



HAL
open science

Calcul dynamique de chemin par des drones autonomes en essaim compact dans le cadre de missions en environnements complexes

Ema Falomir Bagdassarian

► **To cite this version:**

Ema Falomir Bagdassarian. Calcul dynamique de chemin par des drones autonomes en essaim compact dans le cadre de missions en environnements complexes. Autre [cs.OH]. Université de Bordeaux, 2019. Français. NNT : 2019BORD0450 . tel-02500580

HAL Id: tel-02500580

<https://theses.hal.science/tel-02500580>

Submitted on 6 Mar 2020

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

THÈSE

PRÉSENTÉE À

L'UNIVERSITÉ DE BORDEAUX

ÉCOLE DOCTORALE DE MATHÉMATIQUES ET
D'INFORMATIQUE

par **Ema FALOMIR-BAGDASSARIAN**

POUR OBTENIR LE GRADE DE

DOCTEUR

SPÉCIALITÉ : INFORMATIQUE

**Calcul dynamique de chemin par des drones
autonomes en essaim compact dans le cadre de
missions en environnements complexes**

Date de soutenance : 17 décembre 2019

Devant la commission d'examen composée de :

Pascal BOUVRY	Professeur,	Rapporteur
Serge CHAUMETTE	Professeur,	Co-directeur de thèse
Eva CRUCK	Docteure,	Examinatrice
Pascal DESBARATS	Professeur,	Président du jury
Johann DRÉO	Docteur,	Examinateur
Gilles GUERRINI	Docteur,	Co-directeur de thèse
Frédéric GUINAND	Professeur,	Rapporteur
Erick LANSARD	Docteur,	Invité
Véronique SERFATY	Docteure,	Examinatrice

À mes grands-parents

Remerciements

Cette thèse CIFRE est le fruit d'une collaboration entre Thales et le LaBRI.

Je tiens tout d'abord à remercier mes directeurs de thèse, Gilles Guerrini et Serge Chaumette, pour leur confiance et leur encadrement.

Mes prochains remerciements vont à Frédéric Guinand et Pascal Bouvry, pour avoir accepté de rapporter cette thèse. Leurs relectures poussées ont permis l'amélioration de ce manuscrit et ont ouvert de nouvelles perspectives.

Je tiens à remercier Rémi Laplace pour sa collaboration ayant permis la réalisation d'une vidéo de simulation¹. Merci également pour ses précieux conseils en compréhension des machines, des humains et de la Science.

Merci à mes collègues et amis avec qui travailler a toujours été un plaisir. Côté LaBRI, merci aux membres de la CVT, passés et présents, et leurs acolytes, Antoine, Antoine, Arnaud, Cindy, Christelle, David, Kendric, Mathilde, Philippe, Rémi, Vincent et Vincent. Côté Thales, merci à Antoine, Christophe, Didier, Fabien, Gilles, Jean-Phi, Olivier, Sabrina, Thibault, et à mes acolytes « doctorants enjoués », Claire et Léo.

Merci à toutes les personnes m'ayant permis de proposer des animations de vulgarisation scientifique, en particulier à Cap Sciences.

D'un point de vue plus personnel, je tiens à remercier mes proches, sans qui cette thèse n'aurait ni commencé ni fini. Merci aux membres ma famille pour leur indéfectible soutien et leur intérêt pour ces travaux ; leur admiration (qu'ils ne savent pas démesurée) donne force et persévérance. Un immense merci à Alexandre, pour, parmi de nombreuses autres raisons, son organisation, ses idées, son enthousiasme, sa patience, les belles figures et le courage qu'il m'a transmis. À Youri, à mes parents, merci.

Enfin, merci à Baptiste, qui, proche ou éloigné, trouve toujours les mots et le temps. Merci à Sandra qui m'aide à garder la tête sur les épaules. Merci à EirSpace et à ses anciens, aux farceurs masqués, à JD, à Bruno (alias le fantôme), aux chanteur·se·s et musiciens.

À celles et ceux qui m'ont accompagnée : merci.

1. <https://youtu.be/KSHonaaDhRA>

Table des matières

Introduction	1
I Contexte et état de l'art	3
1 Applications et enjeux des systèmes sans pilote à bord	5
1.1 Introduction	5
1.2 Terminologie	6
1.3 Enjeux des systèmes sans pilote à bord	13
1.4 Applications des systèmes multiplateformes aériens	14
1.5 Résumé	17
2 Définitions et outils	18
2.1 Graphes	18
2.2 Simulateurs de réseaux	21
3 Caractérisation des contraintes à partir du besoin opérationnel	25
3.1 Introduction	26
3.2 Besoin opérationnel et contraintes induites	26
3.3 Choix du drone	28
3.4 Architectures de contrôle d'un système multiplateforme	29
3.5 Diffusion des informations intra-essaim	31
3.6 Capteurs embarqués pour la détection des obstacles	34
3.7 Modules de communication	35
3.8 Hypothèses opérationnelles	37
3.9 Résumé	38
4 Modèles de mobilité et méthodes de calcul de chemin pour drones et robots	39
4.1 Introduction	40
4.2 Approches locales	42
4.3 Approches globales	46
4.4 Combinaison de plusieurs méthodes	56
4.5 Résumé	58

II	Modèles de mobilité pour un essaim collaboratif de drones autonomes en environnement complexe	60
5	Modèle basé sur les forces virtuelles	62
5.1	Contexte et besoins opérationnels	63
5.2	Forces virtuelles et principe du modèle	65
5.3	Utilisation des forces en fonction du besoin	73
5.4	Simulations et résultats	79
5.5	Résumé	89
6	Calcul de chemin pour un drone autonome en environnement complexe	91
6.1	Contexte	92
6.2	Préambule au calcul de chemin	93
6.3	Méthode de calcul de chemin	98
6.4	Paramètres de discrétisation et de simulation	109
6.5	Résultats et évaluations	112
6.6	Résumé	123
7	Calcul de chemin pour les drones d'un essaim	124
7.1	Introduction	125
7.2	Collaboration interplateforme	125
7.3	Adaptation du modèle pour plusieurs drones	126
7.4	Simulations et résultats	134
7.5	Résumé	137
	Conclusion	138
	Annexes	142

Introduction

Les drones sont devenus des outils incontournables dans de nombreux domaines. Aussi connus sous leur acronyme anglais, UAV (*Autonomous Unmanned Vehicles*), ils peuvent remplir des tâches variées telles que la livraison de sang en urgence, la détection de victimes suite à un cataclysme, l'inspection d'ouvrages d'art ou encore la surveillance de frontières. Au delà des applications de loisirs, les principales utilisations de ces plateformes sans pilote à bord sont désignées par le sigle « 5D » pour *Dull, Dirty, Dangerous, Dear, Difficult* ; soit répétitives, compromettantes, dangereuses, au coût élevé (si réalisées par un autre moyen) et difficiles. Bien que l'utilisation d'un drone permette aux humains de se tenir éloignés du danger et de se relayer tout au long de la mission, le pilotage d'une telle plateforme est une tâche complexe qui requiert en permanence la présence d'un pilote. Afin de diminuer la charge opérateur, certaines tâches de la mission peuvent être automatisées (par exemple le décollage ou le retour à la base), jusqu'à atteindre l'autonomie complète de la plateforme, qui réalisera alors la mission donnée sans intervention humaine.

Une grande variété de types de drones existent et leurs caractéristiques définissent leurs champs d'applications. Le terme « drone » désigne des plateformes de différents types, comprenant par exemple les multicopters (d'environ 50 cm d'envergure, pouvant atteindre 50 km/h et avec une autonomie de l'ordre d'une demi heure) et les drones [Haute Altitude Longue Endurance \(HALE\)](#) (d'une vingtaine de mètres d'envergure, pouvant voler toute une journée à plus de 300 kilomètres par heure). Le choix du drone à utiliser dépend de la mission à remplir. Cette thèse se concentre sur les interventions dans des zones d'intérêt de l'ordre du kilomètre carré, et le déploiement des plateformes doit pouvoir être réalisé rapidement. Enfin, nous favorisons les plateformes autorisant le vol stationnaire, car elles permettent une prise de vue fixe et réduisent les contraintes de mobilité. Nous considérons donc, ici, des drones multicopters.

Pour que les drones soient utiles dans de nombreux domaines, on doit être en mesure de les utiliser dans des contextes variés. En particulier, lorsqu'un drone remplit une mission dans un environnement inconnu, la question de la mobilité est primordiale. Il faut alors permettre au drone de détecter les obstacles qui l'entourent et qu'il réagisse « intelligemment » face à ceux-ci. Plus précisément, il faut élaborer une méthode de calcul de chemin permettant à un drone d'atteindre un point donné tout en évitant les obstacles découverts à la volée.

Compte tenu de la durée de vol restreinte d'un multicopter et de la portée limitée des capteurs embarqués, un drone n'a pas la capacité de remplir des missions de grande envergure (larges zones ou longue permanence). Dès lors, il

est pertinent d'utiliser plusieurs plateformes autonomes de manière conjointe ; on parlera alors d'« essaim de drones ».

La collaboration interdrone peut augmenter significativement les performances du système. Par exemple, la multiplication des plateformes collaboratives permet l'augmentation de la connaissance terrain de chaque drone grâce au partage des informations recueillies par les différentes plateformes. De manière générale, plus un drone dispose d'informations sur son environnement, plus les chemins qu'il calcule pourront être courts. On peut également citer la résilience de l'essaim face à la défaillance d'un nombre restreint de plateformes ; la rapidité accrue de la réalisation de la mission - si les drones se répartissent les différentes tâches - ou encore l'augmentation (quantitative ou qualitative) des données récoltées grâce à la multiplicité des capteurs embarqués par l'ensemble des membres de la flotte. En effet, pour obtenir des informations précises sur une cible, il peut être utile de faire appel à plusieurs capteurs simultanément. Comme la corrélation des données provenant de différents capteurs peut être simplifiée si les angles de vue sont similaires, on souhaite permettre aux plateformes d'évoluer en formation compacte.

L'utilisation d'un essaim permet d'améliorer les résultats obtenus par une plateforme seule, mais fait apparaître de nouvelles problématiques. Une contrainte sous-jacente à l'utilisation d'un essaim de drones est la multiplication des plateformes évoluant dans un espace donné, chacune représentant un obstacle mobile pour les autres. Par ailleurs, la collaboration interplateforme nécessite d'équiper les drones de moyen de communications pair-à-pair, ainsi que de définir les informations qui devront être échangées, leur fréquence d'envoi et leur traitement.

Face à ces problématiques, cette thèse vise à proposer un modèle de mobilité pour drones autonomes collaboratifs, adapté aux environnements inconnus et au vol en formation compacte. La première partie de ce manuscrit introduira le contexte de l'étude, comprenant une présentation des applications et enjeux des systèmes sans pilote à bord (chapitre 1, p. 5), le contexte opérationnel motivant les travaux (3, p. 25), les outils mathématiques et de simulation (chapitre 2, p. 18) et enfin, l'état de l'Art des modèles de mobilité pour drones et robots (4, p. 39). Nos contributions feront l'objet de la deuxième partie de ce document. Nous proposons d'une part, un modèle de mobilité basé sur des forces virtuelles (chapitre 5, p. 62), permettant à un essaim de traverser un environnement complexe, ainsi qu'une méthode de calcul de chemin utilisant les champs de potentiels, adaptée aux environnements urbains, utilisable par un drone seul (chapitre 6, p. 91) ou bien par les drones d'un essaim (chapitre 7, p. 124). Enfin, la conclusion de ce manuscrit résume nos contributions et propose des perspectives à nos travaux (p. 138).

Première partie
Contexte et état de l'art

Chapitre 1

Applications et enjeux des systèmes sans pilote à bord

Sommaire

1.1	Introduction	5
1.2	Terminologie	6
1.2.1	Systèmes sans pilote à bord	6
1.2.2	Systèmes aériens sans pilote à bord	6
1.2.3	Véhicules aériens sans pilote à bord	7
1.2.4	Autonomie	8
1.2.5	Interactions interplateformes	9
1.2.6	MANETs	11
1.2.7	Essaims de véhicules sans pilote à bord	11
1.3	Enjeux des systèmes sans pilote à bord	13
1.4	Applications des systèmes multiplateformes aériens	14
1.4.1	Démonstrations de flottes de drones	14
1.4.2	Essaims autonomes orientés missions	15
1.5	Résumé	17

1.1 Introduction

Ce chapitre a vocation à introduire le domaine des systèmes sans pilote à bord. La première partie de ce chapitre (section 1.2, p. 6) introduit la terminologie associée au domaine des systèmes sans pilote à bord, et se concentre sur les trois caractéristiques principales du contexte de cette étude : les véhicules aériens, l'autonomie des systèmes sans pilote à bord et les interactions interplateformes. Dans un deuxième temps, nous présentons les enjeux des systèmes sans pilote à bord, et en particulier des systèmes aériens multiplateformes (section 1.3, p. 13). Nous montrons notamment que l'utilisation de

plusieurs plateformes collaboratives, ayant chacune des capacités limitées, peut être plus performante que celle d'un seul appareil de grande envergure. Enfin, la troisième partie de ce chapitre (section 1.4, p. 14) présente des applications de systèmes aériens multiplateformes, à différents niveaux d'autonomie.

1.2 Terminologie

La pluralité de domaines d'application des systèmes sans pilote à bord, ainsi que la multiplicité d'engins évoluant dans les airs, sur la terre, sous l'eau ou à sa surface entraîne l'utilisation d'une terminologie vaste et différant en fonction des contextes. À notre connaissance, il n'existe pas de document de référence définissant précisément l'ensemble des termes liés au domaine des « drones ». Cette section vise donc à introduire des définitions aussi précises que possible des termes liés aux systèmes sans pilote à bord, et en particulier des systèmes aériens.

1.2.1 Systèmes sans pilote à bord

Un système sans pilote à bord, *Unmanned System (UMS)*, est défini par le *National Institute of Standards and Technology (NIST)* comme étant un système physique alimenté permettant de remplir des tâches dans le monde physique et dont le principal composant n'embarque pas d'opérateur humain. Le système peut être mobile ou stationnaire et peut inclure des composants complémentaires telle qu'une station de contrôle [NIST, 2008] (la version non-traduite est donnée en annexe 7.5).

On remarque ici la différence importante entre un système et un véhicule ; le véhicule est un des éléments du système global. Le système peut notamment comprendre plusieurs véhicules, potentiellement hétérogènes, ainsi que des éléments associés (liaison de données, station *Command and Control (C2)*, balises disposées au sol, etc.).

On note que la traduction française du terme *Unmanned System* est fréquemment « système autonome », alors que la notion d'autonomie n'est pas mentionnée dans la définition du NIST ; c'est pourquoi « système sans pilote à bord » nous paraît plus approprié. En fonction de l'autonomie du système ainsi que du milieu dans lequel le véhicule évolue, des termes plus précis sont employés (*Unmanned Aerial Vehicle, Remotely Piloted Aircraft Systems*, etc.). La section suivante les introduit.

1.2.2 Systèmes aériens sans pilote à bord

Le terme *Unmanned Aerial System (UAS)*, dont la traduction adoptée par la *Commission Européenne (CE)* est « système d'aéronef sans équipage

à bord », est un cas particulier d'UMS. La Commission européenne définit un UAS comme étant un système composé de « tout aéronef sans équipage à bord et l'équipement servant à le contrôler à distance » [CE, 2019].

L'Organisation de l'Aviation Civile Internationale (OACI) utilise pour terme français « aéronef non habité » et donne une définition d'UAS plus précise que la précédente : « terme désignant l'équipement d'exploitation complet d'un aéronef sans pilote, comprenant l'avion, le poste de commande avec lequel l'aéronef est piloté et la liaison de données sans fil. Le terme inclut les *Remotely Piloted Aircraft Systems* (c.-à-d. systèmes d'avions pilotés à distance, définis dans la section 1.2.3, p. 8) mais n'est pas limitée à ces derniers. [Le terme « aéronef non habité »] regroupe également les aéronefs contrôlés de manière autonome ou semi-autonome tels que les petits drones, les ballons-sondes météorologiques, etc.) » [OACI, 2018]. On peut remarquer que l'expression « aéronef non habité » laisse entendre qu'aucun humain n'est à bord de l'engin, alors que la définition n'exclut pas la possibilité d'embarquer des passagers.

Tant dans les textes de la Commission européenne que dans ceux de l'OACI, l'acronyme anglais UAS reste plus utilisé que sa traduction française. Cet acronyme sera donc utilisé dans ce manuscrit.

1.2.3 Véhicules aériens sans pilote à bord

De nombreuses différenciations peuvent être faites entre UAS, notamment en fonction de leur degré d'autonomie. Néanmoins, des termes différents peuvent également être utilisés pour représenter le même type d'objet. Cette section propose de définir les plus courants d'entre eux, dans le contexte scientifique.

Drones

Depuis quelques années, le mot « drone » est connu du grand public, et fait référence à des appareils aussi variés que les jouets, les futurs moyens de livraison de colis et les outils utilisés par les armées. En français, un « drone » est en général une plateforme aérienne. Pour faire référence à son équivalent terrestre, « robot » sera en général préféré.

Dans les secteurs scientifique, législatif, industriel ou militaire, le mot « drone » est rarement utilisé seul, en particulier car il peut renvoyer à des systèmes très différents les uns des autres.

Dans le cadre de ce manuscrit, le mot « drone » renvoie à un véhicule aérien sans pilote à bord.

Unmanned Aerial Vehicles

D'après le département de la défense des États-Unis (*Department of Defense* (DoD)), le terme *Unmanned Aerial Vehicle* (UAV) réfère à un véhicule

aérien n'embarquant pas d'opérateur humain, pouvant être autonome ou piloté à distance [Gertler, 2012]. Une précision est donnée concernant le caractère réutilisable de l'engin, ce qui exclut de la définition les charges explosives. Un UAV peut néanmoins embarquer des charges utiles larguées en cours de mission.

Remotely Piloted Aircraft Systems

Les *Remotely Piloted Aircraft System (RPAS)*, ou « systèmes d'avion piloté à distance » (traduction de l'OACI) sont des UAS dont l'aéronef est contrôlé depuis un emplacement distant par un pilote à tous les stades du vol [OACI, 2018].

En France, c'est le terme d'« aéronef télépiloté » qui est utilisé dans les textes législatifs. Il est défini comme étant un « aéronef qui circule sans personne à bord sous le contrôle d'un télépilote »¹ [Legifrance, 2015]. Cette définition exclut explicitement l'embarquement d'humains à bord, même passagers.

On peut noter que l'expression « systèmes d'avion piloté à distance » utilisée par l'OACI désigne le système, contrairement au terme utilisé par Legifrance [2015], « aéronef télépiloté », qui porte sur le véhicule. Dans les deux cas, la précision donnée par rapport aux termes UAS ou UAV porte sur le degré d'autonomie de la plateforme.

Rotorcraft UAV

Il est possible de distinguer les UAV en fonction de leur conception ; dans ce cas, deux grandes familles existent. La première est celle des « voilures fixes » (*fixed wings*), qui utilisent le principe de la portance. Les ailes volantes, comme les drones HALE, en sont des exemples. La seconde famille est celle des engins à voilures tournantes (*rotorcrafts* ou *rotary wings*), comme les drones multirotrors et les hélicoptères.

1.2.4 Autonomie

Outre leur milieu d'évolution, les systèmes sans pilote à bord se caractérisent notamment par leur niveau d'autonomie. Là encore, chaque organisme utilise souvent une terminologie qui lui est propre.

1. « Télépilote : personne contrôlant les évolutions d'un aéronef télépiloté, soit manuellement soit, lorsque l'aéronef évolue de manière automatique, en surveillant la trajectoire et en restant en mesure à tout instant d'intervenir sur cette trajectoire pour assurer la sécurité. » [Legifrance, 2015]

La réglementation française distingue trois niveaux d'autonomie [Legifrance, 2015] :

Évolution sous contrôle « manuel » lorsque la trajectoire de l'aéronef télépilote résulte à tout instant de commandes d'un télépilote transmises en temps réel ;

Évolution de manière « automatique » lorsque l'évolution en vol de l'aéronef télépilote a été programmée avant ou pendant le vol et que le vol s'effectue sans intervention d'un télépilote ;

Évolution de manière « autonome » lorsque l'aéronef télépilote évolue de manière automatique et qu'aucun télépilote n'est en mesure d'intervenir sur sa trajectoire.

Le NIST [2008] propose une décomposition plus fine avec les quatre niveaux d'autonomie suivants (la version non-traduite est donnée en annexe 7.5) :

Piloté à distance (*Remote Control*) : un opérateur humain contrôle en continu le système sans pilote à bord, uniquement à partir de ses observations directes. Dans ce contexte, le système sans pilote à bord ne prend aucune initiative et dépend totalement des ordres de l'opérateur.

Téléopéré (*Teleoperation*) : un opérateur humain contrôle en continu le système sans pilote à bord en utilisant les données provenant des capteurs embarqués, soit en contrôlant directement les actionneurs de l'appareil soit en lui assignant des objectifs successifs.

Autonomie partielle (*Semi-Autonomous*) : un système est dit partiellement autonome si le système sans pilote à bord peut réaliser des actions de manière autonome entre des interventions humaines.

Autonomie totale (*Fully Autonomous*) l'UMS remplit la mission qui lui a été attribuée sans intervention humaine, dans un cadre défini, tout en s'adaptant aux conditions opérationnelles et à son environnement.

En résumé, les définitions proposées par ces deux organismes suivent la même logique : l'opérateur humain intervient de moins en moins au cours de la mission lorsque le niveau d'autonomie croît. Néanmoins, on ne peut pas établir d'équivalence entre ces terminologies. En effet, seul le NIST introduit la notion d'« exploitation en vue directe » (*Visual Line Of Sight (VLOS)*), ainsi que la possibilité d'autonomie partielle, dans laquelle le système sans pilote à bord remplit seulement certaines tâches de manière autonome. En revanche, le fait que le système sans pilote à bord puisse être programmé avant ou pendant le vol n'est abordé que par la réglementation française.

1.2.5 Interactions interplateformes

Initialement, l'utilisation d'un engin nécessitait un téléopérateur tout au long de la mission. Bien qu'il soit toujours possible de piloter un véhicule de

son décollage à son atterrissage, les utilisateurs peuvent aujourd’hui compter sur des fonctionnalités de plus en plus perfectionnées offrant une autonomie, au moins partielle, de l’engin.

Dès lors, il devient envisageable de créer un système sans pilote à bord constitué de plusieurs plateformes, au sein duquel les engins interagissent afin d’augmenter les performances du système. Une terminologie précise distingue les différents types d’interactions possibles dans un système multiplateforme, dépendant des paramètres suivants : objectifs partagés, actions individuelles aidant les autres à remplir leurs objectifs et conscience les uns des autres. Ici, on dit qu’un engin est « conscient » de ses coéquipiers s’il prend ses propres décisions en fonction des actions et objectifs des autres.

Action collective

On parle d’action collective lorsque les individus n’ont pas conscience les uns des autres, mais partagent tout de même des objectifs communs, auxquels les actions individuelles contribuent. Dans des actions collectives, les individus ont en général des lois de contrôle local relativement simples, entraînant la réalisation d’un objectif commun lorsqu’un grand nombre d’individus est considéré [Parker, 2008]. Des exemples représentatifs sont les animaux composant des bancs de poissons ou des nuées d’oiseaux.

Coopération

Dans le cas où les entités ont conscience les unes des autres, où elles partagent des objectifs communs et réalisent des actions individuelles bénéficiant aux autres individus, on parle de coopération [Parker, 2008]. Par exemple, l’exploration partagée d’une zone est de tâche coopérative. La coopération n’implique pas le fait que tous les engins aient les mêmes capacités.

Collaboration

La collaboration correspond à la situation où les individus ont des objectifs individuels, ont conscience les uns des autres et où les actions individuelles profitent aux objectifs des autres [Parker, 2008]. Dans ce cas, les individus ont donc des objectifs individuels différents mais compatibles. Par exemple, lorsqu’une équipe pluridisciplinaire travaille sur un projet commun, l’action est menée de manière collaborative.

Coordination

On parle de coordination lorsque les individus ont conscience les uns des autres, n’ont pas d’objectifs communs et où les actions individuelles n’ont pas d’impact positif sur les autres [Parker, 2008]. Par exemple, lorsque plusieurs

véhicules se déplacent dans une même zone (comme une ville), c'est un cas de coordination.

Synthèse

Le tableau 1.1 propose une synthèse des termes définis ci-dessus.

	Conscience des autres	Objectifs communs	Actions individuelles bénéfiques aux autres
Action collective	non	Oui	Oui
Coopération	Oui	Oui	Oui
Collaboration	Oui	non	Oui
Coordination	Oui	non	non

TABLE 1.1 – Synthèse des principaux types d'interactions dans un système multiplateforme.

Les drones peuvent communiquer pour interagir. Les communications via une infrastructure est envisageable mais elles peuvent être rendue difficiles par l'éloignement des drones ou simplement parce que l'infrastructure est inexistante ou endommagée. Un réseau opportuniste, pair-à-pair, permet de s'affranchir des contraintes liées à une infrastructure et également de considérer des drones indépendant de toute entité extérieures à l'essaim.

1.2.6 MANETs

Lorsque plusieurs entités (par exemple des ordinateurs) peuvent communiquer entre eux de manière directe, sans utiliser d'infrastructure extérieure, on parle de « réseau ad hoc ». Dans le contexte d'un réseau, les entités sont généralement appelés « nœuds ». Si les nœuds d'un réseau ad-hoc sont mobiles (par exemple des téléphones portés par des utilisateurs se déplaçant, ou bien des véhicules sans pilote à bord), on parle plus précisément de *Mobile Ad hoc Network* (MANET). Enfin, si les nœuds évoluent dans les airs, on parle alors de *Flying Ad hoc Network* (FANET).

1.2.7 Essaims de véhicules sans pilote à bord

Le mot « essaim » provient du domaine de la biologie où il désigne un groupe d'animaux, tel un essaim d'abeilles, de la même manière que l'on parle de « banc » de poissons ou de « nuée » d'oiseaux. Lorsque de tels groupes d'animaux se déplacent, l'ensemble semble se mouvoir grâce une intelligence unique, alors que chaque individu ne dispose que d'informations limitées. Pour désigner l'action collective réalisée par de tels systèmes, composés d'individus aux

comportements simples, on parle d'« intelligence distribuée », d'« intelligence en essaim » ou encore d'« intelligence collective ».

La robotique s'est inspirée de ces comportements biologiques pour reproduire le même type d'interactions entre des plateformes, et le terme « essaim » est ainsi également utilisé en robotique aujourd'hui. En biologie, les entités composant l'essaim sont en général appelées « individus » ; ce terme est également transposé à la robotique. Dans ce domaine, à notre connaissance, il n'existe pas de définition universellement admise d'« essaim ». Néanmoins, on retrouve des notions communes à la majorité des études sur le sujet ² :

- chaque individu a la capacité de percevoir son environnement et peut communiquer avec ses pairs (avec une portée limitée) ;
- le groupe est autonome et peut par conséquent s'adapter à son environnement ;
- l'essaim a un objectif global et la décomposition des tâches est faite par les individus ;
- le système est extensible (c.-à-d. des individus peuvent être ajoutés sans nécessiter de modification majeure de l'organisation existante) ;

D'autre part, certains aspects varient en fonction des études. Par exemple, certains appellent essaim un ensemble multirobot organisé selon une structure meneur-suiveurs [Brust et Strimbu, 2015] alors que le plus souvent, lorsqu'il est question d'essaim, il est sous-entendu que chaque individu est autonome.

Par ailleurs, dans le cadre de certaines missions, l'essaim considéré est résilient : la perte d'une ou plusieurs plateformes ne remet pas en cause la capacité de l'essaim à atteindre son objectif [Laplace, 2012; Danoy *et al.*, 2015].

De plus, les individus composant un essaim peuvent être en capacité de remplir individuellement des tâches complexes. L'« intelligence » n'est donc pas forcément que collective dans les essaims.

En synthèse, un *essaim* est un ensemble de véhicules sans pilote à bord, partageant un objectif commun. Chaque plateforme constituant l'essaim est autonome et a la capacité d'établir des liens de communication pair-à-pair avec les autres véhicules.

Lorsque toutes les plateformes ont des caractéristiques similaires (vitesse, manoeuvrabilité, performances, etc.), on parle d'essaim « homogène », et sinon, d'essaim « hétérogène ».

Compte tenu de l'autonomie des individus composant l'essaim, ainsi que de leur capacité à coopérer, collaborer ou se coordonner, grâce aux communications qu'ils peuvent établir, l'essaim peut représenter un groupe indivisible autonome d'un point de vue extérieur. Plus précisément, dans un contexte opérationnel, l'essaim peut être considéré comme une entité unique pour un opérateur.

2. Quelques exemples de publications utilisant ces notions : [Saska *et al.*, 2014a; Danoy *et al.*, 2015; Laplace, 2012; Şahin, 2005; Zhou *et al.*, 2017].

1.3 Enjeux des systèmes sans pilote à bord

Les systèmes sans pilote à bord sont utilisés dans de nombreux domaines. Cette section présente les principaux avantages et enjeux de ces systèmes, dans leur ensemble puis plus particulièrement en flotte.

Dans un contexte industriel, une utilisation particulièrement intéressante d'UMS est l'accomplissement de tâches dites *Dull, Dirty and Dangerous* (3D) soit ennuyeuses (ou répétitives), compromettantes et dangereuses. Au delà des 3D, on parle aujourd'hui également de 4D ou 5D, en ajoutant *Dear* (au coût élevé si effectué par un humain) ou *Difficult* (difficile). Parmi les nombreuses utilisations existantes, on peut citer les opérations de *search and rescue* (recherche et sauvetage) [Ghazali *et al.*, 2016], l'inspection d'ouvrages d'art [Yu *et al.*, 2017], la cartographie de zone en 3D [Liu *et al.*, 2017], la réalisation de l'inventaire d'un entrepôt [Barlow *et al.*, 2019] ou encore la surveillance de zones sensibles.

Notre étude porte sur l'utilisation d'essaims de drones, la suite de cette section se focalise donc sur les systèmes aériens multiplateformes.

L'utilisation de plateformes coordonnées augmente encore l'intérêt des systèmes sans pilote à bord. En effet, la multiplication de véhicules peut par exemple accélérer la couverture d'une zone [Yanmaz *et al.*, 2010] ou permettre la résilience du système en cas de défaillance d'un ou plusieurs appareils.

Dans le cadre de missions de trop grande envergure pour être remplies par un seul drone à voilure tournante d'envergure d'un mètre, l'efficacité d'un essaim de multirotors surpasse par plusieurs aspects les performances d'un drone tactique, *Moyenne Altitude Longue Endurance* (MALE) ou *HALE* :

Résilience La pluralité des plateformes constituant un essaim offre la résilience du système dans le cas de la perte de quelques engins. Dans le cas de l'utilisation d'une unique plateforme, la défaillance du véhicule met en péril la mission.

Coût Le coût d'achat et de maintenance de plusieurs multirotors d'envergure de l'ordre d'un mètre est significativement inférieur à celui d'un drone tactique, *MALE* ou *HALE* [Chao *et al.*, 2007].

Discrétion La détection de plusieurs multirotors est plus complexe que celle d'un appareil de grande envergure [To *et al.*, 2009], ce qui peut être crucial dans le cadre applications militaires.

Décollage et atterrissage Les multirotors n'ont besoin d'aucune infrastructure pour décoller et atterrir. Les voilures fixes, en particulier de grande envergure, nécessitent, elles, une grande surface libre pour décoller et atterrir.

Charge de travail pour l'opérateur La répartition des tâches entre les drones d'un essaim se fait au sein même de l'essaim ; du point de vue de l'opérateur, l'essaim est donc considéré comme une entité unique. La gestion

d'une mission n'est donc pas plus complexe que si une plateforme seule était utilisée.

L'inconvénient majeur d'un essaim réside dans la gestion des nombreuses communications interplateformes, nécessaires à la coopération, coordination ou collaboration entre les engins. À ces communications peuvent s'ajouter les éventuelles liaisons avec la station sol. Dans ce cas, il est possible de n'utiliser qu'une partie limitée des plateformes en tant que relais de communication vers la station [Bekmezci *et al.*, 2013].

1.4 Applications des systèmes multiplateformes aériens

Dans ce manuscrit nous nous concentrons sur les applications des systèmes multiplateformes aériens. Les sections suivantes en présentent quelques applications en fonction de leurs niveaux d'autonomie (sections 1.4.1 et 1.4.2).

1.4.1 Démonstrations de flottes de drones

Dans le langage courant, les « essaims de drones » renvoient l'image de chorégraphies réalisées de manière automatique (selon la définition de [Legifrance, 2015]). Depuis quelques années, plusieurs entreprises multiplient leur efforts pour augmenter le nombre d'engins participants à ces démonstrations spectaculaires. En 2016, Intel battait le record mondial du nombre de drones volants simultanément durant l'une des premières représentations, avec 100 drones [Bergounhox, 2016]. Seulement 2 ans plus tard, ce sont plus de 1000 drones qui sont utilisés pour ces spectacles aux allures futuristes, par Intel [Loukil, 2018] mais aussi son principal concurrent Ehang [Aeronews, 2018].

Une technique utilisée pour ces spectacles consiste à fournir à chaque engin une trajectoire prédéfinie, qu'il devra suivre tout au long de la démonstration. L'ensemble des calculs de trajectoire sont donc effectués *a priori*, ce qui permet d'utiliser des ordinateurs qui ne sont pas soumis aux contraintes liées à l'embarqué.

Souvent, des balises sont positionnées au préalable sur la zone de démonstration afin de permettre aux engins de se repérer spatialement. En général les trajectoires précalculées assurent des distances suffisantes entre les drones pour éviter les collisions, même en cas d'imprécision de la mesure de position des plateformes.

D'après la définition d'essaim de robots donnée en section 1.2.7, p. 11, les drones utilisés dans ces démonstrations ne forment pas un essaim.

1.4.2 Essaims autonomes orientés missions

De nombreuses études portent sur la réalisation de missions par des systèmes mult drone. Nous présentons ici les projets les plus proches de notre contexte, c'est à dire l'utilisation d'un essaim autonome et collaboratif pour remplir une mission en environnement inconnu. Le niveau d'autonomie des appareils présentés dans les paragraphes suivants ne correspond à aucune définition de la réglementation française, mais ils peuvent être qualifiés de « partiellement autonomes » ou « totalement autonomes » selon la NIST [2008].

Li et Chen [2017] proposent d'utiliser un essaim de drones coopératifs et totalement autonomes pour rechercher une cible statique dans une zone sans obstacle. Dans la méthode qu'ils proposent, une discrétisation spatiale régulière de l'environnement est effectuée, dont la finesse est adaptée à l'aire pouvant être couverte par un drone en un pas de temps. Chaque engin associe une probabilité de présence de la cible à chaque maille, en fonction de ses propres observations précédentes, mais également en fonction des observations des autres plateformes, partagées à chaque itération. Les engins choisissent leurs déplacements dans le but d'observer les zones ayant la plus grande probabilité de contenir la cible.

Le projet *Cooperative Autonomous Reconfigurable UAV Swarm (CARUS)* a pour but de démontrer la faisabilité d'un essaim de multirobot totalement autonomes en opération [Laplace, 2012; Chaumette et al., 2011]. Le scénario proposé porte sur la surveillance de 15 points d'intérêt par 5 multirobot autonomes. Les points d'intérêt (aussi appelés *Points of Interest (PoI)*) se situent au sol et peuvent être soit en état « normal », soit en « incident ». Dans ce dernier cas, au moins un drone doit se rapprocher du point concerné en restant en vol stationnaire à faible altitude afin de traiter l'incident. Après qu'un drone soit resté à proximité du point pendant une durée prédéfinie, l'incident est considéré comme traité, l'état du PoI repasse à « normal » et le multirobot peut reprendre sa mission de surveillance.

Dans le cadre du projet CARUS, l'environnement dans lequel les drones évoluent ne comporte pas d'obstacle. La solution apportée est flexible à l'ajout et à la suppression de drones durant la mission. Elle est également adaptée à des communications de faible qualité, voire à des pertes de communication. Une problématique prédominante de ce projet réside dans la reconfiguration automatique de l'essaim afin de surveiller l'ensemble des points lorsqu'un drone traite un incident ou après qu'un engin ait quitté l'essaim. Enfin, l'essaim est opérable par un nombre très restreint d'opérateurs [Chaumette et al., 2011] grâce à l'utilisation du modèle de réétiquetage de graphes ADAGRS (*Asynchronous Dynamic Aware Graph Relabeling System*), présenté dans la thèse de Laplace [2012].

Le projet ASIMUT (*Aid to Situation Management based on MULTimodal, MUltiUAVs, MULTilevel acquisition Techniques*) [Bouvry *et al.*, 2016; Rosalie, 2017] vise à réduire la charge de travail d'un opérateur dans le cadre de missions de surveillance par l'utilisation de drones totalement autonomes. Plus précisément, le projet ASIMUT se positionne dans le cadre de scénarios où une mission doit être accomplie sur la base d'informations collectées à partir de différents capteurs embarqués sur un certain nombre de drones [Rosalie, 2017].

L'opérateur demande la réalisation d'une mission à un essaim haut-niveau, constitué de drones de type MALE, qui transmet ensuite des tâches à des essaims bas-niveau composés de multirobot. Certains de ces multirobot collectent les informations sur les cibles concernées (informations sur une zone, sur un véhicule suivi. . .). Enfin, les informations collectées sont transmises via l'essaim de haut-niveau à un centre de renseignements qui évalue la dangerosité éventuelle de la cible. Le modèle de mobilité des multirobot se base à la fois sur un algorithme de type « fourmis » (utilisant le principe des phéromones) et sur un modèle de chaos [Rosalie *et al.*, 2016]. Les phéromones assurent une bonne couverture de la zone d'intérêt, tandis que le modèle chaotique permet qu'un élément extérieur à l'essaim ne puisse pas prévoir les trajectoires des engins.

Boskovic *et al.* [2009] proposent un système partiellement autonome multidrone conçu pour des applications militaires variées telles que le renseignement, la reconnaissance et la surveillance (*Intelligence, Reconnaissance and Surveillance (ISR)*), l'appui aérien rapproché (*Close Air Support (CAS)*) ou la suppression de défenses aériennes adverses (*Suppression of Enemy Aerial Defenses (SEAD)*). Leur solution, CoMPACT (*Collaborative Mission Planning & Autonomous Control Technology*), est un système global comprenant toutes les étapes entre la planification de mission et modèles de mobilité des plateformes. Cette solution est décentralisée, distribuée et robuste aux pertes (ou arrêts volontaires) de communications³. À tout moment des opérateurs peuvent agir sur le système grâce aux données capteurs envoyés par les drones et en fonction des évolutions des objectifs de mission. Dans cette étude, la planification des mouvements des engins est en partie réalisée par des algorithmes évolutionnaires dont les calculs sont réalisés dans une station sol en raison de leur coût calculatoire élevé. D'autres méthodes sont utilisées dans les calculs des déplacements, telles que le suivi de leader, permettant de réduire les besoins calculatoires tout en étant facilement extensibles (c.-à-d. l'augmentation du nombre de plateforme ne remet pas en cause la structure des méthodes existantes).

3. Ce dernier point est particulièrement important dans un contexte militaire où la discrétion est nécessaire à proximité de menaces.

1.5 Résumé

Ce chapitre a introduit la terminologie associée aux systèmes sans pilote à bord (section 1.2, p. 6) et plus précisément les termes liés aux systèmes aériens et aux essais. Nous avons abordé les enjeux des systèmes sans pilote à bord (UMS) et en particuliers de ceux composés de plusieurs véhicules aériens (section 1.3, p. 13). Nous avons montré que les UMS sont particulièrement adaptés à remplir des missions répétitives, compromettantes ou dangereuses et comment la pluralité d'engins permet d'augmenter significativement les performances du système. Enfin, nous avons présenté des exemples d'applications existantes d'UMS composés de plusieurs engins aériens, et ce à différents niveaux d'autonomie (section 1.4, p. 14).

Chapitre 2

Définitions et outils

Sommaire

2.1 Graphes	18
2.1.1 Définitions générales	18
2.1.2 Propriétés recherchées sur les graphes	20
2.2 Simulateurs de réseaux	21
2.2.1 CodeOn	21
2.2.2 JBotSim	21
2.2.3 NECO	22
2.2.4 NEmu	22
2.2.5 ns-3	22
2.2.6 OMNeT++	22
2.2.7 OPNET	22
2.2.8 Choix du simulateur	23

Dans ce chapitre nous donnons tout d’abord des définitions générales sur les graphes. Dans une seconde partie nous présentons les principaux simulateurs de réseaux existant ainsi que les critères de sélection que nous avons utilisés pour en choisir un.

2.1 Graphes

2.1.1 Définitions générales

Un graphe est un ensemble de sommets (ou nœuds) et d’arêtes qui relie certains de ces sommets. On peut modéliser un essaim de drones et leurs liens de communications sous la forme d’un graphe : chaque nœud représente un drone et chaque arête un lien de communication. L’utilité de cet outil mathématique, dans ce cas, est de représenter la présence ou l’absence de lien de communication entre des drones.

Définition 1. Un **graphe non-orienté** $G = (V, E)$ est un couple constitué d'un ensemble de sommets $V(G)$ (nœuds du réseau) et d'un ensemble d'arêtes $E(G)$ reliant deux sommets (liens de communication bidirectionnels entre deux nœuds).

Lorsque deux drones peuvent établir un lien de communication, c.-à-d. lorsque deux nœuds sont reliés par une arête, on dit qu'ils sont « voisins » (ou « voisins directs »). Plus généralement, le voisinage d'un nœud est défini comme suit :

Définition 2. Le **voisinage** d'un sommet u dans un graphe G est l'ensemble des nœuds v de G tels qu'il existe une arête entre u et v .

Une succession de voisins est appelé un chemin. La définition formelle en est la suivante [Laplace, 2012] :

Définition 3. Dans un graphe, un **chemin** est une suite de sommets $P = u_0, u_1, \dots, u_n$ telle que $0 \leq i < n$, $u_i \in V_G$ et $\{u_i, u_{i+1}\} \in E_G$.

Par extension au voisinage, lorsqu'il existe un chemin entre deux nœuds u et v contenant $n - 1$ nœuds intermédiaires, on dit que u et v sont « voisins à n sauts »¹. La figure 2.1 illustre le nombre de sauts entre un nœud et ses voisins.

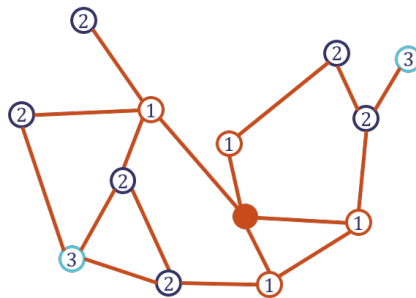


FIGURE 2.1 – Illustration des voisins à 1, 2 et 3 sauts (comme indiqué sur chaque nœud) du nœud représenté par un disque orange. Les arêtes sont représentées par un segment orange.

Les distances entre les drones évoluent au cours du temps puisqu'ils se déplacent au cours de la mission. Ainsi, des liens de communication peuvent être créés ou supprimés. De plus, des drones peuvent apparaître ou disparaître. En assimilant un essaim de drones à un graphe, des nœuds et des arêtes peuvent

1. Deux nœuds voisins directs sont voisins à 1 saut.

être créés ou supprimés. Dans ce cas, on parle de graphes évolutifs, dont la définition est la suivante [Ferreira, 2002] :

Définition 4. Un **graphe évolutif** \mathcal{G} est un triplet $(G, S_G, S_{\mathbb{T}})$ tel que :

- $G = (V, E)$ est le graphe sous-jacent composé de tous les sommets et de toutes les arêtes existant au moins une fois sur le domaine temporel \mathbb{T} (ensemble de toutes les dates possibles) du graphe ;
- $S_G = G_0, G_1, \dots$, avec $G_i = (V_i, E_i)$, est une séquence de sous-graphes statiques de G correspondant chacun à la topologie du graphe sur l'intervalle de temps $[t_i, t_{i+1}[$;
- $S_{\mathbb{T}} = t_0, t_1, \dots$ est une séquence de dates d'événements topologiques tels que t_{i+1} est la date à laquelle l'événement topologique correspondant transforme G_i en G_{i+1} .

Lorsque des graphes non-orientés statiques sont considérés, s'il existe un chemin entre les nœuds u et v , alors il en existe également un entre v et u ; il n'en est pas autant dans les graphes évolutifs. On introduit alors la notion de « trajet », qui est un chemin temporel entre deux sommets d'un graphe évolutif. Par exemple, dans le graphe représenté en figure 2.2, il n'existe jamais de chemin entre c et f , mais il existe un trajet de c vers f . Par contre, il n'existe pas de trajet de f à c .

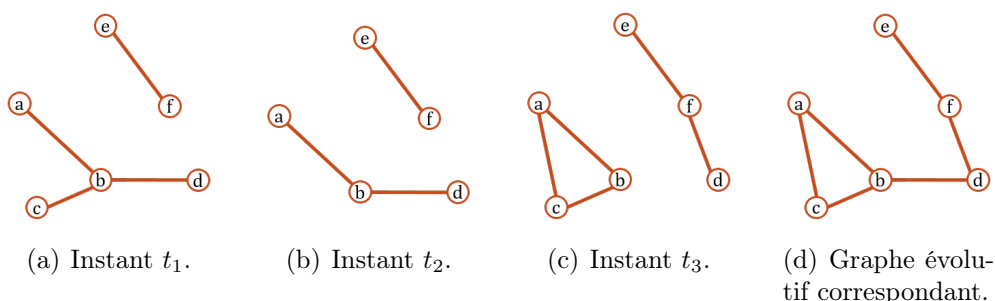


FIGURE 2.2 – Exemple de graphe évolutif.

2.1.2 Propriétés recherchées sur les graphes

Nous chercherons à maintenir des communications au sein de l'essaim considéré même dans le cas de perte de liens de communication ou d'engins, dans le but d'assurer la robustesse du système. Dans le formalisme des graphes, ces propriétés se rapportent à la connexité, dont les définitions sont données ci-dessous [Viennot et Jacquet, 2007] :

Définition 5. Un graphe G est **connexe** si pour toute paire de sommets u et v de G , il existe un chemin les reliant.

Définition 6. Un graphe k -**connexe** est un graphe connexe qui reste connexe après suppression de n'importe quel sous-ensemble de $k-1$ sommets.²

Définition 7. Un graphe est **arête k -connexe** (ou k -arête connexe), avec $k \in |E(G)|, k \neq 0$ s'il est connexe et reste connexe après suppression de n'importe quel sous-ensemble de $k-1$ arêtes.

2.2 Simulateurs de réseaux

L'expérimentation est une étape nécessaire de test, d'évaluation et de validation de nouveaux modèles. Dans le cadre de stratégies de mobilité, cette expérimentation peut être menée par simulation avant d'implémenter les algorithmes sur des plateformes physiques. De nombreux simulateurs de réseaux mobiles communicants sont aujourd'hui disponibles, comme CodeOn, JBotSim, NECO, NEmu, ns-3, OMNeT++ et OPNET. Nous les avons étudié afin d'identifier le plus adapté à la vérification des modèles de mobilité et des algorithmes originaux proposés.

2.2.1 CodeOn

Le simulateur CodeOn a été créé dans le but d'étudier des réseaux ad-hoc de véhicules (VANETs). CodeOn est donc adapté à l'étude de réseaux hautement dynamiques et simule de manière réaliste les communications sans-fils, par exemple en considérant les erreurs de transmission [Li *et al.*, 2011].

2.2.2 JBotSim

JBotSim est une bibliothèque Java permettant la simulation d'algorithmes distribués dans des réseaux dynamiques [Casteigts et Laplace, 2019]. Ce simulateur a été conçu avec l'objectif d'être un outil de prototypage rapide, et il est un simulateur intuitif et facile à utiliser. JBotSim est open source, extensible et propose une interface graphique.

2. Un graphe 2-connexe est aussi dit « bi-connexe ».

2.2.3 NECO

NECO (*NEtwork COding*) est un *framework*³ open source et extensible créé dans le but de permettre l'évaluation de protocoles basés sur du *network coding*. Il inclut notamment la possibilité de définir des graphes représentant la topologie d'un réseau et de décrire de manière modulaire des protocoles utilisant du *network coding*. Une interface graphique est proposée [Ferreira et al., 2009].

2.2.4 NEmu

NEmu (*Network Emulator for mobile universes*) permet d'émuler des réseaux dynamiques virtuels afin de tester et évaluer des prototypes de réseaux ou d'applications distribuées [Autefage et Magoni, 2017]. Cet émulateur permet un contrôle de la topologie du réseau ainsi que des liens de communication.

2.2.5 ns-3

ns-3 (*netwothr simulator*) fait partie de la suite de simulateurs de réseaux à événements discrets ns. ns-3, disponible en ligne [ns 3, 2019], a été conçu pour des applications de recherche et d'enseignement. Ce simulateur gratuit est *open source* et encourage les contributions de la communauté de chercheurs l'utilisant, ainsi que les validations par les pairs.

Un point négatif généralement relevé par les chercheurs concernant ns-3 (tout comme pour son prédécesseur ns-2) est le temps nécessaire à consacrer afin d'apprendre à s'en servir.

2.2.6 OMNeT++

OMNeT++ (*Objective Modular Network Testbed in C++*) est un *framework* extensible, modulaire, basé sur des composants C++, créé pour la simulation de réseaux [OMNeT++, 2019]. OMNeT++ propose une interface de développement basée sur Eclipse ainsi qu'un environnement graphique. Les principaux modèles de simulation de réseaux sont disponibles sur des *frameworks* externes, dont le plus largement utilisé est INET, qui supporte notamment les réseaux filaires et sans fil, ainsi que la mobilité des nœuds.

2.2.7 OPNET

OPNET (*Optimized Network Engineering Tools*) propose plusieurs *frameworks*, dont le plus important dans notre contexte est Rivebed Modeler (anciennement appelé « OPNET Modeler Suite »).

3. Une traduction en français de *framework* est « infrastructure logicielle », mais le terme anglais est le plus courant.

Riverbed Modeler est un simulateur extensible qui fournit des modélisations réalistes de réseaux filaires ou sans fils. Il peut être utilisé dans le but d'évaluer des améliorations de protocoles standards ou, par exemple, de tester des technologies dans des scénarios réalistes [OPNET, 2019].

Riverbed Modeler n'est plus open source depuis son rachat. Dans le cadre de cette thèse, nous souhaitons avoir accès au code du simulateur, dans l'hypothèse où nous aurions du intervenir au niveau des couches basses. Nous ne l'avons donc pas retenu pour la suite de l'étude.

2.2.8 Choix du simulateur

Les deux conditions nécessaires au choix d'un simulateur dans le cadre de cette thèse, sont la gestion de nœuds mobiles et la possibilité d'ajouter ou supprimer des nœuds en cours de simulation (c.-à-d. le simulateur doit pouvoir gérer des réseaux « dynamiques »). Nous nous sommes ensuite intéressés au « niveau » du simulateur. JBotSim a été conçu pour être utilisé au niveau algorithmique [Casteigts et Laplace, 2019]; par exemple, il considère par défaut des transferts de messages instantanés et sans faille. On le qualifie de « haut-niveau ». À l'opposé, OMNeT++ ou ns-3 visent à simuler des protocoles réseaux de manière réaliste, on les qualifie de « bas-niveaux ». Les derniers critères de comparaison ont été la richesse de la documentation, la présence d'une communauté, la variété des exemples proposés ainsi que la fréquence des mises à jour, qui est un bon indicateur de la pérennité du simulateur.

