélice. (Gibiansky, 2012)

$$N = d (\omega_1^2 - \omega_2^2 + \omega_3^2 - \omega_4^2) \quad (2.28)$$

#### 4. Force de gravité dans le repère du corps

La force de gravité agissant sur le véhicule peut être exprimée dans le repère lié au corps en utilisant les angles d'Euler notamment la matrice de transformation  $C_n^b$  de l'équation 2.3. Où  $C_n^b$  est une matrice de passage du repère inertiel au repère lié au corps,  $F_g^n$  et  $F_g^b$  représentent la force de gravité dans le repère inertiel et dans celui fixé au corps respectivement. L'équation 2.29 donne la force de gravité exprimée dans le repère inertiel.

$$F_g^n = \begin{bmatrix} 0 \\ 0 \\ mg \end{bmatrix} \quad (2.29)$$

En multipliant l'équation 2.29 par la matrice de transformation de repère, on obtient la force de gravité exprimée dans le repère du corps :

$$F_g^b = C_n^b F_g^n = \begin{bmatrix} \cos(\theta) \cos(\Psi) & \cos(\theta) \sin(\Psi) & -\sin(\theta) \\ -\cos(\phi) \sin(\Psi) + \cos(\Psi) \sin(\theta) \sin(\phi) & \cos(\Psi) \cos(\phi) + \sin(\theta) \sin(\phi) \sin(\Psi) & \cos(\theta) \sin(\phi) \\ \sin(\Psi) \sin(\phi) + \cos(\Psi) \cos(\phi) \sin(\theta) & -\sin(\phi) \cos(\Psi) + \cos(\phi) \sin(\theta) \sin(\Psi) & \cos(\theta) \cos(\phi) \end{bmatrix} \begin{bmatrix} 0 \\ 0 \\ mg \end{bmatrix} \quad (2.30)$$

Après simplification, on a finalement :

$$F_g^b = \begin{bmatrix} -mg \sin(\theta) \\ mg \sin(\phi) \cos(\theta) \\ mg \cos(\phi) \cos(\theta) \end{bmatrix} \quad (2.31)$$

#### Matrices d'inertie

Le moment d'inertie d'un solide caractérise son inertie vis-à-vis des mouvements de rotation, comme sa masse caractérise son inertie vis-à-vis des mouvements de translation. Le moment d'inertie donne la quantité de moment nécessaire pour faire tourner un objet immobile ou le moment nécessaire pour arrêter un objet en rotation.

Nous regroupons les moments autour des trois axes du véhicule sous forme matricielle.

$$I = \begin{bmatrix} I_{xx} & -I_{xy} & -I_{xz} \\ -I_{xy} & I_{yy} & -I_{yz} \\ -I_{xz} & -I_{yz} & I_{zz} \end{bmatrix} \quad (2.32)$$

Nous avons supposé que notre quadricoptère est symétrique. Pour un corps symétrique, les moments d'inertie sur les côtés opposés du véhicule s'annulent mutuellement. La matrice d'inertie se simplifie comme suit :

$$I = \begin{bmatrix} I_{xx} & 0 & 0 \\ 0 & I_{yy} & 0 \\ 0 & 0 & I_{zz} \end{bmatrix} \quad (2.33)$$

#### Accélérations linéaires et Équation du mouvement en translation

En utilisant la formule de Bour (équation 2.8), nous pouvons convertir l'accélération inertielle dans le repère du corps. L'équation 2.34 montre la conversion de l'accélération inertielle en accélération dans le repère lié au corps solide.

$$\dot{V}_b = \begin{bmatrix} \dot{u} \\ \dot{v} \\ \dot{w} \end{bmatrix}^b + \begin{bmatrix} 0 & -r & q \\ r & 0 & -p \\ -q & p & 0 \end{bmatrix} \begin{bmatrix} u \\ v \\ w \end{bmatrix}^b = \begin{bmatrix} \dot{u} + qw - rv \\ \dot{v} + ru - pw \\ \dot{w} + pv - qu \end{bmatrix} \quad (2.34)$$

En mettant la somme des forces appliquées et les accélérations linéaires exprimées dans le repère du corps dans la seconde loi de Newton exprimée précédemment, on obtient les équations linéaires du mouvement 2.35.

$$\begin{bmatrix} -mg \sin(\theta) \\ mg \sin(\phi) \cos(\theta) \\ -F_1 - F_2 - F_3 - F_4 + mg \cos(\phi) \cos(\theta) \end{bmatrix} = m \begin{bmatrix} \dot{u} + qw - rv \\ \dot{v} + ru - pw \\ \dot{w} + pv - qu \end{bmatrix} \quad (2.35)$$

### Accélération angulaire et Équation du mouvement en rotation

En utilisant à nouveau la formule de Bour (équation 2.8) pour la dérivée temporelle de la vitesse angulaire, nous pouvons évaluer la formule du mouvement de rotation comme suit :

$$\mathbf{M} = \mathbf{I}^b \dot{\boldsymbol{\omega}}_n^b + \boldsymbol{\omega}_n^b \times \mathbf{I}^b \boldsymbol{\omega}_n^b \quad (2.36)$$

$$\begin{bmatrix} L \\ M \\ N \end{bmatrix} = \begin{bmatrix} I_{xx} & 0 & 0 \\ 0 & I_{yy} & 0 \\ 0 & 0 & I_{zz} \end{bmatrix} \begin{bmatrix} \dot{p} \\ \dot{q} \\ \dot{r} \end{bmatrix} + \begin{bmatrix} 0 & -r & q \\ r & 0 & -p \\ -q & p & 0 \end{bmatrix} \begin{bmatrix} I_{xx} & 0 & 0 \\ 0 & I_{yy} & 0 \\ 0 & 0 & I_{zz} \end{bmatrix} \begin{bmatrix} p \\ q \\ r \end{bmatrix} \quad (2.37)$$

$$\begin{bmatrix} L \\ M \\ N \end{bmatrix} = \begin{bmatrix} I_{xx} \dot{p} \\ I_{yy} \dot{q} \\ I_{zz} \dot{r} \end{bmatrix} + \begin{bmatrix} -I_{yy}qr + I_{zz}qr \\ I_{xx}pr - I_{zz}pr \\ -I_{xx}pq + I_{yy}pq \end{bmatrix}$$

### Les vitesses angulaires et les angles d'Euler

Nous savons comment les vitesses angulaires ( $p$ ,  $q$ ,  $r$ ) évoluent dans le temps. Ces valeurs nous seront fournies par le capteur gyroscopique monté sur drone. Une mise en garde s'impose toutefois : les vitesses angulaires dans le repère fixe du corps ne sont pas les mêmes que les taux de variation des angles d'Euler ( $\phi$ ,  $\theta$ ,  $\psi$ ). Les angles d'Euler représentent une séquence de rotations, chacune dans son propre repère de référence intermédiaire, alors que les vitesses angulaires sont les vitesses angulaires instantanées autour de chacun des axes du repère fixé au corps. (Tytler, 2017)

En utilisant la transformation des coordonnées, les vitesses angulaires peuvent être exprimées en fonction de dérivées des angles d'Euler.

$$\begin{bmatrix} p \\ q \\ r \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 \\ 0 & c_\phi & s_\phi \\ 0 & -s_\phi & c_\phi \end{bmatrix} \begin{bmatrix} c_\theta & 0 & -s_\theta \\ 0 & 1 & 0 \\ s_\theta & 0 & c_\theta \end{bmatrix} \begin{bmatrix} 0 \\ 0 \\ \dot{\psi} \end{bmatrix} + \begin{bmatrix} 1 & 0 & 0 \\ 0 & c_\phi & s_\phi \\ 0 & -s_\phi & c_\phi \end{bmatrix} \begin{bmatrix} 0 \\ \dot{\theta} \\ 0 \end{bmatrix} + \begin{bmatrix} \dot{\phi} \\ 0 \\ 0 \end{bmatrix} \quad (2.38)$$

$$\begin{bmatrix} p \\ q \\ r \end{bmatrix} = \begin{bmatrix} -\dot{\psi} s_\theta \\ \dot{\psi} c_\theta s_\phi \\ \dot{\psi} c_\theta c_\phi \end{bmatrix} + \begin{bmatrix} 0 \\ \dot{\theta} c_\phi \\ -\dot{\theta} s_\phi \end{bmatrix} + \begin{bmatrix} \dot{\phi} \\ 0 \\ 0 \end{bmatrix}$$

Nous obtenons finalement l'ensemble des équations suivantes.

$$p = \dot{\phi} - \dot{\psi} \sin \theta \quad (2.39)$$

$$q = \dot{\theta} \cos \phi + \dot{\psi} \cos \theta \sin \phi \quad (2.40)$$

$$r = \dot{\psi} \cos \phi \cos \theta - \dot{\theta} \sin \phi \quad (2.41)$$

### Les coordonnées de navigation

Afin de déterminer la position du véhicule dans le repère inertiel, il suffit d'utiliser la matrice de transformation d'angles d'Euler  $C_b^n$  (équation 2.5) pour convertir les vitesses exprimées dans le repère lié au corps en vitesses en coordonnées inertielles. Où  $x^E$ ,  $y^E$  et  $z^E$  représentent les dérivées des positions X,

Y et Z du véhicule dans le repère inertiel. Nous définissons également la variable de hauteur,  $h$ , comme l'opposé de celui du repère inertiel Nord-Est-Bas (*NED - North-East-Down*) pour obtenir un repère de navigation Nord-Est-Haut (*pour North-East-Up en anglais*). (Tytler, 2017)

$$\begin{bmatrix} \dot{x}^E \\ \dot{y}^E \\ -\dot{h}^E \end{bmatrix} = \mathbf{C}_b^n \begin{bmatrix} u \\ v \\ w \end{bmatrix}^b = \begin{bmatrix} c_\theta c_\psi & -c_\theta s_\psi + s_\phi s_\theta c_\psi & s_\phi s_\psi + c_\phi s_\theta c_\psi \\ c_\theta s_\psi & c_\phi c_\psi + s_\phi s_\theta s_\psi & -s_\phi c_\psi + c_\phi s_\theta s_\psi \\ -s_\theta & s_\phi c_\theta & c_\phi c_\theta \end{bmatrix} \begin{bmatrix} u \\ v \\ w \end{bmatrix}^b \quad (2.42)$$

### Les équations du mouvement du quadricoptère

En rassemblant toutes les équations décrites précédemment en un seul ensemble et en les réarrangeant, nous arrivons à la description complète du mouvement du quadricoptère. (Tytler, 2017)

$$\dot{u} = -g \sin(\theta) + rv - qw \quad (2.43)$$

$$\dot{v} = g \sin(\phi) \cos(\theta) - ru + pw \quad (2.44)$$

$$\dot{w} = \frac{1}{m} (-F_z) + g \cos(\phi) \cos(\theta) + qu - pv \quad (2.45)$$

$$\dot{p} = \frac{1}{I_{xx}} (L + (I_{yy} - I_{zz}) qr) \quad (2.46)$$

$$\dot{q} = \frac{1}{I_{yy}} (M + (I_{zz} - I_{xx}) pr) \quad (2.47)$$

$$\dot{r} = \frac{1}{I_{zz}} (N + (I_{xx} - I_{yy}) pq) \quad (2.48)$$

$$\dot{\phi} = p + (q \sin \phi + r \cos \phi) \tan \theta \quad (2.49)$$

$$\dot{\theta} = q \cos \phi - r \sin \phi \quad (2.50)$$

$$\dot{\psi} = (q \sin \phi + r \cos \phi) \sec \theta \quad (2.51)$$

$$\dot{x}^E = c_\theta c_\psi u^b + (-c_\phi s_\psi + s_\phi s_\theta c_\psi) v^b + (s_\phi s_\psi + c_\phi s_\theta c_\psi) w^b \quad (2.52)$$

$$\dot{y}^E = c_\theta s_\psi u^b + (c_\phi c_\psi + s_\phi s_\theta s_\psi) v^b + (-s_\phi c_\psi + c_\phi s_\theta s_\psi) w^b \quad (2.53)$$

$$\dot{h}^E = -s_\theta u^b + s_\phi c_\theta v^b + c_\phi c_\theta w^b \quad (2.54)$$

# Chapitre 3

## Réalisation du prototype

### 3.1 Composants

Les composants (figure 3.1 et tableau 3.1) pour réaliser le modèle réel du drone sont sélectionnés en fonction de leurs performances et de leur compatibilité avec les autres composants sélectionnés. Pour le choix du microcontrôleur et de la centrale inertielle (*IMU - Inertial Measurement Unit*), nous avons pris en compte également la simplicité de prise en main et le support de cette électronique par l'environnement de simulation et de programmation Matlab Simulink<sup>1</sup>.

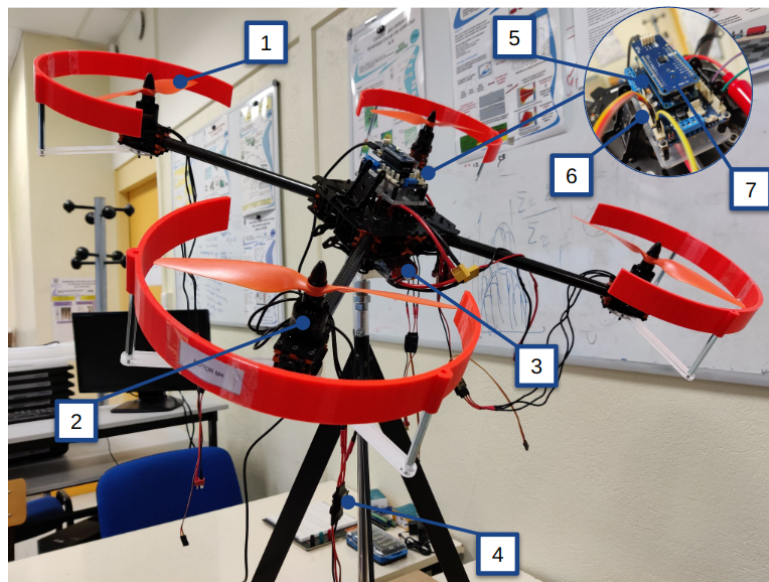


Figure 3.1 – Composants du quadcopter

Repère	Désignation	Nom commercial	Quantité
-	Cadre/Structure reconfigurable	Lynxmotion Multirotor Erector Set (MES)	1
1	Hélice	Lynxmotion 9x4.5R Propeller	4
2	Moteurs à courant continu sans balais	Lynxmotion Brushless Motor 28x30 1000kv	4
3	Batterie	3S LiPo 11.1 V 3800 mAh Conrad energy	1
4	Contrôleur électronique de vitesse	Lynxmotion 12A ESC (SimonK)	4
5	Microcontrôleur	Arduino MKR WiFi 1010	1
6	Carte d'extension pour les moteurs	Arduino MKR Motor Carrier	1
7	Centrale inertielle	Arduino MKR IMU Shield (BNO055 IMU)	1

Table 3.1 – Liste des composants du drone

1. Les bibliothèques Simulink Support Package for Arduino Hardware et MATLAB Support Package for Arduino Hardware

### 3.1.1 Le cadre

Il s'agit du Kit de cadre de drone reconfigurable de l'entreprise Lynxmotion<sup>2</sup>. Le Lynxmotion Multirotor Erector Set (MES) - Reconfigurable Frame est un cadre tout-en-un conçu pour permettre à l'utilisateur de tester facilement une variété de configurations différentes de multirotors. Le cadre utilise des colliers de serrage en aluminium, des plaques en composite G10 et des tubes en fibre de carbone. Une grande variété de designs différents peut être créée en utilisant ce système de cadre unique et chacun peut varier en termes de longueur de bras, de configurations de moteur et d'accessoires. Les matériaux choisis par le concepteur de ce kit permet de réduire considérablement la masse du drone.



