

REPUBLIQUE ALGERIENNE DEMOCRATIQUE ET POPULAIRE

Ministère de l'Enseignement Supérieur et de la Recherche

Scientifique



Ecole Nationale Polytechnique  
Département d'Automatique  
Laboratoire de commande des processus



Mémoire de projet de fin d'études

En vue de l'obtention du diplôme d'ingénieur d'état en  
Automatique

## Commande d'une formation de robots mobiles

Mr. BOUKHEBOUZ Bassem, Mr. LAOUATI Salah Eddine

Sous la direction de : M. Mohammed TADJINE

Professeur

M. Massoud CHAKIR

Docteur

Présenté et soutenu publiquement le 20/06/2017

### Composition du Jury :

Président : Mr. Mohamed Seghir BOUCHERIT Professeur ENP

Promoteurs : Mr. Mohamed TADJINE Professeur ENP

Mr. Messaoud CHAKIR MCB ENP

Examineurs : Mr. Lahcene ABDELOUEL MAA ENP

ENP (2017)



REPUBLIQUE ALGERIENNE DEMOCRATIQUE ET POPULAIRE

Ministère de l'Enseignement Supérieur et de la Recherche

Scientifique



Ecole Nationale Polytechnique

Département d'Automatique

Laboratoire de commande des processus



Mémoire de projet de fin d'études

En vue de l'obtention du diplôme d'ingénieur d'état en

Automatique

## Commande d'une formation de robots mobiles

Mr. BOUKHEBOUZ Bassem, Mr. LAOUATI Salah Eddine

Sous la direction de : M. Mohammed TADJINE

Professeur

M. Massoud CHAKIR

Docteur

Présenté et soutenu publiquement le 20/06/2017

### Composition du Jury :

Président : Mr. Mohamed Seghir BOUCHERIT Professeur ENP

Promoteurs : Mr. Mohamed TADJINE Professeur ENP

Mr. Messaoud CHAKIR MCB ENP

Examineurs : Mr. Lahcene ABDELOUEL MAA ENP

ENP (2017)

# DEDICACE

Je commence par rendre grâce à dieu et sa bonté, pour la patience et le courage qu'il m'a donné pour arriver à ce stade.

Avec tout mon amour éternel et avec l'intensité de mes émotions. Je dédie ce mémoire à l'âme de mon cher père ALI BOUKHEBOUZ, J'espère que tu habites les paradis.

A ma mère NADIA. Je n'oublie jamais tes sacrifices, l'amour qui tu m'as donné, Aucune dédicace ne pourrait exprimer mon respect, ma considération et mes profonds sentiments envers toi maman. Je prie le bon Dieu de te bénir, de veiller sur toi, en espérant que tu seras toujours fier de moi.

À mon très cher MOUAADH, qui n'a jamais cessé d'être pour moi un exemple de courage et de générosité. Les mots ne suffisent guère pour exprimer l'attachement, l'amour et l'affection que je porte pour toi.

À tous les membres de ma famille, petits et grands. Veuillez trouver dans ce travail l'expression de mon affection.

À mes amis KHALED & SIFOU, en témoignage de l'amitié et la fraternité qui nous uni et les conseils que vous m'avez donnés.

À mes amis MOH, WAIL, AKRAM, RAMI, 7IMDA et ANES ..., en témoignage de l'amitié, la fraternité et des souvenirs de tous les moments que nous avons passés ensemble.

À mon binôme SALAH, en témoignage des nuits blanches et des disputes qui nous uni et aux souvenirs mémorables, je suis très reconnaissant aux efforts que vous avez faits.

À tous ceux que j'aime et qui m'aiment

*BOUKHEBOUZ BASSEM*

# DEDICACE

Je dédie ce modeste travail :

À mes très chers Parents ABD ELKRIM & AKILA qui Grâce à leurs tendres encouragements et leurs grands sacrifices, ils ont pu créer le climat affectueux et propice à la poursuite de mes études. Aucune dédicace ne pourrait exprimer mon respect, ma considération et mes profonds sentiments envers eux. Je prie le bon Dieu de les bénir, de veiller sur eux, en espérant qu'ils seront toujours fiers de moi.

À mes très chères sœurs SABAH, SOUAD, HOUDA, MARIAMA et HANA et ma belle-sœur NABILA mes très chers frères NABIL, KHALED, ATMANE et ISMAIL et mes beaux-frères MOURAD et ABD EL KADER, qui n'ont cessé d'être pour moi des exemples de persévérance, de courage et de générosité. Les mots ne suffisent guère pour exprimer l'attachement, l'amour et l'affection que je porte pour vous.

À tous les membres de ma famille, petits et grands. Veuillez trouver dans ce travail l'expression de mon affection.

À mes chers amis, en témoignage de l'amitié qui nous uni et des souvenirs de tous les moments que nous avons passés ensemble.

À mon binôme BASSEM, en témoignage des nuits blanches et des disputes qui nous uni et aux souvenirs mémorables, je suis très reconnaissant aux efforts que vous avez faits.

À tous ceux qui ont participé de près ou de loin pour l'élaboration de ce travail.

**LAOUATI Salah Eddine**

# ***REMERCIEMENTS***

La réalisation de ce mémoire a été possible grâce à ALLAH le tout puissant, pour le courage et la patience qu'il nous a donné pour accomplir ce travail.

Nous souhaiterions adresser nos remerciements les plus sincères aux personnes qui nous ont apportés leur aide et qui ont contribué à l'élaboration de ce mémoire. Nos profonds remerciements s'adressent à nos encadrants ; Monsieur TADJINE Mohamed et Monsieur CHAKIR Messaoud qui se sont toujours montrés disponibles et à l'écoute tout au long du travail. Nous vous remercions aussi pour vos conseils. Veuillez trouver ici, l'expression de notre gratitude et notre grande estime.

Nos remerciements s'adressent aux membres du jury ; Mr BOUCHERIT Mohamed Seghir qui a bien voulu présider ce jury, à Mr ABDELOUEL Lahcene pour avoir bien voulu examiner ce travail. Vous nous faites un grand honneur en acceptant de juger ce travail.

Nous devons un remerciement à tous les enseignants du département Automatique qui nous ont fourni les outils nécessaires à la réussite de nos études universitaires.

Une pensée particulière est adressée aux étudiants qui nous ont côtoyés quotidiennement durant nos années d'études au département en nous apportant leurs supports moral et intellectuel tout au long de notre mémoire. Enfin, nous tenons à remercier chaleureusement, tous nos proches, amis et tous ceux qui, de près ou de loin, nous ont toujours soutenus et encouragés au cours de la réalisation de ce mémoire, pour leur confiance, leur support inestimable et leurs sollicitudes. À toutes ces personnes, nous présentons nos remerciements, notre respect et notre gratitude.

## ملخص

يندرج هذا العمل في إطار التحكم في الأنظمة المتعددة الروبوتات ذلك بهدف تحقيق تشكيل هندسي للروبوتات، أي كيف نضع قواعد التحكم في حركة المركبات من أجل القيام بمهام معينة. من أجل ذلك وضعنا استراتيجيات تعاونية الأولى المتمثلة في استراتيجية القائد والتابع، معززا بإضافة المنطق الغامض، والثانية المعتمدة على حقول الكمون، وأخيرا تم تطوير خوارزمية هجينة مبنية على أساس الاستراتيجيتين السابقتين. هذا العمل ينتهي بمحاكاة ثلاثية الأبعاد في البيئة المسماة

**الكلمات المفتاحية :** روبوت متحرك، التحكم التعاوني، القائد والتابع، حقول الكمون، المنطق المبهم، V-REP.

## Abstract

The work presented in this thesis concerns the multi-robot cooperation, and consists in controlling a group of mobile robots to create a formation, i.e. how to define the motion control for each robot. In order to carry out tasks in a coordinated way, and especially how to keep the formation: assured a geometrical configuration during the navigation of the group. For this, we have developed cooperative strategies. The first is the leader-follower approach, enhanced by the addition of fuzzy logic, the second based on potential fields, and finally developing a hybrid algorithm based on the two previous methods. The work ends with a 3D simulation under the V-REP environment.

**Keywords:** mobile robot, cooperative control, leader-follower, potential fields, fuzzy logic, V-REP.

## Résumé

Le travail présenté dans ce mémoire s'inscrit dans le cadre de la coopération multi-robots, et consiste à contrôler un groupe de robots mobiles pour créer une formation, c'est-à-dire comment définir les commandes de mouvement pour chaque robot du groupe afin de réaliser des tâches d'une manière coordonnée, et surtout comment garder la formation : assuré une configuration géométrique lors de la navigation du groupe. Pour cela nous avons élaboré des techniques de commandes coopératives, la première c'est l'approche leader-follower améliorée par l'ajout de la logique floue, la deuxième est basée sur les champs de potentiel et en fin présenter un algorithme hybride basé sur les deux méthodes précédentes. Le travail s'achève par une simulation 3D sous l'environnement V-REP.

**Mots clés :** robot mobile, commande coopérative, leader-follower, champs de potentiel, logique floue, V-REP.

# SOMMAIRE

LISTE DES FIGURES

LISTE DES SYMBOLES

INTRODUCTION GENERALE .....	13
<b>1 FORMATION DE GROUPE DE ROBOTS, ETAT DE L'ART .....</b>	<b>15</b>
<b>1.1. Introduction .....</b>	<b>15</b>
<b>1.1 Generalites sur les formations .....</b>	<b>17</b>
<b>1.2 Types des robots dans une formation .....</b>	<b>19</b>
1.2.1 robots terrestres .....	19
1.2.2 satellites .....	19
1.2.3 robots aquatiques .....	20
1.2.4 robots volants .....	21
<b>1.3 Architectures de commande .....</b>	<b>22</b>
1.3.1 Centralisée .....	22
1.3.2 Décentralisée .....	22
1.3.3 Hiérarchique.....	23
1.3.4 Hybride.....	24
<b>1.4 Stratégies de commande.....</b>	<b>24</b>
1.4.1 Approches comportementales et champs de potentiels.....	24
1.4.2 Approches structures virtuelle.....	25
1.4.3 Approches leader-follower .....	25
1.4.4 autres approches .....	27
<b>1.5 Evaluation des formations .....</b>	<b>27</b>
1.5.1 avantages d'une formation de robots.....	27
1.5.2 inconvénients d'une formation de robots .....	27
<b>1.6 Applications .....</b>	<b>28</b>
1.6.1 Domaine militaire.....	28
1.6.2 Application civiles.....	28
<b>1.7 Conclusion .....</b>	<b>29</b>

<b>2</b>	<b>COMMANDE COOPERATIVE D'UNE FORMATION DE ROBOTS MOBILES .....</b>	<b>31</b>
<b>2.1</b>	<b>Introduction .....</b>	<b>31</b>
<b>2.2</b>	<b>Commande des formations .....</b>	<b>31</b>
2.2.1	Stabilité .....	31
2.2.2	Commandabilité .....	32
<b>2.3</b>	<b>Approche leader-follower .....</b>	<b>33</b>
2.3.1	Modélisation de la formation leader-follower .....	33
2.3.2	Architecture de l'approche leader-follower .....	35
2.3.3	Evitement d'obstacle .....	36
<b>2.4</b>	<b>Approche comportemental méthode champs de potentiel .....</b>	<b>37</b>
2.4.1	Définition .....	38
2.4.2	Planification de trajectoire par la méthode des champs potentiels .....	38
2.4.3	Potentiel attractif .....	38
2.4.4	Potentiel répulsif .....	40
2.4.5	combinaison des deux comportements attractif et répulsif .....	42
2.4.6	Génération de la formation .....	42
2.4.7	Application de la méthode de champs de potentiels pour la commande d'une formation .....	43
<b>2.5</b>	<b>Approche proposée leader-follower / logique floue .....</b>	<b>43</b>
2.5.1	Synthèse de la loi de commande leader-follower /logique floue .....	43
<b>2.6</b>	<b>Approche hybride .....</b>	<b>44</b>
2.6.1	Limitations des deux approches .....	45
2.6.2	fonctionnement de l'approche hybride .....	45
<b>2.7</b>	<b>Conclusion .....</b>	<b>46</b>
<b>3</b>	<b>TESTS ET RESULTATS DE SIMULATION .....</b>	<b>48</b>
<b>3.1</b>	<b>Introduction .....</b>	<b>48</b>
<b>3.2</b>	<b>Commande d'une formation en utilisant l'approche leader-follower .....</b>	<b>48</b>
3.2.1	Maintien de la formation .....	48
3.2.2	Suivi de trajectoire .....	51
3.2.3	Evitement des obstacles .....	53

<b>3.3</b>	<b>Commande d'une formation en utilisant l'approche comportementale champs de potentiel</b>	<b>54</b>
3.3.1	Maintien de la formation	54
3.3.2	Suivi de trajectoire	55
3.3.3	Evitement des obstacles	57
<b>3.4</b>	<b>Commande d'une formation en utilisant l'approche leader-follower/logique floue</b>	<b>58</b>
<b>3.5</b>	<b>Etude comparative entre l'approche leader-follower et l'approche champs de potentiel</b>	<b>60</b>
<b>3.6</b>	<b>Commande d'une formation en utilisant l'approche hybride</b>	<b>62</b>
<b>3.7</b>	<b>Conclusion</b>	<b>65</b>
<b>4</b>	<b>SIMULATION SOUS L'ENVIRONNEMENT V-REP</b>	<b>67</b>
<b>4.1</b>	<b>Introduction</b>	<b>67</b>
<b>4.2</b>	<b>environnement V-REP</b>	<b>67</b>
<b>4.3</b>	<b>Interface MATLAB/V-REP</b>	<b>72</b>
4.3.1	fonctions utilisees	72
<b>4.4</b>	<b>robot mobile « Pioneer p3dx »</b>	<b>75</b>
<b>4.5</b>	<b>Simulation</b>	<b>77</b>
<b>4.6</b>	<b>Conclusion</b>	<b>80</b>
	<b>CONCLUSION GENERALE</b>	<b>82</b>
	<b>BIBLIOGRAPHIE</b>	<b>85</b>
	<b>ANNEXE A</b>	<b>87</b>
	<b>ANNEXE B</b>	<b>95</b>
	<b>ANNEXE C</b>	<b>99</b>

## LISTE DES FIGURES

Figure 1-1 : Exemples de systèmes biologiques exhibant des comportements coopératifs.....	15
Figure 1-2 : Troupe d'oiseaux volant en formation V.....	16
Figure 1-3 : Exemples de formation.....	18
Figure 1-4 : Différents types de formation : (a) colonne, (b) ligne, (c) V inversé.....	18
Figure 1-5 : Exemple de robots terrestres.....	19
Figure 1-6 : Constellation de satellite de type GPS ou téléphone.....	20
Figure 1-7 : Drone MALE PREDATOR.....	21
Figure 1-8 : Commande centralisée.....	22
Figure 1-9 : Commande décentralisée.....	23
Figure 1-10 : Commande hiérarchique.....	23
Figure 1-11 : (a) leader follower type $L - \varphi$ , (b) leader follower type $L - L$ .....	26
Figure 2-1: Configuration Leader-follower.....	33
Figure 2-2: Formation triangle de six robots mobiles.....	35
Figure 2-3 : Architecture a quatre niveaux de l'approche leader-follower.....	36
Figure 2-4 : Détection d'obstacle.....	37
Figure 2-5 : Approche champs de potentiel avec deux comportements (attractif et répulsif).....	38
Figure 2-6 : Gradient du champ attractif.....	40
Figure 2-7: Représentation 3D du potentiel attractif.....	40
Figure 2-8: Représentation 3D du potentiel répulsif.....	41
Figure 2-9: Gradient du champ répulsif.....	41
Figure 2-10: Combinaison d'un champ attractif et un champ répulsif.....	42
Figure 2-11 : Structure d'un contrôleur floue.....	97
Figure 2-12 : Logigramme de l'approche hybride.....	45
Figure 3-1 : Configuration géométrique « triangle ».....	49
Figure 3-2 : Maintien de la formation « configuration triangle ».....	49
Figure 3-3 : Distance entre le leader et les followers dans la configuration triangle.....	50
Figure 3-4 : Erreur sur la distance.....	50
Figure 3-5 : Erreur sur l'angle relative.....	51
Figure 3-6 : Suivi de trajectoire en formation triangle.....	51
Figure 3-7 : Erreur sur la distance entre le leader et les followers.....	52
Figure 3-8 : Erreur sur l'angle entre le leader et les followers.....	52
Figure 3-9 : Evitement des obstacles avec l'approche leader follower.....	53
Figure 3-10 : Maintien de la formation avec l'approche champs de potentiel.....	54

<i>Figure 3-11 : Erreur sur la distance dans le cas du maintien de la formation</i> .....	55
<i>Figure 3-12 : Suivi de trajectoire dans le cas de l'approche champs de potentiels.</i> .....	56
<i>Figure 3-13 : Erreur sur la distance dans le cas de l'approche champs de potentiel.</i> .....	56
<i>Figure 3-14 : Evitement d'obstacle avec l'approche champs de potentiel.</i> .....	57
<i>Figure 3-15 : Erreur sur la distance en évitant les obstacles</i> .....	58
<i>Figure 3-16 : Suivi de trajectoire : approche leader-follower-logique floue.</i> .....	59
<i>Figure 3-17 : Erreur sur la distance : approche leader-follower / logique floue</i> .....	60
<i>Figure 3-18 : Approche hybride: suivi de trajectoire et évitement d'obstacle.</i> .....	62
<i>Figure 3-19 : Approche hybride : erreur sur la distance.</i> .....	63
<i>Figure 3-20 : Approche hybride : erreur sur l'angle entre robots et leader virtuel.</i> .....	64
<i>Figure 4-1: Structure de Remote API pour V-REP(en violet)</i> .....	68
<i>Figure 4-2 : Relation page, vue et objet visible.</i> .....	69
<i>Figure 4-3: Sélecteur de scène</i> .....	71
<i>Figure 4-4:Ajout d'un modèle du navigateur de modèle.</i> .....	71
<i>Figure 4-5 : Etablissement de la connexion.</i> .....	99
<i>Figure 4-6 : WorkSpace MATLAB.</i> .....	100
<i>Figure 4-7: Création de la connexion.</i> .....	100
<i>Figure 4-8 : Robot mobil Pioneer 3-dx</i> .....	75
<i>Figure 4-9: Dimension du Pioneer 3-dx.</i> .....	76
<i>Figure 4-10: Etat initiale.</i> .....	77
<i>Figure 4-11: Maintien de la formation.</i> .....	77
<i>Figure 4-12: Suivi de trajectoire.</i> .....	78
<i>Figure 4-13: Erreur sur la distance.</i> .....	78
<i>Figure 4-14: Evitement des obstacles.</i> .....	79
<i>Figure 4-15: Erreur sur la distance.</i> .....	79
<i>Figure A-1 Modèle de la roue</i> .....	88
<i>Figure A-2 : Configuration du véhicule.</i> .....	89
<i>Figure A-3 : Trajectoire du robot mobile</i> .....	94
<i>Figure A-4: Commande du robot mobile</i> .....	94

## LISTE DES SYMBOLES

- $\dot{\Delta d}$  : Dérivé de l'erreur sur la distance entre le leader et le follower.
- $\dot{x}, \dot{y}$  : Dérivée des coordonnées cartésiennes du robot à roue.
- $\Delta d$  : Erreur sur la distance entre le leader et son follower.
- $\Delta \varphi$  : Erreur sur l'angle entre le leader et son follower.
- $A$  : Matrice d'adjacence.
- $A$  : Sous ensemble flou.
- $a, b, c$  : Sommet du triangle.
- $B_i$  : Partition de l'univers de sortie
- $d_{des}$  : Distance désirée entre le leader et son follower.
- $d_{lf}$  : Distance entre le leader et son follower.
- $E$  : Ensemble des arcs.
- $f$  : Follower
- $F(q)$  : Force artificielle.
- $f$  : Application de l'ensemble des arcs vers l'ensemble des nœuds.
- $g_{ik}$  : Distance entre le  $i^{ème}$  robot et le centre de la  $k^{ème}$  région interdite.
- $I$  : Point de contact de la roue.
- $k$  : Vitesse de translation du robot.
- $l$  : Leader
- $L$  : Distance entre le leader et son follower.
- $m_k$  : Rayon de la région autour du  $k^{ème}$  obstacle où la répulsion agit.
- $q$  : Cordonnée cartésienne du robot.
- $r$  : Rayon de la région d'attraction.
- $R_0$  : Robot 0.
- $R_1$  : Robot 1.
- $R_3$  : Robot 2.
- $R_f$  : Robot follower.
- $R_l$  : Robot leader.
- $r_0$  : Rayon de la région interdite qui entoure l'obstacle.
- $s$  : Distance de sécurité.
- $S_i$  : Distance entre le  $i^{ème}$  robot et le centre de la région d'attraction.
- $U(q)$  : Champ potentiel artificiel.
- $u_1$  : Vitesse de translation  $v$ .

$u_2$  : Vitesse de rotation  $\omega$ .  
 $U_a$  : Champ de potentiel attractif.  
 $v$  : Degré sortant d'un nœud.  
 $V$  : Ensemble des nœuds ou des sommets.  
 $v_f$  : Vitesse de translation robot follower.  
 $v_l$  : Vitesse de translation robot leader.  
 $x$  : Élément de l'univers.  
 $X$  : Univers de discours.  
 $x, y$  : Cordonnées cartésiennes du point de contact.  
 $x_f, y_f, \theta_f$  : Cordonnées cartésiennes et orientation du robot follower.  
 $x_G, y_G$  : Position du but.  
 $x_l, y_l, \theta_l$  : Cordonnées cartésiennes et orientation du robot leader.  
 $x_{ob}, y_{ob}$  : Position de l'obstacle  
 $\alpha_a$  : Pente du potentiel attractif.  
 $\alpha_r$  : Pente du potentiel répulsif.  
 $\theta, \phi$  : Orientation de la roue et l'angle de la roue mesurée à partir de la verticale.  
 $\mu_A$  : Fonction d'appartenance.  
 $\Psi$  : Angle entre le leader et son follower.  
 $\Psi_{des}$  : Angle désirée entre le leader et le follower.  
 $\Psi_{lf}$  : Angle entre le leader et son follower.  
 $\Psi$  : Angle directrice.  
 $\omega_f$  : Vitesse de rotation robot follower.  
 $\omega_l$  : Vitesse de rotation robot leader.

## INTRODUCTION GENERALE

La robotique telle qu'elle est connue aujourd'hui est une science interdisciplinaire comprenant de vastes champs de recherche : la vision, la planification, la commande du mouvement, le design, etc. Cependant, l'un des problèmes les plus importants reste la coopération, la planification et la coordination des mouvements au sein d'une architecture de commande dans un contexte multi-robots. L'étude des systèmes multi-robots est devenue une préoccupation majeure dans le milieu de la recherche en robotique, car quelles que soient les capacités d'un robot unique, il reste spatialement limité. Bien que le champ des systèmes multi-robots étende la recherche sur les systèmes mono-robot, c'est aussi une discipline à part entière car ils peuvent accomplir des tâches plus complexes que des robots simples ne peuvent pas réaliser.

Ce travail concerne la commande coopérative d'un système robotique constitué de multiples véhicules autonomes homogènes. C'est-à-dire comment définir les commandes de mouvement pour chaque véhicule du groupe pour réaliser des missions d'une manière à garder une formation : garder une configuration géométrique lors de la navigation du groupe dans son environnement.

Dans le chapitre 1 nous allons présenter une introduction aux systèmes multi-robots, en commençant par les définitions et la classification des systèmes multi-robots, leurs architectures de commande et leurs applications dans différents domaines.

Par la suite dans le chapitre 2 nous allons présenter les deux approches utilisées dans la commande d'une formation de robots mobiles à savoir : l'approche leader-follower et l'approche champs de potentiel, et les deux contributions que nous avons proposées pour améliorer les performances d'une formation : leader-follower/logique floue et l'approche hybride.

Le chapitre 3 est consacré aux résultats de simulations des approches présentées dans le deuxième chapitre. Où nous avons effectué plusieurs simulations pour mieux tester les performances de chaque approche.

Au début du chapitre 4 nous allons présenter l'environnement de simulation V-REP et la méthode de l'interfacé avec MATLAB, puis nous montrons les résultats de simulation de l'algorithme hybride appliqué sur une formation de robots mobiles de type « Pioneer 3-dx ».

# **Chapitre1 :**

## **Formation de groupe de robots, Etat de l'art**

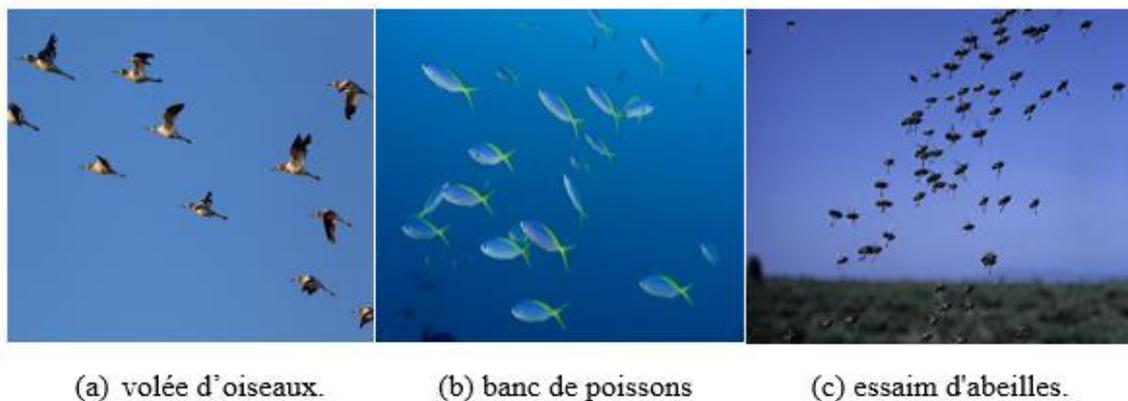
# 1 FORMATION DE GROUPE DE ROBOTS, ETAT DE L'ART

## 1.1. INTRODUCTION

Les systèmes multi-robots sont actuellement une des préoccupations majeures de la recherche en robotique. En effet, les systèmes multi-robots peuvent accomplir des tâches qu'un seul robot ne peut pas effectuer, notamment parce qu'un seul robot est spatialement limité. En 1989, Brooks et Flynn du laboratoire « MIT's Artificial Intelligence » ont proposé ce qu'ils ont appelé une idée radicale dans le domaine de l'exploration du système solaire : remplacer un grand robot explorateur par une collection de petits robots mobiles appelés « rover » [1].

Cette idée radicale présente la principale motivation pour l'utilisation de systèmes multi-robots, en effet, à plusieurs ils peuvent se disperser plus rapidement et couvrir un large espace pour les missions de reconnaissance, ou bien transporter des objets dispersés dans une zone de stockage. Ils peuvent même être utilisés dans des missions sous-marines comme la cartographie des fonds des mers, la récupération des balises échouées ou encore les boites noires suite au crash d'avions.

Cette tendance a été également soutenue par les efforts fournis dans le but d'imiter les systèmes biologiques tels que, les volées d'oiseaux, les bancs de poissons et les essaims d'insectes comme les montrent la figure ci-dessous.



**Figure 1-1 : Exemples de systèmes biologiques exhibant des comportements coopératifs.**

Les comportements collectifs s'observent principalement chez les insectes sociaux (fourmis, termites et abeilles), les animaux se déplaçant en formation (oiseaux migrateurs, bancs de poissons) et, dans une moindre mesure, les mammifères sociaux chassant en meute (loups, hyènes, lions). Ainsi, et pendant les saisons froides, on a couramment observé des

troupe d'oiseaux volant en formation. La forme en V, montrée sur la **Figure 1-2**, est utilisée par les oiseaux de grandes tailles, tels que les canards et les oies. Ce type de formation permet aux oiseaux de réduire les dépenses d'énergie pendant les migrations. Voler en formation favorise également la communication et la coordination dans le groupe et peut expliquer l'avantage de migrer ensemble plutôt qu'en solitude.



**Figure 1-2 : Troupe d'oiseaux volant en formation V.**

Comme le comportement collaboratif des animaux a évolué dans le but d'accroître la probabilité de survie de l'espèce, les chercheurs commencent à penser que peut-être un système multi-robots, où tous les agents collaborent, peut augmenter les chances d'accomplir une tâche prédéfinie. Parmi tous les êtres vivants, l'homme représente le niveau de collaboration le plus élevé, ainsi il a pu modifier l'environnement en l'espace de quelques siècles partant de ses considérations provenant des systèmes naturels.

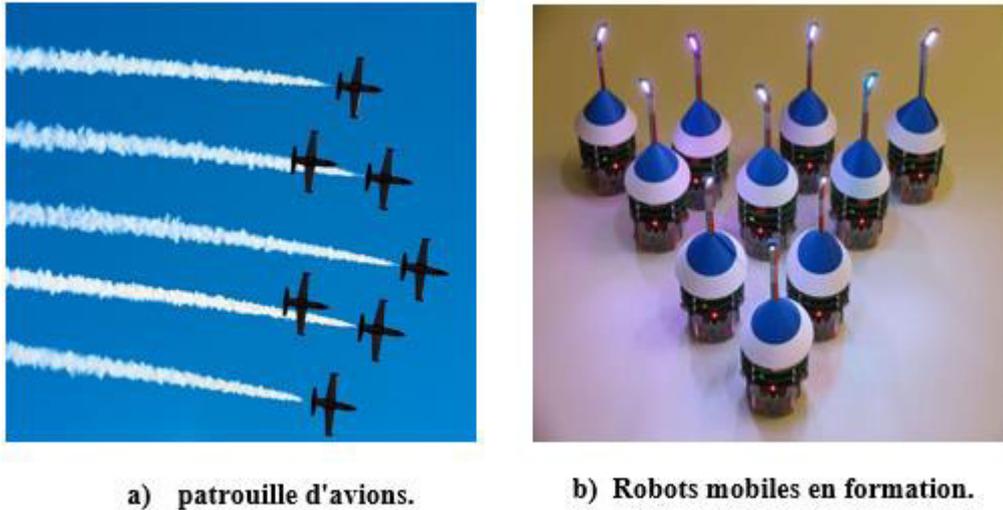
## 1.1 GENERALITES SUR LES FORMATIONS

Dans la mise en œuvre d'un système multi-robots [2], nous pouvons distinguer deux types de comportements :

- Les comportements compétitifs ; où chaque robot cherche à améliorer ses performances au détriment des autres, plus ou moins volontairement.
- Les comportements collaboratifs ; où les robots prennent en considération les objectifs des autres afin d'améliorer un résultat global.

