

Année académique : 2016 - 2017



Etude du déplacement d'un drone par Motion Capture

Nom : NKENKO
Prénom : Kevin Jubal
N° d'étudiant : UL0061 18910

Encadré par :

Stéphane DERRODE & Romain VUILLEMOT

Abstract

This purpose of this project is to study the movements of a quadrotor drone during a flight using the motion capture device. The motion capture is a tool that records in real time the position data of a subject and enable us to analyze the movements recorded. The goal was to determine how precise can a drone be when giving a path to follow.

First of all, it was necessary to get a sufficient insight of motion capture technologies and to learn how to use them. To achieve this, a small research along with available books about the subject and tutorials were very useful.

The next step was to learn how the drone works and what can be expected of it. It was also the moment to learn the API in order to control the drone with a program from the computer.

Then, a simulator of drone has been introduced to simulate the movements and the behaviors of a drone running perfectly as expected. This simulator is supposed to serve as a reference when it comes to comparison with a real drone.

Some methods to evaluate the error between the theoretical trajectory and the real trajectory of the drones has been proposed. Doing so, a program for synchronizing the samples of both trajectories has been set up.

Finally, a program for automatic steering has been developed to enable the drone to follow any path giving as entry.

Résumé

Ce projet traite de l'étude du comportement d'un drone quadri-rotor en vol à l'aide de la motion capture. La motion capture étant ici un dispositif qui permet d'enregistrer en temps réel la position d'une entité (humain, animal, objet...) et qui donne ainsi la possibilité de reconstituer et d'analyser le mouvement enregistré avec un bon niveau de précision. Il était question de voir quel niveau de précision dans les déplacements on peut attendre d'un drone à qui on imposerait une trajectoire donnée.

La première étape du projet était de prendre en main et de s'approprier le logiciel et les équipements installés pour la motion capture. Cela passait nécessairement par une phase de documentation sur le sujet et de suivi des tutoriaux disponibles.

Ensuite, il fallait prendre en main le drone. En effet, il s'agissait de voir tous les mouvements que le drone est capable d'effectuer et de prendre connaissance de l'API de programmation du drone.

Puis, un simulateur de drone « parfait » a été introduit pour servir de point de référence et de comparaison avec le drone réel. Il s'agit d'un programme qui simule les déplacements du drone dans l'hypothèse où celui-ci exécuterait exactement et parfaitement les commandes qui lui sont envoyées.

Des méthodes d'évaluation de l'erreur entre la trajectoire théorique et la trajectoire enregistrée du drone ont été proposées. Il a fallu à ce niveau créer un petit programme pour synchroniser l'échantillonnage des deux trajectoires à étudier.

Un programme de pilotage automatique de vol a enfin été développé pour permettre au drone de suivre n'importe quelle trajectoire qui lui serait donnée.

Table des matières

ABSTRACT	2
RESUME	3
INTRODUCTION	5
I. PRESENTATION DU DISPOSITIF MOCAP	6
1. PRINCIPE	6
2. PIPELINE D'UNE ACQUISITION	8
A. AVANT L'ACQUISITION	8
B. PENDANT L'ACQUISITION	9
C. POST TRAITEMENT	9
3. LES AVANTAGES DE CE DISPOSITIF	10
4. LES INCONVENIENTS DU DISPOSITIF	10
II. PRESENTATION DU DRONE	10
1. SENS DE ROTATION DES PALES	11
2. MOUVEMENTS POSSIBLES	12
A. LE LACET	12
B. LE TANGAGE ET LE ROULIS	13
3. LES MODES DE VOL	14
III. SIMULATEUR DE TRAJECTOIRE	15
1. PARAMETRAGE DE L'ESPACE	16
2. LE MODELE INFORMATIQUE DU SIMULATEUR	17
3. PLANIFICATEUR DE VOL	17
A. LE MODE AVION	17
B. LE MODE « TRANSLATION ONLY »	18
4. ESTIMATION DE L'ERREUR	19
CONCLUSION GENERALE	23
BIBLIOGRAPHIE	24

Introduction

Des technologies qui étaient autrefois réservées aux scientifiques et à un cercle restreint d'experts deviennent peu à peu des outils à la portée de tous. En effet, la vulgarisation des nouvelles technologies continue son expansion et on voit aujourd'hui l'essor grandissant des appareils comme les drones. Ces aéronefs télécommandés sans pilote à bord avec le plus souvent une charge utile embarquée, qui ont été initialement développés pour la surveillance, le renseignement et l'intervention militaire, ont été adaptés au secteur civil à l'exemple d'autres innovations technologiques majeures initialement militaires comme les satellites, le GPS, internet, les moteurs à réaction etc.

La capture de mouvement (motion capture ou mocap) fait aussi parti de ces technologies qui ont servi à l'armée pour les simulations et qui sont aujourd'hui utilisées dans des domaines variés comme la médecine, le sport, l'art et l'ingénierie. L'utilisation de ces deux technologies, drone et capture de mouvement, fait appel à des disciplines variées à l'instar de l'aérodynamique, le traitement et la visualisation d'images, la mécanique, et l'informatique. Dans le cadre de ce projet, nous nous intéresserons à un dispositif de capture de mouvement optique installé dans la salle Amigo fournis par Vicon et à un quadricoptère commercial distribué par Parrot. L'objectif de ce projet est d'abord de découvrir ces technologies et de se les approprier. Ensuite, il est question d'écrire un programme pour piloter le drone automatiquement sans assistance d'un opérateur afin de le garder sur une trajectoire donnée et de déterminer ses niveaux de précisions. Pour parvenir aux fins escomptées, une présentation de ces deux technologies et de leur principe de fonctionnement sera donnée avant la présentation des résultats du projet.

I. Présentation du dispositif MOCAP

La capture de mouvement, Motion capture en anglais qu'on appellera aussi Mocap est une technique permettant d'échantillonner et d'enregistrer la position d'un objet, d'un animal ou d'un humain en mouvement afin de produire des données 3D qui serviront aux analyses, à la reconstruction de mouvements ou encore à l'animation d'avatar. Il existe plusieurs technologies permettant de réaliser une capture de mouvement : les technologies optiques, les technologies mécaniques, les technologies magnétiques et les technologies gyroscopiques. Toutefois le principe reste à peu près le même. Le dispositif installé dans la plateforme Amigo et qui va être décrit dans les lignes qui suivent utilise une technologie optique.

1. Principe

Le système de capture est constitué à la fois d'une partie matérielle et d'une partie logicielle. La partie logicielle est en fait un serveur qui va extraire les données, les traiter ou les transmettre en temps réels à une application de visualisation (figure 1).

