

République Algérienne Démocratique et Populaire
Université Abou Bakr Belkaid– Tlemcen
Faculté des Sciences
Département d'Informatique



Mémoire de fin d'études
Pour l'obtention du diplôme de Master en Informatique

Option : Génie Logiciel (G.L)

Thème

***Étude & implémentation d'une loi de
commande pour le pilotage d'un drone
quadri-rotor***

Réalisé par :

- *BENAISSA Mohammed Rida*
- *BELKHIRI Youcef*

Présenté le 4 Juillet 2017 devant le jury composé de MM.

- *Mr. H. MAHFOUD* (Président)
- *Mr. M. Benaissa* (Examinateur)
- *Mr. S. ZIANI-CHERIF* (Encadreur)

Remerciement :

Nous tenons à remercier tout d'abord notre encadreur, **Dr. Salim Ziani Cherif**, pour sa patience, et surtout pour sa confiance, ses remarques et ses conseils, sa disponibilité et sa bienveillance. Qu'il trouve ici le témoignage de notre profonde gratitude. Nous voudrions également remercier les membres du jury pour avoir accepté d'évaluer ce travail et pour toutes leurs remarques et critiques, ainsi que le personnel et les enseignants du département de l'informatique. Un grand merci à nos parents pour leur soutien moral et matériel. Nous tenons aussi à remercier Mlle. Benaissa Amina ainsi que tous nos amis et collègues pour leur soutien inestimable. A tous nos enseignants qui m'ont initié aux valeurs authentiques, en signe d'un profond respect et d'un profond amour.

Merci à vous tous

Dédicace :

Je dédie ce mémoire A mes parents pour leur amour inestimable, leur confiance, leur soutien, leurs sacrifices et toutes les valeurs qu'ils ont su m'inculquer.

A mes sœurs ainsi qu'à l'esprit de mon frère, mes tentes et mes grands-parents.

A tous mes amis qui m'aident, m'encouragent et m'aiment, à toutes les personnes qui souhaitent mon succès.

Mohammed Rida Benaisa.

Dédicaces

Je dédie ce modeste travail, aux deux êtres les plus chers à mon cœur auxquels je dois mon existence :

Mon père et ma mère ; vous qui étiez toujours à mes côtés pour me soutenir et m'encourager à me battre sans jamais m'arrêter à mi-chemin ; que DIEU vous protège.

A mon cher frère : **Ismail**.

A mes chères **sœurs**.

A mes très chères amis : **Seyf-Allah, Ali, Bilal, Zakaria, Abdou et Khaled**

A tous mes amis de la promotion 2^{ème} année Master informatique « **GL** » 2016/2017 et tous ceux qui m'ont aidé durant cette recherche.

A ceux-là, et à tous ceux que j'aurais oublié de citer, j'exprime mon infaillible reconnaissance et ma sincère gratitude.

BELKHIRI Youcef.

Table des matières :

Introduction générale :	4
Chapitre1 : Modélisation dynamique et loi de commande.	
1. Introduction :	6
2. Aperçu historique sur les drones :	6
3. Les différentes catégories de drones :	7
3.1 Classification en fonction de R.E.A :	7
3.2 Classification selon le mode de vol :	8
3.2.1. Drones à voilure Fixe :	8
3.2.2. Drones à voilure tournante :	8
3.2.3. Drones à ailes battante :	8
4. Modèle dynamique :	9
4.1. Les repères :	9
4.2. Les mouvements du quadrirotor :	10
4.2.1. Mouvement de roulis :	11
4.2.2. Mouvement de tangage :	12
4.2.3. Mouvement de lacet :	12
4.2.4. Mouvement vertical :	13
4.2.5. Translation horizontale :	14
4.3. Les angles d'Euler :	15
4.4. Vitesses angulaires	17
4.5. Vitesses linéaires	17
4.6. Les efforts physiques agissants sur le quadrirotor	18
4.6.1. Les force :	18
4.6.2. Les moments :	18
4.6.3. Effet gyroscopique :	19
4.7. Modèle mathématique du quadrirotor selon Newton-Euler :	20
4.7.1. Equations de mouvement de translation :	21
4.7.2. Equations de mouvement de rotation :	21
4.7.3. Dynamique du rotor :	22
4.8. La représentation d'état du système :	23
5. Les fonctions de transformations :	24
6. Loi de commande par mode glissant :	24
1.1. Régime glissant idéal :	25
1.2. Régime glissant réel :	26
1.3. Conception de la commande par mode glissant :	26
6. Conclusion :	27

Chapitre2: Modélisation SysML.

1. Introduction :	29
2. Qu'est-ce que SysML ?	29
3. Pourquoi SysML ?	29
4. Diagramme d'exigences :	30
5. Diagramme de cas d'utilisations :	31
5.1. Recherche des cas d'utilisation :	31
5.1.1. Acteur 1 : L'application :	32
5.1.2. Acteur 2 : L'utilisateur « Pilote » :	32
6. Diagramme de séquences :	34
6.1. Diagramme de séquences Auto piloter :	34
6.2. Diagramme de séquence contrôler :	36
7. Diagramme d'états transitions :	38
8. Diagramme de définition de blocs :	39
8.1. Qu'est-ce qu'un bloc ?	39
8.2. Diagramme de notre système :	39
9. Conclusion :	40

Chapitre3 : Application développée et résultats de simulations.

1. Introduction :	42
2. Résultat de simulation :	42
2.1. Les valeurs des paramètres du modèle quadrirotor utilisé :	42
2.2. Simulation de la première trajectoire :	43
2.3. Simulation de la deuxième trajectoire :	47
3. Les technologies utilisées :	50
3.1. Enterprise Architect :	50
3.2. PHP 5 :	50
3.3. PHP Desktop chrome :	51
3.4. Dev-c++ :	51
3.5. Langage C :	51
3.6. Photoshop :	51
3.7. Chart Js & Vis JS :	51
4. Interface homme machine :	51
5. Conclusion :	52
Conclusion générale :	54

Introduction général

Introduction générale :

Les différentes disciplines existantes à nos jours ont connu une véritable révolution après l'évolution du domaine informatique qui est devenu un outil incontournable, voire indispensable. Tout particulièrement, il permet de contrôler et piloter les engins d'une manière autonome et presque parfaite. De plus il donne la possibilité d'effectuer des missions qui étaient impossibles auparavant.

Cette grande évolution a donné notamment naissance à la robotique qui compte parmi les domaines informatiques les plus convoités de nos jours. Elle combine plusieurs disciplines à la fois. En outre, elle se base également sur le développement d'un modèle mathématique pour donner plus d'exactitude, de robustesse et d'autonomie, ce qui implique naturellement la limitation de l'intervention humaine en phase opératoire.

La commande des drones est une branche de la robotique aérienne qui combine de nombreuses disciplines dont la mécanique, l'aérodynamique, la physique fondamentale, l'électronique et les mathématiques et voit son aboutissement grâce aux technologies informatiques.

Notre étude a pour objectif l'implémentation d'une loi de commande des drones « quadrirotor ».

Le présent rapport est ainsi organisé en 3 chapitres :

Dans le chapitre 1 nous allons découvrir les différentes catégories de drones. Nous allons aussi étudier la modélisation dynamique d'un drone quadrirotor dans des mouvements conventionnels tels que la rotation et la translation, suivie d'une vision générale sur les fonctions de transformations et finissant par la présentation de la loi de commande utilisée dans le développement de notre système.

Dans le chapitre 2 nous allons présenter une modélisation informatique en utilisant le langage SysML (system modeling language) pour mieux comprendre le système en terme informatique.

Dans le chapitre 3, nous allons présenter la partie IHM de notre projet et les résultats de simulation obtenus par cette dernière pour visualiser l'efficacité de la loi de commande utilisée, ainsi qu'une vision sur les technologies utilisées pour la construction et l'implémentation de cette loi

Chapitre I

Modélisation dynamique

et loi de commande

1. Introduction :

Un drone ou UAV (acronyme de « Unmanned Aerial Vehicle ») est un aéronef inhabité, inclus dans le domaine de la robotique aérienne ; le pilotage de ce genre d'engin se fait d'une manière autonome ou télépiloté.

Les drones peuvent effectuer des missions variées, pour des usages civils ou militaires selon son genre : cela repose sur les caractéristiques et les capacités de contrôle de leur vol dans les différents environnements.

Les performances d'un drone se mesurent par sa manière de réagir face aux différents facteurs de l'environnement tel que : les phénomènes aérodynamiques, les perturbations (vents) ... etc.

Avec l'évolution de la technologie dans le domaine électronique et les systèmes embarqués, la fabrication des engins de différentes tailles et l'embarquement de plusieurs types de capteurs simultanément sont devenu possibles. Ces derniers permettent de minimiser l'intervention humaine dans les missions complexes et dangereuses.

Dans ce chapitre nous allons présenter une vision historique sur les drones, ses différentes catégories, le modèle dynamique d'un drone quadrirotor et enfin une présentation de sa loi de commande.

2. Aperçu historique sur les drones :

Les drones sont avant tout une invention militaire ; leurs origines remontent aux conflits au Proche-Orient à la fin des années 1970.

Dans la guerre entre le colonisateur sioniste et les pays arabes, des modèles réduits des drones avec des appareils photo sont utilisés par les sionistes pour découvrir la région, mais il fallait quarante-cinq minutes pour développer les photos.

L'industrie aérospatiale sioniste (IAI) a lancé le programme Scout (des caméras de télévision volantes avec des ailes, un moteur et un système de communication).

Le développement de drones aériens s'accélère et suscite un intérêt croissant ; après l'utilisation militaire en renseignement et surveillance comme nous avons déjà dit l'utilisation civile a fait irruption dans ce domaine tel que la gestion des infrastructures, la surveillance jour et nuit et l'observation de sites sensibles [1].

3. Les différentes catégories de drones :

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

- ❖ Classification générale des drones en fonction de leur Rayon d'action, l'Endurance et l'Altitude (R.E.A).
- ❖ Classification selon le mode de vol.

3.1 Classification en fonction de R.E.A [2, 3, 4, 5, 6] :

Nom	Rayon d'action	Endurance	Altitude
Drones HALE (Haute Altitude Longue Endurance)	Plusieurs milliers de kilomètre	Trentaine d'heures	20 000 mètres
Drones MALE (Moyenne Altitude Longue Endurance)		Trentaine d'heures	5 000 – 15 000 mètres
Drones de combat UCAV (Unmanned Combat Air Vehicles)	2 800 km		12 200 mètres
Drones tactiques TUAV (Tactical Unmanned Aerial Vehicle)	Plus de 100 km	Dizaine d'heures	200-5000 mètres
Mini drones MAV (Mini Air Vehicule)	30 km		300 mètres
Micro/Nano drones	Dizaine de kilomètres	Vingtaine de minutes	

Tableau I.1 : classification des drones.

3.2 Classification selon le mode de vol :

3.2.1. Drones à voilure Fixe :

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

Parmi les inconvénients de cette configuration [4] :

- La nécessité d'une piste de décollage.
- Elle n'autorise pas le vol stationnaire.
- Le vol à basse vitesse est impossible.
- Le vol à basse altitude est impossible.

3.2.2. Drones à voilure tournante :

Ce genre de drone a la capacité de décoller, voler et atterrir verticalement, ils utilisent des voilures tournantes ou rotor multiples pour voler en état stationnaire, ils n'ont pas besoin d'une piste de décollage/atterrissage et ils peuvent réaliser une très grande variété de missions irréalisables par des drones à voilure fixe [7].

Parmi les types de ce genre :

- Les hélicoptères classiques (figure I.1).
- Engins à rotor tandem
- Engins à rotors coaxiaux contrarotatifs
- Engins à rotors engrenant
- Engins à rotors coaxiaux carénés
- Engins à rotors en tandem pivotants [3].

3.2.3. Drones à ailes battante :

Le guidage et le pilotage de ce genre de drones se fait grâce à des ailes battante. Ils imitent les insectes, ils peuvent aussi faire des vols stationnaires à basse vitesse et effectuées des missions dans des espaces très réduits [8].

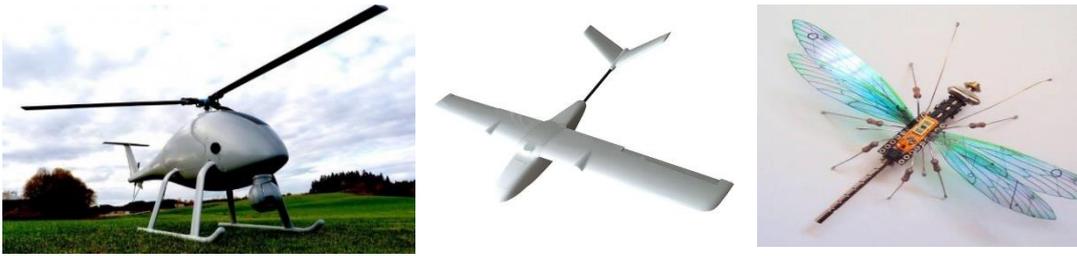


Figure 1.1 : Drone avec différents modes de vol

4. Modèle dynamique :

La conception et l'implémentation d'un contrôleur de vol, nécessite une compréhension profonde des mouvements de l'avion, de sa dynamique et par conséquent des équations dynamiques correspondantes.

Cette compréhension permet de réaliser une bonne conception du contrôleur et d'assurer une simulation plus proche de la réalité lorsque la commande est appliquée.

A cause du nombre énorme d'effets physiques affectés à sa dynamique (tel que les efforts aérodynamiques complexes, couples gyroscopiques, la gravité terrestre et les différents mouvements d'inertie ...etc.) le quadrirotor est classé parmi les systèmes volants les plus complexes [12].

4.1. Les repères :

On utilise deux repères afin de traiter le modèle mathématique du quadrirotor : un repère inertiel ou global attaché à la terre et supposé Galiléen défini par E (O, X, Y, Z) et un autre Local défini par B (o^{\prime} , x^{\prime} , y^{\prime} , z^{\prime}), jointe au centre de masse du corps du quadrirotor et localisé dans l'intersection des deux barres. On utilise les angles d'Euler (φ , θ , ψ) qui représentent respectivement le roulis, le tangage et le lacet [9], afin de déterminer l'orientation au cours de temps du repère quadrirotor par rapport au repère inertiel [10].

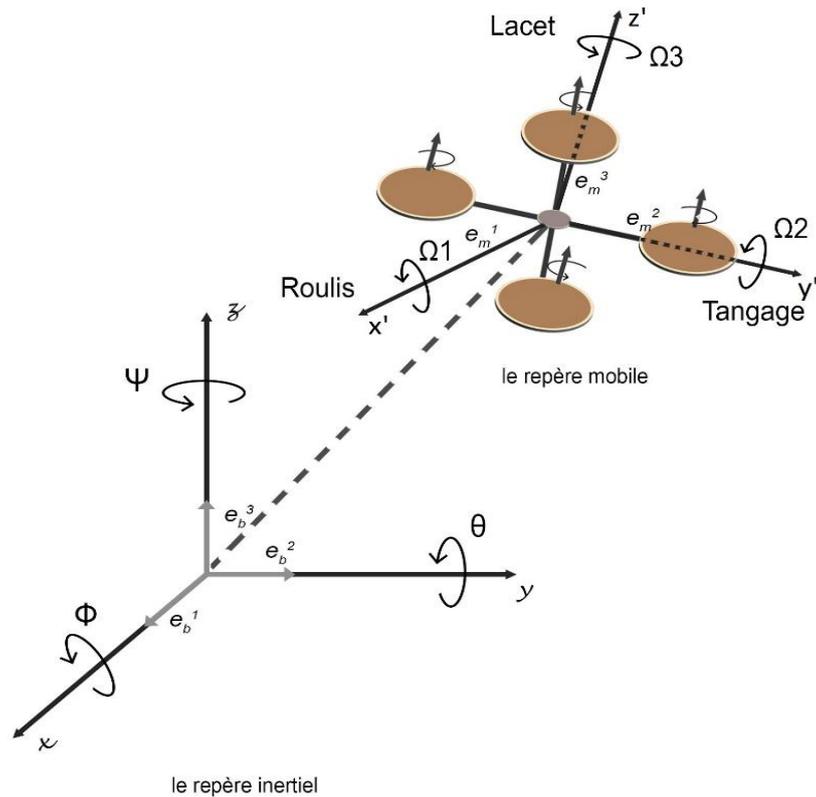


Figure 1.2 : Géométrie du quadrirotor

4.2. Les mouvements du quadrirotor :

Le quadrirotor est composé de 4 hélices, les rotors de droite et gauche (1 et 3) tournent dans le sens des aiguilles d'une montre et les rotors de l'avant et l'arrière (2 et 4) tournent dans le sens inverse pour neutraliser le mouvement du drone [11].

Pour faire glisser le drone il suffit de varier la vitesse des rotors pour produire une force de poussée vers le sens désiré.

Le quadrirotor a cinq mouvements principaux [12] :

- Mouvement de roulis
- Mouvement de tangage
- Mouvement de lacet
- Mouvement vertical
- Translation horizontale

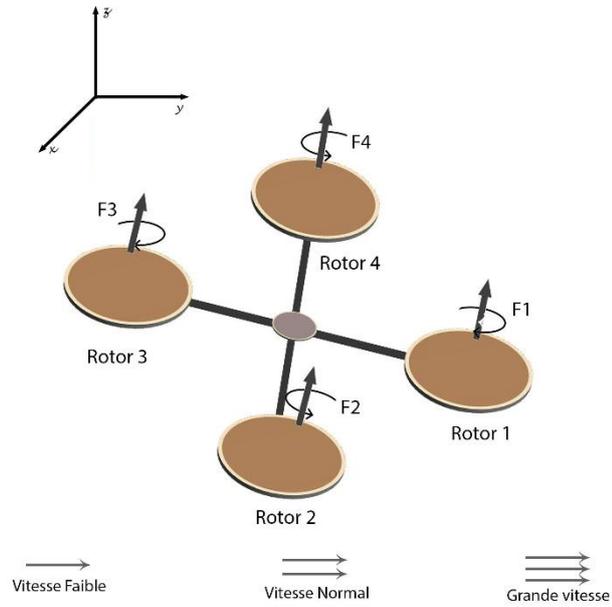


Figure I.3 : Structure générale d'un quadrirotor

4.2.1. Mouvement de roulis :

Pour réaliser ce mouvement on applique une variance de vitesse entre le rotor 1 et le rotor 3 pour faire une rotation autour de l'axe (x).

