

Exploration en essaim robotique intégrant un retour visuel pour l'opérateur

par

Corentin BOUCHER

MÉMOIRE PRÉSENTÉ À L'ÉCOLE DE TECHNOLOGIE SUPÉRIEURE
COMME EXIGENCE PARTIELLE À L'OBTENTION DE LA MAÎTRISE
AVEC MÉMOIRE EN GÉNIE AÉROSPATIAL

M. Sc. A.

MONTRÉAL, LE 23 AOÛT 2022

ÉCOLE DE TECHNOLOGIE SUPÉRIEURE
UNIVERSITÉ DU QUÉBEC



Corentin Boucher, 2022



Cette licence Creative Commons signifie qu'il est permis de diffuser, d'imprimer ou de sauvegarder sur un autre support une partie ou la totalité de cette oeuvre à condition de mentionner l'auteur, que ces utilisations soient faites à des fins non commerciales et que le contenu de l'oeuvre n'ait pas été modifié.

PRÉSENTATION DU JURY

CE MÉMOIRE A ÉTÉ ÉVALUÉ

PAR UN JURY COMPOSÉ DE:

M. David St-Onge, directeur de mémoire
Département de génie mécanique, École de technologie supérieure

M. Vincent Lévesque, président du jury
Département de génie logiciel et des TI, École de technologie supérieure

M. Maarouf Saad, membre du jury
Département de génie électrique, École de technologie supérieure

IL A FAIT L'OBJET D'UNE SOUTENANCE DEVANT JURY ET PUBLIC

LE "26 JUILLET 2022"

À L'ÉCOLE DE TECHNOLOGIE SUPÉRIEURE

REMERCIEMENTS

J'aimerais d'abord remercier toutes les personnes ayant contribué de près ou de loin à ce projet de recherche. Premièrement, je veux remercier mon directeur de recherche, professeur David St-Onge, pour son support tout au long du projet. De plus, j'aimerais remercier tous les membres du laboratoire INIT Robots qui m'ont épaulé pour différentes étapes du développement de la solution et dans le travail de tous les jours tout au long de ma maîtrise.

Ensuite, j'aimerais remercier le Fonds de recherche du Québec pour leur support financier tout au long de ma maîtrise. De plus, j'aimerais aussi remercier l'ÉTS pour leur support financier. Le support de ces deux organisations m'a permis de réaliser ma recherche et de rédiger ce mémoire.

Exploration en essaim robotique intégrant un retour visuel pour l'opérateur

Corentin BOUCHER

RÉSUMÉ

L'utilisation de la robotique gagne en popularité dans différentes sphères d'application en raison des nouvelles possibilités qu'offrent les robots. Notamment grâce à leur capacité d'enregistrer de l'information ainsi que la possibilité de travailler dans des endroits difficilement accessibles aux humains, les robots mobiles sont des compagnons idéals pour l'exploration, que ce soit l'exploration de souterrain, sous l'eau ou même d'endroits éloignés comme d'autres planètes.

En raison de leur robustesse et leur modularité, les essaims de robots sont mieux adaptés que les autres systèmes pour effectuer des missions d'exploration. L'utilisation d'un contrôle décentralisé, au cœur de l'essaim, permet à chaque robot de s'adapter de manière autonome et cohérente aux différentes situations et l'homogénéité des robots permet un remplacement rapide et autonome d'un robot fautif. D'abord inspirés des vols coordonnés d'oiseaux ou des déploiements organisés des colonies de fourmis, les essaims de robots se développent rapidement et ils seront déployés dans des scénarios réels prochainement.

Néanmoins, un aspect critique de ces déploiements d'essaims robotiques reste encore peu développé : l'interaction entre ces robots et leur(s) opérateur(s). Le contrôle d'un essaim peut induire une grande charge cognitive sur l'opérateur, notamment parce que la compréhension du comportement d'un essaim n'est pas intuitive. Dans ce travail, nous tentons d'améliorer cette compréhension en créant un retour d'information visuel pour l'opérateur par mouvements expressifs d'essaim. Lors d'une mission d'exploration, les robots doivent communiquer à l'opérateur régulièrement : lorsqu'un événement nécessitant une attention particulière survient, ou au contraire, que la mission se déroule sans problème. Notre approche est de concevoir des mouvements expressifs permettant d'améliorer la lisibilité des robots, c.-à-d. la facilité pour l'opérateur à comprendre les intentions et les actions de l'essaim.

En utilisant le langage de programmation d'essaim Buzz, nous utilisons des algorithmes d'intelligence d'essaim complètement décentralisés pour créer une solution complète. Nous utilisons premièrement un algorithme d'exploration robuste pour remplir la mission. Lorsque requis pour la communication, nous transférons certains robots non essentiels à un algorithme de formation, dans le but de créer les mouvements expressifs. Ces scripts sont finalement testés sur de petits robots de table appelés Zooids dans le but de créer une étude utilisateurs.

Mots-clés: essaim, robot mobile, expressivité, mouvement expressif

Visual feedback for the operator in swarm exploration

Corentin BOUCHER

ABSTRACT

The use of robotics is gaining in popularity in many different spheres of the industry due to the new possibilities the robots offer. Particularly because of their capacity to record information and their capacity to work in hard to get places for humans, mobile robots are the perfect companions for exploration missions, no matter where they are.

With their robustness and modularity, robot swarms are better adapted than other systems to perform in an exploration scenario. The use of decentralized control enables all robots to adapt autonomously to different situations and the homogeneity of the robots permits a replacement of a faulty robot quickly and autonomously. First inspired from flocks of birds or ant colonies, robot swarms are developing fast and will be deployed in real scenarios soon.

Nonetheless, a critical aspect of the robot swarms deployment is still underdeveloped : the interaction between the operator and the robots. Swarm control can put a high cognitive load on the operator, especially because understanding the swarm behavior is not intuitive. We thus try to improve the swarm state recognition by creating a feedback from swarm expressive motions. During an exploration mission, robots need to communicate to the operator when an event necessitating the user input happens. We thus design expressive movements for the robots to be able to communicate these messages to the operator by improving the legibility of the robots.

Using the Buzz programming language, we use swarm intelligence algorithms in a completely decentralized manner to create a fully functional solution for visual feedback in an exploration mission. We first use a robust chain mapping exploration algorithm to fulfill the mission. When required, we transfer non essential robots to an acyclic graph formation algorithm where they can create the swarm expressive motions and communicate with the operator. These scripts are tested on a swarm of small tabletop robots called Zooids with the goal of creating a user study.

Keywords: swarm, mobile robot, expressivity, expressive movements

TABLE DES MATIÈRES

| | Page |
|---|------|
| INTRODUCTION | 1 |
| CHAPITRE 1 MÉTHODOLOGIE | 5 |
| CHAPITRE 2 REVUE DE LITTÉRATURE | 11 |
| 2.1 Enjeux lors du contrôle d'un essaim | 13 |
| 2.1.1 Complexité des missions avec essaims robotiques | 13 |
| 2.2 Lisibilité des robots par le mouvement | 15 |
| CHAPITRE 3 MOUVEMENTS EXPRESSIFS | 19 |
| 3.1 Premier prototype de séquences | 20 |
| 3.1.1 Développement des séquences | 20 |
| 3.1.2 Étude utilisateurs interne | 21 |
| 3.2 Second prototype : séquences améliorées | 23 |
| 3.2.1 Détails des mouvements | 23 |
| 3.3 Étude utilisateurs | 28 |
| 3.3.1 Résultats de l'étude utilisateurs | 29 |
| CHAPITRE 4 PLATEFORME ROBOTIQUE | 33 |
| 4.1 Détails techniques | 34 |
| 4.1.1 Améliorations | 34 |
| 4.2 Localisation | 40 |
| 4.2.1 Carte sous les robots | 40 |
| 4.3 Programmation et contrôle | 44 |
| 4.3.1 Micrologiciel | 44 |
| 4.3.2 Architecture ROS | 45 |
| 4.3.3 Séquences centralisées | 46 |
| 4.4 Conclusion | 50 |
| CHAPITRE 5 PROGRAMMATION DU CONTRÔLE D'ESSAIM | 53 |
| 5.1 Programmation d'essaim | 53 |
| 5.1.1 Scénario d'exploration | 56 |
| 5.2 Algorithme d'exploration | 57 |
| 5.3 Formation par graphe acyclique | 61 |
| 5.3.1 Modifications aux algorithmes de base | 63 |
| 5.3.2 Programmation des mouvements expressifs | 66 |
| CONCLUSION ET RECOMMANDATIONS | 73 |
| 6.1 Contributions dans la littérature | 76 |
| ANNEXE I DÉTAIL DES GRAPHES | 79 |

| | |
|---------------------|----|
| BIBLIOGRAPHIE | 81 |
|---------------------|----|

LISTE DES TABLEAUX

| | Page | |
|-------------|---|----|
| Tableau 3.1 | Liste des caractéristiques du mouvement pour chacun des types de messages | 21 |
| Tableau 3.2 | Matrice de confusion de l'étude utilisateurs interne | 22 |
| Tableau 4.1 | Spécifications techniques des deux types de moteurs utilisés | 35 |
| Tableau 4.2 | Séquence de projection | 40 |

LISTE DES FIGURES

| | Page | |
|-------------|--|----|
| Figure 3.1 | Animation pour initier la communication avec l'opérateur. Première de deux animations pour ce message | 24 |
| Figure 3.2 | Animation pour initier la communication avec l'opérateur. Deuxième de deux animations pour ce message | 24 |
| Figure 3.3 | Animation pour terminer la communication avec l'opérateur | 25 |
| Figure 3.4 | Animation pour mentionner à l'opérateur qu'il n'y a aucun problème à rapporter | 25 |
| Figure 3.5 | Animation pour mentionner à l'opérateur que la mission requiert une intervention. Première de deux animations pour ce message | 26 |
| Figure 3.6 | Animation pour mentionner à l'opérateur que la mission requiert une intervention. Deuxième de deux animations pour ce message | 26 |
| Figure 3.7 | Animation pour mentionner à l'opérateur qu'il y a un problème de batterie dans l'essaim. Première de deux animations pour ce message | 27 |
| Figure 3.8 | Animation pour mentionner à l'opérateur qu'il y a un problème de batterie dans l'essaim. Deuxième de deux animations pour ce message | 27 |
| Figure 3.9 | Animation pour mentionner à l'opérateur qu'il y a un bris de communication entre les robots | 28 |
| Figure 3.10 | Comparaison pour les messages avec le plus haut rang pour chaque animation. * statistiquement significatif à $p < .05$, ** statistiquement significatif à $p < .01$, *** statistiquement significatif à $p < .001$ | 30 |
| Figure 4.1 | Les deux versions de Zoids utilisées, l'originale (petits) et améliorée (gros) | 34 |
| Figure 4.2 | Les deux versions de moteurs, ceux de la nouvelle version sont plus longs que pour l'ancienne version | 36 |
| Figure 4.3 | Les deux versions de châssis utilisées, l'originale (gauche) et améliorée (droite) | 37 |
| Figure 4.4 | Les deux versions de support utilisées, l'originale (droite) et améliorée (gauche) | 37 |

| | | |
|-------------|---|----|
| Figure 4.5 | Station de charge originale | 38 |
| Figure 4.6 | Nouvelle station de charge | 38 |
| Figure 4.7 | Support pour le PCB | 39 |
| Figure 4.8 | Les deux versions de Zoids complètement assemblées | 39 |
| Figure 4.9 | Image 24 bits contenant toutes les images de la séquence de projection | 41 |
| Figure 4.10 | Image combinée contenant une carte | 42 |
| Figure 4.11 | Projection de la carte sous les Zoids | 43 |
| Figure 4.12 | Perspective de l'opérateur par rapport aux Zoids | 45 |
| Figure 4.13 | Architecture de contrôle | 47 |
| Figure 5.1 | Formation de la chaîne pour se rendre à un but lors de l'exploration | 58 |
| Figure 5.2 | Jeu entre le travailleur et les réseauteurs Tirée de Varadharajan <i>et al.</i> (2020, p.5) | 60 |
| Figure 5.3 | La machine d'état de la formation par graphe acyclique adaptée Tirée de Li <i>et al.</i> (2019, p.8) | 63 |
| Figure 5.4 | Drones simulés dans Gazebo. Quatre ont décollé, le travailleur (cercle vert) et trois réseauteurs font le lien avec la station de contrôle (cercle jaune) | 64 |
| Figure 5.5 | Exemple de chaîne avec un travailleur (vert) et 10 réseauteurs forment deux liens avec la station de contrôle (jaune) | 65 |
| Figure 5.6 | Étapes principales pour compléter l'exploration et le retour visuel vers l'opérateur par mouvements expressifs | 66 |
| Figure 5.7 | Un sous-essaim forme la chaîne de connectivité avec le travailleur (cercle vert) au but, pendant que les autres robots forment un graphe | 67 |

LISTE DES ABRÉVIATIONS, SIGLES ET ACRONYMES

| | |
|--------|--|
| ETS | École de Technologie Supérieure |
| DARPA | Defense Advanced Research Projects Agency |
| ROS | Robot Operating System |
| BVM | Buzz Virtual Machine |
| AGF | Acyclic Graph Formation |
| ACFAS | Association canadienne-française pour l'avancement des sciences |
| ICRA | International Conference on Robotics and Automation |
| RO-MAN | International Symposium on Robot and Human Interactive Communication |

INTRODUCTION

L'exploration, un élément clé du développement des connaissances de l'humain, a toujours été un procédé très complexe à travers les époques. Que ce soit des endroits souterrains, éloignés, sous l'eau ou dans l'espace comme sur la planète Mars ou la Lune, l'humain repousse les limites de l'exploration pour en apprendre plus sur son environnement. Cependant, de plus en plus les milieux d'exploration peuvent être difficiles à explorer et voir même dangereux pour l'être humain.

C'est pourquoi l'humain utilise la technologie pour l'assister dans ces explorations. De plus en plus, la robotique crée de nouveaux outils pour l'humain lui permettant de repousser les limites de l'exploration. Un exemple est l'utilisation du robot mobile « Perseverance » sur Mars dans le but d'explorer la planète et d'améliorer la compréhension de son histoire pour les humains. Bref, les robots permettent d'effectuer des tâches que les humains ne peuvent pas réaliser ou de les aider à réaliser des tâches complexes.

Bien que l'utilisation de systèmes robotiques soit utile, elle comporte son lot de défis. Si jamais un problème survenait au robot, par exemple un bris mécanique ou un problème de batterie, la mission serait donc un échec et à recommencer. Nous voyons alors l'utilisation de groupes de robots, effectuant la même tâche qu'un robot seul, gagner en popularité en raison de la grande robustesse qu'offre une équipe de robots lors d'une exploration. La robustesse vient du fait que dans un groupe de robots, si l'un d'eux a un problème, un autre peut prendre sa place et continuer le travail sans problème. Ceci fait en sorte qu'une défaillance ne mets pas la mission en péril. Cependant, il peut être complexe de contrôler plusieurs robots à partir d'un seul ordinateur central. Que ce soit une limite de puissance de calcul ou une difficulté de communiquer avec chaque robot dans des endroits éloignés, il est extrêmement complexe d'obtenir un groupe de robots fonctionnel commandé d'un seul ordinateur. Donc, l'utilisation d'une intelligence de

groupe où chaque robot contribue à sa façon permet aux robots de travailler, peu importe la situation.

S'inspirer de la nature en ingénierie, aussi appelé le biomimétisme, est courant et permet de trouver des solutions éprouvées chez les insectes ou les animaux. C'est pourquoi les humains sont intéressés à étudier les essaims de robots en reproduisant les essaims retrouvés dans la nature comme les colonies de fourmis ou les volées d'oiseaux. Des travaux ont même déjà identifié les essaims comme étant plus efficaces que des groupes de robots centralisés (Schranz, Umlauft, Sende & Elmenreich, 2020).

Les essaims de robots ont plusieurs caractéristiques les différenciant des groupes de robots centralisés. D'abord, les essaims sont complètement décentralisés ce qui veut dire qu'il n'y a pas de maître commandant tous les mouvements de chacun des robots les permettant de réagir à leur environnement. Ceci fait en sorte que sans communication, un membre de l'essaim n'a aucun moyen de savoir ce que les autres membres de l'essaim pensent et ce qu'ils font. C'est donc avec une communication locale, c.-à-d. entre les voisins de l'essaim, que les membres s'échangent de l'information. Cette communication locale peut sembler limitée, mais en réagissant et en s'organisant avec les membres autour de lui, chaque individu de l'essaim est en mesure de s'adapter aux différentes situations sans connaître l'état de tout l'essaim. Par exemple, une colonie de fourmis utilise seulement une communication locale et il n'y a pas de dirigeant donnant des ordres aux membres de l'essaim, mais ils sont tout de même en mesure de créer des comportements de groupe phénoménaux comme des ponts ou des radeaux pour se déplacer.

De plus en plus, les essaims robotiques se développent et s'approchent des déploiements réels. Par exemple, en 2017 DARPA (Defense Advanced Research Projects Agency) a lancé un programme dans lequel les équipes doivent localiser et sécuriser plusieurs objets dans un scénario urbain à l'aide d'essaims de robots (Chung, 2021). La recherche dans le domaine s'intensifie et on peut s'attendre à voir plus d'applications dans des scénarios réels dans les années à venir.

Cependant, avant d'arriver à ces déploiements réels, il reste plusieurs défis à relever, notamment la difficulté à commander un essaim de robots lors d'une mission d'exploration. Les opérateurs sont rarement des experts et ont souvent peu d'entraînement avec les essaims de robots avant d'effectuer une mission. De plus, les opérateurs ont d'autres tâches en plus du contrôle des robots. Par exemple, un astronaute qui pourrait être en charge d'une mission d'exploration lunaire pourrait utiliser un essaim de robots pour la mission, mais il est aussi en charge d'autres astronautes et tous les autres éléments de la mission. Il faut donc être en mesure de créer un contrôle intuitif, donc ne demandant pas d'entraînement spécifique, de l'essaim pour que l'opérateur puisse être en contrôle de tous les éléments de sa mission.

Une des difficultés du contrôle de l'essaim est la compréhension du comportement de l'essaim lors de la mission. Il n'est pas toujours intuitif de comprendre l'état des robots dans la mission pendant qu'ils effectuent leurs tâches. Ceci ajoute alors à la charge cognitive de l'opérateur en rendant son travail plus complexe ce qui ensuite réduit sa compréhension de la situation et de l'environnement ainsi que sa performance pour les autres tâches de la mission. Le but est donc de réduire cet impact sur la charge cognitive de l'opérateur en améliorant sa compréhension de l'état de l'essaim lors de la mission.

La solution étudiée dans ce rapport est l'utilisation de mouvements expressifs d'essaim dans le but de créer une communication visuelle de l'essaim vers l'opérateur. Cette communication permettra d'envoyer des messages relatifs à l'état de la mission à l'opérateur ce qui lui permettra de comprendre plus facilement le déroulement de la mission du point de vue des robots.

Dans un milieu éloigné, des éléments de communications comme la lumière ou le son ne sont pas toujours disponibles, ce qui fait en sorte que la communication par le mouvement, qui est aussi adaptable à n'importe quel type de robot mobile, est la technique la mieux adaptée pour créer ces messages visuels lors d'une mission d'exploration. La question devient alors : comment générer des mouvements expressifs avec un essaim de robots lors d'une mission d'exploration ?

Ce rapport détaille les étapes entreprises pour répondre à cette question. D'abord, la méthodologie se base sur la technique du « Design Thinking ». En utilisant les cinq étapes de cette technique, nous avons pu établir clairement quels sont les défis et définir la meilleure solution, du point de vue de l'opérateur. La méthodologie ainsi que la revue de littérature présentée à la section suivante, permettent ensemble de créer un portrait détaillé des défis à relever et des outils disponibles.

Les trois autres sections de ce rapport couvrent les aspects techniques des sous-problèmes résolus dans la réalisation de notre solution. D'abord, les mouvements expressifs sont conçus à l'aide d'un groupe d'experts en art et psychologie. Ces mouvements représentent six types de messages différents permettant à l'opérateur d'obtenir un retour d'information venant de l'essaim. Ces mouvements sont développés sous forme d'animations permettant de faire une étude utilisateurs pour tester l'efficacité des mouvements.

Ensuite, la plateforme robotique est étudiée. De petits robots de table conçus à Stanford et fabriqués à l'ÉTS sont utilisés pour représenter des robots en mission d'exploration. Le fonctionnement de ces robots ainsi que les améliorations apportées sont détaillés.

Finalement, le contrôle d'essaim ainsi que les algorithmes d'exploration et de formation pour les mouvements expressifs sont expliqués. Le langage de programmation Buzz, un langage conçu pour le contrôle d'essaim, ainsi que deux algorithmes modifiés permettent d'effectuer les mouvements expressifs lors d'une mission d'exploration avec les robots.

CHAPITRE 1

MÉTHODOLOGIE

Afin de guider les étapes de la recherche et de la résolution du problème, une technique de conception a été choisie. La technique du « Design Thinking » a été utilisée pour diriger les différentes étapes du projet dans le but d'engager l'utilisateur dans le processus de développement (Sandry, Gomez & Nakamura, 2021). Le « Design Thinking » est une méthode de conception se concentrant sur l'être humain utilisateur ou dans notre cas l'opérateur de la mission. Cette technique comporte cinq étapes : compatir, définition du problème, génération d'idées, prototypage et tests (Dam, 2021).

Dans cette recherche, nous commençons d'abord par compatir avec l'utilisateur en effectuant des recherches dans la littérature et en discutant avec des experts dans le domaine. Ensuite, avec le groupe d'experts nous définissons le problème et les questions des sous-problèmes auxquels nous répondons avec le groupe d'experts. Une fois qu'une solution a été choisie, nous passons au prototypage, le corps de ce rapport, où un prototype de la solution est développé. Finalement, une partie de la solution est testée à l'aide d'une étude utilisateur et, suite à des révisions, un prototype amélioré de la solution est prêt à être testé avec une nouvelle étude utilisateurs.

Compatir

La première étape du « Design Thinking », soit compatir, consiste à obtenir une compréhension empathique du problème. En communiquant avec des experts du domaine ou directement avec les personnes concernées par le problème, le but est de comprendre le problème du point de vue de l'utilisateur pour visualiser les problèmes pouvant subvenir et essayer de trouver les solutions les mieux adaptées à l'utilisateur. Donc, en ajoutant aux connaissances pour prouver les paradigmes que nous pouvons avoir sur le problème et en communiquant de manière empathique avec les utilisateurs directement, nous pouvons mieux définir le problème.

Nous effectuons alors cette étape en premier lieu dans ce projet. Nous commençons d'abord par une recherche dans la littérature pour déterminer quels sont les problèmes auxquels les opérateurs d'essaim de robots font face lors d'une mission d'exploration (voir chapitre 2). Des travaux comme le travail sur le terrain de St-Onge *et al.* (2020a) permettent entre autres de déterminer que la charge cognitive requise pour le contrôle d'un essaim de robots limite ce que l'opérateur est en mesure de faire. En plus des tâches d'exploration, par exemple avec le reste d'une équipe, le contrôle de l'essaim peut surcharger l'opérateur. Ensuite, des discussions avec des experts permettent de déterminer que la compréhension du comportement de l'essaim joue un rôle important dans le déroulement de la mission. De plus, la communication avec l'essaim, soit le contrôle de l'opérateur vers l'essaim ou le retour d'information de l'essaim vers l'opérateur, a un impact significatif sur la charge cognitive de l'opérateur.

Définition du problème

Une fois bien informé sur le problème, il faut définir les sous-problèmes et les requis dans le but de résoudre le problème. Cette étape est cruciale pour réduire le temps qui est ensuite passé dans la génération d'idée, car une bonne définition du problème permet de guider plus précisément les pistes de solution. Ceci permettra de générer des questions pour chaque sous-problème permettant de définir quelles seront les fonctionnalités requises pour la solution.

