
SÛRETÉ DES ESSAIS DE ROBOTS MOBILES
—
RÉSEAU DYNAMIQUE ET RESSOURCES LIMITÉES

Tuteur de stage :
Etienne Payet

Responsable de stage :
Rémy Courdier

Auteur :
Pierre Jaffuer
N° étudiant : 37007799

Encadrants :



Pierre Courtieu
Xavier Urbain



Christian Delhommé
Frédéric Mesnard
Étienne Payet

Un stage réalisé au laboratoire du LIM, en collaboration avec le laboratoire LIRIS

Remerciements

Je tiens à exprimer ma reconnaissance envers toutes les personnes qui ont contribué, de près ou de loin, à la réussite et à l'accomplissement de ce stage, en particulier je souhaite exprimer ma gratitude envers mes encadrants du LIM : Christian DELHOMMÉ, Frédéric MESNARD et Étienne PAYET pour m'avoir aiguillé sur cette thématique.

Je souhaite également exprimer ma gratitude envers mes encadrants du LIRIS : Pierre COURTIEU et Xavier URBAIN. Malgré la distance géographique qui nous sépare, ils m'ont apporté leur aide et leur encadrement depuis la France métropolitaine. Je leur suis reconnaissant pour leur disponibilité et pour avoir proposé ce sujet passionnant.

J'aimerais également remercier Marianne MORILLON, qui a été une source d'inspiration tout au long de la réalisation du protocole. Je remercie chaleureusement Youssef BOUDABBOUS qui m'a aidé et éclairé lorsque j'étais bloqué. Mes remerciements vont également à Nathan AKY, Rémy COURDIER et Olivier SEBASTIEN pour leurs conseils qui ont contribué à apaiser le stress de la poursuite de mes études. Je suis reconnaissant envers Anil CASSAM CHENAI et Mickael FRANCOMME pour m'avoir offert une précieuse opportunité d'expérience en entreprise pendant le Master. Leur formation m'a permis de rester concentré et a joué un rôle crucial à la fois sur le plan technique et organisationnel durant le stage.

Enfin, merci à mes amis et camarades de classe : Tristan LE LIDEC, Donovan ROULLAND et Jérémy ROBERT pour la bonne ambiance de travail.

À tous, je suis profondément reconnaissant pour votre précieux soutien.



« Premature optimization is the root of all evil (or at least most of it) in programming. »

~

Donald Knuth



« Une vision à 360° des vos composants permet de voir au-delà du Win32StaticDebug et de contempler le RichGraph scintillant. »

~

Donovan Roulent



« Les log sont comme des témoins silencieux des activités les plus intimes »

~

Tristan Le Lit Deck

UNIVERSITÉ DE LA RÉUNION

Faculté Sciences et Technologies
LIM

Abstract

Computer science student

SÛRETÉ DES ESSAIMS DE ROBOTS MOBILES – RÉSEAU DYNAMIQUE ET RESSOURCES LIMITÉES

by Pierre JAFFUER

The growing use of networks of autonomous robots can be seen in various fields, such as the exploration of other wise difficult-to-access or devastated environments. In these situations, it is possible to deploy a fleet of robots to establish a connection between a rescue team and their base. The aim of this internship is to design a protocol that will enable robots to be safely repatriated to their base, even when it is out of sight. As part of this work, a 3D simulator was simultaneously developed to facilitate the development of the return protocol. Emphasis was placed on justifying the different protocol approaches explored, and the results obtained during the course showed encouraging promise.

Key words : *oblivious robots, robot swarm, distributed protocol, simulation, formal verification, proof*

Résumé

Étudiant en informatique

SÛRETÉ DES ESSAIMS DE ROBOTS MOBILES – RÉSEAU DYNAMIQUE ET RESSOURCES LIMITÉES

par Pierre JAFFUER

L'utilisation croissante des réseaux de robots autonomes est observée dans divers domaines, tels que l'exploration d'environnements difficiles d'accès ou dévastés. Dans ces situations, il est possible de déployer une flotte de robots afin d'établir une connexion entre une équipe de secours et leur base. Le sujet du stage consiste à concevoir un protocole permettant le rapatriement sécurisé des robots vers la base, même lorsque celle-ci n'est pas visible. Dans le cadre de ce travail, un simulateur en 3D a été développé simultanément pour faciliter le développement du protocole de retour. L'accent a été mis sur la justification des différentes approches du protocole explorées et les résultats obtenus au cours du stage ont montré une promesse encourageante.

Mots-clés : *robots oblivious, essaim de robots, protocole distribué, simulation, vérification formelle, preuve*

Table des matières

1	Introduction	6
2	Gestion de projet	7
2.1	Contexte organisationnel	7
2.2	Outils utilisés et communications avec les encadrants	7
2.3	Méthode de travail et planification des tâches	7
2.4	Déroulement global du stage	8
3	Protocole de maintien de connexion existant	9
3.1	Champ de vision	9
3.2	Règle d'évolution des graphes de maintien de connexion	10
3.3	L'invariant assurant le maintien de la connexion	11
4	Contribution : protocole de retour	11
4.1	Suivre un chemin amenant à la base	11
4.1.1	Première idée : bi-partitionnement du graphe des drones	12
4.1.2	Idée retenue : élagage du graphe des drones	13
4.2	Revenir à la base en sécurité	14
4.2.1	Nouveau champ de vision	15
4.3	Différencier le retour du maintien de connexion	15
4.4	Description complète et informelle du Protocole final	16
4.5	Formalisation du protocole	18
5	Contribution : simulateur d'essaim de robots	23
5.1	Choix de la technologie	23
5.2	Architecture	23
5.3	Fonctionnalités	24
6	Conclusion	25
	Appendices	27
A	Organismes d'accueil	28
A.1	LIM	28
A.2	LIRIS	29
B	Tableaux de bord de suivi mensuel projet	31
C	Définitions et propriétés sur les graphes	33
D	Représentation UML de l'architecture du simulateur	34

Liste des Abréviations

LIM	Laboratoire d'Informatique et de Mathématiques
LIRIS	Laboratoire d'InfoRmatique en Image et Systèmes d'information
GUI	Graphical User Iinterface
WYSIWYG	What You See Is What You Get
IDE	Integrated Development Environment
FSYNC	Fully Synchronous

1 Introduction

L'utilisation de la robotique, en particulier en essaim, peu largement être utilisée pour rechercher des survivants dans des zones où des moyens conventionnels de communication ne peuvent être mis en place. Dans ces situations, une équipe d'intervention humaine au sol ne peut pas être laissée sans assistance ni compter sur une infrastructure de communication fixe. Par conséquent, une flotte de robots équipés de relais à portée limitée peut être déployée pour établir une connexion de proche en proche avec une base de secours. La propriété clé étudiée dans ce contexte est le maintien continu d'une telle connexion.

Cependant, lorsque des vies sont en jeu, il est crucial d'avoir des garanties solides de correction sur le comportement de l'essaim. Le laboratoire LIRIS travaille sur le projet SAP-PORO¹ qui vise à proposer un cadre formel prouvable (assisté par ordinateur grâce à l'assistant de preuve Coq) pour évaluer l'exactitude des protocoles distribués localisés au cœur des réseaux de capteurs mobiles dynamiques[4]. En effet, les réseaux de robots autonomes ont suscité un grand intérêt dans la communauté de l'algorithmique distribuée, l'industrie et les médias en raison de leurs nouvelles applications et de leur capacité à résoudre des problèmes complexes tels que l'exploration d'espaces difficiles d'accès ou dévastés par une catastrophe, la recherche de survivants, la décontamination et le démantèlement de centrales dangereuses.

L'objectif de ce stage est de concevoir un protocole permettant le rapatriement sécurisé des robots jusqu'à la base de manière à éviter toute collision, même lorsque la base est hors de vue. Ce protocole est destiné à être utilisé lors du retrait des robots du système. L'accent a été mis sur la justification des différentes approches explorées et les résultats obtenus au cours du stage ont montré une promesse encourageante. Enfin, un simulateur avec rendu 3D a été développé en parallèle afin de tester, visualiser et assister à l'élaboration du protocole de retour.

1. <https://sapporo.liris.cnrs.fr/>

2 Gestion de projet

2.1 Contexte organisationnel

Le LIM contribue à la formation par la recherche en collaborant avec différentes institutions, notamment l'UFR Sciences et Technologies, l'ESIROI et l'ESPE. Le LIM est également affilié à l'École doctorale STS et à la fédération de recherche OMNCG, ainsi qu'à l'UR. Depuis quelques temps, le LIM échange avec des membres du LIRIS qui est une UMR (Unité Mixte de Recherche) regroupant le CNRS, l'INSA de Lyon, l'Université Claude Bernard Lyon 1, l'Université Lumière Lyon 2 et l'École Centrale de Lyon. Le laboratoire se consacre à la recherche sur les enjeux du monde numérique tels que l'intelligence artificielle, l'analyse de données, la vision par ordinateur et la sécurité informatique.

J'ai effectué mon stage de recherche dans le domaine de la sécurité informatique sur les essais de robots mobiles. Ce stage résulte d'une collaboration entre le le LIM et le LIRIS.

2.2 Outils utilisés et communications avec les encadrants

Pour mener à bien mon travail de recherche et de développement, j'utilise plusieurs outils essentiels :

- Godot² : un moteur de jeu polyvalent et opensource que j'utilise pour la création et la simulation d'environnements virtuels. C'est la base sur laquelle repose mon simulateur.
- Git : système de contrôle de version qui me permet de gérer efficacement le suivi des modifications de mon code source
- Github³ : plateforme de collaboration et d'hébergement de code utilisant Git, où je partage le code de mon simulateur.
- L^AT_EX : un langage de composition de texte pour la rédaction de documents scientifiques que j'ai utilisé pour toutes mes productions écrites.
- PlantUML⁴ : un générateur de diagramme (Gantt, UML, activité, etc.) à partir d'un langage textuel simple.

J'ai fait plusieurs présentations au LIM pour faire le point sur l'avancement de mon stage. Les réunions se sont tenues en personne avec mes encadrants du LIM et en visioconférence avec ceux du LIRIS. Évidemment, j'ai aussi beaucoup communiqué par mail avec mes encadrants pour échanger des informations et discuter du projet.

2.3 Méthode de travail et planification des tâches

Dans le cadre de mon stage, l'organisation de mes tâches n'a pas été définie de manière stricte. J'ai bénéficié d'une grande liberté tant que mes objectifs étaient en adéquation avec l'orientation générale du stage : prototyper et tester des idées, trouver des arguments pour justifier informellement la validité du protocole, et, si possible, les démontrer formellement. J'ai décidé de développer un simulateur en parallèle de la conception du protocole de retour. Les tâches principales que je me suis imposées étaient les suivantes :

2. <https://godotengine.org/>

3. <https://github.com/>

4. <https://plantuml.com/fr/>

1. Développer le simulateur pour tester le protocole de maintien de connexion de Balabonski et al. ⁵[1]
2. Utiliser ce simulateur pour déterminer en priorité un moyen de retour vers la base en ignorant les collisions.
3. Modifier ce nouveau protocole en y ajoutant la gestion de l'évitement des collisions lors du retour.
4. Formaliser la définition du protocole.