Les cas qui nous intéressent sont de type collaboratif. On peut distinguer plusieurs niveaux dans la collaboration entre robots :

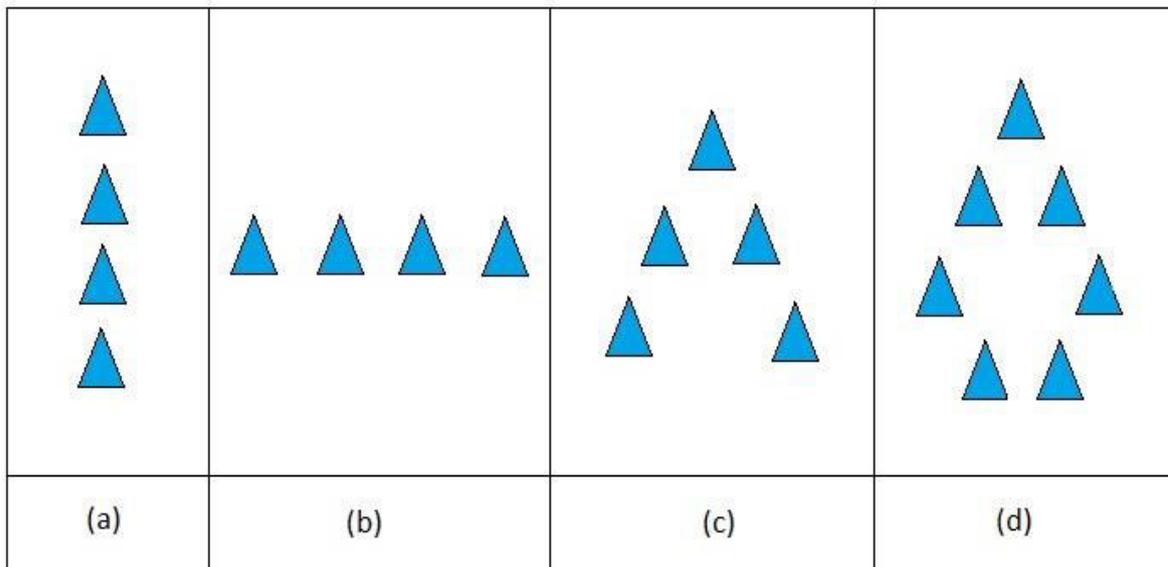
- La coordination est la forme la plus simple. Elle consiste essentiellement à synchroniser les actions des différents robots de façon à respecter l'ordre des tâches d'un plan global. Chaque robot a ses buts et sa liste de tâches propres. Un exemple de coordination est la régulation du trafic aérien de façon à éviter les collisions entre avions dont les plans de vol sont indépendants. (**Figure 1-3 (a)**)
- La coopération vise à mettre en œuvre plusieurs robots pour réaliser une tâche commune. Cette situation peut venir du fait qu'un robot seul n'a pas la capacité ou les outils pour réaliser la tâche demandée. La coopération implique évidemment l'existence d'une bonne coordination entre les robots. La coopération peut être une action ponctuelle au cours de l'exécution des missions des robots, ou s'étendre sur une plus longue période.
- La formation est un cas particulier de coopération. Au cours d'un déplacement en formation, les robots conservent un but commun en termes de position à atteindre ou de trajectoire à réaliser. De plus, ce déplacement se fait dans un espace de dimensions réduites. L'objectif est de pouvoir réaliser à tout moment des tâches en coopération, les robots assurant la coordination spatiale et temporelle. Ce type de déplacement est très bien adapté au convoyage d'un ensemble de robots. (**Figure 1-3 (b)**).



**Figure 1-3 : Exemples de formation.**

**Définition :** Une formation est une organisation d'un groupe d'engins se déplaçant ensemble, dans un espace restreint, et partageant des buts et des contraintes [3].

Au cours d'un déplacement en formation, nous pouvons définir de façon géométrique les relations de distance entre les robots. La géométrie d'une formation doit avant tout avoir un intérêt pratique d'amélioration de l'efficacité et de la sécurité du groupe. Notre objectif est de déterminer une géométrie adaptée à l'environnement en s'affranchissant des schémas classiques tels que les formations en « V inversé », « diamant », « colonne », « ligne » et « cercle ». Ces formations sont illustrées dans la figure suivante.



**Figure 1-4 : Différents types de formation : (a) colonne, (b) ligne, (c) V inversé.**

## 1.2 TYPES DES ROBOTS DANS UNE FORMATION

Il convient d'observer la diversité des systèmes où l'utilisation d'une formation est mise en œuvre. Nous faisons ici la distinction en fonction du milieu dans lequel évoluent les robots [4].

### 1.2.1 ROBOTS TERRESTRES

L'utilisation de la formation pour les robots terrestres a été assez peu étudiée et essentiellement pour des systèmes simples. Les robots mobiles holonomes ou non holonomes ont été largement utilisés pour démontrer l'efficacité de différentes méthodes. La majorité des études porte sur la coordination et la coopération entre plusieurs robots autonomes.



**Figure 1-5 : Exemple de robots terrestres.**

### 1.2.2 SATELLITES

Les constellations de satellites sont de plus en plus utilisées dans les domaines des télécommunications, du positionnement par satellites et de l'observation spatiale.

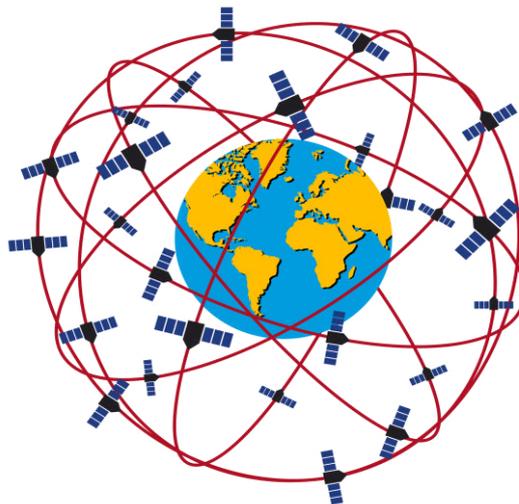
Les satellites de communication (télévision, internet. . .) sont généralement placés sur orbite géostationnaire et doivent assurer un espacement constant entre eux tout en maintenant leurs antennes pointées vers la Terre. Les problématiques auxquelles sont confrontés ces systèmes relèvent purement d'un contrôle automatique. La configuration de la formation est très simple et ne change pas pendant la durée de service des satellites.

Les constellations de type GPS « Global Positioning System » ou téléphone par satellites sont sur des orbites basses ou moyennes. La problématique n'est plus ici de maintenir une formation serrée, mais de placer les satellites sur des orbites décalées et synchronisées, de façon à avoir la meilleure couverture au sol pour assurer une bonne qualité

de service (il faut voir au moins quatre satellites GPS simultanément pour avoir une position). (Figure 1-6)

Enfin, les systèmes d'observation de la Terre ou de l'espace utilisent de plus en plus des instruments placés sur des satellites différents.

Dans l'avenir, nous verrons probablement le développement des microsatellites pour assister les astronautes dans la construction de stations orbitales ou pour effectuer des diagnostics ou des réparations autonomes lorsque l'humain n'est pas présent ou incapable d'intervenir. La coopération, la reconfiguration et l'évitement d'obstacles seront au centre de toutes les attentions.



**Figure 1-6 : Constellation de satellite de type GPS ou téléphone.**

### **1.2.3 ROBOTS AQUATIQUES**

Les problématiques liées au contrôle de formation pour des robots marins sont très proches de celles des avions civils, avec néanmoins des dynamiques assez différentes. Les applications possibles sont le convoyage de plusieurs cargos avec des équipages réduits, la navigation automatique dans des zones de grand trafic (par exemple, le rail d'Ouessant) ou à l'approche des zones portuaires, des systèmes sous-marins exploitent eux aussi les déplacements en formation. Il existe par ailleurs des initiatives plus originales, telles que le contrôle d'une formation de robots poissons biomimétiques.

### 1.2.4 ROBOTS VOLANTS

Les robots aériens ou drones « Unmanned Aerial Vehicle, UAV » offrent une très grande variété de types. Ils sont généralement classés selon leur taille et leur endurance. On peut ainsi distinguer les drones HALE « Haute Altitude Longue Endurance », les drones MALE « Moyenne Altitude Longue Endurance », les drones de courte et moyenne portée et les mini drones. Ils peuvent également être caractérisés par leur fonction : drones stratégiques, drones tactiques ou drones de combat « Unmanned Combat Air Vehicle,UCAV ». Les types d'engins vont des avions à ailes fixes aux voilures tournantes, en passant par des systèmes hybrides.

Les drones stratégiques sont des drones HALE et ne sont pas vraiment concernés par les problématiques de vols en formation, car ils sont principalement destinés à des missions de reconnaissance ou de guerre électronique où un seul avion est généralement suffisant. Ils pourront néanmoins être intéressants à l'avenir dans le cas de ravitaillements en vol, où ils pourraient servir d'avions tanker.

Les drones tactiques concernent une très large gamme d'engins, allant du mini ou micro drones aux drones MALE. Ils sont donc amenés à exécuter un panel très large de missions. Le drone MALE le plus connu est le Predator américain. (**Figure 1-7**)



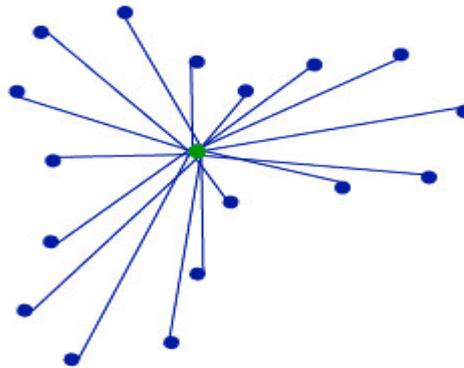
**Figure 1-7 : Drone MALE PREDATOR.**

### 1.3 ARCHITECTURES DE COMMANDE

La robustesse des systèmes multi-robots est strictement liée à la structure de commande utilisée pour organiser les agents et obtenir le comportement émergent désiré. Quatre architectures différentes peuvent être identifiées dans le domaine des systèmes multi-robots, qui sont résumées ci-dessous [3].

#### 1.3.1 CENTRALISEE

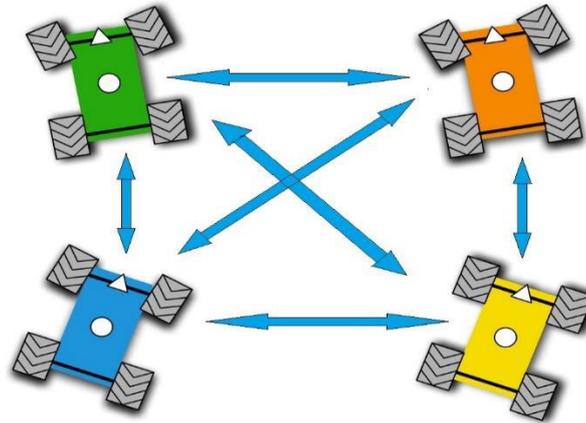
Un superviseur global reçoit l'information des membres du groupe et le comportement de l'essaim est coordonné à partir d'un point de contrôle unique. Cette architecture présente l'avantage que toutes les informations sont collectées par une seule unité (**Figure 1-8**), ce nœud doit être assez puissant pour calculer la loi de commande pour chaque robot avec la prise en considération la possibilité d'avoir des tâches complexes. Le prix à payer est la complexité de calcul et la dépendance des robots vis-à-vis du superviseur. En pratique, le nombre de calculs augmente très rapidement en fonction du nombre de robots et le problème devient rapidement insoluble pour une flottille de plus de dix robots.



**Figure 1-8 : Commande centralisée.**

#### 1.3.2 DECENTRALISEE

C'est l'architecture la plus utilisée pour la commande des systèmes multi-robots. Dans cette architecture, l'action de chaque robot est fondée seulement sur la connaissance locale de l'état de ses voisins et de son environnement. Elles nécessitent généralement un flux de communications assez élevé afin de transmettre des requêtes informatives aux autres individus (**Figure 1-9**). Le protocole peut inclure des notions d'intention et d'engagement à partir desquelles chaque robot élabore sa propre trajectoire en prenant en considération les activités des autres robots.

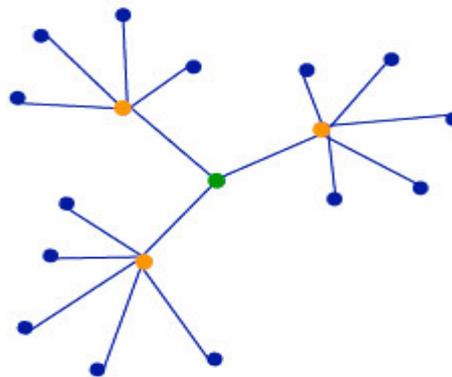


**Figure 1-9 : Commande décentralisée.**

Les avantages d'une telle stratégie sont multiples : absence de superviseur, diminution des temps de calcul par rapport à une approche centralisée, plus grande robustesse (par exemple une défaillance d'un ou plusieurs robots n'entraîne pas généralement une défaillance de l'ensemble du système). Le prix à payer est bien sûr une incapacité à délivrer une performance optimale puisqu'à chaque instant, chaque robot ne dispose que d'informations limitées et incomplètes sur les autres robots de la flottille en plus, il présente l'inconvénient d'être limité en puissance de calcul. Cela signifie qu'il n'est pas aisé d'implémenter des tâches complexes d'une manière distribuée.

### 1.3.3 HIERARCHIQUE

Cette architecture est inspirée directement du protocole de commandement militaire est préconisée pour certaines applications. Elle est basée sur l'idée que certains robots peuvent commander, comme des leaders (superviseurs), c'est-à-dire un petit groupe de robots. Une fois encore, comme dans les architectures centralisées, le problème avec cette approche réside dans le cas de la défaillance des superviseurs. (**Figure 1-10**)



**Figure 1-10 : Commande hiérarchique.**

### **1.3.4 HYBRIDE**

Cette approche est un compromis entre l'architecture centralisée et l'architecture décentralisée. En particulier, elle est basée sur l'idée que l'un ou plusieurs superviseurs de haut niveau affectent les tâches et les ressources, et les robots de niveau inférieur dans la hiérarchie exploitent les informations locales afin d'accomplir les tâches prédéfinies. Cette architecture hybride a été utilisée dans de nombreuses applications basées sur la commande multi-robots [5].

## **1.4 STRATEGIES DE COMMANDE**

Dans cette section, différentes stratégies de commande et méthodes de représentation de formes sont examinées. Pour la commande de formation de robots mobiles, différentes topologies de commande peuvent être adoptées selon les scénarios et les spécificités de chaque mission. Il peut y avoir un ou plusieurs leaders dans le groupe, tandis que les autres robots suivent un ou plusieurs leaders d'une manière spécifique. Chaque robot est équipé de moyens de détection et de calcul. Dans d'autres applications, les robots sont limités en communication. Généralement, la connaissance globale du système n'est pas disponible pour chaque robot. Une méthode centralisée n'est pas utilisée dans ce cas de figure. De ce fait il est nécessaire de concevoir un contrôleur pour chaque robot en se basant sur les informations locales. S'il n'y a pas de leader, chaque robot doit coordonner ses actions avec les autres en comptant sur un certain consensus global pour réaliser l'objectif commun.

### **1.4.1 APPROCHES COMPORTEMENTALES ET CHAMPS DE POTENTIELS**

Les approches comportementales sont souvent liées aux approches basées sur les champs de potentiels. Ces approches ont été inspirées en observant les phénomènes physiques biologiques. Dans telles approches, plusieurs comportements désirés (évitement des collisions, maintien de la formation, poursuite de cible, etc.) sont définis pour chaque robot de la formation. Chaque comportement correspond à un schéma de réaction élémentaire.

L'action qui va effectivement être appliquée est une moyenne pondérée des vecteurs de réactions élémentaires. Le choix des poids associés à chaque réaction peut s'avérer très complexe et fait appel le plus souvent à l'expertise du concepteur.

Le formalisme et l'analyse mathématique de cette approche sont difficiles et par conséquent la convergence de la formation à une configuration désirée n'est pas garantie.

Dans l'approche comportementale, la commande est distribuée. Le système est évolutif et il n'y a aucun sous-système dont l'utilisation augmente d'une manière significative quand de nouveaux robots sont ajoutés à la formation. L'ajout de nouveaux robots entraîne l'auto-organisation du système.

L'approche comportementale est préconisée pour certaines classes de tâches comme le transport de grande quantité de petits objets, encercler un intrus ou la poursuite des intrus, elle peut être également appliquée à la commande de formation d'avion, la cartographie et l'exploration collaborative.

#### **1.4.2 APPROCHES STRUCTURES VIRTUELLE**

Dans l'approche structure virtuelle la formation entière est traitée comme un corps rigide simple. Le concept de la structure virtuelle a été introduit par M.Lewis [6]. L'approche structure virtuelle est typiquement utilisée pour le contrôle de vol en formation des avions et la constellation des petits satellites [7]. La structure virtuelle peut adapter sa forme et s'élargir dans une direction spécifique tout en maintenant une relation géométrique rigide entre les agents. Cette approche a été proposée dans le but d'accroître la précision dans la commande des formations de robots mobiles.

La structure de formation est généralement très serrée et précise dans ces méthodes pendant les tâches. Son inconvénient principal est dans la complexité de calcul.

Dans les approches dites structure virtuelle, la trajectoire désirée n'est pas assignée à chaque robot, mais à la formation entière. Le contrôleur est synthétisé en trois étapes. Premièrement, la dynamique de la structure virtuelle est définie. Ensuite, le mouvement désiré de la structure virtuelle est transposé vers les mouvements désirés pour chacun des agents qui composent la structure. Enfin, des contrôleurs individuels sont dérivés pour le suivi de chaque agent.

Le principal avantage de cette approche c'est qu'il est simple de prescrire le comportement pour le groupe entier. La structure de formation est généralement très serrée et précise dans ces méthodes pendant les tâches. Par contre, son inconvénient principal réside dans la complexité de calcul en plus la nature centralisée de ces approches les rend moins robustes aux échecs.

#### **1.4.3 APPROCHES LEADER-FOLLOWER**

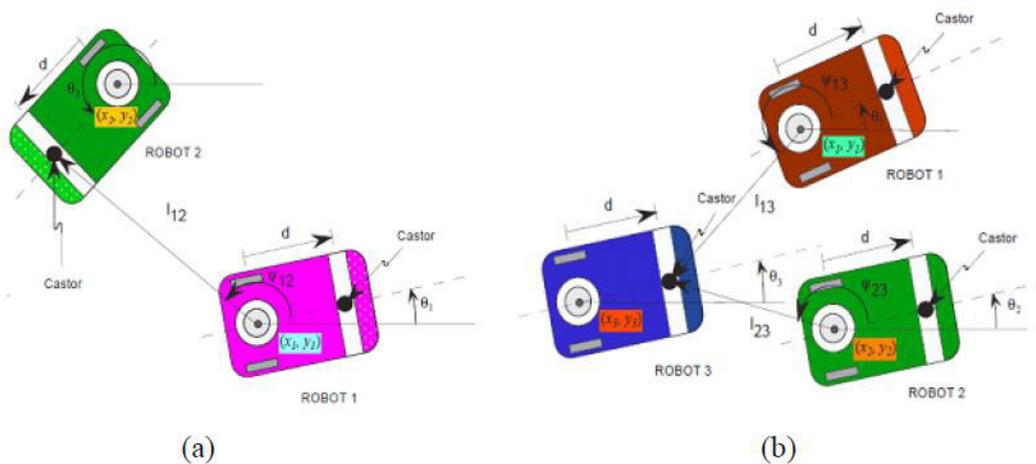
La stratégie dite *leader-follower* a été introduite la première fois dans la régulation des systèmes par l'économiste allemand « *Heinrich Freiherr von Stackelberg* ». Ces méthodologies de contrôle sont connues également sous le nom de stratégies de *Stackelberg*

qui sont appropriées pour des classes des problèmes de système où il y a de multiples critères, des décideurs multiples et de l'information décentralisée. Dans cette stratégie, les actions de contrôle de suiveur sont basées sur l'état et la commande du leader. Par ailleurs, le concept du leader-follower est couramment employé dans la commande des systèmes multi-robots.

Dans les approches leader-follower, chaque agent robotique est positionné dans la formation par sa géométrie relative par rapport à ses voisins prédéfinis. Chaque robot suit un leader prédéfini en respectant une certaine relation géométrique. En utilisant la relation leader-follower, une forme géométrique de  $n$  robots peut être obtenue. En général on a un seul robot leader dans la formation. Le leader ne suit aucun robot dans la formation, mais il suit une trajectoire prédéfinie.

On distingue deux types de contrôleurs pour le maintien d'une formation de plusieurs véhicules :

- Le premier est de type L -  $\varphi$  dont l'objectif est de maintenir une distance  $l$  et un angle  $\varphi$  entre le leader et son follower. (**Figure 1-11 (a)**)
- Le second est de type L - L qui considère la distance relative entre trois robots dont un seulement est le follower des deux autres. (**Figure 1-11 (b)**)



**Figure 1-11 : (a) leader follower type L –  $\varphi$ , (b) leader follower type L – L.**

L'approche leader-follower est plus sûre que celle de la structure virtuelle, car lorsqu'un robot de la formation tombe en panne, le reste de la formation continue l'exécution de la tâche, par contre lorsque le leader tombe en panne c'est toute la formation entière qui échoue. Dans la plupart des méthodes leader-follower, il n'y pas de retour de l'information, ainsi lorsque les robots suiveurs traînent derrière le leader pour n'importe quelle raison, par exemple des perturbations qui apparaissent dans l'environnement, cet événement n'affecte

pas le mouvement du leader, mais la formation sera détruite. Malgré ces insuffisances l'approche leader-follower est particulièrement appréciée dans la littérature en raison de sa simplicité et son évolutivité.

#### **1.4.4 AUTRES APPROCHES**

On peut trouver d'autres stratégies de commande de formation faisant appel au contrôle non-linéaire, à la logique floue, aux algorithmes génétiques ou à l'apprentissage par renforcement. Certaines de ces méthodes peuvent venir en complément de solutions plus classiques. Par exemple, l'utilisation de modèles prédictifs permet de pallier l'imperfection des communications entre robots. On peut également signaler les Boids [8], très utilisés dans l'animation par ordinateur, qui permettent de simuler les réactions d'un grand nombre d'individus en utilisant les méthodes comportementales sur des informations très locales. Les résultats sont très proches des réactions naturelles des oiseaux ou des bancs de poissons.

### **1.5 EVALUATION DES FORMATIONS**

#### **1.5.1 AVANTAGES D'UNE FORMATION DE ROBOTS**

- Un faible coût pour une couverture plus étendue,
- Une capacité de redondance (si l'un des robots est défaillant en raison d'une panne, d'un blocage, etc. un autre robot peut prendre des mesures pour le dépanner ou le remplacer dans sa tâche)
- La capacité à couvrir une grande surface ont par exemple montré (via une simulation appliquée au cas de l'île de Lampedusa) en 2014 qu'un essaim de 1000 petits drones aquatiques dispersés en mer à partir de 2 bases pourraient en 24 heures faire un bilan de surveillance sur une bande maritime longue de 20 km [9].

#### **1.5.2 INCONVENIENTS D'UNE FORMATION DE ROBOTS**

À ce jour, les formations de robots ne peuvent remplir que des tâches relativement simples, ils sont souvent limités par leur besoin en énergie. De manière plus générale, les difficultés d'interopérabilité quand on veut associer des robots de nature et d'origines différentes sont aussi encore très limitantes.

## 1.6 APPLICATIONS

### 1.6.1 DOMAINE MILITAIRE

Il est évident que les utilisateurs privilégiés pour ce type de systèmes sont les militaires. De nombreuses études sont menées sur la planification de missions militaires pour des formations de drones de combat ou d'observation.

Parmi les applications militaires les plus communes on en cite :

- Le vol en formation (R. M. Murray 2007), où des UAVs sont mis en formation dans un espace 3D. Pour une mission critique, un contrôle robuste de la formation doit être maintenu, pour un avion suiveur, l'objectif est la poursuite du leader. Cependant, la planification de la trajectoire du leader est un élément important dans la conception du groupe entier. L'un des problèmes à régler est de minimiser l'énergie dépensée tout en maintenant la formation.
- La surveillance coopérative est une application très intéressante. Elle peut être définie par le problème de surveillance d'une zone géographique par un groupe de véhicules décentralisé.

### 1.6.2 APPLICATION CIVILES

Les applications potentielles pour les formations des robots sont en effet énormes. Il comprend des tâches qui exigent la miniaturisation, comme les tâches de détection réparties dans les micromachines ou le corps humain. L'une de ces utilisations les plus prometteuses est les missions de sauvetage en cas de catastrophe.

Des robots de différentes tailles pourraient être envoyés aux endroits où les secouristes ne peuvent pas atteindre en toute sécurité pour détecter la présence de la vie via des capteurs infrarouges. D'autre part, Elle peut être adaptée aux tâches qui exigent des designs peu coûteux, par exemple des tâches d'extraction ou de recherche agricole. De plus, certains artistes utilisent des techniques de robotique pour réaliser de nouvelles formes d'art interactif.

## 1.7 CONCLUSION

Ce chapitre nous a permis en premier lieu d'introduire le domaine de la robotique en groupe et ses applications. Étant donné que l'objectif de notre travail est de contrôler un groupe de robots mobiles naviguant en formation dans un environnement peu connu, nous avons cité les trois approches principales de commande de formations de robots mobiles à savoir, l'approche comportementale, l'approche structure virtuelle et l'approche leader-follower, ces trois approches ont été étudiées d'une manière intensive par la communauté scientifique.

# **Chapitre 2 :**

## **Commande coopérative d'une formation de robots mobiles**

## 2 COMMANDE COOPERATIVE D'UNE FORMATION DE ROBOTS MOBILES

### 2.1 INTRODUCTION

Dans ce chapitre nous allons nous intéresser à la commande d'une formation de robots mobiles, en introduisant les différentes approches utilisées. Dans un premier temps nous allons présenter l'approche leader-follower qui est basée sur la théorie des graphes orientés, par la suite nous présentons la deuxième approche dite approche comportementale (méthode des champs de potentiels), finalement nous allons développer deux approches proposées à savoir : l'approche leader-follower/logique floue et l'hybridation entre l'approche leader-follower et l'approche comportementale.

### 2.2 COMMANDE DES FORMATIONS :

Il y a plusieurs critères qui doivent être pris en considération pour la conception d'un contrôleur de formation, tels que la stabilité et la commandabilité. Différentes approches pour le contrôle ont été mises en avant pour résoudre ces problèmes, avec notamment les stratégies de leader-follower, les structures virtuelles ou les méthodes comportementales. Cette section aborde ces différentes notions.

#### 2.2.1 STABILITE

La stabilité est la capacité d'un système à revenir naturellement dans une position l'équilibre après avoir reçu une perturbation, l'un des théorèmes qui garantit la stabilité des systèmes est le théorème de Lyapunov, dont l'énoncé est le suivant :

Soit  $q = 0$  un point d'équilibre du système  $\dot{q} = f(q)$ . Soit une fonction  $V : R^n \rightarrow R$  continue différentiable tel que :

1.  $V$  est définie positive sur  $D$ .
2.  $\lim_{t \rightarrow +\infty} \|q\| = \infty \Rightarrow \lim_{t \rightarrow +\infty} V(q) = \infty$ .
3.  $\dot{V} < 0$  pour  $q \neq 0$ .

Alors,  $q = 0$  est globalement asymptotiquement stable.

Pour les formations de robots mobiles on peut distinguer deux concepts de stabilité différents :

### 2.2.1.1 Stabilité de corde et de grille (String and mesh stability)

La stabilité de formation dans l'espace 1D s'appelle la stabilité de corde « string », elle concerne la propagation d'une perturbation dans une formation en colonne. Une colonne est stable si l'effet d'une perturbation à la source est amorti quand elle se déplace vers les followers, la généralisation de cette stabilité à l'espace 2D donne la stabilité de grille « mesh stability » qui garantit une atténuation des erreurs pour des systèmes interconnectés. De plus, la stabilité est [10] préservée lorsque le système est agrandi suivant certaines conditions [11].

Une formation est dite string stable si l'erreur  $\epsilon_i$  de  $i$ ème agent et l'erreur  $\epsilon_{i+1}$  de  $(i+1)$ ème agent telle que :  $\|\epsilon_i\| < \|\epsilon_{i+1}\|$  pour un système linéaire de sortie  $y = h * u$  donne :

$$\|(t)\| < \|h(t)\| \cdot \|u(t)\|$$

Cette condition peut être généralisée pour le cas mesh stabilité (MIMO) et on aura la condition suivante :

$$\|(t)\| < \max \Sigma \|h_{ij}(t)\| \cdot \|(t)\|_{nj=1}$$

### 2.2.1.2 Stabilité LFS (leader-to-formation stability)

Stabilité LFE est basée sur l'approche leader-follower. Cette approche diffère de la stabilité de grille dans le sens où elle met d'avantage l'accent sur la façon dont le comportement du leader peut affecter les erreurs d'interconnexion dans la formation.

On considère une formation de robot mobile qui a le vecteur  $u = [v, \omega]^T$  comme entrée de commande et soit une distance  $\ell$  et un angle relatif  $\varphi$  entre le leader et son suiveur et  $\theta$  l'orientation de robot, soit le vecteur d'état suivant :  $z_{ij} = [l_{ij}, \varphi_{ij}, \theta_{ij}]$

**Définition :** Une formation est dite LFS s'il existe une fonction  $\beta$  et une fonction  $\gamma$ , tel que pour toute erreur initiale  $z(0)$  et pour toute entrée bornée  $u$ , l'erreur  $z$  satisfait :

$$\|(t)\| \leq \beta(\|z(0)\|, t) + \Sigma \gamma(\sup \|u\|_{i \in n})$$

Où les fonctions  $\beta(r, t)$  et  $\gamma(r)$  sont les gains transitoires et asymptotiques de la LFS [10].

### 2.2.2 COMMANDABILITE

La théorie des graphes est un outil important pour l'analyse de la stabilité d'une formation. Elle permet de représenter les interconnexions entre des robots et en particulier les échanges d'informations. La topologie d'un graphe permet d'étudier la stabilité, mais aussi la commandabilité du système. Un système est dit commandable [11] si quel que soit l'état à l'instant initial, et quel que soit l'état à l'instant final, il existe une commande appliquée sur un intervalle de temps fini, qui permet de rejoindre l'état final partant de l'état

initial. Il est également possible d'utiliser cette théorie pour choisir un contrôleur approprié à la configuration de la formation, s'il existe [10].

## 2.3 APPROCHE LEADER-FOLLOWER

Le problème considéré est la commande d'une formation constituée de  $N$  robots mobiles à noter que nous allons utiliser l'approche leader-follower avec la commande  $L - \varphi$ .

### 2.3.1 MODELISATION DE LA FORMATION LEADER-FOLLOWER

Dans la Figure 2-1 on a un robot leader et un robot follower qui sont notés  $R_l$  et  $R_f$  respectivement, les états et les entrées de  $R_l$  et  $R_f$  sont notés comme  $(x_l; y_l; \theta_l)$ ,  $(x_f; y_f; \theta_f)$ ,  $(v_l; \omega_l)$  et  $(v_f; \omega_f)$  où l'indice "l" et "f" signifie leader et follower respectivement. La distance entre le leader et le robot suiveur est notée  $d_{lf}$  et l'angle entre les deux robots est noté  $\psi_{lf}$  sont donnés par :

$$\begin{cases} d_{lf} = \sqrt{(x_l - x_f)^2 + (y_l - y_f)^2} \\ \psi_{lf} = \pi - \tan^{-1}\left(\frac{y_f - y_l}{x_f - x_l}\right) - \theta_l \end{cases} \quad (2.1)$$

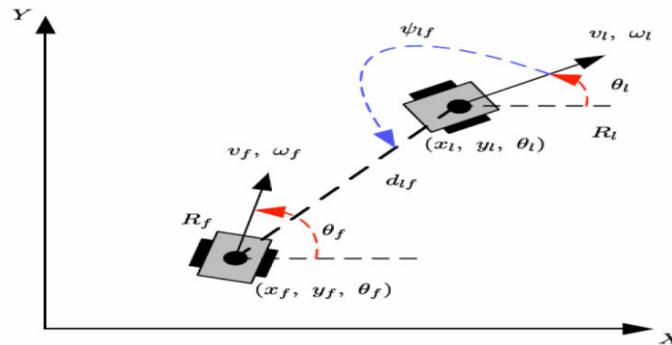


Figure 2-1: Configuration Leader-follower

L'objectif est de permettre au robot follower de suivre le leader en préservant une distance constante et une orientation constante du follower par rapport au leader.

Soit  $d_{des}$  et  $\psi_{des}$  la distance et l'orientation respectivement désirée entre le leader et le follower, on définit les erreurs suivantes :

$$\begin{cases} \Delta d = d_{des} - d_{lf} \\ \Delta \psi = \psi_{des} - \psi_{lf} \end{cases} \quad (2.2)$$

Pour réaliser une formation contenant  $N$  robots on fait rappel à quelques notions de la théorie des graphes :

**Graphe orienté** : un graphe orienté est un triplet  $(\mathbf{V}, \mathbf{E}, \mathbf{f})$ , avec  $\mathbf{V}$  est l'ensemble des nœuds ou des sommets,  $\mathbf{E}$  est l'ensemble des arcs et  $\mathbf{f}$  est une application de l'ensemble des arcs vers l'ensemble des nœuds :  $\mathbf{E} \rightarrow \mathbf{V} \times \mathbf{V}$ . Le degré sortant d'un nœud  $\mathbf{v} \in \mathbf{V}$  est le nombre d'arrêtes qui ont  $\mathbf{v}$  comme sommet initial, par contre le degré entrant représente le nombre d'arrêtes qui ont  $\mathbf{v}$  comme sommet terminal.

