

## TABLE DES MATIÈRES

	Page
INTRODUCTION .....	1
CHAPITRE 1 ÉTAT DE L'ART .....	5
1.1 Définition du terme « Drone » .....	5
1.2 Historique.....	6
1.3 Applications .....	7
1.4 Marché actuel.....	9
1.4.1 Principaux acteurs.....	9
1.4.2 Régulation.....	10
1.4.3 Technologies utilisées.....	11
1.5 Quelles technologies et utilisations restent à explorer? .....	13
CHAPITRE 2 LE PROJET AMOOS .....	15
2.1 Description du projet.....	15
2.2 Simulations virtuelles.....	16
2.2.1 Trajectoire STK .....	17
2.2.2 Utilisation d'Xplane.....	35
2.3 Simulations terrestres : Quadricoptères .....	38
2.4 Retombées dans le domaine civil : AMOSS.....	38
CHAPITRE 3 CONSTRUIRE AVEC LES TECHNOLOGIES COMMERCIALES .....	41
3.1 Construction du drone.....	41
3.1.1 La structure : Walkera.....	41
3.1.2 DJI Wookong.....	43
3.1.3 Bras articulé .....	44
3.2 Architecture matérielle.....	45
3.2.1 À bord du drone .....	45
3.2.2 Station locale au sol .....	48
3.2.3 Poste de pilotage à distance .....	51
3.2.4 Simulateur de vol HeliMod.....	52
3.3 Essais en vol.....	53
CHAPITRE 4 SUPPRESSION DE LA STATION LOCALE AU SOL .....	57
4.1 Nouvelles contraintes.....	57
4.2 Nouvelle architecture .....	58
4.2.1 La Raspberry Pi.....	58
4.2.2 Les transmissions de données .....	59
4.2.3 Les systèmes de sécurité .....	63
4.2.4 Les capteurs .....	65
4.2.5 CAN-Bus Wookong.....	67
4.2.6 Prototype F450.....	68

4.2.7	Poste de control.....	70
4.2.8	Réseau de communication entre drones.....	71
4.3	Créer un système autonome.....	73
4.3.1	Algorithme de navigation.....	73
4.3.2	Asservissement des commandes.....	86
4.3.3	Détections de collision.....	91
4.3.4	Architecture logicielle du drone.....	94
CHAPITRE 5 UTILISATION D'AVIONIQUES G1000.....		99
5.1	Introduction au système Garmin 1000.....	99
5.2	Installation du G1000.....	100
5.3	Débridage.....	101
5.4	Interfaçage avec Xplane.....	104
5.4.1	Câble d'adaptation et HSDB.....	104
5.4.2	Plug-in et modèle 3D.....	107
5.5	Couplage avec le Drone.....	110
CHAPITRE 6 TESTS EN VOL.....		113
6.1	Tests en extérieur.....	113
6.2	Tests en deux dimensions.....	115
6.3	Tests en cage de vol.....	116
CHAPITRE 7 DISCUSSION RELATIVE AUX AMÉLIORATIONS FUTURES.....		121
7.1	FOTAFMS.....	121
7.2	Amélioration des Avioniques.....	122
7.3	TCAS et trajectoires d'évitement.....	123
7.4	Réalité augmentée et support de suspension « gimbal ».....	123
7.5	Détection optique d'obstacles.....	124
7.6	Parachute de sécurité.....	125
7.7	Nouveau calculateur de bord.....	125
7.8	Nouvelle charge utile.....	126
CONCLUSION.....		127
RECOMMANDATIONS.....		129
ANNEXE I	Gestion des UAVs par Transport Canada.....	131
ANNEXE II	Schéma du Simulateur HeliCrew.....	133
ANNEXE III	Création du bras articulé.....	135
ANNEXE IV	Interfaces de pilotage LabVIEW.....	139
ANNEXE V	Sélecteur de canaux.....	141
ANNEXE VI	Diagrammes de la structure logicielle du drone.....	145

ANNEXE VII	Système G1000 Basique 2 écrans .....	147
ANNEXE VIII	Création du banc pour GDUs.....	149
LISTE DES RÉFÉRENCES BIBLIOGRAPHIQUES .....		155



## LISTE DES TABLEAUX

	Page
Tableau 3-1	Inventaire des différents modules nécessaire au drone RC .....46
Tableau 4-1	Inventaire des différents modules nécessaires au drone autonome LTE et satellite .....68
Tableau 6-1	Qualités de vol en contrôle manuel avec architecture commerciale.....114
Tableau 6-2.	Qualités de vol en contrôle automatique avec architecture commerciale (chapitre 3).....115
Tableau 6-3	Qualités de contrôlabilité du modèle en 2 dimensions .....116
Tableau 6-4	Qualités de vol en contrôle manuel avec architecture personnalisée.....116
Tableau 6-5	Qualités de vol en contrôle d'altitude automatique avec architecture personnalisée.....117



## LISTE DES FIGURES

		Page
Figure 2.1	Exemple de simulation STK avec la constellation des satellites Globalstar .....	17
Figure 2.2	Repère ECI (« Earth Central Inertial »), tirée de Landry 2014.....	19
Figure 2.3	Repère ECEF (« Earth Centered Earth Fixed »), tirée de Landry 2014 ...	20
Figure 2.4	Repère ENU .....	20
Figure 2.5	Repère NED (« North, East, Down »), tirée de Landry 2014.....	21
Figure 2.6	Repère du mobile, tirée de Landry 2014.....	22
Figure 2.7	Paramètres d'une orbite elliptique plate, tirée de Landry 2014.....	23
Figure 2.8	Paramètres d'une orbite autour de la terre, tirée de Landry 2014 .....	24
Figure 2.9	Calcul d'une orbite sous Matlab.....	26
Figure 2.10	Orbite importée de Matlab à STK.....	27
Figure 2.11	Changement de repère du satellite à l'ECEF.....	28
Figure 2.12	Latitudes et longitudes, tirée de Landry 2014.....	29
Figure 2.13	Placement d'une latitude et d'une longitude sur un ellipsoïde, tirée de Landry 2014 .....	29
Figure 2.14	Exemple de trajectoire d'approche en spirale et directe sous Matlab .....	33
Figure 2.15	Trajectoire d'approche finale .....	33
Figure 2.16	UAV en 3D sous SolidWorks et DraftSight .....	34
Figure 2.17	UAV en 3D sous SketchUp et STK.....	35
Figure 2.18	Chaine de conversion de ".SLDASM" à ".dae" .....	35
Figure 2.19	ASSP-1B modifié.....	36
Figure 2.20	Tableau de bord ASSP-1B avant et après modifications .....	37
Figure 2.21	HeliCrew simulant un ASSP-1B.....	37

Figure 3.1	Walkera QR X800, tirée de Walkera 2015 .....	42
Figure 3.2	DJI Wookong, tirée de AliExpress 2015 .....	43
Figure 3.3	Bras articulé .....	45
Figure 3.4	Schéma bloc de l'architecture de notre drone.....	47
Figure 3.5	Photo du drone et des équipements à bord.....	47
Figure 3.6	Acheminement du signal vidéo.....	49
Figure 3.7	Acheminement des commandes de vol.....	50
Figure 3.8	Interface de pilotage LabVIEW .....	50
Figure 3.9	Réseau de communication entre l'ÉTS et le drone .....	52
Figure 3.10	Simulateur de Vol HéliMod.....	53
Figure 3.11	Équipements requis pour l'opération de la station locale au sol.....	56
Figure 4.1	Raspberry Pi, tirée de Clubic .....	58
Figure 4.2	Couverture réseau LTE mondiale, tirée de Bennett 2014.....	60
Figure 4.3	Clé LTE/USB/TELUS, tirée de Huawei 2014.....	60
Figure 4.4	Téléphone GSP-1700 Globalstar, adaptée de Globalstar 2014.....	61
Figure 4.5	Architecture matérielle du sélecteur de canaux .....	63
Figure 4.6	Détection de panne Arduino .....	64
Figure 4.7	Capteurs UM7, Ultrason et Lidar, respectivement, tirée de ChRobotics...67	
Figure 4.8	Schéma bloc de Architecture matérielle du drone LTE et satellite .....	69
Figure 4.9	Prototype du F450 modifié à la saveur du projet.....	70
Figure 4.10	Exemple d'une équipe de drone en réseau local .....	72
Figure 4.11	Explication des trajectoires.....	74
Figure 4.12	Correction de cap du drone.....	75
Figure 4.13	Bruits et contraintes externes agissant sur le drone et ses capteurs .....	79

Figure 4.14	Bruits dus aux aléas, au vent et au décalage du magnétomètre respectivement .....	80
Figure 4.15	Exemple de navigation de base avec tous les bruits cumulés .....	80
Figure 4.16	Exemple de commandes synchronisées sur 3 axes .....	82
Figure 4.17	Exemple de correction de navigation avec pilotage des 3 axes .....	83
Figure 4.18	Trajectoires avant et après correction de la dérive des capteurs .....	84
Figure 4.19	Boucle d'asservissement classique .....	86
Figure 4.20	Boucle d'asservissement simplifiée du drone .....	87
Figure 4.21	Qualification du drone en cage de vol. ....	88
Figure 4.22	Asservissement de l'axe vertical .....	89
Figure 4.23	Dérivation et lissage de la position du drone en fonction des commandes .....	89
Figure 4.24	Correction par PID .....	91
Figure 4.25	Exemple de détection de distance minimale entre deux trajectoires. (Trajectoires en bleu et vert et distance minimale en rouge) .....	94
Figure 4.26	Architecture du partage de données entre applications .....	96
Figure 5.1	GDU's tels que réceptionnés .....	100
Figure 5.2	Aperçu final du socle G1000 .....	101
Figure 5.3	Page de config GDU TEST .....	103
Figure 5.4	PCB d'un GDU .....	103
Figure 5.5	Exemple de décodage de trames HSDB .....	106
Figure 5.6	Chaîne de réception des .....	108
Figure 5.7	Étapes pour repositionner le modèle 3D de l'avion sous Xplane .....	109
Figure 5.8	Sphère en 3D sous Plane-Maker .....	110
Figure 5.9	Chaîne complète pour affichage sur G1000 .....	111
Figure 6.1	Test de maintien d'altitude avec PID .....	118

Figure 6.2      Test de maintien d'altitude avec PID et correcteur d'erreur statique.....119

## LISTE DES ABRÉVIATIONS, SIGLES ET ACRONYMES

AHRS	Attitude and Heading Reference System
AMOOS	Autonomous Mission for On-Orbit Servicing
AMOSS	Autonomous Mission for On-Site Servicing
AP	Automatic Pilot
ARINC	Aeronautical Radio INCorporated
CANBus	Controller Area Network Bus
DCM	Direction Cosine Matrix
DLL	Dynamic Link Library
ECEF	Earth-Centered, Earth-Fixed
ECI	Earth-centered inertial
ESC	Electronic Speed Control
FMS	Flight Management System
FPV	First-person view
GDC	Garmin Data Computer
GEA	Garmin Engine/Airframe
GIA	Garmin Integrated Avionics
GMA	Garmin Managment Audio
GPIO	General Purpose Input/Output
GRS	Garmin Reference System
GTX	Garmin Transponder X
HDMI	High Definition Multimedia Interface
HSDB	High Speed Data Bus
I2C	Inter Integrated Circuit
IMU	Inertial measurement unit
INS	Inertial Navigation System
IOSD	Integrated On-Screen Display
IP	Internet Protocol
ISU	International Space University

## XXIV

LCD	Liquid Crystal Display
LEO	Low Earth orbit
LIDAR	LIght Detection And Ranging
LTE	Long Term Evolution
MAC	Media Access Control
MEMS	Microelectromechanical systems
MFD	Multi-Function Display
NMEA	National Marine Electronics Association
PC	Personal Computer
PCB	Printed Circuit Board
PFD	Primary Flight display
PPM	Pulse Position Modulation
PSI	Serial Peripheral Interface
PWM	Pulse With Modulation
RAM	Random Access Memory
RF	Radio frequency
RPAS	Remotely Piloted Aircraft Systems
SSH	Secure SHell
TCAS	Traffic Collision Avoidance System
TCP/IP	Transmission Control Protocol / Internet Protocol
UART	Universal Asynchronous Receiver Transmitter
UAS	Unmanned Aerial Systems
UAV	Unmanned aerial vehicle
USB	Universal Serial Bus
VBA	Visual Basic for Applications
Wifi	Wireless Fidelity

## LISTE DES SYMBOLES ET UNITÉS DE MESURE

$a$	Demi grand axe de l'ellipse du satellite (m)
$b$	Demi petit axe de l'ellipse du satellite (m)
$e$	Excentricité de l'ellipsoïde du satellite
$E$	Anomalie excentrique de l'ellipsoïde du satellite
$f$	Anomalie vraie de l'ellipsoïde du satellite
$r$	Distance radiale du satellite (m)
$\mu$	Anomalie moyenne du satellite
$t$	Temps (s)
$S$	Position du satellite
$P$	Périgée de la trajectoire du satellite
$C$	Centre de masse de la terre
$\omega$	Argument du périégée de la trajectoire du satellite (deg)
$i$	Inclinaison de la trajectoire du satellite (deg)
$\Omega$	Longitude du nœud ascendant de la trajectoire du satellite
$X_{sat}$	Position du satellite selon l'axe x (m)
$Y_{sat}$	Position du satellite selon l'axe y (m)
$Z_{sat}$	Position du satellite selon l'axe z (m)
$\lambda$	Latitude (deg)
$\phi$	Longitude (deg)
$h$	Altitude (m)
$a$	Demi grand axe de la terre (m)
$1/f$	Coefficient d'aplatissement de la terre (m)
$e$	Excentricité de la terre

$\phi$	Roulis de l'aéronef (deg)
$\theta$	Tangage de l'aéronef (deg)
$\psi$	Lacet de l'aéronef (deg)
$C_N^E$	Matrice de rotation du repère ENU au repère ECEF
$C_L^N$	Matrice de rotation du repère NED au repère ENU
$C_L^B$	Matrice de rotation du repère Mobile au repère NED
$\begin{bmatrix} X \\ Y \\ Z \end{bmatrix}_{\text{Repère}}$	Coordonnées cartésienne d'un élément dans un repère particulier (m)
$(a,b,c)$	Coefficients d'une droite en 2D
$(x,y)$	Coefficients d'un point dans un repère 2D
$D$	Distance parcourue par le drone (m)
$\Delta$	Discriminant d'une équation au second degré
$C_{corr}$	Correction de cap demandée au drone (deg)
$(x(s),y(s),z(s))$	Coefficients d'un point de paramètre s dans un repère 3D
$f(s, t)$	Fonction dépendant des paramètres s et t
$\frac{\partial f}{\partial t}$	Dérivée partielle de $f$ en fonction de t
$\frac{\partial f}{\partial s}$	Dérivée partielle de $f$ en fonction de s

## INTRODUCTION

Les UAVs (de l'anglais *Unmanned Aerial Vehicles*) ou plus communément appelés « drones » sont en train d'envahir l'espace aérien (Baillie, Keith et Dave, 2014). Si autrefois leur utilisation et leur développement étaient extrêmement chers et donc réservés aux militaires, aujourd'hui la forte baisse des prix de leur technologie a permis une importante démocratisation des drones. Si bien qu'on trouve des nouvelles applications pour les drones tous les jours : de la livraison de colis à la surveillance privée en passant par les défibrillateurs volants, le choix est vaste. (Austin, 2010, DeGarm, 2004).

Mais le domaine d'évolution des drones ne s'arrête pas au ciel, des UAVs spatiaux sont en cours de conception pour intervenir dans l'espace et réparer les satellites endommagés et/ou arrivés en fin de vie (Landry, 2014). Ces technologies, encore à l'essai sont précurseurs dans le domaine du pilotage de drones à très longue portée. Et l'idée d'envoyer des machines autonomes dans le but de remplacer les humains dans les environnements hostiles et dangereux est très intéressante.

Mais différentes questions émergent : Est-il possible d'envoyer de petits drones de type quadricoptères pour effectuer des missions à la place des humains? A-t-on aujourd'hui atteint un niveau de performance technologique civile assez élevé et assez abordable pour mener à bien ces missions? Ces drones, toujours plus nombreux et de plus en plus gros, sont-ils capables de s'intégrer dans les espaces aériens civils et dans les zones urbaines sans les mettre en péril ? Peut-on recréer un environnement de pilotage à très longue distance suffisamment professionnel pour assurer l'introduction des UAVs dans notre quotidien en toute sécurité?

Pour répondre à ces questions, nous commencerons par étudier les missions des drones autonomes orbitaux. Leur autonomie d'action, leurs trajectoires et de leur planification sont très poussées et peuvent facilement être adaptées à moindre échelle. Le domaine spatial à toujours apporté de nombreuses innovations dans le civil (ISU, 2014), prendre exemple sur

de telles missions est donc une approche très intéressante pour introduire des UAVs dans notre espace aérien.

Mais les simulations et approches spatiales posent de nombreux problèmes calculatoires dépendant de plusieurs paramètres qui compliquent l'affichage de leur positionnement :

- Les orbites : circulaires ou elliptiques, plates ou inclinées, les orbites dépendent de facteurs indépendants les uns des autres qui modifient les calculs de positionnement dans l'espace.
- Les repères : pour simuler les approches d'un UAV sur un mobile en mouvement (un satellite par exemple) il est nécessaire de le prendre comme référence et de s'approcher doucement autour de lui. Cependant, d'un point de vu terrestre, le positionnement du drone nécessite de nombreux changements de repères pour que sa trajectoire soit fluide.

Une fois ces différents problèmes calculatoires écartés, nous serons capables d'exploiter ces principes à moindre échelle sur des drones de type quadricoptère. Mais là aussi des problèmes physiques liés au vol non habité nous poseront des difficultés technologiques qu'il nous faudra résoudre :

- La distance : le but étant de contrôler un drone pouvant se situer dans une zone très éloignée. Il sera nécessaire d'identifier et d'implémenter des technologies de communication capables d'émettre à très longue distance.
- La robustesse : l'aéronautique est un domaine qui ne laisse pas de place à l'imprécision et encore moins au doute. Les systèmes développés et présentés dans ce document ont tous été pensés et conçus de manière à ne laisser aucune faille pouvant détériorer un vol.
- La sécurité : nombreux sont les paramètres qui peuvent mettre en péril un vol et rapidement causer une catastrophe. Aussi la principale priorité des travaux menés au cours de ce mémoire aura été de toujours privilégier l'aspect sécuritaire des UAVs en tous temps.

Pour répondre aux différents obstacles énoncés, nous proposerons les contributions suivantes pour simuler les trajectoires et missions spatiales :

- La création d'un algorithme sous Matlab à base de trajectoires orbitales et de changements de repères. Celui-ci sera capable de calculer les coordonnées d'un mobile dans l'espace et de lui faire faire des approches de tous types (spirales, directe, par bonds, etc.)
- La réalisation d'un plug-in pour introduire les coordonnées calculées sous Matlab directement dans le logiciel STK (System Tool Kit) et visualiser les trajectoires réalisées avec des modèles d'UAVs en 3 dimensions et en temps réel.
- La configuration d'un simulateur de vol afin de pratiquer des mises en orbites et des entrées atmosphériques sous Xplane, reproduisant ainsi un poste de pilotage à distance pour simuler des interventions spatiales non habitées.

Pour réaliser et tester nos prototypes de drones, nous proposerons les contributions suivantes :

- La réalisation de plusieurs prototypes de drones intégrant différentes architectures matérielles capables d'être pilotés à très grande distance par LTE durant une trentaine de minutes.
- L'intégration d'un ordinateur de bord sur les drones pour permettre la gestion de périphériques complexes et l'intégration d'algorithmes de commandes avancés.
- La création de systèmes de sécurité manuels et automatiques pour détecter et gérer d'éventuelles pannes de l'ordinateur de vol.
- L'élaboration d'une architecture logicielle innovante, capable de faire de la gestion de vol, de partager des informations entre drones et de maintenir une liaison avec une station au sol à très longue distance.
- La réalisation d'un algorithme de navigation par points de cheminement pour piloter un UAV sur 4 axes de contrôle et son test sous Matlab dans un environnement fortement bruité.
- La mise au point d'un algorithme de détection de collisions entre drones à courte distance capable de prioriser les ordres de passage.

- L'interfaçage d'avioniques certifiées de type G1000 pour réceptionner les informations de vol des drones et les piloter depuis un cockpit déporté.

Pour développer ces contributions, ce mémoire se divise en sept chapitres : le premier fera office d'introduction au domaine des drones, leurs dénominations, leur histoire et leurs utilisations. Son but est de faire un état de l'art et de montrer quelles technologies et méthodologies restent à développer dans le domaine des drones civils. Le second chapitre sera consacré au développement de techniques de simulation de trajectoires d'approches de drones spatiaux sur des satellites. Il traitera des aspects relatifs aux calculs de trajectoires et à l'esthétique de ces simulations. Il fera également office d'introduction aux changements de repères qui nous serviront tout au long de ce mémoire. Le troisième chapitre traitera de la sélection et de la conception d'un drone avec un maximum de technologies commercialisées, le but étant de créer un système muni d'une station locale au sol capable de transmettre des ordres de commandes à très longue distance. Ce chapitre servira également d'introduction au prototypage et à l'architecture des drones, ce qui nous sera utile pendant la suite du mémoire. Dans le quatrième chapitre, nous verrons comment se passer de cette station locale au sol en employant des technologies bien plus basiques et non spécifiques aux drones. L'utilisation d'un ordinateur de bord personnalisé nous obligera à développer nos propres algorithmes de navigation et notre propre architecture logicielle pour gérer les différents modes de pilotage et s'ouvrir au vol en formation. Le cinquième chapitre sera consacré à l'adaptation et l'interfaçage d'avioniques de type G1000 pour offrir à nos drones un pilotage plus proche de la réalité commerciale, et pour pallier aux problèmes de sécurité, de robustesse et d'efficacité. Le chapitre six montrera les résultats obtenus en vol et le septième chapitre conclura ce mémoire en évoquant les différentes évolutions innovantes qui pourraient composer la suite de ce projet.

# CHAPITRE 1

## ÉTAT DE L'ART

Le terme « drone » est aujourd'hui florissant dans tous les médias et partout sur internet, mais l'utilisation du terme est souvent abusive et floue. Ce premier chapitre éclaircira le lecteur sur la définition d'un drone, son histoire et ses récentes applications. Cet état de l'art permettra également de faire un bilan sur les développements et les technologies rencontrés aujourd'hui dans les milieux privés, industriels et militaires. Ainsi nous pourrons voir quels domaines n'ont pas encore été explorés et orienter nos recherches vers des solutions innovantes notamment dans le domaine des communications longue distance.

### 1.1 Définition du terme « Drone »

Bien que le terme « drone » semble aujourd'hui définir un aéronef télé-piloté, il désigne en réalité toute forme de robot ou véhicule terrestre, aérien ou aquatique. Ainsi, un sous-marin ou encore un module d'exploration spatial peuvent également être qualifiés de drones.

Cependant il existe plusieurs termes plus spécifiques désignant les avions télé-piloté en fonction de leurs spécificités.

Le terme le plus commun étant UAV (pour Unmanned Aerial Vehicle), il désigne un avion qui est capable d'opérer sans pilote à bord (OACI, 2011), il englobe tous les types d'aéronefs sans pilotes en ne faisant aucune distinction. Cette définition de l'OACI fait office de référence auprès de beaucoup d'organismes mais certains pays comme le Royaume-Uni préfèrent utiliser l'acronyme UA (Unmanned Aircraft) (CAA, 2012).

Le mot UAS (pour Unmanned Air Systems), très utilisé par les militaires, désigne l'UAV lui-même et tous les éléments qui opèrent sans pilote à bord (OACI, 2011).

Une sous-classe des UAVs est appelée RPAS (pour Remote Piloted Aircraft Systems), elle concerne les UAVs qui ne sont pas totalement autonomes, autrement dit les aéronefs disposant toujours d'un pilote aux commandes derrière l'électronique. Le terme RPAS désigne à la fois l'appareil, ses moyens de communication et la ou les stations au sol (OACI, 2011).

Pour finir, on retrouve le terme SUA (en anglais « Small Unmanned Aircraft ») qui représente une sous-classe d'UAV dont le poids ne dépasse pas 20 Kg à vide (CAA, 2012).

Il existe d'autres termes plus rares comme RPV (Remotely Piloted Vehicles) ou encore VTUAV (Vertical Take-Off and Landing Tactical UAV), l'Office of the Secretary of Defense offre une liste regroupant tous ces acronymes (U.S., 2005).

Tous ces termes ne sont pas exclusifs les uns des autres, en effet un UAV peut par exemple être à la fois un SUA et un RPAS (ce qui sera notre cas dans la suite de ce mémoire).

Cependant tout au long de ce document, nous désignerons nos avions sans pilotes à travers les termes « drone » et « UAV » pour faciliter la compréhension du lecteur et lui éviter d'avoir à penser systématiquement à une classe particulière.

## **1.2 Historique**

Ce paragraphe retrace brièvement l'évolution des aéronefs sans pilote depuis ses débuts jusqu'à aujourd'hui. Il permettra notamment au lecteur de se rendre compte que comme pour beaucoup de technologies, les drones étaient avant tout des dispositifs militaires et que ce n'est que depuis peu de temps que leur utilisation est passée dans le domaine civil.

L'histoire des drones commence en 1916, le premier projet étant de créer des appareils sans pilote pour jouer le rôle d'un avion cible pour entraîner les pilotes de l'armée britannique (Higham, 2006).

En 1917, les français furent les premiers à faire voler un avion sans intervention humaine sur une distance d'un kilomètre (Centre d'études stratégiques aérospatiales, 2006).

Depuis ce jour, la technologie n'a pas cessé d'évoluer et de se perfectionner devenant de plus en plus sophistiquée et meurtrière : pendant la seconde guerre mondiale, l'utilisation de drones comme les V1 et V2 (missiles allemands autoguidés) devint tristement célèbre (Jarry, 2010).

D'autres types de drones avec des usages plus diversifiés comme la reconnaissance et l'entraînement au tir virent le jour dans les années 1950 avec notamment le Ryan FireBee qui fut utilisé pendant la guerre du Vietnam (Newcome, 2004).

Plus tard, les américains principalement et beaucoup d'autres pays mirent au point une flotte d'UAV très diversifiée allant de l'hélicoptère de surveillance (Northrop Grumman, Black hornet nano,...) à l'avion de chasse (USAF Reaper, Global Hawk, Dassault Neurone, ...) en passant par le drone spatial (Boeing X-37B).

Ce n'est que dans les années 1990 que la banalisation des technologies de communication sans fils et l'avènement des capteurs à des prix abordables permirent le développement de drones civils par des entreprises privées.

### **1.3 Applications**

Comme nous l'avons vu plus haut, la majorité des applications pour un drone sont restées militaires pendant longtemps. Mais de nos jours, la démocratisation des drones offre une multitude de nouvelles applications dont la seule limite est l'imagination des utilisateurs.

L'exemple le plus connu est le fameux drone Amazone<sup>®</sup> capable de livrer ses colis partout en ville dans un délai record, ouvrant ainsi la voie à l'espace aérien urbain. Beaucoup de compagnies se sont donc lancées à la conquête des villes en proposant des services de surveillance aérienne (Google, FEDEX, UPS, Domino's, etc.).

Mais le domaine des drones ne s'arrête pas à la surveillance et la livraison, Austin (2010, p 34) et DeGarmo (2004, p. 1.14) proposent une multitude d'applications dans différents domaines :

- La sécurité intérieure incluant: surveillance des frontières, surveillance des sites sensibles, observation des trafics de drogue, surveillance du trafic routier, sécurité portuaire.
- Les applications civiles gouvernementales incluant : les interventions d'urgence, les applications de surveillance de la loi, le sauvetage, la surveillance des incendies de forêt de recherche, les relais de communication, la cartographie des inondations, l'imagerie en haute altitude, la détection et le suivi des émissions nucléaires, biologiques et chimiques, l'aide humanitaire, la cartographie et la surveillance des déversements chimiques de pétrole.

- Les applications scientifiques pour les relevés de données incluant : la recherche et la surveillance sur les risques naturels, la surveillance et la cartographie de l'environnement, la surveillance atmosphérique sur site, l'imagerie hyper spectrale, l'observation des mers et de l'écoulement glaciaire, l'imagerie de l'humidité du sol, etc.
- Les applications civiles privées incluant : la photographie, l'agriculture (surveillance et épandage), la surveillance de propriété, la livraison de biens, l'imagerie immobilière, la météorologie, etc.

Pour les militaires, on distingue principalement 11 types d'usages développés par Sarris (2001, p. 2):

- Reconnaissance visuelle, surveillance et identification de cibles;
- Surveillance avec un radar à synthèse d'ouverture, en temps de paix ou pour le combat;
- Opérations de déception;
- Opérations maritimes (support au feu, acquisition de cibles au-delà de l'horizon, classification de navire);
- Guerre électronique et espionnage;
- Opérations spéciales et psychologiques;
- Missions de météorologie;
- Soutien à l'itinéraire et à la reconnaissance pour l'atterrissage;
- Ajustement du feu indirect et appui aérien rapproché;
- Confirmation de dommages dus au combat;
- Relais radio.

Un dernier usage est lui aussi en plein essor ces dernières années surtout avec les quadripales, il s'agit de l'usage récréatif qui permet à n'importe qui d'utiliser un drone à partir d'une simple tablette, par exemple.

## **1.4 Marché actuel**

Du fait de sa démocratisation, le marché des drones est en plein boom économique (marché de 6.6 milliards en 2013 et estimé à 11.4 milliards pour 2022) (Eurocontrol, 2012), et il paraît important de citer les principaux acteurs et les organismes qui les régulent.

### **1.4.1 Principaux acteurs**

Les principaux acteurs peuvent être classifiés en deux principales catégories : les fabricants et les utilisateurs.

Au sein de ces deux catégories, nous pouvons également identifier deux sous-catégories : les militaires et les civils pour les fabricants et les organisations privées et publiques pour les utilisateurs. Il est aujourd'hui impossible de tenir des listes précises de tous les acteurs mais nous allons citer les principaux afin que le lecteur ait une idée de leur nombre et de leur importance.

Les principaux fabricants militaires sont pour la plupart des avionneurs bien connus : Boeing, Lockheed Martin, U.S. Air Force, Airbus, BAE Systems, Dassault Aviation, Northrop Grumman, General Atomics, etc.

Pour ce qui est des fabricants civils, les acteurs sont peu connus car principalement issus des milieux récréatifs. Un seul se démarque du lot : la NASA, les autres sont d'envergure bien moins importante : Parrot, DJI, Delta Drone, Walkera, Micro Drones, Delta Tech, etc.

Dans les utilisateurs publics des drones, nous retrouvons principalement des organismes d'État comme la NASA, CED ou encore Roscosmos.

Enfin pour terminer, les utilisateurs privés sont majoritairement des petites entreprises de services qui se spécialisent dans l'utilisation de drones pour les différents usages qui ont été cités plus haut (par exemple Locadrone ou encore Vuduciel). Également, on retrouve les particuliers pour leurs usages personnels.

### 1.4.2 Régulation

Comme nous venons de le voir, ce n'est que récemment que le nombre de drones a fortement augmenté, créant ainsi une demande de régulation pour gérer l'espace aérien.

La législation aérienne classique fut le fruit de plusieurs dizaines d'années de travail pour les autorités, aussi la mise en place d'une réglementation pour les drones ne se fait pas du jour au lendemain. Les différentes autorités à travers le monde peinent à produire les lois et à mettre en place des organismes qui réguleront les UAVs.

Au Canada, la gestion des drones est laissée à Transport Canada qui a défini très récemment des directives en novembre 2014 (Transport Canada, 2014), elles sont basées sur deux critères : l'utilisation du drone (récréatif ou non) et son poids. Si le drone est récréatif et inférieur à 35kg, il n'y a pas besoin d'autorisation de vol mais il faut suivre un code de conduite (pas de survols de foules, vols de jour uniquement, respect de la vie privée, etc.). Dans les autres cas, une série de formulaire est prévue en fonction du poids et du degré d'autonomie du drone. La réglementation est donnée en annexe I de ce document.

La FAA (Federal Aviation Administration) Américaine a quant à elle produit un document en mars dernier expliquant qu'une législation allait arriver mais prendrait du temps à être écrite (FAA, 2014).

En Europe, d'une manière générale, la charge de cette réglementation est laissée à chaque pays et l'état des lois est très variable d'un pays à un autre. Le seul élément commun entre tous ces pays est un poids limite à partir duquel une réglementation est systématique.

Au niveau international, l'OACI (Organisation de l'Aviation Civile Internationale) a produit un rapport définissant les types de drones et tous les termes s'y relatant mais n'a émis aucune législation, laissant la possibilité à chaque pays de créer ses propres lois (OACI, 2011).

Un seul élément est universel entre tous les pays, il s'agit de la gestion des UAVs militaires qui, eux, sont toujours soumis à la réglementation aérienne locale pour un contrôle draconien de leurs activités (généralement sous la direction du département de la défense).

### 1.4.3 Technologies utilisées

Maintenant que les bases théoriques ont été posées, nous allons nous intéresser à un aspect plus technologique des UAVs. Dans un premier temps, nous verrons les différents types de structures, puis nous nous intéresserons à leurs types de propulsions et d'alimentations à bord. Enfin pour finir, nous verrons quelles sont les technologies de communications utilisées actuellement pour commander les drones.

#### 1.4.3.1 Types de structure

Il n'existe pas de réelle classification de types de structures d'UAV car il y a trop de disparité d'un fabricant à un autre. Néanmoins, ce paragraphe est une ébauche pour permettre au lecteur de connaître les formes de drones les plus courantes.

- ***Les aéronefs à voilure fixes*** : il s'agit de drones ressemblant le plus à des avions, ils disposent d'une aile fixe qui les porte à partir d'une certaine vitesse. Ils sont autopropulsés par un ou plusieurs moteurs et requièrent généralement une piste pour le décollage et l'atterrissage.
- ***Les aéronefs à voilures tournantes*** : les voilures tournantes se rapprochent des hélicoptères et peuvent effectuer des atterrissages et décollages verticaux. Ils disposent d'hélices orientées vers le ciel qui produisent assez de poussée pour soulever la machine. Cependant, contrairement aux hélicoptères, les drones peuvent avoir jusqu'à 8 moteurs équipés chacun d'une hélice.
- ***Les aéronefs avec ballon rempli de gaz*** : il en existe peu et sont très peu manœuvrant mais ils disposent d'une grande autonomie due à leur maintien en l'air qui ne demande pas de propulsion.
- ***Les autres*** : il existe de nombreux autres drones dont le type est inclassable car trop spécifique, on prendra par exemple le X-37B qui est drone spatial ou encore le Colibri "nano" qui reproduit le battement d'ailes des oiseaux.

S'il n'existe pas de classification de types de structure d'UAV, il existe cependant des classifications en fonction de leurs utilités et en fonction de leur catégorie (Micro/Mini

drones, Drones tactiques, Drones stratégiques, Drones spécialisés), leur poids, leur altitude de croisière et leur endurance. Bento (2006, p 55) offre par exemple un tableau très complet classifiant les drones en 13 types.

### **1.4.3.2 Propulsion**

On distingue principalement deux types de propulsion: les propulsions thermiques (moteurs à énergies fossiles) et les propulsions électriques.

La majorité des drones sont équipés de moteurs électriques car plus faciles à entretenir et à faire fonctionner, ils sont également moins bruyants et plus écologiques. Malheureusement, ils nécessitent des batteries relativement lourdes, ce qui réduit considérablement l'autonomie du drone.

À partir d'un certain poids, l'énergie demandée pour élever un UAV dans les airs est considérable, aussi les plus gros drones sont tous équipés de moteurs thermiques divisés en deux classes : les moteurs à pistons (plus lents, moins puissants, moins chers et moins gourmands en carburant) et les moteurs à turbines (relativement bruyants, très consommateurs mais beaucoup plus puissants).

L'utilisation de ce type de moteur présente de nombreux avantages en termes d'autonomie et de vitesse de vol mais complique beaucoup les systèmes de régulation des drones. De plus, leur démarrage est très complexe au sol et souvent impossible en vol. Pour finir, bien que leur principale source d'énergie vienne du carburant, les drones équipés de moteurs thermiques (à quelques exceptions près) nécessitent aussi une batterie pour faire fonctionner les systèmes de commande et gestion de vol.

### **1.4.3.3 Communications**

L'utilisation d'un drone nécessite une liaison avec le sol pour le piloter et/ou lui envoyer des instructions. Aussi, il est obligatoire d'embarquer un système de communication à bord, mais il existe plusieurs types de technologies avec chacune leurs forces et leurs faiblesses.

- La technologie RF (Radio Fréquence) avec les télécommandes RC (Remote Control). Cette technologie est la moins chère, la plus vieille. Elle est entièrement analogique.

Les télécommandes RC utilisent généralement des fréquences de 2.4 GHz et émettent des PPMs (Pulse Position Modulation) qui permettent de communiquer plusieurs canaux sur une seule fréquence. La portée de cette technologie est d'environ 2 à 3 km en fonction de la puissance de l'émetteur et du récepteur et à condition qu'il n'y ait pas d'obstacle. Elle équipe la quasi-totalité des drones civils récréatifs.

- Le Wifi : encore très peu développé, le Wifi commence à équiper quelques drones, sa portée est d'environ 300m et fonctionne sur une bande de fréquence de 2.4GHz. La principale différence avec le RF est que le Wifi est une technologie numérique. Elle permet d'envoyer des ordres et de récupérer des signaux vidéo numériques. Cette technologie n'est pas très dispendieuse mais est assez énergivore et n'a pas une bonne portée.
- Les technologies militaires : ces technologies de communication longue distances sont classifiées, nous ne savons donc pas comment elles fonctionnent ni sur quelles bandes de fréquence elles transitent. Une analyse ne serait que spéculative. Elles sont probablement relayées par satellite et quelques antennes réseaux, mais cela ne reste que des suppositions car le sujet militaire est très peu documenté.

### **1.5 Quelles technologies et utilisations restent à explorer?**

Le présent chapitre a expliqué quel est l'état actuel du marché et quelles sont les applications actuellement attribuées aux drones. La question est maintenant de savoir comment innover dans ce domaine.

De ce que nous avons vu, les drones civils sont en pleine expansion mais leur exploitation est limitée par la portée de leur système de guidage et de communication, par leur robustesse et par la sécurité du système. De plus, leur intervention est surtout préventive et ne sert qu'à faire des observations ou des relevés.

L'idée est donc de trouver un moyen de piloter des drones se trouvant à de longues distances et d'offrir à l'utilisateur un service de pilotage semblable à celui d'un drone de chasse pour les militaires mais avec des technologies plus abordables, tout en étant robuste et sécuritaire.

La finalité étant de créer un UAV capable d'intervenir à la place d'un homme et plus rapidement tout en gardant l'utilisateur en sécurité. Ce drone pourrait prendre différentes formes selon les applications mais avec des éléments de base jugés essentiels pour un maximum d'applications.

En nous basant sur les descriptions technologiques citées ci-dessus et en nous référant à la littérature, il semble qu'un moyen de communication ne soit pas encore utilisé dans le domaine civil : le LTE. Ohleger, Xie et Gibson (2013, p 5104 à 5112) traitent de l'acheminement d'un signal vidéo monté sur un drone et envoyé via le LTE et la société Alcatel-Lucent a réussi à piloter un drone avec des protocoles LTE uniquement et sans se servir de réseaux. Mais aujourd'hui il n'existe aucun système capable de piloter un drone connecté en LTE via le réseau internet. À travers ce mémoire nous allons donc tenter d'exploiter ce moyen de communication aujourd'hui absent du domaine des drones.

Au cours de ce document, nous allons expliquer comment nous avons tenté d'innover en réalisant un drone versatile, capable d'être piloté à de grandes distances par LTE en offrant un poste de pilotage avec de réelles avioniques. Ce drone sera également équipé d'un bras télescopique dans le but de faire des interventions à distance à la place d'un homme pour faire du « On-Site Servicing » (Service sur Site) de manière robuste et sécuritaire.

## CHAPITRE 2

### LE PROJET AMOOS

Le projet AMOOS ("Autonomous Mission for On-Orbite Servicing" pour mission automatisée de maintenance en orbite des satellites) est le projet initiateur de cette maîtrise. Il est issu de la session 2014 de l'ISU (International Space University) à l'ÉTS pendant laquelle une équipe de 32 étudiants s'est penchée sur une étude de faisabilité visant à réparer un satellite en orbite grâce à un UAV.

Au cours de ce chapitre, nous verrons comment nous avons créé des simulations de trajectoires et de missions spatiales opérées par des UAVs orbitaux. Le but étant d'acquérir des connaissances sur les missions autonomes d'aéronefs non-habités. Ce chapitre fera également office d'introduction aux changements de repères dans l'espace qui nous seront utiles tout au long de ce mémoire.

