

UNIVERSITÉ POLYTECHNIQUE HAUTS-DE-FRANCE THÈSE

Soutenance prévue le 13 septembre 2018
pour obtenir le titre de

DOCTORAT EN INFORMATIQUE

par

Zeineb BAKLOUTI

Système de planification de chemins aériens en 3D : préparation de missions et replanification en cas d'urgence

Composition du jury

<i>Rapporteurs :</i>	Nicolas DURAND	Professeur	ENAC, Toulouse
	Luc JAULIN	Professeur	Lab-STICC, ENSTA-Bretagne
<i>Examineurs :</i>	Feng CHU	Professeur	IBISC, Université d'Évry
	Laetitia JOURDAN	Professeur	CRISTAL, Université de Lille1
	Jean-Loup FARGES	Ingénieur de Recherche	ONERA Toulouse
<i>Invités :</i>	Nicolas BELANGER	Innovation Manager	Airbus Helicopters
	Alain DUCOLLET	Expert de la navigation	Airbus Helicopters
<i>Directeurs :</i>	Rabie BEN ATITALLAH	MdC, HdR	LAMIH, UPHF
	Abdelhakim ARTIBA	Professeur	LAMIH, UPHF
<i>Co-encadrant :</i>	David DUVIVIER	Professeur	LAMIH, UPHF

UNIVERSITÉ POLYTECHNIQUE HAUTS-DE-FRANCE

ECOLE DOCTORALE SCIENCES POUR L'INGÉNIEUR UNIVERSITÉ LILLE NORD-DE-FRANCE-072

Laboratoire d'Automatique, de Mécanique et d'Informatique industrielles et Humaines 8201

UVHC, Le Mont Houy, 59313 Valenciennes Cedex 9

Tél. : +33 (0)3 27 51 13 50 – Télécopie : +33 (0)3 27 51 19 40

Remerciements

Tout d'abord, mes remerciements s'adressent à l'université de Valenciennes et du Hainaut-Cambrésis et AIRBUS Helicopters qui m'ont permis de mener mes travaux de recherches dans les meilleures conditions.

Ensuite, Je tiens à remercier mon directeur de thèse ainsi que mes encadrants pour leur dévouement à leur fonction et leur professionnalisme.

J'adresse ma profonde gratitude à mon directeur de thèse Dr. HDR Rabie Ben Atitallah qui a renforcé ma motivation et mon intérêt pour ce travail et m'a formé sur des sujets de recherche innovants pour Airbus Helicopters. Je le remercie pour sa disponibilité et son aide précieuse.

Je tiens à remercier M. Pierre Feyzeau Ingénieur Navigation chez AH pour avoir assuré le suivi de ma thèse. Toute ma reconnaissance pour son encadrement et ses conseils judicieux.

Je tiens à remercier également mon encadrant industriel Dr. Nicolas Belanger pour son accueil chaleureux lors de mes déplacements chez AIRBUS Helicopters ainsi que pour ses multiples encouragements.

Je remercie Pr Nicolas Durand et Pr Luc Jaulin qui ont accepté de rapporter mes travaux de thèse. Et je remercie aussi l'ensemble des membres du jury pour l'intérêt qu'il a porté à ces travaux.

Je remercie Pr Jean-François Côté pour son aide, ses conseils et son soutien indéfectible lors de mon stage de recherche au laboratoire de recherche CIRRELET (Québec).

Enfin je tiens à saluer les personnes qui de près ou de loin ont contribué à la concrétisation et à l'achèvement de ces travaux, dans le cadre de ma thèse de doctorat.

Je remercie mes collègues de bureau Karim et Walid pour l'ambiance chaleureuse qu'ils ont su créer.

J'adresse également tous mes remerciements au personnel du Laboratoire LAMIH. Merci pour votre aide et votre soutien sans faille.

Merci à mes amis Anis, Hanane, Hiba, Lydia, Marko, Tariq et Yunfei, sur lesquelles j'ai pu compter pour surmonter les moments les plus difficiles (il y en aura d'autres c'est sûr) !

Merci à ma mère et mon père qui ont tout fait pour moi. Je tiens à vous exprimer ma reconnaissance éternelle pour votre soutien affectif et moral.

Pour terminer je tiens à remercier mon frère Ahmad et ma sœur Fatma qui ont été présents pendant les étapes importantes de ma vie. Je vous exprime ma reconnaissance pour votre amour, votre bonne humeur et vos encouragements !

Résumé

L'enjeu de planification de vol à bord d'un hélicoptère en tenant compte des différents paramètres environnementaux (terrain, conditions climatiques, situation d'urgence, etc.) constitue un facteur clé dans le secteur aéronautique afin d'assurer une mission en toute sécurité avec un coût réduit. Ce défi concerne à la fois la phase de préparation de la mission sur une station au sol mais aussi en cours de vol pour faire face à un événement imprévu (panne mécanique, obstacle dynamique, mauvaises conditions météorologiques, etc.). Nous citons un premier exemple de mission de type recherche et sauvetage qui dispose d'un temps limité pour localiser et rechercher des personnes en danger. Pour ce faire, le plan de vol généré doit suivre le relief du terrain à des altitudes relativement basses entre des points de passage désignés. L'objectif est de permettre au pilote de localiser une victime dans une durée bornée. Un deuxième type de mission comme l'assistance médicale a la particularité d'assurer un vol qui favorise le confort du passager ainsi qu'une route qui minimise le temps de vol selon la criticité de la situation. Pendant la phase dynamique (en cours de vol), lorsqu'il s'agit d'un événement complexe telle qu'une panne moteur, une replanification de la mission devient nécessaire afin de trouver un chemin aérien qui permet d'atterrir en toute sécurité dans les plus brefs délais. Dans un autre exemple comme l'évitement d'obstacle dynamique ou de zone dangereuse, il s'agit de calculer un autre plan de vol pour atteindre la destination. Cependant à ce jour, les pilotes ne bénéficient pas de système d'autoroutage 3D permettant la replanification dynamique de mission face à une situation d'urgence.

Face au défi de génération d'un plan de vol optimal, nos réflexions profondes ont abouti à la proposition d'un nouveau système de planification de chemin, pour des aéronefs, dédié à la préparation de mission avant le vol ainsi qu'à la replanification dynamique en cours de vol face à une situation d'urgence. Le système de planification peut être déployé sur une station au sol ou bien intégré comme fonction avionique au sein de l'aéronef. Les résultats obtenus ont abouti à un brevet déposé devant l'INPI et à plusieurs actions de transfert technologique au sein d'Airbus Helicopters. Le système s'appuie sur des techniques de discrétisation de l'espace et de calcul du plus court chemin afin de générer automatiquement des solutions flexibles en terme de profil de chemin en respectant plusieurs contraintes liées à l'appareil, le terrain et l'environnement. La solution proposée est générique et capable de s'adapter selon le type de l'aéronef, le type de la mission et la fonction objectif à minimiser. Le système de planification peut offrir des solutions avec différents compromis entre le temps d'exécution et la qualité du chemin selon le temps disponible qui peut être corrélé avec la criticité de la situation. Le fonctionnement du système de planification de chemin se compose principalement de deux phases : le prétraitement et le routage. La phase de prétraitement permet une discrétisation multi-altitude de l'espace 3D selon une précision donnée et la génération automatique d'un graphe de navigation avec une connexité paramétrable. La phase de routage 3D calcule le chemin en prenant en considération un ensemble de contraintes telles que les limitations an-

gulaires en horizontal et en vertical et une fonction objectif à minimiser tels que la distance, le carburant, le temps, etc. Lors de la phase d'exploration du graphe, le système de planification peut communiquer avec le modèle de performance de l'appareil afin de minimiser une fonction coût liée à la performance et de s'assurer au fur et à mesure de la faisabilité de la mission. La génération automatique des plans de vol et la replanification dynamique représentent une brique essentielle pour concevoir des systèmes d'assistance au pilote ainsi que des aéronefs autonomes. En résumé, nous avons réussi à atteindre les attendus de l'industriel à savoir : l'évaluation de la faisabilité d'une mission, l'amélioration des performances, la réduction de la charge de travail « navigation », ainsi que l'amélioration de la sécurité en vol.

Afin de concrétiser la solution proposée, nous avons conçu un nouvel outil de planification automatique de vol 3D. Notre outil, baptisé DTANAV « Demonstration Tool for Aircraft NAVigation », permet d'appliquer le procédé de planification d'un chemin aérien sur des scénarios réels. A travers l'interface de l'outil, l'utilisateur a la possibilité de saisir des paramètres liés à la mission (point de départ, point d'arrivée, les caractéristiques de l'appareil, plafond de navigation, etc.) et d'autres qui agissent sur la partie algorithmique afin de maîtriser la qualité et le profil de la solution générée. L'outil DTANAV a fait l'objet de plusieurs démonstrations et actions de transfert technologique chez Airbus Helicopters.

Mots-clés : Planification de chemins aériens - Discrétisation de l'espace 3D - Graphe de navigation - Plus court chemin - Replanification de missions.

Abstract

Helicopter flight planning is a key factor in the aeronautics domain in order to ensure a safe mission at a reduced cost taking into consideration different environmental parameters like terrain, weather, emergency situations, etc. This challenge concerns both the mission preparation phase using a ground station and also during the flight to cope with a complex event (mechanical failure, dynamic obstacle, bad weather conditions, etc.).

We quote as a first example a Search and Rescue mission that should be performed in limited time to locate persons in danger. To achieve that, the generated flight path must follow the terrain profile at relatively low altitudes crossing the predefined waypoints. The objective is to allow the pilot to locate a victim in a short time. Another example for a mission of medical assistance where the helicopter should ensure the comfort of patients as well as minimizing the flight time to respect critical situations. During the flight, when a complex event occurs such as engine failure, re-planning the mission becomes necessary in order to find a new path that could guarantee safe landing. Unfortunately, pilots do not benefit from an embedded 3D path planning system that enables dynamic mission re-planning in case of emergency.

To tackle the challenge of generating an optimal flight plan, we proposed new path planning system dedicated to mission preparation and dynamic path re-planning during critical situations. The planning system can be deployed on a ground station or embedded as an avionic function in the aircraft. The achieved results are registered as a patent at National Institute of Intellectual Property and deployed inside Airbus Helicopters through several technology transfers. The system relies on 3D space discretization and shortest path planning techniques to generate automatically flexible path profiles that respect several. The proposed solution is generic and is able to adapt to the aircraft model, mission and the objective function to be minimized. The path planning system can offer solutions with different tradeoffs between timing and path quality within the available runtime depending on the criticality of the situation.

The functioning of the path planning system consists mainly of two phases : preprocessing and routing phase. The preprocessing allows a multi-altitude discretization of the 3D space according to a given precision and generate automatically a navigation graph with a configurable density. The 3D routing phase calculates the path by considering a set of constraints such as horizontal and vertical angular limitations and the objective function such as distance, fuel, time, etc. During the graph exploration phase, the path planning system can communicate with the aircraft performance model to evaluate a given criterion. Automatic flight path generation is an essential building block for designing pilot assistance systems and autonomous aircraft. In summary, we succeeded to reach the industrial expectations namely : the evaluation of mission feasibility, performances improvement, navigation workload reduction, as well as improving the flight safety.

In order to realize the proposed solution, we designed a new tool for automatic 3D flight path planning. We named our tool DTANAV : Demonstration Tool for Aircraft NAVigation. It allows to apply the proposed planning process on real scenarios. Through the tool interface, the user has the possibility to set the parameters related to the mission (starting point, end point, aircraft model, navigation ceiling, etc.). Other algorithmic parameters are defined in order to control the quality and the profile of the generated solution.

Keywords : Path planning - 3D space discretization - Navigation graph - Shortest path
- Dynamic mission replanification.

Table des matières

Table des matières	vii
1 Introduction	1
1.1 Contexte	1
1.2 Problématique	3
1.3 Contributions	4
1.4 Plan de la thèse	7
2 État de l’art	9
2.1 Introduction	9
2.2 État de l’art industriel	10
2.2.1 Les pratiques actuelles au sein d’Airbus Helicopters	10
2.2.2 Étude de quelques brevets du domaine	10
2.3 Techniques de planification de chemins aériens	13
2.3.1 Critères de performances des algorithmes	14
2.3.2 Première catégorie de méthodes : modélisation de l’espace non nécessaire	15
2.3.3 Deuxième catégorie de méthodes : avec modélisation de l’espace . . .	17
2.4 Discrétisation de l’environnement	18
2.4.1 Roadmap decomposition	18
2.4.2 Cell decomposition	20
2.5 Algorithmes du plus court chemin	25
2.5.1 A star (A^*)	25
2.5.2 Variantes de A^*	26
2.6 Synthèse et positionnement	28
2.7 Conclusion	28
3 Système et procédé de planification d’un chemin aérien pour aéronef	31
3.1 Introduction	31
3.2 Le problème industriel posé	31
3.3 Un nouveau système de planification de chemin aérien	32

3.3.1	Principe de la solution apportée	32
3.3.2	Description détaillée de l'invention	33
3.4	Description chronologique détaillée	39
3.4.1	La phase de prétraitement	39
3.4.2	La phase d'optimisation de la trajectoire de décollage et d'approche	49
3.4.3	La phase de routage	50
3.4.4	La phase de simplification du plan de vol	54
3.5	Avantages et restrictions	56
3.6	Conclusion	57
4	Méthodes de discrétisation appliquées à la planification de vol	59
4.1	Introduction	59
4.2	Étude de scénarios réels et critères d'évaluation	60
4.3	Discrétisation 2D : Quadtree et Grille régulière	61
4.3.1	Le Quadtree	62
4.3.2	Construction du graphe de navigation	63
4.3.3	Résultats	66
4.3.3.1	La phase de prétraitement	66
4.3.3.2	La phase de routage	68
4.4	Algorithme de recherche du plus court chemin : Any-angle adaptive A*	72
4.5	Discrétisation 3D : Grille régulière 3D, Octree, Multi-Quadtree	75
4.5.1	Structures de discrétisation	77
4.5.2	Construction des graphes de navigation	79
4.5.3	Résultats	81
4.5.3.1	La phase de prétraitement	81
4.5.3.2	La phase de routage	83
4.6	Conclusion	85
5	DTANAV : environnement de planification et d'exploration de chemins aériens	89
5.1	Introduction	89
5.2	Description des fonctionnalités de l'outillage	90
5.2.1	Diagramme d'utilisation de l'outillage	93
5.3	DTANAV : Demonstration Tool for Aircraft NAVigation	95
5.3.1	Description générale de l'outil DTANAV	95
5.3.2	Utilisant de l'outil DTANAV	98
5.4	Modèle de performances	107
5.4.1	Évaluation du coût d'un segment de vol selon différents critères	107
5.4.2	Déduction des angles de montée/descente	110

5.5	Scénarios de replanification dynamique	113
5.6	Conclusion	118
6	Conclusion et Perspectives	121
6.1	Bilan	122
6.2	Perspectives	124
6.2.1	Système de planification multi-critères	124
6.2.2	Système d'assistance aux pilotes en cas d'urgence	126
6.2.3	Conception d'un nouveau service de transport : Urban Air Mobility .	126
	Bibliographie	129

Chapitre 1

Introduction

1.1 Contexte

L'enjeu de planification de vol à bord d'un hélicoptère en tenant compte des différents paramètres environnementaux (terrain, conditions climatiques, situation d'urgence, etc.) constitue un facteur clé dans le secteur aéronautique afin d'assurer une mission en toute sécurité et avec un coût réduit. Ce défi concerne à la fois la phase de préparation de la mission sur une station au sol mais aussi en cours de vol pour faire face à un événement imprévu (panne mécanique, obstacle dynamique, mauvaises conditions climatiques, etc.). Nous citons un premier exemple de mission de type *recherche et sauvetage* qui dispose d'un temps limité pour localiser et chercher des personnes en danger. Pour ce faire, le plan de vol généré doit suivre le relief du terrain à des altitudes relativement basses entre des points de passage désignés. L'objectif est de permettre au pilote de localiser une victime dans une durée bornée. Un deuxième type de mission comme l'*assistance médicale* a la particularité d'assurer un vol qui favorise le confort des passagers ainsi qu'une route qui minimise le temps de vol selon la criticité de la situation. En cours de vol, lorsqu'il s'agit d'un événement complexe telle qu'une panne moteur, une replanification de la mission est nécessaire afin de trouver un plan de vol qui permet d'atterrir en toute sécurité dans les plus brefs délais. Dans un autre exemple comme l'évitement d'un obstacle dynamique ou une zone dangereuse, il s'agit de trouver un autre plan de vol pour atteindre la destination.

Cependant à ce jour, les pilotes ne bénéficient pas d'un système d'autoroutage 3D permettant l'assistance pour la replanification dynamique de leur mission face à un imprévu complexe ou une situation d'urgence. À titre d'exemple, les rapports des organisations internationales telles que l'IHST (International Helicopter Safety Team) et l'EHST (European Helicopter Safety Team) ont mis l'accent sur les principaux facteurs concernant les accidents des hélicoptères. Il s'avère que l'événement indésirable complexe « entrée dans des conditions IMC » (Instrument Meteorological Conditions), dû à la perte de références visuelles extérieures, couvre un pourcentage important des accidents opérationnels. Face à ce défi

scientifique et industriel, un projet de collaboration entre l'Université de Valenciennes et du Hainaut-Cambrésis (UVHC) et Airbus Helicopters¹ a fait l'objet de la conception d'un système d'assistance aux pilotes qui intègre un outil de planification de missions. Ce dernier doit permettre de générer un plan de vol en préparation de mission ou de replanifier une mission en cours de vol face à un événement complexe. La solution doit être fournie dans une durée fortement liée à la criticité de la situation par une station au sol ou un ordinateur embarqué dans le système avionique. Étant donné la diversification et le niveau de criticité des situations possibles, la recherche d'une solution correspondante à un plan de vol optimal doit être associée à une ou plusieurs contrainte(s) (temps de calcul, carburant disponible, état de l'appareil, etc.) et à un ou plusieurs objectif(s) (minimiser la distance ou le temps de vol, nombre de changements de cap et/ou d'altitude, etc.). Les plans de vol proposés devront tenir compte des contraintes fortes tout en optimisant un ou plusieurs objectifs. Plusieurs plans de vol pourront être présentés à l'équipage sur la base d'un classement multi-critère.

Par ailleurs, la navigation dans un environnement partiellement connu et complété dynamiquement à l'aide de capteurs évolués (radar, vision, etc.) vient augmenter la complexité du problème. Cette complexité est présente à la fois au niveau des algorithmes d'optimisation des chemins, mais elle s'exprime également au niveau des structures de données et des algorithmes sous-jacents utilisés pour stocker et explorer efficacement l'environnement. La nécessité d'appliquer des solutions déterministes et bornées au niveau du temps d'exécution accroît la difficulté de mise au point des algorithmes. Les évolutions technologiques au niveau des ordinateurs embarqués, des capteurs extéroceptifs et des moyens de transmission ont permis l'émergence des systèmes de type « Sense and Avoid » dans le secteur aéronautique. Bien que ces systèmes soient aujourd'hui utilisés pour la navigation autonome des drones, leur déploiement dans un contexte de replanification dynamique de vol au niveau d'un aéronef complexe tel qu'un hélicoptère représente un défi scientifique et industriel. Par conséquent, un système d'assistance pour la replanification de vol à bord d'un hélicoptère, répondant aux exigences précédemment décrites, constitue un facteur différenciateur clé dans la thématique *sécurité du transport aérien*. Pour atteindre cet objectif, nous avons besoin d'élaborer des solutions innovantes sur le plan algorithmique permettant de s'adapter au type de la mission, le relief du terrain ainsi qu'aux paramètres de l'appareil et de l'environnement.

Face au défi de génération d'un plan de vol optimal, des réflexions profondes au sein d'un groupe de travail incluant des chercheurs de l'UVHC et Airbus Helicopters (dans le cadre de la thèse d'Omar Souissi [59]) ont abouti à un état de l'art sur les différentes techniques de planification de chemin 2D ainsi que des orientations scientifiques concernant la discrétisation de l'espace 3D. Dans le cadre de cette thèse, nous nous intéressons à la génération automatique de plans de vol 3D. Il s'agit de trouver l'ensemble des points de passage dans un environnement discrétisé pour atteindre une destination en prenant en considération le relief du terrain,

1. Airbus Helicopters : une filiale du Groupe Airbus connu comme le premier fabricant d'hélicoptères civils au monde et l'un des principaux constructeurs d'hélicoptères militaires

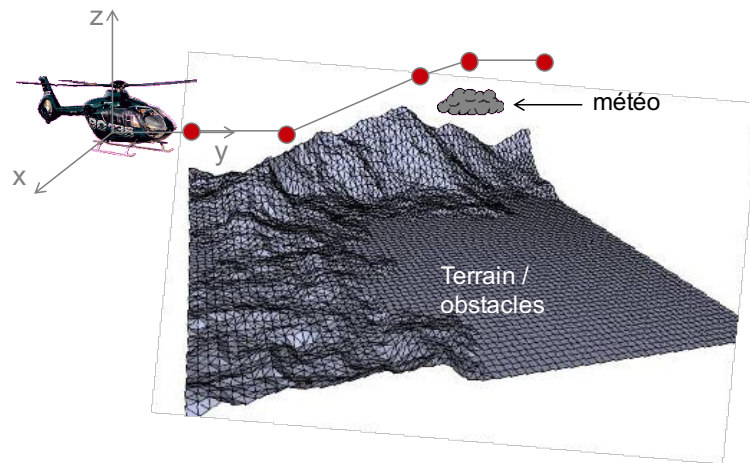


FIGURE 1.1 – Planification de chemin 3D

les contraintes de l'appareil, les conditions météorologiques, le type de mission, etc (voir Fig. 1.1). Le travail réalisé fait appel aux techniques de discrétisation de l'espace ainsi qu'aux algorithmes d'optimisation pour le calcul du plus court chemin.

1.2 Problématique

Jusqu'à présent, la phase de planification de vol en 3D ne bénéficie pas d'un procédé automatique pour la préparation et l'évaluation de la faisabilité d'une mission en prenant en considération le type de la mission, les contraintes de l'appareil, etc. En plus, la replanification en cours de vol avec la calibration de la qualité des solutions en fonction de la criticité de la situation et l'environnement incertain ouvrent de nombreuses opportunités pour la recherche afin de proposer un système de génération automatique de plans de vol à bord d'un hélicoptère. Dans ce cadre, nous avons identifié trois principaux facteurs qui définissent la problématique de notre thèse :

— Un nouveau système de planification de chemins aériens 3D

Les nouvelles générations d'aéronefs en particulier les hélicoptères assurent des missions critiques dans un environnement incertain et ceci requiert des systèmes d'assistance pour aider les pilotes à surmonter des cas d'urgence. *Aujourd'hui, nous avons besoin de définir de nouvelles fonctionnalités au niveau de l'architecture générale du système avionique afin d'augmenter la capacité de l'aéronef à faire face à un événement complexe.* Ce défi nous incite à revisiter le procédé de planification de missions au niveau des stations au sol ainsi qu'au niveau des équipements embarqués au sein de l'appareil. Afin de répondre à ce défi d'ordre technologique, il est nécessaire d'étudier les architectures avioniques existantes et les procédés de planification correspondants dans le but de proposer un

système innovant capable de s'adapter selon le type de la mission, la criticité de la situation, l'environnement extérieur, etc.

— **Maîtrise de la complexité temporelle et spatiale des méthodes de planification de chemins**

Le nouveau système de planification de chemins aériens 3D que nous envisageons concevoir doit répondre à des exigences spécifiques du domaine aéronautique en termes de déterminisme, temps réel, etc. *Pour cette raison, il est nécessaire de maîtriser la complexité temporelle et spatiale des méthodes proposées.* Cet enjeu couvre le domaine technique de discrétisation de l'espace en 2D et en 3D ainsi que le domaine d'optimisation pour le calcul du plus court chemin. Il existe une variété de méthodes de discrétisation offrant différents types de maillage qui impactent le temps et la qualité des chemins générés. De même, plusieurs algorithmes de calcul du plus court chemin pourront être utilisés. L'enjeu scientifique consiste à trouver une solution déterministe et paramétrable (précision de la discrétisation, caractéristiques de l'appareil, type de la mission, etc.) avec une maîtrise de la complexité temporelle et spatiale.

— **Environnement automatique pour la planification des chemins aériens**

Dans l'industrie, les soucis de réduction de la charge de travail, d'amélioration des performances de la mission, et la réduction des coûts sont bien connus. *Face à ces difficultés, il est nécessaire d'utiliser un environnement automatique pour la planification des chemins aériens.* Cet environnement doit offrir un cadre méthodologique et technologique qui permet d'unifier le processus quel que soit le type de l'appareil ou de la mission et ceci en ajustant un nombre réduit de paramètres (les limitations angulaires de l'appareil, le plafond de navigation, etc.). L'outil proposé doit être accessible par le personnel chargé de la planification de missions ou bien par les pilotes sous forme d'un logiciel installé sur une station au sol ou bien une nouvelle fonction embarquée au sein de l'appareil. Par ailleurs, l'outil doit permettre d'explorer différentes solutions possibles avec une évaluation de performances en termes de distance, temps de vol, consommation carburant, etc. L'efficacité de cet environnement automatique doit être interprétée par une réduction de la charge de travail de l'équipe « navigation » et par l'amélioration de la sécurité en vol.

1.3 Contributions

Notre contribution dans le domaine de génération automatique de plans de vol pour des aéronefs consiste à proposer des solutions permettant de lever les verrous scientifiques et industriels présentés dans la section précédente. En résumé nos contributions sont :

— **Procédé de planification d'un chemin aérien pour des aéronefs et le système de planification associé**

Cette contribution a fait l'objet d'un brevet déposé à l'INPI (Institut National de la Propriété Intellectuelle) décrivant un procédé innovant pour la planification de chemins aériens 3D dédié à la préparation de mission avant le vol ainsi qu'à la replanification dynamique en cours de vol face à une situation d'urgence. Le procédé repose sur une méthode de discrétisation de l'espace 3D, la construction d'un graphe de navigation et un algorithme de routage en tenant compte d'un certain nombre de paramètres d'entrée, des contraintes liées à l'environnement et à l'appareil et une fonction objectif à minimiser. Notre approche permet d'ajuster la qualité de la solution en fonction du temps d'exécution disponible en s'appuyant sur un nombre réduit de paramètres (principalement la précision de discrétisation de l'espace et le degré de connexité du graphe de navigation). Par ailleurs, le procédé proposé permet d'obtenir différents profils de plan de vol selon le type de la mission. Le système de planification associé à ce procédé peut être déployé sur une station au sol ou bien intégré comme une fonction avionique au sein de l'aéronef. Afin de maîtriser les temps d'exécution, notre système de planification de chemin fonctionne principalement en deux phases : une phase de prétraitement et une phase de routage. La phase de prétraitement permet une discrétisation de l'espace 3D selon une précision donnée et la génération automatique d'un graphe de navigation avec une densité paramétrable. La phase de routage 3D calcule le chemin en prenant en considération un ensemble de contraintes liées à la mission et à l'appareil afin de calculer un chemin aérien selon un critère donné (exemple : minimiser la distance ou le temps de vol). *La généricité en fonction du type de l'appareil, le type de la mission et la criticité de la situation est l'attrait principal de notre solution.*

— **Étude comparative des méthodes de discrétisation pour la planification 3D**

Cette contribution est le fruit d'un travail d'exploration et de comparaison de différentes méthodes de discrétisation de l'espace en 2D et 3D afin de converger vers une approche capable de répondre aux exigences du problème industriel posé. L'objectif est de proposer une méthode de discrétisation bien adaptée aux terrains étendus (des missions de l'ordre de quelques centaines de kilomètres) avec une maîtrise des temps d'exécution des algorithmes de calcul du plus court chemin ainsi que l'occupation mémoire des structures de données associées. Pour ce faire, nous avons étudié tout d'abord des méthodes conventionnelles avec un maillage régulier (grille 2D et 3D) et irrégulier (Quadtree et Octree) et nous avons démontré leurs limitations pour répondre aux exigences de notre domaine d'application. Pour cette raison, nous avons proposé une nouvelle approche multi-altitude en subdivisant l'espace 3D sous forme de plusieurs paliers 2D qui vont être discrétisés avec des grilles irrégulières (Quadtree). Cette approche permet de réduire la complexité en temps et en mémoire en réalisant une décorrélation entre les deux composantes verticale et horizontale de l'espace. Les Quadtrees seront utilisées pour créer un maillage adaptatif selon la densité des obstacles sur chaque palier. La précision du

maillage pourra être fixée afin d’obtenir des angles de montée et de descente qui respectent les caractéristiques de l’aéronef. Dans l’étape suivante nous avons proposé de construire le graphe de navigation en générant des arcs horizontaux sur chaque palier selon un voisinage variable et des arcs verticaux pour le raccordement entre les différentes altitudes selon les angles de montée et de descente tolérés par l’appareil. *Cette façon de construire le graphe de navigation permet d’obtenir un degré de connexité variable, ce qui permet d’ajuster la qualité de la solution en fonction du temps disponible et ceci en utilisant un nombre réduit de paramètres.* La densité du graphe dépend du temps disponible pour le prétraitement et agit fortement sur la qualité de la solution pendant la phase de routage. En effet, un graphe fortement connecté permet d’obtenir des chemins de très bonne qualité qui convergent rapidement vers la destination. Cette qualité est traduite par la distance de vol, le nombre de segments et le temps de calcul du chemin. De plus, l’ensemble étendu des solutions offre plus de possibilités pour calculer des chemins aériens même pour des scénarios ayant des contraintes difficiles à satisfaire. Ainsi, de tels scénarios exigent un routage automatique afin d’assurer la planification de la mission. Par ailleurs, les temps d’exécution issus de scénarios réels sont très satisfaisants (de l’ordre de quelques secondes) et répondent au besoin de la replanification dynamique en cas d’urgence.

— **DTANAV : environnement de planification et d’exploration de chemins aériens 3D**

Afin de concrétiser les solutions proposées, cette contribution concerne la conception et l’implémentation d’un outil de planification automatique de vol 3D. Cet outil met en œuvre les différentes approches scientifiques étudiées afin d’assurer un environnement convenable pour le personnel chargé de la préparation des missions ou bien les pilotes à bord d’un hélicoptère. Notre outil, baptisé DTANAV « Demonstration Tool for Aircraft NAVigation », permet d’appliquer le procédé de planification d’un chemin aérien sur des scénarios réels. À travers l’interface de l’outil, l’utilisateur a la possibilité de saisir des paramètres liés à la mission (point de départ, point d’arrivée, les caractéristiques de l’appareil, plafond de navigation, etc.) et d’autres qui agissent sur la partie algorithmique afin de maîtriser la qualité et le profil de la solution générée. Cet outil est considéré aussi comme un environnement d’exploration de plusieurs chemins possibles avec une évaluation des coûts (distance ou temps) tout en intégrant le modèle de performance d’un hélicoptère donné. *L’outil DTANAV a fait l’objet de plusieurs démonstrations et d’une action de transfert technologique chez Airbus Helicopters.* En effet, la génération automatique des plans de vol 3D et la replanification dynamique en cas d’urgence, représente une brique essentielle pour concevoir des systèmes d’assistance aux pilotes ainsi que des aéronefs autonomes. *En résumé, nous avons réussi à atteindre les attendus de l’industriel à savoir : l’évaluation de la faisabilité d’une mission, l’amélioration des*

performances, la réduction de la charge de travail « navigation », ainsi que l'amélioration de la sécurité en vol.

1.4 Plan de la thèse

En respectant l'ordre de nos contributions décrites ci-dessus, notre manuscrit est organisé selon le plan suivant :

- **Chapitre 2 : État de l'art.** Dans ce chapitre, nous présentons le contexte de nos travaux qui abordent les techniques de planification de chemins aériens en s'appuyant sur des brevets issus des principaux industriels du domaine aéronautique. Les méthodes de discrétisation de l'espace et les algorithmes de calcul du plus court chemin font également partie de notre analyse de l'existant. À partir de cette étude, nous positionnons nos travaux et nous évoquons les grandes lignes de nos contributions.
- **Chapitre 3 : Système et procédé de planification d'un chemin aérien pour aéronef.** Ce chapitre décrit tout d'abord le problème industriel lié à ce projet de thèse. Ensuite, nous présentons une description détaillée de notre invention concernant un nouveau procédé de planification de chemins aériens 3D. Un exemple de mode de réalisation, les avantages et les limitations de notre solution sont aussi discutés dans ce chapitre.
- **Chapitre 4 : Méthodes de discrétisation appliquées à la planification de vol.** Dans ce chapitre nous nous intéressons aux méthodes de discrétisation et de planification de chemins en 2D et 3D. Plusieurs techniques sont présentées avec une étude comparative en termes de complexité spatiale et temporelle. Cette étude nous a permis de donner des arguments scientifiques concernant les performances de notre solution par rapport aux méthodes de planification de chemins classiques.
- **Chapitre 5 : DTANAV : environnement de planification et d'exploration de chemins aériens.** Ce chapitre décrit DTANAV l'outil de génération automatique de chemins aériens que nous avons développé. Par ailleurs, Il met l'accent sur l'utilisation des fonctions de performances (distance, temps, etc.) de l'appareil pour pouvoir optimiser des chemins selon plusieurs critères. L'utilisation de notre outil afin de générer des chemins avec différents types de mission est démontrée à travers plusieurs exemples issus de scénarios réels.
- **Chapitre 6 : Conclusion et perspectives.** Nous concluons cette thèse par le bilan des travaux effectués et nous détaillons les contributions apportées avant d'aborder quelques perspectives à nos travaux.

Chapitre 2

État de l’art

2.1 Introduction

Les hélicoptères ont de nombreuses applications potentielles, telles que les missions de surveillance et de reconnaissance, les activités de défense du territoire, les opérations de recherche et de sauvetage, la surveillance du trafic, etc. Un système de planification automatique de vol permettra d’améliorer la sécurité en vol et d’aider le personnel à assurer les missions dans des conditions favorables. Comme indiqué dans le chapitre précédent, ce travail s’inscrit dans le cadre d’une collaboration industrielle avec Airbus Helicopters. Il vise le développement d’un système de planification de missions en 3D pour des aéronefs. Pour cette raison, il s’avère nécessaire que notre étude des techniques existantes couvre à la fois des références industrielles (sous forme de brevets) et académiques.

Dans la littérature, de nombreux travaux récents portent sur la planification de vol. Toutefois, jusqu’à présent le problème de préparation de chemin aérien de longue distance n’a pas été profondément étudié. En effet, ce problème engendre une complexité algorithmique incapable de répondre aux exigences de temps réel imposées par la criticité des missions avioniques surtout en cas d’urgence. Par ailleurs, le problème de planification de chemins n’est pas restreint au domaine aéronautique. De ce fait, notre état de l’art évoquera les méthodes de planification de chemin appliquées au transport aérien mais aussi le transport maritime, la robotique, les jeux vidéo, etc. Notre sujet de thèse nous ramène à étudier différents aspects scientifiques liés à la discrétisation de l’espace 3D, l’exploration de graphe et les algorithmes de calcul du plus court chemin.

Cet ensemble d’objectifs fait que ce chapitre est structuré comme suit : la Section 2.2 survole les techniques de planification utilisées par des industriels du domaine aéronautique à travers l’étude de plusieurs brevets. Un tour d’horizon des méthodes de planification de mouvement les plus utilisées dans la littérature sera présenté dans la Section 2.3. Les méthodes de discrétisation de l’espace en 2D et 3D seront étudiées dans la Section 2.4. Ensuite, la Section 2.5 détaille les algorithmes de calcul du plus court chemin. Enfin, la Section 2.6 offre

une synthèse et un positionnement de nos travaux par rapport à l'état de l'art.

2.2 État de l'art industriel

L'objectif de cette section est tout d'abord d'introduire les pratiques industrielles au sein d'Airbus Helicopters pour la préparation de missions et ensuite d'étudier les techniques de planification de chemins aériens proposées par d'autres industriels du domaine aéronautique tels que Lockheed Martin, Honeywell et Boeing.

2.2.1 Les pratiques actuelles au sein d'Airbus Helicopters

Dans notre étude de l'existant, nous nous sommes intéressés au début aux outils de préparation de missions au sein d'Airbus Helicopters en considérant le cas militaire et civil. Aujourd'hui, ces outils ne permettent pas la génération automatique d'un plan de vol 3D d'où la nécessité d'une intervention du pilote. Certaines missions sont très complexes à traiter telle que une opération de type RESCO (REcherche et Sauvetage de COMbat) où l'appareil doit s'infiltrer dans un territoire hostile pour récupérer des hommes au sol. En effet, le pilote dispose d'un outil informatique (station au sol) qui lui propose une carte avec un modèle numérique du terrain. Le pilote peut donc visualiser les zones à éviter avant de définir le plan de vol horizontal. Une fois le plan de vol horizontal élaboré, la préparation de mission génère automatiquement un profil vertical constitué de phases de vol en palier avec des raccordements entre les paliers. Le profil généré peut être ajusté par le pilote. Par la suite, en se basant sur un modèle de prédiction des performances, l'outil vérifie la faisabilité et les caractéristiques de la mission (masse, carburant et temps de vol). Après plusieurs heures, la préparation de mission génère un fichier de mission qui sera chargé dans le FMS (*Flight Management System*) afin de guider l'hélicoptère en vol 2D. C'est le pilote qui gère manuellement le profil vertical du plan de vol préparé.

Dans le cas des missions civiles (secours en montagne par exemple), il n'existe aucun outil de préparation de mission en 3D. La faisabilité d'une mission est évaluée grâce à l'expertise des pilotes (connaissance de l'appareil, de la région, expérience des missions précédentes). Le pilote dispose à bord d'une cartographie numérique sur laquelle il peut indiquer des coordonnées géographiques ou une adresse vers laquelle il va voler avec une route plus ou moins directe suivant les conditions climatiques et le relief du terrain. Un routeur automatique 3D intégrant les performances de l'appareil, le modèle du terrain et les conditions atmosphériques peut aider le pilote.

2.2.2 Étude de quelques brevets du domaine

Lors d'une étude d'antériorité, nous avons examiné des brevets qui traitent la planification de chemins pour des aéronefs qui sont déposés par d'autres industriels du domaine

aéronautique :

US 4812990 A : System and method for optimizing aircraft flight path (MERIT Tech Inc, 1989) : Le système décrit dans ce brevet permet de générer un plan de vol optimal en 2D dans un environnement dynamique entre deux positions données de l'espace. En effet, le système de planification discrétise le rectangle délimité par les deux points en grille régulière puis applique une solution de la programmation dynamique. Cette méthode calcule le chemin qui optimise un coût résultant d'une combinaison entre « terrain cost », « threat cost », « cost of deviation from center line » et « path length ». À chaque itération, l'algorithme met à jour les coûts des différents nœuds. La méthode appliquée est optimale et donc requiert un temps de calcul important.

Le système proposé est adapté aux avions de ligne. Il repose sur une méthode optimale qui demande un temps de calcul important pour les terrains étendus. La solution s'appuie sur la construction de la grille 2D ainsi que le calcul des coûts des arcs avant d'appliquer une méthode de programmation dynamique pour trouver un plan de vol. Ceci empêche l'application de la solution dans des situations critiques nécessitant un résultat dans une durée courte. Le fait de discrétiser l'espace en 2D seulement dans la zone rectangulaire délimitée par le point de départ et le point d'arrivée peut éliminer des solutions intéressantes : par exemple dans certaines situations, il est plus intéressant de contourner des obstacles plutôt que de suivre un plan de vol direct avec des fortes variations d'altitude.

US 6266610 B1 : Multi-dimensional route optimizer (Honeywell International Inc, 2001) : Le système permet de calculer la trajectoire d'un aéronef selon ses caractéristiques d'exploitation (temps, carburant, évitement de régions de conflits politiques, distance de vol, etc.) et des phénomènes relatifs à l'environnement dynamique 3D. L'optimisation comprend deux composantes : une composante latérale (qui minimise une fonction coût) et une composante verticale qui repose sur le calcul des variations et qui dépend des paramètres de l'appareil et de la vitesse du vent. De plus, l'optimisation verticale est mise à jour à chaque itération de l'optimisation latérale. Cette dernière repose sur une solution de la programmation dynamique appliquée sur une grille régulière entre la position de départ et la position d'arrivée.

Bien que le système décrit permet le calcul d'un plan de vol en optimisant un ou plusieurs critères dans un environnement 3D, il présente des limitations. L'application d'un algorithme de programmation dynamique garantit l'optimalité par contre la complexité en temps de calcul sera plus importante. Le découplage entre l'optimisation latérale et verticale agit sur la qualité de la solution et nécessite un temps de calcul supplémentaire. Ceci ne répond pas aux exigences liées à la replanification dynamique.

US 006085147 A : System for determination of optimal travel path in a multi-dimensional space (University Corp for Atmospheric Research, 2000) : Le système décrit permet de déterminer un plan de vol en 3D entre une position de départ et une position

d'arrivée dans un environnement dynamique en appliquant un algorithme de programmation dynamique sur un graphe 3D. La solution permet de minimiser une fonction coût qui combine entre le temps de vol, le carburant consommé et les dangers de la zone. Chacun des critères d'optimisation dépend des paramètres liés aux performances de l'appareil et aux variations de l'environnement durant le vol. Pour la mise à jour, des capteurs sont mis en place afin de communiquer toute information à l'origine d'un changement. L'espace 3D entre les points de départ et d'arrivée est discrétisé avec une grille régulière 3D. Par la suite, un graphe orienté allant de la position de départ vers la position d'arrivée est construit. À chaque arc est affecté un coût variable en fonction du temps. Enfin, un algorithme de programmation dynamique est appliqué sur le graphe afin de déduire le plus court chemin.

Le graphe 3D est construit à partir d'un maillage régulier, ce qui consomme énormément de mémoire pour des terrains étendus avec une fine précision. D'ailleurs, le graphe est composé d'un ensemble d'arcs permettant de relier uniquement deux nœuds voisins. Ceci influence la qualité de la solution pour les chemins de grande distance comme il n'est pas possible de créer des arcs directs entre deux nœuds éloignés. L'algorithme de recherche de plus court chemin est optimal : il s'agit d'une solution basée sur la programmation dynamique. Son application sur des graphes de grande taille implique une complexité de calcul importante incohérente avec l'environnement dynamique du système décrit. Pour planifier le vol, le système considère uniquement l'espace situé entre les positions de départ et d'arrivée. Pour certains types d'appareil, il est possible de revenir en arrière avant de trouver le plus court chemin. Ainsi, le fait d'appliquer une telle discrétisation, rend la réutilisation du graphe calculé impossible pour de futurs scénarios.

US20140142787 A1 : Determination of Flight Path for Unmanned Aircraft in Event of In-Flight Contingency (Boeing Co, 2014) :

Le brevet décrit un système de planification de vol pour des UAVs (*Unmanned Aerial Vehicle*) dans une perspective de maintenir la continuité des communications avec l'opérateur pendant la mission. Cette fonctionnalité est très critique surtout pour des missions de surveillance dans des régions montagneuses où la perte de communication peut causer l'échec de la mission et le crash de l'appareil. Pour cette raison, le procédé de planification est guidé par la position des antennes de communication. Lors d'une perturbation, le système va essayer de joindre le point de passage qui se trouve dans la zone de couverture la plus proche. Pour ce faire, il s'appuie sur les points de passage ainsi que les zones d'atterrissage qui sont connus en avance. Il s'agit d'une replanification à court terme où le système se limite à trouver le prochain point de passage et étudier la faisabilité de la trajectoire au niveau opérationnel. Dans ce brevet, les auteurs n'abordent pas le problème de complexité liée à la taille du terrain.

► **Synthèse :** Dans les travaux mentionnés dans les brevets ci-dessus, les méthodes proposées ne permettent pas de maîtriser la complexité et la qualité de la solution en fonction des paramètres liés à l'appareil, la mission et l'environnement. Ces méthodes assurent une

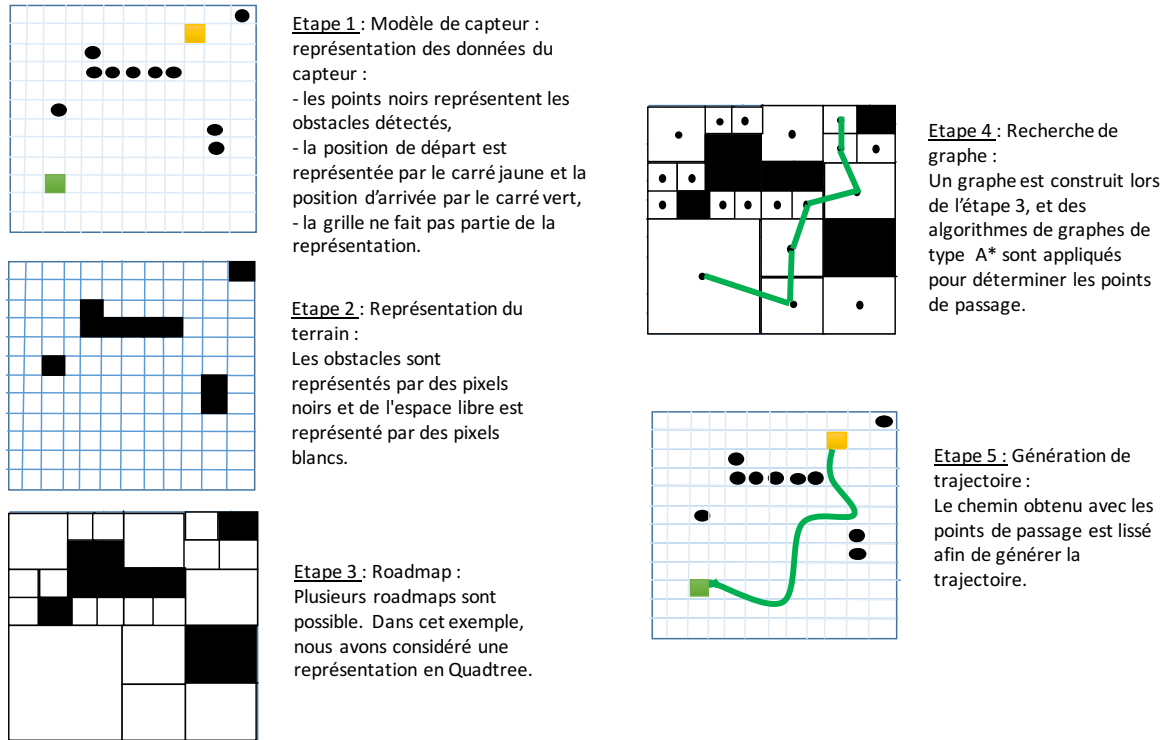


FIGURE 2.1 – Étapes de planification : de l'acquisition des données à la planification de la trajectoire

discretisation partielle de l'espace 3D ce qui impacte la qualité de la solution trouvée. Nous envisageons une solution générique et adaptative qui permet d'obtenir un profil de vol selon le type de la mission dans des délais bornés selon le temps disponible pour préparer la mission.

2.3 Techniques de planification de chemins aériens

Cette section introduit le problème de planification de vol d'une manière générale. Tout d'abord, nous décrivons les différentes étapes de planification. Par la suite, nous évoquons les solutions les plus appliquées ainsi que leurs avantages et inconvénients. Pour conclure, nous offrons une synthèse et un positionnement de nos travaux par rapport à l'état de l'art en optant pour des approches pouvant répondre aux exigences de notre contexte de recherche. Afin de décrire les différentes étapes de planification, nous modélisons l'aéronef comme un point véhicule. Étant donné que les aéronefs ne doivent pas entrer dans des espaces restreints en vol, la simplification de la délimitation de l'appareil par un solide symétrique en rotation a peu d'effet sur la trajectoire [21]. Dans de nombreux problèmes, les contraintes différentielles sont indispensables car elles permettent la prise en compte explicite de certains critères tels que le temps ou l'énergie. Ceci conduit à la génération de trajectoires faisables. La Fig. 2.1

montre les étapes de planification de mouvement à plusieurs stades et illustre l'interaction entre le traitement des données du capteur et la planification du mouvement. De nombreux algorithmes peuvent combiner deux ou plusieurs de ces étapes. Dans notre recherche, nous nous intéressons aux étapes 2, 3 et 4. Ces dernières s'intéressent principalement à la modélisation de l'espace si elle est nécessaire et à la planification du chemin aérien.

2.3.1 Critères de performances des algorithmes

Dans le domaine de planification de chemins, plusieurs classifications permettent de distinguer la nature des méthodes utilisées. Parmi ces classifications, nous trouvons celles qui s'appuient sur une modélisation de l'espace ou non, le déterminisme ou l'optimalité de la solution, celles qui sont en ligne ou hors ligne, etc. À titre d'exemple, pour des algorithmes comme RRT (*Rapidly exploring Random Trees*) [35] ou l'algorithme de champs de potentiel [32] la modélisation de l'espace n'est pas nécessaire tandis que des algorithmes comme Dijkstra [28] et A star (A*) [22] nécessitent une phase de modélisation de l'environnement afin de trouver une solution faisable ou optimale. Ces deux derniers algorithmes offrent des méthodes déterministes lorsqu'ils sont appliqués sur des grilles ou un graphe de visibilité. Toutefois dans certaines applications, ils ne répondent pas à des exigences de temps réel. Des méthodes comme l'algorithme de colonies de fourmis [8] ou RRT offrent des performances meilleures en termes de temps d'exécution mais leur nature probabiliste empêche leur utilisation dans des domaines critiques. En guise de conclusion, que le choix d'une méthode de planification dépend fortement du domaine d'application et des propriétés à satisfaire par la solution générée. Dans notre étude nous nous sommes intéressés à un certain nombre de propriétés en liaison avec le contexte industriel de notre thèse à savoir :

- la génération d'un chemin optimal ou sub-optimal avec un coût réduit [52],
- la maîtrise de la complexité temporelle et spatiale,
- la déterminisme de l'approche : une solution doit être trouvée en cas d'absence d'obstacles ou de contrainte serrée qui empêche la génération d'un chemin,
- la générique en fonction du type de l'appareil, la mission, l'environnement, etc.