Pour cette recherche, les informations obtenues lors de la première étape permettent de définir le problème en répondant aux besoins de l'opérateur. La première question à se poser est donc : comment réduire la charge cognitive de l'opérateur lors de l'utilisation d'un essaim robotique ? Ceci est une question générale qui dépasse le spectre de cette recherche, donc une partie de solution doit être étudiée. Suite aux discussions du groupe d'experts, la direction choisie pour réduire la charge cognitive de l'opérateur est d'améliorer la compréhension du comportement de l'essaim en améliorant le retour d'information fourni par les robots. Cependant, une mission d'exploration peut être menée dans différents environnements et avec différents robots ce qui limite les solutions comme l'utilisation du son et de la lumière. Une solution serait donc de rendre les robots expressifs pour transmettre de l'information par leur expressivité. Cependant,

les robots utilisés lors d'une mission d'exploration ne sont pas anthropomorphiques et donc leur expressivité n'est pas intuitive individuellement. Donc, la solution avec le plus de potentiel et n'ayant toujours pas été étudiée est le retour d'information par le mouvement expressif d'essaim. L'expressivité des robots peut venir de la forme qu'ils projettent ou des mouvements qu'ils effectuent, cependant plus de recherche sont requises pour identifier comment utiliser cette expressivité pour transmettre des messages. La question devient alors : comment concevoir des mouvements expressifs d'essaim permettant d'informer l'opérateur des développements de la mission ?

Ceci crée cependant d'autres questionnements qui viennent mettre des contraintes sur la solution dans le but de répondre aux besoins de l'opérateur. D'abord, comme l'opérateur a plusieurs tâches lors de sa mission, il faut être en mesure d'attirer son attention pour lui transmettre de l'information par le mouvement. De plus, cette transmission d'informations doit être rapide et donc intuitive pour l'opérateur pour ne pas nuire aux autres tâches de la mission. Donc, une fois que l'essaim obtient l'attention de l'opérateur, il doit rapidement être en mesure d'effectuer les mouvements expressifs pour transmettre l'information. Ensuite, les opérateurs ne sont souvent pas des experts en mouvements expressifs et n'ont pas d'entraînement par rapport à ce type de communication. Ces mouvements expressifs d'essaim doivent donc être intuitifs et doivent pouvoir être interprétés par un utilisateur sans expérience. Finalement, le groupe de robots est en fait un essaim de robots, donc il n'y a pas d'ordinateur central contrôlant la position de chacun des robots. La question est alors : comment est-il possible de programmer des mouvements expressifs de manière décentralisée ?

Génération d'idées

Ensuite, la génération d'idée vient répondre aux questions posées lors de la définition du problème avec plusieurs idées. Il y a plusieurs techniques pour la génération d'idée et le choix d'une solution, par exemple le remue-méninges et la maison de la qualité respectivement, qui permettent de trouver une solution optimale pour répondre aux besoins définis lors de la deuxième étape.

De notre côté, suite à la définition du problème, certaines idées sont ressorties du lot permettant de déterminer la solution finale. D'abord, les mouvements expressifs sont utilisés dans le but de communiquer certaines informations relatives à la mission à l'opérateur. Certaines contraintes sont à respecter lors de la conception de ces mouvements pour répondre à certains sous-problèmes et le tout est détaillé dans le chapitre 3. Ensuite, pour le contrôle de l'essaim et la programmation des mouvements expressifs d'essaim, le langage de programmation Buzz est utilisé. Deux scripts Buzz existants sont aussi utilisés pour effectuer cette tâche et le tout est détaillé dans le chapitre 5.

Prototypage

Ceci permettra ensuite de passer à la quatrième étape, le prototypage, qui sert à effectuer la conception de la solution et d'obtenir une solution partiellement fonctionnelle qui répond le mieux possible aux besoins de l'opérateur. Il ne faut pas oublier que le « Design Thinking » est centré sur l'opérateur et il faut toujours garder en tête les problèmes et les besoins de l'utilisateur pour que la solution soit optimale. Le prototypage doit donc refléter cette pensée.

Pour le prototypage de cette recherche, la première étape est de créer des animations des mouvements expressifs dans le but de tester l'efficacité de ces mouvements. Ensuite, ces mouvements sont implémentés sur des robots en simulation avec le simulateur Gazebo. Une fois que la simulation fonctionne, cette simulation peut être représentée avec de petits robots de table appelés Zooids qui représentent des robots en mission d'exploration et peuvent alors effectuer les mouvements expressifs. Ces robots sont détaillés au chapitre 4.

Tests

Finalement, une fois qu'un prototype a été mis sur pieds, l'équipe peut passer à l'étape des tests. Les tests permettent de vérifier si la solution fonctionne et si elle permet de répondre correctement aux besoins de l'utilisateur. Comme le processus de conception est normalement itératif, le « Design Thinking » vise à recommencer avec la première étape, donc compatir avec

l'utilisateur, lors des tests du prototype. Ceci permet de découvrir de nouveaux problèmes n'étant pas survenus lors de la recherche initiale et ceci permettra de définir de nouveaux problèmes et de recommencer le processus itérativement jusqu'à ce qu'une solution optimale soit développée.

Suite au prototypage, des tests sont alors prêts à être effectués pour améliorer la solution de façon itérative. D'abord, une première étude utilisateur est effectuée en utilisant les animations des mouvements expressifs. Les mouvements ont donc été améliorés avant d'être implémentés dans la simulation et ensuite représentés par les Zooids. La prochaine étape est donc d'effectuer une nouvelle étude utilisateur en utilisant les Zooids directement pour effectuer les mouvements expressifs d'essaim. Suite à cette étude utilisateurs, les mouvements expressifs pourront être améliorés et retestés jusqu'à ce qu'une solution optimale soit développée.

CHAPITRE 2

REVUE DE LITTÉRATURE

Les essaims robotiques sont inspirés de colonies de fourmis ou de groupes d'oiseaux. Cependant, la question que l'on peut se poser : comment est-ce que ces essaims sont capables d'effectuer des tâches complexes seulement avec des interactions locales entre les individus ? Theraulaz (2014) répond à cette question en partie en montrant que le succès des essaims vient de l'échange simple d'information entre les individus voisins. Cependant, cette information est généralement invisible pour les observateurs, de même que la connaissance des règles de cohésion du groupe, ce qui rend la compréhension de l'essaim très difficile (Dorigo, Theraulaz & Trianni, 2021). Bref, comprendre les intentions d'un essaim est très complexe, il faut donc essayer de rendre cette interaction plus intuitive pour l'opérateur.

Un autre problème qui survient lors du travail avec des essaims robotiques est la question du contrôle des robots. Il est, en quelque sorte, simple de contrôler un seul robot, mais la complexité augmente grandement lorsque l'on ajoute des robots. Les algorithmes de contrôle d'essaim, qui sont souvent mentionnés comme étant des *comportements* d'essaim, sont normalement conçus de façon à utiliser toute la puissance de l'essaim en répartissant les tâches parmi tous les robots. L'utilisation d'un contrôle décentralisé permet cela, mais complexifie la compréhension de la réalisation des tâches. Les défis de programmation de tels algorithmes ont été traités dans des travaux antérieurs (St-Onge, Varadharajan, Švogor & Beltrame, 2020b) ce qui a mené à différents déploiements réussis sur le terrain (McGuire, de Wagter, Tuyls, Kappen & de Croon, 2019). Un autre type de solution montre la possibilité d'utiliser un algorithme décentralisé sur deux robots étant capable d'arriver à un consensus et de s'aligner sans avoir un maître contrôlant les déplacements des robots (Zheng & Han, 2020).

Une fois un algorithme de contrôle défini, il faut déterminer une manière pour l'opérateur d'utiliser intuitivement ces algorithmes pour prendre en charge sa mission. Certaines de ces techniques de contrôle utilisent le mouvement de l'opérateur, comme le mouvement des mains, dans le but de contrôler l'essaim de robots (Kim, Drew, Domova & Follmer, 2020), tandis que

d'autres utilisent la danse dans le but de refléter l'expression de l'artiste par des comportements d'essaim expressifs (St-Onge, Cote Allard, Glette, Gosselin & Beltrame, 2019a). Cependant, tandis que ces travaux se concentrent sur l'envoi de commandes vers les robots, nous nous concentrerons sur le contraire, soit le retour d'information des robots vers l'opérateur.

Quelques travaux ont essayé d'améliorer la perception des comportements d'essaim par l'opérateur. Walker (2017) dans sa thèse effectue quatre études utilisateurs dans le but de déterminer l'effet des différents types de comportements d'essaim sur la perception, la prédition, le contrôle et la confiance envers l'essaim. Walker teste différents paramètres ayant un impact sur l'opérateur. D'abord, le type de comportement d'essaim influence directement la reconnaissance de la tâche de l'essaim dans un environnement comportant du bruit. Selon l'étude de Walker, le comportement de type rendez-vous, aussi appelé *agrégation*, dans lequel tous les robots se regroupent en un point, est plus facilement reconnaissable que les comportements de dispersion et de mouvements coordonnés. Cependant, un comportement de type dispersion permet une meilleure prédition de la position finale de l'essaim lors d'un déplacement selon la même étude.

Dans le but de transmettre de l'information vers l'opérateur, Lee, Jo, Kannan & Min (2021) utilise une technique de communication déictique, une communication faisant référence à la situation actuelle ou à l'environnement. En utilisant un groupe de robots et une table comportant plusieurs mots, ils réussissent à montrer que l'opérateur est en mesure de se rappeler plus facilement du mot, ou du message, qui leur a été transmis lorsque les robots l'encerclent que si le mot est seulement mentionné ou encerclé avec un crayon rouge. Ceci montre que le déplacement des robots permet à l'opérateur de mieux assimiler l'information qui lui est transmise.

De leur côté, Patel & Pincioli (2020) recherchent l'impact que de diviser les tâches vers plusieurs opérateurs peut avoir sur la performance des opérateurs lors d'un contrôle d'essaim. Ils montrent que le fait d'avoir plusieurs opérateurs effectuant différentes tâches peut avoir un effet néfaste sur la mission comme aucun opérateur ne sera au courant de tous les éléments de la mission. Donc, au lieu de diviser les tâches dans une mission il serait plus efficace d'avoir seulement un

opérateur effectuant toutes les tâches, mais de réduire la difficulté de ses tâches permettant alors de réduire sa charge cognitive.

De notre côté, nous voulons mieux comprendre comment améliorer le retour d'information vers l'opérateur lors d'une mission d'exploration. Les travaux de Lee *et al.* (2021) et de Patel & Pincioli (2020) montrent que la solution d'un opérateur obtenant de l'information par le mouvement de l'essaim est plausible tandis que nous pouvons alors nous baser sur le travail de Walker (2017) pour comprendre la perception de certains comportements par l'opérateur. Nous devons cependant comprendre quels sont les défis auxquels l'opérateur est confronté lors de sa mission avant de détailler ces solutions.

2.1 Enjeux lors du contrôle d'un essaim

Pour être en mesure de créer une solution pour un opérateur en mission d'exploration avec un essaim de robots, il faut comprendre ses défis et besoins. D'abord, un défi principal qui rend la tâche de contrôler un essaim de robots si difficile est le fait que comprendre le comportement de l'essaim n'est pas intuitif et il faut normalement un expert pour assimiler le comportement de l'essaim. Ceci met une grande charge cognitive sur l'opérateur ce qui rend ensuite le reste de son travail plus difficile. De plus, le besoin de suivre du regard plusieurs objets agissant individuellement peut avoir un grand impact sur la conscience de l'environnement pour l'opérateur (Memar & Esfahani, 2018) et conséquemment sur sa capacité de percevoir, comprendre et prédire le comportement des robots de l'essaim est diminuée.

2.1.1 Complexité des missions avec essaims robotiques

Le but est donc de minimiser l'impact sur la charge cognitive de l'opérateur lorsqu'il est responsable d'un essaim. Plusieurs travaux se sont penchés sur la question comme Hocraffer & Nam (2017) qui déclarent dans leur méta-analyse qu'augmenter l'autonomie d'un essaim dans un scénario avec une interface humain-système permet de réduire la charge cognitive mise sur l'utilisateur. De plus, d'autres techniques réduisant le nombre de tâches effectuées par l'opérateur

ou permettant de réduire la difficulté des tâches sont aussi démontrées ayant un grand impact sur la charge cognitive et donc sur la conscience de la situation chez l'opérateur. Leur travail met l'accent sur l'interface utilisateur et ils indiquent que la possibilité de modifier cette interface selon les goûts de l'opérateur et selon le type d'essaim permet de réduire la charge cognitive de l'opérateur. Ensuite, dans l'idée de réduire le nombre de tâches de l'opérateur, Podevijn, O'Grady, Carole & Dorigo (2016) démontre qu'augmenter le nombre de robots n'influence pas la charge cognitive de l'opérateur si le contrôle est effectué sur l'essaim en tant qu'objet et non sur chaque robot individuellement. Ceci démontre alors qu'utiliser des comportements d'essaim comme le mouvement coordonné et le regroupement, deux comportements parmi plusieurs pouvant être trouvés dans la littérature (Brambilla, Ferrante, Birattari & Dorigo, 2013), permettrait d'aider significativement l'opérateur dans sa mission.

Un autre moyen de réduire la charge cognitive de l'opérateur et d'augmenter sa conscience de la situation est l'ajout de multiples types de retour d'information ou la présentation d'un meilleur type d'information à l'opérateur (Menda *et al.*, 2011). Dans leur étude, Menda *et al.* (2011) montrent que, par exemple, utiliser une vue de globale, ou vue troisième personne, au lieu de la caméra avant d'un drone réduit la charge cognitive de l'opérateur. Cela amène à repenser les interfaces utilisateurs et à vérifier tous les types d'information disponibles pour l'opérateur. D'autres travaux ont aussi réussi à améliorer l'efficacité des tâches effectuées et la communication générale entre l'opérateur et l'essaim en utilisant la réalité augmentée (Walker, Hedayati, Lee & Szafir, 2018) ou des lumières clignotantes (May, Dondrup & Hanheide, 2015). Cependant, ces outils sont rarement disponibles ou fonctionnels lors d'une mission sur le terrain, ce qui fait en sorte que d'autres types de retours d'information doivent être développés pour être utilisés dans ces situations.

Bref, la littérature actuelle montre que les essaims de robots qui sont autonomes, qui sont contrôlés en tant qu'unité collective et qui utilisent plusieurs types de retours d'information sont les plus efficaces pour réduire la charge cognitive de l'opérateur lors d'une mission d'exploration.

De plus, utiliser le comportement d'essaim collectif peut être exploité de manière à créer un retour d'information pour l'opérateur avec des mouvements de groupe qui ont pour but de susciter une communication basée sur le mouvement collectif (Levillain, St-Onge, Zibetti & Beltrame, 2018). Dans l'étude utilisateurs effectuée dans le travail de Levillain *et al.* (2018), la distance entre les robots et la synchronisation du mouvement des robots, autant la synchronisation spatiale donc le mouvement dans la même direction que la synchronisation temporelle donc le déplacement au même moment sont les caractéristiques du mouvement étudiées. Les résultats montrent que la tendance des robots à rester en groupe ainsi que la synchronisation spatiale sont des caractéristiques du mouvement permettant d'augmenter l'expressivité de l'essaim. Bref, pour concevoir et implémenter ces mouvements efficacement, il faut comprendre plus en profondeur comment ces mouvements sont perçus par l'opérateur.

2.2 **Lisibilité des robots par le mouvement**

Trouver une manière de concevoir des intentions lisibles pour les robots a longtemps été le sujet de recherches. Des experts en animations cinématographiques ont recherché parmi des principes d'animation des façons d'améliorer l'expressivité des robots en créant des réactions expressives à différents résultats lors d'une certaine tâche (Takayama, Dooley & Ju, 2011). Donc, en créant de la prévoyance, de la réaction, de l'engagement, de la confiance et un bon jeu de temps, les principes d'animation permettent d'augmenter l'expressivité des robots. Dans leur travail, ces caractéristiques ajoutées aux mouvements d'un robot ont amélioré la lisibilité des mouvements du robot, c.-à-d. la facilité pour l'opérateur à comprendre les intentions et les actions de l'essaim, ainsi que la perception du comportement du robot.

Il y a déjà quelques travaux ayant fait des recherches sur la perception de mouvements collectifs par l'opérateur. Kim & Follmer (2017) mentionnent qu'il y a deux techniques principales pour utiliser un essaim pour transmettre un message. D'abord, la première technique est de positionner l'essaim dans une forme d'icône reconnaissable par l'utilisateur qui pourra en déduire un message. Des formations immobiles ont été testées utilisant des drones essayant de guider des humains en représentant des flèches ou un signe d'arrêt et les utilisateurs ont

été en mesure d'interpréter le message du groupe de robot et d'agir en conséquence (Grispino, Lyons & Nguyen, 2020). Ensuite, la deuxième technique est l'utilisation de mouvements abstraits pouvant être comparés au langage corporel humain dans le but de transmettre une information. Particulièrement, des mouvements synchronisés et des mouvements ayant tendance à créer des formes aident à transmettre de l'information à l'opérateur (Levillain, St-Onge, Beltrame & Zibetti, 2019).

Différents travaux ont montré que les émotions primaires, joie, tristesse, dégoût, peur, colère et surprise, peuvent être représentées par des mouvements de groupe (St-Onge, Levillain, Zibetti & Beltrame, 2019b) (Santos & Egerstedt, 2021). Dans le travail de St-Onge *et al.* (2019b), suite à une étude permettant l'association des émotions aux caractéristiques du mouvement, la distance entre les robots et la synchronisation spatiale et temporelle, des mouvements expressifs sont utilisés pour représenter chaque émotion. La peur est représentée par un regroupement des robots avec un déplacement rapide, la joie est représentée par une poursuite cyclique, la tristesse est représentée par des déplacements lents en groupe, la surprise par un déploiement uniforme rapide, le dégoût par une formation graphique et la colère par un déploiement aléatoire rapide. Les caractéristiques de ces mouvements peuvent donc être prises en compte lors de la conception de futurs mouvements ayant un lien avec ces émotions.

Dans le travail de Santos & Egerstedt (2021), les auteurs divisent d'abord les émotions selon leur positivité et leur intensité. La peur et le dégoût sont très similaires étant les deux des émotions négatives avec une forte intensité qui sont alors représentées par des mouvements lents créant des formes pointues. Ces deux émotions sont donc très difficiles à distinguer. Ensuite, les deux émotions positives, la surprise et la joie, sont représentées par de grandes formes rondes et des mouvements réguliers. Finalement, la colère, avec une grande intensité est représentée par des mouvements rapides et angulaires tandis que la tristesse, avec une basse intensité, est représentée par des mouvements lents créant de petites formes circulaires. Bref, en général ces caractéristiques correspondent à celles des autres travaux mentionnés.

Ces travaux cherchent à représenter des émotions alors que des caractéristiques d'expressivité plus générales sont plus utiles pour concevoir des mouvements de communication plus précis. Il faut alors trouver quels éléments permettront de lire ces mouvements.

Différentes caractéristiques du déplacement comme la vitesse, la régularité et la synchronisation des mouvements ont un impact direct sur la réponse émotionnelle des utilisateurs (Dietz, Jane, Washington, Kim & Follmer, 2017). Le travail de Dietz *et al.* (2017) mentionne que la vitesse est la caractéristique du mouvement ayant le plus grand impact sur la perception de l'intensité et de la positivité de l'essaim. Il est donc primordial de concevoir les mouvements expressifs en utilisant premièrement la vitesse pour guider la conception. Ensuite, la synchronisation des robots, dans ce cas les robots synchrones soit ceux ayant un but commun, permet de lire l'intention d'un essaim plus facilement en créant un aspect de travail d'équipe entre les robots où les utilisateurs peuvent relier ces comportements aux comportements humains. Finalement, la régularité des mouvements permet de créer un effet de positivisme dans un mouvement expressif. La régularité est créée par des mouvements sans arrêt et sans changement de vitesse fréquent ou de changements brusques de direction. Les mouvements irréguliers seront donc nerveux dans un sens créant un effet de négativité. Un mouvement expressif d'essaim doit donc inclure toutes ces caractéristiques dans le but de générer une réponse émotionnelle chez l'utilisateur permettant de transmettre un message ou un état.

De plus, des recherches se concentrant sur la capacité d'un groupe de robots à communiquer ses intentions à l'utilisateur, aussi appelé lisibilité, montrent que des variables du mouvement, la trajectoire et la dispersion du groupe, sont importantes pour la précision avec laquelle l'utilisateur peut prédire l'intention des robots, tandis que la rigidité du mouvement est importante pour la rapidité de la communication avec l'opérateur (Capelli, Secchi & Sabattini, 2019a) (Capelli, Villani, Secchi & Sabattini, 2019b). Dans leur travail, Capelli et al. définissent deux types de trajectoires comme étant une variable du mouvement. La première trajectoire contient un minimum de mouvements brusques et donc généralement fluides en direction du but, tandis que la deuxième trajectoire consiste en un arc trapézoïdal. Donc, la trajectoire droite est meilleure que la trajectoire en arc pour la prédiction de la direction de l'essaim de robots. Ensuite, ils

définissent la variable de dispersion comme la distance entre les robots calculée par une force d'attraction entre les robots. Une petite dispersion entre les robots permet aussi d'améliorer la prédiction de la direction et donc le but des robots. Finalement, la rigidité des mouvements des robots est définie par une force d'attraction entre la direction du robot et la direction vers le but. Donc, plus la rigidité est haute, plus la direction de l'essaim sera en ligne droite vers le but. Bref, plus la rigidité est haute plus le temps de réponse des participants est rapide. Cependant, le travail de Capelli et al. se concentre sur des essaims de robots complètement connectés, donc dans lesquels tous les robots savent où se trouvent tous les autres robots de l'essaim en tout temps. Néanmoins, la majorité des algorithmes de comportements d'essaim sont conçus pour s'adapter à un différent nombre de robots et à différentes topologies de communication.

La lisibilité est aussi mentionnée dans de récents travaux de Kim & Follmer (2021) : ils y démontrent qu'un mouvement basé sur un comportement de type « rendez-vous » et autres caractéristiques du mouvement, comme le contrôle, la trajectoire et la densité, ont un impact sur la lisibilité d'un groupe de robots. Les auteurs en viennent à la conclusion que pour créer un mouvement collectif d'essaim lisible il faut combiner plusieurs variables du déplacement. Ceci est confirmé par le travail de Capelli & Sabattini (2020) : leur étude utilisateurs sur certaines variables du mouvement n'a pas été concluante, car les mouvements n'étaient pas assez lisible dû à l'absence de certains paramètres.

Alors, plusieurs caractéristiques du mouvement (vitesse, synchronisation, trajectoire, etc.) ont été identifiées comme ayant un impact sur la lisibilité et sur la réussite de l'interprétation des comportements d'essaim. Ceci permet alors d'identifier les éléments importants de la conception des mouvements expressifs qui devront être directement utilisés dans la création des différentes séquences de rétroaction d'information pour l'opérateur.

CHAPITRE 3

MOUVEMENTS EXPRESSIFS

Après une revue de la littérature, nous en sommes venus à la conclusion que les mouvements expressifs d'essaim peuvent devenir un moyen efficace de transmettre de l'information à l'opérateur lors d'une mission d'exploration. La prochaine étape est donc de concevoir des mouvements pouvant être interprétés intuitivement et relayant de l'information utile à l'opérateur. Dans ce chapitre, nous nous concentrerons sur la conception de ces mouvements expressifs suivant la procédure suivante : 1- Identifier quelles informations doivent être partagées avec l'opérateur lors d'une mission d'exploration ; 2- Travailler avec un groupe d'experts pour concevoir des animations des mouvements expressifs ; 3- Valider la perception des animations avec une étude utilisateurs. Le contenu de ce chapitre a servi aux contributions d'un manuscrit soumis au journal *Autonomous Robots* (en révision).