- Un arbre est un graphe orienté sans cycle.
- Matrice d'adjacence : la matrice d'adjacence  $\mathbf{A} = (\mathbf{a}_{ij})_{n \times n}$  de  $\mathbf{G}$  est définie par:

$$\mathbf{a}_{ij} = \begin{cases} \mathbf{1} & \text{si } \mathbf{v}_i, \mathbf{v}_j \in \mathbf{E} \\ \mathbf{0} & \text{sinon} \end{cases} \quad (2.3)$$

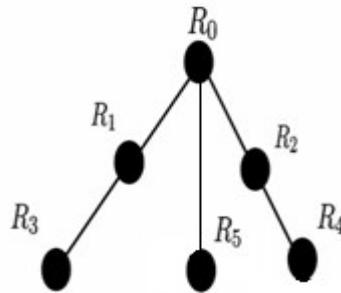
En utilisant les définitions précédentes, nous pouvons établir une relation de correspondance entre un arbre graphe et une formation de robots mobiles, tel que les nœuds sont représentés par les différents robots mobiles, la racine représente le robot leader, les feuilles représentent les followers et les arcs orientés représentent les connexions entre les différentes paires leader-follower. Dans la Figure 2-2 on représente une structure de six robots mobiles en formation, où les nœuds de l'arbre sont les robots mobiles et les arrêtes directes représentent la relation entre les paires leader-follower. Par exemple,  $\mathbf{R}_1 \rightarrow \mathbf{R}_3$  veut dire que  $\mathbf{R}_1$  est le leader de  $\mathbf{R}_3$ .

Cette matrice indique les interconnexions de cette formation, où chaque ligne représente un suiveur (follower) et chaque colonne représente son leader candidat. Si une ligne  $\mathbf{i}$  a une entrée non nulle sur la colonne  $\mathbf{j}$ , alors le robot  $\mathbf{i}$  suit le robot  $\mathbf{j}$ . Comme le robot  $\mathbf{R}_0$  est le leader, c'est-à-dire qu'il ne suit aucun autre véhicule, il n'y a aucune entrée non nulle sur la première ligne. La matrice d'adjacence de la structure est la suivante :

$$\begin{array}{c} \mathbf{R0} \ \mathbf{R1} \ \mathbf{R2} \ \mathbf{R3} \ \mathbf{R4} \ \mathbf{R5} \\ \mathbf{R0} \ \mathbf{R1} \ \mathbf{R2} \ \mathbf{R3} \ \mathbf{R4} \ \mathbf{R5} \end{array} \begin{pmatrix} \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} \\ \mathbf{1} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} \\ \mathbf{1} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} \\ \mathbf{1} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} \\ \mathbf{1} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} \\ \mathbf{1} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} \end{pmatrix}$$

Le robot navigue seul et se dirige vers la position indiquée dans la formation afin de construire un modèle géométrique, pour cela on décompose l'approche leader-follower en quatre étapes séparées :

- a. Définition du modèle géométrique : V inversé, ligne, colonne ou diamant.
- b. Planification du chemin pour le leader (robot réel ou virtuel).
- c. Planification de trajectoire pour chaque robot suiveur dans la formation.
- d. Contrôle des robots.



**Figure 2-2: Formation triangle de six robots mobiles**

### 2.3.2 ARCHITECTURE DE L'APPROCHE LEADER-FOLLOWER

Dans la commande d'une formation de robots mobiles par l'approche leader-follower nous utilisons une architecture à quatre niveaux [12] telle qu'elle est montrée dans la **Figure 2-3**.

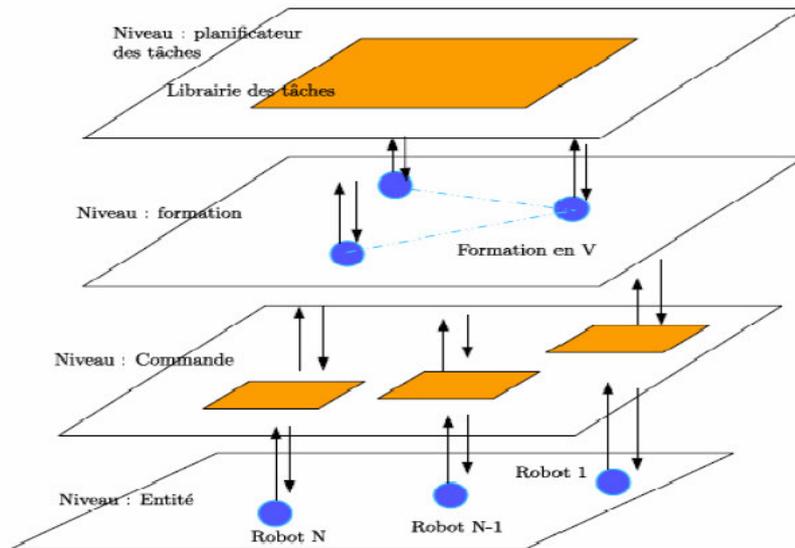
**Niveau de planification :** Le niveau de planification des tâches permet au superviseur de spécifier le plan des tâches, y compris le modèle de formation pour que le groupe entier accomplisse la tâche assignée. Ce niveau de planification de tâche commande le niveau de formation.

**Niveau de formation :** Ce niveau est responsable de la coordination. Les agents de ce niveau sont le leader et les followers. Une fois que le rapport spécifique est établi, le niveau de formation communique avec les robots au niveau de contrôle pour coordonner leur mouvement dans le modèle désiré de formation. La formation reste inchangée tant que le niveau de planification de tâche n'envoie pas une nouvelle commande.

**Niveau de contrôle :** À ce niveau on détermine les forces de contrôle locales pour que les robots suiveurs exécutent les stratégies de contrôle qui leur permettent d'avancer aux positions montrées et de garder alors leurs positions sans heurter des obstacles ou d'autres robots le long de leurs chemins. Le comportement est décomposé en modes de contrôle

élémentaires : évitement de collision, positionnement, maintien de position. Le niveau de contrôle est responsable des passages parmi les différents modes.

**Niveau d'entité :** Dans ce niveau, on décrit les variables d'état dans la dynamique d'un robot individuel. À ce niveau, non seulement les commandes de mouvement sont exécutées, mais également les positions de référence des robots sont calculées.



**Figure 2-3 : Architecture a quatre niveaux de l'approche leader-follower.**

### 2.3.3 EVITEMENT D'OBSTACLE

Le principe de l'algorithme d'évitement d'obstacle proposé pour l'approche Leader - Follower est très simple, il s'agit de :

- Détection d'obstacle : on utilise trois capteurs de proximité pour calculer la distance à l'obstacle, placés à 0 [deg], 60 [deg] et -60 [deg] par rapport à la direction du robot afin de détecter les obstacles dans les trois directions : avant, droit et gauche. (**Figure 2-4**)
- Condition d'évitement : Après avoir calculé la distance à l'obstacle, si elle est inférieure à une distance de sécurité le robot doit éviter l'obstacle sinon le robot continue son mouvement dans la même direction.
- Choix de la direction : dans l'évitement d'obstacle, le robot doit faire un choix entre les deux directions : droit ou gauche, ce choix est basé sur deux critères : le premier, c'est l'inexistence d'obstacle dans la direction à choisir, le deuxième, c'est la distance minimale vers le but.

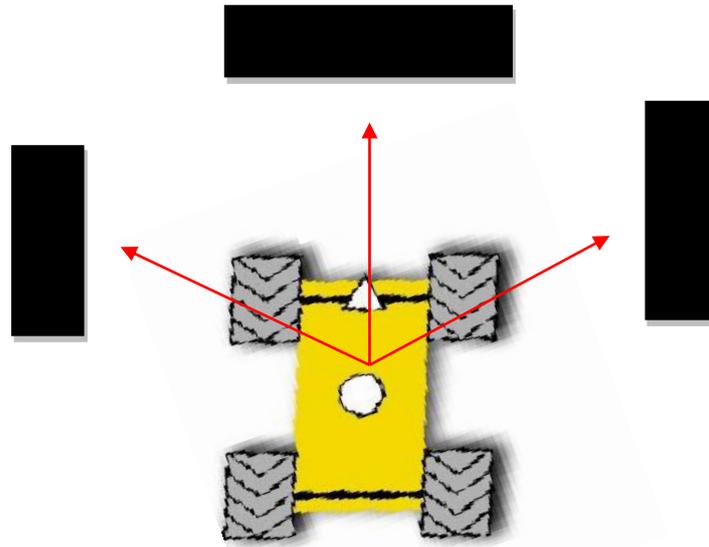


Figure 2-4 : Détection d'obstacle.

## 2.4 APPROCHE COMPORTEMENTAL METHODE CHAMPS DE POTENTIEL

Quand on pense au champ de potentiel, la première idée qui vient à l'esprit est une particule chargée se déplaçant dans un champ magnétique ou bien une balle qui se dirige vers le bas d'une colline. L'idée de base est le comportement montré par la particule ou la balle qui dépend de la forme du champ ou de la colline. À la différence des collines où la topologie est spécifiée par des conditions environnementales extérieures, la topologie des champs de potentiel qu'un robot peut observer est déterminée par le concepteur. Plus particulièrement, le concepteur [13] :

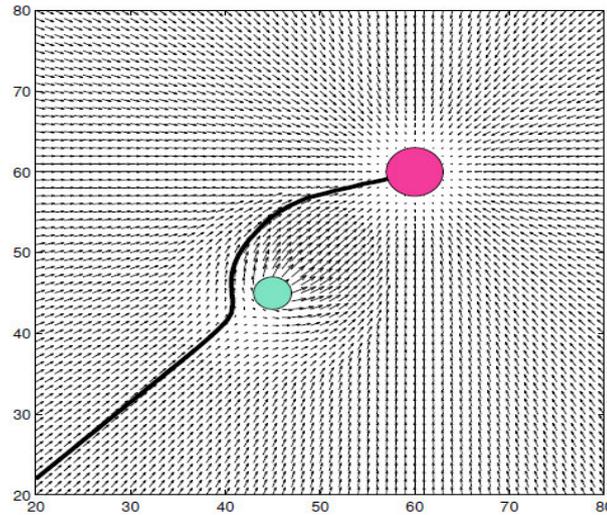
- Crée plusieurs comportements, chaque comportement est assigné à une tâche particulière ou fonction.
- Il représente chacun de ces comportements par un champ de potentiel.
- Il combine tous ces comportements pour produire le mouvement des robots en combinant les champs de potentiel.

Nous allons traiter deux comportements très simples à savoir :

- Aller vers un but « SeekGoal » (potentiel attractif).

- Evitement des obstacles (potentiel répulsif).

La **Figure 2-5** illustre le résultat des deux comportements dans un environnement.



**Figure 2-5 : Approche champs de potentiel avec deux comportements (attractif et répulsif)**

#### 2.4.1 DEFINITION

La méthode des champs de potentiel consiste à remplir l'espace de travail du robot avec des champs de potentiels artificiels, dans lequel le robot est attiré vers la position but (potentiel attractif) et repoussé par les obstacles (potentiel répulsif). La combinaison de la force attractive vers la position but et la force répulsive des obstacles assure des trajectoires sécurisées vers le point but.

#### 2.4.2 PLANIFICATION DE TRAJECTOIRE PAR LA METHODE DES CHAMPS POTENTIELS

Dans la planification de trajectoire, le robot est considéré dans l'espace de configuration  $C$  comme une particule soumise à un champ potentiel artificiel  $U(\mathbf{q})$ , avec  $\mathbf{q}$  est l'état du robot (typiquement, pour un robot mobile  $\mathbf{q} = (\mathbf{x} ; \mathbf{y})$ ). À chaque itération, la force artificielle  $\mathbf{F}(\mathbf{q}) = -\Delta U(\mathbf{q})$  induite par le champ potentiel indiquera alors la direction la plus prometteuse.

$$\mathbf{F}(\mathbf{q}) = -\Delta U(\mathbf{q}) = \left( \frac{\partial U}{\partial x}, \frac{\partial U}{\partial y} \right)^T \quad (2.4)$$

#### 2.4.3 POTENTIEL ATTRACTIF

La méthode de champs des potentiels est basée principalement sur la génération d'un vecteur d'action qui correspond à la vitesse et l'orientation du robot mobile.

Le comportement attractif agit sur le robot d'une manière à l'amener au but désiré en chaque point de l'espace le gradient du champ est dirigé vers le but.

La forme de la région d'attraction est un cercle de rayon  $r$ . La valeur du potentiel dans la région d'attraction est constante et croît à l'extérieur du cercle. Le champ de potentiel qui agit sur les robots est donné par l'équation suivante [14]:

$$U_a = \begin{cases} 0 & \text{si } 0 < S_i < r \\ \alpha_a(S_i - r)^3 & \text{si } S_i \geq r \end{cases} \quad (2.5)$$

Avec  $\alpha_a$  est le facteur qui détermine la pente du potentiel à l'extérieur de la région d'attraction,  $S_i$  est la distance entre le  $i^{\text{ème}}$  robot et le centre de la région d'attraction. La fonction du potentiel artificiel donné par l'équation précédente est représentée par la **Figure 2-6**.

La force générée par ce champ de potentiel est donnée par l'équation (2.6) :

$$F_a = \begin{cases} 0 & \text{si } 0 < S_i < r \\ \frac{3}{2}\alpha_a(S_i - r) & \text{si } S_i \geq r \end{cases} \quad (2.6)$$

Soit «  $\mathbf{q}$  » le vecteur de position du robot mobile  $\mathbf{q} = [x, y]^T$ ,  $\Delta$  le gradient de la position tel que :  $\Delta = [\Delta x, \Delta y]^T$ ,  $(x_G, y_G)$  la position du but et  $r$  son diamètre.

- On calcule la distance «  $d$  » entre l'agent et le but :  $S_i = \sqrt{(x_G - x)^2 + (y - y_G)^2}$ .
- On calcule l'angle entre l'agent et le but thêta «  $\theta$  » :  $\theta = \tan^{-1}\left(\frac{y_G - y}{x_G - x}\right)$ .
- Calcule  $\Delta x$  et  $\Delta y$  tel que :

$$\begin{array}{ll} \text{Si } S_i < r & \Delta_x = \Delta_y = 0 \\ \text{Si } r \leq S_i \leq s + r & \begin{cases} \Delta_x = \alpha_a \cdot (S_i - r) \cdot \cos(\theta) \\ \Delta_y = \alpha_a \cdot (S_i - r) \cdot \sin(\theta) \end{cases} \\ \text{Si } S_i > s + r & \begin{cases} \Delta_x = \alpha_a \cdot s \cdot \cos(\theta) \\ \Delta_y = \alpha_a \cdot s \cdot \sin(\theta) \end{cases} \end{array} \quad (2.7)$$

Quand l'agent atteint le but il ne subira aucune force, cela si la distance  $S_i < r$  donc  $\Delta x$  et  $\Delta y$  sont égales à zéro, en dehors de la cercle de rayon  $s + r$  l'attraction du champ est maximal. (**Figure 2-7**) [13]

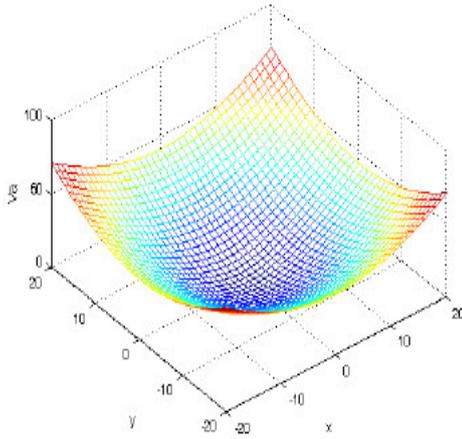


Figure 2-7: Représentation 3D du potentiel attractif

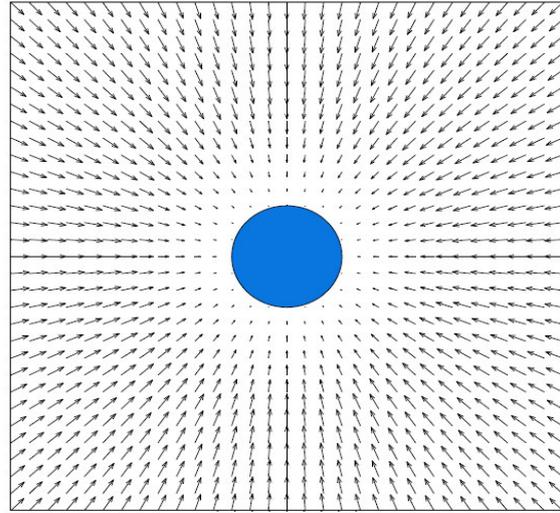


Figure 2-6 : Gradient du champ attractif

#### 2.4.4 POTENTIEL REPULSIF

Le potentiel répulsif permet d'éloigner le robot mobile de l'obstacle. La valeur du potentiel est maximale lorsque le robot mobile est proche de l'obstacle et elle diminue lorsque le robot mobile s'éloigne de ce dernier (**Figure 2-9**). De plus, on ne veut pas que ce potentiel affecte le mouvement du robot mobile lorsque celui-ci est suffisamment loin des obstacles.

Le champ de potentiel artificiel qui agit sur le robot quand il est près de l'obstacle est donné par l'équation suivante :

$$U_r = \begin{cases} \infty & 0 < g_{ik} < r_{o_k} \\ \frac{\alpha_r}{||g_{ik}|| - r_{o_k}} + g_{ik} \cdot \frac{\alpha_r}{(m_k - r_{o_k})^2} & r_{o_k} < g_{ik} < m_k \\ \alpha_r \cdot \frac{2 \cdot m_k - r_{o_k}}{(m_k - r_{o_k})^2} & g_{ik} \geq m_k \end{cases} \quad (2.8)$$

Avec  $r_{o_k}$  le rayon de la région interdite qui entoure le  $k^{\text{ème}}$  obstacle,  $m_k$  est le rayon de la région autour du  $k^{\text{ème}}$  obstacle où la répulsion agit (la répulsion diminue à l'extérieur),  $g_{ik}$  est la distance entre le  $i^{\text{ème}}$  robot et le centre de la  $k^{\text{ème}}$  région interdite et  $\alpha_r$  détermine la pente du potentiel artificiel autour de la région interdite. La valeur du potentiel artificiel dans la région interdite n'est pas définie.

La fonction potentielle a une pente considérable près des frontières de la région interdite pour empêcher les robots d'accéder à l'intérieur. La force répulsive donnée par l'équation (2.9) est présentée dans la **Figure 2-8**.

$$F_r = \begin{cases} \infty & \mathbf{0} < g_{ik} < r_{ok} \\ -\frac{\alpha_r}{(g_{ik} - r_{ok})^2} + \frac{\alpha_r}{(m_k - r_{ok})^2} & r_{ok} < g_{ik} < m_k \\ \mathbf{0} & g_{ik} \geq m_k \end{cases} \quad (2.9)$$

Soit «  $\mathbf{q}$  » le vecteur de position du robot mobile  $\mathbf{q} = [x, y]^T$ ,  $\Delta$  le gradient de la position tel que :  $\Delta = [\Delta x, \Delta y]^T$ ,  $(x_{ob}, y_{ob})$  la position de l'obstacle et  $r_{ok}$  son diamètre :

➤ On calcule la distance «  $g_{ik}$  » entre l'agent et l'obstacle :

$$g_{ik} = \sqrt{(x_{ob} - x)^2 + (y - y_{ob})^2}$$

➤ On calcule l'angle entre l'agent et l'obstacle  $\theta$  :  $\theta = \tan^{-1}\left(\frac{y_{ob} - y}{x_{ob} - x}\right)$ .

➤ Calcule  $\Delta x$  et  $\Delta y$  tel que :

$$\begin{aligned} \text{Si } g_{ik} < r_{ok} & \begin{cases} \Delta_x = -\text{sign}(\cos(\theta)) \cdot \infty \\ \Delta_y = -\text{sign}(\sin(\theta)) \cdot \infty \end{cases} \\ \text{Si } r_{ok} \leq g_{ik} \leq s + r_{ok} & \begin{cases} \Delta_x = -\alpha_a \cdot (s + r_{ok} - g_{ik}) \cdot \cos(\theta) \\ \Delta_y = -\alpha_a \cdot (s + r_{ok} - g_{ik}) \cdot \sin(\theta) \end{cases} \\ \text{Si } g_{ik} > s + r_{ok} & \Delta_x = \Delta_y = \mathbf{0} \end{aligned} \quad (2.10)$$

À l'intérieur de l'obstacle, le champ potentiel répulsif est infini et se dirige vers l'extérieur de l'obstacle. En dehors du cercle d'influence, le potentiel répulsif est égal à zéro. Dans le cercle d'influence, mais en dehors du rayon de l'obstacle, la magnitude du vecteur se développe à partir de zéro lorsque  $g_{ik} = s + r_{ok}$  c'est le cas où l'agent est sur le bord du cercle d'influence. [13]

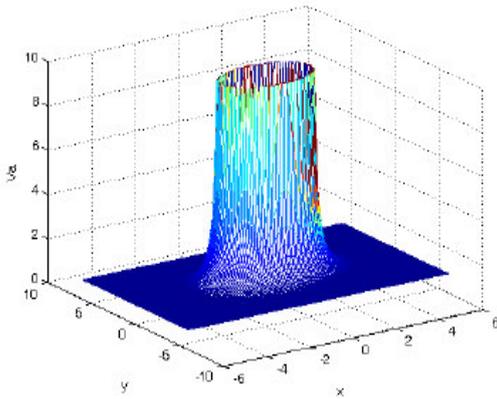


Figure 2-8: Représentation 3D du potentiel répulsif

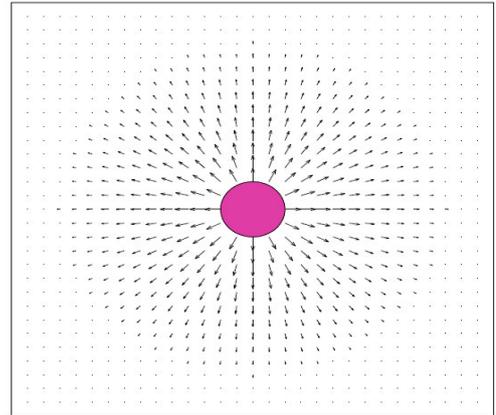


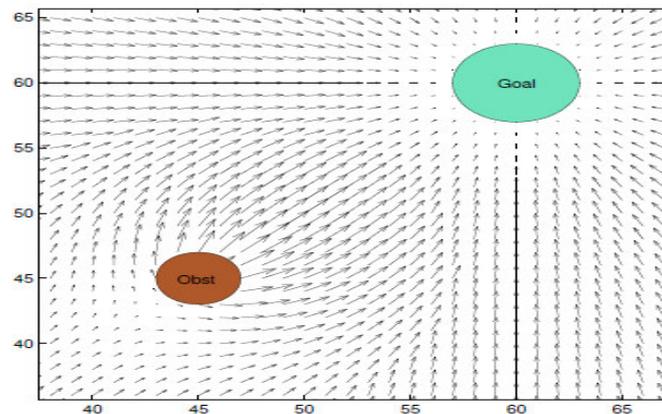
Figure 2-9: Gradient du champ répulsif

### 2.4.5 COMBINAISON DES DEUX COMPORTEMENTS ATTRACTIF ET REPULSIF

Le robot mobile a deux mission à accomplir : aller vers un but « SeekGoal » (potentiel attractif) et l'évitement des obstacles (potentiel répulsif) cela est attient avec la combinaison de potentiel attractif et répulsif donc à chaque instant, la force résultante agissant sur le robot est la somme de la force attractive et la force répulsive (**Figure 2-10**).

$$\vec{F}_{totale} = \vec{F}_{attractive} + \vec{F}_{repulsive}$$

La planification du mouvement se déroule d'une façon itérative. À chaque itération, la force résultante dans la position courante indique la direction la plus engageante du déplacement et la génération du mouvement aura lieu le long de cette direction.



**Figure 2-10: Combinaison d'un champ attractif et un champ répulsif**

### 2.4.6 GENERATION DE LA FORMATION

L'objectif est de diriger le robot mobile vers une position désirée en évitant les obstacles. On distingue quatre types d'environnement de navigation pour le robot : un environnement connu et statique, un environnement inconnu et statique, un environnement connu et dynamique et finalement un environnement inconnu et dynamique. Dans le cas d'une formation de robot chacun des robots est dans une zone de travail dynamique avec les autres robots mobiles donc notre environnement est inconnu et dynamique. La construction du modèle géométrique de la formation passe par trois étapes :

- a. Définition du modèle géométrique : V inversé, ligne, colonne ou diamant.
- b. Planification de trajectoire pour chaque robot.
- c. Contrôle de la formation.

### **2.4.7 APPLICATION DE LA METHODE DE CHAMPS DE POTENTIELS POUR LA COMMANDE D'UNE FORMATION**

Nous allons introduire la méthode des champs de potentiel dans la commande d'une formation pour créer des forces attractives entre chaque robot et sa position désirée dans la formation et des forces répulsives entre les robots d'une part et entre les robots et l'environnement d'autre part. La formation est modélisée par un système masse-ressort, où on crée entre les robots des forces d'attraction si l'un sort de la formation et des forces de répulsion si un robot approche d'un autre. Mais pour plus de souplesse et pour minimiser les calculs : nous utilisons un modèle de groupement en créant seulement des forces attractives entre les robots et leurs positions désirées dans la formation. La fonction de répulsion est assurée par la fonction anticollision puisque chaque robot considère son voisin comme un obstacle.

## **2.5 APPROCHE PROPOSEE LEADER-FOLLOWER / LOGIQUE**

### **FLOUE**

La logique floue a été introduite par le professeur Lotfi Zadeh professeur à l'université de Californie à Berkeley en 1965, cette logique stipule qu'un élément peut plus ou moins appartenir à certain ensemble avec un degré d'appartenance donné, donc les imprécisions et les incertitudes peuvent ainsi être modélisées et on aura une flexibilité que ne permet pas la logique classique. Cette force a suscité beaucoup d'intérêt dans le domaine du contrôle.

C'est justement pour ces raisons qu'on s'intéresse à élaborer une commande par logique floue pour la commande d'une formation de robots mobiles, en utilisant l'approche leader-follower avec la commande  $l - \varphi$ .

### **2.5.1 SYNTHÈSE DE LA LOI DE COMMANDE LEADER-FOLLOWER /LOGIQUE FLOUE**

Le but de la loi de la commande basée sur l'approche leader-follower est de préserver une distance constante et un angle constant du follower par rapport à son leader. Pour cela nous générons un vecteur de module «  $k$  » et d'orientation «  $\Psi$  », tel que l'orientation «  $\Psi$  » ce vecteur relie le point actuel et le but (emplacement du robot dans la formation), «  $k$  » qui représente la vitesse de translation est calculé en utilisant un contrôleur floue qui a comme

entrée l'erreur sur la distance et sa dérivée. Les différents paramètres du régulateur flou sont les suivants :

$$\begin{cases} \Delta d = d_{des} - d_{lf} \\ \dot{\Delta d} = \Delta d(k) - \Delta d(k-1) \end{cases}$$

Les fonctions d'appartenance des variables d'entrées  $\Delta d$  et  $\dot{\Delta d}$  et la variable de sortie  $k$  sont fuzzifiées chacune en cinq variables linguistiques : négative grande (**NG**), négative petite (**NP**), nulle (**Z**), positive petite (**PP**) et positive grande (**PG**).

Pour chaque entrée on a cinq variables linguistiques donc on aura 25 règles floues pour commander la vitesse de translation  $k$  du robot mobile.

Le tableau ci-dessous montre les **25** règles floues pour commander la vitesse de translation  $k$ .

$\dot{\Delta d} \backslash \Delta d$	<b>NG</b>	<b>NP</b>	<b>Z</b>	<b>PP</b>	<b>PG</b>
<b>NG</b>	<b>PG</b>	<b>PG</b>	<b>PG</b>	<b>PP</b>	<b>Z</b>
<b>NP</b>	<b>PG</b>	<b>PG</b>	<b>PP</b>	<b>Z</b>	<b>NP</b>
<b>Z</b>	<b>PG</b>	<b>PP</b>	<b>Z</b>	<b>NP</b>	<b>NG</b>
<b>PP</b>	<b>PP</b>	<b>PP</b>	<b>NP</b>	<b>NP</b>	<b>NG</b>
<b>PG</b>	<b>Z</b>	<b>Z</b>	<b>NG</b>	<b>NG</b>	<b>NG</b>

**Tableau 1: Règles floues.**

## 2.6 APPROCHE HYBRIDE

Dans cette partie nous allons proposer une approche pour la commande d'une formation de robot mobile afin d'éviter les problèmes majeurs rencontrés dans chaque approche : l'approche leader-follower et l'approche champs de potentiel. Nous nous sommes engagé dans la réalisation d'une architecture de commande « hybride », pouvant donner de bonne performance afin d'atteindre les objectifs désirées (maintien de la formation, suivi de trajectoire et l'évitement d'obstacle).

Nous commençons par citer les inconvénients des approches précédemment présentées : les limitations de l'approche leader follower et de l'approche comportementale, ensuite nous décrivons l'algorithme proposé pour l'hybridation.

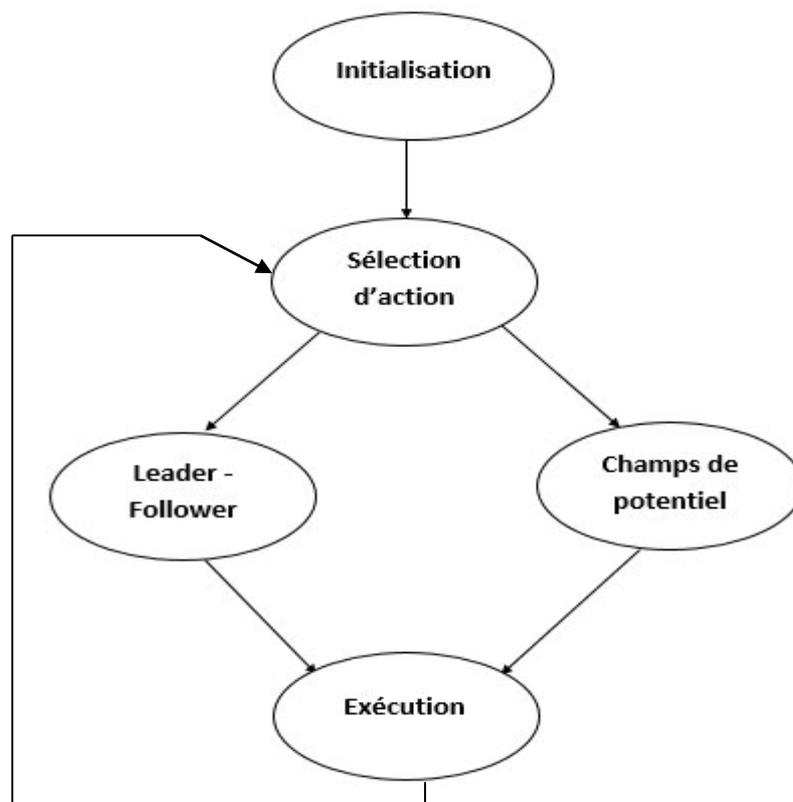
### 2.6.1 LIMITATIONS DES DEUX APPROCHES

L'algorithme leader-follower est largement utilisé dans la commande d'une formation de robot, ce dernier assure un bon maintien de la géométrie de la formation et le suivi de trajectoire mais il donne lieu à de fortes oscillations du mouvement autour des obstacles.

Par contre, la méthode des champs de potentiel garantit le maximum de sécurité des robots mobiles en cours de déplacement en formation : évitement d'obstacle et évitement de collision entre robot, or que cette dernière sujette à l'inconvénient d'interférence des champs de potentiel, par conséquent la convergence vers les but n'est pas assurée.

