
Remerciements

Je tiens à exprimer toute ma reconnaissance à mon Directeur de mémoire professeur Cherif Foudil . Je la remercie de m'avoir encadré, orienté, aidé et conseillé.

J'adresse mes sincères remerciements à tous les professeurs, intervenants et toutes les personnes qui par leurs paroles, leurs écrits, leurs conseils et leurs critiques ont guidé mes réflexions et ont accepté à me rencontrer et répondre à mes questions durant mes recherches.

Je remercie M. Khalid pour se réponse à mes questions au cours de mes recherches.

Je remercie mes très chers parents qui ont toujours été là pour moi,« Vous avez tout sacrifié pour vos enfants n'épargnant ni santé ni efforts. Vous m'avez donné un magnifique modèle de labeur et de persévérance. Je suis redevable d'une éducation dont je suis fier ».

Je remercie ma grand famille pour leur encouragement.

Je tiens à remercier tous mes Ami(e)s , pour leur amitié, leur soutien inconditionnel et leur encouragement et confiance.

À tous ces intervenants, je présente mes remerciements, mon respect et ma gratitude.

résumé

La robotique en essaim est un domaine émergent de la robotique, qui s'intéresse aux systèmes multi-robots auto-organisés, souvent caractérisés par de grands nombres d'unités. Actuellement, un nombre d'applications ont été établi pour répondre au besoin croissant des robots d'essaims dans notre vie quotidienne. On cite par exemple des tâches dans des activités récréatives et artistiques telles que la peinture, la musique ou même la danse.

On va présenter dans ce mémoire une étude typique des systèmes multi-robots, à savoir la formation de patterns. Ces derniers, sont inspirés des phénomènes importants trouvés dans les organismes vivants (tels que les animaux et les plantes), constituent un défi visant à résoudre le problème de l'organisation d'un groupe de robots en formation des patterns complets avec des formes simples telles que les hexagones, les lignes, ou la distribution uniforme.

On a utilisé la conception et à la synthèse de contrôleurs pour des systèmes des robots d'essaim afin d'obtenir des formes de manière auto-organisée en peu de temps avec différents capteurs. Notre approche est inspirée de la nature, en particulier des forces bio-mécaniques impliquées dans les études des cellules internes, afin de produire efficacement des formes d'agrégations auto-organisées avec des robots. Nous avons utilisé une simulation expérimentale basée sur le simulateur ARGoS (Autonomous Robots Go Swarming). Ce simulateur nous a permis d'étudier des aspects du comportement collectifs auto-organisés.

ملخص

روبوتات السرب هي مجال ناشئ من الروبوتات ، والذي يركز على أنظمة متعددة من الروبوتات المنظمة ذاتيا ، وغالبا ما تتميز أعداد كبيرة من الوحدات حاليا ، تم إنشاء عدد من التطبيقات لتلبية الحاجة المتزايدة للروبوتات سرب في حياتنا اليومية. وتشمل الأمثلة المواقع في الأنشطة الترفيهية والفنية مثل الرسم أو الموسيقى أو حتى الرقص.

سنقدم في هذه الورقة دراسة نموذجية للأنظمة متعددة الروبوتات ، وهي تشكيل الأنماط ، المستوحاة من الظواهر المهمة الموجودة في الكائنات الحية (مثل الحيوانات والنباتات) تشكل تحديا لحل مشكلة تنظيم مجموعة من الروبوتات في تشكيل أنماط كاملة بأشكال بسيطة مثل السداسي أو الخطوط أو التوزيع الموحد.

تم استخدام تصميم وإعداد وحدات التحكم لأنظمة روبوت سرب للحصول على الأشكال بطريقة منظمة في وقت قصير مع أجهزة استشعار مختلفة. نهجنا مستوحاة من الطبيعة، ولا سيما القوى الميكانيكية الحيوية المشاركة في الدراسات الخلوية الداخلية ، من أجل إنتاج أشكال منظمة ذاتيا من التجميع باستخدام الروبوتات. استخدمنا محاكاة تجريبية بناءً على محاكي أرغوس . حيث يسمح لنا بدراسة جوانب السلوك الجماعي المنظم .

TABLE DES MATIÈRES

Introduction générale

Introduction générale.....	9
----------------------------	---

Chapitre 1

L'intelligence en essaim

1.1 Introduction.....	13
1.2 Vue d'ensemble des robots en essaim.....	13
1.3 Motivation et inspiration sociales des insectes.....	13
1.4 Définition et Propriétés.....	15
1.4.1 Définition.....	15
1.4.2 Propriétés.....	15
1.5 Algorithmes basés sur le comportement des essaims naturels.....	16
1.5.1 L'optimisation par essaims particulaires (OEP ou PSO).....	16
1.5.2 L'optimisation par colonies de fourmis.....	17
1.6 Robotique en essaim.....	17
1.7 Définition.....	18
1.8 Champs d'application de la robotique en essaim.....	18
1.8.1 Les tâches couvrent une grande surface.....	18
1.8.2 Tâches dangereuses pour le robot.....	19
1.8.3 Les tâches nécessitent une population d'échelle.....	19
1.8.4 Les tâches nécessitent une redondance.....	19
1.9 Les problèmes de robotique en essaim.....	19
1.10 Plateformes de simulation robotique en essaim.....	20

1.11. Conclusion.....	22
-----------------------	----

Chapitre 2

Patterns d'agrégation dans les systèmes robotique en essaim

2.1 Introduction.....	24
2.2 Agrégation dans la nature.....	24
2.3 Émergence de pattern d'agrégation d'essaims naturels	24
2.4 Étude de cas: patterns de flochage.....	25
2.4.1 Modèles sur les motifs de flochage.....	25
2.5 Patterns d'agrégation dans la robotique en essaim.....	26
2.5.1 Agrégation dans la robotique en essaim.....	26
2.5.2 Émergence de patterns dans l'essaim de robotique agrégée.....	27
2.6 Approches sur les schémas d'agrégation en robotique en essaim.....	27
2.6.1 Méthodes à base de signal	27
2.6.2 Méthodes auto-organisées.....	25
2.7 Conclusion.....	30

Chapitre 3

La Plateforme ARGoS

3.1 Introduction.....	32
3.2 Simulation Platform (ARGoS).....	32
3.3 Plusieurs moteurs physiques	32
3.3.1 L'architecture d'ARGoS.....	32
3.3.2 Plateforme robotique.....	34

3.4 Système de détection / activation.....	35
3.4.1 Capteur de distance et de portée infrarouge, (range and bearing).....	35
3.4.2 Capteurs de proximité infrarouges.....	36
3.4.3 caméra omnidirectionnelle.....	38
3.4.4 Actionneur de roues.....	38
3.5 Les méthodes.....	40
3.5.1 Le modèle viscoélastique: un modèle inspiré par la biomécanique.....	40
Vue d'ensemble de la cellule interne Bio-mécanique.....	40
3.6 Conclusion.....	42

Chapitre 4

Le modèle viscoélastique

4.1 Introduction.....	44
4.2 Le modèle d'interaction viscoélastique des robots en essaim.....	44
4.3 Les méthodes	45
4.3.1. Contrôle de proximité (PC).....	48
4.3.2 Contrôle répulsif (RC).....	49
4.3.3 Contrôle du mouvement angulaire dépendant de l'avant (FDAMC).....	51
4.4 Conclusion.....	52

Chapitre 5

Résultats

5.1 Résultats.....	54
5.A résultat avec le capteur « range and bearing » (RAB).....	55

5.B résultat avec le capteur omnidirectionnel_camera.....	59
5.2 Tests.....	63
5.3 Conclusion.....	67
Conclusion générale.....	69
Bibliographie.....	71

Liste des figures

Figure 1.1 : Exemples de systèmes d’essaims naturels: (A) Des milliers d’ étourneaux volant ensemble et formant un grand troupeau. (B) Les fourmis tirent le bonbon pour le stocker (C) Essaimage d'abeilles à la recherche d'un nouveau site de nidification.....	14
Figure 2.2: Les trois comportements de base déterminant le mouvement des boids. (A) Séparation : réagit uniquement aux membres du groupe voisins. (B) Alignement: direction vers la direction moyenne des membres du groupe. (C) Cohésion: se rapprocher de la position moyenne des boids voisins.....	26
Figure 3.3 : Description schématique de l'architecture ARGoS.....	33
Figure 3.4: Principaux dispositifs composant le robot ARGoS "Le foot-bot"	35
Figure 3.5: Les mesures de position relative dans le système de communication.....	36
Figure 3.6: Position des capteurs de proximité infrarouge dans le robot foot-bot	37
Figure 3.7: configuration des capteurs de proximité IR en XML.....	37
Figure 3.8: Cinématique d’un robot mobile à entraînement différentiel à deux roues.....	38
Figure 3.9: Le modèle viscoélastique sous-cellulaire de Jamali, Azimi et Mofrad.....	42
Figure 4.10: Le modèle d'interaction viscoélastique des robots en essaim.....	45
Figure 4.11: Le modèle de contrôle global.....	46
Figure 4.12: Un modèle de la configuration de connexion viscoélastique entre quatre robots.....	49
Figure 5.13 : Exemple de distribution aléatoire des robots dans l'arène	54
Figure 5.14 3 : robot formant un triangle équilatéral.....	55
Figure 5.15 4 robot formant un carré.....	55
Figure 5.16 : 5 robot formant un pentagone régulier	56
Figure 5.17 : 6 robot formant un hexagone régulier.....	56
Figure 5.18: 7 robot formant un heptagone régulier.....	57

Figure 5.19 : 8 robot formant un octogone régulier.....	57
Figure 5.20: 9 robot formant un octogone régulier avec un robot au centre.....	58
Figure 5.21 : 10 robot formant un nonagone régulier avec un robot au centre.....	58
Figure 5.22 : 4 forme au même espace (4 robot , 6robot, 7 robot 10 robot) avecRAB....	59
Figure 5.23 : 3 robot formant un triangle équilatéral.....	59
Figure 5.24 : 4 robot formant un carré.....	60
Figure 5.25 : 5 robot formant un pentagone régulier.....	60
Figure 5.26 : 6 robot formant un hexagone régulier.	60
Figure 5.27 : 7 robot formant un heptagone régulier.....	61
Figure 5.28 : 8 robot formant un octogone régulier.....	61
Figure 5.29 : 9 robot formant un octogone régulier avec un robot au centre.....	61
Figure 5.30 : 10 robot formant un nonagone régulier avec un robot au centre.....	62
Figure 5.31: plusieurs forme au même espace (4 robot , 6robot, 7 robot 10 robot).....	62
Figure 5.32: Taux de changement de $v_i(t)$ pour les 6 foot-bot (rb0, rb1, rb2, rb3, rb4 et rb5). axe des x des graphes représente l'évolution du pas de temps t dans la camera omnidirectionnel	63
Figure 5.33: Taux de changement de $v_i(t)$ pour les 6 foot-bot (rb0, rb1, rb2, rb3, rb4 et rb5). axe des x des graphes représente l'évolution du pas de temps t dans le capture range and bearing.....	64
Figure 5.34: Taux de changement de $v_i(t)$ pour les foot-bot de 10 foot-bot . axe des x des graphes représente l'évolution du pas de temps t dans la camera omnidirectionnel	65
Figure 5.35: Taux de changement de $v_i(t)$ pour les 10 foot-bot . axe des x des graphes représente l'évolution du pas de temps t dans le capture range and bearing.....	66
Figure 5.36: temps de stabilité de chaque forme différent avec RAB.....	66
Liste des tableaux	
Tableau1.1 : Problèmes de la robotique en essaim.....	20
Tableau4.2: Paramètres du modèle de formation de forme régulière.....	47

Introduction générale

Depuis longtemps, remontant au milieu du vingtième siècle, les robots ont été effectivement impliqués dans de nombreux secteurs industriels. L'utilisation précoce de robots industriels visait en définitive à automatiser la plupart des tâches purement répétitives et mécaniques souvent effectuées par des humains. Cela pourrait être avantageux à plusieurs égards, en offrant de meilleurs degrés de cohérence et des vitesses de production beaucoup plus élevées à des degrés de précision importants. Aujourd'hui, la plupart des robots utilisés dans de telles tâches industrielles sont conçus pour effectuer des opérations bien définies à un moment donné et à un emplacement fixe, sans aucune flexibilité dans leur reconfigurabilité morphologique. En conséquence, cette catégorie de robots n'est utilisée que dans des environnements connus et se limite à des tâches relativement simples.

D'autre part, d'autres tâches doivent être effectuées dans des environnements non structurés, susceptibles de changer avec le temps et nécessitant un degré d'autonomie et de flexibilité que l'on ne retrouve pas dans les robots industriels. L'exploration et la couverture de vastes zones, la coopération en matière de transport et d'agrégation, la surveillance de l'environnement, les missions de recherche et de sauvetage, la recherche de nourriture et l'exploration de l'espace sont des exemples. Cependant, avec l'apparition de nouveaux défis tels que la décentralisation du contrôle et de l'auto-organisation, ainsi que certaines tâches nécessitant la réalisation efficace de grands groupes de robots autonomes. Un intérêt important pour l'application de techniques de renseignement en essaim dans des systèmes multi-robotiques a donné lieu à un nouveau domaine de recherche appelé robotique en essaim.[Ying Tan et al 2013]

L'idée principale derrière la robotique en essaim consiste à étudier comment coordonner de grands groupes de robots relativement simples grâce à l'utilisation de règles locales. Il se concentre sur l'étude de la conception d'un grand nombre de robots relativement simples, de leur corps physique et de leurs comportements de contrôle afin de réaliser une tâche spécifique qui dépasse les capacités d'un seul robot. La robotique en essaim est étroitement liée à l'idée d'intelligence en essaim et partage son intérêt pour les systèmes décentralisés auto-organisés. Il offre plusieurs avantages pour les applications robotiques telles que l'évolutivité, la flexibilité et la robustesse dues à la redondance. [Iñaki Navarro et al ,2012] Récemment, avec les énormes progrès réalisés dans ce domaine, les chercheurs se sont principalement concentrés sur la manière dont un système de robot essaim peut être impliqué dans notre vie réelle. Aujourd'hui, ils peuvent participer efficacement aux activités militaires (détection de bombes collectives, recherche et exploration en coopération), logistique (gestion d'entrepôts et livraison de produits aux clients, par exemple), agriculture (ensemencement, récolte et stockage de céréales) et d'urgence (par exemple, sauvetage en cas de catastrophe), etc...

L'objectif de notre travail est de proposer une solution simple à la tâche de contrôle de la formation pour un système robotique en essaim. En utilisant un modèle d'interaction

viscoélastique virtuelle pour réaliser des formations géométriques de base telles que des triangles, des carrés, des pentagones et surtout, en prenant en compte le temps d'exécution de la tâche en utilisant deux capteurs différents.

Le mémoire est organisé en 5 chapitres regroupés en deux parties principales : La première partie (chapitre 1 et chapitre 2,3), il place le lecteur dans le contexte du travail et lui présente progressivement l'état de l'art et les études connexes de la recherche présentée dans ce mémoire, dans la seconde partie (chapitre 4 et chapitre 5) de cette partie, nous avons présenté notre approche validée par quelques résultats de simulation.

Au chapitre 1, un aperçu détaillé du domaine de la robotique en essaim est présenté de manière chronologique. Le chapitre commence par présenter le domaine de l'intelligence des essaims et ses principaux concepts fondamentaux. Il introduit ensuite le domaine de la robotique en essaim en tant que sous-domaine spécifique à la multi-robotique dans lequel la théorie de l'intelligence en essaim est appliquée, en précisant les principales différences entre les deux domaines de recherche (biologie, robotique).

Le chapitre 2 explore les principaux problèmes abordés dans ce mémoire et propose des études pertinentes sur les modèles d'agrégation. Plus précisément, ce chapitre présente des études sur l'agrégation de motifs au sein d'essaims naturels, puis passe en revue la manière dont ces études ont été utilisées pour traiter le problème de la formation de motifs à l'aide de systèmes d'essaims robotisés. Le chapitre traite principalement de deux types d'études sur l'agrégation de motifs: (1) une méthode basée sur les repères et (2) une méthode basée sur l'auto-organisation, en mettant davantage l'accent sur la seconde méthode.

Le chapitre 3 présente les matériaux et les méthodes utilisés pour synthétiser des contrôleurs afin d'agréger des études de modèles au sein de l'essaim de robots. Il présente spécifiquement le simulateur et son architecture ainsi que la plateforme robotique adoptée dans nos études. De plus, il aborde tous les matériaux requis en termes de systèmes de détection et d'activation embarqués de robots, particulièrement utilisés pour mettre en œuvre les contrôleurs à agrégation proposés. En outre, il présente notre modèle d'interaction viscoélastique de la robotique en essaim en tant que modèle basé sur la biomécanique qui s'inspire de la biomécanique impliquée dans les cellules internes.

Nous présentons dans les chapitres 4 et 5 une solution simple à la tâche de contrôle de la formation pour un système robotique en essaim. Le chapitre 4 décrit d'abord la tâche à exécuter par les robots, et la configuration expérimentale définie pour cette tâche. Il présente ensuite et explique en détail le contrôleur de robot global utilisé pour effectuer la tâche en utilisant un modèle d'interaction viscoélastique virtuelle. Nous réalisons des formations géométriques régulières à l'aide de robots foot-bots, avec deux types de capteurs pour étudier la qualité de performance de chaque capteur. Dans le chapitre 5, nous illustrons la mise en œuvre du contrôleur global proposé de robot foot-bot et nous évaluons les performances de la solution sous différentes métriques de performance.

Le mémoire est clôturé par une conclusion générale et des perspectives envisagées.

Chapitre I
L'intelligence en essaim

Chapitre 1

L'intelligence en essaim

1.1 Introduction

En informatique et intelligence artificielle (IA), Intelligence en essaim (IS) est un domaine considéré comme un concept important aux propriétés émergentes, qui conçoit et étudie des méthodes de calcul efficaces pour la résolution des problèmes complexes.

Il s'inspire notamment du comportement des systèmes biologiques tels que de vrais essaims ou des colonies d'insectes. Intelligence en essaim est le comportement collectif de nature décentralisée de systèmes auto-organisés. C'est naturel ou artificiel. Le concept SI est appliqué dans de nombreux travaux en intelligence artificielle.

[Gerardo Beni et al,1993] nt introduit ce concept en 1989, dans le contexte des systèmes robotiques cellulaires. Ils sont constitués d'une population d'agents simples ou de boids qui suivent des règles très simples en interagissant localement entre eux et avec leur environnement. En l'absence de structure de contrôle centralisée dictant le comportement des agents individuels, ceux-ci fonctionnent de manière aléatoire et les interactions entre ces agents conduisent à l'émergence d'un comportement global "intelligent", inconnu des agents individuels. Les colonies de fourmis, l'assèchement des oiseaux, l'élevage, la croissance bactérienne et l'élevage des poissons sont des exemples naturels d'Intelligence en essaim.

1.2 Vue d'ensemble des robots en essaim

C'est un domaine de recherche émergent dans lequel la théorie de l'intelligence en essaim est appliquée à des systèmes multi-robotiques, la robotique en essaim est un sous-domaine très particulier de la robotique collective qui étudie comment coordonner de grands groupes de robots relativement simples, en utilisant des règles locales. Il se concentre sur l'étude de la conception d'une grande quantité de robots relativement simples, leur corps physiques et leurs comportements de contrôle. Depuis son introduction en 2000, plusieurs expériences réussies ont été réalisées, et jusqu'à maintenant plus de projets sont à l'étude.

1.3 Motivation et inspiration sociales des insectes

Les comportements collectifs des insectes sociaux, tels que la danse de l'abeille mellifère, la construction du nid de guêpe, la construction du sentier qui suit les fourmis, ont longtemps été considérés comme des aspects étranges et mystérieux de la biologie. Les chercheurs ont démontré que les individus n'ont pas besoin de représentation ni de connaissances sophistiquées pour produire des comportements aussi complexes.

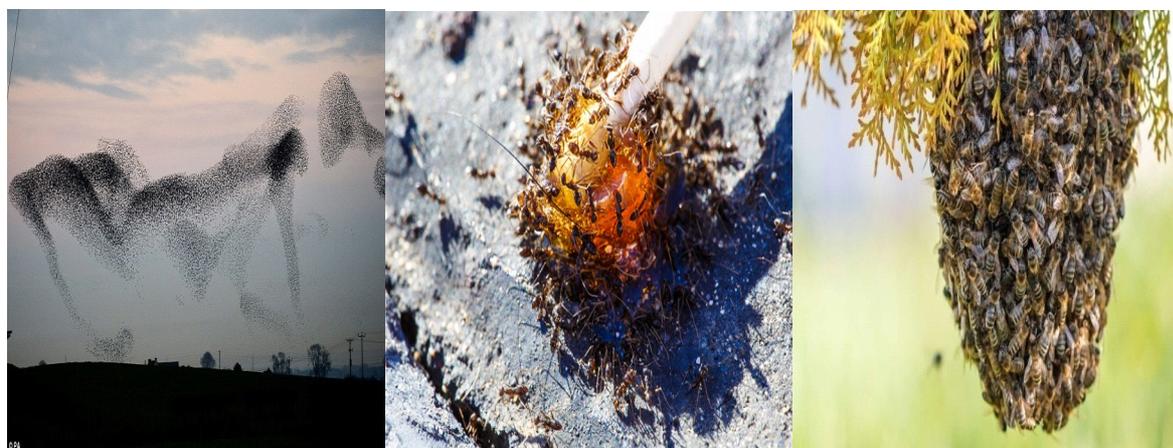
Chez les insectes sociaux, les individus sont capables d'échanger des informations sans aucune aide, ni un chef pour les guider; par exemple, de communiquer l'emplacement d'une source de nourriture, une zone de recherche de nourriture favorable ou la présence d'un

danger pour leurs compagnons. Cette interaction entre les individus est basée sur le concept de localité, où il n'y a aucune connaissance de la situation globale.

La communication implicite résultant des modifications apportées à l'environnement est appelée stigmergie. Les insectes modifient leurs comportements en raison des modifications antérieures apportées par leurs partenaires dans l'environnement. Cela se voit dans la construction de nids de termites, où les changements de comportement des travailleurs sont déterminés par la structure du nid.

L'organisation émerge des interactions entre les individus et entre les individus et l'environnement. Ces interactions se propagent dans toute la colonie et, par conséquent, la colonie peut résoudre des tâches qui ne pourraient pas être résolues par un seul individu. Ces comportements collectifs sont définis comme des comportements auto-organiseurs. Les théories sur l'auto-organisation, empruntées aux domaines de la physique et de la chimie, peuvent être utilisées pour expliquer comment les insectes sociaux manifestent un comportement collectif complexe résultant d'interactions entre individus.