Les résultats de la comparaison que nous avons menée sont résumés dans le tableau 2.1.

	CodeOn	JBotSim	NECO	NEmu	ns-3	OMNeT++
Mobilité	Oui	Oui	Oui	Oui	Oui	Oui
Dynamisme	Oui	Oui	Oui	Oui	Oui	Oui
Niveau	Bas	Haut	Bas	Bas	Bas	Bas
Documentation	Oui	Oui	Pauvre	Oui	Oui	Oui
Communauté	Non	Non	Non	Non	Oui	Oui
Exemples	Non-adaptés	Adaptés	Adaptables	Non-adaptés	Adaptables	Adaptables
Mises à jour	Arrêt en 2014	Régulières	Arrêt en 2012	Arrêt en 2013	Régulières	Régulières

TABLE 2.1 – Comparatif de simulateurs.

De nombreux simulateurs de réseaux permettent de simuler des réseaux mobiles dynamiques, ce qui est au cœur de nos travaux.

Dans le cadre d'une première étude de modèles de mobilité, nous avons choisi JBotSim dont le haut niveau permet une implémentation rapide et intuitive.

Dans un second temps, des résultats plus réalistes sont recherchés, en terme de durée de mission notamment, ce qui nécessite la simulation des délais de

transmission et de traitements des messages, et donc une étude plus bas niveau. Par la richesse de leur documentation, la présence d'une communauté et le support des simulateurs, OMNeT++ et ns-3 sont ainsi ressortis de cette étude comparative. Ces deux simulateurs ont des performances comparables en termes de temps de calcul et de consommation mémoire [Khan *et al.*, 2012]. Bien que tous deux autorisent la gestion de réseaux mobiles, le module associé de OMNeT++ est plus complet que celui de ns-3. Dans le cadre de l'étude de modèles de mobilité, c'est la simulation des déplacements des drones qui est primordiale. Ainsi, c'est OMNeT++ qui a été retenu dans la deuxième partie de cette thèse.

Chapitre 3

Caractérisation des contraintes à partir du besoin opérationnel

Sommaire

3.1	Introduction	26
3.2	Besoin opérationnel et contraintes induites	26
3.3	Choix du drone	28
3.4	Architectures de contrôle d'un système multiplateforme	29
3.4.1	Architecture centralisée	29
3.4.2	Architecture hiérarchique	31
3.4.3	Architecture décentralisée	31
3.5	Diffusion des informations intra-essaim	31
3.5.1	Routage	31
3.5.2	Inondation	32
3.5.3	Synthèse	33
3.6	Capteurs embarqués pour la détection des obstacles	34
3.6.1	SONAR	34
3.6.2	RADAR	34
3.6.3	LIDAR	34
3.6.4	Caméra stéréoscopique	35
3.6.5	Synthèse	35
3.7	Modules de communication	35
3.8	Hypothèses opérationnelles	37
3.9	Résumé	38

Ce chapitre présente le besoin opérationnel que nous considérons dans cette thèse ainsi que les contraintes qui en découlent. Chaque contrainte fait l'objet d'une section spécifique dans laquelle les alternatives et les choix réalisés sont détaillés.

3.1 Introduction

Dans le chapitre 1 (p. 5), nous avons décrit les systèmes sans pilote à bord (UMS) comme des outils performants pouvant réaliser des tâches variées de manière réactive. Lorsque une mission doit être réalisée rapidement ou qu'elle doit nécessairement être menée à terme, l'utilisation conjointe de plusieurs véhicules est particulièrement adaptée.

Les véhicules sans pilote à bord (*Unmanned Vehicle (UV)*) remplissant des missions embarquent des capteurs leur permettant de recueillir des informations sur leur environnement. Ils possèdent également un module de communication afin d'échanger des informations avec une station sol ou avec d'autres engins.

L'utilisation d'un UV soulève de nombreuses problématiques telles que le contrôle de sa trajectoire, la stabilisation de l'appareil, la gestion de l'énergie et des communications. À ces problématiques individuelles s'ajoutent de nombreuses contraintes lorsqu'ils sont utilisés en groupe : communications interplateformes, partage des objectifs et répartition des tâches, évitement « des autres véhicules de la flotte partageant la zone » etc.

3.2 Besoin opérationnel et contraintes induites

Deux missions de surveillance sont considérées dans cette thèse. La première se déroule dans deux zones d'intérêt dépourvues d'obstacles, séparées par un tunnel constituant un environnement inconnu et contraint.

La seconde mission est menée dans un environnement de type urbain inconnu, ne contenant pas d'autres obstacles mobiles que les drones de l'essaim. Les zones d'intérêt ont une surface inférieure à un kilomètre carré. Dans les deux types de mission considérées, les drones auront à se déplacer. De ce fait, un modèle de mobilité permettant à chaque drone d'atteindre un point cible est nécessaire à la réalisation de la mission demandée.

Les missions considérées répondent aux quatre besoins opérationnels suivants :

1. Le système doit pouvoir être mis en place rapidement dans des environnements variés ;
2. L'essaim doit être autonome de tout système extérieur une fois la mission commencée¹ ;
3. La zone d'intérêt n'est pas connue *a priori* ;
4. La mission doit être menée à terme.

1. Néanmoins, si les objectifs de la mission évoluent, alors l'essaim est averti par une entité extérieure.

3. Caractérisation des contraintes à partir du besoin opérationnel

Ces besoins induisent les contraintes suivantes :

1. Dans le but de proposer un système déployable rapidement dans divers environnements, il faut que les plateformes le composant ne nécessitent pas d'infrastructure (voir section 3.3, p. 28).

2. Pour que l'essaim soit autonome, l'ensemble des calculs doit être embarqué. La charge utile étant limitée en poids, énergie et volume, le ou les processeurs embarqués sont donc de relativement faible capacité. Par conséquent, le coût calculatoire du modèle de mobilité doit donc être restreint.

3. Les drones évoluant en zone inconnue, ils doivent embarquer des capteurs leur permettant d'analyser leur environnement, et plus particulièrement de détecter les obstacles (voir section 3.6, p. 34) afin de les éviter. Le modèle de mobilité doit y être adapté. De plus, comme les engins découvrent leur environnement à la volée, les calculs doivent être faits en temps réel.

4. La contrainte de mener la mission à terme induit que l'essaim soit résilient à la perte d'un ou plusieurs drones. Pour remplir cette contrainte, on choisit d'utiliser une architecture décentralisée (voir section 3.4, p. 29) d'une part, et d'autre part, il faut que les informations concernant la mission soient régulièrement échangées. Les engins doivent donc embarquer un module de communication (voir section 3.7, p. 35). Les envois d'informations seront effectués par inondation afin de maximiser le partage d'informations (voir section 3.5, p. 31).

Le tableau 3.1 synthétise les besoins opérationnels, les contraintes systèmes associées et les contraintes de modèle qui en découlent.

Besoin opérationnel	Contrainte système	Contrainte de modèle
Déploiement rapide	Pas d'infrastructure	–
Système autonome	Calculs embarqués	Coût de calcul limité
Environnement inconnu	Capteurs embarqués, calculs en ligne	Calculs à effectuer en temps réel, évitement d'obstacles à intégrer
Mission à mener à terme	Essaim résilient	Calculs distribués, communications par inondation

TABLE 3.1 – Synthèse des besoins opérationnels, contraintes systèmes induites et contraintes de modèles induites également.

3.3 Choix du drone

Le premier choix matériel à faire est celui du type de plateforme. En effet, le choix du drone est dimensionnant pour différents paramètres tels que l'autonomie de la batterie, la charge utile pouvant être embarquée et la rapidité de déploiement.

Le contexte opérationnel considéré ici nécessite l'obtention de données précises provenant d'un environnement inconnu et l'assurance que la mission soit menée à terme. Deux échelles de drones peuvent être adaptées pour répondre à ces besoins : les voilures fixes et tournantes d'une envergure inférieure au mètre, et les drones de grande envergure de type [MALE](#) ou [HALE](#).

Les drones [MALE](#) et [HALE](#), comme leur nom l'indique, peuvent évoluer durant de longues durées (de l'ordre de la journée), et ce, à des vitesses élevées (jusqu'à plus de 300 kilomètres par heure) ; ils peuvent donc couvrir de larges zones. Ce type d'engin permet par exemple de couvrir une frontière entre deux pays, plusieurs fois dans la journée. Les drones de grande envergure ont la capacité d'embarquer des charges utiles lourdes (une centaine de kilogrammes), ce qui leur permet d'utiliser des capteurs nombreux et de haute performance. Enfin, ce type d'aéronef est soumis à de nombreux contrôles et inspections et évolue à une altitude à laquelle les obstacles sont rares, ils sont donc peu faillibles. Néanmoins, le coût élevé de ces plateformes est rédhibitoire pour des utilisations à petite échelle. De plus, elles nécessitent de lourdes infrastructures tels qu'un aérodrome et un centre de contrôle. Ces aéronefs ne sont donc pas adaptés à notre contexte.

On peut noter que des drones à voilure fixe de plus faible envergure (quelques mètres) existent ; leurs performances sont inférieures à celles des drones [MALE](#) et [HALE](#), et ils nécessitent également des infrastructures au décollage. Ils ne sont donc pas adaptés non plus à notre cas d'étude.

À une échelle inférieure, les drones à voilure fixe d'envergure de l'ordre du mètre peuvent dépasser les 50 kilomètres par heure et voler pendant environ 30 minutes. Ils ont la capacité d'embarquer quelques capteurs ainsi qu'un module de communication. Le décollage des ces drones nécessite en général une impulsion initiale, pouvant être fournie par un humain. Cependant, les aéronefs à voilure fixes ont un rayon de virage non négligeable et ne sont donc pas adaptées à une évolution en environnement contraint et inconnu.

Pour ce qui concerne les multirotors, ils peuvent atteindre une vitesse supérieure à 50 kilomètres par heure et ont une autonomie de l'ordre de 20 minutes. Comme les ailes volantes, ils ont la capacité d'embarquer quelques capteurs ainsi qu'un module de communication. Néanmoins l'augmentation du poids de la charge utile impacte leur autonomie. Les drones à voilure tournante peuvent décoller et atterrir verticalement et ne nécessitent donc pas d'infrastructure spécifique. Les multirotors conviennent bien à notre besoin.

Néanmoins, un multirotor ne pouvant transporter qu'une charge utile limi-

tée, les capteurs embarqués sont nécessairement limités également, que ce soit en poids, en encombrement, ou en consommation. Cela entraîne généralement des limitations concernant leur portée et leur précision. Ces plateformes auront donc à s'approcher des objets d'intérêt et, par conséquent, des obstacles qui les entourent. Bien que les multirotors aient la capacité d'effectuer des vols stationnaires et soient soumis à de faibles contraintes cinématiques, la proximité d'obstacles entraîne l'augmentation de la probabilité d'une collision. De plus, les capteurs embarqués ayant une portée limitée, et l'autonomie maximale de ces engins étant de l'ordre de 20 minutes, un multirotor seul ne peut pas couvrir une large zone. Pour ces raisons, il est préférable d'utiliser plusieurs engins collaboratifs lorsque les zones d'intérêt sont larges. L'utilisation d'un système multiplateforme permet également de rendre le système résilient à la perte d'un nombre limité d'engins. C'est pourquoi nous considérons, dans cette thèse, un essaim de drones multirotors, afin de répondre au besoin de missions complexes.

3.4 Architectures de contrôle d'un système multiplateforme

La section 1.2.5 (p. 9) présente les différents types d'interactions possibles entre les engins d'une flotte (action collective, coopération, collaboration ou coordination). De manière complémentaire, il faut aussi aborder la question de la répartition des calculs dans l'essaim. Il existe principalement trois architectures pour les flottes d'objets mobiles : centralisée, hiérarchique et décentralisée [Autefage, 2015] :

Architecture centralisée : elle repose sur une entité unique qui contrôle les engins de la flotte (voir figure 3.1(a)) ;

Architecture hiérarchique : elle utilise plusieurs entités de niveaux différents pour contrôler les engins de la flotte (voir figure 3.1(b)) ;

Architecture décentralisée : elle repose en totalité sur des systèmes autonomes indépendants (voir figure 3.1(c)).

Ces différentes architectures sont décrites dans les sections suivantes. Il est à noter qu'au cours d'une mission donnée, il est envisageable d'adapter le type d'architecture à l'environnement et aux besoins de la mission.

3.4.1 Architecture centralisée

L'utilisation d'une architecture centralisée dont le donneur d'ordre a une connaissance globale de la flotte, permet d'optimiser les actions de chaque élément de l'essaim. Néanmoins, ce type d'architecture nécessite le maintien d'un lien de communication entre l'entité de contrôle et au moins un des véhicules.

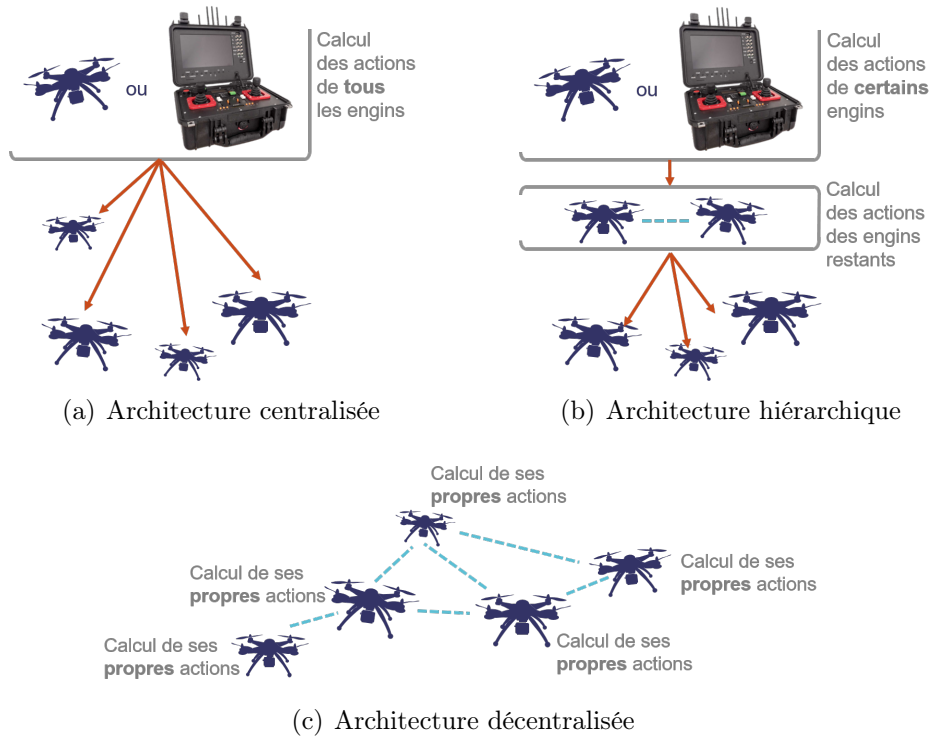


FIGURE 3.1 – Architectures de contrôle. Les flèches orange représentent la transmission d'instructions et les pointillés bleus des échanges d'informations pouvant être utiles au calcul des actions.

La rupture de ce lien entraînerait un arrêt prématuré de la mission. La perte de communication entre le drone relais et les autres plateformes conduirait au même résultat.

D'autre part, l'utilisation d'une architecture centralisée limite la taille maximale de la flotte [Autefage, 2015]. En effet, plus le nombre de véhicules pour lesquels il faut calculer les actions est important, plus la quantité de calculs à réaliser est élevée, tout comme la quantité d'informations à transmettre à l'essaim. L'entité de contrôle doit donc disposer de ressources suffisantes.

Comme dans notre étude, l'essaim doit être indépendant de toute entité extérieure, l'usage d'une architecture centralisée nécessiterait l'utilisation d'une plateforme en tant qu'entité de contrôle. Cette solution paraît difficilement envisageable dans notre contexte, notamment en raison de la portée limitée des modules radio pouvant être embarqués sur des multicopters (voir section 3.7, p. 35) dans le cadre de mission de grande envergure. De plus, le besoin de résilience de l'essaim n'autorise pas le fait que la gestion de mission repose sur une plateforme unique. L'architecture centralisée n'est donc pas adaptée à notre étude.

3.4.2 Architecture hiérarchique

L'architecture hiérarchique utilise plusieurs niveaux de contrôle de la flotte. On peut la considérer comme une dérivée de l'architecture centralisée. Bien que l'architecture hiérarchique soit plus tolérante à l'extension de la flotte, elle peut entraîner des défaillances du système en cas de panne des entités de contrôle, notamment celles de haut niveau. Ce type d'architecture n'est donc pas adapté à notre besoin.

3.4.3 Architecture décentralisée

Au sein d'une architecture décentralisée, chaque entité est autonome. Bien que les engins puissent échanger des informations utiles à leurs propres calculs, chacun dispose des capacités nécessaires à son auto-gestion. Ce type d'architecture permet l'extensibilité de la flotte car elle ne nécessite ni le maintien de liens de communication, ni l'utilisation de quelques entités pour des calculs intensifs [Autefage, 2015]. Enfin, l'architecture décentralisée offre une forte tolérance aux pannes ; c'est donc celle-ci que nous choisissons.

3.5 Diffusion des informations intra-essaim

Compte tenu de la portée limitée des capteurs embarqués et de l'évolution des distances interplateformes, tous les drones ne sont pas nécessairement en mesure d'établir des liaisons pair-à-pair à chaque instant. Afin de permettre les échanges de messages entre drones dont la distance est supérieure à la portée de communication, on utilise des nœuds intermédiaires qui transmettront tour à tour les messages.

Lorsque deux nœuds peuvent échanger un message de manière directe, on dit qu'ils sont « voisins » ou bien « voisins à 1 saut ». Si un émetteur et un destinataire ont à utiliser au moins $n - 1$ nœuds relais pour communiquer, alors ils sont « voisins à n sauts ». Plus généralement, quand des nœuds intermédiaires sont utilisés dans la transmission d'une information, on parle de réseau « multisaut ».

Pour qu'un message soit correctement transmis dans ce type de réseau, il existe deux techniques principales : le routage et l'inondation. Elles font l'objet des sections 3.5.1 (p. 31) et 3.5.2 (p. 32) respectivement.

3.5.1 Routage

Le routage consiste à identifier un chemin (ou « route ») entre une source et une destination, afin d'acheminer un message. L'expéditeur n'envoie son message qu'au premier nœud du chemin, celui-ci ne l'envoiera qu'au suivant, etc.

jusqu'à réception par le destinataire final. La topologie d'un réseau [MANET](#)² évolue au cours du temps puisque les agents qui le composent se déplacent ; des liens de communication sont régulièrement créés et détruits. Une route utilisée à un instant donné entre deux nœuds ne sera donc pas nécessairement réutilisable à l'instant suivant.

Plusieurs classifications des protocoles de routage existent. La plus utilisée est certainement celle distinguant les routages proactifs, réactifs et hybrides [[Naimi, 2015](#)]. Ces trois types de routage sont présentés dans les paragraphes suivants.

Un protocole de routage proactif maintient en permanence une table des routes disponibles entre les nœuds du réseau par le biais de messages de contrôle diffusés périodiquement. Chaque nœud dispose alors d'une table de routage contenant la liste des nœuds relais nécessaires à la transmission d'un message jusqu'à un destinataire donné. Ces tables sont en général échangées entre voisins dans le but de maintenir des routes cohérentes. Cette méthode permet donc de connaître en permanence (à quelques rares exceptions près), une route permettant d'acheminer un message d'une source à un destinataire. Néanmoins, elle nécessite un grand nombre de messages et donc une utilisation non négligeable du réseau, afin de maintenir les tables à jour.

Contrairement à un protocole proactif qui maintient des tables de routage même en l'absence de trafic, un protocole réactif crée une route au moment où un message doit être transmis. Cette approche dispense donc de l'utilisation d'échanges réguliers entre les nœuds. Cependant, la création d'une route nécessite une procédure de découverte du réseau. L'établissement d'une route peut donc nécessiter un délai non-négligeable.

Dans le cas d'un protocole hybride, différents sous-ensembles de nœuds sont définis. Une approche réactive est utilisée au sein de ces sous-ensembles, et les échanges entre ceux-ci sont réalisés via une stratégie proactive. Les performances des protocoles hybrides dépendent fortement des dimensions des sous-ensembles choisis et ils sont bien adaptés aux réseaux de grandes dimensions [[Autefage, 2015](#)].

En conclusion, les protocoles proactifs nécessitent une forte utilisation de la bande passante, ce qui ne paraît pas adapté à notre cas, dans lequel les engins communiquent régulièrement. Dans le cas des protocoles réactifs ou hybrides, la forte mobilité des drones que nous considérons peut entraîner des disparitions régulières de nœuds relais, ce qui provoquerait la perte du message ; ces protocoles ne sont donc pas adaptés à notre besoin.

3.5.2 Inondation

Contrairement au routage dans lequel une route est établie avant de commencer la transmission du message, l'inondation consiste à émettre l'informa-

2. Type de réseau défini en section [1.2.6](#), p. 11.

tion vers tous les nœuds voisins, qui font ensuite de même. Cette technique maximise les chances de réception du message par le destinataire.

La principale faiblesse de l'inondation est qu'elle peut entraîner une saturation du réseau puisque chaque nœud est en quelque sorte utilisé en tant que relais. Néanmoins, des solutions existent, comme la désynchronisation de l'envoi des messages, permettant de limiter les collisions.

3.5.3 Synthèse

Dans les sections précédentes nous avons brièvement introduit le routage et l'inondation à partir de l'exemple le plus simple contenant un destinataire unique. Dans le cadre de notre étude, on souhaite que tous les membres de l'essaim reçoivent l'ensemble des messages, afin que chacun d'entre eux ait une vision la plus complète possible. Pour cela, on peut utiliser une méthode de routage ou d'inondation où chaque drone de l'essaim est destinataire. Néanmoins, tant le routage que l'inondation comportent des points faibles (comme détaillé dans les sections associées).

Nous utilisons une approche alternative : les drones envoient tour à tour la vision la plus globale qu'ils possèdent, à l'ensemble de leurs voisins directs. Dès qu'un drone reçoit un message, il met à jour sa vision de l'environnement en combinant les informations dont il dispose déjà avec celles qu'il a reçues. Lorsqu'un drone envoie sa propre vision de l'environnement, le message contient donc la combinaison des informations qu'il a lui-même découvertes grâce à ses capteurs embarqués, et des informations reçues de la part de ses voisins.

Dans cette approche, un message est envoyé par un drone de l'essaim tous les Δt s. Dans un essaim de n drones, un drone envoie un message toutes les $n \times \Delta t$ s. Donc plus il y a de drones dans l'essaim, plus le délai entre l'émission de la vision d'un drone donné et sa réception par la totalité des nœuds est important. Néanmoins, le trafic est diminué par rapport au cas où tous les drones sont destinataires. De plus, pour que les drones envoient tour à tour les informations dont ils disposent, les plateformes doivent être synchronisées.

Dans notre étude, les messages échangés entre les plateformes contiennent les informations liées à l'environnement déjà découvert. La collaboration interplateforme aide les engins à calculer efficacement un chemin mais un drone seul est capable de se déplacer. De plus, un engin qui n'est pas dans la portée de communication d'une autre plateforme en est potentiellement assez éloigné, et n'a donc pas une utilité immédiate des informations d'environnement que le second possède. Dans notre cas, il est donc plus intéressant d'aborder une approche tour à tour où chaque engin diffuse la fusion de sa vision de l'environnement et des visions qu'il a déjà reçues.

3.6 Capteurs embarqués pour la détection des obstacles

Lorsque des véhicules autonomes évoluent dans un environnement inconnu, l'utilisation de capteurs embarqués permettant de détecter les obstacles est indispensable à l'évitement de collisions. Quatre technologies sont majoritairement utilisés dans ce but sur les multicopters d'envergure inférieure à 50 cm : le sonar, le lidar, le radar et la caméra stéréoscopique. Les paragraphes suivants présentent les principales caractéristiques de ces capteurs dans le contexte de notre étude ainsi que les choix que nous avons fait pour cette étude.

3.6.1 SONAR

Le sonar, abréviation de *sound navigation and ranging*, détecte des objets par la réflexion d'une onde sonore qu'il engendre. Les sonars sont peu coûteux, légers et consomment peu d'énergie ; ils peuvent donc être embarqués sur des multicopters.

Néanmoins, les sonars ne sont pas adaptés à des détections à grande distance. En effet, les ondes sonores émises sont affectées par l'environnement, leur réflexion également. La portée d'un sonar embarqué sur un drone est ainsi de l'ordre de la dizaine de mètres. On peut noter que ce type de technologie est mieux adapté aux applications sous marines car l'environnement y est plus homogène.

3.6.2 RADAR

Le mot radar signifie *radio detection and ranging*. Un radar utilise également le principe de la réflexion d'une onde sur un objet mais avec des ondes radios et non sonores. Les radars embarqués dans des multicopters de moins d'un mètre d'envergure atteignent une portée de plusieurs kilomètres mais avec une faible précision.

3.6.3 LIDAR

Le lidar, de *light detection and ranging* utilise la réflexion d'une onde lumineuse afin de calculer la distance le séparant de l'objet sur lequel le rayon lumineux s'est réfléchi. Les lidars embarqués utilisent un faisceau lumineux concentré sur un endroit précis, permettant une détection de haute résolution, et peuvent avoir une portée de plusieurs dizaines de mètres. Puisque chaque signal lumineux permet la détection dans une direction donnée, un lidar utilisé pour détecter les obstacles autour d'un drone doit être monté sur un servomoteur.

3.6.4 Caméra stéréoscopique

Le principe de la stéréovision est de réaliser une prise de vue simultanée d'une zone donnée depuis plusieurs points de vue. Un traitement automatique est fait à partir des points d'intérêt apparaissant sur les différentes images, permettant l'analyse des objets de la scène. Ces traitements déterminent les distances entre les capteurs de prise de vue et les objets d'intérêt, ainsi que la dimension de ces derniers. La stéréovision permet donc de restituer les trois dimensions spatiales. Les caméras stéréoscopiques sont légères et consomment suffisamment peu d'énergie pour être utilisées en temps réel sur des multirotors [Park et Kim, 2012; Hou *et al.*, 2016].

3.6.5 Synthèse

Pour notre cas d'étude, les sonars ont une portée trop faible dans l'air et les radars n'offrent pas une précision suffisante. Nous ne pouvons donc pas utiliser ces capteurs dans notre contexte.

Un lidar fournit des données précises et a une grande portée, mais ne permet la détection d'obstacles que sur un seul plan. Or, l'assiette d'un multirotor n'est stable qu'à de faibles vitesses. Il n'est donc pas adapté à notre étude.

Le capteur le plus adapté pour la détection d'obstacles dans notre étude est donc la caméra stéréoscopique. C'est d'ailleurs la technologie utilisée par un des drones du marché proposant les meilleures capacités de détection et d'évitement d'obstacles : le DJI Mavic Air [coptrz, 2018; Prindle, 2018; Corrigan, 2018]. Ce drone utilise plusieurs caméras stéréo ayant une portée maximale de 24 m. On suppose donc dans notre étude que les drones embarquent des capteurs équivalents. De plus, on les suppose asservis par des servomoteurs, afin d'obtenir une vision à 360°, comme illustré sur la figure 3.2 (p. 36).

3.7 Modules de communication

Pour pouvoir collaborer, les engins partagent des informations entre eux ; ils doivent donc embarquer un module de communication.

Dans le cadre de nos travaux, les drones constituant l'essaim évoluent dans une même zone d'intérêt, de dimension de l'ordre de 500 m par 500 m. Ils peuvent donc être éloignés de plusieurs de mètres. A contrario, ils peuvent se déplacer en formation compacte, au sein de laquelle la distance interplateforme est de l'ordre du mètre. Compte tenu de ces distances, la flotte forme « réseau local sans fil », (*Wireless Local Area Network* - WLAN).

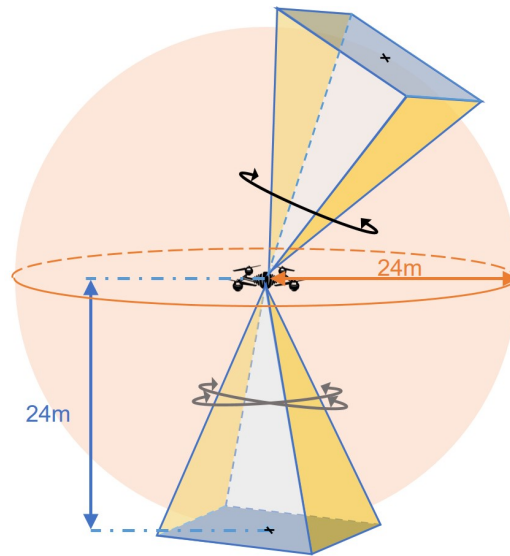


FIGURE 3.2 – Champ de vision des caméras stéréoscopiques montées sur des cardans, permettant au drone une vision à 360°.

Dans le cadre de réseaux de drones ad-hoc à l'échelle des WLAN, deux technologies sont majoritairement utilisées : le XBee [Laplace, 2012; Asadpour *et al.*, 2013; Zhih *et al.*, 2015; Yuan *et al.*, 2017] et le Wi-Fi [Asadpour *et al.*, 2014; Bekmezci *et al.*, 2015; Guillen-Perez *et al.*, 2016; Guillen-Perez et Cano, 2018].

XBee

Un XBee est un module pouvant utiliser plusieurs protocoles sans fils et en particulier le ZigBee. En fonction des versions, un module XBee peut offrir une portée théorique allant de 90 m à plus de 3 km, en environnement libre. Cette technologie au débit théorique faible (au maximum 250 kbps) a pour avantage d'avoir une faible consommation en énergie (entre 40 mA et 300 mA, théoriques, en fonction des versions) [Asadpour *et al.*, 2013; SparkFun, 2019]. La faible consommation du XBee lui permet d'être particulièrement adapté à une utilisation embarquée, mais son faible débit le limite aux usages ne nécessitant qu'une faible quantité de données à échanger.

Wi-Fi

Le Wi-Fi est un protocole de communication pouvant être utilisé selon plusieurs modes de fonctionnement. Dans notre cas, on souhaite que les drones forment un FANET, c'est donc le mode « ad-hoc » qui nous intéresse. À titre d'exemple, la carte Intel Dual Band Wireless-AC 7260, adaptée aux communications par Wi-Fi sur un multirobot, a une consommation de l'ordre de 600 mA,

ce qui est supérieur au Xbee mais reste adapté à une utilisation sur un multitor. Dans la bande 2,4 GHz, sous le standard IEEE 802.11.n, les performances théoriques en liaison montante d'un tel système sont prometteuses : débit pouvant atteindre les 300 Mbps pour une portée de l'ordre de 200 m [Guillen-Perez *et al.*, 2016].

Synthèse

Dans le cadre de notre étude, nous souhaitons que chaque appareil ait, à chaque instant, une vision aussi globale que possible de l'environnement. Ces informations représentent un volume de données non négligeable (cf. section 6.4, p. 109), nous choisissons donc d'utiliser le protocole Wi-Fi via un module Intel Dual Band Wireless-AC. L'agence nationale des fréquences (AFNR) autorise les drones utilisant le Wi-Fi à accéder à la bande fréquence des 2,45 GHz, sous réserve d'une puissance maximale d'émission de 100 mW. Lors des tests menés par Guillen-Perez *et al.* [2016], le débit minimal obtenu utilisant le Wi-Fi en mode ad-hoc est de 1 Mbps ; c'est donc cette valeur que nous utiliserons, ainsi qu'une portée maximale de 100 m.

3.8 Hypothèses opérationnelles

Dans les sections précédentes, nous avons présenté les principaux aspects inhérents aux systèmes de drones en réseau. Puisque cette étude se concentre sur les modèles de mobilité des engins. Dans ce cadre, nous supposons que les systèmes embarqués fonctionnent parfaitement en posant les hypothèses suivantes :

- Chaque drone connaît en temps réel sa localisation avec une précision suffisante.
- Les capteurs de détection d'obstacles embarqués ont une précision suffisante et opèrent tout autour du drone sans zone d'ombre. Les données sont analysées en temps réel par des méthodes existantes qui ne seront pas traitées dans le cadre de la thèse.
- Les capacités de calcul ainsi que la batterie des appareils sont suffisantes pour remplir la mission.

D'autre part, nous posons des hypothèses concernant la zone d'intérêt :

- Au début de la mission, les drones connaissent les limites de la zone d'intérêt, de forme rectangulaire, dans laquelle se déroule la mission (ce qui est représentatif d'un contexte opérationnel réel) ;
- La zone d'intérêt ne contient pas d'obstacles mobiles (en dehors des drones de l'essaim).

3.9 Résumé

Ce chapitre présentait le contexte opérationnel des missions d'intérêt de cette étude. Les missions considérées nécessitent l'utilisation d'un essaim autonome évoluant en environnement inconnu. Il doit mener la mission qui lui a été donnée à terme, même en cas de défaillance de certains drones. Plusieurs contraintes apparaissent donc au niveau du système : l'ensemble des calculs doivent être embarqués et menés en temps réel, à partir de données issues des capteurs. De plus, l'essaim doit être résilient à la perte d'engin. Pour que le système réponde à ces contraintes, plusieurs facteurs doivent être pris en compte : l'architecture de contrôle régissant les interactions au sein de la flotte, la méthode de diffusion des informations entre les engins et les capteurs embarqués, nécessaires à la détection des obstacles et à la création de liens de communication.

Enfin, ce chapitre a introduit les principales hypothèses posées pour cette étude.

Chapitre 4

Modèles de mobilité et méthodes de calcul de chemin pour drones et robots

Sommaire

4.1	Introduction	40
4.1.1	Chemins et trajectoires	40
4.1.2	Caractéristiques du modèle de mobilité recherché	41
4.2	Approches locales	42
4.2.1	Modèles élémentaires	42
4.2.2	Algorithmes utilisant la logique floue	43
4.2.3	Boids	43
4.2.4	Forces virtuelles	44
4.2.5	<i>Dynamic Window Approach</i> (DWA)	45
4.2.6	Synthèse	46
4.3	Approches globales	46
4.3.1	Méthodes reposant sur des graphes	47
4.3.2	Algorithmes évolutionnaires	50
4.3.3	Essaims particuliers	51
4.3.4	Champs de potentiels	51
4.3.5	Synthèse	56
4.4	Combinaison de plusieurs méthodes	56
4.4.1	Association de méthodes locales	57
4.4.2	Association de méthodes globales	57
4.4.3	Combinaison de méthodes globales et locales	58
4.4.4	Synthèse	58
4.5	Résumé	58

4.1 Introduction

Dans toutes les missions considérées dans cette thèse, les drones auront à se déplacer. De ce fait, un modèle de mobilité est nécessaire à la réalisation de toute mission. Ce chapitre a vocation à présenter l'état de l'art des modèles de mobilité adaptés aux déplacements de véhicules sans pilote à bord. Certaines méthodes abordées dans ce chapitre ne sont pas spécifiques aux véhicules aériens, dans ce cas nous employons le terme générique « véhicule ».

Les « modèles de mobilité » regroupent les calculs de chemin et les calculs de trajectoire. La distinction entre ces termes ainsi que les principales applications de ces modèles sont présentées en section 4.1.1 (p. 40). De plus, notre cas d'étude induit des contraintes sur le modèle de mobilité recherché, ce qui oriente donc l'état de l'art, selon les axes présentés en section 4.1.2 (p. 41).

4.1.1 Chemins et trajectoires

L'expression « modèle de mobilité » désigne un ensemble de règles régissant les mouvements d'un véhicule. Plus précisément, lorsque seules les positions successives sont considérées dans le calcul, on parle de calcul de « chemin ». Si on s'intéresse également à la cinétique de la plateforme le long de ce chemin, alors on parle de calcul de « trajectoire ». Les modèles de mobilité peuvent considérer le calcul de chemin ou de trajectoire.

La nécessité d'un calcul de trajectoire et pas seulement de chemin dépend notamment des caractéristiques du véhicule considéré, dont son caractère holonome :

Définition 8. Un système est dit « holonome » s'il peut se déplacer sans contrainte dans toutes les directions de l'espace considéré. En d'autres termes, les variables en x , y et z , attachées au système, sont indépendantes.

Dans le cas de véhicules non-holonomes, un calcul de chemin s'avère insuffisant. Pour illustrer ce propos, prenons l'exemple d'une voiture évoluant dans le plan. Ce type de véhicule ne peut pas avoir de vitesse latérale sans vitesse longitudinale, c'est donc un véhicule non-holonome. Il serait illusoire de donner à ce type de véhicules l'instruction de faire demi-tour sur lui-même. Le modèle de mobilité utilisé pour ce système doit donc nécessairement prendre en compte ses contraintes cinétiques ; un calcul de chemin seul n'est donc pas adapté. Au contraire, le robot sphérique BB-8¹ est un exemple de véhicule holonome dans le plan. En trois dimensions, les drones à voilure fixe sont non-holonomes, à l'opposé des multirotores.

Dans le cas des véhicules holonomes, le calcul de chemin est suffisant. Or notre étude porte sur les multirotores, qui sont holonomes. Nous nous concentrons donc sur les modèles de calcul de chemin.

1. Personnage de la saga Star Wars, apparaissant pour la 1^{ère} fois dans l'épisode VII.

4.1.2 Caractéristiques du modèle de mobilité recherché

Comme détaillé dans la section 3.2 (p. 26) dédiée au besoin opérationnel et aux contraintes induites, le modèle de mobilité recherché doit répondre aux contraintes suivantes :

- Les calculs étant embarqués, le **coût des calculs** associés à la mobilité doit être modéré.
- Les calculs doivent être distribués et chaque plateforme doit être capable de remplir la mission ; on ne peut donc pas se restreindre à l'étude des modèles de groupe mais également analyser des **modèles individuels**.
- Puisque l'environnement est inconnu, le modèle de mobilité doit permettre l'**évitement d'obstacles** et les calculs doivent être réalisés en temps réel.
- Pour que les drones puissent communiquer, à l'aide de modules embarqués à portée limitée, le modèle doit pouvoir prendre en compte les **distances interdrones**.
- Les drones ne doivent pas interrompre la mission tant qu'elle n'est pas remplie, le modèle doit donc garantir l'**absence d'optima locaux**².

L'ensemble de ces contraintes fait apparaître des antinomies. Tout d'abord, il faut utiliser un modèle individuel qui puisse intégrer des contraintes dépendant des voisins. Les modèles individuels, par définition, ne prennent pas en compte d'autres plateformes (en dehors des besoins d'évitement). Il faut donc analyser autant des modèles de groupe que des modèles individuels et tirer profit de chacun d'eux.

Par ailleurs, les véhicules doivent prendre des décisions à partir d'informations locales (car l'environnement est partiellement connu) pour remplir une mission globale. Les approches locales utilisées seules ne peuvent pas garantir l'absence d'optima locaux, mais les approches globales nécessitent une large connaissance de la zone d'intérêt, ce qui n'est pas possible dans notre étude. Là encore, il faut analyser les deux types de modèles et utiliser les avantages de chacun.

Les sections suivantes présentent différents modèles de mobilité répondant à au moins une des contraintes données. La section 4.2 (p. 42) introduit des approches locales, individuelles et de groupe. Les approches globales, essentiellement individuelles, font l'objet de la section 4.3 (p. 46). Enfin, des modèles de mobilité combinant plusieurs approches seront présentées en section 4.4 (p. 56).

2. Dans le contexte du calcul de chemin, un optima local représente un état atteint par le véhicule que les calculs locaux ne permettent pas d'améliorer, le véhicule ne se déplace donc plus

4.2 Approches locales

Un modèle de mobilité basé sur une approche locale vise à calculer, à partir d'une configuration donnée, un déplacement permettant au système de se rapprocher de la configuration finale recherchée, en évitant toute collision [Boissonat *et al.*, 1998]. Ces méthodes sont en général relativement peu coûteuses en termes de calcul et de mémoire, ce qui permet leur mise en œuvre sur des systèmes embarqués. Enfin, puisqu'elles ne nécessitent pas une connaissance complète de l'environnement, les données provenant des capteurs embarqués peuvent suffire au calcul du prochain déplacement.

4.2.1 Modèles élémentaires

En dehors de l'état stationnaire, le mouvement le plus simple est le déplacement rectiligne uniforme. Ce modèle de mobilité est rarement utilisé en robotique car sa simplicité s'oppose à son utilité. La première variation de ce modèle est probablement celle de la direction aléatoire (*random direction*), consistant à suivre, à vitesse constante, une direction choisie aléatoirement selon une distribution uniforme jusqu'à atteindre les limites de la zone considérée, où une nouvelle direction est choisie.

Le modèle de marche aléatoire (*random walk*) en est une extension. Cette méthode consiste à choisir la direction ainsi que la vitesse de déplacement de manière aléatoire et à les suivre pendant une certaine durée ou sur une certaine distance.

Dans la même lignée, le modèle *random waypoint* (littéralement « point cible temporaire aléatoire »)³ suit le même principe, avec, également, des vitesse et direction aléatoires, mais le changement se fait une fois le *waypoint* sélectionné atteint. Ces modèles sont « sans mémoire » puisque les choix successifs n'ont pas de lien les uns avec les autres. On peut noter que des contraintes cinétiques peuvent être ajoutées à ces modèles pour qu'il puissent être mis en œuvre sur des véhicules non-holonomes.

Ces méthodes simples, mais à l'efficacité limitée, peuvent être utilisées dans des missions de couverture de zone, notamment avec plusieurs véhicules. Yamaz et Bettstetter [2010] proposent une variante du modèle « direction aléatoire » : d'une part, le changement de direction se fait soit à la limite de la zone, soit pour éviter une collision avec un voisin et, d'autre part, la nouvelle direction n'est pas choisie selon une loi uniforme. En effet, dans le but de couvrir une zone le plus rapidement possible, il est pertinent de mémoriser les divisions ayant été explorées afin de se diriger vers celles encore inconnues. Dans leur modèle, le choix de la direction dépend donc des déplacements précédents du véhicule, ainsi que des couvertures ayant déjà été effectuées par les autres

3. Le terme anglais *waypoint* peut être traduit par « point cible temporaire », mais l'anglicisme est souvent utilisé à la place de la traduction française.

engins avec lequel il a pu établir un lien de communication et dont il a donc pu prendre connaissance.

Dans cette lignée, il est possible d'intégrer d'autres éléments dans le choix de la prochaine direction, tels que la connectivité entre les véhicules [Xing *et al.*, 2005] ou bien la topographie du terrain [Jardosh *et al.*, 2003].

4.2.2 Algorithmes utilisant la logique floue

En logique classique, dite « bivalente », les propositions prennent soit la valeur « vrai », soit la valeur « faux ». Les propositions en logique floue prennent des valeurs réelles comprises entre 0 et 1. Cette forme de logique est utilisée pour prendre une décision acceptable à partir de plusieurs facteurs. Elle permet de donner de meilleures approximations de la réalité.

Des méthodes de calcul de chemin sont basées sur ce principe, par exemple pour éviter les obstacles. Le calcul du prochain mouvement du véhicule dépend de la distance entre lui-même et l'obstacle, prenant par exemple comme valeur {proche, moyen, éloigné}, ainsi que la direction vers l'obstacle [Du et Liu, 2016; Ibrahim *et al.*, 2014].

4.2.3 Boids

Reynolds [1987] a introduit le modèle des Boids (contraction de *bird* et *oid*, littéralement « qui a la forme d'un oiseau ») permettant de reproduire le comportement d'une nuée d'oiseaux ou d'un banc de poissons. Cette méthode, par nature distribuée, permet de modéliser des comportements émergents à partir d'agents (aussi appelés « particules ») suivant des règles locales et simples. En effet, le mouvement de chaque particule suit 3 règles :

séparation vise à éviter un regroupement dense de particules

alignement vise à se diriger vers la direction moyenne prise par le voisinage

cohésion vise à se diriger vers le barycentre des voisins

Le voisinage d'une particule est l'ensemble des autres particules situées dans son champ de vision (défini par un angle et une distance).

Cette méthode peut être adaptée pour un environnement contenant des obstacles, en ajoutant une règle liée à l'évitement de ceux-ci dans le calcul de mouvement [Saska *et al.*, 2014b].

Dans l'algorithme Boids, les informations nécessaires aux décisions de chaque agent ne dépendent que de données temporellement et spatialement limitées, qui peuvent donc être récupérées par des capteurs embarqués. Le calcul de chaque déplacement nécessite la détection de tous les voisins présents dans une zone sélectionnée, ainsi que de la direction et la vitesse de chacun d'entre eux ; cette analyse n'est pas triviale. Néanmoins, une fois ces détections faites, le choix d'un mouvement se fait par des calculs simples au coût unitaire assez

faible. Enfin, les particules n'ont pas besoin d'établir des liens de communication. Cette méthode est donc adaptée aux expérimentations sur des véhicules autonomes, terrestres [Lam, 2015] ou aériens [Saska *et al.*, 2014b].

4.2.4 Forces virtuelles

Les méthodes basées sur les forces virtuelles sont largement étudiées depuis leur introduction par Khatib [1986]. Cette publication présentait en réalité le principe des champs de potentiels (objet de la section 4.3.4, p. 51), desquels les forces virtuelles dérivent. Néanmoins, la littérature compte aujourd'hui des travaux variés utilisant les champs de potentiels et les forces virtuelles de manière distincte. Les forces virtuelles sont largement utilisées de manière locale spatialement et temporellement (une prochaine position est calculée à chaque itération) alors que le principe des champs de potentiels porte souvent sur des calculs plus globaux. Cette section se concentre sur les utilisations locales des forces virtuelles, tandis que les méthodes basées sur les champs de potentiels font l'objet de la section 4.3.4 (p. 51).

Le principe de la méthode est le suivant : à chaque itération le véhicule fait un bilan des forces virtuelles qui s'appliquent à lui. Le mouvement est déterminé par la résultante de la somme vectorielle de ces forces. Les forces virtuelles les plus utilisées sont nommées « forces ressort ». Elles visent à maintenir une distance donnée entre deux véhicules. S'ils sont trop éloignés, la force est attractive, les véhicules se rapprochent et inversement. La figure 4.1 illustre les mouvements de véhicules soumis à de telles forces.

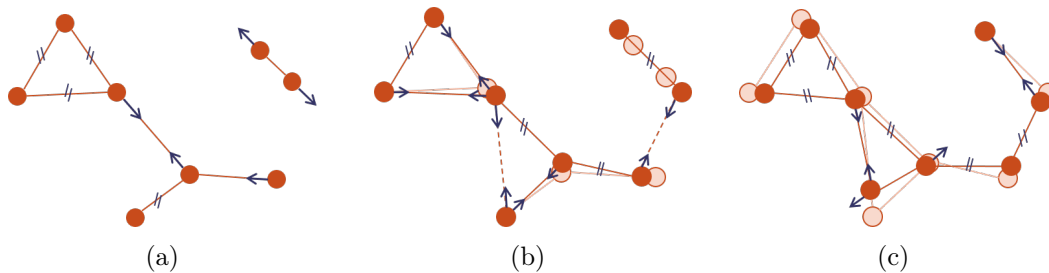


FIGURE 4.1 – Exemple de mouvements de véhicules soumis à des « forces ressort » sur 3 itérations, pour lesquelles la distance d'équilibre est symbolisée par « // ». À l'instant considéré, les véhicules sont représentés par des disques orange et les liens de communication par des segments orange. Les pointillés représentent la création d'un lien de communication. Les disques et lignes plus clairs rappellent l'état du système à l'instant précédent.

Les méthodes basées sur les forces virtuelles sont distribuées par nature, ce qui est un avantage considérable pour des utilisations sur des véhicules autonomes. De plus, elles ne nécessitent pas de grandes ressources de calcul, ce qui permet leur implémentation sur des systèmes embarqués.

Puisque les « forces ressort » permettent de maintenir une distance donnée entre des véhicules, le principe des forces virtuelles est particulièrement adapté aux problèmes de maximisation de couverture tout en maintenant des liens de communication entre les plateformes.

Par exemple, [Messous *et al.* \[2016\]](#) proposent un modèle de mobilité basé sur le principe des forces virtuelles, pour une flotte de drones autonomes remplissant une mission de couverture de zone. Les drones collaborent pour remplir la mission tout en cherchant à minimiser l'énergie consommée (donc leurs déplacements). Les forces entre les drones sont définies de manière à dépendre du nombre de voisins, du niveau de batterie et du nombre de sauts à la station sol avec laquelle la flotte doit rester connectée (cette dernière information n'est pas locale mais construite par des communications de proche en proche au sein de la flotte).

Les forces virtuelles peuvent également être utilisées afin de maximiser la couverture dans des milieux complexes [[Howard *et al.*, 2002](#)]. Dans ce cas, une force de répulsion liée aux obstacles découverts est aussi prise en compte.

En plus de la maximisation de couverture de zone, il peut être intéressant de maximiser le nombre de liens de communication. Le premier de ces objectifs nécessite la maximisation du diamètre de l'essaim, contrairement au second qui tend à minimiser cette dimension. Un compromis est donc à trouver afin de remplir ces deux objectifs antinomiques de manière simultanée. Dans ce contexte les forces virtuelles d'attraction et de répulsion interplateformes sont des outils efficaces, puisqu'elles permettent de faire tendre le système vers une organisation stable, formée de triangles équilatéraux [[Garetto *et al.*, 2007](#)].

Ajoutée aux objectifs maximisations de couverture et de liens de communication objectifs, la biconnexion de l'essaim⁴ peut permettre la tolérance aux pannes de n'importe lequel des véhicules. Une méthode classique pour remplir l'ensemble de ces contraintes, consiste à donner aux véhicules un point de référence commun autour duquel se grouper [[Liu *et al.*, 2010](#)]. Néanmoins, il est possible d'atteindre ces trois objectifs sans utiliser de point de référence, à condition que le réseau soit initialement connecté, en ajoutant des forces angulaires aux forces linéaires [[Casteigts *et al.*, 2010, 2012](#)]. Enfin, la méthode des forces virtuelles permet également d'atteindre ces objectifs dans des environnements contenant des obstacles, au travers desquels les communications sont possibles ou non [[Gobel et Krzesinski, 2008](#)].

4.2.5 *Dynamic Window Approach* (DWA)

La *Dynamic Window Approach* (DWA) (littéralement « approche de la fenêtre dynamique ») a été introduite par [Fox *et al.* \[1997\]](#). Cette méthode vise à calculer les chemins de véhicules dans des environnements inconnus en consi-

4. Définition 6, p. 21.

dérant leurs contraintes cinétiques, ainsi que la perception de portée limitée des capteurs embarqués. Les contraintes cinétiques sont prises en compte en cherchant des chemins dans l'espace des vitesses et accélérations atteignables du véhicule ; celles-ci sont appelées les vitesses et accélérations « admissibles ».

La méthode se compose en deux étapes ; la première consiste à identifier la « fenêtre dynamique », c'est à dire l'espace de recherche relatif aux capacités cinématiques et cinétiques du véhicule depuis la configuration où il se trouve, pour un intervalle de temps donné. Toutes les vitesses admissibles dans la fenêtre sont considérées pour rechercher le prochain mouvement. La seconde étape détermine la solution optimale dans l'espace dans la fenêtre identifiée, en fonction de l'orientation de la plateforme, de sa vitesse et de la distance la séparant du point cible.