### 2.6.2 FONCTIONNEMENT DE L'APPROCHE HYBRIDE

L'approche hybride a comme objectifs d'éliminer les limitations des deux approches présentées précédemment, en faisant une commutation entre les deux commandes coopératives de la manière suivante : on assure le maintien de la configuration géométrique et le suivi de trajectoire par l'approche leader follower/logique floue, et l'évitement d'obstacle par l'approche des champs de potentiel. Où le critère de sélection d'action est la distance entre les robots et les obstacles présents dans l'environnement de la formation (**Figure 2-11**).



**Figure 2-11 : Logigramme de l'approche hybride.**

**Etape initialisation :** Cette étape s'exécute une seule fois dans l'algorithme où nous définissons la géométrie désirée de la formation et nous générons la trajectoire à suivre.

**Etape sélection d'action :** le critère de la sélection d'action est la distance entre le robot et l'obstacle, si elle est inférieure à la distance critique on exécute l'algorithme de champs de potentiel sinon on exécute l'algorithme de leader follower.

**Etape leader-follower :** on fait appel à l'algorithme leader follower pour générer les nouvelles coordonnées des robots pour suivre la trajectoire prédéfinie.

**Etape Champs de potentiel :** on fait appel à l'algorithme de champs de potentiel pour générer les nouvelles coordonnées des robots afin d'éviter les obstacles et la collision entre robots.

**Etape exécution :** on exécute la commande générée dans l'étape précédente.

## 2.7 CONCLUSION

Dans ce chapitre nous avons présenté quelques travaux concernant l'application des approches de commandes coopératives sur un groupe de robots mobiles à savoir : l'approche leader-follower et l'approche comportementale champs de potentiel. Puis les approches proposées : l'approche leader-follower/logique floue et l'hybridation entre l'approche leader-follower et l'approche comportementale champs de potentiel.

Le premier travail concerne l'approche leader-follower qui sert à commander une formation de robot mobile. Le leader seul possède l'information sur la trajectoire de référence et il envoie les coordonnées désirées aux followers afin d'assurer une configuration géométrique désirée. Le deuxième travail concerne l'approche comportementale « champs de potentiel » qui permis de réaliser un ensemble de tâches. Les robots essaient de rattraper une formation virtuelle, tout en gardant la configuration géométrique désirée. Le troisième travail concerne une présentation générale des concepts de la logique floue et leurs utilisations afin d'améliorer les performances de l'algorithme de leader-follower : accélération de la convergence vers le but des robots, diminution de l'erreur sur la distance entre le leader et les followers. Le quatrième travail concerne l'approche hybride où nous avons utilisé les deux approches leader follower et champ de potentiel afin d'atteindre les objectifs désirées (maintien de la formation, suivi de trajectoire et l'évitement d'obstacle) avec les meilleures performances

Le prochain chapitre est consacré aux résultats et simulations des différentes approches de commande de formation de robots mobiles pour valider les algorithmes présentés.

# **Chapitre 3 :**

## **Tests et résultats de simulation**

## 3 TESTS ET RESULTATS DE SIMULATION

### 3.1 INTRODUCTION

Dans ce chapitre, nous allons présenter les résultats de simulations obtenus des approches présentées dans le chapitre précédent, afin de consolider la validation des résultats théoriques. Les résultats qui vont être présentés concernant :

- Les simulations de la commande coopérative d'une formation de robots mobiles en utilisant l'approche leader-follower  $L - \varphi$ .
- Les simulations de la commande coopérative d'une formation de robots mobiles en utilisant l'approche comportementale (champs de potentiels).
- Les simulations de la commande coopérative d'une formation de robots mobiles issue de l'application de la logique floue sur l'approche leader-follower.
- Les simulations de la commande coopérative d'une formation de robots mobiles en utilisant l'hybridation entre l'approche leader-follower et l'approche comportementale (champs de potentiels).

### 3.2 COMMANDE D'UNE FORMATION EN UTILISANT

#### L'APPROCHE LEADER-FOLLOWER

##### 3.2.1 MAINTIEN DE LA FORMATION

Notre but est d'atteindre une configuration triangle (prédéfinie), en faisant déplacer les robots mobiles de leurs postures initiales vers un but désiré, notre formation contient six robots mobiles qui forment le triangle montrée dans la **Figure 3-1**. En considérant qu'on a un seul leader (en bleu) la distance désirée «  $l$  » et l'angle désirée «  $\psi$  » entre chaque follower et le leader est : follower 1 : (20 cm, 135 deg), follower 2 : (20 cm, -135 deg), follower 3 : (40 cm, 135 deg), follower 4 : (35 cm, 180 deg) et follower 5 : (40 cm, -135 deg).

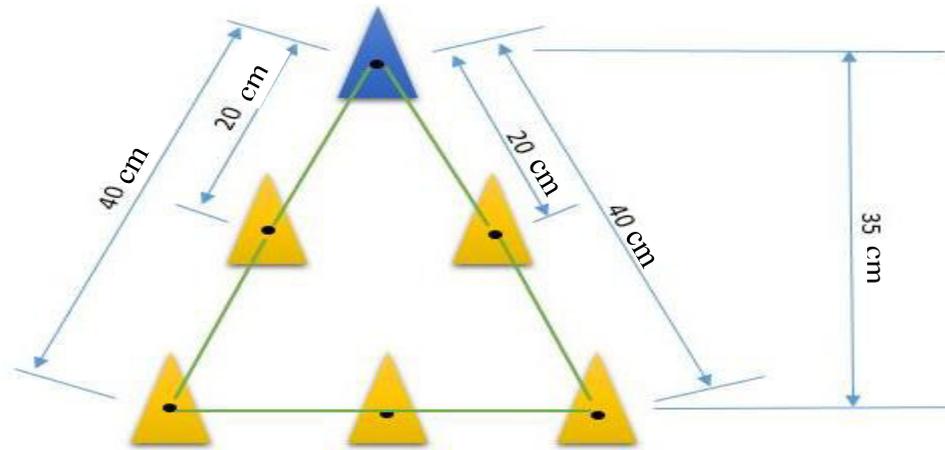


Figure 3-1 : Configuration géométrique « triangle ».

Nous montrons respectivement le maintien de la configuration triangle de la formation (**Figure 3-2**), la distance entre chaque follower et le leader (**Figure 3-3**), les erreurs sur la distance entre le leader et les followers (**Figure 3-4**) et les erreurs sur l'angle entre chaque follower et le leader (**Figure 3-5**).

Dans la **Figure 3-2** on remarque que les robots mobiles ont réussi à former le triangle (la configuration géométrique désirée).

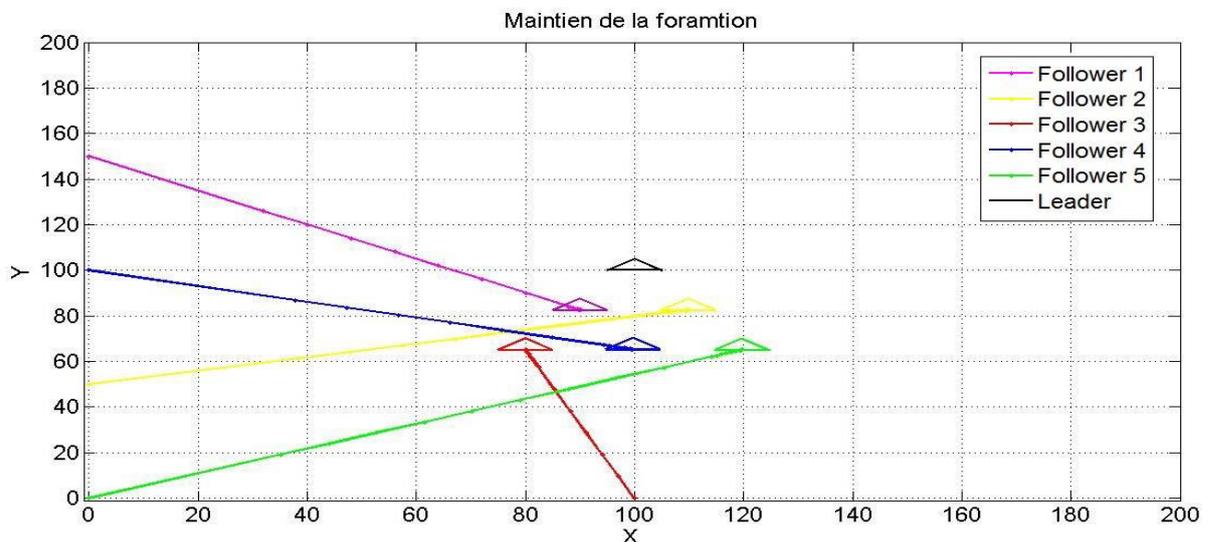
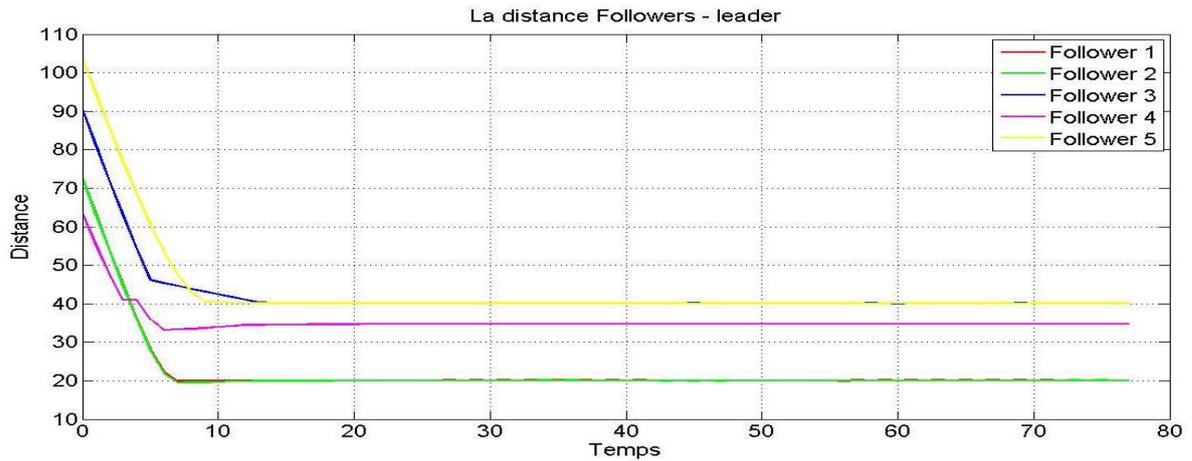


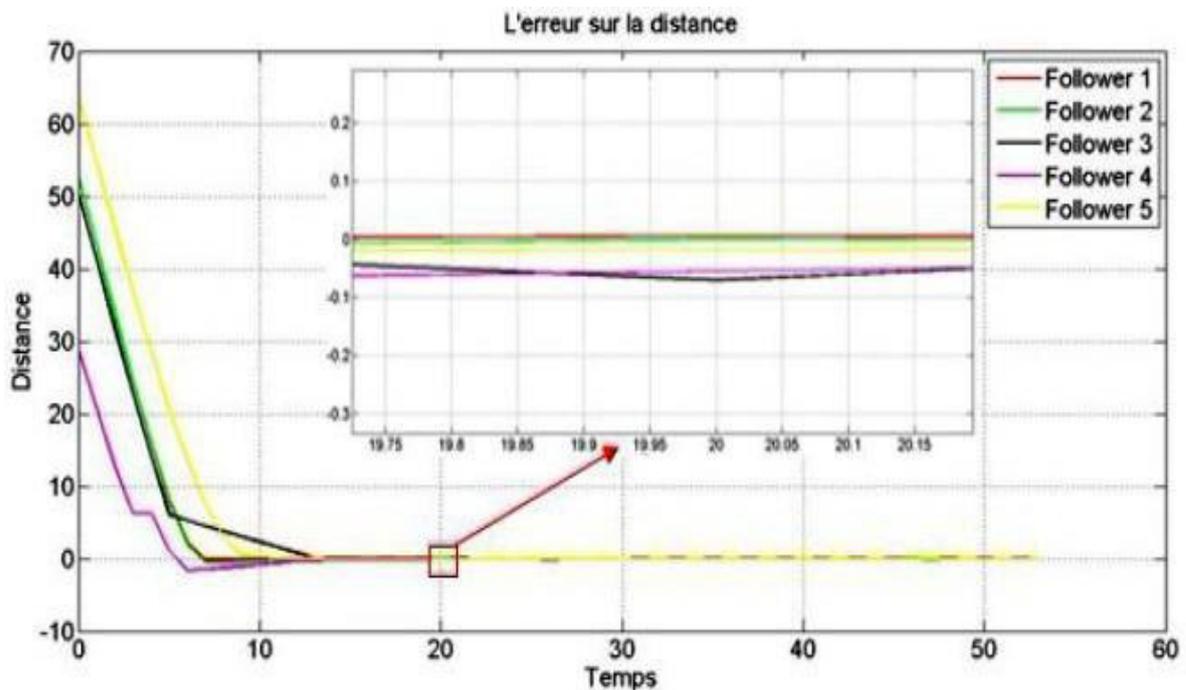
Figure 3-2 : Maintien de la formation « configuration triangle ».



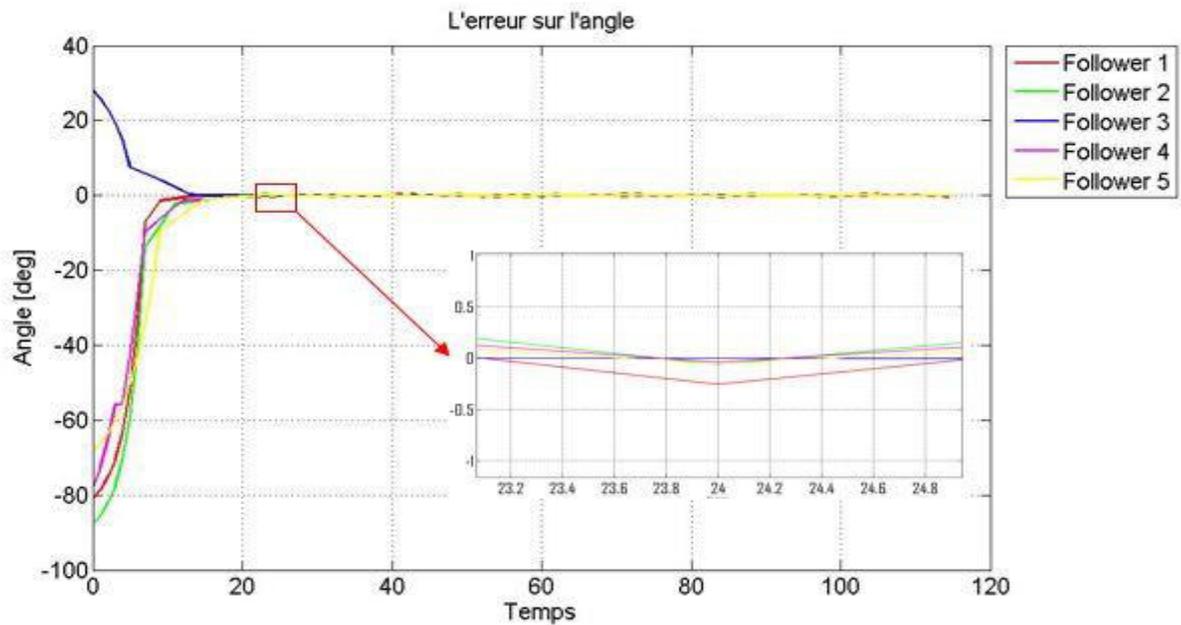
**Figure 3-3 : Distance entre le leader et les followers dans la configuration triangle**

La **Figure 3-3** montre que la distance entre les robots mobiles et le leader tend vers la distance désirée dans un temps de 12 secondes.

Nous remarquons dans la **Figure 3-4** et la **Figure 3-5**, que l’erreur sur la distance et sur l’angle relative est une fonction décroissante, elle commence par une grande valeur qui dépend de la posture initiale du robot puis elle diminue jusqu’à atteindre une valeur négligeable : moins que 0.1 [cm] pour la distance et moins que 0.3 [deg] pour l’angle qui nous a donné une formation stable. Cela montre l’efficacité de l’approche leader follower dans le maintien de la formation.



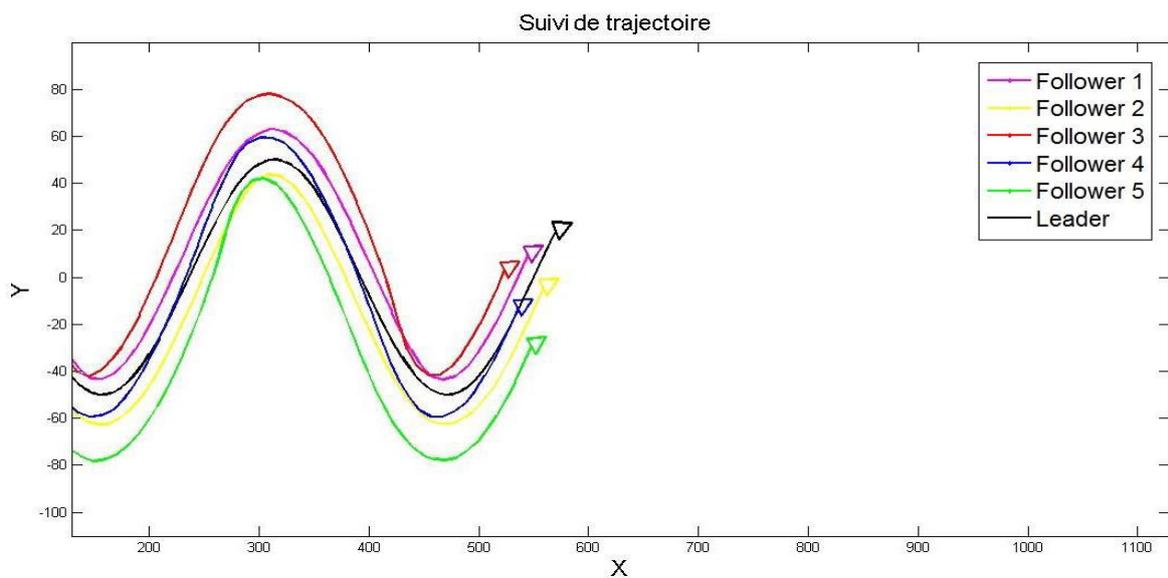
**Figure 3-4 : Erreur sur la distance**



**Figure 3-5 : Erreur sur l'angle relative**

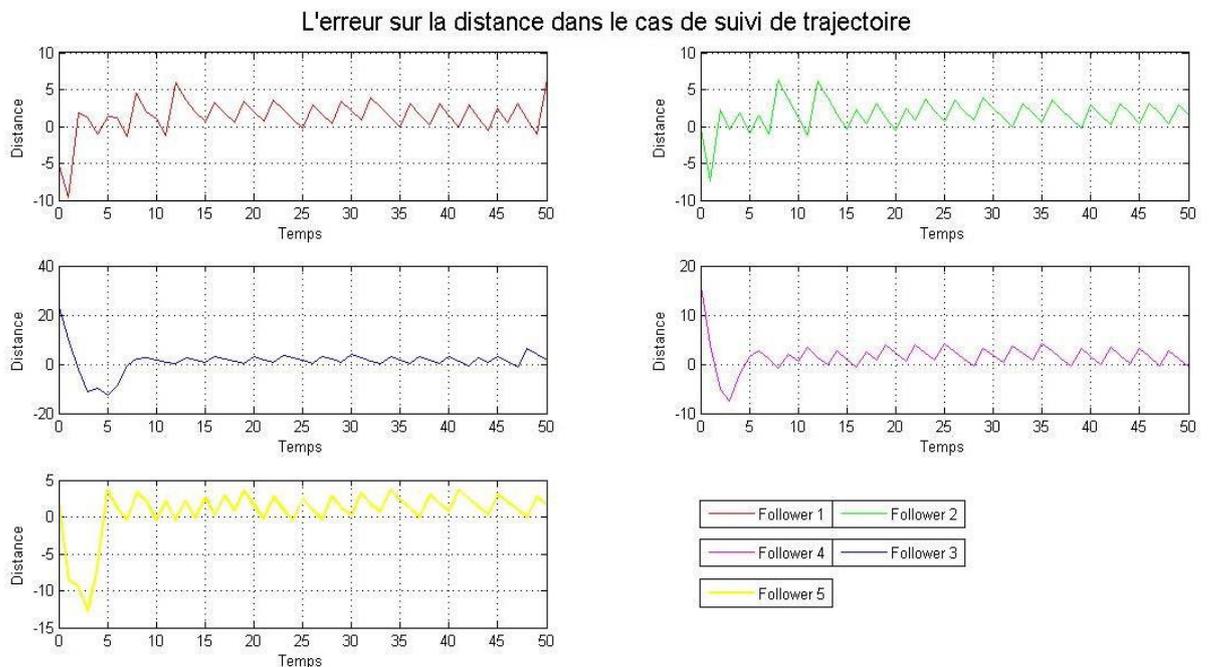
### 3.2.2 SUIVI DE TRAJECTOIRE

Dans cette partie nous avons simulé une flottille de six robots mobiles évoluant en formation triangle, le robot leader suit une trajectoire prédéfinie. Nous montrons respectivement, les trajectoires des robots (**Figure 3-6**), l'erreur sur la distance entre chaque robot follower et le leader (**Figure 3-7**) et l'erreur sur l'angle entre les robots followers et le leader (**Figure 3-8**).

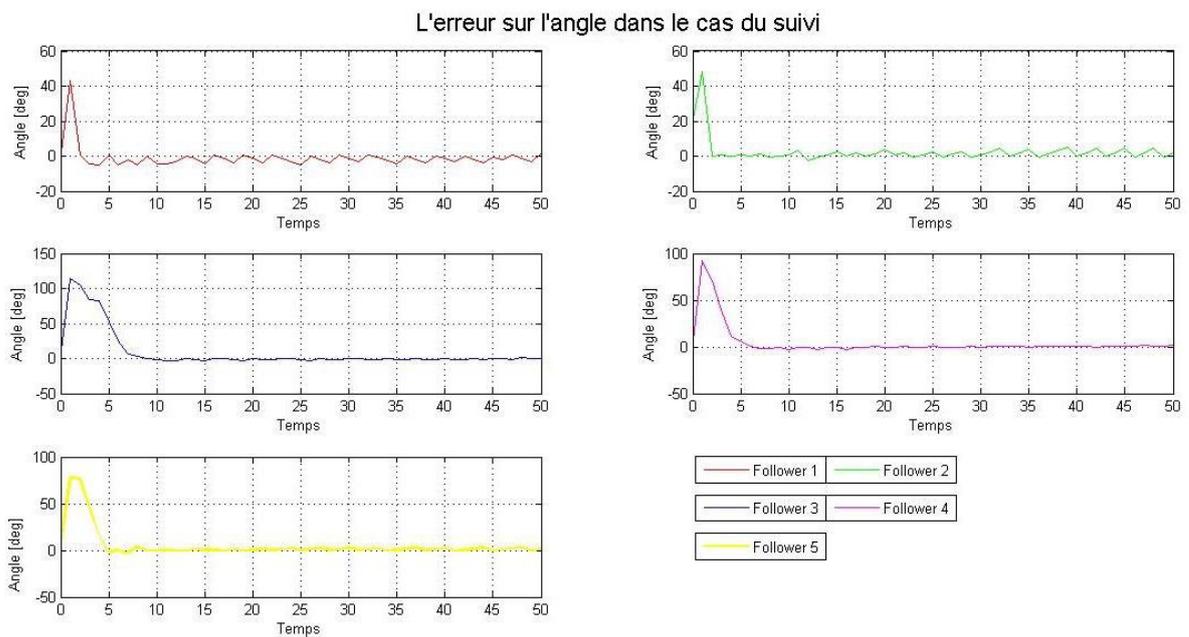


**Figure 3-6 : Suivi de trajectoire en formation triangle.**

Dans la **Figure 3-6**, le leader suit une trajectoire de référence (en noir) et les followers planifient leurs mouvements en fonction de la position du leader afin de garder la formation triangle, nous remarquons que cette approche assure un bon suivi de trajectoire en gardant la forme géométrique de la formation.



**Figure 3-7 : Erreur sur la distance entre le leader et les followers.**



**Figure 3-8 : Erreur sur l'angle entre le leader et les followers.**

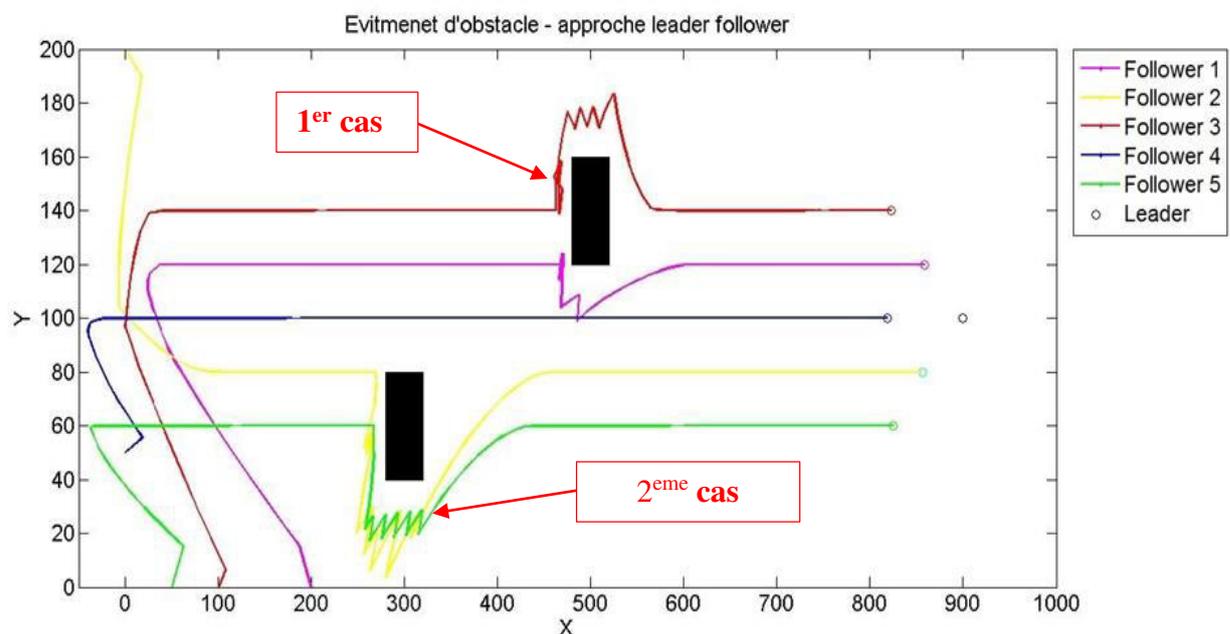
La **Figure 3-7** et la **Figure 3-8** montrent que l'erreur sur la distance et l'erreur sur l'angle diminue pour les cinq followers. On remarque au départ une grande valeur de l'erreur

cela est causé par la position initial du chaque follower, puis l'erreur varier entre 0 et 5 [cm]. Par contre la variation de l'erreur sur l'angle est quasiment négligeable.

Afin d'éliminer cette variation de l'erreur sur la distance dans cette approche on a opté à utiliser la logique floue.

### 3.2.3 EVITEMENT DES OBSTACLES

Notre but est d'éviter les obstacles que rencontre la formation, nous avons dans notre environnement deux obstacles. La configuration désirée (triangle) de la formation est gardée la même. Nous montrons les trajectoires des robots durant l'évitement des obstacles dans la **Figure 3-9**.



**Figure 3-9 : Evitement des obstacles avec l'approche leader follower.**

La formation doit se déformer pour que chaque robot puisse éviter les obstacles présents dans son chemin, une fois les obstacles sont évités la formation se régénère à nouveau. On remarque aussi que la manière dont les robots évitent les obstacles est imparfaite :

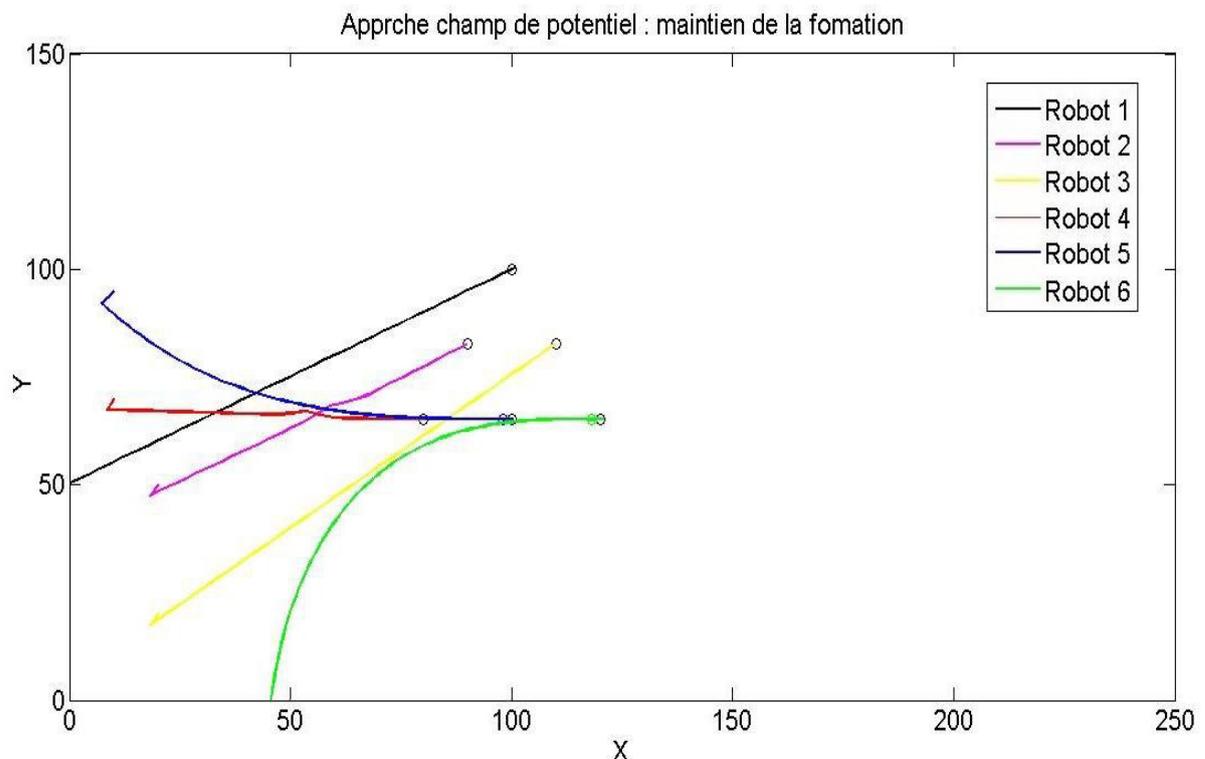
- 1<sup>er</sup> cas : Une singularité dans l'algorithme où la prochaine position du robot mobile dans la formation se trouve à l'intérieur de l'obstacle, alors le robot fait des oscillations jusqu'à recevoir de nouveau une position en dehors de l'obstacle.
- 2<sup>eme</sup> cas : De fortes oscillations dans l'évitement de l'obstacle : cause par la permutation entre la tâche d'évitement et la tâche de suivi, par exemple, le

robots 6(en vert) une fois la distance entre lui et l'obstacle est inférieure à la distance critique le robot entre permute du régime suivi vers le régime évitement. Mais à la prochaine itération il sera à l'extérieur du zone critique il revient au régime suivi de trajectoire bien qu'il n'a pas évité l'obstacle complètement ce qui va lui obliger a rentre la zone critique.

