

École polytechnique de Louvain

Développement et validation d'un système de guidage autonome pour l'assemblage d'éléments drone-compatibles



Auteur: **Mathieu MONNART**
Promoteurs: **Pierre LATTEUR, Ramin SADRE**
Lecteurs: **Sébastien GOESSENS, Benoît RAUCENT**
Année académique 2018–2019
Master [120]: ingénieur civil des constructions

Résumé

Depuis plusieurs années, la technologie des drones se développe dans de nombreux domaines, celui de la construction n'y échappe pas. C'est dans cette optique que Pierre Latteur, professeur à l'UCLouvain, développe un projet de recherche visant à créer un système de construction autonome par drone. Les premières années de recherche ont permis de développer des éléments de construction compatibles avec les drones, des systèmes de levage pour ces éléments et des systèmes de positionnement, dans le but de connaître la position du drone. Ce travail présente tous les éléments nécessaires pour concevoir un système de construction autonome par drone, en passant des composants hardware aux systèmes de positionnement, de guidage et de levage. Il retrace aussi les essais effectués qui nous ont permis de développer ce système complet.

Remerciements

Je tiens particulièrement à remercier les personnes suivantes :

- **Pierre Latteur**, professeur à l'UCLouvain et promoteur de ce projet de recherche, pour ses idées, ses conseils et son temps accordé même à l'autre bout du monde ;
- **Sébastien Goessens**, assistant à l'UCLouvain réalisant une thèse sur le sujet de la construction par drone, pour sa disponibilité tout au long de l'année, ses suggestions et la mise à disposition de tout le matériel dont une imprimante 3D ;
- L'équipe du Laboratoire Essais mécaniques, Structures et Génie civil, et plus particulièrement **Alex** et **Samuel** pour leur aide et leurs conseils en électronique ;
- **Julien**, **Marie-Marie** et **Nicolas** pour la bonne ambiance de travail et leur persévérance dans ce projet ;
- **Ma famille** et **mes amis**, pour leur soutien tout au long de mon cursus à l'UCLouvain.

Table des matières

Introduction	5
1 Description du système autonome de construction	12
1.1 Fonctionnement d'un drone et éléments le constituant	12
1.2 L'ordinateur embarqué	18
1.3 Contrôleur de vol	21
1.4 Système complet	23
2 Les systèmes de positionnement	25
2.1 Global et local	25
2.2 Station totale	26
2.3 Balise UWB	32
2.4 Caméra Realsense	36
2.5 Comparaison des systèmes	39
2.6 Algorithme de positionnement	40
2.7 Mise en place du système	41
3 Système de guidage	45
3.1 Définir une mission	45
3.2 Contrôler le drone	48
3.3 Sécurité	50
4 Système de levage	51
4.1 Choix du système de levage	51
4.2 Fonctionnement du système	54
4.3 Quel type d'électro-aimant choisir ?	55
4.4 Automatisation du fonctionnement du système	57
4.5 Guide de levage du drone	61
4.6 Support	62

5	Essais réalisés	64
5.1	Tests manuels	64
5.2	Tests des systèmes de positionnement	65
5.3	Tests de vol autonome	70
5.4	Tests de levage de bloc	75
	Conclusion	79
	Annexes	82
A	Offsets	82
B	Fiches descriptives du cône de test et son guide	84
C	Modèles 3D	91

Introduction

Contexte

Depuis de nombreuses années, les domaines de l'ingénierie ne cessent d'automatiser et d'améliorer leur efficience par le développement de technologies telles que les lignes de production, les robots ou l'intelligence artificielle.

Les drones font partie de ces technologies et sont utilisés de plus en plus fréquemment dans de nombreux secteurs, comme celui de la protection civile où ils sont utilisés pour l'observation de manifestations, ou celui du cinéma où ils facilitent les prises de vues aériennes, ou encore celui de la livraison de colis.

Les possibilités d'application de ces nouveaux outils ultramodulables (car légers, peu couteux, très flexibles dans leur utilisation, et capables de porter une charge plus lourde que leur poids propre) peuvent encore être développées dans un grand nombre de domaines de l'ingénierie.

Paradoxalement, le secteur de la construction ne fait que peu appel à ce type de technologies alors que la conception et la réalisation d'un projet s'en trouveraient améliorées.

Actuellement, pour la conception d'un projet de construction, les acteurs du milieu travaillent encore en grande partie sur des plans et des esquisses papiers, parfois annotés manuellement, ce qui peut amener à des erreurs, une mauvaise communication, et des pertes de temps et d'argent. En 2014, une étude de l'ADEB (Association Des Entrepreneurs de Belgique) a démontré que ces types de problèmes constituent de 5 à 13% du budget total d'un projet. Pour pallier les difficultés rencontrées au moment de la conception, de plus en plus de projets et d'entreprises se tournent vers l'automatisation et s'intéressent notamment au BIM. Ce nouvel outil est à la fois un processus d'échange de données et de contrôle des informations relatives à un projet (*Building Information Management*), et est aussi un modèle de conception 3D qui permet une meilleure collaboration entre les acteurs d'un projet et une meilleure coordination sur chantier (*Building Information Modeling*).^[9]

Quant à l'automatisation de la phase de réalisation, elle reste peu développée alors qu'elle permettrait de réduire les coûts de la main d'œuvre, d'assurer une meilleure sécurité sur chantier et d'éviter trop de manutention humaine. Même

si des projets de recherche (tels que la conception d'imprimantes 3D macroscopiques - ayant pour but d'imprimer les éléments structurels d'une maison -, ou d'éléments préfabriqués de plus en plus complets - maison en kit) participent à cette automatisation, ils manquent de flexibilité et peuvent s'avérer trop contraignants.

C'est sur la base de ces observations qu'en 2015 l'Université Catholique de Louvain-la-Neuve a lancé un projet de recherche visant à développer la construction autonome par drone. Ce projet est supervisé par Pierre Latteur, Professeur de Structures à l'UCL, et Sébastien Goessens, chercheur en systèmes constructifs drone-compatibles.

Projet de Recherche

Notre travail s'inscrit dans le projet de recherche portant sur l'utilisation de drones dans le secteur de la construction lancé depuis quelques années à L'UCLouvain. L'ensemble des points suivants retracent les différentes étapes de l'évolution de ce projet.

Éléments drone-compatibles

Intéressons nous aux éléments de construction pour commencer.^[8] En effet, deux contraintes sont importantes à prendre en compte quand il s'agit de construire à l'aide de drones. Premièrement, l'imprécision du drone. Celle-ci peut être due soit à des phénomènes extérieurs au système, comme le vent, ou soit à l'imprécision des systèmes de positionnement utilisés. De nouveaux types de blocs comme les driks (compression de "drone" et "brick") ont été développés pour pallier cette première contrainte. Leur forme permet de corriger une certaine imprécision lors de leur emboîtement.

La deuxième contrainte est celle du poids des éléments. On conçoit facilement qu'un drone est limité par la charge qu'il peut transporter. Si un engin volant venait à devoir transporter des charges de plus de 100 kg, il serait davantage considéré comme un mini-hélicoptère que comme un drone. Cette seconde contrainte force à revoir le processus de conception des bâtiments construits par drone. Ceux-ci devront être composés uniquement d'éléments petits et légers.

Les éléments de construction qui respectent ces deux contraintes sont appelés "drone-compatibles".

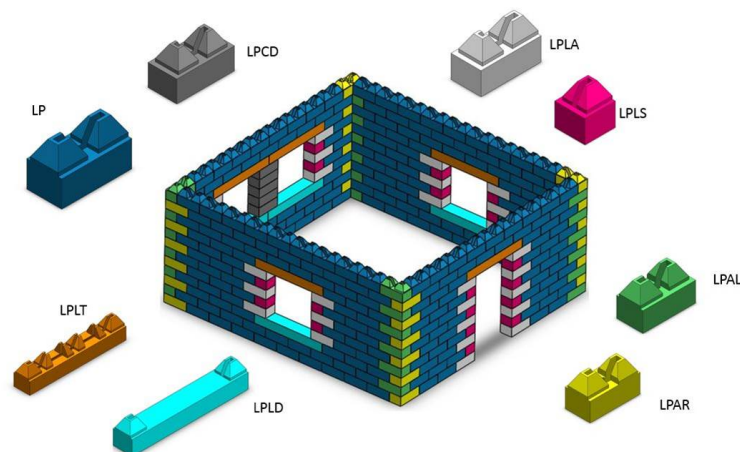


Figure 1 – Représentation de dricks ainsi que d’un bâtiment composé uniquement d’éléments drone-compatibles.

Élaboration d’une méthode de construction par drones

Le déroulement d’un projet de construction est différent quand il s’agit de construire à l’aide de drones. Le projet se compose de 4 étapes indispensables, illustrées à la figure 2.

1. Conception et dimensionnement du projet : cette étape est commune à tous les projets de construction, elle est réalisée par un architecte, un bureau d’études et le client. Cette étape se conclut par la création d’une maquette 3D BIM du résultat final.
2. Tous les éléments structuraux de cette maquette 3D sont convertis en éléments drone-compatibles.
3. Une liste d’instructions drone-compatibles, est créée pour par rapport à la maquette constituée.
4. Cette liste d’instructions est exécutée par le drone sur le terrain.

Essais manuels

Après le développement de nombreux éléments drone-compatibles, principalement axés sur les dricks, des essais ont été réalisés. Le but de ces tests, réalisés manuellement avec un pilote professionnel, était de prouver la faisabilité de la construction par drone avec ces éléments. Ils furent concluants car de petits murs et colonnes (avec d’autres éléments que les dricks) ont pu être construits. Avec ces résultats encourageants, l’équipe de recherche a décidé de continuer sur cette voie

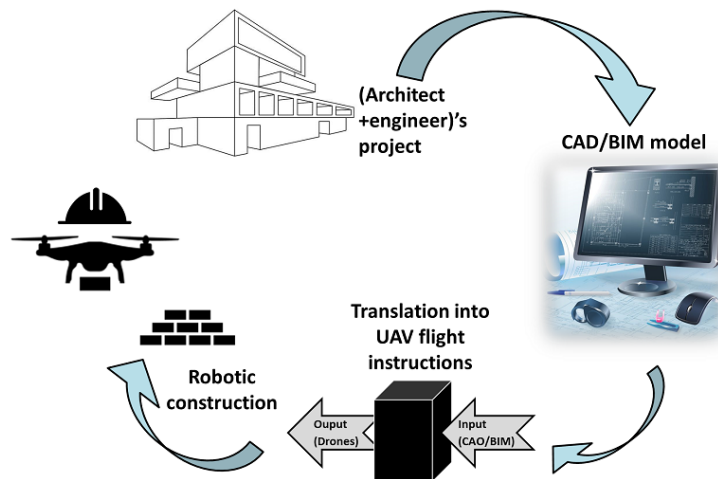


Figure 2 – Processus d’élaboration d’un projet réalisé avec des drones.

en développant cette fois un système sans intervention humaine (représentée par le pilote dans la phase test) : un système autonome.



Figure 3 – Essai manuel réalisé à l’UCLouvain.

Système de positionnement

Dans le but de développer un système de construction autonome, la première étape a été de développer des systèmes de positionnement, afin de connaître la position du drone.

Station totale

L'année passée, lors de leur TFE, Nicolas Sorensen et Rémy Vermeiren, deux étudiants de la section ingénieur civil en informatique, ont développé un système de positionnement utilisant une station totale.[10] Le système fonctionne à l'aide de deux éléments principaux : une station totale TCRP1203 de la firme *Leica* et un prisme embarqué sur le drone. La station totale est un élément mécanique qui permet de calculer des positions à l'aide d'un laser. Ce laser est émis par la station totale et réalise un aller-retour jusqu'à un point ciblé. Le temps de parcours du laser permet de connaître la distance à ce point. L'utilisation d'un prisme permet à la station totale de suivre en continu un élément auquel le prisme est attaché, en l'occurrence le drone. Le programme qu'ils ont créé permet premièrement de trouver le prisme et, dans un deuxième temps, de récupérer les valeurs de position de ce prisme et donc celles du drone. Le montage est assez simple comme le montre la figure 4. L'avantage de cette solution est sa précision de l'ordre de 5 mm. Son plus gros défaut découle directement des capacités mécaniques de la station. En effet, ces dernières limitent la précision de la station totale par une vitesse de rotation trop faible, ce qui entraîne une perte de signal dans le cas où le drone vole trop vite à une distance proche de la station.

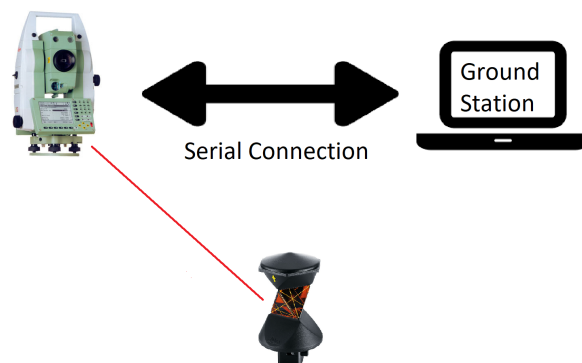


Figure 4 – Schéma du système station totale.

Système UWB

Le système de positionnement UWB pour Ultra Wide Band (Ultra Large Bande) a été développé l'année passée par Zakariya Bouhaddi, étudiant en section ingénieur industriel en électronique à l'ECAM.[6] Il consiste à utiliser des ancrs fixes et une balise embarquée sur le drone. Cette dernière envoie un signal radio ultra large bande que réceptionnent les ancrs. Une fois ce signal reçu, une onde retour est alors envoyée par ces ancrs et la balise peut trianguler sa position en connaissant le temps de propagation de ces ondes, comme le montre la figure 5. Ce système

est complémentaire du précédent car sa faible précision de l'ordre de la dizaine de centimètres est compensée par sa bonne fiabilité.

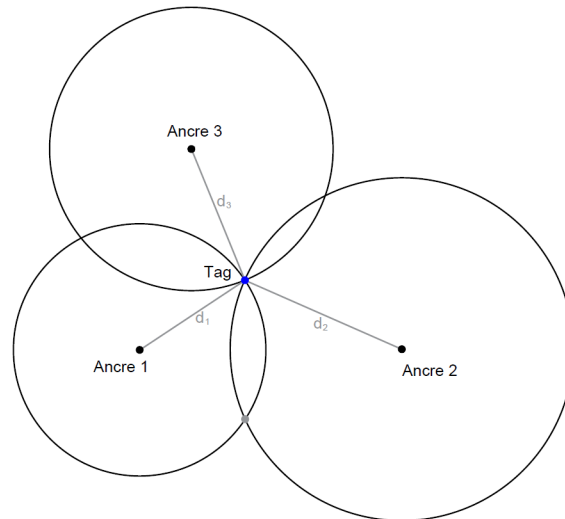


Figure 5 – Schéma de tringulation de la balise par les ancras UWB.

Objectifs

C'est en rapport avec le contexte décrit ci-dessus et sur la base des avancées déjà réalisées que notre travail s'est appuyé.

L'objectif principal de notre mémoire est de s'inscrire dans la continuité et de développer un système de vol et de levage pour drone rendant celui-ci capable d'exécuter une mission de construction de manière autonome. J'entends par système autonome le fait qu'un drone puisse rallier une position définie par ordinateur, sans intervention humaine extérieure. Pour cela, il a fallu décomposer cet objectif en plusieurs points.

- Récupérer les données de vol du drone : niveau de batterie, position, mode de vol, ...
- Placer les systèmes de positionnement dans un repère unique.
- Faire décoller le drone et lui faire réaliser un vol stationnaire de manière autonome.
- Faire voler le drone de point en point de manière autonome.
- Développer un système de levage activable depuis l'ordinateur embarqué sur le drone.

- Prendre un bloc avec le drone de manière autonome.
- Déplacer le drone avec un bloc et le poser à une position définie.

L'ensemble de ce travail a été réalisé avec 3 étudiants d'ingénieur civil en informatique à l'UCLouvain, Julien Amalaberque, Marie-Marie van der Beek et Nicolas Vanvyve. Leur travail a porté sur la partie informatique du projet. [5].

Chapitre 1

Description du système autonome de construction

Ce chapitre décrit les différents composants qui interviennent dans la construction de manière automatisée à l'aide d'un drone. Il aborde la façon dont ceux-ci communiquent entre eux et l'objectif de ces différents échanges. Cet ensemble est composé principalement d'un drone et de tous ses composants embarqués : différents systèmes de positionnement et un système de levage de bloc. En plus des systèmes embarqués, il y a une station de contrôle au sol dédiée à l'exécution et à la surveillance du bon déroulement des processus par un opérateur humain. Tous ces éléments sont présentés dans ce chapitre ou les suivants.

1.1 Fonctionnement d'un drone et éléments le constituant

Cette section est consacrée à l'explication du fonctionnement d'un drone et des éléments nécessaires pour le faire voler, ainsi que des éléments ajoutés dans le but de développer un système de construction autonome.

1.1.1 Les 3 drones de l'UCLouvain

Premièrement, l'équipe de recherche possède 3 drones de tailles, de constitutions et d'usages différents. Ils sont tous 3 présentés ci-dessous.

Grand drone

Ce modèle de drone est le prototype à taille réelle utilisé pour le transport et le placement de blocs de maçonnerie. Il peut transporter une charge allant jusqu'à 30

kg. Il est construit sur un châssis personnalisé composé d'éléments en carbone. Il est uniquement utilisé si, et seulement si, le système testé est robuste et fiable.



Figure 1.1 – Grand drone de l'UCLouvain.

Moyen drone

Le moyen drone est l'engin intermédiaire entre le prototype à taille réelle (grand drone) et le drone de tests (petit drone). Il est utilisé pour tester à une échelle plus réaliste les systèmes en développement déjà validés sur le petit drone. Sa conception lui permet de transporter des éléments de 5 à 10 kg. Il se situe à mi-chemin entre les deux autres drones.



Figure 1.2 – Moyen drone de l'UCLouvain.

Petit drone

Ce dernier modèle est construit sur la base du châssis F550 de la firme *DJI*¹, cet engin est capable de porter une charge de l'ordre du kilo. Le faible coût que

1. <https://www.dji.com/be/flame-wheel-arf/feature>

représente ce drone (comparé aux deux autres) en fait un drone de tests facile à reconstruire en cas de destruction d'éléments de sa structure. Son architecture nous permet de moduler ses composants à notre guise. Cela est très utile et important pour tester de nouvelles solutions. À l'exception de sa base, dans laquelle est insérée un circuit imprimé, son châssis est entièrement imprimable en 3D. Cela est pratique si l'on doit remplacer un élément.



Figure 1.3 – Petit drone de tests de l’UCLouvain.

1.1.2 Éléments indispensables au fonctionnement d’un drone

Le schéma de la figure 1.6 décrit l’agencement de tous les éléments essentiels au fonctionnement d’un drone. Leurs fonctions sont toutes décrites ci-dessous.

- **Contrôleur de vol** : c’est le cerveau du drone. Il reçoit les instructions de pilotage et les transmet aux moteurs par le biais des ESCs. Son fonctionnement est décrit plus précisément au point 1.3.
- **ESC (Electronic Speed Controller)** : il s’agit du contrôleur de vitesse électronique. Les informations de puissance à envoyer aux moteurs lui sont fournies par le contrôleur de vol. Les ESCs reliés à la batterie, ajustent alors le courant envoyé aux moteurs pour les faire tourner plus ou moins vite selon les instructions du contrôleur de vol.
- **Moteurs et hélices** : ces éléments sont indispensables au fonctionnement d’un drone. Ils peuvent être 4, 6 ou 8 dans les cas les plus courants en fonction de l’architecture voulue du drone. Un moteur sur deux tourne dans un sens horlogique et les autres tournent dans le sens anti-horlogique. Cela a pour objectif de contrer le couple induit par la rotation de chaque hélices et d’éviter

que le drone ne tourne sur lui-même. Les moteurs utilisés sur un drone sont des moteurs sans balai, préférés à des moteurs avec balai pour leur rapport poids/puissance plus élevé et leur moindre nuisance sonore.

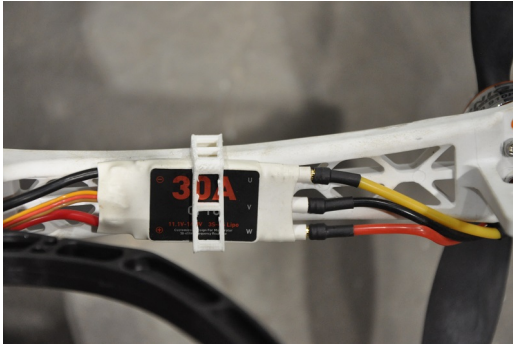


Figure 1.4 – ESC sur le petit drone.



Figure 1.5 – Moteur sur le petit drone.

- **Télécommandes et récepteur** : c'est le dispositif utilisé pour contrôler le drone manuellement. Les télécommandes sont définies par le nombre de canaux qu'elles possèdent. Chaque canal représente une variable que la commande peut faire varier. Le récepteur reçoit les informations de chaque canal et les transmet au contrôleur de vol.
- **Batterie** : elle alimente le contrôleur de vol et les ESCs. Des batteries LiPo (Lithium Polymère) sont utilisées sur les drones pour leur capacité à délivrer beaucoup d'énergie en comparaison avec d'autres types de batteries (Ni-Cd, Lithium, etc.) Cependant, elles sont plus fragiles, possèdent un long temps de recharge et ont une durée de vie relativement faible (300 cycles de chargement)².
- **Châssis** : il est composé de tous les éléments structurels du drone. Les pieds, la base et les bras. Il permet de transporter tout le matériel nécessaire au vol du drone. Son architecture est variable en fonction du nombre de moteurs voulus. Un circuit imprimé permettant de transmettre le courant de la batterie aux ESCs est parfois intégré à la base.
- **Switch de sécurité** : c'est un interrupteur hardware qu'il est nécessaire d'activer à chaque utilisation. Il bloque le système au démarrage, son activation est obligatoire pour pouvoir armer les moteurs et donc décoller.

2. <https://www.amateursdedrones.fr/batteries-pour-drones/>

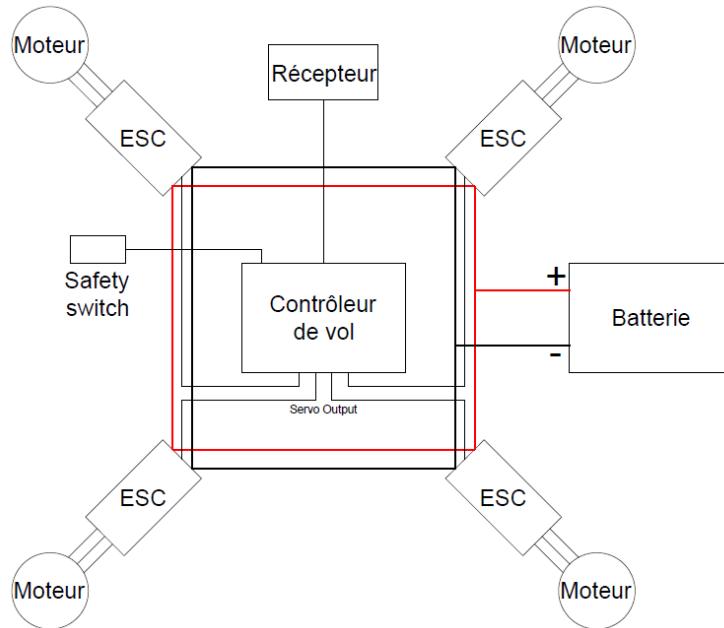


Figure 1.6 – Schéma d'un drone classique.

1.1.3 Éléments supplémentaires pour pouvoir construire de manière autonome

Sur la figure 1.7 est représenté le schéma de composition d'un drone destiné à construire de manière autonome. Les éléments supplémentaires à ceux d'un drone classique sont définis ci-dessous. Logiquement le récepteur de la radio-commande ne fait plus partie de ce schéma.

