



**Université Batna 2 - Mostefa Ben Boulaid**  
**Faculté de Technologie**  
**Département d'Electronique**



**Thèse**

Présentée pour l'obtention du diplôme de :  
**Doctorat en Sciences en Electronique**  
**Option : Robotique**

Sous le Thème :

**Commande Robuste Appliquée à la Navigation d'un Robot  
Mobile**

Présentée par :

**MALLEM Ali**

**Devant le jury composé de :**

Mr. SAIDI Lamir	Prof.	Université de Batna 2	Président
Mr. SLIMANE Noureddine	MCA.	Université de Batna 2	Rapporteur
Mr. BOUKABOU Abdelkrim	Prof.	Université de Jijel	Examineur
Mr. ABDESSEMED Yassine	Prof.	Université de Batna 2	Examineur
Mr. SOUKKOU Ammar	MCA.	Université de Jijel	Examineur
Mr. CHAIBA Azzeddine	MCA	Université de Khenchela	Examineur

**07 Novembre 2018**



**Université Batna 2 - Mostefa Ben Boulaid**  
**Faculté de Technologie**  
**Département d'Electronique**



**Thèse**

Présentée pour l'obtention du diplôme de :  
**Doctorat en Sciences en Electronique**  
**Option : Robotique**

Sous le Thème :

**Commande Robuste Appliquée à la Navigation d'un Robot  
Mobile**

Présentée par :

**MALLEM Ali**

**Devant le jury composé de :**

Mr. SAIDI Lamir	Prof.	Université de Batna 2	Président
Mr. SLIMANE Noureddine	MCA.	Université de Batna 2	Rapporteur
Mr. BOUKABOU Abdelkrim	Prof.	Université de Jijel	Examineur
Mr. ABDESSEMED Yassine	Prof.	Université de Batna 2	Examineur
Mr. SOUKKOU Ammar	MCA.	Université de Jijel	Examineur
Mr. CHAIBA Azzeddine	MCA	Université de Khenchela	Examineur

**07 Novembre 2018**

## ملخص

العمل المقدم في هذه الرسالة يتركز على تطوير الخوارزميات المختلفة لضمان ملاحه الروبوت المحمول. تهدف الإستراتيجية المقترحة إلى دمج المنطق الضبابي في نظام ملاحه مستقل من خلال الانضمام إلى هدف محدد مع تجنب العقبات في بيئة معروفة جزئياً. نحن نستخدم نظام تحكم قوي يمكنه تسهيل مهمة الملاحه ، والتغلب على أوجه عدم التيقن المقدمة في العالم الحقيقي ومساعدة الروبوت على إتباع المسار. نحن مهتمون من خلال تقنية التحكم بالانزلاق السريع لتتبع المسار ، بناءً على النموذج الكنماتيكي والديناميكي للروبوت. تجمع الإستراتيجية المقترحة بين مقاربتين للتحكم ، التحكم الحركي لتحديد السرعات الكافية لضمان استقرار النظام في وقت محدد. بالإضافة إلى التحكم الديناميكي بتقنية الأسلوب □ السريع للانزلاق العام لتحقيق استقرار أخطاء السرعات ، ووحدة تحكم الشبكة العصبية لتقريب الدالة غير الخطية في وجود الاضطرابات.

⌘ **كلمات مفتاحية** — الروبوت المتحرك ، وتجنبالحواجز ، خوارزمية المجال الاصطناعي المحتملة ، والمنطق الضبابي ، والتحكم في وضع الانزلاق السريع العام ، واستقرار Lyapunov ، الشبكات العصبية .

## Résumé

Le travail présenté dans cette thèse consiste à élaborer différents algorithmes pour assurer une bonne et robuste navigation à un robot mobile. Une première approche utilisant la théorie des champs de potentiels artificiels a été appliquée pour l'évitement d'obstacles, par la suite une autre stratégie, intégrant le raisonnement flou, est proposée dans un système de navigation autonome pour permettre au robot mobile de rejoindre un but bien défini avec évitement d'obstacles dans un environnement partiellement connu. Par ailleurs pour effectivement faciliter cette tâche de navigation, de surmonter les incertitudes présentées dans le monde réel et d'aider le robot à suivre ces trajectoires, un modèle de commande robuste est proposé. On s'est orienté vers l'approche d'une commande par mode glissant rapide et global pour le suivi de trajectoires, en se basant sur les modèles cinématique et dynamique du robot. La stratégie proposée dans ce cas combine deux approches de commande, une commande cinématique pour déterminer les vitesses adéquates afin d'assurer une bonne stabilité du système en un temps fini et une commande dynamique utilisant le mode glissant rapide global pour stabiliser les erreurs de vitesses. De plus pour pallier aux effets des perturbations existantes aussi bien internes qu'externes, cette dernière commande a été associée à un contrôleur neuronal pour éventuellement approximer une fonction non linéaire reflétant ces différentes perturbations.

**Mots clés**—robot mobile, évitement d'obstacles, champs de potentiels artificiels, logique floue, commande par mode glissant rapide global, stabilité de Lyapunov, réseaux de neurones.

## Abstract

The work presented in this thesis consists in working out various algorithms to ensure a good and robust navigation to a mobile robot. A first approach using the artificial potential field theory was applied for the obstacles avoidance, thereafter another strategy, integrating the fuzzy reasoning, is proposed in an autonomous navigation system to allow the mobile robot to join a well defined goal with obstacles avoidance in a partially known environment. In addition for facilitating this navigation task indeed, to overcome the uncertainties presented in the real-world and to help the robot to follow these trajectories, a robust control model is proposed. One was interested in the approach of global fast sliding mode control for the trajectories follow-up, using the kinematics and dynamics models of the robot. The strategy suggested in this case combines two control approaches, a kinematic control to determine adequate velocities in order to ensure a good stability of the system in a finished time and a dynamic control using the global fast sliding mode to stabilize the velocities errors. Moreover to mitigate the effects of the existing disturbances as well internal as external, this last control was associated with a neuronal controller for if required approximating a nonlinear function reflecting these various disturbances.

**Keywords**—*mobile robot, obstacles avoidance, artificial potential field, fuzzy logic, global fast sliding mode control, Lyapunov stability, neural network.*

# Remerciements

Je voudrais exprimer, par ce modeste travail, toute ma gratitude et ma considération à Monsieur **Noureddine SLIMANE**, maître de conférences à l'université de Batna 2, pour avoir suivi mon travail, pour ses remarques, ses conseils, son aide et son soutien, qu'il trouve ici l'expression de ma reconnaissance la plus sincère.

Je remercie très sincèrement Monsieur le Professeur **Lamir SAIDI** de l'université de Batna 2 pour l'intérêt qu'il a manifesté pour ce travail et d'avoir accepté de présider le jury de soutenance de cette thèse.

Toute ma gratitude va également à Monsieur le Professeur **Yassine ABDESSEMED** de l'université de Batna 2 pour avoir accepté d'évaluer ce modeste travail.

Que Monsieur **Abdelkrim BOUKABOU** Professeur à l'université de Jijel soit remercié pour avoir accepté d'examiner mon travail et pour sa participation au jury de soutenance.

Monsieur **Ammar SOUKKOU**, maître de conférences à l'université de Jijel et Monsieur **Azzeddine CHAIBA**, maître de conférences à l'université de Khenchela, ont aimablement accepté d'examiner mon travail et de participer au jury, je les remercie pour l'intérêt qu'ils ont porté à ce travail.

Je remercie très spécialement mon collègue Walid **BENAZIZA**, doctorant au département d'électronique à l'université de Batna 2, pour toute l'aide qu'il m'a apportée. Qu'il trouve ici l'expression de ma profonde gratitude.

Je présente ma gratitude, mon respect et ma reconnaissance à tous les membres de ma famille pour leur soutien et leur patience. Je remercie de tout mon cœur mes parents, ma femme, mes sœurs et mes frères.

J'exprime mes remerciements aussi à tous ceux qui ont contribué de près ou de loin à l'élaboration de ce travail.

Enfin, j'adresse mes remerciements à tous mes amis.

# *Dédicace*

À mes très chers parents

À ma femme, mon fils Nibrasse , ainsi que mes frères et sœurs

À toute mes amis sans exception.

RESUME ET MOTS CLES	
SOMMAIRE	
NOTATIONS ET SYMBOLES	
INTRODUCTION GENERALE	1

## CHAPITRE 1

### MODELISATION DES ROBOTS MOBILES

I.1. Introduction	7
I.2. Modélisation cinématique	7
I.2.1 Robots mobiles de type unicycle	8
I.2.2 Robots mobiles de type tricycle	9
I.2.3 Robot mobile différentiel	11
I.3. Modélisation dynamique	14
I.3.1. Modélisation dynamique générale du robot mobile	14
I.3.2 L'approche de Newton-Euler	15
I.3.3 Application de l'approche de Lagrange	16
I.3.4 Application de l'approche Newton –Euler	27
I.4. Conclusion	31

## CHAPITRE 2

### ALGORITHMES D'EVITEMENTS D'OBSTACLES

II.1 Introduction	32
II.2 Algorithme des champs de potentiel artificiel	33
II.2.1. Définition	33
II.2.2. Concepts de base des champs de potentiels artificiels	34
II.2.3 Développement de l'algorithme des champs de potentiels artificiels	35
II.2.4 Champs de potentiels avancés	39
II.3 Elaboration d'un contrôleur flou pour la navigation d'un robot mobile	42
II.3.1 Navigation libre	44
II.3.1.1 Fuzzification	44
II.3.1.2 Inférence	47
II.3.1.3 Défuzzification	47
II.3.1.4 Résultats de simulation	47
II.3.2 Navigation avec évitement d'obstacles	50
II.3.2.1 Fuzzification	51
II.3.2.2 Inférence	52



II.3.2.3 Défuzzification	52
II.3.3 Stratégie de navigation	52
II.3.4 Résultats de simulation	53
II.4 Conclusion	57

## **CHAPITRE 3**

### **COMMANDE PAR MODE GLISSANT**

III.1 Introduction	59
III.2 Conception de la commande par mode glissant classique	60
III.2.1 Choix de la surface de glissement	61
III.2.2 Conditions de convergence et d'existence	62
III.2.2.1 Fonction discrète de commutation	62
III.2.2.2 Fonction de Lyapunov	63
III.2.3 Détermination de la loi de commande	63
III.2.3.1 Méthode de Philippov	63
III.2.3.2 Méthode de la commande équivalente d'Utkin	64
III.3. la commande par mode glissant basée sur la loi d'arrivée	64
III.3.1 Loi d'arrivée avec une vitesse constante	65
III.3.2 Loi d'arrivée exponentielle	65
III.3.3 Loi d'arrivée avec vitesse puissante	66
III.4. Commande par mode glissant basée sur le quasi-mode glissant	66
III.4.1. Quasi-mode glissant basé sur la fonction de saturation	66
III.4.2. Quasi-mode glissant basé sur la fonction relais	67
III.5. Mode glissant terminal (Terminal Sliding Mode)	67
III.5.1. Mode glissant terminal normal (Normal terminal sliding mode)	68
III.5.2. Mode glissant terminal rapide (Fast terminal sliding mode)	69
III.6. Conception d'un contrôleur basé sur la commande par mode glissant rapide pour le suivi de trajectoires d'un robot mobile	71
III.6.1. Commande cinématique	72
III.6.1.1. Modélisation cinématique	72
III.6.1.2. Conception du contrôleur cinématique pour le suivi de trajectoires	73
III.6.1.3. Résultats de simulation	76
III.6.2. Commande dynamique	81
III.6.2.1. Modélisation dynamique	81
III.6.2.2. Conception du contrôleur dynamique pour le suivi de trajectoires	82
III.6.2.3. Résultats de simulation	84
III.7 Conclusion	86

**CHAPITRE 4**  
**COMMANDE ROBUSTE APPLIQUEE A LA POURSUITE DE TRAJECTOIRES**  
**D'UN ROBOT MOBILE**

IV.1 Introduction	87
IV.2 Réseaux de neurones artificiels (RNA) appliqués dans la commande des systèmes	88
IV.2.1 Modèle mathématique d'un réseau neuronal	89
IV.2.2 Réseaux de neurones MLP	90
IV.2.3 Réseaux de neurones RBF	91
IV.3 Application d'une commande neuronale robuste pour la poursuite de trajectoires pour un robot mobile	92
IV.3.1. Contrôleur cinématique	92
IV.3.2. Contrôleur dynamique	93
IV.3.2.1. Contrôle en absence de perturbations	93
IV.3.2.2. Contrôle en présence de perturbations	95
IV.3.3. Résultats de simulation	97
IV.4 Conclusion	102
CONCLUSION GENERALE	103
REFERENCES BIBLIOGRAPHIQUES	106

## NOTATIONS ET SYMBOLES

*WMR* : Wheeled Mobile Robot ( Robots Mobiles à Roues)

$p$  : vecteur de coordonnées généralisées

$Q$  : Point de contact

$\phi$  : Angle d'orientation

$v_\phi$  : La vitesse angulaire

$\dot{\phi}$  : La vitesse angulaire

$P$  : Vecteur des coordonnées

$J$  : La matrice jacobéenne.

$v_1$  : La vitesse linéaire

$v_2$  : La vitesse angulaire

$\omega_\psi$  : La vitesse angulaire de la direction des roues

$v_w$  : La vitesse linéaire

$\omega_w$  : La vitesse angulaire

$r$  : Le rayon de la roue

$v$  : La vitesse du robot.

$\dot{\psi}$  : La vitesse angulaire du volant

$\dot{\theta}$  : Le vecteur des vitesses communes (variables de contrôle)

$R$  : Le rayon de courbure instantané.

$G$  : Le centre de gravité

$v_l$  : La vitesse linéaire de la roue gauche.

$v_r$  : La vitesse linéaire de la roue droite.

$v_Q$  : La vitesse du point médian  $Q$  du WMR.

$n$  : Nombre de degrés de libertés.

$E_i$  : Le moment linéaire.

$m_i$  : La masse du corps.

$S_i$  : La position du centre de gravité (CG).

$G_i$  : Le moment angulaire total

$\tau_i$  : Le couple externe total.

$I_i$  : le tenseur d'inertie.

$\rho_i$  : la densité de masse.

$dV$  : le volume d'un élément infinitésimal du  $B_i$  .

$\omega_i$  : le vecteur de vitesse angulaire.

$V_i$  : le volume.

$q_n$  : les coordonnées généralisées.

$L$  : Lagrangien.  
 $K$  : L'énergie cinétique totale.  
 $P$  : L'énergie potentielle totale.  
 $\lambda_j$  : Le multiplicateur de lagrangien.  
 $Q$  : Les forces non conservatrices dans le système.  
 $\tau$  : Le vecteur de la force généralisée externe appliqué au corps.  
 $v$  : la vitesse linéaire du centre de gravité.  
 $\omega$  : la vitesse angulaire de la rotation.  
 $M(q)$  : la matrice d'inertie symétrique définie positive  
 $V(.)$  : la matrice d'inertie symétrique définie positive.  
 $F(.)$  : la matrice de surface de friction.  
 $G(q)$  : le vecteur de gravité.  
 $\tau_d$  : Perturbations inconnues.  
 $B(q)$  : la matrice de transformation d'entrée.  
 $\lambda$  : le vecteur des forces de contrainte.  
 $U(q)$  : Potentiel des champs artificiels.  
 $U_{att}$  : potentiel attractif  
 $U_{rép}$  : potentiel répulsif  
 $U_{start}$  : potentiel du point de départ.  
 $F_{att}$  : Force attractive.  
 $F_{rep}$  : Force répulsive.  
 $p_0$  : la distance d'influence des obstacles.  
 $K_{obs}$  : le coefficient de répulsion.  
 $S$  : la surface de glissement.  
 $e_i$  : est le vecteur d'écart.  
 $V$  : la fonction de Lyapunov.  
 $Sgn$  : la fonction sign.  
 $\theta(s)$  : la fonction relais.  
 $t_s$  : le temps d'arrivée.  
 $p_e$  : L'erreur de posture.  
 $T$  : matrice de transformation.  
 $x_e$  : l'erreur sur l'axe ox.  
 $y_e$  : l'erreur sur l'axe oy.  
 $\theta_e$  : l'erreur d'orientation.  
 $W^T$  : vecteur de poids.  
 $\varepsilon$  : l'erreur de l'approximation.  
 $c_{ij}$  : les centroids.  
 $ev$  : vecteur des erreurs des vitesses.  
 $\hat{f}(z)$  : la fonction estimée.  
 $\zeta$  : l'élément robuste.

---

# **INTRODUCTION GENERALE**

---

# INTRODUCTION GENERALE

Le mot *robot* a des racines dans les langues slaves. La signification de "robota" en polonais est le travail ou le travailleur, tandis que le mot est plus archaïque en tchèque ou en langue slovène qui signifie le travail ou la corvée. L'écrivain tchèque Karel Capek a inventé le mot robot et il l'a utilisé dans les jeux de science-fiction R.U.R. (Universal Robots de Rossum). Le robot dans le jeu est une sorte d'humain artificiel; dans les terminologies modernes cyborg ou androïde serait plus approprié. La pièce a connu un énorme succès et le mot robot a été adopté par la majorité des langues du monde. Tandis que le mot robot n'a même pas 100 ans, l'idée de créatures mécaniques va très profondément dans l'histoire [1].

Le mot *mobile* a ses racines dans le mot latin de la même signification, "Mobilis." La majorité des espèces animales possèdent la capacité de locomotion. Alors que certains animaux utilisent des systèmes passifs pour le faire (ils peuvent bouger avec mouvement de l'eau ou de l'air), d'autres ont développé plus ou moins des mécanismes sophistiqués pour le mouvement actif. Certains animaux effectuent une locomotion dans l'espace 3D (nager dans l'eau, voler dans les airs), et d'autres suivent plus ou moins la surface 2D, sur l'eau ou sur le sol, tandis que certains animaux sont capables de combiner différentes façons de mouvement. Dans le contexte des robots mobiles, on est concerné par les systèmes qui peuvent se déplacer en utilisant leur appareil de locomotion. Ce dernier très souvent imite celui d'un humain ou d'un animal.

Les systèmes robots mobiles peuvent être définis comme des systèmes qui ne sont pas attachés à l'environnement et peuvent se déplacer dans un certain espace. En termes d'environnement de déplacement, ils peuvent être classés selon les types suivants:

- **Systèmes mobiles au sol :** Différents types de plates-formes mobiles peuvent être trouvés dans cette classe tels que les véhicules mobiles à roues ou

chenilles, robots à pattes (humanoïdes ou animaux imitant), ou des robots qui imitent un autre type de locomotion animale, par exemple, les serpents.

- **Systèmes mobiles aériens** : Ce groupe comprend les systèmes mobiles qui volent dans une certaine partie de l'espace (avions, hélicoptères, drones, fusées, systèmes de vol imitant les animaux; utilisés sans pilote, ils sont appelés véhicules aériens sans pilote).
- **Systèmes mobiles aquatiques et sous-marins** : Dans ce groupe, nous trouvons différents types de navires, bateaux, sous-marins, véhicules sous-marins autonomes, etc.

Les robots mobiles sont considérés comme autonomes lorsqu'ils sont capables de se mouvoir avec autonomie dans leur environnement. L'autonomie doit être garantie :

- **du point de vue énergétique**: le robot doit avoir une source d'énergie.
- **du point de vue de la décision**: le robot doit être capable de prendre certaines décisions et effectuer les actions appropriées.

En réalité, cela signifie que le système mobile prend les commandes de l'opérateur humain basé sur le niveau d'autonomie dont le système est capable.

La navigation autonome d'un robot mobile doit mettre en œuvre un certain nombre de fonctionnalités :

- Planification de mouvement** : La planification de mouvement d'un robot mobile est le calcul d'un chemin sans collision, entre une configuration initiale et une configuration finale. Dans la littérature, on trouve une synthèse de ces techniques dans [2,3].
- Localisation** : Une fois que le mouvement est planifié, le robot doit se localiser dans son environnement. La localisation est l'estimation de la position du robot par rapport à un repère fixe de l'environnement. Cette estimation de la position peut s'effectuer soit par une mesure des déplacements du robot soit par une mesure de sa position absolue dans l'environnement [4,5].
- Suivi de trajectoire** : Le suivi de trajectoire consiste à proposer des commandes qui permettent au robot d'effectuer le mouvement planifié.

Différentes techniques sont proposées pour asservir le système dans une trajectoire de référence.

**d. *Évitement d'obstacles*** : En cas de présence des obstacles dans la trajectoire planifiée, des collisions peuvent se produire lors du déplacement du robot. Ces collisions sont dues aux :

- Une localisation imparfaite,
- Un plan imprécis,
- Les obstacles ne sont pas dans le modèle de l'environnement utilisé.

Donc, des stratégies d'évitement d'obstacles doivent être mises en œuvre pour éviter la collision avec les obstacles.

**e. *Parking*** : Le parking est la phase finale de la navigation autonome. Par ailleurs, le parking s'effectue généralement «relativement» à des éléments de l'environnement, et des méthodes de mouvement référencé sur des amers doivent être utilisées.

Plusieurs approches ont été proposées pour guider le robot afin qu'il puisse accomplir une quelconque tâche. La plupart de ces approches comprennent une étape de planification de la trajectoire, effectuée sur une carte de l'environnement, avant que le robot commence la navigation. Ceci est également appelé planification globale du chemin. En d'autres termes, le robot mobile a une carte globale de l'environnement qui l'entoure, puis l'algorithme de contrôle planifie les actions pour accomplir la tâche désirée en se basant sur cette carte [6]. Les approches rapportées dans [7,8] sont de bons exemples délibératifs.

