



Option Robotique & Apprentissage

Etat de l'art - Projet Robotique

travail réalisé par

**Camille Parratt
Margo Biret
Lucas Gavériaux
Théodore Gigault**

Rapport de projet : Drone de reforestation massive

Année universitaire 2023 - 2024

Encadrement

M. Julien GENESTE
M. Vincent PADOIS

Porteur de projet
Responsable pédagogique

Résumé

L'état de l'art sur les technologies de localisation et de cartographie par drones, spécifiquement pour le projet "Instant Forest" mené par KARIBOO, se concentre sur l'utilisation de drones équipés d'un système de canon à air comprimé pour la reforestation. Le projet vise à développer un système de navigation semi-autonome pour les drones, permettant la dispersion précise de graines selon un quadrillage prédéfini.

Un prototype de drone avec un mécanisme de canon à graines intégré a déjà été élaboré, comprenant un module de projection de graines, un système de stockage, un système de guidage de drone et un algorithme de recherche de positions idéales pour le semis. Plusieurs technologies de localisation sont explorées, avec un accent particulier sur le GPS Precise Point Positioning (PPP), qui utilise des données de correction fournies par un réseau de stations de référence globales pour compenser les erreurs inhérentes aux signaux GPS.

La génération de trajectoire est un autre aspect crucial, visant à créer le chemin le plus court possible tout en s'adaptant à l'environnement du drone. Cette section explore diverses méthodes et algorithmes pour déterminer la trajectoire la plus optimale, en tenant compte des obstacles, de la régularité de la trajectoire et de la continuité des vitesses angulaires. Le lissage des chemins obtenus et la régularisation de la trajectoire sont essentiels pour garantir un vol sûr et précis. Enfin, le processus d'optimisation cherche à minimiser une fonction de coût tout en respectant des contraintes sur la trajectoire, telles que des limites sur les points de contrôle.

Table des matières

Table des figures	vii
1 Etat de l'art matériel et logiciel	3
1.1 Middleware de contrôle et communication	3
1.1.1 Choix du middleware	3
1.1.2 Choix de la version	4
1.2 Contrôleur automatique de vol	4
1.3 Station de contrôle au sol	5
2 Localisation du drone dans l'espace	7
2.1 Méthodes de localisation	7
2.1.1 GPS Precise Point Positioning	7
2.1.2 GPS avec INS	8
2.1.3 GPS avec RTK	9
2.1.4 Conclusion sur la méthode de localisation	10
2.2 Modélisation de l'espace de plantation	10
2.2.1 La photogrammétrie	10
2.2.2 LIDAR 3D	12
2.2.3 Caméra 3D	13
3 Génération de la trajectoire	15
3.1 Objectifs de la trajectoire	15
3.1.1 Minimiser les temps de vol	15
3.1.2 Optimiser la couverture de plantation	16
3.2 Adaptation de la trajectoire en temps-réel	19
3.2.1 Génération d'un chemin pour éviter les obstacles	19
3.2.2 Lissage du chemin et régularisation de la trajectoire	20
3.2.3 Optimisations de la Trajectoire	23
4 Synthèse	25
Conclusion & perspectives	27
Bibliographie	30

Table des figures

2.1	Schéma explicatif du PPP	8
2.2	Schéma explicatif du RTK	9
2.3	Modélisation 3D d'arbres par photogrammétrie	11
2.4	LIDAR 3D de Robosense	12
2.5	LIDAR 3D de LSLIDAR	12
2.6	Caméra stéréoscopique Intel RealSense d445	13
3.1	Trajectoire adoptée par le drone pour l'expérimentation en Indonésie (36graines/hectare) [1]	16
3.2	Visualisation des trajectoires obtenues pour FLPMW et RCPP sur 3 polygones Q_A , Q_B et Q_C [2]	18
3.3	Exemple de chemin (Γ_g) généré en utilisant l'algorithme A* [3]	20
3.4	Exemple de chemin (Γ_g) généré en utilisant l'algorithme RRT [3]	21
3.5	Exemple de chemin (Γ_g) généré lorsqu'il n'y a pas d'obstacle [3]	21
3.6	Exemple d'une courbe de Bézier générée par 4 points de contrôle [3].	22
3.7	Exemple d'une spline quadratique générée par 4 points de contrôle [3]	22
3.8	Correspondance entre les temps de calcul et de parcours des courbes [3]	23

Introduction

Le projet "Instant Forest", dirigé par KARIBOO, vise à mettre en œuvre l'utilisation de drones munis d'un système de canon à air comprimé pour mener une vaste opération de reforestation afin de contrer les incendies qui se sont déroulés dans le département de la Gironde ces dernières années. L'objectif principal du projet est de développer un système de navigation semi-autonome pour le drone, facilitant ainsi la dispersion des graines aux emplacements prédéfinis.

KARIBOO, fondée en septembre 2022 par deux passionnés de drones, Julien Geneste, ingénieur en Systèmes Électroniques Embarqués, et Paul Lebiez, ingénieur en Conception Logicielle Embarquée, a pour ambition principale de rendre l'utilisation des drones accessible à tous tout en assurant une fiabilité accrue. L'entreprise aspire à transformer le drone en un outil quotidien polyvalent au service de divers secteurs, tout en minimisant son impact sur l'environnement.

Les étudiants de l'année précédente ont déjà élaboré un prototype (Proof of Concept, POC) du drone, comprenant le mécanisme du canon, une structure de type F550 équipée d'un cube noir (Black Cube) et d'un système de télétransmission. Ils ont conçu et commencé à mettre en œuvre un mécanisme de canon à graines intégré à un drone. Les composants incluent un module de projection de graines, un système de stockage, un système de guidage de drone et un algorithme de recherche de positions idéales pour le semis. L'accent a été mis sur la précision de la projection tout en préservant la stabilité du drone. En l'état, le drone est capable de décoller, de voler dans une direction souhaitée en étant télécommandé, puis de se poser.

Ce rapport établit un état de l'art plus approfondi relatif aux enjeux de l'étape suivante du projet, à savoir l'implémentation d'une plantation complètement automatisée et autonome. Parmi ces enjeux, on compte notamment la cartographie de la zone en temps réel, la génération de trajectoire à partir de données d'entrées comme la zone de plantation et la densité de graines, etc. A cette fin, il est nécessaire de déterminer quels logiciels et matériel utiliser.

1 Etat de l'art matériel et logiciel

Cette section offre une vue d'ensemble des divers outils logiciels destinées au contrôle d'un drone et à la communication entre les différents éléments du système : la station de contrôle, les capteurs tels que les lidars et les caméras. Cette analyse de l'état de l'art vise à expliquer les différents choix effectués pour ce projet basés sur des critères pertinents pour celui-ci.

1.1 Middleware de contrôle et communication

Parmi les éléments à définir, le middleware est la liaison entre tous. Il est donc le premier à devoir être fixé.

1.1.1 Choix du middleware

La robotique, et plus spécifiquement les drones dans le contexte de ce projet, implique la coordination de nombreux éléments matériels par le biais d'une communication efficace. Pour cela, différents middlewares sont disponibles pour faciliter la compréhension de chaque composant. Seules les principales alternatives de ROS (Robot Operating System) et les caractéristiques relevées notamment dans les articles [4] et [5] sont présentées ici car il en existe de nombreuses et il est difficile d'être exhaustif.

	Environnement de simulation	Open-source	drones	Personnalisation / Configuration	Diversité et compatibilité des outils
ROS (Robot Operating System)	✓	✓	✓	✓	✓
MOOS (Mission Oriented Operating Suite)	×	✓	×	✓	×
Orocos (Open RObots COntrol Software)	×	✓	✓	✓	×
ASEBA (Soul for Education and Entertainment through Behavioral Animation)	✓	✓	×	×	×
OPRoS (Open Platform for Robotic Services)	✓	✓	✓	✓	×
YARP (Yet Another Reverse Proxy)	×	✓	✓	✓	×
Marie (Middleware for Robotic Applications Integration and Execution)	✓	✓	×	✓	×

TABLE 1.1 – Comparaison des différents middleware selon nos besoins

Il est important de noter sur le tableau 1.1 que les middleware ASEBA et MOOS ([6], [7]) n'ont pas été conçus pour des utilisations avec drones. En effet, ASEBA propose des outils à visée éducative tandis que MOOS est initialement conçu pour les sous-marins autonomes. En outre, malgré une conception modulaire similaire à celle des ROS, les autres middlewares présentés ici sont moins développés par la communauté et proposent donc moins d'outils ou de packages compatibles avec les différents composants des robots.