Ce mouvement est couplé avec un mouvement de translation selon l'axe (y) [13].

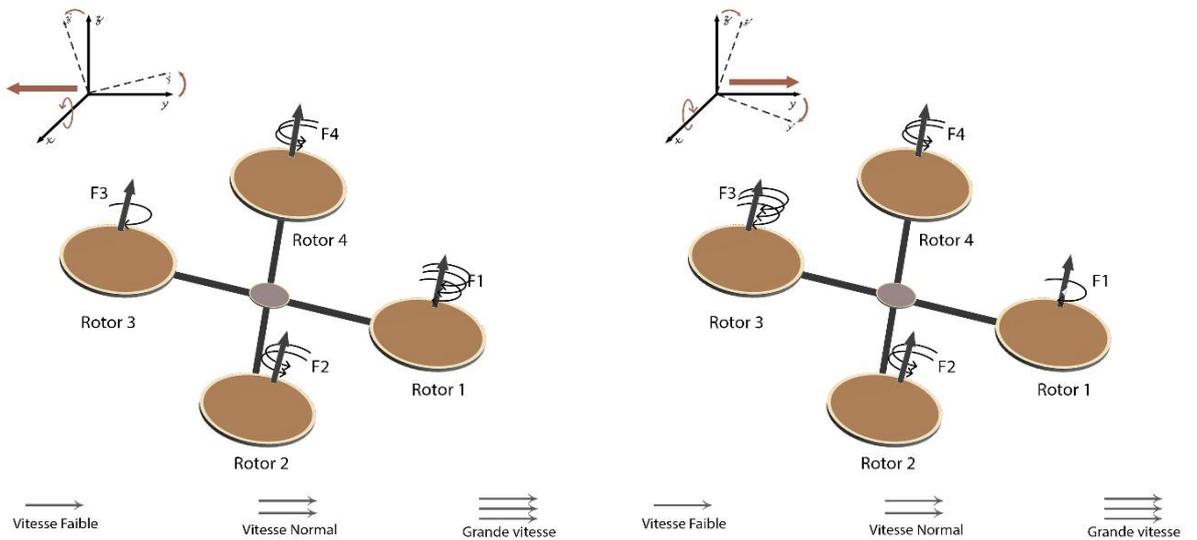


Figure I.4 : Illustration du mouvement de roulis

4.2.2. Mouvement de tangage :

Pour réaliser ce mouvement on applique une variance de vitesse entre le rotor 2 et le rotor 4 pour faire une rotation autour de l'axe (y).

Ce mouvement est couplé avec un mouvement de translation selon l'axe (x) [12, 13].

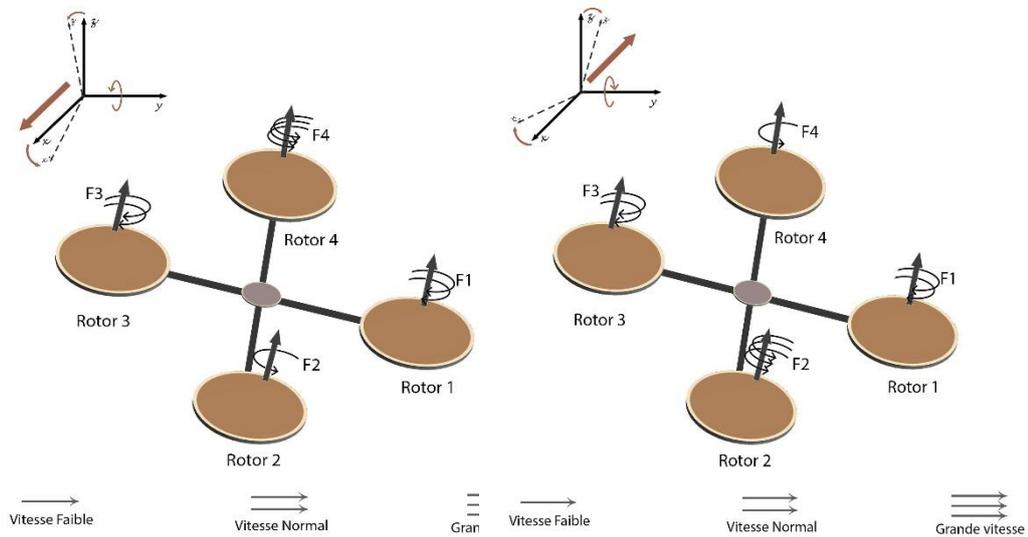


Figure 1.5 : Illustration du mouvement de tangage

4.2.3. Mouvement de lacet :

Pour réaliser ce mouvement on applique une variance de vitesse entre les rotors {1,3} et les rotors {2,4} (i.e. {1,3} de même vitesse et {2,4} de même vitesse) pour faire une rotation autour de l'axe (z).

La direction de la force de poussée ne décale pas pendant le mouvement, mais l'augmentation de la force de portance dans une paire de rotors doit être égale à la diminution des autres paires pour s'assurer que toute la force de poussée demeure la même [12, 13].

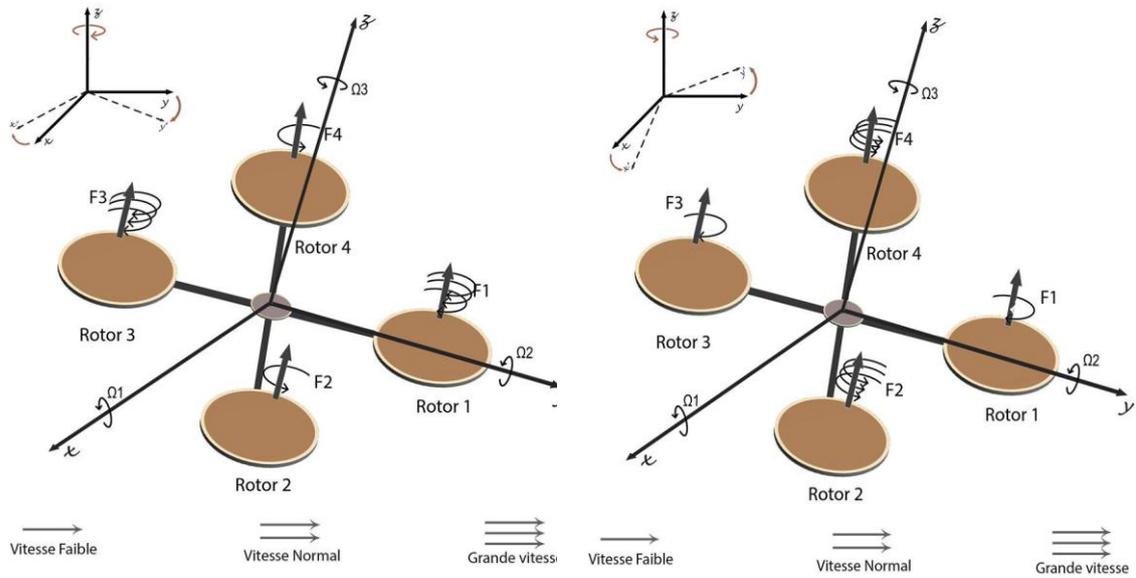


Figure 1.6 : Illustration du mouvement de lacet

4.2.4. Mouvement vertical :

Un mouvement ascendant ou descendant est réalisé par la variation de vitesse de rotation des quatre rotors en même temps et avec la même force de poussée [12, 13].

- Pour un mouvement ascendant on applique une grande vitesse de rotation.
- Pour un mouvement descendant on applique une vitesse faible de rotation.
- Pour un vol stationnaire on applique une vitesse moyenne de rotation.

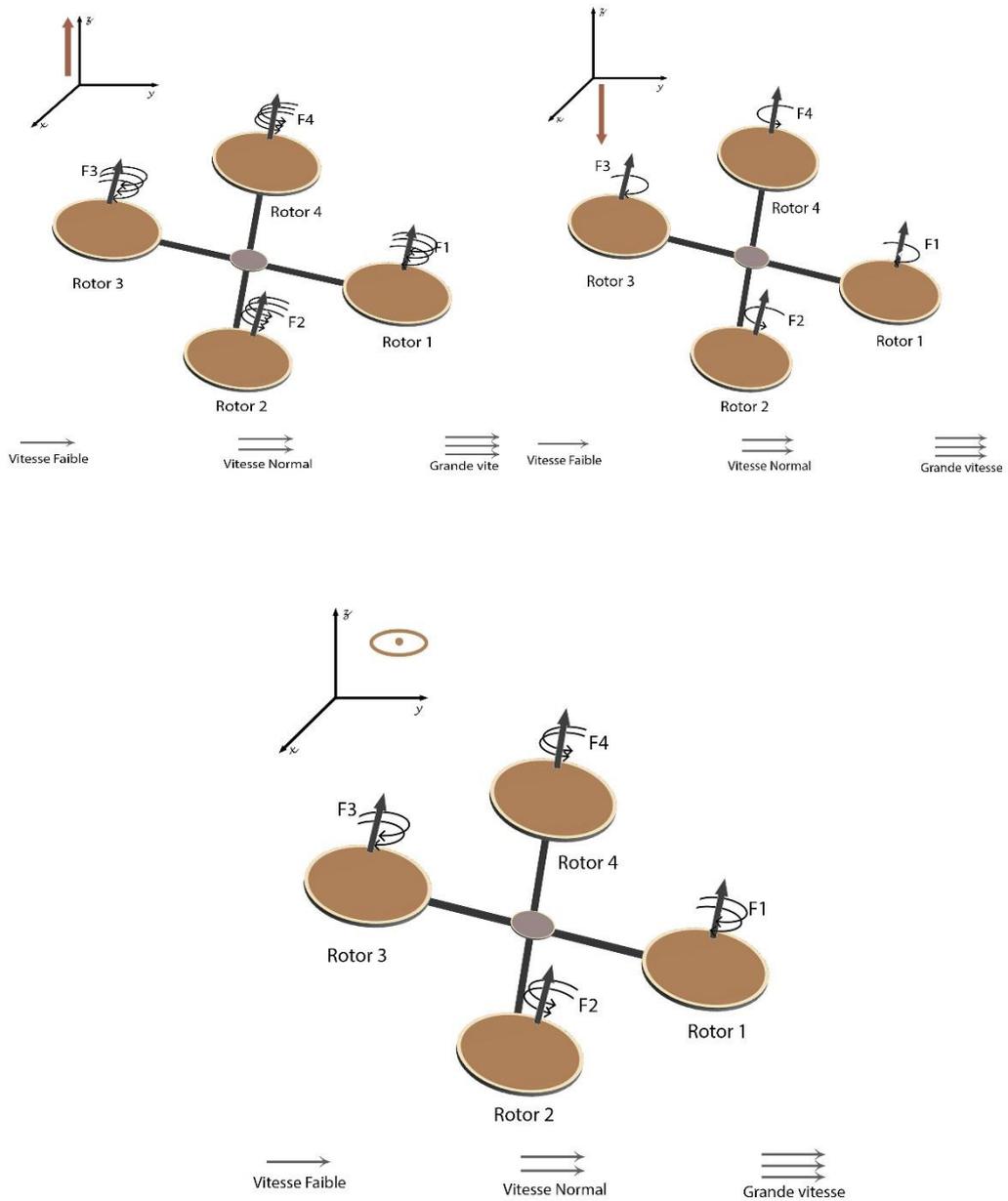


Figure 1.7 : Illustration du mouvement vertical

4.2.5. Translation horizontale :

Pour réaliser cela on applique une variance de vitesse entre 2 rotors opposé et garder une vitesse moyenne pour les deux autres. Dans ce cas la translation sera vers le sens du rotor qui porte la force de poussée la plus faible [12, 13].

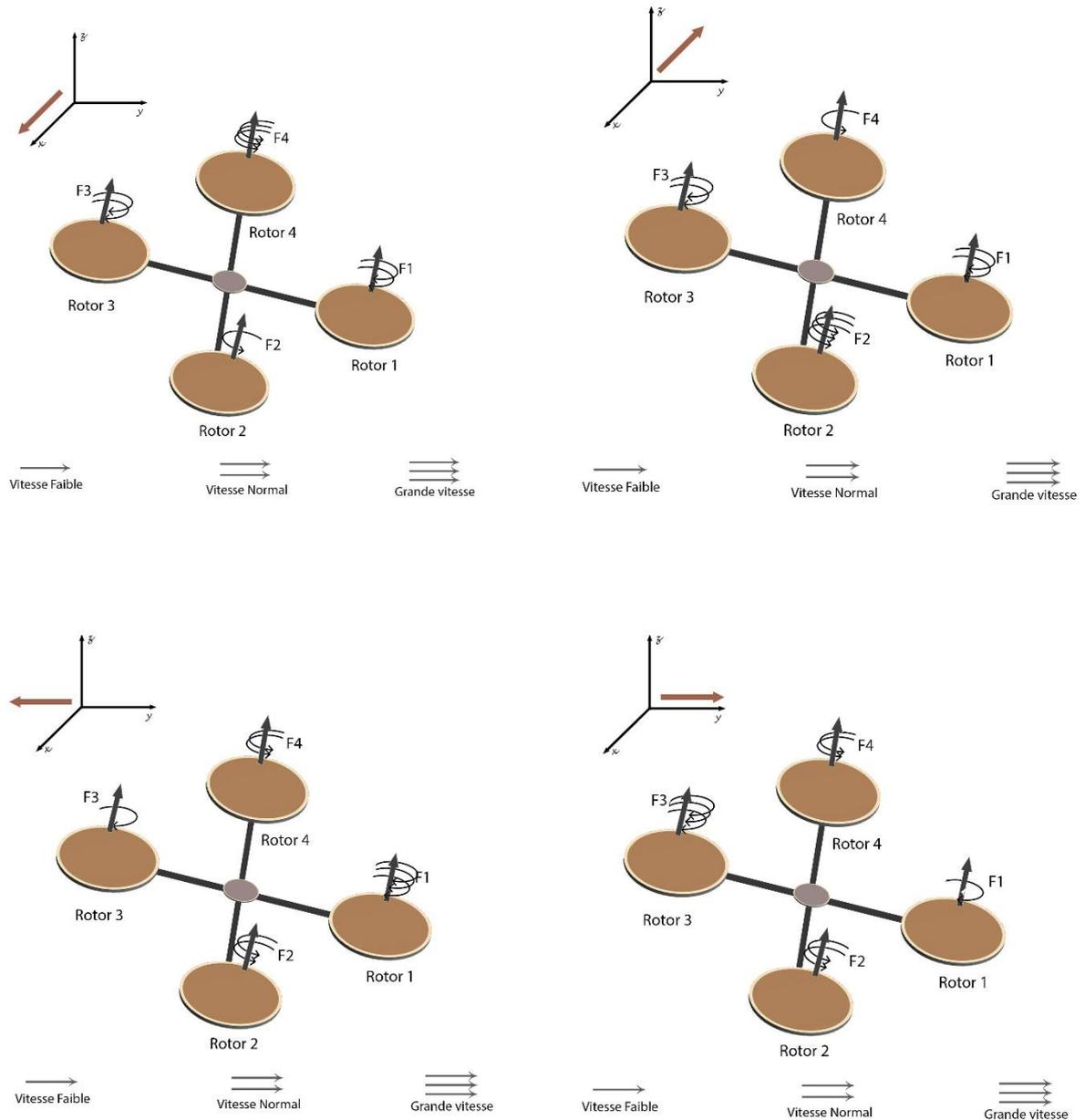


Figure 1.8 : Illustration de la translation horizontale

4.3. Les angles d'Euler :

Une matrice de rotation orthogonale peut définir la coordination entre les repères. Généralement, les applications robotiques utilisent fréquemment la paramétrisation de la matrice de rotation par les angles d'Euler [14]. Puisqu'on est en présence de forces conservatives (le poids est toujours dans la direction Z du repère fixe de la Terre E (O, X, Y, Z), il est nécessaire d'avoir une matrice de rotation 3D pour les entrainer dans le repère

mobile B (o', x', y', z'), De ce fait les forces seront formulées à l'aide des angles montrer respectivement :

- L'angle du Roulis ϕ tel que $-\frac{\pi}{2} < \phi < \frac{\pi}{2}$
- L'angle de Tangage θ tel que $-\frac{\pi}{2} < \theta < \frac{\pi}{2}$
- L'angle du Lacet ψ .

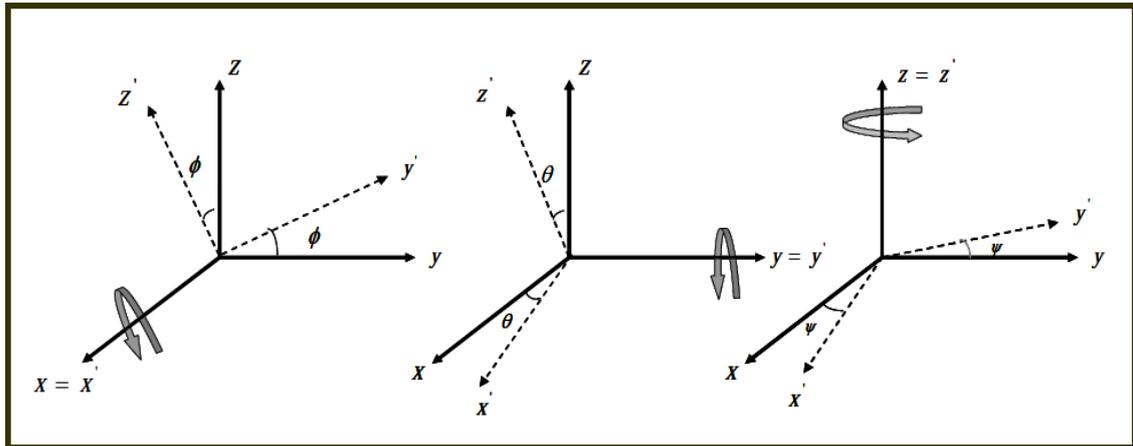


Figure 1.9 : les angles d'Euler

On peut mettre les trois angles sous forme d'une matrice élémentaire et effectuer la multiplication :

$$R = \text{Rot}_x(\psi) * \text{Rot}_y(\theta) * \text{Rot}_z(\phi) \dots\dots\dots (I.1).$$

$$R = \begin{bmatrix} c\psi & -s\psi & 0 \\ s\psi & c\psi & 0 \\ 0 & 0 & 1 \end{bmatrix} \times \begin{bmatrix} c\theta & 0 & s\theta \\ 0 & 1 & 0 \\ -s\theta & 0 & c\theta \end{bmatrix} \times \begin{bmatrix} 1 & 0 & 0 \\ 0 & c\phi & -s\phi \\ 0 & s\phi & c\phi \end{bmatrix} \dots\dots\dots (I.2).$$

$$R = \begin{bmatrix} c\psi c\theta & s\phi s\theta c\psi - s\psi c\phi & c\phi s\theta c\psi + s\psi s\phi \\ s\psi c\theta & s\phi s\theta s\psi + c\psi c\theta & c\phi s\theta s\psi - s\phi c\psi \\ -s\theta & s\phi c\theta & c\phi c\theta \end{bmatrix} \dots\dots\dots (I.3).$$

R : matrice de rotation.