À partir du travail de Levillain *et al.* (2019), nous évaluons qu'il y a six types de messages particulièrement importants pour l'opérateur lors d'une mission d'exploration. Dans leur travail, ils identifient différents types d'information utiles pour l'opérateur à l'aide d'une étude utilisateurs. Le premier élément à communiquer à l'opérateur est l'état actuel de l'essaim par exemple le niveau de batterie des robots de l'essaim. De plus, signaler un problème de communication entre les robots est utile pour que l'opérateur soit en mesure d'identifier si les membres de l'essaim communiquent toujours entre eux. Ensuite, l'état de la mission, par exemple le fait qu'il n'y a aucun problème, est aussi un élément important à rapporter à l'opérateur. Finalement, une requête d'intervention de l'opérateur par les robots à un certain endroit est utile pour indiquer à l'opérateur s'il doit prendre en charge un élément de la mission. De plus, nous croyons qu'il est important d'ajouter une communication phatique, un élément de communication permettant d'établir une communication entre l'orateur et l'auditeur sans toutefois transmettre un message (Jakobson, 1963), dans le but d'initier la communication. Nous ajoutons alors à cela un message pour terminer la communication permettant d'informer l'opérateur que la transmission est terminée. Bref, nous retenons alors six types de messages : initier la communication, terminer la

communication, aucun problème à rapporter, intervention requise, bris de communication entre les robots et problème de batterie dans la flotte de robots.

3.1 Premier prototype de séquences

Une première version des mouvements expressifs a été générée en animation avec comme objectif d'améliorer ces mouvements lors de discussions ultérieures avec le groupe d'experts. Ces mouvements ont premièrement été développés sous forme d'animations afin de vérifier à première vue si les séquences sont lisibles par l'opérateur. Ensuite, ces animations ont été testées avec des membres du laboratoire de recherche INIT Robots, membres de l'équipe de recherche tous habitués avec des systèmes robotiques, permettant de vérifier quelles animations sont efficaces et lesquelles sont à retravailler.

3.1.1 Développement des séquences

Des recherches précédentes sur l'utilisation des essaims comme outil de communication identifient deux techniques principales pour transmettre un message (Kim & Follmer, 2017). La première technique consiste à positionner l'essaim sous une forme reconnaissable par l'opérateur ce qui lui permet de comprendre le message. Un exemple de ce type de communication est avec un signe d'arrêt ou une flèche (Grispino *et al.*, 2020). La deuxième technique consiste à performer un mouvement abstrait pouvant être interprété par l'opérateur selon les caractéristiques du mouvement discutées dans la revue de littérature.

Avec ceci en tête, nous avons mis sur pied un groupe d'experts en art et psychologie ayant pour tâche de discuter des caractéristiques du mouvement pouvant représenter chacun des six messages importants identifiés pour une mission d'exploration. Basés sur les travaux de Santos & Egerstedt (2021) ainsi que St-Onge *et al.* (2019b), nous avons extrait les caractéristiques principales de chaque message. Ceci inclut des caractéristiques de formes comme la circularité, l'angularité et l'iconicité, tandis que les caractéristiques du mouvement peuvent être divisées sous la synchronisation (spatiale et temporelle), la vitesse et la trajectoire. Toutes ces variables sont donc

manipulées pour influencer la perception du mouvement par l'opérateur. Les caractéristiques de chaque message seront expliquées dans la section 3.2.1. Le tableau 3.1 résume les six types de messages et les caractéristiques du mouvement associé à chaque message.

Toutes les animations ont été conçues en utilisant six robots. Ce choix a été fait en prenant compte d'un essaim de 12 robots dont la moitié sont disponibles pour effectuer les mouvements expressifs. Cependant, les animations peuvent tout de même être effectuées avec des quantités différentes de robots (un minimum de 4 est requis pour comprendre la séquence).

Tableau 3.1 Liste des caractéristiques du mouvement pour chacun des types de messages

| Message | Caractéristiques | |
|---------------------------|-------------------------|---------------------|
| | Forme | Mouvement |
| Initier la communication | Uniforme | Oscillations lentes |
| Terminer la communication | Quelconque | Dispersion rapide |
| Aucun problème | Circulaire | Rotation lente |
| Intervention requise | Regroupement | Rapide/Vibrations |
| Problème de batterie | Perte d'amplitude | Lent |
| Bris de communication | Bris de formation | Cyclique |

3.1.2 Étude utilisateurs interne

Après avoir développé une première version des animations pour les six messages identifiés, nous avons créé un questionnaire pour vérifier la perception des 11 animations développées. Ce questionnaire a été rempli par neuf membres du laboratoire INIT Robots dans le but de déterminer l'efficacité de chaque animation. Après une courte mise en situation, le répondant devait choisir, pour chacune des animations, un niveau de représentation allant de « ne représente pas du tout » (ou 0 sur l'échelle numérique) jusqu'à « représente exactement » (ou 3) pour chaque type de message, donc six réponses pour chaque animation. Les résultats de ce sondage sont présentés dans le tableau 3.2 où la colonne « Séquence » représente les 11 différentes animations testées. Pour réduire les dimensions du tableau, le nom des messages est remplacé par des numéros : 1- Initier la communication, 2- Terminer la communication, 3- Aucun problème, 4- Besoin d'intervention, 5- Problème de batterie, 6- Bris de communication.

Tableau 3.2 Matrice de confusion de l'étude utilisateurs interne

| Séquence | 1 | 2 | 3 | 4 | 5 | 6 |
|----------------------------|------------|-------------|-------------|-------------|------|-------------|
| Initier la communication 1 | 0.24 | 0.27 | 0.29 | 0.12 | 0.03 | 0.05 |
| Initier la communication 2 | 0.3 | 0.14 | 0.19 | 0.14 | 0.11 | 0.14 |
| Fermer la communication | 0.18 | 0.09 | 0.24 | 0.15 | 0.03 | 0.32 |
| Aucun problème 1 | 0.21 | 0.25 | 0.32 | 0.1 | 0.06 | 0.06 |
| Aucun problème 2 | 0.23 | 0.21 | 0.29 | 0.15 | 0.06 | 0.06 |
| Aucun problème 3 | 0.25 | 0.22 | 0.37 | 0.12 | 0.02 | 0.02 |
| Besoin d'intervention 1 | 0.18 | 0.14 | 0.12 | 0.16 | 0.18 | 0.24 |
| Besoin d'intervention 2 | 0.16 | 0.25 | 0.16 | 0.22 | 0.11 | 0.09 |
| Problème de batterie | 0.19 | 0.14 | 0.05 | 0.26 | 0.12 | 0.23 |
| Bris de communication 1 | 0.2 | 0.22 | 0.17 | 0.28 | 0.02 | 0.11 |
| Bris de communication 2 | 0.08 | 0.06 | 0.05 | 0.24 | 0.24 | 0.33 |

Les réponses obtenues ont donc permis de créer un pointage pour chaque type de message pour chaque question du sondage selon le rang (entre 0 et 3) choisi par le répondant. Le pointage est donc la somme de tous les rangs, ce qui fait en sorte que le type de message ayant le plus haut rang est celui ayant eu le plus de support pour une certaine question ou mouvement. Ce pointage a ensuite été transformé sous forme de ratio permettant une compréhension simplifiée de la matrice de confusion.

Comme les participants de cette première exploration sont peu nombreux, une étude statistique approfondie ne permettrait pas d'obtenir des résultats significatifs. De plus, tous les participants ont des connaissances en robotique ce qui diminue la généralisation des résultats. Cependant, nous pouvons tout de même tirer quelques conclusions de ce sondage : les mouvements pour initier la communication et pour signaler qu'il n'y a pas de problème sont généralement efficaces, les mouvements demandant une intervention doivent être ajustés pour être plus efficaces, le mouvement signalant un problème de batterie ne fonctionne pas et doit être complètement modifié et finalement, pour le bris de communication, seulement la deuxième animation fonctionne. Bref, certains mouvements sont à retravailler et d'autres sont à recommencer complètement.

3.2 Second prototype : séquences améliorées

Suite aux résultats au questionnaire des mouvements de la première itération, le groupe d'experts s'est consulté de nouveau pour discuter des améliorations à amener aux animations. Après ces discussions, ce sont neuf mouvements expressifs d'essaim qui ont été retenus et qui ont été programmés ensuite sous forme d'animation. Ce sont ces neuf animations qui sont ensuite utilisées pour une seconde étude utilisateurs.

3.2.1 Détails des mouvements

La première animation est conçue dans le but d'initier la communication avec l'opérateur. Selon les discussions avec le groupe d'experts, nous avons établi qu'un signal ayant pour but de réaliser une communication phatique, une fonction de la communication qui a pour but d'établir la communication entre le locuteur et le récepteur, devrait attirer l'attention de l'opérateur. Une fois que l'attention de l'opérateur est obtenue, il y a plus de chances que celui-ci soit en mesure de retirer de l'information des séquences suivantes. Ceci est important particulièrement parce que ce premier élément de communication permet à l'opérateur de voir l'essaim en tant qu'agent et membre de la mission au lieu de seulement comme un outil ce qui améliore la communication entre les partis (Brinck & Balkenius, 2019). Donc, pour obtenir l'attention de l'opérateur, le groupe d'expert conclu qu'il faut créer un élément de surprise chez l'opérateur. À partir du travail de St-Onge *et al.* (2019b), nous savons alors que des déplacements rapides avec un déploiement uniforme sont souvent associés avec cet élément de surprise ce sont donc ces caractéristiques que nous mettons de l'avant lors de la conception de cette animation. Nous avons ensuite décidé de créer un mouvement cyclique qui peut être répété jusqu'à ce que l'attention de l'opérateur soit obtenue. Donc, la première séquence ayant pour but de créer un signal visuel pour l'opérateur pour initier la communication humain-essaim est une formation en cercle qui oscille, le diamètre du cercle augmente et ensuite diminue avec le temps, comme un pouls ou une respiration. La figure 3.1 illustre cette animation.

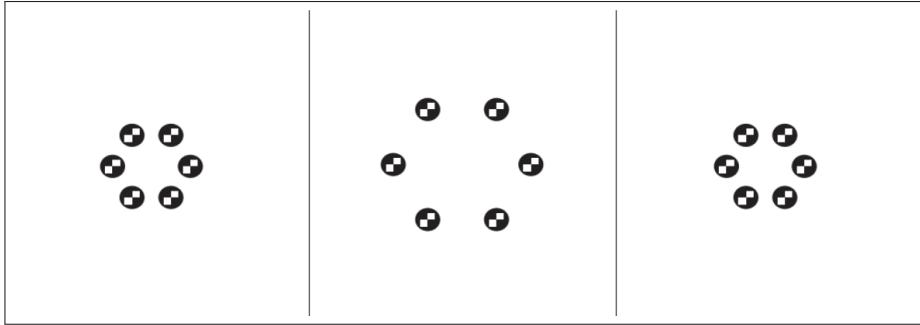


Figure 3.1 Animation pour initier la communication avec l'opérateur. Première de deux animations pour ce message

Une deuxième animation a été créée pour le même but d'initier la communication. Avoir plus d'une animation par type de message permet de tester certaines hypothèses et de voir s'il y a une différence dans la perception des animations basées sur les mêmes principes. Donc, toujours en voulant créer un élément de surprise, la deuxième animation pour initier la communication consiste en une ligne agissant aussi comme un pouls en augmentant et diminuant la longueur de la ligne avec le temps. L'animation est illustrée à la figure 3.2.

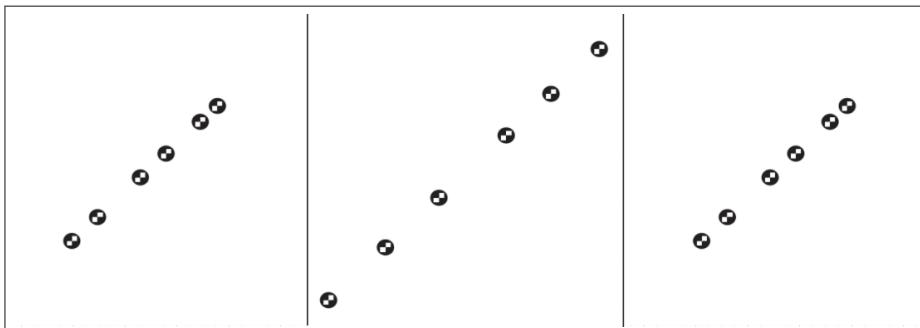


Figure 3.2 Animation pour initier la communication avec l'opérateur. Deuxième de deux animations pour ce message

Lors de la conception d'une animation pour terminer la communication avec l'opérateur, nous devons représenter que l'essaim retourne à son mode normal d'opération. Pour réaliser cet effet, il est important de briser les structures ou formes créées par d'autres séquences voulant transmettre un message à l'opérateur. L'idée est donc de disperser les robots rapidement dans différentes directions semblant aléatoires avant d'effectuer un arrêt complet. Ceci a pour but de

perdre l'attention de l'opérateur et donc de terminer la communication. Dans cette animation, nous avons choisi arbitrairement une formation de départ en ligne, mais en gardant à l'esprit que ceci pourrait être effectué après n'importe quelle formation. L'animation est illustrée par la figure 3.3.

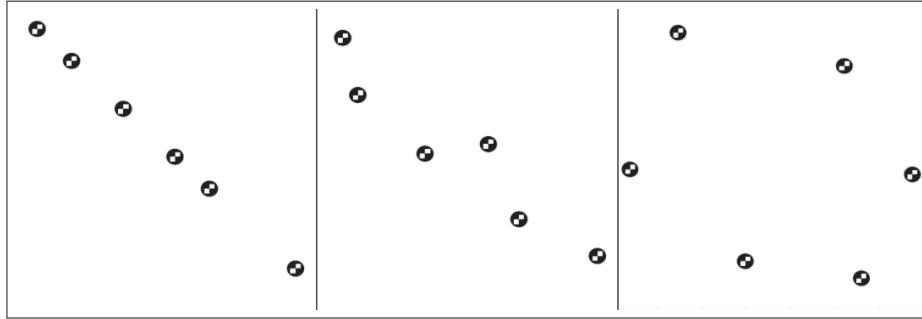


Figure 3.3 Animation pour terminer la communication avec l'opérateur

Pour mentionner à l'opérateur que la mission se déroule sans problème, il faut que les robots effectuent un mouvement apaisant pour qu'il n'y ait aucune réaction forte créée chez l'opérateur. Pour réaliser un mouvement de ce type, nous souhaitons effectuer un mouvement lent et circulaire dans une forme simple qui n'alertera pas l'opérateur : les robots forment un cercle et se déplacent lentement le long de la circonférence du cercle. L'animation est illustrée par la figure 3.4.

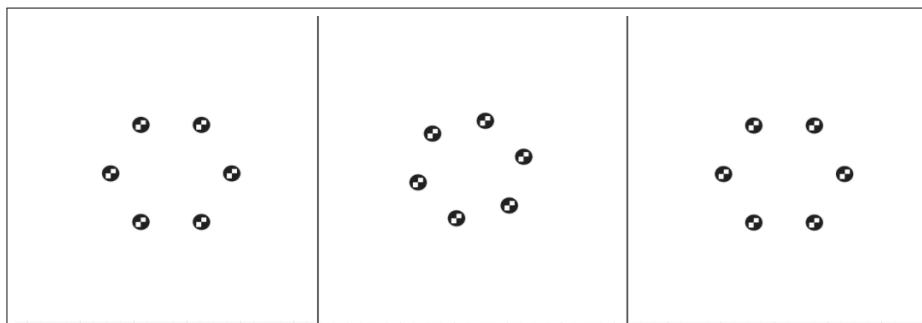


Figure 3.4 Animation pour mentionner à l'opérateur qu'il n'y a aucun problème à rapporter

Pour que l'opérateur intervienne lors d'une situation problématique lors de la mission, les robots doivent créer un élément d'alerte chez l'opérateur. Nous assumons alors que les mouvements identifiés par l'étude de St-Onge *et al.* (2019b) comme ayant un élément de peur représentent la

meilleure solution pour créer cet élément d'urgence chez l'opérateur. L'étude indique que des mouvements rapides avec des oscillations et un regroupement des robots sont les caractéristiques à respecter pour induire la peur chez l'opérateur. La première animation est créée avec une formation semblant aléatoire où certains robots vont effectuer des déplacements rapides autour d'autres robots qui oscillent sans arrêt. Les robots se déplaçant vont donc changer d'emplacement entre eux en se déplaçant entre les robots immobiles. L'animation est illustrée par la figure 3.5.

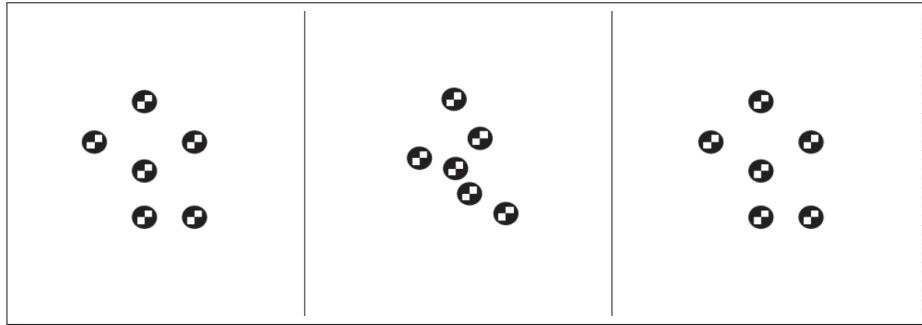


Figure 3.5 Animation pour mentionner à l'opérateur que la mission requiert une intervention. Première de deux animations pour ce message

Pour la deuxième animation ayant pour but de faire intervenir l'opérateur, les robots commencent à des positions arbitraires avant de rapidement se rencontrer en un seul point et d'osciller une fois à leur position de rencontre. La figure 3.6 illustre cette animation.

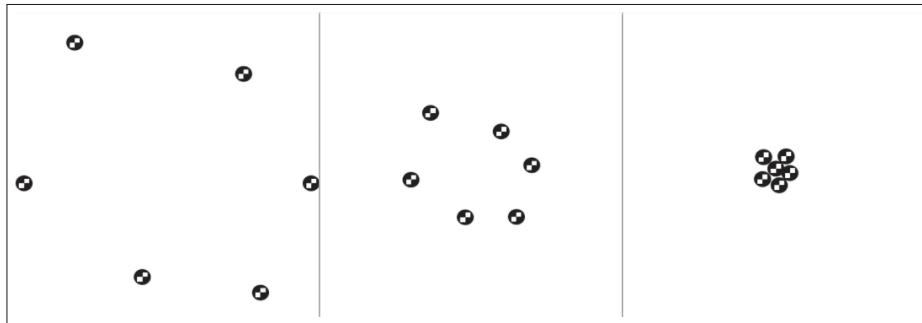


Figure 3.6 Animation pour mentionner à l'opérateur que la mission requiert une intervention. Deuxième de deux animations pour ce message

Pour représenter un problème de batterie dans la flotte de robots, nous avons mis l'accent sur la perte d'énergie. Des mouvements perdant de la vitesse graduellement ou des formes perdant

de leur amplitude ont été identifiés comme étant des signaux visuels d'une perte d'énergie. Deux animations ont donc été conçues pour ce message. D'abord, la première animation est formée d'une ligne diagonale pouvant représenter le niveau de batterie de la flotte. Peu à peu, les robots « tombent » les uns après les autres, lentement, jusqu'à obtenir une ligne droite. La ligne horizontale signifie donc que la flotte a perdu toute son énergie. La figure 3.7 illustre cette animation.

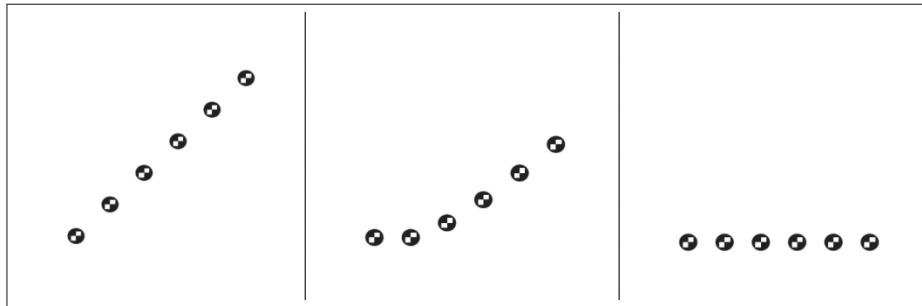


Figure 3.7 Animation pour mentionner à l'opérateur qu'il y a un problème de batterie dans l'essaim. Première de deux animations pour ce message

La deuxième animation représentant un problème de batterie est créée par la formation d'une sinusoïdale dynamique par les robots. Cette sinusoïdale perd de l'amplitude avec le temps jusqu'à créer une ligne droite signifiant la perte d'énergie des robots. Cette animation est illustrée par la figure 3.8.

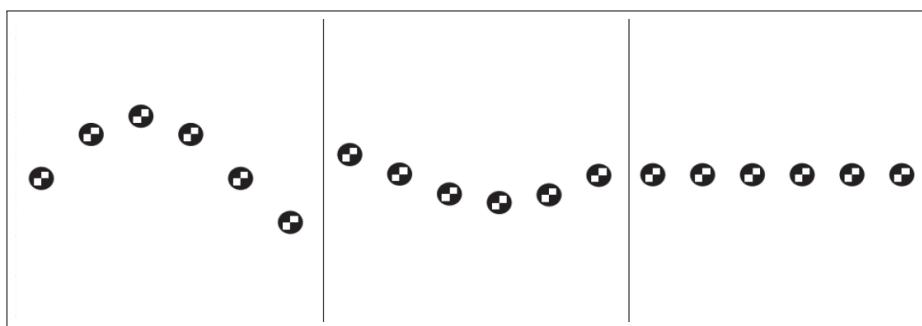


Figure 3.8 Animation pour mentionner à l'opérateur qu'il y a un problème de batterie dans l'essaim. Deuxième de deux animations pour ce message

Finalement, la dernière animation a pour but d'annoncer à l'opérateur qu'il y a eu un bris de communication entre les robots. Ce message doit alors être représenté par une perte de

synchronisation ou la perte de symétrie dans une forme. Ceci permettrait de montrer que les robots ne communiquent plus entre eux et qu'ils ne sont plus capables d'agir en groupe. Nous avons alors conçu cette animation s'inspirant du message d'aucun problème à rapporter jusqu'à ce qu'un robot brise la formation et que les mouvements perdent leur synchronisation. Donc, le mouvement commence par une ligne droite agissant comme un pouls, la longueur de la ligne augmente et diminue, jusqu'à ce qu'un robot se détache de la ligne et que les deux robots maintenant séparés des trois autres arrêtent de bouger. Les trois robots pouvant toujours communiquer ensemble continuent le mouvement tandis que les trois robots détachés restent immobiles. Le but est donc que l'opérateur en déduise qu'un des robots a brisé le lien de communication et qu'il doit réparer ce problème. La figure 3.9 illustre cette animation.

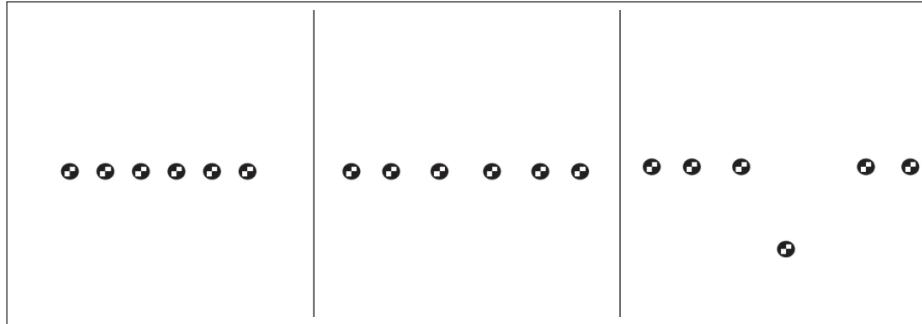


Figure 3.9 Animation pour mentionner à l'opérateur qu'il y a un bris de communication entre les robots

Bref, ces neuf animations sont le fruit d'un travail itératif avec un groupe d'expert en psychologie et en art dans le but de représenter des messages pouvant être utiles à un opérateur lors d'une mission d'exploration. Les prochaines étapes sont de vérifier l'efficacité de ces animations et de les implémenter avec un essaim de robots.

