

**REPUBLIQUE ALGERIENNE DEMOCRATIQUE ET POPULAIRE**  
Ministère de l'Enseignement Supérieur et de la Recherche Scientifique



**Université Batna 2 – Mostefa Ben Boulaïd**



**Faculté de Technologie**

**Département d'Electronique**

**Thèse**

Préparée au sein du laboratoire d'électronique avancée

Présentée pour l'obtention du diplôme de :

**Doctorat en Sciences en Electronique**

**Option : Contrôle**

Sous le Thème :

**Commande robuste intelligente appliquée**

**au suivi de trajectoires d'un**

**Système robotique**

Présentée par :

**BENAZIZA Walid**

**Devant le jury composé de :**

M. ABDESSEMED Yassine	Prof.	Université Batna 2	Président
M. SLIMANE Nouredine	Prof.	Université Batna 2	Rapporteur
M. HASSAM Abdelouahab	Prof.	Université de Sétif 1	Examineur
M. BOUOUDEN Sofiane	Prof.	Université de Khenchela	Examineur

**Janvier 2019**

## *Dédicace*

*À la mémoire de mon frère Nadhir*

*À la mémoire de ma grand-mère*

*À mes très chers parents*

*À ma femme, ainsi que mes frères et sœur*

*À toute mes amis sans exception.*

# REMERCIEMENTS

Je tiens à assurer de ma sincère et totale gratitude et de ma profonde reconnaissance mon directeur de thèse, Monsieur Pr. **Noureddine SLIMANE** qui par sa confiance, ses conseils et ses explications pertinentes a contribué de manière essentielle à la finalisation de ce modeste travail. Il m'a aidé à me structurer et à travailler en toute quiétude. Sa bonne humeur communicative et ses conseils judicieux en théorie de la commande par mode de glissement m'ont permis d'atteindre ces résultats.

Je suis très honoré que monsieur Yassine ABDESSEMED professeur à l'université de Batna 2 ait accepté de présider le jury délibération de cette thèse de doctorat, je le remercie aussi pour ses encouragements et sa bienveillance.

J'exprime ma profonde reconnaissance aux membres du jury qui ont accepté de juger cette thèse et m'ont permis d'améliorer son contenu par leurs remarques pertinentes :

Monsieur le professeur Abdelouahab HASSAM de l'université de Sétif 1 et Monsieur le professeur Sofiane BOUOUDEN de l'université de Khenchela pour l'intérêt qu'ils ont porté à ce travail en acceptant d'examiner cette thèse de doctorat.

Je remercie en particulier mon cher frère Nadhir que dieu l'accueille dans son vaste paradis qui m'a accompagné moralement tout au long de ce parcours.

Je remercie également tous les membres de ma famille.

Je remercie très spécialement mon collègue Dr. Ali MALLEM, enseignant de l'université de khenchela, pour toute l'aide qu'il m'a apportée.

Je n'oublie pas d'adresser mes vifs remerciements à mes amis Halim et Hafed BENDIB.

Mes remerciements vont également à tous les enseignants ainsi qu'à tous les responsables de notre département.

## Résumé

Le travail présenté dans cette thèse vise principalement le suivi de trajectoires pour un robot mobile non holonome. Le travail est divisé en deux parties essentielles; la première partie considère la localisation d'un robot mobile dans un environnement complètement connu. L'algorithme de localisation est basé sur le filtre de kalman étendu (EKF), en tenant compte que l'environnement est structuré par un certain nombre d'amers statiques dont les positions sont à priori connues. Les entrées de commande qui sont les vitesses et les observations sont considérées entachées de bruits. Des travaux de simulations sont illustrés pour confirmer la performance de l'algorithme dans différents cas d'environnement. La deuxième partie consiste à proposer des algorithmes hybrides de suivi de trajectoires pour un robot non holonome, les algorithmes sont basés sur le mode glissant terminal et le mode glissant classique ainsi que la logique floue. Plusieurs techniques, utilisant le PSO et les algorithmes génétiques (GA) pour l'obtention des paramètres optimaux des contrôleurs sont utilisées pour assurer une convergence rapide du robot vers les trajectoires de référence et pour lui permettre d'arriver convenablement à son but et tout cela en présence de perturbations. Des travaux de simulation sont élaborés pour montrer la robustesse des lois de commande proposées sous différentes trajectoires de références.

**Mots Clés:** Robot mobile, estimateur EKF, localisation, mode glissant terminal, logique floue, PSO, algorithmes génétiques (GA), suivi de trajectoires, stabilité de Lyapunov.

## Abstract

The work presented in this thesis aims mainly the trajectory tracking for a nonholonomic mobile robot. The work is divided in two essential parts; the first part considers the mobile robot localization in a completely known environment. The proposed localization algorithm is based on the extended kalman filter (EKF), taking into account that the environment is structured by a number of static landmarks, which positions are known. The control inputs, which are the velocities and the observations, are considered sullied by noises. Simulation works are illustrated to confirm the proposed algorithm performance in different environmental cases. The second part consists to propose hybrid algorithms of trajectory tracking for nonholonomic robot, the algorithms are based on terminal sliding mode, classical sliding mode and fuzzy logic. Several techniques, using PSO and genetic algorithms (GA) for obtaining the optimal parameters of the controllers are used to ensure a fast convergence of the robot towards the reference trajectories and to enable him to arrive suitably at its goal and all that in the presence of disturbances. Simulation works are undertaken to show the robustness of the proposed control laws under various references trajectories.

**Key Words:** Mobile robot, EKF estimator, localization, terminal sliding mode, fuzzy logic, PSO, genetic algorithms (GA), trajectories tracking, Lyapunov stability.

## ملخص

يهدف العمل المقدم في هذه الرسالة وبشكل أساسي إلى اتباع مسار روبوت متقل غير متخصص. ينقسم العمل إلى قسمين أساسيين؛ الجزء الأول يعتبر تموضع الروبوت المحمول في بيئة معروفة. تعتمد خوارزمية التموضع المقترحة على مرشح كالمان الموسع (Kalman)، مع الأخذ بعين الاعتبار أن البيئة يتم تنظيمها بواسطة عدد معين من المعالم الأولية المعروفة. تعتبر المعالم المستخدمة بشكل عام ثابتة بالإضافة إلى ضوضاء التحكم في السرعة وضوضاء المستشعر أثناء المراقبة. سيتم توضيح أعمال المحاكاة لتأكيد كفاءة الخوارزمية المقترحة في الحالات البيئية المختلفة. الجزء الثاني يتمثل في اقتراح خوارزميات هجينة لتتبع مسار الروبوت غير المتخصص، وتستند هذه الخوارزميات إلى وضع الانزلاق الطرفي ووضع الانزلاق الكلاسيكي وكذلك المنطق الضبابي. من جهة أخرى، وفي ظل بيئة مضطربة سيتم تحسين العديد من التقنيات، باستخدام خوارزمية التحسين بسرب الجسيمات (PSO) والخوارزميات الجينية (GA) للحصول على الإعدادات المثلى لضمان وصول الروبوت إلى معلمه الأصلي بشكل مثالي للمسارات المرجعية المختلفة.

**الكلمات الدالة:** الروبوت المحمول، مقدر كالمان (EKF)، التحكم في الوضع المنزلق، المنطق الضبابي،

خوارزمية التحسين بسرب الجسيمات (PSO)، الخوارزميات الجينية (GA)، وظيفة Lyapunov

# TABLES DES MATIERES

Introduction générale .....	1
<b>Chapitre 1 : Aperçu sur la robotique mobile</b>	
1.1 Introduction.....	7
1.2 Robots mobiles.....	7
1.3 Perception .....	9
1.4 Méthodes de navigation .....	10
1.4.1 Approche d'un objet .....	10
1.4.2 Guidage .....	11
1.4.3 Action associe à un lieu .....	10
1.4.4 Navigation topologique.....	11
1.4.5 Navigation métrique .....	11
1.4.6 Navigation par reconstruction intérieur .....	12
1.5 Méthode d'évitement d'obstacles .....	13
1.5.1 Méthode de champ potentiel(PFM) .....	13
1.5.2 Méthode de restriction d'obstacle(ORM) .....	14
1.5.3 Approche de fenêtre dynamique .....	16
1.5.4 Autre approches .....	16
1.6 Types de robot mobile et leur modélisation .....	17
1.6.1 Modélisation cinématique des robots mobiles .....	17
1.6.1.1 Robot unicycle .....	17
1.6.1.2 Robot tricycle.....	19
1.6.1.3 Robot de type voiture .....	21
1.6.2 Modélisation dynamiques des robots mobiles.....	23
1.6.2.1 Robot différentiel .....	24
1.6.2.2 Robot unicycle .....	29
1.7 Conclusion .....	30

## Chapitre 2 : Localisation des robots mobiles

2.1 Introduction .....	35
2.2 Types d'environnements .....	32
2.2.1 Environnements intérieurs et extérieurs .....	33
2.2.2 Environnements statique et dynamique .....	33
2.3 Différentes représentations d'environnements .....	33
2.3.1 Cartes topologiques .....	33
2.3.2 Cartes métriques .....	35
2.3.3 Modélisation hybride .....	35
2.3.4 Modèle de primitives (carte d'amers) .....	35
2.4 Types de capteurs .....	36
2.4.1 Capteurs proprioceptifs .....	36
2.4.1.1 L'odométrie .....	36
2.4.1.1 Systèmes inertiels .....	36
2.4.2 Capteurs extéroceptifs .....	36
2.4.2.1 Capteurs ultrasons .....	37
2.4.2.2 Télémètres lasers .....	37
2.4.2.3 Systèmes de vision .....	38
2.5 Différentes techniques de la localisation .....	38
2.5.1 Localisation relative .....	38
2.5.2 Localisation absolue .....	39
2.5.3 Approches probabilistes .....	39
2.6 Localisation basée sur le filtre de kalman étendu .....	40
2.6.1 Filtre de kalman linéaire(LKF).....	40
2.6.2 Filtre de kalman non linéaire(FKE).....	41
2.6.3 Description des modèles .....	42
2.6.3.1 Modèle d'évolution du robot mobile .....	42
2.6.3.2 Modèle d'amer et le modèle d'observation .....	43
2.6.4 Algorithme de localisation basé sur le EKF .....	45
2.6.4.1 Prédiction .....	46
2.6.4.2 Observation .....	47
2.6.4.3 Appariements .....	48
2.6.4.4 Mise à jour .....	48

2.7 Travaux de simulation .....	49
2.8 Conclusion .....	56

**Chapitre 3 : Elaboration d’algorithmes robustes par mode glissant flou avec optimisation par PSO pour le suivi de trajectoires**

3.1 Introduction .....	57
3.2 Mode de glissement terminal rapide (FTSM).....	58
3.2.1 Conception de la surface de glissement par la méthode FTSM .....	58
3.2.1.1 Méthode traditionnelle (FTSM) .....	58
3.2.2.2 Méthode globale (GFTSM) .....	59
3.3 Généralité sur la logique floue type-1 .....	60
3.3.1 Fuzzification .....	60
3.3.2 Règles de la logique floue .....	62
3.3.3 Méthode d’inférence floue .....	62
3.3.3.1 Implication de Mamdani .....	62
3.3.3.2 Implication de Takagi-Sugeno kang (TSK).....	63
3.3.4 Défuzzification .....	64
3.4 Commande cinématique par mode de glissement flou .....	65
3.4.1 Modélisation cinématique du robot mobile .....	65
3.4.2 Conception de la loi de commande .....	67
3.4.2.1 Choix de la fonction de commutation .....	67
3.4.2.2 Loi de commande .....	68
3.4.2.3 Choix de la fonction de Lyapunov .....	69
3.4.2.4 Application de la logique floue .....	69
3.5 Commande dynamique robuste par mode glissant .....	72
3.5.1 Modèle dynamique du robot mobile .....	72
3.5.2 Contrôle dynamique basé sur le mode glissant .....	73
3.6 Commande dynamique robuste par PSO et mode glissant.....	76
3.6.1 Optimisation par PSO .....	76
3.6.2 Principe d’optimisation .....	77
3.6.3 Application de PSO .....	78
3.7 Travaux de simulation et résultats .....	80
3.8 Conclusion .....	90

**Chapitre 4 : Commande robuste intelligente appliquée au suivi de trajectoire d’un robot mobile**

4.1 Introduction .....	<b>91</b>
4.2 Mode de glissement terminal (TSM) .....	<b>92</b>
4.3 Algorithmes génétiques(GA) .....	<b>93</b>
4.3.1 Introduction .....	93
4.3.2 Représentation et évaluation des chromosomes .....	94
4.3.3 Évaluation de population .....	95
4.3.4 Sélection et reproduction .....	96
4.3.5 Croisement .....	96
4.3.5.1 Croisement en un point .....	97
4.3.5.2 Croisement en deux points .....	97
4.3.6 Mutation .....	97
4.4 Application d'une commande robuste pour le suivi d'une trajectoire .....	<b>98</b>
4.4.1 Commande cinématique par mode de glissant flou .....	98
4.4.1.1 Contrôle de la vitesse angulaire .....	98
4.4.1.2 Contrôle de la vitesse linéaire .....	99
4.4.2 Commande robuste par mode de glissant terminal basée sur AG .....	101
4.4.2.1 Commande dynamique sans AG.....	101
4.4.2.2 Commande dynamique avec AG .....	103
4.5 Travaux de simulation .....	<b>106</b>
4.6 Conclusion .....	<b>118</b>
Conclusion générale .....	<b>119</b>
Bibliographie .....	<b>122</b>



# INDEX DES FIGURES

1.1	Exemple de robot à jambes .....	8
1.2	Exemple de robot à roues.....	8
1.3	Exemple d'un processus de perception .....	9
1.4	Action associée à un lieu .....	11
1.5	Navigation topologique .....	11
1.6	Navigation métrique .....	12
1.7	Carte d'intérieur 3D à partir d'un plan vertical balayé LADAR .....	12
1.8	(a) Sens résultant du déplacement (b) Lignes du champ de potentiel .....	13
1.9	(a) Répartition des sous-objectifs $x_i$ . (b) Les deux ensembles de directions indésirable .....	15
1.10	Approche dynamique de la fenêtre .....	16
1.11	Robot unicycle .....	17
1.12	Robot tricycle .....	19
1.13	Robot type voiture .....	21
1.14	Repérage d'un robot mobile de type différentiel .....	25
1.15	Repérage d'un robot mobile de type unicycle .....	29
2.1	Positionnement du robot .....	31
2.2	Organigramme de navigation.....	32
2.3	Représentation topologique d'un espace d'intérieur.....	34
2.4	Représentation topologique des pièces et des couloirs .....	34
2.5	Représentation d'un modèle d'une carte d'amers .....	35
2.6	Schéma de télémétrie laser par mesure de déphasage .....	37
2.7	Principe de l'odométrie .....	39
2.8	Modèle d'un robot unicycle .....	42
2.9	Distance et angle de direction mesurés par télémétrie .....	44

2.10	Organigramme de la localisation par le FKE .....	45
2.11	Environnement constitué de 20 amers .....	50
2.12	Localisation obtenue après 500 itérations avec 20 amers .....	50
2.13	Erreurs de position et d'orientation après 500 itérations .....	51
2.14	Localisation obtenue après 300 itérations avec 20 amers .....	51
2.15	Erreurs de position et d'orientation après 300 itérations .....	52
2.16	Localisation obtenue après 500 itérations avec 24 amers .....	52
2.17	Erreur de position et d'orientation après 500 itérations avec 24 amers .....	53
2.18	Localisation obtenue après 300 itérations avec 24 amers .....	53
2.19	Erreur de position et d'orientation après 300 itérations avec 24 amers.....	54
2.20	Localisation obtenue après 500 itérations avec 30 amers .....	54
2.21	Erreur de position et d'orientation après 500 itérations avec 30 amers .....	55
2.22	Localisation obtenue après 300 itérations avec 30 amers .....	55
2.23	Erreur de position et d'orientation après 300 itérations avec 30 amers .....	56
3.1	Diagramme de la logique floue type 1 .....	60
3.2	Formes de la fonction d'appartenance .....	62
3.3	Diagramme du robot mobile .....	65
3.4	Architecture de la commande cinématique par mode glissant flou .....	70
3.5	Ensembles flous d'entrée $s_2 \dot{s}_2$ .....	71
3.6	Ensembles flous de la sortie $\Delta G(t)$ .....	71
3.7	Architecture de contrôle dynamique par mode glissant .....	73
3.8	Architecture de contrôle dynamique utilisant le PSO .....	79
3.9	Poursuite d'une trajectoire circulaire .....	82
3.10	Erreurs de poursuite avec et sans PSO .....	82
3.11	Vitesses de commande $v$ et $\omega$ avec et sans PSO.....	83
3.12	Couples $\tau_1$ et $\tau_2$ avec et sans PSO .....	83
3.13	Erreurs de vitesses avec et sans PSO.....	84

3.14	Poursuite d'une trajectoire sinusoïdale .....	85
3.15	Erreurs de poursuite avec et sans PSO .....	85
3.16	Vitesses de commande $v$ et $\omega$ avec et sans PSO.....	86
3.17	Couples $\tau_1$ et $\tau_2$ avec et sans PSO .....	86
3.18	Erreurs de vitesses avec et sans PSO.....	87
3.19	Poursuite d'une trajectoire aléatoire .....	87
3.20	Erreurs de poursuite avec et sans PSO .....	88
3.21	Vitesses de commande $v$ et $\omega$ avec et sans PSO.....	88
3.22	Couples $\tau_1$ et $\tau_2$ avec et sans PSO .....	89
3.23	Erreurs de vitesses avec et sans PSO.....	89
3.24	Evolution de la fonction fitness.....	90
4.1	Cycle de l'algorithme génétique .....	94
4.2	Diagramme de sélection .....	96
4.3	Diagramme de contrôle basé sur les AG .....	104
4.4	Organigramme de l'algorithme génétique .....	105
4.5	Ensembles flous d'entrée $s_1 \dot{s}_1$ .....	108
4.6	Ensembles flous de la sortie $\Delta G(t)$ .....	108
4.7	Poursuite d'une trajectoire circulaire .....	109
4.8	Erreurs de poursuite avec et sans AG.....	109
4.9	Vitesses de commande $v$ et $\omega$ obtenues par contrôle dynamique avec et sans AG....	110
4.10	Couples $\tau_1$ et $\tau_2$ pour les deux cas avec et sans AG.....	110
4.11	Erreurs de vitesses pour les deux cas avec et sans AG.....	111
4.12	Poursuite d'une trajectoire sinusoïdale .....	112
4.13	Erreurs de poursuite utilisant les lois dynamiques avec et sans AG .....	112
4.14	Vitesses de commande $v$ et $\omega$ obtenues par contrôle dynamique avec et sans AG ...	113
4.15	Couples $\tau_1$ et $\tau_2$ .....	113
4.16	Erreurs de vitesses .....	114

4.17	Poursuite d'une trajectoire aléatoire .....	114
4.18	Erreurs de poursuite utilisant les lois dynamiques avec et sans AG .....	115
4.19	Vitesses de commande $v$ et $\omega$ obtenues par contrôle dynamique avec et sans AG ....	115
4.20	Couples $\tau_1$ et $\tau_2$ avec et sans AG.....	116
4.21	Erreurs de vitesses pour les deux cas avec et sans AG.....	116
4.22	Evolution de la fonction fitness.....	117

# LISTE DES ABREVIATIONS

RMR: Robot mobile à roue

SLAM: *Simultaneous localization and mapping*, Localisation et modélisation simultanées de l'environnement

EKF : Extended Kalman Filter, Filtre de Kalman étendu

GPS : *Global positioning system*, Système mondial de positionnement

FLC : *Fuzzy logic controller*, Contrôleur de la logique floue

GA : *Genetic algorithm*, Génétique algorithm

TSM : *Terminal sliding mode*, mode de glissement terminal

TSMC : *Terminal sliding mode controller*, Contrôle du mode glissant terminal

PD : *proportional derivative*, dérivateur proportionnel

PID : *proportional integral derivative*, proportionnel intégral et dérivé

FTSM : *Fast terminal sliding mode* , Mode glissant terminal rapide

GFTSM : *Global fast terminal sliding mode*, Mode de glissement terminal global rapide

FTF : *Fast terminal function*, Fonction terminale rapide

PSO : *particle swarm optimization*, Essaim des particules

PFM : *potential field method*, Méthode de champ potentiel

ORM : *Obstacle restriction method*, Méthode de restriction d'obstacle

UKF : *Unscented kalman filter*, Filtre de Kalman non parfumé

FKL : Filtre de kalman linéaire

SMC : *Sliding mode controller*, Contrôleur de mode glissement

TSK : *Takagi sugeno kang*

Pbest : *personal best*, Meilleur position individuelle

Pbest : *global best*, Meilleur position globale

# **INTRODUCTION GENERALE**



## **INTRODUCTION GENERALE**

La commande des robots mobiles a connu une grande importance ces dernières années, en raison de leur utilisation dans les différents domaines notamment la fabrication industrielle, l'éducation et les applications militaires [1, 2, 3] et de leur intervention dans les milieux où l'être humain ne peut pas agir.

Les robots mobiles à roues non holonomes sont des robots qui ont une architecture simple et peuvent fonctionner dans différents environnements et opérer dans des endroits limités. La maniabilité d'un robot mobile à roues (RMR) dépend des roues et des instructions utilisées.

Il existe de nombreux types de systèmes mécaniques qui nécessitent un contrôle, tels que les robots manipulateurs [4], les manipulateurs parallèles, les robots mobiles multiples et les systèmes multi-agents. Pour chaque type de système mécanique, le contrôle doit être conçu en tenant compte des modèles cinématiques et dynamiques.

Avant de contrôler un système robotique quelconque, un modèle mathématique est obligatoire. De ce fait, la littérature propose deux approches différentes de modélisation qui sont la modélisation cinématique et la modélisation dynamique. Dans un modèle cinématique, seul le mouvement du véhicule est considéré et la commande peut se faire par dérivation du modèle en tenant compte des contraintes non holonomes. Le modèle cinématique est facile à dériver et sa représentation est généralement simple. Cependant, ce modèle cinématique ne tient pas compte de l'accélération, de ce fait son évolution peut être différente du mouvement réel lorsque les limites de maniabilité du véhicule sont atteintes. En raison de sa simplicité, il est largement utilisé pour la conception et l'élaboration de contrôleurs robustes [5, 6, 7].

Par contre, la modélisation dynamique tient généralement compte des caractéristiques du véhicule liées à son accélération, telle que la masse, le centre de gravité, l'adhérence des roues. Ce type de modélisation est plus fidèle à la conduite réelle du véhicule mais la représentation mathématique du système qui en découle est beaucoup plus compliquée [8, 9].

La complexité des tâches à exécuter par le robot dépend largement des performances de son algorithme de localisation (robustesse, précision, etc.) et de la mobilité que lui présente son système de mouvement. Un RMR est alors conçu, généralement, pour exécuter un certain nombre de tâches que l'on peut grouper en fonction du type d'environnement dans lequel le robot évolue (structuré, non-structuré, milieu naturel, etc.), du taux de connaissance de son environnement (objets fixes ou dynamiques) et de sa propriété d'évolution.

La navigation des robots mobiles est essentiellement limitée à leur performance à percevoir et agir avec leur environnement de manière délibérative. La propriété désirable qu'un robot mobile doit posséder est évidemment la compétence obligatoire à reconnaître les repères et les objets qui l'entourent et pouvoir se localiser vis à vis de son espace de travail. Cette connaissance est importante pour la réussite des tâches de navigation intelligente. Cependant pour qu'une telle interaction ait lieu, un modèle ou une description de l'environnement doit être défini [10, 11]. Si la description des éléments présents dans l'environnement est disponible, le problème revient alors à l'interprétation et à l'association des données du capteur sur ces modèles d'objets précédemment enregistrés. En plus, si les objets reconnus sont fixes, ils peuvent être considérés comme des points de repère ainsi qu'ils peuvent être utilisés comme des références pour une auto-localisation.

L'auto-localisation des robots mobiles est une tâche nécessaire pour atteindre une autonomie complète durant la navigation. Plusieurs solutions en robotique sont développées pour résoudre ce problème. Les solutions développées peuvent être classées en deux groupes majeurs: localisation relative et localisation absolue.

Différents problèmes de la localisation des robots mobiles sont étudiés, certains développent un système intégrant une combinaison de capteurs tels que le gyroscope et les capteurs à ultrasons pour déterminer l'emplacement exact du robot mobile. D'autres travaux de recherches se concentrent sur l'utilisation d'un seul amer pour la localisation [12] et d'autres utilisent un ensemble d'amers naturels [13, 14]. L'un des problèmes majeurs en robotique mobile qui combine les différents aléas de la localisation est la localisation et cartographie simultanées (SLAM) [15, 16,17].

Dans ce sens, le filtre de Kalman étendu (FKE) constitue la méthode la plus utilisée à des fins d'estimation en raison de sa simplicité d'implémentation [18]. En effet, plusieurs travaux de recherche utilisent le FKE dans le but d'extraire les entités linéaires des données du



scanner laser [19], alors que certains adoptent les GPS pour la localisation lors de la cartographie de l'environnement. L'algorithme génétique est également utilisé pour fusionner les cartes des différentes grilles d'occupation établies lors du mouvement des véhicules [20]. D'autres travaux s'articulent sur la fusion de capteurs de données radar, ultrasonores et odométriques pour la localisation d'un robot mobile [21].

Pour améliorer le processus d'auto-localisation du robot mobile, le filtre de Kalman étendu est introduit dans ce travail de recherche en se basant sur des amers artificiels. Des résultats de simulation sont présentés pour plusieurs cas d'environnements avec un nombre différent d'amers.

De nombreuses recherches sont effectuées pour contrôler le mouvement d'un robot mobile notamment pour le suivi de trajectoire [22, 23, 24]. Certaines utilisent un modèle perturbé du robot, en tenant compte des incertitudes du modèle [25, 26], d'autres se concentrent sur le modèle cinématique pour concevoir des lois de contrôle fournissant une stabilité asymptotique de tout le système en utilisant la théorie de Lyapunov [27]. D'autres travaux utilisent le modèle dynamique en présence d'incertitudes et de perturbations limitées [28, 29]. Plusieurs travaux utilisent le contrôle par mode glissant pour le suivi de trajectoires et proposent un contrôle robuste [30, 31]. Un autre type de contrôle de suivi de trajectoires pour un robot mobile, utilisant un contrôleur flou (FLC) est proposé dans [32, 33, 34] et une méthode hybride utilisant la logique floue et un algorithme génétique (GA) est proposée dans [35].

La plupart des méthodes de contrôle ne peuvent atteindre que la stabilité asymptotique de suivi de la position, c'est-à-dire en temps infini pour atteindre leur objectif de contrôle. Par contre, les approches de contrôle de la stabilité en temps fini ont un taux de convergence plus rapide avec une grande robustesse vis-à-vis des incertitudes et des perturbations externes du système [36, 37]. Le contrôle par mode glissant terminal (TSMC) est une méthode de contrôle qui a une stabilité dans un temps fini.

Une méthode récente [38, 39] suggère une commande adaptative en mode glissant terminal flou pour les systèmes non linéaires en présence de perturbations externes afin d'obtenir une convergence rapide. Une commande robuste utilisant une surface de glissement PD pour un système non linéaire pour stabiliser le système, gérer l'effet des perturbations et les erreurs de suivi jusqu'à l'origine est proposée dans [40]. Une recherche récente utilise une

nouvelle approche pour résoudre le problème de singularité de la commande par mode glissant terminal pour les systèmes non linéaires [41] et une autre propose une commande adaptative par mode glissant terminal intégral flou pour un robot mobile non holonome [42].

Généralement, les PSO se caractérisent par un concept simple, facile à mettre en œuvre et efficace du point de vue informatique. Contrairement aux autres techniques heuristiques, le PSO dispose d'un mécanisme souple et bien équilibré pour améliorer les capacités d'exploration globale et locale. Dans ce cadre, une nouvelle approche basée sur les PSO pour la conception optimale d'un contrôleur PID pour le suivi de trajectoire de robot mobile est proposée dans [43].

Notre objectif dans ce travail vise essentiellement les différents aspects qui n'ont pas été abordés et traités dans la littérature, en orientant notre méthodologie sur les axes majeurs suivants :

(1) La mise en œuvre d'un modèle de contrôle qui doit permettre l'introduction de nouvelles stratégies de commande pour le secteur de la robotique. Pour cela on propose de nouvelles solutions en utilisant des techniques de commande basées sur les modes de glissement et la logique floue.

(2) L'élaboration de contrôleurs robustes en utilisant des outils de l'intelligence artificielle pour l'optimisation des paramètres impliqués dans le processus de commande entre autre le PSO et les algorithmes génétiques. Plusieurs méthodes, utilisant l'algorithme PSO et les algorithmes génétique (GA) pour l'optimisation des paramètres de la commande sont développées pour permettre à ces systèmes d'évoluer d'une manière autonome et sûre dans leur milieu de travail, à accoster convenablement sans risque de collision, d'éviter des dérives très importantes et d'arriver convenablement au but tout en tenant compte des contraintes de non holonomie. Des travaux de simulation sont réalisés pour éventuellement vérifier la portée des algorithmes de commande qui sont élaborés durant notre travail de recherche.

## **PROBLEMATIQUE**

Un robot autonome introduit dans un environnement inconnu doit être capable de percevoir et d'identifier les principales composantes de cet environnement pour pouvoir se localiser et éventuellement suivre des trajectoires prédéfinies. Cependant la problématique principale reste la conception d'un système de commande permettant au robot d'évoluer

d'une manière sûre dans un environnement perturbé. Une telle commande nécessite le développement de différentes stratégies, selon les objectifs assignés et les contraintes des tâches à exécuter.

## **PLAN DE LA THESE**

Le travail de recherche que nous présentons vise principalement la localisation et le suivi de trajectoires d'un robot mobile nonholonome. Les algorithmes présentés dans ce travail de recherche traitent convenablement des problèmes survenant lors du mouvement d'un robot mobile autonome et de la robustesse visée à vis des perturbations et des incertitudes. Pour appréhender cette problématique, nous avons structuré notre travail en quatre chapitres:

Le premier chapitre traite des différents types de navigation d'un robot mobile ainsi que des modélisations cinématique et dynamique. De ce fait une présentation de ces différents modèles est donnée dans ce chapitre, dans le but d'appliquer ultérieurement les commandes proposées.

Le deuxième chapitre est consacré à la description des différents types de localisation existants ainsi qu'aux différents algorithmes de localisation proposés dans la littérature. Un algorithme de localisation utilisant le filtre de Kalman étendu (FKE) basé sur des amers artificiels est présenté dans le but d'estimer la position du robot dans son environnement d'évolution. Des travaux de simulation sont entrepris et présentés en utilisant différents cas d'environnements.

Le troisième chapitre reflète une partie de notre contribution dans cette thèse; on a commencé par une description des méthodes du mode glissant terminal rapide (FTSM) et terminal rapide global (GFTSM). Des généralités sur la logique floue sont présentées afin de proposer un contrôleur cinématique basé sur le mode glissant flou appliqué sur un modèle cinématique perturbé du robot mobile. Par la suite un contrôleur dynamique basé sur le mode glissant exponentiel est appliqué sur le modèle dynamique perturbé. Une optimisation par essaim de particules (PSO) est appliquée sur la loi de commande proposée pour en extraire les paramètres optimaux. Une étude de stabilité est faite en utilisant la méthode de Lyapunov. Des travaux de simulation sont illustrés pour démontrer la robustesse des algorithmes proposés.

Le quatrième chapitre est consacré à la proposition d'une commande robuste appliquée au suivi de trajectoires d'un robot mobile nonholonome. Une description sur le mode de glissement terminal rapide (FTF) ainsi que des généralités sur les algorithmes génétiques (GA) sont présentées en détail. Un contrôleur cinématique est premièrement proposé et appliqué sur le modèle cinématique du robot; ce contrôleur, basé sur le mode de glissement et la logique floue, est proposé dans le but de faire converger l'erreur de posture du robot vers zéro. Une étude de la stabilité du robot est faite pour assurer la convergence asymptotique des erreurs de suivi. Dans le même contexte, un contrôleur dynamique est proposé en se basant sur le mode de glissement terminal rapide (FTF) afin d'implémenter les algorithmes génétiques dans le but d'avoir des paramètres optimaux de la surface de glissement choisie. Des travaux de simulation sont entrepris avec et sans utilisation des algorithmes GA et une analyse comparative entre les algorithmes proposés est présentée.

Finalement, on termine notre travail par une conclusion générale.

# **1<sup>ER</sup> CHAPITRE**

## **APERCUS SUR LA ROBOTIQUE MOBILE**

---

# **CHAPITRE 1**

## **Aperçus sur la robotique mobile**

### **1.1 Introduction**

Dans ce chapitre, on traite d'une manière générale les tâches les plus intéressantes des robots mobile à roues notamment la perception qui est généralement une tâche de compression de données en termes d'extraction des caractéristiques qui permettent la reconnaissance d'environnements et la description de propriétés particulières qui sont nécessaires pour permettre à un robot d'exécuter ses tâches. Ensuite, une étude générale sur les stratégies de la navigation est présentée avec différentes approches.

L'évitement d'obstacle est une compétence qui est aussi importante en robotique mobile, il facilite la modulation de la trajectoire du robot et permet d'éviter d'une manière générale les collisions. Une variété d'approches est présentée dans ce chapitre afin de démontrer l'importance de l'évitement d'obstacles.

Dans la dernière partie du chapitre, on s'intéresse d'une manière détaillée aux différents types de modélisation. Ceci permet de caractériser la mobilité des robots mobiles afin d'introduire les modèles cinématiques nécessaires à la conception des lois de contrôle. Les modèles dynamiques du robot unicycle et différentiel sont exposés en tenant compte respectivement des vitesses et des couples engendrés par les actionneurs comme entrées du système étudié.

### **1.2 Robots mobiles**

Les robots mobiles sont des robots qui peuvent se déplacer de manière autonome, c'est-à-dire sans l'aide d'opérateurs humains externes. Contrairement à la majorité des robots industriels qui ne peuvent se déplacer que dans un espace de travail spécifique, les robots mobiles ont la particularité de se déplacer librement dans un espace de travail prédéfini pour atteindre leurs objectifs souhaités.

Il existe une grande variété de robots mobiles qui peuvent être regroupés en deux types essentiels, les robots mobile à jambes [44] et les robots mobiles à roues [45,46]. Les deux catégories sont illustrées dans les figures (1.1) et (1.2) respectivement.

Les robots mobiles à roues sont utilisés dans de nombreuses applications, ils consomment moins d'énergie et se déplacent plus rapidement que d'autres mécanismes de locomotion (par exemple, les robots à jambes ou véhicules à chenilles). Du point de vue du contrôle, moins d'effort est nécessaire en raison de leurs mécanismes simples et les problèmes de stabilité sont réduits. Les robots mobiles à roues conviennent à une grande classe d'environnements cibles dans des applications pratiques.



Figure 1.1 : Exemple de robot à jambes [47]

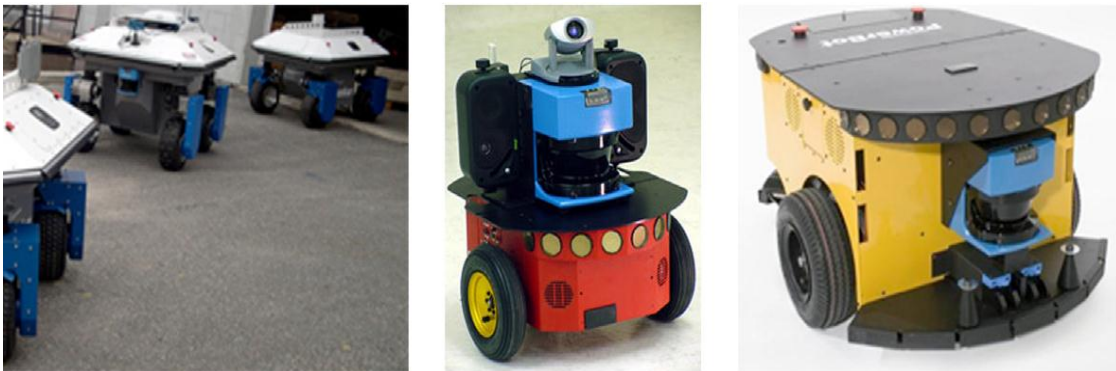


Figure 1.2 : Exemple de robot à roue [48]

### 1.3 La perception

La perception en robotique mobile est l'une des tâches les plus importantes d'un système autonome, elle permet d'acquérir des connaissances sur l'environnement. Ceci est réalisé en prenant des mesures par utilisation de divers capteurs, les informations significatives sont extraites à partir de ces mesures.

La perception dépend de la capacité du robot à manipuler et à mettre en forme les informations qui sont pertinentes pour pouvoir évoluer convenablement dans l'environnement qui l'entoure [49,50]. Aussi, dans le but de tirer les informations pertinentes pour l'accomplissement de tâches, il est indispensable que le robot soit doté de capteurs qui observent d'une manière continue son état interne et l'environnement. Le choix des capteurs est relatif à l'application considérée. La perception est essentielle pour la sécurité du robot, la modélisation de l'environnement et l'évitement d'obstacles. Les moyens utilisés dans la perception de l'environnement sont très nombreux, on peut citer les télémètres laser et ultrasonores, les capteurs optiques et infrarouges, les capteurs tactiles et les systèmes de vision.

Pour structurer le processus de perception, un modèle général comme indiqué dans la figure (1.3) est adopté.

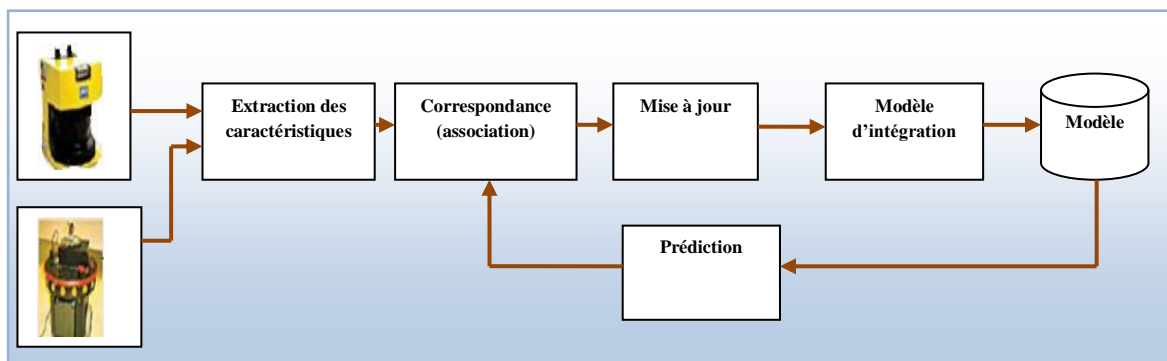


Figure 1.3 : Exemple d'un processus de perception [51]

Dans ce modèle sont incluses les opérations courantes appliquées pour intégrer les données du capteur avec le modèle du monde. En fonction de la tâche en question, certains des modules inclus peuvent être embarrassants et d'autres peuvent eux-mêmes prendre une structure compliquée. Cependant, le modèle fourni suffit à illustrer bon nombre de problèmes de détection et d'estimation.



## 1.4 Les méthodes de navigation

Les méthodes de navigation sont généralement conçues pour permettre au robot mobile d'avancer vers un but défini. Nous examinons dans cette partie une classification proposée par Trullier et al. [52, 53]. Cette méthode de classification possède plusieurs types, nous allons décrire en détail ces types dans ce qui suit, de l'approche simple à l'approche complexe.

### 1.4.1 Approche d'un objet

Cette approche comporte une performance d'avancer vers un objet visible depuis la position actuelle du robot. Elle est basée essentiellement sur une remontée de gradient de la perception de l'objet, cette technique est peut être vue dans des véhicules de Valentino Braitenberg [54] qui placent deux capteurs de lumière pour avoir une source lumineuse. Cette approche est axée sur les actions réflexes dont dans la perception est lie à une action, cette stratégie est dite locale par son fonctionnement dans des cibles visibles.

### 1.4.2 Guidage

Cette approche est différente de celle présentée précédemment. Elle permet d'atteindre une cible qui n'est pas un objet directement visible, mais peut être un point de l'espace qui a des caractéristiques d'une configuration d'un ensemble d'objets capturés, ou amers, qui l'entourent. Cette méthode de navigation d'un robot est souvent une descente de gradient qui permette aussi de se diriger dans le sens de cette configuration. Cette approche peut être utilisée par plusieurs robots et est inspirée de la nature des insectes, notamment les abeilles [55, 56].

### 1.4.3 Action associée à un lieu

Cette approche est la première qui réalise une navigation globale, en effet elle permette d'atteindre une cible depuis des positions et cette cible qui est peut-être un amer est considérée invisible [56]. Elle nécessite une représentation interne de l'environnement qui permet de remplacer les lieux comme des zones de l'espace, par conséquent les perceptions restent du même ordre pour une éventuelle action à réaliser (figure 1.4).

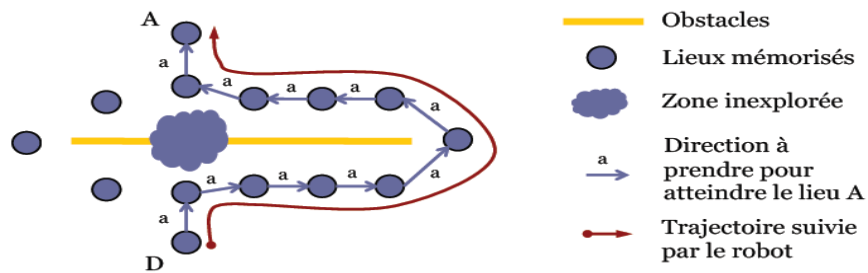


Figure 1.4 : Action associée à un lieu [57]

L'enchaînement de ces actions définit un chemin qui permet d'atteindre une cible. Ces modèles assurent donc une autonomie plus intéressante du robot mais sont bornés à une cible fixée. Un chemin qui permet d'atteindre une cible ne pourra jamais être utilisé pour rejoindre une autre cible différente. Par conséquent, changer une cible entraînera l'apprentissage d'un nouveau chemin.

#### 1.4.4 Navigation topologique

Cette approche est une prolongation de la méthode précédente. Elle permet de stocker dans le modèle interne les liaisons spatiales entre les lieux comme présenté dans la figure (1.5). Ces liaisons montrent la possibilité d'avancer d'un lieu à un autre, en effet elles ne sont pas associées à une cible particulière. Le modèle interne de cette approche consiste à calculer différents chemins entre deux lieux aléatoires, alors que ce modèle ne peut pas planifier les déplacements du robot entre les lieux connus.

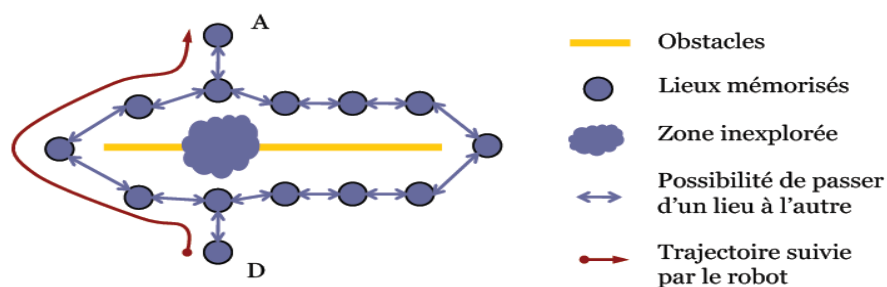


Figure 1.5 : Navigation topologique [57]

#### 1.4.5 Navigation métrique

Cette approche est considérée comme une extension de l'approche précédente. Elle permet au robot de planifier des trajectoires dans des zones inexplorées de son espace. Elle

peut stocker les positions métriques relatives des différents lieux et permet d'avoir la possibilité de se déplacer d'un lieu à l'autre. Les positions dans cette approche consistent à calculer un chemin commençant d'un lieu à un autre, cependant cette possibilité de déplacement n'est pas stockée ou mémorisée comme un lien (figure 1.6). Cette technique permet de définir le chemin optimal entre deux lieux mémorisés.

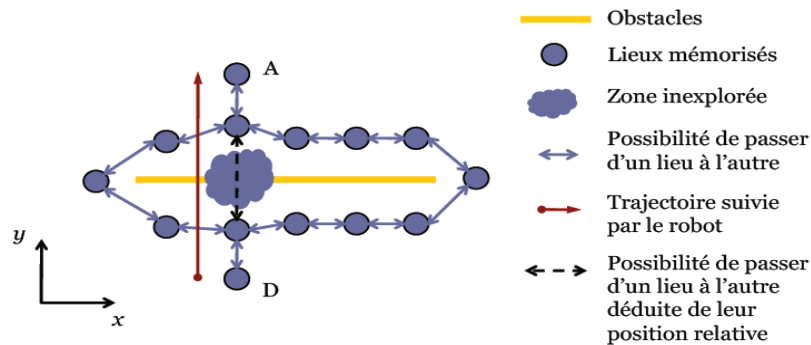


Figure 1.6 : Navigation métrique [57].

#### 1.4.6 Navigation par reconstruction intérieure

L'algorithme du SLAM utilisant des télémètres laser 2D peut reconstruire des plans d'étage avec une précision centimétrique. Certaines recherches étendent ce travail à la reconstruction 3D, utilisant des lasers 2-D balayant le déplacement du robot. Le nuage de points résultant est généralement enregistré en utilisant la position du robot corrigée par l'algorithme SLAM 2D [58]. Les points bruts peuvent être présentés comme une image 3D, ou transformés pour donner une reconstruction plane ou maillée des surfaces intérieures. Ce dernier est particulièrement convaincant lorsque les images de la caméra sont texturées sur les surfaces, pour créer un modèle 3D réaliste.



Figure 1.7 : Carte d'intérieur 3D à partir d'un plan vertical balayé LADAR.

La figure (1.7) montre un résultat de la cartographie intérieure utilisant cette technique. L'enregistrement se fait à partir d'une LADAR horizontale utilisant algorithmes SLAM.

## 1.5 Méthodes d'évitement d'obstacles

Nous décrivons ici une présentation des techniques d'évitement d'obstacles et quelques méthodes représentatives. Les méthodes heuristiques sont les premières techniques utilisées pour générer un mouvement basé sur des capteurs [59]. Les méthodes d'analogies physiques assimilent les évitements d'obstacles à un problème physique connu. Nous discutons ici des méthodes du champ de potentiel.

### 1.5.1 Méthode du champ de potentiel (PFM)

La méthode du champ de potentiel (PFM) utilise une analogie dont le robot est une particule qui se déplace dans la configuration d'espace sous l'influence d'un champ de force. L'emplacement cible exerce une force  $F_{att}$  qui attire la particule et les obstacles exercent des forces répulsives  $F_{rep}$  (figure 1.8).