Puisque cette méthode considère, par nature, les contraintes cinétiques des véhicules, elle est particulièrement adaptée aux véhicules non-holonomes. Il faut cependant noter que son coût de calcul assez important ne la rend utilisable en temps réel que pour des vitesses limitées. Ainsi, la [DWA](#) est notamment utilisée sur des véhicules terrestres de type voiture, évoluant en intérieur, dans des environnements avec obstacles, à des vitesses de l'ordre du mètre par seconde [[Brock et Khatib, 1999](#); [Saranrittichai et al., 2013](#); [Choi et al., 2012](#)]. Néanmoins, des minima locaux peuvent entraîner l'arrêt prématuré du véhicule. Des améliorations ont donc été proposées pour pallier cet inconvénient, comme la *Global Dynamic Window Approach* (GDWA), méthode basée sur la [DWA](#) dans laquelle les informations sur la connectivité des configurations admissibles sont ajoutées [[Brock et Khatib, 1999](#)].

4.2.6 Synthèse

L'utilisation d'une approche locale est nécessaire dans le cas de calcul de chemin en environnement inconnu. Les méthodes basées sur ce principe permettent de réagir à l'environnement et sont efficaces d'un point de vue calculatoire, ce qui rend possible leur utilisation en embarqué et en temps réel.

Néanmoins, puisque le principe inhérent aux méthodes locales est de calculer une prochaine position du véhicule, elles ne peuvent pas garantir l'accomplissement de la mission à cause de la présence possible d'optima locaux. De plus, les chemins calculés de manière locale peuvent être de mauvaise qualité ; il est possible que le véhicule se déplace bien en direction de la cible mais qu'il aboutisse dans une impasse. Enfin, les méthodes locales sans mémoire peuvent contenir des boucles infinies.

4.3 Approches globales

Une approche globale nécessite la connaissance des points de départ et d'arrivée, mais aussi de l'environnement. Le principe d'un modèle de mobilité

basé sur une approche globale repose sur trois étapes :

1. modéliser l'espace libre de la zone d'intérêt c.-à-d. l'espace dans lequel les véhicules peuvent évoluer ;
2. calculer un chemin dans cette modélisation ;
3. transposer ce chemin (ou calculer la trajectoire associée) dans l'environnement réel.

La modélisation de l'espace peut être faite de manière exacte dans les cas simples, mais les environnements complexes peuvent nécessiter l'utilisation d'une discrétisation spatiale [Boissonat *et al.*, 1998].

Dans le cas de l'utilisation de méthodes globales, les calculs sont le plus souvent effectués avant le début de la mission. Ceci a l'avantage de s'affranchir des problématiques inhérentes à l'embarqué ; d'une part, les calculs n'ont pas à être réalisés en temps réel et d'autre part, ils peuvent être faits sur des machines puissantes, sans considération de leur poids, encombrement et consommation d'énergie. La forte réduction (voire la suppression) des contraintes calculatoires peut donc permettre la recherche d'une solution optimale.

Il est également possible de mettre à jour des chemins ou trajectoire par des méthodes globales, au cours de la mission. Dans ce cas, les calculs sont soit embarqués (ce qui nécessite des puissances de calcul en général supérieures à celles disponibles sur des multirobots), soit effectués par une autre entité que le véhicule (une station sol ou une autre plateforme) mais dans ce cas, le véhicule n'est pas autonome.

4.3.1 Méthodes reposant sur des graphes

Il est possible de représenter une zone d'intérêt par un graphe. Un nœud représente alors un point atteignable et les arêtes sont associées aux trajets autorisés. Une fois un tel graphe construit, on peut utiliser les méthodes classiques de parcours au sein de ceux-ci afin de trouver le plus court chemin entre deux nœuds et en déduire le chemin à suivre par le véhicule. Les paragraphes suivants présentent des algorithmes de calcul de chemin au sein d'un graphe et les résultats associés concernant la navigation de véhicules.

Algorithme de Dijkstra

L'algorithme de Dijkstra [1959] a été l'une des premières méthode de référence permettant de résoudre le problème de calcul de plus court chemin dans un graphe. On peut notamment l'appliquer dans des graphes orientés ou non (voir définition 1, p. 19) et où les arêtes n'ont pas les mêmes longueurs. La longueur du chemin entre la source et l'arrivée correspond alors à la somme des poids (ou longueurs) des arêtes empruntées.

Le principe de l'algorithme est le suivant. À l'initialisation, les distances entre chaque sommet et la source sont considérées comme infinies⁵. Le sommet situé à la plus faible distance de la source est sélectionné et ajouté à un sous-graphe qui sera construit itérativement. À chaque itération, le sommet le plus proche de la source qui n'est pas dans le sous-graphe est sélectionné et ajouté à ce sous-graphe. Les distances entre la source et chaque sommet du sous-graphe sont alors mises à jour. L'algorithme termine lorsque le nœud d'arrivée est sélectionné.

Cet algorithme est particulièrement adapté, par exemple, à la recherche de chemin en ville en fonction du trafic routier s'il est localement amélioré en fonction du besoin. Par exemple, lorsque le chemin est calculé pour un individu utilisant les transports en commun, il est pertinent de minimiser le nombre de changements de mode de transport, donc d'ajouter un coût au changement de nœud [Bozyigit *et al.*, 2017]. Les applications de cet algorithme de calcul de chemin ne sont pas limitées à la circulation routière; il peut également être utilisé, par exemple, pour calculer le chemin de véhicules évoluant dans des environnements avec obstacles [Fusic *et al.*, 2018].

Cet algorithme a plusieurs faiblesses, notamment la possibilité d'entrer dans une boucle infinie et devient peu performant lorsque plusieurs chemins au coût le plus faible sont simultanément identifiés dans le sous graphe contenant les distances à la source [Wang et Zhao, 2011].

Algorithme A* et dérivés

L'algorithme A* (prononcé « A étoile ») a été introduit par Hart *et al.* [1968] dans le but de trouver rapidement un chemin entre deux nœuds d'un graphe, quitte à ne pas fournir le résultat optimal dans un premier temps. L'algorithme A* est basé sur le même principe que l'algorithme de Dijkstra, auquel est ajoutée une fonction heuristique permettant d'orienter la recherche en direction de la destination finale. Ainsi, le premier chemin trouvé est souvent de bonne qualité, mais l'optimalité n'est pas garantie. Ceci est pertinent dans le cas d'environnements fortement dynamiques, pour lesquels la rapidité de calcul est privilégiée à l'optimalité de la solution.

De très nombreuses variantes de cet algorithme ont été proposées, comme D* (pour *Dynamic A**), introduit par Stentz [1994] dans le but de calculer des chemins dans des environnements inconnus. Cet algorithme fonctionne de la même manière qu'A*, mais les coûts des arêtes peuvent évoluer en fonction des découvertes de l'environnement. D* se rapproche donc d'une méthode locale. L'algorithme Θ^* est une autre extension d'A* [Daniel *et al.*, 2010], conçue pour calculer des plus courts chemins dans un maillage, sans se restreindre à la forme de la grille, ce qui permet de raccourcir les trajets.

La méthode *Exploration Priority Based Heuristic Approach* est également une variante de A* et plus précisément de D* [Al Redwan Newaz *et al.*, 2013].

5. à l'exception de la distance entre la source et elle-même, qui est nulle

Cet algorithme propose une exploration dans un maillage, avec une limitation à quatre directions à chaque itération et il est utilisable sans système de localisation global en environnement inconnu.

L'algorithme A* et ses variantes, parfois avec des ajustements dépendant des plateformes considérées, sont aussi adaptables au calcul des chemins de plusieurs véhicules évoluant dans une même zone [Wang *et al.*, 2015].

Méthodes probabilistes : RRT et variations

Le principe des méthodes probabilistes consiste à représenter l'environnement libre (c.-à-d. autorisé, sans obstacle) par un graphe, aussi appelé dans ce cas « réseau ». Le premier nœud construit est associé à la configuration (position et orientation) initiale du véhicule et les nœuds suivants sont construits à partir de configurations choisies aléatoirement.

L'objectif de la méthode RRT (*Rapidly exploring Random Trees*), introduite par LaValle [1998], est de trouver un chemin sans collision et adapté aux contraintes cinétiques du véhicule, entre les configurations initiale et finale. Pour cela, à chaque itération, une nouvelle configuration est sélectionnée aléatoirement dans l'espace de recherche. Si un chemin existe entre celle-ci et une des configurations déjà sélectionnées, elle est connectée au nœud le plus proche d'elle. Le graphe ainsi construit est un arbre et toutes les configurations sélectionnées sont accessibles depuis le point de départ. L'algorithme termine dès que la distance entre une configuration et la configuration finale est inférieure à une valeur prédéfinie.

Puisque cette méthode est par nature probabiliste, il est fréquent que le chemin solution ne soit pas (du tout) optimal car irrégulier ou long ; un lissage (suppression de certains points, moyenne, ou autre méthode) est donc appliqué.

Il est possible d'utiliser la méthode RRT et ses variations pour plusieurs véhicules évoluant dans une même zone, par exemple dans des missions coopératives de couverture de zone, à condition que l'extension des arbres de chaque véhicule se fasse de manière synchronisée [Račinský, 2016]. La méthode peut également permettre de remplir des missions telles que le suivi de cible mobile [Li *et al.*, 2017].

Une faiblesse de la l'algorithme RRT est qu'il est peu efficace dans des environnements contenant des passages étroits (puisque la sélection des configurations est faite aléatoirement, si le seul passage possible est étroit, il est peu probable de le sélectionner rapidement). Une amélioration, nommée *RRT-path* consiste à appliquer l'algorithme RRT sur un chemin précalculé [Vonásek *et al.*, 2009]. Dans notre cas, les véhicules évoluent en environnement inconnu, il ne peut donc pas y avoir de chemin précalculé.

4.3.2 Algorithmes évolutionnaires

Les algorithmes évolutionnaires (ou évolutionnistes), comme leur nom l'indique, s'inspirent de la théorie de l'évolution. Ils sont utilisés dans le cadre de problèmes d'optimisation lorsqu'il n'existe pas de méthode de résolution exacte. Chaque « individu » de la « population » représente alors une solution. Une itération de la méthode peut être assimilée à une nouvelle génération d'êtres vivants : certains individus perdurent quand d'autres disparaissent et quelques couples se reproduisent. Des mutations peuvent également apparaître.

Les étapes principales des algorithmes évolutionnaires sont décrites par le pseudo-code ci-dessous [Srinivas et Patnaik, 1994] :

Algorithme 1 Principales étapes des algorithmes évolutionnaires.

Initialisation de la population ;
 Évaluation de la population ;
tant que critère d'arrêt non-atteint **faire**
 Sélection des solutions pour la prochaine population ;
 Reproduction par croisements et/ou mutations ;
 Évaluation de la nouvelle population
fin tant que

Un croisement correspond à la création d'une solution à partir de la combinaisons de plusieurs parties d'autres solutions de la population. Une mutation représente une modification d'une partie d'une solution existante, ce qui aboutit à une nouvelle solution.

Différents critères d'arrêt peuvent être envisagés. On peut rechercher le ou les extrema d'une fonction objectif appelée *fitness function*, fixer le nombre de générations ou encore arrêter l'algorithme lorsque la population n'évolue plus suffisamment rapidement [Durand, 2004].

Parmi les nombreuses applications de ces algorithmes, nous n'abordons ici que les études portant sur le calcul de chemin.

Les algorithmes évolutionnaires sont notamment adaptés au calcul de chemin de drones de grande envergure, que ce soit à une altitude donnée [Cakir, 2015] ou bien en trois dimensions [Ozalp et Sahingoz, 2013]. Dans ces cas, l'environnement considéré contient des zones interdites et la topologie du terrain est considérée. Des chemins pour drones peuvent être produits hors ligne, mais aussi en temps réel [Ioannis K. *et al.*, 2003].

Les véhicules terrestres de taille modeste peuvent également être considérés. Les travaux de Floreano et Mondada [1996] montrent qu'un robot Khepera peut évoluer dans un environnement contraint grâce à un algorithme génétique, en n'utilisant que les informations provenant de ses capteurs embarqués. Le véhicule peut atteindre un point cible tout en prenant en compte son niveau de batterie dans le calcul du chemin. Dans leurs travaux, les calculs sont réalisés sur une machine extérieure au système.

4.3.3 Essaims particuliers

La méthode d'optimisation par essaims particuliers, ou *Particle Swarm Optimization (PSO)*, s'inspire du modèle Boids (présenté en section 4.2.3, p. 43) et en utilise la terminologie. Cette méthode, proposée par Kennedy et Eberhart [1995] s'inspire de la biologie, tout comme les algorithmes évolutionnaires. Contrairement au modèle Boids, le terme « essaim » ne fait pas référence ici à un groupe d'UMS, mais désigne l'ensemble des « particules » représentant chacune une solution au problème d'optimisation à résoudre. Cette méthode d'optimisation au coût calculatoire faible est utilisée dans des domaines très variés tels que le design d'antennes, la finance, les réseaux de communication ou bien les calculs de chemin [Mario *et al.*, 2014].

Les règles suivies par les particules dans la méthode PSO sont proches de celles régissant les mouvements des agents du modèle Boids. Néanmoins une notion de « mémoire », individuelle et collective, apparaît dans la méthode PSO. D'une part, chaque particule se souvient de la meilleure situation qu'elle a elle-même atteinte (notion nommée « nostalgie » par les auteurs). D'autre part, toutes les particules connaissent la meilleure solution qui ait été atteinte par un des individus de l'essaim. Ces deux informations sont prises en compte dans le calcul de la position suivante.

La méthode PSO ne nécessite que peu de communications entre les véhicules (partage des meilleures solutions déjà trouvées). Les autres données nécessaires lorsque l'on considère un calcul de chemin sont les localisations des obstacles à proximité, lesquelles peuvent être fournies par des capteurs embarqués.

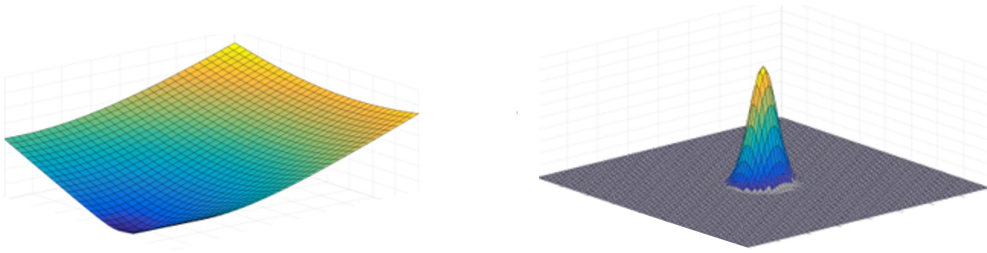
On peut utiliser la méthode PSO de manière directe avec des véhicules, c'est-à-dire que la position des particules représente la position des véhicules [Mario *et al.*, 2014]. Dans ce cas on souhaite en général que les véhicules atteignent un point donné, c'est donc un problème avec un optima unique pour lequel le PSO donne des résultats concluants [Di Mario *et al.*, 2013; Mario *et al.*, 2014].

La méthode PSO peut également être utilisée pour calculer le chemin d'un essaim et non plus de véhicules individuels. Dans ce cas, chaque particule représente l'ensemble des positions des véhicules à un instant donné [Saska *et al.*, 2014a].

4.3.4 Champs de potentiels

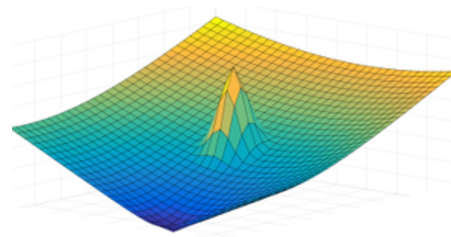
La méthode des champs de potentiels artificiels, (en anglais *Artificial Potential Field (APF)*) a été introduite par Khatib [1986]. Le principe de cette technique très intuitive est d'assimiler le véhicule à une bille soumise à la gravité et se déplaçant sur un tissu ondulé ; la bille suit en permanence les pentes les plus prononcées. La bille représente donc le véhicule et le tissu (le champ

de potentiels) est une représentation de l'environnement. Ce champ de potentiel artificiel (noté par la suite APF) est composé d'un champ attractif vers le point cible et de champs répulsifs liés à chacun des obstacles (voir figure 4.2, p. 52). Les déplacements du véhicule, calculés itérativement par une descente de gradient, lui permettent donc d'atteindre son point cible tout en évitant les obstacles, comme l'illustre la figure 4.3 (p. 53).



(a) Champ attractif, lié à un point cible.

(b) Champ répulsif, lié à un obstacle.



(c) Somme des deux champs de potentiels, permettant d'atteindre le point cible en évitant l'obstacle.

FIGURE 4.2 – Exemple de champ de potentiels artificiels (APFs). Les valeurs de potentiel sont illustrées par un dégradé allant du jaune (valeurs les plus élevées) au bleu (valeurs les plus faibles).

Khatib [1986] propose la définition de champ attractif suivante :

$$U_{x_d}(x) = \frac{1}{2}k_p(x - x_d)^2, \quad (4.1)$$

où x représente la position du véhicule ; x_d représente la position du point cible et k_p est un coefficient d'attraction. Le champ répulsif lié à un obstacle est défini par :

$$U_o(x) = \begin{cases} \frac{1}{2}\eta \left(\frac{1}{\rho} - \frac{1}{\rho_0} \right) & \text{si } \rho \leq \rho_0 \\ 0 & \text{si } \rho > \rho_0 \end{cases}, \quad (4.2)$$

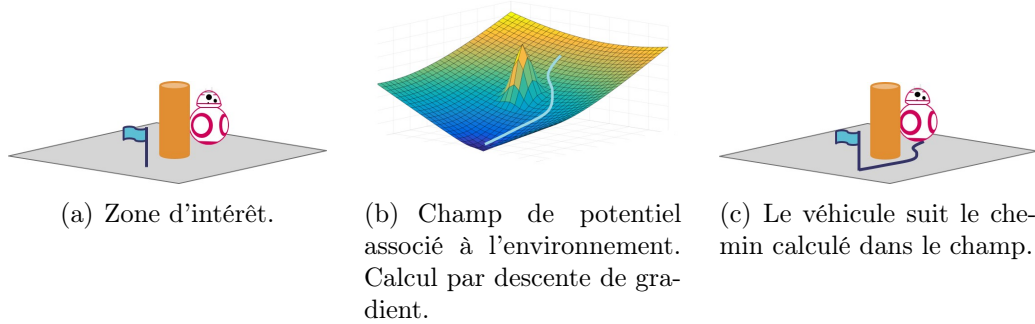


FIGURE 4.3 – Principe de la méthode de calcul de chemin par APFs. Ici, un obstacle (cylindre orange) est présent entre le véhicule (un robot terrestre holonome) et son point cible (symbolisé par un drapeau).

où η est un coefficient de répulsion ; ρ_0 la distance maximale d'influence de l'obstacle ; ρ la plus petite distance entre le véhicule et l'obstacle considéré.

Depuis sa création, cette méthode a été très largement étudiée en robotique, certainement du fait de sa simplicité de mise en œuvre et son élégance, mais aussi peut-être à cause de ses inhérentes faiblesses (décrites dans les paragraphes suivants).

La principale fragilité des méthodes basées sur les champs de potentiels est certainement la présence possible de minima locaux, pouvant entraîner l'arrêt du véhicule avant sa destination (voir figure 4.4) [Koren et Borenstein, 1991].

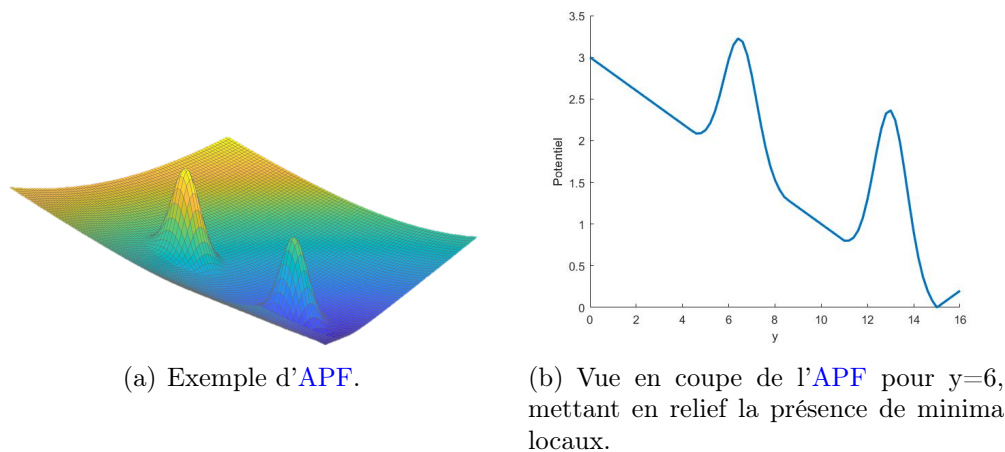


FIGURE 4.4 – Exemple d'APF avec minima locaux.

Le second point faible majeur de cette technique réside dans l'impossibilité pour un véhicule d'atteindre son point cible dans le cas où un obstacle se trouve à proximité de celui-ci ; ce problème est parfois nommé *Goal Non Reachable*

with Obstacle Nearby (GNRON). En effet, si le champ de potentiel répulsif lié à l'obstacle englobe le point cible, alors celui-ci n'est pas nécessairement le point de plus faible potentiel de la carte (voir figure 4.5, p. 54) [Qian Jia et Xingsong Wang, 2009].

Enfin, une autre faiblesse de la technique est le risque qu'un véhicule ne puisse pas passer entre deux obstacles proches. En effet, si les champs répulsifs liés à chacun des obstacles se recouvrent, le véhicule ne peut pas emprunter cette voie (voir figure 4.6, p. 54) [Koren et Borenstein, 1991].

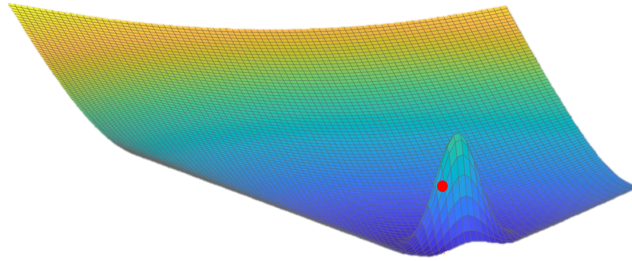


FIGURE 4.5 – Exemple d'APF ne permettant pas d'atteindre le point cible, représenté sur le champ par un disque rouge.

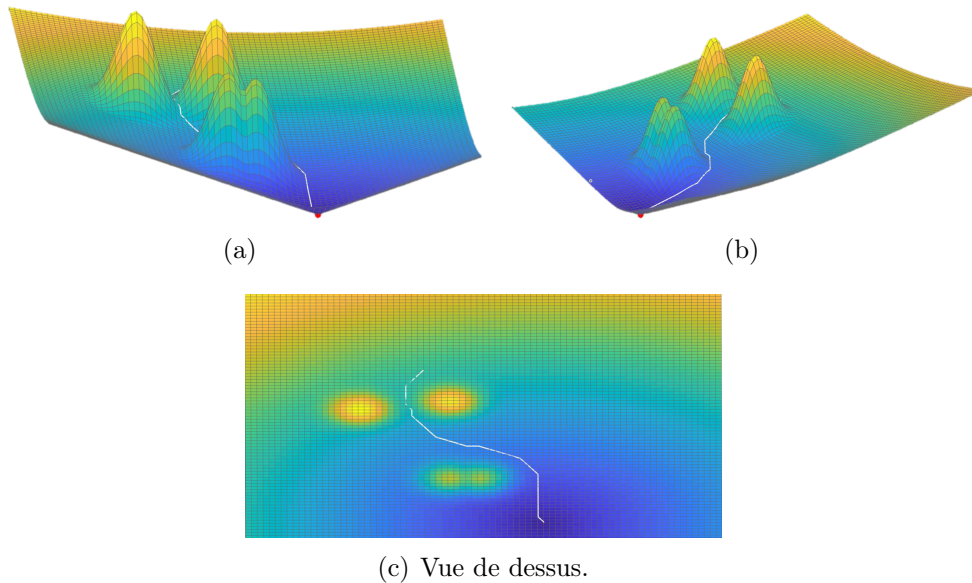


FIGURE 4.6 – Exemple d'APF ne permettant pas au véhicule de passer entre tous les obstacles ; son chemin calculé par descente de gradient est représenté par une ligne blanche.

Pour pallier ces problèmes, de nombreux travaux ont proposé des solutions ponctuelles. Liu et Zhao [2016] proposent de créer un point cible temporaire (en général appelé *waypoint*) lorsqu'un véhicule se trouve dans un minimum

local. Cela permet de calculer un nouveau champ de potentiel, temporaire, et donc de supprimer le minimum local. Par ailleurs, pour limiter les minima dus aux obstacles proches, la méthode proposée par les auteurs considère les obstacles dont les zones d'influence se recouvriraient, comme un seul obstacle de taille supérieure. Ceci permet de réduire considérablement les minima locaux dans les environnements à forte densité d'obstacles, mais a pour inconvénient d'augmenter la surface totale « interdite », ce qui peut par conséquent rallonger les chemins, voire mener au problème GNRON (et donc empêcher le véhicule d'atteindre sa cible).

Au delà des modifications locales ou temporaires des champs de potentiels, il est possible de modifier les définitions mêmes des champs.

Sun *et al.* [2017] proposent de résoudre le problème des minima locaux en modifiant la définition du champ répulsif lié aux obstacles. Dans leur définition de ce champ, un facteur intégrant la distance entre la cible et le véhicule est introduit. Cela assure le fait que le point cible est le seul minimum de l'environnement considéré.

Sun *et al.* [2017] s'intéressent également au problème d'oscillations pouvant apparaître à proximité d'obstacles, plus précisément à la limite du champ répulsif qui leur est associé. Ces oscillations rallongent inutilement le chemin du véhicule et peuvent considérablement le ralentir s'il est soumis à des contraintes cinématiques. Pour pallier ce problème, ils proposent que la distance maximale à parcourir à la prochaine itération dépende de l'angle entre le vecteur vitesse actuel et le prochain déplacement calculé. Si, cet angle est trop grand, alors la norme du prochain vecteur vitesse est limitée.

Au delà de l'introduction de la distance véhicule-cible dans la définition du champ de potentiel répulsif, Mac *et al.* [2016] proposent une modification du champ attractif faisant apparaître la vitesse du véhicule. Cette adaptation permet de supprimer le minimum local apparaissant dans la méthode traditionnelle lorsque véhicule, obstacle et point cible sont alignés. Mac *et al.* [2016] utilisent cette méthode de calcul de chemin en temps réel, en utilisant uniquement les capteurs embarqués sur leur véhicule de test (un multirobot AR. Drone 2.0).

Dans des environnements complexes, tous les obstacles ne représentent pas forcément le même danger. Pour traduire cette notion en termes mathématique, il est possible d'utiliser des « potentiels fractionnaires » [Poty, 2006; Receveur *et al.*, 2018]. En introduisant la notion de dérivée fractionnaire dans la définition des champs répulsifs, on peut modifier la forme de ces champs en fonction de la dangerosité des obstacles.

Il est également possible d'adapter la méthode des champs de potentiels à

l'évitement d'obstacles mobiles ayant des trajectoires inconnues. Dans ce cas, on peut introduire dans la définition du champ répulsif un facteur lié à la distance entre le véhicule et l'obstacle considéré dépendant du temps. De plus, le point cible visé par le véhicule peut être mobile également ; le champ attractif est alors régulièrement recalculé [Boonyarak et Prempraneerach, 2014; Ge et Cui, 2002].

Par ailleurs, la méthode des APF peut aussi être utilisée pour calculer le chemin de véhicules non-holonomes, en intégrant une loi de contrôle à l'algorithme [Boonyarak et Prempraneerach, 2014].

Enfin, cette méthode peut être utilisée dans un environnement discrétisé [Xiaodong Zhuang *et al.*, 2002], ce qui permet de diminuer significativement le coût calculatoire tout en maintenant une bonne qualité des résultats. L'algorithme de descente de gradient est alors discrétisé également : à chaque itération le véhicule se déplace sur une des mailles adjacentes à celle qu'il occupe.

4.3.5 Synthèse

Les méthodes globales peuvent fournir une trajectoire optimale en environnement complexe mais leur performance est soumise à la connaissance *a priori* de l'environnement.

L'intérêt des méthodes globales est de trouver une trajectoire (ou au moins un chemin) de qualité suffisante pour relier deux points donnés. La complexité de la recherche de chemin est d'autant plus importante que la zone est grande et complexe. Ce point n'est pas limitant lorsque les calculs sont menés sur des machines performantes puis transférés sur les véhicules.

Néanmoins, si les calculs sont menés avant la mission, le véhicule n'a connaissance que d'une trajectoire à suivre, et il ne pourra pas réagir à son environnement ; l'utilisation de cette pratique est donc limitée aux environnements parfaitement connus. Enfin, les zones d'intérêt considérées sont limitées à celles contenant uniquement des obstacles fixes, ou des obstacles mobiles dont les caractéristiques et mouvements sont connus.

4.4 Combinaison de plusieurs méthodes

Tant les méthodes globales que locales sont imparfaites, comme nous l'avons vu dans les sections précédentes. L'utilisation conjointe de plusieurs d'entre elles peut permettre de tirer profit des avantages de chacune tout en atténuant leurs faiblesses.

Compte tenu de la variété de méthodes de calcul de chemin, de très nombreuses combinaisons ont été étudiées. Nous présentons ici les plus pertinentes

rapport à notre contexte.

4.4.1 Association de méthodes locales

Pour pallier les inconvénients des modèles basés uniquement sur les forces virtuelles tels que l'absence de continuité du chemin calculé ou la possibilité de choir dans un minimum local, [Zhuoning et al. \[2010\]](#) proposent un modèle de mobilité pour UAV basé sur les forces virtuelles et la logique floue. L'environnement considéré contient des obstacles inconnus et le niveau de danger de la mission peut évoluer au cours de celle-ci, ce qui doit être pris en compte par le modèle de mobilité. Le calcul de chemin est réalisé par une méthode utilisant les forces virtuelles, intégrant un paramètre représentant le risque autorisé dans l'évitement des obstacles. La valeur de ce paramètre est déterminée en utilisant la logique floue en fonction de deux critères : le niveau de réactivité nécessaire et les capacités actuelles de la plateforme (son énergie restante, sa probabilité de défaillance etc.).

La logique floue peut également être combinée à l'approche de la fenêtre dynamique ([DWA](#)). Les configurations admissibles correspondant à la fenêtre de recherche de la [DWA](#) dépendent de plusieurs éléments, tels que la vitesse de déplacement et la distance à maintenir autour des obstacles. En fonction de la situation, l'importance de chaque critère peut évoluer. [Hong et al. \[2015\]](#) proposent donc d'utiliser la logique floue dans le but de les pondérer de manière dynamique.

4.4.2 Association de méthodes globales

La méthode [PSO](#) peut fournir des chemins de bonne qualité, mais en général sous la condition d'un grand nombre d'itérations. Afin d'accélérer le processus de convergence, [Huang et Tsai \[2011\]](#) proposent d'intégrer un algorithme génétique⁶ au cœur de la méthode [PSO](#). Plus précisément, une fois que les particules ont évalué la qualité de leur solution, un algorithme génétique opère des mutations et croisements entre celles-ci, avant la recherche de la meilleure solution globale et la mise à jour en position et en vitesse de chaque particule. Le chemin résultant est ensuite lissé.

Dans le cas de l'utilisation simultanée de plusieurs véhicules en environnement statique, [Garip et al. \[2018\]](#) proposent un calcul de chemin utilisant à la fois A^* et un algorithme génétique. Dans leur méthode, la fonction heuristique utilisée dans A^* pour évaluer la pertinence des nœuds (du graphe dans lequel le chemin est cherché) est également utilisée dans l'algorithme génétique en tant que fonction objectif.

6. Les algorithmes génétiques font partie de la famille des algorithmes évolutionnaires.

4.4.3 Combinaison de méthodes globales et locales

L'association d'une méthode globale et d'une méthode locale peut être particulièrement efficace. L'idée générale de ce type d'association est de calculer un chemin en amont de la mission par une méthode globale, puis de gérer la trajectoire et les événements ponctuels imprévus par une méthode locale.

Dans le cadre de calcul de trajectoire pour voitures autonomes, de nombreux critères sont à prendre en considération, notamment le danger, la longueur et le temps de trajet, l'énergie consommée et le confort du passager [Receveur *et al.*, 2017; Receveur, 2019]. Tous les éléments n'ayant pas la même importance, on peut pondérer chaque critère et rechercher à minimiser la fonction de coût correspondant à la somme pondérée de ces éléments. Dans le contexte de la voiture autonome, deux échelles sont à prendre en compte : la ville et le véhicule. Plus précisément, [Receveur, 2019] propose d'utiliser un algorithme génétique pour le calcul de chemin global (quelles rues emprunter), recherchant l'optimalité du trajet complet. La trajectoire, elle, doit être réactive aux imprévus (piéton traversant la route, véhicule lent, etc.). La trajectoire est calculée de manière locale par une méthode basée sur les champs de potentiels. Dans le but de suivre une trajectoire quasi-optimale, une cible attractive dynamique y est alors considérée. Cette cible suit la trajectoire calculée par l'algorithme génétique.

De manière similaire, Dong *et al.* [2011] proposent un calcul de trajectoire pour UAVs, utilisant l'algorithme A* pour le calcul de chemin global et les forces virtuelles pour la réactivité ponctuelle.

4.4.4 Synthèse

L'utilisation conjointe de plusieurs méthodes de calcul de chemin permet de tirer profit des avantages de chacune et d'atténuer leurs faiblesses, tels que la présence de minima locaux ou les longues durées de convergence.

Lorsque l'environnement est globalement connu et seulement soumis à des aléas locaux, l'utilisation d'une méthode globale de calcul de chemin, associée à une gestion de trajectoire fournit de bons résultats.

La combinaison de plusieurs méthodes répond en général à des besoins spécifiques et ne peut pas toujours être généralisée.

4.5 Résumé

Dans ce chapitre, nous avons introduit les notions liées à la mobilité de véhicules, notamment celles de chemin et trajectoire (section 4.1, p. 40). Les modèles de mobilité les plus couramment utilisés dans le contexte de véhicules sans pilotes à bord ont été présentés, et nous avons distingué deux catégories : les approches locales et les approches globales (sections 4.2, p. 42 et 4.3, p. 46

respectivement). Les deux types de méthodes présentent des résultats satisfaisants, mais n'ont pas le même domaine d'application. D'une part, les méthodes globales peuvent fournir une trajectoire de bonne qualité, voire optimale, entre deux points donnés, mais à la condition de connaître l'environnement *a priori*. Quant aux méthodes locales, elles sont réactives et ont un faible coût calculatoire, mais ne garantissent pas d'atteindre le point cible sélectionné. Compte tenu de la complémentarité de ces approches, l'association d'une méthode globale et d'une méthode locale est bien adaptée au calcul de chemin dans des environnements partiellement connus (section 4.4, p. 56).

Dans cette thèse, nous considérons successivement deux types d'environnement, tous deux inconnus. Ces deux contextes, présentés ci-dessous, contraignent le choix de méthode de mobilité à utiliser.

Dans un premier temps, l'essaim de drones doit traverser un passage fortement contraint afin d'atteindre une zone d'intérêt (chapitre 5, p. 62). Dans ce contexte, les drones n'ont pas à mémoriser l'environnement car ils ne le traversent qu'une seule fois. De plus, ils ont connaissance d'une direction globale à suivre mais n'ont pas à atteindre un point précis. Enfin, les drones sont soumis à la contrainte de se déplacer en formation compacte dans ce passage. Dans ce contexte, une approche locale nous a paru adaptée et nous avons choisi de créer un modèle de mobilité utilisant les forces virtuelles.

Dans un second temps, nous nous intéressons aux déplacements de drones dans un environnement de type urbain, avec ou sans contrainte de maintien de formation (chapitres 6, p. 91 et 7, p. 124). Les drones auront à atteindre différents points cible en empruntant les chemins les plus courts possibles, tout en découvrant leur environnement à la volée. Compte tenu du caractère inconnu de l'environnement, une méthode locale est nécessaire. Néanmoins, la collaboration interplateforme permet d'étendre la connaissance de l'environnement de chacun des drones de l'essaim. Ainsi, il est possible de calculer un chemin et pas seulement un prochain déplacement, et une approche globale peut être utilisée en complément de l'approche locale. Nous choisissons donc de baser le calcul de chemin sur les champs de potentiels.

Deuxième partie

Modèles de mobilité pour un
essaim collaboratif de drones
autonomes en environnement
complexe

Chapitre 5

Modèle basé sur les forces virtuelles

Sommaire

5.1	Contexte et besoins opérationnels	63
5.1.1	Hypothèses	63
5.1.2	Contraintes	64
5.2	Forces virtuelles et principe du modèle	65
5.2.1	Distance de sécurité	65
5.2.2	Distance de référence inter-drones	66
5.2.3	Attraction-répulsion entre drones	67
5.2.4	Répulsion due aux obstacles	68
5.2.5	Attraction vers un point-cible	69
5.2.6	Combinaison des forces appliquées sur un drone	71
5.2.7	Distance maximale à parcourir par itération	71
5.2.8	Synthèse	73
5.3	Utilisation des forces en fonction du besoin	73
5.3.1	Scénario	73
5.3.2	Préparation à la surveillance	75
5.3.3	Surveillance	75
5.3.4	Mise en formation compacte	76
5.3.5	Traversée du tunnel	78
5.3.6	Synthèse	79
5.4	Simulations et résultats	79
5.4.1	Environnements et paramètres de simulation	79
5.4.2	Métriques et résultats	81
5.5	Résumé	89

Dans cette étude, nous considérerons un essaim de multirobots évoluant dans le cadre d'une mission de surveillance. Celle-ci se déroule successivement dans plusieurs zones d'intérêt, séparées par un tunnel, dont la traversée nécessite la mise en place d'une formation compacte. De plus, le tunnel peut contenir des obstacles fixes non connus *a priori*.

5.1 Contexte et besoins opérationnels

On considère une mission de surveillance de deux zones d'intérêt (notées ZI) séparées par un tunnel (voir figure 5.1, p. 63). La mission a pour objectif la surveillance successive des deux ZIs et l'identification d'une cible immobile présente dans une de ces deux zones. L'objectif de cette première étude est la modélisation de la mise en formation compacte et le maintien de celle-ci en environnement contraint, c'est-à-dire, ici, le tunnel.

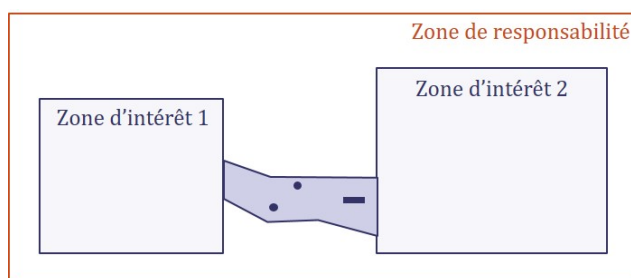


FIGURE 5.1 – Vue schématique de l'environnement considéré.

Dans cette étude, les seuls obstacles extérieurs à l'essaim qui sont considérés sont fixes. Un drone entrant en collision avec un obstacle (fixe ou un autre drone) est considéré comme perdu. Enfin, les besoins définis en section 3.2 (p. 26) (autonomie, résilience, évolution en environnement inconnu) s'ajoutent à ceux décrits ici. Les drones considérés sont des multirobots d'envergure inférieure à un mètre, pouvant tous remplir des missions de surveillance (donc détecter des objets), mais un seul d'entre eux peut effectuer une identification. Ce drone-ci est connu de tous.

Tous les drones embarquent les capteurs suivants (voir détails dans les sections 3.6, p. 34 et 3.7, p. 35) :

- Des caméras stéréos montées sur cardans, avec une portée de 24 mètres ;
- Un module de communication XBee d'une portée de 40 mètres.

5.1.1 Hypothèses

Nous concentrons nos travaux sur les déplacements des drones en environnement contraint, et on pose donc plusieurs hypothèses au niveau de la

mission. D'une part, on considère que les seules raisons de perte d'engins sont les collisions, entre eux ou avec un obstacle fixe. D'autre part, on suppose que les seuls obstacles présents se trouvent à l'intérieur du tunnel. De plus, on admet que la batterie est suffisante pour remplir la mission et que les capteurs embarqués restent en parfait état de fonctionnement. Enfin, on suppose que les données capteurs sont parfaites, qu'elles sont traitées instantanément et que les capteurs couvrent les 360 ° autour du drone.

Des hypothèses sont également posées au niveau du modèle ; on considère que chaque drone de l'essaim surveille l'ensemble de la zone dans la portée de son capteur, ainsi, dès que la cible se trouve dans cette zone, elle est considérée comme détectée. De même, dès que celle-ci est dans la portée de détection du drone pouvant effectuer une identification, elle est considérée comme identifiée.

Dans cette mission, les drones volent à une altitude fixe, identique pour tous. Les modèles de mobilité sont donc développés en deux dimensions. Les drones évoluant à la même altitude, le modèle doit permettre l'évitement des drones entre eux.

Les déplacements des drones se font de manière discrète dans le temps, mais l'espace est considéré comme continu. On considère une itération par seconde. À chaque itération, chaque drone calcule son prochain mouvement en fonction de son objectif de mission, des obstacles à éviter et des contraintes de connectivité avec ses voisins, puis se déplace dans la direction résultante.

5.1.2 Contraintes

Durant la mission, des pertes de communications peuvent survenir. De plus, si un drone entre en collision avec une autre plateforme ou un obstacle fixe, alors on le considère comme perdu. Pour répondre au besoin d'assurer la mission, des solutions doivent être mises en place au niveau algorithmique.

Au cours de la mission de surveillance, on souhaite que tous les drones aient la vision la plus récente possible de la zone d'intérêt. Pour cela, il faut que les drones s'échangent régulièrement les informations dont ils disposent. On souhaite donc qu'à tout instant, il existe un chemin entre n'importe quelle paire de nœuds du graphe : on recherche un graphe connexe. Puisque les seuls obstacles présents dans la zone de surveillance sont les drones de l'essaim, si le modèle de mobilité assure l'absence de collision entre membres de la flotte, alors il n'y aura pas de collision d'engins. On peut donc en conclure qu'une 1-connexité (définition 6, p. 21) est suffisante dans ce cas.

Des obstacles fixes sont considérés dans le tunnel, ce qui augmente la possibilité de collisions. À terme, un partage des positions des obstacles détectés sera élaboré, permettant aux drones d'anticiper les évitements. Pour cette raison, on souhaite, dès à présent, maintenir un graphe 2-connexe, afin que la perte d'un engin permette toujours l'échange de messages entre n'importe quelle paire de nœuds. De plus, un maintien de formation est souhaité dans

le tunnel, mais la présence d'obstacles peut empêcher cela. Par conséquent, les distances entre certains drones peuvent augmenter au-delà de la portée de communication et donc rompre des liens de communication existants. Ainsi, nous chercherons également à avoir un graphe arête 2-connexe (définition 7). Il est à noter que ces propriétés de connexité ne pourront pas nécessairement être remplies tout au long de la traversée du tunnel, tout comme le maintien de formation. En effet, la densité et la position des obstacles présents peuvent rendre impossible le maintien de formation.

On notera que si un graphe est 2-connexe il est nécessairement arête 2-connexe. On ne parlera donc que de la contrainte la plus forte (la 2-connexité) dans la suite de cette section.

5.2 Forces virtuelles et principe du modèle

Pour remplir la mission, les drones se déplacent selon des directions calculées dynamiquement à partir d'une représentation du système global sous forme de forces virtuelles attractives ou répulsives. Cette section présente les forces utilisées dans le modèle de mobilité permettant aux drones de se déplacer en maintenant les liens de communication et en évitant les collisions. À chaque itération, chaque drone fait un bilan des forces qui s'appliquent à lui, afin de déterminer son déplacement suivant.

Les forces à valeurs positives sont attractives, les forces à valeur négatives sont répulsives. Un enjeu crucial de la mission étant l'évitement de collisions, les forces attractives ne doivent jamais être prépondérantes. Nous choisissons arbitrairement que la magnitude d'une force attractive ne dépasse pas la moitié de la magnitude maximale d'une force répulsive, ce qui s'avèrera suffisant pour éviter les collisions. Nous faisons donc le choix que la magnitude des forces soit comprise entre -1 et $1/2$.

5.2.1 Distance de sécurité

Pour éviter toute collision aux drones, on souhaite qu'une distance de sécurité soit maintenue autour de chacun d'eux. Dans cette première étude, on choisit arbitrairement qu'il doit toujours y avoir une distance au moins égale au diamètre du drone autour de lui. Nous pouvons définir une distance de sécurité ne dépendant pas de la vitesse des drones car leur inertie est négligée. On rappelle que l'on considère que les drones ont une enveloppe circulaire.

La distance de sécurité D_{sec} , calculée à partir du centre du drone, est donc définie de la manière suivante :

$$D_{\text{sec}} = 1,5 \times d_{\text{drone}}, \quad (5.1)$$

où d_{drone} représente le diamètre du drone.

La figure 5.2 ci-dessous illustre la définition de la distance de sécurité par rapport à la taille du drone.

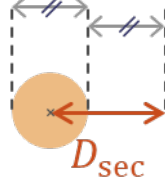


FIGURE 5.2 – Distance de sécurité à maintenir.

Il est à noter que tous les drones de l'essaim étant identiques, la distance de sécurité est la même pour tous. De plus, la portée du capteur permettant la détection des obstacles est très supérieure à la taille du drone, donc très supérieure à la distance de sécurité.

5.2.2 Distance de référence inter-drones

Lorsqu'un essaim de drones est utilisée pour remplir une mission de surveillance ou de couverture de zone, un compromis est à trouver entre l'étalement des appareils permettant de couvrir une large zone et leur proximité assurant le maintien des liens de communication. Puisqu'on cherche à préserver la connexité de l'essaim, la distance inter-drones doit être inférieure à la portée de communication. On souhaite également remplir la mission de surveillance le plus rapidement possible, donc minimiser le recouvrement des zones vues par les capteurs embarqués. De plus, il est en général préférable de ne pas laisser d'espace non observé entre les drones pour minimiser le nombre de passages dans chaque zone ; la distance inter-drones doit donc être inférieure à deux fois la portée du capteur embarqué. Ainsi, la distance de référence entre deux engins est définie comme suit :

$$D_{\text{ref}}(k) = k \times \min(2p_{\text{capteur}}, p_{\text{comm}}) \quad (5.2)$$

où p_{capteur} désigne la portée du capteur permettant de détecter des obstacles et la cible à identifier, p_{comm} la portée du module radio et k est un réel strictement positif.

Pour que le modèle soit valide, la distance recherchée entre deux drones doit être supérieure à la distance de sécurité. Comme représenté sur le graphique 5.3, on doit donc nécessairement choisir un k supérieur à 0,015, pour les paramètres donnés en légende.

Pour que des drones séparés d'une distance D_{ref} puissent échanger des messages, indépendamment de la portée capteur de détection d'obstacle, on doit choisir un k inférieur à 1.

Dans la suite, on aura donc nécessairement $k \in]0, 015 ; 1[$.

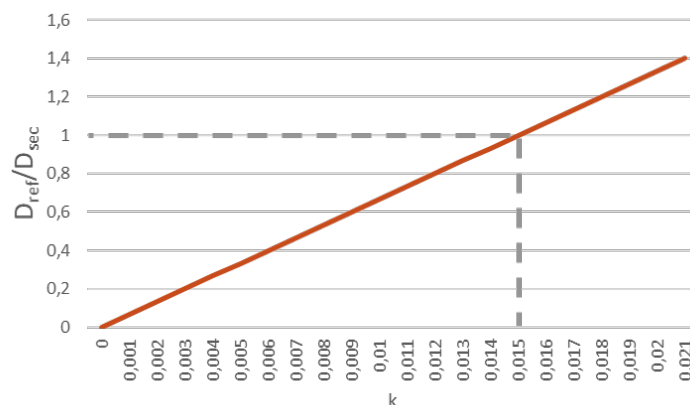


FIGURE 5.3 – Rapport $\frac{D_{\text{ref}}}{D_{\text{sec}}}$ en fonction du réel k , calculé avec les paramètres suivants : $d_{\text{drone}} = 0,4 \text{ m}$; $p_{\text{comm}} = 40 \text{ m}$; $p_{\text{capteur}} = 24 \text{ m}$. Si $k > 0,015$, alors la distance de référence est supérieure à la distance de sécurité.

Plus k est proche de 1, plus les drones seront éloignés. Durant la phase de surveillance on prendra donc un k grand.

Durant la traversée du tunnel on cherche également à maintenir une certaine distance inter-drones. Dans ce contexte là, fortement contraint, on souhaite réduire les distances entre les engins. On choisira donc un k proche de 0,015.

5.2.3 Attraction-répulsion entre drones

Des forces virtuelles d'attraction-répulsion entre les drones voisins sont utilisées afin de tendre vers la distance de référence D_{ref} définie dans la section précédente.

Quand la force est négative, les drones s'éloignent, dans le cas contraire ils se rapprochent. On rappelle que la répulsion est plus forte que l'attraction car éviter les collisions est une priorité, même si un lien de communication doit être perdu pour cela.

Les forces d'attraction-répulsion entre drones sont utiles pour maintenir l'essaim en formation, qu'elle soit compacte comme dans le tunnel où plus étendue comme dans le cas de la surveillance.

Pour définir la force d'attraction-répulsion inter-drones, il faut prendre en compte plusieurs objectifs :

- Les collisions doivent être évitées. Si la distance inter-drones est inférieure à D_{sec} , la répulsion maximale doit être appliquée.
- D_{ref} est la distance de référence inter-drones. Il ne doit donc n'y avoir ni attraction ni répulsion à cette distance, et la force doit être continue autour de celle-ci (attractive aux distances inférieures et inversement).

- Il ne doit pas y avoir de discontinuité de couverture de zone. L'attraction doit donc être maximale au delà d'une distance de $2p_{\text{capteur}}$.
- Au delà de p_{comm} , les drones ne peuvent plus communiquer. Et comme on suppose $p_{\text{comm}} > p_{\text{capteur}}$, ils ne peuvent pas se détecter non plus. Aucune force n'est donc appliquée.

À partir de ces différents objectifs on définit la force d'attraction-répulsion entre les drones i et j , F_{ij}^{ar} , de la manière suivante :

$$F_{ij}^{ar}(d_{ij}) = \begin{cases} -1 & \text{if } d_{ij} \leq D_{\text{sec}} \\ \frac{d_{ij} - D_{\text{ref}}}{D_{\text{ref}} - D_{\text{sec}}} & \text{if } D_{\text{sec}} < d_{ij} \leq D_{\text{ref}} \\ \frac{k}{2(1-k)} \left(\frac{d_{ij}}{D_{\text{ref}}} - 1 \right) & \text{if } D_{\text{ref}} \leq d_{ij} < \frac{D_{\text{ref}}}{k} \\ \frac{1}{2} & \text{if } \frac{D_{\text{ref}}}{k} \leq d_{ij} < p_{\text{comm}} \\ 0 & \text{else} \end{cases} \quad (5.3)$$

où d_{ij} représente la distance séparant les drones i et j .

La grandeur D_{ref} dépend du minimum de $2p_{\text{capteur}}$ et p_{comm} . Pour cette raison, deux représentations graphiques de cette force existent, comme représenté en figure 5.4.