Cependant, la faiblesse de telles d'approches est qu'après avoir planifié un chemin, l'environnement ne peut plus être changé. Si cela arrive, par exemple si un obstacle n'apparaît pas sur la carte environnementale, dans ce cas la navigation libre sans collision n'est plus garantie.

D'autres approches pour éviter les obstacles reposent sur un paradigme réactif [6] dont l'hypothèse de base est que le robot n'a aucune connaissance à priori sur l'environnement de navigation, il réagit simplement selon sa perception. Par conséquent, la navigation réactive exige un système de détection adapté et aussi beaucoup moins de calcul pour travailler avec les informations locales (présence / absence d'un obstacle proche du robot). Les changements d'environnement ne sont



plus un problème, le robot perçoit les changements et réagit en conséquence. Donc, la navigation réactive est plus adaptée à l'environnement. La détection de bord [9] et les champs de potentiels [10] sont des exemples classiques de stratégies de navigation réactive, de récents exemples de navigation réactive sont rapportés dans [11] et [12].

Pour surmonter le problème de collision avec des obstacles imprévus, certaines approches délibératives [13] et [14] changent temporairement la trajectoire planifiée, réagissant ainsi à la présence de ces obstacles. Par conséquent, elles sont classées comme des approches hybrides [6].

Plusieurs contrôleurs ont été proposés pour la commande des robots mobiles avec des contraintes non holonomiques, où les deux principales approches de contrôle sont la stabilisation de posture et le suivi de trajectoires. Le but de la stabilisation de la posture est de stabiliser le robot à un point de référence, alors que le but du suivi de trajectoires est de faire suivre au robot une trajectoire de référence.

Ces dernières années, différentes techniques de contrôle ont été introduites pour contrôler la navigation du robot mobile et résoudre le problème du suivi de trajectoires. Du fait de la non-linéarité intrinsèque de la dynamique du robot mobile et des contraintes non holonomes, des architectures non linéaires comme les approches adaptatives et intelligentes [15-17], backstepping [18,19], linéarisation par rétroaction [20] et contrôle par mode glissement [21] sont proposées.

La commande par mode glissant classique est une approche de contrôle robuste et efficace pour la stabilisation des systèmes non linéaires [22,23]. La méthode traditionnelle est basée sur la définition d'une surface exponentiellement stable en fonction des états du système, utilisant la théorie de Lyapunov pour assurer que toutes les trajectoires du système en boucle fermée atteignent cette surface en un temps fini. Le principal inconvénient de la commande par mode glissant est l'effet de chattering (broutement) dans les implémentations réelles lorsque les trajectoires du système présentent des oscillations à haute fréquence autour de la surface de glissement. Les contrôleurs dans le mode glissant pour les surfaces de glissement d'ordre supérieur, permettent généralement de réduire l'effet de chattering tout en fournissant des convergences à temps fini à la surface de glissement [24,25].

Dans les nouvelles recherches de contrôle pour le suivi de trajectoires, une commande en un temps fini a été déployée au cours des dernières années [26,27]. Un

contrôleur global de suivi en temps fini pour les systèmes non holonomes est présenté dans [28,29]. L'analyse de l'équation différentielle du temps permet de calculer le temps de convergence de la surface de glissement.

Récemment, de nombreuses recherches ont été effectuées sur les applications des réseaux de neurones pour le contrôle des processus dynamiques non linéaires. Ces travaux sont soutenus par deux des capacités les plus importantes des réseaux de neurones, leur capacité à apprendre et leur bonne performance pour l'approximation des fonctions non linéaires [30].

Dans cette thèse de doctorat, on va exposer les différentes techniques qui sont proposées pour l'évitement d'obstacles et la poursuite de trajectoires. Notre contribution consiste à l'amélioration des performances des ces approches par introduction des techniques de l'intelligence artificielle telles que les réseaux de neurones et la logique floue.

Cette thèse de doctorat s'articule autour de quatre chapitres, à savoir:

Le premier chapitre est consacré à la modélisation cinématique et dynamique des différents types de robots mobiles et à leurs structures.

Le deuxième chapitre est consacré à la présentation des algorithmes de navigation avec évitement d'obstacles. On expose en premier temps la technique des champs de potentiels artificiels classique et avancée puis en deuxième temps on propose un autre algorithme d'évitement d'obstacles basé sur la logique floue. Les résultats de simulation sont présentés et commentés à la fin du chapitre.

Dans le troisième chapitre on expose les notions de la commande par mode glissant. On va présenter une commande cinématique destinée à la poursuite de trajectoires d'un robot mobile qui est basée sur le mode glissant terminal rapide et une commande dynamique basée sur un contrôleur PID en l'absence des perturbations. Les résultats de simulation sont aussi présentés et interprétés.

Une commande robuste en présence des perturbations est introduite dans le quatrième chapitre pour améliorer les résultats obtenus dans le troisième chapitre. La stratégie de commande dans ce chapitre est basée sur deux contrôleurs.

- Un contrôleur cinématique basé sur la commande par mode glissant terminal rapide (GFSM), pour déterminer les vitesses adéquates pour la stabilité du système en temps fini.
- Un contrôleur dynamique combinant deux techniques de contrôle, le GFSM pour la stabilisation des erreurs de vitesse, et un contrôleur neuronal afin d'approximer la fonction non linéaire et traiter les perturbations.

Ce travail est clôturé par une conclusion générale à travers laquelle, on analyse les principaux résultats obtenus et on proposera des perspectives à envisager comme futur travail.

---

**CHAPITRE 1**  
**MODELISATION DES ROBOTS MOBILES**

---

# CHAPITRE 1

## MODELISATION DES ROBOTS MOBILES

### I.1. Introduction

La modélisation d'une plate-forme de robot mobile consiste en une modélisation cinématique et dynamique en plus de la modélisation des actionneurs du système. La modélisation cinématique traite des relations géométriques qui régissent le système et étudie les mathématiques du mouvement sans tenir compte des forces qui les affectent. D'autre part, la modélisation dynamique est l'étude du mouvement dans lequel les forces et les énergies sont modélisées et incluses. La modélisation de l'actionneur est nécessaire pour trouver la relation entre le signal de contrôle et l'entrée du système mécanique. Chaque partie de la modélisation de ce système sera expliquée séparément dans ce chapitre.

### I.2. Modélisation cinématique

La cinématique du robot traite la configuration des robots dans leur espace de travail, des relations entre leurs paramètres géométriques et des contraintes imposées dans leurs trajectoires. Les équations cinématiques dépendent de la structure géométrique du robot. Par exemple, un robot fixe peut avoir un repère cartésien, cylindrique, sphérique ou une structure articulée et un robot mobile peut avoir une, deux, trois roues ou plus avec ou sans contraintes dans leur mouvement [31-34]. L'étude de la cinématique est une condition préalable fondamentale pour l'étude de la dynamique, des caractéristiques de stabilité et du contrôle du robot.

Le développement de nouvelles structures cinématiques robotisées spécifiques est un sujet de recherche toujours en cours, notamment la construction de robots qui peuvent effectuer plusieurs tâches sophistiquées et complexes dans les applications industrielles [35, 36].

Dans cette partie on va présenter la modélisation cinématique de quelques robots mobiles non holonomes, entre autres l'unicycle, le tricycle et le robot différentiel.

### I.2.1 Robots mobiles de type unicycle

#### *Description*

On désigne par *unicycle* un robot actionné par deux roues indépendantes et possédant éventuellement un certain nombre de roues folles assurant sa stabilité. Le schéma des robots de type unicycle est donné à la figure 1.1. On a omis les roues folles qui n'interviennent pas dans la cinématique, dans la mesure où elles sont judicieusement placées.

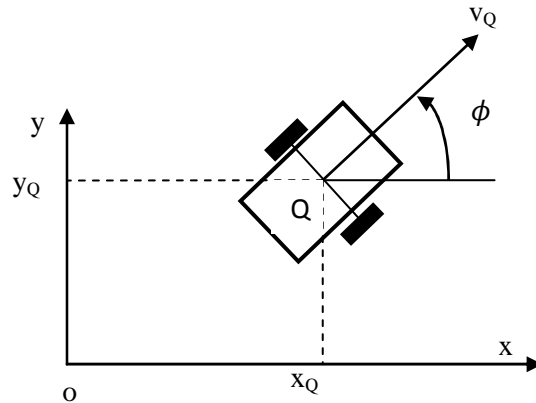


Figure 1.1: Robot mobile de type unicycle

Ce type de robot est très répandu en raison de sa simplicité de construction et de ses propriétés cinématiques intéressantes.

#### *Modélisation cinématique*

L'unicycle a un modèle cinématique qui est utilisé comme une référence de base pour de nombreux types des robots mobiles à roues non holonomes. Pour cette raison, ce modèle a attiré beaucoup d'attention de la part des chercheurs travaillant sur les systèmes non linéaires.

Sa configuration est décrite par un vecteur de coordonnées généralisées:

$$p = [x_Q, y_Q, \phi] \quad (1.1)$$

C'est-à-dire les coordonnées de position du point de contact  $Q$  dans le repère orthonormé  $oxy$  et son angle d'orientation  $\phi$  par rapport à l'axe des abscisses  $x$ . La vitesse linéaire de la roue est  $v_Q$  et sa vitesse angulaire par rapport à son axe de rotation instantané est  $v_\phi$ :

$$\dot{x}_Q = v_Q \cos \phi$$

$$\dot{y}_Q = v_Q \sin \phi \quad (1.2)$$

$$\dot{\phi} = v_\phi$$

En éliminant  $v_Q$  des deux premières équations, nous trouvons la contrainte de non holonomie :

$$\dot{x}_Q \sin\phi + \dot{y}_Q \cos\phi = 0 \quad (1.3)$$

En utilisant la notation  $v_1=v_Q$  et  $v_2=v_\phi$ , pour simplifier, le modèle cinématique (1.2) de l'unicycle peut être écrit comme:

$$\dot{p} = \begin{bmatrix} \cos\phi \\ \sin\phi \\ 0 \end{bmatrix} v_1 + \begin{bmatrix} 0 \\ 0 \\ 1 \end{bmatrix} v_2, \dot{p} = \begin{bmatrix} \dot{x}_Q \\ \dot{y}_Q \\ \dot{\phi} \end{bmatrix} \quad (1.4)$$

où :

$$\dot{p} = J\dot{q}, \dot{q} = v_1, v_2^T, \quad J = \begin{bmatrix} \cos\phi & 0 \\ \sin\phi & 0 \\ 0 & 1 \end{bmatrix}.$$

où J est la matrice jacobéenne.

La vitesse linéaire  $v_1=v_Q$  et la vitesse angulaire  $v_2=v_\phi$  sont supposées être les variables d'action du système.

### I.2.2 Robots mobiles de type tricycle

Considérons tout d'abord le cas du tricycle, représenté à la figure 1.2. Ce robot est constitué de deux roues fixes de même axe et d'une roue centrée orientable placée sur l'axe longitudinal du robot. Le mouvement est conféré au robot par deux actions : la vitesse longitudinale et l'orientation de la roue orientable. De ce point de vue, il est donc très proche d'une voiture. L'intérêt pratique de ce type de robot (peu stable) reste limité.

Le mouvement de ce robot mobile est contrôlé par la vitesse angulaire de la direction des roues  $\omega_\psi$  et sa vitesse linéaire  $v_w$  (ou sa vitesse  $\omega_w = \dot{\theta}_w = v_w/r$ , où  $r$  est le rayon de la roue).

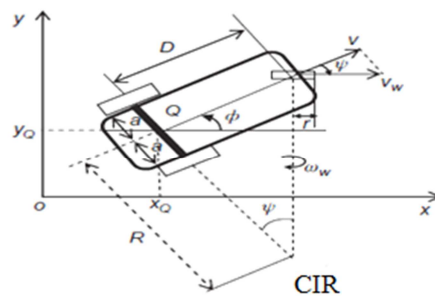


Figure 1.2: Robot mobile de type tricycle

### Modélisation cinématique

L'angle d'orientation et la vitesse angulaire sont respectivement  $\phi$  et  $\dot{\phi}$ . C'est supposé que le robot a son point de guidage  $Q$  à l'arrière de la roue motrice.

Le vecteur d'état du mouvement est :

$$p = x_Q, y_Q, \phi, \psi \quad (1.5)$$

Sachant que les variables cinématiques sont :

- Vitesse linéaire du volant :

$$v_w = r\dot{\theta}_w \quad (1.6)$$

- Vitesse du robot :

$$v = v_w \cos \psi = r(\cos \psi)\dot{\theta}_w \quad (1.7)$$

- Vitesse d'orientation du robot :

$$\dot{\phi} = \left(\frac{1}{D}\right)v_w \sin \psi \quad (1.8)$$

- Vitesse angulaire du volant :

$$\dot{\psi} = \omega_w \quad (1.9)$$

Utilisons les relations précédentes on trouve que :

$$\dot{p} = \begin{bmatrix} \dot{x}_Q \\ \dot{y}_Q \\ \dot{\phi} \\ \dot{\psi} \end{bmatrix} = \begin{bmatrix} v \cos \phi \\ v \sin \phi \\ \dot{\phi} \\ \dot{\psi} \end{bmatrix} = \begin{bmatrix} r \cos \psi \cos \phi \\ r \cos \psi \sin \phi \\ \left(\frac{r}{D}\right) \sin \psi \\ 0 \end{bmatrix} \dot{\theta}_w + \begin{bmatrix} 0 \\ 0 \\ 0 \\ 1 \end{bmatrix} \dot{\psi} = J\dot{\theta} \quad (1.10)$$

Ou  $\dot{\theta} = [\dot{\theta}_w, \dot{\psi}]^T$  est le vecteur des vitesses communes (variables de contrôle), et :

$$J = \begin{bmatrix} r \cos \psi \cos \phi & 0 \\ r \cos \psi \sin \phi & 0 \\ \left(\frac{r}{D}\right) \sin \psi & 0 \\ 0 & 1 \end{bmatrix} \quad (1.11)$$

$J$  est la matrice Jacobienne. Ce Jacobien est encore non inversible, mais nous pouvons trouver les équations cinématiques inverses en utilisant directement les relations:



$$\frac{\dot{\phi}}{v} = \left(\frac{1}{D}\right) \text{tg} \varphi \quad \text{où} \quad \psi = \text{arctg} \left( D \frac{\dot{\phi}}{v} \right) \quad (1.12)$$

et

$$\dot{\theta}_w = \frac{v_w}{r} = \frac{1}{r} \sqrt{v^2 + (D\dot{\phi})^2} \quad (1.13)$$

Le rayon de courbure instantané  $R$  est donné par (figure 1.2):

$$R = D \text{tg} \left( \frac{\pi}{2} - \psi(t) \right) \quad (1.14)$$

A partir de l'équation (1.10) on voit que le tricycle est encore un système affine sans dérive à deux entrées avec des champs de vecteurs:

$$g_1 = \begin{bmatrix} r \cos \psi \cos \phi \\ r \cos \psi \sin \phi \\ \left(\frac{r}{D}\right) \sin \psi \\ 0 \end{bmatrix}, \quad g_2 = \begin{bmatrix} 0 \\ 0 \\ 0 \\ 1 \end{bmatrix} \quad (1.15)$$

qui permettent le mouvement du volant  $\dot{\theta}_w$  et le mouvement de l'angle de braquage  $\psi$  respectivement.

### 1.2.3 Robot mobile différentiel

Le robot Pioneer présenté sur figure 1.3 est un exemple de robot mobile différentiel.



Figure 1.3: Robot mobile de type Pioneer

#### *Modélisation cinématique*

Les paramètres géométriques et cinématiques de ce robot sont illustrés sur la figure 1.4.

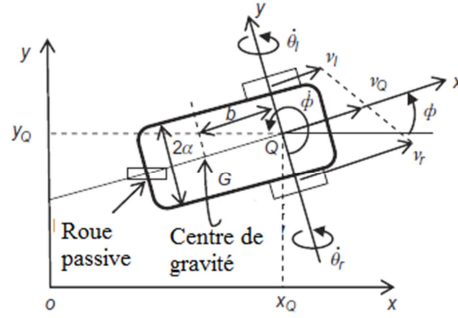


Figure 1.4: Paramétrage géométrique d'un robot mobile différentiel

Le vecteur de posture (position / orientation) du WDR et sa vitesse sont respectivement:

$$p = \begin{bmatrix} x_Q \\ y_Q \\ \phi \end{bmatrix}, \dot{p} = \begin{bmatrix} \dot{x}_Q \\ \dot{y}_Q \\ \dot{\phi} \end{bmatrix} \quad (1.16)$$

Les positions angulaires et les vitesses des roues gauche et droite sont  $\{\theta_l, \dot{\theta}_l\}$ ,  $\{\theta_r, \dot{\theta}_r\}$  respectivement.

Les hypothèses suivantes sont considérées:

- Les roues roulent sans glissement
- L'axe de guidage (direction) est perpendiculaire au plan Oxy
- Le point  $Q$  coïncide avec le centre de gravité  $G$ .

Soit  $v_l$  et  $v_r$  la vitesse linéaire de la roue gauche et droite respectivement, et  $v_Q$  la vitesse du point médian  $Q$  du WDR. Alors, à partir de la figure 1.4, nous obtenons:

$$v_r = v_Q + a\dot{\phi} \quad (1.17)$$

$$v_l = v_Q - a\dot{\phi} \quad (1.18)$$

Additionner et soustraire  $v_r$  et  $v_l$  nous obtenons :

$$v_Q = 1/2(v_r + v_l), 2a\dot{\phi} = v_r - v_l \quad (1.19)$$

En raison de l'hypothèse de non dérapage, nous avons :

$$v_r = r\dot{\theta}_r \text{ et } v_l = r\dot{\theta}_l \quad (1.20)$$

Comme le cas d'un unicycle où :  $\dot{x}_Q = v_Q \cos\phi$ ,  $\dot{y}_Q = v_Q \sin\phi$ , alors le modèle cinématique de ce WDR est décrit par les équations suivantes:

$$\dot{x}_Q = (r/2)(\dot{\theta}_r \cos\phi + \dot{\theta}_l \cos\phi) \quad (1.21)$$

$$\dot{y}_Q = (r/2)(\dot{\theta}_r \sin\phi + \dot{\theta}_l \sin\phi) \quad (1.22)$$

$$\dot{\phi} = (r/2a)(\dot{\theta}_r - \dot{\theta}_l) \quad (1.23)$$

Le modèle cinématique peut être écrit sous la forme matricielle suivante:

$$\dot{P} = \begin{bmatrix} (r/2)\cos\phi \\ (r/2)\sin\phi \\ r/2a \end{bmatrix} \dot{\theta}_r + \begin{bmatrix} (r/2)\cos\phi \\ (r/2)\sin\phi \\ -r/2a \end{bmatrix} \dot{\theta}_l \quad (1.24)$$

où :

$$\dot{p} = J\dot{q}$$

et :

$$\dot{p} = \begin{bmatrix} \dot{x}_Q \\ \dot{y}_Q \\ \dot{\phi} \end{bmatrix}, \dot{q} = \begin{bmatrix} \dot{\theta}_r \\ \dot{\theta}_l \end{bmatrix}$$

J est la matrice jacobéenne :

$$J = \begin{bmatrix} (r/2)\cos\phi & (r/2)\cos\phi \\ (r/2)\sin\phi & (r/2)\sin\phi \\ r/2a & -r/2a \end{bmatrix} \quad (1.25)$$

et :

$$g_1 = \begin{bmatrix} (r/2)\cos\phi \\ (r/2)\sin\phi \\ r/2a \end{bmatrix} \quad (1.26)$$

$$g_2 = \begin{bmatrix} (r/2)\cos\phi \\ (r/2)\sin\phi \\ -r/2a \end{bmatrix} \quad (1.27)$$

$g_1$  permet la rotation de la roue droite, et  $g_2$  permet la rotation de la roue gauche. Éliminant  $v_Q$  dans l'équation (1.2) on obtient comme d'habitude la contrainte non holonomique :

$$-\dot{x}_Q \sin\phi + \dot{y}_Q \cos\phi = 0 \quad (1.28)$$

### I.3. Modélisation dynamique

La modélisation dynamique est effectuée en utilisant les lois de la mécanique qui sont basées sur les trois éléments physiques: l'inertie, l'élasticité, et les frottements qui sont présents dans tout système mécanique réel tel que le robot mobile.

La dynamique des robots est un domaine difficile en soi et a attiré l'attention des chercheurs et des ingénieurs au fil des ans. La plupart des robots mobiles, employés dans la pratique, utilisent des roues conventionnelles et sont sujets à des contraintes qui nécessitent un traitement particulier. La stabilité délicate et les problèmes de contrôle, qui doivent souvent être confrontés à la conception d'un robot mobile, sont dus à l'existence de glissement longitudinal et latéral dans le mouvement du robot mobile à roues [37-39].

#### I.3.1. Modélisation dynamique générale du robot mobile

La modélisation dynamique des robots traite de la dérivation des équations dynamiques de mouvement du robot. Cela peut être fait en utilisant les deux méthodes suivantes:

- Newton\_Euler
- Lagrange

La complexité de la méthode de Newton-Euler est  $O(n)$ , alors que la complexité de la méthode de Lagrange ne peut être réduite qu'à  $O(n^3)$ , où  $n$  est le nombre de degrés de libertés.