- **Ordinateur embarqué** : c'est l'élément central de l'ensemble. Il fait le lien entre les systèmes de positionnement, le système de vol et celui de levage de bloc de construction. L'ordinateur embarqué et ses fonctions sont décrits à la section 1.2.
- **Éléments de positionnement** : regroupe tous les éléments liés aux différents systèmes de positionnement. Dans notre cas, un prisme, une balise UWB et une caméra Realsense. Tous ces composants sont décrits dans le chapitre 2, relatif au système de positionnement.
- **Télémétrie** : ce dispositif de communication sans fil permet de recevoir des données de la part du contrôleur de vol. Ces données peuvent être de tous types : comme le niveau de charge de la batterie, la position du drone, ses avertissements, etc. Son utilisation est décrite au point 1.3.

- **Station de contrôle au sol** : c'est un ordinateur qui reste au sol avec lequel l'opérateur humain va pouvoir exécuter les différents programmes pour lancer la mission de construction. Cet ordinateur permet aussi un contrôle de la part de l'opérateur humain sur la mission en cours. En effet, beaucoup d'informations sont transmises à la station de contrôle au sol, comme celles transmises par la télémétrie et d'autres, relatives au déroulement de la mission de construction.
- **Système de levage** : ce système regroupe tous les éléments qui vont permettre de soulever un bloc drone-compatible avec le drone. L'ensemble de ce système est décrit au chapitre 4.

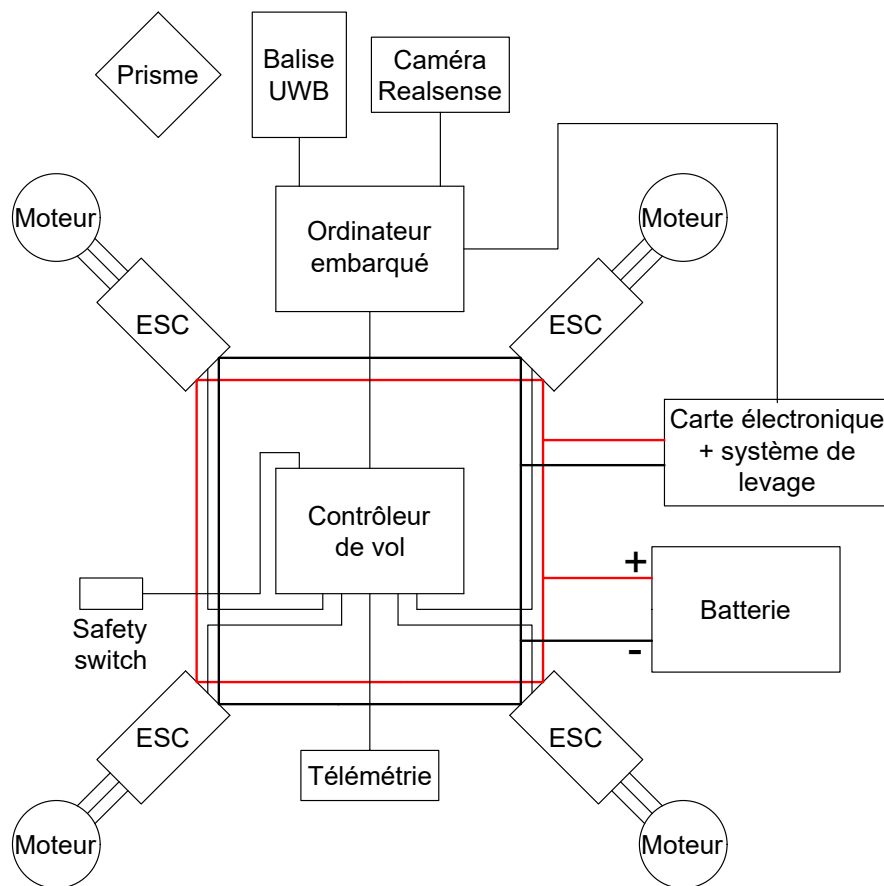


Figure 1.7 – Schéma d'un drone prévu pour la construction autonome.

1.1.4 Position d'un drone

La position exacte d'un drone est définie par ses 3 coordonnées dans l'espace (x, y, z) et ses 3 rotations autour de ses axes (*yaw*, *pitch*, *roll*) comme le montre la figure 1.8.

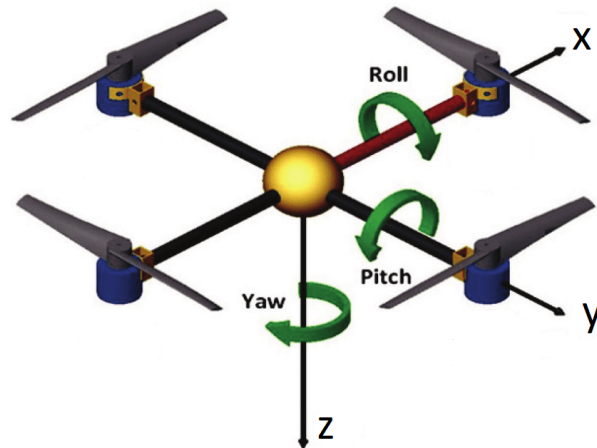


Figure 1.8 – Les 6 données de positionnement d'un drone.

1.2 L'ordinateur embarqué

L'ordinateur embarqué par le drone est l'élément central du système. C'est lui qui utilise toutes les ressources extérieures et celles du drone pour mener à bien la mission souhaitée. Pour cela, nous utilisons la carte *Odroid XU4* pour ses grandes capacités de calcul dues à son processeur 8 cœurs notamment³.

Pour l'utilisation que nous faisons de l'ordinateur (embarqué sur un système mobile), logiquement, aucun périphérique d'utilisation (clavier, souris, écran, etc.) n'y est connecté. Pour pouvoir interagir avec l'ordinateur embarqué, un routeur génère un réseau WiFi auquel l'Odroid se connecte automatiquement via une clé Wifi. Ensuite, nous connectons la station de contrôle au sol manuellement sur le même réseau afin de pouvoir le contrôler à l'aide d'une connexion SSH (Secure SHell). Cette connexion étant effectuée par l'intermédiaire de la station de contrôle, l'opérateur humain utilise ce nano-ordinateur pour exécuter les différents programmes d'une mission de construction. Les 3 rôles remplis par cet ordinateur embarqué sont décrits ci-dessous.

3. <https://www.hardkernel.com/shop/odroid-xu4-special-price/>



Figure 1.9 – Odroid XU4.

1.2.1 Le système de positionnement

Le premier rôle de l'ordinateur embarqué va être d'exécuter les différents scripts des systèmes de positionnement. L'ordinateur reçoit leur position et envoie la plus pertinente au contrôleur de vol. Ce processus se déroule en trois phases.

1. Récupération des données de positionnement de chaque système.
 - L'UWB envoie ses données à l'Odroid via un câble USB/USB micro B. Celui-ci peut les lire à l'aide du script qu'il exécute. L'UWB fournit les données (x, y, z) .
 - La station totale envoie les données de positionnement depuis la station au sol qui exécute le script de tracking du prisme. L'Odroid reçoit ces données via le réseau WiFi et le protocole UDP (User Datagram Protocol). La station totale fournit les coordonnées (x, y, z) mais aussi un **flag** relatif à la précision de la donnée envoyée.
 - La caméra Realsense envoie les données de positionnement via un câble USB/USB 3.0 relié à l'Odroid. Celui-ci, en exécutant le script de la caméra, récupère les données $(x, y, z, yaw, pitch, roll)$.
2. Sélection de la position la plus pertinente parmi toutes celles reçues. Le fonctionnement de ce processus de sélection et donc du choix de la position envoyée est discuté à la section 2.6.
3. Envoi de la position retenue au contrôleur de vol via le protocole de communication *MAVROS* [4] (explication ci-dessous).

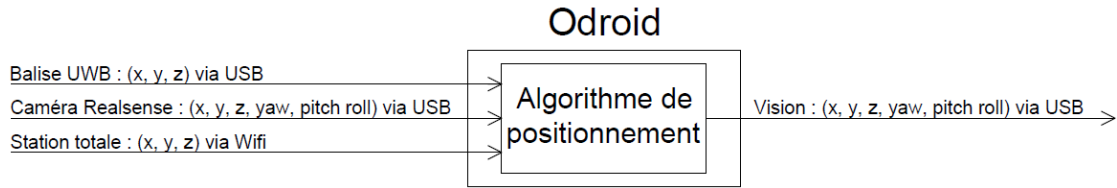


Figure 1.10 – Traitement des positions par l’Odroid.

MAVROS

MAVROS est un nœud de communication utilisant le système *ROS* (Robot Operating System). ROS est un outil open source informatique spécialisé dans la robotique. Les systèmes ROS sont des nœuds qui communiquent entre eux via des topics. Chaque nœud peut être **publisher**, **subscriber**, ou les deux. *MAVROS* est un nœud ROS utilisant le protocole de communication *MAVLink* (Micro Air Vehicle Link) [3]. Ce protocole de messagerie très léger est utilisé pour la communication avec les drones et entre les composants de drones embarqués.

1.2.2 Le système de guidage

La seconde fonction de l’ordinateur embarqué est d’envoyer les instructions de vol au contrôleur de vol. Pour cela, l’Odroid envoie au contrôleur de vol le prochain point à rallier à l’aide d’un nœud ROS que nous appelons "nœud de contrôle". Avec les positions qu’il reçoit du système de positionnement, le contrôleur dirige le drone (via les ESCs) vers la position donnée par le nœud de contrôle. Le fonctionnement complet de ce système est décrit au chapitre 3.

1.2.3 Le système de levage

Le dernier rôle de l’ordinateur embarqué va être de déclencher le système de levage. Ce système est activé ou désactivé à la prise ou à la pose d’un bloc. Le déclenchement est effectué dans le nœud de contrôle au moment où la position souhaitée est atteinte. La description du fonctionnement de ce système se fait au point 4.4.

1.3 Contrôleur de vol

Le contrôleur de vol est l'élément central d'un drone, il agit comme son cerveau. C'est un circuit imprimé plus ou moins complexe en fonction des modèles. Son rôle est d'envoyer les différentes informations aux ESCs dans le but de transmettre la bonne puissance aux moteurs du drone. Il permet de déplacer le drone ou de le maintenir en vol stationnaire. Le contrôleur de vol s'aide de différents capteurs tels qu'un gyroscope qui permet par exemple de stabiliser le drone en vol, une boussole interne pour connaître son orientation, ou encore un baromètre qui est utilisé pour connaître l'altitude du drone. Il est configurable, et propose différents modes de vol qui dépendent de l'utilisation souhaitée de l'engin. Pour cela, les contrôleurs sont fournis avec des logiciels qui permettent de configurer les modes. La plupart de ces programmes proposent des fonctions avancées, comme le retour au point de décollage ou la programmation d'un plan de vol en utilisant des points de passage.

1.3.1 Le Pixhawk

Pixhawk est un projet indépendant créant du matériel et des programmes open source. [2] Il trouve son origine dans un projet réalisé par des étudiants de l'ETH de Zurich en 2008, et est maintenant le système le plus utilisé dans l'industrie des drones. Son objectif est de développer un produit référentiel dans le domaine des contrôleurs de vol en créant un composant entièrement programmable, flexible, de très bonne qualité et à un prix abordable. Ce sont ces différents éléments qui nous ont poussés à choisir ce contrôleur de vol pour notre projet.

Jusqu'à présent, le *Pixhawk 1* était utilisé sur les drones de L'UCLouvain dans ce projet de recherche. Différentes raisons nous ont encouragés à installer une version plus récente en début d'année : le *Pixhawk 2*.



Figure 1.11 – Contrôleur de vol *Pixhawk 2*, le *Cube*.

Ce contrôleur de vol aussi appelé *le Cube* est une version améliorée de la première à plusieurs niveaux :

- Même s'ils ont tous les deux un processeur identique (*Cortex M4*), *le Cube* possède un port capable d'accueillir un mini-ordinateur *Intel Edison* pour augmenter ses capacités de calcul.
- Avec ses 3 accéléromètres, 3 gyroscopes, 3 magnétomètres et 2 baromètres, le *Pixhawk 2* possède une triple redondance de ses centrales inertielles. Ces centrales sont des systèmes utilisant les données de tous les capteurs pour estimer sa position (x, y, z) , son orientation *yaw* et son inclinaison *pitch* et *roll*.
- Tous les capteurs sont séparés du système de gestion de vol pour éviter les interférences.
- Le système est entièrement isolé avec un contrôle de la température, et est posé sur de la mousse pour réduire les vibrations.

La grande force du contrôleur de vol *Pixhawk* est de supporter les 2 programmes d'autopilote les plus importants dans l'industrie des drones, à savoir : *ArduPilot* et *PX4*. Ces deux programmes développent chacun un autopilote open source. Les développeurs utilisent les contributions de la communauté et des partenaires commerciaux pour constamment faire évoluer leurs programmes. Ces derniers permettent donc de paramétrer un grand nombre d'éléments et de programmer énormément de comportements pour le drone. Pour cela, ces autopilotes mettent à disposition des utilisateurs un logiciel. Pour le projet *PX4* [11], ce logiciel s'appelle *QGroundControl*.

QGroundControl

QGroundControl est le logiciel qui permet de configurer l'autopilote de *PX4*. [1] Il est aussi utilisé comme station terrestre lorsque le drone est en vol. En effet, il dispose d'une interface fournissant un grand nombre de données en temps réel. Il se présente comme montré dans la figure 1.12. Ce logiciel est aussi indispensable pour les différentes calibrations requises lors de la première utilisation du contrôleur. Il est important de réitérer ces calibrations si le *Pixhawk* est changé de châssis.

Le logiciel nécessite logiquement une connexion avec le contrôleur de vol. Pour la calibration des différents capteurs, un câble USB suffit. Cependant, pour utiliser le programme comme station terrestre, lorsque le drone est en vol, et ainsi recevoir ses informations, il est nécessaire de recourir à un dispositif sans fil. Pour cela, nous utilisons des émetteurs-récepteurs de télémetrie (processus d'acquisition et d'envoi de données pour des engins à distance). Ce système fonctionne avec deux émetteurs-récepteurs, un embarqué sur le drone et relié au *Pixhawk*, l'autre à l'ordinateur qui sert de station terrestre sur lequel *QGroundControl* est exécuté.

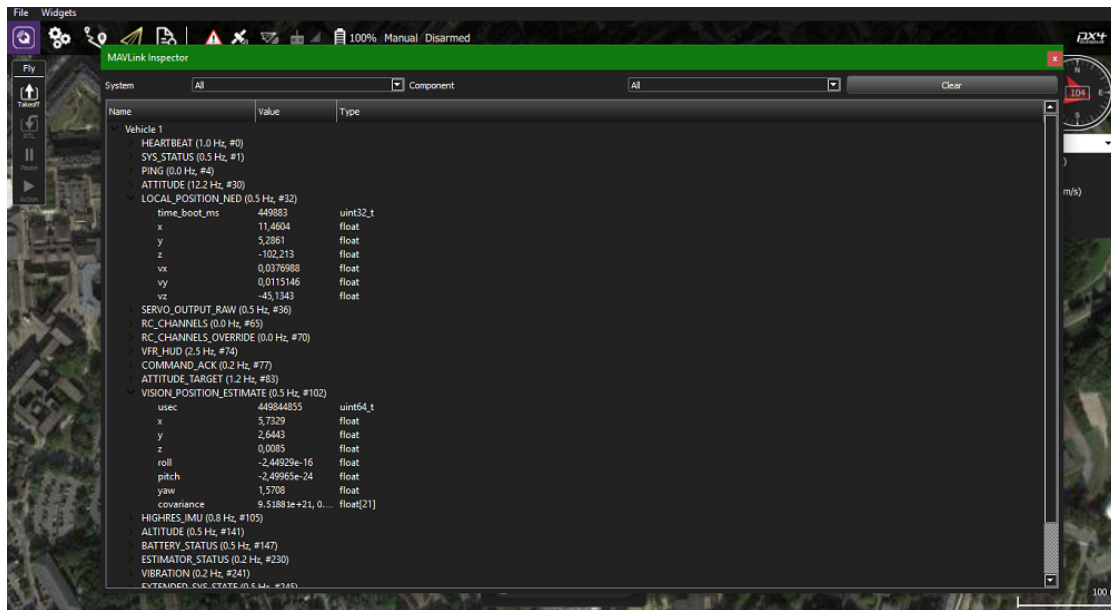


Figure 1.12 – Interface du logiciel *QGroundControl*.

Un nombre important d'informations peut être contrôlé pendant le vol :

- Le niveau de batterie. Cette information est essentielle et permet d'arrêter le drone quand il n'est plus capable de voler. Si le niveau est critique, le drone atterrit quelle que soit sa position, cela peut être dangereux. Il vaut mieux l'arrêter avant.
- Les messages d'erreur ou d'avertissement sont indispensables lorsque l'on veut connaître la raison pour laquelle le drone ne décolle pas ou ne réagit pas comme souhaité.
- La position locale et vision du drone. Ces deux positions sont définies au point 3.2.2.
- Son état : l'état du drone peut prendre plusieurs formes : stationnaire, manuel, autonome, ... Ces états sont présentés au point 3.3.

1.4 Système complet

La figure 1.14 présente l'ensemble des éléments constituant ce système de construction autonome par drone ainsi que leur connexion.



Figure 1.13 – Émetteur/Récepteur mro sik telemetry radio v2 433mhz.

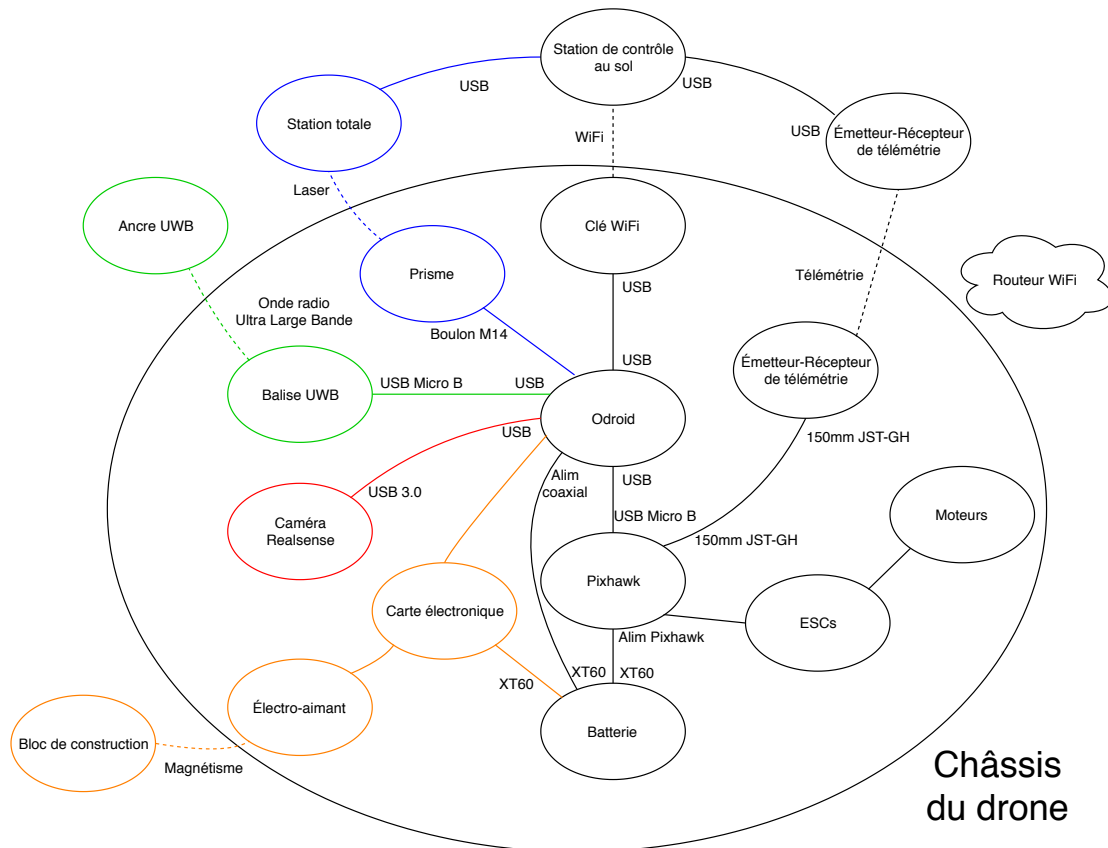


Figure 1.14 – Schéma des éléments du système complet et leur connexion.

Chapitre 2

Les systèmes de positionnement

Ce chapitre a pour objectif de décrire les différents systèmes de positionnement développés et utilisés dans ce projet. Pour chaque système, son fonctionnement, sa mise en place, sa précision seront expliqués et commentés. Les avantages et désavantages des différents systèmes permettront de comprendre le rôle de chacun dans un système complet de positionnement. Le fonctionnement de l'algorithme de positionnement utilisant tous les systèmes est décrit ainsi que la manière de les faire fonctionner ensemble dans un même repère.

2.1 Global et local

Premièrement, une mission de construction autonome à l'aide de drone s'envisage en deux phases aux regards des systèmes de positionnement :

- **Le déplacement vers un point** : pour pouvoir diriger le drone, il est indispensable d'utiliser des systèmes de positionnement capables de le situer lorsque celui-ci se déplace dans la zone de vol.
- **La recherche, la prise et la pose de bloc** : cette seconde partie nécessite un système de positionnement qui analyse directement ce qui se trouve en dessous du drone pour pouvoir guider celui-ci lorsqu'une grande précision est requise.

La première partie nécessite donc des systèmes de positionnement globaux, qui peuvent être de plusieurs types, tandis que la seconde requière des systèmes de positionnement locaux, par exemple une caméra à analyse d'images. Les deux situations sont représentées sur les figures [2.1](#) et [2.2](#).

Notre travail s'est concentré sur la caractérisation et l'intégration des systèmes globaux. Les différents systèmes présentés ci-dessous sont donc tous des systèmes permettant de situer le drone en vol.

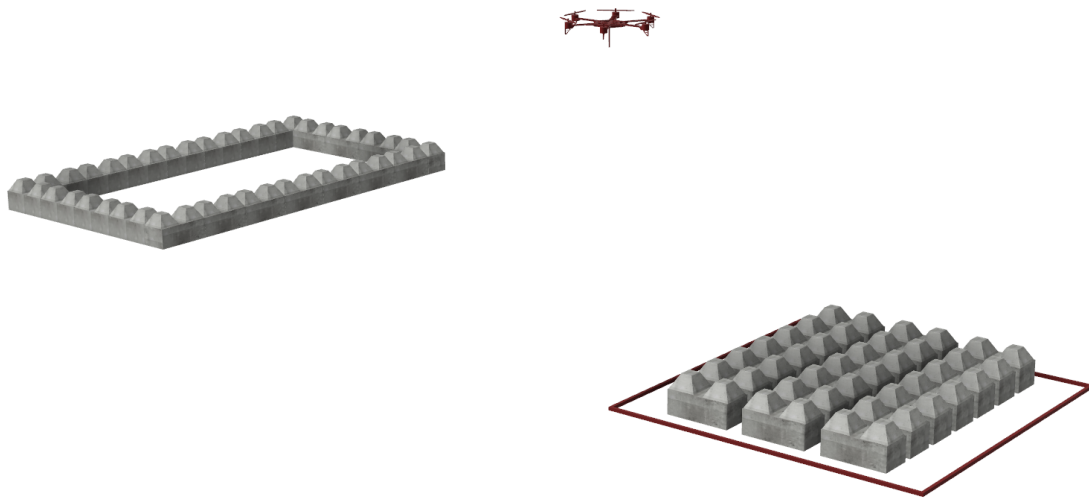


Figure 2.1 – Utilisation d'un système global.

