

*République Algérienne Démocratique et Populaire*  
*Ministère de l'Enseignement supérieur et de la recherche scientifique*  
*Ecole Nationale Polytechnique.*



**DEPARTEMENT D'ELECTRONIQUE**

**THESE DE DOCTORAT EN ELECTRONIQUE**

Présentée par :

**Ouahiba Azouaoui**

Magister en Cybernétique (CDTA)

**PLANIFICATION ET CONTROLE DES  
COMPORTEMENTS EN GROUPE DES SYSTEMES  
ROBOTIQUES AUTONOMES**

**Soutenue le 20 Septembre 2006**

**Devant le Jury d'examen composé de :**

Président :	<b>Dr. Mohamed Mehenni</b>	Professeur	ENP
Examineurs :	<b>Dr. Fares Boudjema</b>	Professeur	ENP
	<b>Dr. Youcef Smara</b>	Professeur	USTHB
	<b>Dr. Djamel Benazouz</b>	Professeur	Univ. Boumerdes
Rapporteurs :	<b>Dr. Ahcène Farah</b>	Professeur	ENP
	<b>Dr. Karim Achour</b>	Directeur de recherche	CDTA
Invité :	<b>Dr. Mustapha Hamerlain</b>	Directeur de recherche	CDTA



# Avant-Propos

Les travaux présentés dans ce manuscrit ont été effectués à la Division Robotique et Productique du Centre de Développement des Technologies Avancées au sein de l'équipe Structure et Dynamique des Robots.

Je tiens à remercier mes Directeurs de thèse, Ahcène Farah et Karim Achour, pour les conseils, les critiques et la patience qui m'ont permis de mener cette thèse à son terme.

Je suis reconnaissante à Monsieur Mohamed Mehenni pour avoir présidé le jury de cette thèse. J'exprime également ma reconnaissance à Messieurs Djamel Benazouz, Fares Boudjema, Mustapha Hamerlain et Youcef Smara pour l'intérêt qu'ils ont bien voulu porter à ce travail et pour leur participation au jury en tant qu'examineurs.

Pour avoir relu le manuscrit de thèse, je remercie particulièrement Houria Yazid ainsi que Leila Sahli et Samia Ourari.

Pour leur disponibilité, je remercie Hassen Bendifellah, Mohamed Kadri et Rachid Tiar. Enfin, je remercie mes collègues pour leur sympathie et leurs encouragements.



## ملخص:

في عدة تطبيقات معقدة، استعمال عدة أنظمة آلية ذات تحكم ذاتي (*Robots*) أصبحت ضرورة لتحقيق مهام مختلفة كتنقل الأشياء الثقيلة و المضايقة بأقل كلفة و أكثر فعالية. وبالتالي، لابد على هذه الأنظمة أن تكون لديها الاستطاعة على إنشاء أشكال هندسية تتلاءم مع شكل الشيء المعني متنوع بانتقال هذه التشكيلة نحو هدفها. لهذا، على الأنظمة الآلية أن تتعلم تجنب الاصطدام فيما بينهم و مع الحواجز. تم اقتراح طريقتين لتجنب الاصطدام و الانجذاب نحو الهدف. الأولى تعتمد على الشبكات العصبية حيث تم إنجاز شبكة عصبية غامضة *ArtMap*. بالفعل على كل نظام آلي تعلم استنتاج القاعدة الملائمة لتجنب الاصطدام و الانجذاب نحو الهدف. الطريقة الثانية تعتمد على التعلم بالتوطيد حيث كل نظام آلي يتعلم تجنب الأنظمة الأخرى انطلاقاً من تفاعله مع محيطه مع إصابة هدفه. طريقة التعليم هذه تمكن الآليات من الاستفادة من تجربة الآخرين. هذه الطريقة تزود الآليات باستطاعة اكتساب تصرف لتجنب الاصطدام فيما بينها و الانجذاب نحو الهدف. انطلاقاً من تصرفات أولية باستعمال تعلم بالمحاولة و الخطأ. بعد اكتساب هذا التصرف، على الآليات تعلم إنجاز أشكال هندسية و الانتقال محتفظين على هذه التشكيلة انطلاقاً من تفاعلها مع المحيط. على هذه الطريقة الثالثة تزويد الآليات باستطاعة اكتساب التنقل جماعة انطلاقاً من تصرفات أولية مختارة بمهارة. نتائج هذه الطرق تبين إمكانية الآليات على الانتقال جماعة في محيطات فعالة. هذا الانتقال مع تصرفه التعاوني يجعل الآليات قادرة على العمل معاً لإنجاز المهمة المقصودة بنجاح.

مفاتيح الكلمات: أنظمة آلية ذات تحكم ذاتي، تنقل جماعة، تنقل في تشكيلة، تجنب الاصطدام، تصرفات أولية، تعلم بالتوطيد، شبكة عصبية غامضة *ArtMap*.

## Résumé :

Dans diverses applications complexes, l'utilisation de plusieurs Systèmes Robotiques Autonomes (SRAs) devient une nécessité afin de réaliser différentes tâches telle que le transport d'objets lourds et encombrants avec un moindre coût et plus d'efficacité. Ainsi, ils doivent avoir la capacité de former la figure géométrique correspondant à la forme de l'objet en question suivie par le déplacement de cette formation vers son objectif. Pour ce faire, les SRAs doivent d'abord apprendre à éviter les collisions entre eux et avec les obstacles. Deux approches d'évitement de collisions et d'attraction vers la cible ont donc été proposées. La première se base sur les réseaux de neurones où un réseau neuro-flou *ArtMap* a été développé. En effet, chaque robot doit apprendre, en utilisant le système sensoriel *LOCISS* (*LOcally Communicable Infrared Sensory System*) pour chaque relation de position à déduire la règle appropriée d'évitement de collisions et d'attraction vers la cible. La seconde approche se base sur le paradigme de l'apprentissage par renforcement où chaque robot apprend comment éviter les autres robots à partir de son interaction avec l'environnement tout en atteignant son but. Ce processus d'apprentissage permet aux robots de bénéficier de l'expérience des autres. Il fournit aux robots la capacité d'acquérir le comportement d'évitement de collisions et d'attraction vers la cible entre plusieurs robots à partir de comportements élémentaires en utilisant un apprentissage par essai et erreur. Une fois le comportement d'évitement de collisions et d'attraction vers la cible est acquis par les robots, ceux-ci doivent apprendre à former des figures géométriques et à naviguer en formation à partir de leur interaction avec l'environnement. Cette troisième approche doit fournir aux SRAs la capacité d'acquérir la navigation en groupe, par apprentissage, à partir de comportements élémentaires choisis judicieusement. Des résultats de simulation des approches proposées sont présentés et montrent l'aptitude des robots à naviguer en groupe et en formation dans des environnements dynamiques. Cette navigation avec son comportement coopératif rend les SRAs capables de travailler ensemble afin d'accomplir la tâche désirée avec succès.

Mots-clés: Systèmes Robotiques Autonomes (SRAs), Navigation en groupe, Navigation en formation, Evitement de collisions, Comportements élémentaires, Apprentissage par renforcement, réseaux de neurones flou *ArtMap*.

## Abstract :

In several complex applications, the use of multiple Autonomous Robotic Systems (ARSs) becomes necessary to achieve different tasks such as transport of heavy and large objects with less cost and more efficiency. Therefore, they must have the capability to form geometric figures and navigate without collisions while maintaining the formation. At first, the ARS must learn how to avoid collisions among them and with obstacles. Two approaches are suggested for this purpose. The first one is based on neural networks where a fuzzy *ArtMap* neural network based collision avoidance approach is developed. Indeed, each robot must learn, using *LOCally Communicable Infrared Sensory System* (*LOCISS*), for each positional relation, to deduce the appropriate rule to avoid collisions. The second one is based on reinforcement learning paradigm where each robot learns how to avoid the others from its interaction with the environment while reaching its target. This learning process allows ARS to benefit from the experience of the others using the same value base. This approach provides ARS with capability to acquire the collision avoidance behavior among several ARS from elementary behaviours by learning with trial and error search. Once the collision avoidance behaviour is acquired by the ARSs, they have to learn how to form a figure and navigate in formation from their interaction with the environment. This third approach must provide ARS with capability to acquire the group navigation among several ARS from elementary behaviours by learning. Then, simulation results display the ability of the suggested approaches to provide ARS with capability to navigate in a group formation in dynamic environments. This approach with its cooperative behavior makes ARS able to work together to successfully fulfill the desired task.

Keywords: Autonomous Robotic Systems (ARS), Group Navigation, Navigation in Formation, Collision Avoidance, Elementary Behaviors, Reinforcement Learning, Fuzzy *ArtMap* Neural Networks.



# Table des Matières

INTRODUCTION GENERALE .....	1
-----------------------------	---

## Chapitre I

### Robotique Collective : *Etat de l'art et synthèse*

I.1. INTRODUCTION .....	4
I.2. INTELLIGENCE COLLECTIVE .....	5
I.2.1. DEFINITION .....	5
I.2.2. NOTION D'EMERGENCE .....	5
I.2.3. INTELLIGENCE COLLECTIVE EN ROBOTIQUE .....	6
<b>I.3. AUTO-ORGANISATION.....</b>	<b>7</b>
I.3.1. QUELQUES DEFINITIONS.....	7
I.3.2. CARACTERISTIQUES DES SYSTEMES AUTO-ORGANISATEURS .....	7
I.3.3. AVANTAGES DE L' AUTO-ORGANISATION .....	8
I.3.4. AUTO-ORGANISATION CHEZ LES INSECTES .....	8
I.3.4.1. <i>Différentes formes d'organisation des sociétés animales</i> .....	8
I.3.4.1.1. Conditions de formation du groupe social.....	8
I.3.4.1.2. Persistance du groupe durant le déplacement.....	8
I.3.4.1.3. Autres comportements .....	9
I.3.4.2. <i>Fonctionnement collectif</i> .....	9
I.3.5. SOCIETE ANIMALE ET GROUPES DE ROBOTS MOBILES .....	10
<b>I.4. SYSTEMES ROBOTIQUES AUTONOMES.....</b>	<b>10</b>
I.4.1. ARCHITECTURE SUBSOMPTION.....	10
I.4.2. NAVIGATION AUTONOME .....	12
I.4.2.1. <i>Approches classiques</i> .....	13
I.4.2.2. <i>Approches actuelles</i> .....	13
<b>I.5. SYSTEMES MULTI-ROBOTS .....</b>	<b>13</b>
<b>I.6. ROBOTIQUE COLLECTIVE.....</b>	<b>15</b>
I.6.1. DEFINITION .....	16
I.6.2. LOGIQUE DE FONCTIONNEMENT DE L' INTELLIGENCE COLLECTIVE.....	16
I.6.3. POSITION DU PROBLEME .....	17
I.6.4. ARCHITECTURE DE GROUPE .....	18
I.6.4.1. <i>Centralisation/décentralisation</i> .....	18
I.6.4.1.1. Systèmes centralisés.....	18
I.6.4.1.2. Systèmes décentralisés.....	19

I.6.4.2. <i>Différentiation</i> .....	20
I.6.4.3. <i>Structures de communication et interaction des robots</i> .....	21
I.6.4.3.1. Interaction via l'environnement .....	21
I.6.4.3.2. Interaction via la perception .....	21
I.6.4.3.3. Interaction via la communication .....	22
I.6.4.3.4. De l'interaction de l'agent à la "cognition" et au comportement .....	22
I.6.4.4. MODELISATION DES AUTRES AGENTS .....	23
I.6.4.5. ARCHITECTURES DE CONTROLE .....	23
I.6.4.5.1. <i>Architectures de contrôle réactives (ne pas penser mais agir)</i> .....	23
I.6.4.5.2. <i>Architectures de contrôle délibératives (penser puis agir)</i> .....	24
I.6.4.5.3. <i>Architectures de contrôle hybrides (penser et agir indépendamment en parallèle)</i> .....	24
I.6.4.5.4. <i>Architectures de contrôle comportementales (penser la façon d'agir)</i> .....	25
I.6.4.5.5. <i>Comparaison et discussion</i> .....	26
I.6.5. CONFLITS DE RESSOURCES .....	27
I.6.6. ORIGINE DE LA COOPERATION .....	28
I.6.7. COOPERATION .....	28
I.6.7.1. <i>Méthodes de coopération</i> .....	29
I.6.7.1.1. Regroupement et Multiplication.....	29
I.6.7.1.2. Communication .....	29
I.6.7.1.3. Spécialisation.....	30
I.6.7.1.4. Collaboration .....	30
I.6.7.1.5. Coordination d'actions.....	30
I.6.7.1.6. Résolution de conflits par arbitrage et négociation.....	30
I.6.7.2. <i>Avantages de la coopération</i> .....	31
I.6.8. APPRENTISSAGE .....	31
I.6.9. QUELQUES EXPERIENCES DE LA ROBOTIQUE COLLECTIVE .....	32
I.6.9.1. <i>Combinaisons de primitives comportementales</i> .....	32
I.6.9.2. <i>Problèmes géométriques</i> .....	33
I.6.9.2.1. Planification de chemins de plusieurs robots .....	33
I.6.9.2.2. Problèmes de formation du groupe et de formation en marche.....	33
<b>I.7. CONCLUSION .....</b>	<b>34</b>

## Chapitre II

# Navigation des Systèmes Robotiques Autonomes dans un Environnement Dynamique : *Cas de deux robots*

<b>II.1. INTRODUCTION.....</b>	<b>36</b>
<b>II.2. DEFINITION DU PROBLEME A RESOUDRE.....</b>	<b>37</b>
II.2.1. TACHE CONSIDEREE .....	38
II.2.1.1. <i>Description de la tâche à réaliser</i> .....	38
II.2.1.2. <i>Etapas de l'apprentissage</i> .....	39



II.2.2. ARCHITECTURE, MOBILITE, ENVIRONNEMENT ET TRAITEMENT .....	40
II.2.2.1. Robots visés.....	40
II.2.2.2. Planification et contrôle de la navigation visés.....	41
II.2.2.3. Environnements dynamiques partiellement structurés .....	42
<b>II.3. NAVIGATION PAR LOCISS (LOCALLY COMMUNICABLE INFRARED SENSORY SYSTEM) .....</b>	<b>43</b>
II.3.1. DESCRIPTION DE LOCISS .....	43
II.3.2 DETECTION PAR LOCISS.....	44
II.3.3. COOPERATION DANS LES SYSTEMES MULTI-ROBOTS .....	44
II.3.3.1. Coordination d'action .....	45
II.3.3.2. Communication.....	45
<b>II.4. QUELQUES APPROCHES DE NAVIGATION.....</b>	<b>45</b>
<b>II.5. APPROCHE NEURONALE DE NAVIGATION DES SYSTEMES ROBOTIQUES AUTONOMES DANS UN ENVIRONNEMENT PARTIELLEMENT STRUCTURE .....</b>	<b>47</b>
II.5.1. REGLES PREDETERMINEES .....	47
II.5.1.1. Evitement d'un robot.....	48
II.5.1.2. Evitement d'un objet statique .....	49
II.5.2. APPROCHE PROPOSEE D'EVITEMENT DE COLLISIONS ET D'ATTRACTION VERS LA CIBLE BASEE SUR UN RESEAU DE NEURONES ARTMAP FLOU (RNAMF) .....	50
II.5.2.1. Architecture de contrôle utilisée.....	50
II.5.2.2. Description du système.....	50
II.5.2.3. Apprentissage par l'algorithme AMFS (ArtMap Floue Simplifiée).....	52
II.5.2.4. Entraînement de RNAMF.....	54
<b>II.6. CONCLUSION.....</b>	<b>54</b>

## Chapitre III

# Navigation des Systèmes Robotiques Autonomes dans un Environnement Dynamique : *Vers une navigation en groupe*

<b>III.1. INTRODUCTION .....</b>	<b>56</b>
<b>III.2. QUELQUES APPROCHES BASEES SUR L'AR .....</b>	<b>57</b>
<b>III.3. APPROCHE DE NAVIGATION BASEE SUR L'APPRENTISSAGE PAR RENFORCEMENT PROPOSEE.....</b>	<b>59</b>
III.3.1. ROBOT (L'AGENT) ET CAPTEURS .....	60
III.3.2. DESCRIPTION DU SYSTEME .....	60
III.3.2.1. Architecture de contrôle utilisée.....	61
III.3.2.2. Heuristiques .....	61
III.3.2.3. Comportements élémentaires.....	62
III.3.2.3.1. Comportement avancer .....	62

III.3.2.3.2. Comportement d'évitement.....	62
III.3.2.3.3. Actions générées .....	62
III.3.3. APPRENTISSAGE PAR INTERACTION .....	63
III.3.3.1. <i>Q-learning</i> .....	63
III.3.3.2. <i>Procédure d'apprentissage</i> .....	65
III.3.3.2.1. Perception de l'environnement.....	65
III.3.3.2.2. Sélection d'un comportement.....	65
III.3.3.2.3. Exécution du mouvement.....	66
III.3.3.2.4. Réception du renforcement (fonction de renforcement) .....	67
III.3.3.2.5. Mise à jour de la table des valeurs.....	67
III.3.3.3. <i>Description de l'algorithme Q_Learning utilisé</i> .....	67
III.3.3.3.1. Algorithme Q_Learning avec des tables de valeurs privées .....	67
III.3.3.3.2. Algorithme Q_Learning avec une table de valeurs commune .....	68
<b>III.4. CONCLUSION .....</b>	<b>69</b>

## Chapitre IV

# Comportements Intelligents d'un Groupe de Systèmes Robotiques Autonomes : *Vers une navigation en formation*

<b>IV.1. INTRODUCTION .....</b>	<b>70</b>
<b>IV.2. QUELQUES APPROCHES DE NAVIGATION EN GROUPE.....</b>	<b>71</b>
<b>IV.3. APPROCHE DE NAVIGATION EN GROUPE PROPOSEE.....</b>	<b>72</b>
IV.3.1. DESCRIPTION DU SYSTEME .....	73
IV.3.2. STRATEGIE DE FORMATION DES FIGURES GEOMETRIQUES PAR LES SRAS .....	74
IV.3.2.1. FORMATION D'UN CERCLE PAR LES SRAS.....	75
IV.3.2.1.1. Comportements élémentaires utilisés pour la formation d'un cercle.....	75
IV.3.2.1.2. Fonction de renforcement pour le cercle.....	78
IV.3.2.1.3. Formation d'autres figures .....	79
IV.3.2.2. Formation d'un cercle plein par les SRAs .....	80
IV.3.2.2.1. Comportements élémentaires utilisés pour la formation d'un cercle plein.....	80
IV.3.2.2.2. Fonction de renforcement pour la formation d'un cercle plein.....	80
IV.3.2.3. <i>Arrêt de la formation de SRAs</i> .....	81
IV.3.2.4. <i>Alignement des orientations des SRAs</i> .....	81
IV.3.3. STRATEGIE DE DEPLACEMENT EN FORMATION D'UN GROUPE DE SRAS .....	81
IV.3.3.1. <i>Comportements élémentaires pour le déplacement en formation</i> .....	81
IV.3.3.1.1. Comportement de suivi.....	82
IV.3.3.1.2. Comportement d'évitement de collisions et d'attraction vers la cible.....	82
IV.3.3.1.3. Actions utilisées .....	82
IV.3.3.2. <i>Détermination du "meneur"</i> .....	83
IV.3.3.3. <i>Formation en "meneur-suiveurs" des SRAs</i> .....	83
<b>IV.4. CONCLUSION .....</b>	<b>84</b>

## Chapitre V

# Simulation, Tests et Résultats des Approches de Navigation Proposées des Systèmes Robotiques Autonomes

<b>V.1. INTRODUCTION.....</b>	<b>85</b>
<b>V.2. ENVIRONNEMENT DE SIMULATION.....</b>	<b>85</b>
V.2.1. SIMULATION DES ENVIRONNEMENTS DYNAMIQUES PARTIELLEMENT STRUCTURES.....	85
V.2.2. SIMULATION DES ROBOTS.....	86
V.2.3. SIMULATION DES CAPTEURS A INFRAROUGES (IR).....	86
V.2.4. LANGAGE DE PROGRAMMATION.....	86
V.2.5. HYPOTHESES.....	86
<b>V.3. APPROCHE NEURONALE DE NAVIGATION DES SYSTEMES ROBOTIQUES AUTONOMES.....</b>	<b>87</b>
V.3.1. ENTRAINEMENT DE RNAMF.....	87
V.3.2. UTILISATION DE RNAMF.....	88
V.3.2.1. <i>Evitement de collisions et attraction vers la cible entre un SRA et des obstacles.....</i>	<i>88</i>
V.3.2.2. <i>Evitement de collisions et attraction vers la cible entre deux SRAs dans un environnement simple.....</i>	<i>88</i>
V.3.2.2.1. Environnement sans obstacles.....	88
V.3.2.2.2. Environnement avec obstacles.....	89
V.3.2.3. <i>Evitement de collisions et attraction vers la cible entre deux SRAs dans un environnement complexe.....</i>	<i>90</i>
V.3.3. CONCLUSION.....	91
<b>V.4. NAVIGATION DES SYSTEMES ROBOTIQUES AUTONOMES (SRAs) DANS UN ENVIRONNEMENT DYNAMIQUE : VERS UNE NAVIGATION EN GROUPE.....</b>	<b>92</b>
V.4.1. APPRENTISSAGE DU COMPORTEMENT D'EVITEMENT DE COLLISIONS ET D'ATTRACTION VERS LA CIBLE DES SRAs.....	92
V.4.1.1. <i>Apprentissage du comportement d'évitement de collisions et d'attraction vers la cible par un SRA seul.....</i>	<i>93</i>
V.4.1.2. <i>Apprentissage du comportement d'évitement de collisions et d'attraction vers la cible par plusieurs SRAs.....</i>	<i>94</i>
V.4.1.3. <i>Caractéristiques et performance de l'approche proposée d'évitement de collisions et d'attraction vers la cible.....</i>	<i>96</i>
V.4.1.3.1. Utilisation d'une table de valeurs commune versus plusieurs tables de valeurs privées.....	96
V.4.1.3.2. Performance de l'approche proposée.....	97
V.4.2. TEST DE L'APPROCHE PROPOSEE DANS DES ENVIRONNEMENTS INCONNUS.....	98
V.4.2.1. <i>Retrait d'un robot ou d'un obstacle de l'environnement.....</i>	<i>99</i>
V.4.2.1.1. Cas de l'utilisation de tables de valeurs privées.....	99
V.4.2.1.2. Cas de l'utilisation d'une table de valeurs commune.....	100

V.4.2.2. <i>Environnement complexe</i> .....	102
V.4.2.2.1. Cas de l'utilisation de tables de valeurs privées .....	102
V.4.2.2.2. Cas de l'utilisation d'une table de valeurs commune.....	103
V.4.3. CONCLUSION.....	104
<b>V. 5. COMPORTEMENTS INTELLIGENTS D'UN GROUPE DE SYSTEMES ROBOTIQUES AUTONOMES : VERS UNE NAVIGATION EN FORMATION .....</b>	<b>105</b>
V.5.1. STRATEGIE DE FORMATION DE FIGURES GEOMETRIQUES PAR LES SRAS .....	105
V.5.1.1. <i>Formation d'un cercle</i> .....	105
V.5.1.1.1. Apprentissage de la formation d'un cercle.....	106
V.5.1.1.2. Apprentissage de la formation d'un cercle dans un environnement complexe ..	108
V.5.1.1.2. Test et performance de l'apprentissage .....	109
V.5.1.2. <i>Formation d'autres figures</i> .....	110
V.5.1.2.1. Formation d'un polygone simple.....	110
V.5.1.2.2. Formation d'une ligne.....	111
V.5.2. FORMATION D'UN CERCLE PLEIN .....	111
V.5.2.1. <i>Apprentissage de la formation d'un cercle plein</i> .....	112
V.5.1.2. <i>Stratégie de déplacement en formation d'un groupe de SRAs</i> .....	113
V.5.1.2.1. Navigation de la formation en cercle.....	113
V.5.1.2.2. Navigation de la formation en cercle plein .....	114
V.5.3. CONCLUSION.....	115
<b>V.6. CONTROLEUR FLOU DE MOUVEMENTS : EXECUTION DES ACTIONS GENEREES PAR LES APPROCHES DE PLANIFICATION ET DE CONTROLE PROPOSEES .....</b>	<b>116</b>
V.6.1. DESCRIPTION DU CONTROLEUR.....	116
V.6.2. UTILISATION DU CONTROLEUR FLOU .....	117
<b>V.7. CONCLUSION.....</b>	<b>120</b>
 <b>CONCLUSION GENERALE ET PERSPECTIVES.....</b>	<b>121</b>

## **BIBLIOGRAPHIE**

## **ANNEXES**

## **Introduction Générale**

Ces dernières années, la robotique collective s'est imposée comme un domaine à part entière en robotique mobile. De plus en plus d'approches basées sur l'utilisation de plusieurs Systèmes Robotiques Autonomes (SRAs) se sont développées afin de résoudre divers problèmes allant des applications terrestres aux applications spatiales, où un robot isolé s'avère incapable de faire face seul à la réalisation de son but, requérant impérativement la coopération de plusieurs robots. En effet, depuis quelques années, les systèmes multi-robots ont pris une place de plus en plus importante, que ce soit dans le domaine de l'Intelligence Artificielle (IA), ou dans celui de la robotique collective, en introduisant la problématique de l'intelligence collective et de l'émergence de structures par interactions [32, 67].

Par ailleurs, les recherches dans le domaine de la robotique collective suivent deux objectifs majeurs : le premier concerne l'analyse théorique du comportement lorsque plusieurs robots autonomes interagissent et le deuxième s'intéresse à la réalisation de robots capables d'accomplir des tâches complexes par coopération et interaction. Ces robots en question doivent travailler en groupe pour effectuer toutes sortes de tâches comme surveiller des bâtiments, explorer des espaces dangereux ou inconnus, transporter de gros objets, etc.

En fait, l'intérêt et l'étude de groupes de SRAs affichant un comportement collectif qu'il soit coopératif ou non a émergé comme un domaine actif et plein de défis [14, 16, 77, 78, 121, 132]. L'évolution croissante et significative des systèmes capables de résoudre le problème d'efficacité de calcul ainsi que la miniaturisation des dispositifs processeurs/capteurs/actionneurs ont permis et surtout facilité la création et la mise en œuvre de systèmes multi-robots distribués collectifs dans diverses applications. Cependant, ces systèmes sont plus complexes puisqu'à la complexité liée à chaque entité (SRA) s'ajoute celle induite par le groupe.

La diversité des domaines d'application pour ce type de collectif de robots génère une diversité des exigences en terme de coopération. En effet, lorsqu'un groupe de robots est utilisé pour une mission de surveillance donnée, l'augmentation du nombre de robots va permettre de dresser une carte du site en un temps beaucoup plus court. De la même façon, le transport d'objet par une équipe de robots est très bénéfique quand l'objet est trop grand pour être déplacé par un seul robot. Dans les deux cas, un groupe de robots est plus avantageux qu'un seul robot mais leurs exigences en terme de coopération sont différentes : dans le cas de surveillance, les robots du groupe s'entre aident en distribuant la tâche parmi eux alors que dans le cas de transport d'objet, les robots doivent impérativement garder la figure formée inchangée durant le transport de l'objet. Une attention considérable a été portée vers la formation de figures géométriques [46, 67, 148, 149] où l'objectif final était de réaliser la figure. Malgré cet intérêt, beaucoup de travail reste à faire puisque les approches existantes n'introduisent pas l'apprentissage dans le processus de réalisation de la formation d'une part, et peu de travaux ont traité le déplacement de la formation dans un environnement donné d'autre part.

Dans ce contexte, on s'est alors intéressé à la planification et le contrôle des comportements en groupe des SRAs pouvant être acquis en utilisant un minimum d'informations et aboutissant à l'émergence de comportements complexes afin de réaliser la tâche de transport d'objets lourds et/ou encombrants. En fait, le comportement intelligent collectif résulte des interactions locales basées sur de simples règles. Des comportements élémentaires sont proposés comme méthodologie pour structurer ces règles à travers un processus de synthèse et d'évaluation. Ces comportements sont des lois de contrôle qui rassemblent un certain ensemble de contraintes afin de réaliser et de maintenir un but. Par exemple, "l'évitement" est un comportement qui maintient le but d'éviter les collisions lors du déplacement du SRA. L'accent est mis sur l'étude de ces comportements par rapport à la navigation des SRAs dans des environnements partiellement structurés. La difficulté réside, en fait, dans le travail simultané des SRAs lorsque chacun des robots voit son environnement changer sans qu'il puisse le prédire. Leur navigation doit donc être sensible aux changements produits par les autres interventions.

Cette approche ascendante qui, à partir d'un certain nombre de comportements élémentaires, construit un comportement plus complexe, permet la réalisation de divers objectifs à la lumière d'une connaissance assez restreinte de la tâche à accomplir. En effet, afin de mener à bien la tâche de transport, les robots doivent d'une part réaliser une figure géométrique correspondant à l'objet à transporter et d'autre part à le déplacer en maintenant cette formation. Pour ce faire, deux aspects complémentaires sont traités à savoir :

- L'introduction de l'apprentissage dans la réalisation de la figure géométrique ;
- Le déplacement du groupe ainsi formé en maintenant cette formation.

Ces deux phases en fait utilisent l'interaction des robots avec l'environnement afin d'atteindre leurs objectifs en exploitant certains comportements élémentaires tels que le suivi, l'évitement, le regroupement et la séparation.

Des approches, utilisant ces comportements élémentaires pour générer plusieurs comportements de groupe robustes incluant l'évitement de collisions et l'attraction vers la cible, la formation de figures géométriques et le déplacement en formation, sont proposées dans cette thèse. L'apprentissage est également introduit afin de permettre au groupe de robots d'apprendre la tâche désirée. Deux types d'apprentissages sont alors présentés, un basé sur la collecte d'exemples d'entraînement et l'autre sur l'interaction des robots avec l'environnement. Le premier exploite les Réseaux de Neurones (RN) avec leur capacité de généralisation alors que le second basé sur l'Apprentissage par Renforcement (AR) utilise des comportements élémentaires comme unité de représentation et permet au groupe de robots d'apprendre des tâches complexes.

Finalement, les approches proposées sont validées par des résultats de simulation réalisés sur des groupes homogènes de SRAs.

Ce document est organisé en cinq chapitres correspondant aux thèmes suivants :

Le premier chapitre introduit des généralités sur les SRAs, leurs architectures, leur coopération et leur communication, ainsi que sur les approches de navigation telles que les approches locales et globales. Divers aspects de la robotique collective qui cherche à résoudre ces problèmes en exploitant cette notion de coopération entre les différents robots sont abordés. L'intelligence collective qui est en fait la base de cette nouvelle discipline est présentée ainsi que le concept de l'auto-organisation qui est une caractéristique très utile et

essentielle aux systèmes multi-robots. La coopération est ensuite présentée en mettant l'action sur les différentes méthodes de coopération.

Le chapitre II quant à lui présente une approche d'évitement de collisions et d'attraction vers un but donné (cible). Elle se base sur un apprentissage de ce comportement essentiel dans la navigation des SRAs. Cette approche exploite les caractéristiques combinées des RN, de la Logique Floue (LF) et de la théorie de la Résonance Adaptative (RA) sous forme d'un système hybride intelligent afin de permettre, après apprentissage, à deux robots d'atteindre leurs cibles respectives tout en évitant les collisions. En effet, chaque robot doit apprendre, en utilisant le système sensoriel LOCISS (LOcally Communicable Infrared Sensory System) pour chaque relation de position (i.e., pour chaque position de l'obstacle par rapport au robot considéré) à déduire la règle appropriée d'évitement de collisions et d'attraction vers la cible. Ainsi, cette approche doit rendre les robots capables, après apprentissage: AMFS (ArtMap Floue Simplifiée), de déterminer et d'utiliser la règle permettant l'évitement de collisions et l'attraction vers la cible.

Dans le chapitre III, une approche plus générale basée sur une très forte interaction avec l'environnement est introduite. Dans cette approche, chaque robot doit apprendre comment éviter les autres robots à partir de son interaction avec l'environnement tout en atteignant son but. Ce processus d'apprentissage permet aux robots de bénéficier de l'expérience des autres en partageant la même table de valeurs ce qui revient à partager la connaissance des uns et des autres. Cette approche doit fournir aux robots la capacité d'acquérir le comportement d'évitement de collision et d'attraction vers la cible entre plusieurs robots à partir de comportements élémentaires en utilisant un apprentissage par essai et erreur.

Le chapitre IV traite du problème de transport d'objets en question. Dans ce chapitre, une approche de navigation en groupe est proposée afin de doter ces SRAs d'une stratégie de contrôle leur permettant de réaliser la tâche demandée. Cette approche consiste en la formation de la figure géométrique correspondant à l'objet en question suivie par le déplacement de cette formation vers son objectif. Elle est basée sur une stratégie décentralisée de formation d'une figure géométrique et une stratégie centralisée de déplacement. Elle utilise l'AR afin d'aboutir à l'émergence d'un comportement plus complexe résolvant le problème posé et introduit également la technique de "meneur-suiveurs" modifiée dans le but de garder impérativement la formation réalisée dans un environnement à priori inconnu.

Dans le chapitre V, les résultats de simulation des approches de navigation proposées sont donnés. Ces résultats montrent la capacité de ces approches à faire évoluer les robots dans un environnement à priori inconnu. Celles-ci permettent d'atteindre les buts (cibles) des robots tout en évitant les obstacles de l'environnement. Ces approches sont mises en œuvre grâce à un contrôleur flou permettant l'exécution des actions générées par ces approches.





## Chapitre I

# Robotique Collective : *Etat de l'art et synthèse*

### I.1. Introduction

Ces dernières années ont été riches en exploits dans le domaine de la recherche et du développement des Systèmes Robotiques Autonomes (SRAs) intelligents, en particulier dans les explorations spatiales et sous-marines ainsi que les applications terrestres (industrielles, agricoles, médicales, etc.). Ceci témoigne d'une avancée certaine à la fois dans la recherche en robotique et en intelligence artificielle et plus particulièrement dans la coopération ainsi que la planification et le contrôle de la navigation des SRAs en environnements qui peuvent être imprécis, dynamiques et partiellement ou non structurés.

Pour certaines tâches robotiques, telles que soulever et transporter un grand objet, assembler de multiples parties, il est plus fiable, robuste et parfois nécessaire d'employer un groupe de petits robots mobiles. En effet, la robotique collective auto-organisée, exploitant les concepts développés dans le domaine de l'intelligence collective, est orientée vers des agents qui présentent des comportements simples (par exemple de type stimulus-réponse) et dépourvus de représentation ou de connaissance logique comme c'est le cas dans l'Intelligence Artificielle Distribuée (IAD) [32]. L'intelligence collective trouve son origine notamment dans les systèmes biologiques tels que les sociétés d'insectes, et c'est ainsi qu'il y a eu apparition de l'idée de développer des groupes intelligents de robots simples s'inspirant de la logique de fonctionnement des sociétés animales. En effet, il existe dans la nature de nombreux systèmes collectifs capables d'accomplir des tâches difficiles, dans des environnements dynamiques et variés, et ceci sans pilotage ni contrôle externes et sans coordination centrale. Les colonies d'insectes par exemple, plus particulièrement les colonies de fourmis, d'abeilles, de guêpes ou de termites, possèdent cette capacité de réaliser des tâches routinières nécessaires à leur survie, aussi bien que dans leurs réponses à des événements fortement perturbateurs.

Plusieurs motivations ont alors poussé les chercheurs en robotique à s'intéresser à cette discipline notamment le besoin d'utiliser un grand nombre d'agents (identiques ou non) pour effectuer une tâche donnée plus rapidement ou réaliser une tâche complexe. Le besoin de robustesse face à l'aléatoire ou encore l'ambition de produire une performance collective qualitativement supérieure à celle des unités ont également contribué à cet intérêt. La technologie aujourd'hui disponible pour construire des groupes de SRAs aptes à présenter une intelligence collective à partir d'agents simples est largement suffisante. Il s'agit alors :

- de trouver les algorithmes comportementaux individuels permettant de réaliser une performance collective à partir d'agents simples,

- et d'utiliser la technologie pour concevoir un groupe d'agents évoluant dans un monde réel qui se révèle souvent beaucoup plus difficile à maîtriser qu'un monde simulé.

Ce chapitre se décompose donc en deux grandes parties : une partie traitant de l'intelligence collective proprement dite et une seconde partie se consacrant à la robotique collective. Dans la première partie, on présente quelques définitions et généralités qui sont susceptibles d'être utilisées dans la robotique collective. La seconde partie traite de l'auto-organisation, de la coopération des systèmes robotiques, des architectures de contrôle correspondantes et se termine par la présentation de quelques expériences robotiques développées en s'inspirant de cette thématique de groupe.

## **I.2. Intelligence collective**

### **I.2.1. Définition**

La notion d'intelligence collective est née dans les années 60, de l'application à la robotique des analyses des éthologues sur le comportement des sociétés d'insectes.

Comment expliquer, chez les insectes, que chaque individu participe à un comportement collectif complexe auto-régulé sans pour autant bénéficier d'une vision globale du but à atteindre ?

La réponse réside dans la notion d'intelligence collective. Une notion qui découle de l'observation des sociétés animales, une observation qui est au cœur des travaux de nombreux cybernéticiens [32]. Le point de vue le plus utilisé est : on considère des entités qui individuellement sont dotées de capacités cognitives restreintes- on ne parle donc pas ou peu d'intelligence en ce qui les concerne- mais qui collectivement peuvent manifester des comportements intelligents. C'est donc au collectif, défini comme un ensemble d'entités ayant entre elles des interactions "fonctionnellement" pertinentes, que s'applique le terme intelligence.

Lorsqu'on parle alors d'intelligence, le pré-requis minimal est l'existence d'un agent en interaction avec son environnement. Cet agent est dit intelligent si et seulement si, dans ses interactions avec son environnement, il réalise des actions-perceptions bouclées de sorte à satisfaire une contrainte de type "viabilité". Cette contrainte est rigoureuse mais elle ne spécifie pas la stratégie à employer. L'exemple en est fourni par les organismes vivants – la contrainte est ici la suivante : l'organisme doit assurer, d'instant en instant et dans la durée, le maintien de son identité (d'où l'expression "contrainte de type viabilité"). Toutefois cette définition peut être étendue à des robots et autres réalisations de type "vie artificielle" [32, 99]. Un exemple d'une telle contrainte applicable à un robot serait de ne pas rester immobile, de ne pas se cogner contre des obstacles et de ne pas tomber par les bords de son terrain.

L'intelligence collective est en fait un phénomène émergent qui dépend des relations entre les agents. Il y aura donc des interactions entre ces agents, avec émergence d'une entité supérieure (super organisme, colonie, société, etc.) [32].

### **I.2.2. Notion d'émergence**

L'émergence est un nouveau concept qui commence à être utilisé dans divers domaines tels que la physique, la biologie, la systémique et la robotique. C'est un phénomène en cours d'exploration, particulièrement dans les systèmes multi-robots.

Une fonctionnalité qui émerge est une fonctionnalité qui n'a pas été explicitement programmée, elle est le résultat d'interaction entre deux ou plusieurs comportements. En effet, la fonctionnalité accomplie par une société n'est pas la simple somme des fonctionnalités de ses agents, mais résulte de l'action commune des agents en interdépendance [98, 99]. Par exemple, si on veut qu'un robot (en marche) suive un mur (comportement *suivre le mur*), on définit deux comportements (antagonistes) : *être attiré par le mur* et *être repoussé par le mur*. De l'interaction de ces deux comportements va émerger un nouveau comportement : *suivre le mur*.

Dans ce cas, on parle d'émergence de comportement, mais il existe aussi d'autres formes d'émergence telles que l'émergence de rôles, l'émergence de stratégies, l'émergence de contrôle et l'émergence des structures de coordination. L'émergence est un phénomène compliqué et très peu compris, beaucoup de chercheurs s'y intéressent de nos jours [98].

### I.2.3. Intelligence collective en robotique

L'idée fondamentale de l'intelligence collective, c'est qu'à l'image des sociétés d'insectes (comme les fourmis et les abeilles), une grande quantité de petits robots autonomes, dotés d'une perception réduite de leur environnement et d'une programmation relativement simple, seraient capables de résoudre des tâches complexes, impossibles à réaliser par un robot doté d'une intelligence artificielle sophistiquée.

Les exemples sont nombreux, particulièrement dans les sociétés d'insectes (fourmis, abeilles, termites), mais aussi dans le système de défense de l'organisme (globules blancs) ou le système immunitaire (interactions macrophages-lymphocytes) [32, 99]. Appliquée à la cybernétique et à la robotique cette notion débouche sur la robotique collective auto-organisée.

Cette même notion a été utilisée également à la fin des années 80 en IAD. Il s'agit de l'étude et de l'utilisation des principes de fonctionnement des sociétés d'insectes. En effet, ce sont des systèmes composés d'agents très simples (réactifs) qui peuvent produire des constructions complexes et des solutions à des problèmes non triviaux (tris, parcours optimaux, répartition de tâches, etc.) [32].

L'intelligence collective représente en fait les principes d'auto-organisation et d'émergence présents dans ces sociétés d'insectes. La méthodologie principalement empirique qui l'accompagne – simulation, observation, perturbation – est toutefois renforcée par les acquis de l'éthologie et de la biologie.

Les avantages liés à une telle approche peuvent se résumer ainsi :

- fiabilité du système dans son ensemble (la perte d'un agent ne remet pas en cause le processus général),
- production d'une performance collective qualitativement supérieure à celle des agents,
- une plus grande flexibilité face aux aléas des environnements réels, tolérance aux défaillances et utilisation des événements aléatoires.

Dans ce paradigme, la coopération est présente à travers l'auto-organisation des agents. Ce sont des comportements collectifs intelligents qui émergent de nombreuses interactions entre les agents et leur environnement.

Ces principes ont été repris par plusieurs chercheurs dans la conception de robots autonomes. Il s'agit d'une part des systèmes composés de robots cellulaires [28] (concept de

l'intelligence en essaim) et d'autre part de robots "à base de comportements simples" non communicants et moins nombreux. Ces derniers ont montré leur aptitude à traiter des tâches de regroupement [109], de partage des ressources [143], de navigation en formation [21, 23], de transport "émergents" (Box-pushing) [96] et de fourragement [9, 108]. La plupart de ces exemples ou tous utilisent cette notion d'auto-organisation dans leur évolution dans la réalisation de leurs tâches.

### **I.3. Auto-organisation**

#### **I.3.1. Quelques définitions**

Le terme d'auto-organisation a initialement été défini par Farley et Clark du laboratoire Lincoln [64]. Cem unsal [162] cite cette définition dans sa thèse : *un système auto-organisateur est un système qui change sa structure de base en fonction de son expérience et de son environnement.*

Toru Ishida *et al.* [85] donnent une définition plus générale de l'auto-organisation. Elle permet à une organisation d'agents de s'adapter elle-même aux situations dynamiquement changeantes.

Selon Young-pa So *et al.* [141], les agents coopérants travaillent typiquement dans des environnements changeants. Il est donc crucial que les mécanismes de coordination qu'ils utilisent soient capables de s'adapter à ces changements. En particulier, l'organisation doit évoluer dans le temps lorsque les circonstances le nécessitent.

Pour Pierre Glize et son équipe [41], l'auto-organisation permet à un système de s'adapter de manière autonome aux conditions changeantes du milieu. Elle consiste plus précisément en une transformation de la topologie (c'est-à-dire des relations entre les agents) en tant que résultat du fonctionnement de ce même réseau dans le cadre de son couplage structurel avec l'environnement. Ils ajoutent que les systèmes auto-organiseurs appartiennent à la classe des systèmes autonomes (systèmes spécifiés par des mécanismes internes d'auto-organisation) et non pas hétéronomes (systèmes définis par des mécanismes extérieurs de contrôle). Cela implique que les règles d'organisation sont intérieures au système qui apparaît ainsi informationnellement clos [69]. C'est le cas des réseaux de neurones par exemple.

#### **I.3.2. Caractéristiques des systèmes auto-organiseurs**

L'auto-organisation possède trois caractéristiques importantes. Premièrement, un système auto-organisateur peut accomplir des tâches complexes avec un comportement individuel simple. Deuxièmement, un changement dans l'environnement peut influencer le même système à générer une tâche différente, sans aucun changement dans les caractéristiques comportementales. Finalement, n'importe quelle petite différence dans le comportement individuel peut influencer le comportement collectif du système [162].

Dans un système auto-organisateur, le comportement individuel n'a pas besoin d'être modifié dans le but d'avoir un comportement collectif différent. Cette caractéristique d'auto-organisation est très avantageuse pour des robots en essaim puisque de simples comportements individuels peuvent être réalisés avec une conception relativement simple et peu coûteuse.

### **I.3.3. Avantages de l'auto-organisation**

On rencontre l'auto-organisation dans plusieurs domaines tels que la biologie (écosystème), la chimie (thermodynamique) et l'informatique (réseaux de neurones, logique floue), grâce aux nombreux avantages qu'elle présente. Dans le monde de la robotique collective, on peut citer parmi les avantages de l'auto-organisation, la coordination d'un grand nombre de robots dans un système collectif en interaction avec l'environnement. Un tel système s'avère simple et fiable puisque la panne d'un robot n'influe pas sur la réalisation de la tâche à effectuer, ce qui ne peut pas être le cas dans un système déterministe tel qu'une chaîne de production.

Tous ces exemples s'inspirent des sociétés animales et plus précisément des insectes sociaux. En effet, l'étude du comportement de colonies d'insectes a mis en évidence des hypothèses intéressantes quant à leur organisation : bien qu'apparemment l'activité individuelle de chaque insecte ne soit pas supervisée, la colonie semble être totalement auto-organisée.

### **I.3.4. Auto-organisation chez les insectes**

L'auto-organisation se base essentiellement voire totalement sur l'interaction entre individus et entre ceux-ci et leur environnement. En effet, chaque individu réagit par une réponse automatique à un certain stimulus significatif ou à une modification de l'environnement, crée une nouvelle stimulation, laquelle agira sur un quelconque congénère. Chez les termites, comme étudié dans [32], la construction ordonne, par elle-même, les réactions des sujets et aboutit, par la voie de l'automatisme et de la plus totale inconscience individuelle, à une œuvre parfaitement cohérente, donnant l'illusion de répondre à un plan défini. Pratiquement, tous les insectes sociaux agissent ou réagissent de cette manière donnant effectivement cette impression d'un travail planifié au préalable.

#### **I.3.4.1. Différentes formes d'organisation des sociétés animales**

##### **I.3.4.1.1. Conditions de formation du groupe social**

Deux conditions sont nécessaires à la formation d'un groupe social et qui sont la disparition de l'agressivité interindividuelle et l'existence d'un principe de rassemblement. Cette dernière peut correspondre à une interattraction directe ou à une attraction commune vers une étendue significative. Tout groupe à fonctionnement collectif devra posséder un tel principe pour éviter la dispersion.

##### **I.3.4.1.2. Persistance du groupe durant le déplacement**

Dans les formes les plus simples de coordination, le maintien à proximité d'animaux engagés dans un comportement spécifique (par exemple prise de nourriture) ne fait qu'indiquer la persistance d'un stimulus de contact qui peut se traduire par de simples cris (pépiements des poussins) ou des informations directionnelles plus précises (déplacement rapide ou à longue distance) qui renseignent sur la présence des congénères. Le cas le plus intéressant est sans doute celui des déplacements, parfois à longue distance, qu'effectuent certains animaux migrateurs : criquets, oiseaux ou harengs.

Le déplacement de groupe utilise plusieurs niveaux d'intégration des comportements individuels. Dans le cas le plus simple, n'impliquant qu'une limitation de la dispersion, chaque individu a pour le reste un comportement solitaire. Le rôle du groupe se limite sans doute à une certaine défense contre la prédation, sans coopération précise pour les autres

activités. Les dispositifs plus précis d'ajustement de positions correspondent à des déplacements plus rapides, accroissant les risques de heurt ou de dispersion. Enfin, la réalisation d'une forme plus précise a généralement une signification fonctionnelle directe et témoigne en cela que le déplacement considéré s'inscrit dans une fonction biologique précise s'exerçant collectivement, comme dans le cas des chasseurs en bande.

#### **I.3.4.1.3. Autres comportements**

Il y a également d'autres manières de s'organiser par différenciation par exemple ou alors le comportement de récolte de nourriture qui se fait par la chasse collective d'une proie chez les animaux alors que chez les insectes elle se traduit par la danse chez les abeilles, par le recrutement chez les fourmis, etc. On peut également trouver un comportement de défense du groupe chez plusieurs espèces. En effet, il n'y a pas un dispositif particulier de défense du groupe : chaque individu réagit au danger indépendamment des autres en fuyant ou en se cachant à l'arrivée d'un prédateur. Seule la multiplicité des récepteurs sensoriels due à la présence de plusieurs individus possède une valeur adaptative, ce qui peut être le cas lors de regroupement étroit d'un vol de moineaux à l'arrivée d'un faucon, cercle des gnous autour des jeunes en face d'une attaque de lions, etc.

#### **I.3.4.2. Fonctionnement collectif**

On parle "d'intelligence" collective lorsqu'un groupe social peut résoudre un problème dans un cas où un agent isolé en serait incapable. Les traits généraux d'un tel fonctionnement collectif sont notamment dus à quatre propriétés, émergeant des interactions entre individus et qui permettent en particulier de pallier les contraintes liées aux faibles capacités cognitives des insectes [32] :

- Un gain "computationnel" par une utilisation pertinente des informations où le fonctionnement du groupe doit optimiser la transmission de signaux entre agents. Cela se fait par une gestion locale n'impliquant pas de représentation globale de la situation et par une utilisation de la réponse de chaque agent pour produire un signal qui soit significatif pour les autres membres du groupe.
- Un ajustement des réponses accroissant l'efficacité résultante par une simple juxtaposition des réponses individuelles sans dispositif particulier (ainsi le simple accroissement du nombre des récepteurs dans un groupe permet la détection plus facile d'un prédateur) ou par la spécialisation des individus qui permet la réalisation simultanée de plusieurs fonctions.
- Une large redondance de fonctionnement grâce à la présence de systèmes fonctionnels en surnombre qui permettent de pallier le dysfonctionnement d'un agent ou qui consistent à réagir aux mouvements de fuite ou autre du congénère.
- Une structure fonctionnelle apparue au terme d'une histoire évolutive où le mécanisme sélectif implique que chaque étape d'évolution corresponde à un mode d'organisation qui soit au moins aussi viable, dans le milieu alors fréquenté, que celui qui existait lors de l'étape précédente.

### **I.3.5. Société animale et groupes de robots mobiles**

La réalisation collective d'une performance adaptée qui peut être qualifiée d'intelligence collective repose sur des principes d'organisation pouvant s'appliquer à tout groupe d'agents aptes à répondre à des signaux, c'est-à-dire à traiter de l'information [32].

Ce qui importe si l'on cherche à s'inspirer de modèles biologiques pour construire des groupes coordonnés d'agents artificiels n'est donc pas tant de copier les solutions adoptées que de s'inspirer de la façon, simple et efficace, dont les systèmes naturels ont pu contourner les contraintes qui s'imposaient.

(i) la première contrainte concerne les paramètres propres de la tâche à accomplir et les priorités que celle-ci comporte.

(ii) la deuxième contrainte concerne les moyens dont disposent les animaux pour communiquer et pour traiter les informations qui leur sont transmises.

(iii) Enfin, la dernière contrainte concerne leur origine historique.

La redondance peut représenter un choix délibéré, destiné à accroître la fiabilité en minimisant le risque de panne ou de blocage du fonctionnement.

En s'aidant de ces contraintes, des systèmes multi-robots peuvent être construits en imitant les comportements sociaux et le fonctionnement social des insectes tout en tenant compte de leurs caractéristiques propres.

Quelques travaux sur les robots mobiles autonomes, les systèmes multi-robots et le comportement de robots sont présentés qui permettraient de montrer l'utilisation de certaines caractéristiques des sociétés animales présentées dans cette section.

## **I.4. Systèmes robotiques autonomes**

L'architecture de contrôle "subsumption" dotant les SRAs est présentée dans ce qui suit ainsi que le principe de la navigation permettant aux robots de se déplacer sans collisions dans un environnement dynamique.

### **I.4.1. Architecture subsumption**

Le terme subsumption est dérivé de "subsumer" qui signifie : conditionner (une idée, un principe, etc.) à une règle, à une classe ou à une situation. Cela se rapporte à l'architecture d'un système comprenant une série de comportements différents. La sélection d'un comportement et la mise en œuvre d'une aptitude sont fonction de la situation dans laquelle le système se trouve incorporé, situation telle qu'elle est appréhendée par les entrées/sorties (senseurs et éléments d'action se conduisant en tant qu'éléments d'entrée/sortie). Un niveau d'organisation plus élevé peut être présent.

L'architecture subsumption contrôlant les robots mobiles a été introduite par Brooks [39]. Dans une telle architecture, les couches du système de contrôle sont construites afin de permettre au robot de fonctionner à des niveaux de compétition différents et croissants et sont constituées de modules asynchrones communicants. Chaque module est une machine de traitement simple, et les couches des niveaux hauts peuvent surpasser les sorties des niveaux plus bas (subsumption). Mais, les niveaux bas continuent de fonctionner quand les niveaux hauts (qui peuvent interférer avec leurs données d'entrées) sont ajoutés. Chaque niveau génère un comportement et la compétition du robot est augmentée en ajoutant de nouvelles couches. L'architecture subsumption est basée sur la décomposition du robot mobile en termes de

comportements plutôt qu'en termes de modules fonctionnels. Un module de contrôle centralisé n'est pas nécessaire puisque tout le système de contrôle peut être vu comme un système d'agents agissant séparément (voir figure I.1).

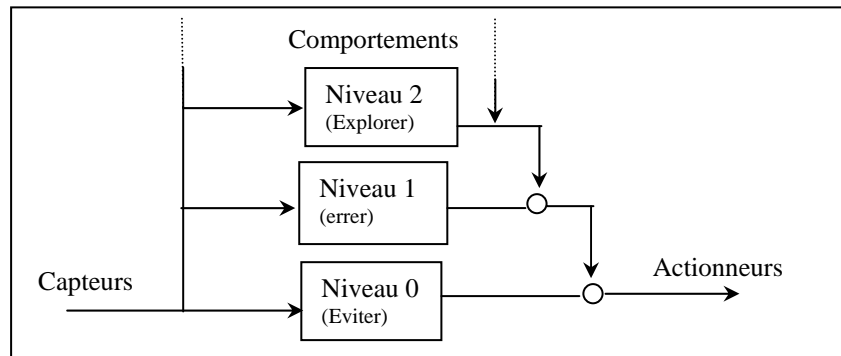


Figure I.1. Architecture de contrôle subsomption.

En même temps, certains chercheurs ont reconnu qu'une itération du cycle "perception-puis-planification-puis-action" donne une réponse trop longue inintéressante pour plusieurs tâches robotiques. Des architectures qui permettent le traitement à différents niveaux de résolution ont été proposées. Brooks introduisit des idées similaires dans l'architecture subsomption en proposant une architecture de contrôle constituée par des modules de traitement verticaux plutôt que la traditionnelle décomposition horizontale en modules fonctionnels (voir figure I.2).

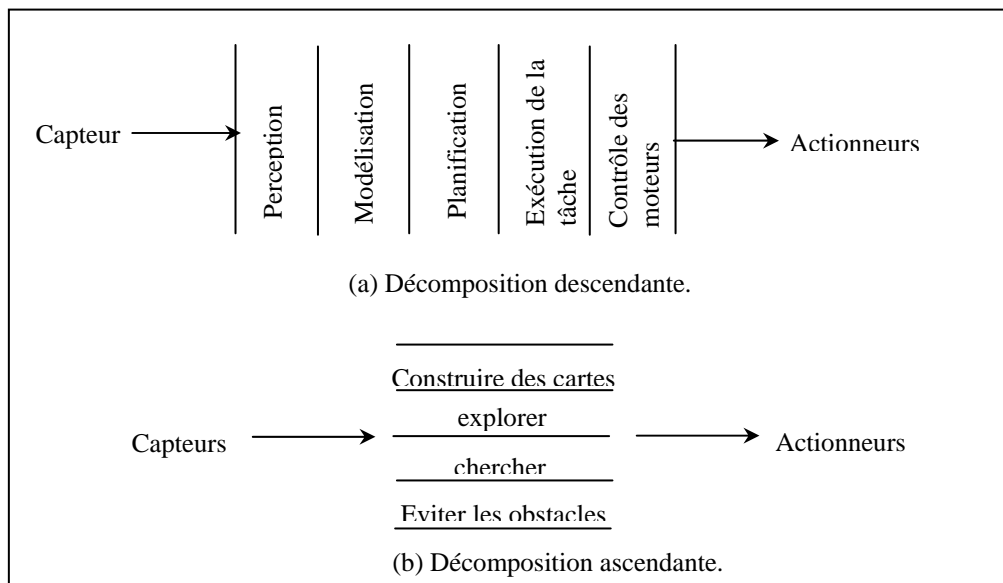


Figure I.2. Décomposition classique (descendante) versus comportementale (ascendante).

L'approche comportementale est en fait vue comme une philosophie à part entière plutôt que d'être associée uniquement à l'architecture subsomption de Brooks. Sa force est dans les caractéristiques des agents *intégrés*<sup>1</sup> et *situés*<sup>2</sup> qui travaillent en temps-réel et les représentations explicites sont évitées [36].

<sup>1</sup> *Intégré (embodied en anglais) : concerne les systèmes physiques (leur dynamique) et théoriques (relations entre les signaux des capteurs, contrôle des moteurs), i.e., un agent intégré peut être un robot physique. En effet, ce terme se réfère au fait que l'intelligence ne peut exister sous la forme d'un algorithme abstrait mais requiert une instantiation physique, un corps.*

<sup>2</sup> *situé : Robot situé est un système intelligent qui ne doit pas fonder son comportement sur les raisonnements strictement abstraits effectués dans un système clos, mais doit agir et se comporter de façon à assurer sa survie dans son monde environnant.*



Un exemple d'architecture subsomption est l'architecture de Squirt<sup>3</sup>, un très petit robot mobile intelligent (voir figure I.3). Squirt [140] agit comme une punaise (insecte), se dissimulant dans les coins noirs et s'aventurant dans la direction des bruits uniquement quand ces bruits disparaissent, cherchant après une nouvelle place pour se dissimuler près de l'endroit d'où sont venus les bruits. Le fait le plus intéressant concernant Squirt est la manière dont son comportement de haut-niveau émerge d'un certain nombre d'interactions avec l'environnement. Son niveau de comportement le plus bas conduit le robot à chercher après le noir. Le second niveau de comportement est activé une fois un endroit noir est trouvé. La direction des bruits est détectée via des capteurs et quand quelques minutes de silence suivent un bruit aigu, Squirt se déplace dans la direction du dernier bruit entendu, refoulant le désir de rester dans le noir. Après une période de temps, le premier niveau n'est plus refoulé et devient actif.

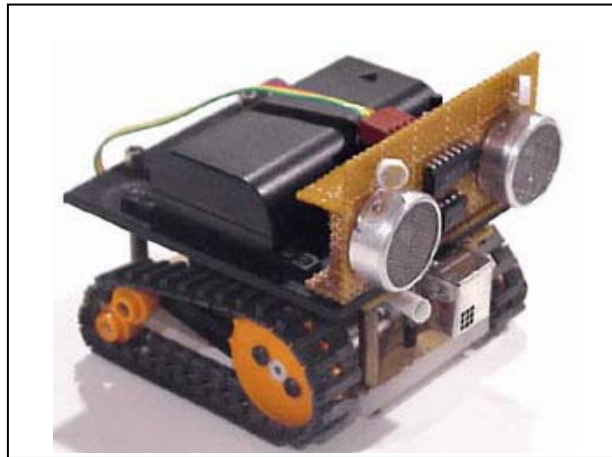


Figure I.3. Squirt [140].

L'architecture subsomption a également montré une navigation robuste des robots mobiles dans des environnements dynamiques et changeants [39, 109].

#### I.4.2. Navigation autonome

La plus importante fonction (ou la première couche d'une architecture de contrôle subsomption) d'un robot mobile est sa capacité à éviter les obstacles. Un robot autonome reconnaît son environnement en utilisant des capteurs et décide de l'action à réaliser en se basant sur les données de ces capteurs. Rodin et Amin [134] ont défini une structure générale d'un algorithme de navigation intelligent afin de résoudre le problème de contrôle temps-réel dans un environnement avec des obstacles mobiles et qui est constitué de trois niveaux (identificateur, sélecteur de buts et adaptateur). D'autres approches utilisent l'apprentissage pour réaliser cette tâche de navigation et qui est en fait efficace dans des environnements dynamiques [21, 49].

L'objectif de la navigation est l'amélioration de l'autonomie, de l'intelligence et des temps de réponse de ces robots qui impliquent trois aspects essentiels. En premier, ces robots doivent être capables d'exécuter efficacement des tâches telles que la reconnaissance, l'apprentissage, l'adaptation, le raisonnement, la prise de décisions et l'action. Ils doivent également être capables de réduire la charge d'un opérateur par l'utilisation d'un langage

<sup>3</sup> Squirt pèse environ 50 gr. de 5/4 inches de volume. Il a un microprocesseur à 8 bits embarqué, une batterie, trois capteurs et un système de propulsion.

naturel afin de permettre une facilité dans la prise de décision. Finalement, ils doivent accomplir des tâches à un niveau humain à l'aide de capacités d'apprentissage, d'adaptation et de généralisation.

#### **I.4.2.1. Approches classiques**

Dans les approches classiques de planification et de contrôle de la navigation, il a été souvent supposé une connaissance à priori et exacte du robot, de ses capteurs et plus encore de son environnement. Ceci a limité le type d'environnements où de tels robots peuvent accomplir des tâches et par conséquent le type de tâches que ces robots peuvent résoudre. En effet, ces limitations sont dues principalement aux difficultés qu'ont les concepteurs à fournir des modèles exacts des robots et de leurs environnements. Même si une telle connaissance est disponible, la rendant accessible par ordinateur, le codage à la main de modèles explicites a souvent exigé des temps de programmation non raisonnables. Par ailleurs, atteindre une facilité de contrôle du calcul ("*tractability*") est un sérieux obstacle pour la conception de structures de contrôle pour les robots dans des environnements dynamiques complexes [49, 154].

Les approches classiques de planification et de contrôle de la navigation ont largement été développées [1, 100]. On cite notamment les approches basées sur le raisonnement géométrique, sur la configuration d'espace, sur les heuristiques appliquées généralement aux environnements statiques ainsi que celles basées sur le champ de potentiel artificiel appliquées aux environnements statiques et dynamiques.