Figure 3.2 – Cadre du quadricoptère (Lynxmotion Multirotor Erector Set)

### 3.1.2 Les hélices

Lorsque les hélices tournent, elles produisent une poussée vers le bas pour vaincre la force de gravité permettant ainsi au quadricoptère de voler. Les quadricoptères peuvent utiliser plusieurs types d'hélices, notamment des hélices à poussée ou à traction (*push or pull en anglais*) ainsi que des hélices comportant plus de deux pales. Plusieurs variables permettent d'effectuer un choix d'hélice notamment le pas, le diamètre, la corde et le matériau. Les paramètres caractéristiques pas et diamètre permettent d'identifier la puissance et la poussée. Chaque hélice doit au moins fournir une force de poussée égale au quart du poids du véhicule tout en tournant dans la plage de régime du moteur pour effectuer un vol stationnaire. (Ferry, 2017)



Figure 3.3 – Les hélices (Lynxmotion 9x4.5R Propeller)

---

2. Lynxmotion - <http://www.lynxmotion.com/>

### 3.1.3 Les moteurs à courant continu sans balais - Brushless DC Motors

Les moteurs à courant continu sans balais (Brushless DC Motors - BLDC) sont majoritairement utilisés pour les quadricoptères, car ils présentent plusieurs avantages en comparaison avec les moteurs avec balais. Le premier avantage est le rendement, ces moteurs fournissent plus de couple que les moteurs à balais. Le second avantage est la contrôlabilité, les moteurs sans balais peuvent être contrôlés, à l'aide de mécanismes de rétroaction, pour fournir précisément le couple et la vitesse de rotation souhaités. Un dernier avantage serait associé à la durabilité de ces moteurs, ils n'ont pas besoin de maintenance aussi souvent, comme les moteurs à balais.

Il existe deux types de moteurs à courant continu sans balais, les moteurs à rotation interne et les moteurs à rotation externe selon que c'est la partie intérieure ou extérieure du moteur qui tourne. Les moteurs à rotation interne tournent plus rapidement que les moteurs à rotation externe, tandis que ces derniers fournissent plus de couple, ce qui les rend populaires pour les quadricoptères. Les moteurs sont disponibles avec différents  $K_v$  qui donnent le régime maximal du moteur lorsqu'il est multiplié par la tension maximale fournie par la batterie (Maker, 2016). Le tableau 3.2 donne les caractéristiques des moteurs utilisés.



Figure 3.4 – Les moteurs à courant continu sans balais (Lynxmotion Brushless Motor 28x30 1000kv)

Batterie	Batterie Cellules LiPo 2-4S/11.1~14.8V
$K_v$	1000 (RPM/Volt)
Courant continu maximal	15 A
Dimensions	27.7mm x 32mm
Diamètre de l'arbre	3mm
Masse	58g

Table 3.2 – Spécifications du moteur à courant continu sans balais Lynxmotion Brushless Motor 28x30 1000kv

### 3.1.4 Contrôleur électronique de vitesse - Electronic Speed Controllers (ESCs)

Les contrôleurs électroniques de vitesse permettent comme leur nom le suggère de contrôler la vitesse des moteurs à courant continu sans balais en fonction des signaux émis par le microcontrôleur. Il s'agit d'une carte électronique qui assure le contrôle et l'alimentation du moteur en courant alternatif triphasé. Les ESCs sont choisis en fonction du courant maximal tiré par les moteurs. Le tableau 3.2 montre que chaque moteur peut tirer un courant maximal de 15 ampères. Notre véhicule ne sera que très peu voire pas sollicité pour demander un courant de 15 A. Prenant en compte ce dernier facteur, les contrôleurs électroniques de vitesse de 12A (Figure 3.5) ont été sélectionnés pour contrôler les moteurs. Le tableau 3.3 présente les spécifications des ESCs sélectionnés.



Figure 3.5 – Contrôleur électronique de vitesse (Lynxmotion 12A ESC SimonK)

Courant maximal tiré en continu	12A
Pic de courant	16A
Plage de tension	2-4S LiPo, 6-12 Cells NiMh
Fréquence d'entrée	50Hz
Micrologiciel	SimonK Firmware for MultiRotors
Dimensions	27mm x 22mm x 7mm
Masse	9g

Table 3.3 – Spécifications du contrôleur électronique de vitesse (Lynxmotion 12A ESC SimonK)

Dans un quadricoptère, deux des hélices tournent dans le sens des aiguilles d'une montre et les deux autres tournent dans le sens inverse des aiguilles d'une montre pour équilibrer le couple produit par la rotation des hélices. Les branchements corrects des moteurs avec les contrôleurs de vitesse sont donnés sur la figure 3.6. Pour les moteurs tournant dans le sens horaire (*CW - Clockwise*), le branchement des câbles entre les deux composants est intuitif, la gauche va à la gauche, le milieu au milieu, et la droite à la droite. Par contre pour les moteurs tournant dans le sens anti-horaire (*CCW - Counterclockwise*), il faut veiller à inverser un peu les choses, le câble gauche de l'ESC doit être relié au câble droit du moteur. De même, le câble droit de l'ESC doit être relié au câble gauche du moteur. Le câble du milieu restant de l'ESC doit être connecter au câble du milieu du moteur.

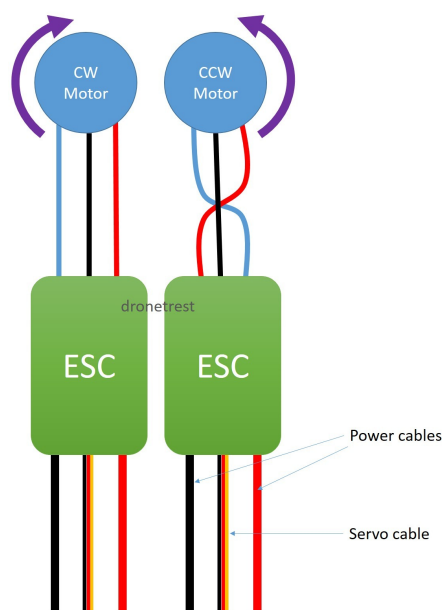


Figure 3.6 – Branchements des moteurs sans balais aux ESCs (<https://www.dronetrest.com/>)

### 3.1.5 Batterie

Les batteries sont majoritairement utilisées dans le quadricoptère comme source d'énergie pour l'alimentation du contrôleur de vol, les ESCs, les différents capteurs et les moteurs. La batterie représente un poids significatif du drone. Les critères de choix d'une batterie pour un quadricoptère sont associés à l'optimisation du temps de vol et des performances du drone (vitesse et poussée).

1. La tension de la batterie (en volts) : elle dépend du nombre de cellules qu'elle embarque. Chaque cellule dispose de 3.7 V pour les accumulateurs LiPo. Une batterie 3S déploie une tension nominale de 11.1V. Le choix de la batterie doit être compatible avec le contrôleur de vitesse et le moteur choisis (voir tableaux 3.2 et 3.3). Ici les batteries 2 à 4 cellules conviennent.

2. La capacité de la batterie : elle définit son autonomie. Exprimé en mAh, il s'agit de la quantité d'énergie pouvant être délivrée une fois la batterie complètement rechargée. Notre batterie a une capacité de 3800 mAh, c'est-à-dire qu'elle peut restituer 3,8 A pendant une heure.

3. Le taux de décharge : désigné par le nombre de « C », Il correspond à la capacité maximale de décharge en continu de la batterie sans risque. Notre batterie a un taux de décharge de 30C, c'est-à-dire qu'elle sera capable de débiter  $20 C \times 3,8 Ah = 76A$ .

4. L'énergie spécifique : le rapport de l'énergie embarquée sur le poids de la batterie, exprimé en Wh/kg.



Figure 3.7 – Batterie 3S LiPo 11.1 V 3800 mAh Conrad energy

### 3.1.6 Microcontrôleur - Arduino MKR WiFi 1010

Le microcontrôleur utilisé dans ce projet est une carte Arduino MKR WiFi 1010<sup>3</sup> basé sur le processeur Arm® Cortex®-M0 32-bit SAMD21. Nous avons choisi cet Arduino en raison de sa facilité d'utilisation, de son support matériel (différentes cartes d'extension pour le capteur inertielle et pour la commande de moteurs) et de sa compatibilité avec le support Matlab Simulink pour Arduino<sup>4</sup>.

3. La datasheet complète de cette carte peut être retrouvée sur le site officiel d'Arduino <https://docs.arduino.cc/hardware/mkr-wifi-1010>.

4. Les bibliothèques Simulink Support Package for Arduino Hardware et MATLAB Support Package for Arduino Hardware

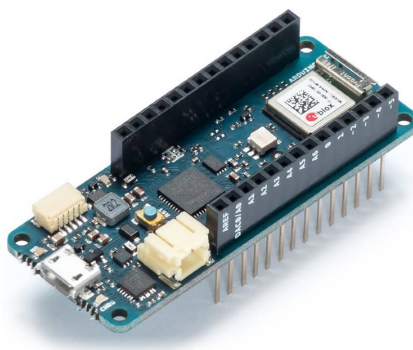


Figure 3.8 – Microcontrôleur - Arduino MKR WiFi 1010

### 3.1.7 Carte d'extension pour la commande des moteurs - Arduino MKR Motor Carrier

Afin de connecter les câbles de commande de différents moteurs BLDC, nous utilisons une carte d'extension du microcontrôleur Arduino MKR WiFi 1010, la carte Arduino MKR Motor Carrier<sup>5</sup>. L'Arduino MKR Motor Carrier est une carte d'extension pour contrôler les servomoteurs, les moteurs DC et les moteurs pas à pas. Il peut également être utilisé pour connecter d'autres actionneurs et capteurs via une série de connecteurs mâles à 3 broches.

Dans notre application, nous utilisons les entrées analogiques IN1, IN2, IN3 et IN4 (Compatibles avec le PWM) pour envoyer les signaux PWM de commande de moteurs BLDC en utilisant la librairie Arduino Servo<sup>6</sup>. Nous reviendrons plus en détails sur cette commande des moteurs lorsque nous parlerons de l'implémentation matérielle du contrôleur de vol sur le drone.

Le brochage pour la connexion des moteurs BLDC est donné sur le tableau 3.4 et indiqué sur la figure 3.9.

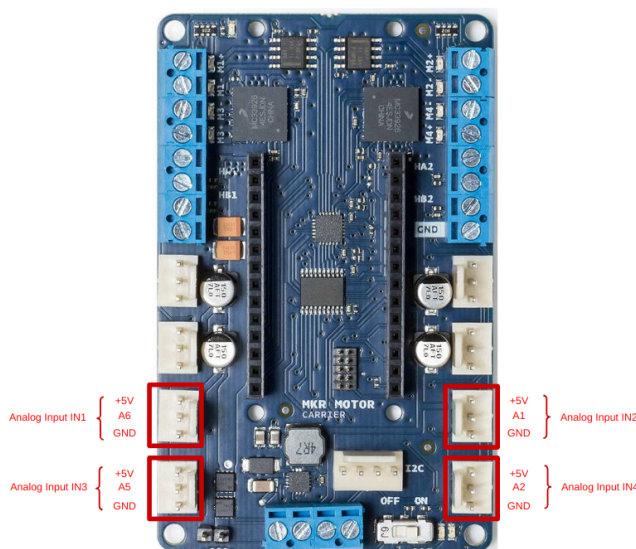


Figure 3.9 – Carte d'extension pour la commande des moteurs - Arduino MKR Motor Carrier

5. La datasheet complète de cette carte peut être retrouvée sur le site officiel d'Arduino <https://store.arduino.cc/products/arduino-mkr-motor-carrier>.

6. Simulink Standard Servo Write block - <https://fr.mathworks.com/help/supportpkg/arduino/ref/servo-control.html>.

Moteur	Désignation	Numero de broche Arduino
MOTOR1	IN1	A6 (21)
MOTOR2	IN2	A1 (16)
MOTOR3	IN3	A5 (20)
MOTOR4	IN4	A2 (17)

Table 3.4 – Branchement des ESCs à la carte d'extension pour la commande des moteurs - Arduino MKR Motor Carrier

### 3.1.8 Centrale inertielle - Arduino MKR IMU Shield (BNO055 IMU)

La carte d'extension Arduino MKR IMU Shield <sup>7</sup> est basée sur le capteur d'orientation absolue BNO055 de Bosch Sensortec GmbH qui intègre un accéléromètre triaxial de 14 bits, un gyroscope triaxial de 16 bits avec une plage de  $\pm 2000$  degrés par seconde et un capteur géomagnétique triaxial. Ce capteur embarque un microcontrôleur 32 bits exécutant le logiciel FusionLib BSX3.0 <sup>8</sup>. Il peut être utilisé en mode fusion et non-fusion pour fournir des données brutes ou déjà fusionnées avec l'algorithme tournant sur son microcontrôleur embarqué.

Une centrale inertielle est un appareil électronique qui estime l'orientation d'un corps à partir de la mesure des accélérations tridimensionnelles et des vitesses angulaires. BNO055 est un capteur d'orientation absolue à 9 axes avec fusion données, il mesure les accélérations linéaires, les vitesses angulaires et le champ magnétique.

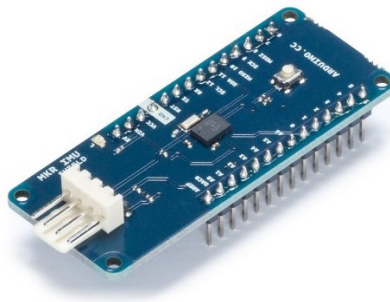


Figure 3.10 – Centrale inertielle - Arduino MKR IMU Shield (BNO055 IMU)

7. Une documentation plus détaillée sur cette carte et la datasheet complète du capteur BNO055 peuvent être retrouvées sur le site officiel d'Arduino <https://store.arduino.cc/products/arduino-mkr-imu-shield>.

8. Logiciel FusionLib BSX3.0 - <https://www.bosch-sensortec.com/software-tools/software/>

### 3.2 Banc d'essai à 4 degrés de liberté pour quadricoptère

Un banc de test a été mis en place afin de tester et de valider le fonctionnement du quadricoptère (figure 3.12). Ce banc de test permet de tester quatre degrés de liberté du mouvement du drone. Il comprend un pivot glissant permettant de tester le mouvement suivant l'axe z (le déplacement vertical) et une rotule pour les mouvements de roulis, de tangage et de lacet.

Le schéma cinématique de ce banc de test est donné sur la figure 3.11. Le repère 0 est le bâti, le repère 1 le coulisseau et le repère 2 la pièce de fixation du drone. Les dessins de définition des différentes pièces de ce banc d'essai ainsi qu'un dessin d'ensemble sont fournis en annexe de ce rapport.

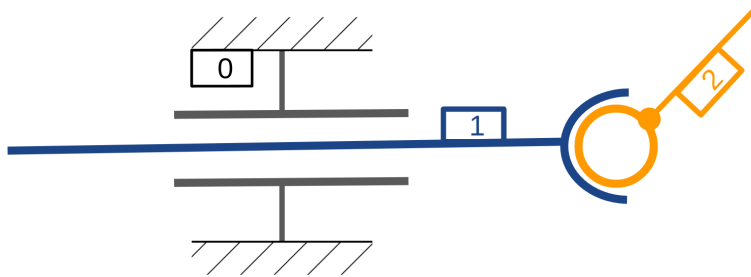


Figure 3.11 – Schéma cinématique du banc d'essai à 4 degrés de liberté



Figure 3.12 – Banc d'essai à 4 degrés de liberté pour quadricoptère

## Chapitre 4

# Modélisation et contrôle avec Matlab Simulink

### 4.1 Modélisation Simulink du quadricoptère

Nous allons maintenant construire le modèle physique de notre quadricoptère avec son contrôleur pour ce faire, nous utilisons le schéma-bloc donné sur la figure 4.1. Ce modèle pour la simulation est construit à partir des équations non linéaires de la dynamique du véhicule que nous avons déterminées au paragraphe 2.3.

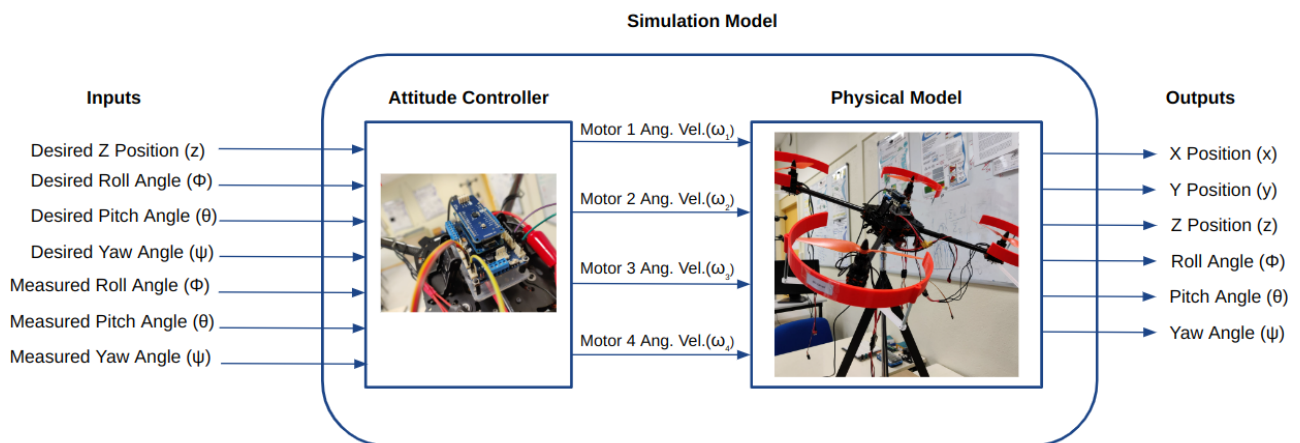


Figure 4.1 – Modèle de simulation du quadricoptère

Les paramètres sont des constantes qui affectent les états, comme le coefficient de traînée et de poussée. Les sorties de l'installation sont basées sur les états précédents, les paramètres du modèle et les entrées.