La Fig. 2.2 résume notre classification des méthodes de planification de mouvement les plus utilisées dans différents domaines tels que la robotique, l'aéronautique, les jeux vidéo, etc. Nous notifions que cette classification ne couvre pas toutes les méthodes vu l'ampleur du domaine. Dans les sections qui suivent, nous donnons tout d'abord un aperçu sur les méthodes qui ne nécessitent pas une discrétisation de l'espace, ensuite nous introduisons la deuxième catégorie de méthodes en mettant l'accent sur les techniques de modélisation de l'environnement ainsi que les algorithmes de calcul du plus court chemin correspondants.

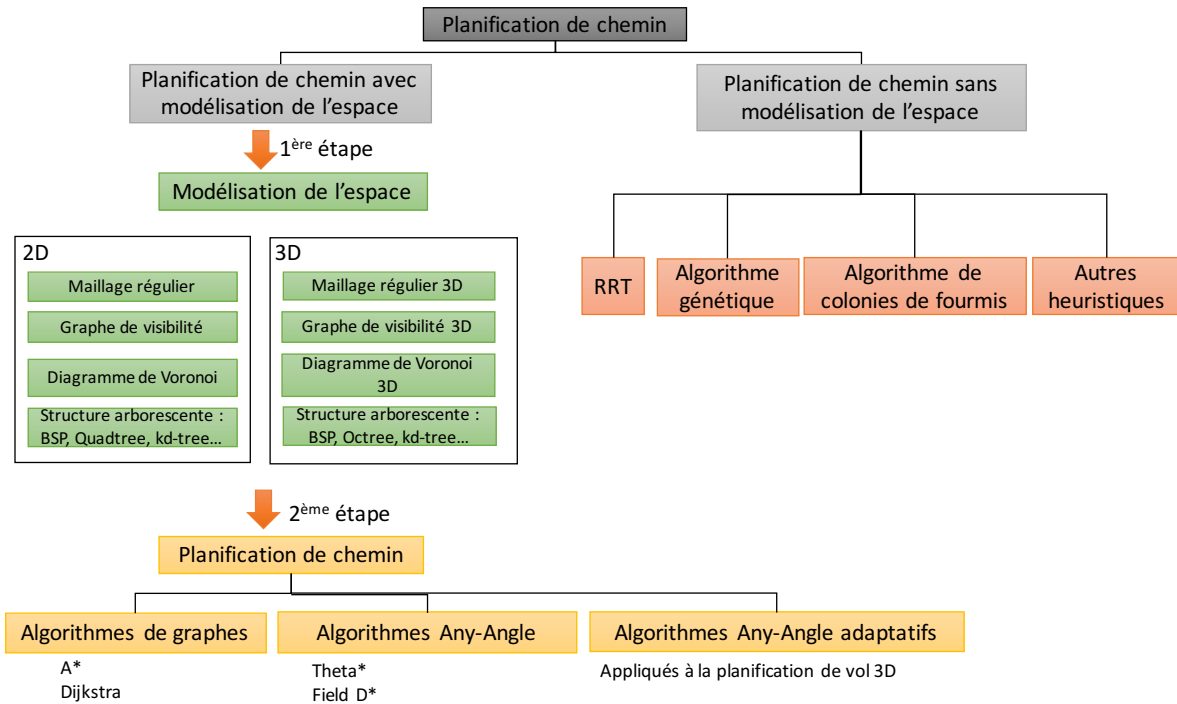


FIGURE 2.2 – Classification des méthodes de planification de mouvement les plus utilisées

2.3.2 Première catégorie de méthodes : modélisation de l'espace non nécessaire

Cette catégorie couvre des méthodes non-déterministes (solution ou temps d'exécution non garanti(e)). Comme exemples de méthodes appliquées à la planification de mouvement, nous citons l'algorithme de colonies de fourmis [8], l'algorithme génétique [58]. Certains travaux de recherche [57][53] ont appliqué ce type de méthodes pour la planification des routes pour des drones. Dans la suite, nous citons les méthodes les plus connues de cette catégorie :

Probabilistic roadmap : ou « Probabilistic Road Mapping » [31][5][30][53] génèrent plusieurs configurations aléatoires, élaguent celles qui sont bloquées et lient les libres adjacentes dans un graphe. La planification de chemin est effectuée en reliant les configurations initiale et cible dans le graphe, puis en cherchant un chemin entre elles. Cette approche présente des solutions à des problèmes de complexité arbitraire. Elle est généralement utilisée dans les problèmes de manipulateur, où l'espace de configuration est souvent de grande dimension et complexe. Cependant, le temps de convergence est lent et les chemins qu'elle produit ne sont pas optimaux. Dans le cas d'une planification de chemin 2D ou 3D, cette méthode a la particularité de générer des routes avec un nombre élevé de points de passage ce qui rend l'étape de lissage du chemin indispensable avant d'aborder la phase de génération de trajectoire.

RRT : Cette heuristique [35][37][36][19][46] fonctionne en appliquant une recherche sto-

chastique sur l'image centrée sur l'objet et en développant un arbre à travers un échantillonnage aléatoire de l'espace de configuration. Cet algorithme s'est avéré complet dans le sens probabiliste et produit des chemins de bonne qualité avec des durées de calcul réduites. Cependant, il n'y a aucune preuve du taux de convergence ou de l'optimalité. De plus, cette heuristique n'arrête pas la recherche en cas d'impossibilité de trouver une solution.

Algorithme de champs de potentiel : Parmi les approches d'évitement d'obstacles en temps réel, nous évoquons un algorithme utilisant le concept des champs de potentiel pour permettre la planification de chemin pour des véhicules en temps réel dans un environnement complexe. La méthode des champs de potentiel a été initialement proposée par O. Khatib [32]. Cet algorithme est basé sur l'idée d'accorder une fonction potentielle à l'espace libre, et de simuler le véhicule comme une particule réagissant aux forces dues au champ potentiel. Le point de destination a le plus faible potentiel et donc attire le véhicule, tandis que les obstacles le repoussent. Une variante de l'algorithme de champs de potentiel commence par essayer un chemin dans l'espace de configuration et applique des champs de potentiel répulsifs autour des obstacles pour modifier le chemin afin qu'il se déplace vers l'espace libre jusqu'à la convergence vers un chemin sûr. L'approche des champs de potentiel fournit une technique simple pour la planification de chemin, facile à mettre en œuvre et adaptée à la navigation avec des exigences en temps réel vu sa faible complexité de calcul. Elle est toujours considérée comme l'une des meilleures méthodes pour la planification de chemin en ligne, voir [68], [42] et [38] dans lesquels certaines variantes de cette approche ont été publiées au cours de la dernière décennie. Cependant, elle représente des inconvénients principalement les minimas locaux : le potentiel total qui permet de guider le véhicule est la somme d'un potentiel attractif et répulsif de signes opposés. Ainsi, pour certains obstacles dans l'environnement étudié, cette fonction peut avoir un nombre de minimas locaux dans lequel le véhicule sera piégé. Par conséquent, l'approche du champ de potentiel peut être très limitée dans les environnements encombrés. La méthode est non déterministe ce qui rend son application difficile pour la navigation autonome. Pour pallier à ces limitations, plusieurs travaux de recherche avaient pour but d'éliminer les minima locaux (par exemple Hussein et al. [25]). D'autres travaux se sont concentrés sur la conception d'un planificateur de chemin complet avec un champ de potentiel ayant les propriétés d'une fonction de navigation [47].

► **Synthèse :** Bien que ces méthodes offrent des solutions de bonne qualité avec des temps d'exécution réduits, leur utilisation dans notre contexte s'avère inadéquate vu qu'elles ne garantissent ni des solutions déterministes ni des temps d'exécution bornés. Cet aspect est crucial surtout pour le cas de replanification dynamique en cas d'urgence.

2.3.3 Deuxième catégorie de méthodes : avec modélisation de l'espace

La deuxième catégorie offre des méthodes déterministes. Ces méthodes comprennent deux types d'approches : les approches optimales et suboptimales. Elles sont généralement composées de deux phases : la phase de prétraitement ou de modélisation de l'espace de configuration et la phase de routage qui consiste à parcourir la structure de données générée lors de la première phase et calculer le plus court chemin (voir Fig. 2.2).

Un premier type de planification de chemin repose sur une modélisation complète de l'espace de solutions et offre des chemins optimaux. Parmi les méthodes associées à ce type, nous trouvons le diagramme de Voronoï [16] et le graphe de visibilité [41]. Malgré leur optimalité, ces approches présentent un inconvénient majeur lors de la planification de chemin de longue distance. En effet, la complexité spatiale et temporelle engendrée empêche la résolution du problème dans des situations qui nécessitent une réponse en temps réel.

Le deuxième type de planification de route permet d'atténuer cette limitation. Il regroupe les méthodes qui discrétisent l'espace selon un maillage approximatif telles que les grilles régulières ou irrégulières comme les Quadrees [29]. La résolution du problème de planification de chemin à travers ces méthodes est couplée avec des algorithmes qui consistent à parcourir un graphe afin de trouver le plus court chemin qui évite les obstacles. À titre d'exemple, nous citons l'algorithme A* [22], Theta* [15], etc.

Nous évoquons quelques travaux qui ont traité des solutions déterministes suboptimales pour résoudre des problèmes de planification de mouvement. Les travaux de recherche décrits dans [29] consistent à une approche hiérarchique basée sur l'algorithme A* qui explore une grille de type Quadtree (voir Section 2.4.2). Cette méthode permet de réduire le temps de prétraitement et de routage. L'idée consiste à appliquer un « A* hiérarchique ». Pour ce faire, des « Pruned Quadtree » sont générés. Ces Quadrees présentent trois types de nœuds : libres, obstacles et gris. Les nœuds « gris » représentent des régions contenant des obstacles. Ils sont obtenus en arrêtant le découpage à un seuil donné. D'où la réduction de la durée de prétraitement. Par la suite, l'algorithme A* est appliqué pour trouver le chemin approximatif (nœuds « libres » et « gris ») entre une position de départ et une position d'arrivée. Finalement, le path est raffiné en trouvant les chemins dans/entre les nœuds « gris ». Sur base de la littérature du domaine, ce « A* hiérarchique » est de 3 à 10 fois plus rapide que l'algorithme A* appliqué sur un simple Quadtree. Cependant, la complexité de cette approche ne peut pas être estimée dans un contexte 3D. Dans le cas d'une planification de chemin dynamique, les nœuds « gris » représentent une manière élégante pour décrire les zones inexplorées.

Un autre exemple représente un planificateur dynamique pour des missions de drones de petite taille [67]. Il intègre la totalité des étapes nécessaires dès l'acquisition des données terrain jusqu'à la génération de la trajectoire 3D tout en respectant les limitations dynamiques du véhicule, les obstacles, etc. Toutefois, ce système est spécifique aux missions à court terme c'est-à-dire qu'il planifie des routes de l'ordre de quelques dizaines de mètres. Il compare deux

approches différentes. La première couple une discrétisation du terrain en Qaudtree sur des tranches planes avec une méthode permettant de construire, d'évaluer plusieurs chemins 3D sur base de cette discrétisation et de sélectionner celui ayant le moindre coût en appliquant la logique floue. La limitation fondamentale de ce planificateur provient de l'utilisation itérative d'une méthode 2D pour construire des routes 3D ce qui alourdit les calculs. La deuxième approche s'appuie sur l'algorithme de recherche A* appliqué sur la matrice d'altitude 3D. Il planifie directement le chemin en 3D en étant trois fois plus rapide que le premier. L'utilisation de ces structures de données engendre de lourdes conséquences sur la complexité du calcul et la qualité de la solution lors de l'application de l'approche sur des terrains de grandes dimensions.

► **Synthèse :** Étant donné les exigences imposées par Airbus Helicopters pour des raisons de sécurité, nous nous focalisons sur les méthodes déterministes. Pour ces méthodes, une modélisation de l'espace s'avère indispensable. Ainsi, la planification de chemin est composée de deux phases : la modélisation de l'espace et la recherche du plus court chemin qui seront étudiées dans les sections qui suivent.

2.4 Discrétisation de l'environnement

La modélisation de l'espace représente une étape cruciale pour certaines méthodes. La planification de vol nécessite l'étude des structures de données permettant différents types de maillage en 2D et 3D. Une description des méthodes de discrétisation 2D permettra une meilleure compréhension du contexte de la recherche et des structures de données 3D.

La modélisation de l'espace compte deux catégories, la première concerne les méthodes *Cell decomposition* tels que les grilles régulières, irrégulières, etc. Les nœuds associés offrent une représentation de la topologie de l'espace libre. Les méthodes de la deuxième catégorie *Roadmap decomposition* appliquent un pré-calcul de chemins entre des points répartis dans l'environnement modélisé. Parmi ces méthodes, nous citons les graphes de visibilité et les diagrammes de Voronoï.

2.4.1 Roadmap decomposition

Graphe de Visibilité : Ce graphe [41] représente une solution exacte pour le problème de planification de chemin. Admettant que le plus court chemin touche les obstacles polygonaux à leurs sommets, cette méthode construit une *roadmap* qui relie chaque sommet avec tous les sommets visibles depuis sa position (voir Fig. 2.3 (a)). Un graphe construit a normalement une position de départ, une position d'arrivée, l'ensemble des sommets, et l'ensemble des arcs reliant chaque paire de sommets visibles. Certains travaux combinent le graphe de visibilité et l'algorithme A* pour sélectionner un chemin optimal comme le système de planification

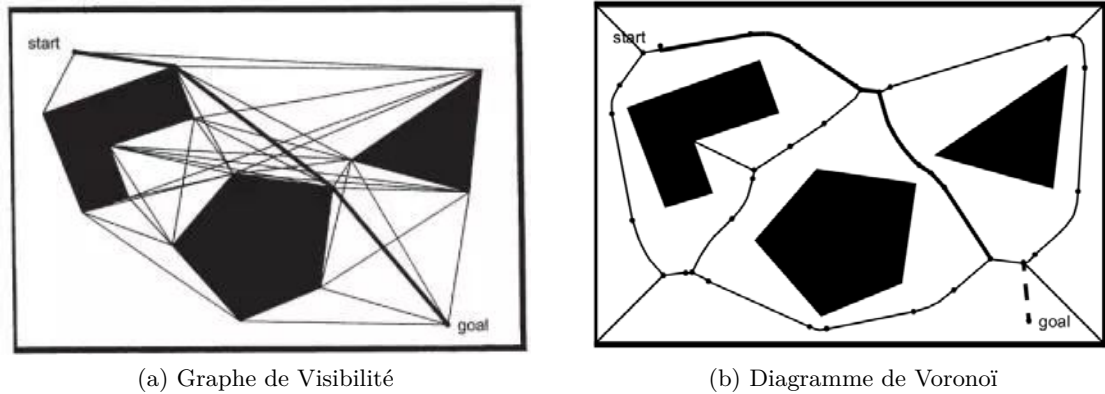


FIGURE 2.3 – Planification de chemins en utilisant le graphe de visibilité et le digramme de Voronoï [56]

de chemin décrit dans [2]. Le chemin le plus court de la position de départ à la position de destination est déterminé par l'algorithme A^* . Cette solution ayant une complexité de $O(n^2)$ est complète et optimale. Par contre, dans un espace 3D, une solution exacte s'avère NP-Hard [10]. Étant donné qu'un chemin calculé est arbitrairement proche des obstacles à plusieurs reprises, cette approche n'offre pas de marge de sécurité pour empêcher les collisions dans le cas des systèmes dont l'emplacement est incertain. Il existe différentes variantes du graphe de visibilité comme le « Tangent Graph » qui est décrit dans [40].

Certaines méthodes de maillage 2D ont été généralisées dans le but de contribuer à la modélisation de l'espace 3D. Parmi lesquelles, nous trouvons les graphes de visibilité décrits dans [41][2] [10]. Un travail récent [45] implémente un planificateur de vol pour des micro-drones en appliquant le graphe visibilité et l'algorithme A^* . Ce système assure des vols de courtes distances (de l'ordre de kilomètres) ce qui explique la maîtrise de la complexité avec la structure de données utilisée.

Diagramme de Voronoï [16] : C'est une structure de données spatiale construite en sélectionnant les nœuds du graphe (voir Fig. 2.3 (a)). L'inconvénient de cette méthode est la complexité engendrée lors de son application sur des espaces de tailles importantes. La réduction de la dimension de l'ensemble à chercher réduit généralement la complexité temporelle de la recherche. L'un des avantages de ce diagramme est qu'il conduit à des chemins robustes. Takahashi et Schilling [66] planifient un robot rectangulaire avec des obstacles polygonaux à l'aide d'une heuristique appliquée au diagramme de Voronoï. D'autres travaux [14] utilisent le diagramme de Voronoï dans la planification 3D. L'application de cette structure est restreinte aux espaces de configuration de taille réduite vu la complexité spatiale et temporelle qu'elle engendre.

2.4.2 Cell decomposition

Malgré les solutions optimales garanties par les méthodes de modélisation *Roadmap decomposition*, leur complexité algorithmique limite leur exploitation dans le domaine aéronautique, en particulier elles ne sont pas adaptées pour offrir différents profils de vol selon le type de la mission. Pour cela, les recherches se sont orientées vers les approches *Cell decomposition*. Les grilles régulières sont applicables dans plusieurs domaines comme la robotique et les jeux vidéo vu la facilité de leur mise en œuvre et la simplicité de leur mise à jour. Cependant, une résolution relativement élevée accroît la consommation mémoire qui représente un obstacle lors de la modélisation des terrains étendus. Pour pallier à ces limitations, des travaux de recherche ont contribué au développement des méthodes de discrétisation irrégulières [51][49]. Ces méthodes offrent un maillage adaptatif selon la densité des obstacles dans l'espace et donc une réduction de la consommation mémoire. Elles permettent une représentation hiérarchique de l'espace de configuration ce qui conduit à la simplicité des opérations algorithmiques qui lui sont associées. Dans cette section, nous décrivons les structures de discrétisation de l'espace les plus utilisées.

Grille régulière : Elle [73][67] divise l'espace de configuration en régions rectangulaires en les marquant comme « obstacle » (noir) ou « libre » (blanc). Cette méthode est déterministe et complète en résolution car elle garantit de trouver une solution si elle existe et d'indiquer qu'il n'y a pas de solution dans le cas contraire. L'exemple le plus courant est celui de l'application des algorithmes de type A^* [22] ou D^* [60] sur une grille régulière 2D et 3D. Dans [18], les auteurs implémentent ce type de maillage en étudiant les algorithmes de planification de chemin statiques et dynamiques. Les travaux [71][70] appliquent l'algorithme A^* sur une grille régulière en 3D. Malgré les durées de calcul acceptables, la qualité des solutions générées est très limitée, en plus de la complexité en termes de mémoire, ce qui complique son application dans le domaine aéronautique surtout pour des terrains très étendus. Afin de réduire la complexité des grilles régulières, d'autres méthodes ont été proposées afin d'assurer la modélisation de l'espace 3D comme la « Hiérarchie du Volume Englobant ».

Bounding Volume Hierarchy (BVH) : ou bien la hiérarchie des volumes englobants est une structure arborescente d'objets géométriques. Ces objets sont enveloppés dans des volumes englobants qui forment les feuilles de l'arbre. Ces nœuds sont ensuite regroupés en sous-ensembles et inclus dans des volumes englobants plus grands [69]. Ils sont utilisés dans la détection d'obstacles ou le lancer de rayons (calcul optique pour la génération d'images, algorithme de peintre, etc.). La complexité logarithmique en fonction du nombre des objets constitue le point fort de la structure. Étant donné leur efficacité, les BVH offrent une bonne représentation des obstacles pour la replanification dynamique.

Arbre K-D (1975) : La structure des arbres K-D a été proposée en 1975 pour la recherche multidimensionnelle (voir Fig. 2.4). Un arbre K-D (ou K-D tree, pour K-Dimensional tree) [4] est une structure de données de partitionnement de l'espace permettant de stocker des points

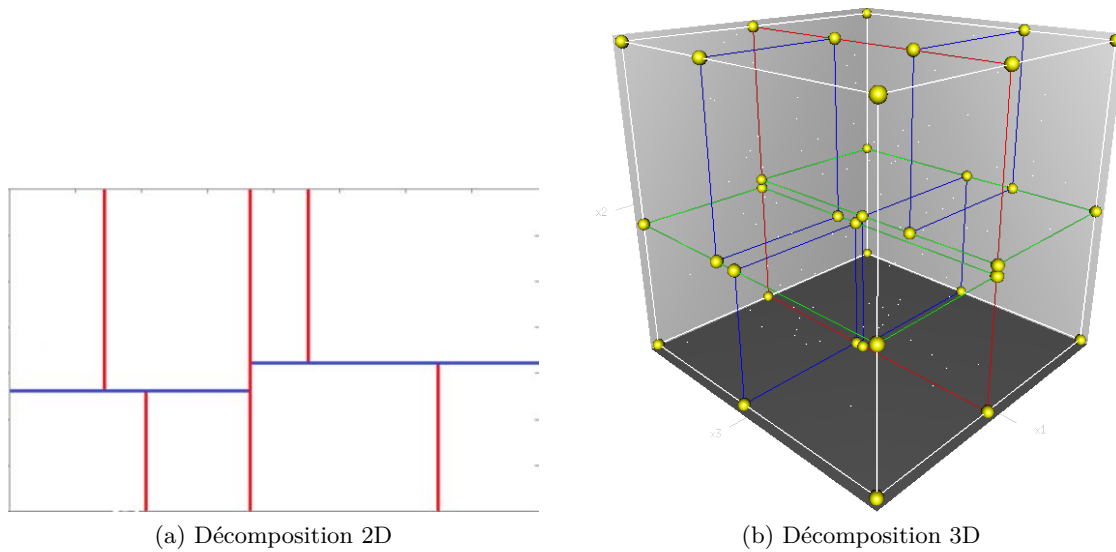


FIGURE 2.4 – Représentation de l'espace en arbre K-D

et de faire des recherches (recherche par plage, plus proche voisin, etc.) plus rapidement qu'en parcourant linéairement le tableau de points. Les arbres K-D sont des cas particuliers des arbres BSP (Binary Space Partition trees).

Partition Binaire de l'Espace (BSP, 1980) : La partition binaire de l'espace (Binary Space Partitioning Tree) [1] représente l'extension de l'arbre K-D à la représentation d'objets 3D en partitionnant l'espace de manière récursive. Un traitement de division de l'espace de configuration est appliqué de manière récursive jusqu'à ce qu'un critère d'arrêt soit atteint. Le BSP est un système utilisé pour diviser l'espace en zones convexes. Les nœuds sont délimités par des hyperplans. Dans le cas d'un espace à deux dimensions (plan), les séparations sont des droites et les nœuds sont des quadrilatères (souvent des rectangles). Dans le cas d'un espace à trois dimensions, les séparations sont des plans et les cellules sont des polyèdres (souvent des parallélépipèdes rectangles). Les nœuds sont disposés en arbre binaire appelé arbre BSP. L'arbre K-D (1975) est un cas particulier du BSP. Il utilise uniquement des plans de division perpendiculaires à l'axe des coordonnées. La Fig. 2.5 montre la différence entre une représentation de l'espace en Quadtree, Arbre K-D et BSP. Le BSP est intéressant pour le rendu 3D et il est utilisé dans plusieurs applications telles que la gestion des graphismes de certains jeux vidéo, l'infographie, l'enlèvement de surface cachée, le *Ray tracing*, la planification de mouvement pour des robot, etc. Cependant la structure BSP présente des inconvénients qui sont à l'origine de plusieurs bugs comme les nombreux plans quasi-confondus, etc. La complexité de l'arbre BSP dépend également de l'application. Elle peut être $O(n^2)$ et $O(n)$ pour le cas moyen, où n est le nombre des obstacles polygonaux présents dans l'espace de configuration.

Quadtree (1974) : C'est une structure de données arborescente [20][48][50][54][49][51] dans laquelle chaque nœud a quatre fils (voir Fig. 2.6). Cette structure introduite en 1974 a été

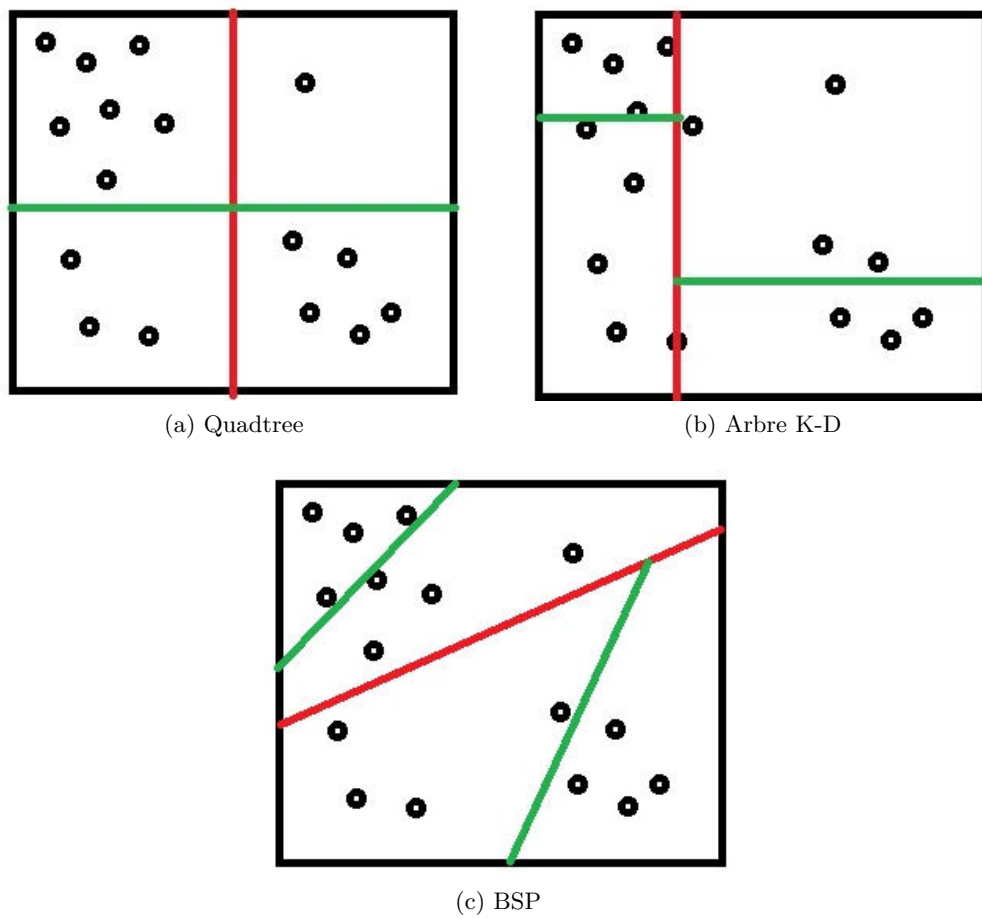


FIGURE 2.5 – Exemples de représentations arborescentes de l'espace

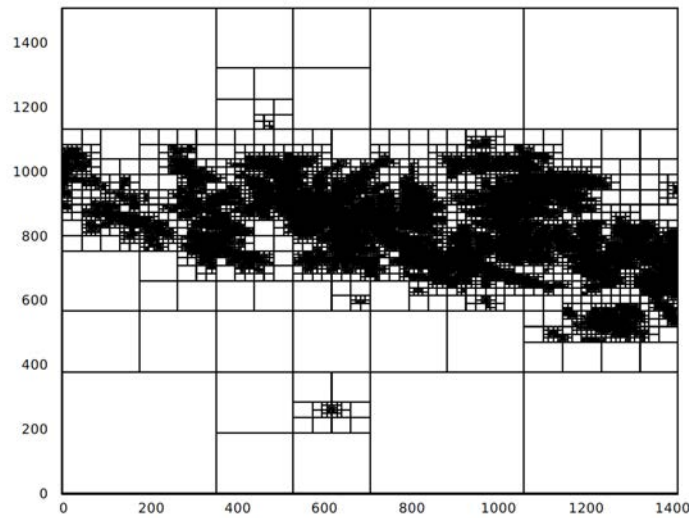


FIGURE 2.6 – Représentation de l'espace en Quadtree

souvent utilisée pour partitionner un espace bidimensionnel en le subdivisant récursivement en quatre nœuds. Chaque nœud d'un Quadtree divise l'espace qu'il représente en quatre sous-espaces. Les Quadtrees ont été largement utilisés dans la planification de mouvement en robotique [29]. Leur utilisation a été étendue pour planifier des missions navales pour des robots sous-marin autonomes (*Autonomous Underwater Vehicle*) [11] et des drones de surface (*Unmanned Surface Vehicle*) [55]. Dans l'aéronautique, les Quadtrees ont été appliqués afin de planifier des missions urbaines pour des drones [67].

Framed Quadtree [13] : C'est une structure de données inspirée des Quadtrees [50] [49]. Il a été conçu dans le but de combiner la précision de la grille régulière et l'efficacité du maillage adaptatif (voir Fig. 2.7 (b)). La représentation d'un *Framed Quadtree* est similaire à un Quadtree classique avec des nœuds supplémentaires sur la frontière de chacun des nœuds libres. Ces nœuds ont la résolution maximale du maillage. Le nombre des nœuds supplémentaires ajoutés sur la frontière d'un nœud est égal à $(\text{taille de la frontière du nœud} \times 4 - 4)$ où cette taille est évaluée par un nombre entier multiple de la résolution de la carte.

Chen et al. [13] introduisent le *Framed Quadtree* en prouvant que le temps lié à la navigation d'un algorithme de planification de chemin à travers cette structure est linéaire avec des garanties d'obtention de chemins indépendamment de la nature de l'environnement 2D étudié. En termes d'optimalité, ils montrent que le *Framed Quadtree* dépasse le Quadtree [13]. Un exemple de comparaison entre la qualité du chemin généré par un Quadtree et un *Framed Quadtree* est donné par la Fig. 2.7. Nous pouvons remarquer l'impact des nœuds supplémentaires sur la frontière de chacun des nœuds libres sur la qualité de la solution générée avec un surcoût en termes de temps d'exécution et de mémoire. Par ailleurs, Yahja et al. [72] montrent que la discrétisation de l'espace avec le *Framed Quadtree* est plus appropriée aux

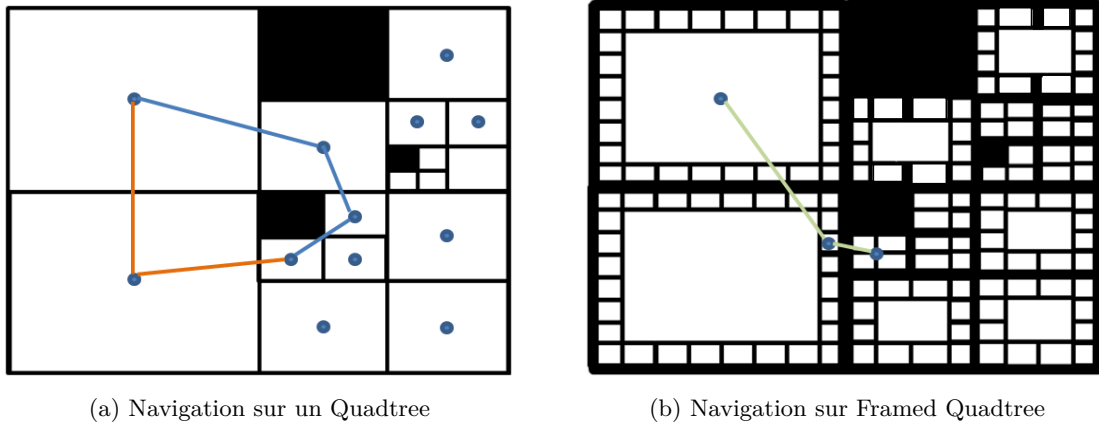


FIGURE 2.7 – Comparaison de performances entre Quadtree et Framed Quadtree [59]

environnements qui présentent une faible densité d'obstacles. Cet article présente des résultats qui prouvent que pour les régions denses en obstacles, les durées de calcul et la consommation mémoire augmentent. En effet, pour ce type de région, les grilles régulières offrent de meilleures performances que les *Framed Quadtree*.

Octree (1988) : Un Octree [43][27] est une décomposition récursive d'un espace tridimensionnel en octants comme le montre Fig. 2.8. Il est l'analogie tridimensionnelle du Quadtree. Dans un premier temps, tout l'espace est représenté par un nœud unique dans l'arbre, appelé nœud racine. Si le volume cubique est homogène (complètement rempli d'obstacles ou complètement vide), alors la racine n'est pas décomposée et comprend la description complète de l'espace. Dans le cas contraire, il est divisé en huit octants de tailles égales, qui deviennent les enfants de la racine. Ce processus se poursuit jusqu'à ce que tous les nœuds soient homogènes ou jusqu'à ce qu'une limite de résolution soit atteinte. Les nœuds correspondant à des octants complètement pleins sont appelés nœuds feuilles « Obstacle ». Les nœuds correspondant aux régions vides sont appelés nœuds feuilles « Libre ». L'Octree offre une représentation indexée de l'espace 3D. Ceci mène à des algorithmes de détection d'obstacles très rapides. Pour trouver un chemin, cette structure fournit une décomposition de l'espace libre en cubes, dont chacun peut être considéré comme un nœud dans un graphe de recherche. Certaines méthodes permettent de déduire le voisinage d'un nœud et donc déterminer les arcs du graphe. La nature hiérarchique et multirésolution des Octrees peut être utilisée pour améliorer la vitesse des algorithmes de recherche [3]. Ainsi, une représentation de l'espace en Octree [51][49] mène à une planification de chemin rapide [23]. Toutefois, l'Octree présente des limitations lors de la planification des chemins aériens. En effet, sa structure géométrique ne permet pas de répondre aux contraintes avioniques dans le cas de missions spécifiques. La résolution de la structure mène à un maillage grossier de l'environnement 3D ce qui mène à des problèmes de précision lors de l'élaboration d'un plan de vol.

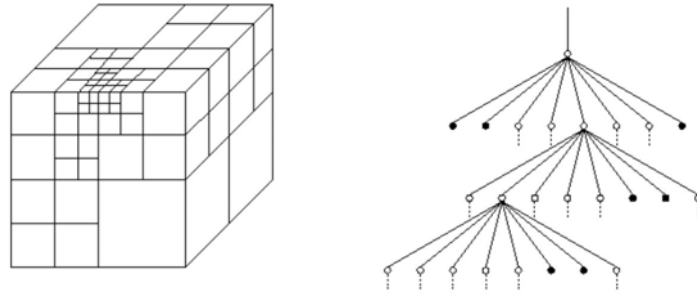


FIGURE 2.8 – Représentation de l'espace 3D en Octree

► **Synthèse :** Nous avons étudié différentes structures de discrétisation de l'espace 3D afin de trouver la solution la plus appropriée à notre contexte industriel et celle qui offre une complexité maîtrisée en temps et en mémoire. Certaines méthodes seront par la suite implémentées et testées sur des terrains réels. L'application des algorithmes de calcul du plus court chemin sur ces structures nous permettra d'évaluer la qualité des solutions générées en termes de temps d'exécution, le profil de vol et la qualité du chemin.

2.5 Algorithmes du plus court chemin

La phase de prétraitement, étant statique, permet au planificateur d'appeler l'algorithme de routage plusieurs fois en se basant sur la modélisation de l'espace générée. Dans cette section, nous introduisons tout d'abord l'algorithme A* qui pourra être appliqué sur différents types de structures de données ensuite nous exposons des variantes de cet algorithme de routage applicables dans diverses conditions.

2.5.1 A star (A*)

L'algorithme A* [22] représente l'algorithme de graphe standard pour le problème du plus court chemin. Pour une recherche de chemin entre deux nœuds, la fonction coût nommé f permet de déduire le plus court chemin. Pendant l'exploration, l'algorithme A* considère deux listes de nœuds : la liste des ouverts qui contient les nœuds à visiter lors de l'itération suivante et la liste des fermés qui contient les nœuds déjà visités. Le A* consiste à étendre le nœud ayant le moindre coût dans de la liste des ouverts. Étendre un nœud signifie le placer dans la liste des fermés, insérer les voisins dans la liste des ouverts et évaluer la fonction coût f . L'algorithme s'arrête lorsque le nœud destination est atteint. Pour un nœud visité n , $f(n) = g(n) + h(n)$. Le terme $g(n)$ désigne le coût cumulé du nœud de départ au nœud n et $h(n)$ est une estimation heuristique du coût restant entre le nœud n et la destination. Le choix de l'heuristique est important afin d'offrir des solutions de bonne qualité. Bien que le A* a été largement utilisé car il permet de calculer le plus court chemin et affirmer s'il

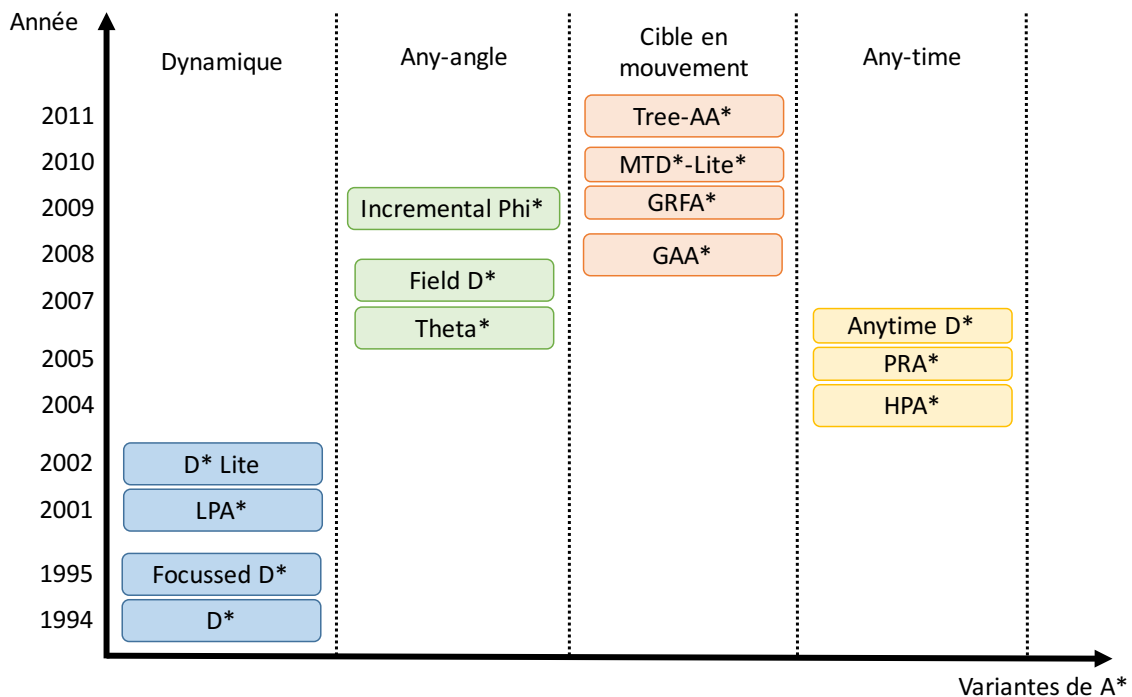


FIGURE 2.9 – Variantes de l'algorithme A*

existe, cet algorithme reste relativement lent et donne des chemins de qualité réduite lors de son application sur une grille régulière de taille importante.

2.5.2 Variantes de A*

Plusieurs travaux ont ciblé l'amélioration de la qualité de la solution générée par l'algorithme A* ainsi que l'adaptation de l'algorithme selon le contexte de l'application. La Fig. 2.9 présente une classification des principales variantes de l'algorithme A* au cours des deux dernières décennies. Une première catégorie vise le cas dynamique afin de réagir rapidement lors d'un changement dans le graphe. Nous citons par exemple le **D*** [60] qui est inspiré du A* avec une propriété dynamique en cas d'apparition d'un objet mobile. Les inconvénients de cet algorithme sont le coût élevé en termes de mémoire et la difficulté de mise en œuvre. Pour cela le **Focussed D*** est proposé comme une amélioration du D*. En effet, il garde le même algorithme mais en réduisant le coût de calcul [61]. Dans cette catégorie nous trouvons aussi l'algorithme **LPA*** (*Lifelong Planning A**) qui réagit dans sa première recherche comme le A*. Par contre, les recherches suivantes sont beaucoup plus rapides puisqu'elles réutilisent la recherche précédente afin de réduire le nombre de nœuds à examiner [34]. Le **D* Lite** utilise la même stratégie de navigation que le D* mais pas le même algorithme [33]. Les auteurs ont appliqué leur propre algorithme LPA* afin de développer le D* Lite qui est capable de

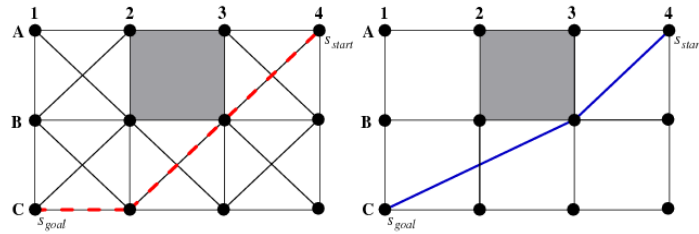


FIGURE 2.10 – Comparaison entre le A* et le Theta*

planifier dynamiquement le plus court chemin entre le nœud courant et la cible. Le D* Lite est considéré beaucoup plus simple à comprendre et à implémenter que le D*.

Dans la seconde catégorie d'algorithmes, nous trouvons les variantes *Any-angle*. À titre d'exemple, nous citons le **Field D***, une variante du D* Lite, qui ne limite pas le mouvement à une grille. Le meilleur chemin peut faire que le mobile se déplace le long de n'importe quel angle et pas seulement de 45° ou 90° entre les points de la grille [12]. L'algorithme **Theta*** [15] est une variante du A* qui donne des chemins plus courts que le A* et le D*. La Fig. 2.10 illustre un exemple pédagogique concernant la qualité de la solution générée pour un A* et un Theta*. Ce dernier a la propriété de trouver des chemins plus directs en vérifiant l'absence d'obstacles avec des nœuds parents avec un surcoût en termes de temps d'exécution. Cependant, le Theta* ne possède pas les capacités de replanification rapide car il est basé sur le A* contrairement au Field D* qui est basé sur D* Lite.

Dans la catégorie *Moving Target* ou bien cible en mouvement, nous trouvons comme algorithmes le **GAA*** (*Generalized Adaptive A**) [63], le **MTD* Lite** (*Moving Target D* Lite*) [65], le **GRFA*** (*Generalized Fringe-Retrieving A**) [64] et le **Tree-AA*** (*Tree Adaptive A**) [24].

La catégorie d'algorithmes *Any-Time* vise à éviter une collision en extrême urgence sans nécessairement prévoir une replanification du chemin. Une solution suboptimale dans ce cas est suffisante. Ce type de calcul du plus court chemin est connu aussi par *Hierarchical path planning*. Nous citons comme exemple d'algorithmes dans cette catégorie le **HPA*** (*Hierarchical Path-Finding A**) [6] et **Anytime D*** [39].

► **Synthèse :** Nous avons étudié les propriétés de plusieurs algorithmes de calcul du plus court chemin afin de proposer une solution adéquate en termes de complexité temporelle et spatiale. Le choix d'un algorithme ne peut pas se faire indépendamment de la modélisation de l'environnement. En effet, la solution doit respecter plusieurs contraintes : éviter les obstacles du terrain, respecter les limitations angulaires de l'appareil, s'adapter selon le type de la mission, respecter le temps réel en cas de replanification dynamique, etc. En plus, le critère de flexibilité est une exigence primordiale dans notre contexte afin de pouvoir spécifier des contraintes liées à l'environnement dans la phase d'exploration.

2.6 Synthèse et positionnement

Après ce tour d'horizon couvrant l'état de l'art industriel et académique, nous allons positionner notre travail en considérant le contexte et les attendus de notre partenaire Airbus Helicopters. La Fig. 2.11 souligne nos choix stratégiques (colorés en vert) au niveau de chaque étape du processus de planification de chemins. Tout d'abord, du fait que nous envisageons une solution générique et adaptative selon le type de l'appareil et de la mission, nous avons choisi :

1) Une solution qui s'appuie sur une modélisation complète de l'espace 3D. En effet, ce choix va nous permettre (i) d'obtenir différents types de profils de vol : navigation proche du relief du terrain, navigation en paliers, navigation en basses altitudes pour le cas militaire (s'échapper d'une zone radar) ou l'inverse pour une mission de type VIP. (ii) de prendre en considération la situation météorologique, à titre d'exemple naviguer au dessous d'un certain seuil afin d'éviter le givrage de l'appareil (iii) d'éviter les obstacles statiques (région montagneuse) et dynamiques.

2) Une technique multi-altitude pour la discrétisation de l'espace 3D (elle sera détaillée dans le Chapitre 3). En effet, nous avons écarté (i) les grilles régulières 3D à cause de la complexité temporelle et spatiale lors de la phase de planification (ii) les grilles irrégulières tels que les Octrees car elles ne permettent pas de maîtriser les angles de montée et de descente selon les caractéristiques de l'appareil (iii) le graphe de visibilité et le diagramme de Voronoï car ils ne permettent pas d'obtenir des chemins avec différents types de profils de vol.

3) La génération d'un graphe de navigation 3D couvrant l'espace 3D discrétisé. il tient en compte (i) des obstacles statiques (relief du terrain, zones non navigables, etc.) (ii) des caractéristiques de l'appareil en termes d'angles de montée et de descente. (iii) La densité du graphe est ajustable selon la complexité en temps et en mémoire (iv) Il est calculé une seule fois lors de la phase de prétraitement.

4) Un algorithme de calcul du plus court chemin de type A* (i) adaptatif en termes de contraintes liées à l'exploration du graphe, à titre d'exemple nous pouvons ajouter une contrainte liée au plafond de navigation ou bien l'angle de changement de direction, les fortes sollicitations de l'appareil, etc. (ii) Actif dans le sens où l'exploration considère un voisinage variable selon la densité du graphe (iii) le temps d'exécution est borné selon la taille du graphe (iv) une exploration de type *Any angle* tant que la solution respecte les contraintes mécaniques de l'appareil (v) Ajustable pour obtenir différents types de profils de vol.

2.7 Conclusion

Dans ce chapitre, nous avons donné un aperçu sur les approches de planification de mouvement les plus appliquées dans la littérature. Une planification de chemin optimale requiert une modélisation complète de l'espace ce qui est le cas pour le diagramme de Voronoï, le

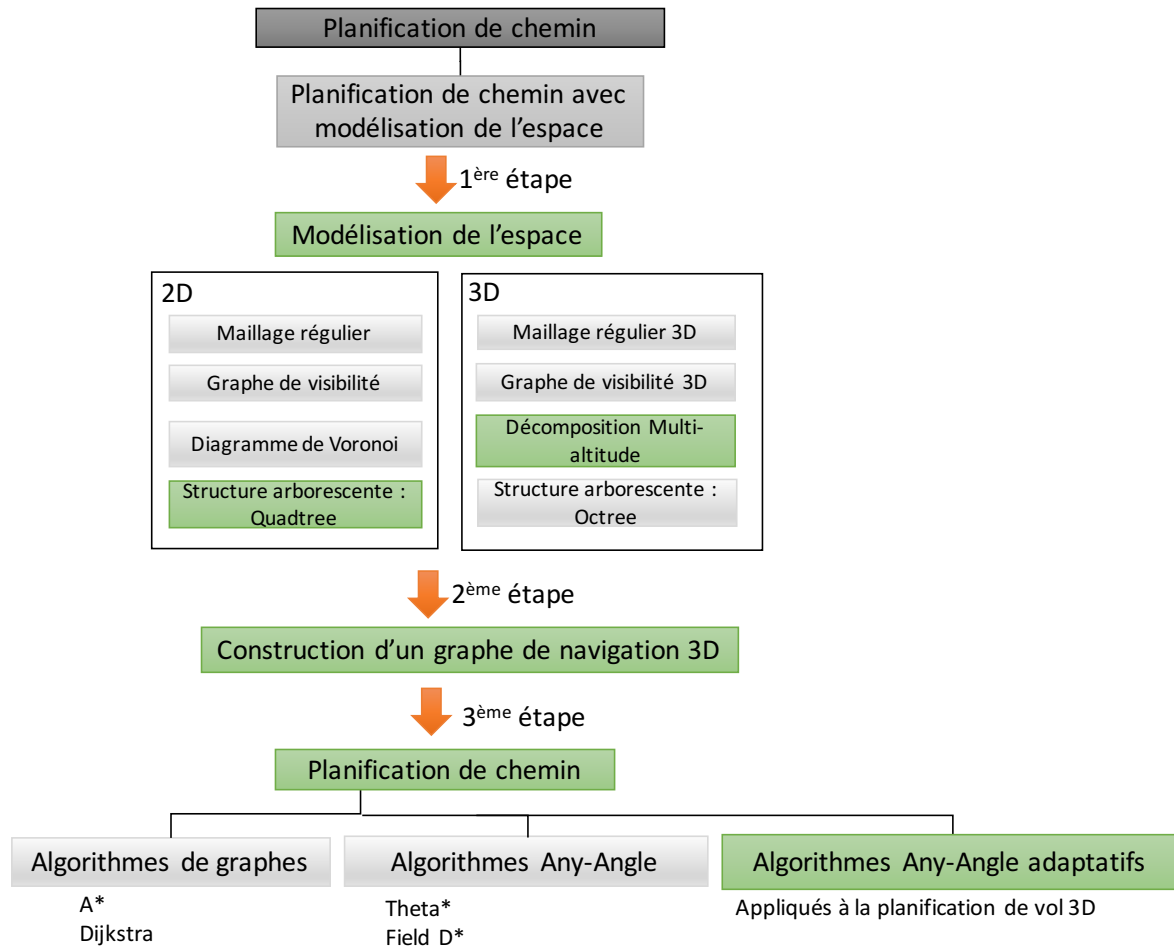


FIGURE 2.11 – Positionnement de nos travaux par rapport à l'état de l'art

graphe de visibilité, etc. Ces solutions impliquent une complexité spatiale et temporelle importante qui limite leur application à la planification de vol de longue distance. Un autre type de planification repose sur les heuristiques tels que RRT, l'algorithme de colonie de fourmis, etc. Malgré leurs durées de calcul intéressantes et la qualité de leurs chemins, ces méthodes sont souvent probabilistes et/ou incomplètes ce qui complique leur application au domaine avionique vu les contraintes de sécurité. Un troisième type d'approches est fondé sur une recherche de graphe. Les méthodes associées sont déterministes, complètes et présentent des durées de calcul relativement acceptables même si elles ne garantissent pas l'optimalité de la solution trouvée. En se référant à cet état de l'art, nous allons proposer notre système de planification de chemins aériens 3D que nous détaillerons dans le Chapitre 3. Ce système est basé sur une solution hybride qui fusionne la représentation hiérarchique de l'espace 3D et la modélisation en graphe de navigation inspirée du graphe de visibilité. La solution que nous proposons profite des avantages liés au maillage adaptatif en termes de coût en temps et en mémoire ainsi que la qualité offerte par les graphes de visibilité. Le système construit un graphe de navigation en considérant un maillage adaptatif de l'espace 3D (multi-Quadtree). Le degré de connexité du graphe est ajustable selon un ensemble de paramètres. Cette façon de construire le graphe permet la planification de vol en permettant d'obtenir différents profils de chemins selon le type de la mission et les contraintes de l'appareil et de satisfaire les contraintes temporelles en situations d'urgence.