Comme la modélisation cinématique, la dynamique du robot mobile est subdivisée en :

- Dynamique directe
- Dynamique inverse

La dynamique directe fournit les équations dynamiques qui décrivent la dynamique des réponses du robot à des forces/couples donnés  $\tau_1, \tau_2, \dots, \tau_m$  qui sont exercés par les moteurs.

La dynamique inverse fournit les forces / couples nécessaires pour obtenir les trajectoires souhaitées par le robot.

La modélisation dynamique directe et inverse est illustrée à la figure 1.5.

Dans le modèle dynamique inverse, les entrées sont les trajectoires désirées, et les sorties sont les couples moteur.

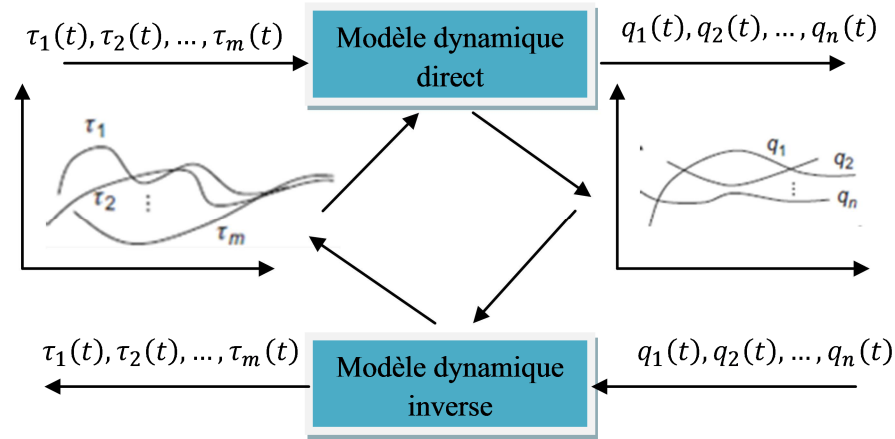


Figure 1.5. Modélisation dynamique directe et inverse

### I.3.2 L'approche de Newton-Euler

Ce modèle est dérivé de l'application directe des équations de Newton-Euler pour le mouvement de translation et de rotation. Considérons l'objet  $B_i$  (robot WMR) de la figure 1.6 auquel une force totale  $f_i$  est appliquée à son centre de gravité.

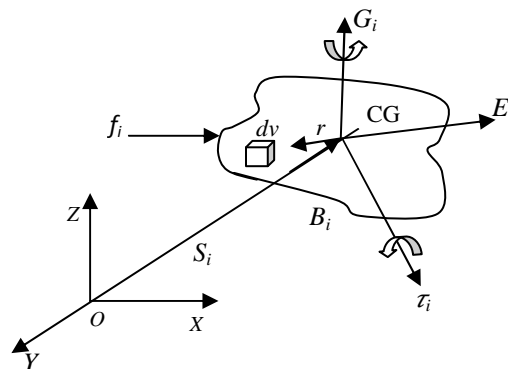


Figure 1.6. Référentiel inertiel Oxyz d'un corps solide.

Donc le mouvement de translation est donné par :

$$(dE_i) / dt = f_i \tag{1.29}$$

Dans ce cas  $E_i$  est le moment linéaire donné par :

$$E_i = m_i \dot{s}_i \tag{1.30}$$

où  $m_i$  est la masse du corps et  $s_i$  est la position du centre de gravité (CG) par rapport aux coordonnées inertiels Oxyz. En supposant que  $m_i$  est constant, alors les équations (1.29) et (1.30) donnent:

$$m_i \ddot{s}_i = F_i \tag{1.31}$$

qui est le modèle dynamique général de translation.

Le mouvement de rotation de  $B_i$  est décrit par:

$$(dG_i)/dt = \tau_i \quad (1.32)$$

Où  $G_i$  est le moment angulaire total de  $B_i$  par rapport au centre de gravité, et  $\tau_i$  est le couple externe total qui produit le mouvement de rotation du corps. Le moment total  $G_i$  est donné par:

$$G_i = I_i \omega_i \quad (1.33)$$

Alors  $I_i$  est le tenseur d'inertie donné par l'intégrale de volume:

$$I_i = \int [r^T r I_3 - r r^T] \rho_i dV \quad (1.34)$$

où  $\rho_i$  est la densité de masse de  $B_i$ ,  $dV$  est le volume d'un élément infinitésimal du  $B_i$  se trouvant à la position  $r$  par rapport au centre de gravité,  $\omega_i$  est le vecteur de vitesse angulaire à propos de l'axe inertiel passant par le centre de gravité,  $I_3$  est la matrice de  $3 \times 3$  unités, et  $V_i$  est le volume de  $B_i$ .

### I.3.3 Application de l'approche de Lagrange :

Le modèle dynamique général de Lagrange d'un corps solide est décrit par :

$$\frac{d}{dt} \left( \frac{\partial L}{\partial \dot{q}_i} \right) - \frac{\partial L}{\partial q_i} = \sum_{j=1}^n \lambda_j a_{ji} + Q_i, i=1,2,\dots,n \quad (1.35)$$

Et :

$$\sum_{i=1}^n a_{ij} dq_i + a_{jt} dt = 0, j=1,2,\dots,m \quad (1.36)$$

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

$L = K - P$  C'est le Lagrangien qui est la différence entre l'énergie cinétique et l'énergie potentielle.

$\lambda_j$  Est le multiplicateur de lagrangien.

$Q_i$  les forces non conservatrices dans le système.

$\tau$  est le vecteur de la force généralisée externe appliqué au corps (c'est-à-dire, force pour le mouvement de translation, et couple pour rotation motion), et  $L$  est la fonction lagrangienne définie par:

$$L = K - P \quad (1.37)$$

$K$  est l'énergie cinétique totale,  $P$  est l'énergie potentielle totale du corps solide qui sont données par :

$$K = K_1 + K_2 + \dots + K_n \quad (1.38)$$

$$P = P_1 + P_2 + \dots + P_n \quad (1.39)$$

L'énergie cinétique  $K$  du corps est égale à:

$$K = \frac{1}{2} m \dot{v}^T v + \frac{1}{2} \omega^T I \omega \quad (1.40)$$

Où  $v$  est la vitesse linéaire du centre de gravité,  $\omega$  est la vitesse angulaire de la rotation,  $m$  est la masse, et  $I$  le tenseur d'inertie du corps.

La première étape dans la recherche du modèle dynamique est de trouver l'énergie cinétique et potentielle du système. L'énergie potentielle du système est nulle car le mouvement est limité au sol. La fonction d'énergie cinétique du robot peut être dérivée en fonction des vitesses indiquées sur la figure 1.7:

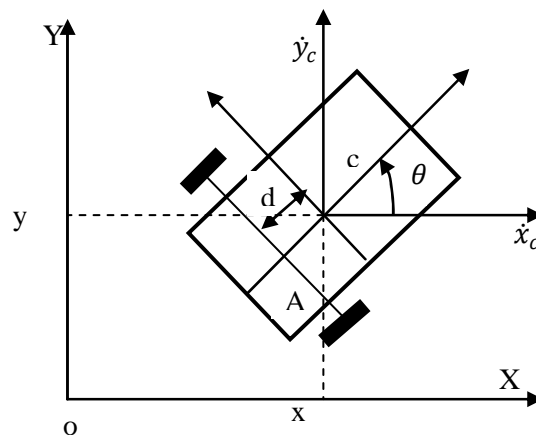


Figure 1.7. Représentation des vitesses d'un robot mobile différentiel.

D'abord il faut trouver les vitesses des points d et c :

$$x_c = x_A + d \cos \theta \quad (1.41)$$

$$y_c = y_A + d \sin \theta \quad (1.42)$$

Alors :

$$\dot{x}_c = \dot{x}_A - d \dot{\theta} \sin \theta \quad (1.43)$$

$$\dot{y}_c = \dot{y}_A + d \dot{\theta} \cos \theta \quad (1.44)$$

La vitesse de centre du robot est :

$$v_A = \dot{x}_c \vec{i} + \dot{y}_c \vec{j} + d \dot{\theta} \sin \theta \vec{i} - d \dot{\theta} \cos \theta \vec{j} \quad (1.45)$$

$$v_A = (\dot{x}_c + d \dot{\theta} \sin \theta) \vec{i} + (\dot{y}_c - d \dot{\theta} \cos \theta) \vec{j} \quad (1.46)$$

L'énergie cinétique totale est la somme de l'énergie cinétique de translation et cinétique de rotation :

$$K = \frac{1}{2} m v_A^2 + \frac{1}{2} I_A \dot{\theta}^2 \quad (1.47)$$

$$K = \frac{1}{2} m \left[ (\dot{x}_c + d \dot{\theta} \sin \theta)^2 + (\dot{y}_c - d \dot{\theta} \cos \theta)^2 \right]$$

$$= \frac{1}{2} m \left[ \dot{x}_c^2 + 2 \dot{x}_c d \dot{\theta} \sin \theta + d^2 \dot{\theta}^2 \sin^2 \theta + \dot{y}_c^2 - 2 \dot{y}_c d \dot{\theta} \cos \theta + d^2 \dot{\theta}^2 \cos^2 \theta \right] + \frac{1}{2} I_A \dot{\theta}^2$$

$$K = \frac{1}{2} m \dot{x}_c^2 + \frac{1}{2} m \dot{y}_c^2 + m \dot{x}_c d \dot{\theta} \sin \theta - m \dot{y}_c d \dot{\theta} \cos \theta + \frac{1}{2} m d^2 \dot{\theta}^2 + \frac{1}{2} I_A \dot{\theta}^2 \quad (1.48)$$

Du théorème de l'axe parallèle, nous pouvons dire que:

$$I_A = I_C + m d^2 \quad (1.49)$$

Alors l'énergie cinétique du système sera :

$$T = \frac{1}{2} m \dot{x}_c^2 + \frac{1}{2} m \dot{y}_c^2 + \frac{1}{2} I_C \dot{\theta}^2 + m d^2 \dot{\theta}^2 + m \dot{x}_c d \dot{\theta} \sin \theta - m \dot{y}_c d \dot{\theta} \cos \theta \quad (1.50)$$

A partir de l'énergie cinétique du système et sachant que l'énergie potentielle égale à zéro le lagrangien devient :

$$L = \frac{1}{2} m \dot{x}_c^2 + \frac{1}{2} m \dot{y}_c^2 + \frac{1}{2} I_C \dot{\theta}^2 + m d^2 \dot{\theta}^2 + m \dot{x}_c d \dot{\theta} \sin \theta - m \dot{y}_c d \dot{\theta} \cos \theta \quad (1.51)$$



Les coordonnées généralisées utilisées dans la formulation de Lagrange sont :

$$q = [x_c, y_c, \theta] \quad (1.52)$$

$$\dot{q} = [\dot{x}_c, \dot{y}_c, \dot{\theta}]$$

Basant sur les dérivées de Lagrange, on trouve :

$$\frac{\partial L}{\partial \dot{x}_c} = m\dot{x}_c + md\dot{\theta}\sin\theta \quad (1.53)$$

$$\frac{d}{dt} \left( \frac{\partial L}{\partial \dot{x}_c} \right) = m\ddot{x}_c + md\ddot{\theta}\sin\theta + md\dot{\theta}^2\cos\theta \quad (1.54)$$

$$\frac{\partial L}{\partial \dot{y}_c} = m\dot{y}_c - md\dot{\theta}\cos\theta \quad (1.55)$$

$$\frac{d}{dt} \left( \frac{\partial L}{\partial \dot{y}_c} \right) = m\ddot{y}_c - md\ddot{\theta}\cos\theta + md\dot{\theta}^2\sin\theta \quad (1.56)$$

$$\frac{\partial L}{\partial \dot{\theta}} = I_c\dot{\theta} + 2md^2\dot{\theta} + m\dot{x}_c d\sin\theta - m\dot{y}_c d\cos\theta \quad (1.57)$$

$$\frac{d}{dt} \left( \frac{\partial L}{\partial \dot{\theta}} \right) = I_c\ddot{\theta} + 2md^2\ddot{\theta} + m\ddot{x}_c d\sin\theta + m\dot{x}_c \dot{\theta}\cos\theta - m\ddot{y}_c d\cos\theta + m\dot{y}_c \dot{\theta}\sin\theta \quad (1.58)$$

Et le reste des dérivées sont:

$$\frac{\partial L}{\partial x_c} = 0, \frac{\partial L}{\partial y_c} = 0$$

$$\frac{\partial L}{\partial \theta} = m\dot{x}_c d\dot{\theta}\cos\theta + m\dot{y}_c d\dot{\theta}\sin\theta \quad (1.59)$$

A partir des termes précédents et l'équation de Lagrange on trouve :

$$m\ddot{x}_c + md\ddot{\theta}\sin\theta + md\dot{\theta}^2\cos\theta = F_x + C_1 \quad (1.60)$$

$$m\ddot{y}_c - md\ddot{\theta}\cos\theta + md\dot{\theta}^2\sin\theta = F_y + C_2 \quad (1.61)$$

$$(I_c + 2md^2)\ddot{\theta} + m\ddot{x}_c d\sin\theta + m\dot{x}_c \dot{\theta}\cos\theta - m\ddot{y}_c d\cos\theta + m\dot{y}_c \dot{\theta}\sin\theta - m\dot{x}_c d\dot{\theta}\cos\theta -$$

$$m\dot{y}_c d\dot{\theta}\sin\theta = \tau + C_3 \quad (1.62)$$

Simplifiant l'équation (1.62) on trouve :

$$(I_c + 2md^2)\ddot{\theta} + md\ddot{x}_c \sin\theta - md\dot{y}_c \cos\theta = \tau + C_3 \quad (1.63)$$

ou:

$F_x$  la force de l'actionneur dans la direction  $x$

$F_y$  la force de l'actionneur dans la direction  $y$

$\tau$  est le couple de rotation de l'actionneur sur le robot

$C_x, C_y, C_\theta$  sont les forces de contrainte dans les directions  $x, y$  et  $\theta$ .

Représentant les équations ci-dessus sous la forme matricielle nous avons:

$$\begin{bmatrix} m & 0 & mdsin\theta \\ 0 & m & -mdcos\theta \\ mdsin\theta & -mdcos\theta & I_c + 2md^2 \end{bmatrix} \ddot{q} + \begin{bmatrix} md\dot{\theta}^2 cos\theta \\ md\dot{\theta}^2 sin\theta \\ 0 \end{bmatrix} = \begin{bmatrix} F_x \\ F_y \\ \tau \end{bmatrix} + \begin{bmatrix} C_x \\ C_y \\ C_\theta \end{bmatrix} \quad (1.64)$$

On considère que les moteurs des roues gauche et droites ne sont pas les mêmes donc :

$$F_x = (F_r + F_l) \cos\theta \quad (1.65)$$

$$F_y = (F_r - F_l) \sin\theta \quad (1.66)$$

$$\tau = (F_r - F_l) L \quad (1.67)$$

Sachant que  $F_r$  est la force de l'actuateur de la roue droite et  $F_l$  est la force de l'actuateur de la roue gauche.

$L$  est la longueur de l'axe des roues.

et on a :

$$F_l = \frac{\tau_r}{R} \quad (1.68)$$

$$F_r = \frac{\tau_l}{R} \quad (1.69)$$

Et  $R$  est le rayon des roues.

$$F_x = \left( \frac{\tau_r}{R} + \frac{\tau_l}{R} \right) \cos\theta = \frac{\cos\theta}{R} (\tau_r + \tau_l) \quad (1.70)$$

$$F_y = \left( \frac{\tau_r}{R} + \frac{\tau_l}{R} \right) \sin\theta = \frac{\sin\theta}{R} (\tau_r + \tau_l) \quad (1.71)$$

$$\tau = (F_r - F_l)L = \frac{L}{R} (\tau_r - \tau_l) \quad (1.72)$$

A partir des équations ci-dessus, nous pouvons écrire la matrice de force d'entrée pour l'équation dynamique du système comme suit:

$$\begin{bmatrix} F_x \\ F_y \\ \tau \end{bmatrix} = \begin{bmatrix} \frac{\cos\theta}{R} (\tau_r + \tau_l) \\ \frac{\sin\theta}{R} (\tau_r + \tau_l) \\ \frac{L}{R} (\tau_r - \tau_l) \end{bmatrix} = \frac{1}{R} \begin{bmatrix} \cos\theta & \cos\theta \\ \sin\theta & \sin\theta \\ L & -L \end{bmatrix} \begin{bmatrix} \tau_r \\ \tau_l \end{bmatrix} \quad (1.73)$$

Pour suivre la modélisation dynamique du système, il faut prendre en compte les contraintes cinématiques non holonomiques

### *Les contraintes cinématiques non holonomiques*

Les hypothèses suivantes concernant le mouvement de la roue provoquent les contraintes cinématiques du robot:

- Mouvement sur un plan horizontal
- Point de contact entre les roues et le sol
- Les roues ne sont pas déformables
- Roulement pur qui signifie que nous avons le centre instantané de la vitesse nulle au point de contact des roues et du sol.
- Pas de glissement
- Pas de friction pour la rotation autour des points de contact
- Axes de direction sont orthogonaux à la surface
- Les roues sont reliées par un cadre rigide (châssis)

Considérant les hypothèses ci-dessus concernant le mouvement de la roue, le robot aura une sorte de contrainte spéciale appelée contrainte non holonomique. Une contrainte non holonomique est une contrainte sur les vitesses réalisables d'un corps. Dans le cas du robot

mobile à roues différentiels, cela peut simplement signifier que le robot peut se déplacer dans certaines directions (vers l'avant et vers l'arrière) mais pas d'autres (vers le côté).

Considérant un robot mobile différentiel, Les équations des contraintes non holonomiques sont les suivantes:

- Aucune contrainte de glissement latéral:

$$\dot{y}_c \cos\theta - \dot{x}_c \sin\theta - \dot{\theta}d = 0 \quad (1.74)$$

$\dot{y}_c$  et  $\dot{x}_c$  sont les composantes de la vitesse du robot dans le cadre inertiel. Cette contrainte signifie que la vitesse du point central du robot sera dans la direction de l'axe de symétrie et que le mouvement dans le plan orthogonal sera nul.

- Contrainte de roulement pure:

$$\dot{x}_c \cos\theta + \dot{y}_c \sin\theta + L\dot{\theta} = R\dot{\phi}_r \quad (1.75)$$

$$\dot{x}_c \cos\theta + \dot{y}_c \sin\theta - L\dot{\theta} = R\dot{\phi}_l \quad (1.76)$$

Avec :

$\dot{\phi}_r$  : La vitesse de rotation de la roue droite.

$\dot{\phi}_l$  : La vitesse de rotation de la roue gauche.

Cette contrainte montre que les roues motrices ne glissent pas. Les trois contraintes peuvent être écrites sous la forme suivante:

$$A(q)\dot{q} = 0 \quad (1.77)$$

Ou:

$$A(q) = \begin{bmatrix} -\sin\theta & \cos\theta & a & 0 & 0 \\ \cos\theta & \sin\theta & L & -R & 0 \\ \cos\theta & \sin\theta & -L & 0 & -R \end{bmatrix}$$

$$\dot{q} = \begin{bmatrix} \dot{x}_c \\ \dot{y}_c \\ \dot{\theta} \\ \dot{\phi}_r \\ \dot{\phi}_l \end{bmatrix}$$

Selon les équations de contraintes non holonomique, les forces de contrainte deviendront:

$$C_X = m(\dot{x}_c \cos\theta + \dot{y}_c \sin\theta)\dot{\theta} \sin\theta \quad (1.78)$$

$$C_Y = -m(\dot{x}_c \cos\theta + \dot{y}_c \sin\theta)\dot{\theta} \cos\theta \quad (1.79)$$

$$C_\theta = ma(\dot{x}_c \cos\theta + \dot{y}_c \sin\theta)\dot{\theta} \quad (1.80)$$

La représentation matricielle des forces de contrainte ci-dessus est:

$$\begin{bmatrix} C_X \\ C_Y \\ C_\theta \end{bmatrix} = \begin{bmatrix} m(\dot{x}_c \cos\theta + \dot{y}_c \sin\theta)\dot{\theta} \sin\theta \\ -m(\dot{x}_c \cos\theta + \dot{y}_c \sin\theta)\dot{\theta} \cos\theta \\ ma(\dot{x}_c \cos\theta + \dot{y}_c \sin\theta)\dot{\theta} \end{bmatrix} = A^T(q)\lambda \quad (1.81)$$

$$A^T(q) = \begin{bmatrix} -\sin\theta \\ \cos\theta \\ -d \end{bmatrix}$$

$$\lambda = -m(\dot{x}_c \cos\theta + \dot{y}_c \sin\theta)\dot{\theta} \quad (1.82)$$

A partir des équations précédentes ci-dessus, la forme complète de l'équation dynamique du robot mobile différentiel est la suivante:

$$\begin{bmatrix} m & 0 & mdsin\theta \\ 0 & m & -mdcos\theta \\ mdsin\theta & -mdcos\theta & I_c + 2md^2 \end{bmatrix} \ddot{q} + \begin{bmatrix} md\dot{\theta}^2 \cos\theta \\ md\dot{\theta}^2 \sin\theta \\ 0 \end{bmatrix} = \frac{1}{R} \begin{bmatrix} \cos\theta & \cos\theta \\ \sin\theta & \sin\theta \\ L & -L \end{bmatrix} \begin{bmatrix} \tau_r \\ \tau_l \end{bmatrix} + \begin{bmatrix} m(\dot{x}_c \cos\theta + \dot{y}_c \sin\theta)\dot{\theta} \sin\theta \\ -m(\dot{x}_c \cos\theta + \dot{y}_c \sin\theta)\dot{\theta} \cos\theta \\ md(\dot{x}_c \cos\theta + \dot{y}_c \sin\theta)\dot{\theta} \end{bmatrix} \quad (1.83)$$

Un robot mobile ayant un espace de configuration à  $n$  dimensions avec des coordonnées généralisées  $(q_1, q_2, \dots, q_n)$  et soumises à des contraintes  $m$  peut être décrit par l'équation dynamique générale suivante:

$$M(q)\ddot{q} + V(q, \dot{q}) + F(\dot{q}) + G(q) + \tau_d = B(q)\tau - A^T(q)\lambda \quad (1.84)$$

$M(q)$  est la matrice d'inertie symétrique définie positive

$V(q, \dot{q})$  est la matrice de coriolis

$F(\dot{q})$  est la matrice de surface de friction

$G(q)$  est le vecteur de gravité

$\tau_d$  Perturbations inconnues bornées identifiées, y compris les dynamiques non structurées et non structurées.