#### **I.4.2.2. Approches actuelles**

Dans les approches conventionnelles, le raisonnement sur le comportement du système robotique et sur le comportement global désiré se base sur le formalisme de descriptions mathématiques caractérisé par un haut niveau d'abstraction. Dans le cas où le modèle mathématique du système robotique n'est pas disponible ou ne peut être obtenu qu'avec un grand effort et un coût élevé comme c'est le cas avec les robots mobiles, la stratégie de contrôle devrait être basée sur une approche complètement différente. Au lieu d'une description mathématique, une description comportementale du processus est nécessaire se basant à la fois sur des expressions qualitatives et sur l'expérience humaine [95].

Afin de prendre en charge tous les problèmes en environnements dynamiques réels, ce qui représente le défi de ces dernières années, on doit introduire de nouvelles approches de planification et de contrôle, s'inspirant du comportement humain, appelées contrôles intelligents [22, 95]. Ces nouvelles approches se basent essentiellement sur le "Soft Computing" (SC) introduisant particulièrement les réseaux de neurones, la logique floue, les algorithmes génétiques, la résonance adaptative ainsi que l'apprentissage par renforcement. Le SC est un domaine regroupant spécifiquement les approches visant à comprendre et à mimer le fonctionnement du cerveau humain. En effet, ces approches sont développées afin d'assurer: l'auto-calibration des robots et l'auto-réparation (dépannage ou remise en état), le mouvement habile (avec dextérité), l'auto-reproduction de structures intelligentes, l'apprentissage et l'évolution (comportement adaptatif et mécanismes d'évolution).

### **I.5. Systèmes multi-robots**

Ces dernières années, de plus en plus de systèmes à plusieurs robots mobiles sont utilisés où des projets tels que les missions planétaires et le maniement de déchets dangereux,

ont émergé. En effet, de larges populations de robots mobiles, comme les Systèmes Robotiques Décentralisés (SRD), sont plus avantageux que les systèmes centralisés, particulièrement quand une grande fiabilité est requise dans les tâches à exécuter telles que les tâches de maintenance dans les sites nucléaires.

L'utilisation de plusieurs robots mobiles dans les missions planétaires est aussi soulignée par Miller dans [118]. Ce travail montre qu'un groupe de petits robots mobiles autonomes est plus avantageux à exploiter qu'un robot complexe et sophistiqué pour diverses raisons dont, entre autre, un moindre coût, une petite masse et une fiabilité de la mission. Les méthodes de contrôle basées sur le comportement décrites dans la section précédente peuvent être utilisées pour concevoir de tels petits robots. L'utilisation de balises (radio) fixes est aussi à envisager avec la nécessité de sélection d'un "meneur" pour la formation d'un groupe coordonné (la sélection du "meneur" peut être réalisée en assignant un numéro de série aux robots par exemple). Miller a aussi souligné le fait que les balises codées et les lecteurs de balises sur chaque robot, avec d'autres signaux simples, peuvent être suffisants pour réaliser une tâche complexe avec un comportement individuel puisque les techniques de navigation sont bien développées pour les robots mobiles autonomes.

Le terme "système robotique distribué" (SRD) est parfois utilisé pour décrire un système multi-robots basé sur des réponses instinctives et sur la coopération. Des simulations de SRD conçu pour rechercher des polluants sont développées par Genovese *et al.* [76], et sont basées sur les systèmes biologiques et l'architecture subsomption. Cette conception suggère un utilisateur superviseur qui peut localiser des agents à chaque fois qu'il en a besoin. Dans ce contexte, le système n'est pas un essaim. Les communications entre agents sont plus complexes que celles décrites par Miller et incluent une transmission par radio commande. Dans [29], Beni et Wang supposent que tous les agents requis dans une opération doivent se communiquer leur intention afin d'exécuter leur partie de l'opération. Attestant que les protocoles d'engagement (commitment protocols) sont la base de la construction de blocs d'algorithmes de traitement distribués, une telle communication est définie comme une caractéristique exigée. Au contraire, les systèmes biologiques décrits dans § I.3 sont capables de travailler comme un système auto-organisé sans communication directe (des intentions). Le facteur clé ici est le grand nombre d'agents.

Des travaux expérimentaux sur plusieurs robots mobiles incluent ACTRESS (ACTor based Robots and Equivalent Synthetic Systems) développé par Habib *et al.* [80], et les robots Yamabica réalisés par Yuta et Premvuti [173]. ACTRESS, système robotique autonome et décentralisé, n'est pas constitué uniquement de robots mobiles mais aussi de n'importe quel type de système robotique et/ou d'ordinateurs (voir figure I.4). Les robots Yamabica, plus compacts que les robots d'ACTRESS, sont utilisés afin de déterminer la solution pour une situation de blocage causée par plusieurs robots mobiles avec un recouvrement des exécutions. La méthode décrite dans [173] fournit un processus de dérivation afin de résoudre le problème de blocage. Cependant, il requiert une diffusion continue de l'information (e.g., la position actuelle) et une connaissance du monde, et est basé sur des modules de décision complexe gérant l'information obtenue des capteurs.

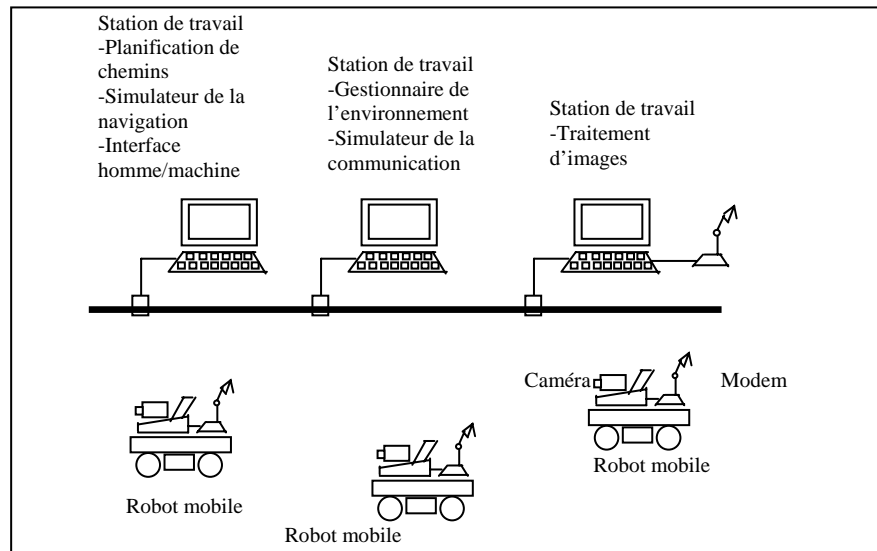


Figure I.4. Architecture d'ACTRESS.

D'autres approches basées sur l'apprentissage sont développées afin de faire évoluer un groupe de robots dans des environnements dynamiques [19, 20, 47]. On ne peut terminer cette section sans parler du travail intéressant de Sugihara et Suzuki [148, 149]. Ils ont développé une approche pour la coordination de mouvement d'un groupe de robots mobiles. Chaque robot planifie son déplacement individuellement basé sur un but défini et la position détectée d'autres robots. Cette approche est complètement distribuée dans ce sens et montre qu'un comportement intelligent peut émerger de simples comportements individuels. Sugihara et Suzuki ont pu développer différentes formes géométriques en définissant des algorithmes simples à exécuter par un grand nombre d'agents. Leur simulation montre que les robots peuvent former des cercles, des polygones, etc.

Tous ces exemples montrent l'utilité de l'étude des sociétés animales qui a permis aux chercheurs en robotique et en cybernétique de développer des robots capables de résoudre différents problèmes avec de simples comportements de base. Cette nouvelle branche de la robotique appelée robotique collective est abordée plus en détail dans la prochaine section.

## I.6. Robotique collective

La robotique collective est née de l'observation des sociétés animales telles que les fourmilières ou les termitières. L'intelligence collective (voir § I.2.3) solutionne donc le problème inhérent à l'utilisation de plusieurs robots en concevant des groupes de robots simples, robustes et peu onéreux, par opposition aux robots "omniscients" mais fragiles et coûteux. Ces robots sont dotés de moyens sensoriels élémentaires leur permettant de réagir aux changements de leur environnement et de communiquer de manière sommaire avec leurs congénères. Le but en fait de l'intelligence collective est d'exploiter les capacités d'auto-organisation qui sont générées grâce à une multitude d'interactions (simultanées ou successives) entre les agents, ou entre les agents et leur environnement. Il en résulte des performances collectives largement supérieures à celle des agents, telle que la production de formes.

Depuis quelques années, beaucoup d'activités de recherche ont été menées afin de réaliser des systèmes à plusieurs robots mobiles dotés d'un comportement collectif [109]. De tels systèmes sont intéressants pour diverses raisons :

- (1) les tâches peuvent être trop complexes pour un seul robot ou alors de meilleures performances peuvent être réalisées en utilisant plusieurs robots ;
- (2) la conception et l'utilisation de plusieurs robots simples peuvent être plus faciles ce qui les rend peu coûteux, plus flexibles et plus tolérants aux pannes.

### I.6.1. Définition

Les systèmes multi-robots sont dotés de comportement collectif caractérisé par la coopération. Le dictionnaire de Webster [117] définit "coopérer" comme étant "*s'associer avec un autre ou d'autres pour un bénéfice économique mutuel*". Des définitions explicites de la coopération dans la littérature robotique (peu abondante) incluent :

- (i) un comportement collaboratif utilisé pour atteindre un but où il existe un intérêt commun ou une récompense [25] ;
- (ii) une forme d'interaction, souvent basée sur la communication [109] ; et
- (iii) se rassembler pour fournir un travail qui crée un résultat progressif comme l'augmentation de performance ou un gain en temps [131].

Ces définitions montrent le large éventail de perspectives. Les définitions de la première catégorie (i) conduisent typiquement à l'étude de décomposition de tâches, d'allocation de tâches et d'autres axes de l'IAD (e.g., apprentissage et rationalité). Des définitions du type (ii) reflètent l'intérêt pour l'information ou d'autres ressources et peuvent être accompagnées par des études relatives à la tolérance aux pannes par exemple. Finalement, la définition (iii) permet par exemple l'augmentation de la rapidité afin de terminer au plus tôt une tâche donnée. Ainsi, dans ces définitions, on retrouve trois points fondamentaux : la tâche, le mécanisme de coopération et la performance du système.

Un comportement coopératif est alors défini comme suit : étant donnée une tâche spécifiée par un utilisateur, le système multi-robots manifeste un comportement coopératif si grâce à un certain mécanisme sous-jacent (i.e., le mécanisme de coopération), il y a une augmentation dans l'utilité globale du système [109, 111].

### I.6.2. Logique de fonctionnement de l'intelligence collective

La logique de fonctionnement de l'intelligence collective auto-organisée est une logique décentralisée fondée sur la coopération d'unités disjointes, simples et aléatoires, distribuées dans l'environnement, ne disposant que d'informations locales [32]. Chaque agent possède dès lors un équipement sensoriel et de communication relativement simple qui lui permet de répondre localement à deux types de signaux :

- (i) des signaux provenant de l'environnement non social, sur lesquels il construit ses réponses ;
- (ii) des signaux attractifs-répulsifs, inhibiteurs-activateurs, etc. émis par ses congénères.

Dans les systèmes à robots multiples, la solution n'est pas programmée explicitement mais émerge de par le grand nombre d'interactions et parce que chaque agent répond localement aux signaux environnementaux et sociaux. La résolution du problème posé se

ramène à un processus de réorganisation des agents du groupe et de l'environnement qui aboutit à une configuration-solution.

Il faut noter que la résolution collective d'une tâche n'est pas nécessairement coopérative : il se peut que la multiplication des agents soit tout simplement un moyen d'augmenter la rapidité de la résolution (comme par exemple des robots tondeuses), mais la performance collective est alors qualitativement similaire à celle d'un agent isolé ; il se peut également que le processus de résolution ne soit pas explicitement coopératif, mais qu'un "effet de coopération" émerge (cas de certains exemples de fourragement chez les fourmis). Certaines tâches ne peuvent cependant pas être réalisées sans coopération explicite : c'est le cas du transport collectif d'objets trop lourds.

Une solution de type "intelligence collective" offre alors plusieurs avantages :

- (i) la fiabilité du système est augmentée d'une part par la grande redondance des unités et d'autre part par l'augmentation de la fiabilité de chaque unité due à sa simplicité (comportements et communications).
- (ii) Le groupe réagit directement à l'environnement (flexibilité).
- (iii) Robustesse aux pannes : ceci conduit à envisager des applications où il est difficile d'avoir accès aux agents sur leur site de travail ou encore de les en ramener.
- (iv) Coût réduit des agents.
- (v) Opérationnalité immédiate : l'association de robots est une solution pour développer des outils opérationnels avec les agents à capacité limitées.

Un problème donné peut combiner plusieurs de ces avantages mais également présenter un mélange de conditions favorables à l'application de concepts de l'intelligence collective. En fait, il ne suffit pas que les agents communiquent pour acquérir les propriétés décrites ci-dessus. Il est donc nécessaire de rechercher les conditions de fonctionnement collectif, c'est-à-dire de déterminer les communications et les comportements qui permettront à un groupe de robots de résoudre un problème.

### **I.6.3. Position du problème**

Chercher un mécanisme de coopération peut être redéfini comme étant "le problème de conception de comportements coopératifs" : ayant un groupe de robots, un environnement et une tâche, comment un comportement coopératif survient ? Dans un certain sens, chaque travail en robotique coopérative a traité certaines facettes de ce problème et la plupart des axes de recherche du domaine découlent de ce problème.

Premièrement, la réalisation d'un comportement coopératif doit compter sur une infrastructure qui est l'architecture de groupe. Cette architecture englobe certains concepts comme l'hétérogénéité/homogénéité des robots, la capacité du robot à reconnaître et modéliser d'autres robots, ou la structure de communication. Deuxièmement, pour plusieurs robots qui partagent le même environnement, manipulent des objets dans l'environnement et probablement communiquent entre eux, un mécanisme est requis pour résoudre les conflits de ressources. Le troisième axe de recherche, origine de la coopération, traite les manières dont le comportement coopératif est obtenu. L'apprentissage est la quatrième clé pour réaliser un comportement coopératif parce que l'adaptabilité et la flexibilité sont des traits essentiels dans la résolution de tâches par un groupe de robots.

### I.6.4. Architecture de groupe

L'architecture d'un système multi-robots fournit le cadre nécessaire pour définir le type de comportement collectif souhaité et détermine les capacités et les limitations du système. L'architecture peut être définie comme la partie fixe du système qui ne peut être changée que par l'intervention d'un opérateur externe [163]. Le choix d'une architecture donnée pendant la phase de spécification permet donc de définir les modalités de coopération et de spécifier les paramètres conditionnant la nature des interactions entre les différentes entités du monde. Certaines caractéristiques clés de ces architectures pour les SRAs sont présentées à savoir : centralisation/décentralisation, différenciation, communication et capacité de modéliser les autres agents.

#### I.6.4.1. Centralisation/décentralisation

La plus importante décision à prendre quand on définit une architecture de groupe est : est-ce que le système est centralisé ou décentralisé et s'il est décentralisé est-ce que le système est hiérarchique ou distribué. Les architectures centralisées sont caractérisées par un seul agent de contrôle. Les architectures décentralisées sont démunies d'un tel agent. Il y a deux types d'architectures décentralisées : les architectures distribuées dans lesquelles tous les agents sont égaux d'un point de vue du contrôle et les architectures hiérarchiques qui sont localement centralisées.

Actuellement, le paradigme dominant est l'approche décentralisée. Le comportement des systèmes décentralisés est souvent décrit par certains termes telles que l'émergence et l'auto-organisation. On a souvent avancé que les architectures décentralisées (e.g., [9, 27, 109, 143]) ont divers avantages inhérents que les architectures centralisées ne possèdent pas telles que la tolérance aux pannes, l'exploitation naturelle du parallélisme et la fiabilité.

##### I.6.4.1.1. Systèmes centralisés

Un système centralisé est un système ayant plusieurs entités logiques ou physiques telles que chacune de ces entités envoie ses informations locales et reçoit ses actions futures d'un seul et unique processus central (Figure I.5). En effet, dans un tel système, une seule entité est capable de modéliser le monde et par conséquent d'agir *intelligemment* sur son environnement. Toutes les autres entités ne sont reconnues par le processus central que comme des actionneurs et des capteurs faisant partie de l'environnement et non pas comme des entités indépendantes [147].

Une architecture centralisée suppose que le processus central a une vue globale du monde à tout instant et que les attributs du monde sont figés entre le moment de la planification et celui de l'exécution. Avoir une vue complète du monde permet au processus central de mieux gérer ses ressources afin de produire théoriquement la solution *optimale* au problème posé.

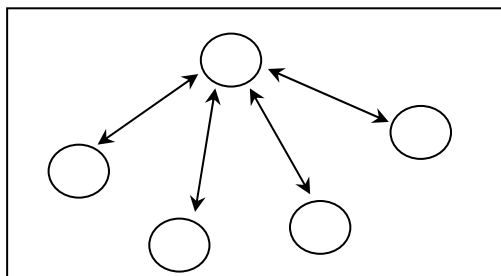


Figure I.5. Système centralisé.

### I.6.4.1.2. Systèmes décentralisés

Un système décentralisé comporte plusieurs entités logiquement et/ou physiquement indépendantes ayant un pouvoir décisionnel et d'action sur leur environnement. D'un point de vue individuel, la dynamique de l'environnement dans un système décentralisé est déterminée par les autres agents d'une manière imprévisible.

On peut classer les systèmes décentralisés en deux catégories suivant le niveau d'autonomie associé à chaque agent [42] :

- Systèmes décentralisés hiérarchiques ;
- Systèmes décentralisés distribués.

#### *Systèmes décentralisés hiérarchiques*

Un système décentralisé est dit hiérarchique (localement centralisé) si le pouvoir décisionnel est distribué sur un sous-ensemble d'agents appelés coordinateurs (Figure I.6). Ces agents jouent localement le rôle du processus central dans les systèmes centralisés en planifiant les tâches de leurs "esclaves" pour produire des plans partiels.

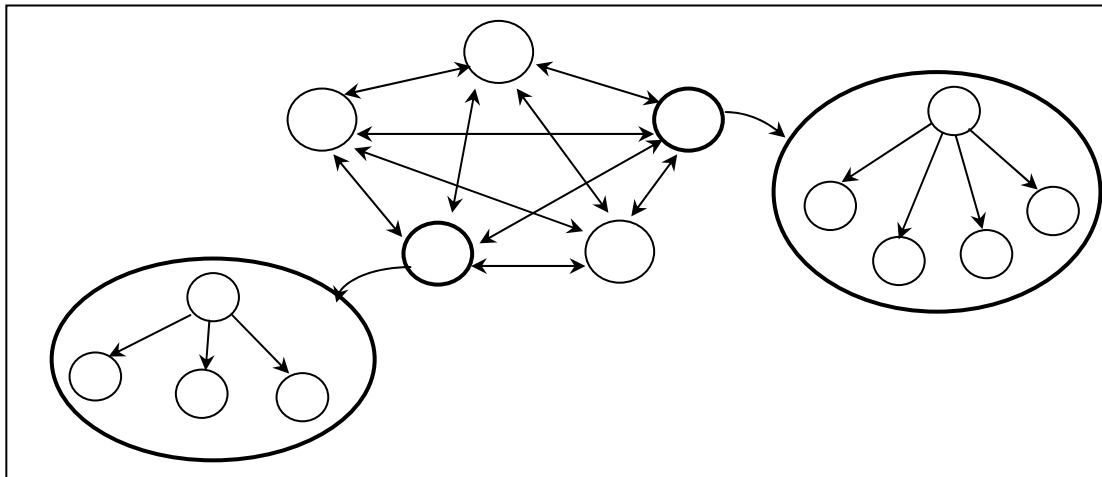


Figure I.6. Système décentralisé hiérarchique.

La hiérarchisation décisionnelle permet globalement d'améliorer les performances du système par la génération de plans partiels optimaux. Le problème réside dans la manière avec laquelle ces coordinateurs concilient ces plans partiels pour produire un plan consistant. La fusion de plans se fait par négociation en se basant sur un processus d'optimisation du coût global, en optimisant les coûts locaux [63].

Une autre manière de réaliser cette structure hiérarchique est de distribuer la communication entre les agents (cas de ACTRESS) [161]. L'architecture décrit un arbre hiérarchique où les agents correspondent aux nœuds et où les arcs définissent la nature des liens entre ces agents. Un nœud de niveau  $N$  est considéré comme l'esclave du nœud père de niveau  $(N-1)$  et le coordinateur (médiateur) des nœuds fils du niveau  $(N+1)$ . Les agents respectent cette hiérarchie et ne peuvent communiquer qu'à travers les arcs de cet arbre [62].

#### *Systèmes décentralisés distribués*

Un système décentralisé est dit distribué ou plat si tous les agents sont égaux au niveau de la prise de décision et du contrôle (Figure I.7). Chaque agent a un niveau d'autonomie qui lui permet de produire ses propres plans en fonction de ses propres buts [87].



Dotant l'agent d'une grande autonomie, la capacité de percevoir l'environnement, de raisonner sur la tâche, de prendre des décisions et finalement d'exécuter le plan produit, permet au système d'afficher des propriétés séduisantes du point de vue de l'agent (fiabilité, flexibilité). Cependant, cela crée des difficultés importantes du point de vue de la mission (cohérence, convergence) [125].

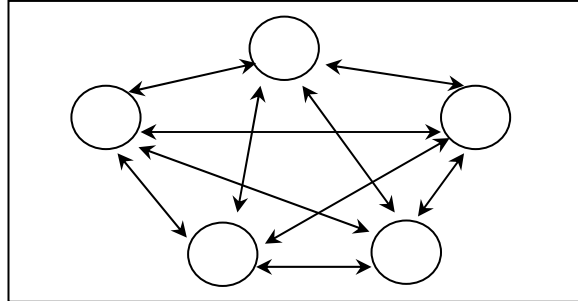


Figure I.7. Système distribué.

Dans le domaine de la robotique, plusieurs travaux se basent sur de tels systèmes. En effet, Beni [30] a abordé le domaine de l'intelligence distribuée émergente (robotique en essaim). Brooks [38] utilise l'architecture distribuée multi-couches [39] afin de construire des robots maçons dédiés à la construction d'une base lunaire. Wang [164] propose un ensemble de primitives génériques pour les systèmes robotiques distribués. Rosenchein [136] utilise la théorie des jeux pour modéliser les interactions entre agents.

Tous ces exemples s'inspirent en fait de ce qui se passe souvent dans le monde animal (colonie d'insectes, vol en formation, etc.), de façon distribuée en laissant à chaque robot le soin d'agir en fonction de la perception locale qu'il a de son environnement.

Ainsi, un contrôle centralisé considère le groupe de robots comme un "super-robot" dont l'espace des états correspond au produit des espaces d'états de tous les robots le constituant. La taille de l'espace d'états peut être très grande et la planification centralisée de la moindre tâche devient rapidement embrouillée. L'idée d'un contrôle distribué se révèle donc plus attrayante [32].

#### I.6.4.2. Différentiation

La différenciation peut être définie comme étant la capacité des robots à avoir des fonctions différentes à accomplir au sein du groupe.

Un groupe de robot est dit homogène si les capacités de chaque robot sont identiques et hétérogène dans le cas contraire. En général, l'hétérogénéité introduit une plus grande complexité puisque l'allocation de tâches devient plus difficile et les agents ont plus besoin de modéliser les autres individus du groupe. La littérature actuelle est beaucoup plus dominée par des travaux qui supposent l'utilisation de groupes de robots homogènes [21, 109]. Cependant, certaines architectures particulières peuvent prendre en charge l'hétérogénéité, e.g., ACTRESS et ALLIANCE (voir § I.6.4.1). Dans les groupes hétérogènes, l'allocation de tâches peut être déterminée par les capacités individuelles alors que dans les systèmes homogènes, les agents peuvent avoir besoin de se différencier en des rôles distincts qui peuvent être connus lors de la conception ou survenir dynamiquement au cours de l'exécution.

### I.6.4.3. Structures de communication et interaction des robots

La structure de communication d'un groupe détermine les modes possibles d'interaction inter-agents. Il existe trois types d'interactions:

- Interaction via l'environnement,
- Interaction via la perception,
- Interaction via la communication.

#### I.6.4.3.1. Interaction via l'environnement

Le premier mécanisme d'interaction entre robots est à travers leur environnement partagé. Cette forme d'interaction est indirecte puisqu'il n'y a aucune communication explicite ou une interaction physique entre agents. C'est une puissante méthode d'interaction qui peut être utilisée par de simples agents sans capacités de raisonnement complexe ou de communication directe.

La stigmergie est un exemple d'interaction à travers l'environnement employée dans diverses sociétés d'insectes. Cette notion est à l'origine utilisée afin de décrire le comportement de construction du nid des termites et des fourmis [66]. La stigmergie est définie comme étant "*le processus par lequel la coordination de tâches et la régulation de construction ne dépend pas directement des travailleurs mais des constructions elles-mêmes*" [88]. En effet, il a été montré que la coordination de l'activité de construction dans une colonie de termites n'est pas inhérente aux termites elles-mêmes mais plutôt aux mécanismes de coordination qui sont régulés par l'environnement de la tâche, dans ce cas de figure l'évolution de la structure du nid [32].

Des exemples de systèmes artificiels où les agents interagissent via l'environnement incluent la construction distribuée [32], le tri [56, 57], le rassemblement [27], la manipulation d'objets [58], l'analyse de la congestion dans les réseaux informatiques [84], etc. D'autres systèmes qui dépendent de ce type d'interaction sont également décrits dans [9, 27, 144, 159].

#### I.6.4.3.2. Interaction via la perception

Le second mécanisme d'interaction entre robots est à travers la perception. Comme décrit par Cao *et al.* [42], l'interaction via la perception "se réfère à une interaction locale qui survient entre agents comme résultat de la perception mutuelle de l'un par rapport à l'autre mais sans communication explicite". Cette forme d'interaction est également indirecte puisqu'il n'y a aucune communication explicite entre agents ; cependant, elle requiert de chacun des agents d'être capable de distinguer d'autres agents de divers objets dans l'environnement. L'interaction via la perception peut être utilisée par un agent afin de modéliser le comportement d'un autre agent ou afin de déterminer ce que fait un autre agent pour prendre une décision et répondre convenablement à une demande donnée. Par exemple, le regroupement d'oiseaux utilise la perception pour surveiller les actions d'autres oiseaux dans leur voisinage afin de corriger leurs propres mouvements. Il a été montré qu'un regroupement efficace résulte de simples règles locales suivies par chacun des oiseaux de la société, répondant à la direction et à la vitesse des voisins locaux [132]. De telles méthodes d'interaction via la perception sont utilisées dans le regroupement des robots mobiles, le suivi, et le fourragement [108], les robots footballeurs [169], les formations de robots [21, 68] et des simulations d'animations d'un banc de poissons [158]. D'autres applications de l'interaction via la perception incluent l'interaction visuelle entre agents physiques [68, 121] incluant la capacité à comprendre et influencer les raisons d'autres agents physiques [123].

#### **I.6.4.3.3. Interaction via la communication**

Le troisième mécanisme d'interaction entre agents est via la communication directe. Les agents s'adressent directement aux autres agents, d'une manière spécifique au système ou via un protocole de communication standard tel que KQML [65]. De tels agents orientés communication peuvent être utilisés afin de demander des informations ou des actions d'autres agents ou répondre à une requête reçue des autres agents.

Plusieurs axes du domaine des réseaux en télécommunication sont exploités, incluant la conception des topologies des réseaux et des protocoles de communication parce que les architectures qui utilisent cette forme de communication sont similaires aux réseaux (par exemple, Wang [165] proposa une communication inter-robot en utilisant un protocole "media access protocole" similaire à Ethernet). Dans [86], des robots avec une plage de communication limitée communiquent entre eux en utilisant le protocole 'hello-call' par lequel les robots établissent des chaînes pour étendre leurs plages effectives. Gage [75] décrit des méthodes pour communiquer entre des milliers de robots. Arai *et al.* [7] proposent un protocole de communication modélisé après diffusion chimique.

#### **I.6.4.3.4. De l'interaction de l'agent à la "cognition" et au comportement**

Dans les systèmes où un comportement global et complexe émerge à partir d'interactions d'une société de simples agents (comme la fonction complexe du cerveau émerge d'interactions d'une large société de neurones) la complexité résultante ne peut pas être attribuée à un seul agent mais plutôt à l'interaction de tous les agents. Les agents et leurs interactions (souvent locales) entre eux et avec l'environnement génèrent le comportement global du système- sans un mécanisme externe de coordination. L'interaction dans une société d'agents peut être réalisée, comme on vient de le voir, par différents mécanismes incluant l'interaction via l'environnement, via la perception ou via la communication. L'interaction entre les membres d'une société ne suffit donc pas à elle seule afin de produire un résultat global intéressant ou utile. Pour que ces agents produisent un comportement cohérent, un certain mécanisme de coordination qui organise les interactions en temps et en espace est nécessaire. Plutôt que d'avoir un coordinateur central, une société d'agents coordonne ses interactions pour produire un comportement désiré du système à travers ce type de mécanismes tels que l'auto-organisation, les techniques de "machine learning" ou encore par des mécanismes de négociation plus complexes.

En effet, les techniques d'auto-organisation sont basées sur un ensemble de mécanismes dynamiques par lesquels des structures apparaissent au niveau global du système à partir d'interactions entre ses composantes de bas niveau. Les règles spécifiant les interactions entre les unités constituant le système sont exécutées à la base d'informations purement locales sans se référer au modèle global. Ce dernier est une propriété émergente du système plutôt qu'une propriété imposée au système par une influence externe [32]. Des méthodes, tels que les algorithmes génétiques [83] et l'apprentissage par renforcement [150], peuvent être utilisées pour concevoir des agents et leurs interactions de sorte que le comportement résultant corresponde aux objectifs du système. Les agents peuvent également négocier entre eux des ressources et l'allocation de tâches afin de coordonner leur comportement. Plusieurs approches ont vu le jour telles que la coordination basée sur le marché [78] ou la théorie des jeux [135]. Dans la plupart de ces modèles de coordination par négociation, les agents raisonnent à partir des croyances, des désirs et des intentions des autres agents et influencent ceux utilisant des techniques spécialisées [33].

#### I.6.4.4. Modélisation des autres agents

La modélisation des intuitions, des croyances, des actions, des capacités, et des actions des autres agents peut mener à une meilleure coopération effective entre agents. Les exigences en communication peuvent aussi être réduites si chaque agent a la capacité de modéliser d'autres agents. Notons que la modélisation d'autres agents entraîne plus qu'une communication implicite via l'environnement ou une perception. La modélisation requiert que le "modeleur" a une certaine représentation d'un autre agent et que cette représentation peut être utilisée pour tirer des conclusions (faire des inférences) concernant les actions des autres agents. C'est aussi une caractéristique qui n'est utilisée que dans les systèmes hétérogènes, puisque les systèmes homogènes, les agents étant tous identiques, n'ont aucun besoin de modéliser les uns les autres.

#### I.6.4.5. Architectures de contrôle

Il existe quatre classes fondamentales d'architectures de contrôle et qui sont :

- Architecture de contrôle réactive (ne pas penser, mais agir).
- Architecture de contrôle délibérative (penser puis agir).
- Architecture de contrôle hybride (penser et agir indépendamment en parallèle).
- Architecture de contrôle comportementale (penser la façon d'agir).

Chacune des architectures possède des avantages et des inconvénients et toutes jouent un rôle important dans différents problèmes et applications. Le contrôle du robot implique les compromis inévitables suivants :

- (i) Penser est lent mais la réaction doit souvent être rapide,
- (ii) Penser permet de planifier afin d'éviter les mauvaises actions. Mais penser trop longtemps peut être dangereux (e.g., tomber).
- (iii) Pour penser, le robot a besoin d'une grande quantité d'informations précises. L'information doit être mise à jour activement. Mais le monde change quand le robot pense, donc plus il pense plus la solution est imprécise.

Certains robots ne pensent pas du tout mais exécutent tout simplement des réactions pré-programmées alors que d'autres pensent beaucoup et agissent peu. La plupart des robots actuels se trouvent entre ces deux extrêmes et beaucoup utilisent la pensée et la réaction.

##### I.6.4.5.1. Architectures de contrôle réactives (ne pas penser mais agir)

L'architecture de contrôle réactive est utilisée lorsque le robot a besoin de répondre immédiatement à un stimulus (voir figure I.8). Elle est généralement réalisée par un fort couplage entre les entrées des capteurs et les sorties des actionneurs, afin de permettre au robot de répondre très rapidement aux environnements non-structurés et changeants [39]. C'est une méthode de contrôle puissante, très utilisée par les robots situés qui est également très présente chez une grande partie d'animaux. Les limitations d'une telle architecture incluent l'incapacité du robot à stocker un grand nombre d'informations, à établir des représentations internes du monde [36], ou à apprendre hors fonctionnement. Un compromis est fait en faveur d'un temps de réaction rapide contre un raisonnement complexe. Une analyse formelle a montré que pour des environnements et des tâches qui peuvent être caractérisés à priori, des architectures de contrôle réactives sont plus puissantes [2, 138]. Si de plus, ces architectures sont bien structurées, elles sont capables de performance optimale pour certaines classes de problèmes. Cependant, dans d'autres types d'environnements et de tâches

où des modèles internes, une mémoire, ou un apprentissage sont requis, le contrôle réactif est insuffisant.

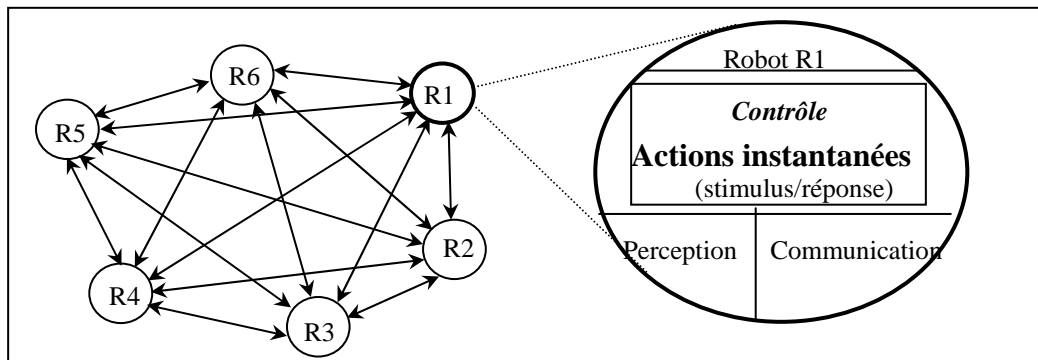


Figure I.8. Architecture réactive.

#### I.6.4.5.2. Architectures de contrôle délibératives (penser puis agir)

Dans ce type d'architecture, le robot utilise toutes les informations sensorielles disponibles et toutes les connaissances internes sauvegardées et raisonne sur les prochaines actions à réaliser (voir figure I.9). Le raisonnement est souvent sous forme de planification, demande de recherche de séquences d'états-actions possibles et leurs résultats. La planification, une composante essentielle en intelligence artificielle, est connue pour son exigence en temps de traitement. Le robot doit construire et ensuite évaluer tous les plans possibles jusqu'à trouver celui qui lui donne comment atteindre son but, résoudre le problème ou décider d'une trajectoire à exécuter. La planification requiert l'existence d'une représentation interne du monde qui permet au robot d'avoir une idée du futur et de prédire les résultats d'actions possibles dans différents états de manière à générer des plans. Le modèle interne doit donc être précis et récent. Quand le temps est suffisant afin de générer un plan et le modèle du monde précis, cette approche permet au robot d'agir stratégiquement sélectionnant la meilleure action pour une situation donnée. Mais comme en général, le robot est situé dans un monde bruité et dynamique, ce type de contrôle est souvent impossible bien qu'il existe quelques robots situés complètement délibératifs.

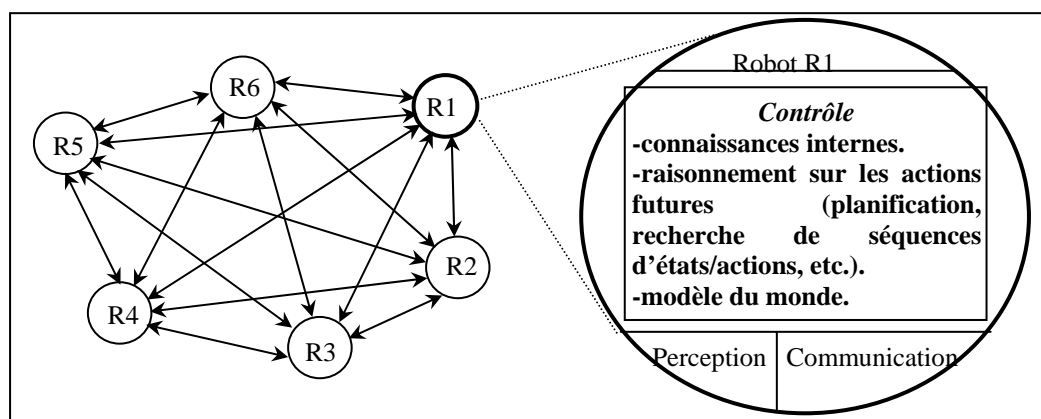


Figure I.9. Architecture délibérative.

#### I.6.4.5.3. Architectures de contrôle hybrides (penser et agir indépendamment en parallèle)

L'architecture de contrôle hybride combine les meilleurs aspects des architectures réactive et délibérative : elle combine la réponse temps-réel de la réactivité avec la rationalité

et l'efficacité de la délibération (voir figure I.10). En effet, la composante réactive s'occupe des besoins immédiats du robot tel que l'évitement d'obstacles et fonctionne ainsi en un temps court et utilise des données et signaux externes directs, alors que la composante délibérative utilise des représentations internes, abstraites et symboliques du monde et fonctionne en un temps plus long. Tant que les sorties des deux composantes ne sont pas en conflit, le système n'a pas besoin de plus de coordination. Cependant, ces deux parties doivent interagir si elles tirent profit l'une de l'autre. La composante réactive doit outrepasser la composante délibérative si le monde présente un certain défi immédiat et imprévisible et la partie délibérative doit informer la partie réactive afin de guider le robot vers des trajectoires et des buts plus efficaces. L'interaction des deux parties de l'architecture requiert une composante intermédiaire dont sa construction est particulièrement le plus grand défi dans la conception hybride. Ainsi, les architectures hybrides sont souvent appelées "architectures à trois couches", composées des couches réactive, intermédiaire et délibérative. Un grand effort de recherche a été conduit dans la manière de concevoir ces composantes et leurs interactions [10].

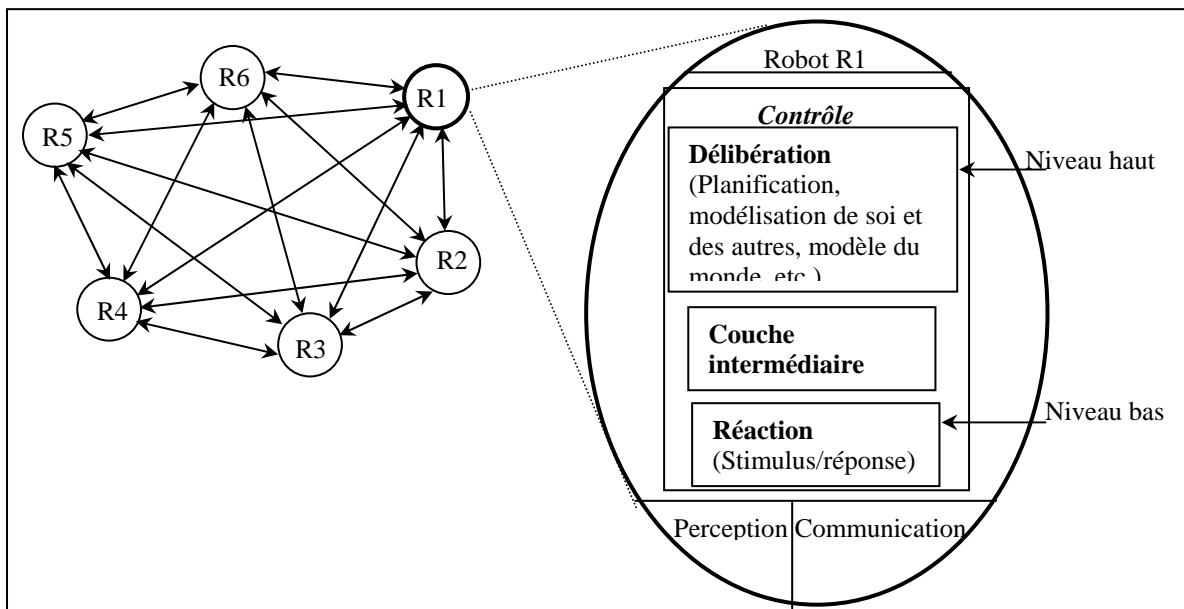


Figure I.10. Architecture hybride.

#### I.6.4.5.4. Architectures de contrôle comportementales (penser la façon d'agir)

L'architecture de contrôle comportementale tire son inspiration de la biologie et essaie de modéliser le comportement des animaux dans leurs environnements complexes d'évolution. Les composantes de base de ces architectures sont appelées comportements : ceux-ci sont des "patterns" observés d'activité émergeant des interactions entre le robot et son environnement (voir figure I.11). De tels systèmes sont construits d'une façon ascendante commençant par un ensemble de comportements de survie tel que l'évitement de collisions qui couple les entrées perçues aux actions du robot. Des comportements sont ajoutés pour fournir au robot plus de capacités lui permettant de réaliser des tâches de plus en plus complexes tels que le suivi de mur, la poursuite d'une cible, l'exploration ou le "homing" (rentrer chez-soi). Ces nouveaux comportements sont introduits dans le système d'une manière incrémentale des plus simples aux plus compliqués jusqu'à ce que leurs interactions aboutissent aux capacités désirées du robot. Comme les architectures hybrides, les architectures comportementales peuvent être organisées en couches mais à l'inverse de ces

architectures, les couches ne diffèrent pas beaucoup les unes des autres en termes de temps et de représentations utilisées. Toutes les couches sont représentées par des comportements, des processus qui récoltent des entrées et s'envoient des sorties.

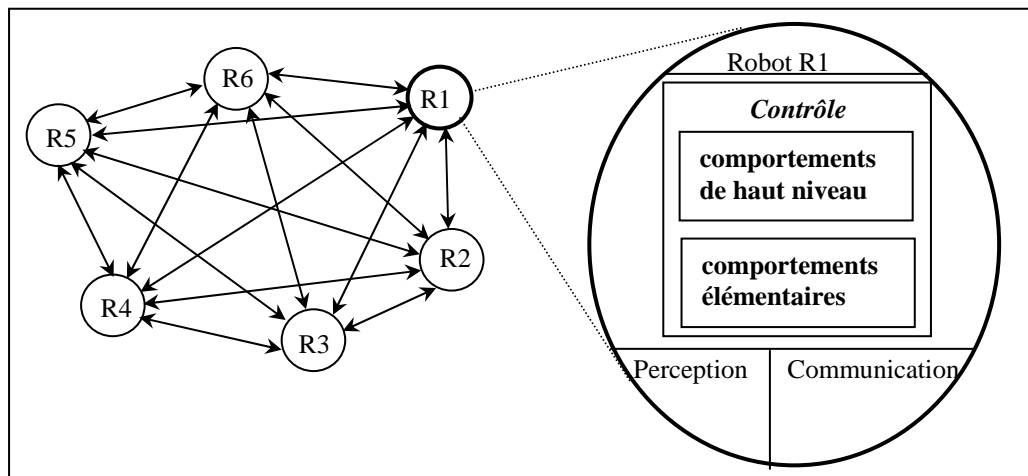


Figure I.11. Architecture comportementale.

Les architectures comportementales et les architectures réactives partagent certaines propriétés : toutes les deux sont construites d'une manière incrémentale et ascendante, et sont constituées de modules distribués. Cependant, les architectures comportementales sont fondamentalement plus puissantes parce qu'elles peuvent stocker des représentations [112] alors que les architectures réactives ne le peuvent pas. Les représentations dans les architectures comportementales sont stockées d'une façon distribuée de sorte à faire correspondre le bon comportement qui cause la bonne réaction du robot. Ainsi, si le robot a besoin de planifier à l'avance, il le fait donc dans un réseau par communication de comportements plutôt qu'un simple planificateur centralisé. Si le robot a besoin de sauvegarder une large carte, la carte peut être distribuée sur plusieurs modules comportementaux. Ceux-ci représentent les composantes telles que les bornes, comme présenté dans [113], qui permettent de raisonner activement sur la carte par exemple en utilisant la technique par passage de message du réseau de bornes. Ainsi, les composantes de la planification et du raisonnement du système utilisent les mêmes mécanismes tels que la perception et les comportements orientés action et ainsi opèrent sur la même représentation. Dans ce sens, "penser" est organisé de la même manière que "agir".

#### I.6.4.5.5. Comparaison et discussion

Les architectures comportementales et hybrides ont les mêmes capacités expressives et "computationnelles" : toutes les deux peuvent stocker des représentations et anticiper. Mais elles travaillent de manières très différentes et les deux approches ont trouvé différentes applications dans le domaine de la robotique mobile et dans d'autres domaines d'application. En effet, les architectures hybrides dominent le domaine du contrôle d'un seul robot bien que celui-ci demande quelques fois qu'une architecture réactive soit utilisée. Les architectures comportementales quant à elles dominent le domaine du contrôle multi-robots parce que la notion de collection de comportements dans l'architecture permet de contrôler et de prendre en compte des populations de robots induisant un comportement de groupe robuste et adaptatif.

Le temps dont dispose le robot (ou ne dispose pas) détermine quel type d'architecture est la plus appropriée. En effet, les systèmes réactifs sont le meilleur choix pour des environnements demandant des réponses très rapides ; cette capacité est obtenue au prix d'une perte d'information ou de connaissance sur le passé ou le futur. Les systèmes réactifs sont aussi un choix souvent utilisé dans les environnements hautement stochastiques et les environnements qui peuvent être bien caractérisés de manière à être codés en une cartographie (plan, mapping) d'entrée-sortie réactif. Les systèmes délibératifs d'un autre côté est le meilleur choix pour des domaines qui requièrent beaucoup de stratégie, d'optimisation et en retour recherchent et planifient. De tels domaines, cependant, ne sont pas typiques dans la robotique située mais plus dans l'ordonnancement, et la théorie des jeux par exemple. Les systèmes hybrides quant à eux sont bien appropriés pour des environnements et des tâches où des modèles internes et une planification peuvent être employés et les demandes en temps-réel ne sont pas nombreuses ou sont suffisamment indépendantes du raisonnement de plus haut niveau. Alors que, les systèmes comportementaux "qui pensent alors qu'ils agissent", au contraire sont plus appropriés aux environnements avec des changements dynamiques très significatifs où une réponse rapide et une adaptativité sont nécessaires mais la capacité de prévoir le futur et éviter quelques erreurs du passé est également requise. Ces capacités s'étendent aux comportements actifs en utilisant si nécessaire des représentations actives [107]. Ainsi, ces systèmes "pensent la manière d'agir".

Actuellement, les approches utilisées sont très souvent comportementales fournissant des contrôleurs distribués, inspirés de la biologie et qui peuvent être étendus à un très grand nombre de robots qui tirent avantage de l'interaction dynamique avec l'environnement plutôt qu'un raisonnement explicite et/ou une planification. En effet, différentes applications sont basées sur ce type d'approches. On cite, entre autres, les robots footballeurs [11], la robotique coopérative [78, 94] et la robotique humanoïde [35, 37]. Dans tous ces exemples, le fait d'être situés dans un environnement dynamique tout en essayant d'exécuter une tâche avec sécurité présente un certain nombre de défis qui requièrent un contrôle temps-réel, adaptatif et robuste. La capacité d'améliorer la performance dans le temps, dans le contexte d'un environnement changeant et dynamique est aussi un important axe de recherche dans la robotique située. Contrairement à l'apprentissage classique où le but est d'optimiser la performance en un laps de temps relativement long, dans l'apprentissage situé l'objectif est de s'adapter plus rapidement, réalisant une grande efficacité même en présence d'incertitude. Des modèles de la biologie sont souvent considérés et les modèles d'apprentissage par renforcement sont particulièrement populaires avec leur capacité à apprendre directement à partir de la réponse de l'environnement [150]. Cet axe continue à se développer et traite des problèmes de contrôle de robots de plus en plus complexes. Pour plus de détail, le lecteur peut se référer aux articles de Arkin et Mataric [9, 106].

### **I.6.5. Conflits de ressources**

Quand une seule ressource indivisible est requise par plusieurs robots, un conflit de ressource survient. Cet axe a été étudié dans divers thèmes particulièrement dans le problème de l'exclusion mutuelle dans les algorithmes distribués et dans le problème multi-accès dans les réseaux à micro-ordinateurs. Avec plusieurs robots, les conflits de ressources surviennent quand il y a un besoin de partage d'espace, d'objets à manipuler ou de moyens de



communication (voir figure I.12). Quelques travaux ont traité spécifiquement le problème du partage d'objets ou du partage des moyens de communication [52, 139].

Dans un système multi-robots, chaque robot peut planifier un chemin et expliquer aux autres robots et à l'environnement global via la configuration espace-temps, des modèles explicites d'autres agents, ou d'autres techniques. Cependant, les chercheurs en considérant les systèmes réels multi-robots concluent que la planification de chemins en avance est impossible. Ainsi, les robots sont souvent restreints à suivre des chemins pré-définis avec des règles (comme les lois du trafic dans le monde des humains) et utiliser des communications pour éviter les collisions et les impasses (serrure) [40].

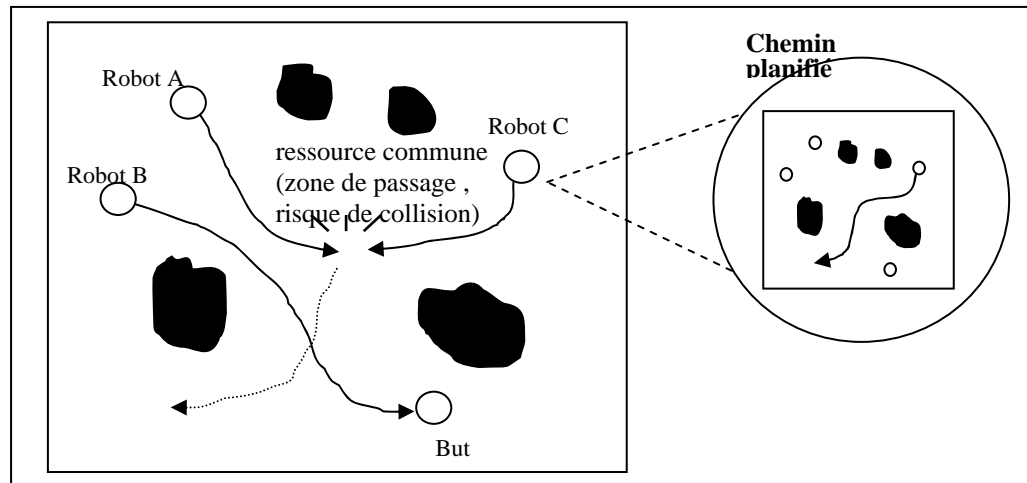


Figure I.12. Figure avec ressource commune, i.e., même zone.

### I.6.6. Origine de la coopération

Dans la plupart des travaux en robotique collective, il est supposé que la coopération est explicitement conçue dans le système. Un problème intéressant est d'étudier comment la coopération peut survenir sans une motivation humaine explicite parmi les agents intéressés possibles.

McFarland [114] distingue entre deux types complètement différents de comportements de groupe qui se trouvent dans la nature: le comportement eusocial et le comportement coopératif. Le comportement eusocial se trouve dans plusieurs espèces d'insectes (e.g., les colonies de fourmis ou d'abeilles), et est le résultat d'un comportement individuel génétiquement déterminé. Dans les sociétés eusociales, les individus ne sont pas très capables mais en apparence (paradoxalement) un comportement intelligent résulte de leurs interactions. Ce comportement "coopératif" est nécessaire pour la survie des individus dans la colonie [170].

D'un autre côté, McFarland [114] définit le comportement coopératif comme le comportement social observé dans les animaux supérieurs (les vertébrés); la coopération est le résultat d'interactions entre agents intéressés. A l'inverse du comportement eusocial, le comportement coopératif n'est pas motivé par un comportement inné, mais par un désir intentionnel de coopérer pour maximiser l'utilité individuelle.

### I.6.7. Coopération

La coopération consiste à amplifier les capacités des SRAs pris isolément et à augmenter leurs performances.

La coopération est la forme générale d'interaction la plus étudiée dans les systèmes multi-robots. Elle consiste à faire en sorte que des robots travaillent à la satisfaction d'un but commun, voire à la satisfaction de tous les buts individuels. De manière simplifiée, le problème de la coopération peut se ramener à résoudre différents sous-problèmes que constituent l'allocation des tâches, la coordination et la résolution de conflits [133].

Du fait de leur autonomie, les robots sont amenés à se trouver dans des situations où leurs intérêts peuvent être contradictoires, ils sont alors en situation de conflits. Ces situations proviennent, essentiellement, d'un problème d'accès à des ressources limitées : deux robots désirent au même moment quelque chose qui ne peut pas être partagée, ou pour lequel toute répartition diminue ce que l'un des robots aurait pu obtenir si l'autre n'avait pas été présent. Pour gérer cette opposition, il est nécessaire d'utiliser des techniques de résolution de conflits.

### **I.6.7.1. Méthodes de coopération**

Comment coopérer ?

Pour répondre à cette question, on présente dans ce qui suit les moyens que l'on peut mettre en œuvre pour coopérer. Ils sont au nombre de six:

- Le regroupement et la multiplication;
- La communication ;
- La spécialisation ;
- La collaboration par partage des tâches et des ressources ;
- La coordination d'actions ;
- La résolution de conflit par arbitrage et négociation.

#### **I.6.7.1.1. Regroupement et Multiplication**

Cette méthode est la plus évidente. Elle consiste tout simplement pour les SRAs à se rapprocher physiquement. Cela revient à construire soit un bloc plus ou moins homogène dans l'espace, soit un réseau de communication permettant à plusieurs robots de se comporter comme s'ils étaient physiquement les uns à côté des autres.

Dans cette méthode, les groupes simplifient les problèmes de navigation, on peut prendre comme exemple les formations d'oiseaux migrateurs, les bancs de poissons ou les troupeaux d'éléphants. L'effort de navigation est globalement diminué si un seul (ou un nombre déterminé) décide de l'endroit où aller, les autres suivant le mouvement.

D'autre part, la multiplication, c'est-à-dire la simple augmentation quantitative des robots dans un système donné présente des avantages considérables, tant du point de vue de l'augmentation des performances que de celui de sa fiabilité. On peut prendre comme exemple, le cas de robots récupérateurs de minerai, le fait de disposer d'un grand nombre de robots n'est pas neutre quant à la fiabilité du système. Si la performance globale en cas d'absence de pannes est égale à la somme des performances individuelles, c'est-à-dire que l'augmentation différentielle de performance est nulle, il n'est plus de même si les robots peuvent avoir des pannes. Si l'on ne dispose que d'un seul robot, toute panne peut avoir des conséquences fâcheuses car la tâche ne pourra pas être effectuée. Au contraire, si l'on dispose de plusieurs robots, la redondance permettra d'accroître la fiabilité du système.

#### **I.6.7.1.2. Communication**

Le système de communication qui lie un ensemble de SRAs agit comme une sorte de système nerveux qui met en contact des agents parfois séparés. La communication en effet

agrandit les capacités perceptives des robots en leur permettant de bénéficier des informations et du savoir-faire des autres SRAs.

La communication est indispensable à la coopération. Elle constitue l'un des moyens fondamentaux pour assurer la répartition des tâches et la coordination des actions. Elle est effectuée par l'envoi de messages ou par diffusion d'un signal dans l'environnement.

#### **I.6.7.1.3. Spécialisation**

La spécialisation est le processus par lequel des SRAs deviennent de plus en plus adaptés à leurs tâches. Cependant, il est souvent difficile de construire des robots mobiles qui soient spécialisés dans toutes les tâches [133].

La spécialisation n'est pas nécessairement le fruit d'un choix à priori. Des SRAs initialement totipotents peuvent se spécialiser progressivement dans l'accomplissement de leurs tâches, ce qui revient à leur associer des rôles temporaires par une sorte d'adaptation individuelle. Cette spécialisation peut être bénéfique à la collectivité en augmentant la capacité du groupe à résoudre plus rapidement un problème donné.

#### **I.6.7.1.4. Collaboration**

La collaboration consiste à travailler à plusieurs sur un projet, une tâche commune. On appelle collaboration, l'ensemble des techniques permettant à des SRAs de se répartir des tâches, des informations et des ressources de manière à réaliser une œuvre commune.

Résoudre un problème de collaboration consiste à répondre à la question "Qui fait quoi ?" par rapport à un travail donné. Il existe de nombreuses manières de répartir des tâches. Dans les systèmes de robots mobiles, l'allocation des tâches passe par des mécanismes d'offre et de demande. On distingue alors les mécanismes de répartition centralisés, dans lesquels un robot coordinateur centralise les offres et les demandes et les répartit ensuite au mieux, et des mécanismes distribués dans lesquels un robot peut être à la fois offrant ou demandeur sans qu'il existe un organe centralisateur (voir § I.6.4.1).

#### **I.6.7.1.5. Coordination d'actions**

Gérer un ensemble de SRAs suppose l'exécution d'un certain nombre de tâches supplémentaires. Celles-ci ne sont pas directement productives mais servent simplement à faire en sorte que les actions productives puissent s'accomplir dans les meilleures conditions.

Lorsqu'il s'agit de système monolithique, ces tâches supplémentaires font partie du système organisationnel mais dans le cas d'un système multi-robots, c'est-à-dire d'un travail nécessairement distribué, on parle de tâches de coordination. Celles-ci sont indispensables dès que l'on se trouve en présence d'un ensemble de robots qui poursuivent leurs propres buts, la réalisation de tâches productives entraînant avec elles tout un cortège de coordination sans laquelle les premières ne peuvent être accomplies. La phase de coordination d'actions est directement impliquée dans la définition de l'ordre des actions à effectuer [133].

#### **I.6.7.1.6. Résolution de conflits par arbitrage et négociation**

L'arbitrage et la négociation sont les deux moyens utilisés par les systèmes multi-robots pour résoudre les conflits et empêcher que des désaccords entre robots ne se transforment en luttes ouvertes et que le système dans son entier ne dégrade ses performances. L'arbitrage conduit à la définition de règles de comportements qui agissent comme des contraintes sur

l'ensemble des SRAs. Mais, le résultat global a pour effet de limiter les conflits et de préserver à la fois les robots mais surtout les sociétés de robots.

Dans les sociétés humaines, le comportement des individus est contraint par un ensemble de lois et réglementations qui régissent les activités sociales et limitent leur activité, pour le bien de tous (cas idéal). Il existe un organe d'arbitrage, la justice, dont la fonction est de faire respecter ces règles collectives et de veiller à leur application. Cet organe agit aussi pour déterminer les responsabilités dans les conflits et décider des punitions que les personnes jugées coupables doivent subir. Dans les systèmes multi-robots, l'ensemble des règles sociales agit comme un ensemble de contraintes comportementales. Evidemment, ces lois ne peuvent s'appliquer qu'à des robots de très haut niveau, qui disposent d'une sorte de libre arbitre et sont capables de peser le pour et le contre des conséquences de leurs actions.

Lorsque des robots mobiles entrent en conflit d'objectif ou de ressources, on préfère souvent ne pas recourir à un arbitre, mais laisser plutôt les robots résoudre eux mêmes le conflit par la recherche d'un accord bilatéral au cours d'un processus de négociation.

### **I.6.7.2. Avantages de la coopération**

La coopération, en favorisant les performances collectives des SRAs, offre ainsi un certain nombre d'avantages. Ceux-ci s'expriment soit par des améliorations individuelles, telles que l'accroissement de productivité ou l'augmentation de la probabilité de survie des individus, soit par des améliorations de groupe telles que l'accomplissement des tâches impossibles autrement à réaliser. Une liste non exhaustive des conséquences positives de la coopération est présentée comme suit [32] :

- (i) Accomplir des tâches impossibles à réaliser par un seul robot. Il s'agit d'un avantage qualitatif essentiel comme on l'a vu. Dans ce type de situation seule la collaboration permet d'accomplir l'action désirée.
- (ii) Améliorer la productivité de chacun des SRAs.
  1. Augmenter le nombre de tâches réalisées dans un délai imparti ou diminuer le temps de réalisation d'une tâche. La simple multiplication du nombre de SRAs permet d'augmenter quantitativement le nombre de tâches réalisables et de diminuer ainsi le temps d'exécution nécessaire à l'accomplissement d'un ensemble de tâches données. Cependant, ce nombre ne peut augmenter indéfiniment.
  2. Améliorer l'utilisation des ressources. En gérant au mieux l'utilisation de certaines ressources, il est possible de diminuer les coûts de fonctionnement en équilibrant les besoins, ce qui revient alors soit à augmenter la productivité du système soit à améliorer les performances et rendre possible d'autres actions.

Il existe bien évidemment un lien très étroit entre les performances collectives et les capacités de survie individuelles. L'action des SRAs contribue à la performance collective qui en retour assure une meilleure survie de chacun. Cependant, cet avantage impose des contraintes vis-à-vis de chacun des robots en limitant leur marge de manœuvre.

### **I.6.8. Apprentissage**

Trouver les valeurs correctes des paramètres de contrôle qui conduisent à un comportement coopératif désiré peut être une tâche difficile et consommant énormément de

temps pour un concepteur humain. Ainsi, il est très désirable pour un système multi-robots d'être capable d'apprendre pour optimiser la performance de sa tâche et de s'adapter aux changements dans l'environnement. L'Apprentissage par Renforcement (AR) [26, 89] est souvent utilisé dans la robotique coopérative [109]. En effet, Mataric [110] a proposé une reformulation du paradigme de l'AR en utilisant des niveaux d'abstraction plus élevés afin de rendre les robots capables d'apprendre le comportement de fourragement. Elle utilise dans son approche des conditions, des comportements et des fonctions récompenses hétérogènes avec des estimateurs progressifs au lieu des états, actions et renforcement de l'apprentissage par renforcement classique. Parker [124] a utilisé les algorithmes de renforcement standards pour augmenter la performance des robots coopérants dans l'architecture L-ALLIANCE en ayant des agents qui apprennent comment mieux estimer la performance des autres agents.

En plus, des techniques inspirées de l'évolution biologique ont aussi été utilisées dans la robotique coopérative. Werner et Dyer [170] ont utilisé un algorithme génétique [79] afin de développer des contrôleurs neuronaux en simulant des proies qui apprennent un comportement de rassemblement qui aide à éviter les prédateurs.

## I.6.9. Quelques expériences de la robotique collective

### I.6.9.1. Combinaisons de primitives comportementales

Mataric [109] a étudié la façon dont on pourrait exprimer simplement les tâches à réaliser par un groupe de robots. Elle a proposé, dans l'esprit des architectures de subsomption [39], d'exprimer tout un ensemble de comportements collectifs à partir de primitives comportementales qui peuvent être combinées (voir figure I.13). Elle a appliqué son modèle à une tâche de fourragement [57, 59].