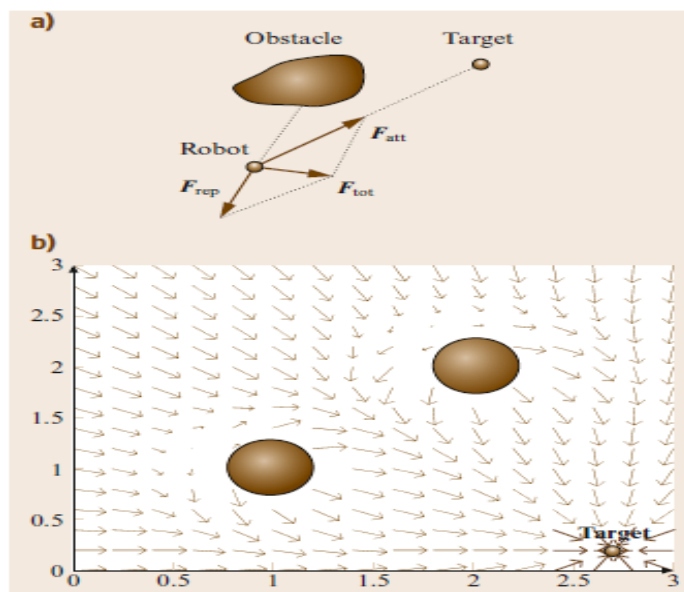


Figure 1.8 : (a) Sens résultant du déplacement.

(b) Lignes du champ de potentiel [51].

À chaque temps  $t_i$ , le mouvement est calculé pour suivre la direction de la force artificielle induite par la somme des deux potentiels  $F_{tot}(q_{ti}) = F_{att}(q_{ti}) + F_{rep}(q_{ti})$  (la direction la plus prometteuse du mouvement).

$$F_{att}(q_{t_i}) = K_{att} nq_{Target} \quad (1.1)$$

$$F_{rep}(q_{t_i}) = \begin{cases} K_{rep} \sum_j \left( \frac{1}{d(q_{t_i}, p_j)} - \frac{1}{d_0} \right) n_{p_j}, & \text{si } d(q_{t_i}, p_j) < d_0 \\ 0, & \text{ailleurs} \end{cases} \quad (1.2)$$

Où  $K_{att}$  et  $K_{rep}$  sont les constantes des forces,  $d_0$  est la distance d'influence des obstacles  $p_j$ ,  $q_{ti}$  est la configuration actuelle du véhicule,  $nq_{target}$  et  $n_{jp}$  sont les vecteurs unitaires qui pointent de  $q_{ti}$  à la cible de chaque obstacle  $p_j$ , respectivement. De  $F_{tot}(q_{ti})$ , le contrôle  $u_i$  peut être obtenu avec un contrôle de position ou de force [51]. Ceci est la version classique où les potentiels seulement dépendent de la configuration actuelle du véhicule. Complémentairement, les potentiels généralisés dépendent également de la vitesse instantanée du robot et l'accélération du véhicule.

### 1.5.2 Méthode de restriction d'obstacle (ORM)

L'ORM résout le problème en trois étapes. Le résultat des deux premières étapes est un ensemble de candidats de mouvement de direction. La première étape consiste à calculer une valeur instantanée et la deuxième étape associe une contrainte de mouvement à chaque obstacle et se joint ensuite pour calculer l'ensemble des directions souhaitables. La dernière étape est une stratégie pour calculer le mouvement, étant donné cet ensemble.

- **La sélection de cible instantanée :**

Cette étape calcule un but quand il est préférable de diriger le mouvement vers une zone donnée de l'espace (qui améliore la situation pour atteindre le dernier objectif), plutôt que directement vers le but lui-même. Les sous-objectifs sont situés entre les obstacles ou au bord d'un obstacle (figure 1.9 a). Ensuite, le processus vérifie en utilisant un algorithme local si l'objectif peut être atteint de l'emplacement du robot. Si ce n'est pas le cas, alors l'objectif le plus accessible est sélectionné. Pour vérifier si un point peut être atteint, il existe un algorithme local qui calcule l'existence d'un chemin local qui joint deux emplacements [60]:

Soit  $x_a$  et  $x_b$  deux emplacements de l'espace,  $R$  le rayon du robot, et  $L_p$  une liste de points d'obstacles, où  $p_i$  est un obstacle de la liste. Soit  $A$  et  $B$  deux demi plans divisés par la ligne qui joint  $x_a$  et  $x_b$ . Alors, si pour tous les points de  $L_p$ ,  $d(p_j, p_k) > 2R$  (avec  $p_j \in A$  et  $p_k \in B$ ), alors il existe un chemin sans collision qui relie les deux emplacements. Si cette condition n'est pas remplie, alors il y a pas de chemin local, bien qu'un chemin global puisse exister.

▪ **Ensemble de directives du candidat :**

Pour chaque obstacle  $i$ , un ensemble de directions non souhaitables de mouvement  $S_{nD}^i$  est calculée (contrainte de mouvement). Cet ensemble est l'union de deux sous-ensembles:  $S_1^i$  et  $S_2^i$ ,  $S_1^i$  représente le côté de l'obstacle non apte à éviter et  $S_2^i$  est une zone d'exclusion autour de l'obstacle (figure 1.9 b). La contrainte de mouvement pour l'obstacle est l'union de ces deux ensembles  $S_{nD}^i = S_1^i \cup S_2^i$ . L'ensemble des directions de mouvement souhaitées sont l'ensemble complémentaire  $S_D = \{[-\pi, \pi] / d \& nD\}$ , ou  $S_{nD} = \cup_i S_{nD}^i$ .

▪ **Calcul de mouvement :**

La dernière étape consiste à sélectionner une direction de mouvement. Cela se fait selon trois cas, en fonction de l'ensemble des orientations souhaitables  $S_D$  et de la direction de la cible  $\theta_{\text{target}}$  [51].

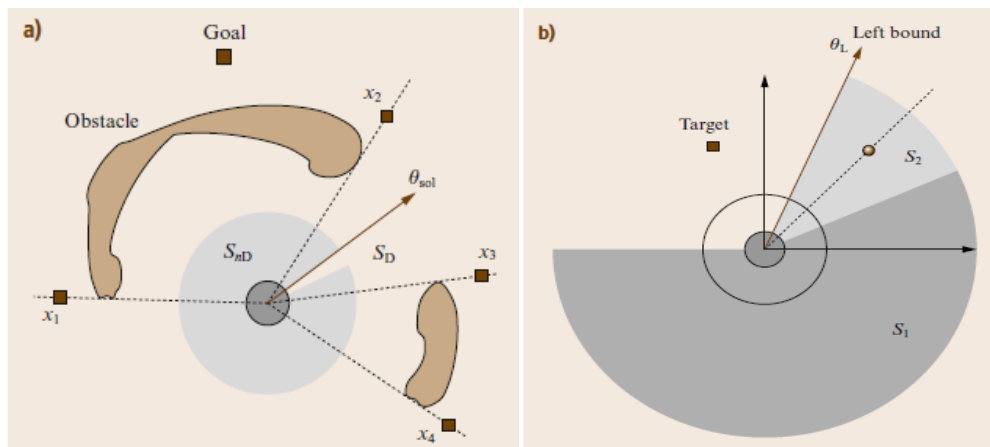


Figure 1.9 : (a) Répartition des sous-objectifs  $x_i$ . (b) Les deux ensembles de directions indésirables  $S_1$  et  $S_2$  pour un obstacle donné [51].

### 1.5.3 Approches par fenêtre dynamique

Une autre technique permettant de prendre en compte les contraintes cinématiques des robots est la méthode par fenêtre dynamique. Un modèle dynamique simple mais très efficace donne son nom à cette approche (figure 1.10). Deux de ces approches sont discutées dans la littérature. L'approche de la fenêtre dynamique [61] de Fox, Burgard et Thrun et l'approche de la fenêtre dynamique globale [62] de Brock et Khatib.

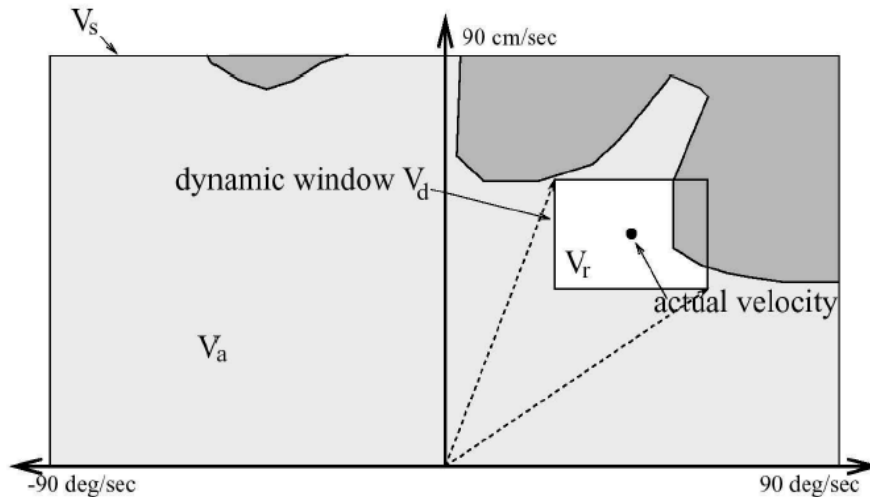


Figure 1.10 : L'approche dynamique de la fenêtre [61].

La fenêtre rectangulaire indique les vitesses possibles  $V$  et le chevauchement avec des obstacles dans l'espace de configuration.

### 1.5.4 Autres approches

Les approches décrites ci-dessus font partie des systèmes d'évitement d'obstacles les plus couramment utilisés. Il existe cependant de nombreuses techniques supplémentaires permettant d'éviter les obstacles dans la communauté de la robotique mobile. Par exemple, des travaux fournissant un aperçu des approches floues et neuro-floues pour l'évitement d'obstacles sont présentés dans [63]. Inspirés par la nature, une approche biologique reproduisant le réseau neuronal d'une blatte est présentée dans [64]. Le réseau est ensuite appliqué à un modèle de véhicule à quatre roues. Tous les obstacles sont définis comme des anticibles avec une position exacte et une forme circulaire. Les anticibles sont ensuite utilisées pour créer les lois de contrôle du système [65].

## 1.6 Types de robots mobiles et modélisation

Il existe de nombreuses alternatives de conception de robots mobiles à roues. Les problèmes de conception comprennent le choix des roues et leurs placements ainsi que la détermination des paramètres cinématiques. Dans cette section, on présente deux axes essentiels de locomotion d'un robot mobile : La modélisation cinématique et la modélisation dynamique.

### 1.6.1 Modélisation cinématique des robots mobiles

Le modèle cinématique de posture est le modèle le plus simple de l'espace d'état qui est capable de donner une description globale du robot. Il permet de décrire le comportement cinématique de toutes les variables de configuration.

#### 1.6.1.1 Robot unicycle

Un robot unicycle est un robot qui est piloté par deux roues indépendantes ayant éventuellement des roues folles et libres pour assurer sa stabilité, ce type de robot est très utilisé en raison de sa simplicité et de ses propriétés cinématiques. Son modèle cinématique est représenté par [66]:

$$\begin{cases} \dot{x} = v \cdot \cos \theta \\ \dot{y} = v \cdot \sin \theta \\ \dot{\theta} = \omega \end{cases} \quad (1.3)$$

Avec  $(x, y)$  les coordonnées cartésiennes du milieu du robot et  $\theta$  l'orientation par rapport à l'axe  $x$  (figure 1.11).

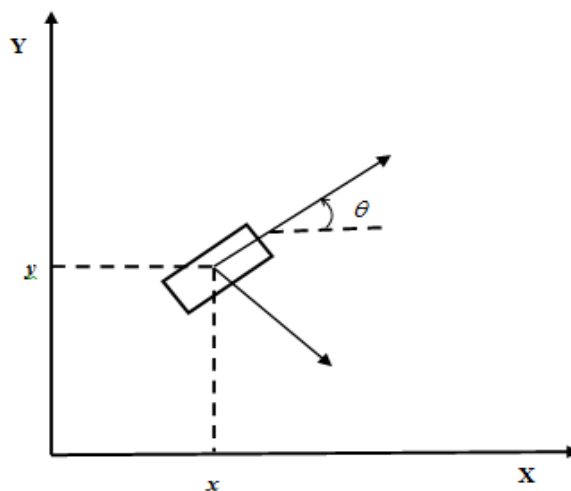


Figure 1.11 : Robot unicycle



En effet, par l'élimination de la vitesse linéaire du robot  $v$ , on obtient la contrainte nonholonomique :

$$-x \sin\theta + y \cos\theta = 0 \tag{1.4}$$

Le modèle cinématique (1.3) peut être présenté comme :

$$\dot{p} = \begin{bmatrix} \cos\theta \\ \sin\theta \\ 0 \end{bmatrix} v + \begin{bmatrix} 0 \\ 0 \\ 0 \end{bmatrix} \omega, \quad \dot{p} = \begin{bmatrix} \dot{x} \\ \dot{y} \\ \dot{\theta} \end{bmatrix} \tag{1.5}$$

Une autre représentation de (1.5) :

$$\dot{p} = J\dot{q}, \quad q = \begin{bmatrix} v \\ \omega \end{bmatrix}^T \tag{1.6}$$

Avec  $J$  représentant la matrice jacobéenne du système.

$$J = \begin{bmatrix} \cos\theta & 0 \\ \sin\theta & 0 \\ 0 & 1 \end{bmatrix} \tag{1.7}$$

Le modèle (1.5) est considéré parmi les systèmes non linéaires, appelé généralement les systèmes affines, il peut être représenté sous forme:

$$\begin{aligned} \dot{x} &= f_0(x) + \sum_{i=1}^m f_i(x)u_i \\ &= f_0(x) + F(x)u \end{aligned} \tag{1.8}$$

Avec,  $u_i = (1, \dots, m)$  et  $x = [x_1, x_2, \dots, x_n]^T \in X$ ,  $u = [u_1, u_2, \dots, u_m]^T \in U$

$$F(x) = [f_1(x), f_2(x), \dots, f_m(x)] \in X \tag{1.9}$$

Avec  $x$  le vecteur d'état du système et  $u$  le vecteur de contrôle.

Si  $m < n$  le système a moins de variables de contrôles que les degrés de liberté par contre si  $m > n$  le système devient suractif. En réalité, on prend généralement  $m < n$ .

Les colonnes de matrice  $F(x)$  peuvent être écrites :

$$f_1(x) = \begin{bmatrix} f_{11}(x) \\ f_{12}(x) \\ \cdot \\ \cdot \\ f_{1n}(x) \end{bmatrix}, \quad f_2(x) = \begin{bmatrix} f_{21}(x) \\ f_{22}(x) \\ \cdot \\ \cdot \\ f_{2n}(x) \end{bmatrix}, \dots, \quad f_m(x) = \begin{bmatrix} f_{m1}(x) \\ f_{m2}(x) \\ \cdot \\ \cdot \\ f_{mn}(x) \end{bmatrix} \tag{1.10}$$

La formulation (1.10) est appelée les champs vectoriels du système.

Le modèle cinématique du robot unicycle (1.5) est un système affine à deux entrées de champs de vecteur:

$$f_1(\theta) = \begin{bmatrix} \cos \theta \\ \sin \theta \\ 0 \end{bmatrix}, f_2(\theta) = \begin{bmatrix} 0 \\ 0 \\ 1 \end{bmatrix} \quad (1.11)$$

La matrice jacobéenne (1.7) arrange les champs de vecteurs  $f_1, f_2$  dans  $J=F$ . le champ de vecteur  $f_1(\theta)$  permet un mouvement de translation du robot et  $f_2(\theta)$  un mouvement de rotation.

### 1.6.1.2 Robot tricycle

Ce type du robot est équipé par deux roues fixes de même axe ainsi qu'une roue orientable situé sur l'axe longitudinal du robot. Le mouvement de ce robot est contrôlé par la vitesse linéaire  $v$  de la direction de la roue et la vitesse angulaire  $\dot{\phi}$  (figure 1.12).

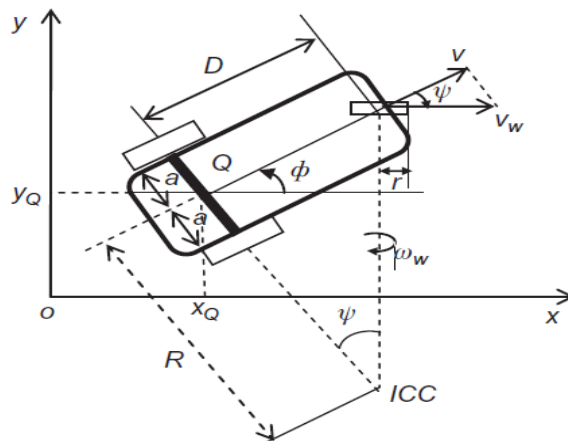


Figure 1.12 : Robot tricycle [67]

L'angle d'orientation et la vitesse angulaire sont respectivement  $\phi$  et  $\dot{\phi}$ . L'état du mouvement du robot est représenté par:

$$p = \begin{bmatrix} x_Q \\ y_Q \\ \phi \\ \psi \end{bmatrix} \quad (1.12)$$

Les variables cinématiques du robot sont :

La vitesse linéaire de la roue orientable  $v_w = r\dot{\theta}_d$

La vitesse linéaire du robot  $v = v_w \cos \psi = r \cos \psi \dot{\theta}_w$

La vitesse angulaire du robot  $\dot{\phi} = \frac{1}{D} v_w \sin \psi$

La vitesse angulaire de la roue orientable  $\dot{\psi} = \omega_\psi$

Par utilisation des relations précédentes, on obtient le système suivant :

$$\dot{p} = \begin{bmatrix} \dot{x}_O \\ \dot{y}_O \\ \dot{\phi} \\ \dot{\psi} \end{bmatrix} = \begin{bmatrix} v \cos \phi \\ v \sin \phi \\ \dot{\phi} \\ \dot{\psi} \end{bmatrix} = \begin{bmatrix} r \cos \psi \cos \phi \\ r \cos \psi \sin \phi \\ \frac{r}{D} \sin \psi \\ 0 \end{bmatrix} \dot{\theta}_d + \begin{bmatrix} 0 \\ 0 \\ 0 \\ 1 \end{bmatrix} \dot{\psi} = F \dot{\theta} \quad (1.13)$$

Avec  $\dot{\theta} = [\dot{\theta}_w \quad \dot{\psi}]^T$  représente le vecteur de contrôle et  $F$  la matrice jacobéenne.

$$F = \begin{bmatrix} r \cos \psi \cos \phi & 0 \\ r \cos \psi \sin \phi & 0 \\ \frac{r}{D} \sin \psi & 0 \\ 0 & 1 \end{bmatrix} \quad (1.14)$$

La matrice jacobéenne  $F$  n'est pas inversible, mais avec une autre manière on peut trouver le modèle inverse tout en utilisant les relations suivantes:

$$\frac{\dot{\phi}}{v} = \frac{1}{D} \tan \psi \quad (1.15)$$

$$\dot{\theta}_w = \frac{v_d}{r} = \frac{1}{r} \sqrt{v^2 + (D\dot{\phi})^2} \quad (1.16)$$

Le centre instantané de rotation  $\rho$  est donné par :

$$\rho = D \tan\left(\frac{\pi}{2} - \psi(t)\right) \quad (1.17)$$

A partir de l'équation (1.13), on peut également remarquer que le modèle du robot tricycle est un système affine à deux entrées avec des champs de vecteur qui sont donnés par:

$$f_1 = \begin{bmatrix} r \cos \psi \cos \phi \\ r \cos \psi \sin \phi \\ \frac{r}{D} \sin \psi \\ 0 \end{bmatrix}, f_2 = \begin{bmatrix} 0 \\ 0 \\ 0 \\ 1 \end{bmatrix} \quad (1.18)$$

Les deux champs de vecteur permettent au robot de faire un mouvement de rotation de la roue orientable.

### 1.6.1.3 Robot de type voiture

Le robot type voiture est équivalent au robot tricycle, mais il suffit de placer la roue orientable du tricycle au centre de l'axe des roues avant de la voiture.

La géométrie du robot mobile de type voiture est illustrée sur la figure (1.13).

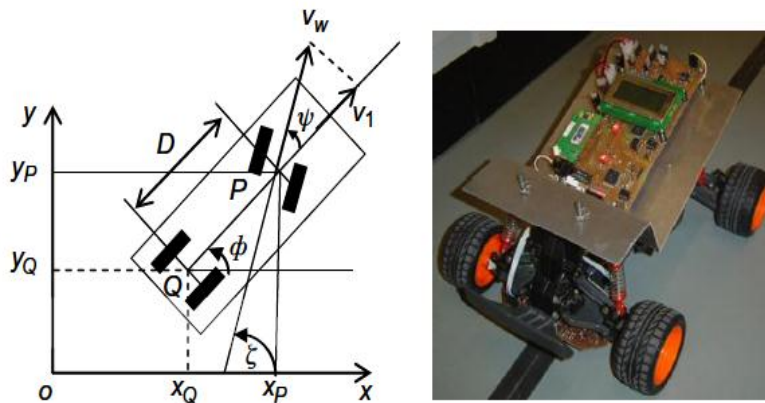


Figure 1.13 : Robot type voiture [68]

L'état du mouvement du robot est représenté par le vecteur [69]:

$$P = \begin{bmatrix} x_Q \\ y_Q \\ \phi \\ \psi \end{bmatrix} \quad (1.19)$$

où  $x_Q, y_Q$  sont les coordonnées cartésiennes du point milieu de l'axe des roues,  $\phi$  est l'angle d'orientation du véhicule et  $\psi$  angle est l'angle de braquage. Dans ce cas, il existe des contraintes non holonomes pour chaque roue, c'est-à-dire:

$$-\dot{x}_Q \sin \phi + \dot{y}_Q \cos \phi = 0 \quad (1.20)$$

$$-\dot{x}_P \sin(\phi + \psi) + \dot{y}_P \cos(\phi + \psi) = 0 \quad (1.21)$$

Où  $x_P$  et  $y_P$  sont les coordonnées du point de milieu des roues avant. On obtient d'après la figure (1.11):

$$x_P = x_Q + D \cos \phi, y_P = y_Q + D \sin \phi \quad (1.22)$$

En utilisant la contrainte cinématique (1.21) et l'équation (1.22) on obtient:

$$-\dot{x}_Q \sin(\phi + \psi) + \dot{y}_Q \cos(\phi + \psi) + D(\cos \psi)\dot{\phi} \quad (1.23)$$

Une forme matricielle des contraintes non holonome peut s'écrire comme suit :

$$M(p)p = 0 \quad (1.24)$$

Avec :

$$M(p) = \begin{bmatrix} -\sin \phi & \cos \phi & 0 & 0 \\ -\sin(\phi + \psi) & \cos(\phi + \psi) & D \cos \psi & 0 \end{bmatrix} \quad (1.25)$$

Les équations cinématiques pour une voiture à propulsion arrière se trouvent être

$$\begin{aligned} \dot{x}_Q &= v_1 \cos \phi \\ \dot{y}_Q &= v_1 \sin \phi \\ \dot{\phi} &= \frac{1}{D} v_w \sin \psi = \frac{1}{D} v_1 \operatorname{tg} \psi \\ \dot{\psi} &= v_2 \end{aligned} \quad (1.26)$$

Ces équations peuvent être écrites sous la forme affine:

$$\begin{bmatrix} \dot{x}_Q \\ \dot{y}_Q \\ \dot{\phi} \\ \dot{\psi} \end{bmatrix} = \begin{bmatrix} \cos \phi \\ \sin \phi \\ (1/D) \operatorname{tg} \psi \\ 0 \end{bmatrix} v_1 + \begin{bmatrix} 0 \\ 0 \\ 0 \\ 1 \end{bmatrix} v_2 \quad (1.27)$$

Les champs de vecteur de cette représentation d'état sont donnés comme suit :

$$g_1 = \begin{bmatrix} \cos \phi \\ \sin \phi \\ (1/D) \operatorname{tg} \psi \\ 0 \end{bmatrix}, \quad g_2 = \begin{bmatrix} 0 \\ 0 \\ 0 \\ 1 \end{bmatrix} \quad (1.28)$$

Les champs de vecteurs  $g_1, g_2$  permettent un mouvement de conduite et de direction, respectivement. La forme jacobéenne de (1.27) est donnée par :

$$\dot{p} = Jv \quad (1.29)$$

Avec  $v = (v_1 \ v_2)^T$ .

$$J = \begin{bmatrix} \cos \phi & 0 \\ \sin \phi & 0 \\ (\operatorname{tg} \psi)/D & 0 \\ 0 & 1 \end{bmatrix} \quad (1.30)$$

Par utilisation des nouvelles entrées :

$$u_1 = v_1, u_2 = (1/D) \sin(\zeta - \phi) v_1 + v_2$$

On obtient le modèle suivant :

$$\begin{bmatrix} \dot{x}_p \\ \dot{y}_p \\ \dot{\phi} \\ \dot{\psi} \end{bmatrix} = \begin{bmatrix} \cos \zeta \\ \sin \zeta \\ (1/D) \sin(\zeta - \phi) \\ 0 \end{bmatrix} u_1 + \begin{bmatrix} 0 \\ 0 \\ 0 \\ 1 \end{bmatrix} u_2 \quad (1.31)$$

où  $\zeta = \phi + \psi$  est l'angle de braquage.

### 1.6.2 Modélisation Dynamiques des robots mobiles

Plusieurs travaux ont considéré la modélisation dynamique du robot comme une partie intéressante pour établir les différentes forces qui affectent son mouvement ainsi qu'elle permet de faire l'analyse et le contrôle [70, 71,72]. La modélisation dynamique est généralement une étude du mouvement du système dans lequel les forces sont modélisées en tenant compte des énergies et des vitesses du mouvement.

Dans cette partie on s'intéresse à la modélisation dynamique du robot différentiel et de l'unicycle. Le modèle dynamique du robot mobile à roue non-holonomes, ayant une dimension  $n$  de coordonnées généralisées  $\dot{q} \in \mathfrak{R}^{n \times 1}$  et soumis à  $m$  contraintes est donné par [73,74] :

$$M(q)\ddot{q} + V(q, \dot{q}) + F(\dot{q}) + G(q) + \tau_d = B(q)\tau - H^T(q)\lambda \quad (1.32)$$

$\dot{q} \in \mathfrak{R}^{n \times 1}$  est le vecteur de la vitesse.

$\ddot{q} \in \mathfrak{R}^{n \times 1}$  est le vecteur de l'accélération.

$M(q) \in \mathfrak{R}^{n \times n}$  est une matrice d'inertie définie symétrique positive.

$V(q, \dot{q}) \in \mathfrak{R}^{n \times n}$  est la matrice centripète et Coriolis.

$F(\dot{q}) \in \mathfrak{R}^{n \times 1}$  représente le frottement de la surface.

$G(q) \in \mathfrak{R}^{n \times 1}$  désigne le vecteur gravitationnel.

$\tau_d \in \mathfrak{R}^{n \times 1}$  est une perturbation inconnue bornée contenant des dynamiques non structurées du modèle.

$B(q) \in \mathfrak{R}^{n \times (n-m)}$  représente la matrice de transformation d'entrée,  $\tau \in \mathfrak{R}^{n \times 1}$  est le vecteur d'entrée,  $H^T \in \mathfrak{R}^{m \times n}$  représente la matrice associée aux contraintes,  $\lambda \in \mathfrak{R}^{m \times 1}$  est le vecteur des forces de contrainte.

### 1.6.2.1 Robot différentiel :

Pour obtenir un modèle dynamique d'un robot mobile différentiel (figure 1.14), il suffit d'appliquer une des deux méthodes suivantes:

- Lagrange
- Newton-Euler

Dans cette partie on s'intéresse uniquement à la méthode de Lagrange. L'approche dynamique de Lagrange est une méthode très puissante pour formuler les équations du mouvement des systèmes mécaniques. Cette méthode, qui a été introduit par Lagrange est utilisée pour dériver systématiquement les équations du mouvement en considérant l'énergie cinétique et potentielle du système.

L'équation de Lagrange d'un système mécanique est décrite par [75,76] :

$$\frac{d}{dt} \left( \frac{\partial L}{\partial \dot{q}_i} \right) - \frac{\partial L}{\partial q_i} = \sum_{j=1}^n \lambda_j a_{ji} + Q_i, i = 1, 2, \dots, n \quad (1.33)$$

$q_1, q_2, \dots, q_n$  sont les coordonnées généralisées.

$L = T - P$  est la fonction lagrangienne,  $T$  est l'énergie cinétique du système,  $P$  est l'énergie potentielle du système,  $q_i$  sont des coordonnées généralisés,  $Q_i$  sont les forces non conservatrices dans le système,  $a_{ji}$  sont des fonctions des coordonnées généralisées et  $\lambda_j$  est le vecteur des multiplicateurs de Lagrange associés à la contrainte.

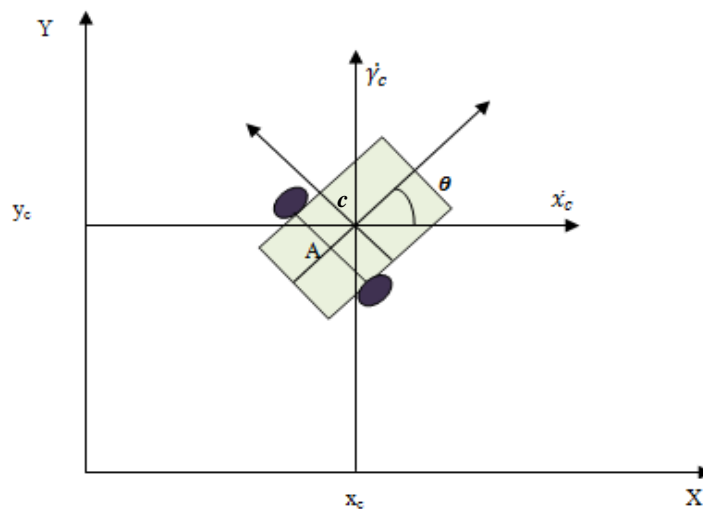


Figure 1.14 : Repérage d'un robot mobile de type différentiel

Les coordonnées généralisées sont sélectionnées comme :

$$q = [x_c \quad y_c \quad \theta \quad \varphi_R \quad \varphi_L]^T \quad (1.34)$$

Les énergies cinétiques du robot sont la somme de l'énergie cinétique de la plate-forme de robot sans roues plus les énergies cinétiques de la roue et actionneurs.

L'énergie cinétique est donnée par :

$$T = \frac{1}{2} m v_A^2 + \frac{1}{2} I_A \dot{\theta}^2 \quad (1.35)$$

Toutes les vitesses sont exprimées en fonction des coordonnées en utilisant l'équation de vitesse générale dans le cadre inertiel :

$$v_i^2 = \dot{x}_i^2 + \dot{y}_i^2 \quad (1.36)$$

Les composantes  $x_i$  et  $y_i$  du centre de masse peuvent être obtenus en termes de coordonnées généralisées:

$$\begin{cases} x_c = x_A + a \cos \theta \\ y_c = y_A + a \sin \theta \end{cases} \quad (1.37)$$

où,  $a$  représente la distance entre le centre de gravité de robot et le centre de l'axe des roues.

En utilisant les équations (1.35) avec les équations (1.36) et (1.37) l'énergie cinétique du robot est donnée par:

$$L = \frac{1}{2} m (\dot{x}_c^2 + \dot{y}_c^2) - ma\dot{\theta}(\dot{y}_c \cos \theta - \dot{x}_c \sin \theta) + \frac{1}{2} ma^2 \dot{\theta}^2 + \frac{1}{2} I_A \dot{\theta}^2 \quad (1.38)$$

où  $P$  est nulle,  $\dot{x}_c$  et  $\dot{y}_c$  représentent les vitesses du robot dans les axes  $x$ ,  $y$ .

Du théorème de l'axe parallèle, on obtient :

$$I_A = I_c + ma^2 \quad (1.39)$$

Par introduction de l'équation (1.39) dans (1.38), on obtient :

$$L = \frac{1}{2} m (\dot{x}_c^2 + \dot{y}_c^2) - ma\dot{\theta}(\dot{y}_c \cos \theta - \dot{x}_c \sin \theta) + ma^2 \dot{\theta}^2 + \frac{1}{2} I_c \dot{\theta}^2 \quad (1.40)$$

En utilisant l'équation (1.33) avec la fonction lagrangienne  $L=T$ , les équations de mouvement sont données par :

$$\frac{\partial}{\partial t} \left( \frac{\partial L}{\partial \dot{x}_c} \right) = m\ddot{x}_c + ma\ddot{\theta} \sin \theta + ma\dot{\theta}^2 \cos \theta \quad (1.41)$$



$$\frac{\partial}{\partial t} \left( \frac{\partial L}{\partial \dot{y}_c} \right) = m\ddot{y}_c - ma\ddot{\theta} \cos \theta + ma\dot{\theta}^2 \sin \theta \quad (1.42)$$

$$\begin{aligned} \frac{\partial}{\partial t} \left( \frac{\partial L}{\partial \dot{\theta}} \right) &= I_c \ddot{\theta} + 2ma^2 \ddot{\theta} + ma\ddot{x}_c \sin \theta + ma\dot{x}_c \dot{\theta} \cos \theta \\ &- ma\ddot{y}_c \cos \theta + ma\dot{y}_c \dot{\theta} \sin \theta \end{aligned} \quad (1.43)$$

En substituant les équations (1.41), (1.42) et (1.43) dans (1.33), on obtient:

$$m\ddot{x}_c + ma\ddot{\theta} \sin \theta + ma\dot{\theta}^2 \cos \theta = F_1 + C_1 \quad (1.44)$$

$$m\ddot{y}_c - ma\ddot{\theta} \cos \theta + ma\dot{\theta}^2 \sin \theta = F_2 + C_2 \quad (1.45)$$

$$(I_c + 2ma^2) \ddot{\theta} + ma\ddot{x}_c \sin \theta - ma\dot{\theta}^2 \dot{y}_c \cos \theta = \tau + C_3 \quad (1.46)$$

Avec  $F_1$ ,  $F_2$  sont des forces de l'actionneur sur les axes  $x$ , et  $y$ , ( $C_1$ ,  $C_2$ ,  $C_3$ ) sont des coefficients liés à la contrainte cinématique qui peuvent être écrits en fonction des multiplicateurs de Lagrange  $\lambda$  et de la matrice des contraintes cinématiques  $H$  :

$$H^T(q)\lambda = \begin{bmatrix} C_1 \\ C_2 \\ C_3 \end{bmatrix} \quad (1.47)$$

$$F = \begin{bmatrix} F_1 \\ F_2 \\ \tau \end{bmatrix} \quad (1.48)$$

La représentation matricielle de ces équations donne:

$$\begin{bmatrix} m & 0 & ma \sin \theta \\ 0 & m & -ma \cos \theta \\ ma \sin \theta & -ma \cos \theta & I_c + 2ma^2 \end{bmatrix} \ddot{q} + \begin{bmatrix} ma\dot{\theta}^2 \cos \theta \\ ma\dot{\theta}^2 \sin \theta \\ 0 \end{bmatrix} = \begin{bmatrix} F_1 \\ F_2 \\ \tau \end{bmatrix} + \begin{bmatrix} C_1 \\ C_2 \\ C_3 \end{bmatrix} \quad (1.49)$$

En tenant compte des forces engendrées par les actionneurs, on a:

$$F_1 = \left( \frac{\tau_1 + \tau_2}{R_a} \right) \cos \theta \quad (1.50)$$

$$F_2 = \left( \frac{\tau_1 + \tau_2}{R_a} \right) \sin \theta \quad (1.51)$$

$$\tau = \left( \frac{L}{R_a} \right) (\tau_1 - \tau_2) \quad (1.52)$$

Avec  $\tau_1$  et  $\tau_2$  les couples des roues droite et gauche et  $R_a$  le rayon de la roue.

La représentation matricielle de ces équations donne :

$$\begin{bmatrix} F_1 \\ F_2 \\ \tau \end{bmatrix} = \frac{1}{R_a} \begin{bmatrix} \cos \theta & \cos \theta \\ \sin \theta & \sin \theta \\ L & -L \end{bmatrix} \begin{bmatrix} \tau_1 \\ \tau_2 \end{bmatrix} \quad (1.53)$$

En tenant compte des contraintes non holonomes du robot différentiel:

- contrainte de glissement latéral:

$$\dot{y}_c \cos \theta - \dot{x}_c \sin \theta - \dot{\theta} a = 0 \quad (1.54)$$

- Contrainte de roulement pure:

$$\dot{x}_c \cos \theta + \dot{y}_c \sin \theta + L \dot{\theta} = R_a \dot{\phi}_r \quad (1.55)$$

$$\dot{x}_c \cos \theta + \dot{y}_c \sin \theta - L \dot{\theta} = R_a \dot{\phi}_l \quad (1.56)$$

Selon ces équations de contraintes, les forces de contrainte deviennent.

$$C_1 = m(\dot{x}_c \cos \theta + \dot{y}_c \sin \theta) \dot{\theta} \sin \theta \quad (1.57)$$

$$C_2 = -m(\dot{x}_c \cos \theta + \dot{y}_c \sin \theta) \dot{\theta} \cos \theta \quad (1.58)$$

$$C_3 = ma(\dot{x}_c \cos \theta + \dot{y}_c \sin \theta) \dot{\theta} \quad (1.59)$$

Une représentation matricielle des équations précédentes donne:

$$\begin{bmatrix} C_1 \\ C_2 \\ C_3 \end{bmatrix} = \begin{bmatrix} m(\dot{x}_c \cos \theta + \dot{y}_c \sin \theta) \dot{\theta} \sin \theta \\ -m(\dot{x}_c \cos \theta + \dot{y}_c \sin \theta) \dot{\theta} \cos \theta \\ md(\dot{x}_c \cos \theta + \dot{y}_c \sin \theta) \dot{\theta} \end{bmatrix} = H^T(q) \lambda \quad (1.60)$$

Avec:

$$H^T(q) = \begin{bmatrix} -\sin \theta \\ \cos \theta \\ -a \end{bmatrix} \quad (1.61)$$

$$\lambda = -m(\dot{x}_c \cos \theta + \dot{y}_c \sin \theta) \dot{\theta} \quad (1.62)$$

Toutes les équations du mouvement obtenues peuvent être représentées sous la forme générale suivante :

$$\begin{bmatrix} m & 0 & ma \sin \theta \\ 0 & m & -ma \cos \theta \\ ma \sin \theta & -ma \cos \theta & I_c + 2ma^2 \end{bmatrix} \ddot{q} + \begin{bmatrix} ma\dot{\theta}^2 \cos \theta \\ ma\dot{\theta}^2 \sin \theta \\ 0 \end{bmatrix} = \frac{1}{R_a} \begin{bmatrix} \cos \theta & \cos \theta \\ \sin \theta & \sin \theta \\ L & -L \end{bmatrix} \begin{bmatrix} \tau_1 \\ \tau_2 \end{bmatrix} + \begin{bmatrix} -\sin \theta \\ \cos \theta \\ -a \end{bmatrix} \lambda \quad (1.63)$$

Par identification de l'équation (1.63) et la forme générale (1.32), on obtient:

$$M(q) = \begin{bmatrix} m & 0 & ma \sin \theta \\ 0 & m & -ma \cos \theta \\ ma \sin \theta & -ma \cos \theta & I_c + 2ma^2 \end{bmatrix}, V(q, \dot{q}) = \begin{bmatrix} ma\dot{\theta}^2 \cos \theta \\ ma\dot{\theta}^2 \sin \theta \\ 0 \end{bmatrix}, \tau = \begin{bmatrix} \tau_1 \\ \tau_2 \end{bmatrix},$$

$$H^T(q) = \begin{bmatrix} -\sin \theta \\ \cos \theta \\ -a \end{bmatrix}, B(q) = \frac{1}{R_a} \begin{bmatrix} \cos \theta & \cos \theta \\ \sin \theta & \sin \theta \\ L & -L \end{bmatrix}. \quad (1.64)$$

Dans ce cas, on considère les frottements de la surface  $F(\dot{q})=0$  et le vecteur gravitationnel  $G(q)=0$ .

Les composantes du système (1.63) possèdent les caractéristiques suivantes :

- La matrice  $M(q)$ ,  $V(q, \dot{q})$  sont limitées.
- La matrice  $M - 2V(q, \dot{q})$  est asymétrique.

Cette propriété est particulièrement importante pour l'analyse de la stabilité du système.

Le système (1.63) est généralement transformé en une représentation plus appropriée, dans ce cas on doit faire une transformation tout en utilisant les deux équations suivantes :

$$V = \begin{bmatrix} v \\ \omega \end{bmatrix} \quad (1.65)$$

$$S(q) = \begin{bmatrix} \cos \theta & -a \sin \theta \\ \sin \theta & a \cos \theta \\ 0 & 1 \end{bmatrix} \quad (1.66)$$

Le modèle cinématique direct :

$$\dot{q} = \begin{bmatrix} \dot{x}_c \\ \dot{y}_c \\ \dot{\theta} \end{bmatrix} = \begin{bmatrix} \cos \theta & -a \sin \theta \\ \sin \theta & a \cos \theta \\ 0 & 1 \end{bmatrix} \begin{bmatrix} v \\ \omega \end{bmatrix} = S(q)V \quad (1.67)$$

où  $V(t)$  est un vecteur de vitesse.

Différencier l'équation (1.67) en substituant ce résultat dans (1.32), puis en multipliant par  $S^T(q)$ , les équations complètes de mouvement de la plate-forme mobile nonholonome sont données par :

$$\dot{q} = S(q)\mathcal{V} \quad (1.68)$$

$$S^T M S \dot{\mathcal{V}} + S^T (M \dot{S} + V(q, \dot{q})S) \mathcal{V} + S^T F(\dot{q}) + S^T G(q) + S^T \tau_d = S^T B \tau - S^T H^T \quad (1.69)$$

L'équation (1.69) peut être réécrite comme :

$$\bar{M}(q)\dot{\mathcal{V}} + \bar{V}(q, \dot{q})\mathcal{V} + \bar{F}(\dot{q}) + \bar{G}(q) + \bar{\tau}_d = \bar{B}(q)\tau \quad (1.70)$$

Avec

$$\bar{M}(q) = S^T M S = \begin{bmatrix} m & 0 \\ 0 & I_c + ma^2 \end{bmatrix}, S^T(q)H^T(q) = 0$$

$$\bar{V}(q, \dot{q}) = S^T (M \dot{S} + V(q, \dot{q})S) = \begin{bmatrix} 0 & 0 \\ 0 & 0 \end{bmatrix}$$

$$\bar{F}(\dot{q}) = S^T F(\dot{q}) = 0, \bar{\tau}_d = S^T \tau_d, \bar{G}(q) = S^T G(q) = 0, \bar{B}(q) = S^T B = \frac{1}{R_a} \begin{bmatrix} 1 & 1 \\ L & -L \end{bmatrix}.$$

### 1.6.2.2 Robot unicycle:

La modélisation dynamique d'un robot unicycle (figure 1.15) est généralement similaire à celle du robot différentiel tout en utilisant la formulation d'Euler-Lagrange [77,78].

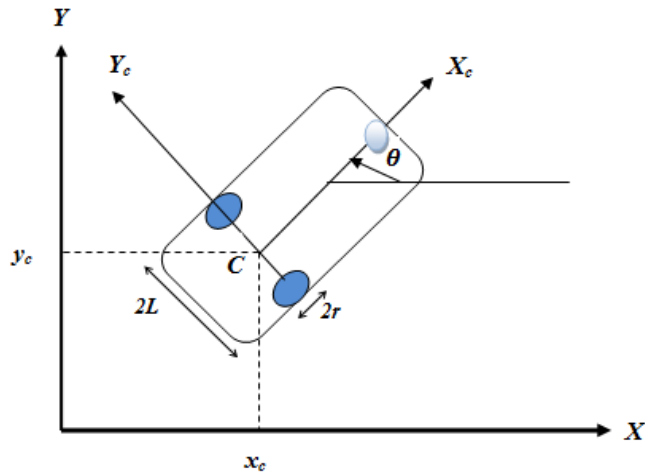


Figure 1.15 : Repérage d'un robot mobile de type unicycle

Le modèle dynamique du robot mobile unicycle, en tenant compte des incertitudes et des perturbations, peut être décrit par :

$$M(q)\ddot{q} + V(q, \dot{q}) + F(\dot{q}) + G(q) + \tau_d = B(q)\tau - H^T(q)\lambda \quad (1.71)$$

Avec :

$$M(q) = \begin{bmatrix} m & 0 & 0 \\ 0 & m & 0 \\ 0 & 0 & I \end{bmatrix}, V(q, \dot{q}) = 0, F(\dot{q}) = 0, G(q) = 0, \tau = \begin{bmatrix} \tau_1 \\ \tau_2 \end{bmatrix}, H^T(q) = \begin{bmatrix} -\sin\theta \\ \cos\theta \\ 0 \end{bmatrix},$$

$$\lambda = -m(\dot{x}_c \cos\theta + \dot{y}_c \sin\theta)\theta, B(q) = \frac{1}{r} \begin{bmatrix} \cos\theta & \cos\theta \\ \sin\theta & \sin\theta \\ L & L \end{bmatrix}$$

Le système (1.70) est généralement transformé en un modèle plus approprié qui peut être donné par :

$$\bar{M}(q)\dot{V} + \bar{V}(q, \dot{q})V + \bar{F}(\dot{q}) + \bar{G}(q) + \bar{\tau}_d = \bar{B}(q)\tau - S^T H^T \quad (1.72)$$

$$\bar{M}(q)\dot{V} + \bar{\tau}_d = \bar{B}(q)\tau \quad (1.73)$$

Avec :

$$S(q) = \begin{bmatrix} \cos\theta & 0 \\ \sin\theta & 0 \\ 0 & 1 \end{bmatrix}, \bar{M}(q) = S^T M S = \begin{bmatrix} m & 0 \\ 0 & I \end{bmatrix}, \bar{V}(q, \dot{q}) = \begin{bmatrix} 0 & 0 \\ 0 & 0 \end{bmatrix}, \bar{F}(\dot{q}) = S^T F(\dot{q}) = 0,$$

$$\bar{\tau}_d = S^T \tau_d, \bar{G}(q) = S^T G(q) = 0, \bar{B}(q) = S^T B = \frac{1}{r} \begin{bmatrix} 1 & 1 \\ L & -L \end{bmatrix}, S^T(q)H^T(q) = 0.$$

## 1.7 Conclusion

Dans ce chapitre, nous avons donné brièvement une présentation sur les tâches les plus intéressantes dans le domaine de la robotique mobile qui sont notamment la perception, la navigation et l'évitement d'obstacles.

Différents types de robots mobiles à roues sont présentés avec une modélisation cinématique pour chaque type de robot. Ensuite, on s'est limité également à la modélisation dynamique des robots unicycle et différentiel tout en utilisant l'approche de Lagrange dans le but d'appliquer par la suite des lois de commande.

## **2<sup>ER</sup> CHAPITRE**

# **LOCALISATION DES ROBOT MOBILE**

---

---

## **CHAPITRE 2**

### **Localisation des robots mobiles**

#### **2.1 Introduction**

Il est bien connu que la navigation est l'une des propriétés les plus importantes en robotique mobile [79] et de ce fait, pour garantir une bonne navigation, quelques composantes sont exigées.

*La perception* : Le robot doit analyser les informations fournies par les capteurs pour extraire les données pertinentes.

*La localisation* : Le robot doit situer sa position dans l'environnement.

*La cognition* : Le robot doit décider comment agir pour atteindre ses objectifs.

*Le contrôle de mouvement* : Le robot doit moduler ses sorties moteur pour atteindre la trajectoire souhaitée.

La localisation du robot est définie comme étant la solution majeure pour assurer l'autonomie du robot. Par conséquent, si le robot ne connaît pas sa position actuelle, il lui serait difficile de poursuivre sa mission (figure 2.1).