Avec : c = cos et s = sin

4.4. Vitesses angulaires

$\Omega_1, \Omega_2, \Omega_3$, représentent les vitesses angulaires exprimées dans le repère fixe tel que :

$$\Omega = \begin{bmatrix} \Omega_1 \\ \Omega_2 \\ \Omega_3 \end{bmatrix} = \begin{bmatrix} \dot{\phi} \\ 0 \\ 0 \end{bmatrix} + \text{Rot}_x(\phi)^{-1} \begin{bmatrix} 0 \\ \dot{\theta} \\ 0 \end{bmatrix} + (\text{Rot}_y(\theta) \text{Rot}_x(\phi))^{-1} \begin{bmatrix} 0 \\ 0 \\ \dot{\psi} \end{bmatrix} \dots\dots\dots(\text{I.4}).$$

En effet, lorsque la rotation en roulis a lieu, les repères doivent être encore confondus. Puis, concernant le tangage, le vecteur représentant la rotation doit être révélé dans le repère fixe.

De ce fait, il est multiplié par $\text{Rot}_x(\phi)^{-1}$.

De même, le vecteur représentant la rotation en lacet doit être exposé dans le repère fixe qui a déjà subi deux rotations. On arrive ainsi à :

$$\Omega = \begin{bmatrix} \dot{\phi} \\ 0 \\ 0 \end{bmatrix} + \begin{bmatrix} 0 \\ \dot{\theta} c\phi \\ -\dot{\theta} s\phi \end{bmatrix} + \begin{bmatrix} -\dot{\psi} s\theta \\ \dot{\psi} s\phi c\theta \\ \dot{\psi} c\phi c\theta \end{bmatrix} = \begin{bmatrix} \dot{\phi} - \dot{\psi} s\theta \\ \dot{\theta} c\phi + \dot{\psi} s\phi c\theta \\ \dot{\psi} c\phi c\theta - \dot{\theta} s\phi \end{bmatrix} \dots\dots\dots(\text{I.5}).$$

$$\Omega = \begin{bmatrix} 1 & 0 & -s\theta \\ 0 & c\phi & s\phi c\theta \\ 0 & -s\phi & c\phi c\theta \end{bmatrix} \times \begin{bmatrix} \dot{\phi} \\ \dot{\theta} \\ \dot{\psi} \end{bmatrix} \dots\dots\dots(\text{I.6}).$$

Dans le cas où les mouvements angulaires de faibles amplitudes sont accomplis par le quadrirotor, les approximations suivantes sont adoptées :

$$\begin{cases} c\phi = c\theta = c\psi = 1 \\ s\phi = s\theta = s\psi = 0 \end{cases} \dots\dots\dots(\text{I.7}).$$

Le vecteur Ω peut être assimilé à $[\phi, \theta, \psi]^T$.

4.5. Vitesses linéaires

Les vitesses linéaires v_x^b, v_y^b, v_z^b , dans le repère fixe sont exprimées en fonction des vitesses linéaires v_x^m, v_y^m, v_z^m , dans le repère mobile comme :

$$v = \begin{bmatrix} v_x^b \\ v_y^b \\ v_z^b \end{bmatrix} = R \times \begin{bmatrix} v_x^m \\ v_y^m \\ v_z^m \end{bmatrix} \dots\dots\dots(\text{I.8}).$$

R : la matrice de rotation.

4.6. Les efforts physiques agissants sur le quadrirotor

4.6.1. Les force :

Les forces s'appuient sur le système suivant :

- **Le poids du quadri-rotors** : il est défini par

$$P = mg \dots\dots\dots(I.9).$$

m : la masse totale et g : l'accélération de la pesanteur.

- **Les forces de poussée « portance »** :

Ce sont des forces orthogonales sur le plan des hélices, provoquées par la rotation des moteurs et relatives au carrée de la vitesse de rotation des moteurs :

$$F_i = b\omega_i^2 \dots\dots\dots(I.10).$$

Avec $i = \overline{1:4}$ ¹, b : est le coefficient de portance qui dépend de de la forme et le nombre des pales et de la densité de l'air.

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

- ✓ La traînée dans les hélices : elle agit sur les pales (elle est relative à la densité de l'air), la forme des pales et le carrée de la vitesse de rotation de l'hélice, elle est présentée par la relation suivante :

$$T_i = d\omega_i^2 \dots\dots\dots(I.11).$$

Avec $i = \overline{1:4}$, d : est le coefficient de traînée.

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

$$F_t = K_{ft}v \dots\dots\dots(I.12).$$

K_{ft} : Le coefficient de force de traînée de translation.

v : La vitesse linéaire.

4.6.2. Les moments :

Il y a plusieurs moments agissants sur le quadrirotor (elles sont causées par les forces de poussée, de traînée et aux effets gyroscopiques).

- **Moments dus aux forces de poussée des hélices**

¹ i : le numéro de rotor (rotor n° i) Il y a 4 rotors dans un drone quadrirotor.

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

$$M_x = l(F_4 - F_2) = lb(\omega_4^2 - \omega_2^2) \dots\dots\dots(I.13).$$

l : est la distance entre le centre de masse du quadrirotor et l'axe de rotation du rotor.

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

$$M_y = l(F_3 - F_1) = lb(\omega_3^2 - \omega_1^2) \dots\dots\dots(I.14).$$

▪ **Moments dus aux forces de traînée des hélices**

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

$$M_z = d(\omega_1^2 - \omega_2^2 + \omega_3^2 - \omega_4^2) \dots\dots\dots (I.15).$$

Moment résultant des frottements aérodynamiques, il est déterminé par :

$$M_a = K_{fa}\Omega^2 \dots\dots\dots (I.16).$$

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

4.6.3. Effet gyroscopique :

Le mot « gyroscope » se compose de deux mots d'origines grecques : gyro qui signifie rotation et scope qui désigne observer. L'effet gyroscopique est alors désigné en référence au mode de fonctionnement de gyroscope qui indique un appareil de contrôle de mouvement utilisé dans l'aviation, il est donc la difficulté du bouleversement de la position ou la disposition du plan de rotation d'une masse tournante [15].

Dans cette étude, il existe deux moments gyroscopiques, le premier se trouve dans le moment gyroscopique des hélices et l'autre le moment gyroscopique dû aux mouvements de quadrirotor [10].

Moment gyroscopique des hélices : il est défini par la relation suivante :

$$M_{gh} = \sum_{i=1}^4 \Omega \wedge J_r [0 \ 0 \ (-1)^{i+1} \omega_i]^T \dots\dots\dots (I.17).$$

Avec : J_r est l'inertie des rotors.

Moment gyroscopique dû aux mouvements de quadrirotor, il est déterminé par la relation suivante :

$$M_{gm} = \Omega \wedge J. \Omega \dots\dots\dots (I.18).$$

Avec : J est l'inertie du système.

4.7. Modèle mathématique du quadrirotor selon Newton-Euler :

Après avoir décrit les différentes équations, nous pouvons maintenant élaborer le modèle mathématique en utilisant la formulation de Newton-Euler, les équations sont alors notées sous la forme suivante :

$$\begin{cases} \dot{\zeta} = v \\ m\ddot{\zeta} = F_f + F_t + F_g \\ R = RS(\Omega) \\ J\dot{\Omega} = -\Omega \wedge J\Omega + M_f - M_a - M_{g_h} \end{cases} \dots\dots\dots(I.19).$$

Avec :

ζ : Est le vecteur de position du quadrirotor $\zeta=[x \ y \ z]^T$,

m : la masse totale du quadrirotor,

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

$$J = \begin{bmatrix} I_x & 0 & 0 \\ 0 & I_y & 0 \\ 0 & 0 & I_z \end{bmatrix} \dots\dots\dots(I.20).$$

Ω : La vitesse angulaire exprimée dans le repère inertiel « fixe».

R : La matrice de rotation,

\wedge : Le produit vectoriel,

$S(\Omega)$: est la matrice antisymétrique ; pour un vecteur de vitesse $\Omega = [\Omega_1 \ \Omega_2 \ \Omega_3]^T$, elle est définie comme suit :

$$S(\Omega) = \begin{bmatrix} 0 & -\Omega_3 & \Omega_2 \\ \Omega_3 & 0 & -\Omega_1 \\ -\Omega_2 & \Omega_1 & 0 \end{bmatrix} \dots\dots\dots(I.21).$$

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

$$F_f = R \times [0 \ 0 \ \sum_{i=1}^4 F_i]^T \dots\dots\dots(I.22).$$

$$F_i = b\omega_i^2 \dots\dots\dots(I.23).$$

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

$$F_t = \begin{bmatrix} -K_{ftx} & 0 & 0 \\ 0 & -K_{fty} & 0 \\ 0 & 0 & -K_{ftz} \end{bmatrix} \dot{\zeta} \dots\dots\dots(I.24).$$

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

F_g : force de gravité, elle est donnée par :

$$F_g = \begin{bmatrix} 0 \\ 0 \\ -mg \end{bmatrix} \dots\dots\dots(I.25).$$

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

$$M_f = \begin{bmatrix} I(F_4 - F_2) \\ I(F_3 - F_1) \\ d(\omega_1^2 - \omega_2^2 + \omega_3^2 - \omega_4^2) \end{bmatrix} \dots\dots\dots(I.26).$$

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

$$M_a = \begin{bmatrix} K_{fax}\dot{\phi}^2 \\ K_{fay}\dot{\theta}^2 \\ k_{faz}\dot{\psi}^2 \end{bmatrix} \dots\dots\dots(I.27).$$

K_{fax} K_{fay} K_{faz} : Les coefficients des frottements aérodynamiques.

4.7.1. Equations de mouvement de translation :

On a:

$$m \ddot{\zeta} = F_f + F_t + F_g \dots\dots\dots(I.28).$$

$$m \begin{bmatrix} \ddot{x} \\ \ddot{y} \\ \ddot{z} \end{bmatrix} = \begin{bmatrix} c\phi c\psi s\theta + s\phi s\psi \\ c\phi s\theta s\psi - s\phi c\psi \\ c\phi c\theta \end{bmatrix} \sum_{i=1}^4 F_i - \begin{bmatrix} K_{ftx}\dot{x} \\ K_{fty}\dot{y} \\ K_{ftz}\dot{z} \end{bmatrix} - \begin{bmatrix} 0 \\ 0 \\ mg \end{bmatrix} \dots\dots\dots(I.29).$$

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

$$\begin{cases} \ddot{x} = \frac{1}{m} (c\phi c\psi s\theta + s\phi s\psi) [\sum_{i=1}^4 F_i] - \frac{K_{ftx}}{m} \dot{x} \\ \ddot{y} = \frac{1}{m} (c\phi s\theta s\psi - s\phi c\psi) [\sum_{i=1}^4 F_i] - \frac{K_{fty}}{m} \dot{y} \dots\dots\dots (I.30). \\ \ddot{z} = \frac{1}{m} (c\phi c\theta) [\sum_{i=1}^4 F_i] - \frac{K_{ftz}}{m} \dot{z} - g \end{cases}$$

4.7.2. Equations de mouvement de rotation :

On a:

$$J\dot{\Omega} = -\Omega \wedge J\Omega + M_f - M_a - M_{gh} \dots\dots\dots(I.31).$$

Si on remplace par les termes par ces équations on obtient la forme suivante :

$$\begin{bmatrix} I_x & 0 & 0 \\ 0 & I_y & 0 \\ 0 & 0 & I_z \end{bmatrix} \begin{bmatrix} \ddot{\phi} \\ \ddot{\theta} \\ \ddot{\psi} \end{bmatrix} = - \begin{bmatrix} \dot{\phi} \\ \dot{\theta} \\ \dot{\psi} \end{bmatrix} \wedge \begin{bmatrix} I_x & 0 & 0 \\ 0 & I_y & 0 \\ 0 & 0 & I_z \end{bmatrix} \begin{bmatrix} \dot{\phi} \\ \dot{\theta} \\ \dot{\psi} \end{bmatrix} - \begin{bmatrix} K_{fax}\dot{\phi}^2 \\ K_{fay}\dot{\theta}^2 \\ K_{faz}\dot{\psi}^2 \end{bmatrix} \\ + \begin{bmatrix} lb(\omega_4^2 - \omega_2^2) \\ lb(\omega_3^2 - \omega_1^2) \\ d(\omega_1^2 - \omega_2^2 + \omega_3^2 - \omega_4^2) \end{bmatrix} - \begin{bmatrix} J_r\Omega_r\dot{\theta} \\ -J_r\Omega_r\dot{\phi} \\ 0 \end{bmatrix} \dots\dots\dots(I.31).$$

Ça conduit aux équations différentielles définissant le mouvement de rotation :

$$\begin{cases} I_x\ddot{\phi} = -\dot{\theta}\dot{\psi}(I_z - I_y) - K_{fax}\dot{\phi}^2 - J_r\Omega_r\dot{\theta} + lb(\omega_4^2 - \omega_2^2) \\ I_y\ddot{\theta} = \dot{\phi}\dot{\psi}(I_z - I_x) - k_{fay}\dot{\theta}^2 + J_r\Omega_r\dot{\phi} + lb(\omega_3^2 - \omega_1^2) \dots\dots\dots (I.32). \\ I_z\ddot{\psi} = \dot{\phi}\dot{\theta}(I_x - I_y) - K_{faz}\dot{\psi}^2 + d(\omega_1^2 - \omega_2^2 + \omega_3^2 - \omega_4^2) \end{cases}$$

Notant que u_1 u_2 u_3 u_4 sont les entrées de commande (de translation selon l'axe «x, y, z» et la rotation selon lacet) du système et qui s'écrivent en fonction des vitesses angulaires des quatre rotors comme suit [17] :

$$\begin{bmatrix} u_1 \\ u_2 \\ u_3 \\ u_4 \end{bmatrix} = \begin{bmatrix} b & b & b & b \\ 0 & -lb & 0 & lb \\ -lb & 0 & lb & 0 \\ d & -d & d & -d \end{bmatrix} = \begin{bmatrix} \omega_1^2 \\ \omega_2^2 \\ \omega_3^2 \\ \omega_4^2 \end{bmatrix} \dots\dots\dots(I.33).$$

$$\text{Et : } \Omega_r = \omega_1 - \omega_2 + \omega_3 - \omega_4$$

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

$$\left\{ \begin{array}{l} \ddot{\phi} = \frac{(I_y - I_z)}{I_x} \dot{\theta} \dot{\psi} - \frac{J_r}{I_x} \Omega_r \dot{\theta} - \frac{K_{fax}}{I_x} \dot{\phi}^2 + \frac{1}{I_x} u_2 \\ \ddot{\theta} = \frac{(I_z - I_x)}{I_y} \dot{\phi} \dot{\psi} - \frac{J_r}{I_y} \Omega_r \dot{\phi} - \frac{K_{fay}}{I_y} \dot{\theta}^2 + \frac{1}{I_y} u_3 \\ \ddot{\psi} = \frac{(I_x - I_y)}{I_z} \dot{\theta} \dot{\phi} - \frac{K_{faz}}{I_z} \dot{\psi}^2 + \frac{1}{I_z} u_4 \\ \ddot{x} = -\frac{K_{ftx}}{m} \dot{x} + \frac{1}{m} u_x u_1 \\ \ddot{y} = -\frac{K_{fty}}{m} \dot{y} + \frac{1}{m} u_y u_1 \\ \ddot{z} = -\frac{K_{ftz}}{m} \dot{z} - g + \frac{(c\phi c\theta)}{m} u_1 \end{array} \right. \dots\dots\dots (I.34).$$

Avec :

$$\begin{cases} u_x = (c\phi c\psi s\theta + s\phi s\psi) \\ u_y = (c\phi s\theta s\psi - s\phi c\psi) \end{cases} \dots\dots\dots (I.35).$$

4.7.3. Dynamique du rotor [12] :

Le rotor est un ensemble d'élément constitué d'un moteur à courant continu entraînant une hélice via un réducteur ; le moteur à courant continu est régi par les équations dynamiques suivantes :

$$\left\{ \begin{array}{l} V = ri + L \frac{di}{dt} + k_e \omega \\ k_m i = J_r \frac{d\omega}{dt} + C_s + k_r \omega^2 \end{array} \right. \dots\dots\dots(I.36).$$

Les différents paramètres du moteur sont définis comme suit :

V : est la tension d'entrée du moteur.

ω_i : est la vitesse angulaire du rotor.

k_e , k_m : représentent respectivement les constantes des couples électrique et mécanique.

k_r : Est la constante du couple de charge.

r, L : La résistance du moteur et l'inductance.

J_r : L'inertie du rotor.

C_s : représente le frottement sec.