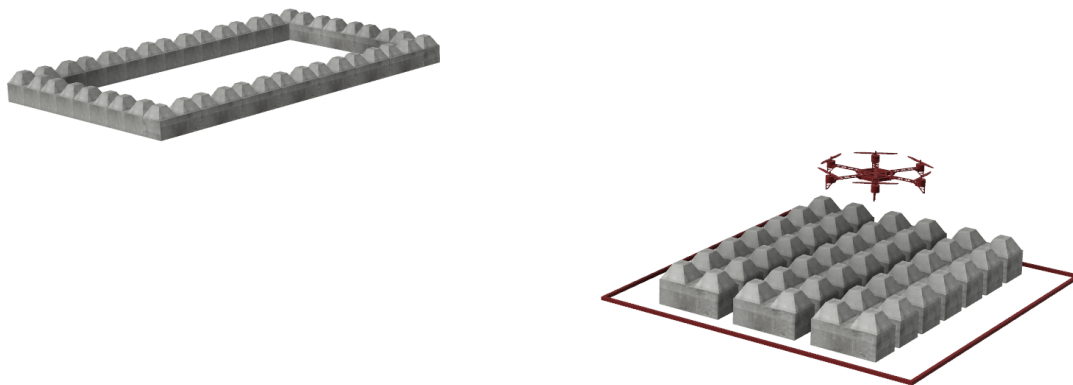


Figure 2.2 – Utilisation d'un système local.

2.2 Station totale

Le premier système de positionnement présenté a la particularité d'être le seul dont le script de calcul de la position n'est pas embarqué sur le drone. Il utilise une station totale, dans notre cas, le modèle TCRP1205¹ de la firme *Leica*.

Fonctionnement du système

Ce système de positionnement fonctionne à l'aide d'un laser qui se reflète dans un prisme. Le temps de trajet aller-retour du laser permet à la station de

1. http://www.surveyequipment.com/PDFs/TPS1200_User_Manual.pdf



Figure 2.3 – Station totale et prisme.

connaître la distance qui sépare la station totale du prisme. Cette information, une fois combinée à la rotation de la station par rapport à une position de référence, permet de connaître la position du prisme. La figure 2.4 montre la transformation à effectuer pour connaître la position du point P (position du prisme) par rapport au point S de référence.

$$\begin{aligned} - x &= \rho \sin\phi \cos\theta \\ - y &= \rho \sin\phi \sin\theta \\ - z &= \rho \sin\phi \end{aligned}$$

Desctripion du script

Deux scripts sont exécutés sur la station de contrôle au sol. Un premier qui définit le repère de la station puis cherche le prisme, le `lock` et le suit. Le second récupère les positions de la station et les envoie à l'ordinateur embarqué sur le drone. On peut décrire leur fonctionnement en 3 étapes :

1. **Mise en station** : cela permet de fixer le repère de coordonnées de la station totale. Pour cela, la station est orientée vers son axe x . La position au sol de la station totale est l'origine de ce repère.
2. **Recherche du prisme** : la station cherche le prisme embarqué sur le drone. Pour faciliter et accélérer cette phase, la station peut être manuellement

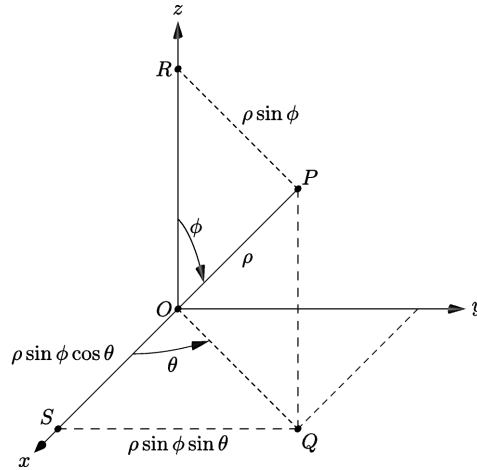


Figure 2.4 – Comment calculer les coordonnées cartésiennes à partir de coordonnées sphériques, dans un système de coordonnées en trois dimensions.

orientée vers le drone. Une fois le prisme trouvé, la station le **lock** et le suit pendant ses déplacements.

3. **Envoi des positions** : une fois que la station a **lock** le prisme, elle envoie, via un câble USB, la position du prisme dans le repère établi au point 1. Ces positions sont reçues par la station de contrôle au sol sur laquelle les scripts sont lancés, et celle-ci envoie les positions à l'ordinateur embarqué via le réseau WiFi avec le protocole de communication UDP.

Caractéristiques du système

Chaque donnée de positionnement contient un **flag** qui est un chiffre de 0 à 3 relatif à la précision de la donnée à laquelle il est associé. Une valeur de 0 pour ce **flag** constitue une précision optimale, 1 signifie que la station ne reçoit plus le retour du laser, et le calcul de la position repose donc sur la dernière position connue et la rotation qu'elle effectue. Les **flag** 2 et 3 ne sont pas accompagnés de données car ils sont trop mauvais.

La station totale perd en précision quand le mouvement du prisme lui demande une rotation trop rapide. Cette dernière dépend donc de la vitesse du prisme mais aussi de sa distance avec la station. Au plus le prisme sera loin de la station, au plus elle fournira des données précises. La station peut mesurer la distance au prisme jusqu'à une distance de 300 m en mode **lock**.

Positionnements de la station et du prisme

Comme vu ci-dessus, le placement de la station est important pour la précision qu'elle va donner mais aussi pour ne pas perdre le prisme. Pour assurer la meilleure précision possible, la station doit se situer dans un coin de la zone de construction pour être la plus éloignée possible du drone. Ensuite, pour suivre le prisme à tout moment, il faut aussi s'assurer que l'espace entre le laser de la station totale et le prisme soit toujours dégagé, ce qui est difficile à garantir sur site. L'utilisation de ce système est donc difficilement compatible avec le reste des activités d'un chantier (mouvement de grue, mouvement humain si la station est à faible hauteur, ...)

Deuxièmement, le prisme étant embarqué sur le drone, il faut aussi s'assurer que le drone lui-même ne gêne pas la trajectoire du laser, en l'occurrence, ses bras et ses hélices. Pour cela, la hauteur du prisme sur le drone fixe un angle maximal avec l'horizontale afin que le laser puisse toujours suivre le prisme, comme sur la figure 2.5. Cet angle est aussi l'angle maximal que la station totale ne peut dépasser verticalement, il établit une hauteur maximale que le drone ne peut pas dépasser en fonction de son éloignement de la station. Pour empêcher cela, deux solutions :

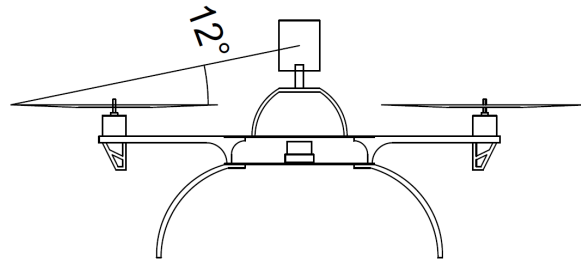


Figure 2.5 – Angle maximum du laser par rapport à la position du prisme.

- Élever la station sur un échafaudage, par exemple, pour que la hauteur maximale de vol du drone soit adaptée aux caractéristiques du chantier, comme montré sur la figure 2.6.
- Élever le prisme sur le drone. Cela a comme conséquence importante d'augmenter l'imprécision de la position quand le drone s'incline (autour des axes *Pitch* et *Roll*). Ce problème est schématisé à la figure 2.7. Cependant, plus le prisme est haut par rapport au centre de gravité du drone, plus le drone aura de difficulté à s'équilibrer dû au couple induit par le prisme.

Précision de la station totale

- **Statique** : pour quantifier la précision de la station totale de manière statique, les valeurs des 3 positions x , y et z ont été enregistrées pendant que le drone

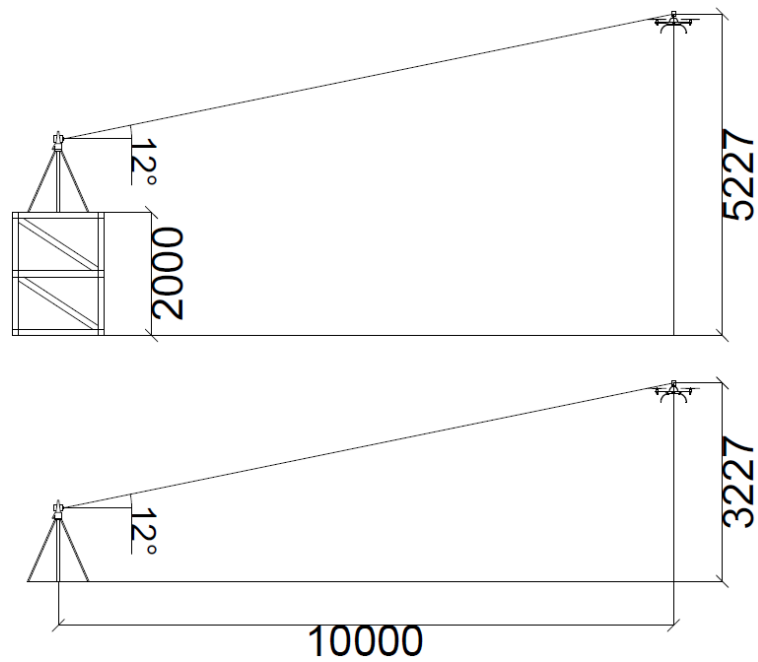


Figure 2.6 – Influence de la position de la station totale.

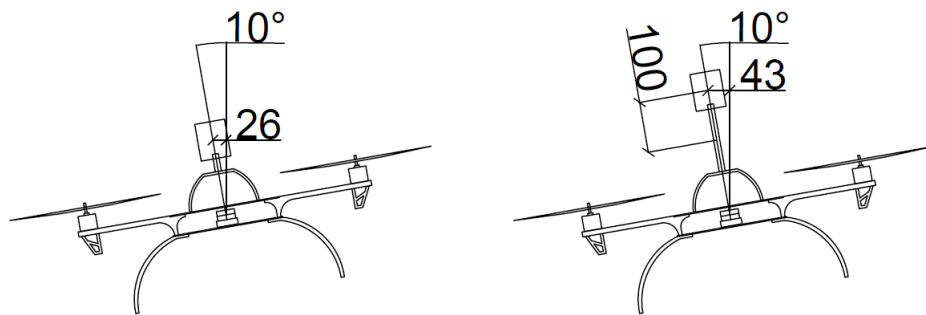


Figure 2.7 – Influence de la hauteur du prisme.

était au repos sur le sol. Les boxPlots des valeurs pour chaque position permettent de quantifier cette précision. L'écartement de ces positions est montré à la figure 2.8. Les boxPlots ont toutes été récentrées par rapport à leur moyenne pour être autour de 0. On remarque que la station totale est précise à (2 mm, 2 mm, 1 mm) pour les données en (x, y, z) .

- **Dynamique** : la précision de la station totale en mouvement n'a pas pu être quantifiée. En effet, pour que cela soit possible il existe deux solutions. Soit utiliser un système de positionnement encore plus précis, l'aligner dans le même repère que la station totale et comparer leurs données. Le système plus

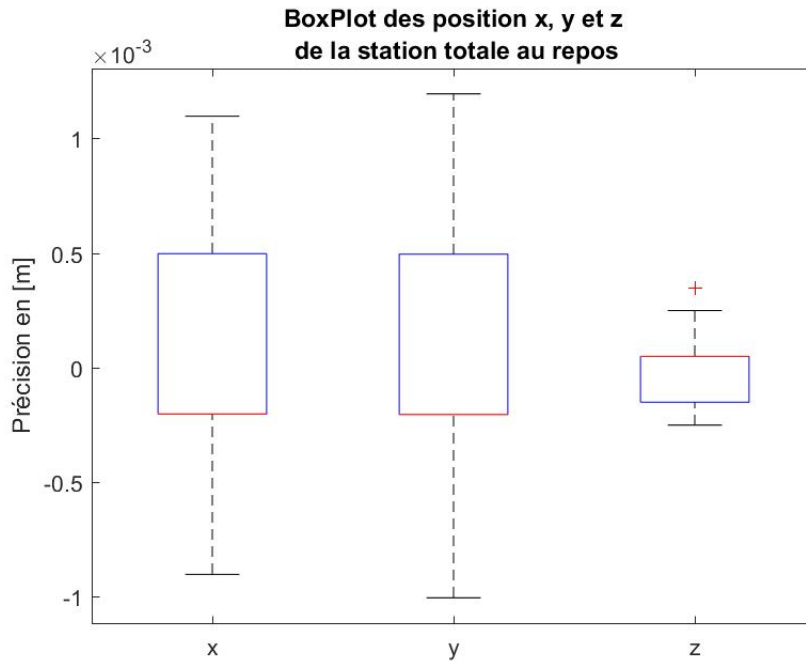


Figure 2.8 – Précision de la station totale au repos.

précis est alors utilisé comme référentiel et ses positions sont soustraient à celles de la station totale pour connaître l'imprécision de la station. Soit utiliser un système mécanique qui nous garantit le trajet du prisme, par exemple un système de rail. Les coordonnées du rail seraient mesurées de manière statique et, ensuite, le prisme serait déplacé sur ce rail. Les deux séries de données pourraient alors être comparées et la précision de la station totale en mouvement pourrait être définie.

Cependant, lorsque la station totale envoie des données avec le **flag 0**, donc précises, nous pouvons estimer la précision dynamique comme mesurée en statique. Le manque de précision interviendra avec la latence entre la prise de mesure et la considération de celle-ci par le contrôleur de vol.

Avantages

- Grande précision.
- Facilité de mise en place, installation rapide du système.

Désavantages

- Conditions difficiles pour que le système soit très précis.

— Risque de perte du signal.

Conclusion

Ce système de positionnement a le gros avantage d'être extrêmement précis et facile à mettre en œuvre. Le problème est qu'il est difficile de remplir les conditions pour garantir sa bonne précision. C'est donc un système très précis mais fragile.

2.3 Balise UWB

Le système de positionnement UWB pour Ultra Wide Band (Ultra Large Bande) utilise une balise émettrice d'ondes radio Ultra Large Bandes pour se situer.

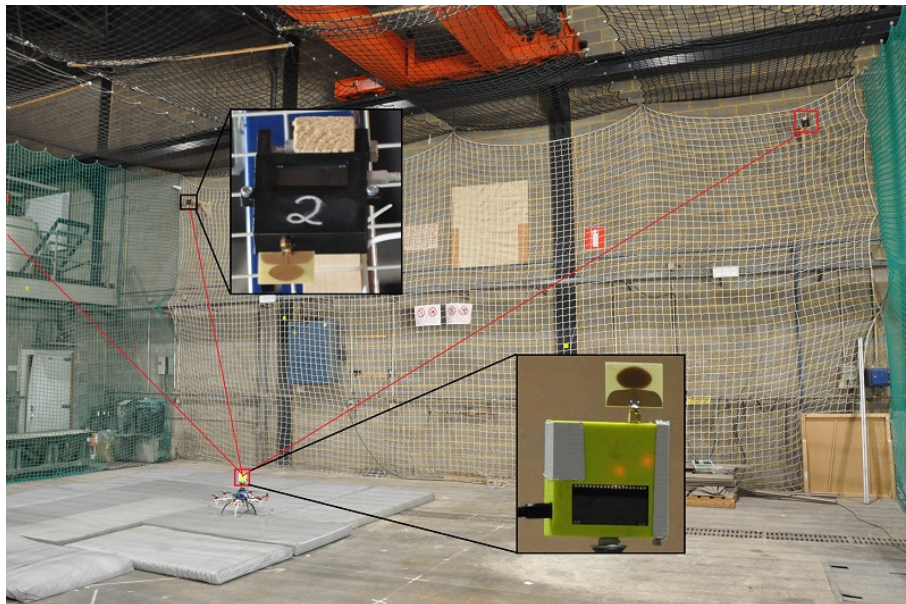


Figure 2.9 – Système UWB.

Fonctionnement du système

La balise envoie un signal radio ultra large bande que réceptionnent les ancres. Une fois ce signal reçu, une onde retour est alors envoyée par ces ancres et la balise peut trianguler sa position en connaissant le temps de propagation aller-retour des ces ondes, comme le montre la figure 2.10. Une fois que la balise est située entre les ancres, celles-ci étant placées dans un repère défini, les coordonnées de la balise sont connues. La balise est reliée à l'Odroid sur lequel tourne le script de positionnement de l'UWB, via un câble USB/USB micro B.

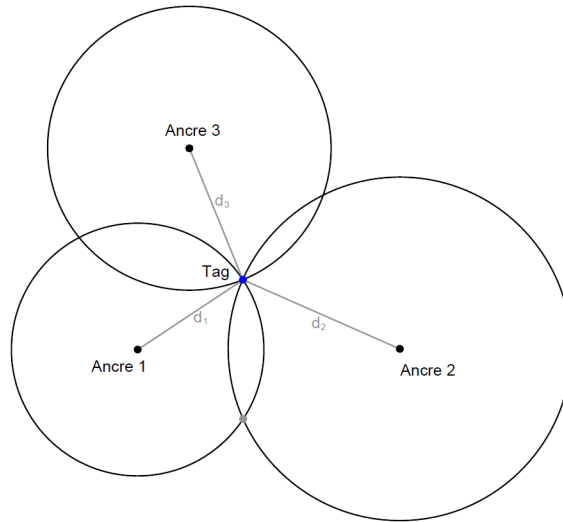


Figure 2.10 – Fonctionnement de l’UWB.

Descriptipon du script

Un script est exécuté par l’ordinateur embarqué. Il lui permet de récupérer les données de positionnement de la balise UWB fixée sur le drone. Son fonctionnement est décrit en 2 étapes :

1. Remplir le fichier de configuration avec les coordonnées exactes des 3 ancrs dans le repère choisi.
2. Lancer le script qui publie les positions calculées sur un topics ROS.

Positionnements des ancrs et de la balise

Plusieurs éléments sont à prendre en compte lors du placement des ancrs autour de la zone de vol et de la balise sur le drone :

- Ce système est sensible aux interférences qui peuvent être causées par des éléments métalliques entre la balise et ses ancrs. Sur le terrain, la possibilité d’être confronté à de telles interférences est forte. L’utilisation de ce système est compliquée lorsque dans la zone de vol se trouvent de gros équipements de chantier (grue, nacelle, container, groupe électrogène, ...)
- Ensuite, il est important de placer les ancrs de telle sorte que l’on couvre le mieux possible la zone de vol. Celles-ci doivent aussi se situer en hauteur. En effet, des ancrs trop proches du sol risqueraient de provoquer des réflexions sur celui-ci et ainsi perturber la précision du système.

- Dernière considération importante : la position de la balise sur le drone. Ce dernier transportant beaucoup de composants électroniques, il est donc important d'en écarter la balise pour éviter les interférences. Pour cela, la balise est placée en hauteur sur le drone. Au même titre que pour le prisme de la station totale, une balise placée trop haut présente plus de risques d'imprécision en x et y lorsque le drone s'incline autour de ses axes *roll* ou *pitch*. Les mêmes considérations sont à prendre en compte que pour le prisme au niveau du couple induit par le placement de la balise.

Les différentes interférences mentionnées ci-dessus n'ont pas été quantifiées ni même ressenties cette année. Cependant, Zakariya Bouhaddi, dans son travail sur le développement du système, a pu constater ces interférences l'année passée, notamment dues aux racks métalliques en bordure de drone zone. Les indications de positionnement décrites ci-dessus, servent donc de prévention à ces possibles interférences.

Précision du système

- **Statique** : pour quantifier la précision du système UWB de manière statique, les valeurs des 3 positions x , y et z ont été enregistrées pendant que le drone était au repos sur le sol. Les boxPlots des valeurs pour chaque position permettent de quantifier cette précision. L'écartement de ces positions est montré à la figure 2.11. Les boxPlots ont toutes été récentrées par rapport à leur moyenne pour être autour de 0. On observe donc que le système UWB est précis à (6 cm, 7 cm, 6 cm) pour les données en (x, y, z) .
- **Dynamique** : pour caractériser la précision de ce système de manière dynamique, nous avons utilisé la station totale, considérée comme suffisamment précise, comme référentiel. La quantification de la précision du système UWB sera donc à mettre en regard de celle de la station totale. Chaque donnée de positionnement en (x, y, z) de la station totale a été soustraite à celle de l'UWB. Ces différences sont ensuite mises en boxPlot, comme montré sur la figure 2.12. Celles-ci expriment l'écartement des données de l'UWB par rapport à celles de la station totale. On remarque que la précision dynamique de l'UWB en (x, y, z) est de l'ordre de (15 cm, 10 cm, 30 cm). L'imprécision du système de référence utilisé, à savoir la station totale, est 100 fois plus petite que celle de l'UWB ce qui nous permet de ne pas en tenir compte.

Avantage

- Système très sûr s'il est correctement mis en place.

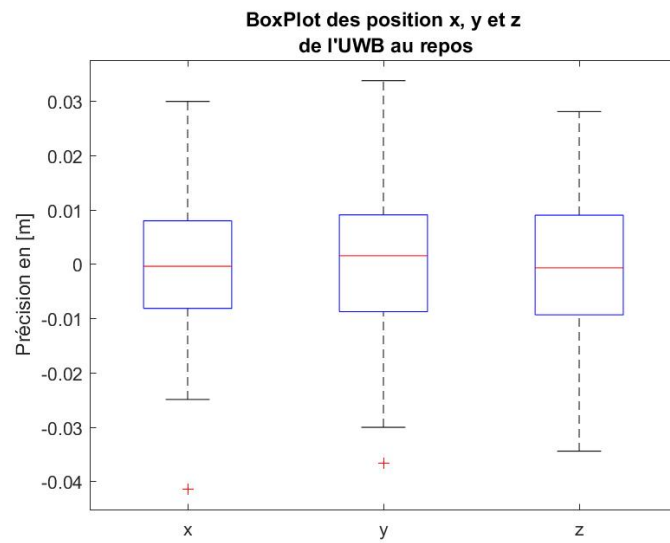


Figure 2.11 – Précision de l'UWB au repos.

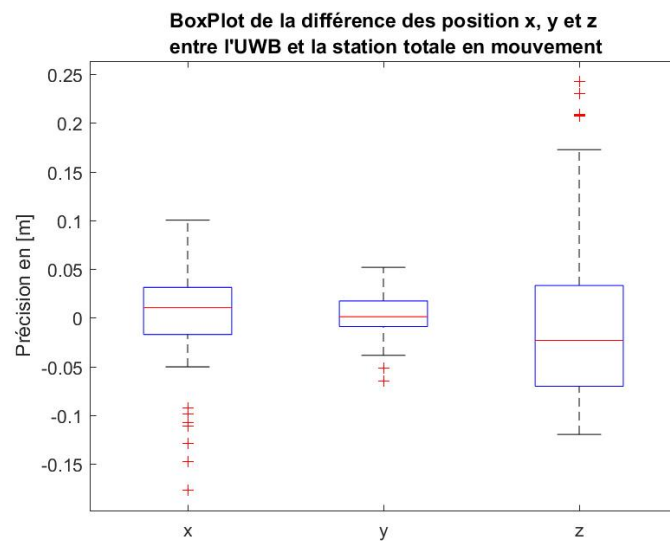


Figure 2.12 – Précision de l'UWB par rapport à la station totale en mouvement.

Désavantages

- Faible précision.
- Difficile à mettre en place. Doit être placé en hauteur et éloigné de toutes sources métalliques. Il faut relever les coordonnées précises des 3 ancres.

Conclusion

Ce système possède une précision moyenne mais présente une faible sensibilité à la perte du signal. Cela fait de lui un système très fiable comme système global. Il peut aussi servir de sécurité dans le cas où un système plus précis mais moins fiable viendrait à perdre son signal, comme la station totale par exemple.

2.4 Caméra Realsense

Le dernier système de positionnement utilisé dans ce projet est une caméra Realsense. Cette technologie est nouvelle car la caméra que nous utilisons, *Intel T265*, est sortie fin février 2019. Ce système est le seul entièrement embarqué sur le drone.

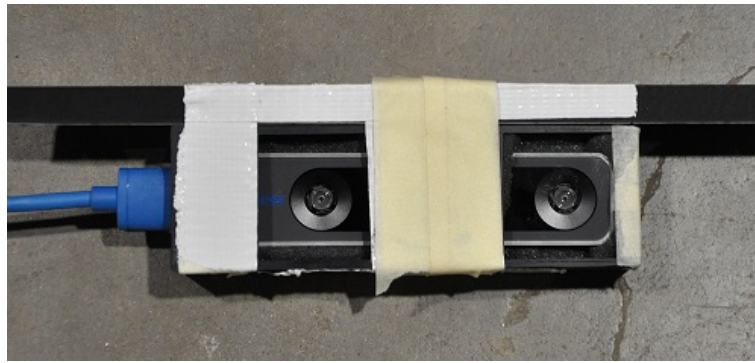


Figure 2.13 – Caméra Intel T265.