L'auto-organisation repose sur la combinaison des quatre règles de base suivantes : rétroaction positive, rétroaction négative, caractère aléatoire et interactions multiples. [Iñaki Navarro et al 2013(2012)]



A

B

C

FIGURE 1.1 : Exemples de systèmes d'essaims naturels: (A) Des milliers d' étourneaux volant ensemble et formant un grand troupeau. (B) Les fourmis tirent le bonbon pour le stocker (C) Essaimage d'abeilles à la recherche d'un nouveau site de nidification

1.4 Définition et Propriétés

Les systèmes d'essaims où de nombreuses caractéristiques communes sont partagés entre eux. L'idée la plus commune réside dans le degré élevé de coordination affiché lorsque des groupes d'essaims individuels exercent leurs activités. Ce qui est fascinant, c'est qu'aucune de ces entités individuelles n'a pu saisir l'ensemble, mais chacune d'elles contribue au succès du groupe. Les groupes se comportent également comme une seule entité, mais le comportement collectif qui en résulte est le résultat de nombreuses actions individuelles exécutées simultanément. Les entités individuelles interagissent localement entre elles et leur environnement en suivant un ensemble de règles simples, même leur compréhension de l'activité collective est faible ou inexistante .

Les groupes sont organisés de manière à ce qu'aucun chef ne contrôle l'ensemble des entités et qu'il n'existe pas de hiérarchie claire entre les individus. Ce type d'organisation qui semble fuir toute structure prédéfinie est appelé auto-organisation décentralisée.

1.4.1 Définition

Nous adoptons la définition qui est décrite par [Marco Dorigo, et al. 2007]

«La discipline qui traite des systèmes naturels et artificiels, composée de nombreux individus et coordonnée à l'aide d'un contrôle décentralisé et d'une auto-organisation. En particulier, la discipline est axée sur les comportements collectifs résultant des interactions locales des individus entre eux et avec leur environnement ».

Ainsi, un système de renseignement en essaim comprend généralement une population de simples agents n'interagissent que localement avec eux-mêmes et avec leur environnement, sans avoir une connaissance globale de leur propre état et de l'état du monde. De plus, le comportement général observé apparaît en réponse à l'environnement local et aux interactions locales entre les agents qui suivent des règles souvent très simples. Un système de renseignement essaim a une dimension fascinante qui réside dans sa capacité à agir de manière coordonnée, même en l'absence de coordinateur externe. Par conséquent, même si aucun individu n'est en charge d'un groupe, celui-ci affiche toujours un comportement globalement intelligent.

1.4.2 Propriétés :

Un système de renseignement essaim est caractérisé par une : [Hiroaki Kitano.,2004]

- **Robustesse** : Cette fonctionnalité est définie par Kitano comme une propriété permettant le système pour maintenir ses fonctions malgré les perturbations externes et internes. Cela signifie que le système devrait toujours fonctionner même si certaines personnes échouent.

- **Adaptative (flexibilité):** L'adaptabilité est un phénomène biologique de base, dans lequel un organisme devient mieux adapté à son habitat. Cela signifie que le système a la capacité de s'adapter à tout environnement en mutation.
- **Évolutivité:** Cela signifie que les hauts niveaux de fonctionnalité du système doivent être maintenus même si la taille des personnes augmente. Le comportement de l'ensemble des essaims ne devrait pas être perturbé par l'ajout d'un nouvel individu qui ne peut influencer que le comportement de quelques autres. Dans les systèmes artificiels, cela est extrêmement important car les performances d'un système évolutif peuvent être améliorées simplement en augmentant la taille sans avoir à reprogrammer.
- **Auto-organisation (contrôle décentralisé):** largement connu dans les systèmes biologiques tels que les cellules, les organismes et les groupes qui possèdent un grand nombre de sous-unités. Les sous-unités individuelles fonctionnent en tant que groupe sans supervision locale ni contrôle central.
- **Fonctionnalité parallèle :** Ceci est possible dans un système essaim car différentes opérations peuvent être effectuées simultanément par différentes entités. Cela aide à assouplir un système d'intelligence d'essaim artificiel et lui permet de s'auto-organiser puissamment et d'exécuter différents aspects d'une tâche extrêmement complexe.

1.5 Algorithmes basés sur le comportement des essaims naturels.

Des théories basées sur des essaims naturels ont été appliquées pour résoudre des problèmes d'ingénierie analogues à plusieurs domaines tels que l'optimisation combinatoire, l'enracinement, les réseaux de communication, les applications robotiques, etc...

1.5.1 L'optimisation par essaims particuliers (OEP ou PSO)

L'optimisation par essaims particuliers est une méthode « méta-heuristique » de type évolutionnaire qui utilise une « population » de solutions candidates pour déterminer une solution optimale du problème posé. Le degré d'optimalité est mesuré par une fonction fitness (aptitude) définie par l'utilisateur. L'OEP diffère des autres méthodes de calcul évolutionnaire par le fait que les membres de la population appelés « particules » sont dispersés dans l'espace du problème. Le comportement de l'essaim particulière doit donc être décrit en se plaçant du point de vue d'une particule. Chacune des particules est dotée : [Roger SERRA, et al ,2017]

- d'une position, c'est-à-dire ses coordonnées dans l'ensemble de définition.
- d'une vitesse qui permet à la particule de se déplacer. De cette façon, au cours des itérations, chaque particule change de position. Elle évolue en fonction de son meilleur voisin, de sa meilleure position, et de sa position précédente. C'est cette évolution qui permet de s'approcher d'une particule optimale.
- d'un voisinage, c'est-à-dire un ensemble de particules qui interagissent directement sur la particule, en particulier celle qui a le meilleur critère.

A tout instant t , chaque particule connaît donc :

- sa meilleure position visitée. On retient essentiellement la valeur du critère calculée ainsi que ses coordonnées.
- la position du meilleur voisin de l'essaim qui correspond à l'ordonnement optimal.
- la valeur qu'elle donne à la fonction objective (fitness) car, à chaque itération, il faut une comparaison entre la valeur du critère donné par la particule courante et la valeur optimale.

1.5.2 L'optimisation par colonies de fourmis

L'optimisation par colonies de fourmis (ACO, Ant colony optimization) est une méta heuristique à base d'agents appelés fourmis artificielles qui est utilisée pour trouver des solutions à différents problèmes d'optimisation tel que le problème du voyageur de commerce. Ces Algorithmes sont inspirés du comportement des fourmis et constituent une famille distincte de méta heuristiques d'optimisation.

Le TSP (Traveling Salesman Problem) est le problème consistant à trouver un circuit hamiltonien de longueur minimale sur le graphe. Le voyageur de commerce est un problème NP-complet. La métaphore de la colonisation de fourmis s'y applique particulièrement bien, l'algorithme général est relativement simple, et repose sur un ensemble de fourmis, chacune parcourant un trajet parmi ceux possibles. À chaque étape, la fourmi choisit de passer d'une ville à une autre en fonction de quelques règles : [DJELLAT Soulimane 2013]

- elle ne peut visiter qu'une fois chaque ville.
- plus une ville est loin, moins elle a de chance d'être choisie (c'est la « visibilité »).
- plus l'intensité de la piste de phéromone disposée sur l'arête entre deux villes est grande, plus le trajet aura de chance d'être choisi.
- une fois son trajet terminé, la fourmi dépose, sur l'ensemble des arêtes parcourues, plus de phéromones si le trajet est court .
- les pistes de phéromones s'évaporent à chaque itération.

1.6 Robotique en essaim

La robotique fait partie des applications visées par l'intelligence en essaim; cette déclinaison de la robotique porte le nom de robotique en essaim. La robotique en essaim ajoute les contraintes du monde physique à celle de la théorisation des phénomènes en essaim. Sous un autre angle de vue, elle propose un terrain d'expérimentation et de validation stimulant pour les algorithmes de l'intelligence en essaim.

La robotique en essaim cherche en effet à se démarquer de la robotique distribuée ou collective par sa bio inspiration explicite. Une définition du domaine est donnée dans : [Rodolphe Charrier,2009]

1.7 Définition :

La robotique en essaim est l'étude de la façon dont un grand nombre d'agents relativement simples incarnés physiquement peuvent être conçus pour qu'un comportement voulu puisse émerger des interactions locales entre agents et entre les agents et l'environnement. La robotique en essaim considère des robots en général identiques, de conception simple qui ont des perceptions et interactions uniquement locales. Ces essaims de robots sont susceptibles d'offrir les caractéristiques suivantes qui rejoignent celles de la robotique collective : [Sahin et al., 2008]

- La robustesse : la redondance inhérente au système, la coordination décentralisée, la simplicité de conception de chaque entité et la distribution de la perception, font des essaims de robots des systèmes robustes aux pannes et aux variations de l'environnement,
- La flexibilité concerne la capacité des essaims de robots à traiter différents types de problèmes.
- Scalabilité : les performances ne sont globalement pas impactées par la taille de l'essaim.

1.8 Champs d'application de la robotique en essaim

Les applications potentielles de la robotique en essaim comprennent les tâches qui exigent la miniaturisation, telles que les tâches de détection distribuées dans des micro-machines ou le corps humain. D'autre part, la robotique en essaim peut être adaptée aux tâches qui requièrent des conceptions bon marché, telles que la tâche d'extraction ou la tâche de recherche de nourriture . La robotique en essaim peut également être impliquée dans des tâches qui nécessitent un coût en espace et en temps important, et sont dangereuses pour l'homme ou les robots eux-mêmes, telles que l'aide post-sinistre, la recherche de cible, les applications militaires, etc. [Y Tan 2013]

Plusieurs domaines d'applications potentiels pour les systèmes de robotique en essaim qui sont très appropriés sont décrits ci-dessous, classés principalement en quatre types: [Y Tan 2013]

1.8.1 Les tâches couvrent une grande surface

Le système de robotique en essaim est distribué et spécialisé pour les tâches nécessitant une grande surface d'espace, par exemple, les tâches couvrent de grandes surfaces. Un exemple simple serait la recherche et la collecte de plusieurs cibles dans une zone ouverte. L'essaim essaie de rechercher avec la coopération de groupe pour accélérer la recherche. La zone peut être très vaste et l'essaim peut tirer parti de la recherche parallèle avec plusieurs petits groupes situés dans les zones de détection des robots.

1.8.2 Tâches dangereuses pour le robot

Grâce à l'évolutivité et à la stabilité, l'essaim fournit une redondance pour traiter des tâches dangereuses. L'essaim peut subir une perte considérable de robots avant que le travail ne soit interrompu. Les robots sont très bon marché et sont préférés pour les zones qui endommagent probablement les travailleurs. Dans certaines tâches, les robots peuvent être irrécupérables et l'utilisation de robots complexes et coûteux est donc inacceptable sur le plan économique, tandis que la robotique en essaim avec des individus bon marché peut fournir des solutions raisonnables.

1.8.3 Les tâches nécessitent une population d'échelle

La charge de travail de certaines tâches peut changer avec le temps, et la taille de l'essaim doit être réduite en fonction de la charge de travail actuelle pour une efficacité élevée, à la fois en termes de temps et d'économie. Par exemple, pour éliminer les fuites d'huile après les accidents de la citerne, l'essaim devrait maintenir une population élevée lorsque l'huile fuit rapidement au début de la tâche et réduire progressivement le nombre de robots lorsque la source de la fuite est bouchée et que la zone de fuite est presque entièrement nettoyée.

1.8.4 Les tâches nécessitent une redondance

La robustesse des systèmes robotiques en essaim profite principalement de la redondance de l'essaim, c'est-à-dire que le retrait de certains robots n'a pas d'impact significatif sur les performances. Certaines tâches se concentrent sur le résultat plutôt que sur le processus, c'est-à-dire que le système doit s'assurer que la tâche sera complétée avec succès, principalement sous la forme d'une redondance croissante.

1.9 Les problèmes de robotique en essaim

Dans le tableau 2.2, nous résumons notre étude en donnant : une brève définition du problème à résoudre, sa source d'inspiration, les approches utilisées pour modéliser le problème, des exemples des recherches en cours qui appartiennent au problème et enfin la classification des problèmes.

Problématique	Sources d'inspiration	Approches de modélisation
Agrégation : regroupement de robots d'essaims dans une région de l'environnement.	Nature (par exemple, bactéries d'agrégation, blattes, abeilles, poissons, etc.).	<ul style="list-style-type: none">• Machines à états finis probabilistes.• Évolution artificielle.
Formation de motifs : déployer des robots de manière régulière et répétitive en respectant des distances spécifiques afin de créer le motif souhaité.	<ul style="list-style-type: none">• Biologie (par exemple, la disposition spatiale des colonies bactériennes et les schémas chromatiques de certains animaux).	Conception virtuelle basée sur la physique.

	<ul style="list-style-type: none"> Physique (distribution de molécules et formation de cristaux). 	
<p>Formation de chaînes: des robots à positionnement automatique se connectent en deux points. La chaîne qu'ils forment peut ensuite être utilisée comme guide de navigation ou de surveillance.</p>	Fourmis	<ul style="list-style-type: none"> Machines à états finis probabilistes. Conception basée sur la physique virtuelle. Évolution artificielle.
<p>Auto-assemblage et morphogenèse: Relier des robots d'essaims physiques pour créer des structures (morphologies).</p>	Fourmis (ponts, radeaux, murs...).	<ul style="list-style-type: none"> Machines à états finis probabilistes. Conception basée sur la physique virtuelle. Évolution artificielle.
<p>Exploration collective</p>	Animaux sociaux (fourmis, abeilles, etc.).	<ul style="list-style-type: none"> Machines à états finis probabilistes. Conception virtuelle basée sur la physique. Routage réseau
<p>Transport collectif: Coopérer pour transporter un objet.</p>	Les coopératives portent leurs proies dans les colonies de fourmis.	<ul style="list-style-type: none"> Machines à états finis probabilistes. Évolution artificielle

Tableau 1.1 : Problèmes de la robotique en essaim

1.10 Plateformes de simulation robotique en essaim

L'utilisation de nombreux robots physiques dans les recherches sur la robotique en essaim est difficilement abordable. Ainsi, des simulations informatiques sont développées pour tester visuellement les structures et les algorithmes sur ordinateur avant de se lancer dans des tests de robots physiques réels. L'utilisation de simulations informatiques, qui sont généralement plus faciles à configurer et moins onéreuses, est normalement plus rapide et plus pratique à utiliser que des essaims physiques. Dans la section ci-dessous, nous mettons en évidence les plateformes de simulation bien connues et largement utilisées dans la robotique en essaim de recherche.

Joueur / étape (Player/stage)¹

- est un package combiné d'outils logiciels libres développés pour les applications de robot et de capteur.
- Le contrôle des robots peut être programmé à l'aide d'un langage de programmation multiple pouvant être exécuté dans n'importe quel ordinateur connecté au robot.
- Le composant de scène est un simulateur de plusieurs robots interfacé avec Player.

Gazebo²

- Il inclut une simulation précise de la physique des corps rigides (un retour réaliste du capteur et des interactions possibles entre les objets peuvent ensuite être générés.)
- Gazebo présente une interface standard de Player en plus de sa propre interface native.

ARGoS³

- ARGoS est un nouveau moteur multi-physique enfichable permettant de simuler en temps réel la robotique massive et hétérogène d'essaims.
- facile à mettre en œuvre et à utiliser.
- les multiples moteurs physiques peuvent être utilisés dans une seule expérience,
- les robots peuvent migrer de l'un à l'autre de manière transparente.
- ARGoS peut également être implémenté en parallèle dans la simulation.

Enki⁴

- C'est un simulateur de robot basé sur la physique 2D rapide écrit en C++.
- simuler les essaims de robots. Il est également capable de simuler la cinématique, la collision, les capteurs et les caméras de robots travaillant sur une surface plane.
- Enki est conçu pour prendre en charge plusieurs systèmes de robots réels existants.

V-REP⁵

- est un simulateur de robot 3D open source permettant de créer des systèmes robotiques complets.
- écrit en C / C++, Python, Java, Lua, Matlab, Octave ou Urbi.
- idéal pour une application multi-robot. (peut être directement attaché aux objets dans la scène et en cours d'exécution)

- V-REP est utilisé pour le développement d'algorithmes rapides, les simulations d'automatisation d'usine, le prototypage et la vérification rapides, la formation à la robotique, la surveillance à distance, le double contrôle de la sécurité, etc.

UberSim⁶

- est un simulateur développé par Carnegie Mellon pour une validation rapide avant de télécharger le programme sur de vrais scénarios de football robot.
- UberSim utilise le moteur physique ODE pour des mouvements et des interactions réalistes.
- écrit en C dans le simulateur et le programme peut être chargé sur les robots à l'aide de TCP / IP.

USARSim (Unified System for Automation and Robot Simulation):⁷

- est un simulateur multi-robot haute-fidélité développé à l'origine pour les activités de recherche en recherche et sauvetage (SAR) du concours Robocup.
- Il repose sur un moteur de jeu largement utilisé, Unreal Engine 2.0.
- Le simulateur tire pleinement parti de la physique haute précision, de la simulation de bruit et de nombreuses géométries et modèles du moteur.

1.11 Conclusion

La robotique en essaim est un domaine de recherche relativement nouveau qui s'inspire de l'intelligence et de la robotique en essaim. C'est le résultat de l'application de techniques d'intelligence en essaim à la multi-robotique. Dans le présent chapitre, nous avons présenté un aperçu de la robotique en essaim ainsi que les théories basées sur ce domaine pour mieux comprendre la recherche de spécialité.

Chapitre II

Les Patterns d'agrégation dans les systèmes robotique en essaim

Chapitre 2

Les Patterns d'agrégation dans les systèmes robotique en essaim

2.1 Introduction

Dans ce chapitre, la question principale abordée dans ce travail est discutée. En particulier, les systèmes d'essaims robotisés sont présentés en détail : (1) modèles d'agrégation et. Dans un premier temps, nous présentons d'abord quelques-unes des études portant sur l'agrégation de motifs dans des systèmes d'essaims naturels, puis sur ces études, nous passons en revue un certain nombre d'ouvrages pertinents sur les motifs d'agrégation par robotique d'essaims. Les travaux sont classés en deux types principaux d'approches : (1) une méthode "basée sur les repères" et (2) une méthode auto-organisée. Nous nous concentrons essentiellement sur la deuxième méthode et, par conséquent, nous discutons plus en détail les différentes approches d'agrégation auto-organisées qui ont été utilisées jusqu'à présent dans la littérature.

2.2 Agrégation dans la nature

L'agrégation (ou la réunion) est un comportement fondamental observé dans de nombreux organismes biologiques, tels que les insectes sociaux et les animaux vivant en groupe. Les sociétés animales doivent impérativement accomplir collectivement des comportements d'essaimage complexes. Cela peut être utile dans différentes tâches comme la survie des individus, l'évitement des prédateurs, l'augmentation des chances de trouver des aliments, etc. Dans la nature, l'agrégation peut être obtenue à l'aide de gradients externes appelés signaux, tels que l'humidité pour les cloportes [ou la température pour les abeilles domestiques] .Ce type d'agrégation est principalement connu sous le nom d'agrégation basée sur les mémoires où les zones optimales sont marquées par une mémoire spécifique qui initie le processus d'agrégation. L'agrégation peut également être réalisée de manière auto-organisée, où le processus d'agrégation est activé sans aucune indication externe. Des exemples fascinants d'un tel comportement peuvent être observés dans les troupeaux d'oiseaux, les bancs de poissons et les troupeaux de mammifères. Dans ce type d'agrégation, aucun signal externe n'est requis pour former des agrégations, les individus forment plutôt des zones d'agrégation aléatoires sans préférence particulière pour leur condition.[khaldi ,2018]

2.3 Émergence de pattern d'agrégation d'essaims naturels

Dans la nature, de merveilleux modèles auto-organisés s'observent dans les comportements collectifs de nombreux organismes biologiques au cours de leur processus d'agrégation. On les voit particulièrement dans les bancs de poissons, les flocages d'oiseaux, les colonies d'insectes et les essaims de bactéries. Les motifs ou formes observés, définis comme des «ordres intégrés dans la règle», peuvent ressembler à des arrangements spatiaux ou à des séries temporelles, et les éléments qui le composent peuvent sembler identiques ou avec des variations.[Cheng-Ming et al ,2009].

Biologiquement, le processus à l'origine de la création et de la génération de ces ordres fascinants a été appelé «formation de modèles». Il fait référence au processus par lequel un ensemble cohérent d'associations entre les États de l'élément est formé et persiste pendant un certain temps, il capture l'essence de l'auto-organisation et de l'émergence dans toutes sortes de systèmes. Une question intéressante à se poser est la suivante : comment ces modèles ont-ils été générés ou formés? Une réponse à cette question peut faire une distinction fondamentale entre les modèles créés par l'ordre imposé par une autre organisation externe et ceux qui ont été créés par le modèle généré en interne. Intéressant pour ceux qui sont générés en interne, on pense que l'ordre spatio-temporel au niveau du groupe de ces modèles auto-organisés ne se dégage que de simples règles d'interaction locale entre les composants de niveau inférieur du groupe. De plus, ces règles sont spécifiées en impliquant des décisions individuelles non indépendantes par le biais d'un transfert d'informations local entre les membres du groupe.[khaldi ,2018]

2.4 Étude de cas : patterns de flocage

Les modèles de flocage sont l'un des phénomènes fascinants auto-organisant dans l'espace que l'on observe dans un troupeau d'animaux de taille et d'orientation corporelle similaire. Ils se produisent souvent lorsqu'un grand nombre d'individus se déplacent en masse ou migrent dans la même direction avec un objectif de groupe commun. Des exemples familiers d'un tel phénomène peuvent être observés chez les oiseaux qui affluent et les bancs de poissons. Un «murmure» d'étourneaux chez les oiseaux qui se rassemblent, par exemple, est l'un des phénomènes les plus captivants exposé l'hiver devant nos yeux. Ils sont capables de produire des motifs de troupeaux remarquablement dynamiques et de formes très variables, et se déplacent de manière coordonnée.[George F Young et al,2013]

2.4.1 Modèles sur les motifs de flocage

Mathématiquement, plusieurs modèles ont été développés pour montrer comment ces comportements collectifs complexes émergent de l'auto-organisation à partir de quelques règles d'interaction simples entre individus. L'un des modèles les plus anciens et les plus simples de comportement de flocage d'oiseaux basé sur les individus fait référence au modèle de Boids de Reynolds. Dans ce modèle, le comportement collectif global découle de l'application de trois règles heuristiques (séparation, alignement et cohésion), qui se sont révélées efficaces dans de nombreux comportements de groupes biologiques. Plus tard, un autre modèle de comportement populaire mais collectif.