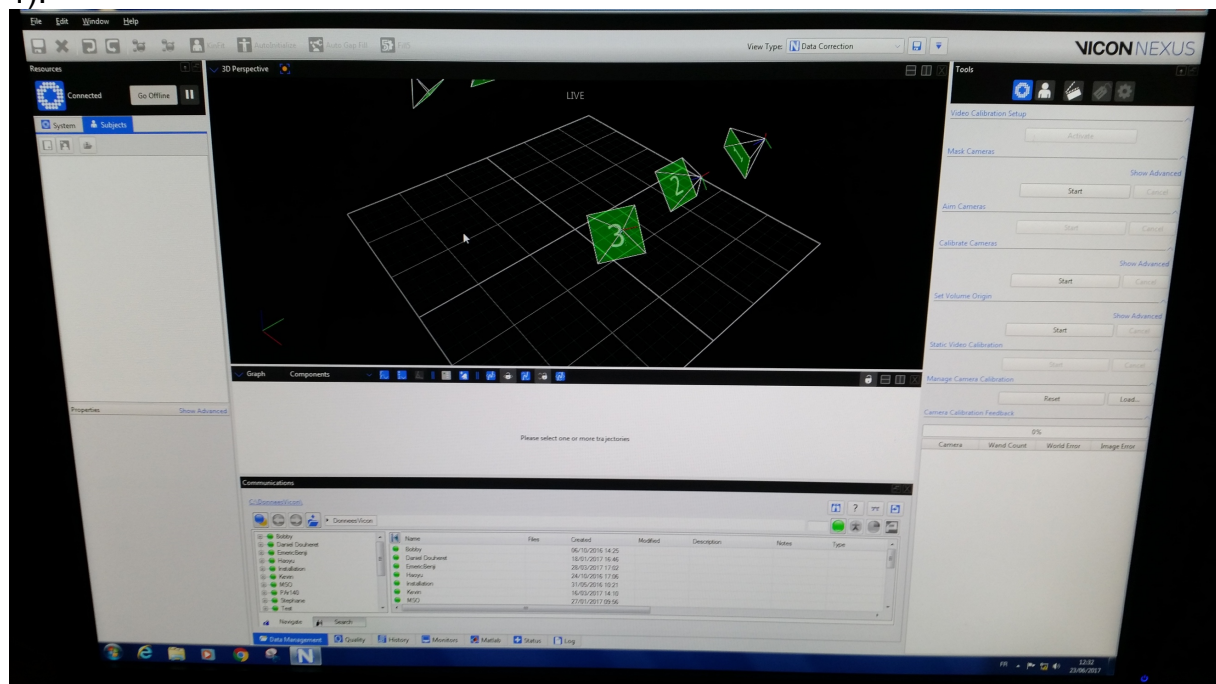


Figure 1- Logiciel Nexus vicon

La partie matérielle est constituée d'un ordinateur central qui est couplé à un réseau Ethernet de caméras digitales. Ces caméras digitales sont des appareils photosensibles équipées de diodes électroluminescentes, qui utilisent une matrice de cellules photoélectriques (pixels) pour capturer la lumière, et ensuite calcule l'intensité de la lumière pour chaque cellule. Les caméras utilisées dans le cadre de ce travail sont des caméras Vicon Bonita 10 (figure 2) qui ont une résolution de 1 mégapixel, une précision de 0.5 mm et une fréquence maximale de 250 fps (frames par seconde).



Figure 2 - Caméra Vicon Bonita 10

On peut penser qu'une plus grande résolution permet d'avoir des résultats plus précis. Cependant cela ne suffit pas. Il est aussi important de tenir compte de la fréquence d'échantillonnage du dispositif. Plus on elle est grande, plus on peut enregistrer les subtilités des mouvements avec une meilleure précision. Nous utilisons généralement 100 échantillons par seconde (fps) sur le dispositif de la salle Amigo. Il est aussi crucial de synchroniser la vitesse de l'obturateur avec les LEDs qui sont fixées sur les caméras et qui servent de source de lumière. Lors de l'acquisition, les LEDs des caméras émettent un rayonnement infrarouge qui est réfléchi par les marqueurs disposés sur le sujet. Ces marqueurs (figure 3) sont des sphères dont le diamètre varie entre 5mm et 5 cm et dont la surface est constituée d'une matière réfléchissante.



Figure 3 - Marqueurs

2. Pipeline d'une acquisition

a. Avant l'acquisition

La première étape consiste à préparer et à calibrer l'équipement. Cela passe aussi par une étape de vérification des vues des différentes caméra ; il faut appliquer des masques sur les caméras afin de supprimer les points parasites. Il faut également bien délimiter le périmètre du sujet. Il s'agit de l'espace de la salle couvert par toutes les caméras à la fois.

Le calibrage

Le dispositif de capture de mouvement est calibré en utilisant toutes les caméras pour traquer un objet dont les dimensions sont connues par le logiciel. Cet objet est une baguette active LED (Active Wand LEDs) qui est un dispositif autonome (avec une batterie) en forme de « T » équipé de LEDs et qui est automatiquement reconnu par le logiciel. Cet objet est aussi équipé d'une photodiode qui non seulement synchronise la baguette avec les caméras mais aussi ajuste la luminosité des LEDs pour qu'elles soient adaptées à l'environnement de travail. Cet objet permet également de définir un repère cartésien dans le périmètre d'acquisition.



Figure 4 - Active Wand

En combinant les vues des différentes caméras de l'objet dont les dimensions sont connues, la position exacte de chaque caméra de l'environnement de travail est calculée. Un tout petit déplacement d'une des caméras nécessite un recalibrage. Il est donc conseillé de calibrer le dispositif avant chaque acquisition.

Pour déterminer la position tridimensionnelle d'un point, il faut un minimum de 2 caméras. Avoir plus de caméras permet de déterminer avec plus de précision les positions surtout lorsque le sujet est équipé de plusieurs marqueurs. Cependant, avoir plus de caméras implique aussi plus de temps de post-traitement et plus d'investissement. Il faut donc faire un compromis et déterminer le nombre approprié

de caméras en fonction du périmètre d'acquisition, la lumière disponible et de la vitesse des mouvements à enregistrer.

Le positionnement des marqueurs

Il faut également placer les marqueurs intelligemment et mettre une combinaison si le sujet est humain. Il est plus évident d'identifier les marqueurs lorsqu'ils sont placés de manière asymétrique. Lorsque le sujet est humain, il est recommandé de placer les marqueurs au niveau des os des articulations.

b. Pendant l'acquisition

Tout au long de l'acquisition, il faut veiller à ce que le sujet ne sorte pas du périmètre délimité et qu'aucun marqueur ne soit caché (par un bras par exemple) de la vue des caméras.

c. Post traitement

Une fois la capture terminée, le logiciel produit un playback qui reconstruit le mouvement des marqueurs. Il existe plusieurs méthodes utilisées pour réduire le bruit, isoler les marqueurs et les séparer de l'environnement extérieur. L'approche la plus utilisée consiste à séparer les groupes de pixels selon un seuil de luminosité défini dans le logiciel. Ce dernier utilise les échantillons adjacents pour traiter les échantillons douteux.

Ensuite, les coordonnées bidimensionnelles de chaque marqueur sont déterminées par rapport à la vue de chaque caméra. Ces données seront combinées avec la position des caméras pour obtenir les coordonnées tridimensionnelles de chaque marqueur par triangulation.

L'étape suivante consiste à identifier chaque marqueur selon une séquence. C'est l'étape la plus délicate puisque chaque marqueur sera identifié manuellement. Plus on a de marqueurs, plus cette étape sera longue. Après cette étape d'identification, le logiciel essaie de conserver la même identification des marqueurs en reconstruisant la scène enregistrée. Cependant le logiciel perd souvent la trace des capteurs. Il peut avoir des difficultés à suivre les marqueurs lorsque ceux-ci sont cachés à la vue d'une ou plusieurs caméras (on parle d'obstruction) ou lorsque deux marqueurs passent très près l'un de l'autre (on parle de crossover). Il peut aussi arriver que le logiciel fasse une confusion ou un échange accidentel d'identification lorsque les marqueurs passent très près les uns des autres. Il y a alors des trous (marqueurs manquants) et des marqueurs que le logiciel n'a pas pu identifier. A ce niveau, il faut identifier une fois de plus les marqueurs manuellement et combler les trous en précisant pour les marqueurs manquants, les marqueurs les plus proches. Lorsque tout la scène est reconstituée sans encombre avec les marqueurs proprement identifiés, on peut exporter les données de position de ces marqueurs dans un fichier.