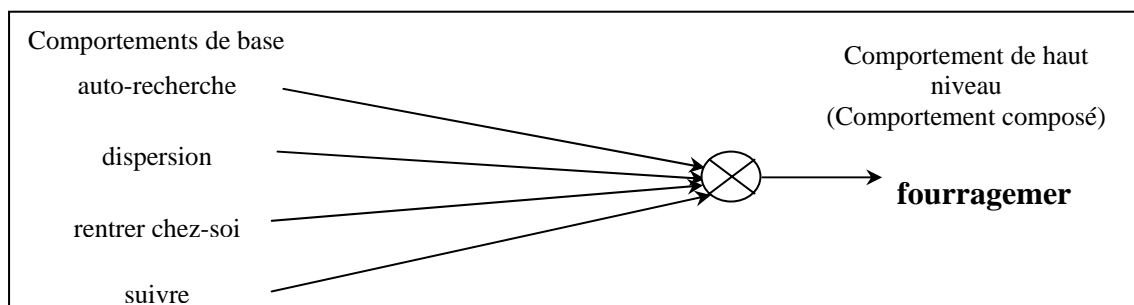


Figure I.13. Exemple d'obtention d'un comportement de plus haut niveau à partir de comportements de base.

Ces comportements de base, astucieusement combinés, peuvent donc permettre la mise en œuvre d'une véritable résolution collective de problèmes [109]. L'idée est que si l'on veut construire une société de robots, il est difficile de savoir a priori quel comportement collectif peut produire un ensemble d'interactions donné. En étudiant un ensemble restreint d'interactions simples, représentées par des comportements primitifs (ou de base), une connaissance précise des comportements collectifs peut être obtenue. Cette approche repose en fait sur l'idée qu'un comportement, pour être exploitable, doit être reproductible autant de fois que l'on désire, robuste et éventuellement réalisable à d'autres échelles, i.e. avec un nombre variable de robots. Ce dernier point peut constituer l'un des problèmes essentiels susceptibles de limiter la performance collective de groupes relativement importants.

### **I.6.9.2. Problèmes géométriques**

Des problèmes géométriques sont inhérents aux systèmes à robots multiples parce que les robots mobiles peuvent se mouvoir dans le monde physique et doivent interagir entre eux physiquement. Différents problèmes géométriques ont été étudiés dans la robotique coopérative et incluent la planification de chemins de plusieurs robots, le déplacement en une formation (et son maintien) et la génération de modèles.

#### **I.6.9.2.1. Planification de chemins de plusieurs robots**

On rappelle que la planification de chemins pour plusieurs robots requiert une planification de chemins qui ne se croisent pas (pas d'intersection). Ce qui correspond en fait à un cas de conflit de ressource puisque les agents et leurs buts occupent un même environnement à un moment donné.

Des détails de la planification de chemins peuvent être trouvés dans [8, 72, 101, 139]. Pour Fujimura [72], la planification de chemins est centralisée (avec un planificateur universel prenant des décisions) ou distribuée (avec des agents individuels planifiant et ajustant leurs chemins). Arai et Ota [8] ont fait la même différence dans la nature du planificateur et présentent aussi des systèmes hybrides qui peuvent être des combinaisons en ligne, hors ligne, centralisés ou décentralisés. Latombe [101] donna une taxonomie assez différente : sa "planification centralisée" est une planification qui tient compte de tous les robots alors que "sa planification découplée" entraîne la planification d'un chemin de chaque robot indépendamment. Pour une planification centralisée, plusieurs méthodes utilisées pour des systèmes à un seul robot peuvent être appliquées. Pour une planification découplée, deux approches sont données :

- i) planification avec priorité qui considère un robot à un instant donné par rapport à une priorité globale, et
- ii) méthode de coordination de chemins qui planifie essentiellement des chemins en ordonnant la ressource dans l'espace-temps des configurations.

#### **I.6.9.2.2. Problèmes de formation du groupe et de formation en marche**

La formation et le déplacement en formation requièrent respectivement plusieurs robots pour former et se déplacer suivant un certain modèle. Solutionner ces problèmes est assez intéressant en termes d'algorithmes distribués [149], d'équilibrage entre une connaissance globale ou locale [125], et d'exigences intrinsèques d'information pour une tâche donnée. Des solutions pour la formation et le déplacement sont aussi des primitives utiles pour des tâches plus complexes, e.g., déplacer un large objet par un groupe de robots [146].

Le vol en formation d'oiseaux ou le comportement des bancs de poissons sont régis par des règles locales très simples. Chaque individu adapte sa direction et sa vitesse en fonction de celle de ses voisins. Dès qu'un individu détecte un obstacle, il l'évite, forçant les autres membres de la formation à réajuster leur position et leur vitesse afin de maintenir les distances au sein de la formation. Ces principes peuvent être transposés au contrôle distribué d'un groupe de robots. Citons quelques exemples d'application de ces principes. Steels [145] et Deneubourg *et al.* [55] utilisent ce mécanisme pour le ramassage d'objets par une colonie de robots. Stilwell et Bay [146] simulent le transport par un groupe de robots d'un objet lourd ne pouvant pas être déplacé par un robot isolé. Dans ce cas, le contrôle distribué est basé sur une stratégie "meneur-suiveurs". Les robots sont supposés connaître leur destination et le robot le

plus proche de cette destination devient le “meneur”. Les autres imitent alors ses mouvements et poussent l’objet dans la même direction.

Le problème de formation paraît difficile, e.g., les travaux existants donnent des algorithmes distribués de “formation-en-cercle” qui forment la figure considérée mais ne garantissent pas que les robots arriveraient à leur but en gardant la figure géométrique formée (un cercle). Cependant, la meilleure solution est celle de Sugihara [148] qui garantit uniquement que les robots vont se distribuer suivant une figure donnée. On remarque que ce problème, bien qu’il paraisse simple, révèle plusieurs pièges dans la formulation des tâches géométriques distribuées. Par exemple, la capacité d’un agent individu de percevoir les attributs de la formation doit être considérée soigneusement: trop d’informations rend le problème trivial mais trop peu d’informations (e.g., retournées par des capteurs localisés) peut empêcher de trouver une solution (e.g., les robots qui ne peuvent se trouver). Le minimum d’informations données (e.g., pour que les robots soient capables de réaliser qu’ils ont accompli la formation demandée) est aussi largement non traité dans la littérature.

## **I.7. Conclusion**

Les systèmes multi-robots comme on l’a vu dans ce chapitre ne sont plus des systèmes renfermés sur leur propre raisonnement qui ignorent leur environnement, mais sont de véritables sociétés d’êtres. Ils doivent se mouvoir, planifier, communiquer, percevoir, agir, réagir et d’une manière générale vivre et travailler dans un milieu dans lequel ils entrent parfois en conflit avec d’autres robots mobiles. Ces systèmes sont alors situés dans leur environnement si l’on veut qu’ils réalisent des tâches réelles et en temps réel. Cette notion de “situatedness” se fonde sur le postulat qu’un système intelligent ne doit pas fonder son comportement sur les raisonnements strictement abstraits effectués dans un système clos. Au contraire, il doit agir et se comporter de façon à assurer sa survie dans son monde environnant. Son comportement est strictement en adaptation continue avec la représentation qu’il se fait de cet environnement considéré comme complexe. L’accent est mis sur une relation organique entre le robot (le système) et son monde. Il s’agit donc d’un tournant vers l’adaptabilité des systèmes.

C’est également la base de ces systèmes multi-robots qui sont particulièrement caractérisés par une très forte dynamique. Comme les ajouts, les suppressions (un robot en panne par exemple) et les modifications d’agents (robots) y sont fréquentes, il devient difficile (voire impossible) de prévoir toutes les situations. L’imprévu est donc inhérent à la vie du système et l’auto-organisation devient un moyen pour parvenir à gérer la dynamique de l’environnement de manière autonome. Cette auto-organisation caractérise alors la capacité d’un système à s’organiser à partir d’une multitude d’interactions entre ses éléments constitutifs. Elle permet donc la production collective de structures complexes. Elle ne résulte pas nécessairement d’une complexité comportementale et cognitive au niveau individuel mais peut spontanément émerger à partir d’interactions entre des agents relativement “simples”. Le but donc de la robotique collective est d’aboutir à des systèmes artificiels adaptatifs, décentralisés, flexibles et robustes inspirés des insectes sociaux et qui soient capables de résoudre des problèmes posés tels que le fourragement, le déplacement en formation, etc. Il est donc nécessaire de comprendre les mécanismes qui produisent les comportements collectifs de manière à arriver à transposer les méthodes générales de résolution distribuée de problèmes mises en œuvre par les colonies d’insectes, moyennant des modifications

appropriées. Pour cela, l'auto-organisation, les architectures de groupe, la coopération et l'apprentissage, entre autres, ont été présentés et continuent d'être étudiées dans le domaine de la robotique afin d'aboutir à des approches qui permettent de solutionner différents problèmes dans le but de réaliser une meilleure prise en charge des problèmes de la vie.

Actuellement, l'approche la plus utilisée est une approche ascendante locale où chaque robot participe à la réorganisation sans avoir une vision globale du système. Chaque agent décide d'appliquer un changement local qui peut pousser les autres agents à réaliser à leur tour des changements locaux et donc engendrer des modifications en cascade, de telle sorte que la réorganisation globale émerge des comportements individuels. C'est le cas dans diverses tâches en particulier les tâches de formations, de fourragement, de transport d'objets lourds et/ou encombrants, etc. Dans toutes ces tâches, des performances collectives apparaissent et se mettent en place exigeant une adaptation précise des réponses individuelles. En effet, la notion de performance collective recouvre non seulement l'idée d'une réponse ajustée mais également et surtout celle d'une réponse adaptative qui nécessite des capacités cognitives au niveau collectif. Ces capacités cognitives collectives qui caractérisent les phénomènes d'intelligence collective, sont mises en œuvre sans régulateur central capable de détecter et de compenser des déviations globales et sans non plus qu'une connaissance suffisante existe au niveau de chaque agent pour que l'on puisse lui attribuer une perception globale de la situation.

Ainsi donc à partir d'agents relativement simples et donc faciles à réaliser, on pourra concevoir des systèmes capables de résoudre des tâches difficiles. De plus le caractère collectif du système, allié à la simplicité de chacun des composants, permettra une plus grande robustesse, une plus grande souplesse, une plus grande simplicité, etc. La NASA par exemple a mis sur pied un programme de colonies de robots explorateurs-mineurs dont le coût de réalisation est bien inférieur à celui d'un seul robot compliqué, sensible aux moindres perturbations, susceptible de tomber en panne et de ne plus faire son travail. Si l'un des robots de la colonie subit une avarie ou un dysfonctionnement quelconque, il reste les autres robots pour la tâche à accomplir et la perte financière est minime. Cette nouvelle vision de la robotique peut être généralisée même dans la vie de tous les jours, et pourquoi pas, où des robots de service pourraient prendre en charge toutes les tâches ménagères de la maison, des robots médicaux plus précis pourraient être utilisés dans les hôpitaux (qui commencent à être utilisés), etc. et construire ainsi un monde mixte.



## Chapitre II

# Navigation des Systèmes Robotiques Autonomes dans un Environnement Dynamique : *Cas de deux robots*

### II.1. Introduction

Avec la demande accrue de précision du contrôle automatique afin de réaliser un travail coopératif par plusieurs SRAs, les approches de contrôle conventionnelles sont incapables de prendre en charge efficacement la complexité, les non-linéarités et l'incertitude du système. Le contrôle intelligent, qui est plutôt basé sur l'expérience que sur le modèle, est conçu comme une discipline émergente afin de surmonter ces problèmes [22, 109, 128]. Ce type de discipline est nécessaire pour le contrôle des robots dans diverses applications robotiques temps-réel [21, 107] en particulier le problème d'évitement de collisions et d'attraction vers un but donné (cible) entre plusieurs robots. Les robots dotés de ce comportement ont la capacité de se déplacer et d'être auto-suffisants dans des environnements multi-robots. Ils doivent être capables de réaliser des comportements orientés tâche ou situation s'ils doivent réagir utilement à leur environnement. Une modélisation à priori de toutes les réactions possibles à des événements particuliers est dans la plupart des cas impossible. C'est pourquoi le développement de systèmes de contrôle pour les robots a conduit systématiquement au développement de systèmes adaptatifs. Ces systèmes réagissent aux changements dans leur environnement, apprennent à partir des erreurs et peuvent résoudre certaines classes de situations inconnues [31, 128]. Ainsi, de tels robots peuvent naviguer et réaliser des comportements semblables à ceux des êtres humains dans des environnements réels. Ceci est devenu une réalité avec l'émergence de la notion d'apprentissage permettant l'émulation des situations humaines ambiguës qui sont souvent rencontrées dans la vie réelle.

La plupart des méthodes de programmation de ces robots utilisaient jusqu'à présent une représentation du monde environnant sous forme symbolique (robots cognitifs). Cependant, avec l'avènement de la robotique collective l'on a vu un nouveau courant s'imposer pour une programmation se basant cette fois directement sur les valeurs fournies par les capteurs (robots réactifs). Cela exige du contrôle de la navigation qu'il prenne en charge les conséquences d'une telle programmation (diversité et bruits dus au capteurs, temps réel, etc.). Il est clair que face à de tels problèmes, il est plus judicieux de pouvoir disposer de la capacité d'apprentissage progressif et incrémental. Les techniques d'apprentissage basées sur les RN, la LF ou encore la théorie de la RA sont parmi les rares voies pouvant déboucher sur cette propriété. Parmi ces approches adaptatives disponibles aujourd'hui, les réseaux de neurones sont parmi les approches les plus traitées [21, 49, 69, 93, 128, 167, 171].

Diverses approches actuelles ont donc été développées et sont basées, entre autre, sur les RN, la LF et la théorie de la RA individuellement [21, 22, 49, 81, 82, 116, 172] ou en combinaison conduisant aux systèmes hybrides intelligents [17, 18, 24, 50, 61, 115, 152].

L'objectif de ce chapitre est d'étudier les comportements des SRAs en groupe par rapport à leur navigation dans des environnements partiellement structurés. L'approche proposée dans ce chapitre se traduit par un système hybride intelligent capable de fournir à deux robots un traitement temps-réel, une meilleure autonomie et de l'intelligence. Elle illustre un dispositif de coordination pour prévenir les risques de collisions entre les robots et entre les robots et les obstacles. Ce mécanisme se présente sous forme de règles comportementales apprises: "les règles prédéterminées", que les robots appliquent en se basant sur l'information concernant les mouvements des robots, fournie par le système sensoriel LOCISS (LOcally Communicable Infrared Sensory System) [5, 6].

Cette approche utilise la théorie de la RA pour apprendre les règles utilisées dans la navigation des SRAs dans un environnement dynamique. Un réseau de neurones artmap flou (RNAMEF) est donc proposé. Cette approche exploite le travail de recherche de Arai *et al.* [6] et suggère une amélioration par apprentissage de l'ensemble de règles définies en utilisant les technologies du SC. En effet, il est plus intéressant d'introduire une manière intelligente basée sur l'apprentissage afin de déduire la règle appropriée à partir des informations (vitesse, identificateur du robot (ID), orientation) délivrées par LOCISS au lieu d'avoir une table comme c'est le cas dans [6]. Chacun des deux robots doit apprendre en utilisant LOCISS un ensemble de règles d'évitement de collisions et d'attraction vers la cible. Ainsi, cette approche doit fournir aux robots, après apprentissage basé sur l'apprentissage supervisé ArtMap Floue Simplifiée (AMFS), la capacité de déterminer la règle permettant au robot d'éviter les collisions.

Dans ce chapitre, une description de la tâche considérée est présentée dans la section II.2 alors qu'un aperçu bref du système LOCISS est décrit dans la section II.3. Un aperçu de différentes approches développées dans la littérature résolvant le problème d'évitement de collisions et d'attraction vers la cible est donné dans la section II.4. L'approche proposée comme système hybride intelligent combinant RN, LF et la théorie de la RA afin d'apporter au SRA un comportement d'évitement de collisions et d'attraction vers la cible proche de celui de l'être humain dans la reconnaissance, l'apprentissage et la prise de décision est développée dans la section II.5.

## II.2. Définition du problème à résoudre

L'objectif de notre travail est d'étudier les comportements collectifs des SRAs et de développer par la suite des approches efficaces permettant de contrôler ces robots individuellement ou collectivement afin de réaliser des tâches complexes en les dotant de comportements adaptatifs et intelligents. Les tâches visées dans ce travail concernent essentiellement les applications terrestres en environnements dynamiques, voire dangereux, i.e., les gares, les ports, les aéroports, les usines et sites dangereux, ainsi que le transport d'objets lourds et/ou encombrants dans ces mêmes environnements.

Ces robots doivent alors apprendre à se comporter dans des environnements dynamiques partiellement structurés et à priori inconnus. Pour ce faire, la notion d'apprentissage est essentielle afin de mener à bien les tâches allouées à ces robots. En effet,

ces robots exploitent des comportements élémentaires pour réaliser un comportement plus complexe de manière à permettre, par apprentissage, la réalisation de la tâche demandée.

## II.2.1. Tâche considérée

### II.2.1.1. Description de la tâche à réaliser

Pour développer un comportement coopératif au sein d'un groupe de SRAs, les robots doivent être capables d'exécuter des règles de contrôle simples telles que:

- Rester au sein du groupe.
- Eviter des obstacles en groupe.
- Se déplacer en groupe.

Certaines tâches nécessitent la mobilisation d'un grand nombre de SRAs. On prend comme exemple le transport d'un objet pouvant avoir plusieurs formes d'un point à un autre dans le milieu industriel. Les robots sont alors amenés à former des figures correspondant à l'objet à transporter et ensuite à se déplacer vers le but désiré.

La figure II.1 illustre le synoptique du système proposé pour l'acquisition de comportements intelligents que les SRAs doivent avoir afin de réaliser la tâche qui leur est assignée.

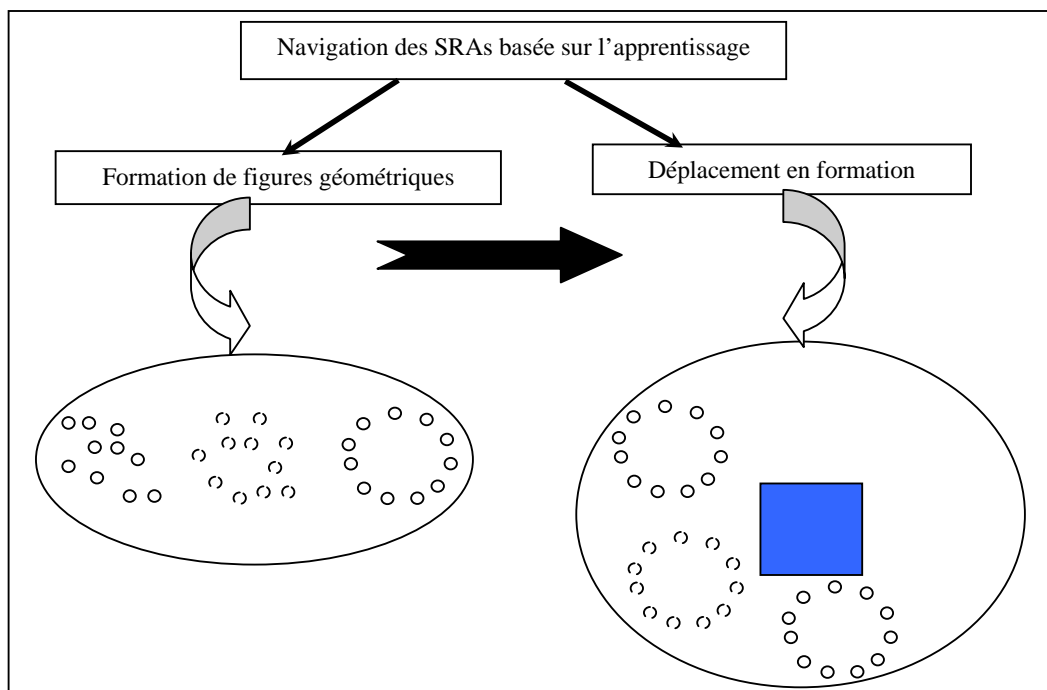


Figure II.1. Synoptique de la tâche à réaliser.

Cette tâche de transport se décompose donc en deux étapes comme montré sur la figure II.1 et qui sont :

- Formation de figures géométriques,
- Déplacement en formation.

Première étape : Formation de figures géométriques

La formation de figures géométriques favorise les performances collectives des robots et offre ainsi un certain nombre d'avantages. Ces derniers s'expriment par des améliorations

de groupe, telle que l'accomplissement des tâches impossibles autrement à réaliser. En effet, dans le chapitre IV, la formation de figures géométriques est abordée.

Deuxième étape : déplacement en formation du groupe de SRAs

Le groupe ainsi formé se déplace dans un environnement dynamique en appliquant d'une part la navigation présentée au chapitre III et d'autre part la technique du "meneur-suiveurs" présentée dans le chapitre IV.

Grâce donc à ces deux étapes, les robots vont pouvoir réaliser la tâche qui leur est assignée en se basant essentiellement sur l'apprentissage.

### II.2.1.2. Etapes de l'apprentissage

La figure II.2 montre la structure globale du contrôleur pour un apprentissage multi-niveaux. Cet apprentissage est constitué de deux niveaux de contrôle modulaires qui correspondent aux deux étapes présentées précédemment. Le processus d'apprentissage est divisé en deux parties, commençant par les problèmes les plus faciles jusqu'aux plus compliqués.

#### 1. Evitement de collision et attraction vers la cible :

- Apprentissage d'évitement d'un seul obstacle par un seul robot.
- Apprentissage d'évitement de plusieurs obstacles par un seul robot.

#### 2. Apprentissage en utilisant un groupe de SRAs :

- Navigation de plusieurs robots dans un environnement dynamique.
- Formation de figures géométriques.
- Déplacement en formation tout en évitant les collisions.

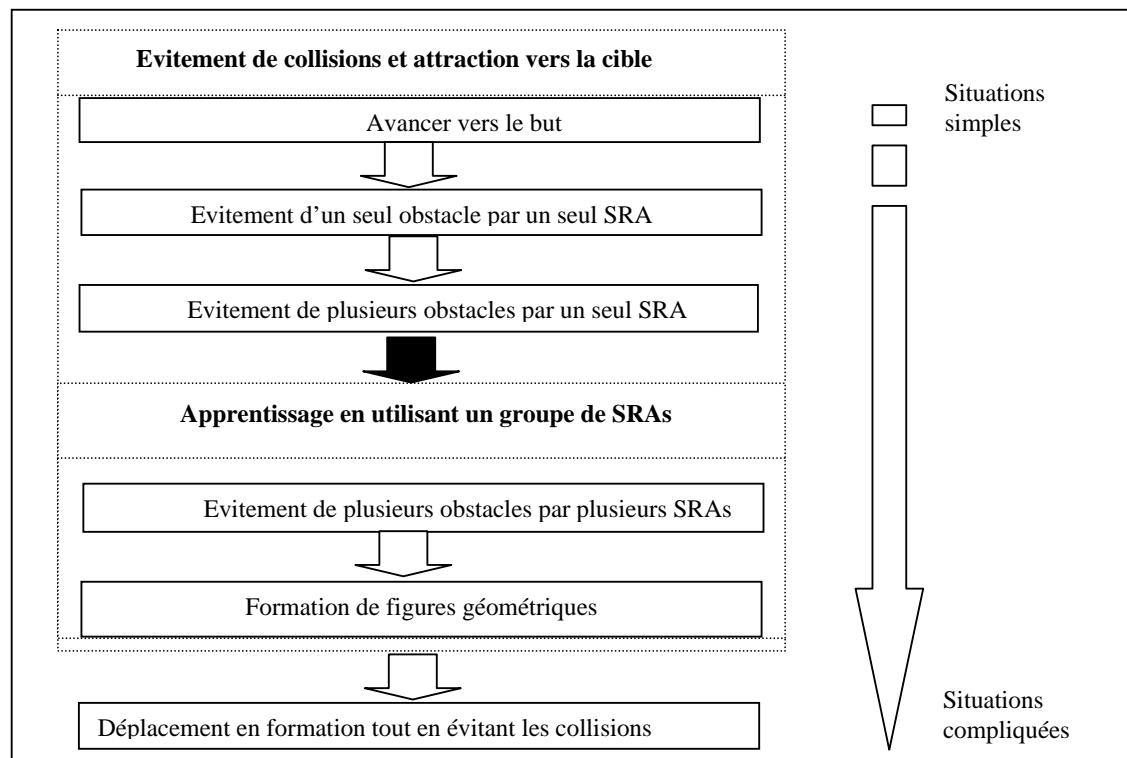


Figure II.2. Phases d'apprentissage utilisées.

## II.2.2. Architecture, mobilité, environnement et traitement

L'architecture, la mobilité, l'environnement et le traitement sont les principaux éléments qui définissent le type de robot envisagé.

### II.2.2.1. Robots visés

On vise essentiellement les robots omnidirectionnels en environnements dynamiques partiellement structurés. Ces robots qui opèrent dans l'espace Cartésien doivent avoir 3 degrés de liberté à savoir, 2 degrés pour le mouvement en translation et 1 degré pour le mouvement en rotation. Souvent, des roues spéciales équipées de rollers sont utilisées pour les robots mobiles omnidirectionnels qui peuvent transmettre la force motrice suivant une certaine direction seulement.

Une grande mobilité des robots est donc requise afin d'étendre leurs applications. Cependant, les robots mobiles conventionnels à roues se heurtent à des contraintes non-holonomes qui réduisent la mobilité et rend la planification de mouvements difficile. Dans le but de réaliser un mouvement coopératif, des mécanismes pour une grande mobilité est requise, qui habilite un contrôle de mouvement libre de chaque robot mobile. Les robots mobiles omnidirectionnels holonomes fournissent une telle mobilité.

Le robot considéré dans notre travail permet alors le contrôle de mouvement pour trois degrés de liberté par trois actionneurs d'une manière découplée. Le robot considéré est donc le robot développé au Laboratoire RIKEN [12]. C'est un robot à quatre roues spécifiques. Une roue est basée sur un arrangement de rollers libres de manière à former une roue [12].

La figure II.3 montre l'arrangement des quatre roues du robot considéré. Le mouvement en translation le long de  $X_r$  est généré par les roues 2 et 4, et celui le long de  $Y_r$  est généré par les roues 1 et 3. Les mouvements en translation suivant n'importe quelle direction peuvent être produits par la composition des mouvements en translation le long de  $X_r$  et  $Y_r$ . Le mouvement en rotation autour de  $Z_r$  est actionné par les quatre roues tournant dans la même direction.

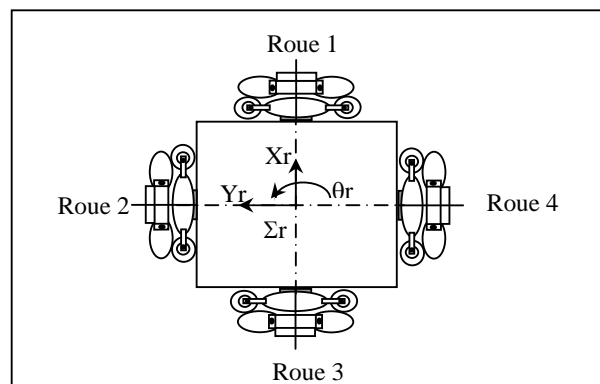


Figure II.3. Arrangement des roues.

Ce robot, montré sur la figure II.4, a comme caractéristiques :  
 Taille : longueur : 410mm, largeur : 410mm, hauteur : 176mm.  
 Poids : 11,6Kg.

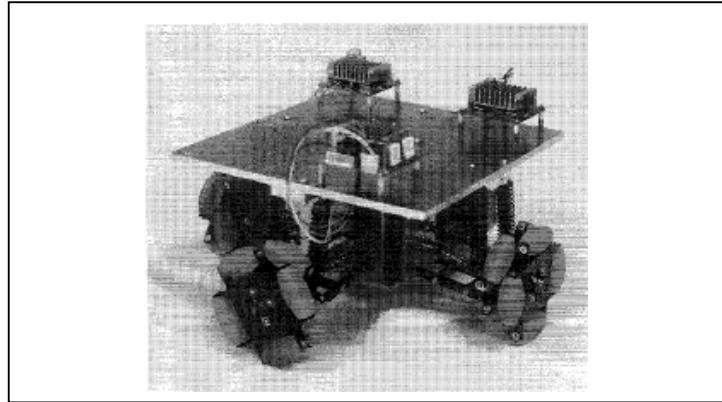


Figure II.4. Robot Mobile OmniDirectionnel (RMOD) considéré.

Les caractéristiques cinématiques du robot ainsi que les équations régissant son déplacement sont données en Annexe A1.

### II.2.2.2. Planification et contrôle de la navigation visés

Les approches actuelles de planification et de contrôle de la navigation ont supprimé la supposition d'une connaissance parfaite de l'environnement. Elles ont également supprimé la représentation explicite de la connaissance pour une représentation implicite basée sur l'acquisition de comportements intelligents qui habilite le robot à interagir efficacement avec son environnement [46]. Ainsi, les SRAs se heurtent à des environnements moins prévisibles et plus complexes, ils ont à s'orienter, à explorer leurs environnements de manière autonome, à s'auto-dépanner et à exécuter des familles complètes de tâches en temps réel. De plus, si les robots manquent d'une connaissance initiale les concernant eux-mêmes et leurs environnements, l'apprentissage devient inévitable. Cet apprentissage se réfère à une variété d'algorithmes caractérisés par leurs capacités de remplacer une connaissance de l'environnement manquante ou incorrecte par l'expérimentation, l'observation et la généralisation. En d'autres termes, l'apprentissage des robots compte sur l'interaction avec l'environnement pour extraire l'information utile afin d'atteindre un but donné [154].

Ainsi, les robots doivent être capables de comprendre la structure de leurs environnements pour réaliser la tâche désirée (tel que, atteindre leurs cibles sans collisions avec d'éventuels obstacles rencontrés, transporter un objet, etc.). Ils doivent donc avoir la capacité de réaliser des comportements intelligents tels que l'évitement d'obstacles et l'attraction vers la cible, la prise de décisions, l'action, le regroupement, la dispersion et la coopération. Ces comportements intelligents font appel à la reconnaissance (traitement de données imprécises issues de capteurs, construction d'une base de connaissance, et établissement d'une carte de l'environnement), l'apprentissage, l'adaptation, le raisonnement et prise de décisions et l'action. En fait, ces derniers constituent les principaux problèmes des SRAs en environnements dynamiques.

Afin de résoudre ces problèmes, les approches classiques [53], ont rapidement été remplacées par les approches actuelles basées sur la collection de méthodologies du SC impliquant les Réseaux de Neurones (RN), la Logique Floue (LF), les Algorithmes Génétiques (AG), la Résonance Adaptative (RA), et l'Apprentissage par Renforcement (AR) utilisés individuellement en Systèmes Intelligents (SI) ou en différentes combinaisons en Systèmes Hybrides Intelligents (SHI) [46, 108, 150, 154, 172]. En fait, d'une part, le but du

SC est de faire évoluer le comportement des SRAs vers celui de l'être humain [3, 22, 69, 79, 91, 93, 97, 126, 150]. D'autre part, ces SI et SHI sont reconnus pour leur amélioration des capacités d'apprentissage, d'adaptation, et de généralisation relatives aux variations dans des environnements où l'information est qualitative, imprécise, incertaine ou incomplète.

Dans ce travail, on vise alors une planification et un contrôle des comportements collectifs des robots qui puissent leur assurer plus d'autonomie et plus d'intelligence avec des capacités de traitement temps réel. Par conséquent, les comportements intelligents de ces robots doivent se baser sur les approches actuelles du SC et donc sur l'apprentissage.

### II.2.2.3. Environnements dynamiques partiellement structurés

Les applications de transport et de surveillance impliquent des environnements dynamiques et généralement partiellement structurés. L'incertitude de la connaissance du monde réel est le principal problème des SRAs [130]. Cette incertitude est due au fait que la représentation des environnements se base essentiellement sur les systèmes de perception des robots qui doit être prise en compte dans la représentation de données.

Pour les stratégies de navigation qu'on propose dans les chapitres II, III et IV, le système de perception utilise un système basé sur des capteurs à InfraRouges (IR) [151] pour la détection d'obstacles et pour permettre la communication entre les robots. En effet, le problème d'une évaluation correcte de données bruitées et incorrectes pour l'interprétation des signaux des capteurs est souvent rencontré. Ce problème est pris en compte dans ces stratégies de navigation par les outils utilisés avec leurs caractéristiques inhérentes d'adaptation et de haute tolérance aux défaillances par rapport aux capteurs défectueux ou aux données bruitées (erronées) des capteurs [49]. Par ailleurs, l'intérêt dans la robotique mobile particulièrement dans les environnements tels que les usines, les gares, les ports et les aéroports a conduit à une description d'éventuels obstacles rencontrés dans de tels environnements. Ces environnements sont dynamiques partiellement structurés avec des obstacles statiques, dynamiques et des SRAs. L'environnement est en fait modélisé par un ensemble de segments de droites représentant les limites de l'espace occupé : le modèle global. Les informations de distance délivrées par les capteurs sont enregistrées dans le modèle local qui représente ponctuellement l'espace libre autour de la position supposée du robot.

En réalité, les obstacles statiques de la figure II.5 de différentes formes représentent des murs, des piliers, des machines, des bureaux, des tables, des chaises, etc., les cercles vides représentent les robots et les cercles pleins leurs cibles respectives.

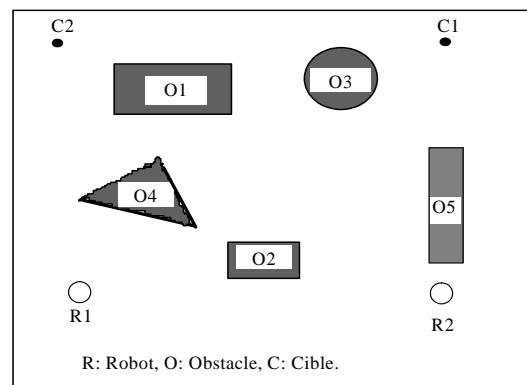


Figure II.5. Exemple d'environnement dynamique partiellement structuré.

### II.3. Navigation par LOCISS (Locally Communicable Infrared Sensory System)

Afin de réaliser un travail de coopération avec des SRAs tels que le transport d'objets ou le fourragement, les robots ont à résoudre en premier le problème de l'évitement de collisions et d'attraction vers la cible. Si un robot en rencontre un autre ou un obstacle statique sur son chemin, il évite d'entrer en collision en stoppant ou en tournant pour libérer le chemin.

Les méthodes de planification et de contrôle de navigation se basaient uniquement sur les distances obtenues par les capteurs séparant les obstacles des robots. Cependant cette information obtenue n'est utile que pour l'évitement de collisions entre robots et obstacles statiques et est pauvre pour un évitement de collisions dans le cas des obstacles dynamiques et particulièrement des robots en mouvement.

Afin d'appréhender le problème de la navigation en groupe de robots, Arai et ses coauteurs [6] ont proposé le système LOCISS. En fournissant une communication locale, ce système permet à chaque robot d'obtenir des informations sur les mouvements des autres robots, en vue d'une planification d'évitement de collisions et d'attraction vers la cible.

#### II.3.1. Description de LOCISS

LOCISS est un système sensoriel qui permet un échange d'informations nécessaires pour l'évitement de collisions sous forme de signaux codés via des émissions d'infrarouge. L'information nécessaire pour l'évitement de collisions est constituée d'un nombre ID identifiant le robot, de sa vitesse et de son orientation. Ce système est construit avec huit (8) paires d'émetteurs/récepteurs montées comme le montre la figure II.6, de sorte que chaque capteur/émetteur (émetteur/récepteur) ait une résolution de  $45^\circ$ . Ce nombre de capteurs dépend des capacités de traitement de l'ordinateur de bord et peut être augmenté pour une meilleure résolution [6].

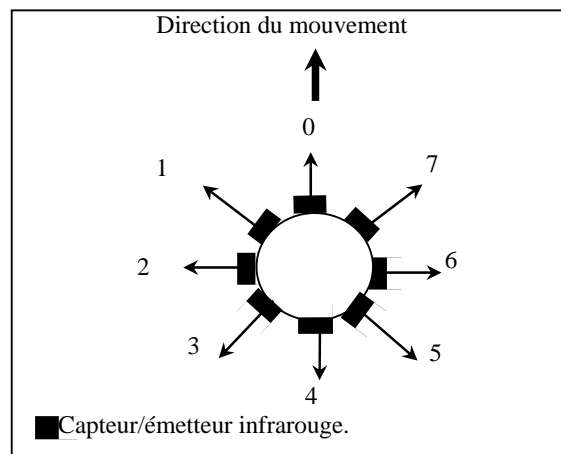


Figure II.6. Système sensoriel du robot.



Un robot doté de ce système peut, pendant son évolution, détecter les obstacles statiques (à une portée de 30 cm) ou communiquer avec les autres robots (à une portée de 80 cm) et ceci dans toutes les directions [7, 151].

Chaque robot a pour tâche d'atteindre sa cible (dont les coordonnées lui sont données au départ).

A tout moment et en accomplissant sa tâche, le robot peut rencontrer sur son chemin soit des obstacles statiques (murs, objets quelconques, robots en état d'arrêt), soit des obstacles dynamiques (robots en mouvement).

### II.3.2 Détection par LOCISS

Dans un système multi-robots, les robots se déplacent vers un but prédéfini dans un environnement où on peut trouver des obstacles statiques et des robots.

Les obstacles stationnaires peuvent être détectés par la réflexion d'un nombre ID identifiant le robot qui émet un signal infrarouge (Figure II.7.a). Par contre, pour détecter un autre robot, son nombre ID est retourné au robot concerné (voir Figure II.7.b). L'échange d'informations est donc nécessaire pour l'évitement de collisions.

### II.3.3. Coopération dans les Systèmes Multi-Robots

Grâce au système sensoriel LOCISS, les robots peuvent évoluer dans un environnement à priori inconnu tout en évitant les collisions entre eux et avec les obstacles. Ces robots s'échangent donc des informations (voir §.I.6.7.1.2) et donc coopèrent afin de réaliser leurs objectifs respectifs. Ils ont alors besoin de coordonner leurs actions (voir §.I.6.7.1.5).

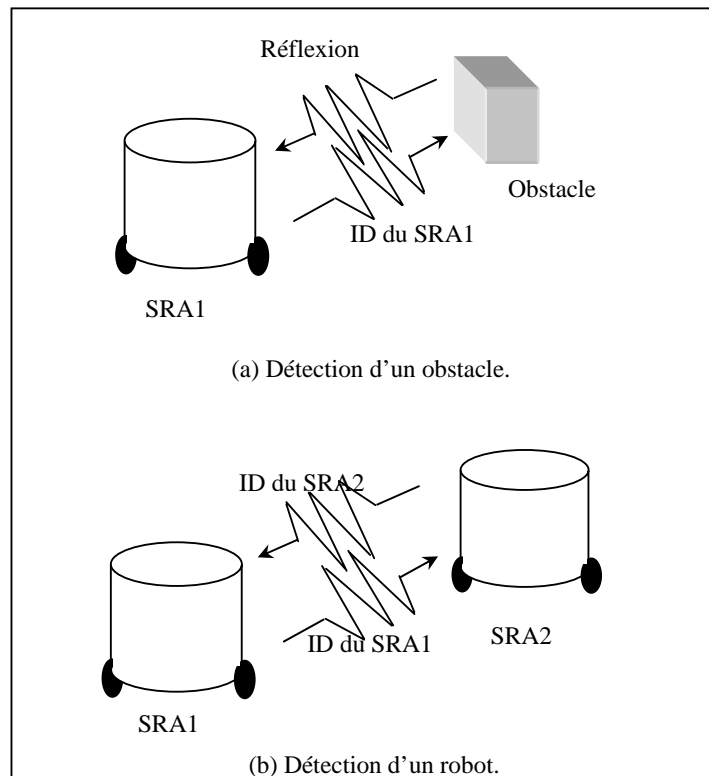


Figure II.7. Détection par LOCISS.

### II.3.3.1. Coordination d'action

Les situations conflictuelles peuvent être résolues par l'intermédiaire de mécanismes de coordination d'actions et de résolution de conflits. Certaines sont particulièrement simples : par exemple, la loi du plus fort est une technique de résolution de conflits, qui passe par la constitution d'antagonismes dans lesquels seuls les plus forts sont capables d'avoir accès aux ressources, les plus faibles en étant privés. Les techniques de négociation par exemple, sont utilisées pour sortir d'une situation de conflit, par des compromis, qui modèrent les satisfactions et les privations des uns et des autres [32].

Dans ce travail, la technique de coordination choisie est basée sur les règles prédéterminées. Elle est utilisée pour anticiper sur les conflits à venir et faire en sorte qu'ils soient gérés par anticipation.

### II.3.3.2. Communication

La communication constitue l'un des moyens fondamentaux pour assurer la coordination des actions dans un environnement multi-robots, du fait de sa possibilité de permettre aux robots d'échanger des informations ou de véhiculer des requêtes. Le dispositif LOCISS dont sont dotés nos robots permet le développement de nouvelles approches de navigation basées sur la communication.

## II.4. Quelques approches de navigation

Dans les approches conventionnelles de planification et de contrôle, le raisonnement sur le comportement du système robotique et sur le comportement global désiré se base sur le formalisme de descriptions mathématiques caractérisé par un haut niveau d'abstraction. Dans le cas où le modèle mathématique du système robotique ou de l'environnement n'est pas disponible ou ne peut être obtenu qu'avec un grand effort et un coût élevé comme c'est le cas avec les robots mobiles, la stratégie de contrôle devrait être basée sur une approche complètement différente. Au lieu d'une description mathématique, une description comportementale du processus est nécessaire se basant à la fois sur des expressions qualitatives et sur l'expérience humaine [95].

Afin de prendre alors en charge tous les problèmes en environnements dynamiques réels, ce qui représente le défi de ces dernières années, on introduit les nouvelles approches de planification et de contrôle, s'inspirant du comportement humain, appelées contrôles intelligents [21, 49, 95]. Cette nouvelle tendance vise à comprendre et à mimer le fonctionnement du cerveau humain. Ces technologies permettent de développer les comportements nécessaires et adéquats afin d'obtenir des robots aptes à évoluer dans des environnements inconnus et même à réaliser des tâches qui deviennent de plus en plus complexes.

Ces approches exploitent les caractéristiques de ces technologies afin de résoudre divers problèmes de la navigation des SRAs. En effet elles utilisent, d'une part, la capacité des RN à se rappeler et à apprendre, plusieurs choses sans nécessairement oublier celles apprises dans le passé et qui représente en fait la principale fonction utilisée dans les RN pour mimer le cerveau humain [3, 93, 168]. Essentiellement, ces RN sont capables d'effectuer des tâches ayant rapport avec la connaissance telles que l'apprentissage, l'adaptation, la généralisation et l'optimisation qui sont intéressantes pour la résolution de problèmes non-linéaires, incertains, incomplets ou non-stationnaires. D'autre part, afin d'appréhender l'environnement du robot, la LF fournit des moyens efficaces pour saisir la nature approximée et inexacte de

l'environnement réel [103, 127, 174, 175]. Ainsi, la LF peut être vue comme une tentative d'évolution des concepts mathématiques conventionnels précis vers ceux de prise de décisions de l'être humain [103, 174, 175]. Les systèmes flous sont également présentés comme des systèmes parallèles à raisonnement associatif, générés à partir de la connaissance d'un expert ou des données d'entraînement [93]. Quant à la théorie de la RA, introduite par S. Grossberg en 1976, c'est une théorie de traitement de l'information des processus par lesquels un être humain acquiert des informations sur son environnement. Elle a depuis mené à un développement de séries de modèles<sup>1</sup> combinant RN et RA pour l'apprentissage de catégories et la reconnaissance de types [43, 69].

Ainsi, diverses approches utilisant les RN, la LF et la RA séparément ou en combinaison ont été développées afin de résoudre le problème de la navigation des SRAs. En effet, la planification basée sur les RN proposée dans [82] a été formulée comme un problème de classification où les types de classes sont cartographiés sur l'ensemble des mouvements manoeuvrables du robot. De plus, pour déterminer un chemin sans danger dans des environnements cartographiés en grille, le traitement distribué et parallèle, en temps réel, d'un modèle basé sur les RN a prouvé son efficacité [104]. L'approche développée dans [24] se basant également sur un système neuronal de contrôle fournit au robot MAVIN (Mobile Adaptive Visual Navigator) la capacité à s'adapter à son environnement et à apprendre à partir d'expériences. Ce système utilise les RN pour la perception, le contrôle de mouvements, l'apprentissage de types (formes) et la reconnaissance d'objets, les associations d'objets, les actions comportementales, etc. Les RN sont également utilisés pour contrôler les comportements d'évitement de collisions et de localisation de la cible dans un environnement dynamique [81]. De plus, le système NEURO-NAV développé [116] pour des véhicules d'intérieur se base sur les RN qui sont entraînés afin d'interpréter les informations visuelles (par l'image) et d'exécuter des comportements de la navigation tels que le suivi de mur (dans un couloir) et la détection de points de repérage (au sol). Une autre approche utilisant un champ de température a été développée par Sorouchyari *et al.* [142]. La tâche du robot consiste à se diriger vers le maximum de ce champ tout en évitant les obstacles. L'environnement consiste en différentes situations spatiales (couloir, virage à gauche, virage à droite, etc.). Le robot mobile dispose de trois actions possibles (aller en avant, tourner à gauche et tourner à droite) et l'association de ces dernières aux situations spatiales se fait grâce à un RN. Le RN habituellement entraîné par un signal, exprimant la réponse désirée, fournie par un processus externe (opérateur humain), est entraîné dans ce travail par un simple signal de renforcement. Il considère son robot comme un système sachant extraire les informations nécessaires qui caractérisent son environnement et découvrir les associations états/actions convenables à l'évitement de l'activation du signal de punition lors des collisions, bien que démuné de toute connaissance au préalable de l'environnement.

<sup>1</sup> RA1: classer des types d'entrées binaires,

RA2: classer des types d'entrées analogiques ou binaires,

et RA3: peut prendre en charge une recherche parallèle ou un test d'hypothèse des codes distribués de reconnaissance en une hiérarchie multi-niveaux de réseau.

Le modèle RA1 a été utilisé afin de concevoir une architecture de réseaux hiérarchiques "ArtMap" qui peut rapidement auto-organiser des correspondances de catégories stables entre des vecteurs d'entrée à  $i$ -dimensions et des vecteurs de sortie à  $n$ -dimensions [48]. La "ArtMap" a été utilisée à l'origine pour apprendre des correspondances entre des vecteurs d'entrée binaires et des vecteurs de sortie binaires. Le modèle de la "Art" Floue développé dans [48] généralise la RA1 à l'apprentissage et reconnaissance stable de catégories en réponse à la fois aux types d'entrée analogiques et binaires. Ce modèle a été incorporé en une architecture "ArtMap" Floue qui est capable d'un apprentissage stable et rapide de reconnaissance de catégories en réponse aux séquences arbitraires de types d'entrée analogiques ou binaires. L'architecture "ArtMap" Floue réalise ainsi une synthèse de la LF et les RN utilisant la RA par l'exploitation d'une proche similarité formelle entre les calculs des sous-ensembles flous et du choix de catégories par la RA, la résonance et l'apprentissage [49].

Des approches combinant les RN, la LF et la RA ont également montré leurs performances quant à la résolution de problèmes liés à la navigation. Ainsi, l'approche développée dans [61] utilise les RNAMF pour une classification d'espaces, à partir de la perception, pour le comportement d'évitement de collisions. Les RNAMF ont également été utilisés dans un contrôleur de planification de mouvements pour le suivi de trajectoires afin de reconnaître les images de caméra [151] et apprendre un positionnement qualitatif d'un robot mobile d'intérieur équipé de capteurs à UltraSons (US) [60]. Une autre approche utilise la ArtMap floue dans un RN pour permettre au robot d'apprendre à éviter les obstacles dans un environnement partiellement structuré [47]. Dans le travail présenté dans [17, 18], divers classificateurs ont été proposés afin de résoudre le problème de l'évitement de collisions et d'attraction vers la cible. Une synthèse sur l'ensemble de ces classificateurs a également été présentée soulignant le fait que le classificateur utilisant les RN, la LF et la RA permet de donner de meilleurs résultats que les autres basés seulement sur les RN ou sur la LF. Dans ces approches, les RNAMF ont été utilisés pour leur capacité de *généralisation*, leur *robustesse*, et leur *apprentissage rapide et stable* [43, 44, 50, 102, 137, 142]. En fait, les RNAMF ont des capacités significatives de généralisation de la connaissance. Par ailleurs, les mesures des capteurs et les images de caméra sont très utilisées pour le comportement d'évitement de collisions et d'attraction vers la cible, mais leurs signaux sont souvent bruités donnant des données incorrectes. Les approches basées sur les RNAMF, avec leurs caractéristiques inhérentes d'*adaptation* et de *grande tolérance aux défaillances*, prennent efficacement en charge ce problème rendant ainsi ces approches effectivement *robustes*.

D'où l'intérêt que nous portons particulièrement aux systèmes hybrides intelligents utilisant les RNAMF [17, 18, 43, 44, 50, 126] par *apprentissage supervisé* par la AMFS [126].

Dans la section suivante, une approche intelligente basée sur les RN utilisant la théorie de la RA est proposée afin de réaliser l'évitement de collisions et l'attraction vers la cible dans des environnements multi-robots. Cette approche permet aux robots d'interagir et d'adapter leurs comportements afin de naviguer intelligemment dans un environnement dynamique.

## **II.5. Approche Neuronale de Navigation des Systèmes Robotiques Autonomes dans un environnement partiellement structuré [19]**

Un RN peut être défini comme un modèle mathématique, s'inspirant du fonctionnement de l'activité cérébrale dont on simule le processus sur ordinateur. Ce sont donc des modèles de calculs avec des propriétés particulières telles que leur capacité à s'adapter ou à apprendre, à généraliser ou à organiser les données dont l'opération est basée sur le traitement parallèle [69, 91, 93].

Ainsi, l'approche proposée utilise les RN afin de réaliser le comportement d'évitement de collisions et d'attraction vers la cible en exploitant les propriétés de ces réseaux de manière à apprendre les bons comportements à effectuer aux bons moments. Elle tire profit également de la LF et de la RA afin de représenter au mieux les données nécessaires à l'apprentissage.

### **II.5.1. Règles prédéterminées**

Dans cette section, les règles prédéterminées introduites par Arai *et al.* [6] sont présentées qui permettent à un robot donné de se référer au mouvement des autres robots

grâce au système sensoriel LOCISS afin de décider de l'action à réaliser. Il faut noter que ces règles ont été introduites pour deux robots seulement.

### II.5.1.1. Evitement d'un robot

Le principe des règles prédéterminées pour l'évitement de collisions et l'attraction vers la cible entre deux robots est simple du fait que deux robots qui se rencontrent ne peuvent se trouver que dans une situation parmi quatre (voir annexe A2).

Les robots disposent alors de quatre comportements de base :

- *Ignorer* : le robot se dirige vers son but ignorant les objets autour de lui.
- *Stopper* : le robot s'arrête pendant un moment.
- *Tourner* : le robot effectue un mouvement à gauche ou à droite par rapport à son orientation.
- *Suivre* : le robot suit son prédécesseur en tenant compte de la vitesse de ce dernier.

Un robot détecte un autre robot dans huit (08) directions lorsqu'il se déplace. Il y a alors soixante quatre (64) combinaisons de la direction du mouvement du robot. Ces combinaisons sont appelées relations positionnelles.

En utilisant les comportements définis ci-dessus, les règles d'évitement de collisions et d'attraction vers la cible sont alors les suivantes pour un obstacle dynamique, i.e., un autre robot [5, 6] :

- R1: les deux robots tournent.
  - R2: un robot ignore et l'autre s'arrête.
  - R3: un robot ignore et l'autre suit le premier.
  - R4: les deux robots s'ignorent mutuellement.
- La règle R1, étant la plus élémentaire, est appliquée lorsque deux robots se trouvent face à face et peuvent éviter la collision en se déportant.
  - La règle R2 est appliquée quand les chemins des deux robots se croisent. Un des deux robots cède le passage à l'autre robot pour ensuite reprendre son chemin vers sa cible.
  - La règle R3 est appliquée dans le cas où un robot se trouve derrière un autre et qu'ils vont approximativement dans la même direction. Le premier robot ignore le suivant pendant que ce dernier ajuste sa vitesse pour ne pas le rattraper et engendrer une collision.
  - La règle R4 est appliquée lorsqu'il n'y a pas besoin d'évitement.

En appliquant ces règles pour les soixante quatre situations de relations de position, une matrice de règles, montrée dans le tableau II.1, est obtenue.

Tableau II.1. Matrice de règles pour un robot.

		Code de la direction reçue							
		0	1	2	3	4	5	6	7
Code du capteur recevant les données	0	R1	R2	R2	R3	R3	R3	R2	R2
	1	R2	R1	R4	R4	R4	R3	R2	R2
	2	R2	R4	R4	R4	R4	R4	R4	R2
	3	R3	R4	R4	R4	R4	R4	R4	R3
	4	R3	R4	R4	R4	R4	R4	R4	R4
	5	R3	R3	R4	R4	R4	R4	R4	R4
	6	R2	R2	R4	R4	R4	R4	R4	R4
	7	R2	R2	R2	R3	R4	R4	R4	R1

La Figure II.8 donne un exemple de l'utilisation de la table de règles (Tableau II.1).

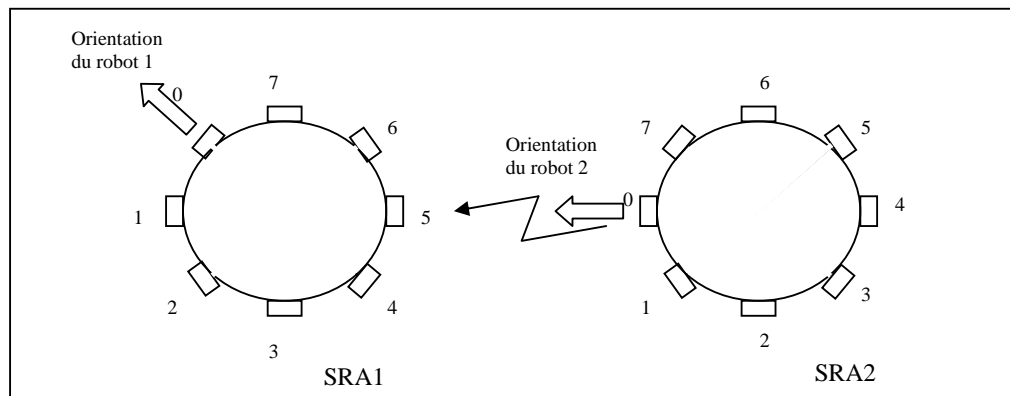


Figure II.8. Exemple de détection et d'échange d'information.

La case du tableau II.1 correspondant à l'intersection de la colonne n°0 et de la ligne n°5 représente la règle R3. En effet, pour qu'il n'y ait pas de collision, le robot SRA2 doit réduire sa vitesse alors que le robot SRA1 doit l'ignorer et poursuivre son chemin.

### II.5.1.2. Evitement d'un objet statique

Pour un obstacle statique, le comportement du robot est simple puisqu'il tourne quand il y a un obstacle à éviter ou alors il ignore l'obstacle quand il n'y a pas de collision. Ce cas est donc pris en considération par le système de navigation.

De même, une matrice (Tableau II.2) de règles pour l'évitement de collisions et l'attraction vers la cible avec les obstacles statiques est construite en ne tenant compte cette fois que de deux (02) situations.

Si un obstacle ou un robot en état d'arrêt est détecté par l'un des capteurs du robot, ce dernier applique la règle R5, sinon c'est la règle R6 qu'il appliquera puisqu'il n'y a pas de risque de collision et donc pas de nécessité d'évitement.

- R5: le robot tourne pour éviter l'obstacle.
- R6: le robot ignore l'obstacle et continue son mouvement.

Tableau II.2. Matrice de règles entre un robot et un objet.

	0	1	2	3	4	5	6	7
R5	R5	R5	R6	R6	R6	R6	R6	R5

← Numéro du capteur recevant le signal

Règle à appliquer

### II.5.2. Approche proposée d'évitement de collisions et d'attraction vers la cible basée sur un Réseau de Neurones ArtMap Flou (RNAMF)

Dans cette section, le problème d'évitement de collisions et d'attraction vers la cible est résolu en utilisant l'approche neuro-flou proposée comme Système Hybride Intelligent (SHI) reconnaissant différentes relations de position dans le but d'éviter les obstacles dynamiques dans des environnements partiellement structurés [17]. En effet, les SHI ont été reconnus pour l'amélioration des capacités d'apprentissage et d'adaptation relatives aux variations dans des environnements où l'information est qualitative, imprécise, incertaine ou incomplète [115]. En particulier, l'utilisation de SHI combinant RN, LF et la théorie de la RA [93, 115, 142] est nécessaire pour fournir aux robots plus d'autonomie et d'intelligence avec des capacités de traitement temps-réel.

#### II.5.2.1. Architecture de contrôle utilisée

L'architecture de contrôle utilisée est une architecture comportementale où plusieurs comportements sont utilisés comme montré sur la figure II.9. La prise de décision dans le choix de l'un ou de l'autre de ces comportements est réalisée via le réseau RNAMF développé.

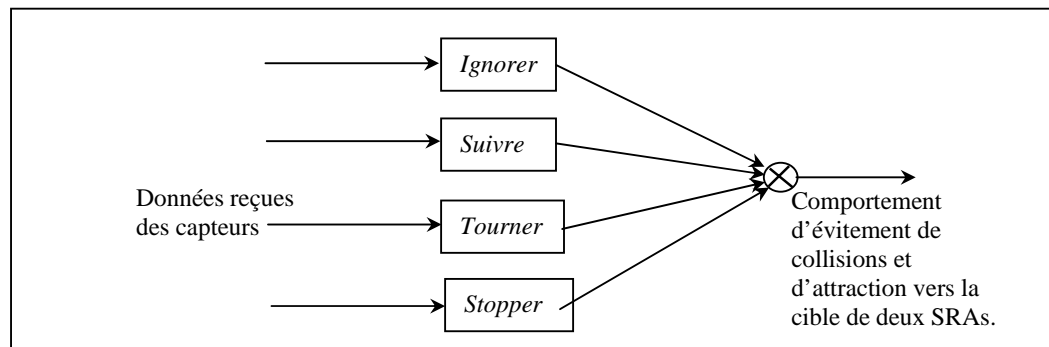


Figure II.9. Architecture adoptée pour le comportement d'évitement de collisions et d'attraction vers la cible.

Ces comportements permettent alors aux robots de s'éviter et d'atteindre leurs cibles avec succès grâce à l'apprentissage réalisé.

#### II.5.2.2. Description du système

Les règles prédéfinies sont apprises par le robot qui acquiert la capacité d'éviter les collisions avec un obstacle dynamique intelligemment. Ce comportement d'évitement de collisions et d'attraction vers la cible est basé sur une reconnaissance par RNAMF acquise par

l'apprentissage à partir du système LOCISS du robot comme montré sur la figure II.10. Ainsi, l'approche proposée doit rendre le robot capable, après apprentissage, d'éviter un obstacle dynamique (un autre robot) par RNAMF, ou un obstacle statique par un simple module de traitement.

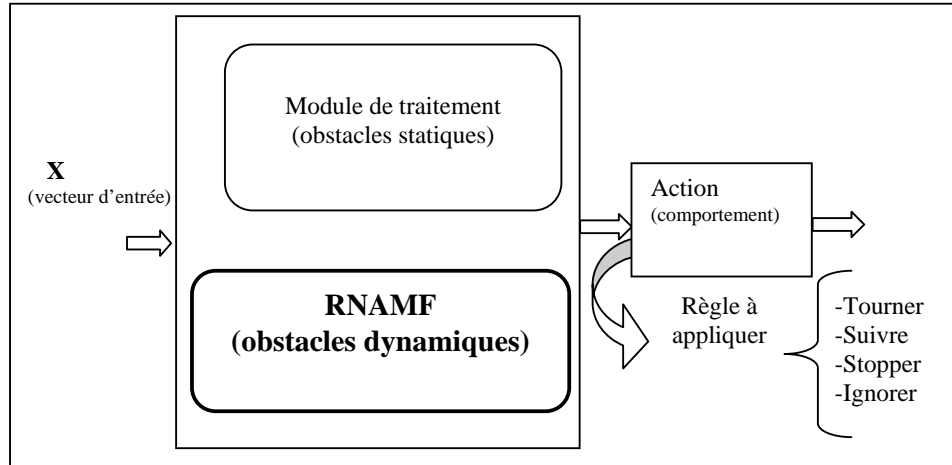


Figure II.10. Synoptique du système proposé.

Huit (08) mouvements ( $M_i, i=1, \dots, 8$ ) dans huit (08) directions ( $0^\circ, 45^\circ, 90^\circ, 135^\circ, 180^\circ, 225^\circ, 270^\circ, \text{ et } 315^\circ$ ) correspondent aux huit (08) capteurs dotant chaque robot comme montré sur la figure II.11.

Le code du capteur est remplacé par l'angle correspondant. En d'autres termes, le vecteur d'entrée de RNAMF est constitué de deux angles relatifs entre les directions des mouvements des robots,  $X = [\theta_1, \theta_2]^T$  (voir figure II.11).

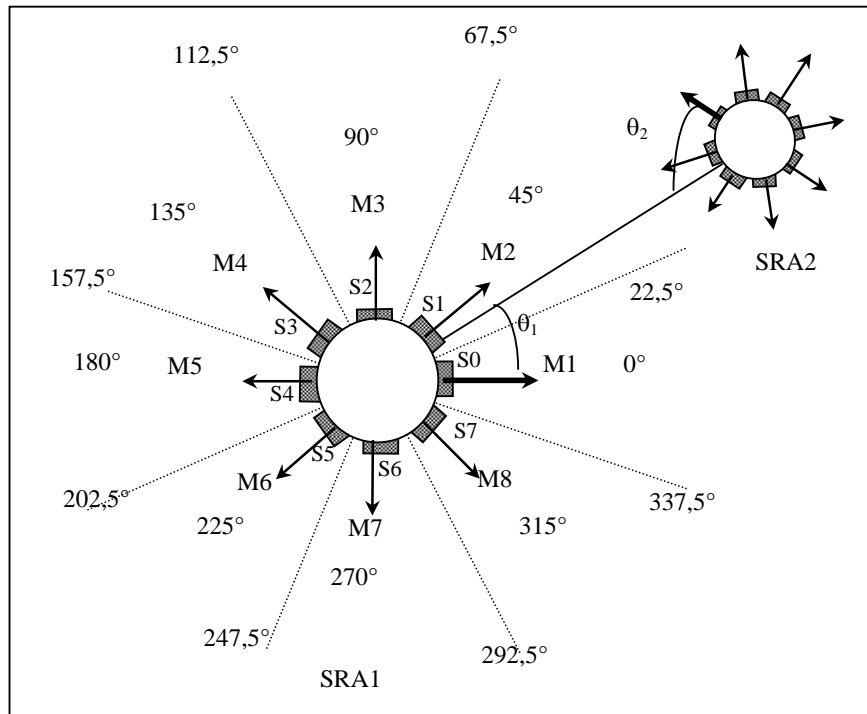


Figure II.11. Modèle du robot mobile dans RNAMF.



