

وزارة التعليم العالي و البحث العلمي
MINISTÈRE DE L'ENSEIGNEMENT SUPÉRIEUR ET DE LA RECHERCHE
SCIENTIFIQUE

جامعة سطيف 1
UNIVERSITÉ SÉTIF 1

THÈSE

Présentée au département d'Electrotechnique
Faculté de Technologie

Pour obtenir le diplôme de

Doctorat Sciences en Automatique

Par

AKKA Khaled

Thème

***Contribution à l'optimisation de trajectoires
des robots mobiles***

Soutenu devant la commission d'examen composée de :

<i>RAHMAN Lazhar</i>	<i>Prof. à l'université Sétif 1</i>	<i>Président</i>
<i>KHABER Farid</i>	<i>Prof. à l'université Sétif 1</i>	<i>Rapporteur</i>
<i>MOSTEFAI Mohammed</i>	<i>Prof. à l'université Sétif 1</i>	<i>Examineur</i>
<i>BAHI Tahar</i>	<i>Prof. à l'université d'Annaba</i>	<i>Examineur</i>
<i>LADACI Samir</i>	<i>Prof. à l'Ecole Polytechnique de Constantine</i>	<i>Examineur</i>
<i>MOUSSAOUI Abdelkrim</i>	<i>Prof. à l'université de Guelma</i>	<i>Examineur</i>

REMERCIEMENTS

Le travail que nous présentons dans le cadre de cette thèse de Doctorat a été réalisé dans le laboratoire de la Qualité d'Energie dans les Réseaux Electriques (QUERE) sous la direction du Professeur **Farid Khaber**.

Avant tous, Je remercie **ALLAH** le tout puissant de m'avoir donné la volonté, le courage et la patience lors de toutes ces années d'étude.

Je tiens à remercier vivement, Monsieur **Farid Khaber**, Professeur à l'université Ferhat ABBAS de Sétif pour la proposition du sujet de cette thèse, pour son soutien scientifique et humain et pour ses conseils et ses critiques objectifs tout au long de ce travail. Merci Monsieur pour la confiance que vous m'avez accordée au cours de ces années d'encadrement.

J'adresse mes sincères remerciements à Monsieur **Lazhar RAHMANI**, Professeur à l'université Ferhat ABBAS de Sétif pour avoir accepté de juger ce travail et de présider le jury.

Je remercie également Messieurs **Mohammed MOSTEFAI**, Professeur à l'université Ferhat ABBAS de Sétif, **Tahar BAHI**, Professeur à l'université de Annaba, **Samir LADACI**, Professeur à l'école nationale polytechnique de Constantine et **Abdelkrim MOUSSAOUI**, Professeur à l'université de Guelma, pour l'intérêt qu'ils ont bien voulu porter à ce travail, en acceptant de l'examiner.

Enfin, je n'oublie pas d'adresser mes vifs remerciements à tous les membres de ma famille pour leurs encouragements, leur soutien moral et la patience qu'ils m'ont manifestés durant mes études.

DÉDICACES

Je dédie cette thèse

à mes parents,

à mes frères et mes sœurs,

et à tous ceux qui m'aiment

ABRÉVIATIONS

dire: direction

dist: distance

vite: vitesse

SCF1: Sous Contrôleur Flou 1

SCF2: Sous Contrôleur Flou 2

TS: Takagi Sugeno

PDC: Compensation Parallèle Distribuée (Parallel Distributed Compensation)

LQ: Linéaire Quadratique (Linear Quadratic)

LQR: Régulateur Linéaire Quadratique (Linear Quadratic Regulator)

sp: probabilité stimulante (stimulating probability)

NOTATIONS

v : Vitesse linéaire

w : Vitesse angulaire

v_p : Vitesse linéaire de poursuite

w_p : Vitesse angulaire de poursuite

v_{ob} : Vitesse linéaire d'évitement d'obstacles

w_{ob} : Vitesse angulaire d'évitement d'obstacles

v_r : Vitesse linéaire de référence

w_r : Vitesse angulaire de référence

\dot{v} : Accélération

V_a : Vitesses admissibles

V_d : Fenêtre dynamique

V_s : Vitesses possibles

V_r : Vitesses résultantes

E_p : Erreur de position

E_o : Erreur d'orientation

Pv : Produit vectoriel

D_v : Distance verticale

(x_d, y_d) : Coordonnées instantanées de la trajectoire désirée

XOY: Repère orthogonal homogène

P_{ij}^k : Règle de déplacement des fourmis

η_{ij} : Visibilité de la grille

τ_{ij} : Phéromone

ρ : Taux d'évaporation de phéromone

LISTE DES TABLEAUX

Chapitre I Commande floue pour le suivi de trajectoires planifiées par la fenêtre dynamique floue

I.1: Les règles floues pour déterminer les facteurs de pondération	27
I.2: Les règles floues de la vitesse linéaire et angulaire	31
I.3: Les règles floues de la vitesse angulaire	33
I.4: Les règles floues de la variation dans la vitesse linéaire	33

Chapitre II Poursuite de trajectoires et évitement d'obstacles par une commande optimale floue

II.1: Les règles floues pour définir les valeurs de Q et R	59
II.2: Les règles floues pour déterminer les valeurs du gain F	60

Chapitre III Planification de trajectoires par un algorithme amélioré de colonies de fourmis

III.1. Résultats statistiques des algorithmes dans les deux environnements	80
--	----

LISTE DES FIGURES

Chapitre I Commande floue pour le suivi de trajectoires planifiées par la fenêtre dynamique floue

I.1. Représentation de la roue	07
I.2. Robot mobile unicycle	08
I.3. Structure d'un contrôleur flou	15
I.4. Fonctions d'appartenance symétriques et équidistantes	17
I.5. Chevauchement insuffisant des fonctions d'appartenance	17
I.6. Chevauchement excessif des fonctions d'appartenance	17
I.7. Défuzzification par centre de gravité	19
I.8. Défuzzification par valeur maximale	20
I.9. Fenêtre de sélection des vitesses convenables	23
I.10. Robot incapable d'éviter le premier obstacle	24
I.11. Robot incapable de passer à travers les espaces étroits	25
I.12. Robot incapable d'atteindre la cible	25
I.13. Robot incapable d'emprunter la trajectoire minimale	26
I.14. Fonctions d'appartenance, (a) des entrées et (b) des sorties	27
I.15. Organigramme de la fenêtre dynamique floue	28
I.16. Suivi de trajectoire basé sur l'erreur de position et d'orientation	29
I.17. Fonctions d'appartenance des variables, (a) d'entrée et (b) de sortie	31
I.18. Amélioration de l'orientation	32
I.19. Fonctions d'appartenance des variables, (a) d'entrée et (b) de sortie	33
I.20. Contrôleur flou global de poursuite	34
I.21. Diagramme bloc de la stratégie de contrôle proposée	35
I.22. (a) Trajectoire optimale issue de la fenêtre dynamique floue, (b), (c) et (d) facteurs de pondération correspondants	37
I.23. (a) Trajectoire optimale issue de la fenêtre dynamique floue dans un environnement complexe, (b), (c) et (d) facteurs de pondération correspondants	38
I.24. Planification de trajectoire par la fenêtre dynamique floue,	

(a) à partir de plusieurs positions et (b) vers différentes positions de la cible	38
I.25. (a) Poursuite de trajectoire, (b) erreurs de poursuite, (c) vitesse linéaire et (d) vitesse angulaire	39
I.26. (a) Poursuite de trajectoire, (b) erreurs de poursuite, (c) vitesse linéaire et (d) vitesse angulaire	40
I.27. (a) Poursuite de trajectoire, (b) erreurs de poursuite, (c) vitesse linéaire et (d) vitesse angulaire	41
I.28. (a) Poursuite de trajectoire, (b) erreurs de poursuite, (c) vitesse linéaire et (d) vitesse angulaire	41
I.29. (a) Poursuite de trajectoire, (b) erreurs de poursuite, (c) vitesse linéaire et (d) vitesse angulaire	42
I.30. Poursuite des trajectoires lemniscates	42
I.31. Poursuite de trajectoire: (a) résultats obtenus, (b) résultats de [Kla-07]	43
I.32. Signaux de commande: (a) résultats obtenus, (b) résultats de [Kla-07]	43

Chapitre II Poursuite de trajectoires et évitement d'obstacles par une commande optimale floue

II.1. Représentation du concept de l'approche PDC	51
II.2. Modèle du robot mobile	55
II.3. Erreur de posture	55
II.4. Fonctions d'appartenance des entrées	58
II.5. Fonctions d'appartenance des sorties	59
II.6. Fonctions d'appartenance des variables d'entrée	60
II.7. Fonctions d'appartenance de la variable de sortie	60
II.8. Diagramme bloc de l'approche de contrôle proposée	61
II.9. Fonctions d'appartenance des entrées	62
II.10. (a) Poursuite de trajectoire et (b) erreurs de poursuite	62
II.11. Poursuite de trajectoire: (a) résultat obtenu, (b) résultat de [Kim-09], (c) résultat de [Wha-10]	63
II.12. Poursuite de trajectoire et évitement d'obstacles	64
II.13. Erreurs de poursuite de trajectoire	64
II.14. Signaux de commande: (a) vitesse linéaire et (b) vitesse angulaire	65

TABLE DES MATIÈRES

TABLE DES MATIÈRES

Introduction générale	01
Chapitre I	
Commande floue pour le suivi de trajectoires planifiées par la fenêtre dynamique floue	04
I.1. Introduction	04
I.2. Modélisation et caractéristiques structurelles de l'unicycle	06
I.2.1. Hypothèses	06
I.2.2. Roulement sans glissement	06
I.2.3. Commandabilité	08
I.2.4. Modèle cinématique du robot de type unicycle	08
I.3. Concepts fondamentaux de la logique floue	10
I.3.1. Ensemble flou	10
I.3.2. Propriétés d'un ensemble flou	11
I.3.3. Variables linguistiques	11
I.3.4. Fonctions d'appartenance	12
I.3.5. Opérateurs flous	13
I.3.6. Règles floues	13
I.4. Commande floue	15
I.4.1. Configuration de base d'un contrôleur flou	15
I.4.1.1. Fuzzification	15
I.4.1.1.1. Fuzzification par singleton	16
I.4.1.1.2. Fuzzification non singleton	16
I.4.1.2. Base de connaissances	16
I.4.1.2.1. Base de données	16
I.4.1.2.2. Base de règles floues	17
I.4.1.3. Mécanisme d'inférence	18
I.4.1.3.1. Méthode d'inférence max-min	18
I.4.1.3.2. Méthode d'inférence max-prod	18
I.4.1.3.3. Méthode d'inférence somme-prod	18

I.4.1.4. Défuzzification	19
I.4.1.4.1. Défuzzification par centre de gravité	19
I.4.1.4.2. Défuzzification par valeur maximale	19
I.4.1.4.3. Défuzzification par la moyenne des maximums	20
I.4.2. Différents types de systèmes flous	20
I.4.2.1. Système flou de Mamdani	20
I.4.2.2. Système flou de Takagi-Sugeno	20
I.5. Planification de trajectoire	21
I.5.1. Principe de l'approche de la fenêtre dynamique	21
I.5.2. Etapes de l'algorithme de la fenêtre dynamique	21
I.5.2.1. Espace de recherche des vitesses	21
I.5.2.1.1. Trajectoires circulaires	22
I.5.2.1.2. Vitesses admissibles	22
I.5.2.1.3. Fenêtre dynamique	22
I.5.2.2. Optimisation du mouvement	23
I.5.3. Inconvénients de la fenêtre dynamique	24
I.5.4. Ajustement des facteurs de pondération en temps réel	26
I.6. Poursuite de trajectoire	28
I.6.1. Stratégie de contrôle	29
I.6.2. Contrôleur flou de position et d'orientation	29
I.6.3. Contrôleur flou d'amélioration de l'orientation	31
I.6.4. Contrôleur flou global de poursuite	34
I.7. Exemples de simulation	35
I.7.1. Résultats de simulation de planification	36
I.7.2. Résultats de simulation de poursuite	38
I.8. Etude comparative	43
I.9. Conclusion	44
Chapitre II Poursuite de trajectoires et évitement d'obstacles par une commande optimale floue	45
II.1. Introduction	45
II.2. Modélisation de Takagi Sugeno	46
II.3. Obtention des modèles flous TS	47
II.3.1. Linéarisation	48

III.5.3. Règle de mise à jour de phéromone	77
III.5.4. Stratégie d'évaporation dynamique	78
III.6. Exemples de simulation	78
III.7. Etude comparative	80
III.8. Conclusion	81
Conclusion générale	83
Bibliographie	86

INTRODUCTION GÉNÉRALE

INTRODUCTION GÉNÉRALE

Récemment, la robotique est devenue un domaine de recherche très intéressant en raison de la révolution qu'elle a provoqué dans le domaine de l'ingénierie à travers les diverses contributions théoriques et pratiques réalisées avec succès. Spécifiquement, le domaine de la robotique mobile a eu la partie majeure de cette évolution, notamment en ce qui concerne la conception, le contrôle, l'application et d'autres aspects qui rendent le robot mobile utile aux activités humaines.

Un robot mobile est un système mécanique capable de se mouvoir dans un environnement pour effectuer des tâches de façon autonome ou supervisée. Les robots mobiles autonomes représentent l'axe principal de la plupart des recherches sur la robotique mobile car ils sont des agents intelligents pouvant opérer dans différents environnements connus ou inconnus et réagir aux changements dynamiques de ces environnements sans intervention humaine. Pour cela, ils doivent être équipés de capteurs afin d'extraire des informations de l'environnement, d'actionneurs qui leur permettent l'exécution des mouvements et de régulateurs ou contrôleurs qui commandent les actionneurs. Les robots mobiles autonomes peuvent être des robots marcheurs, aériens, sous-marins ou à roues. Ces derniers se présentent sous de nombreuses formes et types tels que des robots à deux ou trois roues, des robots de type voiture, des robots omnidirectionnels et des manipulateurs mobiles, et se caractérisent par leur multifonctionnalité qui a attiré l'attention de beaucoup de roboticiens depuis les années 70, et c'est pourquoi leur présence est fortement accrue dans l'industrie manufacturière, militaire, spatiale, la médecine, le transport et les loisirs, soit pour assister l'être humain soit pour le remplacer lorsque il s'agit de l'exécution des tâches difficiles, répétitives ou dangereuses. La performance de ces agents intelligents durant leur fonctionnement dépend de l'efficacité des systèmes de navigation utilisés. Avec un bon système de navigation autonome, le robot mobile va réussir à accomplir son travail en toute sécurité, en consommant moins d'énergie et dans un temps raisonnable.

Malgré la diversité des travaux dans le domaine de la navigation autonome des robots mobiles, il reste toujours des problèmes à résoudre et des applications à découvrir, c'est à dire,

aucun système de navigation autonome ne peut être employé pour résoudre tous les problèmes. Dans ce contexte, le cadre général de cette thèse est la navigation des robots mobiles autonomes dans des environnements statiques. En particulier, les travaux envisagés ont comme objectifs principaux, le développement d'une stratégie de contrôle flou pour le suivi de trajectoires planifiées par la fenêtre dynamique floue, l'établissement d'une structure de commande optimale floue qui permet de réaliser la poursuite de trajectoires en l'absence et en présence d'obstacles sur la trajectoire désirée et l'élaboration d'un algorithme amélioré de colonies de fourmis afin de planifier des trajectoires optimales par rapport à celles obtenues par l'algorithme de base dans des environnements de grilles. Les approches de commande proposées ne prennent en charge que les robots mobiles de type unicycle à travers une commande en vitesse basée sur le modèle cinématique.

L'ensemble des travaux réalisés dans le cadre de cette thèse est structuré en trois chapitres, en plus d'une introduction générale et une conclusion générale.

Le premier chapitre traite le problème de la planification de trajectoires par la fenêtre dynamique et le problème de la poursuite de trajectoires via la commande floue. Nous commençons par un état de l'art non exhaustif du robot mobile unicycle et des notions de base de l'approche floue et de la commande floue ainsi que les outils mathématiques nécessaires à leur manipulation. Ensuite nous exposerons en détails la fenêtre dynamique originale et son amélioration par un contrôleur flou qui permet de modifier dynamiquement les paramètres de la fonction "objectif". Enfin, nous présenterons la synthèse de deux contrôleurs flous fonctionnant alternativement pour assurer la poursuite de trajectoires qui sont soit exprimées par des équations mathématiques soit planifiées à travers la fenêtre dynamique floue.

Dans le deuxième chapitre, nous proposerons une architecture de contrôle qui combine la technique de compensation parallèle distribuée, la commande linéaire quadratique et la commande floue pour résoudre le problème de suivi de trajectoires en l'absence et en présence d'obstacles statiques sur la trajectoire de référence. Nous débuterons par la définition des différents outils indispensables à la mise en place de l'approche développée, puis, nous expliquerons comment concevoir le contrôleur de poursuite basé sur les modèles flous de Takagi-Sugeno à l'aide de la compensation parallèle distribuée et la commande linéaire quadratique. Par la suite, nous montrerons comment faire fonctionner un régulateur linéaire quadratique comme un contrôleur d'évitement d'obstacles en ajustant les valeurs de ses

matrices de pondération par un contrôleur flou. Finalement, nous incorporerons les commandes obtenues en utilisant un gain de fusion floue.

Dans le troisième chapitre, il s'agit de la planification de trajectoires par la mise en œuvre d'un algorithme amélioré de colonies de fourmis dans des environnements modélisés en grilles. Un aperçu de l'algorithme classique de colonies de fourmis sera donné, puis, nous expliciterons les améliorations apportées qui seront réalisées en quatre phases: (i) l'intégration d'une probabilité stimulante dans la règle de déplacement pour que la transition des fourmis soit entre des grilles accessibles et sans collision, (ii) l'usage du principe de longueur de pas libre à la configuration de l'information heuristique pour faire monter la précision de la visibilité ainsi que l'efficacité de la recherche, (iii) l'application d'une nouvelle méthode de mise à jour de phéromone et (iv) l'adoption d'une nouvelle règle d'évaporation de phéromone. Les deux dernières améliorations visent à mettre en valeur la capacité de recherche et à expédier la convergence.

Cette thèse se termine par une conclusion générale qui résume le bilan des travaux réalisés et les perspectives envisagées.

CHAPITRE I

COMMANDE FLOUE POUR LE SUIVI DE TRAJECTOIRES PLANIFIÉES PAR LA FENÊTRE DYNAMIQUE FLOUE

COMMANDE FLOUE POUR LE SUIVI DE TRAJECTOIRES PLANIFIÉES PAR LA FENÊTRE DYNAMIQUE FLOUE

I.1. Introduction

La navigation des robots mobiles consiste à contrôler leurs mouvements du point de départ au point d'arriver selon les caractéristiques de l'environnement. Elle comprend de nombreuses fonctions interconnectées entre elles, par des variables ou des objectifs, comme la planification de trajectoire, le suivi de trajectoire, la localisation, l'évitement d'obstacles et le stationnement. Dans la majorité des travaux, les fonctions de la navigation sont étudiées de façon séparée où chaque tâche représente une problématique distincte. Cependant, afin de se conformer aux développements exigés dans les différents domaines, des combinaisons entre ces fonctions sont devenues obligatoires (la structure de contrôle nécessite la réalisation simultanée de plus d'une fonction), notamment pour la poursuite et la planification de trajectoires.

La poursuite de trajectoires est une fonction nécessaire au robot mobile pour naviguer le long d'une trajectoire désirée définie par un opérateur humain ou planifiée à l'aide d'une technique de planification. Le problème majeur de cette tâche réside dans le développement des systèmes de commande robustes et fiables face aux incertitudes et à la complexité de l'environnement pour contrôler la position et l'orientation du robot mobile par rapport à la trajectoire de référence. Plusieurs approches ont été proposées pour aborder ce problème. Les plus répondues sont celles qui s'appuient sur le raisonnement humain et la prise de décision. En particulier, la logique floue créée en 1965 par le professeur Lotfi A. Zadeh [Zad-65], était la plus appropriée pour la synthèse des contrôleurs non linéaires prenant en charge cette fonction de navigation [Ant-07], [Lee-03], [Liu-94], [Liu-09], [Mou-15] et [Oll-94]. Cela est dû au fait qu'elle n'exige pas la disponibilité du modèle mathématique du processus à commander et qu'elle permet d'exprimer les pensées et les concepts de l'expert humain décrivant le comportement dynamique de ce processus et/ou la stratégie de sa commande par des variables linguistiques incertaines et imprécises.

La planification de trajectoire est un processus essentiel dans la navigation des robots mobiles. Elle permet de générer des trajectoires au robot mobile en temps réel pour qu'il se déplace sans collision dans n'importe quel environnement. Beaucoup de travaux qui traitent le problème de la planification de trajectoire sont présentés dans littérature. Les techniques proposées sont divisées en deux catégories, les techniques de planification globale [Bha-07], [Bro-99], [Can-87], [Dec-85], [Gel-77], [Lat-91], [Lav-98], [Sed-07], [Ski-90], [Vad-00] et [Yan-04], qui utilisent des informations préalables sur tout l'environnement pour créer le meilleur chemin possible, et les techniques de planification locale [Bor-91], [Fox-96], [Kha-86], [Min-00], [Sim-96], et [Nak-98], également connues par les méthodes réactives qui se reposent sur les informations partielles de l'environnement autour du robot mobile pour mettre à jour le chemin dans un court laps de temps. Les méthodes réactives sont préférables et faciles à implémenter car elles nécessitent moins d'informations environnementales et moins de calcul, et elles sont aussi capables de s'adapter aux changements de l'environnement.

Les approches réactives de planification peuvent être des approches directionnelles qui génèrent des directions vers lesquelles le robot mobile se dirige [Bor-91], [Kha-86] et [Min-00], ou des approches qui travaillent dans l'espace des vitesses dans lequel la recherche des commandes du robot mobile est effectuée [Fox-96], [Sim-96] et [Nak-98]. Ces dernières sont très efficaces grâce à la prise en compte directe du modèle cinématique et des contraintes dynamiques, notamment la fenêtre dynamique qui est la plus convenable dans le cas des déplacements rapides [Arr-02], [Bro-99], [Fox-96] et [Phi-03].

La fenêtre dynamique a été proposée en 1996 par Fox, Burgard et Thrun [Fox-96] pour être utilisée dans la navigation à grande vitesse des robots mobiles. Les commandes sont obtenues dans le domaine de recherche des vitesses en maximisant une fonction "objectif" définie par la somme pondérée de trois termes responsables de la conduite rapide du robot mobile en toute sécurité vers la cible. Malheureusement, cela n'est pas toujours garanti, en raison du problème de choix des poids appropriés. Peu d'études ont été menées pour améliorer cette approche en se basant sur la modification de la fonction "objectif", telle que la combinaison de la fenêtre dynamique avec un filtre à particules pour optimiser la fonction "objectif" [Sch-07], l'utilisation de la logique floue pour changer la vitesse en fonction de la densité d'obstacles [Cho-12], l'exploitation d'un schéma de navigation de la fenêtre dynamique basé sur un contrôleur prédictif plutôt que sur la fonction "objectif" [Kis-12] et l'introduction de la logique floue en deux étapes, la première, pour déterminer l'angle d'orientation et la deuxième, pour calculer la vitesse du robot [Cha-15]. Dans toutes les améliorations

mentionnées précédemment les auteurs ont fait évoluer la fenêtre dynamique en agissant sur les termes pondérés sans prendre en considération l'effet résultant du changement de poids.

Ce chapitre a comme objectif, l'amélioration de la fenêtre dynamique en utilisant un contrôleur flou qui permet de régler en temps réel les coefficients de pondération de la fonction "objectif", et la conception d'un contrôleur flou de poursuite de trajectoires données ou planifiées par la fenêtre dynamique floue. Nous allons commencer par donner un état de l'art sur la robotique mobile en décrivant le modèle le plus utilisé (l'unicycle), ainsi que des concepts de base de la logique floue et de la commande floue, ensuite nous allons présenter la synthèse des approches proposées et leurs validations sous l'environnement Matlab.

I.2. Modélisation et caractéristiques structurelles du robot l'unicycle

D'après la forme du corps, le nombre de roues, leur emplacement et leur type, nous pouvons distinguer plusieurs classes de modèles de robots mobiles. Dans nos travaux nous allons s'intéresser aux robots mobiles unicycles.

I.2.1. Hypothèses

Dans la majorité des stratégies proposées pour résoudre la problématique de la commande des robots mobiles, la synthèse est basée sur un modèle de commande en vitesse au lieu d'un modèle de commande en couple. Cela est dû au fait que le calcul de la commande pour un modèle cinématique est plus facile par rapport à celui d'un modèle dynamique. Ce dernier nécessite l'identification des paramètres de la géométrie et de l'inertie qui sont généralement complexes. C'est pourquoi nous allons considérer que le modèle cinématique en introduisant les hypothèses simplificatrices suivantes :

- le robot mobile est considéré comme un véhicule rigide qui se déplace dans un plan horizontal.
- le robot mobile est doté de roues conventionnelles indéformables, de rayon fixe.
- le contact entre la roue et le sol est réduit à un point.
- les roues sont soumises à la contrainte de rouler sans glisser sur le sol.

I.2.2. Roulement sans glissement

Soit une roue verticale qui roule sans glisser sur un plan horizontal (voir figure I.1). Le roulement sans glissement signifie que la vitesse du point de contact de la roue avec le sol est nulle.

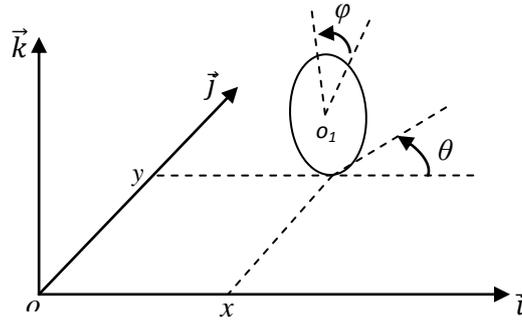


Figure I.1. Représentation de la roue

En utilisant les notations de la figure (I.1), nous pouvons écrire l'expression du vecteur de la vitesse correspondante au point de contact de la roue avec le sol comme suit:

$$\begin{aligned}
 \vec{v} &= \dot{x}\vec{i} + \dot{y}\vec{j} + (\dot{\theta}\vec{k} + \dot{\phi}(-\sin\theta\vec{i} + \cos\theta\vec{j})) \wedge (-r\vec{k}) \\
 &= (\dot{x} - r\dot{\phi}\cos\theta)\vec{i} + (\dot{y} - r\dot{\phi}\sin\theta)\vec{j} \\
 &= 0
 \end{aligned} \tag{I.1}$$

où r est le rayon de la roue, x et y sont les coordonnées du point o_1 dans le repère cartésien $(o, \vec{i}, \vec{j}, \vec{k})$.

D'après l'équation (I.1), nous avons les deux contraintes suivantes:

$$\begin{cases} \dot{x} - r\dot{\phi}\cos\theta = 0 \\ \dot{y} - r\dot{\phi}\sin\theta = 0 \end{cases} \tag{I.2}$$

Pour faire apparaître les composantes des vitesses de la roue, nous réécrivons les contraintes (I.2) comme suit:

$$\begin{cases} \dot{x}\cos\theta + \dot{y}\sin\theta = r\dot{\phi} \\ -\dot{x}\sin\theta + \dot{y}\cos\theta = 0 \end{cases} \tag{I.3}$$

En prenant $v = r\dot{\phi}$ la vitesse de roulement de la roue et $w = \dot{\theta}$ sa vitesse de rotation autour de l'axe \vec{k} , nous obtenons le modèle cinématique de la roue suivant:

$$\begin{bmatrix} \dot{x} \\ \dot{y} \\ \dot{\theta} \end{bmatrix} = \begin{bmatrix} \cos\theta & 0 \\ \sin\theta & 0 \\ 0 & 1 \end{bmatrix} \begin{bmatrix} v \\ w \end{bmatrix} \tag{I.4}$$

I.2.3. Commandabilité

De la perspective de la commande, le modèle cinématique (I.4) possède deux caractéristiques de base:

- Sa linéarisation autour d'un point d'équilibre (x_0, y_0, θ_0) est non commandable parce que le rang de la matrice de commandabilité est inférieur à la dimension de l'état du système (I.4).
- Dans sa forme non linéaire (I.4), il est commandable, car la condition suffisante suivante est remplie [Nij-90]:

Si le système (I.4) est réécrit sous la forme:

$$\dot{q} = f_1(q)v + f_2(q)w \quad (\text{I.5})$$

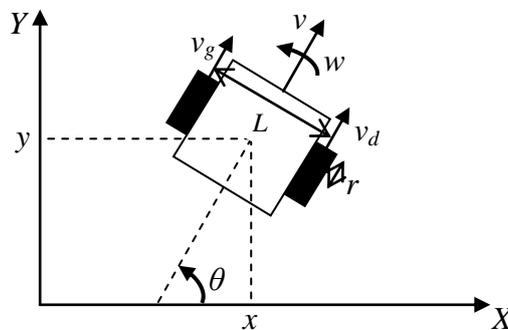
où $q = [x, y, \theta]^T$, $f_1(q) = [\cos(\theta), \sin(\theta), 0]^T$ et $f_2(q) = [0, 0, 1]^T$.

Alors, pour qu'il soit commandable, il faut que la matrice suivante obtenue en calculant le crochet de Lie des champs de vecteur f_1 et f_2 soit inversible pour toute valeur de θ .

$$[f_1(q), f_2(q), [f_1(q), f_2(q)]] = \begin{bmatrix} \cos(\theta) & 0 & -\sin(\theta) \\ \sin(\theta) & 0 & \cos(\theta) \\ 0 & 1 & 0 \end{bmatrix} \quad (\text{I.6})$$

I.2.4. Modèle cinématique du robot de type unicycle

En général, le robot mobile de type unicycle se compose de deux roues motrices fixes placées au milieu du corps et commandées indépendamment (voir figure I.2). Il y'a aussi d'autres robots mobiles de ce type qui peuvent avoir des roues supplémentaires (roues folles) non commandées placées avant ou derrière les roues motrices, ces roues assurent la stabilité du robot mobile et n'interviennent pas dans la cinématique.



I.2. Robot mobile unicycle

En prenant x et y les coordonnées du centre du robot mobile, θ son orientation, v sa vitesse de translation, w sa vitesse de rotation, v_d et v_g les vitesses linéaires de la roue droite et la roue gauche respectivement, r le rayon des roues et L la distance entre elles. À partir des équations (I.1) et (I.2), nous pouvons déduire les contraintes de roulement sans glissement de chaque roue comme suit:

-La roue droite

$$\dot{x} + \frac{L}{2} \dot{\theta} \cos \theta - r \dot{\phi}_d \cos \theta = 0 \quad (\text{I.7})$$

$$\dot{y} + \frac{L}{2} \dot{\theta} \sin \theta - r \dot{\phi}_d \sin \theta = 0 \quad (\text{I.8})$$

-La roue gauche

$$\dot{x} - \frac{L}{2} \dot{\theta} \cos \theta - r \dot{\phi}_g \cos \theta = 0 \quad (\text{I.9})$$

$$\dot{y} - \frac{L}{2} \dot{\theta} \sin \theta - r \dot{\phi}_g \sin \theta = 0 \quad (\text{I.10})$$

Nous multiplions (I.7) et (I.9) par $\cos \theta$, et (I.8) et (I.10) par $\sin \theta$ puis nous faisons la somme de (I.7) avec (I.8) et (I.9) avec (I.10), nous aurons les contraintes suivantes:

$$\dot{x} \cos \theta + \dot{y} \sin \theta + \frac{L}{2} \dot{\theta} = r \dot{\phi}_d \quad (\text{I.11})$$

$$\dot{x} \cos \theta + \dot{y} \sin \theta - \frac{L}{2} \dot{\theta} = r \dot{\phi}_g \quad (\text{I.12})$$

La somme et la soustraction entre (I.11) et (I.12), respectivement, nous donne:

$$\dot{x} \cos \theta + \dot{y} \sin \theta = \frac{r}{2} (\dot{\phi}_d + \dot{\phi}_g) \quad (\text{I.13})$$

$$L \dot{\theta} = r (\dot{\phi}_d - \dot{\phi}_g) \quad (\text{I.14})$$

D'après le modèle cinématique de la roue (I.4), nous effectuons les substitutions nécessaires dans (I.13) et (I.14) pour avoir l'expression de la vitesse linéaire et angulaire en fonction des vitesses linéaires des roues comme suit:

-La vitesse linéaire

$$v = \frac{1}{2} (v_d + v_g) \quad (\text{I.15})$$

-La vitesse angulaire

$$w = \frac{1}{L}(v_d - v_g) \quad (\text{I.16})$$

Par conséquent, le modèle cinématique du robot mobile unicycle sera exprimé par:

$$\begin{cases} \dot{x} = \frac{1}{2}(v_d + v_g) \cos \theta \\ \dot{y} = \frac{1}{2}(v_d + v_g) \sin \theta \\ \dot{\theta} = \frac{1}{L}(v_d - v_g) \end{cases} \quad (\text{I.17})$$

I.3. Concepts fondamentaux de la logique floue

Le raisonnement approximatif ou flou, effectué pour exprimer un comportement, un phénomène ou un processus peut se traduire par l'utilisation des ensembles flous afin de réaliser une interface entre la donnée numérique (quantitative) et la variable linguistique (qualitative) qui est généralement normalisée sur un intervalle spécifique nommé univers de discours. Pour pouvoir manipuler les variables linguistiques, il faut constituer une représentation mathématique à base de fonctions d'appartenance prenant des valeurs comprises entre 0 et 1, correspondant au degré d'appartenance (valeur de vérité) de ces variables linguistiques par rapport aux différents sous ensembles flous de la même classe.

I.3.1. Ensemble flou

L'introduction du concept d'ensembles flous a permis de considérer les états inclus entre le tout et le rien en utilisant des valeurs approximatives (presque 10 m) et des classes avec des frontières mal déterminées (proche ou loin), autrement dit, dans la théorie des ensembles flous, un certain élément peut appartenir en partie à un sous ensemble flou.