Fonctionnement du système

La caméra possède un algorithme SLAM (Simultaneous Localization And Mapping) qui consiste en la localisation et l'orientation d'un objet en cartographiant l'environnement autour de lui. La caméra fournit sa position par rapport à son environnement direct en cartographiant en temps réel l'espace dans lequel elle évolue. Son algorithme fonctionne en accrochant des points spécifiques dans l'espace. La caméra analyse ensuite le déplacement de ces points par rapport à son mouvement et combine ces informations avec celles de ses capteurs internes (gyroscope, accéléromètre, compas). Elle peut alors se positionner dans l'espace. Il est important que l'environnement dans lequel évolue la caméra soit fort contrasté pour qu'elle puisse trouver des points d'accroche.

Descriptipion du script

Le script exécuté sur l'ordinateur embarqué suit 3 étapes :

1. Remplir le fichier de configuration avec la position de départ de la caméra (équivalente à celle du drone). Cela permet à la caméra de se situer dans l'espace.
2. Orienter la caméra face à l'axe x positif. Cela permet avec le point 1 de fixer le repère dans lequel la caméra va évoluer.
3. Lancer le script qui publie les positions et orientations calculées par son algorithme.

Positionnement de la caméra

Vu que ce système est entièrement embarqué sur drone, seule la position de la caméra sur celui-ci est à considérer.

Le système repose entièrement sur ce qu'observe la caméra, il est donc important que celle-ci aie une vue bien dégagée sans qu'aucun élément du drone ne vienne perturber son champ de vision.

Pour naturellement placer la caméra dans le même repère de coordonnées que le drone, celle-ci doit être placée à l'avant du drone, orientée dans la même direction que celui-ci.

Précision du système

- **Statique** : pour quantifier la précision de la caméra de manière statique, les valeurs des 3 positions x , y et z ont été enregistrées pendant que le drone était au repos sur le sol. Les boxPlots des valeurs pour chaque position permettent de quantifier cette précision. L'écartement de ces positions est montré à la figure 2.14. Les boxPlots ont toutes été récentrées par rapport à leur moyenne pour être autour de 0. On observe que la précision statique de la caméra est en dessous du millimètre, ce qui est une très bonne précision.
- **Dynamique** : pour caractériser la précision de la caméra de manière dynamique, nous avons utilisé la station totale, considérée comme suffisamment précise, comme référentiel. La quantification de la précision de ce système sera donc à mettre en regard de celle de la station totale. Chaque donnée de positionnement en (x, y, z) de la station totale a été soustraite à celle de la caméra. Ces différences sont ensuite mises en BoxPlot, comme montré sur la figure 2.15. Celles-ci expriment l'écartement des données de la caméra par rapport à celles de la station totale. On remarque que la précision de la caméra en mouvement est supérieure au mètre. Cette précision est très

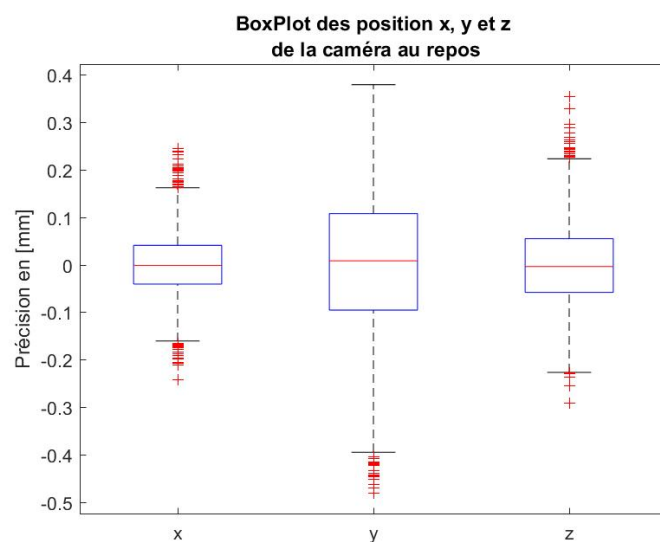


Figure 2.14 – Précision de la caméra au repos.

mauvaise et ce n'est d'ailleurs pas la précision réelle de cette caméra. Les causes de l'obtention de ces résultats sont discutées au point 5.2.2.

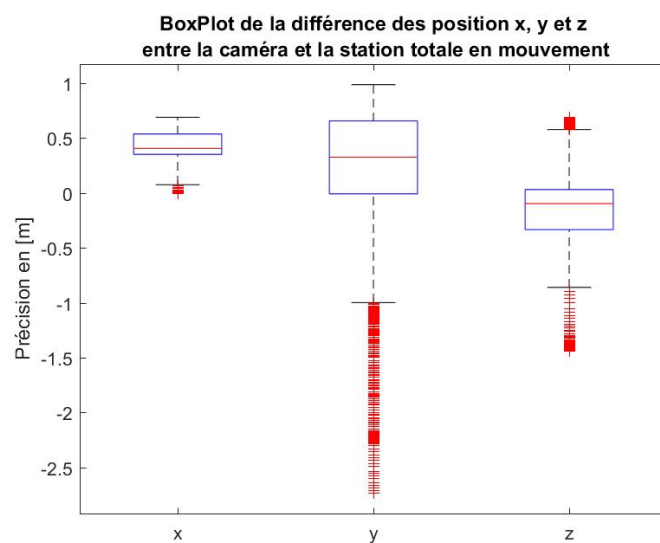


Figure 2.15 – Précision de la caméra en mouvement par rapport à la station totale.

Avantages

- Fournit une donnée d'orientation.
- Système complètement embarqué sur le drone donc pas de problème de communication ou de perte de données.
- Grande précision théorique.
- Système simple à mettre en place.

Désavantage

- Nécessite beaucoup de contraste dans son champ de vision.

Conclusion

La caméra Realsense et l'algorithme SLAM intégré dans celle-ci semblent très prometteurs comme système de positionnement embarqué sur drone. Cependant, cette technologie étant toute récente, nous n'avons pu en tirer tous les avantages comme le montre la précision dynamique très peu cohérente.

2.5 Comparaison des systèmes

Maintenant que les trois systèmes ont été présentés, il est important de les comparer. Cela va permettre de créer un système global intégrant tous les systèmes pour que les avantages de chacun comblient les défauts des autres. Le tableau 2.1 compare les trois systèmes selon les critères les plus importants pour caractériser un système de positionnement global.

Critères	Station totale	UWB	Caméra Realsense
Données	x, y, z	x, y, z	$x, y, z, yaw, pitch, roll$
Ordre de précision	millimètre	dizaine de centimètres	millimètre ?
Fiabilité	-	+	+/-

Table 2.1 – Comparaison des systèmes de positionnement.

Les informations de la caméra comme la fiabilité et la précision ne sont pas certaines. La caméra est censée être un système très fiable et très précis mais nous n'avons pas pu le constater dans nos essais.

Les données qui nous intéressent pour un système de positionnement sont les trois coordonnées (x, y, z) ainsi que l'orientation *yaw*. Le *pitch* et le *roll* sont naturellement dirigés par le drone pour se déplacer ou se stabiliser.

Il est important d'utiliser au moins trois systèmes. En effet, en utilisant deux systèmes, si l'un est défectueux et n'envoie plus de données cohérentes, il n'y a pas de moyen qui permette de savoir lequel est incohérent. Ce problème n'intervient pas quand trois systèmes sont utilisés.

2.6 Algorithme de positionnement

Une seule position (x, y, z, yaw) comprenant quatre variables est envoyée au *Pixhawk*. L'algorithme de positionnement est le script qui va sélectionner pour chacune des quatre données le système le plus pertinent. En fonction des circonstances, les données peuvent être une fois celles de l'UWB, une fois celles de la station totale par exemple.

Le premier choix de l'algorithme va dépendre du moment de sa mission. S'il est en vol pour se déplacer, il utilisera ses systèmes de positionnement globaux ; s'il doit chercher un bloc ou poser celui-ci, les positions envoyées au *Pixhawk* seront celles d'un système local de positionnement.

Choix du système global

Intéressons nous ici à l'algorithme pour les systèmes globaux.

L'algorithme va fonctionner en "échelle". C'est-à-dire qu'un système de positionnement sera toujours choisi par défaut. Si celui-ci venait à être défectueux ou à envoyer des données non-cohérentes, les données du second système seraient envoyées. Le troisième système serait utilisé si le deuxième venait lui aussi à être défaillant.

Le premier système, la station totale, sera le plus précis. Cela permet en conditions optimales d'envoyer au *Pixhawk* les données les plus précises. Cependant, comme énoncé plus haut, la station totale est un système fragile qui peut facilement perdre le prisme. Si c'est le cas, le second système, l'UWB, sera alors utilisé. Moins précis mais beaucoup plus fiable, ce système est parfait pour seconder la station totale. La caméra ne donnant pas de données cohérentes pour le moment, elle n'est utilisée quand dernier recours dans l'algorithme. Cependant, si ces incohérences sont corrigées, ce système pourrait largement devenir le premier de l'algorithme.

Le fait que le système soit embarqué le rend très fiable quant à la position réelle du drone.

Pour l'orientation, *yaw*, seule la caméra peut fournir une valeur. Il n'y a donc, pour le moment, pas de solution de back-up.

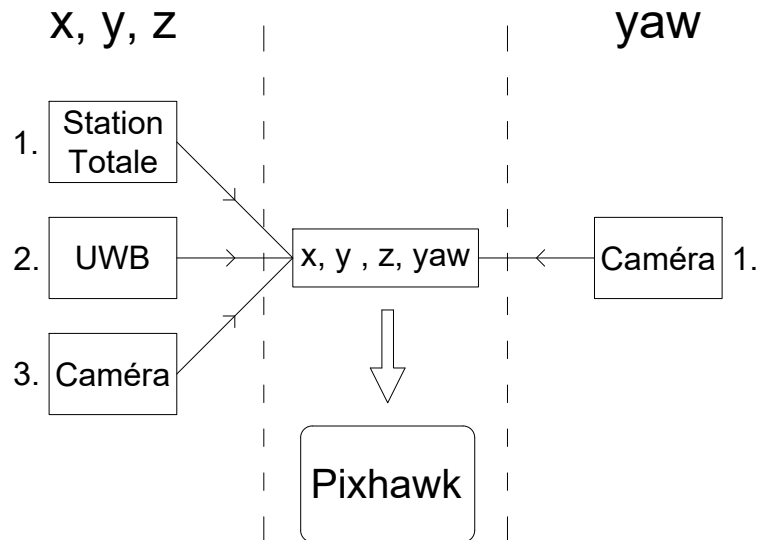


Figure 2.16 – Schéma de l'algorithme de positionnement.

L'algorithme prend donc trois entrées différentes et ne fournit qu'une sortie.

Entrées

- Station totale : x, y, z .
- Position UWB : x, y, z .
- Position Caméra Realsense : x, y, z, yaw .

Sortie

- Une position : x, y, z, yaw .

2.7 Mise en place du système

Le but de cette section est de parcourir la mise en place de tous les systèmes pour que ceux-ci travaillent ensemble, dans le même repère. Le point qui requiert le plus d'attention dans cette démarche est le premier qui consiste à définir un repère simple et cohérent avec le projet dès sa conception. Cela facilitera les étapes de mise en place des systèmes et la compréhension des coordonnées des éléments à prendre ou à poser. Le repère peut par exemple être défini parallèlement aux axes

du futur bâtiment à construire. L'origine de ce repère doit être la plus éloignée possible vu qu'elle définit la position de la station totale, qui elle-même doit être éloignée.

Étapes de mise en place

Toutes les étapes définies ci-dessous sont illustrées à la figure [2.17](#).

1. Définir un repère arbitraire.
2. Placer les ancres UWB à 3 endroits distincts les uns des autres pour couvrir toute la zone de vol.
3. Placer la station totale en $(x; y) = (0; 0)$ du repère défini et la placer le plus haut possible pour surplomber le site de construction et faire la mise en place de la station totale selon ce repère.
4. Relever les coordonnées des 3 ancres UWB à l'aide de la station et les introduire dans le fichier de configuration.
5. Placer le drone à sa position **Home** (ce sera sa position de décollage et d'atterrissage), orienté vers les x positifs. Cela permet au drone et à la caméra de setup leur *yaw* à 0.

Offsets

Une fois le matériel de chaque système mis en place, il est important d'intégrer au fichier de configuration tous les offsets des différents systèmes. Le point de référence du drone est en son centre pour x , y et au niveau de ses pieds pour z . Aucun système n'a d'offsets pour l'orientation car la caméra (le seul système qui fournit un *yaw*) est placée dans le sens du drone. Les différents décalages en z sont montrés sur le schéma à la figure [2.18](#). Les offsets en x , y sont aussi à prendre en compte, même entre le prisme et la balise.

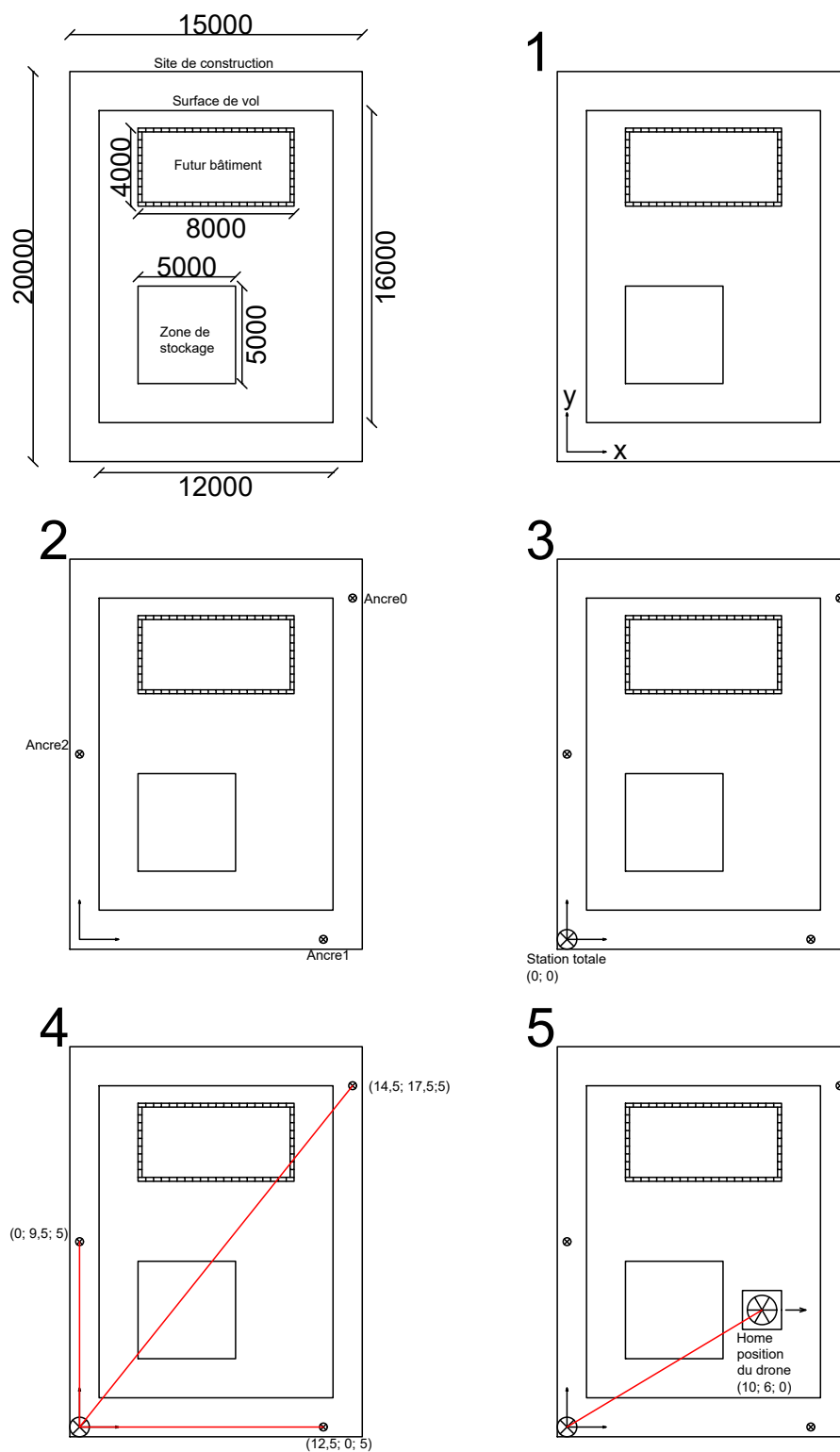


Figure 2.17 – Schéma des étapes de la mise en place du système de positionnement.

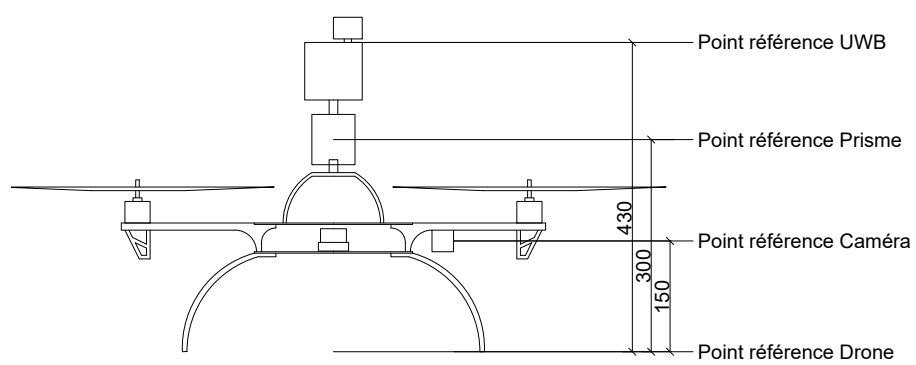


Figure 2.18 – Les différents offsets à prendre en compte pour chaque système de positionnement.

Chapitre 3

Système de guidage

Le système de guidage est l'ensemble des composants et des programmes qui permettent de contrôler le drone en vol. Ce système permet au drone d'exécuter de manière autonome une mission de vol tout en maintenant simultanément certaines mesures de sécurité. L'objectif de ce projet étant de permettre à un drone de construire un mur de maçonnerie, nous devons créer un système de guidage permettant au drone de se déplacer de manière autonome pour pouvoir prendre et déposer des éléments drone-compatibles.

3.1 Définir une mission

Une mission de construction par drone est l'ensemble des actions que le drone va exécuter pour construire le projet voulu. Vu la complexité du problème, il est important de décomposer cette mission en action élémentaire. Cela est plus simple à concevoir, à comprendre et à entretenir. Cette décomposition est schématisée sur la figure [3.1](#).

Actions élémentaires

3 actions permettent le déroulement complet d'une mission de construction par drone :

- Se déplacer jusqu'à une certaine position et orientation.
- Scanner une zone à l'aide d'un système local comme énoncé au point [2.1](#).
- Prendre ou déposer un bloc.

La première action permet de déplacer le drone, par exemple, de la zone de stockage, où il récupère les blocs, jusqu'à la zone de construction, où il pose les blocs. Cette action est réalisée grâce aux systèmes de positionnement globaux présentés au chapitre [2](#).

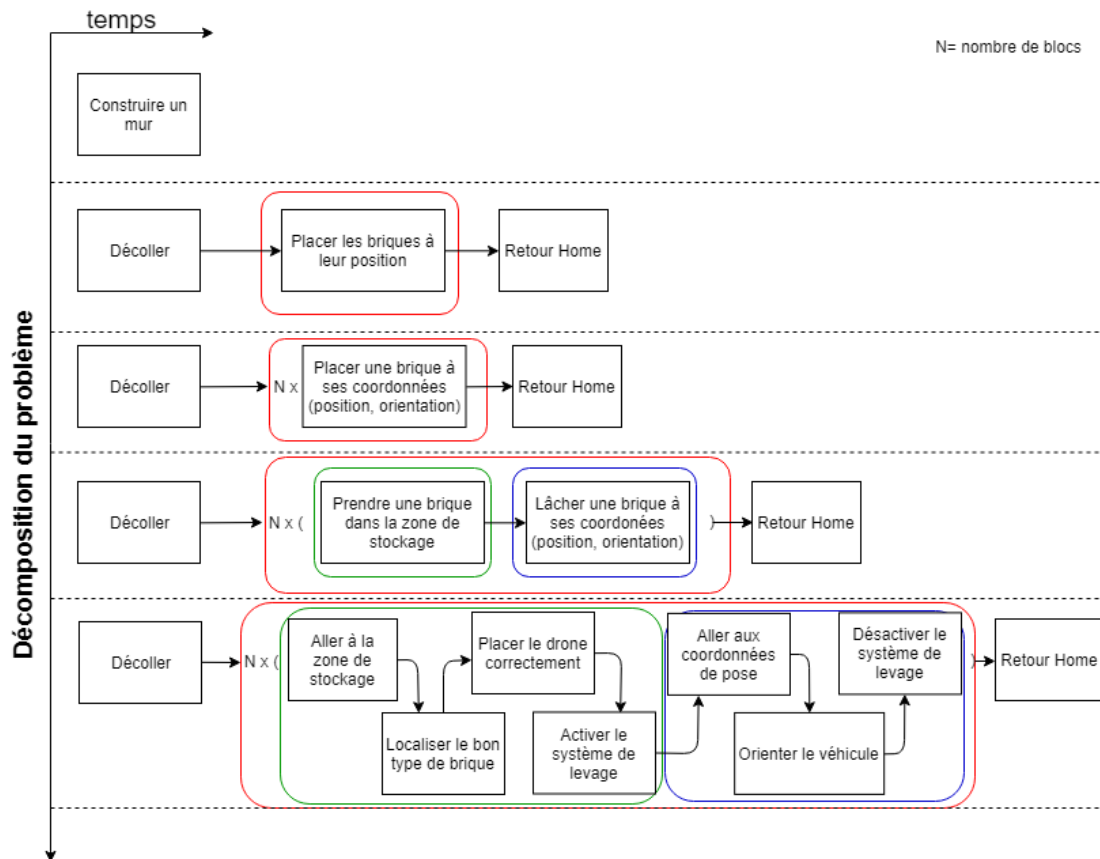


Figure 3.1 – Décomposition d’une mission de construction d’un mur.

La seconde action utilise un système local de précision pour scanner la zone de la prise du bloc et de la pose d’un bloc de construction. Ce système permet de reprendre l’imprécision des systèmes globaux avec un scan en temps réel de la zone située en dessous de lui.

La dernière action est propre au système de levage et permet d’activer et de désactiver celui-ci dans le but de prendre ou de déposer un bloc.

Déroulement d’une mission

Toutes les missions commencent une fois que les systèmes de positionnement sont mis en place. Le drone se trouve sur sa position **Home**. Cette position définit les coordonnées et l’orientation du drone avant le décollage. Elle est utile car cette position sert à l’atterrissage.

Tout est mis en place et la mission peut démarrer, le drone décolle à sa hauteur de vol. Celle-ci doit être définie en fonction de la zone de vol disponible et, de

préférence, au-dessus de la taille humaine pour éviter tout contact.

Le drone se rend à la zone de stockage des éléments drone-compatibles. Cette zone est définie par les coordonnées de ses 4 coins. Une fois à son niveau, le drone descend à une hauteur propre au système de positionnement local. C'est lui qui se charge de scanner la zone de stockage à la recherche du bon type de bloc. Une fois le bloc adéquat trouvé, le drone descend à hauteur du bloc et le saisit. Il remonte à sa hauteur de vol et se dirige vers la position de pose du bloc.

Ensuite, le drone se positionne aux coordonnées de pose du bloc emportées précédemment. Le système local reprend le dessus et guide le drone pour poser le bloc. Le système de levage est désactivé et le bloc se pose à son emplacement final. Le drone monte à sa hauteur de vol et, soit il retourne chercher un nouveau bloc dans la zone de stockage, soit il atterrit à sa position Home si c'était le dernier bloc.

La représentation complète du déroulement d'une mission, du décollage jusqu'à la pose du bloc, décrite ci-dessus, est représentée à la figure 3.2.