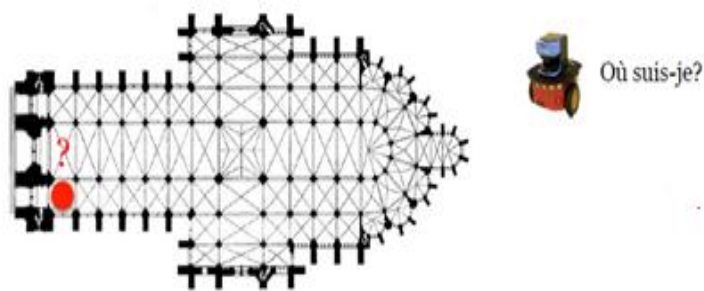


Figure 2.1: Positionnement du robot [80]

Il est évident que la localisation du robot exige des informations et des mesures concernant sa situation dans l'environnement de travail et à partir de ces informations, il doit déterminer sa position aussi précisément que possible. La figure (2.2) suivante illustre un

organigramme de navigation d'un robot mobile précisant l'emplacement du bloc de la localisation.

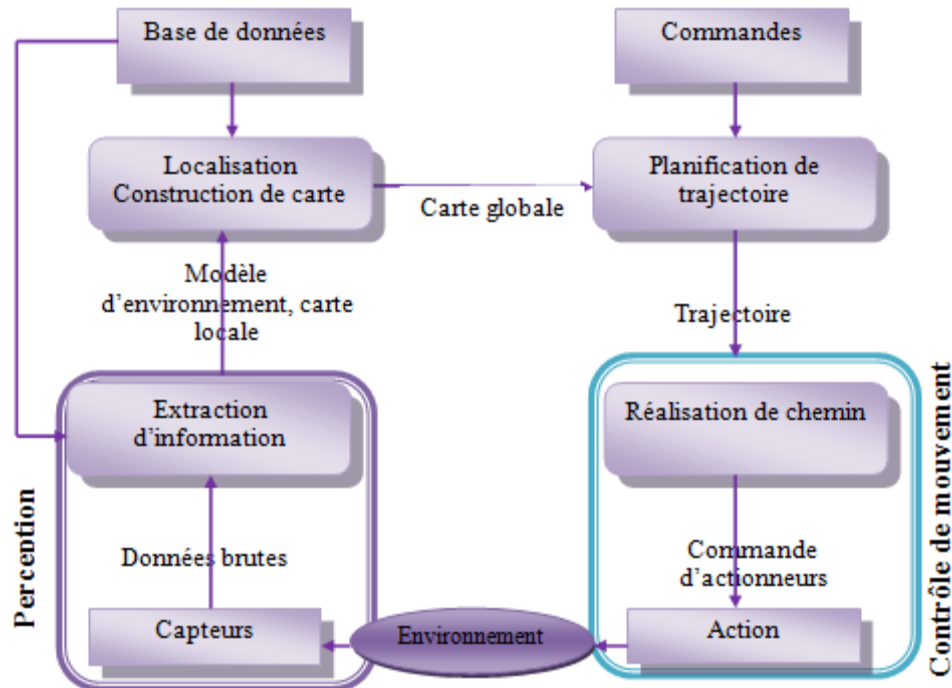


Figure 2.2 : Organigramme de navigation [80]

Dans ce chapitre on s'intéresse seulement à la localisation des robots mobiles qui joue un rôle très important dans la navigation et qui a reçu beaucoup d'échos de la part des chercheurs en robotique durant la dernière décennie [81,82].

Nous allons décrire les différents types d'environnements et les différents capteurs utilisés en robotique mobile. Quelques techniques de localisation sont aussi présentées. Par la suite, une méthode de localisation basée sur des amers et utilisant le filtre de kalman étendu (FKE) est présentée. Des travaux de simulations sous différents types d'environnement avec différents nombres d'amers sont entrepris et une conclusion concernant les résultats obtenus est donnée.

## 2.2 Types d'environnement

Pour la localisation des robots mobiles, les environnements sont généralement supposés connus, mais le problème qui reste posé est que l'environnement n'est pas idéalement connu.



Aussi la solution est de le représenter sous forme d'une carte et cette dernière peut avoir différentes représentations possibles.

### **2.2.1 Environnements intérieurs et extérieurs**

En général, il est possible de distinguer entre les environnements intérieurs et extérieurs. Pour les environnements extérieurs, la représentation réside dans l'extérieur de bâtiments et généralement l'environnement est difficile et non structuré. Prenant comme un exemple la localisation d'un robot mobile dans une ville, le capteur qui est capable d'examiner l'environnement dans de telle condition est le GPS. Mais par contre, les capteurs sonars ne sont pas généralement efficaces car l'environnement est de grande taille ainsi que l'odométrie qui est particulièrement insuffisante [83,84].

Concernant la localisation dans un environnement interne, on peut prendre comme exemple un aspirateur dans une maison. Dans ce cas, les environnements sont considérés comme structurés et de petites tailles. Les capteurs de type sonar, télémètres laser et caméras sont les plus utilisés pour la localisation interne du robot. Par contre le capteur GPS est mal placé dans ce type d'environnement [85].

### **2.2.2 Environnements statiques et dynamiques**

Pour un environnement statique, les objets sont considérés immobiles et ne changent pas en fonction du temps (arbres, murs...etc.). Par conséquent, le seul problème réside dans la détermination de la posture du robot.

Concernant un environnement dynamique, les objets sont dynamiques et changent en fonction du temps. Autrement dit, il y a des objets qui évoluent dans l'environnement. On peut conclure que la localisation dans un environnement dynamique est plus difficile que celle d'un environnement statique. Dans ce travail, on s'intéresse uniquement à la localisation du robot dans un environnement statique.

## **2.3 Différentes représentations d'environnements**

En robotique mobile, il existe plusieurs types de représentations d'environnements. Chacun a des avantages et des inconvénients qui sont plus ou moins adéquats selon la tâche à accomplir.

### **2.3.1 Cartes topologiques**

Les approches topologiques évitent la mesure directe des qualités environnementales géométriques, en se concentrant plutôt sur les caractéristiques de l'environnement qui sont les plus pertinentes pour la localisation du robot.

Elles permettent de représenter l'environnement sous forme d'un graphe qui spécifie deux choses, les nœuds et les connectivités entre ces nœuds. Les nœuds sont utilisés pour désigner les zones du monde et les arcs sont utilisés pour dénoter la continuité des paires de nœuds [86].

La figure (2.3) représente une carte topologique d'un ensemble de couloirs et de bureaux dans un environnement intérieur.

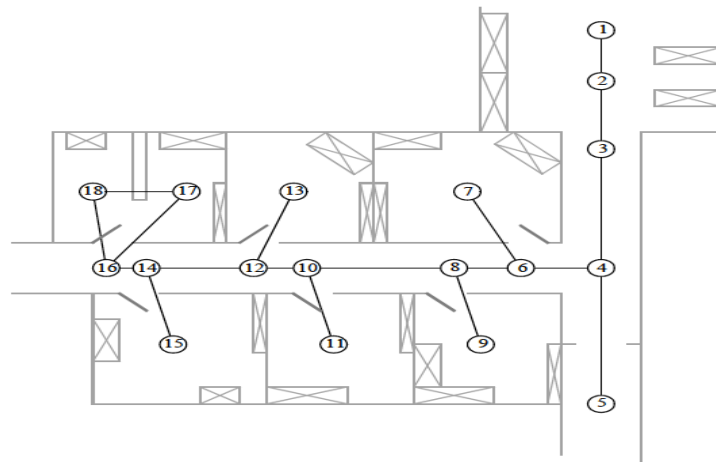


Figure 2.3: Représentation topologique d'un espace de bureau intérieur [79].

La figure (2.4) représente une carte topologique lorsque les capteurs n'utilisent pas les modèles métriques, alors que les informations sont en général enregistrées dans une carte topologique [87].

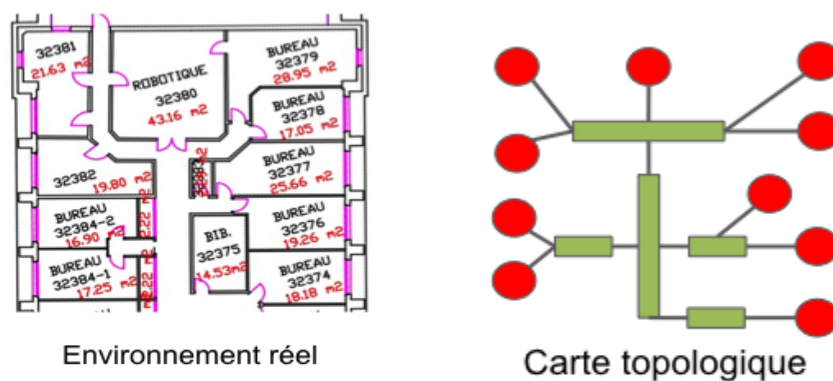


Figure 2.4: Représentation topologique des pièces et des couloirs.

### 2.3.2 Cartes métriques

Pour les cartes métriques, l'environnement est représenté par une série d'objets auxquels sont accouplées des positions dans un plan métrique. Les positions des objets sont alors calculées tout en utilisant la position estimée du robot [88].

### 2.3.3 Modélisation hybride

Les cartes hybrides sont des cartes possédant plusieurs représentations possibles qui permettent de profiter des avantages de chacune d'elles. Par exemple, plusieurs travaux proposent des méthodes hybrides métriques-topologiques dans le but de faire combiner la fiabilité des cartes métriques avec des cartes topologiques [89].

### 2.3.4 Modèle de primitives (carte d'amers)

Les primitives ou les amers sont des parties importantes dans l'environnement que l'on peut facilement extraire via un capteur donné et qui ont une description paramétrique. Les balises représentent des amers artificiels qui peuvent très bien être utilisées lorsque l'environnement est structuré, comme dans les environnements de bureau et d'usine. En extérieur, les inconvénients peuvent être trop importants. Dans ce cas, on utilise les repères naturels qui semblent les mieux adaptés.

Le problème de la localisation basé sur une carte d'amers exige une identification des caractéristiques. En d'autre terme, il faut calculer la position du robot tout en prenant plusieurs amers avec leurs observations (figure 2.5), sachant que les observations sont déterminées par le robot.

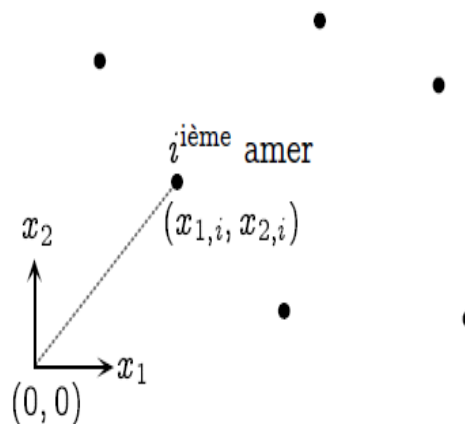


Figure 2.5 : Représentation d'un modèle d'une carte d'amers [90]

## 2.4 Types de capteurs

Les capteurs sont les éléments fondamentaux du robot mobile durant le processus de perception, donc il est indispensable que les capteurs puissent percevoir l'état de l'environnement. Le bruit du capteur introduit une limitation de la cohérence des lectures et souvent certaines caractéristiques environnementales ne sont pas capturées par le robot et sont donc éliminées. Le problème majeur des capteurs demeure dans le fait que l'information perçue ne soit pas directement exploitable [90].

On présente dans cette section les capteurs les plus utilisés en robotique mobile pour les besoins de la navigation. On classe généralement les capteurs en deux catégories, les capteurs proprioceptifs et les capteurs extéroceptifs.

### 2.4.1 Capteurs proprioceptifs

Les capteurs proprioceptifs sont des capteurs qui permettent une mesure du déplacement du robot. Ils peuvent être utilisés pour la localisation mais présentent une dérive importante au cours du temps.

#### 2.4.1.1 Odométrie

L'odomètre fonctionne en intégrant des informations incrémentales au cours du temps. En utilisant des encodeurs de roue pour compter le nombre de tours de chaque roue, le robot mesure la distance parcourue et sa direction. L'odomètre est largement utilisé car il donne une bonne précision à court terme, est peu coûteux et permet des taux d'échantillonnage très élevés. Cependant, en raison de la dérive et du glissement, l'intégration des tours de roue conduit à des erreurs en distance parcourue et en orientation qui s'accumulent au fil du temps [91]. Un autre inconvénient de l'odométrie est sa sensibilité sur le terrain.

#### 2.4.1.2 Systèmes inertiels

Les techniques de navigation inertielle utilisent des gyroscopes et des accéléromètres pour mesurer le taux de rotation et d'accélération du robot. Les gyroscopes détectent les petites accélérations d'orientation.

Les accéléromètres mesurent les petites accélérations d'un véhicule mobile. Ils subissent une forte dérive et sont sensibles au sol [92].

### 2.4.2 Capteurs extéroceptifs

Les capteurs extéroceptifs sont des capteurs qui donnent une information relative sur l'environnement et permettent aussi de percevoir l'évolution du robot.

#### 2.4.2.1 Capteurs ultrasons

Le principe de base d'un capteur ultrasons est de transmettre un paquet d'ondes ultrasonores et de mesurer le temps de vol (aller-retour). La distance  $d$  entre l'objet et le capteur peut être calculée en fonction de la vitesse de propagation du son  $c$  et du temps de vol  $t$ .

$$d = \frac{ct}{2} \quad (2.1)$$

L'onde ultrasonore a généralement une fréquence entre 40 et 180 kHz et elle est générée par un transducteur piézoélectrique ou électrostatique.

#### 2.4.2.2 Télémètres lasers

Le télémètre laser est un capteur qui apporte des améliorations significatives par rapport au capteur ultrasonique dû à l'utilisation de la lumière laser au lieu du son. Ce type de capteur consiste en un émetteur qui illumine une cible avec un faisceau parallèle et un récepteur capable de détecter la composante de la lumière, qui est essentiellement coaxiale au faisceau transmis.

Une façon de mesurer le temps de vol du faisceau lumineux consiste à utiliser un laser puisé, puis à mesurer directement le temps écoulé, comme dans la solution ultrasonique décrite précédemment. Une seconde méthode consiste à mesurer la fréquence de battement entre une onde continue modulée en fréquence et sa réflexion reçue. Une autre méthode encore plus simple consiste à mesurer le déphasage de la lumière réfléchie.

*Mesure de déphasage* : la lumière proche infrarouge (d'une diode électroluminescente [LED] ou laser) est transmise depuis l'émetteur et atteint un point P d'environnement (figure 2.6).

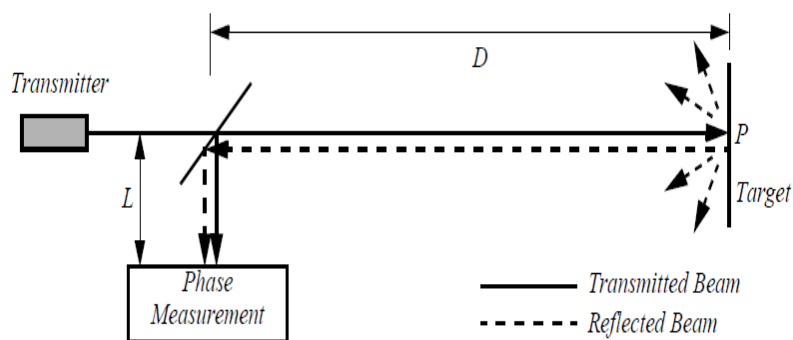


Figure 2.6 : Schéma de télémétrie laser par mesure de déphasage.

### 2.4.2.3 Systèmes de vision

La vision est un capteur très performant, il fournit une énorme quantité d'informations sur l'environnement et permet une interaction riche et intelligente dans les environnements dynamiques. Les systèmes de vision permettent l'identification des amers et la localisation tout en utilisant une carte de primitives ou une carte topologique. Généralement, il existe deux types d'amers, artificiels et naturels.

Les amers artificiels sont des repères conçus pour être reconnus par les robots [93]. Ils sont placés à des endroits dans l'environnement qui sont connus à l'avance et qui sont bien visibles pour les capteurs du robot. Ces repères sont spécifiquement conçus et placés dans l'environnement pour la localisation des robots.

Les amers naturels sont utilisés comme moyen de localisation pour les robots [94] et font déjà partie de l'environnement du robot.

## 2.5 Différentes techniques de localisation

### 2.5.1 Localisation relative

La localisation relative est effectuée à l'estime, c'est-à-dire par la mesure du déplacement d'un robot mobile entre deux emplacements. Les estimations individuelles des positions locales ne sont pas exactes, les erreurs sont accumulées et l'erreur absolue dans l'estimation du mouvement total augmente avec la distance parcourue [95].

Pour un robot mobile, la méthode d'estimation se fait par odométrie et repose sur les données obtenues à partir des codeurs incrémentaux [96].

Les codeurs sont utilisés pour calculer le déplacement de chacune des deux roues. On note ces déplacements  $dUg$  et  $dUd$  (figure 2.7). Ceux-ci sont en fait des longueurs d'arcs de cercle, mais le déplacement étant petit on peut les estimer égales à celles des cordes correspondantes [97].

L'hypothèse de base de l'odométrie est que les déplacements des roues peuvent être transformés en déplacements linéaires par rapport au sol. Cette hypothèse est rarement valable en raison du patinage des roues et d'autres causes.

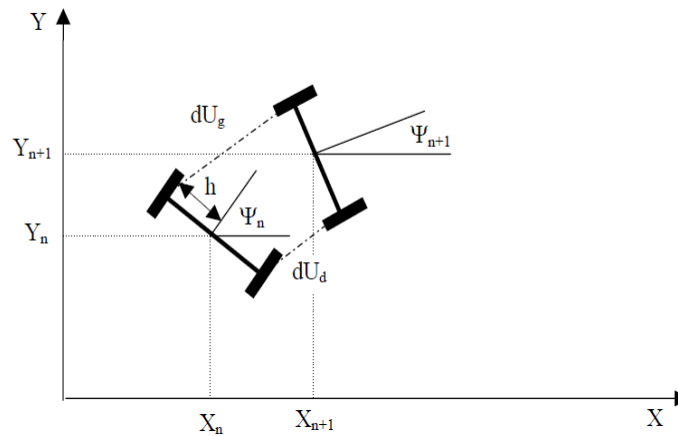


Figure 2.7 : Principe de l'odométrie [105]

### 2.5.2 Localisation absolue

Les repères et les balises constituent la base de la localisation absolue d'un robot mobile [98]. Les repères peuvent avoir ou non des identités. Si les repères ont des identités, aucune connaissance préalable de la position du robot n'est nécessaire. Sinon, si les repères n'ont aucune identité, certaines connaissances approximatives de la position du robot doit être disponible. Si des repères naturels sont utilisés, les identités dépendent alors de l'environnement.

### 2.5.3 Approches probabilistes

L'approche de Markov et le filtrage de Kalman sont deux stratégies populaires pour la localisation des robots mobiles dans des environnements d'intérieur. Certaines techniques qui méritent d'être mentionnées sont la localisation du filtre de Kalman non parfumé (UKF) [99,100], la localisation par grille et la localisation par Monte Carlo. Le filtre UKF est similaire à EKF en ce sens qu'il suppose également des distributions gaussiennes mais repose sur une manière différente pour linéariser les modèles de mouvement et de mesure. Inversement, la localisation par grille et la localisation par Monte Carlo ne sont pas limitées aux distributions uni-modales. La localisation par grille utilise l'histogramme pour représenter la croyance de présence du robot et la localisation par Monte Carlo utilise les filtres particuliers [101,102].

## 2.6 Localisation basée sur le filtre de Kalman étendu

### 2.6.1 Filtre de Kalman linéaire (FKL)

Le filtre linéaire de Kalman (FKL) est un algorithme récursif de traitement de données qui estime l'état d'un système dynamique linéaire basé sur l'utilisation de mesures bruitées [103,104,105]. Le FKL consiste en une étape d'initialisation après laquelle viennent les étapes de prédiction et de correction. Il prédit d'abord l'état du système puis il corrige sa prédiction en utilisant les mesures ou informations présentes.

#### *Algorithme*

L'état du filtre est représenté par:

$\hat{X}_{k/k}$  : L'estimation l'état à l'instant k.

$P_{k/k}$  : La matrice de covariance à l'instant k.

#### *Equations de Prédiction*

À chaque pas, le LKF calcule une nouvelle estimation en utilisant les équations suivantes:

$$\hat{X}_{k/k-1} = F_k \hat{X}_{k-1/k-1} + W_k U_{k-1} \quad (2.2)$$

$$P_{k/k-1} = F_k P_{k-1/k-1} F_k^T + Q_k \quad (2.3)$$

Avec,  $F_k$  et  $W_k$  sont des matrices connues,  $U_k$  est un vecteur de contrôle,  $P$  est une matrice de covariance et  $Q_k$  est une matrice gaussienne.

#### *Equations de Correction*

$$\hat{Y}_k = Y_k - H_k \hat{X}_{k/k-1} \quad (2.4)$$

$$P_k = H_k P_{k/k-1} H_k^T + R_k \quad (2.5)$$

$$K_k = P_{k/k-1} H_k^T P_k^{-1} + R_k \quad (2.6)$$

$$\hat{X}_{k/k} = F_k \hat{X}_{k/k-1} + K_k \hat{Y}_k \quad (2.7)$$

$$P_{k/k} = (I - K_k H_k) P_{k/k-1} \quad (2.8)$$

où  $Y_k$  est un vecteur d'observation ou de mesure,  $H_k$  est une matrice jacobienne connue,  $R_k$  est une matrice gaussienne et  $I$  la matrice d'identité.



### 2.6.2 Filtre de kalman non linéaire (FKE)

Le filtre de Kalman étendu est probablement le filtre le plus généralement utilisé dans les applications.

On considère le système non linéaire suivant:

$$\begin{cases} x_{k+1} = f(x_k, u_k, w_k) \\ y_{k+1} = h(x_k, v_k) \end{cases} \quad (2.9)$$

où  $x_k$  est le vecteur d'état,  $u_k$  est le vecteur de contrôle, et  $y_k$  est le vecteur de mesure.  $f$  est la fonction d'évolution du système,  $h$  représente la relation entre le vecteur d'état et les mesures.

$$Q = E[w_k, w_k^T] \quad (2.10)$$

$$R = E[v_k, v_k^T] \quad (2.11)$$

$w_k$  et  $v_k$  sont respectivement les bruits gaussiens d'état et de mesure.

#### Algorithme

Le filtre de kalman étendu est également basé sur deux étapes, prédiction et correction

#### Equations de Prédiction

$$\begin{cases} \hat{x}_{k+1/k} = f(x_{k/k}, u_k, 0) \\ P_{k+1/k} = F_k P_{k/k} F_k^T + W_k Q W_k^T \end{cases} \quad (2.12)$$

#### Equations de Correction

$$\begin{cases} K_k = P_{k+1/k} H_k^T (H_k P_{k+1/k} H_k^T + V_k R V_k^T)^{-1} \\ \hat{x}_{k+1/k+1} = \hat{x}_{k+1/k} + K_k (y_k - h(\hat{x}_{k+1/k}, 0)) \\ P_{k+1/k+1} = P_{k+1/k} - K_k H_k P_{k+1/k} \end{cases} \quad (2.13)$$

où  $\hat{x}_{k+1/k}$  dénote le vecteur de prédiction d'état à priori,  $\hat{x}_{k+1/k+1}$  est le vecteur de prédiction d'état à postériori,  $P_{k+1/k}$  désigne la matrice de covariance d'erreur de prédiction à priori,  $P_{k+1/k+1}$  est la matrice de covariance d'erreur de prédiction à postériori et  $K$  est la matrice du gain.

où,  $F_k$ ,  $W_k$ ,  $H_k$  et  $V_k$  sont les matrices jacobéennes données par:

$$F_k = \left. \frac{\partial f(x, 0)}{\partial x} \right|_{x=\hat{x}_k}, W_k = \left. \frac{\partial f(\hat{x}_k, w)}{\partial w} \right|_{w=0}, H_k = \left. \frac{\partial h(x, 0)}{\partial x} \right|_{x=\hat{x}_k}, V_k = \left. \frac{\partial h(\hat{x}_k, v)}{\partial v} \right|_{v=0} \quad (2.14)$$

### 2.6.3 Descriptions des modèles

L'algorithme pour la résolution du problème de la localisation utilise les deux modèles suivants :

1. Modèle d'évolution du robot représenté par la fonction  $f(\cdot)$

$$X(k+1) = f(X(k), u(k)) + w_k \quad (2.15)$$

2. Modèle d'observation des amers représenté par la fonction  $h(\cdot)$

$$Z(k) = h(X(k)) + v_k \quad (2.16)$$

Avec :  $Z(k) = [r \ \varphi]^T$  est 1

e vecteur d'observations.

L'équation (2.15) est caractérisée par une fonction  $f(\cdot)$  qui dénote l'évolution du robot et l'équation (2.16) représente des observations des amers,  $v_k$  et  $w_k$  représentent des bruits gaussiens.

#### 2.6.3.1 Modèle d'évolution du robot mobile

Considérons un robot mobile de type unicycle avec deux roues motrices, montées sur les côtés gauche et droit du robot, avec leur axe commun passant par le centre du robot (figure 2.8).

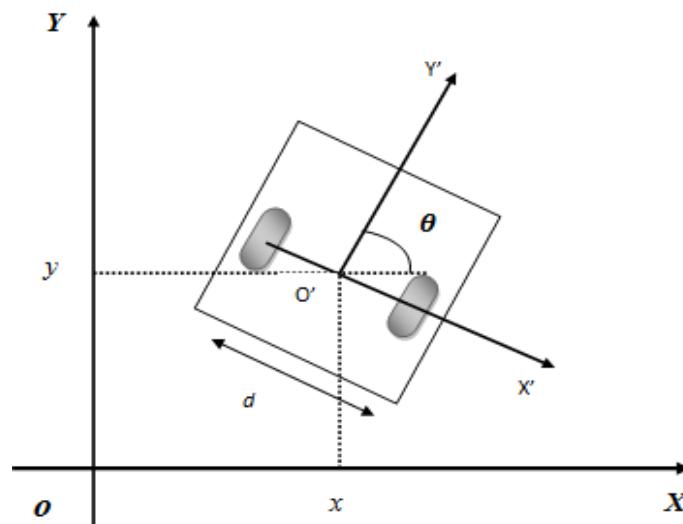


Figure 2.8 : Modèle d'un robot unicycle

Le modèle du robot unicycle est décrit par les équations suivantes:

$$\begin{cases} \dot{x}(t) = v(t) \cos \theta(t) \\ \dot{y}(t) = v(t) \sin \theta(t) \\ \dot{\theta}(t) = \omega(t) \end{cases} \quad (2.17)$$

Avec

$$v(t) = \frac{v_r(t) + v_l(t)}{2} \quad (2.18)$$

$$\omega(t) = \frac{v_r(t) - v_l(t)}{d} \quad (2.19)$$

Avec  $x(t)$  et  $y(t)$  sont les coordonnées du centre de l'axe des roues motrices,  $\theta(t)$  est l'angle d'orientation du robot,  $v(t)$  et  $\omega(t)$  sont les différentes vitesses du robot,  $v_r(t)$  et  $v_l(t)$  sont les vitesses de déplacement droit et gauche et  $d$  est la distance entre les deux roues du robot. Les encodeurs placés sur les roues motrices fournissent une mesure des angles sur une période d'échantillonnage. Les mesures odométriques sont utilisées pour obtenir une estimation du déplacement du robot et les vitesses qui sont constantes sur une période d'échantillonnage. Le système ci-dessus est discrétisé et exprimé sous forme linéaire [106]:

$$\begin{cases} x(k+1) = x(k) + \Delta T (v_k + \delta_v) \cos \theta(k) \\ y(k+1) = y(k) + \Delta T (v_k + \delta_v) \sin \theta(k) \\ \theta(k+1) = \theta(k) + \Delta T (\omega_k + \delta_\omega) \end{cases} \quad (2.20)$$

Avec :  $X(k) = \begin{bmatrix} x(k) \\ y(k) \\ \theta(k) \end{bmatrix}$ ,  $u(k) = \begin{bmatrix} \Delta T v_k \\ \Delta T \omega_k \end{bmatrix}$ ,  $\delta(k) = \begin{bmatrix} \Delta T \delta_v \\ \Delta T \delta_\omega \end{bmatrix}$ ,  $\delta_v$  et  $\delta_\omega$  sont des bruits sur les

vitesses du robot.

On peut écrire (2.20) sous la forme :

$$X(k+1) = f(X(k), u(k), \delta(k)) \quad (2.21)$$

### 2.6.3.2 Modèle d'amer et le modèle d'observation

Dans le contexte de la localisation, un amer est une entité (point, segment, plan, objet...etc.) de l'environnement qui peut être observée par les capteurs du robot.

Pour l'algorithme de la localisation, les amers sont supposés des points fixes sachant qu'ils sont présentés dans le repère global du robot. Donc le modèle simple des amers est :

$$M = [m_1 \quad m_2 \quad m_3 \quad \dots \quad m_n] \quad (2.22)$$

Avec :

$$m_i = \begin{bmatrix} x_i^a & y_i^a \end{bmatrix}^T \quad (2.23)$$

$x_i^a$  la position de l'amer dans l'axe x.

$y_i^a$  la position de l'amer dans l'axe y.

Les perceptions donnent des observations sur la distance et l'angle d'un amer de position connu, calculé par rapport au robot, (figure 2.9). On admet qu'on a  $m_i$  amers dans l'environnement représentés par des points connus. On peut également considérer un capteur télémétrique laser placé à la verticale du milieu des roues.

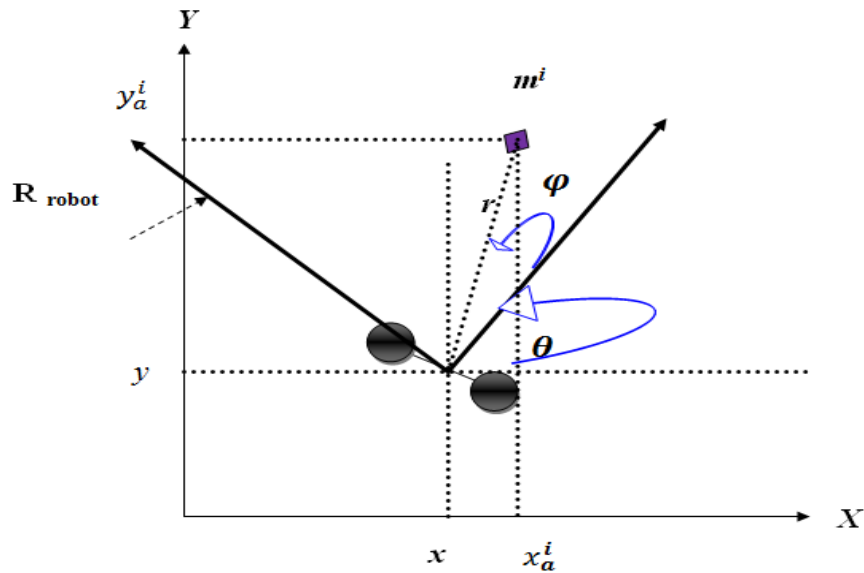


Figure 2.9: Distance et angle de direction mesurés par télémétrie

La distance et l'angle par rapport à un amer  $m_i$  sont traités comme des mesures de l'angle  $\varphi(k)$  par rapport à l'axe des  $x$  et la distance  $r(k)$  entre les coordonnées cartésiennes du robot  $(x, y)$ . L'amer  $(x_i^a, y_i^a)$  à un instant donné  $k$  peut être lié aux variables d'état du système  $(x, y, \theta)$  [107]:

$$\begin{cases} r(k) = \sqrt{(y_a^i - y)^2 + (x_a^i - x)^2} + r_v \\ \varphi(k) = \arctan 2(y_a^i - y, x_a^i - x) + \theta_v \end{cases} \quad (2.24)$$

où  $r_v, \theta_v$  sont des bruits de mesure.

La fonction ( $\text{atan2}(x, y)$ ) est similaire à la fonction ( $\arctan(y/x)$ ), mais elle possède deux arguments et renvoie des valeurs bornées entre  $-\pi$  et  $+\pi$ . L'équation (2.24) peut s'écrire sous la forme suivante:

$$Z(k) = h(X_k, M) \quad (2.25)$$

$$Z_{i,k} = \begin{bmatrix} r_i(k) \\ \varphi_i(k) \end{bmatrix} = \begin{bmatrix} \sqrt{(y_a^i - y)^2 + (x_a^i - x)^2} + r_v \\ \text{atan2}(y_a^i - y, x_a^i - x) + \theta_v \end{bmatrix} \quad (2.26)$$

Avec  $h(\cdot)$  est la fonction d'observation. L'indice  $i$  correspond à chaque amer repéré.

#### 2.6.4 Algorithme de localisation basé sur le FKE

Dans cette section on présente l'algorithme de localisation utilisé pour produire les estimations de la position du robot.

On utilise le formalisme du filtre de Kalman étendu. L'objectif de cet algorithme de localisation est de donner une estimation de la position du robot à l'instant  $k+1$  en utilisant l'estimation de la position à l'instant  $k$ , l'entrée de commande  $u(k)$  et la nouvelle observation de l'amer [108].

L'algorithme est constitué des différentes étapes illustrées sur la figure (2.10).

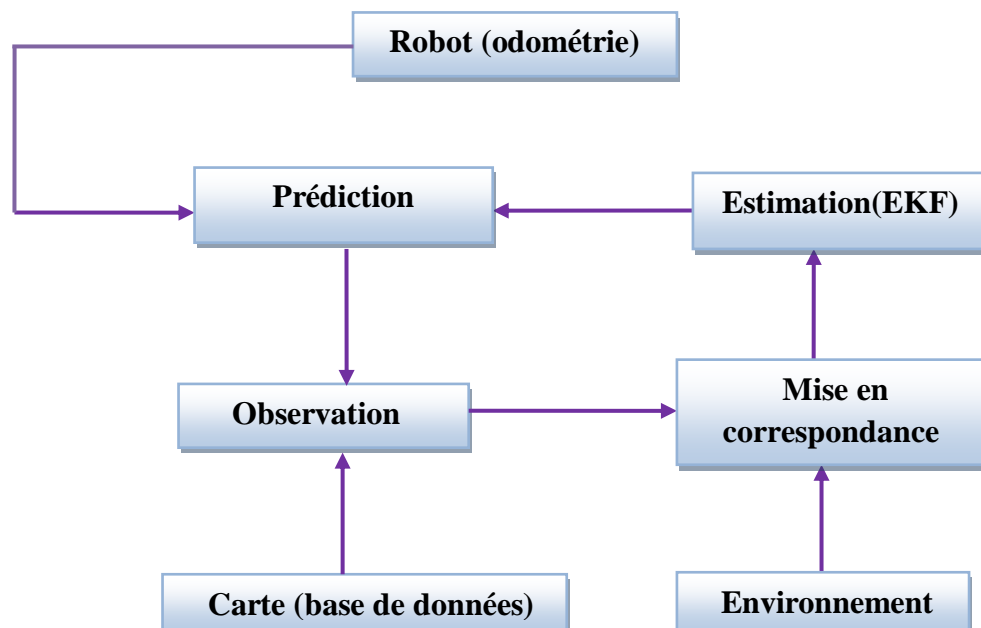


Figure 2.10 : Organigramme de la localisation par EKF

**Etape d'observation:** Dans l'étape d'observation les capteurs externes fournissent des données sur l'environnement.

**Etape de mise en correspondance :** Dans cette étape, on doit identifier toutes les mesures reçues par rapport aux observations prédites dans le repère local du robot.

**Estimation :** Elle consiste à intégrer les données issues de la perception ainsi que l'état prédit du robot pour donner une estimation de la position du robot et des incertitudes.

#### 2.6.4.1 Prédiction

L'étape de prédiction du FKE utilise le modèle de mouvement du robot pour produire une estimation de sa position  $f(\hat{X}_{k/k}, u(k))$  à l'instant  $k+1$  sachant que les informations précédentes de  $k$  sont disponibles:

$$\begin{cases} \hat{x}_{k+1/k} = \hat{x}_{k/k} + \Delta T v_k \cos \hat{\theta}_{k/k} \\ \hat{y}_{k+1/k} = \hat{y}_{k/k} + \Delta T v_k \sin \hat{\theta}_{k/k} \\ \hat{\theta}_{k+1/k} = \hat{\theta}_{k/k} + \Delta T \omega_k \end{cases} \quad (2.27)$$

$$\hat{X}_{k+1/k} = f(\hat{X}_{k/k}, u(k)) = \hat{X}_{k/k} + \begin{bmatrix} \Delta T v_k \cos \hat{\theta}_{k/k} \\ \Delta T v_k \sin \hat{\theta}_{k/k} \\ \Delta T \omega_k \end{bmatrix} \quad (2.28)$$

$$\text{Où } u(k) = \begin{bmatrix} \Delta T v(k) \\ \Delta T \omega(k) \end{bmatrix}.$$

Dans l'équation (2.28),  $\hat{X}_{k+1/k}$  indique la moyenne estimée de la distribution de posture. Pour calculer une nouvelle moyenne de la distribution de posture du robot  $\hat{X}_{k+1/k}$ , on a besoin de calculer la covariance estimée  $P(k+1)$  pour la nouvelle posture. Pour ce faire, on multiplie la covariance  $P(k)$  par la dérivée (également appelée jacobienne) du modèle qui est donné comme suit:

$$F = \frac{\partial f(\hat{X}_{k/k}, u(k))}{\partial \hat{X}_{k/k}} = \begin{bmatrix} 1 & 0 & -\Delta T v_k \sin \hat{\theta}_{k/k} \\ 0 & 1 & \Delta T v_k \cos \hat{\theta}_{k/k} \\ 0 & 0 & 1 \end{bmatrix} \quad (2.29)$$

On doit également calculer le bruit (ou l'incertitude) supplémentaire associé au mouvement qui s'est produit entre  $k$  et  $k+1$  :

$$Q = \begin{bmatrix} \delta_v^2 & 0 \\ 0 & \delta_\omega^2 \end{bmatrix} \quad (2.30)$$

$n_v, n_\omega$  sont des paramètres de bruit qui doivent être ajustés manuellement pour le robot utilisé.

On a besoin aussi de la matrice jacobienne du modèle de mouvement par rapport la commande  $U(k)$ :

$$G = \frac{\partial f(\hat{X}_{k/k}, U(k))}{\partial U(k)} = \begin{bmatrix} \Delta T \cos \hat{\theta}_{k/k} & 0 \\ \Delta T \sin \hat{\theta}_{k/k} & 0 \\ 0 & \Delta T \end{bmatrix} \quad (2.31)$$

On peut donc associer le bruit  $Q$  avec la multiplication  $GQG^T$  pour obtenir la covariance complète de mise à jour  $P_{k+1/k}$  :

$$P_{k+1/k} = FP(k)F^T + GQG^T \quad (2.32)$$

On remarque que la covariance augmente dans cette étape. Cela signifie que l'incertitude à propos de la posture du robot augmente aussi dans l'étape de prédiction. On peut réduire cette incertitude en ajoutant les informations des capteurs dans l'étape suivante.

#### 2.6.4.2 Observation

La mise à jour des observations peut affiner l'estimation de la posture du robot et diminuer l'incertitude.

Dans cette étape, on doit calculer la mise à jour des observation de la distance et de l'orientation des amers en fonction de la position estimée  $\hat{X}_{k+1/k}$  et du modèle de mesure  $h(X(k), M)$ . Le modèle des amers doit être transféré du repère global au repère lié au robot.

$$\hat{Z}_{j,k} = \begin{bmatrix} \hat{r}_j(k) \\ \hat{\phi}_j(k) \end{bmatrix} = \begin{bmatrix} \sqrt{(y_a^j - \hat{y}_{k+1/k})^2 + (x_a^j - \hat{x}_{k+1/k})^2} \\ a \tan 2(y_a^j - \hat{y}_{k+1/k}, x_a^j - \hat{x}_{k+1/k}) \end{bmatrix} \quad (2.33)$$

#### 2.6.4.3 Mise en correspondance

Dans cette étape, on doit calculer la meilleure identité entre les observations reçues des capteurs et les observations prédites.

Il faut associer les mesures à leur prédiction dans le repère local du robot. On calcule l'innovation  $C_k$  qui est égale :

$$C_k = Z_{i,k} - \hat{Z}_{j,k} \quad (2.34)$$

On doit également calculer la covariance associée à la mesure. Pour ce faire, la matrice jacobienne du modèle de mesure doit être calculée par rapport à la posture du robot :

$$H_k^j = \frac{\partial h}{\partial X} = \begin{bmatrix} -\frac{x_a^i - \hat{x}_{k+1/k}}{\hat{r}(k)} & -\frac{y_a^i - \hat{y}_{k+1/k}}{\hat{r}(k)} & 0 \\ \frac{y_a^i - \hat{y}_{k+1/k}}{\hat{r}^2(k)} & -\frac{x_a^i - \hat{x}_{k+1/k}}{\hat{r}^2(k)} & -1 \end{bmatrix} \quad (2.35)$$

La covariance de mesure est calculée en multipliant la covariance estimée antérieure par le modèle de mesure jacobienne et en ajoutant un bruit supplémentaire pour la nouvelle mesure.

$$E_k = H_k^j P_{k+1/k} (H_k^j)^T + R \quad (2.36)$$

Le bruit supplémentaire  $R(k)$  est une matrice constante pour toutes les mesures. C'est une matrice diagonale de dimensions  $2 \times 2$  avec les paramètres  $r_v$  et  $\theta_v$  qui représentent le bruit correspondant respectivement à l'éloignement et à l'orientation de l'amer.

L'innovation et la matrice de covariance peuvent être utilisées pour valider les mesures avant leur intégration dans le calcul des états estimés.

#### 2.6.4.4 Mise à jour

La covariance est alors utilisée pour calculer le gain de kalman  $K_k$  qui détermine à quel point les informations du capteur doivent être pondérées lors de la correction de la nouvelle posture estimée.

$$K_k = P_{k+1/k} (H_k)^T (E_k)^{-1} \quad (2.37)$$

Le gain de Kalman est utilisé pour mettre à jour la moyenne et la covariance estimées  $\hat{X}_{k+1/k}$  et  $P_{k+1/k}$  comme suit:

$$\hat{X}_{k+1/k+1} = \hat{X}_{k+1/k} + K_k C_k \quad (2.38)$$

$$P_{k+1/k+1} = P_{k+1/k} - K_k E_k K_k^T \quad (2.39)$$

La correction de posture estimée dépend de  $K_k$  et de la différence entre la mesure prédite  $\hat{Z}_{j,k}$  et la mesure réelle  $Z_{i,k}$ . Un écart plus important entre la mesure prédite et la mesure réelle entraînera une correction plus importante.



## 2.7 Travaux de simulation

Dans les travaux de simulation, on considère que le robot se déplace avec une vitesse de translation  $v=0.4m/s$  et une vitesse angulaire  $\omega=0.1rad/s$ . La période d'échantillonnage  $dT=0.1s$ . Le filtre de Kalman étendu a une estimation initiale:  $\hat{X}=(\hat{x},\hat{y},\hat{\theta})=[0\ 0\ 0]$  avec une matrice de covariance  $P_{0/0}=[10^{-4}\ 10^{-4}\ 10^{-4}]$  (Matrice diagonale à trois dimensions). La position initiale du robot est considérée comme :  $X=(x,y,\theta)=[0\ 0\ 0]$

Les bruits sur les vitesses de commandes linéaire et angulaire :  $\delta_v=0.01m/s$ ,  $\delta_\omega=0.01rad/s$ . Les bruits d'observation du capteur :  $r_v=0.1\ m$ ,  $\theta_v=0.1\ rad$ .

Les matrices de covariance des bruits de commande et de mesure sont respectivement:

$$Q = \begin{bmatrix} \delta_v^2 & 0 \\ 0 & \delta_\omega^2 \end{bmatrix}, R = \begin{bmatrix} r_v^2 & 0 \\ 0 & \theta_v^2 \end{bmatrix}$$

Dans un premier cas, on s'intéresse à un environnement structuré par 20 amers. Une trajectoire circulaire est proposée et l'application du filtre de Kalman étendu permet d'avoir les résultats suivants:

### Premier cas : Environnement constitué de 20 amers

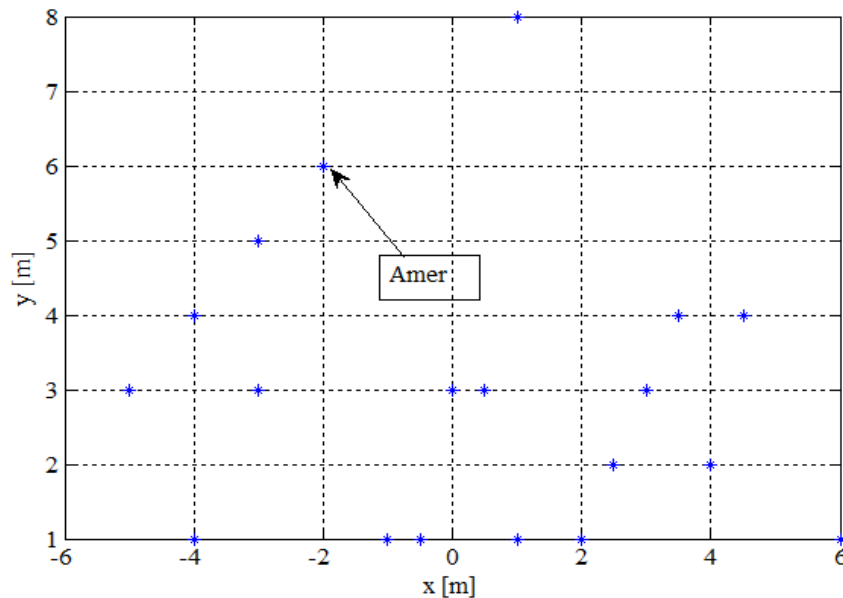


Figure 2.11 : Environnement constitué de 20 amers

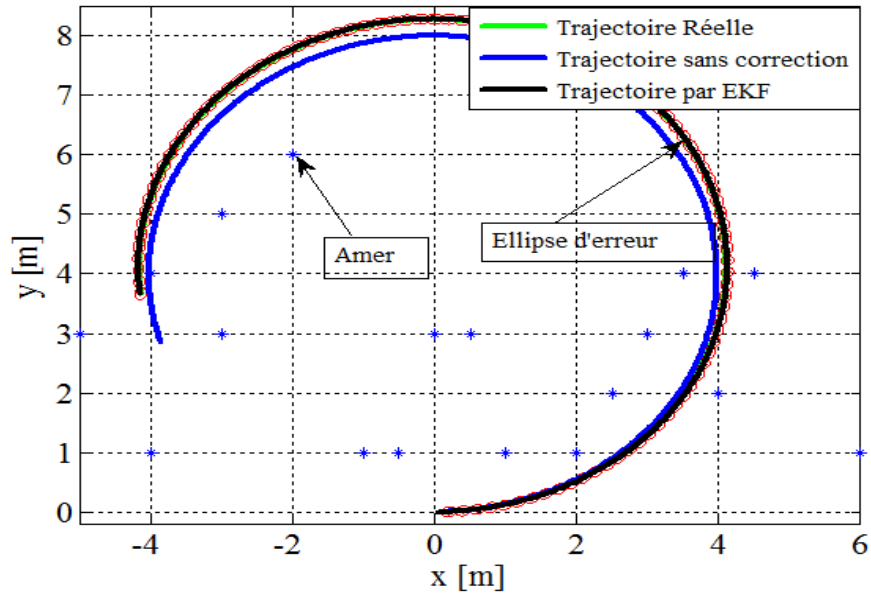


Figure 2.12 : Localisation obtenue après 500 itérations avec 20 amers

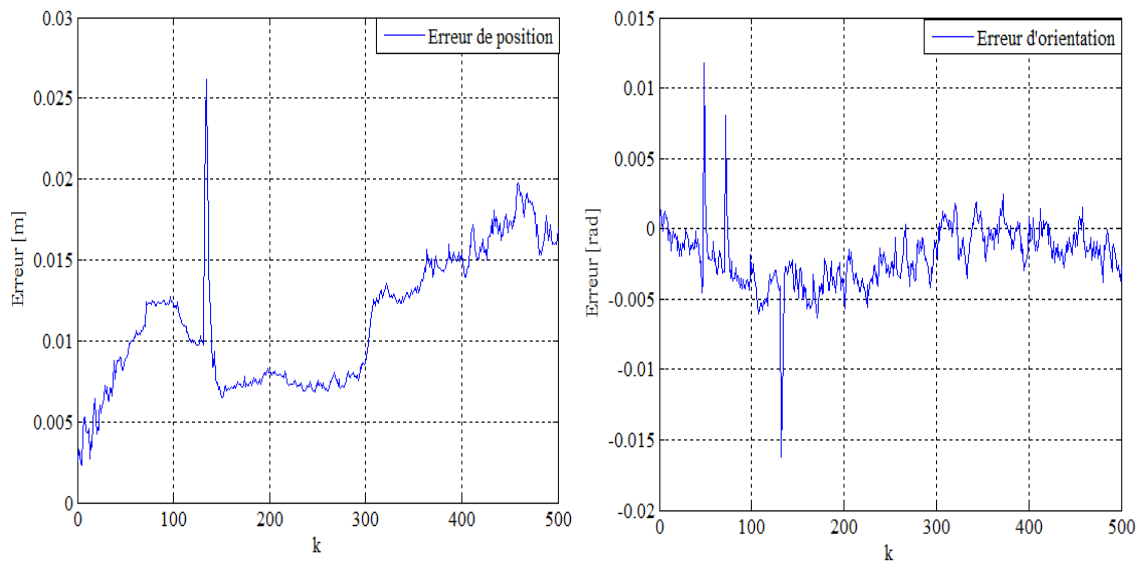


Figure 2.13 : Erreurs de position et d'orientation après 500 itérations

Une nouvelle trajectoire est simulée. On considère que le robot se déplace avec des vitesses différentes.