Chapitre 3

Système et procédé de planification d'un chemin aérien pour aéronef

3.1 Introduction

Dans ce chapitre, nous allons décrire notre système de planification d'un chemin aérien pour aéronef en considérant la phase hors ligne pour la préparation d'une mission ainsi que la phase en ligne face à l'occurrence d'un évènement complexe. Le contenu de ce chapitre a été déposé sous forme d'un mémoire de brevet à l'Institut National de la Propriété Intellectuelle (INPI). L'invention concerne un système de planification de vol 3D (en ligne et hors ligne) permettant de calculer automatiquement un plan de vol pour un aéronef donné tout en considérant un certain nombre de paramètres d'entrée, des contraintes liées à l'environnement et à l'appareil, et une fonction objectif à minimiser. Le système sera utilisé hors ligne lors de la préparation d'une mission (avant le vol) ou en ligne (durant le vol) pour faire face à un évènement complexe qui peut entraîner la mise à jour du plan de vol. Ce système de planification permet d'obtenir différents profils de plan de vol selon le type de la mission.

Ce chapitre sera organisé comme suit : Dans la Section 3.2, nous décrirons le problème industriel lié à ce projet de thèse. La Section 3.3 introduira le nouveau système de planification de chemins aériens pour aéronef ainsi que le procédé correspondant. Ensuite, dans la Section 3.4, nous présenterons une description chronologique détaillée de notre invention avec des scénarios réels. Les avantages et les restrictions de notre solution seront discutés dans la Section 3.5.

3.2 Le problème industriel posé

La génération d'un plan de vol pour un aéronef en considérant différents paramètres environnementaux (terrain, conditions climatiques, type de mission, etc.) constitue un facteur clé afin d'assurer une mission en toute sécurité et avec un coût réduit (minimisation du temps de

vol, distance, carburant). Aussi, face à l'occurrence d'un évènement complexe en cours de vol, il est crucial de replanifier la mission dans une durée bornée qui dépend de la criticité de la situation. Cet enjeu couvre le domaine technique de discrétisation de l'espace en 2D et en 3D ainsi que le domaine d'optimisation pour le calcul du plus court chemin. Parmi les méthodes de discrétisation, il existe une variété qui offre différents types de maillage : maillage régulier, maillage irrégulier (Quadtree, Octree, BSP...). Pour des raisons de sécurité, la solution fournie doit assurer le déterminisme comportemental et temporel. Malgré les restrictions, l'intervalle des solutions explorées demeure très large. Pour la recherche du plus court chemin, le déterminisme nous a imposé l'élimination de certaines heuristiques malgré leurs capacités à calculer des chemins de bonne qualité dans des durées très courtes (exemple l'algorithme RRT). Par ailleurs, il existe des algorithmes qui offrent des solutions optimales tels que Dijkstra et des algorithmes issus de la programmation dynamique mais qui nécessitent des temps de calcul importants en particulier si nous considérons des terrains étendus avec une discrétisation assez fine de l'espace. Pour contourner cette limitation, il existe des méthodes hybrides sub-optimales. Elles garantissent le déterminisme de la solution (famille de A^* , Θ^* ...) et offrent des résultats satisfaisants dans des durées bornées. Bien que certaines approches soient efficaces lors de l'exploration des petits terrains, elles ne permettent ni de gérer la complexité en termes de temps d'exécution, ni de maîtriser la qualité du plan de vol selon le type de la mission et la criticité de la situation. Les techniques existantes prennent en considération un seul type d'aéronef (avion de ligne, drone, hélicoptère, etc.) pour optimiser le plan de vol selon la dynamique du système sans offrir une solution générique. Une solution générique est capable de s'adapter selon le type de l'appareil et le type de la mission afin de trouver des chemins de différents profils (très proche du terrain pour une mission de recherche et de sauvetage, un vol en palier pour une mission VIP, etc.) Au sol, les durées d'exécution doivent être réduites (inférieur à la minute) afin de générer un plan de vol de type sauvetage (SAMU, SAR) ou intervention (POLICE, ARMEE) vers toute position géographique et en présence d'un environnement complexe (obstacles statiques et dynamiques, zones à éviter, conditions climatiques, etc.). Pendant le vol, une replanification de la mission doit pouvoir s'exécuter en moins d'une minute pour mettre à jour le plan de vol et faire face à une situation d'urgence.

3.3 Un nouveau système de planification de chemin aérien

3.3.1 Principe de la solution apportée

L'invention concerne un système de planification de chemin, pour des aéronefs, dédié à la préparation de missions avant le vol ainsi qu'à la replanification dynamique en cours de vol face à une situation d'urgence (danger météo, dégradation des performances de l'aéronef). Le système de planification peut être déployé sur une station au sol ou bien intégré comme une fonction avionique au sein de l'aéronef. Le système s'appuie sur des techniques de dis-

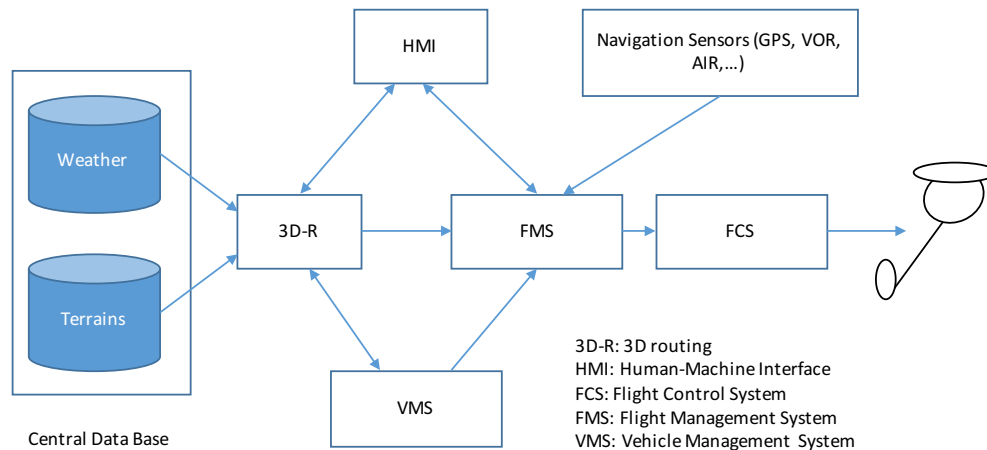


FIGURE 3.1 – Intégration de la nouvelle fonction de routage 3D dans l'architecture générale du système avionique

crétisation de l'espace et le calcul du plus court chemin afin de générer automatiquement des solutions flexibles en termes de profil de chemin en respectant plusieurs contraintes liées à l'appareil, le terrain et l'environnement. La solution proposée est générique et capable de s'adapter selon le type de l'aéronef, le type de la mission et la fonction objectif à minimiser. Le système de planification peut offrir des solutions avec différents compromis entre le temps d'exécution et la qualité du chemin selon le temps disponible qui peut être corrélé avec la criticité de la situation. Le fonctionnement du système de planification de chemin se décline principalement en deux phases : une phase de prétraitement et une phase de routage. La phase de prétraitement permet une discrétisation multi-altitude de l'espace 3D selon une précision donnée et la génération automatique d'un graphe de navigation avec une densité paramétrable. La phase de routage 3D calcule le chemin en prenant en considération un ensemble de contraintes telles que les limitations angulaires en horizontal et en vertical et une fonction objectif à minimiser tels que la distance, le carburant, le temps, etc. Lors de la phase d'exploration du graphe, le système de planification peut communiquer avec le modèle de performances de l'appareil afin de minimiser une fonction coût donnée et de s'assurer au fur et à mesure de l'exploration de la faisabilité de la mission. La génération automatique des plans de vol et la replanification dynamique représentent une brique essentielle pour concevoir des systèmes d'assistance au pilote ainsi que des aéronefs autonomes.

3.3.2 Description détaillée de l'invention

La Fig. 3.1 représente la solution que nous proposons pour automatiser et embarquer la fonction de planification de vol pour un aéronef donné. Dans ce mode de réalisation, nous projetons les concepts de l'invention sur des aéronefs de type hélicoptère. La nouvelle fonc-

tion de routage 3D va s'intégrer dans l'architecture générale présente dans la majorité des hélicoptères (Fig. 3.1). Aujourd'hui, après la préparation de la mission au sol, le plan de vol est transféré au FMS (*Flight Management System*) sous forme d'un ensemble de points de passage (*Waypoints*). Le calcul du plan de vol peut se faire soit dans la phase de préparation de la mission ou dans le cas d'une replanification en cours de vol face à un événement complexe. Le fonctionnement de notre système peut être décrit comme suit : à travers l'Interface HMI (*Human-Machine Interface*), l'utilisateur va saisir l'ensemble des informations liées à la mission à savoir :

- **Le terrain à explorer** : ce paramètre va aider à sélectionner la carte de navigation adéquate à partir d'une base de données géographique.
- **Le type de l'appareil** : ce paramètre nous permet de déduire la masse initiale de l'hélicoptère, les limitations angulaires en montée, en descente et lors d'un changement de direction et la fonction performance liée à l'appareil et qui est appliquée pour l'optimisation du temps de vol ou du carburant consommé.
- **Le type de la mission (recherche et sauvetage, VIP, militaire, etc.)** : ce paramètre est important pour déterminer le profil du plan de vol à générer. À titre d'exemple, pour une mission de recherche et de sauvetage, le calcul du plan de vol doit aboutir à un chemin qui suit le relief du terrain. Pour une mission VIP, le calcul du parcours de l'hélicoptère doit minimiser le nombre de changements de palier ainsi que les fortes manœuvres de l'appareil.
- **Le point de départ, la destination et les zones de passage** : Autre que les points de départ et d'arrivée, l'utilisateur peut imposer à travers l'IHM un ou plusieurs points (zones) de passage lors de la mission.
- **La quantité du carburant disponible** : ce paramètre va nous permettre de valider la faisabilité du plan de vol calculé.
- **La masse de l'appareil** : En plus de la masse initiale, nous considérons le carburant, l'équipage et la charge de l'appareil.
- **Le plafond de navigation (*ceiling*)** : ce paramètre permet d'éviter les zones de danger causées par les conditions météorologiques (manque de visibilité) ainsi que les altitudes de vol présentant un risque de givrage. Ce paramètre nous permet aussi d'obtenir des profils de chemin favorisant le vol en palier (plus confortable pour certains types de mission).
- **Les limitations angulaires en montée, en descente et lors d'un changement de direction** : ces paramètres peuvent être déduits à partir du type de l'appareil mais l'opérateur peut restreindre les capacités de l'appareil afin d'éviter les fortes sollicitations qui ne sont pas souhaitables dans certains types de mission telle qu'une mission VIP

ou assistance médicale. Dans d'autres types de mission (militaire), nous considérons les capacités maximales de l'appareil dans la phase de planification.

La nouvelle fonction embarquée 3D-R va s'appuyer sur cet ensemble de paramètres ainsi que sur des bases de données relatives au terrain et à la situation météorologique pour calculer le plan de vol. Pour ce faire, le 3D-R exécute tout d'abord une phase de prétraitement pour discrétiser l'espace 3D et construire un graphe de navigation. Dans la deuxième phase d'exploration pour aller d'un point de départ à une destination, la fonction 3D-R va solliciter le VMS (Vehicle Management System) intégrant le modèle de performance de l'hélicoptère afin d'évaluer le coût des alternatives possibles et choisir la meilleure solution par rapport à une fonction objectif donnée (minimiser la distance, le temps, etc.). Après la validation du plan de vol par l'opérateur (pilote), il sera communiqué au FMS qui se chargera de la génération de la trajectoire. Il s'agit de calculer les transitions géométriques et la vitesse entre les segments du plan de vol. Enfin, le FCS (Flight Control System) intervient pour contrôler la puissance motrice et les différents actionneurs nécessaires pour exécuter des lois de guidage sur la trajectoire tout en contrôlant la stabilité de l'appareil. Les principales propriétés à satisfaire par la solution de routage 3D à développer sont :

- Complexité maîtrisée en temps et en mémoire
- Capacité de générer différents profils de vol
- Paramétrable pour s'adapter selon le type de la mission
- Déterminisme
- Simplicité du plan de vol généré (pour faciliter l'édition ultérieure)

Ciblant la satisfaction des propriétés précédentes, nous allons décrire le fonctionnement de la nouvelle fonction 3D-R qui se décline principalement en deux phases : le prétraitement (discrétisation de l'espace et construction du graphe de navigation) et le routage 3D.

- **La phase de prétraitement** permet de :
 - Discrétiser l'espace 3D sous forme de plusieurs paliers 2D qui vont être discrétisés avec des grilles irrégulières comme schématisé dans la Fig. 3.2. Cette façon pour discrétiser l'espace permet de réduire la complexité en temps et en mémoire en réalisant une décorrélation entre les deux composantes verticale et horizontale de l'espace. Les grilles irrégulières seront utilisées pour créer un maillage adaptatif selon la densité des obstacles sur chaque palier. La précision du maillage pourra être fixée afin d'obtenir des angles de montée et de descente respectant les caractéristiques de l'aéronef.
 - Construire le graphe de navigation en générant des arcs horizontaux sur chaque palier selon un voisinage variable et des arcs verticaux pour le raccordement entre les différents niveaux selon les angles de montée et de descente tolérés par l'appareil.

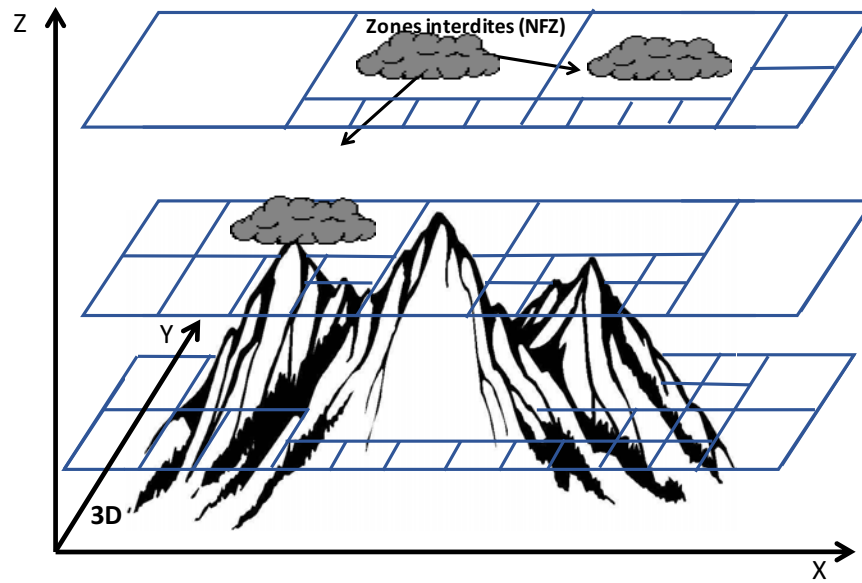


FIGURE 3.2 – Découpage multi-altitude de l'espace 3D par des grilles irrégulières

Cette façon de construire le graphe permet d'obtenir une densité variable, ce qui permet d'ajuster la qualité de la solution en fonction du temps disponible.

- **La phase de routage** permet d'explorer le graphe de navigation et de trouver un chemin qui répond aux paramètres de la mission (limitations angulaires, ceiling, changement de direction...) tout en minimisant une fonction coût (temps, distance, etc.). L'algorithme de routage élabore des solutions suboptimales. Par contre, il offre des résultats satisfaisants et il est capable de converger vers l'optimalité avec un graphe complet au détriment du temps d'exécution.

Lors de la préparation d'une mission, la phase de prétraitement est lancée une seule fois. Même si le temps d'exécution de cette phase est important, ceci n'influence pas la phase de routage notamment en cas d'urgence où le système doit fournir une solution en temps réel. Le diagramme de la Fig. 3.3 décrit les différentes étapes nécessaires pour le fonctionnement du système. Ces étapes sont réparties principalement dans deux phases : la première phase constitue le prétraitement qui est exécuté une seule fois lors de la préparation d'une mission. Les données générées seront exploitées avant et tout au long de la mission. En effet après la phase de prétraitement, la deuxième phase de routage permet d'explorer plusieurs solutions possibles en faisant varier les paramètres de la mission (coordonnées de départ/arrivée, ceiling, les limitations angulaires de l'appareil...). L'algorithme de routage peut être appliqué à la fois en préparation de mission et en cours du vol pour la replanification en cas d'urgence. Bien qu'ils exploitent les mêmes structures de données relatives à la mission et aux performances de l'appareil, les deux phases sont totalement découplées.

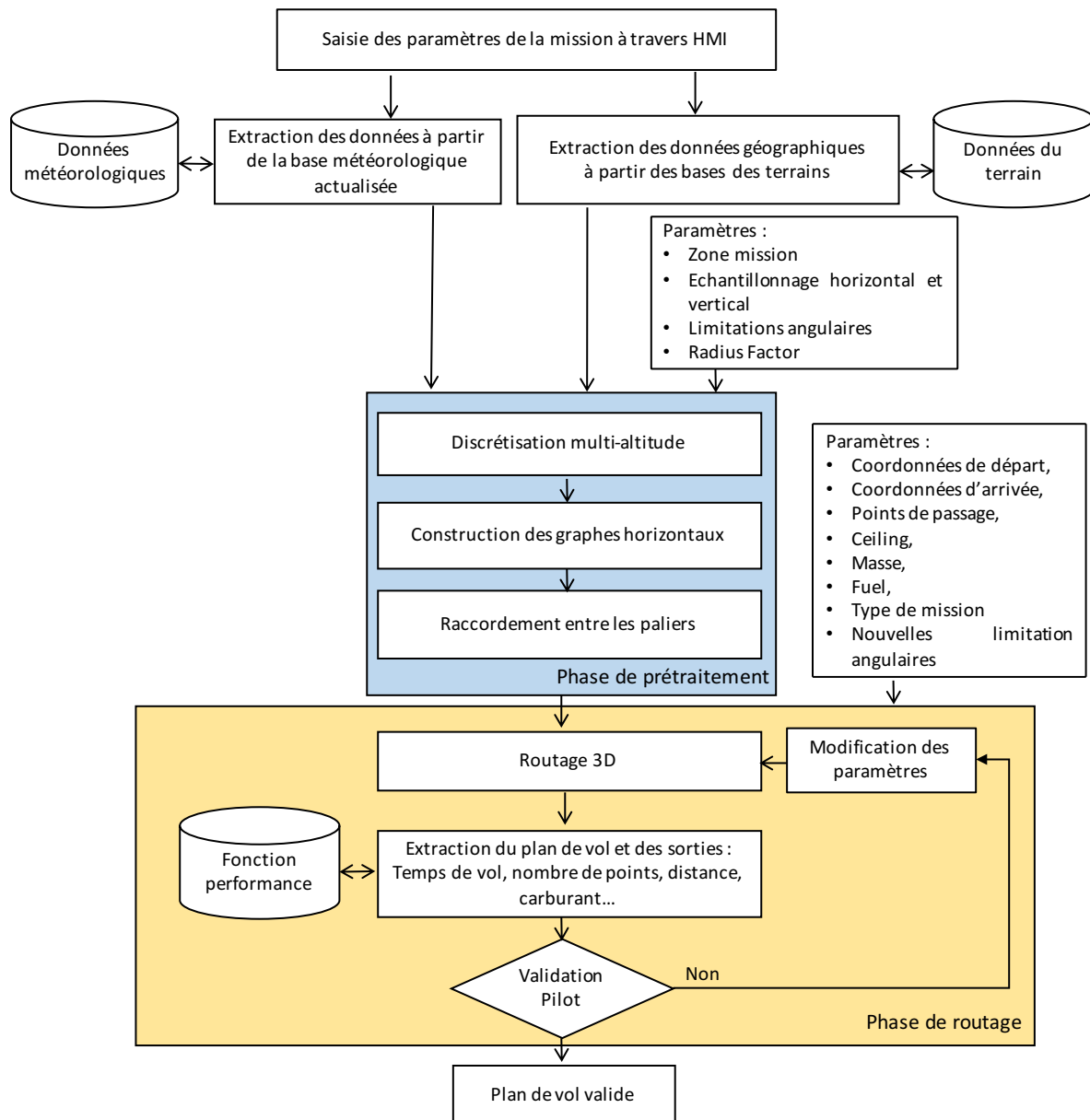


FIGURE 3.3 – Description détaillée de la fonction de planification de missions

Le fonctionnement du système commence par la récupération des paramètres de la mission fournis par l'opérateur. Le système distingue 4 types de paramètres comme mentionné dans le Tableau 3.1 :

- les paramètres de mission requis pour la phase de prétraitement
- les paramètres de l'algorithme requis pour la phase de prétraitement
- les paramètres de mission requis pour la phase traitement (ou routage)
- les paramètres de l'algorithme requis pour la phase traitement

TABLE 3.1 – Paramètres du système de planification de plans de vol

	Prétraitement (création du graphe de navigation)	Traitement (exploration du graphe)
Mission	Zone mission (BD terrain et obstacles)	Origine, destination, contraintes (points imposés), masse, fuel, contraintes météo
Algorithme	Échantillonnage horizontal et vertical, <i>Radius Factor</i> , limitations angulaires	Nouvelles limitation angulaires adaptées au type de mission

Une fois que le système dispose des paramètres requis pour le prétraitement, il va procéder comme suit :

1. Extraction des données à partir des bases de données géographiques et météorologiques nécessaires pour la phase de prétraitement en plus des paramètres de la mission,
2. Discrétisation horizontale multi-altitude sous forme de différents paliers 2D (entre une altitude minimale et une altitude maximale) en utilisant une structure irrégulière de type Quadtree. Il s'agit d'un maillage adaptatif selon la densité des obstacles permettant de réduire la taille mémoire en comparaison avec une structure régulière. La méthode utilisée est itérative et permet d'ajuster la précision du maillage selon les performances de l'appareil,
3. Construction d'un graphe de navigation 2D sur chacun des paliers en fonction du paramètre *Radius Factor*. Ce paramètre permettra d'ajuster la densité du graphe selon le temps de prétraitement disponible et donc va agir sur la qualité de la solution trouvée,
4. Raccordement entre les paliers (consécutifs, pris deux à deux) en choisissant des points de montée d'un niveau à un autre et en construisant des arcs verticaux qui respectent les limitations angulaires et qui évitent les obstacles. Les arcs de raccordement entre un nœud et le palier au-dessus constituent un cône creux permettant de respecter les limitations angulaires liées à l'appareil,

5. Construction du graphe complet en fusionnant les graphes de navigation 2D et les arcs de raccordement verticaux. Les arcs de ce graphe constituent l'ensemble des chemins possibles dans l'espace de solutions. Tous ces chemins permettent d'éviter les obstacles, Lorsque le système dispose des paramètres requis pour la phase de traitement (exploration) :
6. La phase de routage peut être appelée d'une façon statique ou dynamique à plusieurs reprises. En effet, l'application de l'algorithme de routage sur le graphe de navigation permet de calculer un chemin entre deux points de l'espace 3D tout en optimisant l'un des critères suivants : distance, temps de vol, carburant, etc. L'algorithme permet d'explorer des arcs au lieu des nœuds ce qui garantit le respect des contraintes angulaires de la mission et de l'appareil. De plus, il est possible de coupler le routeur 3D avec la fonction performance ce qui permettra une exploration qui met à jour le coût de chaque arc visité (temps de vol, masse). Le profil et la qualité de la solution générée dépendent de l'ensemble des paramètres qui ont été saisis. En particulier, nous soulignons que l'augmentation des valeurs du *Radius Factor*, de l'intervalle des angles de montée/descente, ainsi que la précision de la discrétisation permet d'améliorer la qualité de la solution obtenue au détriment du temps d'exécution des phases de prétraitement et de routage.
7. L'opérateur évalue la qualité du résultat qui regroupe les différentes sorties de l'algorithme : plan de vol, distance parcourue, temps de vol, nombre de segments, etc. Si la sortie est valide, le plan de vol sera transmis à la prochaine fonction. Sinon, l'opérateur modifie les paramètres de la mission et relance la fonction de routage.

3.4 Description chronologique détaillée

Pour décrire le fonctionnement du système, nous avons projeté les différentes étapes sur un terrain qui représente une zone des « Pyrénées ». La représentation numérique du terrain est fournie par une matrice d'élévation sous forme d'une grille régulière avec une précision donnée sur chacun des axes x et y (repère cartésien). La Fig. 3.4 montre la zone géographique ainsi que le relief du terrain pris en exemple. Il s'agit d'une région hétérogène en termes de relief avec des altitudes allant de 150m jusqu'à 3450m.

3.4.1 La phase de prétraitement

► **La prise de marges de sécurité :** Au préalable de la discrétisation de l'espace, il est nécessaire d'augmenter la taille des obstacles de manière à garantir une marge de sécurité horizontale et verticale réglementaire. Cette opération est réalisée par un rehaussé forfaitaire du terrain, suivi d'une opération de dilatation horizontale.

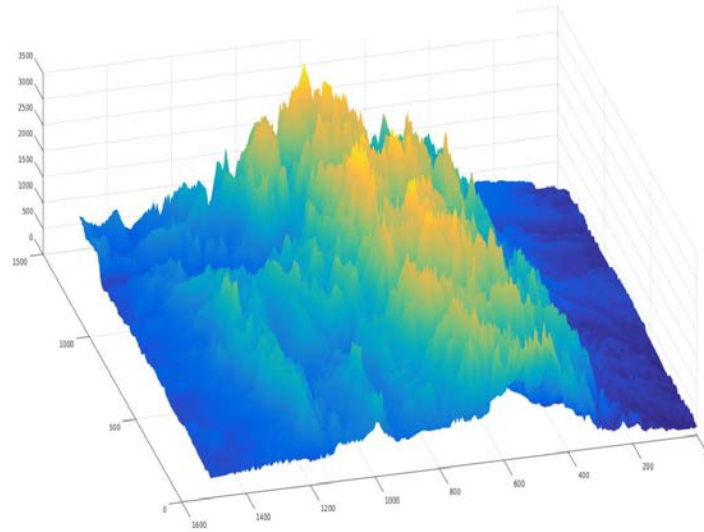


FIGURE 3.4 – Exemple de terrain : Les Pyrénées

► **La discrétisation de l'espace en 3D :** Afin de discrétiser notre espace 3D, nous avons opté pour l'approche multi-altitude sous forme de N paliers 2D séparés d'une distance P entre $Altitude_{min}$ et $Altitude_{max}$ du terrain. À titre d'exemple, la valeur de P pourra être fixée à $150m$ dans le cas d'un aéronef de type hélicoptère. Cette valeur peut changer selon le type de l'appareil, la taille du terrain, la qualité de la solution, etc. Sur les différents paliers de vol (séparés de $150m$), nous avons appliqué un maillage de type Quadtree : un maillage adaptatif selon la densité des obstacles. Le maillage en Quadtree consiste à une division récursive de l'espace 2D en quatre rectangles tant qu'il y a un obstacle dans le rectangle traité jusqu'à atteindre une précision donnée. Ce type de discrétisation présente plusieurs avantages : tout d'abord le temps réduit pour générer la structure de données ainsi que la faible taille mémoire consommée par cette structure. Ensuite, la précision du maillage (taille de la plus petite et la plus grande case) pourra être fixée selon plusieurs critères telles que la satisfaction des limitations angulaires en montée et en descente, l'optimalité de la solution, etc. La structure de données générée consiste à l'ensemble des cases vides qui vont être utilisées par la suite pour naviguer de façon horizontale sur chaque plan 2D. Le raccordement entre ces paliers deux à deux par des liens verticaux respectant les limitations angulaires de l'appareil va permettre la navigation dans l'espace 3D. Comme illustré dans la Fig. 3.5, la solution multi-altitude permet de générer des plans de vol qui évitent les obstacles et qui respectent les contraintes angulaires liées à l'appareil et à la mission ce que nous allons détailler dans la partie construction du graphe de navigation. Dans la littérature, il existe différents types de maillage pour discrétiser l'espace 3D, nous citons ici les principales méthodes ainsi que leurs limitations :

— Les grilles régulières 3D : ces structures nécessitent un très grand nombre de nœuds pour

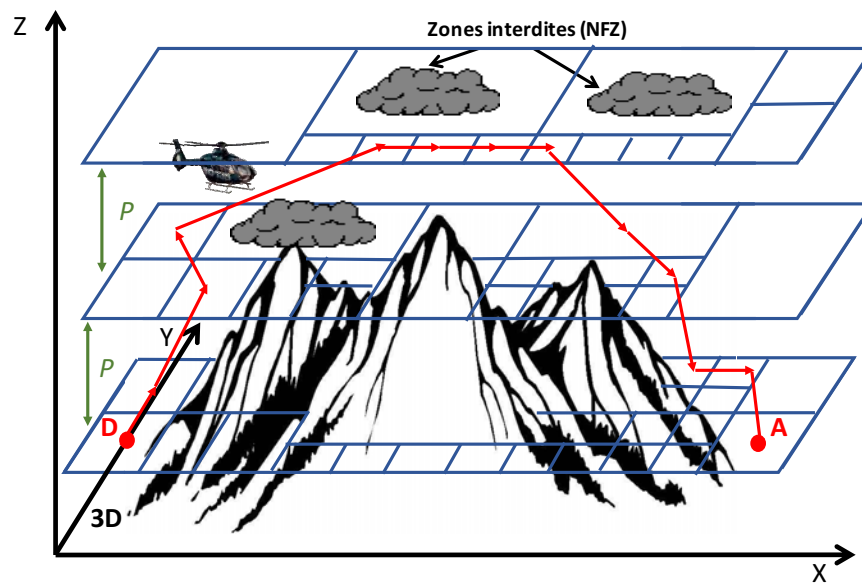


FIGURE 3.5 – Exemple de chemin 3D permettant d'éviter les obstacles dans une zone montagneuse

modéliser le terrain ce qui implique une complexité spatiale et temporelle insolvable lors de la génération du graphe de navigation ou le calcul du plus court chemin.

- Les Octrees : il s'agit d'un maillage adaptatif de l'espace 3D selon la densité d'obstacles. La construction de l'Octree est récursive. En effet, le système découpe l'espace de solutions en 8 octants récursivement tant qu'il y a des obstacles dans la zone en cours de traitement et jusqu'à atteindre une précision donnée. Malgré le gain en temps de calcul et en mémoire, les Octrees présentent un maillage irrégulier en 3D ce qui ne permet pas de maîtriser les angles de montée et de descente. Ceci n'est pas applicable pour des aéronefs de type hélicoptère.
- Les BSP (Binary Space Partitioning) : est un système utilisé pour diviser l'espace en zones convexes. Ces cellules sont délimitées par des hyperplans. Dans le cas d'un espace 3D, les séparations sont des plans et les cellules sont des polyèdres (souvent des parallélépipèdes rectangles). En plus, de la complexité de calcul introduite par l'hétérogénéité de cette structure, les BSP présentent les mêmes inconvénients que les Octree.
- Les graphes de visibilité : cette structure permet de définir le graphe en délimitant les obstacles dans des formes géométriques dont les sommets représentent les nœuds. En robotique, l'application des graphes de visibilité sur des drones est avantageuse (un plan de vol proche du relief du terrain). Néanmoins, pour d'autres appareils dont les performances et les types de missions à assurer demandent plus de flexibilité lors du changement de palier de vol, cette structure ne permet pas de trouver des plans de vol

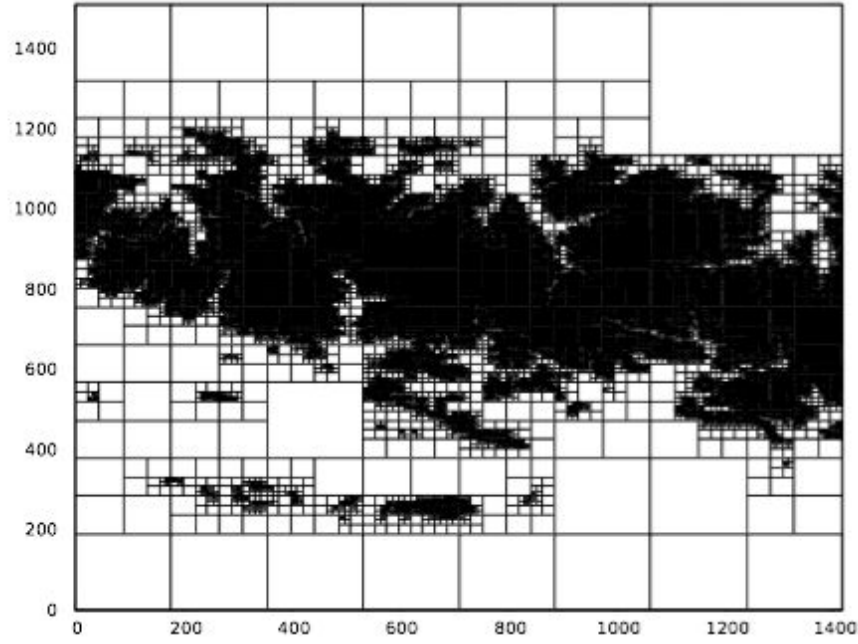


FIGURE 3.6 – Quadtree « standard » appliqué sur la carte des « Pyrénées » à une altitude 1500m

avec différents types de profils.

La Fig. 3.6 est une représentation du Quadtree qui correspond à une coupe horizontale du terrain « Pyrénées » à une altitude de 1500m. Nous distinguons les cases noires qui constituent les zones où il y a des obstacles à éviter et les cases vides navigables de différentes tailles. En effet, le maillage est assez fin (cases de petite taille) dans les zones dominées par des obstacles contrairement aux zones libres où le maillage est grossier (cases de grande taille). À des altitudes élevées, les obstacles sont moins denses ce qui réduit le nombre des cases générées. C'est le cas de l'exemple de la Fig. 3.7 qui représente une coupe horizontale du terrain « Pyrénées » à une altitude de 2250m. Seulement les cases vides seront stockées dans une structure de données pour la navigation horizontale. Cette structure rassemble les feuilles du Quadtree. Afin d'identifier les centres des cases vides appartenant aux différents paliers, une indexation unique est adoptée : M est le nombre total des cases vides, $I_{min,k}$ est l'indice de la première case du palier k , $I_{max,k}$ est l'indice de la dernière case du palier k avec $k \in [1..N]$, donc la liste totale des indices de notre terrain 3D discrétisé est définie comme suit : $[1..I_{max,1}, \dots, I_{min,k}..I_{max,k}, I_{min,k+1}..I_{max,k+1}, \dots, I_{min,N}..M]$.

Malgré le gain en mémoire et en temps de calcul, l'utilisation des Quadrees présente une limitation dans notre contexte. En effet, les Quadrees générés à des altitudes élevées possèdent de très grandes cases vu la faible présence d'obstacles, ce qui ne permet pas d'atteindre la précision nécessaire pour construire les arcs de raccordement en montée et en descente.

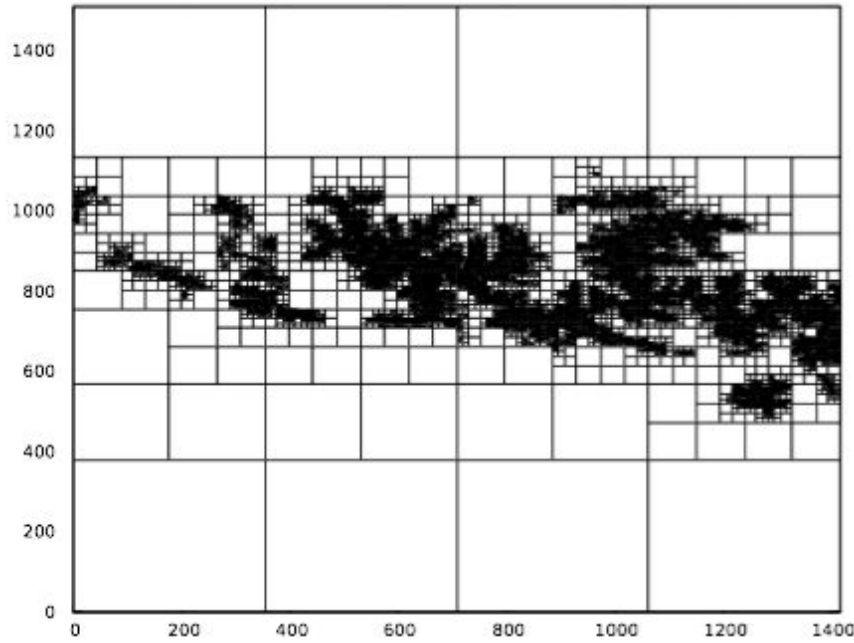


FIGURE 3.7 – Quadtree « standard » appliqué sur la carte des « Pyrénées » à une altitude 2250m

Ceci est susceptible d'influencer la phase de routage en interdisant des chemins possibles malgré l'absence d'obstacles. Pour pallier cette limitation, nous avons introduit une nouvelle option de maillage qui consiste à discrétiser davantage le terrain même en cas d'absence d'obstacles jusqu'à arriver à la précision requise par le système $A = f(P, \alpha_{max})$ où $\alpha_{max} = \max(\alpha_{max\ climb}, \alpha_{max\ descent})$.

Contrairement au mode standard (Fig. 3.7), pour lequel la discrétisation s'achève dans deux cas : la précision souhaitée est atteinte (précision de la matrice d'altitude) ou la case traitée est uniforme (elle contient seulement des obstacles ou présente une zone vide), le mode « *Accurate* » garantit un maillage plus fin permettant d'atteindre les limitations angulaires requises par l'aéronef. La Fig. 3.8 montre le maillage « *Quadtree Accurate* » appliqué sur la carte des « Pyrénées » à une altitude de 2250m. La précision A agira dans les prochaines étapes sur la qualité de la solution trouvée. Le principe de découpage du mode « *Accurate* » satisfait les conditions citées dans le mode « *Quadtree Standard* » avec une condition supplémentaire : si la case en cours de traitement est vide, le terrain est récursivement découpé jusqu'à atteindre la précision A . Ce qui n'est pas le cas pour les cases « obstacles » pour lesquelles le découpage est arrêté dès qu'elles ne contiennent que des obstacles. Avec ce mode de discrétisation, nous garantissons la précision qui correspond à l'utilisation d'une grille régulière tout en réduisant le nombre de cases « obstacles ». Ceci permettra d'économiser le temps de calcul global et la mémoire consommée. Avec le mode « *Quadtree Accurate* », la complexité spatiale et temporelle

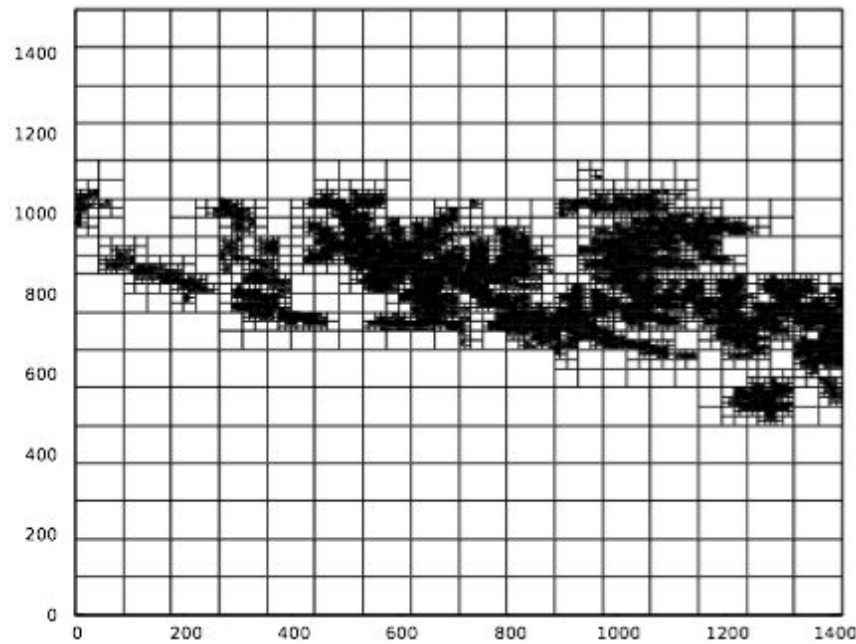


FIGURE 3.8 – Quadtree « Accurate » appliqué sur la carte des « Pyrénées » à une altitude 2250m

est plus importante par contre elle garantit une solution de meilleure qualité.

► **Génération du graphe de navigation :** Après la génération des Quadrees correspondants aux différentes altitudes, la construction du graphe de navigation constitue la deuxième étape du prétraitement. Ce graphe représente le résultat final du prétraitement qui sera utilisé par la phase de routage 3D. Les centres des M cases vides des différents Quadrees sur les N paliers allant de $Altitude_{min}$ jusqu'à $Altitude_{max}$ sont considérés comme l'ensemble des nœuds du graphe.

Comme première étape, nous construisons le graphe horizontal 2D correspondant à chaque palier k avec $k \in [1..N]$. Par la suite, nous assurons la construction des arcs verticaux afin de raccorder entre les paliers en respectant les limitations angulaires. Le raccordement est répété $N - 1$ fois pour relier entre chaque paire de paliers consécutifs (k et $k + 1$). Enfin, le graphe de navigation résulte de la fusion entre les graphes horizontaux et les arcs de raccordement entre les paliers.

La construction du graphe représente une étape cruciale pour le fonctionnement du système. Elle intègre la partie de détection d'obstacles avant la phase du routage 3D ce qui accélère le calcul du plan de vol. Elle permet également de construire des connexions directes entre les nœuds selon plusieurs paramètres ce qui va ajuster la qualité des solutions postérieurement générées en fonction du temps disponible. En effet, plus le graphe est connexe plus la

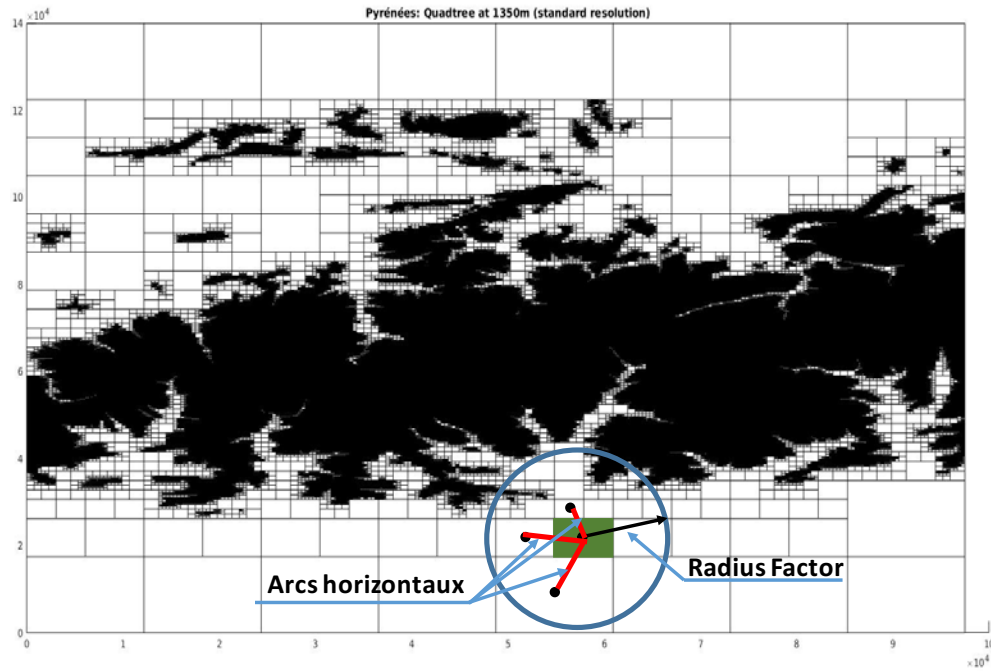


FIGURE 3.9 – Description de la construction du graphe horizontal en se référant aux centres des cases du Quadtree

taille mémoire et le temps nécessaires pour sa construction sont importants. Ce paramétrage est primordial lors de la replanification de la mission en cas d'urgence selon la criticité de la situation. Par conséquent, il est important de trouver le compromis entre la qualité de solution et le temps disponible pour le prétraitement.

Étape 1 - Construction des graphes 2D : Afin de générer le graphe 2D correspondant à chaque altitude, nous avons introduit le paramètre *Radius Factor* (RF). Ce paramètre définit le voisinage de chaque nœud afin de trouver des connexions directes vers d'autres nœuds ce qui permet d'ajuster la densité du graphe. L'ensemble des graphes 2D représente les composantes horizontales du graphe de navigation.

La Fig. 3.9 montre la façon avec laquelle le voisinage est déterminé en fonction du RF . Ce voisinage est défini par un cercle dont les propriétés sont décrites comme suit : le centre du cercle correspond au centre $C_{i,k}$ de la case i ($i \in [I_{min,k}..I_{max,k}]$) du palier k , le rayon du cercle est défini par la multiplication du RF et la diagonale de la case associée. L'ensemble des voisins du nœud i est assuré par un calcul géométrique et par l'application d'un algorithme de tracé de segment pour la détection d'obstacles. Un nœud ($j \in [i..I_{max,k}]$) est considéré comme voisin s'il remplit les conditions suivantes :

- $C_{j,k}$ (centre de la case j) appartient au cercle, ceci est vérifié par une comparaison entre la distance d_{C_i,C_j} et le rayon du cercle.

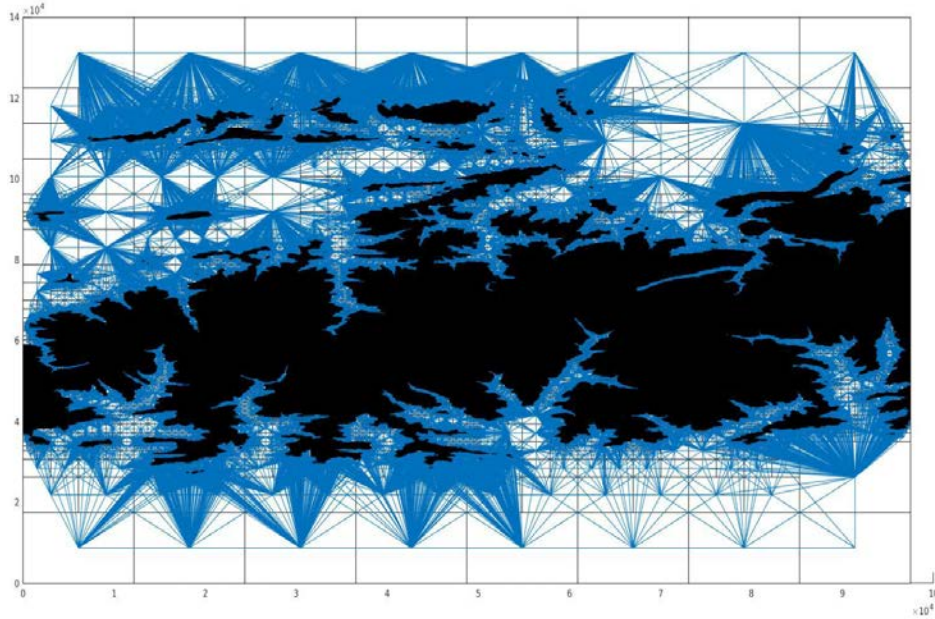


FIGURE 3.10 – Affichage du graphe horizontal calqué sur le Quadtree correspondant à la carte des « Pyrénées » à une altitude 2250m et un $RF = 1$

- Il n'y a pas d'obstacle qui coupe le segment entre $C_{i,k}$ et $C_{j,k}$. Cette condition est vérifiée grâce à l'utilisation d'un algorithme de tracé de segment en 2D. L'algorithme est appliqué sur la région rectangulaire située entre les deux nœuds en question en se référant à la matrice d'élévation.

Dans ce cas, l'arc entre les deux nœuds est construit et l'algorithme passe à l'itération suivante ($j + 1$) jusqu'au traitement de la totalité des nœuds du palier en question ($j = I_{max,k}$). Une fois le graphe 2D du palier k est déterminé, l'algorithme réitère en passant au palier supérieur jusqu'à atteindre $Altitude_{max}$ ($k = N$). Les Fig. 3.10 et 3.11 illustrent les graphes 2D correspondants au terrain des « Pyrénées » à une altitude de 2250m respectivement pour un $RF = 1$ et un $RF = 2$. D'un côté, avec un RF plus grand le voisinage exploré demeure plus étendu et donc le nombre d'arcs construits est plus important, ce qui explique une meilleure densité du graphe. De l'autre côté, l'augmentation du RF permet d'améliorer la qualité de la solution en trouvant des arcs directs entre des nœuds éloignés.

Pour une région hétérogène en termes d'obstacles, le maillage est relativement fin. C'est le cas d'un Quadtree construit à une altitude moyenne. Par conséquent, il est préférable d'appliquer un RF important. Ceci permet de relier les centres des différentes cases et d'assurer une densité acceptable. Pour les altitudes élevées, le maillage est généralement grossier. Un RF réduit est suffisant pour construire des arcs longs. Un tel graphe offre une qualité de solution adéquate au cas d'utilisation avionique.