Le fichier généré par le logiciel est en fait une séquence de la position des marqueurs au cours du temps. Ceci dit, le fichier contient juste les coordonnées

cartésiennes des marqueurs pour chaque échantillon sans aucune autre hiérarchie ou précision sur les rotations.

3. Les avantages de ce dispositif

- Les systèmes optiques sont d'une très bonne précision. Précision de l'ordre du millimètre.
- Ils permettent de suivre un grand nombre de marqueurs et il est facile de changer les configurations des marqueurs manuellement.
- On peut modéliser un squelette en réalisant des groupes de marqueurs (segments de 2 ou 3 marqueurs).
- Il est aussi important de noter que le sujet n'est pas encombré par des câbles pendant l'acquisition.
- Les systèmes optiques permettent de réaliser des acquisitions dans un plus large périmètre comparés aux autres systèmes.
- Les systèmes optiques ont une plus grande fréquence d'échantillonnage ce qui signifie plus de mesures par seconde.

4. Les inconvénients du dispositif

- Les dispositifs optiques demandent un lourd et fastidieux posttraitement.
- Le matériel est très onéreux. Il peut atteindre 50000 \$ pour un équipement entrée de gamme et dépasser le million de dollars pour un équipement plus haut de gamme.
- Les systèmes optiques ne peuvent plus suivre le mouvement si un ou plusieurs capteurs sont obstrués (cachés) pendant longtemps.
- L'acquisition doit se faire dans un environnement à l'abri de la lumière jaune et des bruits lumineux réfléchissants. C'est pourquoi il est nécessaire de porter une combinaison spéciale pour les sujets humains lors de l'acquisition.

II. Présentation du drone

Un drone quadrirotor est un hélicoptère constitué de quatre rotors à voilure tournante. Cet engin qui peut aller dans toutes les directions peut même effectuer un décollage vertical et un atterrissage vertical. Il est constitué de quatre pales à pas fixes accouplées à des moteurs à courant continu. Le drone utilisé ici est l'AR Drone 2 commercialisé par Parrot.



Figure 5 - Parrot Ar Drone 2

1. Sens de rotation des pales

Afin d'éviter un mouvement de lacet (l'appareil tourne sur lui-même), il est nécessaire que deux rotors tournent dans un sens et que les deux autres tournent dans l'autre sens. En plus, pour diriger l'appareil, les rotors tournant dans le même sens doivent être placés l'un en face de l'autre (figure 6). De plus, pour faciliter la gestion des commandes et les calculs, le rotor numéro 1 est choisi comme l'avant du micro hélicoptère.

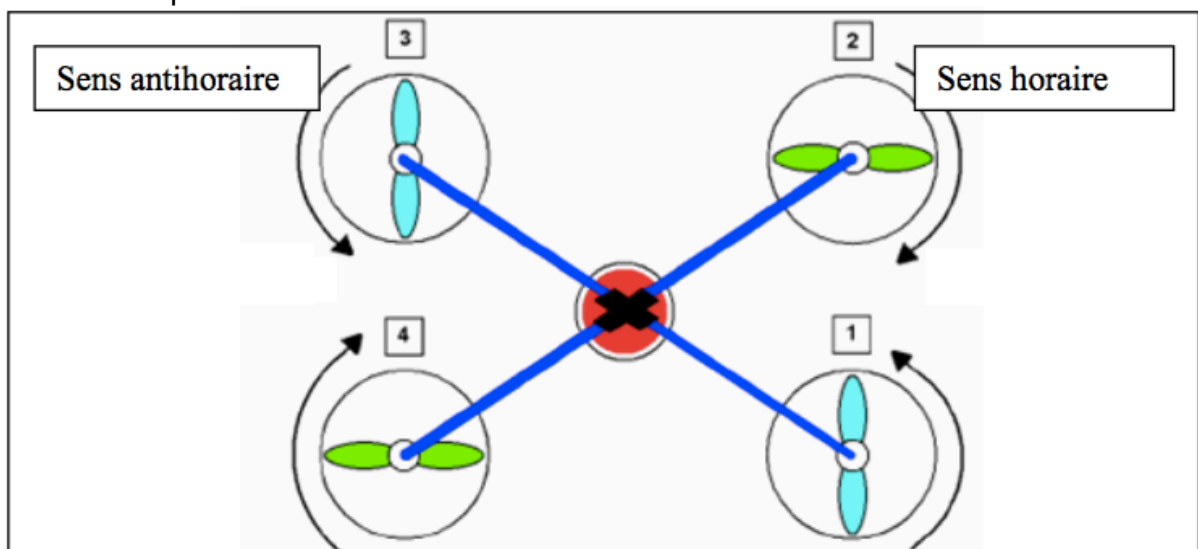


Figure 6 - Sens de rotation des pales

2. Mouvements possibles

Il existe quatre mouvements possibles pour un quadrirotor : la poussée (les gaz), le lacet, le tangage et le roulis. Le mouvement de gaz correspond à la montée ou à la descente de l'appareil. Pour faire s'élever le quadrirotor, il suffit d'augmenter la vitesse des quatre moteurs et pour le faire redescendre, il faut réduire cette dernière.

a. Le lacet

La rotation autour de l'axe z est appelée lacet en aéronautique. Lorsque les moteurs tournent à vitesse égale le couple d'anti-rotation est nul et l'hélicoptère ne tourne pas. Pour modifier l'angle de lacet, il faut faire varier la vitesse de rotation sur le couple d'hélices un et trois ou deux et quatre en fonction de l'angle désiré sur le lacet (figure 7).

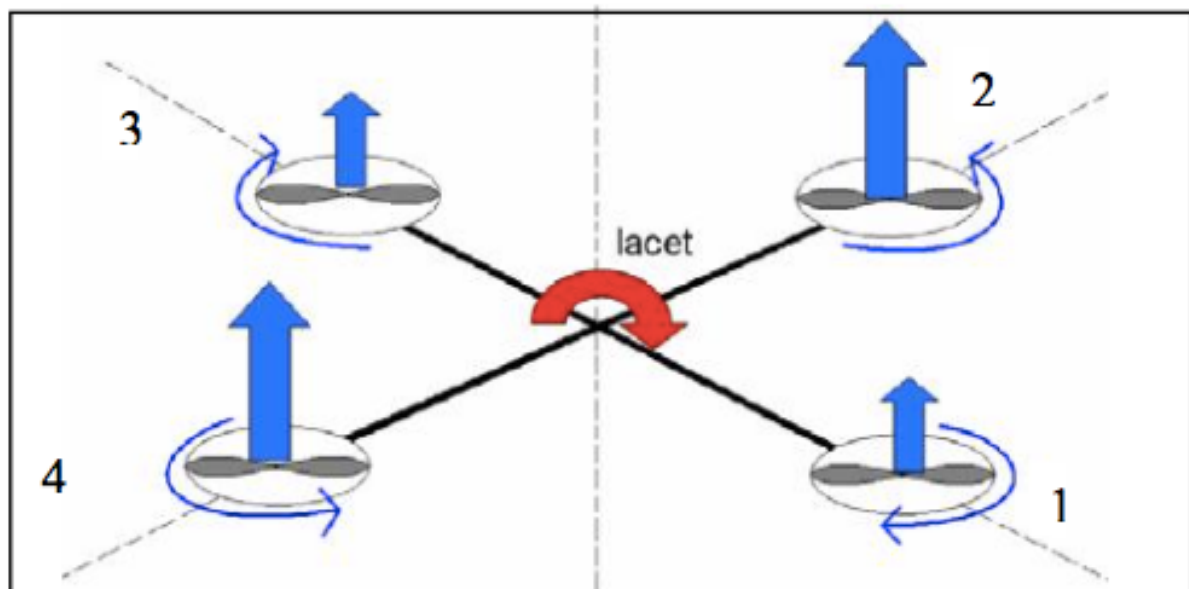


Figure 7 - Lacet

En effet, lorsque l'on projette les forces aérodynamiques exercées par l'air sur la pale, on s'aperçoit qu'un rotor a toujours tendance à faire tourner le micro hélicoptère dans le sens inverse de sa rotation (figure 8).