### 3.3 COMMANDE D'UNE FORMATION EN UTILISANT L'APPROCHE COMPORTEMENTALE CHAMPS DE POTENTIEL

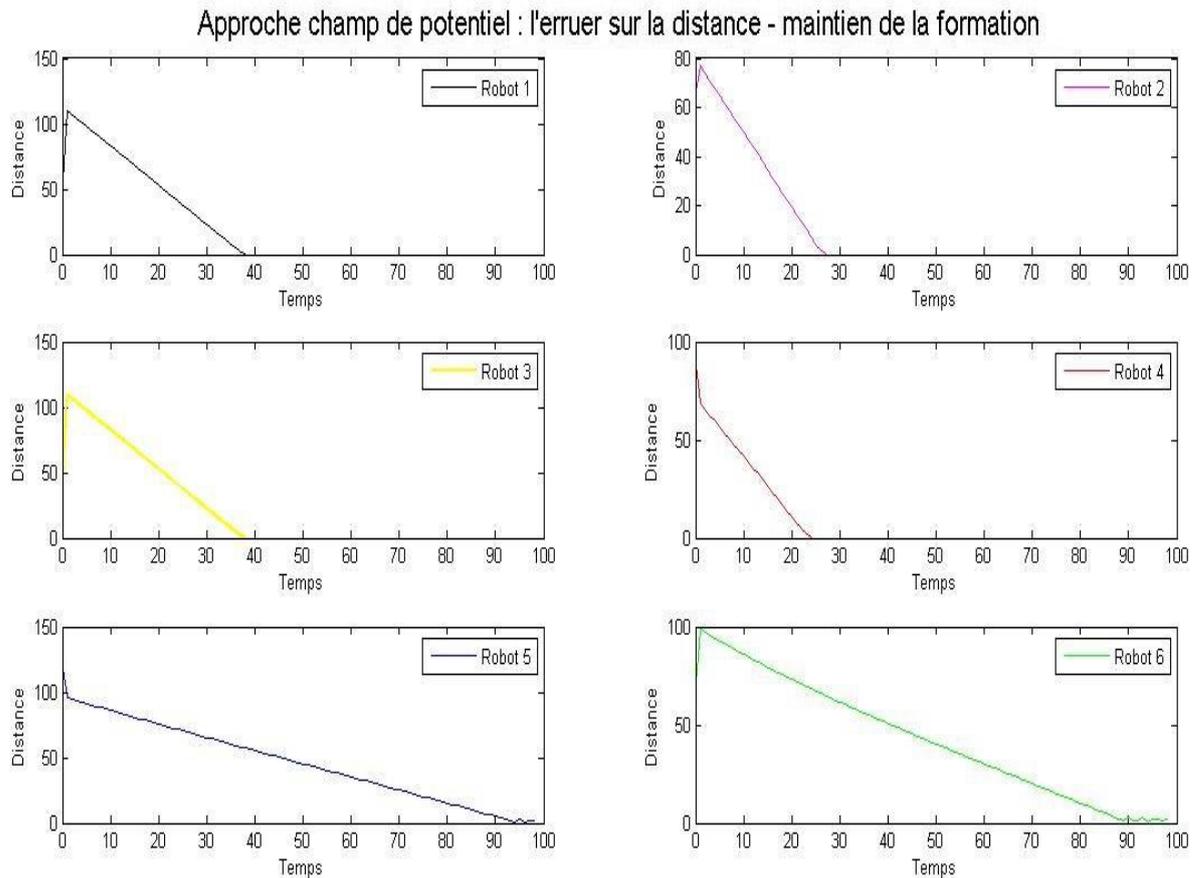
#### 3.3.1 MAINTIEN DE LA FORMATION

Pour vérifier le cas du maintien de la formation avec l'approche des champs de potentiel, nous allons simuler une formation triangle de six robots mobiles, comme c'est montré dans la **Figure 3-1**. Pour chaque robot mobile nous allons considérer le comportement « go to goal » où le but est sa position désirée dans la formation et le comportement « avoid obstacle » où les obstacles sont les autres robots mobiles constituant la formation afin d'éviter la collision entre eux.



**Figure 3-10 : Maintien de la formation avec l'approche champs de potentiel.**

Dans la **Figure 3-10** les cercles en noir représentent les positions désirées « goal » pour avoir une configuration « triangle », on remarque que les robots oscillent autour du but avant de converger à cause de l'influence des forces répulsives générées par chaque robot dans la formation.

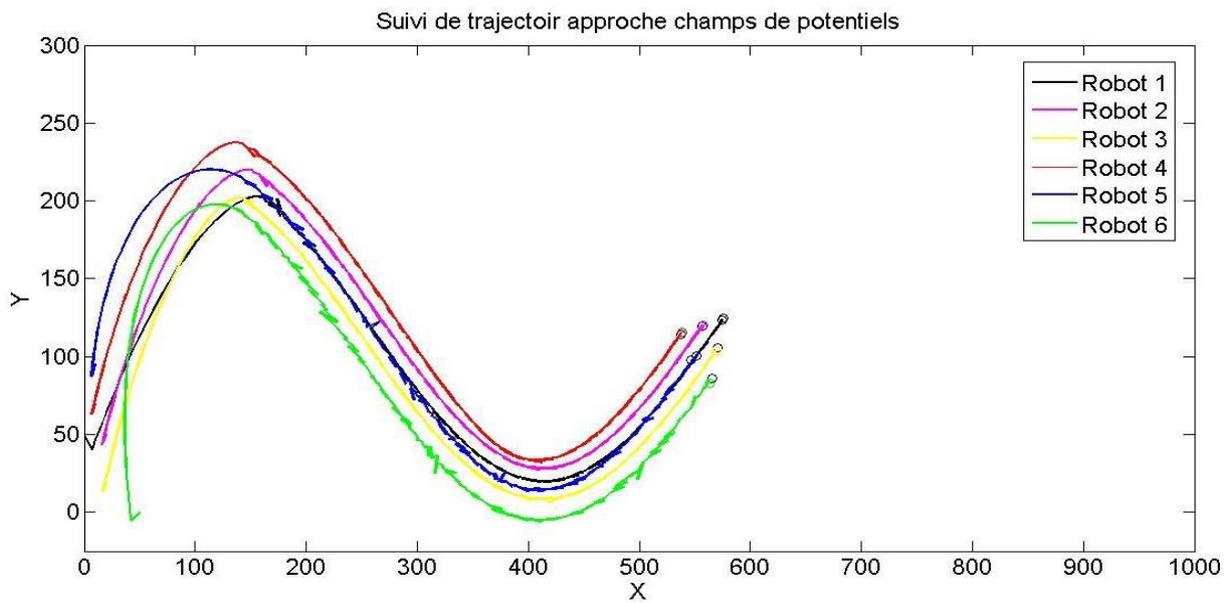


**Figure 3-11 : Erreur sur la distance dans le cas du maintien de la formation**

La **Figure 3-11** montre que l'erreur sur la distance diminue lentement jusqu'à atteindre une valeur nulle avec des petites oscillations autour du zéro pour les robots 1, 2, 3 et 4. Et elle reste oscillatoire pour les robots 5 et 6.

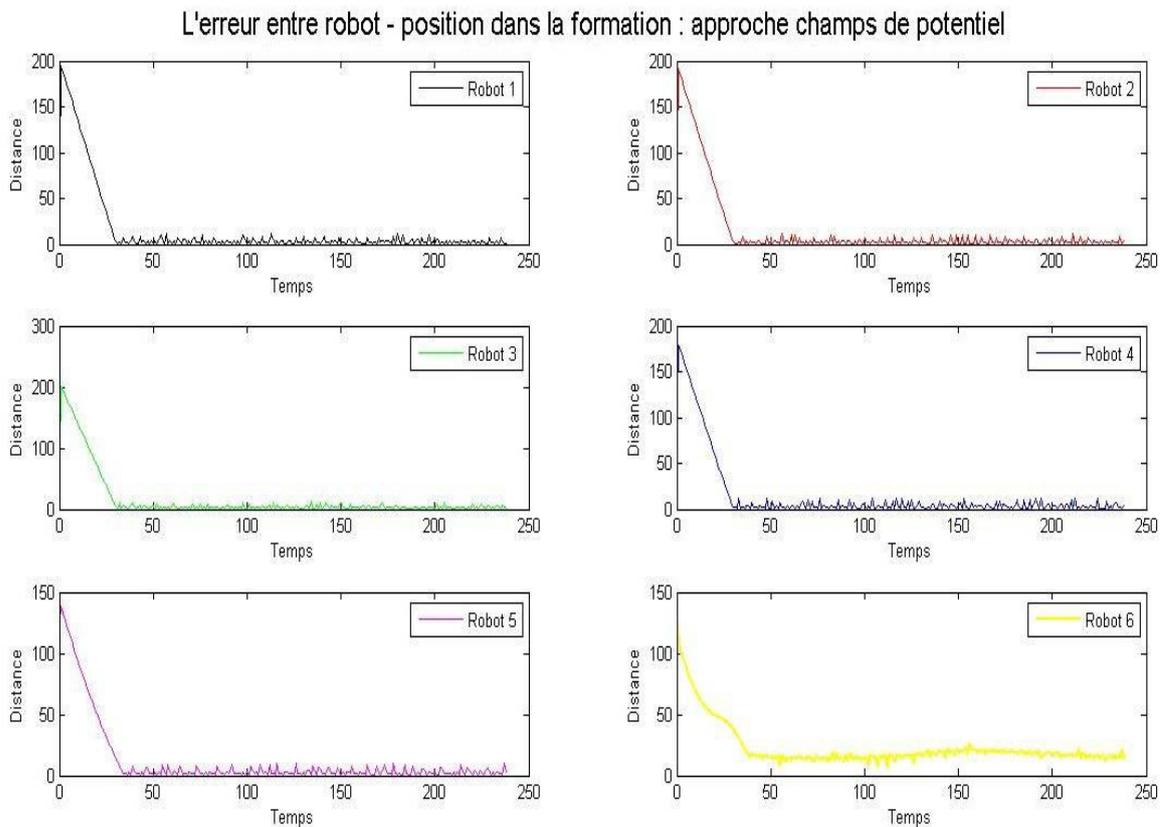
### 3.3.2 SUIVI DE TRAJECTOIRE

Nous avons conçu une formation virtuelle qui a une configuration géométrique triangle, cette formation représente la référence de notre formation de robots mobiles, la **Figure 3-12** montre le résultat de la simulation.



**Figure 3-12 : Suivi de trajectoire dans le cas de l'approche champs de potentiels.**

On a un bon suivi de trajectoire sauf qu'on constate qu'il y a des oscillations dans le mouvement de chaque robot et aussi les robots n'arrivent pas à atteindre leurs positions désirées.

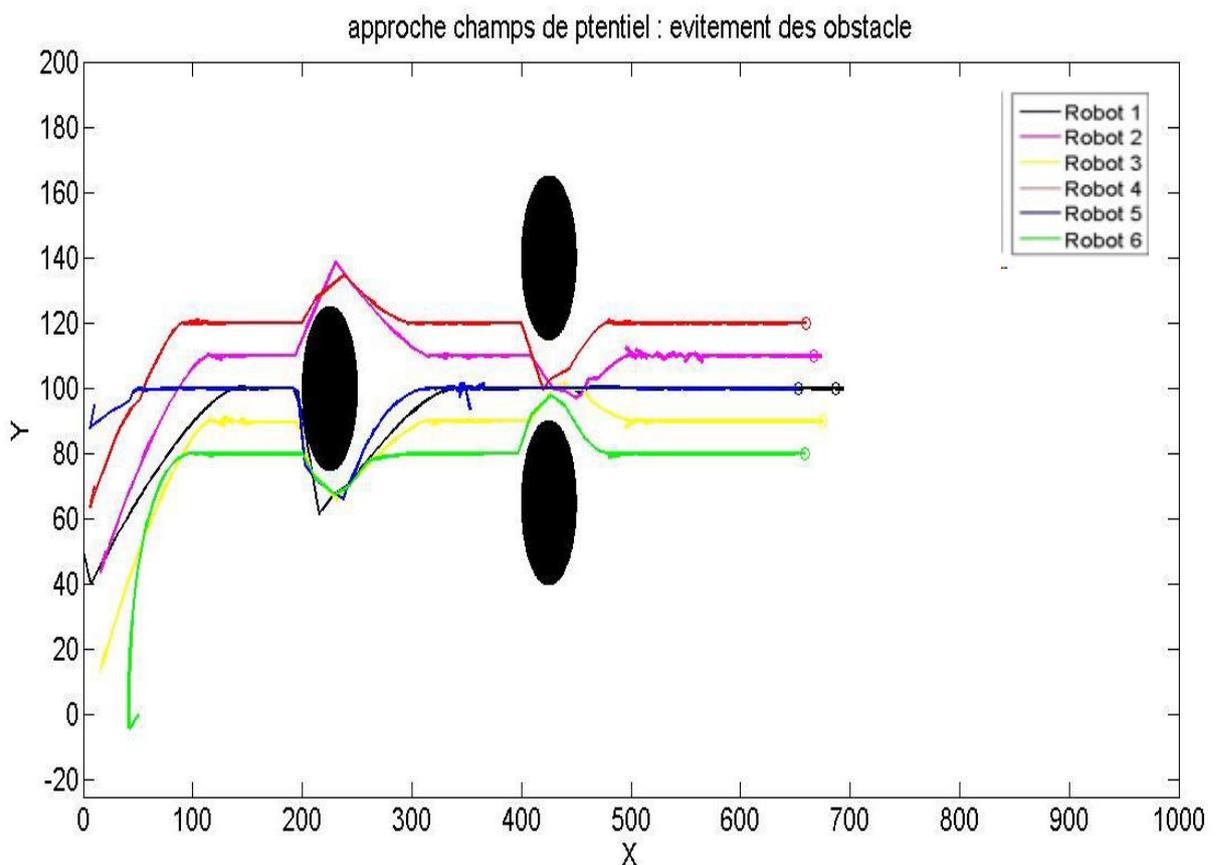


**Figure 3-13 : Erreur sur la distance dans le cas de l'approche champs de potentiel.**

La **Figure 3-13** montre que l'erreur sur la distance entre chaque robot et sa position désirée est une fonction décroissante, mais il existe des pics dans l'erreur lorsque on converge vers le zéro cela confirme que cette approche n'assure pas une bonne convergence vers le but.

### 3.3.3 EVITEMENT DES OBSTACLES

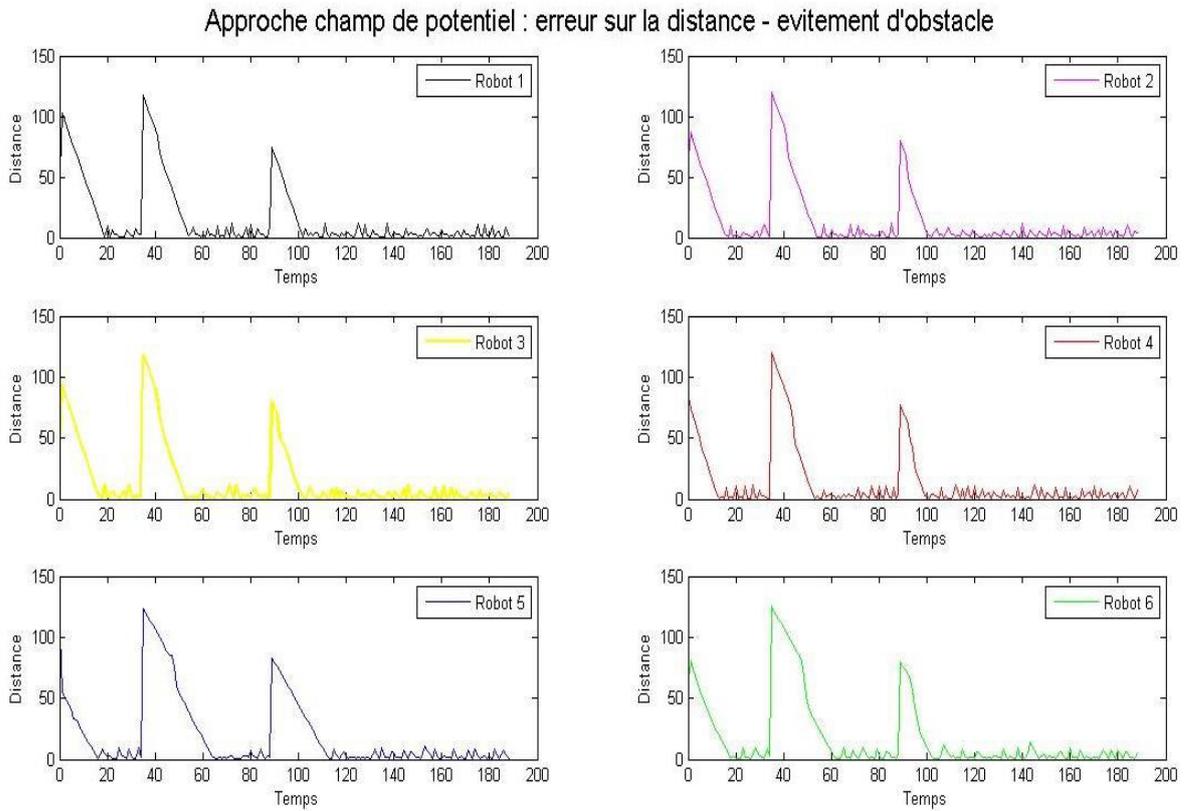
Notre but est d'éviter les obstacles rencontrés par la formation, nous avons dans notre environnement trois obstacles (en noir). La configuration désirée de la formation est un « triangle ». Les robots subissent deux forces en même temps : une force attractive vers la position désirée de chaque robot et une force répulsive générée par les obstacles dans l'environnement, à noter que chaque robot considère les autres robots comme obstacles. Nous montrons les trajectoires des robots durant l'évitement des obstacles dans la **Figure 3-14**.



**Figure 3-14 : Evitement d'obstacle avec l'approche champs de potentiel.**

La formation doit se déformer pour que chaque robot puisse éviter les obstacles présents dans son chemin et dans le passage étroit la forme de la formation à complètement déformer, une fois les obstacles sont évités la formation se régénère à nouveau. On remarque

bien que l'approche des champs de potentiel assure un évitement d'obstacles efficace avec des trajectoires optimales.



**Figure 3-15 : Erreur sur la distance en évitant les obstacles**

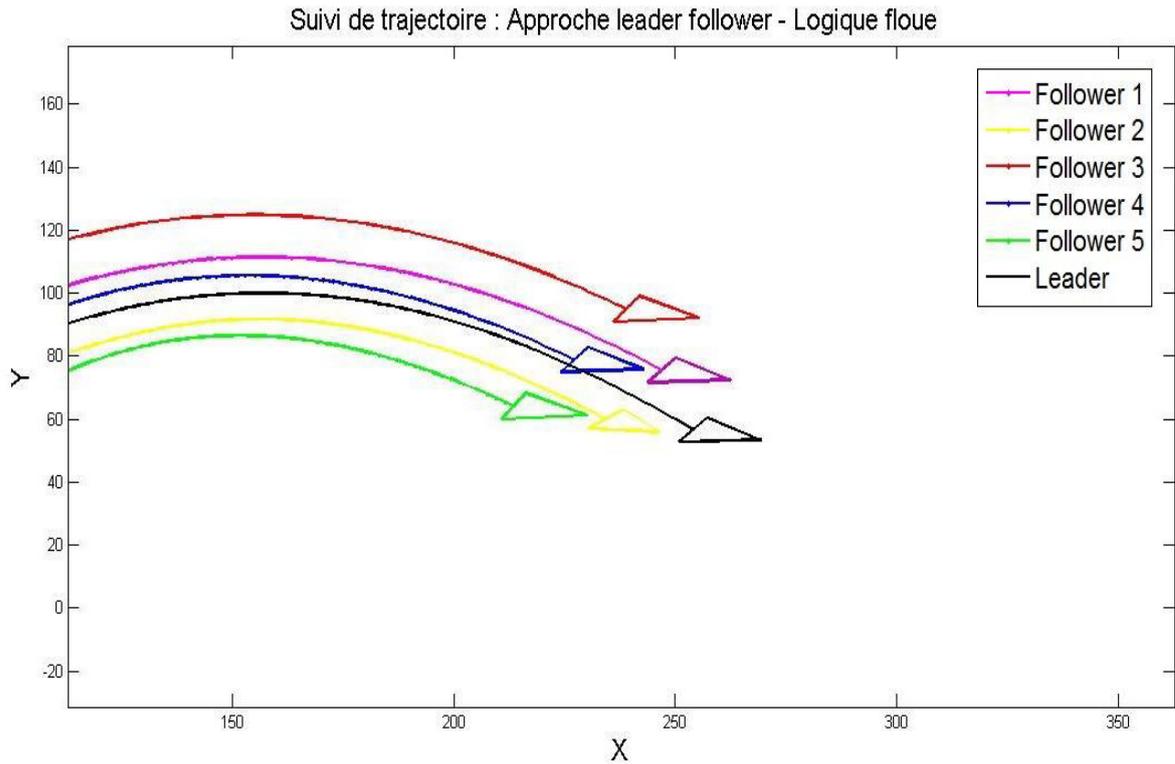
La **Figure 3-15** montre que l'erreur sur la distance diminue jusqu'à atteindre une valeur presque nulle mais oscillatoire pour les six robots. Les grandes variations dans l'erreur sont au moment d'évitement des obstacles.

### 3.4 COMMANDE D'UNE FORMATION EN UTILISANT

#### L'APPROCHE LEADER-FOLLOWER/LOGIQUE FLOUE

Les résultats de la simulation de l'erreur sur la distance entre le leader et les followers dans le cas de suivi de trajectoire ne sont pas acceptables : la valeur de variation de l'erreur est grande à chaque fois que le leader envoie des nouvelles coordonnées pour chaque robot follower, on remarque par exemple pour le follower 1 que l'erreur varie entre 0 et 5 cm cela influence négativement le maintien de la configuration géométrique de la formation.

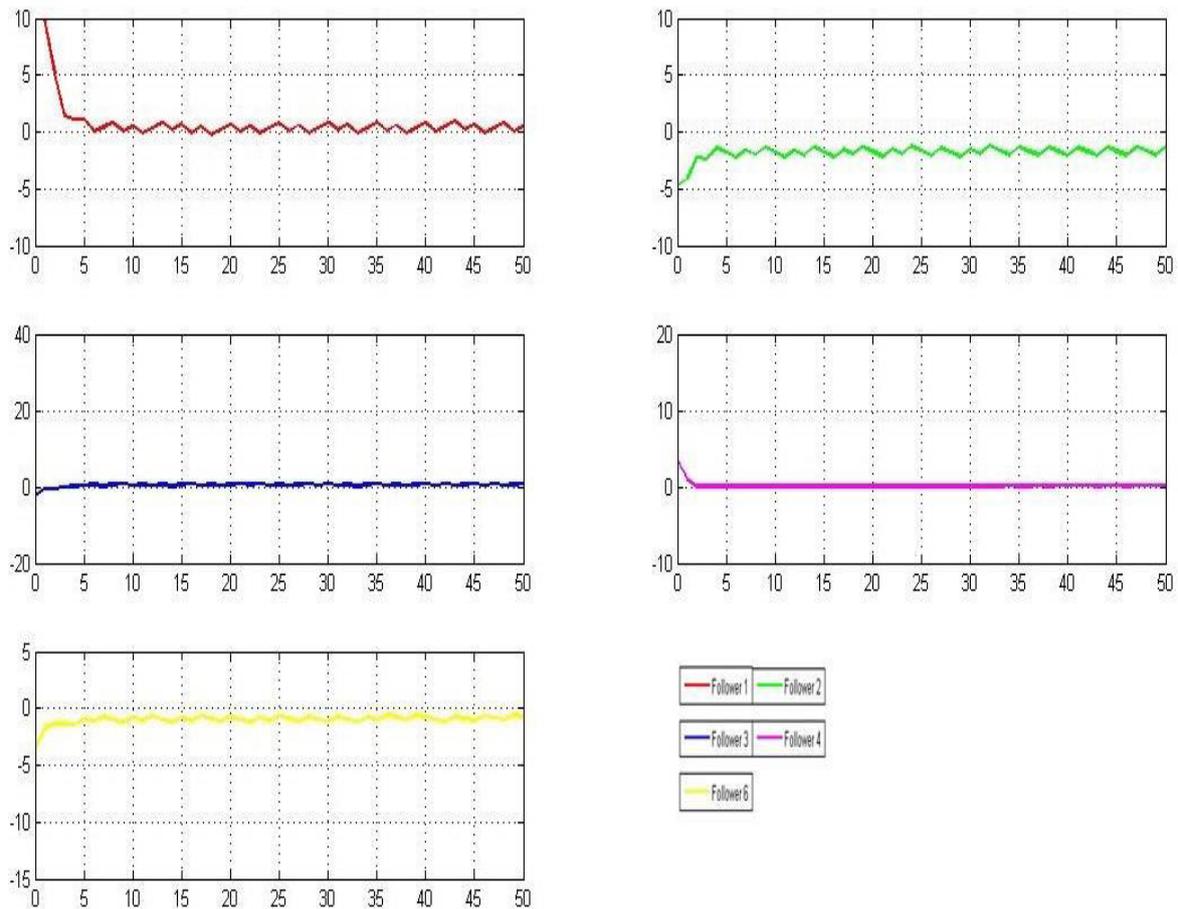
Dans cette partie nous allons présenter les résultats de simulation d'une formation qui suit une trajectoire prédéfinie en utilisant l'approche leader follower améliorée par l'ajout de la logique floue pour contrôler la distance entre le leader-follower  $l$ .



**Figure 3-16 : Suivi de trajectoire : approche leader-follower-logique floue**

D'après la simulation montrée dans la **Figure 3-16**, on remarque que l'approche proposée arrive toujours à bien maintenir une formation de robot mobile sur la trajectoire donnée.

La **Figure 3-17** confirme que l'ajout de la logique floue à l'approche leader-follower augmente ses performances, où on remarque bien que la valeur de l'erreur sur la distance tend vers zéro, et que la plus grande variation de l'erreur est de l'ordre de 1 cm, cela assure un bon maintien de la formation et un excellent suivi de trajectoire.



**Figure 3-17 : Erreur sur la distance : approche leader-follower / logique floue**

### 3.5 ETUDE COMPARATIVE ENTRE L'APPROCHE LEADER-FOLLOWER ET L'APPROCHE CHAMPS DE POTENTIEL

Après avoir simulé les résultats des deux approches : leader-follower et champs de potentiel afin de réaliser les objectifs suivants :

- Maintien d'une formation triangle de six robots mobiles.
- Suivi d'une trajectoire prédéfinie.
- L'évitement des obstacles.

Nous avons décidé de faire une étude comparative (Tableau 2) entre les deux méthodes :

	Approche leader – follower/floue	Approche champs de potentiel
Maintien de la formation	<ul style="list-style-type: none"> <li>• L'erreur sur la distance entre followers et le leader diminue rapidement (temps de convergence court).</li> <li>• Le robot se stabilise à sa position désirée.</li> </ul>	<ul style="list-style-type: none"> <li>• L'erreur sur la distance entre chaque robot et sa position désirée diminue lentement (temps de convergence lent).</li> <li>• Le robot oscille autour de sa position désirée.</li> </ul>
Suivi de trajectoire	<ul style="list-style-type: none"> <li>• Trajectoire lisse.</li> <li>• Petite variation de l'erreur.</li> </ul>	<ul style="list-style-type: none"> <li>• Trajectoire saccadée.</li> <li>• L'existence des pics dans la variation de l'erreur.</li> </ul>
Evitement d'obstacle	<ul style="list-style-type: none"> <li>• Fortes oscillations autour de l'obstacle.</li> <li>• Trajectoire non optimale (distance parcourue).</li> <li>• Temps de régénération de la formation : court.</li> </ul>	<ul style="list-style-type: none"> <li>• Mouvement lisse autour de l'obstacle.</li> <li>• Trajectoire optimale (distance parcourue).</li> <li>• Temps de régénération de la formation : moyen.</li> <li>• Evitement de collision entre robots.</li> </ul>

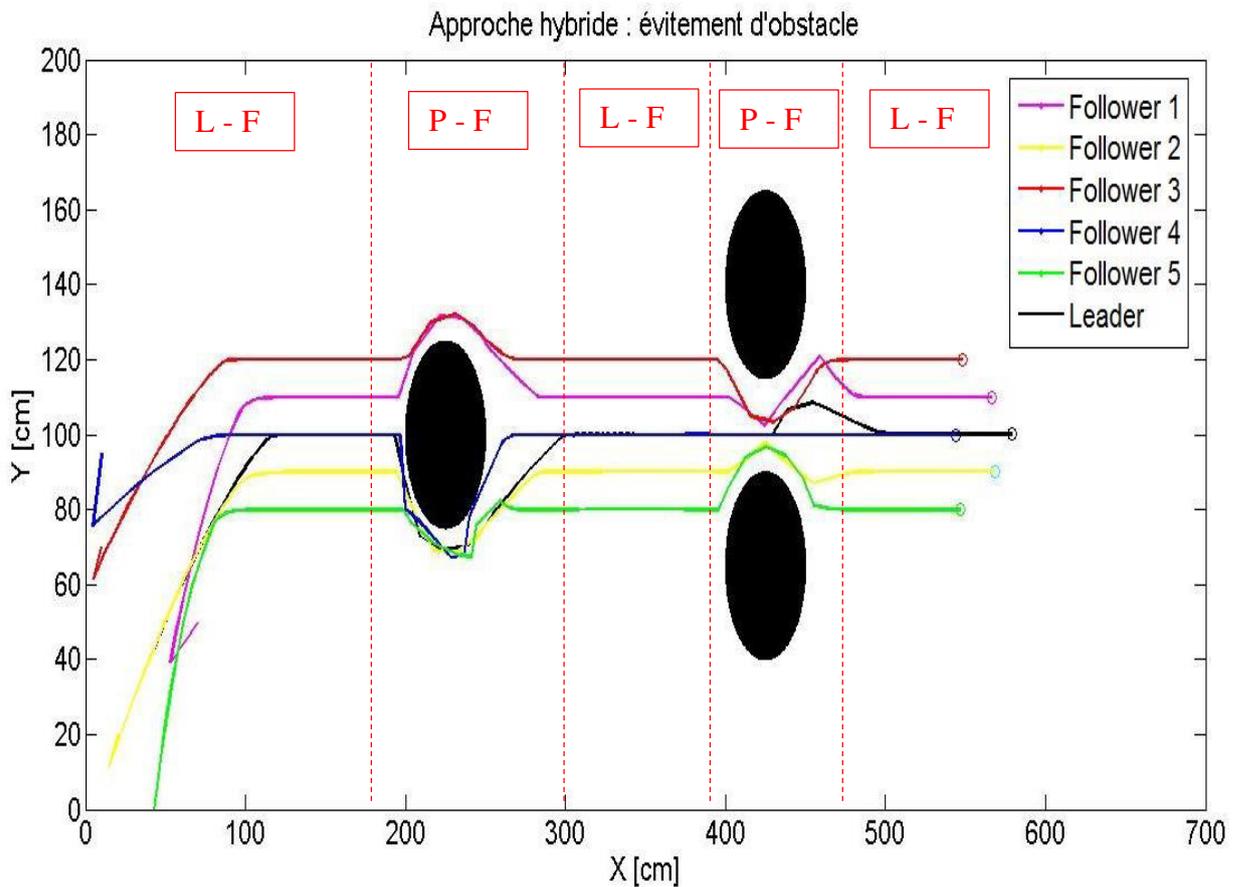
**Tableau 2: Etude comparatif entre les deux approches.**

### 3.6 COMMANDE D'UNE FORMATION EN UTILISANT

#### L'APPROCHE HYBRIDE

Dans cette partie on va simuler les résultats de l'approche hybride où nous allons utiliser la méthode leader-follower pour le maintien de la formation et le suivi de trajectoire et la méthode champs de potentiel pour l'évitement d'obstacle et la collision entre robots.

Le **Figure 3-18** montre les résultats de l'application de l'algorithme hybride pour le suivi d'une trajectoire prédéfinie dans un environnement contenant trois obstacles.