2.4 Déroulement global du stage

Le diagramme de Gantt en Figure 1 ci-dessous offre une vue d'ensemble du stage, en mettant en évidence les différentes phases et les principales étapes réalisées. Toutefois, il est important de noter ce diagramme de Gantt ne fournit qu'une description générale des tâches et de leur chronologie. Pour obtenir une description plus détaillée et exhaustive du déroulement du stage, je vous invite à vous référer à l'annexe B.

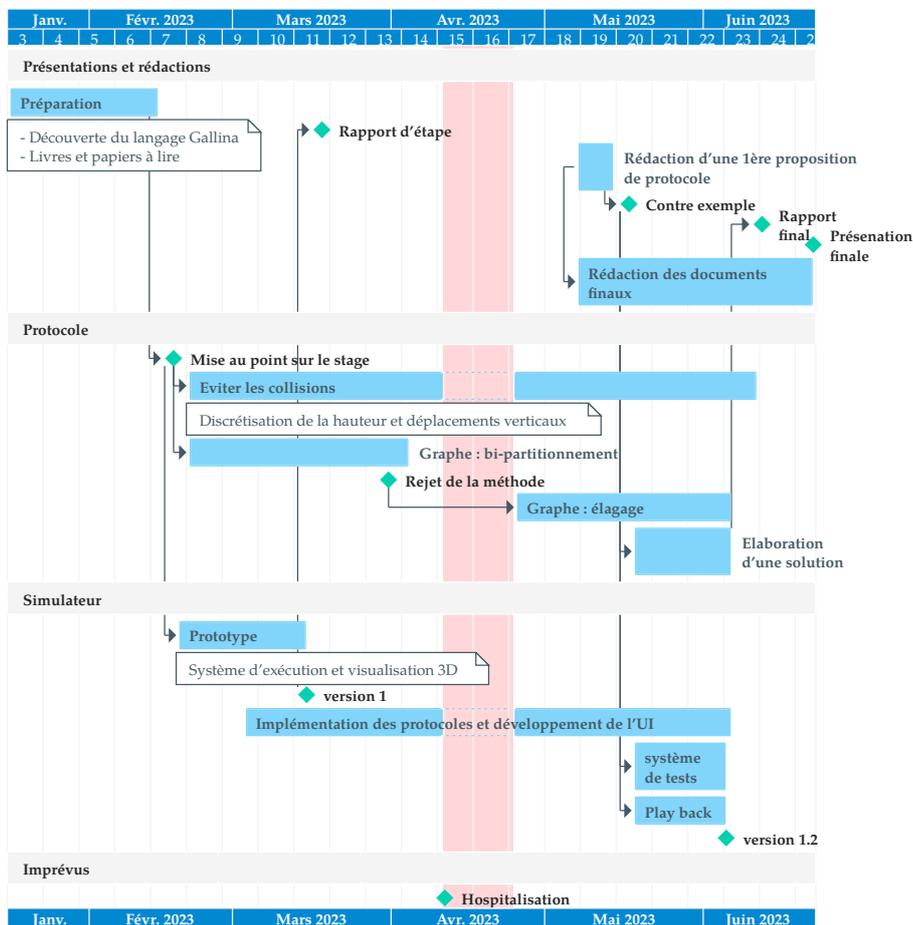


FIGURE 1 – Résumé du déroulement du stage.

5. Thibaut Balabonski, Pierre Courtieu, Robin Pelle, Lionel Rieg, Sébastien Tixeul, et Xavier Urbain

3 Protocole de maintien de connexion existant

Dans la suite du document, nous utiliserons les définitions sur les graphes données en l'annexe C.

La tâche qui m'a été assignée est de finaliser un protocole de maintien de connexion pour des robots oblivous fonctionnant en mode **FSYNC**, qui implique que chaque drone doit se conformer aux contraintes suivantes [1] :

- Déplacement rigide d'une distance D
- Posséder un identifiant unique dans \mathbb{N}
- Seuls les voisins d'identifiant inférieur sont visibles
- Ne pas se déplacer lorsque l'on avertit d'une collision
- Se déplacer :
 - cibler le voisin visible le plus proche en priorisant ceux sans avertissement de collision
 - S'y déplacer si la cible est supérieur à une distance de poursuite

À l'exception près du drone d'identifiant N°0 qui n'est pas contrôlé par le protocole et qui suit toujours l'équipe de recherche.

3.1 Champ de vision

Les drones sont donc munis de plusieurs tranches de visibilité définies ci-dessous [1] :

- Rayon maximal de vision noté D_m
- Rayon de détection de collision égale à la distance de déplacement (D)
- Rayon de poursuite de sa cible noté D_p
- Rayon d'avertissement de collision noté D_c

Afin de garantir que le graphe des connexions ne possède qu'une seule composante, la Figure 2 présente les contraintes liées aux déplacements rigides sur les différents rayons [1].

- $D_m = 7 \times D$
- $D_p = D_m - D$
- $D_c = 2 \times D$

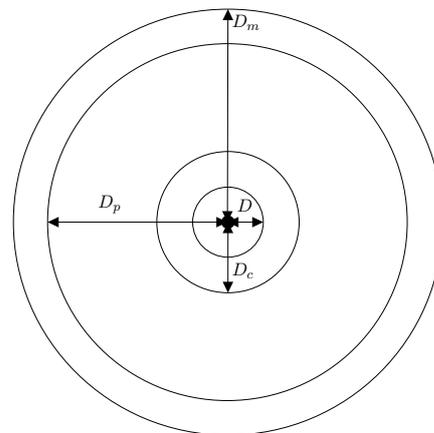


FIGURE 2 – Rayons de vision

3.2 Règle d'évolution des graphes de maintien de connexion

A chaque étape du système, nous représenterons les drones par un graphe orienté défini comme suit :

- Tous les sommets sont positionnés dans le plan de \mathbb{R}^2
- Tous les sommets sont identifiés par des entiers naturels tous disjoints.
- La base possède toujours l'id le plus grand du graphe.
- Les arêtes sont construites de sorte que pour n'importe quel sommet r , r pointe vers tous ses sommets d'id strictement inférieure situés à une distance $\leq D_m = 7D$
- Les sommets possèdent une information supplémentaire : lumière allumée ou éteinte

On notera (G_n) la suite des graphes obtenus après n exécutions du protocole du maintien de connexion, avec $n \in \mathbb{N}$. La suite des graphes est formée comme ceci :

Graphe initial G_0 :

- Il y a exactement 2 sommets étant la base et le sommet d'id 0 (représentant le drone maître) situé à une distance $< 3D$ de la base

Soit G_n le graphe formé à l'étape $n \in \mathbb{N}$, G_{n+1} est formé grâce à ces règles de construction :

- le sommet 0 se déplace d'une distance $\leq D$
- un nouveau sommet r d'id = $\max(\text{id des sommets} \neq \text{base}) + 1$ est ajouté sur la base si tous les sommets pointés par la base sont situés à une distance $\leq 3D$
- On considère maintenant un sommet r tel que $r \neq \text{base}$ et $r \neq 0$:
 - ★ Si r a sa lumière d'allumée alors :
 - r est éliminé si il pointe sur un sommet situé à une distance $\leq D$
 - r éteint sa lumière si tous les sommets pointés par r sont situés à une distance $> 2D$ et :
 - r ne se déplace pas si il pointe sur un sommet situé à une distance $< D_p = D_m - D$
 - si non, r choisi se déplacer d'une distance exactement D vers un des sommets qu'il pointe en choisissant en priorité ceux qui n'ont pas leur lumière d'allumée
 - si non, r garde sa lumière allumée et ne change pas de position
 - ★ Si r a sa lumière d'éteinte alors :
 - r ne se déplace pas et allume sa lumière si il pointe sur un sommet situé à une distance $\leq 2D$
 - r garde sa lumière éteinte et ne se déplace pas si il pointe sur un sommet situé à une distance $< D_p = D_m - D$
 - si non, r garde sa lumière éteinte et se déplace d'une distance exactement D vers un des sommets qu'il pointe en choisissant en priorité ceux qui n'ont pas leur lumière d'allumée

3.3 L'invariant assurant le maintien de la connexion

L'invariant qui assure la communication entre la base et le drone maître (le drone d'id 0) est le suivant :

(I) «Tous les drones relais (en considérant la base comme un drone relais) ont au moins un drone d'identifiant strictement inférieur de visible» [1]

Théorème 1. $\forall n \in \mathbb{N}$, $G(n)$ est faiblement connexe et il existe un chemin à partir du sommet de la base vers le sommet 0.

Démonstration. Soit $n \in \mathbb{N}$, il suffit de montrer que tous les sommets de G_n différents de 0 admettent un chemin vers le sommet 0.

Soit r un tel sommet, l'invariant (I) assure que ce sommet pointe sur un autre strictement inférieur. Ainsi la suite des identifiants observée en choisissant récursivement un sommet pointé est strictement décroissante et donc atteint le sommet 0. \square

Ainsi, la base et l'équipe de recherche ont toujours la possibilité de communiquer.

4 Contribution : protocole de retour

Les drones qui seront en retour vers la base devront suivre un chemin amenant jusqu'à la base en observant les drones qui maintiennent la connexion.

Soit G un des graphes défini en section 3.2. Nous utiliserons le terme « source » pour désigner un sommet de G distinct de celui de la base, ayant un degré entrant 0, c'est-à-dire un sommet distinct de celui de la base qui n'est pas pointé par un autre sommet. Par conséquent, une source correspond à un drone ayant le plus grand identifiant parmi ses voisins.

4.1 Suivre un chemin amenant à la base

Soit G un des graphes défini en section 3.2. Nous ignorons pour le moment les collisions entre les drones.

Théorème 2. G est un graphe fini, orienté, acyclique et faiblement connexe.

Démonstration. G est fini et orienté par définition. G est faiblement connexe d'après le théorème 1.

Supposons par l'absurde que G admettent un cycle C . Considérons le sommet s d'identifiant maximal sur C . s existe car d'après la définition de G , les sommets sont étiquetés dans \mathbb{N} et tous disjoints.

s étant dans C alors il admet un antécédent qui à forcément un identifiant inférieur, ce qui est absurde d'après la définition de G . D'où G est acyclique.

Ainsi, G est bien un graphe fini, orienté, acyclique et faiblement connexe. \square

L'idée proposée consiste à permettre à un drone en retour de visualiser tous les drones relais présents dans un rayon de distance égal ou inférieur à $2D$, sans tenir compte de leur identifiant. Le drone en retour se dirigera toujours vers le drone relais visible ayant l'identifiant le plus élevé. D'après le théorème précédent, G étant acyclique, ce choix de déplacement ne peut amener le drone à « boucler » éternellement.

Cependant, cette méthode simple présente un problème majeur : si le drone en retour se trouve sur une source du graphe G , il risque de ne jamais revenir à la base. Cela peut être illustré par un exemple, comme le montre la Figure 3.