Les bruits sur les vitesses de commande linéaire et angulaire :  $\sigma_v=0.2\text{m/s}$ ,  $\sigma_\omega=0.2\text{rad/s}$ .  
 Les bruits d'observation du capteur sont:  $r_v=0.01\text{ m}$ ,  $\theta_v=0.01\text{ rad}$ .

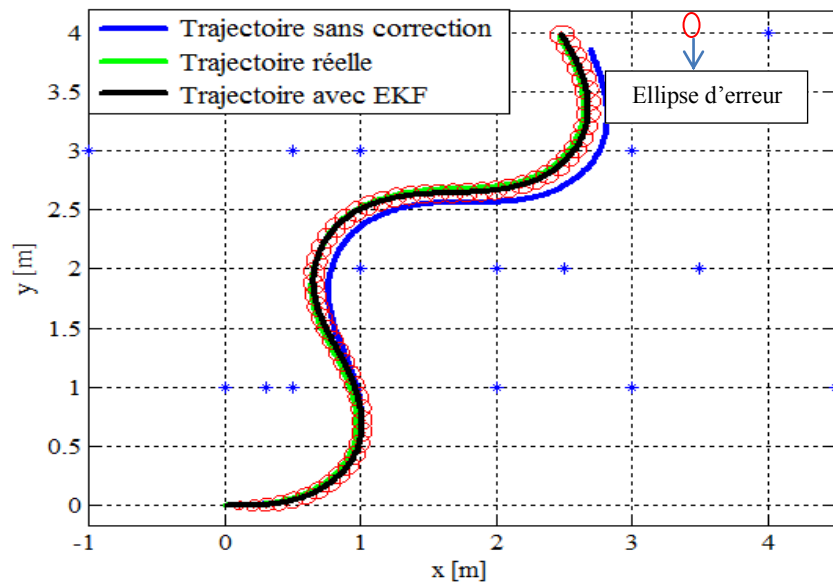


Figure 2.14 : Localisation obtenue après 300 itérations avec 20 amers

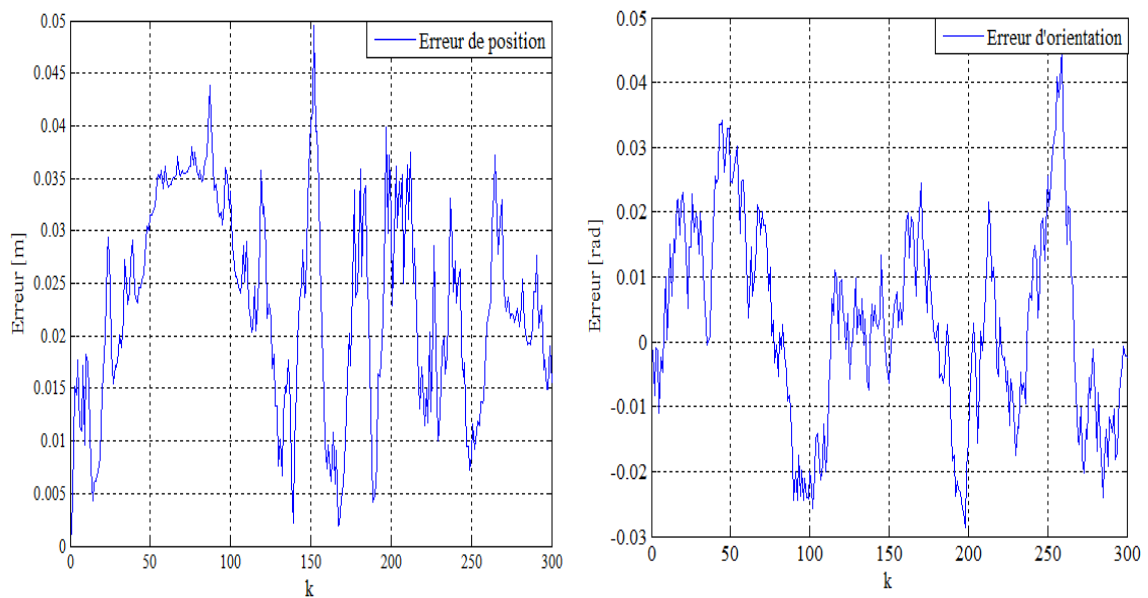


Figure 2.15 : Erreurs de position et d'orientation après 300 itérations

On peut remarquer que l'erreur de position et d'orientation utilisant le filtre de kalman étendu est très petite et la localisation du robot est précise sur les deux trajectoires considérées.

**Deuxième cas : Environnement constitué de 24 amers**

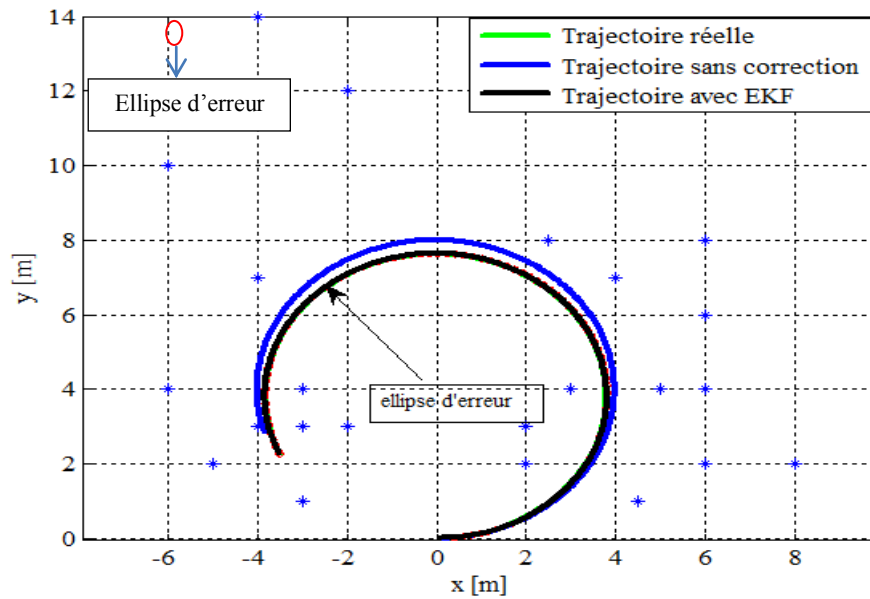


Figure 2.16 : Localisation obtenue après 500 itérations avec 24 amers

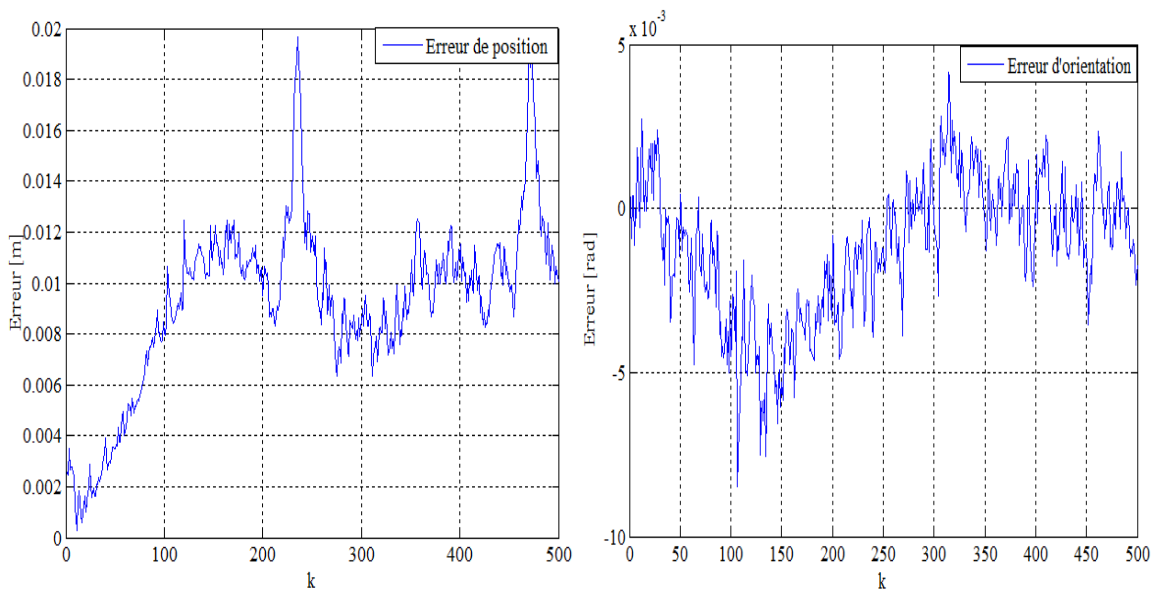


Figure 2.17 : Erreurs de position et d'orientation après 500 itérations avec 24 amers

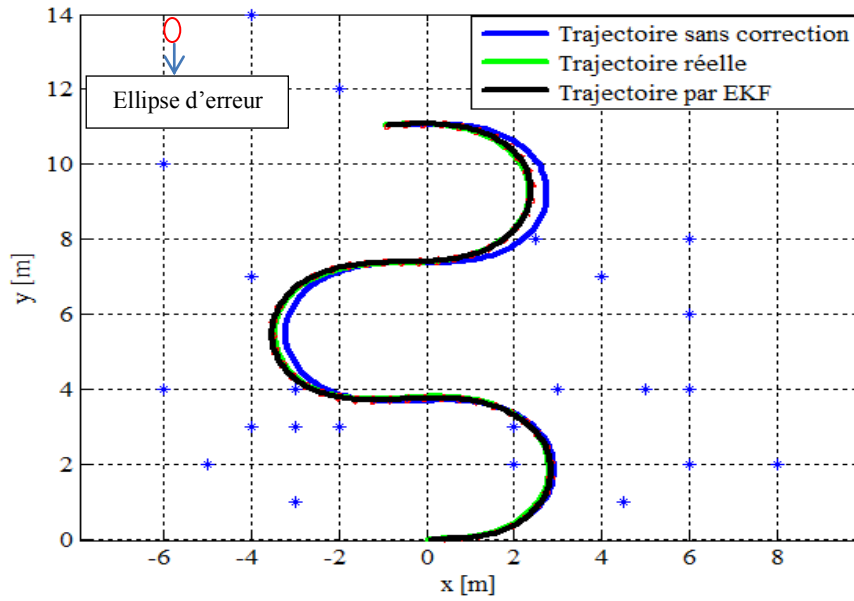


Figure 2.18 : Localisation obtenue après 300 itérations avec 24 amers

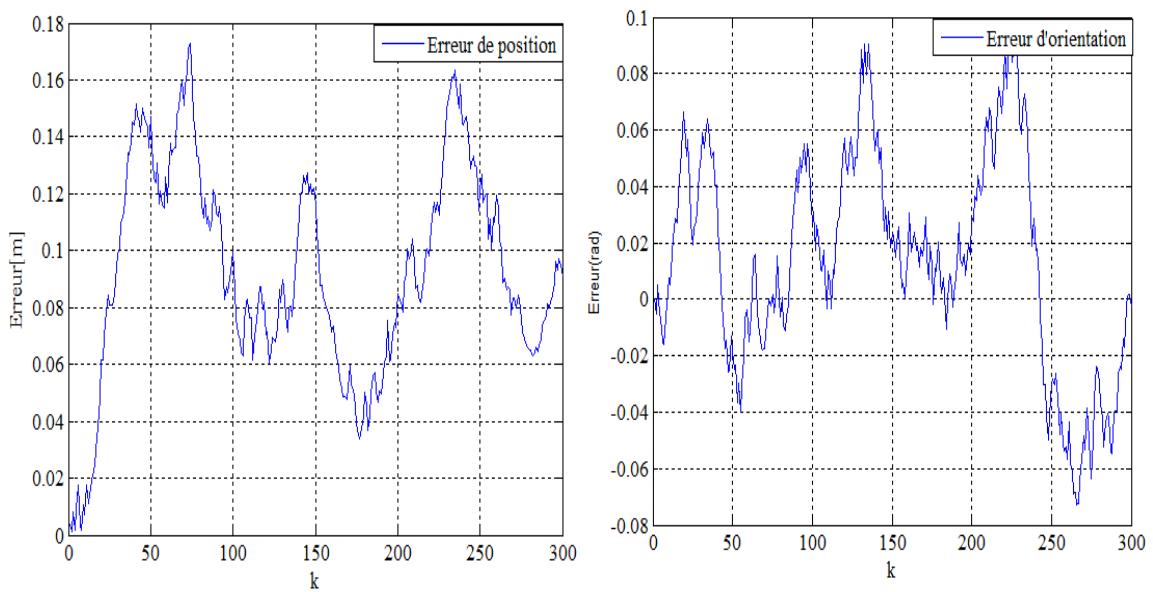


Figure 2.19 : Erreurs de position et d'orientation après 300 itérations avec 24 amers

On peut également remarquer que les résultats obtenus dans ce cas sont meilleurs que ceux obtenus dans le cas précédent.

On s'intéresse maintenant à un environnement structuré par 30 amers.

Troisième cas : Environnement constitué de 30 amers

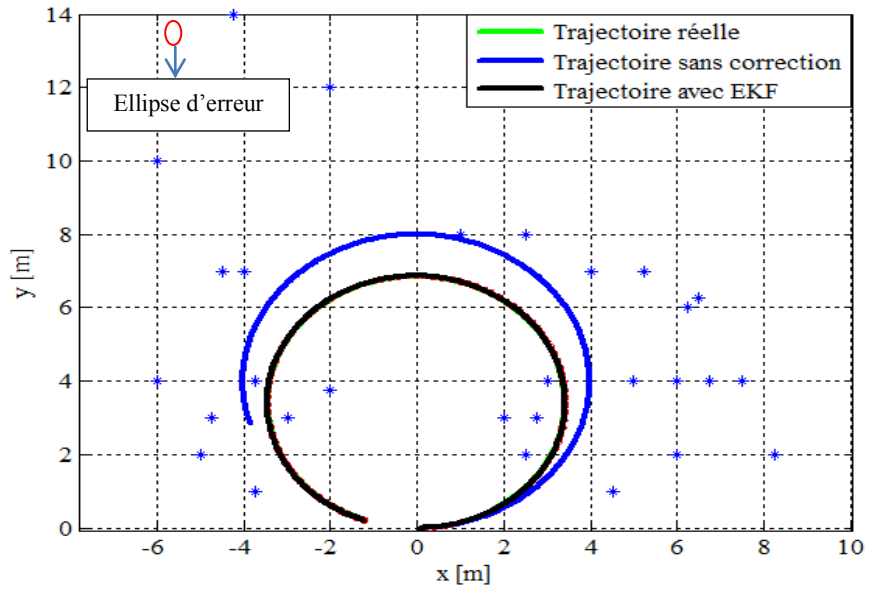


Figure 2.20 : Localisation obtenue après 500 itérations avec 30 amers

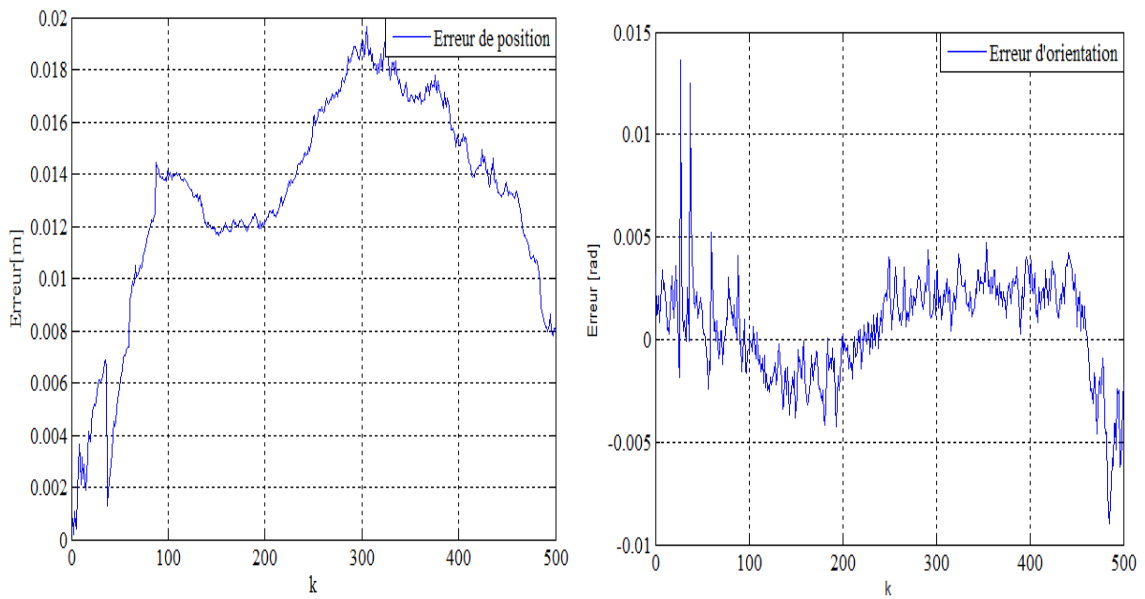


Figure 2.21 : Erreurs de position et d'orientation après 500 itérations avec 30 amers

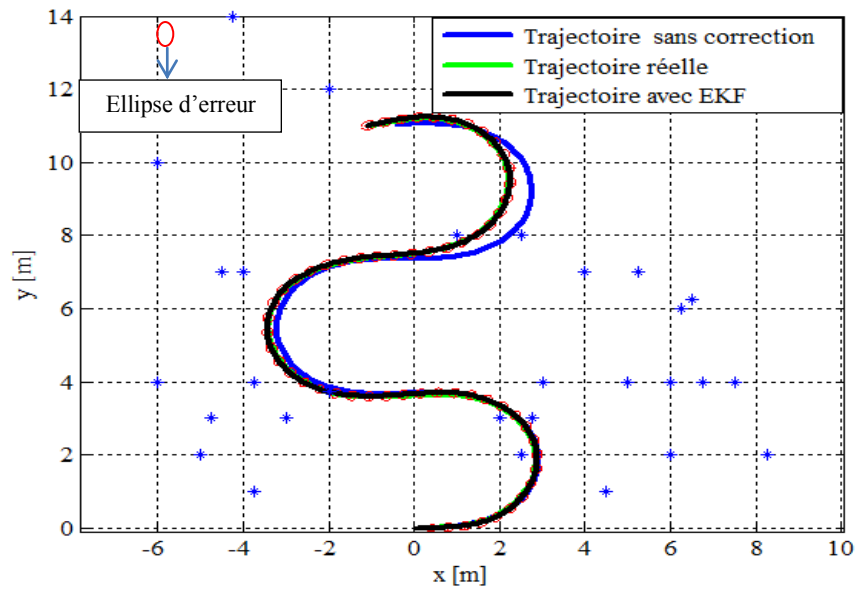


Figure 2.22 : Localisation obtenue après 300 itérations avec 30 amers

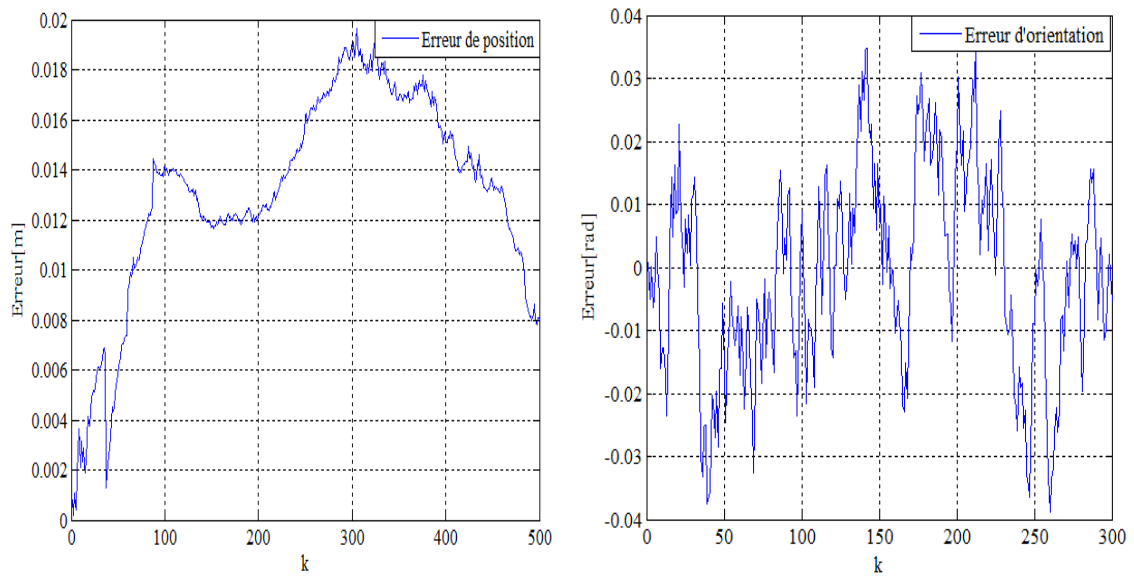


Figure 2.23 : Erreurs de position et d'orientation après 300 itérations avec 30 amers

Les résultats obtenus dans ces cas d'environnements montrent que plus le nombre d'amers est grand, plus la précision de la localisation est meilleure.

## 2.8 Conclusion

Dans ce chapitre, on a présenté les différents types de localisation et on a proposé une méthode de localisation par filtre de Kalman étendu utilisant différentes structures d'environnements.

Lors des travaux de simulation, on a constaté que les bruits du capteur et de la commande n'influent pas sur le bon comportement du filtre de kalman étendu.

D'après les résultats obtenus, on a constaté que la précision de la localisation par FKE dépend du nombre des amers utilisés. Ce résultat est confirmé par les différents cas étudiés. Dans le premier cas, on a utilisé un environnement structuré par 20 amers et puis un environnement structuré par 24 amers et finalement un environnement structuré par 30 amers. Les erreurs de position et d'orientation dans les différents cas proposés deviennent meilleurs à chaque fois plus que le nombre d'amers augmente. Les ellipsoïdes d'erreurs de la trajectoire contrôlée par FKE diminuent à chaque simulation.



## **3<sup>ER</sup> CHAPITRE**

# **ELABORATION D'ALGORITHMES ROBUSTES PAR MODE GLISSANT FLOU AVEC OPTIMISATION PAR PSO**

---

## **CHAPITRE 3**

### **Elaboration d'algorithmes robustes par mode glissant flou avec optimisation par PSO**

#### **3.1 Introduction**

Le suivi de trajectoires représente une tâche primordiale pour la navigation des robots autonomes et on s'intéresse à avoir une bonne stabilité du mouvement lors de ce suivi [109,110]. Le contrôle par mode glissant (SMC) propose des solutions pour la conception des lois de commande et pour la stabilité des systèmes perturbés [111,112]. Le mode de glissement dépend de la surface de glissement et garantit la stabilité asymptotique du système global. Une méthode connue sous le nom de mode glissant conventionnel utilise une fonction discontinue qui produit des oscillations aux fréquences élevées appelées broutement. Pour pallier à ce phénomène, plusieurs auteurs utilisent le mode glissant d'ordre supérieur [113]. Certains auteurs proposent d'autres méthodes pour minimiser ce phénomène en utilisant le contrôle traditionnel des modes de glissement [114, 115, 116]. Un mode de glissement standard est présenté pour être une approche de contrôle efficace dans la stabilisation des systèmes non linéaires [117,118]. L'utilisation du mode glissant réside dans ses bons résultats et dans sa simplicité dans l'application des lois de contrôle [119,120].

Un algorithme de contrôle, basé sur les erreurs de postures du robot mobile, utilisant la logique floue est proposé dans [121]. Ce contrôleur génère des signaux de correction appropriés pour les vitesses des moteurs gauche et droit. La stratégie de contrôle repose sur un bloc FLC qui tient compte des erreurs de posture. De même des approches de suivi de trajectoires utilisant la logique floue pour réduire les effets de broutement sont proposées dans [122,123]. Un contrôleur flou proposé dans [124] permet d'ajuster les paramètres de la surface de glissement et d'accélérer le système pour atteindre la surface de glissement.

Un autre travail adapte un contrôleur adaptatif flou en mode glissant pour les systèmes non linéaires afin d'arriver à une convergence rapide en présence de perturbations externes [125,126].

Dans ce chapitre, on présente la première contribution dans cette thèse qui est principalement la conception de nouveaux contrôleurs de suivi de trajectoires pour les robots mobiles non holonomes. Le travail comprend le développement et l'élaboration de deux contrôleurs cinématique et dynamique, basés sur la logique floue et la commande globale en mode glissant (GFTSM) avec une optimisation par la méthode d'essaim de particules (PSO).

La loi de contrôle cinématique basée sur la commande globale en mode glissant (GFTSM) est conçue pour diminuer les perturbations de la vitesse angulaire du modèle cinématique. Par la suite, un ensemble flou est introduit en association avec le mode glissant pour la vitesse linéaire du robot afin de réduire les perturbations et d'atténuer le phénomène de broutement. La sélection des paramètres par la logique floue permet d'éliminer l'effet des perturbations et tendre vers zéro l'état des erreurs de posture du robot. Par la suite, un contrôleur dynamique dont les paramètres sont générés par un algorithme PSO, est proposé pour avoir des couples optimaux.

Des travaux de simulation sont réalisés et une comparaison entre les deux contrôleurs, avec et sans PSO, est présentée afin d'illustrer les performances des deux algorithmes.

## **3.2 Mode de glissement terminal rapide (FTSM)**

Le mode de glissement terminal rapide (FTSM) a pour objectif d'ajouter des fonctions non linéaires dans la conception de la surface de glissement et d'accélérer l'état du système pour rejoindre la surface dans un temps fini tout en éliminant le phénomène de broutement. L'objectif de cette commande est la convergence de l'erreur de suivi de la trajectoire dans le but de renforcer davantage la stabilité du système tout en se basant sur la théorie de Lyapunov.

### **3.2.1 Conception de la surface de glissement par la méthode FTSM**

#### **3.2.1.1 Méthode traditionnelle (FTSM)**

Une surface de glissement est proposée dans [127]:

$$s = \dot{x} + \beta x^{q/p} = 0 \quad (3.1)$$

Avec  $x \in \mathfrak{R}^n$  et  $\alpha, \beta > 0, p, q (p > q)$  des paramètres positifs.

D'où  $\frac{dx}{dt} = -\beta x^{q/p}$ . Par intégration, on a le temps d'arrivée :

$$t_s = \frac{p}{\beta(p-q)} |x(0)|^{\frac{p-q}{p}} \quad (3.2)$$

Le point d'équilibre  $x=0$  s'appelle aussi attracteur terminal et la vitesse de convergence est améliorée par le terme de non linéarité  $\beta x^{q/p}$ . Le temps de convergence de l'algorithme FTSM n'est pas optimal à cause de la vitesse de convergence de l'équation (3.2) lorsque  $p = q$  et quand l'état du système s'approche de l'équilibre.

### 3.2.1.2 Méthode globale (GFTSM)

Cette nouvelle conception de la surface de glissement est donnée dans [128] :

$$s = \dot{x} + \alpha x + \beta x^{q/p} = 0 \quad (3.3)$$

où  $x \in \mathfrak{R}^n$  et  $\alpha, \beta > 0$  et  $p, q (p > q)$  des paramètres positifs.

D'après l'équation (3.3), on a:

$$\dot{x} = -\alpha x - \beta x^{q/p} \quad (3.4)$$

Après intégration, le temps d'arrivée est donné par:

$$t_s = \frac{p}{\alpha(p-q)} \ln \frac{\alpha x(0)^{\frac{p-q}{p}} + \beta}{\beta} \quad (3.5)$$

Par un choix convenable des valeurs des paramètres  $\alpha, \beta, p, q$  le système peut atteindre la surface dans un temps  $t_s$  optimal. Lorsque l'état du système est loin de l'origine, le temps de convergence est décidé par l'attracteur terminal  $\dot{x} = \beta x^{q/p}$  et quand l'état du

système est proche de l'origine  $x=0$ , le temps de convergence est décidé par l'équation  $\dot{x} = -\alpha x$ .

### 3.3 Généralités sur la logique floue type-1

Lotfi Zadeh a proposé la théorie des ensembles flous en 1965. Un ensemble flou est une classe d'objets avec des grades d'appartenance de 0 à 1 [129]. Les ensembles flous sont utilisés pour décrire le flou en fonction des degrés d'appartenance ainsi qu'ils peuvent être utilisés dans de nombreuses situations réelles avec des termes linguistiques. La structure de base d'un système flou de type 1 se divise en trois parties principales comme indiqué sur la figure (3.1).

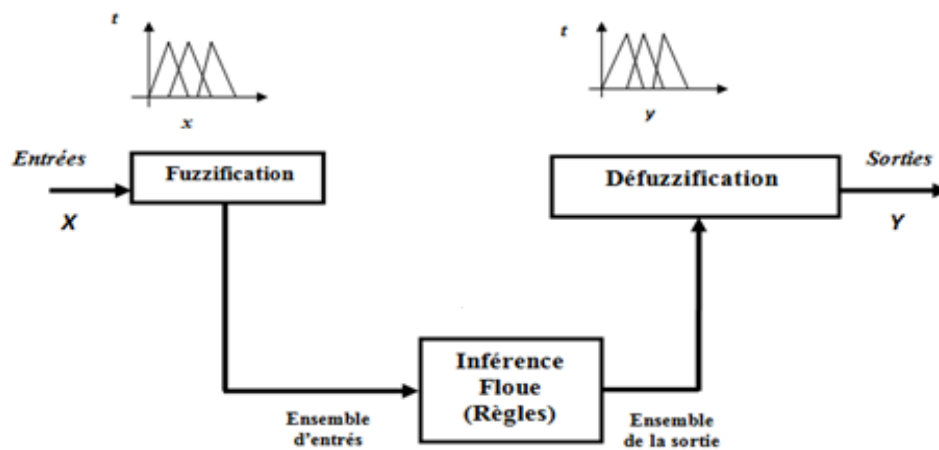


Figure 3.1 : Diagramme de la logique floue type 1

#### 3.3.1 Fuzzification

Dans l'étape de la fuzzification, les variables réelles d'entrée et de sortie doivent être transformées en une nature numérique en termes linguistiques dans le but d'effectuer une logique d'inférence. Ces termes seront utilisés pour mettre les règles d'inférence.

Les fonctions d'appartenances qui présentent ces variables ne sont pas d'une forme précise, pour cela il existe plusieurs fonctions d'appartenance qui peuvent être utilisées dans la logique floue type 1.

La représentation mathématique de certaines fonctions d'appartenance conventionnelles est présentée ci-dessous [130].

- **Forme triangulaire :**

La fonction d'appartenance triangulaire est définie par ses paramètres  $[a, b, c]$  tel que :

$$\mu_A(x) = \begin{cases} \frac{x-a}{b-a} & a \leq x \leq b \\ \frac{c-x}{c-b} & b \leq x \leq c \\ 0 & x \leq a \cup x \geq c \end{cases} \quad (3.6)$$

La forme de cette fonction est illustrée sur la figure (3.2) (a).

- **Forme Trapézoïdale**

La fonction d'appartenance trapézoïdale est définie par les paramètres  $[a, b, c, d]$  tel que :

$$\mu_A(x) = \begin{cases} \frac{x-a}{b-a} & a \leq x \leq b \\ 1 & b \leq x \leq c \\ \frac{d-x}{d-c} & c \leq x \leq d \\ 0 & x \leq a \cup x \geq d \end{cases} \quad (3.7)$$

La forme de cette fonction est illustrée sur la figure (3.2) (b).

- **Forme gaussienne:**

La fonction d'appartenance gaussienne est définie par ses paramètres  $[c, \sigma]$  tel que :

$$\mu_A(x) = e^{-\frac{1}{2} \left( \frac{x-c}{\sigma} \right)^2} \quad (3.8)$$

La forme de cette fonction est illustrée à la figure (3.2) (c).

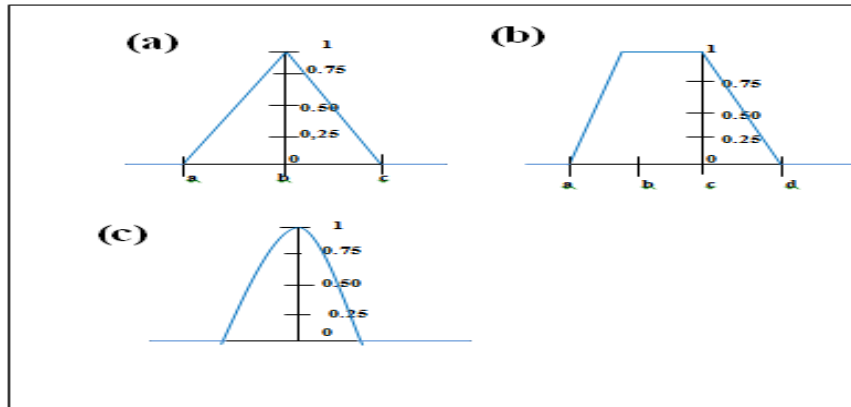


Figure 3.2 : Formes de la fonction d'appartenance

### 3.3.2 Règles de la logique floue

La proposition floue est un concept très important en logique floue. Elle représente des énoncés comme « $V$  est grand» où «grand» est un terme linguistique défini par un ensemble flou des variables  $V$ . Les étiquettes floues (linguistiques) sont également appelées constantes floues, termes flous ou notions floues. Les propositions floues connectent des variables avec des étiquettes linguistiques définies pour ces variables, les relations floues peuvent également être assemblées à partir de connaissances linguistiques, exprimées comme des règles si-alors. Une règle floue peut être définie comme une instruction conditionnelle sous la forme:

Si  $x$  est  $A$  Alors  $y$  est  $B$

où  $x$  est une variable d'entrée et  $A$  un ensemble flou défini dans  $X$ , d'autre part  $y$  est une variable de sortie et  $B$  est son ensemble flou correspondant défini dans  $Y$ . Si la sortie dépend de plusieurs variables, il doit y avoir une relation entre les entrées.

### 3.3.3 Méthode d'inférence floue

L'inférence floue est définie comme le processus de transfert d'une sortie associée d'une entrée donnée à l'aide des degrés d'appartenance. Les méthodes d'inférence floue les plus courantes sont celles de Mamdani et Takagi-Sugeno.

#### 3.3.3.1 Implication de Mamdani

Mamdani a proposé un contrôleur basé sur des règles linguistiques et des ensembles flous. C'est l'une des méthodes d'inférence les plus utilisées en logique floue. Une règle du modèle d'inférence Mamdani de type 1 est exprimée comme suit:

$$\text{Si } x_1 \text{ est } A_1^1 \text{ et } x_2 \text{ est } A_2^1, \text{ ALORS } y \text{ est } B \quad (3.9)$$

où  $A$  et  $B$  sont des ensembles flous.

Son modèle d'inférence pourrait être basé sur la composition min et max selon l'ensemble de règles. Généralement, le modèle d'inférence peut être réalisé avec n'importe quelle composition de T-Normes floues et de T-Conormes. Généralement, la composition floue peut être exprimée par:

$$\mu_{y_{R_1 \circ R_2}}(x, z) = S(T(\mu_{R_1}(x, y), \mu_{R_2}(y, z))) \quad (3.10)$$

où  $S$  est appliqué au résultat de toutes les T-Normes.

### 3.3.3.2 Implication de Takagi-Sugeno et Kang(TSK)

L'inférence Sugeno est différente de Mamdani parce que la défuzzification utilise une équation polynomiale. Ainsi, les règles Sugeno si-alors sont décrites sous une forme différente. Les conséquences de ces règles peuvent être des fonctions linéaires. Takagi Sugeno et Kang en 1985 ont proposé un modèle mathématique pour les systèmes d'inférence basés sur des polynômes, appelé modèle d'inférence Takagi-Sugeno-Kang (TSK). En règle générale, une règle de type 1 de TSK est exprimée comme suit:

$$\text{Si } x_1 \text{ est } A_1^1 \text{ et } x_2 \text{ est } A_2^1, \text{ ALORS } y = f(x_1, x_2) \quad (3.11)$$

Pour la logique floue de type 1, les forces de tir sont liées aux polynômes correspondants. Les conséquences  $y_i$  sont des polynômes où  $i = 1, 2, \dots, M$  et  $j = 0, 1, \dots, p$ .  $M$  est le nombre total de points de tir et  $p$  est le nombre total de variables d'entrée nettes [130].

$$F^i = T(\mu_{f_1}(x_1), \mu_{f_1}(x_1), \dots, \mu_{f_p}(x_p)) \quad (3.12)$$

Chaque force de tir peut être calculée comme suit:

$$y^i = \sum_{k=1}^p (a_k^i x_k^2 + b_k^i x_k + c_k^i) \quad (3.13)$$

La sortie TSK est donc:



$$y_{TSK} = \sum_{y^1 \in Y} \dots \sum_{y^M \in Y} \sum_{f^1} \dots \sum_{f^M} \frac{\sum_{i=1}^M f^i y^i}{\sum_{i=1}^M f^i} \quad (3.14)$$

### 3.3.4 Défuzzification

C'est la dernière étape d'une inférence de logique floue. La défuzzification associe les sorties en un seul numéro précis, il existe plusieurs méthodes pour ce faire et les plus communes sont données ci-dessous:

**Centroïde :** C'est la méthode de défuzzification la plus utilisée dans la logique floue de type 1 également connue sous le nom de Centroïde of Area (COA). Pour cette méthode de défuzzification, la forme du Type 1 de jeu inféré  $\mu_B(y)$  est nécessaire. Par conséquent, l'expression peut être exprimée par:

$$y = \frac{\sum_{i=1}^N y_i \mu_B(y_i)}{\sum_{i=1}^N \mu_B(y_i)} \quad (3.15)$$

**Bissecteur :**

Cette méthode de défuzzification recherche la condition suivante:

$$\sum_{i=\alpha}^{y'} y \mu_B(y) = \sum_{i=y'}^{\beta} y \mu_B(y) \quad (3.16)$$

où  $\alpha = \min(Y | \mu_B(y) \neq 0)$ ,  $\beta = \max(Y | \mu_B(y) \neq 0)$ , tel que  $\alpha, \beta \in Y$

Cela signifie qu'une ligne verticale  $y=y'$  divise la région entre  $[\alpha, \beta]$  et  $[0, \mu_B(y)]$  entre deux régions avec la même zone.

### 3.4 Commande cinématique par mode de glissement flou

#### 3.4.1 Modélisation cinématique du robot mobile

Le robot mobile utilisé dans ce travail est donné sur la figure (3.3).

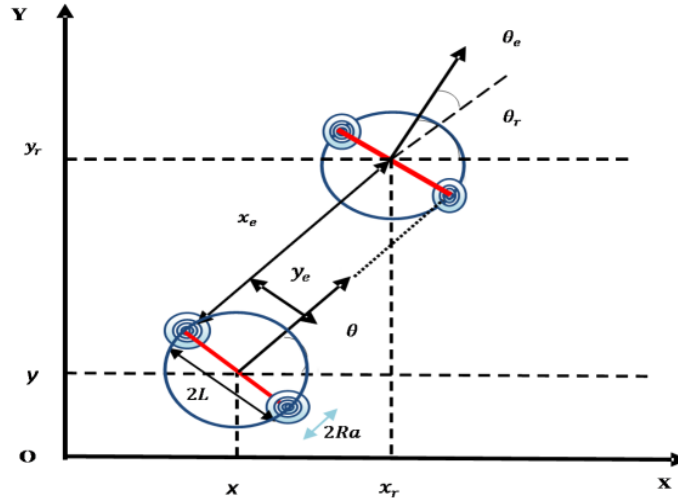


Figure 3.3 : Diagramme du robot mobile

Les équations suivantes décrivent le mouvement du robot:

$$v = R_a \left( \frac{\dot{\varphi}_r + \dot{\varphi}_l}{2} \right) \quad (3.17)$$

$$\omega = \frac{R_a}{2L} (\dot{\varphi}_r - \dot{\varphi}_l) \quad (3.18)$$

Avec  $\dot{\varphi}_r$  et  $\dot{\varphi}_l$  les vitesses angulaires des roues droite et gauche.  $\theta$  est l'angle d'orientation du robot mobile,  $v$  la vitesse linéaire et  $R_a$  le rayon de la roue,  $\omega$  la vitesse angulaire et  $2L$  la distance séparant les deux roues.

La posture du robot est donnée par le vecteur réel  $\rho = (x \ y \ \theta)^T$  et le vecteur de contrôle est donné par  $\gamma = (v \ \omega)^T$ . Le modèle cinématique perturbé du robot mobile est donné par [131].

$$\dot{\rho} = \begin{pmatrix} \dot{x} \\ \dot{y} \\ \dot{\theta} \end{pmatrix} = \begin{pmatrix} \cos \theta & 0 \\ \sin \theta & 0 \\ 0 & 1 \end{pmatrix} (\gamma + D) \quad (3.19)$$

où  $D$  représente la perturbation inconnue bornée et exprimée par :

$$D = [dv \quad d\omega]^T \quad (3.20)$$

avec  $|dv| < \zeta_v$  et  $|d\omega| < \zeta_\omega$

$dv$  et  $d\omega$  représentent les perturbations des vitesses linéaires et angulaires et  $\zeta_v$  et  $\zeta_\omega$  sont des constantes limitées positives.

En ce qui concerne le suivi de trajectoires, la posture de référence  $\rho_r = (x_r \quad y_r \quad \theta_r)^T$  du robot mobile et la vitesse désirée  $\gamma_r = (v_r \quad \omega_r)^T$  sont utilisées.

L'erreur de posture  $\rho_e$  du robot est représentée par le système suivant [132]:

$$\rho_e = \begin{bmatrix} x_e \\ y_e \\ \theta_e \end{bmatrix} = \begin{bmatrix} \cos \theta & \sin \theta & 0 \\ -\sin \theta & \cos \theta & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x_r - x \\ y_r - y \\ \theta_r - \theta \end{bmatrix} \quad (3.21)$$

En introduisant les contraintes non holonomes de l'équation (3.22) dans le système (3.21), l'erreur de vitesse sans perturbations est définie par l'équation (3.23) [133] :

$$-\dot{x} \sin \theta + \dot{y} \cos \theta = 0 \quad (3.22)$$

$$\dot{\rho}_e = \begin{bmatrix} \dot{x}_e \\ \dot{y}_e \\ \dot{\theta}_e \end{bmatrix} = \begin{bmatrix} y_e \omega + v_r \cos \theta_e - v \\ -x_e \omega + v_r \sin \theta_e \\ \omega_r - \omega \end{bmatrix} \quad (3.23)$$

Considérant les perturbations sur les vitesses  $v$  et  $\omega$ , l'équation (3.23) devient comme suit:

$$\begin{aligned} \dot{\rho}_e &= \begin{bmatrix} \dot{x}_e \\ \dot{y}_e \\ \dot{\theta}_e \end{bmatrix} = \begin{bmatrix} v_r \cos \theta_e \\ v_r \sin \theta_e \\ \omega_r \end{bmatrix} + \begin{bmatrix} -1 & y_e \\ 0 & -x_e \\ 0 & -1 \end{bmatrix} \begin{bmatrix} v + dv \\ \omega + d\omega \end{bmatrix} \\ &= \begin{bmatrix} v_r \cos \theta_e - (v + dv) + y_e (\omega + d\omega) \\ v_r \sin \theta_e - x_e (\omega + d\omega) \\ \omega_r - (\omega + d\omega) \end{bmatrix} \end{aligned} \quad (3.24)$$

L'erreur de suivi de trajectoire est bornée et converge asymptotiquement vers zéro lorsque  $t \rightarrow \infty$ , en fonction du vecteur d'entrée  $\gamma = (v \ \omega)^T$  en tenant compte des contraintes suivantes:  $|v| \leq v_{max}$  et  $|\omega| \leq \omega_{max}$

Le but de ce travail est de concevoir un contrôleur tel que le robot mobile converge asymptotiquement vers la trajectoire souhaitée. L'équation (3.19) du modèle du robot est utilisée pour élaborer une loi de contrôle basée sur le mode glissant et la logique floue. Cet algorithme de contrôle est appliqué pour faire converger asymptotiquement l'erreur de suivi tout en éliminant l'effet des perturbations qui influent sur les vitesses linéaires et angulaires tel que :

$$\lim_{t \rightarrow \infty} \rho_e = \lim_{t \rightarrow \infty} \|\rho_r(t) - \rho(t)\| = 0 \quad (3.25)$$

### 3.4.2 Conception de la loi de commande

Le processus de commande par mode de glissement peut être divisé en deux étapes:

- Choix d'une surface de glissement ou d'une fonction de commutation appropriée.
- Détermination d'une loi de commande discontinue sur la surface de glissement.