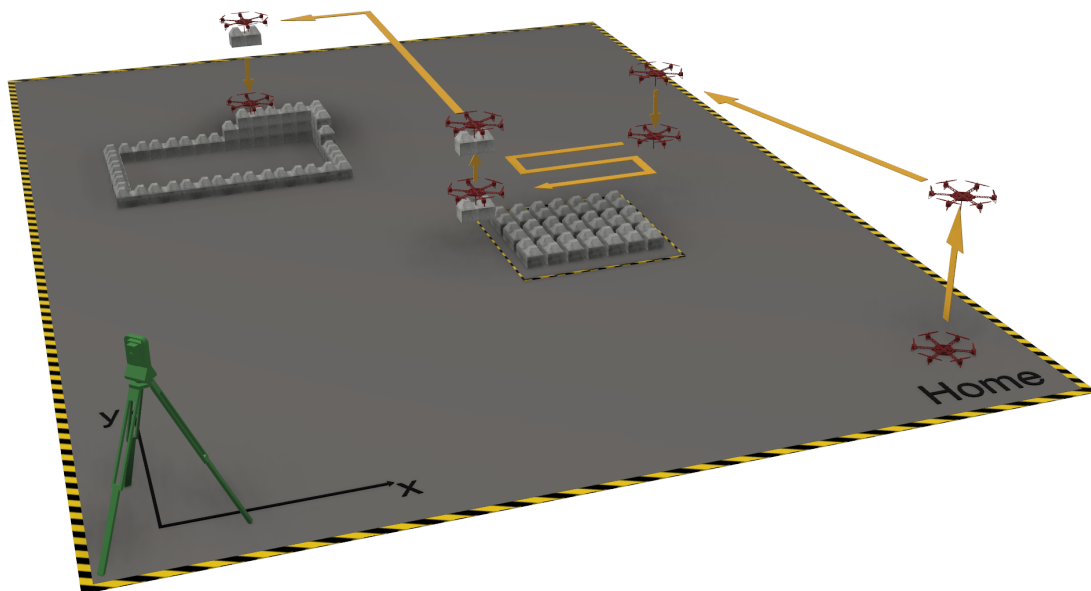


Figure 3.2 – Représentation 3D des étapes d'une mission de construction par drone.

Fichier de mission

Le fichier de mission est le document qui contient toutes les informations propres à la mission. Il est construit comme ceci :

Position Home :	$[x_H, y_H, z_H, o_H]$
Zone de stockage :	$[x_1, y_1, z_1]$ $[x_2, y_2, z_2]$ $[x_3, y_3, z_3]$ $[x_4, y_4, z_4]$
Hauteur de vol :	h_V
Hauteur de scan :	h_S
Informations blocs :	List : $[type_i, x_i, y_i, z_i, o_i]$

La hauteur de scan est dépendante du système de positionnement local. En effet, c'est le système utilisé qui définira la hauteur optimale à laquelle il doit voler pour scanner les blocs.

Naturellement, lors de la pose d'un bloc, le système global amène le drone à la position de pose du bloc en (x, y) et $z + h_S$ pour que le système local puisse scanner l'endroit et placer le bloc de manière précise.

La dernière donnée consiste en une liste où chaque élément représente un bloc. Chaque bloc est défini par sa position et son orientation de pose et par son type. Le type de bloc est précisé en amont lors de la conception du projet. Cette liste détermine l'ordre de pose des blocs, les premiers éléments de la liste sont les premiers blocs posés par le drone. Le premier bloc posé par le drone n'est pas le premier élément posé sur la fondation. En effet, une première rangée de blocs doit y être posée manuellement pour que les blocs amenés par le drone aient un guide pour s'emboîter.

3.2 Contrôler le drone

3.2.1 Mode de vol

Le mode de vol fourni par *PX4* nous permet d'obtenir différents types d'assistances du pilote automatique : du décollage à l'atterrissage automatisé en passant par le maintien de la position du véhicule. Pendant un vol, on peut souhaiter passer d'un mode à l'autre. C'est possible soit via la télécommande en la paramétrant depuis *QGroundControl*, soit via *QGroundControl* directement en vol, soit depuis le script de contrôle sur l'ordinateur embarqué. Une liste complète décrivant tous les modes de vols existants pour notre contrôleur de vol figure dans la documentation

PX4. [12]

Notre système utilise les modes suivants :

- **Offboard** : "Vehicle obeys a position, velocity or attitude setpoint provided over MAVLink (often from a companion computer connected via serial cable or WiFi)." [12] C'est-à-dire que le véhicule obéit à une consigne de position, de vitesse ou d'attitude fournie via *MAVLink*, souvent depuis un ordinateur embarqué par câble ou WiFi.
- **Land** : "Vehicle lands at the position where the mode was engaged." [12] Le véhicule atterrit là où le mode de vol est changé.
- **Hold** : "Vehicle hovers at the current position and altitude." [12] Le véhicule plane à la position et à l'altitude actuelle.

La documentation fournit un autre mode intéressant : **mission**. Ce mode de vol est conçu pour exécuter un plan de vol prédéfini. Cependant, nous avons fait le choix de n'utiliser que le mode **offboard**. En effet, celui-ci nous laisse plus de liberté pour ajouter des commandes de mission comme l'activation de l'électro-aimant.

3.2.2 Setpoints

Une fois en mode **offboard**, la façon de contrôler le drone est de lui envoyer une série de setpoints. Ces setpoints sont des données de position et d'orientation du drone. Le contrôleur de vol, avec les données de positionnement qu'il reçoit de l'algorithme de position présenté au point 2.6, va déplacer le drone jusqu'à atteindre le setpoint.

Pour atteindre un setpoint, le *Pixhawk* utilise la position envoyée par l'algorithme de position. Cette position est la donnée **vision**. Mais cette position n'est pas la seule à être prise en compte. En effet, la position **vision** reçue par le *Pixhawk* va être filtrée et comparée aux données internes de ses capteurs (compas, baromètres, accéléromètres, gyroscopes). La position du *Pixhawk* est donc une moyenne entre ses données de positionnement externes et internes. Cette position est appelée position **local**. Si les données **vision** ne sont pas cohérentes avec les données internes, la position **local** ne sera pas cohérente avec la réalité et le comportement du drone n'est alors plus prévisible.

Le setpoint à envoyer au *Pixhawk* dépend donc de la mission en cours. La donnée du setpoint va être envoyée en boucle au *Pixhawk* jusqu'à ce qu'il l'atteigne. Une fois le setpoint atteint, nous choisissons soit de lui envoyer un setpoint suivant en boucle, soit de lui commander une action du type activer ou désactiver son électro-aimant.

3.3 Sécurité

À tout moment pendant l'exécution d'une mission, nous devons pouvoir prendre le contrôle du drone. C'est vital pour des raisons de sécurité, car le drone peut présenter un comportement étrange, nuire aux personnes se trouvant dans son environnement direct, causer des dommages aux matériaux de construction, voire s'autodétériorer. Ces accidents peuvent avoir des conséquences morales et financières que nous voulons absolument éviter. Nous pourrions simplement vouloir annuler la mission.

Pour cela, les différentes fonctions présentées ci-dessous ont la priorité sur le déroulement de la mission. Toutes ces commandes n'assurent pas l'arrêt de la mission sans dommage pour le drone. Nous devons noter que diverses commandes peuvent également, dans certains cas, endommager le drone.

- **DÉSARMER** : cette commande forcera le drone à couper tous ses moteurs. En conséquence, le drone tombera au sol et se cassera très probablement. Cette commande doit être utilisée si le drone représente un réel danger pour son environnement ou une personne. Il ne faut pas l'activer si une personne se trouve sous ou à proximité du drone. Des pièces du drone pourraient se disperser aux alentours. C'est le moyen le plus rapide de neutraliser le drone.
- **ATTERRIR** : cela oblige le drone à atterrir à sa position actuelle. Si le drone porte un bloc ou est au-dessus d'un obstacle, cela peut être problématique. Cette commande peut être utilisée en cas de danger indirect (par exemple, le vent) ou pour mettre le drone en sécurité (en l'absence de bloc ou d'obstacle).
- **STATIONNAIRE** : cette commande imposera au drone de rester inactif à sa position actuelle. De toute évidence, cette commande doit être suivie d'autres commandes pour que le drone retourne au sol.
- **RETOUR HOME** : cela demandera au drone de suivre son chemin de retour jusqu'au point de départ du drone.
- **DÉSACTIVER ÉLECTRO-AIMANT** : l'électro-aimant n'est plus activé. Tout objet suspendu à l'aide de celui-ci tombe naturellement sur le sol. Cette commande n'est à activer que si l'espace sous le drone est dégagé.
- **MANUEL** : Cette commande forcera le drone à passer en mode manuel, permettant à une personne de reprendre le contrôle grâce à la télécommande. En pratique, cela peut être très complexe car il n'est pas facile de reprendre les commandes d'un drone en vol en sachant par exemple que la position neutre des gaz est 0 sur une télécommande. Cette commande peut avoir le même effet que le désarmement des moteurs pendant un court instant.

Chapitre 4

Système de levage

Ce chapitre décrit le dernier système important pour réaliser un système de construction autonome à l'aide de drone. Le système de levage est l'ensemble des éléments qui vont permettre la prise, le déplacement et la pose des éléments drone-compatibles.

4.1 Choix du système de levage

Plusieurs systèmes de levage ont été développés dans le TFE de Georges Deneffe l'année dernière [7]. Les solutions qu'il a imaginées sont brièvement décrites ci-dessous.

Pince de levage

Le premier système consiste en une pince de levage. Le frottement obtenu entre les deux faces de la pince et du bloc à soulever permet le transport de ce dernier.

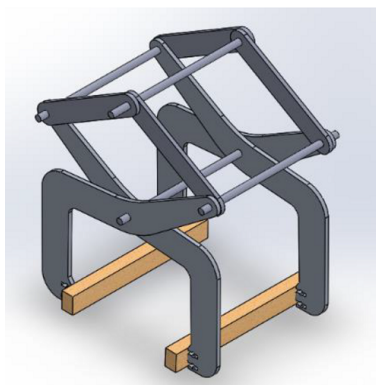


Figure 4.1 – Exemple de pince de levage pour bloc rectangulaire.

Rotation de tige à forme T

Ce second système est une tige en forme de T suspendue au drone. Celle-ci s'insère dans un bloc dans lequel une fente est créée. Le drone pivote ensuite de 90° . À ce moment-là, le bloc peut être soulevé car les deux extrémités qui forment le T empêchent la tige de glisser dans le bloc.

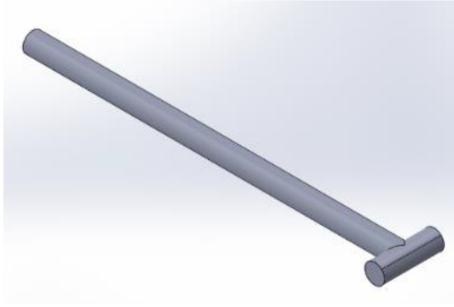


Figure 4.2 – Tige en forme de T.

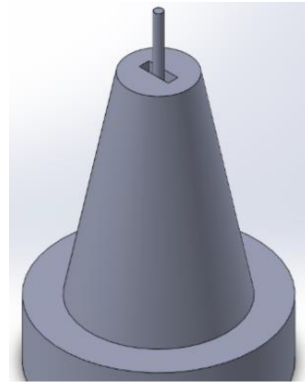


Figure 4.3 – Bloc modifié pour être soulevé par une tige en T.

Gonflement membrane

Ce système se rapproche de la tige en T. Il consiste en une membrane contractée au bout d'une tige. Une fois celle-ci insérée dans la fente du bloc, la membrane est gonflée par air comprimé dans une cavité au sein du bloc pour augmenter son volume. Le bloc est donc suspendu à la membrane qui empêche la tige de glisser.

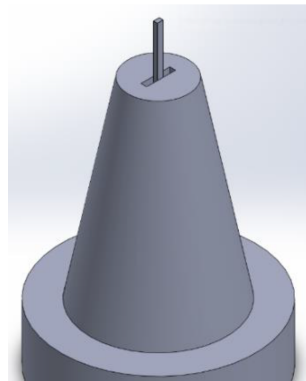


Figure 4.4 – Bloc modifié pour être soulevé par une membrane.

Électro-aimant

Ce dernier système est composé d'un électro-aimant suspendu sur le drone et d'une plaque métallique présente dans le bloc. L'aimant vient se déposer sur la plaque métallique au sommet du bloc. Le drone peut soulever le bloc grâce à la force électromagnétique unissant les deux éléments.



Figure 4.5 – Bloc modifié avec une plaque métallique surmontée d'un électro-aimant.

Système Choisi

Les avantages et désavantages de chacun des systèmes présentés sont repris dans le tableau récapitulatif de la figure 4.6.

Les deux systèmes à tiges nécessitent l'intégration de l'orientation du drone et une grande précision pour pouvoir être utilisés. Par facilité, ces deux systèmes n'ont pas été retenus pour être employés avec notre drone. Quant aux deux derniers systèmes, la modification d'un bloc et la consommation d'énergie n'étant pas des critères très pertinents comparés au poids et à l'encombrement, lors d'essais en laboratoire, nous avons choisi d'intégrer le système d'électro-aimant à notre drone. Celui-ci possède les avantages suivants :

- Robustesse du système.
- Faible encombrement et poids faible.
- Fonctionnement simple.
- Nul besoin de connaître l'angle de rotation du drone.

Électro-aimants	Pince de levage	Rotation de tiges à forme T	Gonflement membrane
+ dimensionnement et fabrication simples	++ bloc pas modifié	+ dimensionnement et fabrication simples	+ consommation d'énergie quasi nulle
+ encombrement	+ consommation d'énergie quasi nulle	+ consommation d'énergie quasi nulle	+ encombrement
+ poids	- poids	+ poids	+ poids (si bonbonne au sol ou rechargeable)
+ sécurité	- encombrement	+ encombrement	- mise en œuvre en laboratoire
+ mise en œuvre en laboratoire	- sécurité	+ coût de fabrication	- coût de fabrication
-- adaptation du bloc		-- adaptation du bloc	-- adaptation du bloc

Figure 4.6 – Tableau comparatif des différents systèmes de levage.

4.2 Fonctionnement du système

Le fonctionnement de ce système repose sur l'électro-aimant qui est embarqué sur le drone. Le but est de se positionner au-dessus d'un bloc dans lequel une plaque métallique est incorporée. Ensuite, par aimantation, une liaison se crée entre le bloc et le drone via l'électro-aimant. Cela permet ensuite au drone de voler jusqu'à la position finale du bloc et de déposer celui-ci en démagnétisant son électro-aimant. Plusieurs éléments sont nécessaires pour pouvoir faire fonctionner ce système de manière autonome. Ils sont décrits dans le schéma 4.7.

- **Électro-aimant** : explications données quant à son choix au point 4.3.
- **Ordinateur embarqué** : son rôle dans le système de levage est d'envoyer l'information à la carte électronique quand l'électro-aimant doit être magnétisé ou non.
- **Carte électronique** : son fonctionnement est expliqué précisément au point 4.4.1.
- **Lien entre ordinateur et carte électronique** : cet élément permet de traiter l'information relative au déclenchement du système envoyée par l'ordinateur et de la transformer en impulsion électrique à transmettre à la carte

électronique. Deux composants ont été envisagés pour jouer ce rôle. Ils sont décrits au point 4.4.

- **Batterie** : les mêmes batteries que celles utilisées pour alimenter les moteurs, le contrôleur de vol et l'ordinateur embarqué sont utilisées pour activer ou désactiver l'électro-aimant.
- **Guide** : cet élément permet de compenser une certaine imprécision du drone. Il est décrit au point 4.5.
- **Support** : cet élément est le lien mécanique qui soutient l'électro-aimant et son guide au drone. Son fonctionnement est décrit au point 4.6.

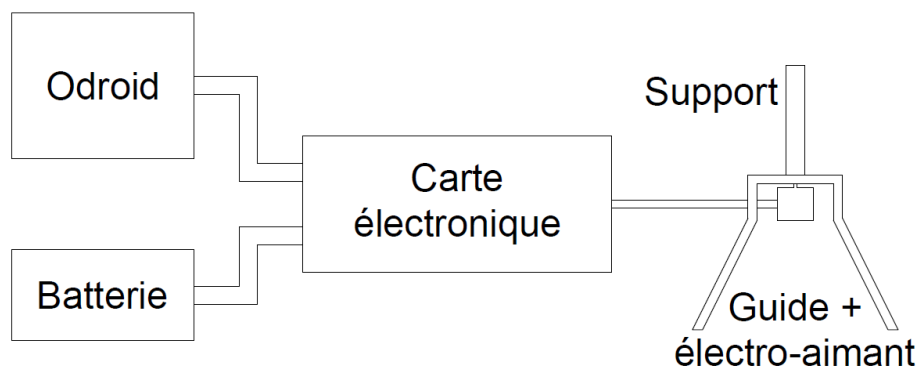


Figure 4.7 – Schéma du système de levage.

4.3 Quel type d'électro-aimant choisir ?

Les électro-aimants sont répartis selon deux catégories, à savoir : les électro-aimants permanents et les non-permanents. Leur différence réside dans le fait d'être magnétisé ou non quand aucun courant ne leur est injecté.

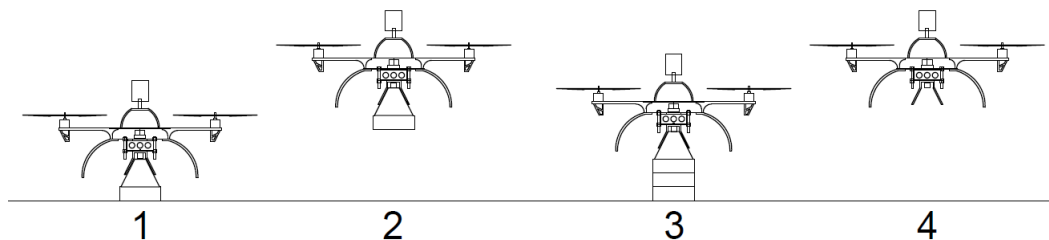


Figure 4.8 – Phase de prise et de pose d'un bloc.

La table 4.1 indique dans les deux dernières colonnes si l'électro-aimant, dépendant de son type, doit être alimenté en courant ou non, en fonction de sa phase de vol. Au vu de celle-ci, nous remarquons facilement qu'il est beaucoup plus intéressant d'utiliser un électro-aimant permanent pour un système fonctionnel. En effet, c'est la consommation d'énergie, problématique très importante dans l'utilisation des drones, qui nous intéresse ici. Au regard de cette optique, nous constatons que l'électro-aimant permanent consomme beaucoup moins que le non-permanent, principalement lors de la phase 2 (vol du drone avec un bloc) où l'électro-aimant permanent ne va pas consommer de courant contrairement au non-permanent. Plus le bloc transporté est lourd, plus le gain d'énergie est important.

Phase de vol	Aimantation	Permanent	Non-permanent
1. Prise du bloc	oui	pas de courant	courant
2. Vol avec le bloc	oui	pas de courant	courant
3. Pose du bloc	non	courant	pas de courant
4. Vol sans bloc	oui/non	pas de courant	pas de courant

Table 4.1 – Comparaison électro-aimants permanent et non-permanent

Cependant, malgré ces observations, dans le cadre de ce projet de recherche, nous utiliserons un électro-aimant non-permanent pour les phases de tests. En effet, il n'est pas toujours facile de désaimanter complètement un aimant permanent. Cela pose problème lors de l'utilisation de blocs de tests relativement légers (imprimés en 3D par exemple) qui peuvent rester accrochés malgré l'alimentation de l'aimant. Dans le cadre de ces phases de tests, l'autonomie de vol n'est pas un problème. Pour la suite des essais, quand de plus gros drones seront utilisés et donc de plus gros blocs seront soulevés, l'utilisation d'électro-aimants permanents sera donc privilégiée. Ces aimants ont comme avantage supplémentaire de garder le bloc attaché en vol si un souci d'alimentation venait à se produire. C'est un plus pour la sécurité du système.

Un dernier point doit attirer l'attention dans le choix de l'électro-aimant, c'est la force rémanente que possèdent certains de ceux-ci. Les électro-aimants possédant cette force gardent une force rémanente qui maintient le contact avec l'objet soulevé malgré la coupure de l'alimentation de l'électro-aimant. Ce phénomène n'est donc pas compatible avec notre système. En effet, notre but est de lâcher le bloc dès que l'ordinateur embarqué commande la coupure de la tension aux bornes de

l'électro-aimant.

4.4 Automatisation du fonctionnement du système

Dans le but de développer un système de construction autonome, la prise et la pose des blocs doivent être commandées par l'ordinateur embarqué. L'électro-aimant utilisé dans notre projet est caractérisé par une tension de 12 V et une résistance de 50 Ohms, il nécessite donc un courant de 240 mA pour être magnétisé au maximum. L'Odroid fournissant un courant maximum de 4 mA et une tension de 1.8 V, ne peut pas fournir le courant nécessaire à l'aimant. Pour cela, une carte électronique avec un relais est utilisée. L'Odroid contrôle ce relais pour que la batterie alimente ou non l'électro-aimant. La batterie fournit la tension et le courant nécessaires pour alimenter l'électro-aimant.

4.4.1 Fonctionnement de la carte électronique

Le but de cette carte électronique est de pouvoir déclencher l'électro-aimant depuis l'ordinateur embarqué, à l'aide de la batterie du drone.

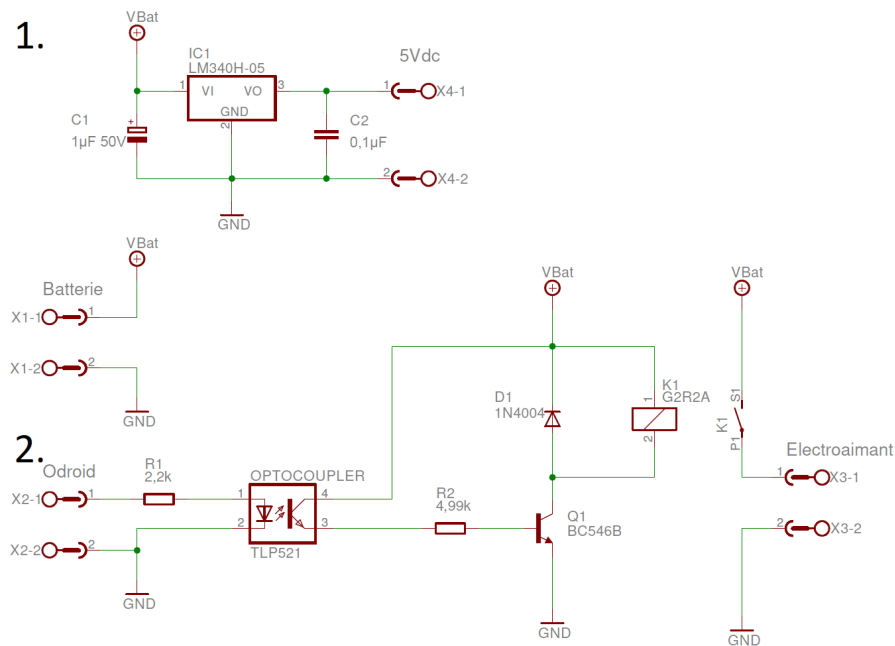


Figure 4.9 – Schéma de la carte électronique.

L'ensemble des éléments représentés sur la partie 2 du schéma, à la figure 4.9, permet l'alimentation de l'aimant contrôlé par l'Odroid. Pour cela, la carte prend en entrée un signal venant de l'Odroid sur la gauche du schéma. Le signal passe le transistor en état saturé, cela a pour conséquence d'activer le relais qui est alimenté par la batterie. Avec le courant passant dans sa bobine, le relais ferme le circuit qui permet à la batterie d'activer l'électro-aimant. Une fois le signal de l'Odroid coupé, le transistor passe en état bloqué. Cela a pour conséquence de ne plus alimenter la bobine du relais qui ouvre donc le circuit de l'électro-aimant.