Les règles prédéfinies sont alors apprises par le RNAMF qui est un réseau de neurones artmap floue constitué d'une couche ligne d'entrée (codage à complément), une couche d'entrée, une couche de sortie de catégorie et une couche catégorie comme montré, après apprentissage, sur la figure II.12.

### II.5.2.3. Apprentissage par l'algorithme AMFS (ArtMap Floue Simplifiée)

Afin d'acquérir le comportement d'évitement de collisions et d'attraction vers la cible, le classificateur RNAMF est entraîné en utilisant le paradigme ArtMap Floue Simplifiée AMFS détaillé dans [92]. AMFS est spécialisé dans la classification de patterns et peut apprendre chaque pattern en seulement quelques itérations d'entraînement. Il démarre sans poids de connexion mais se développe en taille pour englober le problème et contient uniquement un seul paramètre à ajuster : la vigilance de base. Cet apprentissage incrémental a prouvé qu'il est stable parce que tous les poids adaptatifs peuvent seulement décroître avec le temps [43]. De cette manière, le réseau apprend par expérience, i.e., à partir d'exemples. Dans l'approche proposée, les règles qui ont été au préalable prédéfinies constituent les exemples à apprendre.

Durant la navigation, chaque SRA doit construire une carte interne implicite (i.e., obstacles et espaces libres) à partir des données des capteurs, la met à jour et l'utilise afin de contrôler intelligemment son comportement d'évitement de collisions et d'attraction vers la cible. Ce comportement est acquis par apprentissage à partir des informations délivrées par LOCISS.

Les composantes du vecteur d'entrée  $X$  constitué des angles  $\theta_1$  et  $\theta_2$  sont normalisées pour se trouver dans l'intervalle  $[0, 1]$  dans l'équation (II.1) :

$$\theta = \theta / 360 \quad (\text{II.1})$$

*Couche d'entrée ligne:* constituée de deux entrées correspondant aux composantes du vecteur d'entrée  $X$ .

$$X = \begin{pmatrix} \theta_1 \\ \theta_2 \end{pmatrix} \quad (\text{II.2})$$

*Couche d'entrée:* avec quatre nœuds d'entrées, constituant le vecteur d'entrée  $I$  dans l'équation II.3, obtenue des deux entrées complétées à partir du codeur à complément. Ce codeur prend un vecteur d'entrée de dimension  $i$  et produit  $2i$  sorties où les  $i$  valeurs de sortie additionnelles sont les compléments des  $i$  valeurs d'origine. Les composantes d'entrée ajoutées sont fournies pour faciliter la formation de régions de décision par le réseau. Cette couche transmet ses entrées à tous les nœuds de la couche suivante.

$$I = [X_1, X_2, \bar{X}_1, \bar{X}_2] \quad (\text{II.3})$$

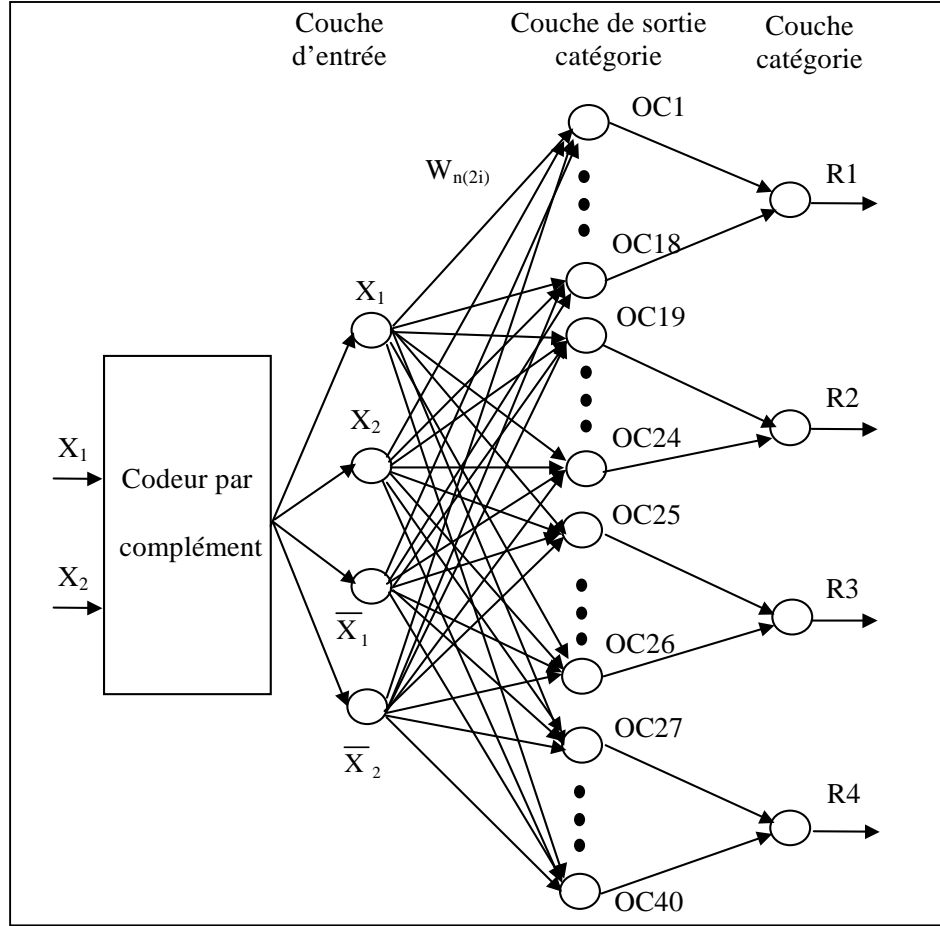


Figure II.12. Architecture de RNAMF.

*Couche de sortie catégorie:* démarre sans nœuds de sortie de catégorie (pas de poids) et augmente en taille en incorporant un exemple dans une catégorie de sortie existante ou en créant un nouveau nœud de sortie de catégorie. Ceci afin d'atteindre finalement  $N$  nœuds de sortie de catégorie représentant les quatre règles utilisées dans l'évitement de collisions avec des obstacles dynamiques.

Pour chaque entrée  $I$ , la fonction de sortie d'activation (fonction choix),  $OC_n$ , est définie par :

$$OC_n(I) = \frac{|I_{(2i)} \wedge W_{n(2i)}|}{\lambda + |W_{n(2i)}|}, \quad (II.4)$$

où  $\lambda$  est une petite constante positive  $0 < \lambda \ll 1$ ,

$W_{n(2i)}$  sont les poids des connexions,

l'opérateur AND flou  $\wedge$  est défini par [174]:

$$(p \wedge q)_m \equiv \text{Min}(p_m, q_m), \quad (II.5)$$

avec  $m = 1, \dots, M$  et la norme  $|\cdot|$  est définie par :

$$|p| \equiv \sum_{m=1}^M |p_m|, \quad (II.6)$$

pour n'importe quel vecteur  $p$  et  $q$  de dimension  $m$  [43, 44, 92].

La catégorie choisie est alors indexée par  $N$ , où :

$$OC_N = \text{Max} \{ OC_n, n = 1, \dots, N \}. \quad (\text{II.7})$$

*Couche catégorie*: avec quatre nœuds catégorie, le maximum de catégories que le réseau peut apprendre, chacune est libellée comme une catégorie unique ou une classe.

#### II.5.2.4. Entraînement de RNAMF

Le classificateur RNAMF est entraîné à partir de deux cents quatre vingt dix-sept (297) exemples de l'ensemble d'entraînement en incorporant un exemple dans un nœud catégorie de sortie déjà existant ou en créant un nouveau nœud catégorie pour l'exemple en question.

Ce RNAMF décide si un ou plusieurs nœuds catégories de sortie sont requis pour représenter une catégorie particulière. Quand la vigilance augmente, ce classificateur crée plus de nœuds catégories de sortie pour les mêmes exemples d'entrée, avec des limites plus fines de catégorie. En effet, le réseau se développe pour représenter le problème comme il le voit au lieu de se faire dire par le concepteur du réseau de fonctionner sous une architecture statique.

Chaque règle est donc représentée par diverses sous-classes (catégories de sortie) comme montré sur la figure II.12. En effet,

- Dix-huit (18) sous-classes correspondent à la règle R1.
- Six (6) sous-classes correspondent à la règle R2.
- Deux (2) sous-classes correspondent à la règle R3.
- Quatorze (14) sous-classes correspondent à la règle R4.

En fait dans AMFS, l'assignation des nœuds de sorties aux classes est laissée au réseau [92]. Ce RNAMF a la capacité de faire face aux environnements changeants non-stationnaires. En supposant une certaine association comme fautive est non définitive due à la loi d'apprentissage non-monotone, elle peut devenir favorable de nouveau. Dans ce cas, le système auto-organise ses réactions aux stimuli reçus. L'apprentissage après une perturbation résulte en une augmentation du nombre de catégories perçues, qui incorpore les capacités du système à traiter les non-stationnarités de l'environnement, ou de sa propre structure. Dans les deux cas, une importance de la caractéristique de LF-RA de préservation de la connaissance déjà stockée tout en incorporant de nouvelles données, peut facilement être tirée [61].

## II.6. Conclusion

L'approche proposée dans ce chapitre pour résoudre le problème de l'évitement de collisions et d'attraction vers la cible se traduit par un SHI exploitant les capacités des RN, la LF et la RA. Ce RNAMF se base sur le paradigme AMFS qui permet d'apprendre diverses règles à partir des informations sur les situations de position pour l'évitement de collisions. En effet, ce RNAMF est utilisé pour sa capacité de généralisation, de robustesse, d'apprentissage AMFS stable et rapide. En fait, les réseaux flous à base de la théorie de la RA ont de grandes capacités de généralisation de la connaissance. D'un autre côté, les capteurs et les images de caméra sont très utilisés pour l'évitement de collisions, mais leurs signaux sont souvent

bruités donnant alors des données incorrectes. Les approches basées sur les RN utilisant LF-RA avec leurs caractéristiques inhérentes d'adaptabilité et de tolérance aux pannes et aux bruits, prennent en charge ce problème rendant ce type d'approches robustes.

Cette approche combine les avantages complémentaires des RN, de la LF et de la théorie de la RA afin d'apprendre plus rapidement (en un seul cycle) les règles produisant les comportements à appliquer au robot afin d'éviter les collisions. Elle opère ainsi rapidement avec moins de paramètres à ajuster sous l'hypothèse qu'elle peut apprendre de nouvelles situations. En d'autres termes, pour une nouvelle classe, le RNAMF doit être entraîné seulement avec les nouveaux exemples d'entraînement ce qui n'est pas le cas avec les RN classiques. Grâce au réseau RNAMF, on peut garantir un évitement de collisions et d'attraction vers la cible entre deux robots ou entre un robot et un ou plusieurs obstacles. Ce réseau a permis de remplacer la table des différentes situations possibles auxquelles les robots doivent faire face améliorant ainsi leur capacité à répondre à priori à des entrées non apprises au préalable. Ce qui permet au robot d'avoir des capacités plus intéressantes face à des situations imprévisibles. Cependant, cette approche n'est applicable que pour deux robots au plus. Quand le nombre de robots augmente les situations de l'environnement deviennent de plus en plus complexes et trouver la base d'entraînement devient très vite impossible. Une solution au problème est d'introduire un raisonnement qui n'exige pas cette base d'entraînement.

En effet, dans ce cas simple on peut facilement définir l'ensemble des règles qui déterminent les types de comportements à adopter suivant les signaux détectés par le système LOCISS. Mais, il est cependant difficile de généraliser cette approche basée sur les règles aux situations où le nombre de robots dépasse trois (le nombre de combinaisons devenant énorme). C'est pour cette raison qu'il est préférable pour nos robots d'adopter une approche de navigation qui leur permette de s'adapter et d'apprendre progressivement des situations les plus simples aux situations les plus difficiles. Ce qui fait l'objet du chapitre suivant.

## Chapitre III

# Navigation des Systèmes Robotiques Autonomes dans un Environnement Dynamique : *Vers une navigation en groupe*

### III.1. Introduction

Pour réaliser des applications robotiques, les robots doivent être dotés d'une approche efficace de navigation leur permettant d'évoluer dans leurs environnements. Ils doivent accomplir leur tâche malgré les incertitudes qu'ils ont sur leur environnement, les actions de chaque robot influent sur la prochaine position donc sur les prochaines actions qu'il pourra effectuer.

Certaines classes de problèmes sont résolues par l'utilisation des approches globales, mais dans de nombreux cas, ces approches sont incapables de répondre à tous les problèmes, comme par exemple le problème de collisions, et cela est dû essentiellement à la présence d'au moins une des trois difficultés suivantes :

- i) Le nombre de situations possibles considérées est calculable, mais trop grand pour préparer toutes les réponses correspondantes.
- ii) La situation est imprévisible parce qu'un processus aléatoire est essentiellement impliqué dans le problème.
- iii) La réponse au problème n'est pas connue au préalable.

Une approche est proposée dans ce chapitre afin de remédier à ces problèmes et particulièrement le premier point. En effet, lorsque le nombre de situations augmente, il est plus difficile voire impossible de définir les règles appropriées afin de résoudre le problème de navigation. Il est donc nécessaire d'introduire d'autres notions, particulièrement la notion de renforcement, qui permettrait d'obtenir une navigation autonome. On propose alors une approche de navigation basée sur l'interaction qui est applicable à un important nombre de robots. En effet, cette approche est une approche d'acquisition de comportement d'évitement de collisions et d'attraction vers la cible pour plusieurs robots capable de fournir à ces robots plus d'autonomie et d'intelligence. Ces robots devraient être capables d'acquérir la connaissance eux-mêmes en utilisant les capteurs et les actionneurs (la perception et l'action) [128]. Cette approche utilise une méthode adaptative afin d'acquérir des comportements plus complexes à partir de comportements élémentaires pour éviter les collisions avec d'autres robots et obstacles. En effet, tel qu'un bébé doit apprendre comment contrôler ses mouvements pour marcher ou manger avant d'apprendre n'importe quelle stratégie complexe, les robots doivent aussi acquérir des aptitudes de bas niveau avant d'exhiber des comportements complexes. La meilleure compréhension de comment agir comme une partie

intégrante d'un groupe est inutile sans la capacité d'exécuter les comportements élémentaires nécessaires. Ainsi, afin d'acquérir le comportement adaptatif, l'apprentissage par renforcement est introduit. Il est montré que les comportements appropriés pour l'évitement de collisions peuvent être acquis avec succès à travers le processus d'apprentissage proposé. Cet apprentissage se rapporte en fait à un contrôle adaptatif du robot à travers l'utilisation de récompenses (reward) scalaires (comme retour) et une interaction directe par essai et erreur avec l'environnement [150, 167]. En effet, le système basé sur l'AR améliore sa performance en recevant un retour sous forme d'une récompense scalaire (ou punition) qui est fonction de la réponse appropriée. Ainsi, cette approche comportementale doit fournir aux SRAs la capacité d'éviter les collisions à partir de leurs interactions avec l'environnement.

Elle utilise l'interaction des robots avec l'environnement pour dégager la politique à adopter afin de réussir la réalisation de leurs tâches. Le seul souci de programmation de cette approche basée sur l'AR est de trouver une bonne fonction de renforcement qui exprime au mieux la tâche, ainsi qu'une représentation de situation reflétant au mieux l'environnement. Cette approche d'apprentissage est efficace pour la résolution des problèmes compliqués quand l'intervention humaine est incapable de définir les règles appropriées ou les lois de contrôle du système [70].

En effet, l'approche proposée est basée sur une coopération simple entre les robots puisqu'ils sont homogènes, se communiquent les informations instantanées (comme leurs positions) et mettent à jour la même politique de décision inspirée par le travail de Tan [153]. Ils bénéficient du partage de la politique apprise parce qu'ils font face au même problème. La politique dans ce travail est représentée par la table de valeurs Q où la connaissance commune est apprise par tous les robots. C'est ainsi qu'ils exploitent et renforcent la même politique. En d'autres termes, chaque agent utilise une connaissance indirecte acquise par l'un des autres agents et un apprentissage rapide est réalisé.

Dans ce chapitre, quelques approches basées sur l'AR sont présentées dans la section III.2. Une approche essentiellement basée sur l'interaction du robot avec son environnement est proposée dans la section III.3. En effet, l'approche utilise des comportements élémentaires afin d'apprendre comment se conduire dans un environnement multi-robots. Cette approche comportementale favorise des actions réflexes simples et associe l'apprentissage entre la perception et l'action. Le robot améliore son comportement comme résultat de son interaction avec son environnement où les situations sont perçues et une action est générée en utilisant une fonction liant les situations aux actions.

### **III.2. Quelques approches basées sur l'AR**

Concevoir des robots qui apprennent par eux-mêmes à réaliser des tâches complexes du monde est toujours un défi ouvert dans le domaine de la robotique et de l'intelligence artificielle [109, 128, 171]. En effet, plusieurs chercheurs ont développé des approches basées sur l'apprentissage par interaction avec l'environnement et par expérience. Ce type d'apprentissage est pertinent dans le contrôle intelligent puisqu'il conduit à des systèmes qui ne dépendent pas à priori de la connaissance dans le processus de prise de décision. L'apprentissage par renforcement maintient une fonction politique explicite qui relie directement des situations à des actions. Il n'utilise pas un modèle explicite du domaine afin de générer une séquence d'actions qui est ensuite exécutée en boucle ouverte comme c'est le cas avec la planification classique [150, 154, 171]. En effet, les approches basées sur l'AR

permettent une réponse rapide à des éventualités et des opportunités imprévues conduisant à des systèmes situés et réactifs qui sont très intéressants en robotique et plus particulièrement dans les systèmes multi-robots. “Situatèdness” et la réactivité sont en fait deux propriétés importantes des systèmes utilisant l’AR. En effet, les robots sont situés puisqu’ils doivent contrôler eux-mêmes toute interaction avec l’environnement, i.e., le monde doit toujours être vu à partir des robots [129]. En d’autres termes, les robots sont situés si leur décision (contrôle) est basée sur la situation actuelle (déterminée par les capteurs et également une connaissance limitée de l’état interne si possible). Par conséquent, les robots doivent être capables d’apporter eux-mêmes de l’expérience au cours du traitement de la situation actuelle [129, 171]. Par contre, les robots sont réactifs s’ils génèrent des actions (comportements) en fonction de la dynamique de l’environnement où ils évoluent. Pour de tels robots, la prise de décision consiste à évaluer une fonction politique qui typiquement requiert un temps minimal [171]. Ces robots caractérisés également par leur apprentissage incrémental adaptent leurs politiques en se basant sur l’expérience accumulée au cours du temps. Ils atteignent graduellement la bonne réponse correcte à travers des approximations successives même si leurs modèles sont incomplets ou imprécis conduisant ainsi à des robots robustes. Cette robustesse implique également que ces modèles peuvent être appris durant le processus de l’AR, qui permet ainsi leur utilisation, par les robots eux-mêmes [150, 171].

L’utilisation de l’AR dans le problème d’évitement de collisions et d’attraction vers la cible pour plusieurs robots est toujours un problème ouvert. Par exemple, une acquisition adaptative pour l’évitement de collisions et d’attraction vers la cible entre plusieurs robots mobiles qui sont équipés du système de perception LOCISS a été développée [4]. Cette approche est basée sur un schéma d’apprentissage par renforcement où le comportement sélectionné est exécuté et est évalué en se basant sur trois distances. Le processus d’apprentissage doit être exécuté afin d’acquérir le comportement d’évitement de collisions et d’attraction vers la cible pour une situation spécifique. Cependant, l’implémentation de cette approche sur des robots réels constituait un problème. Le nombre de situations possibles devient extrêmement important et la capacité des mémoires qui peuvent être montées sur les robots actuels ne peuvent les supporter en raison de l’explosion combinatoire des données d’entrées échangées par LOCISS entre les différents robots. Afin de réduire ce nombre de combinaisons et réaliser un mécanisme de contrôle pouvant être installé sur l’ordinateur de bord du robot, un schéma de contrôle basé sur l’AR multi-couches approprié a été proposé par Fijii *et al.* [70]. Il est constitué de quatre couches de contrôleurs modulaires correspondant aux étapes de l’AR. Le processus d’apprentissage est divisé en quatre étapes tel un programme qui commence par les problèmes les plus simples et traite ensuite les problèmes les plus compliqués. Un autre algorithme d’évitement de collisions et d’attraction vers la cible pour deux robots, proposé dans [71] a été généré par un processus d’apprentissage très simple qui simule le processus d’apprentissage naïf par essai-et-erreur. Cette approche a utilisé seulement les sorties des capteurs du robot et une fonction récompense appropriée où la forme exacte de la fonction récompense a été apprise par les robots d’une manière autonome. Les auteurs ont aussi discuté comment le robot peut utiliser son “expérience” acquise d’un environnement simple afin d’ajuster son comportement à des environnements plus complexes en générant automatiquement un algorithme d’évitement de collisions et d’attraction vers la cible pour une situation à trois robots en utilisant un espace d’état réduit correspondant à deux robots.

Globalement, ces approches d’AR sont basées sur la manière avec laquelle l’être

humain raisonne, établit ses propres décisions et choisit l'action appropriée. En effet, ces approches partagent des caractéristiques qui sont si élémentaires qu'elles sont faciles à contrôler. Toutes impliquent une interaction entre un agent de prise de décision actif et son environnement, dans lequel le robot cherche à atteindre un but malgré l'incertitude sur celui-ci. Les actions de l'agent sont autorisées à affecter l'état futur de l'environnement (e.g., la localisation future du robot), affectant de cette façon les options et les opportunités disponibles de l'agent dans un futur proche. Ainsi, afin de survivre, ils doivent être capables de prédire les événements futurs en généralisant les conséquences de leurs réponses comportementales à des situations similaires expérimentées dans le passé. Ces agents (robots autonomes) n'ont donc pas besoin des détails de l'environnement sur lequel ils vont agir : ils les acquièrent par interaction directe.

Dans la section suivante, une approche intelligente basée sur l'AR est proposée afin de réaliser l'évitement de collisions et l'attraction vers la cible dans des environnements multi-robots en intégrant un mécanisme adaptatif d'acquisition des comportements appropriés dans chaque robot. Cette approche permet aux robots d'interagir et d'adapter leurs comportements afin de réaliser la tâche désirée.

### **III.3. Approche de Navigation basée sur l'Apprentissage par renforcement proposée [19]**

Comme on l'avait présenté au chapitre II, une méthode d'évitement de collisions et d'attraction vers la cible entre deux robots basée sur un RNAMF a été développée en utilisant LOCISS [20]. Il est cependant difficile de réaliser un bon évitement de collisions et d'attraction vers la cible pour trois robots et plus uniquement avec cette méthode : plus le nombre de robots augmente et plus les situations autour d'eux deviennent compliquées. Afin de réaliser un évitement de collisions et d'attraction vers la cible dans de telles situations, il est nécessaire d'adopter une approche adaptative. On introduit alors le paradigme de l'AR basé sur l'algorithme Q-learning dans le processus d'apprentissage du robot afin d'acquérir le comportement adéquat pour naviguer tout en évitant les collisions [19].

En effet, l'apprentissage par renforcement désigne un ensemble de techniques inspirées des méthodes "essai-erreur". Il vise à déterminer une loi de contrôle pour un système autonome dans un environnement inconnu. Ce genre de technique s'applique lorsqu'on suppose que la seule information sur la qualité des actions effectuées est la suivante : si une suite de comportements débouche sur une situation positive (respectivement négative), le système qui a produit ces actions est récompensé (respectivement puni). A partir de cette seule information, la procédure d'apprentissage vise à améliorer le choix des actions afin de maximiser les récompenses, d'où le terme d'apprentissage par renforcement [150].

Contrairement à la plupart des algorithmes d'apprentissage qui ont été étudiés dans le domaine de "Machine Learning" (apprentissage automatique), les techniques d'AR permettent de trouver les séquences d'actions optimales des tâches dont les décisions dépendent du temps où l'évaluation externe est rare et ni les effets des actions ni les délais en temps entre les actions et leurs effets sur la performance de l'apprenant n'est connue par l'apprenant au préalable [155]. Le but principal de l'apprentissage est de trouver une politique optimale pour la sélection des actions qui maximise une récompense future. Afin d'obtenir ce résultat, la plupart des techniques connues de l'AR estiment la valeur des actions, i.e., la



future récompense à laquelle on s'attend comme résultat de l'exécution de l'action en utilisant des techniques d'estimation récurrente [89, 155].

On considère donc un robot situé dans un environnement avec lequel il peut interagir. Il perçoit d'une part cet environnement (perceptions), il reçoit une récompense ( $r$ ), et il peut d'autre part agir (exécuter l'action  $a$ ).

On commence par dresser le schéma d'apprentissage spécifique au problème de navigation qu'on se propose de résoudre.

### III.3.1. Robot (l'agent) et capteurs

Rappelons que chaque SRA possède huit capteurs à IR (émetteurs et récepteurs) comme présenté au chapitre II (§.II.3.1), et est capable de détecter la position relative d'un autre robot dans son champ de perception. En transmettant/recevant les informations sur le mouvement, qui sont, la direction de déplacement et la vitesse, chaque robot peut reconnaître facilement le déplacement d'un autre robot [74].

On considère le cas où le robot ne connaît pas le modèle de l'environnement. De plus, l'algorithme utilisé ne cherche pas à apprendre de tels modèles avant de chercher une politique optimale. En effet, vu que le robot n'a aucune connaissance à priori de l'environnement auquel il fait face, sa structure doit lui permettre de se comporter uniquement à partir des interactions avec cet environnement.

Le robot doit :

- Observer son environnement.
- Prendre des décisions qui influencent l'évolution de l'environnement.
- Modifier son comportement.

### III.3.2. Description du Système

Dans la plupart des applications réelles, on peut se retrouver dans une situation où trois robots ou plus se retrouvent dans une même petite région parsemée d'obstacles. L'objectif du processus d'apprentissage est d'acquérir les comportements appropriés basés sur les informations communiquées par les capteurs afin que chacun des robots atteigne son but tout en évitant les collisions avec les autres robots et obstacles.

Dans le système proposé, une situation  $s$  de l'environnement est définie par l'état de l'ensemble des capteurs. Cet état représente l'existence ou non d'un SRA ou d'un obstacle dans chaque région de perception de chaque capteur. Le problème de l'évitement de collisions et de l'attraction vers la cible dans un tel environnement multi-robots est résolu en interagissant avec lui où chaque robot acquiert la capacité d'éviter intelligemment les collisions avec d'autres robots et avec les obstacles. Ce comportement est basé essentiellement sur un schéma d'AR acquis par interaction comme montré sur la figure III.1. L'objectif du robot est de collecter le maximum de récompenses. Il doit donc choisir les actions les plus récompensées et éviter les plus punies.

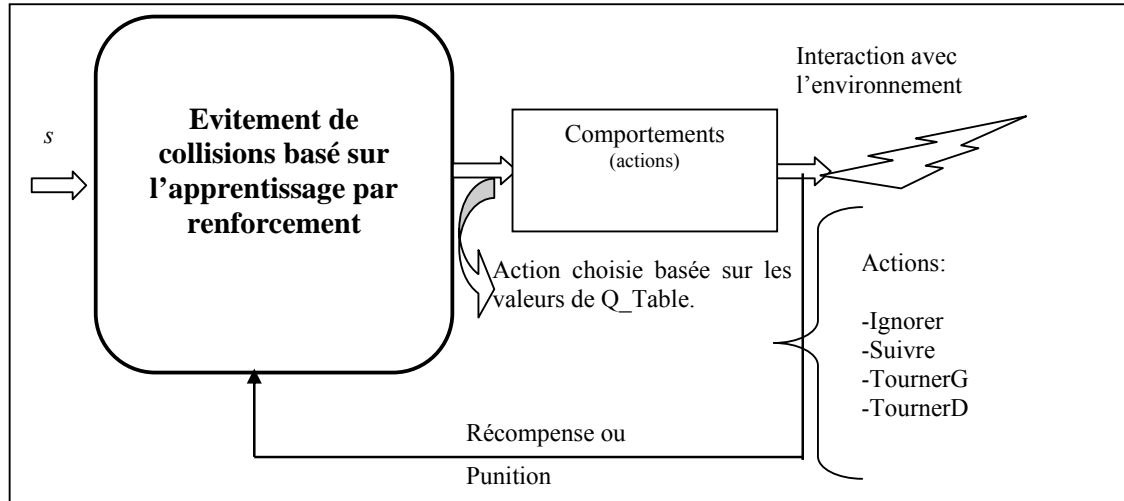


Figure III.1. Synoptique du Système.

### III.3.2.1. Architecture de contrôle utilisée

L'architecture de contrôle utilisée est une architecture comportementale où plusieurs comportements sont utilisés. Ce sont les comportements présentés ci-après. Chaque comportement est choisi en appliquant l'AR qui permet aux robots de prendre les bonnes décisions face aux situations auxquelles ils sont confrontés (voir figure III.2).

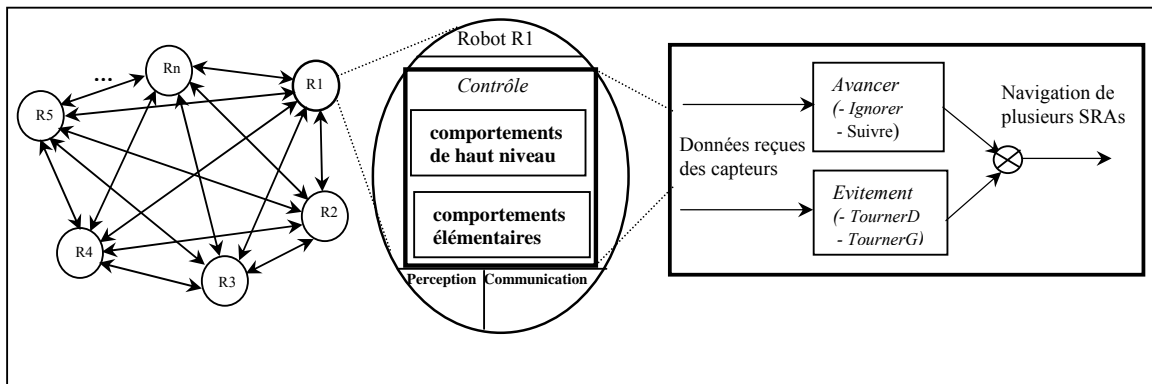


Figure III.2. Architecture adoptée pour le comportement d'évitement de collisions et d'attraction vers la cible de plusieurs SRAs.

### III.3.2.2. Heuristiques

Les heuristiques qui définissent la valeur du signal de renforcement peuvent faire l'objet de plusieurs modifications avant d'être fixées et c'est là qu'intervient l'opérateur. La valeur de ce signal de renforcement doit informer le robot si l'action (comportement élémentaire) qu'il vient d'effectuer est bonne ou mauvaise.

Une action est dite bonne par rapport à une situation, si elle ne conduit pas à une collision et le signal de renforcement qui est à ce moment là une "récompense" prend une valeur positive. Dans le cas contraire c'est à dire si l'action est suivie d'une collision cette action est "punie" par le signal de renforcement prenant cette fois-ci une valeur négative.

### III.3.2.3. Comportements élémentaires

Le robot n'a aucune connaissance a priori de l'environnement où il évolue. Sa structure doit lui permettre d'apprendre à se conduire seulement à partir des interactions avec l'environnement. Le robot utilise deux comportements élémentaires pour agir sur l'environnement afin de changer son état :

- Comportement *avancer*.
- Comportement *éviter*.

Ces comportements élémentaires sont spécifiés en termes de positions  $p$  et distances  $d$ .

Soit  $\mathcal{R}$  l'ensemble des robots tel que :

$$\mathcal{R} = \{R_i\}, 1 \leq i \leq N \text{ où } N \text{ est le nombre total de robots.}$$

Les coordonnées courantes du robot  $R_i$  et celles de sa cible sont respectivement :

$$p_i = \begin{pmatrix} x_i \\ y_i \end{pmatrix} \quad p_{ci} = \begin{pmatrix} x_{ci} \\ y_{ci} \end{pmatrix} \quad (\text{III.1})$$

La distance séparant le robot  $R_i$  de sa cible est donnée par :

$$d_{ci} = \sqrt{(x_{ci} - x_i)^2 + (y_{ci} - y_i)^2} \quad (\text{III.2})$$

#### III.3.2.3.1. Comportement avancer

Ce comportement permet au robot d'arriver à sa cible en s'approchant de celle-ci. L'objectif est de diminuer la distance entre le robot et sa cible et il se traduit par :

$$\frac{dp_i}{dt}(p_i - p_{ci}) \rightarrow 0 \quad (\text{III.3})$$

#### III.3.2.3.2. Comportement d'évitement

Ce comportement permet au robot de tourner lorsqu'il rencontre un obstacle. Il se traduit par le changement de la direction du robot et le déplacement dans cette nouvelle direction.

#### III.3.2.3.3. Actions générées

Ces deux comportements conduisent aux actions suivantes qui sont utilisées dans l'approche proposée :

- *Ignorer* : le robot se déplace vers son but ignorant les objets aux alentours.
- *Suivre* : le robot se déplace vers son but ignorant les objets aux alentours mais avec une vitesse réduite.
- *TournerG* : le robot tourne à gauche par rapport à son orientation.
- *TournerD* : le robot tourne à droite par rapport à son orientation.

L'ensemble des actions est alors:  $A = \{I, S, TG, TD\}$ .

### III.3.3. Apprentissage par interaction

Cet apprentissage permet au SRA d'acquérir des comportements adaptatifs avec une connaissance de l'environnement à priori minimale ou inexistante [119]. En fait, le robot apprend par des interactions par essai et erreur avec l'environnement. Ce comportement est acquis par apprentissage sans aucun signal d'enseignement à partir d'informations de capteurs comme illustré sur la figure III.3.

L'état des capteurs définit une situation  $s$  (voir figure III.3). Pour chaque situation  $s \in S$ , le robot peut prendre une action  $a$  de l'ensemble d'actions  $A$ . L'action  $a \in A$  pour la situation  $s \in S$  cause la transition de la situation à  $s'$ . L'objectif de l'AR est de trouver une politique optimale pour sélectionner l'action  $a$  pour la situation  $s$  qui maximise la somme escomptée des signaux de renforcement  $r$  reçus au cours du temps. L'algorithme Q-learning de Watkins [167] nous donne une solution efficace à ce problème.

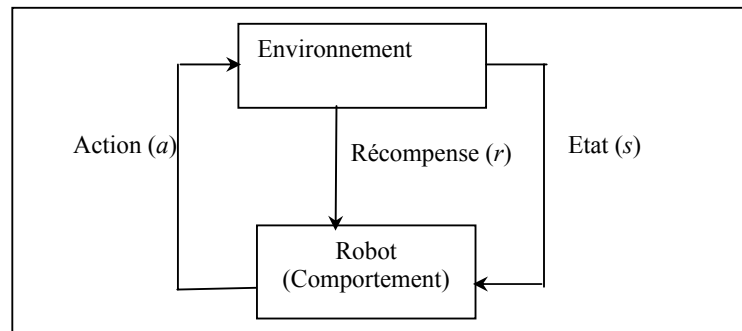


Figure III.3. Modèle de base de l'interaction Robot-Environnement.

#### III.3.3.1. Q-learning [150]

L'AR requiert seulement un critique (évaluateur) qui donne des récompenses scalaires (ou punitions) basées sur le comportement. Une introduction aux méthodes de l'AR est donnée [150]. Le fait de fournir un critique requiert une connaissance d'une certaine mesure de la tâche quant à sa réalisation ou non, et non le besoin de savoir comment réaliser la tâche. Le signal de récompense n'a pas besoin d'être donné immédiatement après la réalisation de l'action, puisque le vrai effet d'une action peut se manifester après un certain laps de temps. Le comportement s'améliore grâce à la connaissance des actions qui conduisent à des récompenses et à celles qui conduisent à des punitions. De cette manière, les bonnes et mauvaises expériences sont toutes les deux des parties importantes du processus d'apprentissage. L'algorithme Q-learning [167] est particulièrement flexible dans ce sens parce qu'il peut apprendre à partir d'actions qu'il n'a pas suggéré lui-même telles que celles provenant de données historiques.

L'agent apprenant est intégré dans un environnement d'une manière à ce qu'il arrive à discriminer l'ensemble  $S$  des situations distinctes du monde et peut prendre l'ensemble  $A$  d'actions dans le monde. Le monde est modélisé comme un processus de Markov, réalisant des transitions stochastiques basées sur l'état actuel et les actions prises par l'agent [150]. Soit  $T(s, a, s')$  la probabilité que le monde transitera à l'état  $s'$  en étant dans l'état  $s$  où l'agent exécute l'action  $a$ . En plus, pour chaque état  $s$  et action  $a$ ,  $r(s, a)$  est la valeur de renforcement de la prise de l'action  $a$  dans la situation  $s$ . En général, cette valeur est un scalaire aléatoire

variable, qui doit avoir une distribution stationnaire, mais la même paire situation-action peut avoir différents résultats dans les différents essais.

Le problème de l'AR est défini comme étant : trouver une politique qui maximise le renforcement escompté. Une politique  $\pi$  est une fonction de  $S$  vers  $A$ . Le renforcement escompté de la police  $\pi$  dans la situation  $s$  est définie par:

$$\sum_{t=0}^{\infty} \gamma^t r_t, \quad (\text{III.1})$$

où  $r_t$  est la valeur escomptée du renforcement obtenue à l'itération  $t$  donnée sachant que l'agent a débuté dans la situation  $s$  avec la politique exécutée  $\pi$ . La variable  $\gamma$  est le facteur d'escompte, il contrôle à quel degré les récompenses dans un futur proche affectent la valeur totale de la politique et est toujours un peu plus petit que 1.

L'algorithme Q-learning de Watkins donne une méthode efficace pour apprendre simultanément la dynamique du monde et construire la politique.

Soit  $Q^*(s, a)$  le renforcement escompté pour prendre l'action  $a$  dans la situation  $s$  et continue ainsi avec la politique optimale. Notons que  $V^*(s)$  est la valeur de  $s$  en supposant que la meilleure action est prise initialement, et ainsi :

$$V^*(s) = \max_a Q^*(s, a), \quad (\text{III.2})$$

$Q^*(s)$  peut ainsi être définie récursivement comme:

$$Q^*(s, a) = r(s, a) + \gamma \sum_{s' \in S} T(s, a, s') \max_{a' \in A} Q^*(s', a'), \quad (\text{III.3})$$

Notons aussi que, puisque  $V^*(s) = \max_a Q^*(s, a)$ ,

$$\pi^*(s) = \arg \max_a Q^*(s, a), \quad (\text{III.4})$$

est prise comme une politique optimale.

Des estimées incrémentales des valeurs de  $Q$  sont construites en-ligne parce que  $T$  et  $r$  ne sont pas connus initialement. En commençant avec  $Q(s, a)$  avec n'importe quelle valeur (mais généralement 0), à chaque fois qu'une action est choisie, la mise à jour des valeurs de  $Q$  se fait comme suit :

$$Q(s, a) = Q(s, a) + \alpha (r + \gamma \max_{a' \in A} Q(s', a') - Q(s, a)), \quad (\text{III.5})$$

Où  $r$  est la valeur de renforcement actuelle reçue pour avoir choisi l'action  $a$  dans la situation  $s$ ,  $s'$  est l'état suivant et  $\alpha$  est un taux d'apprentissage (valeur comprise entre 0 et 1).

Ayant les valeurs de  $Q$ , il y a une politique définie en prenant, dans n'importe quelle situation  $s$ , l'action  $a$  qui maximise  $Q(s, a)$ . Watkins [167] a montré que les valeurs de  $Q$  vont converger aux vraies valeurs de  $Q^*$  sous certaines hypothèses en incluant que chaque paire état-action est essayée indéfiniment. Ainsi, il est facile de calculer la politique optimale  $\pi^*(s)$  en sélectionnant simplement toutes les actions possibles à partir d'un certain état et sélectionner celle qui a la plus grande valeur :

$$\pi^*(s) = \arg \max_a Q(s, a) \quad (\text{III.6})$$

### III.3.3.2. Procédure d'apprentissage

L'algorithme Q-learning utilisé fonctionne en mettant à jour d'une manière incrémentale les valeurs attendues des actions correspondant à des états donnés. Pour chaque état possible, chaque action possible lui est assignée une valeur qui est une fonction de la récompense immédiate pour avoir choisi cette action-là et la récompense attendue dans le futur en se basant sur le nouvel état qui est le résultat pour avoir considéré cette action-là.

Le processus adaptatif d'acquisition de l'algorithme Q-learning est alors guidé, pour une situation spécifique perçue par le système de perception, suivant la procédure montrée sur la figure III.4. La base des valeurs qui est la table des valeurs (i.e., matrice  $Q$ ) de Watkins [167] est une série de valeurs permettant la sélection des actions et est calculée par l'équation III.5.

Un SRA apprend un comportement donné en répondant à la question s'il exécute bien ou mal une action dans chaque situation donnée. Comme "feedback", il reçoit une seule information de l'environnement. Le feedback est interprété comme une valeur de renforcement positive ou négative. L'objectif du système d'apprentissage est de maximiser le renforcement positif (récompense) et/ou de minimiser le renforcement négatif (punition) au cours du temps. Par des essais et/ou erreurs successifs, le robot détermine une fonction "mapping" (la fonction valeur représentée par  $Q$ ) qui est adaptée durant la phase d'apprentissage comme illustré sur la figure III.4.

L'objectif du robot, comme tout autre agent dans un schéma d'apprentissage par renforcement, est de rassembler le maximum de récompenses. Pour ce faire, il devra choisir les actions les plus récompensées et en éviter les plus punies : principe utilisé également dans notre approche.

L'approche de navigation proposée, comme le montre la figure III.4, se décompose en cinq (5) étapes :

- La perception de l'environnement,
- La sélection d'un comportement,
- L'exécution du mouvement,
- La réception du renforcement (émis par la fonction récompense),
- La mise à jour de la table des valeurs.

#### III.3.3.2.1. Perception de l'environnement

Le robot fait correspondre à chaque configuration des états des capteurs une situation  $s$  de l'environnement.

#### III.3.3.2.2. Sélection d'un comportement

Le but du robot, dans le cadre de l'apprentissage par renforcement est de trouver le comportement le plus efficace, c'est à dire savoir dans chaque situation possible, quelle action accomplir pour maximiser l'espérance de ses gains.

Pour des raisons d'exploration, le robot choisit une action de façon aléatoire et cela à partir d'une fonction qui lui permet de faire ce choix. La possibilité du choix diminue au cours du temps pour arriver à une pure exploitation [171]. En effet, l'une des questions qui se pose dans l'apprentissage par renforcement et non dans les autres types d'apprentissage est la question du compromis entre l'exploration et l'exploitation [156]. Pour obtenir un maximum de récompenses, le robot doit préférer les actions déjà tentées dans le passé pour leur apport

en récompenses. Mais pour découvrir de telles actions, il doit essayer les actions non sélectionnées auparavant. Le robot doit exploiter ce qu'il sait déjà pour obtenir les récompenses, mais doit également explorer pour améliorer sa sélection d'action dans le futur [156].

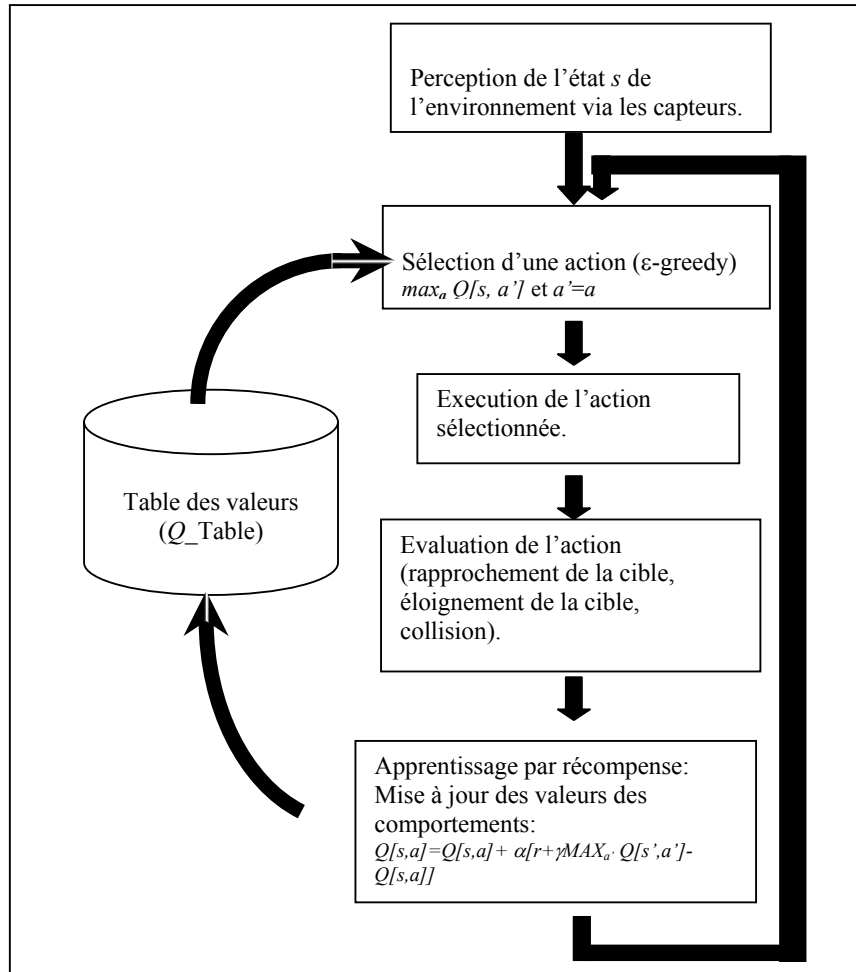


Figure III.4. Procédure d'apprentissage.

La politique utilisée donc pour résoudre ce compromis entre exploitation versus exploration est la politique usuelle  $\epsilon$ -greedy (où  $\epsilon \in [0,1]$ ) dont les étapes essentielles sont les suivantes [150]:

- (a) tirer uniformément un nombre  $p$  au hasard dans l'intervalle  $[0,1]$  ;
- (b) si  $p < \epsilon$ , tirer uniformément une action  $a_t$  au hasard dans  $A$  (exploration) ;
- (c) si  $p \geq \epsilon$ , choisir l'action  $a_t = \operatorname{argmax} Q(s_t, a)$ ,  $a \in A$  (exploitation).

### III.3.3.2.3. Exécution du mouvement

Après la sélection de l'action  $a$ , le robot sauvegarde sa position et exécute le mouvement approprié. Ce mouvement peut provoquer une collision comme il peut ne pas en provoquer. Dans le cas d'une collision le robot annule le mouvement qu'il vient d'exécuter en revenant à la position précédente, après cette étape le robot tente une nouvelle action  $a'$ .

#### III.3.3.2.4. Réception du renforcement (fonction de renforcement)

La fonction de renforcement définit le but dans un problème à AR. Elle relie les états perçus (ou les paires état-action) de l'environnement à un seul nombre, le renforcement, indiquant l'avantage intrinsèque de l'état et définit quels sont les bons et mauvais événements pour l'agent. L'objectif de l'agent doté d'un AR est de maximiser le renforcement global qu'il reçoit à long terme. Les renforcements sont les caractéristiques immédiates (traits) qui définissent le problème auquel l'agent fait face. Comme telle, la fonction de renforcement doit nécessairement être fixe [150]. La valeur de cette fonction est souvent mise au point à la main et émerge après plusieurs essais. En effet dans l'AR, le comportement est synthétisé en utilisant, comme unique source d'information, un scalaire, la valeur de renforcement qui évalue les actions des comportements. Le robot reçoit alors des renforcements positifs ou négatifs par rapport à l'utilité (i.e., avantage, attrait) de la situation obtenue comme conséquence de l'action exécutée.

Le robot reçoit donc la fonction récompense suivante durant l'apprentissage :

$$R_n = \begin{cases} p & \text{Si le robot se rapproche de sa cible} \\ n_1 & \text{Si le robot s'éloigne de sa cible} \\ n_2 & \text{S'il y a collision} \end{cases} \quad (\text{III.7})$$

Où  $p$ ,  $n_1$  et  $n_2$  sont les valeurs de renforcement.

Avec  $n_1 > n_2$ .

#### III.3.3.2.5. Mise à jour de la table des valeurs

La mise à jour de la table des valeurs est donnée par l'équation III.5 présentée dans §.III.3.3.1. Le robot est alors prêt pour une nouvelle étape avec la nouvelle situation  $s'$ , et une nouvelle table des valeurs est remise à jour, sur laquelle il se base pour un nouveau choix.

Grâce à l'approche proposée, les robots arrivent à acquérir le comportement adéquat pour réaliser la tâche demandée avec succès et ce en interagissant avec leur environnement.

#### III.3.3.3. Description de l'algorithme Q\_Learning utilisé

On propose deux manières pour la mise à jour de la table de valeurs ( $Q\_Table$ ), à savoir :

- Chaque robot met à jour sa propre table.
- Tous les robots mettent à jour la même table.

##### III.3.3.3.1. Algorithme Q\_Learning avec des tables de valeurs privées

Soit  $N$  le nombre de robots et  $Q_1, \dots, Q_n$  leurs tables respectives des valeurs.

Soient  $np_1$ ,  $np_2$ ,  $pp$ , les valeurs de renforcements tels que :

- $np_1$  est le renforcement que prend le robot s'il y a eu une collision.  $np_1$  est une valeur négative ( $np_1 < 0$ ).
- $np_2$  est le renforcement que prend le robot s'il n'y a pas eu de collision et si le robot s'est éloigné de sa cible.  $np_2$  est une valeur négative ( $np_2 < 0$  et  $np_2 > np_1$ ).
- $pp$  est le renforcement que prend le robot s'il n'y a pas eu collision et si le robot s'est rapproché de sa cible.  $pp$  est une valeur positive ( $pp > 0$ ).

L'algorithme correspondant est donné sur la figure III.5.



Pour chaque robot faire

1. Perception de l'environnement ( $s$ ).
2. Sélection de l'action  $a$  tel que :  
 $Max_a Q_i [s][a']$  et  $a'=a$ .  
 Suivant une fonction de choix pour améliorer l'exploration ( $\epsilon$ -greedy)  
 $a = \text{random} (a_0, a_1, a_2, a_3)$ .
3. Evaluation du comportement:  
 Si (collision)  $r \longleftarrow np_1$   
 Sinon  
 Si (rapprochement)  $r \longleftarrow pp$   
 Sinon  $r \longleftarrow np_2$
4. Mise à jour des paramètres :  
 perception de l'environnement ( $s'$ )  
 $Q_i[s][a] \longleftarrow Q_i[s][a] + ALPHA ( r + GAMMA Q_i[s'][a'] - Q_i[s][a] )$
5.  $s \longleftarrow s'$  Aller à 2.

Figure III.5. Algorithme Q-learning avec des tables de valeurs privées.

### III.3.3.3.2. Algorithme Q\_Learning avec une table de valeurs commune

Soit  $N$  le nombre de robots et  $Q$  la table collective des valeurs.

Soient  $nc_1$ ,  $nc_2$ ,  $pc$ , les valeurs de renforcements tels que :

- $nc_1$  est le renforcement que prend le robot s'il y a eu une collision,  $nc_1$  est une valeur négative ( $nc_1 < 0$ ).
- $nc_2$  est le renforcement que prend le robot s'il n'y a pas eu de collision et si le robot s'est éloigné de sa cible,  $nc_2$  est une valeur négative ( $nc_2 < 0$  et  $nc_2 > nc_1$ ).
- $pc$  est le renforcement que prend le robot s'il n'y a pas eu de collision et si le robot s'est rapproché de sa cible.  $pc$  est une valeur positive ( $pc > 0$ ).

L'algorithme correspondant est donné sur la figure III.6.

Pour chaque robot faire

1. Perception de l'environnement ( $s$ ).
2. Sélection de l'action  $a$  tel que :  
 $Max_a Q [s][a']$  et  $a'=a$ .  
 Suivant une fonction de choix pour améliorer l'exploration  
 $a = \text{random} (a_0, a_1, a_2, a_3)$ .
3. Evaluation du comportement:  
 Si (collision)  $r \longleftarrow nc_1$   
 Sinon  
 Si (rapprochement)  $r \longleftarrow pc$   
 Sinon  $r \longleftarrow nc_2$
4. Mise à jour des paramètres :  
 perception de l'environnement ( $s'$ )  
 $Q[s][a] \longleftarrow Q [s][a] + ALPHA ( r + GAMMA Q [s'][a'] - Q [s][a] )$
5.  $s \longleftarrow s'$  Aller à 2.

Figure III.6. Algorithme Q-learning avec une table de valeurs commune.

Cette procédure d'apprentissage utilise une seule table de valeurs commune partagée et mise à jour par tous les SRAs. L'apprentissage est donc partagé dans le groupe et les SRAs apprennent plus rapidement parce qu'ils profitent des expériences des autres robots. En fait, il est nécessaire de définir des mécanismes coopératifs qui aident à réduire la recherche en fournissant au robot d'autres sources auxiliaires d'expérience (tel que l'apprentissage partagé). Ces mécanismes coopératifs sont essentiels pour réaliser des tâches complexes où dans la plupart des cas les robots ne sont pas isolés mais intégrés dans une société qui guide et structure l'apprentissage. Ces robots ne peuvent pas espérer apprendre seuls des tâches complexes par essai et erreur seulement. Par contre, ils doivent être placés dans des environnements coopératifs, et des algorithmes doivent être développés afin de faciliter le transfert de la connaissance entre eux [69].

En résumé, l'avantage qu'apporte cette deuxième manière d'apprentissage est le partage de l'apprentissage au sein du groupe, ainsi les robots apprennent vite puisqu'ils profitent des expériences des autres.

### III.4. Conclusion

Afin de résoudre le problème de l'évitement de collisions et d'attraction vers la cible de plusieurs SRAs dans des environnements multi-robots, une approche se basant sur l'apprentissage par renforcement est proposée.

Dans ce type d'environnement multi-robots, il est difficile de préparer un signal d'enseignement (exemples d'entraînement dans le cas des RN) ou de collecter les échantillons nécessaires pour l'entraînement. En effet, l'AR permet, au minimum, en principe, de surpasser les problèmes de construction d'un modèle explicite du comportement à synthétiser ou alors le besoin d'une base d'apprentissage significative pour l'apprentissage supervisé. Avec cette approche d'apprentissage adaptatif, le robot est guidé uniquement par les renforcements retournés par l'environnement et l'apprentissage est réalisé d'une manière incrémentale et progressive puisque ses paramètres sont mis à jour à chaque itération et ainsi sont sensibles à tous les changements dans l'environnement et particulièrement à la table de valeurs. Cette dernière, qui donne une évaluation du comportement choisi, permet un apprentissage par interaction avec l'environnement. Deux manières sont traitées dans la prise en compte de cette table. La première utilise pour chaque robot sa propre table de valeurs et la seconde ne considère qu'une seule table commune à tous les robots. Dans ce cas précis, l'apprentissage est partagé dans le groupe et les SRAs apprennent plus rapidement parce qu'ils profitent des expériences des autres robots du groupe ce qui n'est pas le cas lorsque la table est privée.

Grâce à cette approche, les robots acquièrent le comportement d'évitement de collisions et d'attraction vers la cible afin de leur permettre de naviguer intelligemment dans des environnements inconnus. Ils sont alors capables a priori de faire face aux situations qu'ils n'ont jamais rencontrées au préalable grâce à leur capacité d'adaptation et de prise de décision. Ce qui a été rendu possible grâce à l'information délivrée par l'environnement avec lequel les robots sont en interaction continue.

Ce comportement va être utilisé afin de permettre aux robots d'apprendre des tâches plus complexes, tel que le transport d'objets lourds et/ou encombrants. Ils vont donc exploiter les connaissances apprises dans ce chapitre afin de réaliser la tâche de transport demandée. Ce travail fait l'objet du chapitre suivant.

## Chapitre IV

# Comportements Intelligents d'un Groupe de Systèmes Robotiques Autonomes: *Vers une navigation en formation*

### IV.1. Introduction

L'intégration de la notion de comportement de groupe dans le domaine des systèmes coopérants devient de plus en plus importante et utile dans diverses applications complexes. En effet, pour plusieurs applications robotiques spatiales, sous-marines et industrielles, il est plus intéressant et quelques fois nécessaire d'utiliser un groupe de SRAs à la place des êtres humains afin de réaliser l'objectif désiré [22, 45, 166]. Certaines de ces applications ont besoin de la tâche de transport d'objets lourds et/ou encombrants afin d'effectuer dans de bonnes conditions l'objectif fixé. Pour ce type de tâches, il est plus avantageux de se déplacer en formation d'un point donné à un autre.

Bien que diverses approches sur le problème du contrôle de plusieurs robots en formation ont été développées [45, 148, 149, 166], aujourd'hui plus d'efforts doivent être fournis afin d'appréhender et comprendre la notion de comportement de groupe et de comment le déplacement des SRAs peut être coordonné efficacement pour réaliser l'objectif visé.

Dans ce chapitre, une approche de navigation en groupe pour les SRAs est proposée. Cette approche exploite le travail de Sugihara [148, 149] en l'améliorant d'un côté et en l'étendant d'un autre côté afin d'effectuer la tâche de transport par le groupe en formation tout en évitant les obstacles dans des environnements partiellement structurés.

Pour ce faire, un apprentissage des figures géométriques est réalisé en utilisant l'AR. Une fois ces figures obtenues, les robots doivent se déplacer tout en gardant cette formation jusqu'à atteindre leur cible. L'approche proposée se base essentiellement sur l'interaction des robots avec l'environnement afin de mener à bien la tâche assignée en utilisant la navigation présentée au chapitre III.

Dans ce chapitre, un aperçu de différentes approches développées dans la littérature traitant de la notion de comportement de groupe est donné dans la section IV.2. Ces approches utilisent cette notion de groupe de différentes manières et à différents niveaux d'abstraction. L'approche proposée pour la résolution du problème de formation de figures géométriques et de leur déplacement en introduisant l'AR est développée dans la section IV.3. Celle-ci se base donc sur un ensemble de comportements élémentaires choisis judicieusement afin de réaliser l'objectif fixé. Elle se scinde en deux stratégies et qui sont : une stratégie de formation de figures géométriques et une stratégie de déplacement en formation. La première stratégie est totalement distribuée dans le sens où chaque robot planifie et décide de l'action à entreprendre

en se basant sur son processus de contrôle et de sa propre expérience grâce à l'AR. Alors que la seconde est centralisée puisque afin de garder la figure réalisée, un "meneur" est choisi pour guider le groupe vers sa cible. Cette stratégie exploite également l'AR et les robots interagissent alors avec l'environnement afin d'acquérir les bons comportements qui leur permettraient de réaliser l'objectif final.

## IV.2. Quelques approches de navigation en formation

Le problème de contrôler et coordonner plusieurs SRAs a intéressé plusieurs chercheurs ces dernières années. Le système reconfigurable développé dans [90] consiste en plusieurs cellules simples qui peuvent se détacher et se re-combiner d'une manière autonome afin de changer sa forme qui dépend de la tâche et de l'environnement. Mataric [112] a étudié comment des interactions locales d'agents artificiels peuvent être conçues afin de produire des comportements de groupe émergents et comment différents niveaux de l'intelligence individuelle d'un robot affectent la dynamique du groupe. Fredslund et Mataric [67, 68] ont étudié le problème de réalisation d'un comportement global dans un groupe de robots en utilisant uniquement une perception locale et une interaction dans le domaine d'une marche en bande de footballeurs. Dans ce travail, il n'est pas nécessaire d'avoir un système de positionnement global qui a quand même un avantage certain quant à la performance qui est beaucoup plus fiable. Wang [166] s'est également intéressé au comportement dynamique global d'un groupe de robots en développant des stratégies de navigation simples qui se basent sur une poursuite du "plus proche voisin" ou une poursuite "multi-voisins". Fredslund *et al.* [67] ont développé une approche comportementale pour des formations où chaque robot possède un coéquipier unique à suivre par un contrôle visuel. Une approche décentralisée pour assembler et maintenir des formations de robots est présentée dans [120]. Cette approche permet des formations de différentes formes telles que des lignes et des formations plus larges et est différente des autres approches puisqu'il n'y a pas d'ordres pré attribués aux membres de la formation. Un autre travail développé dans [85] est basé sur le concept des systèmes multi-agents. Ce travail consiste en l'organisation de plusieurs agents coopérants en utilisant la négociation et en complétant les fonctions manquantes ou perdues par d'autres agents. Sugihara et Suzuki [148, 149] ont développé une approche pour contrôler un groupe de robots d'une manière distribuée. Ils ont présenté des algorithmes distribués pour former une approximation de la forme d'un simple objet, distribuer les robots uniformément dans une zone donnée avec une forme particulière ou alors diviser les robots en deux ou plusieurs groupes. Chen et Luh [45] ont présenté des algorithmes basés sur ceux développés dans [148] pour former des modèles géométriques (patterns).

Cependant, ces approches n'introduisent pas dans leur processus de contrôle l'apprentissage qui est un point important quand on utilise des systèmes intégrés pour le comportement de groupe dans la réalisation de la tâche demandée. Dans ce type de tâches, chaque robot implique une interaction avec l'environnement où il cherche à atteindre son objectif malgré l'incertitude liée à l'environnement et où les robots doivent coordonner leurs mouvements pour réaliser la tâche demandée. Les actions du robot vont affecter l'état futur de l'environnement (e.g., la localisation suivante du robot), affectant de cette façon les options et les opportunités disponibles à l'agent ultérieurement.

### IV.3. Approche de navigation en formation proposée [14, 15]

A l'instar des insectes qui arrivent à réaliser des tâches complexes grâce à la coopération entre les individus, notre objectif est de développer une approche de coopération d'entités robotiques (SRAs), aux capacités décisionnelles et perceptuelles limitées, afin de réaliser des tâches complexes tel que le transport d'objets. Les intérêts sont le faible coût de chaque robot et la robustesse de l'exécution des tâches puisque le nombre important d'entités permet de compenser la défection de certains robots.