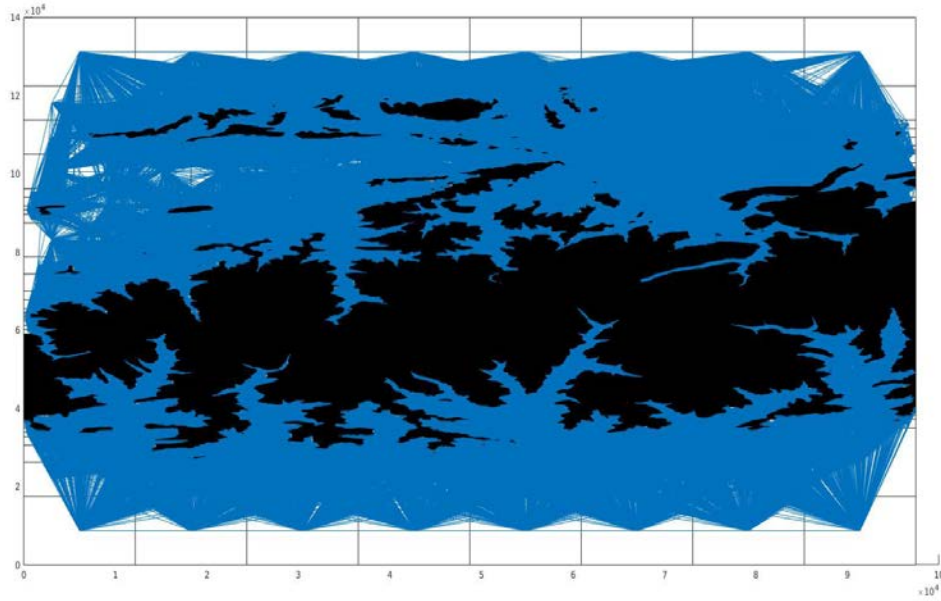


FIGURE 3.11 – Affichage du graphe horizontal calqué sur le Quadtree correspondant à la carte des « Pyrénées » à une altitude 2250m et un $RF = 2$

Étape 2 - Raccordement entre les paliers (composante verticale du graphe de navigation) : Après la construction des graphes 2D, l'algorithme procède à la génération des arcs verticaux pour le raccordement entre les différents paliers. Ce raccordement se fait en choisissant des points de montée d'un niveau à un autre et en construisant des arcs verticaux qui respectent les limitations angulaires et qui évitent les obstacles comme illustré par la Fig. 3.12. Un obstacle qui coupe l'éventuel arc $\widehat{AA_1}$ interdit sa construction. Contrairement à $\widehat{AA_1}$, l'arc $\widehat{AA_2}$ évite les obstacles mais ne respecte pas les limitations angulaires puisque son angle par rapport au plan horizontal α_2 est supérieur à α_{max} . Ceci empêche la construction de l'arc $\widehat{AA_2}$. Seulement l'arc $\widehat{AA_3}$ peut être construit puisqu'il remplit les conditions : il respecte la contrainte angulaire et évite les obstacles.

À partir du centre $C_{i,k}$ d'une case i appartenant à un palier k avec $i \in [I_{min,k}..I_{max,k}]$ et $k \in [1..N - 1]$, nous allons chercher des raccordements possibles avec l'ensemble des centres de case $C_{j,k+1}$ appartenant au palier supérieur $k + 1$ avec $j \in [I_{min,k+1}..I_{max,k+1}]$. Un nouvel arc vertical est construit si les conditions suivantes sont satisfaites :

- Il n'y a pas d'obstacle qui coupe le segment $C_{i,k}$ et $C_{j,k+1}$. Cette condition est vérifiée grâce à l'utilisation d'un algorithme de tracé de segment en 3D. L'algorithme est appliqué sur le parallélépipède dont les extrémités sont définies par les deux nœuds en question en se référant à la matrice d'élévation.
- L'angle vertical du vecteur $\overrightarrow{C_{i,k}C_{j,k+1}}$ est compris entre α_{min} et α_{max} (les limitations angulaires de l'appareil). En d'autres termes, l'ensemble des arcs qui respectent cette

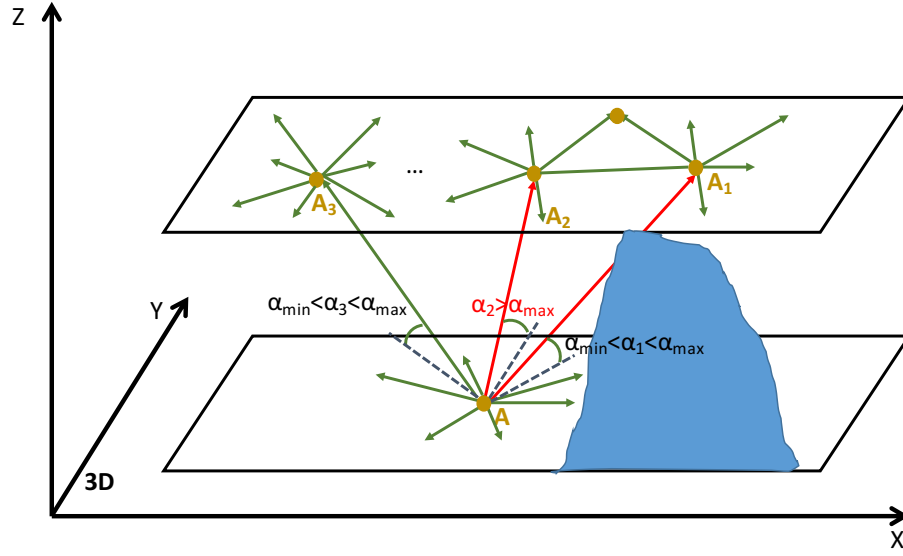


FIGURE 3.12 – Construction des arcs de raccordement entre deux paliers consécutifs

condition appartiennent à un cône creux dont $R_1 = f(\alpha_{max})$ et $R_2 = f(\alpha_{min})$ avec $R_2 > R_1$.

Ce traitement est appliqué pour l'ensemble des nœuds des différents paliers jusqu'à $k = N - 1$. Les limitations angulaires peuvent être ajustées par le pilote selon le type de la mission et le modèle de performance de l'appareil.

Étape 3 - Construction du graphe complet : Le graphe 3D résulte de la fusion entre l'ensemble des graphes 2D correspondants aux N paliers et les arcs de raccordement entre les paliers. Le graphe 3D généré présente plusieurs avantages à savoir :

- Le graphe est non-orienté et couvre la totalité du terrain, ce qui permet de se déplacer dans un environnement 3D et d'explorer des chemins dans tous les sens possibles ;
- Le graphe peut être construit de façon adaptative en fonction des caractéristiques de l'appareil et le type de mission ;
- Les propriétés du graphe peuvent être vérifiées avec des outils conventionnels telles que la connexité et la densité ;
- La détection d'obstacles se fait d'une façon statique lors du prétraitement ce qui garantit un temps de routage réduit par rapport au nombre d'arcs à explorer ;
- La décorrélation entre les composantes horizontale et verticale du graphe ce qui permet de réduire la complexité ;
- La flexibilité de la construction du graphe en fonction du RF pour la composante

horizontale et des limitations angulaires pour la composante horizontale permet une maîtrise de la complexité de la solution en temps et en mémoire.

Un graphe avec un RF important devient plus dense ce qui permet de générer des connexions 2D plus directes entre les nœuds ce qui signifie des solutions de meilleure qualité au détriment du temps de calcul. De même, un intervalle $[\alpha_{min}, \alpha_{max}]$ de montée et de descente plus important permet de construire un graphe plus riche en arcs verticaux, ce qui apportera à l'algorithme de routage une grande flexibilité pour trouver un plan de vol qui respecte les limitations lors des changements de palier.

3.4.2 La phase d'optimisation de la trajectoire de décollage et d'approche

Une fois l'origine et la destination connues, et en préalable à la phase de routage « croisière », une optimisation des axes de décollage et d'approche est calculée pour relier un point géographique au sol jusqu'à une hauteur de sécurité correspondante à un palier du graphe généré lors du prétraitement car le routage n'autorise pas la création de segments proches des obstacles ou du terrain.

Pour la phase de décollage, il s'agit de déterminer le meilleur axe de montée (route et pente) en fonction des obstacles et du vent. Si le départ a lieu depuis un terrain aménagé (piste ou héliport), le système restreindra la recherche de l'axe aux contraintes réglementaires du terrain. Le système privilégiera par ordre de priorité décroissant :

- Recherche d'une route dans les contraintes réglementaires applicables pour les terrains aménagés.
- Route face au vent si vent > seuil
- Route en direction de la destination

Chaque route fera l'objet d'une vérification de l'évitement des obstacles pour la pente maximisant la vitesse de montée. Si nécessaire, la pente sera augmentée jusqu'à une valeur limite. Si aucune solution n'est trouvée pour la route optimale, d'autres routes, alternativement de part et d'autre de la route optimale seront testées jusqu'à trouver une solution satisfaisante. Cette opération permet de définir une route et une pente de décollage ainsi que le point d'intersection entre cet axe et le premier palier accessible du graphe. Ce point servira à initier le routage.

Pour la phase d'approche, la recherche est comparable au cas du décollage avec en plus l'introduction d'une contrainte supplémentaire lors de la phase de vérification de l'évitement d'obstacles : la route de descente prolongée au-delà du point cible doit être sans obstacles afin de permettre une remise des gaz (approche manquée) en toute sécurité. Cette opération permet de définir une route et pente d'approche finale ainsi que le point d'intersection entre cet axe et le dernier palier accessible du graphe avant le sol. Ce point constituera la cible à atteindre pour le routage.

3.4.3 La phase de routage

Cette phase consiste à appliquer un algorithme de calcul du plus court chemin sur le graphe construit afin de trouver un plan de vol entre deux points du terrain en respectant un certain nombre de contraintes liées à l'appareil, la mission et l'environnement. En effet, lors de la construction du graphe un coût est associé à chaque arc en termes de distance. L'algorithme de routage détermine l'ensemble des points intermédiaires permettant de minimiser la distance totale entre le point de départ et d'arrivée. Le coût de l'arc en termes de distance peut être calculé à priori (dans la phase de prétraitement) car il est invariant lors de la phase d'exploration ce qui réduit la durée du routage. Cependant, l'exploration du plus court chemin peut se baser sur d'autres critères telle que l'optimisation du temps de vol ou du carburant. Pour cette raison, les arcs du graphe sont aussi annotés avec d'autres champs de type masse, vitesse, etc. (voir Fig. 3.13). Avec des critères qui dépendent de la dynamique de l'appareil, le coût d'un arc dépend de l'arc précédemment exploré ainsi que le sens de parcours de l'arc (montée ou descente). Afin de garantir la généralité de la solution et de garder les propriétés géométriques, nous considérons la même structure du graphe : basée sur la distance lors du prétraitement. Pendant le routage et plus précisément lors de l'exploration, nous calculons le coût d'un arc relatif à un critère donné grâce à une fonction liée aux performances de l'appareil (fonction VMS) et nous mettons à jour les valeurs des différents champs (temps, masse, vitesse, etc.). Ce calcul ne peut pas être fait à priori (phase de prétraitement) car les coûts dépendent des arcs précédemment visités. La phase d'exploration se fait de façon itérative à partir du nœud de départ, nous commençons par le calcul des coûts des nœuds candidats (les voisins), une fonction objectif (par exemple minimiser la distance) permet de choisir le meilleur nœud, l'étape suivante est d'explorer les voisins du nouveau nœud et nous itérons jusqu'à arriver à la destination.

Dans notre système, l'opérateur peut fixer des contraintes spécifiques liées à la mission ou à l'environnement telles que les limitations angulaires verticales en montée et en descente, la limitation angulaire horizontale lors d'un changement de direction et le plafond de navigation (*ceiling*). La phase d'exploration prend en considération toutes ces contraintes afin de trouver une solution faisable. Pour un hélicoptère, le changement de direction est limité entre $[-45^\circ, 45^\circ]$, les angles de changement de palier sont déduits à partir du modèle de performance pour le confort des passagers à bord. Généralement, la pente ne dépasse pas 15° pour assurer la sécurité des passagers.

Pour ce faire, l'algorithme d'exploration opère sur les paires d'arcs. À partir de l'arc courant, nous pouvons calculer l'angle de changement de direction θ par rapport à son parent en projetant les vecteurs associés aux deux arcs sur un plan horizontal comme illustré dans la Fig. 3.13 pour les deux vecteurs $\overrightarrow{AA'}$ et $\overrightarrow{A'A''}$. Le même raisonnement est appliqué pour le calcul des angles verticaux lors d'un changement de palier en montée et en descente. Une autre contrainte (optionnelle) peut être prise en considération : le *ceiling*. Pour des terrains

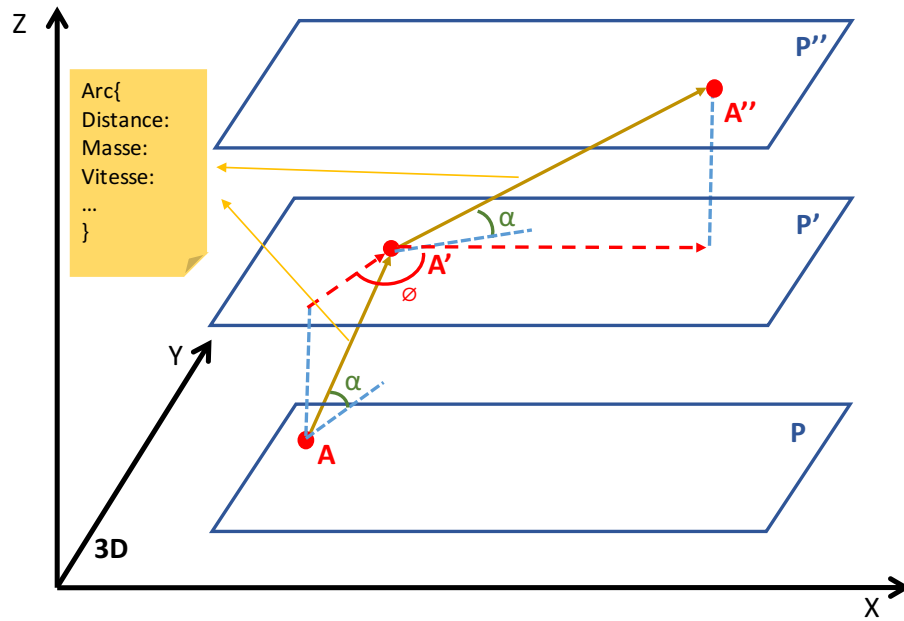


FIGURE 3.13 – Schématisation du calcul géométrique permettant de construire les arcs de raccordement entre les paliers tout en respectant les contraintes angulaires

donnés et pour des conditions météorologiques, la navigation au-dessus d'un plafond donné est interdite afin d'éviter le givrage. Dans notre algorithme, une condition peut être activée, elle permet d'écarter les arcs ayant une altitude supérieure au *ceiling* défini.

Nous avons lancé quatre scénarios sur une machine équipée d'un processeur Intel Core i7-4800MQ (quad-core) cadencé à 2.7 GHz et d'une mémoire RAM de 8 Go offrant des performances équivalentes à 3.47 GFLOPS/core. À chaque exécution, nous avons modifié un seul paramètre parmi RF , angles de *montée/descente* et *ceiling*. Les points de départ et d'arrivée sont les mêmes.

Les Figures 3.14 (a), 3.14 (b), 3.14 (c), 3.14 (d) représentent des captures d'écran issues d'un outil de démonstration en affichant à chaque fois la composante latérale et le profil vertical qui correspondent au plan de vol généré.

Le widget central montre la cartographie du terrain sur laquelle est tracé le plan de vol ainsi que les positions de départ et d'arrivée. Le widget en haut de la fenêtre représente le profil vertical du plan de vol (courbe en vert) et le profil du terrain (courbe en rouge).

Le Tableau 3.2 résume les valeurs des paramètres, les différentes sorties et les durées de calcul relatives à chaque scénario.

► **1^{er} scénario - Fig. 3.14 (a) :** Avec les paramètres suivants : $RF = 1$, angles de montée/descente entre $[4^\circ, 15^\circ]$, pas de ceiling, le prétraitement a pris 151s et nous avons

TABLE 3.2 – Sorties des différents scénarios élaborés

Scénario	RF	Angles	Ceiling	Temps Rou-tage(s)	Temps Prétraitement (s)	Distance (km)	Nombre de points de pas-sage
1	1	$[4^\circ, 15^\circ]$	Pas de ceiling	10	151	141	87
2	2	$[4^\circ, 15^\circ]$	Pas de ceiling	3	175	139	61
3	2	$[10^\circ, 15^\circ]$	Pas de ceiling	1	-	139	69
4	2	$[10^\circ, 15^\circ]$	2000	4	-	142	63

obtenu un plan de vol dans une durée de 10 secondes. La distance du chemin est égale à 141km avec 87 points de passage.

► **2^{ème} scénario - Fig. 3.14 (b) :** Nous avons effectué un nouveau prétraitement en modifiant la valeur du $RF = 2$ au lieu de 1. Le temps de prétraitement est de 175s. Sans modifier ses paramètres, le routage a pris seulement 3s. En effet, le graphe devient plus dense avec des connexions directes ce qui explique la convergence rapide de l'algorithme vers la destination. Il est clair qu'avec un RF plus important, le plan de vol est de meilleure qualité : la distance passe à 139km avec seulement 61 points de passage.

► **3^{ème} scénario - Fig. 3.14 (c) :** Sans procéder à un nouveau prétraitement, nous avons réduit l'intervalle des angles de montée/ descente. Il passe à $[10^\circ, 15^\circ]$. Cette restriction est à l'origine de la génération d'un chemin favorisant le vol en palier comme le montre le profil vertical. Avec le paramètres « angles », l'opérateur peut ajuster le profil du plan de vol selon le type de la mission.

► **4^{ème} scénario - Fig. 3.14 (d) :** Ici, nous avons introduit un plafond égal à 2000m. L'allure du plan de vol a considérablement changé puisque le chemin a contourné les obstacles en horizontal. Le profil vertical reste en palier comme les restrictions sur les angles sont toujours maintenues. Par contre, le chemin devient plus long, il passe à 142km au lieu de 139km.

► **Conclusion :** Ces résultats montrent la flexibilité de notre système pour générer des plans de vol avec différents profils.

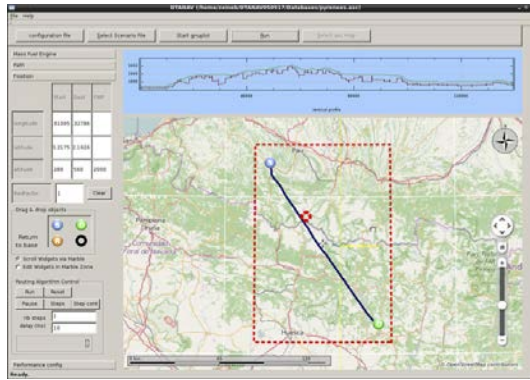
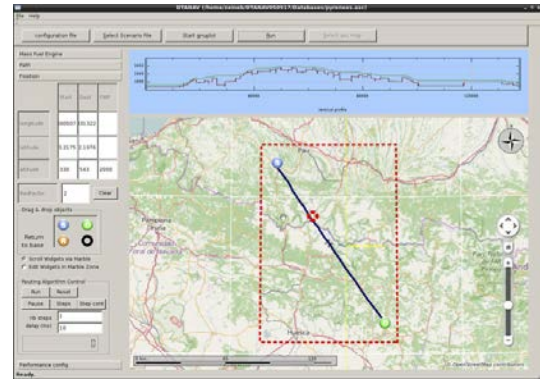
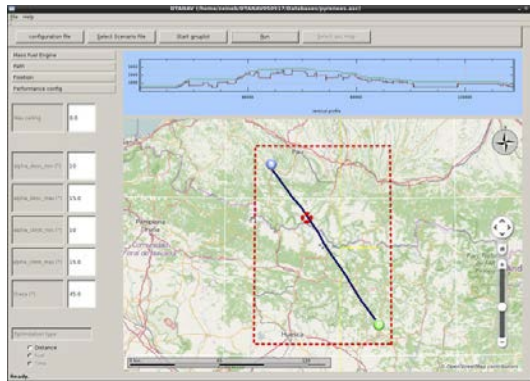
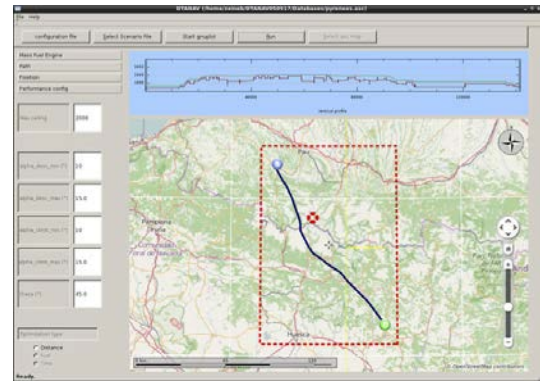
(a) Plan de vol avec un $RF = 1$ (b) Plan de vol avec un $RF = 2$ (c) Plan de vol avec des angles M/D entre 10° et 15° (d) Plan de vol avec un $ceiling = 2000m$

FIGURE 3.14 – Calcul d'un plan de vol avec différents scénarios

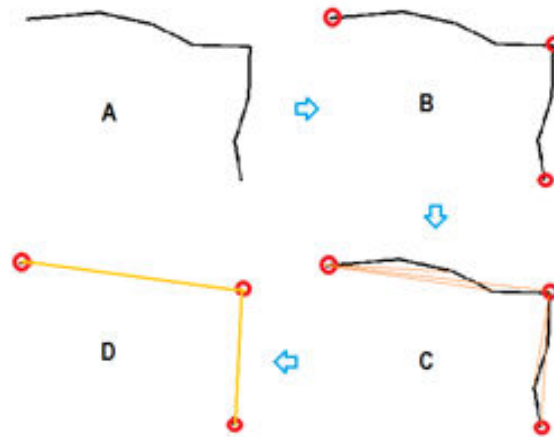


FIGURE 3.15 – Exemple d'étapes de simplification pour améliorer la qualité d'un plan de vol

3.4.4 La phase de simplification du plan de vol

Les plans de vol générés par le procédé de routage décrit précédemment comportent de nombreux segments la plupart du temps quasi alignés dans un plan horizontal et de nombreuses variations de pente dans le plan vertical. Il est nécessaire de réduire le nombre de segments/points tournants jusqu'à obtenir une complexité comparable à celle d'un plan de vol créé manuellement. Cette simplification facilitera la surveillance du vol par le pilote, améliorera le confort en réduisant le nombre de transition et permettra le cas échéant au pilote de manipuler un plan de vol moins complexe s'il souhaite le modifier. La Fig. 3.15 montre un exemple d'étapes de simplification qui pourront être utilisées afin de réduire le nombre de points de passage et trouver des chemins plus directes.

Dans le plan horizontal, en considérant le plan de vol généré (A) (Fig. 3.16), une identification des points tournants majeurs (comportant un changement angulaire supérieur à un seuil) est réalisée (B).

Entre chaque point tournant majeur, les points tournants mineurs sont supprimés tant qu'aucune collision avec le terrain et les obstacles n'est détectée (C). Si lors d'une tentative de suppression d'un point tournant mineur, une collision est détectée, le point mineur en question devient majeur et la recherche de suppression se poursuit en aval de ce dernier. Après avoir parcouru l'ensemble du plan de vol, celui-ci est simplifié par retrait de tous les points tournants mineurs (D).

Dans le plan vertical, le plan de vol généré comporte de nombreuses ruptures de pente, des paliers courts et des descentes suivies immédiatement de montée (A) :

La première étape de la simplification consiste à supprimer tous les paliers courts situés entre deux pentes de même signe (B). Chaque nouveau segment créé étant situé au-dessus des 2 segments supprimés.

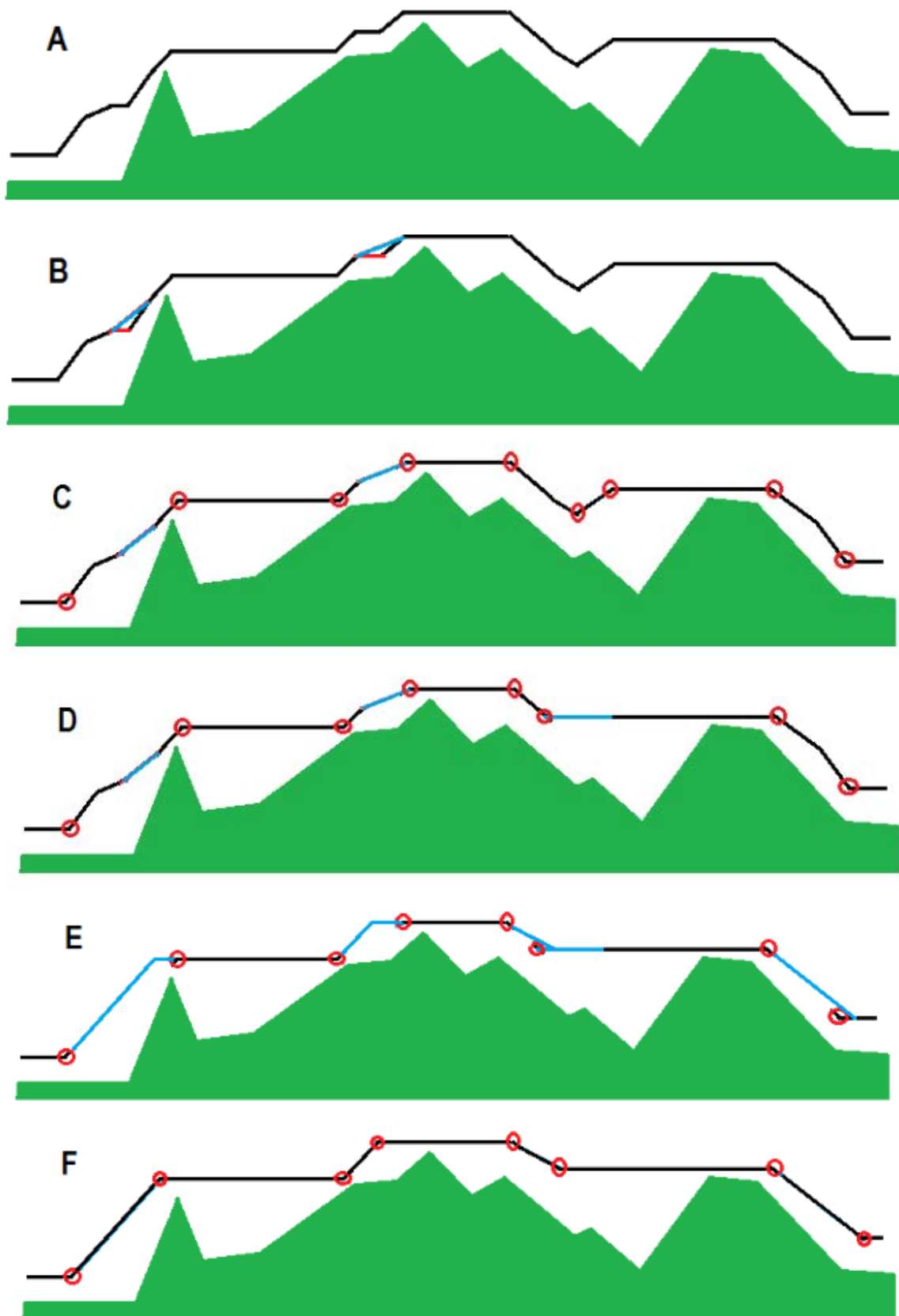


FIGURE 3.16 – Phases de simplification d'un plan de vol généré automatiquement

La seconde étape de la simplification consiste à identifier chaque transition ponctuelle descente suivie de montée et la supprimer (C) et (D) en prolongeant le palier de plus basse altitude jusqu'à joindre le segment de changement de palier opposé.

La troisième étape consiste à supprimer les ruptures de pentes (E) et (F) en calculant les pentes optimales de montée et descente compte tenu des performances de la machine, des conditions météorologiques et du profil de mission sélectionné. Cela entraîne une modification des points de début/fin de montée et des points de début/fin de descente.

Les critères de simplification seront adaptés au profil de la mission et à la fonction coût à minimiser associée.

3.5 Avantages et restrictions

L'invention présente plusieurs avantages :

- Le système peut s'appliquer à la fois en phase de préparation de mission et donc s'intégrer à une station au sol et en cours de vol afin de replanifier la mission face à l'occurrence d'un événement complexe. Dans le deuxième cas, le système sera considéré comme une brique du système d'assistance au pilote dans le but d'apporter plus d'intelligence et d'autonomie à l'appareil ;
- Le système porte sur une approche générique, paramétrable selon le type et les performances de l'appareil, la nature de la mission et de l'environnement ;
- Le système permet de maîtriser la complexité en temps de calcul et en mémoire consommée même pour des terrains étendus ;
- L'opérateur agit sur un nombre limité de paramètres permettant de simplifier l'utilisation du système et de garantir une flexibilité lors du calcul de la solution afin de fournir un profil de vol adapté au type de la mission ;
- Le système donne une évaluation de la faisabilité de la mission pour les cas limites (aide à la décision) avant le départ (et pendant le vol si les paramètres changent) ;
- Amélioration de la performance mission pour les pilotes nouvellement affecté à une région d'intervention ;
- Amélioration de la sécurité des vols en cas de mauvaise visibilité par le calcul de routes à distance de sécurité du terrain ;
- Réduction de la charge de travail « navigation » par mise à jour du plan de vol rapidement en cas d'événement interne (perte d'un moteur par exemple) ou externe en cours de mission communiqués par un radar météo par exemple.

Quelques restrictions peuvent être aussi soulignées :

- Les paramètres du système apportent une grande flexibilité lors du calcul des solutions. Par contre, un mauvais paramétrage lors du lancement du prétraitement peut agir sur la qualité du chemin.
- Dans le cas où aucune solution n'existe, le système continue à explorer le graphe jusqu'à ce qu'il constate l'impossibilité de trouver un chemin.

3.6 Conclusion

Dans ce chapitre nous avons détaillé notre invention décrivant un procédé innovant pour la planification de chemins aériens 3D dédié à la préparation de mission avant le vol ainsi qu'à la replanification dynamique en cours de vol face à une situation d'urgence. Le procédé repose sur une méthode de discrétisation de l'espace 3D, la construction d'un graphe de navigation et un algorithme de routage en prenant en considération un certain nombre de paramètres d'entrée, des contraintes liées à l'environnement et à l'appareil et une fonction objectif à minimiser. Notre approche permet d'ajuster la qualité de la solution en fonction du temps d'exécution disponible en s'appuyant sur un nombre réduit de paramètres. Par ailleurs, le procédé proposé permet d'obtenir différents profils de plan de vol selon le type de la mission. Le système de planification associé à ce procédé peut être déployé sur une station au sol ou bien intégré comme une fonction avionique au sein de l'aéronef. Afin de maîtriser les temps d'exécution, notre système de planification de chemin fonctionne principalement en deux phases : une phase de prétraitement et une phase de routage. La phase de prétraitement permet une discrétisation de l'espace 3D selon une précision donnée et la génération automatique d'un graphe de navigation avec une densité paramétrable. La phase de routage 3D calcule le chemin en prenant en considération un ensemble de contraintes liées à la mission et à l'appareil afin de calculer un chemin aérien selon un critère donné (exemple : minimiser la distance ou le temps de vol). La genericité en fonction du type de l'appareil, le type de la mission et la criticité de la situation est l'attrait principal de notre solution.

Chapitre 4

Méthodes de discrétisation appliquées à la planification de vol

4.1 Introduction

Plusieurs travaux ont démontré l'intérêt lié à l'utilisation des outils de planification de missions pour des aéronefs en termes de réduction des coûts, amélioration de la sécurité en vol et optimisation des performances [11] [67]. La génération automatique de plans de vol repose sur diverses solutions algorithmiques. Parmi les méthodes de planification, nous citons les solutions basées sur des heuristiques. Certaines d'entre elles offrent des résultats satisfaisants avec des temps de calcul réduits [35]. Toutefois, les restrictions imposées par notre domaine d'application (aéronautique) nous ont amenés à nous focaliser sur les méthodes déterministes. Certaines solutions sont déduites de l'application des graphes de visibilité [40] qui élaborent des chemins directs permettant d'éviter les obstacles du terrain. En revanche, leur utilisation est inappropriée à notre contexte comme nous exigeons des plans de vol avec des profils variables selon le type de la mission. En effet, les graphes de visibilité ne respectent pas les contraintes liées à la variation d'altitude de vol. Par conséquent, nous avons réduit l'intervalle des solutions possibles en optant pour des méthodes qui exigent tout d'abord la discrétisation du terrain en 2D ou en 3D et ensuite l'application des algorithmes sub-optimaux de type A* qui sont bien déterministes, avec des temps de calcul bornés sans garantir l'optimalité du chemin calculé. La génération d'un plan de vol 3D qui respecte les limitations de l'aéronef et qui répond aux normes de sécurité (distance par rapport aux obstacles, nombre de points de passage, etc.) demande un choix bien élaboré qui étudie les différentes solutions possibles. Dans ce chapitre, nous allons étudier différentes méthodes de discrétisation permettant un maillage 2D ou 3D de l'espace et nous allons comparer les performances des chemins calculés en explorant ces méthodes.

Ce chapitre sera organisé comme suit : Dans la Section 4.2, nous introduisons notre environnement de travail pour expérimenter plusieurs méthodes de discrétisation de l'espace ainsi

que la comparaison des performances. Dans une première étape, nous nous intéressons aux méthodes de discrétisation et la planification de chemin en 2D (Section 4.3). Par la suite, nous présentons notre algorithme de recherche du plus court chemin dans la Section 4.4. Le passage d'un maillage de l'espace 2D vers 3D sera étudié dans la Section 4.5. Plusieurs techniques seront présentées avec une étude comparative en termes de complexité spatiale et temporelle. La Section 4.6 offre une synthèse de ce chapitre ainsi que des conclusions sur les choix scientifiques pour répondre à notre problème industriel.

4.2 Étude de scénarios réels et critères d'évaluation

Dans notre travail, nous projetons l'étude des méthodes de discrétisation et de planification de chemin sur des scénarios réels. Ces scénarios représentent des missions de vol de différents types (recherche et sauvetage, assistance médicale, VIP, etc.) sur des terrains de notre globe y compris la France. Dans le chapitre précédent, nous avons présenté notre procédé de planification de chemin pour des aéronefs de façon générale. Dans ce chapitre, les exemples étudiés seront destinés pour des hélicoptères vu notre partenariat industriel avec Airbus Helicopters. Des contraintes spécifiques à l'appareil seront considérées tels que les angles de montée, de descente ou de changement de direction.

La Fig. 4.1 décrit un exemple de terrain des « Pyrénées » de taille $97473m \times 139854m$. Ce terrain étendu qui représente un relief hétérogène permet d'évaluer la complexité spatiale et temporelle de notre problème. À partir d'une zone géographique donnée (quadrillage rouge dans la Fig. 4.1), notre première étape consiste à l'extraction des informations numériques sous forme d'une matrice d'élévation avec une précision de l'ordre de dizaines de mètres sur les axes X et Y. En conséquence, à chaque coordonnées (x, y) de la matrice correspond une altitude z . Cette altitude est exprimée en mètre (format ArcInfo ASCII Grid dont le fichier est d'extension .ASC). Nous allons considérer également deux autres exemples de terrains : « Grenoble » et « Andorre » dont la taille est respectivement de $120733m \times 171001m$ et $8184m \times 11118m$.

Pour comparer quantitativement les différentes méthodes, plusieurs mesures sont utilisées : le temps d'exécution, l'occupation mémoire, la qualité de la solution en termes de distance et la satisfaction des contraintes dynamiques de l'appareil, etc. Pour des missions spécifiques, d'autres critères peuvent être considérés tel que le nombre de changements de paliers pour une mission VIP. Afin de démontrer la faisabilité de nos solutions, les méthodes et les algorithmes explorés dans ce chapitre seront exécutés sur une machine dont les performances sont comparables à un ordinateur embarqué (performance du processeur et capacité mémoire). Pour cela, nous avons utilisé une machine équipée d'un processeur Intel Core i7-4810MQ (quad-core) cadencé à 2.8 GHz et d'une mémoire RAM de 16 Go offrant des performances équivalentes à 2.96 GFLOPS/core. Différents outils de visualisation en 2D et 3D nous aiderons à vérifier

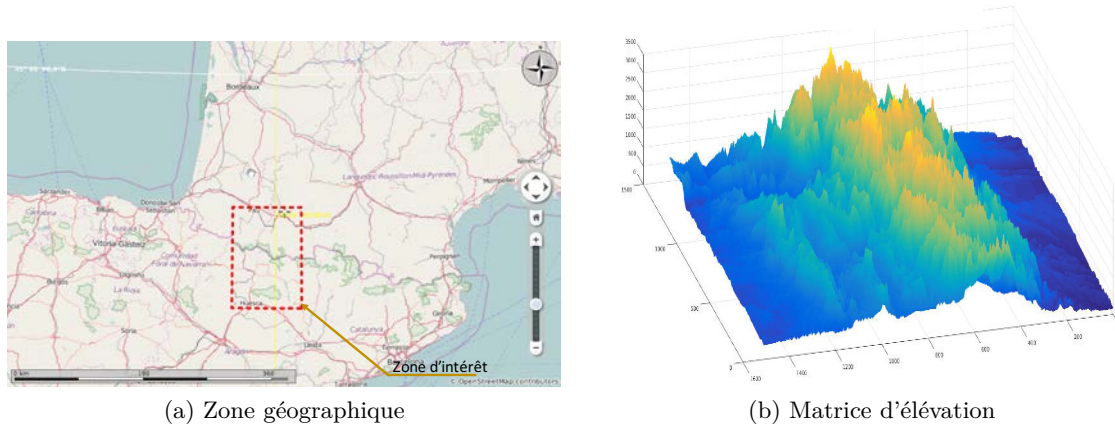


FIGURE 4.1 – Exemple de terrain : les Pyrénées

la qualité des solutions générées.

4.3 Discrétisation 2D : Quadtree et Grille régulière

Afin de maîtriser la complexité des solutions de planification de chemins aériens basées sur une discrétisation de l'espace en 3D, nous entamons dans une première étape l'étude de complexité des méthodes de discrétisation 2D. En effet, les méthodes de planification 3D existants présentent des limitations liées à la complexité algorithmique, la précision, la violation des contraintes avioniques, etc. Certaines de ces limitations seront évoquées dans la Section 4.5.

Afin de pallier ces problèmes, nous avons conçu une nouvelle approche de planification de chemins aériens en 3D qui repose sur une méthode de modélisation multi-altitude 2D avec un maillage adaptatif et la construction d'un graphe de navigation 3D. Sa description a été détaillée dans le Chapitre 3. Dans une première étape, nous avons procédé à une étude de complexité de quelques solutions 2D. Cette étude que nous abordons nous aidera dans la suite à déterminer la faisabilité de l'approche proposée.

Dans cette section, nous comparons les performances de notre approche adoptant une discrétisation adaptative de l'espace par rapport au maillage régulier. Notre approche est basée sur deux phases :

1. Le prétraitement (phase statique) qui se décline en deux étapes :
 - la génération du Quadtree : nous décrivons le maillage adaptatif 2D et son application à notre recherche,
 - la construction du graphe : nous expliquons notre méthode de construction du graphe de navigation à partir du maillage en Quadtree. Nous évoquons par la suite l'impact de cette approche sur la phase dynamique.

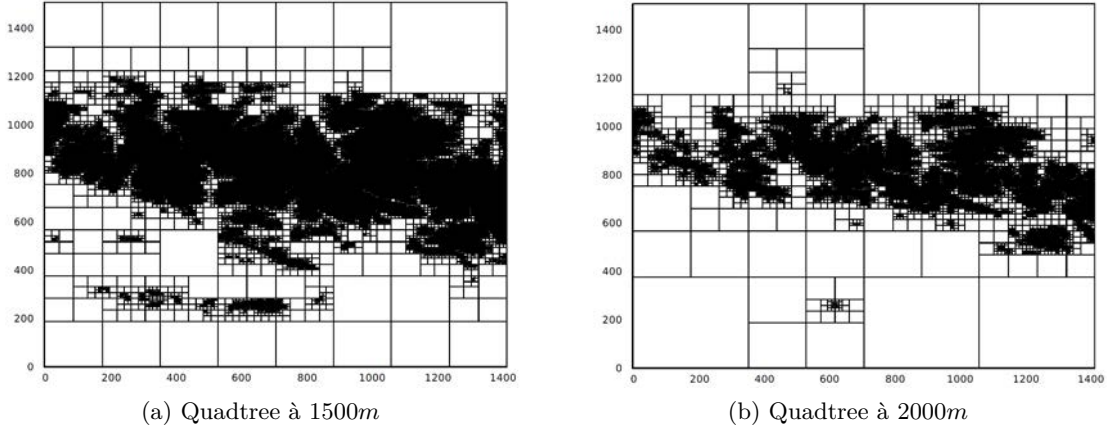


FIGURE 4.2 – Quadtrees projetés sur la carte des « Pyrénées » à des altitudes 1500m et 2000m

2. Le routage (phase dynamique) : nous appliquons notre algorithme de recherche du plus court chemin sur les graphes générés à partir d’une grille régulière 2D et du Quadtree afin de comparer les performances de chacune des méthodes de discrétisation de l’espace.

4.3.1 Le Quadtree

Selon [49], un Quadtree est une décomposition récursive d’une image bidimensionnelle en blocs de 2×2 uniformément colorés (voir Fig. 4.2). Il s’agit d’une structure binaire. Chaque nœud libre représente une région d’espace libre. Un nœud obstacle est un nœud représentant une région d’obstacles ou une région hétérogène. Dans notre recherche, nous étudions les Quadtrees linéaires ce qui signifie que nous allons traiter uniquement les feuilles de la structure (cases libres) sans considérer les parents. Par ailleurs, les centres de ces cases libres représenteront les nœuds du graphe de navigation que nous allons construire (Section 4.3.2).

Nous comparons les performances de ce maillage par rapport aux grilles uniformes. Ces dernières ont fait l’objet de plusieurs travaux de planification de chemin [73] [22]. En effet, si nous appliquons une discrétisation régulière d’une résolution de 100m par exemple sur le terrain des « Pyrénées » de taille $97473 \times 139854m$, nous obtenons une grille avec un nombre de nœuds libres de l’ordre de 2 millions. Ce type de maillage engendre une complexité temporelle et spatiale importante ce qui empêche son application dans notre contexte industriel. Ce dernier doit répondre à des contraintes temporelles liées à la criticité de l’évènement nécessitant la planification rapide d’une mission (exemple mission *recherche et sauvetage*) ou la replanification en cours de vol pour atterrir en toute sécurité.

Pour pallier aux problèmes liés à un maillage régulier, plusieurs travaux [67] [72] [29] optent pour le maillage adaptatif de type Quadtree idéal pour les problèmes de « path planning » de longue distance. En effet, cette structure permet de réduire le nombre de nœuds et par conséquent réduire la complexité spatiale et temporelle de la phase de prétraitement. À

titre d'exemple, l'application d'un Quadtree sur le terrain des « Pyrénées » à une altitude de 2000m génère 15837 nœuds libres (soit une réduction de 99%). Cette représentation est adaptative selon la densité des obstacles ce qui fait que ce type de maillage est très avantageux à des altitudes élevées pour lesquelles les obstacles sont peu abondants. Ceci est traduit dans l'exemple décrit dans la Fig. 4.2. Elle montre un Quadtree projeté sur le terrain des « Pyrénées » à des altitudes de 1500m et 2000m. Ainsi, si nous considérons une grille de taille $n \times n$, la complexité du maillage passe de $O(n^2)$ à $O(n \times \log(n))$. Cette complexité peut être encore réduite si nous utilisons que les cases libres qui représentent les feuilles dans cette structure. En effet, Kambhampati et al. [29] prouvent que le nombre des nœuds feuilles dans cette structure est de l'ordre de $2/3 \times O(p)$ où p est égale au périmètre de l'ensemble des obstacles polygonaux présents dans la région étudiée.

Pour le prétraitement, cette structure adaptative est intéressante en termes de consommation mémoire et temps de calcul particulièrement pour les terrains ayant une densité d'obstacles hétérogène. D'un point de vue planification de chemin, il est démontré que les méthodes de planification de chemin multi-résolution sont plus efficaces par rapport aux méthodes conçues avec des grilles régulières [49]. En effet, la complexité d'exploration d'un Quadtree est de l'ordre de $O(n * \log(n))$ [49] pour une région de taille $n \times n$ discrétisée en unité de plus basse résolution (dans notre cas, les points de la matrice d'élévation) ce qui représente une réduction substantielle du temps de prétraitement. Dans notre recherche, nous nous sommes inspirés de la planification de chemin avec un maillage en Quadtree : nous avons introduit une nouvelle méthode de planification de vol à travers l'exploration d'un graphe de navigation. Nous construisons le graphe suite à la génération du Quadtree. Il permet de relier entre plusieurs nœuds navigables avec un degré de connexité variable ce qui permet d'ajuster la qualité de la solution selon le temps disponible pour le prétraitement. Par conséquent, cette nouvelle méthode de planification offre un bon compromis entre les algorithmes d'espace libre (Graphe de visibilité) et les méthodes de recherche de chemin basées sur des grilles régulières. D'ailleurs, les chemins obtenus via cette approche multi-résolution, bien que non « optimaux » sont « efficaces » et calculés dans des durées réduites et bornées, ce qui répond aux besoins de notre contexte industriel.

4.3.2 Construction du graphe de navigation

► **Quadtree** Dans la construction du graphe, nous avons introduit la notion d'adaptativité à travers un paramètre que nous avons appelé *Radius Factor*. Ce paramètre permet d'ajuster le degré de connexité du graphe de navigation. Ce graphe est défini par l'ensemble des nœuds qui représentent les centres des cases du Quadtree généré au préalable et l'ensemble des arcs établissant des liaisons entre ses nœuds. Construit lors du prétraitement, notre graphe présente deux propriétés importantes qui ont un impact considérable lors de la phase dynamique (calcul du plan de vol) :

- il permet de connecter des nœuds éloignés ce qui a tendance à améliorer la qualité des chemins obtenus lors du routage,
- il intègre un traitement de détection d'obstacles appelé lors de la construction des arcs. Ce traitement permet d'éviter les calculs supplémentaires dans la phase dynamique.

Pour construire le graphe de navigation il faut :

- disposer de l'ensemble des nœuds : le centre de chacune des cases du Quadtree généré au préalable représente un nœud du graphe,
- fixer le *Radius Factor* à une valeur réelle entre 1 et 7. Ce paramètre permet d'ajuster le degré de connexité du graphe,
- faire appel à l'algorithme de *Bresenham* [9] pour assurer la détection d'obstacles. L'algorithme détermine les points d'un plan discret qui doivent être tracés afin de former une approximation de segment de droite entre deux points donnés. Autrement dit, en se basant sur la matrice d'altitude, cet algorithme détecte les éventuels obstacles sur le segment entre deux points donnés du terrain.

L'algorithme de construction du graphe est itératif. Il procède comme suit : Pour chaque nœud i , $i \in [1, \dots, N]$ correspondant à une case Q_i du Quadtree où N représente le nombre de nœuds dans le graphe :

- Le *Radius Factor* (noté RF) est multiplié par la diagonale D de Q_i , ce qui permet d'obtenir la zone de voisinage de notre case sous forme d'un cercle de rayon R où $R = D \times RF$.
- Pour chaque nœud k du graphe, la distance d entre les nœuds i et k est calculée. Si $d > R$ alors le nœud k est écarté, sinon : l'algorithme de Bresenham est appliqué entre les nœuds i et k . En cas d'absence d'obstacles, l'arc reliant les deux nœuds est alors construit.

De ce processus résulte l'ensemble des arcs du graphe et donc l'espace navigable de notre terrain 2D. Ainsi, nous obtenons le graphe de navigation 2D qui sera par la suite considéré comme la sortie du prétraitement.

La Fig. 4.3 traduit l'impact du paramètre *Radius Factor* sur le degré de connexité du graphe. En appliquant un *Radius Factor* de valeur relativement élevée, le graphe obtenu est plus riche en arcs ce qui implique l'amélioration de la qualité des solutions trouvées lors de la phase de routage. Cependant cette connexité s'avère coûteuse en termes de temps de calcul et de mémoire consommée d'où l'importance de bien choisir la valeur qui offre un bon compromis entre le temps disponible pour le prétraitement et la qualité de chemins souhaitée.