Etant donné un ensemble flou A sur un univers de discours X , la représentation mathématique de A peut être sous la forme d'un ensemble de couples ordonnés comme suit :

$$A = \{x, \mu_A(x); x \in X\} \quad (\text{I.18})$$

où $\mu_A(x)$ est la fonction d'appartenance associée à chaque élément x de X définie par:

$$\mu_A(x) = \begin{cases} \text{valeur} \in [0,1] & \text{si } x \in A \\ 0 & \text{si } x \notin A \end{cases} \quad (\text{I.19})$$

I.3.2. Propriétés d'un ensemble flou

Un ensemble flou A dans l'univers de discours X peut avoir les propriétés suivantes :

- **Support:** est un sous ensemble normal (comme dans le cas classique) des éléments de X pour lesquels la fonction d'appartenance mesure une valeur strictement supérieure à 0.

$$supp(A) = \{x \in X; 0 < \mu_A(x) \leq 1\} \quad (I.20)$$

- **Point de croisement:** est un sous ensemble classique des éléments de X , associés à la fonction d'appartenance de valeur égale à 0.5.

$$C(A) = \{x \in X; \mu_A(x) = 0.5\} \quad (I.21)$$

- **Noyau:** est un sous ensemble ordinaire des éléments de X pour lesquels la fonction d'appartenance mesure une valeur égale à 1.

$$N(A) = \{x \in X; \mu_A(x) = 1\} \quad (I.22)$$

L'ensemble flou A est normalisée si $N(A) \neq \emptyset$.

- **Hauteur:** correspond au degré d'appartenance maximale de A

$$H(A) = \max\{\mu_A(x); x \in X\} \quad (I.23)$$

- **Cardinal:** représente la somme de degrés d'appartenance des éléments du support fini de l'ensemble flou A .

$$Card(A) = \sum_{x \in supp(A)} \mu_A(x) \quad (I.24)$$

Si le support est infini, le cardinal de A prend la formule suivante:

$$Card(A) = \int_x \mu_A(x) dm(x) \quad \text{où} \quad \int_x dm(x) = 1 \quad (I.25)$$

I.3.3. Variables linguistiques

Les variables linguistiques désignent les termes exploités pour exprimer les pensées d'experts à base de connaissances vagues durant la description d'une situation, d'un processus

ou d'une grandeur physique comme l'âge (jeune, vieux), la distance (proche, loin), le poids (lourd, léger), la vitesse (rapide, lente), le volume (grand, petit), la quantité (beaucoup, peu), la température (chaude, froide)...etc.

Chaque variable linguistique est déterminée par un triplet (x, X, T_x) où x est le nom de la variable (vitesse, distance...), X l'univers de discours sur lequel x est définie et T_x l'ensemble des valeurs linguistiques que peut prendre x [Buh-94].

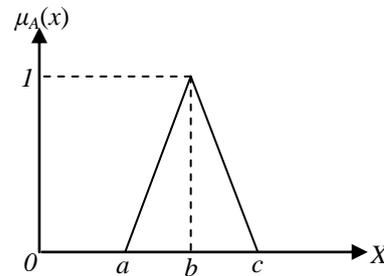
I.3.4. Fonctions d'appartenance

Les fonctions d'appartenance sont des fonctions mathématiques spécifiques, qui expriment les valeurs des variables linguistiques et décrivent leur zone de prédominance par rapport à leur domaine de variation (univers de discours). Elles permettent de manipuler numériquement les variables linguistiques où chaque valeur dans l'intervalle $[0,1]$ est associée à une fonction d'appartenance $\mu_A(x)$ qui mesure le degré avec lequel x appartient à A .

Parmi les formes qu'une fonction d'appartenance peut prendre, nous citons [Buh-94]:

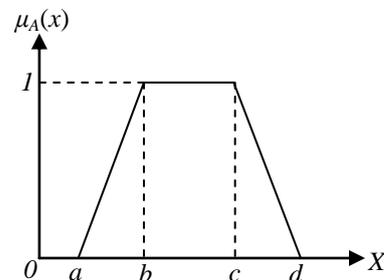
- **Fonction de forme triangulaire:** elle possède une représentation mathématique qui comprend trois paramètres $\{a, b, c\}$:

$$\mu_A(x) = \begin{cases} x-a/b-a & \text{si } x \in [a, b] \\ c-x/c-b & \text{si } x \in [b, c] \\ 0 & \text{ailleurs} \end{cases}$$



- **Fonction de forme trapézoïdale:** sa représentation mathématique dépend de quatre paramètres $\{a, b, c, d\}$:

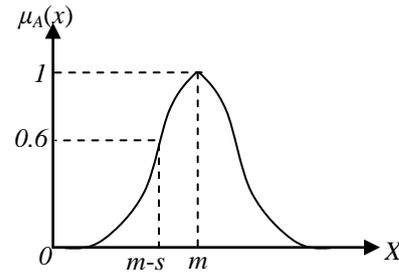
$$\mu_A(x) = \begin{cases} x-a/b-a & \text{si } x \in [a, b] \\ 1 & \text{si } x \in [b, c] \\ d-x/d-c & \text{si } x \in [c, d] \\ 0 & \text{ailleurs} \end{cases}$$



- **Fonction de forme gaussienne:** elle a une représentation mathématique basée sur deux paramètres $\{m, s\}$:

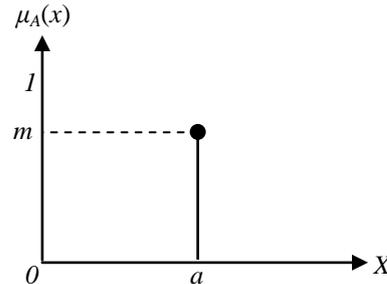
$$\mu_A(x) = \exp\left(-\left(\frac{x-m}{2s}\right)^2\right)$$

avec m : le centre de la gaussienne
et s : sa largeur



- **Fonction de forme singleton:** elle peut être définie par une valeur m comme suit:

$$\mu_A(x) = \begin{cases} m & \text{si } x = a \\ 0 & \text{si } x \neq a \end{cases}$$



I.3.5. Opérateurs flous

Les opérateurs flous permettent de décrire la manière d'interaction des ensembles flous entre eux par des combinaisons logiques. En général, il y a de nombreuses variantes dans ces opérateurs mais les plus utilisés sont les suivants [Buh-94]:

- **Intersection:** il s'agit de la valeur minimale entre les degrés avec lesquels x appartient à l'ensemble A et à l'ensemble B :

$$\mu_{A \cap B}(x) = \min[\mu_A(x), \mu_B(x)] \quad (\text{I.26})$$

- **Union:** cet opérateur correspond à la valeur maximale entre les degrés avec lesquels x appartient à l'ensemble A et à l'ensemble B :

$$\mu_{A \cup B}(x) = \max[\mu_A(x), \mu_B(x)] \quad (\text{I.27})$$

- **Complément:** il donne le degré avec lequel x n'appartient pas à l'ensemble A :

$$\mu_{\neg A}(x) = 1 - \mu_A(x) \quad (\text{I.28})$$

I.3.6. Règles floues

Les règles floues sont le moyen qui sert à former des relations (combinaisons) floues permettant de lier les variables linguistiques afin de faire sortir des conclusions.

L'établissement de ces règles floues est basé sur des connaissances issues de l'expertise humaine qui représente le facteur principal dans la construction d'une base de règles floues capables de décrire l'évolution du système.

Une règle floue peut être définie comme une implication floue entre deux propositions floues. Elle possède la forme 'Si « prémisses » Alors « conclusion »' où la partie prémisses est compatible à la première proposition et la partie conclusion correspond à la deuxième proposition [Buh-94].

$$\text{Si } x \text{ est } A \text{ Alors } y \text{ est } B \quad (\text{I.29})$$

La prémisses et la conclusion d'une règle floue peuvent être constituées de plusieurs propositions combinées par différents opérateurs, telle que la conjonction et la disjonction [Buh-94].

$$\text{Si } x \text{ est } A \text{ et } y \text{ est } B \text{ et... Alors } z_1 \text{ est } C \text{ et } z_2 \text{ est } D \text{ et...} \quad (\text{I.30})$$

Un ensemble de règles floues forme ce que nous appelons une base de règles floues, qui permet l'inclusion du maximum d'informations nécessaires pour une représentation détaillée du système [Buh-94].

$$\begin{aligned} R_1 : & \text{Si } x \text{ est } A_1 \text{ et } y \text{ est } B_1 \text{ et... Alors } z_1 \text{ est } C_1 \text{ et } z_2 \text{ est } D_1 \text{ et...} \\ R_2 : & \text{Si } x \text{ est } A_2 \text{ et } y \text{ est } B_2 \text{ et... Alors } z_1 \text{ est } C_2 \text{ et } z_2 \text{ est } D_2 \text{ et...} \\ & \vdots \\ R_n : & \text{Si } x \text{ est } A_n \text{ et } y \text{ est } B_n \text{ et... Alors } z_1 \text{ est } C_n \text{ et } z_2 \text{ est } D_n \text{ et...} \end{aligned} \quad (\text{I.31})$$

Chaque règle floue R_i ($i=1, \dots, n$) de la base de règles (I.31) est prise en compte lors de l'étape d'inférence si le degré de vérité des propositions de son prédicat est non nulle. Elle a une valeur de vérité μ_{R_i} , dépendante des degrés d'appartenance des variables linguistiques associées à la partie prémisses et à la partie conclusion.

-Cas d'une implication floue entre deux propositions (I.29):

$$\mu_R(x, y) = \text{imp}(\mu_A(x), \mu_B(y)) \quad (\text{I.32})$$

-Cas d'une implication floue qui comprend plusieurs propositions (I.31):

$$\mu_{R_i}(x, y, \dots) = \text{imp}(\{\mu_{A_i}(x) \text{ et } \mu_{B_i}(y) \text{ et...}\}, \{\mu_{C_i}(z_1) \text{ et } \mu_{D_i}(z_2) \text{ et...}\}) \quad (\text{I.33})$$

I.4. Commande floue

En raison de sa synthèse simple et rapide, sa facilité d'implémentation et sa robustesse, la logique floue a été largement exploitée pour résoudre les problèmes de commande de procédés. La commande floue, comme une commande de l'automatique conventionnelle, vise à l'obtention d'un fonctionnement désiré ou correct d'un processus en manipulant une ou plusieurs consignes de ce dernier. Cependant, les contrôleurs flous sont beaucoup plus compréhensibles, maintenables, ne nécessitent pas de puissance de calcul, et capables de prendre en charge les systèmes complexes.

I.4.1. Configuration de base d'un contrôleur flou

Un système de commande floue est constitué principalement de quatre blocs [Buh-94] comme le montre la figure I.3.

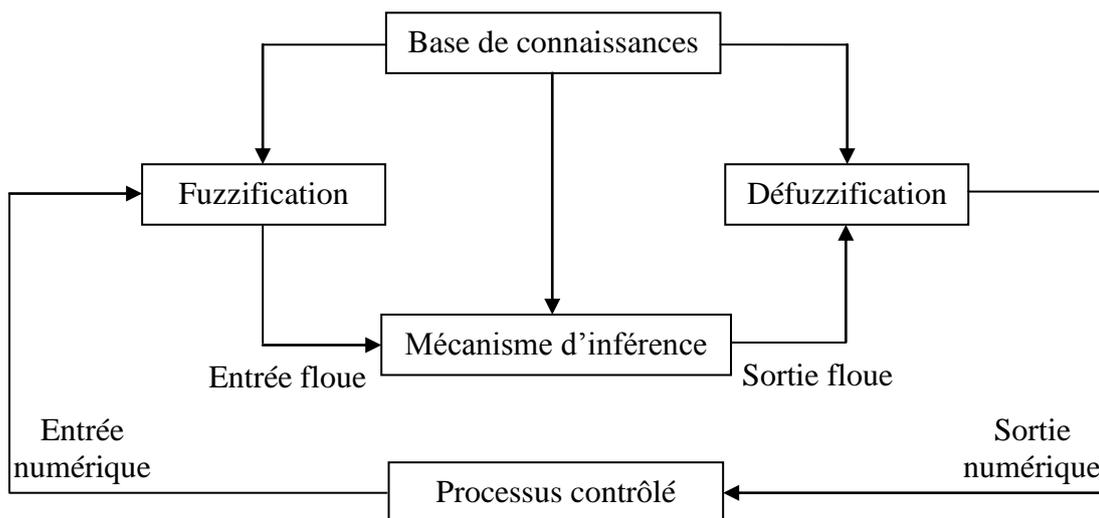


Figure I.3. Structure d'un contrôleur flou

I.4.1.1. Fuzzification

La fuzzification est l'opération effectuée à l'entrée du contrôleur flou afin de convertir les variables d'entrée numériques en variables linguistiques (variables floues). Lors de cette phase, chaque variable d'entrée va avoir un degré de vérité avec lequel sa mesure appartient à un ensemble flou déterminé sur l'univers de discours. Suivant l'exactitude des valeurs d'entrées, deux types de fuzzification peuvent être distingués:

I.4.1.1.1. Fuzzification par singleton

Ce genre de fuzzification est adopté dans le cas où les valeurs d'entrée sont considérées comme précises, et il est très courant dans les applications du contrôle flou grâce à sa facilité de mise en œuvre.

Soit A_x un ensemble flou de l'univers de discours X , la fuzzification par singleton de la valeur x_0 est la partition floue numérique de A_x pour laquelle, la fonction d'appartenance est exprimée par:

$$\mu_{A_x}(x) = \begin{cases} 1 & \text{si } x = x_0 \\ 0 & \text{si } x \neq x_0 \end{cases} \quad (\text{I.34})$$

I.4.1.1.2. Fuzzification non singleton

Dans le cas où les valeurs d'entrée ne sont pas exactes, c'est-à-dire, elles comprennent des erreurs de la mesure ou elles sont perturbées par un bruit, la fonction d'appartenance peut être définie comme suit:

$$\mu_{A_x}(x) = \max \left\{ 0; 1 - \frac{|x - x_0|}{\varepsilon} \right\} \quad (\text{I.35})$$

ou

$$\mu_{A_x}(x) = \exp \left(\frac{-(x - x_0)^T (x - x_0)}{\delta^2} \right) \quad (\text{I.36})$$

où ε et δ sont deux paramètres caractérisant la forme de la fonction d'appartenance μ_{A_x} .

I.4.1.2. Base de connaissances

La construction d'un système flou est réalisée durant l'étape de la base de connaissance en passant par la base de données et la base de règles floues.

I.4.1.2.1. Base de données

La base de données inclut la détermination des ensembles flous des variables d'entrée et de sortie, la répartition de ces ensembles flous dans l'univers de discours et la spécification des fonctions d'appartenance qui les expriment. Lors de la réalisation de ces trois tâches, il est préférable de prendre en considération les conseils suivants:

- L'ensemble des valeurs possibles pour une variable constitue son univers de discours.
- Concernant le nombre des ensembles flous, il faut savoir qu'au départ, un minimum de règles floues est mieux. Considérer 3 ou 5 ensembles flous, c'est-à-dire, chaque variable possède 3 ou 5 fonctions d'appartenance.
- De préférence, choisir des fonctions d'appartenance de formes triangulaires et / ou trapézoïdales, qui sont réparties de façon symétrique et équidistante (voir figure I.4).

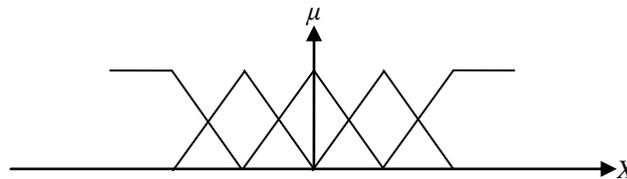


Figure I.4. Fonctions d'appartenance symétriques et équidistantes

-le nombre, la forme et la manière de répartition des fonctions d'appartenance peuvent être modifiés durant l'exécution du processus afin d'atteindre les résultats désirés. Cependant, il faut tenir en compte que le chevauchement insuffisant des fonctions d'appartenance (voir figure I.5) va créer des zones où le contrôleur n'intervient pas, et il faut empêcher le chevauchement excessif, notamment, lorsque μ vaut 1 (voir figure I.6).

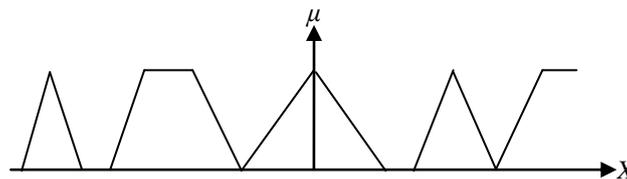


Figure I.5. Chevauchement insuffisant des fonctions d'appartenance

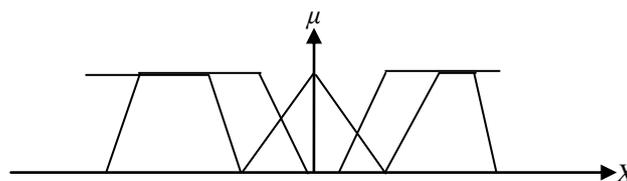


Figure I.6. Chevauchement excessif des fonctions d'appartenance

I.4.1.2.2. Base de règles floues

Une base de règles floues est constituée d'un ensemble de règles floues comme le montre l'équation (I.31) dans la section (§ I.3.6), la structure et le nombre de ces règles, respectivement, dépendent du nombre des variables d'entrée et de sortie et du nombre des ensembles flous décrivant ces variables. L'efficacité de la commande floue est liée à la

crédibilité et à la validité de la base des règles floues, ce qui est la responsabilité de l'expert humain.

N'importe quel système peut être étudié par plusieurs experts du domaine selon différents raisonnements, ce qui conduit à différentes bases de règles floues en termes de structure et de nombre de règles floues, c'est-à-dire un système peut avoir plusieurs bases de règles floues décrivant son comportement, son fonctionnement ou son contrôle. Alors pour constituer une base de règles floues d'un tel système, c'est mieux prendre en considération les bases de règles floues qui déjà existent sachant qu'il faut éviter les contradictions entre les règles floues.

I.4.1.3. Mécanisme d'inférence

Le mécanisme d'inférence consiste à manipuler les règles floues pour convertir la partie floue obtenue après la fuzzification en une nouvelle partie floue qui représente la sortie floue du contrôleur. Cette phase permet de déterminer l'ensemble flou caractérisant la commande en combinant les valeurs d'entrée et les valeurs de sortie par les opérateurs flous décrivant les méthodes d'inférence dont les plus utilisées sont [Buh-94]:

I.4.1.3.1. Méthode d'inférence max-min

Cette méthode représente l'opérateur 'OU' qui est responsable de la disjonction entre les propositions et de la liaison entre les règles floues par l'opération maximum, l'opérateur 'ET' qui permet la conjonction entre les propositions par l'opération minimum, et le mot 'ALORS' qui indique la déclaration de la conclusion de chaque règle par l'opération minimum.

I.4.1.3.2. Méthode d'inférence max-prod

Les opérateurs dans cette méthode sont représentés comme dans la méthode précédente, sauf le mot 'ALORS' est réalisé par l'opération de multiplication.

I.4.1.3.3. Méthode d'inférence somme-prod

Dans cette méthode, l'opérateur 'ET', ainsi que le mot 'ALORS' sont réalisés par l'opération de multiplication, et l'opérateur 'OU' se traduit par la valeur moyenne des degrés d'appartenance utilisés pendant le processus de transformation.

I.4.1.4. Défuzzification

La défuzzification est l'opération effectuée à la sortie du contrôleur flou pour convertir la fonction d'appartenance produite par l'inférence des règles floues en une grandeur de commande précise qui sera appliquée sur le système réel. Nombreuses techniques de défuzzification peuvent être distinguées dont les plus courantes sont [Buh-94]:

I.4.1.4.1. Défuzzification par centre de gravité

Cette technique est énormément exploitée lors de la conception des contrôleurs flous. Il s'agit de déterminer la position du centre de gravité de la fonction d'appartenance résultante $\mu_{res}(X_R)$ où l'abscisse X_R^* (voir l'équation I.37) de ce centre représente la sortie du contrôleur flou, donc elle correspond à la commande du système réel. Les inconvénients de cette méthode résident dans la complexité et la durée du calcul.

$$X_R^* = \frac{\int_{X_{R1}}^{X_{R2}} X_R \cdot \mu_{res}(X_R) \cdot dX_R}{\int_{X_{R1}}^{X_{R2}} \mu_{res}(X_R) \cdot dX_R} \quad (I.37)$$

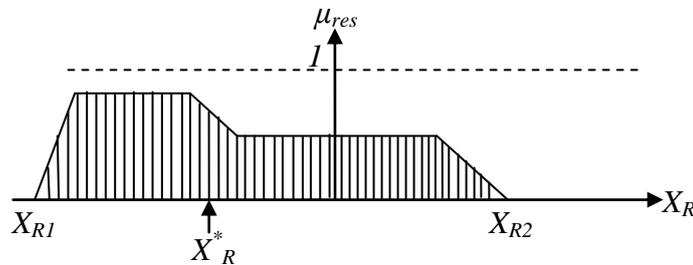


Figure I.7. Défuzzification par centre de gravité

I.4.1.4.2. Défuzzification par valeur maximale

La défuzzification par cette méthode est très simple et rapide par rapport à la méthode précédente, car la valeur de la commande est l'abscisse du maximum de la fonction d'appartenance résultante $\mu_{res}(X_R)$ (voir figure I.8a). L'inconvénient majeur de cette technique apparaît lorsque la fonction d'appartenance possède plusieurs valeurs maximales (voir figure I.8b et I.8c), ce qui se traduit par une indécision ou une discontinuité du signal de commande.

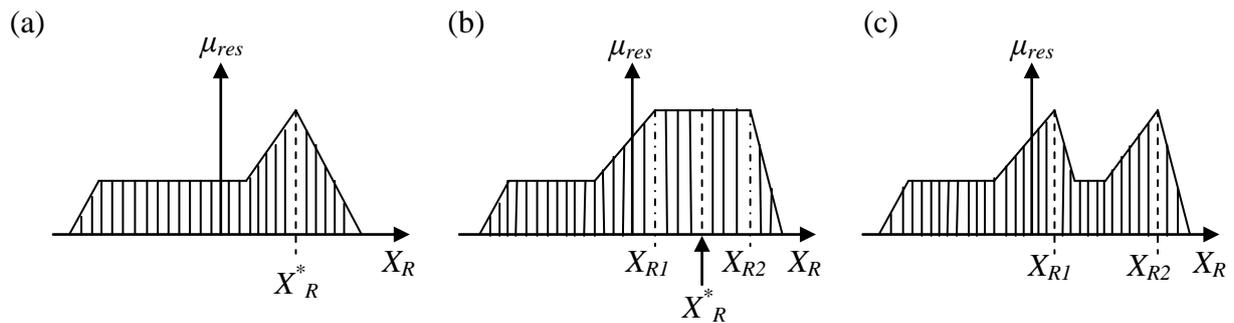


Figure I.8. Défuzzification par valeur maximale

I.4.1.4.3. Défuzzification par la moyenne des maximums

Cette méthode est destinée à la résolution du problème de l'indécision rencontré dans la méthode précédente (voir figure I.8b et I.8c) et cela en calculant la valeur moyenne des maximums.

I.4.2. Différents types de systèmes flous

Généralement, tous les systèmes flous possèdent les mêmes étapes de conception comme présenté précédemment. La seule différence se trouve dans le mécanisme d'inférence, au niveau des règles floues, en particulier, dans la structure des propositions conséquentes. Suivant cette structure, deux types de modèles flous sont proposés: les systèmes flous de Mamdani et ceux de Takagi-Sugeno.

I.4.2.1. Système flou de Mamdani

Ce type de système est considéré comme la première topologie d'un système de commande floue, il est caractérisé par sa facilité de mise en œuvre et sa clarté d'interprétation. Son modèle repose sur des règles floues qui ne comportent que des descriptions linguistiques, c'est-à-dire, les propositions des antécédents et les propositions des conséquents qui constituent les règles floues sont de nature linguistique absolue [Mam-74].

I.4.2.2. Système flou de Takagi-Sugeno

Les systèmes flous de Takagi-Sugeno forment une classe spécifique de contrôleurs flous qui sont capables de commander des processus réels à partir de la description analytique de leurs comportements dynamiques. Ils ont la même structure de ceux de Mamdani, sauf que les propositions des conséquents de ses règles floues sont exprimées numériquement, soit par des constantes, des polynômes, des équations différentielles ou d'autres fonctions [Tak-85].

I.5. Planification de trajectoire

Le problème de planification de trajectoire consiste à commander un robot mobile qui se déplace dans un espace de travail particulier, pour l'amener d'une position initiale (position de départ) à une position finale (position d'arrivée) suivant une trajectoire légitime et sans collision. Dans le même environnement, il peut y avoir différentes trajectoires assurant des déplacements sûrs, cependant, elles doivent respecter les contraintes cinématiques du robot mobile, c'est-à-dire, ce dernier peut les réaliser sans glisser ou se renverser. Nombreuses méthodes ont été développées pour sélectionner la meilleure trajectoire admissible dont celle de l'approche de la fenêtre dynamique.

I.5.1. Principe de l'approche de la fenêtre dynamique

L'approche de la fenêtre dynamique garantit un contrôle instantané au robot mobile à travers des informations locales provenant de la perception de l'environnement qui l'entoure. À chaque pas de temps, un nouveau couple (v,w) de vitesses constantes, de translation et de rotation, parmi les vitesses satisfaisant les restrictions cinématiques et dynamiques dans l'espace de recherche des vitesses est sélectionné. L'application de ce couple de vitesses au robot mobile aboutit à une trajectoire locale circulaire qui le maintient loin des obstacles et l'arrête avant toute collision.

I.5.2. Etapes de l'algorithme de la fenêtre dynamique

En général, l'algorithme de la fenêtre dynamique est résumé en deux étapes principales [Fox-96], la détermination de l'espace de recherche des vitesses possibles et la formulation du processus de sélection du couple de vitesses (v,w) en tant que problème d'optimisation.

I.5.2.1. Espace de recherche des vitesses

L'espace de recherche de la paire de commande (v,w) du robot mobile se compose de trois ensembles de vitesses. Le premier doit respecter les restrictions cinématiques, le deuxième doit satisfaire les contraintes dynamiques et le troisième doit assurer un mouvement sans collision. Le mécanisme de recherche dans l'espace des vitesses, pendant un cycle de l'algorithme, se fait en trois étapes [Fox-96].

I.5.2.1.1. Trajectoires circulaires

Dans l'approche de la fenêtre dynamique, un robot mobile se déplace vers une cible à travers une séquence de trajectoires circulaires (courbures, arcs circulaires), où chacune d'elles est uniquement déterminée par un vecteur de commande (v,w) . Cela implique que la recherche des commandes doit être toujours faisable, ce qui se traduit par un espace de recherche des vitesses à deux dimensions, une répétition de la recherche à chaque intervalle de temps et une stabilité des commandes en l'absence de contraintes de l'environnement.

I.5.2.1.2. Vitesses admissibles

La notion de vitesses admissibles vient des restrictions imposées par les obstacles proches du robot mobile (des contraintes en fonction de la distance entre le robot et le plus proche obstacle). Il s'agit de choisir des vitesses de translation et de rotation qui ne risquent pas de faire entrer le robot mobile en collision avec un obstacle. Autrement dit, une paire (v,w) est considérée admissible, si elle est capable de faire arrêter le robot mobile avant qu'il atteigne le premier obstacle rencontré sur la trajectoire correspondante (la trajectoire produite par l'application du couple (v,w)). Donc pour pouvoir assurer un déplacement sûr du robot mobile, l'ensemble des vitesses admissibles est défini comme suit [Fox-96]:

$$V_a = \left\{ (v, w) \mid v \leq \sqrt{2 \cdot \text{dist}(v, w) \cdot \dot{v}_r} \wedge w \leq \sqrt{2 \cdot \text{dist}(v, w) \cdot \dot{w}_r} \right\} \quad (\text{I.38})$$

où $\text{dist}(v,w)$ est la distance entre le robot mobile et le plus proche obstacle sur l'arc circulaire, \dot{v}_r et \dot{w}_r , respectivement, sont l'accélération de rupture linéaire et angulaire.

I.5.2.1.3. Fenêtre dynamique

Les contraintes dynamiques s'expriment par les limites imposées sur les accélérations exercées par les moteurs. Elles réduisent l'espace de recherche à un espace spécifique appelé fenêtre dynamique. Cette dernière est centrée autour du couple des vitesses actuelles, sa taille dépend des accélérations délivrées et elle n'inclut que les vitesses pouvant être atteintes dans le prochain pas de temps. C'est-à-dire que la fenêtre dynamique restreint l'ensemble des vitesses admissibles à l'ensemble des vitesses suivant [Fox-96]:

$$V_d = \left\{ (v, w) \mid v \in [v_a - \dot{v} \cdot \Delta t, v_a + \dot{v} \cdot \Delta t] \wedge w \in [w_a - \dot{w} \cdot \Delta t, w_a + \dot{w} \cdot \Delta t] \right\} \quad (\text{I.39})$$

où Δt est le laps de temps pendant lequel les accélérations linéaire et angulaire \dot{v} et \dot{w} sont appliquées, v_a et w_a sont les vitesses actuelles de translation et de rotation respectivement.

Après avoir passé par toutes les restrictions, l'espace de recherche résultant V_r (illustré en blanc dans la figure I.9) est déterminé par l'intersection de l'espace des vitesses possibles V_s avec l'espace des vitesses admissibles V_a et la fenêtre dynamique V_d .

$$V_r = V_s \cap V_a \cap V_d \quad (\text{I.40})$$

Finalement, les valeurs appropriées des vitesses dans le prochain pas de temps seront sélectionnées de l'espace V_r en maximisant une fonction de coût afin d'optimiser le mouvement.

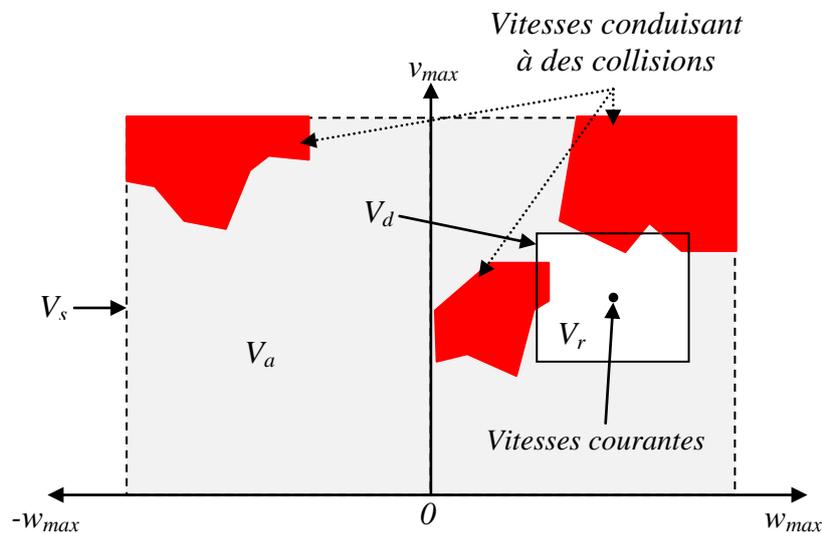


Figure I.9. Fenêtre de sélection des vitesses convenables

I.5.2.2. Optimisation du mouvement

Le fait que dans la méthode de la fenêtre dynamique, la production de trajectoires exprimant le mouvement du robot mobile repose directement sur le choix des vitesses appropriées, alors, pour avoir un mouvement optimal, il faut choisir des vitesses optimales dans l'ensemble V_r . Il s'agit de la maximisation d'une fonction "objectif" qui rassemble trois termes pondérés permettant un déplacement sûr, rapide et optimal [Fox-96]:

$$G(v, w) = \alpha \cdot \text{dire}(v, w) + \beta \cdot \text{dist}(v, w) + \gamma \cdot \text{vite}(v, w) \quad (\text{I.41})$$

où *dire* est l'angle avec lequel le robot mobile va se diriger vers la cible, elle le garde toujours orienté vers cette dernière, et prend sa valeur maximale si l'orientation est absolue, *dist* est la distance entre le robot mobile et le plus proche obstacle sur l'arc circulaire, elle prévient le robot de toute collision, *vite* est la vitesse de progression du robot mobile sur la trajectoire correspondante, α , β et γ sont des facteurs de pondération utilisés pour contrôler l'influence des termes inclus dans la fonction "objectif" sur le comportement du robot mobile.

I.5.3. Inconvénients de la fenêtre dynamique

Au cours du processus de planification de trajectoires à l'aide de la fenêtre dynamique, nous avons constaté que le robot mobile peut atteindre la cible facilement et en toute sécurité dans plusieurs environnements comportant moins d'obstacles. Malheureusement, ce n'est pas le cas si l'environnement devient encombré.

Nous avons effectué de nombreuses simulations en modifiant les valeurs des facteurs de pondération α , β et γ afin d'obtenir un ensemble universel ou un ensemble qui conviendrait à beaucoup d'environnements complexes, mais sans succès car le robot mobile s'est confronté à des situations de blocage ou a emprunté un chemin non optimal [Akk-18a]:

- **Incapable de dépasser ou d'éviter le premier obstacle rencontré sur son chemin**

La figure I.10 montre que si nous choisissons une valeur élevée pour α , une valeur faible pour β et quelle que soit la valeur de γ , le robot mobile sera toujours arrêté par le premier obstacle sur sa trajectoire (pas de progression de mouvement).

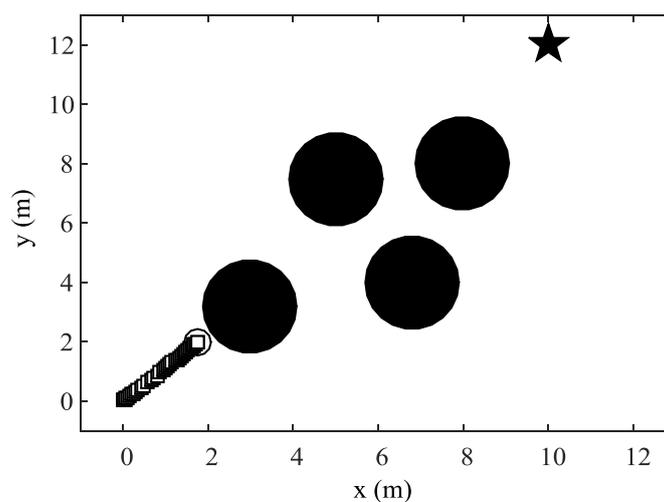


Figure I.10. Robot incapable d'éviter le premier obstacle

- **Incapable de traverser l'espace étroit entre les obstacles**

La figure I.11 illustre que si nous choisissons une petite valeur de α avec une grande valeur de β , le robot va emprunter un long chemin pour atteindre la cible.