Le modèle choisis pour le rotor est le suivant :

$$\dot{\omega}_i = bV_i - \beta_0 - \beta_1\omega_i - \beta_2\omega_i^2 \quad i \in [1,4] \dots\dots\dots (I.37).$$

Avec :

$$\beta_0 = \frac{c_s}{J_r}, \beta_1 = \frac{k_e k_m}{r J_r}, \beta_2 = \frac{k_r}{J_r} \quad \text{et} \quad b = \frac{k_m}{r J_r} \dots\dots\dots (I.38).$$

4.8. La représentation d'état² du système [12] :

Il existe de multiples représentations d'état pour un système physique, dans le cas de cette étude, on considère $X = [X_1 \dots X_{12}]^T$ comme vecteur d'état du système :

Soit :

$$\underline{X} = [\phi \quad \dot{\phi} \quad \theta \quad \dot{\theta} \quad \psi \quad \dot{\psi} \quad x \quad \dot{x} \quad y \quad \dot{y} \quad z \quad \dot{z}]^T \dots\dots\dots (I.39).$$

On obtient la représentation d'état sous la forme $\dot{X} = f(X) + g(X, u)$:

$$\left\{ \begin{array}{l} \dot{x}_1 = x_2 \\ \dot{x}_2 = f_1(X) + g_1(X)u_2 \\ \dot{x}_3 = x_4 \\ \dot{x}_4 = f_2(X) + g_2(X)u_3 \\ \dot{x}_5 = x_6 \\ \dot{x}_6 = f_3(X) + g_3(X)u_4 \\ \dot{x}_7 = x_8 \\ \dot{x}_8 = f_4(X) + g_4(X)u_x \\ \dot{x}_9 = x_{10} \\ \dot{x}_{10} = f_5(X) + g_5(X)u_y \\ \dot{x}_{11} = x_{12} \\ \dot{x}_{12} = f_6(X) + g_6(X)u_1 \end{array} \right. \dots\dots\dots (I.40).$$

Avec :

$$\begin{array}{ll} f_1(X) = \frac{(I_y - I_x)}{I_x} x_4 x_6 - \frac{I_r}{I_x} \Omega_r x_4 - \frac{K_{fax}}{I_x} x_2^2 & , g_1(X) = \frac{1}{I_x} \\ f_2(X) = \frac{(I_z - I_x)}{I_y} x_2 x_6 - \frac{I_r}{I_y} \Omega_r x_2 - \frac{K_{fay}}{I_y} x_4^2 & , g_2(X) = \frac{1}{I_y} \\ f_3(X) = \frac{(I_x - I_y)}{I_z} x_2 x_4 - \frac{K_{faz}}{I_z} x_6^2 & , g_3(X) = \frac{1}{I_z} \\ f_4(X) = -\frac{K_{ftx}}{m} x_8 & , g_4(X) = \frac{u_1}{m} \\ f_5(X) = -\frac{K_{fty}}{m} x_{10} & , g_5(X) = \frac{u_1}{m} \\ f_6(X) = -\frac{K_{ftz}}{m} x_{12} - g & , g_6(X) = \frac{\cos \phi \cos \theta}{m} \end{array}$$

² Etat : est une modélisation d'un système dynamique sous forme matricielle en utilisant des variables d'état (on l'appelle aussi vecteur d'état).

5. Les fonctions de transformations :

Un système de control se compose d'une sortie ainsi qu'une entrée de signal.

La sortie est liée à l'entrée via une fonction appelé fonction de transformation. Cette fonction est représentée par un bloc dans le diagramme du système de control.

Pour chaque système de control il existe une entrée de référence appelé excitation qui sera transformé via une opération de transformation et produit un effet résultant appelé sortie ou réponse contrôlée [18].



$$G(s) = \frac{C(s)}{R(s)} \Rightarrow R(s).G(s) = C(s) \dots\dots\dots(I.41).$$

Ce n'est pas nécessaire que les entrées/sorties soient de même catégorie par exemple dans un moteur électrique l'entrée est un signal électrique tandis que la sortie est un signal mécanique puisque l'énergie électrique est nécessaire pour faire tourner les moteurs.

Mais pour l'analyse mathématique d'un système toutes sortes de signaux devraient être représentés sous une forme similaire, cela se fait en transformant toutes sortes de signaux en leur forme de Laplace.

De même, la fonction de transfert d'un système est représentée par la forme de Laplace qui est un rapport de la sortie de la transformé de Laplace à l'entrée de la transformé de Laplace en supposant que toutes les conditions initiales soient nulles [18].



$$Where, R(s) = \mathcal{L}r(t), C(s) = \mathcal{L}c(t) \& G(s) = \frac{\mathcal{L}c(t)}{\mathcal{L}r(t)}$$

6. Loi de commande par mode glissant :

Les techniques de commande non linéaire nécessitent une stratégie de commande reposé fondamentalement sur la décomposition du système d'origine en deux sous-systèmes : le premier examine la commande en position cependant le deuxième concerne la commande en orientation. Le présent schéma montre la stratégie de commande adoptée [10].

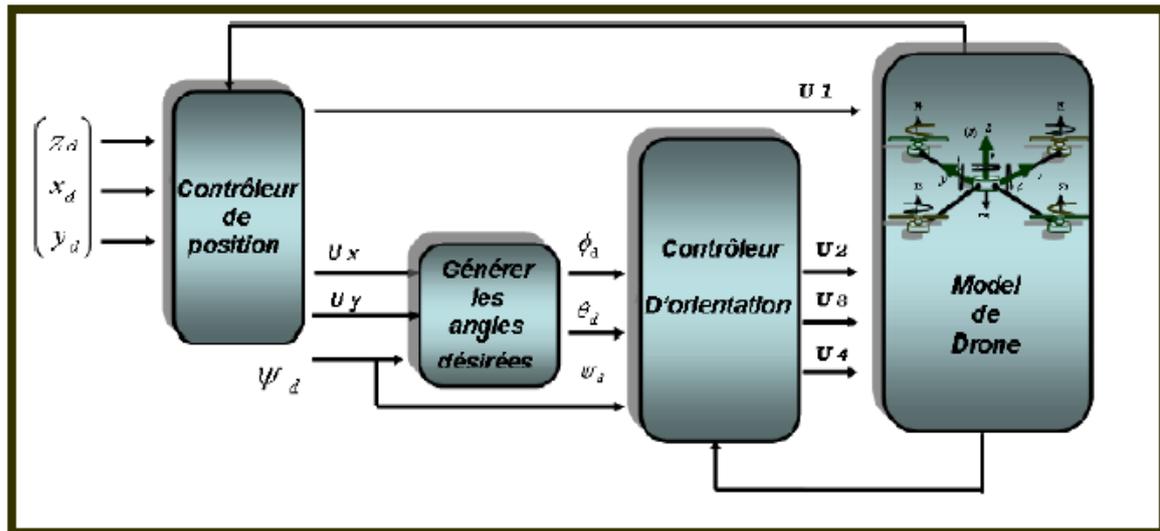


Figure I.10 : Schéma synoptique de la commande proposée [10].

Ce travail est une implémentation d'une loi de commande dite « par mode glissant » dont le choix, les justifications et les fondements se trouvent dans [4].

La commande par mode glissant est utilisée pour contrôler les systèmes linéaires ainsi que les systèmes non linéaires en robotique, en aviation militaire, en milieu sous-marin ... etc. Cette commande est caractérisée par la discontinuité aux passages par la surface de glissement. Elle est utilisée pour donner plus de robustesse aux systèmes non linéaires ayant des paramètres variantes [10].

1.1.Régime glissant idéal :

Théoriquement, il est supposé que « l'organe de commutation »³ est insensible aux perturbations, la trajectoire suivie par le quadricopteur en régime glissant décrit exactement l'équation $S(x)=0$. Le point représentatif de l'évolution du système trace un glissement parfait sur la surface globale de commutation [19].

³ L'ensemble des capteurs et dispositifs qui réagissent avec l'environnement et fournissent des informations au système.

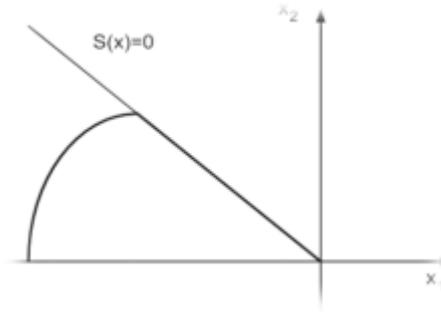


Figure 1.11 : Glissement idéal.

1.2.Régime glissant réel :

Pratiquement, l'organe de commutation est réalisé à partir des relais qui présentent des imperfections comme les retards de commutations, dans ce cas, la trajectoire de phase du régime glissant reste au voisinage de la surface de commutation donnant naissance à des oscillations indésirables qui éliminent la précision du système et néanmoins sa stabilité (génère des perturbations) [19].

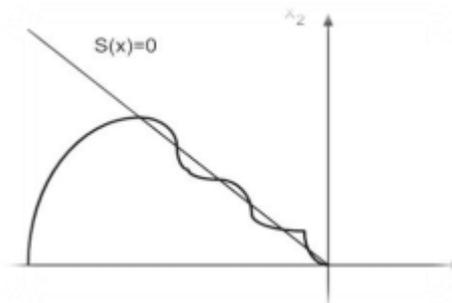


Figure 1.12 : Glissement réel.

1.3.Conception de la commande par mode glissant :

Soit le système définit par l'équation suivante :

$$x^{(n)} = f(x, t) + g(x, t)u + w(x, t) \dots \dots \dots (I.42).$$

Où :

$X(t) \in \mathcal{R}^n$: représente le vecteur d'état, tel que $x(t) = (x, \dot{x}, \dots, x^{n-1})$.

$u(t) \in \mathcal{R}^m$: Vecteur de commande.

$w(x, t) \in \mathcal{R}^n$: Vecteur de perturbation.

$f(x, t) \in \mathfrak{R}^{n \times n}$ et $g(x, t) \in \mathfrak{R}^{n \times m}$ sont des fonctions non linéaires du vecteur d'état décrivant le système.

Le but du contrôle en poursuite est de trouver une loi de commande telle que (étant donné une trajectoire désirée $x_d(t)$) l'erreur de poursuite tend vers zéro malgré la présence des perturbations. L'erreur de poursuite est décrite par l'écart :

$$e(t) = x(t) - x_d(t). \dots\dots\dots(I.43).$$

La mise en œuvre d'une commande par mode glissant passe par trois étapes [19] :

- Le choix de la surface de glissement.
- Conditions d'existence et de convergence.
- L'établissement de la loi de commande.

6. Conclusion :

Dans ce chapitre nous avons découvert que Les drones est un domaine très vaste se base sur la discipline de la robotique, s'évolue rapidement et les différentes catégories des drones existants prouve cette énorme évolution.

La réalisation de ce genre de projet nécessite principalement une modélisation de sa dynamique pour étudier les différents mouvements qu'un drone peut fait avant entamer la phase de l'implémentation d'une loi de commande qui permet la manipulation de ces mouvements en se basent sur des modèles mathématiques.

Enfin nous avons donné une idée sur les fonctions de transformations que nous allons utiliser plus tard.

Chapitre II

Modélisation SysML

1. Introduction :

La réalisation d'un projet nécessite une étape fondamentale dite « conception et modélisation ». Dans un projet de pilotage d'un drone quadrirotor notre modélisation portera sur un système entier avec toutes ses composantes, physique (composants physique), informatique (applications) ou électronique (capteurs). Or les outils usuels de modélisation en informatique sont incomplets donc inefficaces face à cet étalage de composantes ; pour cela il faut utiliser un langage qui est plus adapté à notre besoin.

Dans ce chapitre nous allons présenter une modélisation du système étudié en utilisant le langage SysML.

2. Qu'est-ce que SysML ?

Le langage SysML (System Modelling Language) est une technologie basée sur UML2.0 (Unified Modelling Language) dédié à la modélisation des systèmes, et permettant une bonne planification pour avoir un bon produit final [20].

Pour analyser un système par SysML, il faut d'abord définir le contexte du système étudié.

Dans le cas de notre système, deux sous-systèmes seront étudiés :

- ✓ Le Drone lui-même.
- ✓ La loi de commande à implémenter.

Ensuite, on doit identifier les acteurs principaux et secondaires en précisant leurs cas d'utilisations et en finissant par les autres diagrammes SysML (diagramme de séquences, diagramme de blocks, etc.)

3. Pourquoi SysML ? [20]

SysML est utilisé dans l'ingénierie des systèmes à cause des arguments suivants :

- ❖ SysML est à l'ingénierie des systèmes ce qu'est UML aux systèmes informatiques.
- ❖ Pour définir un système, SysML permet la collaboration autour d'un modèle commun par des acteurs de corps de métier différents, il partage des spécifications d'un système complexe entre tous les corps de métiers.
- ❖ SysML permet la simplification de projets complexes.

- ❖ SysML est un langage qui rend les systèmes évolutifs et maintenables en Identifiant tous les risques liés au projet et en créant une base d'analyse commune à tous les participants d'un projet.
- ❖ SysML est un profil d'UML 2 apporté une contribution dans l'évolution de la modélisation par la résolution des problématiques d'informatique.

La figure ci-dessous (II.1) représente les différents diagrammes SysML :

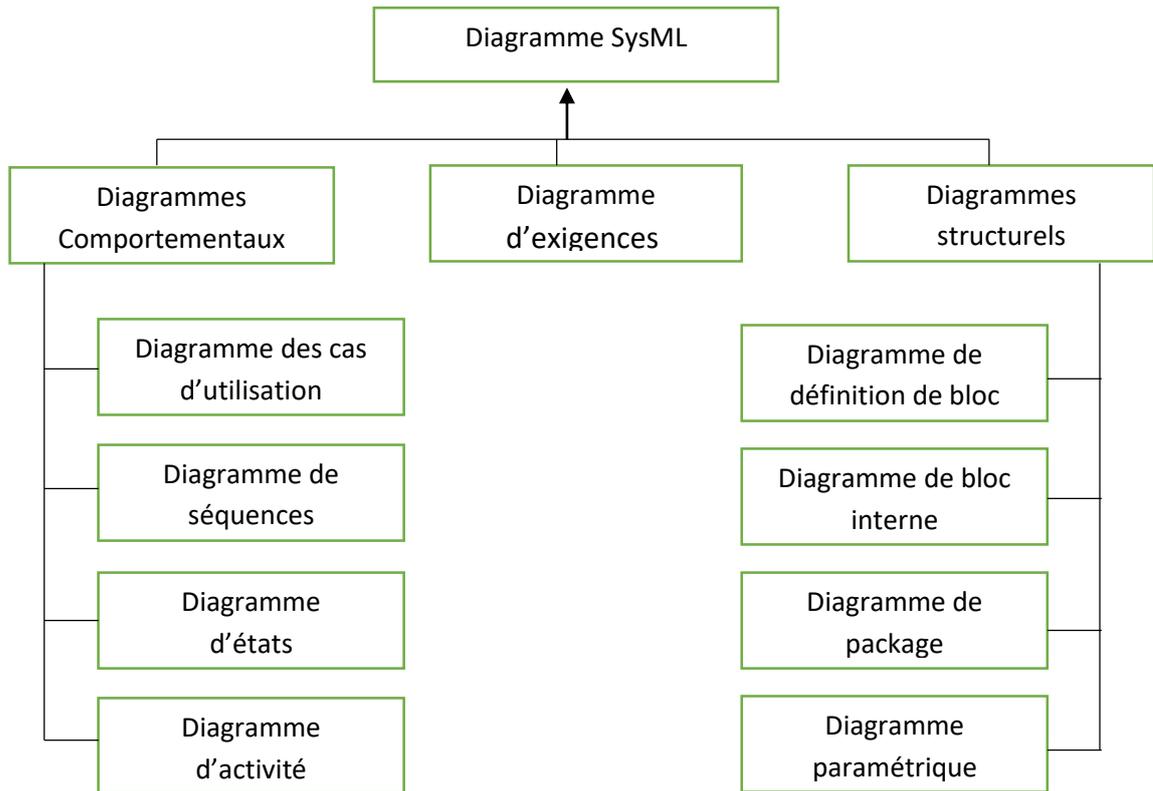


Figure II.1 : Les différents diagrammes SysML.

4. Diagramme d'exigences :

Une exigence spécifie une capacité ou une condition qui doit (ou devrait) être satisfaite. Dans le cas d'une modélisation d'un système, une exigence spécifie un niveau de performance qu'une fonctionnalité de ce système doit atteindre (en termes de sécurité, fiabilité, etc.). Un diagramme des exigences permet de hiérarchiser et de décrire les exigences d'un système sous la forme de graphiques, tableaux ou d'arborescences.

On retrouve aussi des exigences dans d'autres diagrammes SysML afin de les mettre en relation avec d'autres éléments de modélisation (cas d'utilisation, bloc, etc.).

On peut se baser dans la composition de notre diagramme sur deux éléments essentiels de tous les produits :

- La performance.
- La fiabilité des résultats.

Il existe deux autres éléments qui ne nous intéressent pas dans notre étude, ce sont Le coût et La maintenabilité.

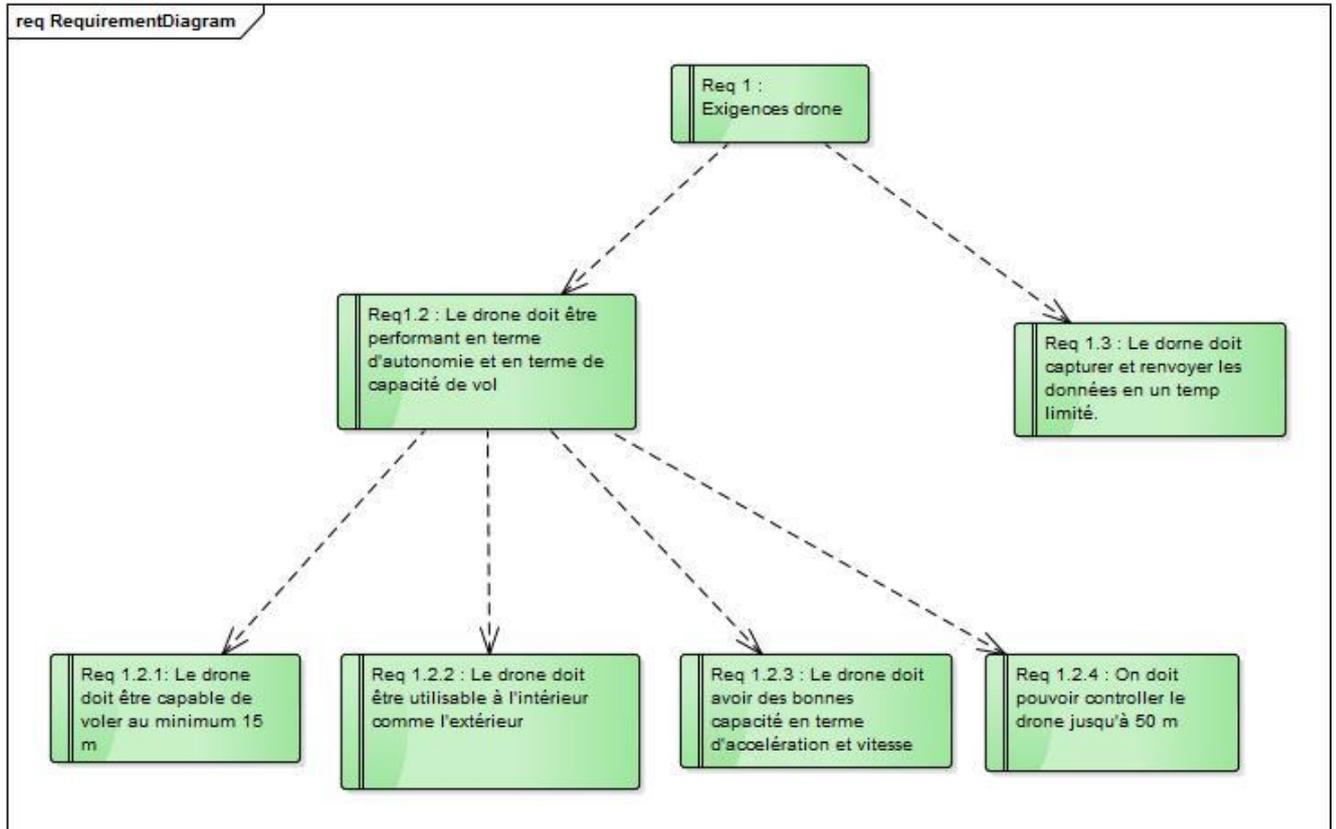


Figure II.2 : Diagramme d'exigences.