Les valeurs des paramètres du modèle de notre quadricoptère sont données sur le tableau 4.1. Les différents calculs afin d'identifier ces paramètres sont détaillés sur l'Annexe A.

Paramètre	Désignation	Valeur	Unité
m	Masse	2.0878	kg
g	Gravité (accél. de la pesanteur)	9.81	m/s <sup>2</sup>
l	Distance CG - centre d'un rotor	0.32	m
$I_{xx}$	Moment d'inertie autour de l'axe x	0.07989	kg.m <sup>2</sup>
$I_{yy}$	Moment d'inertie autour de l'axe y	0.07989	kg.m <sup>2</sup>
$I_{zz}$	Moment d'inertie autour de l'axe z	0.15594	kg.m <sup>2</sup>
$I_r$	Moment d'inertie de l'hélice /axe z	0.04439	kg.m <sup>2</sup>
b	Coefficient de poussée	$3.59 \cdot 10^{-5}$	kg.m
d	Coefficient de traînée	$2.0810 \cdot 10^{-6}$	kg.m
$A_x$	Résistance de l'air suivant l'axe x	0.1	kg/s
$A_y$	Résistance de l'air suivant l'axe y	0.1	kg/s
$A_z$	Résistance de l'air suivant l'axe z	0.1	kg/s

Table 4.1 – Paramètres du modèle du quadricoptère

### 4.1.1 Le modèle physique

Le sous-système pour la modélisation de la physique du drone est créé et nommé "Plant Model Subsystem". Ce sous-système a 4 entrées ( $\omega_1, \omega_2, \omega_3, \omega_4$ ) qui correspondent aux vitesses de rotation des rotors et 6 sorties (x, y, z,  $\phi, \theta, \psi$ ), la position et l'orientation du véhicule (voir figure 4.2).

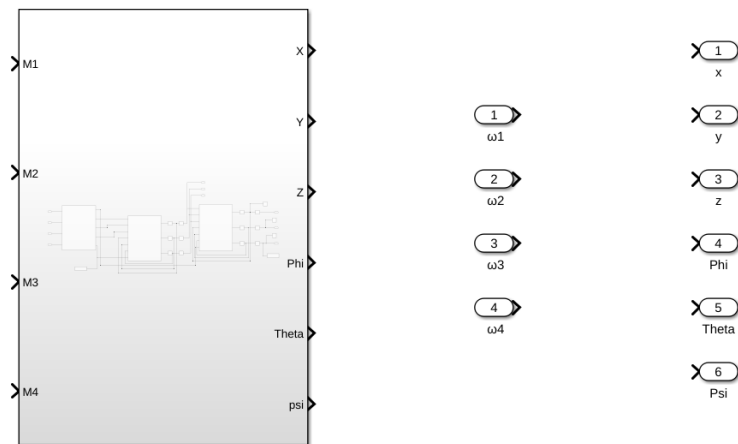


Figure 4.2 – Le modèle physique du quadricoptère avec ses entrées et ses sorties

Ce sous-système englobe plusieurs sous-systèmes pour représenter les différentes dynamiques du modèle notamment le sous-système "Orientation and Motors" pour décrire l'orientation du véhicule en fonction des vitesses de rotation des moteurs, le sous-système "Angular Acceleration" pour les équations décrivant les accélérations angulaires du véhicule en fonction de la poussée ou des couples demandés en entrée et enfin le sous-système "Linear Acceleration" pour les équations des accélérations linéaires du véhicule. L'objectif est de créer un modèle composé de toutes les équations dynamiques du véhicule (figure 4.3).

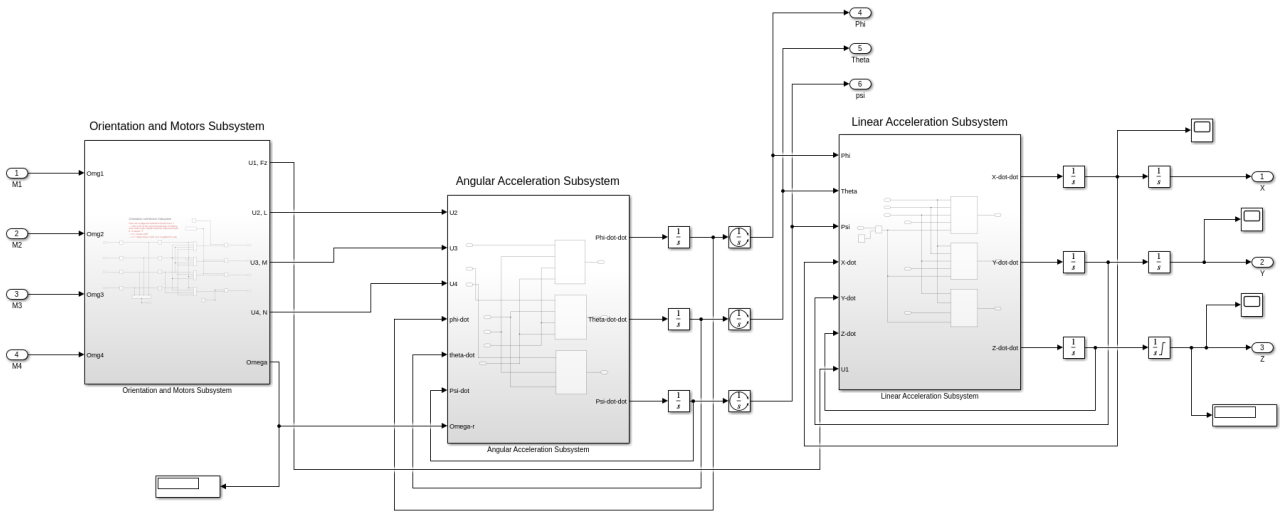


Figure 4.3 – Les différents sous-systèmes du modèle physique

**Le sous-système - Orientation du véhicule**

Le sous-système “Orientation and Motors” permet de décrire l’orientation du véhicule en fonction des vitesses de rotation des moteurs. Il est construit à partir de l’équation de la poussée (2.21) et des équations de moments (2.22, 2.23 et 2.28). Nous avons dérivé les équations pour l’orientation X de notre véhicule.

Ce sous-système prend les commandes des moteurs et les convertit en force de poussée et moments pour contrôler l’orientation et l’altitude du véhicule (voir figures 4.4 et 4.5).

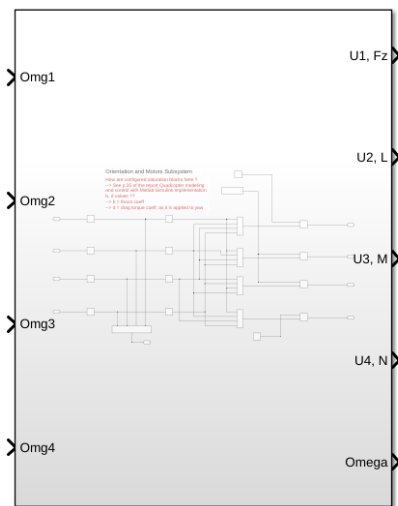


Figure 4.4 – Le sous-système - Orientation du véhicule

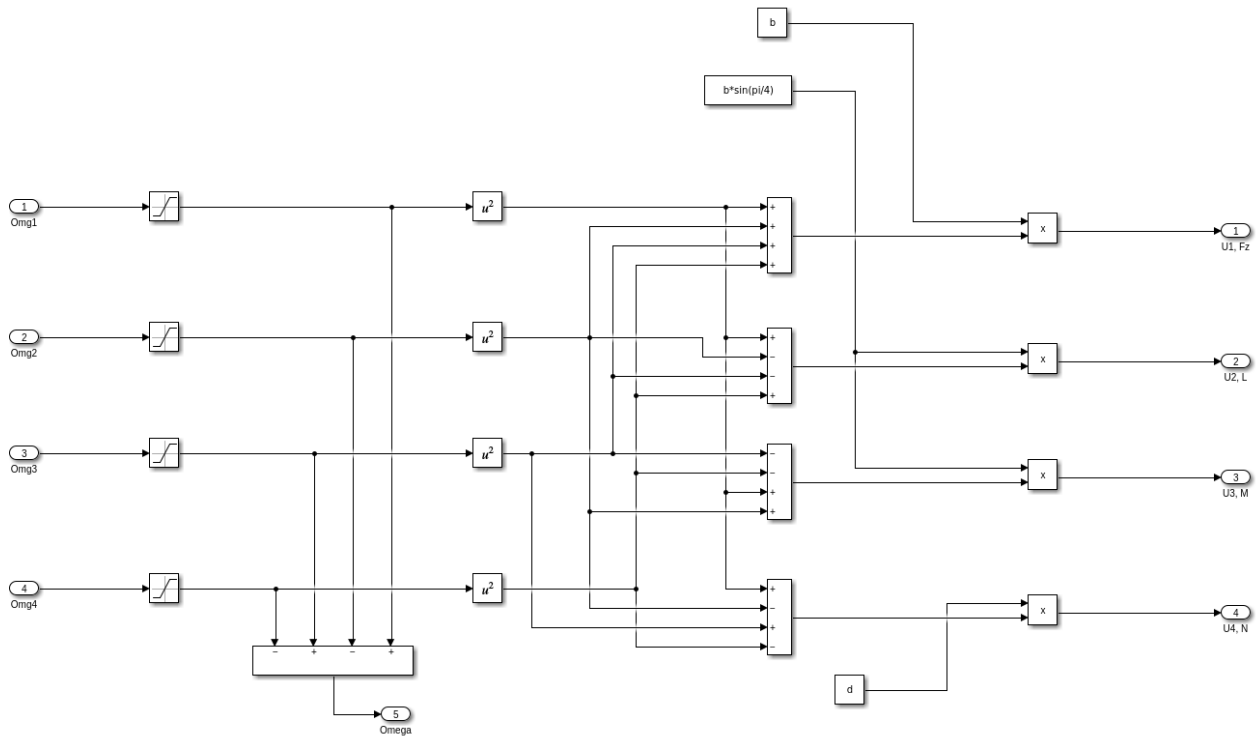


Figure 4.5 – Les équations donnant la poussée et les moments en fonction des vitesses de rotation de rotors

### Le sous-système des accélérations angulaires

Ce sous-système (figure 4.6) est basé sur les équations 2.46, 2.47 et 2.48 dans lesquelles la poussée et les moments générés par les moteurs sont convertis en accélérations angulaires.

Le sous-système des accélérations angulaires comporte trois autres sous-systèmes pour décrire séparément les accélérations de roulis, de tangage et de lacet (Figure 4.7).

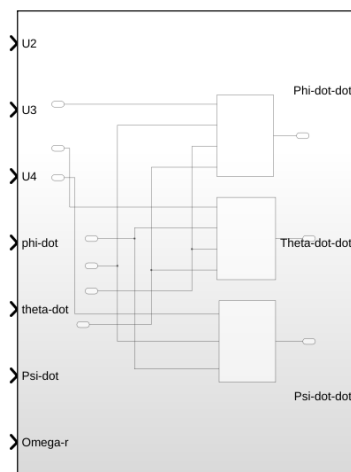


Figure 4.6 – Le sous-système des accélérations angulaires

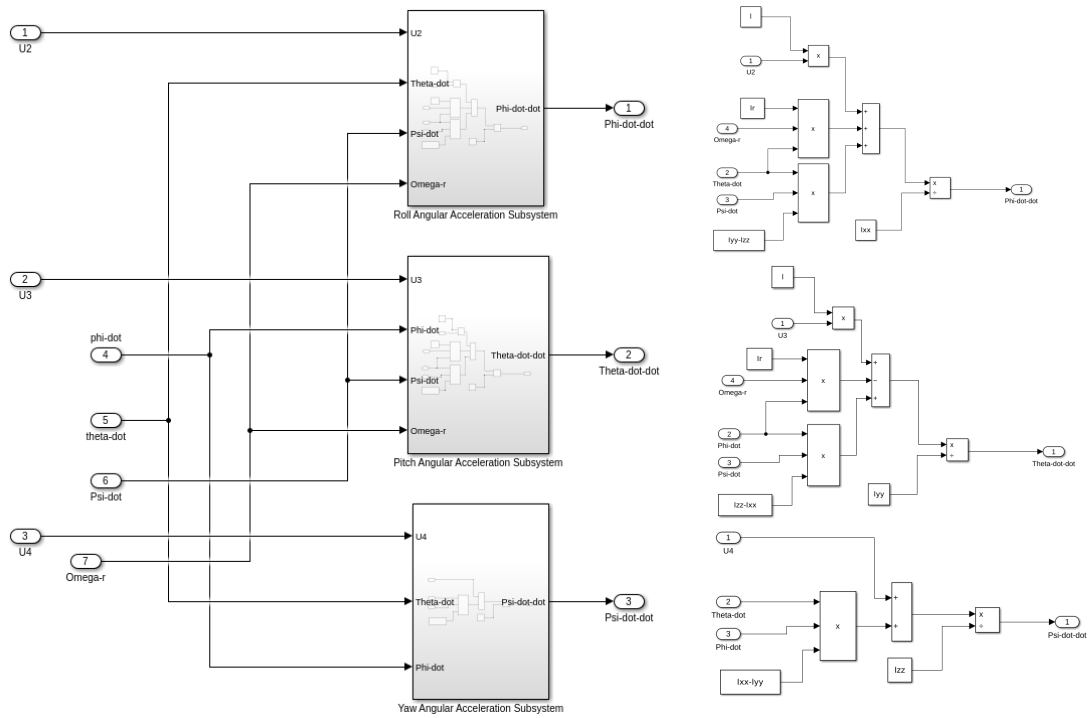


Figure 4.7 – Les équations des accélérations angulaires du quadricoptère

### Le sous-système des accélérations linéaires

Le sous-système des accélérations linéaires (figure 4.8) est basé sur les équations du mouvement en translation 2.34 et 2.35.

Le sous-système des accélérations linéaires comporte également trois sous-systèmes pour la détermination des accélérations suivant les axes x, y et z séparément (Figure 4.9).