L'optocoupleur présent juste après le signal d'entrée, permet d'isoler de manière galvanique deux parties de circuit. Pour cela, c'est une led émettrice d'infrarouge et un récepteur qui permettent la transmission du signal. L'état flottant du signal de l'Odroid rend cette isolation galvanique obligatoire. Sans celle-ci, lorsque l'Odroid commande la coupure du signal, une tension parasite le garde actif et l'alimentation de l'électro-aimant ne se coupe jamais.

Cette carte possède aussi une fonction secondaire : elle permet d'alimenter l'Odroid depuis la batterie du drone. Elle est représentée sur la partie 1 du schéma sur la figure 4.9. L'ordinateur nécessite une tension d'alimentation de 5 V. Un régulateur de tension linéaire est donc utilisé pour dissiper la tension de la batterie (entre 14.8 et 16.8 V) jusqu'à une tension de 5 V.

4.4.2 Composant intermédiaire

Cependant, un composant intermédiaire est utilisé entre l'Odroid et la carte électronique. Cet élément permet entre autres d'avoir une barrière électronique entre la carte et l'ordinateur. En effet, nous ne sommes pas à l'abri d'un court-circuit ou d'un dysfonctionnement de la carte électronique. Sans la présence de cet élément, l'ordinateur pourrait être directement affecté par cette potentielle défaillance. L'ordinateur contrôlant entre autres le vol du drone, il n'est pas permis qu'une défaillance du système de levage entraîne des répercussions sur le contrôle du drone. D'un point de vue financier, remplacer cet élément est bien plus avantageux que de changer l'ordinateur en cas de dommages non-réversibles. Deux options ont été étudiées pour cet élément : un micro-contrôleur et le shifter shield de l'Odroid XU4.

Micro-contrôleur

C'est une carte électronique rassemblant tous les éléments importants d'un ordinateur comme un processeur, une horloge de cadence, ou encore une mémoire. Ces composants se trouvent être très efficaces d'un point de vue de la consommation d'énergie lorsqu'ils sont affectés à une tâche précise (ici, transformer une information

en impulsion électrique). Dans notre cas, nous avons utilisé la carte Mega 2560¹ de la firme *Arduino*, montrée à la figure 4.10.

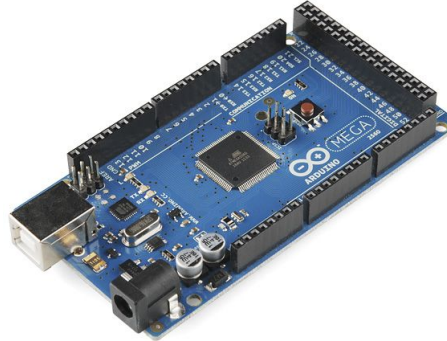


Figure 4.10 – Micro-contrôleur Arduino Mega 2560.

Ce micro-contrôleur ne peut faire tourner qu'un programme à la fois, ce dernier est transmis depuis un ordinateur. Tous les programmes qui tournent sur l'*Arduino* suivent un schéma particulier en 2 parties : une initialisation et une boucle. Dans notre cas, ils sont très simples. La première partie permet d'initialiser le périphérique de sortie du courant électrique (ici, le pin 13), et la seconde permet d'envoyer ou non du courant.

Premier programme

```
1 void setup() {  
2     pinMode(13,OUTPUT);  
3 }  
4 void loop() {  
5     digitalWrite(13,HIGH);  
6 }
```

Second programme

```
1 void setup() {  
2     pinMode(13,OUTPUT);  
3 }  
4 void loop() {  
5     digitalWrite(13,LOW);  
6 }
```

Comme on le voit dans le code ci-dessus, le premier programme est envoyé au micro-contrôleur quand l'ordinateur veut activer l'électro-aimant. L'*Arduino*

1. <https://www.arduino.cc/en/Guide/ArduinoMega2560>

va donc effectuer le setup de la pin 13, et ensuite fournir en permanence 5 V sur cette pin. Quand on veut désactiver l'aimant, le second programme est envoyé et, après un setup de la pin, le programme n'envoie rien. Cette solution comporte un désavantage : la complexité informatique pour envoyer et changer à chaque fois le programme. En effet, le plus simple serait d'envoyer une variable à l'*Arduino* qui changerait l'état de la pin en fonction de l'utilisation de l'électro-aimant, malheureusement l'*Arduino* ne le permet pas. Nous n'avons donc techniquement pas réussi à mettre cette solution en place. C'est pourquoi nous avons exploré une autre option : le shifter shield.

Shifter shield

Le shifter shield² est un élément complémentaire à l'Odroid XU4 utilisé comme ordinateur embarqué dans ce projet. Ce composant vient se greffer sur les pins GPIO (General Purpose Input/Output) de l'Odroid. Il permet de convertir les deux GPIOs 12 et 30 broches de l'Odroid vers un unique GPIO 40 broches. Les pins GPIO de l'Odroid XU4 ont un voltage de sortie de 1.8 V, qui n'est pas un standard. Le shifter shield est équipé d'un interrupteur permettant de choisir le voltage de sortie, 3.3 ou 5 V qui est plus standardisé dans le domaine de l'électronique. Le voltage de sortie de l'Odroid est remonté par 3 "bi-directional level shifter" présents dans le shifter shield. Celui-ci fournit un courant de 4 mA maximum à travers ses broches. Deux de celles-ci vont donc être utilisées pour envoyer une tension de 3.3 V à la carte électronique dans le but de déclencher l'électro-aimant.

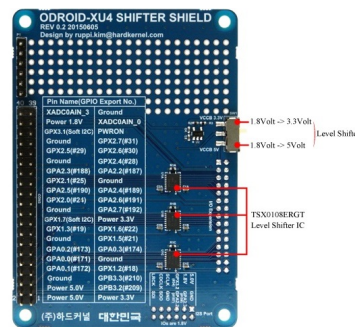


Figure 4.11 – shifter shield.

Pour cela, la librairie *WiringPi*³ a été utilisée. Elle est utilisable en C ou Python sur Odroid et permet de facilement activer ou non les pins du GPIO du shifter shield. Le code ci-dessous présente la façon dont elle est utilisée dans notre projet.

2. <https://www.hardkernel.com/shop/xu4-shifter-shield/>
3. https://wiki.odroid.com/odroid-xu4/application_note/gpio/wiringpi

```

1  #Setup la pin 0 en mode OUTPUT
2  import wiringpi as wpi
3  wpi.wiringPiSetup()
4  wpi.pinMode(0, 1)
5
6  #Active la pin 0
7  wpi.digitalWrite(0, 1)
8
9  #Desactive la pin 0
10 wpi.digitalWrite(0, 0)

```

La ligne 4 permet de setup la pin 0 de *WiringPi* sur le mode OUTPUT, alors que les lignes 7 et 10 permettent respectivement d'activer ou de désactiver la tension sur la pin 0. Cette pin 0 correspond à la pin 11 sur le shifter shield. Il suffit d'utiliser la commande **\$ gpio readall** pour connaître la carte du GPIO ainsi que ses valeurs pour *WiringPi*.

Pour déclencher le système de levage, il suffit donc de mettre les lignes de setup au début du nœud de contrôle ROS utilisé pour le système de guidance et utiliser les lignes de commande 7 et 10 pour activer ou désactiver le système de levage et ainsi porter ou déposer un bloc de construction.

4.5 Guide de levage du drone

L'utilité d'un guide dans le système de levage est de compenser l'imprécision possible du drone pour que celui-ci soit parfaitement centré par rapport au bloc lors de sa prise, comme montré figure 4.12. Cela permet d'aligner les centres de gravité des deux éléments, ce qui est essentiel pour éviter de déstabiliser le drone en vol. Dans le cadre d'un système avec électro-aimant, il est très important que celui-ci soit bien centré sur la plaque métallique du bloc. En effet, une imprécision du drone entraînerait un contact partiel de l'électro-aimant, ce qui réduirait la force magnétique de celui-ci et donc diminuerait la charge qu'il est capable de soulever.

Pour ce faire, le guide a une forme large à sa base pour couvrir une grande surface autour de l'axe de gravité du drone. Le guide s'affine de manière croissante dans le but d'épouser l'extrémité supérieure du bloc à transporter. L'imprécision que le guide est capable de compenser se caractérise par la distance entre les axes de gravité du drone et du bloc, lorsque l'extrémité du guide est au niveau de l'extrémité supérieure du bloc. Cette distance est représentée par la donnée géométrique X dans la figure 4.13.

On comprend rapidement que, plus le guide est large et long, plus l'imprécision que le guide peut compenser est grande. Cependant, nous sommes limités par deux facteurs :

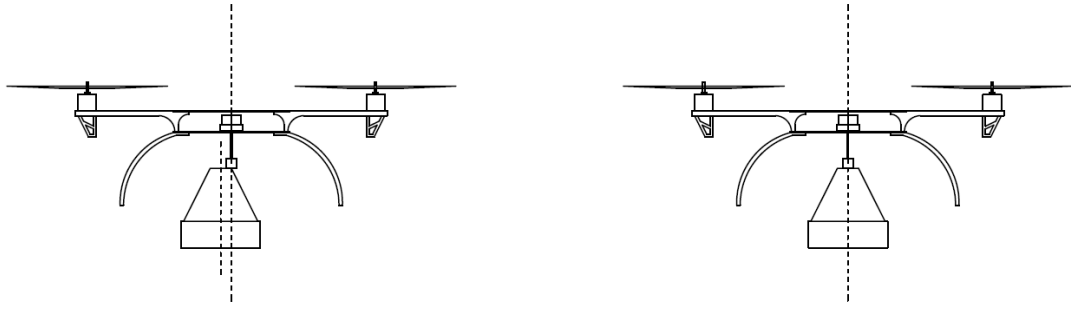


Figure 4.12 – Mauvais alignement vs bon alignement pendant la prise d'un bloc.

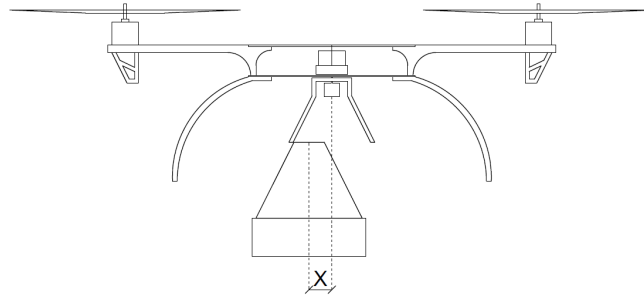


Figure 4.13 – Imprécision qu'un guide est capable de compenser.

- La largeur du guide par rapport au bloc : en effet, un guide plus large autorise une plus grande imprécision. On peut donc être tenté de réaliser un guide très large. Cela pose problème si le guide n'est pas assez pentu, il risque de se poser de travers sur le bloc sans s'emboîter dans celui-ci. La bonne pente d'un guide est à déterminer sur la base d'essais.
- La distance sous le drone : la taille du cône est limitée par la place sous le drone. Effectivement, si le guide venait à dépasser la longueur des pieds du drone, celui-ci ne pourrait se reposer sur ceux-ci lors du décollage et de l'atterrissage ce qui pose évidemment un problème de stabilité.

4.6 Support

Le support est l'élément mécanique qui va soutenir l'électro-aimant et son guide sous le drone. Sa conception n'est pas à laisser au hasard quand on utilise un électro-aimant comme système de levage. En effet, pour prendre un bloc, celui-ci se magnétise et dégage donc un champ magnétique autour de lui. Si l'électro-aimant se situe trop près du contrôleur de vol, il perturbe ses compas internes qui sont

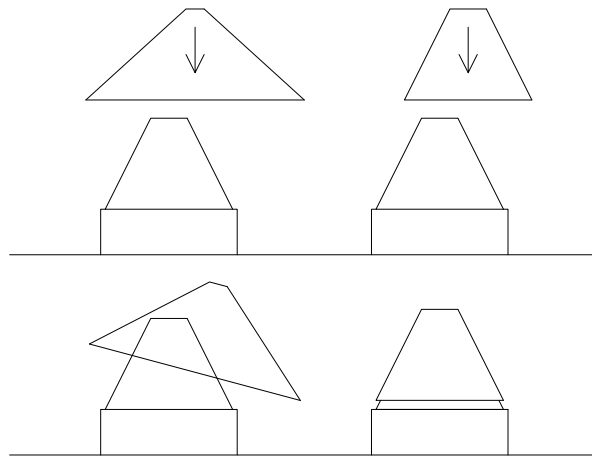


Figure 4.14 – Guide mal conçu vs bien conçu.

indispensables pour faire voler le drone. Pour éviter cela, il est nécessaire d'éloigner l'électro-aimant du contrôleur de vol. Deux solutions ont été envisagées :

- Utiliser des pieds plus grands et un support plus long, par exemple une longue tige filetée.
- Utiliser un support télescopique qui s'allonge au décollage et se referme à l'atterrissage. Il suffit dans ce cas d'activer l'électro-aimant uniquement en vol quand le support est déployé.

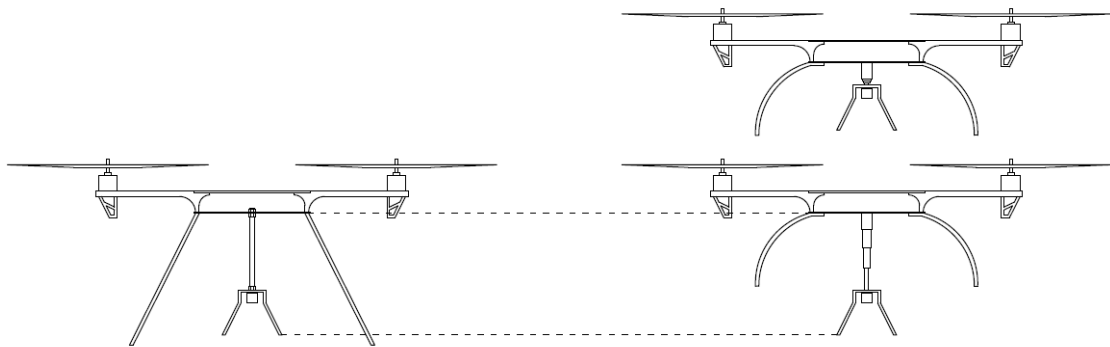


Figure 4.15 – Différence entre les deux types de supports.

Chapitre 5

Essais réalisés

Afin de tester le système complet, nous avons effectué différents types de tests. Les premiers types d'essais développés avaient pour but de s'assurer que le drone est fonctionnel, que ses moteurs sont capables de soulever le poids du drone et ses divers équipements. Ils ont été réalisés de manière manuelle à l'aide de la télécommande.

Nous avons ensuite testé la précision des systèmes de positionnement en déplaçant le drone à la main. Une fois cette étape validée et les résultats suffisamment bons, nous avons effectué des vols autonomes. Ces vols étaient de plus en plus sophistiqués, dans le but de prouver le bon fonctionnement et les possibilités du système, mais aussi de quantifier sa précision. Les derniers essais réalisés concernaient le système de levage, là encore le but recherché était de valider son fonctionnement.

L'ensemble des tests présentés dans ce chapitre ont été réalisés dans la zone de l'UCLouvain avec le petit drone.

5.1 Tests manuels

Trois tests ont été réalisés manuellement. Tous visant à vérifier le bon fonctionnement du drone. Le premier test est basique puis les 2 suivants augmentent en complexité. La liste des tests est donnée ci-dessous :

1. Vol stationnaire à 1 m du sol, non-chargé.
2. Vol en mouvement à différentes altitude et position, non-chargé.
3. Vol en mouvement à différentes altitude et position, chargé.

La différence entre les deux premiers tests et le dernier est l'ajout de tous les éléments non-indispensables au vol du drone mais nécessaires pour le fonctionnement de notre système (le prisme, la station totale, l'Odroid, la caméra, le système de levage, etc.)



Figure 5.1 – DroneZone de l’UCLouvain.

Le critère de validation de ces tests était visuel, le but était d’observer les réactions du drone aux commandes et sa façon de voler.

Les conclusions de ces tests sont :

- Le *Pixhawk* s’allume correctement et les moteurs s’arment bien.
- Le drone est stable en vol et facilement maniable, l’ajout de la charge ne le déséquilibre pas.
- Le drone vole normalement malgré l’alimentation de l’électro-aimant.
- Il atterrit sans problème.

5.2 Tests des systèmes de positionnement

L’ensemble des tests liés aux systèmes de positionnement ont été réalisés à la main, c’est-à-dire sans faire voler le drone. Une première série fut consacrée à étudier la précision des systèmes de manière statique, et la seconde de manière dynamique, le drone étant déplacé à la main. Quatre données nous ont intéressés lors de ces tests :

- **Total** : position du prisme donnée par la station totale.
- **UWB** : position de la balise fournie par celle-ci par rapport aux coordonnées des ancrs UWB.

- **Caméra** : position de la caméra donnée par elle-même par rapport à son lieu de départ fixé à $(0, 0, 0)$.
- **Local** : position du *Pixhawk*. Elle est présentée au point 3.2.2.

5.2.1 Les problèmes

Tout d’abord, nous avons cherché à savoir comment le *Pixhawk* intègre les positions que nous lui envoyons. Nous avons donc envoyé les données de l’UWB au *Pixhawk* et nous avons comparé ces données à celles intégrées par le contrôleur de vol.

La figure 5.2 montre les deux positions. Nous voyons que le *Pixhawk* n’intègre pas du tout la position de l’UWB lorsque le drone est déplacé. On remarque cependant un léger mouvement de la position locale, cela doit être dû à un instrument interne.

Après vérification, ce problème était dû à un mauvais paramétrage de *LPE vision*, l’estimateur du *Pixhawk* qui intègre la position extérieure.

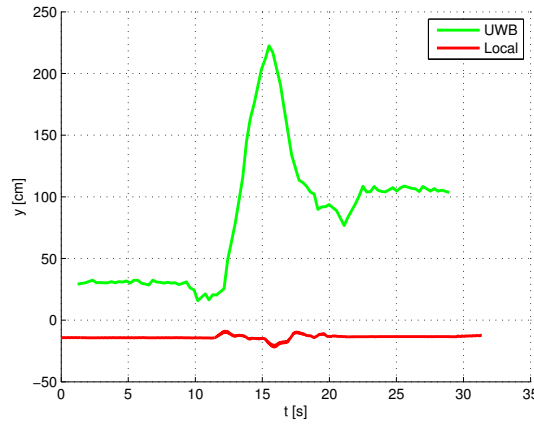


Figure 5.2 – Mauvaise intégration du *Pixhawk* d’une position donnée extérieure.

Un second problème s’est présenté quand nous avons observé la position y de la station totale. En effet, comme le montre la figure 5.3, à certain moment, la donnée y de la station totale nous donne une valeur négative égale en valeur absolue à sa réelle position. Une erreur de signe s’était glissée dans le script d’envoi des positions de la station. Cette erreur ne se présente pas sur toute la plage de données car elle n’arrivait qu’une fois que la station perdait en précision.

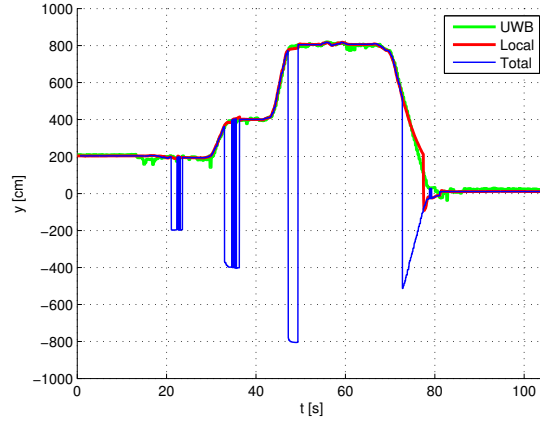


Figure 5.3 – Valeurs négatives en y de la station totale.

5.2.2 Quantifier la précision

Statique

Nous avons quantifié la précision des trois systèmes de positionnement (station totale, UWB et caméra) de manière statique. Pour cela, les données de chaque système ont été enregistrées quand le drone est laissé au niveau du sol sans bouger.

Les graphes ci-dessous représentent des BoxPlots qui expriment la dispersion de chaque variable (x , y , z) pour chaque système. Chaque boxPlot est recentrée par rapport à sa moyenne pour une meilleure lisibilité. Pour rappel, la ligne rouge d'une boxPlot définit la médiane des données, les lignes bleues encadrent 50% des données (0.25 - 0.75), et les traits noirs représentent les valeurs maximales et minimales de l'échantillon de données. Les possibles valeurs aberrantes sont représentées par des croix rouges à l'extérieur de la boxPlot.

Les données de l'UWB ont une précision entre 6 et 8 cm. Celle de la station totale est comprise entre 2 mm et celle de la caméra est inférieure au millimètre malgré quelques valeurs aberrantes.

En comparant les moyennes des données des trois systèmes pour une même variable, cela permet d'observer les offsets de chaque système par rapport aux autres. Ces différences sont utilisées pour compenser ce décalage dans le fichier de configuration du système. Les graphes montrant les offsets entre l'UWB et la station pour chaque variable se trouvent à l'annexe [A](#).

Dynamique

Pour comparer les différentes positions en mouvement, nous avons besoin d'un référentiel. Ce référentiel doit être un système de positionnement plus précis que

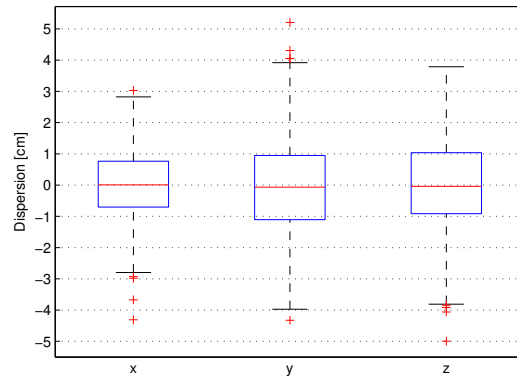


Figure 5.4 – Dispersion des données statiques de l’UWB.

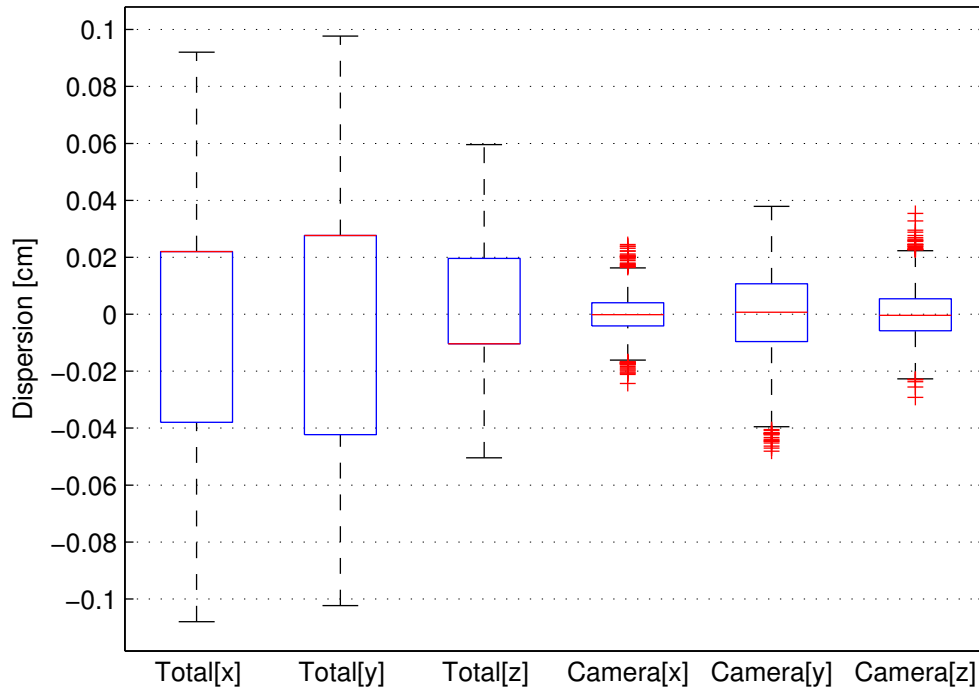


Figure 5.5 – Dispersion des données statiques de la station totale et de la caméra

ceux testés. Ce référentiel peut être mécanique, par exemple un rail qui garantit le

mouvement du drone, ou dans notre cas, la station totale qui est le système le plus précis. N'ayant pas de référentiel plus précis que la station totale, nous n'avons donc pas pu quantifier sa précision de façon dynamique. Nous pouvons néanmoins penser que sa précision est aussi bonne en dynamique qu'en statique si ses données envoyées sont précises. L'imprécision du mouvement vient du lag entre la prise de mesure et l'intégration de celle-ci par le *Pixhawk*.