5. Diagramme de cas d'utilisations :

5.1. Recherche des cas d'utilisation :

Un cas d'utilisation représente une fonctionnalité du système pour un acteur. Il est primordial de se placer du point de vue des acteurs pour identifier les cas d'utilisation. Une fois le cas d'utilisation identifié, on le nomme avec un verbe à l'infinitif suivi d'un complément. Il faut toujours se rappeler que, pour construire un diagramme de cas d'utilisation, il faut éviter de penser à l'aspect temporel : on ne cherche pas à trouver l'ordre de déclenchement des cas d'utilisation.

5.1.1. Acteur 1 : L'application :

Dans un premier temps on constate que le rôle principal de l'application est de contrôler le drone, on peut le détailler en quatre cas d'utilisations :

- Contrôler le décollage et l'atterrissage.
- Ajuster l'altitude (monter/descendre) et Faire tourner le drone suivant 3 axes.
- Contrôler le déplacement du drone.
- L'équilibrage du fonctionnement des moteurs en cas de panne.

NB : On peut étendre le fonctionnement de notre application en intégrant un module de récupération des données (localisation, mesure de l'énergie, etc.)

5.1.2. Acteur 2 : L'utilisateur « Pilote » :

L'application n'est pas le seul acteur en interaction avec le drone ce qui signifie qu'il existe un nouvel acteur qui peut optionnellement définir la trajectoire à suivre par le drone.

Notre système doit aussi informer l'utilisateur sur l'état du drone (niveau de la batterie, la position et la vitesse, etc.), diapositif d'informations à led situées près des moteurs et sous le drone donnent directement à l'utilisateur les informations les plus importantes sur l'état du système.

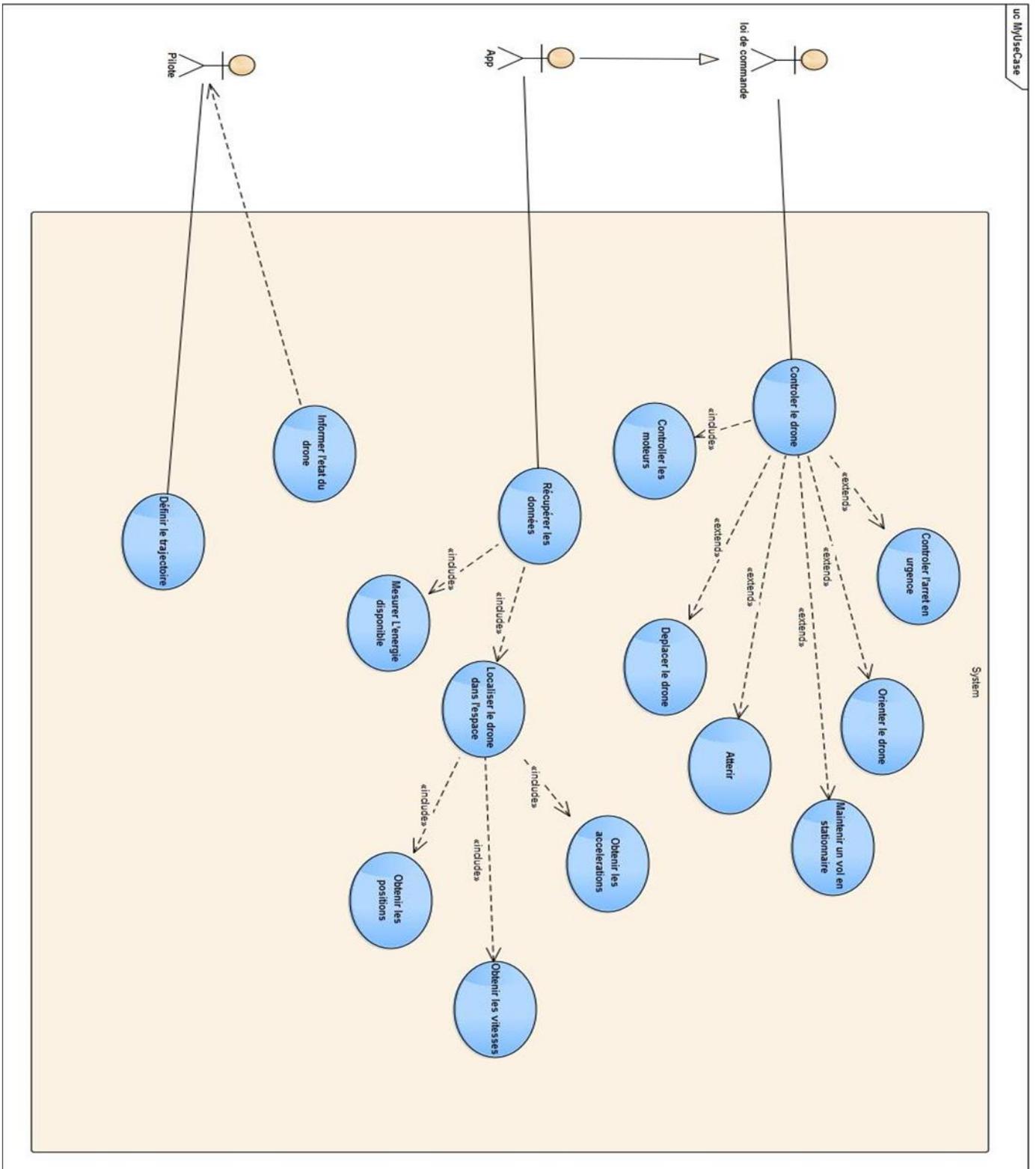


Figure II.3: Diagramme de cas d'utilisations.

6. Diagramme de séquences :

6.1. Diagramme de séquences Auto piloter :

La figure au-dessous montre les différentes interactions entre le Pilote, l'application et le drone lui-même. Les interactions décrites ici sont celles avec les éléments extérieurs. Les détails internes au fonctionnement de drone et de l'application sont masqués. Le diagramme de séquence montre que le pilote peut régler « quelques » paramètres de vol avant le décollage (fragment combiné « opt »), puis plusieurs fonctions s'exécutent en parallèle (fragment combiné « par ») :

- L'application gère le mode d'atterrissage selon 2 cas, une pour les cas d'urgences et la deuxième pour le cas normal.
- L'application diffuse les différentes données essentielles ou demandées par le pilote (les vitesses, les accélérations, l'altitude, l'énergie disponible, etc.)
- Le Pilote peut définir des trajectoires à suivre par le drone grâce à un panneau de configuration visualisé par l'IHM de l'application.

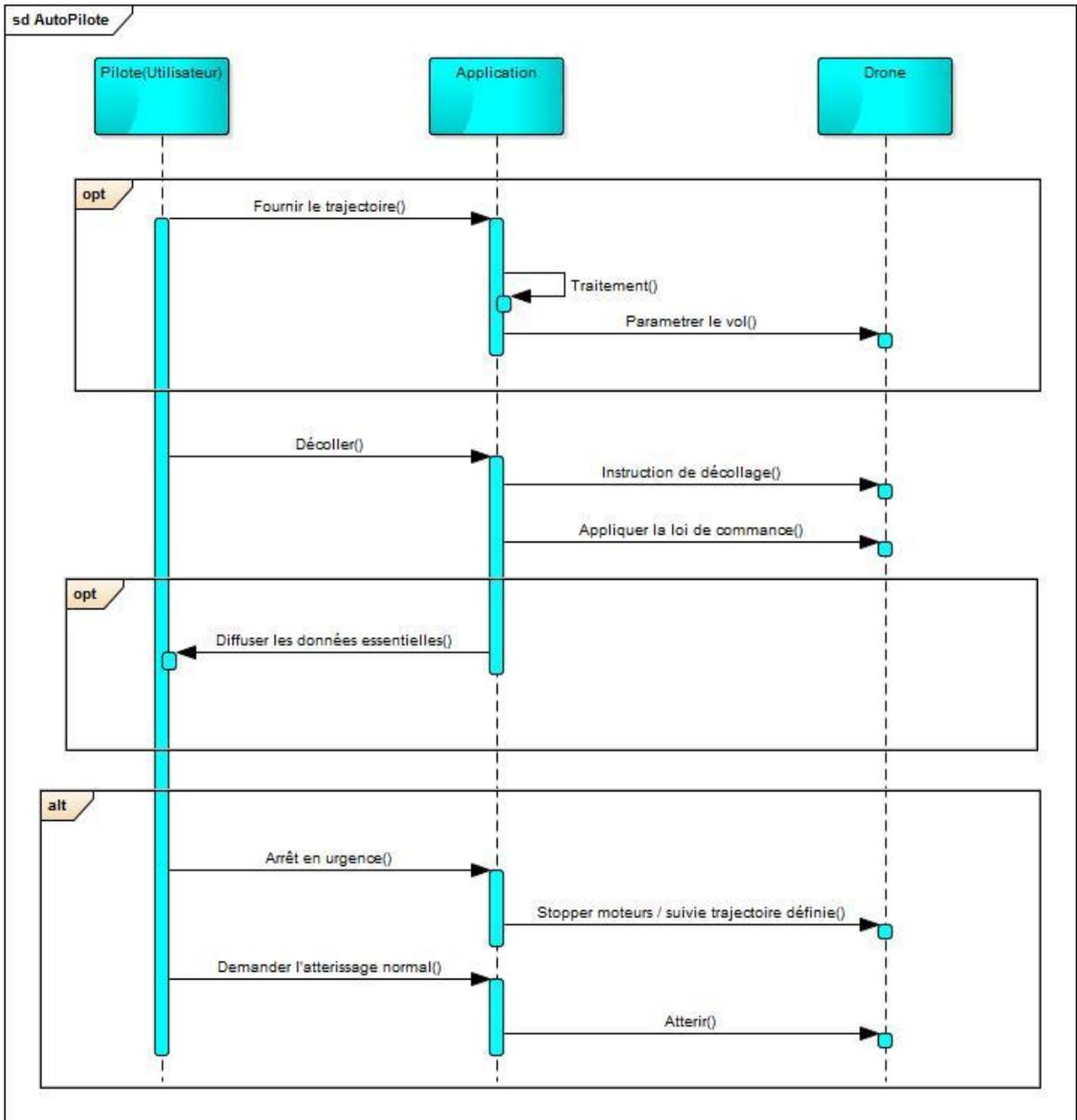


Figure II.4 : Diagramme de séquences Auto piloter

6.2.Diagramme de séquence contrôler :

La figure suivante détaille le cas d'utilisation « contrôler », toujours du point de vue système. Une fois le décollage effectué, plusieurs options s'offrent au Pilote (fragment combiné « par » et « opt ») :

- Faire déplacer le drone (monter/descendre, tourner à droite ou à gauche).
- Effectuer un vol stationnaire.
- En cas de problème, demander un atterrissage d'urgence.

Certaines de ces actions peuvent s'effectuer en parallèle (ex. monter en se déplaçant). Le pilote termine le scénario en demandant l'atterrissage de drone.

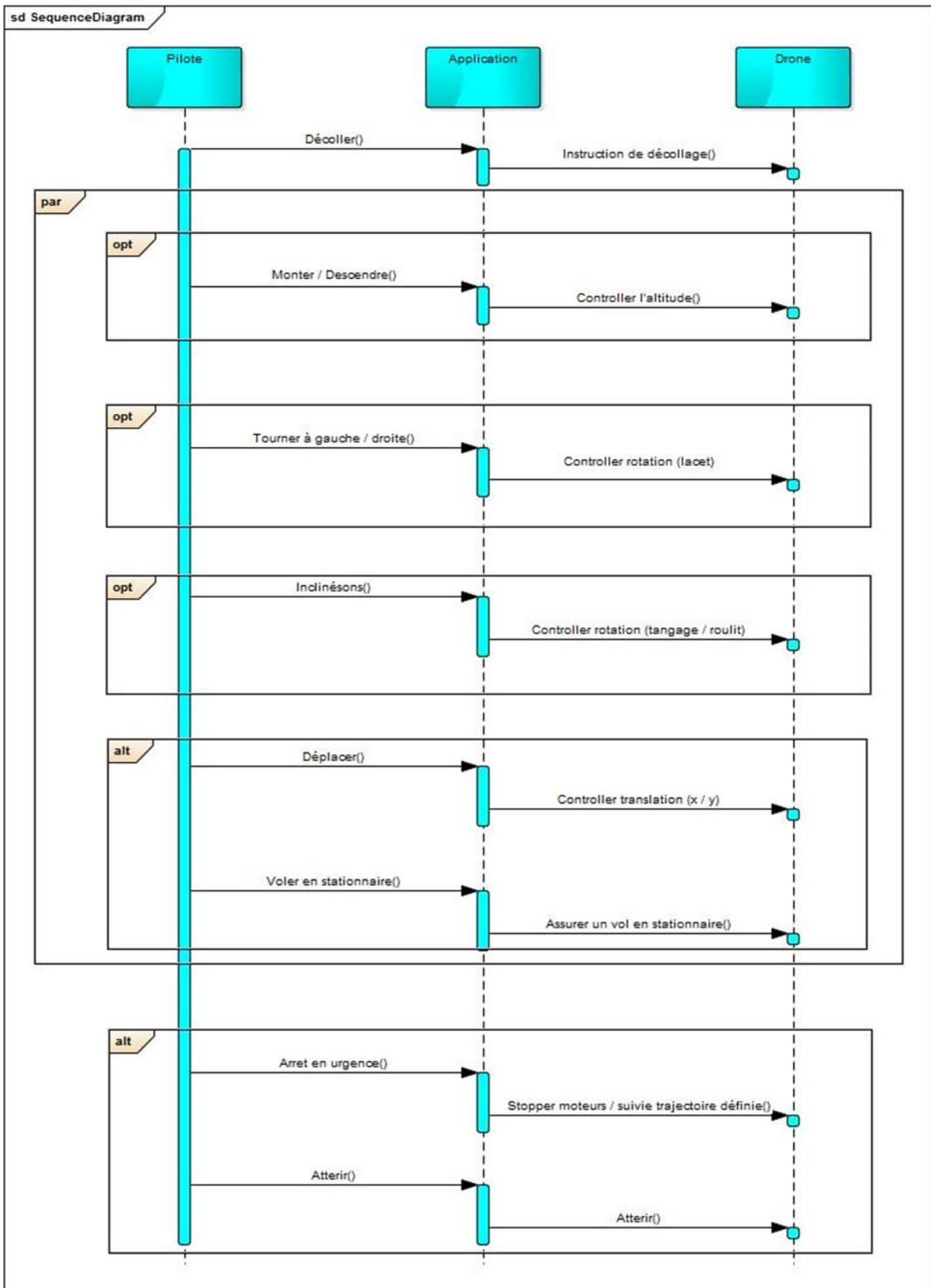


Figure II.5 : Diagramme de séquence pour le cas « contrôler »

7. Diagramme d'états transitions :

Certains comportements complexes d'un système peuvent être modélisés au travers de machines à états finis. Une machine à états décrit le comportement d'un objet en termes d'états et de transitions.

Les activités, qui sont déclenchées durant une transition, en entrée ou en sortie d'un état, sont spécifiées avec les événements et les conditions de garde associé.

Les états complexes peuvent être décrits sous la forme d'états composites. Dans ce cas ils sont composés de plusieurs états pouvant être séquentiels ou concurrents, c'est à dire fonctionner en parallèle.