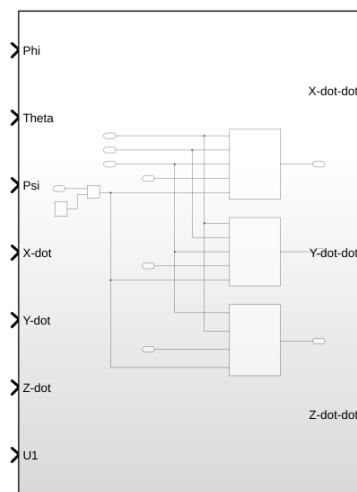


Figure 4.8 – Le sous-système des accélérations linéaires

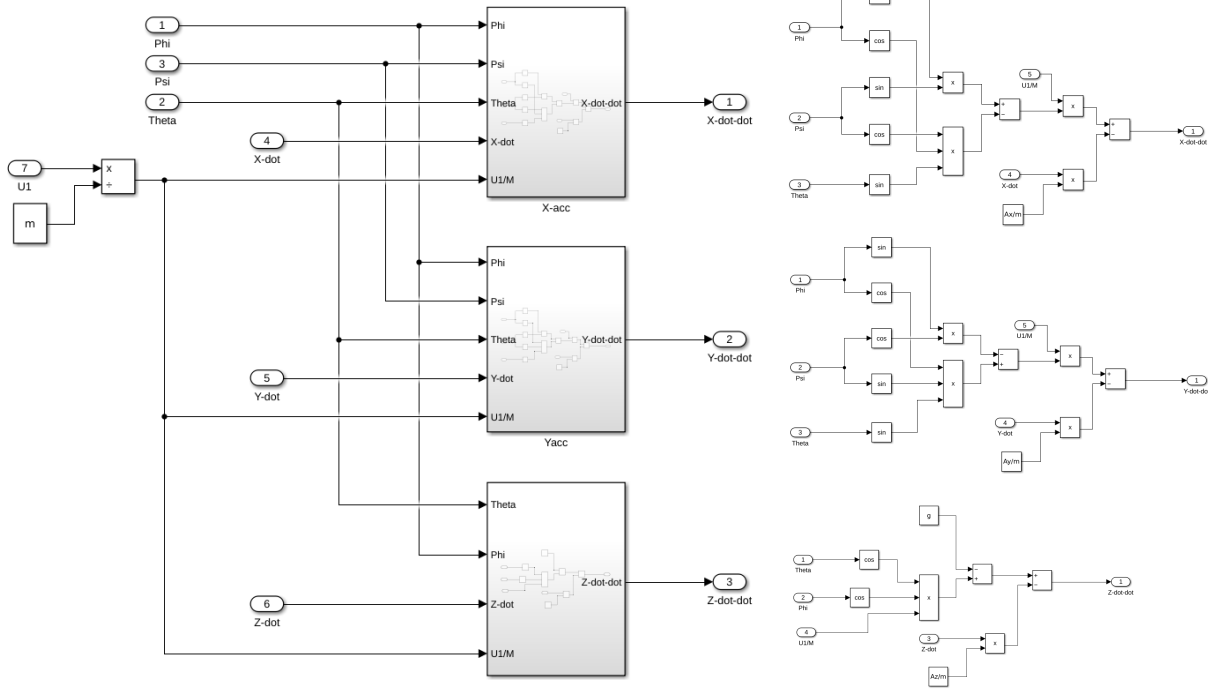


Figure 4.9 – Les équations des accélérations linéaires du quadricoptère

### Limitations physiques

Nous disposons maintenant d'un modèle complet décrivant la dynamique de notre véhicule. Il est important de remarquer que ce modèle ne représente pas parfaitement notre quadricoptère. Cette simulation ne prend pas en compte les limitations du monde réel, nous introduisons pour cela certaines contraintes dans notre modèle.

La hauteur  $h$  est limitée en introduisant zéro comme limite inférieure dans l'intégrateur qui convertit la vitesse  $z$  en position  $z$  car le véhicule ne peut pas descendre sous la surface de la terre. Les vitesses des moteurs sont limitées par des saturateurs qui éliminent les signaux négatifs pour éviter une rotation dans la direction opposée à celle choisie (voir paragraphe 2.1). Dans certains quadricoptères, les rotors peuvent tourner dans les deux sens pour effectuer des manœuvres de saltos ou voler la tête en bas, ce qui n'est pas le cas dans notre application. Les angles de roulis, de tangage et de lacet sont saturés entre  $\pi$  et  $-\pi$ . (Ferry, 2017)

### 4.1.2 Vérification du modèle du quadricoptère

Nous pouvons vérifier le modèle de notre quadricoptère en appliquant les entrées de vitesses des moteurs au modèle physique. Nous devons obtenir un résultat similaire à la figure 4.10.

a) Vérification de la prise d'altitude : en faisant tourner tous les moteurs à la même vitesse, le véhicule prend de la hauteur, on peut observer que l'altitude change alors que les angles de roulis, de tangage et de lacet restent les mêmes (la réponse du véhicule est illustrée à la figure 4.11)

b) Vérification du mouvement de roulis et de tangage : la modification simultanée des vitesses des moteurs M1 et M4 ou M2 et M3 entraîne une modification de l'angle de roulis. De même, la modification simultanée des vitesses M1 et M2 ou M3 et M4 entraîne une modification de l'angle de tangage. Pour la vérification, nous mettons les moteurs M1 et M4 à la même vitesse, ce qui entraîne une modification de l'angle de roulis  $\phi$  (Figure 4.12).

c) Vérification du mouvement de lacet : la modification de la vitesse des moteurs M1 et M3 affecte l'angle de lacet  $\psi$ . Ces différentes vérifications permettent de valider le bon fonctionnement du modèle. (Figure 4.13).

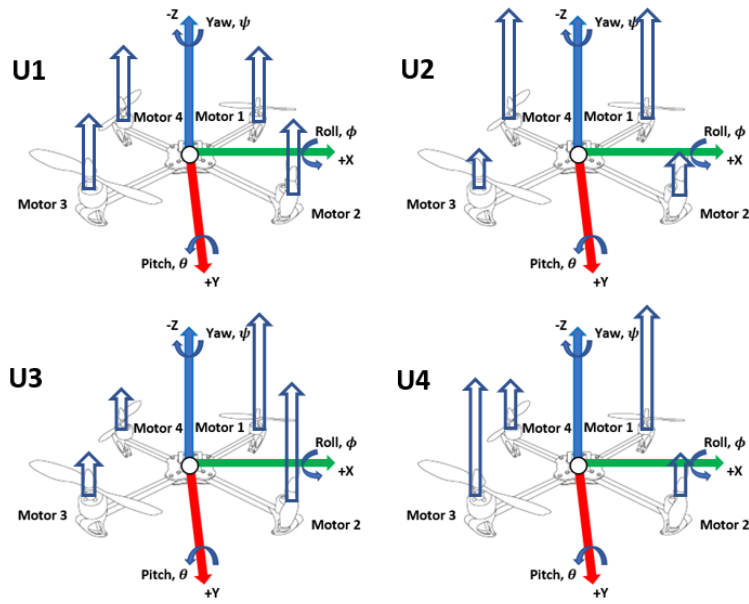


Figure 4.10 – Vérification du modèle du quadricoptère comportements souhaités (Ferry, 2017)

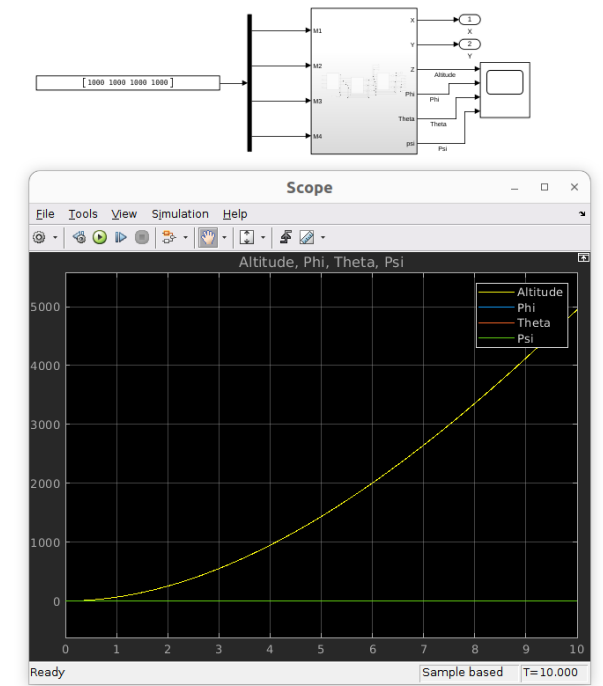


Figure 4.11 – Vérification de la prise d'altitude

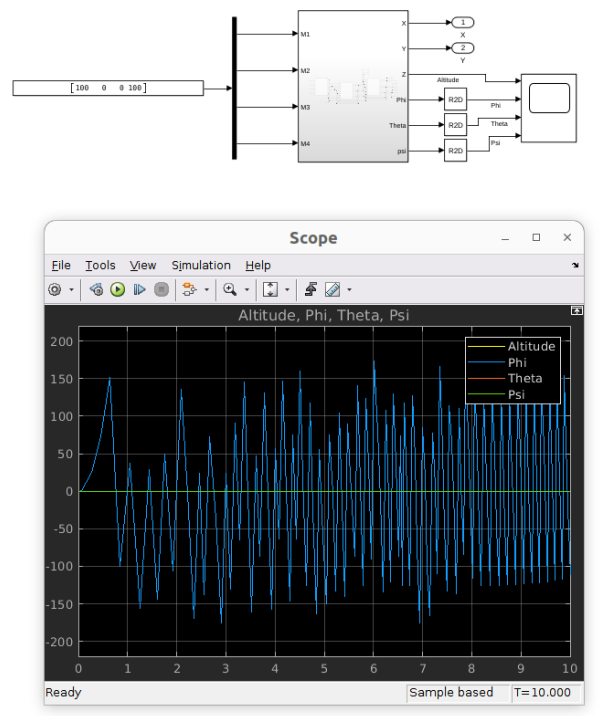


Figure 4.12 – Vérification du mouvement de roulis

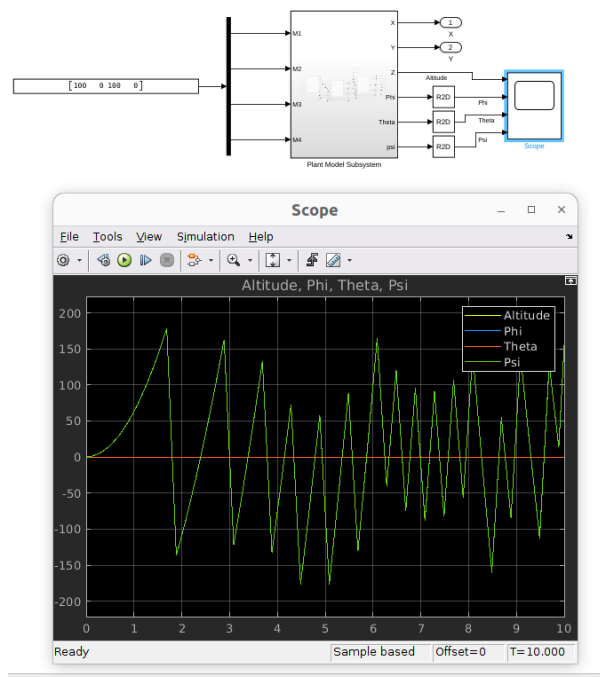


Figure 4.13 – Vérification du mouvement de lacet

## 4.2 Contrôleur PID

Le contrôleur est une partie très importante d'un système dynamique, sans l'ajout du contrôleur, le système que nous avons modélisé précédemment serait intrinsèquement instable. Il existe de nombreuses stratégies de contrôle notamment des stratégies s'appuyant sur la représentation d'états (placement de pôles, LQR - linear quadratic regulator, l'observation de certains états du système, etc.). Pour notre application et pour de raisons de simplicité, nous utilisons un contrôleur PID (figure 4.14).

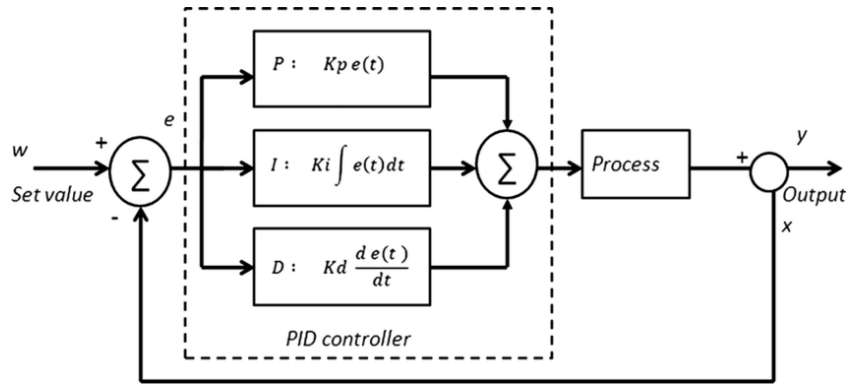


Figure 4.14 – Schéma-bloc du contrôleur PID

Le contrôleur PID (Proportionnel, Intégral et Dérivé) possède trois termes de contrôle. Il prend en entrée un signal d'erreur (la différence entre la sortie/réponse mesurée et la consigne souhaitée) qui est multiplié par les gains du PID pour obtenir le signal de contrôle du système ou du processus. Le terme Proportionnel multiplie simplement le signal d'erreur par une constante. Le terme Intégral permet de réduire voire de supprimer l'erreur statique en régime permanent en intégrant la valeur de l'erreur dans le temps jusqu'à ce qu'elle atteigne zéro. Le terme Dérivé estime l'erreur future par le taux de variation de l'erreur dans le temps et la multiplie par une constante afin de la réduire. (Pham, 2018)

### 4.2.1 Le contrôleur d'attitude

Les aspects de la dynamique du quadricoptère que nous souhaitons contrôlés sont l'altitude, l'angle de roulis, l'angle de tangage et l'angle de lacet. Le retour d'information provenant du modèle physique du drone qui a été modélisé dans la section précédente est utilisé pour fermer la boucle de rétroaction. Le système de contrôle en boucle fermée est donnée sur la figure 4.15.

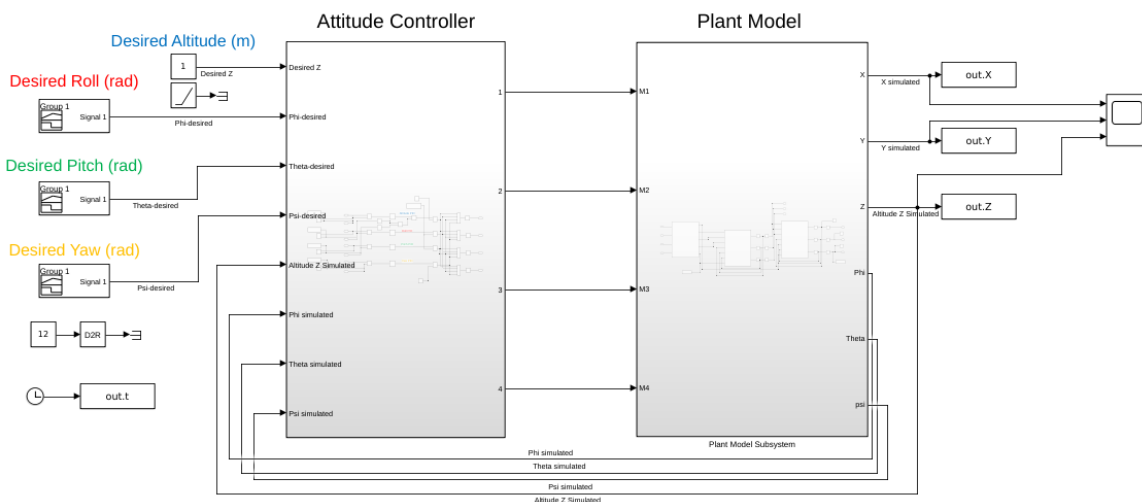


Figure 4.15 – Le système de contrôle en boucle fermée

Dans de nombreux cas, l'utilisateur contrôle plutôt l'accélérateur du quadricoptère plutôt que l'altitude, nous utiliserons cette commande du pédale d'accélérateur lorsque nous implémenterons le contrôleur sur le drone réel vu que nous ne disposons pas d'un capteur d'altitude. Cependant, il est assez difficile de maintenir une altitude constante en simulation sans contrôleur, c'est pourquoi nous avons ajouté un contrôleur pour la commande de l'altitude  $z$  du drone.

Au total, quatre contrôleurs PID ont été construits pour la commande de l'altitude, du roulis, du tangage et du lacet. Les sorties des contrôleurs PID sont transformées en commandes de moteurs pour la configuration en  $x$  du quadricoptère. Le modèle Simulink du contrôleur d'attitude est donné sur la figure 4.16.