Pour effectuer ces mesures de précision dynamique, le drone a été déplacé à la main selon un chemin défini, présenté à la figure 5.6. Ce plan de la drone zone permet de visualiser le chemin suivi pendant ce test, il permet aussi de voir l'emplacement des ancres UWB (B0, B1, B2) et d'observer que la station totale est bien à l'origine du plan, comme expliqué au point 2.7.

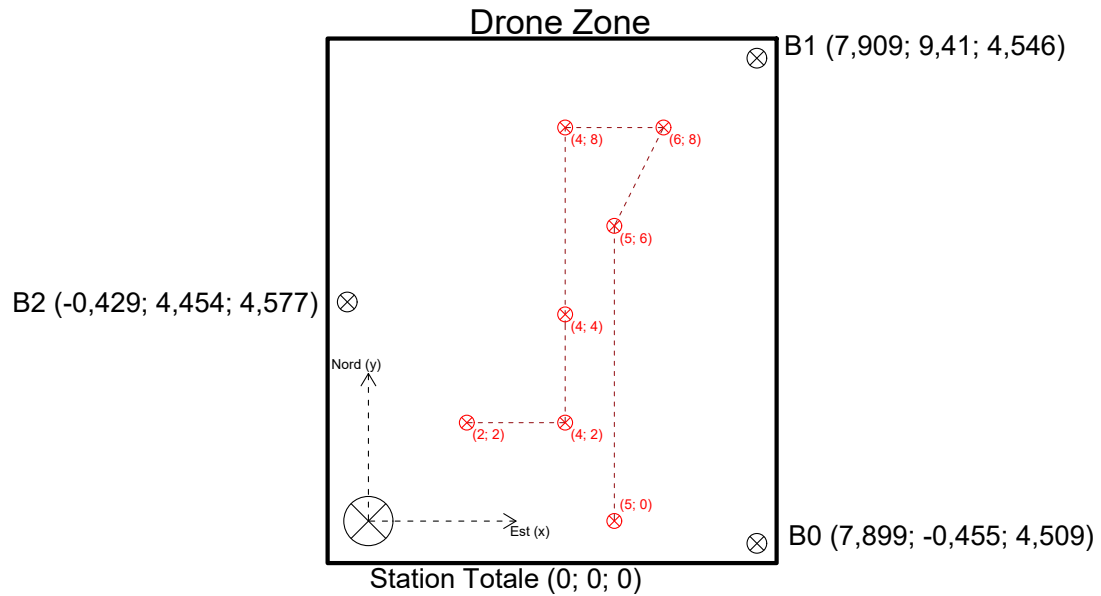


Figure 5.6 – Plan du chemin réalisé en drone zone pour l'essai dynamique.

La figure 5.7 présente les boxPlots de chaque axe pour chaque système, représentant leur dispersion par rapport au référentiel, à savoir la station totale. Pour la position locale, la position extérieure intégrée par le *Pixhawk* est celle de la station totale.

On constate que l'UWB possède une imprécision maximale de 30 cm ce qui était attendu. Par contre, on constate que la précision de la caméra en mouvement n'est pas du tout satisfaisante, l'écart est supérieur à 1,5 m en y . Dans la figure 5.8, toutes les données x et y de tous les systèmes, on remarque que la caméra est la seule à dévier du chemin effectué manuellement. La proximité des données de l'UWB et de la station totale comparé à celle de la caméra nous permet d'affirmer que le problème rencontré est bien dû à la caméra elle-même et non aux autres

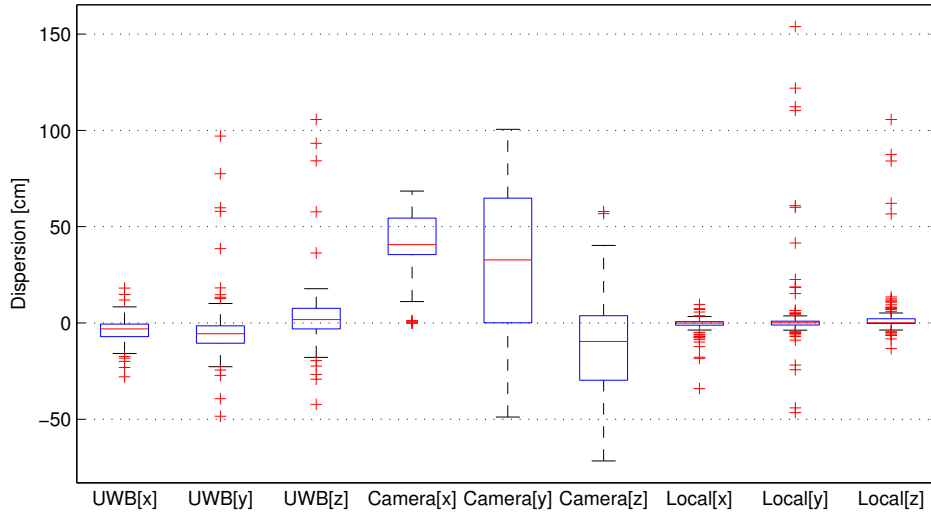


Figure 5.7 – Dispersion des données de chaque axe, pour chaque système par rapport à la station totale en dynamique.

systèmes, d'où l'avantage de travailler avec 3 systèmes différents. Ce problème peut être dû à une mauvaise intégration des données de la caméra ou à un manque de contraste et donc une difficulté pour la caméra de trouver des points d'accroche dans la drone zone.

5.3 Tests de vol autonome

5.3.1 Vol stationnaire

Afin de connaître la précision du drone en vol stationnaire, nous avons enregistré ses positions en vol stationnaire autonome (c'est-à-dire uniquement guidé par l'ordinateur embarqué). La trajectoire du drone pendant ce vol est représentée en 2D et 3D aux figures 5.9 et 5.10. Le cerle et la sphère représentés sur ces figures montrent la distance maximale que le drone a atteint par rapport à sa position de vol stationnaire, à savoir $(0, 0, 0)$.

Nous remarquons dans la figure 5.11 que la variation de position est de plus de 50 cm par rapport à sa position de vol stationnaire. Cette imprécision est donc bien trop grande pour espérer prendre ou déposer un bloc. Cela confirme la nécessité de recourir à un système de positionnement local pour ces phases d'une mission de construction.

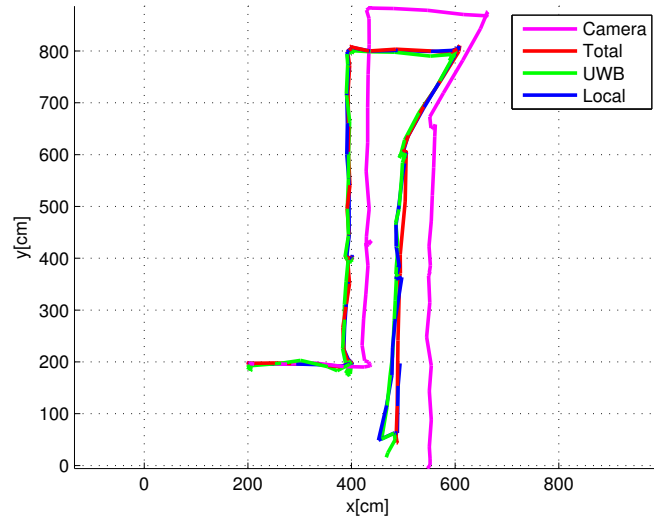


Figure 5.8 – Plan des données de chaque système lors de l’essai dynamique.

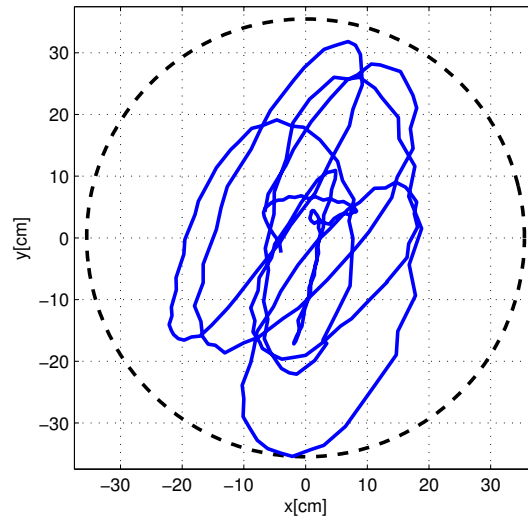


Figure 5.9 – Plan de la trajectoire du drone pendant 48 secondes en vol stationnaire.

5.3.2 Mission

Une fois que le drone volait bien de manière stationnaire, nous avons testé ses réactions et sa précision en déplacement. Les premiers vols étaient juste des déplacements dans une direction. Ceux-ci étant concluants, nous avons testé le drone en lui faisant réaliser les mêmes actions que lors d’une mission de construction.

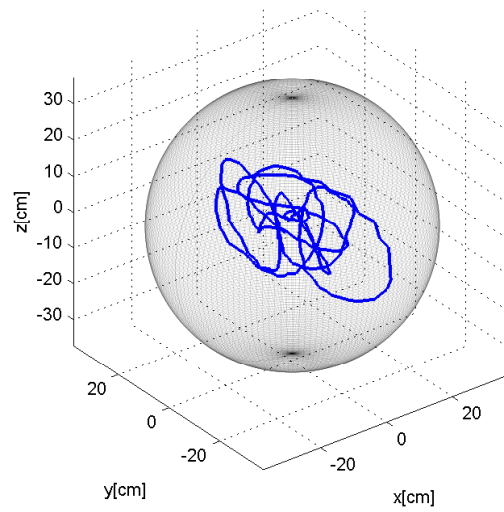


Figure 5.10 – Représentation 3D de la trajectoire du drone pendant 48 secondes en vol stationnaire.

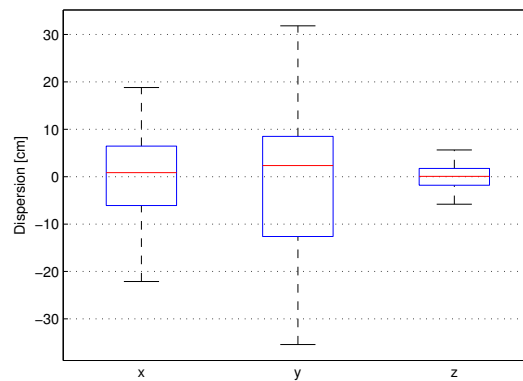


Figure 5.11 – Dispersion des données pour chaque axe en vol stationnaire.

Les différentes étapes sont décrites ci-dessous :

1. Décoller à 2 m.
2. Voler jusqu'à la zone de stockage.
3. Descendre à 1 m pour être suffisamment proche pour scanner les blocs avec un système local (il n'est pas implémenté ici mais nous simulons son intégration dans le système).

4. Déplacement dans la zone de stockage (simulation du comportement avec un système local).
5. Bloc trouvé, descente pour le prendre.
6. Monter jusqu'à 2 m.
7. Descendre à l'endroit du bloc (atterrissage du drone dans notre cas).

La hauteur de vol est de 2 m et la hauteur de scan est égale à 1 m dans cet essai.

Les trois figures 5.12, 5.13 et 5.14 montrent respectivement les positions x , y et z de tous les systèmes en fonction du temps. Le trait noir représente le setpoint envoyé au *Pixhawk* que le drone doit atteindre. Une fois atteint, le setpoint suivant est envoyé au *Pixhawk*.

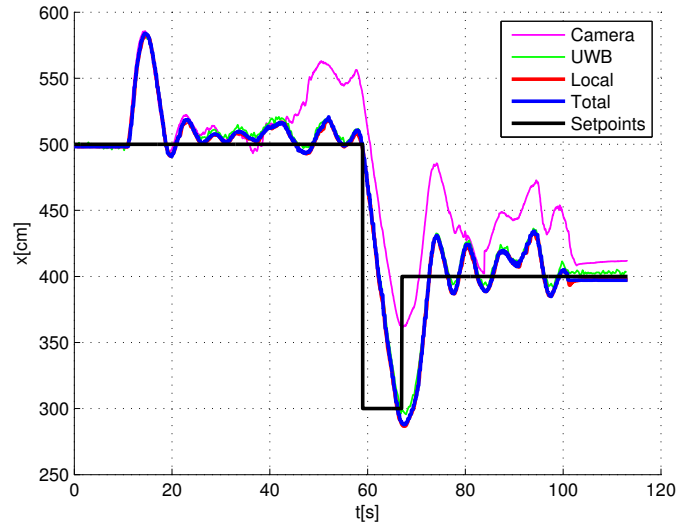


Figure 5.12 – Positions et setpoint en x en fonction du temps lors de la mission.

Nous pouvons voir plusieurs choses sur ces différents graphiques.

Premièrement, pour l'axe des x , nous observons que le drone a tendance, à chaque décollage, à avancer d'un peu moins d'un mètre. Une hypothèse est que le drone n'est pas bien équilibré en raison de la présence de la caméra à l'avant du drone, ce qui induit un mouvement vers l'avant lorsque le drone quitte le sol, ce qui est ensuite compensé par le contrôleur de vol.

Deuxièmement, pour l'axe z , nous pouvons voir que lorsque le drone grimpe pour atteindre un setpoint, il le fait assez rapidement jusqu'à ce qu'il soit à une trentaine de centimètres de la cible, où il ralentit pour atteindre précisément sa cible. Ce comportement est spécifique au contrôleur de vol.

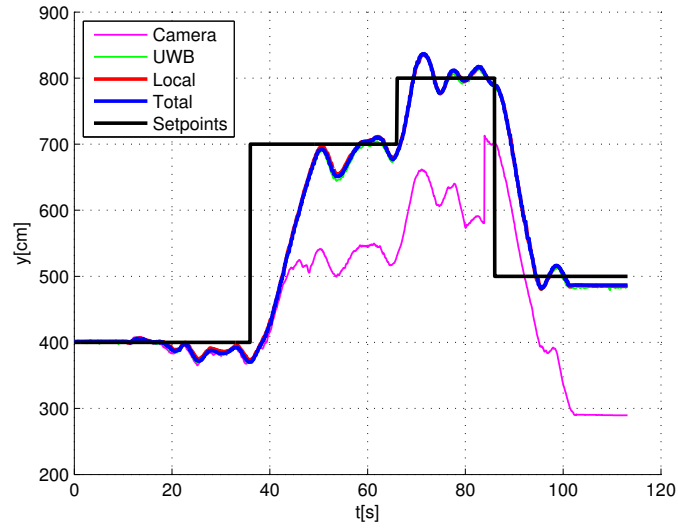


Figure 5.13 – Positions et setpoint en y en fonction du temps lors de la mission.

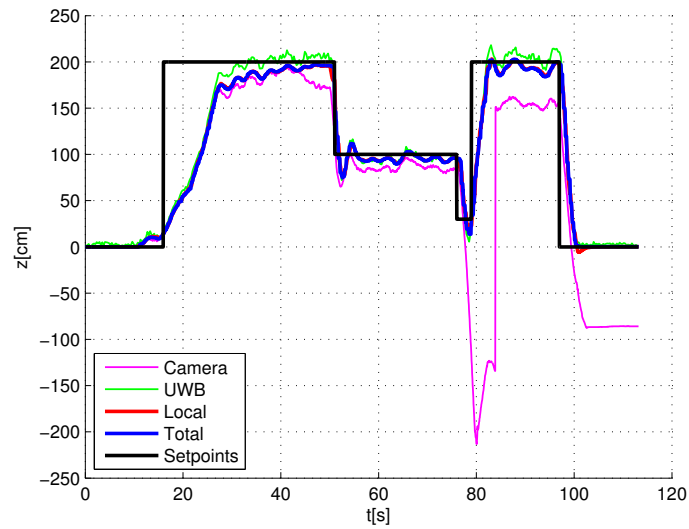


Figure 5.14 – Positions et setpoint en z en fonction du temps lors de la mission.

Troisièmement, toujours pour l'axe z , on remarque que lorsque le drone fait une plongée, il descend rapidement, la caméra exagère le mouvement et, dans ce cas, descend de plus de deux mètres trop bas, à $t = 80$ s. L'erreur est partiellement corrigée pendant le téléchargement mais est répétée dans $t = 100$ s. Encore une fois cette problématique peut être due à une mauvaise intégration des données de

la caméra ou à un manque de contraste dans la drone zone.

5.4 Tests de levage de bloc

La dernière série d'essais menée pendant notre travail concerne le système de levage.

Comme présenté au début du chapitre, tous les tests réalisés utilisent le petit drone. Celui-ci n'étant capable de porter qu'une charge de l'ordre du kilo en plus de tout le matériel qu'il transporte, il a fallu imaginer de nouveaux blocs adaptés à ses capacités. Ces blocs sont bien évidemment drone-compatibles.

5.4.1 Le cône de test

Ce premier bloc imaginé pour les tests a une forme conique. Cette forme permet de ne pas tenir compte de l'orientation du drone et donc de son système de levage. En effet, les premiers tests réalisés n'intègrent pas la position du drone, un élément de forme conique permet de ne pas tenir compte de cette information, contrairement à un élément rectangulaire. Le cône a pour but d'être soulevé par un électro-aimant, une plaque métallique est donc collée sur sa base supérieure. Il joue le rôle de modèle réduit de vrais blocs de construction drone-compatibles. Il est imprimé en 3D pour être plus rapidement produit et plus léger.

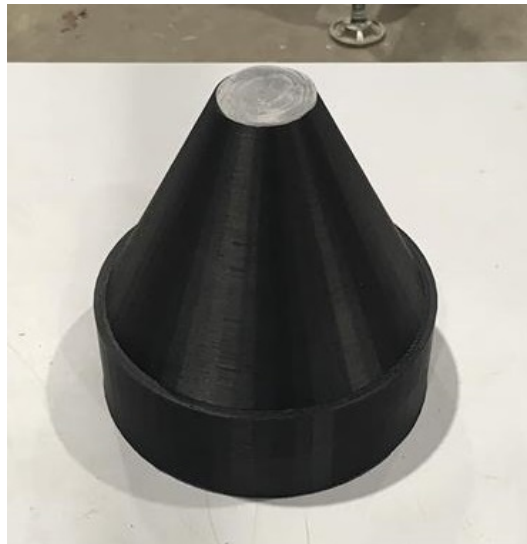


Figure 5.15 – Cône 1.2.

Cet élément structurel est utilisé pour la construction de colonnes. Avec sa base droite, il possède une hauteur d'empilement de 5 cm. Ce qui signifie qu'il faut 20

blocs pour atteindre 1 m (si on ne tient pas compte de la partie conique du dernier). Son guide possède lui aussi deux parties, une inférieure conique pour épouser la forme du bloc, et une cylindrique supérieure pour y placer l'électro-aimant. Toutes les caractéristiques de ces deux éléments se trouvent à l'annexe [B](#).



Figure 5.16 – Guide du Cône 1.2.

5.4.2 Essais de prise de bloc

Les premiers essais ont été réalisés de façon manuelle, le drone étant déplacé à la main. En plaçant le drone sur un cône et en activant l'électro-aimant depuis l'ordinateur embarqué, nous avons réussi à soulever le cône. En désactivant l'aimant, le bloc se décroche bien et chute. Le contrôle de l'électro-aimant depuis l'ordinateur embarqué est donc validé.

Nous avons aussi pu faire fonctionner l'électro-aimant en vol manuel, c'est-à-dire avec la télécommande. Le système fonctionne bien.

5.4.3 Vol autonome avec un bloc

L'absence de système de positionnement local, utilisant par exemple le traitement d'images, ne nous permet pas d'avoir la précision nécessaire pour positionner le drone au-dessus d'un bloc. Nous avons donc choisi de tester l'intégration du système de levage uniquement en lâchant un bloc. Le but est donc de décoller directement

avec un bloc. À ce moment-là, un problème survient : sur le petit drone, l'électro-aimant est trop proche du *Pixhawk* et, par conséquent, son champ magnétique interfère avec les compas de ce dernier. Le drone ne peut pas être armé et donc décoller parceque les compas ne sont pas disponibles.

Deux solutions sont possibles : la première consiste à utiliser des pieds plus grands et un support plus long. La seconde consiste à utiliser un support télescopique qui s'étend au décollage et se referme à l'atterrissage. Dans ce cas, il suffit d'activer l'électro-aimant en vol uniquement lorsque le support est déployé. Les deux systèmes sont illustrés sur les figures 5.17 et 5.18.

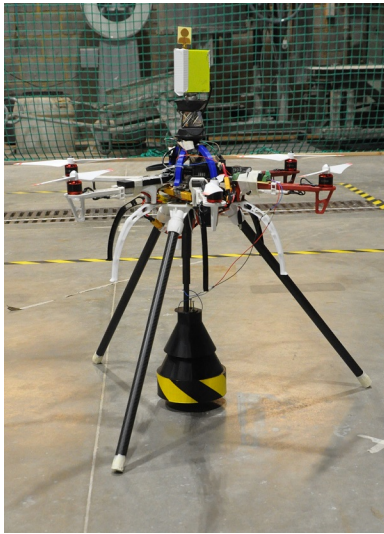


Figure 5.17 – Le petit drone avec de long pieds.



Figure 5.18 – Bras télescopique déplié

Nous avons finalement utilisé les 3 pieds comme solution, par facilité. Avec cette disposition, nous avons réussi à voler de manière autonome et à lâcher le bloc au moment voulu comme le montre la figure 5.19.

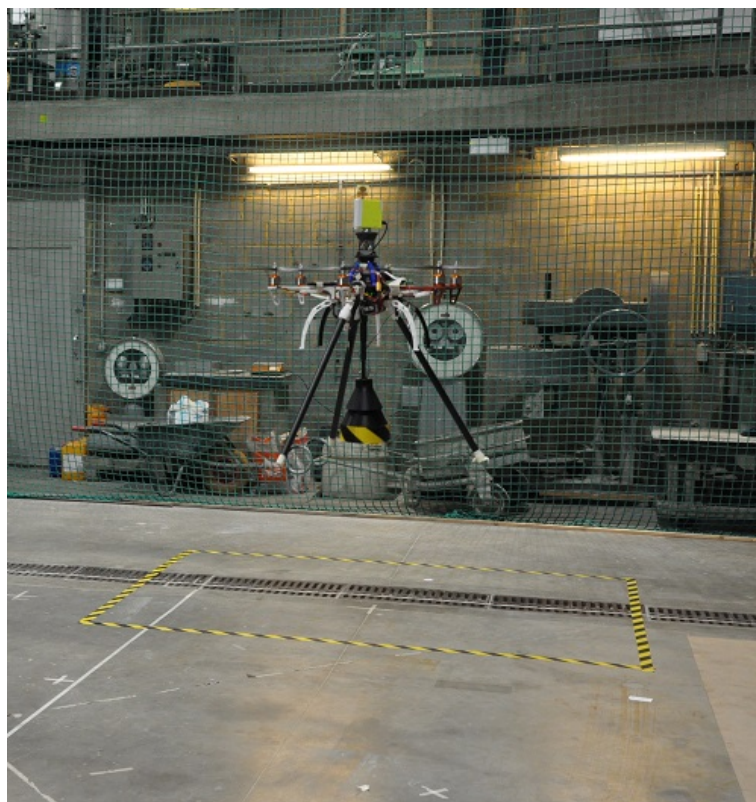


Figure 5.19 – Vol Autonome avec un bloc

Conclusion

Ce travail présente tous les éléments nécessaires pour concevoir un système de construction autonome par drone. Pour commencer, ce travail a décrit l'ensemble des composants hardware nécessaires ainsi que leur connexion respective. L'ordinateur embarqué et le contrôleur de vol, ainsi que leur utilisation dans ce système, ont également été présentés. Ensuite, les différents systèmes de positionnement, leurs caractéristiques et la façon dont ils fonctionnent ensemble ont été abordés.