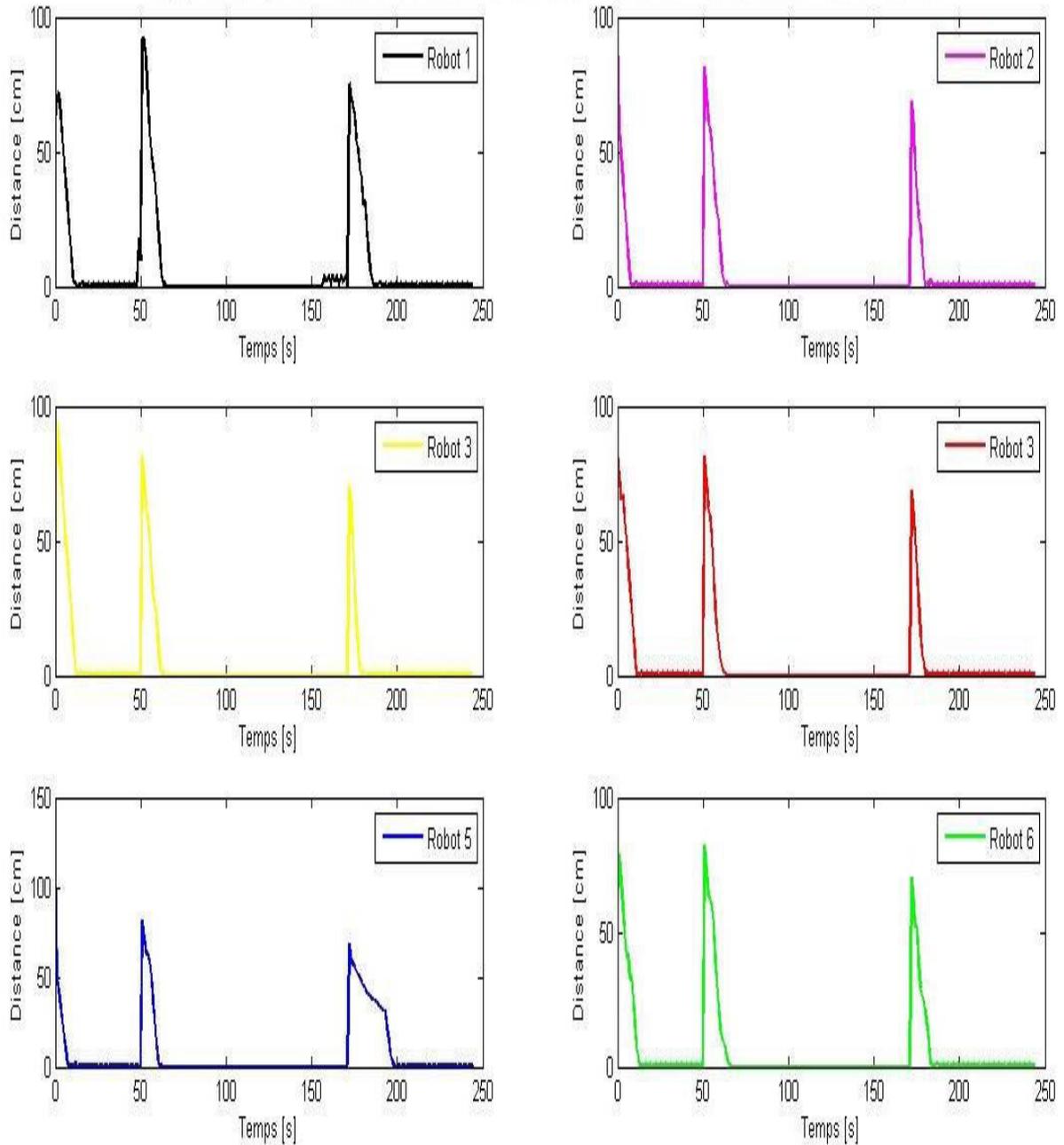
**Figure 3-18 : Approche hybride: suivi de trajectoire et évitement d'obstacle.**

Dans la **Figure 3-18**, nous avons indiqué les zones d'utilisation de chaque méthode :

- L - F : Zone de l'utilisation de l'approche leader – follower.
- P - F : Zone de l'utilisation de l'approche champ de potentiel « Potentiel Field ».

Cette dernière confirme l'efficacité de l'approche hybride.

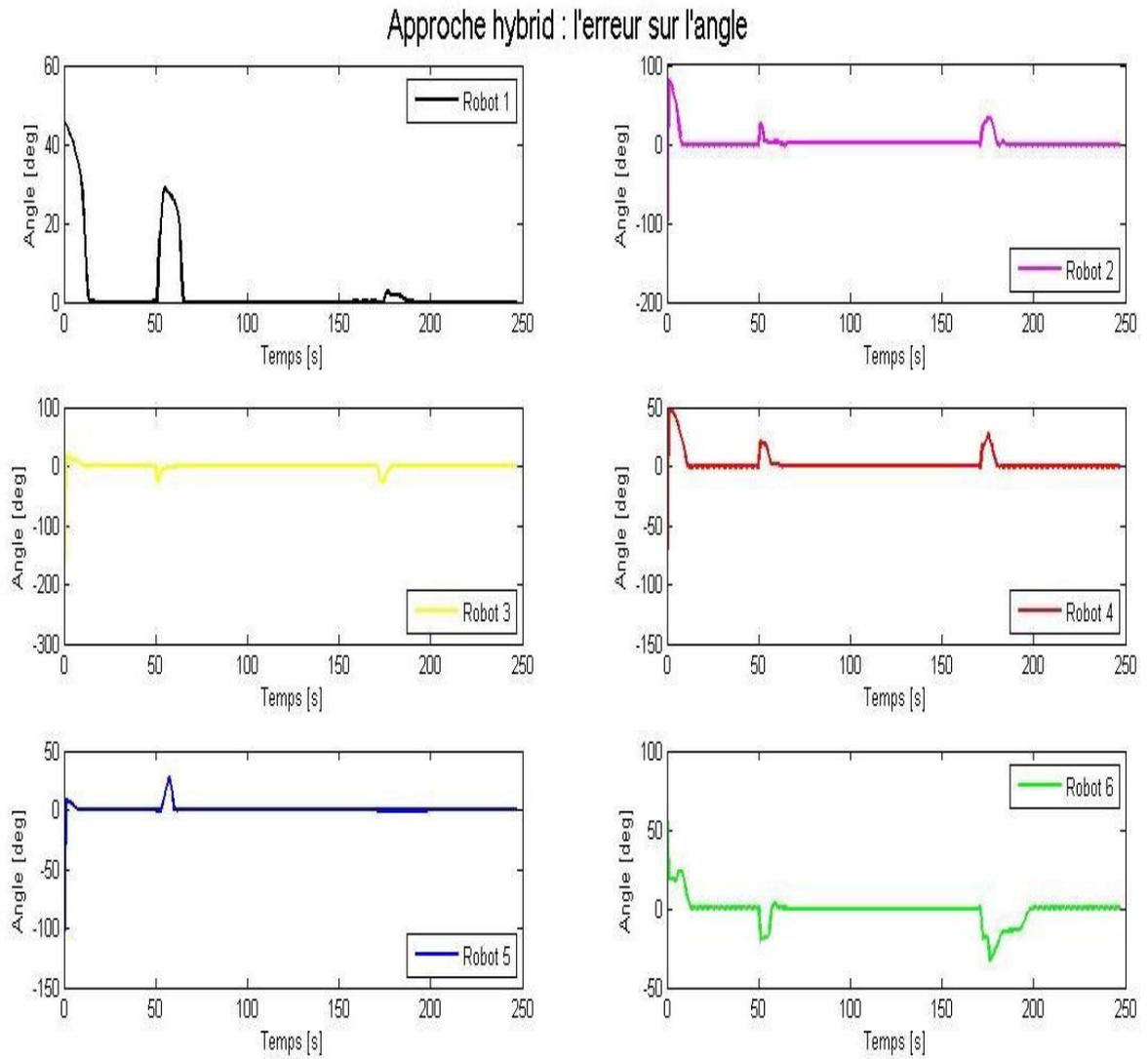
### Approche hybride : erreur sur la distance robot - position désirée dans la formation



**Figure 3-19 : Approche hybride : erreur sur la distance.**

On remarque dans la **Figure 3-19** que les erreurs sur la distance sont négligeables pendant le suivi de trajectoire, les pics représentent l'erreur au moment de l'évitement d'obstacle et on constate que l'erreur converge rapidement vers zéro.

La **Figure 3-20** représente les erreurs sur l'angle entre les robots et le leader virtuel, qui montre l'erreur sur l'angle est nulle sauf dans les moments d'évitement des obstacles.



**Figure 3-20 : Approche hybride : erreur sur l'angle entre robots et leader virtuel.**

### 3.7 CONCLUSION

Dans ce chapitre, nous avons présenté les résultats de simulations des quatre approches de commande d'une formation de robots mobiles, en commençant par les résultats de simulations de la commande coopérative « leader follower », puis celle de l'approche champs de potentiels, ensuite celle des deux approches : l'approche leader follower améliorée par la logique floue et l'approche hybride, afin de réaliser les tâches du maintien de la formation, de suivi de trajectoire et enfin l'évitement des obstacles.

Les résultats de simulation de l'approche leader-follower prouvent que cette dernière réalise le but de la mission des tâches : maintien de la formation et suivi de trajectoire avec des bonnes performances. Celles de l'approche champs de potentiel prouvent que cette dernière réalise le but de la mission avec un accomplissement performant dans l'évitement d'obstacle.

Les résultats de simulation de l'approche leader follower améliorée par la logique floue ont prouvé l'efficacité d'utiliser la logique floue pour le control de la distance entre leader et followers. Pour remédier aux inconvénients des deux approches nous avons utilisé l'approche hybride, cette dernière a donné des résultats qui confirment qu'elle est capable d'assurer toutes les taches : maintien de la formation, suivi de trajectoire et évitement d'obstacles.

Le chapitre suivant sera consacré à la présentation de l'environnement de simulation 3D : V-REP, l'application de la commande coopérative d'une formation de robots mobiles où l'entité utilisée est le robot Pioneer 3-DX.

# **Chapitre 4 :**

## **Simulation sous l'environnement V-REP**

## 4 SIMULATION SOUS L'ENVIRONNEMENT V-REP

### 4.1 INTRODUCTION

V-REP est une Plateforme de simulation utilisée pour le développement des algorithmes rapides, les simulations d'automatisation d'usine, le prototypage et la vérification rapide, l'éducation relative à la robotique, le contrôle à distance, le double contrôle de la sécurité, etc.

Dans ce chapitre on va présenter l'environnement de simulation V-REP : informations générales sur ces concepts et sa terminologie. Ensuite, nous nous concentrons sur les étapes nécessaires pour le connecter avec MATLAB en utilisant les « Remote API » afin de contrôler une formation de robots mobiles. Par la suite, on introduit le robot mobile « Pioneer 3-DX », Et enfin nous présentons les résultats des simulations.

### 4.2 ENVIRONNEMENT V-REP

V-REP (Virtual Robot Experimentation Platform) est un simulateur de robot, développé et entretenu par « Coppelia Robotics », l'environnement de simulation est modélisé à l'aide de l'éditeur de scène inclus. Il contient un grand nombre de modèles prédéfinis pour différents types de robots. [15]

La version actuelle (V-REP PRO EDU, Version 3.4.0. rev. 1) présente de nombreuses fonctionnalités intéressantes, on cite :

- **Contenu portable et multiplateforme :** Il permet la création de contenu portable, évolutif et facile à entretenir ou modifiable : un seul fichier portable peut contenir un modèle ou une scène entièrement fonctionnelle, y compris un code de contrôle.
- **Remote API :** Plus de 100 fonctions V-REP incorporables : contrôlez une simulation ou le simulateur lui-même à distance (par exemple, à partir d'un robot réel ou d'un autre PC). Facile à utiliser, prend en charge le mode d'opération synchrone ou asynchrone, il est optimisé pour un transfert de données important et réduit le délai de communication. Six langages de programmation sont pris en charge : C / C ++, Python, Java, Urbi, Matlab et Octave. Cela rend V-REP très polyvalent et idéal pour les applications multi-robots (**Figure 4-1**).

- **Détection de collision et calcul de distance :** Contrôle rapide des interférences et calcul de la distance minimale.
- **Simulation du capteur de vision.**
- **Simulation du capteur de proximité.**
- **Interfaces utilisateur personnalisées :** Un nombre illimité d'éléments d'interface utilisateur entièrement personnalisables. Style Windows ou style OpenGL.
- **Chemin / planification de mouvement :** Elle est prise en charge de manière très flexible via la bibliothèque OMPL enveloppée dans un plugin V-REP.

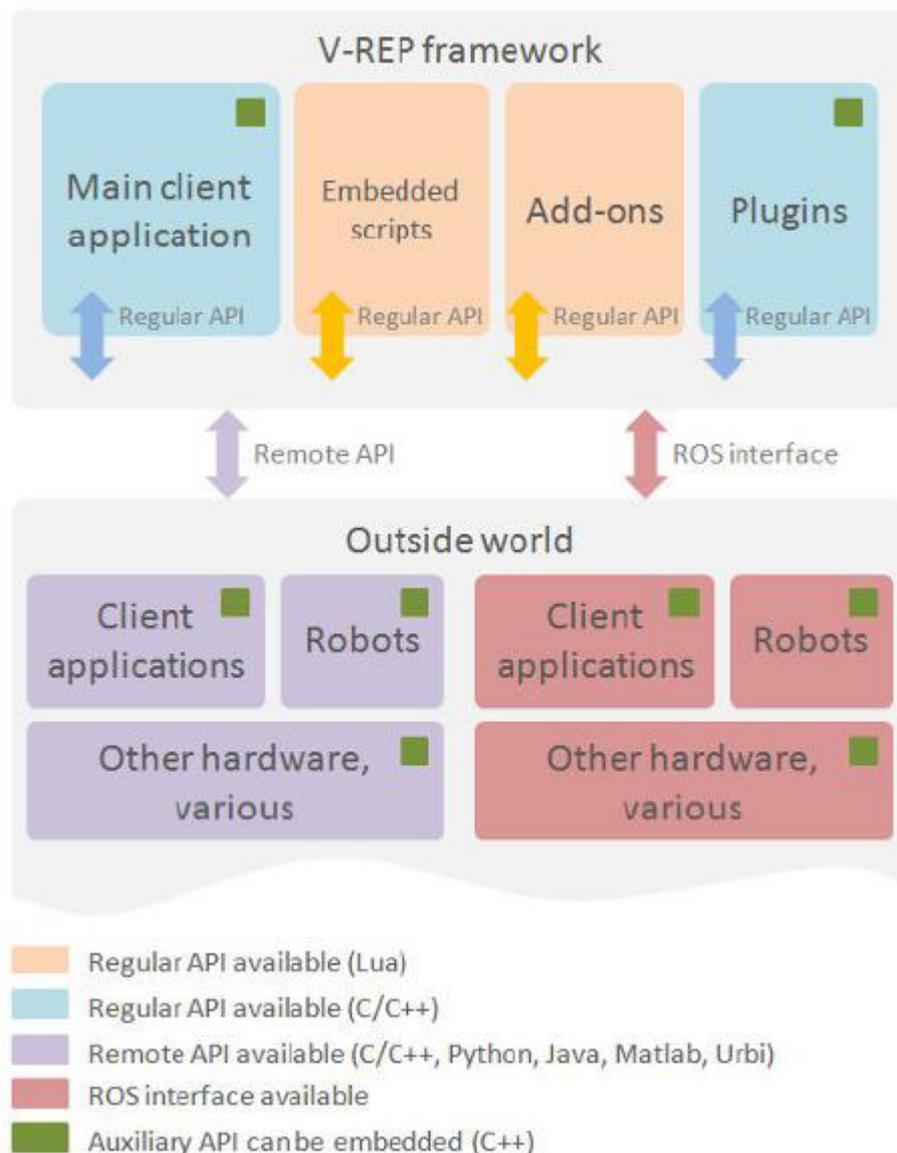
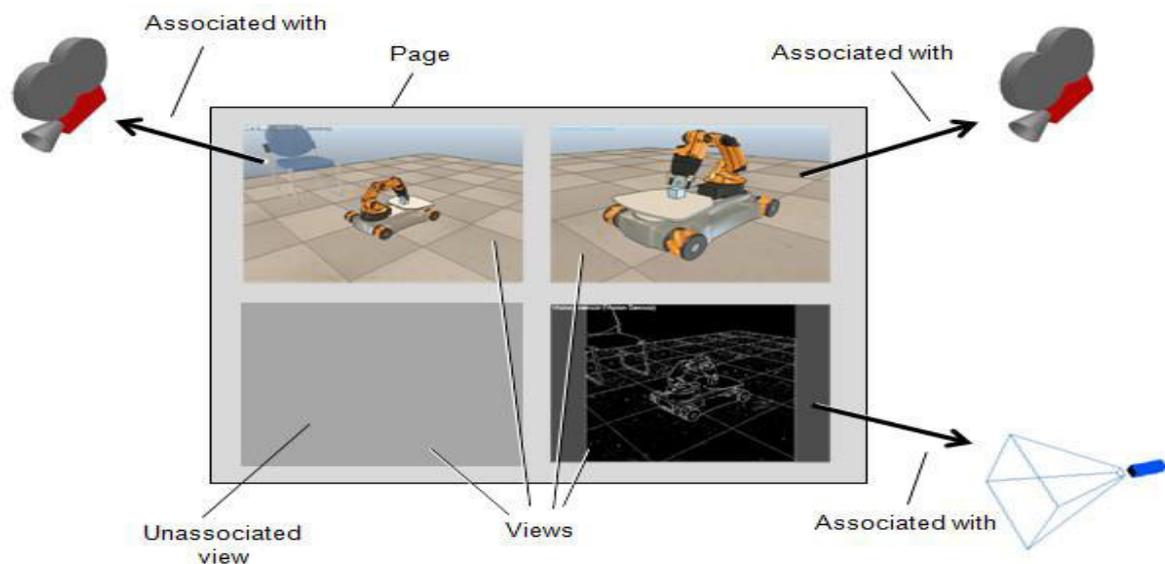


Figure 4-1: Structure de Remote API pour V-REP(en violet)

Les simulations dans V-REP sont organisées en scènes. Une scène représente le monde simulé composé d'un certain nombre d'éléments :

- **Environnement** : L'environnement dans V-REP définit les propriétés et les paramètres qui font partie d'une scène, mais ce ne sont pas des objets de scène. Il contient les éléments suivants : la couleur du fond, paramètres de brouillard, lumière ambiante, informations sur la création de scène et paramètres additionnels.
- **Le script principal** : Le script principal est un script de simulation. Par défaut, chaque scène de V-REP aura un script principal. Il contient le code de base qui permet d'exécuter une simulation. Sans script principal, une simulation en cours d'exécution ne fera rien. Le script principal est appelé à chaque étape de simulation exactement une fois. En plus de cela, on l'appelle une fois lors de l'initialisation (au début d'une simulation), et une fois lors de la restauration (à la fin d'une simulation).
- **Pages et vues** : Une page dans V-REP est la principale surface d'affichage d'une scène. Ce n'est pas directement une vue, mais peut contenir une, deux ou autant de vues que nécessaire. Une vue est ce qui permet d'afficher le contenu de l'image d'un objet spécifique qui doit être un objet visible. Si une vue est associée à un objet de caméra par exemple, elle peut afficher ce que la caméra voit. La **Figure 4-2** explique la relation d'objet page, vue et visible.



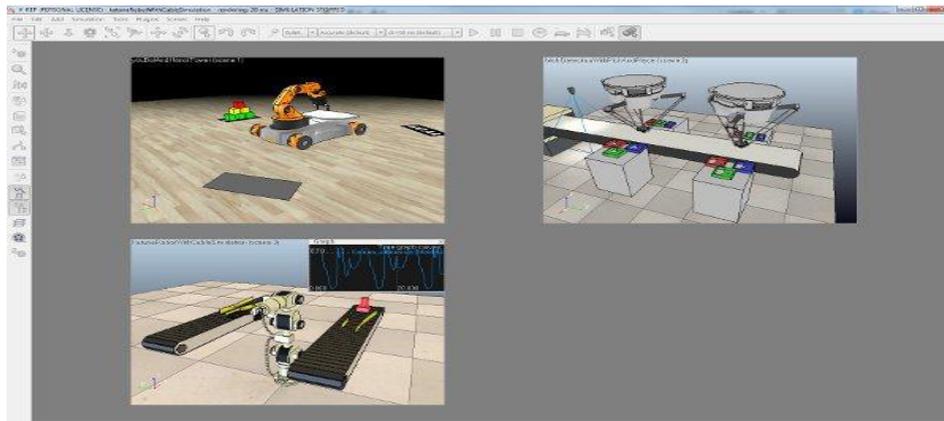
**Figure 4-2 : Relation page, vue et objet visible.**

Une scène ou un contenu d'image de scène peut être vu à travers un objet visible associé à une vue, elle-même contenue dans une page. Lors de la création d'une nouvelle scène ([**Menu bar --> File --> New Scene**]), la scène par défaut contiendra les éléments suivants :

- **Plusieurs objets caméra** : les caméras permettent de voir la scène s'ils sont associés à une vue.
- **Plusieurs objets de lumière** : sans lumière, la scène serait à peine visible. La lumière sert à éclairer la scène.
- **Plusieurs vues** : une vue est associée à une caméra et affiche ce que la caméra voit.
- **Plusieurs pages** : une page contient une ou plusieurs vues.
- **L'environnement.**
- **Le plancher** : le sol est constitué par les objets groupés dans un modèle.
- **Le script principal par défaut.**

Les scènes peuvent être ouvertes (chargées) avec [**Menu bar --> File --> Open Scene...**] et sauvegardées avec [**Menu bar --> File --> Save Scene**] ou [**Menu Bar --> File --> Save Scene as...**]. Les fichiers de scène (« \* .ttt – files ») supportent également les opérations de glisser-déposer entre la fenêtre de l'explorateur et la fenêtre de l'application. Les fichiers de scène peuvent également être double-cliqué, auquel cas ils lanceront l'application V-REP et seront ouverts.

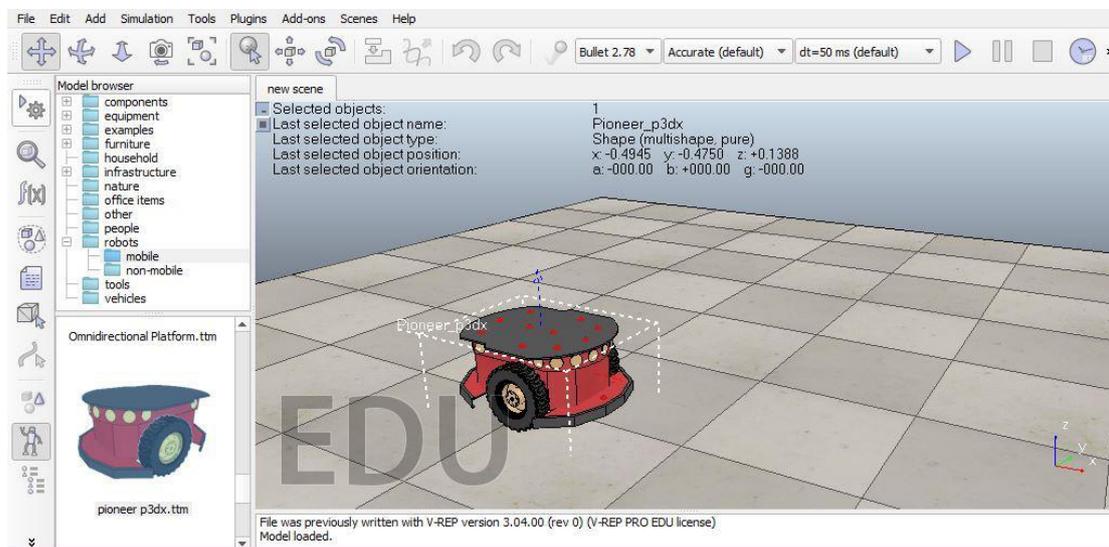
La commutation entre les scènes ouvertes peut être obtenue en un seul clic dans la partie supérieure de la hiérarchie de la scène (toutes les scènes ouvertes sont regroupées en haut de la hiérarchie des scènes) ou en utilisant le sélecteur de scène via le bouton de la barre d'outils concernée. (**Figure 4-3**).



**Figure 4-3: Sélecteur de scène**

Aussi V-REP contient des modèles prédéfinies, un modèle est un sous-élément d'une scène. Un modèle en soi ne peut pas exister, sauf dans un fichier ("\* .ttm" - type de fichier). Un modèle doit être contenu dans une scène pour être opérationnel.

Les modèles sont définis par une sélection d'objets de scène construites sur un même arbre hiérarchique, où la base de l'arbre doit être un objet marqué comme objet est une base de modèle. Ils peuvent être chargés avec [Menu bar --> File --> Load model...]. Il est cependant beaucoup plus facile et pratique de charger un modèle avec une opération de glisser-déplacer entre le navigateur modèle et une vue de scène. (Figure 4-4).



**Figure 4-4: Ajout d'un modèle du navigateur de modèle.**

## 4.3 INTERFACE MATLAB/V-REP

### 4.3.1 FONCTIONS UTILISEES

Nous allons expliquer les fonctions de la bibliothèque « REMAPI » que nous avons utilisé [16] .

#### Simxstart :

<b>La description</b>	Démarre un thread de communication avec le serveur (c'est-à-dire V-REP). Un même client peut démarrer plusieurs threads de communication (mais un seul thread de communication pour une adresse IP et un port).
<b>Prototype</b>	<b>[clientID] = vrep.simxStart (connectionAddress, connectionPort, waitUntilConnected, doNotReconnectOnceDisconnected, timeOutInMs, commThreadCycleInMs).</b>
<b>Paramètres</b>	<p><b>ConnectionAddress</b> : l'adresse IP où se trouve le serveur (c'est-à-dire V-REP).</p> <p><b>ConnectionPort</b> : le numéro de port où se connecter.</p> <p><b>WaitUntilConnected</b> : si <b>vrai</b>, alors la fonction se bloque jusqu'à ce qu'elle soit connectée (ou expiré).</p> <p><b>DoNotReconnectOnceDisconnected</b> : s'il est <b>vrai</b>, le thread de communication ne tentera pas une seconde connexion si une connexion a été perdue.</p> <p><b>TimeOutInMs</b> :</p> <p style="padding-left: 20px;"><b>Si positif</b> : le délai de connexion en millisecondes pour la première tentative de connexion. Dans ce cas, le délai d'attente pour bloquer les appels de fonction est de 5000 millisecondes.</p> <p style="padding-left: 20px;"><b>Si négatif</b> : sa valeur positive est le délai d'attente pour bloquer les appels de fonction. Dans ce cas, le délai de connexion pour la première tentative de connexion est de 5000 millisecondes.</p> <p><b>CommThreadCycleInMs</b> : indique à quelle fréquence les paquets de données sont envoyés en va-et-vient. La réduction de ce nombre améliore la réactivité et une valeur par défaut de 5 est recommandée.</p>
<b>Valeurs de retour</b>	<b>ClientID</b> : l' <b>ID du client</b> , ou <b>-1</b> si la connexion au serveur n'était pas possible (c'est-à-dire un délai d'attente a été atteint). Un appel à

	<b>simxStart</b> doit toujours être suivi à la fin avec un appel à <b>simxFinish</b> si <b>simxStart</b> n'a pas retourné <b>-1</b> .
--	---

**Simxfinish :**

<b>La description</b>	Termine le fil de communication. Elle devrait être la dernière fonction <b>remoteAPI</b> appelée côté client. <b>SimxFinish</b> ne doit être appelé qu'après un appel réussie à <b>simxStart</b> . Il s'agit d'une fonction d'aide de <b>remoteAPI</b> .
<b>Prototype</b>	<b>vrep.simxFinish</b> (numéro ou clientID)
<b>Paramètres</b>	<b>ClientID</b> : l'ID du client. Se référer à <b>simxStart</b> . Peut-être <b>-1</b> pour mettre fin à tous les fils de communication en cours d'exécution.

**SimxGetObjectHandle :**

<b>La description</b>	Récupère un Handler d'objet en fonction de son nom.
<b>Prototype</b>	[returnCode, handle] = <b>vrep.simxGetObjectHandle</b> ( <b>clientID</b> , <b>objectName</b> , <b>operationMode</b> )
<b>Paramètres</b>	<b>ClientID</b> : l'ID du client. Se référer à <b>simxStart</b> . <b>ObjectName</b> : nom de l'objet. Si possible spécifiez toujours le nom complet de l'objet, y compris le #. <b>OperationMode</b> : mode de fonctionnement de la fonction <b>remoteAPI</b> . Le mode de fonctionnement recommandé pour cette fonction est <b>simx_opmode_blocking</b>

**SimxGetObjectOrientation :**

<b>La description</b>	Récupère l'orientation (angles d'Euler) d'un objet.
<b>Prototype</b>	[returnCode, eulerAngles] = <b>vrep.simxGetObjectOrientation</b> ( <b>clientID</b> , <b>objectHandle</b> , <b>relativeToObjectHandle</b> , <b>operationMode</b> )
<b>Paramètres</b>	<b>ClientID</b> : l'ID du client. Se référer à <b>simxStart</b> . <b>ObjectHandle</b> : handle de l'objet. <b>RelativeToObjectHandle</b> : indique par rapport à quel cadre de référence nous voulons l'orientation. Spécifiez <b>-1</b> pour récupérer

	<p>l'orientation absolue, <b>sim_handle_parent</b> pour récupérer l'orientation par rapport au parent de l'objet, ou un <b>ObjectHandle</b> relatif à l'image de référence dont vous souhaitez.</p> <p><b>orientationOperationMode</b> : mode de fonctionnement de la fonction <b>remoteAPI</b>. Les modes de fonctionnement recommandés sont <b>simx_opmode_streaming</b> (le premier appel) et <b>simx_opmode_buffer</b> (les appels suivants).</p>
<b>Valeurs de retour</b>	<p><b>ReturnCode</b> : un code de retour de la fonction <b>remoteAPI</b>.</p> <p><b>EulerAngles</b> : 3 valeurs représentant les angles d'Euler (<b>alpha</b>, <b>bêta</b> et <b>gamma</b>).</p>

#### SimxGetObjectPosition :

<b>La description</b>	Récupère la position d'un objet.
<b>Prototype</b>	<b>[ReturnCode, position du tableau] = vrep.simxGetObjectPosition (clientID, objectHandle, relativeToObjectHandle, operationMode).</b>
<b>Paramètres</b>	<p><b>ClientID</b> : l'ID du client. Se référer à <b>simxStart</b>.</p> <p><b>ObjectHandle</b> : handle de l'objet</p> <p><b>RelativeToObjectHandle</b> : indique par rapport à quel cadre de référence nous voulons la position. Spécifiez <b>-1</b> pour récupérer la position absolue, <b>sim_handle_parent</b> pour récupérer la position par rapport au parent de l'objet, ou un <b>ObjectHandle</b> relatif à la référence dont vous souhaitez la position.</p> <p><b>OperationMode</b> : mode de fonctionnement de la fonction <b>remoteAPI</b>. Les modes de fonctionnement recommandés pour cette fonction sont <b>simx_opmode_streaming</b> (le premier appel) et <b>simx_opmode_buffer</b> (les appels suivants).</p>
<b>Valeurs de retour</b>	<p><b>ReturnCode</b> : un code de retour de la fonction <b>remoteAPI</b>.</p> <p><b>Position</b> : 3 valeurs représentant la position (<b>x, y, z</b>).</p>

#### SimxSetJointTargetVelocity :

<b>La description</b>	Définit la vitesse intrinsèque cible d'une articulation non sphérique.
-----------------------	--

<b>Prototype</b>	<b>[returnCode] = vrep.simxSetJointTargetVelocity (clientID, jointHandle, targetVelocity, operationMode)</b>
<b>Paramètres</b>	<p><b>ClientID</b> : l'ID du client. Se référer à <b>simxStart</b>.</p> <p><b>JointHandle</b> : handle de l'articulation</p> <p><b>TargetVelocity</b> : vitesse cible de l'articulation (vitesse linéaire ou angulaire selon le type de joint).</p> <p><b>OperationMode</b> : mode de fonctionnement de la fonction <b>remoteAPI</b>. Les modes de fonctionnement recommandés pour cette fonction sont <b>simx_opmode_oneshot</b> ou <b>simx_opmode_streaming</b>.</p>
<b>Valeurs de retour</b>	<b>ReturnCode</b> : un code de retour de la fonction <b>remoteAPI</b> .