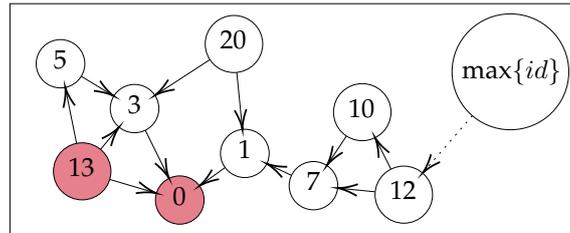


FIGURE 3 – Un drone en retour au niveau de 0 se retrouvera bloqué en 13

4.1.1 Première idée : bi-partitionnement du graphe des drones

Afin de palier au problème mentionné précédemment, nous souhaitons mettre en place deux mesures préventives :

- Empêcher un drone en retour de se diriger vers une source
- Rediriger un drone en retour se trouvant sur un chemin menant à une source vers un chemin menant à la base

Il suffit de partitionner le graphe des drones relais en deux parties :

- Partition A : les sommets accessibles depuis la base
- Partition B : les sommets non accessibles depuis la base

Par conséquent, lorsqu'un drone est en retour, il prendra systématiquement l'une des deux décisions suivantes :

- Si aucun sommet de A sont visibles, alors il se dirigera vers le sommet visible de B d'identifiant minimum
- Si non, il se dirigera vers le sommet visible de A d'identifiant maximum

Afin de permettre aux drones en retour de prendre ces décisions, il est nécessaire de fournir à chaque drone relais un moyen de communiquer sa partition respective. Une approche possible serait d'équiper les drones relais d'une lumière supplémentaire qui s'activera à chaque étape si ce dernier détermine qu'il est une source (tous ses voisins visibles sont d'identifiant inférieur) ou s'il n'a que des prédécesseurs conduisant à un chemin menant à une source (tous ses voisins visibles d'identifiant supérieur ont leur lumière de partitionnement allumée).

Cet algorithme distribué permet de partitionner le graphe en deux parties distinctes :

- La partition B dont tous les sommets ont cette nouvelle lumière allumée
- La partition A dont tous ses sommets ont cette nouvelle lumière éteinte

Cette procédure est illustrée par l'exemple présenté dans la Figure 4.

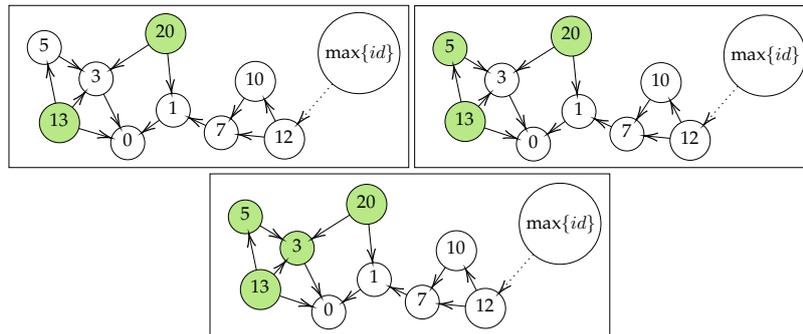


FIGURE 4 – Algorithme itératif de partitionnement : partition B en vert et partition A en blanc

Cette solution présente un inconvénient majeur : l'installation d'une nouvelle lumière.

4.1.2 Idée retenue : élagage du graphe des drones

En reprenant les mêmes conventions que la section précédente et la section 3.2, nous avons que les drones de la partition B deviennent pratiquement inutiles pour maintenir la connexion. Ceci est dû aux règles d'évolution des graphes présentées dans la section 3.2, qui dépendent uniquement de conditions impliquant des drones d'identifiants inférieurs. Par conséquent, étant donné qu'aucun drone ne peut voir les sources, ces derniers peuvent être supprimés et le graphe résultant respecte toujours l'invariant de connexité explicité en section .

L'idée consiste à continuellement éliminer les drones qui détectent être une source. Un exemple d'exécution de cet algorithme itératif est présenté en Figure 5.

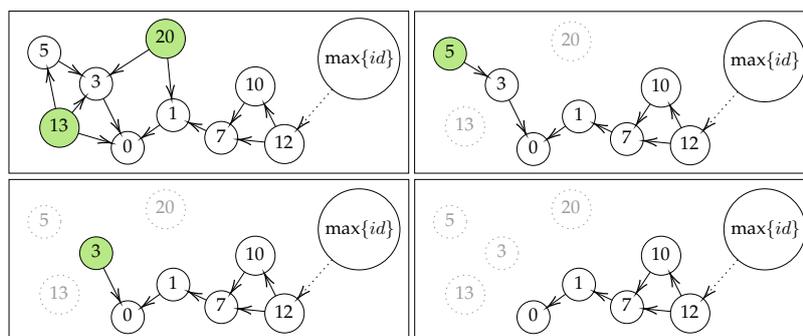


FIGURE 5 – Élimination itérative des sources (en vert).

Il est intéressant de noter que le sous-graphe obtenu après convergence correspond exactement à la partition A (les sommets accessibles depuis la base). Cela signifie qu'un drone en retour n'a plus qu'à se diriger vers le sommet visible d'identifiant maximum.

Il est important de noter qu'un drone en retour peut effectivement se retrouver sur une source. Cependant, puisque la source est destinée à être éliminée à chaque itération,

le drone en retour sera naturellement attiré vers le prochain sommet visible d'identifiant maximum. Ce processus se répétera ainsi de suite, permettant au drone en retour de se rapprocher continuellement du sommet visible suivant avec l'identifiant maximum et donc d'arriver à la base.

4.2 Revenir à la base en sécurité

Rappel : en cas de déplacement, les drones se déplacent d'une distance fixe notée D ,

Définition 1. (*distance horizontale* Nous dirons « distance horizontale » entre deux drones pour représenter la distance euclidienne sur \mathbb{R}^R entre deux drones. La composante verticale est ignorée.

Les drones seront équipés d'un axe vertical qui leur permet de percevoir leur hauteur relative par rapport à leurs voisins, grâce à des capteurs tels que des accéléromètres par exemple.

Lorsqu'un drone est éliminé suite à une collision ou s'il était une source, ce dernier effectue un déplacement uniquement vertical d'une distance fixe D vers le haut. La distance maximale entre 2 drones du plan de maintien de la connexion étant de $7D$ (voir règles en section 3.2), afin d'assurer que le drone éliminé voit encore quelqu'un après sa montée, il suffit d'élargir son champ de vision horizontal d'une distance D supplémentaire ce qui ramène à considérer une vision horizontale maximale de $8D$. De plus, pour éviter les collisions avec des drones qui se déplacent en dessous d'eux, les drones en mode retour maintiennent une distance verticale supplémentaire avec ceux situés en dessous et à une distance horizontale inférieure ou égale à $7D$. En conséquence, un drone en retour peut effectuer un déplacement vertical de $2D$ sur deux cycles d'exécution du protocole (Figure 6).

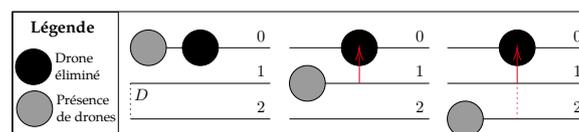


FIGURE 6 – Élimination d'un drone.

Les drones possèdent maintenant un champ de vision cylindrique de rayon $8D$ et de hauteur $2D$ décrit en section 4.2.1.

L'objectif est d'obliger les drones en retour à suivre horizontalement le drone ayant l'identifiant maximal parmi ceux visibles en dessous de lui. Si aucun drone en dessous n'est visible, cela signifie que le drone suivi à l'étape précédente a soit été capturé par la base, soit s'est déplacé vers le bas. Dans ce cas, il suffit de descendre également jusqu'à rattraper verticalement sa cible précédente ou de se trouver dans la zone de capture de la base.

Voici la prise de décision d'un drone en retour noté r (dans l'ordre) :

1. Si la base est visible à une distance horizontale inférieure $\leq 7D$ et située verticalement à $2D$ en dessous, alors je suis capturé par la base et je ne suis plus visible pour les autres drones
2. Si il existe un drone visible situé à une distance horizontale $\leq 7D$ en dessous de r alors r monte de D
3. Si il existe un drone visible situé à une distance horizontale $\leq 7D$ et d'identifiant inférieur au même niveau que r alors r monte de D
4. Si aucun drone en dessous n'est visible alors r descend de D
5. Si non, r se déplace horizontalement vers le drone d'identifiant maximal parmi ceux visibles en dessous de r

4.2.1 Nouveau champ de vision

La vision d'un drone peut être modélisée par trois disques de rayon $D_m = 8D$, centrés sur le drone et espacés verticalement de D vers le bas. En d'autres termes, la vision du drone peut être assimilée à trois coupes d'un cylindre d'une hauteur de $2D$ et d'un rayon de D_m . La Figure 7 illustre cette modélisation où les couches 1 et 2 sont utilisées pour observer les drones situés en dessous, et la couche 0 est utilisée pour détecter les drones situés à la même hauteur.

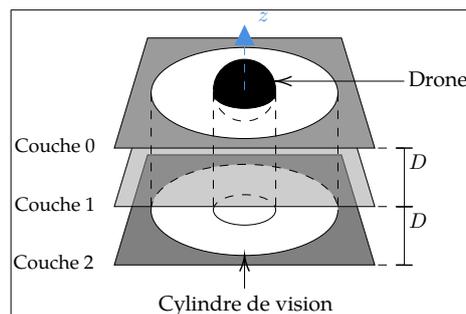


FIGURE 7 – Représentation en 3 dimensions.

4.3 Différencier le retour du maintien de connexion

Étant donné que les robots n'ont pas de mémoire et ne peuvent pas se rappeler qu'ils sont en train de retourner à la base après avoir éliminé un drone, il est nécessaire de déduire leur état de retour à partir d'une observation.

Il est plus aisé de déterminer si un drone est dans le plan de maintien de la connexion que s'il est en mode retour. Comme expliqué dans la section précédente, nous pouvons observer que chaque drone dans le plan de maintien de la connexion satisfait les invariants suivants :

1. Il n'y a aucun drone de visible en dessous.
2. Au moins un drone est visible dans le même plan à une distance $\leq 7D$.

Toutefois, comme cela est illustré dans la Figure 8, ces deux conditions ne sont pas suffisantes pour distinguer les drones en retour des drones qui maintiennent la connexion.

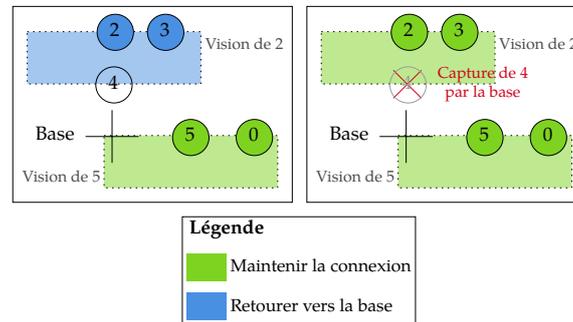


FIGURE 8 – 2 et 3 se retrouvent dans la même configuration que 5 et 0.