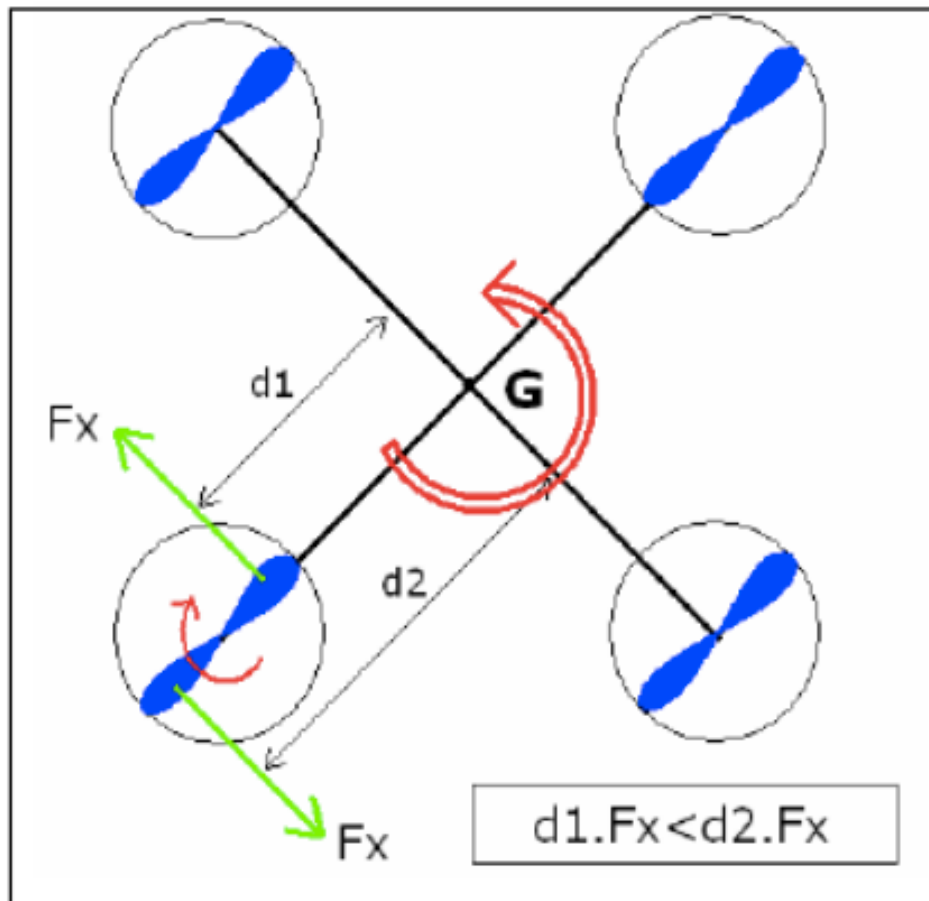


Figure 8 - Forces aérodynamiques

b. Le tangage et le roulis

Le roulis et le tangage sont des mouvements qui provoquent l'inclinaison du quadricoptère selon les axes horizontaux (figure 9). Ces mouvements peuvent être assurés par le drone, en agissant sur la vitesse de l'un des quatre moteurs, et par conséquent sur la force des rotors.

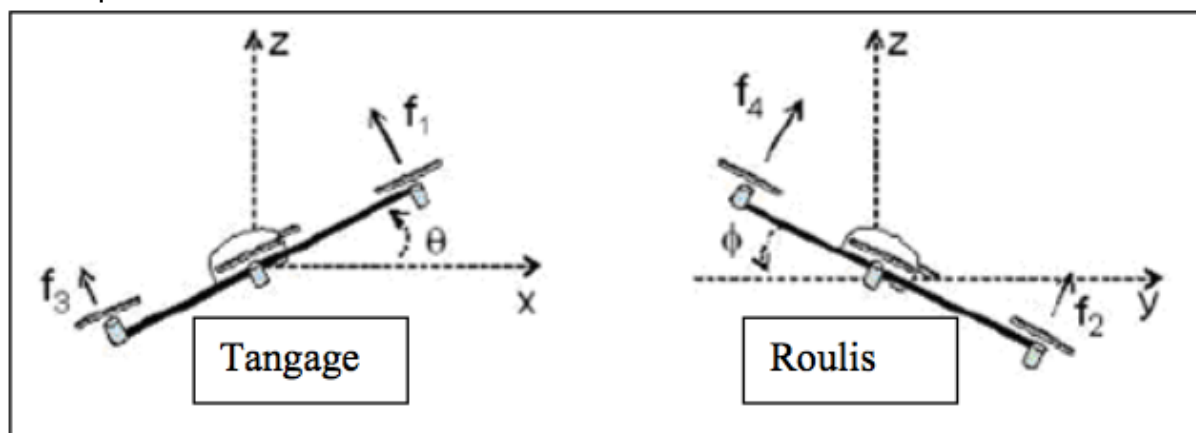


Figure 9 - Tangage et roulis

La rotation autour de l'axe y est appelée tangage en aéronautique. Une variation de l'angle de tangage est obtenue grâce à une différence de vitesse entre le moteur un, et trois (figure 10).

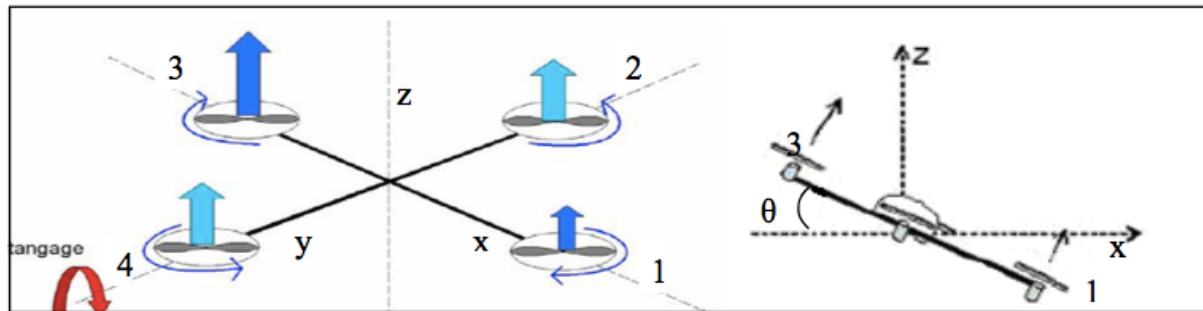


Figure 10 - Mouvement de tangage

En aéronautique, la rotation autour de l'axe x est appelée roulis. Une variation de l'angle de roulis est obtenue grâce à une différence de vitesse entre le moteur deux, et quatre (figure 11).