L'approche retenue repose sur une approche réactive et comportementale. Pour chaque entité robotique est définie une bibliothèque de comportements élémentaires fondés sur des couples stimulus-réponse (évitement d'obstacles, attraction vers la cible, etc.). Les comportements élémentaires sont organisés dans une architecture de contrôle hiérarchique fondée d'une part, sur le principe de subsumption introduit par R. Brooks [39] et d'autre part, sur la fusion de comportements. L'émergence de comportements complexes à partir des comportements élémentaires est liée à l'implémentation d'une architecture de contrôle distribuée sur les SRAs et à l'interaction des entités entre elles et avec leur environnement.

Afin d'appréhender leur environnement et pouvoir réaliser la tâche demandée, les robots sont donc amenés à interagir entre eux et avec leur environnement. Ils doivent communiquer, coordonner leurs mouvements, s'organiser afin d'atteindre l'objectif fixé. Ils sont amenés à apprendre un certain nombre de comportements intelligents de manière à réaliser la tâche demandée. Celle-ci se résume en la formation par les robots de figures géométriques correspondant à l'objet à transporter et ensuite de faire naviguer ce groupe tout en maintenant cette formation (voir Figure IV.1).

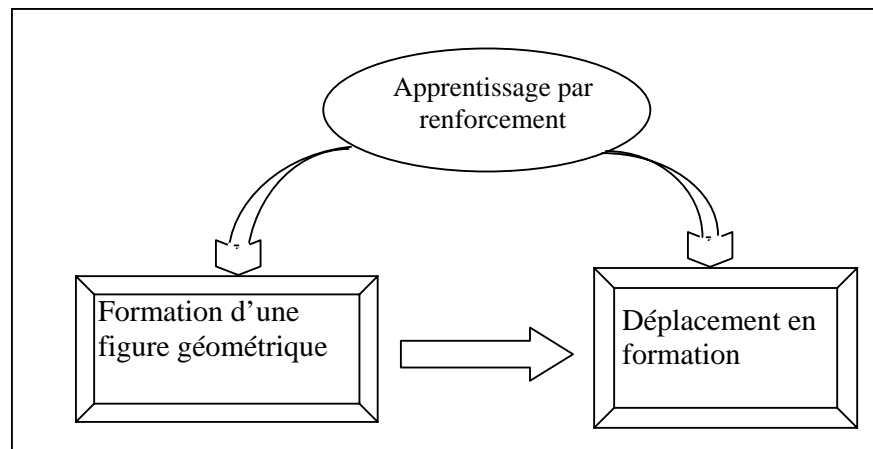


Figure IV.1. Synoptique de l'approche proposée.

Cette approche s'inspire de l'approche de formation de figures géométriques développée dans [148, 149] qui est améliorée dans le présent travail en :

1. introduisant l'apprentissage par renforcement dans la réalisation de ces figures,
2. terminant l'algorithme de la formation du cercle d'une manière automatique une fois la figure bien approximée au lieu de le faire manuellement comme dans [148],
3. dotant les SRAs d'une navigation par renforcement afin d'atteindre leur cible tout en évitant les obstacles.

### IV.3.1. Description du système

Afin de réaliser la tâche demandée, l'approche proposée permet grâce à l'AR de faire apprendre aux robots d'une part à former des figures et d'autre part à se déplacer en formation. La figure IV.2 donne le synoptique de cette approche où l'entrée du système correspond à la situation  $s$  où les robots se retrouvent. Ces derniers vont agir sur l'environnement afin de réaliser un maximum de récompenses et le processus se poursuit jusqu'à ce qu'ils apprennent leur tâche.

Une approche comportementale est adoptée afin de réaliser la tâche préconisée. Elle met donc en évidence, à partir de comportements élémentaires très simples, l'émergence de comportements beaucoup plus complexes lorsque les éléments sont en interaction les uns avec les autres. Les robots vont donc interagir les uns avec les autres et mettre en évidence l'émergence d'un but complexe, à savoir la formation d'une figure géométrique et son déplacement (voir figure IV.3).

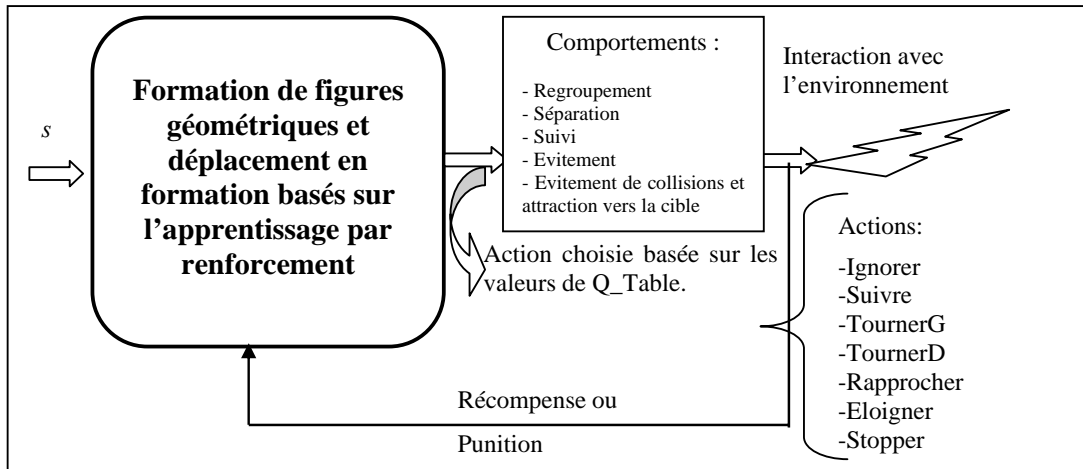


Figure IV.2. Synoptique du Système.

Une approche comportementale est adoptée afin de réaliser la tâche préconisée. Elle met donc en évidence, à partir de comportements élémentaires très simples, l'émergence de comportements beaucoup plus complexes lorsque les éléments sont en interaction les uns avec les autres. Les robots vont donc interagir les uns avec les autres et mettre en évidence l'émergence d'un but complexe, à savoir la formation d'une figure géométrique et son déplacement (voir figure IV.3).

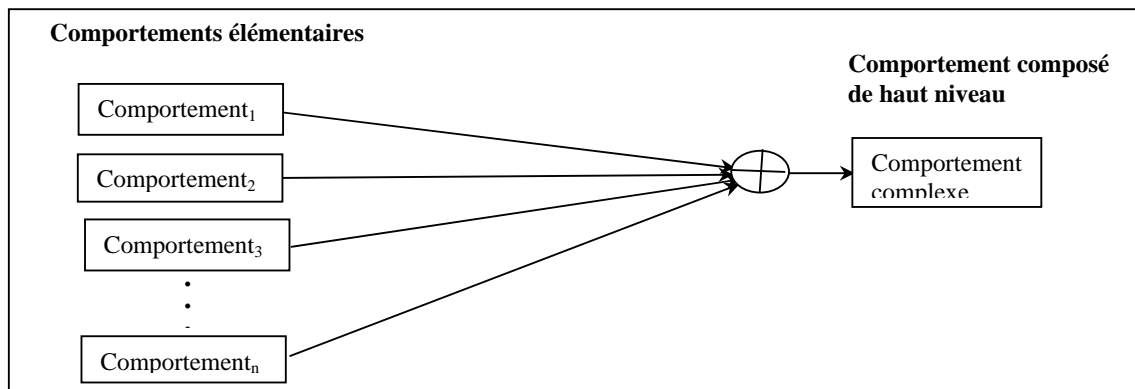


Figure IV.3. Obtention d'un comportement de haut niveau à partir de comportements élémentaires.

Le système de contrôle par renforcement proposé sur la figure IV.2 utilise donc des comportements élémentaires qui lui permettent de réaliser la tâche désirée. Ces comportements élémentaires découlent de la tâche en question. En effet, pour former un cercle par exemple, les robots doivent pouvoir se regrouper, se séparer, s'éviter, etc.

Les comportements élémentaires utilisés sont au nombre de cinq (5), à savoir :

1. Regroupement,
2. Séparation,
3. Evitement,
4. Suivi,
5. Evitement de collisions et attraction vers la cible.

Rappelons que ces comportements élémentaires sont spécifiés en termes de positions  $p$ , distances  $d$  et distances limites  $d_{reg}$  et  $d_{sep}$ .

$\mathcal{R}$  est l'ensemble des robots tel que :  $\mathcal{R} = \{R_i\}$ ,  $1 \leq i \leq N$  où  $N$  est le nombre total de robots.

Les coordonnées courantes du robot  $R_i$  et celles de sa cible sont respectivement :

$$p_i = \begin{pmatrix} x_i \\ y_i \end{pmatrix} \quad p_{ci} = \begin{pmatrix} x_{ci} \\ y_{ci} \end{pmatrix} \quad (IV.1)$$

La distance séparant le robot  $R_i$  de sa cible est donnée par :

$$d_{ci} = \sqrt{(x_{ci} - x_i)^2 + (y_{ci} - y_i)^2} \quad (IV.2)$$

La distance séparant deux robots  $R_i$  et  $R_j$  est donnée par :

$$d_{i,j} = \sqrt{(x_i - x_j)^2 + (y_i - y_j)^2} \quad (IV.3)$$

### IV.3.2. Stratégie de formation des figures géométriques par les SRAs [14, 15]

Les robots utilisent une approche comportementale afin de réaliser la figure géométrique demandée. En effet, des comportements de base judicieusement combinés permettent à un groupe de SRAs dispersés au hasard de former des figures géométriques. Grâce donc à l'AR, ces comportements permettent l'émergence d'un comportement plus complexe. L'action et l'interaction deviennent en fait plus importantes que l'intelligence individuelle que pourrait avoir un robot seul auquel on demande d'exécuter la même tâche (dans le cas où il peut le faire).

Ces figures géométriques se résument dans le présent travail à :

- Formation d'un cercle,
- Formation d'un cercle plein,
- Formation d'un polygone,
- Formation d'une ligne.

Cette stratégie est décentralisée dans le sens où tous les SRAs planifient leur déplacement (leurs mouvements) individuellement en se basant uniquement sur le but désiré du groupe (formation de la figure) et sur les informations nécessaires à leur déplacement (positions des autres SRAs) qui sont acquises par leurs systèmes de communication (voir §.II.3). Ainsi, ces SRAs, avec un comportement coopératif, agissent d'une manière

asynchrone [73, 166] afin de réaliser le but désiré avec succès. Ce but ne coïncide pas avec les buts des SRAs pris individuellement mais émerge de leur coopération en réalisant de simples sous-buts [105].

Dans le processus de distribution de la tâche entre les éléments identiques d'un groupe de SRAs, les problèmes suivants doivent être résolus afin d'assurer le succès :

- Formation d'une figure géométrique et coordination des SRAs dans le groupe (à l'intérieur du groupe) ;
- Alignement des orientations de tous les SRAs avec la direction de déplacement pour initier la stratégie de déplacement.

#### IV.3.2.1. Formation d'un cercle par les SRAs

L'objectif des robots est de réaliser un cercle à partir d'une distribution initiale aléatoire. Ils doivent donc apprendre à se regrouper ou à se séparer quand il le faut grâce à l'apprentissage par renforcement.

Les robots n'ont aucune connaissance à priori de l'environnement où ils se déplacent. Leur structure doit leur permettre d'apprendre à se comporter uniquement à partir de leurs interactions avec l'environnement. Ces robots vont utiliser un certain nombre de comportements élémentaires afin de réaliser la figure demandée à savoir un cercle. Chaque robot utilise donc trois comportements élémentaires afin d'agir sur l'environnement pour changer son état.

##### IV.3.2.1.1. Comportements élémentaires utilisés pour la formation d'un cercle

Les comportements élémentaires utilisés pour former un cercle sont illustrés sur la figure IV.4 et qui sont :

- Regroupement,
- Séparation,
- Evitement.

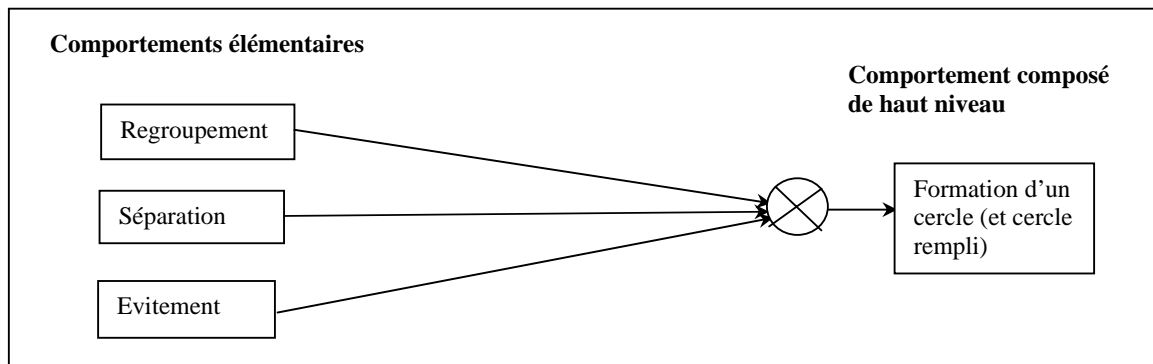


Figure IV.4. Obtention d'un comportement de haut niveau de formation d'un cercle à partir de comportements élémentaires.

##### a) Comportement de regroupement

Ce comportement permet au robot qui l'utilise de s'approcher d'un autre robot tout en maintenant une distance de séparation prédéfinie. Chaque robot doit alors avoir la capacité de savoir quand il doit se rapprocher d'un de ses congénères.



L'objectif de ce comportement est alors de réaliser et de maintenir une distance maximale  $d_{reg}$  entre les robots comme montré sur la figure IV.5.

$$\forall(R_j) \quad d_{i,j} \leq d_{reg} \quad (IV.4)$$

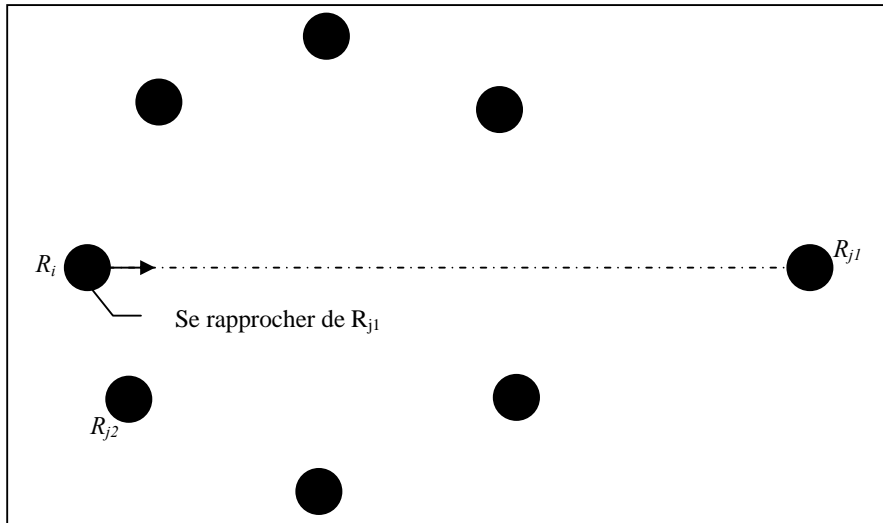


Figure IV.5. Comportement de regroupement.

Ce comportement utilise la distribution locale des robots (localisation des autres robots via le système de perception) de manière à décider quelle action entreprendre. Le robot  $R_i$  va détecter le robot le plus lointain de lui et va s'en rapprocher.

Ce comportement peut être résumé ainsi :

Si  $R_j$  est en dehors de  $d_{reg}$   
Alors  $R_i$  s'approche de  $R_j$

Où  $d_{reg}$  est la distance maximale permise entre les robots  $R_i$  et  $R_j$ .

$$d_{reg} = D$$

avec  $D$  le diamètre du cercle à former.

#### b) Comportement de séparation

Ce comportement permet aux robots trop proches de s'éloigner l'un de l'autre en maintenant une distance de séparation. Son objectif est donc de réaliser et de maintenir une distance minimale  $d_{sep}$  entre les robots. Il se traduit par :

$$\forall(R_j) \quad d_{i,j} \geq d_{sep} \quad (IV.5)$$

Ce comportement utilise lui aussi les localisations des autres robots pour déterminer les robots les plus proches. En effet, les deux robots les plus proches sont déterminés et suivant le cas, différentes actions sont prises.

#### 1<sup>er</sup> cas :

$\forall R_{j1}$

Si  $R_{j1}$  est à l'intérieur de  $d_{sep1}$   
Alors  $R_i$  s'éloigne de  $R_{j1}$

$d_{sep1} = D - \delta$  avec  $\delta$  : une constante positive (infiniment petit).

Lorsque la distance est inférieure à  $d_{sep1}$ , les robots vont s'éloigner en utilisant ce comportement (voir figure IV.6).

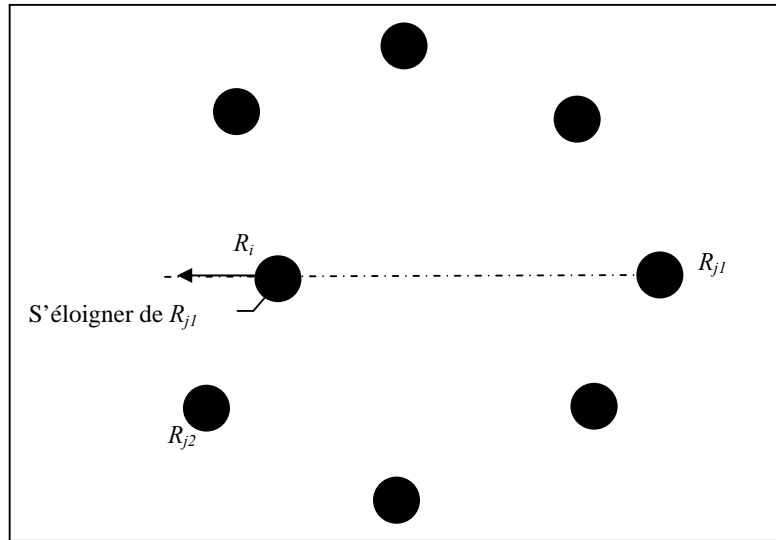


Figure IV.6. Comportement de séparation (1<sup>er</sup> cas).

**2<sup>ème</sup> cas :**

$\forall R_{j2}$ ,

Si  $R_{j2}$  se trouve entre  $d_{sep}$  et  $d_{reg}$

Alors  $R_i$  s'éloigne de  $R_{j2}$

Dans ce cas le robot  $R_i$  s'éloigne du robot  $R_{j2}$  afin de mieux se distribuer sur le lieu géométrique comme illustré sur la figure IV.7.

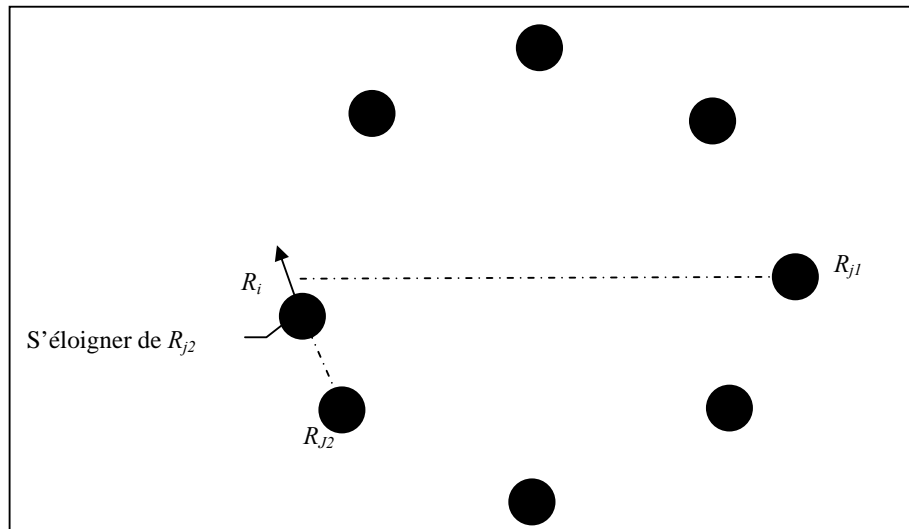


Figure IV.7. Comportement de séparation (2<sup>ème</sup> cas).

## c) Comportement d'évitement

Ce comportement permet aux robots de détecter les obstacles et de décider de l'action à entreprendre afin de les éviter (voir chapitre III).

## d) Actions utilisées

Ces comportements conduisent aux actions suivantes utilisées dans l'approche proposée :

*TournerG* : le robot tourne à gauche par rapport à son orientation.

*TournerD* : le robot tourne à droite par rapport à son orientation.

*RApprocher* : le robot se déplace vers le robot le plus lointain tout en maintenant une distance de séparation minimale prédéfinie.

*ELoigner1* : le robot s'éloigne du robot  $R_{j1}$  tout en maintenant une distance de séparation prédéfinie.

*ELoigner2* : le robot s'éloigne du robot  $R_{j2}$  tout en maintenant une distance de séparation prédéfinie.

*Stopper*: le robot s'arrête.

L'ensemble de ces actions est alors:  $A_{FC} = \{TG, TD, RA, EL1, EL2, S\}$ .

La table des valeurs  $Q$  est donc représentée par un tableau à deux dimensions où les lignes correspondent aux états (situations) et les colonnes aux actions ci-dessus.

**IV.3.2.1.2. Fonction de renforcement pour le cercle**

Dans cette section, la fonction de renforcement correspondant à la formation d'un cercle est donnée. Les robots reçoivent alors la fonction de renforcement suivante durant l'apprentissage :

$$R_c = \begin{cases} r_{c1} & \text{Si cercle formé} \\ r_{c_r} & \text{Si regroupement (d > D)} \\ r_{c_s} & \text{Si séparation ((d < D - \delta) ou (D - \delta < d < D))} \end{cases}$$

Où:

$$r_{c_r} = \begin{cases} r_{c2} & \text{Si s'approcher} \\ r_{c3} & \text{Si s'éloigner} \\ r_{c4} & \text{Si collision} \end{cases}$$

$$r_{c_s} = \begin{cases} r_{c5} & \text{Si s'éloigner} \\ r_{c6} & \text{Si s'approcher} \\ r_{c7} & \text{Si collision} \end{cases}$$

Après avoir défini les différents éléments constituant le système d'AR, le processus d'apprentissage est lancé en utilisant l'algorithme Q-learning de Watkins [167] (voir ci-dessous) appliqué à notre problème d'apprentissage de la formation d'un cercle. La table des valeurs de cet algorithme est une table commune (voir chapitre III). Grâce donc à l'utilisation

de la même table, les robots vont pouvoir réaliser la formation du cercle en interagissant seulement avec l'environnement.

<p>Pour chaque robot faire</p> <ol style="list-style-type: none"> <li>1. Perception de l'environnement (<math>s</math>).</li> <li>2. Sélection de l'action <math>a</math> tel que :  <math>Max_a: Qc [s][a']</math> et <math>a' = a</math>.            Suivant une fonction de choix pour améliorer l'exploration, dans notre cas <math>\epsilon</math>-greedy [150]  <math>a = \text{random} (TG, TD, RA, EL1, EL2, S)</math>.</li> <li>3. Evaluation du comportement: fonction de renforcement.            Valeur de renforcement obtenue après interaction avec l'environnement en exploitant les fonctions de renforcement données plus haut.</li> <li>4. Mise à jour des paramètres :            perception de l'environnement (<math>s'</math>)  <math>Qc[s][a] \leftarrow Qc [s][a] + \alpha (r + \gamma Qc [s'] [a'] - Qc [s][a])</math></li> <li>5. <math>s \leftarrow s'</math> Aller à 2.</li> </ol>
---

#### IV.3.2.1.3. Formation d'autres figures

Grâce à la formation en cercle, les robots peuvent former d'autres figures. En effet, ils peuvent réaliser un polygone ou une ligne en utilisant la formation du cercle [148].

##### a) Formation d'un polygone

Dans cette section, la formation d'un polygone à  $n$  côtés est présentée, où  $n \geq 3$  et est petit devant le nombre total des robots  $N$ .

Afin de réaliser la formation en polygone, les robots vont devoir tout d'abord apprendre à former un cercle. Une fois le cercle formé, chaque robot va devoir reconnaître son voisin de droite  $R_{droite}$  et son voisin de gauche  $R_{gauche}$  immédiats. Ces données vont permettre aux robots d'exécuter l'algorithme présenté par Sugihara *et al.* [148, 149] (Voir annexe A3) afin de réaliser un polygone à  $n$  sommets.

##### b) Formation d'une ligne

C'est un problème similaire à celui de la formation d'un polygone, dans le sens où la formation d'un segment de ligne peut être vue comme un cas particulier de polygone avec seulement deux sommets [148].

L'utilisateur sélectionne deux robots qui sont les deux extrémités de la ligne à former, les deux robots sélectionnés doivent être voisins. Ensuite, on exécute les deux étapes suivantes :

1. Déplacer les deux robots manuellement vers leurs positions finales désirées, de façon à ce que l'ordre dans lequel les  $n$  robots apparaissent dans la ligne sera le même dans l'approximation courante du cercle [148].

2. Permettre aux autres robots d'exécuter l'algorithme présenté dans [148] donné en annexe A3.

**IV.3.2.2. Formation d'un cercle plein par les SRAs**

Dans cette section, les robots doivent apprendre à former un cercle plein. Ils vont alors utilisés des comportements élémentaires pour réussir leur mission.

**IV.3.2.2.1. Comportements élémentaires utilisés pour la formation d'un cercle plein**

Ces comportements sont les mêmes que dans la formation du cercle, à savoir :

- Regroupement,
- Séparation,
- Evitement.

Les comportements de regroupement et d'évitement de collisions sont identiques à ceux utilisés dans le §.IV.3.2.1.1 Par contre le comportement de séparation est différent et est défini ci-après.

*a) Comportement de séparation*

Dans ce comportement, le robot le plus proche est déterminé de la manière suivante :

$\forall R_j$

Si  $R_j$  est à l'intérieur de  $d_{sep1}$

Alors  $R_i$  s'éloigne de  $R_j$

Lorsque la distance est inférieure à  $d_{sep1}$ , le robot  $R_i$  va s'éloigner en utilisant ce comportement.

*b) Actions utilisées*

Ces comportements conduisent aux actions suivantes :

*TournerG* : le robot tourne à gauche par rapport à son orientation.

*TournerD* : le robot tourne à droite par rapport à son orientation.

*RApprocher* : le robot se déplace vers le robot le plus lointain tout en maintenant une distance de séparation minimale prédéfinie.

*ELoigner1* : le robot s'éloigne du robot le plus proche tout en maintenant une distance de séparation prédéfinie.

*Stopper*: le robot s'arrête.

L'ensemble de ces actions est alors:  $A_{FCP} = \{TG, TD, RA, EL1, S\}$ .

**IV.3.2.2.2. Fonction de renforcement pour la formation d'un cercle plein**

La fonction de renforcement est donnée par :

$$R_{cr} = \begin{cases} r_{cr1} & \text{Si cercle plein formé} \\ r_{cr_r} & \text{Si regroupement (d > D)} \\ r_{cr_s} & \text{Si séparation (d < D - \delta)} \end{cases}$$

Où:

$$r_{cr_r} = \begin{cases} r_{cr_2} & \text{Si s'approcher} \\ r_{cr_3} & \text{Si s'éloigner} \\ r_{cr_4} & \text{Si collision} \end{cases}$$

$$r_{cr_s} = \begin{cases} r_{cr_5} & \text{Si s'éloigner} \\ r_{cr_6} & \text{Si s'approcher} \\ r_{cr_7} & \text{Si collision} \end{cases}$$

L'algorithme Q-learning est le même que dans la formation du cercle vide sauf que les actions et la fonction de renforcement sont celles définies dans cette section.

#### IV.3.2.3. Arrêt de la formation de SRAs

L'algorithme de Sugihara [148] n'a aucun mécanisme pour terminer son exécution, une certaine intervention humaine est nécessaire pour le faire arrêter. Dans notre cas, il n'y a pas d'intervention humaine puisque la tâche des robots est justement de réaliser la figure géométrique et tant qu'ils ne l'ont pas réalisée, le processus d'apprentissage continu.

#### IV.3.2.4. Alignement des orientations des SRAs

Les SRAs viennent à leurs positions dans le cercle de différentes localisations impliquant donc des orientations différentes. Ces directions doivent être alignées avec la direction de transport de l'objet. Pour ce faire, chaque SRA doit connaître les coordonnées du repère de référence et sa propre orientation actuelle par rapport à ce repère de référence en plus de la direction pour lancer la stratégie de déplacement [45].

### IV.3.3. Stratégie de déplacement en formation d'un groupe de SRAs

Une fois le groupe de SRAs a formé une figure géométrique et a aligné ses directions par rapport au but désiré (déplacer un objet d'une position initiale à une position finale), chaque SRA doit coordonner ses mouvements avec ceux des autres robots. Cet effort combiné tend alors à réaliser le but désiré de déplacer un objet lourd et/ou large. En effet, ce groupe de SRAs peut être comparé à une volée d'oiseaux, une nuée d'insectes ou un banc de poissons, où le modèle peut être assimilé à un suivi d'un meneur, dépendant de la position relative de celui qui le précède ou de n'importe quelle autre combinaison [45, 85, 166].

L'approche proposée est basée sur une formation en "meneur-suiveurs" [45, 122, 166] où le "meneur" change durant le déplacement [21] et est basée sur l'apprentissage par renforcement.

#### IV.3.3.1. Comportements élémentaires pour le déplacement en formation

Afin de se comporter intelligemment dans un environnement dynamique, les robots doivent acquérir les comportements d'évitement de collisions et d'attraction vers la cible et de suivi (voir figure IV.8). En effet, ces comportements ont été appris par interaction avec

l'environnement. Cet apprentissage permet aux robots de naviguer dans un environnement partiellement structuré en s'évitant mutuellement et en évitant les obstacles. D'ailleurs, cette approche, comme on l'a vu dans le chapitre III, s'adapte bien au problème posé à savoir le déplacement en formation où l'objectif final est commun à tous les robots. L'utilisation donc d'une table commune permet et facilite la réalisation de l'objectif.

Les comportements de base utilisés dans cette phase sont alors :

- Evitement de collisions et attraction vers la cible,
- Suivi.

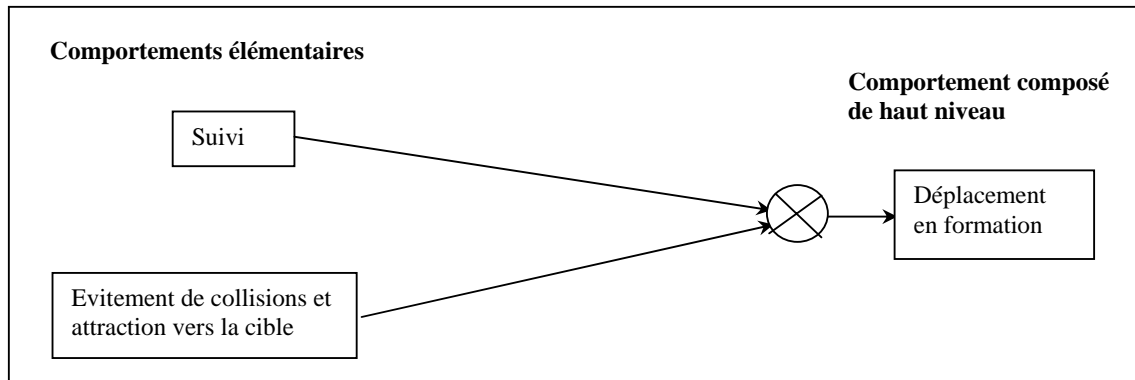


Figure IV.8. Comportements utilisés pour le déplacement en formation.

#### IV.3.3.1.1. Comportement de suivi

Ce comportement est utilisé par rapport au robot "suiveur". Il est réalisé en imposant au "suiveur" la direction du "meneur". Cette direction lui est transmise par le "meneur" via le système de perception LOCISS afin d'initier son déplacement. Ainsi il imite le "meneur" en adoptant la même orientation tout en maintenant la formation établie.

Soient  $R_i$  le "meneur" et  $R_j$  le "suiveur" et  $\theta_i$  l'orientation de  $R_i$ .

Si  $\theta$  change alors le robot "suiveur" ajuste son orientation en prenant  $\theta$  comme orientation.

#### IV.3.3.1.2. Comportement d'évitement de collisions et d'attraction vers la cible

Le comportement d'évitement de collisions et d'attraction vers la cible est celui défini au chapitre III. L'approche par renforcement présentée dans le chapitre III est donc exploitée afin de permettre aux robots de se déplacer sans collisions dans un environnement inconnu.

#### IV.3.3.1.3. Actions utilisées

Ces comportements conduisent aux actions suivantes :

*Ignorer* : le robot se déplace vers sa cible en ignorant les objets alentour.

*TournerG* : le robot tourne à gauche par rapport à son orientation.

*TournerD* : le robot tourne à droite par rapport à son orientation.

*Suivre* : le robot suit son "meneur" dans son déplacement.

L'ensemble des actions est alors :  $A_d = \{I, TG, TD, SV\}$ .

Grâce à ces actions, les robots vont pouvoir apprendre à atteindre leur cible sans collisions tout en maintenant la formation réalisée.

Les robots vont donc utiliser l'apprentissage par renforcement afin d'atteindre leur cible (une cible commune). Cependant, ils doivent d'abord déterminer le robot "meneur" qui mènera le groupe vers sa cible avec succès. En effet, dans cette approche, afin de décider du choix du "meneur", tous les SRAs acquièrent l'information (distances SRAs-obstacles et/ou SRAs-cible) via leurs systèmes de communication et de perception (capteurs) par interaction avec l'environnement. Ces informations sont nécessaires du fait du changement du "meneur" durant le déplacement.

#### IV.3.3.2. Détermination du "meneur"

La plupart des approches de navigation en groupe basées sur la méthode du "meneur-suiveurs" utilisent un seul SRA pour guider le groupe entier à une certaine destination afin d'exécuter la mission demandée. Dans ces approches, les "suiveurs" peuvent avoir un système de communication et/ou de contrôle très simple [45, 160]. Cependant, dans l'approche proposée, tous les SRAs doivent utiliser la même approche de navigation puisque le "meneur" change au cours du déplacement [21].

Dans l'approche proposée, un SRA approprié est sélectionné comme "meneur" durant le déplacement qui dépend d'un certain critère. En fait, le "meneur" est choisi comme étant le robot le plus proche de la cible afin de commencer le déplacement du groupe. Mais durant le mouvement, ce "meneur" change pour devenir le robot le plus proche des obstacles et est donné par :

$$d_{meneur} = \min_{i,j} (d_{R_i,O_j}, d_{R_i,C}) \quad (IV.6)$$

Où

$i = 1, \dots, n$  avec  $n$  le nombre de robots.

$j = 1, \dots, m$  avec  $m$  le nombre d'obstacles détectés par les robots.

$d_{R_i,O_j}$  représente la distance entre le robot et l'obstacle le plus proche.

$d_{R_i,C}$  représente la distance entre le robot et la cible.

Ce "meneur" doit alors guider le groupe de SRAs pour se diriger vers leur cible (position finale du groupe) tout en évitant les obstacles rencontrés en utilisant l'approche proposée dans le chapitre III.

#### IV.3.2.3. Formation en "meneur-suiveurs" des SRAs

Le robot "meneur" détermine le comportement du groupe et diffuse l'information (son orientation) qui est reçue par les robots "suiveurs" via leurs systèmes de communication. Cette information est utilisée par ces "suiveurs" pour déterminer leurs propres déplacements par rapport à ceux du "meneur".

Ainsi, pour garder la formation réalisée, le "meneur" détecte les obstacles, localise la cible et se dirige vers elle alors que les autres robots suivent ce "meneur" dans sa navigation. Un contrôle centralisé est donc adopté où le "meneur" donne à ses co-équipiers les directives pour naviguer dans des environnements partiellement structurés. De cette manière, les SRAs sont capables de rester groupés et dévient pour éviter les collisions tout en se dirigeant vers leur position finale. Ainsi, pour réaliser la stratégie centralisée de déplacement, ce groupe agit



à la base de perceptions locales d'une manière autonome dans des environnements partiellement structurés.

#### **IV.4. Conclusion**

Développer une approche de navigation en groupe contrôlant le comportement des SRAs est crucial pour la réussite d'applications requérant des groupes de SRAs coopératifs. Dans ce chapitre, une approche de navigation en groupe est proposée. Cette approche se base sur les formations en figures géométriques selon le principe "meneur-suiveurs" afin de résoudre le problème de la tâche de transport d'objets lourds et/ou encombrants. En effet, la formation de figures géométriques utilise la stratégie décentralisée de formation de figures géométriques alors que la formation en "meneur-suiveurs" utilise la stratégie centralisée de déplacement avec un "meneur" qui change.

En fait, la stratégie décentralisée de formation de figures améliore l'approche proposée par Sugihara et Suzuki [148, 149] en la dotant de comportements intelligents acquis par apprentissage. Plusieurs figures ont été réalisées à savoir : le cercle, le cercle plein, le polygone et la ligne. Une fois la formation réalisée, la stratégie centralisée de déplacement permet au groupe de naviguer intelligemment, sans collisions, dans des environnements partiellement structurés. L'approche de navigation en groupe proposée est alors capable de fournir au groupe de SRAs plus de flexibilité, d'autonomie et d'intelligence. On a alors appliqué les concepts de coopération de type collaboration simple, car les compétences des robots sont insuffisantes pour que ceux-ci réalisent l'objectif global (une forme géométrique). Cette coopération est réalisée en appliquant des contraintes de position et d'orientation comme présenté dans ce chapitre.



## Chapitre V

# Simulation, Tests et Résultats des Approches de Navigation Proposées des Systèmes Robotiques Autonomes

### V.1. Introduction

Afin de refléter les comportements des SRAs acquis par apprentissage et de montrer la capacité de généralisation des approches proposées de planification et de contrôle de la navigation des SRAs, la navigation est simulée dans différents environnements dynamiques partiellement structurés.

Chaque robot simulé ne peut disposer que de deux (02) données connues au départ: ses positions initiale et finale (cible), ainsi qu'une orientation initiale, et il est doté de huit (08) paires de capteurs IR comme ceux utilisés dans [5, 6] mesurant les distances entre le robot et les obstacles. A partir de ces données, il doit atteindre sa cible tout en évitant les éventuels obstacles rencontrés en utilisant les approches de navigation proposées. Ainsi, pour chaque approche, les résultats sont présentés dans différents environnements: à obstacles statiques, à obstacles dynamiques intelligents, et dans un environnement dynamique.

Les actions produites par ces approches qui conduisent à la réalisation de la tâche assignée sont exécutées grâce à un contrôleur flou. Celui-ci permet l'exécution de ces actions en un temps minimal permettant ainsi d'obtenir des trajectoires lisses.

### V.2. Environnement de simulation

L'environnement de simulation est constitué des environnements dynamiques partiellement structurés (obstacles statiques), des robots (positions, vitesses, orientations, et capteurs à IR), ainsi que leurs cibles.

#### V.2.1. Simulation des environnements dynamiques partiellement structurés

La simulation des environnements dynamiques partiellement structurés, voir figure V.1, implique la simulation des obstacles et des robots détaillés dans §.II.2.2. Dans cette simulation, les obstacles statiques sont représentés par des carrés, des rectangles, des triangles, et des cercles de couleur noire.

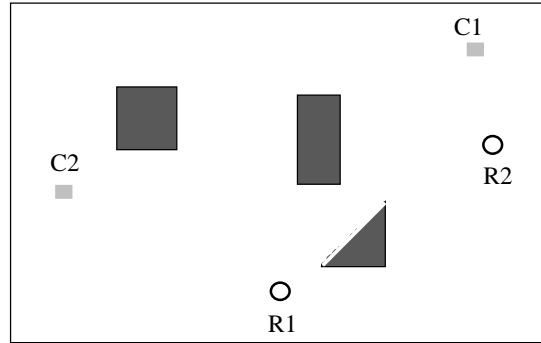


Figure V.1. Environnement dynamique partiellement structuré.

### V.2.2. Simulation des robots

Les robots, présentés au §.II.2.2, sont simulés par un cercle de bordure noire évoluant avec une vitesse donnée, voir figure V.1. Chaque robot est représenté par sa position (une composante selon l'axe des x et une composante selon l'axe des y) et son orientation puisque, dans cette simulation, les robots sont contrôlés en orientation.

### V.2.3. Simulation des capteurs à InfraRouges (IR)

A chaque pas, les régions de perception, définies pour chaque approche de navigation, sont parcourues, sur une distance d'un (01) jusqu'à trente (30) cm dans chaque région de perception. Ainsi, les obstacles ne sont détectés qu'à une distance du robot égale ou inférieure à 30 cm qui est la distance de détection des capteurs (le signal est atténué lors de la réflexion) et qui correspond en fait à 30 pixels dans notre application. La détection de pixels noirs correspond alors à la détection d'obstacles.

Le développement de l'environnement de simulation nous a permis non seulement d'entraîner et de mettre en oeuvre les différentes approches de navigation proposées, mais également de simuler les stratégies de formation de figures géométriques et de déplacement en formation dans des environnements dynamiques partiellement structurés.

### V.2.4. Langage de programmation

L'environnement de développement à choisir doit permettre la réalisation d'un produit optimisé répondant aux exigences du problème traité. Dans notre travail, on a choisi un langage de programmation orienté objet permettant la création de classes mères qui englobent des sous-classes sur lesquelles on travaille et qui permet également de créer des objets et de les manipuler.

L'application a été développée sur un PC PENTIUM de 512Mo de RAM, avec le langage Borland C++.

### V.2.5. Hypothèses

- La tâche attribuée aux robots est de réaliser un objectif fixé (atteindre une cible, former une figure géométrique et se déplacer en formation).
- Les coordonnées des cibles sont données au départ.
- Les robots n'ont aucune connaissance au préalable de l'environnement et ne le découvrent que pendant leur évolution.
- La communication entre les robots n'est établie que si la distance les séparant est inférieure à 80cm.

- Un robot ne peut percevoir un obstacle ou l'autre robot à l'arrêt qu'à une distance de 30cm.
- Comme nous l'avions mentionné dans le chapitre II, les robots peuvent évoluer avec l'une des trois vitesses suivantes :

$$V1 : 10 \text{ cm/s,}$$

$$V2 : 20 \text{ cm/s,}$$

$$V3 : 30 \text{ cm/s.}$$

Cependant, le robot roule à une vitesse maximale de 20 cm/s de manière à ne pas entrer en collision avec un objet donné de l'environnement. Ceci est dû au fait que le signal réfléchi par l'objet n'arrive au robot qu'à une distance de 30 cm [6].

### V.3. Approche neuronale de navigation des systèmes robotiques autonomes

L'environnement dans lequel les robots évoluent est à priori inconnu. Les robots doivent découvrir au fur et à mesure de leur évolution les obstacles et se découvrir mutuellement. Grâce au SHI (Réseau de Neurones Artmap Floue (RNAMEF)) développé (voir chapitre II), ces robots arrivent à atteindre leurs cibles respectives avec succès grâce à l'apprentissage réalisé par la AMFS. Cet apprentissage a permis aux robots d'éviter les obstacles d'un côté et à s'éviter mutuellement de l'autre.

On rappelle que cette simulation est réalisée pour un nombre de robots égal à deux (02).

#### V.3.1. Entraînement de RNAMEF

Ce SHI a été entraîné pour un nombre d'exemples égal à 276 exemples d'apprentissage représentés par le couple d'entrée  $\begin{pmatrix} \theta_1 \\ \theta_2 \end{pmatrix}$ .

Où  $\theta_i$  est l'angle relatif entre les directions des mouvements des robots.

Les paramètres de la AMFS pour l'entraînement de ce SHI sont donnés sur le tableau V.1.

Tableau V.1. Paramètres de la AMFS.

Paramètres	Valeurs
taux d'apprentissage $\eta$	1.0
constante (infiniment petit) $\lambda$	0.00001
vigilance de base $\sigma$	0.85
nombre de cycles $CN$	1
nombre de neurones de sortie $N$	40

Notons que dans les réseaux entraînés par la AMFS, la vigilance de base est le seul paramètre à ajuster empiriquement.

Ce RNAMEF a atteint le nombre de  $N=40$  nœuds catégories de sortie et converge sous un nombre de cycle  $CN=1$ .

Chaque règle a été représentée par un certain nombre de catégories de sortie. En effet,

- La règle R1 utilise 18 catégories de sortie,
- La règle R2 utilise 6 catégories de sortie,
- La règle R3 utilise 2 catégories de sortie,

- La règle R4 utilise 14 catégories de sortie.

Grâce à ces différentes catégories de sortie, les robots vont pouvoir naviguer dans un environnement donné sans difficulté. Le réseau a donc regroupé les différentes situations possibles dans les 40 catégories de sortie.

### V.3.2. Utilisation de RNAMEF

Afin de valider le comportement d'évitement de collision et d'attraction vers la cible des SRAs basé sur l'approche proposée, la navigation des robots est simulée dans différents environnements partiellement structurés. En effet, les résultats de simulation des figures V.2, V.3, V.4, V.5 et V.6 illustrent l'apprentissage du comportement d'évitement de collisions et d'attraction vers la cible.

Les robots arrivent donc à choisir le comportement adéquat parmi quatre (*Ignorer*, *Suivre*, *Tourner* et *Stopper*) afin de prendre l'action appropriée leur permettant de naviguer sans encombre dans un environnement a priori inconnu. Le robot choisit alors un de ces comportements suivant la situation à laquelle il est confronté.

#### V.3.2.1. Evitement de collisions et attraction vers la cible entre un SRA et des obstacles

Dans cet exemple, le robot évolue dans un environnement parsemé d'obstacles de différentes formes et grâce à l'approche proposée, il arrive à les éviter et à atteindre sa cible avec succès en utilisant les comportements "*Tourner*" et "*Ignorer*" comme montré sur la figure V.2. Le premier comportement lui permet d'éviter les obstacles et le second le guide vers sa cible.

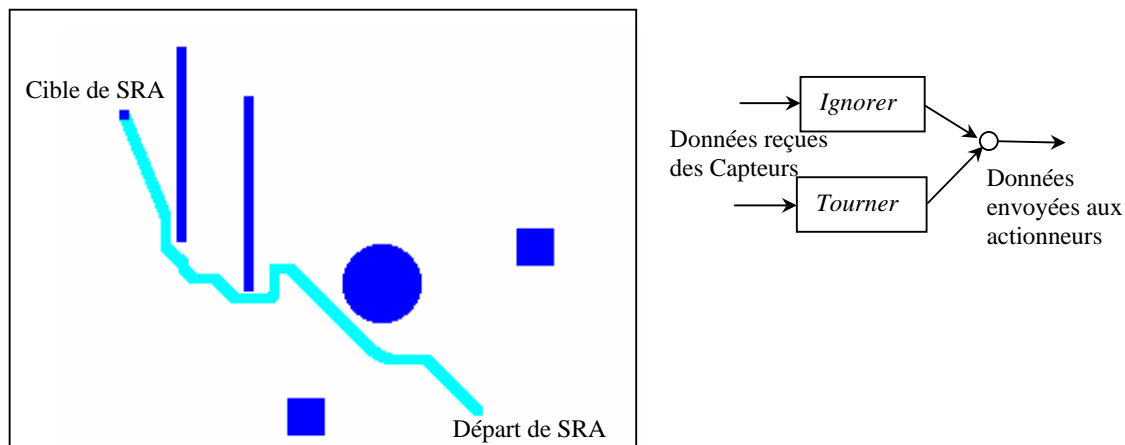


Figure V.2. Evitement de différents obstacles et attraction vers la cible par un SRA.

#### V.3.2.2. Evitement de collisions et attraction vers la cible entre deux SRAs dans un environnement simple

La simulation a été réalisée dans deux environnements différents, un ne contenant pas d'obstacles et l'autre parsemé d'obstacles de différentes formes.

##### V.3.2.2.1. Environnement sans obstacles

Les robots évoluent dans un environnement sans obstacles. Ils doivent s'éviter mutuellement afin d'atteindre leurs cibles respectives avec succès. En effet, grâce au SHI RNAMEF ces robots réussissent à s'éviter (voir figure V.3). En fait, ce sont les sorties OC1 à OC18 du réseau RNAMEF qui sont activées dans ce cas. Ces sorties correspondent à la règle

R1 et l'on remarque effectivement que les deux robots qui allaient l'un vers l'autre s'évitent parfaitement. Les deux robots se contournent en utilisant le comportement "Tourner" et se dirigent vers leurs cibles respectives en s'aidant du comportement "Ignorer".

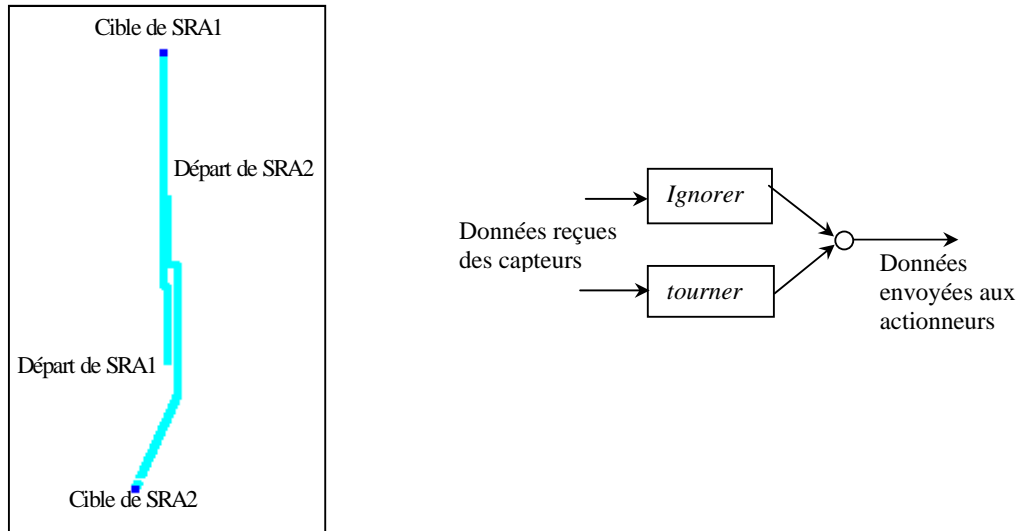


Figure V.3. Evitement de collisions et attraction vers la cible entre deux robots face à face.

Par contre dans l'exemple de la figure V.4, le robot SRA1 s'arrête pour laisser passer le robot SRA2 qui l'ignore et continue son chemin vers sa cible. C'est donc les comportements "Stopper" et "Ignorer" qui sont utilisés dans ce cas. Les robots ont donc appris les comportements adéquats donnés par les sorties OC19 à OC24 du réseau RNAMF correspondant à la règle R2.

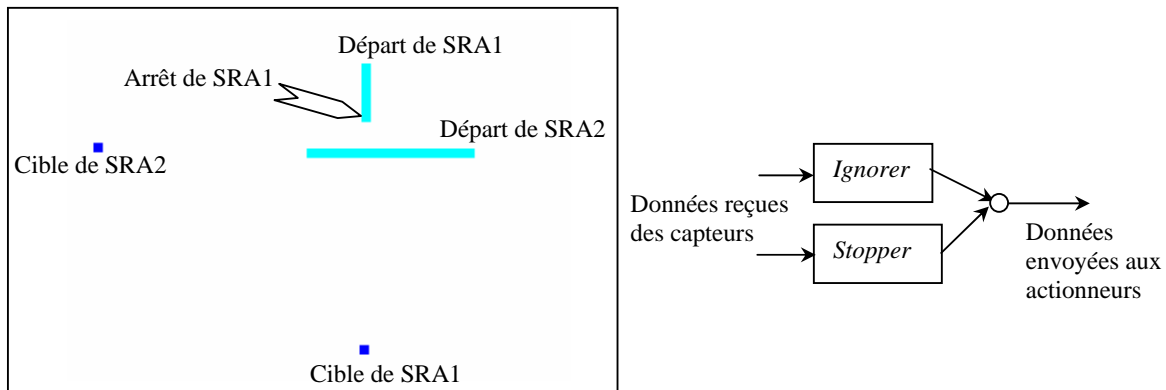


Figure V.4. Evitement de collisions et attraction vers la cible entre deux SRAs (arrêt de SRA1).

### V.3.2.2.2. Environnement avec obstacles

Dans cet exemple, les robots doivent, en plus de s'éviter mutuellement, éviter un obstacle. Grâce au réseau proposé, ils réussissent à éviter la collision en appliquant les règles R3 et R5 comme montré sur la figure V.5. Dans cette figure, le robot SRA2 ignore le robot SRA1 qui le suit et qui est amené à réduire sa vitesse pour ne pas entrer en collision avec SRA2. En effet, l'un des robots exploite le comportement "Ignorer" et le second le comportement "Suivre". Nous remarquons également qu'à l'approche d'un obstacle, le robot SRA2 l'évite en utilisant le comportement "Tourner". C'est donc les sorties OC25 à OC26 de RNAMF qui sont utilisées dans cet exemple.

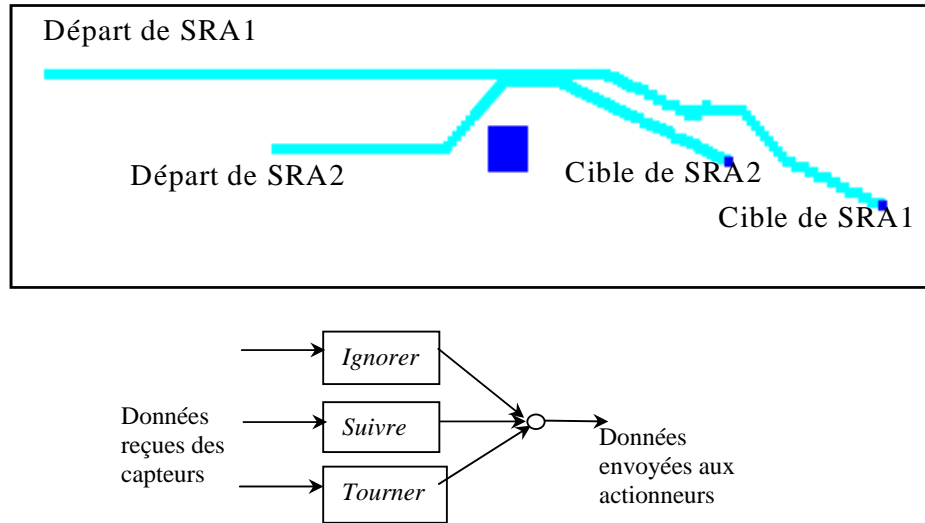


Figure V.5. Evitement de collisions et attraction vers la cible entre deux SRAs avec réduction de la vitesse (de SRA1).

### V.3.2.3. Evitement de collisions et attraction vers la cible entre deux SRAs dans un environnement complexe

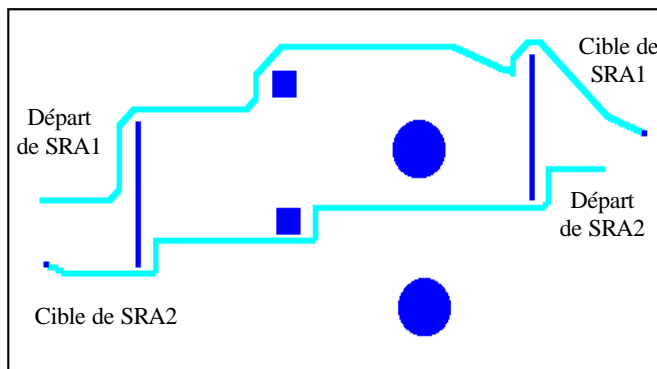
La figure V.6 montre l'évolution des robots dans différents environnements de plus en plus complexes. Les robots arrivent à atteindre leurs cibles en utilisant le réseau RNAMF malgré les différents obstacles de différentes tailles et de différentes formes.

Dans l'exemple de la figure V.6(a), les robots exploitent la règle R6 pour éviter plusieurs obstacles de différentes formes. C'est donc le comportement "Tourner" qui est utilisé à chaque fois que l'un des robots rencontre un obstacle.

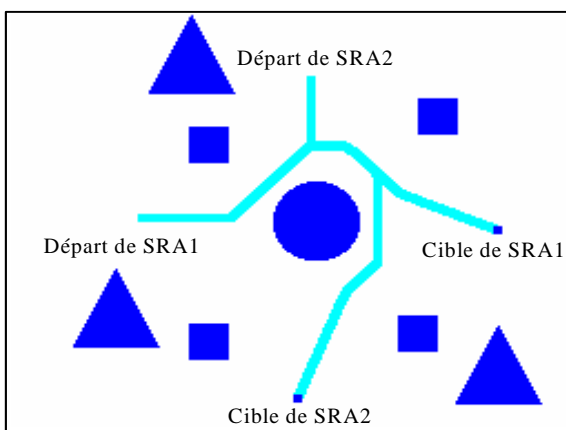
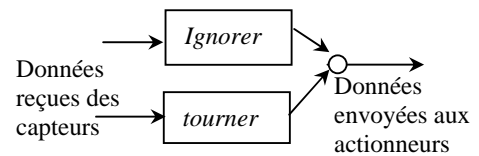
Dans le deuxième environnement de la figure V.6(b), les robots commencent par éviter les obstacles qu'ils rencontrent grâce au comportement "Tourner" et ensuite ils sont amenés à s'éviter vu qu'à un moment donné ils se retrouvent proches et donc SRA1 ignore SRA2 qui va suivre le premier.

Dans le troisième environnement de la figure V.6(c), le robot SRA2 va suivre le robot SRA1. C'est donc les sorties OC25 à OC26 qui sont utilisées afin de permettre aux robots de choisir le bon comportement à exécuter. En même temps, ils sont obligés d'éviter les obstacles se trouvant dans l'environnement. C'est pour cela qu'ils utilisent le comportement "Tourner" jusqu'à atteindre leurs cibles.

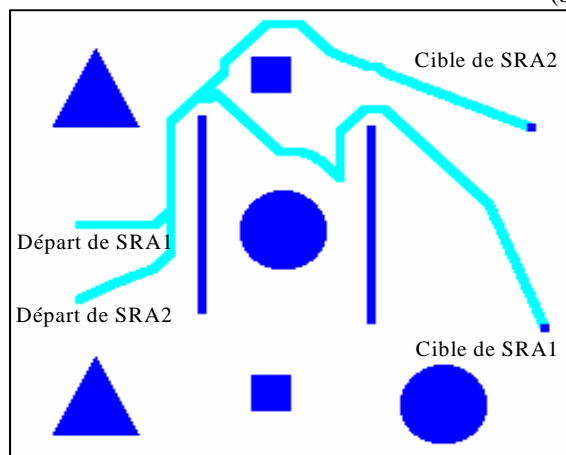
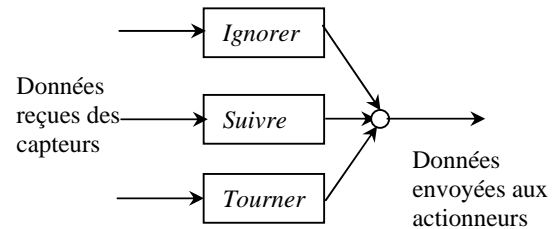




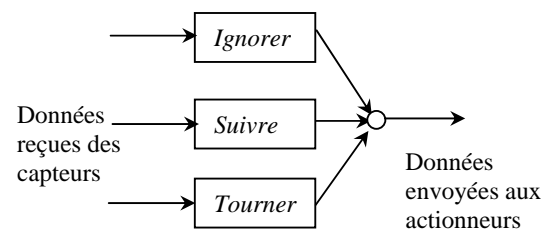
(a)



(b)



(c)



Dans les exemples (b) et (c), la vitesse de SRA2 est supérieure à celle de SRA1.

Figure V.6. Evitement de collisions et attraction vers la cible entre deux robots dans différents environnements partiellement structurés.

### V.3.3. Conclusion

Les résultats de simulation montrent la capacité de généralisation et d'adaptation de l'approche proposée à permettre aux SRAs d'éviter intelligemment les collisions avec les obstacles dans de nouveaux environnements non-visités auparavant. Un tel SHI est capable de

fournir à ces robots plus de traitement temps réel, plus d'autonomie et d'intelligence. Cependant, cette approche n'est applicable qu'à un nombre de robots ne dépassant pas deux. C'est pour cela qu'on a introduit la seconde approche basée sur l'AR et qui est présentée dans les prochaines sections.

#### V.4. Navigation des Systèmes Robotiques Autonomes (SRAs) dans un environnement dynamique : vers une navigation en groupe

Afin d'illustrer le comportement d'évitement de collisions et d'attraction vers la cible des SRAs dotés de l'approche par renforcement proposée, la navigation de ces robots est simulée dans différents environnements et les paramètres utilisés durant l'apprentissage sont résumés sur le tableau V.2.

Tableau V.2. Paramètres de l'algorithme Q-learning.

Paramètres	Valeurs
Taux d'apprentissage $\alpha$	0.3
Facteur d'escompte $\gamma$	0.9
Taux d'exploration $\varepsilon$	0.125

Les capteurs à IR dotant les robots [5, 6] mesurent la distance entre le bord du robot et l'obstacle et fournissent un vecteur à 8-dimensions  $s$ . Pour une situation  $s$ , le robot se déplace en utilisant une des quatre (04) actions ; *Ignorer*, *Suivre*, *TournerG*, *TournerD*, définies dans §.III.3.2.3.3.

Après plusieurs épisodes<sup>1</sup> correspondant aux déplacements du SRA du point de départ au point d'arrivée (sa cible), le SRA atteint sa cible avec succès sans collisions avec les autres robots et avec les obstacles. Le SRA a appris à éviter les collisions du départ à l'arrivée à la cible. En effet, une politique optimale a été trouvée et la performance d'atteindre sa cible sans collisions a été réalisée. Dans ce cas, la table de valeurs ne change pas et ainsi il n'y a plus d'amélioration du processus d'apprentissage du comportement d'évitement de collision. Cet apprentissage en-ligne permet au robot de devenir plus autonome parce qu'il s'adapte aux conditions changeantes et imprévues survenant dans l'environnement.

Les résultats de simulation présentés dans les sections suivantes illustrent l'apprentissage des robots du comportement d'évitement de collisions et d'attraction vers la cible par interaction avec l'environnement.

##### V.4.1. Apprentissage du comportement d'évitement de collisions et d'attraction vers la cible des SRAs

Dans l'approche proposée, l'apprentissage se fait via une table appelée table de valeurs. Dans cette table, les récompenses (ou les punitions) durant le processus d'apprentissage ont été calculées grâce à la formule de Watkins [167]. En effet, ces récompenses (ou punitions) permettent de prendre les différentes actions que le SRA va exécuter à chaque situation (voir figure V.7). En fait, pour une situation donnée, le SRA va choisir l'action ayant la plus grande récompense et c'est cette action qui sera exécutée. A la fin de l'apprentissage, la table de

<sup>1</sup> Episode : c'est un essai qui correspond au déroulement de la tâche du début à la fin. Dans notre cas, un épisode représente le déplacement du robot du point de départ au point d'arrivée (la cible).

valeurs obtenue renfermera toutes les connaissances nécessaires à la prise de la meilleure décision.