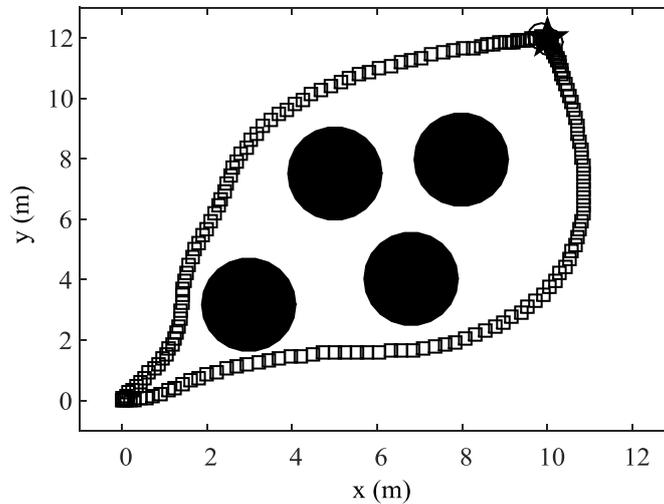


Figure I.11. Robot incapable de passer à travers les espaces étroits

- **Dépasser la cible et se diriger à un espace libre**

La figure I.12 indique que si nous choisissons des grandes valeurs pour β et γ , le robot va perdre son orientation vers la cible.

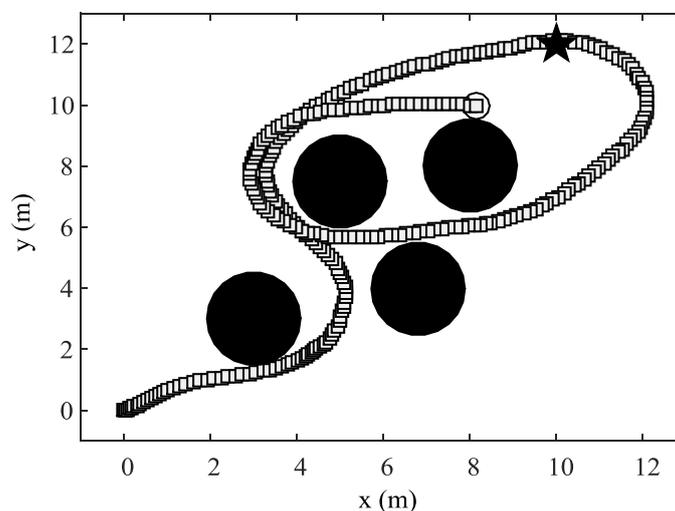


Figure I.12. Robot incapable d'atteindre la cible

- **Atteindre la cible sans effectuer un chemin optimal**

La figure I.13 révèle que si nous choisissons des valeurs égales ou très proches de α , β et γ , pour différents ensembles de cas, le robot peut atteindre la cible avec un chemin plus court par rapport à ceux de la figure I.11; cependant, sans être l'optimal.

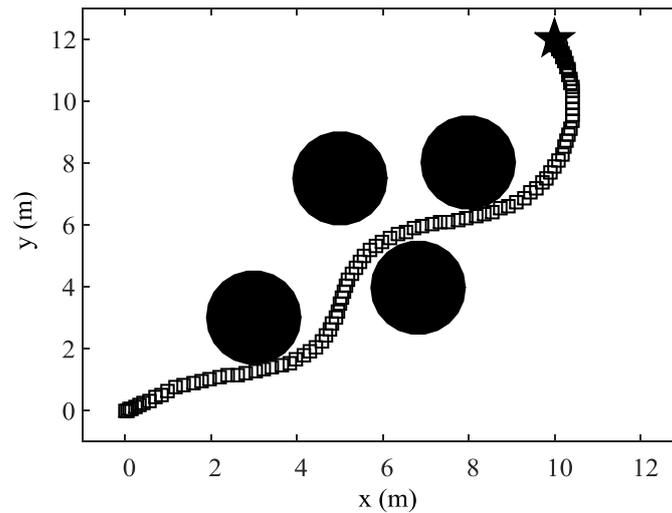


Figure I.13. Robot incapable d'emprunter la trajectoire minimale

I.5.4. Ajustement des facteurs de pondération en temps réel

La fonction "objectif" est responsable de la performance obtenue par la fenêtre dynamique car elle incorpore trois termes exprimant des préférences sur les vitesses, les trajectoires et la direction. Un bon compromis entre ces termes permettra d'avoir une performance optimale. Cependant, ce n'est pas toujours le cas, car chaque terme peut être prioritaire ou partager la priorité suivant les caractéristiques de l'environnement, ce qui exige un changement des priorités en temps réel par l'ajustement dynamique des facteurs de pondération [Akk-18a].

Dans notre travail, nous utilisons un contrôleur flou de Mamdani pour pouvoir manipuler les facteurs de pondération selon la situation courante du robot mobile dans l'environnement. Ce contrôleur flou possède trois entrées qui sont la distance actuelle (*dist*) entre le robot et chaque obstacle détecté sur la trajectoire, la vitesse de progression (*vite*) et l'angle d'orientation vers la cible (*dire*), et trois sorties qui sont les facteurs de pondération α , β et γ [Akk-18a].

L'univers de discours de chaque variable d'entrée et de sortie est constitué de trois sous-ensembles flous qui sont: proche, moyenne et loin pour la variable *dist*, lente, normale et rapide pour la variable *vite*, peu, plus et beaucoup pour la variable *dire* et faible, moyen et élevé pour les facteurs de pondération. La forme et la répartition des fonctions d'appartenance sont représentées sur la figure I.14a et I.14b.

Le tableau I.1 résume les règles floues utilisées dans la conception du contrôleur en se basant sur des arguments logiques tels que:

-Si le robot est loin d'un obstacle, alors les valeurs de α et γ doivent être élevés et la valeur de β doit être faible pour qu'il se déplace rapidement vers la cible.

-Si le robot s'approche d'un obstacle, alors nous devons augmenter la valeur de β et réduire les valeurs de α et γ pour l'empêcher d'entrer en collision.

-Si le robot est en train de passer un obstacle, alors nous devons réduire la valeur de β et augmenter les valeurs de α et γ pour qu'il recommence à avancer rapidement vers la cible.

-Si le robot s'approche de la cible, alors nous devons diminuer les valeurs de β et γ et augmenter la valeur de α pour qu'il s'arrête à la position de cette cible.

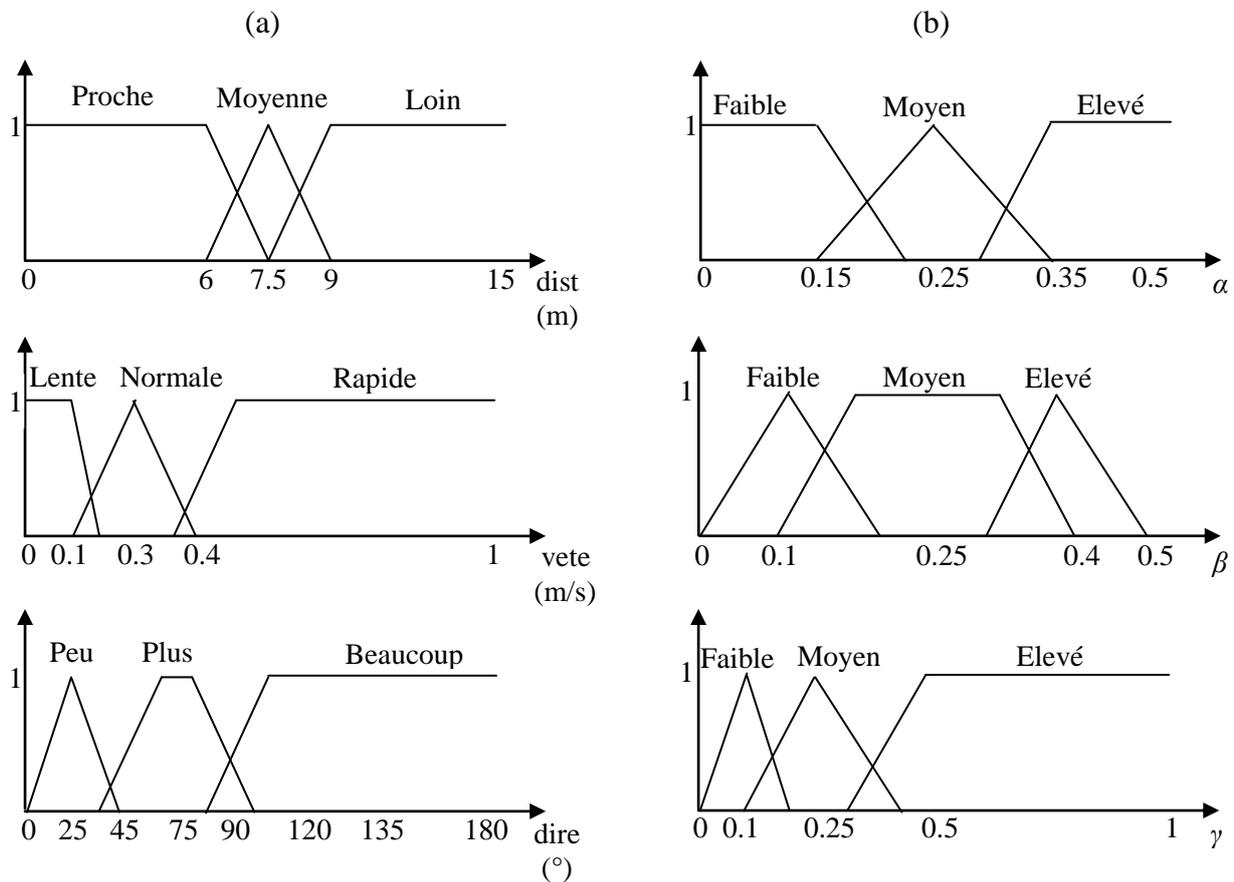


Figure I.14. Fonctions d'appartenance, (a) des entrées et (b) des sorties

Tableau I.1: Les règles floues pour déterminer les facteurs de pondération

Numéro de règle	Entrées linguistiques			Sorties linguistiques		
	<i>dist</i>	<i>vite</i>	<i>dire</i>	α	β	γ
1	Proche	Lente	Peu	Faible	Elevé	Faible
2	Proche	Normale	Plus	Faible	Elevé	Faible
3	Proche	Rapide	Beaucoup	Faible	Elevé	Faible
4	Moyenne	Lente	Peu	Moyen	Faible	Moyen
5	Moyenne	Normale	Plus	Moyen	Faible	Moyen
6	Moyenne	Rapide	Beaucoup	Faible	Moyen	Moyen
7	Loin	Lente	Peu	Elevé	Faible	Elevé
8	Loin	Normale	Plus	Elevé	Faible	Elevé
9	Loin	Rapide	Beaucoup	Elevé	Faible	Elevé

Dans la figure I.15, nous montrons l'organigramme de la fenêtre dynamique floue proposée. La première étape consiste à initialiser les facteurs de pondération, puis dans la deuxième étape le robot va recevoir des informations sur les obstacles proches et la cible, ensuite lors de la troisième étape la fenêtre dynamique est appliquée, les poids sont ajustés et la trajectoire est générée, dans la quatrième étape, si le robot est arrivé à la cible, alors la tâche est terminée, sinon nous revenons à la deuxième étape et ainsi de suite, à la fin nous allons avoir une séquence de trajectoires.

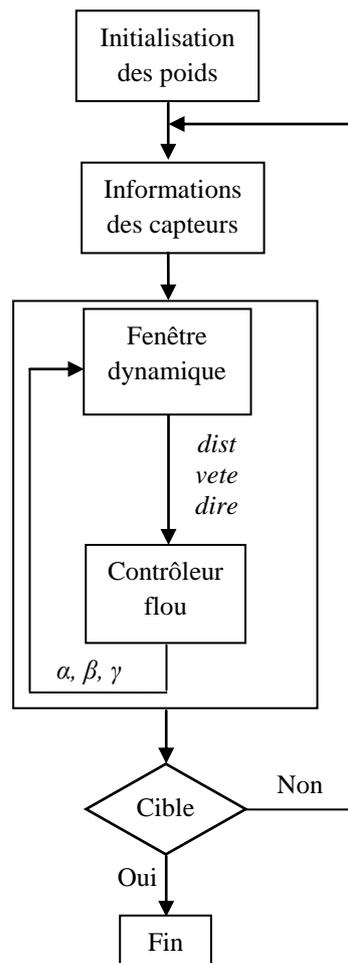


Figure I.15. Organigramme de la fenêtre dynamique floue

I.6. Poursuite de trajectoire

Le problème de poursuite de trajectoire réside dans la détermination des commandes à chaque pas de temps afin que le robot mobile suive une trajectoire de référence. Cela nécessite la perception de l'environnement en temps réel, ce qui permet de définir la position et l'orientation du robot mobile par rapport à la trajectoire désirée. Si à un certain instant, le

robot mobile est mal placé vis-à-vis de la position désirée, le contrôleur doit le ramener vers cette dernière en minimisant les erreurs de position et d'orientation.

I.6.1. Stratégie de contrôle

Il s'agit de la synthèse des commandes floues permettant la poursuite des trajectoires de formes particulières exprimées à l'aide des équations mathématiques, et des trajectoires planifiées par la fenêtre dynamique floue. Dans les deux cas, nous traitons chaque point de la trajectoire désirée comme une position cible que le robot mobile va atteindre, c'est-à-dire, les points constituant la trajectoire de référence représentent les différentes positions instantanées d'une cible dynamique. Le rôle du contrôle flou est la mise à jour de la position et l'orientation du robot mobile en temps réel par rapport à ces points ordonnés et orientés en appliquant la vitesse linéaire et angulaire, ainsi que l'amélioration de l'orientation en manipulant l'accélération ou la décélération et la vitesse de rotation.

I.6.2. Contrôleur flou de position et d'orientation

Soit $v_c = (x \ y \ \theta)^T$ le vecteur d'état qui décrit la position courante du robot mobile et $v_d = (x_d \ y_d \ \theta_d)^T$ le vecteur qui exprime la position de la cible (voir figure I.16). L'objectif du contrôleur flou est d'amener le robot mobile de n'importe quelle position actuelle vers n'importe quelle position désirée, tout en assurant la convergence rapide des erreurs de poursuite vers zéro.

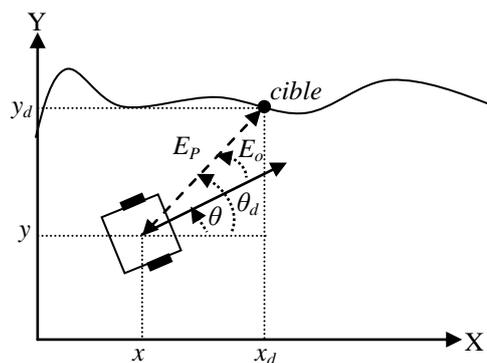


Figure I.16. Suivi de trajectoire basé sur l'erreur de position et d'orientation

Le contrôleur flou conçu est de type Mamdani, il a deux variables d'entrée, deux variables de sortie et une base de règles de trente cinq règles floues.

- **Variables d'entrée**

-La première variable est l'erreur de position E_p qui indique la distance entre la position actuelle du robot mobile et la position désirée (voir figure I.16). Elle est

décrite par cinq variables linguistiques: Z(zéro), P(positif), M(moyenne), G(grande) et TG(très grande) qui sont représentées via des fonctions d'appartenance de type triangulaire et trapézoïdale dans l'univers de discours $[0, 2m]$ (voir figure I.17a).

$$E_p = \sqrt{(x_d - x)^2 + (y_d - y)^2} \quad (I.42)$$

-La deuxième variable est l'erreur d'orientation E_o qui désigne l'angle entre l'orientation courante du robot mobile et l'orientation désirée (voir figure I.16). Elle est caractérisée par sept variables linguistiques: NG (négative grande), NM(négative moyenne), NP(négative petite), Z(zéro), PP(positive petite), PM(positive moyenne) et PG(positive grande) qui sont exprimées à l'aide des fonctions d'appartenance triangulaires et trapézoïdales dans l'univers de discours $[-4, 4rad]$ (voir figure I.17a).

$$E_o = \theta_d - \theta \quad (I.43)$$

où

$$\theta_d = \begin{cases} \arctg\left(\frac{\Delta y}{\Delta x}\right) & \text{si } x_d - x > 0 \text{ et } y_d - y > 0 \\ -\arctg\left(\frac{|\Delta y|}{\Delta x}\right) & \text{si } x_d - x > 0 \text{ et } y_d - y < 0 \\ -\pi - \arctg\left(\frac{\Delta y}{|\Delta x|}\right) & \text{si } x_d - x < 0 \text{ et } y_d - y \geq 0 \\ -\pi + \arctg\left(\frac{|\Delta y|}{\Delta x}\right) & \text{si } x_d - x < 0 \text{ et } y_d - y < 0 \end{cases} \quad (I.44)$$

- **Variables de sortie**

-La première variable est la vitesse linéaire ou de translation v qui permet le déplacement du robot mobile. Elle est définie par cinq variables linguistiques: Z, P, M, G et TG qui sont décrites à travers des fonctions d'appartenance triangulaires dans l'univers de discours $[0, 2.5m/s]$ (voir figure I.17b).

-La deuxième variable est la vitesse angulaire ou de rotation w qui dirige le robot mobile. Elle est spécifiée par sept variables linguistiques: NG, NM, NP, Z, PP, PM et PG qui sont présentées sous forme des fonctions d'appartenance triangulaires et trapézoïdales dans l'univers de discours $[-4, 4rad/s]$ (voir figure I.17b).

Les règles floues combinant les variables d'entrée et de sortie sont obtenues à partir des raisonnements logiques issus de la connaissance de la conduite automobile (voir tableau I.2).

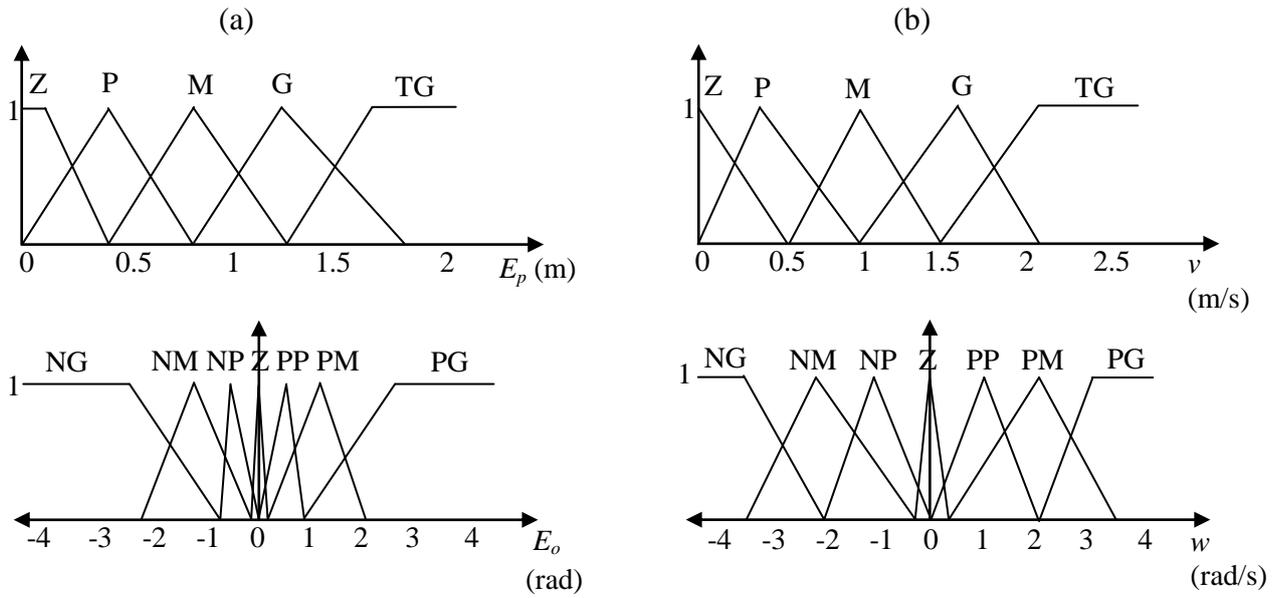


Figure I.17. Fonctions d'appartenance des variables, (a) d'entrée et (b) de sortie

Tableau I.2: Les règles floues de la vitesse linéaire et angulaire

v/w		E_o						
		NG	NM	NP	Z	PP	PM	PG
E_p	Z	P/PG	Z/PG	Z/PM	P/Z	Z/NM	Z/NG	P/NG
	P	P/PG	P/PM	P/PM	M/Z	P/NM	P/NM	P/NG
	M	P/PG	P/PM	M/PP	M/Z	M/NP	P/NM	P/NG
	G	P/PM	M/PM	G/PP	G/Z	G/NP	M/NM	P/NM
	TG	P/PM	M/PM	G/PP	TG/Z	G/NP	M/NM	P/NM

I.6.3. Contrôleur flou d'amélioration de l'orientation

Soit $\vec{v}_p = x\vec{i} + y\vec{j}$ le vecteur de la position du robot mobile, $\vec{v}_d = x_d\vec{i} + y_d\vec{j}$ le vecteur de la position désirée, $\vec{v} = \vec{v}_d - \vec{v}_p = (x_d - x)\vec{i} + (y_d - y)\vec{j}$ le vecteur de la distance entre le robot mobile et la cible, et $\vec{v}_o = \cos(\theta)\vec{i} + \sin(\theta)\vec{j}$ le vecteur de l'orientation du robot mobile (voir figure I.18), le contrôleur flou établi sert à maintenir la stabilité du robot mobile le long de la trajectoire désirée, c'est-à-dire, nous voulons qu'à chaque pas de temps, le vecteur de l'orientation soit toujours dans le sens du vecteur de la distance, mathématiquement, il s'agit du produit vectoriel nul de \vec{v}_o et \vec{v} ($\vec{v}_o \wedge \vec{v} = 0$).

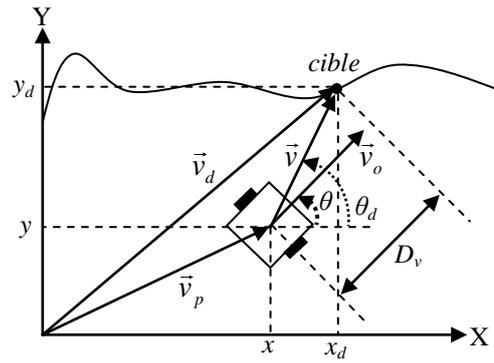


Figure I.18. Amélioration de l'orientation

L'élaboration du contrôleur flou est basée sur le concept de Mamdani en utilisant deux variables d'entrée, deux variables de sortie et une base de règles constituée de quatorze règles floues.

- **Variables d'entrée**

-La première variable est le produit vectoriel P_v entre le vecteur de l'orientation courante et le vecteur de la distance divisé par sa norme (vecteur unitaire), elle permet de savoir la direction et le sens du vecteur de l'orientation du robot mobile par rapport au vecteur de la distance. Cette variable est exprimée en utilisant sept variables linguistiques: NG (négative grande), NM(négative moyenne), NP(négative petite), Z(zéro), PP(positive petite), PM(positive moyenne) et PG(positive grande) qui sont illustrées par des fonctions d'appartenance triangulaires et trapézoïdales dans l'univers de discours $[-3, 3rad]$ (voir figure I.19a).

$$P_v = \vec{v}_o \wedge \frac{\vec{v}}{\|\vec{v}\|} = \frac{(y_d - y)\cos(\theta) - (x_d - x)\sin(\theta)}{\sqrt{(x_d - x)^2 + (y_d - y)^2}} \quad (I.45)$$

-La deuxième variable est la distance verticale D_v entre le robot mobile et l'horizon de la cible (nous voulons dire l'horizon vertical avec l'orientation actuelle du robot mobile (voir figure I.18)). Elle est déterminée par sept variables linguistiques: NG, NM, NP, Z, PP, PM et PG qui sont définies dans l'univers de discours $[-1, 1m]$ via des fonctions d'appartenance triangulaires et trapézoïdales (voir figure I.19a).

$$D_v = \|\vec{v}\| \cos E_o \quad (I.46)$$

• **Variables de sortie**

-La première variable est la vitesse angulaire ou de rotation w (voir figure I.19b), elle possède la même description et le même objectif comme dans le cas du contrôleur flou de la position et de l'orientation (voir variables de sortie dans la section § I.6.2).

-La deuxième variable \dot{v} peut être une accélération ou une décélération si elle est respectivement, positive ou négative. Elle est linguistiquement décrite par cinq variables: NM, NP, Z, PP et PM qui sont interprétées sous forme de fonctions d'appartenance triangulaires et trapézoïdales dans l'univers de discours $[-1, 1m/s^2]$ (voir figure I.19b).

Les règles floues utilisées pour améliorer l'orientation du robot mobile (réduire l'erreur d'orientation) sont présentées dans les tableaux: I.3 et I.4.

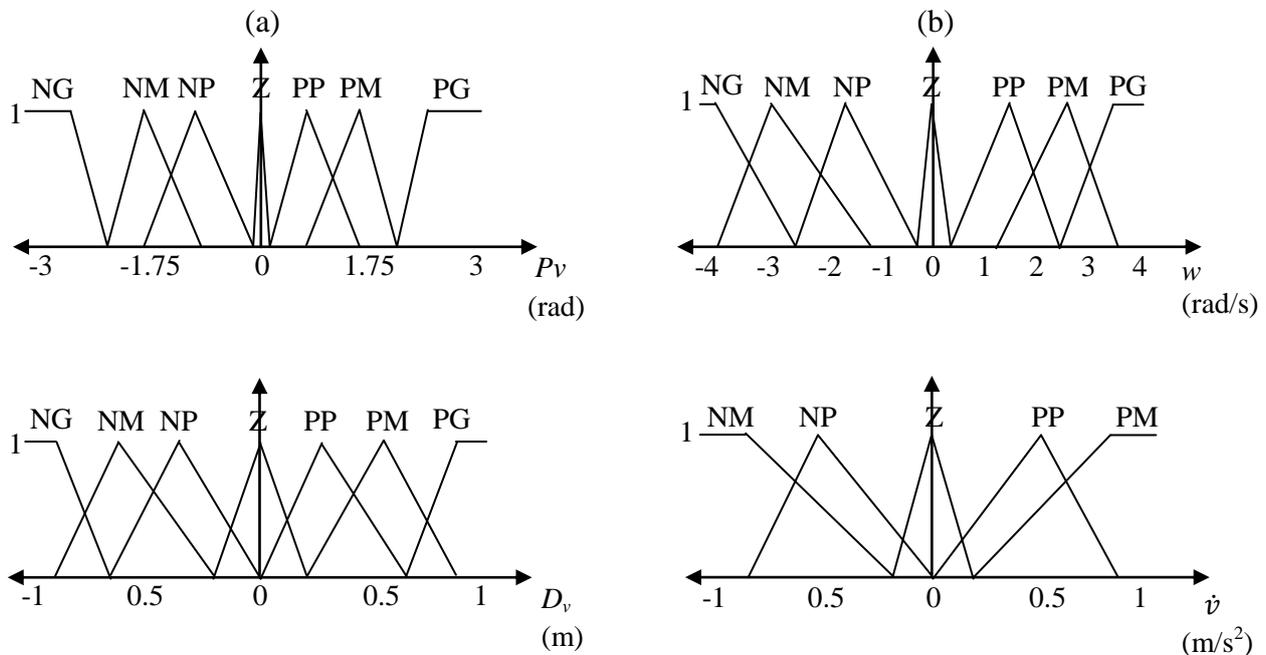


Figure I.19. Fonctions d'appartenance des variables, (a) d'entrée et (b) de sortie

Tableau I.3: Les règles floues de la vitesse angulaire

P_v	NG	NM	NP	Z	PP	PM	PG
w	NG	NM	NP	Z	PP	PM	PG

Tableau I.4: Les règles floues de la variation dans la vitesse linéaire

D_v	NG	NM	NP	Z	PP	PM	PG
\dot{v}	NM	NP	Z	Z	Z	PP	PM

I.6.4. Contrôleur flou global de poursuite

Nous considérons les deux contrôleurs flous présentés précédemment comme des sous contrôleurs d'un contrôleur flou global (voir figure I.20). Le premier sous contrôleur flou 'SCF1' intervient lorsqu'il s'agit d'une grande valeur de l'erreur de la position ' E_{pmax} ' pour assurer une convergence rapide du robot mobile vers la trajectoire désirée. Une fois que ce dernier est proche de la destination requise, le deuxième sous contrôleur flou 'SCF2' prend en charge le robot pour le garder bien orienté. Donc le SCF1 et le SCF2 ne fonctionnent pas simultanément et ils sont exclusivement actifs en fonction de la valeur de l'erreur de position. Pour réaliser l'alternance entre ces deux contrôleurs, nous utilisons une méthode de commutation simple basée sur une variable booléenne ψ comme suit [Oua-08]:

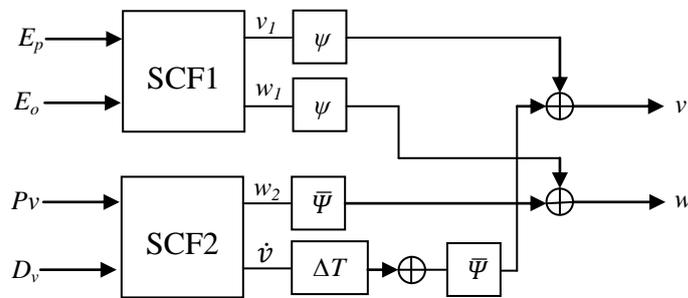


Figure I.20. Contrôleur flou global de poursuite

Notons v_1 et w_1 les variables de sortie du SCF1 ainsi que w_2 et \dot{v} les variables de sortie du SCF2, avec ΔT le pas du temps. Les variables de sortie du contrôleur flou global sont obtenues par:

$$\begin{cases} v = \psi v_1 + \bar{\psi} \Delta T \dot{v} \\ w = \psi w_1 + \bar{\psi} w_2 \end{cases} \quad (\text{I.47})$$

où

$$\psi = \begin{cases} 1 & \text{si } E_p \geq E_{pmax} \\ 0 & \text{sinon} \end{cases} \quad (\text{I.48})$$

La figure I.21 montre un diagramme bloc qui exprime la stratégie proposée pour le contrôle flou en poursuite du robot mobile.

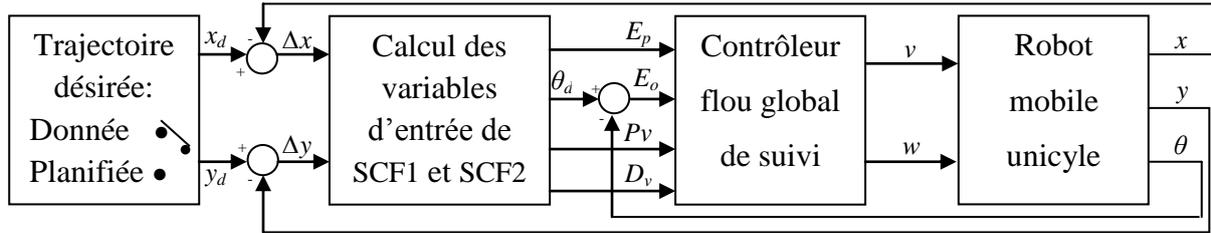


Figure I.21. Diagramme bloc de la stratégie de contrôle proposée

I.7. Exemples de simulation

Dans le but de mettre en évidence la performance obtenue par les approches développées dans ce chapitre, la fenêtre dynamique floue est implémentée dans Matlab pour planifier des trajectoires dans des environnements simples ou complexes. Le contrôleur flou global proposé est mis en œuvre pour forcer le robot mobile à suivre des trajectoires désirées qui sont spécifiées par des équations mathématiques ou issues de la fenêtre dynamique floue.

Lors de la planification de trajectoires par la fenêtre dynamique floue, les poids de la fonction "objectif" doivent satisfaire la condition: $\alpha + \beta + \gamma \leq 1$ à chaque pas de temps (à chaque itération).

Outre la trajectoire prévue par la fenêtre dynamique floue, le robot mobile va suivre les trajectoires décrites par les équations suivantes:

-Trajectoire en ligne droite

$$\begin{cases} x_d = t \\ y_d = \frac{t}{2} + \frac{3}{2} \end{cases}$$

-Trajectoire sinusoïdale

$$\begin{cases} x_d = \frac{t}{2} \\ y_d = \frac{\pi}{30} \sin \frac{t}{2} \end{cases}$$

-Trajectoire en sinus et cosinus

$$\begin{cases} x_d = \frac{t}{2} \\ y_d = 0.1 \left[\sin \frac{t}{4} + \cos \frac{t}{2} \right] \end{cases}$$

-Trajectoire circulaire

$$\begin{cases} x_d = \sin\left(\frac{t-\pi}{2}\right) \\ y_d = \sin\frac{t}{2} \end{cases}$$

-Trajectoires ovales (Lemniscates)

$$\begin{cases} x_d = 1.1 + 0.7 \sin\frac{2\pi}{30}t \\ y_d = 0.9 + 0.7 \sin\frac{4\pi}{30}t \end{cases},$$

$$\begin{cases} x_d = 2.5 \sin\frac{2\pi}{30}t \\ y_d = -2 \sin\frac{2\pi}{20}t \end{cases}$$

I.7.1. Résultats de simulation de planification

La figure I.22a révèle la trajectoire planifiée par la fenêtre dynamique floue dans le même environnement que celui des figures I.10-I.13. Les figures I.22b-I.22d montrent comment les poids de la fonction "objectif" changent en fonction de la situation environnementale.

En comparant la trajectoire de la figure I.22a avec celles des figures I.10-I.13, il en ressort que la fenêtre dynamique floue proposée a amélioré la performance du robot mobile en lui permettant d'atteindre la cible en toute sécurité à travers un chemin plus court.

Les résultats présentés sur la figure I.23 démontrent également la capacité de la fenêtre dynamique floue à conduire le robot mobile vers la cible dans un environnement complexe (encombré) et illustrent l'évolution des facteurs de pondération correspondants.

Dans la figure I.24a, le robot mobile peut atteindre la cible à partir de plusieurs positions de départ. La figure I.24b montre que le robot mobile est capable de se déplacer d'une même position initiale vers différentes positions cibles, ce qui confirme la fiabilité de l'approche développée.