Pour résoudre ce problème, nous ajoutons une lumière supplémentaire aux drones afin d'indiquer aux autres robots qu'ils sont en train de retourner à la base. De cette manière, un drone est considéré comme étant dans le plan de maintien de la connexion si et seulement si les trois conditions suivantes sont remplies :

1. Il n'y a aucun drone de visible en dessous.
2. Au moins un drone est visible dans le même plan à une distance $\leq 7D$.
3. Tous les voisins du même plan n'ont pas leur lumière de retour d'allumée.

Cette lumière supplémentaire permet de verrouiller l'état de retour entre voisins, comme illustré dans la Figure 9.

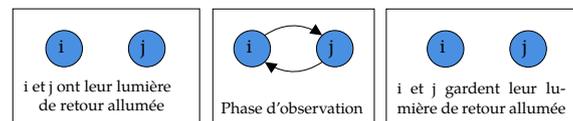


FIGURE 9 – Co-verrouillage du mode de retour grâce aux lumières d'indication de retour.

4.4 Description complète et informelle du Protocole final

Définition 2. (Déplacement fixe)

Si un drone effectue un déplacement, ce dernier se déplace toujours d'une distance fixe noté ici D .

Définition 3. (État d'un drone)

Un drone est représenté à chaque instant par son état :

- Un identifiant unique représenté par un entier naturel
- S'il est déployé ou non (en attente à la base)
- Si sa lumière d'avertissement de collision est allumée ou non
- Si sa lumière d'avertissement de retour est allumée ou non

- Sa position dans \mathbb{R}^3 .

Définition 4. (*La base*)

La base est représentée par un drone déployé d'identifiant toujours plus grand que les autres drones. La base ne se déplace jamais et n'allume jamais ses lumières. La base n'exécute pas le protocole décrit dans cette section.

Définition 5. (*L'équipe de recherche*)

L'équipe de recherche est représentée par le drone déployé d'identifiant 0, ce dernier n'allume jamais ses lumières et est toujours verticalement positionné au niveau de la base. Le déplacement horizontal de ce dernier ne dépend pas du protocole.

Définition 6. (*être visible*)

Soit r un drone déployé autre que la base. Soit r' un drone différent de r . r voit r' si et seulement si :

- r' est à une distance horizontale $\leq 8D$ de r
- r' est situé en dessous et à une distance verticale $< 2D$ ou dans le même plan que r
- r' est déployé

Définition 7. (*Être un drone relais*)

Soit r un drone déployé autre que la base. r est un drone relais (qui est sur le plan du maintien de la connexion) si et seulement si :

1. Il n'y a aucun drone de visible en dessous de r
2. Il existe au moins un drone r' situé à une distance horizontale $\leq 7D$ de r et d'identifiant inférieur à celui de r
3. Tous les drones visibles par r ont leur lumière de retour éteinte

Définition 8. (*Être une source*)

Soit r un drone déployé autre que la base. r est une source si et seulement si :

1. Il est un drone relais
2. Tous les drones visibles par r situé à une distance horizontale $\leq 7D$ ont un identifiant inférieur à celui de r

A chaque étape d'exécution du protocole, les règles suivantes sont appliquées :

Règle 1. (*Déploiement d'un nouveau drone*)

un nouveau drone d'identifiant = $\max(\text{identifiants des drones déployés}) + 1$ est déployé sur la position de la base si et seulement si il n'y a aucun drone de déployé sur le même plan que la base et situé à une distance $\leq 3D$ de la base.

Règle 2. (*Evolution des drones du système*)

Soit r un drone déployé autre que la base et l'équipe de recherche. r évolue en suivant ces règles dans l'ordre (r éteint toujours ses lumières sauf contre indication) :

1. Si r est un drone relais alors :
 - (a) Si r est une source alors il allume sa lumière de d'avertissement de retour et monte

- (b) Si non :
- i. S'il existe un drone visible situé à une distance horizontale $\leq D$ alors r allume sa lumière d'avertissement de retour et monte
 - ii. S'il existe un drone visible situé à une distance horizontale $\leq 2D$ alors r ne se déplace pas et allume sa lumière d'avertissement de collision
 - iii. S'il existe un drone visible situé à une distance horizontale $< 6D$ alors r ne se déplace pas
 - iv. Si non, r se déplace dans le plan vers un des drones visibles d'identifiant inférieur et situé à une distance horizontale $\leq 7D$
2. si non, r allume sa lumière d'avertissement de retour et :
- (a) S'il existe un drone visible situé à une distance horizontale $\leq 7D$ alors r monte
 - (b) Si r ne voit personne ou si ses drones visibles sont situés à une distance horizontale $> 7D$ alors r descend
 - (c) Si la base est située à une distance horizontale $\leq D$ et est positionnée à $2D$ en dessous alors r est capturé par la base et n'est plus en mode déployé
 - (d) Si non, r se déplace horizontalement vers le drone visible d'identifiant le plus élevé situé en dessous de r (distance verticale $\neq 0$)

4.5 Formalisation du protocole

Définition 9. (*Ensemble des valeurs booléennes*)

Nous noterons \mathbb{B} l'ensemble $\{\top, \perp\}$ des valeurs booléennes.

L'espace vertical de vol des drones est subdivisée en « plans horizontaux » superposés et espacés d'une distance D . Par convention, les plans sont numérotés par ordre croissant de la hauteur qu'ils représentent et le plan 0 représente le plan de maintien de la connexion. Nous ignorons les plans en dessous du plan 0. Ainsi, la position des drones est représentée par $\mathbb{R}^2 \times \mathbb{N}$, où la première composante correspond aux coordonnées dans le plan et la seconde composante correspond au numéro du plan en question.

Nous représenterons un drone par ses états :

- $id \in \mathbb{N}$ (identifiant unique)
- $DP \in \mathbb{B}$ (déployé ou en attente à la base)
- $LC \in \mathbb{B}$ (lumière d'avertissement de collision)
- $LR \in \mathbb{B}$ (lumière d'avertissement de retour)
- $position \in \mathbb{R}^2$ (position horizontale)
- $plan \in \mathbb{N}$ (N° du plan horizontal)

Définition 10. (*Drone byzantin*)

Nous appellerons « drone byzantin » un drone du système dont l'évolution de ses états ne dépend pas du protocole.

Définition 11. (Suite des ensembles des états des drones)

Soit $k \in \mathbb{N}$ le nombre de drones total du système.

Nous noterons **Drones** = $(\mathbb{N} \times \mathbb{B} \times \mathbb{B} \times \mathbb{B} \times \mathbb{R}^2 \times \mathbb{N})^k$ l'ensemble des états des drones pour chaque étape du système et nous noterons $(S_n) \in (\mathbf{Drones})^{\mathbb{N}}$ la suite des états de tous les drones du système.

Axiom 1. (Unicité des identifiants)

$$\forall n \in \mathbb{N}, \forall r = (\text{id}_r, \dots) \in S_n, \forall r' = (\text{id}_{r'}, \dots) \in S_n \setminus \{r\}, \text{id}_r \neq \text{id}_{r'}$$

Axiom 2. (Ensemble des drone byzantin)

$$\forall n \in \mathbb{N}, \forall r = (\text{id}, \dots) \in S_n, (r \in \mathbf{Byzantin}) \iff (\text{id} = 0)$$

Axiom 3. (Propriété des drones byzantins)

$$\begin{aligned} \forall n \in \mathbb{N}, \forall r = (\text{id}, \text{DP}, \text{LC}, \text{LR}, \text{position}, \text{plan}) \in S_n, \\ (r \in \mathbf{Byzantin}) \implies (\text{DP} = \top) \wedge (\text{LC} = \perp) \wedge (\text{LR} = \perp) \wedge (\text{plan} = 0) \end{aligned}$$

Théorème 3. (Unicité du drone byzantin)

A chaque étape du système, il existe un unique drone byzantin et celui-ci à pour identifiant 0.

Démonstration. Soient $n \in \mathbb{N}$ et $r = (\text{id}, \text{DP}, \text{LC}, \text{LR}, \text{position}, \text{plan}) \in S_n$ tel que r représente un drone byzantin. D'après l'axiome 2 nous avons $\text{id} = 0$. De plus, d'après l'axiome 1 nous avons que r est l'unique drone d'identifiant 0. \square

Définition 12. (L'équipe de secours)

Nous assimilons l'équipe de secours à l'unique drone byzantin (identifiant 0).

Définition 13. (Distance dans le plan)

Nous noterons d pour la distance euclidienne sur \mathbb{R}^2

Définition 14. (Distance de déplacement)

Tous les drones se déplacent, si déplacement il y a lieu, d'une distance fixe notée ici D .

Axiom 4. (Déplacement fixe de D)

$$\begin{aligned} \forall n \in \mathbb{N}, \forall r = (\text{id}_r, \dots, \text{position}_r, \text{plan}_r) \in S_n, \forall r' = (\text{id}_{r'}, \dots, \text{position}_{r'}, \text{plan}_{r'}) \in S_{n+1}, \\ (\text{DP}_r = \top) \wedge (\text{id}_r = \text{id}_{r'}) \wedge (\text{position}_r \neq \text{position}_{r'}) \implies (d(\text{position}_r, \text{position}_{r'}) = D) \end{aligned}$$

Axiom 5. (Observation « obs »)

$$\begin{aligned} \forall n \in \mathbb{N}, \forall r = (\text{id}_r, \text{DP}_r, \text{LC}_r, \text{LR}_r, \text{position}_r, \text{plan}_r) \in S_n, \\ \forall r' = (\text{id}_{r'}, \text{DP}_{r'}, \text{LC}_{r'}, \text{LR}_{r'}, \text{position}_{r'}, \text{plan}_{r'}) \in S_n, (r' \in \text{obs}_r) \iff ((\text{id}_{r'} \neq \text{id}_r) \wedge \\ (d(\text{position}_{r'}, \text{position}_r) \leq 8D) \wedge (\text{DP}_{r'} = \top) \wedge (0 \leq (\text{plan}_r - \text{plan}_{r'}) \leq 2)) \end{aligned}$$

Axiom 6. (Être un drone « relais »)

$$\begin{aligned} \forall n \in \mathbb{N}, \forall r = (\text{id}_r, \text{DP}_r, \text{LC}_r, \text{LR}_r, \text{position}_r, \text{plan}_r) \in S_n, \\ (r \in \mathbf{Relais}) \iff (\text{DP}_r = \top) \wedge \\ (\forall r' = (\dots, \text{LR}_{r'}, \text{position}_{r'}, \text{plan}_{r'}) \in \text{obs}_r, (\text{plan}_r = \text{plan}_{r'}) \wedge (\text{LR}_{r'} = \perp)) \wedge \\ (\exists r' = (\text{id}_{r'}, \dots, \text{position}_{r'}, \dots) \in \text{obs}_r, (d(\text{position}_r, \text{position}_{r'}) \leq 7D) \wedge (\text{id}_r > \text{id}_{r'})) \end{aligned}$$

Axiom 7. (*Être une « source »*)

$$\begin{aligned} \forall n \in \mathbb{N}, \forall r = (\text{id}_r, \text{DP}_r, \text{LC}_r, \text{LR}_r, \text{position}_r, \text{plan}_r) \in S_n, \\ (r \in \mathbf{Source}) \iff (\text{DP}_r = \top) \wedge (r \in \mathbf{Relais}) \wedge \\ (\forall r' = (\text{id}_{r'}, \dots, \text{position}_{r'}, \dots) \in \text{obs}_r, (d(\text{position}_r, \text{position}_{r'}) \leq 7D) \implies (\text{id}_{r'} < \text{id}_r)) \end{aligned}$$