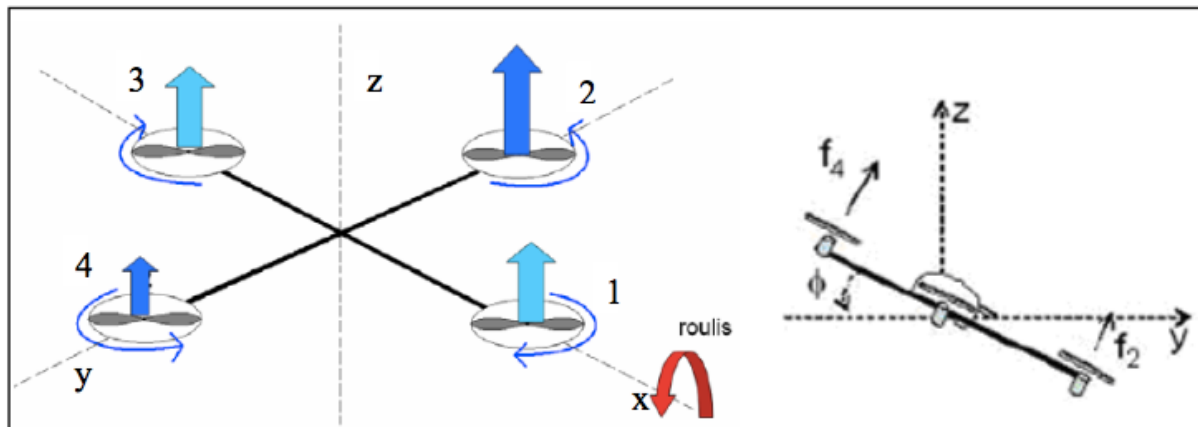


Figure 11 - Mouvement de roulis

3. Les modes de vol

A partir des différents mouvements que le drone peut effectuer, on peut distinguer trois types de vol :

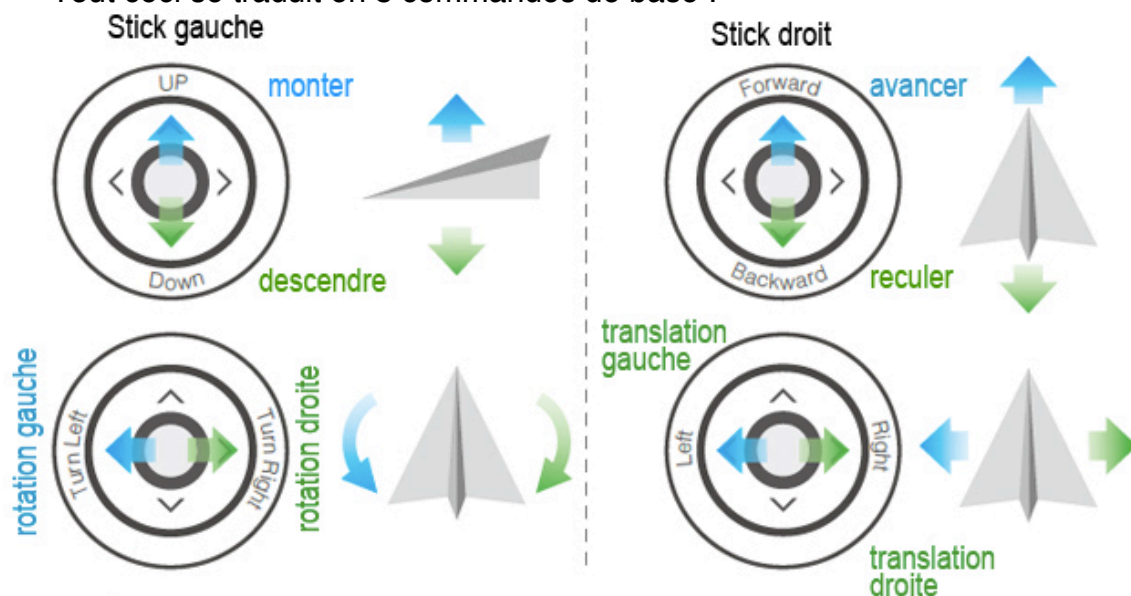
- Le vol vertical : la résultante aérodynamique et le poids total sont deux forces ayant la même direction mais de sens opposés. L'hélicoptère peut monter ou descendre, suivant l'effet aérodynamique soit supérieur ou inférieur au poids de l'appareil.
- Le vol stationnaire : Quand la force de la portance, et celle de pesanteur sont égales et opposées, l'hélicoptère reste immobile. On parle de vol stationnaire.
- Le vol de translation : Le vol de translation correspond à la navigation de l'hélicoptère sur un plan horizontal. Il est assuré en se basant sur les mouvements d'inclinaison tangage, et roulis.

III. Simulateur de trajectoire

Comme susmentionné, le drone peut effectuer quatre mouvements à savoir : la poussée, le lacet, le tangage et le roulis.



Tout ceci se traduit en 8 commandes de base :



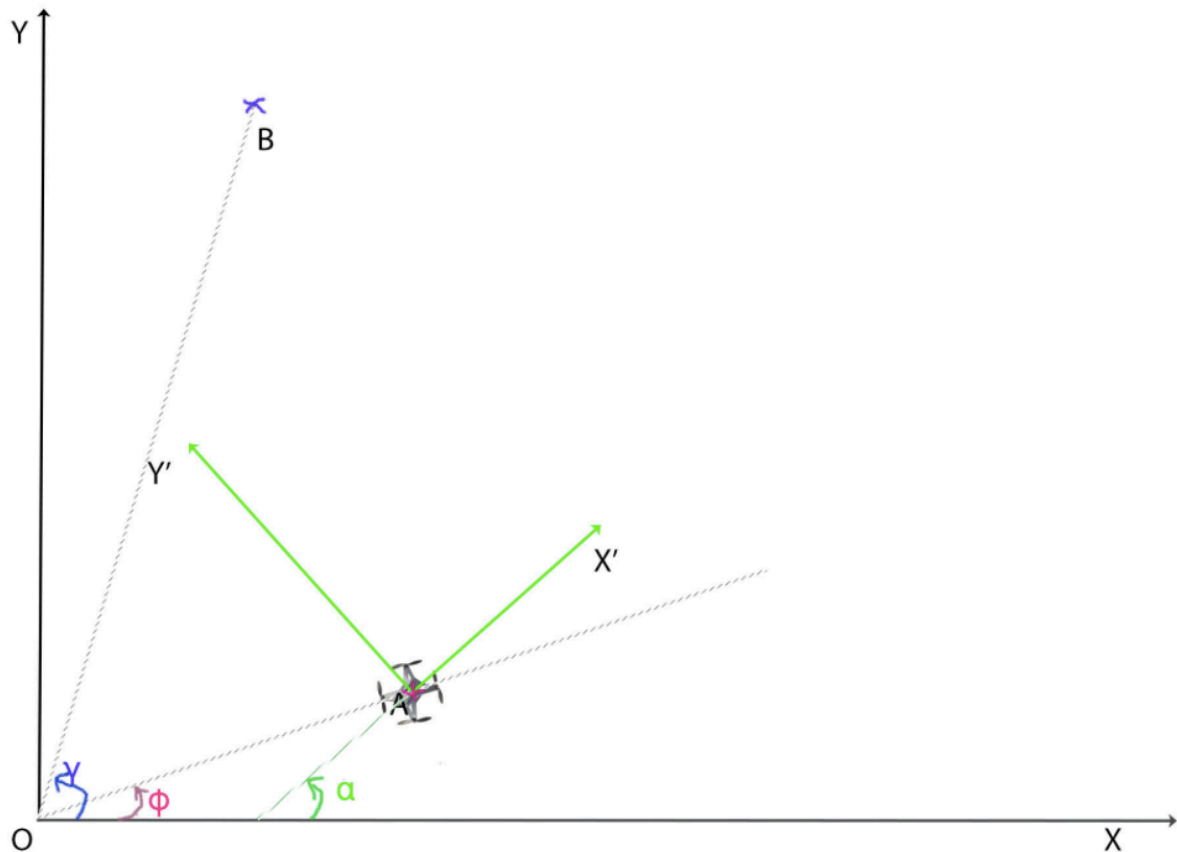
Il y a aussi une commande « décoller » et une commande « atterrir ». On peut utiliser une application Android ou IOS pour piloter le drone comme avec une manette. Il est aussi possible de piloter le drone depuis un ordinateur. Dans les deux cas, la communication entre le dispositif de commande et le drone se fait par un réseau WIFI créé par le drone lui-même auquel il suffit de se connecter.