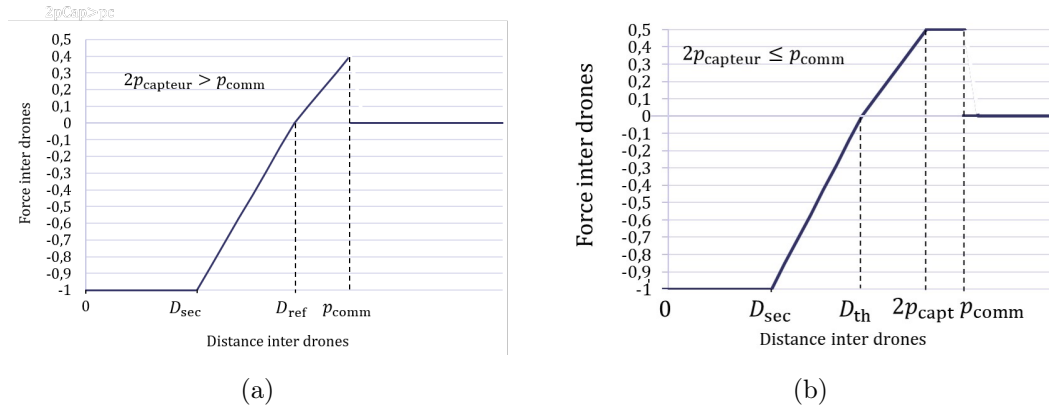


FIGURE 5.4 – Graphique représentant la magnitude de la force d'attraction-répulsion entre deux drones en fonction de la distance les séparant, dans les cas où $2p_{\text{capteur}} > p_{\text{comm}}$ (a) et $2p_{\text{capteur}} \leq p_{\text{comm}}$ (b)

La figure 5.5 illustre le comportement d'un essaim soumis uniquement à ces forces dans un contexte statique (indépendamment de toute mission).

5.2.4 Répulsion due aux obstacles

Dans cette étude, des obstacles peuvent être présents dans le tunnel séparant les deux zones d'intérêt. De plus, les parois de ce tunnel sont également

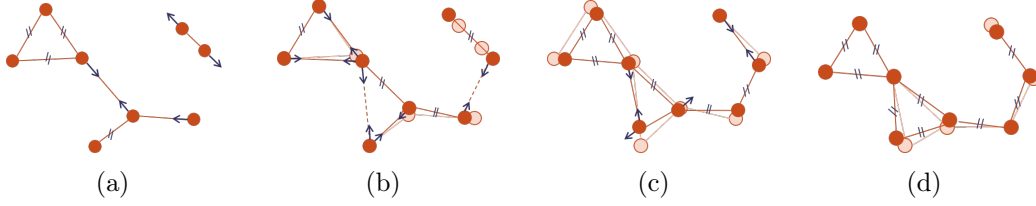


FIGURE 5.5 – Exemples de déplacements dans le but de tendre vers une distance de référence entre les drones sur 4 itérations. À l’instant considéré, les drones sont représentés par des disques orange et les liens de communication par des traits orange. Les pointillés représentent la création d’un lien de communication. Les disques et lignes plus clairs rappellent l’état du système à l’instant précédent.

considérées comme des obstacles. Nous rappelons que seuls des obstacles fixes sont considérés dans cette étude, en plus des drones constituant l’essaim.

Les forces de répulsion associées aux obstacles ont pour unique but d’éviter les collisions drones – objet fixe. La valeur de référence dans le calcul de cette force est donc la distance à maintenir autour du drone pour le garder en sécurité, D_{sec} (définie en équation 5.1)

L’influence d’un obstacle sur un drone n’a pas besoin d’exister dans tout l’espace. On choisit arbitrairement qu’au delà de $3D_{\text{sec}}$, un drone ne subit pas d’effet d’un obstacle fixe. Au contraire, si la distance entre le drone et l’obstacle est inférieure à la distance de sécurité, il faut appliquer la force de répulsion maximale pour éviter une collision. Entre ces deux valeurs, on opte pour une évolution linéaire. La force de répulsion due à l’obstacle α noté o_α appliquée sur le drone i est notée $F_i^{o_\alpha}$, représentée en figure 5.6 et est définie de la manière suivante :

$$F_i^{o_\alpha}(d_{io_\alpha}) = \begin{cases} -1 & \text{if } d_{io_\alpha} < D_{\text{sec}} \\ \frac{1}{2D_{\text{sec}}}d_{io_\alpha} - \frac{3}{2} & \text{if } D_{\text{sec}} \leq d_{io_\alpha} < 3D_{\text{sec}} \\ 0 & \text{else} \end{cases} \quad (5.4)$$

5.2.5 Attraction vers un point-cible

Pour remplir leur mission, les drones doivent se déployer sur les zones d’intérêt, les parcourir et traverser le tunnel. Pour cela, ils calculent au fur et à mesure de la mission des points-cible vers lesquels ils sont attirés¹. La force d’attraction vers leur cible s’ajoute aux forces d’attraction-répulsion entre drones.

L’attraction vers la cible ne doit pas être prépondérante ; les autres forces permettent elles aussi de remplir des besoins de la mission. Quand un drone

1. La création des points cible en fonction du besoin est présentée en section 5.3, p. 73.

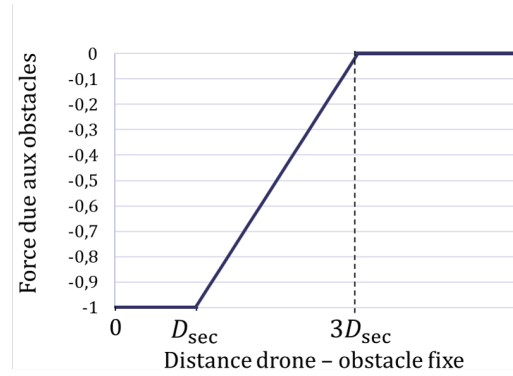


FIGURE 5.6 – Graphique représentant la magnitude de la force de répulsion due à un obstacle en fonction de la distance drone – obstacle.

effectue un bilan des forces, il subit des attractions ou des répulsions de chacun de ses voisins, auxquelles s'ajoutent potentiellement des répulsions associées à plusieurs obstacles. Néanmoins, il n'est attiré que vers un seul point-cible. Pour permettre à un drone de se déplacer dans la direction souhaitée pour la mission, il faut donc que l'attraction vers la cible ne soit pas négligeable dans le bilan des forces. Ainsi, on attribue la magnitude maximale (c.-à-d. $\frac{1}{2}$) à l'attraction vers la cible tant que le drone n'est pas suffisamment proche de celle-ci. En deçà d'une distance de D_{ref} l'attraction doit diminuer afin d'éviter les oscillations autour de la cible. On choisit une évolution linéaire.

La force d'attraction appliquée sur le drone i par sa cible c_i , est notée $F_i^{c_i}$, et est définie en équation (5.5) et représentée en figure 5.7.

$$F_i^{c_i}(d_{ic_i}) = \begin{cases} \frac{1}{2} & \text{if } d_{ic_i} > D_{ref} \\ \frac{d_{ic_i}}{2D_{ref}} & \text{else} \end{cases} \quad (5.5)$$

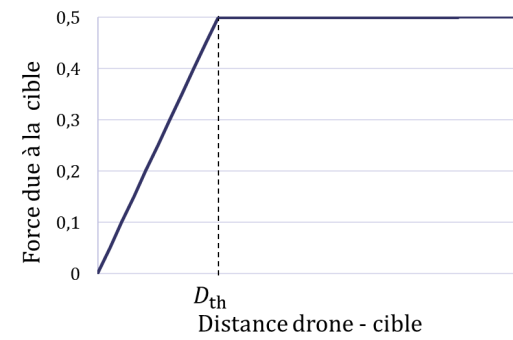


FIGURE 5.7 – Graphique représentant la magnitude de la force d'attraction vers un point-cible en fonction de la distance drone-cible.

5.2.6 Combinaison des forces appliquées sur un drone

À chaque itération, chaque drone est soumis à plusieurs forces virtuelles. Pour calculer son prochain déplacement, le drone fait une somme vectorielle de toutes les forces qui s'appliquent à lui. Le déplacement du drone se fait dans la direction du vecteur résultant. La distance parcourue est proportionnelle à la norme de ce vecteur et bornée par une distance maximale, d_{\max} , définie dans la section suivante (5.2.7).

5.2.7 Distance maximale à parcourir par itération

Le risque de collision le plus grand est celui impliquant deux drones, puisque dans ce cas le drone et son obstacle (l'autre drone) se déplacent tous les deux. Le cas où le risque de collision est maximal est le suivant (illustré en figure 5.8²). À l'itération n , deux drones sont séparés d'une distance égale à D_{ref} (ils ne subissent aucune force de l'un vers l'autre) et leurs point-cibles respectifs se trouvent dans la direction de leur voisin. À l'itération suivante, les drones sont donc espacés de $D_{\text{ref}} - 2d_{\max}$.



(a) Itération n , les drones ne subissent que l'attraction vers leurs points-cible respectifs.

(b) Position des drones à l'itération $n + 1$.

FIGURE 5.8 – Déplacements de 2 drones dans le cas le plus dangereux.

La distance maximale parcourue en une itération, d_{\max} , est donc soumise à la contrainte suivante :

$$d_{\max} \leq \frac{1}{2} \left(D_{\text{ref}} - D_{\text{sec}} - \frac{1}{2}d_{\text{drone}} \right). \quad (5.6)$$

Une seconde contrainte existe : lorsque deux drones sont espacés de D_{ref} , et qu'ils ne subissent donc aucune force inter-drones, on souhaite néanmoins qu'ils maintiennent un lien de communication après leur mouvement (car on souhaite toujours maintenir les liens de communication entre les plateformes). Ce cas est illustré en figure 5.9 (p. 72).

2. On rappelle qu'un drone évalue toujours la distance entre son centre et l'obstacle le plus proche.

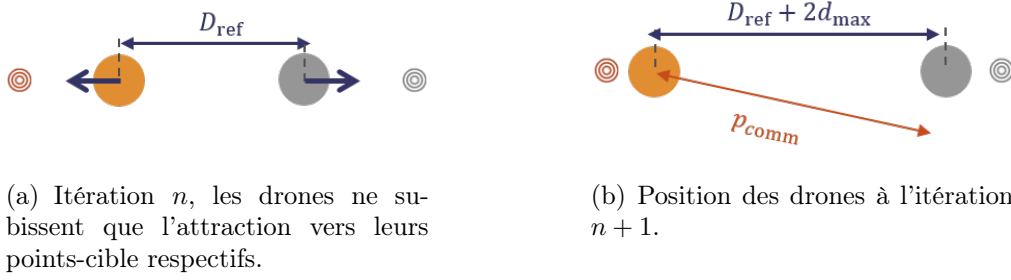


FIGURE 5.9 – Déplacements de deux drones s'éloignant l'un de l'autre.

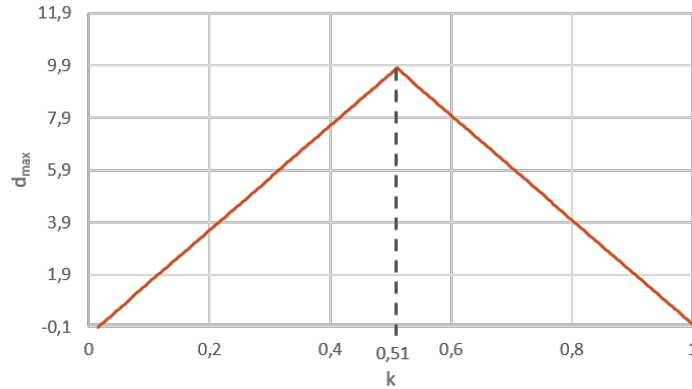
La distance d_{\max} , est donc également soumise à la contrainte suivante :

$$d_{\max} \leq \frac{1}{2} (p_{\text{comm}} - D_{\text{ref}}). \quad (5.7)$$

Enfin, pour que les drones remplissent leurs missions le plus rapidement possible, il faut que la distance parcourue à chaque itération soit la plus grande possible. Pour remplir cet objectif tout en étant soumis aux contraintes (5.6) et (5.7), on définit donc d_{\max} comme suit :

$$d_{\max} = \frac{1}{2} \min \left(D_{\text{ref}} - D_{\text{sec}} - \frac{1}{2} d_{\text{drone}}, p_{\text{comm}} - D_{\text{ref}} \right). \quad (5.8)$$

Une fois les capteurs et le drone choisis (c.-à-d. p_{comm} , p_{capteur} , taille du drone et D_{sec} fixés), la distance parcourue à chaque itération ne dépend plus que du facteur k utilisé dans la définition de la distance inter-drones de référence D_{ref} . La figure 5.10 représente d_{\max} en fonction de k .


 FIGURE 5.10 – Graphique représentant la distance d_{\max} en fonction du réel k pour les paramètres suivants : $d_{\text{drone}} = 0,4 \text{ m}$; $p_{\text{comm}} = 40 \text{ m}$; $p_{\text{capteur}} = 24 \text{ m}$.

Pour les k petits, la contrainte (5.6) (liée à la distance de sécurité) est limitante. Dans le cas illustré par le graphique, on obtient une distance d_{\max}

négative si $k \leq 0,02$. On choisira donc par la suite $k > 0,02$. À partir de $k = 0,51$, la contrainte 5.7, liée à la portée de communication, est limitante.

5.2.8 Synthèse

Cette section a présenté les différentes forces virtuelles utilisées dans le calcul des déplacements d'un drone, pour éviter les collisions (entre drones et avec des obstacles) tout en maintenant les liens de communication. Les forces d'attraction vers des points cible servent quant à elles au déplacement des drones afin de remplir la mission.

La section suivante présente les modèles de mobilité définis à partir de ces forces virtuelles.

5.3 Utilisation des forces en fonction du besoin

Au cours de la mission, les drones doivent successivement surveiller des zones d'intérêt et traverser un tunnel. Les forces virtuelles définies précédemment s'appliquent aux drones pour le calcul de leurs déplacements. Cette section présente le scénario (section 5.3.1) ainsi que le modèle de mobilité associé à chaque étape du scénario (définition des points-cible et des forces utilisées), dans les sections 5.3.2 (p. 75) à 5.3.5 (p. 78).

5.3.1 Scénario

Pour remplir les missions de surveillance successives dans deux zones d'intérêt, nous découpons le scénario en plusieurs étapes (figure 5.11, p. 74).

Au lancement de la mission tous les engins reçoivent les coordonnées des zones d'intérêt ainsi que celles d'entrée du tunnel. Ils connaissent également le nombre de drones constituant l'essaim, ainsi que leur propre position.

Au début de la mission, les drones ne se trouvent pas nécessairement dans la zone d'intérêt, et ne sont, *a fortiori*, pas en configuration adaptée à la surveillance. La première étape consiste donc à placer les drones dans la zone d'intérêt en formation pour la surveillance. Ensuite, l'essaim surveille la première zone d'intérêt jusqu'à ce qu'une entité extérieure (par exemple un drone de haute altitude) envoie l'ordre d'aller observer la seconde. À la réception de la demande, les drones se dirigent vers l'entrée du tunnel et établissent une formation compacte. La traversée du tunnel constitue la quatrième étape. Enfin, les drones répètent les premières étapes en se mettant en configuration puis en surveillant la zone, jusqu'à ce qu'une cible soit détectée et identifiée.

Une fois le tunnel traversé, les étapes précédentes sont répétées, on ne les détaille donc pas. La mission d'identification ne sera pas traitée dans cette étude.

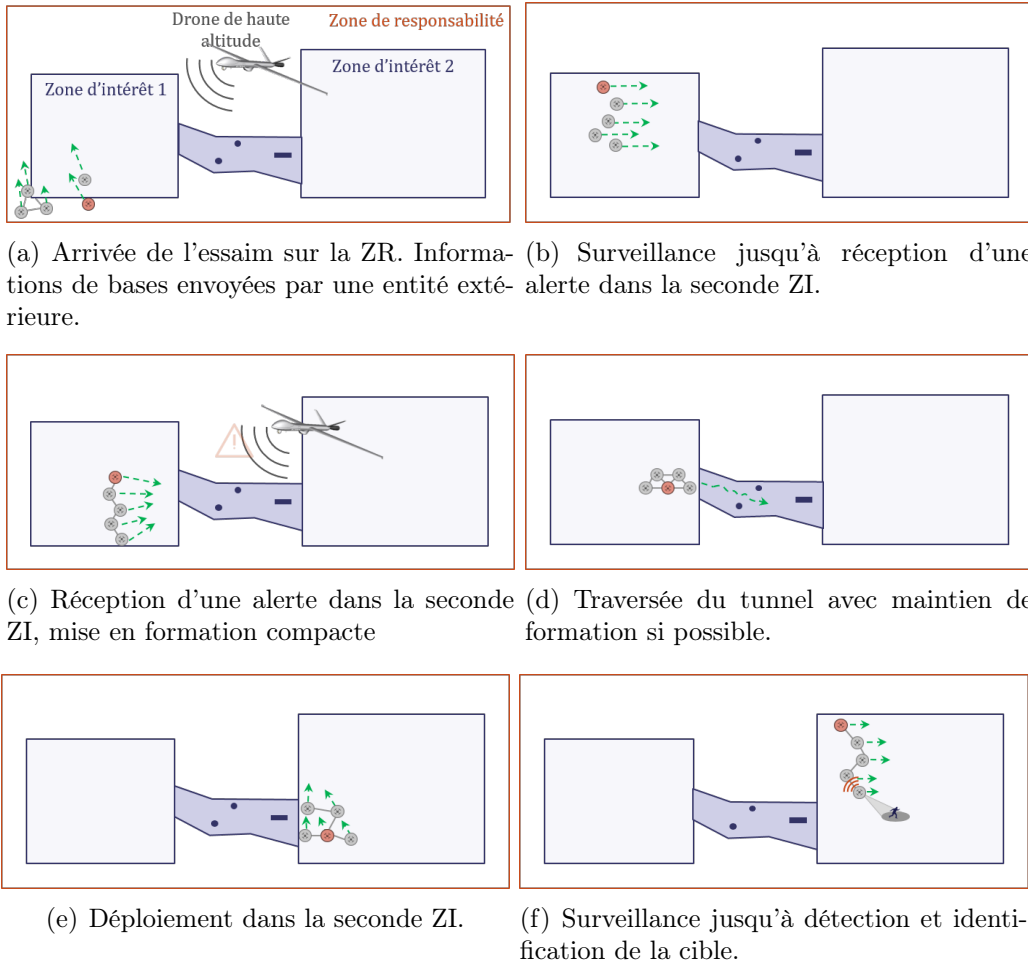


FIGURE 5.11 – Description des différentes étapes du scénario. Le drone permettant l'identification de la cible est représenté en rouge, les autres en gris. Les flèches vertes illustrent les déplacements des engins.

5.3.2 Préparation à la surveillance

L'objectif de cette étape est de mettre les drones en position pour commencer la surveillance.

Durant la surveillance, l'essaim devra être connexe tout en couvrant une zone aussi large que possible. Pour cela, une configuration en ligne où les drones sont espacés de D_{ref} est adaptée. Ainsi, cette première étape sera terminée quand les drones seront alignés, à partir d'un coin de la zone d'intérêt. Chaque drone calcule son premier point-cible dans cette optique, à partir des coordonnées de la zone d'intérêt et de son identifiant : les drones s'alignent par ordre croissant d'identifiant.

Durant le trajet jusqu'à ces points-cible, les forces appliquées sont la répulsion entre drones et les attractions vers les points-cible en question. Comme la surveillance ne commence que lorsque tous les drones ont atteint leur point-cible et que la configuration à atteindre remplit la condition de connexité de l'essaim, il n'est pas nécessaire d'appliquer les forces d'attraction permettant de maintenir les liens de communication. De plus, l'application de celles-ci ralentirait les drones.

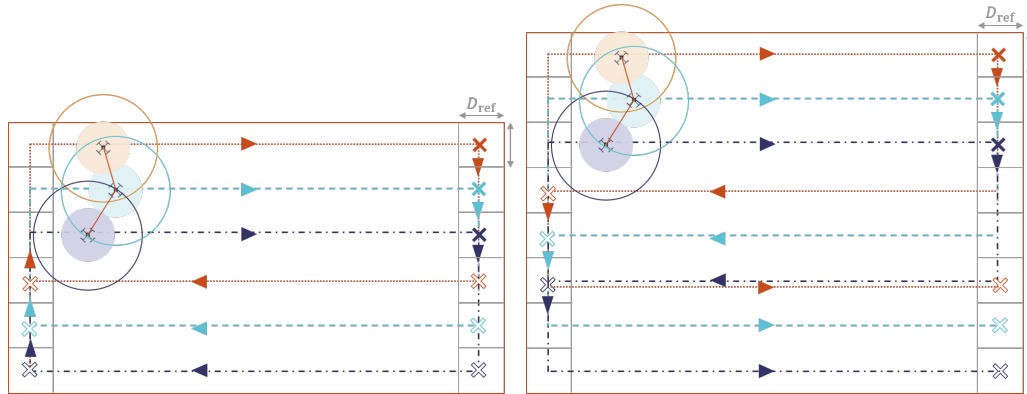
5.3.3 Surveillance

Un modèle simple a été utilisé pour la surveillance : les drones doivent maintenir leur configuration en ligne et parcourir la zone en boustrophédon³. Ce modèle de mobilité permet de maintenir la connexité de l'essaim. Ainsi, il suffira qu'un seul drone reçoive l'ordre de changer de zone de surveillance pour que tous les membres de l'essaim puissent recevoir l'information. De plus, ce modèle étant déterministe, il permet d'évaluer facilement le temps de parcours total de la zone.

Pour suivre ce mouvement en boustrophédon, les drones calculent successivement des point-cibles situés d'une extrémité à l'autre de la zone d'intérêt, en maintenant leur formation en ligne avec un espacement interdrones égal à D_{ref} . La figure 5.12 illustre les mouvements des drones et les emplacements des points-cible successifs.

Dès qu'un drone a atteint un point-cible, il calcule le point-cible suivant, mais ne commence à s'y rendre que lorsque ses voisins à 2 sauts (cf. définition p. 19) ont atteint leur propre cible. Ceci permet de maintenir la connexité de l'essaim.

3. Boustrophédon : Mode d'écriture archaïque [...], le boustrophédon consiste à tracer les lignes alternativement de gauche à droite et de droite à gauche, comme le bœuf (*bous* en grec) labourant le champ (*strophein*, tourner) trace les sillons continus. Cette façon d'écrire se rencontre dans des langues aussi différentes que le sudarabique et l'étrusque, le vieux grec et celle des Indiens Cuna (Panamá), le cananéen ou le latin; elle concerne aussi bien les pictogrammes que les écritures syllabiques ou alphabétiques.



(a) Cas de proportionnalité entre le nombre de drones et le nombre de lignes. (b) Cas de non-proportionnalité entre le nombre de drones et le nombre de lignes.

FIGURE 5.12 – Illustration des parcours de zone en fonction de la hauteur de celle-ci, du nombre de drones et de D_{ref} . Le premier point-cible de chaque appareil est représenté par une croix pleine, les autres par une croix vide.

Lorsque le nombre de lignes de largeur D_{ref} n'est pas proportionnel au nombre de drones, celles situées à une extrémité de la zone sont surveillées plus souvent que les autres, comme l'illustre la figure 5.12(b) (p. 76). Cette méthode n'est pas optimale, mais elle permet de maintenir la connexité de l'essaim.

Durant la surveillance, les drones sont attirés vers leur propre point-cible et sont soumis aux forces d'attraction-répulsion entre eux.

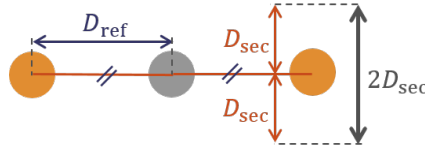
5.3.4 Mise en formation compacte

Quand les drones reçoivent une alerte (par un drone de haute altitude par exemple) signifiant qu'une cible se trouve probablement dans l'autre zone d'intérêt, ils doivent se positionner en formation compacte à l'entrée du tunnel avant de le traverser. Or, pour que les drones établissent une formation stable, il faut qu'un maximum d'entre eux soient espacés de D_{ref} . Dès cette étape de mise en formation, il faut donc diminuer la distance D_{ref} et ainsi permettre l'établissement d'une formation compacte.

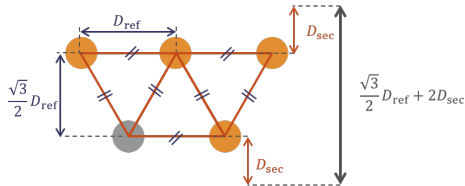
Pour cette configuration, on recherche la 2-connexité de l'essaim (et donc l'arête 2-connexité), permettant de maintenir des chemins de communication entre tous les drones même si l'un d'entre eux entre en collision et disparaît. De plus, comme on cherche à protéger le seul drone qui peut remplir la mission d'identification, dès que la largeur du tunnel le permet, ce drone se trouve proche du centre de l'essaim (comme le détaille le paragraphe suivant).

La formation compacte est d'autant plus stable que le nombre de drones

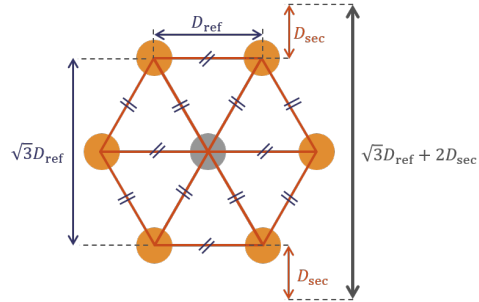
espacés de D_{ref} est grand. Une telle contrainte fait tendre la forme de l'essaim vers des structures en triangles équilatéraux. Par ailleurs, comme exposé dans la section 5.2.4, la force d'évitement des obstacles est calculée à partir de la distance de sécurité des drones D_{sec} . Les formations compactes mises en place pour la traversée du tunnel dépendent donc également de cette distance. Enfin, il est primordial que le drone permettant l'identification ne soit pas détruit à cause d'une collision, donc on cherche à le maintenir au centre de l'essaim. En fonction de la largeur du tunnel, plusieurs formations sont donc envisagées, comme le montre la figure 5.13. Pour établir une formation compacte donnée, chaque drone calcule un point cible, dépendant de la configuration sélectionnée et de son identifiant, à proximité de l'entrée du tunnel.



(a) Configuration en ligne pour les passages les plus étroits.



(b) Configuration en quinconce sur 2 niveaux améliorant la protection du drone avec identification par rapport à la formation 5.13(a).



(c) Configuration en quinconce sur 3 niveaux améliorant la protection du drone avec identification par rapport à la formation 5.13(b).

FIGURE 5.13 – Illustration des différents types de formations compactes en fonction de la largeur du passage. Le drone permettant l'identification est représenté en gris, les autres en orange.

Si le tunnel a une largeur inférieure à $2D_{\text{sec}}$, les drones ne peuvent pas le traverser en évitant les collisions, ils n'y entrent donc pas.

Un tunnel étroit (largeur comprise entre $2D_{\text{sec}}$ et $6D_{\text{sec}} + \frac{\sqrt{3}}{2}D_{\text{ref}}$) ne permet qu'une seule configuration, celle où les drones sont alignés, comme représenté en figure 5.13(a). On voit alors que le drone permettant l'identification n'est ni en première ni en dernière position dans l'essaim, ce qui limite le risque de perte. Néanmoins, il n'est pas protégé sur les côtés. On notera que le changement

de configuration se fait lorsque la distance entre un drone et le mur le plus proche est de $3D_{\text{sec}}$ (et pas seulement D_{sec}). Ceci est dû au fait que l'influence d'un obstacle est spatialement étendue à $3D_{\text{sec}}$ (équation 5.4, p. 69) et on veut maintenir une configuration stable.

Dans le cas de tunnels plus larges, une formation en quinconce est choisie, ajoutant à la configuration précédente la protection du drone d'intérêt sur un côté. Une formation en quinconce sur deux niveaux est établie pour des largeurs de tunnels comprises entre $6D_{\text{sec}} + \frac{\sqrt{3}}{2}D_{\text{ref}}$ et $6D_{\text{sec}} + \sqrt{3}D_{\text{ref}}$ (voir figure 5.13(b)).

Dans les tunnels plus larges, une configuration sur trois niveaux est adoptée (voir figure 5.13(c)). Cette dernière formation place le drone d'intérêt au centre de l'essaim.

5.3.5 Traversée du tunnel

Durant cette étape, les drones doivent traverser un environnement inconnu et contraint afin de rejoindre la deuxième zone d'intérêt. On fait l'hypothèse qu'ils peuvent échanger des messages dans leur portée de communication, même si un obstacle se trouve entre eux.

Pendant la traversée du tunnel, les forces de répulsion dues aux obstacles s'ajoutent aux forces d'attraction et de répulsion des autres drones, et aux forces d'attraction vers une cible. Comme les drones ne connaissent pas la forme du tunnel, ils calculent à chaque itération un point-cible temporaire situé dans la direction de la seconde zone d'intérêt, dans l'alignement de l'entrée du tunnel (illustré en figure 5.16).

Deux cas particuliers ont été traités afin de réduire la durée de traversée du tunnel.

D'une part, si un obstacle est situé entre un drone v et son point-cible, alors l'engin ne prend pas en compte l'attraction vers sa cible. Sans cet ajustement, un drone peut rester dans un équilibre local où la répulsion de l'obstacle et l'attraction vers la cible se compensent. Grâce à ce traitement, les forces d'attraction vers les voisins ont un poids plus important, permettant au drone bloqué de continuer la traversée en suivant les autres drones.

Le second cas particulier est très similaire. Si un drone u et son voisin v sont séparés par un obstacle, alors u ne prendra pas en compte l'attraction éventuelle vers v . En effet, comme abordé dans le paragraphe précédent, un drone bloqué (ici v) peut sortir grâce à l'attraction vers ses voisins : il faut donc que les voisins en question (ici u) continuent leur traversée.

Ces deux cas particuliers sont illustrés sur la figure 5.14.

La figure 5.15 est extraite d'une simulation de ce modèle effectuée dans JBotSim. Elle montre l'essaim au cours de la traversée d'un tunnel contenant des obstacles. On remarque que tant que la géométrie des obstacles le permet, les drones maintiennent une configuration sur deux niveaux.

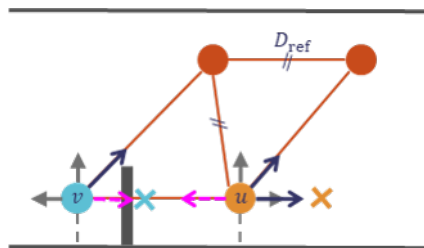


FIGURE 5.14 – Illustration des forces appliquées sur deux drones, u et v , chacun dans un cas particulier. Les forces non prises en compte sont représentées en pointillés roses.

Une fois sortis du tunnel, les drones répètent l'étape de surveillance dans la seconde zone d'intérêt.

5.3.6 Synthèse

Cette section a présenté les modèles de mobilité utilisés tout au long de la mission, décomposée en plusieurs étapes. Au cours de celles-ci, les objectifs et contraintes diffèrent, mais les modèles de mobilité reposent tous sur le même principe : l'association de forces virtuelles.

Dans la phase de surveillance, nous avons vu qu'un modèle déterministe était utilisé, afin de maintenir la connexité du graphe et de permettre l'évaluation du temps de parcours total de la zone. À la réception d'un signal envoyé par le drone de haute altitude, les drones de l'essaim calculent leur cible pour la mise en formation compacte. Une fois que tous les drones ont atteint leur cible, l'essaim traverse le tunnel en détectant et en évitant les obstacles, tout en maintenant les liens de communication. Dans la seconde zone d'intérêt, le même modèle de mobilité pour la surveillance est utilisé.

5.4 Simulations et résultats

5.4.1 Environnements et paramètres de simulation

Une série de simulations a été menée afin de valider les modèles de mobilité définis en section 5.3. Comme détaillé en section 2.2, le simulateur utilisé pour cette étude est JBotSim (simulateur open source créé pour du prototypage rapide de systèmes distribués).

Pour rappel, nous avons choisi de simuler des mini-drones d'enveloppe circulaire de 40 cm, embarquant un module de communication ainsi qu'un capteur permettant la détection d'obstacles et la surveillance. Nous choisissons ici de simuler un module de communication XBee, dont la portée dépend notamment du type d'environnement dans lequel il est utilisé. Dans notre cas, l'environne-

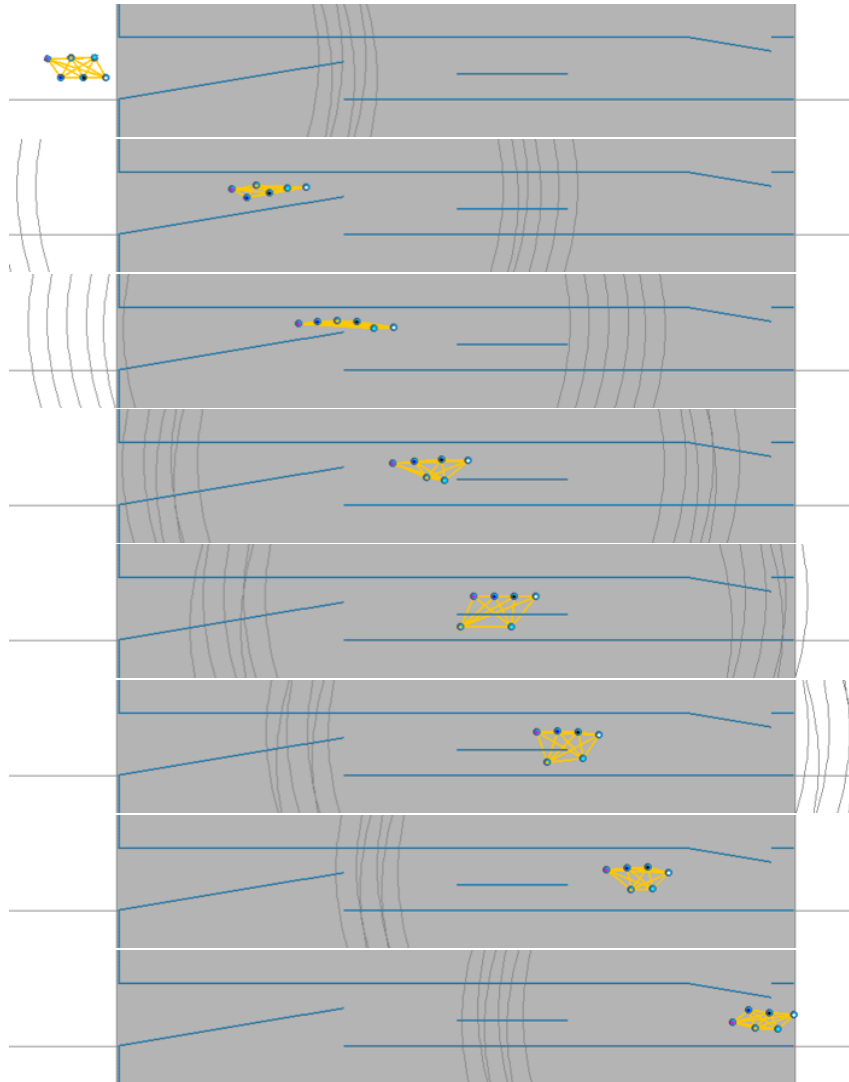


FIGURE 5.15 – Essaim de 6 drones traversant un tunnel contenant des obstacles. Les drones sont représentés par des disques et les liens de communication par des segments jaunes. La portée de communication de chaque drone est illustrée par un cercle (la portée est grande par rapport aux dimensions de l’illustration, on ne voit donc que des arcs).

ment contient des obstacles, et nous choisissons donc de simuler la performance d’un XBee en environnement urbain, soit 40 m [XBee, 2011].

Comme détaillé en section 3.6 (p. 34), les caméras stéréo sont un outil efficace de détection d’obstacles. Nous supposons que les engins embarquent plusieurs caméra stéréo de portée 24 m et qu’un drone dispose d’informations à 360°.

La première zone d’intérêt mesure 360 m sur 300 m. La surveillance de la seconde zone n’est pas présentée ici car le modèle de mobilité est identique à

celui utilisé dans la première.

Durant la mission, on distingue deux grandes étapes, où les besoins en terme de distances inter-drones diffèrent. Au cours de la surveillance, les drones doivent être suffisamment éloignés pour couvrir la surface le plus rapidement possible. Le facteur k utilisé dans la définition de la distance de référence inter-drones D_{ref} (équation 5.2, p. 66) peut donc être grand. Plusieurs valeurs sont testées ici : 0,70 ; 0,75 ; 0,85 et 0,95. Ces valeurs permettent un faible recouvrement des zones surveillées par deux drones voisins ainsi qu'un bon maintien des liens de communication. Les paramètres utilisés dans ces simulations sont résumés dans le tableau 5.1.

Paramètre	Valeur (m)							
p_{comm}	40							
p_{capteur}	24							
d_{drone}	0,4							
D_{sec}	0,6							
	Formation compacte				Surveillance			
k	0,025	0,035	0,05	0,70	0,75	0,85	0,95	
D_{ref}	1	1,4	2	28	30	34	38	
d_{max}	0,1	0,3	0,6	6,0	5,0	3,0	1,0	

TABLE 5.1 – Paramètres de simulation.

Une représentation des zones d'intérêt, du tunnel et des drones, extraite de JBotSim est montrée en figure 5.16. Les simulations ont été faites avec 1 drone seul ainsi qu'avec des essaims de 2 à 6 drones, avec 5 configurations de tunnels représentant des environnements contraints variés (voir figure 5.17, p. 82). Pour chaque jeu de paramètres, 30 simulations ont été lancées. Les zones d'intérêts ne diffèrent pas selon les simulations, et la forme du tunnel n'a pas d'impact sur la surveillance de la première zone d'intérêt. Pour chaque nombre de drones, il y a donc 5 simulations de la surveillance de la première zone d'intérêt.

Il est à noter qu'un tunnel de largeur $2D_{\text{sec}}$, plus étroit que le tunnel a) a été testé ; $2D_{\text{sec}}$ est la distance minimale entre deux obstacles dans laquelle un drone peut passer. Dans ce cas, les drones ne pouvaient pas y entrer car la répulsion due aux obstacles en face d'eux était supérieure à l'attraction vers leur cible.

5.4.2 Métriques et résultats

Afin d'évaluer les performances du modèle de mobilité développé, nous utilisons des métriques quantitatives et qualitatives pour chaque étape du scénario. Pour chacune d'elle, le résultat idéal est donné, (appelé « optimal » ci-dessous) bien qu'il ne soit pas toujours atteignable.

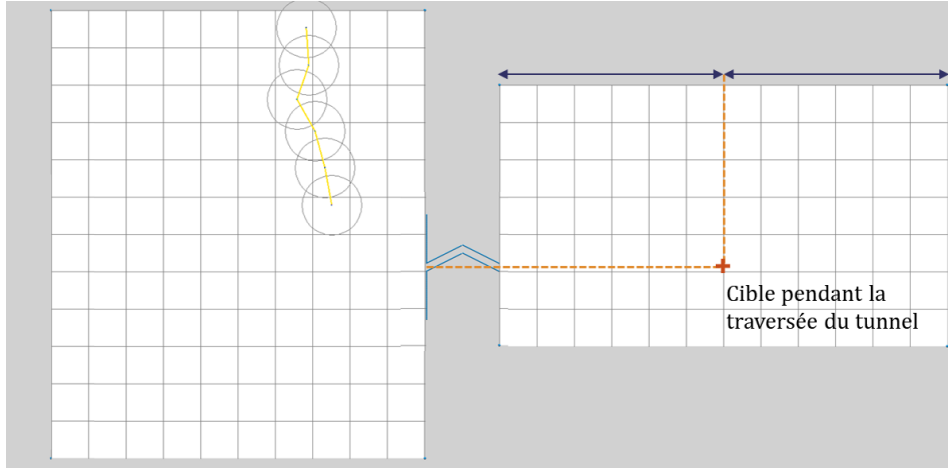


FIGURE 5.16 – Un des environnements de simulation, illustrant la position du point-cible utilisé pendant la traversée du tunnel.

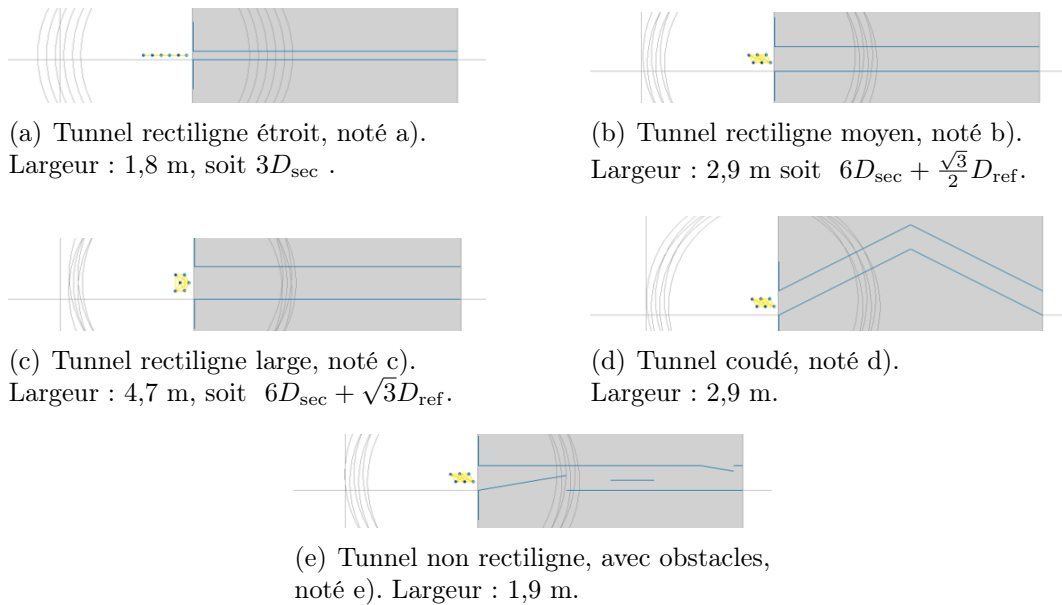


FIGURE 5.17 – Illustration des 5 configurations de tunnels utilisées pour valider le modèle de mobilité.

Mission globale

La contrainte principale de la mission est la sauvegarde des drones. Pour cela, la distance entre un drone et un autre objet doit rester supérieure à la distance de sécurité D_{sec} définie en section 5.2.4. Les métriques associées sont donc les suivantes :

1. **Distance minimale entre deux drones**
→ Optimal : supérieure à la distance de sécurité
2. **Distance minimale entre un drone et un obstacle fixe**
→ Optimal : supérieure à la distance de sécurité

Il n'y a eu aucune collision dans les simulations réalisées. La distance minimale entre deux drones a été de $2,37D_{\text{sec}}$, et celle entre un drone et un obstacle fixe de $1,43D_{\text{sec}}$. Les distances minimales pour chaque jeu de simulation sont données en annexe 7.5 (p. 145).

Surveillance

Dans le but d'avoir la meilleure vision de la zone d'intérêt dans laquelle l'essaim évolue, chaque drone doit avoir à chaque instant la vision de l'ensemble de la zone la plus récente possible. Pour cela, les drones doivent se déployer tout en maintenant les liens de communication entre eux. Les métriques associées sont les suivantes :

3. **Durée de couverture de zone** : durée nécessaire à l'essaim pour couvrir entièrement l'ensemble de la zone
→ Optimal : Inversement proportionnelle au nombre de drones (pour une zone donnée) : $\frac{\text{temps de parcours d'un drone}}{\text{nombre de drones}}$
4. **Pourcentage de temps de connexion** :
→ Optimal : 100%
5. **Nombre de déconnexions** : nombre de liens de communication perdus
→ Optimal : 0
6. **Nombre de reconnexions** : nombre de liens de communication créés après le début de la surveillance
→ Optimal : autant que de déconnexions

Comme attendu, les simulations ont montré que le temps de parcours de zone diminue avec le nombre de drones (comme illustré en figure 5.18 ; données brutes en annexe 7.5).

On constate un écart conséquent entre le nombre d'itérations nécessaire au parcours complet de la zone entre $k = 0,95$ et les valeurs inférieures. Le grand nombre d'itérations pour $k = 0,95$ est du à la faible distance maximale

autorisée dans ce cas (1 m). En effet, comme mis en relief dans le tableau 5.1(p. 81), plus la valeur de k est grande, plus la distance maximale autorisée est faible. Ainsi, il est intéressant de ne pas choisir k trop proche de 1.

Au premier abord il paraît donc souhaitable de choisir k de sorte à maximiser la distance à parcourir, c'est-à-dire $k = 0,51$ (cf. section 5.2.7, p. 71). Dans ce cas, la distante D_{ref} est quasiment égale à la portée de communication.

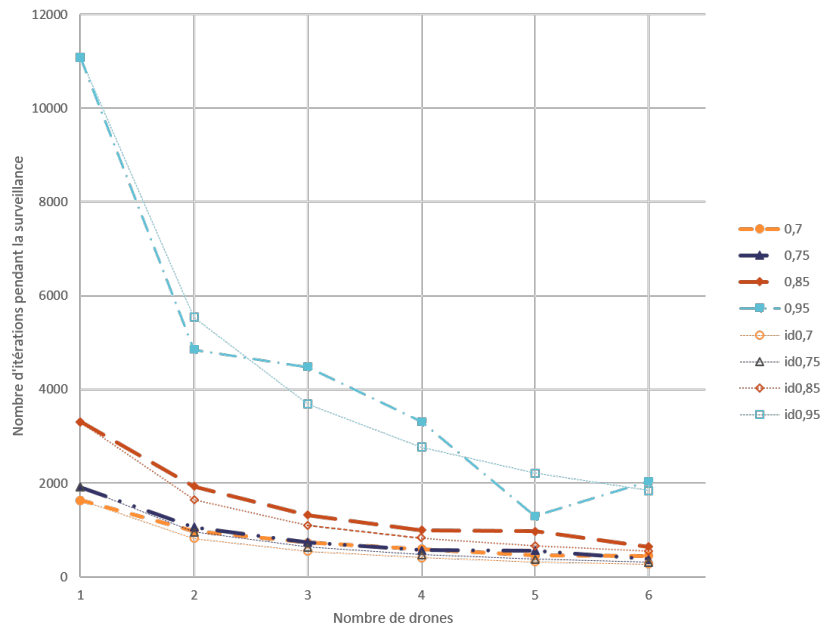
Or, pendant la surveillance, les forces attractives entre les drones interviennent dans le calcul de déplacement et elles sont d'autant plus importantes que la distance entre les drones connectés est grande. C'est pourquoi lorsque k est trop petit ($k < 0,7$ dans ce cas d'étude), des drones qui ne devraient pas être connectés dans la configuration de surveillance se connectent et ne peuvent pas rompre ce lien. Ils ne peuvent donc pas atteindre leur point-cible et l'essaim reste dans une configuration d'équilibre, immobile.

Pour plus de clarté, la figure 5.18(b) n'affiche pas les valeurs associées à $k = 0,95$. Sur ce graphique, on constate que les résultats obtenus sont proches de l'idéal sans toutefois l'atteindre.

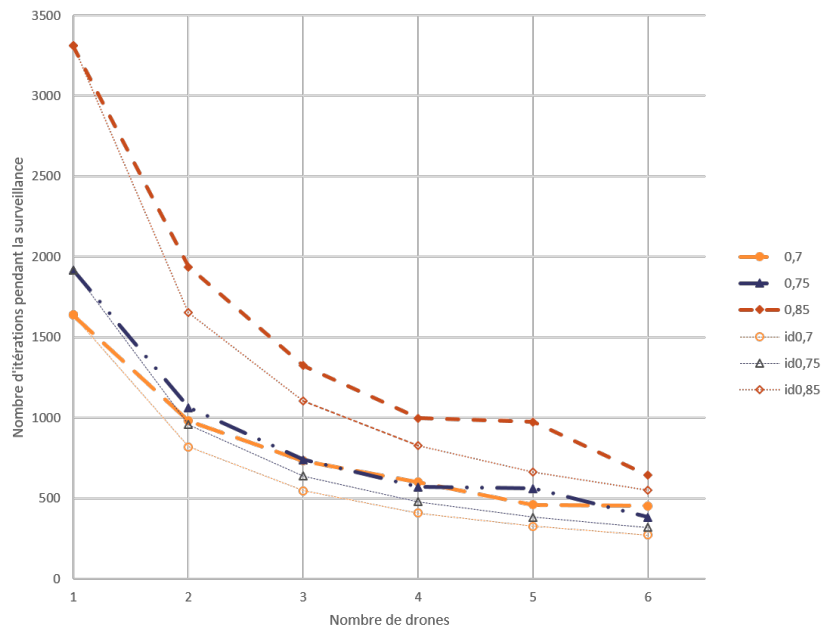
Comme présenté dans la section 5.3.3 (p. 75), les drones parcourent la zone en boustrophédon tout en maintenant une formation en ligne. Le parcours dépend de la distance de référence inter-drones (donc du réel k). Par exemple, lorsque $k = 0,75$ la zone est découpée en 12 lignes de largeur D_{ref} (voir figure 5.12, p. 76). Ainsi, lorsque l'essaim est composé de 4 drones, il doit traverser 3 fois la zone pour la couvrir ; quand la flotte compte 5 aéronefs également. C'est pourquoi le temps de parcours de la zone par des essaims de 4 et 5 drones, avec $k = 0,75$ est quasiment identique. Le même phénomène apparaît avec la décomposition en 11 lignes pour $k = 0,85$.

Dans le cas d'un nombre de drones proportionnel au nombre de lignes, les résultats n'atteignent pas les valeurs idéales car chaque drone ayant atteint son point-cible attend que ses voisins aient fait de même avant de poursuivre le parcours.

Les métriques 4. 5. et 6. concernent le maintien de connexité de l'essaim. Il n'y a qu'un cas où des déconnexions sont apparues : un essaim de 6 drones avec $k = 0,95$; l'essaim n'est resté connecté que 83,8% du temps de la surveillance car un lien a été brisé. La déconnexion se produit lorsque l'essaim a traversé une fois la zone en largeur et doit changer de direction. La faible distance parcourue par itération peut en effet entraîner des délais d'arrivée importants entre les engins. La partie de la flotte arrivée « en avance » a modifié sa direction avant que les autres engins n'aient atteint leur cible. On en déduit que un k trop grand ne permet pas un bon maintien des liens de communication lorsqu'au moins 6 drones sont considérés.



(a) Avec k valant 0,70 ; 0,75 ; 0,85 et 0,95.



(b) Avec k valant 0,70 ; 0,75 et 0,85.

FIGURE 5.18 – Nombre moyen d’itérations nécessaires au parcours de la première zone d’intérêt en fonction du nombre de drones. Le temps de parcours idéal pour n drones, défini comme $\frac{\text{Nb itérations pour 1 drone}}{n}$, est tracé en pointillés pour chaque valeur de k .

Traversée du tunnel

Durant la traversée du tunnel permettant de rejoindre la seconde zone d'intérêt, les drones doivent éviter les collisions entre eux et avec les obstacles fixes. Dans les simulations effectuées, l'alerte a été envoyée uniquement à un drone situé à l'extrémité de l'essaim ; les drones l'ont ensuite transmise entre eux, de proche en proche.