Enfin, pour des raisons de sécurité, les premiers tests effectués seront réalisés en simulateur. ROS dispose déjà de son propre simulateur et est donc plus facile à mettre en place, bien que les autres middlewares pourraient être intégrés dans un existant.

1.1.2 Choix de la version

ROS est un middleware qui facilite la communication entre les composants matériels et logiciels. Créé par Willow Garage en 2007, il est rapidement devenu un projet collaboratif et open source regroupant une communauté diversifiée de contributeurs. En raison de problèmes de sécurité ou encore de gestion en temps réel, l'arrêt du support de la première version de ROS est prévu en 2025. Il a donc été nécessaire de construire une nouvelle version, ROS2, palliant ces problèmes. Les concepteurs ont alors entrepris de reconstruire le middleware depuis le début pour garantir des améliorations significatives sans perturber les robots utilisant déjà ROS, comme mentionné dans cet article [8]. Pour ces raisons, ROS2 va au-delà d'être une simple deuxième version de ROS et nécessite un passage plus complexe de l'un à l'autre.

Tout en s'appuyant sur les mêmes concepts (système modulaire avec des noeuds pouvant communiquer sous le système de sujet et d'abonnement), ROS2 offre la possibilité de connecter plusieurs robots, communiquer en temps réel ou encore s'assurer de l'instanciation de tous les composants avant de démarrer. En outre, pour améliorer la sécurité et le système de communication, ROS2 s'appuie sur le protocole DDS (Data Distribution Service) qui est un moyen de communication standardisé, fiable et en temps réel. Pour finir, ROS2 repose maintenant sur des versions supérieures de Python et C++.

Ce projet se réalisant en 2023, ROS2 a semblé un choix pertinent de version pour exécuter ce projet. En effet, l'équipe diplômée fin 2024 et la fin du support de ROS1 en 2025, laissent supposer que les entreprises auront déjà bien entamé, voire terminé leur transition vers ROS2. Afin de proposer des connaissances et compétences à jour et pertinentes, la version 2 de ROS a été choisie pour ce projet.

1.2 Contrôleur automatique de vol

Un drone étant composé de plusieurs ailes et donc plusieurs moteurs, il est nécessaire d'assurer une cohérence entre la rotation de ces derniers afin de maintenir la stabilité du drone et assurer un vol optimal. De nombreux logiciels de vol embarqué permettent la gestion autonome des moteurs, facilitant ainsi le contrôle du drone. Voici les principaux existants ainsi que leurs avantages et inconvénients :

	Stabilité	Personnalisation / Configuration	Compatible ROS	Compatible avec le cube	Open-source
PX4	✓✓	✓✓✓	✓	✓	✓
ArduPilot	✓	✓✓	✓	✓	✓
DJI SDK	✓✓	×	✓	×	×
Paparazzi	✓	✓	×	✓	✓
OpenPilot	✓	✓	×	×	✓

TABLE 1.2 – Comparaison des différents contrôleurs automatiques de vol

La compatibilité du cube contrôleur de vol est un critère majeur, étant donné que cet élément matériel est déjà en possession du projet et que l'objectif n'est pas de discuter de sa pertinence.

DJI SDK est un ensemble de bibliothèques permettant de piloter et contrôler des drones DJI. Bien qu'offrant une bonne stabilité de vol et des modules ROS, il est limité aux drones DJI, ce qui ne convient pas au projet actuel.

Paparazzi et OpenPilot ne sont pas nativement compatibles avec ROS, posant des défis d'intégration pour ce projet. Malgré la possibilité de développer des ponts entre ces plateformes de pilotage et ROS, il est préférable d'utiliser des outils directement fonctionnels avec ROS. Cela permettra de se concentrer sur la suite du projet.

Enfin, entre PX4 et ArduPilot, le choix est un peu plus nuancé [?]. Bien que les deux répondent aux besoins du projet, PX4 propose une stabilité de vol supérieure, ce qui est primordial pour la précision nécessaire au largage des graines. De plus, PX4 possède des fonctionnalités de sécurité supplémentaires qu'il faudrait intégrer soi-même avec ArduPilot et la personnalisation de certaines configurations est plus simple à réaliser sur PX4.

Ainsi, le choix de PX4 semble pertinent pour assurer le contrôle autonome du vol.

1.3 Station de contrôle au sol

Dans l'optique de contrôler le drone mais également visualiser en temps réel ses paramètres, il est indispensable d'avoir une station au sol. Plusieurs options sont disponibles, mais seules celles compatibles avec PX4 seront mentionnées ci-dessous.

Drone Harmony, UgCS(Universal Ground Control Software) et QGroundControl sont trois solutions disponibles avec l'utilisation de PX4. Malheureusement, seul QGroundControl peut être utilisé avec ROS. Pour cette raison, mais aussi parce qu'il se configure très bien avec PX4, même en simulation, le choix pour ce projet se porte sur QGroundControl.

2 Localisation du drone dans l'espace

2.1 Méthodes de localisation

Diverses technologies existent pour localiser un drone dans l'espace. La plus connue est le Global Positioning System (GPS). Toutefois, cette méthode n'a une précision qu'à quelques mètres et bien que suffisant pour naviguer sur de longues distances ou accomplir des missions ne nécessitant pas une précision élevée, pour une application de plantation de graines selon un quadrillage prédéfini, il faut coupler cette méthode avec d'autres pour augmenter la précision.

2.1.1 GPS Precise Point Positioning

La méthode GPS Precise Point Positioning (PPP) apporte une précision accrue dans diverses applications, notamment dans le pilotage et la cartographie par drones. Le GPS PPP améliore la précision du positionnement en rectifiant les signaux GPS standard. Cette méthode utilise des données de correction fournies par un réseau de stations de référence globales pour compenser les erreurs inhérentes aux signaux GPS, telles que les perturbations atmosphériques, les erreurs de chronométrage des satellites et les inexactitudes orbitales.

Les signaux GPS sont susceptibles d'être déformés en traversant l'atmosphère terrestre. Le PPP utilise des modèles atmosphériques pour corriger ces distorsions, améliorant ainsi la précision des signaux. De plus, les horloges des satellites GPS peuvent présenter de légères déviations, et leurs orbites peuvent légèrement varier. Les stations de référence GPS PPP, comme illustré sur la figure 2.1 surveillent ces variations et fournissent des corrections pour ajuster les données de positionnement.

La précision élevée est la force de cette méthode. Contrairement à la localisation par GPS pure fournissant une localisation à quelques mètres, avec du PPP il est possible de réduire le taux d'erreur à quelques centimètres.[9]

À la différence des méthodes de GPS différentiel comme le RTK, qui sera évoqué par la suite qui nécessitent une station de base et un récepteur mobile, le PPP fonctionne avec un seul récepteur GPS, rendant cette technologie plus flexible et plus facile à déployer particulièrement dans des régions isolées ou difficiles d'accès.

Cette méthode est notamment utilisée pour cartographier de larges zones avec des drones à voile fixe pour une autonomie plus conséquente que les drones à rotors plus populaires.

De plus, Le PPP peut être intégré avec d'autres systèmes de navigation, comme les systèmes de navigation inertielle (INS), également évoquée par la suite, pour une précision améliorée, particulièrement dans les environnements où les signaux GPS sont faibles ou intermittents.