#### 3.4.2.1 Choix de la fonction de commutation

Le modèle cinématique du robot représente deux entrées non linéaires, cependant la fonction de commutation est difficile à choisir, alors pour simplifier le problème tout en se basant sur l'équation (3.4), nous prenons  $\theta_e$  comme la première surface de glissement alors :

$$\dot{\theta}_e = -\alpha \theta_e - \beta \theta_e^{q/p} \quad (3.26)$$

Lorsque  $\theta_e$  converge vers zéro le modèle (3.24) devient :

$$\dot{x}_e = \omega_r y_e + v_r - v - dv \quad (3.27)$$

$$\dot{y}_e = -\omega_r x_e \quad (3.28)$$

La fonction de commutation peut être représentée comme suit :

$$s_2 = x_e - y_e \quad (3.29)$$

Dans le but de faire converger  $s_2$  vers zéro,  $x_e$  converge vers  $y_e$  et tous les deux convergent vers zéro. Les surfaces de glissement peuvent être mises sous la forme suivante :

$$\begin{bmatrix} s_1 \\ s_2 \end{bmatrix} = \begin{bmatrix} \dot{\theta}_e + \alpha \theta_e + \beta \theta_e^{q/p} \\ x_e - y_e \end{bmatrix} \quad (3.30)$$

### 3.4.2.2 Loi de commande

D'après l'équation (3.26), on obtient la loi de contrôle appliquée sur la vitesse angulaire du robot.

$$\omega_c = \omega_r + \beta \theta_e^{q/p} + \alpha \theta_e \quad (3.31)$$

La loi de commande par mode de glissement basée sur l'approche de la loi d'arrivée avec une puissante vitesse d'arrivée est appliquée sur la vitesse linéaire du robot.

$$\dot{s}_2 = -G(t) \text{Sign}(s_2) - ks_2 \quad (3.32)$$

Avec,  $k > 0$

Cette loi d'arrivée augmente la vitesse de convergence lorsque l'état du système est loin de l'origine et la diminue lorsque l'état est proche de l'équilibre, le résultat est plus rapide et le phénomène de broutement est négligeable.

En utilisant l'équation (3.27), (3.28) et en dérivant l'équation (3.29), nous obtenons:

$$\dot{s}_2 = \dot{x}_e - \dot{y}_e = \omega_r y_e + \omega_r x_e + v_r - v_c \quad (3.33)$$

La loi de commande appliquée sur le robot est donnée comme suit:

$$v_c = v_r + \omega_r x_e + \omega_r y_e + G(t) \frac{s_2}{|s_2| + \delta} + ks_2 \quad (3.34)$$

où

$$G(t) = \max\{dv(t)\} + \eta \quad (3.35)$$

Dans l'équation (3.34)  $G(t)$  est utilisé pour compenser l'effet de la perturbation  $dv(t)$  et assure la condition existante du mode de glissement et permet d'éviter le phénomène de broutement.

### 3.4.2.3 Choix de la fonction de Lyapunov

Le choix de fonction de Lyapunov est donné comme suit :

$$T_l = \frac{1}{2} s_2^2 \quad (3.36)$$

La dérivée de cette fonction est donnée par:

$$\dot{T}_l = s_2 \dot{s}_2 = s_2 (\omega_r x_e + \omega_r y_e + v_r - v_c - dv) \quad (3.37)$$

$$\begin{aligned} \dot{T}_l &= s_2 (-dv - G(t) \frac{s_2}{|s_2| + \delta} - ks_2) \\ &= -s_2 dv - G(t) \frac{s_2^2}{|s_2| + \delta} - ks_2^2 \leq -\eta |s_2| - ks_2^2 \end{aligned} \quad (3.38)$$

où  $\eta > 0$ .

Pour assurer la stabilité, le système flou est appliqué et la dérivée de la fonction de Lyapunov doit être négative. En effet,  $dv(t)$  varie temporellement et pour faire ressortir l'effet des perturbations,  $G(t)$  doit suivre temporellement.

### 3.4.2.4 Application de la logique floue

Les règles du système flou sont utilisées pour estimer  $G(t)$  [134,128]. Un système flou est utilisé pour obtenir le paramètre  $G(t)$  de la loi de contrôle (3.34). L'organigramme de la commande par mode glissant flou appliqué sur le robot peut être schématisé par la figure (3.4).

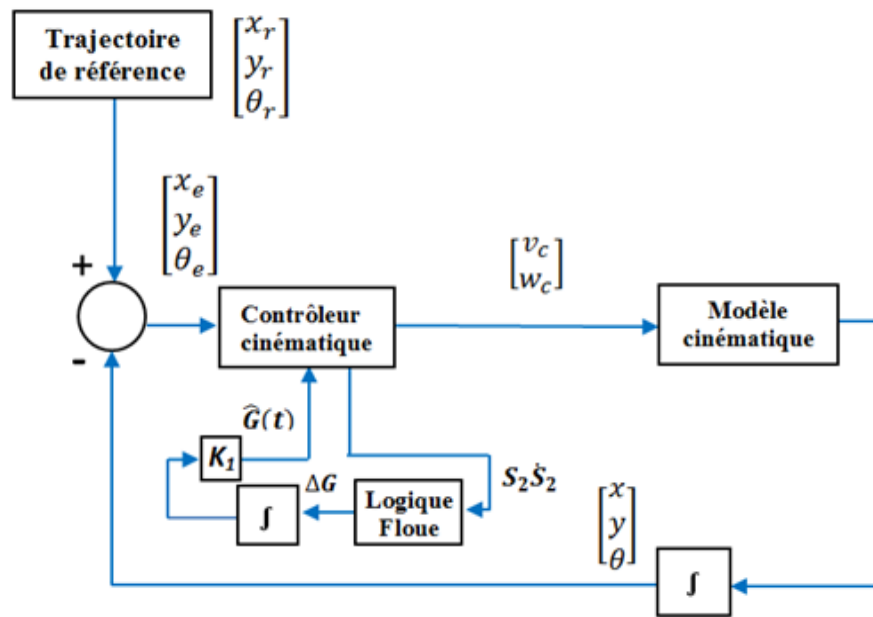


Figure 3.4 : Architecture de la commande cinématique par mode glissant flou

Donc, pour concevoir le contrôle flou, il est intéressant d'assurer la condition du mode glissant, qui est donnée par l'équation suivante:

$$s_2 \dot{s}_2 < 0 \quad (3.39)$$

Si l'équation (3.39) est alors vérifiée, les états du système seront sur la deuxième surface de glissement  $s_2$ .

Dans ce système,  $s_2 \dot{s}_2$  est l'entrée et  $G(t)$  est la sortie. Les ensembles flous des entrées et des sorties sont les mêmes et définis comme suit: (NL, NM, Z, PM, PL).

où NL, NM, Z, PM et PL sont des termes linguistiques définis par négative grande, négative moyenne, zéro, moyenne positive et positive grande. Pour assurer la condition de présence du mode glissant, les règles floues sont appliquées comme:

Si  $s_2 \dot{s}_2$  est NL alors  $\Delta G(t)$  est NL

Si  $s_2 \dot{s}_2$  est NM alors  $\Delta G(t)$  est NM

Si  $s_2 \dot{s}_2$  est Z alors  $\Delta G(t)$  est Z

Si  $s_2 \dot{s}_2$  est PM alors  $\Delta G(t)$  est PM

Si  $s_2 \dot{s}_2$  alors  $\Delta G(t)$  est PL

La fonction d'appartenance aux entrées est représentée à la figure (3.5) et à la figure (3.6). Ainsi, la fonction d'appartenance de la sortie est donnée par la figure (3.7).

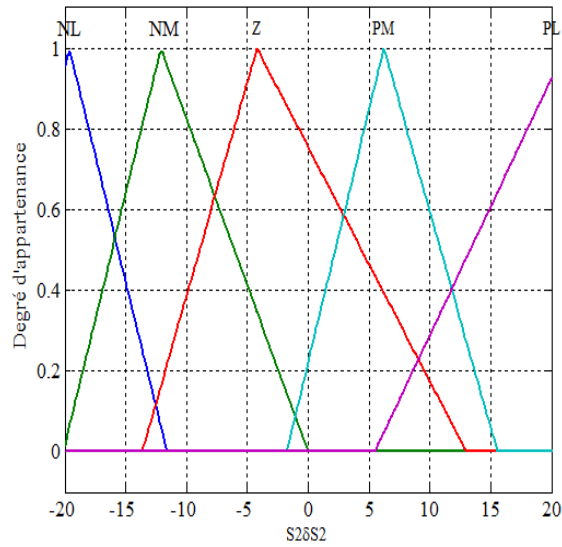


Figure 3.5 : Ensembles flous d'entrée  $s_2 \dot{s}_2$

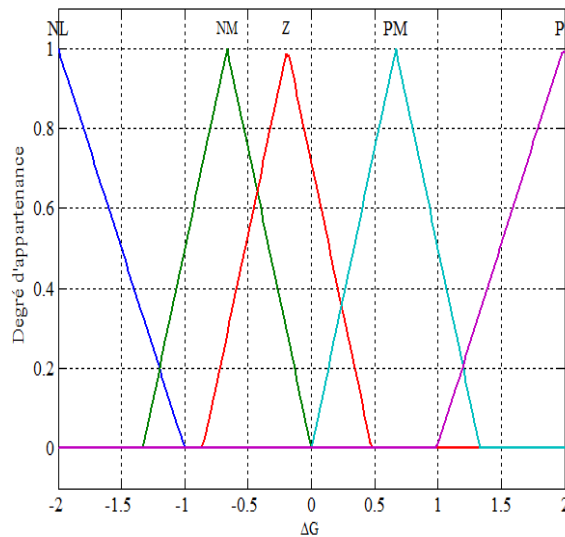


Figure 3.6 : Ensembles flous de la sortie  $\Delta G(t)$

Le paramètre  $G(t)$  est estimé par:

$$\hat{G}(t) = K_1 \int_0^t \Delta G(t) dt \tag{3.40}$$



où  $K_I$  représente le coefficient de proportionnalité et est ajusté selon les essais.

La loi de contrôle (3.34) devient une nouvelle loi de contrôle flou qui est donnée par:

$$v_c = v_r + \omega_r x_e + \omega_r y_e + k s_2 + \hat{G}(t) \frac{s_2}{|s_2| + \delta} \quad (3.41)$$

Pour l'analyse de stabilité, la dérivée de la fonction Lyapunov (3.36) est considérée:

$$\dot{T}_l = -s_2 dv - |s_2| \hat{G}(t) - k s_2^2 \leq -\eta |s_2| - k s_2^2 \leq 0 \quad (3.42)$$

La dérivée de Lyapunov est définie négative avec des valeurs de  $k$  et  $\hat{G}(t)$  supérieur à la perturbation  $dv$ .

Par conséquent, le contrôle  $v_c$  de la surface de glissement  $s_2$  fait converger les états d'erreur vers zéro avec une stabilité asymptotique. La loi de contrôle du modèle cinématique est donnée comme suit :

$$V_c = \begin{bmatrix} v_c \\ \omega_c \end{bmatrix} = \begin{bmatrix} \omega_r + \beta \theta_e^{q/p} + \alpha \theta_e \\ v_r + \omega_r x_e + \omega_r y_e + k s_2 + \hat{G}(t) \frac{s_2}{|s_2| + \delta} \end{bmatrix} \quad (3.43)$$

Par conséquent, si  $v_r = v_c$  et  $\omega_r = \omega_c$ , la boucle fermée du système est asymptotiquement stable.

## 3.5 Commande dynamique robuste par mode glissant

### 3.5.1 Modèle dynamique du robot mobile

Le modèle dynamique du robot mobile est décrit par [135,136] :

$$M(q) \dot{V} + V(q, \dot{q}) V + F(\dot{q}) + G(q) + \tau_d = \beta(q) \tau + R(t) \quad (3.44)$$

où  $V = (v, \omega)^T$  est un vecteur de vitesses et  $\tau = (\tau_r \quad \tau_l)$  représente les couples des roues droite et gauche.

$$M(q) = \begin{bmatrix} m & 0 \\ 0 & I \end{bmatrix} \text{ et } \beta(q) = \frac{1}{R_a} \begin{bmatrix} 1 & 1 \\ L & -L \end{bmatrix}$$

où  $m$  est la masse du robot et  $I$  le moment d'inertie.  $R(t)$  représente le vecteur de perturbation  $2 \times 1$ .  $V(q, \dot{q})$  sont les forces centripète et de Coriolis.  $F(\dot{q})$  est la matrice de friction,  $G(q)$  représente le vecteur gravitationnel et  $\tau_d$  est une perturbation inconnue.

En considérant le modèle dynamique (3.44) et en tenant compte que le robot se déplace dans un plan horizontal, le vecteur gravitationnel, le vecteur centripète, la matrice de Coriolis, la matrice de friction et les perturbations inconnues deviennent nuls. Le modèle dynamique (3.44) devient comme suit:

$$M(q)\dot{V} = \beta(q)\tau + R(t) \tag{3.45}$$

### 3.5.2 Contrôle dynamique basé sur le mode glissant

L'application de la loi de commande par modes glissants sur le robot peut être schématisée par la figure (3.7).

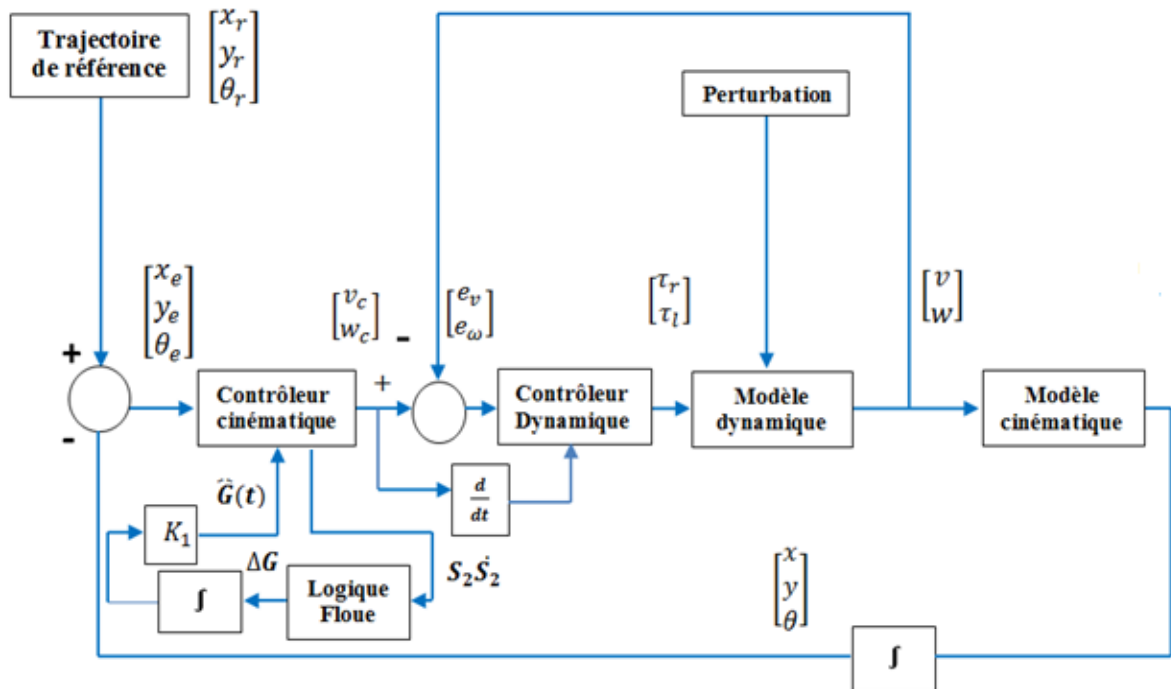


Figure 3.7 : Architecture de contrôle dynamique par mode glissant

Le contrôle  $\tau$  du modèle dynamique est conçu pour amener les vitesses réelles du robot à la référence obtenue avec le contrôleur cinématique. Dans cette section, une commande par

mode glissant exponentiel est utilisée pour la commande du couple  $\tau$  afin de garantir l'erreur de convergence  $V_e$  des vitesses.

$$\lim_{t \rightarrow \infty} V_e = \lim_{t \rightarrow \infty} \|V_c(t) - V(t)\| = 0 \quad (3.46)$$

L'erreur de vitesse est choisie comme:

$$V_e = [e_v \quad e_\omega]^T = \begin{bmatrix} v_c - v \\ \omega_c - \omega \end{bmatrix} \quad (3.47)$$

La dérivée de l'erreur de vitesse est donnée comme suit:

$$\dot{V}_e = \begin{bmatrix} \dot{v}_c - \dot{v} \\ \dot{\omega}_c - \dot{\omega} \end{bmatrix} \quad (3.48)$$

Les surfaces de glissement sélectionnées sont:

$$S = \begin{bmatrix} s_3 \\ s_4 \end{bmatrix} = \begin{bmatrix} v_c - v \\ \omega_c - \omega \end{bmatrix} \quad (3.49)$$

La dérivée de la surface est donnée comme :

$$\dot{S} = \begin{bmatrix} \dot{s}_3 \\ \dot{s}_4 \end{bmatrix} = \begin{bmatrix} \dot{v}_c - \dot{v} \\ \dot{\omega}_c - \dot{\omega} \end{bmatrix} \quad (3.50)$$

En utilisant le contrôle par mode glissant exponentiel:

$$\dot{S} = \begin{bmatrix} -\varepsilon_1 \operatorname{sgn}(s_3) - \mu_1 s_3 \\ -\varepsilon_2 \operatorname{sgn}(s_4) - \mu_2 s_4 \end{bmatrix} \quad (3.51)$$

où  $\varepsilon = \begin{bmatrix} \varepsilon_1 & 0 \\ 0 & \varepsilon_2 \end{bmatrix}$  et  $\mu = \begin{bmatrix} \mu_1 & 0 \\ 0 & \mu_2 \end{bmatrix}$  sont des constantes positives.

En combinant les équations (3.47) et (3.51), le contrôle dynamique est donné comme suit:

$$\tau = M(q)\beta(q)^{-1}(\dot{V}_c + \varepsilon \operatorname{sgn}(S) + \mu S) \quad (3.52)$$

En considérant le modèle dynamique (3.45) avec les perturbations et les incertitudes  $R(t)$ , on a le résultat suivant:

$$\dot{V} = \beta(q)M(q)^{-1}\tau + M(q)^{-1}R(t) \quad (3.53)$$

En choisissant l'équation suivante [137]:

$$\beta(q)M(q)^{-1} = Q = (\hat{Q} + \Delta Q) \quad (3.54)$$

$\Delta Q$  représente les incertitudes et  $\hat{Q}$  est le terme nominal de la matrice. Par conséquent,  $\varphi(t)$  est défini comme l'incertitude de la limite supérieure, alors l'équation (3.54) devient:

$$\Delta Q\tau + M(q)^{-1}R(t) = \varphi(t) \quad (3.55)$$

Où  $|\varphi(t)| \leq \phi$  et  $\phi$  est une constante positive.

La loi de contrôle est déterminée comme suit:

$$\tau = \hat{Q}^{-1}(\dot{V}_c + \varepsilon \operatorname{sgn}(S) + \mu S) \quad (3.56)$$

$$\begin{aligned} \begin{bmatrix} \tau_r \\ \tau_l \end{bmatrix} &= \hat{Q}^{-1} \left( \begin{bmatrix} \dot{v}_c \\ \dot{\omega}_c \end{bmatrix} + \begin{bmatrix} \varepsilon_1 & 0 \\ 0 & \varepsilon_2 \end{bmatrix} \begin{bmatrix} \operatorname{sgn}(s_3) \\ \operatorname{sgn}(s_4) \end{bmatrix} \right. \\ &\left. + \begin{bmatrix} \mu_1 & 0 \\ 0 & \mu_2 \end{bmatrix} \begin{bmatrix} s_3 \\ s_4 \end{bmatrix} \right) \end{aligned} \quad (3.57)$$

Pour éviter le phénomène de broutement créé par la fonction signe, une fonction du mode quasi-glissant est appliquée:

$$\dot{S} = \begin{bmatrix} -\varepsilon_1 \frac{s_3}{|s_3| + \delta_1} - \mu_1 s_3 \\ -\varepsilon_2 \frac{s_4}{|s_4| + \delta_2} - \mu_2 s_4 \end{bmatrix} \quad (3.58)$$

Par conséquent, la loi de contrôle devient:

$$\begin{aligned} \begin{bmatrix} \tau_r \\ \tau_l \end{bmatrix} &= \hat{Q}^{-1} \left( \begin{bmatrix} \dot{v}_c \\ \dot{\omega}_c \end{bmatrix} + \begin{bmatrix} \varepsilon_1 & 0 \\ 0 & \varepsilon_2 \end{bmatrix} + \begin{bmatrix} \frac{s_3}{|s_3| + \delta_1} \\ \frac{s_4}{|s_4| + \delta_2} \end{bmatrix} \right) \\ &\quad \begin{bmatrix} \mu_1 & 0 \\ 0 & \mu_2 \end{bmatrix} \begin{bmatrix} s_3 \\ s_4 \end{bmatrix} \end{aligned} \quad (3.59)$$

La sélection de la fonction de Lyapunov est donnée comme suit:

$$L_d = \frac{1}{2} S^T S \quad (3.60)$$

Sa dérivée est donnée comme :

$$\begin{aligned} \dot{L}_d = S^T \dot{S} &= s_3 \dot{s}_3 + s_4 \dot{s}_4 = S^T \dot{S} = -\varepsilon_1 \frac{s_3^2}{|s_3| + \delta_1} - \mu_1 s_3^2 \\ &\quad - \varepsilon_2 \frac{s_4^2}{|s_4| + \delta_2} - \mu_2 s_4^2 - S^T M(q)^{-1} R(t) = -\varepsilon \|S\| - S^T \mu S - S^T M(q)^{-1} R(t) \leq 0 \end{aligned} \quad (3.61)$$

où  $\text{diag}(\varepsilon_1, \varepsilon_2) > 0$ ,  $\text{diag}(\mu_1, \mu_2) > 0$  et  $\delta_1, \delta_2 > 0$ . Les paramètres  $\text{diag}(\varepsilon_1, \varepsilon_2)$  et  $\text{diag}(\mu_1, \mu_2)$  sont utilisés pour compenser l'effet des perturbations et des incertitudes du système.

La dérivée de Lyapunov est définie négative avec des valeurs de  $\varepsilon$  et  $\mu$  supérieur à la perturbation  $R(t)$ .

## 3.6 Commande dynamique robuste par PSO et mode glissant

### 3.6 Optimisation par PSO

L'optimisation des essaims de particules (PSO) a été inventée par Russell Eberhart et James Kennedy. Elle est inspirée par les schémas des flots et de déplacements des oiseaux et des poissons. Le PSO est une méthode métaheuristique basée sur la population qui met à jour chaque individu à chaque itération de l'exécution. Il manque de stigmatisation, contrairement à l'optimisation des colonies de fourmis, en ce sens que les populations ne communiquent pas ou n'interfèrent pas avec l'environnement d'une manière ou d'une autre pour atteindre leur objectif. La stochastique est utilisée dans une certaine mesure dans l'optimiseur pour assurer une large exploration de l'espace de recherche de problèmes. Le PSO ne comporte cependant

pas le même réseau de topologie en anneau que d'autre méthode d'optimisation, mais utilise à la place une composante de mémoire et de vitesse pour chaque particule.

### 3.6.2 Principe d'optimisation

Lors de l'initialisation, chaque individu crée une solution de position aléatoire. Chaque particule crée également une vitesse aléatoire dans la plage de  $[-r; r]$ . Cette plage dépend de la taille de l'espace de recherche et détermine le niveau de vitesse des particules lors de leur création. La meilleure position de chaque particule est définie sur sa position initiale créée aléatoirement.

La vitesse de chaque particule est modifiée itérativement par sa meilleure position individuelle ( $pbest$ ), et la meilleure position globale ( $gbest$ ) trouvée par les particules dans son intervalle. En conséquence, chaque particule recherche autour d'une région définie par sa meilleure position individuelle ( $pbest$ ) et la meilleure position globale ( $gbest$ ) de son voisinage. Désormais, nous utilisons  $V_j$  pour désigner la vitesse des particules dans l'essaim, et  $p_j$  désigne sa position. À chaque étape (ou itération)  $n$ , en utilisant la meilleure position individuelle ( $pbest$ ) et la meilleure position globale ( $gbest$ ), la vitesse et la position de chaque particule est mise à jour par les équations suivantes [138,139]:

$$V_j = w(v_j(n-1) + c_1r_1(pbest_j - p_j(n-1))) + c_2r_2(gbest_j - p_j(n-1)) \quad (3.62)$$

$$p_j(n) = p_j(n-1) + V_j(n) \quad (3.63)$$

Où  $r_1$  et  $r_2$  sont des nombres aléatoires compris entre 0 et 1.

Pour chaque itération, la vitesse de chaque particule est mise à jour en utilisant sa meilleure position individuelle et la meilleure position trouvée par l'essaim. Les taux d'apprentissage  $c_1$  et  $c_2$  agissent également comme des limiteurs,  $c_1$  pour l'influence que la position la plus connue d'un individu aura sur sa nouvelle vitesse et  $c_2$  pour l'influence que la meilleure particule dans l'essaim aura sur les autres.

$W$  est le facteur de constriction [140] et est défini par :

$$w = \frac{2}{\left|2 - c - \sqrt{c^2 - 4c}\right|}, c = c_1 + c_2, c > 4 \quad (3.64)$$

A chaque étape  $n$ , la position est située dans la plage de  $[p_{min}, p_{max}]$  [141] :

$$\begin{cases} p_{min} & \text{si } p_j < p_{min} \\ p_j & \text{si } p_{max} < p_j < p_{min} \\ p_{max} & \text{si } p_j > p_{max} \end{cases} \quad (3.65)$$

Changer la position de cette manière permet à la particule de rechercher autour de sa meilleure position individuelle et globale. Ce qui suit montre l'étape de conception pour l'implémentation de l'algorithme PSO [140] :

**Etape 1.** Initialiser les particules avec une position et une vitesse aléatoire.

**Etape 2.** Si un nombre prescrit d'itérations est atteint, arrêter l'algorithme.

**Etape 3.** Pour chaque particule, évaluer la fonction désirée de mise en forme et enregistrer chaque meilleure position précédente de la particule et sa meilleure position globale.

**Etape 4.** Modifier la vitesse et la position en fonction des équations (3.62) et (3.63) pour chaque particule

**Etape 5.** Vérifier la position de chaque particule en utilisant (3.65).

**Etape 6.** Revenir à l'étape 2.

### 3.6.3 Application du PSO

Les paramètres sélectionnés sont  $\varepsilon_1, \varepsilon_2, \mu_1, \mu_2$  dans le contrôleur dynamique (3.59). Par conséquent, il est important de trouver leurs valeurs optimales en utilisant l'algorithme PSO. Le diagramme du contrôle est illustré sur la figure (3.8).

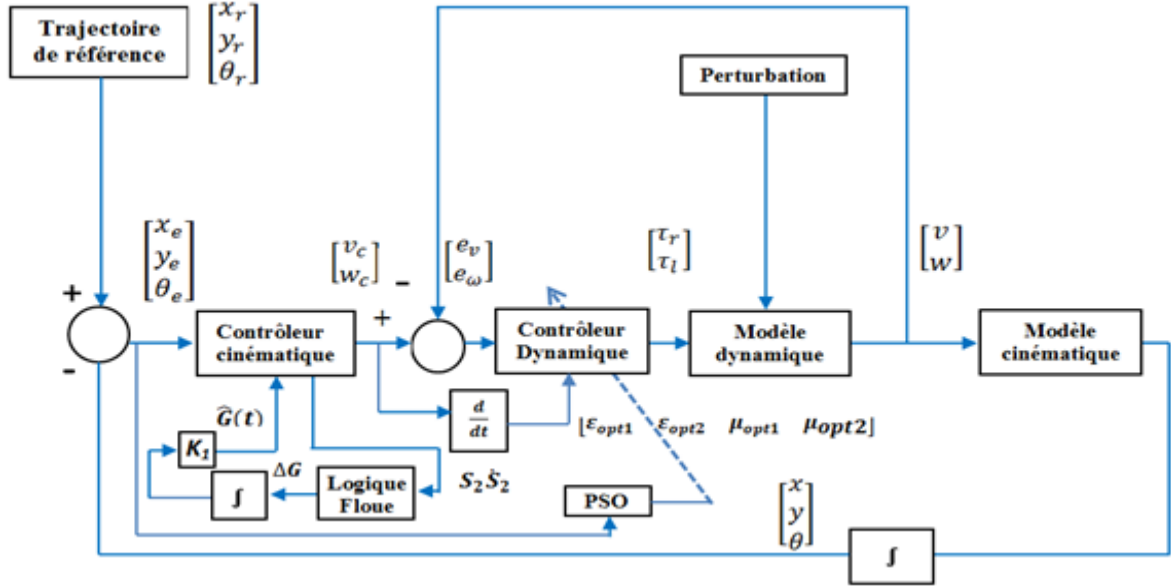


Figure 3.8 : Architecture de contrôle dynamique utilisant le PSO

Le problème de conception est défini comme la recherche des valeurs optimales du contrôleur dynamique par mode glissant. Soit  $p_j = [\varepsilon_1, \varepsilon_2, \mu_1, \mu_2]$  le vecteur des paramètres à optimiser. Pour converger vers la solution optimale, le PSO doit être guidé par une fonction de coût. Par conséquent, elle doit être correctement définie avant l'application du PSO. Donc, la fonction de coût utilisée est définie par la formule suivante:

$$J = \frac{1}{N} \sum_{j=1}^N \|E(j)\|^2 = \frac{1}{N} \sum_{j=1}^N (\|x_e\|^2 + \|y_e\|^2 + \|\theta_e\|^2) \quad (3.66)$$

où  $E(j)$  est l'erreur de suivi et  $N$  est le nombre des échantillons.

En sélectionnant les paramètres  $\varepsilon_1, \varepsilon_2, \mu_1, \mu_2$ , le contrôleur dynamique devient:

$$\begin{aligned} \begin{bmatrix} \tau_r \\ \tau_l \end{bmatrix} &= \hat{Q}^{-1} \left( \begin{bmatrix} \dot{v}_c \\ \dot{w}_c \end{bmatrix} + \begin{bmatrix} \varepsilon_{opt1} & 0 \\ 0 & \varepsilon_{opt2} \end{bmatrix} \begin{bmatrix} \frac{s_3}{|s_3| + \delta_1} \\ \frac{s_4}{|s_4| + \delta_2} \end{bmatrix} \right) \\ &+ \begin{bmatrix} \mu_{opt1} & 0 \\ 0 & \mu_{opt2} \end{bmatrix} \begin{bmatrix} s_3 \\ s_4 \end{bmatrix} \end{aligned} \quad (3.67)$$

où,  $\varepsilon_{opt1}, \varepsilon_{opt2}, \mu_{opt1}, \mu_{opt2}$  sont les paramètres optimaux générés.

La fonction de Lyapunov est définie comme:



$$L_d = \frac{1}{2} S^T S \quad (3.68)$$

La dérivée de la fonction de Lyapunov choisie est donnée par :

$$\begin{aligned} \dot{L}_d = S^T \dot{S} = & -\varepsilon_{opt1} \frac{s_3^2}{|s_3| + \delta_1} - \mu_{opt1} s_3^2 \\ & - \varepsilon_{opt2} \frac{s_4^2}{|s_4| + \delta_2} - \mu_{opt2} s_4^2 - S^T M(q)^{-1} R(t) = -\varepsilon_{opt} \|S\| - S^T \mu_{opt} S - S^T M(q)^{-1} R(t) \leq 0 \end{aligned} \quad (3.69)$$

où  $diag(\varepsilon_{opt1}, \varepsilon_{opt2}) > 0$ ,  $diag(\mu_{opt1}, \mu_{opt2}) > 0$

### 3.7 Travaux de simulation et résultats

Afin de montrer l'efficacité de la loi de contrôle, des travaux de simulation sous environnement Matlab de trois trajectoires différentes (circulaire, sinusoïdale et spécifique) sont envisagées. Les paramètres du robot mobile utilisés pour la simulation sont :  $m=4.5kg$ ,  $I=3kg.m^2$ ,  $R_a=0.03 m$  et  $L=0.15 m$ . Les paramètres de contrôle souhaités sont sélectionnés comme ci-dessous  $v_r = 2 m/s$  et  $\omega_r = 2 rad/s$ . Une fonction de perturbation périodique limitée est introduite entre les temps 4 et 5 secondes.

$$\begin{cases} dv = \sin(t - \pi) \\ d\omega = 2 \sin(t - \pi) \end{cases} \quad (3.70)$$

Les paramètres du contrôleur cinématique sont choisis arbitrairement :

**Tableau 3.1.** Paramètres du contrôleur cinématique

Paramètres	valeurs
$p$	10
$q$	9
$\eta$	1
$k$	30
$K_l$	25
$\alpha$	50
$\beta$	50
$\delta$	0.08

Les paramètres du contrôleur dynamique sans PSO sont sélectionnés avec un indice  $J = 0.2998$ :

**Tableau 3.2.** Paramètres du contrôleur dynamique

Paramètres	valeurs
$\varepsilon_1$	80
$\varepsilon_2$	80
$\mu_1$	30
$\mu_2$	30
$\delta_1$	0.95
$\delta_2$	0.95

Les perturbations dynamiques introduites entre quatre et cinq secondes ( $4 \text{ s} < t < 5 \text{ s}$ ) sont considérées:

$$R(t) = \begin{bmatrix} 2.5 \sin(t - \pi) \\ 1.5 \sin(t - \pi) \end{bmatrix} \quad (3.71)$$

Les paramètres utilisés par le PSO sont considérés comme suit:

Le nombre de particules est égal à 20,  $c_1=2.5$ ,  $c_2=2$ ,  $w=0.5$ , le nombre d'itérations est égal à 100 et dimension=4. Les paramètres optimaux obtenus par le PSO pour le contrôleur dynamique sont variés entre 0 à 80.

Les paramètres obtenus avec un meilleur coût de chaque trajectoire sont données ci-dessous :

**Tableau 3.3.** Paramètres optimaux du contrôleur dynamique obtenus par PSO

Paramètres	Circulaire	Sinusoïdale	Aléatoire
$\varepsilon_{opt1}$	72.133	63.9695	58.1998
$\varepsilon_{opt2}$	74.2128	34.8338	36.4436
$\mu_{opt1}$	13.0861	73.9445	50.0893
$\mu_{opt2}$	58.5592	22.7594	74.7793
$Min(J)$	0.1158	0.1222	0.1241

Par application de la loi de contrôle cinématique (3.43) et les lois de contrôle dynamique (3.59) sans PSO et (3.67) avec PSO, les résultats de simulation pour les différentes trajectoires sont illustrés ci-dessous.

**A- Simulations avec une trajectoire circulaire**

Les erreurs de position et d'orientation initiales sont données par (2m, 1m,  $\pi / 6$  rad).

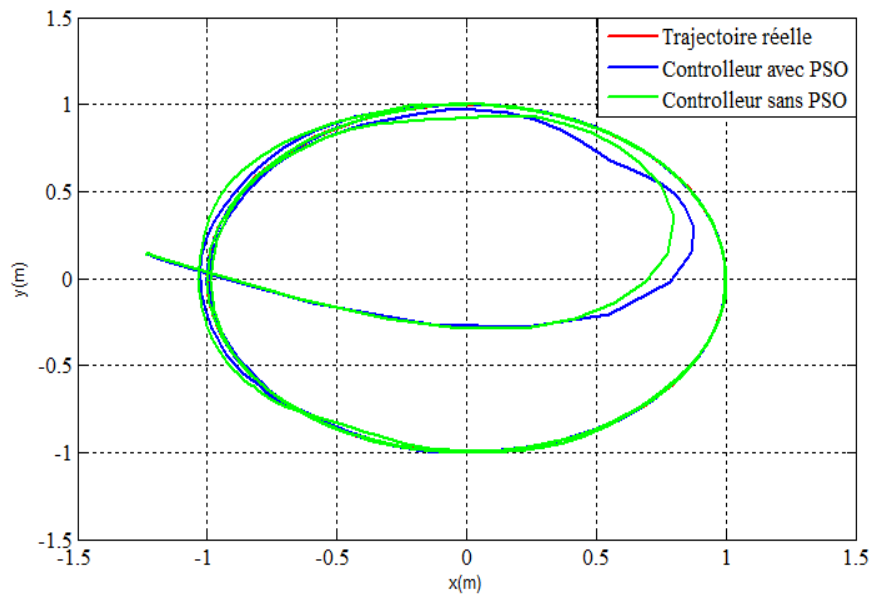


Figure 3.9: Poursuite d'une trajectoire circulaire

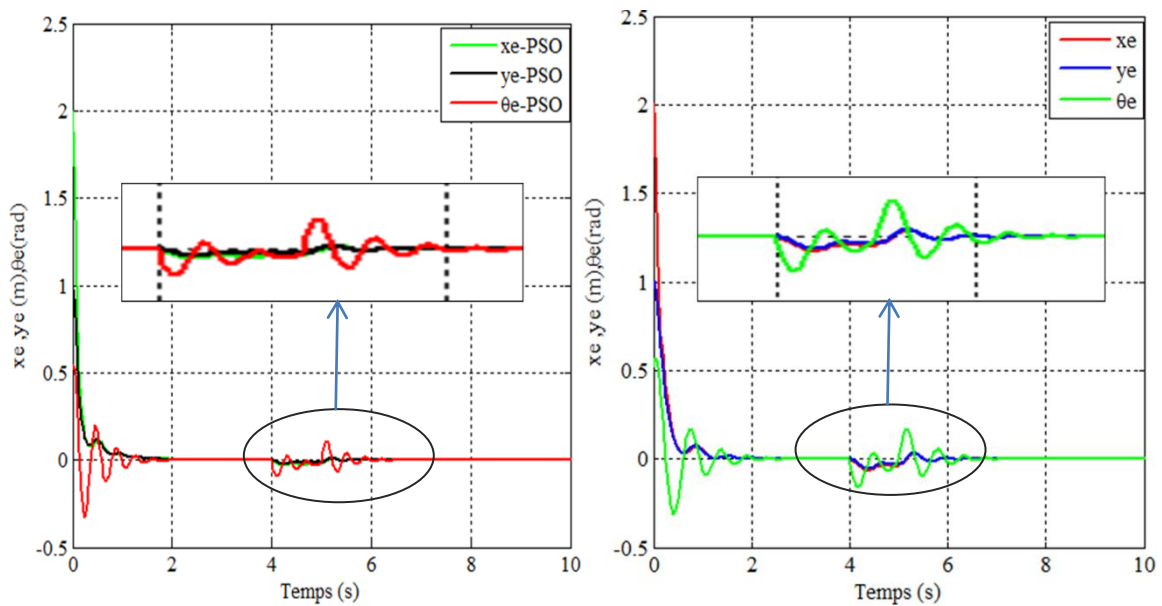


Figure 3.10: Erreurs de poursuite avec et sans PSO

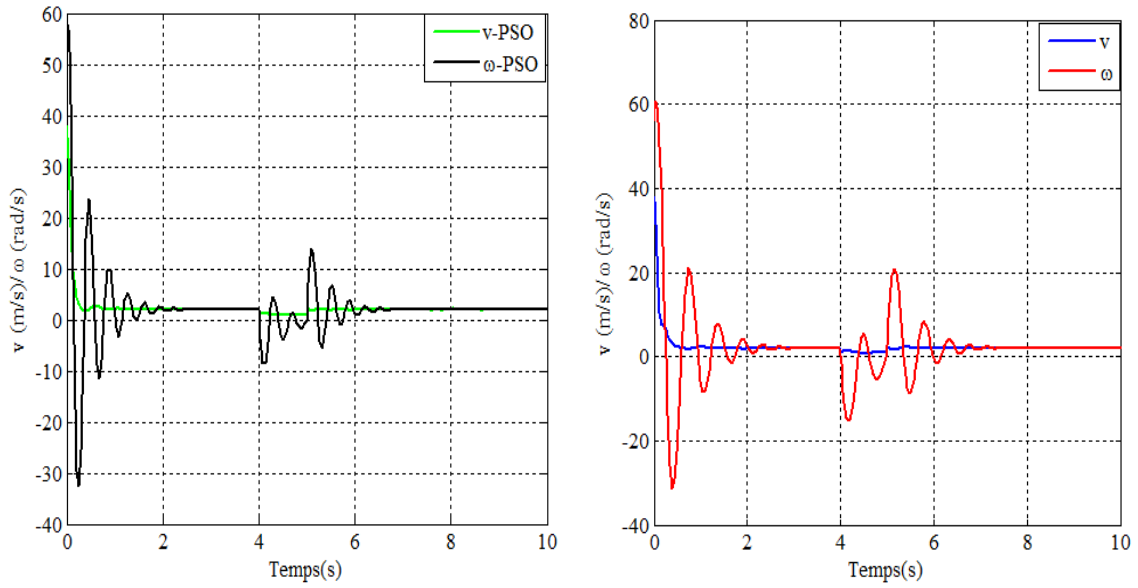


Figure 3.11: Vitesses de commande  $v$  et  $\omega$  avec et sans PSO

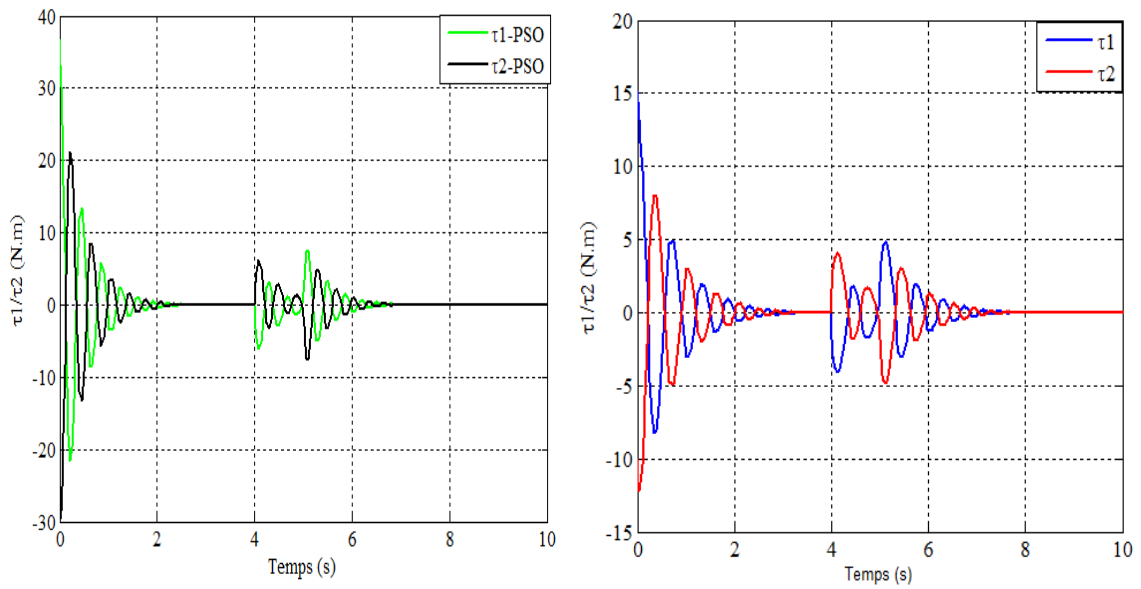


Figure 3.12: Couples  $\tau_1$  et  $\tau_2$  avec et sans PSO

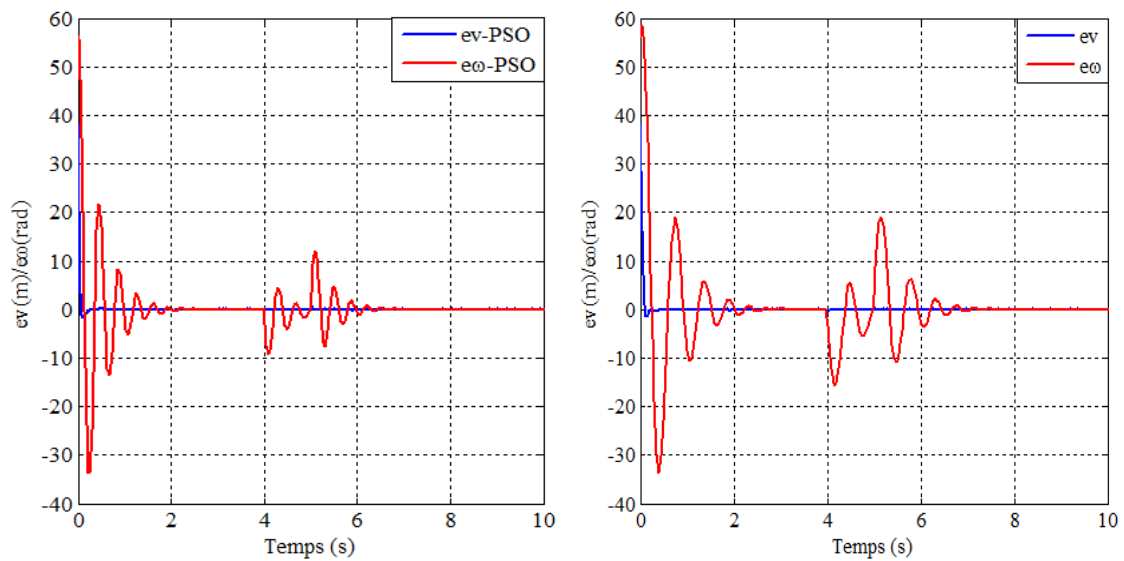


Figure 3.13: Erreurs de vitesses avec et sans PSO

La figures (3.10) illustre la convergence de l'erreur de position et d'orientation du robot en appliquant la loi de commande cinématique et la loi dynamique avec et sans PSO. On peut remarquer que les erreurs convergent vers zéro en temps fini pour les deux contrôleurs par conséquent le contrôleur dynamique par PSO converge les erreurs de postures vers zéros en un temps  $t < 2s$  et  $t = 6s$  après insertion des perturbations.

Les figures (3.10) et (3.12) illustrent respectivement les vitesses de commande et les couples appliqués. On remarque que les couples convergent vers zéros rapidement en utilisant l'optimisation par PSO en présence des perturbations.

Les vitesses linéaire et angulaire réelles du robot convergent rapidement vers les vitesses de références en un temps  $t < 2s$  en appliquant le PSO comme on peut le constater sur les figures (3.13).