#### 2.1 Description du projet

Le nombre de satellites en orbite autour de la terre est en constante augmentation, créant un nombre d'objets célestes de plus en plus important au-dessus de nos têtes. Malheureusement, parmi ces satellites artificiels, beaucoup ne sont plus actifs et deviennent des déchets spatiaux. Le problème est que l'accumulation de ces déchets détériore le fonctionnement des satellites en activité. De plus, beaucoup de scientifiques pensent que les orbites basses commencent à arriver à saturation et qu'une réaction en chaîne de débris entant en collision les uns avec les autres peut arriver à tout moment causant ainsi de multiples dommages en cascade (syndrome de Kessler).

La solution est de dégorger les orbites basses en réduisant le nombre de déchets, pour cela nous disposons de trois options :

- Réparer/recharger les satellites arrivés en fin de vie pour ne pas avoir à les renouvelés.

- Déployer des nouveaux satellites sans créer de déchets supplémentaires avec les lanceurs réutilisables.
- Retirer certains déchets spatiaux ou les propulser sur d'autres orbites afin qu'ils se consomment dans l'atmosphère ou terminent sur des orbites mortes.

Mais l'application de ces solutions n'est pas possible sans le développement et la mise en service de drones spatiaux capables d'aller en orbite, effectuer des opérations et revenir se poser sur terre pour ne créer aucun déchet supplémentaire.

Le Projet AMOOS a pour but d'étudier la conception et l'exploitation des missions autonomes de maintenance en orbite des satellites en utilisant des UAVs (Unmanned Aerial Vehicle). L'étude visait notamment à concevoir un scénario de simulation afin d'apporter des preuves sur les capacités d'un drone à exécuter des missions robotisées et autonomes sur orbite terrestre basse (LEO).

La faisabilité du projet devait être prouvée en effectuant des missions orbitales (simulation par ordinateur) synchronisées en temps réel avec l'exécution d'une mission à échelle réduite avec des drones quadripales, ce qui a ouvert les horizons d'opportunité avec le projet AMOSS qui sera présenté plus loin.

Le projet AMOOS devait également permettre d'effectuer l'identification de technologies spatiales capables de faire de la maintenance en orbite de satellites, déployer de petits satellites et capturer des débris spatiaux à l'abandon. Adapté des articles Substance (Landry, 2014) et International Astronautical Congress (Landry, Vigneron et Andrade, 2014).

L'intervention de l'ISU étant très courte (9 semaines), notre mission fut de préparer ces deux types de simulations afin qu'elles puissent répondre aux attentes techniques de l'équipe AMOOS pendant la réalisation de leur étude (ISU, 2014).

## **2.2 Simulations virtuelles**

Pour réaliser les simulations virtuelles, nous avons utilisé deux logiciels de simulation différents pour deux types de travaux différents.

- Le logiciel STK (Systems Tool Kit) fut utilisé pour simuler les trajectoires des différents UAVs et satellites simulés.
- Le logiciel Xplane fut utilisé pour réaliser des prises de vue plus esthétiques et faire des simulations de pilotage depuis une station au sol.

### 2.2.1 Trajectoire STK

Le logiciel STK développé par la société AGI® est une référence en ingénierie spatiale. Il permet d'effectuer une multitude de simulations de trajectoires prenant en compte tous les paramètres gravitationnels aussi bien terrestres que spatiaux. Il autorise un grand nombre de types de coordonnées et de repères, facilitant ainsi les calculs d'approche et de mise en orbite. Il dispose également d'une base de données contenant tous les lancements spatiaux et les orbites actuelles des satellites, il fut donc l'outil idéal pour les simulations AMOOS. Un exemple d'utilisation de STK est donné en figure 2.1.

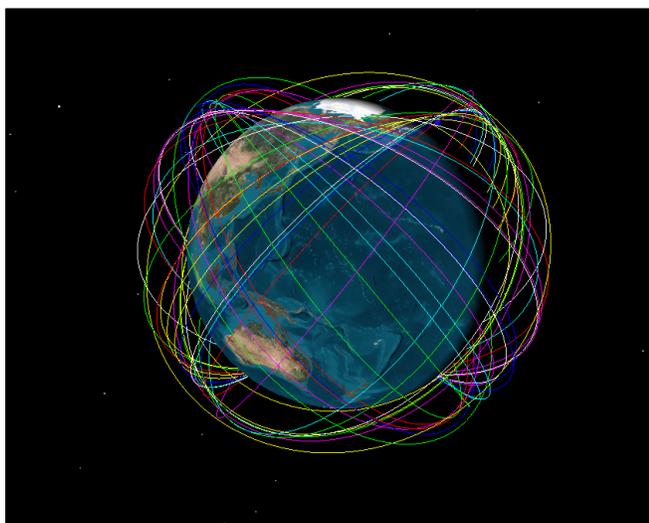


Figure 2.1 Exemple de simulation STK avec la constellation des satellites Globalstar

Cependant, il nous posa deux types de problèmes :

- Les trajectoires des satellites chargées avaient beaucoup trop de paramètres de déviation pour considérer leurs orbites comme stables (frottement et erreur d'orbite,

par exemple). Même STK était incapable de nous donner les paramètres classiques des orbites. Il nous était donc impossible de faire des simulations de rencontres spatiales simples avec une quantité aussi importante de paramètres.

- STK étant principalement utilisé pour des simulations orbitales de longue durée, l'échantillonnage des positions des satellites était trop petit pour nous permettre d'avoir un bon rendu visuel (les courbes étaient trop fractionnées).

Pour ces deux raisons, nous avons pris la décision de simuler nous même les trajectoires points par points sur Matlab afin de maîtriser parfaitement tous les paramètres et l'échantillonnage des positions. Mais pour cela, il nous a fallu étudier les calculs mathématiques des trajectoires des satellites.

### **2.2.1.1 Introduction aux changements de repères et aux trajectoires spatiales**

Afin de positionner un mobile dans l'espace, il est nécessaire de poser des bases de calcul dans un repère galiléen (considéré comme immobile) et de changer plusieurs fois de repères. Nous allons donc commencer par identifier les différents repères dont nous aurons besoin. Dans notre cas, la terre fera office de repère galiléen et sera notre repère immobile pour tous les calculs. Remarquons tout de même qu'en temps normal, la terre n'est pas galiléenne car en mouvement dans l'espace. Mais compte tenu de la précision de nos capteurs et de nos calculs, elle fera un référentiel suffisant. Les repères présentés sont issus des notes de cours de Landry (Chap. 1, 2014).

- La figure 2.2 présente le repère ECI (« Earth Central Inertial »), un repère inertiel terrestre orthonormé fixe au sein de l'univers. Il n'accélère et ne tourne pas, ce qui lui permet d'être utilisé pour les calculs de trajectoire de satellites. Le repère ECI est cartésien (axes XYZ) et est centré autour du centre de masse de la terre. Ses axes sont alignés sur l'axe de rotation de la terre (axe Z) et vers l'équinoxe de printemps sur le plan équatorial (axe X). Étant fixe dans l'univers, le repère ECI prend en compte la rotation de la terre, ce qui n'est pas le cas pour notre prochain repère.

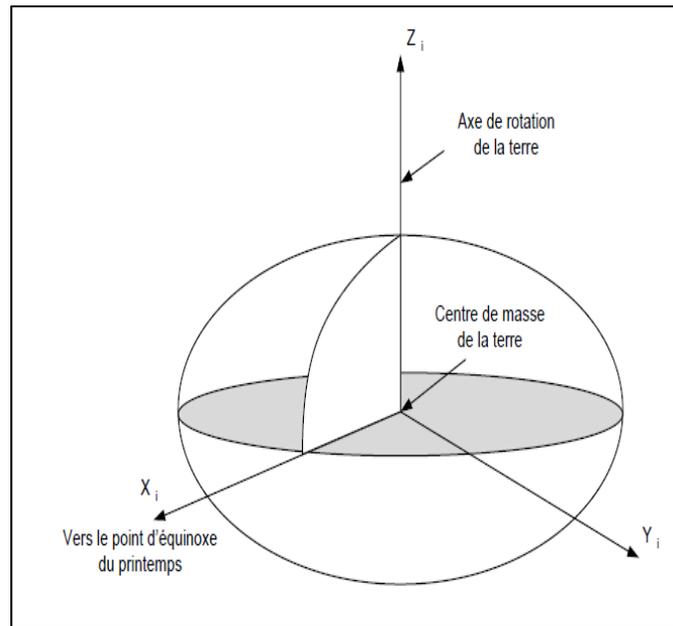


Figure 2.2 Repère ECI  
 (« Earth Central Inertial »),  
 tirée de Landry 2014

- La figure 2.3 présente le repère ECEF (« Earth Centered Earth Fixed »), un repère cartésien orthonormé terrestre mais non fixé dans l'univers. Il est centré sur le centre de masse de la terre et ses axes sont orientés sur l'axe de rotation de la terre (axe Z) et vers le méridien de Greenwich selon le plan équatorial (axe X). Il est le repère à l'origine des coordonnées terrestres latitude et longitude.

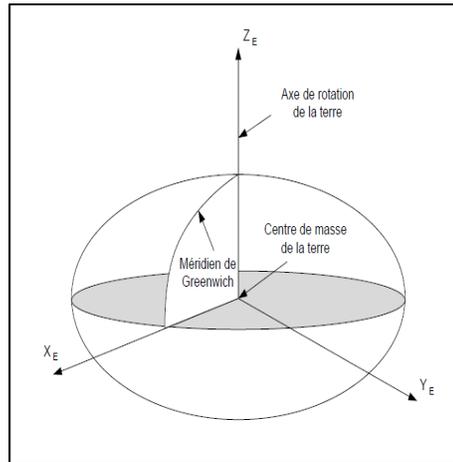


Figure 2.3 Repère ECEF  
 (« Earth Centered Earth Fixed »),  
 tirée de Landry 2014

- La figure 2.4 présente le repère ENU (« East, North, Up »), un repère cartésien orthonormé local, c'est-à-dire qu'il évolue avec le mobile, sa position n'est pas fixe dans un repère galiléen. Le repère ENU est tangent à l'ellipsoïde de la terre et pointe le Nord géographique avec l'axe Y et l'Est avec l'axe X. Selon les conventions, l'axe Z pointe donc à l'opposé du centre de masse de la terre et est normal à la terre. Le repère ENU est principalement utilisé pour localiser un mobile à la surface de la terre.

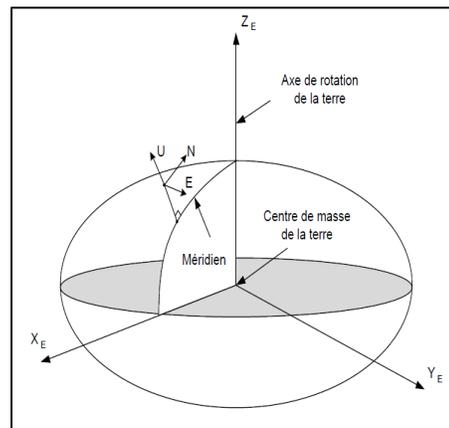


Figure 2.4 Repère ENU  
 (« East, North, Up »),  
 tirée de Landry 2014

- La figure 2.5 présente le repère NED (« North, East, Down »), un repère cartésien orthonormé local. Il est très semblable au repère ENU et partage son centre. Il diffère simplement par l'orientation de ses axes, l'axe X pointe le Nord géographique et l'axe Y pointe l'Est. L'axe Z pointe donc vers le centre de masse de la terre, ce qui permet d'avoir une valeur positive pour l'altitude du mobile par rapport à l'ellipsoïde de la terre. Ce repère est très utilisé en navigation.

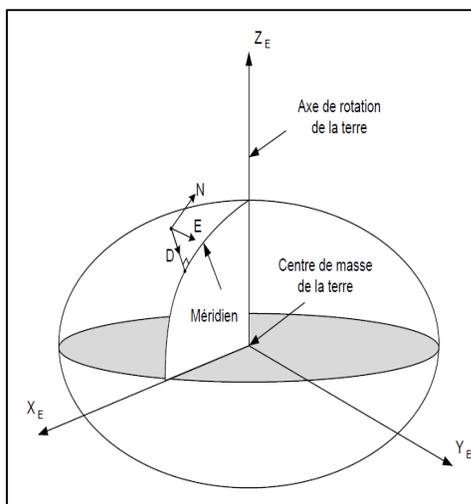


Figure 2.5 Repère NED  
 (« North, East, Down »),  
 tirée de Landry 2014

- La figure 2.6 présente le repère du mobile, un repère cartésien orthonormé local. Il est attaché au mobile que l'on suit, l'axe X pointe vers le devant du mobile et l'axe Z pointe vers le bas. Le centre du repère étant généralement centré sur le centre de gravité du mobile. Ce repère est généralement utilisé pour recalculer les données des capteurs s'ils ne sont pas positionnés au centre du mobile. Le repère du mobile ne forme pas d'angle particulier avec le repère ECEF comme les deux repères précédents, il peut être orienté indépendamment selon les angles d'Euler du mobile.

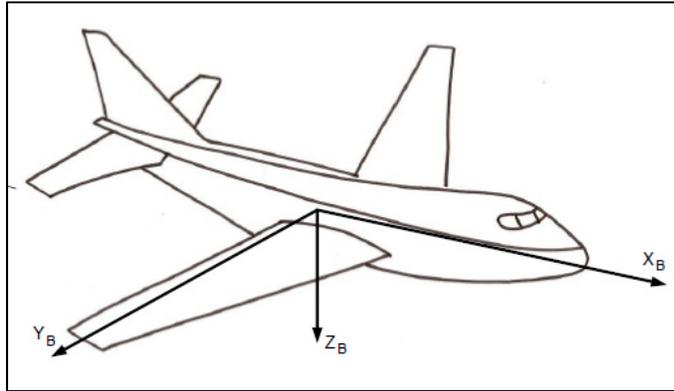


Figure 2.6 Repère du mobile,  
tirée de Landry 2014

Maintenant que nous disposons de repères suffisants, nous allons nous intéresser aux trajectoires satellitaires en elles-mêmes. Ces trajectoires sont très complexes car elles dépendent de beaucoup de paramètres et sont soumises à différentes lois de la physique.

Tout d'abord, comme chaque être de cette planète, les satellites sont soumis à la gravité qui les entraîne vers la terre. Le principe pour maintenir un satellite en orbite est simple, ce dernier est accéléré à une vitesse suffisante perpendiculairement à la terre pour équilibrer la force gravitationnelle qui l'attire vers le centre de la terre (seconde loi de Newton). Mais la vitesse nécessaire à maintenir un satellite en orbite dépend également de son altitude (pour des raisons de forces d'attraction gravitationnelle) ce qui complique la tâche.

Également, les orbites des satellites sont souvent elliptiques, et la vitesse de rotation à l'instant  $t$  dépend donc de la loi des aires (seconde loi de Kepler). Autrement dit, la vitesse du satellite dépend de sa position par rapport au foyer de l'ellipse (Capderou, 2003.)

Mais ce mémoire ne traitant pas de physique fondamentale, nous nous contenterons d'appliquer les formules simplifiées tirées des notes de cours de Landry (Chap. 4, 2014) pour déterminer la position d'un satellite dans un repère ECEF.

La figure 2.7 ci-dessous représente les différents éléments d'une orbite elliptique plate d'un satellite.

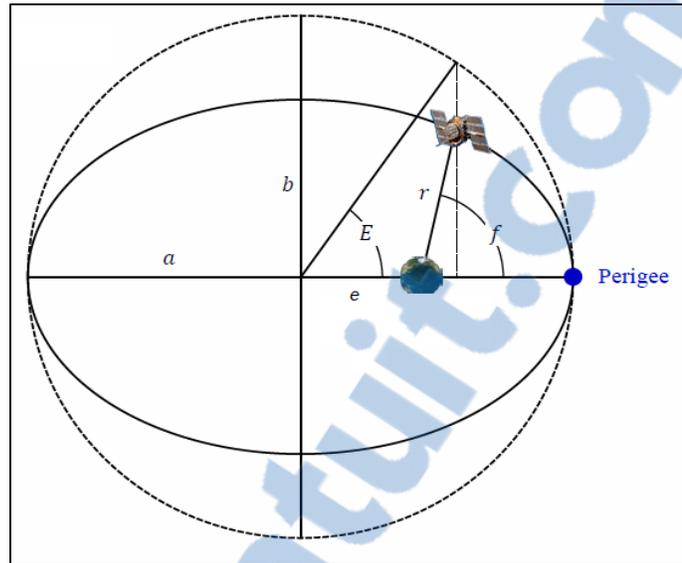


Figure 2.7 Paramètres d'une orbite elliptique plate,  
tirée de Landry 2014

Où l'on distingue :

- $a$  : le demi grand axe;
- $b$  : le demi petit axe;
- $e$  : l'excentricité de l'ellipsoïde;
- $E$  : l'anomalie excentrique;
- $f$  : l'anomalie vraie;
- $r$  : la distance radiale;
- $\mu$  : l'anomalie moyenne.

La première étape consiste à déterminer l'anomalie vraie  $f$  de l'orbite elliptique.

Pour cela, considérons  $t_o = 0s$ , le temps auquel se trouve notre satellite au périgée de son orbite. Nous souhaitons déterminer la position de notre satellite à l'instant  $t < t_o$ .

L'anomalie moyenne est donnée par :

$$\mu = \sqrt{\frac{GM}{a^3}} \quad (2.1)$$

Où  $GM$  représente le produit de la constante gravitationnelle par la masse de la terre.

Ensuite, il nous faut déterminer l'anomalie excentrique qui est donnée par l'application de la formule suivante jusqu'à convergence :

$$E = \mu + e * \sin(E) \quad (2.2)$$

Pour enfin déterminer l'anomalie vraie  $f$  suivante :

$$f = \arctan2(\sqrt{1 - e^2} * \sin(E), \cos(E) - e) \quad (2.3)$$

Et la distance radiale :

$$r = a * (1 - e * \cos(E)) \quad (2.4)$$

Les précédents calculs permettent de connaître l'anomalie vraie et la distance radiale sur une orbite elliptique plane, mais il existe une infinité d'ellipses possible autour d'une planète. Les différents paramètres pour orienter une ellipse sont visibles sur la figure 2.8.

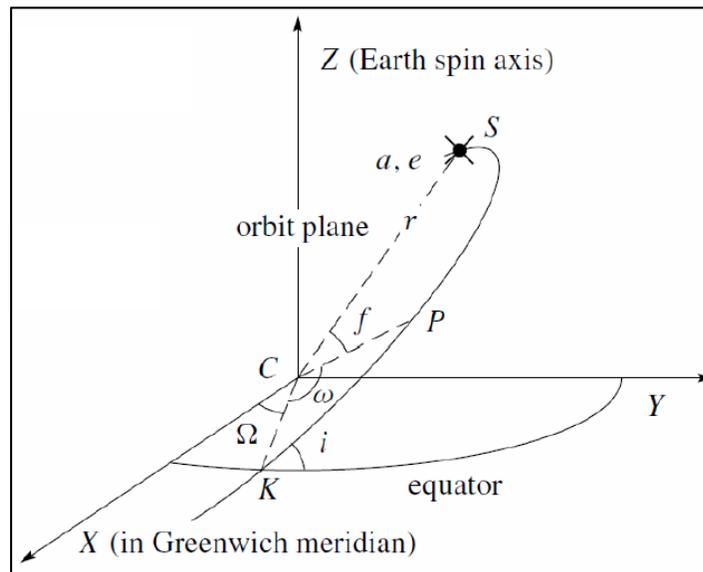


Figure 2.8 Paramètres d'une orbite autour de la terre, tirée de Landry 2014

Où l'on distingue :

- $S$  : la position du satellite;
- $P$  : le périégée de l'ellipse;
- $C$  : le centre de masse de la terre;
- $\omega$  : l'argument du périégée;
- $i$  : l'inclinaison;
- $\Omega$  : la longitude du nœud ascendant.

La seconde étape des calculs consiste donc à déterminer la position du satellite sur l'orbite elliptique orientée par rapport à la terre dans son repère ECEF.

La terre tournant sur elle-même, il faut commencer par appliquer un correctif à la longitude du nœud ascendant qui devient :

$$\Omega_t = \Omega - \omega_e * t \quad (2.5)$$

Avec  $\omega_e = 7.292115147 \cdot 10^{-5}$  rad/s : la vitesse de rotation de la terre sur elle-même.

Ensuite, la méthode la plus simple consiste à résoudre la matrice suivante pour déterminer les coordonnées cartésiennes du satellite à l'instant  $t$  :

$$X_{Sat} = \begin{bmatrix} X_{Sat} \\ Y_{Sat} \\ Z_{Sat} \end{bmatrix} = \begin{bmatrix} r * \cos(\omega + f) * \cos(\Omega) - r * \sin(\omega + f) * \cos(I) * \sin(\Omega) \\ r * \cos(\omega + f) * \sin(\Omega) + r * \sin(\omega + f) * \cos(I) * \cos(\Omega) \\ r * \cos(\omega + f) * \sin(I) \end{bmatrix} \quad (2.6)$$

Bien qu'elle puisse paraître déjà quelque peu complexe pour trois simples coordonnées, la méthode ci-dessus a été bien simplifiée par rapport à la réalité. En effet, l'argument du périégée, la distance radiale et l'inclinaison sont normalement soumis chacun à 2 coefficients de correction qui compliquent grandement les calculs. Mais dans le cas de nos simulations, l'objectif n'était pas de déterminer des trajectoires parfaites, nous avons donc abandonné ces coefficients pour plus de simplicité.

### 2.2.1.2 Lien Matlab vers STK

Afin de tracer nos trajectoires pour les rencontres spatiales, nous avons commencé par calculer point par point une orbite satellitaire simple comme expliqué ci-dessus. Avec comme prévu un échantillonnage de point plus précis que ce celui de STK pour un meilleur rendu visuel. Les calculs furent effectués sous le logiciel Matlab grâce auquel le calcul des quelques 5500 points d'une orbite fut instantané, la figure 2.9 présente un aperçu d'une orbite.

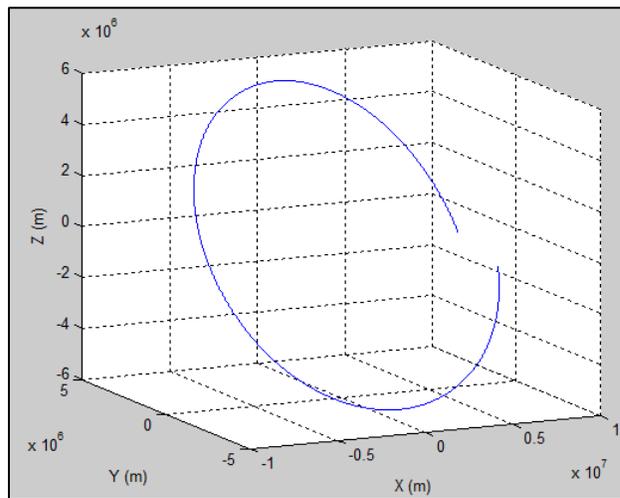


Figure 2.9 Calcul d'une orbite sous Matlab

Le problème étant maintenant de placer ces points sur STK pour procéder aux simulations virtuelles. STK propose de multiples fonctions pour réaliser des trajectoires mais aucune n'est possible grâce à un tableau de points comme celui que nous venions de générer. Le seul moyen nous correspondant le mieux était de placer les 5500 points un à un. Il nous fallut donc trouver une autre solution.

Pour pallier à ce problème, nous nous sommes penchés sur la structure des fichiers système du logiciel STK. Les fichiers représentant les trajectoires des différents mobiles étaient des fichiers munis d'une extension « .ac », une extension propre à STK. Nous avons décidé d'ouvrir ces fichiers en transformant leur extension en « .txt » à la place de « .ac » pour les lire comme un simple texte.

Après observation de l'architecture interne des fichiers, nous avons trouvé où étaient stockées les coordonnées des points avec lesquels STK affichait ses trajectoires. Mais la disposition et le format des coordonnées des points dans les fichiers STK sont très particuliers et nécessitent une modification avec une précision drastique pour pouvoir à nouveau être lus par le logiciel.

Or, il est impossible sous Matlab de faire ce type de traitement sur ce type de fichier. Nous nous sommes alors tournés vers Excel et la fonction VBA (Visual Basic for Applications) qui permet ce genre de modification. Nous avons donc conçu une application de plus de 250 lignes capable de lire des tableaux de coordonnées et de les réimplanter dans des fichiers système de STK. La figure 2.10 présente l'orbite de la figure 2.9 exportée sous STK.

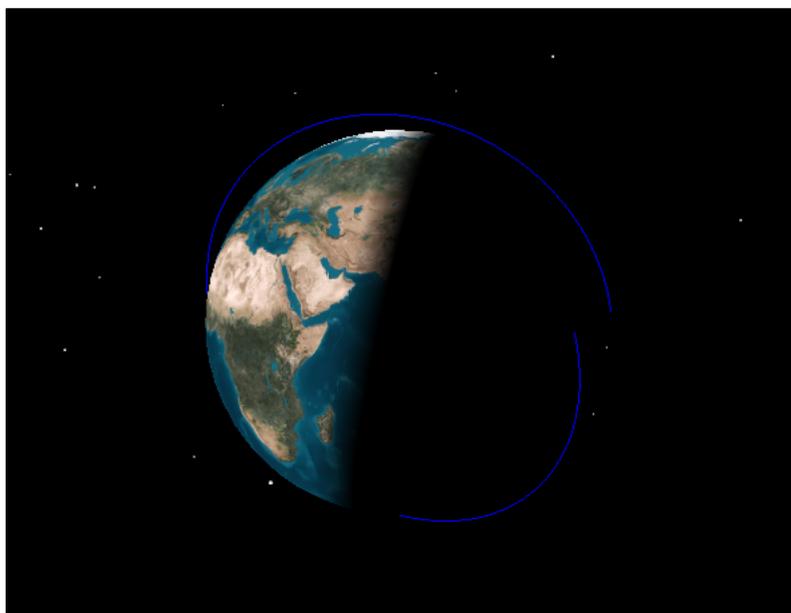


Figure 2.10 Orbite importée de Matlab à STK

À ce moment, nous avons été capables de tracer toutes les trajectoires satellitaires désirées sous Matlab et les réimplanter sous STK pour avoir tous les rendus visuels avec la précision désirée.

### 2.2.1.3 Calculs des trajectoires d'approche

La prochaine étape fut de tracer une orbite similaire à celle du satellite pour l'UAV venant à sa rencontre. La principale difficulté étant de pouvoir faire des trajectoires d'approche de toutes sortes autour du satellite (approche par l'arrière du satellite, approche en spirale, approche par bonds, etc.).

Pour ce faire, nous avons décidé de placer directement l'UAV sur la trajectoire du satellite calculée plus haut et d'utiliser un repère local pour pouvoir déplacer le drone par rapport au satellite et non pas directement par rapport à la terre (d'un point de vu extérieur, le drone bouge uniquement par rapport au satellite). Mais si bouger l'UAV dans le repère du satellite fut simple, il nous fallut ensuite recalculer les coordonnées dans le repère ECEF (repère du satellite) pour pouvoir les implanter dans STK. Nous avons donc dû procéder à des changements de repères pour passer du repère mobile du satellite au repère ECEF. La figure 2.11 illustre les différents changements de repère à effectuer.

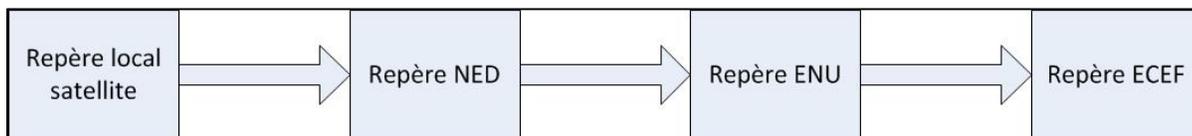


Figure 2.11 Changement de repère du satellite à l'ECEF

Pour passer d'un repère ENU à un repère ECEF, il est nécessaire de passer par des coordonnées polaires autrement dit de passer par les latitudes et longitudes de la terre.

Les latitudes sont des lignes imaginaires traversant la terre d'un pôle géographique à l'autre en ligne droite. Elles découpent la terre en 360 cartiers égaux appelés méridiens dont le premier traverse la ville de Greenwich. Les latitudes sont généralement notées  $\lambda$ .

Les longitudes sont elles aussi des lignes imaginaires mais parallèles à l'équateur. Elles découpent la terre en 180 tranches appelées parallèles. Les longitudes sont généralement notées  $\phi$ . La figure 2.12 donne un aperçu des latitudes et longitudes terrestres.

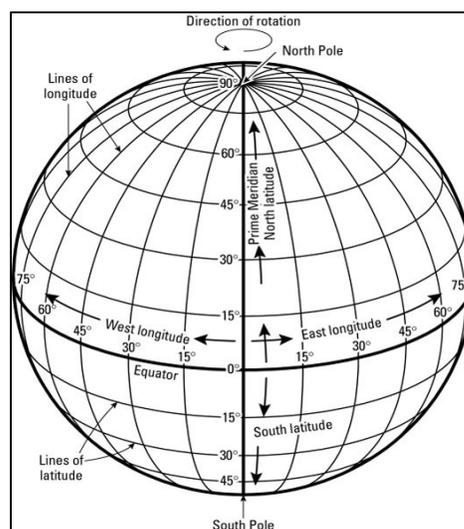


Figure 2.12 Latitudes et longitudes,  
tirée de Landry 2014

Le problème est que la terre n'est pas vraiment sphérique mais plutôt ellipsoïdale comme l'explique la National Imagery and Mapping Agency (NIMA, 2000) et le montre la figure 2.13. Les parallèles ne sont donc pas réellement proportionnels les uns par rapport aux autres et les calculs se compliquent un peu pour retrouver la latitude  $\lambda$  et la longitude  $\phi$  d'une paire de coordonnées cartésiennes  $(x,y,z)$ . Le principal inconvénient étant qu'à la surface d'un ellipsoïde, la normale ne passe pas forcément par le centre contrairement à une sphère.

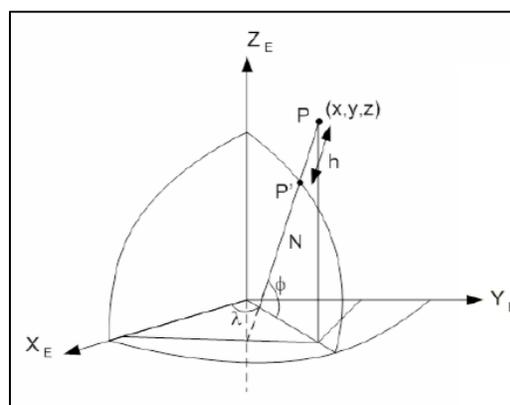


Figure 2.13 Placement d'une latitude et  
d'une longitude sur un ellipsoïde, tirée de  
Landry 2014

Pour calculer les coordonnées géodésiques d'un ellipsoïde à partir de coordonnées cartésiennes, le processus est itératif, c'est-à-dire que les calculs doivent être refait plusieurs fois jusqu'à ce qu'ils convergent vers une valeur fixe. La finalité étant de passer de coordonnées  $[x,y,z]$  cartésiennes à des coordonnées  $[\lambda,\phi,h]$  géodésiques (Landry, 2014).

La première étape consiste à initialiser les valeurs, on pose :

$$h = 0 \quad N = a \quad p = \sqrt{x^2 + y^2} \quad \lambda = \tan^{-1}\left(\frac{x}{y}\right)$$

Ensuite, on effectue les calculs 2.7 à 2.10 jusqu'à ce que le résultat reste inchangé.

$$\sin(\phi) = \frac{z}{(1 - e) * N + h} \quad (2.7)$$

$$\phi = \tan^{-1}\left(\frac{z + e^2 * N * \sin(\phi)}{p}\right) \quad (2.8)$$

$$N = \frac{a}{\sqrt{1 - e^2 * \sin^2(\phi)}} \quad (2.9)$$

$$h = \frac{p}{\cos(\phi)} - N \quad (2.10)$$

À la fin de ces itérations, on obtient les coordonnées géodésiques  $[\lambda,\phi,h]$  du mobile par rapport à un ellipsoïde de paramètre de demi grand axe  $a$  et d'excentricité  $e$ .

Pour définir l'ellipsoïde de la terre, il existe plusieurs modèles, le plus connu et celui que nous utiliserons étant le WGS84 (NIMA, 2000).

Il définit :

- $a = 6378137$  m
- $\frac{1}{f} = 298.257223563$

$\frac{1}{f}$  étant le coefficient d'aplatissement permettant de calculer l'excentricité  $e$  grâce à la formule suivante :

$$e = 2 * f - f^2 \quad (2.11)$$

Une fois les coordonnées géodésiques établies, il a fallu déterminer la matrices de changement de repère afin pouvoir passer du repère ENU au repère ECEF en fonction de  $\lambda$  et  $\phi$  (Landry, 2014). La matrice de rotation est notée conventionnellement  $C_N^E$  pour signifier qu'elle permet de passer un vecteur d'un repère N (ECEF) à un repère E (ENU) (DeSantis , 2011).

$$C_N^E = \begin{bmatrix} -\sin(\lambda) & -\sin(\phi) * \cos(\lambda) & \cos(\phi) * \cos(\lambda) \\ \cos(\lambda) & -\sin(\phi) * \cos(\lambda) & \cos(\phi) * \sin(\lambda) \\ 0 & \cos(\phi) & \sin(\phi) \end{bmatrix} \quad (2.12)$$

Le changement de repère s'effectuant par le calcul suivant :

$$\begin{bmatrix} x \\ y \\ z \end{bmatrix}_{ECEF} = C_N^E * \begin{bmatrix} x \\ y \\ z \end{bmatrix}_{ENU} \quad (2.13)$$

Pour ensuite passer du repère NED au repère ENU par la formule :

$$\begin{bmatrix} x \\ y \\ z \end{bmatrix}_{ENU} = C_L^N * \begin{bmatrix} x \\ y \\ z \end{bmatrix}_{NED} = \begin{bmatrix} 0 & 1 & 0 \\ 1 & 0 & 0 \\ 0 & 0 & -1 \end{bmatrix} * \begin{bmatrix} x \\ y \\ z \end{bmatrix}_{NED} \quad (2.14)$$

La dernière étape consistant à passer du repère mobile du satellite vers le repère local NED. Ce changement de repère est effectué grâce à une matrice DCM (Direct Cosinus Matrice) (Craig, 2005) qui permet de réorienter le repère mobile vers le repère NED via les angles d'Euler (Roulis  $\phi$ , Tangage  $\theta$ , Lacet  $\psi$ ) du satellite. La définition des angles d'Euler ne sera

pas développée dans ce document mais le lecteur peut toujours se référer à l'ouvrage de Stevens (Stevens ,1992) pour de plus amples explications.

$$\begin{bmatrix} x \\ y \\ z \end{bmatrix}_{NED} = C_B^L * \begin{bmatrix} x \\ y \\ z \end{bmatrix}_{Mobile} \quad (2.15)$$

$$= \begin{bmatrix} \cos(\theta) \cos(\psi) & -\cos(\phi) \sin(\psi) + \sin(\phi) \sin(\theta) \cos(\psi) & \sin(\theta) \sin(\psi) + \cos(\phi) \sin(\theta) \cos(\psi) \\ \cos(\theta) \sin(\psi) & \cos(\phi) \cos(\psi) + \sin(\phi) \sin(\theta) \sin(\psi) & -\sin(\phi) \cos(\psi) + \cos(\phi) \sin(\theta) \sin(\psi) \\ -\sin(\theta) & \sin(\phi) \cos(\theta) & \cos(\phi) \cos(\theta) \end{bmatrix} * \begin{bmatrix} x \\ y \\ z \end{bmatrix}_{Mobile}$$

Finalement, si on assemble les calculs 2.13, 2.14 et 2.15, la matrice de passage du repère mobile du satellite au repère ECEF se fait selon la matrice  $C_B^E$  suivante :

$$C_B^E = C_N^E * C_L^N * C_B^L \quad (2.16)$$

Et donc la transposition des coordonnées de l'UAV dans le repère mobile du satellite vers les coordonnées ECEF est finalement effectuées via la formule :

$$\begin{bmatrix} x_{UAV} \\ y_{UAV} \\ z_{UAV} \end{bmatrix}_{ECEF} = \begin{bmatrix} x_{Sat} \\ y_{Sat} \\ z_{Sat} \end{bmatrix}_{ECEF} + C_N^E * C_L^N * C_B^L * \begin{bmatrix} x_{UAV} \\ y_{UAV} \\ z_{UAV} \end{bmatrix}_{Sat} \quad (2.17)$$

Les précédents calculs nous ont permis d'obtenir les différentes trajectoires d'approche souhaitées sous Matlab et ensuite de les exporter sous STK. Les figures 2.14 ci-dessous montrent en rouge une fraction d'une orbite satellitaire et en bleu des approches possibles par un UAV.

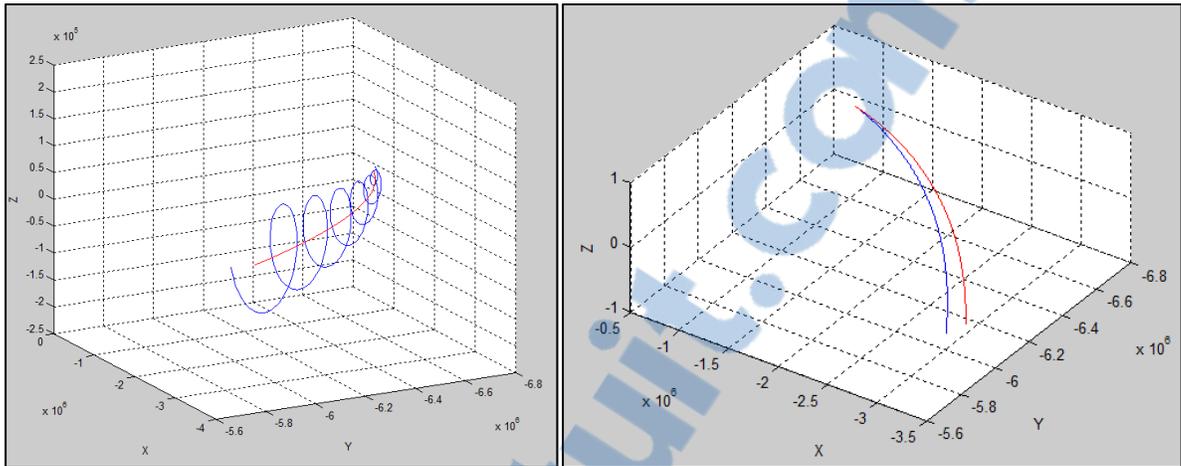


Figure 2.14 Exemple de trajectoire d'approche en spirale et directe sous Matlab

Au final, une trajectoire par bonds successifs fut calculée afin de coller au mieux aux trajectoires spatiales utilisées de nos jours. Cette trajectoire prend environ 8 h soit environ 5 orbites complètes comme le montre la figure 2.15. Cette durée peut sembler grande pour un néophyte mais en réalité, elle n'est que peu réaliste puisqu'actuellement les astronautes mettent environ 2 jours entre le décollage depuis la terre et l'arrimage à la station spatiale internationale (Sivolella, 2014).

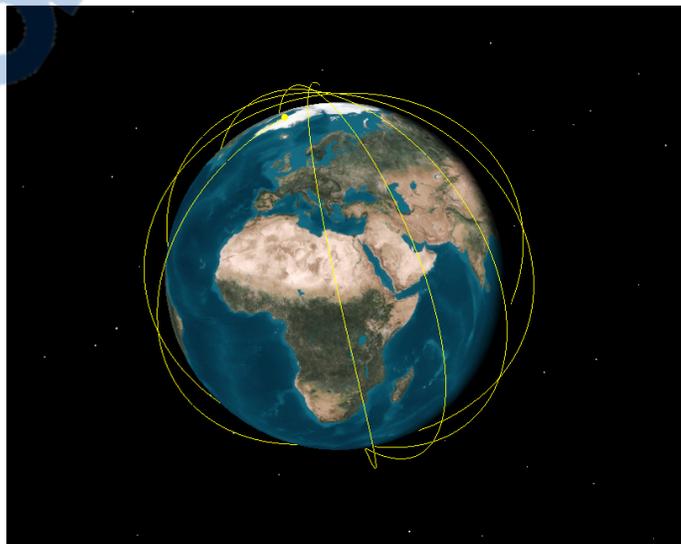


Figure 2.15 Trajectoire d'approche finale

### 2.2.1.4 Modèles UAV 3D

Afin de pouvoir visualiser l'approche sous STK, nous avons besoin d'un modèle 3D d'UAV avec un bras télescopique pour réaliser l'arrimage. L'équipe de l'ISU réalisa un modèle réaliste (comparable aux projets spatiaux existants) en 3D sur le logiciel SolidWorks selon ses besoins. Ce modèle nous fut ensuite transmis pour que nous puissions réaliser les simulations visuelles avec un rendu visuel réaliste.

Malheureusement, STK possède ses propres types de fichiers 3D avec une extension « .dae » très peu compatible. Il nous fallut donc trouver un moyen de passer des fichiers « .SLDASM » de SolidWorks vers une extension « .dae ».