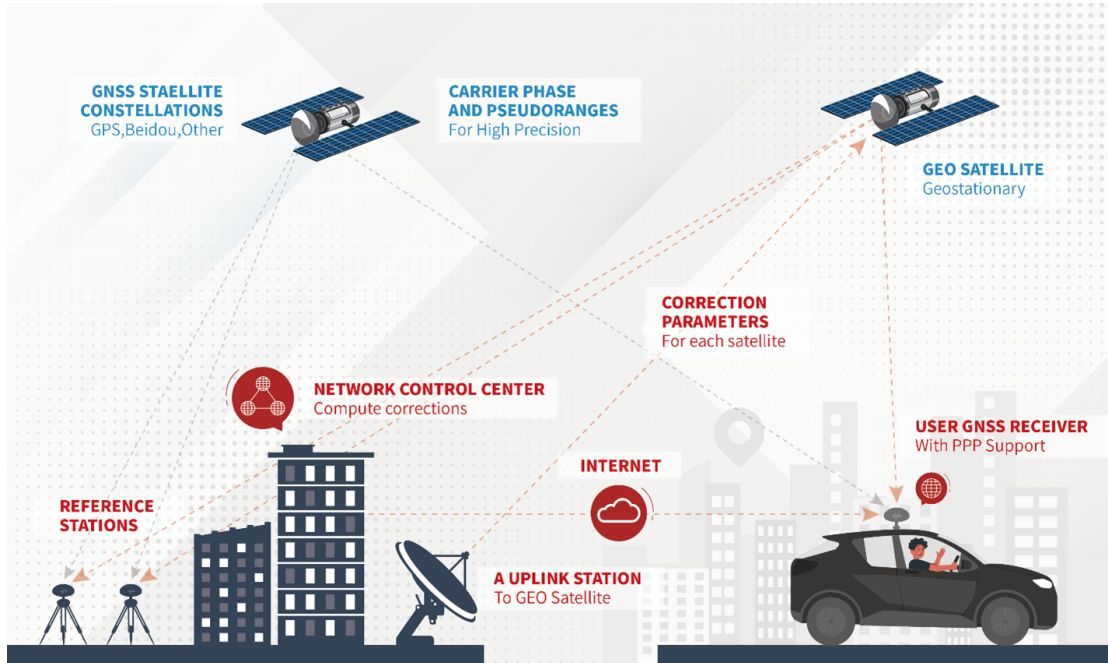


FIGURE 2.1 – L'utilisateur possède un receveur GNSS et reçoit les données satellites. Ces dernières sont aussi comparées à des stations de références et des corrections sont calculées avec des algorithmes dans un centre de calcul. La position avec une précision augmentée est ensuite transmise au receveur.

2.1.2 GPS avec INS

La localisation combinée GPS-INS intègre les données de positionnement et de vitesse fournies par le GPS avec les informations sur l'orientation et l'accélération mesurées par l'INS. L'INS, qui comprend généralement un accéléromètre et un gyroscope, fournit une estimation continue du mouvement, tandis que le GPS corrige périodiquement les erreurs accumulées par l'INS.

L'avantage principal de l'INS est qu'il peut continuer à fournir des données de navigation lorsque le signal GPS est faible ou indisponible (comme dans les canyons urbains ou les forêts denses). Il est toutefois sujet à de la dérive sur de longues périodes sans corrections GPS.

La combinaison des données de l'INS et du GPS se font traditionnellement avec des filtres de Kalman. Toutefois, cette méthode peut se révéler être quelque peu lacunaire lorsque les conditions du milieu dans lequel se déplace le drone sont particulièrement bruitées. Cet article de recherche publié dans le CSAA (Chinese Society of Aeronautics and Astronautics) en 2018 [10] propose un nouvel algorithme pour traiter les données avec un filtre non linéaire adaptatif. Cela permet de diminuer le taux d'erreur pour une meilleure précision. Cette méthode trouve sa force notamment en environnement dense et/ou urbain.

2.1.3 GPS avec RTK

Le RTK est une technique avancée qui utilise des mesures de la phase du signal porteuse des satellites GPS pour améliorer la précision de la position à quelques centimètres près. Elle fonctionne en corrigeant les signaux GPS en temps réel, en utilisant une station de base fixe qui connaît sa position exacte. L'intégration du RTK dans les drones améliore considérablement la précision de leur localisation, essentielle pour des applications telles que la cartographie de précision, l'agriculture de précision, et les inspections structurelles ce qui s'inscrit dans le cadre de notre application pour le projet Kariboo. Le RTK a l'avantage de fournir une correction en temps réel à l'aide d'une station au sol et unité de calcul. La précision obtenue est très élevée. Une étude italienne a évalué 3 types de localisation par RTK à faible coût et a conclu que cette méthode présente une marge d'erreur de moins de 5 cm. [11]

La figure 2.2 illustre le fonctionnement de la localisation par RTK. Par données satellites comme le GPS traditionnel et en couplant avec une station au sol dont la position est connue, on peut localiser le drone de manière très précise en temps réel.

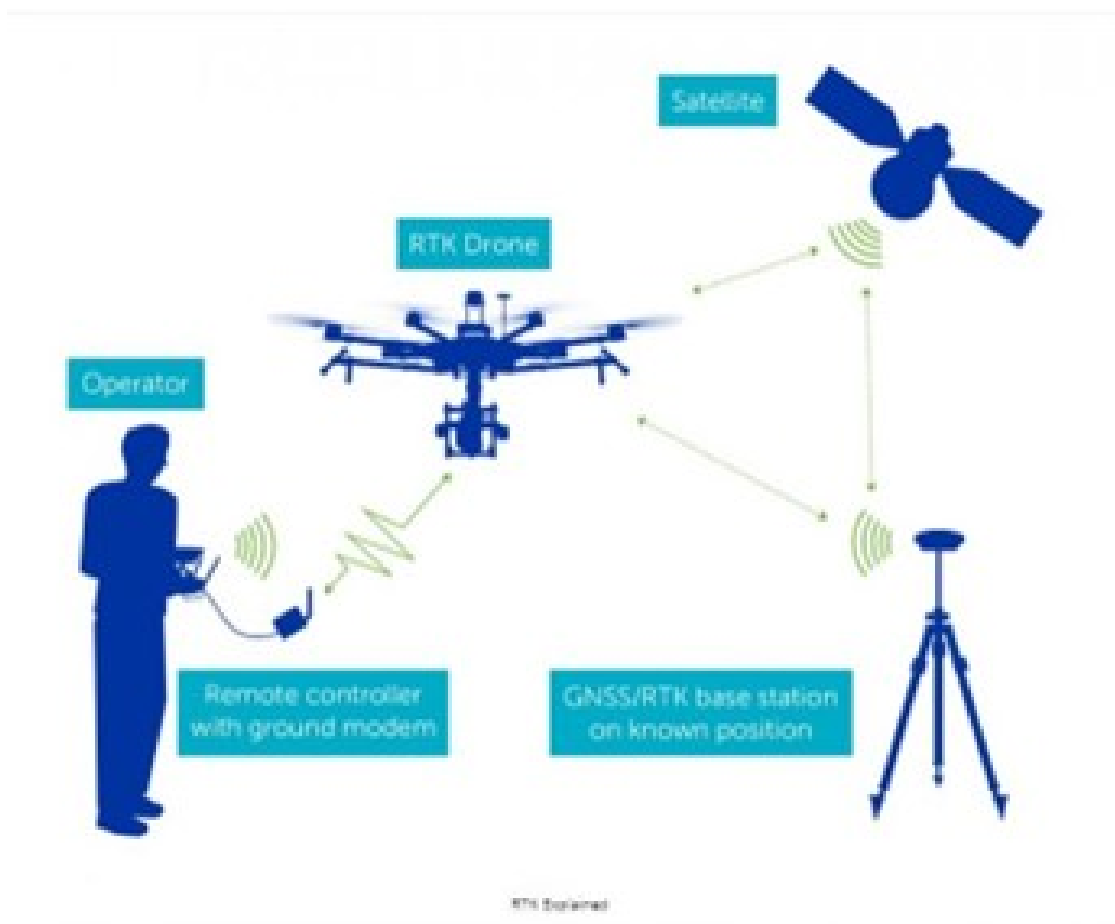


FIGURE 2.2 – Schéma illustrant l'utilisation de la méthode de localisation par RTK[12]

2.1.4 Conclusion sur la méthode de localisation

Dans notre cas, la localisation par RTK est la plus pertinente pour plusieurs raisons. La méthode par PPP est plus avantage pour cartographier des zones extrêmement larges notamment avec des drones à voilures fixes. La méthode avec INS trouve surtout sa force dans des zones encombrées, avec un signal GPS faible et est de plus difficile mathématiquement parlant à mettre en place. La reforestation se fait dans des zones généralement dégagée donc cette méthode n'est pas pertinente ici. En revanche, la méthode par RTK permet d'obtenir une grande précision nécessaire pour planter des graines au bon endroit et suivre les trajectoires générées.