Dans un second temps, les trois systèmes de positionnement, la station totale, l'UWB et la caméra que nous avons utilisés ont été détaillés. Les avantages de chacun ont été utilisés pour combler les défauts des autres dans le but de fournir la position la plus pertinente à tout moment d'une mission de vol. Pour cela, les étapes de leurs mise en place dans un repère unique ont été énumérées.

Pour continuer, les enjeux du système de guidage ont été développés. Une mission type de construction a été décomposée pour en comprendre toutes les étapes. Cette décomposition a également permis de faciliter le développement du système.

Après cela, le dernier système a été abordé : le système de levage. Toutes ses problématiques et les réponses à celles-ci ont été parcourues, aussi bien d'un point de vue mécanique que des points de vue électronique et informatique.

Pour finir, l'ensemble des tests qui nous ont permis de développer les systèmes et leurs résultats ont été présentés.

Les derniers tests réalisés nous ont permis d'effectuer un trajet typique d'une mission de construction, et nous avons réussi à voler avec un bloc et à le lâcher quand nous le souhaitions. Et cela de manière parfaitement autonome. Le manque d'un système de positionnement local ne nous a pas permis de prendre un bloc au sol avec le drone. Mis à part cela, nous pouvons dire que les objectifs fixés au début de notre travail ont été atteints.

Pistes d'avancement

Malgré cela, encore beaucoup de recherches, de développement et de tests seront nécessaires pour que le système soit fonctionnel à échelle réelle.

Une première piste d'avancement concerne la caméra. Ce système de positionnement semble très prometteur. En effet, la caméra étant entièrement embarquée sur le drone, les positions transmises sont très fiables. Ce système se base sur des images en temps réel, ce qui rend le système très utile pour éviter des obstacles potentiels non-prévus, que les autres systèmes ne peuvent pas prévoir. Cependant, nous n'avons pas réussi à l'implémenter de façon efficace. Il serait très intéressant de pousser plus loin le travail sur ce système.

Il sera également important de développer un système de positionnement local comme mentionné dans ce travail. Ce système est indispensable pour guider le drone lors la prise et la pose d'un bloc de construction.

Comme montré plus haut, la précision en vol du drone n'est pas encore assez suffisante malgré la bonne précision de la station totale par exemple. Travailler sur ce point permettrait d'augmenter cette précision et ainsi avoir un système plus sûr.

Pour ce même objectif, il sera important d'implémenter des procédures de sécurité plus importante. Il faudrait prévoir des mesures quand l'ordinateur perd le signal, quand le *Pixhawk* ne reçoit plus de positions, etc.

Ensuite, notre système étant déjà fonctionnel, il sera intéressant de l'implémenter sur les drones plus gros de l'UCLouvain. En effet, les réactions du système risquent de varier en changeant de taille de drone. L'imprécision en vol est aussi sûrement due à la nervosité du petit drone.

Pendant notre travail, nous avons souvent été contraints par la drone zone (station totale trop proche du drone, manque de contraste pour la caméra, pas de long trajet possible, interférence possible pour l'UWB, etc.) Pour cela, il serait très intéressant et indispensable de refaire et de continuer les tests effectués pendant notre travail, cette fois en extérieur. Cela permettrait aussi de découvrir des problèmes propres aux vols de drone en extérieur.

Il serait intéressant de développer un autre système de levage autonome. En effet, comme expliqué plus haut, le système par électro-aimant est très pratique pour une phase de tests mais pas du tout pour une utilisation réelle du système.

Et finalement, il est important de rendre tout ce système plus "user-friendly". En effet, actuellement, pour lancer une mission de vol avec le drone, il faut lancer 4-5 scripts dans des terminaux peu accessibles à des personnes extérieures au projet. Il est important de développer une interface leur permettant de pouvoir utiliser le système le plus facilement possible.

Bibliographie

- [1] Logiciel qgroundcontrol. <http://qgroundcontrol.com/>.
- [2] Projet pixhawk. <http://pixhawk.org/>.
- [3] Protocole mavlink. <https://dev.px4.io/en/middleware/mavlink.html>.
- [4] Protocole mavros. <http://wiki.ros.org/mavros>.
- [5] Julien Amalaberque, Marie-Marie van der Beek, and Nicolas Vanvyve. Design of a positioning and a mission management system for builder drones. Master's thesis, UCLouvain, 2019.
- [6] Zakariya Bouhaddi, Pierre Latteur, and Clémence Flémal. Étude et développement d'un système de localisation et de guidage d'un drone basé sur la technologie UWB. Master's thesis, ECAM, 2018.
- [7] Georges Deneffe. Étude et développement d'éléments de maçonnerie drone-compatibles industrialisables. Master's thesis, UCLouvain, 2018.
- [8] Sébastien Goessens, Caitlin Mueller, and Pierre Latteur. Feasibility study for drone-based masonry construction of real-scale structures. *Automation in Construction*, 2018.
- [9] Pierre Latteur. Building our cities with drones... a utopia? | pierre latteur | tedxuclouvain. https://www.youtube.com/watch?v=tn9SnXz-c4g&ab_channel=TEDxTalks.
- [10] Nicolas Sorensen and Rémy Vermeiren. Development of a guidance and local automatic positionning system for buider drone with a theodolite. Master's thesis, UCLouvain, 2018.
- [11] PX4 Dev Team. Documentation. <https://docs.px4.io/en/>.
- [12] PX4 Dev Team. Flight modes. https://docs.px4.io/en/flight_modes/, 2019.

Annexe A

Offsets

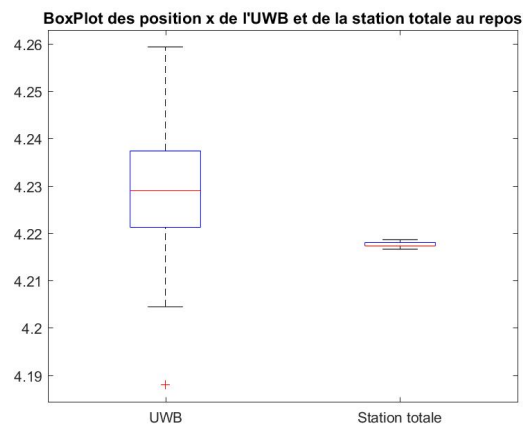


Figure A.1 – Offset en x entre l'UWB et la station totale.

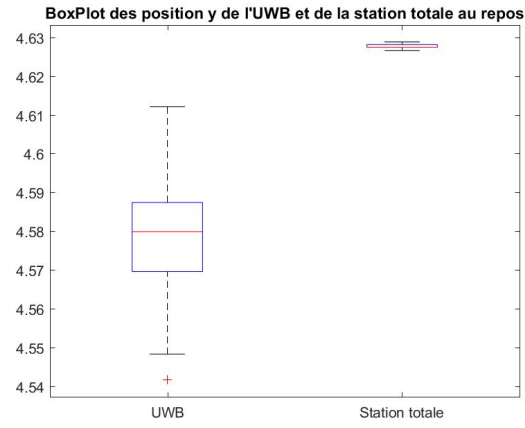


Figure A.2 – Offset en y entre l'UWB et la station totale.

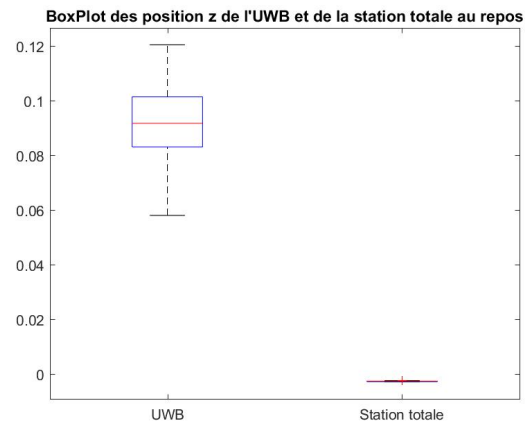
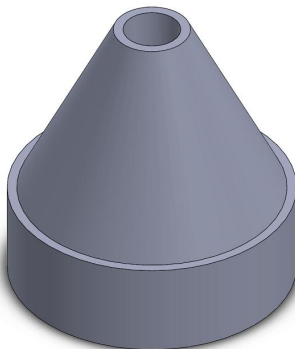
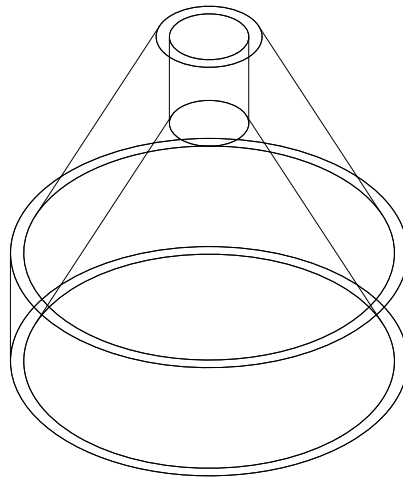
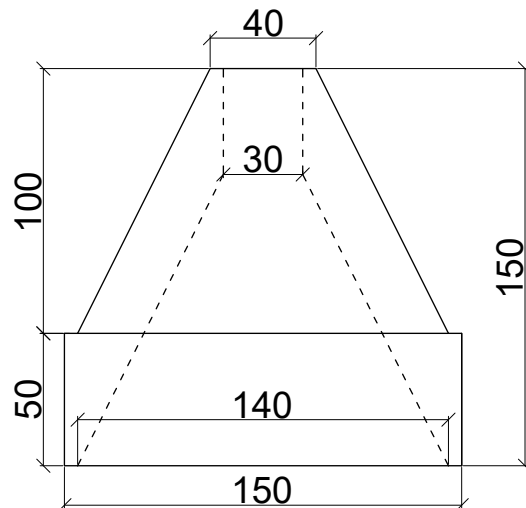


Figure A.3 – Offset en z entre l'UWB et la station totale.

Annexe B

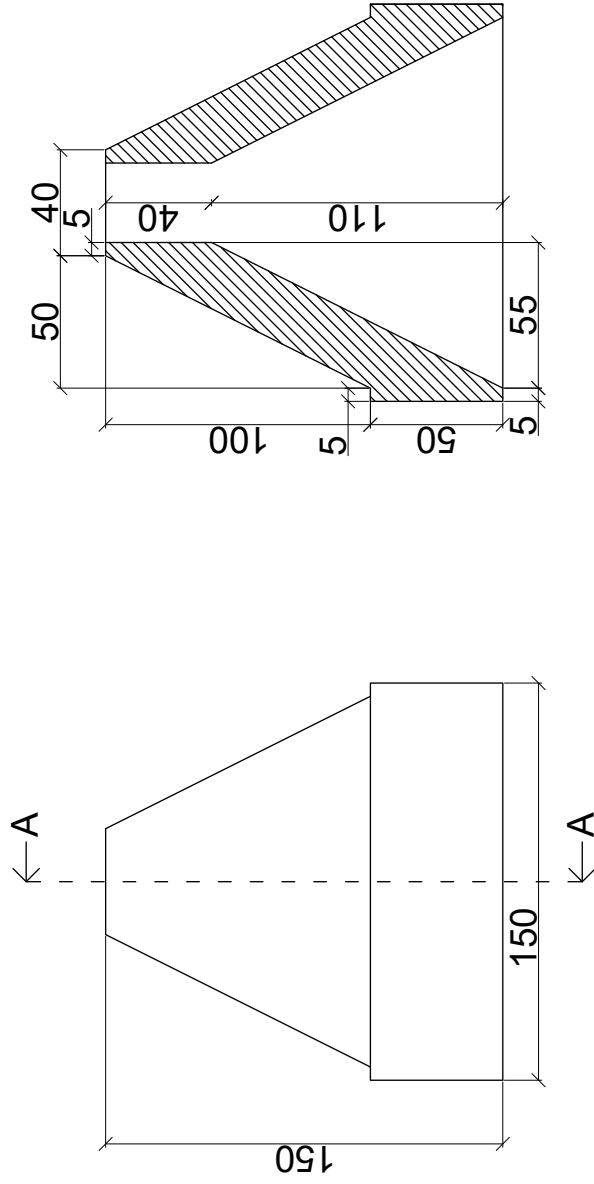
Fiches descriptives du cône de test et son guide

Date : 18.03.19		Auteur : Mathieu Monnart	
Cône 1.2			
Matériau : ABS		Aperçu :	
Côtes : [mm]			
Hauteur 150			
Largeur Base 150			
Largeur base 40			
Diamètre percement 30			
Epaisseur 5			
Hauteur empilement 50			
But : développer des blocs de type drone compatible à imprimer en 3D pour réaliser des tests de manières autonomes avec le petit drone.			
Caractéristiques :			
- Masse théorique pleine (SolidWorks) = 863 g (ABS $\rho = 1020 \text{ kg/m}^3$) - Masse réelle (Cura) = 365g - Volume : 845612 mm³			
Description :			
Ce bloc est logiquement drone compatible.			
La forme conique de ce bloc permet au drone de le prendre ou le poser sans connaître son orientation. Ceci va être utile car les premiers tests autonomes se feront sans système d'orientation du drone.			
Le bloc doit être léger car il est utilisé par le petit drone, d'où son impression 3D et le faible remplissage de matière. Ça permet aussi d'économiser de la matière pour imprimer.			
Ce bloc va être surmonter d'une petite plaque métallique (1.5 mm) collé qui permettra l'accroche avec l'électro-aimant.			
Propriété de l'impression 3D :			
- Tête d'impression : 0.8 mm - Hauteur des couches : 0.4 mm - Remplissage : Tri-hexagonale - Densité : 20%			

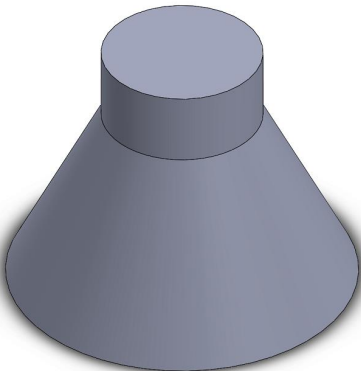


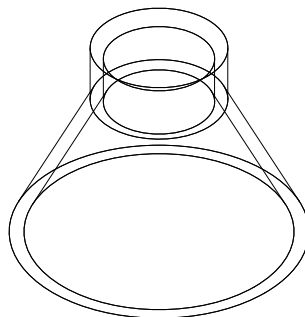
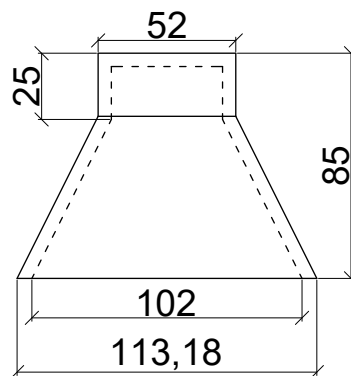
Ecole polytechnique de Louvain (UCL)		Bloc de construction à imprimer en 3D
Etudiant : Mathieu Monnart	Travail de fin d'étude	
Promoteur : Pierre Latteur	Année académique 2018-2019	
Assistant : Sébastien Goesens	14/03/2019	Cône 1.2
1:2	A4	

Coupe AA

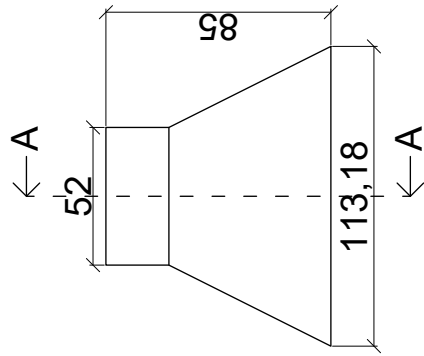


Ecole polytechnique de Louvain (UCL)		Bloc de construction à imprimer en 3D
Etudiant : Mathieu Monnart	Travail de fin d'étude	
Promoteur : Pierre Latteur	Année académique 2018-2019	
Assistant : Sébastien Goesens	14/03/2019	
1:2	A4	
		Cône 1.2

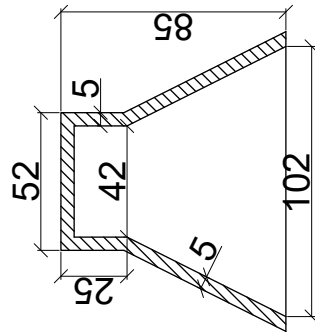
Date : 18.03.19		Auteur : Mathieu Monnart	
Guide cône 1.2			
Matériau : ABS		Aperçu :	
Côtes :	[mm]		
Hauteur	85		
Largeur Base	113,18		
Largeur base	52		
Epaisseur	5		
But : développer un guide pour le cône 1.2 qui aidera le drone pour la prise de ce dernier			
Caractéristiques : - Masse théorique pleine (SolidWorks) = 109,34 g (ABS $\rho = 1020 \text{ kg/m}^3$) - Masse réelle (Cura) = 106g - Volume : 1107200 mm³			
Description : Sa forme épouse celle du cône 1.2. Sa faible hauteur est dû au fait qu'il doit pouvoir être placé sous le petit drone sans être plus grand que les pieds pour que le drone puisse toujours atterrir. Le bloc doit être léger car il est utilisé par le petit drone, d'où son impression 3D et le faible remplissage de matière. Ça permet aussi d'économiser de la matière pour imprimer. Ce guide va contenir un électro-aimant dans sa partie supérieure, c'est lui qui permettra de soulever les blocs de type cône 1.2.			
Propriété de l'impression 3D : - Tête d'impression : 0.8 mm - Hauteur des couches : 0.4 mm - Remplissage : Tri-hexagonale - Densité : 20%			



Ecole polytechnique de Louvain (UCL)		Bloc de construction à imprimer en 3D
Etudiant : Mathieu Monnart	Travail de fin d'étude	
Promoteur : Pierre Latteur	Année académique 2018-2019	
Assistant : Sébastien Goesens	14/03/2019	Guide cône 1.2
1:2	A4	



Coupe AA



Ecole polytechnique de Louvain (UCL)		Bloc de construction à imprimer en 3D
Etudiant : Mathieu Monnart	Travail de fin d'étude	
Promoteur : Pierre Latteur	Année académique 2018-2019	
Assistant : Sébastien Goesens	14/03/2019	
1:2	A4	
		Guide cône 1.2

Annexe C

Modèles 3D

Modèles 3D réalisés dans le cadre du projet. Chaque modèle est créé pour être imprimé en 3D.

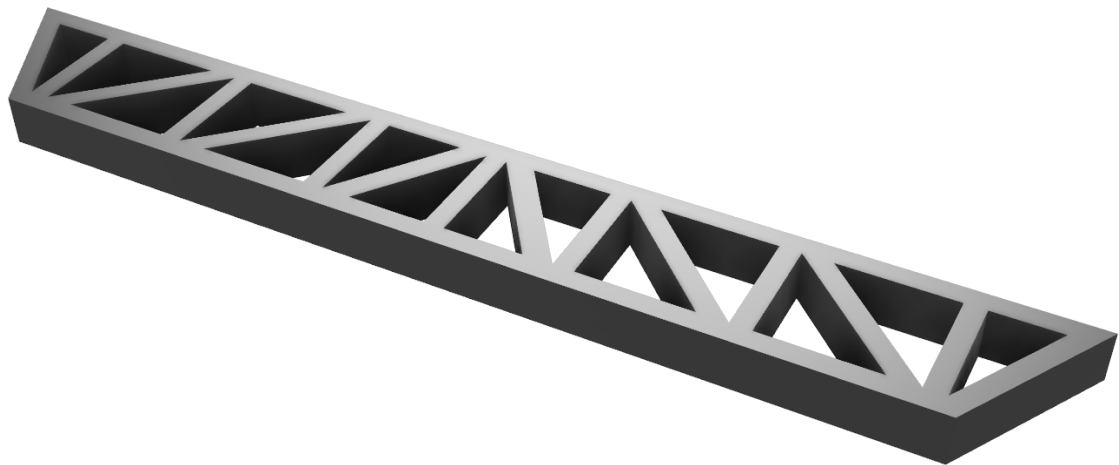


Figure C.1 – Modèle 3D du montant posé entre 2 bras du drone pour soutenir la box de la caméra.

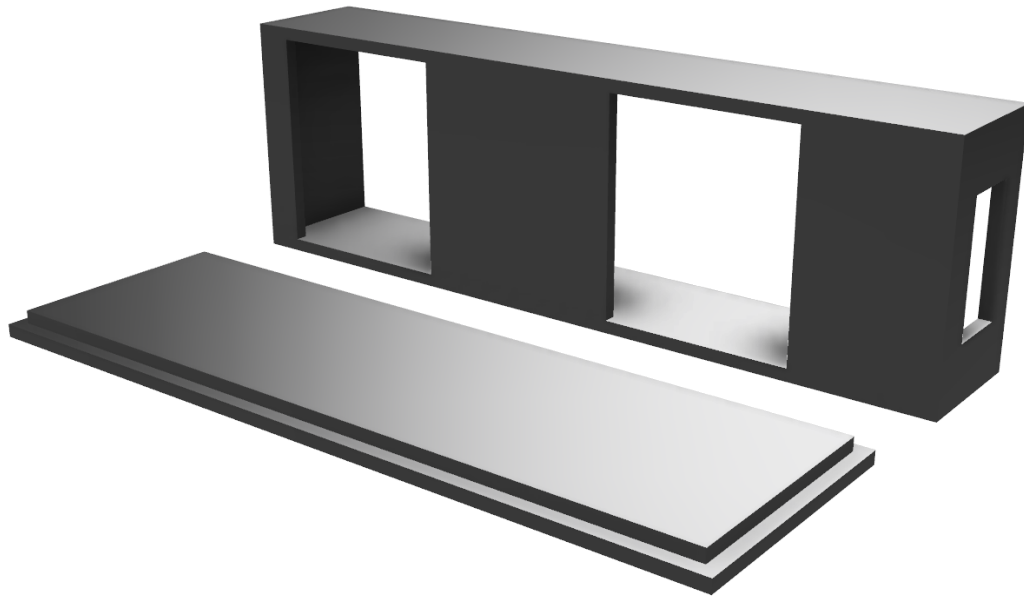


Figure C.2 – Modèle 3D de la box permettant de protéger la caméra *Intel T265*.

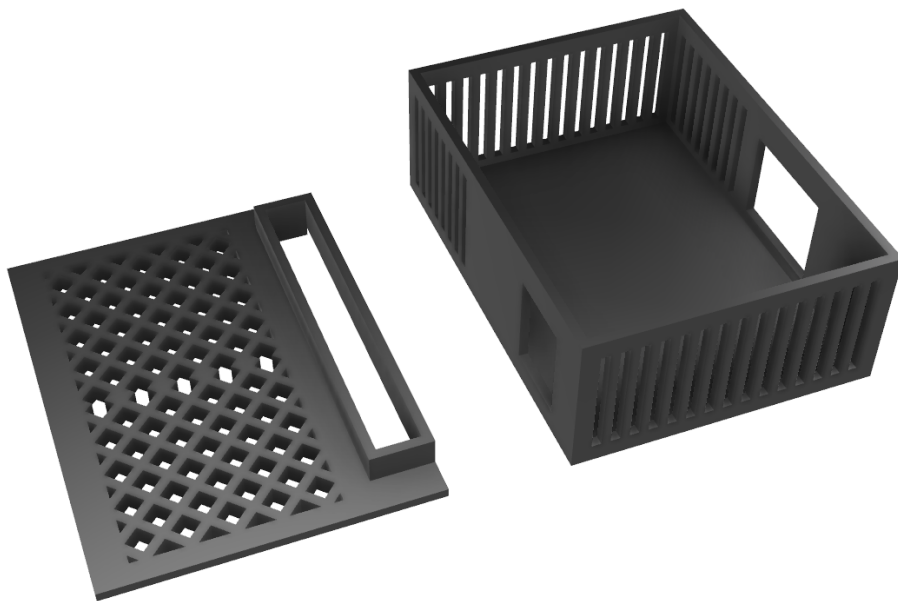


Figure C.3 – Modèle 3D de la box permettant de protéger l'Odroid XU4 et son shifter shield.

UNIVERSITÉ CATHOLIQUE DE LOUVAIN
École polytechnique de Louvain

Rue Archimède, 1 bte L6.11.01, 1348 Louvain-la-Neuve, Belgique | www.uclouvain.be/epl