Pour cela, nous avons fini par trouver une combinaison de 3 logiciels permettant de transformer les fichiers.

La première étape consista à transformer les fichiers « .SLDASM » existant en fichiers « .SAT » via SolidWorks pour pouvoir les exporter sur le logiciel DraftSight beaucoup plus primaire (figures 2.16).

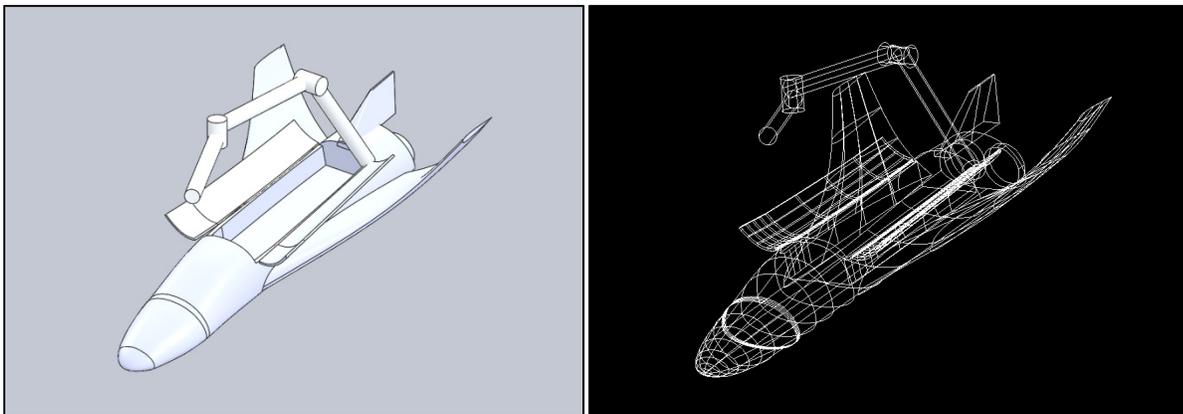


Figure 2.16 UAV en 3D sous SolidWorks et DraftSight

Une fois le modèle sous DraftSight, nous avons dû le redimensionner une première fois afin de contrer les effets d'une importation et convertir à nouveau en « .dwg ».

Le fichier fut ensuite importé dans le logiciel SketchUp pour y être réorienté sur les 3 bons axes et converti finalement en « .dae » compatible avec STK (figure 2.17).

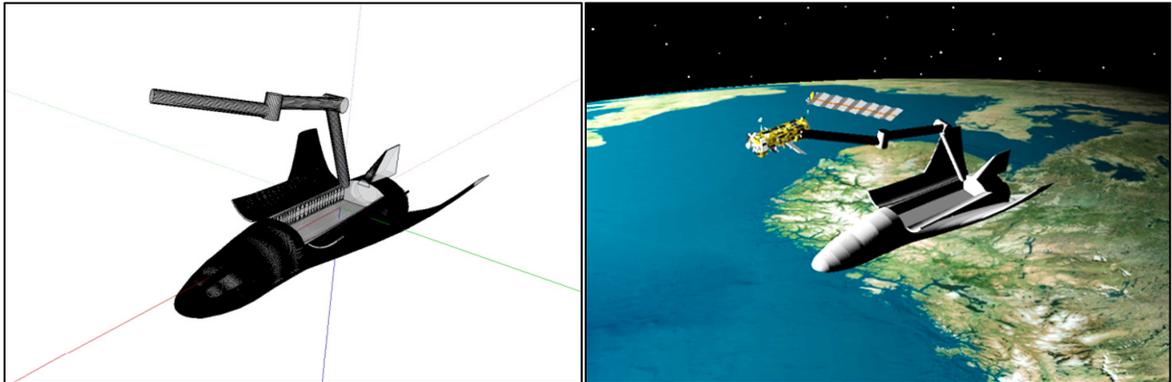


Figure 2.17 UAV en 3D sous SketchUp et STK

La chaîne finale de conversion est représentée sur la figure 2.18 ci-dessous qui récapitule les nombreuses étapes nécessaires pour générer un simple fichier 3D compatible avec STK.

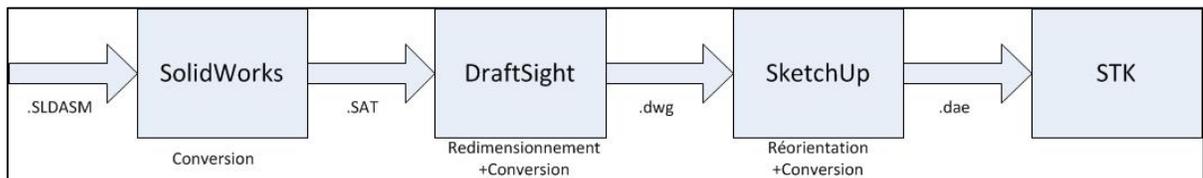


Figure 2.18 Chaîne de conversion de ".SLDASM" à ".dae"

### 2.2.2 Utilisation d'Xplane

Comme expliqué plus haut, une partie des simulations pour le projet AMOOS furent réalisées sur le logiciel Xplane (simulateur de vol) afin d'avoir de beaux rendus visuels et faire des simulations de pilotage d'UAV.

Pour cela, nous avons utilisé le simulateur de vol Hélicrew du département de Génie Électrique. Mais ce matériel était complètement neuf, nous avons dû le paramétrer pour qu'il réponde à nos besoins. Le simulateur est composé de 10 écrans dont 4 tactiles pour le tableau

de bord et comprend 3 ordinateurs pour fonctionner (un pour les instruments, un pour la vue de l'horizon et un pour l'instructeur).

Nous avons commencé par configurer Xplane en réseau sur les 3 ordinateurs afin de les synchroniser et de diviser les affichages. Le détail des connections est en annexe II.

Nous avons ensuite modifié un modèle d'avion ASSP-1B existant pour répondre à nos attentes. Trois types de modifications furent exécutés via le logiciel PlaneMaker :

- Une modification des paramètres moteurs et des propulseurs latéraux pour ne plus prendre en compte la consommation du carburant et augmenter la puissance.
- Une modification partielle du fuselage (retirer le poste de pilotage) et de la peinture (ajout des logos de l'ÉTS et de l'ISU) comme le montre la figure 2.19.



Figure 2.19 ASSP-1B modifié

- Une modification complète du tableau de bord (figure 2.20) pour l'adapter aux écrans de l'Hélicrew.



Figure 2.20 Tableau de bord ASSP-1B avant et après modifications

Une fois ces modifications effectuées, le simulateur de la figure 2.21 fut utilisé pour simuler des décollages, mises en orbite, rentrées atmosphériques et atterrissages d'un ASSP-1B.



Figure 2.21 HeliCrew simulant un ASSP-1B

L'HeliCrew ainsi que les simulations STK furent utilisées pendant toute la durée de l'ISU afin de répondre aux attentes de l'équipe AMOOS et prouver la faisabilité d'une mission spatiale de réparation en orbite.

Il est clair que le modèle d'UAV proposé lors de ces simulations est aujourd'hui irréalisable technologiquement mais le but de cet étude est uniquement basée sur la planification et les calculs de trajectoires. Les aspects liés à la propulsion et à l'autonomie énergétique de l'UAV n'ont donc pas été pris en compte et nécessitent encore beaucoup d'années de recherches. Les modules expérimentaux les plus proches de notre UAV sont actuellement le Boeing X37b, le Dream Chaser Sierra Nevada de la NASA et le Skylon de Reaction Engines d'après l'ISU (2014, p 29).

### **2.3 Simulations terrestres : Quadricoptères**

Comme indiqué dans l'introduction de ce chapitre, une partie des simulations devait être reproduite à échelle réduite avec des drones quadripales. Cependant à cause d'un manque de temps et une énorme différence d'autonomie entre les drones et une mission spatiale, l'équipe AMOOS prit la décision d'abandonner les simulations avec les quadricoptères. Le développement des drones fut donc avorté pour se concentrer sur les simulations virtuelles développées plus haut.

### **2.4 Retombées dans le domaine civil : AMOSS**

Depuis la conquête de l'espace commencée pendant la guerre froide entre les États-Unis et la Russie, les technologies spatiales n'ont pas cessées d'être innovantes. Elles sont à l'origine de nombreux objets du quotidien comme le GPS, les matériaux composites, ou encore la purification d'eau. Si bien que la recherche spatiale est la plus rentable au niveau des retombées économiques indirectes (1 dollar investi = 9 dollars de rentabilité) (ISU, 2014).

Basée sur ce principe, il semble certain qu'un projet spatial comme AMOOS avec ses recherches dans le domaine des UAVs autonomes est susceptible de créer des retombées dans le civil. S'il est ambitieux de commencer à vendre les technologies des drones spatiaux

alors qu'elles ne sont pas encore développées, nous pouvons néanmoins imaginer quels types d'innovations elles vont apporter. La mise en place de missions automatisées par des drones remplaçant les humains dans des lieux éloignés ou dangereux en fait partie.

Le projet AMOSS (Autonomous Mission for On Site Servicing) (LASSENA, 2013) est un projet du laboratoire LASSENA visant à exploiter les retombées du projet AMOOS dans le civil (LASSENA, 2013). La finalité étant de pouvoir créer une flotte de drones autonome capable d'être pilotée sur de très longues distances. La suite de ce mémoire sera dédiée à la conception d'une telle flotte de drones incluant l'interfaçage d'instruments d'avion pour créer un poste de pilotage à distance de façon robuste et sécuritaire, avec de la technologie certifiée pour l'aviation civile internationale.



## **CHAPITRE 3**

### **CONSTRUIRE AVEC LES TECHNOLOGIES COMMERCIALES**

L'intitulé du mémoire faisant allusion à un contrôle de drones à distance; nous avons commencé par bâtir une première flotte de drones avec un maximum de technologies présentes sur le marché afin de réduire le plus possible le temps de mise au point. Mais les technologies habituelles en modélisme sont encore très rudimentaires en termes de moyens de communications drone/station de contrôle. Nous verrons au cours de ce chapitre comment nous avons pu construire un premier prototype de drone et comment la portée de ces équipements commerciaux a pu être augmentée en utilisant une station locale au sol, permettant ainsi un premier pilotage à très longue distance. Ce chapitre fera également office d'introduction à l'architecture des drones, familiarisant ainsi le lecteur avec tous les termes utilisés par la suite dans le document.

#### **3.1 Construction du drone**

Comme l'a montré le chapitre I, il existe sur le marché une multitude de vendeurs de drones préconstruits mais aucun de ces fabricants ne proposaient de produit intéressant technologiquement et/ou financièrement pour nous.

Nous avons donc fait le choix de construire nos propres drones avec des éléments achetés séparément auprès de différents fournisseurs afin de répondre au mieux à nos attentes lors des missions, et notamment en termes de versatilité du drone.

##### **3.1.1 La structure : Walkera**

L'aéronautique est un perpétuel compromis entre les performances et l'autonomie, ainsi, un bon nombre de fabricants proposent des modèles de drones avec des charges utiles (« payload ») importantes et des portées de guidage sur plusieurs kilomètres. Mais ces

aéronefs consomment énormément d'énergie et leur autonomie s'en voit rapidement amputée. Aussi, la majorité des drones à voilure tournante sur le marché possèdent une autonomie d'une quinzaine de minutes seulement, ce qui est bien trop peu pour envisager sérieusement de faire du service sur site (« On-Site Servicing »).

Notre choix de structure a donc principalement été guidé par l'autonomie annoncée par les constructeurs et ensuite sur les capacités de la charge utile du drone.

Considérant une charge utile d'environ 1Kg pour garder une bonne flexibilité sur le type d'équipement à bord que nous pourrions emporter, l'envergure approximative de notre drone devait être d'environ 1 à 1.5 m et le poids d'environ 4 à 5 Kg à pleine charge.

Après de multiples recherches sur internet, nous nous sommes arrêtés sur le modèle QR X800 de Walkera. Son autonomie donnée est de 60 minutes à vide ce qui représentait la meilleure performance offerte sur le marché.

Le QR X800 (figure 3.1) est un drone quadrimoteur avec une structure en fibre de carbone à la fois solide et légère. Il mesure 1.2 m d'envergure et est donné pour un poids total volant de 3,9 Kg.



Figure 3.1 Walkera QR X800, tirée de Walkera 2015

Le drone choisi possédait donc toutes les qualités requises pour mener à bien des missions de durées relativement acceptables, de plus, il était vendu avec un support de suspension (« gimbal », système de stabilisation des équipements de prises de vues). Nous avons donc acheté 2 GoPro afin d'enregistrer nos vols et d'envoyer la vidéo au sol en temps réel ou différé.

Le seul inconvénient de ce drone est qu'il était vendu avec un régulateur de vol (calculateur entre les commandes de vol et les commandes moteur) très peu performant et réfractaire à la moindre modification. Les seules fonctions offertes sont le maintien d'altitude et la stabilisation assistée grâce au GPS intégré. L'intégration d'un nouveau régulateur de vol fut donc nécessaire.

### 3.1.2 DJI Wookong

La société DJI propose toute une gamme de régulateurs de vol et de structure pour tous types de drones, les performances et la fiabilité de leurs produits ne sont plus à démontrer. DJI est l'un des leaders du marché des drones à voilure mobile.

Parmi les produits de la gamme, nous avons choisi d'acheter le Wookong (figure 3.2) : un régulateur de vol pour les drones de taille moyenne comme le nôtre. Le Wookong est un système composé d'un calculateur, une centrale inertielle (IMU) et un capteur GPS avec magnétomètre.



Figure 3.2 DJI Wookong, tirée de AliExpress 2015

Le calculateur est basé sur un filtre de Kalman et permet une stabilisation et une précision à toute épreuve quel que soit le drone qu'il régule.

L'un des inconvénients du prototypage est la modification permanente des prototypes qui cause de nombreux changements de poids et de configuration, modifiant ainsi la géométrie du drone et donc son enveloppe de vol. Le principal avantage du Wookong est qu'il permet une

énorme flexibilité à ce point de vu, ainsi pendant tous nos essais, nous n'avions pas à nous soucier de l'équilibrage des prototypes et du changement de masse due à un nouveau type de charge utile. De plus, un éventuel changement de structure (drone plus petit ou plus lourd) ne nécessite pas de nouveaux réglages logiciels et l'adaptation est quasi-instantanée.

Le Wookong possède également un mode de sécurité appelé FailSafe qui permet de ramener le drone à l'endroit où il a décollé, lors de nos différents essais en vol ce mode constituait notre dernière sécurité pour essayer d'épargner le matériel en cas de perte de contrôle. Le FailSafe peut être déclenché volontairement ou automatiquement si le Wookong détecte que le drone n'est plus à portée de la télécommande. Lorsque il est actif, le système de sécurité positionne le drone à 20m d'altitude et le ramène en ligne droite jusqu'au lieu du décollage grâce à son GPS intégré.

L'un des autres avantages du Wookong est qu'il possède des accessoires vendus séparément très intéressants pour votre projet. Nous avons donc ajouté un IOSD (Integrated On-Screen Display) : un système capable de simuler un affichage tête haute (HUD : Head Up Display) sur les vidéos transmises depuis le drone via la GoPro. Également, nous avons ajouté un Data Link permettant de transmettre des trajectoires de vol depuis le sol et de recevoir les informations de vol sur un logiciel spécifique pour permettre un pilotage automatique par points de cheminement.

### **3.1.3 Bras articulé**

La structure du drone et le régulateur ne suffisant pas à faire du « On-Site Servicing », nous avons donc entamé la construction d'un bras robotique muni d'un électroaimant afin de pouvoir mener des actions à distance.

Le cahier des charges de ce bras imposait deux contraintes principales :

- La possibilité d'accéder à tout le domaine de vision de la caméra;
- Être le plus léger possible, tout en étant robuste et précis.

Après quelques schémas préliminaires, nous avons décidé de construire un bras sur deux axes muni d'un manche télescopique pour atteindre toutes les zones à portée du drone et ce, à n'importe quelle hauteur.

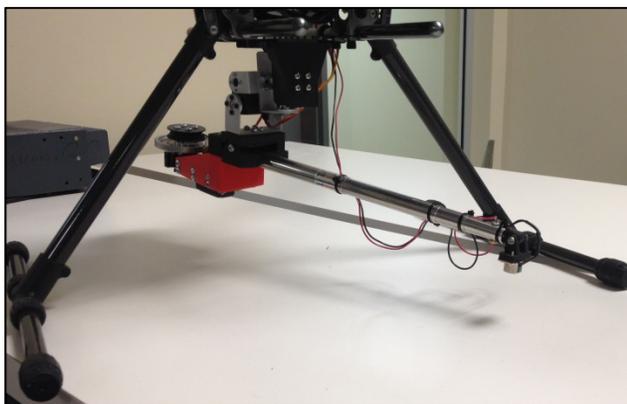


Figure 3.3 Bras articulé

La conception mécanique et électrique du bras à la figure 3.3 a été réalisée de A à Z au LASSENA et est donnée en annexe III de ce document.

Les différentes technologies intégrées lors de sa fabrication nous ont permis de commander le bras à distance depuis le sol en utilisant le multiplexage des signaux de commande.

## 3.2 Architecture matérielle

Les principaux éléments structurels du drone ayant été achetés et/ou conçus, il nous fallut mettre au point toute l'architecture matérielle à bord du drone et au sol.

### 3.2.1 À bord du drone

Pour répondre à nos besoins de navigabilité et d'opérativité à distance, l'architecture interne de notre drone commença à devenir relativement complexe et il était devenu indispensable d'en faire les plans. Le tableau 3-1 ci-dessous fait l'inventaire des différents modules nécessaires pour mener nos missions selon 5 catégories :

- Les capteurs : pour déterminer la position du drone et capter les éléments extérieurs.
- Les actionneurs : pour mouvoir le drone.
- Les contrôleurs : pour faire les calculs embarqués.
- Les moyens de communications : pour communiquer entre le drone et le sol.
- Les moyens de gestion de l'énergie : pour alimenter les systèmes à bord.

Tableau 3-1 Inventaire des différents modules nécessaire au drone RC

Capteurs	Actionneurs	Contrôleurs	Communications	Énergie
GPS	Moteurs + ESC	Wookong	Data Link	Batterie
INS	Support de suspension	IOSD	Récepteur RC	Transfo 5V
GoPro	Train d'atterrissage	Arduino	Transmetteur vidéo	
	Bras			
	LED			

Pour interconnecter tous ces éléments, on compte plusieurs types de liaison parmi lesquelles :

- Les alimentations électriques (5V et 22V);
- Les signaux PWMs (pour la commande des servomoteurs et ESCs);
- Le Can Bus DJI (bus propriétaire pour faire communiquer les éléments DJI);
- Des connections physiques (maintien de la GoPro par le support de suspension, par exemple).

Le schéma bloc final déterminé est visible à la figure 3.4 et un aperçu du montage physique est disponible à la figure 3.5 :



### 3.2.2 Station locale au sol

Pour contrôler le drone et réceptionner les différentes informations envoyées, il nous fallut une infrastructure au sol restant à portée RF du drone dans les bandes de fréquences commerciales.

Habituellement, les drones commerciaux comme le nôtre sont des aéronefs récréatifs et restent à la portée visuelle de l'utilisateur, c'est la raison pour laquelle les moyens de communication sont uniquement analogiques et ont un rayon d'action de 2km environ.

Dans la configuration ci-dessus, pas moins de 3 fréquences radio étaient nécessaires pour piloter chaque drone :

- Un canal 2.4GHz pour l'envoi de données à la télécommande;
- Un canal 5.8Ghz pour réceptionner la vidéo envoyée par le drone;
- Un canal 2.4GHz pour communiquer avec le module DataLink du drone.

Chacun de ces canaux avaient leur propre utilité et devaient être capables d'être interfacé avec un ordinateur pour envisager un contrôle à distance (« Remote Control »). La difficulté pour nous fut donc de convertir ces données analogiques en données numériques pour que l'ordinateur puisse les interpréter.

#### 3.2.2.1 Transmission de la vidéo

À bord du drone, la capture d'image est faite grâce à une caméra vidéo GoPro. Les images sont ensuite retraitées par l'IOSD qui ajoute les informations de vol afin que l'utilisateur ait une vision ressemblant à celle d'un HUD. Les images sont ensuite transmises au sol par l'entremise d'un émetteur et un récepteur RC à 5.8 GHz (analogique). Dans un dernier temps, les images sont finalement converties numériquement grâce à un convertisseur analogique à numérique qui les transmet à l'ordinateur par un lien USB. La chaîne d'acheminement du signal vidéo est résumée dans la figure 3.6.

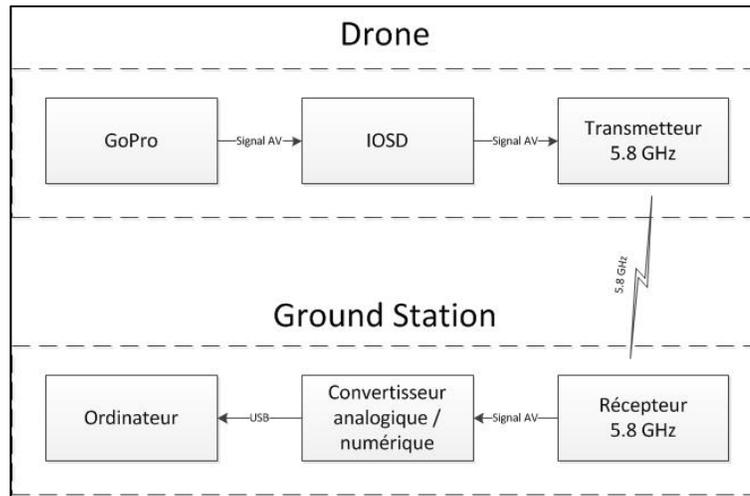


Figure 3.6 Acheminement du signal vidéo

### 3.2.2.2 Transmission des commandes

La numérisation des commandes de vol fut légèrement plus complexe que pour la vidéo, la commande du régulateur de vol Wookong ne peut être faite qu'en PWMs et la transmission des signaux est faite via une télécommande/récepteur RC à 2.4GHz. Le problème est que les télécommandes sont pour la plupart uniquement analogiques et ne font que convertir les données issues des joysticks en PWMs.

La solution proposée fut d'utiliser un module PCTX de la société Endurance R/C. Cet adaptateur permet d'utiliser le mode instructeur de certaines télécommandes pour générer des PWMs commandés par un ordinateur. La chaîne d'acheminement des commandes de vol est résumée dans la figure 3.7.

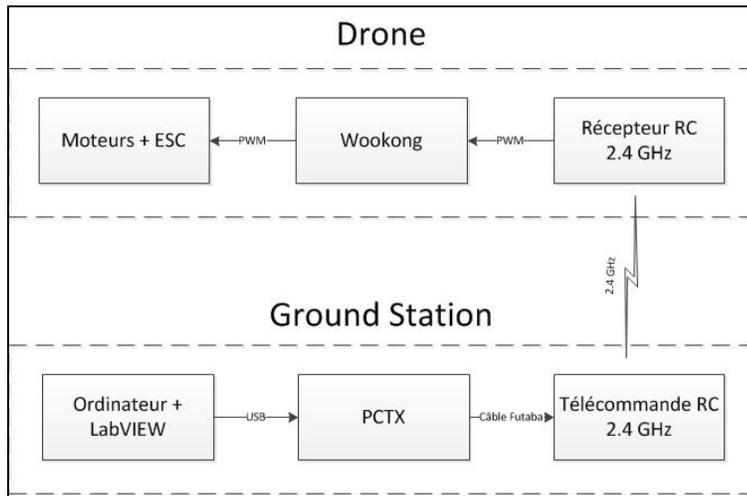


Figure 3.7 Acheminement des commandes de vol

Cependant, le module PCTX était développé que très sommairement et un interfaçage logiciel sur le PC fut nécessaire. Nous avons choisi d'utiliser le logiciel LabVIEW pour effectuer cette adaptation.

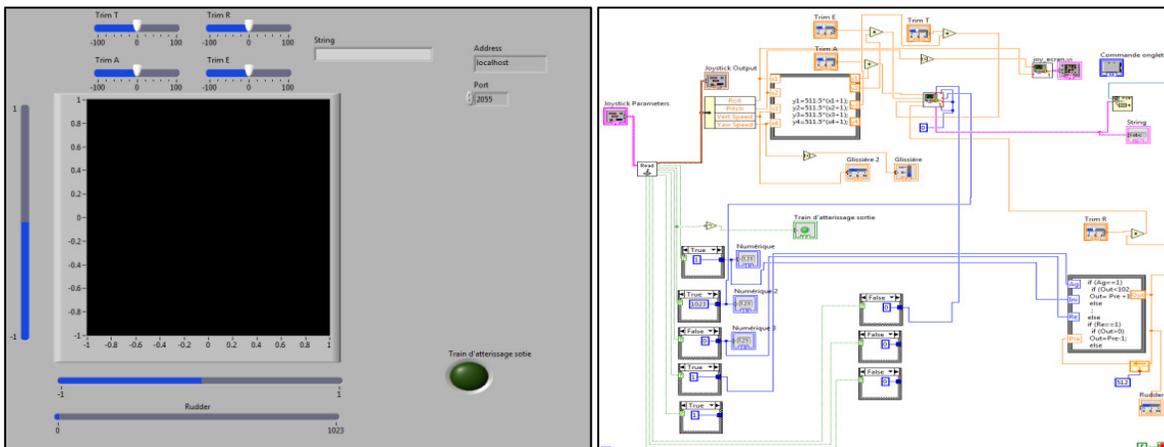


Figure 3.8 Interface de pilotage LabVIEW

Notre adaptation logiciel (illustré à la figure 3.8 et détaillé en annexe IV) intégrait tous les trimes nécessaires pour calibrer les axes du drone ainsi que des boutons spécifiques (démarrage des moteurs, arrêt d'urgence des moteurs et train d'atterrissage). Grâce à cette interface, il était possible de commander le drone avec un joystick d'ordinateur quelconque.

L'intérêt n'étant pas de changer de type de joystick mais d'être capable de numériser les commandes envoyées au drone. De cette manière il est possible de transmettre et de recevoir les commandes depuis un autre poste de pilotage plus éloigné disposant d'un système de communication longue distance plus performant (LTE ou satellite).

À ce point-ci, nous étions donc capables de traiter informatiquement toutes les commandes envoyées et reçues par le drone (commandes de vol, vidéo et DataLink). Nous avons donc commencé à créer des protocoles de communication pour déporter le centre de pilotage hors de la portée RF du drone vers un poste de pilotage à distance.

### **3.2.3 Poste de pilotage à distance**

L'un des objectifs du projet étant de commander le drone depuis une distance très éloignée, nous avons donc bâti une structure à l'ÉTS capable de commander toutes les fonctions du drone. Il nous fallut donc identifier des technologies de télécommunication longue portée, efficace et robuste, tout en étant à la fine pointe de la technologie, donc à faible coût, dimension et poids.

Pour gérer la transmission des informations entre le drone et poste de pilotage de l'ÉTS, nous avons choisi d'utiliser le réseau LTE qui est en pleine expansion et couvre une bonne partie de la planète. De plus, la vitesse de son flux est largement capable de faire passer une vidéo en streaming sans problèmes de décalage (parfait pour le pilotage manuel). Ainsi, nous avons équipé notre Station au sol d'un router LTE nous donnant un accès à internet depuis n'importe quel endroit sous la couverture réseau.

Notre structure de communication entre l'école et le drone est illustrée par la figure 3.9 suivante :

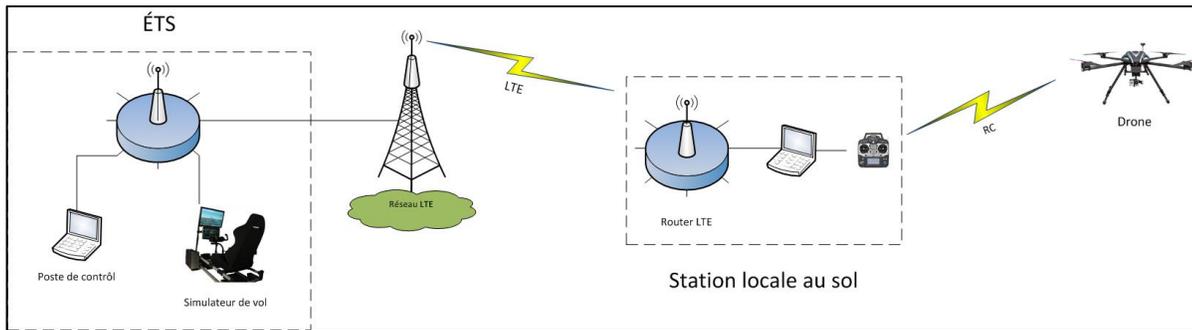


Figure 3.9 Réseau de communication entre l'ÉTS et le drone

D'un point de vue logiciel, nous avons acheminé la vidéo et les informations du DataLink grâce au logiciel TeamViewer qui permet d'avoir un bureau à distance. Ainsi, tous nos agissements depuis l'ÉTS était relayés jusqu'à l'ordinateur de la station locale au sol qui lui-même transmettait les informations au drone.

En ce qui concerne les commandes de vol, le logiciel TeamViewer ne permettait pas d'avoir accès aux ports USB de l'ordinateur distant. Nous avons donc modifié notre programme LabVIEW pour acheminer les commandes d'un ordinateur à un autre via une procédure TCP/IP. Une application LabVIEW tournait sur l'ordinateur à l'ÉTS et faisait office de serveur tandis qu'une autre application LabVIEW tournait sur l'ordinateur de la station locale au sol et faisait office de client.

À ce stade, nous disposions d'une structure capable de piloter un drone aussi bien en mode manuel (grâce au joystick et à la liaison vidéo qui transmettait les données de vol nécessaire) qu'en mode automatique (grâce au module DataLink qui permet de programmer des plans de vol par points de cheminement).

### 3.2.4 Simulateur de vol HeliMod

Pour améliorer encore notre poste de pilotage à distance, nous avons utilisé le simulateur de vol HéliMod de l'école. Ce simulateur (figure 3.10) est composé de 6 écrans (dont 3 tactiles) disposés tout autour de l'utilisateur, il est également composé d'un joystick et d'une commande de gaz.



Figure 3.10 Simulateur de  
Vol HéliMod

La disposition des multiples écrans tout autour de l'utilisateur offre la possibilité d'afficher de nombreux paramètres, parfait pour notre drone qui demandait au minimum 3 écrans pour afficher toutes les informations (vidéo, transmission commandes de vols et carte du DataLink). Nous avons donc temporairement modifié la structure du simulateur afin de l'adapter à nos besoins. Les écrans supérieurs ont été assignés à la transmission vidéo avec un HUD en face de l'utilisateur et une cartographie sur la droite. Les écrans inférieurs ont été utilisés pour afficher les contrôles à distance de TeamViewer sur la gauche, l'IHM de LabVIEW au centre et les contrôles secondaires de LabVIEW sur la droite.

Une structure complète de poste de pilotage à distance en LTE était donc disponible au laboratoire LASSENA pour offrir un service de maintenance sur site « On Site-Servicing » quel que soit l'emplacement géographique du drone à condition qu'il soit sous une couverture LTE. Toutes les zones à l'extérieur des couvertures LTE seront garanties par un autre système de communication d'avant-garde qui sera présenté plus loin.

### 3.3 Essais en vol

Pour terminer le développement de notre première plateforme de contrôle de drone, nous avons effectué quelques tests de pilotage à distance.

Nous nous sommes rendus sur l'île des sœurs à proximité de Montréal pour effectuer nos tests à une distance raisonnable des zones urbaines et pas trop éloignée pour être sous couverture LTE.

Le problème de la réalisation de tels tests est qu'ils demandent des pilotes expérimentés pour contrôler le drone à distance et tenter de lui éviter un écrasement si une erreur se produit. Également et au-delà des éventuels dégâts matériels, il peut également survenir des accidents impliquant des humains. En effet, les drones sont équipés de 4 hélices en fibre de carbone longues de 16 pouces et tournant à près de 12000 RPM. Un impact sur le bout d'une hélice à pleine vitesse se fait à environ 900km/h, de quoi causer de sérieuses blessures. Une attention particulière à la sécurité était donc primordiale.

Lors des tests, nous avions deux types de sécurité :

1. Une prise en main directe du drone avec la télécommande RC était possible en tout moment afin de contourner les commandes émises depuis l'ÉTS.
2. Le mode FailSafe du Wookong pouvait également être activé à tout moment depuis la télécommande ou depuis l'ÉTS afin de ramener le drone à sa position initiale. Nous l'avons mise en œuvre et la fonctionnalité est tout à fait au point.

Les essais ont donc toujours été effectués avec la présence d'un pilote de sécurité, les mains sur la télécommande pour réagir en une seconde maximum.

Nous avons commencé les tests en implémentant des points de cheminement (« Waypoints ») au drone pour lui faire suivre une trajectoire grâce au DataLink. Le résultat fut plutôt satisfaisant et le drone effectua la mission sans problèmes particuliers.

Lors d'un second test, nous avons tenté de piloter le drone manuellement grâce au simulateur HéliMod. Le problème est que l'HéliMod n'est pas équipé d'un joystick de qualité et le manche ne dispose pas de zone neutre où il revient systématiquement quand on le lâche. Ce détail peut sembler infime mais il permet au pilote de recentrer son appareil seulement au toucher, ce qui réduit considérablement sa charge de stabilisation de l'aéronef.

Pendant les essais, le pilote à l'ÉTS a réussi à faire décoller le drone et le maintenir en l'air pendant quelques instants, mais rapidement, il a perdu le contrôle à cause du problème

mentionné ci-dessus. Grâce à la préparation post expérimentale, le second pilote sur place a rapidement repris le drone en main et est parvenu à le poser sans dégâts.

Les tests suivants depuis l'ÉTS ont donc été arrêtés prématurément le temps de trouver un meilleur matériel de pilotage.

Cependant, certains autres paramètres relativement intéressants pour la suite du projet ont pu être relevés :

1. Le temps de transmission des données entre le drone et l'ÉTS a été estimé à environ 500 ms.
2. L'autonomie des drones avec tous les accessoires à bord (sauf le bras) a été mesurée à une trentaine de minutes.

Bien que les tests furent relativement encourageants, notre cahier des charges fut modifié par notre Directeur, demandant d'éliminer la station locale au sol (car trop encombrante et peu pratique comme le montre la figure 3.11) pour commander le drone directement via le signal LTE ou par liaison satellite.

Cependant, cette décision modifia en profondeur la structure matérielle de notre drone comme il sera présenté dans le prochain chapitre.



Figure 3.11 Équipements requis pour l'opération de la station locale au sol

## CHAPITRE 4

### SUPPRESSION DE LA STATION LOCALE AU SOL

Comme annoncé plus haut, à la suite d'une importante décision du Prof. Landry, la station locale au sol fut supprimée, nous obligeant ainsi à changer totalement la structure matérielle et logicielle du drone.

#### 4.1 Nouvelles contraintes

Le nouveau drone fut principalement contraint par ses systèmes de communication. En effet, l'utilisation des technologies LTE et satellitaires sur les drones n'existe pas dans le domaine commercial et il nous a donc fallu innover pour contourner le problème.

La principale difficulté étant que ces technologies sont principalement utilisées par des ordinateurs; elles demandent une architecture logicielle pour les faire fonctionner. Notre nouveau drone devait donc être équipé d'un ordinateur de bord pour maîtriser ces technologies de communication.

Également, compte tenu du fait que la navigation et le positionnement n'étaient plus gérés par la technologie Wookong, il nous fallut envisager la possibilité d'ajouter de nouveaux capteurs à bord pour déterminer l'emplacement exact et l'inclinaison du drone.

De plus, compte tenu de la nouvelle architecture et des nouvelles contraintes d'autonomie, la sécurité pour l'utilisateur et la structure de la machine devait être renforcées.

Si on récapitule, les seules structures restant intactes étaient le corps du drone avec ses moteurs et ses hélices et le Wookong faisant office de stabilisateur de vol. Toutes les fonctions de pilotage à distance ou à proximité aussi bien automatiques que manuelles furent modifiées car ces nouvelles contraintes supprimèrent totalement tous les systèmes RC à la base de notre précédente architecture.

## 4.2 Nouvelle architecture

La nouvelle architecture matérielle souhaitée comportait donc un ordinateur de bord ainsi que ses propres systèmes de communications et capteurs. Cette section présentera les différents éléments choisis pour notre drone.

### 4.2.1 La Raspberry Pi

Nous avons commencé par choisir un ordinateur de bord pour faire le lien entre les différents capteurs et actionneurs à bord.



Figure 4.1 Raspberry Pi, tirée de Clubic

Notre choix s'est arrêté sur la Raspberry Pi (figure 4.1) de la société du même nom, ses avantages sont multiples :

1. Un faible coût (aux alentours de 40\$);
2. Un faible poids (moins de 50g);
3. Une vitesse d'exécution assez élevée (un processeur cadencé à 900 MHz);
4. Un système d'exploitation libre, ce qui signifie que tous les logiciels fonctionnant dessus sont libres et gratuits;
5. Une architecture très modulable car une grande communauté utilise ce genre de système.

La Raspberry offre une multitude de possibilité de connexion de périphériques, on trouve :

- 4 ports USB;
- Un connecteur de carte SD;
- Un port HDMI;
- Un port Ethernet;
- Des connecteurs pour caméra et écran;
- Des interfaces I2C, SPI et UART;
- 26 pines GPIO.

Il existe également un grand nombre de modules compatibles qui permettent d'étendre les possibilités de connexion de la Raspberry. Ainsi, nous avons acheté un module Servo Pi capable de générer 16 PWMs simultanément et pouvant être lu par le Wookong.

La Raspberry est programmable en C et en Python, nous offrant ainsi une certaine flexibilité pour implémenter nos futurs algorithmes.

#### **4.2.2 Les transmissions de données**

L'ordinateur de bord choisi, nous nous sommes ensuite penchés sur les moyens de communication mobiles compatibles et de tailles acceptables pour être embarqués sur le drone.

##### **4.2.2.1 LTE**

Selon les recommandations de mon directeur de maîtrise, nous nous sommes concentrés sur une technologie LTE pour transmettre la majorité de nos données entre le drone et le sol. La LTE ou plus communément appelée 4G par les opérateurs téléphoniques est actuellement l'accès à internet mobile le plus rapide, son débit théorique est de 300Mbit/s et pourrait évoluer jusqu'à 1Gbit/s. Les possibilités du LTE sont énormes et permettent de faire passer une grande quantité de données, notamment un flux vidéo constant (parfait pour notre cas).

La couverture réseau 4G est de plus en plus rependue et aujourd'hui la moitié de la planète en est équipée comme le montre la figure 4.2.

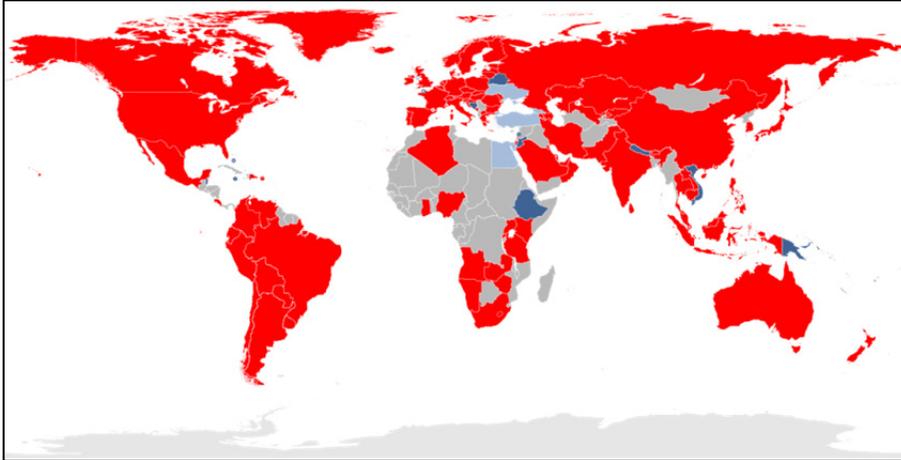


Figure 4.2 Couverture réseau LTE mondiale, tirée de Bennett 2014

Compte tenu de notre statut particulier qui nous obligeait à utiliser une adresse IP fixe pour faciliter notre programmation, nous nous sommes tournées vers le seul opérateur téléphonique proposant ce service (TELUS) avec un abonnement par utilisation mensuelle. TELUS couvre une bonne partie du Canada et dispose d'un bon réseau dans la zone de Montréal, parfait pour nos tests. Nous avons donc acheté une clé USB 4G (figure 4.3) pour équiper notre drone et pouvoir le piloter à distance sans station locale au sol. La clé LTE/USB/TELUS est légère et peu encombrante, donc parfaite pour un système embarqué comme le nôtre.



Figure 4.3 Clé LTE/USB/TELUS, tirée de Huawei 2014

#### 4.2.2.2 Liaison satellite

Comme cité plus haut, le réseau LTE n'est pas encore mondial et même dans les pays équipés, beaucoup de régions isolées ne sont pas couvertes.