2.2 Modélisation de l'espace de plantation

Il est important de bien connaître le terrain au préalable afin d'optimiser la plantation des graines pour une reforestation efficace, notamment la présence de formations rocheuses, des terrains particulièrement secs, eaux stagnantes... Dans cet objectif, il est nécessaire que le drone comporte un moyen d'observer le terrain afin le drone puisse prendre une décision de planter une graine ou non en fonction de la nature du terrain.

2.2.1 La photogrammétrie

L'article "Drone-based photogrammetry for the construction of high-resolution models of individual trees" illustre l'efficacité de la photogrammétrie par drone dans la modélisation tridimensionnelle d'arbres individuels. Cette méthode utilise des drones équipés de caméras pour capturer des images détaillées d'arbres, permettant de créer des modèles 3D précis qui révèlent la structure complexe des branches et de la canopée. L'étude met en évidence l'importance des conditions environnementales, notamment la lumière et le fond, sur la qualité des modèles. Par temps couvert, les modèles obtenus étaient plus précis que ceux capturés par beau temps. Ces modèles 3D que l'on peut voir sur la figure 2.3 offrent des perspectives pour des études approfondies en physiologie et écologie des arbres, soulignant le potentiel de la photogrammétrie par drone dans la recherche scientifique et la gestion environnementale.[13]

Il pourrait être envisagé d'effectuer un procédé similaire pour étudier le terrain avant plantation des graines. Toutefois, le traitement des données est long et coûteux : un ordinateur avec des capacités de traitement d'images puissantes est nécessaire. L'ordinateur utilisé dans l'article avait 4 cartes graphiques et 128Go de RAM pour une durée de traitement de 24 heures. De plus, dû à la difficulté d'effectuer ce traitement, ce dernier ne peut pas se faire de manière embarquée. Dans notre application, il est plutôt souhaitable de pouvoir évaluer en temps réel si le terrain sous le drone est propice à la plantation d'une graine.

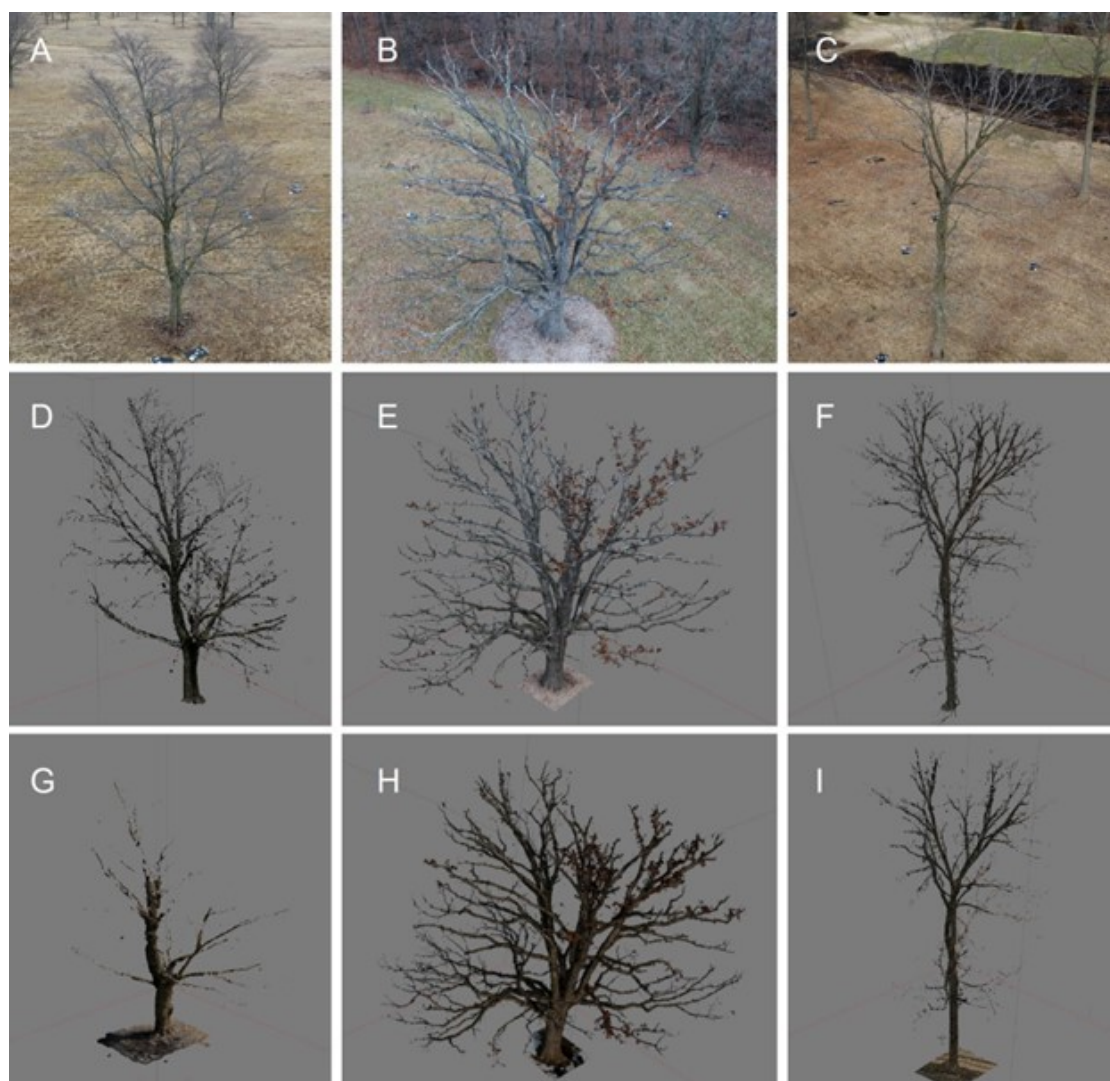


FIGURE 2.3 – Résultat d’une étude sur la reconstruction 3D d’arbres par photogrammétrie [13]

2.2.2 LIDAR 3D

Dans le domaine de la cartographie de terrain, la technologie LiDAR (Light Detection and Ranging) 3D est devenue un outil incontournable, offrant une précision et une résolution spatiale remarquables. Le LiDAR 3D utilise des impulsions laser pour mesurer la distance entre le capteur et la surface du terrain ou des objets, permettant ainsi de générer des modèles de points puis de créer des modèles numériques de terrain (MNT) et de surface (MNS) très détaillés. Cette technologie est cependant extrêmement coûteuse. Le LIDAR 3D 2.4 qui serait potentiellement à disposition ne serait toutefois pas pertinente car celui-ci possède un angle mort en-dessous dû à son angle d'ouverture relativement faible de 30 degrés. [14]

Il existe toutefois des nouveaux modèles de LIDAR 3D qui corrigent ce défaut, notamment le modèle de l'entreprise LSLIDAR représenté sur la figure 2.5.



FIGURE 2.4 – LIDAR3D Robosense



FIGURE 2.5 – LIDAR3D de LSLIDAR à grand angle d'ouverture

2.2.3 Caméra 3D

Une caméra 3D utilise la technologie de vision stéréoscopique pour mesurer la profondeur. La vision stéréoscopique, également connue sous le nom de stéréoscopie, est une technique de vision ou de capture d'images qui crée ou recrée l'illusion de la profondeur dans une image en imitant la vision binoculaire humaine. Fondamentalement, elle repose sur le principe que nos deux yeux voient le monde sous des angles légèrement différents, et notre cerveau fusionne ces deux images pour former une perception tridimensionnelle de notre environnement. En technologie, la vision stéréoscopique est souvent mise en œuvre en utilisant deux caméras, placées à une distance l'une de l'autre similaire à celle des yeux humains, pour capturer deux images distinctes du même objet sous des angles légèrement différents. Ces images sont ensuite traitées à l'aide d'un module infrarouge qui permet une mesure des distances avec des obstacles dans l'image pour calculer la profondeur des objets dans la scène et reproduire en 3D la vision de la caméra.