Une métrique qualitative et une quantitative sont utilisées pour évaluer la qualité de la traversée du tunnel :

7. **Réception de l'alerte** : nombre de drones ayant reçu l'alerte ordonnant la mise en formation compacte (par une entité extérieure ou un autre engin de l'essaim).
 - Optimal : tous les drones de l'essaim.
8. **Durée de traversée du tunnel** : durée écoulée entre l'entrée et la sortie du dernier drone (potentiellement deux engins différents).
 - Optimal : temps de trajet d'un drone seul par le plus court chemin, en évitant les obstacles inconnus.

Pour toutes les simulations, tous les drones de l'essaim ont reçu l'ordre de changer de zone avec le délai minimal (c.-à-d. le voisin à n sauts de celui ayant reçu l'alerte la recevait après n itérations).

La figure 5.19 compare, sur un même graphique, le nombre d'itérations en fonction du nombre de drones, pour différentes configurations de tunnels, pour différentes valeurs de k (les graphiques associés aux autres valeur de k , aux tendances similaires, sont donnés en annexe 7.5, p.147). Les notations des noms des tunnels sont donnés en figure 5.17 (p. 82).

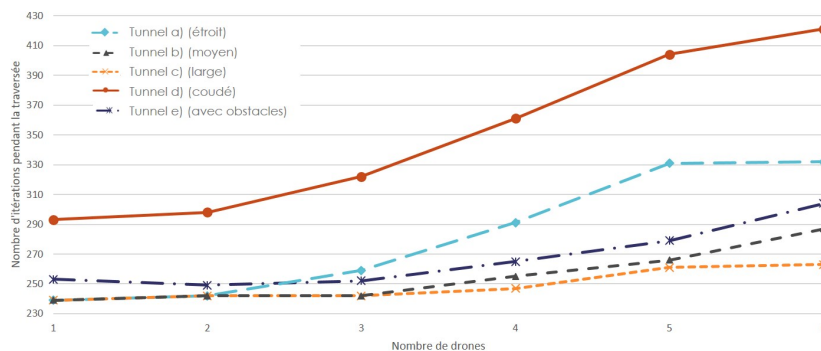


FIGURE 5.19 – Durée moyenne de traversée des tunnels pour $k = 0,05$.

Au cours de la traversée du tunnel, les drones sont attirés vers leur point-cible tout en subissant la répulsion des obstacles fixes ainsi que les attractions-répulsions liées à leurs voisins. La distance de référence inter-drones dans le tunnel étant très inférieure à la portée du module de communication, tous

les drones sont des voisins directs (à 1 saut). Comme les drones tentent de maintenir cette distance avec tous leurs voisins, ils subissent autant de forces qu'ils ont de voisins (en dehors des deux cas particuliers présentés en section 5.3.5). Pour cette raison, plus il y a de drones dans l'essaim, plus chaque engin sera soumis à un grand nombre de forces, potentiellement dans une autre direction que son point-cible. C'est pourquoi les temps de traversée de tunnel augmentent globalement avec le nombre de drones.

Dans le cas du tunnel noté a), le plus étroit, les drones sont les uns derrière les autres. C'est donc le cas où ils subissent le plus de forces dans la direction opposée à leur cible. Plus un drone en formation en ligne est proche de l'entrée du tunnel (c.-à-d. à l'avant de la formation), plus le nombre de drones derrière lui est grand, donc plus les forces d'attraction opposées à la cible sont grandes. Les drones les plus en avant de l'essaim ont donc une vitesse réduite. Le tunnel étant trop étroit pour autoriser les dépassements, les drones suivants vont donc également ralentir. C'est pourquoi la durée de traversée de ce tunnel est très dépendante du nombre de drones. De plus, la symétrie du tunnel fait que les répulsions des murs situés à droite et à gauche de chaque engin se compensent tant qu'ils restent au milieu du passage.

Cependant, on remarque que la durée de traversée d'un essaim de 6 drones est quasiment égale à celui d'un essaim de 5 drones. Dans ces deux cas, l'essaim subit la contrainte inhérente à l'étroitesse du passage avant même d'entrer dans le tunnel. Les drones les plus en tête de l'essaim étant les plus proches des obstacles, ils vont évoluer à allure réduite jusqu'à subir la répulsion des drones qui s'approchent derrière eux. Néanmoins, avec un essaim de 6 drones, la formation en ligne n'est pas maintenue jusqu'à l'entrée de tous les engins dans le tunnel (comme illustré en figure 5.20, p. 88). Cette déformation retarde considérablement l'entrée dans le tunnel du dernier drone de la formation. Or le temps de traversée est défini par la durée écoulée entre l'instant où le dernier drone entre dans le tunnel et l'instant où le dernier drone en sort. Ainsi, bien que la durée de traversée du premier drone de l'essaim soit plus courte dans un essaim de 5 drones que dans un de 6, la durée de traversée de l'essaim de 6 drones est similaire.

Dans le cas des tunnels rectilignes b) (moyen) et c) (large), on constate sur la figure 5.19 une augmentation globale de la durée avec le nombre de drones. L'augmentation est plus forte dans le tunnel b) car dans la formation en quinconce sur deux niveaux, établie dans le tunnel b), il y a un plus grand nombre de drones alignés dans l'axe du tunnel (donc plus de forces s'opposant à la direction de la cible) que dans la formation sur trois niveaux.

La traversée du tunnel d), coudé, est la plus longue de toutes. En effet, les drones ne peuvent quasiment jamais aller dans la direction de leur cible. Une force de répulsion du mur en face d'eux s'oppose donc systématiquement à l'attraction de la cible. Ainsi ils évoluent à vitesse réduite. L'augmentation de durée avec le nombre de drones s'explique, comme pour les autres tunnels,

par l'attraction des voisins derrière eux.

Enfin, la durée de traversée du tunnel e), contenant des obstacles, suit globalement celle de la traversée du tunnel b), avec des valeurs légèrement plus élevées dues aux reconfigurations nécessaires à l'évitement des obstacles.

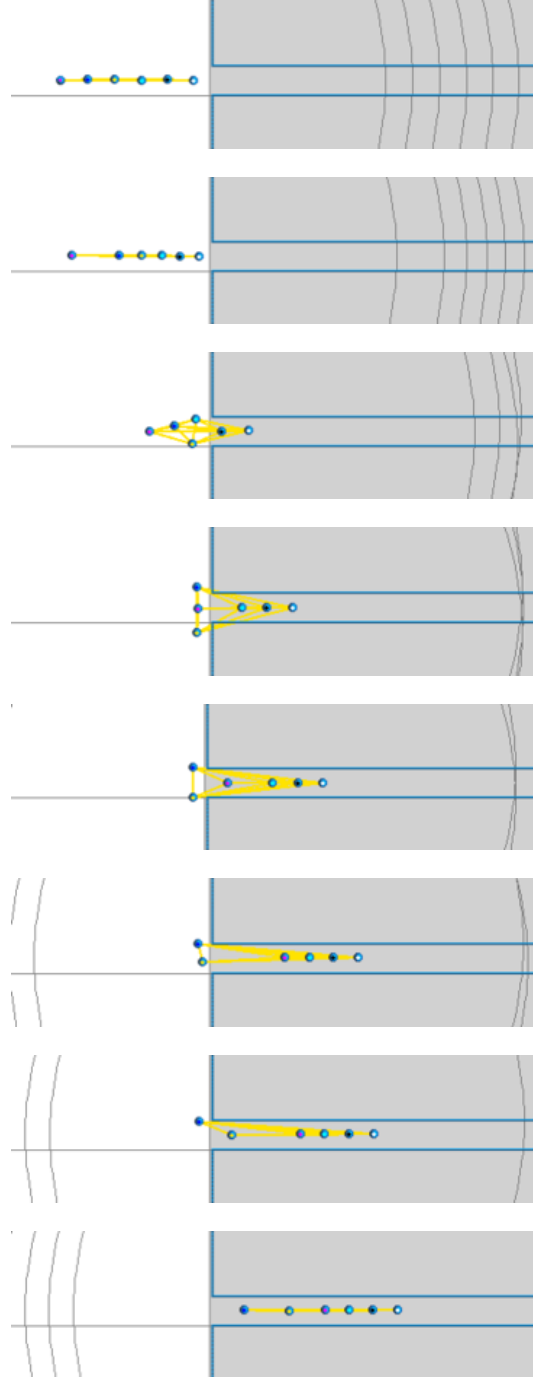


FIGURE 5.20 – Essaim de 6 drones traversant le tunnel étroit noté a).

La figure 5.21 représente les durées de traversée du tunnel b) en fonction du nombre de drones, pour chaque valeur de k testée (les graphiques associés aux autres tunnels, aux tendances similaires, sont donnés en annexe 7.5, p.147).

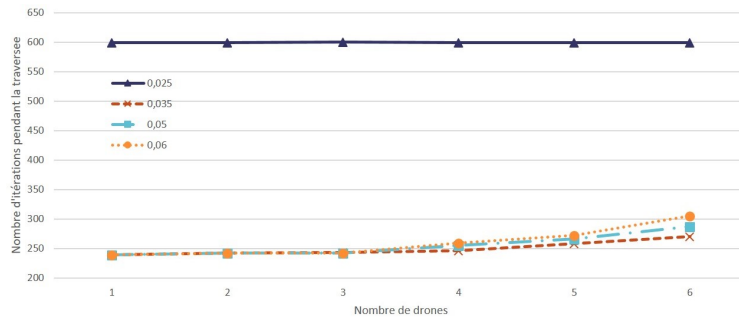


FIGURE 5.21 – Nombre moyen d’itérations nécessaires à la traversée le tunnel b), pour toutes les valeurs de k testées.

On remarque que la durée de traversée du tunnel est nettement plus élevée pour $k = 0,025$ que pour les autres valeurs de k . Durant la phase de surveillance, le plus grand nombre d’itérations était obtenu pour $k = 0,95$ en raison de la faible distance maximale autorisée par itération. Dans le cas du tunnel, le même phénomène apparaît : plus k est petit, plus la distance maximale est faible. Pour les autres valeurs de k , le nombre d’itérations est quasiment constant pour des essaims de 1 à 3 drones et augmente légèrement au delà. Comme noté précédemment, plus le nombre de drones est grand, plus les forces d’attraction-répulsion entre eux sont nombreuses, notamment dans la direction opposée à la cible, ce qui augmente le temps de parcours.

5.5 Résumé

Dans ce chapitre, nous avons introduit un modèle de mobilité permettant à un essaim autonome de drones collaboratifs de remplir une mission. Celle-ci a pour objectif la surveillance successive de deux zones d’intérêts séparées par un environnement fortement contraint (tunnel). Nous nous sommes concentrés sur la traversée du passage pouvant être étroit et contenir des obstacles, inconnus des drones.

Les engins considérés sont des multiroteurs embarquant un capteur permettant de détecter les obstacles et d’opérer une mission de surveillance, ainsi qu’un module de communication permettant la collaboration interplateforme. Un unique drone de l’essaim a la capacité d’identifier une cible, il faut donc minimiser les risques de collisions qu’il encourt.

Dans cette étude, les échanges d’information entre les engins ne sont pas détaillés, mais des contraintes sur le maintien de liens de communication sont posées. Une analogie est faite entre l’essaim de drones communicants et les

graphes dynamiques. Les contraintes de communication sont donc décrites sous la forme de maintien de connexité du graphe associé à l'essaim (section 5.1.2).

Le modèle de mobilité est basé sur le principe des forces virtuelles et considère trois types de forces (section 5.2) :

- L'attraction-répulsion entre drones, évitant les collisions inter-plateformes tout en favorisant le maintien des liens de communication ;
- L'attraction vers un point-cible, utilisée pour qu'un drone se déplace dans le but de remplir la mission ;
- La répulsion due aux obstacles fixes, permettant l'évitement de collisions.

À partir de ces trois types de force, l'ensemble du modèle de mobilité peut être défini en fonction des besoins différents de chaque étape du scénario (section 5.3).

Le simulateur JBotSim a été utilisé afin d'évaluer les performances de ce modèle de mobilité, dans différents environnements (section 5.4.1). Plusieurs métriques ont été définies et les résultats associés sont conformes aux résultats attendus : il n'y a pas eu de collision, les liens de communication ont été maintenus et les temps d'exécution de mission sont satisfaisants (section 5.4.2).

Dans ce chapitre nous avons présenté un modèle comportemental simplifié pour un essaim collaboratif de drones autonomes. Cette étude a permis de réaliser une analyse initiale des contraintes mises en jeu dans le contexte d'une mission menée par un tel système. Pour aboutir à un modèle réaliste, plusieurs hypothèses de simplification doivent être supprimées et des contraintes doivent également être ajoutées.

Pour augmenter le réalisme des simulations, on souhaiterait notamment aborder les points suivants :

- L'ensemble de drones, donc de capteurs embarqués et donc d'informations sur l'environnement doit profiter à l'ensemble des plateformes dans le but d'améliorer les performances de chaque individu, au profit de la mission globale.
- Pour minimiser le coût calculatoire, un drone ne découvrant pas d'obstacle doit pouvoir suivre son chemin sans nouveau calcul.
- Chaque engin doit pouvoir évoluer dans des environnements complexes, même sans voisins. Pour cela, il aura besoin de prévoir ses mouvements plusieurs itérations en avance.

Les besoins liés au calcul d'un chemin et pas seulement d'un mouvement ne peuvent pas être remplis par le modèle présenté ci-dessus. En effet, l'application des forces virtuelles se fait itération par itération. Pour proposer un algorithme fournissant un chemin et pas seulement un mouvement, il faut donc évaluer les forces qui seraient appliquées dans tout (ou partie) de l'environnement. Cela forme un champ de potentiel. C'est sur ce principe que sera créé le second modèle de mobilité.

Chapitre 6

Calcul de chemin pour un drone autonome en environnement complexe

Sommaire

6.1	Contexte	92
6.1.1	Contexte opérationnel	92
6.1.2	Hypothèses	92
6.2	Préambule au calcul de chemin	93
6.2.1	Principe	93
6.2.2	Outils	93
6.3	Méthode de calcul de chemin	98
6.3.1	Définition des champs de potentiels discrets	98
6.3.2	Calcul d'un chemin à partir des champs de potentiels discrets	101
6.3.3	Avantages par rapport à un APF classique	108
6.4	Paramètres de discrétisation et de simulation	109
6.4.1	Discrétisation de l'environnement	109
6.4.2	Distance maximale à parcourir en une itération	109
6.4.3	Environnement simulé	109
6.4.4	Paramètres de simulation	110
6.5	Résultats et évaluations	112
6.5.1	Distance parcourue en fonction de la taille de maille	113
6.5.2	Nombre d'itérations nécessaires en fonction de la taille de maille	113
6.5.3	Limitations du modèle	120
6.5.4	Conclusion sur la performance du modèle	122
6.6	Résumé	123

Bien que l'objectif de cette thèse soit de proposer un calcul de chemin pour les drones d'un essaim, ce chapitre présente notre méthode de calcul de chemin adaptée à un drone unique, lui permettant d'évoluer en environnement complexe. L'adaptation aux calculs par les drones d'un essaim fait l'objet du chapitre suivant (7, p. 124). Cette segmentation a été faite pour deux raisons : d'une part, chaque drone de l'essaim doit être autonome donc le modèle de mobilité doit être utilisable par un drone seul. D'autre part, il est plus clair de présenter le modèle pour un drone, puis de décrire son adaptation pour plusieurs plateformes.

Une première partie expose le contexte de la mission considérée et les besoins opérationnels associés 6.1 (p. 92). Le principe du modèle que nous proposons et les outils qu'il nécessite sont présentés en section 6.3 (p. 98). Notre méthode de calcul de chemin est détaillée en section 6.3 (p. 98). La section suivante (6.4, p. 109) a pour objet les paramètres du modèle et de simulation. Enfin, nous validons notre méthode de calcul de chemin par des simulations réalisées dans un environnement complexe (cf. section 6.5, p. 112).

6.1 Contexte

6.1.1 Contexte opérationnel

Nous considérons une zone sensible dans laquelle un système de surveillance doit pouvoir être déployé rapidement. Plus précisément, la zone d'intérêt étudiée dans ces travaux est une simulation de la centrale nucléaire du Blayais. Cette zone est représentative des milieux dans lesquels notre système pourrait être déployé et contient de nombreux bâtiments de formes et dimensions diverses.

Compte tenu des performances d'un drone multirotor, notamment concernant sa réactivité, c'est un outil utile dans le cadre d'interventions rapides. Dans le but que notre système puisse être déployé dans des environnements variés et sans informations préalables, on considère que le drone découvre son environnement uniquement à partir des capteurs qu'il embarque.

Pour qu'un drone puisse mener une mission de surveillance dans un environnement inconnu, il doit être en mesure d'arpenter la zone d'intérêt. Nous nous attachons ici à la création d'un modèle de mobilité permettant à un multirotor de l'essaim de rejoindre n'importe quel point libre (c.-à-d. ne contenant pas d'obstacle) de la zone d'intérêt depuis sa localisation courante.

6.1.2 Hypothèses

Nous nous intéressons ici au calcul de chemins et pas de trajectoires (cf. définitions en section 4.1.1, p. 40). On suppose que les drones sont des véhicules

holonomes et les contraintes cinématiques ne sont pas considérées ici.

Dans cette étude nous considérons que l'environnement dans lequel se déplace le drone ne contient que des obstacles fixes, qui ne sont pas connus *a priori* ; la plateforme évolue à une altitude suffisante pour éviter les véhicules terrestres, mais les bâtiments sont à prendre en compte, et représente le seul système aérien de la zone d'intérêt. Le drone embarque un capteur permettant la détection d'obstacles.

6.2 Préambule au calcul de chemin

6.2.1 Principe

Dans le concept original des champs de potentiels, [Khatib \[1986\]](#) propose de calculer les mouvements d'un robot à partir d'un champ de forces dérivant de potentiels. Plus précisément, chaque déplacement du robot résulte du bilan des forces s'appliquant à lui à un instant et en un lieu donné (la section [4.3.4](#), p. [51](#) présente en détail le calcul de chemin à partir de champs de potentiels). Notre approche ne repose pas sur l'utilisation de forces, mais s'appuie sur le principe d'un champ dans lequel le drone calcule ses mouvements, dans un environnement discrétisé : à chaque itération, la plateforme se dirigera vers une maille dont la valeur du champ sera inférieure à celle qu'elle occupe. Compte tenu de la similarité de principe avec la méthode de Khatib, ainsi que de la dépendance du champ avec la discrétisation de l'environnement choisie, nous désignons notre méthode comme basée sur des « champs de potentiel discrets ».

Les calculs de chemin sont effectués dans un espace discrétisé en mailles parallélépipédiques à base carrée. Les déplacements des plateformes, en revanche, sont simulés dans un environnement continu. Nous considérons que la distance de sécurité à maintenir autour d'une plateforme pour éviter les collisions est égale à la largeur d'une maille (le choix de la discrétisation est détaillé en section [6.4.1](#), p. [109](#)).

Le principe de notre méthode de calcul de chemin est exposé dans l'algorithme [2](#) (p. [94](#)).

6.2.2 Outils

Les informations à mémoriser pour chaque maille de l'environnement permettant le calcul de chemin sont : la présence ou l'absence d'obstacles fixes et les potentiels.

Ces différentes informations sont chacune stockées dans un tableau à 3 dimensions, dont chaque élément représente une maille de l'environnement discrétisé. Cette section introduit les notations utilisées ainsi que les trois structures de données majeures manipulées par l'algorithme.

Algorithme 2 Principe de notre méthode.

tant que le drone n'a pas atteint son point-cible **faire**
 Détecter les obstacles ;
 si de nouveaux obstacles ont été détectés **alors**
 Mettre à jour les informations d'environnement (tableau **obs**) ;
 Mettre à jour les champs de potentiels répulsifs (tableau **pot**) ;
 fin si
 si de nouveaux obstacles ont été détectés **ou** le drone détecte qu'il est dans une impasse **ou** le drone n'a connaissance d'aucun obstacle entre lui et son point-cible **alors**
 Calculer un nouveau chemin ;
 fin si
 Se déplacer en suivant le chemin calculé ;
fin tant que

Notations

L'environnement discrétisé est de dimension $a \times b \times c$ avec $(a, b, c) \in \mathbb{N}^* \times \mathbb{N}^* \times \mathbb{N}$, représentant respectivement la largeur, la longueur et la hauteur de l'environnement.

On utilise les notations suivantes, avec $(i, j, k) \in \llbracket 1, a \rrbracket \times \llbracket 1, b \rrbracket \times \llbracket 0, c \rrbracket$:

uav_{ijk} maille dans laquelle se trouve le drone courant, dont les indices sont notés uav_i, uav_j, uav_k ;

v_{ijk} maille contenant un voisin du drone courant ;

$cible_{ijk}$ maille contenant la cible du drone courant ;

o_{ijk} maille découverte contenant un obstacle ;

c_{obs} constante correspondant au potentiel répulsif lié à un obstacle.

On utilise une norme euclidienne discrétisée :

$$\|m_{ijk}\| = \sqrt{i^2 + j^2 + k^2}.$$

Tableau obs

Mémoriser uniquement la liste des obstacles détectés n'est pas suffisant car il est tout aussi important de connaître la présence que l'absence d'un obstacle dans une maille. Pour cette raison, nous utilisons un tableau de la dimension de l'environnement discrétisé $a \times b \times c$, noté **obs** dont chaque élément prend une des valeurs suivantes en fonction de la présence ou non d'obstacles dans la maille associée :

-1 pas d'information ;

0 ne contient pas d'obstacle et est sans danger ;

1 ne contient pas d'obstacle et est adjacente à une maille contenant un obstacle ;

2 contient un obstacle (au moins partiellement).

Ce tableau est mis à jour dès réception de nouvelles informations sur l'environnement.

Tableau pot

Pour un chemin donné, deux types d'informations sont valides en permanence : le point-cible à atteindre et les positions et formes des obstacles connus. Chacun de ces éléments est considéré sous forme de potentiels dans le calcul de chemin. Le tableau `pot`, stocke ces champs de potentiels discrets (dont les définitions sont données en section 6.3.1, p. 98).

Ce tableau est mis à jour dès que de nouvelles informations sur l'environnement ou sur la mission sont reçues.

Tableau avoid

Par opposition aux données valides sur le long terme, certaines informations ne sont valides que durant quelques itérations, comme l'anticipation de l'évitement des obstacles qui dépend de la position courante du drone¹. Les potentiels associés à l'anticipation d'évitement des obstacles sont définis en section 6.3.1 (p. 98), et sont enregistrés dans le tableau `avoid` qui est remis à jour à chaque nouveau calcul de chemin.

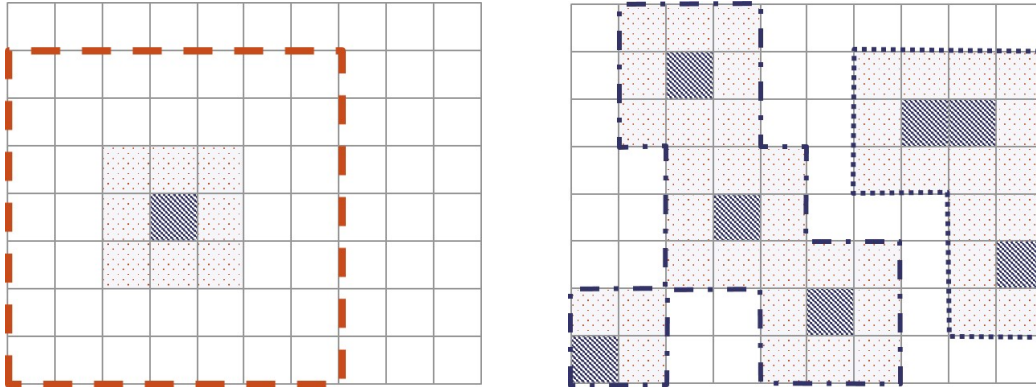
Définition d'un obstacle fixe

Le tableau `obs` contient les informations relatives à la présence d'obstacles maille par maille. Néanmoins, le calcul de chemin est plus efficace si les obstacles sont considérés dans leur globalité plutôt que si les mailles sont traitées indépendamment les unes des autres. Un obstacle est constitué d'un ensemble de mailles adjacentes interdites au drones ; c.-à-d. de valeur 1 ou 2 dans le tableau `obs`. La figure 6.1 (p. 96) illustre des cas d'obstacles constitués de mailles contenant des obstacles et de mailles leur étant adjacentes.

La représentation de l'environnement (taille et forme des obstacles) dépend de la finesse du maillage, ce qui impacte le calcul du chemin. En effet, une zone contenant deux obstacles physiquement séparés par un espace permettant au drone de passer peut être considérée comme une grande zone interdite par le drone, comme l'illustre la figure 6.2 (p. 96).

Certains traitements ne prendront pas en compte seulement l'obstacle mais son enveloppe. Celle-ci est le plus petit rectangle (pavé droit en 3D) contenant

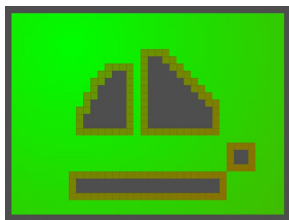
1. Lorsque plusieurs drones sont considérés, la position des drones voisins fait partie de ces informations à validité temporaire.



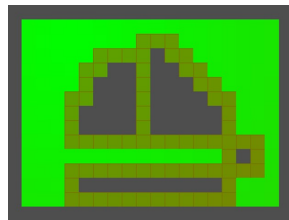
(a) Toutes les mailles contenant un obstacle identifiées dans la zone entourée de pointillés seront considérées comme appartenant au même obstacle que celui composé des 9 mailles colorées (cf. figure 6.1(b)).

(b) Exemple d'environnement dont 7 mailles contiennent un obstacle, formant en tout 2 obstacles (chacun entouré en pointillés).

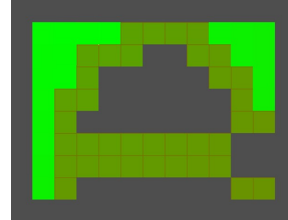
FIGURE 6.1 – Exemple d'obstacles composés de mailles contiguës ou non. Les mailles contenant un obstacle (notées 2 dans le tableau `obs`) sont hachurées en bleu, les mailles qui leur sont adjacentes (notées 1 dans le tableau `obs`) contiennent des pois orange.



(a) Mailles de 1 m. 3 obstacles considérés.



(b) Mailles de 2 m. 1 obstacle considéré.



(c) Mailles de 3 m. 1 obstacle considéré.

FIGURE 6.2 – Environnement discrétisé selon 3 tailles de mailles.

toutes les mailles de l'obstacle (comme illustré en figure 6.3, p. 97). On note $o_{i_{\text{Min}}}$ (resp. $o_{i_{\text{Max}}}$) la référence de la première ligne contenant l'obstacle (resp. dernière). De même pour les colonnes $o_{j_{\text{Min}}}$ et $o_{j_{\text{Max}}}$, et pour les altitudes $o_{k_{\text{Min}}}$ et $o_{k_{\text{Max}}}$.

Détection des obstacles

La première étape d'une itération est la détection des obstacles présents autour du drone, dans la portée de son capteur. Les informations relatives à la présence d'obstacles fixes sont stockées dans le tableau `obs` et les potentiels associés sont calculés et sommés dans le tableau `pot`. Le traitement des

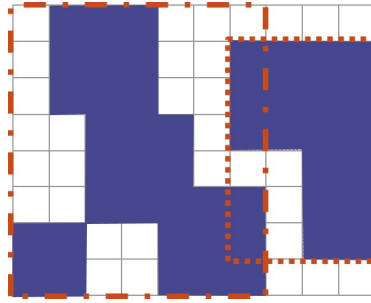


FIGURE 6.3 – Enveloppe d’obstacles. Les obstacles sont représentés en bleu et leur enveloppe par des pointillés orange.

informations liées aux obstacles sont présentés par l’algorithme 3 (p. 97).

De plus, si un obstacle est détecté entre le drone et son point-cible, alors un champ de potentiel temporaire est créé, afin d’anticiper son évitement. Ce champ est stocké dans le tableau `avoid` et est défini en section 6.3.1 (p. 98). Le traitement des informations liées aux obstacles nécessite donc la mise à jour des trois tableaux, `obs`, `pot` et `avoid`.

Algorithme 3 Prise en compte des obstacles détectés.

Entrées : Maille du drone uav_{ijk} , maille cible $cible_{ijk}$, matrices `pot` et `obs`
pour chaque maille m_{ijk} découverte par le capteur de détection des obstacles
faire
 si la maille est détectée comme étant sûre (et qu’elle était inconnue) **alors**
 Mettre à jour la matrice `obs` : $obs_{ijk} \leftarrow 0$
 sinon si un nouvel obstacle est détecté **alors**
 Mettre à jour la matrice `obs`
 Mettre à jour la matrice `pot` (eq. 6.2, p. 99)
 si la maille m_{ijk} est suffisamment proche d’un obstacle connu* **alors**
 L’ajouter à cet obstacle
 sinon
 L’enregistrer en tant que nouvel obstacle
 fin si
fin pour

*Comme illustré par la figure 6.1(a) (p. 96).

6.3 Méthode de calcul de chemin

6.3.1 Définition des champs de potentiels discrets

Les paragraphes ci-dessous introduisent les définitions des champs de potentiels attractif et répulsifs que nous proposons.

Potentiel relatif à la cible

Deux types de champs de potentiels attractifs, liés au point-cible, prédominent dans la littérature : l'un proportionnel à la distance à la cible et l'autre proportionnel au carré de cette distance. Ces définitions correspondent respectivement à une force d'attraction constante et à une force linéaire.

Dans la littérature, le champ répulsif lié à un obstacle est en général constant, c'est ce que nous choisirons ici (cf. paragraphe suivant). Nous considérons ici des zones d'intérêt de grandes dimensions, et souhaitons que tous les obstacles aient la même influence, peu importe leur distance au point-cible. Une attraction constante nous paraît donc plus adaptée. Le potentiel relatif au point-cible est donc défini par la distance euclidienne dans l'espace discrétisé, entre la maille considérée et la maille de la cible. Le potentiel relatif à la cible, en maille d'indice (i, j, k) est noté p_{ijk}^c et est défini comme suit :

$$p_{ijk}^c = \|m_{ijk} - \text{cible}_{ijk}\|. \quad (6.1)$$

Potentiel relatif à un obstacle fixe

Les obstacles fixes ont de l'influence sur le comportement d'un drone pour deux raisons : d'une part l'engin doit éviter les collisions, et d'autre part il doit emprunter un chemin aussi court que possible pour atteindre sa cible. Nous avons choisi de traiter distinctement ces deux aspects. L'évitement est abordé dans cette section tandis que l'influence sur la longueur de chemin est présentée dans le paragraphe suivant.

Le champ de potentiel lié à l'évitement d'obstacle fixe a une portée d'une maille autour de celui-ci, ce qui représente la distance de sécurité à maintenir autour du drone. Si un drone se déplace dans une maille contenant un obstacle ($\text{obs}_{ijk} = 2$), alors il entre en collision avec celui-ci. Le potentiel d'une maille contenant un obstacle doit donc empêcher un drone de la traverser.

On souhaite que les plateformes maintiennent une distance de sécurité autour des obstacles, et donc qu'elles ne traversent pas les mailles adjacentes à un obstacle ($\text{obs}_{ijk} = 1$). On crée donc un potentiel répulsif dans ces mailles-ci également. Néanmoins, si un drone se trouve dans un environnement fortement contraint, il peut être toléré qu'il entre dans une maille adjacente à un obstacle. Le potentiel répulsif des mailles autour d'un obstacle doit donc être inférieur à celui associé aux mailles avec obstacle.

On note (o_i, o_j, o_k) la maille dans laquelle un obstacle vient d'être découvert. Le potentiel lié à cet obstacle, noté p_{ijk}^o , est défini de la manière suivante :

$$\forall m_{ijk} \mid \begin{cases} (i, j, k) \in \llbracket o_i - 1, o_i + 1 \rrbracket \times \llbracket o_j - 1, o_j + 1 \rrbracket \times \llbracket o_k - 1, o_k + 1 \rrbracket \\ \text{obs}_{ijk} \neq 2 \end{cases}$$

$$p_{ijk}^o = \begin{cases} c_{\text{obs}} & \text{si } (i, j, k) = (o_i, o_j, o_k) \\ \frac{c_{\text{obs}}}{2} & \text{sinon} \end{cases} \quad (6.2)$$

Anticipation de l'évitement d'obstacles fixes

Il est utile d'anticiper l'influence d'un obstacle² dès sa découverte, afin de raccourcir le chemin parcourus par le drone. Si l'obstacle découvert est situé sur le chemin calculé par le drone pour rejoindre sa cible, une fonction d'anticipation de son évitement est appelée et un nouveau chemin le contournant est calculé.

Le potentiel associé à l'anticipation de l'évitement d'un obstacle est noté $p_{ijk}^{\text{anticipation}}$ et est défini par l'algorithme 4 (p. 99). Son effet sur le potentiel total est illustré en figure 6.4 (p. 100). Intuitivement, l'anticipation de l'évitement d'un obstacle concerne l'ensemble des mailles situées entre le drone et l'obstacle, et le potentiel associé est d'autant plus élevé que la maille est proche de l'obstacle.

Algorithme 4 Anticipation de l'évitement d'obstacle par la création d'un champ de potentiel discret et dynamique, évoluant avec la position du drone.

Entrées : Mailles uav_{ijk} et cible $cible_{ijk}$ et tableaux pot , obs et avoid .

Maille entre le drone et son point-cible contenant un obstacle, notée o_{ijk} et coordonnées de l'enveloppe de l'obstacle complet*.

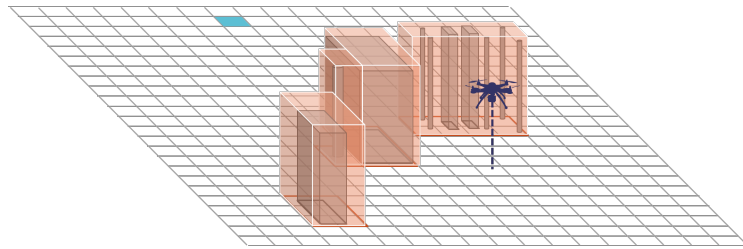
```

pour  $i \leftarrow \min(uav_i, o_{i\text{Min}})$  à  $\max(uav_i, o_{i\text{Max}})$  faire
  pour  $j \leftarrow \min(uav_j, o_{j\text{Min}})$  à  $\max(uav_j, o_{j\text{Max}})$  faire
    pour  $k \leftarrow \min(uav_k, o_{k\text{Min}})$  à  $\max(uav_k, o_{k\text{Max}})$  faire
      si  $\theta_{\text{Min}} < \text{angle}(m_{ijk}, uav_{ijk}, cible_{ijk}) < \theta_{\text{Max}}$  et  $\|cible_{ijk}, o_{ijk}\| <$ 
         $\|uav_{ijk}, o_{ijk}\|$  alors
         $p_{ijk}^{\text{anticipation}} = \text{ceil}\left(\frac{c_{\text{obs}}}{\|m_{ijk}, o_{ijk}\| + 1}\right) + \|uav_{ijk}, o_{ijk}\| - \|m_{ijk}, o_{ijk}\|$ 
      fin si
    fin pour
  fin pour
fin pour

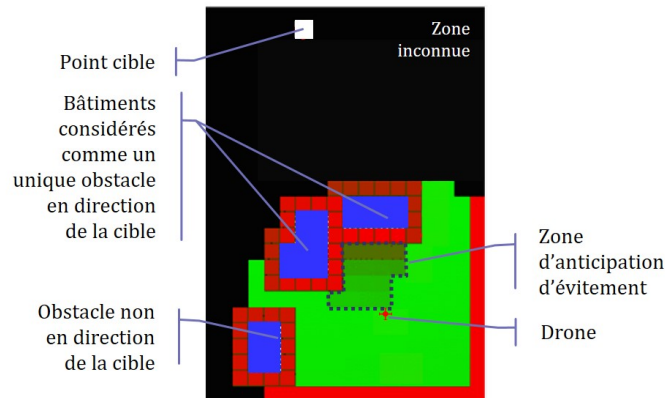
```

*Comme représenté en figure 6.3 (p. 97).

2. potentiellement constitué de plusieurs mailles, comme défini dans la section 6.2.2 (p. 95)



(a) Vue en 3 dimensions de l'environnement. Les parallélépipèdes orange représentent les mailles considérées comme contenant un obstacle, les obstacles physiques sont représentés en gris. La maille colorée est cyan est celle contenant le point-cible.



(b) Vue du dessus. Le point-cible est illustré par un carré blanc. La couleur de chaque maille indique le potentiel total associé : du vert clair pour les plus faibles, au rouge pour les plus élevés. Le potentiel des obstacles fixes n'est pas illustré par souci de lisibilité, car trop élevé en comparaison de ceux des autres mailles, les mailles contenant des obstacles sont colorées en bleu foncé)⁴.

FIGURE 6.4 – Anticipation de l'évitement d'obstacles et représentation de l'environnement par le drone.

Somme des potentiels

Le calcul d'un chemin prend en compte les champs liés à plusieurs éléments : (i) le champ attractif lié au point-cible, (ii) les champs répulsifs liés aux obstacles fixes et (iii) les champs temporaires répulsifs correspondants à l'anticipation de l'évitement des obstacles fixes.

Les champs (i) et (ii) sont les uniques champs stockés dans le tableau `pot`, on a donc :

$$\text{pot}_{ijk} = p_{ijk}^c + p_{ijk}^o. \quad (6.3)$$

Le champ temporaire (iii) est le seul enregistré dans le tableau `avoid`, on a donc :

$$\text{avoid}_{ijk} = p_{ijk}^{\text{anticipation}}. \quad (6.4)$$

Le calcul de chemin est donc réalisé à partir du champ de potentiel résultant de la somme des champs stockés dans les tableaux `pot` et `avoid`. On note p_{ijk}^{tot} la somme de tous les potentiels de la maille m_{ijk} , soit :

$$p_{ijk}^{\text{tot}} = \text{pot}_{ijk} + \text{avoid}_{ijk}. \quad (6.5)$$

Il est important de noter que le fait de sommer les potentiels associés aux différentes contraintes identifiées, permet d'étendre aisément la méthode à de nouvelles exigences, qui s'exprimeraient elles aussi sous la forme de potentiels. C'est ce qui sera réalisé lorsque plusieurs drones seront considérés (cf. chapitre 7 (p. 124)).

6.3.2 Calcul d'un chemin à partir des champs de potentiels discrets

Un chemin est calculé après la détection des obstacles éventuels et leurs prise en compte, comme le montre l'algorithme 2 (p. 94). Lorsqu'un drone calcule son chemin, il dispose donc des informations liées à la présence d'obstacles dans la portée de son capteur, ainsi que des éventuelles données d'environnement qu'il a déjà récoltées.

Plus un modèle de mobilité est générique, plus il y a de chances qu'il soit adapté à des situations imprévues lors de sa conception. C'est donc ce que nous visons dans l'élaboration de notre modèle. Néanmoins, deux cas particuliers ont été identifiés, dans lesquels le modèle était adapté mais non optimal. La présentation des différents modes de fonctionnement de ce modèle fait l'objet des paragraphes suivants.

Cas général

Lorsqu'un véhicule calcule son chemin par une méthode de champs de potentiels, son prochain déplacement est calculé par une descente de gradient. L'équivalent de cette méthode sur un champ de potentiels discret consiste à rechercher la maille adjacente ayant le plus faible potentiel, et à se déplacer dans sa direction (cf. algorithme 5, p. 102). Cette méthode s'avère efficace dans les environnements simples.

Algorithme 5 Calcul d'un déplacement par la méthode de champs de potentiels discrets.

Entrées : Maille du drone uav_{ijk} , maille cible $cible_{ijk}$, matrices pot et obs
 Identifier la maille adjacente de plus faible potentiel, la stocker dans la variable m_{minPot}
 Se déplacer vers m_{minPot}

Néanmoins, malgré l'utilisation de la fonction d'anticipation d'évitement des obstacles, les environnements les plus complexes peuvent faire apparaître des zones de minima locaux. Dans ces cas-là, le drone ne s'arrête certes pas, mais reste dans cette zone sans pouvoir s'en extraire. Or, si le drone prend en compte ses mouvements passés et planifie plusieurs déplacements à la fois, il peut sortir de ces zones de minimum. De plus, en associant l'anticipation d'évitement d'obstacles avec une portée de détection suffisamment grande par rapport aux dimensions des obstacles, il évite la zone de minimum local (en d'autres termes, le drone n'entre pas dans la zone de minima, donc il n'a pas à en sortir).

On propose donc d'étendre l'algorithme 5 (p. 102) au calcul des déplacements pour plusieurs itérations. On appelle alors *waypoints* les mailles sélectionnées. Cette méthode est présentée dans l'algorithme 6 (p. 102).

Algorithme 6 Calcul de plusieurs déplacements successifs par la méthode de champs de potentiels discrets.

Entrées : Mailles uav_{ijk} et $cible_{ijk}$, tableaux pot , obs et $avoid$
 $WP_0 \leftarrow uav_{ijk}$
 $\alpha \leftarrow 1$
tant que $WP_\alpha \neq cible_{ijk}$ **et** $\alpha <$ borne prédéfinie **faire**
 Calculer le potentiel des mailles adjacentes à $WP_{\alpha-1}$
 Stocker la maille au potentiel minimal dans m_{minPot}
 $WP_\alpha \leftarrow m_{minPot}$
 $\alpha \leftarrow \alpha + 1$
fin tant que
 Se déplacer vers WP_0

Dans le cas où, dans une itération donnée, une même maille serait sélectionnée à 2 reprises en tant que *waypoint*, cela signifie que tous les *waypoints* intermédiaires ne sont pas utiles pour atteindre le point-cible puisque le chemin proposé comporte une boucle. Dans ce cas, tous les *waypoints* de cette boucle sont supprimés, et les mailles associées ne peuvent plus être sélectionnées au cours de cette itération.

D'autre part, comme le montre la figure 6.5 (p. 103), si le drone vise successivement les différents *waypoints* calculés, ses mouvements restent limités au maillage, tout comme la distance maximale qu'il peut parcourir en une itération. La suppression de *waypoints* intermédiaires, dès que l'environnement l'autorise, permet de raccourcir le chemin.

L'algorithme 7 (p. 104) présente cette méthode de calcul de *waypoints* et la suppression des intermédiaires non utiles.

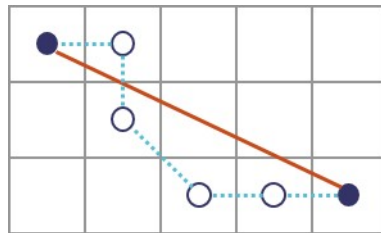


FIGURE 6.5 – Intérêt de la suppression des *waypoints* intermédiaires : le chemin direct entre les *waypoint* extrêmes (disques), illustré en trait plein rouge, est significativement plus court que chemin en pointillés bleus reliant tous les *waypoints* calculés de manière successive (cercles).

Cas sans obstacles en direction de la cible

Dans un environnement sans obstacles, le chemin le plus court est la ligne droite. Pour cette raison, lorsqu'aucun obstacle n'est détecté par le drone entre lui-même et son point-cible, il ne calcule qu'un seul *waypoint* (et supprime tous ceux précédemment calculés). Ce *waypoint* correspond à la maille la plus proche du point-cible qui est dans la portée du capteur de détection d'obstacle. On note ce cas particulier CP1.

Cas d'une impasse

Les environnements complexes peuvent comporter des impasses. Un drone peut, sans le savoir, se engouffrer, car la portée de son capteur de détection d'obstacles est limitée. Cette situation n'est pas problématique si le point-cible s'y trouve également. Dans le cas contraire, on considère cette situation comme un cas particulier, que l'on note CP2.

Algorithme 7 Amélioration du calcul de plusieurs déplacements successifs par la méthode des champs de potentiels discrets.

Entrées : Mailles uav_{ijk} et $cible_{ijk}$, tableaux pot , obs et $avoid$

```

WP0 ← uavijk
α ← 1
tant que WPα ≠ cibleijk et nombre d'essais maximal non atteint faire
  Calculer le potentiel  $p_{ijk}^{tot}$  des mailles non interdites adjacentes à WPα-1
  Stocker la maille au potentiel minimal dans  $m_{minPot}$ 
  // Suppression d'une boucle éventuelle
  si un waypoint d'indice noté β est déjà associé à la maille  $m_{minPot}$  alors
    Interdire de sélection tous les waypoints successifs à WPβ
    Supprimer tous les waypoints d'indice strictement supérieur à β
    α ← β
  sinon
    WPα ←  $m_{minPot}$ 
  fin si
  α ← α + 1
fin tant que
// Suppression des waypoints intermédiaires non utiles
γ ← 2
tant que γ est strictement inférieur au nombre de waypoints faire
  si le chemin est sûr entre WPγ-2 et WPγ alors
    Supprimer WPγ-1
    Décrémenter de 1 les indices des waypoints supérieurs ou égaux à γ
  sinon
    γ ← γ + 1
  fin si
fin tant que
Se déplacer vers WP0

```

La sortie d'une impasse n'est pas triviale à effectuer avec la méthode des champs de potentiels, d'autant plus si le drone, l'extrémité de l'impasse et le point-cible sont alignés, car l'extrémité de l'impasse représente alors un minimum local.

Une méthode efficace pour sortir d'une impasse consiste à rebrousser chemin : le drone considère comme *waypoints* la liste de ses précédentes positions, et supprime éventuellement les *waypoints* non-utiles. Dès que le drone est en dehors de cette impasse, ils reprend son calcul de chemin de manière classique.

Dès qu'un drone détecte qu'il se trouve dans une impasse, il doit en sortir puis s'assurer de ne pas y retourner, tant qu'il n'a pas atteint son point-cible courant⁵. Il faut donc augmenter le potentiel des mailles à l'intérieur de l'im-

5. Il n'est pas exclu qu'un point-cible ultérieur se trouve dans cette impasse, le potentiel

passee. Plus une maille est proche du fond de l'impasse, moins il est pertinent de la sélectionner pour le calcul du chemin vers le point-cible. À l'inverse, il peut être utile au drone de s'approcher de l'entrée de l'impasse. On définit donc le potentiel répulsif des mailles à l'intérieur de l'impasse de sorte à ce qu'il soit d'autant plus élevé que la maille est proche du fond de celle-ci.

On note $p_{\max}^{c,impasse}$ le potentiel lié à la cible le plus élevé, pris par une maille dans l'enveloppe de l'obstacle constituant l'impasse. L'équation 6.6 définit l'augmentation de potentiel pour les mailles à l'intérieur de l'enveloppe de l'obstacle :

$$p_{ijk}^{impasse} = \max(p_{ijk}, 2 \times p_{\max}^{c,impasse} - p_{ijk}) \quad (6.6)$$

La figure 6.6 (p. 106) illustre la modification de potentiel associée à une impasse.

Cas où un drone recalcule son chemin

Notre méthode permet à un drone de ne pas avoir à calculer ses déplacements systématiquement, puisque plusieurs *waypoints* sont calculés en une itération. Plus précisément, les cas où un nouveau chemin doit être calculé sont les suivants :

- Le drone n'a connaissance d'aucun obstacle entre lui et son point-cible (il passe alors dans le cas particulier 1). On note que tant que le drone est dans cette situation, il ne calcule pas un chemin mais seulement le prochain point qu'il devra atteindre.
- Le drone détecte qu'il se trouve dans une impasse (il passe alors dans le cas particulier 2).
- Le drone détecte un obstacle sur le chemin qu'il avait déjà calculé : il calcule à nouveau son chemin en prenant en compte cette nouvelle information.

En dehors de ces cas, le drone continue à suivre le chemin qu'il a déjà calculé et supprime éventuellement les *waypoints* devenus inutiles, du fait de la découverte de l'environnement sans obstacle autour d'eux.

Exemple de calcul de chemin pour les différents cas

La figure 6.7 (p. 107) représente le chemin calculé par un drone sur quelques itérations, considérant le calcul de chemin effectué dans le cas général et dans les deux cas particuliers CP1 et CP2 (présentés dans les paragraphes précédents). L'environnement considéré est en deux dimensions et mesure 120 m de long et 40 m de large, les mailles ont une largeur de 1 m. La vitesse maximale autorisée est de $10 \text{ m}\cdot\text{s}^{-1}$ et la portée de détection d'obstacles est de 24 m.

répulsif associé à celle-ci doit donc être temporaire

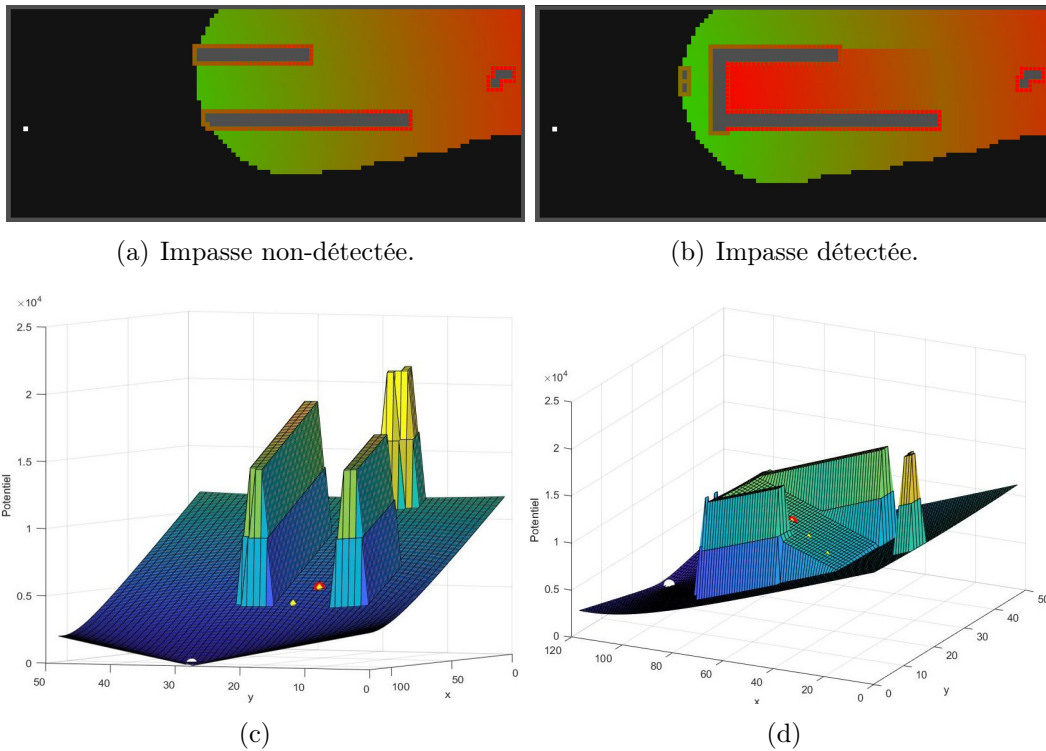


FIGURE 6.6 – Augmentation du potentiel dans une impasse. Les figures 6.6(a) et 6.6(b) sont des vues de dessus de l’environnement connu par le drone à des dates successives. Les mailles inconnues par le drone sont noires, et les nuances de couleurs des mailles connues indiquent leur potentiel (du vert clair pour le plus faible, au rouge pour les plus élevés, obstacles en bleu foncé). Les figures 6.6(c) et 6.6(d) sont des vues en 3D du potentiel calculé par le drone, correspondant respectivement aux figures 6.6(a) et 6.6(b).