Pour pallier à cette absence du réseau LTE qui empêche toute communication avec le drone et ayant comme autre objectif d'offrir une solution à couverture mondiale, nous avons envisagé la possibilité d'utiliser le réseau de communication satellitaire qui, de son côté, couvre la planète entière, au même titre que le système GPS pour la navigation des mobiles.

Un partenariat avec la société Globalstar nous a permis d'obtenir deux téléphones par satellite type GSP-1700 (présenté à la figure 4.4). Ces téléphones permettent de passer des communications téléphoniques et des données depuis des zones hors de couverture réseau classique, complétant ainsi notre solution basée sur la technologie LTE.



Figure 4.4 Téléphone GSP-1700 Globalstar, adaptée de Globalstar 2014

Les téléphones Globalstar ont la capacité d'être connectés à un ordinateur en USB pour servir de modem mais possèdent cependant beaucoup de limites (à ce jour), dont :

- Le débit théorique de données est limité à 9.6 Kbit/s donc extrêmement lent.
- La couverture est mondiale mais n'est ni instantanée, ni intemporelle. Il n'est pas rare de devoir attendre la disponibilité d'un satellite avant de pouvoir utiliser le réseau et des ruptures de connexion sont relativement fréquentes.

- La transmission de données est extrêmement onéreuse.
- Pas d'adresse IP statique.

Compte tenu de ces éléments, l'utilisation de cette technologie à bord d'un drone est limitée et même impossible pour certaines fonctionnalités (transmission de vidéo en temps réel, par exemple).

Nous avons donc pris la décision d'utiliser ce moyen de communication comme secondaire et uniquement pour envoyer des ordres prioritaires au drone (comme, par exemple, l'envoi de nouvelles coordonnées de navigation ou l'ordre d'atterrissage).

L'utilisation de la clé LTE fut priorisée par rapport aux téléphones par satellite pour communiquer avec le drone, ce qui est normal à cette étape-ci de nos investigations. Cependant, ils ne furent pas pour autant mis au rebut et leur usage fut classé comme sécuritaire en cas de rupture de communication via le LTE.

Cependant la deuxième génération de satellites Globalstar est en cours de déploiement et permet d'espérer des débits de données théoriques suffisamment grands pour permettre la transmission de vidéo en temps réel (1 Mb/s dans le sens descendant et 256 kb/s dans le sens montant). Ainsi nous pourront bientôt tester des pilotages de drone en temps réel à très longue distance par des réseaux satellitaires civils.

#### **4.2.2.3 Émetteurs RF 2.4 GHz**

Dans le but de pouvoir communiquer avec la Raspberry à courte distance, nous avons également voulu équiper le drone d'un petit émetteur/récepteur RF à 2.4 GHz. Leur portée n'est pas très grande (1 Km), ils servirent uniquement de relais de communication courte distance pour la sécurité et la transmission des données de vol.

Leur poids très faible et leur faible consommation de courant ne présentent pas de gênes pour les embarquer à bord. Au cours de nos tests, ils furent utilisés pour communiquer avec un ordinateur au sol (collecte de données de vol et interprétation directe) mais aussi pour servir de relais de communication entre drones.

### 4.2.3 Les systèmes de sécurité

Comme l'usage des drones est relativement dangereux en général et encore plus en cas d'un système autonome comme le nôtre, nous avons mis en place quelques infrastructures matérielles afin de détecter et d'intervenir sur d'éventuelles pannes durant les vols d'essai.

#### 4.2.3.1 Le sélecteur de canaux

Compte tenu des caractéristiques autonomes et dynamiques du drone en cas de coupure de liaison avec le sol, nous avons décidé de créer un système de sécurité capable de reprendre le drone en mode manuel avec une télécommande RC. Ce système se devait d'être rapide et simple à déclencher. Nous avons donc mis au point un circuit électrique capable de sélectionner la source (télécommande RC ou Raspberry) des PWMs envoyés au Wookong (qui lui-même dirige le drone). Il était important que ce système soit le plus simple possible et surtout indépendant de tout autre système automatique (notamment la Raspberry).

Le développement de ce circuit est donné en annexe V de ce document.

L'architecture matérielle de ce sélecteur de canaux est représentée dans la figure 4.5 suivante :

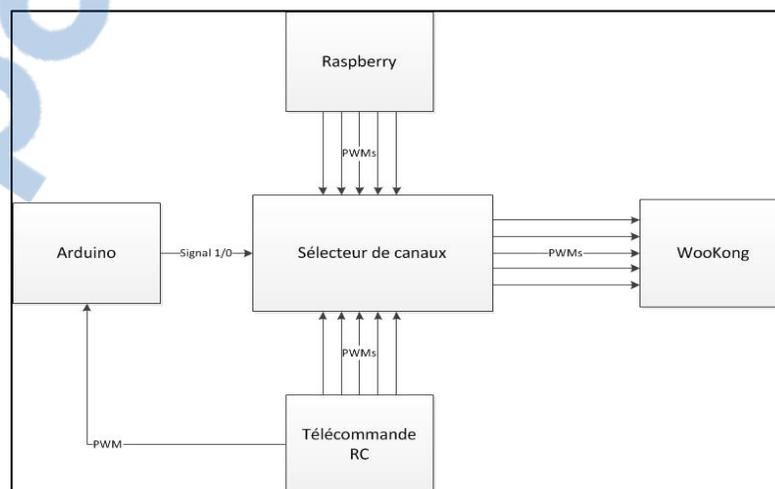


Figure 4.5 Architecture matérielle du sélecteur de canaux

Bien qu'un peu complexe à première vue, ce système est très efficace et constitue une excellente sécurité pour le drone car il permet de prendre le contrôle du drone immédiatement et indépendamment du type de panne de l'ordinateur de bord. Son utilisation fut sollicitée à de nombreuses reprises lors des tests en vol.

#### 4.2.3.2 Détecteur de panne Raspberry

L'une des pannes les plus probables et les plus dangereuses pouvant se passer pendant un vol est un bug dans le programme de la Raspberry. Ce type de panne cause un blocage des commandes PWMs envoyées au Wookong (les dernières valeurs restent en mémoire dans le module PWM de la Raspberry). En imaginant que les dernières commandes soient une vitesse maximale vers l'avant, il est très dangereux de ne pas pouvoir arrêter un drone de 4 Kg lancé à pleine puissance.

Nous avons donc cherché à détecter un éventuel blocage de la Raspberry pendant son fonctionnement normal.

Pour cela, nous avons ajouté une ligne de code produisant un signal PWM aléatoire dans la boucle principale du programme de la Raspberry. Ce signal aléatoire est émis sur un canal indépendant des commandes de vol et est envoyé vers un Arduino. Avec son horloge interne (« timer »), l'Arduino est capable de détecter si les signaux émis par la Raspberry sont effectivement aléatoires. Dans le cas contraire, il émet un signal bloquant les commandes de vol de la Raspberry et déclenchant un GoHome sur le Wookong comme illustré dans la figure 4.6.

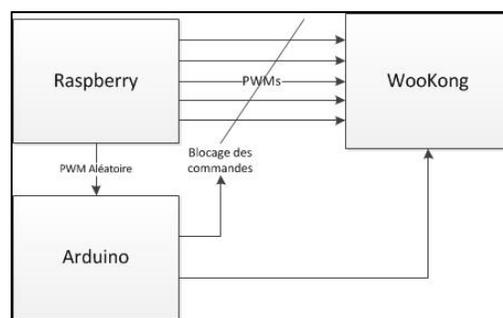


Figure 4.6 Détection de panne Arduino

La boucle principale du programme de la Raspberry étant effectuée toutes les secondes au maximum, une éventuelle panne est donc détectée en moins de 2 secondes et ramène le drone à son point de départ.

Entre le sélecteur de canaux et le détecteur de panne sur Raspberry, nous avons donc deux sécurités (la première manuelle et la deuxième automatique) pour débiter nos tests avec le nouveau drone autonome sans dépendance d'une station locale au sol.

#### 4.2.4 Les capteurs

Pour piloter un drone, nous avons principalement besoin de deux types de capteurs :

- une centrale inertielle avec des accéléromètres et des gyroscopes pour déterminer notre vitesse et nos angles d'Euler.
- un capteur GPS pour avoir notre position dans l'espace et faire de la navigation.

Les capteurs inertiels sont pour la plupart des centrales composées de 3 accéléromètres et 3 gyroscopes qui permettent de déterminer une position sur les 6 axes (3 translations et 3 rotations). Il existe différents types de technologies pour faire une centrale inertielle, les principales sont la technologie gyroscopique mécanique, la technologie laser et les MEMS. Chacune de ses technologies possède ses avantages et ses inconvénients (taille, poids, prix, précisions, etc.), or la plus appropriée pour nous fut la technologie MEMS (la moins cher, la moins encombrante et la moins lourde). Le problème étant que les capteurs MEMS ne sont précis que quelques secondes et dérivent rapidement (2 à 3 secondes, dans cette gamme de qualité de capteurs).

Les capteurs GPS sont eux de plus en plus répandus et tous relativement basés sur la même technologie. Ils recueillent les signaux émanant des satellites GPS (4 au minimum) et déterminent une position absolue sur la planète. Leur précision varie d'un capteur à un autre mais on observe une précision d'environ 1 mètre en entrée de gamme.

Ces deux types de capteurs sont complémentaires (l'inertiel dérive et le GPS est peu précis) et sont souvent couplés grâce à des algorithmes basés sur des filtres de Kalman.

Nous avons choisi de diriger notre drone avec ces deux types de capteurs pour avoir une position absolue et précise à la fois. Pour cela, nous avons acheté un capteur inertielle UM7 de CHrobotics®, la particularité de ce capteur est qu'il intègre un algorithme de Kalman qui compense et corrige directement les erreurs inertielles. De plus, l'UM7 offre la possibilité de brancher directement un capteur GPS en NMEA pour donner une position absolue et précise sans que l'utilisateur n'ait à corriger les différentes erreurs (dérivation, bruits divers, désalignement, etc.)

Notre système de positionnement absolu étant choisi, il se posait encore un problème : la position donnée par notre centrale nous donne une altitude relative par rapport à l'ellipsoïde de la terre (selon la norme WGS84, (NIMA, 2000)) et non pas par rapport à la terre elle-même. En conséquence, nous ne pouvions pas déterminer notre altitude par rapport au sol, ce qui pose de gros problèmes pour ne pas risquer l'écrasement du drone.

Il existe une solution consistant à disposer d'une base de données contenant toutes les corrections d'altitude par rapport à la terre. Mais cette solution n'est qu'approximative car elle ne prend pas en compte les obstacles comme les arbres ou les immeubles et demande beaucoup de place mémoire.

Pour atteindre nos objectifs d'universalité de la technologie développée, nous avons donc choisi d'utiliser un capteur de proximité pour avoir une meilleure approximation de notre altitude par rapport au sol. Encore une fois, plusieurs technologies étaient à disposition : laser, ultrason, capacitif, etc.

Nous avons commencé avec un capteur ultrason (peu cher et avec une sensibilité relativement correcte jusqu'à 4m) avec la possibilité d'évoluer vers un capteur laser LIDAR plus tard (plus précis et avec une grande portée mais cinq fois plus cher).

Avec cette combinaison de 3 capteurs (présentés en figure 4.7), nous avons la possibilité de nous repérer précisément dans l'espace et d'avoir une idée relativement précise de notre altitude par rapport au sol.

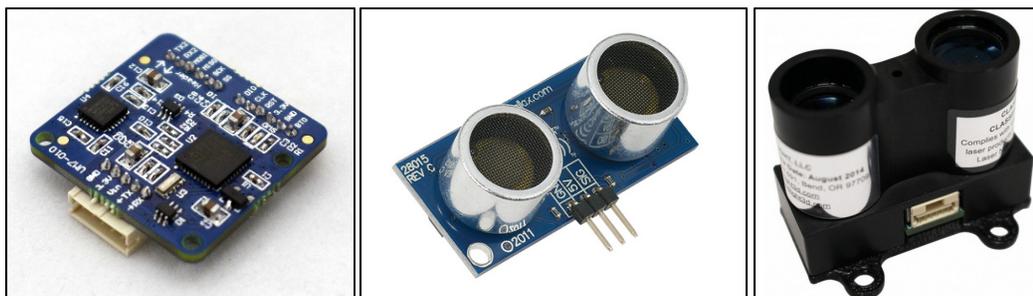


Figure 4.7 Capteurs UM7, Ultrason et Lidar, respectivement, tirée de ChRobotics

À ce stade, notre combinaison de contrôleurs, actionneurs et capteurs fut suffisante pour bâtir notre nouveau drone.

#### 4.2.5 CAN-Bus Wookong

Étant donné qu'avec notre nouvelle architecture, nous devions nous même nous charger de la navigation du drone, il nous fallait donc déterminer notre position et notre vitesse pour pouvoir nous diriger (le Wookong ne servant qu'à la stabilisation du drone et plus à la navigation).

Pour cela, nous avons envisagé de nous servir des capteurs du Wookong (très performants). Le seul problème est que la sortie des capteurs est directement connectée au CAN-Bus du Wookong cité plus haut. Il nous fallait donc trouver un moyen de décrypter le bus via la Raspberry.

Nous avons acheté un module capable de décoder les bus et branché le tout sur le Wookong. Il existe quelques forums sur internet décrivant comment décrypter le bus NASA (modèle en dessous du Wookong) mais peu font mention du bus Wookong, nous avons donc eu quelques problèmes de décodage.

Après plusieurs essais, nous avons remarqué que les informations du bus étaient cryptées avec un masque XOR dont nous n'avions pas la clé de décryptage.

De plus, la taille des trames sur le bus n'était pas constante pour un même label, nous en avons conclu que la Raspberry ne détectait pas tous les messages correctement. Pour vérifier, nous avons essayé de décrypter le bus avec un oscilloscope muni d'une option de lecture

CAN-Bus et un analyseur logique mais ces deux équipements trouvaient encore moins de trames que notre Raspberry.

Pour finir, la fréquence des trames portant les informations importantes n'était pas assez rapide et nous avons finalement pris l'initiative d'arrêter le décodage.

Nous fûmes donc obligés d'installer des capteurs supplémentaires sur notre drone, provoquant une redondance de certains éléments. Mais avec un peu de recul, il est beaucoup plus sain de disposer de ses propres capteurs plutôt que de se baser sur un décodage partiel d'informations dont nous n'avons aucune certitude.

De plus, interfacer nos propres capteurs permet de disposer d'une base afin de pouvoir créer notre propre Wookong lorsque les autres problèmes seront réglés.

#### 4.2.6 Prototype F450

Si nous récapitulons sous forme d'un tableau (4-1) l'ensemble des éléments du drone, nous remarquons qu'une bonne partie diffère par rapport à la description du chapitre 3.2.1.

Tableau 4-1 Inventaire des différents modules nécessaires au drone autonome LTE et satellite

Capteurs	Actionneurs	Contrôleurs	Communications	Énergie
GPS DJI	Moteurs + ESC	Raspberry	LTE	Batterie
INS DJI	Gimbal	Wookong	Récepteur RC	Transfo 5V
GoPro	Train d'atterrissage	IOSD	Satellite	
GPS	Bras	Arduino	RF 2.4 GHz	
INS	LED	Module PWM		
Ultrason				

Nous avons donc redéveloppé complètement l'architecture de la partie matérielle à bord du drone comme le montre la figure 4.8.



en vol autonome. Nous avons donc supprimé le support de suspension avec la GoPro, l'IOSD, le train rétractable et le téléphone satellite pour des raisons de poids.

Aussi, le F450 dispose d'une autonomie bien inférieure à celle du QR X800 (environ 10 minutes), son utilisation était donc réservée à la vérification des algorithmes de navigation et la transmission des données RF et LTE seulement. Ce 2<sup>e</sup> nouveau drone présenté à la figure 4.9 devenait ainsi une plateforme de test idéale pour toute activité d'ingénierie pour le test de nouvelle fonctionnalité jugée « périlleuse ».



Figure 4.9 Prototype du F450 modifié à la saveur du projet

#### 4.2.7 Poste de control

Le drone pour les essais étant terminé, nous nous sommes intéressés au poste de contrôle à distance du drone. Étant donné que les seuls moyens de communication longue portée (LTE et satellite) pour joindre le drone passent directement par Internet, un simple ordinateur avec joystick connecté à internet pouvait maintenant servir de poste de contrôle à distance pour le drone.

Le développement du logiciel de gestion du drone à distance fut réalisé sous Linux car l'architecture de la Raspberry et bien plus semblable au système linuxien qu'à celui de Windows.

Cependant, compte tenu de contraintes relatives à l'interfaçage du G1000, un second ordinateur sous Windows était néanmoins nécessaire pour communiquer les informations du drone vers des instruments d'avion (cette partie est développée dans le chapitre 5).

Pour communiquer directement avec le drone, nous avons utilisé une liaison SSH qui consiste à connecter deux machines avec leur adresse IP de manière sécuritaire. Le drone disposait d'une adresse IP statique (LTE TELUS) ce qui facilitait beaucoup la tâche, en revanche et par sécurité, l'ÉTS possède une gestion des adresses IP relativement complexe qui change les adresses des ordinateurs à chaque connexion.

Lors d'une connexion SSH, les adresses IP des machines sont nécessaires, nous étions donc obligés de démarrer une procédure depuis le poste de contrôle afin de communiquer notre adresse IP au drone. En cas de déconnection, le drone ne pouvait pas savoir à qui envoyer ses données car l'adresse IP du poste de contrôle changeait, cela nous compliqua quelque peu la tâche lors de la programmation.

#### **4.2.8 Réseau de communication entre drones**

L'idée est d'utiliser les antennes de communication RF 2.4 GHz des drones pour partager en permanence des informations entre eux, les avantages sont multiples et très intéressants :

- Les drones peuvent se communiquer leurs coordonnées et s'éviter les uns et les autres en calculant des trajectoires d'évitement.
- Si un drone perd le contact avec le sol avec sa LTE (sortie de la couverture réseau, panne d'antenne, etc.), les autres drones peuvent servir de relais de communication et lui transmettre quand même les ordres du poste de commandement.
- Grâce à ce système, les différents éléments de communications initialement prévus (LTE et satellite) peuvent être répartis sur plusieurs drones pour diminuer les charges emportées à bord et augmenter l'autonomie. Ainsi chaque drone peut disposer d'un équipement qui lui sera spécifique (un système de communication particulier par exemple) et permettre de créer des flottes de drones dont chaque individu est unique et dispose d'un élément qui servira à toute l'équipe.

Les applications d'une communication inter-drones sont aujourd'hui inexplorées et très intéressantes, elles ouvrent le champ de la spécialisation de chaque drone au sein d'une équipe.

Prenons l'exemple de la figure 4.10 pour une mission de sauvetage relativement éloigné, nous enverrons un drone avec un système de communication par satellite, deux autres drones avec les équipements de recherche et un dernier avec l'équipement de sauvetage. Le drone avec le système par satellite permet la communication de tous les drones avec le sol en faisant office de relais. Les drones de recherche utilisent leur autonomie pour trouver la cible et une fois qu'elle est repérée, le dernier drone s'occupe du sauvetage. Chaque drone a son propre équipement et sa mission, répartir les charges permet de diminuer les systèmes à bord et augmenter l'autonomie et accroître la robustesse globale en vue de l'atteinte des objectifs de la mission. Aussi, à chaque mission, la flotte de drone peut être ajustée en envoyant seulement les UAVs équipés des systèmes appropriés, cela évitera d'emporter des équipements inutiles et/ou redondants à bord (si chaque drone possède tous les équipements fixés pour être polyvalent, cela crée des systèmes embarqués inutiles à chaque missions). Le but est de fonctionner comme une division de fantassins : un commandant, un infirmier, un sniper, un agent de communication et des soldats. Chacun a sa spécialité et la répartition des tâches permet d'optimiser l'efficacité de l'équipe, le tout en utilisant des équipements d'ingénierie de qualité et certifiés.

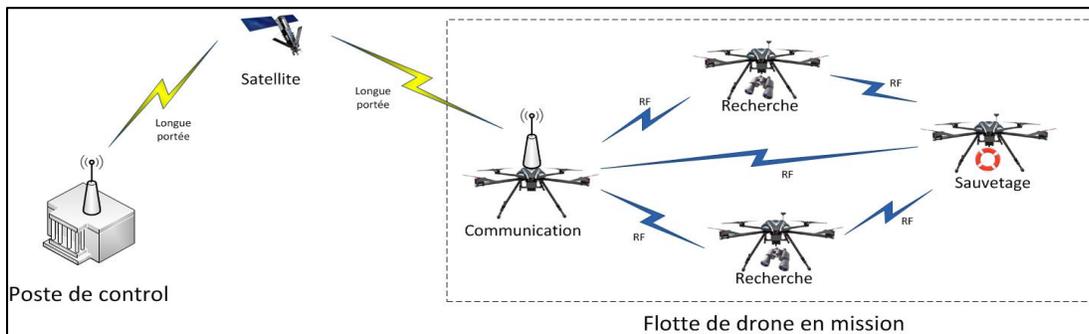


Figure 4.10 Exemple d'une équipe de drone en réseau local

### **4.3 Créer un système autonome**

Maintenant que notre système architectural matériel était prêt, il nous fallut programmer l'architecture logicielle de la Raspberry afin de permettre au drone de voler de manière autonome tout en restant en communication avec son poste de contrôle et en mettant en œuvre des missions complexes avec les fonctionnalités ci-haut présentées.

La difficulté à ce point réside dans l'architecture du logiciel interne du drone qui lui permet de changer de mode (autonome ou manuel) tout en continuant de fonctionner normalement.

Également pour nous permettre de voler de manière autonome, il nous fallut mettre au point un algorithme de navigation GPS capable de corriger un cap, suivre une route définie et éviter d'autres drones et/ou des obstacles.

#### **4.3.1 Algorithme de navigation**

Comme la navigation par points de cheminement n'était plus assurée par le Wookong et le DataLink, il nous fallut mettre au point un algorithme de navigation pour diriger le drone en mode autonome. Plusieurs auteurs comme Biagiotti (2010) et Markus (p 9, 2009) expliquent comment élaborer des trajectoires de vol en intérieur et en extérieur avec différents types de capteurs mais aucun ne traite directement avec un calculateur de vol comme le Wookong. Seul Hoffmann (p 5, 2004) traite du contrôle de suivi de trajectoire axe par axe mais ne développe pas les outils qu'il met en place. Il nous fallut donc créer notre propre algorithme de navigation adapté à nos besoins.

##### **4.3.1.1 Principes et calculs**

La difficulté d'un algorithme de navigation aérien est qu'il ne demande pas de simples corrections de cap pour atteindre le point final au plus vite. Il doit réorienter l'aéronef sur une route précise pour ne pas percuter d'éventuels obstacles au passage.

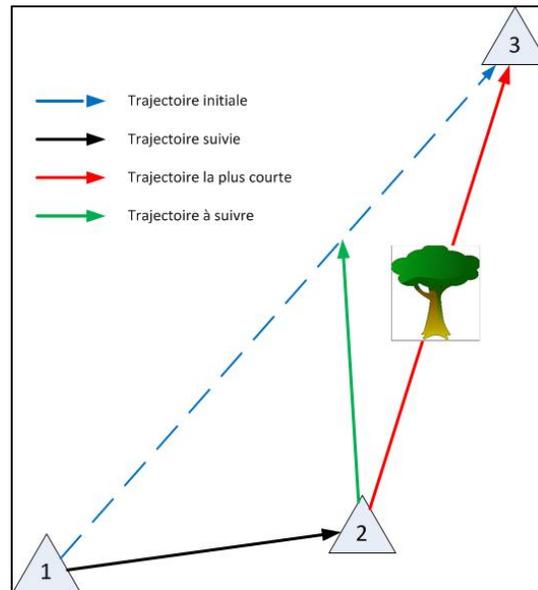


Figure 4.11 Explication des trajectoires

La figure 4.11 ci-dessus illustre l'importance de suivre les trajectoires initialement prévues. Si le drone part du point n°1 pour atteindre le point n°3 et se retrouve à la position n°2 pour une raison quelconque, il est impossible d'atteindre la position finale directement. Le drone est obligé de retourner sur sa trajectoire initiale pour ne pas percuter l'obstacle.

La figure 4.12 ci-dessous fera office de référence et explique la stratégie adoptée pour replacer le drone sur la route à suivre.

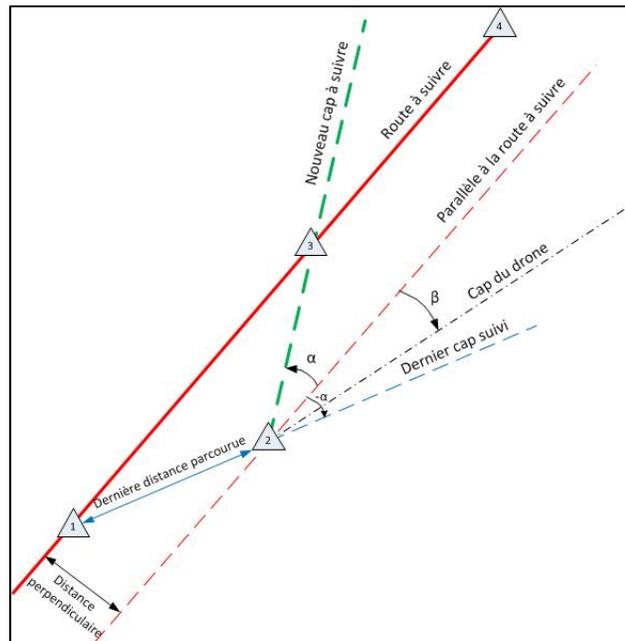


Figure 4.12 Correction de cap du drone

Lorsque le drone se retrouve en position n°2 suite à une perturbation quelconque, notre but est de le replacer sur la trajectoire verte pour qu'il regagne la route à suivre. La stratégie est la suivante : nous allons relever la dernière distance parcourue par le drone (entre la position n°1 et la position n°2) et calculer l'angle  $\alpha$  qui replacera directement le drone sur la route à suivre si l'ont parcouru la même distance.

Pour cet algorithme, nous allons nous placer dans un repère local orthonormé  $(x,y)$  en 2 dimensions.

Dans ces conditions, la représentation d'une droite dans l'espace est donnée par la formule suivante :

$$a * x + b * y + c = 0 \quad (4.1)$$

Par la suite, nous appellerons les coordonnées des points 1, 2, 3 et 4 respectivement  $(x_1,y_1)$ ,  $(x_2,y_2)$ ,  $(x_3,y_3)$  et  $(x_4,y_4)$ .

La droite correspondant à la route à suivre sera notée  $(a,b,c)$ .

$a$ ,  $b$  et  $c$  sont donnés par les équations :

$$a = y_4 - y_1 \quad (4.2)$$

$$b = -(x_4 - x_1) \quad (4.3)$$

$$c = -x_1 * a - y_1 * b \quad (4.4)$$

Et la dernière distance parcourue D est donnée par :

$$D = \sqrt{(x_1 - x_2)^2 + (y_1 - y_2)^2} \quad (4.5)$$

Pour déterminer les coordonnées du point n°3 ( $x_3, y_3$ ), il suffit de trouver l'intersection du cercle de centre n°2 et de rayon D avec la droite calculée ci-dessus.

Nous avons donc :

$$D^2 = (x_3 - x_2)^2 + (y_3 - y_2)^2 \quad (4.6)$$

et

$$x_3 * a + y_3 * b + c = 0 \quad (4.7)$$

D'après (4.7) :

$$x_3 = -((b * y_3 + c)/a) \quad (4.8)$$

(4.6) devient :

$$y_3^2 * \left(1 + \left(\frac{b}{a}\right)^2\right) + y_3 * 2 * \left(\frac{b}{a} * \left(x_2 + \frac{c}{a}\right) - y_2\right) + \left(x_2 + \frac{c}{a}\right)^2 - D^2 + y_2^2 = 0 \quad (4.9)$$

Il faut résoudre cette équation au second degré avec :

$$\Delta = \left( 2 * \left( \frac{b}{a} * \left( x_2 + \frac{c}{a} \right) - y_2 \right) \right)^2 - 4 * \left( y_2 * \left( x_2 + \frac{c}{a} \right) - D^2 \right) * \left( 1 + \left( \frac{b}{a} \right)^2 \right) \quad (4.10)$$

Si  $\Delta > 0$ , les deux solutions sont :

$$y_{3_1} = \left( -2 * \left( \frac{b}{a} * \left( x_2 + \frac{c}{a} \right) - y_2 \right) - \sqrt{\Delta} \right) / \left( 2 * \left( 1 + \left( \frac{b}{a} \right)^2 \right) \right) \quad (4.11)$$

$$y_{3_2} = \left( -2 * \left( \frac{b}{a} * \left( x_2 + \frac{c}{a} \right) - y_2 \right) + \sqrt{\Delta} \right) / \left( 2 * \left( 1 + \left( \frac{b}{a} \right)^2 \right) \right) \quad (4.12)$$

Avec l'équation (4.8), il est possible de déterminer  $x_{3_1}$  et  $x_{3_2}$ . Nous avons donc deux couples de coordonnées possibles  $(x_{3_1}, y_{3_1})$  et  $(x_{3_2}, y_{3_2})$ . Il suffit d'utiliser la formule (4.5) pour déterminer quelle position est la plus proche de  $(x_4, y_4)$ .

Si  $\Delta < 0$ , il faut replacer le drone sur la trajectoire lui permettant de rejoindre sa route initiale au plus vite. Il faut donc calculer l'intersection de la droite perpendiculaire à la route initiale et passant par le point n°2 et la droite  $(a, b, c)$ .

La droite passant par le point n°2 est donnée par le couple  $(b, -a, -x_2 * b + y_2 * a)$ .

Il faut trouver la solution du système :

$$\begin{aligned} a * x_3 + b * y_3 + c &= 0 \text{ et} \\ b * x_3 - a * y_3 + (-x_2 * b + y_2 * a) &= 0 \end{aligned} \quad (4.13)$$

Soit :

$$\begin{aligned} x_3 &= -\left( \frac{b * (-x_2 * b + y_2 * a)}{a + c} \right) / \left( \frac{a + b^2/a}{a} \right) \text{ et} \\ y_3 &= \left( \frac{b * x_3 + (-x_2 * b + y_2 * a)}{a} \right) \end{aligned} \quad (4.14)$$

Si dans l'équation de la droite,  $a=0$  les équations (4.7), (4.10), (4.11), (4.12) et (4.14) deviennent respectivement :

$$y_3 = -c/b \quad (4.15)$$

$$\Delta = (2 * x_2)^2 - a * (x_2^2 + (y_2^2 - y_3^2)^2 - D^2) \quad (4.16)$$

$$x_{3_1} = (2 * x_2 - \sqrt{\Delta})/2 \quad (4.17)$$

$$x_{3_2} = (2 * x_2 + \sqrt{\Delta})/2 \quad (4.18)$$

$$x_3 = x_2 \text{ et } y_3 = -c/b \quad (4.19)$$

Les coordonnées désirées étant maintenant fixées quel que soit les paramètres de la route initiale, le calcul de  $\alpha$  (exprimé dans un repère de navigation) s'exécute de la manière suivante :

$$\alpha = -\arctan\left(\frac{y_3 - y_2}{x_3 - x_2}\right) + 90 \quad (4.20)$$

Et la correction  $C_{corr}$  demandée au drone est donc de :

$$C_{corr} = \alpha - \beta \quad (4.21)$$

La théorie étant maintenant établie, il nous fallut tester nos algorithmes sur Matlab en simulant différents bruits sur le drone pour vérifier la robustesse.

### 4.3.1.2 Simulateur Matlab

Le but du simulateur de trajectoire sous Matlab est principalement de recréer les contraintes extérieures que le drone va subir, nous avons simulé 3 types de bruits pouvant intervenir :

- Des aléas de direction et d'avancement;
- Un décalage de magnétomètre;
- Du vent.

La figure 4.13 ci-dessous illustre physiquement ces différents bruits cités.

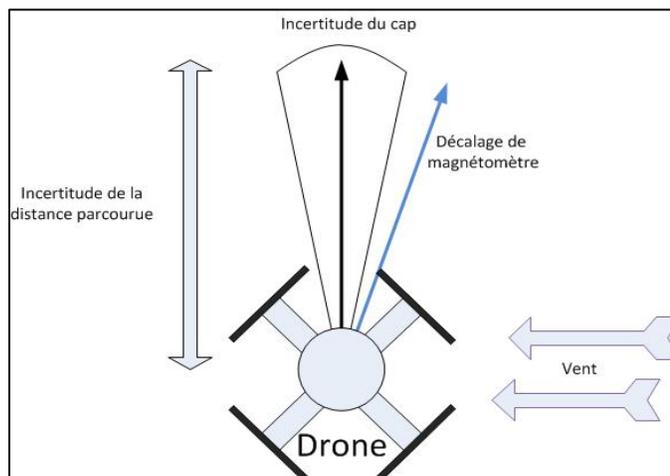


Figure 4.13 Bruits et contraintes externes agissant sur le drone et ses capteurs

Les aléas de direction et d'avancement se modélisent par une incertitude de quelques degrés sur la direction dans laquelle se dirige le drone et la distance sur laquelle il avance. Pour les modéliser, à chaque itération (qui est sensé simuler une seconde de navigation), nous introduisons des facteurs aléatoires de  $10^\circ$  sur la direction et 10% de la vitesse maximale du drone.

Pour le décalage du magnétomètre, il suffit d'intégrer en permanence un terme constant sur le cap après chaque itération d'avancement du drone.

Enfin pour le vent, il faut intégrer une translation dans le déplacement du drone indépendante de la vitesse et du cap du drone. Également pour plus de réalisme, nous avons généré des facteurs aléatoires de  $20^\circ$  sur le cap du vent et de 50% sur sa vitesse.

Les figures 4.14 et 4.15 ci-dessous montrent les réponses de notre algorithme aux différents types de bruits.

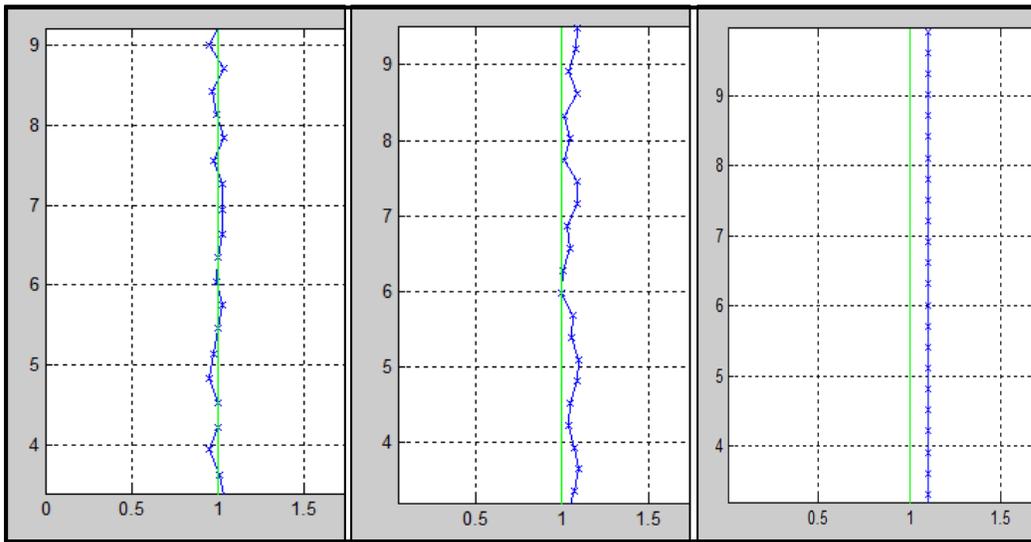


Figure 4.14 Bruits dus aux aléas, au vent et au décalage du magnétomètre respectivement

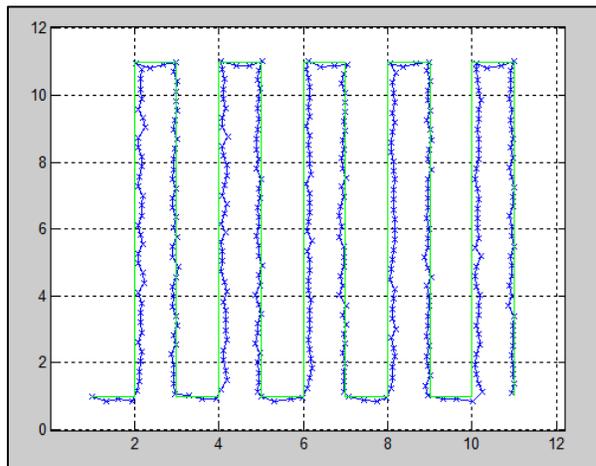


Figure 4.15 Exemple de navigation de base avec tous les bruits cumulés

En observant attentivement les résultats obtenus après simulations, il est possible de déterminer l'impact de chaque type de bruit et de déterminer la gravité de leurs perturbations. La première figure montre les corrections dues à des facteurs aléatoires de petites envergures, ce qui arrive souvent dans une masse d'air un peu perturbée. Comme prévu l'algorithme est capable de corriger ces erreurs et continu à faire avancer le drone sans problèmes particuliers. Les deux autres types de bruits dus au vent et à un décalage magnétométrique sont eux beaucoup plus gênant car ils génèrent des erreurs permanentes de positionnement. Ces décalages sont relativement gênants lors d'une navigation par points de cheminement pour les raisons expliquées plus haut mais restent acceptables car non destructif pour mener à bien la mission.

Bien que satisfaisant au premier abord, il nous fallut encore modifier l'algorithme pour déterminer les commandes à envoyer sur les 3 axes (avant, coté, rotation) pour piloter le contrôleur de vol du drone. Également, l'erreur due au décalage du magnétomètre étant fixe, nous avons essayé de proposer une solution pour la détecter et la compenser. Pour résoudre ce problème, nous avons décidé d'implémenter un facteur correctif dont le but est de minimiser des décalages permanents.

La difficulté pour replacer le drone sur sa trajectoire est de piloter simultanément les 3 axes. L'UAV étant en constante rotation pour replacer l'avant dans le sens de la marche, les deux autres axes doivent changer en permanence pour avoir une trajectoire fixe par rapport au sol alors que le repère du drone change constamment due à sa rotation. Pour résoudre ce problème, nous nous sommes servi de matrices DCM décrites dans le chapitre 2.2.1.3 de ce document. La figure 4.16 montre comment nous avons réussi à simuler un redressement de  $60^\circ$  tout en translatant le drone en ligne droite.

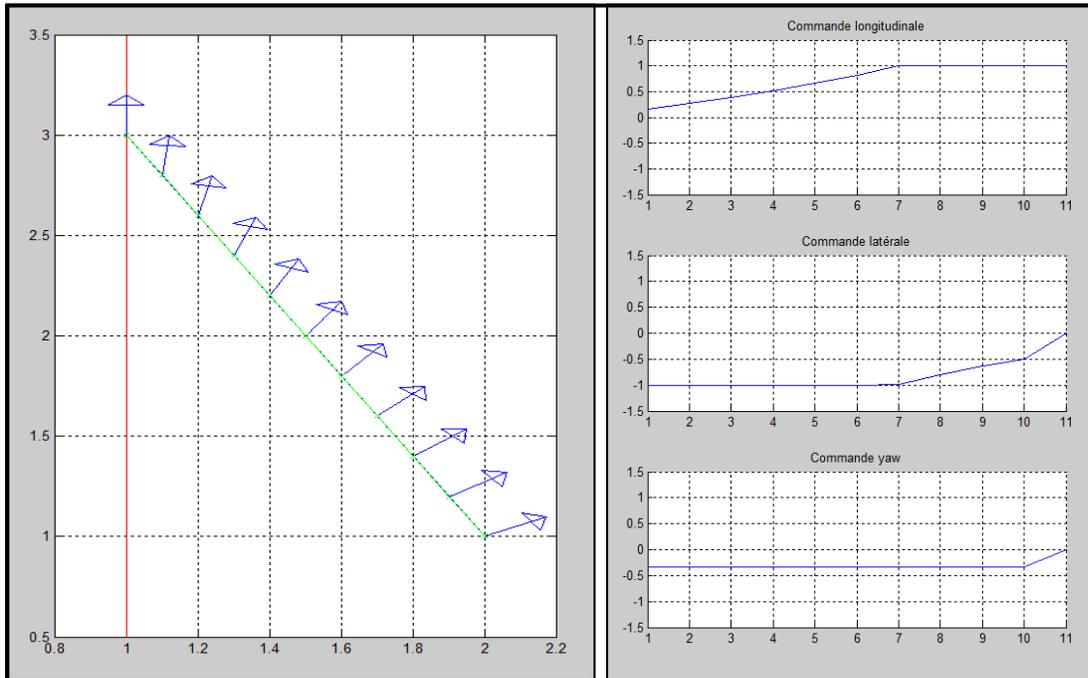


Figure 4.16 Exemple de commandes synchronisées sur 3 axes

Le principe de correction est simple, il part du postulat que notre drone possède des mêmes caractéristiques sur l'axe latéral et longitudinal puisqu'il est symétrique diagonalement au niveau de ses moteurs. Aussi le contrôleur de vol bride ses angles d'Euler maximaux, sa vitesse absolue est donc toujours la même. Donc pour aller vers l'avant à plus ou moins  $45^\circ$ , le drone donne toute sa puissance sur l'axe de l'élévateur et ajuste celui de l'aileron pour modifier l'angle. Au-delà de plus ou moins  $45^\circ$ , c'est l'aileron qui est au maximum et l'élévateur qui ajuste sa puissance pour modifier l'angle. Les graphiques ci-dessus illustrent parfaitement cet exemple.