Axiom 8. (*Allumer la lumière de retour*)

$$\begin{aligned} \forall n \in \mathbb{N}, \forall r = (\text{id}_r, \text{DP}_r, \text{LC}_r, \text{LR}_r, \text{position}_r, \text{plan}_r) \in S_n, \\ \forall r' = (\text{id}_{r'}, \text{DP}_{r'}, \text{LC}_{r'}, \text{LR}_{r'}, \text{position}_{r'}, \text{plan}_{r'}) \in S_{n+1}, \\ (\text{id}_r = \text{id}_{r'}) \wedge (\text{DP}_r = \top) \wedge ((r \in \mathbf{Source}) \vee (r \notin \mathbf{Relais})) \implies (\text{LR}_{r'} = \top) \end{aligned}$$

Axiom 9. (*La base ne capture que des drones en retour à une distance de D et à 2 couches au dessus*)

$$\begin{aligned} \forall n \in \mathbb{N}, \forall r = (\text{id}_r, \text{DP}_r, \text{LC}_r, \text{LR}_r, \text{position}_r, \text{plan}_r) \in S_n, \\ \forall r' = (\text{id}_{r'}, \text{DP}_{r'}, \text{LC}_{r'}, \text{LR}_{r'}, \text{position}_{r'}, \text{plan}_{r'}) \in S_{n+1}, (\text{id}_r = \text{id}_{r'}) \\ \wedge (\text{DP}_r = \top) \wedge (d(\text{position}_r, \text{base}) \leq D) \wedge (\text{plan}_r = 2) \wedge (r \notin \mathbf{Relais}) \implies (\text{DP}_{r'} = \perp) \end{aligned}$$

Axiom 10. (*Allumer la lumière d'avertissement de collision*)

$$\begin{aligned} \forall n \in \mathbb{N}, \forall r = (\text{id}_r, \text{DP}_r, \text{LC}_r, \text{LR}_r, \text{position}_r, \text{plan}_r) \in S_n, \\ \forall r' = (\text{id}_{r'}, \text{DP}_{r'}, \text{LC}_{r'}, \text{LR}_{r'}, \text{position}_{r'}, \text{plan}_{r'}) \in S_{n+1}, (\text{id}_r = \text{id}_{r'}) \wedge (r \in \mathbf{Relais}) \wedge \\ (\exists v = (\text{id}_v, \dots, \text{position}_v, \dots) \in \text{obs}_r, (d(\text{position}_v, \text{position}_r) \leq 2D) \wedge (\text{id}_v < \text{id}_r)) \implies \\ (\text{LC}_{r'} = \top) \end{aligned}$$

Axiom 11. (*Pas de déplacements si on allume sa lumière de collision*)

$$\begin{aligned} \forall n \in \mathbb{N}, \forall r = (\text{id}_r, \text{DP}_r, \text{LC}_r, \text{LR}_r, \text{position}_r, \text{plan}_r) \in S_n, \\ \forall r' = (\text{id}_{r'}, \text{DP}_{r'}, \text{LC}_{r'}, \text{LR}_{r'}, \text{position}_{r'}, \text{plan}_{r'}) \in S_{n+1}, \\ (\text{id}_r = \text{id}_{r'}) \wedge (\text{LC}_{r'} = \top) \implies (\text{position}_r = \text{position}_{r'}) \wedge (\text{plan}_{r'} = \text{plan}_r) \end{aligned}$$

Axiom 12. (*Pas de déplacements horizontaux lors d'un changement de hauteur*)

$$\begin{aligned} \forall n \in \mathbb{N}, \forall r = (\text{id}_r, \text{DP}_r, \text{LC}_r, \text{LR}_r, \text{position}_r, \text{plan}_r) \in S_n, \\ \forall r' = (\text{id}_{r'}, \text{DP}_{r'}, \text{LC}_{r'}, \text{LR}_{r'}, \text{position}_{r'}, \text{plan}_{r'}) \in S_{n+1}, \\ (\text{id}_r = \text{id}_{r'}) \wedge (\text{plan}_{r'} \neq \text{plan}_r) \implies (\text{position}_r = \text{position}_{r'}) \end{aligned}$$

Axiom 13. (*Un drone relais monte s'il est une source*)

$$\begin{aligned} \forall n \in \mathbb{N}, \forall r = (\text{id}_r, \text{DP}_r, \text{LC}_r, \text{LR}_r, \text{position}_r, \text{plan}_r) \in S_n, \\ \forall r' = (\text{id}_{r'}, \text{DP}_{r'}, \text{LC}_{r'}, \text{LR}_{r'}, \text{position}_{r'}, \text{plan}_{r'}) \in S_{n+1}, \\ (\text{id}_r = \text{id}_{r'}) \wedge (r \in \mathbf{Source}) \implies (\text{plan}_{r'} = \text{plan}_r + 1) \end{aligned}$$

Axiom 14. (*Un drone relais monte pour éviter une collision*)

$$\begin{aligned} \forall n \in \mathbb{N}, \forall r = (\text{id}_r, \text{DP}_r, \text{LC}_r, \text{LR}_r, \text{position}_r, \text{plan}_r) \in S_n, \\ \forall r' = (\text{id}_{r'}, \text{DP}_{r'}, \text{LC}_{r'}, \text{LR}_{r'}, \text{position}_{r'}, \text{plan}_{r'}) \in S_{n+1}, (\text{id}_r = \text{id}_{r'}) \wedge (r \in \mathbf{Relais}) \wedge \\ (\exists v = (\text{id}_v, \dots, \text{position}_v, \dots) \in \text{obs}_r, (d(\text{position}_v, \text{position}_r) \leq D) \wedge (\text{id}_v < \text{id}_r)) \implies \\ (\text{plan}_{r'} = \text{plan}_r + 1) \end{aligned}$$

Axiom 15. (Un drone en retour garde une marge verticale avec ses voisins 1 cran en dessous)

$$\begin{aligned} \forall n \in \mathbb{N}, \forall r = (\text{id}_r, \text{DP}_r, \text{LC}_r, \text{LR}_r, \text{position}_r, \text{plan}_r) \in S_n, \\ \forall r' = (\text{id}_{r'}, \text{DP}_{r'}, \text{LC}_{r'}, \text{LR}_{r'}, \text{position}_{r'}, \text{plan}_{r'}) \in S_{n+1}, (\text{id}_r = \text{id}_{r'}) \wedge (\text{DP}_r = \top) \wedge \\ (\exists v = (\dots, \text{position}_v, \text{plan}_v) \in \text{obs}_r, (d(\text{position}_v, \text{position}_r) \leq 7D) \wedge (\text{plan}_r = \text{plan}_v + 1)) \\ \implies (\text{plan}_{r'} = \text{plan}_r + 1) \end{aligned}$$

Axiom 16. (Un drone en retour monte afin de s'isoler des voisins du même plan)

$$\begin{aligned} \forall n \in \mathbb{N}, \forall r = (\text{id}_r, \text{DP}_r, \text{LC}_r, \text{LR}_r, \text{position}_r, \text{plan}_r) \in S_n, \\ \forall r' = (\text{id}_{r'}, \text{DP}_{r'}, \text{LC}_{r'}, \text{LR}_{r'}, \text{position}_{r'}, \text{plan}_{r'}) \in S_{n+1}, (\text{id}_r = \text{id}_{r'}) \wedge (\text{DP}_r = \top) \wedge (r \notin \mathbf{Relais}) \\ (\exists v = (\text{id}_v, \dots, \text{position}_v, \text{plan}_v) \in \text{obs}_r, (d(\text{position}_v, \text{position}_r) \leq 7D) \wedge (\text{id}_v < \text{id}_r) \wedge (\text{plan}_r = \text{plan}_v)) \\ \implies (\text{plan}_{r'} = \text{plan}_r + 1) \end{aligned}$$

Axiom 17. (Un drone en retour descend pour essayer de retrouver un drone visible en dessous)

$$\begin{aligned} \forall n \in \mathbb{N}, \forall r = (\text{id}_r, \text{DP}_r, \text{LC}_r, \text{LR}_r, \text{position}_r, \text{plan}_r) \in S_n, \\ \forall r' = (\text{id}_{r'}, \text{DP}_{r'}, \text{LC}_{r'}, \text{LR}_{r'}, \text{position}_{r'}, \text{plan}_{r'}) \in S_{n+1}, (\text{id}_r = \text{id}_{r'}) \wedge (\text{DP}_r = \top) \wedge (r \notin \mathbf{Relais}) \wedge \\ (\forall v = (\text{id}_v, \dots, \text{position}_v, \dots) \in \text{obs}_r, (\text{plan}_v = \text{plan}_r \implies (d(\text{position}_v, \text{position}_r) > 7D) \vee \\ (\text{id}_v > \text{id}_r)) \wedge (\text{plan}_v \neq \text{plan}_r - 1) \wedge (\text{plan}_v = \text{plan}_r - 2 \implies d(\text{position}_v, \text{position}_r) > 8D)) \\ \implies (\text{plan}_{r'} = \text{plan}_r - 1) \end{aligned}$$

Théorème 4. (Déplacement vertical fixe de 1)

$$\begin{aligned} \forall n \in \mathbb{N}, \forall r = (\text{id}_r, \dots, \text{position}_r, \text{plan}_r) \in S_n, \forall r' = (\text{id}_{r'}, \dots, \text{position}_{r'}, \text{plan}_{r'}) \in S_{n+1}, \\ (\text{DP}_r = \top) \wedge (\text{id}_r = \text{id}_{r'}) \wedge (\text{plan}_r \neq \text{plan}_{r'}) \implies (\text{plan}_{r'} = \text{plan}_r + 1) \vee (\text{plan}_{r'} = \text{plan}_r - 1) \end{aligned}$$

Démonstration. Trivial, voir les axiomes 13, 14, 15, 16 et 17. \square

Axiom 18. (Un drone en retour, s'il ne monte pas ou ne descend pas, se dirige vers le drone visible en dessous d'id le plus grand)

$$\begin{aligned} \forall n \in \mathbb{N}, \forall r = (\text{id}_r, \text{DP}_r, \text{LC}_r, \text{LR}_r, \text{position}_r, \text{plan}_r) \in S_n, \\ \forall r' = (\text{id}_{r'}, \text{DP}_{r'}, \text{LC}_{r'}, \text{LR}_{r'}, \text{position}_{r'}, \text{plan}_{r'}) \in S_{n+1}, \\ (\text{id}_r = \text{id}_{r'}) \wedge (\text{DP}_r = \top) \wedge (r \notin \mathbf{Relais}) \wedge (\text{plan}_{r'} = \text{plan}_r) \\ \implies (\exists v = (\text{id}_v, \dots, \text{position}_v, \text{plan}_v) \in \text{obs}_r, (\text{plan}_v = \text{plan}_r) \wedge \\ (\forall v' = (\text{id}_{v'}, \dots) \in \text{obs}_r, (\text{plan}_v = \text{plan}_{v'}) \implies (\text{id}_v > \text{id}_{v'})) \\ \implies (\text{position}_{r'} = \text{position}_r + \frac{\text{position}_v - \text{position}_r}{d(\text{position}_v, \text{position}_r)} \times D)) \end{aligned}$$