Cette article de recherche montre la possibilité d'utiliser des caméras à vision stéréoscopique pour localiser un drone dans un espace intérieur à l'aide de différents algorithmes. [15]

Certaines caméras, comme la caméra Intel RealSense D445 2.6 qui nous est proposé d'utiliser nous permet de passer outre les difficultés mathématiques car Intel propose déjà des moyens d'intégration simplifiés à l'aide d'un SDK et des packages existent déjà pour intégrer cet outil à une architecture ROS.



FIGURE 2.6 – Caméra stéréoscopique Intel RealSense d445[16]

3 Génération de la trajectoire

La génération de trajectoire pour les drones constitue une composante cruciale dans la majorité des domaines d'utilisation des drones. Dans le contexte du projet mené par Instant Forest et Kariboo, la trajectoire doit être la plus courte possible tout en s'adaptant à l'environnement du drone. Cette section explore les méthodes et algorithmes envisagés, offrant une base conceptuelle robuste pour les développements du projet de reforestation.

3.1 Objectifs de la trajectoire

La trajectoire est déterminée en fonction de deux paramètres principaux qui constituent l'objectif de mission du drone : la délimitation de la zone et la densité de graine à planter. Cette section s'intéresse à d'autres paramètres potentiels pouvant modifier la trajectoire du drone ainsi qu'à leur impact sur son efficacité dans la réalisation de sa mission de reforestation. En d'autres termes, à partir d'une délimitation de zone et d'une densité donnée, comment déterminer la trajectoire la plus optimale afin de recouvrir toute la surface ?

3.1.1 Minimiser les temps de vol

Afin de s'assurer d'une efficacité optimale à tous les niveaux, l'une des variables mesurables et modifiables est la durée totale de vol nécessaire pour réaliser la mission. Selon des expérimentations réalisées, par exemple, en Indonésie [1], il apparaît que cette donnée est minimisable en jouant sur d'autres paramètres clés.

En effet, une donnée intéressante à considérer est le rapport entre la densité de graine et le temps de vol pour une surface donnée. Autrement dit, le nombre de graines plantées par seconde. Lors des tests réalisés en Indonésie, le drone a adopté une méthode de plantation en allers-retours, schématisée en figure 3.1, afin de planter les graines sur une zone d'un hectare suivant une forme de grille, ce qui peut sembler, en première approche, la meilleure option pour une zone rectangulaire (voir discussion sur le sujet en section 3.1.2).

D'après les résultats de l'expérience, présentés en table 3.1, on constate l'importance de la densité vis-à-vis de l'efficacité temporelle. Pour une densité de 36 graines par hectare, on constate une vitesse de plantation de 0.26 graines par seconde, tandis que pour une densité de 440 graines par hectare, une vitesse de 1 graine par seconde.

Ainsi, pour une même superficie et une même vitesse de déplacement du drone (ici 18 km/h), augmenter la densité augmente également la vitesse de plantation, et minimise donc le temps de vol. Il peut donc s'avérer important de maximiser la densité en minimisant, par exemple, la surface de plantation.

Surface à couvrir (hectares)	Densité (graines / hectare)	Temps de vol (s)	Distance (m)
1	36	140	700
	121	220	1100
	440	440	2200

TABLE 3.1 – Résultats expérimentaux d'un plantage de graine par drone sur 1 hectare en Indonésie [1]

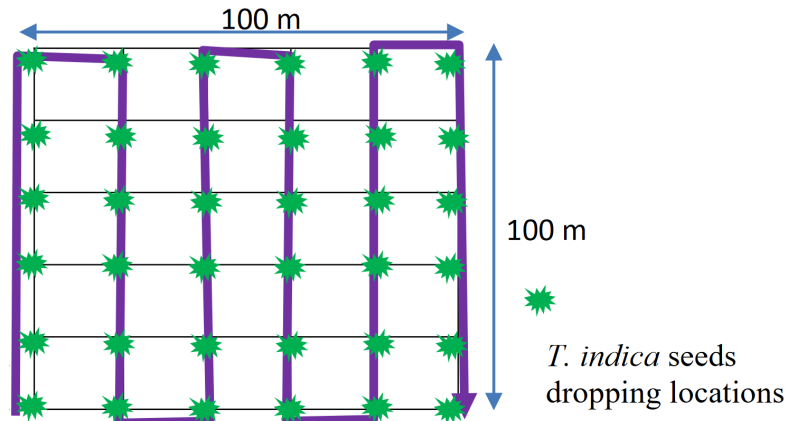


FIGURE 3.1 – Trajectoire adoptée par le drone pour l'expérimentation en Indonésie (36graines/hectare) [1]

En bref, il s'agit de déterminer, en fonction du cas d'usage, quels paramètres sont fixés, et lesquels sont plus flexibles. Par exemple, si l'on veut planter un nombre prédéterminé de graines, réduire la zone sur laquelle elles seront plantées réduira le temps de vol total. À l'inverse, sur une surface fixée, il peut sembler pertinent de maximiser (sans en planter trop, cf section 3.1.2) le nombre de graines pour rentabiliser le temps de vol.

3.1.2 Optimiser la couverture de plantation

Nous avons évoqué en section 3.1.1 une première approche utilisée afin de couvrir une zone rectangulaire. Cette méthode, représentée en figure 3.1, consiste à discrétiser l'espace sous forme de grille et de planter les graines suivant ce schéma avec un espacement équivalent entre chacune des graines. Le drone parcourt la grille en effectuant des allers-retours afin de minimiser la distance totale parcourue.

Cette sous-section questionne le caractère optimal de cette méthode, et explore les autres solutions possibles, notamment pour la couverture de zones non rectangulaires.

La méthode présentée en figure 3.1 est plus communément appelée BFP (*back and forth path*). D'autres chemins de couvertures ont été étudiés, comme la spirale, l'étoile, ou encore le zig-zag (ou dents de scie), mais le BFP est souvent préféré car il est plus facilement modélisable et parce qu'il garantit plus facilement une couverture complète de la surface à parcourir [2].

Il s'agit alors de générer un "chemin de couverture" (*coverage path planning*) suivant la méthode BFP. Il existe des algorithmes permettant de générer de tels chemins, l'un des plus connus étant FLPMW (*flight lines perpendicular to minimum width*) [2]. Il consiste à parcourir le polygone étudié en faisant des allers retours (comme pour la figure 3.1) dans la direction perpendiculaire à la largeur minimale du polygone. Prenons un exemple simple : si le polygone en question est un rectangle, il s'agit de le parcourir dans sa longueur, ce qui permet de minimiser le nombre d'allers-retours, et donc d'optimiser la trajectoire.

Un algorithme alternatif existe, RCPP (*rotating callipers path planning*), présenté en détail en référence [2]. Cet algorithme semble plus efficace que le classique FLPMW, et est donc à privilégier dans une optique d'optimisation de la trajectoire de couverture. La comparaison de ces deux algorithmes est mise en évidence à travers les résultats expérimentaux de la table 3.2 et la figure 3.2.

Poly.	Temps de calcul (s)		Temps de vol estimé (s)		Temps de vol (s)	
	FLPMW	RCPP	FLPMW	RCPP	FLPMW	RCPP
Q_A	0.0027	0.0066	189.3	186.2	174	171
Q_B	0.0021	0.0064	125.9	119.2	108	102
Q_C	0.0023	0.0077	167.1	164.5	142	136

TABLE 3.2 – Comparaison des performances des algorithmes FLPMW et RCPP sur 3 polygones Q_A , Q_B et Q_C [2]

Notons également que pour les surfaces complexes, il semble pertinent de découper celles-ci en des sous-régions plus simples et d'appliquer à chacune un paterne de couverture approprié. De tels découpage sont souvent utilisés dans le milieu de l'agriculture pour optimiser le parcours des champs, aux surfaces très variables [17].