La figure suivante présente le diagramme d'état transition détaillé pour un pilotage complet :

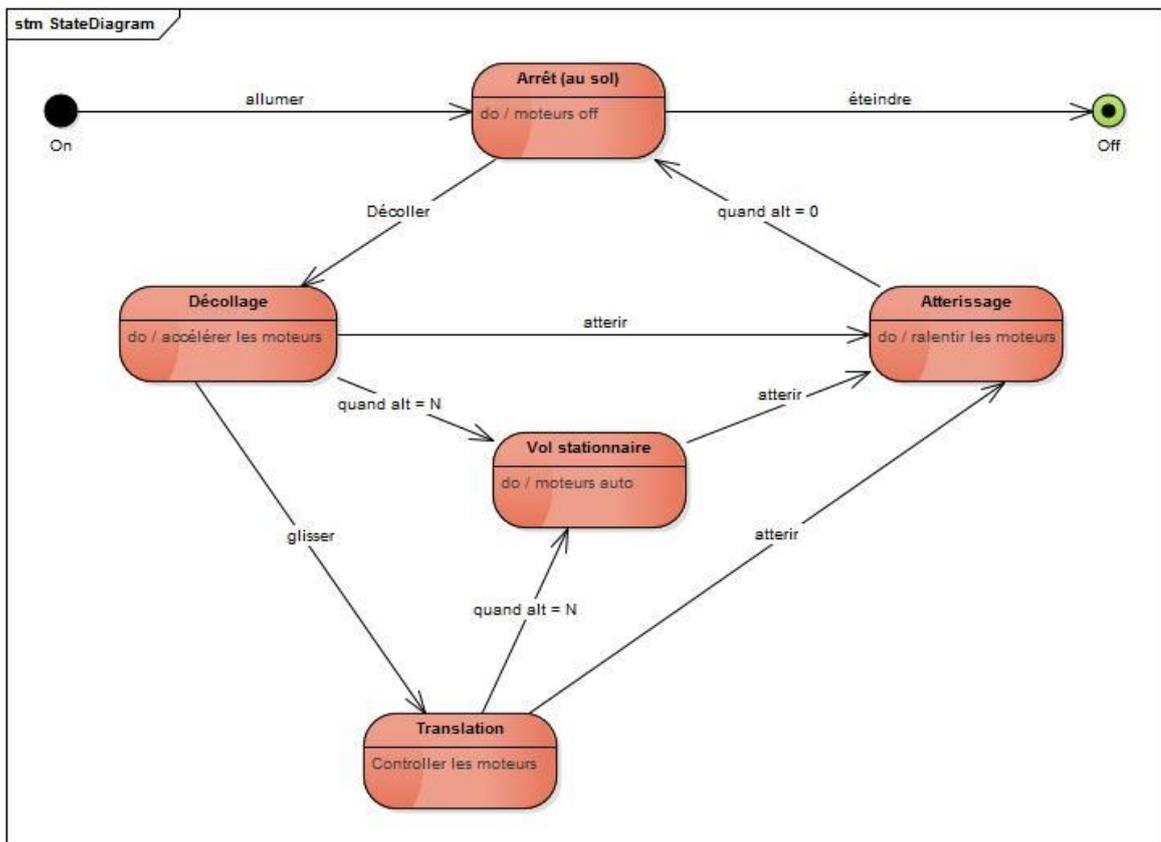


Figure II.6 : Diagramme D'état transition

8. Diagramme de définition de blocs :

Notre système est principalement basé sur le concept de « bloc » proposé par l'architecture SysML.

8.1.Qu'est-ce qu'un bloc ?

Pour réaliser une décomposition logique ou physique d'un système ou un projet complexe, un bloc est un composant unitaire dans un diagramme de blocs, il pourra être de nature logicielle (une classe, une bibliothèque, etc.), matérielle (sous-système électronique ou mécanique, etc.) ou même humaine comme des acteurs.

8.2.Diagramme de notre système :

On peut trouver dans un bloc les parties qui composent celui-ci, les différentes opérations (comportements) possibles sur ce bloc et des valeurs qui permettent de caractériser un bloc.

Le diagramme suivant montre que le système « le drone » est composé de :

- Une carcasse du drone.
- Une application qui commande le drone.
- Un pack Énergie composé d'au moins une batterie, d'un transformateur et d'un chargeur.

Pour que le système soit opérationnel, il doit aussi y avoir un pilote (Utilisateur) et un environnement qui pourra influencer le comportement du drone (vent, sol, arbres, etc.).

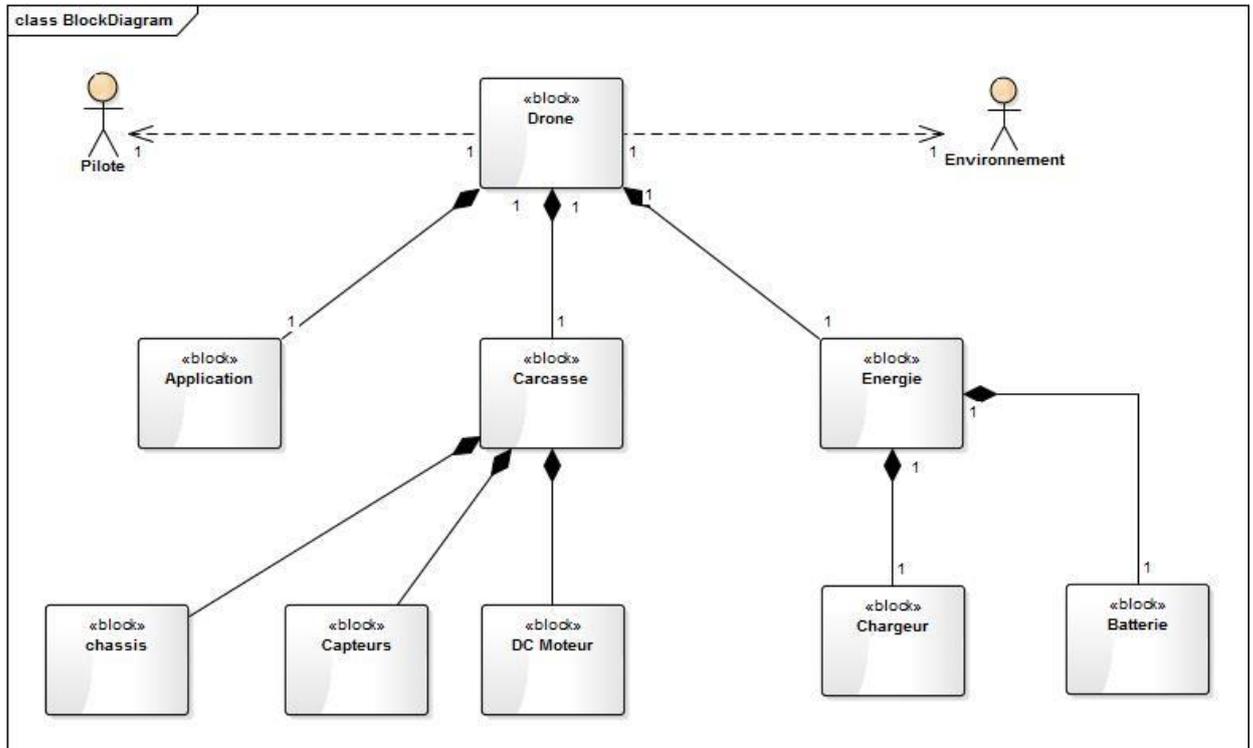


Figure II.7 : Diagramme de définition de blocs

9. Conclusion :

Dans ce chapitre on a abordé tout d'abord la modélisation d'un système de pilotage d'un drone quadrirotor. On a commencé par une première approche théorique en définissant la technologie utilisée qui est représenté par le langage SysML, les objectifs de ce langage, la structure poursuivie pour la réalisation d'un modèle.

En deuxième lieu, on a détaillé la structure suivie en réalisant les différents diagrammes SysML tel que diagramme d'exigences, diagramme de cas utilisation, diagramme de séquence, diagramme d'états transition, diagramme de blocs ... etc. Ces derniers nous ont donné une bonne planification pour réaliser un produit final fiable et efficace.

Chapitre III

Application développée et résultats de simulation

1. Introduction :

Comme une dernière étape, la réalisation de notre projet nécessite l'implémentation de la loi de commande. Pour cela nous avons choisi de créer une bibliothèque en langage C qui contient tous les fonctions nécessaires pour le control de drone. Ce choix arrive comme une tentative de rédiger un code optimal pour être embarqué et réutilisable. Dans ce chapitre nous allons présenter les résultats de simulation obtenus par l'application ainsi que les technologies utilisées pour la réalisation de ce projet et enfin nous allons donner une vision sur l'IHM qui permet aux utilisateurs la manipulation des trajectoires facilement.

2. Résultat de simulation :

2.1. Les valeurs des paramètres du modèle quadrirotor utilisé :

Paramètre	Valeurs
M	0.486 kg
G	9.806 m/s ²
L	0.25 m
B	$2,9842 \times 10^{-5}$ N/rad/s
D	$3,2320 \times 10^{-7}$ N.m/rad/s
J _r	$2,8385 \times 10^{-5}$ kg.m ²
I _x	$3,8278 \times 10^{-3}$ kg.m ²
I _y	$3,8288 \times 10^{-3}$ kg.m ²
I _z	$7,6566 \times 10^{-3}$ kg.m ²
K _{fax}	$5,5670 \times 10^{-4}$ N/rad/s
K _{fay}	$5,5670 \times 10^{-4}$ N/rad/s
K _{faz}	$6,3540 \times 10^{-4}$ N/rad/s
K _{ftx}	$5,5670 \times 10^{-4}$ N/m/s

K_{fty}	$5,5670 \times 10^{-4} \text{ N/m/s}$
K_{ftz}	$6,3540 \times 10^{-4} \text{ N/m/s}$
K_i	$20 \times J_r$
K	3.2×10^{-7}

Tableau III.1 : paramètres du modèle [12].

2.2.Simulation de la première trajectoire :

Cette trajectoire est une « reprise » de la trajectoire sélectionnée dans le mémoire de «Hichem Khebache» [12] dans ses travaux, le but initial étant de « retrouver » des résultats comparables aux sien.

La trajectoire est définie par :

Sur l'axe x par : $f_x(t) = 0.5 \cdot \cos(0.5 \cdot t)$

Sur l'axe y par : $f_y(t) = 0.5 \cdot \sin(0.5 \cdot t)$

Sur l'axe z par : $f_z(t) = 1 + 0.1 \cdot t$

Avec l'angle $\psi = \frac{\pi}{3}$

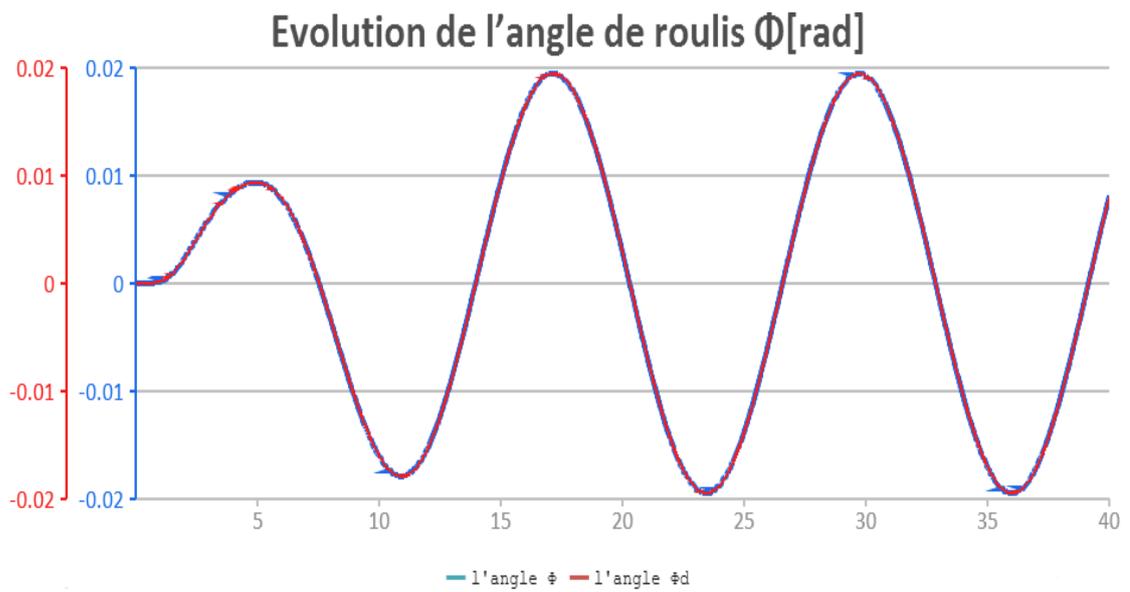


Figure III.1 : Evolution de l'angle de roulis (ϕ)

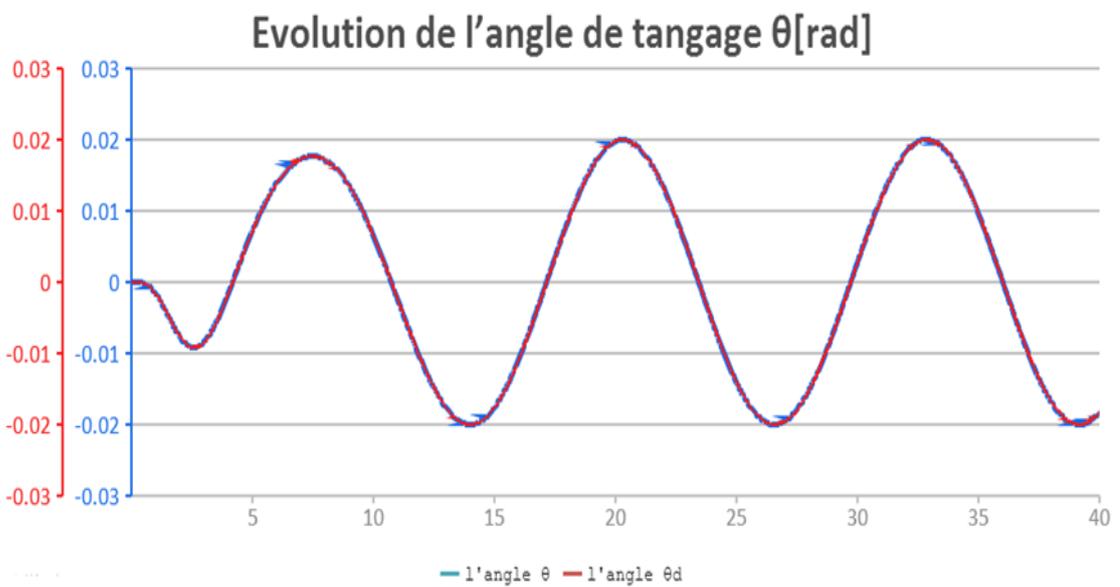


Figure III.2 : Evolution de l'angle de tangage (ϑ).

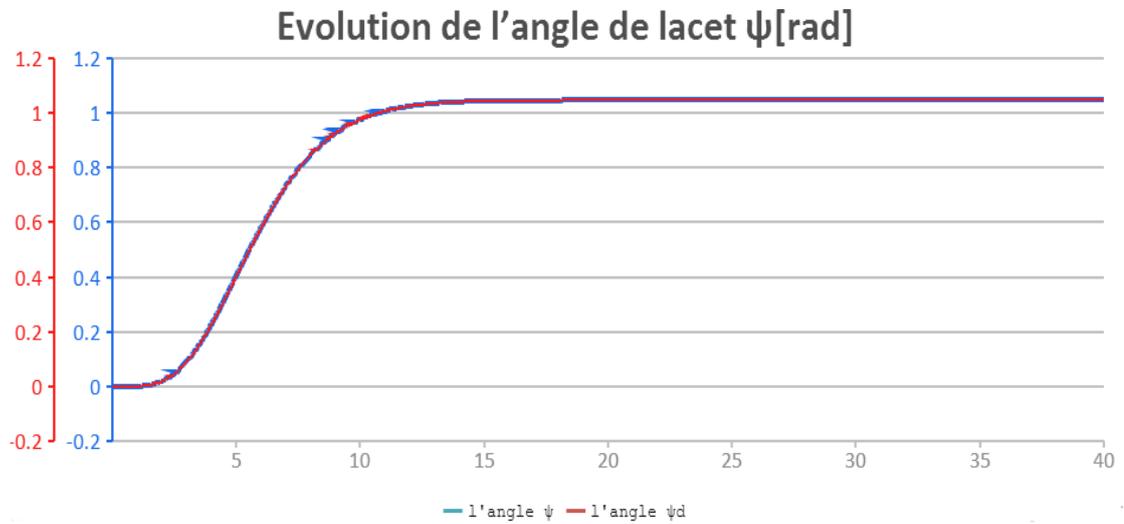


Figure III.3 : Evolution de l'angle de lacet (ψ).

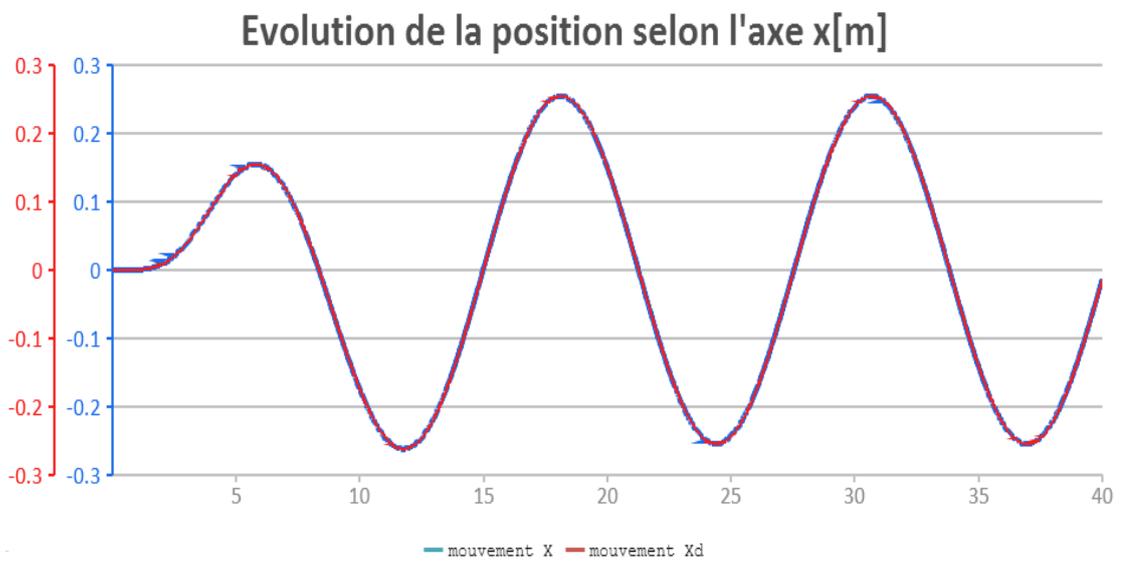


Figure III.4 : Evolution de la position selon l'axe X.



Figure III.5 : Evolution de la position selon l'axe Y.