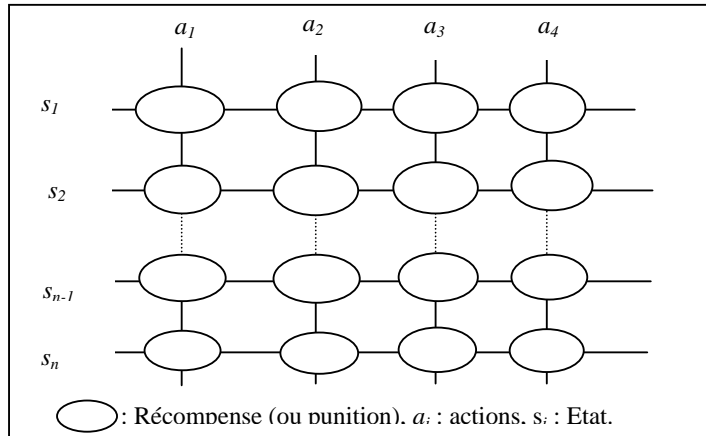


Figure V.7. Table de valeurs.

#### V.4.1.1. Apprentissage du comportement d'évitement de collisions et d'attraction vers la cible par un SRA seul

Sur la figure V.8, un SRA essaie de trouver son chemin afin d'atteindre sa cible. Le robot prend deux (02) chemins différents comme montré sur la figure V.8. Sur la figure V.8(a), le robot trouve facilement son chemin vers sa cible en passant entre deux obstacles. Par contre sur la figure V.8(b), le passage entre les deux obstacles est si petit que le robot ne peut passer. Ainsi, il essaie un autre chemin par essai et erreur jusqu'à trouver le nouveau chemin le conduisant à sa cible (voir figure V.8(b)). Ceci a été possible grâce à l'apprentissage par interaction et particulièrement à la capacité d'exploration de l'approche par renforcement utilisée et à son auto-apprentissage.

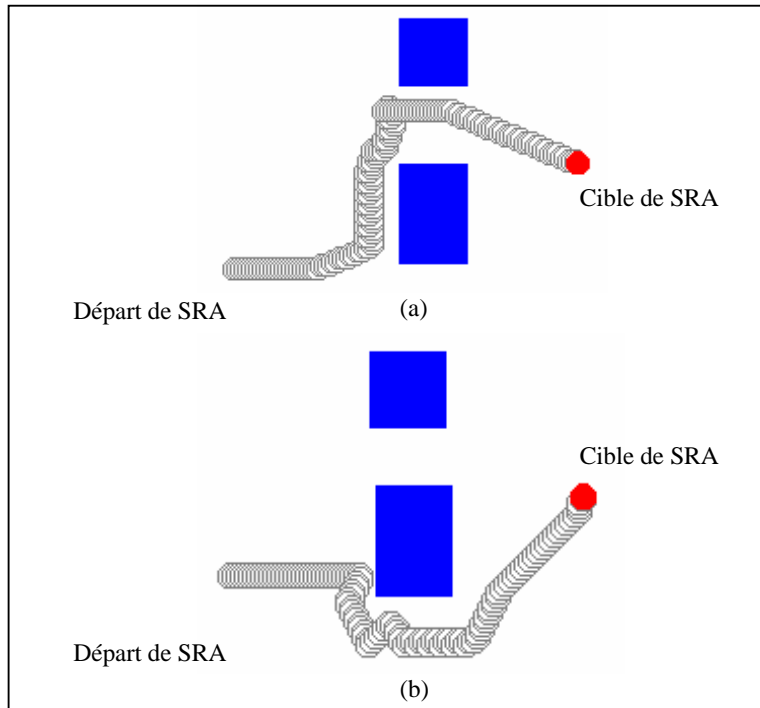
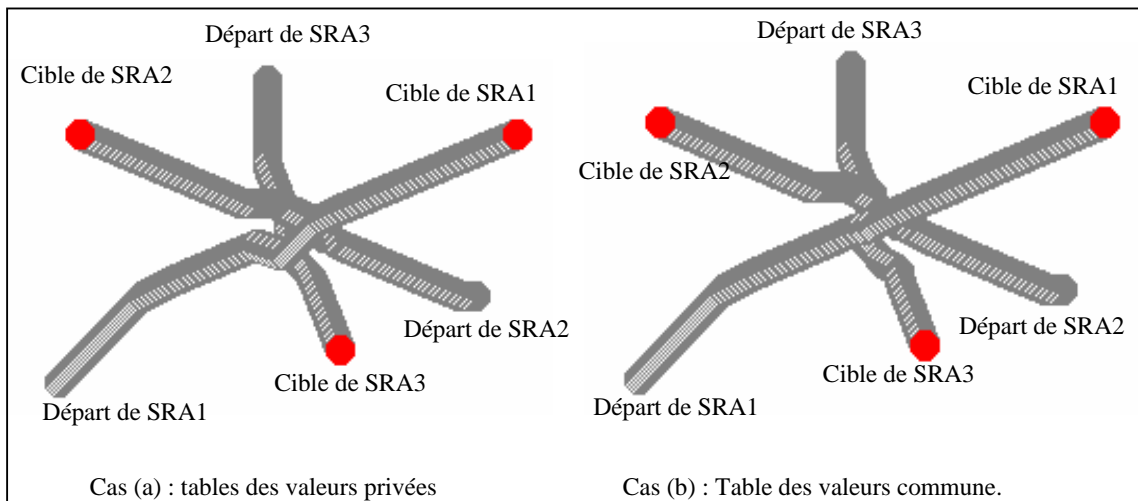


Figure V.8. Evitement de collisions entre un SRA et des obstacles.

En effet, il est connu que l'un des défis dans l'AR et non dans les autres types d'apprentissage est le compromis entre l'exploration et l'exploitation comme on peut le voir dans ces résultats de simulation. Afin d'obtenir plusieurs récompenses, le robot doit préférer des actions déjà essayées dans le passé produisant une récompense. Ces actions sont choisies par rapport à la méthode  $\epsilon$ -greedy [150]. C'est une alternative simple pour se conduire par rapport aux résultats calculés la plupart du temps, mais une fois de temps en temps avec une petite probabilité  $\epsilon$ , sélectionner une action aléatoirement indépendamment des résultats des estimées de la table de valeurs [150]. En effet, afin de découvrir de telles actions le robot doit essayer des actions qu'il n'a pas sélectionnées dans le passé. Le robot doit exploiter ce qu'il connaît déjà afin d'obtenir des récompenses mais il doit également explorer afin d'avoir de meilleures sélections d'actions dans le futur. Il doit essayer une variété d'actions et progressivement favoriser celles qui apparaissent être les meilleures [129, 150].

#### V.4.1.2. Apprentissage du comportement d'évitement de collisions et d'attraction vers la cible par plusieurs SRAs

La navigation de plusieurs SRAs dans un environnement dynamique est illustrée dans les figures V.9 et V.10. Les robots apprennent avec succès comment atteindre leurs cibles sans collisions entre eux et avec les obstacles. En effet, grâce à l'apprentissage réalisé, le problème de la navigation de plusieurs robots dans des environnements dynamiques et inconnus est résolu où les robots ne sont pas considérés comme des obstacles dynamiques mais comme des co-équipiers avec lesquels ils communiquent et coopèrent afin de réaliser la tâche demandée.



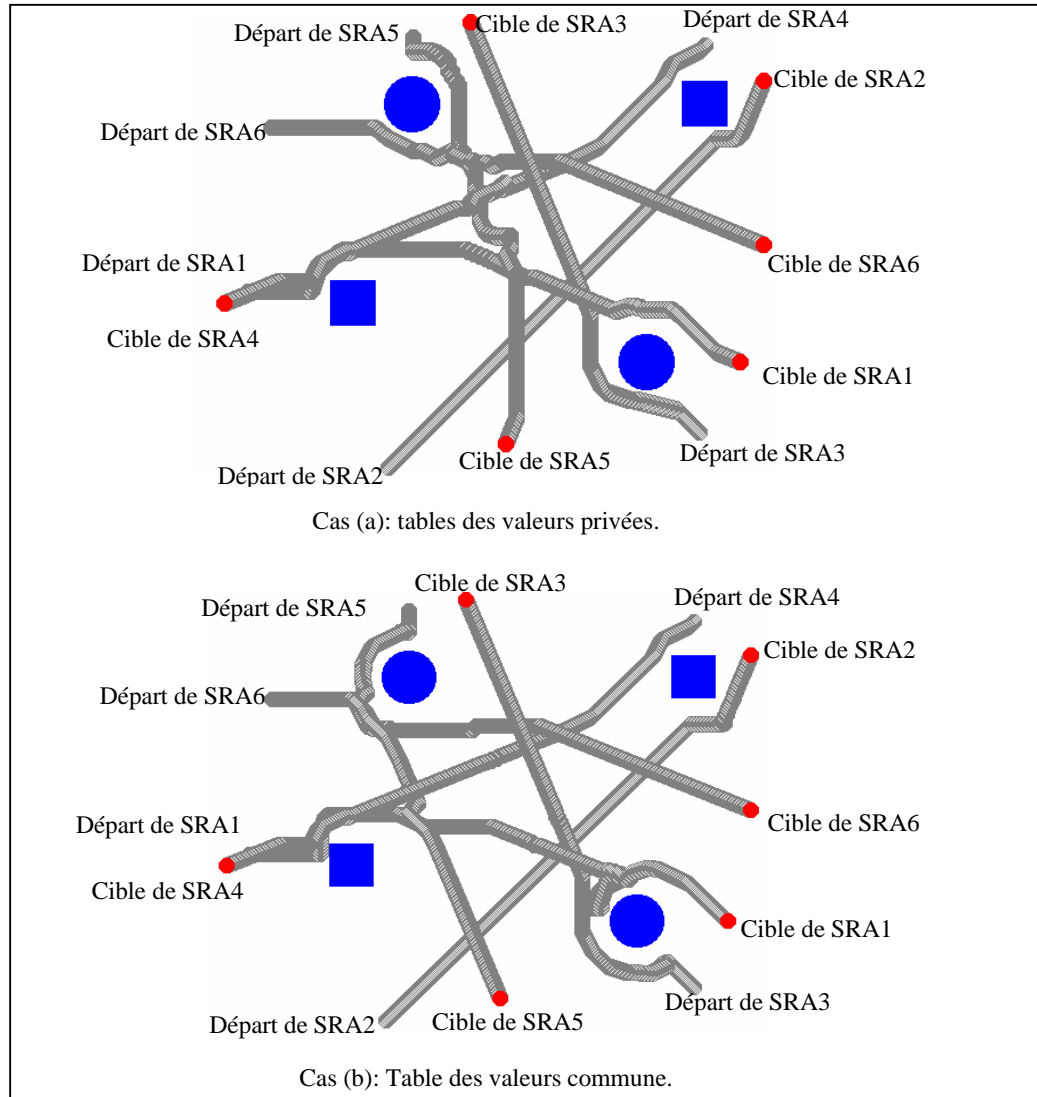


Figure V.10. Evitement de collisions et attraction vers la cible entre plusieurs SRAs dans un environnement dynamique.

Dans ces figures, les résultats des deux types d'apprentissage utilisés (voir §.III.3.3.3.1 (tables de valeurs privées) et §.III.3.3.3.2 (table de valeurs commune)) sont donnés. Les deux algorithmes permettent aux robots d'éviter les collisions avec les obstacles et entre eux tout en se dirigeant vers leurs cibles respectives.

L'approche proposée réalise donc une navigation sans collisions dans un environnement parsemé d'obstacles même quand le nombre de robots est important comme illustré sur la figure V.10, ce qui n'était pas possible avec la première approche. Les deux types d'apprentissage considérés donnent des résultats différents puisque les chemins pris par les robots dans le même environnement sont différents mais ils réalisent la navigation vers leurs cibles.

### V.4.1.3. Caractéristiques et performance de l'approche proposée d'évitement de collisions et d'attraction vers la cible

Dans l'approche proposée, une seule table de valeurs commune est utilisée par tous les robots afin d'apprendre le comportement d'évitement de collisions et d'attraction vers la cible. Cette idée d'utiliser une seule table de valeurs a été inspirée par le fait que cette table représente les informations nécessaires à la navigation et donc pourquoi ne pas partager cette connaissance parmi les robots. C'est pour cela qu'on a considéré les robots utilisant la même table de valeurs plutôt que le cas où chaque robot possède sa propre table de valeurs. De cette façon, les robots vont exploiter la connaissance des autres afin d'apprendre le comportement en question. C'est en fait une manière de coopérer efficace qui permet de partager toutes les connaissances acquises par les différents SRAs. Cette hypothèse est vérifiée dans cette section où une comparaison est réalisée entre l'utilisation d'une ou de plusieurs tables de valeurs.

#### V.4.1.3.1. Utilisation d'une table de valeurs commune versus plusieurs tables de valeurs privées

Pour les trois SRAs de la figure V.9 navigant dans le même environnement, l'évitement de collisions et l'attraction vers la cible a été appris après 4 épisodes quand la table de valeurs est commune alors qu'il est de 11 quand les robots utilisent des tables de valeurs privées (voir figure V.11).

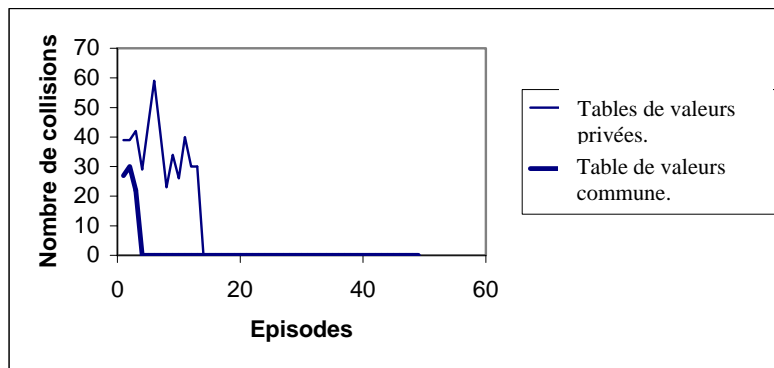
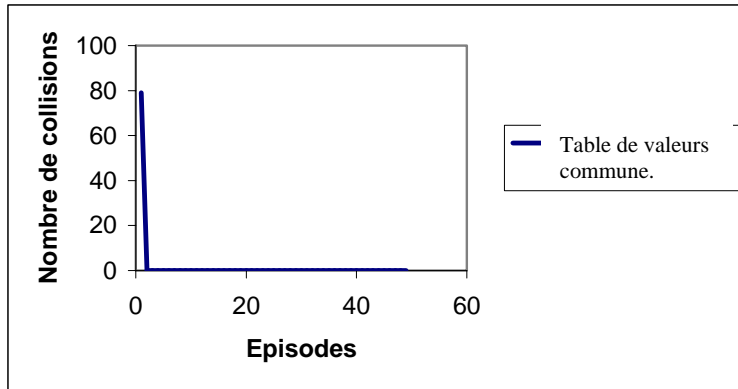


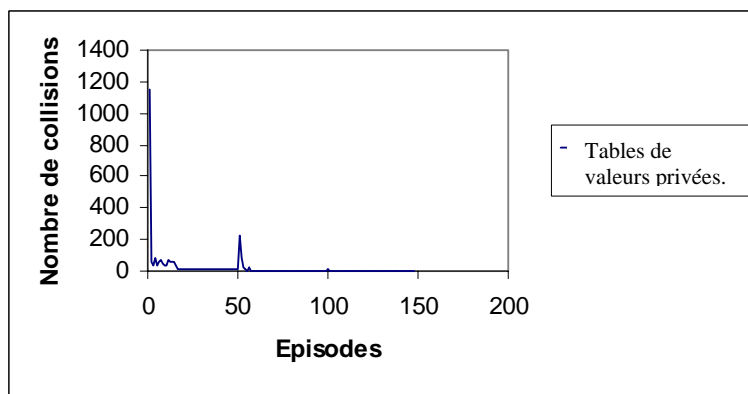
Figure V.11. Comportement de trois SRAs durant l'apprentissage en utilisant l'approche proposée d'évitement de collisions et d'attraction vers la cible.

Sur la figure V.12 correspondant à la figure V.10 où plusieurs robots évoluent, le comportement d'évitement de collisions et d'attraction vers la cible a été appris après 2 épisodes lorsque la table de valeurs est commune et il est de 55 lorsqu'elles sont différentes.

D'après ces résultats, on peut déduire que lorsque la table de valeurs est commune le nombre d'épisodes est plus petit. Ceci s'explique par le fait que l'utilisation d'une table commune revient en fait à partager les informations des uns et des autres. Les robots exploitent donc l'avantage du partage de connaissances afin d'apprendre plus rapidement parce que ce qui a été appris par un robot est également connu (appris) par les autres SRAs. Le robot n'est pas obligé d'apprendre lui-même une autre fois une situation déjà apprise par un autre SRA. Ainsi, les SRAs apprennent plus rapidement à naviguer quand ils partagent la table de valeurs que dans le cas contraire (càd, quand ils ont des tables privées).



(a) SRAs avec une table des valeurs commune.



(b) SRAs avec des tables de valeurs privées.

Figure V.12. Comportement de plusieurs SRAs durant l'apprentissage en utilisant l'approche proposée d'évitement de collisions et d'attraction vers la cible.

#### V.4.1.3.2. Performance de l'approche proposée

Des courbes des figures V.13 et V.14, on peut voir qu'initialement, le robot utilise un nombre important d'itérations parce qu'à ce moment-là, le robot n'a aucune connaissance de son environnement (plus de 500 pour 3 robots et plus de 4000 pour 6 robots). L'apprentissage se poursuit et le robot gagne de plus en plus de connaissances. Le nombre d'itérations dans un épisode décroît et il est plus petit quand on utilise une seule table de valeurs commune comme montré dans les figures. Ces courbes montrent également que l'algorithme avec une table commune prend moins d'épisodes pour converger que celui avec plusieurs tables privées. En effet, l'algorithme avec une table de valeurs commune réduit d'environ 50% ou plus le nombre d'épisodes pris par le robot. D'ailleurs, 6 épisodes au lieu de 12 suffisent au robot à bien se comporter dans le cas où trois SRAs naviguent dans un environnement donné et 4 épisodes au lieu de 56 dans le cas de plusieurs SRAs. Par conséquent, l'utilisation d'une table commune conduit à de meilleurs résultats.

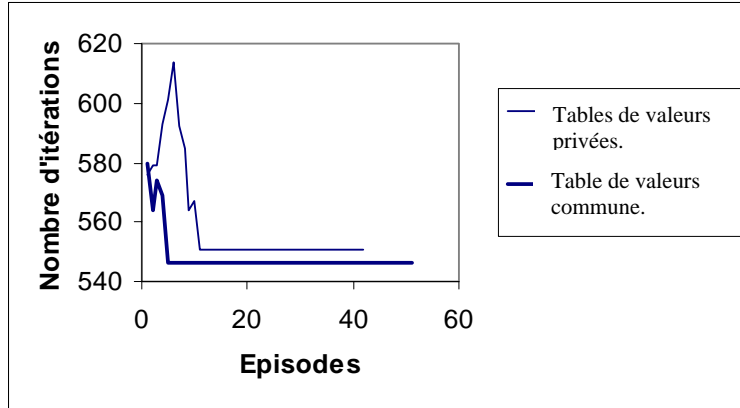


Figure V.13. Nombre d'itérations durant l'apprentissage pour trois SRAs.

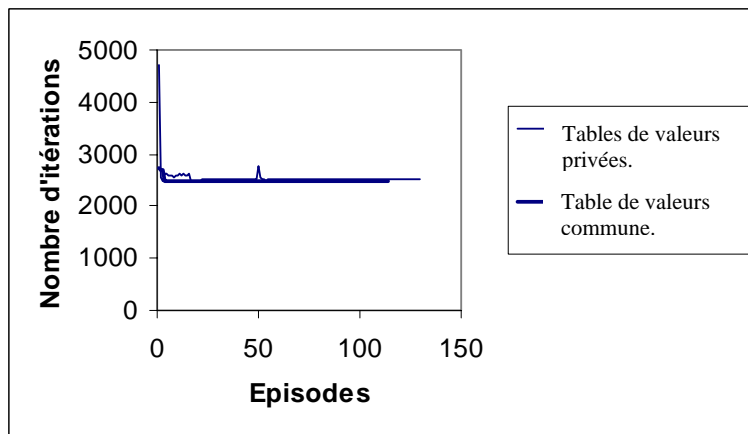


Figure V.14. Nombre d'itérations durant l'apprentissage pour plusieurs SRAs.

Ces résultats étaient prévisibles vu que les robots disposent de beaucoup plus d'informations sur l'environnement en un épisode lorsque la table de valeurs est commune.

Une fois les robots ont appris comment se comporter dans un environnement donné, ils devraient être capables d'évoluer avec succès dans d'autres environnements inconnus sans avoir à réapprendre de nouveau toutes les situations auxquelles ils sont confrontés dans ces nouveaux environnements. Par conséquent, ils sont amenés à utiliser les connaissances recueillies et apprises précédemment pour naviguer dans ces nouveaux environnements. Cette phase de test est abordée dans la section suivante.

#### V.4.2. Test de l'approche proposée dans des environnements inconnus

Afin de montrer que l'approche proposée donne de bons résultats dans l'acquisition du comportement d'évitement de collisions et d'attraction vers la cible, d'autres expériences ont été réalisées utilisant la table ou les tables de valeurs apprises dans les expériences précédentes. Ces SRAs ont à éviter les collisions entre eux et avec les obstacles bien que l'environnement a changé en :

- Enlevant un robot,
- Enlevant un obstacle,
- Utilisant un environnement complexe.

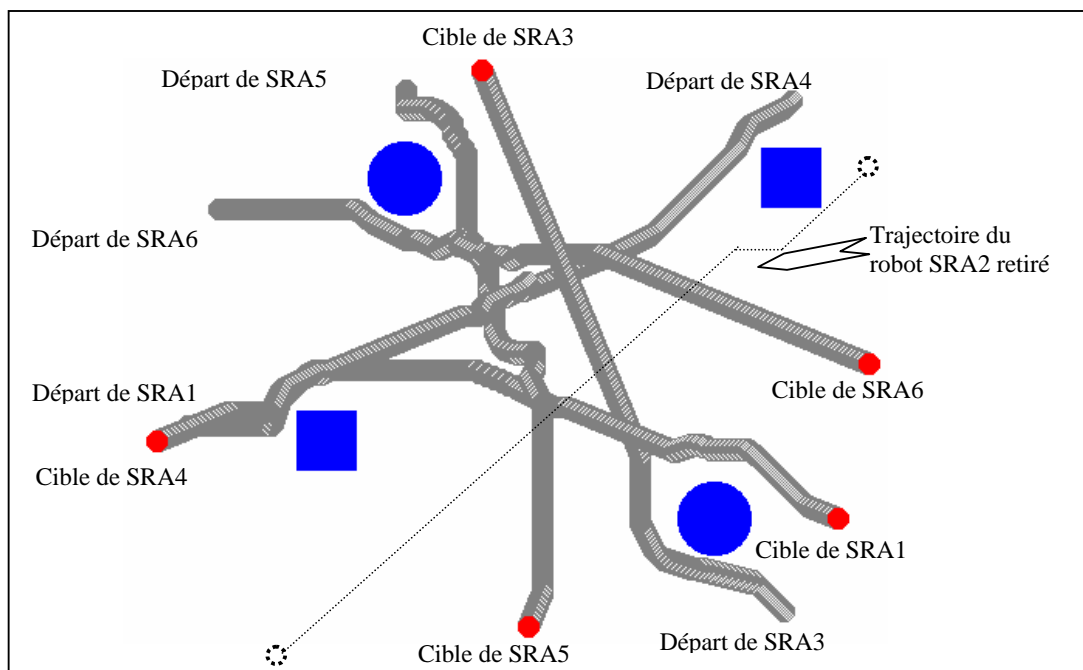


#### V.4.2.1. Retrait d'un robot ou d'un obstacle de l'environnement

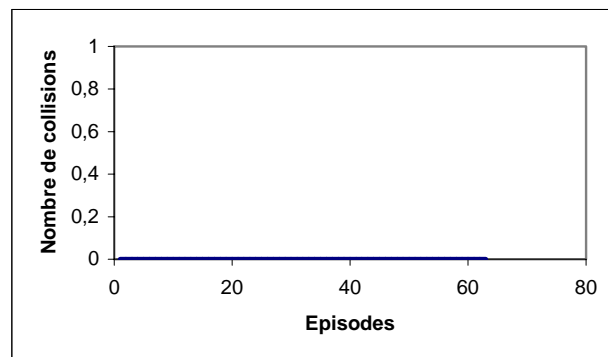
Dans ces expériences, un robot ou un obstacle est retiré de l'environnement d'apprentissage. Les résultats obtenus montrent que l'AR donne de bons résultats face à des situations non connues au préalable.

##### V.4.2.1.1. Cas de l'utilisation de tables de valeurs privées

Malgré le retrait d'un robot de l'environnement - le robot SRA2 (voir figure V.15(a)) - les autres robots réussissent à s'éviter et à éviter les obstacles tout en atteignant leurs cibles. Grâce à l'apprentissage réalisé et donc à l'utilisation des tables de valeurs privées, ces robots n'ont pas besoin d'un apprentissage supplémentaire pour naviguer avec succès dans un environnement différent de celui dans lequel l'apprentissage a été réalisé tel montré sur la figure V.15(b).



(a)

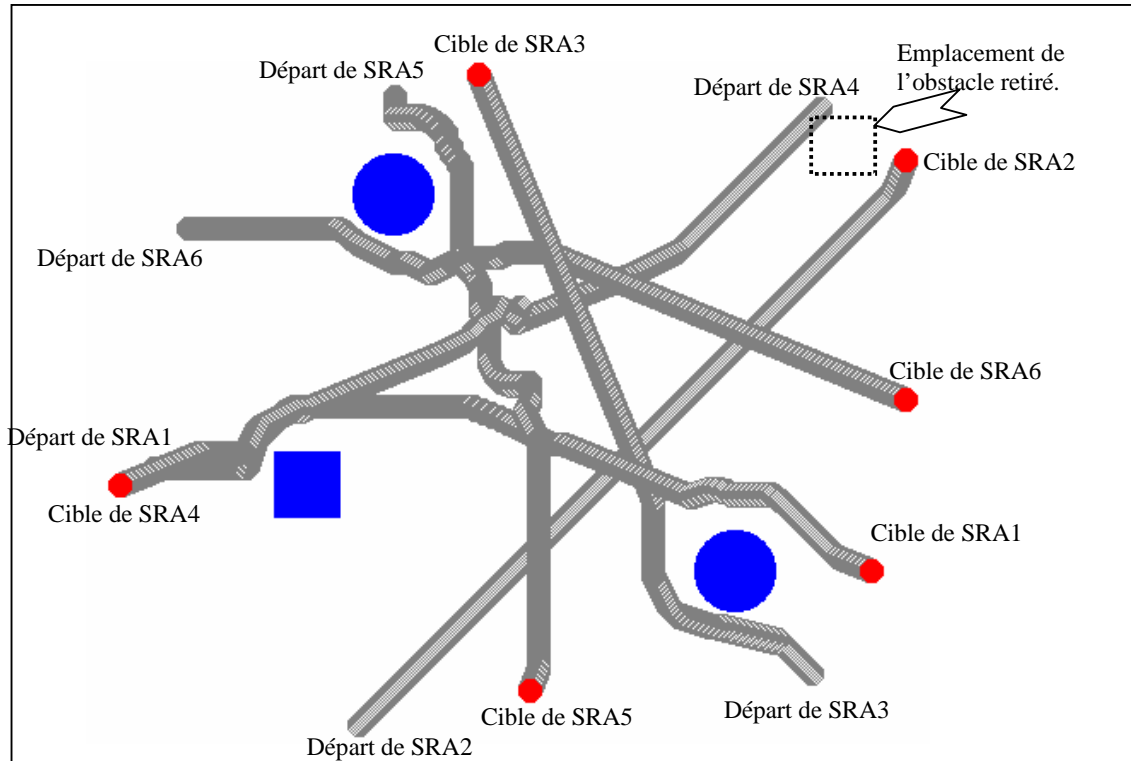


(b)

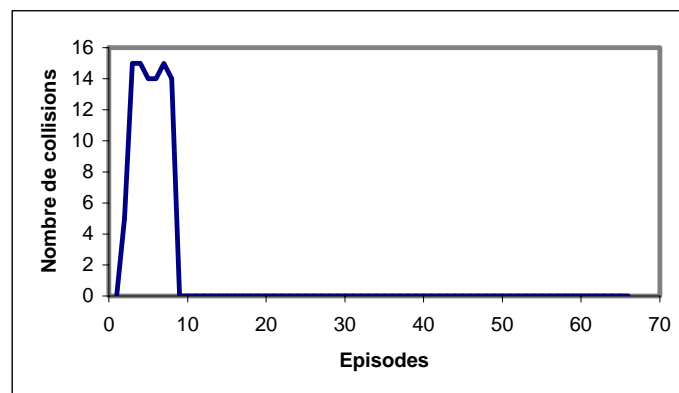
Figure V.15. Retrait d'un robot de l'environnement dans le cas de tables de valeurs privées.

Par contre lorsqu'un obstacle est retiré de l'environnement, comme illustré sur la figure V.16(a), les robots ont besoin d'un apprentissage supplémentaire pour atteindre leurs cibles

respectives sans collisions comme montré sur la figure V.16(b). Il a fallu 9 autres épisodes pour que ces robots réapprennent à se déplacer sans collisions entre eux et avec les obstacles. Ceci s'explique par le fait que les robots n'avaient pas toutes les informations concernant toutes les situations de l'environnement. Ces robots ne connaissant pas certaines situations où ils se sont retrouvés ont été obligés d'apprendre plus pour réussir dans leur tâche.



(a)

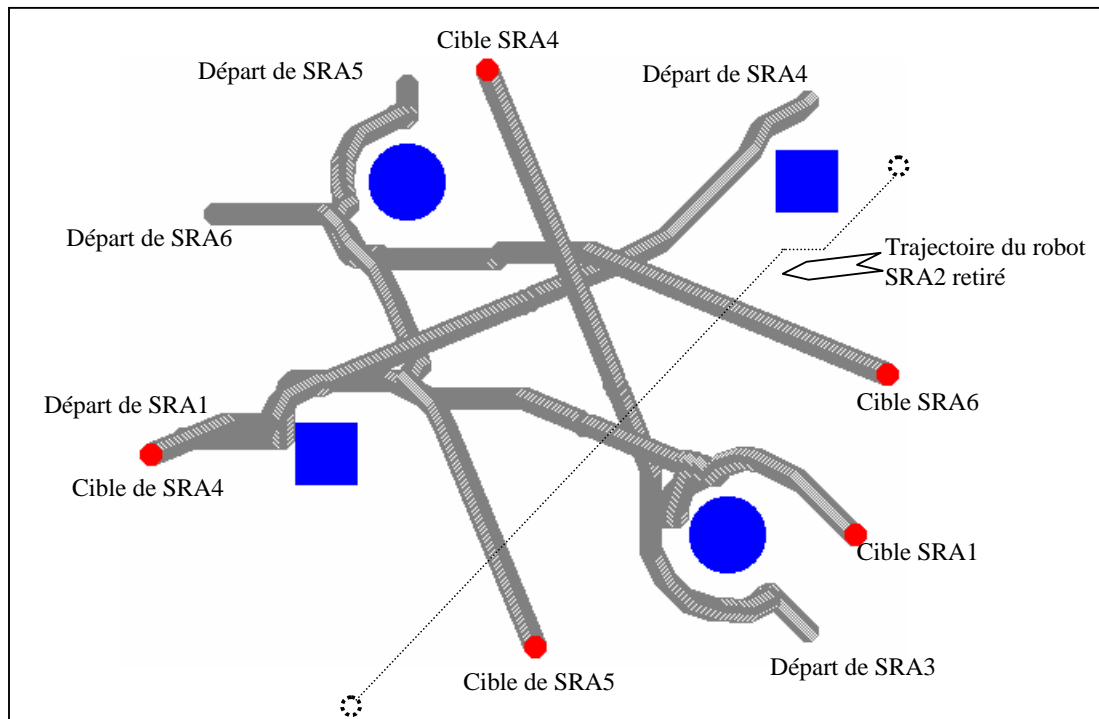


(b)

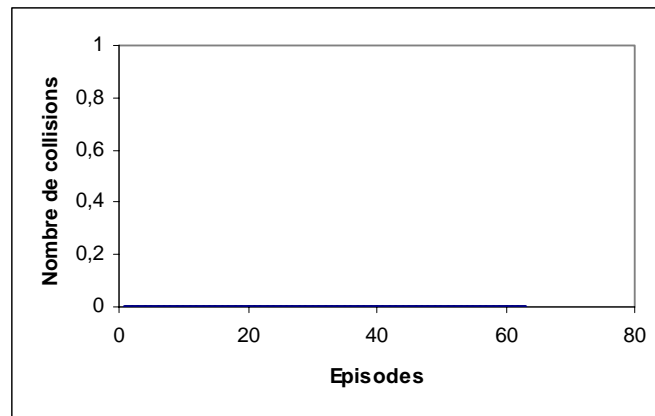
Figure V.16. Retrait d'un obstacle de l'environnement dans le cas de tables de valeurs privées.

#### V.4.2.1.2. Cas de l'utilisation d'une table de valeurs commune

Lorsqu'un robot est retiré de l'environnement (voir figure V.17(a)), les autres robots atteignent leurs cibles sans collisions comme illustré sur la figure V.17(a). La courbe de la figure V.17(b) confirme ce résultat vu qu'aucune collision n'a été notée durant le déplacement des différents robots.



(a)



(b)

Figure V.17. Retrait d'un robot de l'environnement dans le cas d'une table de valeurs commune.

Lorsqu'un obstacle est retiré, les différents robots réussissent malgré la modification de l'environnement à atteindre leurs cibles sans collisions, ce qui n'était pas le cas lorsque les tables de valeurs étaient différentes (voir figure V.18(a)). La figure V.18(b) illustre bien ce fait puisque aucune collision n'a été enregistrée.

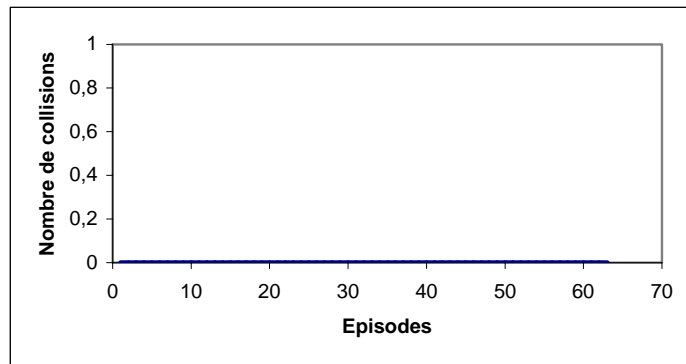
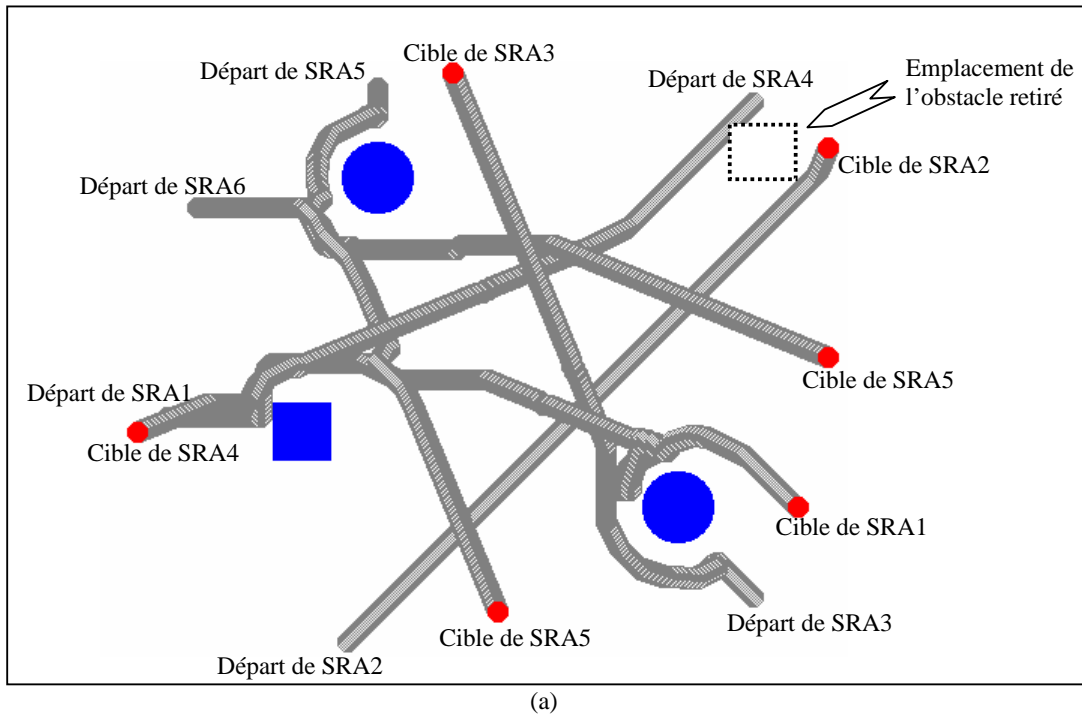


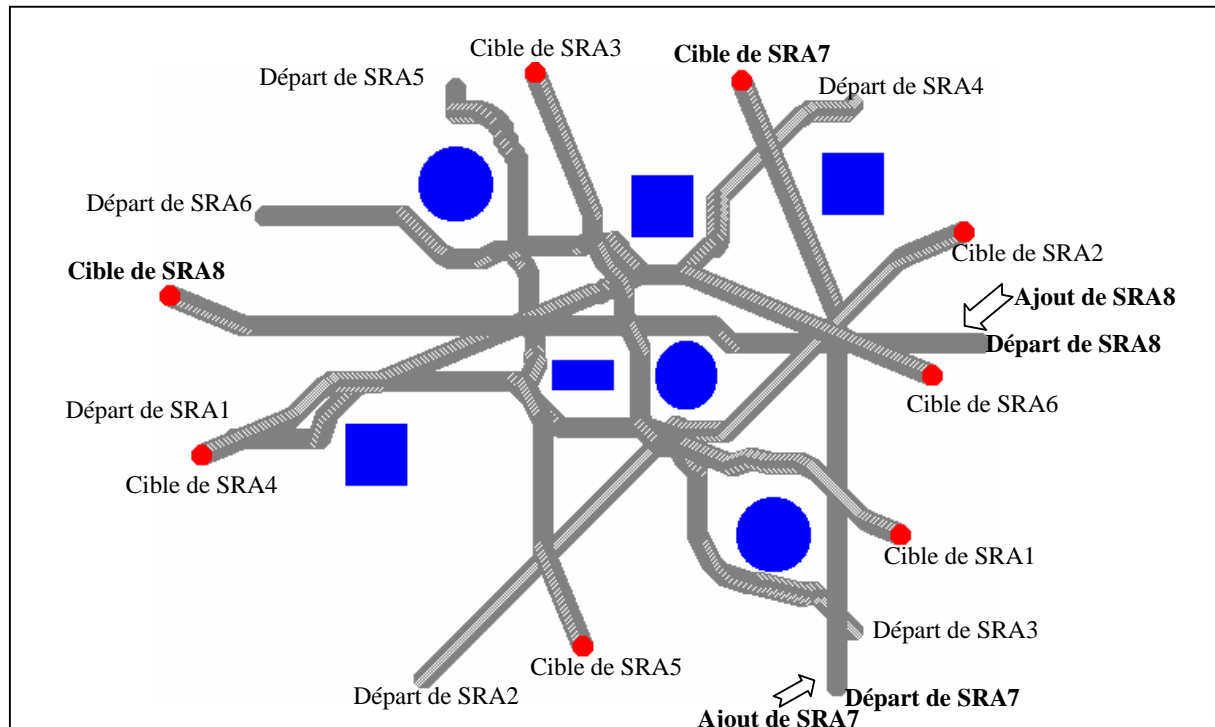
Figure V.18. Retrait d'un obstacle dans le cas d'une table de valeurs commune.

#### V.4.2.2. Environnement complexe

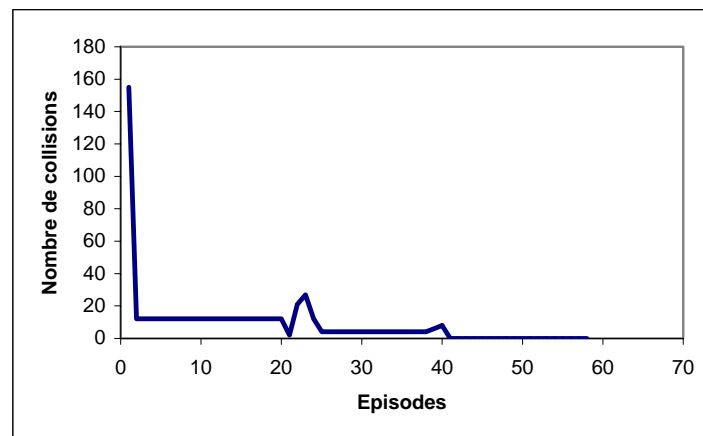
Dans un environnement complexe tel celui représenté sur la figure V.19(a), plusieurs robots ainsi que plusieurs obstacles ont été ajoutés. On rappelle qu'on utilise les tables de valeurs apprises dans les expériences du §.V.4.2.

##### V.4.2.2.1. Cas de l'utilisation de tables de valeurs privées

Dans cette expérience (figure V.19), comme on l'attendait d'après les résultats des expériences précédentes, les robots ne peuvent réaliser une bonne navigation sans un apprentissage supplémentaire. En effet, il a fallu une quarantaine d'épisodes de plus afin de cerner toutes les situations de l'environnement comme montré sur la figure V.19(b).



(a)

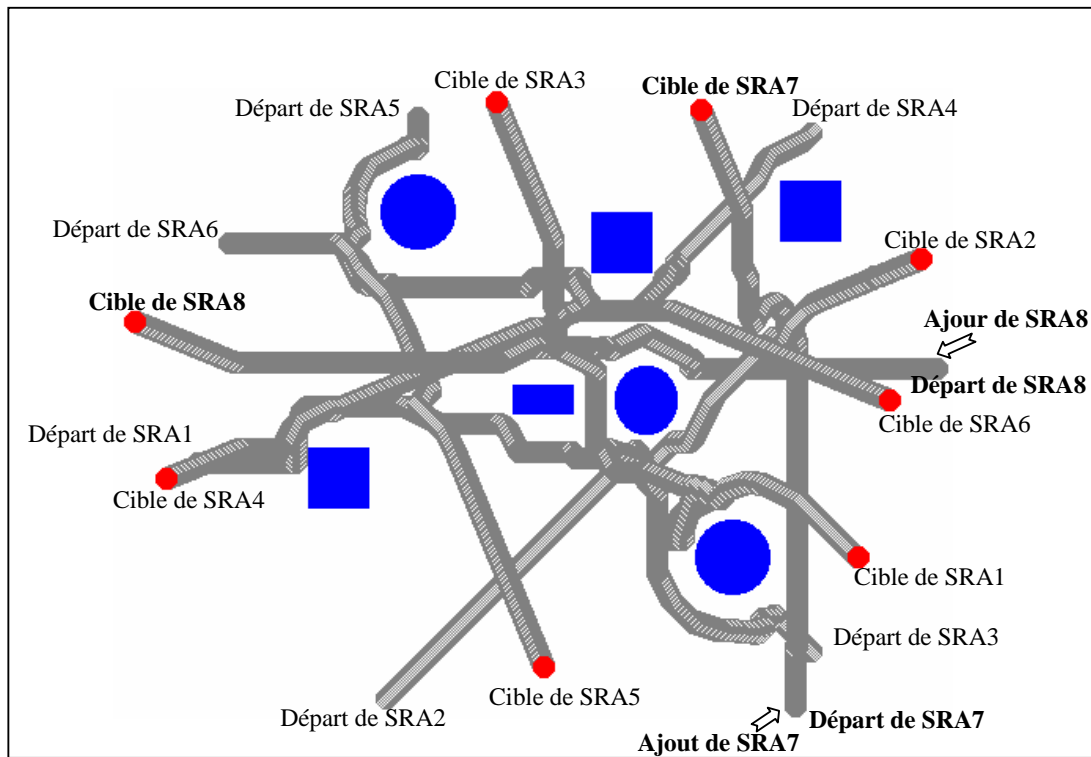


(b)

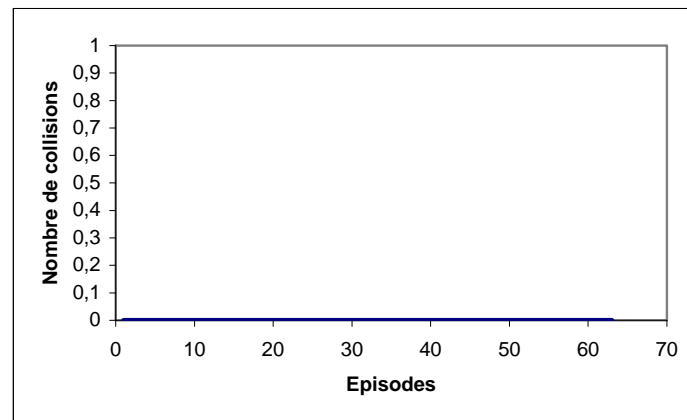
Figure V.19. Environnement complexe dans le cas de tables de valeurs privées.

#### V.4.2.2.2. Cas de l'utilisation d'une table de valeurs commune

La figure V.20 montre le même environnement complexe et les robots réussissent avec la même table de valeurs à atteindre leurs cibles sans collisions comme montré sur la figure V.20(b). La raison est que ces robots possédaient la connaissance nécessaire à leur évolution dans de tels environnements juste par le partage des différentes connaissances individuelles.



(a)



(b)

Figure V.20. Environnement complexe dans le cas d'une table de valeurs commune.

### V.4.3. Conclusion

Ces exemples illustrent la capacité de l'approche proposée avec la table de valeurs commune à acquérir le comportement d'évitement de collisions par les SRAs qui réussissent à bien se comporter dans de nouveaux environnements en utilisant seulement la table de valeurs apprise au préalable. Cependant, dans le cas où la table de valeurs est privée, les SRAs réussissent à éviter les collisions lorsqu'un robot est supprimé mais en supprimant un obstacle, les robots ont besoin de plus d'apprentissage afin d'acquérir le comportement. Ainsi, l'utilisation de la table de valeurs commune permet aux robots d'avoir plus d'informations et réagissent correctement face à de nouvelles situations, ce qui n'est pas le cas lorsque chaque robot possède sa propre table de valeurs.

Ces résultats de simulation montrent la capacité de l'approche proposée à doter chaque robot avec un comportement adaptatif acquis par interaction avec l'environnement afin de réaliser la tâche demandée avec une connaissance restreinte ou inexistante à priori.

Après avoir fait apprendre aux robots comment atteindre leurs cibles respectives sans collisions en acquérant le comportement d'évitement de collisions et d'attraction vers la cible, ces robots sont amenés à exécuter des tâches plus ou moins complexes. La tâche qu'on demande aux robots de réaliser dans ce présent travail est le transport d'un objet lourd et/ou encombrant. Ils doivent donc utiliser les stratégies de formation de figures géométriques et de déplacement en formation développées au chapitre IV. Les sections suivantes abordent cette thématique.

## **V. 5. Comportements intelligents d'un groupe de systèmes robotiques autonomes : vers une navigation en formation**

Les stratégies de formation et de déplacement en formation présentées au chapitre IV sont simulées afin d'une part former différentes figures géométriques et d'autre part se déplacer en une formation définie dans un environnement non connu à priori.

Chaque robot dispose de la distance le séparant des obstacles qui est mesurée par les capteurs IR dont il est doté (voir chapitre II). Ces derniers fournissent donc un vecteur  $s$  à 8-dimensions. Pour une situation  $s$ , le robot se déplace en appliquant les différents comportements définis au chapitre IV (§.IV.3.2.1.1 et §.IV.3.2.2.1). Après plusieurs épisodes correspondant à la formation d'une figure donnée et le déplacement de ce groupe du point de départ au point d'arrivée, les robots réussissent à former des figures géométriques et à atteindre leur cible sans collisions entre eux et avec les obstacles de l'environnement. Chaque robot a appris quand il doit éviter les collisions. En effet, une politique efficace a été trouvée et la performance pour former et maintenir la figure a été atteinte. Dans ce cas, la table Q ne change plus et ainsi il n'y a plus d'amélioration du processus de navigation. Cet apprentissage permet au robot de devenir plus autonome car il s'adapte aux conditions nouvelles ou changeantes survenant dans l'environnement.

### **V.5.1. Stratégie de formation de figures géométriques par les SRAs**

Plusieurs figures ont été considérées à savoir :

- Le cercle,
- Le cercle plein,
- Le polygone,
- La ligne.

#### **V.5.1.1. Formation d'un cercle**

Afin de former un cercle, un ensemble de robots va utiliser trois (03) comportements élémentaires (voir chapitre IV, §.IV.3.2.1.1) pour y réussir et qui sont :

- Regroupement,
- Séparation,
- Evitement de collisions.

Soit  $N= 11$  le nombre de robots. Ces onze robots vont former approximativement un cercle en utilisant la stratégie de formation de figures géométriques proposée au chapitre IV.

Rappelons que les actions qui découlent de ces comportements pour la formation d'un cercle sont au nombre de six (06) et qui sont :

- *TournerG*,
- *TournerD*,
- *RApprocher*,
- *ELoigner1*,
- *ELoigner2*,
- *Stopper*.

#### V.5.1.1.1. Apprentissage de la formation d'un cercle

Afin d'apprendre aux robots à faire un cercle, les paramètres du tableau V.3 sont utilisés.

Tableau V.3. Paramètres de la formation en cercle.

Paramètres	Valeur
$\xi$	0.3
Diamètre du cercle ( $D[m]$ )	8
Taux d'apprentissage ( $\alpha$ )	0.5
Facteur d'escompte ( $\gamma$ )	0.5
Renforcement cercle formé	+5
Renforcement si s'approcher	+4
Renforcement si s'éloigner	-5
Renforcement si collision	-5
Taux d'exploration	5%

En utilisant le comportement de regroupement lorsque les robots s'éloignent trop, le comportement de séparation lorsqu'ils sont trop proches et le comportement d'évitement de collisions pour s'éviter mutuellement, ces robots arrivent à apprendre à former le cercle tel montré sur la figure V.21. Cette figure montre les différents épisodes où les robots essaient d'apprendre à former le cercle en ayant un minimum d'information à leurs dispositions (les distances les séparant). Comme c'est montré (de l'épisode 1 à l'épisode 3), les SRAs n'ont pas appris la figure demandée (i.e., formation du cercle) puisque plusieurs collisions sont survenues dans ces épisodes comme montré sur la figure V.22 (par exemple, 25 dans le premier épisode et 5 dans le deuxième) et le cercle n'est pas bien approximé (voir figure V.21). Cependant, à partir de l'épisode 7, ils ont acquis le bon comportement puisqu'ils ont appris à former le cercle. Rappelons que ces robots utilisent la même table de valeurs qu'ils mettent à jour au fur et à mesure de l'évolution de la formation.



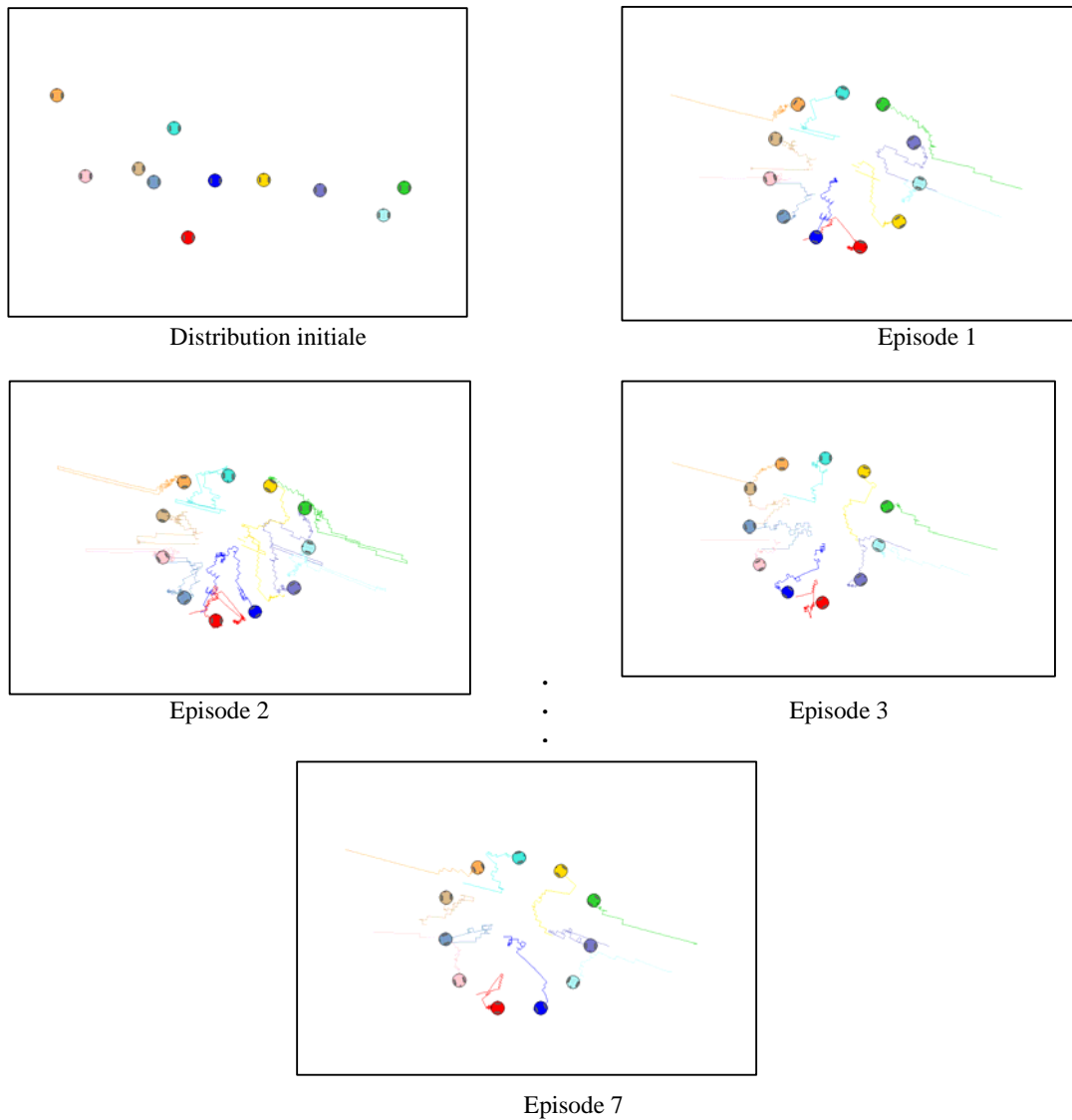


Figure V.21. Etapes de formation d'un cercle.

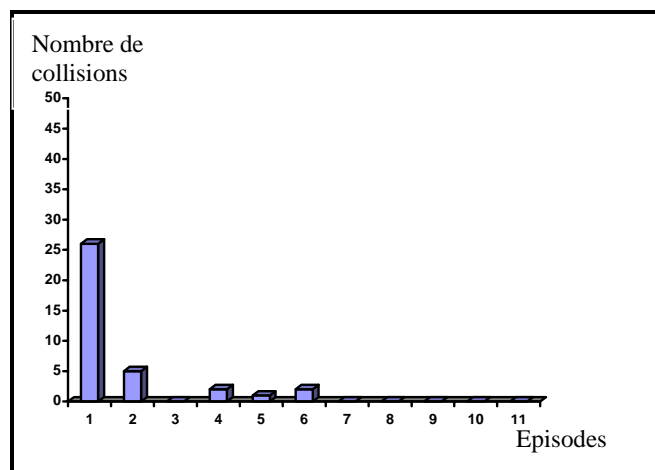


Figure V.22. Graphe de performance de la formation en cercle.

Dans cet exemple, les robots arrivent sans entrer en collision à former un cercle à partir du septième épisode. Après 7 épisodes sans collisions, la connaissance des robots est sauvegardée pour être réutilisée dans un environnement différent.

### V.5.1.1.2. Apprentissage de la formation d'un cercle dans un environnement complexe

#### a) Environnement parsemé d'obstacles

Afin de vérifier la validité de l'approche proposée, on considère un environnement parsemé d'obstacles et où les robots doivent réussir à éviter les obstacles et à former le cercle demandé. On remarque qu'effectivement, les robots dans les premiers épisodes (voir Figure V.24) entrent en collisions avec les obstacles et entre eux mais après plusieurs épisodes, ils ont appris à éviter les obstacles et à réaliser la figure comme montré sur la figure V.23.

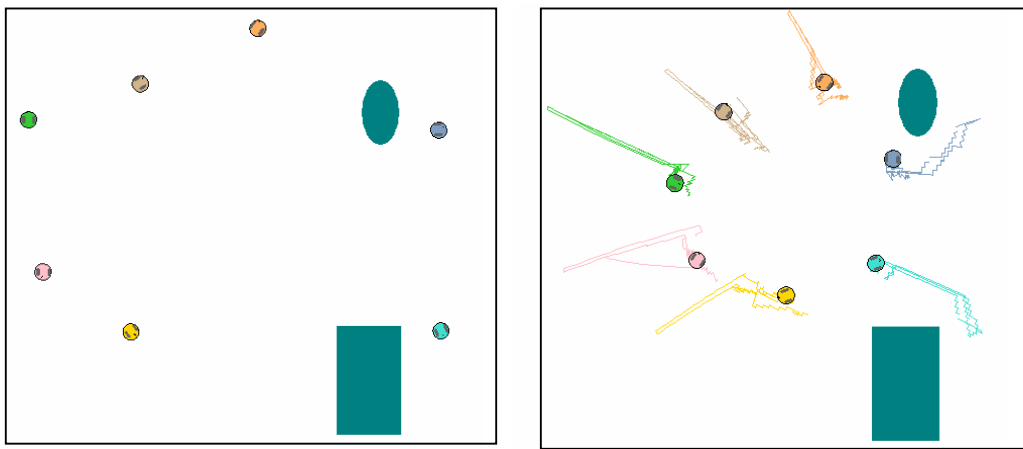


Figure V.23. Apprentissage de la formation d'un cercle dans un environnement parsemé d'obstacles.

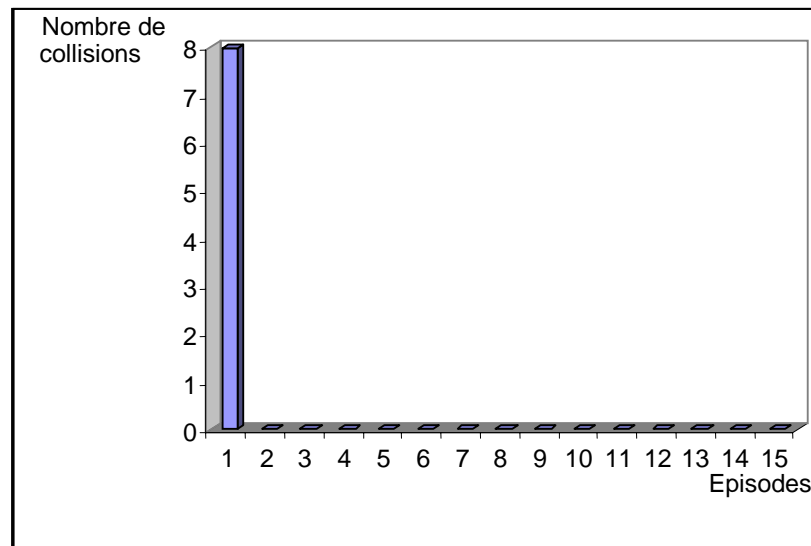


Figure V.24. Performance de la formation d'un cercle dans un environnement parsemé d'obstacles.

*b) Formation d'un cercle autour d'un objet rond*

Dans cette section, on montre l'apprentissage de la formation d'un cercle autour d'un objet comme illustré sur la figure V.25. En effet, les robots apprennent à se positionner autour de l'objet à transporter après 6 épisodes comme montré sur la figure V.26.

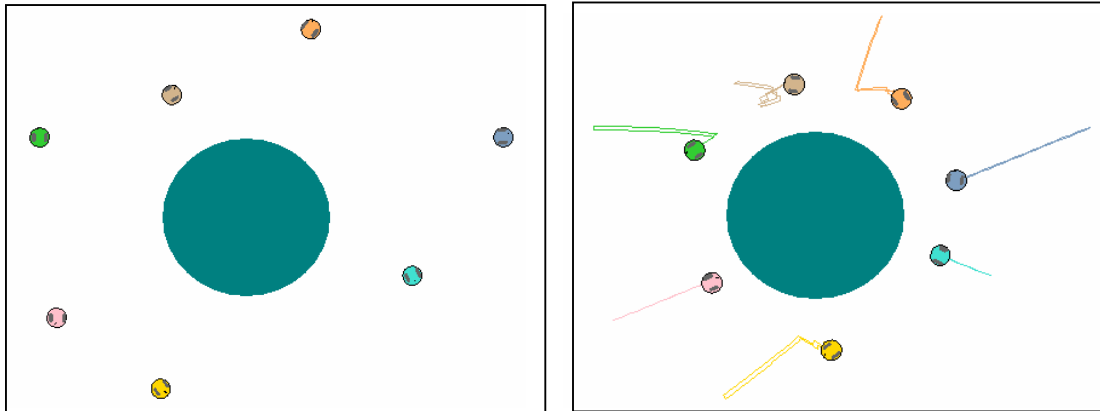


Figure V.25. Apprentissage de la formation d'un cercle autour d'un objet rond.