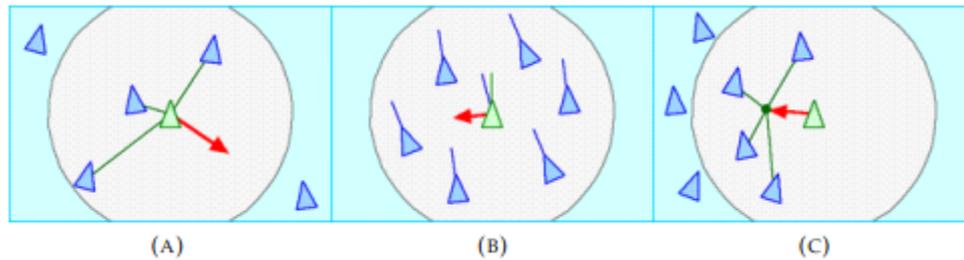


Figure 2.2: Les trois comportements de base déterminant le mouvement des boids. (A) Séparation : réagit uniquement aux membres du groupe voisins. (B) Alignement: direction vers la direction moyenne des membres du groupe. (C) Cohésion: se rapprocher de la position moyenne des boids voisins

Les agents de flocking de ce modèle sont décrits comme des particules automotrices qui ont tendance à se déplacer dans la direction moyenne de leurs voisins en présence de bruit externe. Les directions principales impliquent des modèles sans règle d'alignement explicite, des modèles utilisant des termes cohésifs supplémentaires et des modèles modifiant le terme de bruit.

Dans d'autres études, les recherches se sont tournées pour accorder davantage d'attention à la nature des modèles d'agrégation. Dans la plupart d'entre eux, un ensemble de primitives attraction physique / répulsion (A/R) est appliqué entre individus. Par exemple, dans le modèle A/R, il a été prouvé analytiquement qu'un motif en anneau stable était obtenu en un temps fini en utilisant un ensemble de valeurs minimales de force intermoléculaires. Il a été proposé aussi un modèle A/R de particules auto-entraînées pour analyser les mécanismes de formation de deux motifs d'agrégation distincts : des amas de type liquide et des amas de cristal. Ces deux modèles ne dépendent que de la distance de coupure de la fonction A / R. Dans une autre étude où la dynamique spatiale de la volée d'oiseaux ainsi que la scolarisation du poisson et ont conçu un modèle d'auto-organisation basé sur les A/R qui incorpore une règle d'alignement pour améliorer la formation de groupes et la cohésion entre individus. [Veysel Gazi et al ,2013] [Iain D Couzin et al ,2002]

2.5 Patterns d'agrégation dans la robotique en essaim

2.5.1 Agrégation dans la robotique en essaim

Sur la base des études biologiques citées dans la section précédente, l'agrégation est également un sujet d'intérêt dans diverses études de robotique en essaim. Elle est considérée comme une tâche fondamentale permettant aux essaims de robots d'accomplir des tâches complexes, tels que le mouvement collectif, l'auto-assemblage et la formation de motifs, ou d'échanger des informations. C'est un comportement souhaité qui est appliqué dans les systèmes de robots multi-agents et d'essaims. De plus, de nombreux comportements collectifs perçus chez des essaims biologiques et dont certains sont éventuellement appliqués à l'ingénierie de systèmes de robots multi-agents et d'essaims apparaissent dans des essaims agrégés. [Veysel Gazi et al ,2007]

2.5.2 Émergence de patterns dans l'essaim de robotique agrégée

Les schémas auto-organisés, l'un des comportements qui naissent de l'agrégation d'essaims, constituent une caractéristique intéressante de la robotique en essaims, qui organise depuis peu les comportements qui prennent un intérêt important dans le monde réel. Il peut être impliqué dans plusieurs applications réussies telles que :

Entourant un objet ou une caractéristique de l'environnement, formant une distribution uniforme de robots dans une zone donnée pour la protection de la zone ou la surveillance, l'élection d'un leader ou des situations de suivi du leader, un rassemblement pour échanger des informations ou effectuer d'autres tâches, élimination de mines ou de bombes, exploration et cartographie environnementales, formation de grilles de détection, gestion des processus dans une unité de fabrication, transport, déplacement et assemblage d'objets.

Le terme pattern, signifie des groupes de robots et afin de mener à bien ses tâches, doit être organisé en formations globales ou modèles pouvant aller de modèles simples à complexes. La formation de ces motifs peut faire référence à : des motifs géométriques contestés par le développement de comportements tels que les motifs géométriques souhaités soient formés par des individus d'essaims en mettant l'accent sur les interactions mutuelles entre eux, ou des motifs fonctionnels dictés par la environnement dans lequel la forme ou la taille géométrique des motifs formés est partiellement déterminée par la tâche à accomplir.

2.6 Approches sur les schémas d'agrégation en robotique en essaim

2.6.1 Méthodes à base de signal

Ces méthodes sont inspirées de ses homologues biologiques où des signaux spéciaux sont utilisés pour activer le processus d'agrégation. Dans une étude sur l'agrégation "basée sur les signaux", une source de lumière a été utilisée pour agréger un système robotique autour d'un objet et pour transporter l'objet de manière collaborative vers un autre objectif. En suivant des comportements simples (par exemple, trouver de la lumière et suivre la lumière), la tâche d'agrégation s'est achevée sans aucun mécanisme de communication explicite et proposer une méthode basée sur un repère infrarouge (IR) pour réguler la taille d'un agrégat créé par un système robotique permettant à chaque robot d'approcher la taille de l'agrégat et de décider de le rejoindre ou de le quitter en conséquence. Ils ont utilisé une approche d'agrégation basée sur les repères pour résoudre le problème de la prise de décision collective dans un système de robotique en essaim. Récemment, l'un des modèles d'agrégation à base de "cue" les plus réussis s'est inspiré du comportement collectif des abeilles domestiques, qui préfèrent la cueillette à une température de 36° C. Le modèle BEECLUST était le premier algorithme qui a imité ce comportement ; une source de lumière progressive a été utilisée pour générer un comportement de regroupement dans un système robotique en essaim. Différentes variations du modèle ont été suggérées pour améliorer les performances du processus d'agrégation. Ils ont proposé un nouvel algorithme d'agrégation dans lequel une vitesse dynamique et un temps d'attente comparatif ont été introduits dans le modèle BEECLUST d'origine, ce qui a

contribué à une amélioration significative du temps d'agrégation. Ils ont proposé une autre variante adaptative de BEECLUST, dans laquelle l'algorithme d'origine a été étendu pour s'adapter automatiquement à toutes les conditions d'éclairage. [Belkacem Khaldi et al , , 2018]

2.6.2 Méthodes auto-organisées

Dans les méthodes basées sur l'auto-organisation, les modèles d'agrégation sont obtenus en utilisant de règles simples d'interaction locale entre individus. Dans les sous-sections suivantes, nous soulignons les différentes approches proposées jusqu'à présent dans la littérature.

Approche probabiliste

La plupart des travaux de cette approche utilisaient une machine probabiliste à états finis pour contrôler le comportement de l'essaim. Par exemple, Ils ont adopté une approche probabiliste, inspirée du modèle de la blatte, pour réaliser l'agrégation à l'aide d'un essaim de 20 robots physiques Alice dans des environnements homogènes. Un travail similaire de [Nikolaus Correll et al ,2011] a montré que, lors de l'utilisation de règles d'agrégation probabilistes, une combinaison minimale de portée de communication et de vitesse de déplacement était nécessaire pour obtenir un seul groupe d'agrégats. [Onur Soysal et al,2005] ont suggéré une méthode d'agrégation probabiliste dans laquelle une machine à états finis était utilisée pour combiner un ensemble de comportements simples consistant notamment à éviter un obstacle, à s'approcher, à se repousser et à attendre.

Approche déterministe

Dans cette approche, les robots construisent généralement un graphique de visibilité connecté et veillent à ce qu'il soit maintenu en permanence.[Ando et al,(1999)] . a utilisé un algorithme dans ce sens pour étudier l'agrégation dans un groupe de robots mobiles à portée de détection limitée. La formation du graphique dans ces algorithmes était basée sur l'hypothèse que les robots étaient capables de mesurer à la fois la distance et l'angle de leurs voisins. Cependant, [Gordon, Wagner et al ,2004] ont réussi à obtenir une telle agrégation en utilisant uniquement la mesure de l'angle des voisins du robot. Les performances d'agrégation de ce dernier algorithme ont été améliorées par la suite, en introduisant une capacité supplémentaire de détection de distance grossière permettant de différencier si les robots voisins étaient proches ou lointains. Dans une autre étude, De [Gennaro et al ,2006] ont utilisé la matrice Laplacienne pour permettre à chaque robot de créer son propre graphe de proximité. Le contrôle associé était entièrement décentralisé et les résultats simulés démontraient que le modèle était efficace et augmentaient même la connectivité de tout l'essaim.

Approche de l'évolution artificielle

Dans certains travaux, des modèles d'agrégation auto-organisés ont été abordés à l'aide de techniques d'évolution artificielles. Par exemple, l'agrégation avec des robots simples, appelés

s-bots, a été étudiée. Dans cette étude, des solutions générales au problème de l'agrégation ont été élaborées à l'aide d'un mécanisme robotique évolutif. La méthode était capable de produire des comportements de clustering avec des stratégies comportementales statiques et dynamiques. Dans une autre étude,[Gauci et al,2014]. ont proposé deux algorithmes, un contrôleur réactif sans mémoire et un contrôleur récurrent avec mémoire pour étudier l'agrégation dans un essaim de robots e-puck. Les algorithmes étaient basés sur une technique de programmation évolutive classique et utilisaient un simple capteur binaire avec une plage de détection suffisante pour obtenir une agrégation sans erreur. Les résultats de la simulation et des expériences ont montré que l'agrégation vers un cluster était réalisée avec succès. Cependant, une plage suffisamment longue dans le capteur binaire était nécessaire pour obtenir une agrégation précise.

Approche inspirée de la morphogenèse

La morphogenèse biologique, y compris ses mécanismes internes génétiques et cellulaires, est récemment devenue une source d'inspiration pour de nombreuses études multi-robotiques. Cela a donné naissance à la robotique morphogénétique en tant que nouveau domaine de recherche en robotique émergent pour étudier l'auto-organisation de robots en essaim ou modulaires. Pour résoudre le problème des modèles d'agrégation dans ce contexte, [Guo, et al ,2011] ont établi une métaphore entre systèmes multi-cellulaires et systèmes multi-robots afin de proposer un algorithme décentralisé basé sur GRN (Gene Regulatory Network) pour la construction de formes multi-robots. Grâce à ce modèle GRN, une auto-organisation autonome en différentes formes prédéfinies et une auto-réorganisation adaptative dans des environnements dynamiques peuvent être effectuée par plusieurs robots. Plus tard, les auteurs ont proposé deux extensions du modèle original : en introduisant une représentation de forme libre permettant de créer davantage de motifs 2D ou 3D complexes, à structure b-spline rationnelle non uniforme, et en adoptant un réseau de réglementation hiérarchique des gènes H-GRN (H-GRN). C'est un modèle pour la génération adaptative de modèles multi-robots et la formation dans des environnements changeants.

Approche de la physique artificielle

Cette approche, qui appartient aux méthodes bio-inspirées, s'inspire de l'observation de la physique. Il a été introduit pour la première fois par [William M Spears et al,2004]. comme cadre physico mimétique (ou physique artificielle). Pour contrôler le comportement de l'ensemble du système d'essaims, le cadre utilise des forces de physique virtuelles générées par les interactions des robots. Le cadre a pu, en utilisant deux types de lois de force physiques : la loi de force newtonienne et la loi de force de Leanar-Jones, pour conduire de grands groupes d'agents d'agrégation se déplaçant dans une formation souhaitée tel qu'un réseau hexagonal. En outre, le cadre est étendu pour gérer les formations en mouvement à travers les champs d'obstacles.

Les modèles de contrôle virtuels basés sur les ressorts ont reconnu un intérêt significatif au cours des dernières années. Dans ces modèles, des forces de ressort virtuelles sont appliquées

pour maintenir une distance souhaitée entre les robots agrégés, tout en offrant une certaine flexibilité à la structure et une souplesse au mouvement. [Donghwa Jeong et al ,2014] ont développé un modèle dynamique basé sur un amortisseur de ressort virtuel pour un système d'essaims artificiels, dans lequel des algorithmes de formation de la dispersion et de l'agrégation de lignes sont proposés pour réaliser des forces attractives et répulsives entre les agents artificiels et leurs voisins. L'algorithme de dispersion est basé sur des éléments de planification trigonaux sans chef alors que l'algorithme de formation de ligne utilise des éléments de ligne appariés avec un chef provisoire.

2.7 Conclusion

Nous avons passé en revue les différentes approches utilisées en robotique en essaim pour trouver des solutions aux modèles d'agrégation. Premièrement, en se basant sur les schémas d'agrégation observés dans la nature, un certain nombre d'études sur l'agrégation de robots en essaim ont été examinées. Ces études ont été classées en deux approches principales : une méthode basée sur les signal et une méthode auto-organisée. Dans l'approche basée sur les repères, le processus d'agrégation est lancé à l'aide de gradients externes appelés repères ou signaux (sources de lumière, signaux sonores, etc.), placés quelque part dans l'environnement en tant que marques spécifiques identifiant les zones optimales. L'agrégation devrait avoir lieu. Dans l'approche auto-organisée, le processus d'agrégation est activé quelque part dans l'environnement sans exiger de signaux externes. Les robots forment plutôt des zones d'agrégation aléatoires sans préférence particulière pour leur condition. L'approche auto-organisée a pris beaucoup de place dans ce chapitre.

Chapitre III

La Plateforme ARGoS

Chapitre 3

La Plateforme ARGoS

3.1 Introduction

La simulation est un outil indispensable pour le prototypage de solutions de systèmes multi-robots. Un simulateur permet de tester des idées dans un environnement sûr et contrôlable, en évitant d'endommager les plateformes de robots (souvent onéreuses) et d'autres objets. Un simulateur multi-robot comporte deux exigences principales : la flexibilité et l'efficacité. La flexibilité fait référence à la possibilité pour les utilisateurs d'ajouter de nouvelles fonctionnalités destinées à une expérience particulière, par exemple de nouveaux types de robots ou de nouveaux capteurs. L'efficacité, en revanche, fait référence à la capacité à fournir des performances satisfaisantes à l'exécution.

3.2 Simulation Plateforme (ARGoS)

ARGoS est un simulateur multi-robots open source mis au point dans le cadre du projet «Swarm anoid », financé par l'UE, pour étudier des outils et des stratégies de contrôle pour des essaims hétérogènes de robots. À présent, il s'agit d'une plate-forme de simulation largement utilisée dans de nombreuses recherches et projets consacrés aux commandes de synthèse pour les comportements d'essaimage. ARGoS est fourni avec tous les outils nécessaires au cycle de développement du code de commande de robot, de la conception à la validation sur de vrais robots. Par conséquent, il n'y a pas de différence entre le codage pour la simulation et la réalité. ARGoS peut simuler de grandes échelles d'essaims hétérogènes de robots en temps réel. Son architecture est conçue pour être flexible de telle sorte que des fonctionnalités personnalisées puissent être facilement modifiées ou ajoutées en tant que plug-ins ou modules, et que des ressources de calcul spécifiques puissent être configurées pour répondre aux exigences de certaines expériences .[Carlo Pinciroli et al,2012]

3.3 Moteurs physiques

ARGoS offre la possibilité d'exécuter plusieurs moteurs physiques en parallèle. Cette fonctionnalité est obtenue en découplant l'espace simulé des règles physique qui le mettent à jour. En d'autres termes, un moteur physique est une vue de l'espace simulé. En outre, tous les robots peuvent encore se percevoir, par exemple grâce à des capteurs de proximité et des caméras. Le partitionnement ne concerne que les règles physiques utilisées pour mettre à jour les objets simulés. Un autre aspect important est que les entités ne sont pas confinées dans un moteur physique pendant la totalité d'une expérience. Il est possible, si nécessaire, de déplacer des entités d'un moteur physique à un autre. [Carlo Pinciroli, et all ,2011]

3.3.1 L'architecture d'ARGoS

L'architecture d'ARGoS est décrite dans la figure suivante et une brève description de ses principaux modules est fournie ci-dessous. [Carlo Pinciroli, et all ,2011]

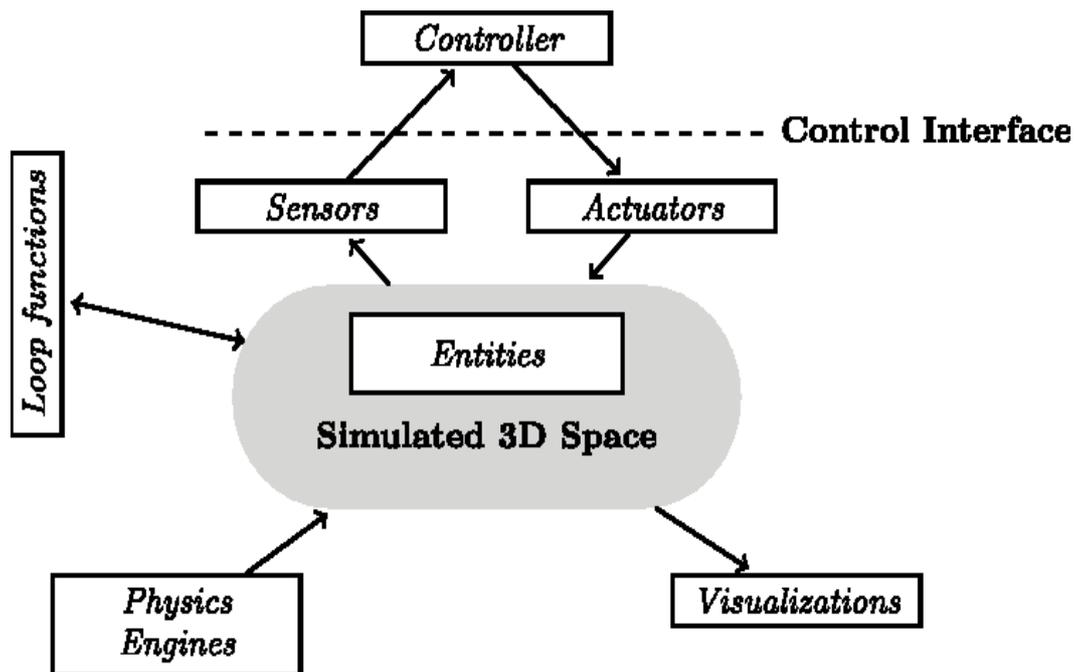


figure 3.3 : Description schématique de l'architecture ARGoS ¹

- **Espace 3D simulé:** Ce module est le composant principal de l'architecture ARGoS. C'est un ensemble d'entités, qui est considéré comme un entrepôt central contenant toutes les informations pertinentes (c'est-à-dire la position et l'orientation de robots ou d'obstacles) sur l'état de la simulation.
- **Capteurs et actionneurs :** les capteurs sont des plug-ins qui lisent l'état de l'espace 3D simulé, avec un accès uniquement à des types d'entités spécifiques pour effectuer leurs calculs (c'est-à-dire qu'un module de capteur de distance doit accéder aux informations relatives aux entités incorporées uniquement). De manière analogue, les plug-ins d'actionneur mettent à jour l'état du composant principal en écrivant dans les composants d'un robot.
- **Moteurs physiques :** avec les modules de moteurs physiques, l'état des entités incorporées peut être mis à jour pendant une expérience en exécutant plusieurs moteurs en parallèle. Cela peut être réalisé en affectant chaque moteur physique à une partie différente des entités incarnées.
- **Visualisations :** les modules de visualisation sont des mécanismes de rendu permettant de générer une représentation en sortie de l'état de l'espace 3D simulé. Par défaut, ARGoS offre une interface utilisateur graphique interactive basée sur Qt et OpenGL. Cependant, un moteur de rendu de haute qualité basé sur POV-Ray peut également être utilisé.
- **Contrôleurs :** le contrôleur est un plug-in qui interagit avec le composant principal via des capteurs et des actionneurs. C'est une implémentation du comportement individuel d'une entité (un robot, par exemple). Actuellement, il peut être implémenté dans ARGoS en utilisant les langages de programmation C++ ou Lua.

¹<https://www.semanticscholar.org/paper/ARGoS%3A-a-modular%2C-parallel%2C-multi-engine-simulator-Pincirol-Trianni/b45c69f287a764deda320160e186d3e9b12bcc2c>

- **Fonctions de boucle** : les fonctions de boucle sont des fonctions qui sont définies par l'utilisateur et qui peuvent être placées à des points précis de la boucle de simulation. Par exemple, ils peuvent être définis au moment de l'initialisation, ou avant et après l'exécution de la phase de mise à jour. Grâce à cette fonctionnalité et à chaque étape de la simulation, le moteur physique et son état peuvent être interrogés et modifiés, tandis que des données peuvent également être collectées pour une visualisation et une analyse plus poussées.

En outre, ARGoS dispose de modèles intégrés pour plusieurs robots bien connus tels que foot-bot, e-puck, kilobot et flybots, etc. Pour exécuter une expérience de simulation basée sur ARGoS, deux composants principaux doivent être fournis : un ensemble de contrôleurs et un fichier de configuration. Les contrôleurs sont des codes d'utilisateur qui incluent des implémentations du comportement individuel des robots et éventuellement des fonctions spécifiques à exécuter dans différentes parties d'ARGoS pour interagir avec l'expérience en cours. Actuellement, ils peuvent être implémentés en utilisant les langages de programmation C++ ou Lua. Le fichier de configuration est un fichier XML dans lequel une description de la structure de l'environnement simulé est fournie. Il contient toutes les informations requises pour configurer des entités simulées telles que l'arène, les robots, les moteurs physiques, etc. [Carlo Pinciroli et al., 2011]

3.3.2 Plateforme robotique

Pour discuter de la modularité d'ARGoS, nous décrivons comment un robot est modélisé dans ARGoS ? Le robot que nous avons choisi pour cette étude de cas est le foot-bot. Le foot-bot est un robot basé au sol qui se déplace avec une combinaison de roues et de pistes (appelée treels). Le robot est équipé de capteur et actionneur et permettent d'interagir avec l'environnement.

Foot-bot

Le foot-bot est un robot mobile différentiel à deux roues d'environ 17 cm de diamètre et de 29 cm de hauteur, conçu et construit dans le cadre du projet SWARM-BOTS. Le foot-bot peut se déplacer en utilisant une combinaison de roues et de pistes (appelée treels). Il est fourni avec divers capteurs et actionneurs permettant une interaction avec l'environnement. Ensuite, une liste des plus pertinentes utilisées dans les études de robotique en essaim :

- Douze (12) LED RVB composant un anneau entourant le corps du robot. Les motifs colorés peuvent être affichés pour les autres robots.
- Une caméra omnidirectionnelle pouvant être utilisée pour percevoir des objets colorés affichés par d'autres robots (jusqu'à une distance d'environ 50 cm).
- Quatre (4) capteurs au sol placés sous le châssis peuvent être utilisés pour percevoir des marqueurs ou des trous au sol.
- 24 capteurs IR pour la détection d'obstacles et de proximité.
- Un dispositif de communication à distance, appelé RAB, pour échanger des messages entre robots dans une plage limitée.
- Un connecteur de préhension permettant au foot-bot d'effectuer des connexions physiques telles que des connexions de préhension ou des connexions de robot à robot.