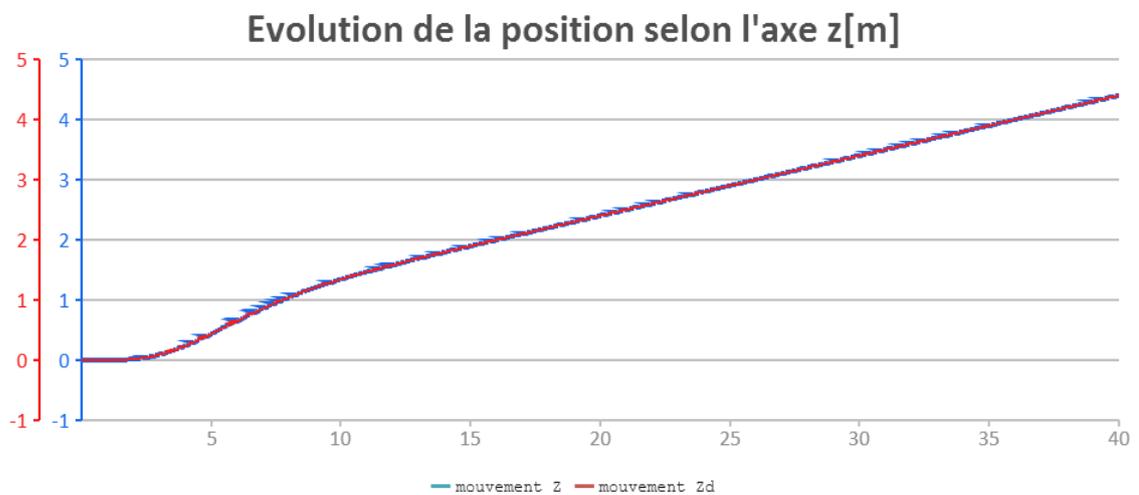


Figure III.6 : Evolution de la position selon l'axe Z.

Mouvement en 3d de consigne
Mouvement en 3d
Réinitialiser

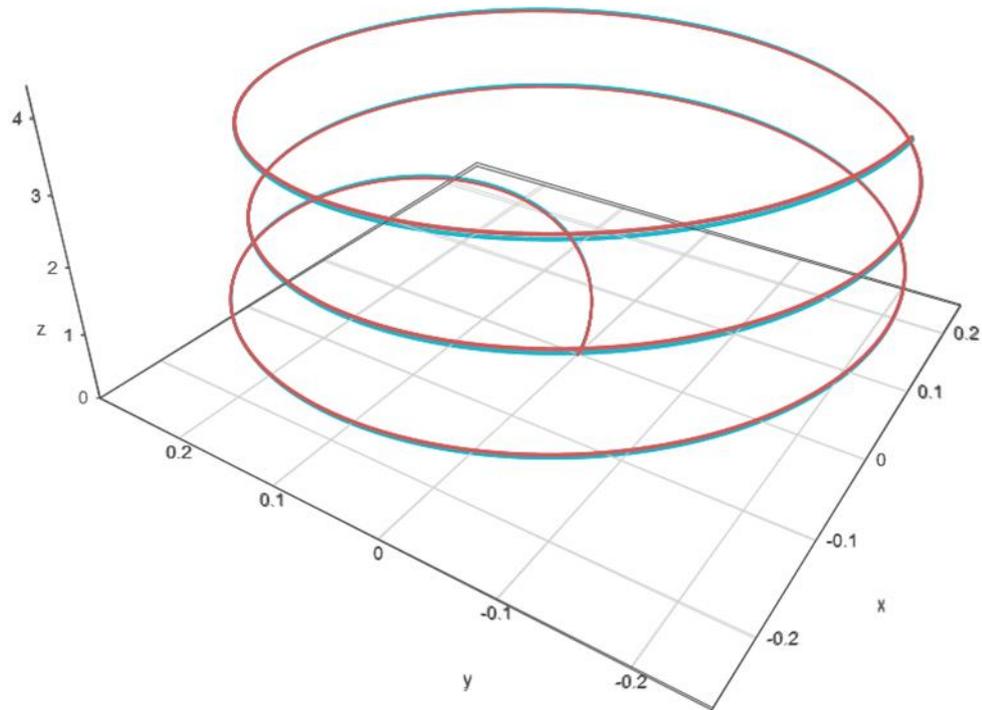


Figure III.7 : Evolution de la position en 3D

2.3.Simulation de la deuxième trajectoire :

Cette trajectoire est une « nouvelle » trajectoire élaborée pour confirmer les performances de la commande et paramètres choisis pour des conditions différentes.

Pour une trajectoire définie :

Sur l'axe x par : $f_x(t) = t$

Sur l'axe y par : $f_y(t) = t$

Sur l'axe z par : $f_z(t) = \cos(t)/\sin(t)$

Avec l'angle $\psi = \frac{\pi}{2}$

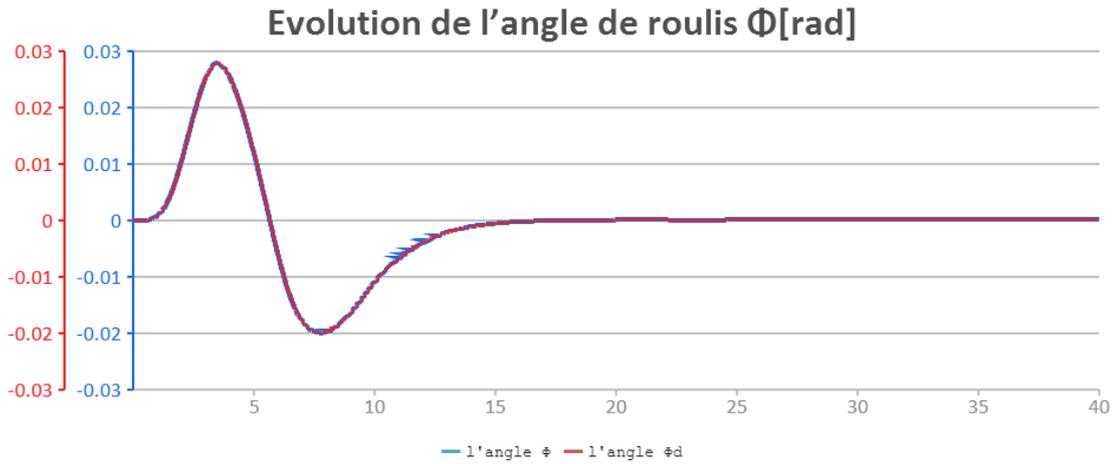


Figure III.8 : Evolution de l'angle de roulis (ϕ).

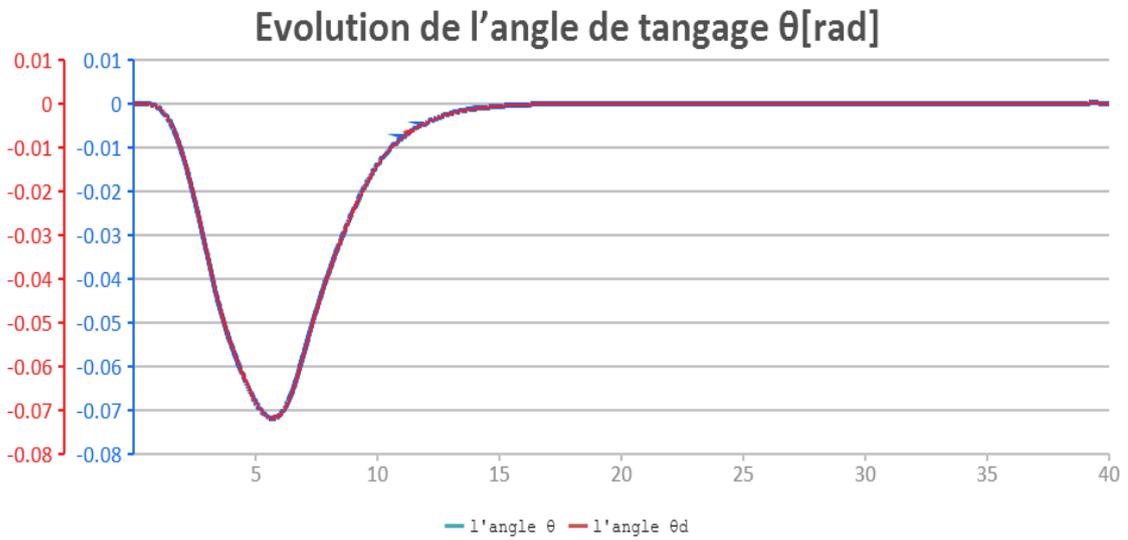


Figure III.9 : Evolution de l'angle de tangage (θ)

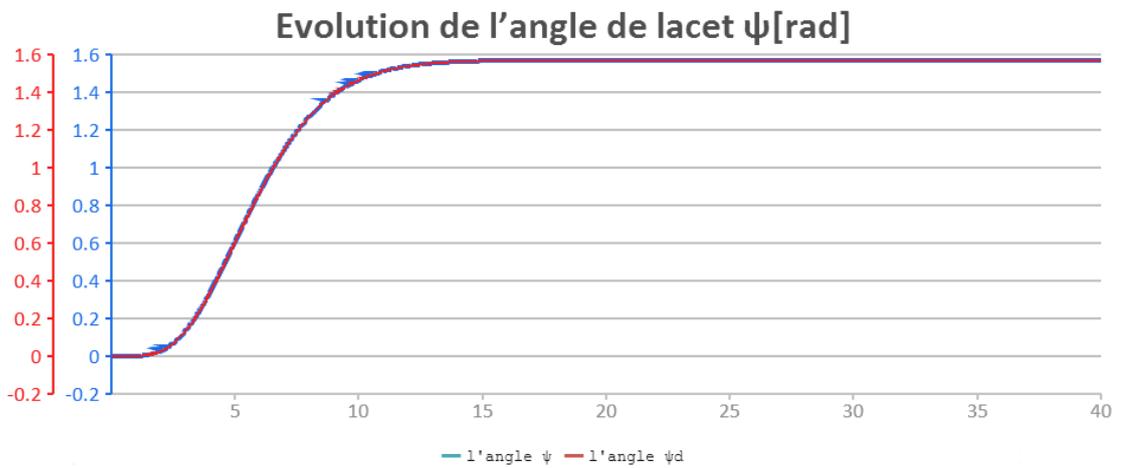


Figure III.10 : Evolution de l'angle de lacet (ψ)

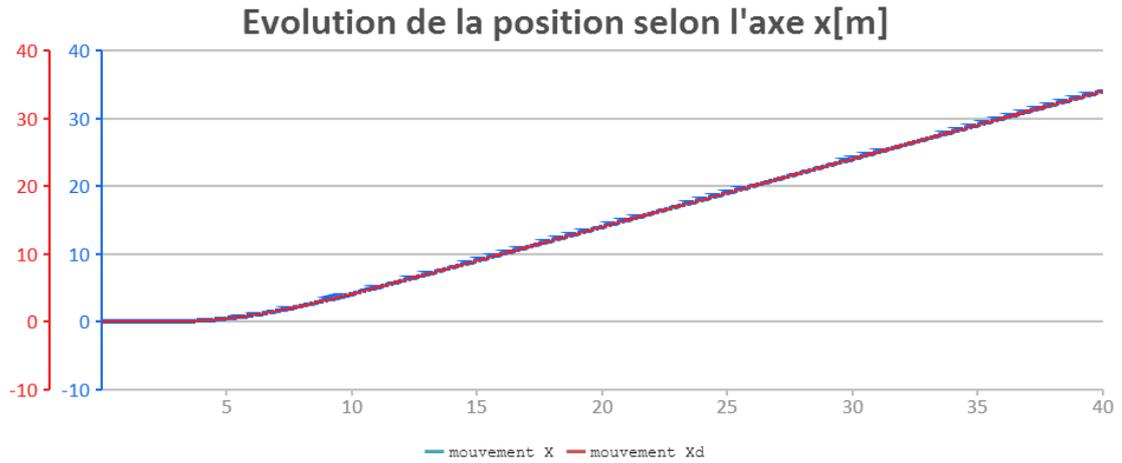


Figure III.11 : Evolution de la position selon l'axe X

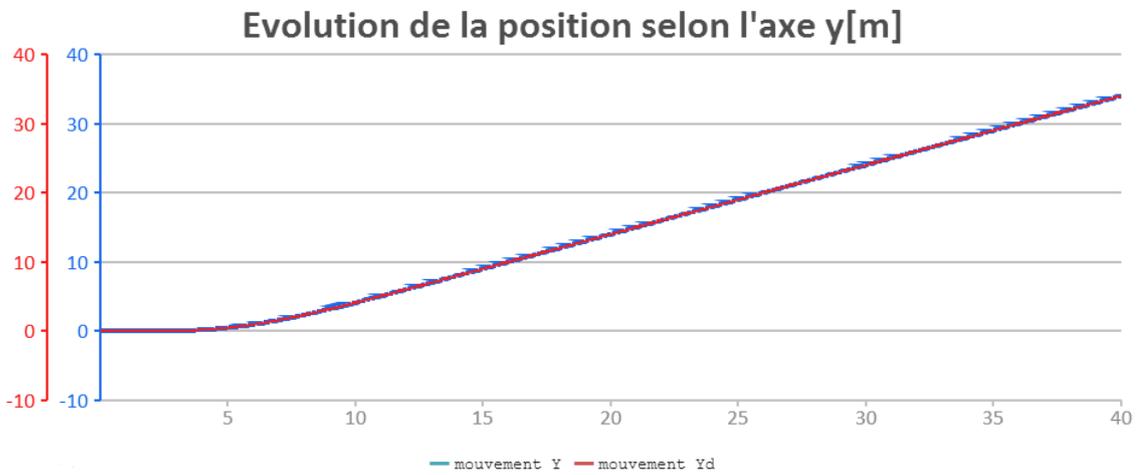


Figure III.12 : Evolution de la position selon l'axe Y



Figure III.13 : Evolution de la position selon l'axe Z

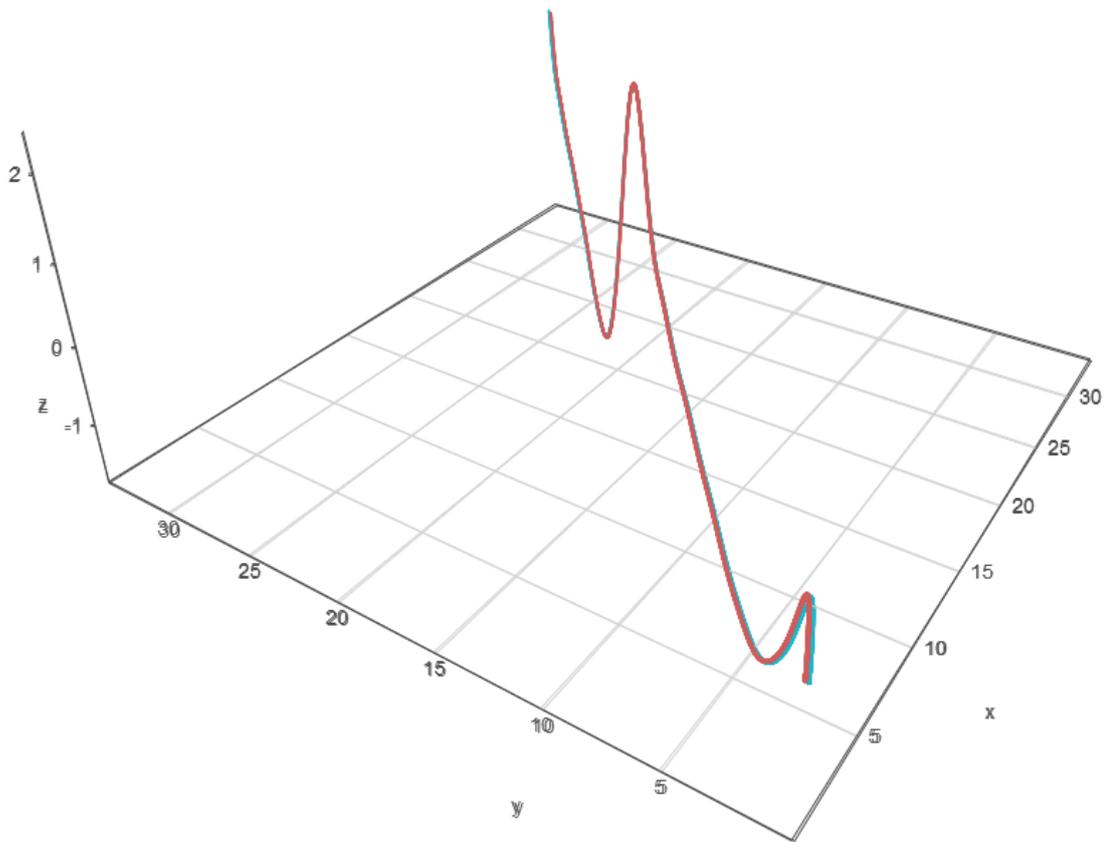


Figure III.14 : Evolution de la position en 3D

3. Les technologies utilisées :

3.1. Enterprise Architect :

Est un outil de modélisation et de conception visuelle basé sur OMG UML (Object management groupe (Unified Modeling Language)) en passant par diverses étapes tel que les étapes d'analyse, la conception de ses différents modèles, la construction de systèmes logiciels, les étapes de teste et d'entretien [21].

Cette outil a été utilisé pour la modélisation de notre système.

3.2. PHP 5 :

Acronyme de (Personnel Home Page) ou bien (HyperText Preprocessor), est un langage de programmation coté serveur, conçue principalement au développement web mais il a aussi d'autre usages dans les différent domine de développement tel que les applications bureau [25].

3.3.PHP Desktop chrome :

PHP Desktop chrome est un projet open source fondé par Czarek Tomczak en 2012 Pour fournir un moyen de développement des applications bureau natives en utilisant des technologies Web telles que PHP, HTML5, JavaScript et SQLite [26].

3.4.Dev-c++ :

Est un environnement de développement intégré gratuit (IDE), distribué sous licence GNU (General Public License pour la programmation en C et C ++) et programmé en Delphi.

Dev-C ++ est généralement considéré comme un programme Windows uniquement, Il existe également une version Linux disponible, mais elle est en version alpha et n'a pas été mis à jour depuis juillet 2002 [22].

Cette outille a été utilisé pour l'écriture du code C.

3.5.Langage C :

Est un langage de programmation procédurale de bas niveau, il a été largement utilisé pour les systèmes d'exploitation et les applications et cela a un large suivi dans la communauté universitaire [23].

Ce langage a été utilisé pour l'implémentation de la loi de commande.