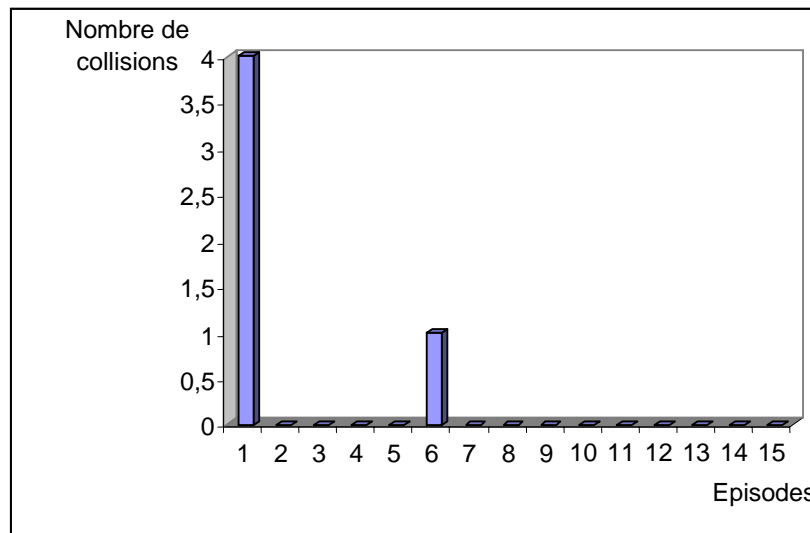
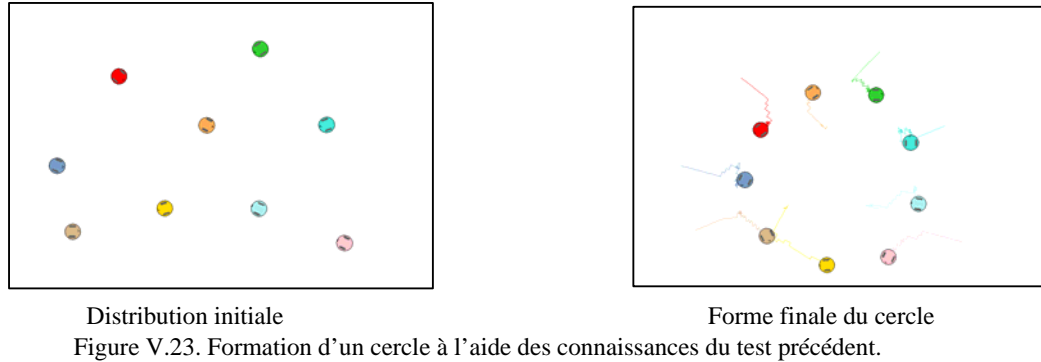


Figure V.26. Graphe de performance de l'apprentissage de la formation d'un cercle autour d'un objet.

**V.5.1.1.2. Test et performance de l'apprentissage**

Pour tester la performance de cette approche, deux robots ont été retirés du groupe de base et une autre distribution initiale est utilisée comme illustré sur la figure V.23. Les robots arrivent à former le cercle en utilisant la connaissance contenue dans la table des valeurs acquise lors de l'apprentissage précédent puisqu'en un seul épisode le cercle est obtenu.



### V.5.1.2. Formation d'autres figures

#### V.5.1.2.1. Formation d'un polygone simple

Comme on l'a présenté au chapitre IV, la formation d'un polygone nécessite d'abord la formation d'un cercle, ensuite on fait le choix du nombre de sommets ainsi que leurs positions et on exécute la procédure définie au chapitre IV.

La figure V.24 montre la simulation d'un groupe de 15 SRAs réalisant un polygone.

Après la formation du cercle (Figure V.24(a)), chaque robot  $R_i$  reconnaît son voisin de gauche  $R_{gauche}$  et celui de droite  $R_{droite}$ .

On a choisi cinq sommets pour cette expérience. Ces sommets sont déplacés manuellement vers leurs positions finales (Figure V.24(b)). La Figure V.24(c), montre le déplacement des robots vers leurs positions finales et la formation du polygone.

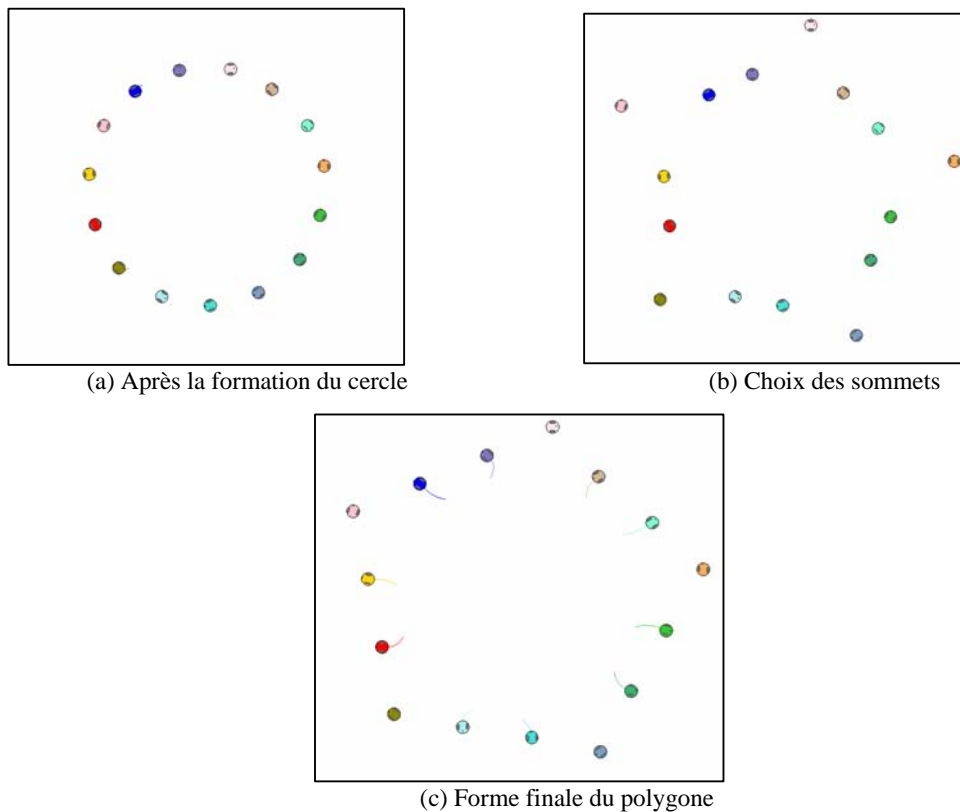
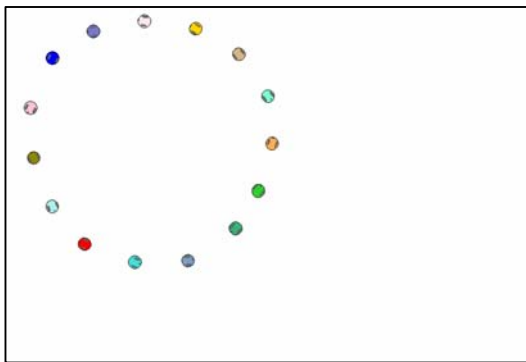


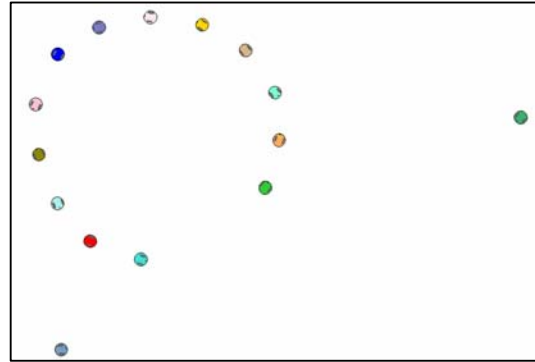
Figure V.24. Étapes d'exécution pour la formation d'un polygone.

### V.5.1.2.2. Formation d'une ligne

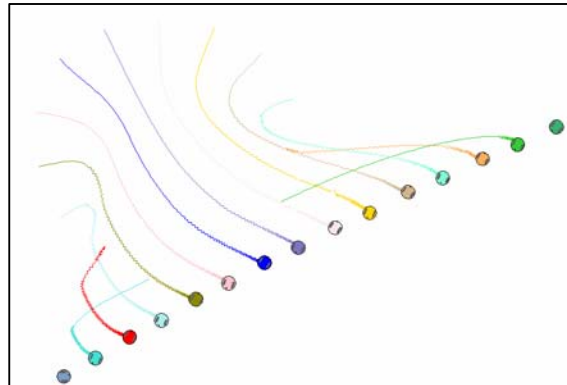
La formation d'une ligne est un cas particulier d'un polygone avec seulement deux sommets. La formation d'une ligne nécessite la formation d'un cercle comme illustré à la figure V.25(a). Deux robots sont ensuite sélectionnés qui seront les deux extrémités de la ligne à former (voir figure V.25(b)) et qui doivent être voisins dans le cercle formé. Ces robots sont déplacés manuellement vers leurs positions finales et ensuite les autres robots vont se déplacer pour atteindre leurs positions finales afin de former la ligne comme montré sur la figure V.25(c).



(a) Après la formation du cercle



(b) Choix des sommets



(c) Forme finale de la ligne

Figure V.25. Etapes de formation de la ligne.

### V.5.2. Formation d'un cercle plein

Soit  $N=10$  le nombre de robots considérés. On a alors dix robots qui vont former une approximation d'un cercle plein en utilisant l'approche proposée au chapitre IV.

Les mêmes comportements élémentaires sont utilisés afin de réaliser la formation d'un cercle plein. Ces comportements conduisent à l'utilisation des actions suivantes :

- *TournerG*,
- *TournerD*,
- *RApprocher*,
- *ELoigner2*,

- Stopper.

### V.5.2.1. Apprentissage de la formation d'un cercle plein

Afin d'obtenir la formation d'un cercle plein, les paramètres de l'apprentissage sont donnés dans le tableau V.4. Les mêmes comportements sont utilisés à l'exception du fait que le comportement de séparation se restreint au deuxième cas présenté dans le §.IV.3.2.2.1.

Tableau V.4. Paramètres de la formation en cercle plein.

Paramètres	Valeur
Diamètre du cercle ( $D$ [m])	6
Taux d'apprentissage ( $\alpha$ )	0.5
Facteur d'escompte ( $\gamma$ )	0.5
Renforcement cercle plein formé	+5
Renforcement si s'approcher	+4
Renforcement si s'éloigner	-5
Renforcement si collision	-5
Taux d'exploration	5%

Ces robots arrivent à apprendre à former le cercle plein comme illustré sur la figure V.26. En effet, les robots réussissent à apprendre la formation en question en quatre épisodes (voir figure V.27). Le nombre de collisions de cet épisode est donc nul impliquant le fait que les robots ont appris la figure comme illustré sur la figure V.27.

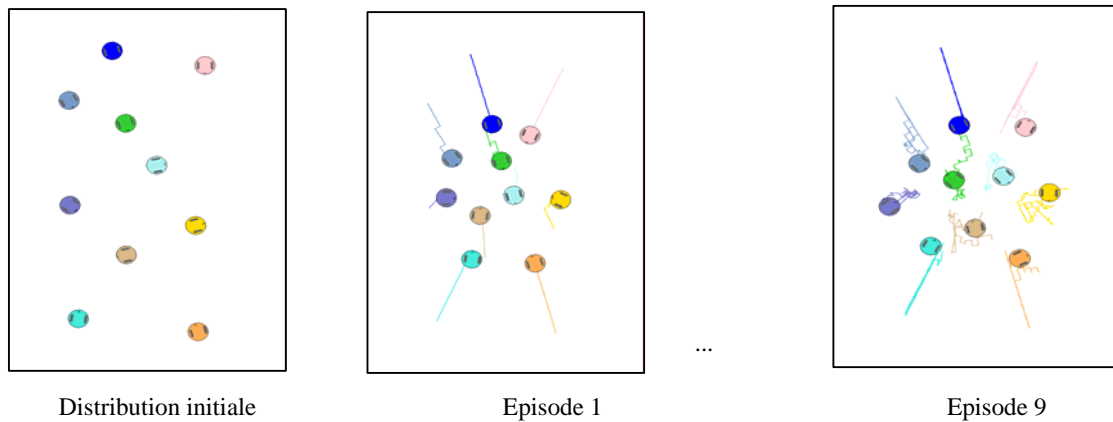


Figure V.26. Étapes de formation d'un cercle plein.

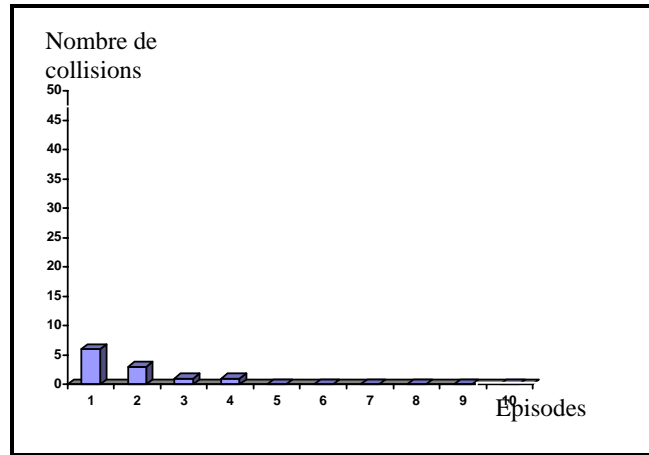


Figure V.27. Graphe de performance pour la formation d'un cercle plein.

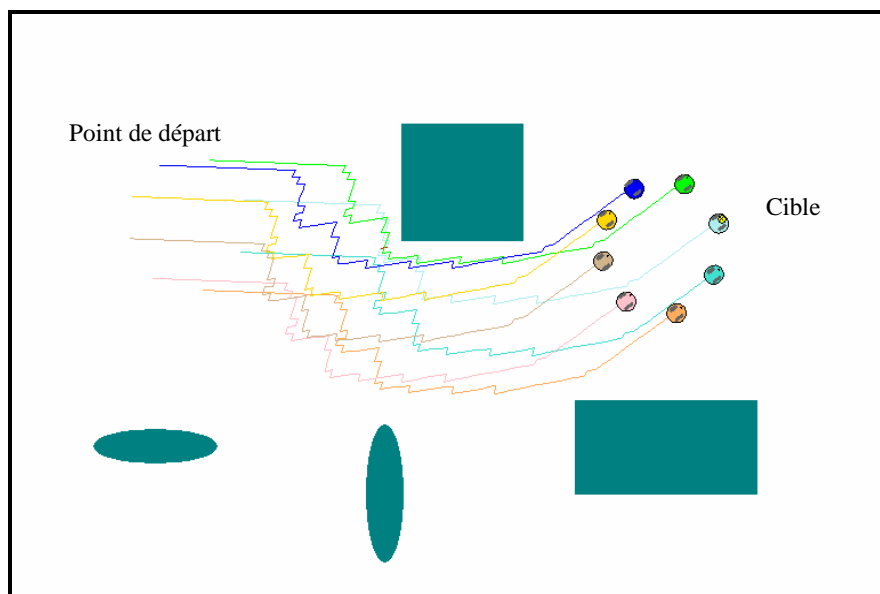
Ces exemples montrent que l'approche proposée permet d'apprendre différentes figures géométriques en utilisant des comportements élémentaires. En d'autres termes, grâce à ces comportements, un comportement complexe a émergé des interactions des SRAs avec l'environnement. La connaissance collectée est alors stockée dans la table de valeurs.

### V.5.2. Stratégie de déplacement en formation d'un groupe de SRAs

Afin de se déplacer en gardant la formation réalisée, les robots utilisent la technique du "meneur-suiveurs" qui permet à un robot de guider l'ensemble vers le but fixé.

#### V.5.2.1. Navigation de la formation en cercle

La figure V.28 montre la simulation d'un groupe de 8 SRAs formant un cercle et se déplaçant dans un environnement inconnu en exploitant la stratégie de déplacement en formation proposée au chapitre IV. Le choix du meneur est fait à chaque itération. A partir du troisième épisode comme le montre la figure V.29, le groupe de robots arrive à atteindre sa cible sans entrer en collision avec les obstacles.

Figure V.28. Navigation d'un groupe de robots mobiles en forme de *cercle*.

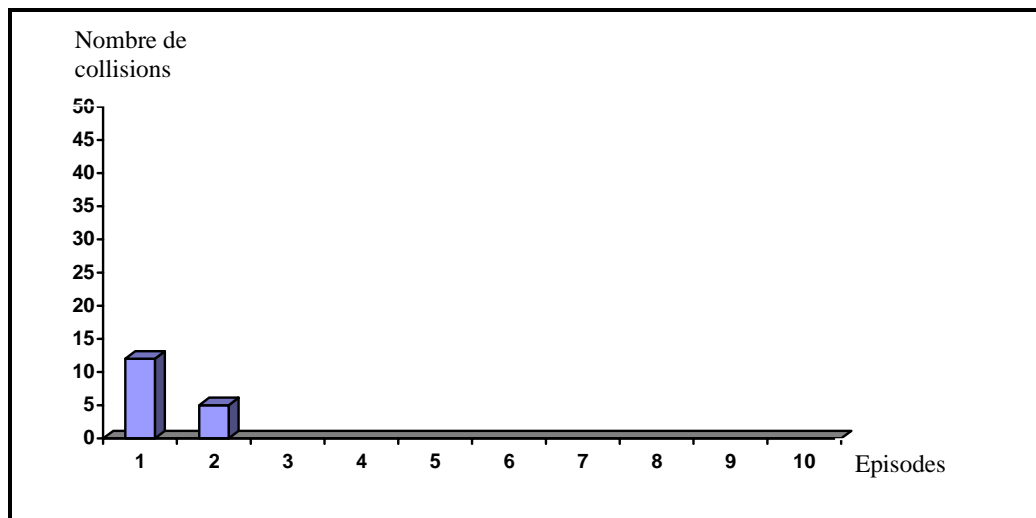


Figure V.29. Graphe de performance de la formation en cercle.

### V.5.2.2. Navigation de la formation en cercle plein

Dans cet exemple, on prend un groupe de 10 SRAs en forme de cercle plein. Cette formation se déplace dans l'environnement de la figure V.30. Ces robots apprennent à éviter les obstacles tout en gardant la formation après le quatrième épisode comme le montre le graphe de performance de la figure V.31.

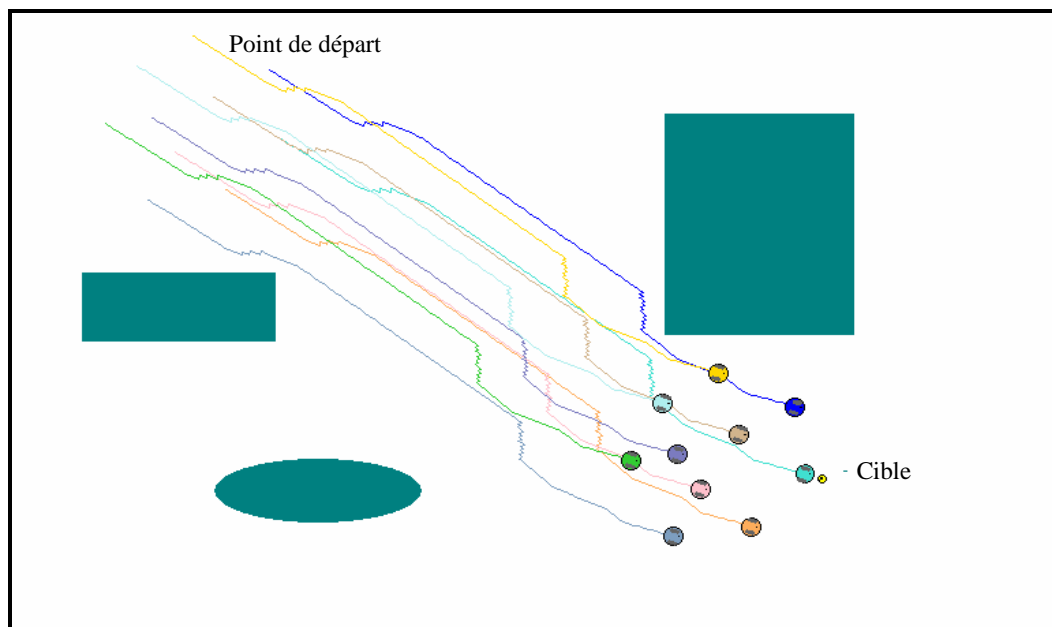


Figure V.30. Navigation d'un groupe de robots mobiles en forme de cercle plein.



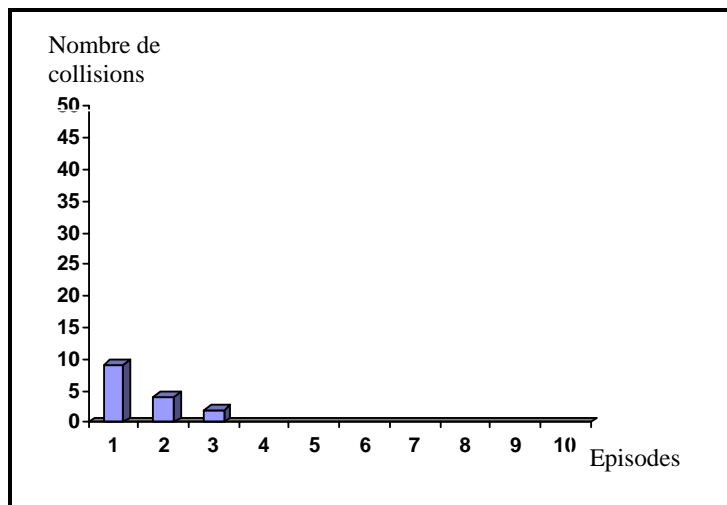


Figure V.31. Graphe de performance de la formation en cercle plein.

Afin de montrer l'efficacité de l'approche proposée, l'environnement de la figure V.28 est utilisé mais la une différence est que l'espace entre les obstacles est plus petit que le diamètre du cercle formé par les robots. Durant les premiers épisodes, les robots rentrent en collision avec les obstacles en essayant d'emprunter le même chemin que celui de la figure V.28 mais par essai et erreur et particulièrement grâce à la capacité d'exploration de l'AR, le groupe trouve un nouveau chemin afin d'atteindre sa cible sans collisions comme montré sur la figure V.32.

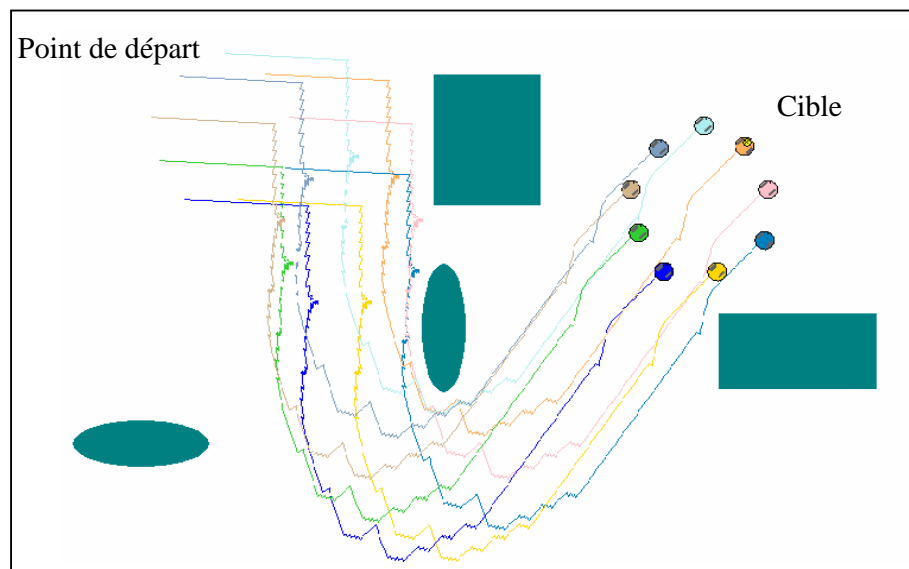


Figure V.32. Navigation en groupe avec un espace faible entre obstacles.

### V.5.3. Conclusion

Afin de résoudre le problème de la formation et du déplacement en formation des SRAs dans un environnement donné, une approche par renforcement est proposée. Dans ce type d'environnement, il est difficile de préparer un signal d'entraînement ou de collecter les

échantillons nécessaires pour l'entraînement. En effet, l'AR permet au moins en principe, de surpasser les problèmes de construction d'un modèle explicite du comportement à synthétiser ou le besoin d'avoir une base d'apprentissage significative pour l'apprentissage supervisé. Avec cet apprentissage adaptatif, le robot est guidé uniquement par des renforcements retournés par l'environnement et est réalisé itérativement et progressivement puisque ses paramètres sont mis à jour à chaque itération et donc sont sensibles à tous les changements de l'environnement particulièrement à la table de valeurs. Cette table, qui donne une évaluation du comportement sélectionné, permet un apprentissage par interaction avec l'environnement.

Les SRAs dotés de cette approche réussissent à réaliser la tâche demandée sans collisions dans un environnement à priori inconnu. Ainsi, cette approche permet une navigation en temps réel basée sur un apprentissage continu dans un environnement dynamique. Cette approche basée sur l'algorithme d'apprentissage Q-learning permet d'apprendre par interaction comment former une figure géométrique et comment éviter les autres SRAs ou des obstacles même en présence d'incertitudes. En effet, dans les applications réelles, il existe des bruits aléatoires imprédictibles et les réponses aux problèmes de la navigation en groupe ne peuvent pas être obtenues facilement par l'être humain. Il est alors nécessaire que les solutions à ces problèmes puissent être acquises d'une manière automatique à travers le processus d'apprentissage dans un monde réel ou simulé (à partir de l'expérience).

Afin de mettre en œuvre cette approche et les approches des §.V.3 et §.V.4, un contrôleur flou a été utilisé qui permet l'exécution des actions générées par ces approches.

## **V.6. Contrôleur flou de mouvements : Exécution des actions générées par les approches de navigation proposées**

Pour exécuter l'action fournie par les approches de planification et de contrôle proposées, on a utilisé un contrôle hybride utilisant le concept du contrôle temps minimum dit "bang bang" et la théorie de la logique floue [48]. Ce système intelligent permet aux robots d'exécuter l'action décidée en un temps minimal lui fournissant ainsi la capacité de naviguer intelligemment dans un environnement dynamique. En d'autres termes, ce contrôleur va donner la commande appropriée aux moteurs du robot exécutant cette action.

### **V.6.1. Description du contrôleur**

Le robot doit apprendre à exécuter l'action appropriée  $A_i$  résultat des approches de navigation proposées dans les chapitres II, III et IV. Ce contrôleur doit classifier les commandes d'action  $CA_k$ . Cette classification se base sur une stratégie de contrôle temps minimum et est utilisée au niveau du contrôle le plus bas afin de déduire les actions de contrôle appropriées à la locomotion du robot. En fait, l'espace d'état du robot (distance et vitesse) est modélisé en intervalles discrets aboutissant à des variables linguistiques.

Après fuzzification des variables d'entrée, à savoir la vitesse et la distance, l'inférence floue permet d'obtenir la commande d'action adéquate qui est défuzzifiée. Celle-ci est ensuite appliquée aux actionneurs des robots afin d'obtenir des mouvements lisses.

La fuzzification des variables d'entrée permet d'obtenir les règles floues de commande qui permettent de contrôler le mouvement du robot et de réaliser la commande en un temps minimal. Les fonctions d'appartenance correspondant à ces variables sont données en Annexe A4. Afin d'obtenir ces règles floues, des expériences ont été effectuées et ont donné les valeurs des distances d'accélération/décélération, pour chaque combinaison de la vitesse

finale, la vitesse initiale, et la distance-but (Annexe A5). A partir de ces expériences, les règles floues de contrôle ont été déduites et utilisées dans un mécanisme d'inférence floue afin d'obtenir la commande finale de contrôle aux actionneurs du robot. L'inférence floue est réalisée par un algorithme de recherche et d'appariement, utilisant les opérateurs MIN-MAX, développé dans [51]. Enfin, l'opération MAX est utilisée pour la défuzzification afin de donner la commande finale. Le détail de ce contrôleur est donné en Annexe 6.

Cette stratégie de contrôle est choisie compte tenu du fait que le problème considéré est d'atteindre un sous-but en un temps minimum avec une certaine précision [34].

Les données d'entrées (vitesse finale désirée, vitesse initiale, distance but) du contrôleur sont fournies à un haut niveau de traitement (voir figure V.33) à partir du système contrôlant les comportements intelligents de navigation.

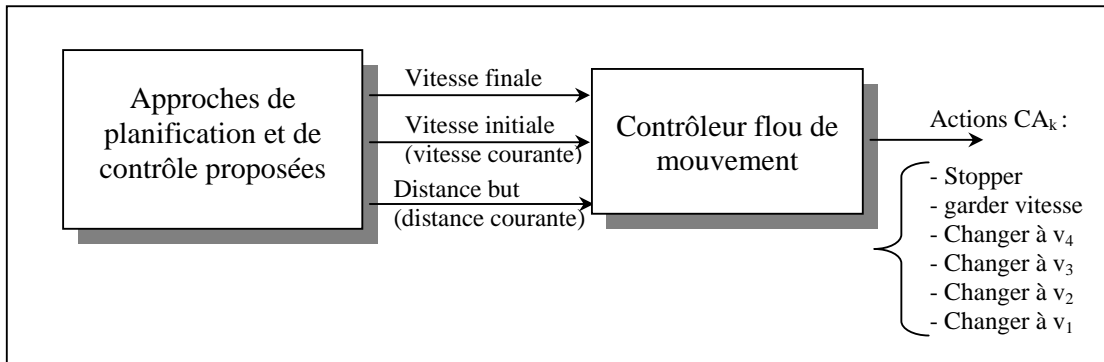


Figure V.33. Données d'entrée et de sortie du contrôleur flou.

La commande d'action  $CA_k$  ( $k=1, \dots, 6$ ) doit être choisie parmi les commandes de base suivantes afin d'entraîner les actionneurs du robot (voir figure V.33) :

1.  $CA_1 = \{\text{changer de vitesse à } v_0, \text{ i.e., stopper l'actionneur}\}$ ,
2.  $CA_2 = \{\text{garder la même vitesse}\}$ ,
3.  $CA_3 = \{\text{changer la vitesse à } v_4\}$ ,
4.  $CA_4 = \{\text{changer la vitesse à } v_3\}$ ,
5.  $CA_5 = \{\text{changer la vitesse à } v_2\}$  et
6.  $CA_6 = \{\text{changer la vitesse à } v_1\}$ .

où  $v_n$  ( $n=0, \dots, 4$ ) sont les vitesses finales désirées.

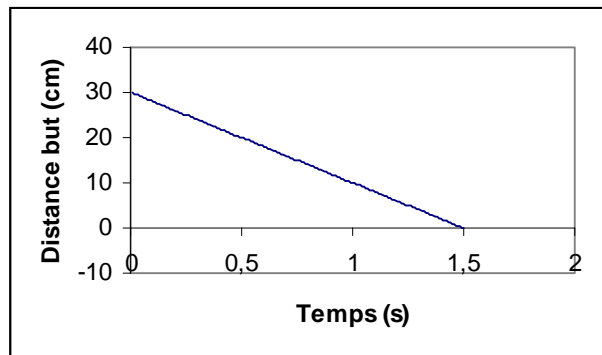
### V.6.2. Utilisation du contrôleur flou

Les principales caractéristiques de ce contrôleur de mouvement sont:

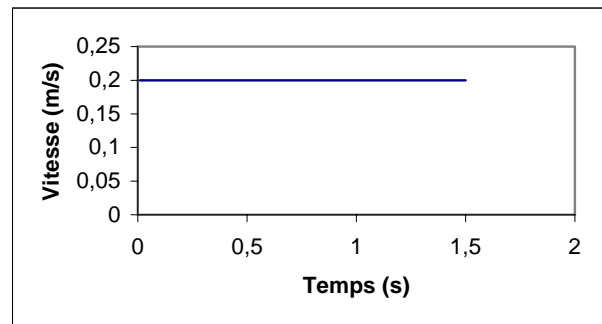
- fonctions d'appartenance trapézoïdales pour la fuzzification,
- trois cents (300) règles floues constituant la base de règles floues,
- opérateurs MIN-MAX pour l'inférence floue,
- opérateur MAX pour la défuzzification.

Les profils de la distance et de la vitesse montrés dans les figures V.34, V.35 et V.36 sont donnés pour illustrer les résultats de simulation du contrôleur développé. La distance but considérée est la distance de détection des obstacles qui est égale à 30 cm et la vitesse maximale utilisée est de 20 cm/s permettant aux robots d'exécuter l'action appropriée [5, 6].

Sur la figure V.34, la distance but de 30 cm est parcourue, à la vitesse de 20 cm/s, en un temps égal à 1.49 s comme montré sur la figure V.34(a). Le robot a donc le temps de réaliser l'action qui lui est demandée bien qu'il avance à la vitesse maximale permise (voir figure V.34(b)).



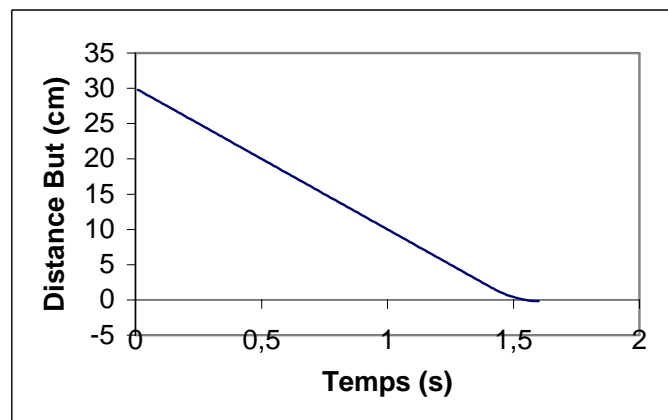
(a)



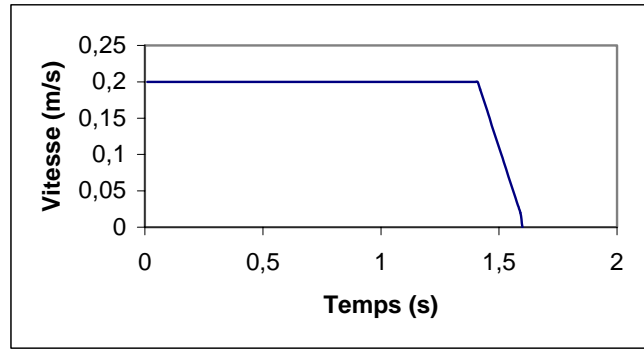
(b)

Figure V.34. Profiles de la distance et de la vitesse pour  $V_{\text{initial}}=0.2$  m/s,  $V_{\text{final}}=0.2$  m/set  $d_{\text{but}}=30$ cm.

Sur la figure V.35(a), la distance but de 30 cm est parcourue en 1.53 s avec une vitesse initiale maximale de 0.2 m/s et diminue pour atteindre la vitesse finale de 0 m/s (voir figure V.35(b)).



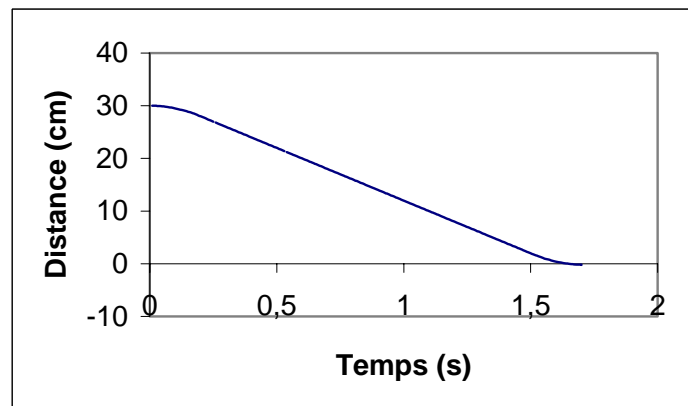
(a)



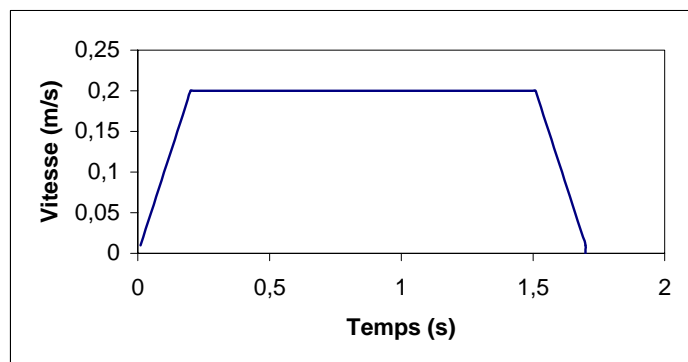
(b)

Figure V.35. Profils de la distance et de la vitesse pour  $V_{\text{initial}}=0.2$  m/s,  $V_{\text{final}}=0$  m/set  $d_{\text{but}}=30$ cm.

La figure V.36(a) montre que la distance-but est parcourue de (30 cm à 0 cm) en moins de 2 secondes (exactement 1.66 s) en générant une trajectoire (mouvement) lisse. La figure V.36(b) montre bien que cette distance (30 cm) a été parcourue en un temps minimum avec une accélération maximale jusqu'à atteindre la vitesse maximale de 0.2 m/s, ensuite en maintenant cette vitesse jusqu'à ce que la distance robot-position désirée (distance-but) soit suffisante (juste assez) pour décélérer à la vitesse finale désirée de 0 m/s.



(a)



(b)

Figure V.36. Profils de la distance et de la vitesse pour  $V_{\text{initial}}=0$  m/s,  $V_{\text{final}}=0$  m/set  $d_{\text{but}}=30$ cm.

Ces résultats de simulation montrent les capacités du contrôleur flou à générer des mouvements lisses illustrant ainsi son adaptation et son intelligence. Un tel contrôleur est

alors capable de remédier aux problèmes des approches analytiques et aux incertitudes conduisant à des effets indésirables tels que les changements de vitesse brutaux.

## V.7. Conclusion

Les résultats de simulation ont montré que les approches proposées réalisent la navigation d'un ou de plusieurs robots dans un environnement dynamique avec succès. En effet, dans l'approche basée sur le SHI RNAMF, nous pouvons conclure que l'apprentissage réalisé par la AMFS permet une navigation parfaite pour un ou deux robots dans un environnement qui leur est totalement inconnu à priori, et ne l'observent que partiellement lors de leur évolution. Cependant, cette approche est restreinte en raison de la difficulté de dériver toutes les règles nécessaires lorsque le nombre de robots augmente. En effet, le nombre de règles à utiliser croît rapidement en fonction du nombre de robots et par conséquent l'efficacité de cette approche ne se sent que lorsque le nombre de robots ne dépasse pas deux.

Tandis que pour l'approche basée sur l'apprentissage par renforcement, nous pouvons voir que l'algorithme Q-learning est efficace quand le nombre de robots augmente et quand il s'agit de programmer des robots à apprendre de manière en-ligne. Ces mêmes résultats nous ont permis d'évaluer la performance de l'approche proposée utilisant une table de valeurs commune. En effet, grâce à cette manière de faire, les robots arrivent à avoir toutes les informations nécessaires à leur évolution grâce au partage de ces mêmes informations. C'est dire que partager une connaissance permet de réaliser de bonnes performances quant à la navigation de ces robots.

Les résultats de simulation illustrent donc les capacités d'apprentissage et/ou de généralisation des approches proposées pour l'acquisition des comportements intelligents de la navigation. Ces résultats ont montré également que ces approches fournissent aux robots plus d'autonomie et d'intelligence et permettent une navigation coopérative et intelligente au sein d'un groupe de robots mobiles autonomes.

Les actions générées par ces approches sont envoyées aux actionneurs des robots afin d'être exécutées grâce à un contrôleur flou de mouvement qui permet la génération de mouvements lisses en temps réel. En effet, une approche expérimentale basée sur la stratégie du contrôle temps-minimal et de la logique floue est développée afin de construire ce contrôleur. En fait, ce contrôleur génère une commande de contrôle optimale à envoyer aux actionneurs afin de contrôler la vitesse du robot. Ces résultats de simulation montrent les capacités de génération de mouvements lisses illustrant ainsi son adaptation et son intelligence.

## **Conclusion Générale et Perspectives**

L'objectif de ce travail était la mise en œuvre de comportements collectifs permettant la réalisation de la tâche de transport. Pour ce faire, trois phases essentielles ont été traitées. La première concerne le développement d'un comportement d'évitement de collisions et d'attraction vers la cible permettant aux robots d'évoluer avec succès dans un même environnement. La seconde phase s'intéresse à la réalisation d'une figure donnée correspondant à l'objet à transporter. Quant à la dernière phase, elle s'occupe de la navigation du groupe ainsi formé.

Afin d'atteindre l'objectif fixé, un ensemble de comportements élémentaires qui permettent l'émergence de comportements plus complexes grâce à l'apprentissage et à l'interaction des robots avec l'environnement ont été proposés. En effet, la synthèse d'un comportement de groupe complexe et cohérent a été traitée afin d'arriver à solutionner le problème du transport d'objets lourds et encombrants. Le préalable aux étapes de résolution de la tâche de transport est l'acquisition du comportement d'évitement de collisions et d'attraction vers la cible qui est en fait l'un des comportements de base qu'un robot doit impérativement acquérir afin d'évoluer avec succès dans un environnement donné. En exploitant ce comportement, une approche permettant le transport d'un objet d'un point de départ à un point d'arrivée dans un environnement à priori inconnu a été proposée. Celle-ci se base sur deux stratégies de contrôle, l'une distribuée où les robots tentent d'atteindre des buts individuels et l'autre centralisée dont le but est commun.

Trois approches ont donc été développées toutes basées sur l'apprentissage qui permet aux robots de s'adapter, d'anticiper et d'interagir avec leur environnement.

La première exploite les RN et particulièrement l'apprentissage par AMFS (ArtMap Floue Simplifiée) pour réaliser le comportement d'évitement de collisions et d'attraction vers la cible de deux robots dans un environnement partiellement structuré. La coopération entre les robots est basée sur la communication, où chacun poursuit un objectif individuel. Afin de résoudre le problème de l'évitement de collisions, on a eu recours au système de communication LOCISS pour pouvoir mettre en œuvre le dispositif des règles prédéterminées qui consiste certes en un ensemble de comportements élémentaires mais qui développe un certain sens du civisme chez nos robots. Cette approche est intéressante lorsqu'on ne dispose que de deux robots d'une part, et d'autre part lorsqu'une connaissance sur la tâche est disponible. Cependant, lorsque le nombre de robots augmente, cette approche devient vite inefficace dû à la complexité des situations auxquelles ces robots doivent faire face.

Une seconde approche a donc été introduite. Celle-ci se base sur l'AR qui permet à un grand nombre de robots évoluant dans le même environnement à réaliser leurs objectifs respectifs (atteindre leurs cibles) sans utiliser une grande quantité d'information. La seule information disponible est un scalaire retourné par l'environnement et à la base duquel les robots arrivent à réaliser leur tâche. Cette approche utilise un certain nombre de comportements élémentaires qui combinés judicieusement grâce à l'apprentissage arrive à

réaliser le comportement d'évitement de collisions et d'attraction vers la cible. L'objectif de l'approche développée est d'étudier les comportements acquis par apprentissage par renforcement ainsi que l'influence que peut avoir l'interaction environnement/robot sur cet apprentissage, pour montrer que malgré cette simplicité de programmation basée directement sur les valeurs fournies par les capteurs, celle-ci engendre plusieurs caractéristiques. D'une part, la capacité de calcul peut ne pas être très importante, le temps de réaction est en général assez court, d'autre part les capteurs peuvent ne pas être très précis, pour un environnement pouvant être dynamique et fortement perturbé.

C'est pour cette raison que cette approche par son algorithme de renforcement habilite un ou plusieurs robots, qui n'ont aucune idée préconçue sur l'environnement, à s'adapter aux situations rencontrées. Ils ont également acquis ce comportement d'évitement de collisions et d'attraction vers la cible plus rapidement par l'utilisation d'une base de connaissance commune qui permet le partage d'informations parmi les robots et de ce fait gagne en temps d'apprentissage et en connaissance comme montré dans cette thèse.

La dernière approche utilise le comportement d'évitement de collisions et d'attraction vers la cible appris par les robots combiné à d'autres comportements afin d'une part, former des figures géométriques correspondant à l'objet à transporter et d'autre part, se déplacer en maintenant cette formation dans un environnement à priori inconnu. Pour cela, deux stratégies de contrôle ont été développées. La première permet à un groupe de robots mobiles d'effectuer une tâche commune et assurer une bonne coopération entre eux. Cette tâche est la formation des figures géométriques tels que le cercle, le cercle rempli, le polygone et la ligne avec un petit ou un grand nombre de robots en jouant sur les paramètres utilisés (e.g., le diamètre). L'apprentissage réalise, par renforcement, la formation d'une figure donnée en utilisant les comportements élémentaires et les fonctions de renforcements proposés. La seconde stratégie permet au groupe formé de naviguer dans un environnement inconnu tout en évitant les obstacles sur son chemin. Elle utilise l'approche d'évitement de collisions et d'attraction vers la cible ainsi que la technique bien connue de "meneur-suiveurs" modifiée dans ce travail afin de maintenir la formation réalisée et la faire naviguer pour atteindre sa cible.

Afin de valider ce travail, des simulations de ces différentes approches ont été réalisées où la navigation d'un ou de plusieurs robots, la formation des figures géométriques ainsi que la navigation en formation ont été abordées. Les résultats de ces simulations effectuées dans différents environnements ont également été présentés et discutés.

En finalité, doté d'un certain nombre de compétences minimales (dans notre cas les comportements élémentaires), le robot apprend par lui-même de nouveaux comportements en se basant sur l'expérience déjà acquise. En effet, d'une part, le comportement d'évitement de collisions et d'attraction vers la cible a été appris par le groupe de robots en exploitant certains comportements élémentaires et il a été montré que grâce à l'apprentissage partagé, ces robots ont pu gagner en temps et ont pu bénéficier des expériences des autres grâce à une coopération simple. D'autre part, la formation de figures géométriques ainsi que leur déplacement ont également été acquis par les robots en exploitant les comportements élémentaires proposés.

Cependant, comme dans tout travail, il y a des améliorations qui peuvent être apportées et des limitations qui peuvent être levées et qui sont principalement liées d'une part à l'apprentissage et d'autre part aux simplifications introduites. Le premier point à relever est lié à l'implémentation de ces approches sur robots réels. En effet, lorsqu'on utilise un robot



réel, l'espace des états est en général trop grand à explorer. Une exploration exhaustive ou une mémorisation de toutes les paires état-action est impossible. Une solution à ce problème est le processus de généralisation à travers l'utilisation des réseaux de neurones artificiels comme c'est proposé dans [161].

Le second point concerne le critère d'évaluation afin de prendre des décisions qui est représenté par la fonction de renforcement dans un schéma d'AR. La construction de cette fonction requiert plusieurs expérimentations comme exposé dans ce travail, il serait intéressant de la générer automatiquement. Une solution serait l'utilisation de la notion d'émotion dans le schéma d'AR [75]. Ainsi, l'émotion peut être introduite dans le processus d'apprentissage afin d'éviter de trouver la fonction de renforcement empiriquement qui généralement est obtenue après plusieurs essais. En utilisant un module émotionnel, les renforcements sont calculés à la base d'une fonction émotionnelle traduisant la "bonne" ou la "mauvaise" qualité de l'action prise. La fonction de renforcement pourrait ainsi être optimisée durant le processus afin d'obtenir le meilleur apprentissage. Le traitement émotionnel ouvre alors l'opportunité de s'occuper des stimuli imprédictibles en prenant les décisions adéquates efficacement, en d'autres termes, sans le besoin de gaspiller du temps dans l'analyse exhaustive de l'aspect cognitif des stimuli [54].

Enfin, les perspectives de ce travail sont multiples. Deux pourraient être dégagées à court terme. La première concerne l'introduction du concept multi-agents afin de rendre l'approche exploitable et applicable sans intervention humaine. En effet, grâce aux avancées réalisées dans le domaine de l'IAD et plus particulièrement les systèmes multi-agents, leur application à la robotique collective est devenue une réalité. Cette avancée a permis d'une part, de passer de travaux totalement théoriques et dans des domaines très abstraits où une perception parfaite était toujours supposée à des travaux tenant compte cette fois-ci de la notion d'incertitude qui est importante quand on traite avec le monde réel. Dans ce contexte, une tentative de proposition d'une architecture de contrôle multi-agents pour la tâche de transport considérée a été étudiée [13]. C'est une architecture comportementale caractérisée par sa flexibilité, sa robustesse et son adaptation dans des environnements réels. Cette architecture inclut plusieurs formations permettant aux robots de choisir la figure correspondant à l'objet à transporter. La décomposition du système en plusieurs agents donne une meilleure réactivité aux situations changeantes où chaque agent prend ses décisions seul sans passer par un superviseur. Ainsi, cette architecture considère des robots qui s'organisent en groupe afin de réaliser la tâche assignée d'une manière coopérative. Tous les agent-robots ont la même architecture de contrôle. C'est un système homogène qui consiste en un ensemble de comportements codant le contrôle et la connaissance du système qui doit répondre aux différents événements. C'est donc une architecture de contrôle comportementale composée d'un ensemble de comportements qui permettent aux robots de réaliser la tâche demandée. Elle est capable de réaliser un raisonnement de haut niveau et réagir aux changements survenant dans l'environnement.

Une autre perspective résiderait dans l'exploitation des travaux de recherche réalisés dans le domaine des systèmes distribués. Les méthodes de détection de blocage et d'allocation de tâches peuvent être utilisées afin de permettre à plusieurs robots de partager des ressources limitées [170]. La communication par diffusion qui est souvent utilisée dans la robotique collective donne de faibles résultats lorsque le nombre de robots augmente. De ce fait, quand ce nombre devient important, des techniques relatives aux réseaux informatiques seraient

intéressantes. Un riche éventail d'algorithmes, de protocoles, de modélisations et d'analyses de performance des réseaux informatiques peut être appliqué à la robotique collective. Ce qui a déjà commencé à se faire avec des robots contrôlés via le web.

# Bibliographie

- [1] M. D. Adams, H. Hu et P.J. Probert. Towards a real-time architecture for obstacle avoidance and path planning in mobile robots. *Proc. Int. IEEE Conf.* CH2876-1/90/0000-0584, 1990, pp. 584-589.
- [2] P. Agre et D. Chapman. What are plans for? In: *Maes P. (ed.) Designing Autonomous Agents*, Cambridge, MA:MIT Press, 1990, pp. 17-34.
- [3] J.A. Anderson. *An introduction to neural networks*. The MIT Press, Cambridge, Massachusetts, London, England, 1995.
- [4] Y. Arai *et al.* Realisation of autonomous navigation in multirobot environment. *Proc. 1998 IEEE/RSJ int. Conf. on Intelligent Robots and Systems*, 1998, pp. 1999-2004.
- [5] Y. Arai, T. Fujii, H. Asama, Y. Yataoka, H. Kaetsu, A. Matsumoto et I. Endo. Adaptive behavior acquisition of collision avoidance among multiple autonomous mobile robots. *Proc. Int. IEEE Conf. On Intelligent Robots and Systems*, Grenoble, France, 1997, pp. 1762-1767.
- [6] Y. Arai, S. Suzuki, S. Kotosaka, H. Asama, H. Kaetsu et I. Endo. Collision avoidance among multiple autonomous mobile robots using LOCISS (LOcally Communicable Infrared Sensory System). *Proc. Int. IEEE Conf. on Robotics and Automation*, Minneapolis, Minnesota, 1996, pp. 2091-2096.
- [7] T. Arai E. Yoshida, M. Yamamoto, J. Ota et D. Kurabayashi. Information diffusion by local communication of multiple mobile robots. *Proc. IEEE Conf. on Systems, man and cybernetics*, 1993, pp. 535-540.
- [8] T. Arai et J. Ota. Motion planning of multiple robots. *Proc. IEEE/RSJ IROS*, 1992, pp. 1761-1768.
- [9] R. Arkin. Cooperation without communication: multiagent schema-based robot navigation. *Int. journal of robotic systems*, 9(3), 1992, pp. 351-364.
- [10] R. Arkin. Towards the unification of navigational planning and reactive control. *Proc. American Association for AI Spring Symp. on Robot Navigation*, Palo Alto, CA: AAAI/MIT Press, 1989, pp. 1-5.
- [11] M. Asada E. Uchibe, S. Noda, S. Tawaratsumida et K. Hosoda. Coordination of multiple behaviors acquired by a vision-based reinforcement learning. *Proc. IEEE/RSJ/JI Int. Conf. on intelligent robots and systems*, Munich: IEEE computer society Press, 1994, pp. 917-924.
- [12] H. Asama M. Sato, L. Bogoni, H. Kaetsu, A. Matsumoto et I. Endo. Development of an omni-directional mobile robot with 3 DOF decoupling drive mechanism. *Proc. of the IEEE Int. Conf. on Robotics and Automation*, 1995, pp. 1925-1930.
- [13] **O. Azouaoui**, H. Yazid, A. Farah et K. Achour. Behavioral multi-agent based control architecture for autonomous robotic systems. Soumis comme chapitre dans un livre sur les Systèmes Multi-Agents. Première révision acceptée, il est en cours d'une seconde révision.
- [14] **O. Azouaoui**, R. Benghanem, Chérifi, A. Farah et K. Achour. Reinforcement learning based group navigation approach for multiple autonomous robotic systems. A apparaît dans *Advanced robotics journal*, Vol.20 N°5, 2006.
- [15] **O. Azouaoui**, A. Chérifi, R. Bensalem et A. Farah. Reinforcement learning based group navigation approach for multiple autonomous robotic systems. *Proc. IEEE Int. Conf. on Mechatronics and Automation (ICMA'05)*, 2005, pp.1539-1544.
- [16] **O. Azouaoui**, H. Yazid, F. Boulahlib et N. Chabane. Multi-agent approach based navigation system for mobile robots: Towards a coordination using traffic rules. *Proc. 17<sup>th</sup> Int. Conf. on Industrial and Engineering Applications of Artificial Intelligence and Expert Systems (IEA/AIE 2004)*, Montréal, 2004.
- [17] **O. Azouaoui** et A. Chohra. Pattern classifiers based on soft computing and their FPGA integration for intelligent behavior control of mobile robots. *Proc. IEEE 11<sup>th</sup> Int. Conf. on Advanced Robotics ICAR'2003*, Coimbra, Portugal, 2003, pp. 148-154.
- [18] **O. Azouaoui** et A. Chohra. Soft computing based pattern classifiers for the obstacle avoidance behavior of Intelligent Autonomous Vehicles (IAV). *Applied Intelligence: The International J. of Artificial Intelligence, Neural Networks, and Complex Problem-Solving Technologies*, Vol. 16, N° 3, May/June 2002, pp. 249-271.
- [19] **O. Azouaoui**, M. Ouaz et A. Farah. Reinforcement Learning (RL) based collision avoidance approach for multiple Autonomous Robotic Systems (ARS). *Proc. IEEE 10th Int. Conf. on Advanced Robotics ICAR'2001*, Budapest, Hungary, 2001, pp. 561-566.
- [20] **O. Azouaoui**, M. Ouaz, A. Chohra, A. Farah et K. Achour. Fuzzy ArtMap Neural Network (FAMNN) based collision avoidance approach for Autonomous Robotic Systems (ARS). *Proc. IEEE 2nd Int. Workshop on Robot Motion and Control RoMoCo'01*, Bukowy Dworek, Poland, 2001, pp. 285-290.
- [21] **O. Azouaoui** et A. Chohra. Neural group navigation approach for Autonomous Robotic Systems (ARS). *Proc. Int. ICSC Symp. on Engineering of Intelligent Systems EIS'2000*, Scotland, U.K. 27-30 June 2000.

- [22] O. Azouaoui et A. Chohra. Evolution, behavior, and intelligence of Autonomous Robotic Systems (ARS). in *Proc. 3rd Int. IFAC Conf. Intelligent Autonomous Vehicles*, Spain, March 25-27 1998, pp. 139-145.
- [23] T. Balch et R. Arkin. Motor schema-based formation control for multiagent robots teams. *Proc. Int. Conf. on multiagent systems*, pp. 10-16, San Francisco, CA, 1995.
- [24] A.A. Baloch et A.M. Waxman. Visual learning, adaptive expectations, and behavioral conditioning of the mobile robot MAVIN. *Neural networks*, vol. 4, 1991, pp. 271-302.
- [25] D. Barnes et J. Gray. Behaviour synthesis for co-operant mobile robot control. *Proc. Int. Conf. on control*, 1991, pp. 1135-1140.
- [26] A.G. Barto, R. S. Sutton et C. J. C. H. Watkins. Learning and sequential decision making. In Gabriel M. and Moore J., editors, *Learning and computational neuroscience: foundations of adaptive networks*, 1989, pp. 539-603.
- [27] R. Beckers O. E. Holland et J.L. Deneubourg. From local actions to global tasks: stigmergy and collective robotics. *Proc. Artificial Life IV*, Brooks R. and Maes P. eds., MIT Press, 1994, pp. 181-189.
- [28] G. Beni et J. Wang. Swarm intelligence in cellular robotic systems. *Proc. of the NATO advanced workshop on robots and biological systems*, Tuscany, Italie, 1994.
- [29] G. Beni et J. Wang. Theoretical problems for the realization of distributed robotic system. *Proc. IEEE Int. Conf. on Robotics and Automation*, CA, April 1991, pp. 1914-1920.
- [30] G. Beni. The concept of cellular robotic system. *Proc. IEEE Int. symposium on intelligent control*, 1988, pp. 57-62.
- [31] K. Berns, R. Dillmann et U. Zachmann. Reinforcement-learning for the control of an autonomous mobile robot. *Proc. IEEE/RSJ Int. Conf. on Intelligent Robots and Systems IROS92*, Raleigh NC, July 7-10 1992, pp. 1808-1815.
- [32] E. Bonabeau, G. Theraulaz, J.L. Deneubourg, S. Aron et S. Camazine. Self-organization in social insects. *Trends in ecology and evolution* 12(5), 1997, pp. 188-193.
- [33] F. Brandt, W. Brauer et G. Weiss. Task assignment in multiagent systems based on vickrey-type auctioning and leveled commitment contracting. *Proc. 4<sup>th</sup> int. workshop on cooperative information agents*, 2000, pp. 95-106.
- [34] W.L. Brogan. *Modern control theory*. Prentice-Hall, NJ, pp. 433-465, 1985.
- [35] R.A. Brooks et L. Stein. Building brains for bodies. *Autonomous Robots* 1(1), 1994, pp. 7-25.
- [36] R.A. Brooks. Intelligence without reason. *Proc. Int. joint Conf. on artificial intelligence*, Sydney, Australia, Cambridge MA: MIT Press, 1991, pp. 569-595.
- [37] R.A. Brooks. Intelligence without representation. *Artificial Intelligence* 47, 1991, pp. 139-160.
- [38] R.A. Brooks, P. Maes, M. J. Mataric et G. More. Lunar base construction robots. In *IEEE workshop IROS*, Tsuchiura, Japan, 1990.
- [39] R.A. Brooks. A Robust layered control system for a mobile robot. *IEEE Journal of Robotics and Automation*, vol. RA-2, no. 1, March 1986, pp. 14-23.
- [40] P. Caloud, W. Choi, J.C. Latombe, C.L. Pape et M. Yim. Indoor automation with many mobile robots. *Proc. IEEE/RSJ IROS*, 1990, pp. 67-72.
- [41] V. Camps. *Vers une théorie de l'auto-organisation dans les systèmes multi-agents basée sur la coopération : application à la recherche d'information dans un système d'information répartie*. Thèse de doctorat, Université Paul Sabatier de Toulouse, 1998.
- [42] Y.U. Cao A.S. Fukunaga, A.B. Kahng et F. Meng. Cooperative mobile robotics: antecedents and directions. in *autonomous robots*, volume 4, 1997, pp. 7-27.
- [43] G.A. Carpenter S rossberg, N. Markuzon, J.H. Reynolds et D.B. Rosen. Fuzzy ARTMAP: A neural network architecture for incremental supervised learning of analog multidimensional maps. *Proc. IEEE Trans. on Neural Networks*, vol. 3, no. 5, September 1992, pp. 698-713.
- [44] G.A. Carpenter, S Grossberg et DB Rosen. Fuzzy ART: Fast stable learning and categorization of analog patterns by an adaptive resonance system. *Neural Networks*, vol. 4, 1991, pp. 759-771.
- [45] Q. Chen, et J. Y. S. Luh. Coordination and control of a group of small mobile robots. *Proc. IEEE conf. on Rob. and Aut.*, 1994, pp.2315-2320.
- [46] S. Cherian et W. Troxell. Intelligent behavior in machines emerging from a collection of interactive control structures. *Computational Intelligence*, Vol. 11, No. 4, November 1995, pp. 565-592.
- [47] A. Chohra et O. Azouaoui. Neural Networks (NN) using Genetic Algorithms (GA) and Gradient Back-Propagation (GBP) for an intelligent obstacle avoidance Behavior. *Proc. Int. Conf. on Advanced Robotics (ICAR03)*, Portugal, June 2003, pp. 917-922.
- [48] A. Chohra, R. Tiar et O. Azouaoui. Fuzzy Motion Controller (FMC) for Intelligent Autonomous Vehicles (IAV). *Proc. Int. ICSC Symposium on Engineering of Intelligent Systems EIS'2000*, Scotland, U.K. 27-30 June 2000.
- [49] A. Chohra A. Farah et C. Benmehrez. Neural navigation approach for Intelligent Autonomous Vehicles (IAV) in Partially structured Environments. *The Int. Journal of Applied Intelligence: Artificial Intelligence*,

- Neural Networks, and Complex Problem-Solving Technologies*, Vol. 8, No. 3, Kluwer Academic Publishers, May/June 1998, pp. 219- 233.
- [50] A. Chohra. Fuzzy ArtMap Neural Networks (FAMNN) based navigation for Intelligent Autonomous Vehicles (IAV) in partially structured environments. *Proc. ICSC Int. IFAC-IEEE Conf. on Neural Computing*, Vienna, Austria, September 23-25 1998, pp. 747-754.
- [51] M.K. Ciliz et C. Isik. Fuzzy rule-based motion controller for an autonomous mobile robot. *Robotica*, Vol. 7, 1989, pp. 37-42.
- [52] R.H. Crites et A.G. Barto. Elevator group control using multiple reinforcement learning agents. *Machine Learning* 33, 1998, pp. 235-262.
- [53] J. L. Crowley. Navigation for an intelligent mobile robot. *IEEE Journal of Robotics and Automation*, Vol. RA-1, No. 1, March 1985, pp. 31-41.
- [54] A. Damasio. *Descartes' error: emotion, reason, and the human brain*. New York: Gosset/Putnam, 1994.
- [55] J.L. Deneubourg, G. Theraulaz et R. Beckers. Swarm-made architectures. In *Varela E.J. et Bourguine P., Eds. Tward a practice of autonomous systems, Proc. of the first European Conf. on Artificial Life*, MIT Press/Bradford Books, 1992.
- [56] J.L. Deneubourg, C. Detrain et L. Chretien. The dynamics of collective sorting: robots-like ants and ant-like robots. *Proc. Simulation of adaptive behavior*, MITPress, 1990, pp. 356-363.
- [57] J.L. Deneubourg, J.M. Pasteels et S. Goss. Self-organization mechanisms in ants societies (II) : learning in foraging and division of labour. In *Pasteels J.M. et Deneubourg J.L. Eds. From individual to collective behaviour in social insectes, Experientia supplementum 54*, Birkhauser Verlag, 1987, pp. 177-196.
- [58] B.R. Donald, J. Jennings et D. Rus. Towards cooperating autonomous mobile robots. *Proc. Int. symposium on robotics research*, Hidden Valley, PA, October, 1993.
- [59] A. Drogoul et J. Ferber. From tom thumb to the dockers : some experiments with foraging robots. *Proc. simulation of adaptive behavior*, 1993.
- [60] A. Dubrawski. Tuning neural networks with stochastic optimization. *Proc. Int. IEEE/RSJ Conf. on Intelligent Robots and Systems*, Vol. 2, Grenoble, France, September 07-11 1997, pp. 614-620.
- [61] A. Dubrawski et J.L. Crowley. Self-supervised neural system for reactive navigation. *Proc. Int. IEEE Conf. on Robotics and Automation*, vol. 3, San Diego, California, May 08-13, 1994, pp. 2076-2081.
- [62] J. Dudek M.R.M. Jenkin, E. Melios et D. Wilkes. Taxonomy for multi-agent robotics. *Journal of autonomous robots*, vol. 3, 1996, pp. 375-397.
- [63] E. Epharti, M.E. Pollack et J.S. Rosenschein. A tractable heuristic that maximizes global utility through local plan combination. *Proc. 1<sup>st</sup> Int. Conf. on multi-agent systems*, 1995.
- [64] B. G. Farley et W. A. Clark. Simulation of self-organizing systems by digital computer. *I.R.E Transactions on Information Theory*, 4, 1954, pp. 76-84.
- [65] T. Finin R. Fritzon, D. McKay et R. McEntire. KQML as an agent communication language. In Bradshaw JM eds, *software agents*. Cambridge, MA:AAAI/MIT Press, 1996.
- [66] N.R. Franks et J.L. Deneubourg. Self-organising nest construction in ants: individual worker behavior and the nest's dynamics. *Animal behaviour* 54, 1997, pp. 779-796.
- [67] J. Fredslund et M.J. Mataric. A general algorithm for robot formations using local sensing and minimal communications. *IEEE Transactions on Robotics and Automation*, 18(5), 2002, pp.837-846.
- [68] J. Fredslund et M. J. Mataric. Robots in formation using local information. *Proc. 7th International Conference on Intelligent Autonomous Systems (IAS-7)*, CA, Mar 25-27 2002, pp. 100-107.
- [69] J.A. Freeman et D.M. Skapura. *Neural networks : Algorithms, applications, and programming techniques*. Addison-Wesley Publishing Company, 1992.
- [70] T. Fujii, Y. Arai, H. Asama et I. Endo. Multilayered reinforcement learning for complicated collision avoidance problems. *Proc. 1998 IEEE Int. Conf. on Robotics and Automation*, 1998, pp. 2186-2191.
- [71] Y. Fujita, S. Fujita, M. Yamashita, I. Suzuki et H. Asama. Learning-based automatic generation of collision avoidance algorithms for multiple autonomous mobile robots. *Proc. IEEE/RSJ int. Conf. on Int. Robots and Sys.*, 1998, pp. 1553-1558.
- [72] K. Fujimura. *Motion planning in dynamic environments*. Springer-Verlag, New York, NY, 1991.
- [73] T. Fukuda et S. Nakagawa. Approach to the dynamically reconfigurable robotic system. *J. of Intelligent and Robotic Systems*, V. 1, 1988, pp.55-72.
- [74] S.D. Gadanho. *Reinforcement learning in autonomous robots: an empirical investigation of the role of emotions*. PhD Thesis, University of Edinburgh, 1999.
- [75] D. Gage. How to communicate to zillions of robots. In *mobile robots VIII*, SPIE, 1993, pp. 250-257.
- [76] V. Genovese, P. Dario, R. Magni et L. Odetti. Self-organizing behavior and swarm intelligence in a pack of mobile miniature robots in Search of pollutants. *Proc. IEEE/RSJ Int. Conf. on Intelligent Robots and Systems*, Raleigh, NC, July 1992, pp. 1575-1582.