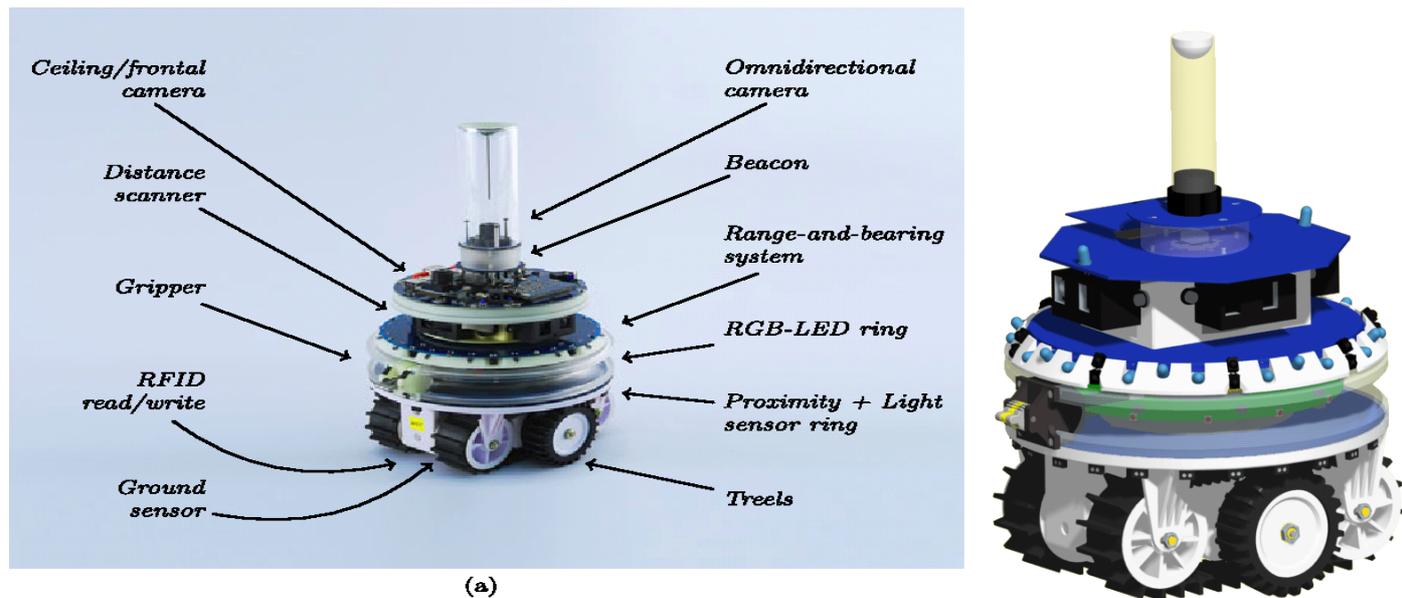


Figure 3.4:Principaux dispositifs composant le robot ARGoS "Le foot-bot" ²

3.4 Système de détection / activation

Dans cette section, nous donnons plus de détails sur les systèmes de détection / d'activation utilisés.

3.4.1 Capteur de distance et de portée infrarouge, (range and bearing)

Le robot foot-bot est équipé d'une carte qui permet de communiquer localement sans recourir à une référence externe. Avec ce système de communication, un robot peut communiquer avec ses voisins, tout en mesurant à la fois la portée et le relèvement (orientation) de l'émetteur (voir figure 3. 5). La carte de communication de la portée et du roulement est composée de 12 capteurs / actionneurs infrarouges qui permettent d'envoyer et de recevoir des messages dans une plage de communication de 6 m maximum et dans 12 directions différentes. Cette plage de communication avec laquelle les signaux infrarouges peuvent être détectés est réglable en temps réel. La particularité de ce module de communication infrarouge réside dans le fait que le même message peut être envoyé dans toutes les directions, ou plus précisément dans une direction par le biais du réglage auquel une paire de capteurs / actionneur est utilisée pour envoyer le message.

²<https://www.semanticscholar.org/paper/ARGoS%3A-a-modular%2C-parallel%2C-multi-engine-simulator-PincirolTrianni/b45c69f287a764deda320160e186d3e9b12bcc2c/figure/8>

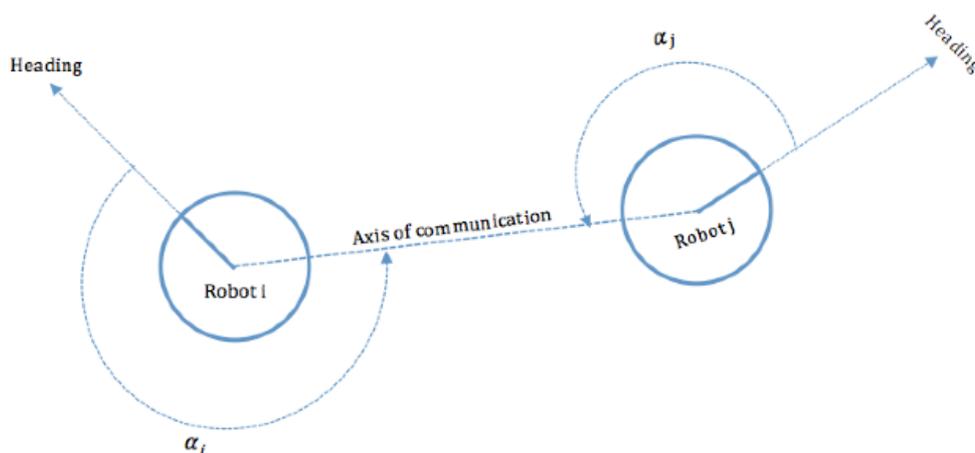


Figure 3.5: Les mesures de position relative dans le système de communication.

Dans ARGoS, l'utilisateur peut configurer un dispositif de communication de distance et de relèvement en tant que module ajouté à la liste des nœuds de capteurs / actionneurs du fichier de configuration XML. Si des paramètres supplémentaires spécifiques sont configurés pour cet appareil, le comportement du capteur sera parfait, ce qui signifie que tout message peut être reçu dans la plage de communication configurée et que la position relative (distance et valeurs de relèvement) de l'émetteur peut être mesurée avec précision. Cependant, plusieurs paramètres sont disponibles pour simuler avec plus de précision la portée et le périphérique de communication tels qu'ils se présentent dans le monde réel. Par exemple, l'utilisateur peut définir le paramètre `max_packets` pour limiter le nombre de messages pouvant être reçus dans un cycle pas à pas de contrôle. L'utilisateur peut également ajouter un bruit gaussien de la forme $N(0, \sigma^2)$ aux mesures de distance et de relèvement en définissant le paramètre `noise_std_dev`. La perte de données pendant la communication peut également être simulée en définissant le paramètre `loss_probability`, ce qui signifie que les paquets peuvent être perdus avec la probabilité configurée. Enfin, il est également possible de reproduire en simulation le caractère très bruyant de la mesure de distance des robots physiques en activant le paramètre `real_range_noise`.

3.4.2 Capteurs de proximité infrarouges

Le robot `foo-bot` est également équipé de capteurs de proximité infrarouges permettant de détecter des objets proches. La particularité de ces capteurs IR est qu'ils permettent à un robot de mesurer la distance de proximité d'un objet en s'appuyant sur le principe de la réflexion. Notez qu'un capteur de proximité IR typique est composé d'un transistor optique IR et d'une DEL IR, les deux pointant dans la même direction. Le principe de la réflexion est le suivant : lorsqu'un signal lumineux infrarouge émis par une DEL infrarouge est réfléchi par un objet, la lumière infrarouge réfléchie est alors détectée par un transistor optique infrarouge. L'intensité de la lumière infrarouge réfléchie dépend de la proximité de l'objet du capteur et de la

propriété de réflexion de la surface de l'objet. Plus la lecture de la LED IR réfléchi est élevée, plus l'objet devant un capteur est proche ; plus la peinture de la surface de l'objet est sombre, plus l'intensité du signal réfléchi est dure.

Pour les robots foot-bot, il existe 24 capteurs de proximité infrarouges d'une portée maximale de 10 cm, répartis équitablement autour du corps du robot (voir Figure 3.6). La lecture de chacun se compose d'un angle et d'une valeur. L'angle est mesuré en radians et correspond à la position relative du capteur par rapport à l'axe x local du robot. Alors que la partie valeur de lecture se situe dans la plage de [0,1] et correspond à l'intensité de la lumière infrarouge réfléchi. Si aucun obstacle n'est détecté, la valeur de lecture est 0, tandis que si un obstacle est détecté, la lecture sera supérieure à 0. À mesure que le robot se rapproche de l'obstacle, la valeur augmente.

La valeur de lecture du capteur est comprise entre 0 et 4095, tandis que la valeur la plus élevée indique à quelle distance l'objet se trouve devant un capteur. À l'instar du robot foot-bot, la lecture contient également les orientations des angles de la position d'un capteur par rapport à l'axe vertical du robot.

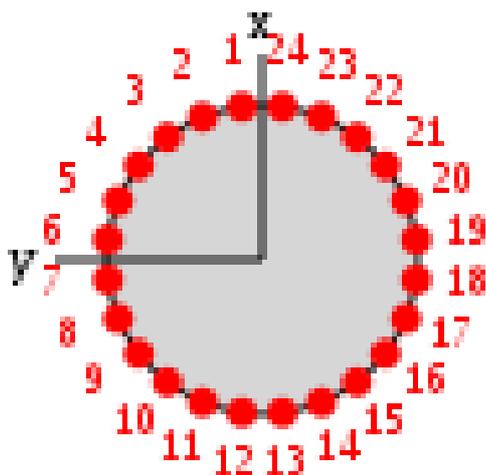


figure3 .6: Position des capteurs de proximité infrarouge dans le robot foot-bot

Dans ARGoS, les capteurs de proximité infrarouge peuvent également être configurés en tant que nœud dans le fichier de configuration XML. Une configuration par défaut des capteurs (voir Fig.) est utilisée pour synthétiser le contrôleur d'évitement d'obstacles pour les robots. Cela donnera un fonctionnement idéal aux capteurs de proximité.

29

```
<footbot_proximity implementation="default" show_rays="true"/>
```

figure 3.7: configuration des capteurs de proximité IR en XML.

3.4.3 Caméra omnidirectionnelle

Le foot-bot a deux caméras : une caméra omnidirectionnelle , L'image résultante est une image à 360 ° de l'entourage des robots. et une caméra qui peut être montée vers le haut ou l'avant."eye-bot's"dispose d'une caméra panoramique, qui est une caméra montée sur un dispositif rotatif. La mise en œuvre des caméras est spécifique à chaque robot. Les caméras des foot-bots sont des capteurs purs, tandis que la caméra pan-et-tilt "eye-bot's" est associée à un actionneur qui contrôle l'attitude de la caméra. Malgré ces différences, les implémentations des capteurs de caméra reposent sur une définition commune étendue aux besoins spécifiques de chaque type de caméra, garantissant ainsi la réutilisation du code.

Alors que les processeurs du traitement d'images auront beaucoup de plaisir à jouer avec le flux en direct, nous aurons quelque chose de plus simple : une détection de blob. Ce n'est pas un observateur extraterrestre, mais un algorithme qui produira une liste de taches de couleur, ainsi que leur position par rapport au centre du robot. Une goutte de couleur dans ARGoS correspond à une LED. La liste des blobs varie en taille avec le temps, en fonction de ce que voient les robots. Pour commencer à collecter des données, vous devez appeler "enable ()". Pour arrêter, appelez "disable ()". [Carlo Pinciroli · Vito Trianni et all ,2012]

3.4.4 Actionneur de roues

Le robot foot-bot est considéré comme un robot mobile à entraînement différentiel à deux roues, dans lequel chaque roue est entraînée indépendamment. Notez que le robot foot-bot se déplace à l'aide d'un système d'arbres composé d'un ensemble de roues et de pistes. Cependant, il est traité comme des roues normales. Le mouvement vers l'avant d'un robot mobile à entraînement différentiel est obtenu lorsque les deux roues sont entraînées à la même vitesse. Dans la théorie du contrôle basée sur un modèle, la cinématique d'un tel type de robots dans le repère global (voir la figure 3.8) est régie par l'équation 3.1.

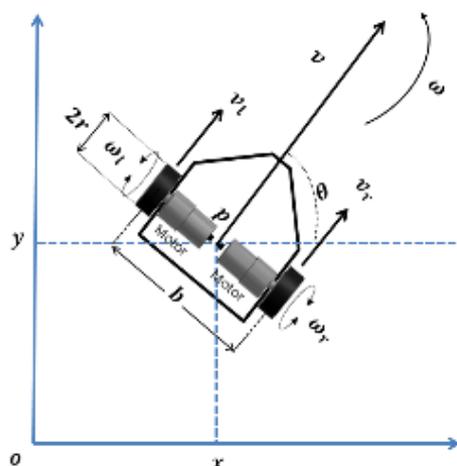


Figure 3.8: Cinématique d'un robot mobile à entraînement différentiel à deux roues.

$$\dot{\mathbf{p}} = \begin{bmatrix} \dot{x} \\ \dot{y} \\ \dot{\theta} \end{bmatrix} = \begin{bmatrix} \cos(\theta) & 0 \\ \sin(\theta) & 0 \\ 0 & 1 \end{bmatrix} \begin{bmatrix} v \\ \omega \end{bmatrix}, \quad 3.1$$

où $\mathbf{P}_i[\mathbf{x} \ \mathbf{y} \ \Theta]$ indique la position du robot dans le cadre global, et $[\mathbf{v} \ \omega]^T$ représente respectivement les vitesses linéaire et angulaire du robot. Ces vitesses peuvent être écrites en fonction de la vitesse angulaire des roues droite et gauche comme suit

$$\begin{bmatrix} v \\ \omega \end{bmatrix} = \begin{bmatrix} \frac{r}{2} & \frac{r}{2} \\ \frac{r}{b} & -\frac{r}{b} \end{bmatrix} \begin{bmatrix} \omega_r \\ \omega_l \end{bmatrix}, \quad 3.2$$

où r est le rayon de la roue et b la distance axiale entre les roues. Puisque les vitesses avant des roues du robot peuvent être calculées en fonction de leurs vitesses angulaires comme $\mathbf{v}_r = r\omega_r$ et $\mathbf{v}_l = r\omega_l$, alors l'équation 3.2 peut être réécrite comme suit:

$$\begin{bmatrix} v \\ \omega \end{bmatrix} = \begin{bmatrix} \frac{1}{2} & \frac{1}{2} \\ \frac{1}{b} & -\frac{1}{b} \end{bmatrix} \begin{bmatrix} v_r \\ v_l \end{bmatrix}, \quad 3.3$$

$$\begin{bmatrix} v_r \\ v_l \end{bmatrix} = \begin{bmatrix} 1 & \frac{b}{2} \\ 1 & -\frac{b}{2} \end{bmatrix} \begin{bmatrix} v \\ \omega \end{bmatrix}, \quad 3.4$$

et par conséquent, \mathbf{v}_r , \mathbf{v}_l peut être déduit de l'équation 3,4 comme suit : En conséquence, la cinématique d'un robot mobile à entraînement différentiel dans le repère de robot est donnée par l'équation suivante :

$$\dot{p} = \begin{bmatrix} \dot{x} \\ \dot{y} \\ \dot{\theta} \end{bmatrix} = \begin{bmatrix} \frac{1}{2} & \frac{1}{2} \\ 0 & 0 \\ \frac{1}{b} & -\frac{1}{b} \end{bmatrix} \begin{bmatrix} v_r \\ v_l \end{bmatrix}, \quad 3.5$$

A partir de l'équation 3,5 , le mouvement du robot est contrôlé en fournissant \mathbf{v}_l et \mathbf{v}_r comme entrées de contrôle. Ces vitesses sont générées par les moteurs associés aux roues. Une fois que ces vitesses sont données, le mouvement du robot se comportera comme suit : lorsque la roue droite est entraînée à une vitesse supérieure à celle de la roue gauche, le robot tourne à gauche ; dans le cas contraire, le robot tourne à droite. Alors que les roues sont entraînées dans la direction opposée à la même vitesse d'avancement, le robot va alors tourner sur place.

Dans ARGoS, cela peut être défini à l'aide de la commande `set_velocity` \mathbf{v}_l , \mathbf{v}_r . Par exemple, si une commande est définie sur `set_velocity (5, 5)`, le robot avancera à 5 cm / s. Des informations supplémentaires sur le mouvement du robot peuvent également être obtenues, telles que : `distance_left` et `distance_right` qui stockent la distance linéaire parcourue par les roues dans le dernier pas temporel, `velocity_left` et `velocity_right` qui stockent la vitesse actuelle de la roue, et `axis_length` qui représente la distance entre les deux roues en cm. Pour pouvoir utiliser ces commandes dans ARGoS, le dispositif d'actionneur différentiel doit être implémenté dans le fichier de configuration `argos.xml`.

3.5 Les méthodes

L'approche basée sur la physique est une méthode bio-inspirée qui s'inspire de l'observation de la physique. L'approche a d'abord été présentée comme un cadre physicomimétique (ou physique artificielle) et elle utilise des forces physiques virtuelles pour contrôler le comportement du système de robots en essaim. Tirant parti de cette approche, nous présentons dans la sous-section suivante une nouvelle approche permettant d'étudier les schémas d'agrégation auto-organisés au sein de systèmes d'essaims robotisés.

3.5.1 Le modèle viscoélastique: un modèle inspiré de la biomécanique

Ils ont proposé une nouvelle approche de physique artificielle qui s'inspire du processus biologique sous-jacent à la mécanique impliquée dans la morphologie cellulaire. En particulier, nous nous concentrons sur la structure cellulaire interne et la biomécanique connue entre ses composants sub-cellulaires tels que les membranes cellulaires, le cytoplasme cellulaire et le noyau. Plus précisément, c'est avec ces sous-composants qu'une cellule peut réagir à des forces mécaniques lui permettant de se déplacer et de changer de forme.

Vue d'ensemble de la cellule interne Bio-mécanique

En règle générale, le contenu intérieur d'une cellule est protégé par un mur appelé membrane cellulaire. Cette barrière, en plus du rôle qui joue dans la séparation du contenu intérieur de la cellule de son environnement, fournit également un mécanisme de contrôle des substances entrer

et sortir de la cellule. Le contenu interne d'une cellule est composé d'un cytoplasme et d'un noyau. Le cytoplasme cellulaire est une structure complexe qui peut être décrite comme un gel épais qui remplit la cellule et enveloppe le noyau. Au sein de celui-ci, de nombreuses activités cellulaires auront lieu, y compris plusieurs processus métaboliques qui sortent du cadre de cette thèse. Le cytoplasme cellulaire est plus qu'un fluide, il peut être décomposé en plus d'un organe. Nous citons ici le cytosquelette, considéré comme un gel polymère constitué de filaments d'actine réticulés . L'organe du cytosquelette est impliqué dans la formation des cellules et l'organisation des parties de la cellule, il fournit également un cadre pour le mouvement et la division cellulaires. Enfin, la membrane du noyau est le cerveau de la cellule qui dirige toutes sortes de fonctions traitées dans la cellule et qui contient l'ADN pour la réplication et la différenciation. [Shiladitya Banerjee,2013]

Malgré la structure hautement biologique d'une cellule typique et les processus biologiques qui y sont impliqués, nous nous intéressons plus particulièrement aux propriétés biomécaniques du composant du cytosquelette cellulaire. Les propriétés biomécaniques d'un cytosquelette cellulaire, telles que l'élasticité et la viscosité, sont essentielles à la validité de tout modèle proposé. [Jamali, Azimi et al,2010]ont suggéré un modèle mathématique pour représenter ces propriétés en utilisant des interactions viscoélastiques entre les membranes du cytosquelette (voir figure 3.9). Ces interactions viscoélastiques ont été modélisées à l'aide de sous-unités voigt, dans lesquelles les ressorts sont des approximations linéaires de l'élasticité de la cellule interne et des amortisseurs sont utilisés pour estimer la viscosité du cytosquelette.

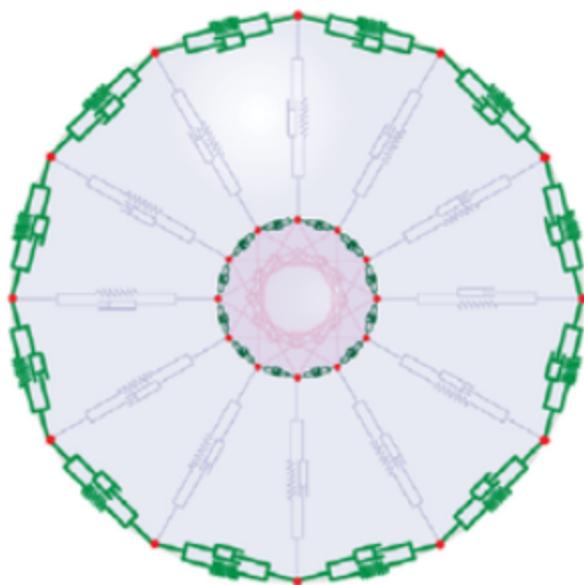


figure 3.9: Le modèle viscoélastique sous-cellulaire de Jamali, Azimi et Mofra

3.6 Conclusion

On a présenté dans ce chapitre les outils nécessaires pour la validation de notre approche. On a commencé par le simulateur ARGOS qui est utilisé par la plupart des chercheurs dans ce domaine. L'architecture principale ainsi que la plateforme robotique (foot-bot) qui sera utilisée pour notre étude. Ces matériaux et méthodes sont essentiels pour la synthèse de contrôleurs afin d'étudier le comportement collectif au sein d'une nuée de robots. De plus, nous avons discuté de tous les matériaux requis en termes de systèmes de détection et d'activation embarqués du robot qui seront particulièrement utilisés pour mettre en œuvre les contrôleurs proposés.

Chapitre IV

Le modèle viscoélastique

Chapitre 4

Le modèle viscoélastique

4.1 Introduction

Nous allons présenter dans ce chapitre notre modèle proposé. Le modèle viscoélastique de contrôle de formation simple pour un système robotique en essaims. Nous rapportons principalement les formations géométriques de base qui peuvent être obtenues en se basant sur la connectivité viscoélastique intra virtuelle entre les voisins. Le modèle est entièrement décentralisé et comprend de nombreux détails sur le modèle du robot, ainsi que des paramètres dynamiques pour les coefficients de ressort et d'amortisseur. Les configurations géométriques de base, telles que les triangles, les carrés, les pentagones et les cercles, peuvent être formées de manière dynamique en ne se basant que sur les distances relatives et les estimations des orientations des voisins. Ces configurations sont réalisées et stabilisées une fois que les forces viscoélastiques virtuelles exercées sur les robots se sont équilibrées.

4.2 Le modèle d'interaction viscoélastique des robots en essaim

Il est inspiré du modèle de Jamali[Yousef Jamali, et all 2010], Les auteurs ont adopté des liens viscoélastiques virtuels pour modéliser les interactions entre les robots de l'essaim et de ne pas régénérer les processus bio-mécaniques des cellules internes. La figure 4.10 schématise une configuration de modèle pour les interactions entre les robots de l'essaim. Chaque robot est alors affecté par les forces viscoélastiques virtuelles exercées par ses voisins. De plus, étant donné que les robots sont placés dans une zone d'entrée entourée de quatre murs et un modèle d'évitement d'obstacles devrait alors être intégré. Un vecteur de contrôle répulsif modélisé comme un champ potentiel et généré autour de chaque robot. Le champ avait une forte influence lorsqu'un robot était proche du champ potentiel et un effet décroissant lorsque le robot s'éloignait.