3.6.Photoshop :

Un logiciel d'édition d'image développé et fabriqué par Adobe Systems Inc, Photoshop est considéré comme l'un des leaders dans le logiciel d'édition photo, il permet aux utilisateurs de manipuler, surgir, Redimensionner, Et corriger la couleur sur les photos numériques [24].

Le logiciel est particulièrement populaire pour les graphistes.

3.7.Chart Js & Vis JS :

Des bibliothèques JavaScript open source permet l'illustration des graphes 2D et 3D dans des pages web [27].

4. Interface homme machine :

Après la modélisation et la réalisation du but initial de notre projet, nous avons convenu tardivement qu'une partie supplémentaire matérialisée par une interface graphique pour faciliter la manipulation des trajectoires de consigne serait de grande utilité.

L'interface graphique se compose de :

- ❖ Un ensemble d'entrées pour définir les fonctions de trajectoire selon 3 axes (x, y, z) et l'angle ψ .
- ❖ Un bouton pour compiler et lancer la simulation.
- ❖ Un cadre dans lequel les graphes de simulation s'affichent.

Après l'entrée des fonctions de trajectoire l'application va lancer 2 lignes de commande Windows (en utilisant le langage PHP), une pour le lancement d'une nouvelle compilation du programme de simulation pour intégrer les nouvelles fonctions de trajectoires dans le code source et la deuxième pour exécuter la simulation et générer les fichiers de coordonnées pour chaque mouvement.

Les coordonnées ont lu et illustré dans le cadre spécifié dans l'interface graphique par un code JavaScript (Chart Js pour les graphes 2D et VisJs pour les graphes 3D) à partir des fichiers générés.

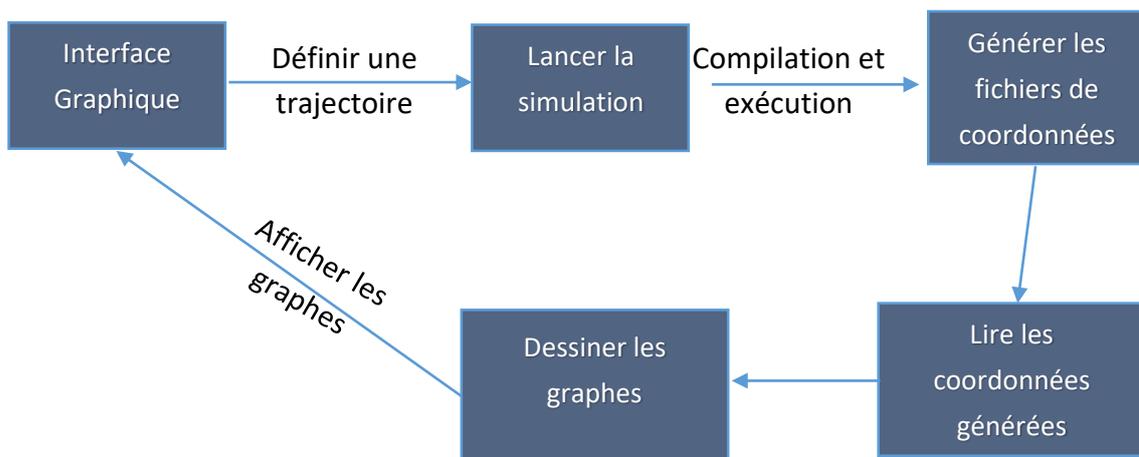


Figure III.15 : Structure de l'IHM.

5. Conclusion :

Dans ce chapitre, nous avons utilisé la commande par mode glissant (Sliding Mode) pour contrôler un système fortement non linéaire composé des mouvements fortement couplé et soumis à de fortes perturbations. La simulation de cette commande donne des résultats parfaits concernant la stabilité et le suivi de trajectoire. Nous espérons que sa mise en œuvre réelle démontrera que la commande par mode glissant est très probante, efficace et robuste pour ce genre d'engins si on prend en considération la capacité du modèle étudié. Nous avons présenté également les différentes technologies utilisées pour la réalisation de notre projet.

La simulation est effectuée à partir d'une interface graphique qui permet la manipulation des trajectoires de consigne selon le choix de l'utilisateur.

Conclusion générale

Conclusion générale :

Le quadrirotor est un engin volant à 4 hélices appartenant au domaine de la robotique aérienne. Il combine en premier abord entre 3 disciplines différentes (mécanique, électronique, informatique) et se base sur des modèles de l'automatique.

Un projet de pilotage d'un quadrirotor se compose de 3 phases essentielles : la phase de modélisation dynamique qui concerne le coté physique de notre étude, la phase de modélisation SysML qui est une étape incontournable de notre « système informatique », et enfin la phase d'implémentation de la loi de commande qui représente l'ensemble des lois mathématiques que le drone suit.

Dans ce contexte, nous avons structuré notre travail selon les aspects suivants :

1. Une vision générale sur les différentes catégories des drones, les différents mouvements qu'un quadrirotor peut faire, les lois physique qu'un quadrirotor subit, une présentation de la loi de commande par mode glissant que l'on peut appliquer sur des systèmes linéaires ainsi que des systèmes non linéaires, ainsi qu'une vision sur les fonctions de transformations.
2. Une modélisation de notre système en utilisant le langage SysML avec une présentation détaillée de ses différents diagrammes.
3. La création d'une interface graphique pour faciliter la manipulation des trajectoires de consigne et donner les graphes de simulation résultants du système implémentée, ces derniers ayant aboutis à une commande efficace et robuste si on prend en considération la capacité du modèle étudié ; nous avons présenté également les différentes technologies utilisées dans les différentes étapes de la réalisation de notre projet.

Parmi les difficultés majeures qu'on a trouvées dans notre projet et inhérentes à ce genre d'applications :

- ❖ L'interprétation de certaines fonctions prédéfinies sous MATLAB en langage C tel que les dérivés filtrés, les dérivateurs, les intégrateurs et les fonctions de transformations car le langage C ne vise pas les calculs mathématiques complexes.
- ❖ La rareté des références concernant le sujet étudié dans le domaine informatique pose un problème de manque d'information, en particulier lorsque l'application est afférente à un engin spécifique.

- ❖ Le travail sur un projet des drones nécessite des connaissances considérables en mathématiques et physiques. Là aussi une équipe pluridisciplinaire serait fortement utile pour plus d'efficacité.
- ❖ Le temps très limité pour la réalisation du projet.

Malgré ces nombreux handicaps, nous aboutissons à des résultats parfaits en simulation et espérons voire une implémentation effective de notre application sur un engin réel.

Nous espérons également que notre approche et notre méthodologie de travail pourront profiter à des applications similaires.

Références bibliographiques :

- [1]. Le parisien, <<http://www.leparisien.fr/high-tech/l-histoire-du-drone-racontee-par-son-inventeur-13-02-2015-4530761.php>> , consulté le 14-6-2017.
- [2]. Kared Saber, Modélisation et Commande d'un Mini-Hélicoptère Drone, thèse de doctorat, département génie électrique, université de skikda, soutenu le : 22-1-2013.
- [3]. R. E. Weibel and R. J. Hansman. Safety Considerations for Operation of Unmanned Aerial Vehicles in the National Airspace System. MIT International Center for Air Transportation, Report No. ICAT-2005-1, Mars 2005.
- [4]. Mokhtari Mohammed Rida, Observation et Commande de Drones Miniatures à voilures tournantes, Thèse de doctorat, faculté de Technologie, Université de Tlemcen, Soutenu le : 22-10-2015.
- [5]. J. M. Pflimlin et T. Hamel P. Soueres. Position control of a ducted fan VTOL UAV in crosswind. International Journal of Control, vol. 80, 666-683, 2007.
- [6]. Alexander. Coaxial Rotary-Wing Mini Aerial Vehicle Aeromechanics. A thesis submitted in fulfillment of the requirements for the degree of Doctor of Philosophy, 2009.
- [7]. Tarek Hamel, Philippe Souères, Modélisation, estimation et contrôle des drones à voilures tournantes : Un aperçu des projets de recherche français, UNSA-CNRS, LAAS-CNRS, Toulouse, France, (sans année).
- [8]. Thomas Rakotomamonjy, Modélisation et contrôle du vol d'un micro drone `a ailes battantes, thèse de doctorat, faculté des sciences et techniques, UNIVERSITÉ PAUL CÉZANNE AIX-MARSEILLE, soutenu le : 8-2-2006.
- [9]. Samir Bouabdallah and Roland Siegwart « Backstepping and Sliding mode Techniques Applied to an Indoor Micro Quadrotor» IEEE international conference on robotics and automation ,Barcelona,spain,april, 2005.
- [10]. ZAKARIA BELLAHCENE, Synthèse de lois de commande robuste pour un hélicoptère à quatre hélices, thèse de Magister, Faculté de Génie Electrique, Université des Sciences et de la Technologie d'Oran Mohamed Boudiaf, soutenu le : 6-3-2013

- [11]. L. Besnard “Control of a quadrotor vehicle using sliding mode disturbance observer”, Master Thesis, Alabama university, 2006.
- [12]. Hicham KHEBBACHE, Tolérance aux défauts via la méthode backstepping des systèmes non linéaires : Application : Système UAV de type Quadrirotor, mémoire de magister, faculté de technologie, UNIVERSITE FERHAT ABBAS DE SETIF, soutenu le : 6-6-2012.
- [13]. STABILISATION D'UN QUADRIROTOR, Rapport PIP 20052006. <http://www.vuedehaut.fr/Quadri/Rapport_PIP_Quadrirotor.pdf>
- [14]. W.Khalil, Dombre « modelling, identification and control of robots » HPS edition, (2002).
- [15]. Pedro Castillo, Rogelio Lozano and Alejandro E. Dzul « Modelling and Control of Mini-Flying Machines » ,Book Springer-Verlag London Limited 2005.
- [17]. K. Nonami , F. Kendoul ,S S W. Wang .D. Nakazawa « Unmanned Aerial Vehicles and Micro Aerial Vehicles » Book Springer Tokyo Dordrecht Heidelberg London New York, Springer 2010.
- [18]. ONLINE ELECTRICAL ENGINEERING STUDY SITE, <<https://www.electrical4u.com/transfer-function>> , consulté le: 9/6/2017 à 10h.
- [19]. Hadjer ABDERREZEK, Contribution à la commande robuste des systèmes non linéaires, memoire de magister, Département d'Electrotechnique, UNIVERSITÉ FERHAT ABBAS -SETIF-1, Soutenu le 2/10/2014.
- [20]. système de gestion de contenu libre Plone, <http://www.uml-sysml.org> , consulté le 2/5/2017 à 18h.
- [21]. Modélisation UML & SysML, <<http://www.umlchannel.com/fr/enterprise-architect>>, consulté le : 20/06/2017 à 13h.
- [22]. Encyclopédie en ligne, Thesaurus, dictionnaire de définitions, <<http://dictionnaire.sensagent.leparisien.fr/Dev-Cpp/en-en>> , consulté le : 20/06/2017 à 14h.
- [23]. TechTarget, <<http://searchwindowsserver.techtarget.com/definition/C>>, consulté le : 20/06/2017 à 14h.
- [24]. Businessdictionary, <<http://www.businessdictionary.com/definition/Photoshop.htm>>, consulté le : 21/06/2017 à 11h.

- [25]. quora, <<https://www.quora.com/What-Is-PHP>>, consulté le : 21/06/2017 à 13h.
- [26]. Phpocean, <<https://phpocean.com/tutorials/design-and-illustration/create-your-first-desktop-application-with-php-and-php-desktop/4>>, consulté le : 21/06/2017 à 15h.
- [27]. Hinault Romaric, <https://www.developpez.com/actu/53168/La-bibliotheque-JavaScript-open-source-Chart-js-disponible-elle-permet-de-generer-des-graphiques-HTML5/>, consulté le: 21/06/2017 à 15h:30.

Liste des figures :

Figure I.1 : Drone avec différents modes de vol	9
Figure I.2 : Géométrie du quadrirotor	10
Figure I.3 : Structure générale d'un quadrirotor	11
Figure I.4 : Illustration du mouvement de roulis	11
Figure I.5 : Illustration du mouvement de tangage	12
Figure I.6 : Illustration du mouvement de lacet	13
Figure I.7 : Illustration du mouvement vertical	14
Figure I.8 : Illustration de la translation horizontale	15
Figure I.9 : les angles d'Euler	16
Figure I.10 : Schéma synoptique de la commande proposée.	25
Figure I.11 : Glissement idéal.	26
Figure I.12 : Glissement réel.	26
Figure II.1 : Les différents diagrammes SysML.	30
Figure II.2 : Diagramme d'exigences.	31
Figure II.3 : Diagramme de cas d'utilisations.	33
Figure II.4 : Diagramme de séquences Auto piloter	35
Figure II.5 : Diagramme de séquence pour le cas « contrôler »	37
Figure II.6 : Diagramme D'état transition	38
Figure II.7 : Diagramme de définition de blocs	40
Figure III.1 : Evolution de l'angle de roulis (φ)	44
Figure III.2 : Evolution de l'angle de tangage (θ).	44
Figure III.3 : Evolution de l'angle de lacet (ψ).	45
Figure III.4 : Evolution de la position selon l'axe X.	45
Figure III.6 : Evolution de la position selon l'axe Z.	46
Figure III.5 : Evolution de la position selon l'axe Y.	46
Figure III.7 : Evolution de la position en 3D	47
Figure III.8 : Evolution de l'angle de roulis (φ).	48
Figure III.9 : Evolution de l'angle de tangage (θ)	48
Figure III.10 : Evolution de l'angle de lacet (ψ)	48
Figure III.12 : Evolution de la position selon l'axe Y	49
Figure III.11 : Evolution de la position selon l'axe X	49
Figure III.13 : Evolution de la position selon l'axe Z	49
Figure III.14 : Evolution de la position en 3D	50
Figure III.15 : Structure de l'IHM.	52

Liste des tableaux :

Tableau I.1 : classification des drones .	7
Tableau III.1 : paramètres du modèle .	43

Liste des abréviations :

UAV : Unmanned Aerial Vehicle.

IAI : Industrie Aérospatiale Sioniste.

R.E.A : Rayon d'action, Endurance et Altitude.

HALE : Haute Altitude Longue Endurance.

MALE : Moyenne Altitude Longue Endurance.

UCAV: Unmanned Combat Air Vehicles.

TUAV: Tactical Unmanned Aerial Vehicle

MAV : Mini Air Vehicle.

3D : 3 dimensions.

2D : 2 dimensions.

Rot : Rotationnelle.

v : vitesse linéaire.

P : le poids de quadri-rotors.

F_i : Les forces de poussée (portance).

ω_i : La vitesse de rotation des moteurs.

T_i : Les forces de traînée (résistance).

b : Le coefficient de portance.

d : Le coefficient de traînée.

K_{ft} : Le coefficient de force de traînée de translation.

M_{x_i} :Le moment de rotation.

M_a : Moment résultant des frottements aérodynamiques.

K_{fa} : Le coefficient des frottements aérodynamiques.

Ω : La vitesse angulaire.

M_{gh} : Moment gyroscopique des hélices.

J_r : L'inertie des rotors.

M_{gm} : Moment gyroscopique dû aux mouvements de quadrirotor.

J : l'inertie de système.

ζ : Est le vecteur de position du quadrirotor.

m : la masse totale du quadrirotor.

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

R : La matrice de rotation.

\wedge : Le produit vectoriel.

$S(\Omega)$: la matrice antisymétrique.

F_f : La force totale générée par les quatre rotors.

F_g : La force de gravité.

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

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

V : est la tension d'entrée du moteur.

k_e : La constante du couple électrique.

k_m : La constante des couples mécaniques.

k_r : La constante du couple de charge.

r : la résistance du moteur.

L : L'inductance du moteur.

J_r : L'inertie du rotor.

C_s : Le frottement sec.

SysML: systems modeling language.

UML: Unified modeling language.

$X(t)$: le vecteur d'état.

$u(t)$: vecteur de commande.

$w(x,t)$: vecteur de perturbation.

$F(x,t), g(x,t)$: fonctions non linéaires du vecteur d'état décrivant le système.

m : la masse

g : l'accélération de pesanteur.

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

I_x, I_y, I_z : les éléments diagonal de la matrice d'inertie J de dimension (3x3).

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

$K_i : i \in \{1, 2, 3, 4\}$ sont des gains positifs.

ϕ : L'angle de roulis.

ψ : L'angle de lacet.

θ : L'angle de tangage.

OMG: Object management group.

MATLAB: Matrix laboratory.

IDE: Integral development environment.

GNU: General public license

Résumé : Le but de ce projet est d'implémenter une loi de commande pour le guidage et le pilotage d'un drone quadrirotor. L'obtention des résultats efficace nécessite une étude approfondie des différents mouvements effectués par le drone, Le choix d'une loi de commande plus adaptée au système étudié, La modélisation du système étudié.

Dans notre cas nous avons un système non linéaire, cela nécessite l'utilisation de la commande par mode glissant qui est plus adapté avec ce genre de système.

Comme une dernière étape nous avons fait une simulation, qui a donné des résultats exacts et très satisfaisants.

Mots clés : drone, quadrirotor, modélisation, loi de commande, mode glissant.

المخلص: الهدف من هذا البحث انشاء نظام تحكم بطائرة بدون طيار من نوع كوادكوبتر

الحصول على نتائج فعالة وإيجابية يستلزم دراسة معمقة لمختلف التحركات التي تقوم بها الطائرة، اختيار قانون تحكم أكثر تلائماً مع النظام المدروس، انشاء تصميم للنظام المدروس.

لدينا نظام غير خطي لذلك وجب علينا استخدام قانون تحكم بوضع انسحاب لأنه أكثر ملائمة معه.

كمرحلة أخيرة قمنا بإجراء عملية محاكاة والتي أعطتنا نتائج دقيقة وجد مرضية

كلمات مفتاحية: طائرة بدون طيار، قانون التحكم، التحكم بطريقة الانسحاب، التصميم.

Abstract: The aim of this project is to implement a control law for the guidance and control of a quadrirotor drone. Achieving effective results requires an in-depth study of the different movements carried out by the drone, the choice of a control law more adapted to the system studied, modeling of the system studied.

In our case, we have a non-linear system; this requires the use of sliding control mode, which is more suitable with this kind of system.

As a final step, we did a simulation, which gave exact and very satisfactory results.

Key words: quadrirotor, drone, sliding mode, control law, modelling.