Partant d'une base de correction toutes les 100 ms et un relevé de position toutes les secondes, il faut calculer les 10 prochaines corrections à chaque itération. Nous avons donc joint les deux codes précédents pour tester la robustesse de notre algorithme.

La figure 4.17 montre en bleu la trajectoire et le cap suivit par le drone et en noir les corrections demandées à chaque itération.

L'exemple ci-dessous montre un décalage de capteur de  $20^\circ$  et un vent venant de la gauche avec une vitesse égale à un tiers de celle du drone.

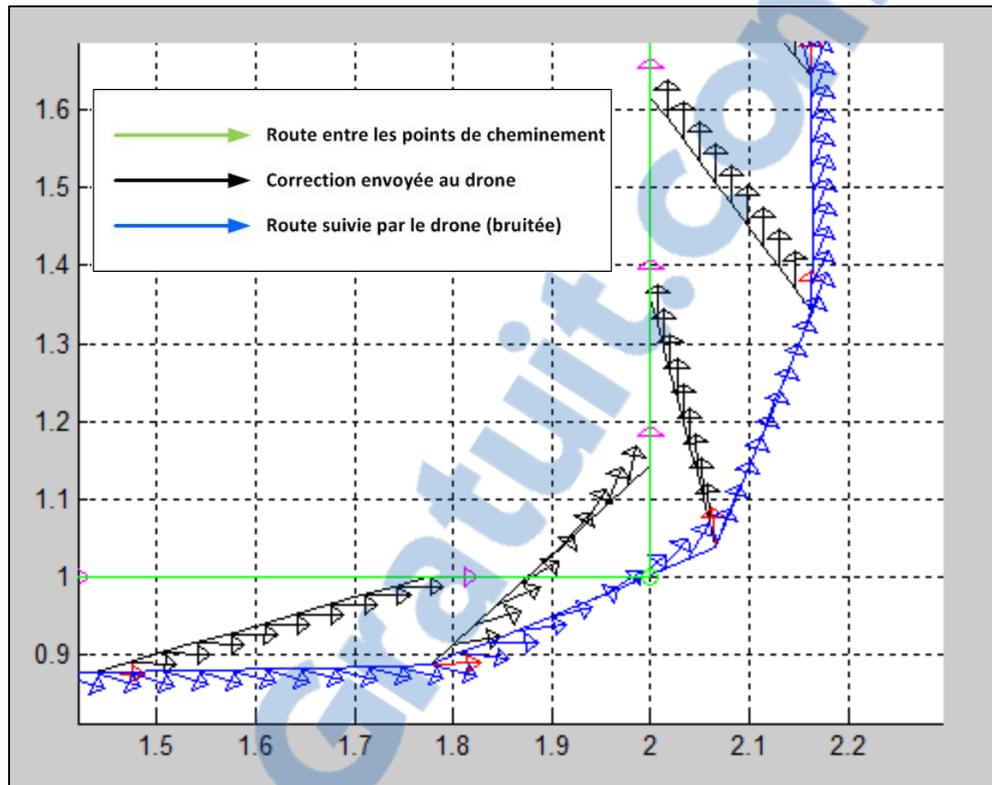


Figure 4.17 Exemple de correction de navigation avec pilotage des 3 axes

Le décalage par rapport à la trajectoire due à une erreur de calibration de capteur magnétométrique étant très important et causant beaucoup de vols non symétriques, nous avons essayé de calculer un terme correctif pour le compenser. Le principal problème est qu'il est difficile de faire la différence entre une erreur de trajectoire due à un décalage de capteur et une erreur due au vent. Le terme correctif doit donc évoluer avec le temps pour ne pas confondre ces deux erreurs. Pour estimer une correction viable, nous avons implémenté un terme évolutif qui change le dixième de sa valeur à chaque itération. La valeur corrective est calculée en fonction de la distance perpendiculaire à la route à suivre (Cf. Figure 4.12) et essaie de la réduire en changeant un dixième de sa valeur. La figure 4.18 ci-dessous montre les améliorations apportées par le terme correctif, les conditions de bruitage sont les mêmes avant et après l'application de la correction.

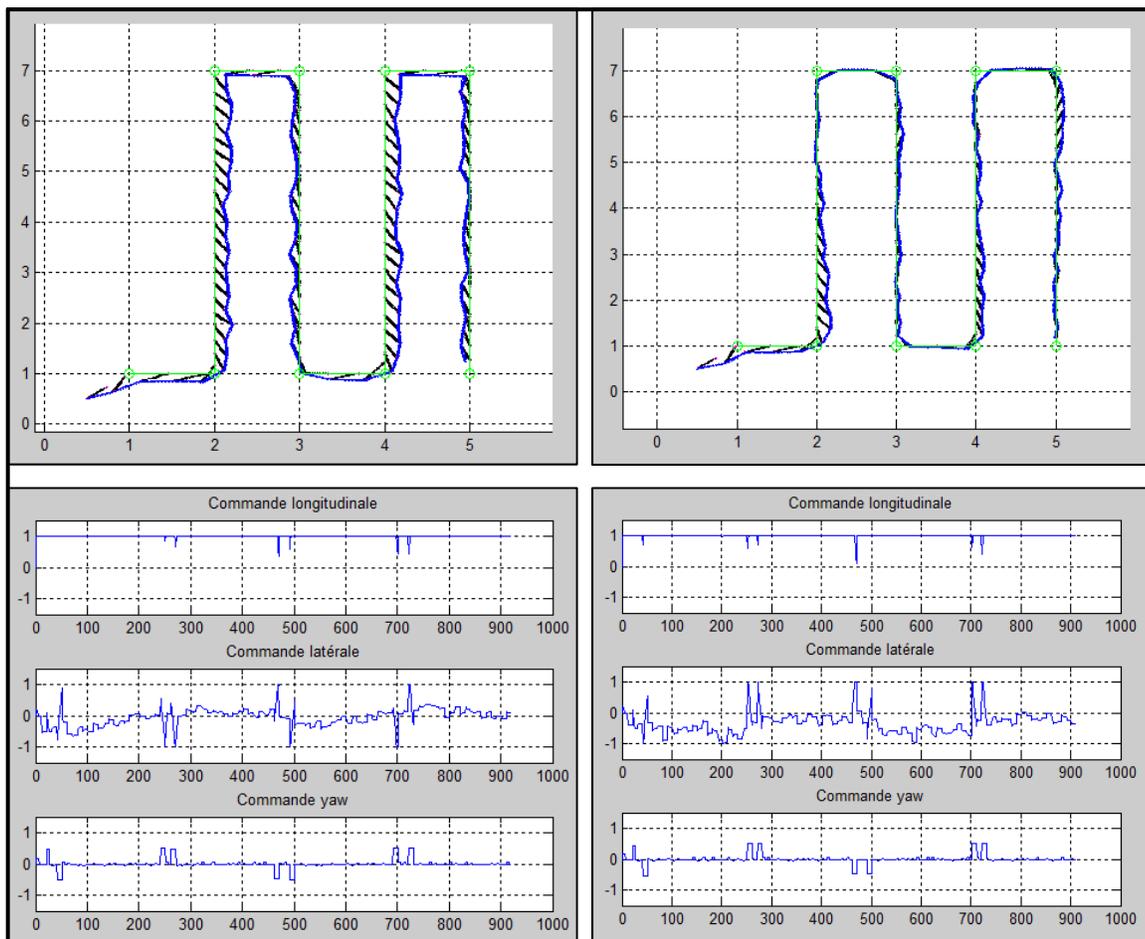


Figure 4.18 Trajectoires avant et après correction de la dérive des capteurs

Visuellement, on constate immédiatement une nette amélioration sur la réduction du décalage permanent après application du terme correctif. Sur ce trajet, la correction de cap détectée et appliquée par l'algorithme de navigation est de  $19^\circ$  en moyenne alors que le décalage de cap réel est de  $20^\circ$ . Ce résultat est très acceptable car il est fortement perturbé par les autres bruits entrant en jeu lors des simulations (aléas et vent). Après analyse des résultats, le terme correctif produit une réduction de l'erreur de positionnement moyenne d'environ 40%.

En regardant de plus près, on constate également une réduction des corrections appliquées sur la commande latérale du drone. Sur ce trajet, on relève une utilisation la commande latérale réduite d'environ 46%. Cette importante réduction apporte une amélioration significative sur la symétrie de vol du drone (réduction de l'angle de glissement). Ce qui est

très appréciable car la caméra fixée sur le nez du drone se trouve alors mieux alignée avec la route suivie.

Fait intéressant, la correction de cap appliquée ne réduit pas le temps de navigation (en moyenne 95 itérations pour les deux simulations), seule la précision est améliorée, pas l'efficacité de l'algorithme.

La dernière étape pour adapter l'algorithme au drone fut de transformer les coordonnées du capteur GPS du drone (en latitude, longitude et élévation) dans le repère local (ENU) utilisé plus haut.

Le chapitre 2 explique dans le paragraphe 2.2.1.1 comment passer d'un repère cartésien à géodésique mais pas l'inverse. À partir de coordonnées géodésique  $(\phi, \lambda, h)$ , les formules pour passer dans le repère cartésien sont :

$$N = \frac{a}{\sqrt{1 - e^2 * (\sin(\phi))^2}} \quad (4.22)$$

$$\begin{bmatrix} x \\ y \\ z \end{bmatrix}_{ECEF} = \begin{bmatrix} (N + h) * \cos(\phi) * \cos(\lambda) \\ (N + h) * \cos(\phi) * \sin(\lambda) \\ (N * (1 - e^2) + h) * \sin(\phi) \end{bmatrix}_{ECEF} \quad (4.23)$$

Le repère utilisé avec Matlab étant un repère ENU. Il nous fallut également calculer la matrice de rotation pour passer d'un repère ECEF à ENU. Or, par propriété des matrices de rotation :

$$C_E^N = C_N^E^{-1} = C_N^E{}^T \quad (4.24)$$

Nous avons donc pu utiliser les formules (2.12) et (2.13) du chapitre 2 pour obtenir les bonnes matrices de transformation.

Pour effectuer les calculs dans un seul plan, il est nécessaire de choisir un point de référence pour fixer un ENU. Dans notre cas, la première position du drone fut ce point, il suffit ensuite d'appliquer la formule suivante à chaque relevé du capteur pour travailler dans le repère local de notre algorithme.

$$\begin{bmatrix} x_{UAV} \\ y_{UAV} \\ z_{UAV} \end{bmatrix}_{ENU} = \begin{bmatrix} -\sin(\lambda_{ref}) & -\sin(\phi_{ref}) * \cos(\lambda_{ref}) & \cos(\phi_{ref}) * \cos(\lambda_{ref}) \\ \cos(\lambda_{ref}) & -\sin(\phi_{ref}) * \cos(\lambda_{ref}) & \cos(\phi_{ref}) * \sin(\lambda_{ref}) \\ 0 & \cos(\phi_{ref}) & \sin(\phi_{ref}) \end{bmatrix}^T \quad (4.25)$$

$$* \left( \begin{bmatrix} (N + h_{UAV}) * \cos(\phi_{UAV}) * \cos(\lambda_{UAV}) \\ (N + h_{UAV}) * \cos(\phi_{UAV}) * \sin(\lambda_{UAV}) \\ (N * (1 - e^2) + h_{UAV}) * \sin(\lambda_{UAV}) \end{bmatrix}_{ECEF} - \begin{bmatrix} (N + h_{ref}) * \cos(\phi_{ref}) * \cos(\lambda_{ref}) \\ (N + h_{ref}) * \cos(\phi_{ref}) * \sin(\lambda_{ref}) \\ (N * (1 - e^2) + h_{ref}) * \sin(\lambda_{ref}) \end{bmatrix}_{ECEF} \right)$$

Nous avons maintenant un algorithme capable de gérer automatiquement la trajectoire du drone avec une assez grande précision et générer les commandes de vol. Il nous fallut donc asservir les commandes de vol avant de pouvoir les implémenter sur le drone.

### 4.3.2 Asservissement des commandes

Afin de piloter l'altitude et le positionnement du drone de manière automatique, il est nécessaire d'asservir les commandes afin qu'elles soient stables et le plus rapide possible. Dans cette optique, il nous fallut réfléchir au design d'une boucle d'asservissement afin de modéliser le drone et calculer des correcteurs.

D'un point de vu général, les boucles d'asservissement peuvent être représentées par la figure 4.19 ci-dessous :

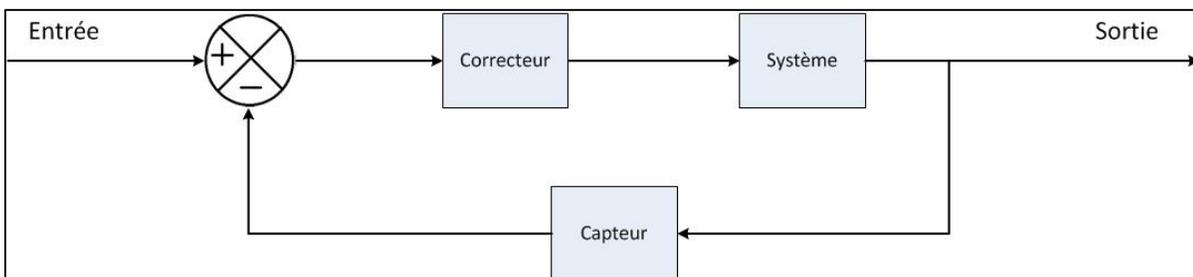


Figure 4.19 Boucle d'asservissement classique

Elles permettent de prédire la sortie en fonction de l'entrée et ainsi déterminer et corriger le comportement d'un système asservis avec un capteur (Bensoussan, 2008).

Dans le cas de notre drone, il existe quatre boucles d'asservissement, une sur chacun des axes (aileron, élévateur, puissance et lacet), mais leur représentation est légèrement différente due à la présence du contrôleur de vol qui dispose lui-même de sa propre boucle interne (non représentée sur la figure 4.20).

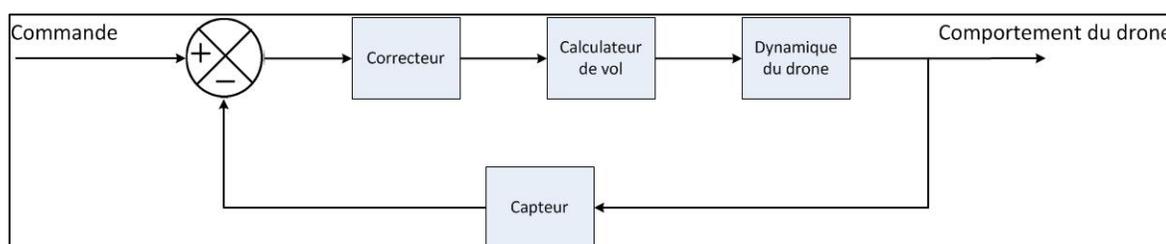


Figure 4.20 Boucle d'asservissement simplifiée du drone

#### 4.3.2.1 Modélisation du drone

Avant de calculer un correcteur dans une boucle d'asservissement, la première étape est de déterminer la dynamique du système asservis. Aussi, nous avons dû déterminer la dynamique de notre drone pour calculer un correcteur sur chacun des axes.

Il existe de nombreuses publications expliquant la modélisation du comportement des drones quadripales en fonction de leurs poids, moteur et géométrie, Pounds (p3-p7, 2005) et Waslander (p 469, 2005) offrent des modélisations très complètes de drones, par exemple.

Leishman (p 33, 2006) propose également des modélisations dynamique pour déterminer la réponse des hélices en fonction de leur profil.

Mais l'expérience montre que bien souvent les modélisations sont simplifiées et assez éloignées de la réalité, nous avons donc pris la décision de ne pas les suivre. De plus dans notre cas, nous n'avons aucune idée de l'asservissement interne du calculateur de vol, ces modélisations auraient donc été inutiles car partielles.

La manière la plus simple et la plus juste pour déterminer un correcteur était de qualifier la dynamique de vol du drone par des essais en vol. Grâce à des relevés capteurs, nous avons essayé d'obtenir une approximation des fonctions de transfert du drone par des filtres passe-bas du second ordre. Le but étant de générer des échelons sur les commandes de vol pour déterminer les coefficients du filtre en fonction de la réponse dynamique.

Le paragraphe suivant prend pour exemple le calcul de la dynamique verticale du drone en fonction de la puissance donnée aux moteurs.

Pour réaliser ces essais, nous sommes allés en cage de vol et avons essayé de stabiliser manuellement l'altitude du drone. Lorsqu'il était stable, nous avons incrémenté brutalement une augmentation des gaz pour obtenir un échelon et avons relevé les altitudes du drone.

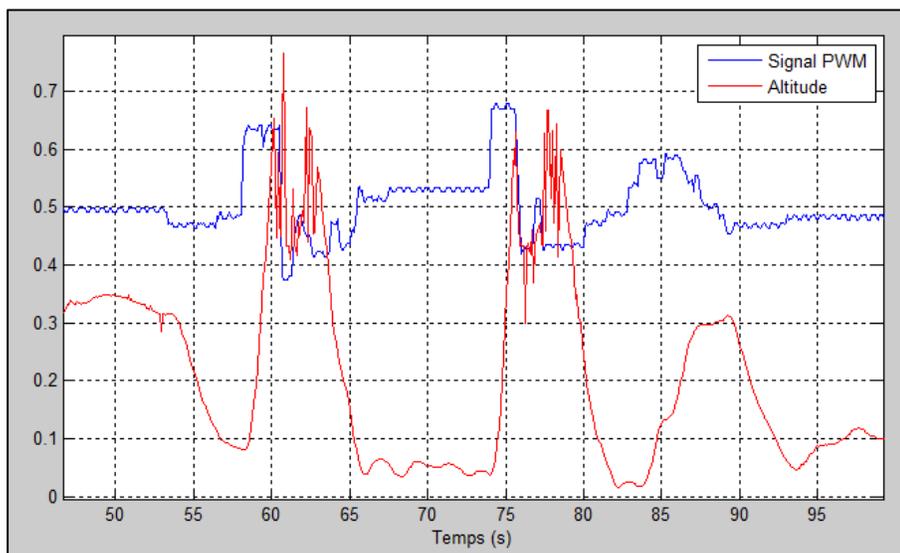


Figure 4.21 Qualification du drone en cage de vol.

La figure 4.21 ci-dessous présente l'altitude du drone (en rouge) en fonction des commandes PWM envoyées au calculateur de vol (en bleu) après mise à l'échelle. Bien que les mesures soient perturbées dans les hautes altitudes, elles permettent de déterminer la réponse du drone en fonction des commandes envoyées. À ceci près que les commandes envoyées sont des vitesses et le résultat une position, une dérivation de l'altitude fut donc nécessaire pour

déterminer la vitesse verticale du drone et approximer la dynamique par un filtre passe-bas. Cet asservissement est illustré dans la figure 4.22 ci-dessous.

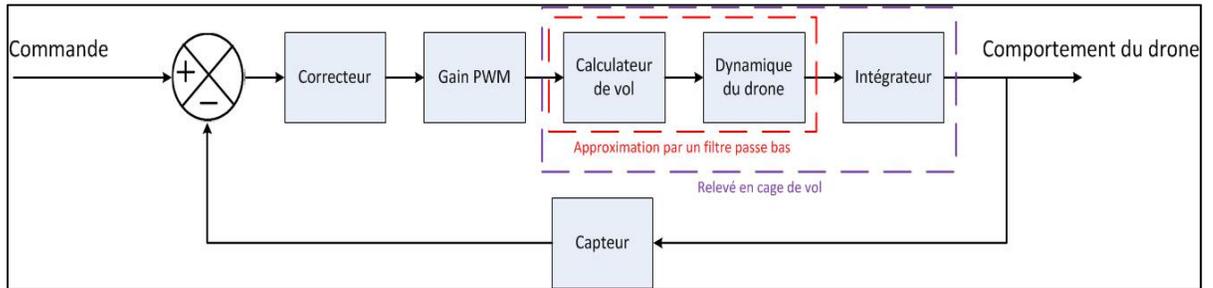


Figure 4.22 Asservissement de l'axe vertical

Après intégration lissage et mise à l'échelle de l'altitude, on obtient une courbe s'approchant de la forme d'un filtre passe-bas. La figure 4.23 ci-dessous montre en vert et rouge la position du drone avant et après lissage, en violet et noir la vitesse verticale du drone avant et après lissage et enfin en bleu la commande en PWM.

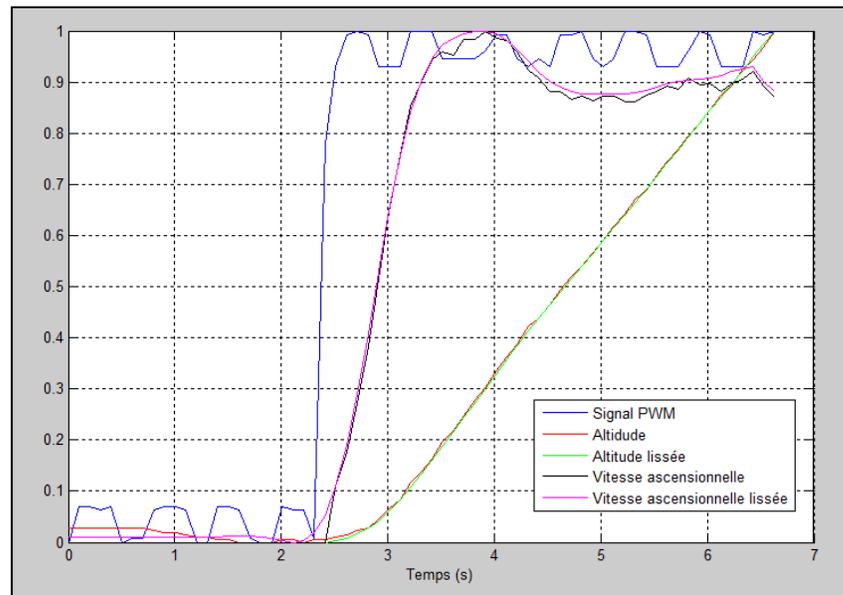


Figure 4.23 Dérivation et lissage de la position du drone en fonction des commandes

La période et le dépassement nous permirent de déterminer un filtre passe-bas qui ajouté à un intégrateur nous donna approximative de la dynamique réelle du drone dans le domaine de Laplace.

$$Dyn = \frac{1}{s(0.1494s^2 + 0.4224s + 1)} \quad (4.26)$$

Le seul moyen de vérifier la véracité de la dynamique estimée du drone est d'implémenter un correcteur et de faire de nouveaux tests en vol. Ces étapes seront développées dans les chapitre 4.3.2.2 et 6.3.

#### 4.3.2.2 Correction par PID

Il existe de nombreux types de correcteurs pour les drones quadripales, linéaires (PID, LQR,  $H_\infty$ , etc.) ou non linéaires (Mode Glissant, linéarisation entrées-sorties, control géométrique, Backstepping, etc.) et de complexités variées (Charland, 2014). Mais compte tenu de l'aspect prototypique et peu précis de notre asservissement, nous avons décidé de prendre le plus flexible et le plus simple : le PID. Le PID est un correcteur Proportionnel Intégrateur Dérivateur, il ajoute des gains à l'intégrale, à la dérivée et la valeur de la consigne d'asservissement elle-même. Bresciani (p 38, 2008) et Kim (p 3, 2003) ont utilisé eux aussi des PIDs pour contrôler leurs drones durant leur recherches, donnant des résultats relativement corrects.

Pour calculer le PID, nous avons simulé l'asservissement sous Simulink et utilisé l'utilitaire de calcul de correcteurs. Les gains ont été choisis pour privilégier la stabilité et le non dépassement de la valeur finale. Bien que la correction finale ne soit pas parfaite, elle permet néanmoins d'avoir une réponse stable avec un temps de réponse de 2 secondes, avec de faibles oscillations comme le montre la figure 4.24.

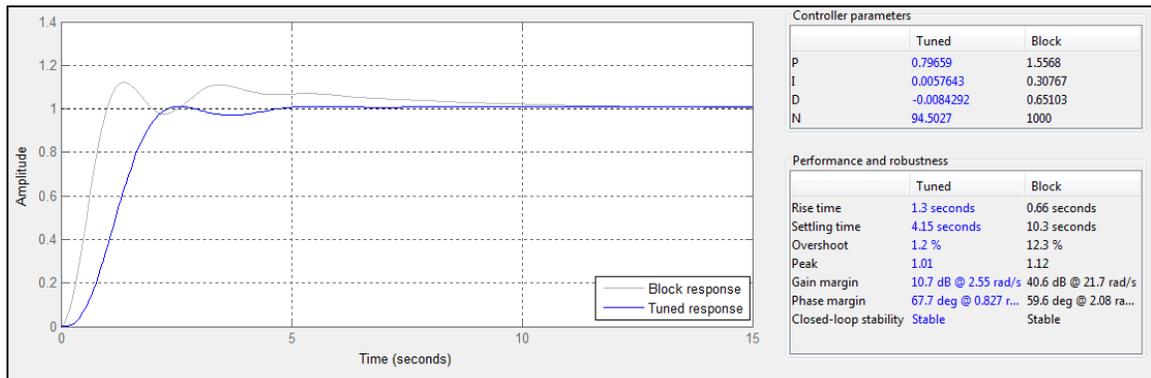


Figure 4.24 Correction par PID

Les gains ont ensuite pu être implémentés en python dans la Raspberry pour les tests en vol sur la stabilité verticale du drone (chapitre 6).

### 4.3.3 Détections de collision

Pour éviter les collisions entre les drones en vol, il nous fallut mettre au point un algorithme de détection de collisions. L'idée principale est de relever la position et les vitesses des drones pour calculer la distance minimum entre les deux trajectoires. Si les trajectoires sont en collisions ou trop proches les unes des autres, des dispositions seront prises pour éviter les abordages.

Pour déterminer la distance entre les deux trajectoires, il est nécessaire de savoir calculer la distance minimale entre deux segments dans l'espace. Les segments peuvent être représentés par des fonctions linéaires bornées sur 3 dimensions dont les paramètres dépendent de la position du drone et de sa vitesse. La formule de distance entre deux points  $(x1, y1, z1)$  et  $(x2, y2, z2)$  est donnée par la formule :

$$D = \sqrt{(x1 - x2)^2 + (y1 - y2)^2 + (z1 - z2)^2} \quad (4.27)$$

Dans notre cas, il suffit de considérer les points comme paramétrables en fonction de deux paramètres indépendants l'un de l'autre. Nos points deviennent  $(x1(s), y1(s), z1(s))$  et  $(x2(t), y2(t), z2(t))$  et il suffit maintenant de minimiser la fonction à deux paramètres  $f$  :

$$f(s, t) = (x1(s) - x2(t))^2 + (y1(s) - y2(t))^2 + (z1(s) - z2(t))^2 \quad (4.28)$$

Les deux trajectoires sont considérées comme des droites dans l'espace. Sauf si elles sont parallèles, il n'existe qu'un seul endroit où leur distance est minimum. À cette combinaison de points  $s$  et  $t$ , les dérivées partielles de la fonction  $f$  sont égales à zéro. On notera les coordonnées des points des droites  $(x1+a1*s, y1+b1*s, z1+c1*s)$  et  $(x2+a2*s, y2+b2*s, z2+c2*s)$ .

$$\begin{aligned} \frac{\partial f}{\partial s} = 2 * s * (a1^2 + b1^2 + c1^2) - 2 * t * (a1 * a2 + b1 * b2 + c1 * c2) - 2 \\ * (a1 * (x2 - x1) + b1 * (y2 - y1) + c1 * (z2 - z1)) \end{aligned} \quad (4.29)$$

$$\begin{aligned} \frac{\partial f}{\partial t} = 2 * t * (a2^2 + b2^2 + c2^2) - 2 * s * (a1 * a2 + b1 * b2 + c1 * c2) - 2 \\ * (a2 * (x1 - x2) + b2 * (y1 - y2) + c2 * (z1 - z2)) \end{aligned} \quad (4.30)$$

En résolvant le système suivant, on détermine la paire de coordonnées  $s$  et  $t$  où les droites sont le plus proche l'une de l'autre car la stabilisation des dérivées indique le minimum de la fonction  $f$  en fonction d'un de ses paramètres.

$$\begin{cases} \frac{\partial f}{\partial s} = 0 = d1 * s + e1 * t + g1 \\ \frac{\partial f}{\partial t} = 0 = d2 * t + e2 * s + g2 \end{cases} \quad (4.31)$$

Soit :

$$\begin{cases} t = \frac{-g1 + \frac{g2}{e2} d1}{e1 - \frac{d2}{a2} d1} \\ s = \frac{-d2 * t - g2}{e2} \end{cases} \quad (4.32)$$

En identifiant les coefficients  $d1, d2, e1, e2, g1, g2$  avec les formules (4.29) et (4.30).

$$\begin{aligned}
 d1 &= (a1^2 + b1^2 + c1^2) \\
 e1 &= -(a1 * a2 + b1 * b2 + c1 * c2) \\
 g1 &= -(a1 * (x2 - x1) + b1 * (y2 - y1) + c1 * (z2 - z1))
 \end{aligned} \tag{4.33}$$

$$\begin{aligned}
 d2 &= (a2^2 + b2^2 + c2^2) \\
 e2 &= -(a1 * a2 + b1 * b2 + c1 * c2) \\
 g2 &= (a2 * (x2 - x1) + b2 * (y2 - y1) + c2 * (z2 - z1))
 \end{aligned} \tag{4.34}$$

Si les valeurs de s et de t sont en dehors des limites qui définissent nos segments, il faut alors borner les termes et considérer à nouveau l'équation (4.28) mais avec un seul paramètre (s si t est borné et t si s est borné). En dérivant simplement cette fonction, on trouve le minimum de distance entre une droite et un point (une seule solution possible, le passage par (4.31) et (4.32) n'est alors plus nécessaire).

Grâce à ces calculs, on obtient finalement le couple (s, t) qui, une fois réintroduit dans la formule (4.27), donne la distance minimale entre les deux drones lors de leurs prochaines trajectoires. Pour éviter la collision, il suffit simplement d'immobiliser l'un des deux UAV quelques instants. La figure 4.25 donne un exemple de détection de collision entre deux trajectoires (en bleu et en vert). La distance minimale calculée est dessinée en rouge.

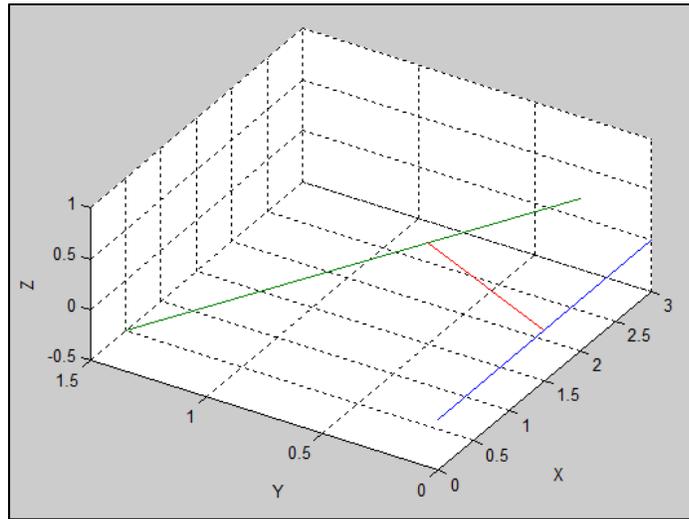


Figure 4.25 Exemple de détection de distance minimale entre deux trajectoires. (Trajectoires en bleu et vert et distance minimale en rouge)

#### 4.3.4 Architecture logicielle du drone

Pour permettre des vols en toute sécurité, il est nécessaire de bâtir une architecture logicielle qui ne laisse place à aucune incertitude lorsque le drone est dans les airs. La principale crainte étant de perdre le contact avec le sol.

Aussi, nous nous sommes efforcés de créer une architecture qui optimise la liaison avec le sol, pour cela nous avons décidé d'innover en matière d'UAV et de créer un système de réseau de partage de commandes entre drones.

##### 4.3.4.1 Partage des commandes

Pour permettre de faire des réseaux de drones comme expliqué dans le paragraphe 4.2.8, il nous fallut mettre en place une structure de communication partageant les informations importantes entre drone.

L'idée est de fonctionner un peu comme la structure d'une suite d'avioniques (plus d'explications dans le chapitre 5) : chaque individu connaît la position et le dernier ordre

envoyé aux autres individus. Connaître la position des autres drones permet d'éviter les collisions en stoppant sa propre trajectoire et connaître les derniers ordres envoyés aux autres drones permet de faire des relais d'informations si tout le monde n'est pas à portée des infrastructures de contrôle.

Notre réseau est bâti sur le modèle suivant : chaque individu émet à courte portée sa propre position et les derniers messages qu'il a reçus en provenance du sol (LTE ou satellite) ou d'autres drones (RF 2.4 GHz) avec l'heure à laquelle ils ont été émis.

Lors de la réception de messages en provenance du sol ou des autres drones, chaque UAV choisit les informations les plus récentes et les met à jour dans sa propre base de données avant de les transmettre à nouveau aux autres drones. Ainsi lorsqu'un ordre arrive du sol sur un des drones en l'air, il finit forcément par atteindre le drone destinataire par ricochets. Cette méthode augmente le trafic de données mais a l'avantage de pouvoir continuer à garder le contact avec un drone s'il devient hors de portée satellite ou LTE ou si une partie de ses équipements de communication tombe en panne.

#### 4.3.4.2 Structure du logiciel

La Raspberry étant une plateforme capable de traiter plusieurs applications simultanément, nous avons choisi de diviser les différentes fonctionnalités du drone en 4 applications qui partagent des variables communes (figure 4.26). L'utilité de séparer ces applications est double :

- La séparation permet de mieux distinguer les différentes fonctions et faciliter d'éventuelles modifications.
- Les différents périphériques du drone (LTE, capteurs, PWM, etc.) ne fonctionnent pas tous à la même fréquence. Les séparer dans des boucles principales indépendantes les unes des autres facilite grandement la programmation et évite d'éventuelles interactions qui pourraient être néfastes pour le système.

Les quatre applications sont divisées de la manière suivante :

1. Émission des messages LTE;

2. Collecte des données des capteurs GPS, inertiel et LIDAR;
3. Réception des messages LTE, réception, tri et émission des messages RF 2.4 GHz;
4. Modes et pilotages, calculs de position et algorithme de navigation.

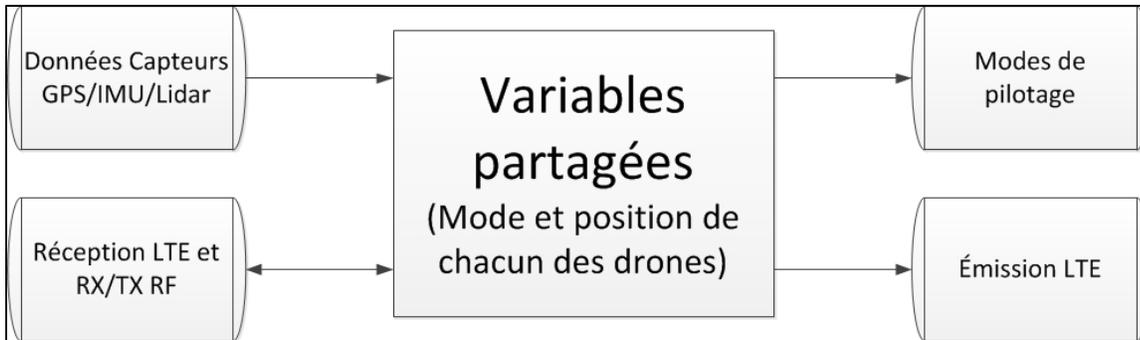


Figure 4.26 Architecture du partage de données entre applications

Les trois premières fonctions sont relativement simples :

- La première application gère l'émission de la LTE (ou satellite au besoin) en piochant dans les variables partagées et les communique au centre de contrôle.
- La seconde application est dédiée aux données des capteurs, elle traite et transforme les signaux émis par le GPS, l'IMU et le lidar à différentes fréquences pour fournir une position précise du drone qui sera ensuite utilisée par les autres applications.
- La troisième application s'occupe de la gestion RF et de la réception LTE. Elle relève les différents messages émis par les différentes sources (LTE et RF), garde les informations les plus récentes en mémoire, les émet en RF et réécrit le tout dans les variables partagées. De cette manière, le drone prend les informations qui lui sont nécessaires et fait office de relais d'informations pour les autres drones.

Enfin, la dernière application s'occupe de la gestion des modes de pilotages. Elle est un peu plus complexe que les premières, elle doit être la plus simple possible et toujours privilégier

la sécurité. Le système proposé comporte des modes de pilotage (de 1 à 6), qui correspondent chacun à un état de commande du drone.

1. Le premier mode est le mode manuel, le drone reçoit les commandes directement depuis le sol et ne fait pas de traitements de données.
2. Le mode 2 est un mode de stabilisation verticale, il maintient le drone à sa position et altitude actuelle. Il permet également le décollage et l'atterrissage automatique du drone.
3. Le mode 3 est un état de transition : avant de lancer une navigation autonome, il est nécessaire de s'assurer que les points de cheminement réceptionnés soient les bons et qu'il n'y aura pas de fausses trajectoires. Pour cette raison, lors des réceptions de données, le drone se stabilise à son altitude et confirme par envoi de données à son tour que les points de cheminement ont été correctement réceptionnés. Seulement après cela, la navigation automatique sera autorisée dans le mode 4.
4. Le mode 4 est la navigation autonome du drone par points de cheminement avec l'algorithme du paragraphe 4.3.1.
5. Le mode 5 est une demande de GoHome faite par l'utilisateur.
6. Le mode 6 coupe les moteurs (fin de mission ou sécurité en cas de danger).

Le principe général de la boucle principale est lui aussi assez simple, toutes les secondes au minimum, le drone vérifie qu'il reçoit des données via LTE ou RF, si ce n'est pas le cas, il se stabilise à l'altitude où il se trouve pendant 3 secondes pour réessayer d'établir à nouveau une connexion. S'il n'y arrive toujours pas, il procède à un GoHome. Également à chaque itération, le drone vérifie s'il ne risque pas d'entrer en collision avec les autres drones, le cas échéant, il se stabilise pour laisser passer l'autre drone. Cette boucle est effectuée en permanence pour toujours garder le contrôle depuis le sol et ce pour tous les modes de commande (autonome et manuel).

Toutes les applications décrites sont disponibles sous formes de schémas blocs dans l'annexe VI.

L'architecture du logiciel présentée répond en tous points aux attentes que nous avons formulées pour piloter le drone. Elle ne dispose pas en théorie de failles et permet une grande modularité pour les prochains étudiants travaillant sur ce projet. La séparation des modes et des applications a permis de simplifier la boucle principale et gérer plus de fonctions de manière plus optimale. L'architecture dispose également d'un avantage non négligeable : elle permet l'intégration d'une flotte de drones en détectant les collisions et en agissant comme un relais d'information.

## CHAPITRE 5

### UTILISATION D'AVIONIQUES G1000

L'un des objectifs de ce mémoire était de transmettre les différents éléments de vol du drone vers des instruments d'avion ou plus communément appelés avioniques. Au cours de ce chapitre, nous verrons comment nous avons réussi à interfacer physiquement et de façon logicielle deux écrans de la suite G1000 avec le reste de notre système. L'idée derrière cette partie des travaux de recherche est d'ouvrir la porte à deux types de nouvelles applications, soient l'utilisation de drones faibles coûts de façon sécuritaire en exploitant les technologies certifiées de l'avionique moderne et le pilotage d'avion à distance en réduisant la charge sur les pilotes. La finalité de cette partie du projet étant de faciliter le passage des avions à 2 pilotes à 1 pilote et les avions à 1 pilote à 0 pilote, soit un UAV.

#### 5.1 Introduction au système Garmin 1000

Le Garmin 1000 ou plus communément appelé G1000 est une suite d'avioniques très répandue dans le domaine de l'aviation légère et moyenne. Il équipe de nombreux avions comme les Cirrus SR22, Beechcraft Barons, Cessna Citation Mustang et bien d'autres. Avec les nombreux éléments qui le composent, le G1000 est très modulable et permet de nombreuses configurations différentes, également son prix relativement abordable dans le monde des avioniques en font l'un des instruments de navigation le plus utilisé.

Le G1000 est principalement composé de 2 ou 3 écrans LCD appelés GDU (Garmin Display Unit) disposés en 1 ou 2 PFD (Primary Flight Display) et un MFD (Multi-Function Display). On retrouve également dans les postes de pilotage 1 ou 2 GMA (contrôleur audio) et accessoirement un GMC et un GCU (claviers et AP déportés).