► **Grille régulière 2D** Afin de pouvoir comparer le Quadtree et la grille régulière, nous avons appliqué l'algorithme de construction du graphe sur la structure régulière 2D en fixant le RF à 1.5. Nous avons choisi cette valeur pour couvrir les voisins proches d'un nœud

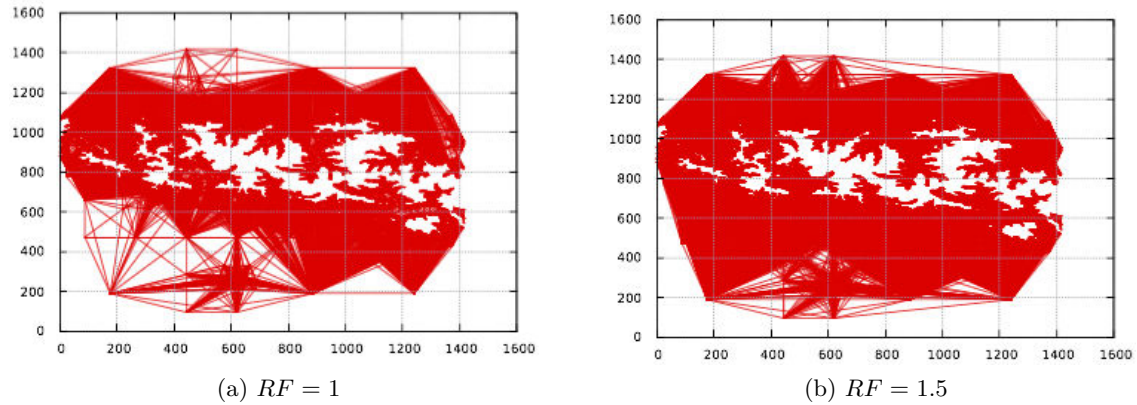


FIGURE 4.3 – Graphes générés à une altitude de 2000m pour $RF = 1$ et $RF = 1.5$ (terrain des « Pyrénées »)

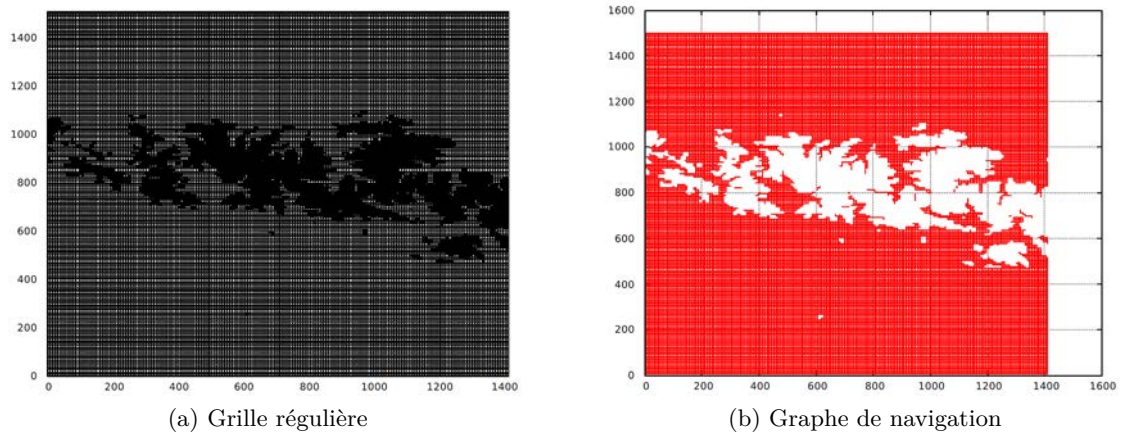


FIGURE 4.4 – Maillage régulier et graphe générés à une altitude de 2000m (terrain des « Pyrénées »)

donné, au-delà de cette valeur le nombre d'arcs générés explose et la complexité ne peut pas être maîtrisée. Ainsi, nous avons obtenu un graphe issu de la grille régulière. Comme pour le cas des Quadtrees, ce graphe est présenté par un ensemble de nœuds et d'arcs. Toutefois, le traitement de détection d'obstacles n'est pas nécessaire pour une grille régulière ce qui introduira une réduction des durées de calcul par rapport au Quadtree. La Fig. 4.4 montre une représentation régulière de la carte des « Pyrénées » à une altitude de 2000m ainsi que le graphe de navigation associé. En comparant cette structure régulière par rapport au maillage en Quadtree représenté dans Fig. 4.2 et Fig. 4.3, nous constatons la réduction de complexité introduite par la structure adaptative. En effet, le graphe de navigation ayant un degré de connexité ajustable construit sur un maillage adaptatif selon la densité des obstacles offre une représentation riche de l'espace de configuration. Ceci introduit un gain important au niveau des durées de calcul et la qualité des plans de vol élaborés lors de la phase dynamique.

4.3.3 Résultats

4.3.3.1 La phase de prétraitement

Nous avons appliqué notre approche de planification de vol sur trois terrains : « Grenoble », « Pyrénées » et « Andorre » de tailles différentes. Les cartes présentent des reliefs montagneux. Pour un premier scénario, nous avons choisi la carte des « Pyrénées » de dimensions $97473m \times 139854m$ à une altitude de $2000m$. Le maillage régulier génère 1812992 nœuds libres (avec la résolution la plus basse : $68.79m \times 92.62m$). D'après nos expérimentations, la construction du graphe de navigation correspondant à ce maillage fin déborde en temps de calcul et en consommation mémoire vu la taille de la structure. Nous précisons que dans l'exemple du graphe généré dans la Fig. 4.4 nous avons diminué la précision de 8 fois par rapport à la résolution de base de notre carte afin de maîtriser la complexité en temps et en mémoire. Cependant, cette résolution ne répond pas par la suite aux exigences de notre contexte en particulier pour construire un graphe 3D avec des angles de montée et de descente convenables.

Par ailleurs, le maillage en Quadtree génère 15837 nœuds (voir Tableau 4.1). Les tailles des cases varient entre $68.79m \times 92.62m$ et $24351.66m \times 34917.74m$. Cette réduction du nombre de nœuds s'explique par la faible complexité spatiale du Quadtree. Nous soulignons que la taille de la plus grande case du Quadtree généré n'est pas également adéquate pour notre contexte, pour cette raison nous allons utiliser des Quadtrees de type *Accurate* comme introduit dans le chapitre précédent. Pour la carte des « Pyrénées » et avec un *Radius Factor* de 1.5, la structure Quadtree et le graphe sont générés au bout de 52.863s. Le Tableau 4.1 montre entre autres la variation de la sortie du prétraitement selon la taille de la carte étudiée. La mémoire consommée (le nombre d'arcs et de nœuds) et la durée du prétraitement évoluent selon la taille du terrain.

Ensuite, nous avons fait varier la valeur du *Radius Factor* de 1.5 à 4 afin de mesurer l'impact de ce paramètre sur la taille du graphe et le temps de prétraitement. Le Tableau 4.2 rapporte les résultats obtenus pour la carte des « Pyrénées » à une altitude $2000m$. Nous avons remarqué que le nombre d'arcs ainsi que le temps de prétraitement varient de façon « quasi-linéaire » en fonction du *Radius Factor* comme illustré sur la Fig. 4.5. Nous considérons que ce résultat est très satisfaisant car il nous permet de maîtriser la complexité temporelle et spatiale en fonction de la connexité de notre graphe (à travers le *Radius Factor*). Néanmoins, nous avons besoin de vérifier ce résultat avec une analyse approfondie du terrain et de la densité d'obstacles.

Pour faire cette analyse, nous allons considérer le plus grand terrain « Grenoble » et étudier le temps de prétraitement en fonction de l'altitude et du *Radius Factor*. Il est à noter que le temps de prétraitement inclut le temps de génération de la structure Quadtree et le temps de génération du graphe. Pour cette raison, nous avons tout d'abord mesuré le temps de génération d'un Quadtree pour les différents terrains étudiés et altitudes et nous avons constaté que les temps mesurés sont toujours inférieurs à $50ms$. Par conséquent, le temps de

TABLE 4.1 – Prétraitement 2D avec un $RF = 1.5$ à une altitude $2000m$

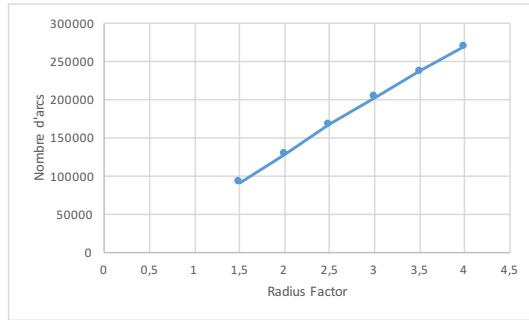
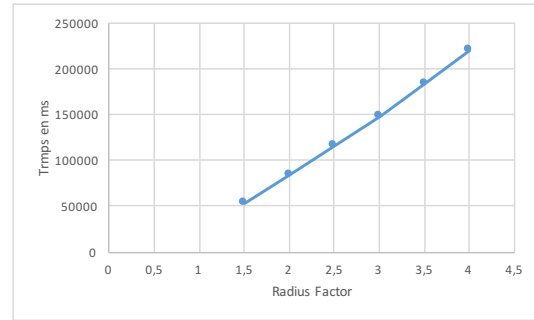
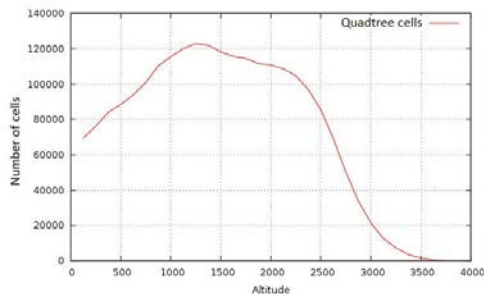
	Pyrénées	Grenoble	Andorre
Taille ($m \times m$)	$97473m \times 139854m$	$120733m \times 171001m$	$8184m \times 11118m$
Nombre de nœuds RG/ Nombre d'arcs	1812992/Non déterminé	3909632/Non déterminé	7598/43943
Nombre de nœuds Quadtree / Nombre d'arcs	15837/90549	33003/151719	235/1068
Durée de prétraitement RG2D (secondes)	débordement mémoire	débordement mémoire	1.702
Durée de prétraitement Quadtree (secondes)	52.863	170.7	0.009
Précision RG (m)	68.79×92.62	66.37×92.63	81.84×111.18
Précision Quadtree (m) : plus petite case	68.79×92.62	66.37×92.63	81.84×111.18
Précision Quadtree (m) : plus grande case	24351.66×34917.74	30198.35×42609.8	2046×2779.5

TABLE 4.2 – Prétraitement appliqué sur la carte des « Pyrénées » à une altitude $2000m$

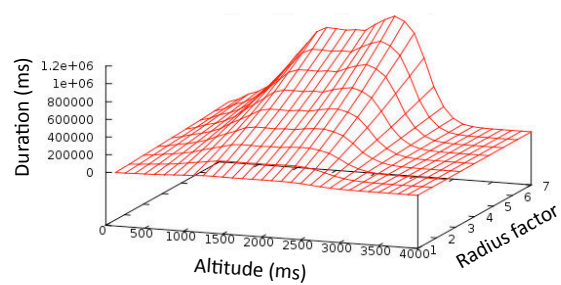
RF	1.5	2	2.5	3	3.5	4
Nombre de nœuds	15837	15837	15837	15837	15837	15837
Nombre d'arcs	140867	207786	267457	327607	377464	426218
Temps de prétraitement (s)	53.6	83.8	116.2	147.7	183.3	219.6

génération du graphe s'avère le plus pénalisant, en particulier, l'opération de détection d'obstacles pour relier deux points éloignés. Le temps de génération du graphe dépend du nombre de cases libres générés par la structure Quadtree pour une altitude donnée. Comme nous l'avons mentionné, le Quadtree engendre un maillage adaptatif selon la densité d'obstacles. La Fig. 4.6 (a) illustre le nombre de cases obtenu en fonction de l'altitude, nous commençons par un nombre réduit de cases pour des altitudes basses (que des obstacles à ces altitudes à cause du relief de la région étudiée) ensuite ce nombre augmente pour des altitudes plus importantes (hétérogénéité entre obstacles et espaces libres), et pour des altitudes plus proche de l'altitude maximale, le nombre de cases décroît (très peu d'obstacles) jusqu'à devenir nul une fois cette valeur dépassée ($Altitude_{max}$ du terrain).

La Fig. 4.6 (b) (appelée « surface de réponse ») illustre le temps de construction du graphe en fonction de l'altitude et du *Radius Factor*. Cette figure montre, selon l'axe des abscisses, une dépendance entre le temps de calcul et l'altitude liée à la « topographie » du terrain qui impacte à son tour le taux de compression du Quadtree. Parallèlement, le temps d'exécution

(a) Nombre d'arcs en fonction du RF (b) Temps de prétraitement en fonction du RF FIGURE 4.5 – Analyse du nombre d'arcs et du temps de prétraitement en fonction du RF (carte des « Pyrénées » à une altitude 2000m)

(a) Nombre de cases en fonction de l'altitude



(b) Analyse du temps de génération de graphe

FIGURE 4.6 – Temps de génération de graphe en fonction de l'altitude et du *Radius Factor* pour le terrain de Grenoble

augmente en fonction du nombre de cases dans la structure Quadtree en cohérence avec ce que nous avons illustré dans la Fig. 4.6 (a). Un résultat a été mis en évidence sur toutes les surfaces de réponse générées pour les cartes utilisées : Le temps de calcul évolue de manière « quasi-linéaire » en fonction de la valeur du *Radius Factor*.

4.3.3.2 La phase de routage

Tandis que le prétraitement s'effectue lors de la phase de préparation de mission, la phase de routage peut être lancée une ou plusieurs fois avant ou en cours de vol. Elle requiert le graphe de navigation afin de pouvoir calculer un plan de vol selon les paramètres de la mission et les contraintes de l'appareil. Pour la phase de routage, nous comparons deux plans de vol générés sur la carte des « Pyrénées » à une altitude de 2000m entre les points situés au « Nord-Ouest » et au « Sud-Est » de la carte en appliquant les deux structures de discrétisation étudiées : le Quadtree et la grille régulière 2D. L'objectif est de comparer la qualité du chemin généré par notre approche (discrétisation Quadtree + génération de graphe

2D) et celle fournie par l'approche classique en s'appuyant sur une grille régulière. Pour ce faire, nous avons utilisé une fonction A^* de la bibliothèque *Boost* qui sera appliquée sur les graphes issus des deux méthodes de discrétisation.

Comme nous l'avons déjà mentionné, nous avons réduit la précision du maillage régulier de $68.79m \times 92.62m$ à $550.3m \times 740.96m$ pour réduire la complexité en temps et en mémoire. Le graphe généré en 18s comprend 29060 nœuds et 166766 arcs. Concernant le graphe de navigation issu du Quadtree en appliquant un *Radius Factor* de 1.5, il est généré en 53.6s et comprend 15837 nœuds et 90549 arcs.

Les résultats numériques concernant le nombre de segments, la distance et le temps de routage sont rapportés par le Tableau 4.3. La solution basée sur la structure Quadtree offre un chemin de meilleure qualité par rapport à celle basée une grille régulière en termes de distance et de nombre de segments. En effet, la première offre un chemin 19% plus court que la deuxième au bout de la même durée de calcul. Le nombre de points de passage générés par l'approche classique (137) est très important ce qui complique la phase de postprocessing nécessaire pour aboutir à un chemin plus lisse. Par contre, notre approche permet d'obtenir des chemins plus directs par rapport à la grille régulière c'est à dire qui convergent rapidement vers la destination avec un nombre réduit de points de passage (9 pour notre exemple). Inspirée des méthodes de planification de chemins *Any Angle*, notre solution permet de trouver des voisins plus éloignés du fait que le graphe a été construit avec un voisinage variable. La Fig. 4.7 illustre la qualité des chemins générés. L'approche classique ne permet de visiter que les voisins proches (nœuds adjacents) et le nombre de nœuds très important ne permet pas de considérer un voisinage plus élevé vu la complexité spatiale et temporelle correspondante.

TABLE 4.3 – Comparaison de performances en explorant un graphe issu d'une grille régulière et d'un Quadtree (terrain des « Pyrénées » à une altitude 2000m)

	Grille régulière	Quadtree avec $RF = 1.5$
Nombre de segments	137	9
Distance (m)	160161	130506
Temps de routage (ms)	361	352

La dernière expérimentation rapportée par le Tableau 4.4 analyse les performances de la phase de routage et la qualité du chemin généré par notre approche en fonction du *Radius Factor*. Nous avons construit le graphe de navigation pour des valeurs de *Radius Factor* variant de 1 à 3.5. Ensuite, nous avons choisi les mêmes points de départ et d'arrivée sur la carte des « Pyrénées » à une altitude de 2000m et nous avons affiché le chemin sur le Quadtree.

En effet, l'augmentation du *Radius Factor* permet de générer un graphe avec un degré de connexité plus élevé. Ceci se traduit par un nombre plus important d'arcs et des connexions plus directes entre les différents nœuds (voir Fig. 4.3). La connexité du graphe a un impact direct sur la qualité du chemin trouvé. Ceci est le cas de notre exemple en passant d'un *Radius Factor* de 1 à 1.5, nous avons obtenu un chemin plus court et avec un nombre de points

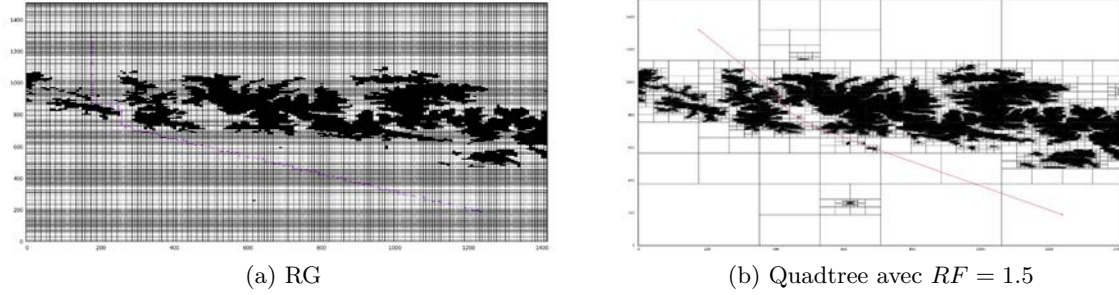


FIGURE 4.7 – Affichage du chemin généré en explorant une grille régulière et un Quadtree (carte des « Pyrénées » à une altitude de $2000m$)

de passage réduit en passant de 20 à 9 points au détriment du temps d'exécution (graphe plus dense). Les deux chemins figurant sur Fig. 4.8 (a) et (b) illustrent l'amélioration de la qualité du chemin en fonction du *Radius Factor*. Toutefois, à partir d'une certaine valeur cette amélioration devient inaperçue avec un graphe très connexe ce qui explique les performances obtenues en passant d'un *Radius Factor* de 1.5 à 2.5 et 2.5 à 3.5. Sur la Fig. 4.8 (c), il est difficile de détecter l'amélioration de la solution ($108m$) en comparaison avec le chemin de la Fig. 4.8 (b). Par ailleurs, nous remarquons que le temps de routage est plus important pour les graphes construits avec des *Radius Factor* élevés. Ces durées restent relativement réduites ($< 1s$) et bornées en considérant des graphes de l'ordre de centaines de milliers d'arcs.

TABLE 4.4 – Qualité du chemin généré et temps de routage en fonction du *Radius Factor* (terrain « Pyrénées » à une altitude $2000m$)

RF	1	1.5	2.5	3.5
Nombre de segments	20	9	9	9
Distance (m)	132488	130506	130398	130398
Temps de routage(ms)	282	352	489	543

Dans cette section, nous avons souligné les avantages de notre approche adaptative de planification de chemins en 2D. Principalement, nous avons proposé une solution qui permet de maîtriser la complexité spatiale et temporelle et ceci pour les deux phases : le prétraitement et le routage. La qualité des solutions générées peut être paramétrée selon le temps d'exécution disponible. Les résultats obtenus par notre approche 2D constituent l'argument principal pour justifier notre solution multi-altitude en considérant l'espace 3D.

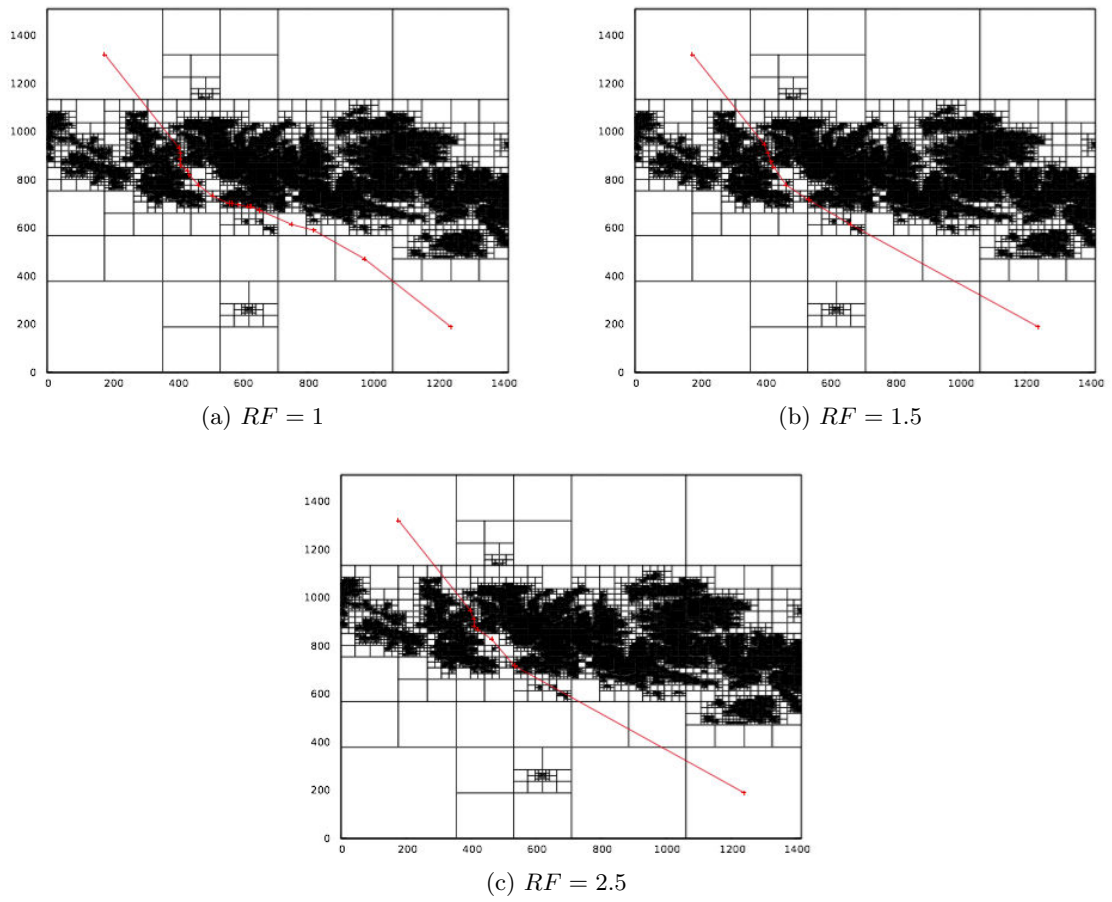


FIGURE 4.8 – Affichage du chemin généré en explorant un graphe de navigation avec $RF = 1$, $RF = 1.5$ et $RF = 2.5$ (carte des « Pyrénées » à une altitude 2000m)

4.4 Algorithme de recherche du plus court chemin : Any-angle adaptive A*

Dans le Chapitre état de l'art nous avons exposé le fonctionnement de l'algorithme A* et ses variantes (Theta*, D*, etc.) dont les propriétés pourront répondre à notre problématique (*Any-angle*, *Dynamique*, etc.). En effet, la phase dynamique représente un élément crucial afin d'explorer le graphe de navigation et trouver une solution qui répond aux exigences de notre domaine. En particulier, nous avons besoin d'un algorithme adaptatif pour considérer différentes contraintes liées au profil du plan de vol et générer une solution qui minimise un critère donné. Pour toutes ces raisons, nous avons développé notre propre algorithme de recherche du plus court chemin. Dans ce qui suit, nous allons détailler notre solution algorithmique en soulignant ses apports en comparaison avec quelques variantes de A*.

1. Nous allons appliquer notre algorithme sur le graphe 3D et non pas sur une grille régulière ou irrégulière. En effet, ce graphe est généré en respectant les limitations angulaires en montée et en descente de l'appareil ainsi que l'évitement de collision avec le terrain. Cette propriété va nous permettre d'accélérer le calcul du chemin lors de la phase dynamique du procédé de planification.
2. Le voisinage d'un nœud donné est variable selon le *Radius Factor*, ce qui nous permet d'obtenir des liaisons directes entre deux nœuds éloignés. le bon paramétrage du *Radius Factor* nous permettra d'obtenir une qualité de solution meilleure que celle du Theta*.
3. Les arcs du graphe sont construits en respectant les limitations angulaires de l'appareil (à titre d'exemple un intervalle de $[1^\circ, 15^\circ]$) donc nous pouvons qualifier notre algorithme de type *Any-angle*
4. L'algorithme que nous proposons est adaptatif, nous pouvons ajouter un ensemble de contraintes liées à la mission tel que le plafond de navigation (*ceiling*).
5. L'exploration du graphe ne s'applique pas sur les nœuds mais sur les arcs afin de respecter certaines contraintes et évaluer différents critères (temps, carburant, etc.).

Nous commençons par présenter un pseudo-code de l'algorithme A* comme présenté dans la littérature. Celui-ci correspond à l'Algorithme 1. Il s'exécute sur un graphe $G = (V, A)$ entre un nœud source $s \in V$ et une destination $t \in V$. Chaque nœud v aura trois étiquettes $g(v)$, $f(v)$ et $parent(v)$. La première $g(v)$ contient la plus petite distance connue de v à la source s . La seconde $f(v)$ contient la distance approximative du chemin partant de s , passant par v et se rendant à la destination t . La troisième $parent(v)$ contient le nœud qui a été utilisé pour se rendre à v dans le plus court chemin de s à v . Au fur et à mesure que l'algorithme itère sur les nœuds, il mettra à jour ces étiquettes. À la fin, $g(t)$ devrait contenir la distance la plus courte, selon A*, de s à t . Le plus court chemin est obtenu en suivant la liste des parents de t jusqu'à s .

Nous aurons aussi besoin d'une file de priorité L . Cette file contient des nœuds ordonnés selon leur priorité $f(v)$. Nous utilisons un tas binaire comme structure de données. La file possède les méthodes suivantes :

- *Ajouter*(v, x) : ajoute le nœud v à la file et la priorité du nœud est x ;
- *ObtenirMin*() : retourne le nœud dont la priorité est la plus petite ;
- *Retirer*(v) : retire le nœud v de la file ;
- *MettreAJour*(v, x) : met à jour la priorité du nœud v avec la nouvelle priorité x .

La méthode débute par initialiser toutes les valeurs à l'infini sauf pour le nœud de départ s . Celui-ci est ajouté à la file avec une priorité de zéro et il sera sélectionné à la première itération de l'algorithme. À chaque itération de la méthode, nous récupérons le nœud v ayant la plus petite priorité. Nous regardons chaque voisin w pour voir s'il existe un nouveau chemin passant par v pour se rendre à w et dont la distance est plus petite. Cette distance se mesure en utilisant la meilleure distance pour se rendre à $g(v)$ et passer par l'arc (v, w) . Si la meilleure distance jusqu'à présent pour w est plus longue que $g(v) + d_{vw}$, alors nous mettons à jour $g(w)$, le parent de w et w est ajouté ou mis à jour dans la file de priorité. L'algorithme itère ainsi jusqu'à ce que t devienne le nœud dont la priorité est la plus petite où que la file soit vide. Si t devient choisis, alors l'algorithme s'arrête, un chemin pour se rendre de s à t a été trouvé.

L'algorithme A^* tel quel ne peut être utilisé pour notre application. Dans notre problématique, l'appareil doit respecter des contraintes angulaires pour des changements de direction. Un exemple est montré dans la Fig. 4.9. Supposons que l'appareil se trouve au nœud 1 et qu'il emprunte l'arc a pour se rendre au nœud 0. Il ne pourra pas se diriger vers les nœuds 2 et 4, car ceux-ci forceraient l'appareil à faire un changement de direction beaucoup trop brusque, ce qui est physiquement impossible. Plusieurs aspects de l'algorithme A^* ont dû être changés radicalement pour tenir compte des changements de direction.

Le premier changement est pour tenir compte de l'angle courant de l'appareil. De manière classique, les algorithmes de plus courts chemins, comme A^* et Dijkstra, itèrent de nœud à nœud. Il est possible d'avoir l'angle de l'appareil à un nœud donné en regardant son nœud parent. Puisque chaque nœud est visité une seule fois, la façon d'atteindre ce nœud se fera à l'aide d'un seul chemin. Tous les arcs engendrant un fort changement de direction seront ignorés. Il est alors possible de perdre des chemins réalisables. Par exemple, si nous sommes arrivés au nœud 0 par le nœud 1, alors les nœuds 2 et 4 deviennent inaccessibles. Par contre, il serait possible d'atteindre le nœud 4 par 0 en passant par le nœud 6.

Nous avons proposé une solution permettant de répondre à cette contrainte angulaire. L'idée est de transformer l'algorithme A^* pour s'exécuter sur les arcs au lieu de sur les nœuds. Les étiquettes $g(v)$, $f(v)$ et $parent(v)$ seront maintenant défini pour chaque arc a : $g(a)$, $f(a)$ et $parent(a)$. Au départ, on suppose que tous les arcs sortant de la source peuvent

Algorithm 1 Algorithme A* classique

```

1: function PLUSCOURTCHEMINA*( $G = (V, A), s, t$ )
2:   for  $v \in V$  do
3:      $f(v) \leftarrow \infty$ 
4:      $g(v) \leftarrow \infty$ 
5:      $parent(v) \leftarrow nul$ 
6:   end for
7:    $g(s) \leftarrow 0$ 
8:    $f(s) \leftarrow 0$ 
9:    $L \leftarrow \emptyset$ 
10:   $L.Ajouter(s, 0)$ 
11:  while  $L \neq \emptyset$  do
12:     $v \leftarrow L.ObtenirMin()$ 
13:    if  $v = t$  then
14:      return chemin trouvé
15:    end if
16:     $L.Retirer(v)$ 
17:    for  $v \in V^+(w)$  do
18:      if  $g(v) + d_{vw} + h(w, t) < f(w)$  then
19:         $f(w) \leftarrow g(v) + d_{vw} + h(w, t)$ 
20:         $g(w) \leftarrow g(v) + d_{vw}$ 
21:         $parent(w) \leftarrow v$ 
22:        if  $w \in L$  then
23:           $L.MettreAJour(w, f(w))$ 
24:        else
25:           $L.Ajouter(w, f(w))$ 
26:        end if
27:      end if
28:    end for
29:  end while
30:  return Chemin non trouvé
31: end function

```

être empruntés. La distance pour les atteindre est la distance de les traverser. Par la suite, l'algorithme sélectionne l'arc $a = (v, w)$ de priorité minimale. Il tentera alors d'atteindre tous les arcs sortant du nœud w . Chaque arc $b = (w, y)$ peut être atteint si l'angle pour se rendre de v à w à y respecte les contraintes de changement de direction. Si cela est le cas, alors les étiquettes de l'arc b sont mises à jour et l'arc est aussi ajouté à la file de priorité au besoin. Le reste des composantes sont similaires à la version classique de A^* . Un pseudo-code de cet algorithme est donné dans l'Algorithme 2.

Si l'angle respecte la contrainte de changement de direction, d'autres conditions peuvent être testées :

- La limitation angulaire verticale
- Le ceiling
- Les volumes d'exclusion

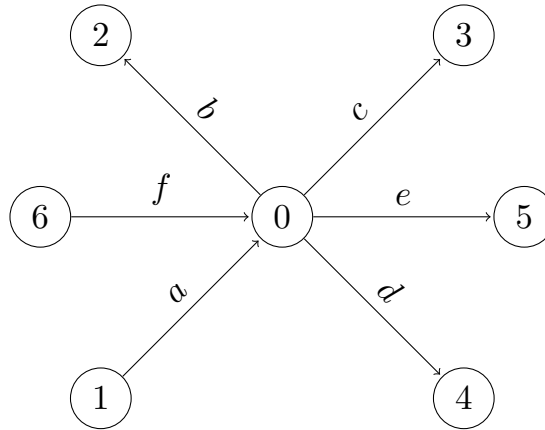


FIGURE 4.9 – Contrainte sur les arcs pour le changement de direction

4.5 Discrétisation 3D : Grille régulière 3D, Octree, Multi-Quadtree

Dans la section précédente, nous avons détaillé une étude comparative des structures de discrétisation 2D dans le but de maîtriser la complexité liée au maillage 2D. Nous avons démontré l'intérêt de l'utilisation du maillage adaptatif avec un graphe de navigation de connexité ajustable. Cette structure permet de réduire la consommation mémoire, le temps de calcul et améliorer la qualité des chemins trouvés. Dans cette section, nous allons étendre cette étude en examinant des structures de discrétisation de l'espace 3D afin d'aboutir à une structure adéquate à la planification de chemins aériens et qui permet la génération de plans de vol paramétrables selon le type de l'appareil et de la mission.

Algorithm 2 Algorithme A* sur les arcs

```

1: function PLUSCOURTCHEMIN( $G = (V, A), s, t$ )
2:   for  $a \in A$  do
3:      $f(a) \leftarrow \infty$ 
4:      $g(a) \leftarrow \infty$ 
5:      $parent(a) \leftarrow nul$ 
6:   end for
7:    $L \leftarrow \emptyset$ 
8:   for  $a = (s, v) \in A^+(s)$  do
9:      $f(a) \leftarrow d_{sv} + h(v, t)$ 
10:     $L.Ajouter(a, f(a))$ 
11:  end for
12:  while  $L \neq \emptyset$  do
13:     $a = (v, w) \leftarrow L.ObtenirMin()$ 
14:    if  $w = t$  then
15:      return chemin trouvé
16:    end if
17:     $L.Retirer(a)$ 
18:    for  $b = (w, y) \in A^+(w)$  do
19:      if  $g(a) + d_{wy} + h(y, t) < f(b)$  then
20:         $f(b) \leftarrow g(a) + d_{wy} + h(y, t)$ 
21:         $g(b) \leftarrow g(a) + d_{wy}$ 
22:         $parent(b) \leftarrow a$ 
23:        if  $b \in L$  then
24:           $L.MettreAJour(b, f(b))$ 
25:        else
26:           $L.Ajouter(b, f(b))$ 
27:        end if
28:      end if
29:    end for
30:  end while
31:  return Chemin non trouvé
32: end function

```

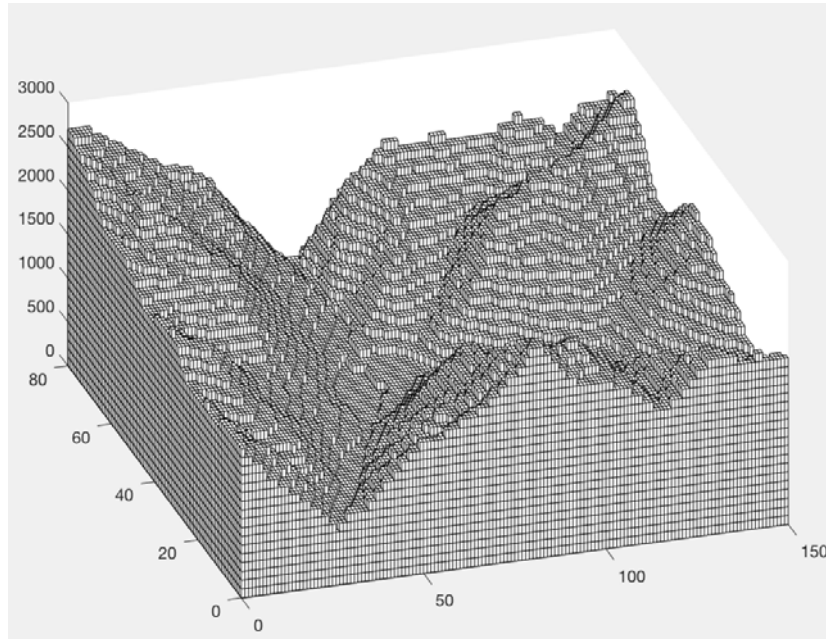


FIGURE 4.10 – Portion du terrain des « Pyrénées » discrétisée en grille régulière 3D

4.5.1 Structures de discrétisation

► **Grille régulière 3D** Nous commençons cette étude par le maillage régulier. Nous donnons l'exemple du terrain des « Pyrénées » de taille $97473m \times 139854m$ avec une altitude s'élevant à $3000m$. Sur la Fig. 4.10, nous montrons une portion de la carte discrétisée en grille régulière. L'application d'un maillage fin de l'ordre de $100m$ engendre une structure avec à peu près 30 millions de nœuds libres. Cette complexité s'avère insolvable en termes de temps de calcul et de consommation mémoire surtout si nous envisageons la génération d'un graphe de navigation en s'appuyant sur cette structure. Pour pallier cette limitation, nous avons examiné un deuxième type de discrétisation : le maillage adaptatif 3D que nous détaillerons dans la section suivante.

► **Octree** La planification de vol sur des terrains de grandes tailles impose une discrétisation irrégulière de l'espace 3D vu la complexité temporelle et spatiale présentée par les grilles régulières. La structure Octree, étant le passage naturel du Quadtree à la 3D, représente un exemple efficace pour le maillage 3D [67] [51]. Cette structure de données hiérarchique est utilisée pour partitionner récursivement l'espace 3D en huit octants à une précision donnée. La taille de l'octant le plus petit définit la résolution (ou la précision) du maillage. Relativement à la grille régulière 3D, l'Octree réduit l'espace mémoire utilisé en raison des grands volumes contigus représentés par un seul nœud feuille. La Fig. 4.11 montre la discrétisation du terrain des « Pyrénées » en Octree. Dans cette figure nous avons gardé que les cuboïdes obstacles

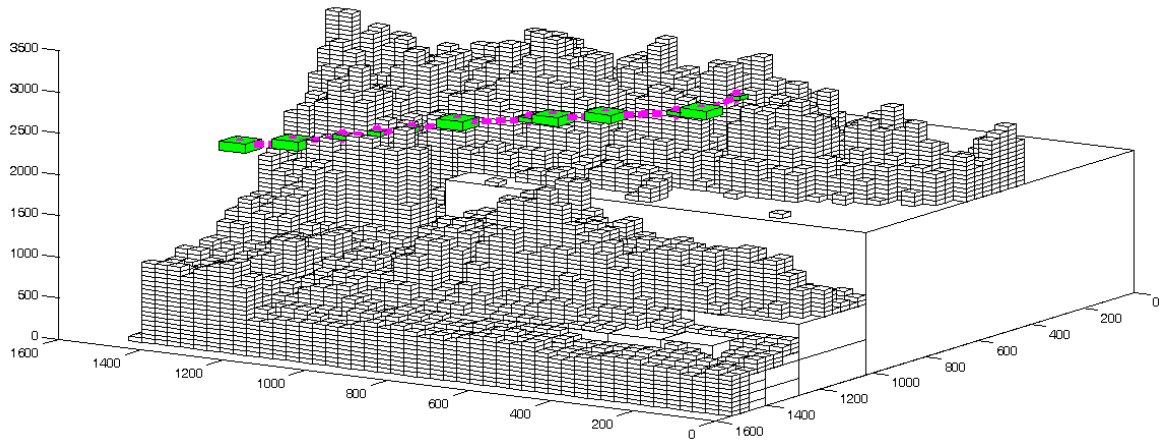


FIGURE 4.11 – Terrain des Pyrénées discrétisé en Octree

où nous remarquons une hétérogénéité en termes de taille et un nombre de voisins qui peut atteindre les dizaines lorsqu'il s'agit d'un cuboïde de grande taille. Pour la navigation 3D, nous utiliserons les cuboïdes libres (les nœuds feuilles de la structure de données hiérarchique) qui seront stockés dans une structure linéaire. Chaque nœud libre contient la charge utile des données : les coordonnées de son emplacement et la liste des voisins. Un exemple de chemin s'appuyant sur les cuboïdes libres est illustré sur la Fig. 4.11. L'application de ce maillage est à l'origine d'une réduction considérable du nombre de nœuds maintenus et de la complexité des calculs lors de la génération du graphe ou du routage. Malgré les avantages liés à ce type de discrétisation, il présente des inconvénients liés à la précision et la qualité des solutions générées. Nous détaillerons ces limitations dans la Section 4.5.3.

► **Multi-Quadtree** Vu les limitations de l'Octree, nous avons introduit une nouvelle méthode de planification de vol 3D profitant des propriétés du maillage en Quadtree et qui s'adapte aisément à la navigation avionique. Nous proposons de découper l'espace 3D sous forme de plusieurs plans 2D discrétisés en Quadtree. Ainsi nous obtenons un maillage en Multi-Quadtree. Comme montré dans la Section 4.3, la représentation spatiale en Quadtree apporte un maillage adaptatif selon la densité des obstacles. La complexité spatiale de cette structure est fonction du nombre de paliers discrétisés (N) soit $N \times O(n \times \log(n))$. Ceci engendre une réduction importante dans la complexité algorithmique liée à ce type de discrétisation. À titre d'exemple, en découplant le terrain des « Pyrénées » en Multi-Quadtree et avec un palier de 150m, nous avons obtenu 23 plans 2D. Sur la Fig. 4.12, nous montrons partiellement le découpage en Multi-Quadtree du terrain des « Pyrénées » (seulement 4 niveaux parmi 23 sont affichés).

De ce fait, nous avons eu l'idée de concevoir une approche multi-altitude afin de permettre la planification de chemins aériens. Cette approche repose sur la construction d'un graphe

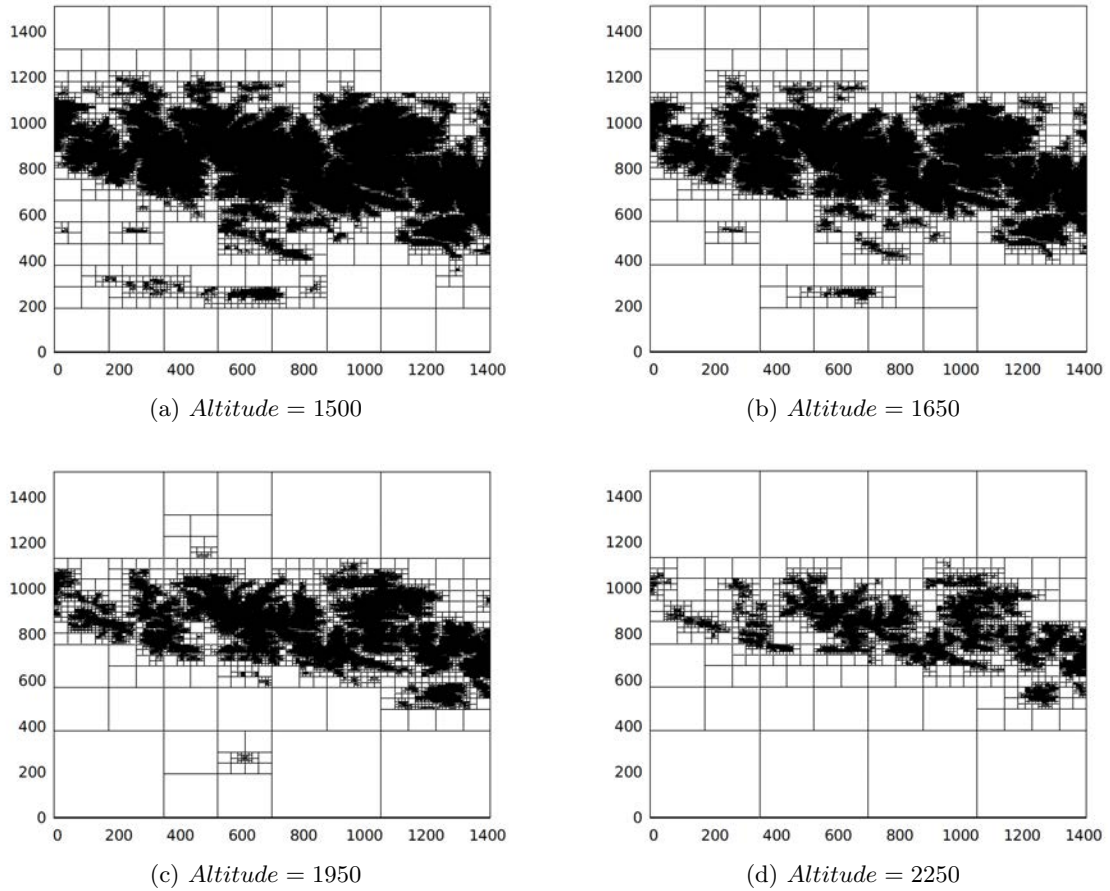


FIGURE 4.12 – Représentation Multi-Quadtree du terrain des « Pyrénées » (seulement 4 niveaux sont représentés)

de navigation basé sur une représentation spatiale en Multi-Quadtree du terrain allant de l'altitude minimale à l'altitude maximale du terrain et séparées d'un palier fixe. La solution de planification associée a été présentée dans le Chapitre 3.

4.5.2 Construction des graphes de navigation

► **Octree** : En se basant sur un maillage en Octree, la construction du graphe de navigation 3D est inspirée de la méthode de génération du graphe 2D. Les nœuds du graphe correspondent aux centres des cuboïdes libres générés par la structure Octree et nous allons utiliser la même notion de *Radius Factor* pour définir le voisinage d'un nœud donné. Pour chaque nœud $i \in [1..N]$, N étant le nombre total des nœuds, nous procédons comme suit :

Nous considérons la sphère ayant comme centre celui du nœud et le rayon r égal à la diagonale de la case de l'Octree multipliée par le *Radius Factor*. Pour chaque nœud j appartenant à l'ensemble des nœuds de l'Octree, nous vérifions s'il appartient au voisinage du

nœud i . Un voisin doit satisfaire à deux conditions :

1. C_j (centre du cuboïde j) appartient à la sphère, ceci est vérifié par une comparaison entre la distance d_{C_i, C_j} et le rayon r de la sphère,
2. il n'y a pas d'obstacles qui coupent le segment entre C_i et C_j . Pour ce faire, nous appliquons l'algorithme de tracé de segment en 3D afin de détecter les éventuels obstacles présents sur le segment reliant les centres des deux cuboïdes.

Si les deux conditions sont vérifiées, l'arc $\widehat{C_i C_j}$ est alors ajouté au graphe. Ainsi, le graphe de navigation est construit selon un voisinage variable qui dépend du *Radius Factor*.

Un graphe est défini par un tableau de L éléments chaque élément représente un arc. Ce dernier est une structure définie par : l'identifiant du nœud source, l'identifiant du nœud destination et la distance de l'arc. Ce tableau est accompagné par un autre tableau de N éléments représentant les nœuds du graphe.

► **Multi-Quadtree** : Le graphe de navigation Multi-Quadtree est issu d'une discrétisation adaptative en plusieurs paliers 2D allant de l'altitude minimale à l'altitude maximale du terrain. Les paliers sont séparés par une hauteur fixe. Suite au maillage, nous procédons à la construction du graphe 3D :

- nous commençons par la génération des graphes 2D sur les différents paliers. Ces graphes permettent la navigation entre les centres des nœuds d'un Quadtree donné,
- nous connectons les différents paliers à travers les arcs de raccordement construits tout en respectant les limitations angulaires de l'appareil,
- nous procédons à la fusion de la composante horizontale et la composante verticale afin d'obtenir le graphe 3D,

L'utilisation de ce type de graphe est recommandée dans le contexte avionique pour plusieurs raisons :

- le maillage en Multi-Quadtree permet la navigation horizontale ce qui s'adapte au vol en palier. Ce type de plan de vol est appliqué dans les missions qui favorisent le confort des passagers,
- le découplage entre la composante horizontale et la composante verticale du graphe permet de décorréler la complexité de calcul.
- la construction des arcs de raccordement en faisant appel à des méthodes géométriques permet de répondre aux limitations angulaires de l'appareil. En effet, cette méthode permet l'élaboration d'un plan de vol avec des durées de calcul très courtes. Ces routes peuvent avoir différents profils selon le type de la mission.
- Le degré de connexité variable des graphes 2D selon le *Radius Factor* permet d'ajuster le temps de prétraitement selon la qualité de solution souhaitée ainsi que le temps disponible pour le prétraitement.

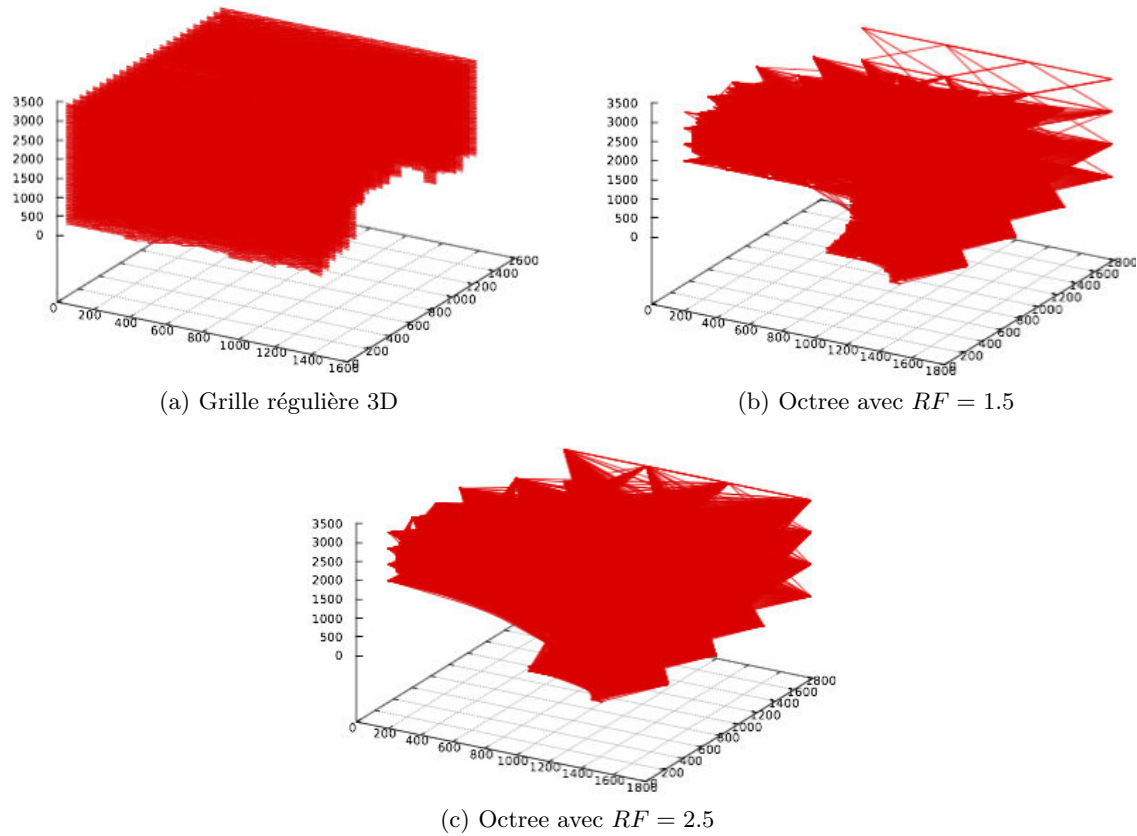


FIGURE 4.13 – Différents graphes 3D générés à partir du terrain des « Pyrénées »

4.5.3 Résultats

4.5.3.1 La phase de prétraitement

Dans cette section, nous projetons les trois types de prétraitement 3D sur la carte des « Pyrénées ». Nous appliquons l'algorithme de routage sur les différentes structures de données afin d'évaluer les performances des phases de prétraitement et de routage. La Fig. 4.13 décrit différents graphes de navigation 3D. Le premier type correspond à la grille régulière. Cette structure doit générer 79024704 nœuds. Vu le débordement mémoire impliqué par les calculs, nous avons écarté son utilisation. En effet, la complexité spatiale et temporelle introduite par cette structure restreint son application à la planification de vol 3D pour des terrains étendus. Un autre inconvénient lié aux graphes réguliers 3D est la longueur constante des arcs. Elle est proportionnelle à la taille d'une case de la grille (de l'ordre de 100m) et varie selon la précision du maillage. Pour les routes de longues distances (de l'ordre de centaines de kilomètres), le chemin calculé doit contenir des centaines de points de passage pour un maillage fin. Ce qui est très difficile à exploiter dans le contexte avionique. Nous précisons que nous avons diminué la résolution de la carte afin d'obtenir le graphe affiché sur la Fig. 4.13 (a).