$B(q)$  est la matrice de transformation d'entrée

$\tau$  est le vecteur d'entrée (couples)

$A^T$  est la matrice associée aux contraintes

$\lambda$  est le vecteur des forces de contrainte

En comparant l'équation (1.83) et l'équation générale de la dynamique du robot (1.84), nous obtenons les paramètres suivants pour le système de robot mobile à entraînement différentiel:

$$M(q) = \begin{bmatrix} m & 0 & mdsin\theta \\ 0 & m & -mdcos\theta \\ mdsin\theta & -mdcos\theta & I_c + 2md^2 \end{bmatrix}$$

$$V(q, \dot{q}) = \begin{bmatrix} md\dot{\theta}^2 cos\theta \\ md\dot{\theta}^2 sin\theta \\ 0 \end{bmatrix}$$

$$\tau = \begin{bmatrix} \tau_r \\ \tau_l \end{bmatrix}$$

$$A^T(q) = \begin{bmatrix} -\sin\theta \\ \cos\theta \\ -d \end{bmatrix}$$

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

$$B(q) = \frac{1}{R} \begin{bmatrix} \cos\theta & \cos\theta \\ \sin\theta & \sin\theta \\ L & -L \end{bmatrix}$$

$F(\dot{q}) = 0$  : Considérée nulle dans ce cas.

$G(q) = 0$  : Le mouvement est contraint au sol

$\tau_d = 0$  : Considérée nulle dans ce cas.

Le système ci-dessus peut être transformé en une représentation plus appropriée au contrôle et simulation. Dans cette transformation, nous essayons de trouver un moyen d'éliminer les contraintes de l'équation. Les deux matrices suivantes sont définies pour effectuer cette transformation:

$$\vartheta(t) = \begin{bmatrix} v \\ \omega \end{bmatrix} \quad (1.85)$$

$$S(q) = \begin{bmatrix} \cos\theta & -d\sin\theta \\ \sin\theta & d\cos\theta \\ 0 & 1 \end{bmatrix} \quad (1.86)$$

La matrice  $S(q)$  est la matrice cinématique directe modifiée qui a deux termes de vitesse liés à la distance entre le centre du robot et l'axe de la roue. Par conséquent, nous pouvons écrire l'équation suivante pour le système:

$$\dot{q} = \begin{bmatrix} \dot{x}_c \\ \dot{y}_c \\ \dot{\theta} \end{bmatrix} = S(q)\vartheta(t) = \begin{bmatrix} \cos\theta & -d\sin\theta \\ \sin\theta & d\cos\theta \\ 0 & 1 \end{bmatrix} \begin{bmatrix} v \\ \omega \end{bmatrix} \quad (1.87)$$

On peut facilement démontrer que la matrice  $S(q)$  a la relation suivante avec la matrice  $A(q)$ :

$$S^T(q)A^T(q) = 0 \quad (1.88)$$

On dérive l'équation (1.87) on trouve:

$$\ddot{q} = \dot{S}(q)\vartheta(t) + S(q)\dot{\vartheta}(t) \quad (1.89)$$

Remplaçons les équations (1.87),(1.89) dans l'équation de la dynamique (1.84) on trouve:

$$M(q)\left[\dot{S}(q)\vartheta(t) + S(q)\dot{\vartheta}(t)\right] + V_m(q, \dot{q})S(q)\vartheta(t) + F(\dot{q}) + G(q) + \tau_d = B(q)\tau - A^T(q)\lambda \quad (1.90)$$

$$(q)\dot{S}(q)\vartheta(t) + M(q)S(q)\dot{\vartheta}(t) + V_m(q, \dot{q})S(q)\vartheta(t) + F(\dot{q}) + G(q) + \tau_d = B(q)\tau - A^T(q)\lambda \quad (1.91)$$

Ou:

$$V_m(q, \dot{q}) = \begin{bmatrix} 0 & 0 & md\dot{\theta}\cos\theta \\ 0 & 0 & md\dot{\theta}\sin\theta \\ 0 & 0 & 0 \end{bmatrix}$$

L'étape suivante pour éliminer la matrice de contraintes  $A^T(q)\lambda$  on multiplie l'équation (1.91) par  $S^T(q)$  comme suit:

$$[S^T(q)M(q)S(q)]\dot{\vartheta}(t) + [S^T(q)M(q)\dot{S}(q) + S^T(q)V_m^T(q, \dot{q})S]\vartheta(t) + S^T(q)F(\dot{q}) + S^T(q)G(q) + S^T(q)\tau_d = S^T(q)B(q)\tau - S^T(q)A^T(q)\lambda \quad (1.92)$$

Comme on peut le voir à partir de l'équation ci-dessus, nous avons  $S^T(q)A^T(q)$  qui est nul selon l'équation (1.88). Par conséquent, le terme de contrainte est éliminé et la nouvelle équation dynamique est:

$$\begin{aligned} & [S^T(q)M(q)S(q)]\dot{\vartheta}(t) + [S^T(q)M(q)\dot{S}(q) + S^T(q)V_m^T(q, \dot{q})S]\vartheta(t) + S^T(q)F(\dot{q}) \\ & + S^T(q)G(q) + S^T(q)\tau_d = S^T(q)B(q)\tau \end{aligned} \quad (1.93)$$

Alors on peut réécrire l'équation dynamique du robot mobile comme suit:

$$Mh(q)\dot{\vartheta}(t) + Vh(q, \dot{q})\vartheta(t) + Fh(\dot{q}) + Gh(q) + \tau_n = Bh(q)\tau \quad (1.94)$$

Avec:

$$Mh(q) = S^T(q)M(q)S(q)$$



$$Vh(q, \dot{q}) = S^T(q)M(q)\dot{S}(q) + S^T(q)V_m^T(q, \dot{q})S$$

$$Fh(\dot{q}) = S^T(q)F(\dot{q}) = 0$$

$$Gh(q) = S^T(q)G(q) = 0$$

$$\tau_h = S^T(q)\tau_d$$

$$Bh(q) = S^T(q)B(q)$$

$$S^T(q)A^T(q)\lambda = 0$$

### I.3.4 Application de l'approche de Newton –Euler

La première et l'une des étapes les plus importantes de la modélisation dynamique par l'approche de Newton consiste à dessiner le diagramme de corps libre du système et à en analyser les forces.

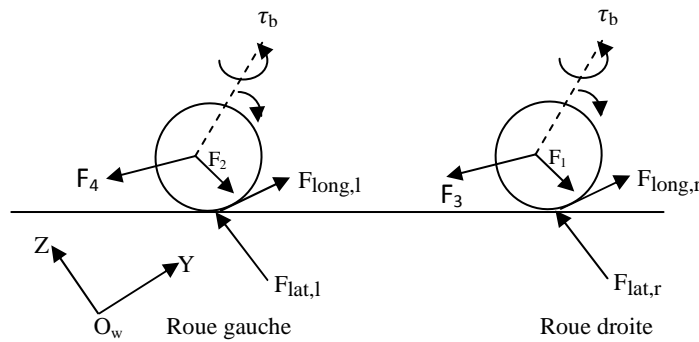


Figure 1.7. Diagramme de couple-forces d'un robot mobile à deux roues.

Les notations suivantes sont introduites dans cette figure et seront utilisées pour la modélisation dynamique par l'approche de Newton:

$v_g$  : vitesse longitudinale.

$v_t$  : vitesse latérale.

$a_g$  : accélération longitudinale.

$a_t$  : accélération latérale.

$F_{gr}$  : Force longitudinale exercée sur la roue droite.

$F_{gl}$  : Force longitudinale exercée sur la roue gauche.

$F_{tr}$  : Force latérale exercée sur la roue droite.

$F_{tl}$  : Force latérale exercée sur la roue gauche.

Comme on peut le voir à partir du diagramme de corps libre (robot mobile) ci-dessus, les seules forces agissant sur le robot sont les forces de l'actionneur agissant sur les roues du robot. Nous commençons la dérivation en représentant la position du robot en utilisant des coordonnées polaires. En supposant que le robot est un corps rigide, sa position peut être représentée en utilisant son angle et son rayon:

$$\hat{r} = re^{i\theta} \quad (1.95)$$

Le dérivé du vecteur de position nous donnera la vitesse et l'accélération du robot:

$$\dot{\hat{r}} = \dot{r}e^{i\theta} + r\dot{\theta}ie^{i\theta} \quad (1.96)$$

$$\ddot{\hat{r}} = \ddot{r}e^{i\theta} + \dot{r}\dot{\theta}ie^{i\theta} - r\dot{\theta}^2e^{i\theta} + \dot{r}\ddot{\theta}ie^{i\theta} + r\ddot{\theta}ie^{i\theta} \quad (1.97)$$

En simplifiant et en écrivant les termes de vitesse et d'accélération en termes radial et tangentiels, nous avons:

$$\dot{\hat{r}} = [\dot{r}]e^{i\theta} + [r\dot{\theta}]e^{i\theta + \frac{\pi}{2}} \quad (1.98)$$

$$\ddot{\hat{r}} = [\ddot{r} - r\dot{\theta}^2]e^{i\theta} + [2\dot{r}\dot{\theta} + r\ddot{\theta}]e^{i\theta + \frac{\pi}{2}} \quad (1.99)$$

Les vitesses et les accélérations sont définies comme suit :

$$v_g = \dot{r} \quad (1.100)$$

$$v_t = r\dot{\theta} \quad (1.101)$$

$$a_g = \ddot{r} - r\dot{\theta}^2 \quad (1.102)$$

$$a_t = 2\dot{r}\dot{\theta} + r\ddot{\theta} \quad (1.103)$$

A partir des quatre équations précédentes, nous pouvons écrire les relations suivantes :

$$a_g = \dot{v}_g - v_t\dot{\theta} \quad (1.104)$$

$$a_t = \dot{v}_t + v_g\dot{\theta} \quad (1.105)$$

L'étape suivante consiste à écrire la deuxième loi de Newton et de trouver la relation entre les forces et les accélérations:

$$\sum F_g = ma_g \quad (1.106)$$

$$ma_g = F_{g_l} + F_{g_r} \quad (1.107)$$

$$\sum F_t = ma_t \quad (1.108)$$

$$ma_t = F_{t_l} + F_{t_r} \quad (1.109)$$

En substituant les termes d'accélération des équations (1.104) et (1.105) dans les équations (1.107) et (1.109), nous avons:

$$\dot{v}_g = v_t \dot{\theta} + \frac{F_{g_l} + F_{g_r}}{m} \quad (1.110)$$

$$\dot{v}_t = -v_g \dot{\theta} + \frac{F_{t_l} + F_{t_r}}{m} \quad (1.111)$$

Les deux équations ci-dessus montrent l'accélération du robot en fonction des forces et des vitesses. La deuxième loi de Newton en rotation autour du centre de masse est:

$$\sum M_c = I_c \ddot{\theta} \quad (1.112)$$

$$\ddot{\theta} = -\frac{(F_{t_l} + F_{t_r})d}{I_c} + \frac{(F_{g_r} - F_{g_l})L}{I_c} \quad (1.113)$$

Les équations (1.110), (1.111) et (1.113) sont les principales équations dynamiques du robot mobile différentiel dérivées de l'approche de Newton.

L'absence de glissement dans les directions longitudinale (roulement pur) et latérale (pas de glissement) crée une indépendance entre les vitesses longitudinales, latérales et angulaires et simplifie les équations dynamiques.

En l'absence de glissement latéral (pas de contrainte de glissement), la vitesse latérale du centre de masse sera:

$$v_l = d \dot{\theta} \quad (1.114)$$

Et l'accélération latérale du centre de masse sera:

$$\dot{v}_t = d\ddot{\theta} \quad (1.115)$$

En combinant les équations (1.111) et (1.104), nous avons:

$$d\ddot{\theta} = -v_g \dot{\theta} + \frac{F_{t_l} + F_{t_r}}{m} \quad (1.116)$$

$$m(d\ddot{\theta} + v_g \dot{\theta}) = F_{t_l} + F_{t_r} \quad (1.117)$$

En combinant les équations (1.116) et (1.105) nous avons:

$$\ddot{\theta} = \frac{(F_{g_r} - F_{g_l})L}{md^2 + I_C} - \frac{mdv_g \dot{\theta}}{md^2 + I_C} \quad (1.118)$$

En combinant les équations (1.113) et (1.103), nous avons:

$$\dot{v}_g = d\dot{\theta}^2 + \frac{F_{g_r} + F_{g_l}}{m} \quad (1.119)$$

Les deux équations (1.118) et (1.119) sont les équations dynamiques du robot considérant les contraintes non-holonomes. Ces deux équations peuvent facilement être transformées en une matrice utilisant les mêmes notations et matrices dans l'approche de Lagrange:

$$(md^2 + I_C)\ddot{\theta} + mdv\dot{\theta} = \frac{(\tau_r - \tau_l)L}{R} \quad (1.120)$$

$$m\dot{v} - md\dot{\theta}^2 = \frac{(\tau_r + \tau_l)}{R} \quad (1.121)$$

La forme matricielle de l'équation ci-dessus est représentée par l'équation suivante:

$$\begin{bmatrix} m & 0 \\ 0 & md^2 + I_C \end{bmatrix} \begin{bmatrix} \dot{v} \\ \ddot{\theta} \end{bmatrix} + \begin{bmatrix} 0 & -md\dot{\theta} \\ md\dot{\theta} & 0 \end{bmatrix} \begin{bmatrix} v \\ \dot{\theta} \end{bmatrix} = \begin{bmatrix} \frac{1}{R} & \frac{1}{R} \\ \frac{L}{R} & \frac{-L}{R} \end{bmatrix} \begin{bmatrix} \tau_r \\ \tau_l \end{bmatrix} \quad (1.122)$$

Finalement, on remarque que les deux méthodes atteindront la même équation dynamique pour le robot mobile.

#### I.4. Conclusion

Dans ce chapitre, la modélisation des robots mobiles est présentée. La modélisation des robots mobiles à roues (Unicycle, Tricycle, différentiel) ont été développés. On s'est

intéressé notamment par le robot mobile unicycle pour la commande cinématique. La modélisation dynamique pour le robot mobile différentiel est présentée par l'approche de Lagrange et Newton-Euler. Pour la commande dynamique des robots mobiles on s'est intéressé par le robot mobile différentiel, parce qu'il est considéré comme référence.

---

**CHAPITRE 2**

**ALGORITHMES D'EVITEMENTS D'OBSTACLES**

---

# ALGORITHMES D'ÉVITEMENTS D'OBSTACLES

### II.1 Introduction

L'évitement d'obstacles est un comportement de base présent dans quasiment tous les robots mobiles. Il est indispensable pour permettre au robot de fonctionner dans un environnement dynamique et pour gérer les écarts entre le modèle interne et le monde réel.

L'évitement d'obstacles est un élément clé de tout système robotique mobile. Si on veut que notre système se déplace dans un environnement réel, on doit considérer l'existence d'obstacles non représentés dans une carte à priori. Les obstacles peuvent être soit statiques comme par exemple une table, une chaise, etc..., ou dynamiques comme un être humain ou un autre robot. Ce que nous entendons par l'évitement d'obstacles c'est la capacité d'aller d'un endroit à un autre sans collision avec des objets présents sur le chemin. Cela signifie que l'évitement d'obstacles est étroitement lié à la planification de trajectoires, ils dépendent tellement les uns des autres qu'ils sont souvent indiscernables.

En général, ils existent deux types de méthodes pour la prévention des obstacles, la planification locale et globale de trajectoires. Un algorithme global d'évitement d'obstacles nécessite une connaissance relativement complète du monde et peut calculer le meilleur chemin du début à la fin du parcours en hors ligne. Comme cela est pratiquement impossible dans un monde dynamique où les obstacles peuvent ne pas être représentés sur la carte.

On trouve dans la littérature de nombreuses techniques et algorithmes pour l'évitement des obstacles. Parmi les approches utilisées, on trouve : la détection de bord classique [40], les champs de potentiel [10], les champs des forces virtuelles [7], les histogramme des champs de vecteur (VFH) [41], la fenêtre dynamique [42,43], la déformation du chemin réactif (RPD) [44], le diagramme polaire (PD) [45], et les techniques qui sont basées sur l'intelligence artificiel (logique floue et réseaux de neurones) [46-48]. Tous ces algorithmes sont proposés pour guider le robot dans l'accomplissement de sa tâche.

Dans notre travail on s'intéresse à deux algorithmes :

- Algorithme des champs de potentiel artificiel.
- Contrôleur flou.

## II.2 Algorithme des champs de potentiel artificiel

### II.2.1. Définition

Dans la méthode d'évitement d'obstacles par champs de potentiel, on assimile le robot à une particule se déplaçant suivant les lignes de courant d'un potentiel créé en fonction de l'environnement perçu par le robot [10]. Ce potentiel traduit différents objectifs tels que l'évitement d'obstacles ou une direction de déplacement préférée. Il est calculé par sommation de différentes primitives de potentiels traduisant chacun de ces objectifs (figure 2.1). Ces différents potentiels peuvent avoir une étendue spatiale limitée ou non, par exemple n'avoir une influence que près des obstacles et leur intensité peut dépendre ou non de la distance.

Le gradient de ce potentiel donne, en chaque point de l'espace, la direction de déplacement du robot. Comme c'est ce gradient qui nous intéresse et non la valeur absolue du potentiel, il est possible de calculer directement en chaque point sa valeur par une simple somme vectorielle en ajoutant les valeurs issues des différents potentiels primitifs.

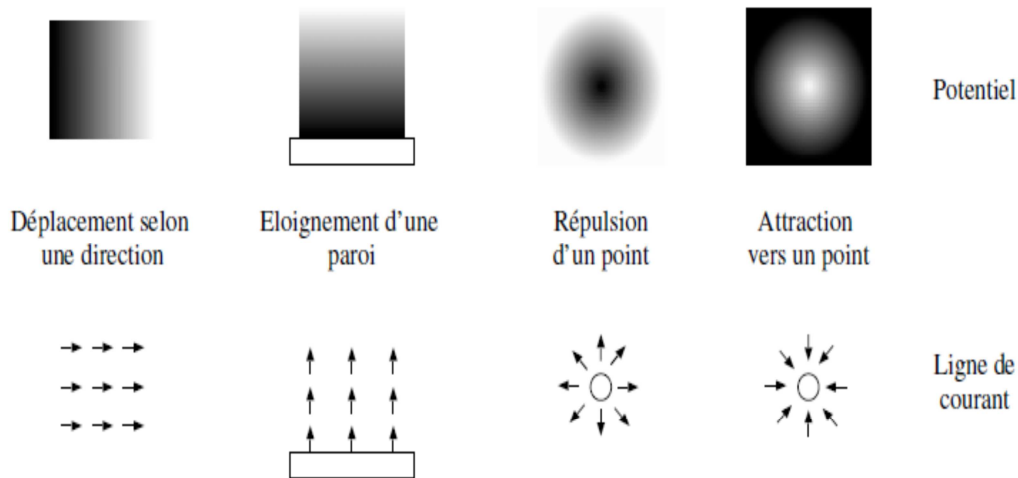


Figure 2.1: Illustration de potentiels primitifs dont la combinaison guide les déplacements du robot. Le robot se déplacera selon les lignes de courant

De plus dans la pratique, pour l'évitement d'obstacles, le potentiel est en général calculé dans l'espace relatif au robot et ne sert qu'à décider de la vitesse et de la direction courante. Il n'est donc nécessaire de l'estimer que pour la position courante du robot, en sommant simplement la contribution des différents éléments perçus.

Le principal inconvénient de cette méthode d'évitement d'obstacles est l'existence, pour certaines configurations d'obstacles (relativement courantes), de minimums locaux du potentiel qui ne permettent pas de décider de la direction à prendre (figure 2.2). Ce problème



peut être traité de différentes façons. Il est par exemple possible de déclencher un comportement particulier lorsque l'on rencontre un tel minimum (déplacement aléatoire, suivi de murs ...). Il est aussi possible d'imposer que le potentiel calculé soit une fonction harmonique, ce qui garantit qu'il n'y a pas de minima mais ceci va compliquer beaucoup son calcul.