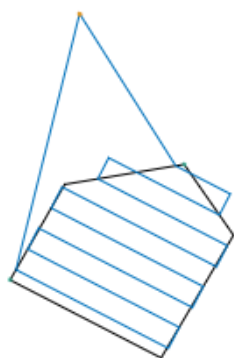
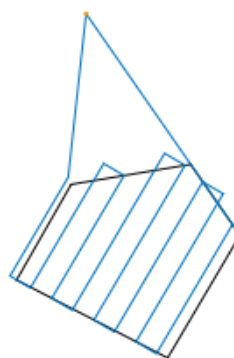
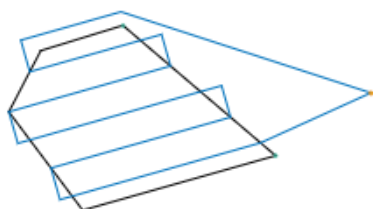
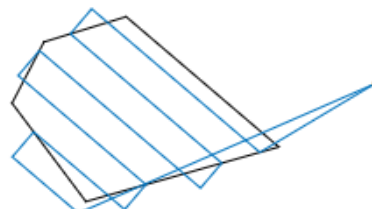
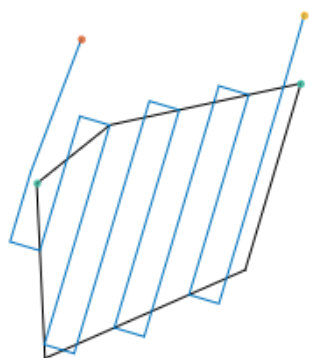
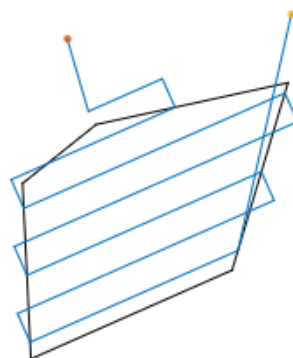
(a) RCPP's path for Q_A .(b) FLPMW's path for Q_A .(c) RCPP's path for Q_B .(d) FLPMW's path for Q_B .(e) RCPP's path for Q_C .(f) FLPMW's path for Q_C .

FIGURE 3.2 – Visualisation des trajectoires obtenues pour FLPMW et RCPP sur 3 polygones Q_A , Q_B et Q_C [2]

3.2 Adaptation de la trajectoire en temps-réel

A partir des chemins générés à la section précédente, on va pouvoir calculer une trajectoire, c'est-à-dire un chemin avec des contraintes et composantes temporelles. Notamment, on veut pouvoir passer au contrôleur de drone une position, mais aussi une vitesse et une accélération.

3.2.1 Génération d'un chemin pour éviter les obstacles

La première étape consiste à discrétiser le chemin précédemment obtenu pour pouvoir le modifier. Ainsi, on va vouloir décomposer les chemins en points de passage, aussi appelés points de contrôle. Les points de contrôle sont des positions spécifiques définies dans l'espace qui influencent la forme de la trajectoire générée. Ces points seront utilisés pour décrire une courbe continue de manière paramétrique, permettant ainsi de spécifier la trajectoire désirée du drone.

Pour générer ces points à partir d'un chemin donné, on va pouvoir sélectionner des points de la courbe de façon régulière en choisissant un pas. Le choix de ce pas est important, puisqu'un pas trop petit créera trop de points et obligera donc le drone à recalculer sa trajectoire trop souvent. À l'inverse, un pas trop grand retirera des informations cruciales du chemin, comme par exemple un virage.

Une fois ces points générés, on va pouvoir calculer la trajectoire du drone en temps-réel entre deux points. La trajectoire optimale est donc construite localement, dans la sphère de perception du drone, en évitant les obstacles et en respectant les contraintes cinématiques et dynamiques de vol.

Le planificateur de trajectoire décrit dans la thèse [3] utilise des informations omnidirectionnelles provenant de caméras 3D. Cela permet de construire une carte des obstacles dans la sphère de perception du drone. On se base sur l'idée que la position et la vitesse du drone sont disponibles en permanence grâce à l'odométrie.

Ainsi, on va pouvoir calculer le chemin local optimale pour éviter les obstacles qui font face au drone. Pour cela, on va pouvoir utiliser un parmi plusieurs algorithmes de recherche de chemin optimal dans un graphe, comme par exemple A* ou Rapidly-exploring Random Tree (RRT).

L'algorithme A* est utilisé pour trouver le chemin le plus court entre deux points dans un graphe. Il utilise une combinaison de coûts réels passés $g(n)$ et d'estimations heuristiques du coût futur $h(n)$ pour évaluer les nœuds. L'algorithme explore les nœuds en fonction de leur coût total $f(n) = g(n) + h(n)$ et met à jour les coûts associés lorsqu'un chemin plus court est découvert. Il maintient deux ensembles de nœuds, *OPEN* et *CLOSED*, pour suivre les nœuds à explorer et ceux qui ont été explorés. L'algorithme se termine lorsque le nœud d'arrivée est atteint ou lorsque l'ensemble *OPEN* est vide. Un exemple permet d'illustrer cela en figure 3.3.

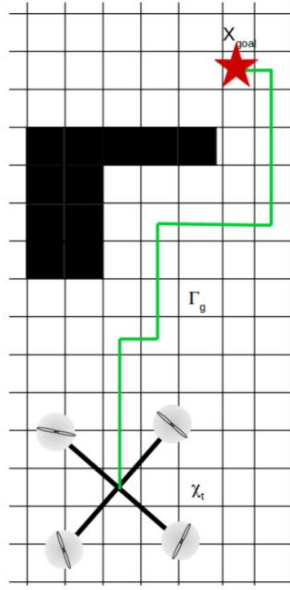


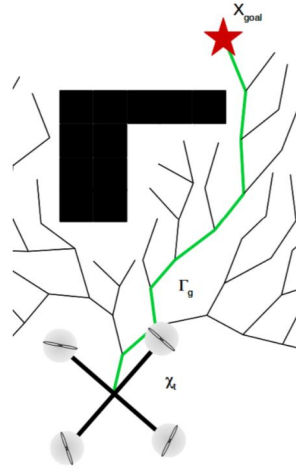
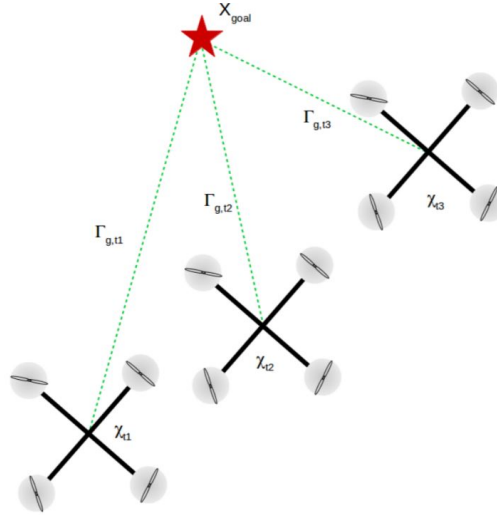
FIGURE 3.3 – Exemple de chemin (Γ_g) généré en utilisant l'algorithme A* [3]

L'algorithme RRT (Rapidly-exploring Random Tree) est utilisé pour la planification de trajectoires dans des espaces complexes. Il démarre avec un arbre contenant le nœud de départ et itère en générant aléatoirement des points dans l'espace, connectant ces points au nœud existant le plus proche tout en respectant les contraintes de l'environnement. L'algorithme se termine lorsqu'un nœud dans la proximité de la cible est atteint ou après un nombre prédéfini d'itérations. Le chemin optimal peut ensuite être reconstruit en remontant l'arbre. RRT excelle dans l'exploration rapide d'espaces inconnus et complexes, en faisant une méthode efficace pour la planification de trajectoires dans des environnements réels. Un exemple permet d'illustrer cela en figure 3.4.

Enfin, si aucune carte n'est disponible, le chemin optimal est alors une ligne droite vers le prochain point, comme montré en figure 3.5.