Lorsqu'on utilise un programme pour piloter le drone, il faudra à chaque fois préciser la vitesse et la durée lors de l'envoi des commandes. Pour avancer (front) pendant 2 secondes (2000 millisecondes) à 80% (0.8) de la vitesse maximale, le bout de commande est formulé comme suit :


```
.after(20000.0, function() {
    this.front(0.8);})
```

1. Paramétrage de l'espace

Soit OXYZ le repère orthonormé fixe lié au sol et AX'Y'Z' le repère orthonormé mobile lié au drone où A est le centre de gravité du drone. Nous travaillons dans un repère cylindrique. Ainsi les directions OZ et OZ' coïncident à tout moment. Pour comprendre le paramétrage en termes d'angle, nous allons considérer un point B. Le drone (en A) est censé aller en B.



$$\overline{OA} = OA \times \cos \phi \cdot \overline{X} + OA \times \sin \phi \cdot \overline{Y}$$

$$\overline{OB} = OB \times \cos \gamma \cdot \overline{X} + OB \times \sin \gamma \cdot \overline{Y}$$

$$\overline{OB} = OB \times \cos(\gamma - \alpha) \cdot \overline{X}' + OB \times \sin(\gamma - \alpha) \cdot \overline{Y}'$$

$$\overline{OA} = OA \times \cos(\phi - \alpha) \cdot \overline{X}' + OA \times \sin(\phi - \alpha) \cdot \overline{Y}'$$

$$\overline{AB} = OB \times \cos(\gamma - \alpha) \cdot \overline{X}' + OB \times \sin(\gamma - \alpha) \cdot \overline{Y}' - OA \times \cos(\phi - \alpha) \cdot \overline{X}' - OA \times \sin(\phi - \alpha) \cdot \overline{Y}'$$

En position initiale, le drone est disposé de manière à ce qu'avancer corresponde à un déplacement suivant les x croissants, et que aller à gauche corresponde à un déplacement selon les y croissants. De ce fait, les valeurs initiales pour α et ϕ sont nulles.

Ce paramétrage est crucial car lorsqu'on demande au drone d'avancer ou d'aller à gauche ou à droite, il le fait selon son repère relatif. Il est donc important de tenir compte de la position et l'angle que fait le repère relatif par rapport au repère absolu. Il faut noter que le simulateur enregistre les coordonnées absolues du drone. Il faut donc à chaque déplacement faire le changement de repère qui s'impose.

2. Le modèle informatique du simulateur

Dans la suite, on va modéliser le drone dans le simulateur par un objet en python, dont les principaux attributs sont les suivants :

- Position dans le repère cartésien (la valeur initiale est fixée par défaut à l'origine du repère absolu) : Il s'agit de tableaux qui enregistrent la position du drone tout au long de ses déplacements comme le ferait la motion capture.
- Vitesse maximale en translation et en rotation

Les commandes de base seront implémentées tout simplement par des méthodes qui mettront à jour la position du drone dans le simulateur. Il y a également des méthodes qui serviront à effectuer la combinaison de deux ou trois commandes de bases en même temps.

3. Planificateur de vol

Nous partons du principe qu'on donne au drone une trajectoire sous forme de nuages de points à suivre. A partir de cela, on va choisir un pas de parcours, distance minimale entre deux points parcourables par le drone. Ainsi, le drone va suivre la trajectoire imposée en effectuant des petits segments. Pour réaliser ces petits segments de chemin, deux approches ont été abordées.

a. Le mode avion

Le drone avance ici comme un avion c'est-à-dire qu'il avance toujours et fait des virages. Dans cette démarche, pour aller d'un point de l'espace à un autre, le drone effectue une rotation sur place jusqu'à viser le point avant d'avancer. Le mouvement du drone est une suite de « tourner » et « avancer ». Lors de l'avancée, le drone fait une combinaison « d'avancer et monter en même temps » ou « d'avancer et descendre en même temps ». Comme il s'agit d'aller d'un point A de coordonnées (x_a, y_a, z_a) à un autre point B de l'espace de coordonnées (x_b, y_b, z_b) , les vitesses qui sont données au drone pour l'avancée et la montée ou la descente sont définies suivant le même rapport que la distance entre les projections des deux points dans le plan OXY et la différence de hauteur entre les deux points. Il suffit de fixer l'une des vitesses.

$$\frac{\text{Vitesse d'avancée}}{\text{Vitesse de montée/descente}} = \frac{\sqrt{(x_a - x_b)^2 + (y_a - y_b)^2}}{|z_a - z_b|}$$

b. Le mode « translation only »

Dans ce cas de figure, le drone effectue uniquement des translations pour aller directement d'un point à un autre sans faire de rotation. Le drone effectue alors une combinaison avancer/reculer + gauche/droite + monter/descendre en même temps. Quittant d'un point A de coordonnées (x_a, y_a, z_a) à un autre point B de l'espace de coordonnées (x_b, y_b, z_b) les vitesses pour chaque commande élémentaires sont définies comme suit :

$$\frac{\text{Vitesse aller à gauche/à droite}}{\text{Vitesse d'avancée/de recul}} = \frac{|y_a - y_b|}{|x_a - x_b|}$$

$$\frac{\text{Vitesse de montée/de descente}}{\text{Vitesse d'avancée/de recul}} = \frac{|z_a - z_b|}{|x_a - x_b|}$$

On détermine finalement les vitesses en fixant l'une d'entre elles.

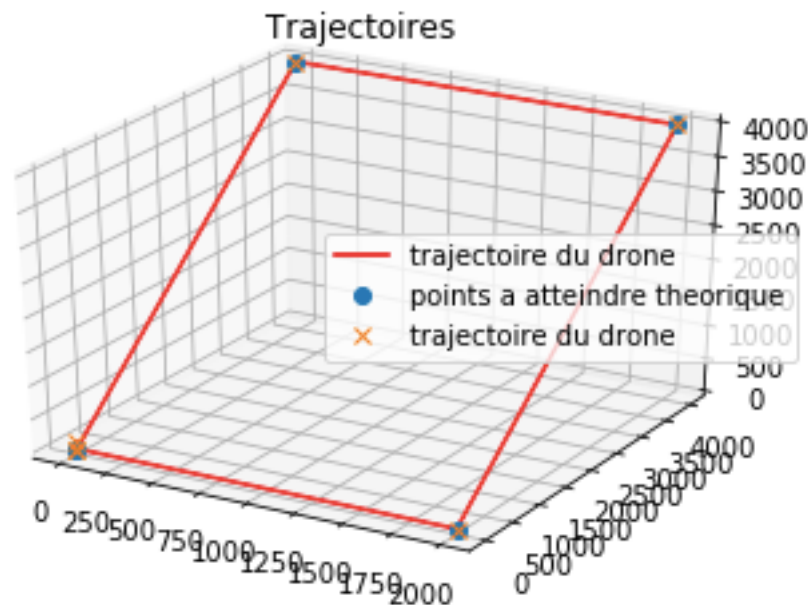
Nous allons privilégier ce mode de de vol par rapport au mode avion car ce dernier est moins naturel et moins fluide.