Le deuxième graphe de navigation affiché sur la Fig. 4.13 (b) résulte d'un maillage en

TABLE 4.5 – Performances de la phase de prétraitement pour différentes structures de données (terrain des « Pyrénées »)

	Grille régulière 3D	<i>Octree</i> avec $RF = 1.5$	<i>Octree</i> avec $RF = 2.5$	Multi-Quadtree avec $RF = 1.5$	Multi-Quadtree avec $RF = 2.5$
Durée de pré-traitement (ms)	Débordement mémoire	8164	21180	164900	187201
Nombre de nœuds	79024704	8701	8701	261807	261807
Précision plus grande case (m)	$68.79 \times 92.62 \times 100$	$70440.96 \times 94842.88 \times 1700$	$70440.96 \times 94842.88 \times 1700$	$24351.66 \times 34917.74 \times 150$	$24351.66 \times 34917.74 \times 150$
Précision plus petite case (m)	$68.79 \times 92.62 \times 150$	$2201 \times 2964 \times 53$	$2201 \times 2964 \times 53$	$68.79 \times 92.62 \times 150$	$68.79 \times 92.62 \times 150$
Nombre d'arcs		129351	325172	10766500	12768353
Temps de pré-traitement (s)	Non déterminé	8.16	21.18	164.9	187.2

Octree avec un *Radius Factor* égal à 1.5. D'après les résultats relevés du Tableau 4.5, ce graphe se caractérise par :

1. un nombre de nœuds très réduit par rapport à la grille régulière : nous obtenons 8701 nœuds au lieu de 79024704. Cette caractéristique est liée au maillage adaptatif permettant de réduire la complexité spatiale.
2. La richesse en arcs de navigation (129351 arcs pour cet exemple) : en effet, le graphe construit avec un *Radius Factor* = 1.5 couvre l'ensemble de l'espace 3D navigable. Nous précisons que les arcs de ce graphe évitent les obstacles du terrain ce qui va accélérer les calcul pendant la phase de routage.

Le troisième graphe est obtenu avec un *Radius Factor* de 2.5 appliqué sur le même *Octree* dans une durée de 21.18s. Ceci a un impact direct sur la connexité. En effet, le voisinage des cases a été élargi ce qui a introduit un graphe plus riche en arcs (325172). Les zones connectées sont plus étendues par rapport au graphe avec un *Radius Factor* de 1.5.

Même avec une connexité élevée, ce type de graphe présente des limitations pour la navigation aérienne de longue distance. En effet, sa structure géométrique ne respecte pas les limitations angulaires imposées par l'appareil. Par exemple, il ne permet pas le vol en palier vu le changement constant entre les d'altitudes. Ceci est dû principalement au maillage hétérogène de l'*Octree* (cuboïdes de tailles différentes).

En s'appuyant sur l'étude de complexité des méthodes de discrétisation 2D, nous avons proposé la génération d'un graphe résultant d'un maillage irrégulier (Multi-Quadtree) per-

mettant de respecter les contraintes de l'appareil et de la mission à accomplir. En utilisant cette nouvelle structure, nous avons généré un graphe reliant 23 niveaux d'altitude avec un palier de 150m, un *Radius Factor* de 2.5 et des angles de montée/descente entre $[4^\circ, 15^\circ]$. Le graphe résultant contient 261807 nœuds et 12768353 arcs (voir Tableau 4.5). Le *Radius Factor* permet d'ajuster la densité horizontale du graphe de navigation alors que le nombre des arcs de raccordement dépend de l'intervalle des angles de montée/descente. Vu la découplage entre les deux composantes horizontale et verticale, la complexité et la connexité associées à chaque composante sont entièrement décorréélées. Ceci permettra l'adaptation des plans de vol calculés au domaine avionique. Concernant les temps de prétraitement, les résultats obtenus sont très satisfaisants : de l'ordre de quelques minutes (2 à 3min) pour des graphes générés avec un $RF = 1.5$ et $RF = 2.5$. Ces durées sont convenables pour des missions qui nécessitent la génération de plan de vol en urgence. Dans d'autres cas, l'utilisateur pourra se permettre de générer des graphes plus denses en termes d'arcs afin d'obtenir des chemins de meilleure qualité dans la phase de routage. Pour ce faire, il faut améliorer la résolution de la structure Multi-Quadtree générée ou bien augmenter la valeur du *Radius Factor*.

4.5.3.2 La phase de routage

Dans cette section, nous allons comparer les performances de la phase de routage ainsi que la qualité des chemins générés à partir des différentes structures 3D étudiées. Les expérimentations seront réalisées sur le terrain des « Pyrénées » en considérant les points situés au « Nord-Ouest » et le « Sud-Est » de la carte. Les altitudes correspondantes sont extraites à partir de la matrice d'élévation. Dans cette étude, nous avons écarté la grille régulière 3D puisque sa complexité spatiale et temporelle ne répond pas à notre domaine d'application. Pour explorer les graphes de navigation en tenant compte de différentes contraintes, nous utilisons notre algorithme *Any-angle adaptive A** décrit dans la section précédente.

Nous avons commencé par l'évaluation des performances de la phase de routage en considérant un graphe généré à partir de la structure Octree avec un *radius Factor* de 1.5 et 2.5. Cette structure présente un nombre de nœuds et d'arcs très réduit par rapport à la grille régulière et la Multi-Quadtree (voir Tableau 4.5). Pour cette raison, les temps de routage obtenus sont de l'ordre de quelques millisecondes (4ms et 7ms respectivement pour $RF = 1.5$ et $RF = 2.5$). La structure fournit aussi un nombre de points de passage réduit (31 et 15 respectivement pour $RF = 1.5$ et $RF = 2.5$) par rapport aux distances des chemins générés (273358 et 224102 respectivement pour $RF = 1.5$ et $RF = 2.5$). Les Fig. 4.14 (a) et (c) illustrent les profils des chemins générés. Notre algorithme de routage a convergé rapidement vers la destination en exploitant le graphe qui évite les obstacles du terrain. Nous remarquons un changement d'altitude à chaque point de passage avec différents angles pour les deux profils, ceci est dû principalement à l'hétérogénéité de la structure. L'augmentation du *Radius Factor* a amélioré la qualité du chemin généré mais ce résultat reste loin d'être exploitable dans notre

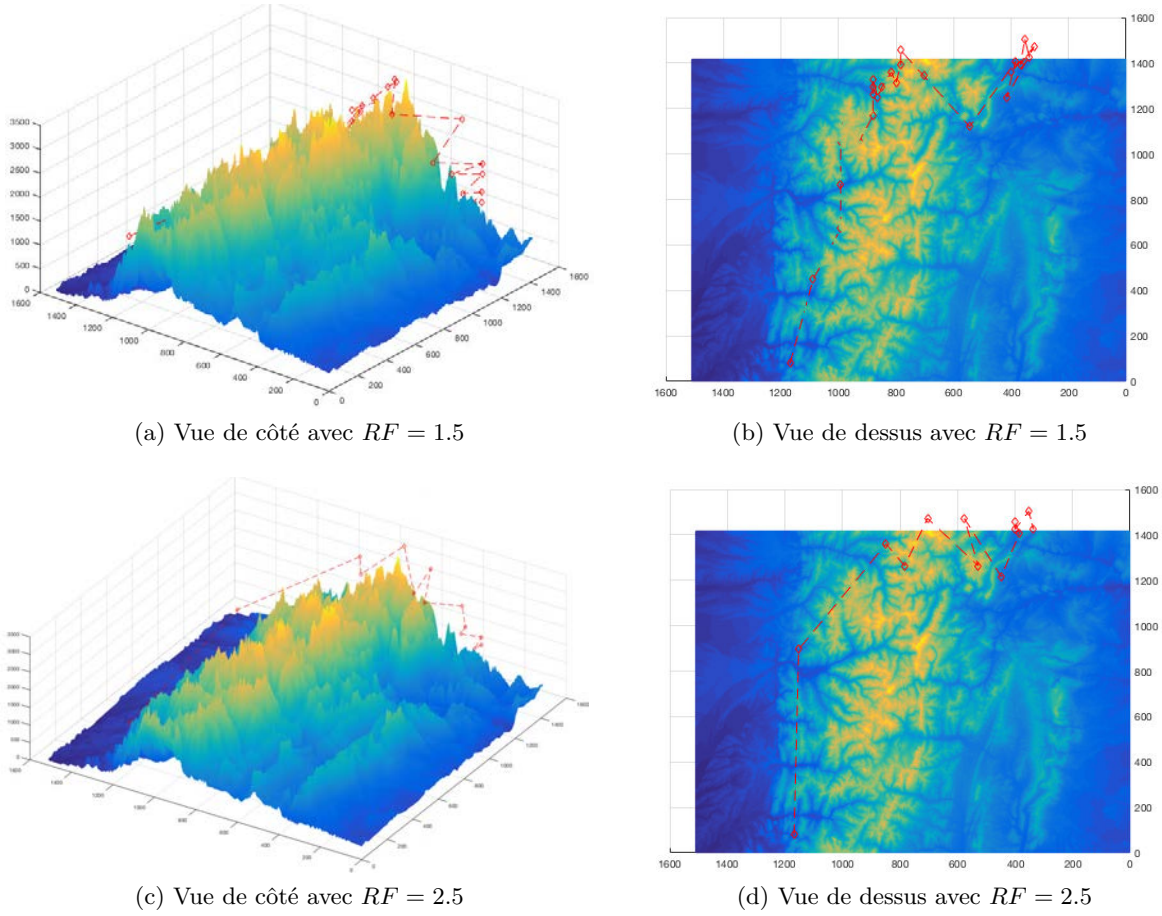


FIGURE 4.14 – Routage sur un graphe généré à partir d’un Octree avec $RF = 1.5$ et $RF = 2.5$ (terrain des « Pyrénées »)

contexte (voir les vues de dessus dans les Fig. 4.14 (b) et (d)).

Par la suite, nous avons étudié les performance de la phase de routage en explorant un graphe généré à partir de la structure Multi-Quadtree de notre terrain. Plusieurs paramètres sont considérés à savoir : le *Radius Factor*, les limitations angulaires et le *ceiling*. La variation du *Radius Factor* nous permettra d’obtenir des graphes avec différents degrés de connexité. Dans notre analyse dynamique, nous faisons varier un seul paramètre à chaque routage soit les angles de montée/descente soit le *ceiling*. Le Tableau 4.6 résume les résultats obtenus en termes de distance, nombre de points de passage et temps de routage.

Pour un graphe généré avec un $RF = 1.5$, nous avons réalisé les scénarios suivants (Fig. 4.15) :

- **Scénario 1** : Les angles de montée/descente sont entre $[4^\circ, 15^\circ]$ et nous n’imposons pas de *ceiling*. Le plan de vol trouvé est de $152.755km$ et contient 87 points de passage. Le routage a été calculé en 4 secondes. Sans contrainte angulaire ou de *ceiling*, le plan de vol suit une ligne droite vers la destination en surmontant les obstacles (voir la vue de dessus

TABLE 4.6 – Routage avec différents paramètres sur un graphe généré à partir d'un Multi-Quadtree (terrain des « Pyrénées »)

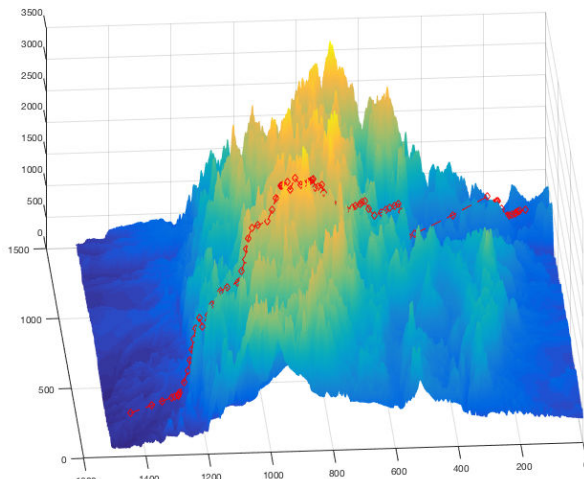
<i>Radius Factor</i>	1.5	1.5	1.5	2.5	2.5	2.5
Angles de montée/descente	$[4^\circ, 15^\circ]$	$[10^\circ, 15^\circ]$	$[4^\circ, 15^\circ]$	$[4^\circ, 15^\circ]$	$[10^\circ, 15^\circ]$	$[4^\circ, 15^\circ]$
<i>Ceiling</i> (m)	0	0	2000	0	0	2000
Distance (m)	152755	153262	156597	151820	152158	155298
Nombre de points de passage	87	108	84	70	80	62
Durée de calcul (s)	4	1	3	4	2	5

dans la Fig. 4.15 (b). Avec un *Radius Factor* égal à 2.5, nous trouvons que le chemin obtenu est légèrement plus court 151820m au lieu de 152755m et Il contient seulement 70 points de passage au lieu de 87. En effet, un graphe généré avec un $RF = 1.5$ est plus riche ce qui se traduit par des connexions plus directes.

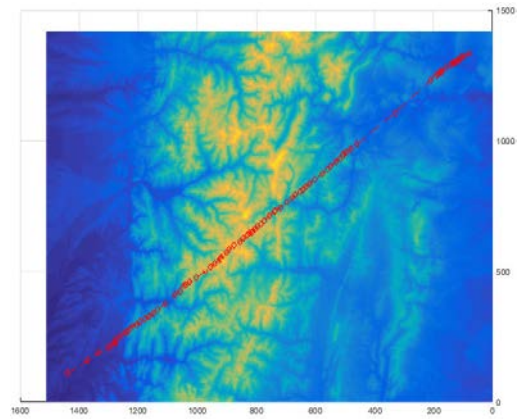
- **Scénario 2 :** Les angles de montée/descente sont compris entre $[10^\circ, 15^\circ]$ et nous n'imposons toujours pas de *ceiling*. Ici, la qualité de la solution s'est légèrement détériorée. Le nombre de points de passage a passé à 108 et la distance à 153262m. Par contre, sur les profils verticaux il est clair qu'en réduisant l'intervalle des angles de montée/descente, le chemin élaboré correspond mieux à un vol en palier (il représente moins de changements d'altitude) comme illustré sur Fig. 4.15 (c). Ceci est souhaitable pour certaines missions comme les missions « VIP » où le confort des passagers est un critère primordial. Pour ce type de missions, nous accordons plus d'importance au confort qu'à la distance et le temps de vol. De ce fait, le paramètre « limitations angulaires » permet d'ajuster le profil vertical de vol. De plus, la réduction de l'intervalle des angles de montée/descente implique la réduction du nombre d'arcs verticaux à explorer ce qui fait que l'algorithme converge rapidement vers la destination. Cette réduction a un impact sur le temps de routage. Pour cet exemple, la durée de routage a passé de 4 à 1s.
- **Scénario 3 :** Les angles de montée/descente sont compris entre $[10^\circ, 15^\circ]$ et nous avons imposé un *ceiling* égal à 2000m en interdisant de voler en dessus de cette altitude. Le but est d'éviter certaines conditions météo. Par exemple, pour certaines zones géographiques dépasser une altitude donnée présente des risques de givrage. Nous avons constaté que le plan de vol calculé est un peu plus long (156597m). En effet, l'algorithme a été amené à contourner les obstacles au lieu de les surmonter comme dans les deux scénarios précédents (voir la vue de dessus dans la Fig. 4.15 (f)). Par contre, le nombre de points de passage a été réduit par rapport au scénario 1 avec une tendance de vol en palier.

4.6 Conclusion

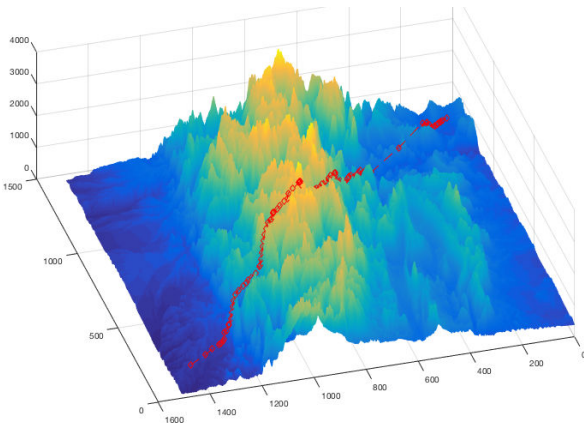
Dans ce chapitre, nous avons abordé une comparaison entre plusieurs structures de données permettant la discrétisation de l'espace en 2D et 3D dans le but d'aboutir à un maillage qui



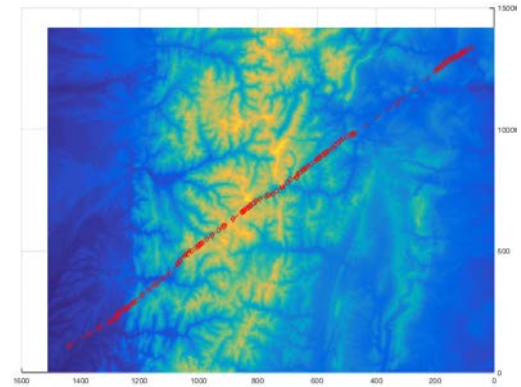
(a) Vue de côté avec une limitation angulaire entre 4° et 15°



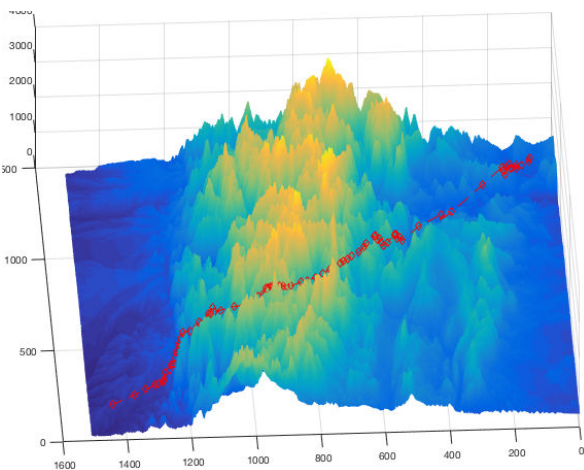
(b) Vue de dessus avec une limitation angulaire entre 4° et 15°



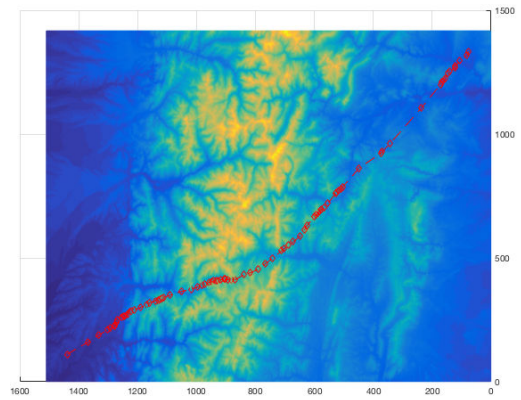
(c) Vue de côté avec avec une limitation angulaire entre 10° et 15°



(d) Vue de dessus avec avec une limitation angulaire entre 10° et 15°



(e) Vue de côté avec un *ceiling*=2000m



(f) Vue de dessus avec un *ceiling*=2000m

FIGURE 4.15 – Routage sur un graphe généré à partir d'un Multi-Quadtree avec un $RF = 1.5$ (terrain des « Pyrénées »)

servira de base pour la planification de vol 3D lors de la préparation d'une mission ou la replanification dynamique en cas d'urgence. Nous avons exploré des structures de données classiques basées sur des grilles régulières 2D et 3D, le Quadtree, l'Octree ainsi que notre propre solution le Multi-Quadtree. La comparaison a été assurée selon plusieurs critères tels que le temps de prétraitement, la consommation mémoire ainsi que la réponse de l'algorithme de routage appliqué sur les différents types de maillage. Le plan de vol calculé doit répondre aux exigences d'un scénario réel tout en respectant les contraintes avioniques.

Nous avons constaté l'impact du maillage adaptatif 2D de type Quadtree. En effet, cette structure de données tend à réduire la complexité algorithmique qui caractérise le maillage régulier et améliorer la qualité des chemins trouvés lors de l'application des algorithmes de routage. Dans la partie comparaison des structures 3D, nous avons évalué les grilles régulières, les Octrees ainsi que notre approche de discrétisation Multi-Quadtree. Malgré le gain en temps de calcul et de consommation mémoire offert par le maillage en Octree par rapport au maillage régulier, la qualité des chemins obtenus ne satisfait pas les contraintes aéronautiques. Notre approche de planification de chemins aériens basée sur une discrétisation en Multi-Quadtree contribue à :

- la réduction de la complexité en temps/mémoire,
- l'élaboration de solutions de qualité ajustable selon les paramètres de prétraitement,
- l'obtention de chemins avec différents profils de vol qui s'adaptent selon le type de la mission et de l'appareil.

Cette étude nous a permis de prouver les performances de notre approche de planification de vol basée sur un maillage Multi-Quadtree par rapport aux méthodes de planification classiques. Dans le chapitre suivant, nous présentons l'environnement de génération de plan de vol qui intègre notre système de planification de chemins aériens. Cet environnement fera partie des systèmes de préparation de missions au sol ou des systèmes d'assistance aux pilotes en cours de vol.

Chapitre 5

DTANAV : environnement de planification et d'exploration de chemins aériens

5.1 Introduction

Jusqu'à présent, les systèmes permettant la planification de missions sont limités en termes de temps réponse et de flexibilité concernant la génération des profils de vol bien adaptés selon le type de la mission. La complexité de cette tâche est fonction du nombre et de la nature des contraintes et du temps disponible pour trouver une solution acceptable. Pour pallier ces limitations, nous allons concevoir un environnement de génération automatique et d'exploration de plans de vol pour des aéronefs. Cet environnement est basé sur le système de planification de chemins aériens que nous avons présenté dans le Chapitre 3. La planification repose sur deux phases : le prétraitement et le routage. La première phase étant statique (appliquée une seule fois lors de la préparation de la mission) assure le maillage multi-altitude de l'espace 3D et la construction du graphe de navigation permettant le raccordement entre les nœuds de la structure de discrétisation. Cette structure de modélisation de l'espace a la particularité de :

- maîtriser la complexité spatiale et temporelle,
- offrir des résultats précis avec des temps de réponse réduits,
- respecter les contraintes qui correspondent à l'appareil et à la mission,
- obtenir des plans de vol avec différents profils selon les paramètres d'entrée et le type de la mission à assurer.

La phase de routage peut être appelée avant le vol (pour préparer une mission) et en cours de vol (en cas de replanification de la mission). Elle permet de calculer un plan de vol

en explorant le graphe de navigation dans un temps réduit. Afin de justifier les choix effectués pour concevoir notre système, nous avons procédé à une étude dans le Chapitre 4 qui compare notre approche à d'autres méthodes qui s'appuient sur différentes structures de discrétisation de l'espace. Dans ce chapitre, nous abordons la mise en œuvre de l'environnement de planification de vol qui intègre le procédé de planification de chemins aériens. L'intégration d'une fonction de performances dans notre environnement ainsi que l'étude de scénarios de replanification dynamique en cas d'urgence font aussi partie de ce chapitre.

Ce chapitre sera organisé comme suit : Dans la Section 5.2, nous décrirons les fonctions avioniques nécessaires lors de la phase de préparation de mission ou en cas de replanification dynamique ainsi que le diagramme d'utilisation de notre environnement en interaction avec ces fonctions. La Section 5.3 détaillera l'architecture logicielle de l'outil DTANAV que nous avons développé ainsi que l'IHM correspondante. Ensuite, dans la Section 5.4, nous présenterons un exemple de fonction de performances qui pourra être utilisée pour évaluer différents critères lors de la phase d'exploration du graphe. La Section 5.5 illustre quelques scénarios de replanification dynamique et démontre l'efficacité de DTANAV pour répondre aux exigences temporelles en cas d'urgence.

5.2 Description des fonctionnalités de l'outillage

La phase de planification de missions a toujours besoin de plusieurs paramètres et informations liés à l'appareil, le type de la mission, le terrain et l'environnement. Certaines informations sont recueillies à partir de différentes bases de données, d'autres proviennent des systèmes de détection d'information ou de communication. En plus, certains paramètres sont saisis par le pilote ou le personnel de la navigation. Dans cette section, nous allons décrire tout d'abord les fonctions avioniques qui doivent interagir avec notre environnement de planification afin d'obtenir les informations nécessaires pour préparer une mission en toute sécurité. Le système avionique ainsi que les différentes étapes nécessaires pour la préparation d'une mission sont présentés dans la Fig. 5.1).

Les principales fonctions avioniques qui interviennent dans notre procédé de planification se composent de :

- Base de données mission (*Mission Database*) : elle comprend les données suivantes :
 - Les données terrain : altitude du terrain et marques de jour utiles
 - Les obstacles : tours, antennes, centrale éolienne, grues ou lignes électriques indiqués sur la carte ou une base de données spécifique
- Systèmes de détection et de communication (*Sensors and Connectivity*) : permettent de fournir :
 - Le plafond de navigation :

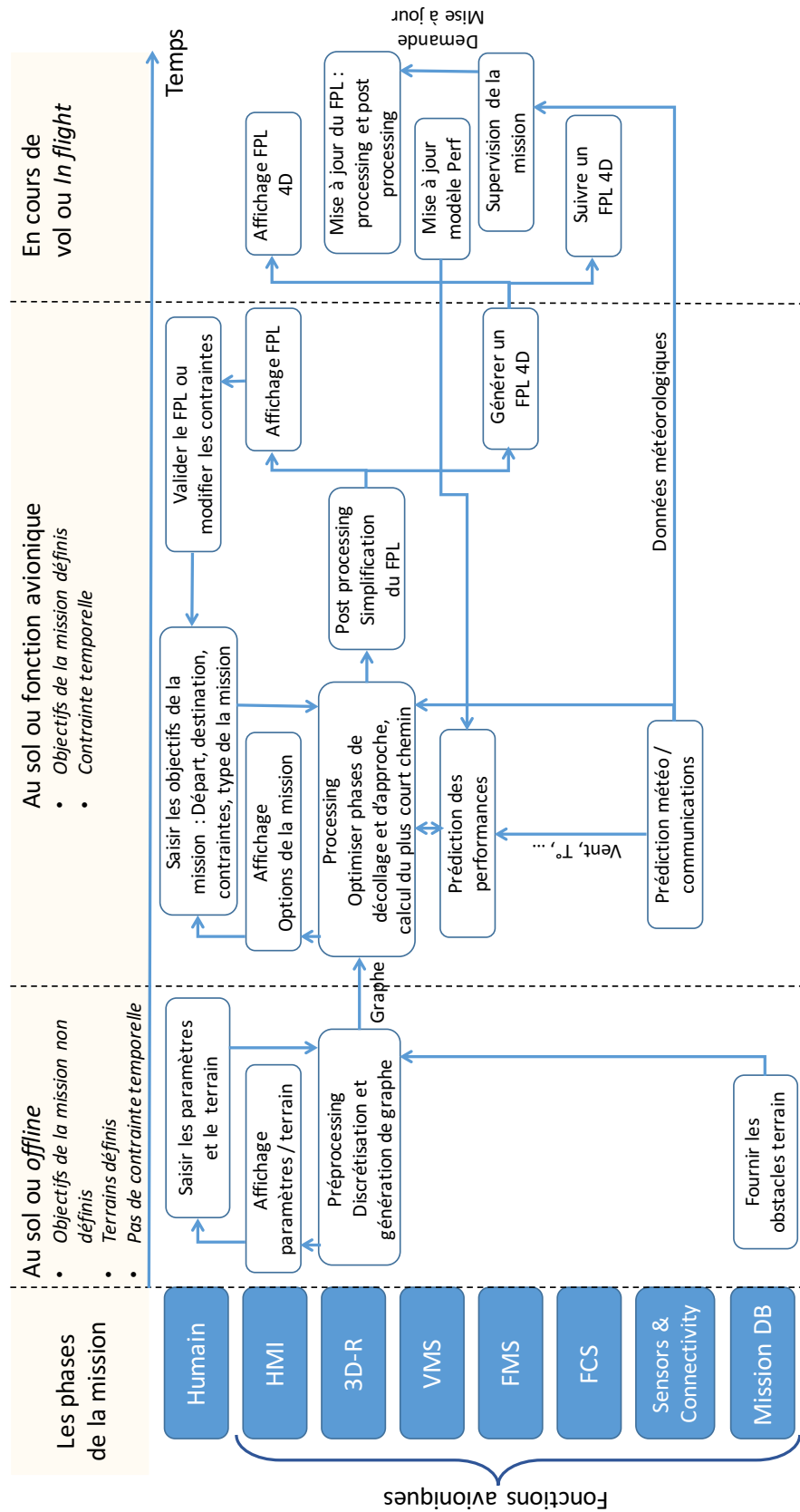


FIGURE 5.1 – Fonctions avioniques et diagramme d'utilisation de l'outil de planification de missions

- ◇ Plafond météorologique (nuage ou plafond de givrage)
 - ◇ Plafond de performance (*In Ground Effect* , *Out of Ground Effect*, plafond de service *All Engine Operative* / *One Engine Inoperative*)
 - Les volumes d'exclusion :
 - ◇ NOTAM (*NOTice To AirMen*) : Espaces aériens interdits/restreints (selon le type de mission) et les limitations ou les dangers liés aux aéroports
 - ◇ Zones sensibles au bruit
 - ◇ Cellules orageuses (détectées par le radar météorologique)
 - Le Plancher de navigation (*navigation floor*) :
 - ◇ Altitude du terrain plus la hauteur de sécurité
 - ◇ Obstacles et hauteur de sécurité
 - La météo : situation actuelle et prévisions pour les nuages, le vent, la température
- FCS (*Flight Control System*) : le système de contrôle de vol
 - FMS (*Flight Management System*) : le système de gestion de vol
 - VMS (*Vehicle Management System*) : le système de gestion du véhicule, il gère les contraintes de performances de l'appareil et consiste à vérifier la faisabilité du plan de vol en ce qui concerne la consommation du carburant. Il fournit :
 - L'altitude de performance de vol
 - Performance de montée/descente
 - Performance de vol stationnaire
 - Charge utile : scénario de changement de masse pendant la mission pour établir la performance de l'appareil. Une étude d'un modèle de performances sera décrite dans la Section 5.4.
 - Le routeur 3D : Il regroupe la phase de prétraitement de l'espace 3D et le routage. Il fait l'objet de notre travail de recherche décrit dans le Chapitre 3. Dans la Section 5.3, nous montrerons la communication homme/machine via l'interface utilisateur qui permettra de lancer/communiquer avec le routeur 3D. La Section 5.4 envisagera la possibilité de couplage du routeur 3D avec le modèle de performances afin d'assurer l'optimisation du plan de vol selon différents critères.
 - HMI : L'Interface Homme Machine (IHM) permet la communication entre le système et l'utilisateur. Cet élément sera détaillé dans la Section 5.3.

5.2.1 Diagramme d'utilisation de l'outillage

Dans cette section, nous décrivons la vision de notre partenaire industriel Airbus Helicopters concernant l'utilisation de l'outil de planification de missions depuis la phase de préparation jusqu'à la phase de supervision du déroulement de la mission en cours de vol. Les différentes fonctions du système avionique doivent interagir entre elles et avec l'opérateur afin de procéder à la planification de mission avant le vol ou bien à la replanification dynamique en cours de vol. Nous distinguons principalement 3 phases en suivant l'axe du temps (voir la Fig. 5.1) :

1. **La phase préparation au sol ou *offline*** : Généralement, dans cette phase les objectifs de la mission ne sont pas encore définis. Néanmoins, le terrain est connu ce qui nous permettra d'entamer le prétraitement requis pour la discrétisation du terrain et la construction du graphe de navigation 3D. Cette phase pourra être exécutée sans contrainte temporelle sauf en cas d'urgence. Elle se décline principalement en deux étapes :

► **Collecte des données terrain** : consiste à collecter des informations à partir de la Base de données mission. En plus de la carte numérique du terrain, nous avons besoin de considérer les espaces aériens interdits et les zones de danger comme les aéroports et tenir compte des obstacles statiques de type tours, antennes, centrale éolienne, grues ou lignes électriques. Ensuite, un traitement qui consiste à augmenter la taille des obstacles est effectué de manière à garantir une marge de sécurité horizontale et verticale réglementaire.

► **Le prétraitement** : il est appliqué uniquement lors de la préparation de la mission au sol. Il est lancé par l'utilisateur via l'IHM. En fonction de l'ensemble des paramètres algorithmiques (palier, *Radius Factor*, etc.) et les informations de la Base de données mission, le graphe de navigation 3D est alors construit afin qu'il soit communiqué à la phase suivante.

2. **La phase de planification sur une station au sol ou par une fonction avionique** : Dans cette phase les objectifs de la mission sont connus et la contrainte de temps pourra être définie en fonction de la criticité de la mission. Notre solution algorithmique est proposée en adéquation avec la capacité de traitement des calculateurs embarqués au sein d'un hélicoptère. Pour cette raison, cette phase pourra être exécutée sur une station au sol ou bien par une fonction avionique. Les étapes nécessaires sont :

► **Collecte des informations de la mission** : À travers l'IHM, le pilote saisit les paramètres liés à la mission (départ, destination, contraintes, type de la mission, etc.)

en plus des informations collectées à partir des systèmes de détection et de communication (exemple prédiction météo). Ces informations sont analysées et combinées avec les connaissances du pilote et le manuel de vol pour définir les contraintes géométriques. Toutes ces contraintes ne sont pas forcément considérées comme strictes, mais il est nécessaire d'en tenir compte pour garantir une mission en toute sécurité.

► **Le routage :** cette étape est dynamique. Elle peut être appelée une ou plusieurs fois avant ou en cours de vol. En fonction des paramètres de la mission et de l'appareil saisis par l'utilisateur via l'IHM, le routeur 3D est couplé avec le VMS afin de trouver un plan de vol qui répond aux différentes contraintes. Il explore le graphe de navigation généré lors de la première phase. Le plan de vol établi permet d'optimiser un critère donné. En préalable à la phase de routage, une fois l'origine et la destination connues, une optimisation des axes de décollage et d'approche est effectuée pour relier un point géographique au sol jusqu'à une hauteur de sécurité correspondant à un palier du graphe généré lors du prétraitement car le routage n'autorise pas la création de segments proches des obstacles ou du terrain.

► **Calcul de performances et faisabilité du plan généré :** consiste à vérifier la faisabilité du plan de vol en ce qui concerne la consommation du carburant via le VMS qui intègre une fonction de calcul des performances (temps, carburant, etc.). Cette activité est essentielle pour la sécurité, mais ne peut pas être effectuée manuellement avec une précision élevée et un temps réduit. Prendre en compte le vent pour calculer la vitesse au sol sur les segments puis calculer le temps et la consommation de carburant par segment donne une estimation fiable du besoin total en carburant. L'opérateur évalue la qualité du résultat qui regroupe les différentes sorties de l'algorithme : plan de vol, distance parcourue, temps de vol et nombre de segments. Si la sortie est valide, le plan de vol sera transmis à la prochaine fonction. Sinon, l'opérateur modifie les paramètres de la mission et relance le routage.

► **La simplification du plan de vol :** en fonction du plan de vol simplifié grâce au *postprocessing*, le FMS permet de générer le plan de vol 4D (la trajectoire). Les plans de vol générés par le procédé de routage décrit précédemment comportent de nombreux segments souvent quasi alignés dans un plan horizontal et de nombreuses variations de pente dans le plan vertical. Il est nécessaire de réduire le nombre de segments/points jusqu'à obtenir une complexité comparable à celle d'un plan de vol créé manuellement. Cette simplification facilitera le suivi du vol par le pilote, améliorera le confort en réduisant le nombre de transition et permettra le cas échéant au pilote de manipuler un plan de vol moins complexe s'il souhaite le modifier. Après la validation du plan de vol par l'opérateur (pilote), le FMS se charge de la génération de la trajectoire.

3. **La phase en cours de vol :** Pendant le vol, il faut assurer une supervision permanente du déroulement de la mission suivant le plan généré. Le système de contrôle de vol (FCS) intervient pour contrôler la puissance motrice et les différents actionneurs nécessaires pour exécuter des lois de guidage sur la trajectoire tout en contrôlant la stabilité de l'appareil.

► **Replanification dynamique :** En cours de vol, même après l'élaboration d'un chemin aérien précis au sol, si la situation météorologique change, un moteur tombe en panne ou bien de nouvelles instructions ATM (*Air Traffic Management*) sont reçues, il est nécessaire de replanifier la mission selon une stratégie définie au préalable et qui est en adéquation par rapport à la situation d'urgence. Dans ce cas, la mise à jour du plan de vol avec les nouveaux paramètres devient indispensable. Nous avons besoin de faire appel à notre fonction avionique qui va proposer au pilote une solution de repli pour éviter un éventuel accident.

5.3 DTANAV : Demonstration Tool for Aircraft NAVigation

5.3.1 Description générale de l'outil DTANAV

Au niveau du système avionique décrit précédemment (Fig. 5.1), nos travaux de recherche s'inscrivent dans la fonction *Autoroutier 3D*. Ils assurent la modélisation de l'espace 3D et la génération automatique d'un plan de vol qui respecte les contraintes de l'appareil et de la mission. Pour concrétiser notre approche, nous avons conçu un environnement logiciel qui assure la communication entre l'utilisateur et la fonction de routage 3D ainsi que la génération automatique d'un plan de vol 3D. Ceci contribue à la réduction de la charge de travail « navigation » et renforce la sécurité en vol et la performance de la mission. Nous avons baptisé notre environnement DTANAV « Demonstration Tool for Aircraft NAVigation ».

L'architecture de notre environnement de démonstration DTANAV est présentée dans la Fig. 5.2. Elle comprend :

- Une IHM de démonstration utilisée pour définir les entrées, la configuration et l'affichage des paramètres, des contraintes et du résultat (les flèches indiquent le flux de données).
- Une fonction de génération FPL (*Flight PPlan*) appelée par la fonction MMI (*Man Machine Interface*) et s'appuyant sur un gestionnaire de requêtes de contraintes.
- Un gestionnaire de requêtes de contraintes qui fournit un service de niveau d'abstraction aux fonctions MMI et Génération FPL.
- Plusieurs modèles de contraintes issus de différentes bases de données (atmosphère, hélicoptères, terrains et espaces aériens).

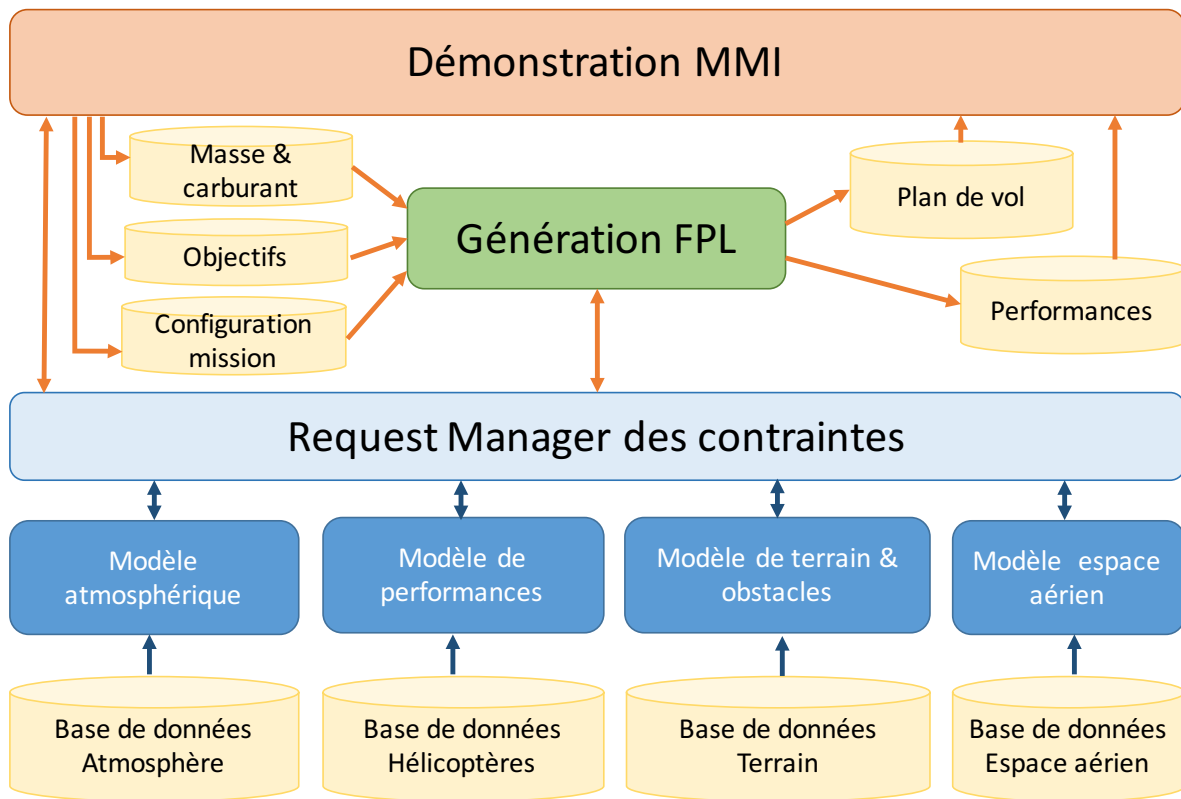


FIGURE 5.2 – Architecture de l'outil de démonstration

L'outil de planification de vol DTANAV permet de calculer un plan de vol en considérant un certain nombre de paramètres d'entrée et de contraintes liées à l'environnement et à l'appareil et une fonction objectif à minimiser. DTANAV permet d'obtenir différents profils de plan de vol selon le type de la mission. Dans cette version, l'outil permet uniquement l'optimisation de la distance du plan de vol. Afin de permettre l'optimisation d'autres critères tel que le temps de vol ou le carburant consommé, la fonction coût devra être couplée avec le modèle de performances que nous détaillerons dans la Section 5.4.

Afin d'élaborer un plan de vol, l'outil procède en deux phases :

► **Le prétraitement** : consiste à la modélisation de l'espace 3D lors de la phase statique afin de préparer le terrain pour la planification de vol. Les étapes de prétraitement sont les suivantes :

1. Extraction des données à partir des bases de données géographiques et météorologiques et des paramètres de la mission saisis par l'utilisateur.
2. Discrétisation horizontale multi-altitude sous forme de différents paliers 2D (exemple : séparés de $150m$) (entre alt_{min} et alt_{max}) en utilisant une structure irrégulière de type Quadtree. Il s'agit d'un maillage adaptatif selon la densité des obstacles permettant

de réduire la taille mémoire par rapport à une structure régulière. La méthode utilisée est itérative et permet d'ajuster la précision du maillage selon les performances de l'appareil.

3. Construction d'un graphe de navigation 2D pour chacun des niveaux de vol en fonction du paramètre *Radius Factor*. Ceci permettra d'ajuster le degré de connexité du graphe selon le temps de prétraitement disponible et donc agit sur la qualité de la solution trouvée en appliquant le routage.
4. Raccordement entre les paliers (consécutifs, pris deux à deux) en choisissant des points de montée d'un niveau à un autre et en construisant des arcs verticaux qui respectent les limitations angulaires et qui évitent les obstacles. Les arcs de raccordement entre un nœud et le palier au-dessus constituent un cône creux permettant de respecter les limitations angulaires liées à l'appareil.
5. Construction du graphe complet en fusionnant les graphes de navigation 2D et les arcs de raccordement verticaux. Les arcs de ce graphe constituent l'ensemble des chemins possibles permettant l'évitement d'obstacle dans l'espace de solution.

► **Le routage :** après le prétraitement, la phase de routage peut être appelée en statique (préparation de mission) ou en dynamique (replanification) plusieurs fois. En effet, l'application de l'algorithme de routage sur le graphe de navigation permet de calculer un chemin entre deux points de l'espace 3D tout en optimisant la distance. Le routage prend en considération des paramètres comme les intervalles d'angles de montée/descente et/ou le plafond de navigation. L'algorithme intègre un calcul géométrique en explorant les arcs au lieu des nœuds ce qui garantit le respect des contraintes angulaires de la mission et de l'appareil. Par ailleurs, il est possible de coupler le routeur 3D avec le modèle de performance ce qui assurera une exploration permettant de mettre à jour le coût de chaque arc visité (temps de vol, masse). Le profil et la qualité de la solution générée dépendent de l'ensemble des paramètres qui ont été saisis. Lors du prétraitement, nous soulignons que l'augmentation des valeurs du *Radius Factor*, les limitations angulaires ainsi que la précision de la discrétisation permet d'obtenir un maillage plus fin et un graphe plus dense ce qui agit sur l'amélioration de la qualité de la solution obtenue au détriment du temps d'exécution de la phase de prétraitement. La durée de prétraitement dépend du temps disponible pour la préparation de la mission d'où l'intérêt de ce paramétrage introduit dans notre approche.

La Fig. 5.3 représente l'IHM de l'outil DTANAV. Cette IHM est formée principalement des composantes suivantes :

- la zone de saisie des paramètres d'entrée,
- le profil vertical pour l'affichage du chemin en considérant une coupe verticale du terrain pour l'ensemble des points de passage,
- le widget Marble pour l'affichage du chemin sur la carte du terrain en 2D.

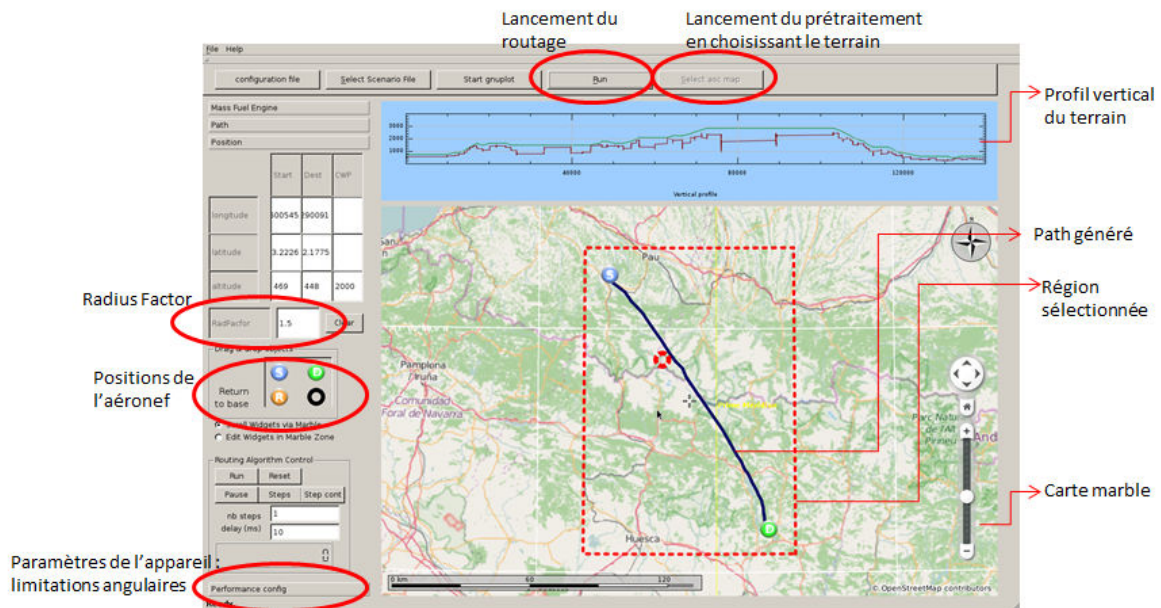


FIGURE 5.3 – IHM de DTANAV

Dans la suite, nous montrons un scénario d'utilisation de l'outil.

5.3.2 Utilisant de l'outil DTANAV

Les paramètres de la mission sont saisis par l'utilisateur (Fig. 5.3) avant de lancer le prétraitement. Une fois, le prétraitement effectué, l'utilisateur peut modifier certains paramètres (les limitations angulaires, le *ceiling*, zone de danger, etc.) à chaque routage jusqu'à obtenir le profil du plan de vol souhaité. En effet, les paramètres de la mission sont :

- Le terrain à explorer : ce paramètre aide à sélectionner la carte de navigation adéquate à partir d'une base de données géographique.
- Les paramètres de l'appareil et le type de la mission
- Le point de départ, la destination et les zones de passage : Autre que les points de départ et d'arrivée, l'utilisateur peut imposer un point de passages lors de la mission.
- Le plafond de navigation (*ceiling*) : ce paramètre permet d'éviter les zones de danger liées aux conditions météorologiques ainsi que les altitudes de vol qui présentent un risque de givrage. Ce paramètre nous permet aussi d'obtenir des profils de chemin favorisant le vol en palier (plus confortable pour certains types de mission).
- Les limitations angulaires en montée, en descente et de changement de direction : ces paramètres sont déduits à partir du modèle de performance de l'appareil mais l'opérateur peut restreindre ces limitations afin d'éviter les fortes sollicitations qui ne sont pas

souhaitables dans certains types de mission telle qu'une mission VIP ou assistance médicale. Dans d'autres types de mission (militaire), les capacités maximales de l'appareil sont souvent considérées dans la phase de prétraitement.