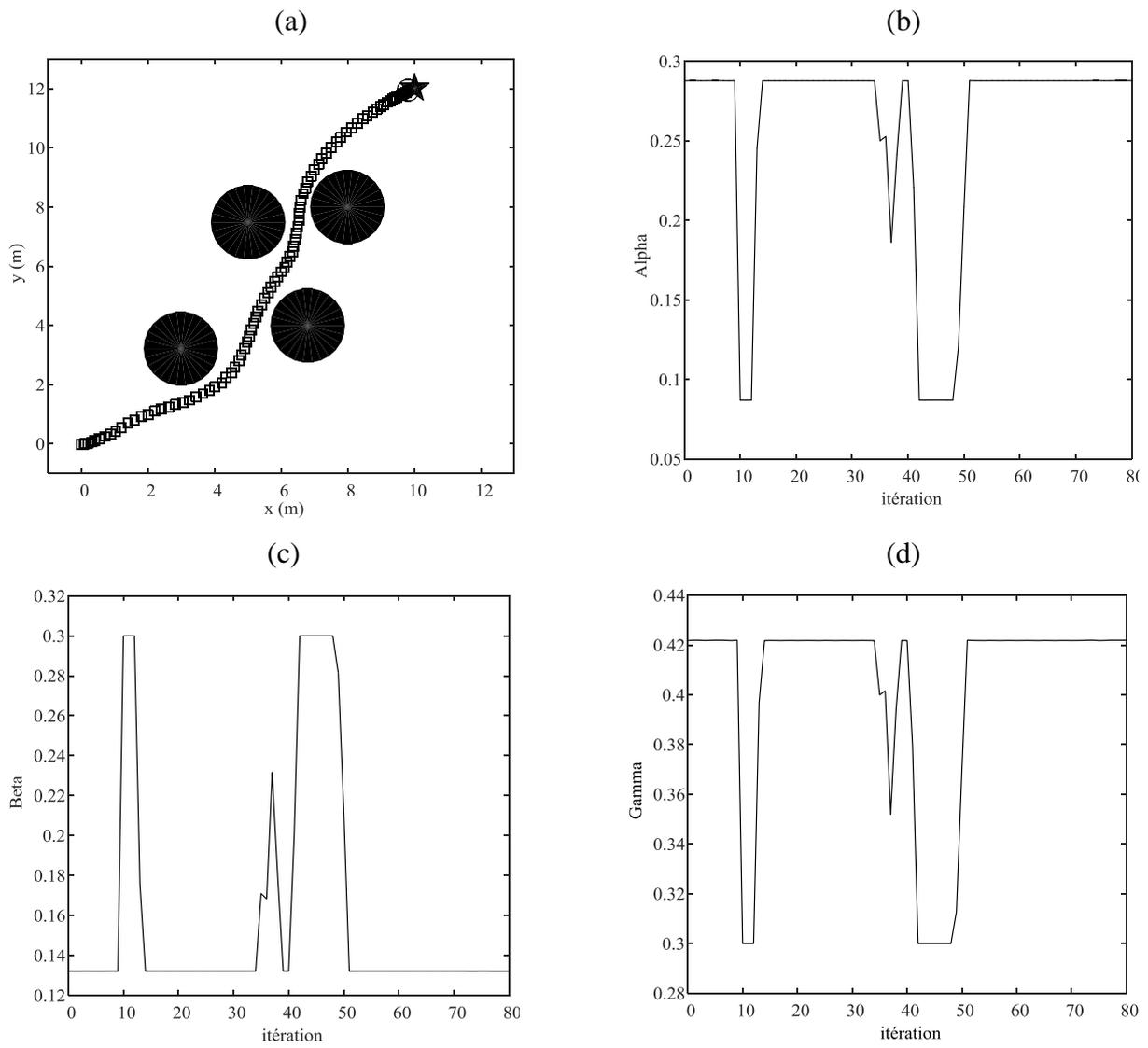
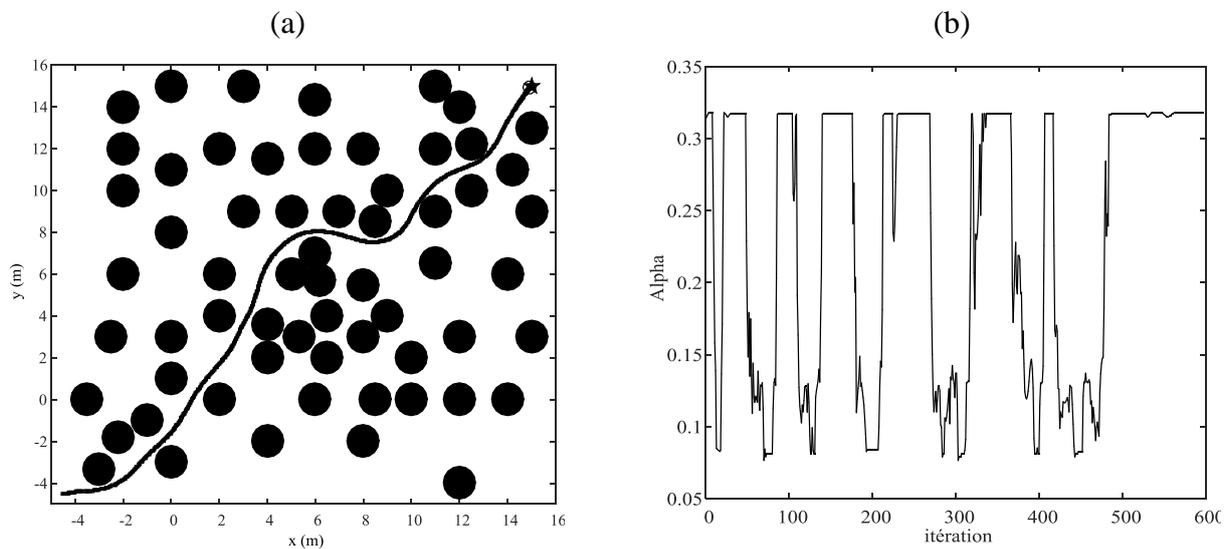


Figure I.22. (a) Trajectoire optimale obtenue par la fenêtre dynamique floue, (b), (c) et (d) facteurs de pondération correspondants



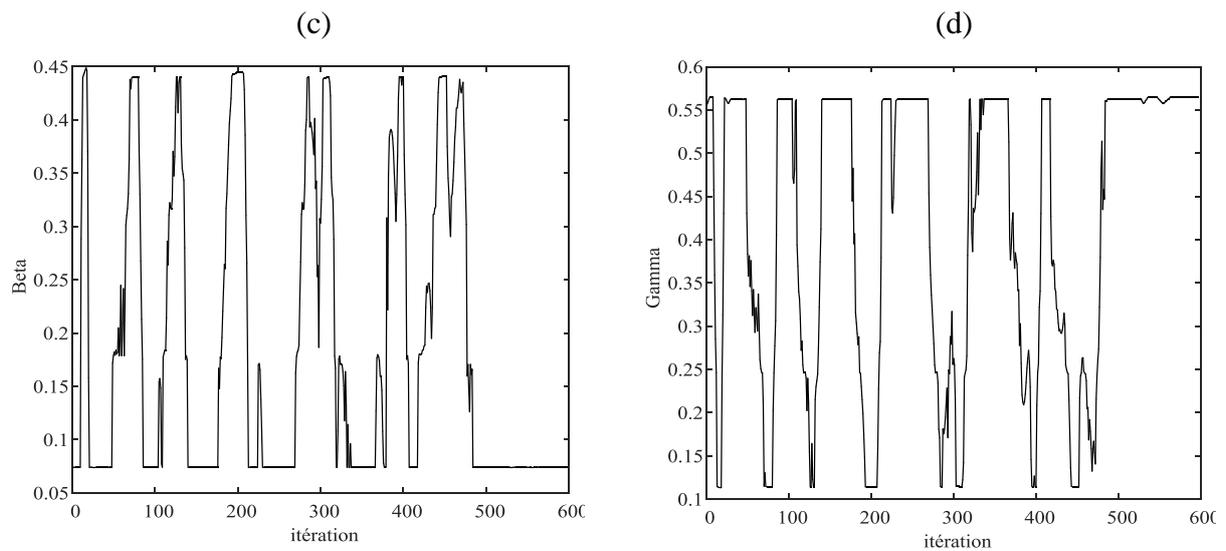


Figure I.23. (a) Trajectoire optimale obtenue par la fenêtre dynamique floue dans un environnement complexe, (b), (c) et (d) facteurs de pondération correspondants

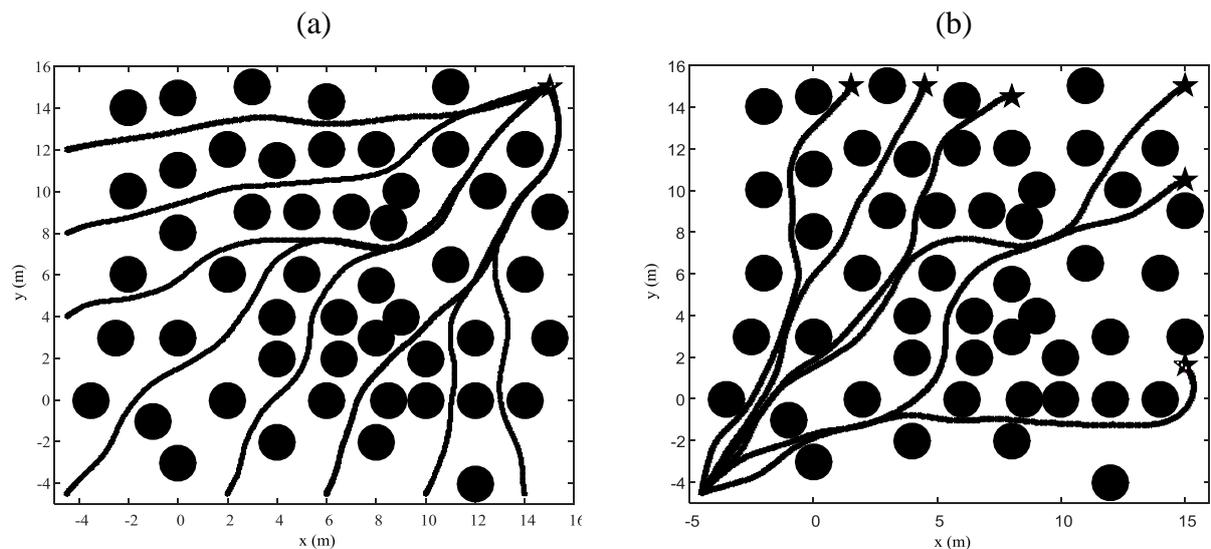


Figure I.24. Planification de trajectoires par la fenêtre dynamique floue, (a) à partir de plusieurs positions et (b) vers différentes cibles

I.7.2. Résultats de simulation de poursuite

Les figures I.25a-I.29a représentent la poursuite des trajectoires désirées effectuée par le robot mobile en appliquant le contrôleur flou synthétisé. Les figures I.25b-I.29b montrent les erreurs de poursuite produites lors du processus de suivi. Les figures I.25c et I.25d-I.29c et I.29d illustrent les signaux de commande (les vitesses linaires et angulaires).

La figure (I.30) justifie l'éligibilité du contrôleur flou conçu par le suivi des trajectoires lemniscates.

À partir des résultats de simulation, la stratégie de contrôle flou élaborée permet de réaliser un bon suivi des trajectoires planifiées ou données, la qualité de la performance s'exprime par la convergence rapide des erreurs de poursuite vers zéro, les vitesses linéaires et angulaires ont une forme lisse et continue.

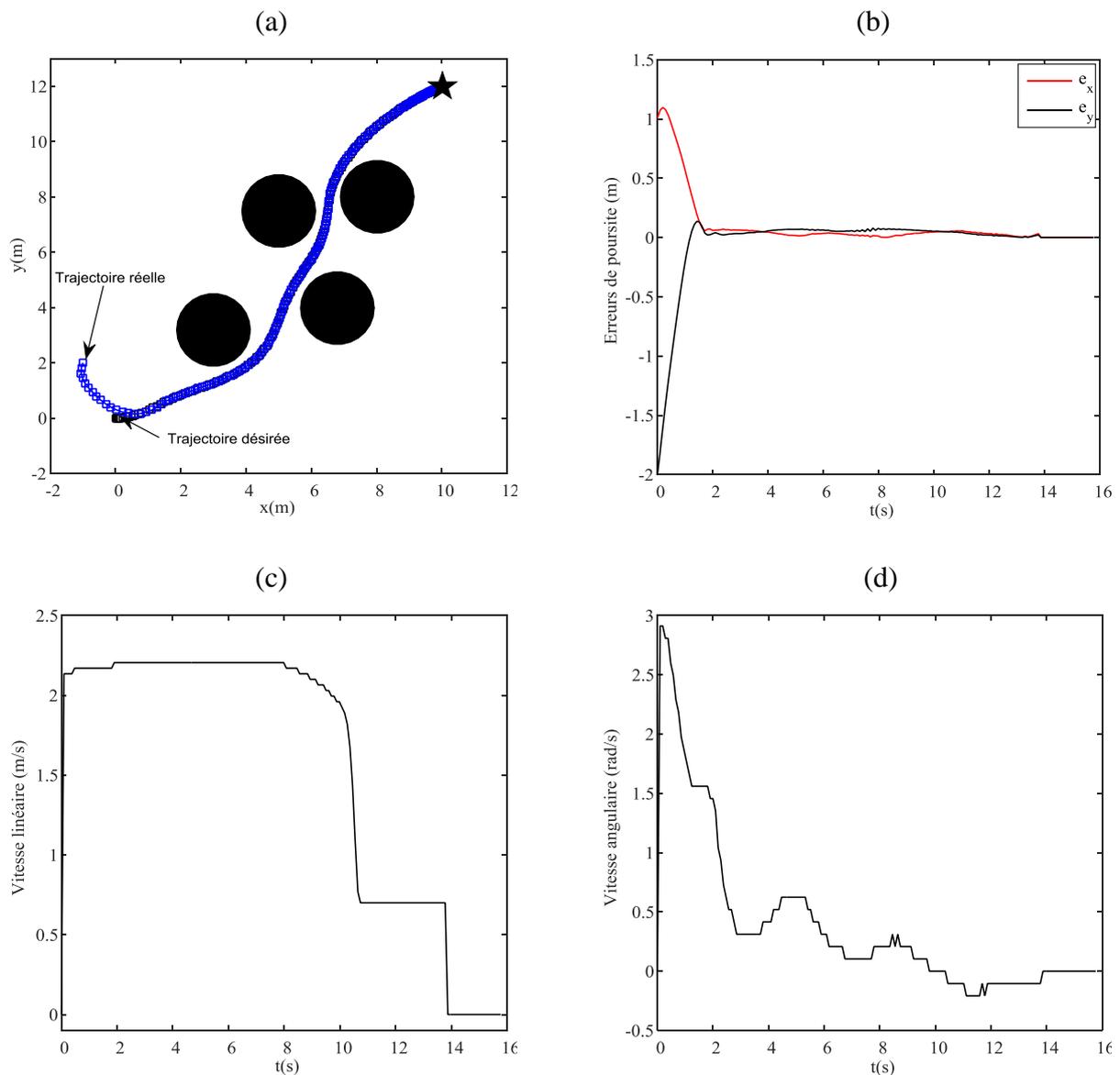


Figure I.25. (a) Poursuite de trajectoire, (b) erreurs de poursuite, (c) vitesse linéaire et (d) vitesse angulaire

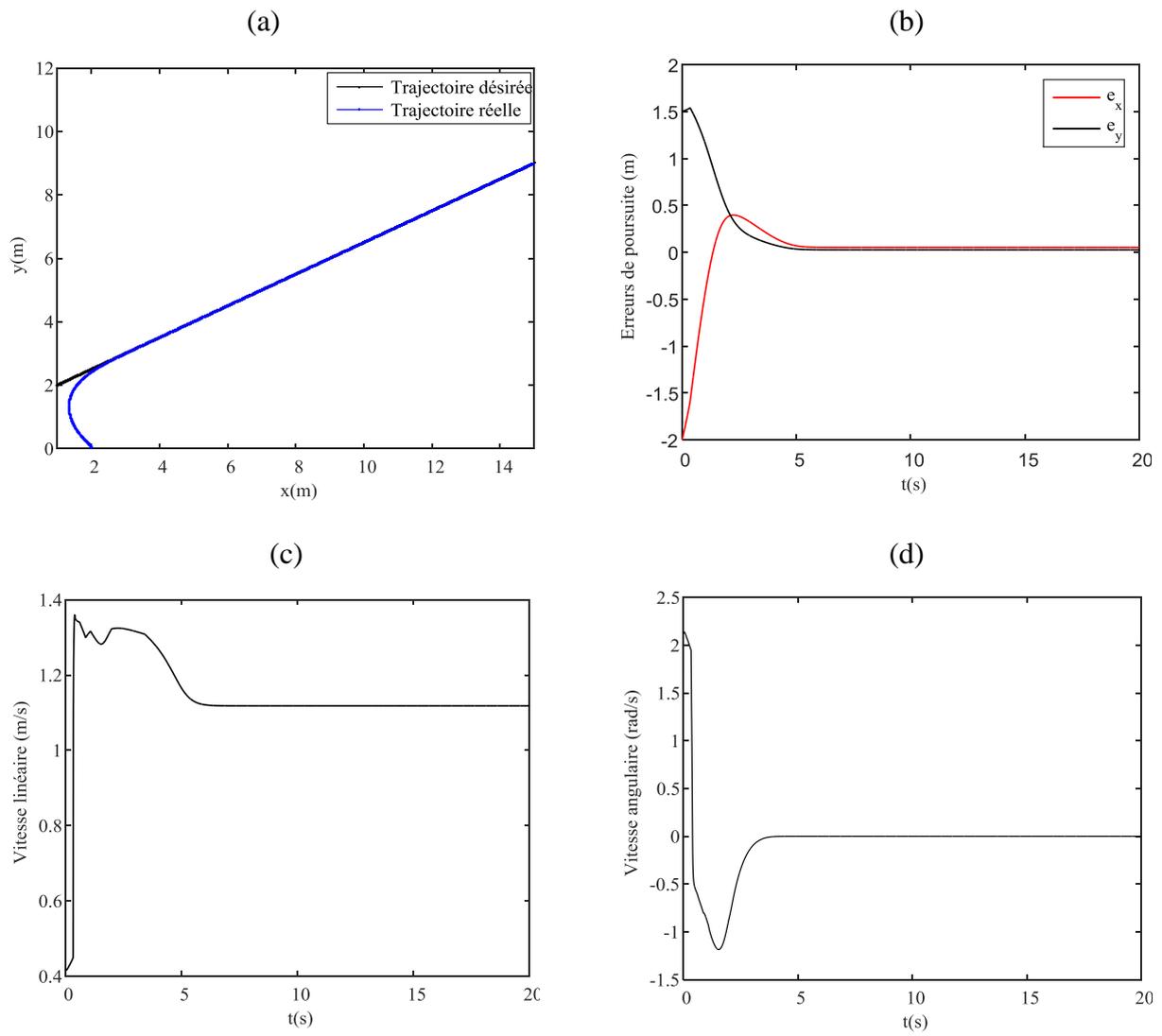
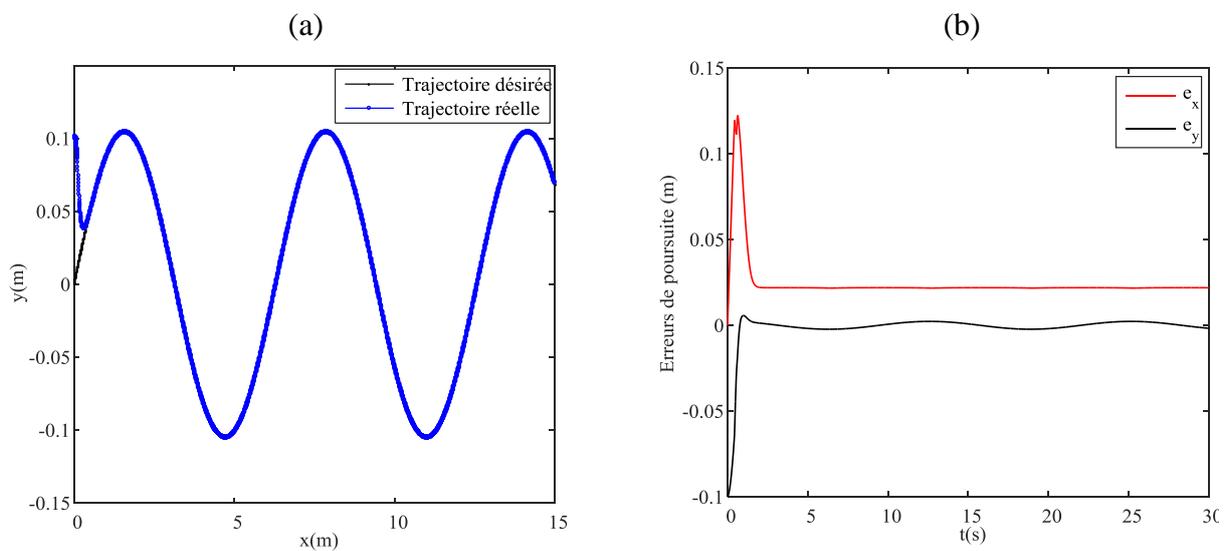


Figure I.26. (a) Poursuite de trajectoire, (b) erreurs de poursuite, (c) vitesse linéaire et (d) vitesse angulaire



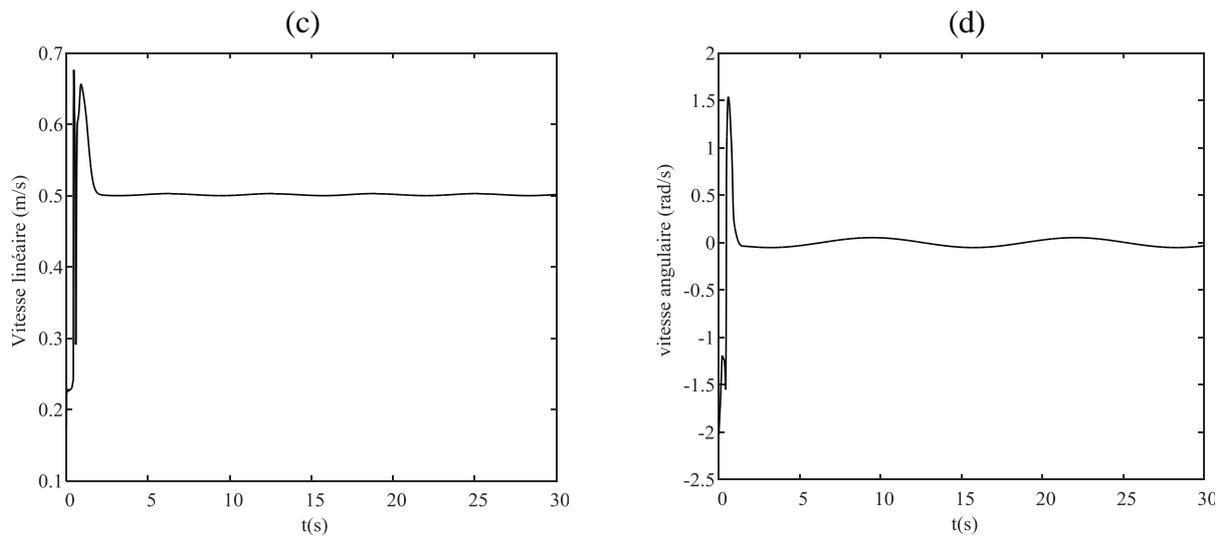


Figure I.27. (a) Poursuite de trajectoire, (b) erreurs de poursuite, (c) vitesse linéaire et (d) vitesse angulaire

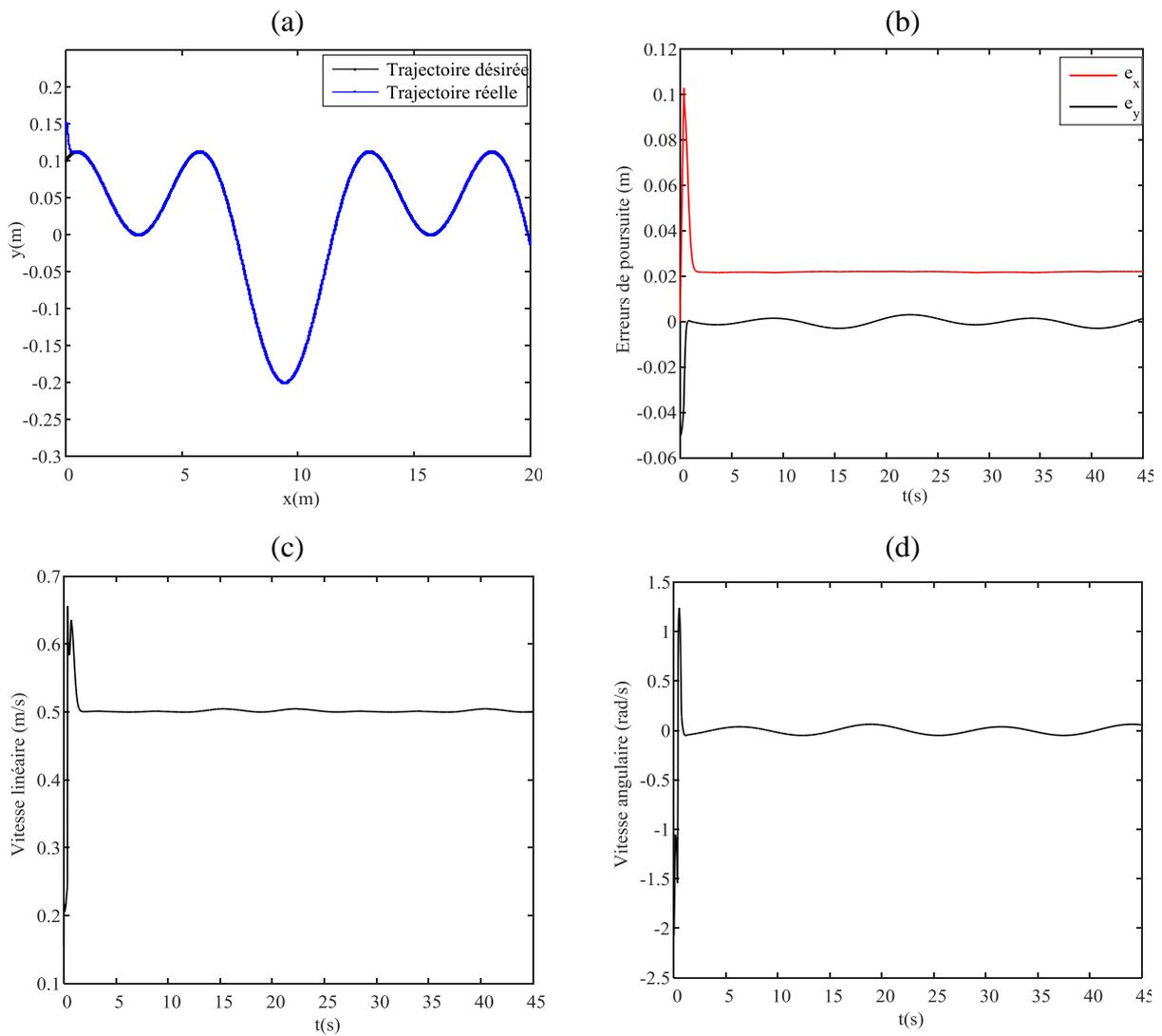


Figure I.28. (a) Poursuite de trajectoire, (b) erreurs de poursuite, (c) vitesse linéaire et (d) vitesse angulaire

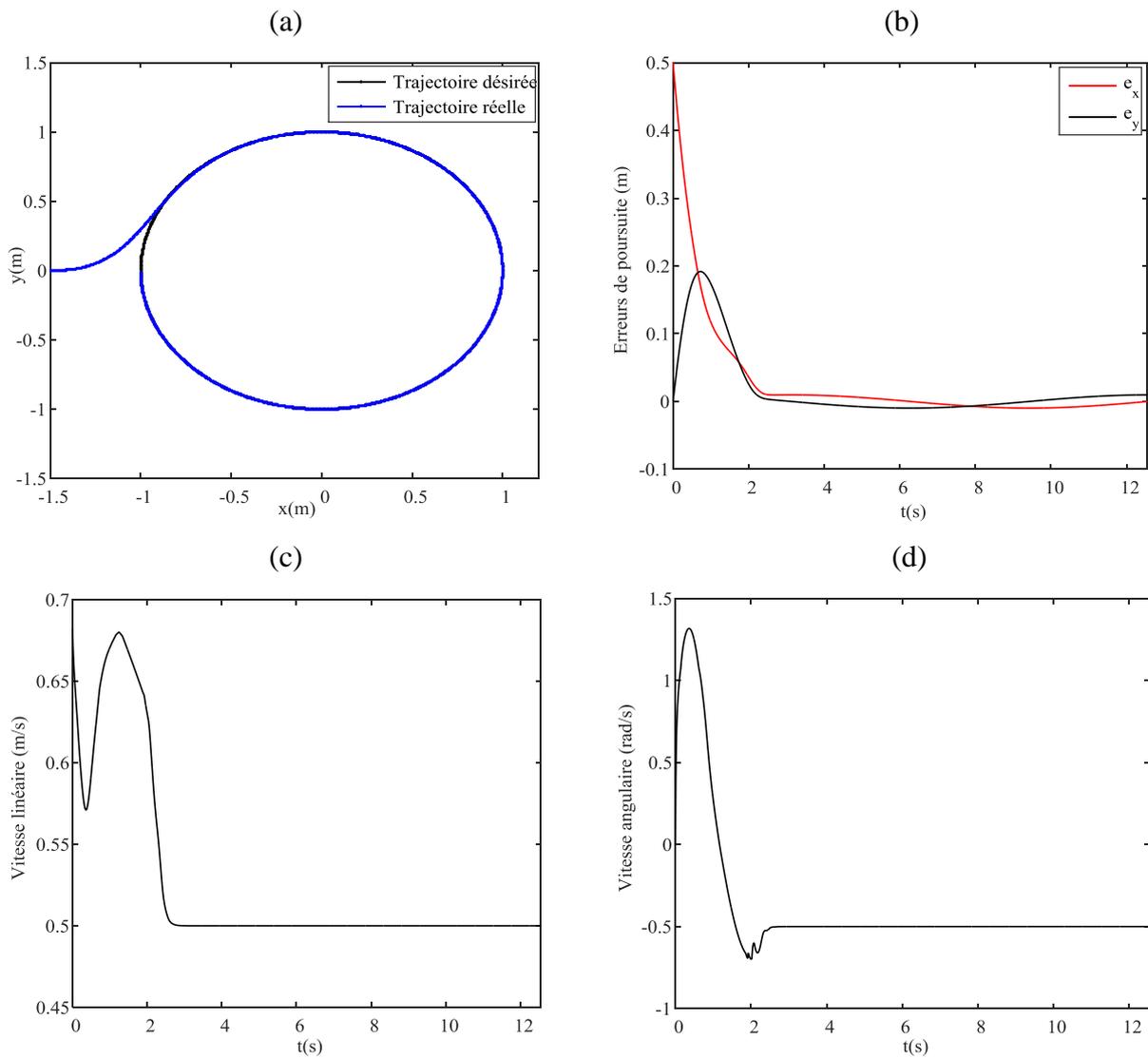


Figure I.29. (a) Poursuite de trajectoire, (b) erreurs de poursuite, (c) vitesse linéaire et (d) vitesse angulaire

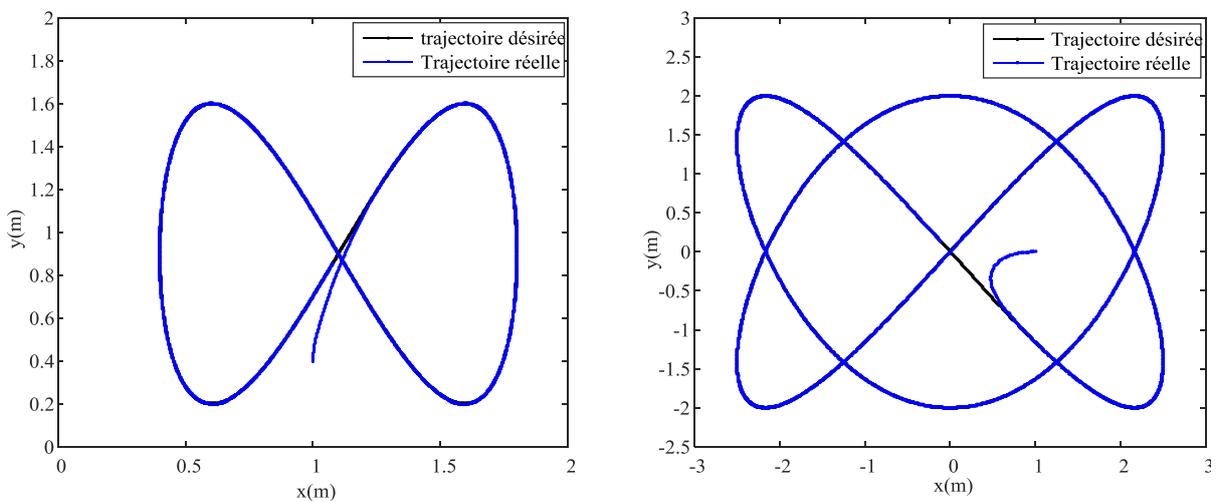


Figure I.30. Poursuite des trajectoires lemniscates

I.8. Etude comparative

Pour situer les performances de la stratégie de commande en poursuite proposée, nous allons comparer les résultats obtenus avec ceux de la référence [Kla-07] en termes de trajectoire poursuivie et de vitesses de déplacement.