**Delete :**

<b>La description</b>	<b>Destructeur</b> : décharger la bibliothèque.
-----------------------	---

#### 4.4 ROBOT MOBILE « PIONEER P3DX »

Les robots de recherche Pioneer sont les robots mobiles intelligents les plus populaires au monde pour l'éducation et la recherche. La polyvalence, la fiabilité et la durabilité les rend la plate-forme préférée pour la robotique intelligente avancée.

Pioneer 3-DX montrée dans la **Figure 4-5** est un petit robot d'entraînement différentiel à deux roues, léger, actionné et idéal pour l'utilisation à l'intérieur d'un laboratoire ou d'une salle d'étude. [17]



**Figure 4-5 : Robot mobil Pioneer 3-dx**

<b>Caractéristiques</b>	
<b>Corps</b>	1,6 mm en aluminium (revêtu de poudre)
<b>Pneus</b>	caoutchouc rempli de mousse
<b>Opération</b>	
<b>Poids du robot</b>	9 Kg
<b>Charge utile</b>	17 Kg
<b>Mouvement différentiel</b>	
<b>Rayon de rotation</b>	0 cm
<b>Rayon oscillant</b>	26.7 cm
<b>Max. Vitesse avant / arrière</b>	1.2 m/s
<b>Vitesse de rotation</b>	300 deg/s
<b>Max. Traversable Step</b>	2.5 cm
<b>Max. Traversable Gap</b>	5 cm
<b>Max. Niveau traversable</b>	25 %
<b>Terrain traversable</b>	intérieur, accessible en fauteuil roulant.

Tableau 3: Caractéristiques du robot P3-DX

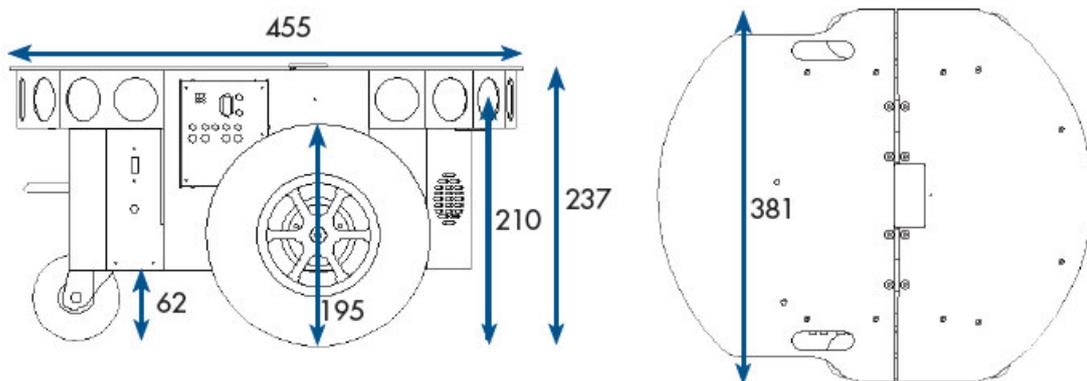
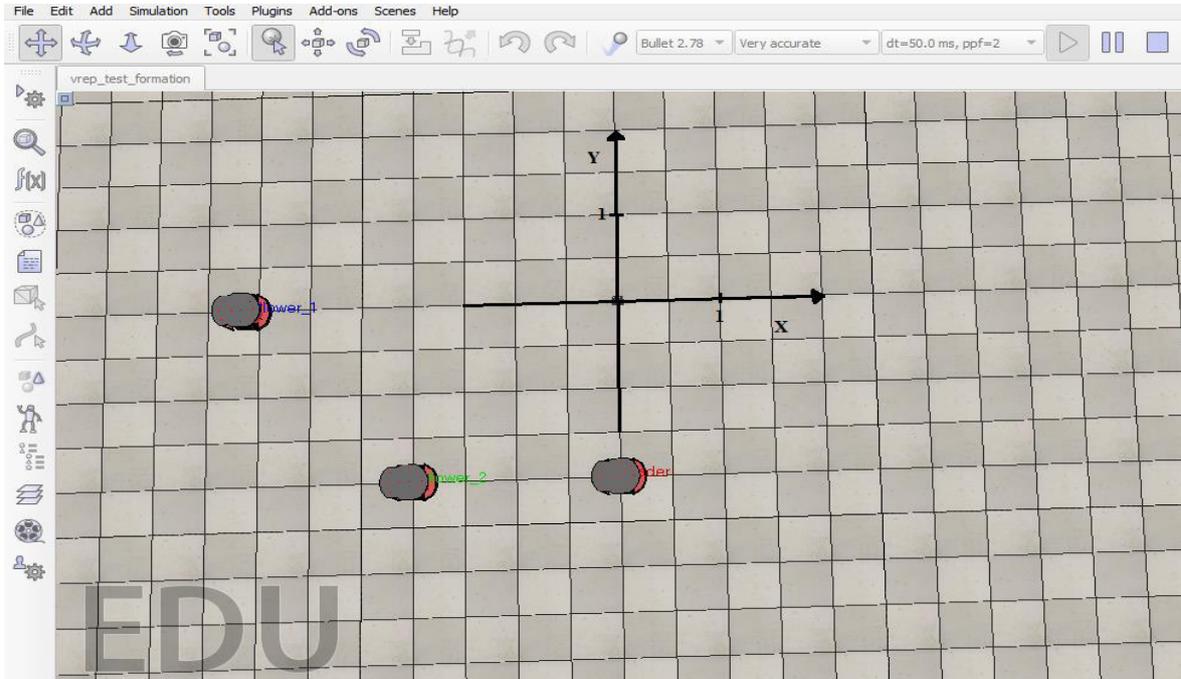


Figure 4-6: Dimension du Pioneer 3-dx.

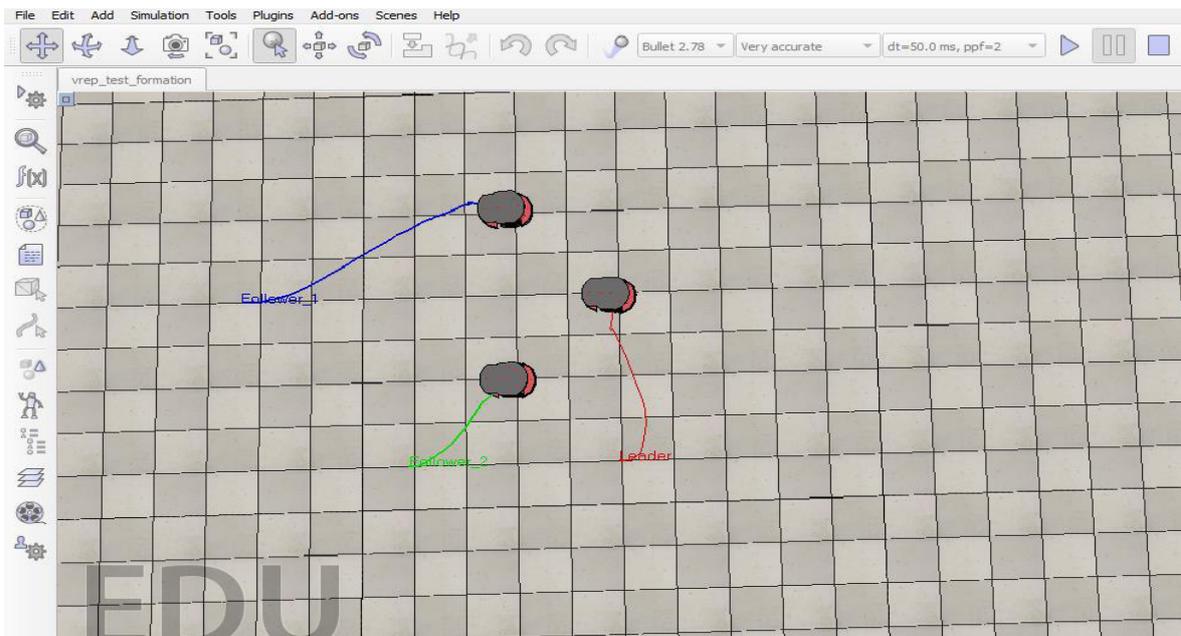
## 4.5 SIMULATION

Dans cette partie nous allons présenter les résultats de simulations de l'algorithme hybride appliqué sur un groupe de trois robots mobiles de type Pioneer 3-DX.

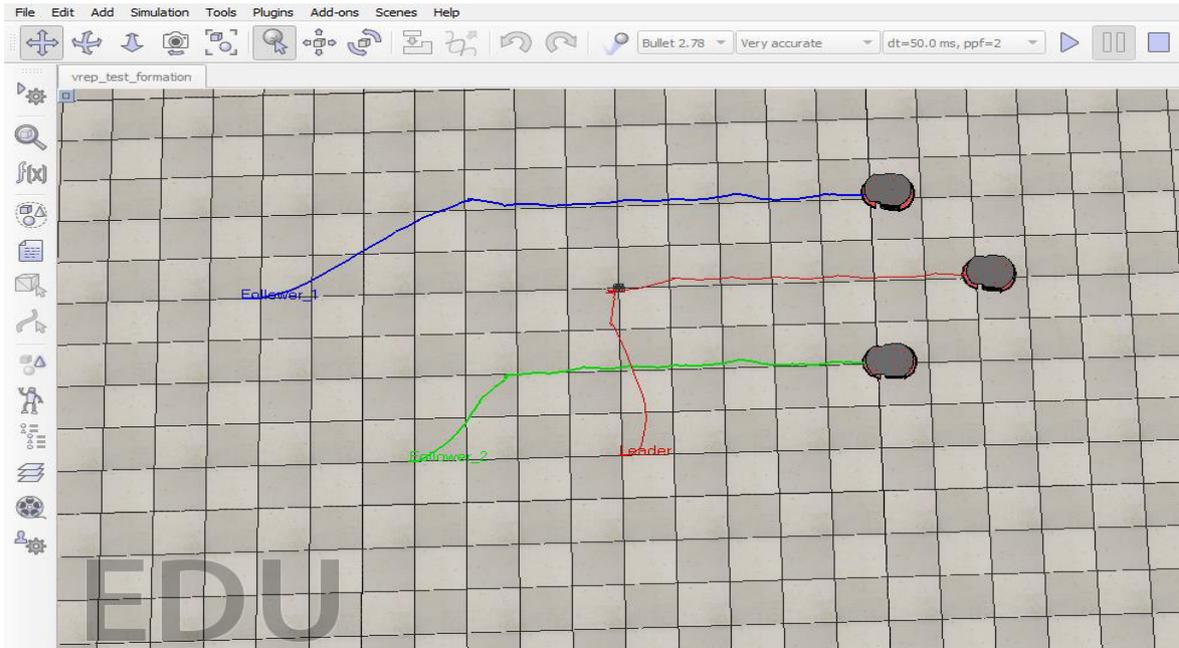
Les **Figure 4-7**, **Figure 4-8** et **Figure 4-9** représentent des scènes de l'évolution de la formation dans l'environnement V-REP :



**Figure 4-7: Etat initiale.**

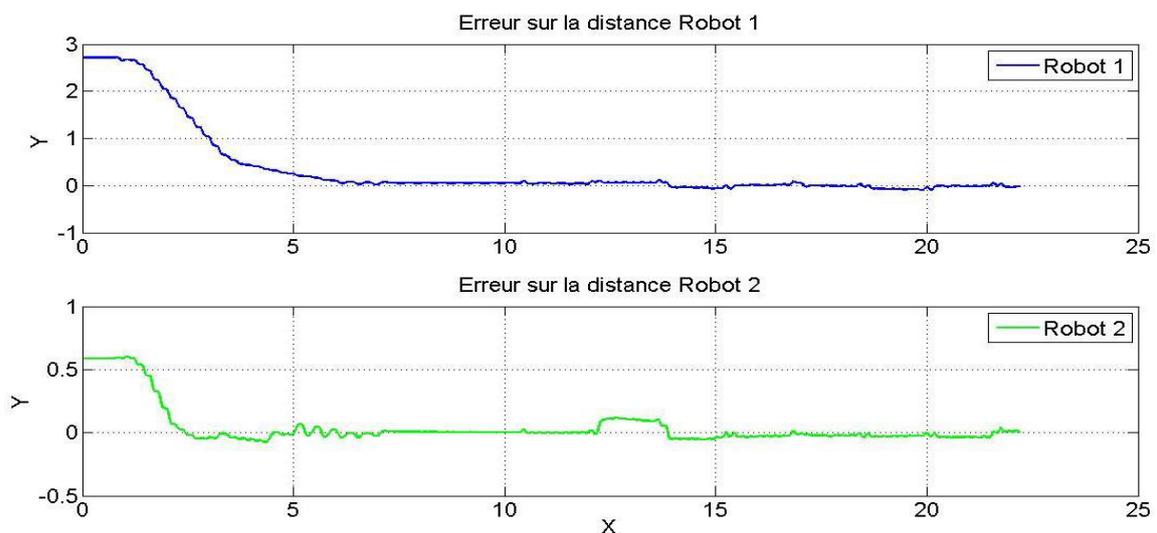


**Figure 4-8: Maintien de la formation.**



**Figure 4-9: Suivi de trajectoire.**

La **Figure 4-7** représente l'état initial de chaque robot dans le repère absolu, la **Figure 4-8** montre que les robots arrivent à assurer la forme géométrique désirée « triangle » et puis ils suivent une trajectoire rectiligne en gardant la formation comme c'est montré dans **Figure 4-9**.

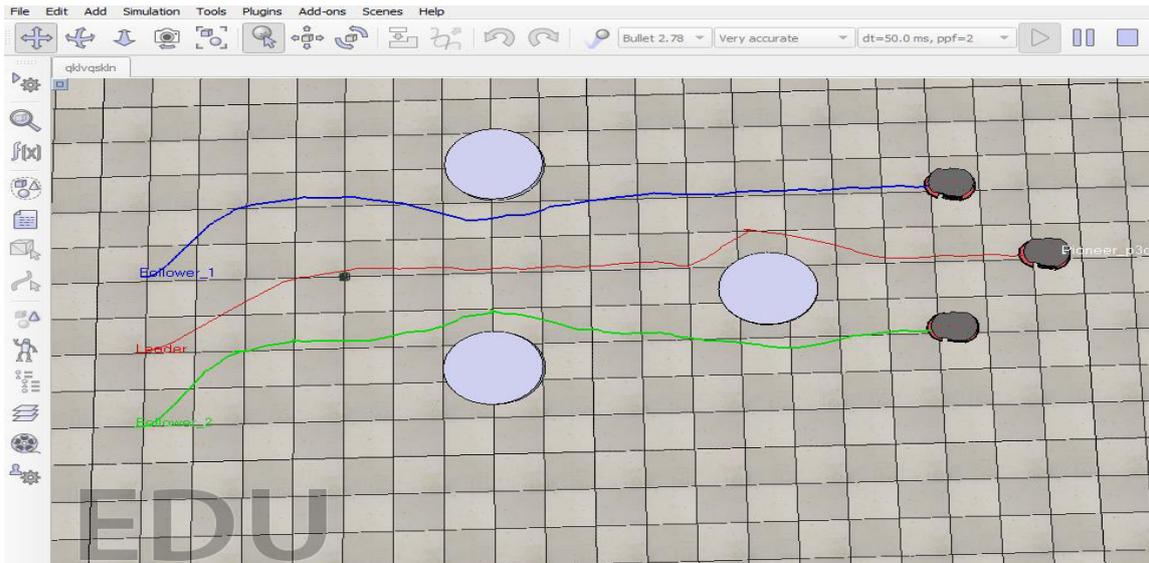


**Figure 4-10: Erreur sur la distance.**

La **Figure 4-10** montre la variation de l'erreur sur la distance entre chaque follower et sa position désirée à noter que la tolérance de l'erreur est de 0.1cm, nous remarquons que

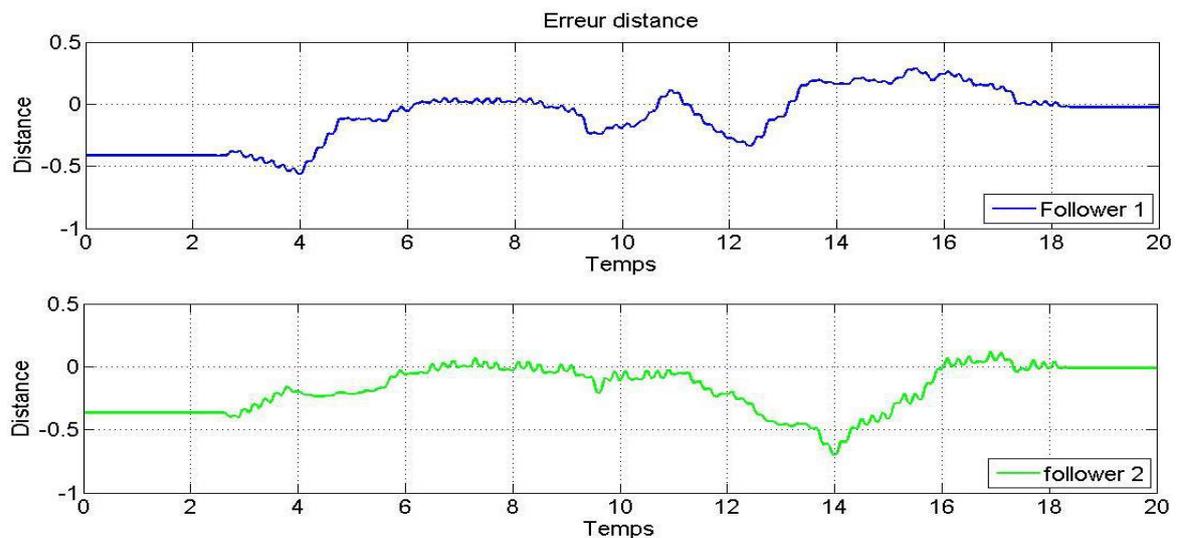
cette variation converge vers le zéro dans 6s qui est le temps nécessaire de la création de la formation et se stabilise dans cette valeur avec de petites oscillations, après les robots commencent à suivre la trajectoire rectiligne et nous remarquons que l'erreur oscille autour de zéro.

La **Figure 4-11** représente la scène de l'évitement des obstacles de la formation dans l'environnement V-REP :



**Figure 4-11: Evitement des obstacles.**

La **Figure 4-11** montre les trajectoires des robots Pioneer dans un environnement contenant des obstacles, nous remarquons que la formation se régénère de nouveau après avoir évité les obstacles.



**Figure 4-12: Erreur sur la distance.**

**Figure 4-12** montre la variation de l'erreur sur la distance entre chaque follower et sa position désirée, à noter que la tolérance de l'erreur est de 0.1cm, nous remarquons que cette variation converge vers le zéro après 6s qui est le temps nécessaire de la création de la formation, et elle se stabilise dans cette valeur avec de petites oscillations. Après les robots commencent à suivre la trajectoire rectiligne, nous remarquons que l'erreur oscille autour de zéro, les grandes variations correspondent aux moments d'évitement des obstacles.

## 4.6 CONCLUSION

Dans ce chapitre nous avons présenté l'environnement de simulation V-REP, les étapes nécessaires pour connecter MATLAB et V-REP en utilisant les « Remote API ». Par la suite nous avons montré les résultats de simulation de l'approche hybride pour la commande d'une formation de robot mobile Pioneer 3-DX. Cette approche a été appliquée avec succès dans le cas de maintien de la formation, le suivi de trajectoire et l'évitement d'obstacle.

A noter que le temps de communication entre MATLAB et V-REP peut être la cause d'une dégradation des performances du robot pioneer cela est due au retard entre l'envoi de la commande par MATLAB et sa réception dans V-REP, c'était un grand défi de trouver un compromis entre communication et performances.

# **Conclusion générale**

## CONCLUSION GENERALE

Le travail réalisé répond à la volonté d'étudier les architectures de commande des systèmes multi-robots mobiles. L'intérêt est porté plus particulièrement sur la coopération des entités robotiques autonomes dans le but de réaliser des tâches spécifiques, Il consiste en premier lieu, à faire l'étude et la commande d'une formation de robots mobiles avec les approches développées dans la littérature pour les systèmes multi-robots, puis faire coopérer leurs actions pour aboutir à un but ou un objectif global du groupe.

Un système multi-robots évoque un certain nombre de problèmes qui n'apparaissent pas dans le cas mono-robot. En effet, la commande de formation de robots mobiles constitue une matière intéressante à l'étude. Le problème de commande est décomposé en trois sous-problèmes :

- La distribution des robots dans leurs positions désirées à l'intérieur de la formation : ils sont obligés d'assurer une configuration géométrique désirée.
- Le suivi de trajectoire : les robots doivent maintenir la formation lorsqu'ils sont en mouvement.
- Évitement de collision entre eux et avec les obstacles.

Les attentes principales de ce travail sont :

- Amélioration des performances de l'approche leader-follower par l'ajout d'un contrôleur flou pour la commande d'une formation de robots mobiles.
- Proposition d'une nouvelle méthode basée sur l'approche leader-follower et l'approche comportementale des champs de potentiels pour la commande d'une formation de robots mobiles. Cette méthode peut être utilisée pour définir n'importe quel type de formation géométrique.

Dans le premier chapitre nous avons présenté une introduction des systèmes multi-robots et la commande d'une formation de robots mobiles où le mot formation veut dire que le groupe de véhicules se déplace d'une manière ordonnée selon une configuration géométrique prédéfinie, ensuite nous avons cité les architectures et les approches de commande à savoir l'approche comportementales, l'approche leader-follower et l'approche structure virtuelle.

Le deuxième chapitre est scindé en deux parties : la première partie a été consacrée à la présentation des deux travaux qui concerne la commande coopérative appliquée pour la

formation. A savoir l'algorithme de leader-follower et l'algorithme champs de potentiel. Dans la deuxième partie nous avons montré deux approches proposées pour la commande d'une formation afin d'améliorer les performances dans un groupe de robots mobiles ; la logique floue ajoutée à l'approche leader-follower et l'approche hybride qui combine l'approche champs de potentiel et leader-follower, ces deux propositions ont donnée de meilleurs résultats pour les tâches effectuées par la formation à savoir : maintien de la formation, suivi de trajectoire et l'évitement des obstacles.

Les résultats des algorithmes précédents ont été présentés dans le chapitre trois. Ils sont acceptables et encourageants pour la commande d'une formation de robots mobiles :

1. Les résultats de simulation de l'approche leader-follower prouvent que cette dernière réalise le but de la mission avec un accomplissement performant : maintien de la formation et suivi de trajectoire.

2. Les résultats de simulation de l'approche champs de potentiel prouvent que cette dernière réalise le but de la mission avec un accomplissement performant dans l'évitement d'obstacle.

3. Les résultats de simulation de l'approche leader follower améliorée par la logique floue ont prouvé l'efficacité d'utiliser la logique floue pour le control de la distance entre leader et followers.

4. Pour remédier aux inconvénients des deux approches nous avons utilisé l'approche hybride, cette dernière a donné des résultats qui confirme qu'elle est capable d'assurer toutes les tâches : maintien de la formation, suivi de trajectoire et évitement d'obstacle.

Le chapitre quatre a été consacré à la présentation de l'environnement de simulation 3D : V-REP, où on a appliquée l'approche hybride pour la commande d'une formation de robots mobiles où l'entité utilisée est le robot Pioneer 3-DX. Cette approche a été appliquée avec succès dans le cas de maintien de la formation, le suivi de trajectoire et l'évitement d'obstacle. A noter que le temps de communication entre MATLAB et V-REP peut être la cause d'une dégradation des performances du robot Pioneer cela est due au retard entre l'envoi de la commande par MATLAB et sa réception dans V-REP, c'était un grand défi à trouver un compromis entre communication et performances.

Au cours de ce projet et à travers les travaux effectués nous avons pu acquérir beaucoup de connaissances qui peuvent nous être utiles par la suite pour des recherches plus approfondies, des connaissances sur le domaine de la commande coopérative des systèmes multi-robots et précisément les systèmes multi-robots, sur l'entité robotique « robot mobile », sa description et modélisation. Les algorithmes présentés peuvent être utilisé pour un grand

nombre de véhicules) comme elle peut être étendue facilement pour le cas tridimensionnel c'est-à-dire pour les UAVs.

Il sera intéressant dans la suite d'améliorer les performances de l'algorithme hybride en essayant de rendre la formation plus adaptative selon l'environnement du robot : changement de la configuration géométrique dans le cas d'évitement d'obstacle par exemple passage de la forme triangle à la forme ligne dans les passages étroits, crée des niveaux de décision qui gèrent la priorité entre les tâches confiées et l'ajout d'autre tâches au système multi-robot comme le « mapping ». Nous recommandons d'implémenter les algorithmes de commande coopérative sur un groupe de robots mobiles réels.

## BIBLIOGRAPHIE

- [1] R.A.Brooks, «A robust layered control system for mobile robot,» *IEEE Journal of Robotics and Automation* , vol. 2, pp. 14-23, 1985.
- [2] G.Dudek, «A taxonomy for multi-agent robotics,» *Autonomous Robots* 3, pp. 375-397, 1996.
- [3] G.HATTENBERGER, «Vol en formation sans formation contrôle et planification pour le vol en formation des avions sans pilote,» chez *Thèse de doctorat*, France, Université de Toulouse III-Paul Sabatier, 2008.
- [4] A.BAZOULA, «Aide à la Navigation et La Coopération de Robots Mobiles Autonomes,» chez *Thèse de doctorat*, Ecole Militaire Polytechnique, 2011.
- [5] Jacobsen, «A generic architecture for hybrid intelligent systems,» *IEEE World Congress on Computational Intelligence* , pp. 709-714, 1998.
- [6] K. e. M.Lewis, «High Precision Formation Control of Mobile Robots Using Virtual Structures,» *Autonomous Robots* , pp. 387-403, 1997.
- [7] R. Beard, «Decentralized Scheme for Spacecraft Formation Flying via the Virtual Structure,» 2004, pp. 73-82.
- [8] C. Reynolds, «Flocks, herds and schools: A distributed behavioral model,» *In Computer Graphics* , pp. 25-34, 1987.
- [9] Duarte, «Evolution of collective behaviors for a real swarm of aquatic surface robots,» 2014.
- [10] H. Tanner, G. Pappas et V. Kumer, «Leader-to-Formation Stability,» 2004.
- [11] Don Grundle, Robert Murphey, Panos Pardalos et Oleg Prokopyer, «Cooperative Systems : Control and Optimisation,» 2007.
- [12] T.S.CHIO, «Hierarchical formation control for a multi-robot team, thèse de doctorat,» Université de Washington, USA, 2002.

- [13] M.A.GOODRICH, «Potential Fields Tutorial,» 2002.
- [14] K. Kolzowski, «Artificial Potentiel Based Control for a Large Scale Formation of Mobile Robots,» *Fourth International Workshop on Robot Motion and Control* , pp. 285-291, 2004.
- [15] [En ligne]. Available: <http://www.coppeliarobotics.com>. [Accès le 16 06 2017].
- [16] [En ligne]. Available: <http://www.coppeliarobotics.com/helpFiles/en/remoteApiFunctionsMatlab.htm>. [Accès le 16 06 2017].
- [17] adept, «Pioneer 3-DX,» *adept mobilerobots*.
- [18] M.Defoort, « Contribution à la planification et à la commande pour les robots mobiles,thèse de doctorat,» Ecole Centrale de Lille, 2007.
- [19] A. ., A. M.Tadjin, «Invariant Manifold Approach for the Stabilization of Nonholonomic Chained Systems: Application to a Mobile Robot,» *Kluwer Academic Publishers* , 2001.
- [20] H.Büler, *Reglage par logique floue*, Press polytechnique et université, 1994.
- [21] M.Jimshidi, «Fuzzy logic and control,» chez *Printice Hall*, USA, 1993.
- [22] M.J.Mataric, «Issues and Approaches in Disgn of Collective Autonomous Agents,» *Robotics and Autonomous Systems* , vol. 16, pp. 321-331, 1995.

---

---

## ANNEXE A

# COMMANDE ET MODELISATION D'UN ROBOT MOBILE

Dans une formation de robots mobiles la condition la plus importante est la coordination de mouvement entre les robots, c'est-à-dire, les robots doivent garder une configuration relative appropriée pendant le déroulement de la mission. Ainsi, pour réaliser correctement la coordination il est nécessaire de comprendre comment commander le mouvement de chaque robot du groupe. Dans notre cas les robots mobiles utilisés sont des robots de type unicycle [4].

Ce type de robots a été étudié intensivement. En général, il est constitué d'une plateforme portée par des roues. Les roues permettent le mouvement relatif entre la plateforme et le sol. Ce genre de robot est mécaniquement simple et facile à construire.

## MODELISATION DU ROBOT MOBILE

### HYPOTHESE DE MODELISATION

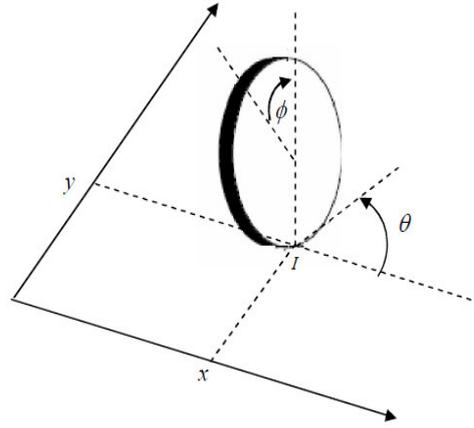
La problématique de la commande des robots mobiles étant trop vaste c'est pour cela nous présentons dans cette partie un certain nombre d'hypothèses simplificatrices [18] :

- Les véhicules sont considérés comme rigides et évoluant sur un plan.
- Les véhicules sont dotés de roues conventionnelles : le point de contact entre la roue et le sol est réduit à un point **I** et la roue est soumise à la contrainte de roulement sans glissement.

### ROULEMENT SANS GLISSEMENT

On parle souvent des robots holonomes et non holonomes, cette classification est liée au type de contraintes dans les systèmes mécanique (robot à roues). Les contraintes sont dites holonomes s'il est possible de les intégrer et elles aboutissent à des relations algébriques liant les paramètres de configuration. Ces relations peuvent être éliminées par un changement de variables approprié et le système est dit holonome. Dans le cas de contraintes non intégrables, l'élimination n'est plus possible et le système est dit non holonome. Dans le cas des véhicules sur roues, ces contraintes cinématiques résultent de l'hypothèse de roulement sans glissement.

Soit une roue verticale (**Figure A-4-13**) roulant sans glisser dans un plan la configuration de l'unicycle est caractérisée par la donnée des variables  $x, y, \theta$  et  $\phi$  où  $x$  et  $y$  dénotent les coordonnées du point de contact  $I$  de la roue avec le sol dans un repère cartésien donné  $R_0 = (O, \vec{i}, \vec{j}, \vec{k})$ ,  $\theta$  l'orientation de la roue et  $\phi$  l'angle de la roue mesurée à partir de la verticale.