**B- Simulations avec une trajectoire sinusoïdale**

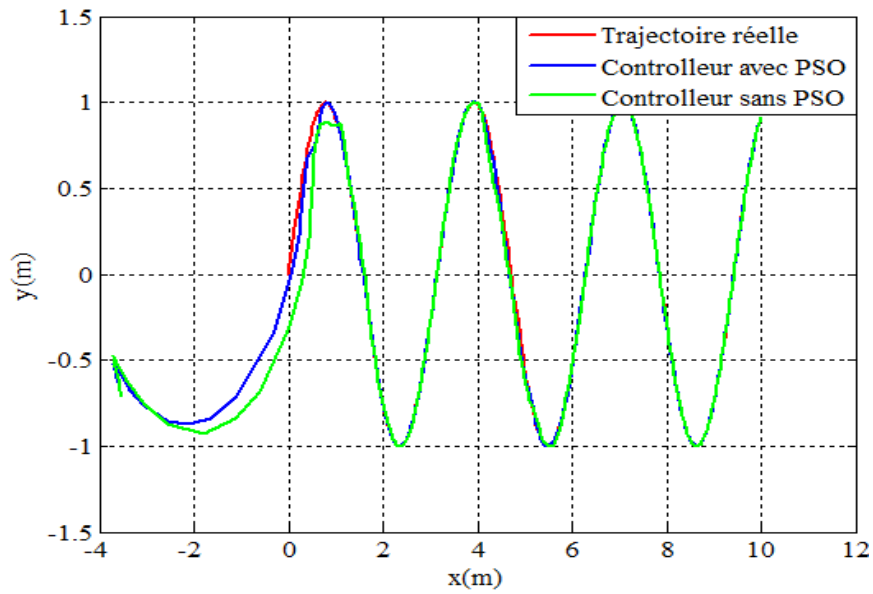


Figure 3.14: Poursuite d'une trajectoire sinusoïdale

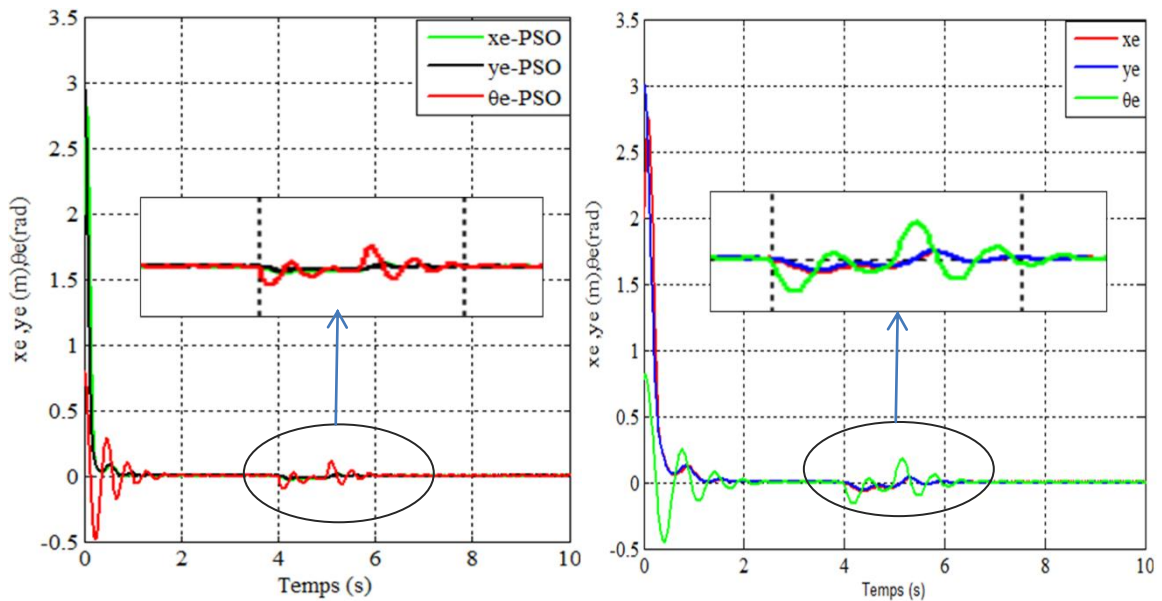


Figure 3.15 : Erreurs de poursuite avec et sans PSO

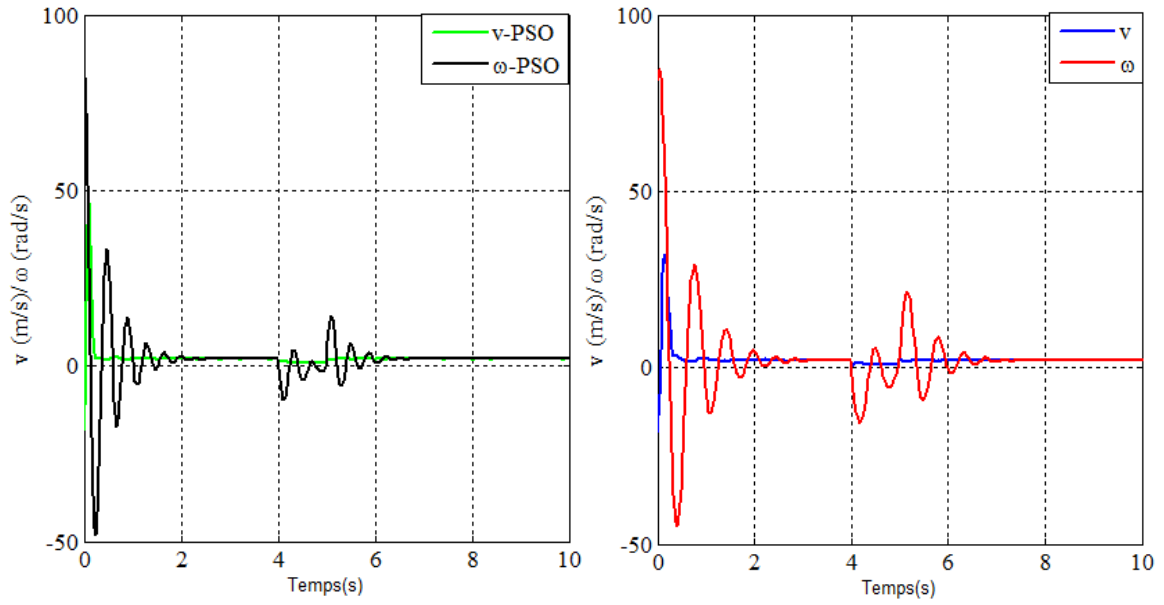


Figure 3.16 : Vitesses de commande  $v$  et  $\omega$  avec et sans PSO

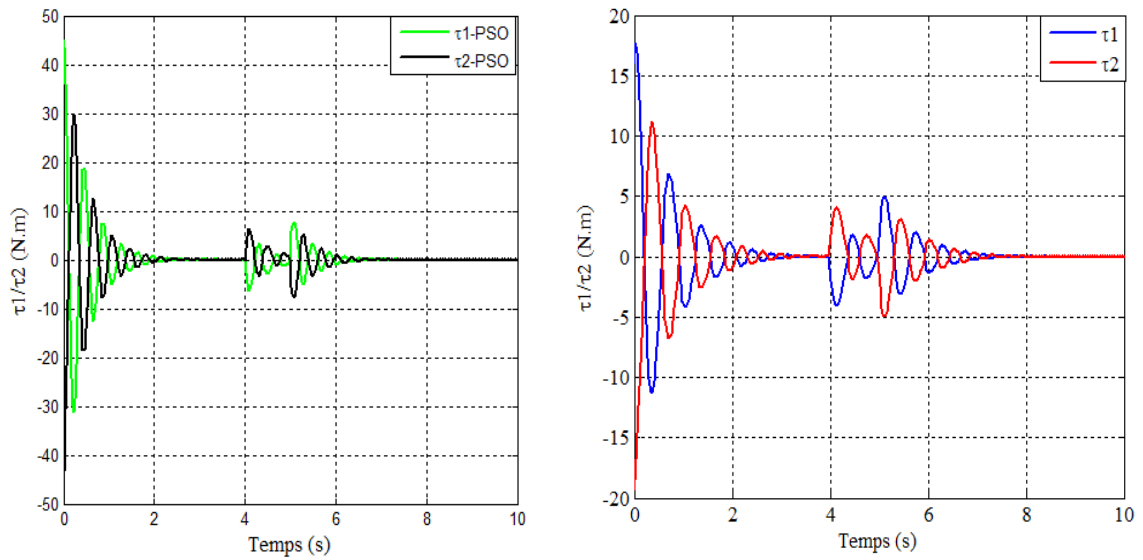


Figure 3.17 : Couples  $\tau_1$  et  $\tau_2$  avec et sans PSO

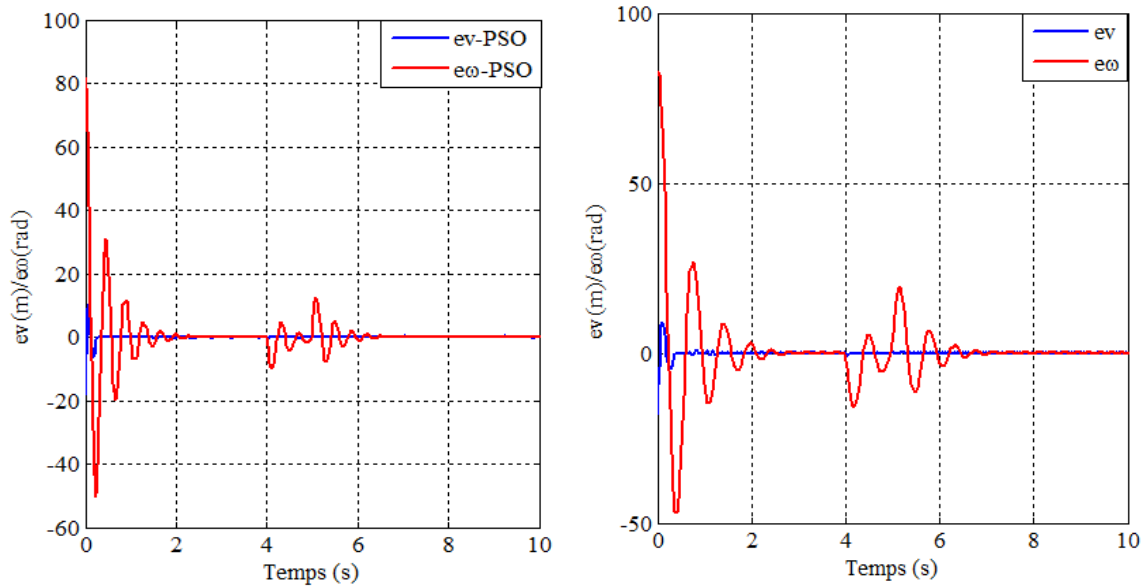


Figure 3.18 : Erreurs de vitesses avec et sans PSO

On peut voir aussi que les résultats sont encore meilleurs en utilisant l'optimiseur (PSO) sur une trajectoire sinusoïdale.

### C- Simulations avec une trajectoire aléatoire

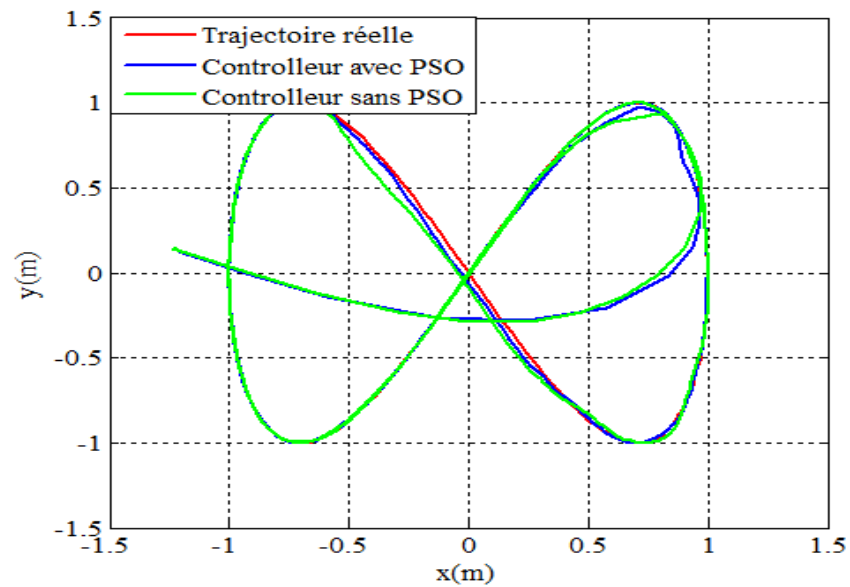


Figure 3.19: Poursuite d'une trajectoire aléatoire



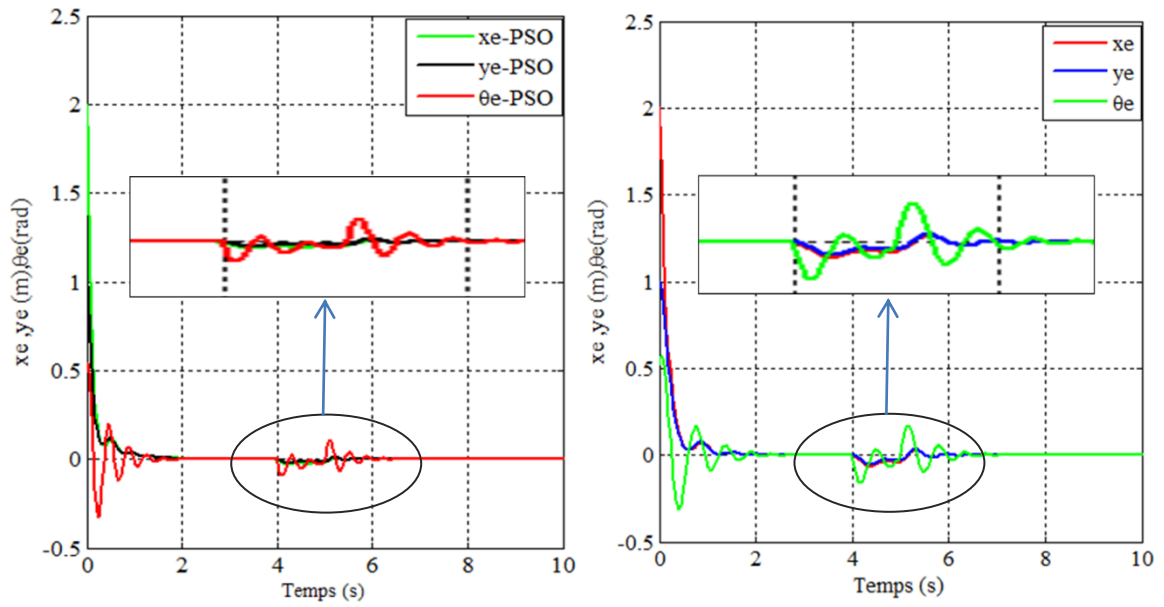


Figure 3.20 : Erreurs de poursuite avec et sans PSO

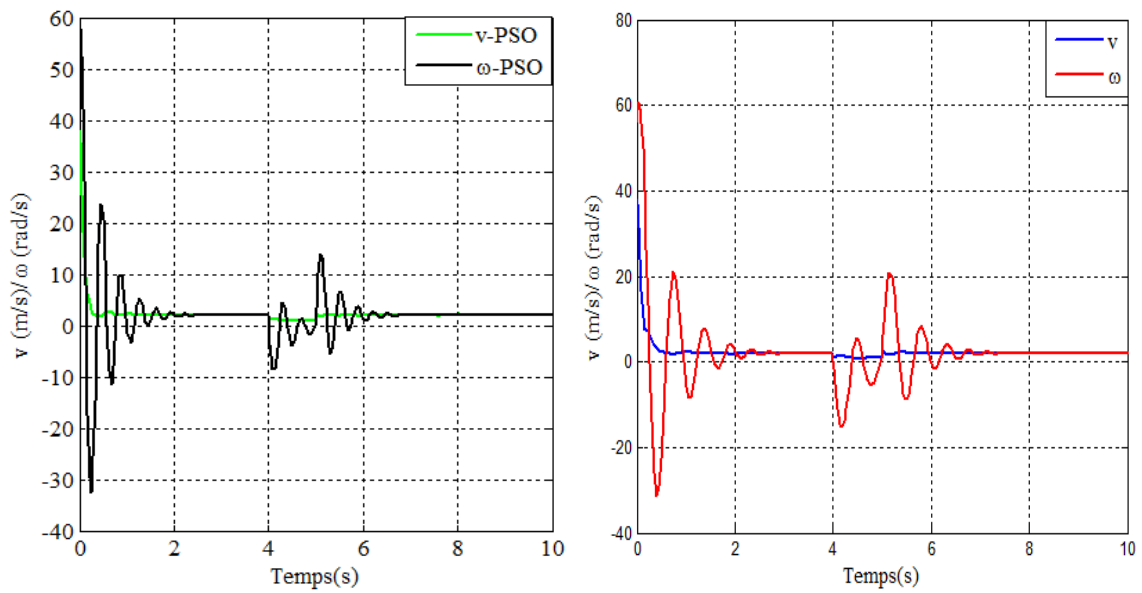


Figure 3.21 : Vitesses de commande  $v$  et  $\omega$  avec et sans PSO

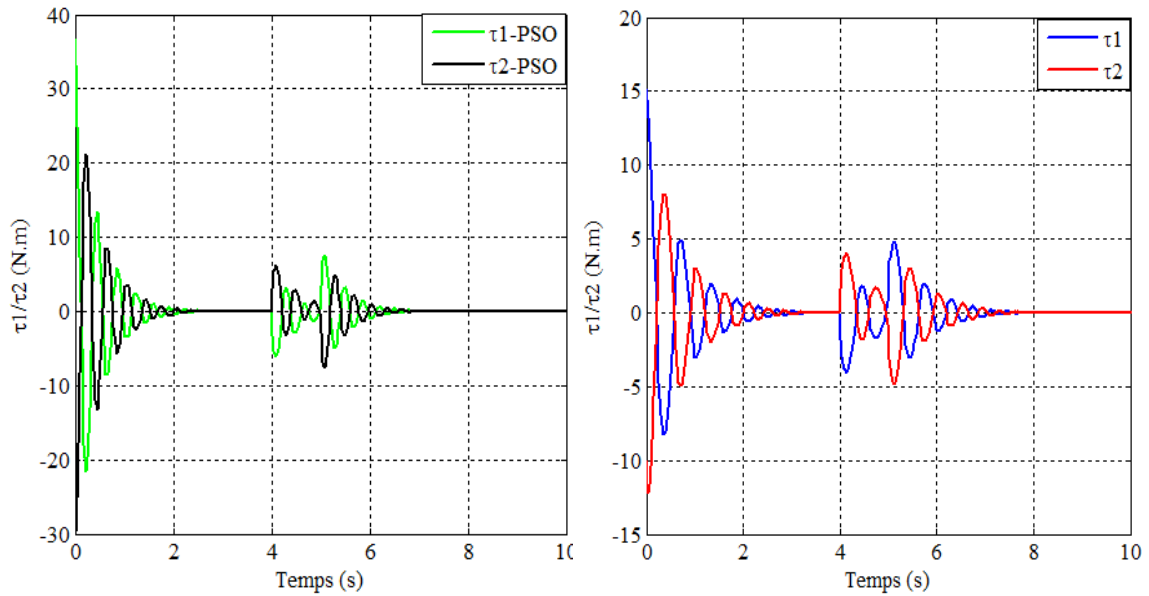


Figure 3.22: Couples  $\tau_1$  et  $\tau_2$  avec et sans PSO

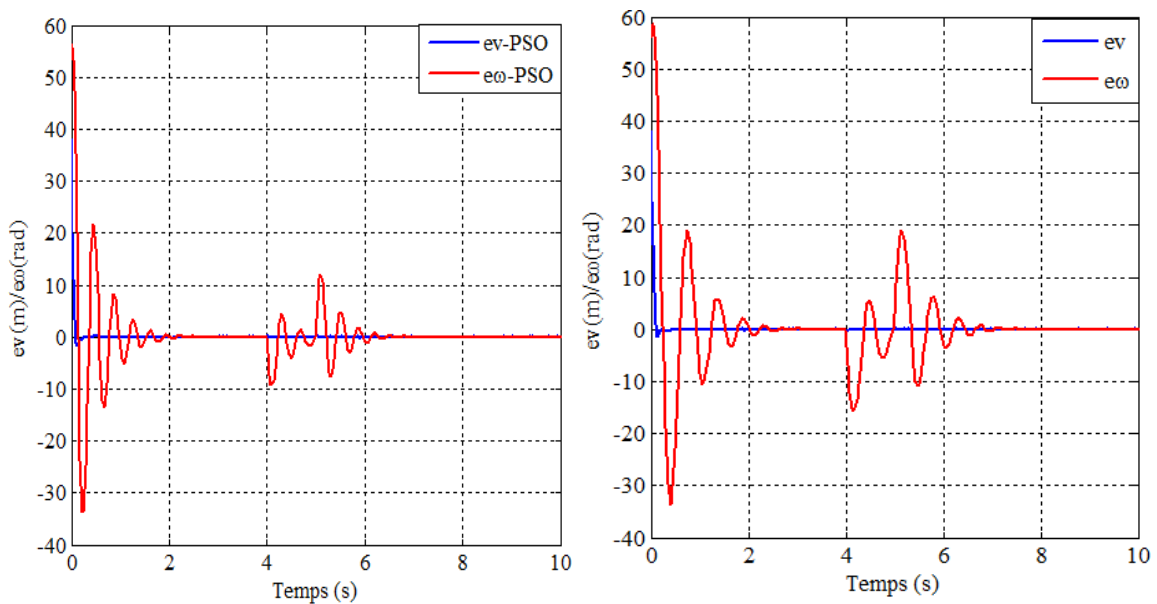


Figure 3.23 : Erreurs de vitesses avec et sans PSO

La commande proposée assure bien la convergence de l'erreur de position et d'orientation du robot vers zéro. Sur la figure (3.20) on constate que le temps de convergence est presque  $t=2s$  et  $t=6s$  après insertion des perturbations en appliquant le PSO. Pour le contrôleur sans PSO les temps sont supérieurs à 2 s et 6s respectivement. La vitesse linéaire et angulaire réelles du robot convergent vers les vitesses de références en un temps  $t < 2s$  par application du PSO, comme on peut le constater sur la figure (3.21).

Les fonctions fitness de trois trajectoires utilisées sont affichées sous dessous :

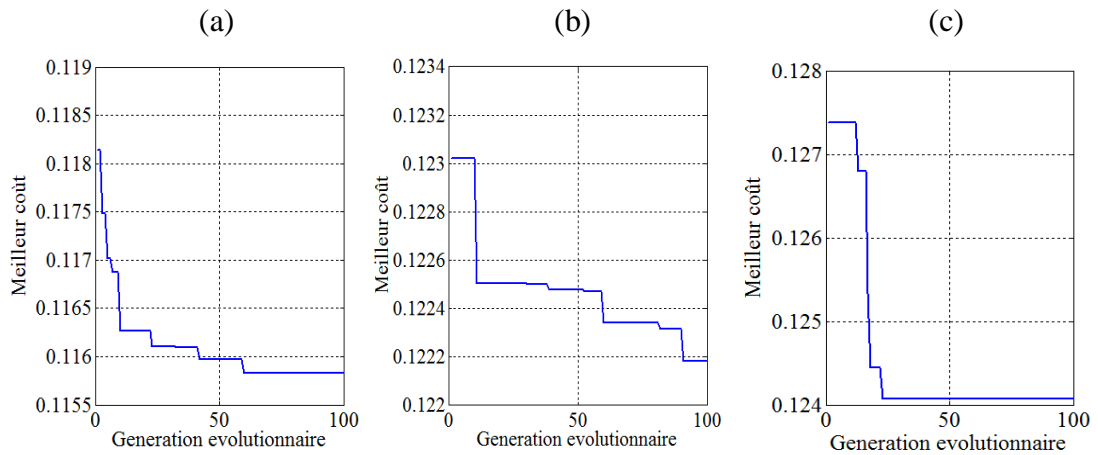


Figure 3.24 : Evolution de la fonction fitness / (a) trajectoire circulaire (b) trajectoire sinusoïdale (c) trajectoire aléatoire

La figure 3.24 montre l'évolution de la fonction objective par la méthode PSO, où l'on remarque la convergence de la fonction.

### 3.8 Conclusion

Dans ce chapitre, différents algorithmes pour le suivi de trajectoires pour un robot mobile nonholonome ont été élaborés. Une commande cinématique basée sur la théorie de la commande par mode glissant et la logique floue est proposée. La convergence asymptotique des erreurs de posture du robot, en présence de perturbations, est assurée par l'introduction de la fonction de Lyapunov.

Une commande dynamique est proposée en se basant sur le mode glissant exponentiel. Cette commande montre une bonne performance du suivi en dépit de la perturbation externe présente dans le modèle dynamique. L'introduction de l'optimisation par essaim de particules (PSO) permet au contrôleur dynamique de donner de bons résultats de simulation par rapport à l'absence du PSO.

Les algorithmes proposés montrent la robustesse pour les trois formes de trajectoires de références considérées, que ce soit une trajectoire circulaire, sinusoïdale ou aléatoire. L'erreur de position et d'orientation du robot converge vers zéro dans de courtes durées avant et après insertion des perturbations.

## **4<sup>ER</sup> CHAPITRE**

# **COMMANDE ROBUSTE INTELLIGENTE APPLIQUEE AUX SUIVI DE TRAJECTOIRES D'UN ROBOT MOBILE**

---

---

## **CHAPITRE 4**

### **Commande robuste intelligente appliquée au suivi de trajectoires d'un robot mobile**

#### **4.1 Introduction**

Dans ce chapitre, de nouveaux contrôleurs sont proposés pour le suivi de trajectoires d'un robot mobile nonholonome. Ces contrôleurs sont composés de deux parties.

La première partie est constituée d'un contrôleur cinématique PD basé sur le mode glissant utilisant la logique floue afin de tendre l'erreur de suivi vers zéros. Le modèle cinématique considéré est influencé par une perturbation apparaissant sur les vitesses linéaire et angulaire du robot. La loi de contrôle proposée fait face à la perturbation et aux incertitudes du système en utilisant la logique floue. Le gain obtenu doit être suffisant pour limiter les effets des perturbations. La loi de contrôle proposée fait converger les erreurs de posture du robot dans un temps très court avec une stabilité asymptotique en tenant compte de la théorie de Lyapunov.

La deuxième partie consiste en un contrôleur dynamique basé sur le mode glissant terminal (FTF) [142] afin de pouvoir générer les couples nécessaires à la convergence de l'erreur de vitesse vers zéro en un temps fini, tout en introduisant les algorithmes génétiques pour l'optimisation des paramètres de la surface de glissement. Cette dernière est une fonction représentant l'erreur de vitesse entre les vitesses angulaires du modèle dynamique et du modèle cinématique. La loi de contrôle proposée fait converger asymptotiquement les erreurs de vitesses vers zéro.

Dans le but de confirmer la robustesse des algorithmes proposés, des travaux de simulation sont entrepris avec différentes trajectoires de référence.

## 4.2 Mode de glissement terminal (TSM)

La méthode du contrôle par mode glissant terminal (TSM) est largement reconnue pour sa robustesse à certaines variations des paramètres et des perturbations externes [143, 144]. Dans cette méthode un attracteur terminal est choisi comme une loi d'arrivée et la fonction terminale est sélectionnée comme une fonction de la surface de glissement. Cette nouvelle conception génère des avantages particuliers sur les performances du système, tels que la convergence à temps fini et une bonne robustesse aux incertitudes [145].

Une fonction terminale (TF) est sélectionnée comme :

$$s = \dot{x} + \alpha x^{\frac{p}{q}} \quad (4.1)$$

où  $x \in \mathfrak{R}$  représente l'état de système,  $\alpha \in \mathfrak{R}^+$  et  $p, q \in \mathbb{N}, p < q$

L'une des caractéristiques essentielles est que l'exposant ( $p / q$ ) est associé à l'état  $x$  et non à sa dérivée. Par rapport à la fonction de glissement linéaire, la fonction (4.1) a une lente vitesse de convergence malgré qu'elle peut converger en un temps fini [146].

Dans le but de remédier à ce problème, Yu Man et al. [147] ont proposé une fonction terminale avec des caractéristiques de convergence rapide appelée fonction terminale rapide (FTF) :

$$s = \dot{x} + \alpha x^\gamma + \beta x^\rho \quad (4.2)$$

Avec,  $\beta \in \mathfrak{R}^+, \gamma, \rho \in \mathfrak{R}^+$  et  $0 < \gamma < 1, \rho > 1$ .

Cette équation possède un terme d'ordre polynomial supérieur à 1 par rapport à l'équation (4.1). Quand  $x \ll 1$ , le terme d'ordre supérieur est négligeable, dans ce cas l'équation (FTF) peut être approximée par l'équation (4.1). Par contre, lorsque  $x \gg 1$ , le terme d'ordre supérieur est capable de renforcer les entrées de contrôle à converger plus rapidement que toute fonction de glissement linéaire.

Un autre type de fonction est proposé dans [148] avec une description globale. Cette fonction est représentée en introduisant de nouvelles fonctions mathématiques:

$$s = \dot{x} + \frac{\alpha}{k} (e^{kx} - 1)^{\frac{q}{p}} x^{-kx} \quad (4.3)$$

Avec  $0 < k < 1$ .

L'inconvénient majeur de TSM est la singularité qui peut apparaître dans certains cas. Une solution pour éviter ce type de problème est proposée et suggère que l'entrée de contrôle soit bornée si certains paramètres vérifient l'inégalité suivante :

$$\rho_{k+1} > \gamma_{k+1} > \frac{n-k-1}{n-k}, \quad 0 \leq k \leq n-1 \quad (4.4)$$

En général, il existe deux attracteurs terminaux qui sont couramment utilisés:

$$\dot{s} + \alpha.s^{\frac{p}{q}} = 0 \quad (4.5)$$

$$\dot{s} + \alpha s^{\gamma} + \beta s^{\rho} = 0 \quad (4.6)$$

En résumé, la rapidité et la singularité sont les deux préoccupations majeures du contrôle par mode glissant terminal. Le FTF peut être utilisé pour améliorer la convergence. La singularité est difficile à gérer dans la réalité et à ce jour les solutions proposées incluent la méthode de contrainte d'inégalité et la fonction terminale non singulière.

## 4.3 Algorithmes génétiques (AG)

### 4.3.1 Introduction

Les algorithmes génétiques sont des approches de recherche heuristique applicables à une large gamme de problèmes d'optimisation [149]. Cette caractéristique les rend attrayants pour beaucoup de problèmes d'optimisation dans la pratique. L'évolution est la base des algorithmes génétiques et la variété des espèces en est une bonne représentation. Les espèces sont capables de s'adapter à leur environnement. Ils développent des structures complexes qui permettent la survie dans différents types d'environnements. L'accouplement et la progéniture interviennent aux grands principes du succès de l'évolution. Ce sont de bons atouts pour l'adaptation des principes de l'évolution à la résolution des problèmes d'optimisation.

Le processus de l'algorithme génétique est illustré sur la figure (4.1) [150].

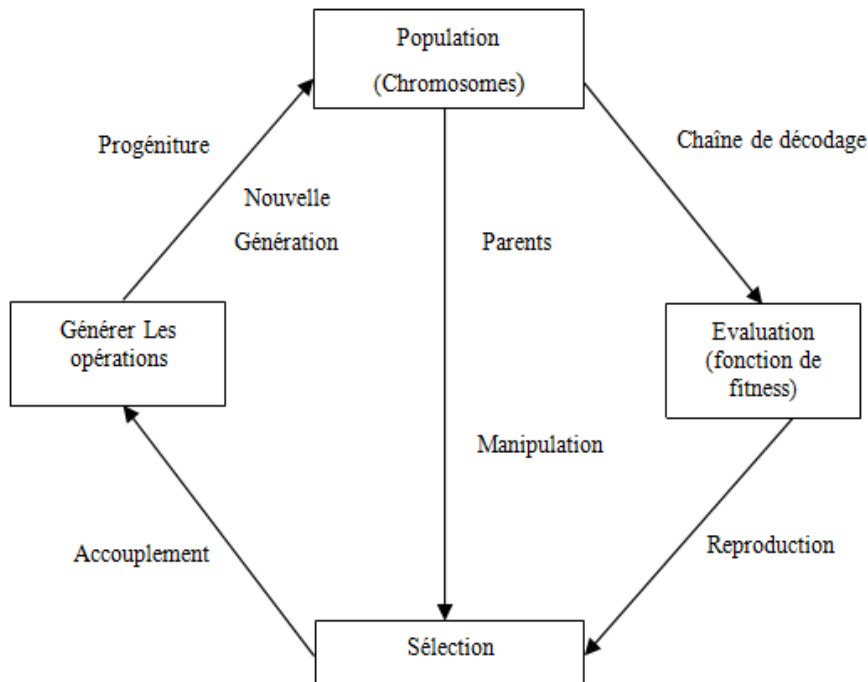


Figure 4.1 : Cycle de l'algorithme génétique

### 4.3.2 Représentation et évaluation des chromosomes

Différentes représentations sont disponibles pour le chromosome. Par conséquent, la sélection de la représentation correcte est spécifique au problème, la bonne représentation est celle qui rend l'espace de recherche plus petit et plus facile à explorer.

Les représentations disponibles pour le chromosome sont notamment:

**Binaire:** C'est la méthode la plus courante qui peut représenter un chromosome comme une chaîne binaire. Chaque chromosome est représenté par une chaîne de zéros et de uns tableau (4.1) [150].

**Tableau 4.1 :** Codage binaire

Chromosome 1	1 1 0 1 0 0 0 1 1 0 1 0
Chromosome 2	0 1 1 1 1 1 1 1 1 0 0



**Permutation:** Chaque chromosome est une chaîne de nombres. Parfois des corrections doivent être effectuées une fois l'opération génétique terminée. Dans le codage par permutation, chaque chromosome est une chaîne de valeurs entières ou réelles comme indiqué sur le tableau (4.2). Le codage par permutation n'est utile que pour résoudre des problèmes.

**Tableau 4.2 :** Codage de permutation

Chromosome A	1 5 3 2 6 4 7 9 8
Chromosome B	8 5 6 7 2 3 1 4 9

**Valeur:** Chaque chromosome est une chaîne de valeurs qui peuvent être connectées au problème comme indiqué sur le tableau (4.3). Ce codage produit les meilleurs résultats pour certains problèmes particuliers. D'autre part, il est souvent nécessaire de mettre au point un nouvel opérateur génétique spécifique au problème. Le codage de valeur directe peut être utilisé dans des problèmes complexes. L'utilisation du codage binaire pour ce type de problèmes est très difficile.

**Tableau 4.3 :** Valeurs codées

Chromosome A	1.2324 5.3243 0.4556 2.3293 2.4545
Chromosome B	ABDJEIFJDHDIERJFDLDFLFEGT
Chromosome C	(blanc). (retour). (droite). (avant). (gauche)

### 4.3.3 Évaluation de population

Chacun des chromosomes d'une génération doit être évalué pour le processus de sélection. Ceci est accompli en examinant le score de chaque gène dans le chromosome en ajoutant les scores et en faisant la moyenne du score pour le chromosome. Dans le cadre du processus d'évaluation, le chromosome d'élite de la génération est déterminé.

### 4.3.4 Sélection et reproduction

La sélection est le processus consistant à choisir deux parents parmi la population après avoir choisi un codage, l'étape suivante consiste à décider comment effectuer la sélection. C'est-à-dire comment choisir des individus dans la population qui créent des enfants pour la prochaine génération. Les chromosomes sont choisis parmi la population initiale pour être

parents pour la reproduction. Le problème est de savoir comment sélectionner ces chromosomes. Selon La théorie de l'évolution de Darwin, les meilleures survécuent pour créer une nouvelle progéniture. La figure (3.10) montre le processus de sélection de base.

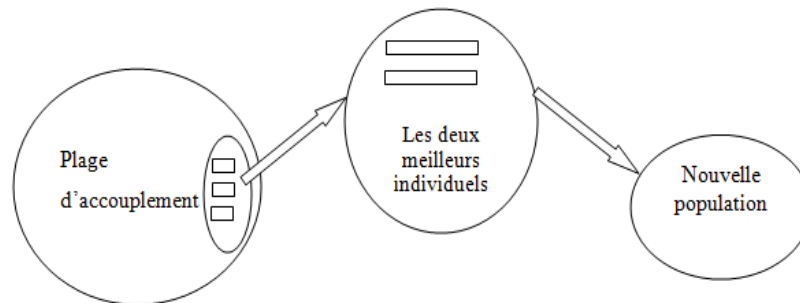


Figure 4.2 : Diagramme de la sélection [150].

### 4.3.5 Croisement

Le Croisement est le processus consistant à prendre deux solutions mères et produire un enfant après le processus de sélection, la population s'enrichit de meilleurs individus. La reproduction fait des clones de bonnes chaînes mais ne crée pas les nouvelles, l'opérateur de croisement est appliqué dans le but de créer une meilleure progéniture. Le croisement est un opérateur de recombinaison qui se déroule en trois étapes:

- L'opérateur de reproduction sélectionne au hasard une paire de deux chaînes individuelles pour l'accouplement.
- Un site transversal est sélectionné au hasard sur la longueur de la chaîne.
- Enfin, les valeurs de position sont échangées entre les deux chaînes à la suite du croisement.

Il existe différentes techniques de croisement tel que :

#### 4.3.5.1 Croisement en un point :

L'algorithme génétique traditionnel utilise un croisement en un seul point où les deux chromosomes sont coupés une fois aux points correspondants et les sections après les coupes sont échangées. La figure (4.2) ci-dessous illustre un croisement en un point et on peut observer que les bits à côté du point de croisement sont échangés pour produire des enfants.

**Tableau 4.4** : Croisement à un point [150]

<b>Parent 1</b>	<b>1 0 1 1 0</b>	<b>0 1 0</b>
<b>Parent 2</b>	<b>1 0 1 0 1</b>	<b>1 1 1</b>

↓

<b>Enfant 1</b>	<b>1 0 1 1 0</b>	<b>1 1 1</b>
<b>Enfant 2</b>	<b>1 0 1 0 1</b>	<b>0 1 0</b>

#### 4.3.5.2 Croisement en deux points

De nombreux algorithmes de croisement ont été mis au point et conçus souvent avec plus d'un point de coupure.

Il convient de noter que l'ajout de nouveaux points de croisement réduit les performances de l'algorithme génétique. Le problème avec l'ajout de points de croisement supplémentaire est que les blocs de construction sont plus susceptibles d'être perturbés. Cependant, l'avantage d'avoir plus de points de croisement est que l'espace de problèmes peut être recherché plus profondément.

#### 4.3.6 Mutation

La mutation est une étape qui empêche l'algorithme d'être piégé dans un minimum local. La mutation joue le rôle de récupérer le matériel génétique perdu ainsi que pour perturber l'information génétique de manière aléatoire. La mutation est traditionnellement considérée comme un simple opérateur de recherche. Si le croisement est censé exploiter la solution actuelle pour trouver une meilleure, la mutation est censée aider à l'exploration de tout l'espace de recherche.

La mutation est considérée comme un opérateur de fond pour maintenir la diversité génétique dans la population. Elle introduit de nouvelles structures génétiques dans la population en modifiant au hasard certains de ses blocs de construction. Elle aide à échapper aux pièges des minima locaux et maintient la diversité de la population.

## 4.4 Application d'une commande robuste pour le suivi de trajectoires

Les modèles cinématique et dynamique utilisés dans ce travail sont considérés comme ceux présentés dans le chapitre 3. Les étapes du contrôle proposé sont basées sur deux contrôleurs, un contrôleur cinématique et un contrôleur dynamique utilisant l'algorithme AG.

### 4.4.1 Commande cinématique par mode glissant flou

La conception du contrôleur cinématique est basée essentiellement sur deux étapes :

- Contrôle de la vitesse angulaire
- Contrôle de la vitesse linéaire

#### 4.4.1.1 Contrôle de la vitesse angulaire

En utilisant la surface de glissement PD, la sélection de la fonction de commutation est donnée comme suit:

$$s_1 = \rho_1 e_1 + \rho_2 \dot{e}_1 \quad (4.7)$$

où  $e_1 = \theta_e$

La dérivée de la première surface de glissement est:

$$\dot{s}_1 = \rho_1 \dot{\theta}_e + \rho_2 \ddot{\theta}_e \quad (4.8)$$

Dans le but d'atténuer le problème de broutement, la fonction de saturation est donnée par:

$$\dot{s}_1 = -k_1 \text{sat}(s_1) \quad (4.9)$$

La loi de contrôle de la vitesse angulaire en utilisant les équations (4.8), (4.9) et le modèle cinématique du robot (3.24) est donnée comme suit:

$$\begin{aligned} -k_1 \text{sat}(s_1) &= \rho_1 \dot{\theta}_e + \rho_2 \ddot{\theta}_e \\ -k_1 \text{sat}(s_1) &= \rho_1 (\omega_r - \omega) + \rho_2 (\dot{\omega}_r - \dot{\omega}) \end{aligned}$$

$$\dot{\omega}_k = \frac{1}{\rho_2} (\rho_1 \omega_r - \rho_1 \omega_k + \rho_2 \dot{\omega}_r + k_1 \text{sat}(s_1)) \quad (4.10)$$

Cette loi de contrôle peut faire converger l'erreur d'orientation  $\theta_e$  vers zéro. Pour éliminer l'effet des perturbations angulaires, un système flou est appliqué. Le paramètre  $k_I$  est choisi avec le système flou pour compenser l'effet des perturbations, pour réduire le problème de broutement et pour assurer la stabilité du système [151].

Pour sélectionner la valeur de  $k_I$ , il faut déterminer la relation entre  $k_I$  et la surface de glissement  $s_1$ . Dans ce cas, les entrées du système flou sont  $s_1 \dot{s}_1$  et la sortie est  $k_f$ .

Les ensembles flous des entrées et des sorties ainsi que les règles sont choisies de la même manière que ceux du chapitre 3.

Le paramètre  $k_I$  est estimé par:

$$k_f = G \int_0^t \Delta k_1(t) dt \quad (4.11)$$

Par conséquent, la loi de contrôle (4.11) devient comme suit:

$$\dot{\omega}_k = \frac{1}{\rho_2} (\rho_1 \omega_r - \rho_1 \omega_k + \rho_2 \dot{\omega}_r + k_f \text{sat}(s_1)) \quad (4.12)$$

Avec  $k_f$  représente un gain choisi par la logique floue.

#### 4.4.1.2 Contrôle de la vitesse linéaire

Dans le but de concevoir un contrôleur de la vitesse linéaire, le mode de glissement est appliqué sur le modèle cinématique du robot. Lorsque l'erreur d'orientation tend vers zéro  $\omega = \omega_r$  et le modèle d'erreur du robot (3.24) devient:

$$\dot{x}_e = \omega_r y_e - v + v_r - \dot{v} \quad (4.13)$$

$$\dot{y}_e = -\omega_r x_e \quad (4.14)$$

La deuxième surface de glissement est sélectionnée comme suit:

$$s_2 = x_e - y_e \quad (4.15)$$

En utilisant le modèle de la loi d'arrivé donnée par:

$$\dot{s}_2 = -k_2 \frac{s_2}{|s_2| + \delta_1} - \alpha s_2 \quad (4.16)$$

En introduisant (4.15) et (4.16), la loi de contrôle de la vitesse linéaire peut être obtenue comme suit:

$$\dot{s}_2 = \dot{x}_e - \dot{y}_e = v_r + \omega_r x_e + \omega_r y_e - v = -k_2 \frac{s_2}{|s_2| + \delta_1} - \alpha s_2 \quad (4.17)$$

$$v_k = v_r + \omega_r x_e + \omega_r y_e + \alpha s_2 + k_2 \frac{s_2}{|s_2| + \delta_1} \quad (4.18)$$

Les surfaces de commutation sélectionnées sont indiquées comme suit:

$$s = \begin{bmatrix} s_1 \\ s_2 \end{bmatrix} = \begin{bmatrix} \rho_1 \theta_e + \rho_2 \dot{\theta}_e \\ x_e - y_e \end{bmatrix} \quad (4.19)$$

La loi de contrôle des vitesses linéaire et angulaire est donnée sous la forme suivante :

$$\begin{bmatrix} v_k \\ \dot{\omega}_k \end{bmatrix} = \begin{bmatrix} v_r + \omega_r x_e + \omega_r y_e + \alpha s_2 + k_2 \frac{s_2}{|s_2| + \delta_1} \\ \frac{1}{\rho_2} (\rho_1 \omega_r - \rho_1 \omega_k + \rho_2 \dot{\omega}_r + k_f \text{sat}(s_1)) \end{bmatrix} \quad (4.20)$$

Pour garantir la stabilité asymptotique du système, une fonction candidate de Lyapunov est sélectionnée.

$$L_s = \frac{1}{2} s_1^2 \quad (4.21)$$

$$\dot{L}_s = s_1 \dot{s}_1 = -k_f |s_1| - \rho_1 d \omega s_1 - \rho_2 d \dot{\omega} s_1 \quad (4.22)$$

où  $k_f > d\omega$

$$\dot{L}_s \leq 0 \quad (4.23)$$

La fonction candidate de Lyapunov est définie négative ce qui implique que les erreurs de suivi convergent asymptotiquement vers zéro.

## 4.4.2 Commande robuste par mode glissant terminal basée sur AG

### 4.4.2.1 Commande dynamique sans AG

Dans cette section, il est intéressant de concevoir les couples de contrôle du robot dans l'objectif de faire converger asymptotiquement l'erreur de vitesse vers zéro.

$$\lim_{t \rightarrow \infty} V_e = \lim_{t \rightarrow \infty} \|V_c(t) - V(t)\| = 0 \quad (4.24)$$

Considérant le modèle dynamique suivant :

$$M(q)\dot{V} = \beta(q)\tau + \tilde{D}(t) \quad (4.25)$$

L'erreur de la vitesse est donnée comme suit :

$$E_v = [e_v \quad e_\omega]^T = \begin{bmatrix} v_k - v \\ \omega_k - \omega \end{bmatrix} \quad (4.26)$$

La dérivée de l'erreur est donnée comme:

$$\dot{E}_v = \begin{bmatrix} \dot{v}_k - \dot{v} \\ \dot{\omega}_k - \dot{\omega} \end{bmatrix} \quad (4.27)$$

La conception de la loi de commande consiste dans le choix de la surface de glissement, alors que la surface de glissement choisie est en fonction de l'erreur entre les vitesses du robot :

$$S = \begin{bmatrix} s_3 \\ s_4 \end{bmatrix} = \begin{bmatrix} e_v \\ e_\omega \end{bmatrix} = \begin{bmatrix} v_k - v \\ \omega_k - \omega \end{bmatrix} \quad (4.28)$$

La dérivée des surfaces glissantes est donnée comme suit:

$$\dot{S} = \begin{bmatrix} \dot{s}_3 \\ \dot{s}_4 \end{bmatrix} = \begin{bmatrix} \dot{v}_k - \dot{v} \\ \dot{\omega}_k - \dot{\omega} \end{bmatrix} \quad (4.29)$$

En utilisant la fonction de glissement terminale rapide (FTF), la forme de la surface de glissement proposée est la suivante :

$$\dot{s} + \lambda_1 s^{\eta_1} + \lambda_2 s^{\eta_2} = 0 \quad (4.30)$$

où  $0 < \eta_1 < 1, \eta_2 > 1, \eta_1, \eta_2 \in \mathfrak{R}$  et  $\lambda_1, \lambda_2 > 0$

La surface de glissement (4.30) est choisie en fonction de l'erreur de vitesse angulaire:

$$\dot{s}_4 = -\lambda_1 s_4^{\eta_1} - \lambda_2 s_4^{\eta_2} \quad (4.31)$$

En utilisant la loi de contrôle par mode de glissement pour l'erreur de vitesse linéaire:

$$\dot{s}_3 = -k_3 \frac{s_3}{|s_3| + \sigma_2} \quad (4.32)$$

Les couples de contrôle en utilisant les équations (4.25) (4.31) et (4.32) sont obtenus comme suit:

$$\begin{bmatrix} \dot{s}_3 \\ \dot{s}_4 \end{bmatrix} = \begin{bmatrix} \dot{v}_k - \dot{v} \\ \dot{\omega}_k - \dot{\omega} \end{bmatrix} = \begin{bmatrix} -k_3 \frac{s_3}{|s_3| + \sigma_2} \\ -\lambda_1 s_4^{\eta_1} - \lambda_2 s_4^{\eta_2} \end{bmatrix} \quad (4.33)$$

$$\begin{bmatrix} \dot{v}_k \\ \dot{\omega}_k \end{bmatrix} - \frac{\beta(q)}{M(q)} \begin{bmatrix} \tau_1 \\ \tau_2 \end{bmatrix} = \begin{bmatrix} -k_3 \frac{s_3}{|s_3| + \sigma_2} \\ -\lambda_1 s_4^{\eta_1} - \lambda_2 s_4^{\eta_2} \end{bmatrix} \quad (4.34)$$

$$\tau_1 = \frac{R_a}{2} \left( m \left( k_3 \frac{s_3}{|s_3| + \sigma_2} + \dot{v}_k \right) + \frac{I}{L} \left( \dot{\omega}_k + \lambda_1 s_4^{\eta_1} + \lambda_2 s_4^{\eta_2} \right) \right) \quad (4.35)$$

$$\tau_2 = \frac{R_a}{2} \left( m \left( k_3 \frac{s_3}{|s_3| + \sigma_2} + \dot{v}_k \right) - \frac{I}{L} \left( \dot{\omega}_k + \lambda_1 s_4^{\eta_1} + \lambda_2 s_4^{\eta_2} \right) \right) \quad (4.36)$$



Cette loi de contrôle peut être écrite sous la forme suivante:

$$\tau = \begin{pmatrix} \tau_1 \\ \tau_2 \end{pmatrix} = \frac{R_a}{2} \left( A \begin{pmatrix} \dot{v}_k \\ \dot{\omega}_k \end{pmatrix} + A \begin{pmatrix} k_3 \frac{s_3}{|s_3| + \sigma_2} \\ \lambda_1 s_4^{\eta_1} + \lambda_2 s_4^{\eta_2} \end{pmatrix} \right) \quad (4.37)$$

$$\text{où } A = \begin{pmatrix} m & \frac{I}{L} \\ m & -\frac{I}{L} \end{pmatrix}$$

Pour assurer la stabilité du système, la fonction de Lyapunov est sélectionnée comme suit:

$$V_2 = \frac{1}{2} S^T S \quad (4.38)$$

Une condition suffisante pour garantir l'attractivité en temps fini de la surface de glissement  $s_4$  est de considérer:

$$s_4 \dot{s}_4 = -\lambda_1 |s_4|^{\eta_1+1} - \lambda_2 |s_4|^{\eta_2+1} \leq 0 \quad (4.39)$$

La dérivée temporelle de la fonction de Lyapunov (4.38) est donnée comme:

$$\begin{aligned} \dot{V}_2 = S^T \dot{S} = s_3 \dot{s}_3 + s_4 \dot{s}_4 = & -k_3 \frac{s_3^2}{|s_3| + \sigma_2} - \lambda_1 s_4^{\eta_1+1} - \lambda_2 s_4^{\eta_2+1} - S^T M(q)^{-1} \tilde{D} = \\ & -k_3 |s_3| - \lambda_1 |s_4|^{\eta_1+1} - \lambda_2 |s_4|^{\eta_2+1} - S^T M(q)^{-1} \tilde{D} \leq 0 \end{aligned} \quad (4.40)$$

où  $k_3, \lambda_1, \lambda_2$  sont supposés être supérieur à  $\tilde{D}$ . La dérivé de la fonction de Lyapunov est définie négative avec des valeurs suffisamment grandes de  $\lambda_1, \lambda_2$ , donc le système est asymptotiquement stable. Dans le but d'avoir une convergence rapide de l'équation (4.31), les paramètres  $\lambda_1, \lambda_2, \eta_1$  et  $\eta_2$  sont sélectionnés par l'algorithme génétique (AG) et les valeurs optimales sont obtenues afin d'accélérer l'erreur de la vitesse angulaire à converger rapidement vers zéro.