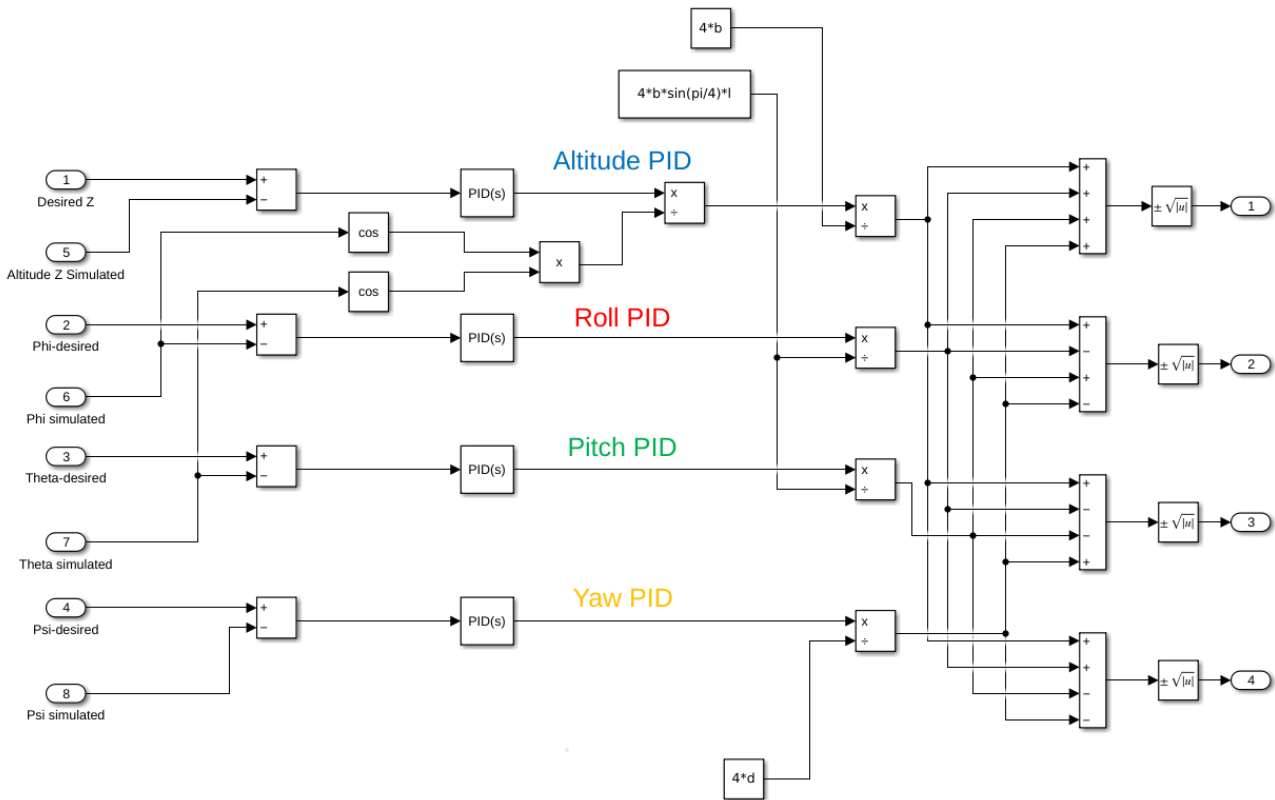


Figure 4.16 – Le contrôleur d'Attitude pour la simulation

### 4.2.2 Réglage du contrôleur

Les contrôleurs PID peuvent être réglés manuellement ou en utilisant la fonction de réglage automatique dans Simulink. Nous choisissons de régler manuellement chacun des PID du contrôleur d'attitude.

Il existe plusieurs méthodes de réglage des contrôleurs PID notamment des méthodes heuristique (Ziegler-Nichols), optimale (optimisation d'une fonction de coût dépendant de la variable de commande) et manuelle. Pour le réglage manuel d'un contrôleur PID, une procédure itérative simple peut être suivie.

Le processus consiste à régler d'abord le gain  $K_p$ , alors que les gains  $K_i$  et  $K_d$  sont nuls. Par essais/erreurs, il faut régler ce coefficient afin d'améliorer le temps de réponse du système tout en faisant attention à ce que le système n'oscille pas beaucoup. Généralement on augmente la valeur de  $K_p$  jusqu'à l'apparition des premières oscillations. Cette valeur du gain  $K_p$  indique simplement la plage des valeurs admissibles du gain  $P$ , qui peut être ajustée lorsque l'on a besoin.

On règle ensuite le gain  $K_d$  en augmentant progressivement sa valeur, ceci a pour effet de diminuer les oscillations dans la simulation et donc de lisser la réponse du système.

On règle enfin le gain  $K_i$  pour supprimer (ou réduire) l'erreur statique en régime permanent.

### Réglage du PID pour l'altitude

Le réglage du PID pour une consigne échelon n'est pas très représentatif de la situation réelle. Dans le monde réel le pilote décolle lentement et l'arrivée à une altitude n'est pas instantanée. Pour imiter ce comportement dans la simulation, nous utilisons une consigne en rampe qui part de l'altitude 0 et augmente avec une pente de 1.5. (Usman, 2020)

Après plusieurs essais/erreurs, le résultat du réglage du PID pour une rampe est donnée sur la figure 4.17. On peut observer qu'au début, pendant environ trois secondes, le quadricoptère ne décolle pas car le signal de commande est insuffisant. Après trois secondes, il décolle très doucement et se stabilise en fonction du signal de commande d'altitude qui augmente continuellement.

Les valeurs de gains du PID pour le contrôle de l'altitude sont données sur le tableau 4.2.

$K_p$	7.4
$K_i$	5.5
$K_d$	4.4

Table 4.2 – Les valeurs des gains du PID pour le contrôle de l'altitude

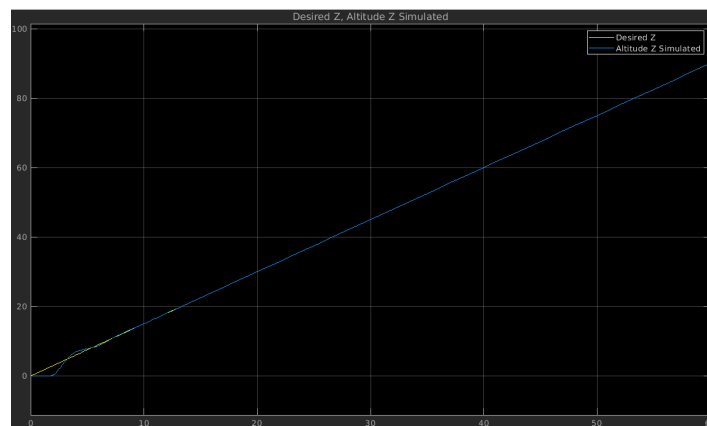


Figure 4.17 – Réponse du système à une rampe - réglage du PID pour le contrôle d'altitude

### Réglage du PID pour le roulis et le tangage

Les contrôleurs PID pour les angles de roulis et de tangage ont les mêmes valeurs de gain, car le roulis et le tangage fonctionnent sur le même principe (voir paragraphe 4.1.2 sur la validation du modèle). La figure 4.18 présente la réponse du système lorsque l'angle souhaité est de 12 degrés (angle échelon de 12 degrés en roulis) et que le drone tente d'atteindre simultanément l'altitude de 1m.

Les valeurs de gains du PID pour le contrôle de l'angle de roulis et de tangage sont données sur le tableau 4.3.

$K_p$	0.055
$K_i$	0.00002
$K_d$	0.11

Table 4.3 – Les valeurs des gains du PID pour le contrôle de l'angle de roulis et de tangage

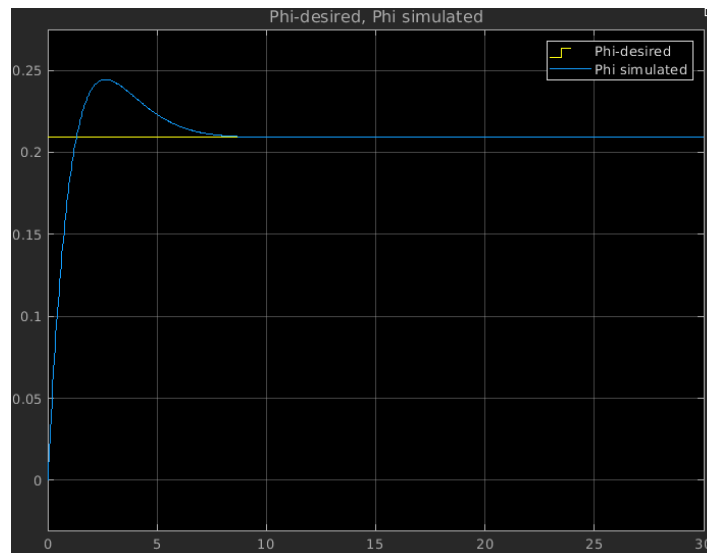


Figure 4.18 – Réponse du système à un angle échelon de 12 degrés en roulis - réglage du PID pour le contrôle de l'angle de roulis

### Réglage du PID pour le lacet

Les valeurs de gains du PID pour le contrôle de l'angle de lacet sont données sur le tableau 4.4.

La figure 4.19 présente la réponse du système lorsque l'angle souhaité est de 12 degrés (angle échelon de 12 degrés en lacet) et que le drone tente d'atteindre simultanément l'altitude de 1m.

$K_p$	0.1
$K_i$	0.0001
$K_d$	0.2

Table 4.4 – Les valeurs des gains du PID pour le contrôle de l'angle de lacet

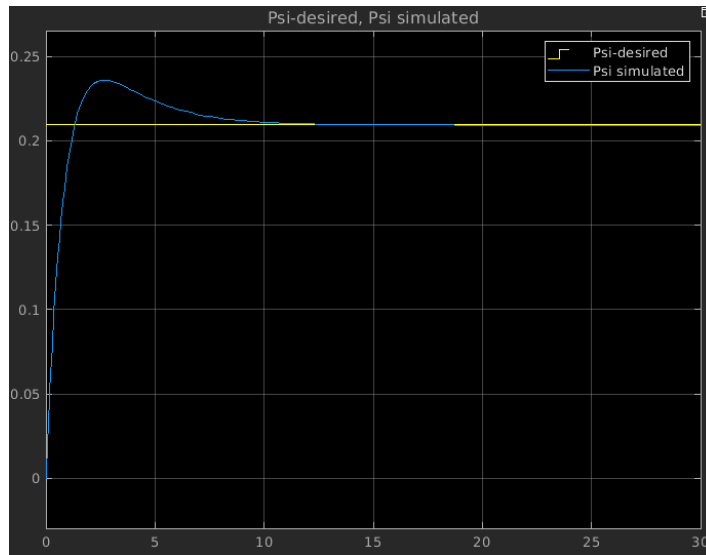


Figure 4.19 – Réponse du système à un angle échelon de 12 degrés en lacet - réglage du PID pour le contrôle de l'angle de lacet

### 4.2.3 Suivi de trajectoire

Une trajectoire est planifiée à l'aide des blocs de construction de signaux dans Simulink, afin d'évaluer les performances du véhicule lorsqu'on lui applique simultanément plusieurs signaux de consigne.

Les différents signaux de consigne pour les angles de roulis, de tangage et de lacet ainsi que les réponses du système à ces consignes sont données sur les figures 4.20, 4.21 et 4.22. La consigne d'altitude est un signal de rampe avec une pente de 1,5. Nous exécutons la simulation pendant 60 secondes.

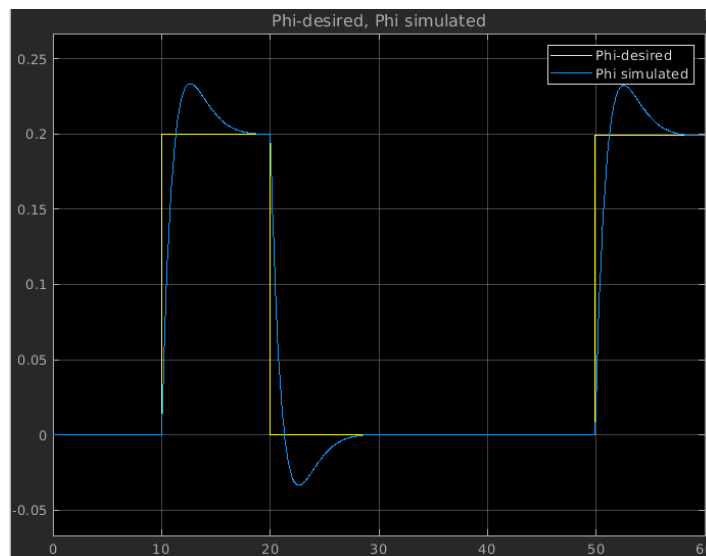


Figure 4.20 – Signal de consigne et réponse du système pour l'angle de roulis

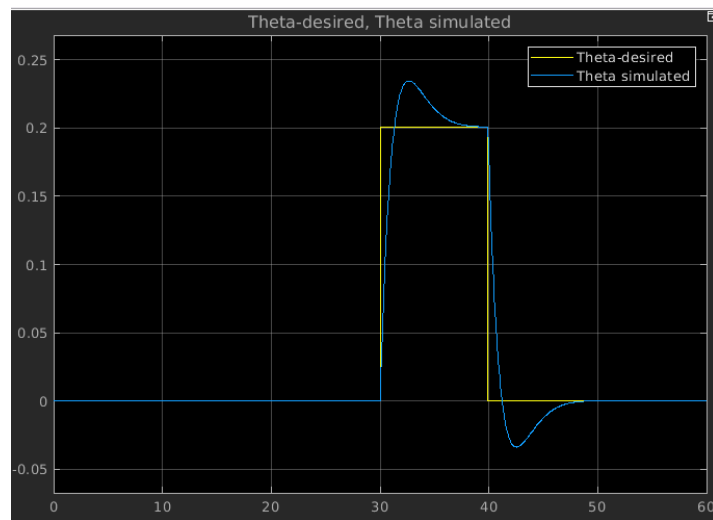


Figure 4.21 – Signal de consigne et réponse du système pour l'angle de tangage

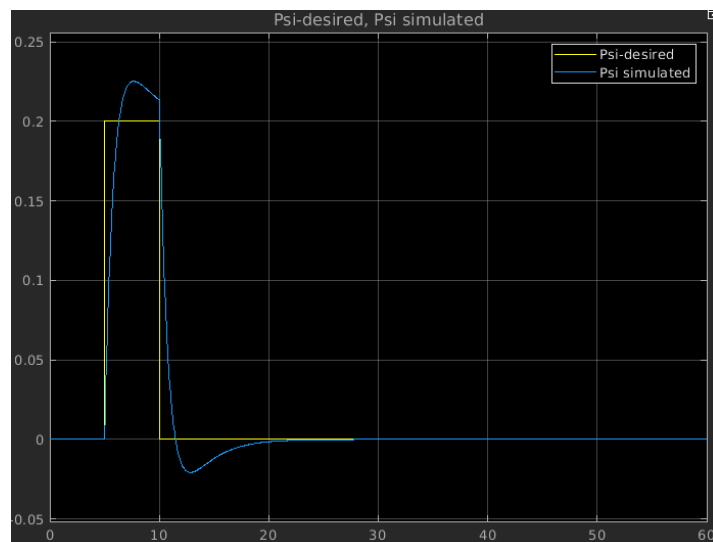


Figure 4.22 – Signal de consigne et réponse du système pour l'angle de lacet

Cette trajectoire est aléatoire dans le repère inertiel car nous donnons les consignes en terme d'angles d'orientation dans le système de coordonnées du corps (les angles de roulis, de tangage et de lacet sont contrôlés dans le repère du corps) au lieu d'utiliser un planificateur de trajectoire qui convertirait la trajectoire en coordonnées inertielles  $X(t)$ ,  $Y(t)$  et  $Z(t)$  en coordonnées du corps  $\phi(t)$ ,  $\theta(t)$  et  $\psi(t)$ .

On peut observer que le système répond bien aux différentes consignes ce qui nous permet de valider le réglage de notre contrôleur. La commande des mouvements de roulis, de tangage et de lacet ainsi que la prise d'altitude semblent être complètement découplées, les différentes réponses aux consignes se superposent (voir figure 4.23). La trajectoire inertielle suivi par le véhicule est donnée sur la figure 4.24.