En plus des paramètres de la mission, l'utilisateur peut intervenir pour fixer les paramètres de l'algorithme :

- Le palier de discrétisation vertical : ce paramètre dépend du type de l'aéronef utilisé. Dans notre travail, nous avons considéré des aéronefs de type hélicoptère et en coordination avec notre partenaire industriel ce paramètre est fixé à $150m$.
- La précision de discrétisation des Quadrees 2D : ce paramètre est calculé automatiquement en fonction du palier de discrétisation vertical et des angles de montée et de descente. Cependant, l'utilisateur peut exiger une certaine précision lors de la discrétisation 2D des plans horizontaux.
- Le *Radius Factor* : il est utilisé lors de la phase de prétraitement. la valeur par défaut permet de garantir un bon compromis entre le temps de calcul et la qualité de la solution. Toutefois, l'utilisateur peut choisir une autre valeur qui s'adapte selon les exigences de la mission.

Dans la suite, nous décrivons un scénario d'utilisation de DTANAV. L'utilisateur commence par saisir les paramètres et les données de prétraitement. Nous avons choisi la carte des « Pyrénées » (voir Fig. 5.4). Pour un premier exemple, nous avons gardé les paramètres par défaut de l'outil. Ces paramètres sont indispensables pour la discrétisation de l'espace 3D et la construction du graphe.

- *Radius Factor* = 1.5,
- Angles de montée et de descente : $\alpha_{desc_min} = 1^\circ$, $\alpha_{desc_max} = 15^\circ$, $\alpha_{climb_min} = 1^\circ$, $\alpha_{climb_max} = 15^\circ$

► **Étape 1 : Prétraitement** Cette étape englobe :

- La phase discrétisation multi-altitude du terrain sous forme de Quadrees jusqu'à une précision donnée sur les différents paliers séparés d'une distance de $150m$,
- La construction d'un graphe horizontal sur chacun des paliers selon le paramètre *Radius Factor*,
- L'établissement de liens entre les différents graphes 2D en construisant les arcs de raccordement selon les limitations angulaires (α_{desc_min} , α_{desc_max} , α_{climb_min} , α_{climb_max}) et la fusion entre les graphes 2D et les arcs de raccordement. La durée de prétraitement relative à cet exemple est égale à $3min$.

► **Étape 2 : Choix des paramètres de routage et recherche du plan de vol**

Dans la Fig. 5.5, nous avons choisi les paramètres de routage : un point de départ S et un

DTANAV : environnement de planification et d'exploration de chemins aériens

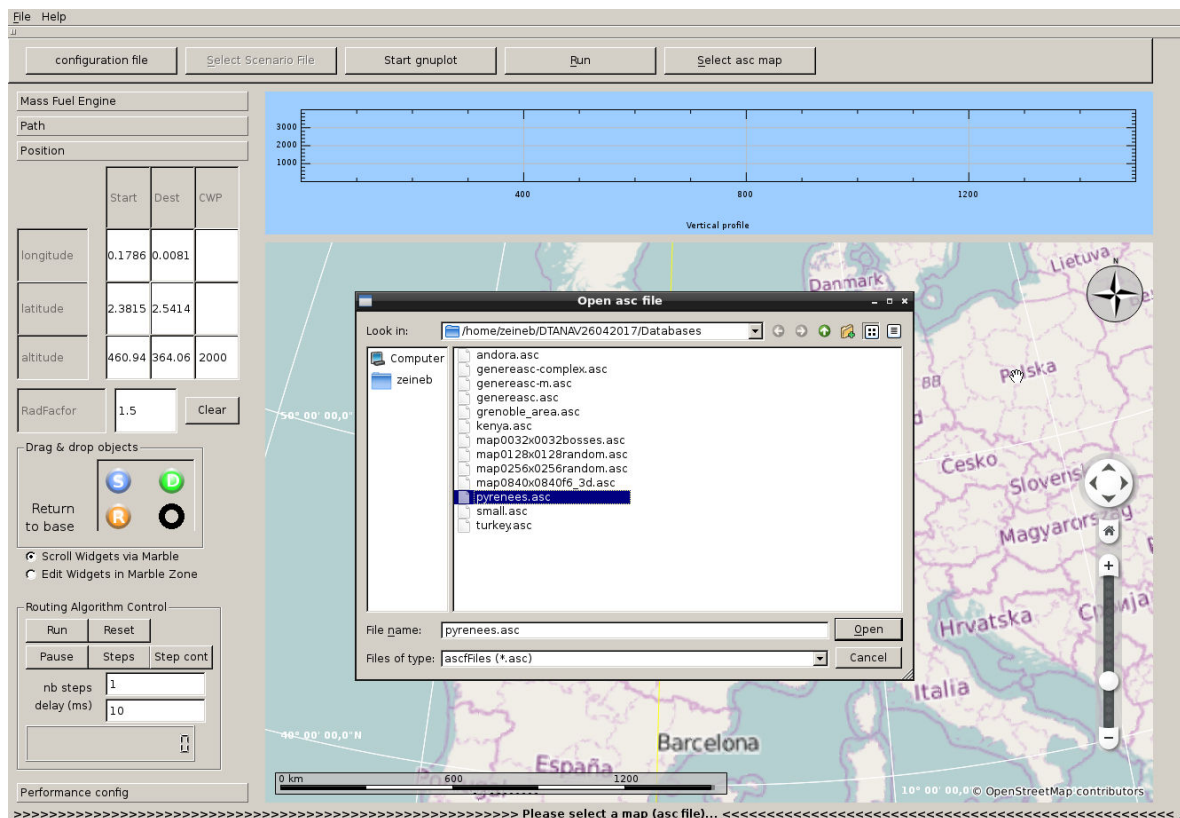


FIGURE 5.4 – Lancement de la planification

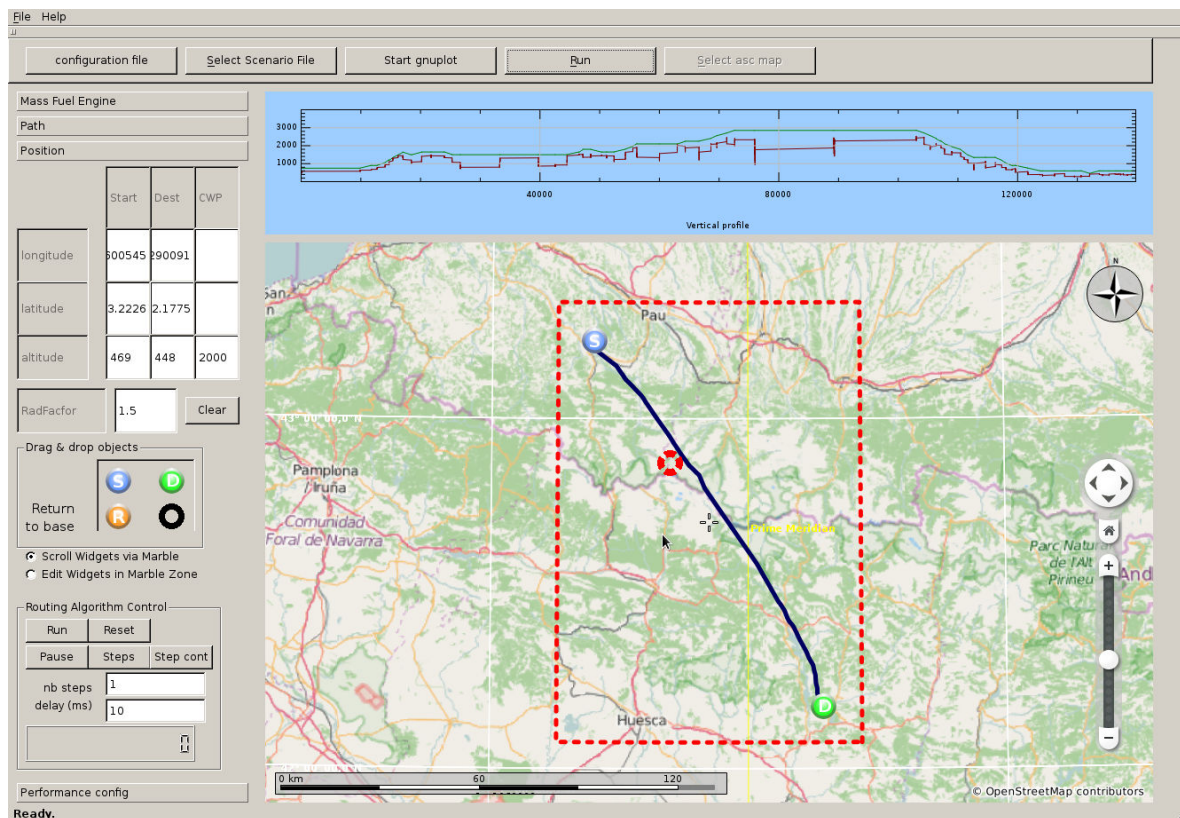


FIGURE 5.5 – Routage 3D avec les paramètres par défaut de l'outil

point d'arrivée D ayant respectivement des altitudes de 469m et 448m. Nous avons utilisé les valeurs des paramètres suivants :

- $Ceiling = 0$ (pas de ceiling),
- $Angle_{\text{changement de direction}} = 45^\circ$,
- Angles de montée et de descente sont : $\alpha_{desc_min} = 4^\circ$, $\alpha_{desc_max} = 15^\circ$, $\alpha_{climb_min} = 4^\circ$, $\alpha_{climb_max} = 15^\circ$.

Puis, nous avons lancé le routage avec le « bouton Run ». À partir du fichier plan de vol généré « path1.dat », nous avons obtenu :

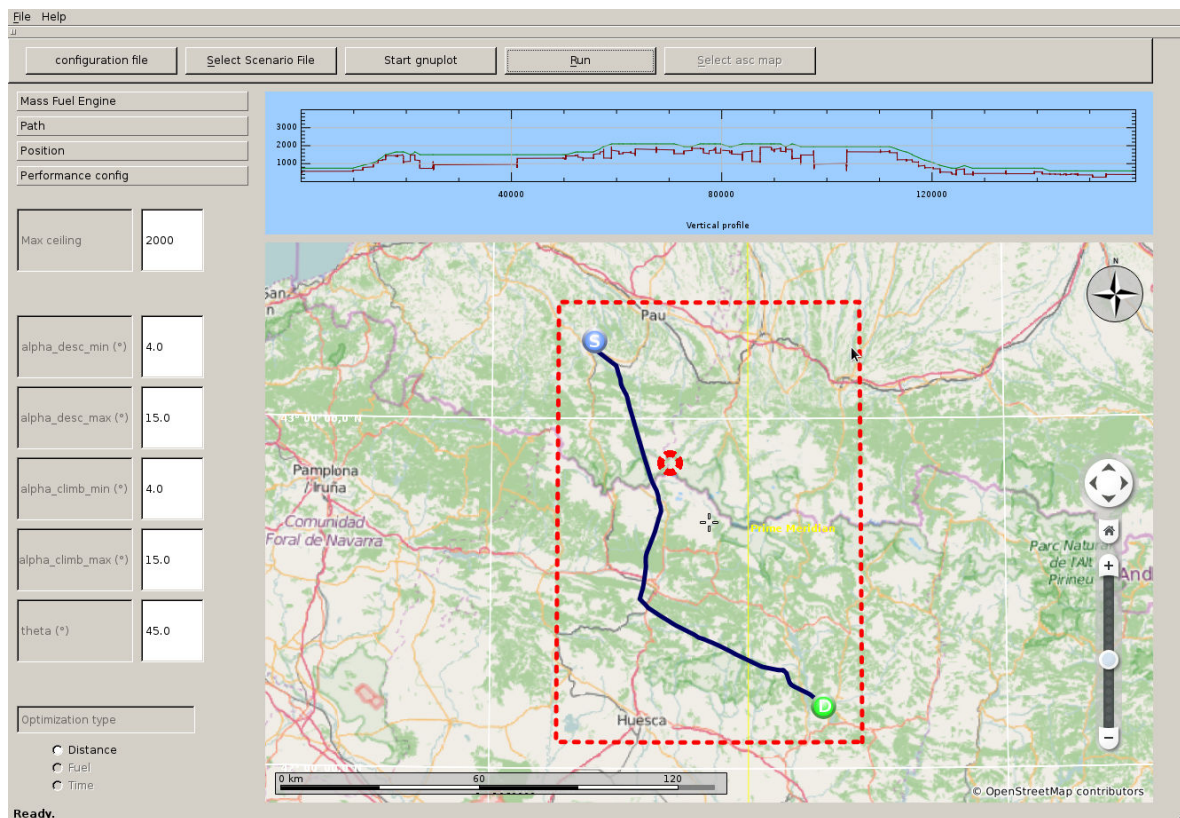
- Le nombre de points de passage est égal à 74, la distance du plan de vol est de 140707m.
- La durée de routage est égale à 12sec.

Comme affiché sur le profil vertical de la Fig. 5.5, le chemin atteint 3000m d'altitude. Cette sortie explique la nature du plan de vol générée. En effet, le chemin représente une ligne droite vers la destination tout en procédant à une phase de montée jusqu'à atteindre 3000m pour éviter les obstacles suivie d'une descente jusqu'à la destination. La durée de calcul de 12sec traduit une rapidité lors de l'exploration du graphe afin d'atteindre la destination. Nous soulignons que la taille de notre graphe est de l'ordre d'une centaine de millions d'arcs qui couvrent toute la zone géographique du terrain en considérant l'altitude maximale détectée (en plus de la marge de sécurité).

Par la suite, nous avons ajouté l'option « ceiling » en saisissant la valeur 2000m dans le champ approprié. Nous avons maintenu les mêmes paramètres et données du routage antérieur. Contrairement au chemin précédent, le plan de vol obtenu contourne les obstacles afin de satisfaire la nouvelle contrainte comme illustré dans la Fig. 5.6, c'est à dire trouver une solution qui ne dépasse pas 2000m d'altitude. Le chemin trouvé est de qualité légèrement inférieure que le précédent : Le nombre de points de passage est égal à 83 et la distance est un peu plus élevée (159450m). Par ailleurs, le temps de routage passe à 59sec au lieu de 12sec. En effet, la condition « ceiling » a poussé l'algorithme à explorer un nombre plus important d'arcs jusqu'à atteindre la destination ce qui est à l'origine du temps de calcul supplémentaire.

Ensuite, nous avons modifié les limitations angulaires (Fig. 5.7) en gardant le ceiling de 2000m. Les angles de montée et de descente sont fixés à $\alpha_{desc_min} = 10^\circ$, $\alpha_{desc_max} = 15^\circ$, $\alpha_{climb_min} = 10^\circ$, $\alpha_{climb_max} = 15^\circ$. Cette restriction angulaire a considérablement réduit le nombre d'arcs explorés ce qui a agit sur le temps de calcul (16sec au lieu de 59sec) au détriment de la qualité du chemin (la distance passe à 164663m et nombre de points de passage est de 89). Ce chemin se caractérise par de fortes pentes et une tendance à contourner les obstacles vu la réduction de l'ensemble d'arcs de raccordement à explorer. Par ailleurs, la restriction sur les arcs verticaux a favorisé le vol en palier ce qui est visible sur le profil vertical. Ce type de chemin convient aux missions qui favorisent le confort des passagers.

Dans l'étape suivante, nous avons réduit le ceiling à 1000m. L'algorithme ne trouve pas de solution ce qui se traduit par le message box « no solution found » dans la Fig. 5.8.

FIGURE 5.6 – Routage avec un *ceiling* de 2000m

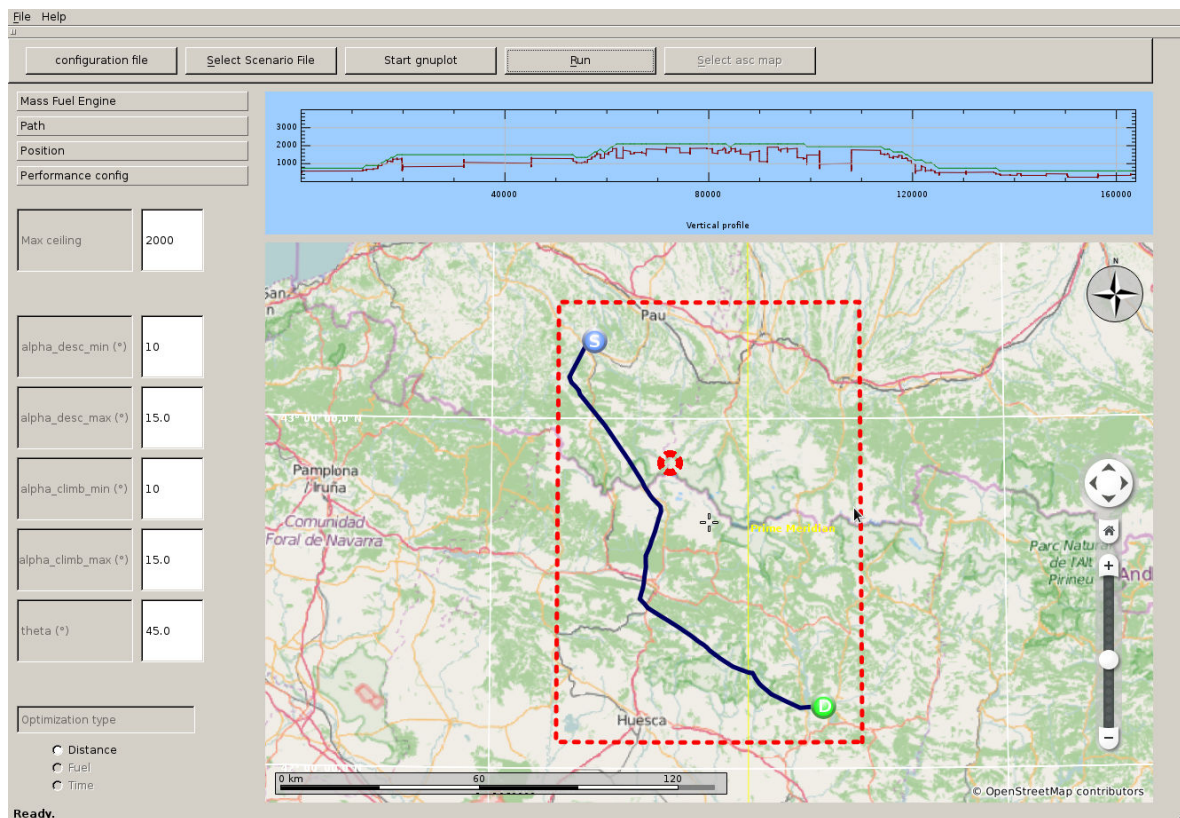
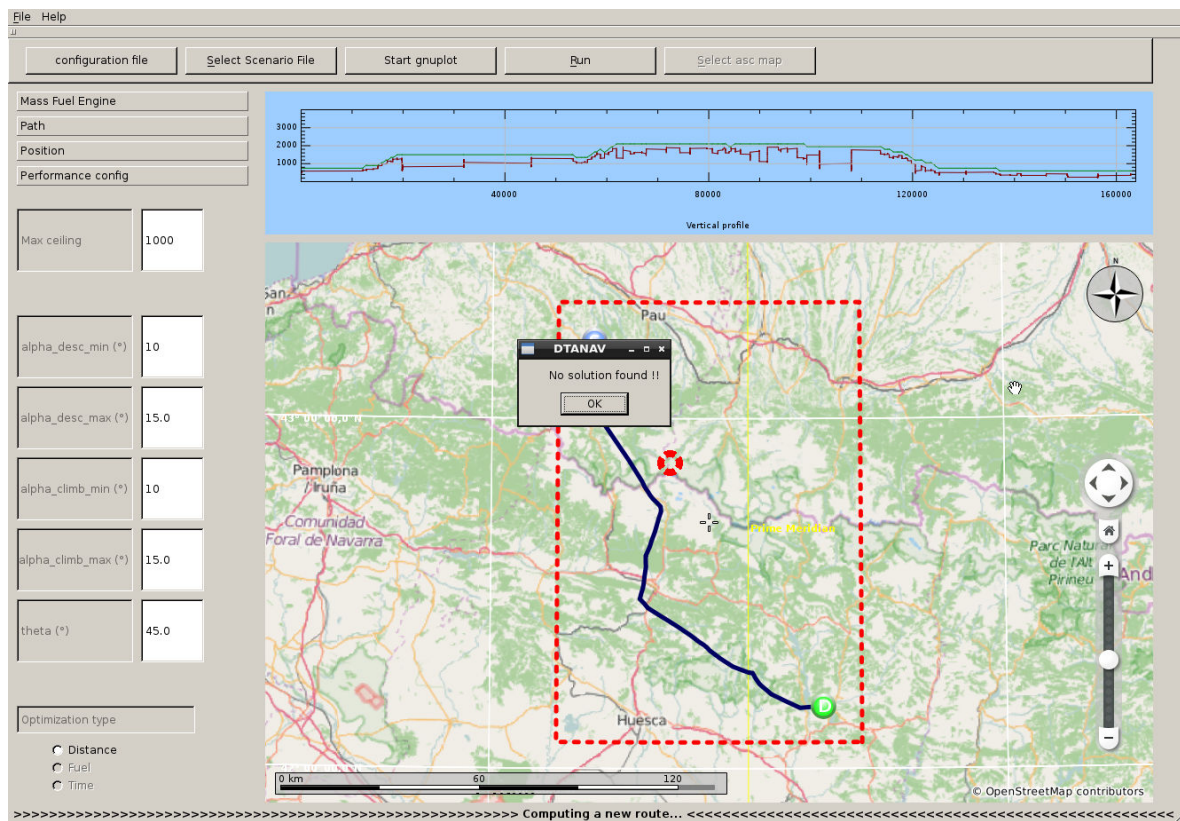


FIGURE 5.7 – Routage avec *ceiling* et modification des limitations angulaires

FIGURE 5.8 – Routage avec un *ceiling* de 1000m

105TANAV : environnement de planification et d'exploration de chemins aériens

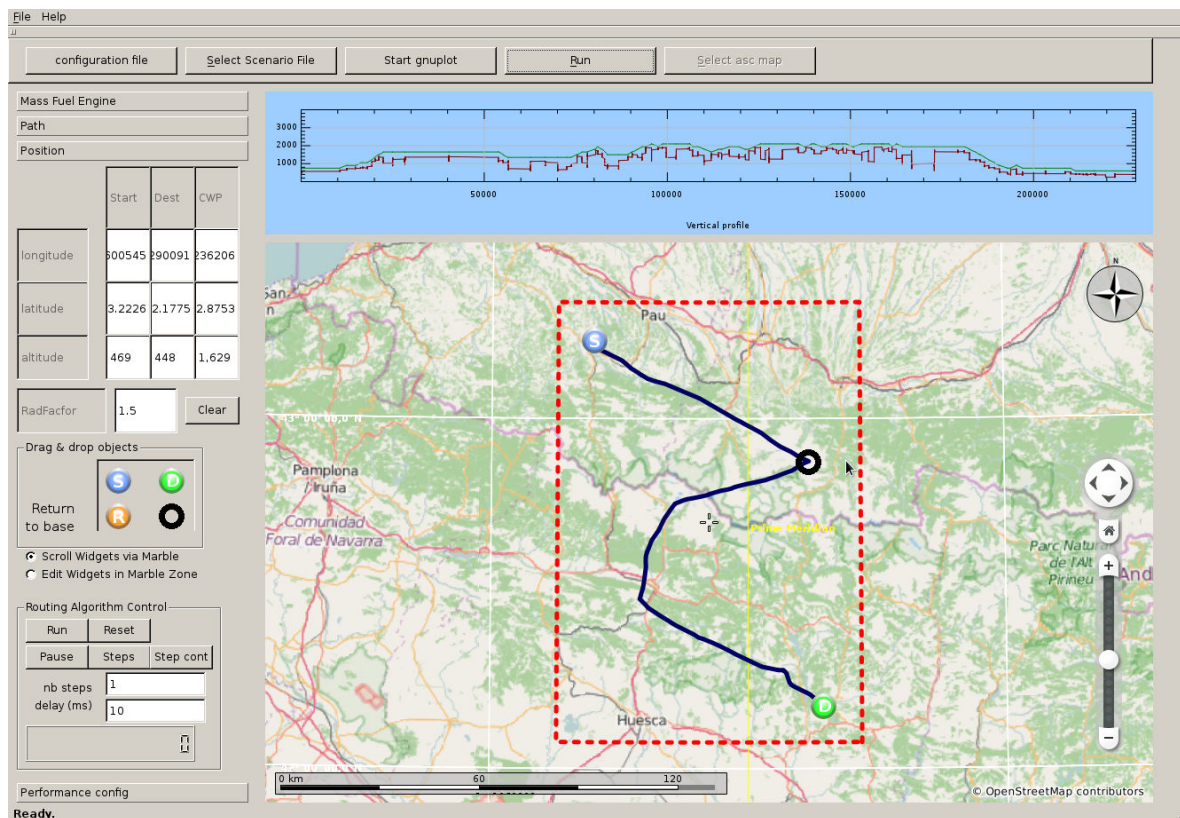


FIGURE 5.9 – Routage avec *Waypoint* et un *ceiling* de 2000m

Pour obtenir le résultat de la Fig. 5.9, nous avons ajouté un « *Waypoint* » (à une altitude de 1629m) tout en remettant le *ceiling* à 2000m et les angles de montée et de descente à : $\alpha_{desc_min} = 10^\circ$, $\alpha_{desc_max} = 15^\circ$, $\alpha_{climb_min} = 10^\circ$, $\alpha_{climb_max} = 15^\circ$. Le chemin est généré en 54sec et contient 117 points de passage. Sa distance est égale à 230042m.

En guise de conclusion que notre outil DTANAV nous a permis d'explorer et de générer automatiquement des plans de vol qui respectent différents types de contraintes. Les temps d'exécution ainsi que la qualité des solutions générées sont très satisfaisants en considérant la taille et la complexité des terrains étudiés. L'efficacité de notre outil DTANAV pourra être exprimée en termes de productivité du personnel en charge de préparation de missions et surtout en termes de sécurité en vol en produisant des solutions qui prennent en compte les paramètres du terrain, l'appareil, la mission et l'environnement.

5.4 Modèle de performances

Dans cette section nous allons étudier un exemple de modèle de performances qui peut être couplé avec la phase d'exploration afin de trouver des chemins aériens qui optimisent différents critères. Notre système de planification de vol décrit dans le Chapitre 3, permet d'assurer un vol en toute sécurité qui :

- optimise un critère donné. Dans la version décrite dans la Section 5.3, il s'agit de la distance,
- respecte les limitations angulaires de l'appareil et le type de la mission afin d'assurer la sécurité en vol et le confort des passagers.

Dans cette section, nous abordons une étude d'un modèle de performances fourni par Airbus Helicopters. Il sera appliqué afin de déduire :

- les meilleurs angles de montée et de descente afin de réduire le temps de vol ou la consommation de carburant,
- les coûts liés à chaque segment de vol selon un critère donné (temps de vol, distance et carburant consommé).

Tout d'abord, nous montrons la façon avec laquelle nous considérons le modèle de performances afin d'évaluer le coût d'un segment de vol selon différents critères. Par la suite, nous présentons une étude expérimentale qui a permis d'extraire les limitations angulaires pour un aéronef donné. La méthode peut être généralisée afin d'être appliquée sur n'importe quel type d'appareil.

5.4.1 Évaluation du coût d'un segment de vol selon différents critères

Dans cette section, nous étudions une fonction de performances permettant d'évaluer le coût d'un segment en termes de temps de vol, de vitesse et de carburant consommé. Pour ce

faire, il suffit de mettre à jour le graphe lors de la phase d'exploration. En effet, la structure de l'algorithme de routage permet d'accorder une étiquette à chaque arc exploré. Cette étiquette correspond à un ou plusieurs coûts (distance, masse, vitesse, etc.). L'ensemble de ces critères peuvent être mis à jour lors de l'exploration même si l'optimisation du chemin est mono-critère. Pour ce faire, il suffit de coupler le modèle de performance avec la fonction objectif. La version actuelle de l'outil de démonstration permet la planification d'un vol qui optimise la distance. Toutefois, d'autres critères peuvent être imposés afin de générer des plans de vol qui optimisent d'autres critères comme le temps de vol ou le carburant. Le modèle est grossier mais son comportement général présente les bonnes tendances. Il permet la mise au point d'algorithmes d'optimisation de plans de vol en 3D. Ce modèle tient compte :

- des paramètres qui représentent les constantes propre à l'appareil,
- des entrées comme la distance horizontale entre deux nœuds, l'écart vertical, la masse de l'appareil.

Il génère une sortie qui est bien la vitesse permettant de relier entre les nœuds et met à jour la variation de la masse totale de l'appareil. La vitesse est calculée à partir de l'intersection entre une droite (la pente entre les 2 nœuds) et l'enveloppe du domaine de vol comme schématisé dans la Fig. 5.10 :

- En vert : une parabole (pour une puissance constante) approximant le comportement de la limite du domaine de vol sauf pour les basses vitesses (non valide aux forts angles)
- En rouge : courbe de limite du domaine de vol pour une puissance donnée
- Une limitation de vitesse verticale minimale
- Une limitation de vitesse horizontale maximale
- En bleu : la pente recherchée

Dans le cas d'une montée, seulement les faibles pentes inférieures à 15° sont tolérées. La vitesse est obtenue en fonction de l'intersection entre la pente (bleu) et la performance (vert). Dans le cas d'une pente négative, plusieurs scénarios peuvent se produire. en effet, il faut tenir compte d'une vitesse horizontale maximale VHZ_{max} (trait pointillé vertical) et d'une vitesse verticale minimale (trait pointillé horizontal) Les différents scénarios sont traités par la formule suivante (voir pentes pointillés en bleus) :

- La solution 1 est obtenue via l'intersection pente/parabole si $(VH < VHZ_{max})$ et $(VZ > VZ_{min})$
- La solution 2 est obtenue via l'intersection pente/limite VHZ_{max} si $(VH > VHZ_{max})$ et $(VZ > VZ_{min})$
- La solution 3 est obtenue via l'intersection pente/limite VZ_{min} si $(VZ < VZ_{min})$

Selon la mission, l'utilisateur peut lancer une planification qui optimise l'un des critère. L'algorithme d'exploration met à jour le coût de l'arc examiné. Un arc exploré est accompagné d'une étiquette permettant de déterminer les coûts relatifs à chacun des critères.

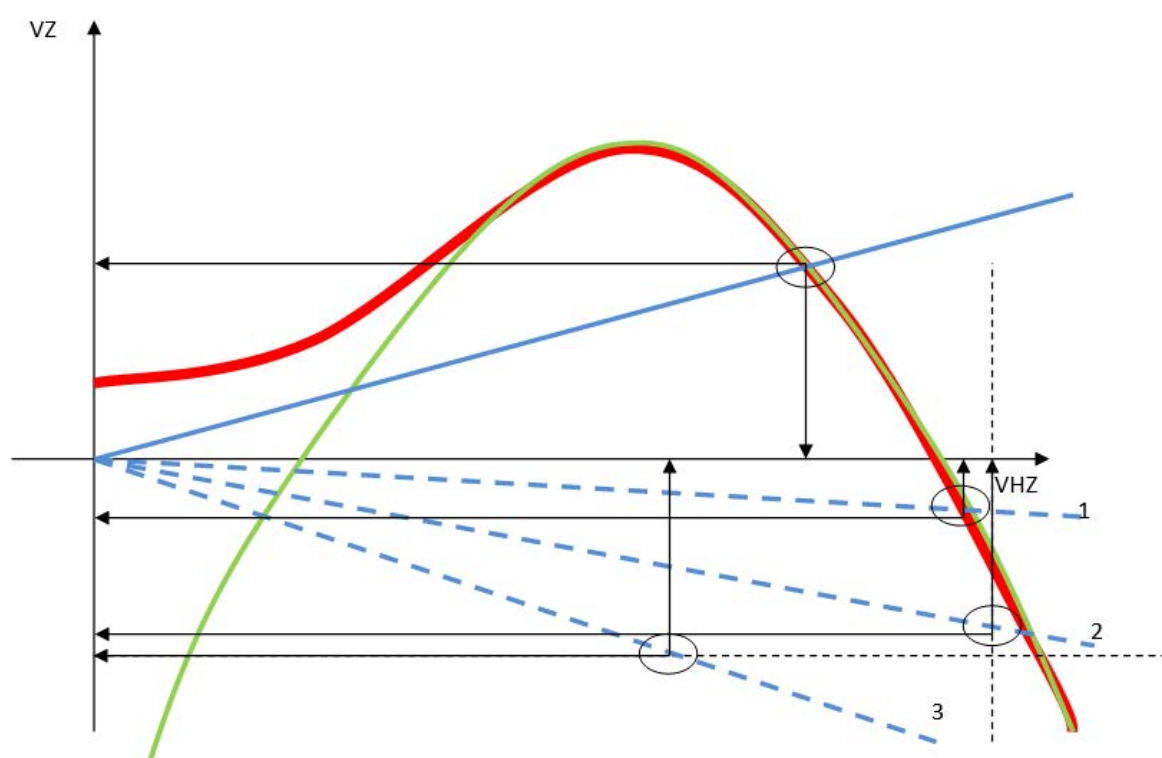


FIGURE 5.10 – Fonction de performances étudiée

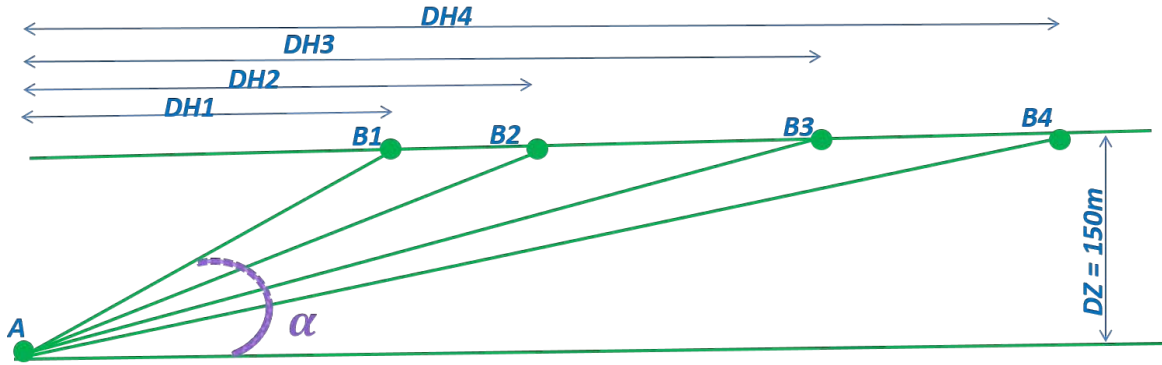


FIGURE 5.11 – Étude de l'angle de montée

5.4.2 Dédution des angles de montée/descente

Dans cette section, nous visons à déterminer les meilleurs angles de montée et de descente afin d'offrir un vol stationnaire qui minimise le temps de vol en s'appuyant sur la fonction de performances. Nous soulignons que le graphe de navigation 3D pourra être construit en exploitant les capacités angulaires maximales de l'appareil. Cependant dans la phase du routage, nous choisirons les meilleurs angles de montée/descente en explorant le graphe de navigation afin de satisfaire un critère donné comme minimiser le temps ou favoriser le confort des passagers. Dans certaines missions ou situations d'urgence, il est indispensable d'avoir recours à des fortes sollicitations angulaires pour éviter un obstacle dynamique par exemple. Pour cette raison, nous construisons un graphe 3D très riche dans la phase de prétraitement. Par ailleurs, les intervalles d'angles permettent à l'utilisateur d'obtenir le profil de vol souhaité selon le type de la mission.

Nous avons considéré l'exemple qui calcule le temps de vol entre deux points séparés par une hauteur de $150m$ et une distance horizontale de $1200m$. Pour ce faire, nous avons fixé un palier de $150m$ et nous avons fait varier l'angle de montée/descente et calculer le coût du segment permettant de franchir le palier (voir Fig. 5.11). Ce coût est calculé via le modèle de performance. La Fig. 5.12 montre que pour une montée, l'angle qui permet de franchir le palier dans un temps minimum est de $10,62^\circ$. Le temps de vol est élevé pour les faibles pentes de montée. Par contre, ce temps diminue en atteignant un certain seuil. En effet, à partir d'un angle de descente de 8.5° le temps de vol affiche une baisse considérable. Ce temps reste constant jusqu'à l'angle de 15° qui représente une limite maximale afin d'assurer un vol stationnaire sans provoquer de fortes sollicitations en montée.

Par la suite, nous avons étudié le cas d'une descente (voir Fig. 5.13) en considérant le même scénario (palier de $150m$ et une distance horizontale de $1200m$). Nous avons relevé le résultat de la Fig. 5.14 et nous avons constaté que le temps de vol est élevé pour les pentes faibles. À partir d'un angle de 3.5° , le temps décroît jusqu'à devenir constant. La variation

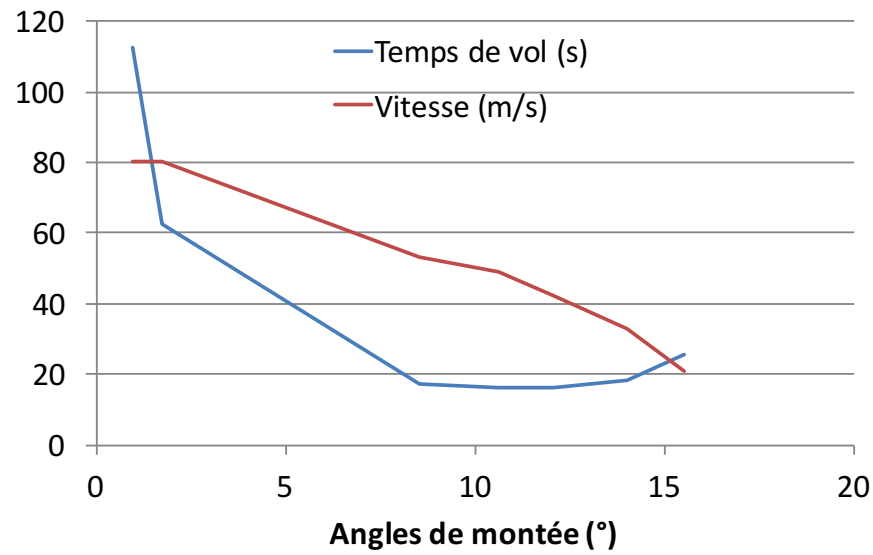


FIGURE 5.12 – Temps de vol et vitesse en fonction de l'angle de montée

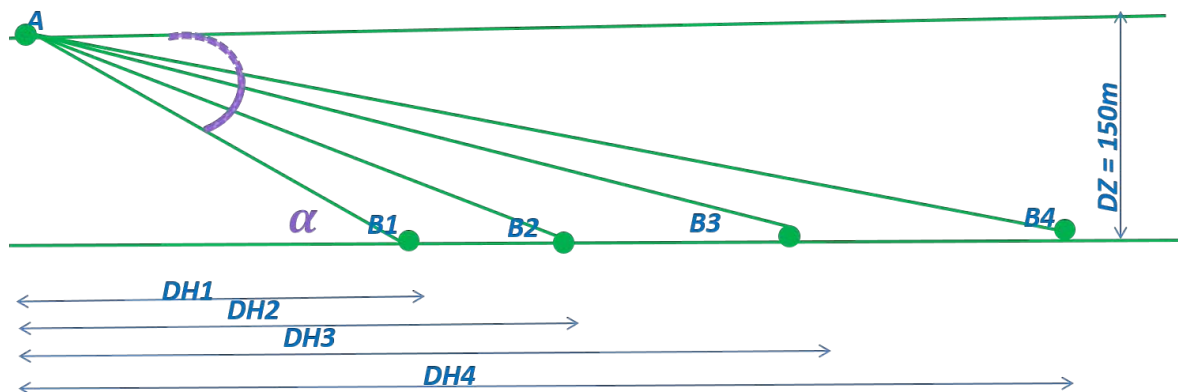


FIGURE 5.13 – Étude de l'angle de descente

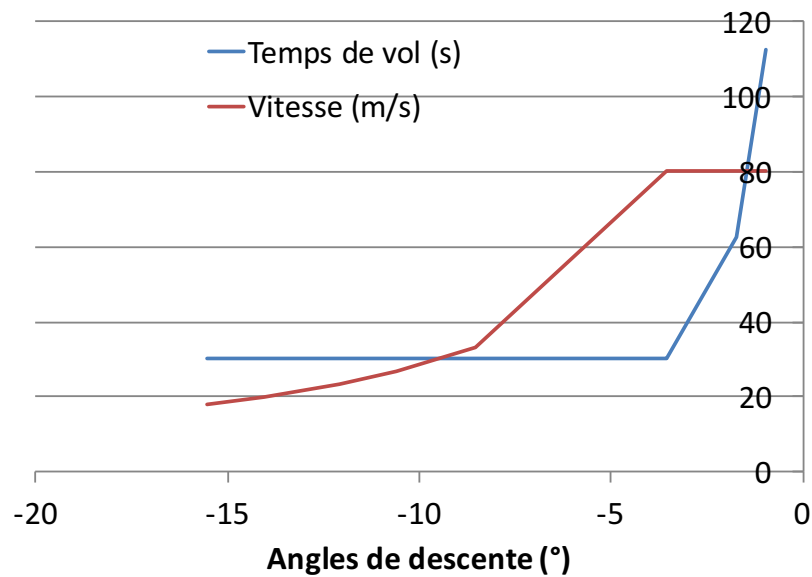


FIGURE 5.14 – Temps de vol et vitesse en fonction de l'angle de descente

de la vitesse suit le comportement du temps de vol.

Sur la Fig. 5.15, nous comparons les temps de vol en montée et en descente en fonction de la pente. Pour les faibles angles, les temps de vol en montée et descentes présentent le même comportement. Au delà de 8.5° , le temps de vol en montée devient plus élevé par rapport à la descente. En effet, ce comportement s'explique par les limitations liées aux descentes à faible vitesse. Les descentes raides posent deux problèmes :

- Le vortex pour les faibles vitesses d'avancement ($< 40kts$),
- La décompression des oreilles des passagers qui peut être douloureuse en dessous de $-500ft/min$.

Le modèle de performance, étant une brique de base dans le système avionique, représente un fort impact sur la conception de notre approche de préparation de missions ainsi que dans la replanification dynamique. En effet, le graphe de navigation permettant de modéliser l'espace 3D selon l'approche multi-altitude repose sur un calcul d'angles de montée/descente déduit de la fonction performance. De même, assurer un vol stationnaire qui optimise la distance, le temps de vol ou le carburant consommé lors de la phase dynamique doit être couplé avec le modèle de performances qui se charge d'évaluer le coût de chaque arc exploré et donc contribuer à l'optimisation du plan de vol.

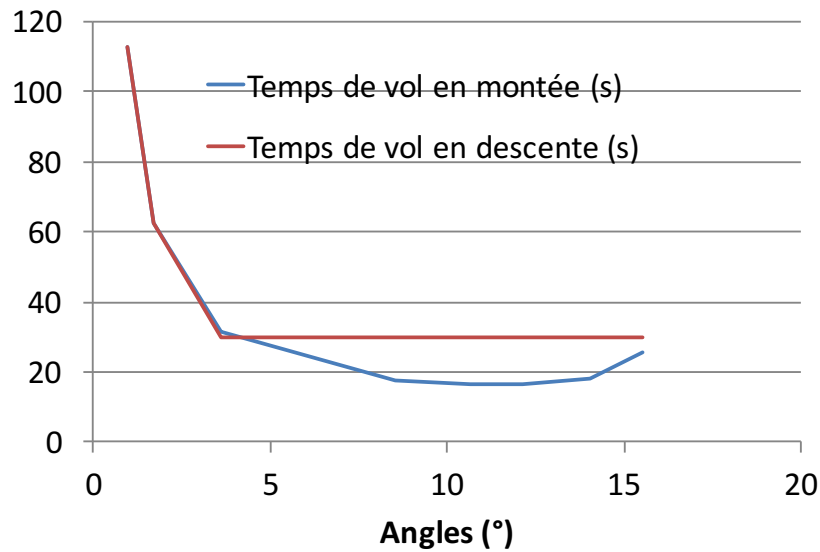


FIGURE 5.15 – Temps de vol en montée et en descente

5.5 Scénarios de replanification dynamique

Jusqu'à maintenant, nous avons illustré l'utilisation de DTANAV dans la phase de planification de missions. Dans cette section, nous allons étudier quelques scénarios qui nécessitent de la replanification dynamique en cours du vol afin de faire face à un obstacle dynamique ou éviter une situation d'urgence telle qu'une panne moteur ou un territoire hostile lors des missions militaires. La replanification est gérée lors de la phase de routage en considérant les nouvelles contraintes de l'appareil et de l'environnement afin de calculer un plan de vol en temps réel. Dans notre travail, nous allons considérer trois cas de replanification à savoir :

- **Le court terme** : cette situation peut être liée à l'apparition des obstacles statiques ou dynamiques non prévus dans la phase de préparation de missions. À titre d'exemple, nous citons l'accident d'un hélicoptère du groupe Airbus Helicopters qui s'est produit dans les gorges du Verdon en 2012 lors d'un accrochage avec un câble d'électricité non déclaré sur la carte du terrain. Malheureusement, le personnel en charge de la préparation de la mission n'a pas tenu compte de cet obstacle statique et cet accrochage a causé la mort de six personnes. Nous pensons que si l'appareil était équipé d'un système de détection couplé avec une fonction de replanification dynamique, il y aurait eu une possibilité d'éviter ce type d'accident à condition d'une réponse temps réel de notre fonction de routage 3D. De façon plus concrète, si notre appareil assure une mission avec une vitesse moyenne de 120 kts soit 220 Km/h ou bien 61 m/s et nous détectons un obstacle qui se trouve à 200 m ou 300 m , une solution de repli doit être générée dans un temps de l'ordre d'une seconde. L'objectif dans cette situation est

d'éviter la collision avec l'objet statique ou dynamique ensuite reprendre le plan de vol initial. Nous citons comme premier exemple de cas d'urgence qui nécessite une replanification à court terme la perte de visibilité et le risque de collision avec le terrain. Nous soulignons que notre travail consiste à trouver un plan de vol pour une situation d'urgence. Il faut tenir compte également de la phase de génération de trajectoire et l'ensemble de manœuvres correspondantes afin d'éviter l'accident. Une panne moteur représente un deuxième exemple de situation qui nécessite une replanification de ce type. Dans ce cas d'urgence, le pilote doit effectuer une manœuvre d'autorotation pour confronter une perte de puissance partielle ou totale du moteur. Cependant, le pilote a environ 2 secondes pour baisser la commande du collectif, ce qui va permettre aux pales de continuer à tourner même sans moteur. En parallèle, l'équipage doit repérer un lieu sécuritaire afin de poser l'appareil, tel qu'un champ et à ce niveau notre système de replanification dynamique pourra assister le pilote dans cette phase d'atterrissage délicate en lui proposant un corridor de secours. Une autorotation parfaitement exécutée sauvera non seulement les passagers, mais aussi l'hélicoptère qui n'aura subi aucun dommage.

- **Le moyen terme :** nous considérons des situations où la phase de replanification est moins contraignante en termes de temps ou de distance telles que la dégradation de la météo, la visite des points de passage non prévus dans le cas d'une mission de *recherche et de sauvetage*, etc. Nous considérons des temps de replanification de l'ordre de plusieurs secondes ou des dizaines de secondes avec des distances qui peuvent aller de quelques kilomètres jusqu'à 50km par exemple. À titre d'exemple, lors d'une catastrophe naturelle nous avons besoin de déployer des équipes de sauvetage dans les zones les plus touchées, pour cela la mission pourra changer en cours de vol. Dans ces situations d'urgence le pilote aura suffisamment de temps pour interagir avec le système de replanification pour valider la solution générée. Afin de réduire les temps de réponse, une replanification partielle du plan de vol initial sera suffisante pour éviter la zone de danger (par exemple éviter une zone orageuse).
- **Le long terme :** ce type de replanification couvre des situations où la contrainte de temps est moins prioritaire que d'autres tels que l'évitement des fortes sollicitations de l'appareil ainsi que la replanification à longues distances. Nous considérons des temps de génération de plans de vol de l'ordre de quelques minutes et des distances de l'ordre de dizaines ou centaines de kilomètres. À titre d'exemple, nous citons les missions militaires où le plan de vol pourra changer en fonction des circonstances sur un terrain de guerre.

Afin de vérifier si notre environnement est capable de répondre à ces différents types de replanification, nous avons réalisé l'expérience suivante. Nous avons mesuré le temps de génération de solutions en fonction de la distance en considérant des dizaines de tests sur différents terrains et contraintes. Nous avons relevé les résultats de la Fig. 5.16. Pour les dis-

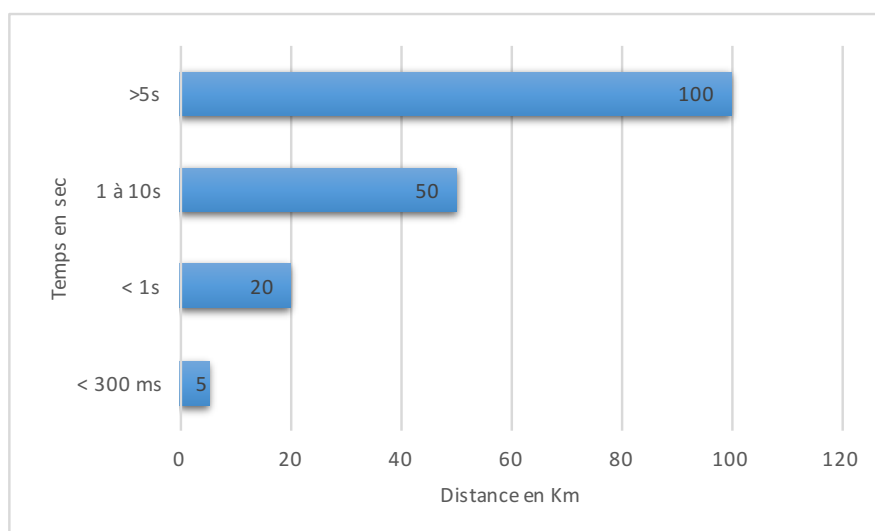


FIGURE 5.16 – Temps de génération de plans de vol en fonction de la distance

tances $< 5Km$, les temps de génération de plans de vol sont toujours inférieurs à $300ms$, ce qui respecte bien les exigences de la replanification à court terme. Concernant les scénarios de replanification à moyen terme, nous avons mesuré des temps inférieurs à $1s$ lorsque la distance ne dépasse pas les $20km$ et des durées comprises entre $1s$ et $10s$ pour des trajets inférieurs à $50Km$. Ces résultats sont très adéquats pour des situations de mauvaises conditions météorologiques ou l'évitement de zones de danger. Pour la replanification à long terme ($> 50Km$) les temps dépassent généralement les quelques secondes et peuvent atteindre $1min$ pour des distances de l'ordre de $200Km$. Nous soulignons que la borne supérieure de chaque intervalle de temps est obtenue en considérant l'ensemble des contraintes sur un chemin donné : les limitations angulaires, le *ceiling* et les zones de danger.