Derrière le tableau de bord se trouvent également tous les capteurs et composants qui participent au fonctionnement du système. Ces éléments peuvent être en simple ou en double en fonction des besoins de l'avion (privé ou transport de passagers, turbine ou pistons, monomoteur ou multi-moteurs, etc.). On retrouve parmi eux principalement le GIA

(Integrated Avionics Unit), le GDC (Air Data Computer), le GRS (Attitude and Heading Reference System (AHRS)), le GEA (Engine/Airframe Unit) et le GTX (Transpondeur).

Tous ces éléments fonctionnent ensemble sous la forme d'un réseau ou chaque système peut communiquer avec un autre même s'ils n'ont pas de lien physique direct (pas de câble de communication direct). Ainsi, si au cours d'un vol un câble de communication se brise, les deux équipements concernés peuvent continuer à communiquer entre eux en faisant un relais via les autres systèmes. Un exemple typique de cette architecture est donné en annexe VII.

Au sein d'un ensemble d'avionique Garmin, on retrouve plusieurs types de bus/protocoles de communication parmi lesquels l'ARINC 429 (protocole très répandu en aérospatial), le RS-232 (norme standard de type série), RS-485 (norme de type réseau) et surtout un bus HSDB communiquant sur un modèle de protocole Ethernet. Notre travail consistera principalement à communiquer entre un ordinateur et les GDU via ce bus HSDB.

## 5.2 Installation du G1000

Pour des raisons de coût et d'encombrement, nous avons commandé seulement 2 GDUs pour mener à bien notre projet (figure 5.1). À leur réception, les deux écrans sont arrivés brut sans support et sans connecteurs préassemblés. Il a donc fallu dans un premier temps créer un support métallique et câbler l'ensemble (communication et alimentation électrique).



Figure 5.1 GDUs tels que réceptionnés

Afin de pouvoir les utiliser, il nous a fallu créer un support métallique pour les installer et câbler l'ensemble des connecteurs spécifiques à Garmin et très mal documentés. Le détail de l'installation de ces avioniques est donné en annexe VIII.

Un aperçu du résultat final de l'installation est visible sur la figure 5.2 suivante :



Figure 5.2 Aperçu final du socle G1000

### 5.3 Débridage

Avec le brochage décrit en annexe VIII, le Garmin fonctionnait parfaitement mais uniquement en mode démonstration, c'est-à-dire que les GDUs simulaient eux même la présence des autres instruments et d'une trajectoire d'avion. Ce mode est très utile pour prendre en main le système G1000 et s'entraîner à le manipuler (aucun message d'erreur et accès illimité aux modes de réglage). Cependant, il ne permet pas de faire de l'interfaçage avec un autre système puisque dans ce mode, les GDUs simulent eux même leur propre position et ne tiennent pas compte des différents capteurs.

Pour désactiver le mode démo, le manuel d'entretien indique qu'il faut libérer la pine N°54 et ne plus la relier à la masse. Cependant aucun changement ne fut visible une fois la pine libérée, les GDUs sont restés en mode démo.

Trois solutions était alors possibles :

- Soit il s'agissait d'un problème de logiciel qui bridait les GDUs en mode démo par des lignes de code (fort probable car l'ensemble de GDU que nous avons acheté est un « Kit démo » et les compagnies disposant de logiciels propriétaires comme Garmin protègent souvent leurs produits vendus pour un but autre que commercial).
- Soit nos GDUs sont des modèles spécifiques dédiés à la démonstration et donc différents de ceux du commerce, à ce moment-là la conception matérielle de la machine elle-même est différente (peu probable car les GDUs fournis possèdent les mêmes caractéristiques matérielles que les originaux et disposent même de leur propre numéro de série).
- Soit un réglage bénin dans les modes de configuration bloquait encore nos GDUs en mode démo.

Dans les deux premiers cas, il est impossible de changer quoi que ce soit puisque nous ne disposons pas des codes sources et si nous ne disposons pas de GDUs originaux, nous n'étions certainement pas disposés à en fabriquer.

Nous nous sommes penchés donc sur la dernière option consistant à vérifier les réglages dans le mode de configuration. Malheureusement, ce fut un constant d'échec puisqu'aucune option ne mentionnait un quelconque mode démo et son activation ou non.

Cependant un détail nous interpela : dans la page de configuration des GDUs une section correspond à la vérification des programmes pine de chaque appareil. Dans le cas présent, le témoin correspondant au mode démo indiquait que la pine était reliée à la masse (voir figure 5.3 ci-dessous) alors que matériellement elle ne l'était pas.

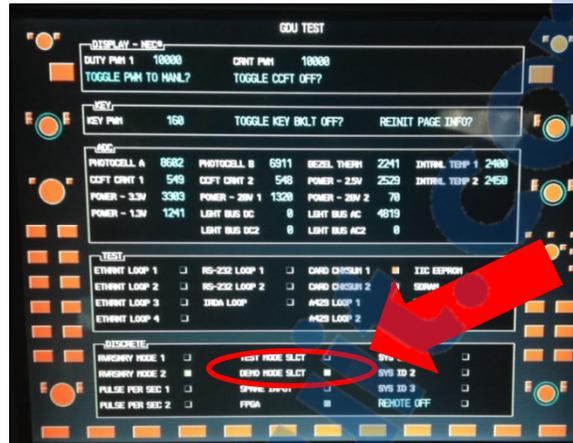


Figure 5.3 Page de config GDU TEST

Par acquis de conscience, nous avons donc vérifié que la pine n°54 était bien à la masse et qu'il ne s'agissait pas d'un codage logiciel. Grâce à un multimètre, nous avons trouvé que la pine était reliée à la masse et non libre comme elle aurait dû l'être. Nous avons donc pris la disposition d'ouvrir un Garmin pour vérifier qu'il ne s'agissait pas d'un simple pontage électronique qui reliait la broche à la masse. Effectivement, nous avons vu juste car après l'ouverture du boîtier (figure 5.4) nous avons remarqué un fil de prototypage traverser une partie du circuit imprimé (totalement interdit dans un équipement aéronautique certifié comme un G1000).

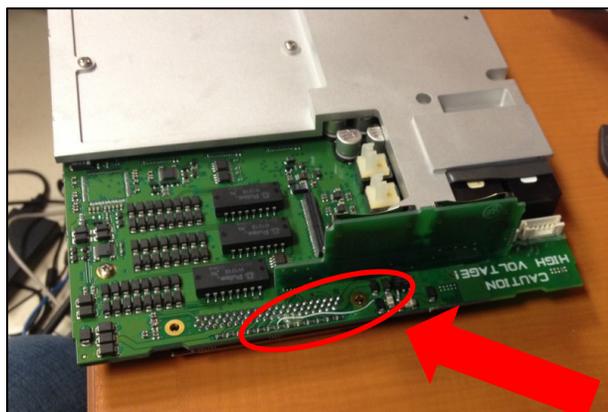


Figure 5.4 PCB d'un GDU

Après extraction de ce fil et remontage des GDUs, nous avons testé de nouveau avec un multimètre, la pine n 54 n'était plus reliée à la masse.

L'allumage des GDUs confirma nos soupçons, le mode Démo était maintenant désactivé et nous pouvions alors utiliser le système en mode normal comme dans un avion en état de vol.

## **5.4 Interfaçage avec Xplane**

Dans un premier temps, il nous parut évident de commencer par utiliser le G1000 avec un simulateur de vol pour ne pas prendre le risque de l'écrasement d'un drone suite à une mauvaise programmation.

Également, nous ne disposons d'aucune piste pour interfacier le G1000 avec un ordinateur, cependant le logiciel Xplane dans sa version Pro propose une option capable d'utiliser des GDUs. Nous avons donc acheté cette version afin d'avoir une base pour commencer notre interfaçage. Le seul problème étant que les personnes utilisant un G1000 avec Xplane sont rares et aucune notice d'emploi ne venait avec le logiciel. Un important travail d'interfaçage de retro-engineering fut donc nécessaire pour faire fonctionner les GDUs.

### **5.4.1 Câble d'adaptation et HSDB**

L'option de Xplane permettant de connecter le G1000 est très peu explicite; elle indique seulement quelques réglages au niveau de l'adresse IP et des ports Ethernet de l'ordinateur. Également, elle propose de régler la version des GDUs utilisés, le numéro d'un GIA et sa version. Aucune explication d'utilisation de cette option n'est disponible sur le manuel d'utilisation d'Xplane ou Garmin, même le programmeur en chef d'Xplane n'a pas voulu communiquer ces informations et nous a renvoyé vers un fabricant de simulateurs de vol. Ce dernier n'a rien voulu communiquer non plus prétextant ne pas vouloir dévoiler ses secrets industriels.

Sachant que le GIA est l'instrument de la suite G1000 qui communique les informations de vol avec les GDUs, nous sommes partis du postulat que l'ordinateur devait communiquer avec les GDUs en simulant la présence d'un GIA pour transmettre ces informations. Si on regarde dans la documentation Garmin, les GIAs et GDUs communiquent en HSDB sur les

ports Ethernet N°1 des GDUs. Nous avons donc câblé la sortie Ethernet d'un ordinateur sur le port Ethernet N 1 du PFD pour tenter une communication mais sans succès, aucune information n'arrivait jusqu'aux GDUs.

Pour tenter de comprendre, nous avons utilisé le logiciel Wireshark® afin de décoder les trames Ethernet échangées entre l'ordinateur et les GDUs.

Les premiers relevés ont été très instructifs. En effet, les seuls messages transmis était des trames de type broadcast (trames demandant la MAC adresse d'une adresse IP spécifique). Généralement, ces trames sont suivies d'autres trames émises de l'appareil avec l'IP demandée et indiquant leur MAC adresse. Dans le cas présent, aucune trame de réponse n'étaient envoyées.

On distinguait deux types de trames :

1. Les trames envoyées depuis l'ordinateur et demandant la MAC adresse de l'adresse IP 172.16.10.3.
2. Les trames envoyées depuis le PFD demandant un broadcast général avec un protocole non connu.

Après recherche dans les documents Garmin, nous avons trouvé qu'une adresse IP est attribuée à chaque port Ethernet d'un système G1000. Ainsi l'adresse IP 172.16.10.3 est attribuée à la sortie Ethernet N°1 du MFD. Nous avons donc fait les changements de câblage en conséquence.

Un nouvel essai avec Xplane nous a permis de remarquer un changement notable sur les GDUs, la position GPS de l'avion simulée par Xplane était transmise aux GDUs mais pas les informations de vol vitales tel que l'horizon artificiel, l'altitude ou encore la vitesse.

Une fois de plus, nous nous sommes servis de Wireshark pour étudier les trames envoyées par les deux systèmes.

Cette fois-ci les trames de broadcast initiales recevaient des réponses et chaque équipement (PC et GDU) émettait des trames UDP et AFS pendant les simulations.

Une des particularités de Wireshark est de pouvoir normalement décoder les trames Ethernet, malheureusement le HSDB est un protocole propriétaire de Garmin, son décodage n'était donc pas pris en compte.

Pour pallier au problème, nous avons donc mis au point un programme VBA sous Excel de plus de 300 lignes afin de décrypter les trames HSDB envoyées.

Ce programme VBA utilise les messages interceptés par Wireshark sous forme hexadécimale et les transforme pour avoir une lecture plus simple des paquets (séparation et décryptage des messages). La figure 5.5 montre un exemple de décodage de trames réalisé avec notre programme.

U	V	W	X	Y	Z	AA	AB	AC	AD	AE				AF	AG	AH	AI	AJ
HWM HSDB									IOPHSDB									
Destinati	Source Ad	Ack/Fi	Pi	Re	Frag	Res	Sequ	Size	Id	Id trad	Size	Va	Pr	Rese	Data			
000000ff	00000100	0010	B	00	00	00	0000	12	100	AHRS #1 roll ONLY for boxes unable to support level A	0004	01	01	0000	ffa3bbb			
000000ff	00000100	0010	B	00	00	00	0000	12	106	**RESERVED**	0004	01	01	0000	00000000			
000000ff	00000100	0010	B	00	00	00	0000	12	192	Pilot side active VOR deviation	0004	01	01	0000	00000000			
000000ff	00000100	0010	B	00	00	00	0000	28	176	GIA #1 VLC periodic data	0020	01	01	0000	00000000			
000000ff	00000200	0010	B	00	00	00	0000	28	183	GIA #1 VLC periodic data	0020	01	01	0000	00000000			
000000ff	00000100	0010	C	00	00	00	0000	16	538	GIA #1 COMM data	0008	01	01	0000	0001db64			
000000ff	00000200	0010	C	00	00	00	0000	16	546	GIA #2 COMM data	0008	01	01	0000	0001d8a8			
000000ff	00000100	0010	C	00	00	00	0000	44	50	AFCs annunciators	0036	01	01	0000	00000000			
000000ff	00000100	0010	B	00	00	00	0000	12	64	**RESERVED**	0004	01	01	0000	4072792b			
000000ff	00000100	0010	B	00	00	00	0000	12	66	**RESERVED**	0004	01	01	0000	3c5dc567			
000000ff	00000100	0010	B	00	00	00	0000	12	70	**RESERVED**	0004	01	01	0000	439d0acd			
000000ff	00000100	0010	B	00	00	00	0000	12	72	**RESERVED**	0004	01	01	0000	414f563d			

Figure 5.5 Exemple de décodage de trames HSDB

Grâce à ce programme, nous avons pu obtenir deux informations importantes :

- La première est que beaucoup de labels envoyés par Xplane nous sont illisibles (ID marqués **\*\*Reserved\*\***) et donc il était impossible de lire ces informations. Après élimination des autres IDs, on se rend compte qu'il s'agit des labels les plus importants (informations de vol primaires). Ces informations sont volontairement cachées par Garmin pour éviter que n'importe qui ne crée des plug-ins compatibles avec leur matériel.
- La seconde est que certains labels n'ont pas la longueur décrite dans la fiche technique, en effet, plusieurs trames envoyées par l'ordinateur ont une longueur inférieure à celle qu'elles devraient avoir. Preuve que les labels ne sont plus les mêmes ou ont évolué.

Après vérification de nos documents et de notre matériel, il s'est avéré que nous n'avions pas les mêmes versions de logiciel partout. Les GDUs ont été vendus avec une version logicielle 11.03, nos documents étaient pour une version 10.00 et Xplane est conçu par défaut pour une version 7.01.

Bien que cette vérification aurait pu être effectuée plus tôt dans nos recherches, il est très surprenant qu'une compagnie comme Garmin change la description de ces labels en fonction de la version logicielle. En temps normal, les constructeurs sont obligés de faire des mises à jour assurant la rétrocompatibilité des équipements avec les anciennes versions logicielles.

Étant donné qu'Xplane n'assure plus les mises à jour pour son plug-in avec le G1000, la seule solution pour débloquer notre situation consistait à down grader la version logicielle des GDUs pour coller avec celle de Xplane.

Le problème est qu'actuellement, on ne trouve sur le marché que des dernières mises à jour pour GDUs (version 12.03). Par chance, une compagnie de maintenance d'avion de l'aéroport de Saint Hubert possédait encore une vieille version dans ses archives.

Nous avons donc pu changer le logiciel des GDUs vers une version 8.03 (procédure relativement complexe normalement réservée à une main d'œuvre qualifiée (Cessna Aircraft Compagnie, 2012))).

Le résultat fut à la hauteur de nos attentes car grâce à cette nouvelle version les GDUs acceptèrent de recevoir les fonctions de vol primaires (horizon artificiel, Altitude, vitesse, etc.). En revanche, les paramètres moteurs et la radionavigation ne fonctionnent toujours pas. Mais dans le cadre de nos recherches expérimentales, les paramètres de vol primaires et la position GPS suffisent pour faire fonctionner nos drones, en guise d'interfaçage entre Xplane et les GDUs, le travail pu être considéré comme accompli.

#### **5.4.2 Plug-in et modèle 3D**

Les précédentes réalisations nous ont permis de connecter deux GDUs au logiciel Xplane et de réaliser que les données auxquelles nous avons accès ne nous permettent pas de créer notre propre plug-in indépendant pour interfacier un G1000.

En revanche comme Xplane est capable de communiquer avec le HSDB des GDUs, la solution la plus simple fut de créer un plug-in pour Xplane afin de lui envoyer les informations de vol transmises par le drone. Dans cette situation, Xplane sert donc d'interface entre les données du drone réceptionnées au sol et le HSDB du G1000 comme le montre la figure 5.6 ci-dessous.

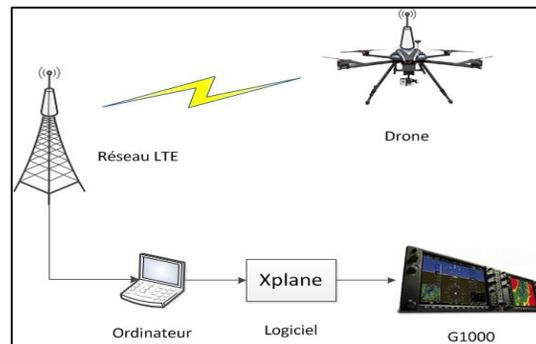


Figure 5.6 Chaîne de réception des informations du drone jusqu'au G1000

Il existe un grand nombre de plug-in développés par des particuliers pour fonctionner avec Xplane, ainsi beaucoup de sites web expliquent comment créer son propre plug-in.

L'architecture générale d'Xplane est la suivante : le programme regarde la dernière position 3D de l'avion, calcule sa nouvelle position et ses nouvelles vitesses en fonction de sa géométrie et des forces s'appliquant sur lui, puis calcule graphiquement les différents changements à appliquer au modèle 3D. Une illustration de cette chaîne est donnée à la figure 5.7.

Cette itération est appliquée toutes les 100 ms, autrement dit à une fréquence de 10 Hz. Lorsque l'on installe un plug-in, celui-ci est exécuté une fois à chaque itération entre la lecture des anciens paramètres et le calcul de la nouvelle position.

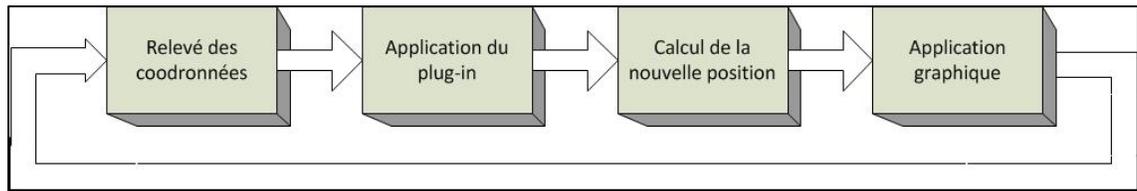


Figure 5.7 Étapes pour repositionner le modèle 3D de l'avion sous Xplane

Sur la théorie, il nous suffit donc de repositionner l'avion 3D avec les bons angles et les bonnes vitesses à chaque itération pour qu'Xplane transmette les informations au G1000.

Comme bon nombre de logiciel 3D, Xplane utilise un repère OpenGL, ce qui nécessite un changement de repère depuis les coordonnées ECI transmises par le drone. Cependant, Xplane ne transmet pas les coordonnées de l'origine du repère, il est donc impossible de déterminer la matrice de rotation, en revanche, il existe une fonction `XPLMWorldtoLocal` qui permet de passer du repère ECI au repère OpenGL de Xplane.

Notre plug-in se limite donc à repositionner l'avion à chaque itération en lui transmettant les coordonnées ECEF, les angles d'Euler et les différentes vitesses (translation et rotation) pour avoir tous les éléments désirés affichés ensuite sur les GDUs.

Cependant comme indiqué ci-dessus après chaque application du plug-in, Xplane recalcule la nouvelle position en fonction de la géométrie de l'avion, ce qui a pour conséquence de faire trembler l'image et les instruments (due au double remplacement de l'avion (une fois par le plug-in et une fois par Xplane)).

Pour parer à ce problème, nous avons eu l'idée de créer un nouvel avion très particulier sous Plane-Maker. En effet, la seule géométrie qui ne modifie pas les angles d'Euler lorsqu'elle est soumise aux frottements de l'air est une sphère. Nous avons donc construit le profil sphérique (figure 5.8) le plus petit possible avec un coefficient de frottement le plus petit possible lui aussi. Le but final étant d'offrir le moins prise au vent possible et les mêmes réactions quel que soit les angles d'Euler de l'avion.

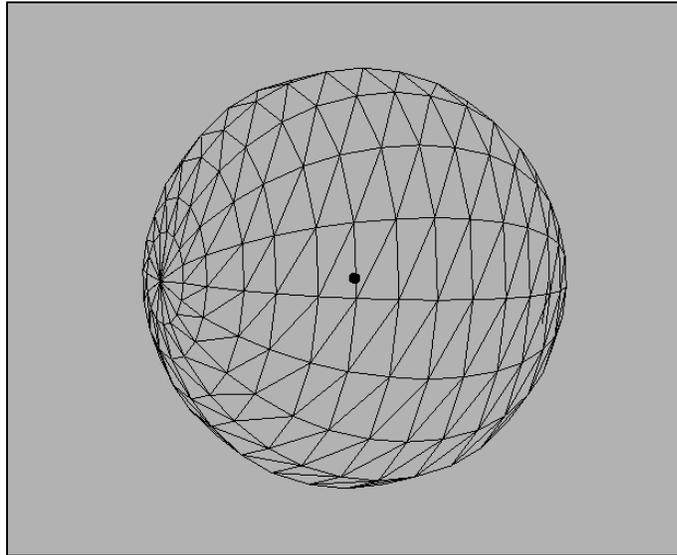


Figure 5.8 Sphère en 3D sous Plane-Maker

Le résultat fut très concluant car les instruments sont devenus relativement stables une fois cet aéronef chargé sous Xplane.

## 5.5 Couplage avec le Drone

Étant donné que les plug-ins d’Xplane sont en réalité des DLL, le problème est qu’il est impossible d’y stocker des données et/ou de maintenir une connexion avec un périphérique et/ou un autre ordinateur.

Il est donc impossible de créer un lien directement entre la Raspberry à bord du drone et Xplane car l’établissement d’une nouvelle connexion toutes les 100 ms serait trop longue et très instable pour la DLL.

Le seul moyen de récupérer les données du drone en temps réel et ensuite de les transmettre à Xplane est de créer une application externe qui maintiendrait la connexion avec la Raspberry. Cette application pourrait ensuite écrire les données du drone dans des variables partagées qui seraient à leur tour lues par le plug-in Xplane.

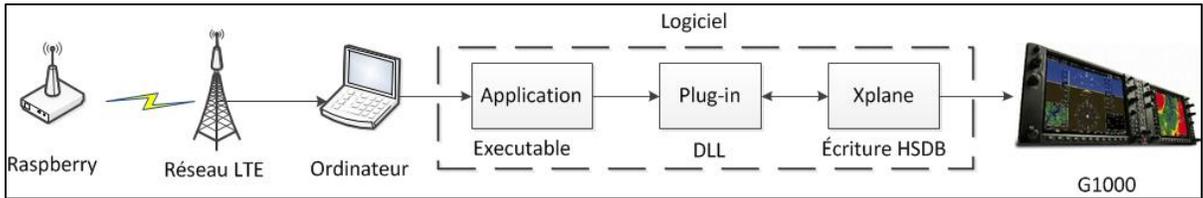


Figure 5.9 Chaîne complète pour affichage sur G1000

Cette chaîne de transmission de données de la figure 5.9 peut paraître complexe mais est nécessaire pour pouvoir afficher nos données sur un G1000.

Les tests préliminaires dans le laboratoire ont donné de très bons résultats quant à la vitesse de transmission des données entre la Raspberry et le G1000, il est difficile de la quantifier mais nous l'estimons à environ 500 ms avec une vitesse de rafraîchissement de 10 Hz (largement suffisant pour le pilotage).

Nous pourrions améliorer et simplifier cette architecture si nous disposions des données complètes du HSDB Garmin, en effet, ces informations nous permettraient de ne plus utiliser Xplane et d'envoyer nous même les trames au G1000. Dans cette situation, un seul exécutable permettrait de gérer à la fois la réception des données de la Raspberry et le HSDB. Malheureusement le seul moyen d'avoir les labels Garmin est d'acheter une licence programmeur auprès du constructeur. Mais ce type de licence est extrêmement onéreux et ne rentre pas dans le budget d'un projet comme le nôtre.



## CHAPITRE 6

### TESTS EN VOL

Pour vérifier la validité du matériel mis en place et des algorithmes implémentés, des tests en vol en cage et en extérieur furent organisés et réalisés. Il fallut également juger des qualités de vol, aussi bien pour le pilotage manuel que pour le pilotage automatique.

Il existe un guide de Rockwell International Corporation (1976) relatant la construction d'avions pilotés à distance qui propose une description détaillée des qualités de vol requises. Mais ce document semble aujourd'hui un peu obsolète et certains auteurs comme Williams (2003) déplorent l'absence de standards prenant en compte la performance des équipements et des algorithmes de contrôle des drones.

C'est la raison pour laquelle dans le cadre de nos tests, nous nous sommes basés sur les critères d'évaluation d'Akhrif (2013) basés sur les aéronefs généraux. Ils classent nos drones comme des aéronefs de classes I (< 5000 kg) et IV (haute manœuvrabilité). Les phases de vol (A, B et C) ont toutes été testées pour déterminer les qualités de vol (satisfaisantes, acceptables ou contrôlables) selon Nelson (1997, p 165).

#### 6.1 Tests en extérieur

Les premiers tests ont été effectués en extérieur avec la configuration décrite dans le chapitre 3. Ils se déroulèrent en cinq phases pour le contrôle manuel et sont récapitulés dans le tableau 6-1:

- Contrôle en RC avec vue directe;
- Contrôle en RC avec vue indirecte (FPV);
- Contrôle via LTE avec vue directe;
- Contrôle via LTE avec vue indirecte à distance rapprochée;
- Contrôle via LTE avec vue indirecte à longue distance (depuis l'ETS, environ 20km).

Tableau 6-1 Qualités de vol en contrôle manuel avec architecture commerciale

Phase de vol	Configuration RC avec vision directe	Configuration RC avec vision indirecte	Configuration LTE à courte distance avec vision directe	Configuration LTE à courte distance avec vision indirecte	Configuration LTE à très longue distance
A: Phase non terminale avec manœuvre rapide	Satisfaisant	Satisfaisant	Satisfaisant	Satisfaisant	Contrôlable
B: Phase non terminale avec manœuvre graduelle	Satisfaisant	Satisfaisant	Satisfaisant	Satisfaisant	Acceptable
C: Phases terminale graduelle	Satisfaisant	Satisfaisant	Satisfaisant	Satisfaisant	Non testé

Les résultats pour le control LTE à très longue distance sont relativement médiocres pour les raisons citées à la fin du chapitre 3 mais quelques améliorations matérielles mineures auraient pu rendre ces qualités de vol relativement satisfaisantes (cependant la campagne de test fut arrêtée car c'est à ce moment que nous avons décidé de changer la configuration matérielle du drone (passage du chapitre 3 au chapitre 4)).

Les qualités de vol en contrôle automatique pour l'architecture commerciale ont également été testées : le vol par points de cheminement (classe B) et les phases de décollage et d'atterrissage (Classe C). La Phases A n'a pas pu être testée car les manœuvres rapides ne sont pas envisageables en vol autonome. Ces tests sont résumés dans le tableau 6-2.

Tableau 6-2 Qualités de vol en contrôle automatique avec architecture commerciale (chapitre 3)

Phase de vol	A: Phase non terminale avec manœuvre rapide	B: Phase non terminale avec manœuvre graduelle	C: Phases terminale graduelle
Programmation de plan de vol à courte distance	non disponible	Satisfaisant	Satisfaisant
Programmation de plan de vol à très longue distance (LTE)	non disponible	Satisfaisant	Satisfaisant

En raison des conditions climatiques, les tests en vol avec la configuration décrite dans le chapitre 4 n'ont pas pu être réalisés. Le froid et la neige auraient rapidement eu raison de nos équipements et des batteries.

## 6.2 Tests en deux dimensions

Pour tester le pilotage manuel avec la transmission de données par LTE, nous avons commencé par monter la Raspberry et l'ensemble des capteurs sur une voiture téléguidée dans un premier temps. Les commandes des servomoteurs des modèles réduits sont assez semblables à celles des calculateurs de vol (commandes par PWMs) et presque aucune adaptation de commandes ne fut nécessaire.

Ce montage nous permis de vérifier le retour vidéo sur écran et les informations sur le G1000. Bien qu'il ne s'agisse pas vraiment de pilotage, ce premier prototype a permis de vérifier la fluidité des commandes en toute sécurité. Les résultats sont indiqués dans le tableau 6-3.

Tableau 6-3 Qualités de contrôlabilité du modèle  
en 2 dimensions

Phase de vol	A: Phase non terminale avec manœuvre rapide	B: Phase non terminale avec manœuvre graduelle	C: Phases terminale graduelle
Commande par LTE longue distance	Satisfaisant	Satisfaisant	Satisfaisant

### 6.3 Tests en cage de vol

Pour tester de façon plus représentative la contrôlabilité de la nouvelle architecture créée dans le chapitre 4 malgré les conditions météorologiques non favorables, nous avons effectué quelques tests en vol dans la cage de vol du club Dronolab à l'intérieur de l'atelier des clubs de l'ÉTS.

Ces tests ont une dimension assez restrictive car la cage de vol est limitée en termes de volume, les signaux satellites sont bloqués et la bande passante de la LTE est limitée. La navigation autonome et le pilotage à distance en FPV sont donc impossibles.

Cependant, des tests de commandes manuelles à vue et d'asservissement d'altitude automatique ont pu être réalisés et résumés dans le tableau 6-4.

Tableau 6-4 Qualités de vol en contrôle manuel avec  
architecture personnalisée

Phase de vol	A: Phase non terminale avec manœuvre rapide	B: Phase non terminale avec manœuvre graduelle	C: Phases terminale graduelle
Configuration RC avec vision directe	Satisfaisant	Satisfaisant	Satisfaisant
Configuration LTE à courte distance avec vision directe	Satisfaisant	Satisfaisant	Satisfaisant

Pour l'asservissement automatique de l'altitude, nous avons gardé le contrôle des axes latéral et longitudinal en manuel et laissé le contrôle de la puissance des moteurs à la Raspberry. Nous avons donc pu tester les capacités de stabilisation, changement d'altitude, décollage et d'atterrissage en mode automatique. Les résultats sont indiqués dans le tableau 6-5.

Tableau 6-5 Qualités de vol en contrôle d'altitude automatique avec architecture personnalisée

Phase de vol	A: Phase non terminale avec manœuvre rapide	B: Phase non terminale avec manœuvre graduelle	C: Phases terminale graduelle
Commande par LTE longue distance	Non disponible	Satisfaisant	Satisfaisant

Les résultats des tests ont été très satisfaisant en termes de stabilité mais un peu moins en termes de précision d'erreur statique. Comme le montre la figure 6.1 ci-dessous avec en rouge l'altitude de commande et en bleu l'altitude du drone.

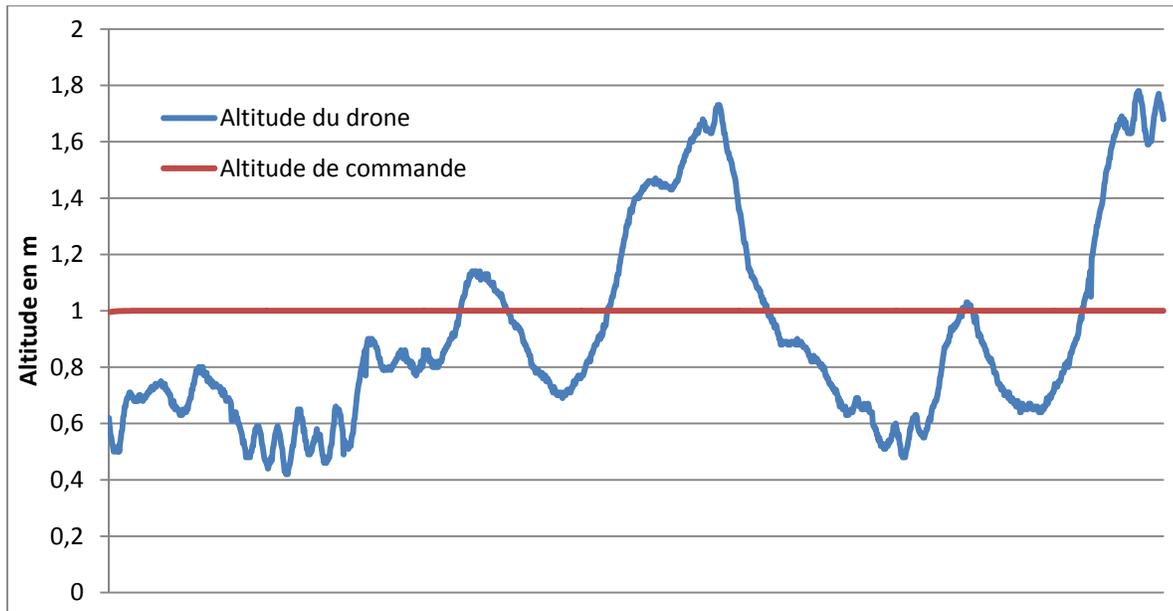


Figure 6.1 Test de maintien d'altitude avec PID

L'asservissement en altitude présentait des pics de plus ou moins 70 cm et des erreurs statiques allant jusqu'à 40cm due aux variations de tension de la batterie. Nous avons donc ajoutée un terme correctif évolutif comme celui utilisé pour contrer l'erreur de position dans l'algorithme de navigation. Ce terme augmente progressivement la valeur des PWMs quand le drone est statique pour minimiser les erreurs en régime permanent.

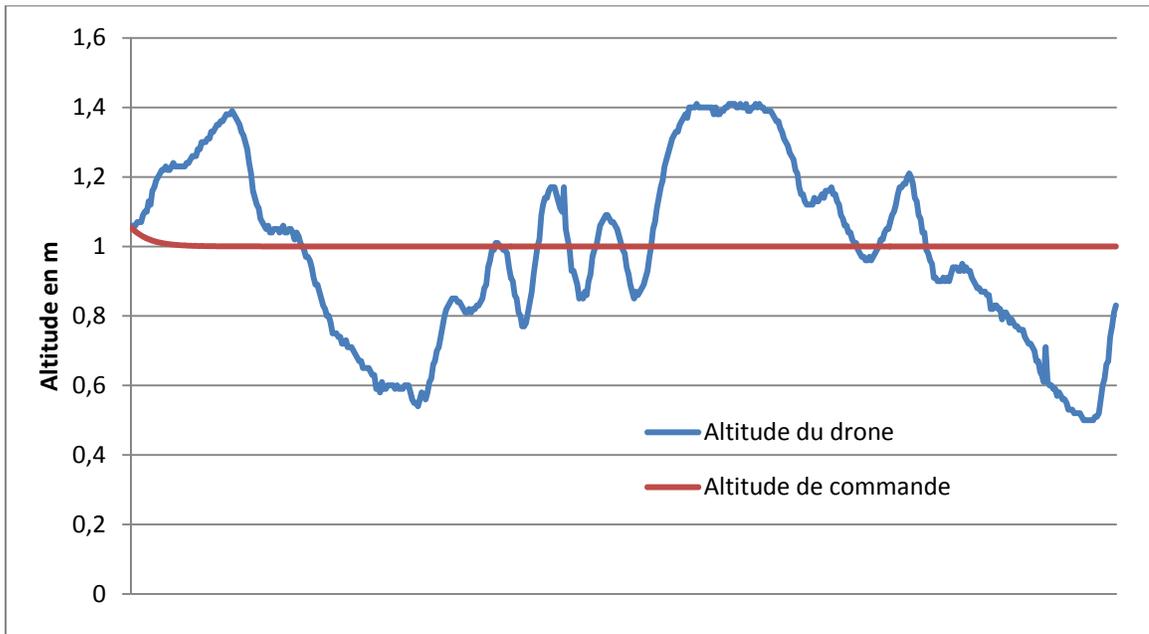


Figure 6.2 Test de maintien d'altitude avec PID et correcteur d'erreur statique

Bien que le terme correctif ne soit pas aussi efficace que pour l'algorithme de navigation du chapitre 4, la figure 6.2 montre qu'il permet au moins de diminuer les pics de 70cm à 40cm, réduisant ainsi les erreurs maximales d'environ 40%.

Après plusieurs tests, nous nous sommes également rendu compte que la majorité des erreurs venaient du contrôleur de vol. Étant programmé pour maintenir l'attitude du drone, il se charge de le stabiliser avec sa centrale inertielle. Mais les tests ayant lieu à l'intérieur, il ne dispose d'aucun moyen pour recalibrer les capteurs MEMSs avec le récepteur GPS, par conséquent, il dérive avec le temps, provoquant la majorité des changements d'altitude observés.

L'utilisation du drone en extérieur devrait régler ce problème mais nous n'avons malheureusement pas eu la possibilité d'effectuer ces tests.



## CHAPITRE 7

### DISCUSSION RELATIVE AUX AMÉLIORATIONS FUTURES

Bien que cette maîtrise se termine, il reste encore beaucoup d'améliorations possibles à apporter à ces drones. Le présent chapitre avisera le lecteur de quelques axes sur lesquels un approfondissement serait possible au sein du laboratoire LASSENA.

#### 7.1 FOTAFMS

Le FOTAFMS est l'abréviation de Fly Over The Air FMS (Flight Management System). L'idée du Professeur Landry dans ce projet est de pouvoir d'abord rapporter au sol toutes les informations des capteurs/données d'un aéronef et de permettre des mises à jour du plan de vol d'un avion via le FMS au sol et un lien sans fil sécurisé, FMS qui se veut être une réplique en tout point du FMS dans l'avion. En effet, l'enjeu est double, i.e. permettre à nouveau de traiter les données de vol et/ou prendre le contrôle complet de l'aéronef depuis le sol si un problème arrive en vol.

Les avantages d'une telle technologie sont multiples et les applications très innovantes :

- Les calculs correctifs faits à bord de l'aéronef peuvent être refaits et approfondis pour détecter d'éventuelles pannes et/ou d'éventuelles corrections supplémentaires. Comme les calculateurs ne sont pas embarqués à bord de l'avion, ils peuvent être beaucoup plus gros, beaucoup plus puissants et plus énergivores.
- Lors d'une panne à bord, les pilotes sont souvent livrés à eux-mêmes et manquent d'expérience en matière de pannes. Confier les commandes à une équipe au sol plus nombreuse et plus expérimentée en termes de pannes est une sécurité qui pourrait éviter bien des incidents, technologie qui aurait pu permettre d'éviter plusieurs accidents du passé.
- Également, les systèmes au sol ne seront pas considérés comme des équipements de vol. Ils seront donc soumis à des contraintes de fabrication et de certification bien

moins drastiques que celles appliquées aux avions. La rentabilité coûts de développements versus sécurité ajoutée est donc très intéressante.

- Le système permettrait d'agir comme une boîte noire virtuelle, telle les boîtes noires actuellement présentes dans les avions, mais avec un « backup » constant au sol, ce qui permettrait le suivi plus précis de l'évolution d'un avion.

Avec le matériel développé sur le drone et le G1000 au sol, une solide base est déjà en place pour tester cette nouvelle technologie et l'appliquer à des drones ou des avions avec pilotes.

## **7.2 Amélioration des Avioniques**

Le chapitre 5 de ce document a expliqué comment interfacer un G1000 avec les moyens à disposition aussi bien matériels que monétaires. L'interfaçage est terminé mais reste soumis aux contraintes de Xplane et ses divers aléas (mises à jour et limitation des fonctions). Cependant, il est encore possible d'améliorer ce système de deux manières différentes :

- L'achat d'une licence programmeur auprès de Garmin pourrait permettre d'interfacer nous même le G1000 sans passer par Xplane. Actuellement, cette solution n'est pas possible car nous ne disposons ni des droits ni des fichiers data permettant de le faire.
- Le LASSENA est en pleine discussion avec le groupe CMC Electronics pour négocier l'offre d'un FMS de type FlightDeck pour le laboratoire. Si ces discussions aboutissent, le FlightDeck, très semblable au G1000 pourra le remplacer ce qui donnerait un atout majeur à CMC avec son partenariat avec le LASSENA. Le principal avantage est que cette fois-ci le constructeur (CMC Electronic) fournira aussi les datasheets permettant d'interfacer le FMS nous même sans passer par Xplane.

Les deux solutions présentées permettraient entre autre de débiter des recherches plus approfondies sur le FOTAFMS cité au paragraphe précédent.

### 7.3 TCAS et trajectoires d'évitement

Le système de réseau de drone décrit dans le chapitre 4.2.8 est très novateur dans le domaine des UAVs mais peut avoir de très nombreuses applications comme nous l'avons expliqué.

L'une des principales est la détection et l'évitement des autres drones, dans l'aviation civile et militaire, il existe un système très semblable appelé « TCAS » (de l'anglais : Traffic Collision Avoidance System). Le « TCAS » est un système permettant de détecter et d'afficher les autres aéronefs à proximité, la finalité est bien sûr d'éviter les collisions en vol. Si deux avions de dernière génération sont équipés de TCAS et sont susceptibles d'entrer en collision, le PFD du pilote est capable de proposer des trajectoires d'évitement.