- [77] B.P. Gerkey et M.J. Mataric. Sold! auction methods for multi-robot coordination. *IEEE transactions on robotics and automation, special issue on multi-robot systems* Vol. 18, 2002.
- [78] B. Gerkey et M. Mataric. Principled communication for dynamic multi-robot task allocation. In Rus D. and Singh S. (eds.) *Proc. of the Int. symposium on experimental robotics*, Waikiki Hawaii, Berlin Heidelberg:Springer-Verlag, 2002, pp. 341-352.
- [79] D. Goldberg. *Genetic algorithms in search, optimization, and machine learning*. Addison Wesley, 1989.
- [80] M.K. Habib, H. Asama, I. Endo, A. Matsumoto et Y. Ishida. Simulation environment for an autonomous decentralized multi-agent robotic system. *Proc. IEEE/RSJ Int. Conf. on Intelligent Robots and Systems*, NC, July 1992, pp. 1550-1557.
- [81] H. Herbstreith, L. Gmeiner, P. Preuss. A target-directed neurally controlled vehicle. *Proc. Int. IFAC Conf. Artificial Intelligence in Real-Time Control*, Delft, The Netherlands, 1992, pp. 67-71.
- [82] A.W. Ho et G.C. Fox. Neural network near-optimal motion planning for a mobile robot on binary and varied terrains. *Proc. IEEE Int. Work. on Int. Rob. and Sys., IROS'90*, 1990, pp. 593-600.
- [83] J.H. Holland. *Adaptation in natural and artificial systems*. Ann Arbor, MI: the university of Michigan Press, 1975.
- [84] A Huberman et R.M. Lukose. Social dilemmas and internet congestion. *Science*, 277, 1997, pp. 535-537.
- [85] T. Ishida. Towards organizational problem solving. *Proc. IEEE Int. Conf. on Rob. and Aut. 1050-4729/93 \$3.00*, 1993, pp. 839-845.
- [86] S. Ishikawa, F. Hara, H. Hosokpj. Cooperative route-searching behavior of multi-robot system using hello-call communication. *Proc. IEEE/RSJ IROS*, 1993, pp. 1156-1993.
- [87] N. Jennings. Coordination techniques for distributed artificial intelligence. *In contribution to foundations of DAI*, 1996, pp. 187-210.
- [88] C. Jones M.J. Mataric et B. Werger. Cognitive processing through the interaction of many agents. *Invited contribution to the Encyclopedia of cognitive science, nature publishing group*, Mac Millan reference limited, Nov 2002.
- [89] L.P. Kaelbling. *Learning in embedded systems*. PhD thesis, MIT Press, 1993.
- [90] Y. Kawachi M. Inaba et T. Fukuda. A Principle of distributed decision making of cellular robotic system (CEBOT). *Proc. IEEE Int. Conf. on Rob. and Aut. 1050-4729/93 \$3.00*, 1993, pp.833-838.
- [91] T. Khanna. *Foundations of neural networks*. Addison-Wesley, New York, 1990.
- [92] T. Kasuba. Simplified fuzzy artmap. *AI EXPERT*, November 1993, pp. 18-25.
- [93] B. Kosko. *Neural networks and fuzzy systems*. University of Southern California, Prentice Hall, A Simon & Schuster Company, Englewood Cliffs, New Jersey 07632, 1992.
- [94] M. Krieger, J.B. Billeter et L. Keller. Ant-like task allocation and recruitment in cooperative robots. *Nature* 406, 2000, pp. 992.
- [95] A. J. Krijgsman et R. Jager. Intelligent control: Theory and applications. *Proc. Int. IFAC Conf. Intelligent Components and Instruments for Control Applications*, Malaga, Spain, 1992, pp. 57-67.
- [96] C.R. Kube et E. Bonabeau. Cooperative transport by ants and robots. *Robotics and autonomous systems*, 30, 2000, Issue 1-2:85-101.
- [97] S.Y. Kung. *Digital neural networks*. PTR Prentice-Hall, New Jersey 07632, 1993.
- [98] S. Labidi et W. Lejouad. De l'intelligence artificielle distribuée aux systèmes multi-agents. *Rapport de recherche INRIA N°2004*, Août 1993.
- [99] C.G. Langton. *Artificial life*. Addison-Wesley Publishing Company, 1989.
- [100] J.P. Laumond. feasible trajectories for mobile robots with kinematic and environment constraints. *Proc. Int. Conf. on Autonomous Systems*, Amsterdam, Netherland, 1987, pp.---
- [101] J. Latombe. *Robot motion planning*. Kluwer academic, Boston, MA, 1991.
- [102] H.M. Lee et C.S. Lai. Supervised extended ART: A fast neural network classifier trained by combining supervised and unsupervised learning. *The International Journal of Applied Intelligence: Artificial Intelligence, Neural Networks, and Complex Problem-Solving Technologies*, 6, Kluwer Academic Publishers, 1996, pp. 117-128.
- [103] C.C. Lee. Fuzzy logic in control systems: Fuzzy logic controller, Part I and Part II. *IEEE Trans. on Systems, Man and Cybernetics*, Vol. 20, No. 2, March/April 1990, pp. 404-435.
- [104] C.S. Lin et C. D. Wann. A parallel processing model for robot path planning on grid terrains. *Int. J. Robotics and Automation*, Vol. 6, No. 1, 1991, pp. 01-11.
- [105] G. Lucarini, M. Varoli, R. Cerutti et G. Sandini. Cellular robotics: Simulation and HW implementation. *IEEE Int. Conf. on Rob. and Aut. 1050-4729/93 \$3.00*, 1993, pp.846-852.
- [106] M.J. Mataric. Behavior-based robotics as a tool for synthesis of artificial behavior and analysis of natural behavior. *Trends in cognitive science* 2 (3), 1998, pp. 82-87.
- [107] M.J. Mataric. Behavior-based control: Examples from navigation, learning and group behavior. *Journal of experimental and theoretical artificial intelligence* 9, 1997, pp. 323-336.

- [108] M.J. Mataric. Issues and approaches in the design of collective autonomous agents. *Robotics and autonomous systems*, 1995, 16 n 2-4: 321-331.
- [109] M.J. Mataric. *Interaction and intelligent behavior*. MIT EECS PhD thesis, MIT AI Lab, 1994.
- [110] M.J. Mataric. Reward functions for accelerated learning. *Proc. of the Int. machine learning Conf.*, 1994, pp. 181-189.
- [111] M.J. Mataric. Distributed approaches to behavior control. In *SPIE-sensor fusion V*, vol. 1828, 1992, pp. 373-382.
- [112] M.J. Mataric. Behavior-based control: Main properties and implications. *Proc. IEEE Int. Conf. on Robotics and Automation, Workshop on Architectures for Intelligent Control Systems*, France, 1992, pp. 46-54.
- [113] R. Mataric. Navigating with a rat brain: A neurobiologically-inspired model for robot spatial representation. In Mayer J-A and Wilson S. (eds.) *Proc. from animals to animats 1, 1<sup>st</sup> internaional Conf. on simulation of adaptive behavior*, Cambridge MA: MIT press, 1990, pp. 169-175.
- [114] D. McFarland. Towards robot cooperation. *Proc. simulation of adaptive behavior*, 1994.
- [115] L.R. Medsker. *Hybrid intelligent systems*. Kluwer Academic Publishers, 1995.
- [116] M. Meng et A.C. Kak. Mobile robot navigation using neural networks and nonmetrical environment models. *IEEE Control Systems*, October 1993, pp. 30-39.
- [117] Merriam-Webster, *Webster's 7<sup>th</sup> collegiate dictionary*, 1963.
- [118] D.P. Miller. Multiple behavior-controlled micro-robots for planetary Surface Missions. *Proc. 1990 IEEE Int. Conf. on Systems, Man, and Cybernetics*, Los Angeles, Ca, Nov. 1990, pp. 281-292.
- [119] H. Murao et S. Kitamura. Q-Learning with Adaptive State Segmentation (QLASS). *0-8186-8138-1/97 \$10.00 © 1997 IEEE*, 1997, pp. 179-184.
- [120] D.J. Naffin et G. S. Sukhatme. Negotiated formations, *Proc. of the 8<sup>th</sup> Conference on intelligent autonomous systems*. Amsterdam, the Netherlands, 2004, pp.181-190.
- [121] M. Nicolescu et M.J. Mataric. Learning cooperation from human-robot interaction. *Proc. Fifth Int. symposium on distributed autonomous robotic systems (DARS-00)*, 4-6 Oct, TN, 2000.
- [122] T.N. Nguyen et al. BUGS: Multiple robots approach to UXO & mine field neutralization. *3rd Int. IFAC Conf. Intelligent Autonomous Vehicles*, 1998, pp.118-124.
- [123] T. Ogata et S. Sugano. Development of emotional communication between robots and humans. *Proc. IEEE/RSJ Int. Conf. on intelligent robots and systems (IROS-00)*, Japan, Nov 2000, pp. 175-180.
- [124] L. Parker. ALLIANCE : An architecture for fault tolerant, cooperative control of heterogeneous mobile robots. *Proc. of the 1994 Int. Conf. on intelligent robots and systems*, 1994 Germany, 1994, pp. 776-783.
- [125] L. Parker. *Heterogeneous multi-robot cooperation*. PhD thesis, MIT EECS Dept, 1994.
- [126] D.W. Patterson. *Artificial neural networks: Theory and applications*. Prentice-Hall, Simon & Schuster (Asia) Pte Ltd, Singapore, 1996.
- [127] W. Pedrycz. *Fuzzy control and fuzzy systems*. John Wiley & Sons Inc., 1988.
- [128] R. Pfeifer et C. Scheier. *Understanding intelligence*. The MIT Press, Cambridge Massachusetts, 1999.
- [129] R. Pfeifer. Building 'fungus eaters': Design principles of autonomous agents. *Fourth Int. Conf. on Simulation of Adaptive Behavior: From Animals to Animats*, Massachusetts, USA, September 09-13 1996, pp. 03-12.
- [130] P. Pignon T. Hasegawa, J.P. Laumond. Structuration de l'espace pour les robots mobiles. *Revue Scientifique et Technique de la Défense*, 2, 1994, pp. 17-31.
- [131] S. Premvuti et S. Yuta. Consideration on the cooperation of multiple autonomous mobile robots. *Proc. IEEE/RSJIROS*, 1990, pp. 59-63.
- [132] C. Reynolds. Flocks, herds, and schools: A distributed behavior model. *Computer graphics* 21(4), 1987, pp. 25-34.
- [133] N. Richard. *Description de comportements d'agents autonomes évoluant dans des mondes virtuels*. Thèse de doctorat de l'École Nationale Supérieure des Télécommunications, 2000.
- [134] E.Y Rodin et S.M. Amin. Intelligent navigation for an autonomous mobile robot. *Proc. Int. Symposium on Intelligent Control*, Arlington, VA, 1998, pp. 366-369.
- [135] J.S. Roseinchein et G. Zlotkin. *Rules of encounter: designing conventions of automated negotiation among computers*. Cambridge, MA: MIT Press, 1994.
- [136] J. Rosenchein. *Rational interaction: Cooperation among intelligent agents*. PhD thesis, Stanford university, 1986.
- [137] P. Rusu, E. M. Petriu, T.E. Whalen, A. Cornell, et H. J. W. Spoelder. Behavior-based neuro-fuzzy controller for mobile robot navigation. *IEEE Transactions on Instrumentation and Measurement*, Vol. 52, No. 4, 2003, pp.1335-1340.
- [138] M. Schoppers. Universal plans for reactive robots in unpredictable domains. *Proc., OJCAI-87, Menlo Park, CA:Morgan Kaufman*, 1987, pp. 1039-1046.

- [139] T. Shibata et T. Fukuda. Coordinative behavior in evolutionary multi-agent system by genetic algorithms. *Proc. IEEE Conf. On neural networks*, Vol. 1, 1993, pp. 209-214.
- [140] Squirt. Fast, cheap and out of control: A robot invasion of the solar system, R. A. Brooks and A.M. Flynn. MIT Artificial Intelligence Lab, Cambridge, MA, USA. *Journal of the British Interplanetary Society*, Vol. 42, 1989, pp 478-485.
- [141] Y. So et E.D. Durfee. An organizational self-design model for organizational change. *In working notes of the AAAI-93 workshop on AI and theories of groups and organizations*, 1993.
- [142] E. Sorouchyari. Mobile robot navigation: A neural network approach. *Proc. Art du Colloque Neuromimétique*, Ecole Polytechnique de Lausanne, 1989, pp. 159-175.
- [143] L. Steels. A case study in the behavior-oriented design of autonomous agents. *In proc of the 3<sup>rd</sup> simulation of adaptive behavior Conf.*, MIT Press, Cambridge Ma, 1994, pp. 445-452.
- [144] L. Steels. Cooperation between distributed agents through self-organization. *In European workshop on modelling autonomous agents in a multi-agent world*, 1990, pp. 175-195.
- [145] L. Steels. Cooperation between distributed agents through self organization. *Journal of robotics and autonomous systems*, MIT Press 6, 1989.
- [146] D. Stilwell et J. Bay. Toward the development of a material transport system using swarms of ant-like robots. *Proc. of IEEE Int. Conf. on robotics and automation*, Atlanta, Georgia, 1993, pp. 766-771.
- [147] P. Stone et M. Veloso. Multi-agent systems: A survey from a machine learning perspective. *Technical report, computer science department*, Carnegie Mellon university, no. CMU-CS-97-193, 1997.
- [148] K. Sugihara et I. Suzuki. Distributed algorithms for formation of geometric patterns with many mobile robots. *Journal of robotic systems by John Wiley & Sons, Inc*, 1996.
- [149] K. Sugihara et I. Suzuki. Distributed motion coordination of multiple mobile robots. *Proc. IEEE Int. Symposium on Intelligent Control*, Philadelphia, PA, September 1990, pp. 138-143.
- [150] R. Sutton et A. Barto. *Reinforcement learning : An introduction*. Cambridge, MA: MIT Press, 1998.
- [151] S. Suzuki, H. Asama, A. Uegaki, S. Kotosoka, T. Fujita, A. Matsumoto, H. Kaetsu et I. Endo. An infra-red sensory system with local communication for cooperative multiple mobile robots. *Proc. IEEE/RSJ Int. Conf. on Intelligent Robots and Systems*, 1995, pp. 220-225.
- [152] P. Szykarczyk et A. Masiowski. The fuzzy ARTMAP neural networks as a controller for the mobile robot. *Proc. 3<sup>rd</sup> Int. Symp. on Methods and Models in Automation and Robotics*, Miedzyzdroje, Poland, September 10-13, 1996, pp. 1201-1206.
- [153] M. Tan. Multi-agent reinforcement learning: Independent vs. cooperative agents, *Proc. of the Tenth International Conference on Machine Learning*, 1993, pp. 330-337.
- [154] S. Thrun. A lifelong learning perspective for mobile robot control. *Proc of Int. Conf on Intelligent Robots and Systems IROS'94*, Vol. 1, Munich, Germany, September 12-16, 1994, pp. 23-30.
- [155] S. Thrun et A. Schartz. Issues in using function approximation for reinforcement learning. *Proc. of Fourth Connectionist Models Summer School Lawrence Erlbaum Publisher, Hillsdale, NJ*, Dec, 1993.
- [156] S. Thrun. Efficient exploration in reinforcement learning. *Technical Report CMU-CS-92-102, Carnegie Mellon University, Computer Science Department*, Pittsburgh, PA, 1992.
- [157] C.F. Touzet. Neural reinforcement learning for behavior synthesis. *Int. J. of Robotics and Autonomous Systems*, Vol. 22, Elsevier, 1997, pp. 251-281.
- [158] X. Tu et D. Terzopoulos. Artificial fishes : Physics, locomotion, perception, behavior. *Computer graphics, SIGGRAPH 94 Conf. Proc.*, July 1994, pp. 43-50.
- [159] B. Tung et L. Kleinrock. Distributed control methods. *Proc. of the 2<sup>nd</sup> Int. symposium on high performance distributed computing*, 1993, pp. 206-215.
- [160] T Uyama et T. Fukuda. Knowledge acquisition and distributed decision making. *Proc. IEEE Int. Conf. on Rob. and Aut.* 1050-4729/93 \$3.00, 1993, pp.167-172.
- [161] T. Ueyama et T. Fukuda. Configuration of communication structure for distributed intelligent robot system. *Proc. ICRA*, Nice 1992, pp. 807-812.
- [162] C. Unsal. *Self-organisation in large populations of mobile robots*. Master of science in electrical engineering, Blacksburg, Virginia, 1993.
- [163] K. Vanlehn. Architectures for intelligence. *The 22<sup>nd</sup> Carnegie Mellon symposium on cognition*. Lawrence Erlbaum Associates, editor, 1991.
- [164] J. Wang. Operating primitives supporting traffic regulation and control of mobile robots under distributed robotic systems. *IEEE ICRA*, Nagoya, Japan 1995, pp. 1613-1618.
- [165] J. Wang. On sign-board based inter-robot communication in distributed robotic systems. *Proc. IEEE ICRA*, 1994, pp. 1045-1050.
- [166] P.K.C. Wang. Navigation strategies for multiple autonomous mobile robots moving in formation. *J. of Robotic Systems*, V. 8(2), by John Wiley & Sons Inc., 1991, pp.177-195.
- [167] C. Watkins. *Learning from delayed rewards*. Ph.D. Dissertation, Cambridge University, 1989.



- [168] S.T. Welstead. *Neural network and fuzzy logic applications in C/C++*. John Wiley & Sons Inc., Toronto, 1994.
- [169] B. Werger. Cooperation without deliberation : A minimal behavior-based approach to multi-robot teams. *Artificial intelligence* 110, 1999, pp. 293-320.
- [170] G. Werner et M. Dyer. Evolution of herding behavior in artificial animals. *Proc. simulation of adaptive behavior*, 1992.
- [171] S.D. Whitehead. Reinforcement learning for the adaptive control of perception and action. *Technical Report 406*, University of Rochester, February 1992.
- [172] J. Xiao, Z. Michalewicz, L. Zhang et K. Trojanowski. Adaptive evolutionary planner/navigator for mobile robots. *IEEE Trans. on Evolutionary Computation*, Vol. 1, No. 1, April 1997, pp. 18-28.
- [173] S. Yuta et S. Premvuti. Coordinating autonomous and centralized decision making to achieve cooperative behaviors between multiple mobile robots. *Proc. 1992 IEEE/RSJ Int. Conf. on Intelligent Robots and Systems*, Raleigh, NC, July 1992, pp. 1566-1574.
- [174] L.A. Zadeh. Fuzzy sets. *Information & Control*, vol. 8, 1965, pp. 338-353.
- [175] L.A. Zadeh. The Calculus of fuzzy If/Then rules. *AI Expert*, March 1992, pp. 23-27.

## Nos Publications

### Publications internationales

1. **O. Azouaoui**, H. Yazid, A. Farah et K. Achour. Behavioral multi-agent based control architecture for autonomous robotic systems. Soumis comme chapitre dans un livre sur les Systèmes Multi-Agents. Première révision acceptée, il est en cours d'une seconde révision.
2. **O. Azouaoui**, R. Benghanem, Chérifi, A. Farah et K. Achour. Reinforcement learning based group navigation approach for multiple autonomous robotic systems. A apparaître dans *Advanced robotics journal*, Vol.20 N°5, May 2006.
3. **O. Azouaoui** et A. Chohra. Soft computing based pattern classifiers for the obstacle avoidance behavior of Intelligent Autonomous Vehicles (IAV). *Applied Intelligence: The International J. of Artificial Intelligence, Neural Networks, and Complex Problem-Solving Technologies*, Vol. 16, N° 3, May/June 2002, pp. 249-271.

### Communications internationales

1. **O. Azouaoui**, A. Chérifi, R. Bensalem et A. Farah. Reinforcement learning based group navigation approach for multiple autonomous robotic systems. *Proc. IEEE Int. Conf. on Mechatronics and Automation (ICMA'05)*, 2005, pp.1539-1544.
2. **O. Azouaoui**, H. Yazid, F. Boulahlib et N. Chabane. Multi-agent approach based navigation system for mobile robots: Towards a coordination using traffic rules. *Proc. 17<sup>th</sup> Int. Conf. on Industrial and Engineering Applications of Artificial Intelligence and Expert Systems (IEA/AIE 2004)*, Montréal, 2004.
3. **O. Azouaoui** et A. Chohra. Pattern classifiers based on soft computing and their FPGA integration for intelligent behavior control of mobile robots. *Proc. IEEE 11<sup>th</sup> Int. Conf. on Advanced Robotics ICAR'2003*, Coimbra, Portugal, 2003, pp. 148-154.
4. A. Chohra et **O. Azouaoui**. Neural Networks (NN) using Genetic Algorithms (GA) and Gradient Back-Propagation (GBP) for an intelligent obstacle avoidance Behavior. *Proc. Int. Conf. on Advanced Robotics (ICAR03)*, Portugal, June 2003, pp. 917-922.
5. **O. Azouaoui**, M. Ouaz et A. Farah. Reinforcement Learning (RL) based collision avoidance approach for multiple Autonomous Robotic Systems (ARS). *Proc. IEEE 10<sup>th</sup> Int. Conf. on Advanced Robotics ICAR'2001*, Budapest, Hungary, 2001, pp. 561-566.
6. **O. Azouaoui**, M. Ouaz, A. Chohra, A. Farah et K. Achour. Fuzzy ArtMap Neural Network (FAMNN) based collision avoidance approach for Autonomous Robotic Systems (ARS). *Proc. IEEE 2<sup>nd</sup> Int. Workshop on Robot Motion and Control RoMoCo'01*, Bukowy Dworek, Poland, 2001, pp. 285-290.
7. **O. Azouaoui** et A. Chohra. Neural group navigation approach for Autonomous Robotic Systems (ARS). *Proc. Int. ICSC Symp. on Engineering of Intelligent Systems EIS'2000*, Scotland, U.K. 27-30 June 2000.

8. A. Chohra, R. Tiar et **O. Azouaoui**. *Fuzzy Motion Controller (FMC) for Intelligent Autonomous Vehicles (IAV)*. *Proc. Int. ICSC Symposium on Engineering of Intelligent Systems EIS'2000*, Scotland, U.K. 27-30 June 2000.

## Annexe A1 : Cinématique du robot considéré

### A1.1. Cinématique du robot considéré dans le repère du robot [12]

Trois référentiels sont considérés comme illustré sur la figure A1.1 et qui sont :

- $\Sigma_w$  : système de coordonnées du monde,
- $\Sigma_r$  : système de coordonnées du robot,
- $\Sigma_r^-$  : système de coordonnées instantané du robot.

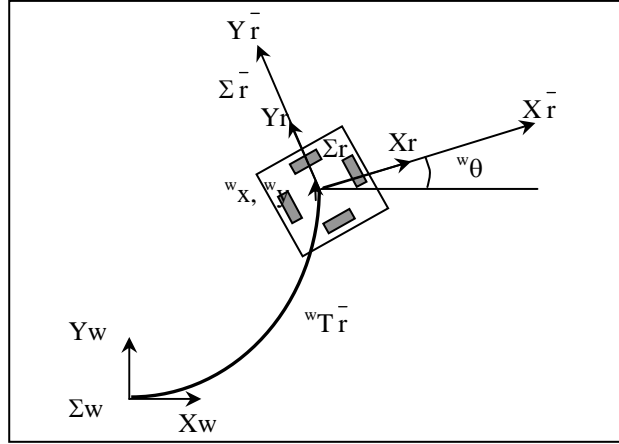


Figure A1.1. Systèmes de coordonnées.

Soit  $r$  le rayon des roues et  $R$  la distance du centre du robot au centre de la roue. La vitesse relative du robot exprimée dans le repère  $\Sigma_r^-$  est représentée par :

$$\begin{aligned} \bar{r} \dot{p} &= [\bar{r} \dot{x}, \bar{r} \dot{y}, \bar{r} \dot{\theta}]^T \\ \bar{r} \dot{p} &= \begin{pmatrix} 2k_1 k_2 r & 0 & 0 \\ 0 & 2k_1 k_2 r & 0 \\ 0 & 0 & k_3 \frac{r}{R} \end{pmatrix} \begin{pmatrix} V_1 \\ V_2 \\ V_3 \end{pmatrix} = \bar{r} J \dot{q} \end{aligned} \quad (A1.1)$$

Où  $k_i$  : rapport de réduction des engrenages<sup>1</sup>.

$\bar{r} J$  représente la matrice Jacobienne exprimée dans le repère  $\Sigma_r^-$  et  $\dot{q}$  représente la vitesse des actionneurs. Par conséquent, puisque  $\bar{r} J$  est une matrice diagonale, il est prouvé que le mouvement de rotation ou le mouvement de translation le long de chaque direction du robot par rapport au repère  $\Sigma_r^-$  est découplé et peut être entraîné par l'actionneur correspondant.

### A1.2. Cinématique avec transformation de coordonnées

La matrice de transformation (transformée de coordonnées homogènes à 2 dimensions) de  $\Sigma_w$  par rapport à  $\Sigma_r^-$  est donnée par :

<sup>1</sup> Les différents types d'engrenage utilisés sont :

- Engrenage roue et vis sans fin (worm gear),
- Engrenage droit (spur gear),
- Engrenage cylindrique (bevel gear).

$${}^w T_{\bar{r}} = \begin{pmatrix} \cos^w \theta & -\sin^w \theta & {}^w x \\ \sin^w \theta & \cos^w \theta & {}^w y \\ 0 & 0 & 1 \end{pmatrix}$$

puisque  $\Sigma_w$  et  $\Sigma_{\bar{r}}$  sont stationnaires, la dérivée de cette matrice comme fonction du temps est nulle.

$${}^w \dot{T}_{\bar{r}} = 0$$

pour la position  $p$ , l'équation de transformation suivante est définie par :

$${}^w p = {}^w T_{\bar{r}} \bar{r} p$$

en substituant les matrices de transformation et l'équation (A0.1) dans l'équation ci-dessus (de la dérivée), l'équation cinématique suivante est obtenue :

$${}^w \dot{p} = {}^w T_{\bar{r}} \bar{r} \dot{p} = {}^w T_{\bar{r}} \bar{r} J \dot{q} \quad (\text{A1.2})$$

où  ${}^w J$  est la matrice Jacobienne par rapport au repère  $\Sigma_w$  qui est obtenue comme suit :

$${}^w J = {}^w T_{\bar{r}} \bar{r} J = \begin{pmatrix} 2k_1 k_2 r \cos^w \theta & -2k_1 k_2 r \sin^w \theta & k_3 \frac{r}{R} {}^w x \\ 2k_1 k_2 r \sin^w \theta & 2k_1 k_2 r \cos^w \theta & k_3 \frac{r}{R} {}^w y \\ 0 & 0 & k_3 \frac{r}{R} \end{pmatrix}$$

d'un autre côté, l'équation cinématique inverse peut être obtenue de l'équation (A1.2) comme suit :

$$\dot{q} = {}^w T^{-1} {}^w \dot{p} = J^{-1} {}^w T_{\bar{r}}^{-1} \dot{p} \quad (\text{A1.3})$$

où

$${}^w J^{-1} = \bar{r} J^{-1} {}^w T_{\bar{r}}^{-1} = \begin{pmatrix} \frac{\cos^w \theta}{2k_1 k_2 r} & \frac{\sin^w \theta}{2k_1 k_2 r} & \frac{-{}^w x \cos^w \theta - {}^w y \sin^w \theta}{2k_1 k_2 r} \\ -\frac{\sin^w \theta}{2k_1 k_2 r} & \frac{\cos^w \theta}{2k_1 k_2 r} & \frac{{}^w x \sin^w \theta - {}^w y \cos^w \theta}{2k_1 k_2 r} \\ 0 & 0 & \frac{R}{k_3 r} \end{pmatrix}$$

la vitesse des actionneurs peut être obtenue en substituant dans l'équation (A1.3) la vitesse requise en translation ou en rotation ;  ${}^w \dot{x}, {}^w \dot{y}$  et  ${}^w \dot{\theta}$  la position et l'orientation actuelles ;  ${}^w x, {}^w y$  et  ${}^w \theta$  par rapport au repère  $\Sigma_w$ .

**Annexe A2 : Règles d'évitement de collisions.**

Les règles d'évitement utilisées pour permettre aux deux robots d'atteindre leurs cibles respectives sans collisions sont données sur la figure A2.1 [6].

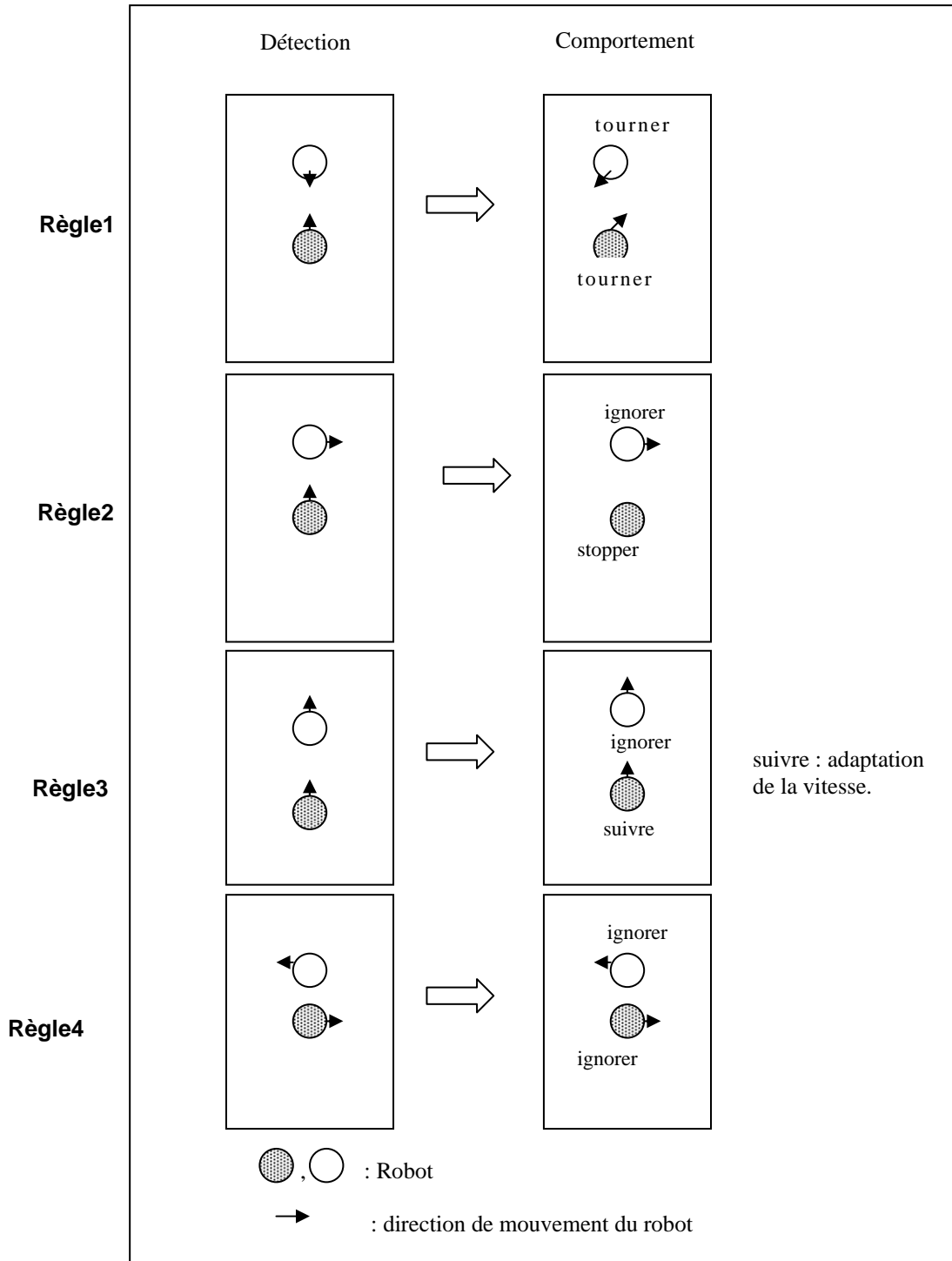


Figure A2.1 : Règles d'évitement de collisions entre deux robots.

- La règle1, étant la plus élémentaire, est appliquée lorsque deux robots se trouvent face à face et peuvent éviter la collision en tournant.

- La règle2 est appliquée quand les chemins des deux robots se croisent. Un des deux robots cède le passage à l'autre robot pour ensuite reprendre son chemin vers sa cible.
- La règle3 est appliquée dans le cas où un robot se trouve derrière un autre et qu'ils vont approximativement dans la même direction. Le premier robot ignore le suivant pendant que ce dernier ajuste sa vitesse pour ne pas le rattraper et engendrer une collision.
- La règle4 est appliquée lorsqu'il n'y a pas besoin d'évitement.

Les règles utilisées pour l'évitement d'obstacles sont au nombre de deux et sont illustrées sur la figure A2.2.

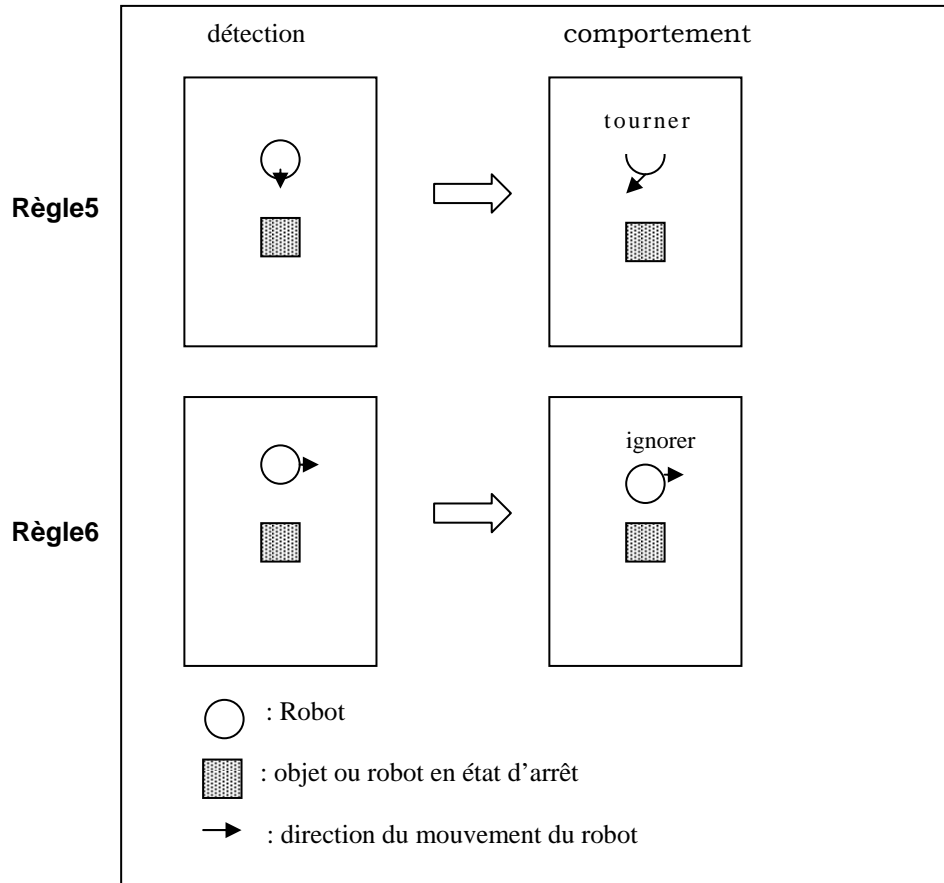


Figure A2.2 : Règles d'évitement de collisions entre un robot et un obstacle statique ou un robot en état d'arrêt.

## Annexe A3 : Formation d'un polygone.

Une méthode pour la formation d'un polygone de  $n$  cotés, où  $n \geq 3$  développée par Sugihara et al. [152, 153] est présentée dans cette annexe.

Les phases suivantes sont suivies afin d'obtenir un polygone :

Premièrement, on commande aux robots d'effectuer un cercle. Chaque robot  $R$  va reconnaître son voisin de gauche et son voisin de droite immédiats, notés  $R_{gauche}$  et  $R_{droite}$  respectivement. Ces voisins sont déterminés une fois seulement. La manière de les sélectionner est la suivante : chaque robot  $R$  considère le plus proche robot comme  $R_{gauche}$ , et  $R_{droite}$  est le plus proche robot dans la moitié du plan ne contenant pas  $R_{gauche}$ , et est déterminé par la ligne passant à travers  $R$  perpendiculaire au segment  $[R_{gauche}, R_{droite}]$  (Figure A3.1).

Les robots n'ont pas besoin d'avoir le même sens d'orientation [152].

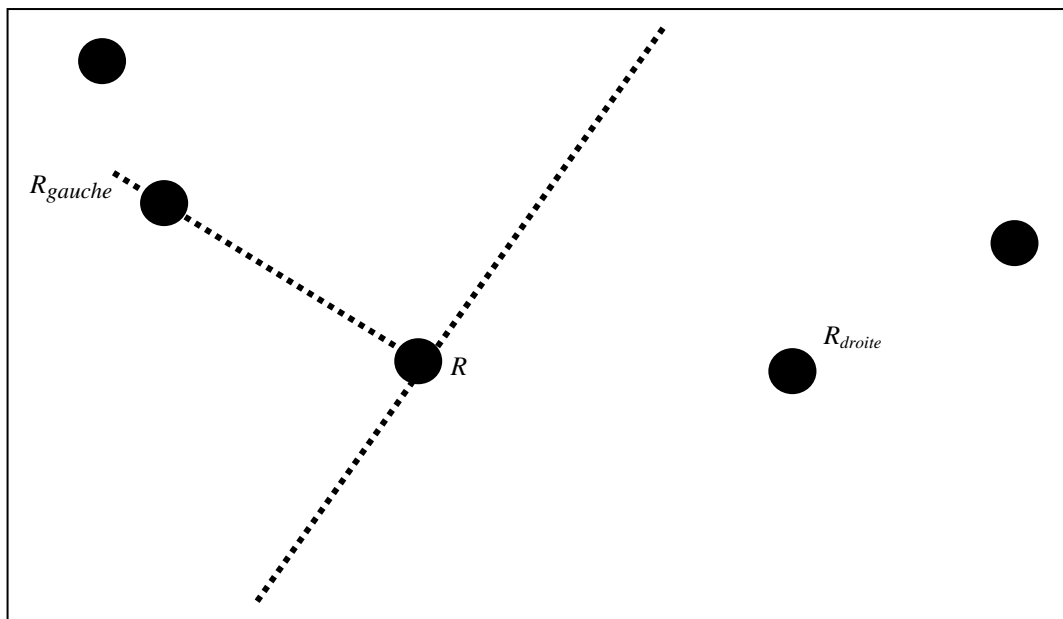


Figure A3.1 : Le choix de  $R_{gauche}$  et  $R_{droite}$ .

Lorsque  $R_{gauche}$  et  $R_{droite}$  sont déterminés par chaque robot  $R$ , l'utilisateur sélectionne  $n$  robots qui seront les sommets du polygone à former. Ensuite, on exécute les deux étapes suivantes :

1. Déplacer les  $n$  robots manuellement vers leurs positions finales désirées, de façon à ce que l'ordre dans lequel les  $n$  robots apparaissent dans le polygone sera le même dans l'approximation courante d'un cercle.

2. Permettre aux autres robots d'exécuter l'algorithme suivant :

Soient  $R_1$  et  $R_2$  deux robots constituant deux sommets du polygone (ces sommets n'exécutent pas les deux étapes précédentes). Ces deux étapes permettent de transformer chaque forme curviligne de robots en une séquence de robots entre  $R_1$  et  $R_2$  (Figure A3.2).

Les robots sont distribués uniformément dans la ligne de segment entre  $R_1$  et  $R_2$ .



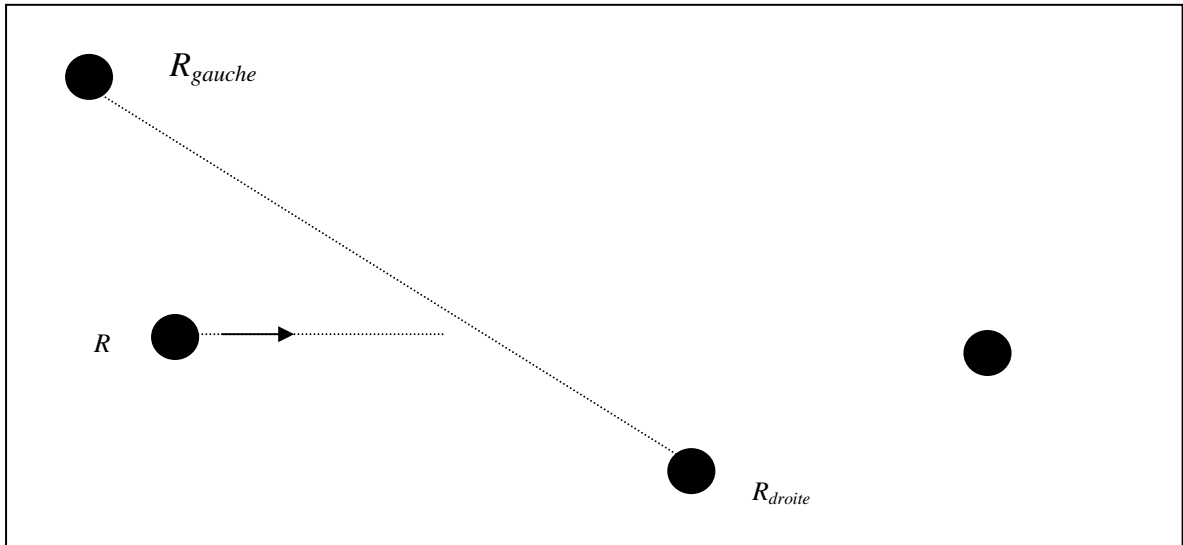
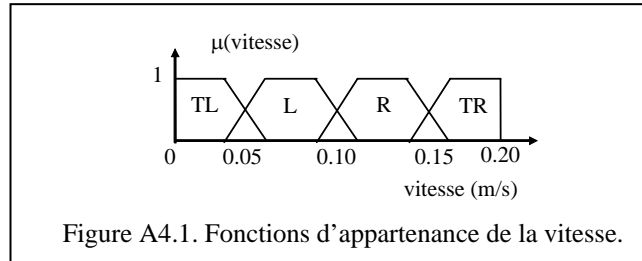


Figure A3.2 : Déplacement des robots compris entre  $R_{gauche}$  et  $R_{droite}$ .

**Annexe A4. Fonctions d'appartenance**

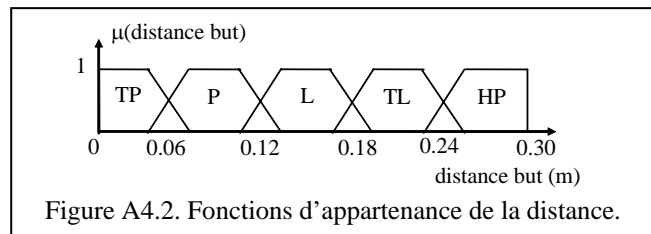
*Fonction d'appartenance de la vitesse*

La vitesse maximale du robot est de 0.2 m/s et la taille choisie de l'intervalle de vitesse est de 0.05 m/s. par conséquent, les intervalles de la vitesse sont  $V_i \in \{V_1, V_2, V_3, V_4\}$  où  $V_1 = [0.000, 0.05]$ ,  $V_2 = [0.05, 0.1]$ ,  $V_3 = [0.1, 0.15]$  et  $V_4 = [0.15, 0.2]$ . Ces quatre intervalles discrets de la vitesse conduisent à quatre variables linguistiques comme montré sur la figure A4.1 où TL, L, R et TR correspondent respectivement à Très Lent, Lent, Rapide et Très Rapide.

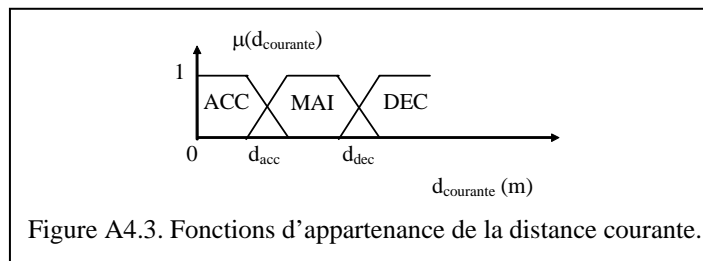


*Fonction d'appartenance de la distance*

Afin de prendre en charge les incertitudes résultant en un "mapping" flou précis, les intervalles discrets doivent être plus grands que les erreurs de mesure des capteurs. En effet, la distance but doit être donnée par un système contrôlant les comportements intelligents de la navigation où le système de perception est basé sur des IR. L'intervalle de la distance but est de  $\Delta d = 0.06$  m. Par conséquent, la distance but peut être dans les intervalles suivants  $D_{buti} \in \{D_{but1}, D_{but2}, D_{but3}, D_{but4}, D_{but5}\}$  où  $D_{but1} = [0.00, 0.06]$ ,  $D_{but2} = [0.06, 0.12]$ ,  $D_{but3} = [0.12, 0.18]$ ,  $D_{but4} = [0.18, 0.24]$ ,  $D_{but5} = [0.24, 0.30]$ . Ces cinq intervalles discrets de la distance but conduisent à cinq variables linguistiques comme montré sur la figure A4.2 où TP, P, L, TL et HP correspondent respectivement à Très Prêt, Prêt, Loin, Très Loin et Hors Portée.



La fonction d'appartenance de la distance courante est donnée par la figure A4.3 où ACC, MAI et DEC correspondent à ACCélérer, MAIntenir et DECélérer, respectivement.



**Annexe A5. Distances d'Accélération/Décélération pour Chaque Combinaison des Vitesse Finale, Vitesse Initiale et Distance-But**

**Tableau A5.1. Distances d'Accélération/Décélération pour Chaque Combinaison des Vitesse Finale, Vitesse Initiale et Distance-But.**

Notons que ces expériences sont établies pour les vitesses finales désirées:

$$V_1 = 0.05 \text{ m/s}, V_2 = 0.1 \text{ m/s}, V_3 = 0.15 \text{ m/s}, \text{ et } V_4 = 0.2 \text{ m/s}.$$

$V_{\text{initiale}}$ (m/s)	$V_{\text{finale}}$ (m/s)	$d_{\text{but}}$ (m)	$d_{\text{acc}}$ (m)	$d_{\text{déc}}$ (m)
0.05	0.05	0.06	0.01875	0.04075
0.05	0.10	0.06	0.01875	0.04475
0.05	0.15	0.06	0.01875	0.05075
0.05	0.20	0.06	0.01875	0.05875
0.10	0.05	0.06	0.01705	0.04105
0.10	0.10	0.06	0.01705	0.04305
0.10	0.15	0.06	0.01705	0.05105
0.10	0.20	0.06	0.01705	0.05905
0.15	0.05	0.06	0.00875	0.04075
0.15	0.10	0.06	0.00875	0.04475
0.15	0.15	0.06	0.00875	0.05075
0.15	0.20	0.06	0.00875	0.05875
0.20	0.05	0.06	0	0.04
0.20	0.10	0.06	0	0.044
0.20	0.15	0.06	0	0.05
0.20	0.20	0.06	0	0.058
0.05	0.05	0.12	0.01875	0.10075
0.05	0.10	0.12	0.01875	0.10475
0.05	0.15	0.12	0.01875	0.11075
0.05	0.20	0.12	0.01875	0.11875
0.10	0.05	0.12	0.01705	0.10105
0.10	0.10	0.12	0.01705	0.10305
0.10	0.15	0.12	0.01705	0.11105
0.10	0.20	0.12	0.01705	0.11905
0.15	0.05	0.12	0.00875	0.10075
0.15	0.10	0.12	0.00875	0.10475
0.15	0.15	0.12	0.00875	0.11075
0.15	0.20	0.12	0.00875	0.11875
0.20	0.05	0.12	0	0.1
0.20	0.10	0.12	0	0.104
0.20	0.15	0.12	0	0.11
0.20	0.20	0.12	0	0.12
0.05	0.05	0.18	0.01875	0.16075
0.05	0.10	0.18	0.01875	0.16475
0.05	0.15	0.18	0.01875	0.17075
0.05	0.20	0.18	0.01875	0.17875
0.10	0.05	0.18	0.01705	0.16105
0.10	0.10	0.18	0.01705	0.16305
0.10	0.15	0.18	0.01705	0.17105
0.10	0.20	0.18	0.01705	0.17905
0.15	0.05	0.18	0.00875	0.16075
0.15	0.10	0.18	0.00875	0.16475
0.15	0.15	0.18	0.00875	0.17075
0.15	0.20	0.18	0.00875	0.17875

$v_{\text{initiale}}$ (m/s)	$v_{\text{finale}}$ (m/s)	$d_{\text{but}}$ (m)	$d_{\text{acc}}$ (m)	$d_{\text{déc}}$ (m)
0.20	0.05	0.18	0	0.06
0.20	0.10	0.18	0	0.164
0.20	0.15	0.18	0	0.17
0.20	0.20	0.18	0	0.18
0.05	0.05	0.24	0.01875	0.22075
0.05	0.10	0.24	0.01875	0.22475
0.05	0.15	0.24	0.01875	0.23075
0.05	0.20	0.24	0.01875	0.23875
0.10	0.05	0.24	0.01705	0.22105
0.10	0.10	0.24	0.01705	0.22305
0.10	0.15	0.24	0.01705	0.23105
0.10	0.20	0.24	0.01705	0.23905
0.15	0.05	0.24	0.00875	0.22075
0.15	0.10	0.24	0.00875	0.22475
0.15	0.15	0.24	0.00875	0.23075
0.15	0.20	0.24	0.00875	0.23875
0.20	0.05	0.24	0	0.22
0.20	0.10	0.24	0	0.224
0.20	0.15	0.24	0	0.23
0.20	0.20	0.24	0	0.24
0.05	0.05	0.3	0.01875	0.28075
0.05	0.10	0.3	0.01875	0.28475
0.05	0.15	0.3	0.01875	0.29075
0.05	0.20	0.3	0.01875	0.29875
0.10	0.05	0.3	0.01705	0.28105
0.10	0.10	0.3	0.01705	0.28305
0.10	0.15	0.3	0.01705	0.29105
0.10	0.20	0.3	0.01705	0.29905
0.15	0.05	0.3	0.00875	0.28075
0.15	0.10	0.3	0.00875	0.28475
0.15	0.15	0.3	0.00875	0.29075
0.15	0.20	0.3	0.00875	0.29875
0.20	0.05	0.3	0	0.28
0.20	0.10	0.3	0	0.284
0.20	0.15	0.3	0	0.29
0.20	0.20	0.3	0	0.3

**Annexe A6. Contrôleur flou de mouvements et base de règles floues**

**A6.1. Contrôleur flou de mouvements**

*A6.1.1. Fuzzification*

L'opération de fuzzification calcule le degré des données mesurées membres des trois fonctions d'appartenance pour les trois variables d'entrée, i.e., pour  $V_{finale}$ , on obtient  $\{\mu_{TL}(V_{finale}), \mu_L(V_{finale}), \mu_R(V_{finale}), \mu_{TR}(V_{finale})\}$  représentant les degrés d'appartenance des sous-ensembles flous TL (Très Lent), L (Lent), R (Rapide), et TR (Très Rapide), respectivement (voir Annexe A4). Pour la distance, on obtient  $\{\mu_{TP}(D_{but}), \mu_P(D_{but}), \mu_L(D_{but}), \mu_{TL}(D_{but}), \mu_{HP}(D_{but})\}$  représentant les degrés d'appartenance des sous-ensembles flous TP (Très Proche), P (Proche), L (Loin), TL (Très Loin) et HP (Hors Portée), respectivement (voir Annexe A4).

*A6.1.2. Expérimentations et règles floues de contrôle de mouvement*

Des expériences ont été effectuées et ont donné les valeurs des distances d'accélération/décélération, pour chaque combinaison de la vitesse finale, la vitesse initiale, et la distance-but données en Annexe A5. A partir des résultats de ces expériences, les règles floues de contrôle de mouvements sont dérivées et sont de la forme [48] :

SI 'vitesse actuelle est  $V_i$  et la distance but est  $D_{but_i}$  et la distance actuelle est  $D_{actuelle_i}$ '

ALORS 'C $\tilde{A}$  i.e., changer la vitesse à  $v_n$  avec une accélération/décélération maximale ou garder la même vitesse'.

où C $\tilde{A}$  correspond au vecteur commande flou donné par :

$$C\tilde{A} = [ \mu(v_{actuelle_m}, d_{but_m}, d_{actuelle_m})(CA_1), \dots, \mu(v_{actuelle_m}, d_{but_m}, d_{actuelle_m})(CA_k), \dots, \mu(v_{actuelle_m}, d_{but_m}, d_{actuelle_m})(CA_6) ], \quad (V.1)$$

où  $\mu(v_{actuelle_m}, d_{but_m}, d_{actuelle_m})(CA_k)$  représente le degré d'appartenance avec  $m=1$  ou 2 [48].

*A6.1.3. Inférence floue*

L'inférence floue est réalisée par un algorithme de recherche et d'appariement, utilisant les opérateurs MIN-MAX, développé dans [51]. Pour chaque situation spécifique du robot, la correspondance des valeurs des vitesses initiales et des distances-buts sur les intervalles discrets forme les sous-ensembles flous  $\tilde{v}_{actuelle}$ ,  $\tilde{d}_{but}$  et  $\tilde{d}_{actuelle}$  exprimés par :

$$\tilde{v}_{actuelle} = \{\mu_1(v_{actuelle}), \mu_2(v_{actuelle})\}, \quad (V.2)$$

$$\tilde{d}_{but} = \{\mu_1(d_{but}), \mu_2(d_{but})\}, \quad (V.3)$$

$$\tilde{d}_{actuelle} = \{\mu_1(d_{actuelle}), \mu_2(d_{actuelle})\}, \quad (V.4)$$

où  $\mu_m(v_{actuelle})$ ,  $\mu_m(d_{but})$  et  $\mu_m(d_{actuelle})$  sont les degrés des fonctions d'appartenance de la vitesse actuelle, de la distance but et de la distance actuelle. Avec cette description, on peut avoir huit conditions possibles correspondant à huit règles floues. Ainsi, le niveau de certitude de chaque triplet condition  $\mu_1$ ,  $\mu_2$ , ..., ou  $\mu_8$  est obtenu en utilisant l'opérateur MIN :

$$\begin{aligned} \mu_{cond}(\mu(v_{actuelle_1}, d_{but_1}, d_{actuelle_1})) &= MIN(\mu_1(v_{actuelle}), \mu_1(d_{but}), \mu_1(d_{actuelle})) = \mu_1, \\ \mu_{cond}(\mu(v_{actuelle_1}, d_{but_1}, d_{actuelle_2})) &= MIN(\mu_1(v_{actuelle}), \mu_1(d_{but}), \mu_2(d_{actuelle})) = \mu_2, \\ &\dots \\ \mu_{cond}(\mu(v_{actuelle_2}, d_{but_2}, d_{actuelle_2})) &= MIN(\mu_2(v_{actuelle}), \mu_2(d_{but}), \mu_2(d_{actuelle})) = \mu_8, \end{aligned} \quad (V.5)$$

où *cond* représente l'ensemble flou de conditions qui est écrit comme suit :

$$cond = \{\mu_1, \mu_2, \dots, \mu_8\}. \quad (V.6)$$

Chaque triplet possible de condition est associé à un vecteur de commande floue. Ainsi, la certitude de chaque situation est obtenue par les opérateurs MAX-MIN comme suit :

$$\begin{aligned} \mu_{CA1} = \text{MAX}\{ & \text{MIN}(\mu_1, \mu(v_{actuelle_1}, d_{but_1}, d_{actuelle_1})(CA_1)), \\ & \text{MIN}(\mu_2, \mu(v_{actuelle_1}, d_{but_1}, d_{actuelle_2})(CA_1)), \\ & \dots \\ & \text{MIN}(\mu_8, \mu(v_{actuelle_2}, d_{but_2}, d_{actuelle_2})(CA_1)) \\ & \}, \\ \mu_{AC6} = \text{MAX}\{ & \dots \\ & \text{MIN}(\mu_1, \mu(v_{actuelle_1}, d_{but_1}, d_{actuelle_1})(CA_6)), \\ & \text{MIN}(\mu_2, \mu(v_{actuelle_1}, d_{but_1}, d_{actuelle_2})(CA_6)), \\ & \dots \\ & \text{MIN}(\mu_8, \mu(v_{actuelle_2}, d_{but_2}, d_{actuelle_2})(CA_6)) \\ & \}. \end{aligned} \quad (V.7)$$

La collection de situations forme le vecteur final de commande floue  $\tilde{CA}$  :

$$\tilde{CA} = \{ \mu_{CA1}(CA_1), \dots, \mu_{CAk}(CA_k), \dots, \mu_{CA6}(CA_6) \} \quad (V.8)$$

#### A6.1.4. Défuzzification

L'opération MAX est utilisée pour la défuzzification afin de donner la commande finale.

$$CA_{finale} = \text{MAX}\{\mu_{CA1}(CA_1), \dots, \mu_{CAk}(CA_k), \dots, \mu_{CA6}(CA_6)\} \quad (V.9)$$

### A6.2. Base de règles floues

Notons que la base de règles floues est établie pour les vitesses finales désirées:

$V_1 = 0.05 \text{ m/s}$ ,  $V_2 = 0.1 \text{ m/s}$ ,  $V_3 = 0.15 \text{ m/s}$ , et  $V_4 = 0.2 \text{ m/s}$ .

$X_{CA1} = V_{finale}$ ,

$X_{CA2} = V_{initiale}, V_{courante}$

$X_{CA3} = d_{but}, d_{courante}$

$$\begin{aligned} & V_{finale} = V_1 \\ \text{Si } V_{courante} \text{ est TL} \quad \text{et } d_{but} \text{ est TP} \quad \text{et } d_{courante} \text{ est ACC} \quad \text{Alors } CA_3 \\ & \quad \text{et } d_{courante} \text{ est MAI} \quad \text{Alors } CA_3 \\ & \quad \text{et } d_{courante} \text{ est DEC} \quad \text{Alors } CA_6 \\ \text{et } d_{but} \text{ est P} \quad \text{et } d_{courante} \text{ est ACC} \quad \text{Alors } CA_3 \\ & \quad \text{et } d_{courante} \text{ est MAI} \quad \text{Alors } CA_3 \\ & \quad \text{et } d_{courante} \text{ est DEC} \quad \text{Alors } CA_6 \\ \text{et } d_{but} \text{ est L} \quad \text{et } d_{courante} \text{ est ACC} \quad \text{Alors } CA_3 \\ & \quad \text{et } d_{courante} \text{ est MAI} \quad \text{Alors } CA_3 \\ & \quad \text{et } d_{courante} \text{ est DEC} \quad \text{Alors } CA_6 \\ \text{et } d_{but} \text{ est TL} \quad \text{et } d_{courante} \text{ est ACC} \quad \text{Alors } CA_3 \\ & \quad \text{et } d_{courante} \text{ est MAI} \quad \text{Alors } CA_3 \\ & \quad \text{et } d_{courante} \text{ est DEC} \quad \text{Alors } CA_6 \\ \text{et } d_{but} \text{ est HP} \quad \text{et } d_{courante} \text{ est ACC} \quad \text{Alors } CA_3 \\ & \quad \text{et } d_{courante} \text{ est MAI} \quad \text{Alors } CA_3 \\ & \quad \text{et } d_{courante} \text{ est DEC} \quad \text{Alors } CA_6 \end{aligned}$$