#### 4.4.2.2 Commande dynamique avec AG

L'algorithme génétique (AG) est une méthode de recherche robuste utilisée pour trouver des solutions approchées par optimisation des problèmes [152]. Le digramme de contrôle est donné sur la figure (4.3).

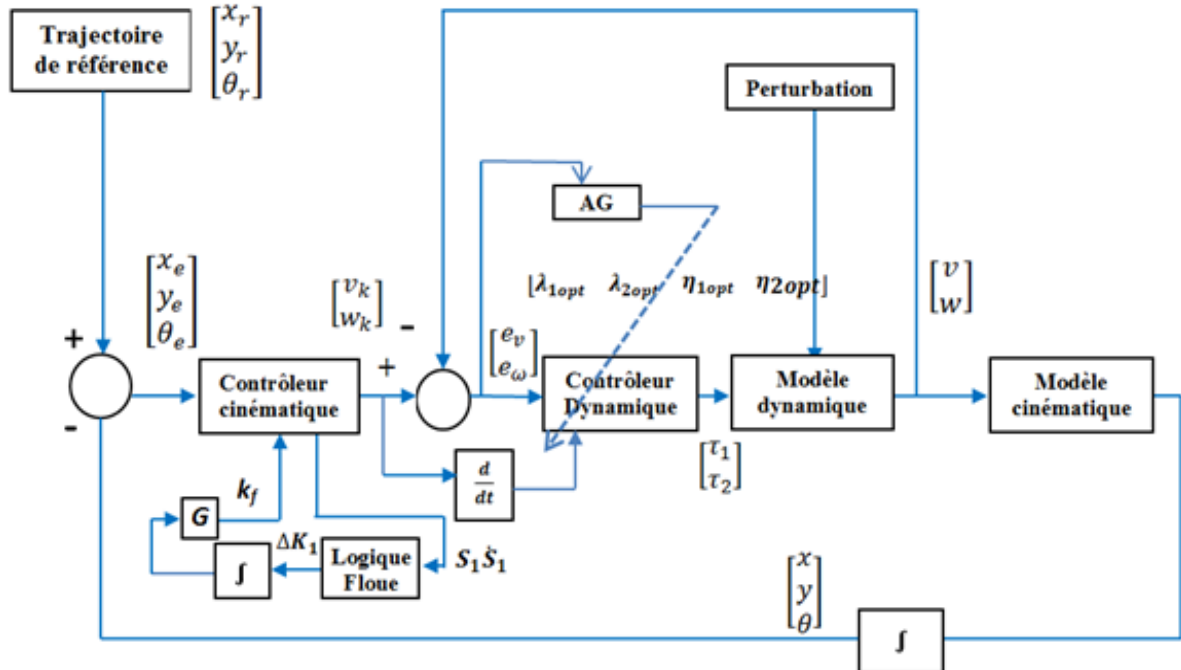


Figure 4.3: Diagramme de contrôle basé sur AG

La structure de l'algorithme génétique utilisée est donnée par les étapes suivantes:

- Produire une population initiale de chromosomes à laquelle on donne une solution appropriée.
- Reformuler la forme physique de tout chromosome de la population.
- Produire une nouvelle population en répétant les étapes jusqu'à ce que la nouvelle population soit complète.
- Choisir deux parents de chromosomes d'une population en fonction de leur forme physique.
- Croiser les parents pour avoir un nouveau-né, si la condition n'est pas satisfaite, la progéniture est la même que celle des parents.
- Mutation de la nouvelle progéniture à chaque position.
- Mettre la nouvelle progéniture dans une nouvelle population.

Finalement, choisir les nouvelles populations de parents, si la condition finale est vérifiée, arrêter et retourner à la solution parfaite de la population. L'opération précédente est répétée jusqu'à ce que la condition soit vérifiée [153].

Un organigramme général de la mise en œuvre de la technique est présenté sur la figure (4.4).

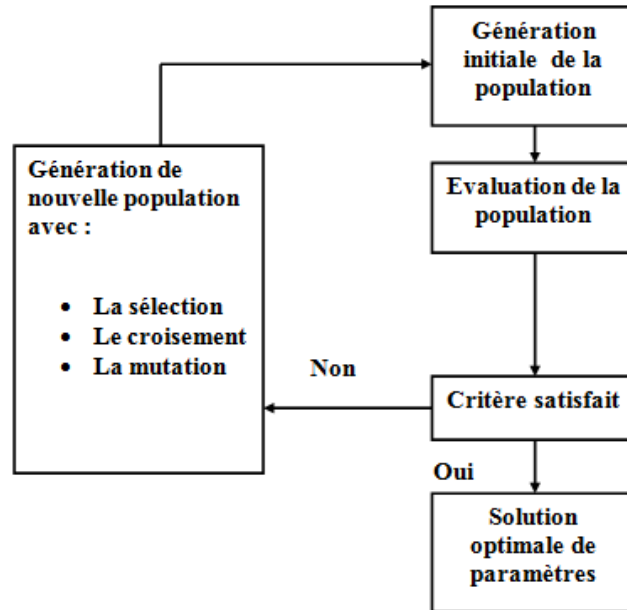


Figure 4.4 : Organigramme de l'algorithme génétique

Dans ce travail, les paramètres à optimiser sont  $(\lambda_1, \lambda_2, \eta_1, \eta_2)$  pour qu'on puisse obtenir de bonnes performances globales pour le système contrôlé. Il est souhaitable d'avoir la vitesse d'atteinte rapide dans l'hyper plan de commutation pendant la phase d'atteinte et que l'état correspondant glisse vers l'origine. En utilisant l'algorithme génétique, la fonction objective choisie est définie comme :

$$f = \frac{1}{N} \sum_{j=1}^N \|e_w(j)\|^2 \quad (4.41)$$

Lorsque la fonction objective est définie, la prospection de l'AG dépendra des exigences de cette fonction. Le choix de la fonction objective est une étape importante qui permet au système d'atteindre les performances souhaitées. Les paramètres de gain  $(\lambda_1, \lambda_2, \eta_1, \eta_2)$  du contrôleur dynamique sont choisis de manière optimale.

L'algorithme commence par choisir des intervalles de paramètres  $\lambda_1 = [\lambda_{11}, \lambda_{12}]$ ,  $\lambda_2 = [\lambda_{21}, \lambda_{22}]$ ,  $\eta_1 = [\eta_{12}, \eta_{12}]$  et  $\eta_2 = [\eta_{21}, \eta_{22}]$ . Les valeurs optimales obtenues ( $\lambda_{1opt}, \lambda_{2opt}, \eta_{opt1}, \eta_{opt2}$ ) par l'algorithme élaboré minimisent la fonction objective  $f$ .

En sélectionnant les paramètres optimaux  $\lambda_{1opt}, \lambda_{2opt}, \eta_{opt1}, \eta_{opt2}$  la loi de contrôle (4.38) devient comme suit:

$$\tau = \begin{pmatrix} \tau_1 \\ \tau_2 \end{pmatrix} = \frac{R_a}{2} \left( A \begin{pmatrix} \dot{v}_k \\ \dot{\omega}_k \end{pmatrix} + A \begin{pmatrix} k_3 \frac{s_3}{|s_3| + \sigma_2} \\ \lambda_{1opt} s_4^{\eta_{opt1}} + \lambda_{2opt} s_4^{\eta_{opt2}} \end{pmatrix} \right) \quad (4.42)$$

Pour vérifier la stabilité du système, de nouveaux paramètres sont choisis et la dérivée de la fonction de Lyapunov est donnée comme:

$$\begin{aligned} \dot{V}_2 = S^T \dot{S} &= -k_3 \frac{s_3^2}{|s_3| + \sigma_2} - \lambda_{1opt} s_4^{\eta_{opt1}+1} - \lambda_{2opt} s_4^{\eta_{opt2}+1} - S^T M(q)^{-1} \tilde{D} = \\ &-k_3 |s_3| - \lambda_1 |s_4|^{\eta_{opt1}+1} - \lambda_2 |s_4|^{\eta_{opt2}+1} - S^T M(q)^{-1} \tilde{D} \leq 0 \end{aligned} \quad (4.43)$$

La fonction de Lyapunov choisie est définie négative, donc les erreurs de vitesses convergent asymptotiquement vers zéro.

## 4.5 Travaux de simulation

Le logiciel Matlab/Simulink est sélectionné pour simuler les lois de contrôle proposées, et illustré le comportement du robot mobile. Dans cette section on évalue par simulation la capacité des contrôleurs afin d'apercevoir comment réagit le robot selon les trajectoires désirées. Différentes trajectoires de référence sont choisies entre autres une trajectoire sinusoïdale, une trajectoire circulaire et une trajectoire spécifique. Dans les travaux de simulations, les vitesses angulaires et linéaires souhaitées sont choisies comme suit:  $v_r=2$ ,  $\omega_r=1$ . Les paramètres du robot mobile sont choisis comme suit:  $m=4.5kg$ ,  $I=3kg.m^2$ ,  $R_a=0.03m$ ,  $L=0.15m$ .

Les paramètres du contrôleur cinématique sont choisis aléatoirement.

**Tableau 4.5 :** Paramètres du contrôleur cinématique

Paramètres	valeurs
$k_2$	25
$\rho_1$	20
$\rho_2$	0.1
$\alpha$	20
$G$	25
$\sigma_1$	0.9

Les perturbations cinématiques sont introduites durant un temps  $3 < t < 4$  secondes et sont données comme suit:

$$d\omega = 1.5 \sin(t - \pi) u(t) \quad (4.44)$$

$$dv = 1.5 \sin(t - \pi) u(t) \quad (4.45)$$

$$\text{où } u(t) = \begin{cases} 1 & \text{for } 3 < t < 4 \\ 0 & \text{ailleurs} \end{cases}$$

**Tableau 4.6 :** Paramètres du contrôleur dynamique sans AG

Paramètres	$\lambda_1$	$\lambda_2$	$\eta_1$	$\eta_2$	$k_3$	$\sigma_2$
valeurs	6	6	0.9	2.5	25	0.95

Nous avons utilisé l'AG avec les paramètres suivants: taille de la population égale à 20, nombre maximal de génération égale à 100, dimension = 4, probabilité de croisement = 0,85 et probabilité de mutation = 0,01. Les paramètres obtenus par l'algorithme AG sont bornés :  $0 < \lambda_1 < 30, 0 < \lambda_2 < 30, 0 < \eta_1 < 1, 0 < \eta_2 < 3$ .

**Tableau 4.7 :** Paramètres du contrôleur dynamique en utilisant AG

Paramètres	Circulaire	Sinusoïdale	Aléatoire
$\lambda_{opt1}$	18,3378	17,7260	17.7460
$\lambda_{opt2}$	19,3745	26,8204	26.9736
$\eta_{opt1}$	0,5264	0.9445	09993
$\eta_{opt2}$	2,7752	2,9923	2.9333
$Min(f)$	0.0494	0.0165	0.0152

Les perturbations dynamiques sont insérées dans le temps  $7 < t < 8$  secondes:

$$\tilde{D}(t) = \begin{bmatrix} 2 \sin(t) p(t) \\ 2 \sin(t) p(t) \end{bmatrix} \tag{4.46}$$

où  $p(t) = \begin{cases} 1 & 7 < t < 8 \\ 0 & \text{ailleurs} \end{cases}$

Les fonctions d'appartenance de l'entrée et de la sortie du système flou sont données sur les figures suivantes:

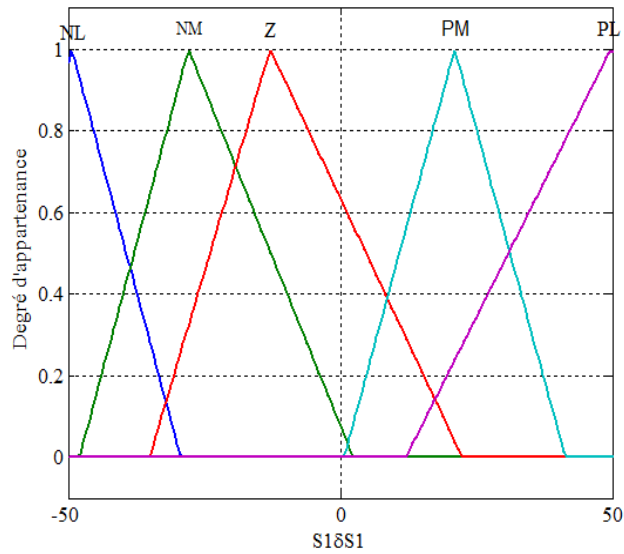


Figure 4.5: Ensembles flous d'entrée  $s_1, \dot{s}_1$

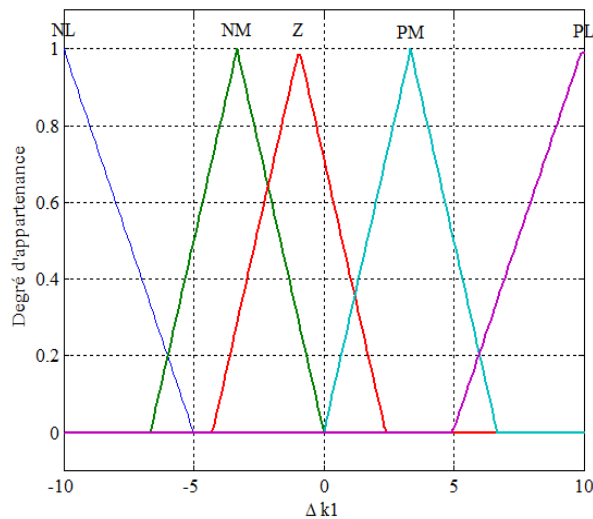


Figure 4.5: Ensembles flous de la sortie  $\Delta k_1$

### A-Simulations pour une trajectoire circulaire

La position initiale et l'erreur d'orientation sont considérées comme étant égales à  $(2m, 3m, \frac{\pi}{3} rad)$ .

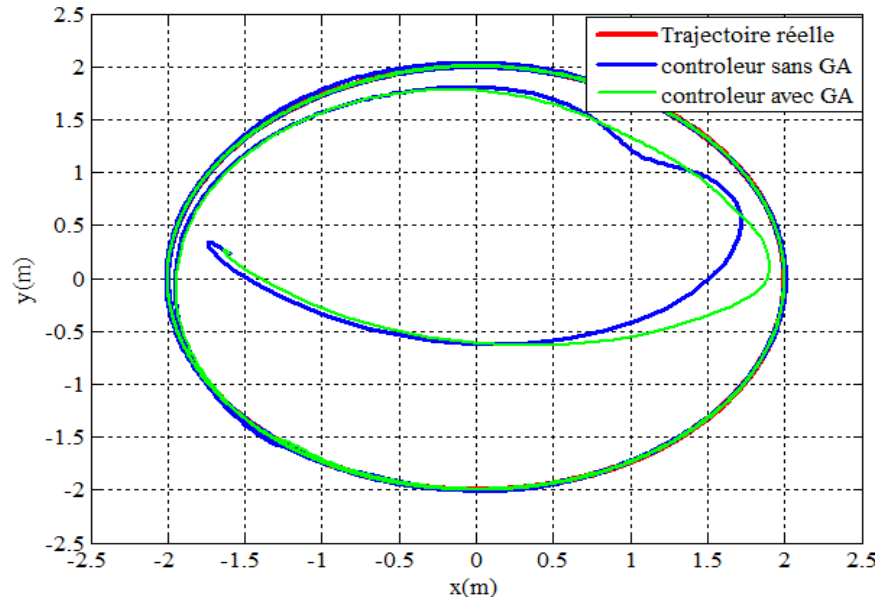


Figure 4.7: Poursuite d'une trajectoire circulaire

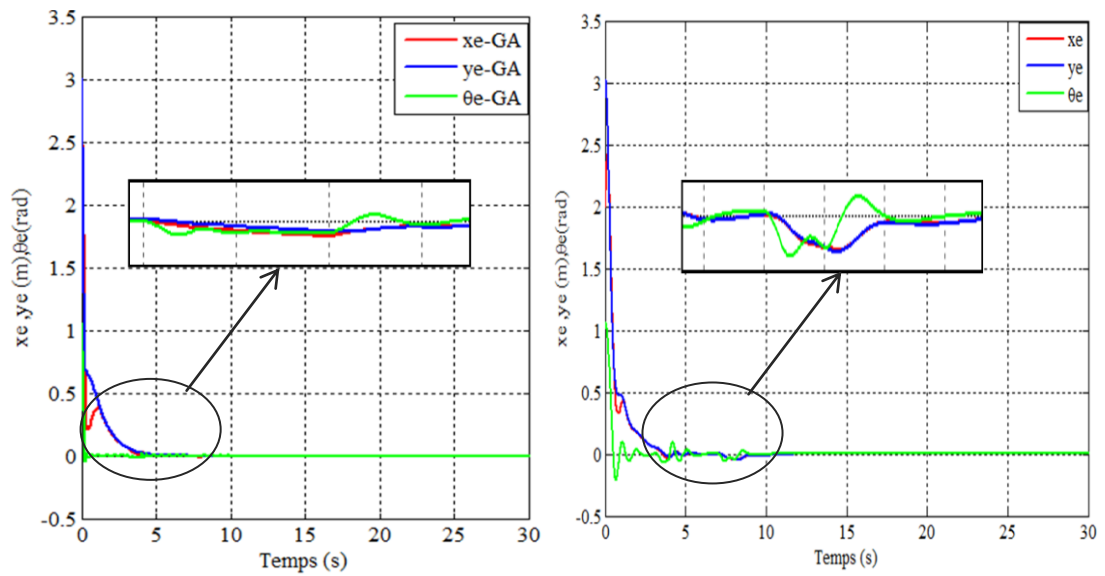


Figure 4.8: Erreurs de poursuite avec et sans AG

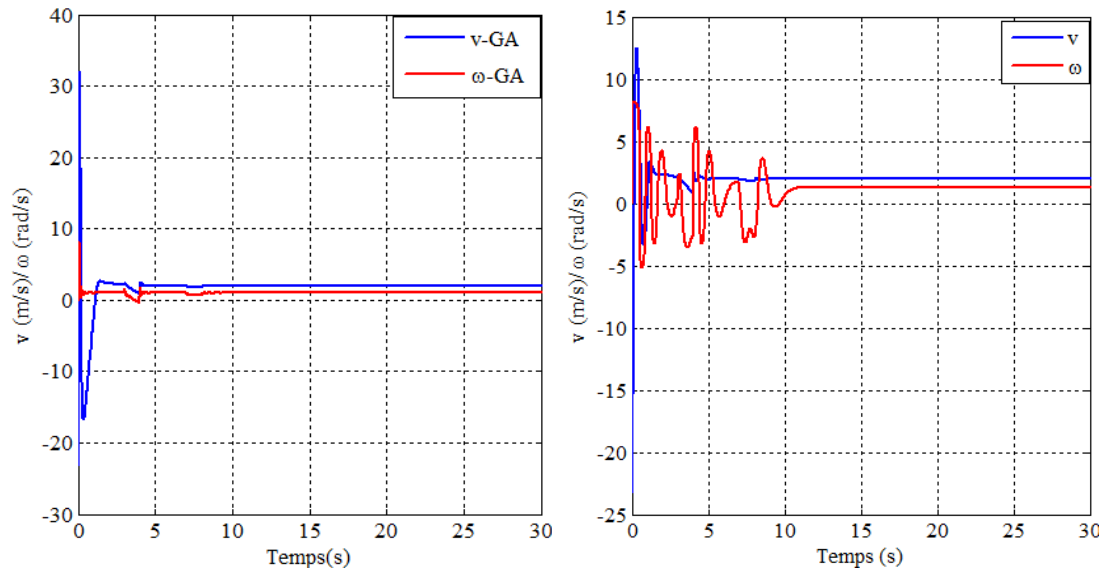


Figure 4.9 : Vitesses de commande  $v$  et  $\omega$  obtenus par contrôle dynamique avec et sans AG

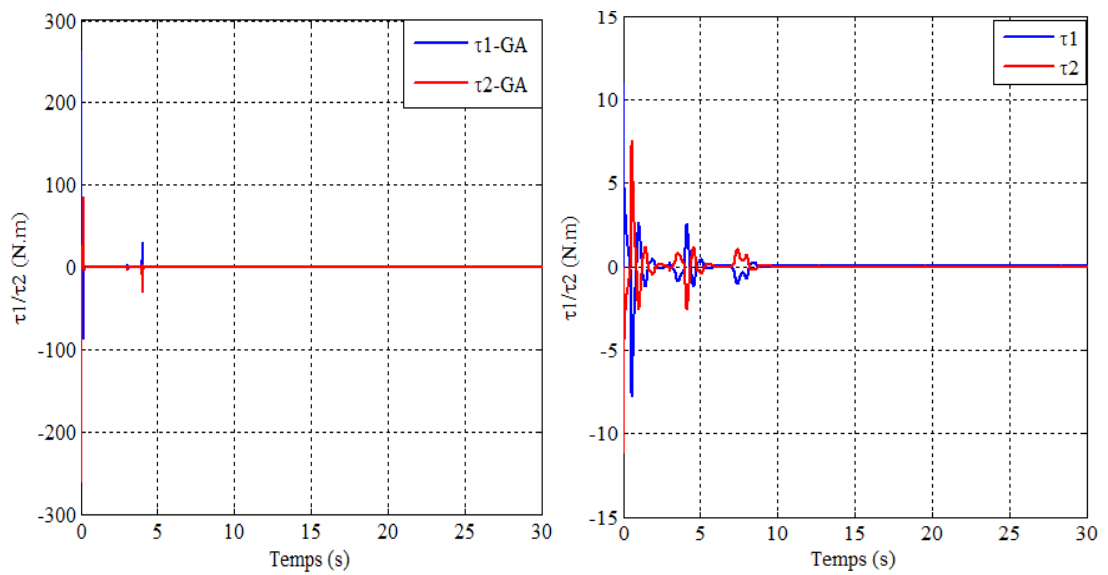


Figure 4.10: Couples  $\tau_1$  et  $\tau_2$  pour les deux cas avec et sans AG



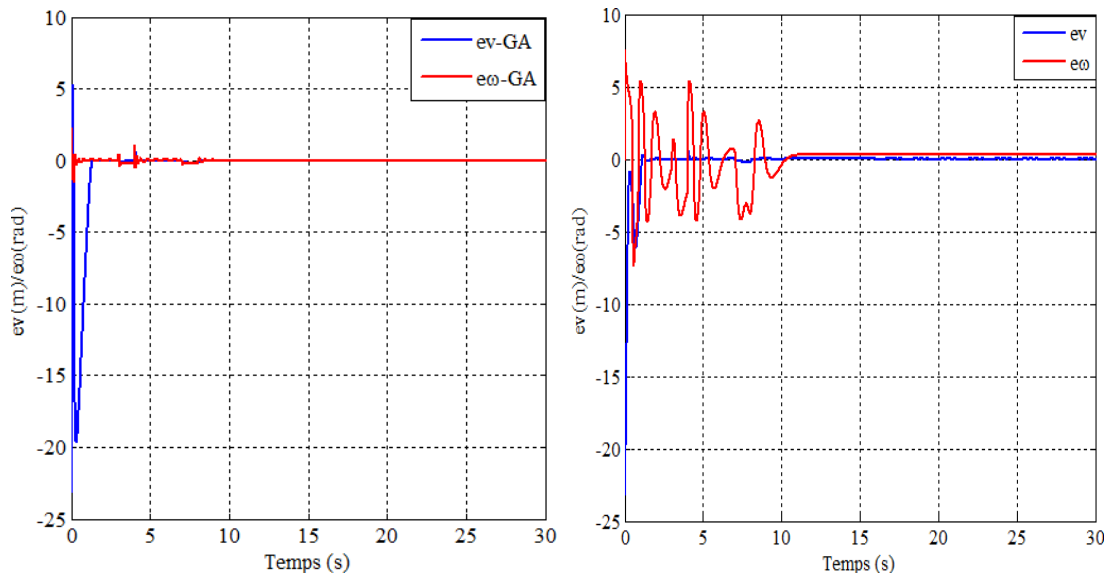


Figure 4.11: Erreurs de vitesses pour les deux cas avec et sans AG

On peut remarquer sur la figure (4.7) que le robot mobile peut atteindre rapidement la trajectoire circulaire en utilisant les deux contrôleurs dynamique avec et sans AG. On peut voir également que le temps de convergence vers la trajectoire de référence pour le contrôleur sans AG est égal à 6s après avoir inséré des perturbations cinématiques et égal à 9s après avoir inséré des perturbations dynamiques. Le temps de convergence avec AG est égal à 4s après insertion des perturbations cinématiques et à 7s après avoir insertion des perturbations dynamiques.

Les vitesses cinématiques du robot peuvent converger vers les vitesses de référence par utilisation du contrôle cinématique en un temps très court comme indiqué sur la figure (4.9). On remarque également que par utilisation de l'AG les vitesses cinématiques et dynamiques du robot convergent rapidement vers la référence en un temps  $t=8s$ , par contre sans l'AG elles convergent en un temps  $t=11s$ . Les couples du système convergent vers zéro comme indiqué sur la figure (4.10). La stabilité asymptotique du robot est donc assurée et les erreurs de suivi peuvent converger vers zéro.

### B-Simulations pour une trajectoire sinusoïdale

La position initiale et l'erreur d'orientation sont :  $(2m, 1m, \frac{\pi}{4} rad)$ .

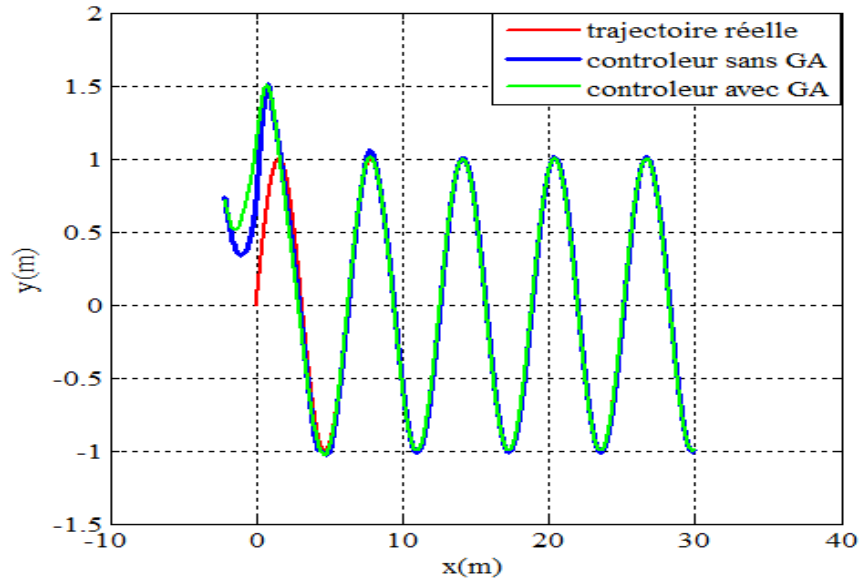


Figure 4.12: Poursuite d'une trajectoire sinusoïdale

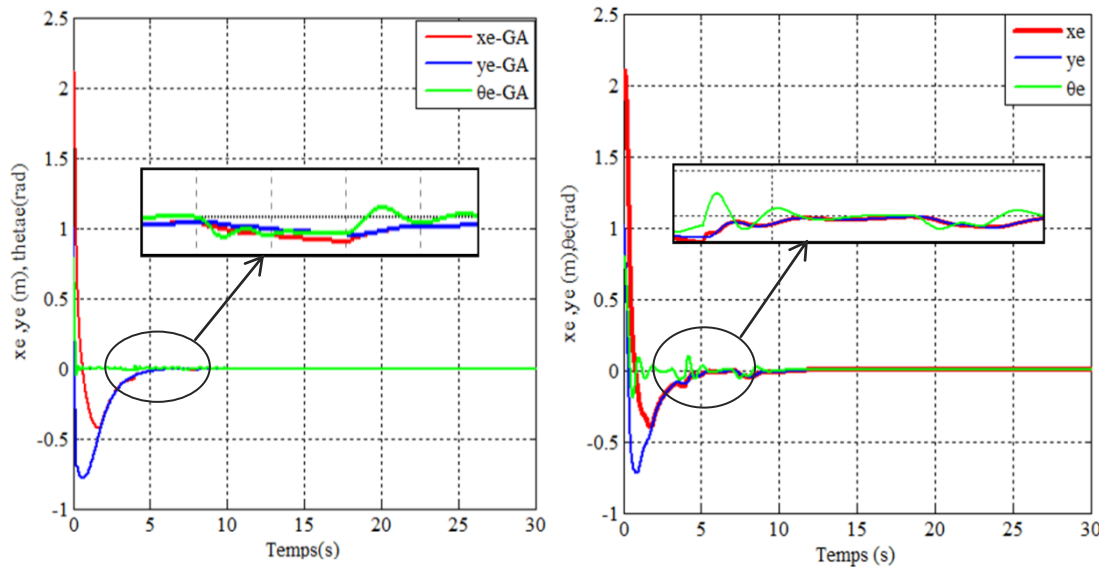


Figure 4.13: Erreurs de poursuite utilisant les lois dynamiques avec et sans AG

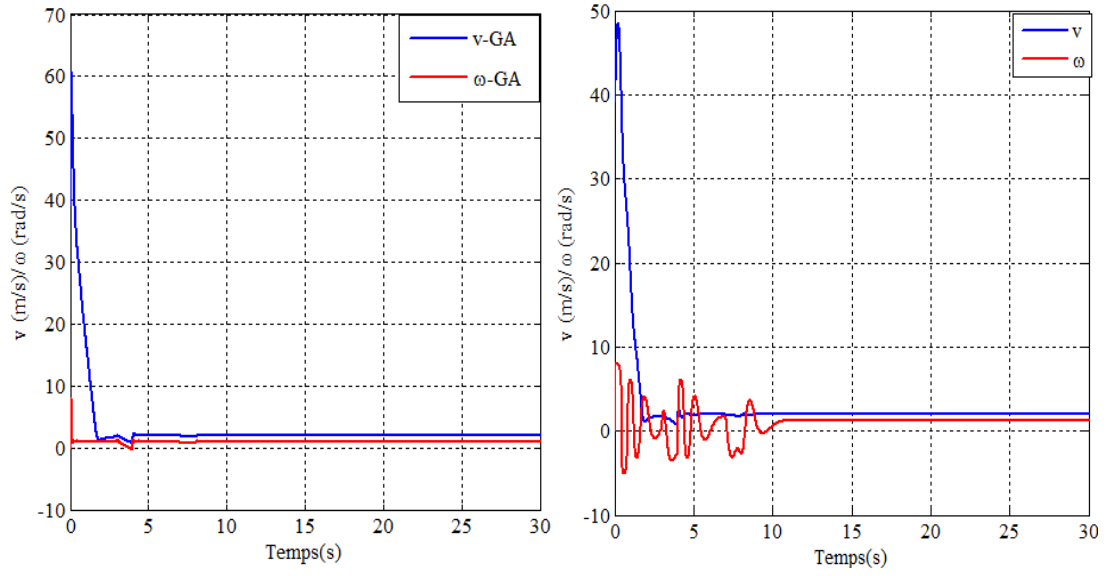


Figure 4.14: Vitesses de commande  $v$  et  $\omega$  obtenues par loi de contrôle dynamique avec et sans AG

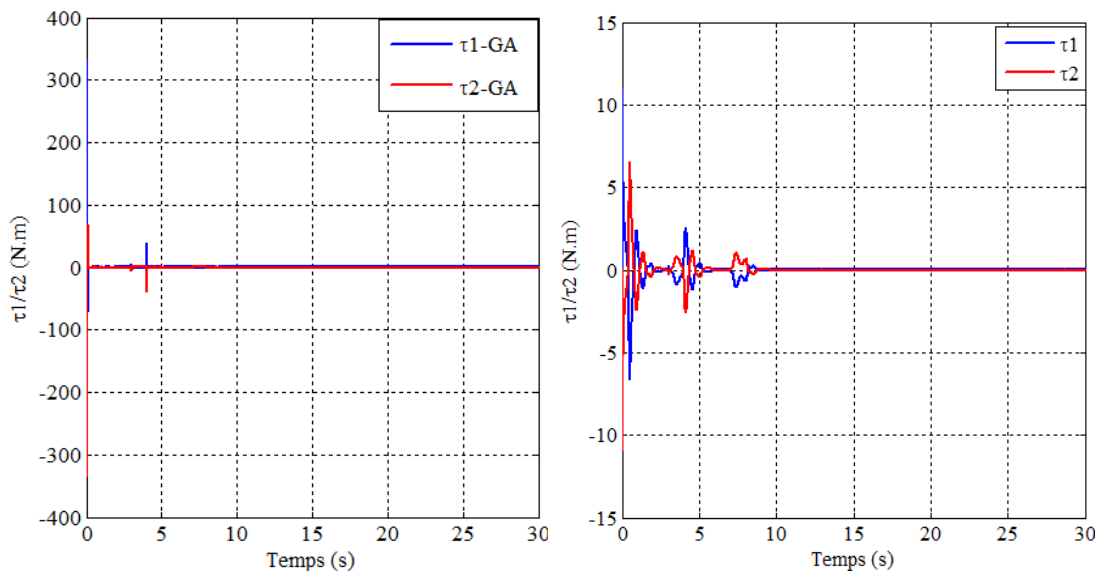


Figure 4.15: Couples  $\tau_1$  et  $\tau_2$

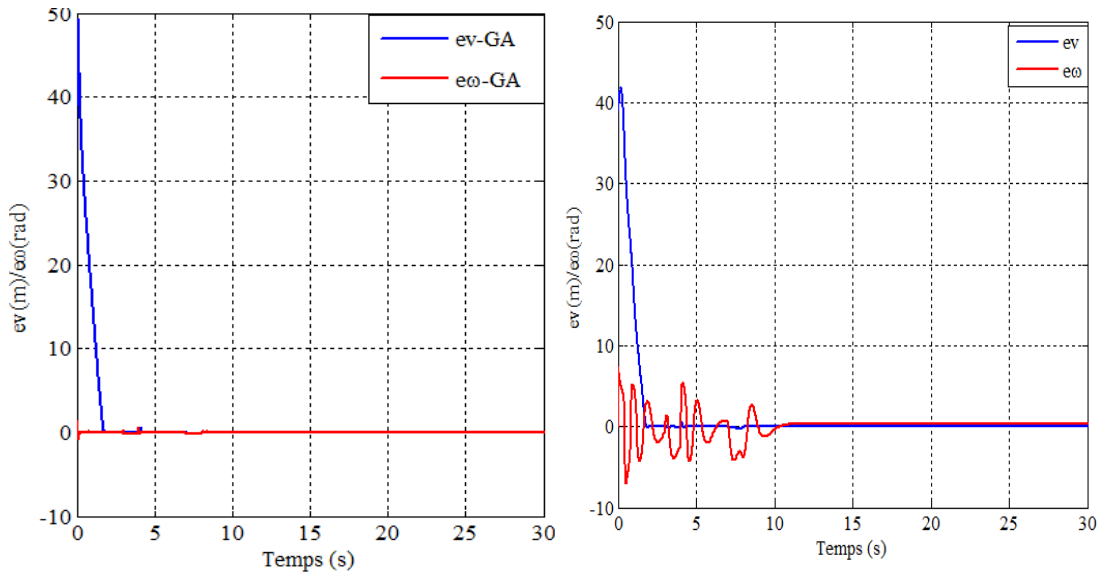


Figure 4.16: Erreurs de vitesses

On remarque que les résultats sont meilleurs en utilisant l'optimiseur AG sur une trajectoire sinusoïdale.

### C-Simulations pour une trajectoire aléatoire

La position initiale et l'erreur d'orientation sont:  $(2m, 3m, \frac{\pi}{3} rad)$

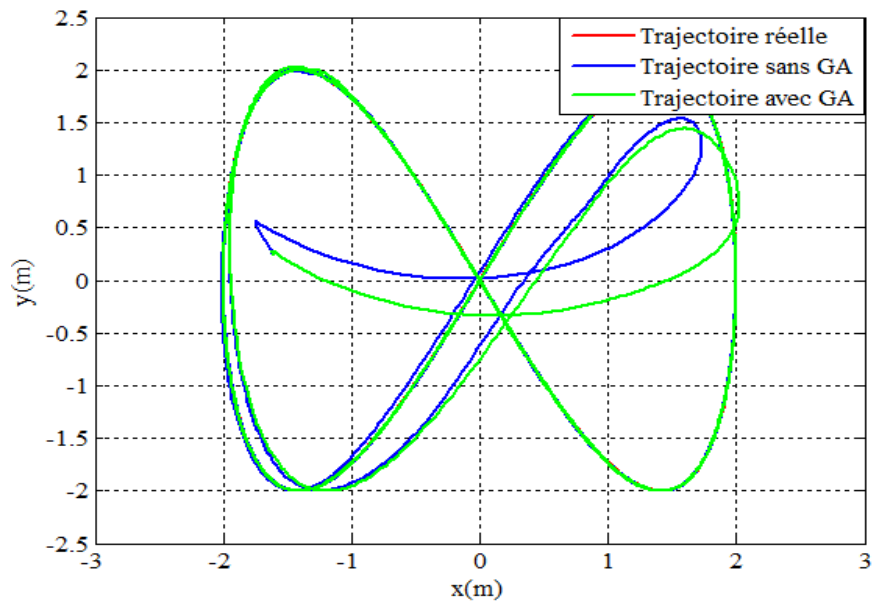


Figure 4.17: Poursuite d'une trajectoire aléatoire

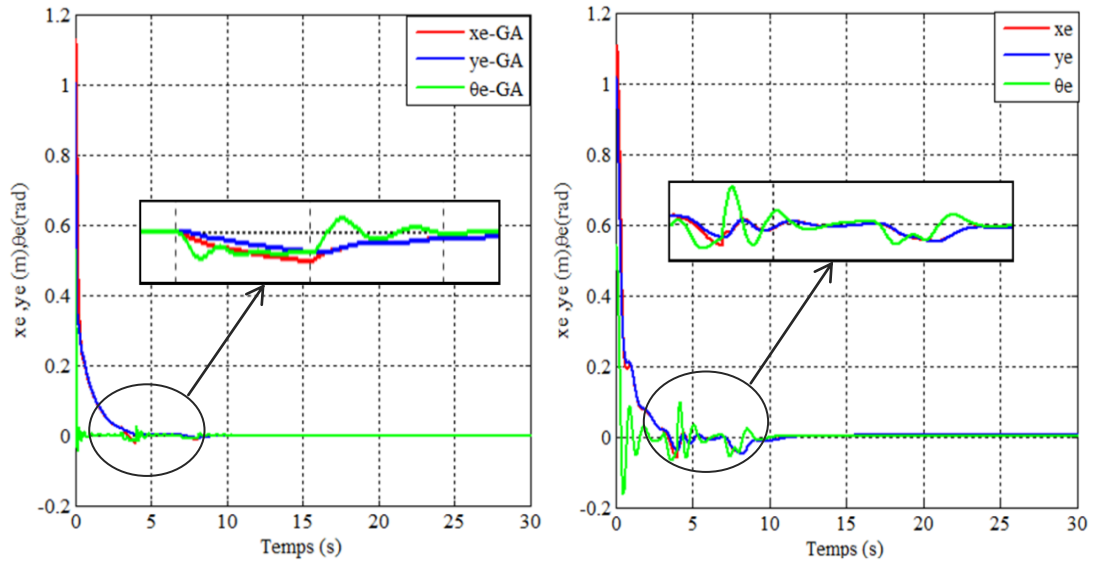


Figure 4.18: Erreurs de poursuite utilisant les lois dynamiques avec et sans AG

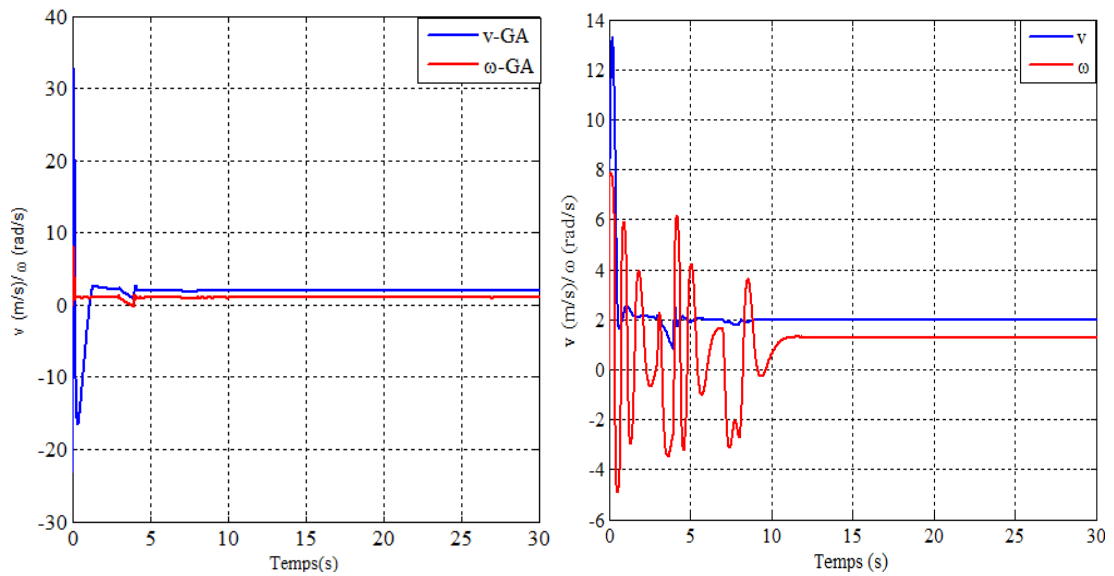


Figure 4.19: Vitesses de commande  $v$  et  $\omega$  obtenues par contrôle dynamique avec et sans AG

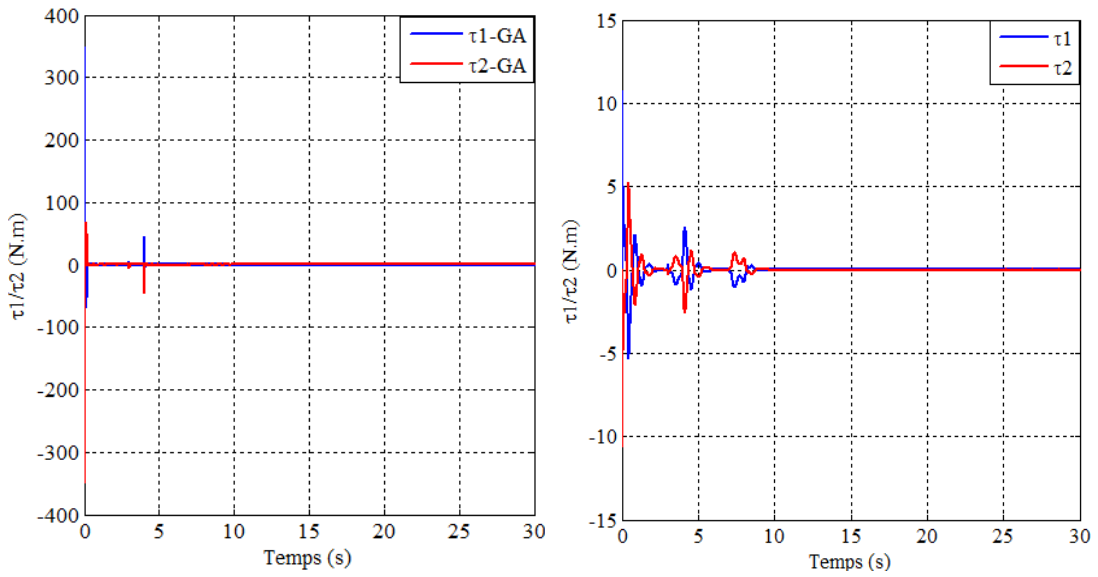


Figure 4.20: Couples  $\tau_1$  et  $\tau_2$  avec et sans AG

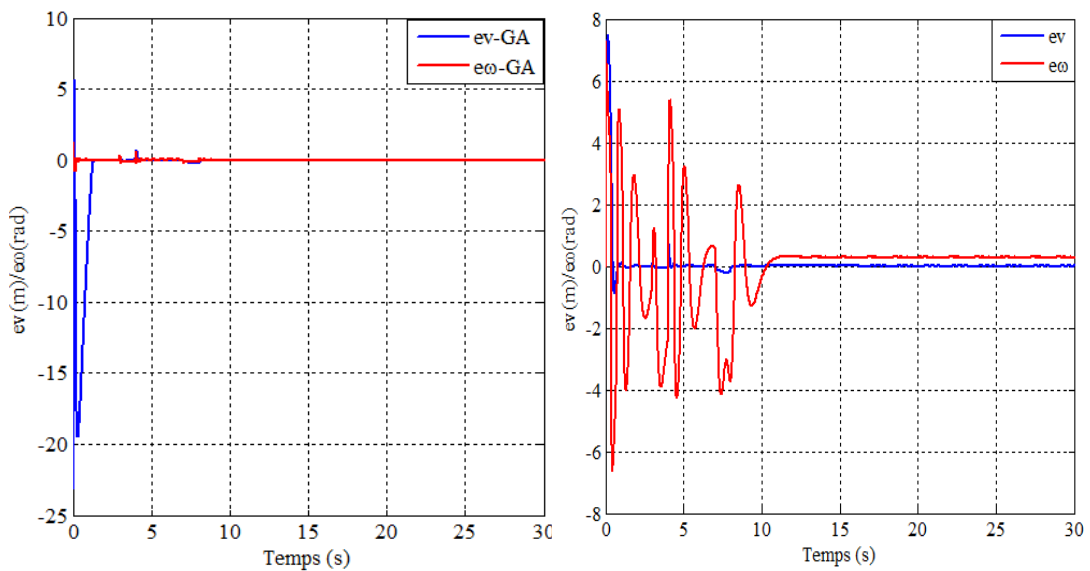


Figure 4.21: Erreurs de vitesses pour les deux cas avec et sans AG

On peut voir sur la figure (4.17) que le robot mobile peut atteindre la trajectoire aléatoire par utilisation des deux contrôleurs dynamique avec et sans AG. On voit également que les erreurs de posture du robot convergent vers zéros comme indiqué sur la figure (4.18).