3.2.2 Lissage du chemin et régularisation de la trajectoire

La clé essentielle pour la réussite dans le calcul de trajectoires pour un drone réside dans l'obtention d'une trajectoire régulière et une continuité des vitesses. Une trajectoire régulière garantit une progression fluide et prévisible du drone dans l'espace, minimisant les variations brusques qui pourraient compromettre la stabilité du vol. La continuité des vitesses angulaires est tout aussi cruciale, car elle permet d'éviter des changements abrupts dans la direction du drone, contribuant ainsi à une navigation plus harmonieuse. Ces deux aspects, la régularité de la trajectoire et la continuité des vitesses angulaires, sont fondamentaux pour assurer un vol sûr et précis du drone, que ce soit dans des environnements dégagés ou complexes. En intégrant ces principes dans le calcul des trajectoires, on optimise la planification des mouvements du drone, offrant une performance

FIGURE 3.4 – Exemple de chemin (Γ_g) généré en utilisant l'algorithme RRT [3]FIGURE 3.5 – Exemple de chemin (Γ_g) généré lorsqu'il n'y a pas d'obstacle [3]

optimale et une meilleure capacité à s'adapter aux contraintes de l'environnement.

Pour cela, on va vouloir lisser les chemins obtenus. Plusieurs algorithmes peuvent être utilisés pour réaliser cela.

Les courbes de Bézier, comme celle présentée en figure 3.6, peuvent être utilisées pour lisser les chemins obtenus à partir d'algorithmes comme A* en introduisant une interpolation douce entre les points du chemin. Les courbes de Bézier sont définies par des points de contrôle qui influent sur la forme de la courbe résultante. Dans le contexte de la planification de trajectoires, on peut utiliser ces courbes pour rendre les virages et les transitions entre les segments du chemin plus fluides.

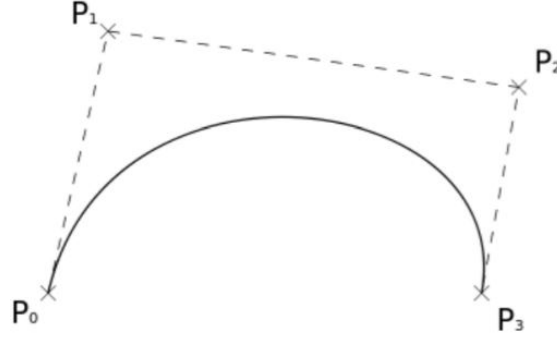


FIGURE 3.6 – Exemple d’une courbe de Bézier générée par 4 points de contrôle [3].

Il existe également d’autres paramétrisations de courbes, comme par exemple les splines cubiques ou quadratiques, dont on peut retrouver en exemple à la figure 3.7. Cependant, le calcul de ces types de splines peut nécessiter la résolution de systèmes d’équations linéaires, ce qui peut être plus coûteux en termes de calculs. Les courbes de Bézier sont souvent plus simples à calculer, et leur évaluation est généralement plus rapide.

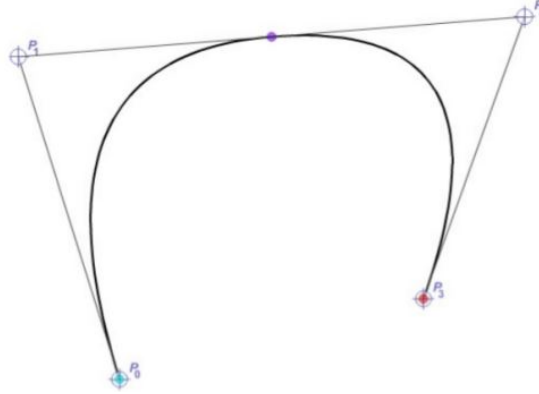


FIGURE 3.7 – Exemple d’une spline quadratique générée par 4 points de contrôle [3]

La formule 3.1 permet de comprendre que la détermination du temps nécessaire pour atteindre un point objectif, symbolisé par la variable τ , constitue en un paramètre ajustable. Ce temps peut être déduit en fixant une vitesse moyenne le long de la courbe de Bézier, ainsi que la longueur B de cette courbe. La fonction s représente l’abscisse curviligne de la courbe en fonction du temps. Ainsi, $B(s(\tau))$ donne la position du drone à l’instant τ .

$$\tau_{f,t} = \tau_{0,t} + \frac{\int_{\tau} B_t(s(\tau))}{v_{moy}} \quad (3.1)$$

La figure 3.8 montre les correspondances entre le temps physique t et le temps nécessaire au drone pour parcourir la courbe entre deux points de contrôle. On calcule la

courbe de Bézier entre t et $t + \Delta t$. En parallèle, on parcourt la courbe B_t entre $\tau_{0,t}$ et $\tau_{f,t}$. Une fois le calcul de B_t fini, on passe au calcul de la courbe suivante, $B_{t+\Delta t}$, c'est-à-dire dès $t + \Delta t$.

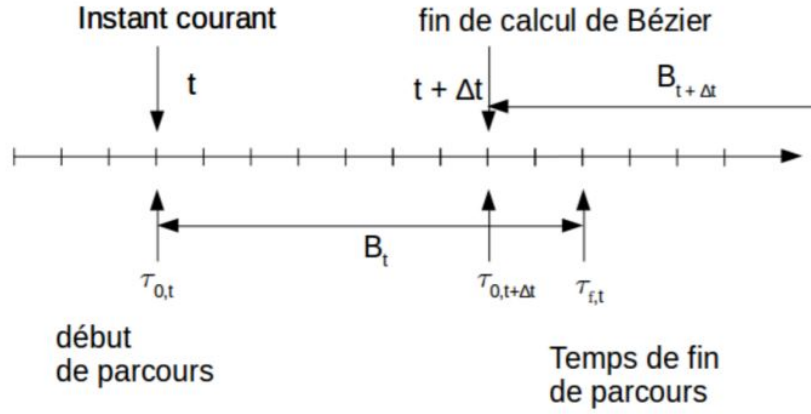


FIGURE 3.8 – Correspondance entre les temps de calcul et de parcours des courbes [3]

3.2.3 Optimisations de la Trajectoire

Le processus d'optimisation vise à déterminer les paramètres optimaux qui définissent la trajectoire du drone. L'objectif principal est de minimiser une fonction de coût tout en respectant des contraintes sur la trajectoire, telles que des limites sur les points de contrôle pour garantir la faisabilité du parcours. Ces contraintes peuvent être formulées mathématiquement et intégrées à la fonction de coût.

Une des optimisations particulièrement intéressantes consiste à ajuster la trajectoire pour suivre au mieux un chemin prédéfini. On peut choisir de minimiser la distance entre la courbe de Bézier B_t et le chemin Γ_g pour rapprocher au mieux la trajectoire finale du drone au chemin prédéfini.

Cependant, il est essentiel de noter que bien que les courbes de Bézier présentent une continuité (continuité $C0$), la continuité de leurs tangentes n'est pas toujours automatiquement assurée (la continuité $C1$ n'est pas induite). Pour assurer la continuité $C1$ d'une courbe de Bézier, il est impératif de garantir que la dérivée première de la courbe est continue à travers les segments successifs. Prenons l'exemple d'une courbe de Bézier cubique définie par quatre points de contrôle P_0, P_1, P_2, P_3 . La formule de la courbe de Bézier est donnée par l'équation 3.2 :

$$B(t) = (1 - t)^3 P_0 + 3(1 - t)^2 t P_1 + 3(1 - t) t^2 P_2 + t^3 P_3 \quad (3.2)$$

Où t est un paramètre variable dans l'intervalle $[0, 1]$ permettant de décrire les différentes positions dans la courbe de Bézier (abscisse curviligne).

La dérivée première de cette courbe par rapport à t est ainsi démontré à la figure 3.3 :

$$B'(t) = -3(1-t)^2(P_0 - P_1) + 3(1-t)(2t-1)(P_1 - P_2) + 3t^2(P_2 - P_3) \quad (3.3)$$

Pour assurer la continuité $C1$ entre deux segments successifs de la courbe de Bézier, il est nécessaire d'établir l'égalité des tangentes à la fin du premier segment avec celles au début du deuxième segment. Cela se traduit par l'équation 3.4 :

$$B'_1(1) = B'_2(0) \quad (3.4)$$

En d'autres termes, les deux derniers points de contrôle de la première partie de la courbe de Bézier doivent être égaux aux deux premiers points de contrôle de la deuxième partie de la courbe.