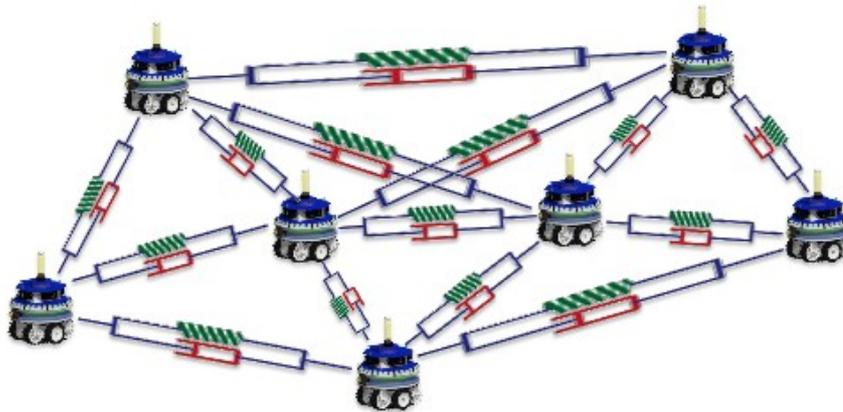


Figure 4.10: Le modèle d'interaction viscoélastique des robots en essaim

4.3 Les méthodes

Le modèle de contrôle viscoélastique virtuel est un modèle basé sur la physique qui a été appliqué avec succès en tant que contrôle proximal pour maintenir et organiser les robots ensemble à une certaine distance. [khaldi ,Foudil et all ,2017]

Nous allons étudier comment un essaim de foot-bots parvient à former des formations régulières en mettant en œuvre le contrôleur général (figure 4.11) . Nous illustrons dans l'algorithme 4.1 la manière dont le scénario de simulation a été implémenté dans le robot foot-bot et nous résumons les principaux événements survenus pendant la simulation.

Pour exécuter une expérience de simulation basée sur ARGoS, deux composants principaux doivent être fournis: un ensemble de contrôleurs (lua)et un fichier de configuration (xml). Les contrôleurs sont des codes d'utilisateur qui incluent des implémentations du comportement individuel des robots et éventuellement des fonctions spécifiques à exécuter dans différentes parties d'ARGoS pour interagir avec l'expérience en cours.

Le fichier xml , Contient des robots foot-bot placés arbitrairement dans une arène entourée de quatre murs . Les capteurs / actionneurs activés du robot foot-bot sont mis en évidence dans les nœuds de capteurs / actionneurs correspondants. Le moteur physique des différentes parties des entités configurées dans le fichier xml est défini sur dynamic2d. Enfin, la configuration de visualisation de la simulation est définie sur open-gl, avec la possibilité de basculer vers différentes vues de visualisation à l'aide des trois caméras configurées dans le noeud de visualisation. Dans le fichier de contrôleur, les robots doivent s'auto-organiser en formation géométrique de base telles que des triangles équilatéraux, des carrés ou des cercles.

Les robots sont supposés être simples avec des informations locales et chaque robot est capable de détecter ses partenaires proches dans son champ de vision.

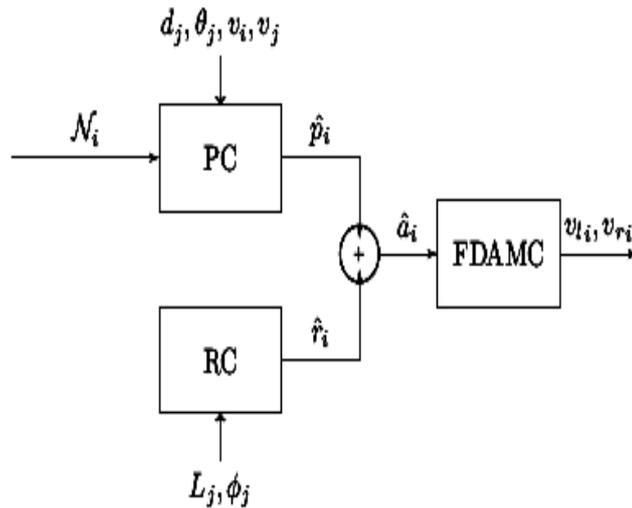


figure:4.11:Le modèle de contrôle global.

4.1: Algorithme principal : Le modèle global

Initialisation : d_0 , L_0 , b , k_ω , k_s , k_r , v_{max} .

début

```

pour ( chaque pas ) faire
|
|   pour (chaque robot) faire
|   |
|   |   Ni  $\leftarrow$  capturer_les_Voisins();
|   |   Pi  $\leftarrow$  PC(Ni); //Contrôle de proximité
|   |   Oi  $\leftarrow$  éviter_obstacle(); // l'ensemble des obstacles
|   |   Ri  $\leftarrow$  RC(Oi ); //Contrôle répulsif
|   |   ai  $\leftarrow$  somme_vecteur (Pi,Ri); // la somme des deux forces virtuelle
|   |   Roues_Vitesse  $\leftarrow$  FDAMC(ai); // Contrôle du mouvement angulaire dépendant
|   |                                     de l'avant
|   |
|   |   Fin
|   |
|   |   Fin
|   |
|   Fin
|
Fin
    
```

Paramètre	Description
d0	Longueur d'équilibre du ressort
L0	Seuil d'influence d'obstacle
b	Distance entre les roues
Kw	Gain de vitesse angulaire
Ks	gain de ressort constant
Kr	Gain de vitesse angulaire
Vmax	vitesse maximale
ω max	vitesse angulaire maximale

Tableau 4.2 Paramètres du modèle de formation de forme régulière

Dans ce modèle, le mouvement de l'essaim est régi par des forces viscoélastiques virtuelles, résultant des interactions des robots les uns avec les autres. Cela permet la connectivité et la cohérence entre les robots de l'essaim lorsqu'ils sont en mouvement.

L'algorithme principal 4.1 illustre le modèle de contrôle global mis en œuvre dans les foot-bots et en simulation algorithmique, elle produit le foot-bot à chaque pas de temps pour se rapprocher de l'objectif final calculé, résultant de la force virtuelle $\hat{\mathbf{a}}_i$ qui gouverne le mouvement de chaque robot. Cette force est utilisée pour actionner les roues du robot à l'aide d'un nouveau contrôle de mouvement proposé (MC) appelé Contrôle du mouvement angulaire dépendant de l'avant et de la force est calculée en faisant la somme des deux forces virtuelles, $\hat{\mathbf{p}}_i$ et $\hat{\mathbf{r}}_i$ 4.1 qui sont implémentées respectivement dans les fonctions PC et RC. La vitesse dans laquelle le robot se déplace vers la cible finale est implémentée dans la fonction FDAMC (Contrôle du mouvement angulaire dépendant de l'avant) conformément aux équations 4.10, 4.11 et 3.4. Le système d'essaims se stabilise là où les forces sont équilibrées et à ce moment la forme se forme automatiquement. Les robots continuent à tourner sur eux-mêmes pour maintenir la forme formée.

$$\hat{\mathbf{a}}_i = \hat{\mathbf{p}}_i + \hat{\mathbf{r}}_i. \quad 4.1$$

où $\hat{\mathbf{p}}_i$ est un contrôle de proximité (PC) qui encode pour les règles viscoélastiques, et $\hat{\mathbf{r}}_i$ est le contrôle répulsif (RC) qui encode pour les règles de potentiel répulsif. [khaldi et al,2017]

Les détails de sous-contrôleurs PC, RC et MC sont décrits dans les sous-sections ci-dessous:

4.3.1. Contrôle de proximité (PC)

Le contrôle de proximité est responsable du calcul de la force viscoélastique virtuelle $\hat{p}_i = [x_{\hat{p}_i} \ y_{\hat{p}_i}]^T$ exercée sur chaque robot. La figure 5.2 illustre une configuration de modèle entre Quatre pieds-bots interagissant avec un modèle viscoélastique, la force résultante de l'interaction de deux robots est calculée par : l'équation suit (4.3) , et algorithmique est comme suivent (algorithme 4.2) :

$$\hat{p}_i = \sum_{j \in N_i} \hat{P}_{ij} \quad 4.2$$

$$\hat{p}_{ij} = K_{ij}^s (d_{ij} - d_0) + K_{ij}^d (\vartheta_i - \vartheta_j) \quad 4.3$$

4.2: Algorithme PC(N_i)

début

var: N_i, V_i;

pour (i= 1 à N_i) **faire**

 f(x,y) ← calculer_la_force_pc(N_i, V_i); // **équation 4.3**

 p(x,y) ← somme_vecteur (p,f) ; // **équation 4.2**

fin

return (p(x,y)) ;

fin

où N_i est le voisin du robot, k_{ij}^s est la constante du ressort, d_{ij} est le vecteur de déplacement qui représente la longueur actuelle du ressort entre deux robots en interaction, d_0 est la longueur d'équilibre du ressort, k_{ij}^d est le coefficient d'amortissement, et v_i , v_j sont respectivement les vitesses en avant du robot focal **i** et de son partenaire voisin **j**.

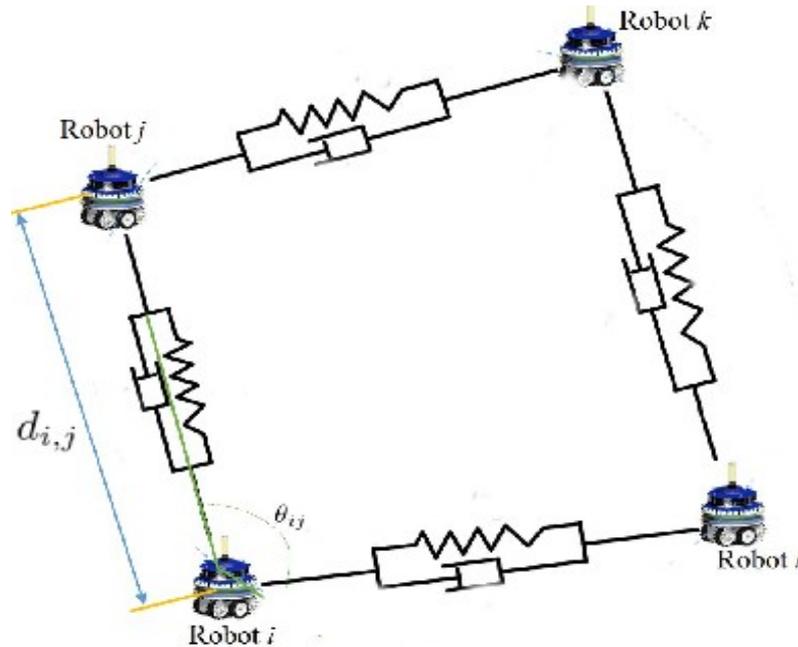


figure 4.12: Un modèle de la configuration de connexion viscoélastique entre quatre robots.

La plupart des travaux associés utilisent des constantes fixes pour les paramètres de ressort et d'amortissement. Dans notre modèle et afin de toujours obtenir une situation stable et de réduire l'oscillation provoquée par de tels systèmes, le coefficient de ressorts \mathbf{k}_{ij}^s et le coefficient d'amortissement \mathbf{k}_{ij}^d sont calculés selon les équations ci-dessous:

$$k_{ij}^s = \frac{k_s}{\sqrt{d_{i,j}}} \quad 4.4$$

$$k_{ij}^d = k_d \sqrt{k_{i,j}^s} \quad 4.5$$

où k_s et k_d sont des constantes de gain. La sortie du contrôle de proximité est le vecteur de force viscoélastique $\hat{\mathbf{p}}_i = [x_{\hat{p}_i} \ y_{\hat{p}_i}]^T$.

4.3.2 Contrôle répulsif (RC)

Les comptes de contrôle répulsifs pour calculer la force de répulsion virtuelle $\hat{\mathbf{r}}_i = [x_{\hat{r}_i} \ y_{\hat{r}_i}]^T$ qui repoussent le robot des autres robots ou des obstacles fixes. Dans notre modèle, la prévention des obstacles et des collisions est gérée à l'aide d'une force potentielle répulsive. La force potentielle répulsive n'agit que dans une zone proche de l'obstacle, et a une forte influence sur le robot lorsque celui-ci est très proche de l'obstacle et a une influence dégradante lorsque le robot est éloigné. Lorsqu'un robot est proche de plus d'un obstacle, il est repoussé par les forces répulsives totales $\hat{\mathbf{p}}_i$ résultant de la somme des forces répulsives $\hat{\mathbf{r}}_{ij}$

excrétées par l'ensemble des obstacles O_i qui illustré dans l'algorithme suivant(algorithme 4.3) .

$$\hat{r} = \sum_{j \in O_i} \hat{r}_{ij} \quad 4.6$$

chaque force potentielle répulsive \hat{r}_{ij} est calculée en utilisant le gradient négatif de l'énergie répulsive potentielle U_{ij} comme suit:

$$\hat{r}_{ij} = -\nabla U_{ij}. \quad 4.7$$

$$U_{ij} = \begin{cases} \frac{1}{2} k_r \left(\frac{1}{L_j} - \frac{1}{L_0} \right)^2, & L_j \leq L_0 \\ 0 & \text{elsewhere} \end{cases} \quad 4.8$$

$$\hat{r}_{ij} = \begin{cases} k_r \left(\frac{1}{L_j} - \frac{1}{L_0} \right) \left(\frac{1}{L_j^2} \right), & L_j \leq L_0 \\ 0 & \text{elsewhere} \end{cases} \quad 4.9$$

Où k_r est une constante d'échelle, L_j est la distance entre le robot et le bord de l'obstacle j le plus proche, et L_0 est le seuil d'influence de l'obstacle, et l'algorithme suivent Explique comment appliquer cette force pratiquement (algorithme 4.4).

4.3: algorithme O_i

début

```
var: vec = { x=0, y=0 }  
pour (i=1 a 24 ) faire  
    vec = {  
        x ← robot.proximity[i].value * math.cos(robot.proximity[i].angle),  
        y ← robot.proximity[i].value * math.sin(robot.proximity[i].angle) }  
    O(x,y) ← somme_vecteur (O,vec)  
fin  
retourne(O) ;
```

fin

4.4: algorithme **Rc**

début

O ← Oi();

Fi ← get_plus_proche_obst(O);

F ← atan2 (Fi.y, Fi.x); //calculer l'angle (en radian) à partir des coordonnées x et y

if (F.angle ≠ 0)

if (F.longueur ≥ d0)

 r ← calcule_répulsive_force; // **équation 4.9**

fin

fin

retourne(r);

fin

4.3.3 Contrôle du mouvement angulaire dépendant de l'avant (FDAMC)

Le FDAMC est utilisé pour contrôler la vitesse à laquelle le robot considéré se déplacera. Le FDAMC convertit le vecteur de contrôle de force virtuel $\hat{\mathbf{a}}_i = [\mathbf{x}_{\hat{\mathbf{a}}_i}, \mathbf{y}_{\hat{\mathbf{a}}_i}]^T$ en une vitesse avant (\mathbf{v}_i) et une vitesse angulaire ω_i , puis transforme ces vitesses en vitesses avant des roues gauche et droite du robot. Nous basons notre FDAMC sur le contrôle de mouvement à vitesse variable (VMC) proposé dans [Eliseo Ferrante et al,2012]. En utilisant ce contrôle de mouvement, les robots se déplacent à une vitesse et une vitesses angulaires variables. En VMC, la vitesse d'avancement \mathbf{v}_i et la vitesse angulaire ω_i sont directement proportionnelles aux composantes x et y. Nous différons notre FDAMC de ce travail en mettant à l'échelle l'angle formé par le vecteur $\hat{\mathbf{a}}_i$ pour obtenir la vitesse angulaire, puis la vitesse linéaire est obtenue en fonction de la vitesse angulaire comme suit:

$$\omega_i = K_\omega \left(\frac{\alpha_i \times 180}{\pi} \right) \quad \mathbf{4.10}$$

$$\vartheta_i = \frac{\vartheta_{max}}{\sqrt{|\omega_i| + 1}} \quad \mathbf{4.11}$$

4.5: algorithme **FDAMC** (force)

début

```
    angle= math.atan2(force.y,force.x) //calculer la force angulaire (en radian) à partir des
                                     coordonnées x et y de l'objectif
    ( ω i , v i ) ← calculer la vitesse angulaire et avancée( angle); // équation 4.10 et 4,11
    (vl ,vr) ← calculer la vitesse de rotation gauche et droite (i, v i); // équation 4.12
```

fin

où $\alpha_i = \text{atan2}(y_{\hat{a}i}, x_{\hat{a}i})$ et k_ω est une constante de gain. v_{\max} est la vitesse linéaire maximale autorisée pour le robot.

Pour obtenir un véritable contrôle du mouvement du robot, la vitesse angulaire doit être limitée à $[-\omega_{\max}, \omega_{\max}]$, alors que l'équation **4,11** garantit que la vitesse linéaire v_i est toujours comprise dans la plage $[0, v_{\max}]$. Comme il s'agit de robots mobiles à entraînement différentiel à deux roues, v_i et ω_i devraient être convertis en signaux qui actionnent les roues gauche et droite du robot. Pour cela, nous utilisons le modèle d'entraînement différentiel illustré dans l'équation **suivante** :

$$\begin{bmatrix} v_r \\ v_l \end{bmatrix} = \begin{bmatrix} 1 & \frac{b}{2} \\ 1 & -\frac{b}{2} \end{bmatrix} \begin{bmatrix} v \\ \omega \end{bmatrix}, \quad 4.12$$

4.4 Conclusion

La formation de motifs est l'un des aspects les plus difficiles des problèmes d'auto-organisation de la robotique en essaim, qui connaît actuellement un intérêt important dans le monde réel actuel. Dans ce chapitre, nous avons fourni une solution simple à ce problème en utilisant notre modèle d'interaction viscoélastique.

Avec ce modèle, nous ne traitons pas les robots comme des particules ponctuelles comme le faisait le contrôle de formation de ressort virtuel le plus connu, nous considérons chaque robot de l'essaim comme un simple robot mobile à deux roues motrices.

Chapitre V

Résultats

Chapitre 5

Résultats

5.1 Résultats

Pour évaluer les performances de la solution proposée, nous avons réalisé plusieurs expériences de simulation utilisant jusqu'à 10 bots. Nous simulons les expériences à l'aide du simulateur ARGoS.

Au début, N robots sont dispersés arbitrairement dans l'arène et se dirigent dans des directions aléatoires (**FIGURE 5.13**). Une fois l'expérience est commencée, les robots se connectent les uns aux autres pour exercer des pouvoirs de viscosité virtuels afin d'accomplir la tâche qu'ils recherchent. Afin de réaliser la tâche collectivement requise, les robots peuvent partager en utilisant la plage infrarouge et porter le dispositif de communication "range and bearing". Lorsque les robots se déplacent, ils évitent également les quatre murs entourant la place à l'aide de capteurs de proximité infrarouges dans la section de subdivision.

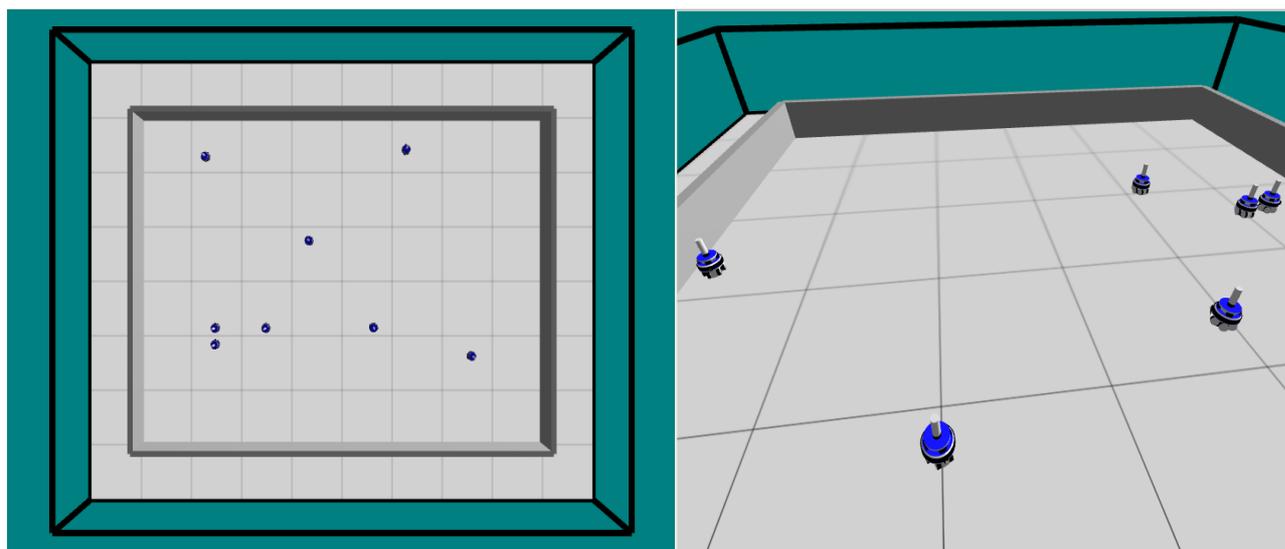


Figure 5.13 : Exemple de distribution aléatoire des robots dans l'arène

Pratiquement, pour atteindre le PC, le foot-bot utilise son dispositif RAB ou omnidirectional_camera pour envoyer et recevoir la vitesse linéaire aux robots proches, Avec les deux dispositif le foot-bot est capable de mesurer la distance et le relèvement relatifs (d_{ij} et φ_{ij}) du $j^{\text{ème}}$ robot voisin. Le PC suppose qu'un robot peut percevoir la portée et le relèvement de ses robots voisins dans une plage donnée.

Pour réaliser la RC, le foot-bot utilise ses capteurs de proximité IR pour obtenir la distance L_j de l'obstacle détecté dans l'armoire et son angle correspondant θ_j . La vitesse linéaire des roues

gauche et droite du robot, calculée à l'aide du FDAMC, est ensuite activée via la commande `set_velocity` (v_l , v_r).

Les illustrations des figures suivantes présentent le même scénario qui est appliqué aux deux capteurs différents sur plusieurs robots de 3 à 10 foot-bot répartis de manière aléatoire dans un espace bidimensionnel et qui tendent enfin à former de manière dynamique, en fonction du nombre de voisins détectés, des polygones réguliers géométriques, avec une propriété qui a tous les côtés égaux jusqu'à former un cercle.