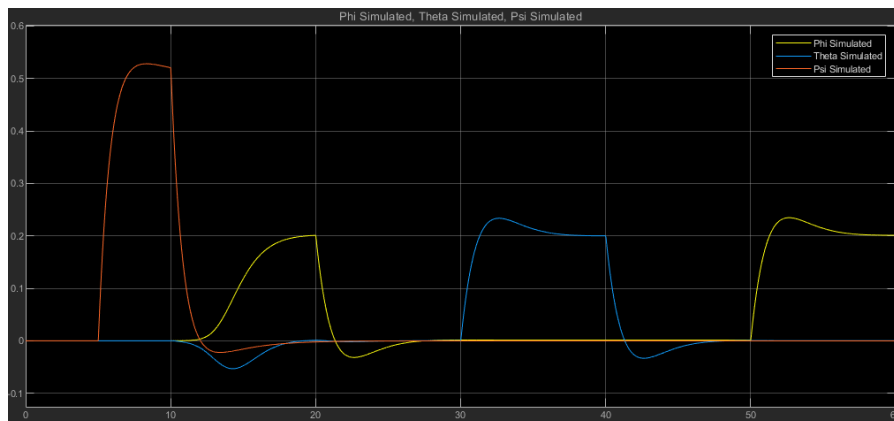


Figure 4.23 – La réponse du système aux trois signaux de consigne d'angle d'orientation

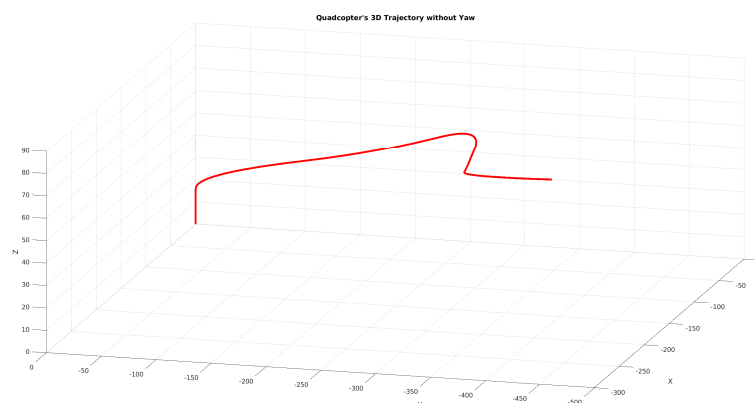


Figure 4.24 – Trajectoire du véhicule en coordonnées inertielles

## 4.3 Implémentation

### 4.3.1 Modèle Simulink pour l'implémentation sur le drone réel

L'ensemble du modèle de simulation construit précédemment ne peut pas être téléversé directement sur le matériel car ce modèle s'appuie sur les équations du dynamique du véhicule pour imiter le fonctionnement du drone dans la simulation.

Afin de développer un contrôleur pour l'implémentation matérielle, seul le sous-système "contrôleur d'attitude" du modèle Simulink est gardé pour cette section et sera modifié à cet effet. Les blocs (sous-systèmes) pour l'acquisition de données inertielles et la commande des moteurs du quadricoptère sont ajoutés. Le Modèle Simulink pour l'implémentation sur le drone est donnée sur la figure 4.25.

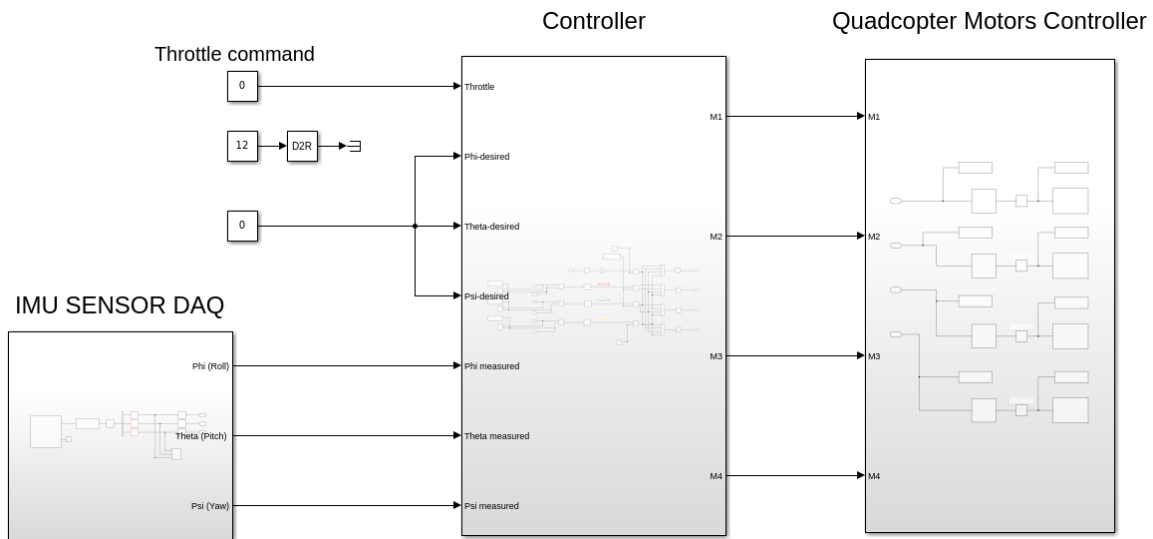


Figure 4.25 – Modèle Simulink pour l'implémentation sur le drone

#### Le sous-système pour l'acquisition de données inertielles

La carte d'extension Arduino MKR IMU Shield<sup>1</sup> est basée sur le capteur d'orientation absolue BNO055 de Bosch Sensortec GmbH, il s'agit d'un capteur d'orientation absolue à 9 axes avec fusion de données, il mesure les accélérations linéaires, les vitesses angulaires et le champ magnétique.

Le bloc Simulink BNO055 Arduino (Figure 4.26) lit les données du capteur IMU BNO055 attaché au matériel. Ce capteur possède deux modes de fonctionnement. En mode fusion, ce bloc fournit des valeurs calibrées et fusionnées des accélérations linéaires, les angles d'Euler et des quaternions, tandis qu'en mode non-fusion, il fournit les données brutes (MATLAB 2022b). Nous avons utilisé les données fusionnées de ce capteur et n'avons pas eu le temps pour entreprendre nous-même cette fusion des données d'orientation inertielles.

1. Une documentation plus détaillée sur cette carte et la datasheet complète du capteur BNO055 peuvent être retrouvées sur le site officiel d'Arduino <https://store.arduino.cc/products/arduino-mkr-imu-shield>.

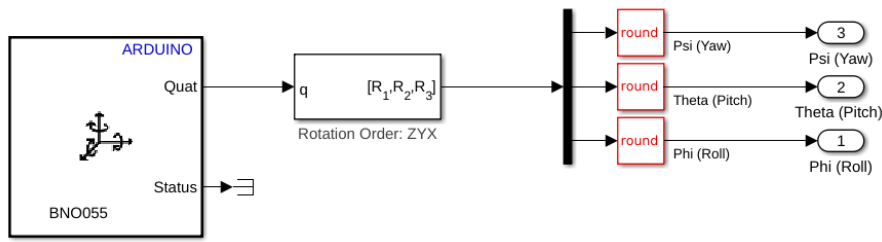


Figure 4.26 – Le sous-système IMU SENSOR DAQ pour l’acquisition de données du capteur inertiel

**Le contrôleur**

Afin de construire le contrôleur d’attitude pour l’implémentation sur le drone, nous effectuons quelques modifications par rapport au contrôleur modélisé au paragraphe 4.2.1.

Nous commençons par enlever le contrôleur PID pour le contrôle d’altitude, nous ne disposons pas en effet d’un capteur pour la mesure de la hauteur atteinte par le drone. Nous introduisons à la place d’une commande PID d’altitude, une commande du pédale d’accélérateur, cette commande agit directement sur la force de poussée qui s’applique sur le drone et permet au véhicule de prendre l’altitude.

Les trois contrôleurs PID restants pour le contrôle des angles de roulis, de tangage et de lacet doivent être discrétisés, nous utilisons dans cette application le bloc Simulink Régulateur PID discret (Discrete PID Controller block) . On aurait pu développer nous même notre contrôleur PID discret en utilisant des bloqueurs d’ordre 0 (Zero Order Hold) pour les signaux mesurés avec l’ajout d’un système anti débordement pour l’action intégrale (Integral anti-reset windup system).

Les sorties des contrôleurs PID sont transformées en commandes de moteurs pour la configuration en x du quadricoptère. Le modèle Simulink du contrôleur d’attitude est donné sur la figure 4.27.

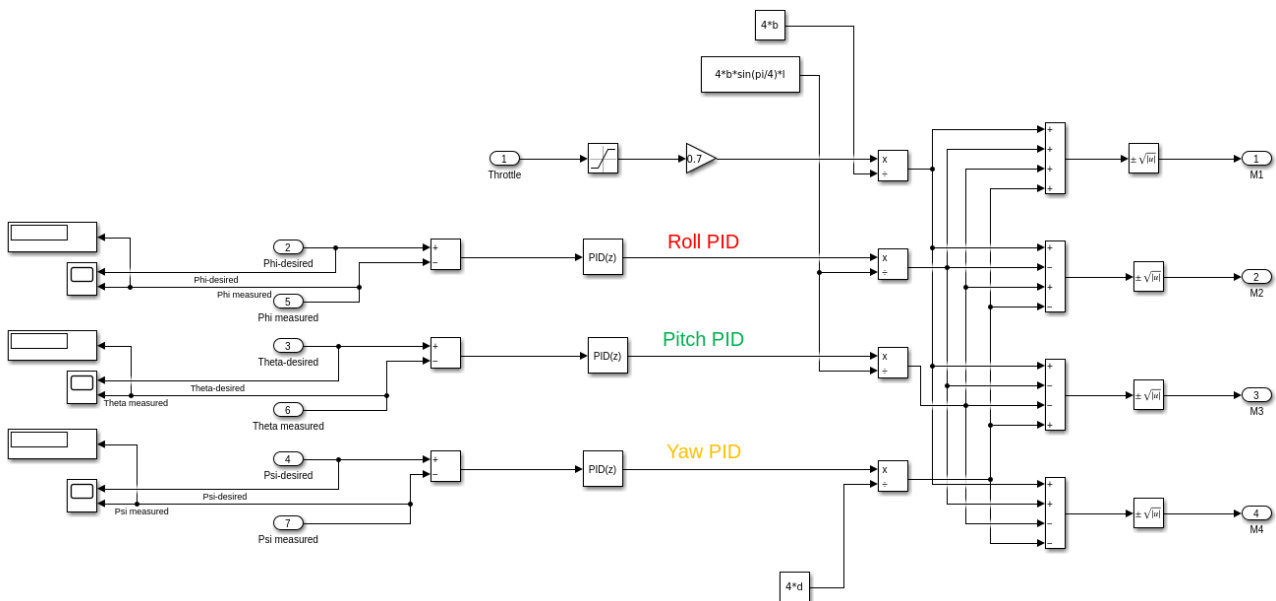


Figure 4.27 – Le contrôleur d’Attitude pour l’implémentation sur le drone

### Le sous-système pour la commande des moteurs

Les moteurs Brushless des rotors sont commandés grâce aux contrôleurs électroniques de vitesses - ESCs. Les ESCs sont pilotés à partir du microcontrôleur par des signaux PWM (modulation de largeur d'impulsion).

Nous utilisons les entrées analogiques IN1, IN2, IN3 et IN4 de l'Arduino MKR Motor Carrier (Compatibles avec le PWM) pour envoyer les signaux PWM de commande de moteurs BLDC en utilisant le bloc Simulink Standard Servo Write. La description du branchement des ESCs sur la carte d'extension des moteurs est décrite au paragraphe 3.1.7. Ce bloc simulink accepte les valeurs d'entrée de 0 à 180.

Les signaux de commande en sorties de PID sont mis en forme pour pouvoir convenir aux valeurs d'entrée du bloc Simulink Standard Servo Write. Nous avons donc mappé ces valeurs de sortie de PID à des valeurs variant entre 35 et 178 (plage déterminée expérimentalement, en dessous de 35 les moteurs ne tournent pas, la tension est insuffisante et la valeur maximale est plafonné à 178 pour des raisons de sécurité) en utilisant des tables de conversion 1-D et nous avons ajouté des blocs de saturation pour se rassurer de rester dans cette plage des valeurs. Le sous-système pour la commande des moteurs est présenté sur la figure 4.28.

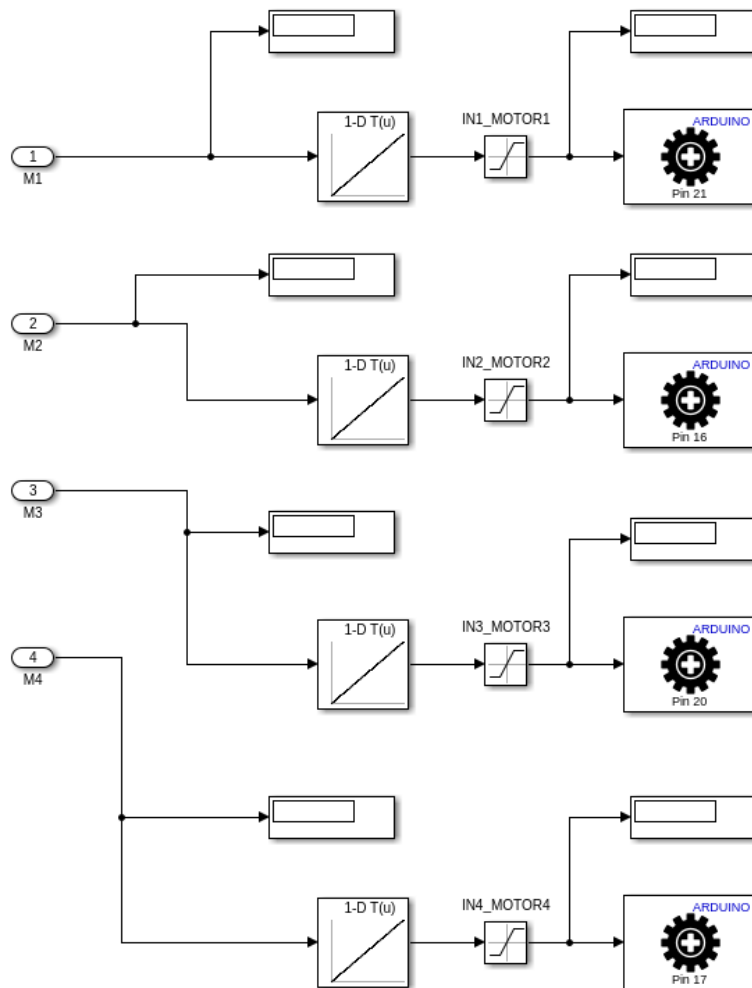


Figure 4.28 – Le sous-système Quadcopter Motors Controller pour le pilotage des moteurs BLDC

### 4.3.2 Réglage du contrôleur

Les valeurs de PIDs du contrôleur d'attitude utilisées pour la simulation du modèle physique du drone ne fonctionnent pas sur le matériel réel comme prévu, ceci pourrait être dû à plusieurs facteurs notamment une connaissance partielle de notre drone (estimation peu précise des coefficients aérodynamiques), la mise en forme de signaux en entrée des ESCs pour pouvoir piloter les moteurs BLDC (introduction des blocs de saturation et table de conversion) et la discrétisation des contrôleurs PIDs.

Nous avons été amené à régler de nouveau les contrôleurs PID pour notre matériel réel pendant que le bloc contrôleur d'attitude tourne en temps réel sur le matériel. Le processus de réglage est le même que celui décrit précédemment (voir paragraphe [4.2.2](#)).

Le réglage des contrôleurs PID était un peu fastidieux, le quadcopter pouvait décoller et tenter de se stabiliser à l'horizontal, mais avec un très mauvais contrôle d'attitude. Le contrôleur a besoin de plus d'ajustements pour améliorer le contrôle du drone.

# Chapitre 5

## Conclusion et perspectives

### 5.1 Conclusion

Le but de ce projet était de mettre en place un nouveau support de travaux pratiques pour illustrer la dynamique du vol des véhicules autonomes et de développer un contrôleur Simulink basé sur le PID ainsi que son implémentation sur le matériel réel pour contrôler l'attitude du véhicule.

Le véhicule a été construit avec succès en utilisant des composants sélectionnés pour leur facilité d'utilisation et leur compatibilité les uns avec les autres. Un banc d'essai à 4 degrés de liberté a été réalisé afin de tester et de valider le fonctionnement du quadricoptère.

Le modèle Simulink du quadricoptère a été dérivé à partir des équations différentielles de la dynamique du vol du drone. Un contrôleur d'attitude basé sur le PID a été conçu et réglé pour le modèle Simulink du quadricoptère.

Le modèle non-linéaire de la dynamique du véhicule que nous avons élaboré dans ce projet bien que complexe ne représente pas parfaitement toutes les dynamiques hautement non-linéaire d'un quadricoptère. Nous avons ignoré d'autres phénomènes notamment le battement des pales (déformation des pales de l'hélice à cause des vitesses élevées et du matériau de pales qui est flexible) et les effets au sol. Nous n'avons pas pu déterminer avec précision les différents coefficients aérodynamiques du quadricoptère (les coefficients de poussée et de traînée), nous avons modélisé partiellement la résistance de l'air dans le modèle Simulink.

Les contrôleurs PID ont été utilisés dans ce projet pour la facilité de leur mise en place, ces contrôleurs n'offrent pas toujours de meilleures performances, il y a un compromis entre rapidité de la réponse et robustesse du système. Le contrôleur d'attitude a besoin de plus d'ajustements pour améliorer le contrôle du drone.