Les erreurs de suivi pour le contrôleur sans AG oscillent autour de zéro après insertion des perturbations cinématiques et convergent vers zéro dans un temps  $t=9s$  après insertion des perturbations dynamiques. Le temps de convergence avec AG est égal à 4s après insertion

des perturbations cinématiques et à 7s après insertion des perturbations dynamiques. Les erreurs de suivi et les erreurs de vitesse du robot convergent asymptotiquement vers zéro.

Les fonctions fitness de trois trajectoires utilisées sont affichées sous dessous :

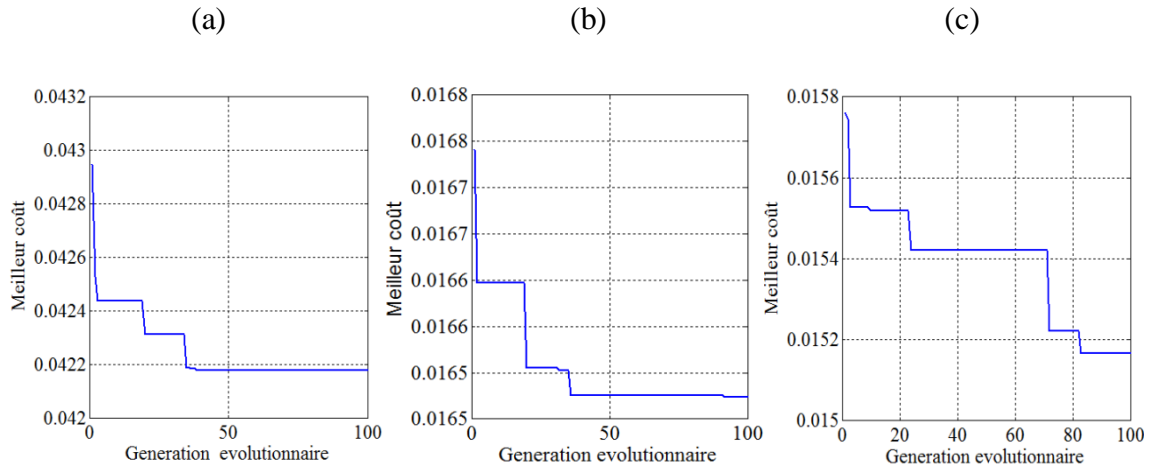


Figure 4.22 : Evolution de la fonction fitness / (a) trajectoire circulaire (b) trajectoire sinusoidale (c) trajectoire aléatoire

Les résultats de la simulation montrent l'efficacité des lois de commande proposées pour le modèle dynamique et cinématique, pour les trois trajectoires de références proposées. Les lois de contrôle proposées assurent d'une manière robuste la convergence des erreurs de suivi vers zéro et garantissent la stabilité asymptotique du système face aux perturbations.

## 4.6 Conclusion

Dans ce travail, des nouvelles lois de contrôle sont proposées pour le suivi de trajectoires des robots mobiles non holonomes. Les lois de contrôle sont divisées en deux parties. Une loi de commande du modèle cinématique est proposée en utilisant la surface de glissement PD et un système flou afin de pallier aux inconvénients des perturbations insérées dans la boucle de commande et assurer la convergence asymptotique des erreurs de suivi vers zéro. La stabilité du système est garantie par une fonction de Lyapunov et le système est asymptotiquement stable. Deuxièmement, une loi de contrôle du modèle dynamique est proposée par utilisation du mode glissant classique et d'une fonction terminale rapide (FTF). Cette loi de contrôle assure d'une manière rapide la convergence des erreurs entre les vitesses dynamiques et les vitesses cinématiques. La stabilité asymptotique est assurée par la proposition d'une fonction candidate de Lyapunov.

Les contrôleurs proposés pour le modèle cinématique et le modèle dynamique ont une capacité de maintenir le robot sur la trajectoire de référence en présence des perturbations qui sont présentes dans les deux modèles considérés. Les algorithmes génétiques sont introduits dans le but d'extraire les paramètres optimaux du contrôleur dynamique. Ces derniers permettent une convergence rapide des erreurs vers zéro. Les travaux de simulation montrent la robustesse des lois de commande proposées face aux perturbations et aux incertitudes présentes dans les deux modèles.



# **CONCLUSION GENERALE**

---

---

## **CONCLUSION GENERALE**

Les travaux présentés dans ce travail de recherche portent sur la localisation et le suivi de trajectoires d'un robot mobile non holonome.

Nous avons commencé notre travail par une description rapide sur la navigation et l'évitement d'obstacles des robots mobiles, ensuite différents types de robots mobiles à roues ont été présentés afin de donner leurs modélisations cinématique et dynamique pour pouvoir élaborer les différents algorithmes de commande.

Ensuite, dans le deuxième chapitre, on a présenté les différents problèmes liés à la localisation sous différentes configurations d'environnements. Des travaux de simulations sont présentés. Dans ce contexte, on a examiné les différents types de localisation utilisés en robotique mobile afin de présenter une méthode de localisation qui se base sur des amers naturels en appliquant le filtre de kalman étendu (FKE). Les travaux de simulation ont été présentés sous différents cas d'environnements avec différents nombres d'amers statiques. Les résultats obtenus montrent clairement que les bruits d'observations et les bruits d'entrées n'influent pas sur l'algorithme. On a également constaté que la performance de l'algorithme de localisation dépend aussi du nombre d'amers utilisés. Cette caractéristique est prouvée par l'utilisation d'environnements structurés par vingt (20), vingt-quatre (24) et trente (30) amers successivement. Suite aux résultats de simulation obtenus, l'algorithme présenté en utilisant FKE donne des réponses adéquates quel que soit le type d'environnements utilisés.

Dans le troisième chapitre, nous avons présenté notre première contribution après une description détaillée du mode glissant terminal (FTSM) et du mode glissant global (GFTSM). La logique floue est utilisée pour l'estimation du gain du contrôleur cinématique. L'optimisation des paramètres du contrôleur dynamique est faite par la méthode d'essaims particuliers (PSO).

- Le premier contrôleur proposé est un contrôleur cinématique basé sur la commande globale (GFTSM) et la logique floue dans le but d'éviter les perturbations de la vitesse linéaire du modèle cinématique du robot et de réduire l'effet de broutement produit ainsi que

la convergence asymptotique des erreurs de posture du robot. Les figures (3.11), (3.16) et (3.21) montrent l'amélioration apportée aux vitesses cinématique du robot.

- Le deuxième contrôleur est un contrôleur dynamique utilisant le mode glissant exponentiel avec optimisation par la méthode d'essaim de particules (PSO) des paramètres imbriqués dans le processus de commande. Cette dernière est implémentée dans le but d'extraire les paramètres optimales de la commande afin d'assurer la convergence asymptotique des erreurs de vitesses du robot, ainsi que les incertitudes présentes dans le modèle dynamique. Les figures (3.12), (3.17) et (3.22) montrent l'amélioration apportée à la convergence des couples impliqués.

Concernant le premier contrôleur, les travaux de simulation illustrent d'une manière détaillée la performance de l'algorithme proposé par utilisation du mode glissant terminal (GFTSM) et la logique floue. Les erreurs de posture du robot convergent asymptotiquement vers zéros dans un temps fini (figure (3.10), figure (3.15) et figure (3.20)). En dépit des perturbations présentes dans les vitesses cinématiques, le robot peut se stabiliser autour de la référence. La stabilité du système est assurée également en appliquant la fonction de Lyapunov. Les paramètres de la commande choisie assurent d'une manière précise la stabilité du système par intégration des règles floues et la dérivée de la fonction candidate de Lyapunov est définie négative.

Pour le deuxième contrôleur, des travaux de simulation sont présentés avec une comparaison entre les deux approches, avec et sans optimisation par essais particuliers des paramètres des contrôleurs dynamiques. Les algorithmes proposés montrent clairement la robustesse vis-à-vis des perturbations et des incertitudes présentes dans le modèle dynamique. Les erreurs entre les vitesses du modèle dynamique et celles obtenues par le modèle cinématique convergent asymptotiquement vers zéros dans un temps très court. De ce fait, le robot mobile suit convenablement les différentes trajectoires de référence considérées, que ce soit une trajectoire circulaire, sinusoïdale ou aléatoire.

Dans la phase finale, nous avons élaboré de nouveaux algorithmes pour le suivi de trajectoire d'un robot mobile en proposant deux contrôleurs.

- Le premier contrôleur est appliqué au modèle cinématique perturbé et est basé sur un PD mode glissant avec logique floue. Cette dernière peut tendre l'erreur de suivi vers zéro en un temps très court. En dépit de la présence des perturbations dans le modèle cinématique, le

contrôleur assure d'une manière exacte la stabilité asymptotique des erreurs de suivi tout en appliquant la fonction candidate de Lyapunov. Cette commande proposée démontre une bonne performance du suivi malgré la présence d'une grande erreur initiale. La posture du robot converge vers la trajectoire souhaitée dans des temps très courts et cette commande nous a permis d'obtenir des résultats satisfaisants en termes de poursuite de trajectoires.

- Le deuxième contrôleur est un contrôleur dynamique basé essentiellement sur le mode glissant terminal (FTF). Ce contrôleur consiste à générer les couples des moteurs droit et gauche du robot afin d'assurer la convergence de l'erreur de vitesse vers zéro en temps fini. En effet, l'implémentation des algorithmes génétiques (GA) permet de ressortir les paramètres optimaux de la surface de glissement terminal (FTF), ce qui nous a permis une convergence rapide de l'erreur de vitesse vers zéro. Des résultats de simulation sont illustrés, ils montrent la robustesse de la loi de contrôle proposée vis à vis des perturbations et des incertitudes du modèle. Les figures (4.7), (4.12) et (4.17) illustrent la performance de l'algorithme proposé, le robot répond en un temps réduit selon différentes trajectoires de référence choisies.

### **Perspectives**

A travers cette étude on a constaté l'importance primordiale de la localisation et le suivi de trajectoires dans le domaine de la robotique mobile. L'ensemble des points examinés dans ce travail de recherche nous a permis de comprendre d'une manière claire les problèmes dont souffre la localisation d'un robot mobile dans une environnement à priori connu ainsi que le suivi de trajectoires. Cette étude nous a permis également de proposer et de décrire quelques perspectives futures. En ce qui concerne la localisation et le suivi de trajectoires, on propose les directives suivantes:

- Développement de l'algorithme de localisation utilisant le filtre de Kalman étendu en tenant compte d'obstacles dynamiques tout en considérant que le robot évolue dans un environnement complètement inconnu.

- Implémentation des techniques de l'intelligence artificielle avec l'algorithme de localisation, par exemple FKE-Flou ou FKE-PSO dans le but d'optimiser les matrices de covariance du filtre de Kalman.

- Appliquer l'approche proposée sur les robots manipulateurs mobiles.

- Validation expérimentale des algorithmes proposés pour la localisation et le suivi de trajectoires.

# **BIBLIOGRAPHIE**



## **BIBLIOGRAPHIE**

- [1] S. Kotani, S. Yasutomi, X. Kin, H. Mori, S. Shigihara, and Y. Matsumuro, “Image processing and motion control of a landmark drawing robot”, In Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems, Yokohama, Japan, 1993, pp. 1035–1041.
- [2] O. Mohareri, R. Dhaouadi, and A.B. Rad, “Indirect Adaptive Tracking Control of a Nonholonomic Mobile Robot via Neural Networks”, *Neurocomput*, vol. 88, pp. 54–66, 2012.
- [3] T. Lozano-Perez, and I. J. Cox, *Autonomous robot vehicles*, Springer Science & Business Media, 2012.
- [4] L. Capisani, A. Ferrara, and A. Pisano, “Second-order sliding mode control with adaptive control authority for the tracking control of robotic manipulators”, In: 18<sup>th</sup> IFAC World Congress, Italy, 2011, pp. 10319-10324.
- [5] Y. Kanayama et al, “A stable tracking control method for an autonomous mobile robot”, in Proc. IEEE Conf. Robot. Autom, pp. 384-389, 1990.
- [6] R. Fierro, and F. L. Lewis, “Control of a Nonholonomic Mobile Robot Using Neural Networks”, in *IEEE Transactions on Neural Networks*, vol. 9, no. 4, pp.589-600, 1998.
- [7] G. Klancar, and I. Skrjanc, “Tracking-error model-based predictive control for mobile robots in real time”, in *Robotics and autonomous systems*, vol. 55, no. 6, pp. 460-469, 2007.
- [8] E.N. Moret, “Dynamic modeling and control of a car-like robot”, Master’s thesis, Virginia Polytechnic Institute and State University, Bradley Department of Electrical and Computer Engineering 2003.

- [9] P. Kachroo, and M. Tomizuka, “Vehicle control for automated highway systems for improved lateral maneuverability”, *In: IEEE International Conference on Systems, Man, and Cybernetics, Vancouver*, vol. 1, pp. 777-782, Canada, 1995.
- [10] N. Ayache, and O. Faugeras, “ Maintaining representations of the environment of a mobile robot”, in *The fourth international symposium*, 1988, pp. 337-350.
- [11] C. Drocourt, “Localisation Et Modélisation De L'environnement D'un Robot Mobile Par Coopération De Deux Capteurs Omnidirectionnels”, Thèse de doctorat, Université de Technologie de Compiègne, 2002.
- [12] X. J. Li, A. So, and S. K. Tso, “CAD-Vision-Range-Based Self-Localization for Mobile Robot Using One landmark”, *Journal of Intelligent and Robotic Systems* vol. 35, no. 1, 2002.
- [13] O. Wijk, and H I. Christensen “Localization and navigation of a mobile robot using natural point landmark extracted from sonar data”, *Robotics and Autonomous Systems*, vol. 31, no. 1, pp. 31-42, 2000.
- [14] R. Madhavan, and H. F. Durrant-Whyte, “Natural landmark-based autonomous vehicle navigation”, *Robotics and Autonomous Systems*, vol. 46, no. 2, pp. 79-95, 2004.
- [15] H. Ahmad, and T. Namerikawa, “Feasibility study of partial observability in H infinity filtering for robot localization and mapping problem”, *American control conference*, Baltimore, USA, 2010, pp. 3980–3985.
- [16] H. Ahmad, and T. Namerikawa, “Robot localization and mapping problem with unknown noise characteristics”, *Multi-conference on systems and control*, pp. 1275-1280, Yokohama, Japan, 2010.
- [17] H. Ahmad, and T. Namerikawa, “Robotic mapping and localization considering unknown noise statistics”, *Journal of System Design and Dynamics*, vol.5, no.1, pp. 70–82., 2011.
- [18] S. Huang, and G. Dissayanake, “Convergence and consistency analysis for extended Kalman filter based SLAM”, *IEEE Transaction on Robotics*, vol. 23, no.5, pp.1036–1049, 2007.
- [19] G. Thomas, and Z. Teresa, “A Simple and Efficient Implementation of EKF-Based SLAM Relying on Laser Scanner in Complex Indoor Environment”, *Journal of Automation, Mobile Robotics and Intelligent Systems*, vol.8, no.2, pp. 58-67, 2014.

- [20] H. Li, M. Tsukada, F. Nashashibi, and M. Parent, “Multi-Vehicle Cooperative Local Mapping: A Methodology Based on Occupancy Grid Map Merging”, *IEEE Transactions on Intelligent Transportation Systems*, vol.15, no.5, pp. 2089-2100, 2014.
- [21] Y. Dobrev, S. Flores, M. Vossiek, “Multi-Modal Sensor Fusion for Indoor Mobile Robot Pose Estimation”, Proceedings of IEEE/ION Position, Location and Navigation Symposium (PLANS), Savannah, GA, USA, pp. 553-556, 2016.
- [22] M. Corradini, and G. Orlando, “Control of mobile robots with uncertainties in the dynamical model: a discrete time sliding mode approach with experimental results”, *Control Engineering Practice*, vol.10, no.1, pp. 23–34, 2002.
- [23] E. Nasim, A. Alireza, and K.Hossein, “Balancing and Trajectory Tracking of Two-Wheeled Mobile Robot Using Backstepping Sliding Mode Control: Design and Experiments”, *Journal of Intelligent and Robotic Systems*, vol. 83, no.4, pp. 601-613, 2017.
- [24] I. Benaoumeur, B. Laredj, R. Amar, E. Hanifi, and A. Zoubir, “Backstepping Approach for Autonomous Mobile Robot Trajectory Tracking”, *Indonesian Journal of Electrical Engineering and Computer Science*, vol. 2, no. 3, pp.478-485, 2016.
- [25] D. Wang, C.B. Low, “Modeling and analysis of skidding and slipping in wheeled mobile robots: control design perspective”, *IEEE Transactions on Robotics*, vol. 24, no.3, pp.676-687, 2008.
- [26] K. Yasmine, B. Mohamed, and D. Tarak, “Robust control of wheeled mobile robot in presence of disturbances and uncertainties”, Sciences and Techniques of Automatic Control and Computer Engineering (STA),Tunisia, 2014.
- [27] W. Haitian, and L. Ge, “Motion Control and Trajectory Tracking Control for a Mobile Robot via Disturbance Observe”, *Wseas Transactions on Systems*, vol.9, no.1, pp. 31-41, 2010.
- [28] M. Ashoorirad, R. Barzamini, A. Afshar, and J. Jouzdani, “Model reference adaptive path following for wheeled mobile robots”, International Conference on Information and Automation, Shandong, China, 2007, pp. 289-294.
- [29] N. Martins, M. Lombardi, W.C. Bertol, and H.F. Filho, “Trajectory tracking of a wheeled mobile robot with uncertainties and disturbances: proposed adaptive neural control”, *Control and Cybernetics*, vol. 44, no.1, pp. 47-98, 2015.



- [30] G. Guoqin, R. Yi, Z. Haiyan, and F. Zhiming, “Smooth sliding mode control for trajectory tracking of greenhouse spraying mobile robot”, *TELKOMNIKA Indonesian Journal of Electrical Engineering*, vol.11, no.2, pp. 642-652, 2013.
- [31] L. Gracia, F. Garell, and A. Sala, “Integrated sliding-mode algorithms in robot tracking applications”, *Robotics and Computer-Integrated Manufacturing*, vol. 29, no.1, pp.53-62, 2013.
- [32] J. Keighobadi, and Y. Mohamadi, “Fuzzy Sliding Mode Control of a Non-Holonomic Wheeled Mobile Robot”, In Proc. International Multiconferenc of Engineers and Computer Scientists, vol. 2, Hong Kong, 2011.
- [33] M. Hong, “Trajectory Tracking Control of Robot via a Fuzzy Sliding-mode Controller”, *Advanced Materials Research*, vol. 433-440, pp.4154-4158, 2012.
- [34] C. Jan, N. Filip, and K. Stefan, “Trajectory tracking with variable sampling using fuzzy logic controller”, IEEE 15th International Carpathian Control Conference, Velke Karlovice, Czech Republic, 2014.
- [35] P. Morin, and C. Samson, “Control of Nonholonomic Mobile Robots Based on the Transverse Function Approach”, In *IEEE Trans. on Robotics*, vol. 25, no.5, pp.1058-1073, 2009.
- [36] Y. Kang, HS. Xi, HB. Ji, and J. Wang, “Fast terminal sliding mode control of uncertain multivariable linear systems”, *J. UnivSciTechnol China*, pp.718–725, 2003.
- [37] SB. Li, KQ. Li, JQ. Wang, and F. Gao, “Nonsingular and fast terminal sliding model control method”, *Inf Control*, vol.38, no.1, pp.1-8, 2009.
- [38] D. Xu, B. Jiang, M. Qian, J. Zhao, “Terminal sliding mode control using adaptive fuzzy-neural observer”, *Math. Problems Eng.*, vol. 2013, Dec. 2013.
- [39] V. Nekoukar, and A. Erfanian, “Adaptive fuzzy terminal sliding mode control for a class of mimo uncertain nonlinear systems”, *Fuzzy Sets and Systems*, vol.179, no.1, pp.34-49, 2011.
- [40] PR. Ouyangn, J. Acob, and V. Pano, “PD with sliding mode control for trajectory tracking of robotic system”, *Robotics and Computer Integrated Manufacturing*, vol.30, no.2, pp. 189–200, 2014.
- [41] Y. Feng, X.Yu, and F. Han, “On nonsingular terminal sliding-mode control of nonlinear systems”, *Automatica*, vol.49, no.6, pp.1715-1722, 2013.

- [42] S. Peng, and W. Shi, “Adaptive fuzzy integral terminal sliding mode control of a nonholonomic wheeled mobile robot”, *Mathematical Problems in Engineering*, vol.2017, pp.1-12, 2017.
- [43] Y. T. Abdalla, AA. Abdulkareem, “PSO-based optimum design of pid controller for mobile robot trajectory tracking department of computer engineering”, *International Journal of Computer Applications*, vol.47,no.23, pp.0975-8887, 2012.
- [44] N. Rickard, “Development of a mobile robot platform”, Master’s Thesis, Lulea, March, 2014.
- [45] A. Kelly, *Mobile Robotics Mathematics, Models, and Methods*, Carnegie Mellon University, 2013.
- [46] S. Hendrik, *Modelling and Controlling of Behaviour for Autonomous Mobile Robots*, Springer Vieweg, 2013.
- [47] Humainoidhistory–wabot.[http://www.humnoid.waseda.ac.jp/booklet/kato\\_2.html](http://www.humnoid.waseda.ac.jp/booklet/kato_2.html).
- [48] (A) Seekur (350 kg, dimensions 1.4 m31.3 m31.1 m), (B) Pioneer 3D-X, and (C)PowerBot.<http://mobilerobots.com/ResearchRobots/ResearchRobots.aspx>;  
[http://www.conscious-robots.com/en/reviews/robots/mobilerobots-pioneer-3p3\\_dx-8.html](http://www.conscious-robots.com/en/reviews/robots/mobilerobots-pioneer-3p3_dx-8.html).
- [49] D. Filliat, *Robotique Mobile*, ENSTA, 2006.
- [50] B. Bayle, *Cours Robotique mobile*, Ecole Nationale Supérieure de Physique de Strasbourg, 2008.
- [51] B. Siciliano, and O.Khatib, *Handbook of Robotics*, Springer-Verlag Berlin Heidelberg, 2008.
- [52] O. Trullier, and J. A. Meyer, “Biomimetic navigation models and strategies in animats”, *AICommunications*, vol.10, no.2, pp.79–92, 1997.
- [53] O. Trullier, S. Wiener, A. Berthoz, and J. A. Meyer, “Biologically-based artificial navigation systems: Review and prospects”, *Progress in Neurobiology*, vol. 51, no.5, pp. 483–544, 1997.
- [54] V. Braitenberg, *Vehicles: Experiments in Synthetic Psychology*, MIT Press, 1986.
- [55] P. Gaussier, C. Joulain, J.P. Banquet, S. Lepretre, and A. Revel, “The visual homing problem: an example of robotics/biology cross-fertilization”, *Robotics and autonomous systems*, vol. 30, no.1-2, pp. 155-180, 2000.
- [56] A. Remazeilles, and F. Chaumette, “Image-based robot navigation from an image memory”, *Robotics and Autonomous Systems*, vol.55, no.4, 2007.

- [57] F. Morbidi, *Localisation et navigation de robot UFR des Sciences*, Département EEAM2 EEAI, 2014/2015.
- [58] D. Anderson, H. Herman, and A. Kelly, “Experimental Characterization of Commercial Flash Ladar Devices”, *Int. Conf. of Sensing and Technology*, Palmerston North, 2005, pp. 17–23.
- [59] R. Chattergy, “Some heuristics for the navigation of a robot”, *Int. J. Robot.Res.* vol.4, no.1, pp. 59–66, 1985.
- [60] J. Minguez, and L. Montano, “Nearness niagram (ND) navigation: collision avoidance in troublesome scenarios”, *IEEE Trans. Robot. Autom.*, vol. 20, no.1, pp. 45–59, 2004.
- [61] D. Fox, W. Burgard, and S. Thrun, “The dynamic window approach to collision avoidance”, *IEEE Robotics and Automation Magazine*, vol. 4, no.1, pp. 23–33, 1997.
- [62] O. Brock, and O. Khatib, “High-speed navigation using the global dynamic window Approach”, in *Proceeding of the IEEE International Conference on Robotics and Automation*, Detroit, 1999, pp. 341-346.
- [63] C.S. Tzafestas, S.G. Tzafestas, “Recent algorithms for fuzzy and neurofuzzy path planning and navigation of autonomous mobile robots”, *Systems-Science*, vol. 25, no.2, pp. 25-39, 1999.
- [64] C.T. Chen, and R.D. Quinn, “A crash avoidance system based upon the cockroach escape response circuit”, in *Proceedings of the IEEE International Conference on Robotics and Automation*, Albuquerque, NM, April 1997.
- [65] J. Vanualailai, S. Nakagiri, and J-H. Ha, “Collision avoidance in a two-point system via Liapunov’s second method”, *Mathematics and Simulation*, vol. 39, no.1-2, pp. 125-141, 1995.
- [66] E. Guechi, “Suivi de trajectoires d’un robot mobile non holonome : approche par modèle flou de Takagi-Sugeno et prise en compte des retards”, thèse de doctorat, 2010.
- [67] S.Tzafestas, *Introduction to Mobile Robot Control*, School of Electrical and Computer Engineering, 2014.
- [68] Kinematic structure of a car-like robot, (B) A car-like robot prototype. <http://sqrt-1.dk/robot/robot.php> .
- [69] J.P. Laumond, “Feedback control of a nonholonomic car-like robot”, *In Robot Motion Planning and Control*, vol. 229, pp. 171-253, 1998.
- [70] G. Dudek, M. Jenkin, “Computational principles of mobile robotics”, Cambridge University Press, 2010.

- [71] M. De Villiers, and N.S. Tlale, “Development of a control model for a four-wheel mecanum vehicle”, *Journal of Dynamic Systems Measurement and Control Transactions of the ASME*, vol. 134, no. 1, pp. 1-6, 2012.
- [72] JB. Song, and KS. Byun, “Design and control of a four-wheeled omnidirectional mobile robot with steerable omnidirectional wheels”, *J Rob System*, vol.21, no.1, pp. 193-208, 2004.
- [73] R. Fierro, and F. L. Lewis, “Control of a nonholonomic mobile robot using neural networks”, *Neural Networks, IEEE Transactions on*, vol.9, no.4, pp. 589-600, 1998.
- [74 ] A. Bloch, P. Crouch, J. Baillieul, and J. Marsden, *Nonholonomic Mechanics and Control*. University of Michigan–Ann Arbor, United States, Interdisciplinary Applied Mathematics, 2003.
- [75] E. J.Haug, “Computer Aided Kinematics and Dynamics of Mechanical Systems ”, *Allyn and Bacon*, vol.1, 1989.
- [76] N. Chakraborty, and A. Ghosal, “Dynamic modeling and simulation of a wheel mobile robot for traversing uneven terrain without slip”, *Journal of Mechanical Design*, vol. 127, no.5, pp. 901-909, 2005.
- [77] Y. Koubaa, M. Boukattaya, and T. Dammak, “Adaptive Sliding-Mode Dynamic Control for Path Tracking of Nonholonomic Wheeled Mobile Robot”, *J. Automation & Systems Engineering*, vol.9, no.2, pp.119-131, 2015.
- [78] Y.C. Chou, and L. Yao, “Adaptive Fuzzy Sliding-Mode Controller with Gaussian Modulated Membership Function on Tracking Trajectory”, *IEEE/ASME International Conference on Advanced Intelligent Mechatronics*, Kaohsiung, Taiwan, 2012, pp.11-14.
- [79] R. Siegwart, I.Reza, and D. Scaramuzza, *Mobile Robots Introduction to Autonomous Mobile Robots*, second edition, Massachusetts Institute of Technology, 2011.
- [80] F.MORBIDI, *Localization et navigation de robots*, UPJV, Département EEA M2 EEAIL, Laboratoire MIS Équipe Perception Robotique, 2017.
- [81] H. Sert, W. Perruquetti, A. Kokosy, X. Jin, and J. Palos, “Localizability of unicycle mobiles robots: an algebraic point of view”, *International Conference on Intelligent Robots and Systems*, Vilamoura, Algrave, Portugal. 2012.
- [82] M. Mirkhani, R. Forsati, A.M. Shahri, and A. Moayedikia, “A novel efficient algorithm for mobile robot localization”, *Robotics and Autonomous Systems*, vol. 61, no.9, pp.920-993, 2013.

- [83] M. Adams, S. Zhang, and L. Xie, "Particle filter based outdoor robot localization using natural features extracted from laser scanners", In: Proc. of the IEEE Int. Conf. Robotics and Automation, USA, 2004.
- [84] M. Agrawal, and K. Konolige, "Real-time Localization in Outdoor Environments using Stereo Vision and Inexpensive GPS", 18th International Conference on Pattern Recognition (ICPR'06), Hong Kong, Chinapp, 2006, pp. 1063-1068.
- [85] Y. Zhou, W.Liu, and P. Huang, "Laser-activated RFID-based Indoor Localization System for Mobile Robots", In IEEE International Conference on Robotics and Automation, Roma, Italy, 2007, pp. 4600-4605.
- [86] UR. Zimmer, "Robust world-modelling and navigation in a real world", *Neurocomputing*, vol. 13, no.2-4, pp. 247- 260, 1996.
- [87] S. Thrun, "Learning metric-topological maps for indoor mobile robot navigation", *Artificial Intelligence*, vol. 99, no.1, pp. 21–71, 1999.
- [88] H. Moravec, "Sensor fusion in certainty grids for mobile robots", *AI Magsin*, vol. 9, no.2, pp. 61-74, 1988.
- [89] K. S. Chong, and L. Kleeman, "Large scale sonarray mapping using multiple connected local maps", In International Conference on Field and Service Robotics, ANU, Canberra, Australia, 1997, pp. 538-545.
- [90] R. Guyonneau, "Méthodes ensemblistes pour la localisation en robotique mobile", Thèse doctorat, Automatique et Robotique, Université d'Angers, 2013.
- [91] J. Borenstein, and L. Feng, "Measurement and correction of systematic odometry errors in mobile robots", *IEEE Transactions on Robotics and Automation*, vol. 12, no.6, pp.869-880, 1996.
- [92] J. Borenstein , " Control and kinematic design of multi-degree-offreedom robots with compliant linkage", *IEEE Transactions on Robotics and Automation*, vol. 11, no.1, pp. 21–35, 1995.
- [93] G. Jang, S. Lee and I.Kweon, "Color landmark based self localization for indoor mobile robots", In IEEE International Conference on Robotics and Automation, pp.1037-1042, 2002.
- [94] N. Ouerhani, and H. Hugli, "Robot self-localization using visual attention", International Symposium on Computational Intelligence in Robotics and Automation, Espoo, Finland, 2005, pp. 309-314.

- [95] J. Borenstein, H.R. Everett, and L. Feng, *Navigating mobile robots: sensors and techniques*. Wellesley, MA: A.K. Peters Ltd, 1999.
- [96] L. Kleeman, “Advanced sonar and odometry error modeling for simultaneous localization and map building”, In IEEE/RSJ international conference on intelligent robots and systems, 2004, pp.1866-1871.
- [97] N. Slimane, “Systeme de localisation pour robots mobiles”, Thèse doctorat, Université de batna, 2005.
- [98] M. Betke, and L. Gurvis, “Mobile robot localization using landmarks”, *IEEE Trans Rob Autom*, vol.13, no.2, pp.251-263, 1997.
- [99] L. DAlfonso, W. Lucia, P. Muraca, and P. Pugliese, “Mobile robot localization via EKF and UKF: A comparison based on real data”, *Robotics and Autonomous Systems*, vol. 74, pp. 122-127, 2015.
- [100] S.J. Julier, and J.K. Uhlmann, “Unscented filtering and nonlinear estimation”, *Proceedings of the IEEE*, vol.92, no.3, pp. 401-422.
- [101] S. Thrun, and D. Fox, W. Burgard, and F. Dellaert, “Robust Monte Carlo localization formobile robots”, *Artificial Intelligence*, vol. 128, no.1-2, pp. 99–141, 2001.
- [102] D. Fox, *KLD-sampling: Adaptive particle filters and mobile robot localization*. Advances in Neural Information Processing Systems, MIT Press, 2001.
- [103] Y. Song and J. W. Grizzle, “The extended Kalman filter as a local asymptotic observer for nonlinear discrete-time systems”, In American Control Conference, IEEE, 1992, pp.3365–3369.
- [104] M.S. Grewal and A.P Andrews, *Kalman filtering\_ Theory and practice using MATLAB*, John Wiley & Sons, Inc, 2008.
- [105] R. G. Brown, Y. Patrick, and C. Hwang, *Introduction to Random Signals and Applied Kalman Filtering with Matlab Exercises*, Wiley, 2012.
- [106] F.Yang, Z. Wang, S.Lauria, and X. Liu, “Mobile robot localization using robust extended  $H_\infty$  filtering”, *Journal of Systems and Control Engineering*, vol. 223, no.8, pp. 1067-1080, 2009.
- [107] H. Ahmada and T.Namerikawab, “Extended Kalman filter-based mobile robot localization with intermittent measurements”, *System Science & Control Engeneering* , vol. 1, no.1, pp.113-126, 2013.

- [108] J.J. Leonard, and H.F. Durrant-Whyte, “Mobile robot localization by tracking geometric beacons”, *IEEE Transactions on Robotics and Automation*, vol.7, no. 1, pp.376–382, 1991
- [109] M. Khatib, H. Jaouni, R. Chatila and J. P. Laumond, “Dynamic path modification for car-like nonholonomic mobile robots”, In: Proceedings of International Conference on Robotics and Automation. Albuquerque, IEEE, 1997, pp. 2920-2925.
- [110] G. Ramirez, and S. Zeghloul, “A New Local Path Planner for Nonholonomic Mobile Robot Navigation in Cluttered Environments”, In: IEEE International Conference on Robotics and Automation, San Francisco, 2000, pp. 2058-2063.
- [111] S. Razvan, and U. Nunes, “Trajectory Planning and Sliding Mode Control Based Trajectory Tracking for Cybercars”, *Integrated Computer-Aided Engineering*, vol. 14, no. 1, pp. 33-47, 2007.
- [112] H. Khenouf, and C. C. De wit, “Quasi- Continuous Exponential Stabilizers for Nonholonomic Systems”, *IFAC Proceedings*, vol. 29, no. 1, pp. 2448-2453, 1996.
- [113] F. Dinuzzo, and A. Ferrara, “Higher order sliding mode controllers with optimal reaching”, *IEEE Transactions on Automatic Control*, vol. 54, no.9, pp. 2126-2136, 2009.
- [114] J. Keighobadi, and Y. Mohamadi, “Fuzzy Sliding Mode Control of non-holonomic Wheeled Mobile Robot”, In: IEEE 9th International Symposium on Applied Machine Intelligence and Informatics, Smolenice, Slovakia, 2011 , pp. 273-278.
- [115] J. M. Yang, and J. H. Kim, “ Sliding mode control for trajectory tracking of nonholonomic wheeled mobile robots”, *Transactions on Robotics and Automation*, vol. 15, no. 3, pp. 578-587, 1999.
- [116] N. M. Dung, V. H. Duy, N. T. Phuong, S. B. Kim, and M. S. Oh, “Two Wheeled Welding Mobile Robot for Tracking a Smooth Curved Welding Path Using Adaptive Sliding-Mode Control Technique”, *International Journal of Control Automation and Systems*, vol. 5, no. 3, pp. 283-294, 2007.
- [117] V.I. Utkin, “Variable structure systems with sliding modes”, *IEEE Transactions on Automatic Control*, vol. 22, no. 2, pp. 212-222, 1977.
- [118] Y. Orlov, *Discontinuous systems: Lyapunov analysis and robust synthesis under uncertainty conditions*, London: Springer-Verlag, 2010.
- [119] C. S. T. Dong, H. H. Vo, T. C. Tran, P. Brandstetter, and P. Simonik, “Application of Sensorless Sliding Mode Observer in Control of Induction Motor Drive”, *Advances in Electrical and Electronic Engineering*, vol. 15, no. 5, pp. 747-753, 2018.

- [120] M. Park, D. Chwa, and S. Hong, “Decoupling Control of A Class of Underactuated Mechanical Systems Based on Sliding Mode Control”, In: SICE-ICASE International Joint Conference, Busan, 2006, pp. 806-810.
- [121] K. N. Faress, M. T. El hargy, and A. A. El kosy, “ Trajectory tracking control for a wheeled mobile robot using fuzzy logic controller”, In Proceedings of the 9th WSEAS International Conference on Systems, Athens, Greece, 2005, pp. 1261-1269.
- [122] S.L Cárdenas, O.Castillo, L.T Aguilar, and N. , Cázarez, (2007) “Tracking Control for a Unicycle Mobile Robot Using a Fuzzy Logic Controller”, *Hybrid Intelligent Systems, Studies in Fuzziness and Soft Computing*, vol.208, pp.243-253, 2007.
- [123] C. Y. Chen, T. S. Li, and Y. Yeh, “EP based kinematic control and adaptive fuzzy sliding-mode dynamic control for wheeled mobile robots”, *Information Sciences*, vol. 179, no. 2, pp. 180-195, 2009.
- [124] M. Hong, “Trajectory Tracking Control of Robot via a Fuzzy Sliding-mode Controller”, *Advanced Materials Research*, vol. 433, no. 1, pp. 4154-4158, 2012.
- [125] S. M. Kazraji, and M. B. B. Sharifian, “Direct Thrust Force and Flux Control of a PMLinearSynchronous Motor Using Fuzzy Sliding- Mode Observer”, *Advances in Electrical and ElectronicEngineering*, vol. 13, no. 1, pp. 1-9, 2015.
- [126] D. Chwa, “Fuzzy adaptive tracking control of wheeled mobile robots with state-dependent kinematic and dynamic disturbances”, *IEEE Trans. Fuzzy Syst.*, vol. 20, no. 3, pp. 587-593, Jun. 2012..
- [127] J. X. Shao, Y. D. Zhao, D. E. Kim, and J. M. Lee, “Trajectory Tracking by Terminal Sliding Mode Control for a Three-Wheeled Mobile Robot”, In: International Conference on Intelligent Robotics and Applications. Cham: Springer, 2017, pp. 215-225.
- [128] J. Liu, and X. Wang, *Advanced sliding mode control for mechanical systems: design, analysis and MATLAB simulation*, Springer, Beijing: Tsinghua University Press, 2012.
- [129] C. Coza, and C.J.B. Macnab, “A new robust adaptive-fuzzy control method applied to quadrotorhelicopter stabilization”, In Fuzzy Information Processing Society, Annual meeting of the North American, 2006, pp. 454-458.
- [130] P. Ponce-Cruz, A. Molina, and M.Brian, *Fuzzy Logic Type 1 and Type 2 Based on LabVIEW™ FPGAPublisher*, Springer International Publishing, 2016.
- [131] N. A. Martins, M. Alencar, W. C. Lombardi, D. W. Bertol, E. R. Pieri, and H. F. Filho, “Trajectory tracking of a wheeled mobile robot with uncertainties and disturbances:



- proposed adaptive neural control”, *Control and Cybernetics*, vol. 44, no.1, pp. 47-98, 2015.
- [132] J. H., Lee, C. Lin, H. Lim, and J. M. Lee, “Sliding mode control for trajectory tracking of mobile robot in the RFID sensor space”, *International Journal of Control, Automation and Systems*, vol. 7, no. 3, pp. 429-435, 2009.
- [133] Y. Koubaa, M. Boukattaya, and T.dammak, “Adaptive Sliding-Mode Dynamic Control For Path Tracking of Nonholonomic Wheeled Mobile Robot”, *Journal of Automation and Systems Engineering*, vol. 9, no.2, pp. 119–131, 2015.
- [134] L. X. Wang, “Stable adaptive fuzzy control of nonlinear systems”, *Transactions on Fuzzy Systems*, vol. 1, no. 2, pp. 146-155, 1993.
- [135] N. Martins, D. Bertol, W. Lombardi, R. Pieri, and B. Castelan, “Trajectory tracking of a nonholonomic mobile robot with parametric and nonparametric uncertainties: A proposed neural control”, In: 16thMediterranean Conference on Control and Automation, Ajaccio, 2008, pp. 315-320.
- [136] ] O. Castillo, L. T. Aguilar, and S. Cardenas, “Fuzzy logic tracking control for unicycle mobile robots”, *Engineering Letters*, vol. 13, no. 1, pp. 73-77, 2006.
- [137] A. Mohagheghi, F. Shabaninia, and M. Salimifard, “Fuzzy Logic and Fuzzy Sliding Mode Tracking Control of Non-holonomic Unicycle Wheeled Mobile Robots”, In: 21st Iranian Conference on Electrical Engineering, Mashhad, pp.1-6, 2013.
- [138] N. Bouarroudj, D. Boukhetala, and F. Boudjema, “Tuning fuzzy PD $\alpha$  sliding-mode controller using pso algorithm for trajectory tracking of a chaotic system”, *Journal of Electrical Engineering*, vol.14, 2014.
- [139]V. Truong Vu , “A comparison of particle swarm optimization and differential evolution”, *International Journal on Soft Computing (IJSC)* ,vol.3, no.3, 2012.
- [140] M. Clerc, “The swarm and the queen: towards adeterministic and adaptive particle swarm optimization”, In: Proceedings of the ICEC, Washington, pp. 1951-1957, 1999.
- [141] C. Wei-Der, and S. Shun-Peng, “PID controller design of nonlinear systems using an improved particle swarm optimization approach”, *Commun Nonlinear SciNumerSimulat*, pp. 3632-3639, 2010.
- [142] S. E. Li, and K. Deng, *Recent Advances in Non-singular Terminal Sliding Mode Control Method*, Springer-Verlag Berlin Heidelberg, 2014.

- [143] K.D. Young, V.I. Utkin, and U. Ozguner, “A control engineer’s guide to sliding mode control”, In: Proceedings of IEEE international workshop on variable structure systems, pp.11–14, 1996.
- [144] V.I. Utkin, “Sliding mode control design principles and applications to electric drives”, *IEEE Transactions on Industrial Electronics*, vol.40, no.1, pp. 23-36, 1993.
- [145] H. Du, and S. Li, “Finite-time cooperative attitude control of multiple spacecraft using terminal sliding mode control technique”, *Int. J. Model Ident Control*, vol. 16, no.4, pp. 327-333, 2012.
- [146] Y. Feng, X. Yu, and Z. Man, “Non-singular terminal sliding mode control and its application for robot manipulators”, In: IEEE international symposium on circuits and systems, Sydney, Australia, pp. 545-548, 2001.
- [147] X. Yu, Z. Man, and Y. Wu, “Terminal sliding modes with fast transient performance”, In: Proceedings of the 36th IEEE conference on decision and control, San Diego, CA, USA, pp 962-963, 1997.
- [148] S. Yu, G. Guo, Z. Man, and J. Du, “Global fast terminal sliding mode control for robotic manipulators”, *Int J Model Ident Control*, vol. 1, no.1, pp. 72-79, 2006.
- [149] O. Kramer, *Genetic Algorithm Essentials Department for Computing Science, Computational Intelligence Group University of Oldenburg Oldenburg, Germany*, 2017.
- [150] S.N.Sivanandam, and S.N.Deepa, *Introduction to Genetic Algorithms*, Springer Berlin Heidelberg, New York, 2008.
- [151] D.R. Farzdaq, R. Yasien, and Q. Khadim, “Chattering Attenuation of Sliding Mode Controller Using Genetic Algorithm and Fuzzy Logic Techniques”, *Eng. And Tech. Journal*, vol. 27, no.14, pp. 2595-2610, 2009.
- [152] S.S. Rao, *Optimization Theory and Practice*, 4<sup>th</sup> Edition, John Wiley and Sons Inc, 2009.
- [153] M. Rahul, S. Narinder, and S. Yaduvir, “Genetic Algorithms: Concepts, Design for Optimization of Process Controller”, *Computer and information science*, vol. 4, no.2, pp. 39-54, 2011.

## **LISTE DES TRAVAUX**

### **Revue Internationale**

- W.Benaziza, N.Slimane, A.Mallem, “Disturbances Elimination with Fuzzy Sliding Mode Control for Mobile Robot Trajectory Tracking”, Advances in electrical and electronic engineering, vol 16 N° 3, 2018, pp 297-310.
- W.Benaziza, N.Slimane, A.Mallem, “PD Terminal Sliding Mode Control Using Fuzzy Genetic Algorithm for Mobile Robot in Presence of Disturbances”, journal of automation, Mobile robotics & intelligent systems, vol 12 N°2, 2018, pp 52-60.
- A.Mallem, N.Slimane, W.Benaziza,” Robust control of mobile robot in presence of disturbances using neural network and global fast sliding mode “; Journal of Intelligent And Fuzzy systems, vol 34 2018,pp 4345-4354.

### **Conférences Internationales**

- W.Benaziza, N.Slimane, A.Mallem, “Mobile robot trajectory based on fast terminal sliding mode control”, International conference on Automatic Control, Telecommunication and Signals (ICATS’2015), Annaba, Algeria.
- W.Benaziza, N.Slimane, A.Mallem, “Mobile robot tracking using terminal sliding mode control”, 6<sup>th</sup> International conference on Systems and Control, (ICSC’2017), Batna, Algeria.
- A.Mallem, N.Slimane, W.Benaziza, “Mobile robot trajectory tracking using PID fast terminal sliding mod inverse dynamic control”, 4th international conference on control Engineering and information technology (CEIT-2016), Hammamet, Tunisia.