5.A Résultat avec le capteur range and bearing (RAB)

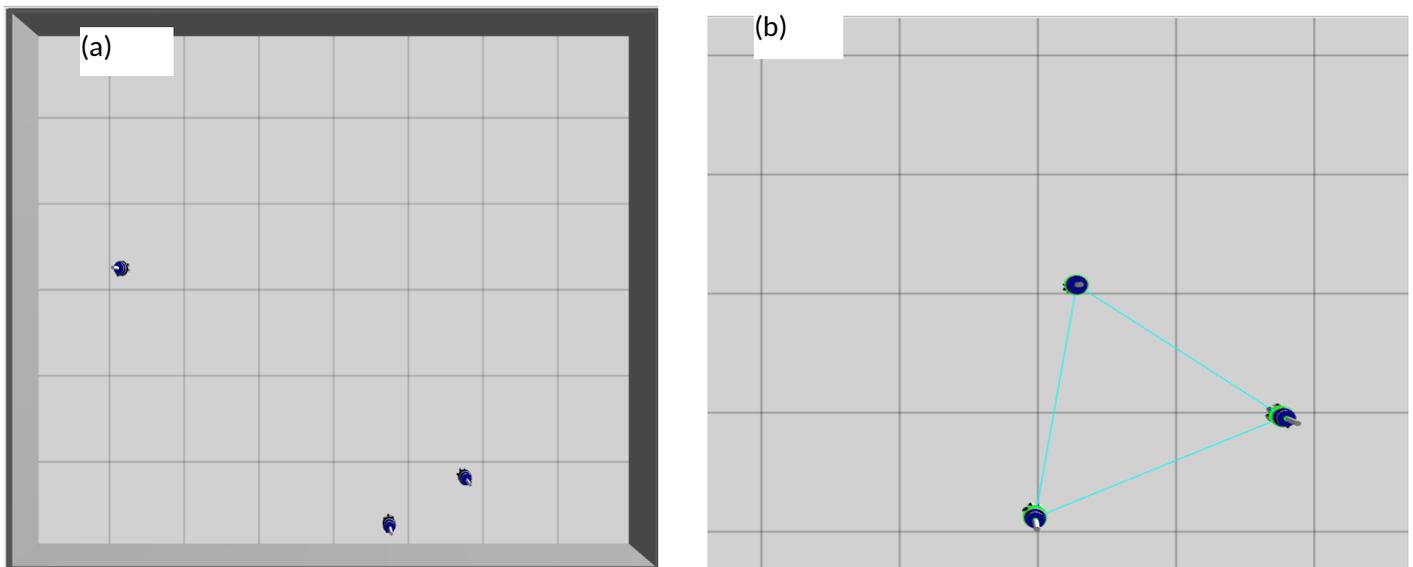


Figure 5.14 3 Robots formant un triangle équilatéral.

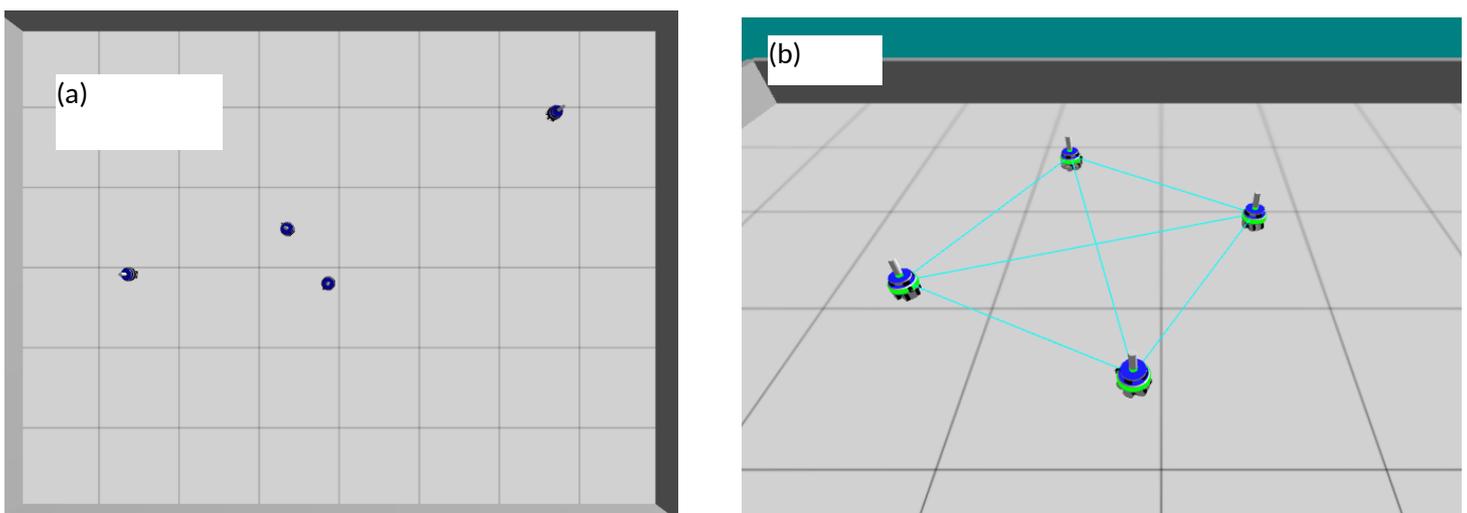


Figure 5.15 4 Robots formant un carré.

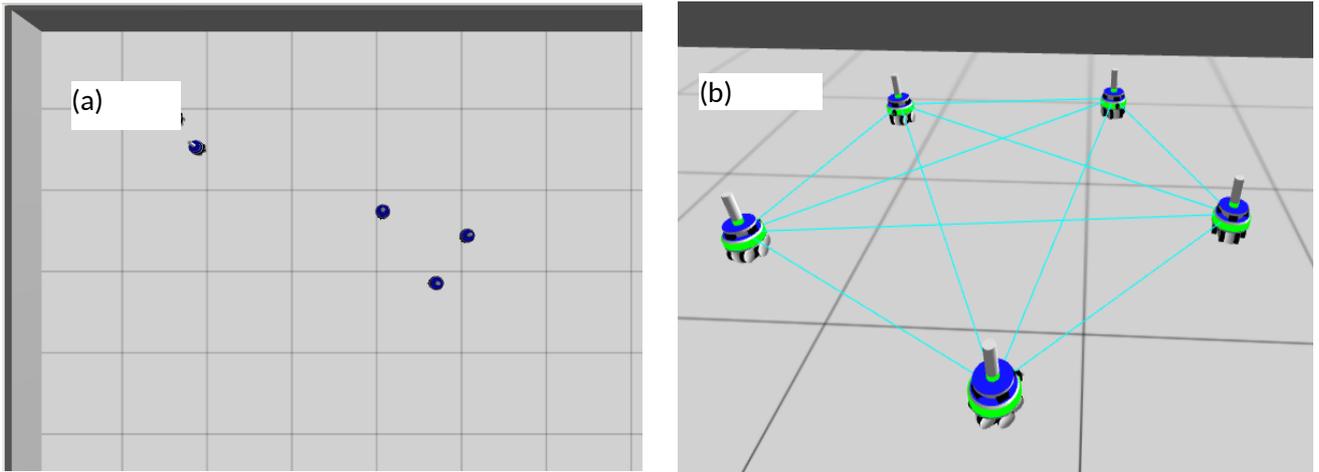


Figure 5.16: 5 Robots formant un pentagone régulier

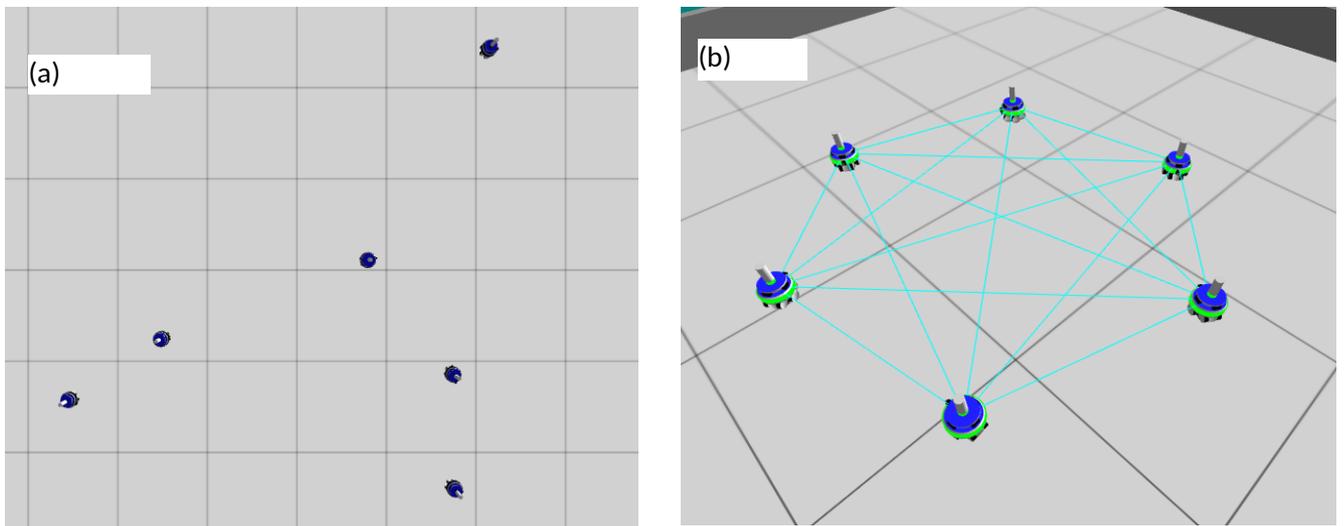
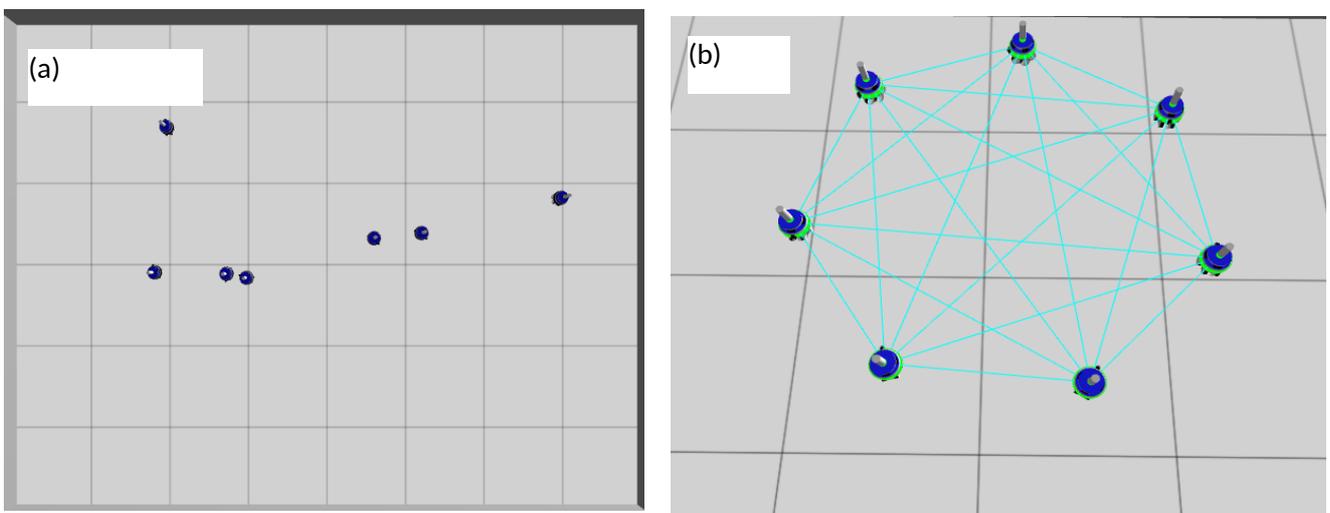


Figure 5.17 : 6 Robots formant un hexagone régulier.



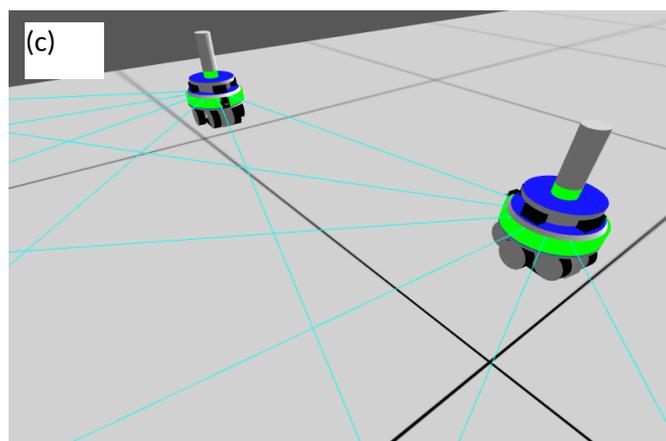


Figure 5.18 : 7 Robots formant un heptagone régulier.

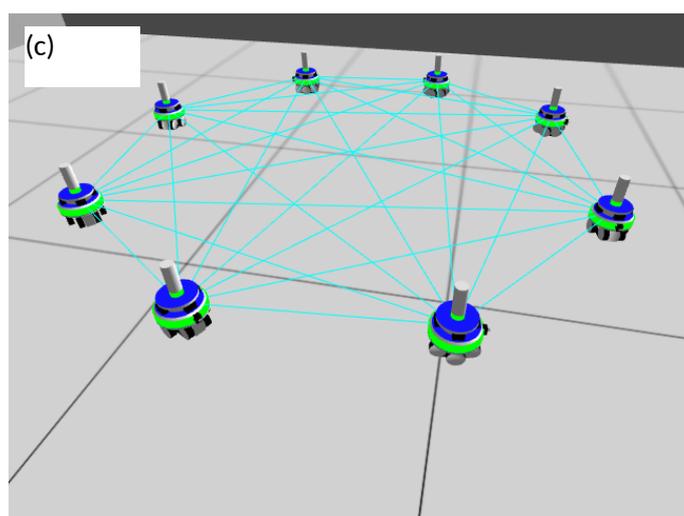
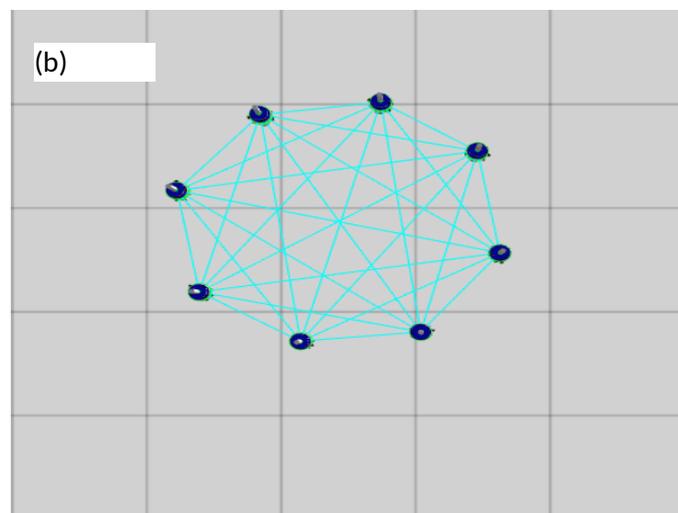
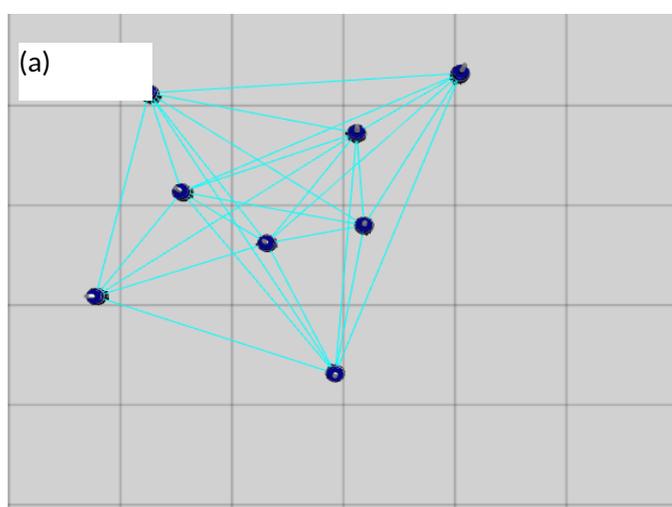


Figure 5.19 : 8 Robots formant un octogone régulier.

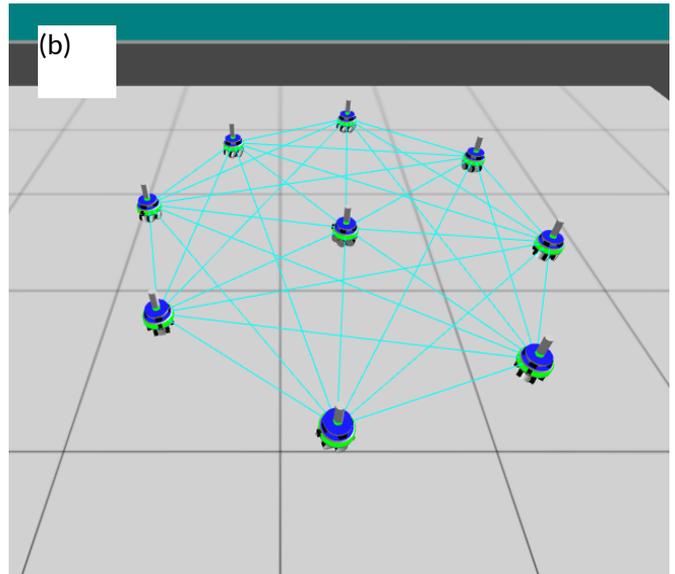
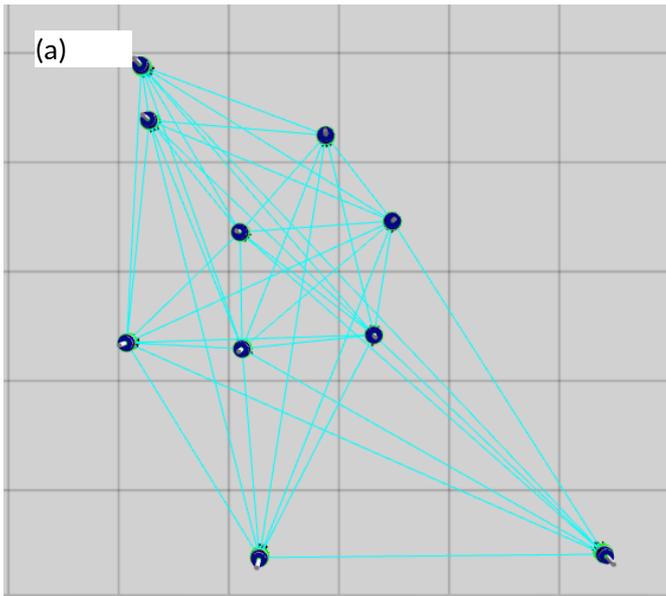


Figure 5.20: 9 robots formant un octogone régulier avec un robot au centre.

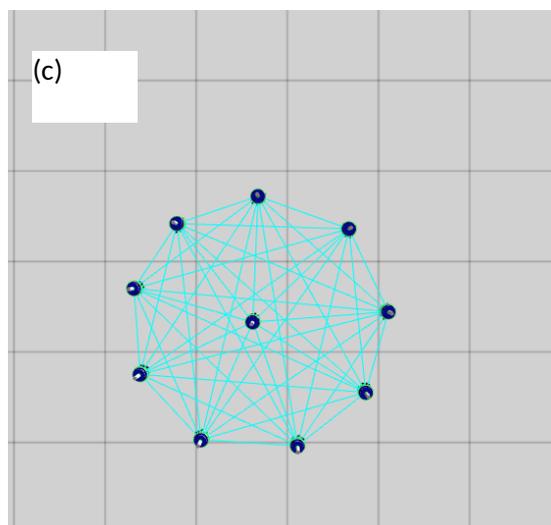
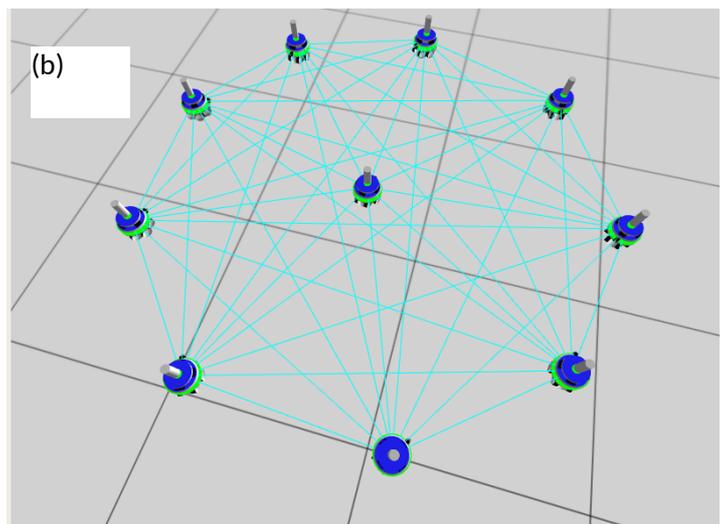
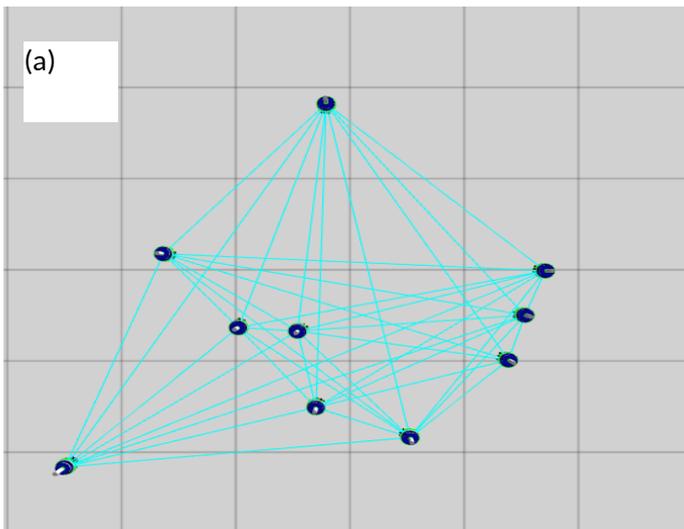


Figure 5.21 : 10 Robots formant un nonagone régulier avec un robot au centre.

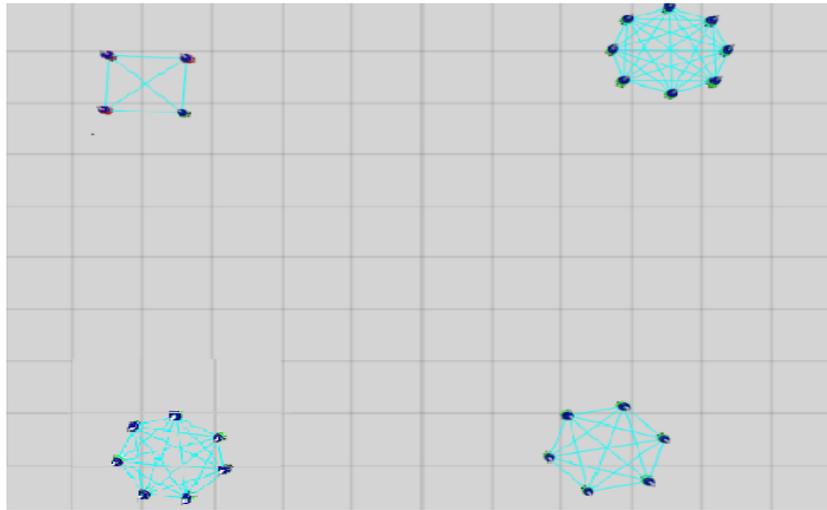


Figure 5.22 : 4 formes au même espace (4 robots , 6 robots, 7 robots, 10 robots) avec RAB.