## 5.2 Perspectives

Il est possible d'améliorer les performances du contrôleur de notre quadcopter en améliorant le modèle du quadcopter et son contrôleur.

On peut agir sur la modélisation de nos actionnaires (tous les moteurs n'ont pas les mêmes caractéristiques), pour cela il faudrait identifier les caractéristiques de chaque moteur et intégrer une stratégie de contrôle qui s'appuie sur les équations des moteurs afin d'obtenir un contrôle précis des moteurs.

On pourrait améliorer la modélisation de nos capteurs en introduisant une fusion de données avec un filtre de Kalman pour obtenir des données plus précises, on peut également penser à l'utilisation d'un filtre complémentaire pour l'estimation précise des angles. L'ajout d'un capteur de position (capteur à fil, infrarouge, ultrason, etc.) pour le contrôle précis de l'altitude du drone serait une amélioration intéressante à envisager.

Le modèle physique du véhicule peut être optimisé également, en introduisant les équations décrivant la dynamique des hélices et des moteurs, la modélisation de la batterie notamment de son état de charge qui a une influence sur la commande de moteurs pour obtenir un comportement dynamique plus proche du quadricoptère dans le monde réel physique. À cela viennent s'ajouter la détermination précise des coefficients de poussée et de traînée.

Pour le contrôleur d'Attitude, d'autres techniques de contrôle peuvent être explorées pour améliorer ses performances, on peut notamment penser à un contrôleur LQR, une commande par linéarisation entrée-sortie, etc. À ce contrôleur on peut ajouter une estimation de certaines variables du modèle dynamique du véhicule en utilisant le filtre de Kalman mentionné précédemment.

Les améliorations introduites ci-dessus devraient être suffisantes pour atteindre les principaux objectifs de ce projet de quadricoptère pour son utilisation en tant que support de travaux pratiques en laboratoire.

# Bibliographie

- Ferry, Nicholas. 2017. *Quadcopter Plant Model and Control System Development With MATLAB/Simulink Implementation*. Department of Electrical and Microelectronic Engineering - Kate Gleason College of Engineering, Rochester Institute of Technology. 129 p.
- Fiaz, Usman A., Mukarram, Aniq, & Khan, Uzair. 2018. *Altitude Control of a Quadcopter*. Pakistan Institute of Engineering and Applied Sciences (PIEAS). Disponible sur [https://www.researchgate.net/publication/309486306\\_Altitude\\_Control\\_of\\_a\\_Quadcopter](https://www.researchgate.net/publication/309486306_Altitude_Control_of_a_Quadcopter). (consulté le 01 février 2023).
- Gibiansky, Andrew. 2012. *Quadcopter Dynamics, Simulation and Simulation*. [site web]. Disponible sur <https://andrew.gibiansky.com/blog/physics/quadcopter-dynamics/>. (consulté le 01 novembre 2022).
- Hossain, Raju, Rideout, Geoff, & Krouglicof, Nicholas. 2010. *Bond graph dynamic modeling and stabilization of a quad-rotor helicopter*. Proceedings of the 2010 Spring Simulation Multiconference. Society for Computer Simulation International.
- Jiinec, Tomas. 2011. *Stabilization and Control of Unmanned Quadcopter*. Faculty of Electrical Engineering Department of Cybernetics - Czech Technical University in Prague.
- Lee, Daewon, Kim, H. Jin, & Sastry, Shankar. 2009. *Feedback linearization vs. adaptive sliding mode control for a quadrotor helicopter*. International Journal of control, Automation and systems 7.3.
- Luukkonen, Teppo. 2011. *Modelling and control of quadcopter*. Independent research project in applied mathematics, Espoo.
- Mahony, Robert, Kumar, Vijay, & Corke, Peter. 2012. *Multicopter aerial vehicles : Modeling, estimation, and control of quadrotor*.
- Maker, Pro. 2016. *DIY Drone : How to Build a Quadcopter*. [site web]. Disponible sur <https://maker.pro/custom/projects/diy-quadcopter-tutorial>. (consulté le 09 janvier 2023).
- Pham, Minh Tu. 2018. *Commande des systèmes linéaires*. Département de Génie Mécanique - Institut National des Sciences Appliquées de Lyon. Polycopié du Cours 142 p.
- Tytler, Charles. 2017. *Modeling Vehicle Dynamics – Quadcopter Equations of Motion*. [site web]. Disponible sur <https://web.archive.org/web/20181122041812/http://charlestytler.com/quadcopter-equations-motion/>. (consulté le 09 janvier 2023).
- Usman, Muhammad. 2020. *Quadcopter Modelling and Control MATLAB/Simulink Implementation*. Department of Mechanical Engineering and Production Technology - LAB University of Applied Sciences Technology Lappeenranta. 68 p.

## Annexe A

# Identification des paramètres du modèle du quadricoptère

### Masse et matrice d'inertie du quadricoptère

Afin de déterminer la masse du véhicule et sa matrice d'inertie, nous avons créé un modèle approché du drone sur un logiciel de conception assistée par ordinateur (figure A.1), nous avons utilisé dans notre application le logiciel Onshape<sup>1</sup>. Ce logiciel permet de déterminer la masse totale du véhicule et sa matrice d'inertie en fournissant les bonnes propriétés de matériau de chaque pièce de l'assemblage, ici notamment la masse volumique. La masse de chaque pièce de l'assemblage a été mesurée à l'aide d'une balance de précision.

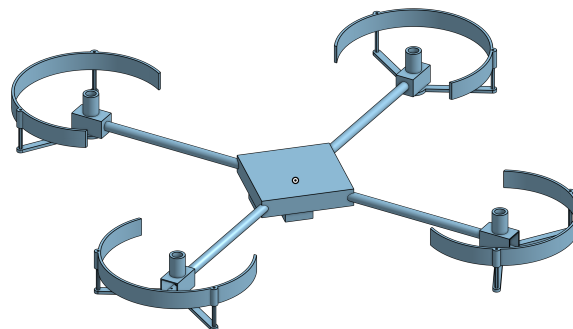


Figure A.1 – Modélisation du quadricoptère sur le logiciel Onshape

Le tableau suivant donne la masse de chaque partie du drone quadricoptère, la masse volumique est obtenue de manière approchée par le rapport de la masse pesée sur le volume de la pièce donné par le logiciel de conception (chaque pièce est supposée homogène).

Pièce	Masse [kg]	Volume [mm <sup>3</sup> ]	Masse volumique [kg/mm <sup>3</sup> ]
bras	0,02045	14074,33500	0,000001453
rotor & hélice	0,12752	19966,44173	0,000006387
protection d'hélice	0,10659	85232,78013	0,000001251
partie centrale	0,44622	132200,818	0,000003375
batterie	0,28024	126000	0,000002224
coulisseau glissière	0,34312	50625	0,000006778
Total pour l'Assemblage	2,08782	785920,04544	-

Table A.1 – Masse et volume de chaque pièce de l'assemblage du drone

1. Onshape est un logiciel de conception assistée par ordinateur développé par l'entreprise PTC.

Notre drone étant symétrique, nous avons obtenu la matrice d'inertie suivante.

$$I = \begin{bmatrix} I_{xx} = 0.07989 \text{ kgm}^2 & 0 & 0 \\ 0 & I_{yy} = 0.07989 \text{ kgm}^2 & 0 \\ 0 & 0 & I_{zz} = 0.15594 \text{ kgm}^2 \end{bmatrix} \quad (\text{A.1})$$

## Moment d'inertie des hélices

Le moment d'inertie des hélices est tiré de travaux de la thèse de maîtrise de Tomas Jiinec ([Jiinec, 2011](#)). Plus précisément dans la section 4.1.1, il calcule le moment d'inertie d'une hélice APC 10x4.7 qui est presque identique à hélice Lynxmotion 9x4.5R utilisée dans ce projet. Il mesure le poids et la densité de l'hélice puis divise l'hélice en plus petites sections et estime le moment d'inertie total comme la somme des moments d'inertie de chaque section en utilisant le théorème des axes parallèles (théorème de Huygens).

$$I_r = 0.04439 \text{ kgm}^2 \quad (\text{A.2})$$

## Coefficients de poussée et de traînée

Le coefficient de poussée dépend du poids du véhicule, du régime des rotors, du pas et du diamètre des hélices. Lorsque le quadcoptère est en vol stationnaire, la poussée des quatre rotors est égale au poids du véhicule. Ainsi, la poussée d'un rotor correspond à un quart du poids du quadcoptère.

$$\text{Poids} = 4 \times \text{coefficient de poussée} \times \text{vitesse de rotation de rotors en vol stationnaire.}$$

Le coefficient de traînée dépend de l'inertie du véhicule, du régime du rotor et du pas et du diamètre des hélices.

Nous n'avons pas été en mesure d'estimer les valeurs de coefficients de poussée et de traînée de notre quadricoptère. Nous avons utilisé pour notre modèle de simulation et pour l'implémentation du contrôleur sur le prototype les valeurs tirées de travaux de Muhammad Usman ([Usman, 2020](#)).  $b$  est le coefficient de poussée et  $d$  le coefficient de traînée.

$$b = 3.59 * 10^{-5} \text{ kgm} \quad (\text{A.3})$$

$$d = 2.0810 * 10^{-6} \text{ kgm} \quad (\text{A.4})$$

La détermination précise des coefficients de poussée et de traînée devrait constituer un enjeu majeur pour la suite du projet.

## La résistance de l'air

Il est difficile d'estimer la résistance de l'air qui s'exerce sur un quadricoptère. Nous avons utilisé dans ce projet une simple valeur estimée dans la thèse de maîtrise de Nicholas Ferry qui n'affecte pas de manière significative le comportement du quadricoptère dans un environnement fermé. ([Ferry, 2017](#))

$$A = 0.1 \text{ kg/s} \quad (\text{A.5})$$

## Annexe B

# Paramètres du modèle du quadricoptère - code MATLAB

```
clear
close all
clc
%% Parameters
% Physical Constants, Inertial and geometrical properties
m = 2.0878; %[Kg] Mass
g = 9.81; % [m/s^2] Gravity
l = 0.32; % [m] Length of the arm
Ixx = 0.07989; % [kg.m2] Moment of inertia along x axis
Iyy = 0.07989; % [kg.m2] Moment of inertia along y axis
Izz = 0.15594; % [kg.m2] Moment of inertia along z axis
Ir = 0.04439; % [kgm2] Propeller Moment of Inertia (Rotor Inertia)
b = 3.59*10^-5; % [kgm] Thrust Coefficient
d = 2.0810*10^-6; % [kgm] Drag Torque Coefficient
Ax = 0.1; % [kg/s] Air Resistance along x axis
Ay = 0.1; % [kg/s] Air Resistance along y axis
Az = 0.1; % [kg/s] Air Resistance along z axis

% The knowledge of b & d has to be enhanced
```

## Annexe C

# Commande des moteurs BLDC avec des ESC sur MATLAB

```
clear
clc
close all

port = '/dev/ttyUSB0'; % Port on which the Arduino is plugged
board = 'Mega2560'; % Arduino board name
a = arduino(port, board, 'Libraries', 'Servo');

% Motors - assign a servo object to each DC Brushless motor
% MOTOR1 & MOTOR3
% MOTOR2 & MOTOR4
MOTOR1 = servo(a, '21');
MOTOR2 = servo(a, '16');
MOTOR3 = servo(a, '20');
MOTOR4 = servo(a, '17');

pwm = 0.6709;
% The functions writePosition takes value from 0 to 1

tic
while toc < 5
    % Set PWM
    writePosition(MOTOR1, pwm);
    writePosition(MOTOR2, pwm);
    writePosition(MOTOR3, pwm);
    writePosition(MOTOR4, pwm);
    % drawnow --> Interesting function to have a look on
end
pwm = 0;
writePosition(MOTOR1, pwm);
writePosition(MOTOR2, pwm);
writePosition(MOTOR3, pwm);
writePosition(MOTOR4, pwm);
```

## Annexe D

# Commande des moteurs BLDC avec des ESC sur IDE Arduino

```
1 // Based on Arduino Sweep Example by BARRAGAN <http://barraganstudio.com>
2
3 // Motors Pins
4 #define MOTOR1 21 // MOTOR_D3
5 #define MOTOR2 16 // MOTOR_D5
6 #define MOTOR3 20 // MOTOR_D6
7 #define MOTOR4 17 // MOTOR_D2
8
9 #include <Servo.h>
10
11 // create servo object to control a servo
12 Servo myservo1;
13 Servo myservo2;
14 Servo myservo3;
15 Servo myservo4;
16
17 int pos = 0; // variable to store the servo position
18
19 void setup()
20 {
21   myservo1.attach(MOTOR1); // attaches the servo on pin 9 to the servo
    object
22   myservo2.attach(MOTOR2); // attaches the servo on pin 9 to the servo
    object
23   myservo3.attach(MOTOR3); // attaches the servo on pin 9 to the servo
    object
24   myservo4.attach(MOTOR4); // attaches the servo on pin 9 to the servo
    object
25 }
26
27
28 void loop()
29 {
30   //digitalWrite(MOTOR1, HIGH);
31   for(pos = 0; pos < 180; pos += 1) // goes from 0 degrees to 180
    degrees
32   {
33     myservo1.write(pos); // in steps of 1 degree // tell servo to go to position in
```

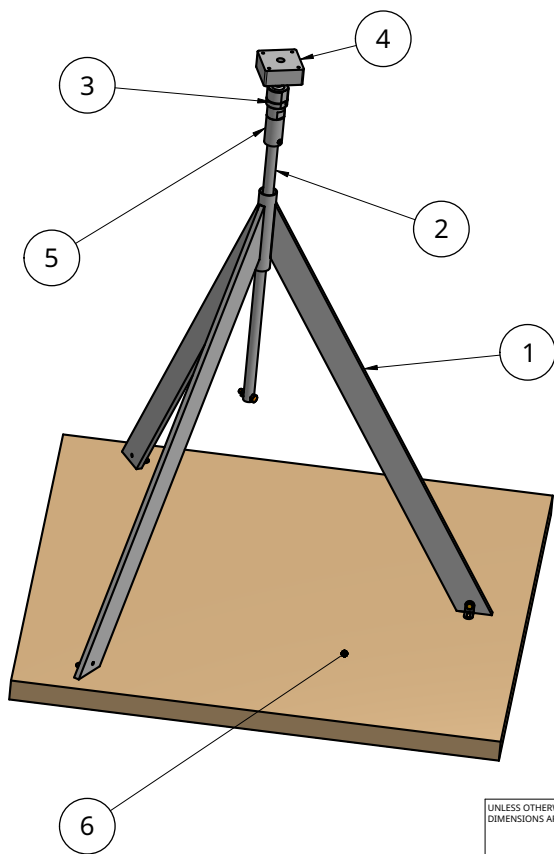
```

        variable 'pos'
34   myservo2.write(pos);
35   myservo3.write(pos);
36   myservo4.write(pos);
37   delay(15);           // waits 15ms for the servo to reach
        the position
38
39   }
40   for(pos = 180; pos>=1; pos-=1) // goes from 180 degrees to 0
        degrees
41   {
42   myservo1.write(pos); // tell servo to go to position in
        variable 'pos'
43   myservo2.write(pos);
44   myservo3.write(pos);
45   myservo4.write(pos);
46   delay(15);           // waits 15ms for the servo to reach
        the position
47   }
48 }

```

## Annexe E

# Dessin d'ensemble - banc d'essai à 4 degrés de liberté pour quadricoptère

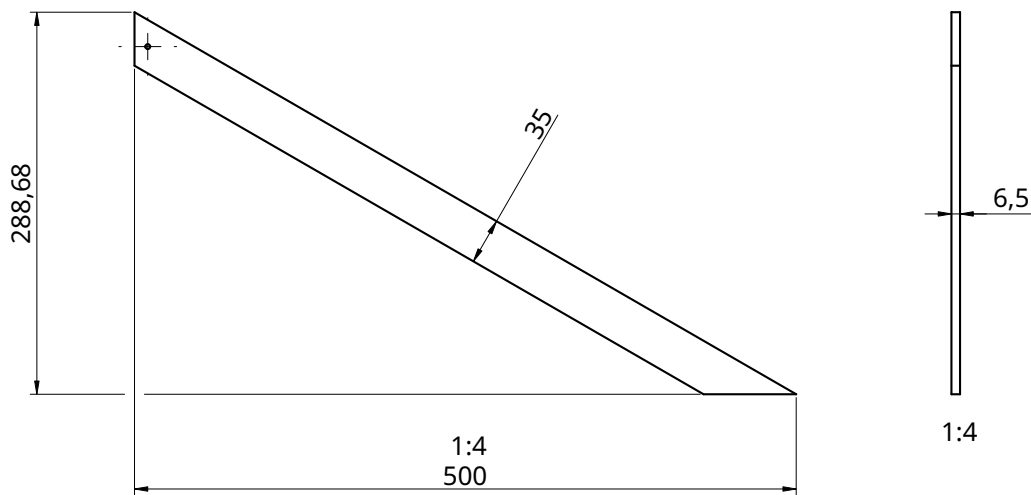


Repère	Désignation	Qté
1	Trépied	1
2	Coulisseau	1
3	Embouts axiaux à rotule réglable	1
4	Fixation drone	1
5	Fixation coulisseau	1
6	Bâti support	1