3.3 Étude utilisateurs

Suite au développement des animations, un test de leur efficacité était de mise avant d'implémenter les mouvements sur les robots et de les tester dans un scénario réel. Avec l'aide de collaborateurs, dont la professeur Elisabetta Zibetti de l'université Paris 8, nous avons donc effectué une étude utilisateurs en ligne de 15 minutes à laquelle 98 participants ont répondu. L'étude a été approuvée

par le comité d'éthique de la recherche de l'ÉTS. Les 98 participants ont été recrutés par courriel ou médias sociaux et étaient des volontaires n'ayant reçu aucune compensation pour leur participation.

Le but de l'étude est donc de déterminer quels mouvements expressifs permettent d'être interprété comme un type de message pouvant être utilisé lors d'une mission d'exploration. Même si les animations ont été conçues avec un type de message en tête, l'étude a été effectuée de manière exploratoire. Comme il y a peu de travaux précédents sur l'interprétation des mouvements expressifs d'essaims, il est possible que nos a priori soient erronés : cette étude exploratoire nous permet de déterminer si une animation peut être interprétée comme étant un message clair, peu importe lequel.

Les neuf animations étaient présentées les unes après les autres selon trois ordres différents et, pour chaque animation, les participants devaient choisir un niveau de représentation pour chacun des six types de messages, comme pour le questionnaire utilisé pour la première itération des mouvements. Les réponses étaient reçues sur une échelle de Likert en cinq points.

3.3.1 Résultats de l'étude utilisateurs

Avec l'aide de la stagiaire postdoctorale Rebecca Stower du laboratoire Init Robots, nous avons effectué une analyse statistique des résultats de l'étude utilisateurs. Un résumé des résultats peut être observé à la figure 3.10 qui illustre les réponses les plus fréquentes pour chacune des animations.

En observant les résultats, nous pouvons en conclure que seulement quelques-unes des animations ont une réponse statistiquement significative, soit les animations « initier la communication 2 », « requiert une intervention 2 » et « problème de batterie 2 » ainsi que partiellement pour l'animation « terminer la communication ». Ces animations représentent le message pour lequel elles ont été conçues sauf l'animation « problème de batterie 2 » qui est interprétée comme un message pour initier la communication. Les autres animations n'ont pas pu être clairement interprétées.

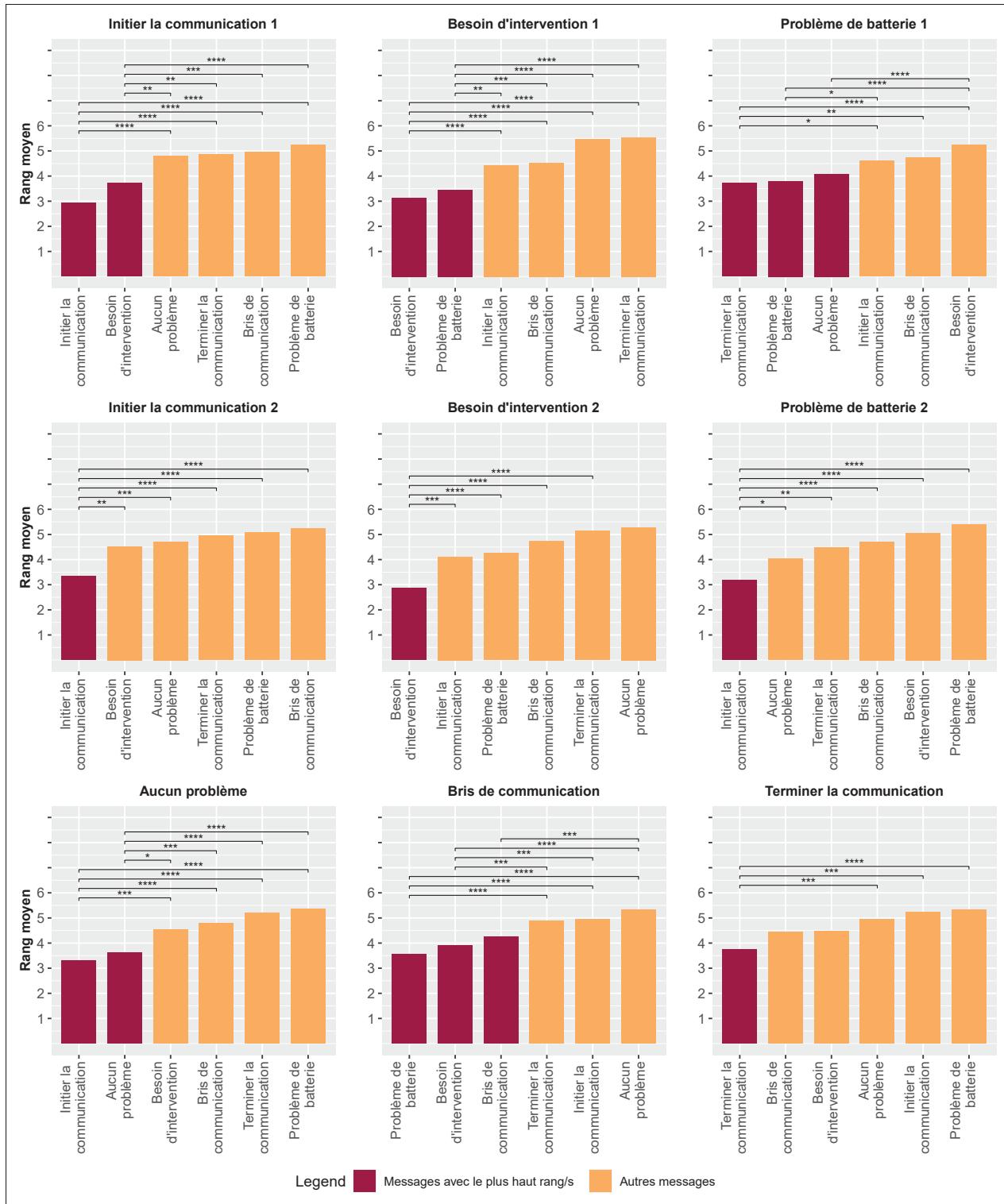


Figure 3.10 Comparaison pour les messages avec le plus haut rang pour chaque animation.

* statistiquement significatif à $p < .05$, ** statistiquement significatif à $p < .01$, *** statistiquement significatif à $p < .001$

Ensuite, une autre conclusion que nous tirons de cette étude est que le message représentant qu'il n'y a aucun problème pourrait être inutile et est souvent interprété de la même manière que le message d'initier la communication. Ceci pourrait aussi être trompeur lors d'une mission, car on peut s'attendre à ce que si les robots veulent transmettre un message, donc effectuent un mouvement pour initier la communication, l'utilisateur s'attend à obtenir une information demandant une interaction de sa part, et pas seulement un message que l'on pourrait qualifier d'inactif pour l'opérateur. Indiquer à l'opérateur qu'il n'y a aucun problème peut alors sembler contre-intuitif et pourrait créer des erreurs de compréhension du message par l'opérateur.

Suite à ces études, nous avons préparé une étude sur de vrais robots dans un scénario réel pour voir les différences de perception entre une animation et un robot réel. Les messages d'initier la communication, terminer la communication et besoin d'intervention ont chacun une ou des animations permettant d'interpréter le message clairement. De plus, le message signifiant aucun problème devrait signifier un succès permettant ainsi à l'opérateur d'obtenir de l'information de la communication. Lors de cette prochaine étude, il sera donc important de déterminer si les mouvements expressifs actuels pour le bris de communication et le problème de batterie fonctionnent dans un contexte réel ou si de nouvelles animations doivent être conçues.

CHAPITRE 4

PLATEFORME ROBOTIQUE

Pour être en mesure de tester la perception que les humains ont par rapport à un mouvement expressif d'essaim, il faut avoir une plateforme robotique permettant ces différents tests. Plusieurs facteurs entrent en jeu lors du choix de la plateforme robotique : la facilité d'implémentation des mouvements, la facilité d'accès à la plateforme robotique et la facilité de la conduite des études utilisateur. L'idéal est donc de trouver des robots représentant un essaim et donc avec lesquels il est simple de déployer des dizaines de robots en même temps. Ensuite, il faut que ces robots puissent être déployés avec un minimum de contraintes nous permettant de conduire des études dans différents environnements.

Notre choix s'est posé sur les Zooids, inspiré du travail de Le Goc *et al.* (2016) : de petits robots de table à entraînement différentiel. Ces petits robots de 2,6 cm de diamètre permettent de représenter d'autres robots lors d'une mission d'exploration et peuvent donc communiquer avec un opérateur qui superviserait une mission à distance à l'aide de leur mouvement. Par exemple, lors d'une mission d'exploration, des drones peuvent être utilisés pour explorer un certain endroit et les Zooids peuvent représenter l'emplacement de chacun des drones dans une situation où l'opérateur reste à la station de contrôle. Les Zooids agissent donc en tant que proxy permettant à l'opérateur de contrôler les drones avec une interface tangible en déplaçant des Zooids et lui permettent d'obtenir de l'information relative à la mission selon les mouvements expressifs des Zooids.

Les Zooids sont d'abord des robots comportant chacun un microcontrôleur de type STM32 avec leur propre algorithme de fonctionnement (micrologiciel). Après avoir obtenu leur position à partir des photodiodes et de la séquence du projecteur, la position est envoyée à un noeud ROS (Robot Operating System) à travers une communication à fréquences radio (2,4 GHz) à l'aide d'une puce NRF24L01+ qui est captée par une carte Arduino, comportant une antenne, qui transmet les messages reçus à l'ordinateur par communication série USB. Toute l'information relative aux Zooids peut être trouvée sur le site GitHub de ShapeLab (ShapeLab, 2019). Cette

section couvrira l'assemblage des Zooids, leur mode de localisation ainsi que les techniques de programmation et de contrôle utilisées pour permettre l'étude utilisateurs.

4.1 Détails techniques

Les Zooids sont donc utilisés pour représenter un essaim de robots dans une mission d'exploration et doivent alors représenter le plus précisément possible les animations des mouvements expressifs. Pour réaliser ces étapes et être en mesure d'effectuer une étude utilisateurs avec les Zooids, certaines améliorations ont dû être apportées à la conception originale. La figure 4.1 montre la flotte de Zooids utilisée comportant des exemplaires de la version originale et des exemplaires de la version améliorée. Les améliorations apportées aux Zooids sont détaillées dans cette section.



Figure 4.1 Les deux versions de Zooids utilisées, l'originale (petits) et améliorée (gros)

4.1.1 Améliorations

Après plusieurs tests, le comportement des robots lors des mouvements expressifs ne répondait pas aussi bien que nous l'avions espéré aux caractéristiques requises pour obtenir une bonne

perception des mouvements. Les robots étaient trop rapides et les mouvements manquaient de précision, ce qui cause entre autres beaucoup de vibrations, souvent perçu comme des *hésitations*. Nous avons alors procédé à deux types de modifications dans le but d'améliorer le déplacement des robots : la modification de la logique de déplacement (discuté dans la section 4.3.1) ainsi qu'un changement des moteurs.

Nous avons donc opté pour des moteurs avec un ratio d'engrenage plus élevé et donc une vitesse plus basse. Les deux versions de moteurs sont détaillées dans le tableau 4.1 (Firgelli Automations, 2022). Ces moteurs nous permettent aussi de modifier la vitesse plus facilement permettant donc une plage de vitesse plus grande lors de la conception des mouvements expressifs. Cependant, ces moteurs sont 2,5 mm plus longs que les moteurs initiaux, comme illustré sur la figure 4.2, faisant en sorte que le châssis et d'autres pièces mécaniques des Zooids devaient être modifiés avant leur fabrication.

Tableau 4.1 Spécifications techniques des deux types de moteurs utilisés

| Mesures (mm) | Ratio d'engrenage | Vitesse sans charge | Courant sans charge |
|--------------|-------------------|---------------------|---------------------|
| 6 x 16,3 | 1 :26 | 1250 tr/min | 40 mA |
| 6 x 18,8 | 1 :136 | 240 tr/min | 60 mA |

D'abord, le châssis a été modifié pour accueillir les nouveaux moteurs en augmentant le diamètre de 2,5 mm. De plus, la hauteur du châssis, illustré à la figure 4.3, a été augmentée permettant d'avoir plus d'espace pour les fils à l'intérieur du robot, un problème qui avait été identifié sur les Zooids originaux. Ensuite, le support à moteurs a aussi été modifié pour deux raisons, la première étant qu'augmenter la longueur du support faisait en sorte que les moteurs soient plus stables dans le châssis. La deuxième raison vise à résoudre un problème rencontré à plusieurs reprises lors de l'assemblage des Zooids originaux : l'interrupteur est difficile à fixer dans le fond du châssis et de la colle est nécessaire pour le fixer, mais risque d'altérer son bon fonctionnement (pièces mobiles saisies dans la colle). L'ajout d'une extrusion se positionnant entre les deux fils de l'interrupteur, illustré à la figure 4.4, vient régler ce problème.

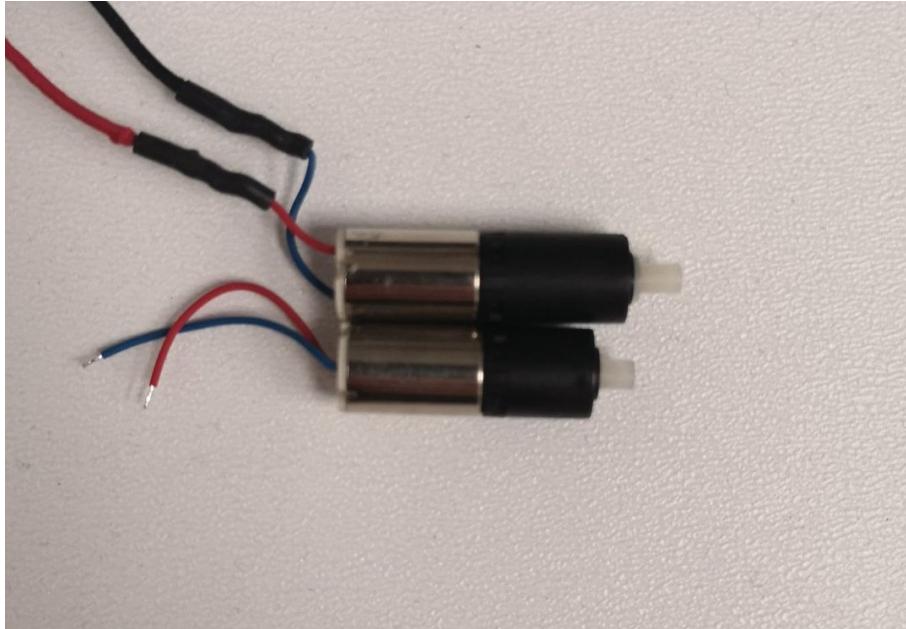


Figure 4.2 Les deux versions de moteurs, ceux de la nouvelle version sont plus longs que pour l'ancienne version

Le changement de dimensions des Zooids a aussi fait en sorte que la station de recharge utilisée, illustrée à la figure 4.5, n'acceptait pas les nouveaux modèles. Une nouvelle station de charge, à la figure 4.6, a donc été conçue et fabriquée pour la recharge du nouveau modèle de Zooids. Cette nouvelle station de charge utilise exactement le même circuit que l'ancienne station, mais sur un circuit imprimé au lieu de plaquettes de prototypage rapide ce qui rend le tout beaucoup plus propre et potentiellement plus durable. Il sera également plus facile d'en produire à nouveau. Le support pour les robots sur le dessus de la station a été imprimé en ABS à l'ÉTS.

Avec le diamètre des châssis augmenté, les PCBs ne tiennent donc plus en place s'ils sont positionnés directement sur le châssis comme dans la conception d'origine. Un support, illustré à la figure 4.7, a donc été ajouté permettant de bien supporter le PCB à l'intérieur du châssis.

Le châssis des Zooids, le support des moteurs ainsi que le support du PCB ont été fabriqués par impression 3D à l'aide d'une imprimante à dépôt de matière avec le matériau Onyx de Markforged. Avec l'achat des PCBs, moteurs, batteries, interrupteurs et boulons, les Zooids peuvent complètement être assemblés à l'aide d'un fer à souder. Ceci peut sembler simple, mais



Figure 4.3 Les deux versions de châssis utilisées, l'originale (gauche) et améliorée (droite)



Figure 4.4 Les deux versions de support utilisées, l'originale (droite) et améliorée (gauche)

vu les petites dimensions des robots, l'assemblage est un travail minutieux qui requiert temps et expertise. Les deux versions de Zooids complètement assemblées sont présentées à la figure 4.8.



Figure 4.5 Station de charge originale

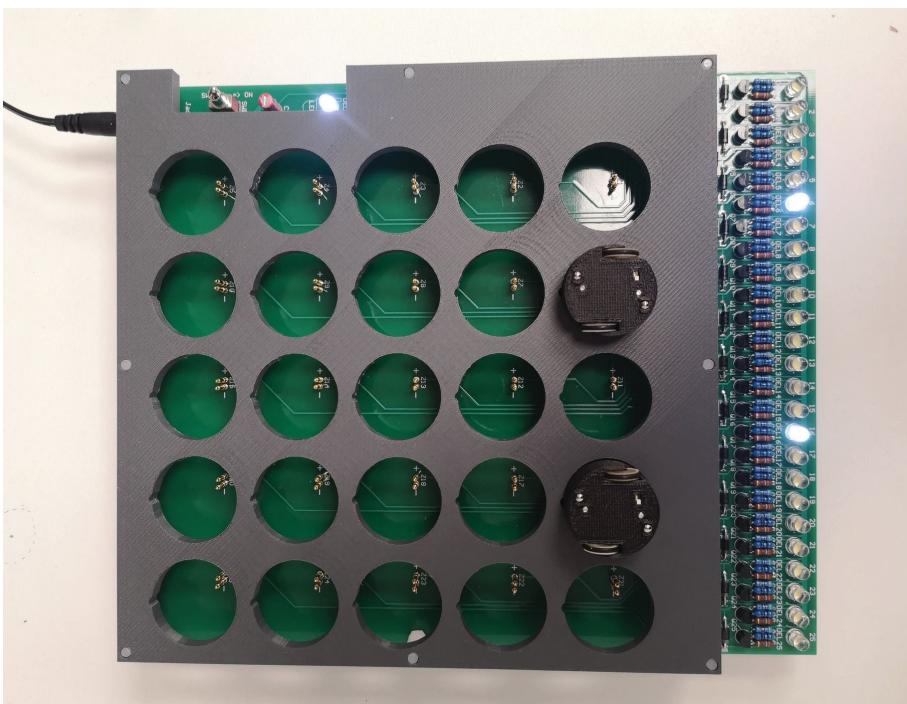


Figure 4.6 Nouvelle station de charge



Figure 4.7 Support pour le PCB

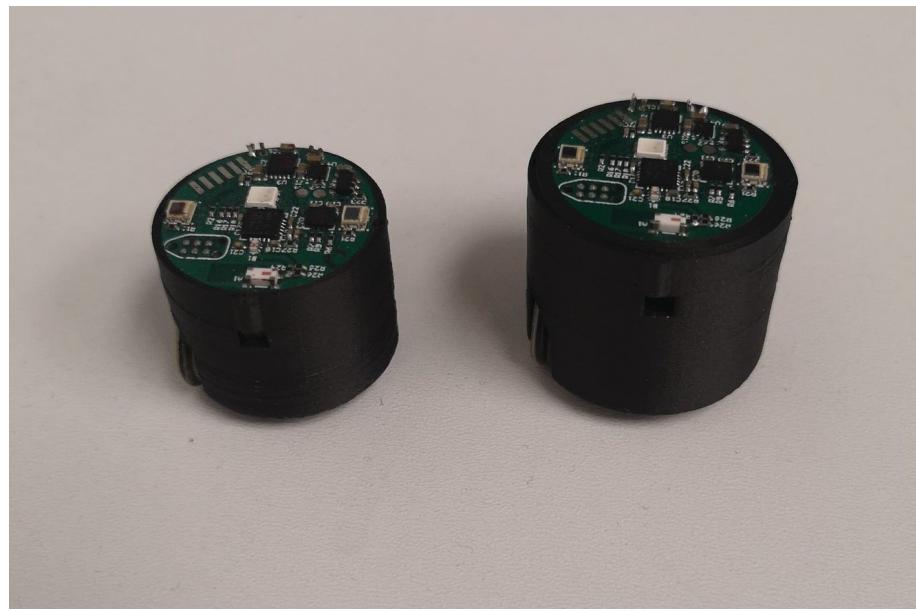


Figure 4.8 Les deux versions de Zooids complètement assemblées

4.2 Localisation

Pour pouvoir se déplacer suivant les trajectoires désirées, les robots doivent connaître leur position dans un référentiel commun. Dans le cas des Zooids, on utilise une arène où les robots peuvent se déplacer et où l'origine du référentiel commun est dans un des coins de l'arène. Un projecteur (DLP LCR4500 EVM) en surplomb de l'arène projette un flux d'images comportant des barres noires et blanches à une vitesse de 3000 Hz. Les robots sont munis de deux photodiodes permettant de détecter ces barres (blanc ou noir) et déterminent leur position grâce à celles-ci. La séquence du flux d'image est présentée dans le tableau 4.2.

À l'aide de cette séquence d'images, les Zooids se positionnent dans une arène étant divisée par 885 divisions en X et 530 en Y. Selon la résolution des images et la puissance du projecteur, nous avons déterminé que les dimensions optimales de l'arène sont de 108 cm en X et 66,5 cm en Y. Ceci fait en sorte que le plus petit déplacement perceptible par les Zooids est de 1,22 mm en X et 1,25 mm en Y. Donc, la résolution du déplacement est plus de 20 fois plus petite que la grosseur des Zooids ce qui permet une bonne précision lors du positionnement des robots.

Tableau 4.2 Séquence de projection

| bit | 0 | 1-10 | 11-20 | 21-40 |
|---------------------|---------------|------------------------|--------------------------|--------------|
| Fonction Couleur | Début Noir | x N et B (vertical) | y N et B (horizontal) | Fin Blanc |

4.2.1 Carte sous les robots

Dans le but de créer un scénario d'exploration pour l'étude utilisateurs, il est utile de pouvoir afficher une carte sous les Zooids lors de leurs déplacements. Pour comprendre comment ceci peut être réalisé, il faut comprendre le fonctionnement du projecteur plus en détail.

D'abord, nous utilisons le mode « Pattern Sequence » du projecteur qui permet de faire défiler différentes images à grande vitesse. Dans notre cas, nous avons 22 images différentes (voir tableau 4.2) à faire défiler à 3000 Hz. Pour permettre au projecteur de lire ces images à une telle

vitesse, nous devons combiner toutes ces images en une seule image de 24 bits où chaque image, des images binaires en noir et blanc, correspond à un bit de l'image combinée. Le projecteur peut donc charger une seule image et projeter seulement le bit voulu pour chaque pixel de l'image. L'image combinée est présentée à la figure 4.9. Ensuite, la séquence est créée en choisissant les bits (G_0 à G_7 , R_0 à R_7 et B_0 à B_7) voulus dans l'ordre voulu, c'est-à-dire les 21 premiers bits une fois (de 0 à 20) et le bit 22 est projeté 20 fois.