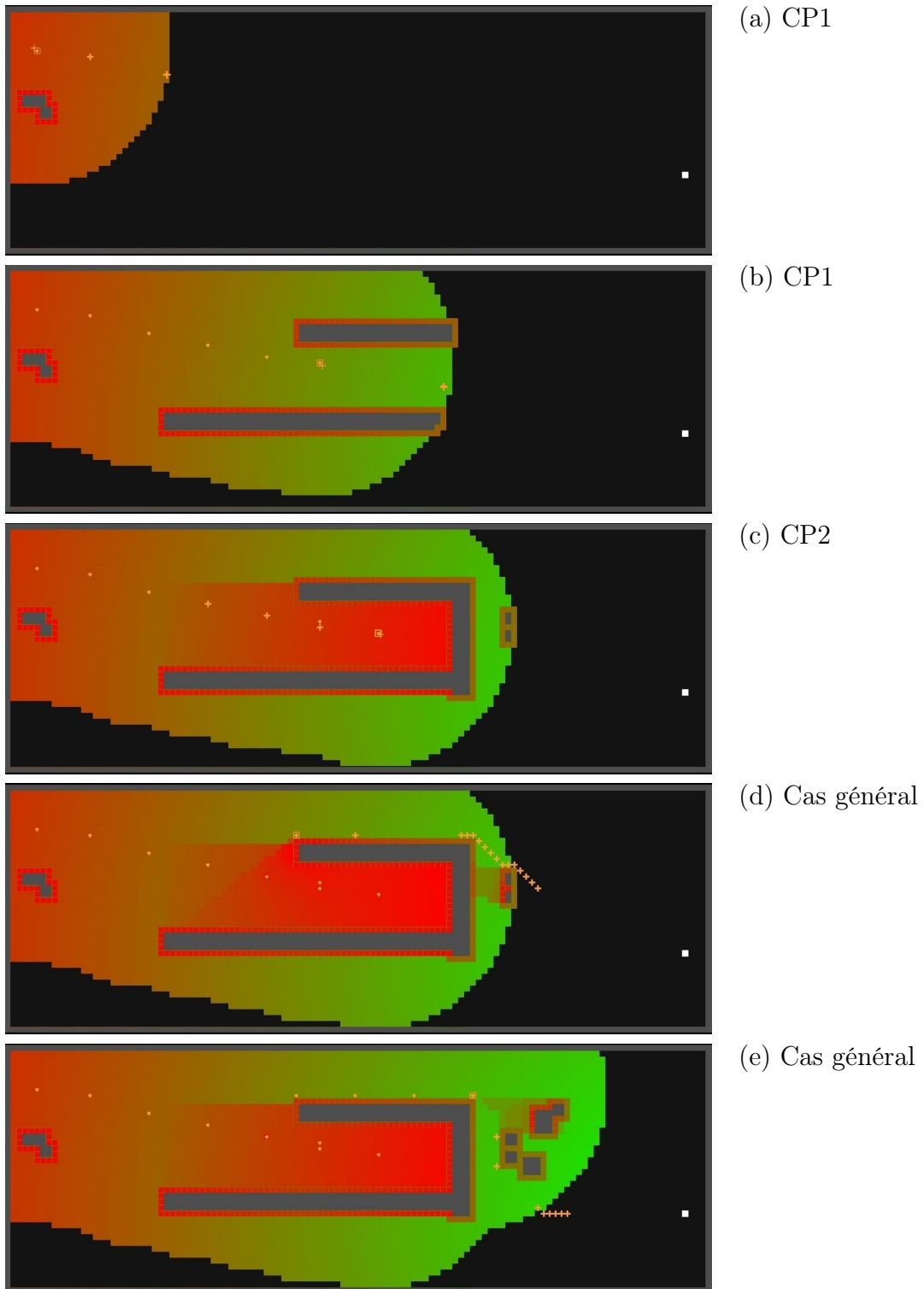


FIGURE 6.7 – Mouvements d'un drone dans un environnement inconnu, pour quelques itérations. Les petites croix représentent les mailles où le drone s'est arrêté à la fin d'une itération, les grandes représentent les *waypoints* calculés.

À la première itération (figure (a)), le drone détecte des obstacles qui ne sont pas entre lui et la cible ; il n'applique donc pas la fonction d'anticipation d'évitement. Dans la portée de son capteur, il n'y a pas d'obstacle entre lui et son point-cible, il vise donc la maille qu'il connaît la plus proche de la cible conformément au cas particulier CP1. Tant que le drone ne découvre pas d'obstacle en direction du point-cible, il continue à calculer son chemin de la même manière (figure (b)).

Lors de l'itération représentée en figure (c), le drone a identifié qu'il se trouvait dans une impasse. Il retourne donc sur ses pas jusqu'à ce qu'il en sorte. De plus, pour ne pas y entrer à nouveau, il crée un potentiel répulsif dans toute la zone intérieure de l'impasse. Ce potentiel répulsif existera tant que le drone n'aura pas atteint le point-cible courant.

Une fois le drone sorti de l'impasse (figure (d), puis (e)), il se trouve dans le cas général de calcul de chemin (il n'est pas dans une impasse et des obstacles sont présents entre lui et son point-cible). Lorsque des obstacles sont détectés entre la plateforme et sa cible, le drone applique la fonction d'anticipation d'évitement, afin de contourner les-dits obstacles au plus tôt.

6.3.3 Avantages par rapport à un APF classique

Comme nous l'avons expliqué en section 4.3.4 (p. 51), la méthode proposée par [Khatib \[1986\]](#) présente trois faiblesses : **GNRON** (point-cible non atteignable si proche d'un obstacle), l'impossibilité de passer entre des obstacles proches et la possibilité de minima locaux. Les paragraphes suivants présentent comment notre méthode adresse ces problématiques.

GNRON

Le potentiel lié à un obstacle que nous proposons n'a d'influence qu'à une maille de distance, ce qui correspond à la distance de sécurité à maintenir autour de celui-ci. Cela permet donc à un drone de s'approcher des obstacles, tant que la distance de sécurité est maintenue.

Impossibilité pour une plateforme de passer entre des obstacles proches

Nous considérons une faible zone d'influence des obstacles, ce qui permet, dans le cas d'une discrétisation fine, de passer entre des obstacles proches.

Minima locaux

L'anticipation de l'évitement des obstacles permet de supprimer la majorité des minima locaux. Certains peuvent néanmoins apparaître dans des cas complexes, mais ils ne provoquent pas l'arrêt du drone compte tenu du calcul de chemin global.

6.4 Paramètres de discrétisation et de simulation

6.4.1 Discrétisation de l'environnement

Les discrétisations horizontale et verticale de l'environnement dépendent des éléments suivants :

1. la taille du drone ;
2. la portée du capteur de détection des obstacles ;
3. le type d'environnement considéré.

1. Le modèle repose sur l'hypothèse que les mailles considérées peuvent contenir un drone entier. La taille de maille doit donc être supérieure à la taille du drone.

2. Le calcul d'un chemin se fait par la sélection successive de plusieurs mailles que le drone cherche à atteindre. À chaque itération, le drone se dirige vers la première maille de son chemin, celle-ci doit nécessairement être libre. Une maille doit donc pouvoir être observée entièrement par le capteur de détection d'obstacle, c'est à dire être plus petite que la portée de détection.

3. Comme la taille et la forme des obstacles discrétisés sont dépendantes du maillage, la définition de la discrétisation influence fortement les chemins calculés (cf. section 6.2.2, p. 95 et figure 6.2, p. 96). Si l'environnement comporte une forte densité d'obstacles, en particulier des obstacles petits et proches les uns des autres, la discrétisation devra être fine. Plus précisément, la largeur de maille devra être plus petite que la distance séparant les obstacles pour qu'un drone puisse passer entre eux. Dans le cas d'une faible densité d'obstacles, un maillage grossier sera suffisant.

6.4.2 Distance maximale à parcourir en une itération

La prochaine maille visée par le drone, ainsi que toutes celles qu'il doit traverser pour l'atteindre, doivent être identifiées comme libres. La distance parcourue en une itération est donc limitée par la portée du détecteur d'obstacle.

6.4.3 Environnement simulé

L'environnement de simulation que nous choisissons est un site protégé, ce qui est représentatif du besoin que nous considérons. Plus précisément, nous avons simulé les bâtiments d'une partie de la centrale du Blayais, à partir des données cadastrales [Geoportail, 2019]. La zone simulée mesure 500 m de large, 520 m de long et 60 m de hauteur, avec une précision d'un mètre. Cet environnement de tests est composé de bâtiments aux formes et dimensions

variées (parallélépipédiques aux proportions diverses, cylindriques ou formes concaves, de hauteur entre 4 et 50 m) et espacés de distances variées également, comme le représente la figure 6.8 (p. 110). Cet environnement peut donc être considéré comme suffisamment représentatif d'un environnement complexe. Le fichier de description de l'environnement utilisé est disponible au format xml à l'adresse suivante : https://www.researchgate.net/publication/336240022_Environnement_de_simulation.

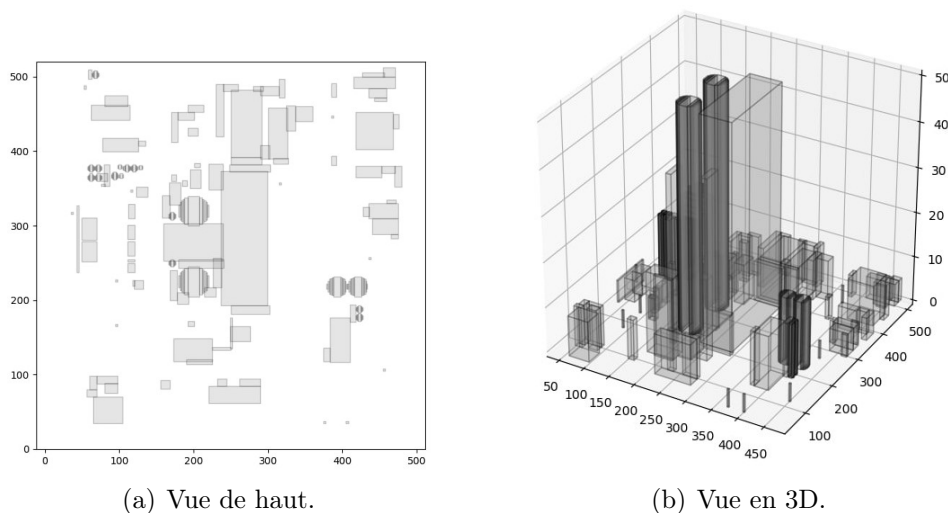


FIGURE 6.8 – Environnement de simulation

6.4.4 Paramètres de simulation

Le drone simulé est un multiroteur de 40 cm d'envergure, que nous limitons à une vitesse maximale de 10 m.s^{-1} . Il peut détecter les obstacles à une portée de 24 m (ce qui représente les capacités d'une caméra stéréo, comme détaillé en section 3.6, p. 34). La validation du modèle se fait ici par la simulation de différents chemins que nous énumérons ci-dessous et que détaillons dans les paragraphes suivants :

- i. Onze points sont tirés aléatoirement dans l'environnement libre (c.-à-d. ces points ne sont pas à l'intérieur d'un obstacle). Le drone débutera sa mission au premier point et devra observer successivement, et dans l'ordre, les dix autres points (représentant chacun un point-cible). On considère qu'un point est observé s'il est dans la portée de détection du capteur embarqué et qu'aucun obstacle n'est présent entre le drone et le point en question. Trois tirages de onze points sont effectués (points donnés en annexe 7.5, p. 149), que l'on nommera échantillons n^{os} 1, 2 et 3.
- ii. Pour chaque tirage, dix tailles de maille sont simulées : de 1 à 10 m de large, avec un pas de 1 m.

iii. Le drone peut évoluer entre 4 et 24 m d'altitude, avec une discrétisation de 4 m.

i. Choix de la mission

Quand le drone débute la mission, il ne dispose d'aucune information sur la présence ou la position des obstacles. Au fur et à mesure de ses déplacements dans l'environnement, il complète ses informations relatives à la présence et l'absence d'obstacles (dans le tableau `obs`) et utilise l'ensemble de ces données afin de calculer son chemin. Compte tenu du fait que les points à observer sont tirés aléatoirement dans l'environnement, plus le drone progresse dans sa mission, plus il est probable qu'il ait couvert une grande partie de la zone. La simulation de l'observation de 10 points successifs permet donc de tester notre modèle tant dans un environnement complètement inconnu, que dans le cas d'un environnement partiellement connu. On simule 3 échantillons de 10 points à atteindre, avec 10 tailles de mailles différentes : nous simulons donc le calcul de 300 chemins entre deux points.

ii. Discrétisation horizontale

Nous savons que la zone d'intérêt contient notamment des obstacles de forme cylindrique et que les espaces entre les bâtiments peuvent être étroits ; une discrétisation fine permettra donc d'explorer un maximum de lieux dans la zone. Conformément aux contraintes présentées en section 6.4.1 (p. 109) et aux paramètres de la plateforme utilisée (présentés en section 6.4.4, p. 111), la maille la plus étroite que nous pouvons simuler mesure 1 m de côté.

Puisqu'un maillage fin permet une représentation fidèle de l'environnement, l'environnement considéré par le drone d'autant plus complexe. C'est pourquoi nous répétons les simulations avec des mailles plus larges, afin d'évaluer l'influence de la discrétisation sur les performances du modèle. La largeur maximale des mailles est soumise à deux contraintes (cf. section 6.4.1, p. 109) : elle doit être inférieure à la distance maximale pouvant être parcourue par le drone en une itération et être inférieure à la portée de détection des obstacles. La plateforme considérée peut atteindre une vitesse maximale de $10 \text{ m}\cdot\text{s}^{-1}$ et la discrétisation temporelle est de 1 s ; conformément à la première contrainte, la largeur de maille ne peut pas excéder 10 m. La portée de détection des obstacles est de 24 m, une maille de 10 m est donc valide. Nous simulerons donc des mailles de largeur 1 à 10 m, avec un pas de 1 m.

iii. Discrétisation verticale

Les principaux objets mobiles dans un site protégé sont les piétons et les véhicules. Or, notre étude ne traite pas les obstacles mobiles. Nous choisissons donc de fixer l'altitude de vol minimale à 4 m, ce qui permet d'éviter ces obstacles.

L'altitude maximale est choisie en fonction de la mission. Ici, nous nous intéressons à la surveillance d'éléments au sol, nous choisissons donc la portée

du capteur embarqué comme altitude maximale (soit 24 m).

La zone d'intérêt que nous considérons est un site protégé dont on sait que les bâtiments sont soit des immeubles de bureaux soit des structures de production d'électricité. La hauteur standard d'un étage dans un immeuble de bureaux est de 3,3 m, et nous ne disposons pas d'informations sur les dimensions des structures de production d'électricité. La discrétisation verticale est donc définie à partir de la seule information connue ; une finesse proche de 3,3 m est suffisante pour représenter l'environnement de façon réaliste. Nous fixons alors la hauteur d'une maille à 4 m.

Le drone pourra donc passer au-dessus des bâtiments les plus bas et devra contourner les autres.

6.5 Résultats et évaluations

Les performances du calcul de chemin sont évaluées en fonction de plusieurs critères, que nous énumérons ci-dessous puis détaillons dans les paragraphes suivants :

- i. Absence de collision ;
- ii. Réalisation de la mission ;
- iii. Distance parcourue par itération ;
- iv. Nombre d'itérations pour remplir la mission.

i. L'absence de collision est une condition *sine qua non* de la performance du calcul de chemin.

ii. Le calcul de chemin doit pouvoir assurer l'arrivée du drone à son point-cible.

iii. Nous choisissons que chaque itération simule une durée fixe du temps réel (1 seconde en l'occurrence). Plus la distance parcourue par itération est grande, plus il est probable que le drone se rapproche rapidement de son point-cible, et donc qu'il remplisse sa mission rapidement. De plus, si le drone parcourt de grandes distances à chaque itération, c'est que le chemin calculé est de bonne qualité, puisqu'il n'a pas à être recalculé.

iv. Une itération représentant une durée fixe du temps réel, plus le nombre d'itérations réalisées est faible, plus le drone remplit sa mission rapidement.

L'ensemble des simulations menées n'ont fait apparaître aucune collision et les 300 chemins ont bien été réalisés.

L'analyse de la distance par itération et du nombre total d'itérations, en fonction de la taille de maille, font l'objet des paragraphes suivants. Il est à noter que compte tenu du fait que l'environnement est découvert à la volée

par le drone, la comparaison d'un chemin réalisé avec le chemin le plus court n'a pas de sens. Nous comparons donc les performances relatives entre les différentes finesses de discrétisation de l'environnement.

6.5.1 Distance parcourue en fonction de la taille de maille

Nous avons extrait des données statistiques sur la distance parcourue par itération en fonction de la taille de maille, à partir des données issues de l'ensemble des simulations menées. Comme le montre la figure 6.9 (p. 113), la distance médiane parcourue par itération est très proche de la distance maximale autorisée (10 m), pour toutes les tailles de maille testées.

Pour les mailles de largeur supérieure à 3 m, la distance moyenne est quasiment constante et proche de 9 m et l'écart-type est faible. Néanmoins, la distance moyenne parcourue avec une maille de 2 mètres de large est inférieure à 9 m et elle est proche de 8 m avec une maille de 1 m. Ceci est dû au fait qu'une fine discrétisation de l'environnement augmente la complexité de l'environnement considéré par le drone : il y a davantage de recoins, dans lesquels le drone ne peut pas parcourir de grandes distances.

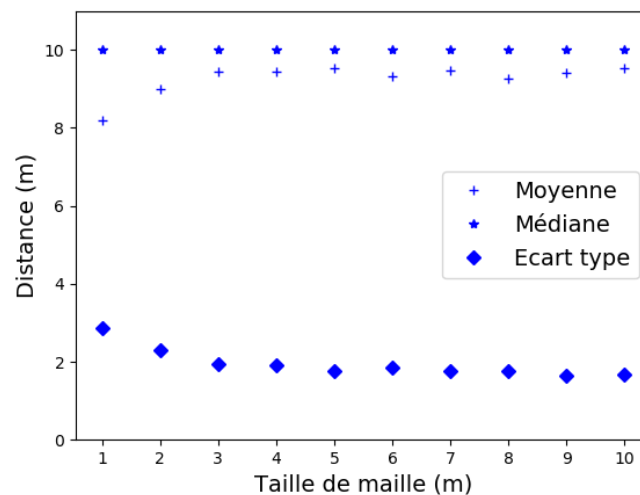


FIGURE 6.9 – Moyenne, médiane et écart-type de la distance parcourue par itération en fonction de la taille de maille.

6.5.2 Nombre d'itérations nécessaires en fonction de la taille de maille

Dans cette section nous cherchons à identifier l'importance de finesse de discrétisation sur la durée de réalisation d'une mission donnée. Nous regardons

donc le nombre d'itérations nécessaires à la réalisation de chaque mission (c.-à-d. l'observation des 10 points tirés aléatoirement de chaque échantillon) en fonction de la taille de maille.

Échantillon n° 1

La figure 6.10 (p. 114) illustre les résultats associés à l'échantillon n° 1. On observe des résultats très similaires pour atteindre tous les points-cible à l'exception du n° 6 et de la maille de 4 m lors du chemin entre les points-cible n°s 3 et 4. Les paragraphes ci-dessous en expliquent les raisons, en s'appuyant sur la figure 6.11 (p. 115).

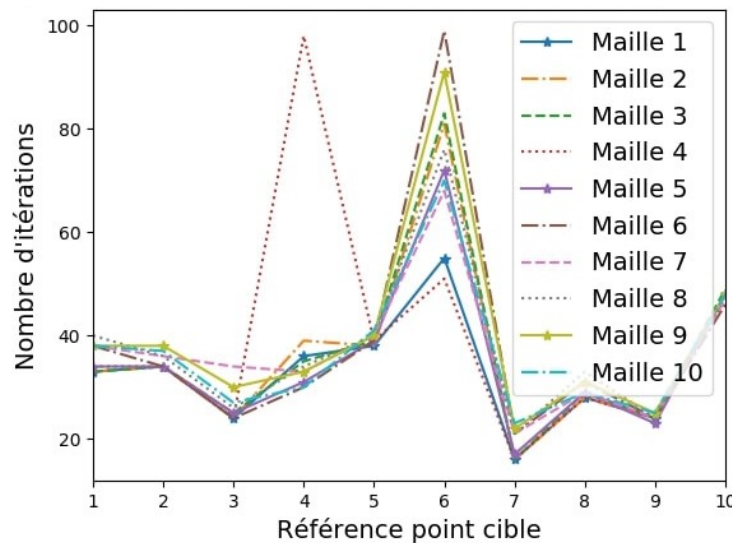


FIGURE 6.10 – Nombre d'itérations entre deux points-cible successifs, pour l'échantillon n° 1, en fonction de la taille de maille.

Le grand nombre d'itérations permettant d'atteindre le 6^e point-cible est dû à la présence d'un obstacle de grandes dimensions (environ 100 m de long, 60 m de large et trop haut pour être survolé) entre les points-cible n°s 5 et 6. À l'approche de cet obstacle, le calcul de chemin prend en compte le potentiel lié à l'anticipation de l'évitement. Ce type de champ de potentiel, adapté à la forme de l'obstacle considéré, impose de choisir dès la sélection du premier *waypoint* un évitement par un côté ou un autre. À partir de cette première décision, la suite du chemin exclut toute une zone de recherche (le côté ne correspondant pas à celui du premier *waypoint*). Ceci représente une des limitations du modèle.

Les chemin calculés avec les mailles de largeur 1 et 4 m contournent l'obstacle par la droite (du point de vue du drone), ce qui s'avère être le chemin le plus court (voir figure 6.11(b), p. 115). Dans le cas des autres largeurs de mailles, 2 cas sont illustrés : l'évitement par la gauche (cf. figure 6.11(a)) et un

6. Calcul de chemin pour un drone autonome en environnement complexe

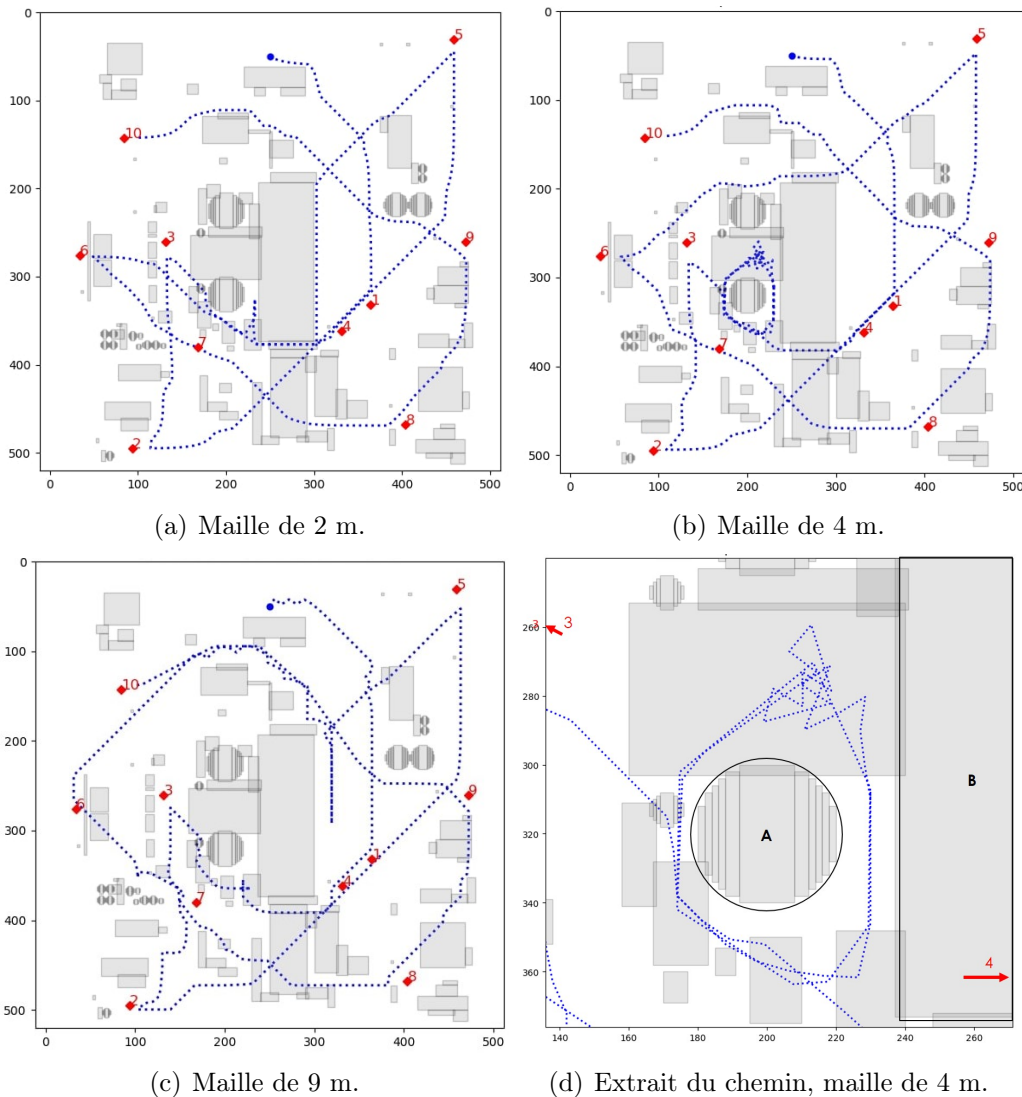


FIGURE 6.11 – Chemin suivi par le drone pour visiter les 10 points-cible (repérés par un carré et numérotés) de l'échantillon n° 1, pour 3 tailles de maille (figures (a) à (c)) et extrait du chemin entre les points n° 3 et 4 pour la maille de 4 m (figure (d) où les flèches indiquent la direction du point-cible référencé à proximité).

début d'évitement par la gauche jusqu'à découvrir la suite de l'obstacle, suivi par un évitement par la droite (cf. figure 6.11(c)).

Le chemin emprunté par le drone pour atteindre le point-cible n° 4, dans le cas d'une maille de 4 m, est en partie illustré en figure 6.11(d) (p. 115). Lorsque le drone cherche à atteindre ce point-cible, il n'a pas la connaissance de la zone représentée dans cette figure. Il commence par contourner par la droite (de son point de vue) l'obstacle noté A, puis se déplace dans la direction du point-cible n° 4. Il découvre alors une partie de l'obstacle B, et tente de l'éviter par la

gauche. Puisque le drone n'a pas encore connaissance des dimensions de l'obstacle B, cette méthode d'évitement est acceptable. Une fois que le drone n'est plus entre les obstacles A et B, le drone est dans un environnement fortement contraint et à la limite de l'anticipation de l'évitement de l'obstacle A. En fonction de l'appel ou non à cette fonction, les mailles au potentiel minimal diffèrent et le drone oscille donc. Puis, il contourne à nouveau l'obstacle A en le laissant à sa gauche, alors que retourner sur ses pas permettrait d'arriver plus rapidement à la droite de l'obstacle B, ce qui constitue un second choix inadapté. Néanmoins, l'obstacle B se trouvant entre le drone et son point-cible, la fonction d'anticipation de l'évitement des obstacles a augmenté le potentiel des mailles se trouvant entre les obstacles A et B, et le drone ne peut donc pas retourner sur ses pas.

Le second choix inadapté apparaît lorsque le drone tente à nouveau de contourner l'obstacle B par la gauche, alors qu'il a connaissance de sa longueur. Ceci est lié à la première limitation du modèle identifiée, correspondant à la limitation de la zone de recherche de chemin dans le cas de l'anticipation de l'évitement d'un obstacle, dès que le premier *waypoint* est sélectionné.

Échantillon n° 2

La figure 6.12 (p. 117) représente le nombre d'itérations nécessaires pour atteindre un point-cible depuis le précédent, en fonction de la taille de la maille, pour l'échantillon n°2. Compte tenu de la différence d'échelle entre les mailles de largeur 1 et 2, et les autres, les graphiques sont présentés séparément.

Nous commençons par décrire la tendance générale des chemins calculés avant de détailler les 4 cas particuliers que nous avons observés dans les paragraphes suivants (maille de 1 m, entre le départ et le point-cible n°1; maille de 2 m entre les n°s 1 et 2; maille de 4 m entre les n°s 7 et 8; maille de 5 m entre les n°s 4 et 5).

La figure 6.13 (p. 118) illustre les chemins calculés pour observer les 10 points-cible, pour 3 tailles de maille. On remarque qu'un maillage fin permet au drone d'emprunter davantage de passages qu'un maillage grossier (les zones notées A et B sur la figure 6.13(a), p. 118 ne sont accessibles qu'avec une maille de largeur inférieure à 4 m, inclu). La zone notée C sur la figure 6.13(b) (p. 118) n'est accessible que pour les mailles d'au plus 8 m de large. Au delà, cette impasse n'est pas accessible, ce qui permet de l'éviter avant même d'y entrer (comme le montre la figure 6.13(c), p. 118). Les maillages compris entre 5 et 8 m permettent d'entrer dans cette zone C, mais celle-ci est considérée comme une impasse. L'exploration de cette zone rallonge donc les chemins.

On en déduit que bien qu'un maillage fin autorise l'accès à davantage de zones qu'un maillage grossier, il peut entraîner la présence d'impasses dont l'exploration et la sortie rallongent le chemin. C'est pourquoi il n'y a pas de

6. Calcul de chemin pour un drone autonome en environnement complexe

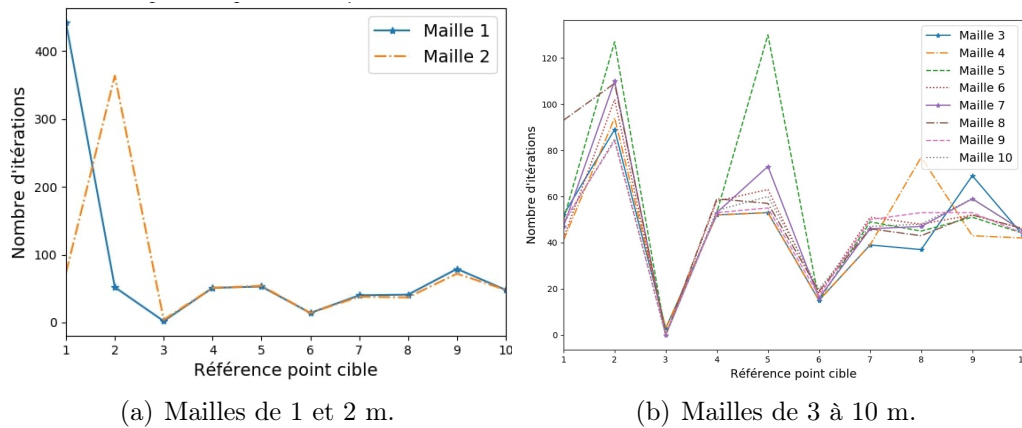


FIGURE 6.12 – Nombre d'itérations entre deux points-cible successifs, pour l'échantillon n° 2, en fonction de la taille de maille.

proportionnalité entre la taille de maille et la longueur des chemins.

La figure 6.14 (p. 119) représente les chemins correspondants aux grands nombres d'itérations ponctuels représentés sur la figure 6.12 (p. 117). Ces chemins sont considérés comme étant de mauvaise qualité car les décisions prises, dans une zone déjà connue, allongent le chemin.

Le cas illustré en figure 6.14(a), p. 119 représente une partie du cas représenté en figure 6.11(d), correspondant à un chemin de l'échantillon n°1, son explication n'est donc pas répétée ici.

Dans le cas du chemin illustré en figure 6.14(b), p. 119, le drone découvre l'obstacle en forme de L une fois qu'il est dans ce coin. Or, ce coin est parfaitement dans l'axe du point-cible qu'il vise, et lorsque le drone découvre l'obstacle, il en est déjà proche. La fonction d'anticipation d'évitement n'augmentant que le potentiel des mailles situées entre le drone et l'obstacle, le drone ne s'en éloigne que maille après maille. Notre modèle comporte donc une autre limitation : la difficulté à sortir d'un obstacle concave aligné avec le point-cible.

La zone illustrée en figure 6.14(c), p. 119 est traversée à 3 reprises par le drone (pour atteindre les points-cible n°s 2, 8 et 10). Lors de son premier passage, le drone effectue un grand nombre d'oscillations dues à la multiplicité d'obstacles, en particulier en direction du point-cible n° 2, dans cette zone fortement contrainte. De plus, des obstacles sont régulièrement découverts, ce qui entraîne le recalcul du chemin, qui ne s'avère que peu efficace ici. Néanmoins, lorsque le drone traverse cette zone par la suite, une fois qu'il en a une connaissance complète (pour atteindre les points-cible n°s 8 et 10), alors son chemin est aussi direct que possible.

La figure 6.14(d), p. 119 représente une zone de l'environnement traversée à 2 reprises par le drone (pour atteindre les points-cible n°s 2 et 5). Tout d'abord,

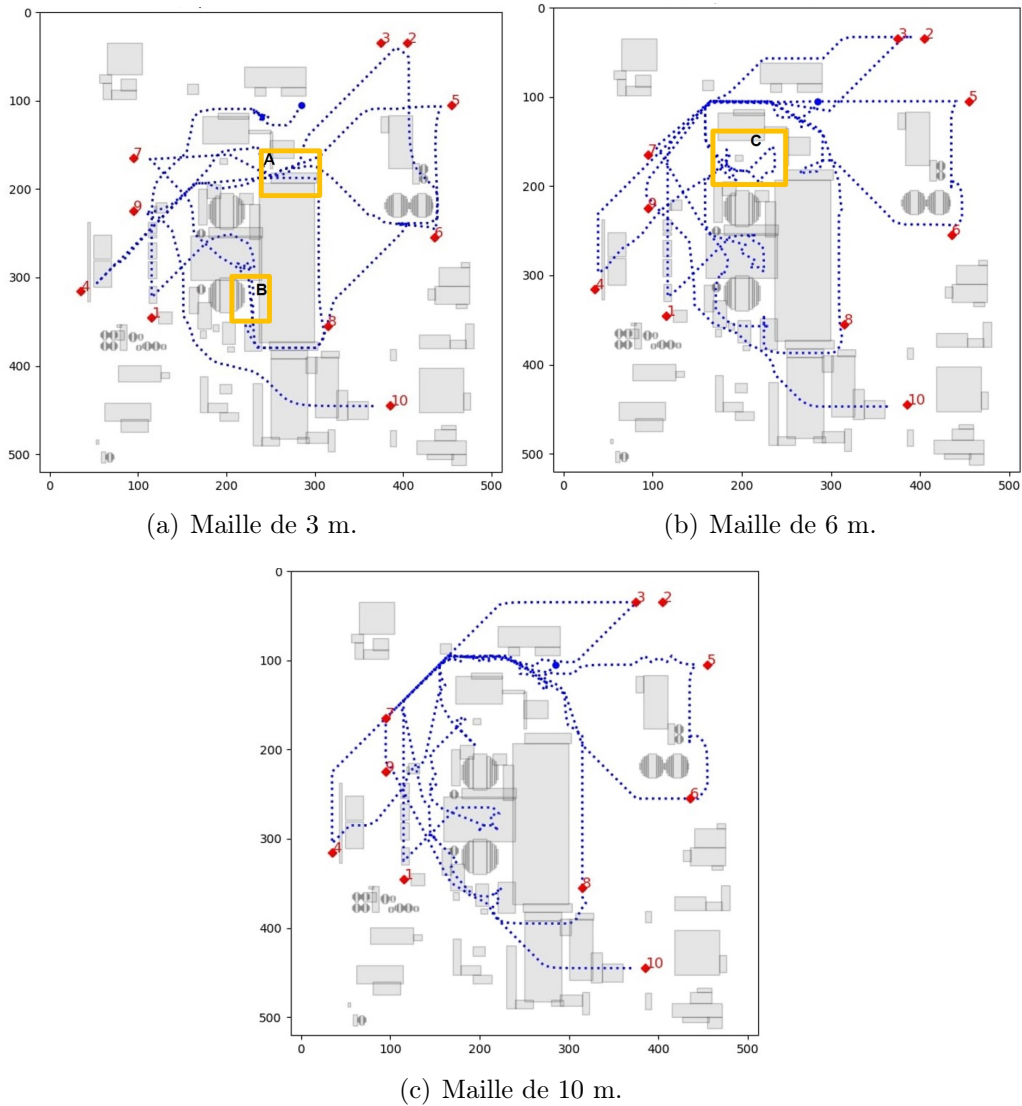


FIGURE 6.13 – Chemin suivi par le drone pour visiter les 10 points-cible de l'échantillon n° 2, pour 3 tailles de maille.

celui-ci découvre cet environnement complexe lorsqu'il tente de rejoindre le point-cible n° 2. À cet instant, puisqu'il découvre les obstacles, l'exploration de cette impasse n'est pas problématique. Néanmoins, le drone effectue un tour complet supplémentaire de la zone sans issue, ce qui allonge inutilement le chemin. De plus, il s'engouffre à nouveau dans cette zone lorsqu'il cherche à atteindre le point n° 5, alors qu'il a déjà connaissance de ces obstacles. Ce choix inadapté est dû à deux éléments : d'une part, à la première limitation identifiée : la restriction de la zone de recherche dans le cas de l'anticipation de l'évitement d'un obstacle lorsque le premier *waypoint* est sélectionné ; d'autre part, bien que l'impasse représentée ait déjà été identifiée en tant que telle,

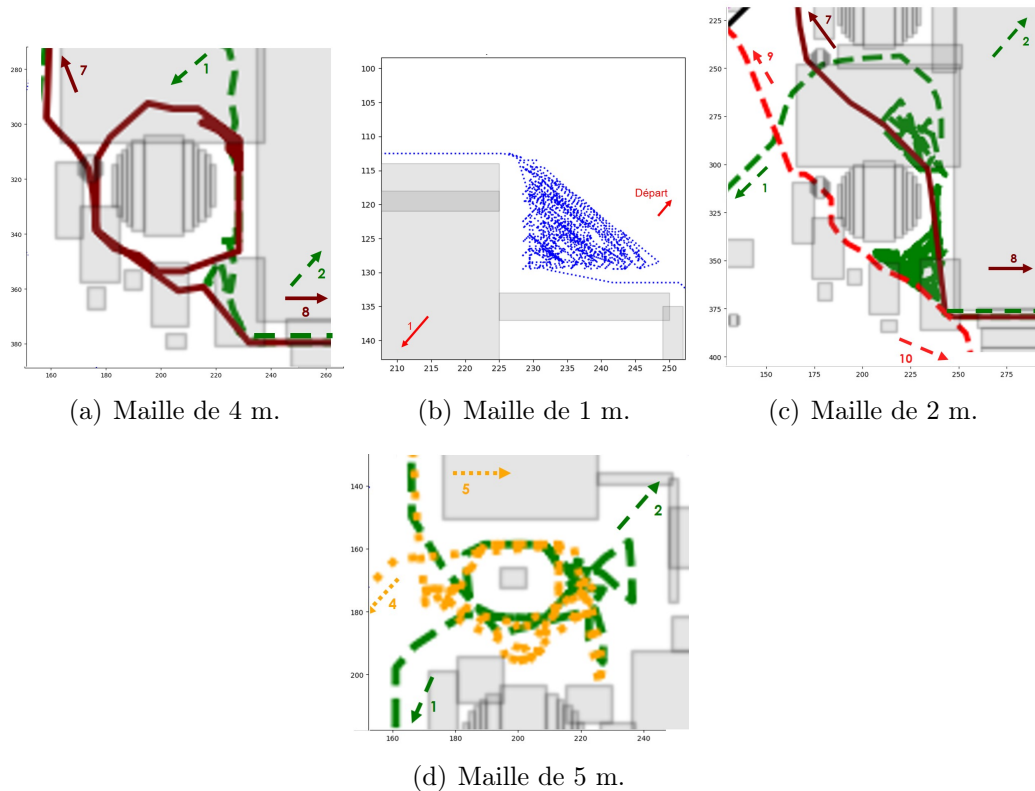


FIGURE 6.14 – Chemins de mauvaise qualité calculés dans le cadre de l'échantillon n° 2. Chaque association type de « tiret - couleur » représentée est associée au chemin entre 2 points-cible.

l'augmentation de potentiel associée n'est valide que jusqu'à l'atteinte du point-cible en cours. Un nombre de *waypoints* calculés plus important aurait pu permettre d'éviter l'entrée dans cette zone.

Échantillon n° 3

Les chemins permettant d'atteindre les points-cible de l'échantillon n° 3 dépendent peu de la taille de maille (comme le montre la figure 6.15, p. 120), car les obstacles entre les différents points sont simples (voir figure 6.16, p. 120). La principale difficulté liée à cet échantillon consiste à atteindre le point n° 6 pour les mailles d'au plus 3 m de large. En effet, ces finesses de discrétisation permettent d'entrer dans la zone notée A sur la figure 6.16(a), p. 120, de laquelle le drone est sorti sans difficulté.

Le nombre d'itérations permettant d'atteindre le point-cible n° 7 est plus important avec une maille de 1 m qu'avec les autres mailles, pour les mêmes raisons que détaillées précédemment.

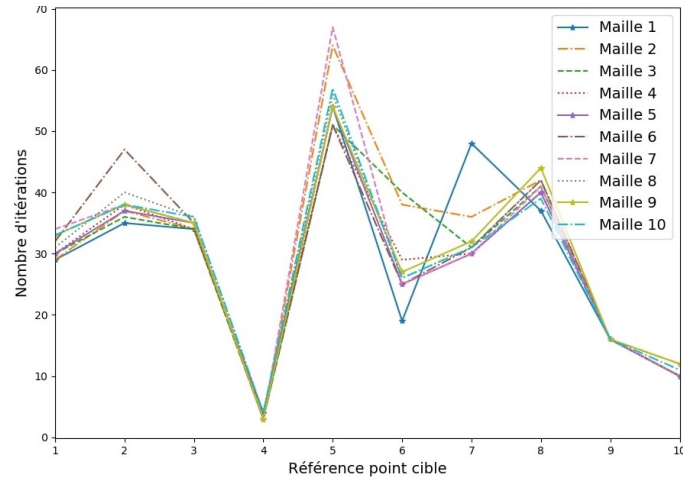


FIGURE 6.15 – Nombre d'itérations entre deux points-cible successifs, pour l'échantillon n° 3, en fonction de la taille de maille.

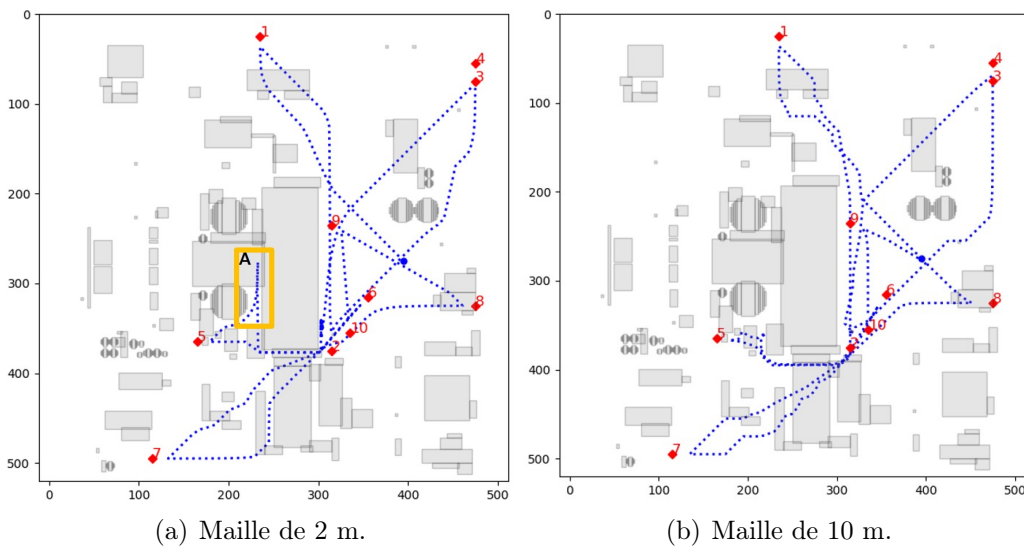


FIGURE 6.16 – Chemin suivi par le drone pour visiter les 10 points-cible de l'échantillon n° 3.

6.5.3 Limitations du modèle

Cette section synthétise les différentes limitations du modèle que nous avons identifiées par l'analyse des résultats de simulation.

Limitation de la zone de recherche dans le cas de l'anticipation d'évitement d'un obstacle

L'utilisation de la fonction d'anticipation d'évitement d'un obstacle limite la zone de calcul de chemin, une fois le premier *waypoint* sélectionné (comme expliqué dans la section 6.5.2, p. 113)

Manque de qualité du chemin dans certains cas

On estime qu'un chemin n'est pas de bonne qualité si des décisions prises, dans une zone déjà connue, allongent le chemin.

Les calculs locaux sont rarement globalement optimaux puisque les décisions sont prises, par définition, à partir d'informations locales. Ils sont tout de même souvent de bonne qualité, comme nous l'avons montré dans les sections précédentes. Néanmoins, nous avons observé 5 cas (sur les 300 chemins simulés, soit moins de 2% des cas) dans lesquels le chemin est de mauvaise qualité.

Dépendance du chemin au maillage

Le drone se dirige vers son point-cible tant qu'il n'a connaissance d'aucun obstacle entre lui et sa cible (CP1) ; il emprunte donc le chemin le plus court. En dehors de ce cas particulier, le calcul de chemin se fait par la sélection successive des mailles au plus petit potentiel, en cherchant dans les 8 mailles adjacentes en 2D, et dans les 26 mailles adjacentes en 3D.

Cependant, la discrétisation utilisée est à mailles carrées et le potentiel attractif relatif au point-cible est proportionnel à la distance euclidienne entre le centre de la maille et le point-cible. Ainsi, la sélection de *waypoints* successifs dans des mailles adjacentes ne permet pas, en l'état, de sélectionner le chemin le plus court (comme illustré en figure 6.17, p. 122).

Pour pallier cette limitation on peut intégrer un coût de déplacement vers les mailles diagonales.

Impossibilité d'emprunter le chemin le plus court dans des cas complexes

Dans le cas où plusieurs obstacles sont imbriqués en direction du point-cible, l'anticipation de l'évitement des obstacles ne permet pas toujours d'emprunter le chemin le plus court (cf. figure 6.18, p. 122).

Difficulté à sortir de l'enveloppe d'un obstacle concave aligné avec le point-cible

Lorsqu'un drone se trouve dans l'enveloppe d'un obstacle concave aligné avec son point-cible, mais ne représentant pas une impasse, alors le contourne-

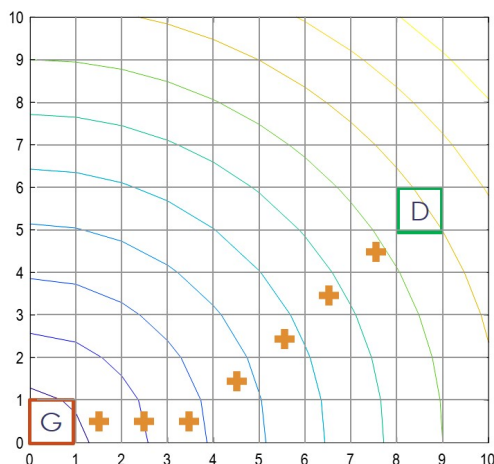


FIGURE 6.17 – Illustration de la dépendance du calcul de chemin au maillage. Chemin calculé par la sélection successive de *waypoints*, dans un champ attractif. Les cercles concentriques ayant pour centre le point-cible (noté G) représente le champ attractif qui lui est associé.



FIGURE 6.18 – Le chemin calculé par descente de gradient n'est pas le plus court possible (il aurait fallu passer entre les 2 obstacles).

ment de cet obstacle peut prendre un grand nombre d'itérations (cas illustré en figure 6.14(b), p. 119).

6.5.4 Conclusion sur la performance du modèle

Les 300 chemins simulés ont bien été réalisés, sans collision. Cependant, 5 chemins ont montré des performances décevantes (c'est-à-dire que les décisions prises, dans une zone déjà connue, allongent le chemin), cependant cela représente moins de 2% des chemins simulés. En dehors de ces quelques cas, les chemins calculés sont efficaces, bien qu'ils ne s'appuient que sur une connaissance partielle de l'environnement.

Les chemins de mauvaise qualité considéraient des mailles de 1, 2, 4 et 5 m. Les résultats associés à des discrétisations plus grossières ont toujours été

satisfaisants.

Nous pouvons en conclure que notre modèle est efficace malgré de rares manques de performance. De plus, on observe que si la mission consiste à observer un point, sans l'atteindre précisément, l'utilisation d'un maillage grossier offrira de meilleures performances. Néanmoins, si l'objectif est que le drone atteigne un point précis, en particulier proche d'un obstacle, alors un maillage fin sera nécessaire.

6.6 Résumé

Ce chapitre présentait une méthode de calcul de chemin basée sur le principe des champs de potentiel discrets. L'approche que nous proposons permet à un engin de se déplacer dans un environnement inconnu *a priori*. Les champs de potentiel répulsifs que nous considérons sont adaptés à la forme de chaque obstacle, et n'ont d'influence que sur la distance de sécurité à maintenir autour du drone. De plus, nous proposons un champ de potentiel discret et temporaire permettant au drone d'anticiper l'évitement d'un obstacle dès qu'il en a connaissance. Ces deux propositions permettent au drone de passer entre des obstacles proches, d'atteindre le point-cible même si celui-ci est proche d'un obstacle et limite considérablement les minima locaux, ce qui représente les trois principales faiblesses des méthodes utilisant les champs de potentiel existantes.

D'autre part, nous proposons de calculer les déplacements du drone sur plusieurs itérations, en répétant la descente de gradient. Cela permet de calculer des chemins plus courts car n'étant pas soumis à la forme du maillage. Cela permet surtout d'éviter au drone d'aller dans des zones non utiles (pour atteindre son point-cible), par la suppression des *waypoints* intermédiaires.

Nous avons testé notre méthode par la simulation de 300 chemins dans un environnement représentatif de l'utilisation considérée, à savoir un site protégé d'environ un quart de kilomètre carré de surface, contenant des obstacles fixes de formes et dimensions variées. Moins de 2% des chemins simulés étaient de mauvaise qualité. Nous avons montré que si la discrétisation était suffisamment large, notre méthode de calcul de chemin est performante.

Ce chapitre s'attachait à la présentation de l'algorithme pour une plateforme unique. Cependant, la méthode a été pensée pour être utilisée par les drones d'un essaim, évoluant dans une même zone, ce qui fait l'objet du chapitre suivant.

Chapitre 7

Calcul de chemin pour les drones d'un essaim

Sommaire

7.1	Introduction	125
7.2	Collaboration interplateforme	125
7.2.1	Principe	125
7.2.2	Informations partagées	125
7.2.3	Échange des informations	126
7.3	Adaptation du modèle pour plusieurs drones	126
7.3.1	Outils	126
7.3.2	Potentiel relatif à un drone voisin	127
7.3.3	Somme des potentiels	127
7.3.4	Algorithme	128
7.3.5	Exemple de calcul de chemin par des drones collaboratifs	128
7.3.6	Cas particulier : vol en formation compacte	129
7.4	Simulations et résultats	134
7.4.1	Paramètres de simulation	134
7.4.2	Résultats	135
7.5	Résumé	137

Dans ce chapitre nous présentons les ajouts à effectuer à la méthode de calcul de chemin présentée dans le chapitre précédent, permettant à plusieurs drones collaboratifs, évoluant dans un même environnement complexe, de l'utiliser. Il est important de noter que la méthode reste fondamentalement identique et que son utilisation dans un système multiplateforme représente une contrainte qui lui est ajoutée.