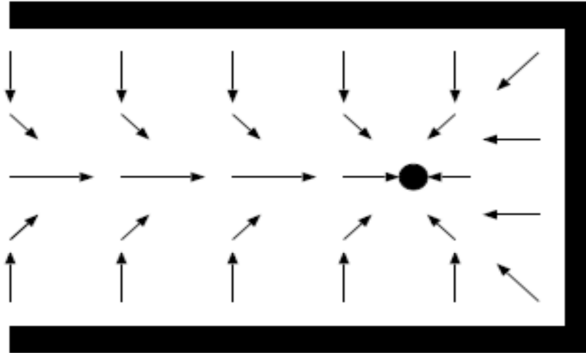


Figure 2.2: Exemple de minimum local dans un champ de potentiel

### II.2.2. Concepts de base des champs de potentiels artificiels

L'algorithme des champs de potentiels artificiels est principalement constitué de vecteurs de force, générés par les obstacles ou les cibles, qui peuvent être linéaires ou tangentiels et qui peuvent avoir des caractéristiques de répulsion, attractives ou aléatoires en fonction de l'état de l'agent par rapport à son environnement. Un exemple d'une fonction de champs de force de potentiel est le suivant :

$$U = k \cdot \exp\left(-\frac{x-x_0}{\alpha} + \frac{y-y_0}{\beta}\right) \quad (2.1)$$

La direction de la force du champ de potentiel peut être trouvée en prenant son gradient [13] :

$$-\frac{dU}{dx} = k \left( \frac{x-x_0}{\alpha} \right) \exp\left(-\frac{x-x_0}{\alpha} + \frac{y-y_0}{\beta}\right) \quad (2.2)$$

$$-\frac{dU}{dy} = -k \left( \frac{y-y_0}{\beta} \right) \exp\left(-\frac{x-x_0}{\alpha} + \frac{y-y_0}{\beta}\right) \quad (2.3)$$

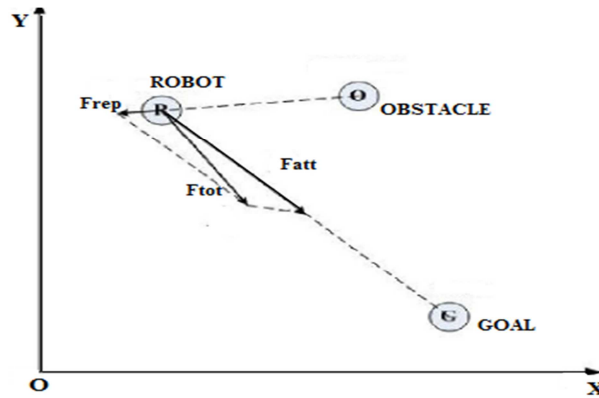


Figure 2.3 : Présentation des forces des champs de potentiels artificiels

### II.2.3 Développement de l'algorithme des champs de potentiels artificiels

L'utilisation des champs de potentiels a été développée principalement et indépendamment dans [49] pour la modélisation de l'environnement immédiat du robot.

Le mouvement du robot dans un champ de potentiel peut être interprété comme le mouvement d'une particule dans un champ de vecteur gradient généré par des particules électriques positives et négatives. Dans cette analogie, le robot représente une charge positive, le but une charge négative et les obstacles sont des ensembles de charges positives. Les gradients dans ce contexte peuvent être interprétés comme des forces qui attirent le robot qui représente des particules chargées positivement à une particule négative qui agit comme le but. La combinaison de la force attractive vers l'objectif et les forces de répulsion des obstacles entraînent le robot dans une voie sûre vers le but.

Le robot suit le chemin le long de la pente négative de la fonction potentielle qui désigne le mouvement de descente vers le point le plus bas dans la vallée. Avec cette analogie, il est clair que le robot peut être piégé dans des minima locaux, loin de l'objectif.

Le potentiel de champ artificiel où le robot se déplace est une fonction scalaire  $U(q) : \mathbb{R}^2 \rightarrow \mathbb{R}$  générée par la superposition de potentiels attractifs et répulsifs

$$U(q) = U_{att}(q) + U_{rep}(q) \quad (2.4)$$

Le potentiel de répulsion est le résultat de la superposition des potentiels répulsifs générés par les obstacles, et ainsi peut être écrit comme :

$$U(q) = U_{att}(q) + \sum_{i=1}^n U_{repi}(q) \quad (2.5)$$

$U_{repi}$  représente le potentiel répulsif généré par l'obstacle  $i$ .

Considérons maintenant que  $U(q)$  est dérivable. A chaque  $q$  la pente du champ de potentiel notée  $\nabla U(q)$ , vecteur qui pointe dans la direction de déplacement du robot, augmente au maximum  $U(q)$ .

En outre, la force qui pousse le robot est le gradient négatif du potentiel artificiel [50] :

$$F(q) = F_{att}(q) + F_{rep}(q) = -\nabla U_{att}(q) - \nabla U_{rep}(q) = \left( \frac{\partial U(x, y)}{\partial x} \quad \frac{\partial U(x, y)}{\partial y} \right)^T \quad (2.6)$$

La force  $F(q)$  est un vecteur qui pointe dans la direction de déplacement du robot et à chaque  $q$  maximale le potentiel  $U$  décroît. Cette force peut être considérée comme le vecteur de la vitesse qui entraîne le robot.

### A. Potentiel Attractif

Le champ attractif peut être simplement défini sous la forme d'une parabolique ou d'une forme canonique (2.8):

$$U_{att}(q) = \frac{1}{2} K_{goal} P_{goal}^2(q) \quad (2.7)$$

$$U_{att}(q) = K_{goal} P_{goal}(q) \quad (2.8)$$

$K_{goal}$  est la constante d'attraction qui est positive et  $P_{goal}$  : la distance entre le robot et le but

On aura alors une force de type :

$$F_{att}(q) = -K_{goal} \nabla p_{goal}(q) = -\frac{\left( q - q_{goal} \right)}{\left\| q - q_{goal} \right\|} \quad (2.9)$$

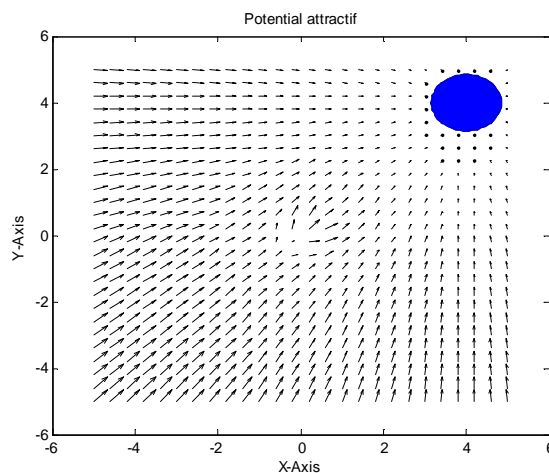


Figure 2.4 : Présentation du potentiel attractif

### B. Potentiel Répulsif

Le potentiel répulsif doit servir à créer une barrière de potentiel autour des obstacles (autour de C-obstacles) qui ne peut être traversé par le robot. De plus, on ne veut pas que ce potentiel affecte le mouvement du robot lorsque celui-ci est suffisamment loin des obstacles.

Une des formes correspondantes à ces objectifs est :

$$U_{rep}(q) = \begin{cases} \frac{1}{2} K_{obs} \left( \frac{1}{p(q)} - \frac{1}{p_0} \right) & \text{si } p(q) < p_0 \\ 0 & \text{si } p(q) \geq p_0 \end{cases} \quad (2.10)$$

$p_0$  est appelée distance d'influence des obstacles et  $k_{obs}$  est la constante de répulsion.

La fonction  $U_{rep}$  est positive ou nulle et tend vers l'infini lorsqu'on se rapproche de la frontière de C-obstacles. Ce potentiel de répulsion est plus fort lorsque le robot est proche de l'obstacle et a une basse influence lorsque le robot est loin. On a alors :

$$F_{rep}(q) = -\nabla U_{rep}(q) \quad (2.11)$$

$$F_{rep}(q) = \begin{cases} K_{obs} \left( \frac{1}{p(q)} - \frac{1}{p_0} \right) \frac{1}{p^2(q)} \nabla p(q) & \text{si } p(q) < p_0 \\ 0 & \text{si } p(q) \geq p_0 \end{cases} \quad (2.12)$$

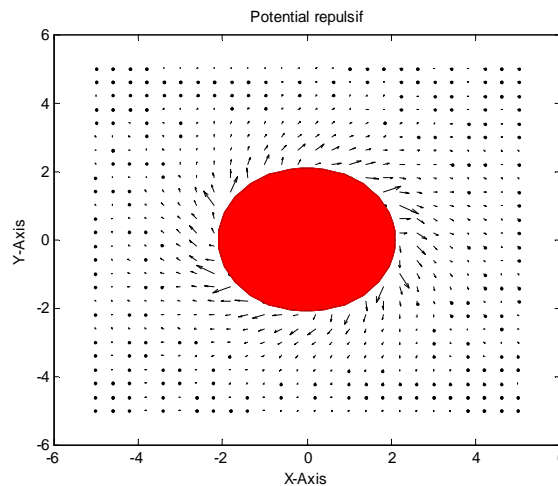


Figure 2.5 : Potentiel répulsif

### C. La descente de Gradient

Pour la planification de trajectoires, la descente du gradient consiste simplement à suivre la direction indiquée par la force  $F$  et à avancer dans cette direction d'un pas de longueur  $\delta_i$ . Par exemple, avec  $q=(x, y)$  on aura :

$$\begin{cases} x(q_{i+1}) = x(q_i) + \delta_i \frac{\partial U(x, y)}{\partial x} \\ y(q_{i+1}) = y(q_i) + \delta_i \frac{\partial U(x, y)}{\partial y} \end{cases} \quad (2.13)$$

L'algorithme s'arrêtera lorsqu'on sera à une distance de  $q_{goal}$  inférieure à  $\delta_i$ , dans le cas où un chemin est déterminé. L'inconvénient de cette méthode est que l'on peut tomber sur un minimum local.

#### D. Simulations

On considère des cas simples où l'espace de configuration est limité à deux dimensions. Faisant varier le nombre des obstacles pour chaque configuration.

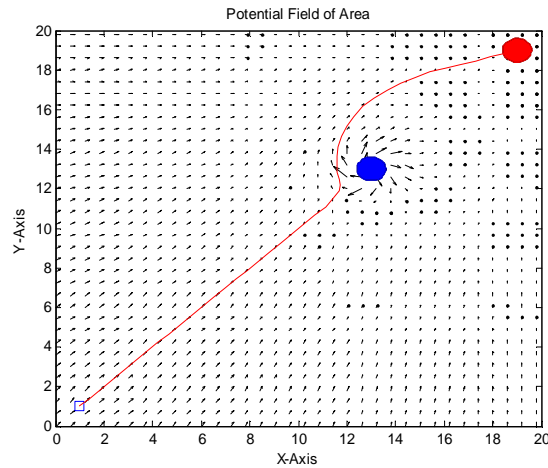


Figure 2.6 : Planification de trajectoire d'un robot mobile par la technique CPA en présence d'un seul obstacle.

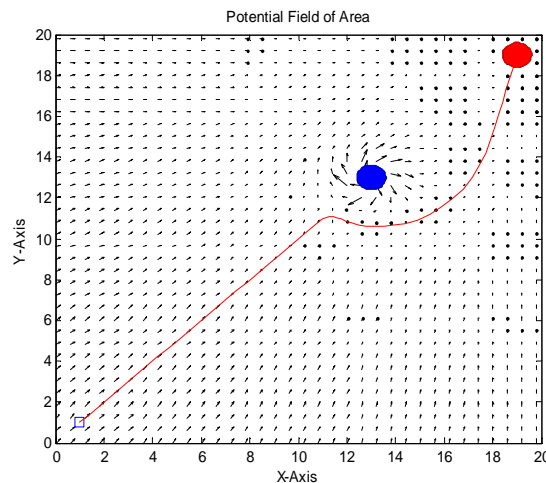


Figure 2.7 : Planification de trajectoire d'un robot mobile par la technique CPA avec variation du coefficient de répulsion.

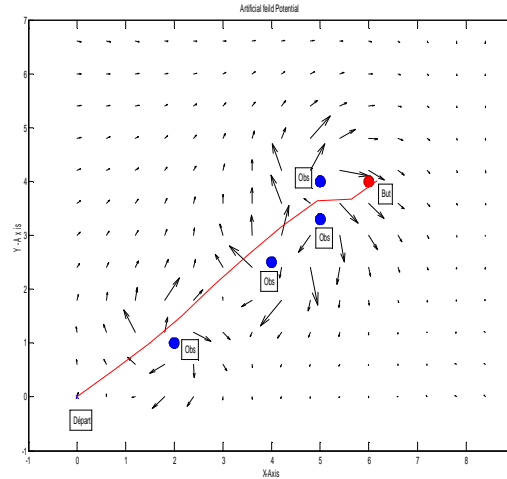


Figure 2.8 : Planification de trajectoire d'un robot mobile par la technique CPA en présence de 4 obstacles.

Les figures (2.6) et (2.7) présentent les trajectoires d'un robot mobile dans le cas de présence d'un seul obstacle, la figure (2.6) présente un évitement à gauche et la figure (2.7) présente un évitement à droite.

La figure (2.8) présente la trajectoire d'un robot mobile dans le cas de présence de plusieurs obstacles, on remarque que le robot mobile se dirige vers le but tout en évitant les obstacles.

## II.2.4 Champs de potentiels avancés

Les champs de potentiels avancés peuvent être représentés par les équations suivantes :

$$U(q) = U_{att}(q) + U_{rep}(q) + U_{start}(q) \quad (2.14)$$

$$U_{rep}(q) = \begin{cases} K_{obs} \exp(-p(q)) & \text{si } p(q) < p_0 \\ 0 & \text{si } p(q) \succ p_0 \end{cases} \quad (2.15)$$

$p(q)$  représente la distance euclidienne entre le robot et les obstacles.  $U_{start}$  est le potentiel du point de départ et  $K_{obs}$  est le coefficient de répulsion.

$$U_{att}(q) = K_{goal} |q - q_{goal}|^2 \quad (2.16)$$

$$U_{start}(q) = K_{goal} |q - q_{start}|^2 \quad (2.17)$$

$K_{goal}$  est le coefficient d'attraction.

La trajectoire du robot mobile peut être calculée selon le système suivant :

$$\begin{bmatrix} x(k+1) \\ y(k+1) \end{bmatrix} = \begin{bmatrix} x(k) \\ y(k) \end{bmatrix} + \rho_0 \begin{bmatrix} \cos \varphi \\ \sin \varphi \end{bmatrix} \quad (2.18)$$

avec  $\varphi$  représentant l'angle de braquage,  $\rho_0$  est appelée distance d'influence des obstacles.

Avec :

$$\varphi = \arctg\left(\frac{\partial U(x, y)}{\partial x}, \frac{\partial U(x, y)}{\partial y}\right) \quad (2.19)$$

### Résultats de simulation

Dans nos travaux de simulation, on considère des cas simples où l'espace de configuration est limité à deux dimensions. Dans cette phase, nous présentons les résultats de simulation en utilisant MATLAB qui est un environnement de travail complet, puissant et performant.

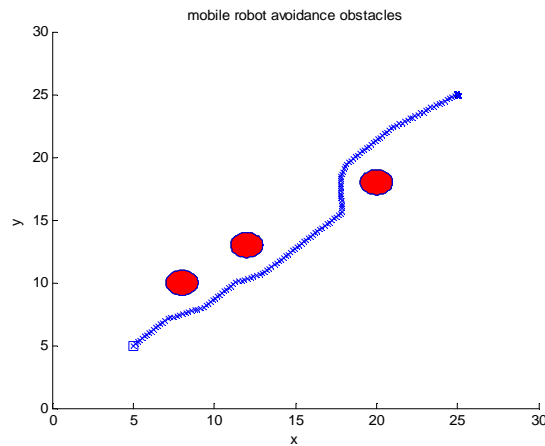


Figure 2.9 : Evitement en présence de trois obstacles

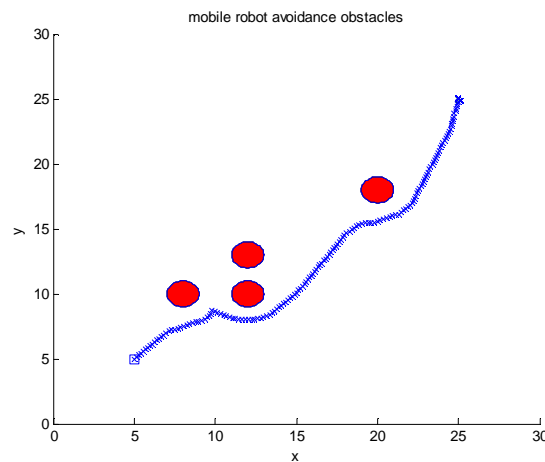


Figure 2.10 : Evitement en présence de quatre obstacles

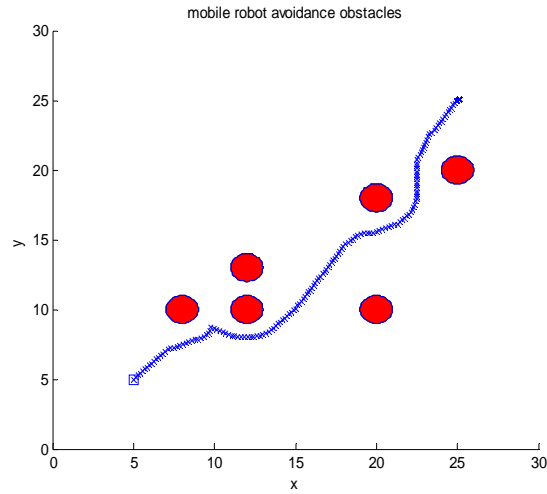


Figure 2.11 : Evitement en présence de six obstacles

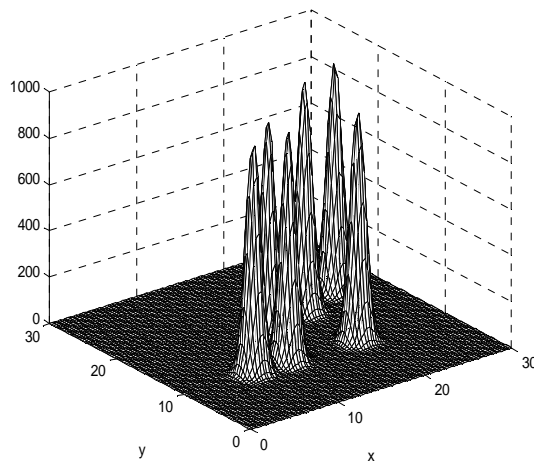


Figure 2.12: Présentation des fonctions gaussiennes des obstacles (cas de 6 obstacles)

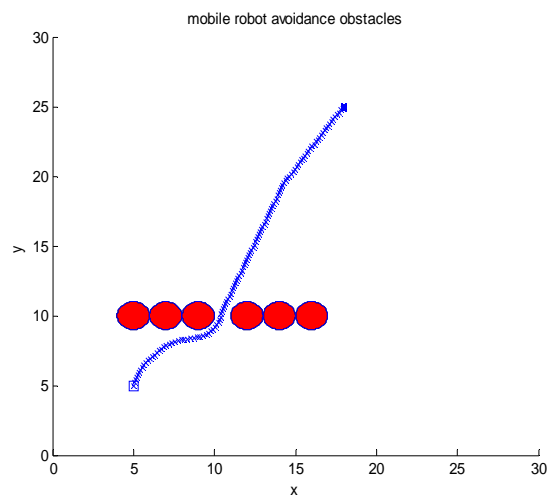


Figure 2.13 : Evitement d'obstacles entre deux murs



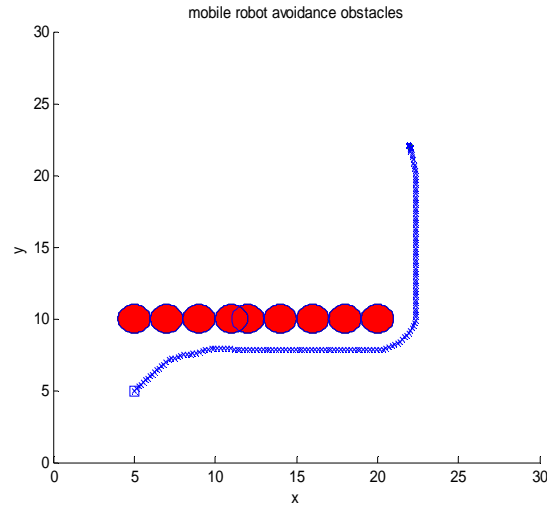


Figure 2.14 : Suivi d'un mur

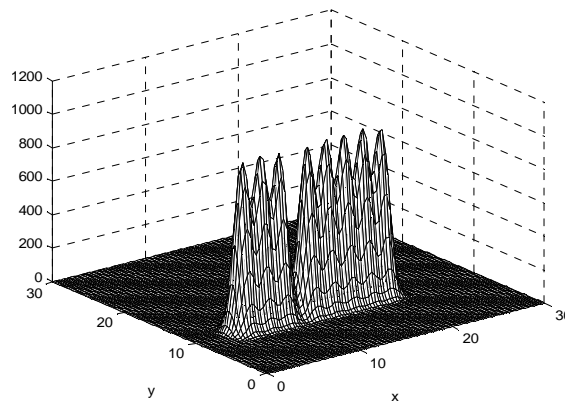


Figure 2.15 : Présentation des fonctions gaussiennes du mur

D'après les essais réalisés sous différents environnements, nous constatons la diffusion du potentiel et la recherche d'une trajectoire optimale qui relie la configuration de départ à la configuration d'arrivée sont assurées. Les résultats obtenus sont satisfaisants et permettent de valider les méthodes des champs de potentiels artificiels présentées.