Aujourd'hui le système embarqué dans nos drones fonctionne de la même manière, chaque individu transmet sa propre position aux autres UAVs, et il est possible de calculer les distances entre drones pour prévoir d'éventuelles collisions. Pour l'instant l'algorithme de navigation se limite à arrêter l'un des deux drones pour laisser passer l'autre. Mais ce système pourrait être amélioré en proposant des trajectoires d'évitement et des priorités entre drones.

À terme ce système pourrait même devenir compatible avec celui des avions commerciaux et permettre une cohabitation UAVs/avions en toute sécurité.

### 7.4 Réalité augmentée et support de suspension « gimbal »

Sachant que les drones sont équipés de support de suspension (« gimbal ») pour la stabilisation de l'image et que leurs angles peuvent être paramétrables, il serait possible d'utiliser des lunettes de type Google Glass équipées d'accéléromètres pour lier les mouvements du support de suspension et la tête du pilote. Cela permettrait de nombreuses améliorations de pilotage :

- Lorsque le pilote veut tourner la tête pour changer d'angle de vu, la caméra lui donnera cette opportunité sans demander d'efforts supplémentaires et sans changer la trajectoire de vol.

- Des éléments pourront être ajoutés pour jouer le rôle de HUD via les lunettes. Le pilote n'aura donc plus à baisser les yeux pour regarder les instruments, ils seront devant lui par transparence.
- Ce type de lunettes permet également de s'ouvrir au monde de la réalité augmentée. Les instruments ne seront pas les seuls éléments affichés, il sera également possible d'identifier les autres appareils, les obstacles, les aéroports, les zones aériennes, etc. Comme il s'agit de réalité augmentée, ces éléments seront représentés en 3D et bougeront en même temps que la tête du pilote grâce aux lunettes équipées d'accéléromètres.

Comme beaucoup de paramètres de vol sont déjà envoyés jusqu'au sol via le lien LTE, les seuls achats nécessaires pour cette application sont une paire de lunettes de type Google Glass et un support de suspension sur 3 axes de rotation (celles équipées actuellement sont sur 2 axes seulement).

## **7.5 Détection optique d'obstacles**

L'un des problèmes actuels de nos drones est qu'ils sont incapables de détecter un objet arrivant de front. Ainsi, si l'opérateur programme une mauvaise trajectoire qui va à l'encontre d'obstacles tels que des immeubles ou des arbres, le drone ne pourra pas les détecter. Une solution élégante consisterait à faire du traitement de l'image avec la GoPro embarquée. En effet les algorithmes de navigation à bord sont programmés pour toujours ajuster la caméra dans le sens de la marche du drone (voir algorithme de navigation chapitre 4.3.1), si un obstacle se présente, il sera alors immédiatement vu par la caméra.

Faire du traitement de l'image en temps réel avec la Raspberry permettrait de pouvoir faire des trajectoires d'évitement, voir à terme de l'autoguidage d'un point A à un point B. Aujourd'hui la législation et l'espace aérien ne sont pas encore prêts pour accepter des UAVs capables de faire de l'autoguidage, mais avec des projets comme AMOSS qui assureront une sécurité et une robustesse assez poussée, il n'est pas impossible de voir émerger ce genre de technologie.

## 7.6 Parachute de sécurité

Aujourd'hui, les drones sont de plus en plus gros et de plus en plus nombreux, et ils commencent à envahir l'espace aérien de manière virale. Bien que leur utilisation se démocratise, ils n'en restent pas moins des équipements de vol dangereux qui ne suivent pas les mêmes directives de sécurité que les appareils transportant des passagers.

Une panne en vol est immédiatement catastrophique car aucun système de sécurité n'est aujourd'hui prévu pour empêcher que l'appareil ne s'écrase. De plus, une collision entre deux drones devient de plus en plus probable vu leur multiplication. Au-delà des dégâts matériels qu'ils peuvent engendrer en s'écrasant, ils peuvent également créer des dommages corporels.

Pour y remédier, il serait judicieux d'incorporer des parachutes pour ralentir leur chute et protéger ce qui se trouve en dessous d'eux. Le poids d'un parachute est relativement faible, cependant il est difficile de le déployer sans qu'il ne s'accroche dans les hélices s'il n'est pas muni d'un système pyrotechnique. Mais déjà, la proximité des drones effraie certaines personnes due à la dangerosité des hélices, il semble inimaginable de les munir d'explosifs.

Une recherche sur le déploiement de systèmes de sécurité passifs de type parachute ou airbag serait donc bénéfique pour les milieux des UAVs.

## 7.7 Nouveau calculateur de bord

Bien que les applications d'aujourd'hui codées sur la Raspberry soient relativement simples, nous avons rencontré des difficultés liées à l'architecture matérielle. En effet, comme nous utilisons de nombreux capteurs simultanément, la Raspberry peine à traiter les informations et le contrôleur Ethernet qui a la charge de gérer les ports USB est rapidement saturé.

Avec ses 512 Mo de RAM et son processeur à 700 MHz, la Raspberry est aujourd'hui largement dépassé par des concurrents comme le Odroid U3 qui propose 2 Go de Ram et un processeur quadricœur à 1.7 GHz. Actuellement, même les téléphones intelligents possèdent des puces plus puissantes que la Raspberry.

Le changement de calculateur de bord serait donc à envisager pour augmenter la puissance de calcul du drone et ne pas gêner le développement de futures applications à bord.

## **7.8 Nouvelle charge utile**

Nos drones ont été conçus de manière à être facilement modifiables et reprogrammables. De plus avec leur charge utile d'environ 1 Kg, ils offrent de multiples applications possibles. Ils peuvent facilement être adaptés pour répondre à tous types de tâches en fonction des capteurs et actionneurs que l'on leur attache.

Si aujourd'hui nous ne disposons que d'un simple bras télescopique et d'une caméra, les applications que nous lui attribuerons dans le futur sont illimitées. On rencontre tous les jours de nouveaux types de drones sur le marché dotés de capacités révolutionnaires : défibrillateurs pour secourir des personnes atteintes de malaises cardiaques, transport de bouées de sauvetage pour les nageurs en difficulté, etc.

Avec la technologie LTE embarquée, notre drone est même capable de transmettre en direct les données collectées à bord. Des nouveaux systèmes de détection peuvent donc être mis en place. Également, nous pouvons imaginer que dans le futur les tests de certification de certains équipements de vol pourront être faits par des drones, réduisant significativement le coût de leur certification en vol.

## CONCLUSION

Au cours de ce mémoire, nous avons eu une brève introduction au domaine des drones. Ce secteur est en plein boom économique et se développe à une vitesse folle. Pour l'instant, les technologies des UAVs sont encore balbutiantes et beaucoup de développements sont encore à faire pour les applications à venir.

Le projet AMOOS nous a donné un aperçu de certains enjeux auxquels il faut s'attendre et quels en sont les dimensions. Il nous a également permis d'acquérir de solides connaissances dans le domaine du spatial et de la simulation de vol. Son aspect théorique a également permis d'obtenir de l'expertise dans le domaine des changements de coordonnées dans les repères de navigation, ce qui fut un réel atout par la suite.

Nous avons commencé notre contrôle de drone à distance en s'efforçant d'utiliser des technologies commerciales mais ces dernières étaient beaucoup trop contraignantes et nous avons été obligés de créer notre propre système. Cependant la construction de notre propre drone avec des moyens plus spartiates nous a permis de bâtir un système de navigation autonome et adaptatif. Le choix d'un ordinateur de bord relativement commun nous a autorisé une grande flexibilité quant aux capteurs, actionneurs et moyens de communication mis en place.

Aujourd'hui le LASSENA est équipé de drones performants capables d'une grande souplesse d'évolution pour le développement des futures technologies embarquées. Ils sont capables d'être pilotés à l'autre bout de la planète avec de véritables instruments d'avion, d'effectuer des manœuvres automatiquement et de communiquer entre eux. Des outils algorithmiques, logiciels et matériels ont également été mis en place pour permettre aux prochains étudiants de continuer le projet sur une base solide et faire facilement évoluer les performances actuelles.



## RECOMMANDATIONS

Au cours de ce mémoire, un certain nombre de tests ont été effectués avec les drones, une attention particulière fut toujours portée sur la dangerosité de ces machines. Il n'est jamais inutile de répéter que leurs hélices tournent à près de 900 Km/h à plein régime et qu'elles ont été conçues pour fendre tout ce qui s'oppose à elles.

L'élaboration d'un test en vol n'est pas à prendre à la légère et ne doit pas être décidée pour des raisons récréatives car ils restent toujours dangereux. Même s'ils n'ont pas toujours été couronnés de succès, les tests effectués lors de cette maîtrise ont toujours été priorisés par un aspect sécuritaire. Et bien que commandés à distance ou totalement automatisés, les tests ont toujours été effectués à portée de vue et avec une possibilité de reprendre un contrôle en RF à tout moment par un pilote qualifié.

Ayant suivi le cours MGA856 (Ingénierie et principes des essais en vol) et ayant un brevet de pilote privé, j'ai personnellement reçu une formation un peu plus approfondie que beaucoup d'élèves quant au déroulement de ce type de test. Mais même ces formations ne sont pas suffisantes pour garantir une sécurité à 100%.

Par expérience personnelle et conseils reçus par des professionnels, il est fortement conseillé de garder en tête la liste des items suivant avant de commencer un test en vol :

- Le test a-t-il lieu d'être? A-t-il un réel enjeu et quelque chose à vérifier ou mesurer?
- Une liste précise des objectifs de ce test est-elle dressée et les différents résultats qu'il peut engendrer sont-ils tous connus? N'y a-t-il pas trop d'éléments nouveaux à tester une seule fois?
- Les tests préliminaires effectués dans le laboratoire ont-ils tous été satisfaisants pour permettre de dire que les tests en vol ont toutes les chances de succès ? N'y a-t-il aucun bug dans le code ou aucune incertitude sur la réaction du drone?

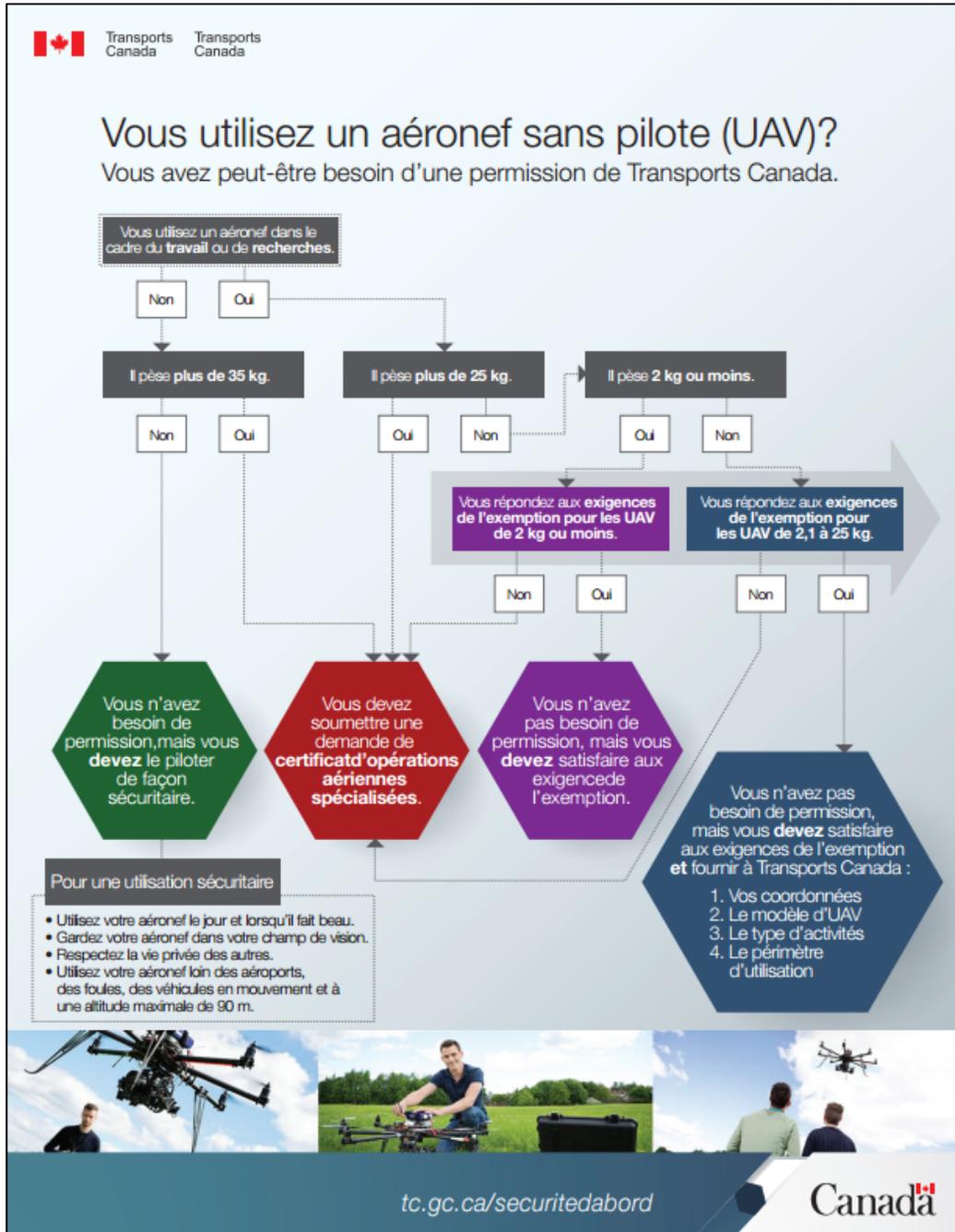
- Le site de test se prête-il au type de test effectué? (une cage de vol ne suffit-elle pas?), Pour les tests en extérieur, le lieu est-il approprié? (pas de personnes autour? pas d'obstacles gênants?).
- La législation est-elle respectée pour ce test? (consulter régulièrement Transport Canada, actuellement les tests conduits sur les petits drones sont toujours fait à vue et avec une priorité sur un contrôle manuel d'un opérateur local, il n'y a donc pas de problème de législation. Si ces conditions de vol évoluent, faire une demande écrite de certificat d'opération à Transport Canada (Transport Canada, 2014) <http://www.tc.gc.ca/fra/aviationcivile/normes/generale-aviationloisir-uav-4161.html>)
- Le matériel est-il approprié? Ai-je le bon type de drone, les bonnes batteries et les bons matériels de communication?
- Les conditions météorologiques du jour sont-elles propices? (beau temps, pas trop de vent et une température clémente pour les batteries).
- Quelles sont les procédures de sécurité en cas de problème? Comment détecter une panne et reprendre le contrôle du drone? Le GoHome est-il une solution envisageable? Quelqu'un est-il au courant de notre position et sommes-nous capables de prévenir les secours en cas de blessure?
- Toutes les personnes présentes lors de ce test sont-elles importantes et ont-elles toutes des tâches assignées précises et correctement identifiées?
- Une check-list est-elle prête pour vérifier tous les systèmes une dernière fois avant le décollage?

Si tous les éléments cités sont réunis, les tests peuvent être envisagés dans les meilleures conditions mais continuent toutefois à présenter de réels dangers.

Un vieux proverbe aéronautique dit « qu'il n'y pas de bon pilote mais que de vieux pilotes », l'interprétation de ce proverbe nous apprend qu'un des principaux ennemis est l'habitude et la répétabilité en aérospatial. Ce domaine est tellement vaste et tellement soumis à diverses contraintes qu'il mérite une attention de tous les instants et une rigueur sans failles pour qu'il puisse continuer à exister. Les tests doivent donc toujours être effectués avec la même attention et la même rigueur du premier au dernier.

# ANNEXE I

## Gestion des UAVs par Transport Canada.



## Exigences des exemptions relatives à l'utilisation sans permission d'UAV

INFOGRAPHIE FOURNIE UNIQUEMENT À DES FINS PRATIQUES. CONSULTER LA VERSION OFFICIELLE DES EXEMPTIONS.

### UAV de 2 kg ou moins

- Être prudent, avoir reçu la formation nécessaire et connaître les règles de l'espace aérien.
- Avoir 18 ans, ou avoir au moins 16 ans pour effectuer des recherches sous la supervision d'un établissement d'enseignement.
- Avoir une assurance responsabilité d'au moins 100 000 \$.
- Être alerte et en aucun temps sous l'influence de drogues ou d'alcool.
- Avant le vol, inspecter l'UAV et l'emplacement pour s'assurer qu'ils sont sécuritaires.
- Demander la permission avant de survoler une propriété privée.
- Communiquer avec les services de la circulation aérienne si l'UAV pénètre un espace aérien contrôlé.
- Accorder la priorité de passage aux aéronefs avec pilote.
- Utiliser l'UAV le jour et lorsqu'il fait beau.
- Utiliser l'aéronef en visibilité directe et le garder dans son champ de vision en tout temps.
- S'assurer que les fréquences ou les transmissions radio n'auront pas d'incidence sur le contrôle de l'UAV.
- Avoir un plan d'urgence.
- Avoir avec soi une copie de l'exemption relative à l'UAV, une preuve d'assurance responsabilité, ses coordonnées et l'information sur les limites des systèmes de l'aéronef.
- Respecter les procédures d'utilisation et d'urgence du constructeur, y compris celles qui s'appliquent si l'aéronef ne répond plus au système de télécommande.
- Respecter les lois de tous les ordres de gouvernement.
- N'utiliser qu'un UAV à la fois avec une seule télécommande.
- Cesser immédiatement toute activité s'il n'est plus possible de satisfaire aux exigences de l'exemption ou si la sécurité des personnes, des biens matériels ou des autres aéronefs est compromise.
- Rester à au moins 30 m des personnes, des animaux, des bâtiments, des structures et des véhicules qui n'ont rien à voir avec l'activité.

### UAV de 2,1 à 25 kg

- Être prudent, avoir reçu la formation nécessaire et connaître les règles de l'espace aérien.
- Avoir 18 ans.
- Avoir une assurance responsabilité d'au moins 100 000 \$.
- Être alerte et en aucun temps sous l'influence de drogues ou d'alcool.
- Avant le vol, inspecter l'UAV et l'emplacement pour s'assurer qu'ils sont sécuritaires.
- Demander la permission avant de survoler une propriété privée.
- Avoir avec soi une copie de l'exemption relative à l'UAV, une preuve d'assurance responsabilité, ses coordonnées et l'information sur les limites des systèmes de l'UAV.
- Respecter les lois de tous les ordres de gouvernement.
- Utiliser l'UAV en visibilité directe et le garder dans son champ de vision en tout temps.
- N'utiliser qu'un UAV à la fois avec une seule télécommande.
- Accorder la priorité de passage aux aéronefs avec pilote; ne pas emprunter leurs trajectoires de vol.
- Utiliser l'UAV le jour et lorsqu'il fait beau (aucun nuage, aucune condition de neige ou de glace).
- Elaborer des procédures relatives à l'atterrissage et à la récupération de l'UAV, ainsi qu'à la communication avec les services d'urgence et de contrôle de la circulation aérienne, et les respecter.
- Avoir un plan d'urgence.
- Respecter les procédures d'utilisation et d'urgence du constructeur, y compris celles qui s'appliquent si l'aéronef ne répond plus au système de télécommande.
- S'assurer que les fréquences ou les transmissions radio et les dispositifs électroniques n'auront pas d'incidence sur le contrôle de l'UAV.
- Évaluer le risque de perte de contact avec l'UAV et décider à quel moment le système d'interruption de vol doit être activé.
- Avoir un extincteur sur les lieux.
- Communiquer avec les services de la circulation aérienne avant que l'UAV ne pénètre dans un espace aérien contrôlé.
- Respecter les instructions d'assemblage et de maintenance du constructeur.
- S'assurer que l'UAV n'est pas muni d'une radiobalise de repérage d'urgence.
- Signaler tout accident à Transports Canada et cesser toute utilisation jusqu'à ce que les risques soient écartés.
- Cesser immédiatement toute activité s'il n'est plus possible de satisfaire aux exigences de l'exemption ou si la sécurité des personnes, des biens matériels ou des autres aéronefs est compromise.
- Restez à au moins 150 m des personnes, des animaux, des bâtiments, des structures et des véhicules qui n'ont rien à voir avec l'activité.

### NE PAS UTILISER UN UAV :

- À moins de 9 km d'un feu de forêt, d'un aéroport, d'un hélicoptère, d'un aérodrome ou d'une zone bâtie.
- Au-dessus de bases militaires, de prisons ou dans un espace aérien contrôlé ou réglementé.
- Au-dessus de foules ou à plus de 90 m d'altitude.
- Pour participer à une activité aéronautique spéciale, à un spectacle aérien ou à une démonstration de systèmes.
- Pour transporter des marchandises dangereuses ou des lasers.

Figure-A I-1 Gestion des UAVs par Transport Canada

## ANNEXE II

### Schéma du Simulateur HeliCrew

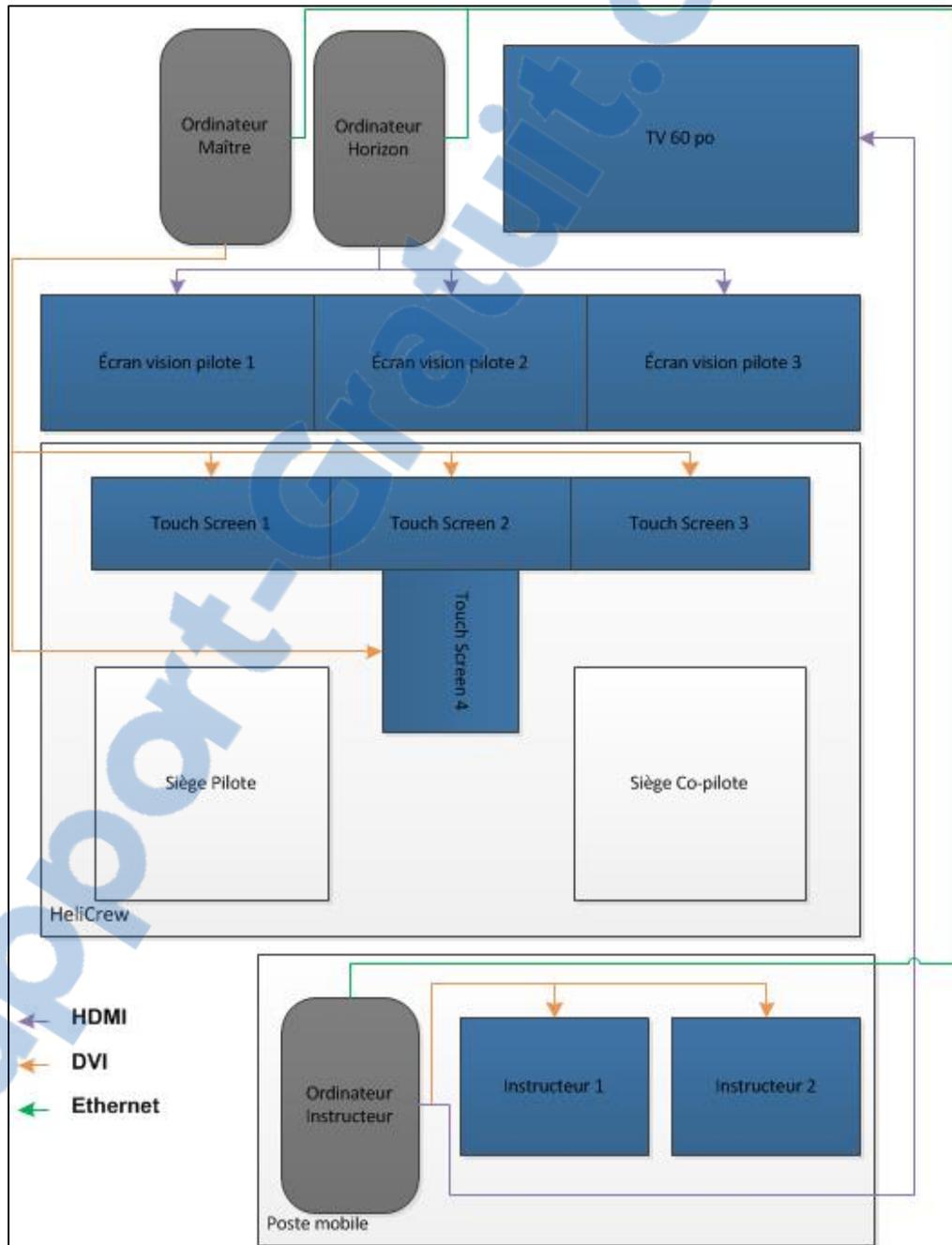


Figure-A II-1 Schéma du simulateur Hélicrew



## ANNEXE III

### Création du bras articulé

#### Structure mécanique :

Afin de réaliser ce bras, nous avons commencé par choisir deux types de servomoteurs, l'un pour contrôler la longueur du manche télescopique (360° de rotation) et l'autre pour diriger ce manche (60° de rotation).

La deuxième étape fut de modéliser ces servomoteurs en 3D afin de construire les autres éléments du bras autour d'eux (figure III-1 et figure III-2).



Figure-A III-1 Servomoteur à 60°

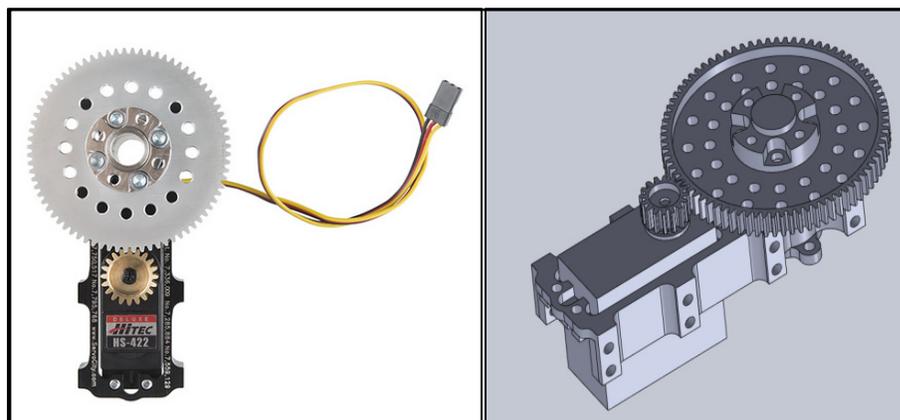


Figure-A III-2 Servomoteur à 360° avec réducteur

La difficulté principale pour concevoir les autres pièces est qu'elle doivent être les plus petites possible en tenant compte de leur solidité (en plastique) et de leurs contraintes de fabrication. En effet, ses pièces sont fabriquées grâce à une imprimante 3D qui dépose du plastique fondu par couches successives. L'imprimante ne peut pas imprimer sur du vide car elle ne dispose pas de support (la fabrication d'un pont n'est pas possible, par exemple), la conception des pièces fut donc contrainte par l'angle sous lequel elle était imprimée.

Ainsi, un totale de 10 pièces furent conçues et imprimées afin de concevoir un bras léger, robuste et maniable (exemples en figure III-3 et aperçu finale en figure III-4).

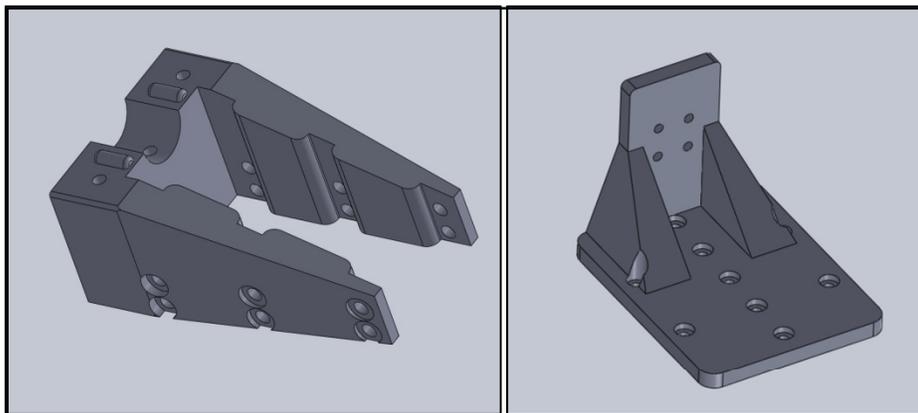


Figure-A III-3 Exemples de pièces fabriquées

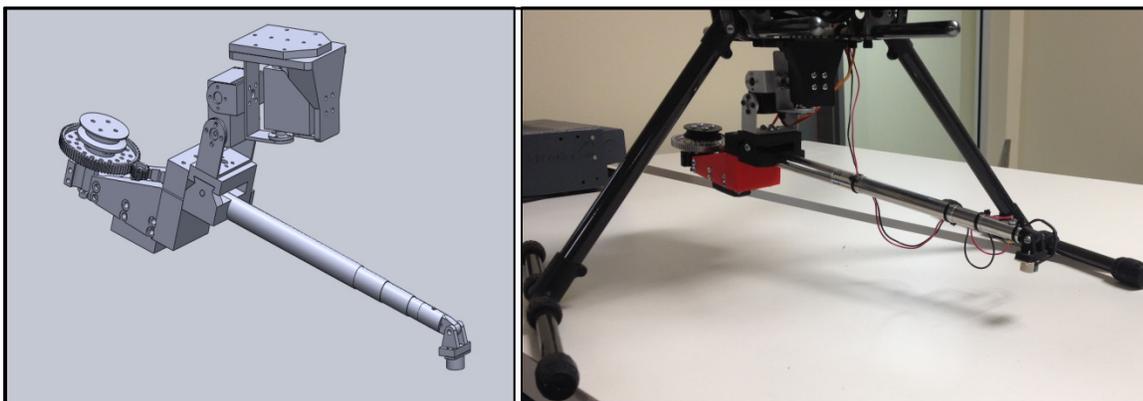


Figure-A III-4 Bras assemblé en 3D et en réalité

### **Commande électrique :**

Une fois assemblé, il nous fallut piloter le bras à distance à l'aide des 3 servomoteurs et de l'électroaimant.

Les seuls moyens de communication avec le drone vendus dans le commerce sont des télécommandes RC à 2.4 GHz qui génèrent des PWMs (Pulse Width Modulation). Les émetteurs permettent de commander jusqu'à 10 canaux simultanément. Malheureusement pour nous, après l'acheminement des commandes de vol, du train d'atterrissage et du support de suspension, il ne nous restait que 2 canaux de libre pour 4 commandes à envoyer (3 servomoteurs et l'électroaimant). Pour cela, nous avons décidé de multiplexer les commandes du drone afin de les faire passer sur les deux canaux restants.

Le principe du multiplexage est de pouvoir commander plusieurs canaux avec seulement 2 commandes (une commande de sélection de canal et une commande avec l'information destinée au canal sélectionné). Pour envoyer un ordre sur l'un des canaux, il faut donc le sélectionner via la première commande et ensuite envoyer l'ordre avec la seconde. Mais lorsque l'on sélectionne un autre canal, le dernier ordre envoyé n'est pas gardé en mémoire. Ce qui est très gênant dans notre cas car les servomoteurs du bras doivent en permanence recevoir des ordres (sinon ils ne conservent pas leur position). Il nous a donc fallu trouver un moyen de conserver les derniers ordres du multiplexeur et de les transmettre en tout temps aux servomoteurs. Pour cela nous avons programmé un Arduino Nano (microcontrôleur programmable) pour qu'il génère en permanence des signaux PWMs afin de maintenir le bras en position à bord du drone. L'Arduino fit aussi office de démultiplexeur afin de transformer les 2 canaux transmis depuis le sol (un canal de sélection et un canal de commande) en 4 canaux de commande pour le bras. La commande du bras est illustrée à la figure III-5 ci-dessous.



Figure-A III-5 Commande du bras télescopique

L'un des autres problèmes fut que l'alimentation de l'électroaimant n'était pas possible directement par l'Arduino car l'intensité de ses pines en sortie n'était pas assez grande. Pour cela, nous avons fait un interfaçage avec un transistor MOSFET (commande en tension représenté figure III-6) pour apporter la puissance nécessaire à l'électroaimant.

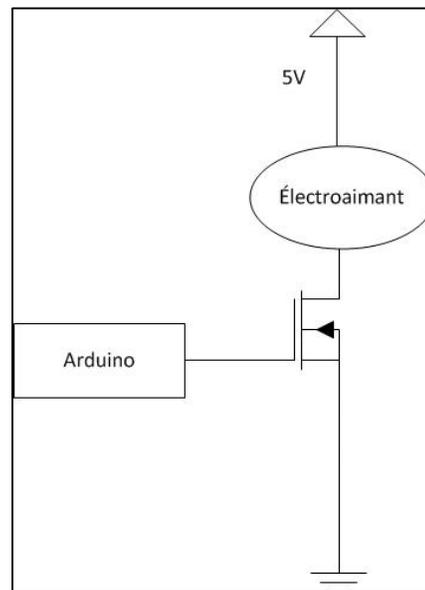


Figure-A III-6 Interfaçage de l'électroaimant

Grâce à ces systèmes, nous avons donc pu commander le bras à distance depuis le sol en utilisant le multiplexage des signaux de commande.

# ANNEXE IV

## Interfaces de pilotage LabVIEW

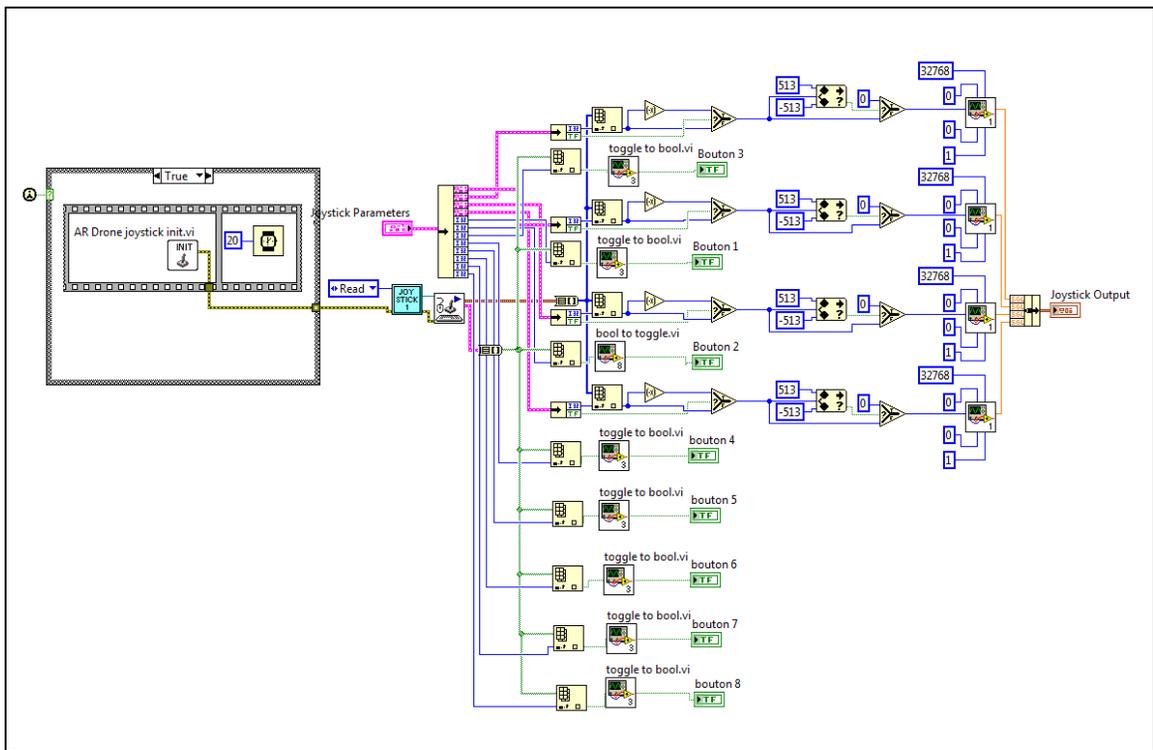
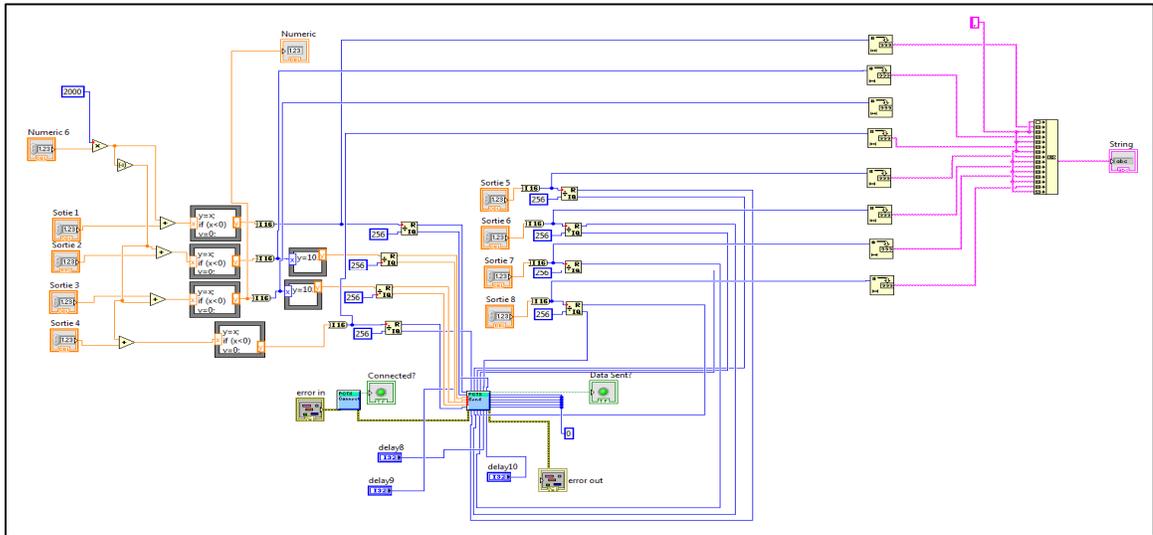


Figure-A IV-1 Interfaces de pilotage LabVIEW



## ANNEXE V

### Sélecteur de canaux

Afin de réaliser le sélecteur de canaux entre l'ordinateur de bord et le récepteur RC, nous avons conçu un montage électrique à base de transistors MOSFET (figure V-1) capable de sélectionner une source de tension en fonction d'un état (haut ou bas) arrivant sur le circuit. Le but était de choisir entre les PWMs issus de la Raspberry ou de la télécommande avec une seule autre commande indépendante.

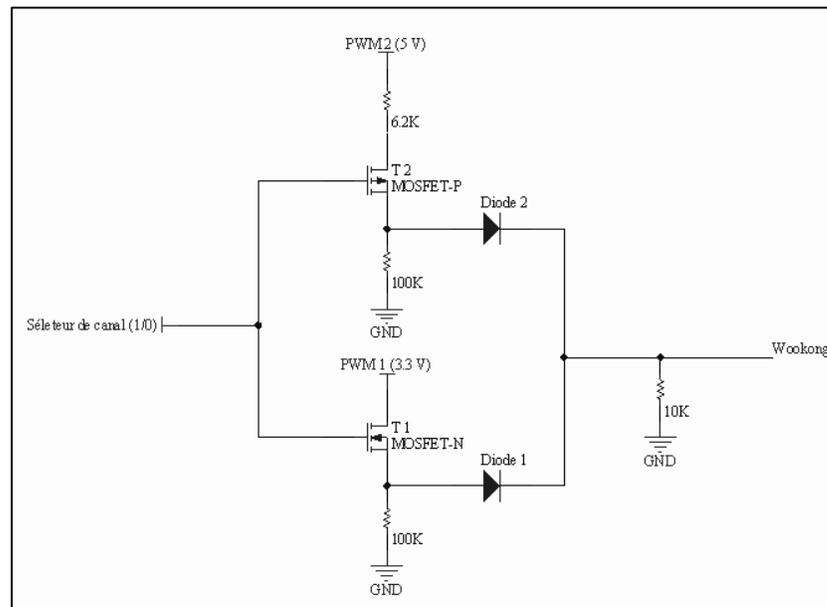


Figure-A V-1 Circuit du sélecteur de tension

La figure ci-dessus représente le circuit électrique réalisé pour un seul canal PWM, la source PWM 1 était raccordée à la télécommande et la source PWM 2 à la Raspberry.

Le fonctionnement du circuit et l'état des composants est décrit dans le tableau V-1 ci-dessous.

Tableau-A V-1 Fonctionnement du sélecteur de canaux

État sélecteur	Transistor 1	Transistor 2	Diode 1	Diode 2	Sortie
Niveau Haut	Passant	Bloqué	Passante	Saturé	PWM 1
Niveau Bas	Bloqué	Passant	Saturé	Passant	PWM2

En réalité, le Wookong a besoin de 5 canaux pour fonctionner (aileron, élévateur, gaz, lacet et mode de vol). Le circuit ci-dessus est donc réalisé 5 fois sur le même PCB pour répondre à nos besoins.