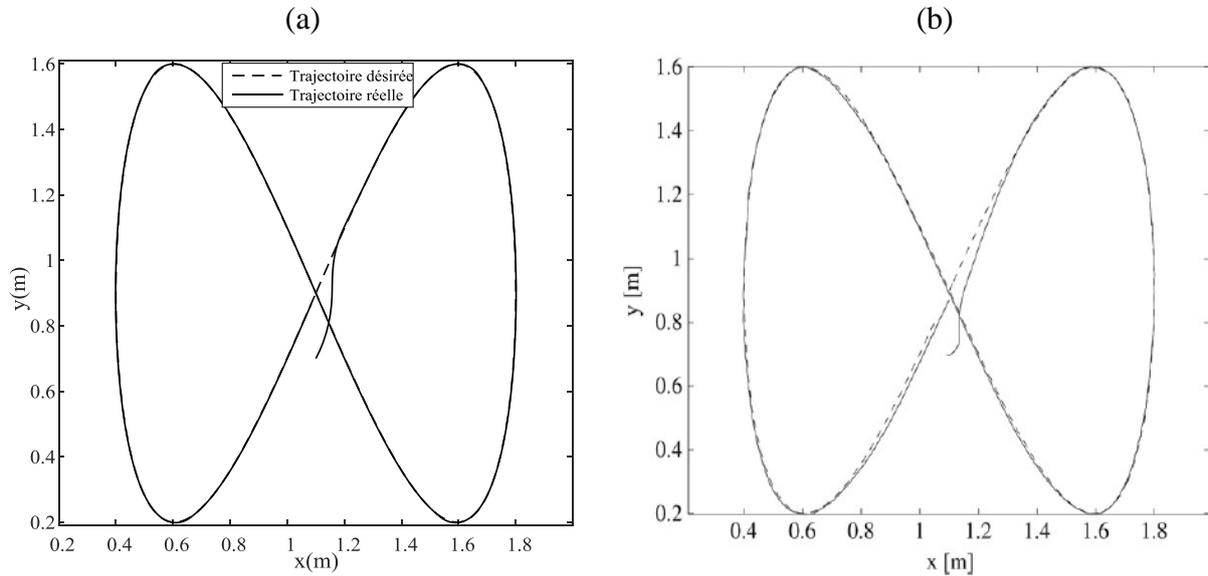


Figure I.31. Poursuite de trajectoire: (a) résultats obtenus, (b) résultats de [Kla-07]

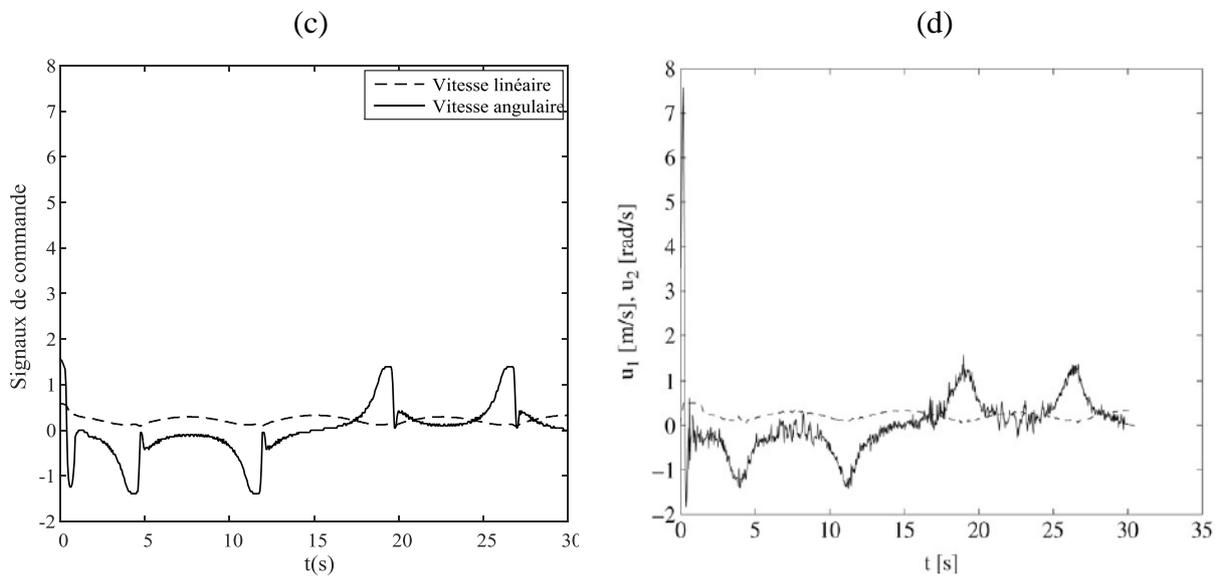


Figure I.32. Signaux de commande: (a) résultats obtenus, (b) résultats de [Kla-07]

En comparant les résultats obtenus par la structure de commande floue développée avec ceux de la référence [Kla-07], nous constatons une convergence plus rapide, une stabilité le

long de la trajectoire de référence (un régime permanent amélioré) (figure I.31) et des signaux de commande plus lisses faciles à implémenter (figure I.32).

I.9. Conclusion

Ce chapitre a été dédié à la planification et à la poursuite de trajectoires d'un robot mobile de type unicycle. Le processus de la planification de trajectoires a été effectué via les approches de la fenêtre dynamique traditionnelle et sa version améliorée que nous avons proposée. Dans la version développée, nous avons utilisé un contrôleur flou pour adapter les facteurs de pondération de la fonction "objectif" aux différentes situations de l'environnement. Le suivi de trajectoires a été réalisé en exploitant deux contrôleurs flous, le premier permet de poursuivre des trajectoires données et planifiées, et le second garantit une meilleure orientation du robot mobile.

Les résultats de simulation obtenus, sous l'environnement Matlab, montrent, d'une part, que les performances de la fenêtre dynamique floue sont remarquablement supérieures à celles de la fenêtre dynamique classique durant la planification de trajectoires, surtout dans les environnements exigus avec beaucoup d'obstacles, d'autre part, ils mettent en évidence l'efficacité des contrôleurs flous synthétisés lors du suivi de trajectoires avec une convergence rapide vers les trajectoires désirées et des signaux de commande faciles à implémenter.

L'étude comparative faite entre les résultats obtenus et ceux de la référence [Kla-07], a confirmé que la stratégie de commande en poursuite développée est plus performante en termes de vitesse de convergence et de stabilité sur la trajectoire de référence avec des signaux de commande plus lisses faciles à implémenter.

CHAPITRE II

POURSUITE DE TRAJECTOIRES ET ÉVITEMENT D'OBSTACLES PAR UNE COMMANDE OPTIMALE FLOUE

POURSUITE DE TRAJECTOIRES ET ÉVITEMENT D'OBSTACLES PAR UNE COMMANDE OPTIMALE FLOUE

II.1. Introduction

Le problème de poursuite de trajectoires et le problème d'évitement d'obstacles ont été largement étudiés dans différents travaux et plusieurs solutions ont été proposées. La plupart des chercheurs ont traité ces deux problèmes séparément et très peu d'entre eux ont inclus l'idée de poursuite de trajectoire en présence d'obstacles. Dans [Kan-10], les auteures ont développé une stratégie de contrôle qui combine la commande prédictive avec la méthode waypoint pour assurer une poursuite de trajectoire sans collision. Dans [Yoo-11], un contrôleur adaptatif basé sur la technique backstepping a été conçu pour garder le robot loin des obstacles tout en continuant à suivre le chemin désiré. Les inconvénients de ces techniques sont la complexité du processus de conception et la difficulté de mise en œuvre.

Parmi les contrôleurs de poursuite de trajectoire qui ont réussi à fournir de bonnes performances, ceux qui ont une méthodologie de conception basée sur le modèle cinématique de l'erreur qui est linéarisé autour de la trajectoire de référence [Kla-07] et représenté par des systèmes flous de Takagi Sugeno (TS) [Kim-09], [Wha-10].

La modélisation par les systèmes flous TS [Tak-85] est l'approche multimodèles la plus utilisée pour représenter un système non linéaire sous forme d'un ensemble de modèles linéaires locaux qui expriment le comportement du système autour de différents points de fonctionnement, et qui sont interconnectés à l'aide des fonctions d'appartenance non linéaires choisies d'une manière judicieuse. Les lois de commande développées pour les systèmes utilisant ce type de modélisation sont basées sur le concept de l'approche PDC (compensation parallèle distribuée) [Lin-01], [Tse-01] qui a pour but d'assurer la stabilité de chaque sous

système et puis selon la synthèse de Lyapunov, des conditions de stabilité du système global seront obtenues. Un certain degré de performance est réalisable si pour chaque sous système, un critère de performance quadratique sera minimisé, ce qui est garanti par le contrôleur LQR (régulateur linéaire quadratique).

Dans ce chapitre, Nous allons concevoir un contrôleur de poursuite de trajectoires en utilisant un modèle cinématique linéaire de l'erreur obtenu par la linéarisation de l'équation dynamique de l'erreur autour de la trajectoire désirée. Le modèle cinématique de l'erreur sera représenté par des systèmes flous TS pour employer la méthode PDC où chaque système local sera stabilisé par un régulateur LQR. En présence d'obstacles, nous allons appliquer un régulateur LQR au système flou global en boucle ouverte afin d'éviter les collisions. Pour donner au régulateur LQR la faculté d'évitement d'obstacles, ses matrices de pondération seront ajustées dynamiquement à l'aide d'un contrôleur flou. Pour que le robot mobile évite les obstacles sans perdre sa capacité de poursuite de trajectoire, un contrôleur flou de fusion sera conçu pour combiner les vitesses de poursuite et d'évitement d'obstacles. Pour valider la stratégie de contrôle proposée et prouver son efficacité, un exemple de simulation sous l'environnement Matlab sera considéré.

II.2. Modélisation de Takagi Sugeno

La modélisation TS est une approche multimodèles proposée par Takagi et Sugeno en 1985 [Tak-85] pour représenter un système non linéaire par des relations dynamiques locales en utilisant un ensemble de règles floues de type «Si-Alors ». Les antécédents de ces règles floues peuvent être considérés comme des conditions de validité du modèle et les conséquents expriment le comportement associé aux parties prémisses par des modèles locaux invariants dans le temps.

Si $x_1(t)$ est M_{i1} etet $x_n(t)$ est M_{in} alors $\dot{x} = A_i x + B_i u$ (II.1)

où $x \in \mathfrak{R}^n$ représente le vecteur d'état. M_{ij} , $j = 1, \dots, n$ est le $j^{\text{ème}}$ ensemble flou de la $i^{\text{ème}}$ règle, $A_i \in \mathfrak{R}^{n \times n}$, $B_i \in \mathfrak{R}^{n \times m}$ sont la matrice d'état et la matrice d'entrée du système respectivement, $u \in \mathfrak{R}^m$ est le vecteur des commandes.

Un poids noté $w_i(x)$ est assigné à chaque règle floue. Il dépend du degré d'appartenance des variables de prémisses $x_j(t)$ aux sous-ensembles flous $\mu_{ij}(x_j(t))$.

$$w_i(x) = \prod_{j=1}^n \mu_{ij}(x_j) \quad (\text{II.2})$$

où $\mu_{ij}(x_j)$ est la valeur de la fonction d'appartenance pour la variable de prémisse x_j dans la $j^{\text{ème}}$ ensemble flou dans la $i^{\text{ème}}$ règle.

Puisque $0 \leq \mu_{ij}(x_j(t)) \leq 1$, les propriétés suivantes peuvent être établies:

$$\begin{cases} w_i(x(t)) \geq 0, \forall t \\ \sum_{i=1}^r w_i(x(t)) > 0 \end{cases} \quad (\text{II.3})$$

où r est le nombre des règles floue.

Soit α_i le coefficient représentant la fonction d'activation de la $i^{\text{ème}}$ règle du modèle flou défini par:

$$\alpha_i = \frac{w_i}{\sum_{i=1}^r w_i} \quad (\text{II.4})$$

Pour $i=1, \dots, r$, ces fonctions satisfont la caractéristique d'une somme convexe:

$$\begin{cases} \alpha_i(x(t)) \geq 0 \\ \sum_{i=1}^r \alpha_i(x(t)) = 1 \end{cases} \quad (\text{II.5})$$

La fuzzification par singleton, la défuzzification par le centre de gravité, et le produit d'inférence permettent d'établir le modèle flou global comme suit:

$$\dot{x} = \frac{\sum_{i=1}^r w_i(x)(A_i x + B_i u)}{\sum_{i=1}^r w_i(x)} \quad (\text{II.6})$$

Par substitution de (II.4) dans (II.6), le modèle flou global sera écrit sous la forme suivante:

$$\dot{x} = \sum_{i=1}^r \alpha_i(x)(A_i x + B_i u) \quad (\text{II.7})$$

II.3. Obtention des modèles flous TS

Principalement, il existe trois méthodes pour obtenir un modèle flou TS d'un système non linéaire affine en la commande. La première s'inspire des techniques d'identification, elle

consiste à déterminer les paramètres des sous systèmes à travers les mesures accordées sur les entrées et les sorties du système [Gas-99], [Gas-00]. La seconde est basée sur la linéarisation du système non linéaire autour de différents points de fonctionnement [Tan-01]. La troisième est connue par l'approche des secteurs non linéaires [Tan-01], elle permet d'exprimer le comportement du système non linéaire de manière exacte dans un espace compact des variables d'état. Dans ce travail, nous nous repons sur la méthode de linéarisation.

II.3.1. Linéarisation

L'application de la technique de linéarisation nécessite la disponibilité d'un modèle mathématique non linéaire du système qui doit être linéarisé autour de différents points de fonctionnement.

II.3.2. Linéarisation d'une fonction

Soit $f : \mathbb{R}^n \rightarrow \mathbb{R}^k$ une fonction différentiable. Le développement de Taylor de f à l'ordre 1 au voisinage du point $\bar{x} \in \mathbb{R}^n$ est donné par:

$$f(x) \simeq f(\bar{x}) + \frac{df}{dx}(\bar{x})(x - \bar{x}) \quad (\text{II.8})$$

où

$$\frac{df}{dx}(\bar{x}) = \begin{pmatrix} \frac{\partial f_1}{\partial x_1}(\bar{x}) & \frac{\partial f_1}{\partial x_2}(\bar{x}) & \dots & \frac{\partial f_1}{\partial x_n}(\bar{x}) \\ \frac{\partial f_2}{\partial x_1}(\bar{x}) & \frac{\partial f_2}{\partial x_2}(\bar{x}) & \dots & \frac{\partial f_2}{\partial x_n}(\bar{x}) \\ \vdots & \vdots & \vdots & \vdots \\ \frac{\partial f_k}{\partial x_1}(\bar{x}) & \frac{\partial f_k}{\partial x_2}(\bar{x}) & \dots & \frac{\partial f_k}{\partial x_n}(\bar{x}) \end{pmatrix} \quad (\text{II.9})$$

La matrice (II.9) est appelée matrice Jacobienne.

Si nous avons par exemple $f \begin{pmatrix} x_1 \\ x_2 \end{pmatrix} = \begin{pmatrix} x_1^2 x_2 \\ x_1^2 + x_1^2 \end{pmatrix}$, sa linéarisation autour du point $\bar{x} = (1, 2)$

sera comme suit :

- Le calcul de la matrice Jacobienne:

$$\frac{df}{dx}(x) = \begin{pmatrix} \frac{\partial f_1}{\partial x_1}(x) & \frac{\partial f_1}{\partial x_2}(x) \\ \frac{\partial f_2}{\partial x_1}(x) & \frac{\partial f_2}{\partial x_2}(x) \end{pmatrix} = \begin{pmatrix} 2x_1x_2 & x_1^2 \\ 2x_1 & 2x_2 \end{pmatrix}$$

- L'application de la relation (II.8) au voisinage du point $\bar{x} = (1, 2)$, donne:

$$f \begin{pmatrix} x_1 \\ x_2 \end{pmatrix} \approx \begin{pmatrix} 2 \\ 5 \end{pmatrix} + \begin{pmatrix} 4 & 1 \\ 2 & 4 \end{pmatrix} \begin{pmatrix} x_1 - 1 \\ x_2 - 2 \end{pmatrix} = \begin{pmatrix} -4 + 4x_1 + x_2 \\ -5 + 2x_1 + 4x_2 \end{pmatrix}$$

II.3.3. Linéarisation d'un système

Considérons le système non linéaire décrit par l'équation suivante:

$$\begin{cases} \dot{x} = f(x, u) \\ y = g(x, u) \end{cases} \quad (\text{II.10})$$

où f et g sont des fonctions non linéaires différentiables, x , u et y sont des vecteurs de dimensions n , m et k , respectivement.

Soit $z = \begin{pmatrix} x \\ u \end{pmatrix}$ et $h(x, u) = \begin{pmatrix} f(x, u) \\ g(x, u) \end{pmatrix}$. D'après l'équation (II.8), la linéarisation de $h(x, u)$

autour du point $\bar{z} = (\bar{x}, \bar{u})$ est donné par:

$$h(z) \approx h(\bar{z}) + \frac{dh}{dz}(\bar{z})(z - \bar{z})$$

Dans ce cas, $\frac{dh}{dz}(\bar{z})$ est une matrice par bloc de la forme suivante:

$$\frac{dh}{dz}(\bar{z}) = \begin{pmatrix} A & B \\ C & D \end{pmatrix} \quad (\text{II.11})$$

avec $A = \frac{df}{dx}(\bar{x}, \bar{u})$, $B = \frac{df}{du}(\bar{x}, \bar{u})$, $C = \frac{dg}{dx}(\bar{x}, \bar{u})$ et $D = \frac{dg}{du}(\bar{x}, \bar{u})$.

Le système linéaire résultant est décrit par:

$$\begin{cases} \dot{x} = f(\bar{x}, \bar{u}) + A(x - \bar{x}) + B(u - \bar{u}) \\ y = g(\bar{x}, \bar{u}) + C(x - \bar{x}) + D(u - \bar{u}) \end{cases} \quad (\text{II.12})$$

II.3.4. Linéarisation d'un système autour d'un point de fonctionnement

Un point (\bar{x}, \bar{u}) est appelé point de fonctionnement si $f(\bar{x}, \bar{u}) = 0$, si $\bar{u} = 0$, le point devient un point d'équilibre. À partir de l'équation (II.12), l'approximation linéaire du système (II.10) sera comme suit:

$$\begin{cases} \dot{x} = A(x - \bar{x}) + B(u - \bar{u}) \\ y = \bar{y} + C(x - \bar{x}) + D(u - \bar{u}) \end{cases}$$

et pour des petites variations, le système linéaire autour du point de fonctionnement (\bar{x}, \bar{u}) est réécrit sous la forme suivante:

$$\begin{cases} \dot{x} = Ax + Bu \\ y = Cx + Du \end{cases} \quad (\text{II.13})$$

II.4. Stabilité des modèles flous TS

La théorie de Lyapunov est l'outil fondamental pour l'étude de la stabilité. La stabilité quadratique des systèmes flous TS consiste à trouver une matrice P définie positive de sorte que la dérivée de la fonction de Lyapunov soit négative.

$$V(x(t)) = x^T(t)Px(t) \quad (\text{II.14})$$

Suivant la seconde méthode de Lyapunov, les conditions quadratiques suffisantes de la stabilité des modèles flous TS sont données par le théorème 1.

Théorème 1

Le modèle flou TS (II.7) est asymptotiquement stable s'il existe une matrice commune $P > 0$ telle que :

$$A_i^T P + PA_i < 0 \quad i = 1, \dots, r \quad (\text{II.15})$$

II.4.1. Stabilisation des modèles flous TS par retour d'état

Pour assurer la stabilité d'un modèle flou TS en boucle fermée, la synthèse d'une loi de commande appropriée doit être réalisée. Généralement la conception des contrôleurs flous pour les modèles TS est basée sur la théorie de Lyapunov où les conditions de synthèse sont obtenues à l'aide des conditions d'analyse de la stabilité (théorème 1) en remplaçant les matrices d'état du système en boucle ouverte par celles du système en boucle fermée. Parmi les contrôleurs flous les plus utilisés avec succès, ceux qui s'appuient sur une structure PDC.

II.4.2. Stabilisation des modèles flous TS par un contrôleur PDC

Le principe de l'approche PDC consiste à concevoir un compensateur pour chaque modèle local du système sous forme d'un contrôleur linéaire par retour d'état afin de le stabiliser [Lin-01], [Tse-01]. Dans ce contexte la stabilité locale est garantie. Le régulateur PDC possède la même structure de synthèse que le modèle flou TS avec les mêmes ensembles flous, donc, il maintient les mêmes parties prémises ainsi que les mêmes fonctions d'appartenance. De manière similaire à la méthode utilisée pour interconnecter les sous modèles, la loi de commande globale est obtenue par l'interconnexion des lois de commande linéaires locales. La figure II.1 présente le concept du régulateur PDC.

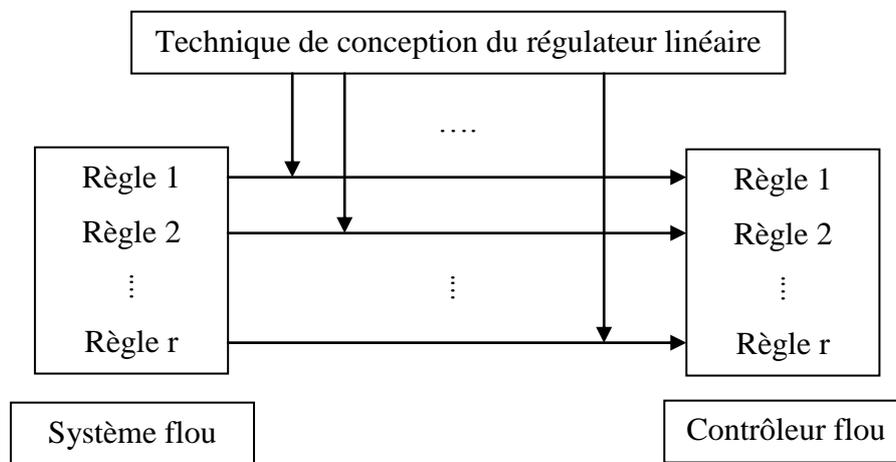


Figure II.1. Représentation du concept de l'approche PDC

La $i^{\text{ème}}$ règle du contrôleur flou est exprimée de la façon suivante:

$$\text{Si } x_1(t) \text{ est } M_{i1} \text{ et } \dots \text{ et } x_n(t) \text{ est } M_{in} \text{ alors } u = -K_i x \quad (\text{II.16})$$

Le contrôleur flou global sera défini par:

$$u = -\sum_{i=1}^r \alpha_i(x) K_i x \quad (\text{II.17})$$

où $K_i \in \mathbb{R}^{m \times n}$ représentent le vecteur des gains du contrôleur par retour d'état.

Par substitution de (II.17) dans (II.7), nous aurons le modèle globale du système en boucle fermée:

$$\dot{x} = \sum_{i=1}^r \sum_{j=1}^r \alpha_i(x) \alpha_j(x) (A_i - B_i K_j) x \quad (\text{II.18})$$

Les conditions de stabilité du système globale en boucle fermée, sont données par Le théorème suivant:

Théorème 2

Le système flou en boucle fermée (II.18) est globalement, asymptotiquement stable s'il existe une matrice commune $P > 0$ qui satisfait les inégalités de Lyapunov suivantes:

$$\begin{cases} (A_i - B_i K_i)^T P + P(A_i - B_i K_i) < 0 & i = 1, \dots, r \\ G_{ij}^T P + P G_{ij} < 0 & i < j \leq r \end{cases} \quad (\text{II.19})$$

avec

$$G_{ij} = A_i - B_i K_j + A_j - B_j K_i \quad (\text{II.20})$$

Nous pouvons constater que le problème de conception d'un régulateur PDC peut être résumé dans la détermination des gains de retour d'état, satisfaisant les conditions du théorème 2.

Remarque 1: Il n'y a pas de méthode analytique pour trouver la matrice commune P , seules les méthodes numériques, comme les techniques d'optimisation convexes peuvent fournir une solution efficace, Matlab LMI Toolbox en est un exemple [Gah-95].

Pour résoudre les inégalités matricielles du théorème 2 à l'aide du logiciel Matlab, il suffit de pré-multiplier et post- multiplier les deux côtés des inégalités (II.19) par P^{-1} et de considérer les changements de variables suivants:

$$\begin{cases} Y = P^{-1} \\ X_i = K_i Y \end{cases} \quad (\text{II.21})$$

Les inégalités matricielles linéaires résultantes sont:

$$\begin{cases} Y A_i^T + A_i Y - B_i X_i - X_i^T B_i^T < 0 & i = 1, \dots, r \\ Y(A_i + A_j)^T + (A_i + A_j)Y & \\ -(B_i X_j + B_j X_i) - (B_i X_j + B_j X_i)^T < 0 & i < j \leq r \end{cases} \quad (\text{II.22})$$

La stabilité du système (II.18) sera garantie si LMIs (II.22) ont une solution commune définie positive.

Dans notre travail, nous considérons que la stabilité seule est insuffisante, et que le contrôleur doit en plus assurer un certain degré de performance. Pour ceci nous allons utiliser le régulateur LQR pour déterminer les gains de retour d'état locaux du système flou (II.18).

II.4.3. Régulateur LQR

La commande linéaire quadratique (LQ) est l'approche de contrôle moderne la plus utilisée pour élaborer une loi de commande par retour d'état [Lew-95]. Elle vise à formuler le problème de commande par retour d'état en termes d'optimisation d'un critère qui traduit un compromis entre l'effort de commande et les contraintes qu'on veut imposer à l'état. Le régulateur LQR est la version de la commande LQ où la borne supérieure de l'intervalle d'intégration de la fonction coût tend vers l'infini.

Considérons le système (II.13), avec x_0 l'état initial à l'instant t_0 , et soit $u(t)$ la commande optimale qui minimise le critère quadratique J .

$$J = \int_{t_0}^{\infty} [x^T(t)Qx + u^T(t)Ru(t)] dt \quad (\text{II.23})$$

où Q est une matrice semi-définie positive qui contrôle le taux de convergence de chaque composante du vecteur d'état, R est une matrice définie positive qui pénalise les entrées de commande.

Pour résoudre le problème de minimisation de J , nous définissons le Hamiltonien par la relation suivante:

$$H = \frac{1}{2} (x^T Q x + u^T R u) + \lambda^T (A x + B u) \quad (\text{II.24})$$

avec λ^T est choisit de sorte à avoir $\dot{\lambda} = -\frac{\partial H}{\partial x}$

Les conditions d'optimalité se présentent alors comme suit:

$$\checkmark \text{ Equation d'état: } \dot{x} = \frac{\partial H}{\partial \lambda} = A x + B u \quad (\text{II.25})$$

$$\checkmark \text{ Equation d'état adjointe: } -\dot{\lambda} = \frac{\partial H}{\partial x} = Q x + A^T \lambda \quad (\text{II.26})$$

$$\checkmark \text{ Vecteur de commande: } \frac{\partial H}{\partial u} = R u + B^T \lambda = 0 \quad (\text{II.27})$$

$$\Rightarrow u_{opt} = -R^{-1} B^T \lambda \quad (\text{II.28})$$

Par la substitution de (II.28) dans (II.25), nous obtenons:

$$\dot{x} = A x - B R^{-1} B^T \lambda \quad (\text{II.29})$$

Ensuite nous construisons le système Hamiltonien homogène en utilisant les équations (II.26) et (II.29):

$$\begin{bmatrix} \dot{x} \\ \dot{\lambda} \end{bmatrix} = \begin{bmatrix} A & -BR^{-1}B^T \\ -Q & -A^T \end{bmatrix} \begin{bmatrix} x \\ \lambda \end{bmatrix} \quad (\text{II.30})$$

À chaque instant t , nous pouvons écrire:

$$\lambda(t) = P(t)x(t) \quad (\text{II.31})$$

où, P est une matrice semi-définie positive à calculer.

En dérivant l'équation (II.31) et en utilisant l'équation d'état adjointe (II.26), nous aboutissons à l'expression suivante appelée équation de Riccati:

$$-\dot{P} = A^T P + PA - PBR^{-1}B^T P + Q \quad (\text{II.32})$$

En régime permanent, l'équation de Riccati (II.32) devient:

$$A^T P + PA - PBR^{-1}B^T P + Q = 0 \quad (\text{II.33})$$

À partir de l'équation de Riccati (II.33), nous calculons la matrice P , puis en utilisant cette dernière dans l'expression (II.31), nous serons capables de calculer le vecteur de commande en remplaçant (II.31) dans (II.28):

$$u_{opt} = -R^{-1}B^T P x(t) = -Kx(t) \quad (\text{II.34})$$

où K est la matrice des gains des commandes optimales.

$$K = R^{-1}B^T P \quad (\text{II.35})$$

II.5. Poursuite de trajectoire et évitement d'obstacles d'un robot mobile

II.5.1. Modélisation du robot mobile

Il s'agit d'un robot mobile de type unicycle avec deux roues motrices. Nous sommes juste intéressés par le modèle cinématique obtenu à partir de la modélisation d'une roue qui roule sans glissement sur le sol.

II.5.1.1. Rappel du modèle cinématique de l'unicycle

Le modèle cinématique du robot mobile présenté sur la figure II.2 est donné comme suit :

$$\begin{bmatrix} \dot{x} \\ \dot{y} \\ \dot{\theta} \end{bmatrix} = \begin{bmatrix} \cos \theta & 0 \\ \sin \theta & 0 \\ 0 & 1 \end{bmatrix} \begin{bmatrix} v \\ w \end{bmatrix} \quad (\text{II.36})$$

où x et y sont les variables de sortie qui représentent les coordonnées du robot, θ est la variable de sortie responsable de l'orientation du robot et v et w sont respectivement la vitesse linéaire et angulaire qui illustrent les entrées de commande.

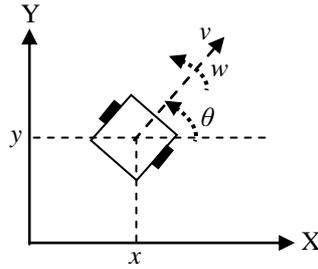


Figure II.2. Modèle du robot mobile

II.5.1.2. Modèle cinématique de l'erreur de poursuite de trajectoire

Supposons que la poursuite de trajectoire est formulée comme si le robot mobile réel suit un robot mobile imaginaire (virtuel) qui se déplace de façon exacte sur la trajectoire de référence avec un profil de vitesse spécifié. L'erreur de posture entre la position du robot réel et celle du robot virtuel (figure II.3) est définie par le modèle suivant [Kan-90]:

$$\begin{bmatrix} e_x \\ e_y \\ e_\theta \end{bmatrix} = \begin{bmatrix} \cos \theta & \sin \theta & 0 \\ -\sin \theta & \cos \theta & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x_r - x \\ y_r - y \\ \theta_r - \theta \end{bmatrix} \quad (\text{II.37})$$

où $e = [e_x \ e_y \ e_\theta]^T$ est le vecteur d'erreur de poursuite, x_r , y_r et θ_r sont les coordonnées du robot mobile virtuel.

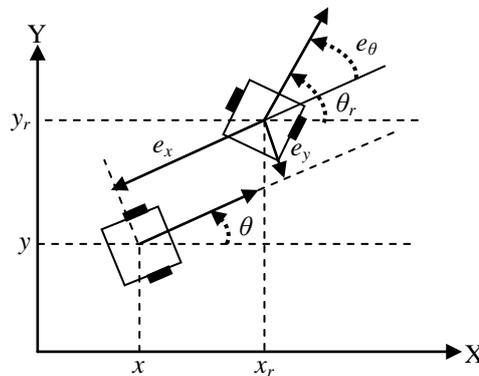


Figure II.3. Erreur de posture

Considérons que le robot imaginaire possède un modèle cinématique identique à celui de (II.36), en dérivant (II.37), nous obtenons le modèle de la dynamique de l'erreur de poursuite suivant:

$$\begin{bmatrix} \dot{e}_x \\ \dot{e}_y \\ \dot{e}_\theta \end{bmatrix} = \begin{bmatrix} \cos e_\theta & 0 \\ \sin e_\theta & 0 \\ 0 & 1 \end{bmatrix} \begin{bmatrix} v_r \\ w_r \end{bmatrix} + \begin{bmatrix} -1 & e_y \\ 0 & -e_x \\ 0 & -1 \end{bmatrix} u \quad (\text{II.38})$$

où v_r est la vitesse linéaire, w_r la vitesse angulaire du robot virtuel, et u le vecteur de commande défini comme étant la somme du vecteur de la commande anticipative et celui de la commande par rétroaction comme suit [Kla-07]:

$$u = u_F + u_B = \begin{bmatrix} v_r \cos e_\theta \\ w_r \end{bmatrix} + \begin{bmatrix} v \\ w \end{bmatrix} \quad (\text{II.39})$$

où u_F est l'action de commande anticipative et u_B la sortie du contrôleur proposé (elle sera définie plus tard).

En substituant (II.39) dans (II.38), le modèle résultant sera donné par:

$$\begin{bmatrix} \dot{e}_x \\ \dot{e}_y \\ \dot{e}_\theta \end{bmatrix} = \begin{bmatrix} 0 & w_r & 0 \\ -w_r & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix} \begin{bmatrix} e_x \\ e_y \\ e_\theta \end{bmatrix} + \begin{bmatrix} 0 \\ \sin e_\theta \\ 0 \end{bmatrix} \cdot v_r + \begin{bmatrix} -1 & 0 \\ 0 & 0 \\ 0 & -1 \end{bmatrix} \begin{bmatrix} v \\ w \end{bmatrix} \quad (\text{II.40})$$

En linéarisant le modèle non linéaire de la dynamique de l'erreur (II.40) autour de la trajectoire de référence ($e_x = e_y = e_\theta = 0$, $u_{B1} = u_{B2} = 0$), nous aurons le modèle cinématique linéaire de l'erreur de poursuite suivant:

$$\begin{bmatrix} \dot{e}_x \\ \dot{e}_y \\ \dot{e}_\theta \end{bmatrix} = \begin{bmatrix} 0 & w_r & 0 \\ -w_r & 0 & v_r \\ 0 & 0 & 0 \end{bmatrix} \begin{bmatrix} e_x \\ e_y \\ e_\theta \end{bmatrix} + \begin{bmatrix} -1 & 0 \\ 0 & 0 \\ 0 & -1 \end{bmatrix} \begin{bmatrix} v \\ w \end{bmatrix} \quad (\text{II.41})$$

Le modèle (II.41) est sous la forme d'une équation d'état linéaire, et il est contrôlable si v_r ou w_r est non nul. Différentes valeurs admissibles de v_r et w_r donnent plusieurs modèles cinématiques de l'erreur de poursuite qui seront utilisés comme des sous systèmes pour former le système flou TS.

II.5.2. Modèle flou TS de l'erreur de poursuite

En utilisant le modèle cinématique linéaire de l'erreur de poursuite (II.41) et les valeurs des vitesses de référence obtenues dans les intervalles: $[v_{r \min}, v_{r \max}]$ et $[w_{r \min}, w_{r \max}]$, la $i^{\text{ème}}$ règle du modèle flou est exprimée par:

$$\text{Si } v_r \text{ est } M_{i1} \text{ et } w_r \text{ est } M_{i2} \text{ alors } \dot{e} = A_i e + B_i u_B \quad (\text{II.42})$$

D'après (II.7), le modèle flou global est:

$$\dot{e} = \sum_{i=1}^r \alpha_i(e) (A_i e + B_i u_B) \quad (\text{II.43})$$

II.5.3. Loi de commande PDC en poursuite

De la même manière que dans (§ II.4.2.), le système flou (II.43) doit être stabilisé par une loi de commande floue de type PDC.