### II.3 Elaboration d'un contrôleur flou pour la navigation d'un robot mobile

Plusieurs travaux de recherche proposent la logique floue comme une solution pour le contrôle des robots mobiles [51,52]. La logique floue est un algorithme qui permet l'exploitation des règles linguistiques selon un format IF-THEN pour le contrôle ou la modélisation de divers systèmes sans une description mathématique de ceux-ci. Les règles linguistiques travaillent avec des paramètres subjectifs, non clairs. En conséquence, l'utilisation de la logique floue est très utile lorsque nous voulons appliquer les connaissances

humaines dans le domaine du contrôle. La structure de base d'un contrôleur flou comprend la fuzzification, l'inférence et la défuzzification.

Le problème à résoudre dans ce travail nécessite une architecture réactive pour que le robot réagisse en temps réel. Nous allons subdiviser le système de navigation en trois modules faciles à gérer afin d'assurer au mieux la navigation autonome du robot et mettre l'accent sur l'exécution de tâches spécifiques. Nous proposons également un module de détection d'obstacles, un contrôleur de navigation libre et un contrôleur d'évitement d'obstacles capables de surmonter les incertitudes présentées par le monde réel et d'aider le robot à suivre de meilleurs et rapides chemins. Cette subdivision est une approche efficace, car elle rend le système modulaire, ce qui simplifie la solution de navigation; ainsi qu'elle offre la possibilité d'ajouter de nouveaux comportements au système sans provoquer une augmentation de la complexité. Dans ce travail, notre objectif est de démontrer la faisabilité de la logique floue pour l'évitement d'obstacles d'un robot mobile

Le principal problème de chemin de routage des robots mobiles est de mettre le robot à partir du point A vers B sans collision avec des obstacles, en respectant les contraintes temporelles. Dans cette étude nous décomposons ce problème en deux sous problèmes :

- Trouver un chemin vers la cible : c'est un problème global mais la plus simple solution est de trouver un chemin à partir d'une position de départ à une donnée.
- Eviter les obstacles : cela peut souvent être résolu en utilisant uniquement l'information locale.

### ***Présentation du robot mobile***

Le type de robot mobile utilisé dans ce travail est le tricycle; il a deux roues arrière indépendantes et une roue devant assurer la stabilité pendant son déplacement [56]. La mission de base du robot mobile est d'atteindre la configuration d'arrivée appelée cible par un calcul de la distance et de l'angle entre le robot et la cible. Pour les simulations, on utilise le modèle cinématique suivant :

$$\begin{cases} \dot{x} = v \cos \theta \\ \dot{y} = v \sin \theta \\ \dot{\theta} = \frac{v}{D} \operatorname{tg} \varphi \end{cases} \quad (2.20)$$

Avec  $\varphi$  représentant l'angle de braquage du robot et  $D$  la longueur du châssis.

La stratégie de navigation de base est réactive, combinant trois comportements de base afin d'assurer la navigation autonome du robot : la navigation libre, l'évitement d'obstacles fixes et le rapprochement du but. Cette subdivision a pour objectif de faciliter et d'assurer au mieux la navigation autonome du robot.

### II.3.1 Navigation libre

Si le robot ne reçoit aucune information de ses capteurs au cours de son déplacement alors il considère que le chemin est libre, c'est-à-dire aucune présence d'obstacles; donc dans cette situation il se dirige vers la cible. Nous avons choisi un contrôleur de type Mamdani, ce type de contrôleur est généralement le plus utilisé dans ce type de problème. Ce modèle présente l'avantage d'être facilement interprétable et bien adapté à notre application. Il reçoit comme entrées  $\theta'$  l'angle entre la cible et le robot et renvoie en sortie la commande  $\varphi$  qui correspond à l'angle de braquage. La figure 2.16 représente la structure du contrôleur de navigation libre.

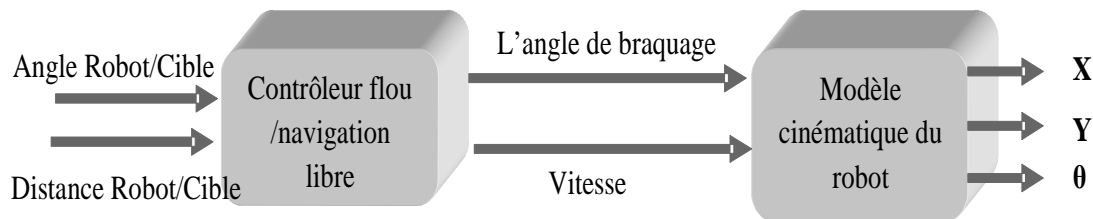


Figure 2.16 : Structure du contrôleur de navigation libre

Ce contrôleur a deux entrées:

- Distance robot / cible noté  $d_{cb}$
- L'angle entre l'orientation du robot et celle de la cible, noté  $\theta'$ .

Les variables de contrôle générées sont l'angle de braquage  $\varphi$  et la vitesse  $v$ .

Dans ce qui suit nous allons décrire brièvement la structure de base de notre contrôleur flou qui comporte trois parties : la fuzzification, l'inférence et la défuzzification.

#### II.3.1.1 Fuzzification

Les valeurs délivrées par les capteurs du robot sont traduites en labels de sous ensemble flous caractérisant les variables linguistiques associées.

**a) Fuzzification de la variable angle robot \_ cible**

La variable d'entrée, l'angle entre le robot et la cible  $\theta'$ , possède cinq variables floues NG (Négatif Grand), NP (Négatif Petit), Z (Zéro), PP (Positif Petit) et PG (Positif Grand) définies comme suit :

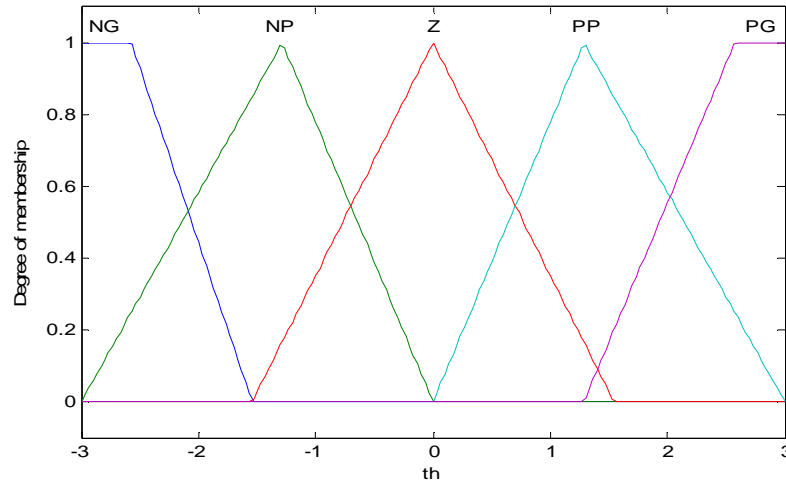


Figure 2.17 : Fonction d'appartenance de l'angle robot \_ cible

**b) Fuzzification de la variable distance robot \_ cible**

La variable d'entrée, distance entre le robot et la cible  $d_{cb}$ , possède trois variables floues Z (Zéro), M (Moyen) et G (Grand) définies comme suit :

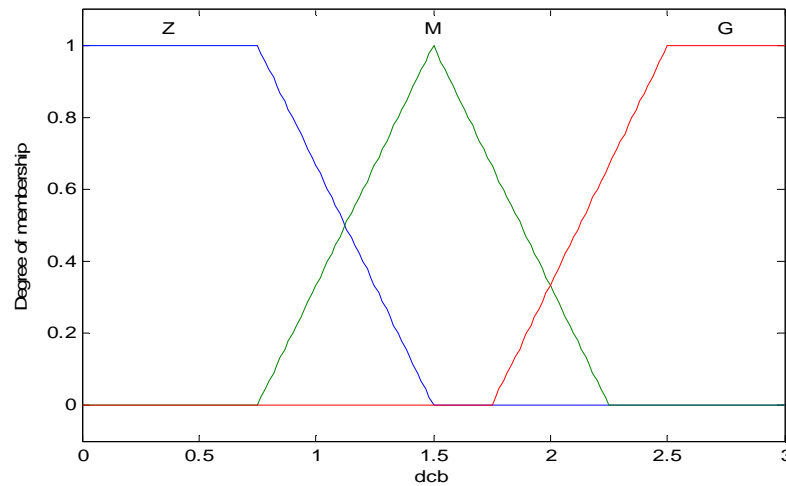


Figure 2.18 : Fonction d'appartenance de la distance robot \_ cible

**c) Fuzzification de la variable angle de braquage**

La variable de sortie qui est l'angle de braquage  $\phi$  possède cinq variables floues NG (Négatif Grand), NP (Négatif Petit), Z (Zéro), PP (Positif Petit) et PG (Positif Grand) définies comme suit :

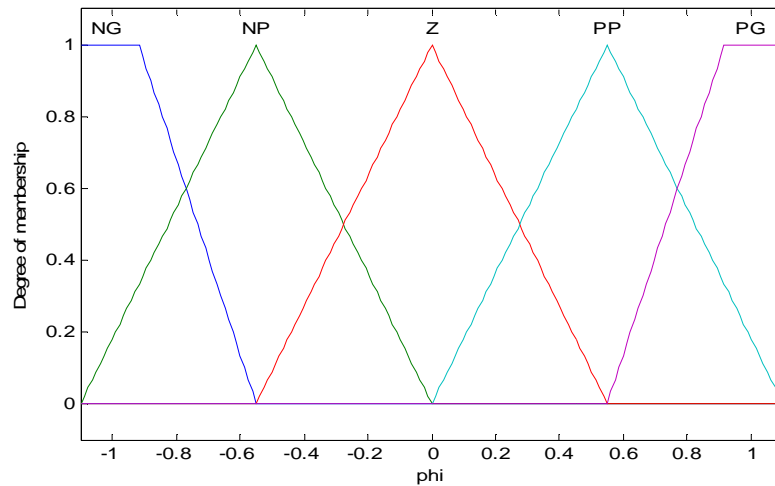


Figure 2.19 : Fonction d'appartenance de l'angle de braquage

**d) Fuzzification de la variable vitesse du robot**

La variable de sortie qui est la vitesse du robot  $v$ , possède trois variables floues Z (Zéro), M (Moyen) et G (Grand) définies comme suit :

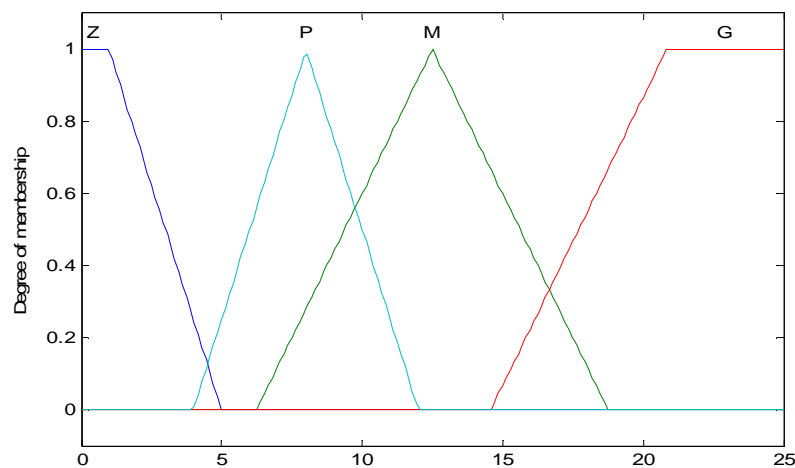


Figure 2.20 : Fonction d'appartenance de la vitesse du robot

### II.3.1.2 Inférence

Un contrôleur flou prend généralement la forme d'une série de règles « Si- alors ». L'inférence est basée sur des opérations min et max afin d'effectuer l'inférence des règles représentées dans le tableau suivant:

			L'angle robot_cible				
			<i>NG</i>	<i>NP</i>	<i>Z</i>	<i>PP</i>	<i>PG</i>
Distance robot_cible	<i>Z</i>	$\varphi$	<i>NG</i>	<i>NP</i>	<i>Z</i>	<i>PP</i>	<i>PG</i>
		$v$	<i>P</i>	<i>P</i>	<i>Z</i>	<i>Z</i>	<i>M</i>
	<i>P</i>	$\varphi$	<i>NG</i>	<i>NP</i>	<i>Z</i>	<i>PP</i>	<i>PG</i>
		$v$	<i>M</i>	<i>M</i>	<i>M</i>	<i>M</i>	<i>M</i>
	<i>M</i>	$\varphi$	<i>NG</i>	<i>NP</i>	<i>Z</i>	<i>PP</i>	<i>PG</i>
		$v$	<i>G</i>	<i>M</i>	<i>M</i>	<i>M</i>	<i>G</i>
	<i>G</i>	$\varphi$	<i>NG</i>	<i>NP</i>	<i>Z</i>	<i>PP</i>	<i>PG</i>
		$v$	<i>G</i>	<i>G</i>	<i>G</i>	<i>G</i>	<i>G</i>

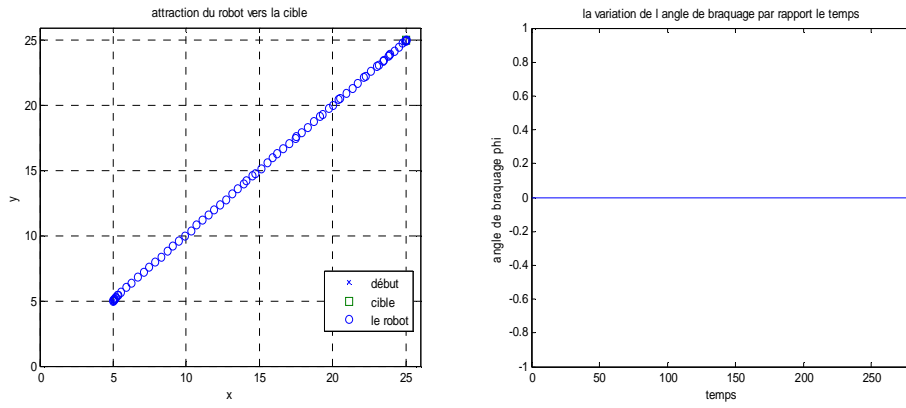
Tableau 2.1 : Règles d'inférence de la navigation libre

### II.3.1.3 Défuzzification

La méthode de défuzzification la plus utilisée est celle de la détermination du centre de gravité de la fonction d'appartenance résultante. Cette méthode est la plus utilisée car elle est caractérisée par un simple calcul très rapide.

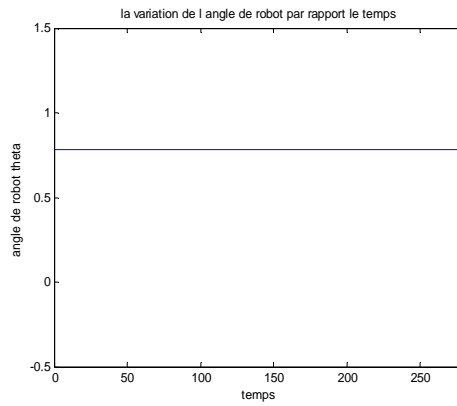
### II.3.1.4 Résultats de simulation

Nous présentons ici quelques exemples réalisées. L'environnement est une salle carrée. Le robot est un rectangle qui se déplace selon le modèle cinématique mentionné auparavant. Nous avons utilisé l'environnement Matlab-Simulink qui est un environnement de travail adéquat. Dans ces travaux de simulation, on va essayer de changer l'orientation du robot par rapport à la cible et on va voir le comportement du robot pour atteindre la cible.



a)

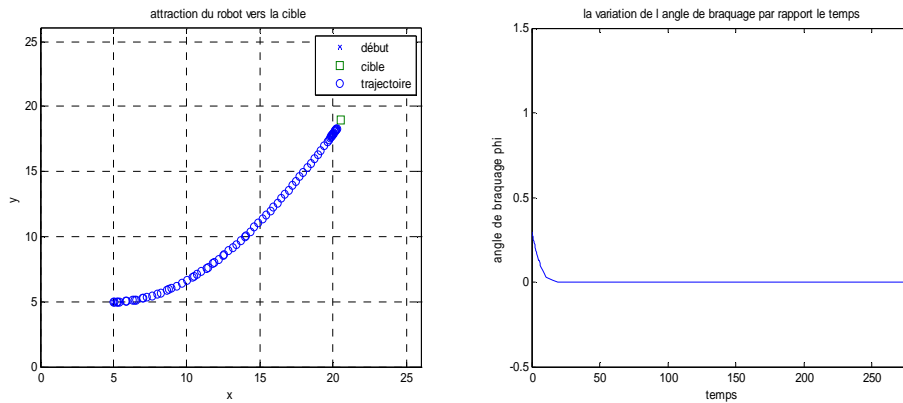
b)



c)

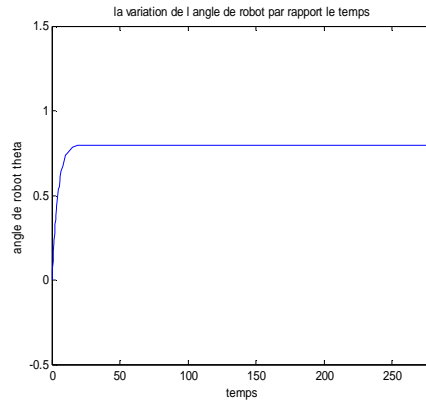
Figure 2.21 : a) Navigation libre où la cible est devant le robot avec  $\theta_0 = \pi / 4$

b) Variation de l'angle de braquage du robot. c) Variation de l'angle d'orientation du robot



a)

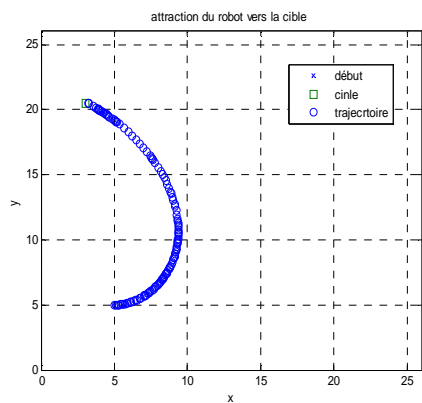
b)



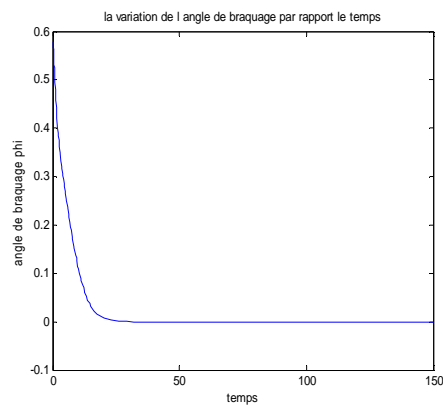
c)

Figure 2.22 : a) Navigation libre où la cible est devant le robot avec  $\theta_0=0$ . b) Variation de l'angle de braquage. c) Variation de l'angle d'orientation du robot.

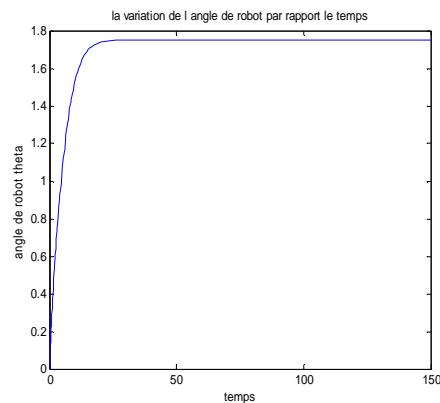
Dans le cas où le robot est initialement orienté vers sa destination, le robot effectue une pure translation à partir de sa position initiale. Les figures 2.21 et 2.22 illustrent ces déplacements quel que soit la position du robot et de la cible.



a)



b)



c)

Figure 2.23 : a) Navigation libre où la cible est derrière le robot avec  $\theta_0=0$ . b) Variation de l'angle de braquage. c) Variation de l'angle d'orientation du robot.



Dans le cas où le robot est orienté dans une direction autre que celle du point d'arrivée, le robot commence à se déplacer en ligne droite avec un braquage permettant de corriger son orientation. Une fois le robot mobile orienté correctement, il reprendra l'étape précédente. La figure 2.23 représente le cas d'un robot mobile dont l'orientation initiale est différente de celle de sa destination.

### II.3.2 Navigation avec évitement d'obstacles

Pour éviter les obstacles, nous avons développé un système de contrôle réactif en utilisant la logique floue. Notre système d'évitement d'obstacles met en œuvre une stratégie de contrôle intelligente réactive en ce sens que l'environnement immédiat réagit (avec l'aide du raisonnement approximatif) aux données sensorielles. Par exemple, si le robot est à proximité immédiate d'un obstacle devant lui, le contrôleur flou permet de déterminer une nouvelle direction à suivre en temps réel afin d'éviter la collision avec l'obstacle. Le robot sera alors immédiatement régi en fonction de la directive émise par le contrôleur flou. Ces comportements réactifs permettent au robot mobile d'éviter les obstacles statiques de l'environnement, sans exiger de modèles prédéfinis. L'existence d'un but nous permet de nous concentrer pleinement sur deux comportements fondamentaux : l'évitement des collisions et le rapprochement du but.

Ce contrôleur travaille en adaptant en étroite collaboration avec les capteurs afin de détecter et d'éviter les obstacles, en adaptant la trajectoire à la situation actuelle. Ce comportement est plus complexe par rapport à la navigation libre. Il reçoit comme entrées la distance et l'angle entre le robot et l'obstacle et renvoie en sortie la commande  $\varphi$  qui correspond au changement de direction du robot (l'angle de braquage) et la vitesse  $v$  du robot. La figure 2.24 montre la structure du contrôleur d'évitement d'obstacles.