A la fin de l'exécution du programme, on obtient un script directement exécutable sur le drone. En voici un exemple :

```
var arDrone = require('ar-drone');
var client  = arDrone.createClient();

client.takeoff();
  .after(1000, function() {
    this.down(1);
  })
  .after(20000.0, function() {
    this.front(1);
    this.right(0);
    this.down(0);
  })
  .after(40000.0, function() {
    this.left(1);
    this.up(1.0);
  })
  .after(19999.9, function() {
    this.back(1);
    this.right(2.04452135222e-11);
    this.down(2.04452135222e-11);
  })
  .after(40000.0, function() {
    this.right(1);
    this.down(1.0);
  })
  .after(0, function() {
    this.stop();
    this.land();
  });
```

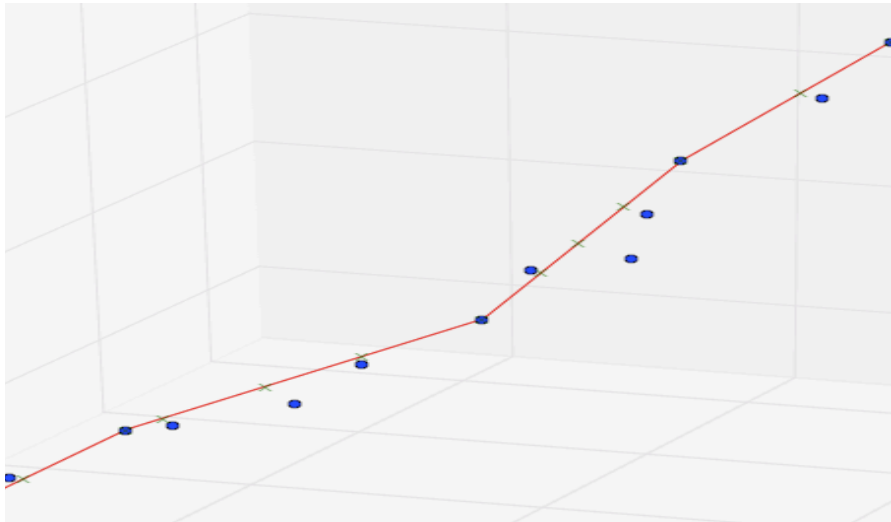
Dans cet exemple, le drone va décoller, descendre en vitesse maximale (1 pour 100%) pendant une seconde (1000 millisecondes), avancer en vitesse maximale pendant 20 secondes, aller vers la gauche tout en montant en vitesse maximale pendant 40 secondes, reculer en vitesse maximale pendant environ 2 secondes, aller vers la droite tout en descendant à vitesse maximale pendant 40 secondes et enfin se poser. Le simulateur donne ceci :



4. Estimation de l'erreur

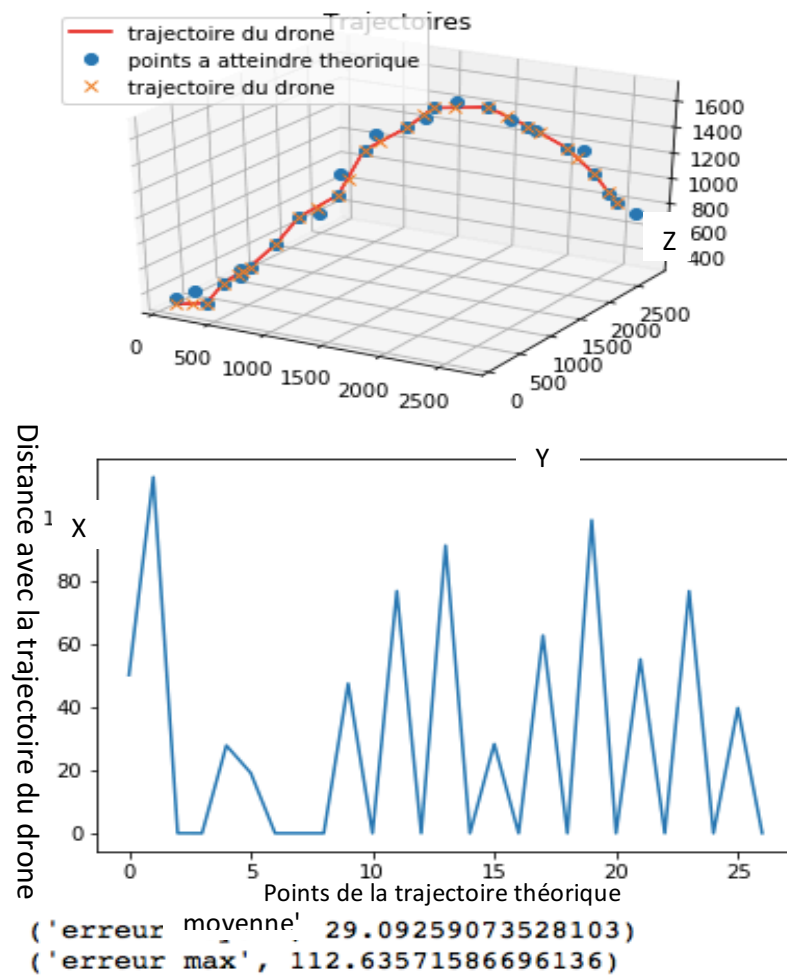
Il est assez délicat de définir une erreur ici puisque nous avons à faire à des courbes paramétrées dans l'espace. En effet, il était difficile de trouver une formule de calcul de l'erreur qui puisse rendre compte fidèlement de la similarité entre les deux trajectoires. Qu'à cela ne tienne, on peut estimer en calculant la distance entre les points de la trajectoire théorique et les points de la trajectoire réelle du drone. Même dans l'hypothèse où le drone parcourait tous les points de la trajectoire théorique imposée, la trajectoire du drone enregistrée avec la mocap paraîtra continue (densité de points élevée). Il faudra alors, ré-échantillonner la trajectoire réelle du drone de façon à obtenir un même nombre de points que celui de la trajectoire théorique. Pour chaque point de la trajectoire la plus dense en points, on ne va garder que le point le plus proche sur l'autre trajectoire.

Dans l'exemple de la figure ci-dessous, on ne retiendra que les points en vert de la trajectoire rouge (plus dense en point que la trajectoire bleue). Il s'agit en fait des projections (approximatives) orthogonales des points bleus sur la trajectoire rouge.