La $i^{\text{ème}}$ règle floue du contrôleur est donnée par:

$$\text{Si } v_r \text{ est } M_{i1} \text{ et } w_r \text{ est } M_{i2} \text{ alors } u_B = -K_i e \quad (\text{II.44})$$

À partir de (II.17), le contrôleur flou global est donné par:

$$u_B = -\sum_{i=1}^r \alpha_i(e) K_i e \quad (\text{II.45})$$

En substituant (II.45) dans (II.43), nous obtenons la dynamique globale de l'erreur de poursuite du système en boucle fermée:

$$\dot{e} = \sum_{i=1}^r \sum_{j=1}^r \alpha_i(e) \alpha_j(e) (A_i - B_i K_j) e \quad (\text{II.46})$$

Les conditions de convergence de (II.46) sont identiques à celles du système (II.18), sauf que la fonction de Lyapunov considérée est donnée par:

$$V(e(t)) = e^T(t) P e(t) \quad P > 0 \quad (\text{II.47})$$

Le vecteur des gains de commande pour chaque sous système est obtenu en utilisant le régulateur LQR où l'indice de performance à minimiser est le suivant:

$$J = \int_{t_0}^{\infty} \left[e^T Q e + u_B^T R u_B \right] dt \quad (\text{II.48})$$

II.5.4. Évitement d'obstacles par un régulateur LQR

Les matrices de pondération contrôlent le niveau de performance et l'optimalité du régulateur LQR, c'est-à-dire que la façon de sélection de ces matrices joue un rôle majeur dans la détermination de la qualité du contrôleur. Généralement les matrices de pondération sont choisies par tâtonnement (essais/erreurs).

Dans ce travail de thèse, la tâche d'évitement des obstacles sur une trajectoire de référence est confiée à un régulateur LQR dont les matrices de pondération sont ajustées en temps réel en utilisant un contrôleur flou avec deux entrées et cinq sorties [Akk-18b].

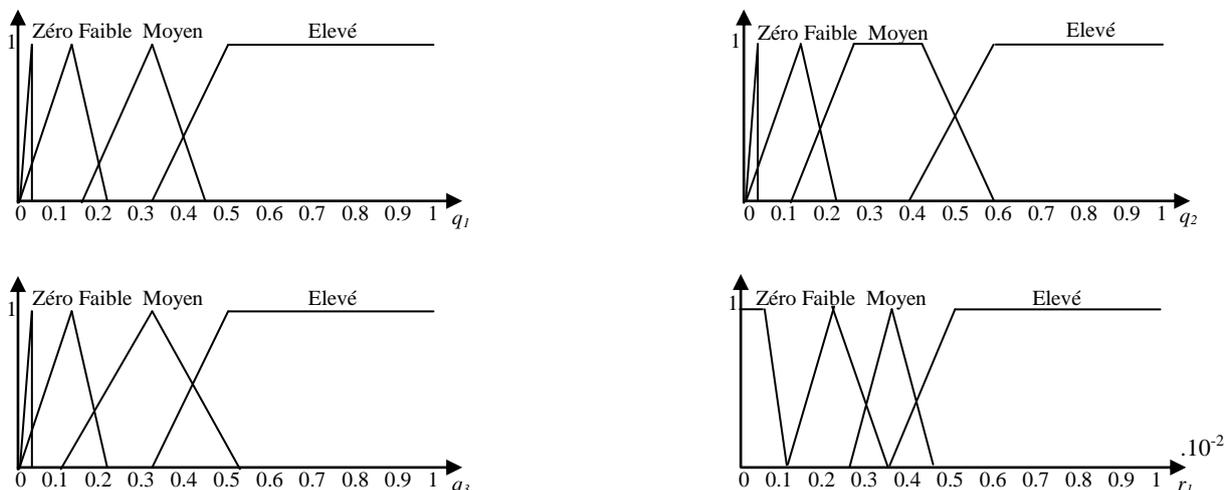
La première entrée est la distance actuelle ($D_{ob} = \sqrt{(x_{ob} - x)^2 + (y_{ob} - y)^2}$) entre le robot mobile et chaque obstacle rencontré sur la trajectoire désirée, et la deuxième entrée est la vitesse linéaire de poursuite (v_p). Les sorties sont les éléments de la diagonale de chaque matrice de pondération, $Q = \text{diag}(q_1, q_2, q_3)$ et $R = \text{diag}(r_1, r_2)$.

Les sous ensembles flous de D_{ob} sont: proche, moyenne et loin; et ils sont répartis sur l'univers $[0, 0.5\text{m}]$, v_p aussi possède trois sous ensembles flous (lente, normale et rapide) dans l'intervalle $[0, 2.5\text{m/s}]$. Les variables linguistiques zéro, faible, moyen et élevé constituent les sous ensembles flous de l'univers de discours de chaque élément de la diagonale de Q et R .

Les fonctions d'appartenance des variables d'entrée et de sortie sont représentées sur les figures II.4 et II.5 respectivement.



Figure II.4. Fonctions d'appartenance des entrées



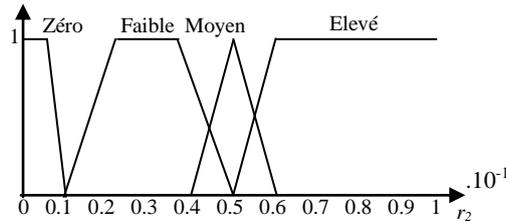


Figure II.5. Fonctions d'appartenance des sorties

Le tableau II.1 récapitule les règles floues utilisées pour déterminer les valeurs des matrices de pondération à travers des raisonnements logiques comme:

- Si un obstacle est proche, les éléments q_i doivent être élevés pour que le robot réagisse rapidement, mais il faut éviter les réactions agressives ou les manœuvres qui lui font perdre sa capacité de poursuite ce qui est la responsabilité des éléments r_i réglés en fonction de la vitesse linéaire de poursuite.
- Si le robot mobile est loin d'un obstacle ou il l'a dépassé, les éléments q_i et r_i doivent être zéros et élevés respectivement pour permettre au robot mobile de suivre la trajectoire de référence sans le comportement d'évitement d'obstacles.

Tableau II.1: Les règles floues pour définir les valeurs de Q et R

Numéro de règle	Entrées linguistiques		Sorties linguistiques				
	D_{ob}	v_p	q_1	q_2	q_3	r_1	r_2
1	Proche	Lente	Elevé	Elevé	Elevé	Zéro	Faible
2	Proche	Normale	Elevé	Elevé	Elevé	Faible	Moyen
3	Proche	Rapide	Elevé	Elevé	Elevé	Elevé	Zéro
4	Moyenne	Lente	Faible	Faible	Faible	Moyen	Moyen
5	Moyenne	Normale	Moyen	Moyen	Moyen	Moyen	Faible
6	Moyenne	Rapide	Elevé	Elevé	Elevé	Elevé	Zéro
7	Loin	Lente	Zéro	Zéro	Zéro	Elevé	Elevé
8	Loin	Normale	Zéro	Zéro	Zéro	Elevé	Elevé
9	Loin	Rapide	Zéro	Zéro	Zéro	Elevé	Elevé

II.5.5. Fusion floue des vitesses de poursuite et d'évitement d'obstacles

En présence d'obstacles sur la trajectoire désirée, le robot mobile doit les éviter tandis qu'il suit toujours le robot mobile virtuel, donc le robot mobile sera contrôlé par les vitesses de poursuite (v_p , w_p) et les vitesses d'évitement d'obstacles (v_{ob} , w_{ob}) simultanément. Ceci est accompli en utilisant un contrôleur flou, dont les entrées sont la distance actuelle (D_{ob}) et

l'angle ($\theta_{ob} = \text{atan2}((x_{ob} - x), (y_{ob} - y))$) entre le robot mobile et chaque obstacle rencontré sur la trajectoire de référence, et la sortie est le gain de fusion (F) [Akk-18b].

L'univers de discours de D_{ob} est constitué de deux sous ensembles flous: proche et loin qui occupent l'univers $[0, 0.5\text{m}]$. L'univers de θ_{ob} est $[-4, 4\text{rad}]$ où les trois sous ensembles flous: négative, zéro et positive sont répartis. F possède trois sous ensembles flous (peu, plus, et beaucoup) dans l'intervalle $[0, 1]$.

Les formes des fonctions d'appartenance sont illustrées sur les figures II.6 et II.7, et les règles floues utilisées pour définir les valeurs du gain de fusion sont résumées dans le tableau II.2.

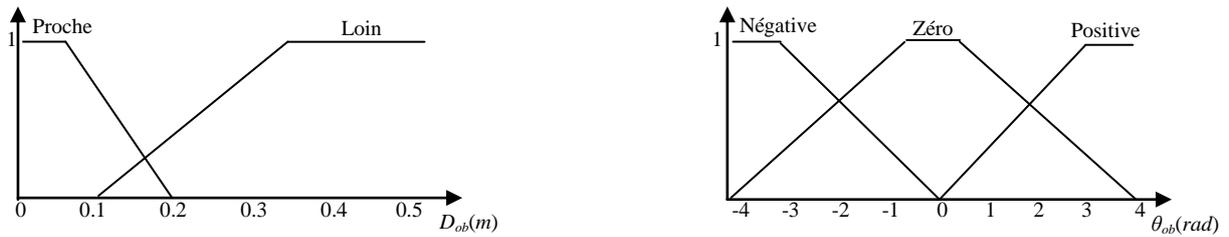


Figure II.6. Fonctions d'appartenance des variables d'entrée

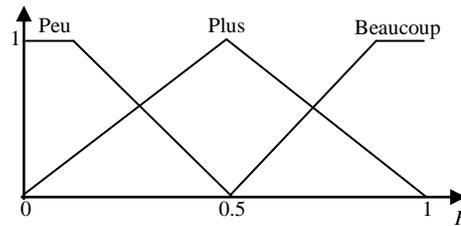


Figure II.7. Fonctions d'appartenance de la variable de sortie

Tableau II.2: Les règles floues pour déterminer les valeurs du gain F

Numéro de règle	Entrées linguistiques		Sorties linguistiques
	D_{ob}	θ_{ob}	F
1	Proche	Négative	Plus
2	Proche	Zéro	Beaucoup
3	Proche	Positive	Plus
4	Loin	Négative	Peu
5	Loin	Zéro	Peu
6	Loin	Positive	Peu

En fusionnant les vitesses de poursuite (v_p, w_p) avec celles d'évitement d'obstacles (v_{ob}, w_{ob}), l'entrée de commande du robot mobile sera obtenue comme suit:

$$\mathbf{u}_B = \begin{bmatrix} v \\ w \end{bmatrix} = \begin{bmatrix} v_p & v_{ob} \\ w_p & w_{ob} \end{bmatrix} \begin{bmatrix} 1-F \\ F \end{bmatrix} \quad (\text{II.49})$$

La stratégie de contrôle proposée est illustrée par la figure suivante:

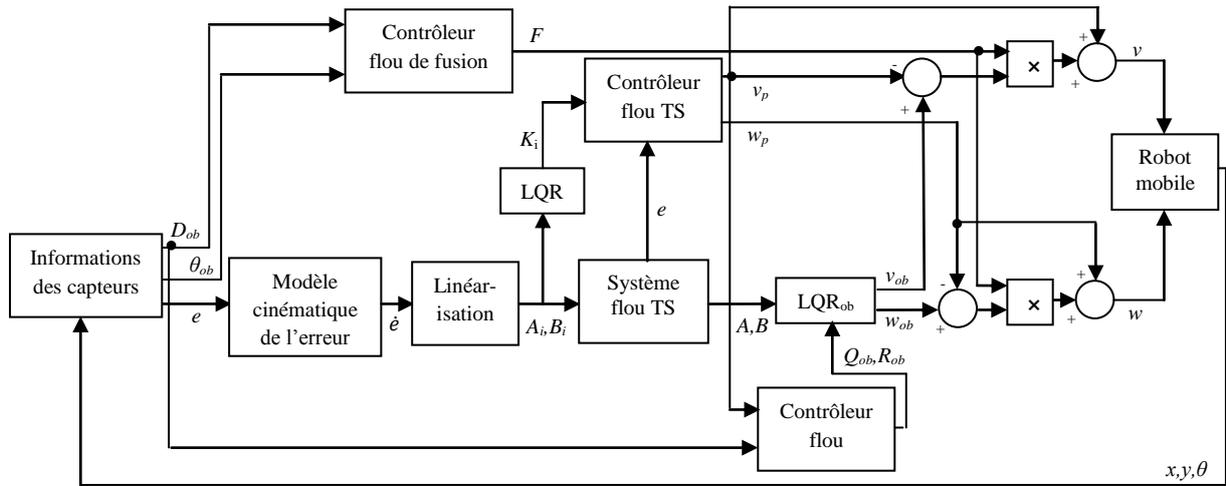


Figure II.8. Diagramme bloc de l'approche de contrôle proposée

II.6. Exemple de simulation

Pour évaluer la performance de la stratégie de contrôle proposée, des simulations sous l'environnement Matlab sont menées afin d'effectuer la poursuite de trajectoires en l'absence et en présence d'obstacles sur la trajectoire désirée.

Nous allons adopter le profil de la vitesse de référence linéaire et angulaire utilisé dans [Kim-09] et [Wha-10]:

$$v_r(t) = 1 + \exp(-2t) \text{ [m / s]}$$

$$w_r(t) = 10 \sin(0.01t) \text{ [rad / s]}$$

Les univers de discours des grandeurs v_r et w_r sont $[1, 6\text{m/s}]$ et $[-10, 10\text{rad/s}]$, respectivement. Les positions initiales sont $p(0) = [-1, -1, 0]^T$ et $p_r(0) = [0, 0, \pi/4]^T$.

Remarque 2: Si les vitesses de référence ne sont pas spécifiées par leurs équations temporelles, nous devons les calculer à partir des trajectoires de référence comme suit [Kla-07]:

$$v_r = \sqrt{\dot{x}_r^2 + \dot{y}_r^2} \text{ et } w_r = \frac{\dot{x}_r \ddot{y}_r - \dot{y}_r \ddot{x}_r}{\dot{x}_r^2 + \dot{y}_r^2}$$

Les règles floues sont données par:

Règle 1: Si v_r est petite et w_r est négative alors $\dot{e}(t) = A_1 e(t) + B_1 u_B(t)$

Règle 2: Si v_r est petite et w_r est positive alors $\dot{e}(t) = A_2 e(t) + B_2 u_B(t)$

Règle 3: Si v_r est grande et w_r est négative alors $\dot{e}(t) = A_3 e(t) + B_3 u_B(t)$

Règle 4: Si v_r est grande et w_r est positive alors $\dot{e}(t) = A_4 e(t) + B_4 u_B(t)$

où $A_1 = \begin{bmatrix} 0 & -10 & 0 \\ 10 & 0 & 1 \\ 0 & 0 & 0 \end{bmatrix}$, $A_2 = \begin{bmatrix} 0 & 10 & 0 \\ -10 & 0 & 1 \\ 0 & 0 & 0 \end{bmatrix}$, $A_3 = \begin{bmatrix} 0 & -10 & 0 \\ 10 & 0 & 6 \\ 0 & 0 & 0 \end{bmatrix}$, $A_4 = \begin{bmatrix} 0 & 10 & 0 \\ -10 & 0 & 6 \\ 0 & 0 & 0 \end{bmatrix}$ et

$$B_1 = B_2 = B_3 = B_4 = \begin{bmatrix} -1 & 0 \\ 0 & 0 \\ 0 & -1 \end{bmatrix}.$$

Les fonctions d'appartenance utilisées dans le modèle flou TS et le contrôleur flou sont représentées sur la figure II.9.

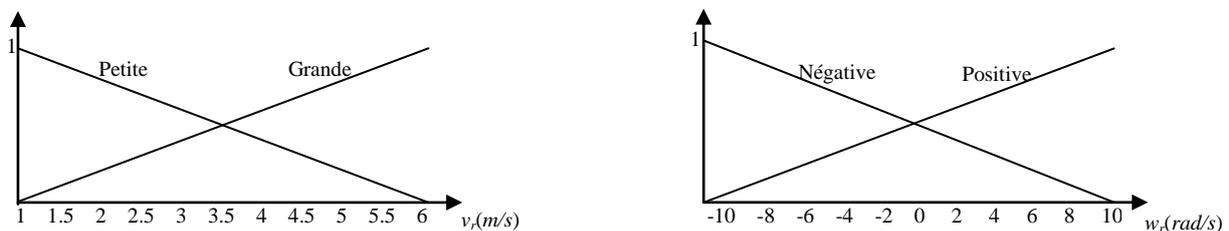


Figure II.9. Fonctions d'appartenance des entrées

II.6.1. Résultats de simulation de poursuite

La figure II.10a montre la poursuite de trajectoire en l'absence d'obstacles sur la trajectoire de référence. La figure II.10b illustre les variations en fonction du temps des erreurs de poursuite de trajectoire.

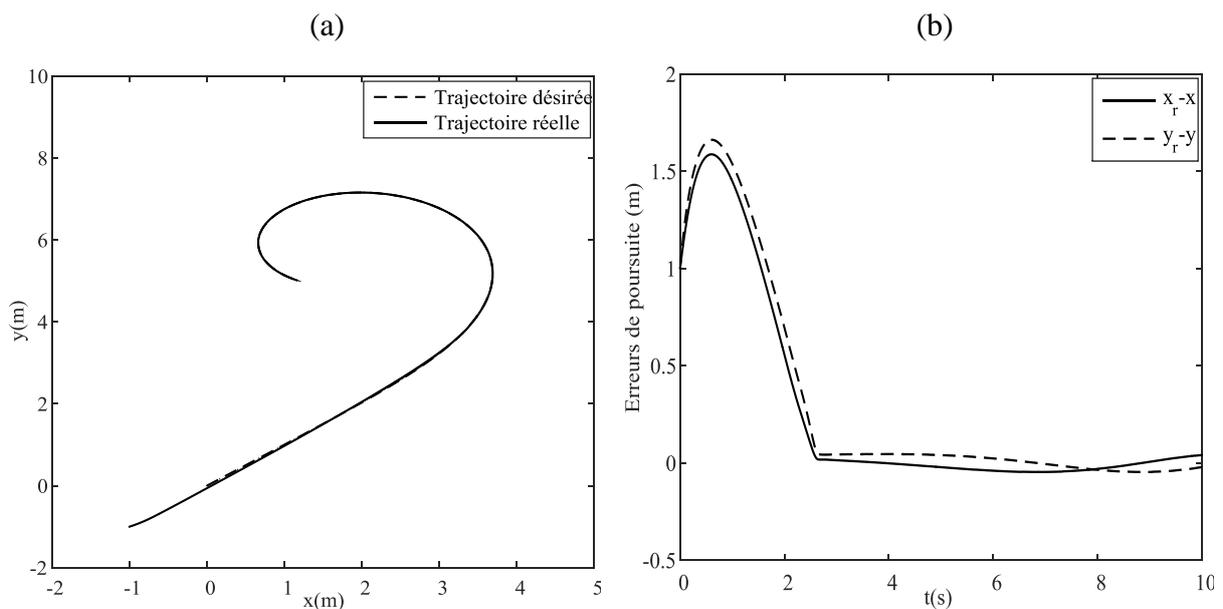


Figure II.10. (a) Poursuite de trajectoire et (b) erreurs de poursuite

II.6.1.1. Etude comparative

Pour pouvoir situer les performances du contrôleur PDC en poursuite établi, nous allons comparer nos résultats avec ceux des références [Kim-09] et [Wha-10] en terme de trajectoire poursuivie.

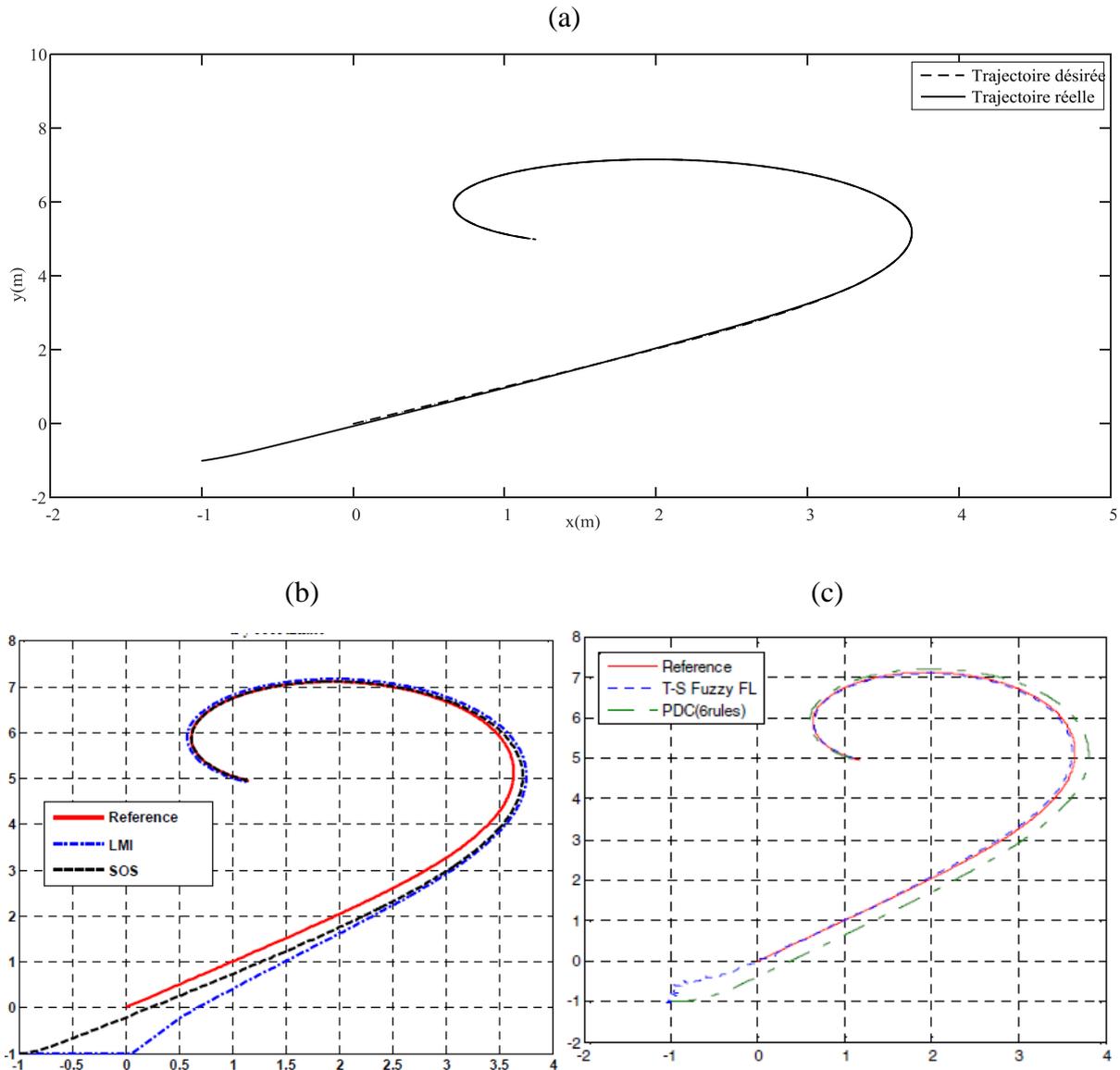


Figure II.11. Poursuite de trajectoire: (a) résultat obtenu, (b) résultat de [Kim-09], (c) résultat de [Wha-10]

En comparant les résultats obtenus (figure II.11a) par la structure de commande PDC en poursuite développée, avec ceux des références [Kim-09] et [Wha-10] (figure II.11b et II.11c), nous remarquons une convergence plus rapide avec des erreurs de poursuite faibles.

II.6.2. Résultats de simulation de poursuite avec évitement d'obstacles

La figure II.12 illustre la poursuite de trajectoire en présence d'obstacles sur la trajectoire désirée. La figure II.13 présente les erreurs de poursuite de trajectoire. La figure II.14 montre l'allure des signaux de commande. Dans la figure II.15, nous pouvons voir comment les éléments de la diagonale de Q et R varient lors du processus d'évitement d'obstacles.

Les résultats obtenus prouvent que la stratégie de contrôle proposée est capable de réaliser la tâche de poursuite de trajectoire et la tâche d'évitement d'obstacles simultanément avec de bonnes performances.

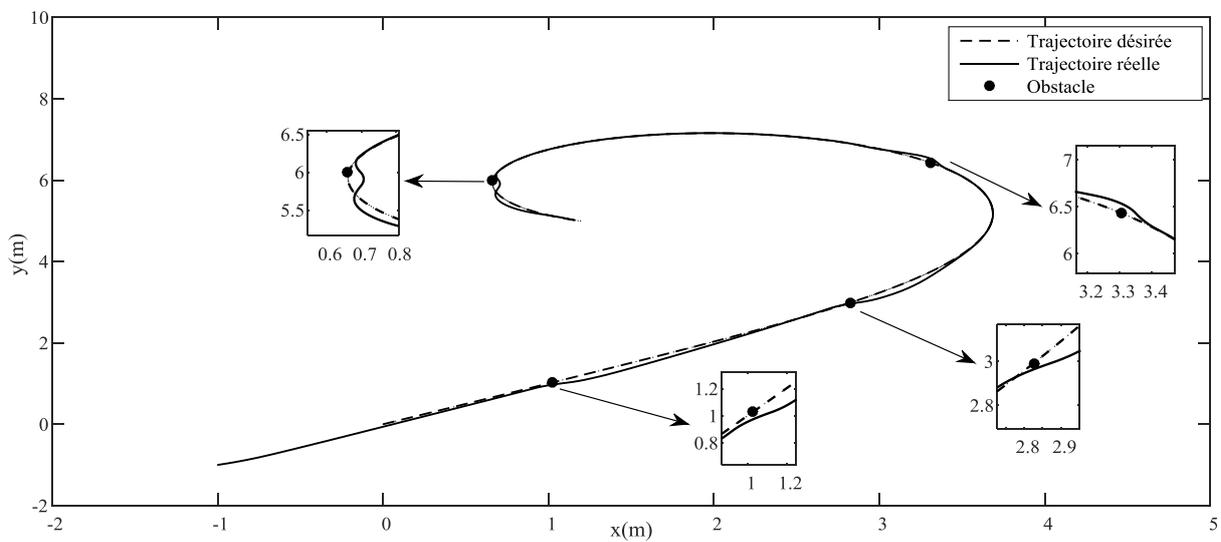


Figure II.12. Poursuite de trajectoire et évitement d'obstacles

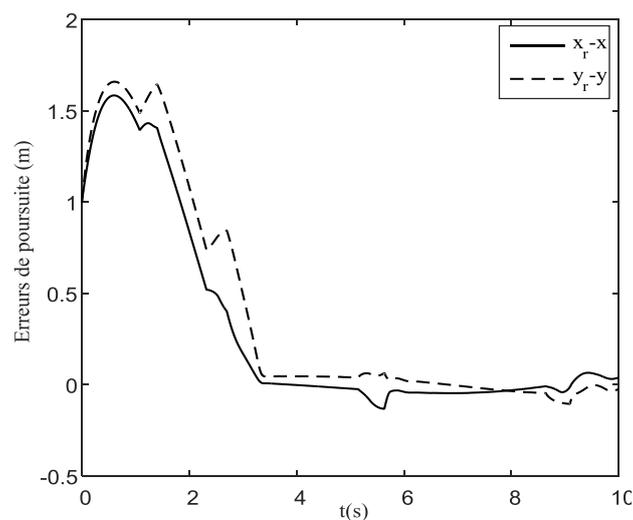


Figure II.13. Erreurs de poursuite de trajectoire

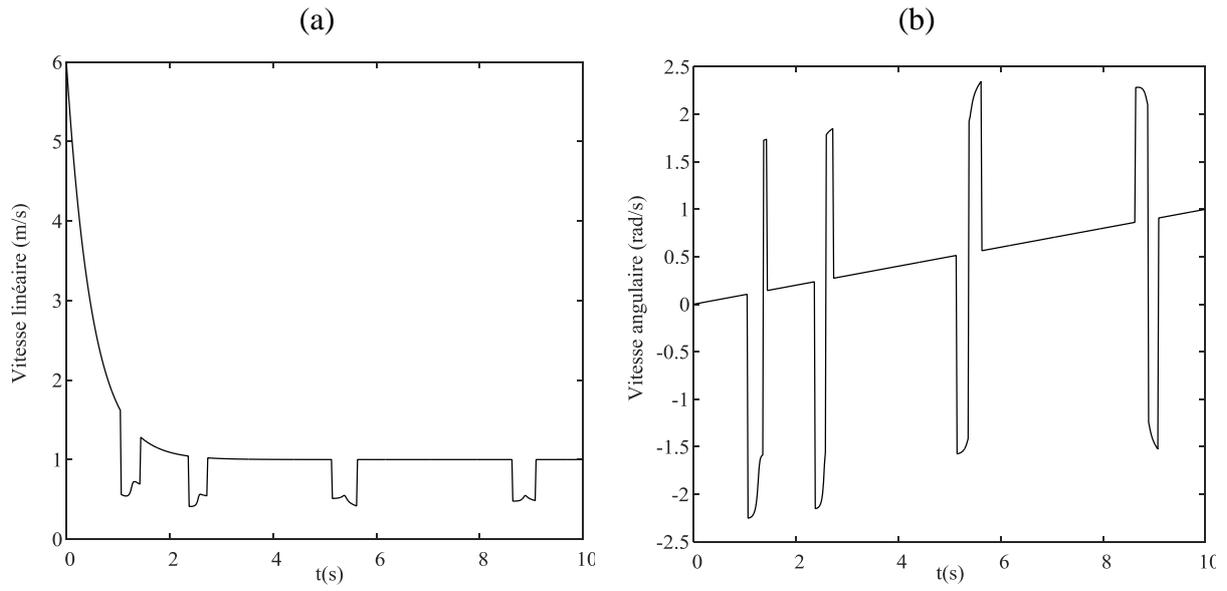
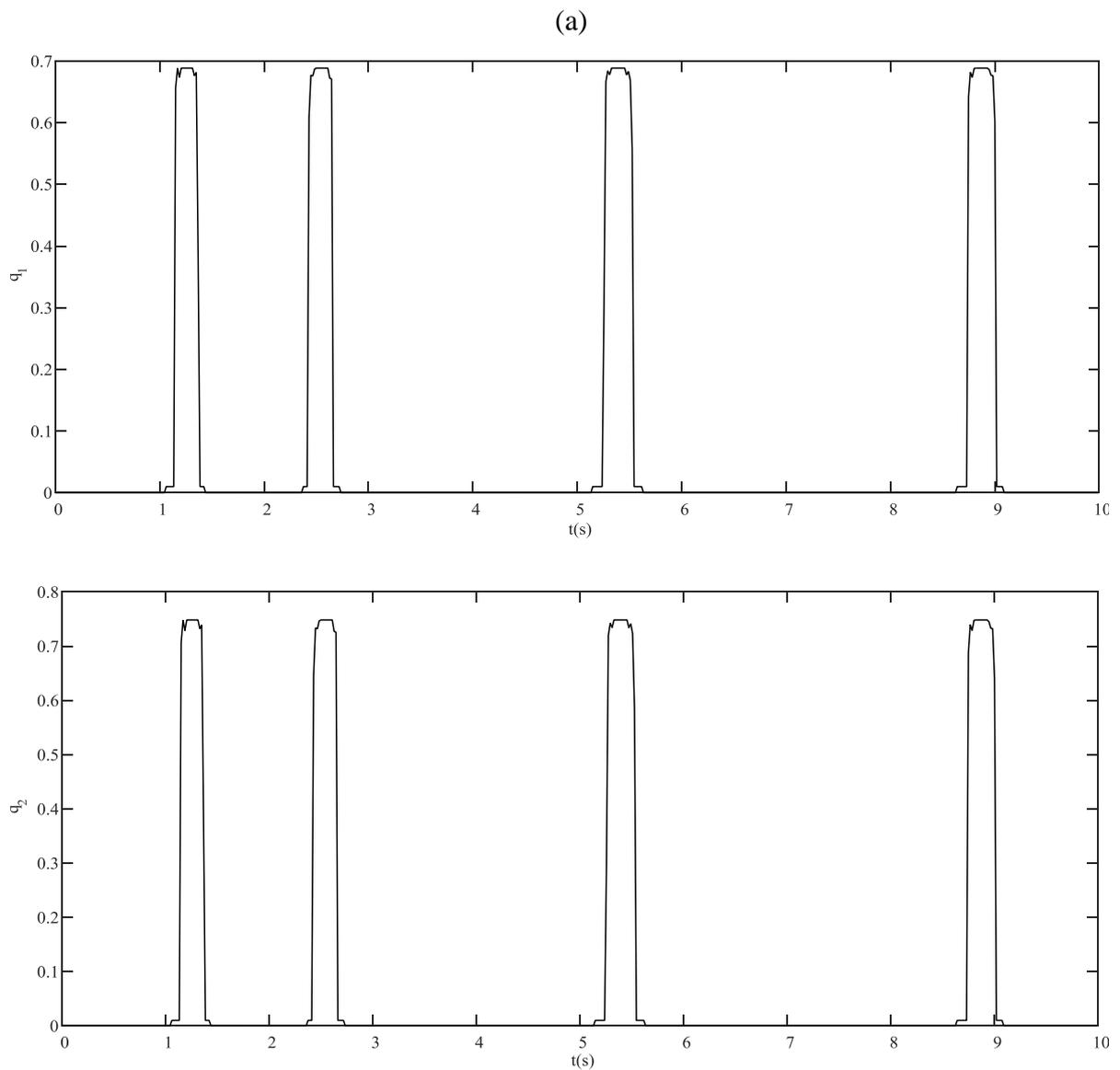
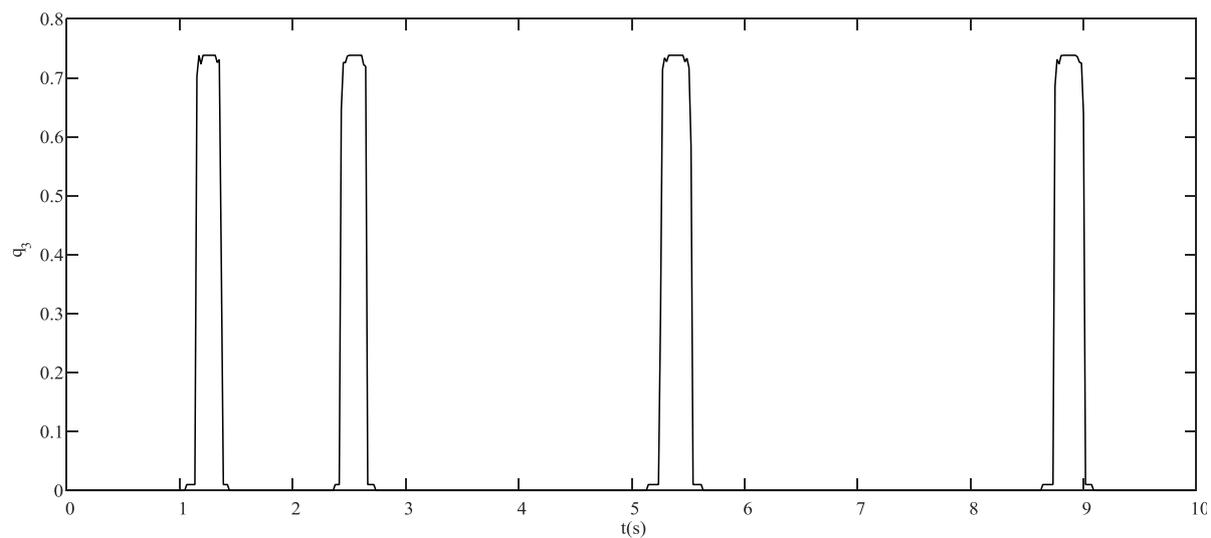
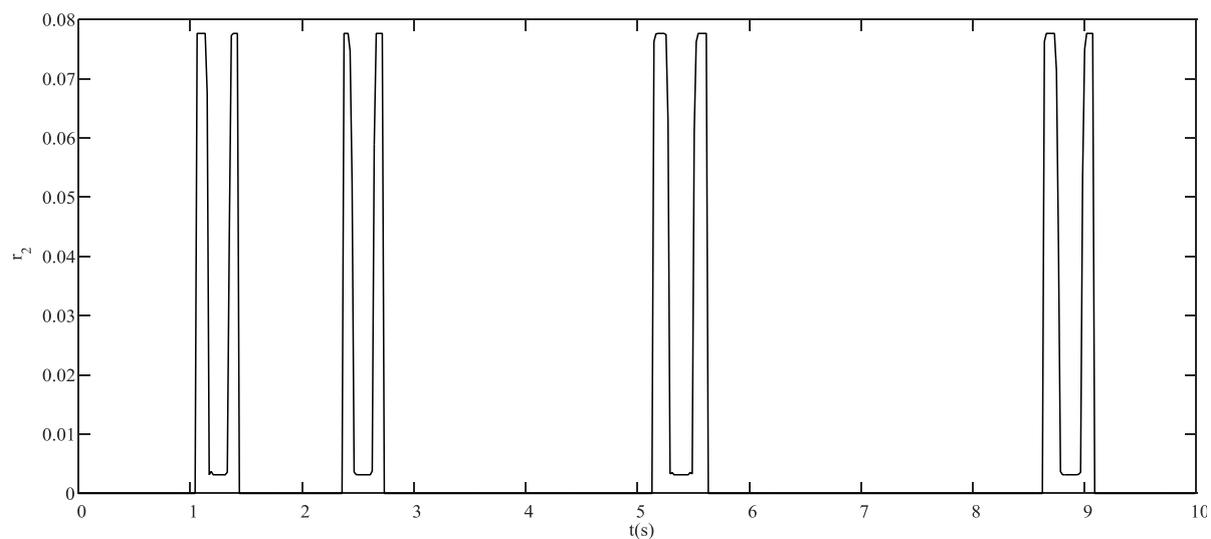
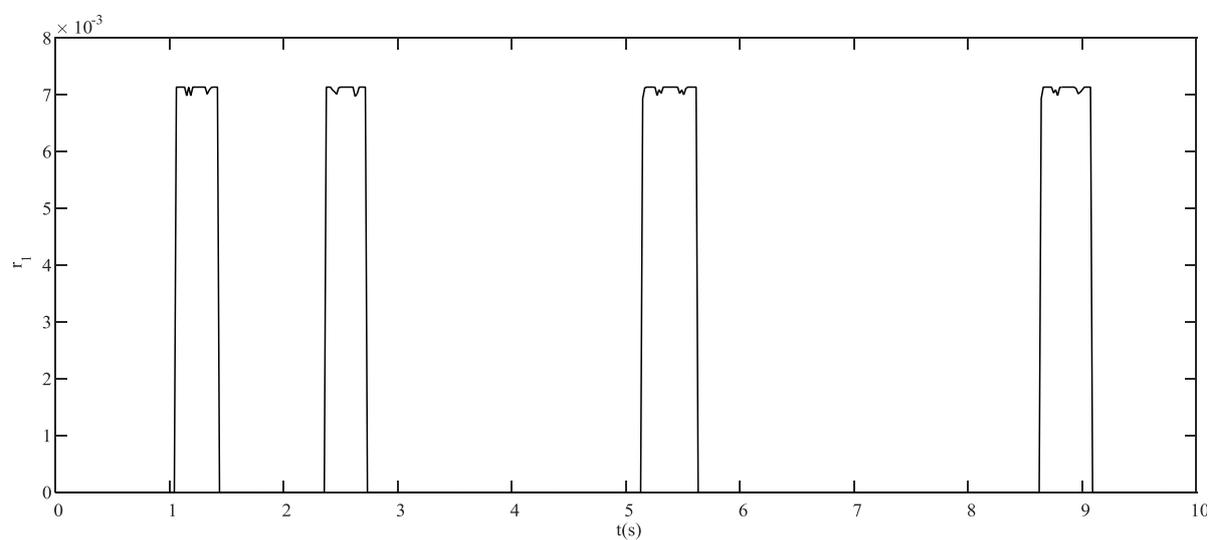