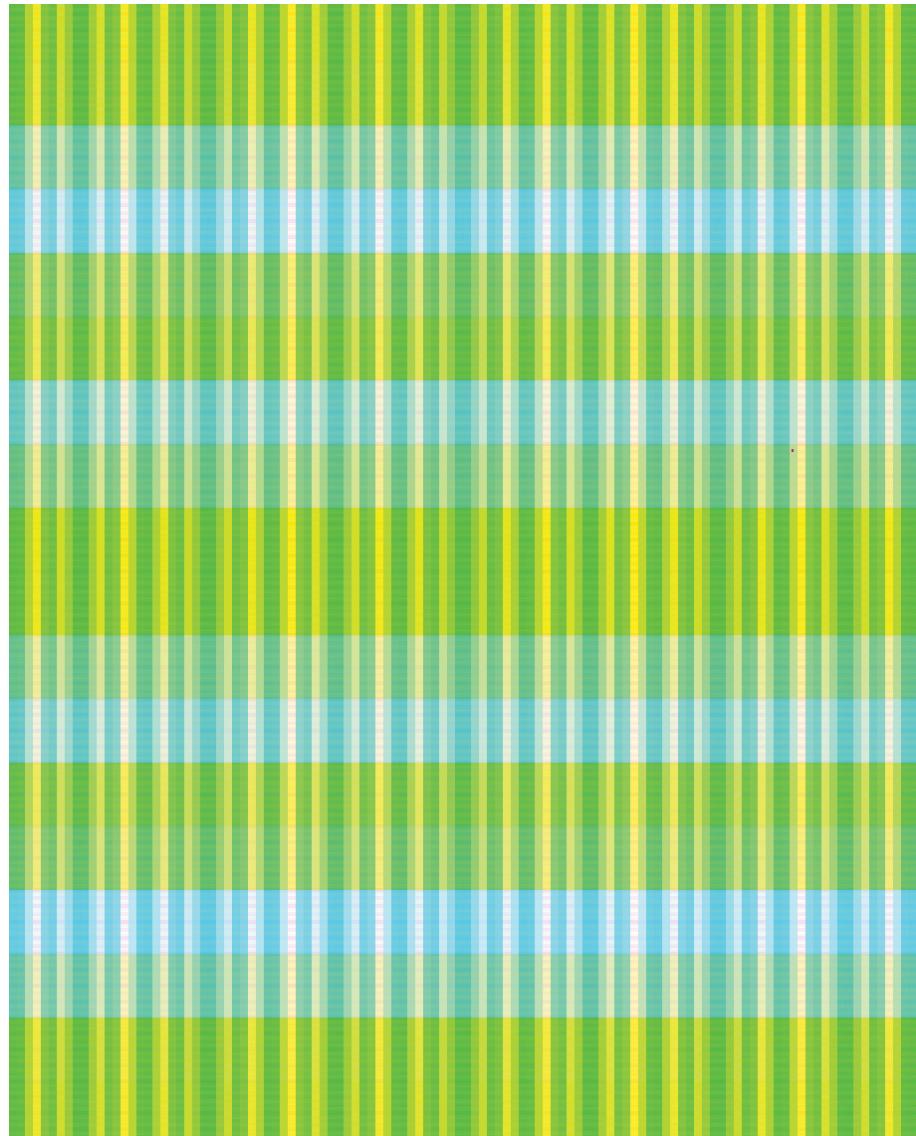


Figure 4.9 Image 24 bits contenant toutes les images de la séquence de projection

Alors, pour projeter une image sous les Zooids, nous devons tenir compte que les Zooids ont besoin de voir 20 images blanches à la fin de la séquence ou presque 50% du temps. Cependant, tout ce qui est autour des Zooids peut être en noir permettant alors de montrer une carte ou le noir serait des obstacles et les robots restent aux endroits blancs. Comme la carte est visible 50% du temps, elle est bien visible à l'oeil humain et il n'y a pas d'impact sur le positionnement des Zooids. La figure 4.10 représente l'image combinée avec la carte et la figure 4.11 montre la carte étant projetée sous les Zooids.

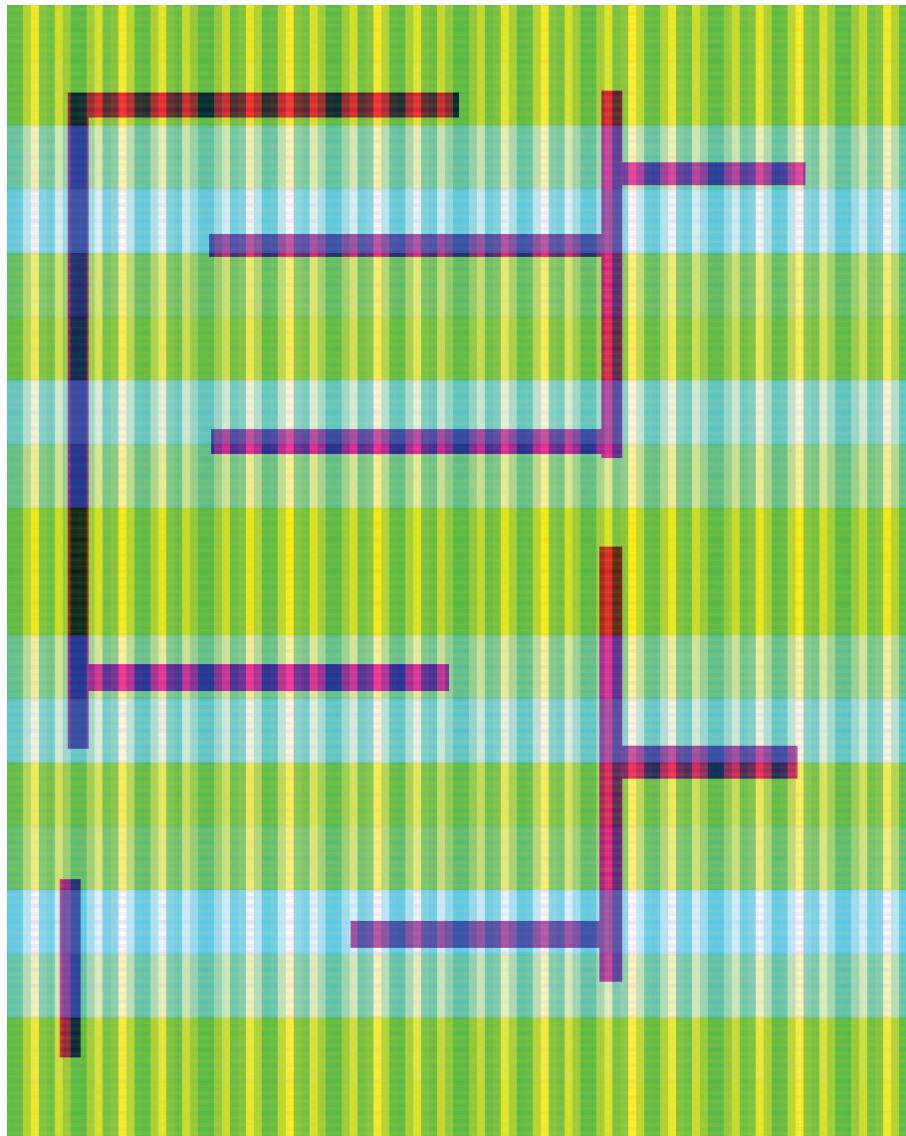


Figure 4.10 Image combinée contenant une carte

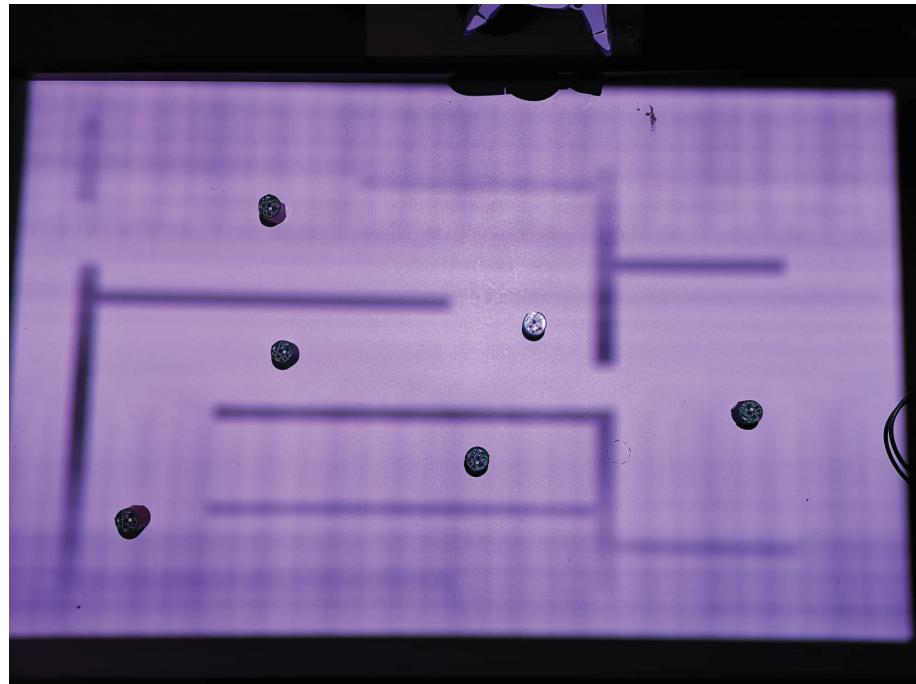


Figure 4.11 Projection de la carte sous les Zooids

Lors d'une mission d'exploration, les robots ne connaissent pas la carte de l'environnement au début de la mission, la carte sera donc mise à jour au fur et à mesure qu'elle est découverte. Pour recréer ce phénomène, nous avons créé un noeud ROS, le « Robot Operating System » créant l'architecture de contrôle des robots (voir section 4.3.2), permettant de créer des images 24 bits, requises par le projecteur, avec une différente carte à chaque itération. Ceci permet à un noeud de contrôle de modifier la carte et de l'envoyer sur un sujet ou « Topic » ROS. Le noeud de création d'images peut ensuite combiner les images de la séquence des Zooids avec la nouvelle carte pour ensuite l'envoyer au projecteur pour être projeté sous les Zooids. Le DLP LCR4500 EVM accepte des images par l'entremise d'un connecteur mini HDMI et agit ainsi comme un deuxième écran à l'ordinateur de contrôle. Avec ce type d'entrée, le projecteur peut seulement projeter à une vitesse de 2880 Hz (Texas Instruments, 2017), mais après plusieurs tests nous pouvons conclure que ceci n'influence pas significativement le positionnement des Zooids. Ceci nous permet alors d'avoir une carte interactive sous les Zooids selon leurs déplacements dans l'espace.

4.3 Programmation et contrôle

Il y a deux techniques permettant de tester les mouvements expressifs conçus avec les animations sur les Zooids. La première, le but de ce travail, est d'utiliser un contrôle complètement décentralisé et de programmer tous les mouvements de manière décentralisée. Cependant, ceci est un long processus, décrit à la section 5, et ne permet pas de tester rapidement les mouvements sur les Zooids. Comme il est important de voir comment les Zooids se comportent lors des mouvements expressifs pour déterminer s'ils ont besoin d'améliorations, nous devons d'abord utiliser un contrôle centralisé pour obtenir des résultats plus rapidement.

Il est aussi important de prendre en compte la perspective de l'opérateur lorsqu'il regarde les Zooids. Si son orientation est différente, sa perception des mouvements sera aussi différente. Nous tenons alors en compte que l'opérateur sera devant l'arène des Zooids et qu'il aura généralement une vue de haut des robots, ressemblant alors à la perspective des utilisateurs regardant les animations. La figure 4.12 illustre la position de l'opérateur par rapport aux Zooids.

Pour être en mesure d'effectuer les mouvements expressifs avec les Zooids avec la meilleure ressemblance aux animations développées, des améliorations logicielles sont apportées autant dans l'architecture de contrôle que dans l'algorithme de déplacement. Dans cette section, nous détaillons d'abord le micrologiciel des Zooids, ensuite l'architecture ROS utilisée et finalement le contrôle centralisé pour effectuer les mouvements expressifs.

4.3.1 Micrologiciel

Le micrologiciel (*firmware*) de base des Zooids ne convenait pas pour reproduire les mouvements expressifs. De base, les robots se déplacent seulement dans un sens et font certains mouvements courbes. Pour obtenir des mouvements plus vifs, il était important d'être en mesure d'avoir des mouvements seulement en ligne droite.

Une modification a donc été apportée pour que le robot s'aligne toujours avec sa cible avant de se déplacer dans le but de seulement se déplacer en ligne droite. De plus, un ajout dans le

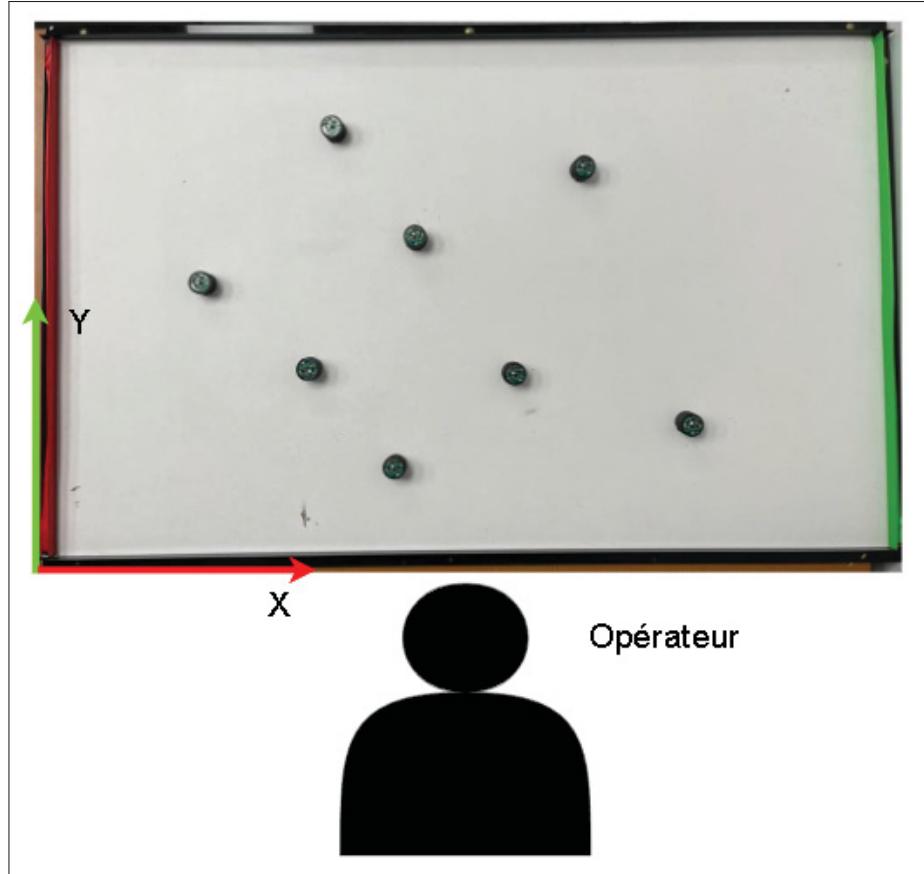


Figure 4.12 Perspective de l'opérateur par rapport aux Zooids

contrôle de l'angle a permis de déterminer le meilleur sens, devant ou derrière, pour le robot de se déplacer. Ainsi, le robot peut s'aligner avec sa position finale automatiquement d'un sens ou de l'autre, le plus rapide pour lui, contrairement à toujours se déplacer du même sens. Les détails de l'algorithme sont illustrés avec le pseudo-code 1.

4.3.2 Architecture ROS

Nous utilisons ROS (Robot operating System) pour l'architecture de contrôle des Zooids. La base de la communication avec les Zooids passe par le noeud que nous appelons «Zooids Bridge» qui fait le pont entre l'antenne, connectée à l'ordinateur par USB, et ROS.

Algorithme 4.1 Algorithme de déplacement d'un Zooid

```

1 Algorithm : Algorithme de déplacement
2 angleDiff = angleBut – angleActuel
3 if angleDiff > 0 then
4   | Rotation antihoraire
5 else
6   | Rotation horaire
7 end if
8 angleDiff = angleBut – angleActuel
9 sens = cos(angleDiff)
10 if sens > 0 then
11   | vitesse = 1
12 else
13   | vitesse = -1
14 end if
```

Le « bridge » est la base de la communication et du contrôle avec les Zooids. Les Zooids utilisent un type de message spécifique permettant de communiquer différentes informations, dont leur position, l'orientation et le niveau de la batterie. Donc, le « bridge » permet de recevoir ces messages à travers la communication USB de l'antenne et de placer les informations dans un dictionnaire. Ensuite, ces informations sont publiées sur un « topic » permettant au noeud de contrôle de recevoir les informations des Zooids.

De plus, le « bridge » lit les destinations voulues à partir du noeud de contrôle pour envoyer ces informations aux Zooids. Les Zooids doivent recevoir un type de message spécifique pour comprendre l'information reçue et le « bridge » prépare et envoie ce message permettant de compléter la communication entre les noeuds ROS et les Zooids.

4.3.3 Séquences centralisées

Un noeud de contrôle centralisé a donc été créé dans le but de tester les mouvements expressifs sur les Zooids plus facilement qu'avec l'architecture décentralisée complète. Le noeud obtient les positions actuelles à partir des messages envoyés par le « Zooids bridge » et enregistre ces

positions dans une liste. Ensuite, selon le mouvement d'essaim choisi, le noeud de contrôle calculera les prochaines destinations de tous les Zooids et enverra ces positions au «Zooids bridge» qui les transmettra aux Zooids. L'architecture de communication pour le contrôle centralisé des Zooids est présentée dans la figure 4.13.

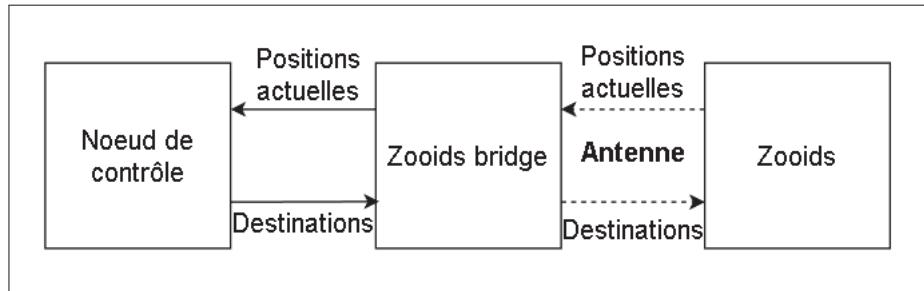


Figure 4.13 Architecture de contrôle

Les algorithmes de contrôle pour chacun des mouvements expressifs sont expliqués ci-dessous. Un menu permet à l'utilisateur de choisir un mouvement expressif à effectuer et la fonction respective est appelée pour calculer les prochaines destinations de chaque robot. Ces fonctions ont pour but de recréer le plus précisément possible les neuf animations utilisées dans l'étude utilisateurs. Le code est fait complètement en Python avec la version 3.8.10.

Initier la communication

Le premier mouvement pour initier la communication, figure 3.1, est simplement créé en calculant la position de chaque robot pour former un petit cercle et un grand cercle. En connaissant le nombre de robots actifs, en regardant la longueur du message venant du «bridge», les points d'un cercle peuvent être calculés avec l'équation 4.1.

$$x = \cos(2\pi/n * i) * r \text{ et } y = \sin(2\pi/n * i) * r \quad (4.1)$$

Avec i de 1 à n , n = nombre de robots et r = rayon du cercle.

Donc, selon ce qui est désiré pour la grosseur du cercle, dans notre cas un rayon de 50 divisions pour le petit cercle et 200 pour le grand cercle, les points de destination sont calculés pour chaque Zooid. Les robots se positionnent donc premièrement sur le petit cercle et lorsque tous les robots sont à une distance de moins de 25 divisions des points voulus du petit cercle ils sont demandés à se déplacer vers le gros cercle et ainsi de suite. Ceci permet alors d'avoir l'effet de pouls avec les Zooids formant un cercle et cette fonction s'adapte au nombre de robots actifs.

Le deuxième mouvement ayant pour but d'initier la communication, figure 3.2, est programmé de sorte qu'il peut aussi s'adapter au nombre de robots actifs. La position des robots est donc calculée à partir d'une position de base de laquelle on additionne ou soustrait, selon le numéro du robot dans la liste, un certain écart entre les robots. Lorsque les robots sont à une distance de moins de 25 divisions de leur destination, l'état change et la valeur de l'écart change aussi soit 25 divisions pour la ligne rétractée et 75 pour la ligne allongée.

Terminer la communication

La fonction pour créer le mouvement pour terminer la communication, figure 3.3, est simple. Peu importe dans quelle formation sont les Zooids, ils sont divisés en deux groupes, les chiffres pairs et impairs selon leur position dans la liste, et sont donnés une position extrême aléatoire. Pour éviter que les robots se positionnent au milieu de l'arène, où ils vont normalement effectuer les mouvements expressifs, la valeur aléatoire est petite et ajoutée à une valeur de base qui amène les robots vers une extrémité de l'arène soit vers le haut et la droite ou le bas et la gauche.

Aucun problème à signaler

Le mouvement pour signaler qu'il n'y a aucun problème, figure 3.4, a été programmé de façon très similaire au premier mouvement pour initier la communication. À l'aide de l'équation 4.1, tous les points du cercle sont calculés. Une fois que les Zooids sont près de leur destination, ils sont dirigés vers le prochain point sur le cercle permettant ainsi aux robots se déplacer

efficacement sur la circonférence du cercle. Ceci permet donc un déplacement synchronisé et fluide le long du cercle.

Requiert une intervention

Le premier mouvement ayant pour but d'indiquer à l'opérateur que la mission requiert une intervention, figure 3.5, est programmé de sorte que les trois premiers robots actifs dans la liste soient les robots en mouvement tandis que tous les autres sont immobiles. Trois points sont donc définis dans l'espace et les trois robots en mouvement se déplacent d'un point à l'autre créant un triangle en mouvement. Les autres robots sont positionnés en ligne et restent immobiles pendant que les trois robots en mouvement se déplacent autour d'eux.

Le deuxième mouvement pour demander une intervention, figure 3.6, est la fonction la plus simple des neuf animations. Le mouvement est créé seulement en envoyant tous les robots actifs au même point. Cependant, comme les robots ne peuvent pas être tous au même point en même temps, ils continuent à essayer de se rendre même lorsqu'ils sont en collision les uns avec les autres créant ainsi un mouvement, ressemblant une vibration, lorsque les robots se poussent mutuellement essayant d'atteindre leur destination. Ceci permet aussi d'entendre le bruit des roues tournant et glissant sur la plateforme de l'arène ajoutant à l'effet créé par le mouvement expressif des robots.

Problème de batterie

Pour signaler un problème de batterie, le premier mouvement expressif, figure 3.7, est une ligne diagonale où les robots « tombent » lentement jusqu'à obtenir une ligne droite, à l'horizontale pour l'observateur. Dans ce cas-ci, il est important de connaître où est positionné l'opérateur et comment il regarde les robots, car ceci aura un impact sur le positionnement de la ligne diagonale. Pour le travail avec les Zooids, nous assumons que l'observateur est devant l'arène de jeu et regarde les robots de haut, permettant de reproduire ce qui a été fait avec l'animation. Les Zooids sont donc positionnés sur une ligne diagonale et leur point de destination est défini selon

leur position dans la liste. Une fois que tous les robots sont à leur destination, les positions de la ligne horizontale sont envoyées aux Zooids créant ainsi l'effet de tombée.