Axiom 19. (Ensemble des drones visibles par un drone relais et qui n'ont pas leur lumière d'avertissement allumée)

$$\begin{aligned} \forall n \in \mathbb{N}, \forall r = (\text{id}_r, \text{DP}_r, \text{LC}_r, \text{LR}_r, \text{position}_r, \text{plan}_r) \in S_n, \\ \forall r' = (\text{id}_{r'}, \text{DP}_{r'}, \text{LC}_{r'}, \text{LR}_{r'}, \text{position}_{r'}, \text{plan}_{r'}) \in S_n, \\ (r \in \mathbf{Relais}) \implies ((r' \in \mathbf{obs_relais_LC_off}_r) \iff \\ (r' \in \text{obs}_r) \wedge (d(\text{position}_r, \text{position}_{r'} \leq 7D)) \wedge (\text{id}_{r'} < \text{id}_r) \wedge (\text{plan}_{r'} = \text{plan}_r) \wedge (\text{LC}_{r'} = \perp)) \end{aligned}$$

Axiom 20. (Si tous ses voisins sont dans sa distance de poursuite, un drone relais se déplace vers l'un d'eux en priorisant ceux qui n'ont pas leur lumière d'avertissement de collision allumée)

$$\begin{aligned}
 & \forall n \in \mathbb{N}, \forall r = (\text{id}_r, \text{DP}_r, \text{LC}_r, \text{LR}_r, \text{position}_r, \text{plan}_r) \in S_n, \\
 & \forall r' = (\text{id}_{r'}, \text{DP}_{r'}, \text{LC}_{r'}, \text{LR}_{r'}, \text{position}_{r'}, \text{plan}_{r'}) \in S_{n+1}, \\
 & (\text{id}_r = \text{id}_{r'}) \wedge (r \in \mathbf{Relais}) \wedge (r \notin \mathbf{Source}) \wedge (\mathbf{obs_relais_LC_off}_r \neq \emptyset) \wedge \\
 & (\forall v = (\dots, \text{position}_v) \in \mathbf{obs_relais_LC_off}_r, d(\text{position}_v, \text{position}_r) \geq 6D) \implies \\
 & (\exists v = (\dots, \text{position}_v) \in \mathbf{obs_relais_LC_off}_r, \text{position}_{r'} = \text{position}_r + \frac{\text{position}_v - \text{position}_r}{d(\text{position}_v, \text{position}_r)} \times D)
 \end{aligned}$$

Axiom 21. (Si tous ses voisins sont dans sa distance de poursuite et ont leur lumière d'avertissement de collision d'allumée, un drone relais se déplace vers un d'eux)

$$\begin{aligned}
 & \forall n \in \mathbb{N}, \forall r = (\text{id}_r, \text{DP}_r, \text{LC}_r, \text{LR}_r, \text{position}_r, \text{plan}_r) \in S_n, \\
 & \forall r' = (\text{id}_{r'}, \text{DP}_{r'}, \text{LC}_{r'}, \text{LR}_{r'}, \text{position}_{r'}, \text{plan}_{r'}) \in S_{n+1}, \\
 & (\text{id}_r = \text{id}_{r'}) \wedge (r \in \mathbf{Relais}) \wedge (r \notin \mathbf{Source}) \wedge (\mathbf{obs_relais_LC_off}_r = \emptyset) \wedge \\
 & (\forall v = (\dots, \text{position}_v) \in \mathbf{obs}_r, d(\text{position}_v, \text{position}_r) \geq 6D) \implies \\
 & (\exists v = (\dots, \text{position}_v) \in \mathbf{obs}_r, (d(\text{position}_v, \text{position}_r) \leq 7D) \wedge \\
 & (\text{position}_{r'} = \text{position}_r + \frac{\text{position}_v - \text{position}_r}{d(\text{position}_v, \text{position}_r)} \times D))
 \end{aligned}$$

5 Contribution : simulateur d'essaim de robots

Durant le stage, un simulateur a été développé en parallèle. Ce simulateur⁶ a été conçu dans le but de tester efficacement et en temps réel des protocoles d'essaim de robots mobiles.[3]

Le principal avantage du simulateur réside dans sa capacité à visualiser l'évolution d'un essaim de robots exécutant un protocole. En effet, il est souvent difficile de visualiser les comportements complexes qui émergent de règles simples dans le contexte de l'algorithmique distribuée. Par exemple, le jeu de la vie de Conway, qui ne possède que deux règles simples pour chaque cellule de la grille, est "Turing-complet" [7].

Le simulateur développé a joué un rôle essentiel dans le perfectionnement des algorithmes sélectionnés pour le protocole final. Notamment, il a été utilisé pour visualiser les points de défaillance de la connexion tout au long de l'élaboration du protocole présenté dans ce document.

5.1 Choix de la technologie

Après avoir testé plusieurs solutions graphiques utilisant des langages tels que Ocaml, Python, C++ ou Java, j'ai opté pour l'utilisation d'un moteur de jeu en raison de sa capacité à développer rapidement des environnements en 3D.

Le simulateur que j'ai développé est écrit avec le moteur de jeu Godot. J'ai choisi ce moteur car il est open source (licence MIT), léger et, comparé à ses concurrents, il propose un IDE intuitif en WYSIWYG pour la création de GUI et d'entités 3D [2]. Une capture d'écran du simulateur est visible sur la Figure 10 ci-dessous :

5.2 Architecture

Afin d'assurer un système modulaire, le simulateur est séparé en trois parties comme indiqué sur la Figure 19 en annexe D :

- Le moteur d'exécution, en GDScript pure, afin de séparer la simulation du rendu
- L'implémentation des protocoles, en GDScript pure, pour les mêmes raisons
- La scène de rendu comportant le GUI et l'environnement 3D.

6. <https://github.com/smallcluster/GDST>

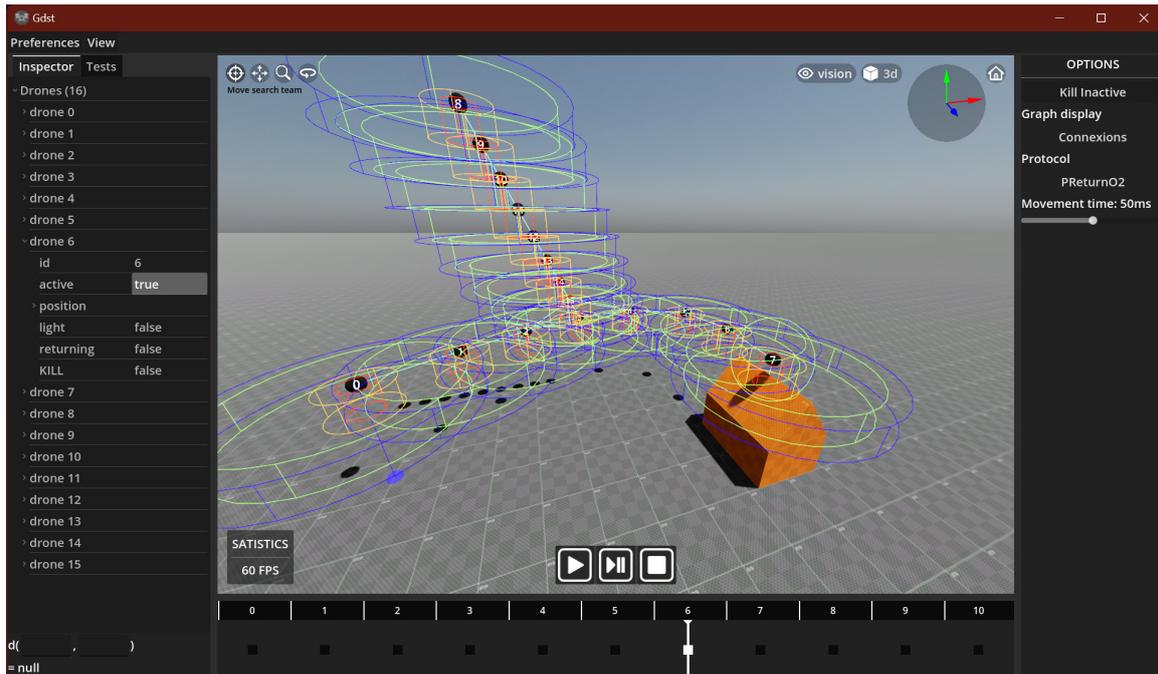
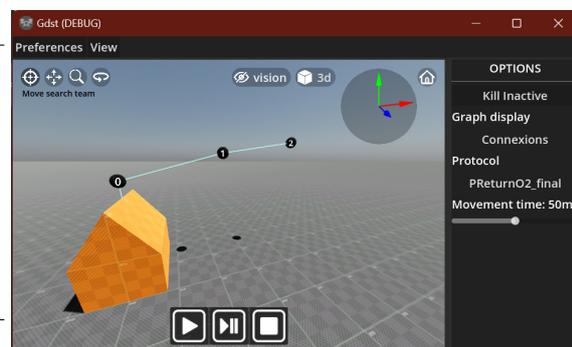


FIGURE 10 – Le simulateur en action

5.3 Fonctionnalités

- Changement de protocoles à la volée
- Test définissables en un format JSON facilement éditable à la main. Exemple en Figure 11.
- Création et chargement de tests.
- Possibilité de revenir en arrière (jusqu'au début de la simulation).
- Inspecteur d'état des drones.
- Système de conditions d'arrêt non bloquant avec message indiquant l'échec du protocole.
- Visualisation en temps réel des drones.

```
{
  "protocol" : "PReturn02_final",
  "states" : [
    {"id":0, "position":[0, 0, 0]},
    {"id":1, "position":["8*D", "2*D", 0]},
    {"id":2, "position":["14*D+D", "3*D", 0]}
  ]
}
```


 FIGURE 11 – Exemple de test avec positions paramétrées en fonction de la distance de déplacement noté D .

6 Conclusion

Mon stage de 6 mois au sein du LIM a été axé sur le développement d'un protocole de retour à la base pour un essaim de robots oblivious. Ce protocole a été conçu pour maintenir la connexion entre une base et une équipe de recherche en utilisant un essaim de drones, tout en assurant un retour sûr à la base en évitant les collisions.

Durant l'élaboration de ce protocole, j'ai rapidement compris l'importance de la visualisation pour observer le comportement des drones exécutant des protocoles simples. Le développement d'un simulateur s'est avéré être un choix judicieux pour tester les différentes idées abordées. Cependant, j'ai également compris que la preuve formelle ne peut être remplacée par des tests, car j'ai découvert des contre-exemples lors du développement du protocole qui étaient non couverts par les tests réalisés. Ce processus m'a permis d'affiner mes compétences en modélisation de systèmes distribués.