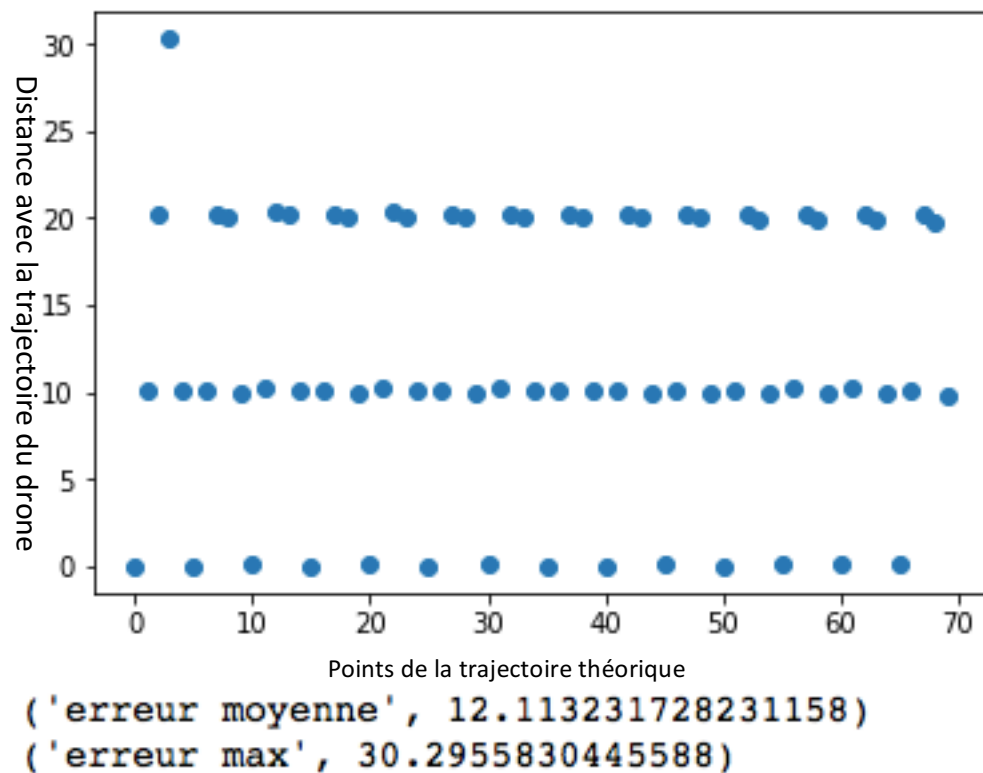
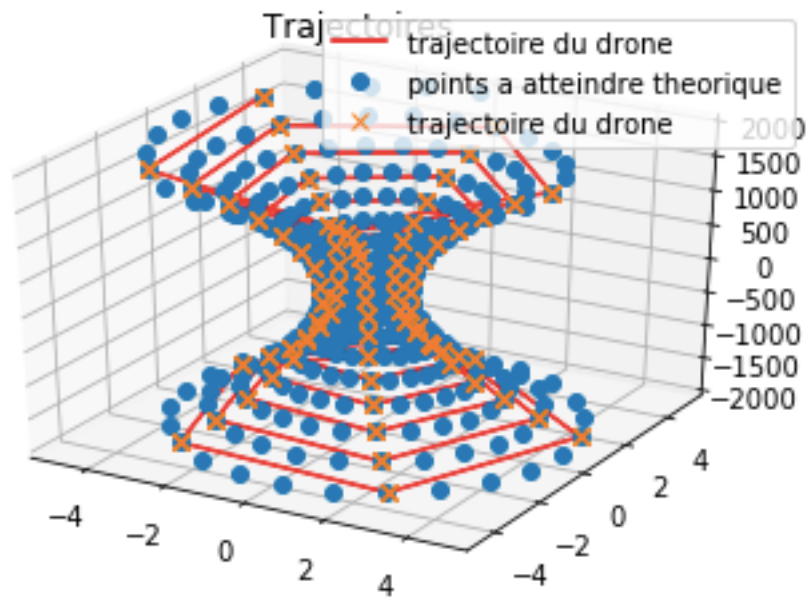
Ce n'est qu'après cela que nous pourrions calculer des distances point à point afin d'établir une moyenne et trouver un écart max.

Voici un exemple de ce que l'on obtient. Ici les points de la trajectoire théorique sont espacés aléatoirement suivant une loi uniforme sur $[0;200]$ cm sur chaque axe du repère et le pas du simulateur a été mis à 100 cm.



A chaque fois que le simulateur passe par un point de la trajectoire théorique, nous avons une erreur nulle d'où les retours à zéro récurrents.

Un autre exemple avec une spirale comme trajectoire théorique.



On remarque sur ce deuxième exemple avec l'erreur représentée sous forme de nuage de points, que cette dernière est quasiment périodique. Ceci est sans doute dû au fait que la trajectoire théorique présente aussi une périodicité (en x et en y) et que l'on simule un drone parfait qui reproduirait un parcours quasi identique pour un même jeu de données.

On peut également appliquer cette méthode pour calculer l'erreur suivant les différents axes en se servant des projections des trajectoires afin de voir si une direction présente plus d'instabilités que les autres.

Conclusion générale

Arrivé au terme de ce projet, j'ai pu découvrir et travailler avec des technologies qui m'étaient inconnues auparavant. La capture de mouvement est un dispositif que j'imaginais exclusivement réservé au domaine du cinéma et des jeux vidéo et dont je ne connaissais pas tout à fait le fonctionnement. C'était également une première fois pour moi de pouvoir manipuler un drone. Il était question tout au long de ce projet, d'avoir une maîtrise de ces technologies et d'utiliser la motion capture pour étudier les déplacements et la stabilité du drone. Pour ce faire, il était nécessaire de s'approprier les différents outils de travail puis de passer à la phase de programmation du drone.

Cependant, le drone présentait une trop grande instabilité et ne permettait pas de faire des tests pratiques directement sur ce dernier. En effet, celui-ci a subi des chocs lors des premiers essais de vol et est devenu très instable. Par ailleurs, il y avait une inadéquation entre la taille de la plateforme Amigo et la taille du drone. Il n'y avait pas suffisamment d'espace pour permettre au drone de se mouvoir sans heurter des obstacles (murs, table...). Suite à tout ceci, il n'a pas été possible d'enregistrer les déplacements du drone avec la motion capture comme prévu. En fin de compte, il a fallu redéfinir l'objectif et s'orienter vers la mise au point d'un simulateur de trajectoire pour compenser les problèmes techniques avec le drone.

L'utilisation combinée de la motion capture et du drone offre encore beaucoup de possibilités pour continuer ce projet. Afin de permettre au drone de corriger sa trajectoire et de rester au plus proche de la trajectoire théorique, il serait pertinent d'utiliser la position du drone en temps réel dans le programme. Cela passe l'ajout des plugins spéciaux au logiciel Nexus de Vicon ([Datastream SDK](#)) qui permettront d'extraire en streaming les données de position des marqueurs au cours de l'acquisition. Il serait même possible dans le même ordre d'idée de programmer le drone de façon à qu'il suive une cible repérée par un ou plusieurs marqueurs. De plus, le drone étant équipé d'une caméra, la reconnaissance de couleur incluse dans les capacités de traitement de son système embarqué peut être mise à profit. En effet, il serait possible de dessiner un circuit 2D à suivre sur le sol (en blanc par exemple).

Bibliographie

1. Alberto MENACHEN, "Understanding motion capture for computer animation — second edition", Burlington, USA, 2011.
2. <https://www.federation-drone.org/les-drones-dans-le-secteur-civil/histoire-du-drone/>
3. AYAD redouane, "Conception et réalisation d'un drone quadrirotor", Mémoire pour l'obtention d'un magistère en électronique, sous la direction de N. NOUIBAT, Oran, Université des Sciences et Technologie d'Oran, 2010.