Figure II.14. Signaux de commande: (a) vitesse linéaire et (b) vitesse angulaire





(b)

Figure II.15. Variations des éléments de la diagonale: (a) de la matrice Q , (b) de la matrice R

II.7. Conclusion

Ce chapitre a été consacré au développement d'une stratégie de contrôle d'un robot mobile pour suivre une trajectoire de référence en l'absence et en présence d'obstacles. Le contrôleur de poursuite a été élaboré en utilisant l'approche PDC afin de combiner la commande floue avec la commande optimale (LQ). Un régulateur LQR a été employé pour jouer le rôle d'un contrôleur d'évitement d'obstacles en ajustant ses matrices de pondération en temps réel par un contrôleur flou. Pour s'assurer que le robot mobile soit capable de réaliser la poursuite de trajectoires tout en évitant les obstacles, nous avons combiné les vitesses de poursuite et d'évitement d'obstacles à l'aide d'un gain de fusion généré par un contrôleur flou.

Les résultats de simulation obtenus montrent l'efficacité de l'approche proposée quant à la prise en charge du problème de la poursuite de trajectoires en l'absence et en présence d'obstacles grâce à des signaux de commande continus et faciles à implémenter.

La comparaison réalisée avec les résultats de [Kim-09] et [Wha-10] a permis de confirmer que le contrôleur conçu est capable de garantir de meilleures performances de poursuite en l'absence d'obstacles sur la trajectoire de référence.

CHAPITRE III

PLANIFICATION DE TRAJECTOIRES PAR UN ALGORITHME AMÉLIORÉ DE COLONIES DE FOURMIS

PLANIFICATION DE TRAJECTOIRES PAR UN ALGORITHME AMÉLIORÉ DE COLONIES DE FOURMIS

III.1. Introduction

L'optimisation par les colonies de fourmis est un paradigme de conception des algorithmes méta-heuristiques pour résoudre les problèmes difficiles d'optimisation combinatoire. Elle a été introduite au début des années 1990 par le chercheur italien Marco Dorigo [Dor-92] dans ces études sur les comportements des fourmis durant leurs transitions dans l'environnement à la recherche du chemin court vers la source de nourriture. Depuis lors, nombreuses variantes de l'algorithme de base ont été proposées dans la littérature, et elles ont montré de bonnes performances dans la résolution de plusieurs problèmes, notamment dans les cas d'affectation quadratique [Man-94], du job-shop [Col-94], du voyageur de commerce [Dor-97], du routage sur réseau [Di-98] et du routage de véhicule [Bul-99]. Grâce à ses avantages de traitement parallèle, de calcul distribué et sa bonne robustesse, l'optimisation par les colonies de fourmis est devenue l'approche la plus intéressante dans la famille des systèmes d'optimisation par essaim et elle a été progressivement appliquée dans le domaine de la navigation des robots mobiles [Yue-06], [Tan-07].

Les algorithmes de colonies de fourmis ont réussi à résoudre le problème de planification de trajectoires des robots mobiles, mais malheureusement avec quelques inconvénients, comme la stagnation facile, la vitesse de convergence lente, le temps de recherche long et l'optimisation locale. Dans ce cadre, plusieurs améliorations et combinaisons avec d'autres approches ont été proposées. Parmi elles, nous citons l'ajustement de l'information heuristique en temps réel [Don-09], la conception des contrôleurs flous pour optimiser les paramètres de pondération [Zha-10], l'adoption d'une stratégie de recherche parallèle bidirectionnelle pour réduire le temps de recherche et l'utilisation d'une nouvelle méthode qui distribue rationnellement le phéromone initial pour accélérer la convergence [Zha-12], l'introduction d'une approche qui combine l'algorithme des colonies de fourmis et les

algorithmes génétiques pour garantir une solution globale et rapide [Châ-14], l'emploi de la fenêtre roulante dans la conception de l'algorithme des colonies de fourmis pour assurer une bonne capacité d'analyse et d'élimination des impasses lors de la planification de trajectoires [Hua-16], la combinaison de l'algorithme des colonies de fourmis avec l'algorithme du recuit simulé pour améliorer le processus de mise à jour de phéromone [Che-16] et le développement d'un système de colonies de fourmis chaotique pour éviter que la recherche soit limitée à trouver l'optimum local [Xia-16].

L'objectif de ce chapitre est de présenter un algorithme amélioré de colonies de fourmis. Nous allons insérer dans la règle de transition une probabilité stimulante et une nouvelle information heuristique basée sur le principe de longueur de pas illimitée. Une nouvelle méthode de mise à jour de phéromone et une stratégie d'ajustement dynamique du taux d'évaporation seront adoptées. L'efficacité de l'algorithme proposé sera mise en évidence par simulation sous l'environnement Matlab, à travers la planification de trajectoire d'un robot mobile dans des cartes de grille.

III.2. Formulation du problème

Considérons une fourmi comme un robot mobile ponctuel qui se déplace dans un environnement à deux dimensions où leur position est spécifiée par rapport à un système de coordonnées fixe.

Il s'agit de développer un algorithme de colonies de fourmis pour trouver une séquence ordonnée de points objectifs qui constituent la trajectoire optimale que le robot mobile va suivre vers la cible en présence d'obstacles.

III.3. Modélisation de l'environnement

La première étape de la planification de trajectoires est d'établir un modèle d'environnement qui représente l'espace de la navigation du robot mobile. En général, il existe trois méthodes de modélisation de l'environnement qui sont la méthode de grille, la méthode géométrique et le graphe topologique. Nous allons utiliser la méthode de grille en raison de sa facilité de mise en œuvre et sa précision. Elle consiste à diviser l'environnement en cellules rectangulaires où chaque cellule présente une grille et porte une information binaire qui exprime la grille libre par '0' et la grille obstacle par '1'. La plateforme de grille est un plan bidimensionnel, placé

dans un repère orthogonal homogène XOY de sorte que toutes les grilles possèdent des coordonnées spécifiques.

En supposant que les grilles sont identifiées par des nombres entiers et leurs longueurs égales à la longueur unitaire des coordonnées, nous pouvons calculer les coordonnées de la grille 'i' comme suit :

$$\begin{cases} x_i = \text{mod}(i-1, N_x) + 0.5 \\ y_i = \text{int}((i-1)/N_y) + 0.5 \end{cases} \quad (\text{III.1})$$

où N_x est le nombre de grilles dans une ligne, N_y le nombre de grilles dans une colonne, mod le reste de la division et int une fonction pour obtenir la partie entière d'un nombre décimal.

La figure III.1 montre un exemple d'environnement modélisé par la méthode de grille où $N_x=N_y=10$. Les grilles en noir représentent les obstacles et les grilles en blanc correspondent à l'espace libre dans lequel le robot mobile peut se déplacer en avant, en arrière, à droite, à gauche, en haut à droite, en bas à droite, en haut à gauche et en bas à gauche (voir la figure III.2).

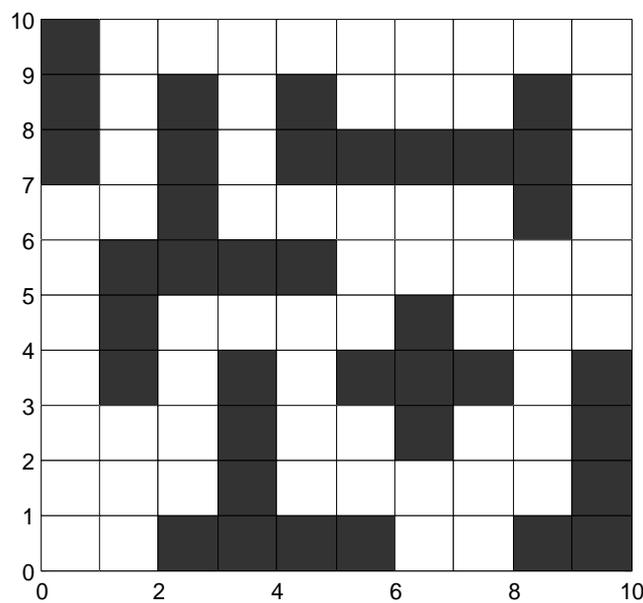


Figure III.1. Environnement modélisé par des grilles

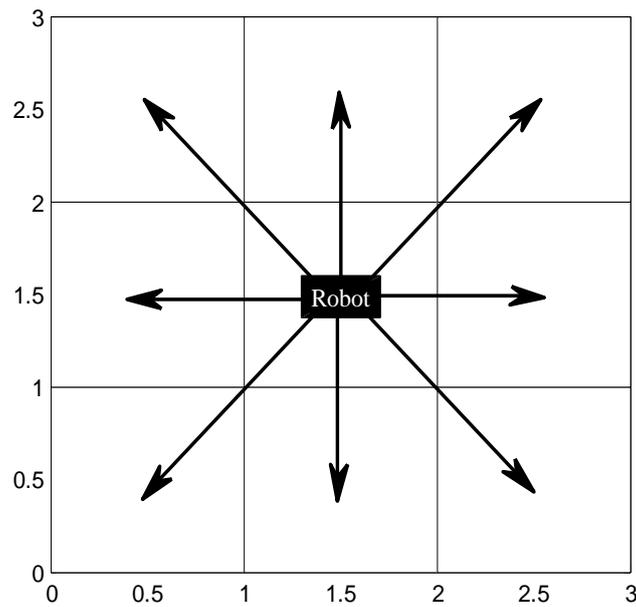


Figure III.2. Directions possibles du mouvement du robot mobile

III.4. Algorithme de colonies de fourmis

L'algorithme de colonie de fourmis a été conçu en s'appuyant sur le comportement collectif des fourmis durant leur processus d'exploitation de l'environnement à la recherche du trajet minimal entre le nid et la source de nourriture. Les fourmis ont la caractéristique de communiquer via des substances volatiles appelées *phéromones*, et elles vont les utiliser pour marquer leurs trajets en construisant des pistes de phéromone. Une fois la source de nourriture est trouvée, les fourmis tendent à se déplacer vers la piste où la concentration de phéromone est élevée, et cela à cause de leur capacité à détecter l'intensité de la piste de phéromone. Tandis que les fourmis continuent à se diriger vers la piste de phéromone forte, la concentration de phéromone sera augmentée ce qui va attirer plus de fourmis, et ainsi la concentration de phéromone sera augmentée encore plus...etc. Selon ce principe, la sélection du trajet optimal est effectuée.

Une fourmi 'k' se déplace de sa grille actuelle 'i' à une grille non visitée 'j' reposant sur la quantité de phéromone sur le bord qui les relie et la distance entre eux. S'il y a plus d'une grille non visitée, la fourmi 'k' va choisir sa prochaine grille en fonction de la probabilité de transition (règle de déplacement) suivante :

$$p_{ij}^k(t) = \begin{cases} \frac{[\tau_{ij}(t)]^\alpha \cdot [\eta_{ij}(t)]^\beta}{\sum_{s \in J_i^k} [\tau_{is}(t)]^\alpha \cdot [\eta_{is}(t)]^\beta} & j \in J_i^k \\ 0 & j \notin J_i^k \end{cases} \quad (\text{III.2})$$

où α et β sont deux paramètres de pondération contrôlant l'influence relative de la concentration de phéromone et l'information heuristique (visibilité), respectivement, J_i^k l'ensemble de grilles non visitées par la fourmi 'k', τ_{ij} la concentration de phéromone sur le chemin entre les grilles 'i' et 'j' et η_{ij} la visibilité de la grille 'j' à partir de la grille 'i' comme suit :

$$\eta_{ij} = \frac{1}{d_{ij}} \quad (\text{III.3})$$

où d_{ij} est la distance entre les centres de deux grilles voisines, et elle peut être calculée par la relation suivante :

$$d_{ij} = \sqrt{(x_i - x_j)^2 + (y_i - y_j)^2} \quad (\text{III.4})$$

Quand toutes les fourmis font un tour complet, les niveaux de phéromones sur tous les arcs seront mis à jour en volatilissant l'ancienne phéromone et en ajoutant la phéromone déposée par chaque fourmi. La formule de mise à jour de phéromone est donnée par :

$$\tau_{ij}(t+1) = (1 - \rho)\tau_{ij}(t) + \sum_{k=1}^n \Delta\tau_{ij}^k \quad (\text{III.5})$$

où ρ est le taux d'évaporation de la phéromone, n le nombre de fourmis et $\Delta\tau_{ij}^k$ la quantité de phéromone laissée par la fourmi 'k' à l'itération courante et elle peut être exprimée comme suit:

$$\Delta\tau_{ij}^k = \begin{cases} \frac{Q}{L_k} & \text{si } (i, j) \in \text{à la tournée} \\ 0 & \text{si } (i, j) \notin \text{à la tournée} \end{cases} \quad (\text{III.6})$$

où L_k est la longueur du trajet trouvé par la fourmi 'k' et Q un paramètre fixé.

Nous pouvons présenter les étapes de planification de trajectoire du robot mobile par les colonies de fourmis dans l'organigramme suivant :

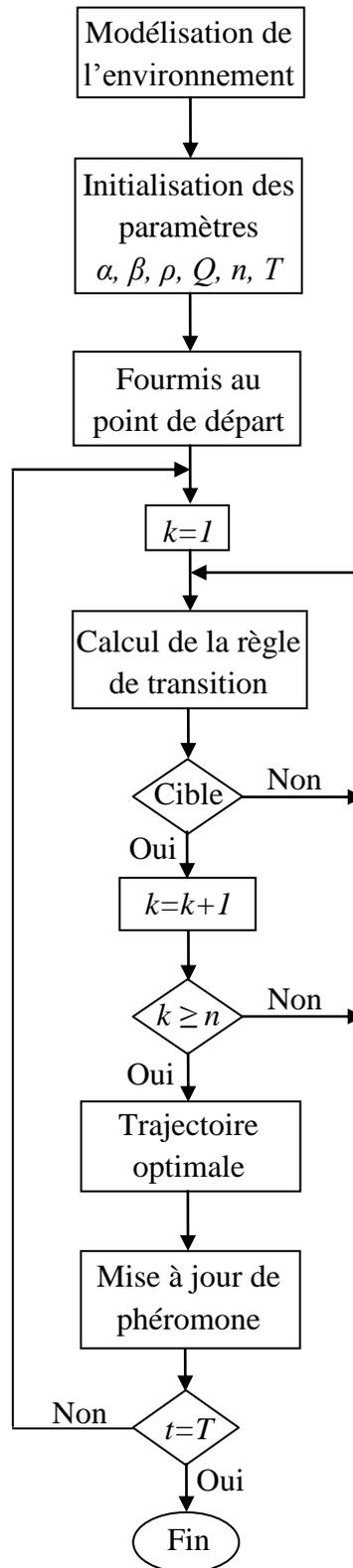


Figure III.3. L'algorithme de colonies de fourmis pour la planification de trajectoire

III.5. Algorithme amélioré de colonies de fourmis

Dans l'algorithme de base des colonies de fourmis, la sélection probabiliste ne garantit pas toujours une solution optimale (dans les premières étapes de l'optimisation, l'algorithme peut tomber dans un optimum local), et la recherche heuristique limitée conduit à une vitesse de convergence faible. Afin de remédier à ces inconvénients, nous allons introduire les améliorations suivantes [Akk-18c] : une probabilité stimulante, une information heuristique globale, une nouvelle règle de mise à jour de phéromone et une nouvelle méthode d'évaporation dynamique.

III.5.1. Probabilité stimulante

III.5.1.1. Notations et préliminaires

III.5.1.1.1. Arrangement

Un arrangement de p éléments pris sans répétition parmi les m éléments ($p \leq m$) d'un ensemble E est une permutation ordonnée de ces p éléments distincts. Le nombre de permutations de p parmi m est noté A_m^p et défini par:

$$A_m^p = \frac{m!}{(m-p)!} \quad (\text{III.7})$$

Où le symbole ! représente le factoriel.

III.5.1.1.2. Combinaison

Soit E un ensemble constitué de m éléments. Nous appelons une combinaison de p éléments pris sans répétition dans E toute collection non ordonnée de E contenant p éléments. Le nombre de combinaisons de p parmi m est noté C_m^p et défini par:

$$C_m^p = \frac{A_m^p}{p!} = \frac{m!}{p!(m-p)!} \quad (\text{III.8})$$

III.5.1.1.3. Probabilité d'un évènement

Considérons Ω un ensemble des évènements élémentaires. Un évènement A de Ω est un sous ensemble qui comprend des éléments de Ω appelés variables aléatoires. La probabilité d'évènement A notée $P(A)$ est le pourcentage de chances que ce dernier se produise de

l'expérience aléatoire. Il s'agit du rapport entre le nombre de résultats favorables à l'évènement et le nombre de résultats possibles de l'expérience aléatoire.

$$P(A) = \frac{\text{nombre de cas favorables}}{\text{nombre de cas possibles}} \quad (\text{III.9})$$

La probabilité stimulante (sp) est une opération combinatoire qui dépend du nombre d'obstacles entourant la grille. Nous l'avons adopté pour permettre à une fourmi de choisir la meilleure prochaine grille, qui doit être accessible et avoir plus d'une sortie vers la cible. Donc une grille est préférable s'il y a moins d'obstacles autour d'elle. Pour déterminer cette probabilité, nous devons avoir les combinaisons suivantes [Akk-18c] :

- $C_8^{N_{obs}}$: est le nombre de distributions possibles des obstacles autour d'une certaine grille 'j'.

Il peut être calculé comme suit :

$$C_8^{N_{obs}} = \frac{8!}{N_{obs}!(8-N_{obs})!} \quad (\text{III.10})$$

- $C_{8-N_{obs}-1}^1$: exprime le nombre des sorties d'où la fourmi 'k' peut quitter la grille 'j' via la relation suivante :

$$C_{8-N_{obs}-1}^1 = \frac{(8-N_{obs}-1)!}{(8-N_{obs}-1-1)!} \quad (\text{III.11})$$

où N_{obs} est le nombre d'obstacles autour de la grille 'j' et huit le nombre maximal d'obstacles.

Remarque 1: Dans le calcul du nombre de sorties, nous devons éliminer celle qui a été utilisée par la fourmi 'k' pour accéder à la grille 'j'.

Remarque 2: Les obstacles et les sorties ont le même nombre de distributions possibles autour d'une grille $C_8^{N_{obs}} = C_8^{8-N_{obs}}$.

La probabilité qui stimule la fourmi 'k' à choisir la grille 'j' est définie par:

$$sp_{ij}^k = \frac{C_{8-N_{obs}-1}^1}{C_8^{N_{obs}}} \quad (\text{III.12})$$

La nouvelle règle de transition est donnée par l'équation suivante :

$$p_{ij}^k(t) = \begin{cases} \frac{[\tau_{ij}(t)]^\alpha \cdot [\eta_{ij}(t)]^\beta \cdot sp_{ij}}{\sum_{s \in J_i^k} [\tau_{is}(t)]^\alpha \cdot [\eta_{is}(t)]^\beta \cdot sp_{is}} & j \in J_i^k \\ 0 & j \notin J_i^k \end{cases} \quad (\text{III.13})$$

III.5.2. Information heuristique globale

Dans l'algorithme classique de colonies de fourmis, la longueur du pas est fixée comme étant la distance entre les centres de deux grilles adjacentes, ce qui réduit le nombre de grilles candidates et la diversité de recherche, conduisant à une vitesse de convergence lente et un comportement de stagnation qui rend l'algorithme incapable de trouver la trajectoire la plus courte. Pour améliorer ce genre de performance nous avons établi une nouvelle information heuristique basée sur le principe de la longueur de pas libre [Zen-16]. Il s'agit d'étendre le champ de vision et la recherche heuristique de sorte que toutes les grilles dans le champ de vision du robot mobile soient incluses lors du processus de sélection de la prochaine grille.

Supposons que 'i', 'j', 'S' et 'E' sont respectivement les grilles actuelle, prochaine, de départ et cible, et d la distance entre leurs centres comme suit : $d(S,i)$, $d(i,E)$, $d(S,j)$, $d(j,E)$, $d(i,j)$ et $d(S,E)$. Notre information heuristique est définie par les formules suivantes [Akk-18c] :

-Dès la première transition, les fourmis doivent être orientées vers la cible.

$$\eta_{SE}(0) = \frac{1}{d(S,E)} \quad (\text{III.14})$$

- Après le premier déplacement, nous augmentons la diversité de la recherche heuristique mais, nous devons garder la direction de la recherche vers la cible.

Si $d(S,i) < d(S,j)$ et $d(i,E) > d(j,E)$ alors : les fourmis avancent et s'approchent de la cible.

$$\eta_{ij}(t) = \frac{d(i,j)}{d(j,E)} \quad (\text{III.15})$$

Si $d(S,i) < d(S,j)$ et $d(i,E) < d(j,E)$ alors : les fourmis ne sont pas orientées vers la cible.

$$\eta_{ij}(t) = \frac{d(i,j)}{d(S,E)} \quad (\text{III.16})$$

Si $d(S,i) > d(S,j)$ alors : les fourmis se déplacent vers l'arrière.

$$\eta_{ij}(t) = \frac{1}{d(S, E)} \cdot \frac{d(j, E)}{d(i, j)} \quad (\text{III.17})$$

La distance d est calculée selon l'équation (III.4).

III.5.3. Règle de mise à jour de phéromone

La quantité de phéromone est l'un des facteurs nécessaires pour choisir les meilleures grilles qui mènent à un chemin court. Cependant, elle contient une certaine intensité de mauvaise phéromone qui est produite par les pires fourmis et qui peut faire déplacer d'autres fourmis vers des grilles où la cible est inaccessible ou peut provoquer une solution prématurée de l'algorithme. Afin d'atteindre une bonne performance, à chaque itération nous avons augmenté la concentration de phéromone dans le meilleur trajet local et nous l'avons diminué dans le pire trajet local en utilisant la règle de mise à jour suivante [Akk-18c] :

$$\tau_{ij}(t+1) = (1-\rho)\tau_{ij}(t) + \sum_{k=1}^n \Delta\tau_{ij}^k(t) + \Delta\tau_{ij}^{best} - \Delta\tau_{ij}^{worst} \quad (\text{III.18})$$

où

$$\Delta\tau_{ij}^{best} = \begin{cases} sp_{\max} \cdot n_{best} \cdot \frac{Q}{L_{best}} & \text{si } (i, j) \in \text{au meilleur trajet local} \\ 0 & \text{si } (i, j) \notin \text{au meilleur trajet local} \end{cases} \quad (\text{III.19})$$

$$\Delta\tau_{ij}^{worst} = \begin{cases} sp_{\min} \cdot n_{worst} \cdot \frac{Q}{L_{worst}} & \text{si } (i, j) \in \text{au pire trajet local} \\ 0 & \text{si } (i, j) \notin \text{au pire trajet local} \end{cases} \quad (\text{III.20})$$

avec sp_{\max} et sp_{\min} sont les probabilités stimulantes maximale et minimale respectivement, n_{best} et n_{worst} les nombres de fourmis meilleures et pires respectivement, L_{best} et L_{worst} respectivement, les longueurs du trajet local meilleur et pire.

Remarque 3: Dans le principe de la longueur de pas libre, lorsque la fourmi choisit sa prochaine grille, il y aura parfois des grilles secondaires entre la grille actuelle et la prochaine grille dans lesquelles la fourmi va laisser une quantité de phéromone. Mais puisqu'elle est juste entrain de passer et non pas de rester, la quantité de phéromone dans ces grilles sera très faible, ce qui ne va pas pris en compte durant le processus de mise à jour de phéromone.

III.5.4. Stratégie d'évaporation dynamique

Dans l'algorithme traditionnel de colonies de fourmis, le taux d'évaporation est une constante dans la gamme $[0,1]$, il a un impact direct sur la capacité de recherche globale et la vitesse de convergence de l'algorithme. Si le taux d'évaporation ρ est trop petit, l'évaporation de phéromone sera très lente, ce qui réduit la capacité de recherche globale et amène l'algorithme à une convergence locale. Si ρ est trop grand, la recherche globale sera améliorée et la vitesse de convergence sera réduite. Pour réaliser un bon équilibre entre l'expansion de l'espace de recherche et l'accélération de la vitesse de convergence, nous avons laissé le taux d'évaporation changer dynamiquement. Au début de l'algorithme nous l'avons fixé à une grande valeur afin d'améliorer la capacité de recherche, puis il sera réduit en fonction du modèle de décroissance suivant [Akk-18c] :

$$\rho(t) = \frac{T \cdot t \cdot \frac{1}{n} \sum_{k=1}^n sp^k \cdot (\tau_{\max} - \tau_{\min})}{T-1} + \frac{\frac{1}{n} \sum_{k=1}^n sp^k \cdot (T \cdot \tau_{\min} - \tau_{\max})}{T-1} \quad (\text{III.21})$$

où t et T sont respectivement les itérations actuelle et maximale, sp^k la probabilité stimulante de la fourmi k , et τ_{\max} et τ_{\min} les quantités maximale et minimale de phéromone respectivement.

III.6. Exemples de simulation

Afin d'illustrer la validité et l'efficacité des améliorations apportées à l'algorithme de colonies de fourmis, des simulations sous l'environnement Matlab sont développées pour réaliser la planification de trajectoire du robot mobile dans deux environnements de grilles.

Les paramètres de l'algorithme sont supposés comme suit :

- la taille du premier environnement : $N_x=N_y=20$.
- la taille du deuxième environnement : $N_x=N_y=30$.
- le nombre de fourmis : $n=30$.
- l'itération maximale : $T=100$.
- les facteurs de pondération : $\alpha = 1$ et $\beta = 5$.
- la constante de phéromone : $Q = 2$.

Dans les deux environnements, la grille de départ 'S' est au coin inférieur gauche et la grille cible 'E' est au coin supérieur droit.

Les figures III.4a et III.4b présentent les trajectoires optimales obtenues en appliquant l'algorithme original et amélioré de colonies de fourmis. Les figures III.5a et III.5b montrent la variation des longueurs des trajectoires optimales correspondantes en fonction de l'itération.

D'après les résultats de simulation, il est évident que l'algorithme amélioré de colonies de fourmis est caractérisé par de bonnes performances de planification et d'optimisation de la trajectoire (trajectoire optimale et convergence rapide de l'algorithme).

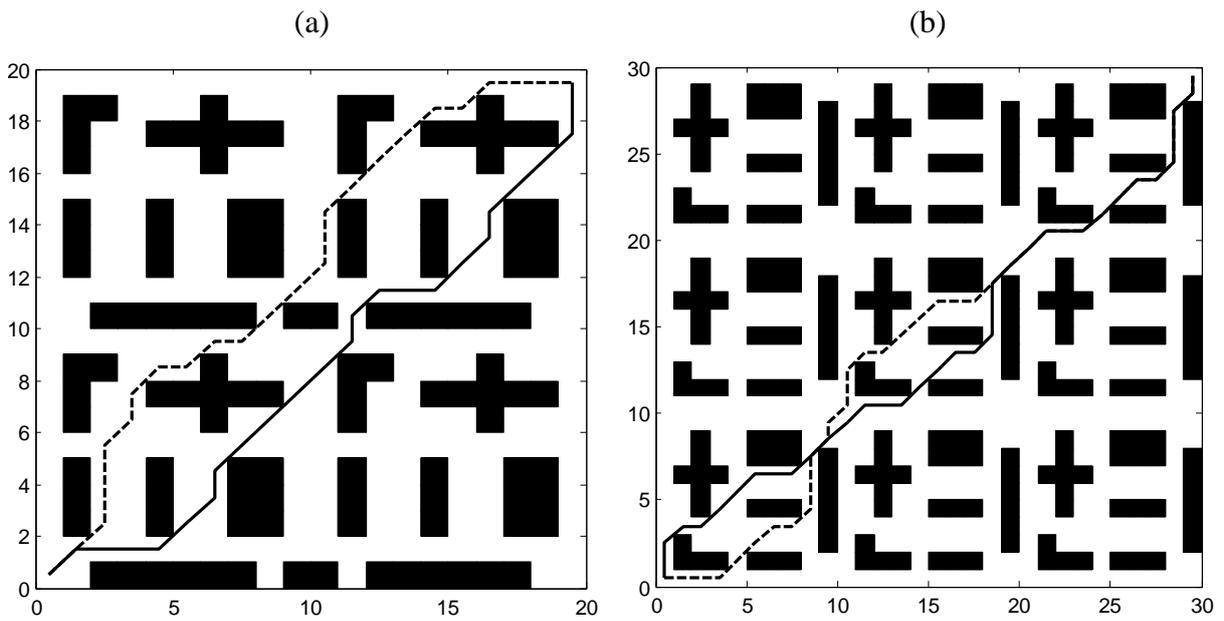


Figure III.4. Planification des trajectoires optimales

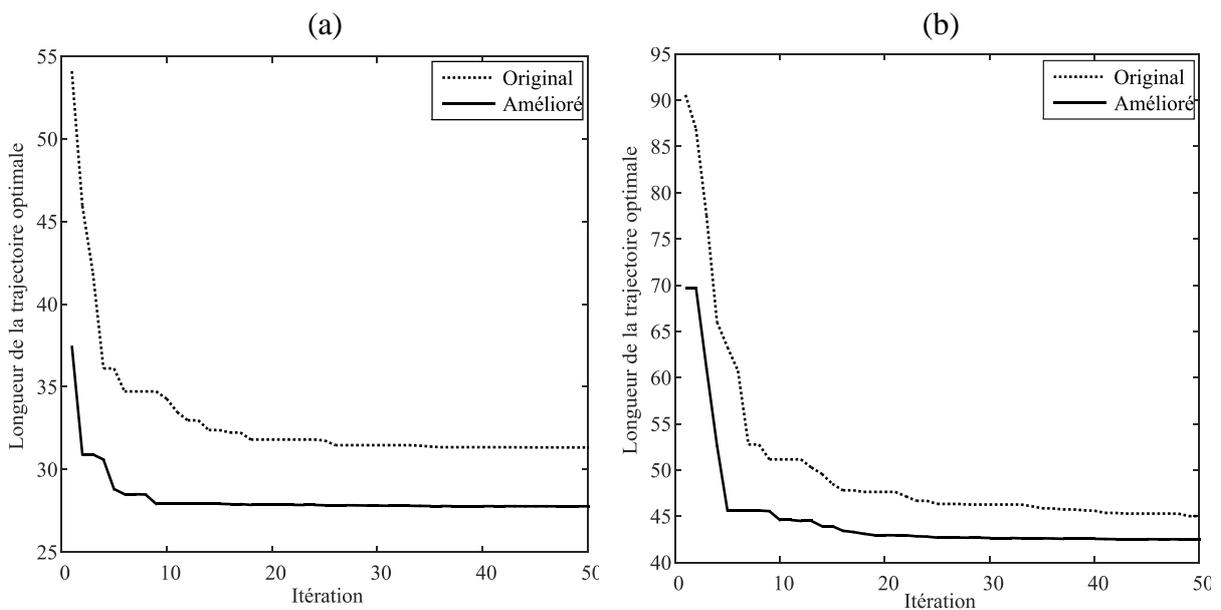


Figure III.5. Evolution des longueurs des trajectoires optimales

Pour quantifier les performances obtenues par l'algorithme de colonies de fourmis classique et celui amélioré, nous allons considérer la distance parcourue par le robot mobile et le temps (nombre d'itération) nécessaire pour atteindre la grille cible dans deux environnements différents (voir tableau III.1).