Par ailleurs, pour assurer la continuité entre deux courbes de Bézier successives, il est impératif que les points de contrôle partagés entre ces courbes soient identiques. En d'autres termes, le dernier point de contrôle de la première courbe de Bézier doit être égal au premier point de contrôle de la deuxième courbe. Cette condition garantit une jonction fluide entre les deux courbes, assurant ainsi la continuité des points. Il serait encore plus judicieux d'assurer la continuité $C1$ (continuité des tangentes) entre deux courbes de Bézier afin d'assurer une continuité dans la vitesse du drone. Cela peut facilement être modélisé, simplement en partageant deux points et non un seul. Ainsi, les deux derniers points de contrôle de la première courbe de Bézier doivent être égaux aux deux premiers points de contrôle de la deuxième courbe.

Dans le contexte de l'optimisation de trajectoire, les méthodes traditionnelles d'optimisation résolvent le problème en ajustant les paramètres de la trajectoire pour minimiser la fonction de coût. Ces approches classiques, comme la descente de gradient, sont largement utilisées pour trouver les valeurs optimales des paramètres. En parallèle, dans le domaine de l'intelligence artificielle, la descente de gradient est également employée comme une technique d'optimisation courante pour régler les paramètres des modèles d'apprentissage automatique en intelligence artificielle.

Par exemple, l'application de variantes de la descente de gradient, comme Adam, pourrait peut-être permettre d'améliorer la précision et l'efficacité des trajectoires générées. De plus, l'intégration de méthodes avancées, telles que la descente de gradient stochastique, pourrait faciliter la gestion de trajectoires dans des environnements dynamiques en temps réel.

4 Synthèse

A la lumière des chapitres 1, 2 et 3, nous avons décidé de retenir une certaine architecture du projet afin d'avancer dans l'implémentation des diverses fonctionnalités. Nous avons notamment choisi de travailler sous ROS2, s'agissant du middleware le plus répandu, et donc potentiellement le mieux documenté. De plus, nous avons préféré la version 2 afin d'éviter de quelconques obsolescences dans les années suivant notre intervention sur le projet.

Parmi un large choix de capteurs aux fonctions diverses disponibles sur le marché, nous pensons utiliser des caméras 3D Intel RealSense (figure 2.6) qui permettent, contrairement aux capteurs LiDAR3D, de visualiser le sol sous le drone. La précision du capteur et la qualité des images produites sont suffisantes pour notre cas d'étude, à savoir la cartographie de la zone de plantation. La localisation par RTK est la méthode qui nous est imposée mais l'étude de différentes technologies nous a permis d'abord d'être convaincu que c'est une méthode efficace et pertinente pour ce projet, et d'étendre notre culture sur ce sujet.

Pour la génération de trajectoires, nous opterons, en première approche, pour des algorithmes simples sur des surfaces rectangulaires (BFP, figure 3.1) comme l'algorithme FLPMW. Si les résultats sont concluants, de nouvelles approches pourront être considérées (algorithme RCPP, cf. section 3.1.2).

Conclusion & perspectives

Cette étude bibliographique a exploré en profondeur les technologies et méthodes actuelles liées à l'utilisation des drones pour des applications telles que la cartographie, la surveillance, et l'agriculture de précision. Nous avons examiné divers aspects, allant de la planification de la trajectoire et l'optimisation de la navigation, aux systèmes de contrôle et aux capteurs utilisés pour la collecte de données. L'accent a été mis sur l'importance de l'adaptabilité et de la précision des drones dans des environnements dynamiques et parfois imprévisibles.

Bibliographie

- [1] A. Andrio, “Development of UAV technology in seed dropping for aerial revegetation practices in Indonesia,” *IOP Conference Series : Earth and Environmental Science*, vol. 308, no. 1, p. 012051, Aug. 2019. [Online]. Available : <https://dx.doi.org/10.1088/1755-1315/308/1/012051>
- [2] J. I. Vasquez-Gomez, M. Marciano-Melchor, L. Valentin, and J. C. Herrera-Lozada, “Coverage Path Planning for 2D Convex Regions,” *Journal of Intelligent & Robotic Systems*, vol. 97, no. 1, pp. 81–94, Jan. 2020. [Online]. Available : <https://doi.org/10.1007/s10846-019-01024-y>
- [3] J. Margraff, “Planification du mouvement 3D en temps réel de drones autonomes dans des environnements dynamiques inconnus,” 2022. [Online]. Available : <https://theses.hal.science/tel-03656405>
- [4] G. Magyar, P. Sinčák, and Z. Krizsán, “Comparison Study of Robotic Middleware for Robotic Applications,” in *Emergent Trends in Robotics and Intelligent Systems*, ser. Advances in Intelligent Systems and Computing, P. Sinčák, P. Hartono, M. Virčíková, J. Vaščák, and R. Jakša, Eds. Cham : Springer International Publishing, 2015, pp. 121–128.
- [5] “Robotics Middleware : A Comprehensive Literature Survey and Attribute-Based Bibliography - ProQuest.” [Online]. Available : <https://www.proquest.com/openview/093e00650cf779b93de0f79351411296/1.pdf?pq-origsite=gscholar&cbl=237781>
- [6] P. M. Newman, *MOOS*, Mar. 2006. [Online]. Available : <https://www.robots.ox.ac.uk/~pnewman/TheMOOS/MOOSII.pdf>
- [7] M. Benjamin and H. Schmidt, “Introduction To MOOS.” [Online]. Available : https://oceanai.mit.edu/2.680/docs/2.680-03-intro_to_moos_2021.pdf
- [8] V. Mazzari, “ROS2 : qu’est-ce qui change par rapport à ROS?” Dec. 2019. [Online]. Available : <https://www.generationrobots.com/blog/fr/ros2-quest-ce-qui-change-par-rapport-a-ros/>
- [9] J. P. M. D. S. G. Ben Grayson, Nigel T. Penna, “Gps precise point positioning for uav photogrammetry,” *The Photogrammetric Record*, Dec. 2018. [Online]. Available : <https://onlinelibrary.wiley.com/doi/10.1111/phor.12259>
- [10] E. MOSTAFA, L. FU, A. IBRAHIM.I., and A. USMAN, “A solution of UAV localization problem using an interacting multiple nonlinear fuzzy adaptive H1 models filter algorithm,” *Chinese Society of Aeronautics and Astronautics & Beihang University*, Jan. 2018. [Online]. Available : <https://www.sciencedirect.com/science/article/pii/S1000936119300937>
- [11] C. G. A. M. A. V. Nicola Angelo Famiglietti, Gianpaolo Cecere, “A test on the potential of a low cost unmanned aerial vehicle rtk/ppk

- solution for precision positioning,” *Sensors*, Jun. 2021. [Online]. Available : <https://www.mdpi.com/1424-8220/21/11/3882>
- [12] “Is rtk the future of drone mapping?” *Drone Engr.* [Online]. Available : <https://www.droneassemble.com/is-rtk-the-future-of-drone-mapping/>
- [13] E. G. . C. H. C. C. Lane Scher, “Drone-based photogrammetry for the construction of high-resolution models of individual trees,” *Trees*, May 2019. [Online]. Available : <https://link.springer.com/article/10.1007/s00468-019-01866-x>
- [14] “Etude expérimentale du lidar robosense rs-lidar16.” [Online]. Available : <https://www.generationrobots.com/blog/fr/etude-experimentale-du-lidar-robosense-rs-lidar16/>
- [15] a. W.-H. L. Young-Hoon Jin, Kwang-Woo Ko, “An indoor location-based positioning system using stereo vision with the drone camera,” *Mobile Information Systems*, Oct. 2018. [Online]. Available : <https://www.hindawi.com/journals/misy/2018/5160543/>
- [16] “Intel® realsense™ depth camera d455.” [Online]. Available : <https://www.intelrealsense.com/depth-camera-d455/>
- [17] J. Jin and L. Tang, “Optimal Coverage Path Planning for Arable Farming on 2D Surfaces.” [Online]. Available : <https://doi.org/10.13031/2013.29488>