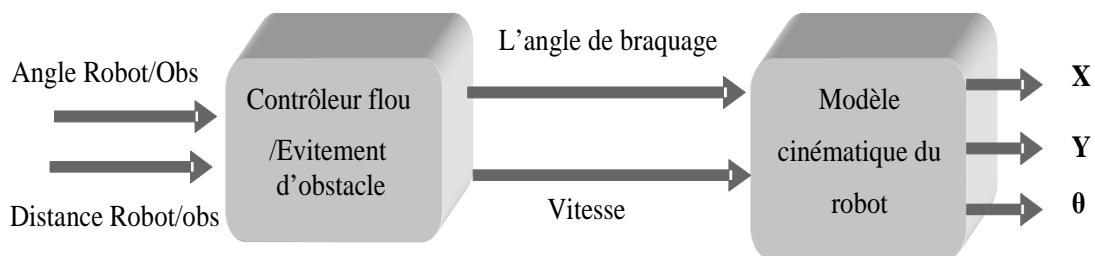


Figure 2.24 : Structure du contrôleur de navigation avec évitement d'obstacles

Ce contrôleur a deux entrées:

- Distance robot / obstacle noté  $d_{obs}$
- L'angle entre l'orientation du robot et celle de l'obstacle, noté  $\theta_{obs}$

Les variables de contrôle générées sont l'angle de braquage  $\varphi$  et la vitesse  $v$ .

### II.3.2.1 Fuzzification

#### a) Fuzzification de la variable angle robot \_ obstacle

La fonction d'appartenance de la variable d'entrée angle robot\_obstacle est exprimée par cinq ensembles flous qui sont : NG (Négatif Grand), NP (Négatif Petit), Z (Zéro), PP (Positif Petit) et PG (Positif Grand). Ces variables sont représentées sur la figure 2.25.

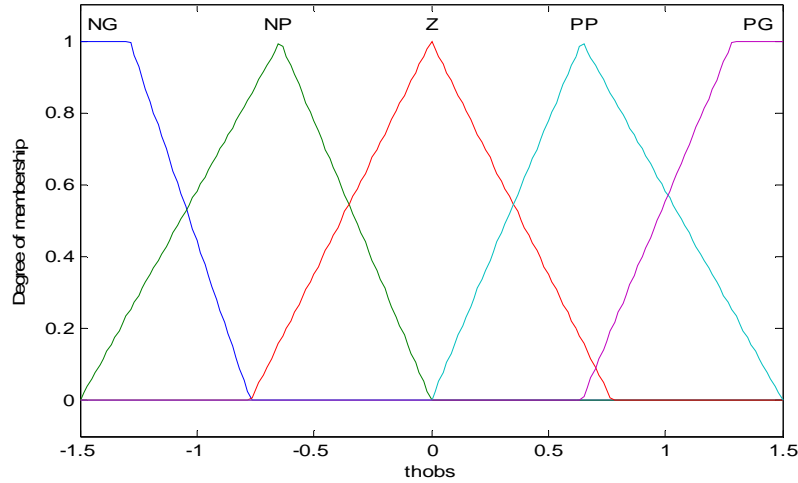


Figure 2.25 : Fonction d'appartenance de l'angle robot\_obstacle

#### b) Fuzzification de la variable distance robot \_ obstacle

La fonction d'appartenance de la variable d'entrée angle robot\_obstacle est exprimée par trois ensembles flous qui sont : Z (Zéro), M (Moyen) et G (Grand). Ces variables sont représentées sur la figure 2.26.

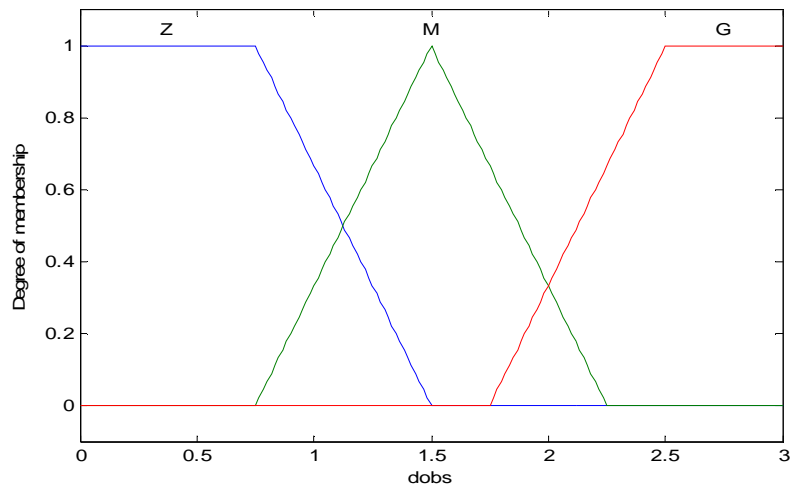


Figure 2.26 : Fonction d'appartenance de la distance robot\_obstacle

Les fonctions d'appartenance des variables de sortie (l'angle de braquage  $\varphi$  et la vitesse  $v$ ) sont exprimées précédemment dans la partie de la navigation libre.

### II.3.2.2 Inférence

Le tableau 2.2 regroupe la série des règles d'inférence floues.

			L'angle robot_obstacle				
			<i>NG</i>	<i>NP</i>	<i>Z</i>	<i>PP</i>	<i>PG</i>
Distance robot_obstacle	<i>Z</i>	$\varphi$	<i>NP</i>	<i>NG</i>	<i>NG</i>	<i>PG</i>	<i>PP</i>
		$v$	<i>G</i>	<i>Z</i>	<i>Z</i>	<i>Z</i>	<i>G</i>
	<i>M</i>	$\varphi$	<i>NP</i>	<i>NP</i>	<i>NG</i>	<i>PG</i>	<i>PP</i>
		$v$	<i>M</i>	<i>M</i>	<i>M</i>	<i>M</i>	<i>M</i>
	<i>G</i>	$\varphi$	<i>Z</i>	<i>NG</i>	<i>NP</i>	<i>PP</i>	<i>Z</i>
		$v$	<i>G</i>	<i>G</i>	<i>G</i>	<i>G</i>	<i>G</i>

Tableau 2.2 : Règles d'inférence de la navigation avec évitement d'obstacles

### II.3.2.3 Défuzzification

La méthode de défuzzification la plus utilisée est celle de la détermination du centre de gravité de la fonction d'appartenance résultante.

### II.3.3 Stratégie de navigation

La stratégie de navigation de base est réactive, combinant trois comportements de base afin d'assurer la navigation autonome du robot : la navigation libre, l'évitement d'obstacles fixes et le rapprochement du but. Cette subdivision a pour objectif de faciliter et d'assurer au mieux la navigation autonome du robot.

La figure 2.27 présente la structure de base du système de navigation composée d'un module d'un contrôleur flou de navigation libre et d'un contrôleur flou d'évitement d'obstacles, et d'un module de navigation qui est considéré comme un switch. Cette structure permet de déterminer en temps réel l'angle de braquage et la vitesse à assigner au robot pour lui permettre de naviguer dans un environnement structuré et parsemé d'obstacles.

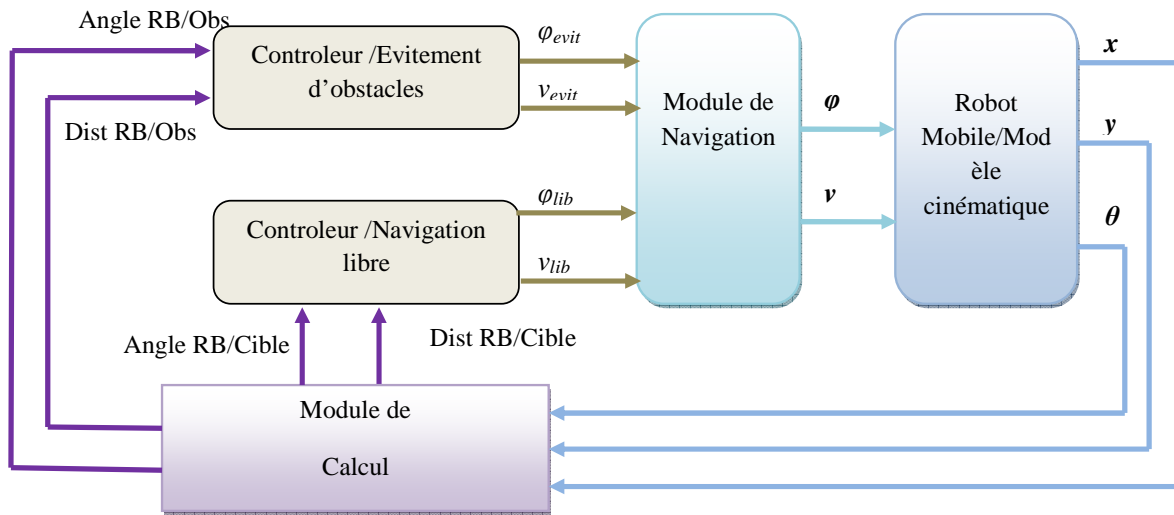


Figure 2.227 : Stratégie de navigation

**Module de navigation :** il est considéré comme un commutateur, il choisit l'angle de braquage et la vitesse du robot parmi les vitesses et les angles de braquage fournis par le contrôleur de navigation libre et évitement d'obstacles selon l'algorithme suivant :

```

Si DistRb_obs < 0,5 Alors
    v = vevit
    φ = φevit
Sinon
    v = vlib
    φ = φlib
Fin SI
    
```

### II.3.4 Résultats de simulation

Si le robot rencontre un obstacle imprévu, celui-ci est détecté par l'intermédiaire d'un module de détection d'obstacles; le robot doit réagir en temps réel c'est-à-dire exécuter une action de contournement autour de l'obstacle détecté.

Les figures ci-dessous montrent quelques cas de notre système de navigation avec évitement d'obstacles.

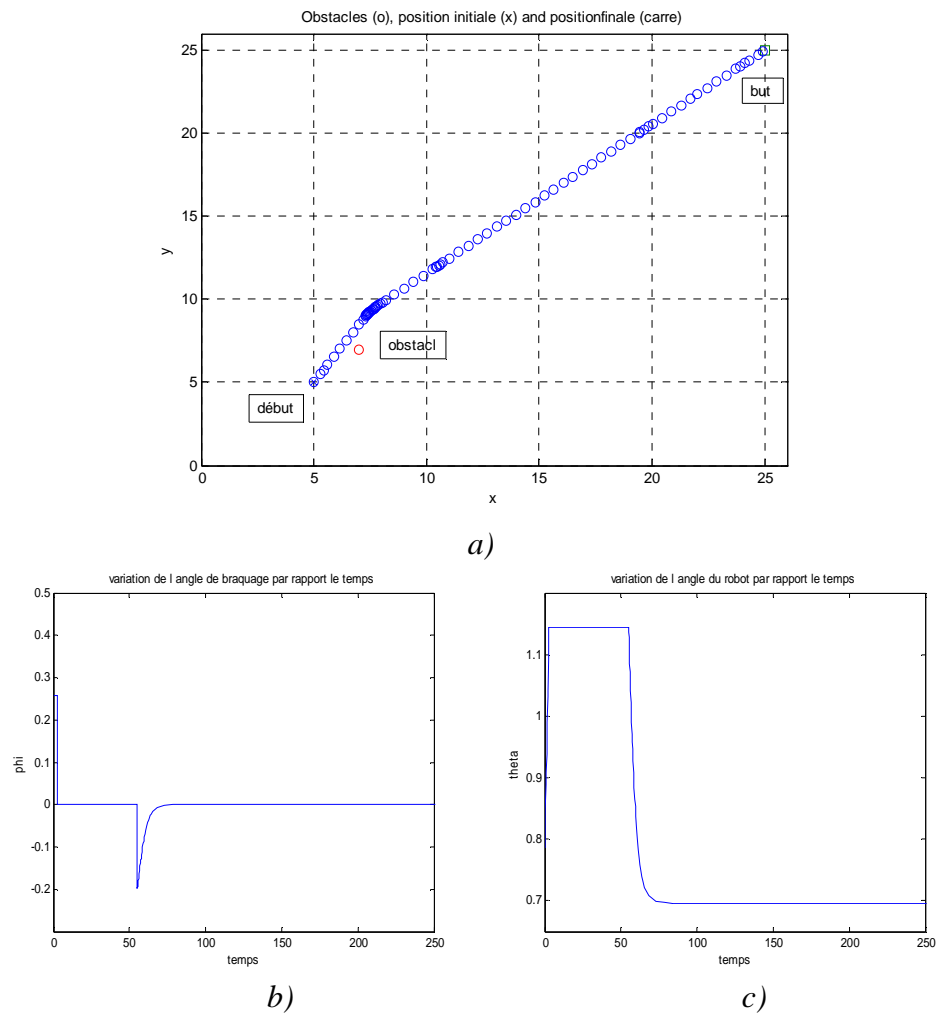
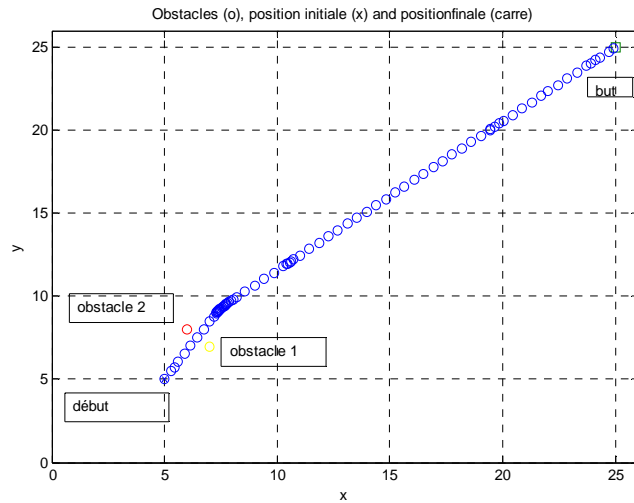
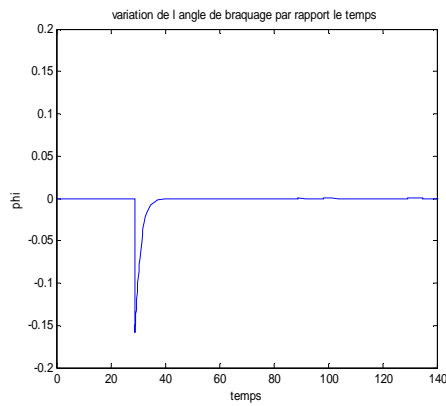


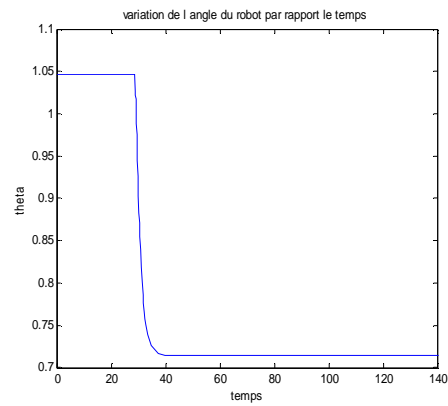
Figure 2.28 : a) Navigation avec évitement d'un seul obstacle. b) Variation de l'angle de braquage. c) Variation de l'angle d'orientation du robot.



a)

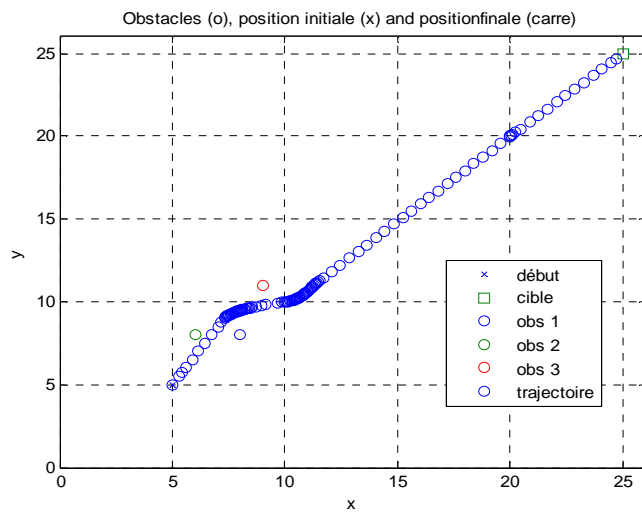


b)



c)

Figure 2.29 : a) Navigation avec évitement de deux obstacles. b) Variation de l'angle de braquage. c) Variation de l'angle d'orientation du robot.



a)

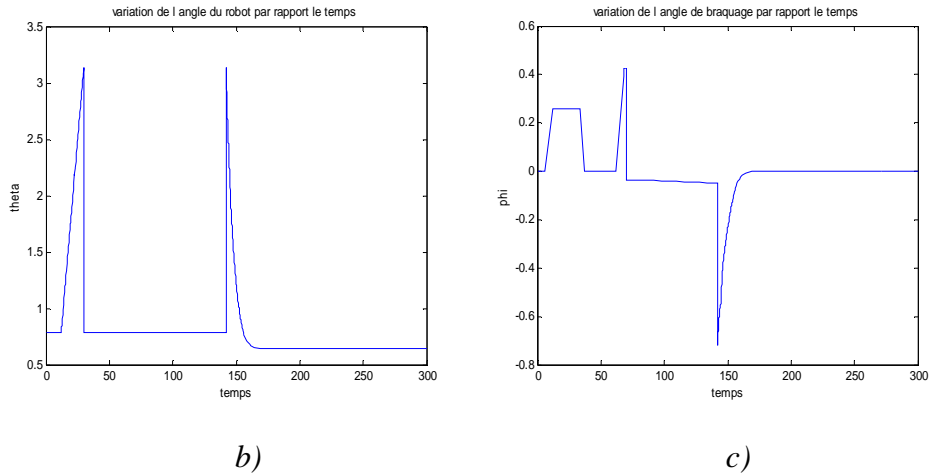


Figure 2.30 : a) Navigation avec évitement de trois obstacles. b) Variation de l'angle de braquage. c) Variation de l'angle d'orientation du robot.

Dans le premier cas de test présenté aux figures 2.28-a et 2.29-a, un simple environnement avec un nombre réduit de petits obstacles a été testé. Le robot utilise l'angle de braquage afin d'éviter l'obstacle ou les deux obstacles, ensuite il se déplace selon des trajectoires plus ou moins rectilignes vers le but. A l'approche de la cible, le robot ralentit et s'immobilise.

Dans le deuxième cas de test présenté sur la figure 2.30-a, de simple environnement avec présence de trois obstacles ont été testés. Le robot utilise toujours l'angle de braquage afin d'éviter les obstacles, ensuite il se déplace selon des courbes vers le but assigné. Il ralentit et s'immobilise à l'approche de la cible.

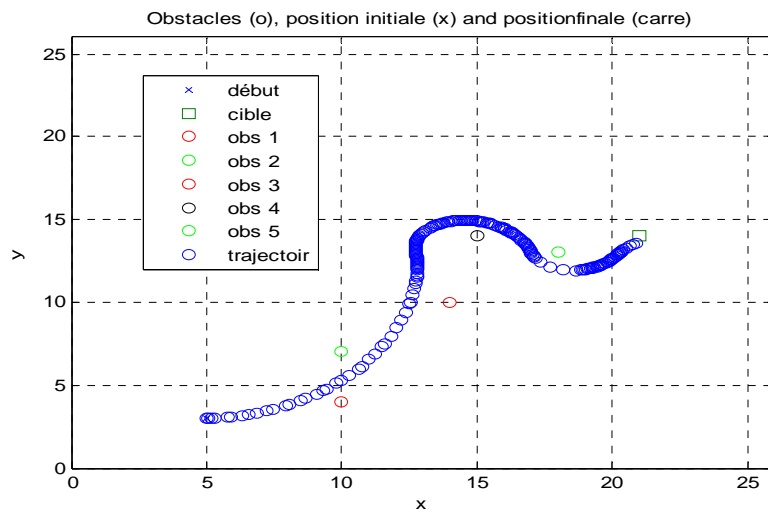


Figure 2.31 : Navigation avec évitement de cinq obstacles

## II.4 Conclusion

Nous avons proposé dans ce chapitre des algorithmes de navigation d'un robot mobile qui sont basées sur la technique des champs de potentiels et la logique floue. La planification de trajectoires avec évitement d'obstacles d'un robot guidé par des champs de potentiels, qui est présentée, est une voie simple de technique de planification qui a un principe de fonctionnement intuitif basé sur les champs de type énergie.

La logique floue est un outil puissant, simple à implémenter. Ces contrôleurs présentent néanmoins quelques inconvénients, entre autre la nécessité d'évaluer et d'affiner les contrôleurs (les fonctions d'appartenance d'entrée /sortie et les tables d'inférences). Cette évaluation est difficile à cause des règles d'inférence et nécessite plus de temps, c'est-à-dire les règles doivent être construites par un expert, qui peut définir pour chaque règle la conclusion correspondante.

La faisabilité du contrôleur flou intelligent pour mettre en œuvre le calcul approximatif de raisonnement nécessaire pour la manipulation de l'incertitude, inhérente au problème d'évitement d'obstacles, avec le minimum de capteurs, a été présentée dans ce chapitre.

Tous les travaux de simulation entrepris ont montré l'efficacité des deux techniques. Les résultats obtenus montrent bien ces évitements. De plus pour un environnement connu et structuré, ces méthodes peuvent être utilisées en temps réel par des algorithmiques appropriées.