Le deuxième mouvement expressif pour signaler un problème de batterie, figure 3.8, est une sinusoïde diminuant d'amplitude avec le temps. Les robots ont donc une position en X fixe et leur position en Y est calculée avec l'équation 4.2

$$y = K * \sin(t + (i * \pi * 0.25)) \text{ avec } K = 100 - 5 * t \quad (4.2)$$

La variable t varie avec le temps, elle augmente de 1 chaque demi-seconde, ce qui fait qu'après 10 secondes la sinusoïde a perdu toute son amplitude et est une ligne droite horizontale.

Bris de communication

La fonction pour le mouvement expressif du bris de communication, figure 3.9, ressemble beaucoup à la fonction pour le deuxième mouvement ayant pour but d'initier la communication, avec la différence que la ligne est à l'horizontale au lieu de la diagonale. Le début est donc exactement la même logique que pour le mouvement pour initier la communication, soit une ligne s'allongeant et se rétractant lorsque les robots atteignent leur position. Une fois que le cycle a été effectué trois fois, le robot avec l'identifiant #1 dans la liste, au milieu de la ligne, « tombe » et brise la ligne. Les robots avec un identifiant impair, à droite du robot #1, restent immobiles à partir de ce moment pendant que les trois autres robots continuent à se déplacer normalement selon l'écart calculé.

4.4 Conclusion

Pour conclure, les Zooids sont d'excellents petits robots permettant d'avoir un grand nombre de robots et ainsi avoir un essaim dans un petit espace et sans utiliser une énorme quantité de ressources. De plus, le fait qu'ils sont petits fait en sorte qu'ils sont transportables et permettent d'effectuer des études utilisateurs dans différents environnements. Finalement, ils sont idéals

pour le prototypage rapide des mouvements expressifs. Pour permettre des améliorations par itérations aux mouvements expressifs un premier contrôle centralisé est requis et les Zooids permettent d'effectuer ces tests rapidement et efficacement avant d'aussi être utilisés pour le contrôle décentralisé.

Cependant, suite au travail effectué sur les Zooids, quelques inconvénients ont été identifiés même après les améliorations amenées aux petits robots. D'abord, ils ne sont pas conscients de leur environnement ce qui fait en sorte que si l'arène n'est pas complètement vide, ils entreront en contact avec les objets sur leur route. De plus, si le noeud de contrôle, centralisé ou décentralisé, ne comprend pas d'algorithme d'évitement d'obstacle entre les robots, les Zooids tenteront toujours de se déplacer en ligne droite et entreront en contact entre eux si leur route se croise. Il n'y a présentement aucune façon d'ajouter un évitement d'obstacle directement dans le micrologiciel des Zooids comme ils n'ont pas de capteurs leur permettant de connaître leur environnement. Dans notre travail, le contrôle centralisé ne comprend aucun évitement d'obstacle ce qui peut poser problème lors de la performance des mouvements expressifs. Ensuite, les Zooids ont besoin du projecteur pour fonctionner ce qui réduit la possibilité de leur utilisation dans certains environnements. Les Zooids doivent être dans un environnement où l'intensité lumineuse est basse, sinon ils ne peuvent pas lire correctement la séquence provenant du projecteur. Ceci demande alors un ajustement à l'environnement comme l'ajout d'une tente à l'extérieur pour réduire l'impact des sources lumineuses autres que le projecteur. Finalement, l'assemblage des Zooids est un travail demandant une expertise spécifique pour réaliser un travail très minutieux et complexe. Le manque de robustesse de certaines pièces et des soudures sur les PCBs fait en sorte qu'il est difficile d'assembler et de maintenir une flotte de Zooids. De plus, les problèmes de production de pièces électroniques liés à la Covid-19 complexifient la production de nouveaux PCBs pour la fabrication de nouveaux Zooids.

CHAPITRE 5

PROGRAMMATION DU CONTRÔLE D'ESSAIM

Le contrôle décentralisé est un paradigme de programmation complexe qui requiert un expert et rend son utilisation réelle plus complexe. Les robots doivent agir avec uniquement une connaissance de leur environnement rapproché, c.-à-d. ce qu'ils peuvent voir avec leurs capteurs limités, et une communication simple avec leurs voisins. C'est la collaboration par cette communication qui étend le potentiel de l'essaim : elle leur permet de réagir correctement à leur environnement et aux tâches qui leurs sont demandées sans avoir recourt à un ordinateur centralisé prenant les décisions. La question est cependant : comment programmer ces robots pour qu'ils obtiennent ces informations et prennent des décisions en groupe ?

Pour effectuer le contrôle de manière complètement décentralisé, nous utilisons le langage de programmation Buzz (Pincioli & Beltrame, 2016b), un langage dédié à la programmation d'essaim. De plus, nous sommes en mesure d'utiliser Buzz avec le « Robot Operating System » (ROS), un outil fiable grandement utilisé permettant de contrôler et de communiquer avec les robots à un plus bas niveau, pour le contrôle des robots, autant en simulation qu'avec les robots réels. Ce chapitre décrit d'abord l'utilisation et le fonctionnement du langage de programmation Buzz pour ensuite expliquer l'adaptation de deux algorithmes de contrôle d'essaim existants dans le but d'avoir une solution complète pour effectuer des mouvements expressifs lors d'une mission d'exploration.

5.1 **Programmation d'essaim**

Buzz voit un essaim de robots comme étant un groupe de robots interagissant les uns avec les autres. Chaque robot exécute une version de la « Buzz Virtual Machine » (BVM) ce qui permet l'utilisation des mêmes scripts Buzz simultanément sur chaque robot tout en intégrant la structure de communication entre les voisins. Buzz est basé sur le concept de la communication localisée, c'est-à-dire un type de communication qui envoie la position du robot avec le message dans le référentiel du robot recevant le message. Ceci permet donc à tous les robots de connaître

la position relative de tous les robots de l'essaim tant et aussi longtemps que la chaîne de communication entre les robots de l'essaim est active. Bref, même en communiquant seulement avec leurs voisins, chaque robot est en mesure de connaître certains états de tous les robots de l'essaim.

Ce langage de programmation crée la flexibilité nécessaire pour notre déploiement en utilisant des primitives d'essaim permettant la gestion de l'essaim. Le travail effectué par Pincioli & Beltrame (2016a) explique plus en détail comment l'objet de l'essaim a été développé et dans quel objectif. Comme mentionné dans leur travail « the swarm construct of Buzz allows the programmer to tag a set of robot according to a certain condition » ou, en d'autres mots, Buzz permet d'assigner certains robots à une certaine valeur selon une certaine condition. En utilisant l'outil de gestion d'essaim de Buzz, nous sommes capables de diviser l'essaim en sous-essaims selon leur condition, par exemple leur disponibilité à effectuer une certaine tâche. Un sous-essaim peut donc recevoir différentes tâches sans être retenu par les autres sous-essaims. Ceci permet entre-autres, à n'importe quel moment, de mettre tous les robots respectant une certaine condition, par exemple tous les robots n'ayant pas de tâches assignées, dans un sous-essaim et de les utiliser pour effectuer une tâche différente des autres sous-essaims, par exemple pour effectuer un mouvement expressif ou une formation permettant de transmettre de l'information à l'opérateur.

D'autres objets sont disponibles dans Buzz facilitant l'implémentation de différents algorithmes de comportements d'essaims. D'abord, l'objet des voisins contient de l'information par rapport aux voisins à l'intérieur de la portée de communication, comme leur identifiant, position et autres. De plus, l'objet de la stigmergie est un tuple, un élément à deux variables, permettant de communiquer des messages entre les robots de manière robuste assurant qu'il n'y a pas de pertes d'information et pouvant se propager à tous les robots (Pincioli, Lee-Brown & Beltrame, 2016).

Les deux algorithmes utilisés pour créer la solution finale ont été développés avec Buzz. Cette section présentera comment ces deux algorithmes sont utilisés dans le but de créer un retour d'information pour l'opérateur lorsque les robots sont en cours de mission d'exploration. En résumé, la stratégie implique l'utilisation des primitives de gestion d'essaim et de sous-essaims

dans Buzz permettant de changer certains robots d'un comportement à un autre ; une première démonstration de cette fonctionnalité. Plusieurs sous-essaims sont créés dans l'algorithme d'exploration, un pour les robots essentiels gardant le lien de communication principal actif et d'autres pour les robots de chacun des autres liens. Ces liens non essentiels, que l'on peut aussi appeler liens redondants, sont utiles pour améliorer la robustesse de l'algorithme tout en permettant une meilleure couverture des robots. De plus, avec l'utilisation des sous-essaims, certains robots peuvent être changés de tâches et chaque sous-essaim peut être contrôlé individuellement. Par la suite, les robots peuvent revenir à leur tâche initiale d'exploration et ainsi de suite, permettant à l'opérateur d'utiliser les robots pour effectuer plusieurs tâches simultanément tant et aussi longtemps qu'un nombre de robots minimum est maintenu dans la chaîne de communication principale.

Pour être en mesure de déployer ces algorithmes sur de vrais robots, nous utilisons ROS. Cependant, Buzz ne fonctionne pas nativement dans ROS, nous utilisons donc ROSBuzz, l'implémentation ROS de la machine virtuelle de Buzz (BVM) (St-Onge *et al.*, 2020b). ROSBuzz combine ROS et Buzz avec l'utilisation d'un noeud ROS qui inclut la machine virtuelle de Buzz tout en utilisant le protocole « Micro-Air Vehicle Link » (MAVLink), disponible avec l'implémentation MAVROS, pour communiquer entre les membres de l'essaim. En utilisant un fichier de lancement, l'utilisateur doit mentionner quel comportement d'essaim il veut utiliser dans Buzz et ROSBuzz démarrera la boucle principale ROS avec toutes les configurations et les fonctions, « subscribers », « publishers », « callbacks » et services, requises pour faire fonctionner l'architecture ROS. Le script Buzz est ensuite exécuté dans la BVM et reçoit les messages ROS à travers la communication MAVROS. Bref, ceci permet d'utiliser Buzz sur de vrais robots utilisant ROS, comme les Zooids ou autres.

Cependant, avant de faire des essais sur de vrais robots, nous utilisons le simulateur Gazebo pour simuler les comportements d'essaim. Nous utilisons 12 drones, chacun exécutant le même script Buzz permettant d'effectuer une mission d'exploration tout en incluant un retour d'information visuel à l'opérateur avec l'implémentation de mouvements expressifs. Les deux algorithmes

ayant été adaptés et combinés pour obtenir la solution finale sont décrits dans les sections suivantes : l'algorithme d'exploration et l'algorithme de formation de graphe acyclique.

De plus, une fois que le comportement en simulation est satisfaisant, nous testons ces comportements directement sur les Zooids. Dans ce cas, les Zooids peuvent agir comme proxy représentant la mission en cours créant alors une interface avec des robots réels pour l'opérateur. Pour ce faire, nous utilisons les Zooids pour représenter les déplacements des drones en simulation dans leur arène. Donc, pendant que la simulation et Gazebo sont exécutées sur l'ordinateur, les positions des drones sont envoyées au « Zooid bridge » pour ensuite envoyer ces positions aux Zooids. Dans la simulation, les drones sont limités à un terrain de 140 m par 67 m qui est représenté par l'arène des Zooids. De plus, la simulation publie sans arrêt la position de tous les robots de l'essaim sur un sujet ROS qui est donc lu par le « Zooid bridge » pour obtenir la position à envoyer aux Zooids. Cependant, cette position est en GPS et doit donc être convertie dans le référentiel des Zooids pour leur envoyer la bonne position. Donc, une distance en mètre est calculée par rapport à l'origine du terrain des drones et cette distance est transformée dans le référentiel des Zooids pour obtenir la position de chaque drone sur l'arène des Zooids. Bref, une fois que cette conversion est faite, le « bridge » envoie les positions aux Zooids correspondants ce qui permet de représenter la simulation avec de vrais robots.

5.1.1 Scénario d'exploration

Une fois que la solution fonctionne sur les robots, le but est de créer un scénario d'exploration pour tester l'efficacité des mouvements expressifs dans un scénario réel. L'objectif est donc que les Zooids représentent les drones lors de leur exploration, mais ils font aussi partie de l'interface tangible où l'opérateur peut déplacer les Zooids dans le but de contrôler l'essaim.

Le scénario créé est alors le suivant : l'opérateur a sous la main un essaim de 12 robots et il doit explorer un certain territoire pour trouver des éléments d'intérêt. L'opérateur et les robots connaissent une carte pré-déterminée du territoire, mais des obstacles inconnus peuvent tout de même s'y trouver. C'est face à des problèmes de la sorte que l'essaim doit communiquer

avec l'opérateur pour qu'il soit en mesure de régler le problème. Nous avons donc identifié trois scénarios possibles comportant chacun un problème différent possible lors d'une mission d'exploration représentant les communications demandant une action de l'opérateur soit : problème de batterie, bris de communication et requiert une intervention. Dans ces scénarios, l'opérateur doit déployer deux robots à la fois, lorsqu'ils sont requis, simulant une station de décollage.

D'abord, lorsque les robots manquent de batterie lors du déplacement vers l'un des éléments d'intérêt, certains d'entre eux vont complètement s'arrêter mettant en danger le succès de la mission. Cependant, si l'essaim effectue le mouvement expressif du problème de batterie avant que certains des robots s'éteignent, l'opérateur peut agir en conséquence et ramener les robots au point de départ pour recharger leurs batteries.

Ensuite, s'il y a un bris de communication entre les robots, l'essaim arrête son exploration et l'opérateur peut difficilement comprendre le problème. Cependant, si les robots disponibles effectuent le mouvement expressif pour informer l'opérateur, celui-ci sera en mesure de comprendre le problème et de déployer de nouveaux robots dans le but de réparer le bris de communication.

Finalement, si les robots rencontrent un obstacle absent de la carte, l'opérateur doit leur mentionner d'utiliser un autre chemin pour se rendre au but. Donc, lorsque les robots sont bloqués par l'obstacle, ils effectuent le mouvement expressif demandant une intervention à l'opérateur. À ce moment, l'opérateur peut réagir et déplacer le but de l'essaim pour contourner l'obstacle et être en mesure de continuer l'exploration.

5.2 Algorithme d'exploration

L'algorithme principal utilisé pour la mission d'exploration est celui d'exploration en chaîne publié par Varadharajan, St-Onge, Adams & Beltrame (2020). L'algorithme original crée progressivement une formation en chaîne où tous les robots suivent le travailleur, ou « worker », le robot de tête, vers un but. Les robots créant la chaîne sont appelés réseateurs, ou « networker »,

et leur seule tâche est de maintenir la connectivité entre le travailleur et la station de contrôle. Ce processus est illustré à la figure 5.1. Pour améliorer la robustesse de la mission par rapport aux problèmes de robots ou de communication, les réseauteurs peuvent créer plusieurs chaînes de connectivité entre le travailleur et la station de contrôle. Le travailleur élit premièrement plusieurs réseauteurs pour commencer à créer les chaînes comme montré en simulation à la figure 5.4. Ensuite, chaque robot élu doit élire un autre robot pour le suivre et continuer la chaîne de connectivité avec la station de contrôle et ainsi de suite jusqu'à ce que le travailleur arrive à son but. La figure 5.1 montre ce processus.

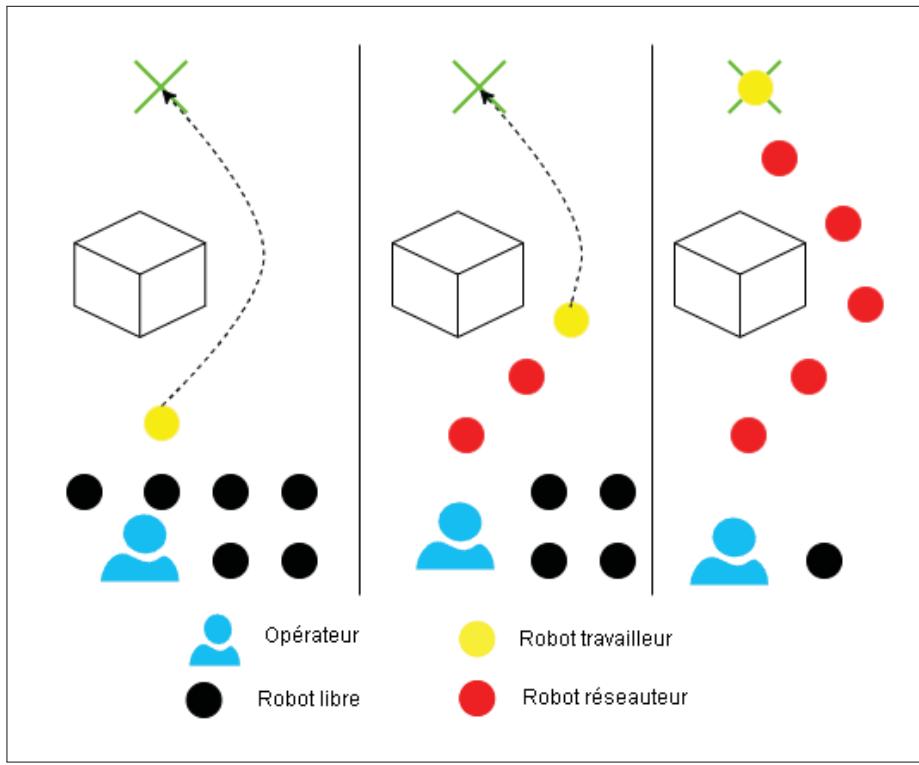


Figure 5.1 Formation de la chaîne pour se rendre à un but lors de l'exploration

Une fois qu'un but est décidé par l'opérateur et envoyé à l'essaim, pour éviter une grande quantité de messages entre les robots, c'est le travailleur qui calcule le chemin vers le but et qui le transmet aux robots réseauteurs qu'il élit. Ceci fait en sorte que, comme le robot travailleur est aussi le robot de tête, il sera le premier à rencontrer des obstacles et il pourra donc faire les ajustements nécessaires. Donc, comme illustré à la figure 5.2, le travailleur, une fois avoir calculé

la trajectoire, décidera le nombre de chaînes et le nombre de robots requis dans chaque chaîne pour se rendre jusqu'au but. Lorsque plus de robots sont requis pour se joindre à la chaîne, la requête passe par les réseauteurs jusqu'à ce qu'un robot libre se joigne à la chaîne. Lorsque le bon nombre de robots est joint à la chaîne, les robots vont se déplacer vers leur but. À l'aide de l'équation 5.1, les robots vérifient si la distance entre eux et leur parent est plus petite que la distance critique de communication. Si oui, aucun problème les robots continuent à se déplacer, sinon la chaîne rétrécit dans le but de réduire la distance entre les robots et ne pas dépasser la distance de communication critique entre les robots.

$$f_c(d_i^C) \begin{cases} 1 \text{ if } d_i^C \leq d_s \\ 0 \text{ otherwise} \end{cases} \quad (5.1)$$

Pour se déplacer, les robots suivent la loi de contrôle de l'équation 5.2.

$$\begin{aligned} u_i &= \arg \min_{u'_i \in Au_i} \sum_{j \in N_i^{close}} \alpha \frac{1}{tvo_j(u'_i)} + \|u_i^{pref} - u'_i\| \\ Au_i &= \left\{ u'_i \mid \|u'_i\| \leq u_i^{conn} \right\}, u_i^{conn} = \min_{j \in P \cup C} \frac{(d_s - d_{ij})}{2\Delta t} \end{aligned} \quad (5.2)$$

Cette équation permet de trouver un vecteur de vitesse adapté à la situation en s'éloignant des collisions futures tout en s'approchant du but en restant dans une plage acceptable Au_i . La fonction $tvo_j(u'_i)$ calcule une pénalité selon les collisions futures tandis que la norme est une pénalité de déviation de la trajectoire préférée u_i^{pref} . N_i^{close} est un ensemble de voisins à l'intérieur du rayon de collision tandis que u_i^{conn} est la vitesse maximale permise. Ceci permet alors aux robots de se déplacer vers leur but tout en évitant les collisions entre les membres de l'essaim pouvant mettre en danger la réussite de la mission.

Dans le cas d'un problème avec un ou certains des robots, l'algorithme est dit robuste, car il possède la capacité de réparation autonome ce qui permet de réparer les liens brisés dans la

chaîne par un ou des robots non fonctionnels. Ceci est donc fait à l'aide d'une communication locale entre les membres de l'essaim où la distance de communication fonctionnelle pour chaque robot permettra de déterminer la distance maximale entre les réseauteurs dans la chaîne. Donc, si un membre de l'essaim, le robot de tête par exemple, veut se déplacer vers son but, mais que la distance avec le lien précédent dans la chaîne devient trop grande, le robot arrête et attend que le lien précédent se rapproche dans le but de maintenir la communication. S'il est détecté que le robot du lien précédent est non fonctionnel, la chaîne doit être réparé. Cependant, ceci est complexe comme maintenant l'essaim est divisé en deux groupes ne pouvant pas communiquer et s'entendre sur la façon de réparer la chaîne. Ceci est donc effectué par l'ajout d'une communication périodique venant du travailleur informant chaque individu de l'essaim de sa position dans la chaîne et la position de ses voisins. Donc, si la communication est brisée, les robots sont en mesure de savoir quel robot ne répond plus et l'endroit où il faut réparer la chaîne. Ceci permet alors aux robots entourant le robot défectueux de tenter de se rapprocher de l'un l'autre dans le but de réparer la communication.

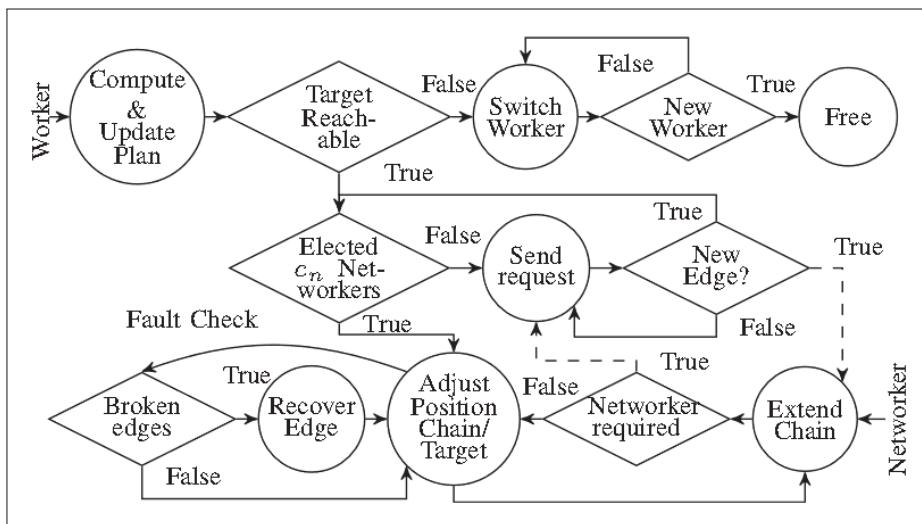


Figure 5.2 Jeu entre le travailleur et les réseauteurs
Tirée de Varadharajan *et al.* (2020, p.5)

5.3 Formation par graphe acyclique

Comme expliqué dans les sections précédentes, nous croyons qu'utiliser des mouvements expressifs permet de transmettre efficacement de l'information vers l'opérateur lors d'une mission d'exploration. Pour effectuer ces mouvements de manière complètement décentralisée, nous utilisons une version modifiée de l'algorithme de formation progressive par graphe acyclique présenté par Li *et al.* (2019). L'algorithme de formation par graphe acyclique (AGF) positionne les robots selon un graphe prédéfini où les positions de chaque robot sont définies par rapport au robot initiateur de la formation. Le robot initiateur est élu entre les robots selon celui du sous-essaim ayant l'identifiant le plus bas, mais plusieurs autres règles d'élection sont possibles. Donc, lorsque les robots changent de comportement, ils communiquent ensemble, parmi les robots du sous-essaim, pour déterminer qui sera l'initiateur. La formation initiale est donc celle d'un graphe acyclique où les robots peuvent trouver leur position utilisant la position d'un ou deux autres robots, appelés parents, dans la forme de référence. Tous les robots connaissent donc la forme à produire, mais aucun n'a de position prédéfinie. La forme finale est construite dynamiquement et itérativement : chaque nouveau robot se joint à la forme seulement après avoir reçu la permission par un des parents en utilisant seulement de la communication locale entre les voisins. Comme montré dans la figure 5.3, la formation par graphe acyclique fonctionne à l'aide d'une machine d'état pour prendre les robots d'un état libre, lorsqu'ils quittent la formation en chaîne, et les amener jusqu'à un état « barré » où ils pourront effectuer les mouvements expressifs.