En ouverture, l'avenir du protocole développé pendant mon stage réside dans sa formulation et sa démonstration formelle à l'aide du framework Pactole qui sera réalisé plus tard au sein du LIRIS. De plus, le simulateur pourrait être adapté en Ocaml afin de formuler directement les protocoles en Coq, sans passer par un langage intermédiaire.

Références

- [1] Thibaut BALABONSKI et al. *Computer Aided Formal Design of Swarm Robotics Algorithms*. <https://arxiv.org/abs/2101.06966>. 2021.
- [2] *Godot*. <https://godotengine.org/>. Dernière consultation le 05/06/2023.
- [3] Pierre JAFFUER. *GDST : Godot Drone Simulation Thingy*. <https://github.com/smallcluster/GDST>. Dernière consultation le 05/06/2023. 2023.
- [4] LIRIS. *Pactole*. <https://pactole.liris.cnrs.fr/>. Dernière consultation le 05/06/2023.
- [5] *Présentation du LIM*. <https://lim.univ-reunion.fr/organisation/presentation/>. Dernière consultation le 05/06/2023.
- [6] *Présentation du LIRIS*. <https://liris.cnrs.fr/liris>. Dernière consultation le 05/06/2023.
- [7] Paul RENDELL. *Turing Machine Universality of the Game of Life*. Springer, 2016.

Appendices

A Organismes d'accueil

A.1 LIM



FIGURE 12 – Logo du LIM

Le Laboratoire d'Informatique et de Mathématiques (LIM) de l'Université de la Réunion (UR) est une équipe d'accueil (EA2525) structurée en trois axes :

- Epistémologie et Didactique de l'Informatique et des Mathématiques (EDIM)
- InformaTique et Applications (ITA)
- Mathématiques (MATHS)

Le LIM développe, à travers ses membres, des partenariats académiques aux niveaux local, régional (zone Océan Indien), national et international, ainsi que des partenariats industriels locaux. Il participe à la formation par la recherche, adosse le Master Informatique et Mathématiques de l'UFR Sciences et Technologies, le département Informatique et Télécommunications de l'ESIROI (Ecole Supérieure d'Ingénieurs Réunion-Océan Indien), et il co-adosse le master MEEF (Métiers de l'Enseignement, de l'Éducation et de la Formation) de l'ESPE (École Supérieure du Professorat et de l'Éducation). Il est membre de l'Ecole doctorale STS (Sciences Technologies Santé) et de la fédération de recherche OMNCG (Observatoire des milieux naturels et des changements globaux), de l'UR.

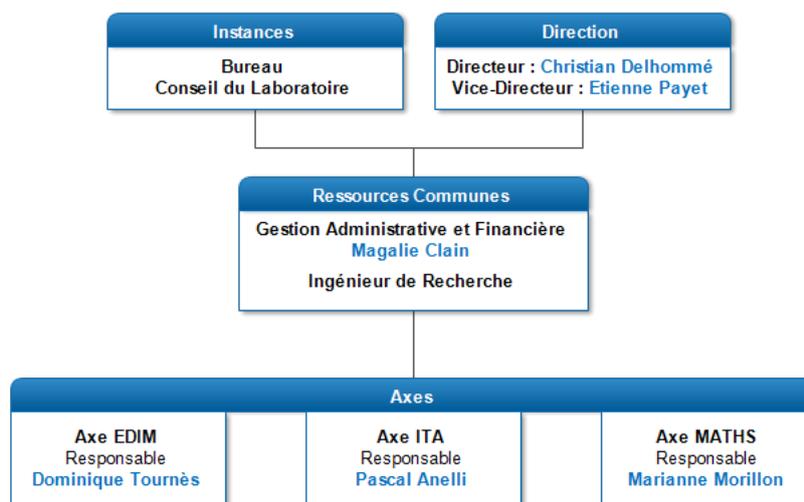


FIGURE 13 – Organigramme du LIM

Source : voir [5]

A.2 LIRIS



FIGURE 14 – Logo du LIRIS

Le Laboratoire d'InfoRmatique en Image et Systèmes d'information (LIRIS) est une unité mixte de recherche (UMR 5205) du CNRS, de l'INSA de Lyon, de l'Université Claude Bernard Lyon 1, de l'Université Lumière Lyon 2 et de l'Ecole Centrale de Lyon. Il compte 330 membres. Les recherches du LIRIS concernent un large spectre de la science informatique au sein de ses douze équipes de recherche structurées en six pôles de compétences :

- Données, Système et Sécurité (équipes BD, DRIM, SOC et DM2L)
- Informatique Graphique et Géométrie (équipe ORIGAMI)
- Images, Vision et Apprentissage (équipe IMAGINE)
- Interactions et cognition (équipes SICAL, SyCoSMA et TWEAK)
- Algorithmique et Combinatoire (équipe GOAL)
- Simulation et Sciences du Vivant (équipes SAARA et BEAGLE)

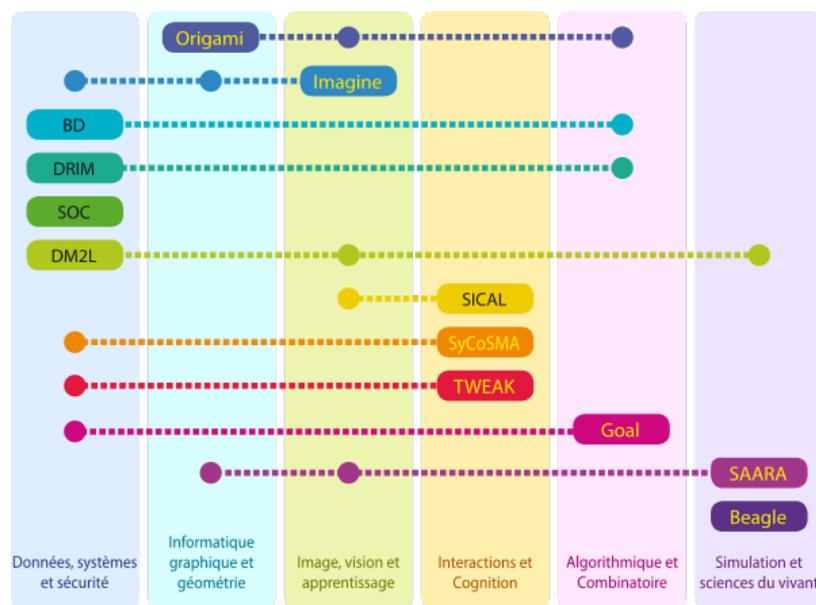


FIGURE 15 – Pôles de compétences du LIRIS

Le LIRIS est un laboratoire de recherche qui se concentre sur les défis du monde numérique tels que l'intelligence artificielle, l'analyse de données, la vision par ordinateur et la sécurité informatique. Il collabore avec différentes disciplines telles que les sciences humaines et sociales, l'ingénierie, la médecine, les sciences de la vie et l'environnement. Le

laboratoire met également l'accent sur la médiation scientifique pour rendre les avancées en informatique accessibles au grand public. De plus, le LIRIS s'engage dans des enjeux sociétaux tels que la souveraineté numérique et le développement durable en favorisant une utilisation responsable des technologies numériques et en prenant en compte l'impact environnemental de ses activités de recherche.

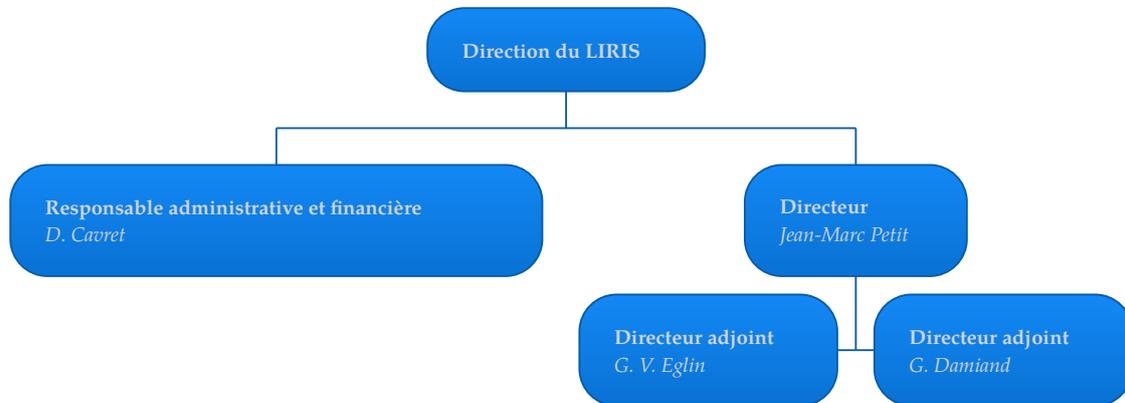


FIGURE 16 – Organigramme de la direction du LIRIS

Source : voir [6]

B Tableaux de bord de suivi mensuel projet

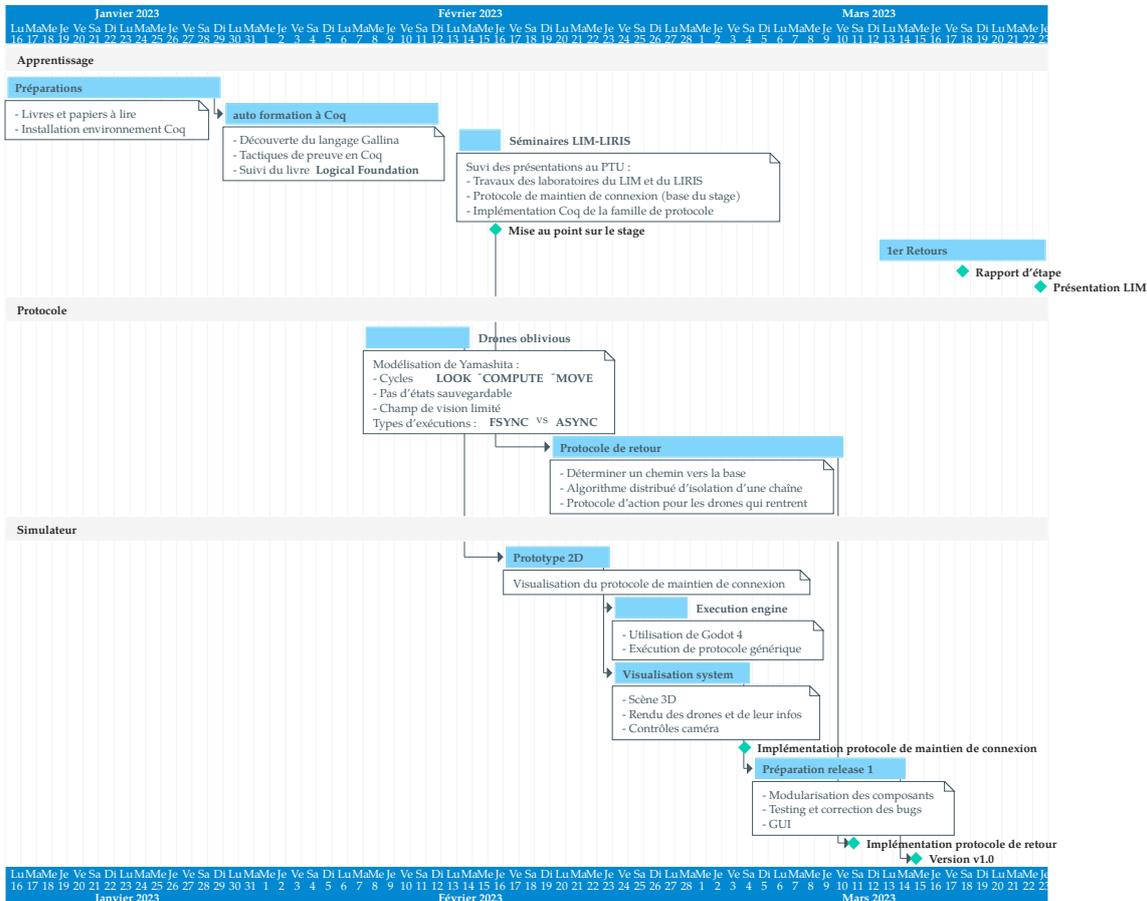


FIGURE 17 – Déroulement des trois premiers mois.