**Figure A-4-13 Modèle de la roue**

Le roulement sans glissement se traduit par une vitesse nulle du point  $I$  de la roue en contact avec le sol.

$$\begin{aligned} \overline{V}(I|R_0) &= \dot{x}\vec{i} + \dot{y}\vec{j} + (\dot{\theta}\vec{k} + \phi(\sin\theta\vec{i} + \cos\theta\vec{j}))\Lambda(-r\vec{k}) \\ &= (\dot{x} - r\dot{\phi}\cos\theta)\vec{i} + (\dot{y} - r\dot{\phi}\sin\theta)\vec{j} \\ &= 0 \end{aligned} \quad (a.1)$$

Où  $r$  est le rayon de la roue. On en déduit deux contraintes :

$$\begin{cases} \dot{x} - r\dot{\phi}\cos\theta = 0 \\ \dot{y} - r\dot{\phi}\sin\theta = 0 \end{cases} \quad (a.2)$$

Qui peuvent se réécrire comme la suite :

$$\begin{cases} \dot{x}\cos\theta + \dot{y}\sin\theta = r\dot{\phi} \\ -\dot{x}\sin\theta + \dot{y}\cos\theta = 0 \end{cases} \quad (a.3)$$

Ces deux équations sont des contraintes de type non holonome. En introduisant  $v = r\dot{\phi}$  la vitesse de roulement de la roue et  $\omega = \dot{\theta}$  sa vitesse de rotation autour de l'axe  $\vec{k}$ , on déduit le modèle cinématique du robot mobile non-holonome de type unicycle qui est un modèle non linéaire :

$$\begin{cases} \dot{x} = v \cos \theta \\ \dot{y} = v \sin \theta \\ \dot{\theta} = \omega \end{cases} \quad (a.4)$$

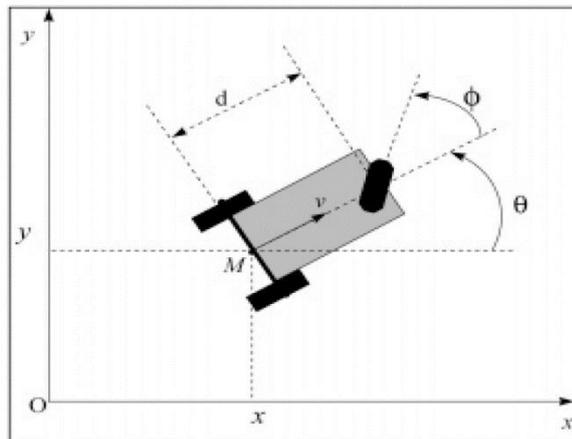
### MODELE DU ROBOT DITE « A CAR LIKE A MOBILE ROBOT »

Dans notre application le robot mobile montré dans la Figure A-4-14 est composé d'une roue motrice indépendante placée en avant du corps principal du robot et deux roues non commandées à l'arrière (roues folles) destinées uniquement à assurer la stabilité du robot. La commande de mouvement du robot peut être faite par le biais de la vitesse linéaire  $v$  du point  $M$  et la vitesse de rotation  $\omega$  [19].

La configuration du véhicule [Figure 2.2] peut être représentée par le quadruplet  $(x, y, \theta, \phi)$  où  $(x, y)$  sont les coordonnées du point  $M$ ,  $\theta$  l'orientation du véhicule,  $\phi$  l'angle de la roue mesurée à partir de la verticale  $d$  la distance entre le point  $M$  et le centre de la roue motrice.

Les conditions de roulement sans glissement s'obtiennent en écrivant que les vitesses latérales des roues avant et arrière sont nulles :

$$\begin{cases} -\dot{x} \sin \theta + \dot{y} \cos \theta = 0 \\ \dot{x} \sin(\theta + \phi) - \dot{y} \cos(\theta + \phi) - d \cos(\phi) \dot{\theta} = 0 \end{cases} \quad (a.5)$$



**Figure A-4-14 : Configuration du véhicule.**

Le modèle cinématique d'un robot mobile non-holonome devient après avoir transformé l'équation précédente :

$$\begin{cases} \dot{x} = u_1 \cos \theta \\ \dot{y} = u_2 \sin \theta \\ \dot{\theta} = u_2 \frac{\tan \phi}{d} \\ \dot{\phi} = u_2 \end{cases} \quad (a.6)$$

Avec :  $u_1$  c'est la vitesse de translation  $v$ ,  $u_2$  c'est la vitesse de rotation  $\omega$ .

## COMMANDE DU ROBOT MOBILE

### L'APPROCHE « INVARIANT MANIFOLD » POUR LA COMMANDE DU ROBOT MOBILE

Le problème est la stabilisation des systèmes non-holonomes qui peuvent être transformé en un système chaîné [19]:

$$\begin{aligned}\dot{x}_1 &= u_1 \\ \dot{x}_2 &= u_2 \\ \dot{x}_3 &= x_2 u_1 \\ &\vdots \\ &\vdots \\ \dot{x}_n &= x_{n-1} u_1\end{aligned}\tag{a.7}$$

Avec :  $X = (x_1, x_2, \dots, x_n)$  c'est le vecteur d'état,  $(u_1, u_2)$  c'est le vecteur de commande.

**Définition1** : soit l'application  $\Phi : R^n \rightarrow R^p$  .l'ensemble  $M = \{x \in R^n / \Phi(x) = 0\}$  est dit invariant pour le système  $\dot{x} = f(x, u)$  si tous les trajectoires qui commence à  $M$  à  $t = t_0$  reste dans cet ensemble pour  $t > t_0$  . En d'autre terme la dérivé de lie de  $\Phi$  par rapport à  $f$  est nulle pour toute  $x \in M$  .

**Définition2** : un ensemble  $M = \{x \in R^n / \Phi(x) = 0\}$  est dit asymptotiquement attractive dans l'ouvert  $\Omega$  de  $R^n$  si pour toutes  $t_0 \in R_+$  tel que  $(t_0) \in \Omega$   $\lim_{t \rightarrow \infty} x(t) \in M$  .

L'objectif est de construire une commande  $U = [u_1(x), u_2(x)]^T$  qui stabilise le système chaîné exprimée par :

$$u_1(x) = v_1(x) , \quad u_2(x) = v_2(x) + \omega_2(x).$$

### CONSTRUCTION DE LA COMMANDE AVEC L'APPROCHE « INVARIANT MANIFOLD »

On pose au début  $\omega_2(x) = 0$ , et on prend  $v_1(x)$  et  $v_2(x)$  comme un simple retour d'état, on aura :

$$u_1(x) = -k_1 x_1 \quad u_2(x) = -k_2 x_1 - k_3 x_2$$

avec :  $k_1$  et  $k_3$  sont des paramètres positives tel que  $k_1 \neq k_3$  , et  $k_2 \in R$  . En intégrant le vecteur  $\dot{x}$  par rapport au temps en aura :

$$\begin{aligned} x_1(t) &= x_{10} \exp(-k_1 t), \\ x_2(t) &= \left( x_{20} - \frac{k_2 x_{10}}{K_a} \right) \exp(-k_3 t) + \frac{k_2 x_{10}}{K_a} \exp(-k_1 t), \\ x_3(t) &= \frac{k_1}{K_b} x_{10} \left( x_{20} - \frac{k_2 x_{10}}{K_a} \right) \exp(-K_b t), \\ &\quad + \frac{k_2}{2K_a} x_{10}^2 \exp(-2k_1 t) + S_3(x_0), \\ x_4(t) &= \frac{k_1^2 x_{10}^2}{K_b(k_1 + K_b)} \left( x_{20} - \frac{k_2 x_{10}}{K_a} \right) \exp(-(k_1 + K_b)t) \\ &\quad + \frac{k_2 x_{10}^3}{6K_a} \exp(-3k_1 t) + x_{10} S_3(x_0) \exp(-k_1 t) + S_4(x_0), \\ &\quad \vdots \\ x_n(t) &= \frac{k_1^{n-2} x_{10}^{n-2}}{K_b(K_b + k_1) \cdots (K_b + (n-3)k_1)} \left( x_{10} - \frac{k_2 x_{10}}{K_a} \right) \exp(-(K_b + (n-3)k_1)t) \\ &\quad + \frac{k_2 x_{10}^{n-1}}{(n-1)! K_a} \exp(-(n-1)k_1 t) \\ &\quad + \sum_{i=1}^{n-3} \frac{x_{10}^i}{i!} S_{n-i}(x_0) \exp(-k_1 i t) + S_n(x_0), \end{aligned}$$

Avec  $k_a = k_1 - k_3$ ,  $k_b = k_1 + k_3$  et  $S_3(x_0), S_4(x_0), \dots, S_n(x_0)$  les constantes d'intégration qui sont déterminées à  $t = 0$  on aura :

$$\begin{aligned}
 S_3(x) &= x_3 - \frac{k_1}{K_b} x_1 x_2 + \frac{k_2}{2K_b} x_1^2, \\
 S_4(x) &= x_4 - x_1 x_3 + \frac{k_1}{(k_1 + K_b)} x_1^2 x_2 - \frac{k_2}{3(k_1 + K_b)} x_1^3, \\
 S_5(x) &= x_5 - x_1 x_4 + \frac{1}{2} x_1^2 x_3 - \frac{k_1}{2(2k_1 + K_b)} x_1^3 x_2 + \frac{k_2}{8(2k_1 + K_b)} x_1^4, \\
 &\vdots \\
 S_n(x) &= x_n + \sum_{i=1}^{n-3} \frac{(-1)^i x_1^i x_{n-i}}{i!} + \frac{(-1)^n k_1 x_1^{n-2}}{(K_b + (n-3)k_1)(n-3)!} \\
 &\quad + \frac{(-1)^{n-1} k_2 x_1^{n-1}}{(K_b + (n-3)k_1)(n-3)!(n-1)}.
 \end{aligned}$$

Soit les vecteurs et les matrices suivants :

$$S = \begin{pmatrix} S_3 \\ S_4 \\ \vdots \\ S_n \end{pmatrix}, \quad C = \begin{pmatrix} c_3 \\ c_4 \\ \vdots \\ c_n \end{pmatrix}, \quad A = \begin{pmatrix} k_1 & 0 & \dots & 0 \\ 0 & 2k_1 & & \\ & & \ddots & \\ \vdots & & \ddots & \vdots \\ 0 & & \dots & 0 & (n-2)k_1 \end{pmatrix},$$

$$B = \begin{pmatrix} b_3 \\ b_4 \\ \vdots \\ b_n \end{pmatrix} = \begin{pmatrix} \frac{-k_1}{K_b} \\ \frac{k_1}{k_1 + K_b} \\ \vdots \\ \frac{(-1)^n k_1}{(n-3)!(k_b + (n-3)k_1)} \end{pmatrix}$$

$$Q(x_1) = \begin{pmatrix} \frac{1}{x_1} & 0 & \dots & 0 \\ 0 & \frac{1}{x_1^2} & \ddots & \vdots \\ \vdots & \ddots & \ddots & 0 \\ 0 & \dots & 0 & \frac{1}{x_1^{n-2}} \end{pmatrix}.$$

La variable  $\omega_2$  devient :  $\omega_2 = C^T Q(x_1)S$ . On applique cette commande sur le robot mobile : les coordonnées  $(x, y, \theta, \phi)$  définies dans le sous ensemble  $T$  suivant :

$T = \{(x, y, \theta, \phi) \in R^4 / \theta \neq \frac{\pi}{2} \text{ mod } \pi, \phi \neq \frac{\pi}{2} \text{ mod } \pi\}$  On peut rendre le système du robot mobile un système chainé avec les transformations suivantes :

$$\begin{aligned} x_1 &= x & , & & x_2 &= \frac{\tan \phi}{d \cos^3 \theta} \\ x_3 &= \tan \theta & , & & x_4 &= y \end{aligned} \quad (a.13)$$

Et la vitesse de translation et rotation devient :

$$\begin{aligned} v &= \frac{u_1}{\cos \theta} \\ \omega &= -\frac{3 \sin^2 \phi \sin \theta}{d \cos^2 \theta} u_1 + d \cos^2 \phi \cos^3 \theta u_2 \end{aligned} \quad (a.14)$$

Dans notre simulation on a choisi  $k_1 = 1, k_2 = 0, k_3 = 3$  et  $k_b = k_1 + k_3$  aussi  $S = [S_3 \ S_4]$  avec :

$$\begin{aligned} S_3(x) &= x_3 - \frac{k_1}{K_b} x_1 x_2 + \frac{k_2}{2K_b} x_1^2, \\ S_4(x) &= x_4 - x_1 x_3 + \frac{k_1}{(k_1 + K_b)} x_1^2 x_2 - \frac{k_2}{3(k_1 + K_b)} x_1^3. \end{aligned}$$

Les matrices A, B et Q sont donnée par :

$$A = \begin{bmatrix} k_1 & 0 \\ 0 & 2k_1 \end{bmatrix}, \quad B = \begin{bmatrix} \frac{-k_1}{K_b} \\ \frac{k_1}{k_1 + K_b} \end{bmatrix}, \quad \text{Et} \quad v_1 = \begin{bmatrix} \frac{1}{x_1} & 0 \\ 0 & \frac{1}{x_1^2} \end{bmatrix}.$$

Et on choisi  $C^T = [-24 \ -60]$  pour avoir les pôles égale à -1 et -2. Les résultats de simulation : trajectoire du robot et les commandes de vitesse de translation et de rotation sont représentés ci-dessous :

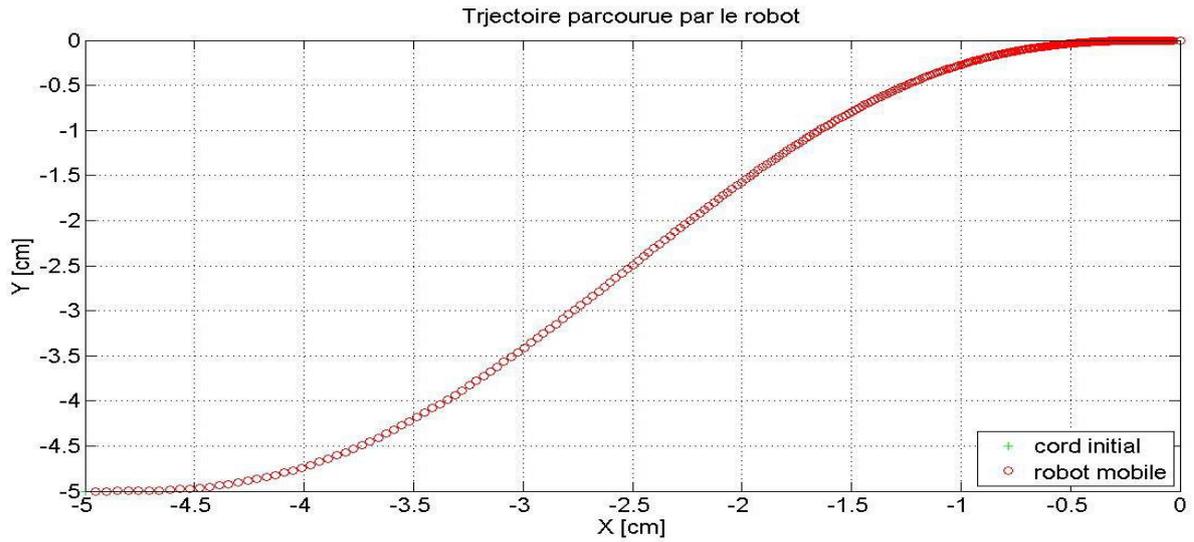


Figure A-4-15 : Trajectoire du robot mobile

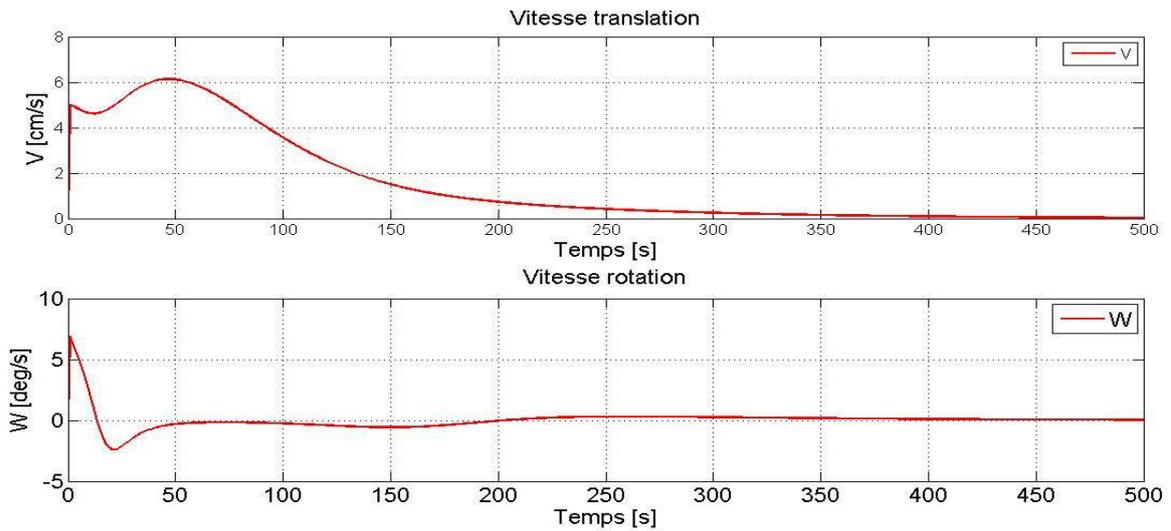


Figure A-4-16: Commande du robot mobile

## ANNEXE B

### Rappels théoriques

#### Notions de base sur la logique floue

Nous allons présenter les concepts de la théorie de la logique floue nécessaires pour la conception des régulateurs flous. [20]

- **Ensemble floue** : Soit  $X$  une collection continue ou discrète d'objets dénotés  $\{X\}$ ,  $X$  est appelé univers de discours et représente l'élément générique de  $x$ . Un ensemble flou peut être vu comme une généralisation du concept d'ensemble ordinaire dont la fonction d'appartenance prend seulement deux valeurs  $\{0,1\}$  (ensemble booléen). Ainsi sur un univers de discours  $X$ , un sous ensemble flou  $A$  est caractérisé par une fonction d'appartenance  $\mu_A$  prenant ses valeurs dans l'intervalle  $[0,1]$ , alors l'ensemble flou  $A$  dans  $X$  est défini par l'ensemble des paires (élément générique, degrés d'appartenance).

$$A = \{(x, \mu_x(x)) / x \in X\}$$

- **Variable linguistiques** : La description d'une certaine situation, d'un phénomène ou d'un procédé contient en général des expressions linguistiques (floues), comme : proche, loin, grand, petit, ...etc. Les expressions de ce genre forment les valeurs d'une variable linguistique, qu'on appelle valeurs floues qu'on peut représenter par des fonctions d'appartenance.

- **Fonction d'appartenance** : On associe à chaque valeur de la variable linguistique une fonction d'appartenance désignée par  $\mu_A(x)$  où  $x$  est la variable linguistique, tandis que  $A$  indique l'ensemble concerné. Une valeur précise de  $\mu_A(x)$ , sera désignée par le degré ou le facteur d'appartenance. Il est à noter que l'ensemble des éléments de  $x$  pour lesquels  $\mu_A(x) > 0$ , est appelé support de  $A$ . On va citer seulement la fonction d'appartenance qu'on va utiliser pour notre commande, qui est une **fonction triangulaire** définie par trois paramètres  $\{a, b, c\}$ , qui déterminent les coordonnées des trois sommets [21]:

$$\mu(x) = \max\left(\min\left(\frac{x-a}{b-a}, \frac{c-x}{c-b}\right), 0\right)$$

- **Raisonnement flou** : Les ensembles flous et la théorie des possibilités sont des éléments importants de la représentation des connaissances imparfaitement définies. Pour

raisonner sur de telles connaissances, la logique classique ne suffit pas. On utilise la logique floue lorsque les connaissances sont imprécises, vagues et éventuellement incertaines. Voici quelques définitions utilisées :

Soit  $x$  une variable linguistique et  $A$  une caractéristique :

- **Définition d'une proposition :** Une proposition floue est définie à partir d'une variable linguistique ( $x, T(x), X$ ) par la qualification : " **$x$  est  $A$** ".
- **Définition de la conjonction :** La conjonction de deux propositions floues est réalisée par l'opérateur **ET** par exemple : " **$x_1$  est  $A_1$  ET  $x_2$  est  $A_2$** ".
- **Définition de la disjonction :** La conjonction de deux propositions floues est réalisée par l'opérateur **OU** par exemple : " **$x_1$  est  $A_1$  OU  $x_2$  est  $A_2$** ".
- **Définition de l'implication :** L'implication entre deux propositions floues définit aussi une proposition floue que l'on peut exprimer par : " **$x_1$  est  $A_1$  ALORS  $x_2$  est  $A_2$** ".

Généralement, on peut construire des propositions floues par conjonction, disjonction ou implication portant sur des propositions floues élémentaires.

- **Définition d'une règle floue :** Une règle floue est une proposition floue utilisant une implication entre deux propositions floues quelconques. Par exemple : "**SI  $x_1$  est  $A_1$  ET  $x_2$  est  $A_2$  ALORS  $x_3$  est  $A_3$** ".

Où : " **$x_1$  est  $A_1$  ET  $x_2$  est  $A_2$** " est la prémisse de la règle et " **$x_3$  est  $A_3$** " est sa conclusion. En utilisant les règles de composition d'inférence, nous pouvons formaliser une procédure d'inférence, appelé raisonnement flou sur l'ensemble des règles (**Si - Alors**).

### **Notions de base sur la commande par logique floue**

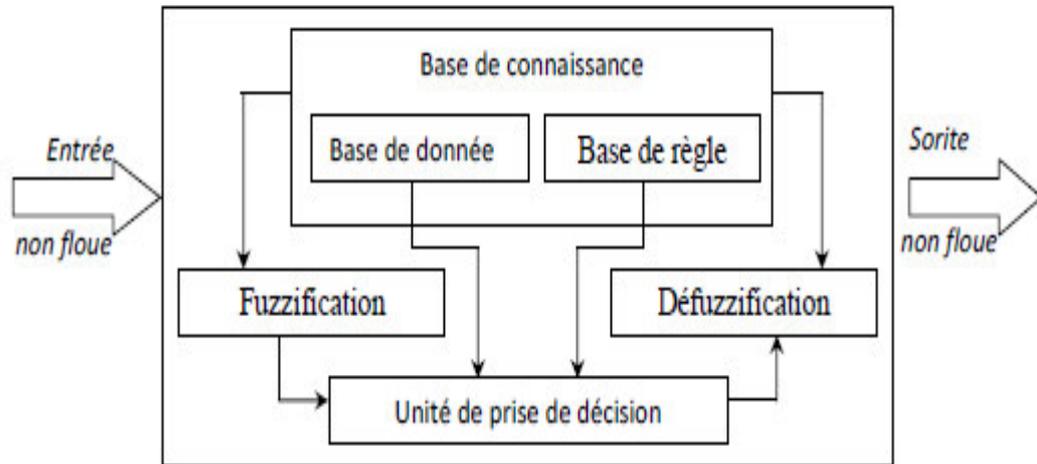
Le but de la commande par logique floue est de traiter des problèmes de commande de processus, le plus souvent à partir des connaissances des experts.

#### **Contrôleur flou :**

La configuration de base d'un contrôleur flou (**Figure B-1**) comprend les éléments suivants :

- **Fuzzification :** C'est la partie du contrôleur flou chargée de convertir les valeurs numériques en variables linguistiques.

- **Base de règles :** Elle contient les définitions des termes utilisés dans la commande et l'ensemble des règles caractérisant la cible de la commande et décrivant la conduite de l'expert.



**Figure B-1 : Structure d'un contrôleur flou**

- **Inférence :** L'inférence transforme à l'aide du jeu de règles (en manipulant la base de règles). La partie floue issue de la Fuzzification en une nouvelle partie floue qui caractérise la sortie du contrôleur.
- **Défuzzification :** La défuzzification consiste à convertir la partie floue issue de l'inférence en une valeur numérique.

Plusieurs stratégies de défuzzification sont utilisées :

- **Méthode du maximum :** La commande est égale à la valeur dont le degré d'appartenance est le plus fort.
- **Méthode de la moyenne des maximas :** La commande sera égale à la moyenne des valeurs dont le degré d'appartenance est maximal.
- **Méthode du centre de gravité :** C'est la méthode la plus utilisée dans les contrôleurs flous. Dans celle-ci la commande sera égale au centre de gravité de l'ensemble flou de sortie, on obtient donc pour :
- **Univers de discours discret :** 
$$u = \frac{\sum_{i=1}^n x_i \mu_A(x_i)}{\sum_{i=1}^n \mu_A(x_i)}$$
- **Univers de discours continu :** 
$$u = \frac{\int x \mu_A dx}{\int \mu_A dx}$$

**Types de régulateurs flous :**

Il existe plusieurs types de régulateurs flous, qui diffèrent de par leur mécanisme d'inférence utilisé, on en cite : régulateur de Mamdani, de Sugeno, ... etc.

Le régulateur flou qu'on a utilisé est un régulateur de type Mamdani : il utilise le minimum comme opérateur de jonction et l'implication pour représenter le graphe flou associé à chaque règle et l'opérateur maximum pour l'agrégation.

Dans une règle  $i$  : **Si  $x_1$  est  $A_1$  et ... et  $x_n$  est  $A_n$  Alors  $y$  est  $B_i$**

Où les  $B_i$  sont des sous-ensembles flous. Les  $B_i$  forment en général une partition de l'univers de sortie.

## ANNEXE C

## Étapes nécessaires pour établir une connexion entre MATLAB et V-REP :

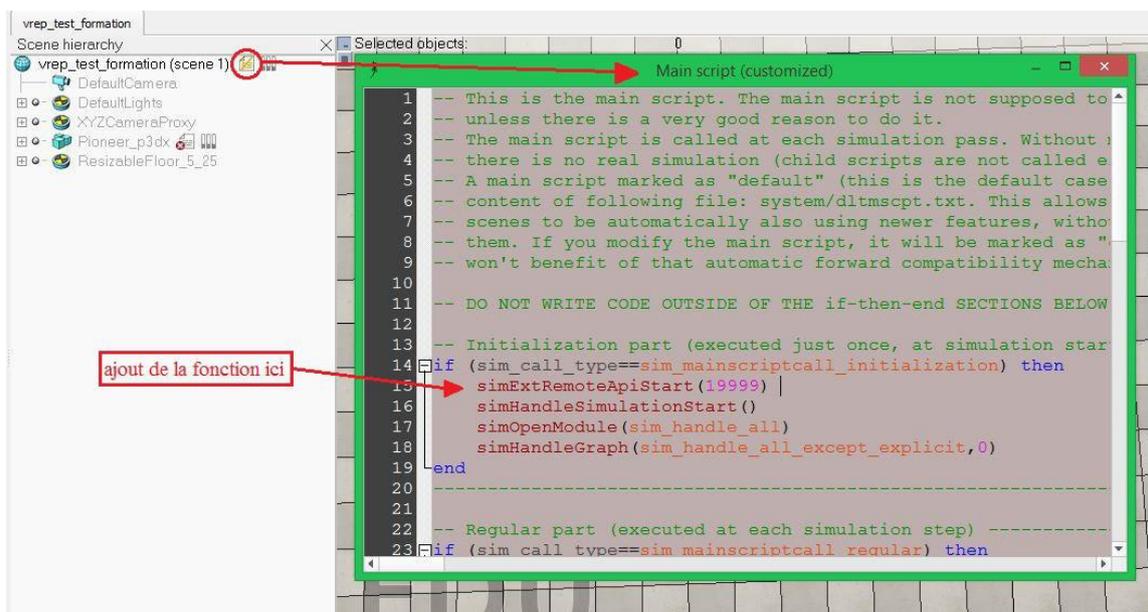
**Etape 1 :** Créer un dossier qui va jouer le rôle du « **Workspace** » (espace de travail).

**Etape 2 :** Copier les trois fichiers suivant du répertoire de V-REP : [C:\Program Files\V-REP3\V-REP\_PRO\_EDU\programming\remoteApiBindings\matlab\matlab] vers le « **Workspace** » :

- remApi.m.
- remoteApiProto.m.

**Etape 3 :** Copier le fichier [remoteApi.dll] du répertoire V-REP : [C:\Program Files\V-REP3\V-REP\_PRO\_EDU\programming\remoteApiBindings\lib\lib], ici on doit choisir l'extension qui correspond au celle de **MATLAB** qui nous utilisons soit **x64 bit** ou **x32 bit**.

**Etape 4 :** Lancer **V-REP**, ensuite crée une nouvelle scène et ouvrir le « **Main Script** » puis dans la condition de l'initialisation ajouté le appelle de la fonction { **simExtRemoteApiStart(19999)** } (**Figure c-1**), qui va assurer la connexion avec **MATLAB** par « **RemoteAPI** » à chaque lancement de simulation sur le port numéro 19999, Enfin sauvegarder la scène dans le « **Workspace** » (créée à l'étape 1).



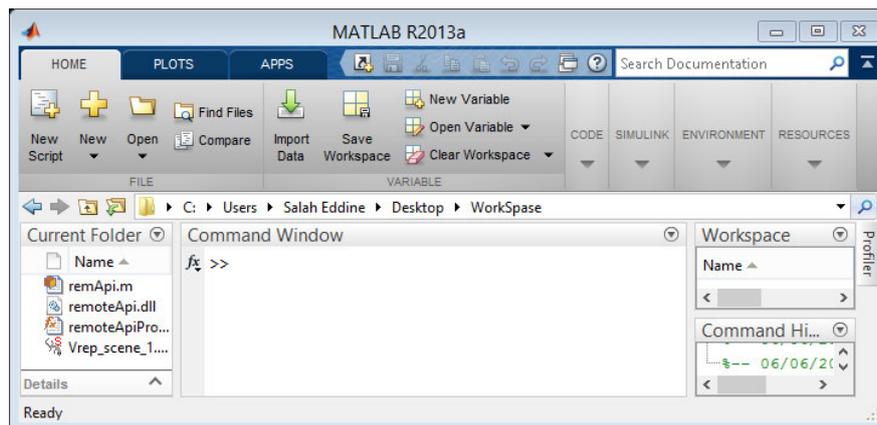
**Figure c-1 : Etablissement de la connexion.**

**Etape 5 :** Maintenant que nous avons configuré la scène V-REP et copier les fichiers de la bibliothèque, on doit pointer le « **WorkSpace** » de MATLAB vers notre « **WorkSpace** » (le dossier crée à l'étape 1) (**Figure c-2**).

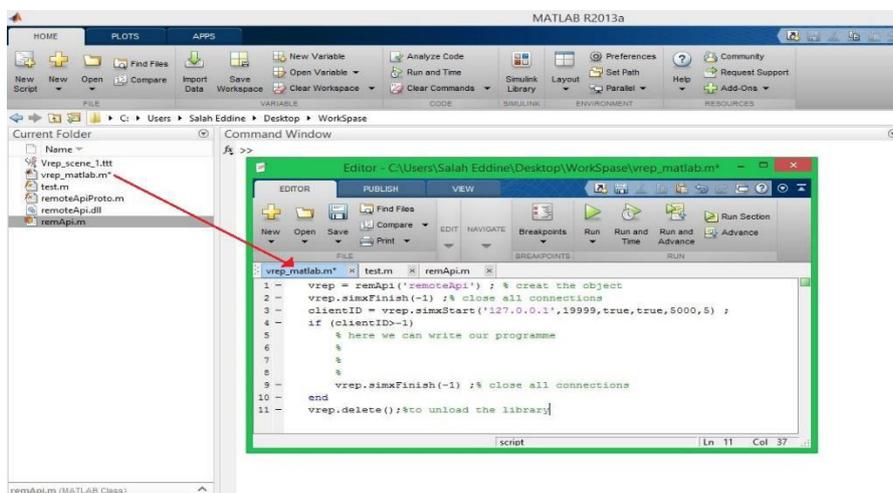
**Etape 6 :** Crée un nouveau script [Matlab---> Home---> New---> Script]. Dans ce script on va établir la connexion avec V-REP comme suit (**Figurec-3**) :

- Création d'un nouvel objet de type « **remApi** », nommé par exemple vrep.
- Fermer toutes les connexions ouvertes avec V-REP en utilisant la fonction « **vrep.simxFinish (-1)** » pour éviter la possibilité perturbé notre travail.
- Crée une connexion avec V-REP par la fonction : « **vrep.simxStart** » en utilisant l'adresse IP local « **127.0.0.1** » sur le port « **19999** ».

Une fois qu'on termine l'établissement de la connexion entre **V-REP** et **MATLAB** on peut commencer notre travail, Il est préférable d'appelé le destructeur : « **delete ()** » à la fin du travail.



**Figure c-2 : WorkSpace MATLAB.**



**Figure c-3 : Création de la connexion.**