D'abord, lorsqu'ils sont dans l'état libre, les robots recherchent une étiquette, ou position, libre dans le graphe. Si le ou les parents de cette étiquette, donc des robots ayant déjà joint le graphe, est dans la mire du robot libre, celui-ci passera en mode « demandant ». Dans ce mode, le robot envoie une requête au(x) parent(s) et passera en mode « se joint » si la requête est acceptée et retournera en mode libre si la requête n'est pas acceptée, ce qui est fort probable lorsque plusieurs robots libres envoient chacun une requête pour la même étiquette. Il est important que les parents restent toujours dans la mire du robot demandant lors de cette étape, la loi de contrôle dans ce mode suit donc une force d'attraction entre ces robots pour garder la proximité

entre les robots tout en utilisant une force de répulsion avec les robots libres environnants pour éviter les collision. L'équation de cette loi de contrôle est la suivante :

$$u = f(F_p + F_a) \quad (5.3)$$

où F_p est la force d'attraction créée par les robots parents tandis que F_a est la force de répulsion avec les robots voisins permettant d'éviter les collisions. Le vecteur de force résultant est ensuite transformé en vecteur de vitesse permettant au robot de se déplacer vers ses parents tout en évitant les collisions avec les autres robots.

Dans l'état « se joint », une fois la requête acceptée, le robot se déplace à sa position dans le graphe selon la position relative entre lui et son ou ses parents. La loi de contrôle est défini par des champs de différence de potentiel permettant de créer des forces pouvant ensuite être transformées en vecteur de vitesse pour que le robot se déplace. La loi de contrôle suivi par les robots dans le mode « se joint » est la suivante :

$$u_i = -\nabla\phi_i(p_i, p_j, p_k) \quad (5.4)$$

Ceci permet alors au robot (i) de connaître la position de ses parents (j et k) dans son référentiel local et de calculer son but, connaissant le graphe à produire, à partir de la position de ses parents. Ensuite, la loi de potentiel de Lennard-Jones est utilisée pour calculer les forces F_j et F_k qui s'additionnent pour former la force F_i qui sera transformée en vecteur vitesse déplaçant alors le robot. Ceci est effectué jusqu'à ce que le robot arrive à sa position finale et passe en mode « est joint ». Dans ce mode, le robot pourra alors traiter les requêtes des étiquettes dont il est le parent. Une fois que toutes les positions du graphe sont remplies ou que tous les robots de l'essaim se sont joint au graphe, les robots passent à l'état « barré ».

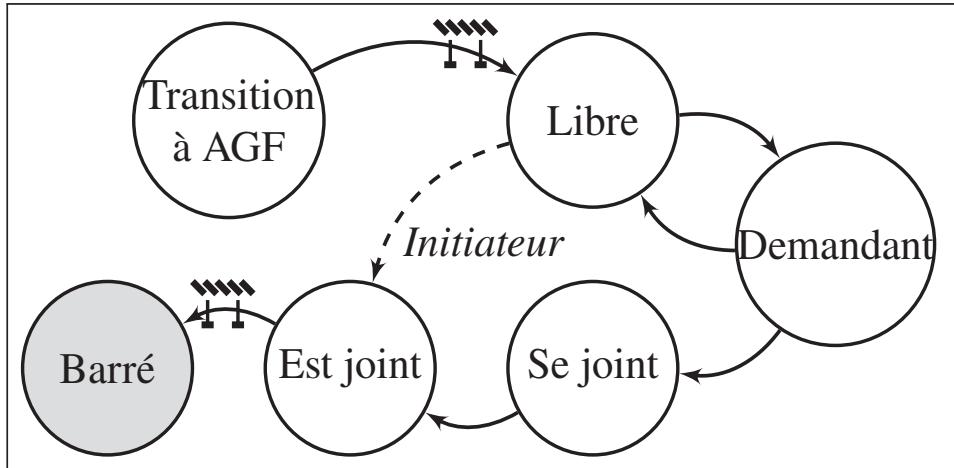


Figure 5.3 La machine d'état de la formation par graphe acyclique adaptée
Tirée de Li *et al.* (2019, p.8)

5.3.1 Modifications aux algorithmes de base

Dans le but de créer la solution finale, les deux algorithmes ont dû être adaptés pour être en mesure de les combiner. D'abord, dans l'algorithme d'exploration en chaîne, l'algorithme original n'inclue aucune façon de différencier les réseauteurs, car ils sont tous identiques. Donc, nous avons modifié le code original pour inclure un identifiant d'essaim dans le message de requête envoyé par le travailleur aux premiers réseauteurs de chaque chaîne. L'identifiant d'essaim est donc différent pour tous les premiers réseauteurs élus par le travailleur et ces derniers peuvent créer des sous-essaïms avec ce numéro. Cet identifiant est donc passé aux prochains robots se joignant à la chaîne. En acceptant la requête, les nouveaux réseauteurs se joignent à l'essaim de leur parent pour que chaque chaîne soit représentée par un différent sous-essaim. Cette nouvelle fonctionnalité ouvre la possibilité d'utiliser les liens secondaires ou redondants (seulement un lien est requis pour maintenir la connectivité) pour différentes tâches sans mettre en péril la mission.

Bref, à n'importe quel moment lors de la mission, un sous-essaim non essentiel peut changer de comportement vers la formation par graphe acyclique tant qu'il y a un minimum de robots qui maintiennent la communication dans la chaîne principale. Au besoin, et si disponibles, des

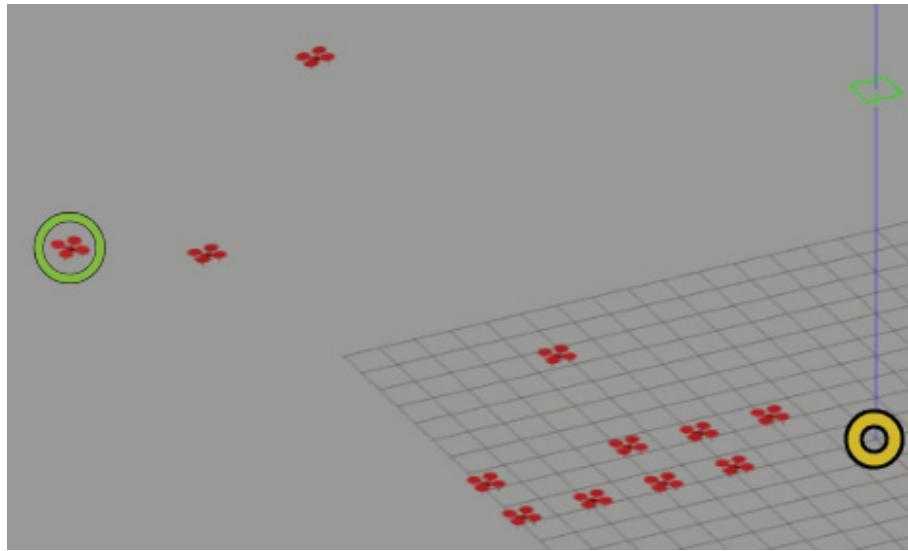


Figure 5.4 Drones simulés dans Gazebo. Quatre ont décollé, le travailleur (cercle vert) et trois réseauteurs font le lien avec la station de contrôle (cercle jaune)

robots au sol encore libres peuvent aussi joindre la formation par graphe acyclique si jamais il y a un manque de robots. Cependant, il faut gérer la transition correctement de sorte qu'il n'y ait pas d'interférence entre les sous-essaims effectuant différents comportements. Si les robots ayant changé de formation sont encore considérés comme actif dans la formation en chaîne, l'algorithme pourrait faire appel à des robots n'étant plus disponibles. Ceci ferait en sorte que si un problème survenait à un robot de la chaîne essentiel, l'algorithme ne réparerait pas la chaîne immédiatement comme la deuxième chaîne est active. De plus, les deux algorithmes utilisent une communication entre ses membres et pour éliminer la possibilité de confusion entre les messages des deux algorithmes, les protocoles d'écoute des messages de la formation en chaîne sont arrêtés sur les robots changeant de comportement vers la formation par graphe acyclique. Une fois que cela est fait, le sous-essaim effectue la formation demandée et le mouvement expressif le cas échéant. C'est le travailleur qui envoie la commande au sous-essaim d'effectuer la formation selon la situation. Comme le travailleur est le robot de tête et est le premier à être informé des différentes situations, il envoie une variable « statut » aux différents membres de l'essaim à travers la stigmergie de Buzz dans le but d'informer tous les membres de l'essaim de la situation actuelle de la mission. Ceci permet alors aux différents robots de l'essaim de

réagir selon la situation actuelle et d'effectuer les mouvements expressifs requis pour en informer l'opérateur. Une chaîne complète est présentée à la figure 5.5.

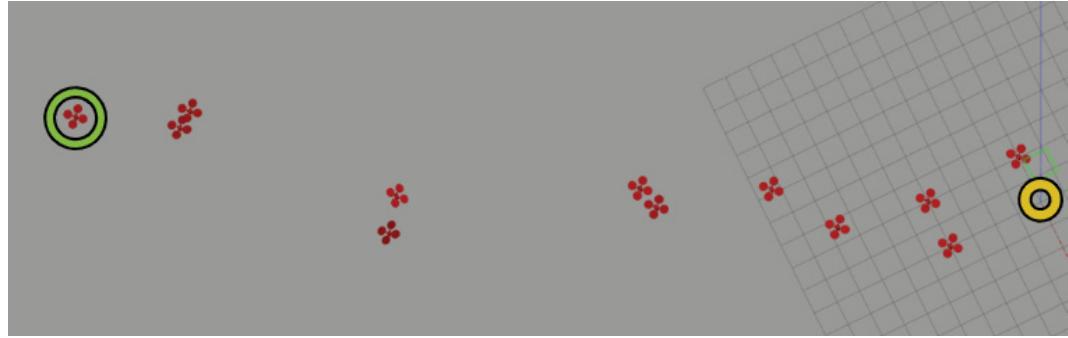


Figure 5.5 Exemple de chaîne avec un travailleur (vert) et 10 réseateurs forment deux liens avec la station de contrôle (jaune)

De plus, les deux algorithmes adaptés pour la solution utilisent le mécanisme de barrière. Ce mécanisme permet d'attendre que tous les robots de l'essaim soient prêts ou qu'il y ait un consensus avant de faire la transition vers un nouvel état. Comme dans ce cas-ci nous utilisons des sous-essaïms, cette barrière a été adaptée pour questionner seulement le sous-essaïm actif, c.-à-d. les robots exécutant le même comportement par exemple la formation par graphe acyclique, au lieu de tous les robots de l'essaim. La dernière barrière dans la formation par graphe acyclique permet d'effectuer la transition vers l'état « barré » qui se fait seulement lorsque toutes les positions de la forme sont occupées par un robot, ou que tous les robots disponibles sont à leur position (dans le cas qu'il y aurait moins de robots disponibles que de positions dans le graphe). Lorsque les robots font la transition, ils reçoivent ensuite de nouvelles instructions selon leur position et la forme exécutée pour effectuer des mouvements expressifs.

Pour illustrer l'intégration complète de l'algorithme, nous allons regarder l'exemple d'un essaim mentionnant à son opérateur qu'il a atteint son but et qu'il veut initier la communication. La figure 5.6 résume les trois étapes principales : 1- La formation en chaîne démarre et le travailleur commence à se diriger vers son but en demandant à deux robots de le suivre en tant que réseateurs ; 2- Le travailleur, suivi de deux chaînes chacune étant son propre sous-essaïm, atteint son but et un sous-essaïm se prépare à changer vers la formation par graphe acyclique ; 3- La

formation par graphe acyclique crée une forme prédéfinie et commence ensuite le mouvement expressif relié à cette forme pour informer l'opérateur de la situation de l'essaim pendant que l'autre chaîne maintient la connectivité avec la station de contrôle. La figure 5.7 illustre la dernière étape dans la simulation.

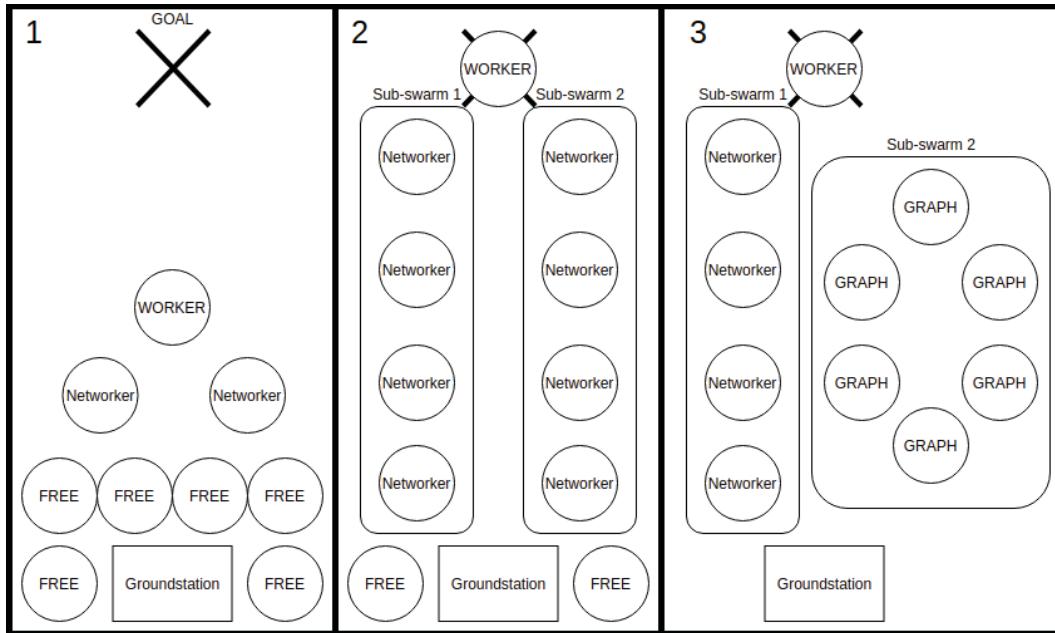


Figure 5.6 Étapes principales pour compléter l'exploration et le retour visuel vers l'opérateur par mouvements expressifs

5.3.2 Programmation des mouvements expressifs

Lors du passage en mode *expressif*, toutes les positions sont étiquetées et identifiées comme les noeuds d'un graphe permettant de contrôler chaque noeud individuellement une fois que le graphe est barré. À partir de la formation initiale, donc les positions prédéfinies dans le graphe, nous pouvons générer des mouvements cycliques, des translations, rotations, vibrations, etc. Les vitesses peuvent aussi être modifiées individuellement pour chaque robot, selon leur étiquette, pour effectuer les différentes séquences de déplacements menant aux mouvements expressifs. Cette section expliquera les techniques utilisées pour effectuer les mouvements expressifs de manière décentralisée. Cependant, ce ne sont pas toutes les animations qui ont été implémentées dans cet algorithme, seulement celles déployées dans le scénario d'étude

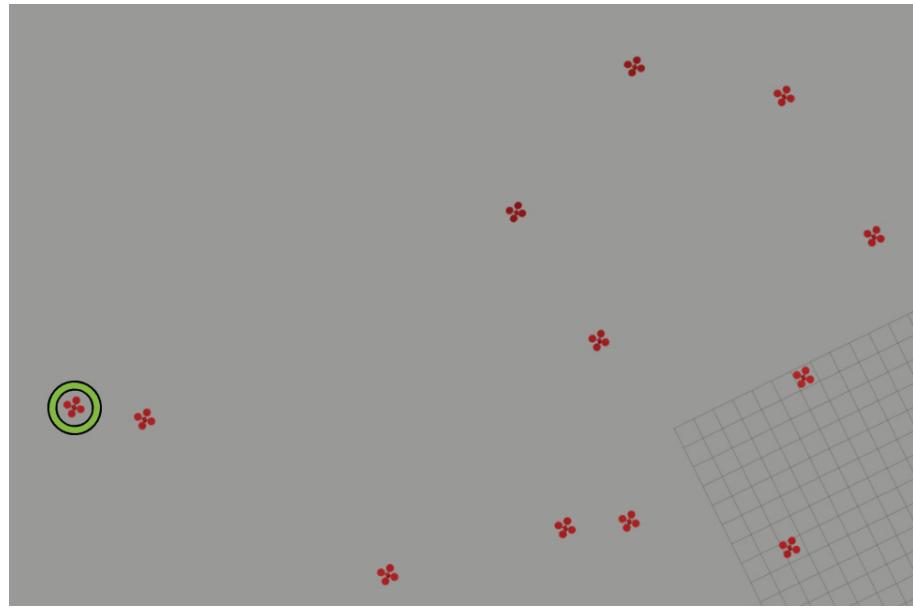


Figure 5.7 Un sous-essaim forme la chaîne de connectivité avec le travailleur (cercle vert) au but, pendant que les autres robots forment un graphe

utilisateurs avec les Zooids. Les animations implémentées ont été choisies selon leur utilité basée sur l'étude utilisateurs en ligne effectuée avec les animations et sont les suivantes : initier la communication (les deux animations), terminer la communication, aucun problème ou succès, requiert une intervention (deuxième animation), problème de batterie (première animation) et bris de communication.

Initier la communication

Le premier mouvement pour initier la communication a été implémenté avec l'algorithme décentralisé à l'aide d'un graphe formant un cercle avant d'ajouter les mouvements pour chaque robot permettant de créer le mouvement expressif. Une fois que les robots passent à la formation par graphe acyclique, l'algorithme utilise le graphe détaillé au tableau I-1 pour former un cercle avec six robots. Une fois que le cercle est formé et que les robots passent à l'état barré, les robots peuvent commencer le mouvement expressif. D'abord, le centre du cercle est calculé à partir de l'objet des voisins où la position de chaque voisin est lue et emmagasinée à chaque itération de la boucle de la machine virtuelle de Buzz. Ceci permet donc de trouver le centre du cercle par

rapport au robot avec l'utilisation de l'équation 5.5. Comme l'algorithme est décentralisé, ces étapes se font sur tous les robots en même temps et donc tous les robots calculent ces éléments et obtiennent les positions, distances et angles dans leur propre référentiel. Donc, chaque robot connaît la position du centre du cercle dans leur référentiel ce qui leur permet de se déplacer vers le centre ou au contraire s'éloigner du centre. Il ne reste donc qu'à avoir une certaine période de temps permettant aux robots de savoir quand ils doivent se rapprocher ou s'éloigner du centre du cercle. Les robots changent donc de direction, dans ce cas-ci, après 20 itérations, ou pas, de la boucle de la machine virtuelle de Buzz. Après ces 20 pas, ou deux secondes comme la boucle est effectuée 10 fois par seconde, les robots inversent le vecteur utilisé pour se déplacer et commencent donc à se déplacer dans l'autre direction et ainsi de suite créant ainsi le mouvement expressif.

$$(x_c, y_c) = \sum_1^n \frac{(x_v, y_v)}{n} \rightarrow (x_v, y_v) : \text{position de chaque voisin} \quad (5.5)$$

et n = nombre de voisins

Le deuxième mouvement pour initier la communication est premièrement formé par un graphe positionnant les robots en ligne droite détaillé dans le tableau I-2. Une fois que les robots ont formé la ligne et qu'ils passent à l'état barré, ils peuvent commencer leur déplacement. Pour ce mouvement, les robots ayant un angle de 0.0 dans le graphe vont se déplacer dans un sens et tous les autres robots se déplacent dans l'autre sens s'éloignant les uns des autres, tous à la même vitesse. Après 25 pas, le vecteur de direction change et les robots se rapprochent donc les uns des autres et ainsi de suite créant le mouvement expressif.

Terminer la communication

Pour indiquer que les robots veulent terminer la communication, les robots se positionnent premièrement sur un graphe qui peut sembler aléatoire. Le graphe détaillé dans le tableau I-3 est formé pour qu'une fois que les robots se déplacent, leur position semble aléatoire. Une

fois que les robots sont en position et qu'ils passent à l'état barré, ils sont tous très près les uns des autres. Ceci ne serait peut-être pas le cas lors d'une mission d'exploration où leur positionnement serait la position de fin de n'importe quel autre mouvement expressif, mais sans scénario d'exploration défini, ce graphe permet de mettre l'accent sur le mouvement. Donc, une fois en état barré, les robots calculent la position centrale entre tous les robots de l'essaim à l'aide de l'objet de voisins avec l'équation 5.5. Les robots vont ensuite s'éloigner de ce centre, en calculant le vecteur inverse, pendant un certain temps, dans ce cas-ci 60 pas, soit six secondes comme la machine virtuelle de Buzz s'exécute 10 fois par seconde. Après ces 60 pas, les robots s'immobilisent ce qui crée une forme semblant aléatoire et l'immobilité des robots signifie la fin de la communication de la même façon que le mouvement expressif a été créé avec l'animation.

Aucun problème

Le mouvement représentant qu'il n'y a aucun problème utilise le même graphe que le premier mouvement pour initier la communication soit le graphe détaillé au tableau I-1. Une fois que le cercle est formé, les robots utilisent l'objet des voisins pour obtenir le centre du cercle dans leur référentiel. Ceci leur permet ensuite de calculer le vecteur perpendiculaire au centre du cercle et de se déplacer dans la direction de ce vecteur. Comme le centre du cercle et le vecteur perpendiculaire sont recalculés à chaque pas, les robots vont tourner autour du centre du cercle et donc ceci forme un cercle qui tourne comme ce qui a été créé avec l'animation.

Requiert une intervention

Ce mouvement peut être effectué peu importe la position actuelle des robots, mais comme il est probable que dans un tel scénario le mouvement soit effectué suite au mouvement pour initier la communication, le graphe utilisé forme un grand cercle (imparfait, détaillé dans le tableau I-4) avant d'effectuer la logique de déplacement. Une fois avoir formé le graphe, les robots utilisent l'objet des voisins pour trouver la position centrale de l'essaim, utilisant l'équation 5.5, et se déplacent directement tous vers ce point jusqu'à tous être au même point. Comme tous les robots se dirigent vers le même point, l'algorithme d'évitement de collision utilisé sur les robots aura

un grand impact sur les résultats du mouvement. Sans évitement de collision, les robots entreront en contact, ou dans le cas de la simulation avec des drones à différentes altitudes ils seront au même point dans le plan à des altitudes différentes, tandis qu'un algorithme d'évitement de collision arrêterait les robots avant d'arriver au point voulu, mais l'hésitation entre les robots et les commandes envoyées permettraient tout de même de créer l'effet voulu avec le mouvement expressif.

Problème de batterie

Pour le mouvement exprimant un problème de batterie, un graphe formant une ligne diagonale est utilisé. Le tableau I-5 détaille le graphe de la diagonale. Pour ce mouvement expressif, la position et l'orientation de l'opérateur sont importantes pour que le message puisse être bien interprété. Nous assumons que l'opérateur regarde les robots de haut, donc il regarde le plan XY, avec X positif allant vers la droite ce qui permettra de représenter le mouvement de la même façon qu'avec l'animation. Donc, une fois que la diagonale est formée, les robots ont tous une vitesse différente, en Y négatif, permettant ainsi de revenir tranquillement se positionner sur une ligne droite horizontale. Ceci est fait avec des vitesses précises pendant 60 pas le temps que les robots forment la ligne horizontale permettant ainsi de créer le mouvement expressif.