Tableau III.1. Résultats statistiques des algorithmes dans les deux environnements.

Environnement	L'algorithme original		L'algorithme amélioré	
	Itérations	Longueur du trajet (m)	Itérations	Longueur du trajet (m)
1	25	31.841	10	28.7612
2	40	46.8605	25	43.6874

III.7. Etude comparative

Afin de situer les performances de l'algorithme amélioré de colonies de fourmis, nous allons comparer les résultats obtenus avec ceux de la référence [Jin-16] en termes de trajectoire optimale planifiée, de vitesse de convergence et de longueur moyenne de la trajectoire optimale.

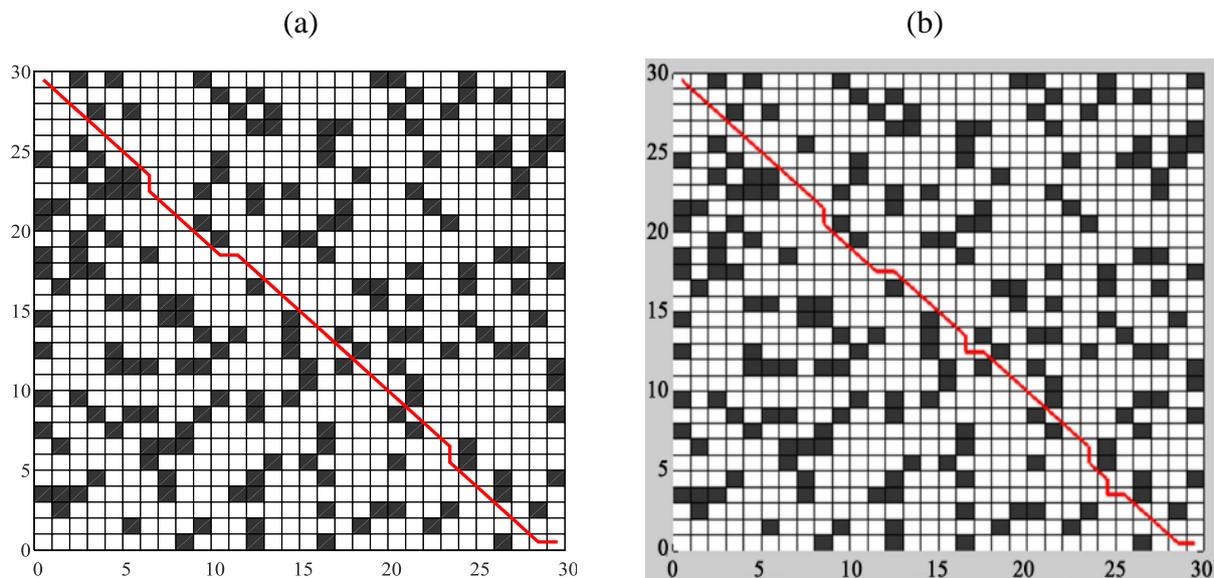


Figure III.6. Trajectoire optimale: (a) résultats obtenus, (b) résultats de [Jin-16]

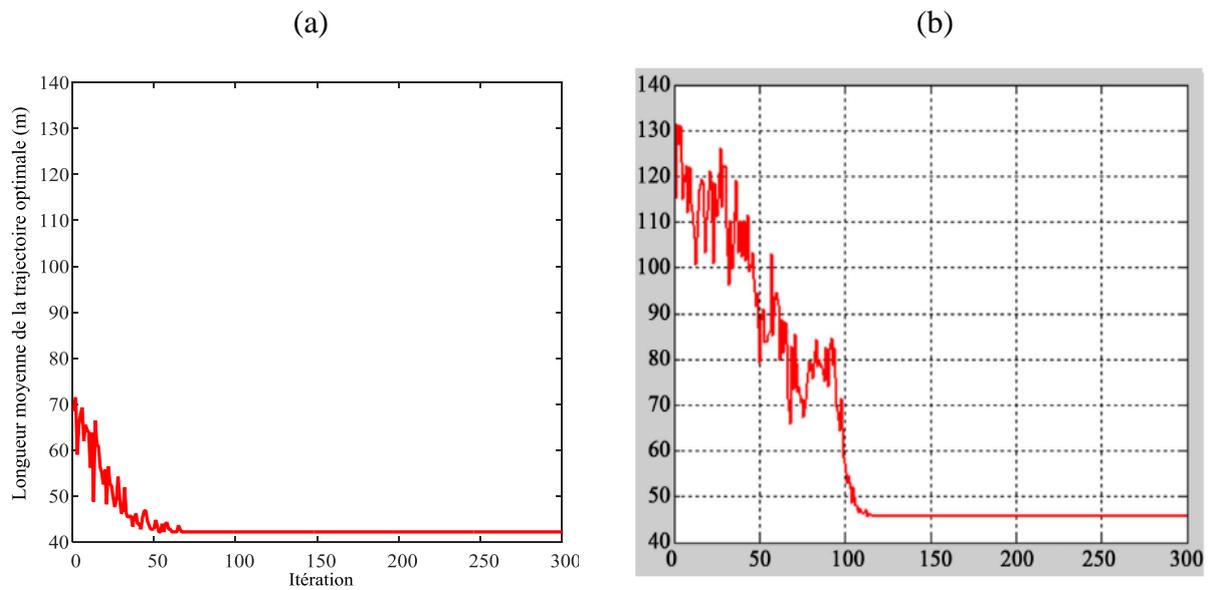


Figure III.7. Longueur moyenne de la trajectoire optimale: (a) résultats obtenus, (b) résultats de [Jin-16].

La comparaison des résultats obtenus avec ceux de la référence [Jin-16] montre que notre trajectoire optimale est plus courte (figure III.6 et III.7) et que la convergence de notre algorithme est plus rapide (figure III.7).

III.8. Conclusion

Dans ce chapitre, un algorithme amélioré de colonies de fourmis a été proposé, et sa performance a été comparée avec celle de l'algorithme de base lors de la planification des trajectoires dans des environnements modélisés en grilles. Les améliorations ont été effectuées par: (i) l'introduction d'une probabilité stimulante au niveau de la règle de transition pour choisir une grille sécurisée et accessible, (ii) l'exhaustivité de l'information heuristique en utilisant le principe de la longueur de pas libre afin d'augmenter la précision de la visibilité ainsi que l'efficacité de la recherche, (iii) l'établissement d'une règle de mise à jour de phéromone et (iv) l'élaboration d'une stratégie d'évaporation dynamique. Les deux dernières améliorations ont pour objectifs d'accroître la capacité de recherche et d'accélérer la convergence.

Les exemples de simulation, sous l'environnement Matlab, montrent le potentiel de l'algorithme de colonies de fourmis proposé pour résoudre le problème de la planification de trajectoires et mettent en évidence sa performance en termes de minimisation de la longueur de trajectoire et de vitesse de convergence.

L'étude comparative que nous avons réalisée entre les résultats obtenus et ceux de la référence [Jin-16], a prouvé l'efficacité de l'amélioration apportée à l'algorithme de colonies de fourmis à travers la rapidité de convergence et la minimisation de la longueur du trajet (l'optimalité de la trajectoire planifiée).

CONCLUSION GÉNÉRALE

CONCLUSION GÉNÉRALE

Les travaux présentés dans cette thèse traitent le problème de la navigation autonome des robots mobiles de type unicycle dans des environnements statiques. En fait, il s'agit de quelques fonctions de la navigation qui sont la planification de trajectoires, la poursuite de trajectoires et l'évitement d'obstacles. Au cours de la synthèse des méthodologies de contrôle des robots unicycle, ces tâches ont été considérées comme des problématiques singulières et des problématiques composées (la structure de contrôle comprend deux fonctions compatibles de la navigation).

La première partie a été consacrée au développement des approches permettant de résoudre le problème de la planification et le suivi des trajectoires. Pour planifier les trajectoires nous avons adopté une méthode réactive appelée la fenêtre dynamique et pour suivre ces trajectoires nous avons utilisé le contrôle flou en poursuite. Après un bref aperçu sur les robots mobiles unicycle et des principes de base de la logique floue et de la commande floue, nous avons présenté une étude détaillée de la fenêtre dynamique classique ainsi que l'amélioration que nous avons proposée et qui consiste à employer un contrôleur flou afin de régler en temps réel les facteurs de pondération de la fonction "objectif" par rapport aux conditions environnementales. Ensuite, deux contrôleurs flous ont été conçus pour mettre en œuvre la poursuite des trajectoires données et produites par la fenêtre dynamique floue. Le premier amène le robot mobile vers la position et l'orientation désirées et le second améliore l'orientation.

Dans la seconde partie, nous avons proposé et développé une technique de contrôle qui sert à intégrer la poursuite de trajectoires avec l'évitement d'obstacles. L'idée proposée consiste à diriger le robot mobile unicycle pour suivre un robot virtuel qui se déplace avec précision sur la trajectoire de référence tout en évitant les obstacles rencontrés sans abandonner l'attitude de poursuite. La structure de contrôle est basée sur le modèle cinématique de l'erreur de posture entre le robot mobile réel et le robot mobile virtuel. Pour réaliser la tâche de poursuite, nous avons exploité l'approche PDC, dans laquelle un régulateur LQR a été assigné à chaque sous

système. Afin de prévenir les collisions, nous avons fait fonctionner un régulateur LQR comme un contrôleur d'évitement d'obstacles en ajustant les valeurs de ses matrices de pondération en temps réel par un contrôleur flou. Les commandes de poursuite et d'évitement d'obstacles ont été fusionnées à travers un gain provenant d'un contrôleur flou.

La dernière partie a été dédiée à la planification de trajectoires des robots mobiles à l'aide d'un algorithme méta-heuristique. Il s'agit de l'application d'un algorithme de colonies de fourmis dans des environnements de grilles pour planifier la trajectoire optimale. D'abord, nous avons présenté le concept de l'algorithme de base de colonies de fourmis, puis expliqué les améliorations apportées à savoir; (i) l'insertion d'une probabilité stimulante dans la règle de transition pour sélectionner une grille sûre et facile à accéder, (ii) l'intégralité de l'information heuristique suivant le principe de pas illimité afin d'hausser la précision de la visibilité ainsi que l'efficacité de la recherche, (iii) la mise en place d'une règle d'actualisation de phéromone et (iv) l'élaboration d'une méthode d'évaporation dynamique de phéromone. Les deux dernières améliorations permettent de renforcer la capacité de recherche et de hâter la convergence.

Toutes les approches proposées ont été développées et simulées sous l'environnement Matlab. Les résultats obtenus, qui sont très performants en termes de vitesse de convergence, de stabilité sur la trajectoire de référence, d'optimalité de la trajectoire planifiée et de caractéristiques des signaux de commande, ont permis de mettre en évidence la contribution de nos stratégies quant à la résolution de certains problèmes de navigation des robots mobiles.

Pour donner plus de crédibilité aux travaux réalisés dans le cadre de cette thèse, nous avons situé les résultats obtenus par rapport à ceux des références du domaine. Dans le chapitre I, la comparaison faite avec [Kla-07] a montré que la stratégie de commande en poursuite proposée est plus efficace en ce qui concerne la vitesse de convergence, la stabilité sur la trajectoire de référence et les propriétés des signaux de commande. Dans le chapitre II, la comparaison des résultats avec [Kim-09] et [Wha-10] a prouvé que l'approche proposée est capable d'assurer de meilleures performances de poursuite en l'absence d'obstacles sur la trajectoire désirée. Dans le chapitre III, l'étude comparative effectuée avec [Jin-16] a démontré que les améliorations apportées à l'algorithme de colonies de fourmis ont significativement bonifié la performance de la planification en termes de vitesse de convergence et d'optimalité du trajet.

D'après les résultats obtenus, qui sont très encourageants, plusieurs perspectives peuvent être envisagées. À court terme, il serait intéressant d'étendre les approches proposées pour d'autres modèles de robots mobiles et de considérer des environnements avec des obstacles dynamiques. À long terme, une validation expérimentale donnera plus de poids et plus de crédibilité aux approches proposées.

BIBLIOGRAPHIE

BIBLIOGRAPHIE

[Akk-18a]

Akka, K., and Khaber, F., Optimal tracking control of a trajectory planned via fuzzy reactive approach for an autonomous mobile robot, *International journal of advanced robotic systems*, Vol. 15, N°. 1, Doi: [10.1177/1729881418760624](https://doi.org/10.1177/1729881418760624), 2018.

[Akk-18b]

Akka, K., and Khaber, F., Optimal fuzzy tracking control with obstacles avoidance for a mobile robot based on Takagi-Sugeno fuzzy model, *Transactions of the institute of measurement and control*, Doi: [10.1177/0142331218811462](https://doi.org/10.1177/0142331218811462), 2018.

[Akk-18b]

Akka, K., and Khaber, F., Mobile robot path planning using an improved ant colony optimization, *International journal of advanced robotic systems*, Vol. 15, N°. 3, Doi: [10.1177/1729881418774673](https://doi.org/10.1177/1729881418774673), 2018.

[Ant-07]

Antonelli, G., Chiaverini, S., and Fusco, G., A fuzzy-logic-based approach for mobile robot path tracking, *IEEE transactions on fuzzy systems*, Vol. 15, N°. 2, pp. 211-221, 2007.

[Arr-02]

Arras, K.O., Persson, J., Tomatis, N., and Siegwart R., Real-time obstacle avoidance for polygonal robots with a reduced dynamic window. *Proc. of IEEE international conference on robotics and automation*, Washington, DC, USA, Vol. 3, pp. 3050–3055, 2002.

[Bha-07]

Bhattacharya, P., and Gavrilova, M.L., Voronoi diagram in optimal path planning, *Proc. of the 4th international symposium on voronoi diagrams in science and engineering*, Glamorgan, UK, pp. 38-47, 2007.

[Bor-91]

Borenstein, J., and Koren, Y., The Vector field histogram - fast obstacle avoidance for mobile robots, *IEEE transactions on robotics and automation*, Vol. 7, N°. 3, pp. 278-288, 1991.

[Bro-99]

Brock, O., and Khatib O., High-speed navigation using the global dynamic window approach, *Proc. of the IEEE international conference on robotics and automation*, Detroit, MI, Us, Vol. 1, pp. 341-346, 1999.

[Büh-94]

Bühler, H., *Réglage par logique floue*, Presse Polytechniques et Universitaires Romandes, 1994.

[Bul-99]

Bullheimer, B., Hartl, R.F., and Strauss, C., Applying the ant system to the vehicle routing problem, *Dordrecht: Kluwer Academic Publishers*, pp. 285-296, 1999.

[Can-87]

Canny, J., A new algebraic method for robot motion planning and real geometry, 28th annual symposium on foundations of computer science, Los Angeles, CA, USA, pp. 39-48, 1987.

[Châ-14]

Châari, I., Koubâa, A., Trigui, S., Bennaceur, H., Ammar, A., and Al-Shalfan K., SmartPATH: an efficient hybrid ACO-GA algorithm for solving the global path planning problem of mobile robots, *Int. J. Adv. Robot Syst.*, Vol. 11, N°. 7, pp. 94-108, 2014.

[Cha-15]

Chand, P., Fuzzy reactive control for wheeled mobile robots. *Proc. of the 6th international conference on automation, robotics and applications*, Queenstown, New Zealand, pp. 167-172, 2015.

[Che-16]

Cheng, J., Miao, Z., Li, B., and Xu, W., An improved ACO algorithm for mobile robot path planning, *Proc. of IEEE international conference on information and automation*, Ningbo, China, pp. 963-968, 2016.

[Cho-12]

Choi, B., Kim, B., Kim, E., and Yang K.W., A modified dynamic window approach in crowded indoor environment for intelligent transport robot. *Proc. of the 12th international conference on control, automation and systems*, JeJu Island, South Korea, pp. 1007-1009, 2012.

[Col-94]

Colorni, A., Dorigo, M., Maniezzo V., and Trubian, M., Ant system for job-shop scheduling, *Belgian J. Oper. Res. Stat. Comput. Sci*, Vol. 34, pp. 39–53, 1994.

[Dec-85]

Dechter, R., and Pearl, J., Generalized best-first search strategies and the optimality of A*, *Journal of the ACM*, Vol. 32, N° 3, pp. 505–536, 1985.

[Di-98]

Di-Caro, G., and Dorigo, M., Distributed stigmergetic control for communications networks, *J. Artif. Int. Res.*, Vol. 9, pp. 317-365, 1998.

[Don-09]

Dong, J., Liu B., Peng, K., and Yin, Y., Robot obstacle avoidance based on an improved ant colony algorithm, *Proc. of IEEE computer society global congress on intelligent systems*, Xiamen, China, pp. 103-106, 2009.

[Dor-92]

Dorigo, M., Optimization learning and natural algorithms, PhD thesis, University of Milano, Italy, 1992.

[Dor-97]

Dorigo, M., and Luca, M.G., Ant colony system: a cooperative learning approach to the traveling salesman problem, *IEEE Trans. Evolut. Comput.*, Vol. 1, N° 1, pp. 53-66, 1997.

[Fox-96]

Fox, D., Burgard, W., and Thrun, S., Controlling synchro-drive robots with the dynamic window approach to collision avoidance, *Proc. of the IEEE/RSJ international conference on intelligent robots and systems*, Osaka, Japan, Vol. 3, pp. 1280–1287. 1996.

[Gah-95]

Gahinet, P., Nemirovski, A., Laub, A., and chilali M., LMI control toolbox. *The MathWorks, Inc*, 1995.

[Gas-99]

Gasso, K., Mourot, G., and Ragot, J., Ozone concentration modelling using a multiple model approach, *Proc. Of the EUFIT*, Aachen, Germany, 1999.

[Gas-00]

Gasso, K., Mouro, G., and Ragot, J., Identification of an output error Takagi-Sugeno model. *Proc. of IEEE conference on systems, man and cybernetics*, Nashville, USA, Vol. 1, pp. 14-19, 2000.

[Gel-77]

Gelperin, D., On the optimality of A^* , *Artificial intelligence*, Vol. 8, N°. 1, pp. 69-76, 1977.

[Hua-16]

Huang, Q., and Zheng, G., Route optimization for autonomous container truck based on rolling window. *Int. J. Adv. Robot Syst.*, Vol. 13, N°. 3, pp. 112-121, 2016.

[Jin-16]

Jingang, C., Robot global path planning based on an improved ant colony algorithm. *Journal of computer and communications*, Vol. 4, N°. 2, pp. 11-19, 2016.

[Kan-90]

Kanayama, Y., Kimura, Y., Miyazaki, F., and Noguchi, T., A stable tracking control method for an autonomous mobile robot. *Proc. of the IEEE international conference on robotics and automation*, Cincinnati, OH, USA, pp. 384-389, 1990.

[Kan-10]

Kanjanawanishkul, K., Hofmeister, M., and Zell, A., Path following with an optimal forward velocity for a mobile robot. *Proc. of the 7th IFAC symposium on intelligent autonomous vehicles*, Lecce, Italy, pp. 19-24, 2010.

[Kha-86]

Khatib, O., Real-Time obstacle avoidance for manipulators and mobile robots, *The international journal of robotics research*, Vol. 5, N°. 1, pp. 90-98, 1986.

[Kim-09]

Kim, C.J, Kwon, J.W., and Chwa, D., Polynomial fuzzy modeling and tracking control of wheeled mobile robot via sum squares approach, *Proc. of the IEEE international conference on industrial technology*, Gippsland, VIC, Australia, pp. 973-977, 2009.

[Kis-12]

Kiss, D., and Tevesz, G., Advanced dynamic window based navigation approach using model predictive control. *Proc. of the 17th international conference on methods and models in automation and robotics*, Miedzyzdrojcie, Poland, pp. 149-153, 2012.

[Kla-07]

Klančar, G., and Škrjanc, I., Tracking-error model-based predictive control for mobile robots in real time, *Robotics and autonomous systems*, Vol. 55, N°. 6, pp. 460-469, 2007.

[Lat-91]

Latombe, J.C., Robot motion planning, *Kluwer Academic Publishers*, Dodrecht, Netherlands, 1991.

[Lav-98]

Lavalle, S.M., Rapidly-exploring random trees: A new tool for path planning, 1998.

[Lee-03]

Lee, T.H., Lam, H.K., Leung, F.H.F., and Tam, P.K.S., A practical fuzzy logic controller for the path tracking of wheeled mobile robots, *IEEE control systems magazine*, pp. 60-65, 2003.

[Lew-95]

Lewis, F.L., and Syrmos, V.L., Optimal control, *John Wiley and Sons*, New York, 1995.

[Lin-01]

Lin, C.M., and Mon, Y.J., A fuzzy-PDC-based control for robotic systems, *Informations sciences*, Vol. 137, N°.1-4, pp. 135-155, 2001.

[Liu-94]

Liu, K., and Lewis, F.L., Fuzzy logic-based navigation controller for an autonomous mobile robot, *Proc. of IEEE [international conference on systems, man and cybernetics](#)*, Vol. 2, pp. 1782-1789, 1994.

[Liu-09]

Liu, N., Intelligent path following method for nonholonomic robot using fuzzy control, *Proc. of the 2nd international conference on intelligent networks and intelligent systems*, Tianjin, China, pp. 282-285, 2009.

[Mam-74]

Mamdani, E.H., Application of Fuzzy Algorithms for Simple Dynamic Plant, *Proc. of IEEE control and science*, Vol. 121, N°. 12, pp. 1585-1588, 1974.

[Man-94]

Maniezzo, V., Colorni, A., and Dorigo, M., The ant system applied to the quadratic assignment problem, *Technical report IRIDIA*, Bruxelles, Belgium, 1994.

[Min-00]

Minguez, J., and Montano, L., Nearness diagram navigation (ND): A new real time collision Avoidance approach, *Proc. of the IEEE/RSJ international conference on intelligent robots and systems*, Takamatsu, Japan, Vol. 3, pp. 2094-2100, 2000.

[Mou-15]

Mousavi, M.A., A fast fuzzy path tracking controller for mobile robots, *Proc. of 3rd RSI International conference on robotics and mechatronics*, Tehran, Iran, pp. 675-680, 2015.

[Nak-98]

Nak, Y.K., and Simmons, R., The lane-curvature method for local obstacle avoidance, *Proc. of the IEEE/RSJ international conference on intelligent robots and systems*, Victoria, BC, Canada, Vol. 3, pp. 1615-1621, 1998.

[Nij-90]

Nijmeijer, H., and Schaft, A.V.D., Nonlinear dynamical control systems, *Springer Verlag*, New York, 1990.

[Oll-94]

Ollero, A., Garcia-Cereso, A., and Martinez, J.L., Fuzzy supervisory path tracking of mobile robots, *Control Eng. practice*, Vol. 2, N°. 2, pp. 313-319, 1994.

[Oua-08]

Ouadah, N., Ourak, L., and Boudjema, F., Car-like mobile robot oriented positioning by fuzzy controllers, *International journal of advanced robotic systems*, Vol. 5, N°. 3, pp. 249-256, 2008.

[Phi-03]

Philippsen, R., and Siegwart, R., Smooth and efficient obstacle avoidance for a tour guide robot. *Proc. of IEEE international conference on robotics and automation*, Taipei, Taiwan, Vol. 1, pp. 446–451, 2003.

[Sch-07]

Schröter, C., Höchemer, M., and Gross, H.M., A particle filter for the dynamic window approach to mobile robot control. *Proc. of the international scientific colloquium (IWK)*, Ilmenau, Germany, Vol. 1, pp. 425-430, 2007.

[Šed-07]

Šeda, M., Roadmap methods vs. cell decomposition in robot motion planning, *Proc. of the 6th WSEAS international conference on signal processing, Robotics and automation*, pp. 127-132, 2007.

[Sim-96]

Simmons, R., The curvature-velocity method for local obstacle avoidance, *Proc. of IEEE international conference on robotics and automation*, Minneapolis, MN, USA, Vol. 4, pp. 3375-3382, 1996.

[Ski-90]

Skiena, S., Dijkstra's algorithm, in implementing discrete mathematics: Combinatorics and graph theory with mathematica, *Addison-Wesley*, Boston, Mass, USA, pp. 225-227, 1990.

[Tak-85]

Takagi, T., and Sugeno, M., Fuzzy Identification of Systems and its Applications to Modeling and Control, *IEEE Trans. on systems, man and cybernetics*, Vol. 15, N°. 1, pp. 116-132, 1985.

[Tan-01]

Tanaka K., Hua O., Wang H.O., Fuzzy control systems design and analysis a linear matrix inequality, *John Wiley and Sons*, New York, 2001.

[Tan-07]

Tan, G.Z., HE, H., and Sloman, A., Ant colony system algorithm for real time globally optimal path planning of mobile robots, *Acta Autom. Sinica*, Vol. 33, pp. 279–285, 2007.

[Tse-01]

Tseng, C.H., Chen, B.S., and Uang, H.J., Fuzzy tracking control design for nonlinear dynamic systems via T-S fuzzy model, *IEEE transactions on fuzzy systems*, Vol. 9, N°. 3, pp. 381–392, 2001.

[Vad-00]

Vadakkepat, P., Tan, K.C., and Ming-Liang, W., Evolutionary artificial potential fields and their application in real time robot path planning, *Proc. of the IEEE congress on evolutionary computation*, Vol. 1, pp. 256-263, 2000.

[Wha-10]

Whang, G., Wang, F.G., Yoon, T.S., and Park, S.K., Control of mobile robot based on T-S fuzzy linearization. *Proc. of the IEEE international conference on intelligent systems design applications*, Cairo, Egypt, pp. 192-196, 2010.

[Xia-16]

Xiaoming, Y., Kai, L., and Sheng, L., A chaotic ant colony system for path planning of mobile robot, *International journal of hybrid information technology*, Vol. 9, N°. 1, pp. 329-338, 2016.

[Yan-04]

Yang, S.X., and Luo, C., A neural network approach to complete coverage path planning, *IEEE transactions on systems, man, and cybernetics, part b: cybernetics*, Vol. 34, N°. 1, pp. 718–725, 2004.

[Yoo-11]

Yoo, S.J., Adaptive tracking and obstacle avoidance for a class of mobile robots in the presence of unknown skidding and slipping, *IET control theory and applications*, Vol. 5, N°. 14, pp. 1597-1608, 2011.

[Yue-06]

Yue, FZ., Cui, P.Y., and Cui, H.T., Planetary rover path planning based on ant colony optimization algorithm, *Control Dec.* Vol. 21, pp. 1437-1440, 2006.

[Zad-65]

Zadeh, L.A., Information and control, *Fuzzy sets*, Vol. 8, pp. 338-353, 1965.

[Zen-16]

Zeng, M.R., Xi, L., and Xiao, A.M., The free step length ant colony algorithm in mobile robot path planning, *Advanced robotics*, Vol. 30, N°. 23, pp. 1509-1514, 2016.

[Zha-10]

Zhao, J.P., Liu, J.G., Gao, X.W., and Chen, Y.Q., Research of path planning for mobile robot based on improved ant colony optimization algorithm, *Proc. of the 2nd IEEE international conference on advanced computer control*, Shenyang, China, pp. 241-245, 2010.

[Zha-12]

Zhao, J., and Fu, X., Improved ant colony optimization algorithm and its application on path planning of mobile robot. *Journal of computer*, Vol. 7, N°. 8, pp. 2055–2062, 2012.

ANNEXE

Publications personnelles

AKKA Khaled and KHABER Farid

Optimal tracking control of a trajectory planned via fuzzy reactive approach for an autonomous mobile robot,

International journal of advanced robotic systems, Vol. 15, N°. 1,

Doi: [10.1177/1729881418760624](https://doi.org/10.1177/1729881418760624), 2018.

Impact Factor: **0,952**

AKKA Khaled and KHABER Farid

Optimal fuzzy tracking control with obstacles avoidance for a mobile robot based on Takagi-Sugeno fuzzy model,

Transactions of the institute of measurement and control, 2018.

Doi: [10.1177/0142331218811462](https://doi.org/10.1177/0142331218811462),

Impact Factor: **1,579**

AKKA Khaled and KHABER Farid

Mobile robot path planning using an improved ant colony optimization,

International journal of advanced robotic systems, Vol. 15, N°. 3, 2018.

Doi: [10.1177/1729881418774673](https://doi.org/10.1177/1729881418774673),

Impact Factor: **0,952**

Résumé: Les travaux présentés dans le cadre de cette thèse ont pour objectif la résolution des problèmes de la navigation des robots mobiles autonomes de type unicycle dans des environnements statiques. Dans ce contexte, il s'agit d'élaborer une stratégie de contrôle flou pour le suivi de trajectoires planifiées par la fenêtre dynamique floue, de concevoir une structure de commande optimale floue permettant la poursuite de trajectoires en l'absence et en présence d'obstacles et d'améliorer un algorithme de colonies de fourmis afin de planifier des trajectoires optimales dans des environnements de grilles. La fenêtre dynamique a été bonifiée en ajustant les paramètres de sa fonction "objectif" par un contrôleur flou et la trajectoire produite a été suivie en exploitant deux contrôleurs flous, le premier amène le robot mobile vers la position et l'orientation désirées, le second améliore l'orientation. La compensation parallèle distribuée a été combinée avec la commande linéaire quadratique pour assurer la poursuite de trajectoire et un régulateur linéaire quadratique a été utilisé comme un contrôleur d'évitement d'obstacles en manipulant ces matrices de pondération à travers un contrôleur flou. Les commandes générées ont été fusionnées par un gain flou. L'algorithme de colonies de fourmis a été amélioré selon quatre aspects: (i) l'introduction d'une probabilité stimulante dans la règle de transition, (ii) l'extension de l'information heuristique, (iii) l'adoption d'une nouvelle stratégie de mise à jour de phéromone et (iv) la proposition d'une méthode d'évaporation dynamique.

Mots-clés: Robots mobiles unicycle, Planification de trajectoires, Suivi de trajectoires, Évitement d'obstacles, Fenêtre dynamique, Commande floue, Commande linéaire quadratique, Algorithme de colonies de fourmis.

Abstract: The works presented in this thesis aim to solve navigation problems for autonomous unicycle mobile robots in static environments. In this context, it is about elaborating a fuzzy control strategy to follow trajectories planned by the fuzzy dynamic window, designing an optimal fuzzy control structure that allows the trajectory tracking in the absence and in the presence of obstacles and improving an ant colony algorithm for optimal path planning in grid maps. The dynamic window has been improved by adjusting the parameters of its objective function via a fuzzy controller and the resulting path has been followed by exploiting two fuzzy controllers, the first brings the mobile robot to the desired position and orientation, the second enhances the orientation. The parallel distributed compensation has been combined with the linear quadratic controller to ensure the trajectory tracking and a linear quadratic regulator has been employed as an obstacles avoidance controller by manipulating its weighting matrices using a fuzzy controller. The generated control signals have been merged by a fuzzy gain. The ant colony algorithm has been improved through four aspects: (i) introducing a stimulating probability into the transition rule, (ii) extending the heuristic information, (iii) adopting a new pheromone updating strategy and (iv) proposing a dynamic evaporation method.

Key words: Unicycle mobile robots, Path planning, Trajectory tracking, Obstacles avoidance, Dynamic window, Fuzzy control, Linear quadratic control, Ant colony algorithm.

ملخص: تهدف الأعمال المقدمة في إطار هذا البحث إلى حل مشاكل الملاحة للروبوتات المتحركة الذاتية من نوع إنيسكيل في البيئات الساكنة. في هذا السياق، الأمر يتعلق بوضع إستراتيجية تحكم غامض لتتبع المسارات المخططة بواسطة النافذة الديناميكية الغامضة، بتصميم بنية تحكم مثالية غامضة تسمح بتتبع المسار في غياب وجود عقبات وتحسين خوارزمية مستعمرة النمل لتخطيط المسارات في بيئات مقسمة على شكل شبكات. النافذة الديناميكية تم تحسينها بضبط معاملات دالة الهدف باستخدام التحكم الغامض و المسار الناتج تم تتبعه باستغلال نظامين غامضين ، الأول يقود الروبوت نحو الموضع والتوجيه المرغوبين و الثاني يحسن التوجيه. التوزيع التكميلي المتوازي تم جمعه بالتحكم الخطي التريبيعي لضمان تتبع المسار و وحدة التحكم الخطي التريبيعي تم توظيفها كنظام لتجنب الحواجز وذلك بتغيير قيم مصفوفات الترجيح باستخدام وحدة تحكم غامض. إشارات التحكم الناتجة تم دمجها بمعامل دمج غامض . خوارزمية مستعمرة النمل تم تحسينها من خلال أربعة جوانب: (أ) إدخال احتمال محفز في القاعدة الانتقالية، (ب) توسيع نطاق المعلومات الإرشادية، (ج) اعتماد إستراتيجية جديدة لتحديث الفرمون و (د) اقتراح طريقة ديناميكية للتبخر.

الكلمات المفتاحية: الروبوتات المتحركة إنيسكيل، تخطيط المسار، تتبع المسار، تجنب العقبات ، النافذة الديناميكية ، التحكم الغامض، التحكم الخطي التريبيعي، خوارزمية مستعمرة النمل.