Pendant le premier mois de mon stage, j'ai consacré mon temps à préparer ce dernier. Une partie de cette préparation a consisté à me familiariser avec les modèles de robots utilisés dans le nouveau protocole, ainsi qu'à examiner quelques preuves d'infaisabilité du regroupement de robots en mode semi-synchrone. Ensuite, j'ai commencé à travailler sur le protocole de retour en étudiant un protocole de maintien de connexion déjà existant[1]. Vers la fin du mois de février, j'ai présenté mes premières idées, notamment celle de diviser le graphe de connexion en deux parties, que j'ai ensuite illustrées à l'aide d'un prototype de simulateur.

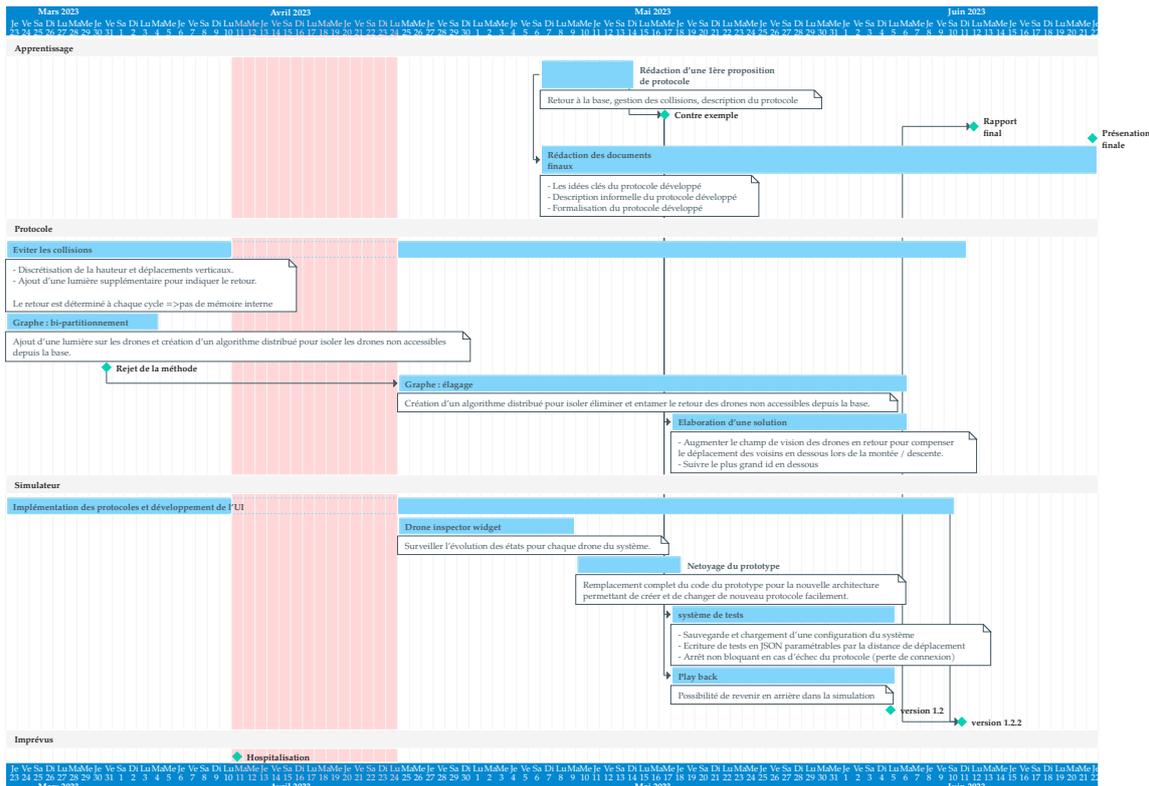


FIGURE 18 – Déroulement des trois derniers mois.

Fin mars, mon travail s'est concentré sur deux aspects : l'esquive des collisions et la détermination d'un chemin pour effectuer un retour vers la base. Malheureusement, début avril, j'ai dû être hospitalisé, ce qui a interrompu mon travail. Cependant, j'ai pu reprendre mes activités fin avril, en me concentrant sur la nouvelle méthode d'élimination des drones sous à une réunion avec mes encadrants en début avril. En milieu mai, j'ai rencontré un problème de conception dans mon protocole, mais grâce à l'aide de mes encadrants et à l'élaboration d'un système de test sur mon simulateur, j'ai pu le résoudre début juin.

Malheureusement, je n'ai pas eu suffisamment de temps pour commencer à prouver formellement la dernière solution. Néanmoins, je suis satisfait d'avoir réussi à convaincre de manière informelle la validité de mon protocole. La démonstration formelle de la validité (ou de l'invalidité) de ce protocole sera présentée dans un article publié au sein du [abv :LIRIS](#) via leur framework Pactole[4].

C Définitions et propriétés sur les graphes

Notation 1. (arrête) Soit $G = (V, E)$ un graphe où V est l'ensemble de ses sommets et E l'ensemble des liens entre deux sommets de G .

Soit $a = (s, s') \in E$, nous dirons que a est une « arrête » de G pour dire que a représente un lien non orienté entre s et s' .

Notation 2. (arc) Soit $G = (V, E)$ un graphe.

Soit $a = (s, s') \in E$, nous dirons que a est une « arrête » de G pour dire que a représente un lien orienté de s vers s' .

Définition 15. (graphe orienté ou non orienté) Nous dirons que G est « orienté » si tous ses liens entre deux sommets sont des arcs. Nous dirons que G est « non orienté » si tous ses liens entre deux sommets sont des arrêtes.

Définition 16. (Chaîne) Soit G un graphe non orienté.

Nous appellerons « chaîne » entre deux sommets s et s' de V toute suite d'arrêtes reliant s à s' .

Définition 17. (Chemin) Soit G un graphe orienté.

Nous appellerons « chemin » entre deux sommets s et s' de V toute suite d'arcs consécutifs reliant s à s' .

Par exemple, $\forall n \in \mathbb{N}, c = ((d_1, f_1), \dots, (d_n, f_n)) \in E^n$, c est un chemin (fini) de s vers s' si :

- $d_1 = s$
- $f_n = s'$
- $\forall k \in \llbracket 1, n - 1 \rrbracket, f_k = d_{k+1}$

Définition 18. (connexité) Soit G un graphe non orienté.

Nous dirons que G est « connexe » si pour toute paire de sommets s et s' de V , il existe une chaîne entre s et s' .

Définition 19. (connexité faible) Soit G un graphe orienté.

Nous dirons que G est « faiblement connexe », s'il est connexe en remplaçant tous ses arcs par des arrêtes.

Remarque : si un graphe est connexe/faiblement connexe alors il est en « un seul morceau ».

D Représentation UML de l'architecture du simulateur

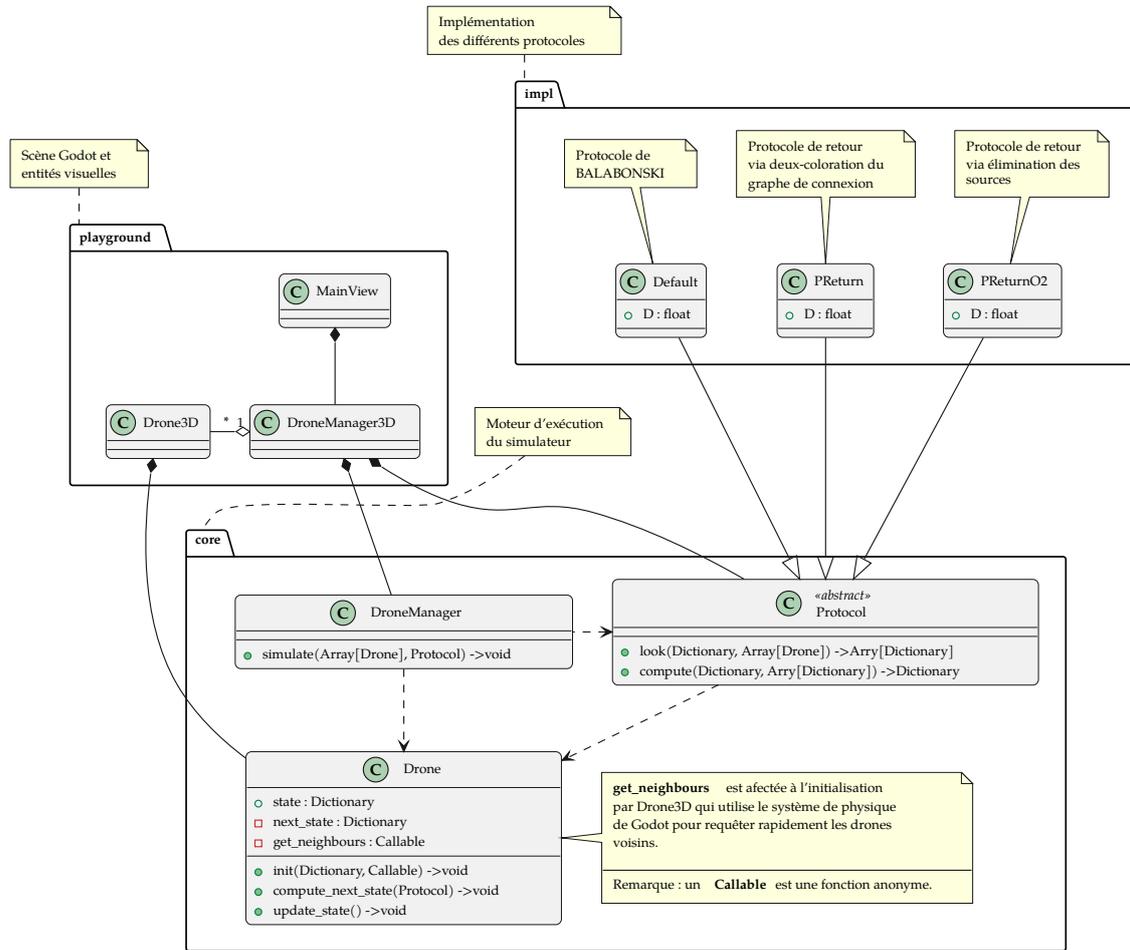


FIGURE 19 – Architecture UML du simulateur