Bris de communication

Le mouvement pour le bris de communication utilise le graphe en ligne du tableau I-2 pour la formation initiale et c'est ensuite une question de synchronisation lors du changement de direction des robots. Une fois que les robots ont formé la ligne, ils commencent à bouger de la même façon que pour le deuxième mouvement pour initier la communication. Après avoir effectué l'oscillation trois fois, soit après 150 pas, ou 15 secondes, le robot avec l'étiquette 1 se déplace dans l'axe Y au lieu de l'axe X pendant 25 pas, le même temps que les autres robots se déplacent avant de changer de direction. Ensuite, tous les robots ayant un angle de 0 par rapport à l'initiateur dans le graphe, soit le robot avec l'étiquette 1 et les deux robots mis à l'écart, s'immobilisent pendant que les autres robots continuent leurs déplacements. Cette image illustre

donc la perte de synchronisation entre les robots et représente le bris de communication entre les robots.

CONCLUSION ET RECOMMANDATIONS

L'objectif principal de cette recherche est d'améliorer le contrôle des essaims en réduisant les efforts reliés à la compréhension de leurs états sur la charge cognitive de l'opérateur. Notre scénario est celui d'un opérateur explorant un endroit éloigné étant en contrôle d'un essaim de robots l'aident à effectuer son exploration. En utilisant la méthode du « Design Thinking », nous avons défini l'enjeu principal de la situation comme étant l'impact du contrôle d'essaim sur la charge cognitive de l'opérateur. Avec la complexité des comportements d'essaim auto-organisés et émergeant, les recherches antérieures nous montre qu'il est peu intuitif de contrôler un essaim de robots. Cette complexité est due à la difficulté de compréhension de l'état des robots. Nous avons donc proposé un système de retour d'information par le mouvement expressif d'essaim qui permet d'aider l'opérateur à mieux comprendre l'état de l'essaim et l'état de la mission. Plusieurs techniques peuvent être utilisées pour effectuer un retour d'information, mais la solution choisie, permettant d'être utilisée par tous les robots dans presque toutes les circonstances, est le retour d'information visuelle par mouvement expressif de l'essaim. Cette recherche se penche sur le développement de ces mouvements expressifs et leur perception par les humains ainsi que sur l'implémentation de ces mouvements sur les petits robots de table (Zoids) suivant un contrôle complètement décentralisé.

Suite à notre revue de la littérature, nous concluons que la vitesse, la trajectoire, la synchronisation, spatiale et temporelle, ainsi que la régularité sont toutes des caractéristiques du mouvement ayant un impact sur la perception de l'expressivité des robots. Ceci influence alors la lisibilité : la lisibilité des robots permet d'augmenter les chances que l'opérateur perçoive correctement le message venant d'un mouvement expressif d'essaim. De plus, les figures formées par les robots ont aussi un impact sur la perception des messages, car certaines formes peuvent être reconnues plus aisément par un large public, comme un X ou une flèche, tandis que certaines autres formes, comme une forme plus ronde ou avec plus d'angularité, génèrent différentes réponses émotionnelles chez les observateurs selon leur culture.

Suite à cette revue de littérature, et à l'aide d'un groupe d'experts, des mouvements expressifs d'essaim ont été conçus dans le but de créer un retour d'information pour l'opérateur. Nous avons identifié les messages importants à transmettre comme étant les suivants : initier la communication, terminer la communication, aucun problème à rapporter, besoin d'intervention de la part de l'opérateur, problème de batterie et bris de communication entre les robots de l'essaim. Neuf mouvements différents ont été conçus sous forme d'animations représentant ces six types de messages. Suite à une étude utilisateurs avec 98 participants, nous avons conclu que quatre des neuf animations sont perçues comme transmettant un message distinct étant ainsi possible de les inclure dans une étude comportant un scénario réel avec de vrais robots.

Le but était donc d'implémenter ces mouvements expressifs sur un essaim de robots pour être en mesure d'effectuer une étude utilisateurs dans un contexte réel. Nous avons donc apporté des modifications aux petits robots de tables, les Zooids, permettant d'améliorer leurs déplacements pour plus facilement représenter les animations conçues. Avec de moteurs plus lents et un modèle plus gros ainsi qu'avec des modifications au contrôle en déplacement, les Zooids se déplacent plus lentement en ligne droite pour se rendre d'un point à un autre. Des modifications apportées à la séquence de localisation avec le projecteur permettent d'obtenir une carte dynamique sous les Zooids permettant ainsi de créer un scénario d'exploration réel où les robots découvrent la carte avec le temps. Finalement, un noeud de contrôle ROS permet d'effectuer les mouvements expressifs d'essaim de manière centralisée permettant ainsi de tester la perception des mouvements plus facilement.

Même si un contrôle centralisé est possible, utiliser des essaims de robots comporte plus d'avantages que les groupes de robots centralisés. Nous utilisons alors Buzz pour créer des scripts de comportements d'essaim pour effectuer la mission d'exploration. La principale contribution de cette section est la combinaison de deux algorithmes Buzz, l'exploration en chaîne et la formation par graphe acyclique, en un seul script ayant pour but d'inclure un retour

visuel pour l'opérateur pendant une mission d'exploration. Après avoir effectué les chaînes nécessaires pour maintenir la connectivité lors de l'exploration, un sous-essaim se détache de la formation en chaîne pour effectuer la formation par graphe acyclique et ainsi effectuer les mouvements expressifs d'essaim. Finalement, les Zooids peuvent représenter la position des drones utilisés en simulation et explorer une carte dynamique tout en incluant un retour visuel par mouvements expressifs d'essaim pour l'opérateur.

Suite au travail effectué dans cette recherche, la prochaine étape est de préparer une étude utilisateurs avec les Zooids. En utilisant la carte dynamique, un scénario d'exploration peut être créé pour que le participant soit immergé dans la mission pour représenter le mieux possible une exploration réelle. Utilisant les scripts Buzz, les robots peuvent effectuer l'exploration tout en incluant le retour visuel avec les mouvements expressifs d'essaim.

Cette configuration permet plusieurs libertés pour concevoir l'étude utilisateurs et permettrait de répondre à différentes questions de recherche. Une première étape serait de tester si les robots sont en mesure d'attirer l'attention de l'opérateur avec les mouvements expressifs conçus à cet effet. De plus, il serait intéressant de tester si les robots peuvent attirer l'attention de l'opérateur sans utiliser un mouvement expressif initial pour attirer l'attention, mais directement un message relatif à la mission. Ceci permettrait de déterminer s'il est requis d'utiliser un mouvement intermédiaire pour attirer l'attention de l'opérateur.

Ensuite, tous les mouvements expressifs implémentés dans le script Buzz peuvent être testés de la même manière que pour les animations, mais en utilisant les Zooids et le scénario d'exploration. Ceci permettrait de déterminer si l'étude avec les animations représente assez bien la réalité ou s'il y a des différences majeures de perception des mouvements entre les animations et les robots réels.

Finalement, en plus de valider la perception du bon message, la vitesse à laquelle l’interprétation est faite est importante et peut être testée avec cette configuration. Ceci permettrait de déterminer si l’opérateur doit passer un temps significatif à regarder le mouvement expressif pour obtenir de l’information ou si seulement un regard rapide lui permet de déterminer l’état de l’essaim. Évidemment, ce temps aura un impact sur la performance de l’opérateur à compléter les autres tâches de sa mission.

Les directions de recherche mentionnées peuvent être accomplies avec les Zooids, mais Buzz et les mouvements expressifs peuvent aussi être déployés sur d’autres robots autant au sol que dans les airs. Une fois le travail effectué avec les Zooids, les mêmes scripts pourraient être déployés sur des robots mobiles à plus grande échelle directement sur le terrain. On peut alors penser à des drones ou des véhicules à roues sur le terrain lors d’un scénario d’exploration où un opérateur doit directement contrôler les robots. Ce déploiement réel permettrait de vérifier la robustesse des scripts ainsi que l’efficacité des mouvements expressifs comme retour d’information visuel. La charge cognitive de l’opérateur pourrait ainsi être analysée avec et sans le retour d’information pour vérifier l’impact sur l’opérateur. Bref, ceci mènera à l’utilisation de cette stratégie dans une application réelle d’exploration.

6.1 Contributions dans la littérature

Le fruit de cette recherche est l’objet de différentes contributions dans la littérature.

D’abord, les mouvements expressifs ont été implémentés de manière centralisée sur différents groupes de robots soit les Dingos de Clearpath, les Crazyflies de Bitcraze ainsi que les Zooids. Ce travail a été soumis sous forme de vidéo pour l’atelier « Robot Swarms in the Real World » pour ICRA 2021 :

Boucher, C., Gomes Braga, R., Salas Gordoniz, J. E., Urrestilla Anguiozar, N., Montazer Zohour, H., Duval, A. & St-Onge D. (2021, 2 août). *Conveying information for groups of robots through*

expressive motion. Vidéo présentée à ICRA 2021. https://www.youtube.com/watch?v=OMX_mlfYm98&t=129s.

Un article pour une édition spéciale du journal « Autonomous Robot » nommée « Robot Swarms in the Real World : From Design to Deployment » a été rédigé représentant la majeure partie de ce travail dont l'architecture de contrôle décentralisé ainsi que l'étude utilisateur sur les mouvements expressifs. Cet article est soumis et en révision :

Boucher, C., Stower, R., Zibetti, E., Levillain, F. & St-Onge, D. (2022). *Motion-based communication for robotic swarms in exploration missions. Autonomous Robot, Robot Swarms in the Real World : From Design to Deployment.* Manuscrit soumis pour publication.

De plus, un article pour la conférence RO-MAN a été rédigé utilisant une analyse complémentaire de l'étude utilisateur sur les mouvements expressifs :

Stower, R., Zibetti, E., St-Onge, D. (2022). "Bots of a Feather : Exploring User Perceptions of Group Cohesiveness for Application in Robotic Swarms". In Proceedings of the 31st IEEE International Conference on Robot and Human Interactive Communication (RO-MAN 2022)

Finalement, le travail a été présenté dans le cadre du colloque REPARTI de l'ACFAS le 13 mai 2022. Voici le lien vers la page du colloque REPARTI : <https://www.acfas.ca/evenements/congres/programme/89/200/205/c>

ANNEXE I

DÉTAIL DES GRAPHES

Tableau-A I-1 Graphe en cercle utilisé pour le premier mouvement pour initier la communication

| Étiquette | Prédécesseur | Angle (rad) | Distance (cm) |
|-----------|--------------|-------------|---------------|
| 0 | -1 | -1 | -1 |
| 1 | 0 | 0.5236 | 500 |
| 2 | 0 | 1.0472 | 866 |
| 3 | 0 | 1.5708 | 1000 |
| 4 | 0 | 2.0944 | 866 |
| 5 | 0 | 2.6180 | 500 |

Tableau-A I-2 Graphe en forme de ligne utilisé pour le deuxième mouvement pour initier la communication

| Étiquette | Prédécesseur | Angle (rad) | Distance (cm) |
|-----------|--------------|-------------|---------------|
| 0 | -1 | -1 | -1 |
| 1 | 0 | 0.0 | 500 |
| 2 | 0 | 0.0 | 1000 |
| 3 | 0 | 0.0 | 1500 |
| 4 | 0 | 3.1416 | 500 |
| 5 | 0 | 3.1416 | 1000 |

Tableau-A I-3 Graphe à allure aléatoire utilisé pour terminer la communication

| Étiquette | Prédécesseur | Angle (rad) | Distance (cm) |
|-----------|--------------|-------------|---------------|
| 0 | -1 | -1 | -1 |
| 1 | 0 | 0.0 | 100 |
| 2 | 0 | 0.8 | 100 |
| 3 | 0 | 1.8 | 100 |
| 4 | 0 | 3.2 | 100 |
| 5 | 0 | 3.8 | 100 |

Tableau-A I-4 Graphe en forme de cercle imparfait utilisé pour le mouvement pour informer l'opérateur que la mission requiert une intervention

| Étiquette | Prédécesseur | Angle (rad) | Distance (cm) |
|-----------|--------------|-------------|---------------|
| 0 | -1 | -1 | -1 |
| 1 | 0 | 0.0 | 1000 |
| 2 | 0 | 0.8 | 1000 |
| 3 | 0 | 1.8 | 1000 |
| 4 | 0 | 3.2 | 1000 |
| 5 | 0 | 3.8 | 1000 |

Tableau-A I-5 Graphe en ligne diagonale utilisé pour le mouvement pour informer l'opérateur qu'il y a un problème de batterie

| Étiquette | Prédécesseur | Angle (rad) | Distance (cm) |
|-----------|--------------|-------------|---------------|
| 0 | -1 | -1 | -1 |
| 1 | 0 | 0.7854 | 500 |
| 2 | 0 | 0.7854 | 1000 |
| 3 | 0 | 0.7854 | 1500 |
| 4 | 0 | 0.7854 | 2000 |
| 5 | 0 | 0.7854 | 2500 |

BIBLIOGRAPHIE

- Bender, R. & Lange, S. (2001). Adjusting for multiple testing—when and how ? *Journal of Clinical Epidemiology*, 54(4), 343–349.
- Brambilla, M., Ferrante, E., Birattari, M. & Dorigo, M. (2013). Swarm robotics : a review from the swarm engineering perspective. *Swarm Intelligence*, 7(1), 1–41.
- Brinck, I. & Balkenius, C. (2019). Recognition in Human-Robot Interaction : The Gateway to Engagement. *2019 Joint IEEE 9th International Conference on Development and Learning and Epigenetic Robotics (ICDL-EpiRob)*, pp. 31-36.
- Capelli, B. & Sabattini, L. (2020). Human-Friendly Multi-Robot Systems : Legibility Analysis. *Human-Friendly Robotics 2019*, (Springer Proceedings in Advanced Robotics), 15–28.
- Capelli, B., Secchi, C. & Sabattini, L. (2019a). Communication Through Motion : Legibility of Multi-Robot Systems. *2019 International Symposium on Multi-Robot and Multi-Agent Systems (MRS)*, pp. 126–132.
- Capelli, B., Villani, V., Secchi, C. & Sabattini, L. (2019b). Understanding Multi-Robot Systems : on the Concept of Legibility. *2019 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pp. 7355–7361.
- Chung, T. H. (2021). *OFFensive Swarm-Enabled Tactics (OFFSET) : Briefing for the Naval Counter-Improvised Threat Knowledge Network*.
- Dam, R. F. (2021). 5 Stages in the Design Thinking Process. Repéré le 2022-04-27 à <https://www.interaction-design.org/literature/article/5-stages-in-the-design-thinking-process>.
- Dietz, G., Jane, E., Washington, P., Kim, L. H. & Follmer, S. (2017). Human Perception of Swarm Robot Motion. *Proceedings of the 2017 CHI Conference Extended Abstracts on Human Factors in Computing Systems*, (CHI EA '17), 2520–2527.
- Dorigo, M., Theraulaz, G. & Trianni, V. (2021). Swarm Robotics : Past, Present, and Future. *Proceedings of the IEEE*, 109(7), 1152–1165.
- Firgelli Automations. (2022). Plastic Planetary Micro DC Motor - OD : 6mm L : 16.3/18.8/21mm. Repéré le 2022-05-04 à <https://www.firgelliauto.com/products/micro-motor>.
- Grispino, K., Lyons, D. & Nguyen, T.-H. D. (2020). Evaluating the Potential of Drone Swarms in Nonverbal HRI Communication. *2020 International Conference on Human-Machine Systems (ICHMS)*, pp. 1–6.

- Hocraffer, A. & Nam, C. S. (2017). A meta-analysis of human-system interfaces in unmanned aerial vehicle (UAV) swarm management. *Applied Ergonomics*, 58, 66–80.
- Jakobson, R. (1963). Essais de Linguistique Générale. *Les Etudes Philosophiques*, 18(4), 465–465.
- Kim, L., Drew, D., Domova, V. & Follmer, S. (2020). User-defined Swarm Robot Control. pp. 1–13.
- Kim, L. H. & Follmer, S. (2017). UbiSwarm : Ubiquitous Robotic Interfaces and Investigation of Abstract Motion as a Display. *Proceedings of the ACM on Interactive, Mobile, Wearable and Ubiquitous Technologies*, 1(3), 66 :1–66 :20.
- Kim, L. H. & Follmer, S. (2021). Generating Legible and Glanceable Swarm Robot Motion through Trajectory, Collective Behavior, and Pre-attentive Processing Features. *ACM Transactions on Human-Robot Interaction*, 10(3), 21 :1–21 :25.
- Le Goc, M., Kim, L. H., Parsaei, A., Fekete, J.-D., Dragicevic, P. & Follmer, S. (2016). Zooids : Building Blocks for Swarm User Interfaces. *Proceedings of the 29th Annual Symposium on User Interface Software and Technology*, (UIST '16), 97–109.
- Lee, A., Jo, W., Kannan, S. S. & Min, B.-C. (2021). Investigating the Effect of Deictic Movements of a Multi-Robot. *International Journal of Human–Computer Interaction*, 37(3), 197–210.
- Levillain, F., St-Onge, D., Zibetti, E. & Beltrame, G. (2018). More Than the Sum of its Parts : Assessing the Coherence and Expressivity of a Robotic Swarm. *2018 27th IEEE International Symposium on Robot and Human Interactive Communication (RO-MAN)*, pp. 583–588.
- Levillain, F., St-Onge, D., Beltrame, G. & Zibetti, E. (2019). Towards situational awareness from robotic group motion. *2019 28th IEEE International Conference on Robot and Human Interactive Communication (RO-MAN)*, pp. 1–6.
- Li, G., St-Onge, D., Pincioli, C., Gasparri, A., Garone, E. & Beltrame, G. (2019). Decentralized progressive shape formation with robot swarms. *Autonomous Robots*, 43(6), 1505–1521.
- May, A. D., Dondrup, C. & Hanheide, M. (2015). Show me your moves ! Conveying navigation intention of a mobile robot to humans. *2015 European Conference on Mobile Robots (ECMR)*, pp. 1–6.

- McGuire, K. N., de Wagter, C., Tuyls, K., Kappen, H. J. & de Croon, G. C. (2019). Minimal navigation solution for a swarm of tiny flying robots to explore an unknown environment. *Science Robotics*, 4(35).
- Memar, A. H. & Esfahani, E. T. (2018). Physiological Measures for Human Performance Analysis in Human-Robot Teamwork : Case of Tele-Exploration. *IEEE Access*, 6, 3694–3705. Conference Name : IEEE Access.
- Menda, J., Hing, J., Ayaz, H., Shewokis, P., Izzetoglu, K., Onaral, B. & Oh, P. (2011). Optical Brain Imaging to Enhance UAV Operator Training, Evaluation, and Interface Development. *Journal of Intelligent and Robotic Systems*, 61, 423-443.
- Patel, J. & Pincioli, C. (2020). Improving Human Performance Using Mixed Granularity of Control in Multi-Human Multi-Robot Interaction. *2020 29th IEEE International Conference on Robot and Human Interactive Communication (RO-MAN)*, pp. 1135–1142.
- Pincioli, C. & Beltrame, G. (2016a). Swarm-Oriented Programming of Distributed Robot Networks. *Computer*, 49(12), 32-41.
- Pincioli, C. & Beltrame, G. (2016b). Buzz : An extensible programming language for heterogeneous swarm robotics. *2016 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pp. 3794–3800.
- Pincioli, C., Lee-Brown, A. & Beltrame, G. (2016). A Tuple Space for Data Sharing in Robot Swarms. 2.
- Podevijn, G., O’Grady, R., Carole, F. & Dorigo, M. (2016). Investigating the effect of the reality gap on the human psychophysiological state in the context of human-swarm interaction. *PeerJ Computer Science*, 2.
- Reig, S., Carter, E. J., Fong, T., Forlizzi, J. & Steinfeld, A. (2021). Flailing, Hailing, Prevailing : Perceptions of Multi-Robot Failure Recovery Strategies. *Proceedings of the 2021 ACM/IEEE International Conference on Human-Robot Interaction*, (HRI ’21), 158–167.
- Sandry, E., Gomez, R. & Nakamura, K. (2021). Art, Design and Communication Theory in Creating the Communicative Social Robot ‘Haru’. *Frontiers in Robotics and AI*, 8. doi : 10.3389/frobt.2021.577107.
- Santos, M. & Egerstedt, M. (2021). From Motions to Emotions : Can the Fundamental Emotions be Expressed in a Robot Swarm ? *International Journal of Social Robotics*, 13(4), 751–764.

- Scheel, A. M., Tiokhin, L., Isager, P. M. & Lakens, D. (2021). Why Hypothesis Testers Should Spend Less Time Testing Hypotheses. *Perspectives on Psychological Science*, 16(4), 744–755.
- Schranz, M., Umlauft, M., Sende, M. & Elmenreich, W. (2020). Swarm Robotic Behaviors and Current Applications. *Frontiers in Robotics and AI*, 7. Publisher : Frontiers.
- ShapeLab. (2019). SwarmUI. Repéré à <https://github.com/ShapeLab/SwarmUI>.
- St-Onge, D., Cote Allard, U., Glette, K., Gosselin, B. & Beltrame, G. (2019a). Engaging with Robotic Swarms : Commands from Expressive Motion. *ACM Transactions on Human-Robot Interaction*, 8(2), 11 (26 pp.). Place : USA Publisher : ACM.
- St-Onge, D., Levillain, F., Zibetti, E. & Beltrame, G. (2019b). Collective expression : how robotic swarms convey information with group motion. *Paladyn, Journal of Behavioral Robotics*, 10, 418–435.
- St-Onge, D., Kaufmann, M., Panerati, J., Ramtoula, B., Cao, Y., Coffey, E. B. & Beltrame, G. (2020a). Planetary Exploration With Robot Teams : Implementing Higher Autonomy With Swarm Intelligence. *IEEE Robotics Automation Magazine*, 27(2), 159-168. doi : 10.1109/MRA.2019.2940413.
- St-Onge, D., Varadharajan, V. S., Švogor, I. & Beltrame, G. (2020b). From Design to Deployment : Decentralized Coordination of Heterogeneous Robotic Teams. *Frontiers in Robotics and AI*, 7, 51.
- Stower, R., Zibetti, E. & St-Onge, D. (2022). Bots of a Feather : Exploring User Perceptions of Group Cohesiveness for Application in Robotic Swarms. In *Proceedings of the 31st IEEE International Conference on Robot and Human Interactive Communication (RO-MAN 2022)*.
- Takayama, L., Dooley, D. & Ju, W. (2011). Expressing thought : Improving robot readability with animation principles. *2011 6th ACM/IEEE International Conference on Human-Robot Interaction (HRI)*, pp. 69–76.
- Texas Instruments. (2017). *TI DLP LightCrafter 4500 Evaluation Module User's Guide* (Rapport n°DLP011F). Repéré à <https://www.ti.com/lit/ug/dlp011f/dlp011f.pdf>.
- Theraulaz, G. (2014). Embracing the Creativity of Stigmergy in Social Insects. *Architectural Design*, 84(5), 54–59.

- Varadharajan, V. S., St-Onge, D., Adams, B. & Beltrame, G. (2020). Swarm Relays : Distributed Self-Healing Ground-and-Air Connectivity Chains. *IEEE Robotics and Automation Letters*, 5(4), 5347–5354. Conference Name : IEEE Robotics and Automation Letters.
- Walker, M., Hedayati, H., Lee, J. & Szafir, D. (2018). Communicating Robot Motion Intent with Augmented Reality. *2018 13th ACM/IEEE International Conference on Human-Robot Interaction (HRI)*, pp. 316–324.
- Walker, P. M. (2017). *Improving Operator Recognition and Prediction of Emergent Swarm Behaviors*. (Ph.D., University of Pittsburgh, United States – Pennsylvania).
- Zheng, Y. & Han, Z. (2020). Experimental Implementation of Collective Motion based on Swarm Robotic Control. *2020 15th IEEE Conference on Industrial Electronics and Applications (ICIEA)*, pp. 1526–1531.