7.1 Introduction

On rappelle qu'un essaim est un ensemble de véhicules sans pilote à bord, partageant un objectif global commun. Chaque plateforme constituant l'essaim est autonome et a la capacité d'établir des liens de communication pair-à-pair avec les autres véhicules qui sont à sa portée. Ici, tous les véhicules constituant l'essaim sont des multirotors aux caractéristiques similaires, il s'agit donc d'un essaim homogène.

Lorsqu'un drone détecte d'autres membres de l'essaim, ceux-ci sont appelés « voisins ». Conformément aux paramètres définis dans le chapitre précédent (section 6.4.4, p. 111), la portée de détection des obstacles est inférieure à la portée de communication.

La méthode de calcul de chemin présentée dans le chapitre 6 (p. 91), basée sur le principe des champs de potentiels, a été conçue dans le but d'être utilisée par les drones d'un essaim. Pour des raisons de clarté, le chapitre précédent exposait la méthode et sa validation, pour un drone, et nous abordons ici les aspects propres à l'utilisation en multiplateforme.

Comme expliqué précédemment (section 6.3.1, p. 101), le fait d'associer chaque contrainte considérée à un champ de potentiel et de calculer un chemin à partir de la somme de ces champs, permet d'étendre la méthode à de nouvelles contraintes. C'est ce que nous proposons ici, afin d'utiliser le modèle dans un système multiplateforme collaboratif.

Nous supposons que tous les drones considèrent la même zone d'intérêt et que celle-ci est discrétisée de manière identique pour tous.

7.2 Collaboration interplateforme

7.2.1 Principe

Un drone seul, disposant uniquement des informations d'environnement acquises par son capteur, peut calculer un chemin dans un environnement complexe. Cependant, plus sa connaissance de l'environnement est complète, plus le chemin sera court.

Lorsqu'un essaim est considéré, chaque drone autonome analyse l'environnement qui l'entoure. La collaboration interplateforme se fait par le partage des informations relatives à l'environnement, détectées par chacun des drones. Donc plus il y a de drones, plus le chemin peut être court.

7.2.2 Informations partagées

Les drones envoient tour à tour, à l'ensemble de leurs voisins, la vision de l'environnement la plus globale qu'ils possèdent. La connaissance de la présence d'un obstacle est aussi importante que la connaissance de son absence. C'est

pourquoi les drones n’envoient pas uniquement la position des obstacles qu’ils ont détectés, mais l’ensemble de leur tableau `obs` (contenant l’ensemble des informations acquises sur la présence et l’absence d’obstacles, décrit en section 6.2.2, p. 94).

Puisque les obstacles considérés ici sont fixes, les informations n’ont pas de durée de validité. Lorsqu’un drone reçoit le tableau `obs` d’un de ses voisins, il fusionne les informations qu’il contient avec celles déjà contenues dans son propre tableau, sans distinction d’origine (capteur ou voisin). Ainsi, lorsqu’un drone partage son tableau `obs`, celui-ci contient la fusion des informations qu’il a lui-même découvertes et des informations reçues de ses voisins. Les drones peuvent ainsi disposer d’informations sur des zones qu’ils n’ont pas explorées.

Il est important de noter que seules des informations d’environnement sont partagées et chaque drone en déduit les champs de potentiel associés, en fonction de sa propre position et de celle de son point cible. Ainsi, les drones formant l’essaim peuvent avoir des points cible différents, ainsi que des caractéristiques différentes (portée du capteur de détection d’obstacle, portée de communication, vitesse maximale, etc.).

7.2.3 Échange des informations

Comme expliqué dans la section 3.5.3 (p. 33), nous choisissons d’aborder une approche tour à tour où les messages sont envoyés à destination de ses voisins (directs) de l’émetteur. Ainsi, un message envoyé par un drone ne sera pas nécessairement reçu par l’ensemble des membres de l’essaim. Néanmoins, si deux drones sont trop éloignés pour établir un lien de communication, alors ils n’ont pas d’utilité immédiate des informations d’environnement possédées par l’autre.

Comme tous les voisins considèrent le même environnement, discrétisé de la même manière, leurs tableaux `obs` sont de mêmes dimensions et chacune des mailles qui les composent sont associées à la même zone de l’environnement. Lorsqu’un drone reçoit le tableau `obs` d’un voisin, il fusionne donc les informations qu’il contient avec son propre tableau `obs` par comparaison de la valeur prise par chaque élément.

7.3 Adaptation du modèle pour plusieurs drones

7.3.1 Outils

Les outils utilisés dans cette adaptation pour système multiplateforme sont les mêmes que pour le modèle pour un drone seul. Cependant, un type d’information supplémentaire doit être considéré afin d’éviter les collisions : la position des drones voisins. Ces données seront stockées, sous la forme de potentiels, dans le tableau `avoid`, puisque c’est ce tableau-ci qui enregistre les

informations temporaires. On rappelle que le tableau `avoid` est mis à jour à chaque nouveau calcul de chemin.

7.3.2 Potentiel relatif à un drone voisin

Dans cette étude, les seuls obstacles mobiles considérés sont les autres drones de l'essaim. Chaque drone représente en effet un obstacle pour chacune des autres plateformes de l'essaim, d'autant plus lorsque les drones se dirigent vers des points cible différents.

Comme nous considérons que la distance de sécurité à maintenir autour d'un drone est égale à la largeur d'une maille, le potentiel répulsif relatif à un voisin s'étend à une distance d'une maille autour de celui-ci. On note v_{ijk} la maille contenant un voisin, située en ligne v_i , colonne v_j et altitude v_k . Le potentiel répulsif lié à un voisin est défini comme suit :

$$\forall m_{ijk} \mid \{ (i, j, k) \in \llbracket v_i - 1, v_i + 1 \rrbracket \times \llbracket v_j - 1, v_j + 1 \rrbracket \times \llbracket v_k - 1, v_k + 1 \rrbracket \},$$

$$p_{ijk}^v = c_{\text{obs}}, \quad (7.1)$$

où c_{obs} est la constante utilisée dans la définition du potentiel lié aux obstacles fixes.

On remarque que le potentiel associé aux mailles entourant un voisin est égal à c_{obs} , contrairement à celui lié à l'évitement d'obstacles fixes, qui vaut $\frac{c_{\text{obs}}}{2}$. En effet, une collision entre deux drones entraînerait la perte des deux plateformes. En revanche, les obstacles fixes considérés ici sont des bâtiments, les risques associés à une collision avec un multirobot se résument, dans la grande majorité des cas, à la perte du drone. Ce choix de définir une répulsion supérieure pour les drones voisins que pour les obstacles fixes permet donc de tolérer l'approche d'obstacles fixes si elle constitue la seule alternative à l'évitement d'un drone.

Il est à noter que dans notre modèle, les drones se déplacent tour à tour ; au cours d'une itération les plateformes se déplacent successivement, l'une après l'autre. Dans ce contexte un champ de potentiel d'une maille de large (soit la distance de sécurité) est adapté. Si on souhaite s'affranchir de la synchronisation des plateformes, on peut envisager d'adapter le potentiel répulsif lié à un voisin à la vitesse maximale de ce dernier.

7.3.3 Somme des potentiels

Le potentiel total considéré pour le calcul de chemin pour un drone seul est la somme des potentiels stockés dans les tableaux `pot` et `avoid` (cf. équation 6.5, p. 101). Dans le cadre d'une adaptation multidrone, au delà des champs

de potentiel relatifs à la cible et aux obstacles fixes, nous considérons le champ associé aux voisins. Ce dernier étant stocké dans le tableau `avoid`, la définition du potentiel total ne change pas.

7.3.4 Algorithme

Le principe de notre méthode dynamique de calcul de chemin pour drones en essaim, que nous nommons *Peach*, est décrit par l'algorithme 8 (p. 128).

Algorithme 8 Principe de la méthode *Peach*.

```
tant que le drone n'a pas atteint son point cible faire
  si un tableau obs a été reçu alors
    Fusionner le tableau obs reçu avec l'existant ;
  fin si
  Détecter les obstacles fixes et les voisins ;
  si de nouveaux obstacles fixes ont été détectés alors
    Mettre à jour les informations d'environnement (tableau obs) ;
    Mettre à jour les champs de potentiels répulsifs (tableau pot) ;
  fin si
  si des voisins ont été détectés alors
    Mettre à jour les champs de potentiels répulsifs temporaires (tableau
      avoid) ;
  fin si
  si un voisin a été détecté sur le chemin ou de nouveaux obstacles ont été
    détectés sur le chemin ou le drone détecte qu'il est dans une impasse ou le
    drone n'a connaissance d'aucun obstacle entre lui et son point cible alors
    Calculer un nouveau chemin ;
  fin si
  Se déplacer en suivant le chemin calculé ;
  si c'est au drone d'envoyer son tableau obs alors
    Diffuser le tableau obs ;
  fin si
fin tant que
```

7.3.5 Exemple de calcul de chemin par des drones collaboratifs

La figure 7.1 (p. 130) représente les chemins empruntés par 2 drones collaboratifs, calculés par le modèle *Peach*, sur plusieurs itérations. Chaque colonne illustre la vision de l'environnement d'un des drones, le potentiel total associé à chaque maille (p_{ijk}^{tot}) ainsi que les *waypoints* qu'il a calculés. L'environnement

représenté mesure 120 m de long et 50 m de large, et les mailles ont une largeur de 1 m. La vitesse maximale considérée est de 10 m.s^{-1} et la portée de détection d'obstacles est de 24 m. La portée de communication est de 100 m et le débit est de 1 Mbps.

Dans cet environnement en deux dimensions, 6000 mailles sont considérées ($\frac{120 \text{ m}}{1 \text{ m}} \times \frac{50 \text{ m}}{1 \text{ m}} = 6000$). Comme les éléments du tableau `obs` sont des entiers prenant une valeur parmi 4 (décrites en section section 6.2.2, p. 94), on peut les coder sur 2 bits. Le tableau `obs` à transmettre ici a donc une taille de 12 kbits. Compte tenu du débit du module de communication, l'envoi du tableau représentant cet environnement nécessite environ 12 ms. Comme nous discrétisons le temps en itérations d'une seconde, et qu'au plus un message est envoyé par itération, chaque drone diffusera son message toutes les deux itérations, alternativement.

À la première itération, les deux drones considérés ici découvrent la zone indépendamment l'un de l'autre (figures 7.1(a) et 7.1(b), p. 130). Après leurs calculs de chemin respectifs (tous les deux dans le cas particulier CP1), le drone n° 0 (figure 7.1(a)) envoie ses informations.

Durant l'itération 1, le drone n° 1 détecte qu'il est dans une impasse et commence à retourner sur ses pas suivant le cas particulier CP2 (non illustré). À la fin de l'itération 1, le drone n° 1 diffuse son tableau `obs` et le drone n° 0 fusionne les informations reçues avec celles dont il a déjà connaissance.

Lors de l'itération 2, le drone n° 0 a donc connaissance des obstacles en face de lui, calcule le champ de potentiel répulsif d'anticipation d'évitement, et calcule un chemin qui contourne l'impasse avant même d'y être entré (figure 7.1(c), p. 130). Le drone n° 1 suit son chemin déjà calculé (figure 7.1(d), p. 130). À la fin de l'itération, le drone n° 0 diffuse son tableau `obs`.

À la troisième itération, le drone n° 1 continue de suivre son chemin (il a bien détecté un voisin, mais celui-ci ne se trouve pas sur son chemin). Par contre le drone n° 0 le détecte sur le chemin qu'il avait précédemment calculé, il renouvelle donc son calcul en évitant son voisin.

Les obstacles considérés pour l'anticipation de l'évitement, lors de l'itération n° 8, ne sont pas les mêmes pour les deux plateformes, compte tenu de la différence de chemin emprunté précédemment. Ainsi, le drone n° 1 ne peut pas emprunter le chemin le plus court (entre les deux obstacles), car les potentiels liés à l'anticipation de chacun des deux obstacles se superposent (limitation du modèle identifiée dans le chapitre précédent, section 6.5.3, p. 120), contrairement au drone n° 0 qui peut passer entre ces deux bâtiments..

7.3.6 Cas particulier : vol en formation compacte

Dans certaines missions (par exemple l'identification d'une cible) il peut être utile de traiter des images d'une même cible provenant différentes sources d'information (par exemple une caméra infrarouge et un radar). La corrélation

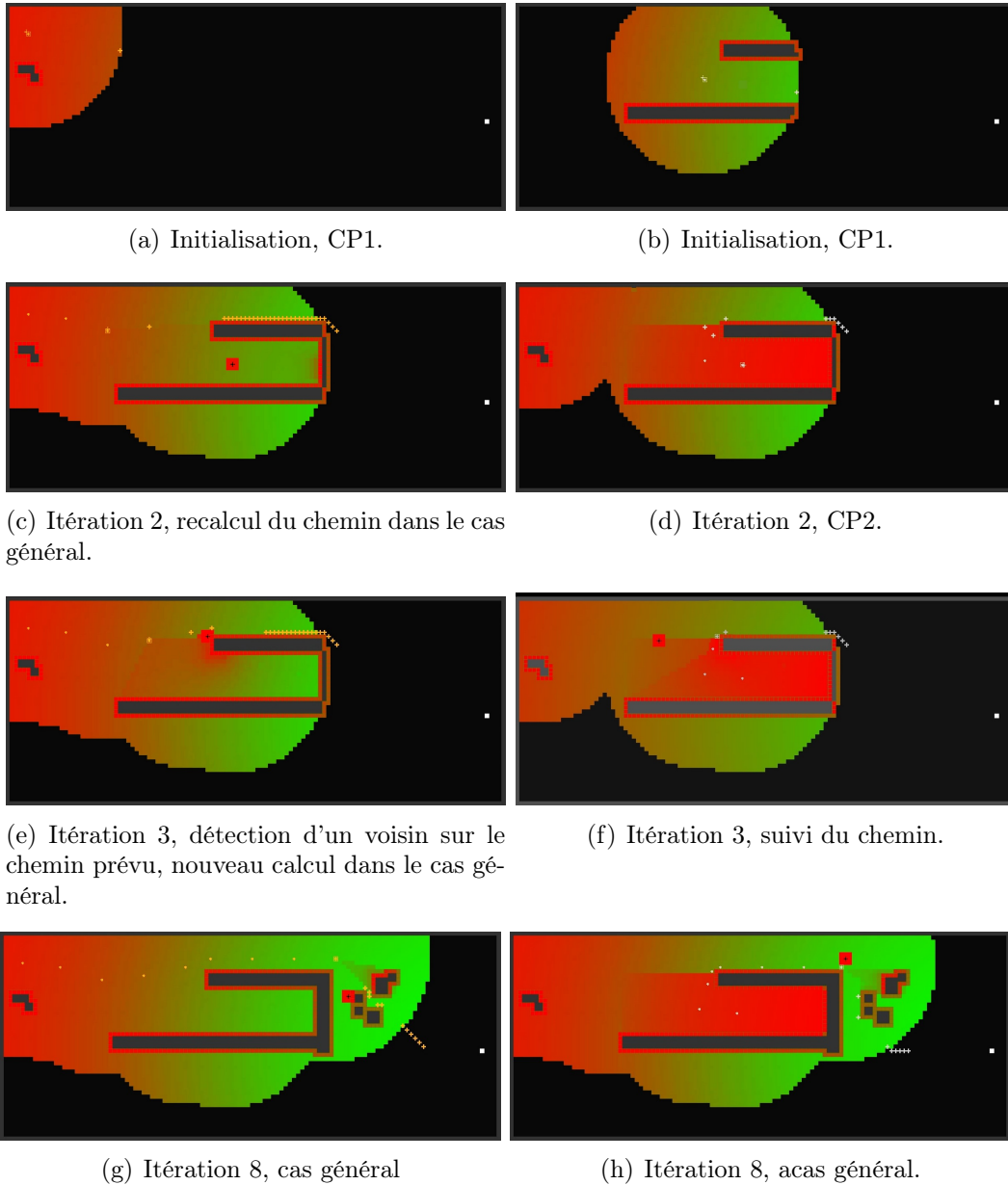


FIGURE 7.1 – Chemins calculés par 2 drones collaboratifs. La colonne de gauche (resp. droite) représente le point de vue du drone n° 0 (resp. n° 1) pour quelques itérations. Les petites croix représentent les mailles où le drone s'est arrêté à la fin d'une itération, les grandes représentent les *waypoints* calculés. Les mailles inconnues sont en noir, celles contenant un obstacle en bleu foncé, et la couleur des mailles restantes est représentative du potentiel qui leur est associé (du plus faible en vert clair au plus élevé en rouge). La maille contenant le point cible est blanche.

des données provenant des différents capteurs peut être simplifiée si les angles de vue sont similaires. La création et le maintien d'une formation compacte répond à ce besoin.

Lorsque l'on souhaite utiliser les données provenant de plusieurs capteurs embarqués sur différentes plateformes, un lien de dépendance apparaît entre les drones en question. Dans ce cas, il devient donc acceptable de limiter l'autonomie de certaines plateformes au profit d'une autre. Plus précisément, dans le contexte de vol en formation compacte, on peut utiliser un système meneur-suiveurs ; le meneur est alors le seul drone autonome.

Notre modèle de mobilité est utilisable directement par un meneur dans le cadre d'un vol en formation compacte : il suffit que le meneur prenne en compte, dans le calcul du chemin, la dimension de la formation au lieu de la taille du drone, sans rien changer à la méthode de calcul. Lorsqu'un meneur calcule un chemin pour une formation compacte, il est possible qu'il considère une taille de maille supérieure à celle associée à ses déplacements lorsqu'il est seul.

À la fin de chaque itération, le meneur envoie aux suiveurs le prochain déplacement qui devra être réalisé¹, identique pour tous les drones. Tous les drones de la formation compacte se déplacent donc, à chaque itération, selon le même vecteur vitesse. Ainsi, la formation de départ est maintenue.

Dans le cadre de *Peach*, un drone considère qu'il est dans la maille contenant le centre de sa structure. Puisque le meneur suit le modèle en considérant la formation compacte comme un seul engin, il considère que la formation se trouve dans la maille contenant son barycentre. Comme on cherche à maintenir une formation établie il est préférable que le meneur soit au centre de celle-ci pour qu'il ait la meilleure vision de l'environnement, indépendamment de la direction de déplacement.

La formation compacte est établie au début du trajet, puis maintenue. La configuration s'appuie sur les travaux menés dans le cadre du modèle de mobilité basé sur les forces virtuelles (chapitre 5, p. 62). Nous utilisons ici aussi une distance de référence, D_{ref} , à maintenir entre les drones de la formation, définie comme dans le chapitre 5 (équation 5.2, p. 66) comme suit :

$$D_{\text{ref}}(k) = k_{\text{compact}} \times \min(2p_{\text{capteur}}, p_{\text{comm}})$$

où p_{capteur} désigne la portée du capteur permettant de détecter des obstacles, p_{comm} la portée du module radio et k_{compact} est un réel strictement positif. Pour établir une formation où les drones sont aussi proches que possible, il faut identifier la valeur de k_{compact} la plus faible permettant d'éviter les collisions.

1. On considère qu'une formation compacte est de dimension très inférieure à la portée de communication des modules radio embarqués. Par conséquent, tous les suiveurs sont des voisins directs du meneur.

Dans la formation compacte, on considère (comme dans le chapitre 5) que les drones doivent être éloignés d'au moins leur diamètre (50 cm). De plus On considère $p_{\text{capteur}} = 24 \text{ m}$ et $p_{\text{comm}} = 100 \text{ m}$. Si on évalue, comme dans le chapitre 5, que la distance à maintenir entre 2 drones au sein de la formation est égale à leur diamètre, alors il faut avoir $k_{\text{compact}} \geq 0,13$ pour que la distance de référence soit supérieure à la distance de sécurité.

On note d_d le diamètre d'un drone. Le plus petit diamètre de l'essaim en formation compacte, c'est-à-dire avec $k_{\text{compact}} = 0,13$, le diamètre de l'essaim, noté d_f , dépend du nombre de drones, de la taille d'un drone et de la distance de sécurité dans la formation. Il est illustré en figure 7.2 (p. 132).

La figure 7.2 illustre les formations d'au plus 7 drones. Si davantage de drones doivent être en formation, on pourra étendre cette structure en 3 dimensions.

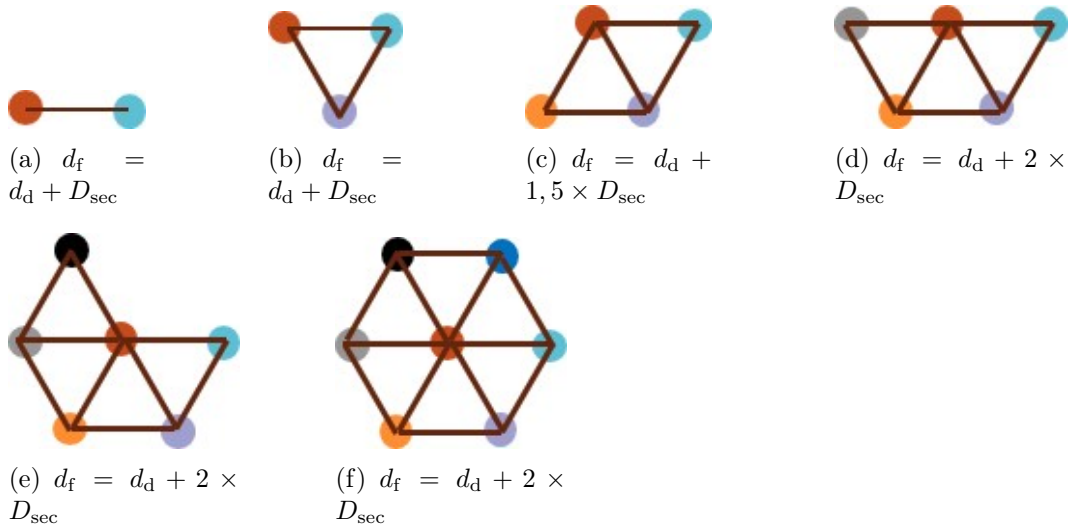


FIGURE 7.2 – Diamètre de la formation compacte pour des configurations de 2 à 7 drones. Le meneur est représenté par un disque orange foncé, les autres drones par des disques d'autres couleurs. La distance de référence D_{ref} est représentée par des segments marron.

Un déplacement en formation compacte peut aussi s'avérer utile lorsque plusieurs drones ont à effectuer un même trajet dans un environnement contraint. La figure 7.3 (p. 133) illustre les déplacements de 4 drones partant d'une même zone et visant le même point cible, pour 2 itérations (n^{os} 1 et 9). On remarque que le potentiel répulsif lié à un voisin (illustré par un carré rouge de 3 mailles de côté) n'entoure pas nécessairement le voisin en question, car ce potentiel n'est calculé que si un voisin se trouve sur le chemin du point cible.

La figure 7.4 (p. 134) représente le chemin équivalent, dans le même environnement, mais dans le cadre d'un vol en formation compacte.

7. Calcul de chemin pour les drones d'un essaim

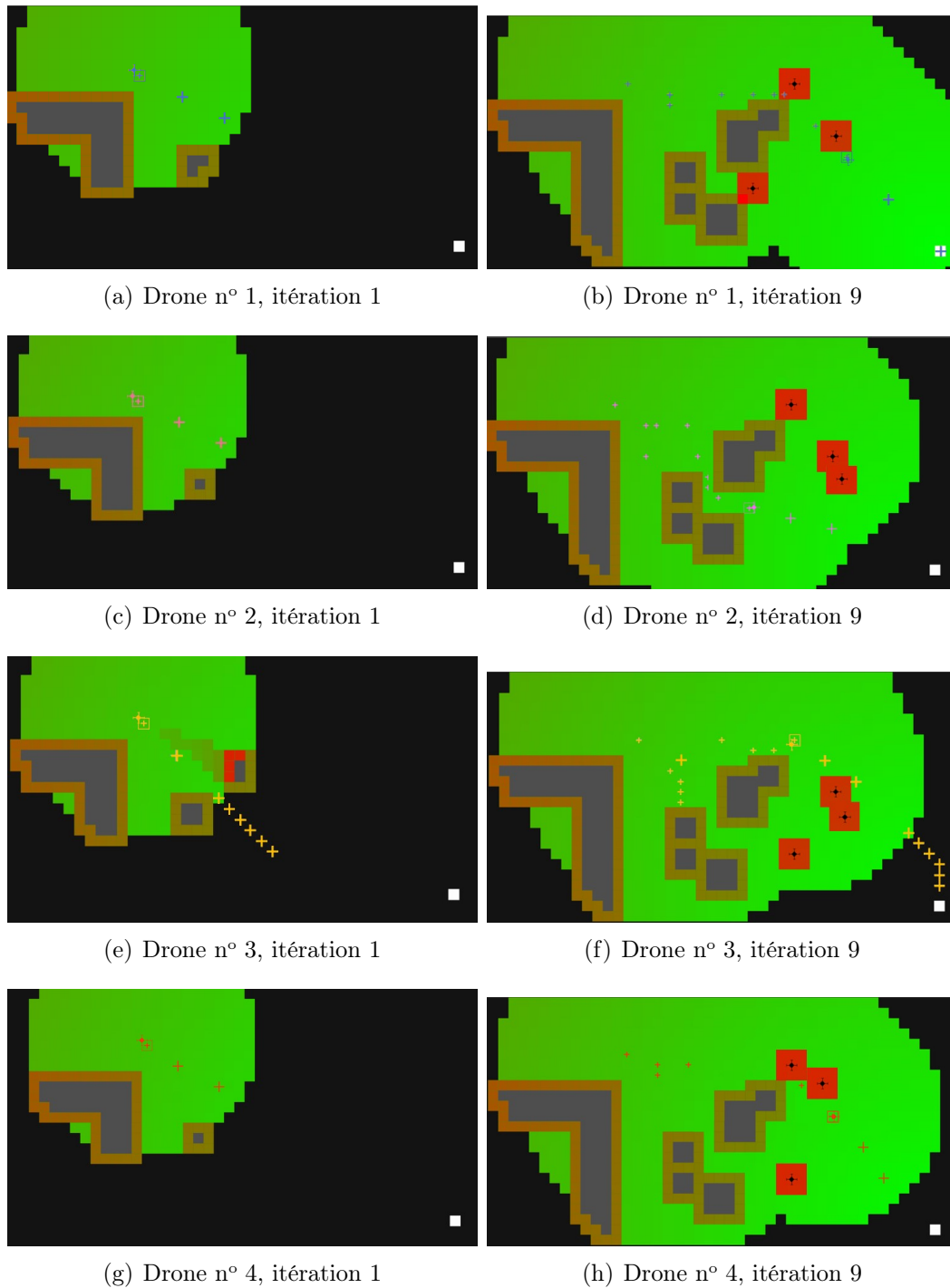


FIGURE 7.3 – Déplacements de 4 drones collaboratifs, à l'itération n° 1 (colonne de gauche) et à l'itération n° 9 (colonne de droite). Les drones voisins sont schématisés en noir.

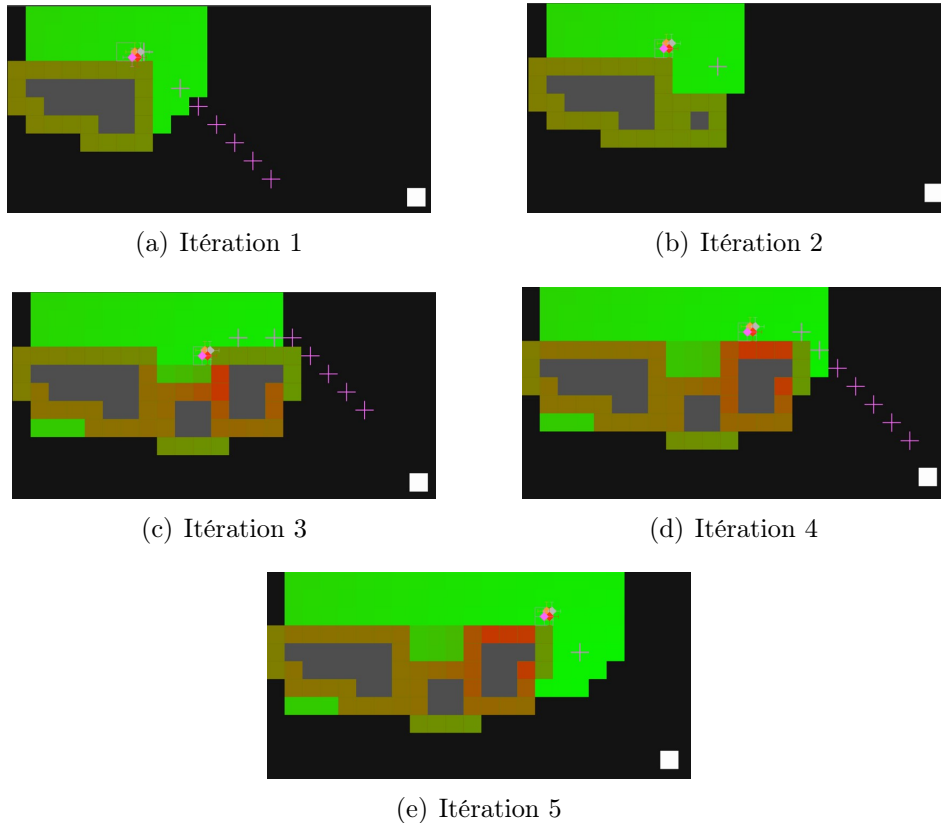


FIGURE 7.4 – Déplacements de 4 drones en formation compacte, dans le même environnement que celui de la figure 7.3 (p. 133). La maille a été agrandie par le meneur car le diamètre de la formation était supérieur à la taille de maille initiale.

7.4 Simulations et résultats

7.4.1 Paramètres de simulation

L'environnement de simulation utilisé ici est identique à celui présenté dans le chapitre précédent (section 6.4.3, p. 109). Nous rappelons brièvement qu'il contient des obstacles aux formes et dimensions variées et qu'il mesure 500 m de large, 52 m de long, 60 m de haut.

Nous considérons un essaim homogène de multicopters de 50 cm d'envergure, pouvant atteindre une vitesse maximale de 10 m.s^{-1} . En plus du capteur de détection d'obstacles d'une portée de 24 m, les drones sont équipés d'un module de communication leur permettant d'établir des liens pair-à-pair avec leurs voisins éloignés d'au plus 100 m, avec un débit de 1 Mbps (performances du module Wi-Fi Intel Dual Band Wireless-AC présenté, en section 3.7, p. 35).

Nous supposons qu'au début de la simulation, les drones sont synchronisés et connaissent le nombre de membres de l'essaim.

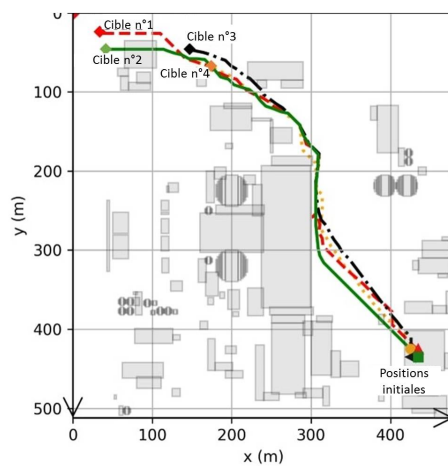
7.4.2 Résultats

Trajet en direction de différents points cible

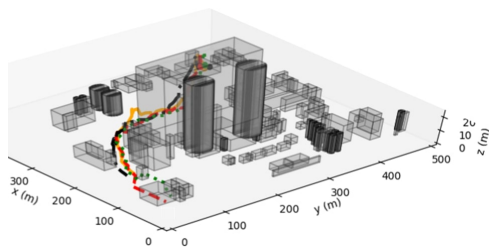
Puisque les seules informations partagées entre les drones sont relatives à l'environnement, les plateformes peuvent collaborer indépendamment du point cible qu'elles ont à atteindre. Néanmoins, si des drones évoluent dans des zones trop éloignées pour communiquer, ils ne pourront pas collaborer.

Nous présentons ici un exemple de chemins suivis par des drones devant se diriger vers différents points-cible, afin d'illustrer cette possibilité qu'offre le modèle. La figure 7.5 illustre les chemins empruntés par les 4 drones d'un essaim, ayant chacun à atteindre un point cible distinct, avec une discrétisation horizontale de 4 m. La discrétisation verticale, comme dans le chapitre précédent, est de 4 m, et les drones peuvent évoluer entre 4 m et 24 m d'altitude.

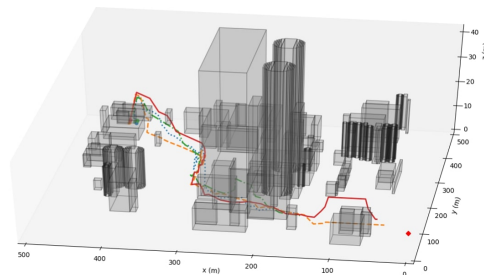
Il n'y a eu aucune collision, ni avec un obstacle fixe, ni entre drones, et chacun d'entre eux a atteint son point cible.



(a) Vue de haut.



(b) Vue en 3D.



(c) Vue en 3D.

FIGURE 7.5 – Chemins suivis par les 4 drones d'un essaim, pour atteindre 4 points cible distincts.

Trajets vers un point cible commun

Nous avons aussi simulé des essais de 3, 5, 7 et 9 drones ayant à atteindre un même point cible, avec des discrétisations horizontales de 2, 4 et 6 m. Cela représente, au total, le calcul de 72 chemins. Nous considérons ici des simulations en 2D, afin de contraindre les drones à évoluer sur le même plan, ce qui complexifie l'évitement interplateforme.

Les figures 7.6(a) à 7.6(c) (p. 136) illustrent les chemins des drones composant des essaims de 3, 5, 7 et 9 drones, avec différentes discrétisations. Les distances parcourues en fonction de la discrétisation sont illustrées en figure 7.7 (p. 137).

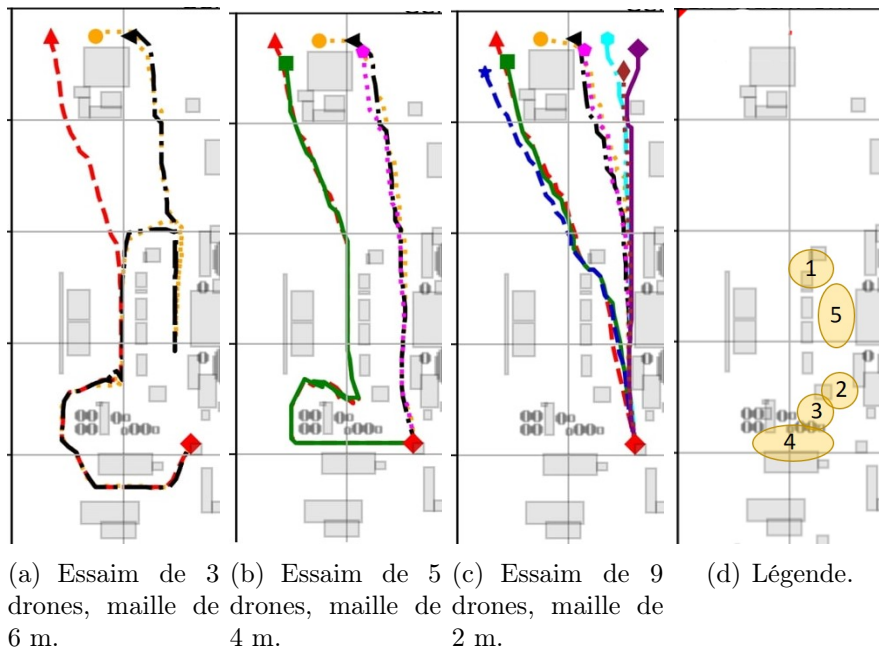


FIGURE 7.6 – Vue de haut des chemins suivis par les 3, 5 ou 9 drones d'un essaim, pour atteindre un même point cible (symbolisé par un carré rouge) avec plusieurs discrétisations.

Les chemins calculés avec une discrétisation de 2 m sont significativement plus courts que ceux associés à des maillages plus grossiers, car la discrétisation la plus fine autorise le passage à la fois dans les zones 1 et 2 (d'après les notations de la figure 7.6(d), p. 136).

La discrétisation de 4 m permet de traverser les zones 2 et 4, ce qui permet d'obtenir un chemin plus court qu'avec un maillage de 6 m. Dans le cas de mailles de 6 m, plusieurs drones entrent dans la zone 5, mais ont à faire demi-tour lorsqu'ils détectent qu'il ne leur est pas possible de traverser la zone 2; le chemin de ces drones jusqu'à la cible est de l'ordre de 900 m. Pourtant, ces drones, calculent un chemin aussi court que possible compte tenu des informa-

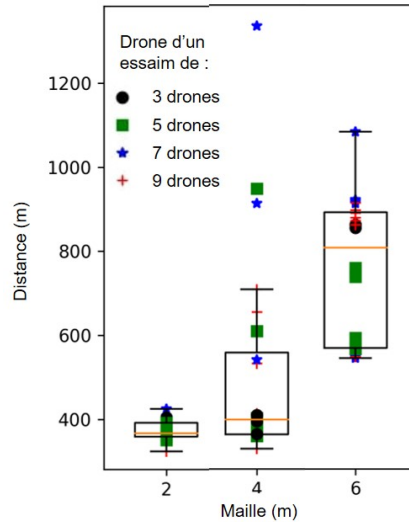


FIGURE 7.7 – Superposition des distances parcourues par chaque plateforme, dans la situation exposée en figure 7.6, p. 136, pour des essaims de 3, 5, 7 et 9 drones, en fonction de la taille de maille.

tions locales dont ils disposent.

Pour chaque taille de maille, les trajets les plus longs sont réalisés par des essaims de 7 ou 9 drones. Cela est dû au fait que lorsque plusieurs d’entre eux se retrouvent dans des zones complexes, il faut éviter les voisins en plus des obstacles fixes, ce qui augmente la longueur du chemin

Lorsque les drones partent d’une même zone pour atteindre un même point, ils disposent d’informations similaires. C’est pourquoi la collaboration inter-plateforme ne suffit pas à augmenter la performance du système global.

7.5 Résumé

Dans le chapitre 6 (p. 91), nous avons présenté une méthode de calcul de chemin pour un drone autonome en environnement complexe. Dans ce chapitre-ci nous avons introduit les extensions permettant de l’utiliser par plusieurs drones formant un essaim. Les drones de l’essaim collaborent entre eux par le partage de leur vision de l’environnement, permettant ainsi à leurs voisins d’étendre leur propre vision, et de calculer des chemins plus efficaces.

Conclusion

L'objectif de cette thèse était de permettre à un essaim de multirobots autonomes d'évoluer de manière collaborative dans un environnement complexe. Dans ce chapitre, nous résumons nos contributions et proposons des axes de recherches futures qui en découlent.

Contributions

Contexte et état de l'art

Le premier chapitre (p. 124) de ce manuscrit porte sur les systèmes sans pilote à bord au sens large. Nous y avons explicité les principaux termes relatifs à ce domaine, en se concentrant sur les spécificités de notre étude : les véhicules aériens, l'autonomie et les interactions interplateformes. Nous avons présenté les enjeux et applications des systèmes sans pilote à bord et en particulier des systèmes multiplateforme aériens.

Le chapitre 2 (p. 18) introduit des définitions sur les graphes, objets auxquels on peut assimiler les systèmes multiplateformes, chaque nœud étant associé à un véhicule, et chaque arête à un lien de communication. Nous présentons également dans ce chapitre les principaux simulateurs de réseaux existants et détaillons les critères nous ayant conduit à utiliser de JBotSim et OMNeT++.

Dans le chapitre 3 (p. 25) nous exposons le besoin opérationnel auquel les travaux de cette thèse doivent répondre et caractérisons les contraintes qui en découlent. Ces contraintes impactent plusieurs aspects du système multiplateforme : le choix des drones, la définition de l'architecture de contrôle, la méthode de diffusion des informations au sein de l'essaim et les caractéristiques des capteurs et modules de communications embarqués. La mobilité des drones autonomes, est une problématique commune à toutes les missions à réaliser. Nous axons donc les travaux sur les méthodes de calculs de chemin pour drones autonomes, en environnements complexes.

Le chapitre 4 (p. 39) présente l'état de l'art des méthodes de calcul de chemin et de trajectoire adaptées aux véhicules sans pilote à bord, et en particuliers aux essaims de drones. Nous introduisons les différentes approches selon leur caractère local ou global, puis présentons des associations de plusieurs méthodes existantes. Deux types d'approches sont retenues pour nos travaux, basées sur les forces virtuelles et sur les champs de potentiel.

Calcul de déplacements à partir de forces virtuelles

Le chapitre 5 (p. 62) introduit un modèle de mobilité permettant à des drones autonomes en essaim de remplir une mission, consistant en la surveillance successive de deux zones d'intérêt séparées par un environnement complexe. Le modèle considère trois types de forces : l'attraction-répulsion entre les drones, qui permet l'évitement des collisions entre les plateformes et favorise le maintien de distance de référence autorisant les communications ; l'attraction vers un point cible, régulièrement mis à jour, qui permet aux drones de se déplacer afin de remplir la mission ; la répulsion liée aux obstacles fixes permettant l'évitement des collisions.

L'association de ces trois types de force et la sélection de points cible adéquats permet d'établir un modèle de mobilité s'adaptant à toutes les phases du scénario considéré. Cette étude a permis de réaliser une analyse initiale des contraintes mises en jeu dans le contexte d'une mission menée par un tel système.

Calcul de chemin par le modèle *Peach*

Les chapitres 6 et 7 (respectivement p. 91 et p. 124) présentent la méthode de calcul de chemin pour drones autonomes collaboratifs que nous proposons. Le chapitre 6 décrit notre méthode de calcul de chemin, basée sur le principe des champs de potentiel discrets, permettant à un drone de se déplacer dans un environnement *a priori* inconnu. Deux faiblesses majeures des méthodes de champs de potentiel classiques sont l'impossibilité de passer entre des obstacles proches et l'impossibilité d'atteindre un point cible situé à proximité d'un obstacle. Notre méthode résout ces deux problèmes en définissant des champs répulsifs liés aux obstacles adaptés aux formes de ceux-ci.

De plus, nous proposons l'utilisation d'un champ de potentiel temporaire permettant d'anticiper l'évitement des obstacles dès que ceux-ci sont connus et de limiter considérablement la présence de minima locaux (autre faiblesse usuelle des méthodes utilisant les champs de potentiel). De plus, nous proposons de calculer un chemin et pas seulement le prochain déplacement du drone, afin de s'affranchir des limitations induites par le maillage.

Nous proposons de calculer les déplacements du drone sur plusieurs itérations, ce qui permet, d'une part, de s'affranchir de la dépendance au maillage, et d'autre part, d'éviter au drone de se déplacer dans des zones qui ne lui sont pas utiles pour atteindre son point cible. Ainsi, les chemins calculés sont raccourcis.

Enfin, le chapitre 7 (p. 124) introduit les extensions à ajouter au modèle monoplateforme pour une utilisation par un essaim. Dans ce modèle, que nous nommons *Peach*, les drones de l'essaim se considèrent les uns les autres comme des obstacles mobiles. De plus, les drones collaborent entre elles en diffusant

régulièrement leur vision de l'environnement, composée de la fusion des informations qu'ils ont eux-mêmes acquises et de celles envoyées par des voisins.

Pour résumer, nos travaux permettent à des drones en essaim de se déplacer dans un environnement complexe. Chaque drone considéré est autonome et les plateformes collaborent afin d'augmenter les performances du système global.

Travaux futurs

Les travaux que nous avons menés nous ont permis de définir un certain nombre de développements et d'extensions que nous souhaiterions mettre en œuvre et étudier.

Tout d'abord, nous envisageons de réaliser un démonstrateur physique permettant de tester notre modèle de mobilité en conditions réelles. Cela permettrait également de tester le modèle dans le cadre de scénarii complets.

Nous souhaitons aussi développer un maillage adaptatif, dont la finesse dépendrait de la proximité des obstacles. Cela permettrait une simplification des calculs dans les zones dégagées, tout en autorisant le passage entre les obstacles proches.

Enfin, nous souhaitons étendre le modèle *Peach* à de nouvelles contraintes, telles que le maintien de communication entre certains drones de l'essaim et la prise en compte de capteurs aux caractéristiques différant entre les plateformes. Nous cherchons également à intégrer dans le modèle la cinétique des plateformes, et en particulier de celles à voilure fixe.

Liste des sigles

- 3D** *Dull, Dirty and Dangerous.* 13
- APF** *Artificial Potential Field.* 51–54, 56
- C2** *Command and Control.* 6
- CARUS** *Cooperative Autonomous Reconfigurable UAV Swarm.* 15
- CAS** *Close Air Support.* 16
- CE** *Commission Européenne.* 6
- DoD** *Department of Defense.* 7
- DWA** *Dynamic Window Approach.* 45, 46, 57
- FANET** *Flying Ad hoc Network.* 11, 36
- GNRON** *Goal Non Reachable with Obstacle Nearby.* 53, 55, 108
- HALE** *Haute Altitude Longue Endurance.* 1, 8, 13, 28
- ISR** *Intelligence, Reconnaissance and Surveillance.* 16
- MALE** *Moyenne Altitude Longue Endurance.* 13, 16, 28
- MANET** *Mobile Ad hoc Network.* 11, 32
- NIST** *National Institute of Standards and Technology.* 6
- OACI** *Organisation de l'Aviation Civile Internationale.* 7, 8
- PoI** *Point of Interest.* 15
- PSO** *Particle Swarm Optimization.* 51, 57
- RPAS** *Remotely Piloted Aircraft System.* 8
- SEAD** *Suppression of Enemy Aerial Defenses.* 16
- UAS** *Unmanned Aerial System.* 6–8
- UAV** *Unmanned Aerial Vehicle.* 7, 8, 58
- UMS** *Unmanned System.* 6, 7, 26
- UV** *Unmanned Vehicle.* 26
- VLOS** *Visual Line Of Sight.* 9

Annexes

Annexe A

Publications réalisées dans le cadre de la thèse

Dans le cadre de cette thèse, trois publications ont été validées dans des comptes rendus de conférences avec comité de relecture :

FALOMIR, Ema, CHAUMETTE, Serge et GUERRINI, Gilles, “A 3D Mobility Model for Autonomous Swarms of Collaborative UAVs”. Pages 196-204 dans *2019 International Conference on Unmanned Aircraft Systems (ICUAS)*, juin 2019. ISSN : 2575-7296, 2373- 6720, doi : 10.1109/ICUAS.2019.8798199, <http://ieeexplore.ieee.org/document/8798199>

FALOMIR, Ema, CHAUMETTE, Serge et GUERRINI, Gilles, “A Mobility Model Based on Improved Artificial Potential Fields for Swarms of UAVs”. Pages 8499-504 dans *2018 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, octobre 2018. ISSN : 2153-0866, 2153-0858. doi : 10.1109/IROS.2018.8593738. <http://ieeexplore.ieee.org/document/8593738>

FALOMIR, Ema, CHAUMETTE, Serge et GUERRINI, Gilles, “Mobility Strategies based on Virtual Forces for Swarms of Autonomous UAVs in Constrained Environments”. Pages 221-29 dans *Proceedings of the 14th International Conference on Informatics in Control, Automation and Robotics (ICINCO) - Volume 1*, juillet 2017. doi : 10.5220/0006418202210229. <http://www.scitepress.org/digitalLibrary/PublicationsDetail.aspx?ID=g0m51z1hJBU=&t=1>

Annexe B

Compléments de définitions

Unmanned System

"A powered physical system, with no human operator aboard the principal components, which acts in the physical world to accomplish assigned tasks. It may be mobile or stationary. It can include any and all associated supporting components such as Operator Control Units. Examples include unmanned ground vehicles (UGV), unmanned aerial vehicles/systems (UAV/UAS), unmanned maritime vehicles (UMV)–unmanned underwater vehicles (UUV) or unmanned water surface borne vehicles (USV)–unattended munitions (UM), and unattended ground sensors (UGS)." [NIST, 2008]

Niveaux d'autonomie

Les définitions non traduites du [NIST \[2008\]](#) sont listées ci-dessous :

Remote Control A mode of UMS operation wherein the human operator controls the UMS on a continuous basis, from a location off the UMS via only her/his direct observation. In this mode, the UMS takes no initiative and relies on continuous or nearly continuous input from the human operator.

Teleoperation A mode of UMS operation wherein the human operator, using sensory feedback, either directly controls the actuators or assigns incremental goals on a continuous basis, from a location off the UMS

Semi-Autonomous A mode of UMS operation wherein the human operator and/or the UMS plan(s) and conduct(s) a mission and requires various levels of HRI. The UMS is capable of autonomous operation in between the human interactions.

Fully Autonomous A mode of UMS operation wherein the UMS accomplishes its assigned mission, within a defined scope, without human intervention while adapting to operational and environmental conditions.

Annexe C

Compléments aux résultats du modèle basé sur les forces virtuelles

C.1 Mission globale

Nb drones	Référence du tunnel				
	a	b	c	d	e
2	3,32	3,32	3,32	3,14	2,58
3	2,92	2,93	2,93	2,62	2,37
4	2,87	2,92	2,92	2,63	2,17
5	2,78	2,90	2,92	2,48	2,56
6	2,42	2,90	2,92	2,43	2,38

TABLE C.1 – Distances minimales entre deux drones pour chaque jeu de paramètres, normalisées par la distance de sécurité D_{sec} .

Nb drones	Référence du tunnel				
	a	b	c	d	e
1	3,32	3,99	6,99	2,48	2,02
2	3,21	3,88	6,88	2,48	1,78
3	2,95	2,95	3,52	2,08	1,43
4	2,77	2,81	3,98	2,09	1,84
5	2,03	2,82	2,64	2,08	1,85
6	1,85	2,61	3,98	1,95	1,48

TABLE C.2 – Distances minimales entre un drone et un obstacle fixe pour chaque jeu de paramètres, normalisées par la distance de sécurité D_{sec} .

C.2 Surveillance

Nombre de drones	Valeurs de k			
	0,70	0,75	0,85	0,95
1	1640	1918	3312	11075
2	983	1064	1937	4849
3	734	741	1324	4480
4	602	571	999	3308
5	462	563	974	1299
6	453	383	644	2037

TABLE C.3 – Nombre d'itérations nécessaires au parcours de la première zone d'intérêt en fonction du nombre de drones dans l'essaim et de la valeur de k .