5.B Résultat avec le capteur omnidirectionnel_camera :

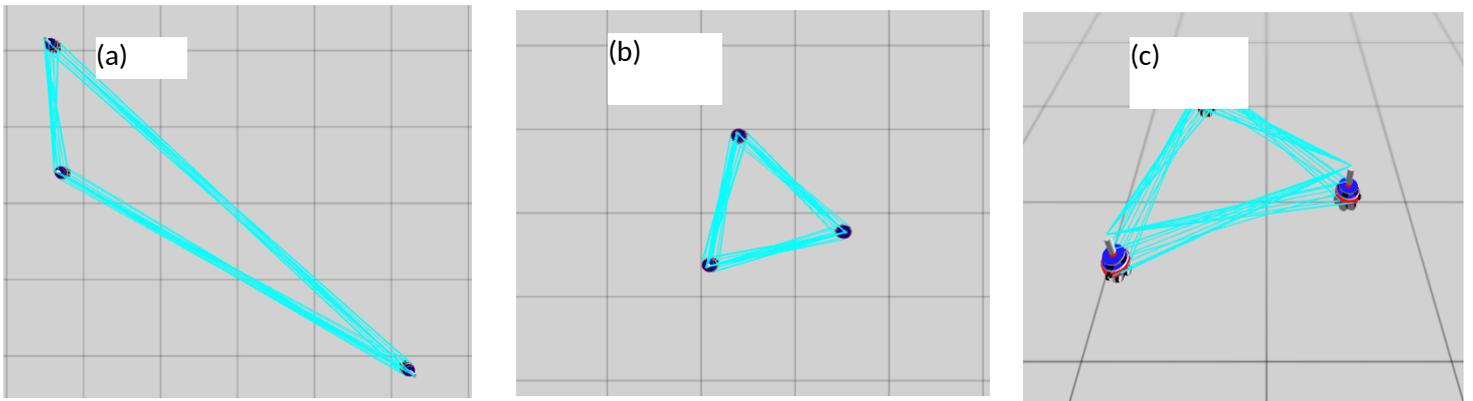


Figure 5.23: 3 robots formant un triangle équilatéral

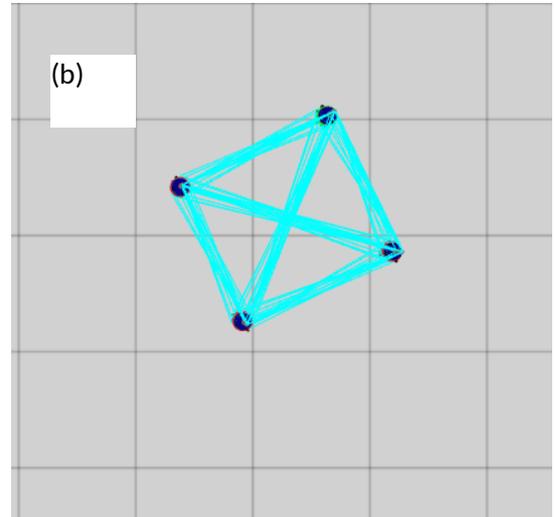
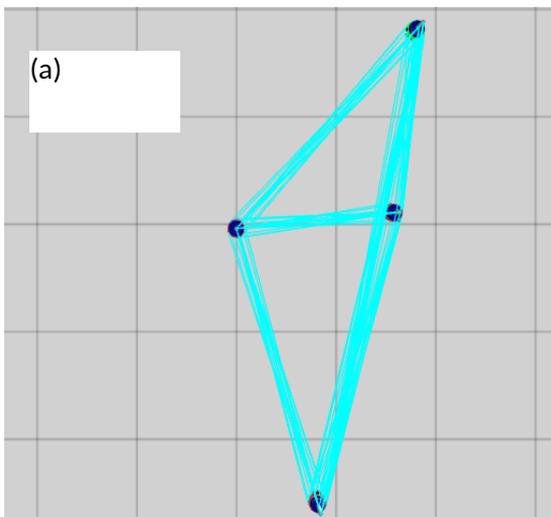


Figure 5.24 : 4 robots formant un carré.

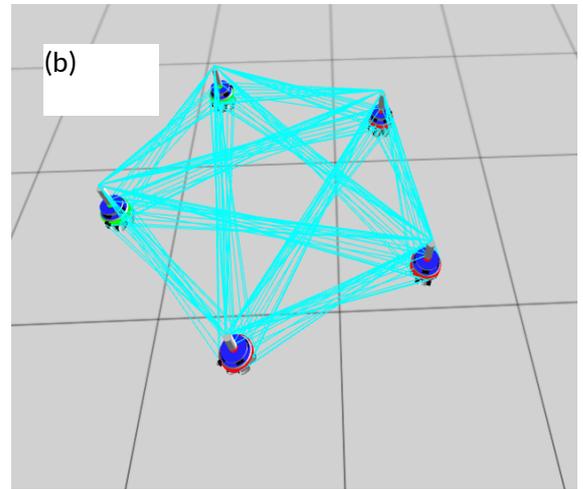
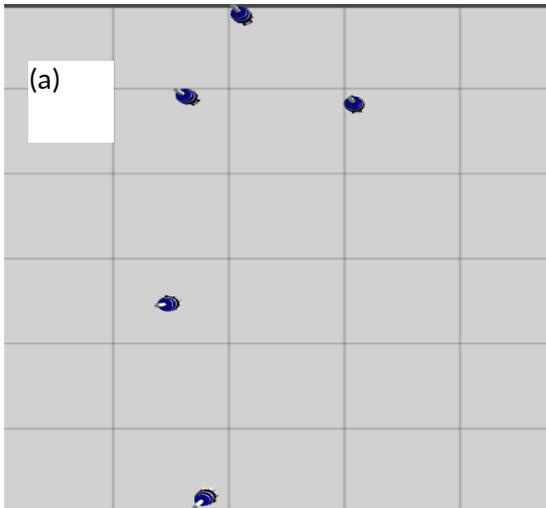


Figure 5.25: 5 robots formant un pentagone régulier.

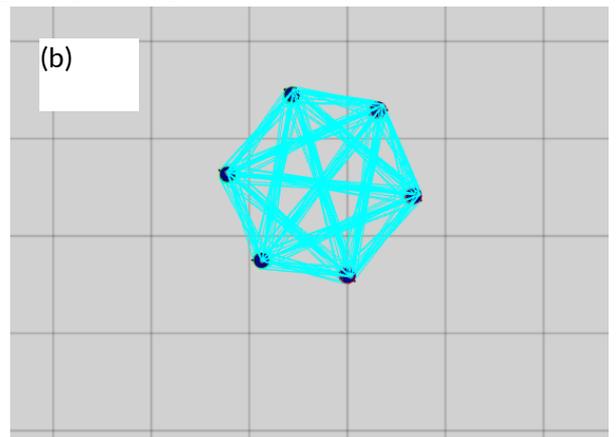
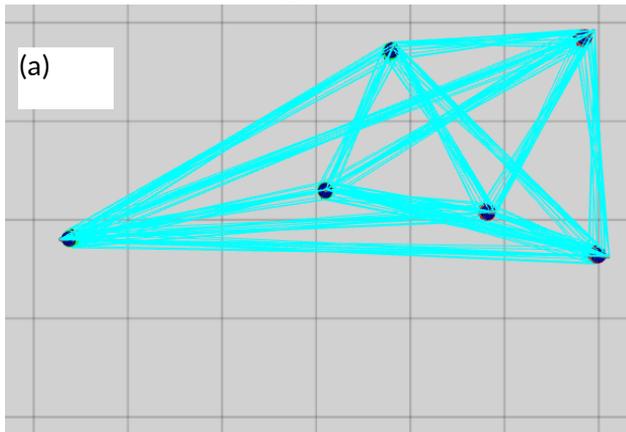


Figure 5.26 : 6 robots formant un hexagone régulier.

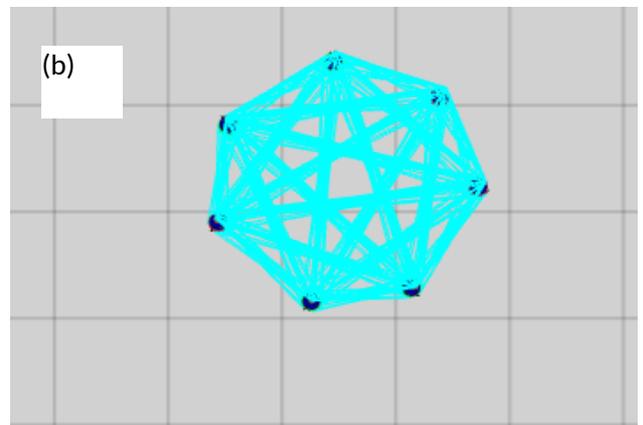
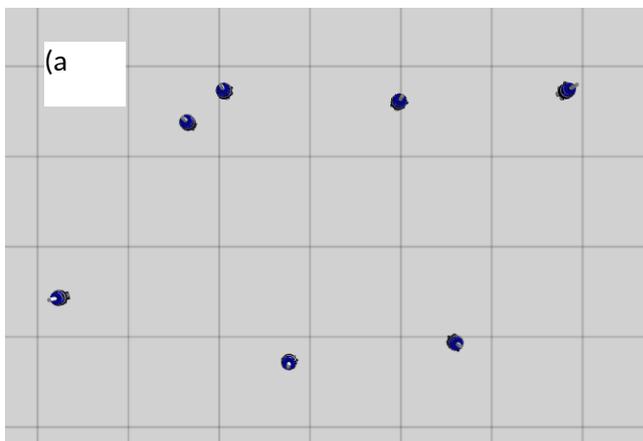


Figure 5.27 : 7 robots formant un heptagone régulier.

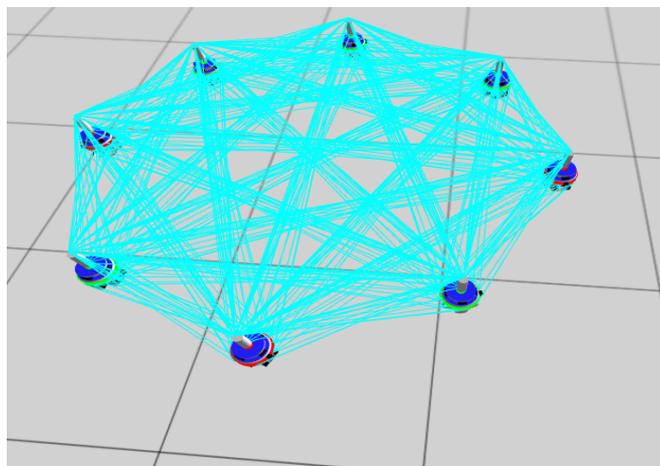
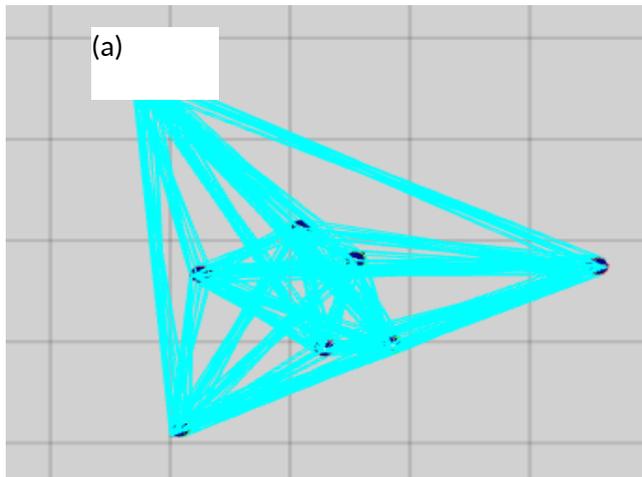


Figure 5.28: 8 robots formant un octogone régulier.

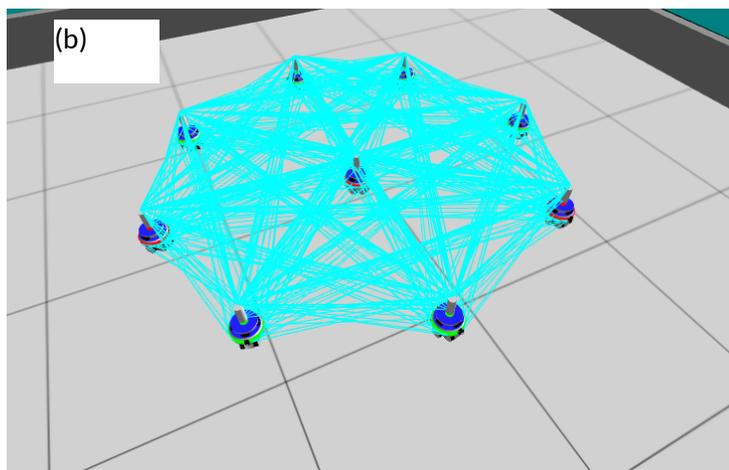
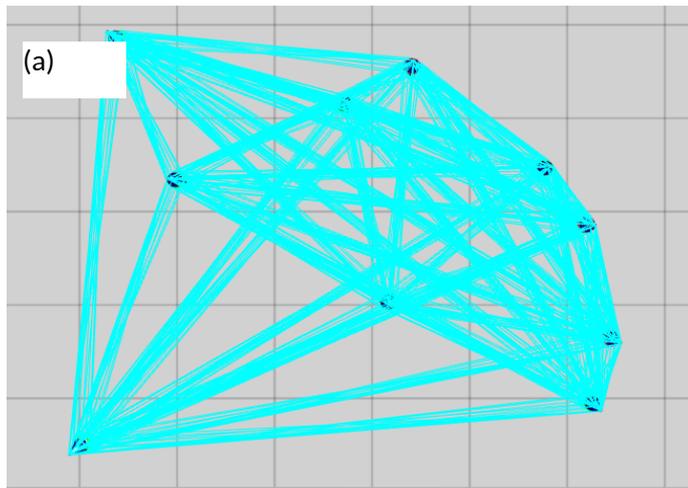


Figure 5.29 : 9 robots formant un octogone régulier avec un robot au centre.

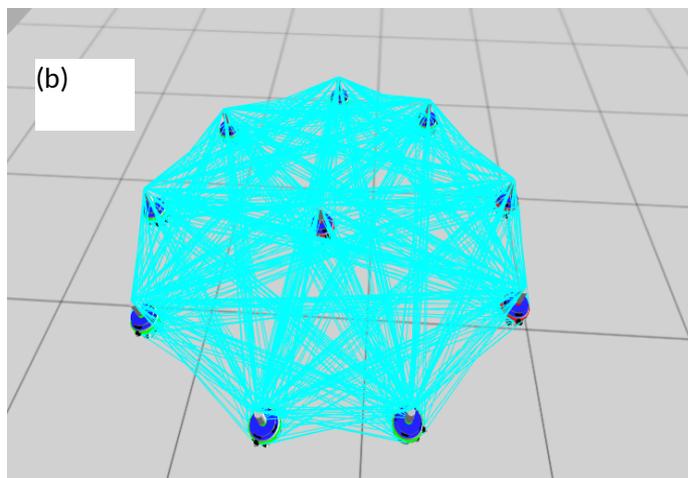
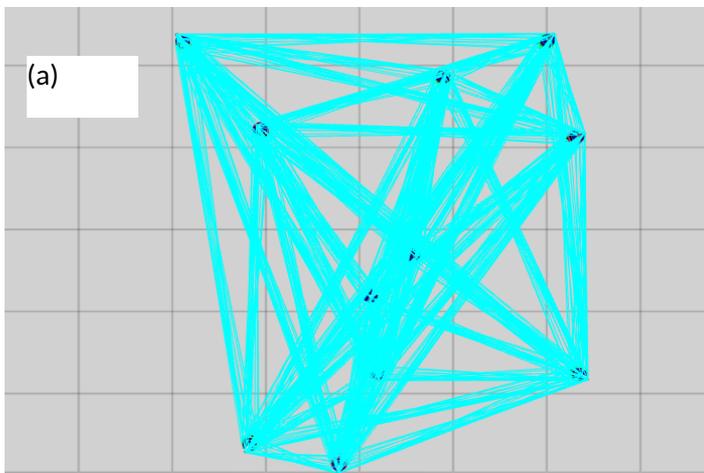


Figure 5.30 : 10 robots formant un nonagone régulier avec un robot au centre.

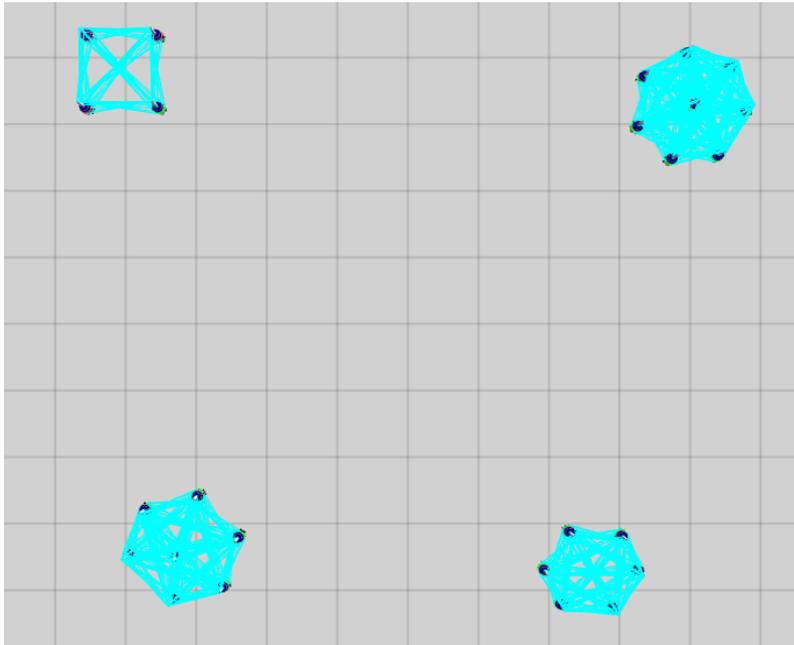


Figure 5.31: plusieurs forme au même espace (4 robots , 6 robots, 7 robots 10 robots).

Discussion :

On a présenté les résultats de notre modèle en utilisant le simulateur ARGOS. On remarque que la forme finale dépend du nombre de robot utilisé :

Les figures montrent :

- ◆ Un triangle équilatéral est formé par trois robots essaims (voir figure 5.14 et 5.23).
- ◆ Un carré équilatéral par quatre robots essaims (5.15 et 5.24)
- ◆ Un pentagone et un hexagone sont réalisés simultanément par 5 robots (5.16 et 5.25).
c) et 6 robots (5.17 et 5.26).
- ◆ Et les formes heptagone et octogonale sont construites par 7 robots (5.18 et 5.27) et 8 robots (5.19 et 5.28)

Notre modèle peut également réaliser d'autres configurations : dans lesquelles un robot peut être positionné au centre de la forme géométrique régulière en cours de formation.

- ◆ Des situations telles que celles obtenues sont basées sur l'équilibre des forces virtuelles principalement influencées par la position réelle des robots. A partir de 9 robots..
Figure 5.20 et 5.29 , 5.21 et 5.30 .
- ◆ On a pu obtenir plusieurs formes dans la même configuration, mais on respectant une distance pour éviter les connections entre les robots des autres groupes. (5.22 et 5.31)

5.2 Tests:

Pour bien comprendre la convergence de notre système, on a exécuter notre programme pour 6 foot-bot (rb0, rb1, rb2, rb3, rb4 et rb5). L'axe des X représente l'évolution du temps et l'axe des Y, les vitesse $V_i(t)$, pour les deux types de cameras : omnidirectionnel et range and bearing

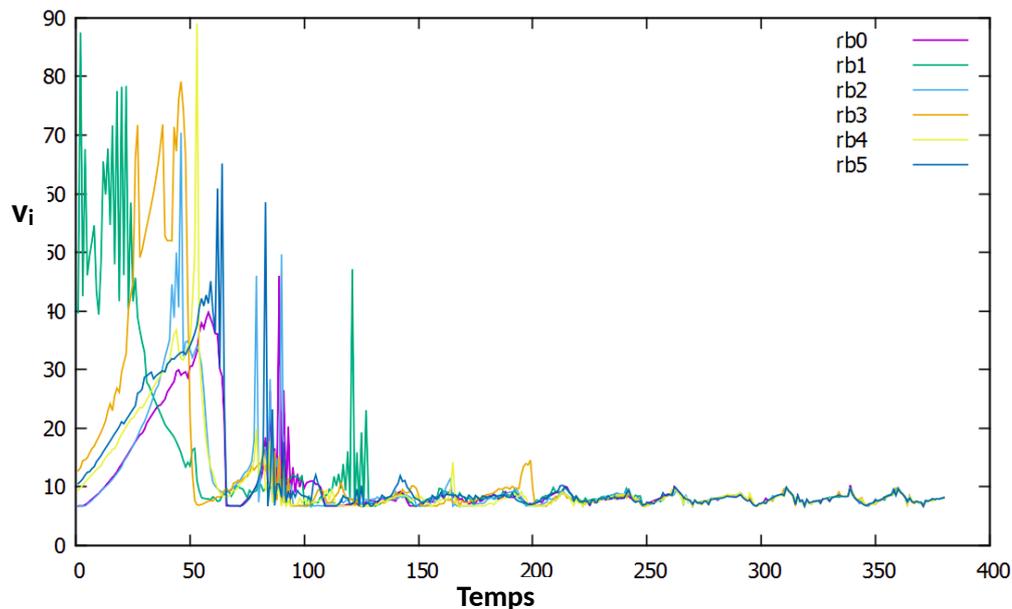


Figure 5.32 : Changement de vitesse $v_i(t)$ pour les 6 foot-bot (rb0, rb1, rb2, rb3, rb4 et rb5) en fonction du temps. Camera omnidirectionnel.