UNLESS OTHERWISE SPECIFIED, DIMENSIONS ARE IN MILLIMETERS		NAME	SIGNATURE	DATE	TITLE Dessin d'ensemble - banc d'essai à 4 degrés de liberté pour quadricoptère
ANGULAR = ± °		JOYCE SUDI		2022-12-07	
SURFACE FINISH					
DO NOT SCALE DRAWING					
BREAK ALL SHARP EDGES AND REMOVE BURRS					SIZE A4
FIRST ANGLE PROJECTION		MATERIAL	FINISH		DWG NO.
					SCALE 1:6
					WEIGHT
					SHEET 1 of 1

# Annexe F

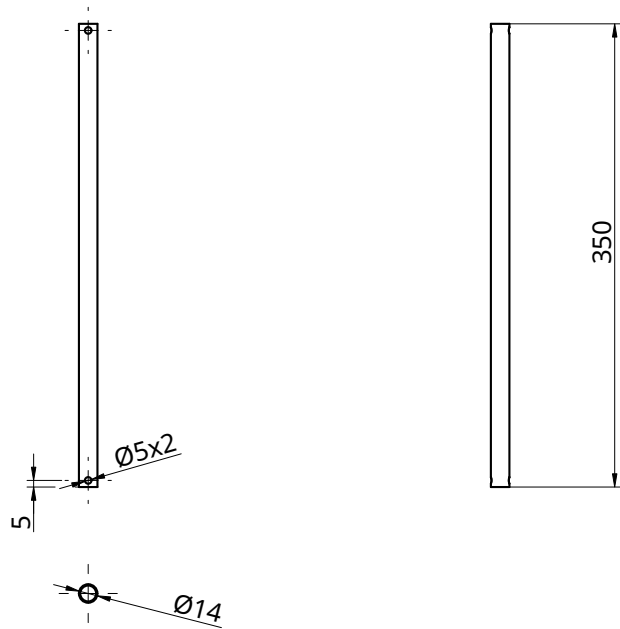
## Dessin de définition - pied



UNLESS OTHERWISE SPECIFIED, DIMENSIONS ARE IN MILLIMETERS  ANGULAR = ± °  SURFACE FINISH ✓  DO NOT SCALE DRAWING  BREAK ALL SHARP EDGES AND REMOVE BURRS  FIRST ANGLE PROJECTION		NAME	SIGNATURE	DATE	TITLE  <b>Pied</b>		
	DRAWN	JOYCE SUDI		2022-12-07			
	CHECKED				SIZE <b>A4</b>		
	APPROVED						
	MATERIAL	FINISH		SCALE	WEIGHT	SHEET	
				<b>1:8</b>		<b>1 of 1</b>	

# Annexe G

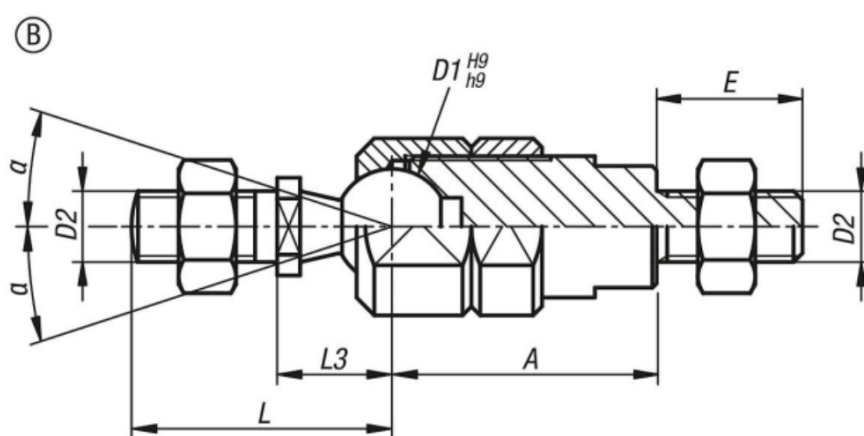
## Dessin de définition - coulisseau



UNLESS OTHERWISE SPECIFIED, DIMENSIONS ARE IN MILLIMETERS  ANGULAR = ± °	DRAWN	NAME	SIGNATURE	DATE	TITLE  <b>Coulisseau</b>		
	CHECKED	JOYCE SUDI		2023-02-04			
	APPROVED						
	DO NOT SCALE DRAWING						
BREAK ALL SHARP EDGES AND REMOVE BURRS					SIZE <b>A4</b>	DWG NO.	REV.
FIRST ANGLE PROJECTION	MATERIAL	FINISH		SCALE <b>1:4</b>	WEIGHT	SHEET <b>1 of 1</b>	

## Annexe H

### Extrait du datasheet - embouts axiaux à rotule réglable M10

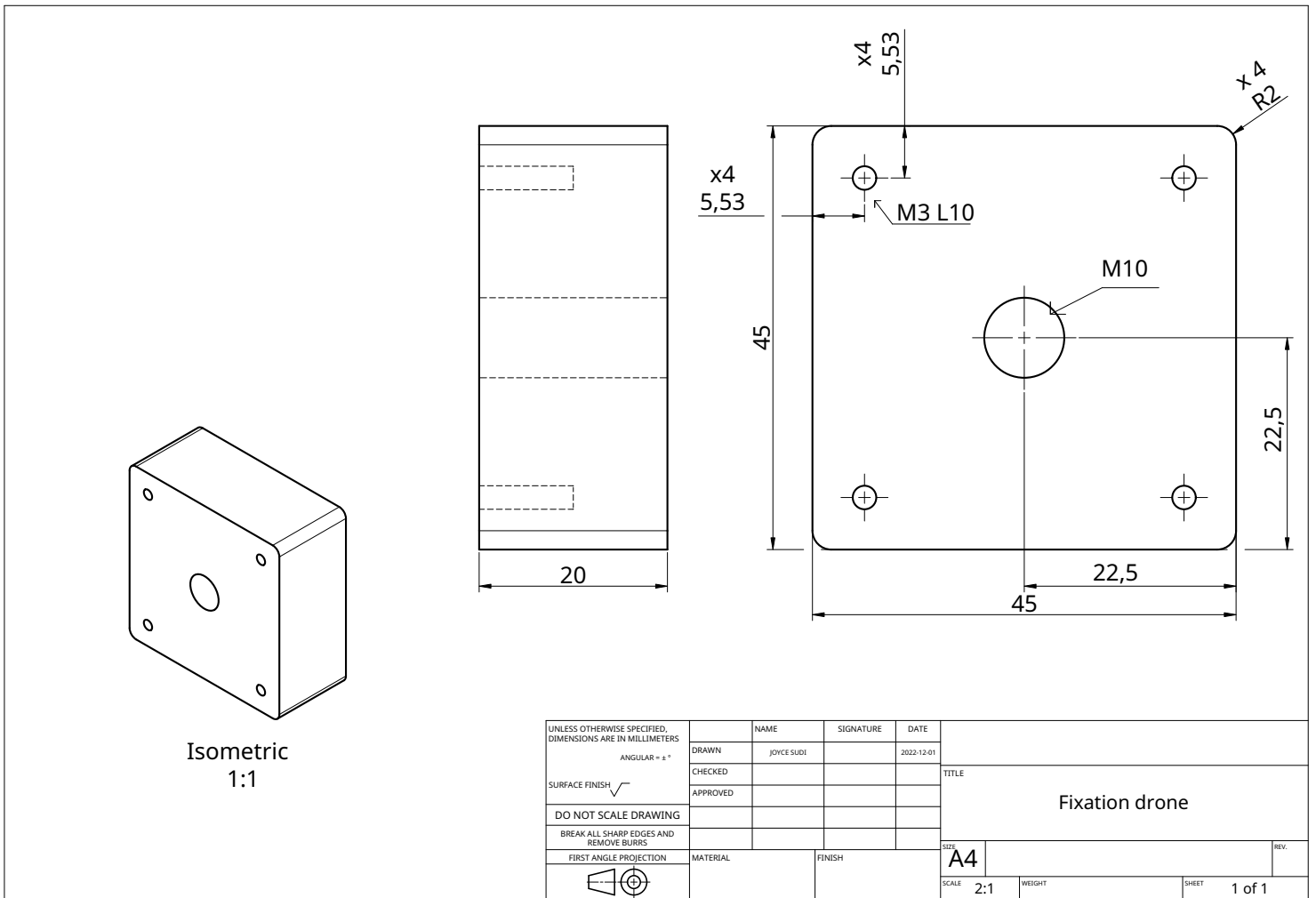


#### Aperçu des articles

Référence	Forme	D1	D2	A	L	L3	E	$\alpha$	F kN	M Nm	Couple de serrage max Nm
27670-21006	B	10	M6	21	23,5	11	12	20°	2,5	5,2	12,4
27670-21308	B	13	M8	26	29,5	13	14	24°	5	8,8	13,2
27670-21610	B	16	M10	35	36	16	16	28°	7,5	14,5	16,4
27670-21612	B	16	M12	35	36	16	16	28°	7,5	14,5	16,4

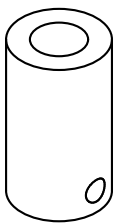
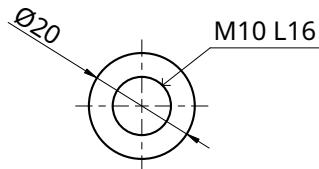
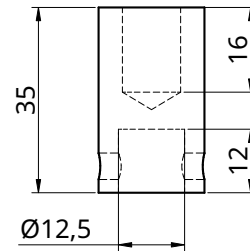
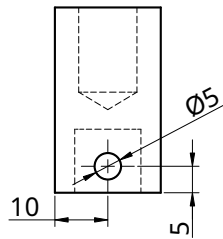
# Annexe I

## Dessin de définition - fixation drone



# Annexe J

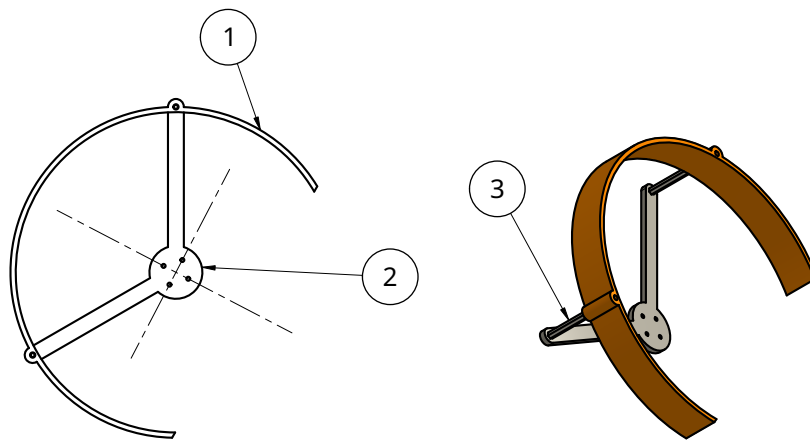
## Dessin de définition - fixation coulisseau



UNLESS OTHERWISE SPECIFIED, DIMENSIONS ARE IN MILLIMETERS  ANGULAR = ± °  SURFACE FINISH $\sqrt{\quad}$  DO NOT SCALE DRAWING  BREAK ALL SHARP EDGES AND REMOVE BURRS  FIRST ANGLE PROJECTION	DRAWN	NAME	SIGNATURE	DATE	TITLE  <b>Fixation coulisseau</b>
	CHECKED	JOYCE SUDI		2022-12-01	
	APPROVED				
	MATERIAL	FINISH		SIZE <b>A4</b>	DWG. NO.
				SCALE 1:1	WEIGHT
					SHEET 1 of 1

# Annexe K

## Dessin d'ensemble - protection d'hélice



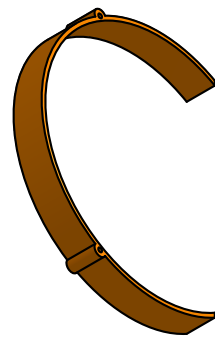
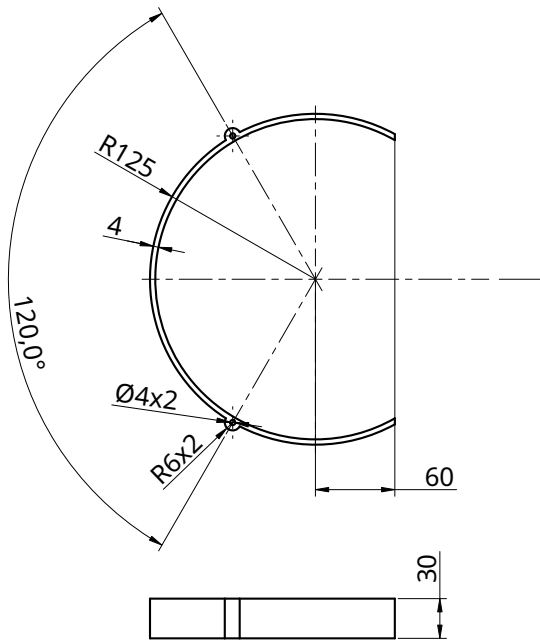
Repère	Désignation	Qté
1	Part 1	1
2	Part 2	1
3	Entretoise Hexagonale M3x50mm	2

<small>UNLESS OTHERWISE SPECIFIED, DIMENSIONS ARE IN MILLIMETERS</small> <small>ANGULAR = ± °</small> <small>SURFACE FINISH</small> ✓ <small>DO NOT SCALE DRAWING</small> <small>BREAK ALL SHARP EDGES AND REMOVE BURRS</small> <small>FIRST ANGLE PROJECTION</small>	NAME	SIGNATURE	DATE	<small>TITLE</small> <b>Dessin d'ensemble - protection d'hélice</b>
	DRAWN	JOYCE SUDI	2023-02-02	
	CHECKED			
	APPROVED			
	MATERIAL	FINISH	<small>SIZE</small> <b>A4</b>	<small>DWG NO.</small> 
			<small>SCALE</small> <b>1:4</b>	<small>WEIGHT</small> 
			<small>SHEET</small> <b>1 of 1</b>	<small>REV.</small> 



# Annexe M

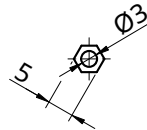
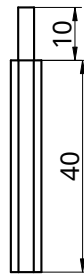
## Dessin de définition - protection d'hélice part 2



UNLESS OTHERWISE SPECIFIED, DIMENSIONS ARE IN MILLIMETERS  ANGULAR = ± °  SURFACE FINISH ✓  DO NOT SCALE DRAWING  BREAK ALL SHARP EDGES AND REMOVE BURRS  FIRST ANGLE PROJECTION	NAME	SIGNATURE	DATE	TITLE  <b>Protection d'hélice part 2</b>		
	DRAWN	JOYCE SUDI	2023-02-02			
	CHECKED					
	APPROVED				SIZE <b>A4</b>	DWG NO.
	MATERIAL	FINISH		SCALE 1:4	WEIGHT	SHEET 1 of 1

# Annexe N

## Dessin de définition - entretoise hexagonale M3x50mm



UNLESS OTHERWISE SPECIFIED, DIMENSIONS ARE IN MILLIMETERS  ANGULAR = ±°  SURFACE FINISH ✓  DO NOT SCALE DRAWING  BREAK ALL SHARP EDGES AND REMOVE BURRS  FIRST ANGLE PROJECTION	NAME	SIGNATURE	DATE	TITLE  <b>Entretoise Hexagonale M3x50mm</b>
	DRAWN	JOYCE SUDI	2023-02-02	
	CHECKED			
	APPROVED			
	MATERIAL	FINISH	SIZE <b>A4</b> DWG NO. SCALE <b>1:1</b> WEIGHT SHEET <b>1 of 1</b>	REV.