C.3 Traversée des tunnels

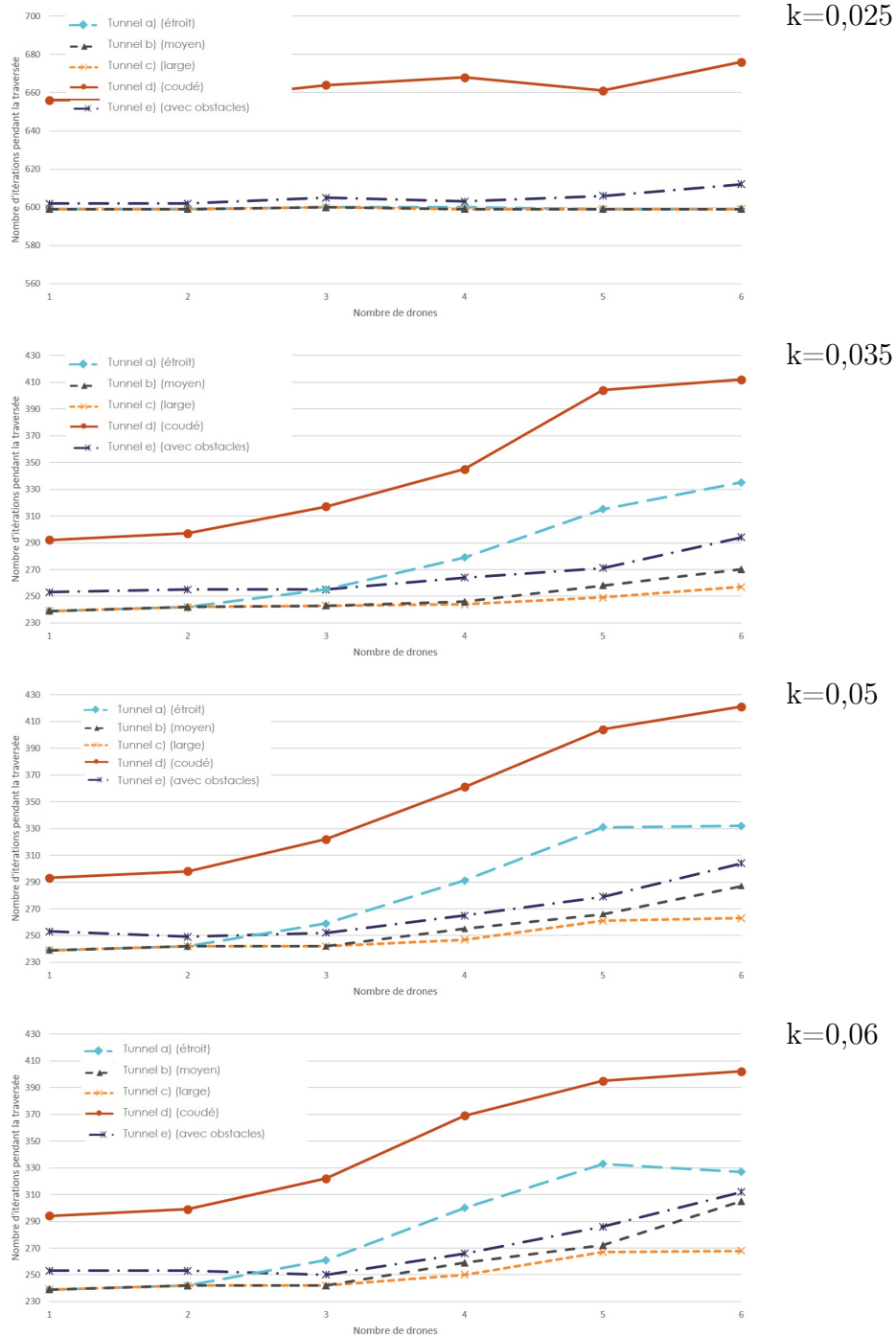


FIGURE C.1 – Nombre moyen d'itérations nécessaires à la traversée de chaque tunnel pour tous les tunnels, distingués par les valeurs de k testées.

C.3. Traversée des tunnels

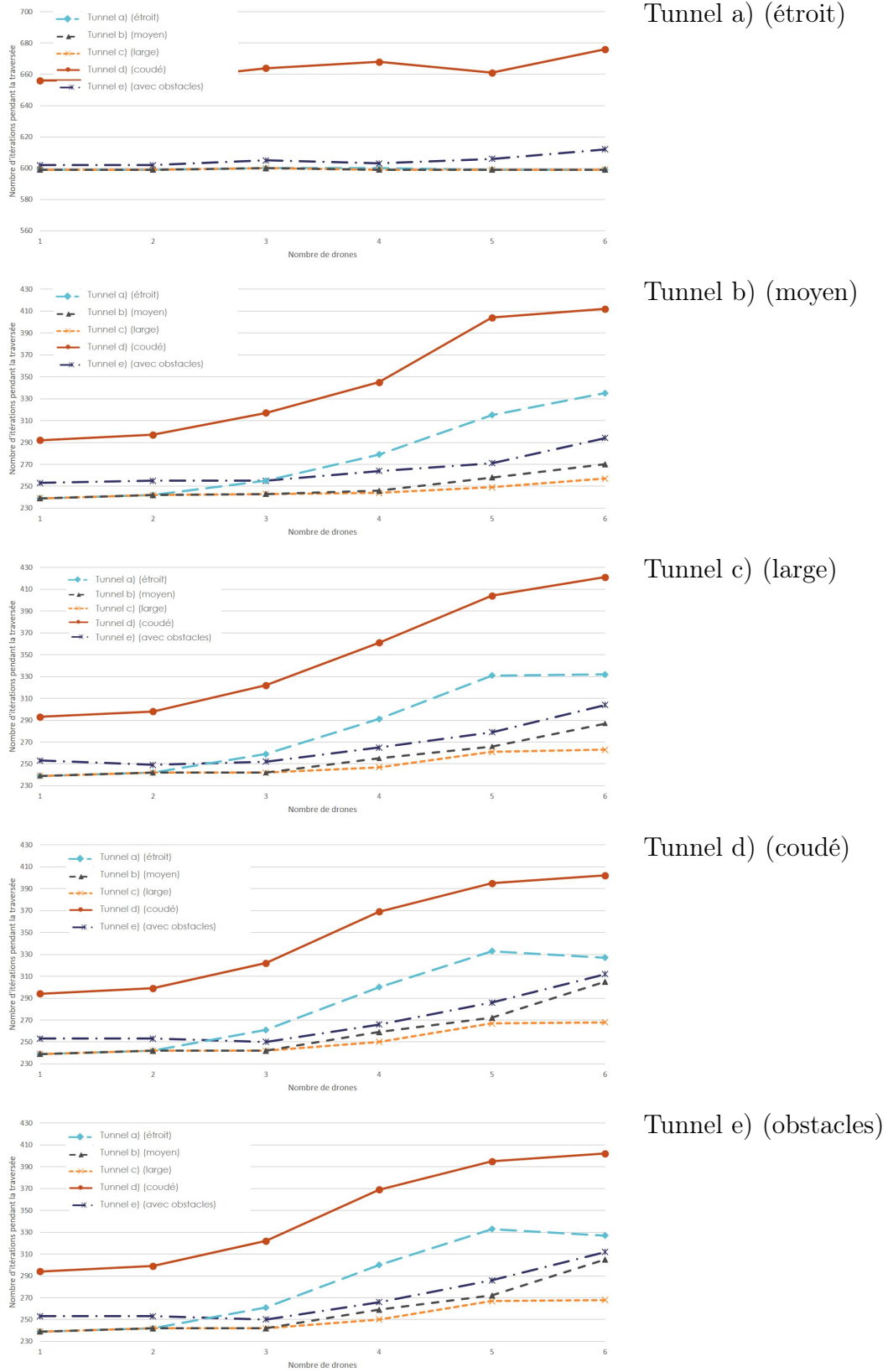


FIGURE C.2 – Nombre moyen d'itérations nécessaires à la traversée de chaque tunnel pour toutes les valeurs de k testées, tunnel par tunnel.

Annexe D

Compléments aux paramètres de simulation du modèle basé sur les champs de potentiels discrets

	Échantillon 1	Échantillon 2	Échantillon 3
Départ	[250,50,5]	[285,105,5]	[395,275,5]
Point cible 1	[364,332,8]	[115,345,21]	[235,25,11]
Point cible 2	[94,495,21]	[405,35,16]	[315,375,21]
Point cible 3	[132,260,17]	[375,35,6]	[475,75,6]
Point cible 4	[331,362,14]	[35,315,16]	[475,55,16]
Point cible 5	[459,31,14]	[455,105,11]	[165,365,21]
Point cible 6	[34,276,21]	[435,255,16]	[355,315,16]
Point cible 7	[168,380,8]	[95,165,21]	[115,495,11]
Point cible 8	[404,468,21]	[315,355,11]	[475,325,16]
Point cible 9	[472,260,14]	[95,225,6]	[315,235,6]
Point cible 10	[84,143,21]	[385,445,11]	[335,355,6]

TABLE D.1 – Coordonnées des points de départ et points cible tirés aléatoirement, pour chaque échantillon.

Bibliographie

- AERONEWS, 2018. Plus de 1 300 drones volent en formation.
URL <https://www.aeronewstv.com/fr/industrie/drones/4183-plus-de-1-300-drones-volent-en-formation.html>
- AL REDWAN NEWAZ, Abdullah, PRATAMA, Ferdian Adi et NAK YOUNG CHONG, 2013. Exploration Priority Based Heuristic Approach to UAV path planning. pages 521–526. IEEE. ISBN 978-1-4799-0509-6. doi : 10.1109/ROMAN.2013.6628557.
URL <http://ieeexplore.ieee.org/document/6628557>
- ASADPOUR, M., BERGH, B. Van den, GIUSTINIANO, D., HUMMEL, K. A., POLLIN, S. et PLATTNER, B., 2014. Micro aerial vehicle networks : an experimental analysis of challenges and opportunities. *IEEE Communications Magazine*, 52(7) :141–149. doi :10.1109/MCOM.2014.6852096.
URL <http://ieeexplore.ieee.org/document/6852096>
- ASADPOUR, Mahdi, GIUSTINIANO, Domenico, HUMMEL, Karin Anna, HEIMLICH, Simon et EGLI, Simon, 2013. Now or later ? : delaying data transfer in time-critical aerial communication. Dans *Proceedings of the ninth ACM conference on Emerging networking experiments and technologies - CoNEXT '13*, pages 127–132. ACM Press, Santa Barbara, California, USA. ISBN 978-1-4503-2101-3. doi :10.1145/2535372.2535409.
URL <http://dl.acm.org/citation.cfm?doid=2535372.2535409>
- AUTEFAGE, Vincent, 2015. *Découverte de services et collaboration au sein d'une flotte hétérogène et hautement dynamique d'objets mobiles communicants autonomes*. Thèse de doctorat, Bordeaux.
URL <http://www.theses.fr/2015BORD0205>
- AUTEFAGE, Vincent et MAGONI, Damien, 2017. NEmu : A distributed testbed for the virtualization of dynamic, fixed and mobile networks. *Computer Communications*, 80 :33–44. doi :10.1016/j.comcom.2016.01.005.
URL <https://hal.archives-ouvertes.fr/hal-01436511>
- BARLOW, Sarah, CHOI, Youngjun, BRICENO, Simon et MAVRIS, Dimitri, 2019. A Multi-UAS Trajectory Optimization Methodology for Complex Enclosed Environments. Dans *Proceedings of the 2019 International Conference on Unmanned Aircraft Systems (ICUAS)*, pages 588–596. IEEE, Atlanta, GA, USA. ISBN 978-1-72810-332-7.
URL <https://ieeexplore.ieee.org/abstract/document/8797839>

- BEKMEZCI, I., SEN, I. et ERKALKAN, E., 2015. Flying ad hoc networks (FANET) test bed implementation. Dans *2015 7th International Conference on Recent Advances in Space Technologies (RAST)*, pages 665–668. doi : 10.1109/RAST.2015.7208426.
URL <https://ieeexplore.ieee.org/document/7208426>
- BEKMEZCI, İlker, SAHINGOZ, Ozgur Koray et TEMEL, Şamil, 2013. Flying Ad-Hoc Networks (FANETs) : A survey. *Ad Hoc Networks*, 11(3) :1254–1270. doi :10.1016/j.adhoc.2012.12.004.
URL <http://dx.doi.org/10.1016/j.adhoc.2012.12.004>
- BERGOUNHOX, Julien, 2016. Vidéo : Intel bat un record mondial avec 100 drones volant simultanément.
URL <https://www.usine-digitale.fr/article/video-intel-bat-un-record-mondial-avec-100-drones-volant-simultanement.N374387>
- BOISSONAT, Jean-Daniel, FAVERJON, Bernard et MERLET, Jean-Pierre, 1998. *Techniques de la robotique*, tome 2 de *Traité des Nouvelles Technologies - série Robotique*. Paris, France, hermes édition. ISBN 2-86601-151-1. Chapitre 7.
- BOONYARAK, S. et PREMPRANEERACH, P., 2014. Real-time obstacle avoidance for car robot using potential field and local incremental planning. Dans *2014 11th International Conference on Electrical Engineering/Electronics, Computer, Telecommunications and Information Technology (ECTI-CON)*, pages 1–5. doi :10.1109/ECTICon.2014.6839898.
URL <https://ieeexplore.ieee.org/abstract/document/6839898>
- BOSKOVIC, Jovan, KNOEBEL, Nathan, MOSHTAGH, Nima, AMIN, Jayesh et LARSON, Gregory, 2009. Collaborative Mission Planning & Autonomous Control Technology (CoMPACT) System Employing Swarms of UAVs. Dans *AIAA Guidance, Navigation, and Control Conference*. American Institute of Aeronautics and Astronautics, Chicago, Illinois. ISBN 978-1-60086-978-5. doi :10.2514/6.2009-5653.
URL <http://arc.aiaa.org/doi/10.2514/6.2009-5653>
- BOUVRY, Pascal, CHAUMETTE, Serge, DANOY, Grégoire, GUERRINI, Gilles, JURQUET, Gilles, KUWERTZ, Achim, MÜLLER, Wilmuth, ROSALIE, Martin et SANDER, Jennifer, 2016. Using Heterogeneous Multilevel Swarms of UAVs and High-Level Data Fusion to Support Situation Management in Surveillance Scenarios. Dans *International Conference on Multisensor Fusion and Integration for Intelligent Systems (MFI 2016)*.
URL <https://hal.archives-ouvertes.fr/hal-01360162/>

- BOZYIĞIT, A., ALANKUŞ, G. et NASIBOĞLU, E., 2017. Public transport route planning : Modified dijkstra's algorithm. Dans *2017 International Conference on Computer Science and Engineering (UBMK)*, pages 502–505. doi : 10.1109/UBMK.2017.8093444.
URL <https://ieeexplore.ieee.org/abstract/document/8093444>
- BROCK, O. et KHATIB, O., 1999. High-speed navigation using the global dynamic window approach. Dans *Proceedings 1999 IEEE International Conference on Robotics and Automation (Cat. No.99CH36288C)*, tome 1, pages 341–346. IEEE, Detroit, MI, USA. ISBN 978-0-7803-5180-6. doi : 10.1109/ROBOT.1999.770002.
URL <http://ieeexplore.ieee.org/document/770002/>
- BRUST, Matthias R. et STRIMBU, Bogdan M., 2015. A networked swarm model for uav deployment in the assessment of forest environments. Dans *Intelligent Sensors, Sensor Networks and Information Processing (ISSNIP), 2015 IEEE Tenth International Conference on*, pages 1–6. IEEE.
URL <http://ieeexplore.ieee.org/abstract/document/7106967/>
- CAKIR, Murat, 2015. 2d path planning of UAVs with genetic algorithm in a constrained environment. Dans *Modeling, Simulation, and Applied Optimization (ICMSAO), 2015 6th International Conference on*, pages 1–5. IEEE.
URL <http://ieeexplore.ieee.org/abstract/document/7152235/>
- CASTEIGTS, A., ALBERT, J., CHAUMETTE, S., NAYAK, A. et STOJMENOVIC, I., 2010. Biconnecting a Network of Mobile Robots Using Virtual Angular Forces. Dans *2010 IEEE 72nd Vehicular Technology Conference - Fall*, pages 1–5. doi :10.1109/VETECONF.2010.5594265.
URL <https://ieeexplore.ieee.org/abstract/document/5594265>
- CASTEIGTS, Arnaud, ALBERT, Jérémie, CHAUMETTE, Serge, NAYAK, Amiya et STOJMENOVIC, Ivan, 2012. Biconnecting a network of mobile robots using virtual angular forces. *Computer Communications*, 35(9) :1038–1046. doi :10.1016/j.comcom.2011.09.008.
URL <http://linkinghub.elsevier.com/retrieve/pii/S0140366411002787>
- CASTEIGTS, Arnaud et LAPLACE, Rémi, 2019. JBotSim, a Tool for Fast Prototyping of Distributed Algorithms in Dynamic Networks. *Proceedings of the 8th international ICST conference on simulation tools and techniques SIMUTools*. ArXiv : 1001.1435.
URL <http://arxiv.org/abs/1001.1435>
- CE, 2019. RÈGLEMENT D'EXÉCUTION (UE) 2019/ 947 DE LA COMMISSION - du 24 mai 2019 - concernant les règles et procédures applicables

à l'exploitation d'aéronefs sans équipage à bord.

URL <https://eur-lex.europa.eu/legal-content/FR/TXT/PDF/?uri=CELEX:32019R0947&from=EN>

CHAO, Haiyang, CAO, Yongcan et CHEN, YangQuan, 2007. Autopilots for Small Fixed-Wing Unmanned Air Vehicles : A Survey. Dans *2007 International Conference on Mechatronics and Automation*, pages 3144–3149. IEEE, Harbin, China. ISBN 978-1-4244-0827-6 978-1-4244-0828-3. doi : 10.1109/ICMA.2007.4304064.

URL <http://ieeexplore.ieee.org/document/4304064/>

CHAUMETTE, S., LAPLACE, R., MAZEL, C., MIRAULT, R., DUNAND, A., LECOUTRE, Y. et PERBET, J-N., 2011. CARUS, an operational retasking application for a swarm of autonomous UAVs : First return on experience. pages 2003–2010. IEEE. ISBN 978-1-4673-0081-0 978-1-4673-0079-7 978-1-4673-0080-3. doi :10.1109/MILCOM.2011.6127613.

URL <http://ieeexplore.ieee.org/document/6127613/>

CHOI, B., KIM, B., KIM, E. et YANG, K. W., 2012. A modified dynamic window approach in crowded indoor environment for intelligent transport robot. Dans *2012 12th International Conference on Control, Automation and Systems*, pages 1007–1009.

COPTRZ, 2018. Top Collision Avoidance Drones.

URL <https://www.coptrz.com/top-collision-avoidance-drones/>

CORRIGAN, Fintan, 2018. Top Collision Avoidance Drones And Obstacle Detection Explained <https://www.dronezon.com/learn-about-drones-quadcopters/top-drones-with-obstacle-detection-collision-avoidance-sensors-explained/>.

URL <https://www.dronezon.com/learn-about-drones-quadcopters/top-drones-with-obstacle-detection-collision-avoidance-sensors-explained/>

DANIEL, K., NASH, A., KOENIG, S. et FELNER, A., 2010. Theta* : Any-Angle Path Planning on Grids. *Journal of Artificial Intelligence Research*, 39 :533–579. doi :10.1613/jair.2994.

URL <https://jair.org/index.php/jair/article/view/10676>

DANOY, Grégoire, R. BRUST, Matthias et BOUVRY, Pascal, 2015. Connectivity Stability in Autonomous Multi-level UAV Swarms for Wide Area Monitoring. ACM, Cancun, Mexico. ISBN 978-1-4503-3760-1. doi : <http://dx.doi.org/10.1145/2815347.2815351>.

URL https://www.researchgate.net/publication/283265492_Connectivity_Stability_in_Autonomous_Multi-level_UAV_Swarms_for_Wide_Area_Monitoring

-
- DI MARIO, E., TALEBPOUR, Z. et MARTINOLI, A., 2013. A comparison of PSO and Reinforcement Learning for multi-robot obstacle avoidance. Dans *2013 IEEE Congress on Evolutionary Computation*, pages 149–156. doi : 10.1109/CEC.2013.6557565.
URL <https://ieeexplore.ieee.org/abstract/document/6557565>
- DIJKSTRA, Edsger W., 1959. A Note on Two Problems in Connexion with Graphs. *Numerische Mathematik*, 1 :269–271.
URL <http://www.springerlink.com/index/uu8608u0u27k7256.pdf>
- DONG, Z., CHEN, Z., ZHOU, R. et ZHANG, R., 2011. A hybrid approach of virtual force and a search algorithm for UAV path re-planning. Dans *2011 6th IEEE Conference on Industrial Electronics and Applications*, pages 1140–1145. doi :10.1109/ICIEA.2011.5975758.
URL <https://ieeexplore.ieee.org/abstract/document/5975758>
- DU, Biqiang et LIU, Shizhao, 2016. A common obstacle avoidance module based on fuzzy algorithm for Unmanned Aerial Vehicle - IEEE Conference Publication. pages 245–248. IEEE. ISBN 978-1-5090-2733-0. doi :10.1109/CYBER.2016.7574830.
URL <http://ieeexplore.ieee.org/document/7574830/>
- DURAND, Nicolas, 2004. *Algorithmes génétiques et autres outils d'optimisation appliqués à la gestion du trafic aérien*. Habilitation à Diriger des Recherches, Institut National Polytechnique de Toulouse.
URL <https://tel.archives-ouvertes.fr/tel-00010950/>
- FERREIRA, Afonso, 2002. On models and algorithms for dynamic communication networks : the case for evolving graphs. page 7.
- FERREIRA, Diogo, LIMA, Luisa et BARROS, Joao, 2009. NECO : Network Coding Simulator. ICST. ISBN 978-963-9799-45-5. doi :10.4108/ICST.SIMUTOOLS2009.5646.
URL <http://eudl.eu/doi/10.4108/ICST.SIMUTOOLS2009.5646>
- FLOREANO, Dario et MONDADA, Francesco, 1996. Evolution of homing navigation in a real mobile robot. *IEEE Transactions on Systems, Man and Cybernetics, Part B (Cybernetics)*, 26(3) :396–407. doi :10.1109/3477.499791.
URL <http://ieeexplore.ieee.org/document/499791/>
- FOX, D., BURGARD, W. et THRUN, S., 1997. The dynamic window approach to collision avoidance. *IEEE Robotics & Automation Magazine*, 4(1) :23–33. doi :10.1109/100.580977.
URL <http://ieeexplore.ieee.org/document/580977>

- FUSIC, S. Julius, RAMKUMAR, P. et HARIHARAN, K., 2018. Path planning of robot using modified dijkstra Algorithm. Dans *2018 National Power Engineering Conference (NPEC)*, pages 1–5. doi :10.1109/NPEC.2018.8476787. URL <https://ieeexplore.ieee.org/abstract/document/8476787/>
- GARETTO, M., GRIBAUDO, M., CHIASSERINI, C. et LEONARDI, E., 2007. A Distributed Sensor Relocatlon Scheme for Environmental Control. Dans *2007 IEEE International Conference on Mobile Adhoc and Sensor Systems*, pages 1–10. doi :10.1109/MOBHOC.2007.4428663. URL <https://ieeexplore.ieee.org/abstract/document/4428663/>
- GARIP, Z. B., ATALI, G., KARAYEL, D. et OZKAN, S. S., 2018. Path Planning for Multiple Mobile Robots in Static Environment using Hybrid Algorithm. Dans *2018 2nd International Symposium on Multidisciplinary Studies and Innovative Technologies (ISMSIT)*, pages 1–4. doi : 10.1109/ISMSIT.2018.8567321. URL <https://ieeexplore.ieee.org/abstract/document/8567321/>
- GE, S S et CUI, Y J, 2002. Dynamic Motion Planning for Mobile Robots Using Potential Field Method. *Autonomous Robot*, 13 :207–222. URL <http://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.581.3474&rep=rep1&type=pdf>
- GEOPORTAIL, 2019. Géoportail. URL [https://www.geoportail.gouv.fr/carte?c=-0.6601155939472929,45.12124680059938&z=17&l0=ORTHOIMAGERY.ORTHOPHOTOS::GEOPORTAIL:OGC:WMTS\(1\)&l1=CADASTRALPARCELS.PARCELS::GEOPORTAIL:OGC:WMTS\(1\)&permalink=yes](https://www.geoportail.gouv.fr/carte?c=-0.6601155939472929,45.12124680059938&z=17&l0=ORTHOIMAGERY.ORTHOPHOTOS::GEOPORTAIL:OGC:WMTS(1)&l1=CADASTRALPARCELS.PARCELS::GEOPORTAIL:OGC:WMTS(1)&permalink=yes)
- GERTLER, Jeremiah, 2012. U.S. Unmanned Aerial Systems. Rapport technique. URL <https://fas.org/sgp/crs/natsec/R42136.pdf>
- GHAZALI, S. N. A. M., ANUAR, H. A., ZAKARIA, S. N. A. S. et YUSOFF, Z., 2016. Determining position of target subjects in Maritime Search and Rescue (MSAR) operations using rotary wing Unmanned Aerial Vehicles (UAVs). Dans *2016 International Conference on Information and Communication Technology (ICICTM)*, pages 1–4. doi :10.1109/ICICTM.2016.7890765. URL <https://ieeexplore.ieee.org/abstract/document/7890765/>
- GOBEL, J. et KRZESINSKI, A. E., 2008. A Model of Autonomous Motion in Ad Hoc Networks to Maximise Area Coverage. Dans *2008 Australasian Telecommunication Networks and Applications Conference*, pages 258–263. doi :10.1109/ATNAC.2008.4783333. URL <https://ieeexplore.ieee.org/abstract/document/4783333/>

- GUILLEN-PEREZ, Antonio et CANO, Maria-Dolores, 2018. Flying Ad Hoc Networks : A New Domain for Network Communications. *Sensors (Basel, Switzerland)*, 18(10). doi :10.3390/s18103571.
URL <https://www.ncbi.nlm.nih.gov/pmc/articles/PMC6209929/>
- GUILLEN-PEREZ, Antonio, SANCHEZ-IBORRA, Ramon, CANO, Maria-Dolores, SANCHEZ-AARNOUTSE, Juan Carlos et GARCIA-HARO, Joan, 2016. WiFi networks on drones. Dans *2016 ITU Kaleidoscope : ICTs for a Sustainable World (ITU WT)*, pages 1–8. IEEE, Bangkok, Thailand. ISBN 978-92-61-20451-8. doi :10.1109/ITU-WT.2016.7805730.
URL <http://ieeexplore.ieee.org/document/7805730/>
- HART, P. E., NILSSON, N. J. et RAPHAEL, B., 1968. A Formal Basis for the Heuristic Determination of Minimum Cost Paths. *IEEE Transactions on Systems Science and Cybernetics*, 4(2) :100–107. doi :10.1109/TSSC.1968.300136.
URL <https://ieeexplore.ieee.org/abstract/document/4082128/>
- HONG, Z., CHUN-LONG, S., ZI-JUN, Z., WEI, A., DE-QIANG, Z. et JING-JING, W., 2015. A Modified Dynamic Window Approach to Obstacle Avoidance Combined with Fuzzy Logic. Dans *2015 14th International Symposium on Distributed Computing and Applications for Business Engineering and Science (DCABES)*, pages 523–526. doi :10.1109/DCABES.2015.136.
URL <https://ieeexplore.ieee.org/abstract/document/7429670/>
- HOU, J., ZHANG, Q., ZHANG, Y., ZHU, K., LV, Y. et YU, C., 2016. Low altitude sense and avoid for MUAV based on stereo vision. Dans *2016 35th Chinese Control Conference (CCC)*, pages 5579–5584. doi :10.1109/ChiCC.2016.7554226.
URL <https://ieeexplore.ieee.org/abstract/document/7554226/>
- HOWARD, Andrew, MATARIĆ, Maja J. et SUKHATME, Gaurav S., 2002. Mobile Sensor Network Deployment using Potential Fields : A Distributed, Scalable Solution to the Area Coverage Problem. Dans Hajime Asama, Tamio Arai, Toshio Fukuda et Tsutomu Hasegawa, rédacteurs, *Distributed Autonomous Robotic Systems 5*, pages 299–308. Springer Japan. ISBN 978-4-431-65943-3 978-4-431-65941-9. doi :10.1007/978-4-431-65941-9_30.
URL http://link.springer.com/chapter/10.1007/978-4-431-65941-9_30
- HUANG, H. et TSAI, C., 2011. Global path planning for autonomous robot navigation using hybrid metaheuristic GA-PSO algorithm. Dans *SICE Annual Conference 2011*, pages 1338–1343.
URL <https://ieeexplore.ieee.org/abstract/document/6060543/>

- IBRAHIM, M. I., SARIFF, N., JOHARI, J. et BUNIYAMIN, N., 2014. Mobile robot obstacle avoidance in various type of static environments using fuzzy logic approach. Dans *2014 2nd International Conference on Electrical, Electronics and System Engineering (ICEESE)*, pages 83–88. doi : 10.1109/ICEESE.2014.7154600.
URL <https://ieeexplore.ieee.org/abstract/document/7154600/>
- IOANNIS K., Nikolos, KIMON P., Valavanis et NIKOS C., Tsourveloudis, 2003. Evolutionary algorithm based offline/online path planner for UAV navigation. *IEEE Transactions on Systems, Man, and Cybernetics, Part B (Cybernetics)*, 33 :898–912. doi :10.1109/TSMCB.2002.804370.
URL <http://ieeexplore.ieee.org/document/1245266/>
- JARDOSH, Amit, BELDING-ROYER, Elizabeth M., ALMERTH, Kevin C. et SURI, Subhash, 2003. Towards realistic mobility models for mobile ad hoc networks. Dans *Proceedings of the 9th annual international conference on Mobile computing and networking - MobiCom '03*, page 217. ACM Press, San Diego, CA, USA. ISBN 978-1-58113-753-8. doi :10.1145/938985.939008.
URL <http://portal.acm.org/citation.cfm?doid=938985.939008>
- KENNEDY, J. et EBERHART, R., 1995. Particle swarm optimization. Dans *IEEE International Conference on Neural Networks, 1995. Proceedings*, tome 4, pages 1942–1948 vol.4. doi :10.1109/ICNN.1995.488968.
URL <https://ieeexplore.ieee.org/abstract/document/488968>
- KHAN, Atta R., BILAL, Sardar M. et OTHMAN, Mazliza, 2012. A performance comparison of open source network simulators for wireless networks. Dans *Control System, Computing and Engineering (ICCSCE), 2012 IEEE International Conference on*, pages 34–38. IEEE.
URL <http://ieeexplore.ieee.org/abstract/document/6487111/>
- KHATIB, Oussama, 1986. Real-time obstacle avoidance for manipulators and mobile robots. *The international journal of robotics research*, 5(1) :90–98.
URL <http://journals.sagepub.com/doi/abs/10.1177/027836498600500106>
- KOREN, Y. et BORENSTEIN, J., 1991. Potential field methods and their inherent limitations for mobile robot navigation. Dans *Proceedings. 1991 IEEE International Conference on Robotics and Automation*, pages 1398–1404. IEEE Comput. Soc. Press, Sacramento, CA, USA. ISBN 978-0-8186-2163-5. doi :10.1109/ROBOT.1991.131810.
URL <http://ieeexplore.ieee.org/document/131810/>
- LAM, Hong-Dang, 2015. Implementing Boids behaviors on the ChIRP robots. Master's Thesis, Norwegian University of Science and Technology, Trond-

- heim.
URL <https://ntnuopen.ntnu.no/ntnu-xmlui/handle/11250/2353550>
- LAPLACE, Rémi, 2012. *Applications et services DTN pour flotte collaborative de drones*. Thèse de doctorat, Université Sciences et Technologies-Bordeaux I.
URL <http://www.theses.fr/2012BOR14747>
- LAVALLE, Steven M., 1998. Rapidly-Exploring Random Trees : A New Tool for Path Planning. doi :10.1.1.35.1853.
URL <http://citeseerx.ist.psu.edu/viewdoc/summary?doi=10.1.1.35.1853>
- LEGIFRANCE, 2015. Arrêté du 17 décembre 2015 relatif à la conception des aéronefs civils qui circulent sans personne à bord, aux conditions de leur emploi et aux capacités requises des personnes qui les utilisent.
URL <https://www.legifrance.gouv.fr/eli/arrete/2015/12/17/DEVA1528542A/jo/texte>
- LI, Ming, YANG, Zhenyu et LOU, Wenjing, 2011. CodeOn : Cooperative Popular Content Distribution for Vehicular Networks using Symbol Level Network Coding. *IEEE Journal on Selected Areas in Communications*, 29(1) :223–235. doi :10.1109/JSAC.2011.110121.
URL <https://ieeexplore.ieee.org/abstract/document/5678796/>
- LI, X. et CHEN, J., 2017. An Efficient Framework for Target Search with Cooperative UAVs in a FANET. Dans *2017 IEEE International Symposium on Parallel and Distributed Processing with Applications and 2017 IEEE International Conference on Ubiquitous Computing and Communications (ISPA/IUCC)*, pages 306–313. doi :10.1109/ISPA/IUCC.2017.00051.
URL <https://ieeexplore.ieee.org/abstract/document/8367281/>
- LI, X. Q., QIU, L., AZIZ, S., PAN, J. F., YUAN, J. P. et ZHANG, B., 2017. Control method of UAV based on RRT for target tracking in cluttered environment. Dans *2017 7th International Conference on Power Electronics Systems and Applications - Smart Mobility, Power Transfer Security (PESA)*, pages 1–4. doi :10.1109/PESA.2017.8277732.
URL <https://ieeexplore.ieee.org/abstract/document/8277732/>
- LIU, Hai, CHU, Xiaowen, LEUNG, Yiu-Wing et DU, Rui, 2010. Simple movement control algorithm for bi-connectivity in robotic sensor networks. *IEEE Journal on Selected Areas in Communications*, 28(7) :994–1005. doi : 10.1109/JSAC.2010.100904.
URL <http://ieeexplore.ieee.org/document/5555924/>

- LIU, Mushuang, WAN, Yan et LEWIS, Frank L., 2017. Analysis of the Random Direction Mobility Model with a Sense-and-Avoid Protocol. Dans *2017 IEEE Globecom Workshops (GC Wkshps)*, pages 1–6. doi :10.1109/GLOCOMW.2017.8269071.
URL <https://ieeexplore.ieee.org/abstract/document/8269071/>
- LIU, Yuecheng et ZHAO, Yongjia, 2016. A virtual-waypoint based artificial potential field method for UAV path planning. Dans *Guidance, Navigation and Control Conference (CGNCC), 2016 IEEE Chinese*, pages 949–953. IEEE.
URL <http://ieeexplore.ieee.org/abstract/document/7828913/>
- LOUKIL, Ridha, 2018. Intel fait voler 2018 drones pour le spectacle de lumière de son 50ème anniversaire - Electronique.
URL <https://www.usinenouvelle.com/article/video-intel-fait-voler-2018-drones-pour-le-spectacle-de-lumiere-de-son-50eme-anniversaire.N721574>
- MAC, Thi Thoa, COPOT, Cosmin, HERNANDEZ, Andres et DE KEYSER, Robin, 2016. Improved potential field method for unknown obstacle avoidance using UAV in indoor environment. Dans *Applied Machine Intelligence and Informatics (SAMi), 2016 IEEE 14th International Symposium on*, pages 345–350. IEEE.
URL <http://ieeexplore.ieee.org/abstract/document/7423032/>
- MARIO, E. Di, NAVARRO, I. et MARTINOLI, A., 2014. The role of environmental and controller complexity in the distributed optimization of multi-robot obstacle avoidance. Dans *2014 IEEE International Conference on Robotics and Automation (ICRA)*, pages 571–577. doi :10.1109/ICRA.2014.6906912.
URL <https://ieeexplore.ieee.org/abstract/document/6906912/>
- MESSOUS, M., SENOUCI, S. et SEDJELMACI, H., 2016. Network connectivity and area coverage for UAV fleet mobility model with energy constraint. Dans *2016 IEEE Wireless Communications and Networking Conference*, pages 1–6. doi :10.1109/WCNC.2016.7565125.
URL <https://ieeexplore.ieee.org/abstract/document/7565125/>
- NAIMI, Sabrine, 2015. *Gestion de la mobilité dans les réseaux Ad Hoc par anticipation des métriques de routage*. Thèse de doctorat, Université Paris Sud - Paris XI ; École nationale d'ingénieurs de Tunis (Tunisie), Paris.
URL <https://tel.archives-ouvertes.fr/tel-01208152/document>
- NIST, 2008. Autonomy Levels for Unmanned Systems (ALFUS) Framework Volume I : Terminology Version 2.0. *NIST Special Publication*, I(2) :47.
URL https://www.nist.gov/sites/default/files/documents/el/isd/ks/NISTSP_1011-I-2-0.pdf

- NS 3, 2019. ns-3.
URL <https://www.nsnam.org/>
- OACI, 2018. Dossier : Les drones, un sujet montant pour l’aviation civile française et internationale.
URL <https://oaci.delegfrance.org/Dossier-Les-drones-un-sujet-montant-pour-l-aviation-civile-francaise-et>
- OMNET++, 2019. OMNeT++ Discrete Event Simulator - Home.
URL <https://omnetpp.org/>
- OPNET, 2019. OPNET Technologies – Network Simulator.
URL <https://www.riverbed.com/fr/products/steelcentral/opnet.html>
- OZALP, Nuri et SAHINGOZ, Ozgur Koray, 2013. Optimal UAV path planning in a 3d threat environment by using parallel evolutionary algorithms. Dans *2013 International Conference on Unmanned Aircraft Systems (ICUAS)*, pages 308–317. IEEE, Atlanta, GA, USA. ISBN 978-1-4799-0817-2 978-1-4799-0815-8 978-1-4799-0816-5. doi :10.1109/ICUAS.2013.6564703.
URL <http://ieeexplore.ieee.org/document/6564703/>
- PARK, J. et KIM, Y., 2012. Stereo vision based collision avoidance of quadrotor UAV. Dans *2012 12th International Conference on Control, Automation and Systems*, pages 173–178.
URL <https://ieeexplore.ieee.org/abstract/document/6393426/>
- PARKER, Lynne E., 2008. Distributed intelligence : overview of the field and its application in multi-robot systems. *Journal of Physical Agents (JoPha)*, 2(1) :5–14. doi :10.14198/JoPha.2008.2.1.02.
URL <http://hdl.handle.net/10045/12591>
- POTY, Alexandre, 2006. *Planification de trajectoire dans un environnement dynamique et génération de mouvement d’ordre non entier*. Thèse de doctorat, Bordeaux 1, Bordeaux.
URL <http://www.theses.fr/2006BOR13202>
- PRINDLE, Drew, 2018. Pocket-sized and practically perfect, the Mavic Air is DJI’s best drone yet <https://www.digitaltrends.com/drone-reviews/dji-mavic-air-review/>.
URL <https://www.digitaltrends.com/drone-reviews/dji-mavic-air-review/>
- QIAN JIA et XINGSONG WANG, 2009. Path planning for mobile robots based on a modified potential model. Dans *2009 International Conference on Mechatronics and Automation*, pages 4946–4951. IEEE, Changchun, China.

ISBN 978-1-4244-2692-8. doi :10.1109/ICMA.2009.5246021.

URL <http://ieeexplore.ieee.org/document/5246021/>

RAČINSKÝ, Matěj, 2016. RRT-path method used for cooperative surveillance by group of helicopters. Rapport technique, Czech Technical University in Prague.

URL <https://dspace.cvut.cz/bitstream/handle/10467/64710/F3-BP-2016-Racinsky-Matej-bakalarka-se-zadanim.pdf>

RECEVEUR, Jean-Baptiste, 2019. *Optimisation de trajectoire en environnement 2D : application au véhicule autonome*. Thèse de doctorat, Bordeaux.

RECEVEUR, Jean-Baptiste, MELCHIOR, Pierre et VICTOR, Stéphane, 2018. Optimisation multi-critère pour véhicules autonomes en environnement dynamique. *Automatique*, 2(1). doi :10.21494/ISTE.OP.2018.0283.

URL <https://www.openscience.fr/Optimisation-multi-critere-pour-vehicules-autonomes-en-environnement-dynamique>

RECEVEUR, Jean-Baptiste, VICTOR, Stéphane et MELCHIOR, Pierre, 2017. Multi-criteria trajectory optimization for autonomous vehicles. *IFAC-PapersOnLine*, 50(1) :12520–12525. doi :10.1016/j.ifacol.2017.08.2063.

URL <https://linkinghub.elsevier.com/retrieve/pii/S2405896317327064>

REYNOLDS, Craig W., 1987. Flocks, Herds and Schools : A Distributed Behavioral Model. Dans *Proceedings of the 14th Annual Conference on Computer Graphics and Interactive Techniques*, SIGGRAPH '87, pages 25–34. ACM, New York, NY, USA. ISBN 978-0-89791-227-3. doi :10.1145/37401.37406.

URL <http://doi.acm.org/10.1145/37401.37406>

ROSALIE, Martin, 2017. ASIMUT Project Website.

URL <https://asimut.gforge.uni.lu/index.html>

ROSALIE, Martin, DANOY, Grégoire, CHAUMETTE, Serge et BOUVRY, Pascal, 2016. From random process to chaotic behavior in swarms of UAVs. Dans *6th ACM Symposium on Development and Analysis of Intelligent Vehicular Networks and Applications*, DIVANet '16 Proceedings of the 6th ACM Symposium on Development and Analysis of Intelligent Vehicular Networks and Applications, pages 9–15. Malta. doi :10.1145/2989275.2989281.

URL <https://hal.archives-ouvertes.fr/hal-01360176>

SARANRITTICHAJ, P., NIPARNAN, N. et SUDSANG, A., 2013. Robust local obstacle avoidance for mobile robot based on Dynamic Window approach. Dans *2013 10th International Conference on Electrical Engineering/Electronics, Computer, Telecommunications and Information Techno-*

logy, pages 1–4. doi :10.1109/ECTICon.2013.6559615.

URL <https://ieeexplore.ieee.org/abstract/document/6559615/>

SASKA, Martin, CHUDOBA, Jan, PRECIL, Libor, THOMAS, Justin, LOIANNI, Giuseppe, TRESNAK, Adam, VONASEK, Vojtech et KUMAR, Vijay, 2014a. Autonomous deployment of swarms of micro-aerial vehicles in cooperative surveillance. Dans *2014 International Conference on Unmanned Aircraft Systems (ICUAS)*, pages 584–595. IEEE, Orlando, FL, USA. ISBN 978-1-4799-2376-2. doi :10.1109/ICUAS.2014.6842301.

URL <http://ieeexplore.ieee.org/document/6842301/>

SASKA, Martin, VAKULA, Jan et PŘEUCIL, Libor, 2014b. Swarms of micro aerial vehicles stabilized under a visual relative localization. Dans *Robotics and Automation (ICRA), 2014 IEEE International Conference on*, pages 3570–3575. IEEE.

URL <https://ieeexplore.ieee.org/abstract/document/6907374/>

SPARKFUN, 2019. XBee Buying Guide - SparkFun Electronics.

URL https://www.sparkfun.com/pages/xbee_guide

SRINIVAS, M. et PATNAIK, Lalit M., 1994. Genetic algorithms : a survey. *Computer*, 27(6) :17–26. doi :10.1109/2.294849.

URL <http://ieeexplore.ieee.org/document/294849/>

STENTZ, A., 1994. Optimal and efficient path planning for partially-known environments. Dans *Proceedings of the 1994 IEEE International Conference on Robotics and Automation*, pages 3310–3317 vol.4. doi :10.1109/ROBOT.1994.351061.

URL https://link.springer.com/chapter/10.1007/978-1-4615-6325-9_11

SUN, Jiayi, TANG, Jun et LAO, Songyang, 2017. Collision Avoidance for Cooperative UAVs With Optimized Artificial Potential Field Algorithm. *IEEE Access*, 5 :18382–18390. doi :10.1109/ACCESS.2017.2746752.

URL <http://ieeexplore.ieee.org/document/8022685/>

TO, Long, BATI, Andre et HILLIARD, Don, 2009. Radar Cross Section measurements of small Unmanned Air Vehicle Systems in non-cooperative field environments. Dans *Antennas and Propagation, 2009. EuCAP 2009. 3rd European Conference on*, pages 3637–3641. IEEE.

URL <http://ieeexplore.ieee.org/abstract/document/5068379/>

VIENNOT, Laurent et JACQUET, Philippe, 2007. Bi-connexité, k-connexité et multipoints relais. page 5.

URL <https://hal.inria.fr/inria-00176939/document>

- VONÁSEK, Vojtěch, FAIGL, Jan, KRAJNÍK, Tomáš et PŘEUČIL, Libor, 2009. RRT-path—a guided rapidly exploring random tree. Dans *Robot Motion and Control 2009*, pages 307–316. Springer.
URL http://link.springer.com/chapter/10.1007/978-1-84882-985-5_28
- WANG, C., WANG, L., QIN, J., WU, Z., DUAN, L., LI, Z., CAO, M., OU, X., SU, X., LI, W., LU, Z., LI, M., WANG, Y., LONG, J., HUANG, M., LI, Y. et WANG, Q., 2015. Path planning of automated guided vehicles based on improved A-Star algorithm. Dans *2015 IEEE International Conference on Information and Automation*, pages 2071–2076. doi : 10.1109/ICInfA.2015.7279630.
URL <https://ieeexplore.ieee.org/abstract/document/7279630/>
- WANG, S. et ZHAO, X., 2011. The improved Dijkstra’s shortest path algorithm. Dans *2011 Seventh International Conference on Natural Computation*, tome 4, pages 2313–2316. doi :10.1109/ICNC.2011.6022589.
URL <https://ieeexplore.ieee.org/abstract/document/6022589/>
- XBEE, 2011. XBee® & XBee-PRO® ZB ZigBee® Embedded RF Module Family for OEMs.
URL https://cdn.sparkfun.com/datasheets/Wireless/Zigbee/ds_xbeezbmodules.pdf
- XIAODONG ZHUANG, QINGCHUN MENG, BO YIN et HANPING WANG, 2002. Robot path planning by artificial potential field optimization based on reinforcement learning with fuzzy state. Dans *Proceedings of the 4th World Congress on Intelligent Control and Automation (Cat. No.02EX527)*, tome 2, pages 1166–1170 vol.2. doi :10.1109/WCICA.2002.1020763.
URL <https://ieeexplore.ieee.org/abstract/document/1020763/>
- XING, Guoliang, WANG, Xiaorui, ZHANG, Yuanfang, LU, Chenyang, PLESS, Robert et GILL, Christopher, 2005. Integrated coverage and connectivity configuration for energy conservation in sensor networks. *ACM Transactions on Sensor Networks*, 1(1) :36–72. doi :10.1145/1077391.1077394.
URL <http://portal.acm.org/citation.cfm?doid=1077391.1077394>
- YANMAZ, Evsen et BETTSTETTER, Christian, 2010. Area coverage with unmanned vehicles : A belief-based approach. Dans *Vehicular Technology Conference (VTC 2010-Spring), 2010 IEEE 71st*, pages 1–5. IEEE.
URL <http://ieeexplore.ieee.org/abstract/document/5493644/>
- YANMAZ, Evsen, COSTANZO, Carmelo, BETTSTETTER, Christian et ELMENREICH, Wilfried, 2010. A discrete stochastic process for coverage analysis of autonomous UAV networks. Dans *2010 IEEE Globecom Workshops*,

- pages 1777–1782. IEEE, Miami, FL, USA. ISBN 978-1-4244-8863-6. doi : 10.1109/GLOCOMW.2010.5700247.
URL <http://ieeexplore.ieee.org/document/5700247/>
- YU, H., YANG, W., ZHANG, H. et HE, W., 2017. A UAV-based crack inspection system for concrete bridge monitoring. Dans *2017 IEEE International Geoscience and Remote Sensing Symposium (IGARSS)*, pages 3305–3308. doi :10.1109/IGARSS.2017.8127704.
URL <https://ieeexplore.ieee.org/abstract/document/8127704/>
- YUAN, Quan, ZHAN, Jingyuan et LI, Xiang, 2017. Outdoor flocking of quadcopter drones with decentralized model predictive control. *ISA Transactions*, 71 :84–92. doi :10.1016/j.isatra.2017.07.005.
URL <http://www.sciencedirect.com/science/article/pii/S0019057817304895>
- ZHIIH, C. C., RAGAVAN, S. K. V. et SHANMUGAVEL, M., 2015. Development of a simple, low-cost autopilot system for multi-rotor UAVs. Dans *2015 IEEE Recent Advances in Intelligent Computational Systems (RAICS)*, pages 285–289. doi :10.1109/RAICS.2015.7488429.
URL <https://ieeexplore.ieee.org/abstract/document/7488429/>
- ZHOU, Xin, WANG, Weiping, WANG, Tao, LI, Xiaobo et LI, Zhifei, 2017. A research framework on mission planning of the UAV swarm. Dans *2017 12th System of Systems Engineering Conference (SoSE)*, pages 1–6. IEEE, Waikoloa, HI, USA. ISBN 978-1-5090-5945-4. doi :10.1109/SYSOSE.2017.7994984.
URL <http://ieeexplore.ieee.org/document/7994984/>
- ZHUONING, Dong, RULIN, Zhang, ZONGJI, Chen et RUI, Zhou, 2010. Study on UAV Path Planning Approach Based on Fuzzy Virtual Force. *Chinese Journal of Aeronautics*, 23(3) :341–350. doi :10.1016/S1000-9361(09)60225-9.
URL <https://linkinghub.elsevier.com/retrieve/pii/S1000936109602259>
- ŞAHIN, Erol, 2005. Swarm Robotics : From Sources of Inspiration to Domains of Application. Dans William M. Spears, rédacteur, *Swarm Robotics*, Lecture Notes in Computer Science, pages 10–20. Springer Berlin Heidelberg. ISBN 978-3-540-30552-1.
URL https://link.springer.com/chapter/10.1007/978-3-540-30552-1_2

Résumé : Les drones sont aujourd’hui devenus des outils incontournables dans de nombreux domaines, aussi bien civils que militaires. Afin de diminuer la charge de travail de leurs opérateurs, certaines étapes des missions qui leur sont confiées peuvent être automatisées, jusqu’à atteindre l’autonomie complète de la plateforme qui opérera sans intervention humaine. La mobilité en environnement complexe est une problématique majeure à traiter lorsqu’il est question d’autonomie. En effet, un drone qui découvre son environnement au fur et à mesure de ses déplacements doit régulièrement calculer un chemin vers son point cible, chemin qui doit éviter les obstacles présents sur le trajet. Dans ce contexte, la contribution principale de cette thèse est une méthode de calcul de chemin pour un drone autonome évoluant en environnement complexe. De plus, si la mission est trop complexe pour être réalisée par un seul drone (par exemple car elle nécessite l’utilisation de nombreux capteurs ou se déroule sur une zone étendue), alors on utilise un essaim composé de plusieurs appareils autonomes, qui pourront, collaborativement, remplir la mission donnée. En revanche, même si l’utilisation conjointe de plusieurs plateformes collaboratives permet d’augmenter les performances du système global, de nouvelles problématiques apparaissent, telles que la gestion des communications et l’évitement des drones voisins. La méthode de calcul de chemin que nous proposons prend en compte ces problématiques et est donc adaptée à l’utilisation par les drones d’un essaim.

Mots-clés : Drone autonome, essaim, modèle de mobilité, calcul de chemin, système distribué, compacité

Abstract : Drones are major tools today to perform numerous tasks in a wide range of operations, both for military and civil activities. In order to reduce the workload of operators, some parts of the drones’ missions can be automated, until achieving a full autonomy of the platforms. In this case, the drones can fulfil their mission without any human intervention. Path planning is a major task to achieve autonomy. Indeed, the drones have to detect obstacles while they discover their environment and dynamically calculate a path. This path avoids the obstacles and allows the drones to fulfil their mission. In this context, one of our contributions is a path planning method for an autonomous drone evolving in a complex environment. Furthermore, the variety of tasks one would like to achieve increases quicker than the capacity of a single drone, therefore one can use a swarm composed of several autonomous platforms. Usage of several drones possibly increases the number and/or the variety of embedded sensors, and the collaboration between them enhances the global system capabilities by supporting some form of resilience. However, swarming introduced new issues, such as communications and drone avoidance. Our path planning method considers these problematics and is then adapted for drones in swarm.

Keywords : Autonomous drone, autonomous UAV, swarm, mobility model, path calculation, distributed system, compactness