Étant donné que les commandes de vol de sécurités étaient sur la télécommande RC, nous avons trouvé logique que la commande de sélection de canal y soit aussi, cependant la télécommande n'est capable de générer que des signaux PWMs. Nous avons donc utilisé à nouveau un Arduino pour reconnaître le type de PWM envoyé par la télécommande et mettre le sélecteur de canal au niveau haut ou au niveau bas en fonction de la demande.

La structure matérielle de notre circuit de commande fut donc celui de la figure V-2 la suivante :

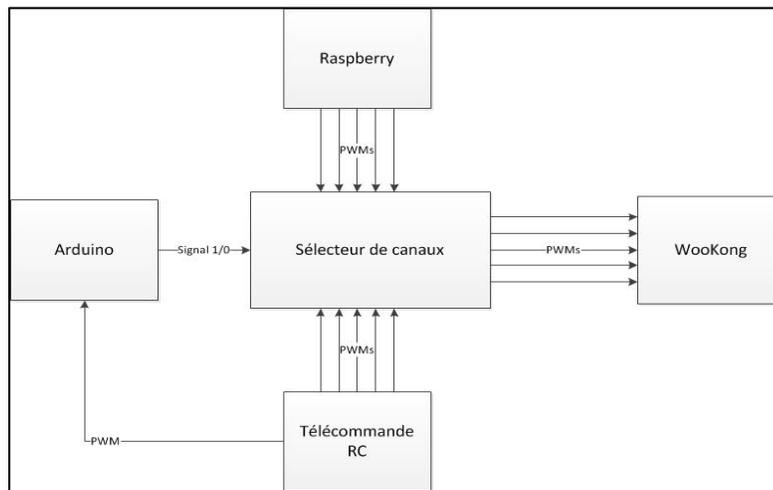


Figure-A V-2 Structure matérielle du sélecteur de canaux

Du fait de la spécificité de notre montage et du nombre peu élevé de circuits que nous étions susceptibles de réaliser, nous avons décidé de ne pas produire de PCB via un ordinateur et de nous contenter de souder les composants directement sur une plaque de prototypage.

Il nous a donc fallu dans un premier temps faire un schéma pour compresser les composants sur le plus petit espace possible et ensuite réaliser le circuit à la main (figure V-3) avec les 46 composants qui le compose.

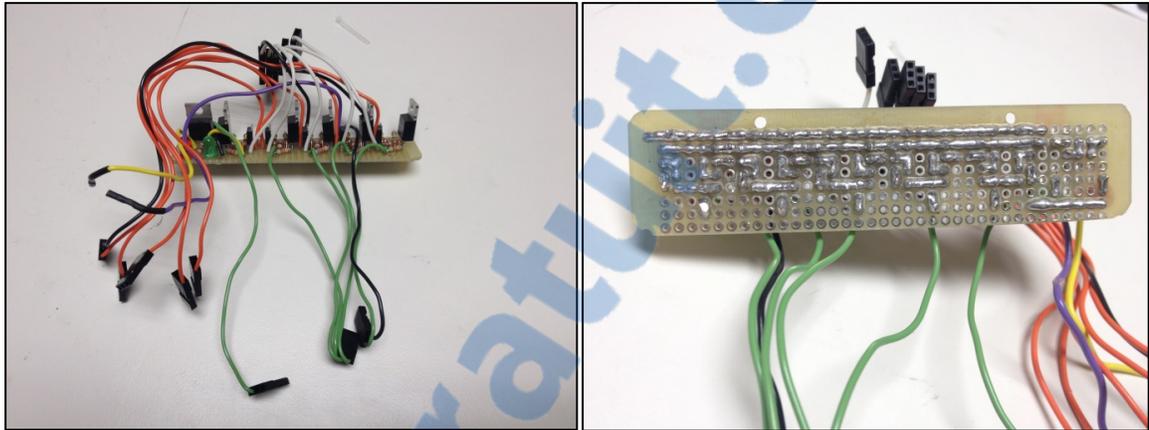


Figure-A V-3 Sélecteur de canaux (5 canaux)



## ANNEXE VI

### Diagrammes de la structure logicielle du drone

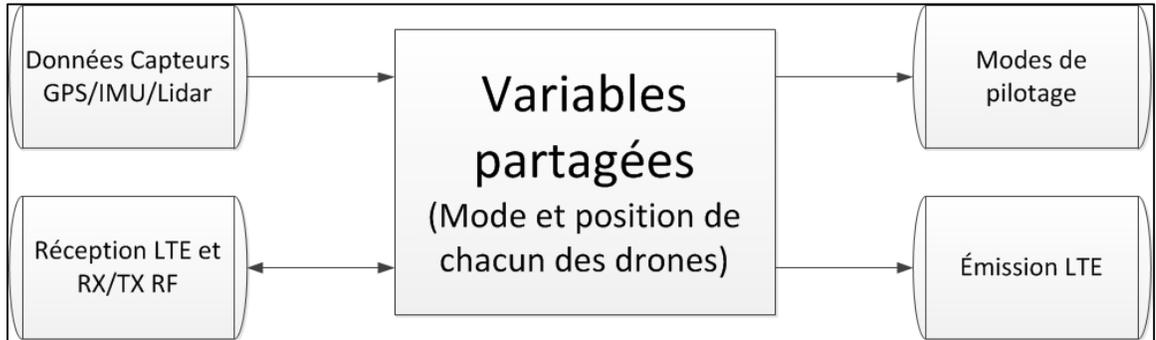


Figure-A VI-1 Architecture du partage de données entre applications

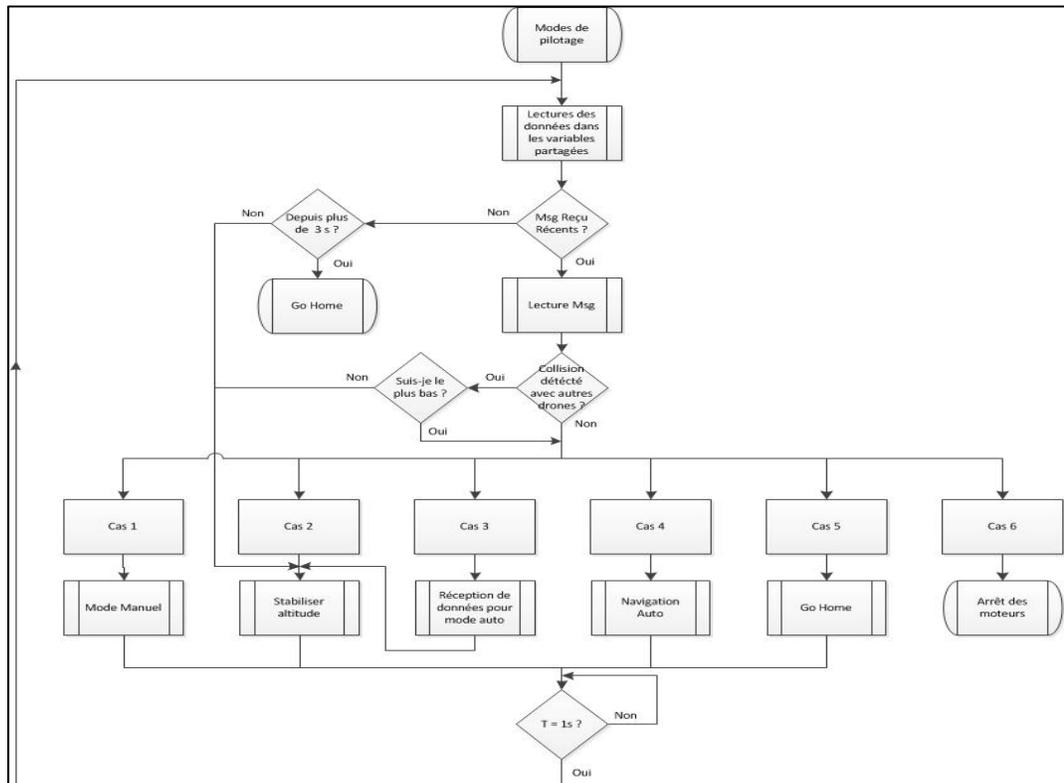


Figure-A VI-2 Structure des modes de pilotage et procédures de sécurité



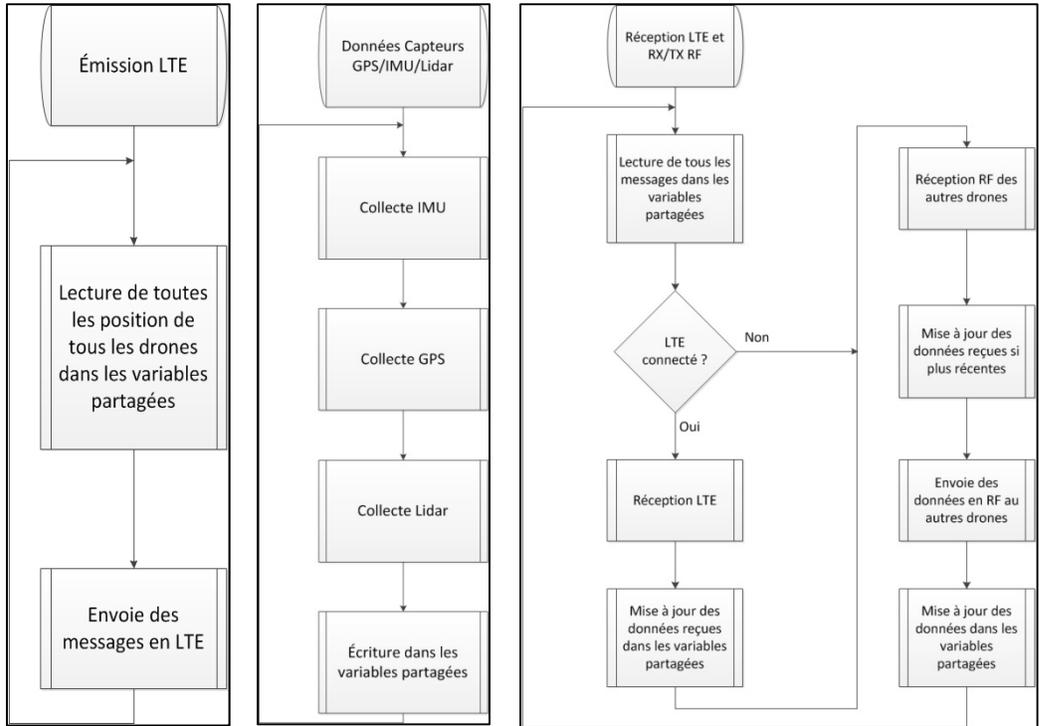


Figure-A VI-3 Structures des applications secondaires

## ANNEXE VII

### Système G1000 Basique 2 écrans

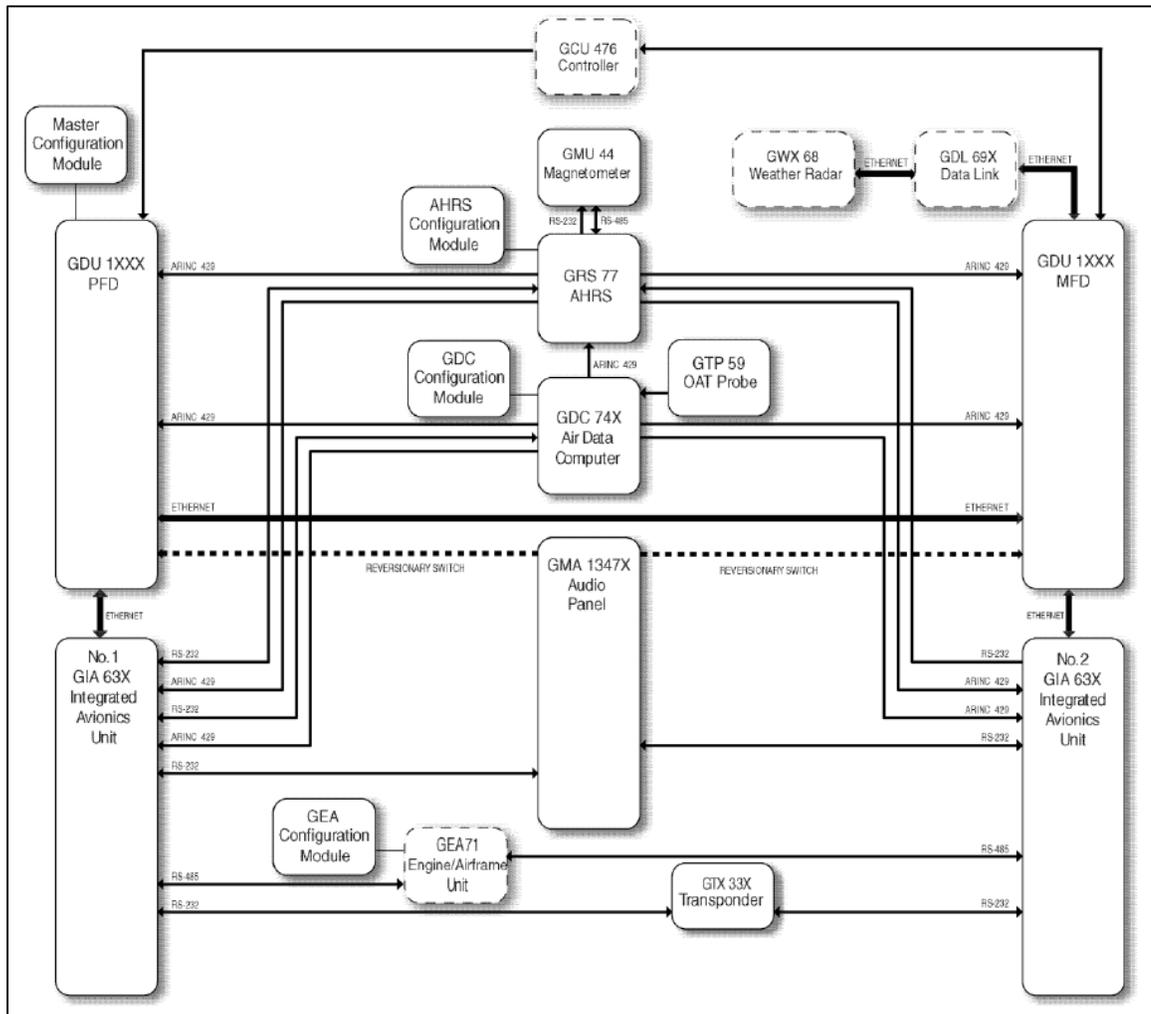


Figure-A VII-1 Système G1000 Basique 2 écrans



## ANNEXE VIII

### Création du banc pour GDUs

#### Support métallique :

Dans une documentation sommaire venant avec les GDU (Garmin, 2009, 2011, 2011), le constructeur Garmin suggérait un support composé de 9 tôles en aluminium de 2 mm pliées et découpées au laser.

L'ÉTS ne possédant pas d'équipements capables de découper et plier des plaques aussi épaisses, nous avons pris l'initiative de procéder autrement en créant notre propre support avec une plaque d'aluminium découpée à la main et une structure Metric® 80/20 capable de la supporter.

Le travail commença par une conception en 3D sous SolidWorks de la structure et de la plaque en aluminium afin de pouvoir par la suite découper les bonnes longueurs de tiges (figure VIII-1).

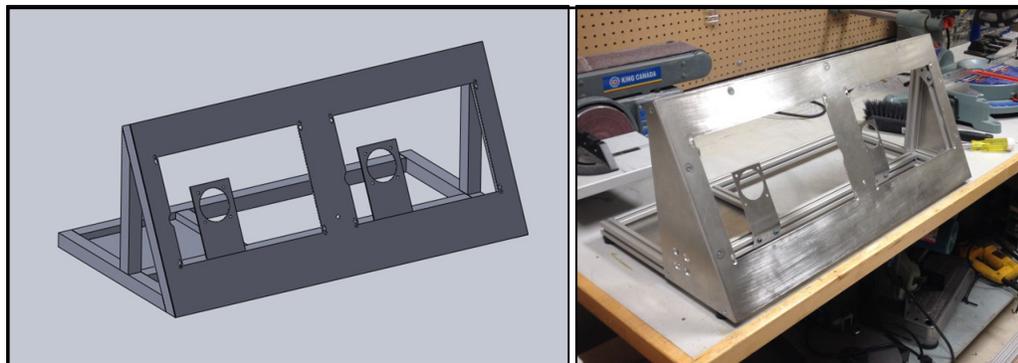


Figure-A VIII-1 Support complet 3D et support assemblé

Le support a finalement été peint en noir pour accueillir les deux GDUs, l'alimentation et les ventilateurs.

**Câblage électrique :**

Le support en métal réalisé, il nous fallut concevoir le câblage électrique des connecteurs 62 pins situés à l'arrière des GDU. Le seul schéma électrique donné par Garmin pour une configuration démo comme la nôtre étant très incomplet (figure VIII-2), il nous fallut modifier l'arrangement des pins pour l'adapter à notre configuration.

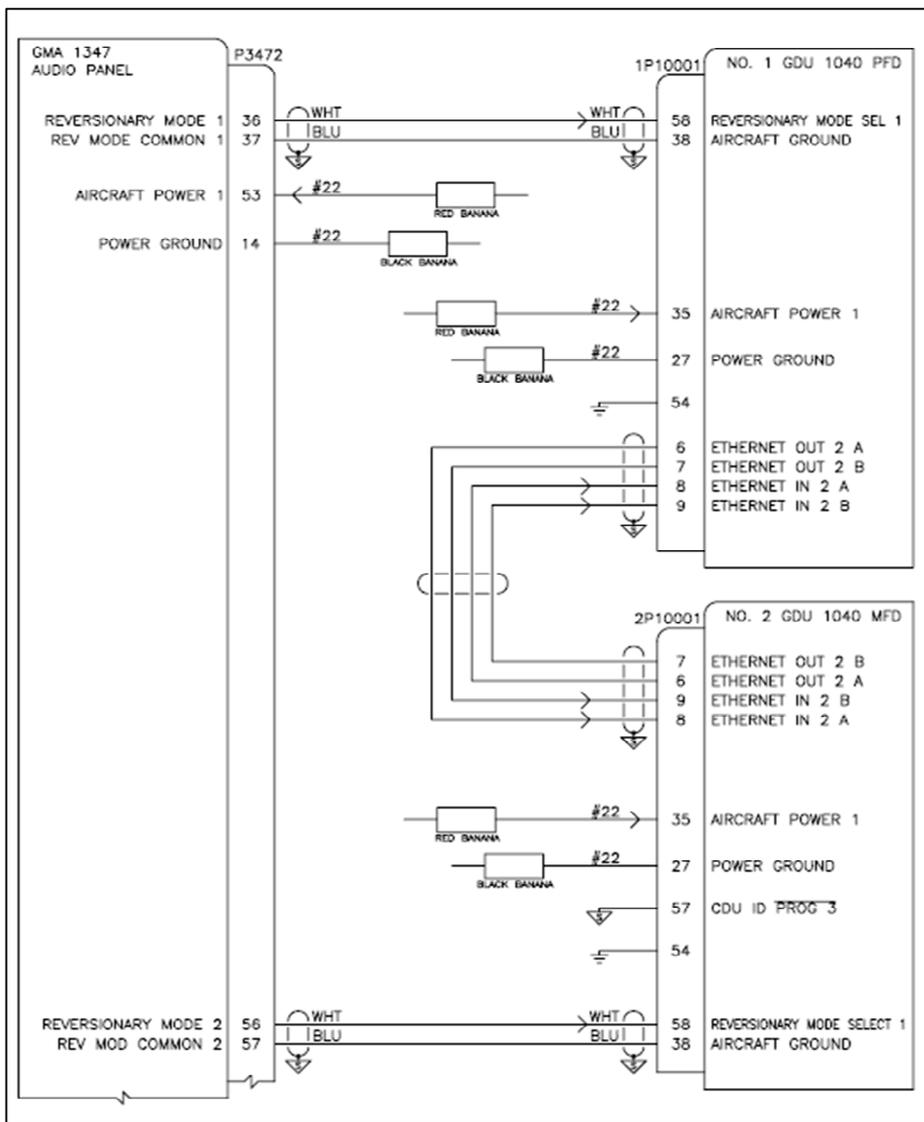


Figure-A VIII-2 Schema de montage démo Garmin

De plus, le constructeur ne mentionne aucun des éléments suivants dans ses schémas :

- Ampérages des fusibles;
- Logique du bouton poussoir pour les modes de sécurité (« reversionary modes »)
- Câblage des ventilateurs;
- Connection au module de sauvegarde des données;
- Connections aux autres ports HSDB pour un échange avec un autre équipement Garmin.

Pour parer au problème de l'ampérage, nous avons regardé dans la documentation technique d'installation qui mentionne l'ampérage maximum des GDU à 3.36 A. En nous basant dessus, nous avons choisi des fusibles à 4A.

Pour la logique du bouton poussoir, nous avons fait plusieurs essais qui nous ont permis de déterminer que lorsque le circuit est fermé le GDU fonctionne en mode normal (affichage normal PFD ou MFD) et lorsque le circuit est ouvert les GDU se mettent en mode de sécurité (même affichage sur les deux écrans). Le but de ce bouton poussoir dans un cockpit est de pouvoir afficher toutes ces informations de vol sur un seul écran au cas où l'un des écrans soit brisé ou dysfonctionne pendant le vol. Avec un peu de recul, on comprend également la logique de Garmin avec son câblage : lors d'un incident de vol, on assiste beaucoup plus souvent à une rupture de câble qu'à une connexion accidentelle. La compagnie a donc choisi de passer en mode de sécurité (« reversionary mode ») lorsqu'on assiste à une rupture de câble électrique.

Afin de pouvoir refroidir les GDUs pendant une longue période d'utilisation, nous avons installé des ventilateurs afin de les refroidir. Le circuit est assez simple et est composé de fusible 500 mA et d'un interrupteur pour pouvoir les arrêter, au besoin.

Les GDUs sont également équipés de modules de sauvegarde de configuration qui sont dissimulés dans les connecteurs 62 pines. Malheureusement, leur brochage ne nous a pas été transmis et il a fallu de longues heures de recherche dans les manuels d'entretiens et d'installations pour trouver les câblages adéquats.

Enfin et d'après le plan fournis, rien n'indique comment connecter un autre système à nos GDUs. Nous avons donc choisi de regarder la description de l'emplacement des ports HSDB

du GDU sur les différentes pines et de relier un connecteur 15 broches qui sert à faire du prototypage plus tard.

Finalement le schéma électrique global fut le suivant présenté en figure VIII-3 :

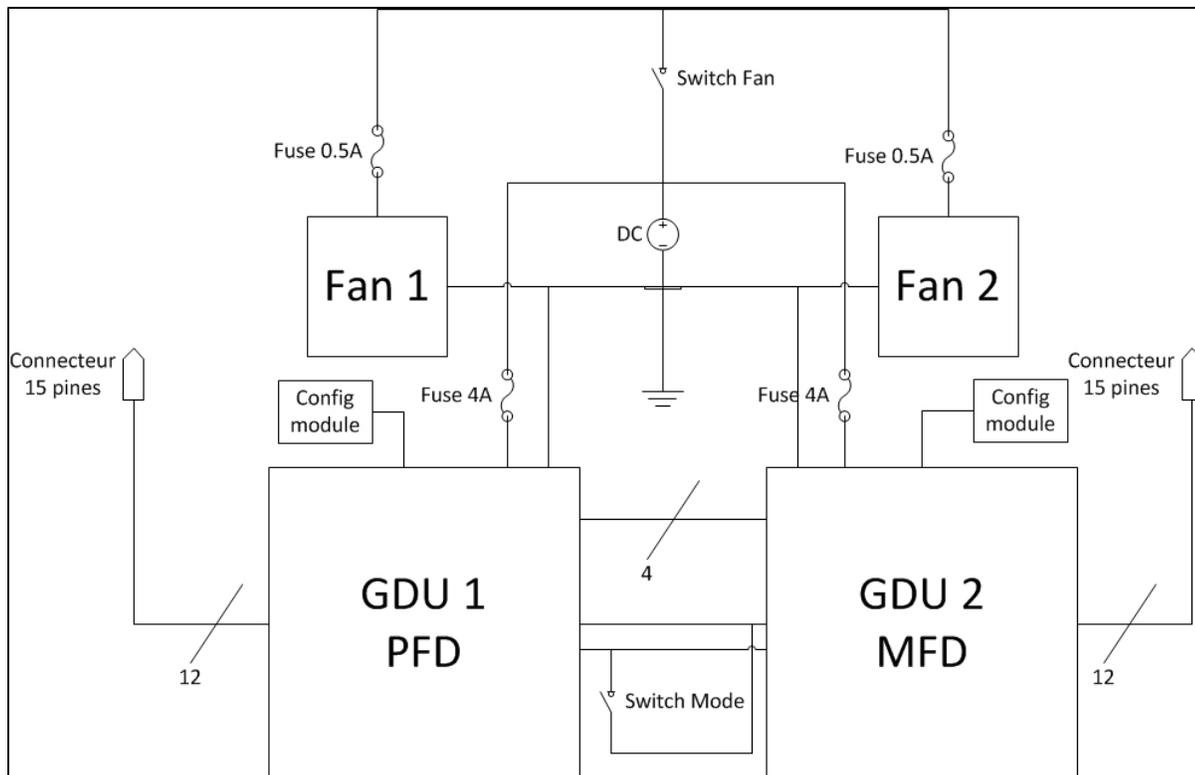


Figure-A VIII-3 Schéma électrique général des GDUs

Avec le brochage donné dans le tableau VIII-1 suivant :

Tableau-A VIII-1 Programme pine des connecteurs Garmin

<b>Pine PFD</b>	<b>Connection</b>	<b>Pine MFD</b>	<b>Connection</b>
<b>1</b>	Config module Ground	<b>1</b>	Config module Ground
<b>2</b>	Pine 1 connecteur 15 pines 1	<b>2</b>	Pine 1 connecteur 15 pines 2
<b>3</b>	Pine 2 connecteur 15 pines 1	<b>3</b>	Pine 2 connecteur 15 pines 2
<b>4</b>	Pine 3 connecteur 15 pines 1	<b>4</b>	Pine 3 connecteur 15 pines 2
<b>5</b>	Pine 4 connecteur 15 pines 1	<b>5</b>	Pine 4 connecteur 15 pines 2
<b>6</b>	MFD pine 8	<b>6</b>	PFD pine 8
<b>7</b>	MFD pine 9	<b>7</b>	PFD pine 9
<b>8</b>	MFD pine 6	<b>8</b>	PFD pine 6
<b>9</b>	MFD pine 7	<b>9</b>	PFD pine 7
<b>10</b>	Pine 5 connecteur 15 pines 1	<b>10</b>	Pine 5 connecteur 15 pines 2
<b>11</b>	Pine 6 connecteur 15 pines 1	<b>11</b>	Pine 6 connecteur 15 pines 2
<b>12</b>	Pine 7 connecteur 15 pines 1	<b>12</b>	Pine 7 connecteur 15 pines 2
<b>13</b>	Pine 8 connecteur 15 pines 1	<b>13</b>	Pine 8 connecteur 15 pines 2
<b>22</b>	Config Module Data	<b>22</b>	Config Module Data
<b>23</b>	Config Module Power Ground	<b>23</b>	Config Module Power Ground
<b>27</b>	Power Ground	<b>27</b>	Power Ground
<b>35</b>	Aircraft Power	<b>35</b>	Aircraft Power
<b>38</b>	Switch Mode 1	<b>38</b>	Switch Mode 1
<b>43</b>	Config Module Clock	<b>43</b>	Config Module Clock
<b>48</b>	Pine 9 connecteur 15 pines 1	<b>48</b>	Pine 9 connecteur 15 pines 2
<b>49</b>	Pine 10 connecteur 15 pines 1	<b>49</b>	Pine 10 connecteur 15 pines 2
<b>50</b>	Pine 11 connecteur 15 pines 1	<b>50</b>	Pine 11 connecteur 15 pines 2
<b>51</b>	Pine 12 connecteur 15 pines 1	<b>51</b>	Pine 12 connecteur 15 pines 2
<b>54</b>	Ground (Demo Mode)	<b>54</b>	Ground (Demo Mode)
<b>58</b>	Switch Mode 2	<b>57</b>	Ground (MDF Mode)
		<b>58</b>	Switch Mode 2



## LISTE DES RÉFÉRENCES BIBLIOGRAPHIQUES

- Akhrif, Ouassima. 2013. *Stabilité et Commandes de vol Fly-by-Wire*. [Notes du cours MGA804]. Montréal : École de Technologie Supérieure.
- AliExpress. 2015. « DJI Wookong M Waypoint (5 points)+ Data Link 900Mhz + Wookong M Autopilot System » En ligne <[http://img.alibaba.com/img/pb/001/439/546/546439001\\_148.jpg](http://img.alibaba.com/img/pb/001/439/546/546439001_148.jpg)>. Consulté le 10 janvier 2015.
- Austin, Ann. 2010. « Unmanned Aircraft Systems : UAV's Design, Development and Deployment ». Aerospace Series. John Wiley & Sons.
- Baillie, Stewart, Meredith Keith et Dave Roughthley. 2014. «Canadian Civil UAS 2014: An Update to the 2008 Report: "Canadian Market Opportunities for UAS: Non-Military Applications" ». En ligne < <http://ow.ly/JW50z> > Consulté le 20 avril 2014.
- Bennett, Frank. 2014. « LTE (réseaux mobiles) ». En ligne <[https://fr.wikipedia.org/wiki/LTE\\_\(r%C3%A9seaux\\_mobiles\)#/media/File:3GPP\\_Long\\_Term\\_Evolution\\_Country\\_Map.svg](https://fr.wikipedia.org/wiki/LTE_(r%C3%A9seaux_mobiles)#/media/File:3GPP_Long_Term_Evolution_Country_Map.svg)>. Consulté le 10 janvier 2015.
- Bento, Maria de Fátima. 2008. « Unmanned Aerial Vehicles: An Overview ». InsideGNSS, vol. 3, n°1, p. 54-61.
- Bensoussan, David. 2008. *Commande Moderne : approche par modèles continus et discrets*. TJ220.B46 2008. Montréal (Qc) : Presses internationales Polytechnique, 381 p.
- Biagiotti Luigi, Melchiorri Claudio. 2009. *Trajectory Planning for Automatic Machines and Robots*. Berlin Heidelberg : Springer-Verlag, 514 p.
- Bresciani, Tommaso. 2008. «Modelling, Identification and control of a Quadrotor Helicopter». Mémoire de maîtrise, Lund University.
- Capderou, Michel, 2003. *Satellites : Orbites et Mission*. 2-287-59772-7. Palaiseau : Springer-Verlag France. 481 p.
- Centre d'études stratégiques aérospatiales. 2006 . *Les drones aériens : passé, présent et avenir*, Coll. « Stratégie aérospatiale ». Paris : La Documentation française, 712 p.
- Cessna Aircraft Compagnie, 2012. *Service Bulletin*. SEB-34-03.Wichita USA :Cessna Customer Service, 16 p.

- Charland, Guillaume. 2014. « Contrôle non linéaire par backstepping d'un hélicoptère de type quadrotor pour de application autonomes ». Mémoire de maîtrise, Montréal, École de Technologie Supérieure, 191 p.
- ChRobotics. 2014. « UM7-LT Orientation Sensor ». En ligne <<http://www.chrobotics.com/>>, Consulté le 10 janvier 2015.
- Civil Aviation Authority (CAA) (U.K.). 2012. Unmanned Aircraft System Operations in UK Airspace – Guidance. CAP 722. Londre, Royaume-Uni: The Stationery Office, 110 p.
- Clubic. 2014. « Le Raspberry Pi Model B+ passe à 25 dollars ». En ligne<<http://img.clubic.com/0190000008041484-photo-raspberry-pi-model-b.jpg>> Consulté le 10 janvier 2015.
- Craig J. John. 2005. «Introduction to Robotics, Mechanics and Control». 3e edition. Pearson Prentice Hall.
- DeGarmo, M.T. 2004. *Issues Concerning Integration of Unmanned Aerial Vehicles in CivilAirspace*. En ligne. Coll. « MITRE Product », MP 04W0000323. McLean, VA, USA:MITRE Corporation (Center for Advanced Aviation System Development), 98 p.
- Department of Defense (U.S.). 1997. *Flying qualities of piloted aircraft*. MIL-STD-1797A.Washington DC, USA, 717 p.
- DeSantis Romano. 2011. «Systèmes de navigation aérienne». 7e édition. Note de cours ELE6209, Département de génie électrique, Polytechnique Montréal.
- Eurocontrol. 2012. « Challenge of growth 2013 ». En ligne. 12 p. <<http://www.eurocontrol.int/sites/default/files/publication/files/201207-challenges-of-growth-2013-task-1.pdf>> Consulté le 12 juin 2014.
- Federal Aviation Administration (FAA). 2014. «FAA Aerospace Forecast Fiscal Years 2014-2034», En ligne. 129 p. <[http://www.bv.transports.gouv.qc.ca/per/0965182/11\\_2014\\_2034.pdf](http://www.bv.transports.gouv.qc.ca/per/0965182/11_2014_2034.pdf)>. Consulté le 12 juin 2014.
- Garmin. 2009. *G900X/G950 : Installation and Maintenance Manual*. 190-00719-00 Revision D. Olathe USA : Garmin international, Inc, 681 p.
- Garmin. 2011. *G1000/G1000H : System Maintenance Manual*. 190-00907-00 Revision E. Olathe USA : Garmin international, Inc, 428 p.

- Garmin. 2011. *G1000: Integrated Flight Deck Pilot's Guide Cessna Nav III*. 190-00498-07 Revision A. Olathe USA : Garmin international, Inc, 632 p.
- GlobalStar. 2014. « GSP-1700 : Téléphone Satellite » En ligne <<http://ca.globalstar.com/images/products/gsp1700/gallery/image3.jpg>>. Consulté le 10 janvier 2015.
- Higham, Robin. 2006 . *Why Air Forces Fail: The Anatomy of Defeat* . UG625.W59. Lexington, Kentucky: The University Press of Kentucky, p 382).
- Hoffmann, Gabriel M., Steven L. Waslander et Claire J. Tomlin. 2004. «Quadrotor Helicopter Trajectory Tracking Control». American Institute of Aeronautics and Astronautics.
- Huawei. 2014. « E3276 ». En ligne <[http://consumer.huawei.com/ucmf/groups/public/documents/productspecification/~export/E3276-EN~3~GUI\\_DEV\\_EN\\_PHONES~SNIPPET\\_DC\\_TEMPLATE/390632-2.jpg](http://consumer.huawei.com/ucmf/groups/public/documents/productspecification/~export/E3276-EN~3~GUI_DEV_EN_PHONES~SNIPPET_DC_TEMPLATE/390632-2.jpg)>. Consulté le 10 janvier 2015.
- International Space University, (ISU). 2014. « Autonomous Mission for On-Orbit Servicing ».En ligne. 115 p. <[https://isulibrary.isunet.edu/opac/doc\\_num.php?explnum\\_id=659](https://isulibrary.isunet.edu/opac/doc_num.php?explnum_id=659)> Consulté le 12 février 2015.
- International Space University, (ISU). 2014. « Open innovation in space: Powering a new era ».En ligne. 119 p. <[https://isulibrary.isunet.edu/opac/doc\\_num.php?explnum\\_id=655](https://isulibrary.isunet.edu/opac/doc_num.php?explnum_id=655)> Consulté le 12 février 2015.
- Jarry, Maud. 2010. *Les Armes V1 et V2 et les Français*, OCLC 560010497. Rennes: Marine Editions, 190 p.
- Kim, H. J. and Shim, D. H. 2003. «A flight control system for aerial robots: Algorithms and experiments». *Control Engineering Practice*, 11(2), 13891400.
- Landry, René Jr. 2014. *Navigation aérienne, GNSS et systèmes inertiels embarqués*. [Notes du cours MGA852]. Montréal : École de Technologie Supérieure.
- Landry, René Jr. 2014. « Mission automatisée de maintenance en orbite des satellites (AMOOS) » . En ligne. < <http://substance.etsmtl.ca/mission-automatisee-de-maintenance-en-orbite-des-satellites-amoos/>>. Consulté le 12 juin 2014.
- Landry, R.Jr., Vigneron, A., Andrade., E.Jr., and al., «Autonomous Mission for On-Orbit Servicing (AMOOS)», 65th International Astronautical Congress, Toronto, Canada, Sept. 29 - Oct. 3, 2014.

- LASSENA, 2013, «Mission automatisée de maintenance sur site (AMOSS) ».In *Site du laboratoire LASSENA*. En ligne < <http://lassena.etsmtl.ca/spip.php?rubrique29#495> >. Consulté le 4 avril 2015.
- LASSENA, 2013, «Mission automatisée de Maintenance en Orbite des Satellites (AMOOS) ».In *Site du laboratoire LASSENA*. En ligne < <http://lassena.etsmtl.ca/spip.php?rubrique29#482> >. Consulté le 4 avril 2015.
- Leishman J. Gordon. 2006. «Principles of Helicopter Aerodynamics». Second editions, number 12 in Cambridge Aerospace Series, Cambridge Press, 497 p.
- Markus Achtelik, Abraham Bachrach, Ruijie He, Samuel Prentice, Nicholas Roy. 2009. « Autonomous Navigation and Exploration of a Quadrotor Helicopter in GPS-denied Indoor Environments ». First Symposium on Indoor Flight Issues. International Aerial Robotics Competition.
- National Imagery and Mapping Agency, (NIMA). 2000. « Departement of World Geodetic System 1984 ». En ligne. 175 p. < <http://earth-info.nga.mil/GandG/publications/tr8350.2/wgs84fin.pdf> >. Consulté de 20 février 2015.
- Nelson, Robert. 1997. *Flight Stability and Automatic Control*. Second edition. Université du Michigan : McGraw-Hill Ryerson, Limited, 284 p.
- Newcome R. Laurence. 2004. *Unmanned Aviation : A Brief History of Unmanned Aerial Vehicles*. AIAA Technology & Engineering. 164 p.
- Office of the Secretary of Defense (U.S.). 2005. « Unmanned Aircraft Systems Roadmap 2005-2030 ». En ligne. Department of Defense (U.S.), 213 p . <[http://www.fas.org/irp/program/collect/uav\\_roadmap2005.pdf](http://www.fas.org/irp/program/collect/uav_roadmap2005.pdf)>.
- Ohleger, Mike, Geoffrey G. Xie et John H. Gibson. 2013. « Extending UAV Video Dissemination via Seamless Handover: A Proof of Concept Evaluation of the IEEE 802.21 Standard ». In 46th Hawaii International Conference on System Sciences. (Grand Wailea Maui Hawaii, January 7 - 10 2013), p. 5104-5112. Monterey, California : Operational & Information Sciences Naval Postgraduate School Publishers.
- Organisation de l'aviation civile internationale, (OACI). 2011. *Unmanned Aircraft Systems (UAS)*. Cir 328 AN/190. Montréal (Qc) : OACI, 38 p.
- Pounds, P., Robert Mahony et Peter Corke. 2010. «Modelling and control of a quad-rotor robot». En ligne.10 p. < <http://core.ac.uk/download/pdf/10898458.pdf> >. Consulté le 23 novembre 2014.

- Rockwell International Corporation. 1976. « RPV Flying Qualities Design Criteria » . En ligne. AFFDL-TR-76-T25, 213 p.  
<<http://www.dtic.mil/cgibin/GetTRDoc?Location=U2&doc=GetTRDoc.pdf&AD=ADA045170>>.
- Transport Canada. 2014. *Certificat d'opérations aériennes spécialisées pour l'utilisation d'un système de véhicule aérien non habité (UAV)*. Deuxième édition, Norme de TC : IP 623-001. Montréal : Transport Canada, 139 p.
- Sarris, Zak. 2001. « Survey of UAV applications in civil markets ». In 9th Mediterranean Conference on Control and Automation. (Dubrovnik, Croatie, 27-29 juin, 2001), p. 1-11. < [http://med.ee.nd.edu/MED9/Papers/Aerial\\_vehicles/med01-164.pdf](http://med.ee.nd.edu/MED9/Papers/Aerial_vehicles/med01-164.pdf) >. Consulté le 10 janvier 2015.
- Sivolella, Davide. 2014. *To Orbit and Back Again: How the Space Shuttle Flew in Space*, Première Edition. Springer-Verlag New York : Springer Science+Business Media New York, 502 p.
- Waslander, L. Steven, Gabriel M. Hoffmann, Jung Soon Jang et Claire J. Tomlin. 2005. «Multi-Agent Quadrotor Testbed Control Design : Integral Sliding Mode vs Reinforcement Learning».165 p.
- Walkeraonline. 2015. « WALKERA (WALKERA-QR-X800) QR X800 GPS Brushless Quadcopter - 2.4GHz ». En ligne <<http://www.walkeraonline.com/walkera-ufo-quadcopter/walkera-qr-x800-gps-quadcopter.html>>. Consulté le 10 janvier 2015.
- Williams, W. 2003. « UAV Handling Qualities.....You Must be Joking ». En ligne. Elizabeth, SA, Australie: Aerospace Sciences Corporation Pty. Ltd., 7 p.