Dans la suite de cette section, nous évoquons deux exemples de replanification dynamique où nous avons fait appel à DTANAV pour générer des solutions de repli dans deux situations d'urgence différentes : une panne moteur et l'évitement d'une zone de danger.

Scénario de panne moteur : Dans ce cas, l'utilisateur doit lancer l'option d'atterrissage d'urgence qui permettra de trouver les points de passage qui peuvent guider le pilote pour atterrir en toute sécurité en minimisant le temps de vol et en respectant les contraintes mécaniques de l'appareil. La Fig. 5.17 montre un scénario simulant le fonctionnement de notre procédé de replanification dynamique en cas d'une panne moteur. Nous considérons que l'hélicoptère était en cours de vol pour une mission initiale (trait pointillé rouge sur la Fig. 5.17) et nous avons reçu soudainement une alarme signalant une anomalie au niveau du moteur. Ceci déclenchera une procédure d'urgence qui va faire appel à l'outil DTANAV qui se chargera de calculer un plan de vol de secours avec les nouvelles contraintes de l'appareil. Le lieu d'atterrissage peut être récupéré à partir de la base de données terrain ou bien saisi par

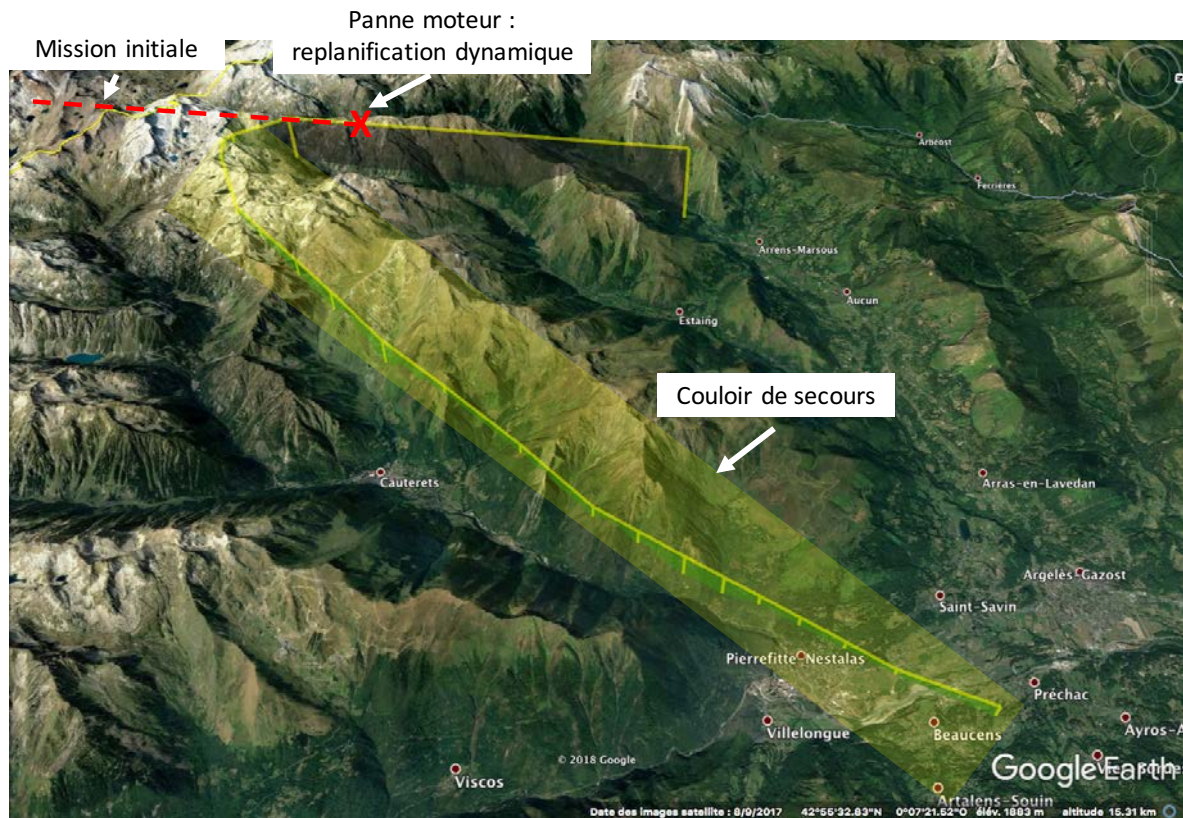


FIGURE 5.17 – Génération d'un plan de vol dans une situation de panne moteur

le pilote à travers une IHM. Nous avons obtenu le résultat affiché sur la carte réelle du terrain 3D (Google Earth) comme illustré sur la Fig. 5.17. Ce résultat est généré au bout de 387ms avec une distance totale de 21Km. La faisabilité réelle de ce chemin pourra être vérifiée avec le pilote automatique de l'appareil, même si ce n'est pas le cas le résultat reste très pertinent. En effet, il permet d'assister le pilote en lui affichant un couloir de secours (rectangle jaune) avec des estimations de performances en termes de distance, temps, etc.

Scénario d'évitement d'une zone de danger : Cette situation peut être liée à de mauvaises conditions météorologiques ou la présence de l'appareil dans une zone de restriction (exemple : territoire hostile pour les missions militaires). La Fig. 5.18 illustre un scénario simulant le fonctionnement de notre procédé de replanification dynamique pour éviter une zone de danger en cours de vol. Le plan de vol initial (chemin jaune sur la Fig. 5.18) a été généré dans la phase de préparation de la mission avec une distance totale de 149Km et au bout de 4s. Au cours de la mission, nous avons reçu une alarme signalant la présence d'une zone de danger (rectangle rouge) par le biais d'un moyen de détection ou de communication. Pour faire face à cette situation, nous allons lancer une phase de replanification dynamique en faisant appel à DTANAV. Notre outil permet à l'équipage de saisir les coordonnées dé-

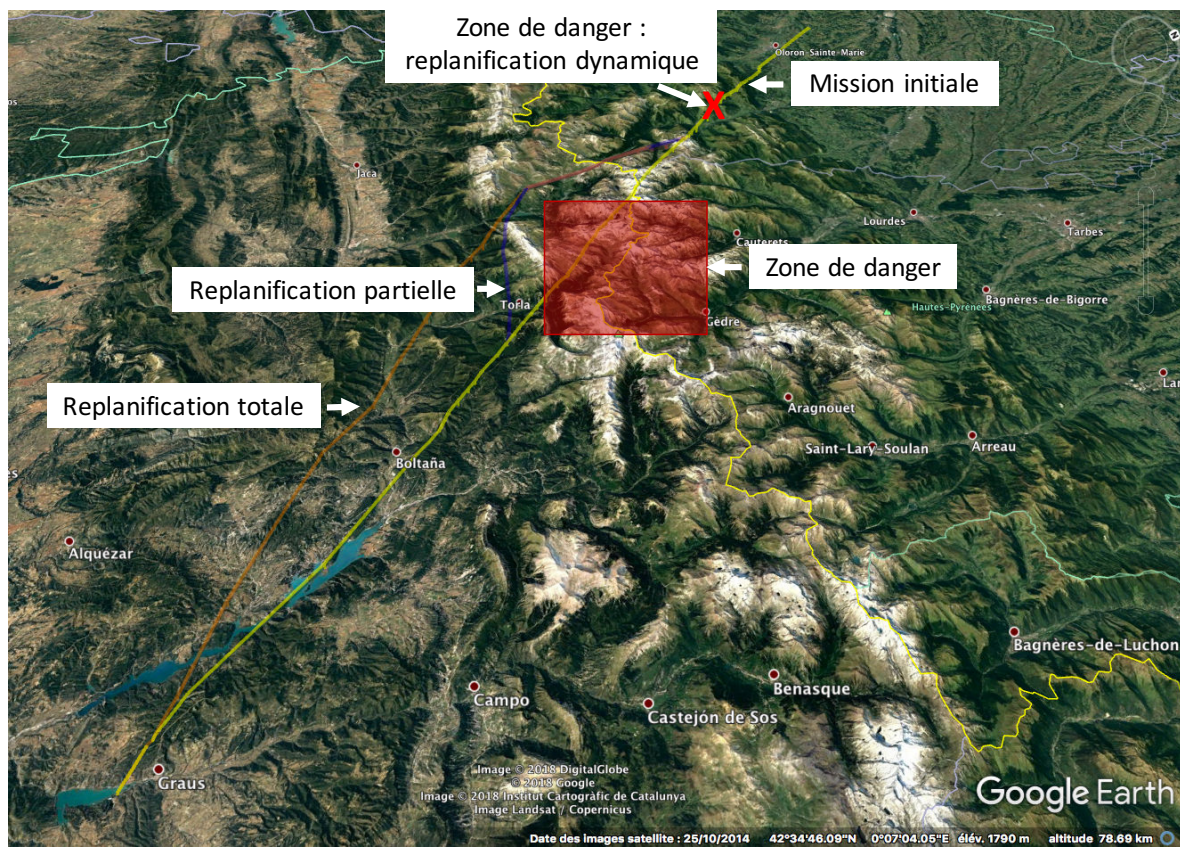


FIGURE 5.18 – Génération d'un plan de vol pour éviter une zone de danger

limitant la zone de danger (sous forme d'un parallélépipède) à travers une IHM ou bien de communiquer directement avec la fonction de routage 3D. Lors du lancement du routage, l'algorithme élimine les arcs du graphe de navigation appartenant à la zone en question et le calcul du nouveau plan de vol peut se faire soit de façon partielle ou totale. Dans le premier cas l'algorithme recalcule un chemin qui va rejoindre le plan de vol initial juste en évitant la zone de danger. En d'autres mots, notre algorithme va considérer cette zone comme un obstacle et il va le contourner. Le chemin bleu illustré sur la Fig. 5.18 représente cette replanification partielle. Sa longueur est de $33Km$ et il est généré au bout de $10s$. Certainement, ce type de replanification ne permet pas d'optimiser la distance parcourue vers la destination ainsi que le temps de vol. Toutefois, nous avons besoin de cette replanification partielle lorsque la situation est critique ou bien en cas d'obligation de passer par les points du chemin initial. Dans le cas d'une replanification totale, nous allons générer une nouvelle solution qui optimise le chemin généré en termes de distance ou de temps depuis la position actuelle et jusqu'à la destination. Le chemin orange illustré sur la Fig. 5.18 représente le résultat d'une replanification totale. Sa longueur est de $109Km$ et il est généré au bout de $8s$. À travers ces scénarios, nous remarquons que le temps de génération d'un plan de vol est fortement impacté par l'ensemble des contraintes que nous appliquons lors de la phase d'exploration du graphe et non seulement par la distance. Néanmoins, nous respectons toujours les délais imposés par la situation d'urgence.

En guise de conclusion, notre outil DTANAV nous a permis de répondre à des besoins variés en termes de replanification dynamique pour améliorer la sécurité en vol à travers sa flexibilité algorithmique.

5.6 Conclusion

Dans ce chapitre, nous avons introduit DTANAV notre environnement de planification de missions qui permet de générer automatiquement un plan de vol selon le procédé présenté dans le Chapitre 3. DTANAV permet aussi l'exploration de plusieurs chemins aériens en fonction des contraintes de la mission. Le paramétrage algorithmique que nous avons introduit dans les Chapitres 3 et 4, nous a permis d'obtenir des profils de vol bien adaptés selon le type de la mission. Notre outil représente une fonction de base soit dans la phase de préparation de la mission ou bien dans le système avionique au sein de l'appareil pour la replanification dynamique dans une situation d'urgence. Nous avons démontré aussi que le couplage de notre outil avec un modèle de performance améliore la qualité des solutions générées et permet l'évaluation de la faisabilité de la solution en se référant au modèle dynamique de l'appareil. Enfin, nous avons illustré des scénarios réels de la replanification dynamique liés à des cas d'urgence. Ce travail ouvre de nombreuses opportunités pour améliorer les résultats obtenus dans le domaine de planification de chemins aériens, quelques perspectives seront proposées

dans le chapitre suivant.

Chapitre 6

Conclusion et Perspectives

6.1	Bilan	122
6.2	Perspectives	124
6.2.1	Système de planification multi-critères	124
6.2.2	Système d'assistance aux pilotes en cas d'urgence	126
6.2.3	Conception d'un nouveau service de transport : Urban Air Mobility	126

6.1 Bilan

Les travaux que nous venons de présenter s'inscrivent dans le cadre du développement d'un nouveau système de planification de chemins aériens en 3D ciblant la phase de préparation de missions ainsi que la phase de replanification dynamique en cas d'urgence. Dans le contexte de notre collaboration avec Airbus Helicopters, nous avons d'abord exposé les défis à relever : principalement la généricité de la solution afin de concevoir un système de planification adaptatif selon le type de la mission, les contraintes de l'appareil, les paramètres de l'environnement, etc. et la maîtrise de la complexité pour la calibration de la qualité des solutions en fonction de la criticité de la situation. La génération automatique d'un chemin aérien 3D est un autre attendu de l'industriel afin de réduire la charge de travail « navigation » et améliorer les performances et ainsi la sécurité en vol. De nos travaux de thèse, il résulte le développement d'un environnement d'exploration de chemins aériens 3D. Cet environnement permet la génération automatique d'un plan de vol à partir d'une spécification de haut niveau de la mission et du terrain à explorer. Sa mise en œuvre a nécessité la proposition d'un nouveau système et procédé de planification de chemins aériens 3D qui permet de générer des profils de vol bien adaptés selon le type de la mission et avec une maîtrise de la complexité temporelle et spatiale.

Pour atteindre cet objectif, nos travaux ont commencé par l'étude de la problématique de planification de chemins aériens en se basant sur quelques procédés industriels proposés par des constructeurs d'équipements aéronautiques ainsi que sur l'état de l'art académique du domaine (Chapitre 2). À travers cette étude, nous avons montré que l'utilisation des méthodes qui s'appuient sur la modélisation complète de l'espace 3D s'avère nécessaire pour répondre au critère de la généricité de la solution selon le type de l'appareil et de la mission. Pour cela, nous avons étudié différentes techniques de modélisation régulières et irrégulières afin d'analyser leur adéquation pour des terrains très étendus. Par la suite, nous nous sommes intéressés aux propriétés (dynamique, *Any-angle*, *Any-time*, etc.) de certains algorithmes de calcul du plus court chemin ainsi qu'à l'évaluation de la complexité correspondante en considérant différentes structures de données. Cette étude de l'existant nous a permis de prendre du recul pour proposer une solution innovante appropriée à notre contexte industriel.

Dans le Chapitre 3, nous avons proposé un procédé innovant pour la planification de chemins aériens 3D dédié à la préparation de mission avant le vol ainsi qu'à la replanification dynamique en cours de vol face à une situation d'urgence. En effet, nous avons réussi à maîtriser la complexité du problème en considérant un procédé de planification en deux phases : la phase de prétraitement et la phase de routage. La première phase se décline en deux étapes : la discrétisation multi-altitude de l'espace et la construction d'un graphe de navigation 3D. À ce niveau, nous avons encore réduit la complexité en décorrélant les deux composantes verticale et horizontale de l'espace lors de la phase de découpage et celle de construction du graphe. Pour ce faire, nous avons découpé l'espace 3D sous forme de plusieurs paliers

2D qui sont discrétisés avec des grilles irrégulières de type Quadtree ce qui nous a permis d'obtenir un maillage adaptatif selon la densité des obstacles et de réduire la complexité en temps et en mémoire. La précision du maillage est fixée afin d'obtenir des angles de montée et de descente respectant les caractéristiques de l'aéronef. Ensuite nous avons construit un graphe de navigation 3D en générant des arcs horizontaux sur chaque palier selon un voisinage variable et des arcs verticaux pour le raccordement entre les différents niveaux selon les angles de montée et de descente tolérés par l'appareil. Cette façon de construire le graphe permet d'obtenir une densité variable, ce qui permet d'ajuster la qualité de la solution en fonction du temps disponible. Lors de la préparation d'une mission, la phase de prétraitement est lancée une seule fois. La deuxième phase de routage de notre procédé permet d'explorer le graphe de navigation et de trouver un chemin qui répond aux paramètres de la mission tout en minimisant une fonction coût (temps, distance, etc.). L'algorithme de routage pourra intégrer différentes contraintes liées à la mission afin d'offrir un profil de vol adapté au type de la mission. La complexité maîtrisée de notre solution en temps et en mémoire nous permet d'intégrer ce procédé soit au niveau d'une station au sol ou bien au niveau de l'architecture du système avionique de l'aéronef. En résumé, notre solution a répondu aux principaux défis à savoir la générique de la solution et la maîtrise de la complexité.

Par la suite, nous avons étudié et implémenté différentes méthodes de discrétisation de l'espace appliquées à la planification de vol 2D et 3D dans le Chapitre 4. Parmi les solutions étudiées, nous citons, les grilles régulières, Quadtree/Octree et le découpage multi-altitude avec une étude comparative en termes de complexité spatiale et temporelle. Cette étude nous a permis en premier lieu de donner des arguments scientifiques concernant les performances de notre solution par rapport aux méthodes de planification de chemins classiques. En second lieu, nous avons pu définir un nombre réduit de paramètres algorithmiques permettant d'ajuster la qualité de la solution par rapport au temps d'exécution disponible à savoir : la précision de discrétisation des paliers 2D, le *Radius Factor* et limitations angulaires. Ces paramètres définissent la densité du graphe de navigation 3D généré. Nous avons constaté qu'un graphe fortement connecté permet d'obtenir des chemins de très bonne qualité en termes de distance de vol, nombre de segments et temps de calcul du chemin. L'ajustement des angles de montée et de descente dans la phase de routage permet d'obtenir des profils de vol de différents types tel qu'un chemin en paliers ou bien un chemin qui suit le relief du terrain. Les temps d'exécution issus de scénarios réels sont très satisfaisants (de l'ordre de quelques secondes) et répondent au besoin de la replanification dynamique en cas d'urgence.

Enfin dans la dernière partie de notre thèse (Chapitre 5), nous avons développé l'outil DTANAV « Demonstration Tool for Aircraft NAVigation » qui met en œuvre le procédé de planification ainsi que les solutions algorithmiques proposés. Nous avons illustré l'usage de DTANAV comme un outil de planification et d'exploration de chemins aériens 3D à travers plusieurs scénarios réels ce qui nous a permis de valider le travail réalisé. À travers une interface

utilisateur conviviale (*User-freindly interface*), le personnel en charge de la préparation de missions a la possibilité de saisir des paramètres liés à la mission (point de départ, point d'arrivée, les caractéristiques de l'appareil, plafond de navigation, etc.) et d'autres qui agissent sur la partie algorithmique afin de maîtriser la qualité et le profil de la solution générée. Nous avons montré aussi le couplage de notre outil avec le modèle de performance d'un hélicoptère donné afin de considérer différentes fonctions objectif à minimiser tel que le temps ou le carburant. À travers un exemple de fonction de performances fournie par Airbus Helicopters, nous avons étudié les meilleurs angles de montée et de descente qui vont nous permettre d'améliorer les performances de la mission en minimisant le temps de vol ou le carburant consommé. Ces paramètres pourront être par la suite utilisés dans la phase d'exploration pour converger vers une solution efficace. En plus, nous avons démontré l'adéquation de notre environnement DTANAV pour répondre à des scénarios de replanification dynamique et faire face à des cas d'urgence comme une panne moteur ou l'évitement d'une zone de danger. Les actions de transfert technologique et l'utilisation de notre outil DTANAV dans différentes plateformes au sein d'Airbus Helicopters représentent une contribution dans le domaine de planification de missions qui ouvre plusieurs voies pour de futurs travaux.

6.2 Perspectives

Le travail effectué dans cette thèse peut être poursuivi suivant de nombreuses directions, nous en présentons ici quelques-unes :

6.2.1 Système de planification multi-critères

Notre outil de planification de missions DTANAV a été validé en considérant principalement le plus court chemin comme fonction objectif (minimiser la distance parcourue). Il pourra être couplé avec différentes fonctions de performance pour réduire par exemple le temps, le carburant consommé, le bruit généré etc. En effet, plusieurs phénomènes liés à l'environnement ou bien aux conditions météorologiques pourront intervenir pour trouver le chemin le plus rapide. À titre d'exemple, en prenant en considération la direction et la force du vent, nous pouvons obtenir un chemin plus long mais plus rapide comme illustré dans la Fig. 6.1. La flexibilité de notre solution nous permet d'obtenir ce type de profils de vol en s'appuyant sur notre graphe 3D. Nous rappelons que les arcs de ce graphe sont annotés selon plusieurs critères (distance, vitesse, vent, etc.) et il suffit de prendre en considération le champs *vent* lors de la phase d'exploration. Nous soulignons qu'il y a des bases de données qui permettent d'obtenir en temps réel des informations sur les conditions météorologiques pour une région donnée et qui pourront être couplées avec notre outil pour la mise à jour du graphe.

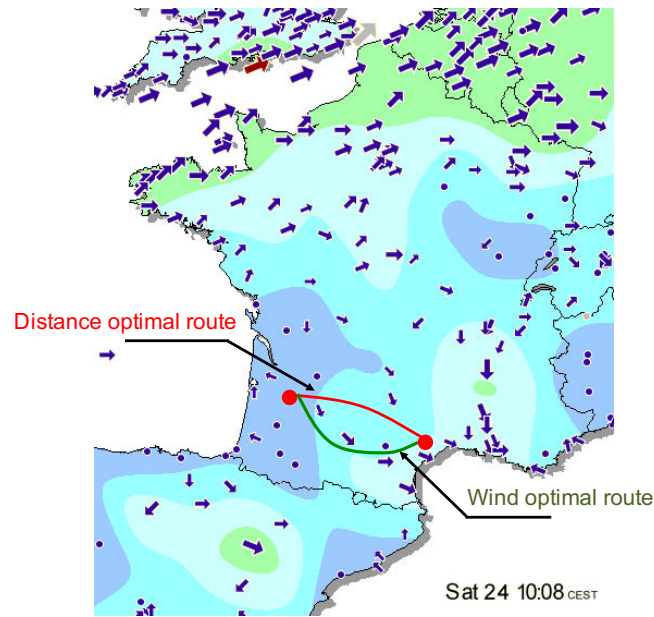


FIGURE 6.1 – Prise en considération du vent lors du procédé de planification

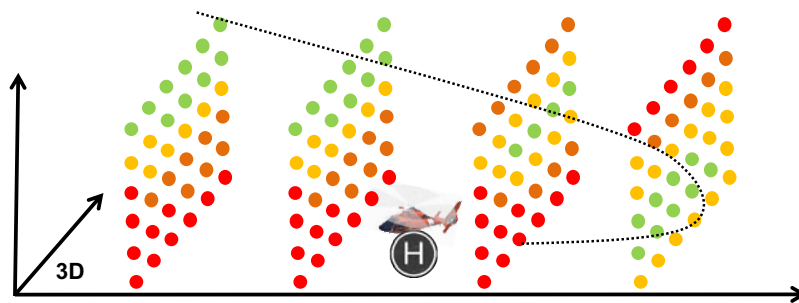


FIGURE 6.2 – Planification de chemins faible bruit

Le programme Européen Clean Sky² promeut un hélicoptère du futur avec des émissions CO₂ réduites et des niveaux de bruit plus faibles. Notre environnement de planification de missions pourra être adapté afin de générer des plans de vol à faible bruit. Pour ce faire, nous avons besoin d'une formulation du problème d'optimisation du bruit tels que les travaux décrits dans [26] qui proposent un modèle de bruit générique (*Rotorcraft Noise Model*) selon les caractéristiques de l'appareil. Ce type de modèle nous permettra de calculer l'intensité du bruit dans notre environnement discrétisé comme illustré dans la Fig. 6.2. Les arcs de notre graphe peuvent être enrichis avec ces informations qui vont guider l'algorithme de routage à trouver un plan de vol à faible bruit [17].

2. Programme européen Clean Sky : <http://www.cleansky.eu/>

6.2.2 Système d'assistance aux pilotes en cas d'urgence

Les situations d'urgence en vol ne bénéficient pas à ce jour d'une représentation et d'un guide des solutions associées. Bien que certains systèmes d'assistance existent et qu'une partie de la connaissance associée aux situations d'urgence ait pu être identifiée, la génération dynamique d'une séquence de manœuvres sous fortes contraintes de temps et dans un environnement non connu à l'avance, représente une nouvelle voie d'exploration. Depuis quelques années, nous avons commencé la conception des systèmes d'assistance aux pilotes en cas d'urgence afin d'améliorer la sécurité en vol. L'architecture de ces systèmes décisionnels se décline en trois niveaux à savoir stratégique, tactique et opérationnel comme illustré dans la Fig. 6.3. Le travail réalisé dans le cadre de cette thèse couvre le niveau tactique. En effet, au niveau stratégique, le système décisionnel s'appuie sur une modélisation du comportement d'un pilote faisant face à un événement imprévu complexe. Le rôle du niveau stratégique est de déterminer s'il est possible de continuer la mission en gardant le plan de vol en cours. Lorsque le pilote est présent dans la boucle de vol, le rôle du système se limite à donner des consignes d'assistance au pilotage. Le niveau stratégique communique au niveau tactique des objectifs sous forme de positions orientées de l'espace à atteindre, des contraintes à satisfaire et une fonction objectif à minimiser.

À ce niveau notre outil DTANAV pourra intervenir au niveau tactique pour le calcul d'un nouveau chemin aérien, en se basant sur l'état courant de l'appareil et le graphe de navigation 3D généré dans la phase de prétraitement. Le rendu de la solution sera par la suite communiqué au niveau supérieur (présence d'une solution, mesure de performances, etc.). Dans le cas d'assistance au pilotage, le pilote décidera de suivre les consignes ou de les ignorer. Après le calcul d'une solution faisable et la validation du pilote, le niveau opérationnel prend en charge la décomposition et l'ordonnancement des manœuvres de pilotage associées à la solution proposée. L'une des perspectives envisagée à court terme est d'implémenter cette architecture décisionnelle en faisant le lien avec les briques des niveaux stratégique et opérationnel qui sont développées respectivement dans les thèses de K. Nikolajevic [44] et H. Ollivier-Legeay (thèse en cours). La modularité du système proposé est un facteur clé pour garantir un système évolutif, facilement programmable et adaptatif selon le type de l'appareil et le cas d'urgence. Notre système décisionnel pourra être implémenté sous forme d'une boucle fermée depuis la détection de l'évènement complexe jusqu'à la génération automatique de la loi de commande ce qui nous permettra d'obtenir des aéronefs avec un niveau d'autonomie élevé voire totalement autonomes.

6.2.3 Conception d'un nouveau service de transport : Urban Air Mobility

La mobilité durable, fluidifiée et sécurisée des personnes est au cœur des politiques de développement de transports urbains. C'est un des enjeux majeurs auxquelles sont confrontées d'une part, les villes dans leur développement face aux problématiques de pollution et de

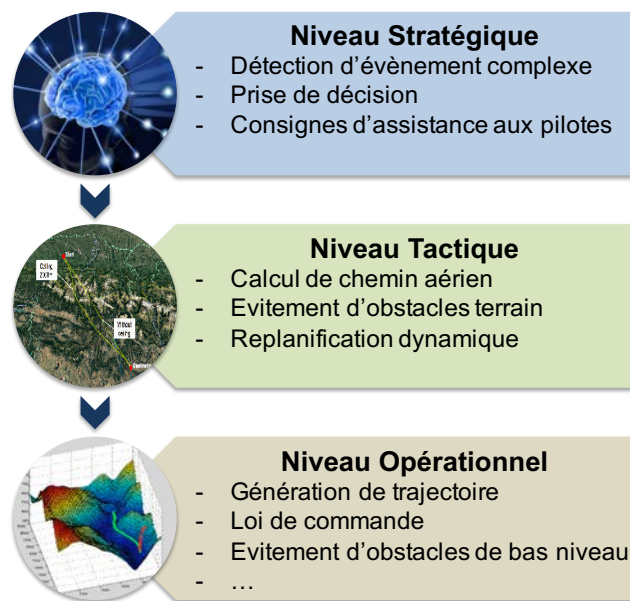


FIGURE 6.3 – Architecture décisionnelle du système d'assistance aux pilotes

congestion de trafic, d'autre part, les Autorités Organisatrice de Transport (AOT) ayant à satisfaire une demande de transport de plus en plus personnalisée.

Avec une grande synergie pilotée par les leaders de l'aéronautique au niveau mondial, une solution émergente consiste à proposer des taxis volants électriques et autonomes (ou bien *Air Taxi*) de petit transport collectif susceptibles de pouvoir se déplacer à la demande dans des environnements divers et variés. Ce nouveau mode de transport concerne des trajets de quelques dizaines de kilomètres pour faire déplacer des usagers entre deux sites spécifiques tels que les grands sites industriels, les aéroports, les centres de congrès, les zones d'activités, ou bien encore les centres hospitaliers. Dans ce contexte, la question de « Urban Air Mobility » est un défi à la fois technologique et organisationnel. En effet, la conception des Taxis volants autonomes à la fois sûrs, durables et efficaces en termes de coût et d'autonomie constitue un premier défi d'ordre technologique. En particulier, ce type d'aéronef autonome requiert d'une part une modélisation du système permettant un *Vertical Take-Off and Landing (VTOL)* afin d'éviter l'utilisation d'une infrastructure particulière, et d'autre part des équipements de type *Sense and Avoid* dédiés à la gestion des obstacles statiques et dynamiques pour garantir la sûreté de fonctionnement du système et la sécurité des passagers. Les premiers vols réalisés dans le cadre du projet VAHANA³ piloté par Airbus Group ont démontré la faisabilité technologique de ce nouveau mode de transport. Un deuxième projet réalisé par la société allemande Lilium⁴ a réussi à concevoir un Taxi volant avec une autonomie de 300 Km

3. Projet VAHANA : <https://vahana.aero/>

4. Projet Lilium : <https://lilium.com/>

et une vitesse de 300 Km/h soit un trajet entre Londres et Paris en une heure.

Cependant, si la maîtrise technologique de l'autonomie est un préalable incontournable, elle ne suffit pas à faire des Taxis volants une solution « miracle ». Le deuxième défi ressort de la capacité de la solution « Urban Air Mobility » d'apporter en termes d'usage une réelle valeur ajoutée aux enjeux de mobilité grâce à sa flexibilité tout en restant économiquement viable. Ce défi est essentiellement d'ordre organisationnel, la plupart des problématiques rencontrées relevant de la logistique. En particulier, la conception d'une solution efficace relève de l'optimisation robuste afin de tenir compte de la nature intrinsèquement dynamique et multi-objectif du système (maximiser la qualité du service offert aux clients, minimiser le coût global de transport, etc.).

Pour répondre à ce défi d'ordre organisationnel, nous pouvons nous appuyer sur les résultats obtenus dans le cadre de cette thèse. En effet, la conception d'un nouveau service de transport de type « Urban Air Mobility » nécessite des outils de planification de missions à court, moyen et long termes. Notre outil DTANAV pourra être utilisé pour calculer des chemins aériens pour des aéronefs de type Taxi volant avec le mode VTOL. Pour ce faire, il suffit de construire des graphes de navigation avec des angles de montée et de descente qui pourront atteindre 90°. Par ailleurs, nous pourrions intégrer des contraintes spécifiques à ce nouveau mode de transport liées aux zones de parking ou de recharge, l'autonomie de l'appareil, etc. Généralement, pour offrir un service efficace et répondre aux besoins d'une large clientèle, il faut utiliser une flotte de taxis autonomes. Pour cette raison, notre travail pourra être étendu pour le déploiement de plusieurs taxis en même temps, donc la planification de chemin pour un aéronef donné doit prendre en considération les autres appareils afin d'éviter des collisions en cours de trajet. Les travaux récents de notre équipe dans ce domaine apportent quelques éléments de réponse concernant l'organisation d'une flotte de drones en mission [62]. Afin d'accroître la sécurité en vol ainsi que la fiabilité du service, notre outil de planification pourra être couplé avec des systèmes de type *Sense and Avoid* afin de détecter rapidement les obstacles statiques et dynamiques et envisager par la suite soit la replanification à court terme ou bien des manœuvres pour éviter les collisions en temps réel. Récemment, nous avons proposé des systèmes de navigation pour des voitures autonomes qui s'appuient sur une fusion de données multi-capteurs qui pourront être utilisés dans ce contexte [7].

Bibliographie

- [1] M. A. Ahmad. The BSP Tree. *Citeseer*, 1980. [21](#)
- [2] C. Alexopoulos and P. M. Griffin. Path planning for a mobile robot. *IEEE Transactions on systems, man, and cybernetics*, 22(2) :318–322, 1992. [19](#)
- [3] F. Andresen, L. Davis, R. Eastman, and S. Kambhampati. Visual algorithms for autonomous navigation. In *Robotics and Automation. Proceedings. 1985 IEEE International Conference on*, volume 2, pages 856–861. IEEE, 1985. [24](#)
- [4] J. L. Bentley. Multidimensional binary search trees used for associative searching. *Communications of the ACM*, 18(9) :509–517, 1975. [20](#)
- [5] R. Bohlin and L. E. Kavraki. Path planning using lazy PRM. In *Robotics and Automation, 2000. Proceedings. ICRA'00. IEEE International Conference on*, volume 1, pages 521–528. IEEE, 2000. [15](#)
- [6] A. Botea, M. Müller, and J. Schaeffer. Near optimal hierarchical path-finding. *Journal of Game Development*, 1 :7–28, 2004. [27](#)
- [7] M. Bouain, R. B. Atitallah, D. Berdjac, and N. Fakhfakh. Multi-Sensor Fusion for Obstacle Detection and Recognition : A Belief-based Approach. In *21st International Conference on Information Fusion*, Jul 2018. [128](#)
- [8] M. Brand, M. Masuda, N. Wehner, and X.-H. Yu. Ant colony optimization algorithm for robot path planning. In *Computer Design and Applications (ICCD), 2010 International Conference on*, volume 3, pages V3–436. IEEE, 2010. [14](#), [15](#)
- [9] J. E. Bresenham. Algorithm for computer control of a digital plotter. *IBM Systems journal*, 4(1) :25–30, 1965. [64](#)
- [10] J. Canny and J. Reif. New lower bound techniques for robot motion planning problems. In *Foundations of Computer Science, 1987., 28th Annual Symposium on*, pages 49–60. IEEE, 1987. [19](#)
- [11] K. P. Carroll, S. R. McClaran, E. L. Nelson, D. M. Barnett, D. K. Friesen, and G. N. William. AUV path planning : an A* approach to path planning with consideration of variable vehicle speeds and multiple, overlapping, time-dependent exclusion zones. In *Autonomous Underwater Vehicle Technology, 1992. AUV'92., Proceedings of the 1992 Symposium on*, pages 79–84. IEEE, 1992. [23](#), [59](#)

- [12] J. Carsten, D. Ferguson, and A. Stentz. 3D Field D : Improved Path Planning and Replanning in Three Dimensions. In *2006 IEEE/RSJ International Conference on Intelligent Robots and Systems*, pages 3381–3386, Oct 2006. [27](#)
- [13] D. Z. Chen, R. J. Szczerba, and J. J. Uhran Jr. Using Framed-Quadrees to find conditional shortest paths in an unknown 2-D environment. 1995. [23](#)
- [14] H. Choset and J. Burdick. Sensor-based exploration : The hierarchical generalized Voronoi graph. *The International Journal of Robotics Research*, 19(2) :96–125, 2000. [19](#)
- [15] K. Daniel, A. Nash, S. Koenig, and A. Felner. Theta* : Any-angle path planning on grids. *Journal of Artificial Intelligence Research*, 39 :533–579, 2010. [17](#), [27](#)
- [16] B. Donald. On motion planning with six degrees of freedom : solving the intersection problems in configuration space. In *Robotics and Automation. Proceedings. 1985 IEEE International Conference on*, volume 2, pages 536–541. IEEE, 1985. [17](#), [19](#)
- [17] R. D. Falck, D. Ingraham, E. Aretskin-Hariton, and J. Berton. Multidisciplinary Optimization of Urban-Air-Mobility Class Aircraft Trajectories with Acoustic Constraints. In *2018 AIAA/IEEE Electric Aircraft Technologies Symposium*, Jul 2018. [125](#)
- [18] D. Ferguson, M. Likhachev, and A. Stentz. A guide to heuristic-based path planning. In *Proceedings of the international workshop on planning under uncertainty for autonomous systems, international conference on automated planning and scheduling (ICAPS)*, pages 9–18, 2005. [20](#)
- [19] E. Frazzoli, M. A. Dahleh, and E. Feron. Real-time motion planning for agile autonomous vehicles. *Journal of Guidance, Control, and Dynamics*, 25(1) :116–129, 2002. [15](#)
- [20] I. Gargantini. An effective way to represent quadrees. *Communications of the ACM*, 25(12) :905–910, 1982. [21](#)
- [21] C. Goerzen, Z. Kong, and B. Mettler. A survey of motion planning algorithms from the perspective of autonomous uav guidance. *Journal of Intelligent and Robotic Systems*, 57(1-4) :65, 2010. [13](#)
- [22] P. E. Hart, N. J. Nilsson, and B. Raphael. A formal basis for the heuristic determination of minimum cost paths. *IEEE transactions on Systems Science and Cybernetics*, 4(2) :100–107, 1968. [14](#), [17](#), [20](#), [25](#), [62](#)
- [23] M. Herman. Fast, three-dimensional, collision-free motion planning. In *Robotics and Automation. Proceedings. 1986 IEEE International Conference on*, volume 3, pages 1056–1063. IEEE, 1986. [24](#)
- [24] C. Hernández, X. Sun, S. Koenig, and P. Meseguer. Tree Adaptive A*. In *The 10th International Conference on Autonomous Agents and Multiagent Systems - Volume 1, AAMAS'11*, pages 123–130, Richland, SC, 2011. International Foundation for Autonomous Agents and Multiagent Systems. [27](#)

-
- [25] A. M. Hussein and A. Elnagar. A fast path planning algorithm for robot navigation with limited visibility. In *Systems, Man and Cybernetics, 2003. IEEE International Conference on*, volume 1, pages 373–377. IEEE, 2003. [16](#)
 - [26] M. J. Lucas and M. A. Marcolini. Rotorcraft noise model. 101, 02 1997. [125](#)
 - [27] C. L. Jackins and S. L. Tanimoto. Octtrees and their use in representing three-dimensional objects. *Computer Graphics and Image Processing*, 14(3) :249–270, 1980. [24](#)
 - [28] D. B. Johnson. A Note on Dijkstra’s Shortest Path Algorithm. *J. ACM*, 20(3) :385–388, July 1973. [14](#)
 - [29] S. Kambhampati and L. Davis. Multi-resolution path planning for mobile robots. *IEEE Journal on Robotics and Automation*, 2(3) :135–145, 1986. [17](#), [23](#), [62](#), [63](#)
 - [30] L. E. Kavraki, M. N. Kolountzakis, and J.-C. Latombe. Analysis of probabilistic road-maps for path planning. *IEEE Transactions on Robotics and Automation*, 14(1) :166–171, 1998. [15](#)
 - [31] L. E. Kavraki, P. Svestka, J.-C. Latombe, and M. H. Overmars. Probabilistic road-maps for path planning in high-dimensional configuration spaces. *IEEE transactions on Robotics and Automation*, 12(4) :566–580, 1996. [15](#)
 - [32] O. Khatib. Real-time obstacle avoidance for manipulators and mobile robots. In *Autonomous robot vehicles*, pages 396–404. Springer, 1986. [14](#), [16](#)
 - [33] S. Koenig and M. Likhachev. D* Lite. *AAAI/IAAI*, 15, 2002. [26](#)
 - [34] S. Koenig and M. Likhachev. Incremental A*. In *Advances in neural information processing systems*, pages 1539–1546, 2002. [26](#)
 - [35] S. M. LaValle. Rapidly-exploring random trees : a new tool for path planning. 1998. [14](#), [15](#), [59](#)
 - [36] S. M. LaValle. From dynamic programming to RRTs : Algorithmic design of feasible trajectories. In *Control Problems in Robotics*, pages 19–37. Springer, 2003. [15](#)
 - [37] S. M. LaValle and J. J. Kuffner Jr. Randomized kinodynamic planning. *The international journal of robotics research*, 20(5) :378–400, 2001. [15](#)
 - [38] G. Li, A. Yamashita, H. Asama, and Y. Tamura. An efficient improved artificial potential field based regression search method for robot path planning. In *Mechatronics and Automation (ICMA), 2012 International Conference on*, pages 1227–1232. IEEE, 2012. [16](#)
 - [39] M. Likhachev, D. Ferguson, G. Gordon, A. Stentz, and S. Thrun. Anytime Dynamic A* : An Anytime, Replanning Algorithm. In *Proceedings of the Fifteenth International Conference on International Conference on Automated Planning and Scheduling*, ICAPS’05, pages 262–271. AAAI Press, 2005. [27](#)

- [40] Y.-H. Liu and S. Arimoto. Path planning using a tangent graph for mobile robots among polygonal and curved obstacles : Communication. *The International Journal of Robotics Research*, 11(4) :376–382, 1992. [19](#), [59](#)
- [41] T. Lozano-Pérez and M. A. Wesley. An algorithm for planning collision-free paths among polyhedral obstacles. *Communications of the ACM*, 22(10) :560–570, 1979. [17](#), [18](#), [19](#)
- [42] G.-C. Luh and W.-W. Liu. An immunological approach to mobile robot reactive navigation. *Applied Soft Computing*, 8(1) :30–45, 2008. [16](#)
- [43] D. Meagher. Geometric modeling using octree encoding. *Computer graphics and image processing*, 19(2) :129–147, 1982. [24](#)
- [44] K. Nikolajevic. *Système décisionnel dynamique et autonome pour le pilotage d'un hélicoptère dans une situation d'urgence*. PhD thesis, Université de Valenciennes, Laboratoire d'Automatique, de Mécanique et d'Informatique industrielles et Humaines, Mar 2016. [126](#)
- [45] S. Nuske, S. Choudhury, S. Jain, A. Chambers, L. Yoder, S. Scherer, L. Chamberlain, H. Cover, and S. Singh. Autonomous exploration and motion planning for an unmanned aerial vehicle navigating rivers. *Journal of Field Robotics*, 32(8) :1141–1162, 2015. [19](#)
- [46] J. Redding, J. Amin, J. Boskovic, Y. Kang, K. Hedrick, J. Howlett, and S. Poll. A real-time obstacle detection and reactive path planning system for autonomous small-scale helicopters. In *AIAA Guidance, Navigation and Control Conference and Exhibit*, page 6413. [15](#)
- [47] E. Rimon and D. E. Koditschek. Exact robot navigation using cost functions : the case of distinct spherical boundaries in $E/\sup n/$. In *Robotics and Automation, 1988. Proceedings., 1988 IEEE International Conference on*, pages 1791–1796. IEEE, 1988. [16](#)
- [48] H. Samet. Neighbor finding techniques for images represented by quadrees. *Computer graphics and image processing*, 18(1) :37–57, 1982. [21](#)
- [49] H. Samet. The quadtree and related hierarchical data structures. *ACM Computing Surveys (CSUR)*, 16(2) :187–260, 1984. [20](#), [21](#), [23](#), [24](#), [62](#), [63](#)
- [50] H. Samet. Using quadrees to represent spatial data. In *Computer Architectures for Spatially Distributed Data*, pages 229–247. Springer, 1985. [21](#), [23](#)
- [51] H. Samet. Hierarchical spatial data structures. In *Symposium on Large Spatial Databases*, pages 191–212. Springer, 1989. [20](#), [21](#), [24](#), [77](#)
- [52] N. Sariiff and N. Buniyamin. An overview of autonomous mobile robot path planning algorithms. In *Research and Development, 2006. SCOReD 2006. 4th Student Conference on*, pages 183–188. IEEE, 2006. [14](#)
- [53] S. Schopferer and F. M. Adolf. Online replanning of time-efficient flight paths for unmanned rotorcraft. *Journal of Intelligent & Robotic Systems*, 84(1-4) :277–296, 2016. [15](#)

-
- [54] C. A. Shaffer and H. Samet. Optimal quadtree construction algorithms. *Computer Vision, Graphics, and Image Processing*, 37(3) :402–419, 1987. [21](#)
 - [55] B. C. Shah and S. K. Gupta. Speeding Up A* Search on Visibility Graphs Defined Over Quadrees to Enable Long Distance Path Planning for Unmanned Surface Vehicles. 2016. [23](#)
 - [56] R. Siegwart and I. R. Nourbakhsh. *Introduction to Autonomous Mobile Robots*. Bradford Company, Scituate, MA, USA, 2004. [19](#)
 - [57] B. Sinopoli, M. Micheli, G. Donato, and T.-J. Koo. Vision based navigation for an unmanned aerial vehicle. In *Robotics and Automation, 2001. Proceedings 2001 ICRA. IEEE International Conference on*, volume 2, pages 1757–1764. IEEE, 2001. [15](#)
 - [58] A. R. Soltani, H. Tawfik, J. Y. Goulermas, and T. Fernando. Path planning in construction sites : performance evaluation of the Dijkstra, A*, and GA search algorithms. *Advanced engineering informatics*, 16(4) :291–303, 2002. [15](#)
 - [59] O. Souissi. *Planification de chemin d’hélicoptères sur une architecture hétérogène CPU FPGA haute performance*. PhD thesis, Université de Valenciennes, Laboratoire d’Automatique, de Mécanique et d’Informatique industrielles et Humaines, January 2015. [2](#), [24](#)
 - [60] A. Stentz. Optimal and efficient path planning for partially-known environments. In *ICRA*, volume 94, pages 3310–3317, 1994. [20](#), [26](#)
 - [61] A. Stentz et al. The focussed D* algorithm for real-time replanning. [26](#)
 - [62] S. A. Sukarno, R. B. Atitallah, and M. Djemai. Approximation Algorithm for 3D Vehicle Routing Problem for a Fleet of Multi-UAVs. In *6th International Conference on Control Engineering & Information Technology*, Oct 2018. [128](#)
 - [63] X. Sun, S. Koenig, and W. Yeoh. Generalized Adaptive A*. In *Proceedings of the 7th International Joint Conference on Autonomous Agents and Multiagent Systems - Volume 1*, AAMAS’08, pages 469–476, Richland, SC, 2008. International Foundation for Autonomous Agents and Multiagent Systems. [27](#)
 - [64] X. Sun, W. Yeoh, and S. Koenig. Generalized Fringe-Retrieving A* : Faster Moving Target Search on State Lattices. In *Proceedings of the 9th International Conference on Autonomous Agents and Multiagent Systems : Volume 1 - Volume 1*, AAMAS’10, pages 1081–1088, Richland, SC, 2010. International Foundation for Autonomous Agents and Multiagent Systems. [27](#)
 - [65] X. Sun, W. Yeoh, and S. Koenig. Moving Target D* Lite. In *Proceedings of the 9th International Conference on Autonomous Agents and Multiagent Systems : Volume 1 - Volume 1*, AAMAS’10, pages 67–74, Richland, SC, 2010. International Foundation for Autonomous Agents and Multiagent Systems. [27](#)

- [66] O. Takahashi and R. J. Schilling. Motion planning in a plane using generalized voronoi diagrams. *IEEE Transactions on robotics and automation*, 5(2) :143–150, 1989. [19](#)
- [67] P. Tsenkov, J. K. Howlett, M. Whalley, G. Schulein, M. Takahashi, M. H. Rhinehart, and B. Mettler. A system for 3D autonomous rotorcraft navigation in urban environments. In *AIAA Guidance, Navigation and Control Conference and Exhibit, Honolulu, HI*, 2008. [17](#), [20](#), [23](#), [59](#), [62](#), [77](#)
- [68] P. Vadakkepat, T. H. Lee, and L. Xin. Application of evolutionary artificial potential field in robot soccer system. In *IFSA World Congress and 20th NAFIPS International Conference, 2001. Joint 9th*, pages 2781–2785. IEEE, 2001. [16](#)
- [69] H. Weghorst, G. Hooper, and D. P. Greenberg. Improved computational methods for ray tracing. *ACM Transactions on Graphics (TOG)*, 3(1) :52–69, 1984. [20](#)
- [70] M. Whalley, M. Takahashi, P. Tsenkov, G. Schulein, and C. Goerzen. Field-testing of a helicopter UAV obstacle field navigation and landing system. In *65th Annual Forum of the American Helicopter Society, Grapevine, TX*, 2009. [20](#)
- [71] M. S. Whalley, M. D. Takahashi, J. W. Fletcher, E. Moralez, L. Ott, L. Olmstead, J. C. Savage, C. L. Goerzen, G. J. Schulein, H. N. Burns, et al. Autonomous black hawk in flight : Obstacle field navigation and landing-site selection on the RASCAL JUH-60A. *Journal of Field Robotics*, 31(4) :591–616, 2014. [20](#)
- [72] A. Yahja, A. Stentz, S. Singh, and B. L. Brumitt. Framed-quadtrees path planning for mobile robots operating in sparse environments. In *Robotics and Automation, 1998. Proceedings. 1998 IEEE International Conference on*, volume 1, pages 650–655. IEEE, 1998. [23](#), [62](#)
- [73] P. Yap. Grid-based path-finding. In *Conference of the Canadian Society for Computational Studies of Intelligence*, pages 44–55. Springer, 2002. [20](#), [62](#)