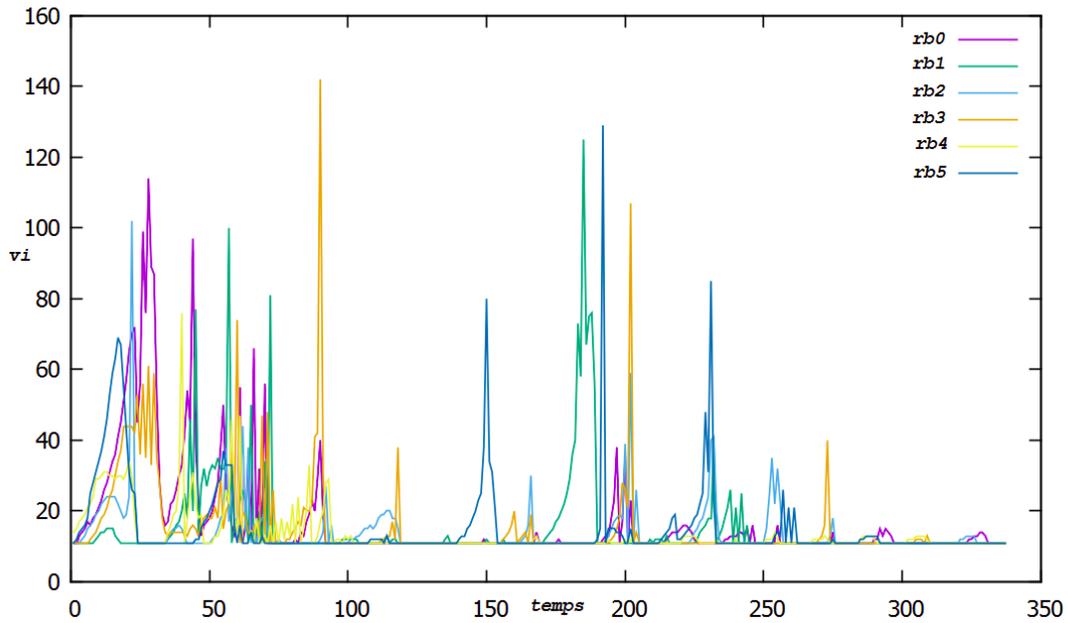


Figure 5.33: Changement de vitesse $v_i(t)$ pour les 6 foot-bot (rb0, rb1, rb2, rb3, rb4 et rb5) en fonction du temps : Capture range and bearing.

Discussion :

Pour $k\omega = 1,5$ la figure (5.32 et 5.33) présentent une analyse du taux de changement de la vitesse linéaire du robot, v_i , au cours d'une formation complète d'hexagone. Les chiffres montrent que (v_i) sont stabilisés pour tous les robots à partir du pas de temps ($t = 270$ pour le capture RAB) et ($t = 190$ pour omnidirectionnel_camera), à ce moment $\hat{\mathbf{a}}_i$ tendent vers zéro et (v_i) a une valeur minimale stable, Cela signifie que les robots ont atteint une forme spécifique.

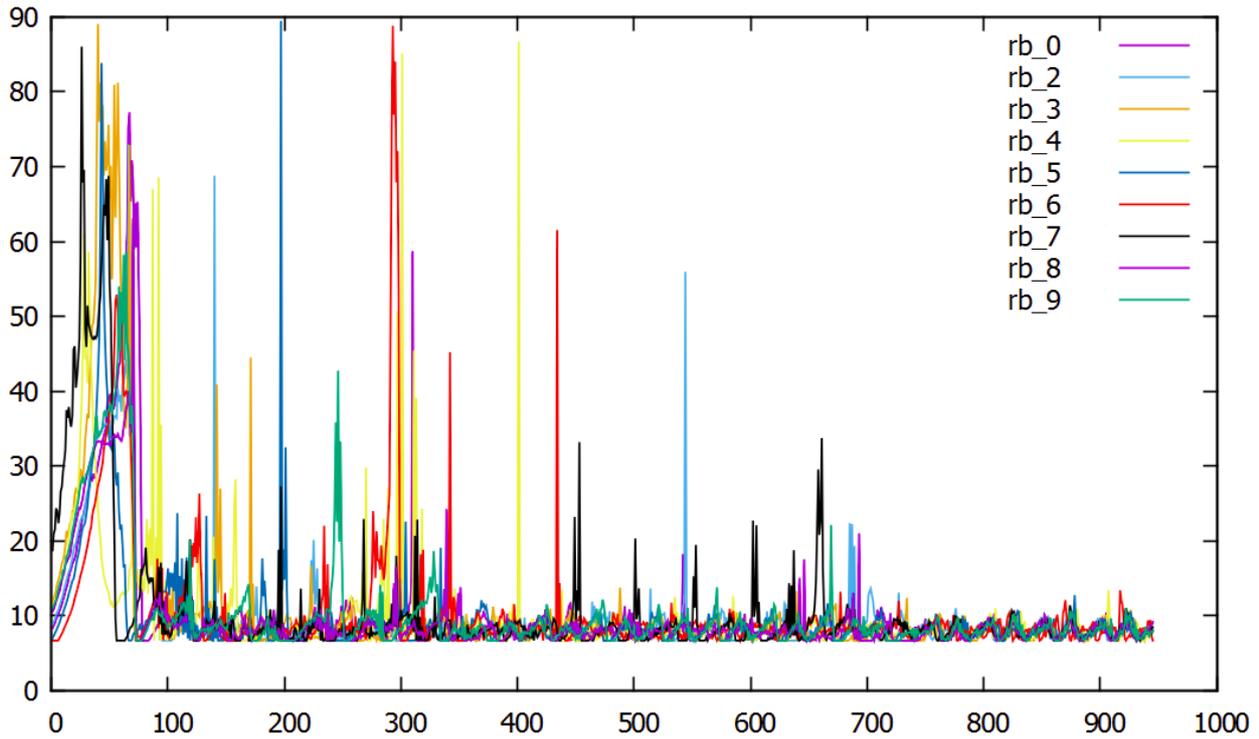


Figure 5.34: Changement de vitesse $v_i(t)$ pour les 10 foot-bot (rb0, rb1, rb2, rb3, rb4 et rb5, rb6, rb7, rb8, rb9) en fonction du temps. Camera omnidirectionnel.

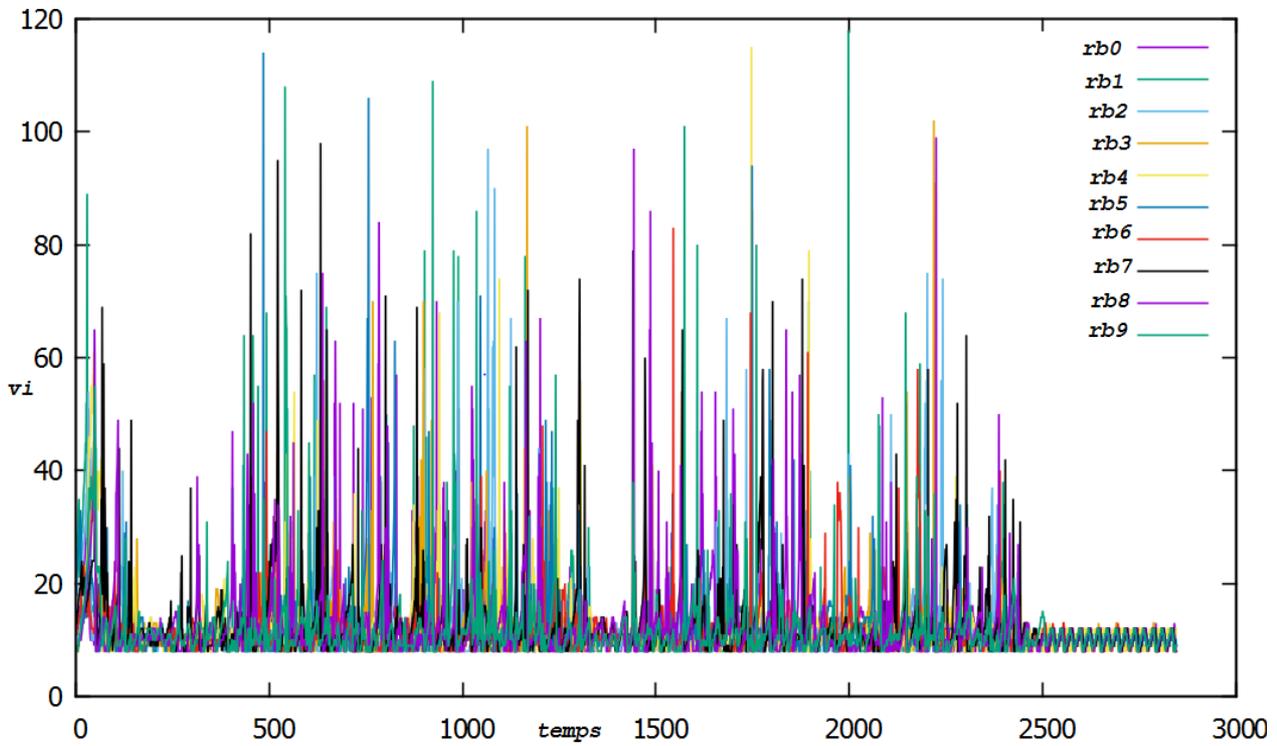


Figure 5.35: Changement de vitesse $v_i(t)$ pour les 10 foot-bot (rb0, rb1, rb2, rb3, rb4 et rb5, rb6, rb7, rb8, rb9) en fonction du temps : Capture range and bearing.

Discussion :

Pour $k\omega = 1,5$, la figure (5.34 et 5.35) présentent une analyse du taux de changement de la vitesse linéaire du robot v_i , au cours d'une forme nonagone régulier avec un robot au centre (10 robots). Les chiffres montrent que (v_i) sont stabilisés pour tous les robots à partir du pas de temps ($t = 270$ pour le capture RAB) et ($t = 700$ pour omnidirectionnel_camera), à ce moment (\hat{a}_i) tendent à zéro et (v_i) a une valeur minimale stable, Cela signifie que les robots ont atteint une forme spécifique.

Dans ce cas, nous avons réalisé des tests pour calculer la vitesse moyenne de différentes formes (un carré, un pentagone régulier, heptagone régulier, octogone régulier avec un robot au centre, et 10 robots qui formes un nonagone régulier avec un robot au centre) et on a fixé les mêmes valeurs pour tous tests : $k\omega=1.5$ et $v_{max}=100s$.

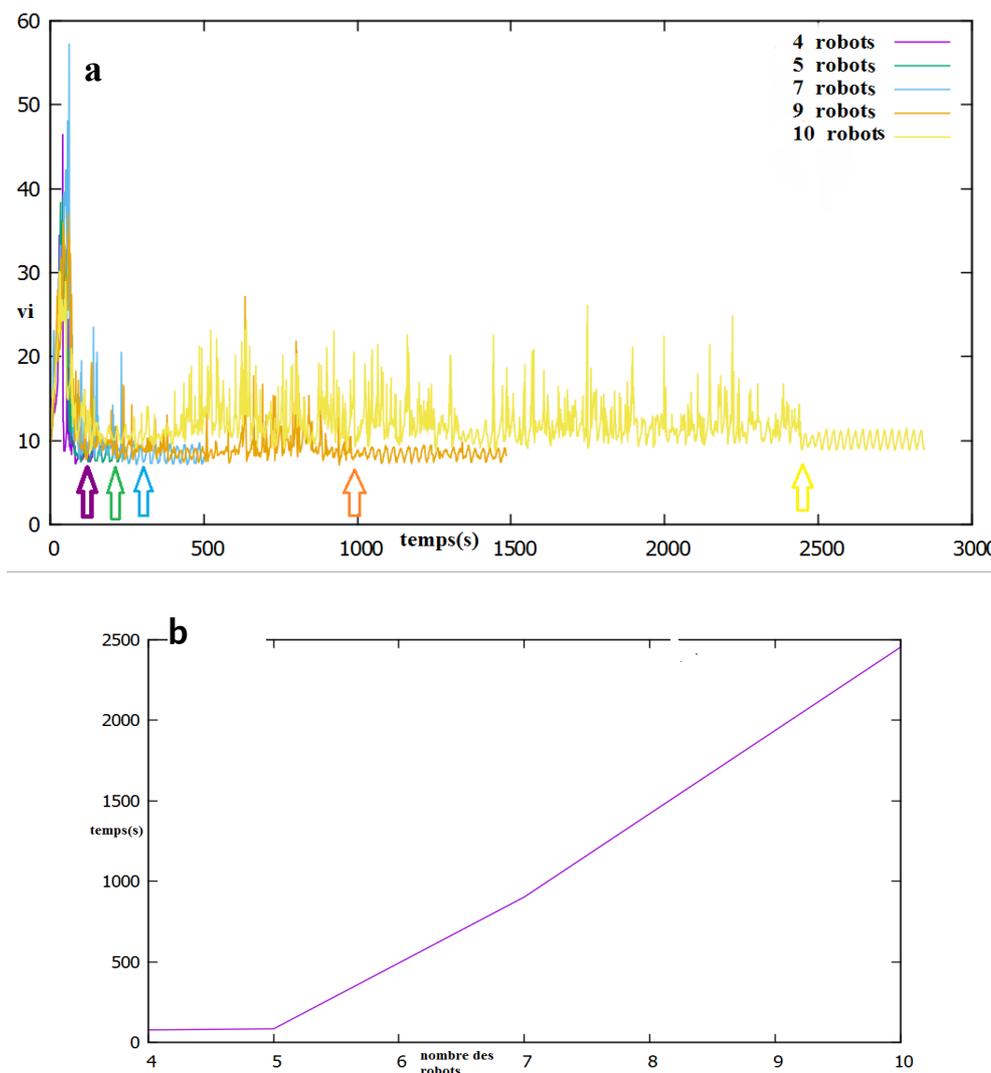


Figure 5.36: temps de stabilité de chaque forme différent avec RAB.

La figure 36 a, présente le temps de stabilité de chacune des formes illustrées, Où on voit que : les 4 robots se stabilisent au temps=77 s , 5 robots au temps=84 s, 7 robots au temps=245 s ,9 robots au temps=903s et 10 robots au temps=2453 s .

La figure 38 b présente la courbe d'évolution en fonction du nombre de robot.

Discussion :

- Le nombre de robot influe sur le temps de stabilité de la forme souhaitée d'une manière exponentielle.
- Le type de capteur aussi influe sur le temps de stabilité et la performance (capteur omnidirectionnel ou camera)

5.3 Conclusion

Nous avons présenté dans ce chapitre les tests de simulation de notre approche proposée dans le chapitre précédent. Pour cela on a testé deux types de capteurs omnidirectionnels ou camera en jouant sur le nombre de robots pour réaliser des formes bien déterminées.

Notre modèle de contrôle de la formation de robots en essaim qui est basé sur une connectivité viscoélastique virtuel entre les voisins. Le modèle peut réaliser dynamiquement des formations géométriques régulières telles que des carrés, des pentagones et des octogones en utilisant uniquement les estimations de distance et d'orientation entre voisins. Le modèle est entièrement décentralisé et évolutif, ce qui permet au nombre de robots impliqués de former de manière dynamique une configuration géométrique de base régulière souhaitée. Le modèle a été implémenté dans une version simulée du véritable robot foot-bot utilisant le simulateur ARGoS.



Conclusion générale

La robotique en essaim est une recherche très intéressante et ambitieuse dans un sous-domaine, qui cherche principalement à intégrer la théorie derrière les essaims dans la nature à des systèmes multi-robotiques.

Les modèles de robotique en essaim sont fondamentalement inspirés des comportements collectifs d'animaux sociaux tels que les oiseaux, les fourmis et les abeilles. Ils sont généralement appliqués dans les études traitant de problèmes de tâches collectives. À titre d'exemple, les schémas de formation et les schémas d'agrégation spécifiquement auto-organisés font partie des problématiques complexes qui intéressent la littérature sur la robotique en essaim.

Cette problématique a été abordée dans ce mémoire en concevant des contrôleurs synthétisés pour des comportements collectifs significatifs en robotique d'essaims. En particulier, les contrôleurs ont été appliqués avec succès à l'étude des tâches de formation de modèles d'auto-organisation robotique en essaim.

Les contrôleurs sont principalement basés sur un modèle viscoélastique artificiel qui s'inspire des propriétés biomécaniques - telles que la viscosité et l'élasticité. Nous avons montré que le contrôleur de base est capable de réaliser des formations géométriques simples au sein de deux types de plates-formes robotiques en essaim: les foot-bots avec succès.

Notre modèle de contrôle est capable de réaliser diverses configurations en utilisant le modèle d'interaction viscoélastique géométrique régulière telles que des triangles, des carrés et des pentagones avec deux types de capteurs. Nous avons réalisé une comparaison afin d'améliorer les performances de notre système.

Nous envisageons appliquer notre modèle pour simuler des formes plus complexes en utilisant soit des robots identiques ou des robots différents.

Bibliographie

Ando Hideki et al. “Distributed memoryless point convergence algorithm for mobile robots with limited visibility”. Dans: IEEE Transactions on Robotics and Automation 15.5 (1999), pp. 818–828.

Belkacem Khaldi, Fouzi Harrou, Foudil Cherif , Ying Sun , « Self-organization in aggregating robot swarms: A DW-KNN topological approach » ,BioSystems Journal, 165 (2018) 106–121 .

Belkacem Khaldi , Fouzi Harrou , Foudil Cherif , Ying Sun , « Monitoring a robot swarm using a data-driven fault detection approach » Robotics and Autonomous Systems 97 (2017) 193–203.

Carlo Pinciroli et al. “ARGoS: a modular, multi-engine simulator for heterogeneous swarm robotics”. Intelligent Robots and Systems (IROS), 2011 IEEE/RSJ International Conference on. IEEE. 2011, pp. 5027–5034.

Carlo Pinciroli et al. “ARGoS: a modular, parallel, multi-engine simulator for multi-robot systems”. Swarm intelligence 6.4 (2012), pp. 271–295.

Carlo Pinciroli, Vito Trianni, Rehan O’Grady, Giovanni Pini, Arne Brutsch , Manuele Brambilla , Nithin Mathews , Eliseo Ferrante , Gianni Di Caro , Frederick Ducatelle , Mauro Birattari , Luca Maria Gambardella et Marco Dorigo , « ARGoS: a modular, parallel, multi-engine simulator for multi-robot systems » Swarm Intell (2012) 6:271–295, DOI 10.1007/s11721-012-0072-5 .

Cheng-Ming Chuong and Michael K Richardson. “Pattern formation today”. The International journal of developmental biology 53.5-6 (2009), p. 653.

Donghwa Jeong et Kiju Lee. “Dispersion and Line Formation in Artificial Swarm Intelligence”. ArXiv preprint arXiv:1407.0014 (2014).

Eliseo Ferrante , Ali Emre Turgut , Cristián Huepe , Alessandro Stranieri , Carlo Pinciroli et Marco Dorigo , « Self-Organized Flocking with a Mobile Robot Swarm: a Novel Motion Control Method ». IRIDIA – Technical Report Series , ISSN 1781-3794.

Gauci Melvin et al. “Evolving aggregation behaviors in multi-robot systems with binary sensors”. Distributed autonomous robotic systems. Springer, 2014, pp. 355–367.

Gerardo Beni et Jing Wang. “Swarm intelligence in cellular robotic systems”. Robots and Biological Systems: Towards a New Bionics? Springer, 1993, pp. 703–712.

George F Young et al. “Starling flock networks manage uncertainty in consensus at low cost”. PLoS computational biology 9.1 (2013), e1002894.

Guo Hongliang, Yan Meng, et Yaochu Jin. “Swarm robot pattern formation using a morphogenetic multi-cellular based self-organizing algorithm”. Robotics and Automation (ICRA), 2011 IEEE International Conference on. IEEE. 2011, pp. 3205–3210.

Hiroaki Kitano. “Biological robustness”. Nature Reviews Genetics 5.11 (2004), pp. 826–837.

Iain D Couzin et al. “Collective memory and spatial sorting in animal groups”. Journal of Theoretical Biology 218.1 (2002), pp. 1–11.

Iñaki Navarro and Fernando Matía. “An introduction to swarm robotics”. ISRN Robotics 2013 .

Maria Carmela De Gennaro et Ali Jadbabaie. “Decentralized control of connectivity for multi-agent systems”. Decision and Control, 2006 45th IEEE Conference on. IEEE. 2006, pp. 3628–3633.

Marco Dorigo, Mauro Birattari, et al. “Swarm intelligence”. Scholarpedia 2.9(2007), p. 146.

Nikolaus Correll et Alcherio Martinoli. “Modeling and designing self-organized aggregation in a swarm of miniature robots”. The International Journal of Robotics Research 30.5 (2011), pp. 615–626.

Noam Gordon, Israel A Wagner, and Alfred M Bruckstein. “Gathering multiple robotic agents with limited sensing capabilities”. International Workshop on Ant Colony Optimization and Swarm Intelligence. Springer. (2004), pp. 142–153.

Onur Soysal et Erol Sahin. “Probabilistic aggregation strategies in swarm robotic systems”. Swarm Intelligence Symposium, 2005. SIS 2005. Proceedings 2005 IEEE. IEEE. 2005, pp. 325–332.

Roger SERRA, Julien OLIVIER ,”Apport de l’optimisation par essaims particuliers pour la détection de défauts d’une poutre à partir de ses fréquences propres” , INSA Centre Val de Loire, Laboratoire de Mécanique et Rhéologie, 3 rue de la chocolaterie, 41000 Blois, France ,INSA Centre Val de Loire, Laboratoire d’Informatique, 3 rue de la chocolaterie, 1000 Blois, France ,(2017) .

Rodolphe Charrier. “ L’intelligence en essaim sous l’angle des systèmes complexes : étude d’un système multi-agent réactif à base d’itérations logistiques couplées “. thèse doctorat ,Université Nancy II, (2009). Français.

Sahin Erol et al. “Swarm Robotics.” Swarm intelligence 1 (2008), pp. 87–100.

Shiladitya Banerjee. “Cell Mechanics From cytoskeletal dynamics to tissue-scale mechanical phenomena”. PhD thesis. Syracuse University, (2013).

Veysel Gazi et Kevin M Passino. “Stability analysis of swarms”. IEEE Transactions on Automatic Control 48.4 (2003), pp. 692–697.

Veysel Gazi et al. “Aggregation, foraging, and formation control of swarms with non-holonomic agents using potential functions and sliding mode techniques”. Turkish Journal of Electrical Engineering & Computer Sciences 15.2 (2007), pp. 149–168.

William M Spears et al. “Distributed, physics-based control of swarms of vehicles”. *Autonomous Robots* 17.2-3 (2004), pp. 137–162.

Yousef Jamali, Mohammad Azimi, et Mohammad RK Mofrad. “A sub-cellular viscoelastic model for cell population mechanics”. *PLoS One* 5.8 (2010), e12097.

¹ Extrait de 26/1/2019 , <http://playerstage.sourceforge.net>

² Extrait de 26/1/2019 , <http://gazebosim.org/>

³ Extrait de 26/1/2019 , <http://iridia.ulb.ac.be/argos/>

⁴ Extrait de 26/1/2019 , <http://home.gna.org/enki/>

⁵ Extrait de 26/1/2019 , <http://www.coppeliarobotics.com/>

⁶ Extrait de 26/1/2019 , www.cs.cmu.edu/robsoccer/ubersim

⁷ Extrait de 26/1/2019 , <http://sourceforge.net/apps/mediawiki/usarsim/>