---

**CHAPITRE 3**

**COMMANDE PAR MODE GLISSANT**

---

# COMMANDE PAR MODE GLISSANT

### III.1 Introduction

A partir de la fin de l'année 1970 jusqu'à aujourd'hui, le contrôle par mode glissant a reçu beaucoup d'attention en raison de son insensibilité aux perturbations et aux variations des paramètres. Le contrôle par mode glissant est un type particulier de contrôle des systèmes par structure variable (VS). Récemment, de nombreuses applications pratiques de contrôle par mode glissant (SMC) ont établi l'importance de la théorie du mode glissant qui a été principalement développée au cours des trois dernières décennies. Ce fait est également attesté par de nombreux numéros spéciaux de revues scientifiques axées sur le contrôle par mode glissant [54,55]. La recherche dans ce domaine a été initiée par Emel'yanov et ses collaborateurs [56,57] et le paradigme de conception forme maintenant une approche mature et établie pour le contrôle robuste et l'estimation. Le concept du contrôle par mode glissant est parvenu à la communauté de contrôle après l'apparition d'un article publié par Utkin [22] et d'un livre publié par Itkis [58].

La Conception du SMC peut être divisée en deux sous-parties, la première constitue la conception d'une surface stable et la deuxième s'articule sur la conception d'une loi de contrôle pour forcer les états du système sur les surfaces en temps fini. La conception de la surface doit répondre à toutes les contraintes requises, elle doit donc être conçue de manière optimale pour répondre à toutes les exigences. Pour éliminer la phase d'atteinte non robuste, un mode glissant intégral, proposé dans [59,60], permet de combiner naturellement le SMC avec la technique  $H_\infty$ ; cela étant très important d'un point de vue pratique. Dans [61,62] une variation de la surface de glissement est proposée pour éliminer la phase d'atteinte. L'efficacité du SMC dans le contrôle robuste des systèmes linéaires incertains a incité les chercheurs dans le domaine du contrôle à l'élargir à d'autres types de systèmes. Ainsi, certains chercheurs ont travaillé sur le contrôle en mode glissant pour les systèmes non linéaires et les systèmes de temporisation [63,64].

Le principe de la commande par mode glissant est basé sur l'atténuation du système à la surface de glissement en fonction des objectifs de la commande fixant la dynamique en boucle fermée : c'est le mode de convergence. La synthèse d'une commande discontinue

permet aux trajectoires du système à atteindre et rester sur la surface de glissement et le mouvement qui se produit est appelé mouvement de glissement.

La trajectoire dans le plan de phase est constituée de trois parties différentes [69] :

- Le mode de convergence
- Le mode de glissement
- Le mode du régime permanent

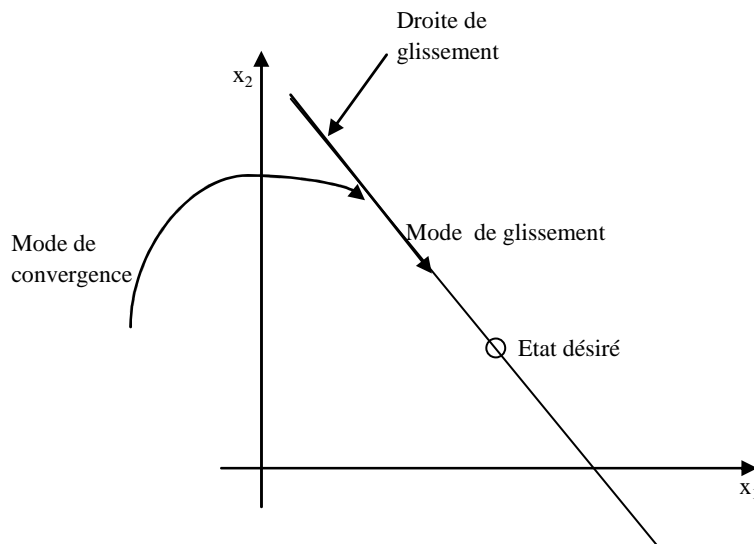


Figure 3.1: Modes de convergence pour la trajectoire d'état

Dans ce chapitre, on va présenter une brève introduction au contrôle par mode glissant avec les différents types de commande. On s'intéresse dans notre travail à la commande par mode glissant en temps fini (Finite Time Sliding Mode Control) qui sera par la suite validée par des travaux de simulations.

### III.2 Conception de la commande par mode glissant classique

Les avantages de la commande par mode glissant sont importants et multiples, entre autres, la haute précision, la bonne stabilité, la simplicité, l'invariance, la robustesse, etc.... Ceci lui permet d'être particulièrement adaptée pour les systèmes ayant des modèles imprécis [70].

La conception de la commande par mode glissant classique peut être effectuée en trois étapes principales très dépendantes l'une de l'autre :

- Choix de la surface.
- Etablissement des conditions d'existence.
- Détermination de la loi de commande.

### III.2.1 Choix de la surface de glissement

La surface de glissement peut être linéaire ou non linéaire. Elle est construite de telle sorte que le système ait toujours une dynamique désirée et est généralement choisie avec un degré relatif égal à un (i.e. la dérivée de la surface  $S(x)$  par rapport au temps fait apparaître explicitement la commande  $u$ ). Pour les systèmes d'ordre entier, plusieurs formes d'écriture sont proposées dans [67].

Considérons une classe de systèmes non linéaires, le vecteur de surface a la même dimension que le vecteur de commande  $u$ .

$$\dot{x} = f(x,t) + g(x,t)u \quad (3.1)$$

où  $x = (x_1, x_2, x_3, \dots, x_n)^T$  est le vecteur d'état du système et  $u = (u_1, u_2, u_3, \dots, u_m)^T$  est le vecteur de commande.

$f$  et  $g$  sont des champs de vecteurs suffisamment différentiables.

La surface est donnée par :

$$S_j(x) = \sum_{i=1}^n c_{ji} \cdot e_i \quad (3.2)$$

où  $s_i(x) = 0$  est la fonction de commutation qui sépare l'espace d'état en deux parties disjointes  $s(x,t) < 0$  et  $s(x,t) > 0$ .

$S(x) = [S_1(x), S_2(x), \dots, S_m(x)]$  est le vecteur de la surface de glissement.

$x_{ref_i} = [x_{ref_1}, x_{ref_2}, \dots, x_{ref_n}]^T$  est le vecteur de référence de l'état désiré.

$e_i = x_i - x_{ref_i}$  représente l'erreur entre la variable à commander et sa référence.

$e_i = [e_1, e_2, \dots, e_n]^T$  est le vecteur d'écart.

$c_{ij}$  représentent les paramètres de la surface de glissement.

$C$  est la matrice  $m \times n$  des paramètres de la surface.

Le choix des paramètres de la surface se fait de façon à satisfaire les performances désirées. Slotine propose une forme d'équation générale pour déterminer la surface de glissement qui assure la convergence d'une variable vers sa valeur désirée [60].

$$S(x) = \left(\frac{\partial}{\partial t} + \lambda\right)^{r-1} \cdot e(x) \quad (3.3)$$

Avec :  $e = x - x_{ref}$

$\lambda$ : Une constante positive

$r$  : degré relatif, égal au nombre de fois qu'il faut dériver la sortie pour faire apparaître la commande.

$s(x) = 0$  est une équation différentielle linéaire dont l'unique solution est  $e(x) = 0$ , ceci implique que l'état du système est égal à l'état désiré.

### III.2.2 Conditions de convergence et d'existence

Les conditions d'existence et de convergence sont les critères qui permettent aux différentes dynamiques du système de converger vers la surface de glissement et d'y rester, indépendamment de la perturbation. Il existe deux considérations pour assurer le mode de convergence.

#### III.2.2.1 Fonction discrète de commutation

Cette approche est la plus ancienne (mode glissant classique). Elle est proposée par Emelyanov et Utkin. Il s'agit de donner à la surface une dynamique convergente vers zéro. Elle est donnée par:

$$U_s = \begin{cases} \dot{S}(x) > 0 \dots \dots \dots S(x, t) < 0 \\ \dot{S}(x) < 0 \dots \dots \dots S(x, t) > 0 \end{cases} \quad (3.4)$$

Cette condition peut être formulée comme suit:

$$\dot{S}(x) \cdot S(x) < 0 \quad (3.5)$$

Elle est globale mais ne garantit pas en revanche un temps d'accès fini. Cette condition est toute fois difficile à utiliser pour faire la synthèse de la loi de commande.

### III.2.2.2 Fonction de Lyapunov

La fonction de Lyapunov est une fonction scalaire positive  $V(x) > 0$  pour les variables d'état du système. La loi de commande doit faire décroître cette fonction,  $\dot{V}(x) < 0$ . L'idée est de choisir une fonction scalaire  $S(x)$  pour garantir l'attraction de la variable à contrôler vers sa valeur de référence et de concevoir une commande  $u$  telle que le carré de la surface correspond à une fonction de Lyapunov.

La fonction de Lyapunov est définie comme suit:

$$V(x) = \frac{1}{2} S^T(x) S(x) \quad (3.6)$$

La dérivée de cette fonction est:

$$\dot{V}(x) = S^T(x) \dot{S}(x) \quad (3.7)$$

Pour que la fonction  $V(x) > 0$  puisse décroître, il suffit d'assurer que  $\dot{V}(x) < 0$ .

### III.2.3 Détermination de la loi de commande

Une fois la surface de glissement et le critère de convergence choisis, il reste à déterminer la condition nécessaire pour ramener la variable à contrôler vers la surface et ensuite vers son point d'équilibre en maintenant le mode de glissement.

#### III.2.3.1 Méthode de Philippov

Cette approche est basée sur les résultats des travaux effectués par ce dernier sur les équations différentielles à second membre discontinu. Il propose une loi de commande qui prend deux valeurs suivant le signe de la surface de glissement :

$$u = \begin{cases} u^+(x) & \text{si } S(x,t) > 0 \\ u^-(x) & \text{si } S(x,t) < 0 \end{cases} \quad (3.8)$$

$u^-$  et  $u^+$  sont des fonctions continues avec  $u^+ \neq u^-$ .

Cette commande  $u$  de nature discontinue va contraindre la trajectoire du système à atteindre la surface de glissement et de rester au voisinage de celle-ci malgré la présence de perturbations.

### III.2.3.2 Méthode de la commande équivalente d'Utkin

Cette méthode décrit la dynamique du système sur la surface de glissement. Elle permet de garder la variable contrôlée sur cette surface en imposant la valeur désirée en régime permanent de la grandeur d'entrée. Pour un système contrôlé donné, il faut trouver un vecteur d'entrée équivalent  $u_{eq}$  de sorte que la trajectoire d'état du système à contrôler reste sur la surface de glissement prédéfinie.

Lorsque le système est en régime glissant, la trajectoire restera sur la surface de commutation.

Cela peut être exprimé par  $S(x,t) = 0$  et  $\dot{S}(x,t) = 0$

La commande équivalente est une fonction continue qui sert à maintenir la variable à contrôler sur la surface de glissement qui est donnée par le scalaire :

$$u_{eq} = -\left(\frac{\partial S}{\partial x} g(x)\right)^{-1} \frac{\partial S}{\partial x} f(x) \quad (3.9)$$

### III.3. la commande par mode glissant basée sur la loi d'arrivée

Le mode glissant basé sur la loi d'arrivée comprend la phase d'arrivée et la phase de glissement. La phase d'arrivée amène le système à la droite de glissement (manifold) et la phase de glissement assure le glissement à l'équilibre. L'idée du mode glissant peut être décrite sur la figure 3.2.

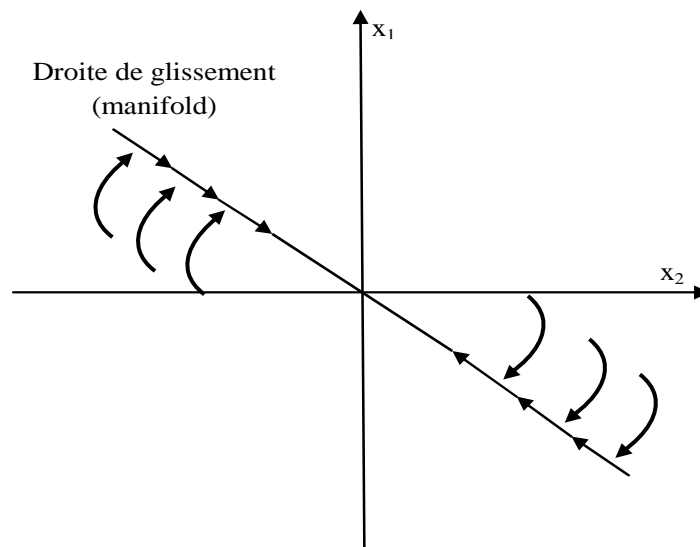


Figure 3.2: Approche du mode glissant

La forme générale de la loi d'arrivée est :

$$\dot{s} = -\varepsilon \operatorname{sgn}(s) - f(s) \quad (3.10)$$

où :

$$f(0)=0 \text{ et } s.f(s)>0 \text{ si } s \neq 0.$$

Pour la loi d'arrivée classique, trois types sont présentés dans [68] sont présentées par la suite.

### III.3.1 Loi d'arrivée avec une vitesse constante

La loi d'arrivée est définie comme suit :

$$\dot{s} = -\varepsilon \operatorname{sgn}(s) \quad (3.11)$$

où  $\varepsilon > 0$  représente la vitesse constante.

Cette loi contraint la variable de commutation à atteindre la droite de glissement  $S$  avec une vitesse constante  $\varepsilon$ . L'avantage de cette loi d'arrivée est sa simplicité. Cependant si  $\varepsilon$  est trop petit, le temps d'arrivée sera trop long et si  $\varepsilon$  est trop grand cela va causer un phénomène de broutement (chattering).

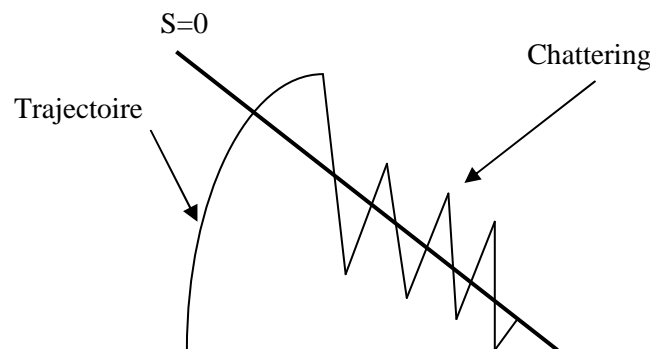


Figure 3.3: Phénomène de broutement (chattering)

### III.3.2 Loi d'arrivée exponentielle

La loi d'arrivée est définie comme suit :

$$\dot{s} = -\varepsilon \operatorname{sgn}(s) - Ks \quad (3.12)$$

Où  $\varepsilon > 0$ ,  $K > 0$  et  $\dot{s} = -Ks$  est le terme exponentiel, sa solution est donnée par  $s = s(0)e^{-Kt}$ .

Avec l'addition du terme de la vitesse proportionnelle  $-Ks$ , l'état du système est forcé d'approcher vers la droite de glissement rapidement dans le cas où  $S$  est large.



### III.3.3 Loi d'arrivée avec vitesse puissante

La loi est définie comme :

$$\dot{s} = -K|s|^\delta \operatorname{sgn}(s) \quad (3.13)$$

avec  $K > 0$ ,  $0 < \delta < 1$

Cette loi d'arrivée augmente la vitesse d'atteinte lorsque l'état est loin de la droite de commutation (manifold). Cependant, il réduit la vitesse lorsque l'état est proche de la droite. Le résultat est rapide et le phénomène de broutement est faible.

Il est évident que les trois lois d'arrivée ci-dessus peuvent satisfaire la condition du mode glissant qui est présentée dans l'équation (3.5).

### III.4. Commande par mode glissant basée sur le quasi-mode glissant

Dans les systèmes pratiques d'ingénierie, le phénomène de broutement pour la commande par mode glissant peut endommager les composants du système tels que les actionneurs. Une façon d'alléger ce phénomène est d'utiliser la méthode du quasi-mode glissant qui permet à l'état de rester dans une certaine gamme au voisinage de la surface. Souvent, cette gamme est nommée bande limite. Le principe de la commande par quasi-mode glissant est de remplacer les fonctions discontinues par des fonctions continues.

Dans un système continu, il existe deux méthodes communes pour la conception du quasi-mode glissant :

- La fonction de saturation
- La fonction relais

#### III.4.1. Quasi-mode glissant basé sur la fonction de saturation

La fonction de saturation est définie comme suit :

$$\operatorname{sat}(s) = \begin{cases} 1 & s < \partial \\ Ks & |s| \leq \partial \\ -1 & s < -\partial \end{cases} \quad (3.14)$$

où  $k = \frac{1}{\partial}$  avec  $\partial$  étant la bande limite.

En dehors de la bande limite, on utilise la commande basée sur les fonctions discontinues (au voisinage de la surface de commutation) et à l'intérieur de la bande limite, on utilise la commande linéaire par feedback.

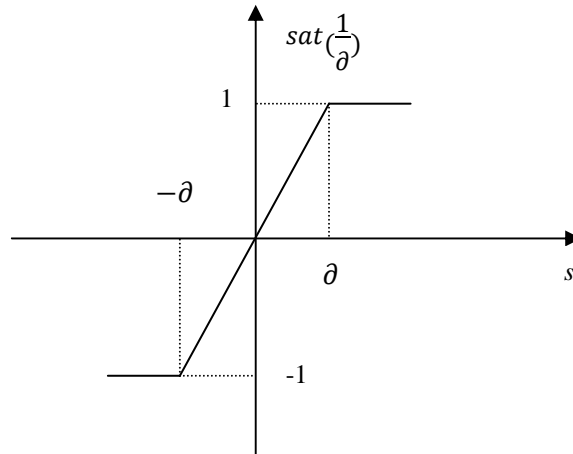


Figure 3.4: Fonction de saturation

#### III.4.2. Quasi-mode glissant basé sur la fonction relais

La fonction relais est définie par :

$$\theta(s) = \frac{s}{|s| + \delta} \quad (3.15)$$

$\delta$  est une constante très petite. Ce type de commande a un grand avantage dans la réduction du phénomène de chattering.

#### III.5. Mode glissant terminal (Terminal Sliding Mode)

La commande par mode glissant terminal ajoute des fonctions non linéaires dans la conception du plan de glissement supérieur. Ainsi, une surface de glissement terminale est construite quand les erreurs de suivi sur la surface de glissement convergent vers zéro dans un temps fini.

Ce type de commande offre des propriétés supérieures telles que la rapidité et la convergence vers la droite de glissement en temps fini [69].

On distingue dans la littérature deux types de commande par mode glissant terminal :

- Mode glissant terminal normal
- Mode glissant terminal rapide

### III.5.1. Mode glissant terminal normal (Normal terminal sliding mode)

Le mode glissant terminal normal est proposé par Feng et al [70].

On considère le système suivant :

$$\begin{cases} \dot{x}_1 = x_2 \\ \dot{x}_2 = f(x) + g(x)u \end{cases} \quad (3.16)$$

où  $x_1$  et  $x_2$  sont les états du système,  $g(x)$  et  $f(x)$  sont des fonctions non linéaires inconnues et  $u$  est la commande.

La surface de glissement est choisie comme :

$$s = x_2 + \frac{1}{\beta} x_1^{q/p} \quad (3.17)$$

où  $x \in R$  est l'état,  $\beta > 0$ ,  $p, q$  ( $p > q$ ) sont des nombres impairs positifs et  $\frac{q}{p} - 1 < 0$ .

*Analyse en temps fini :*

On suppose que le temps d'arrivée est  $t_r$  à partir de  $s(0) \neq 0$  à  $s=0$ . Quand  $t = t_r$ , on aura.  $s(t_r) = 0$  quand  $s \geq 0$  et  $s\dot{s} \leq -\mu|s|$ , on peut avoir :  $\dot{s} \leq -\mu$ .

$$\int_{s(0)}^{s(t_r)} ds \leq \int_{t=0}^{t=t_r} -\mu dt \quad (3.18)$$

$$s(t_r) - s(0) \leq -\mu t_r \quad (3.19)$$

$$t_r \leq \frac{s(0)}{\mu} \quad (3.20)$$

Dans le même temps si  $s \leq 0$ ,  $t_r \leq -\frac{s(0)}{\mu}$ , on aura alors

$$t_r \leq \frac{|s(0)|}{\mu} \quad (3.21)$$

On suppose que le temps d'arrivée est  $t_s$  de  $x_1(t_r) \neq 0$  à  $x_1(t_s + t_r) = 0$ . Dans cette phase  $s = 0$ .

$$x_2 + \frac{1}{\beta} x_1^{q/p} = 0 \quad (3.22)$$

$$\dot{x}_1 = -\frac{1}{\beta} x_1^{q/p} \quad (3.23)$$

Par l'intégration de l'équation différentielle précédente on aura:

$$\int_{x_1(t_r)}^0 x_1^{-\frac{q}{p}} dx_1 = \int_{t_r}^{t_r+t_s} -\frac{1}{\beta} dt \quad (3.24)$$

$$-\frac{p}{p-q} x_1^{1-\frac{q}{p}}(t_r) = -\frac{1}{\beta} t_s$$

$$t_s = \frac{\beta p}{(p-q)} |x_1(t_0)|^{1-\frac{q}{p}} \quad (3.25)$$

### III.5.2. Mode glissant terminal rapide (Fast terminal sliding mode)

La commande par mode glissant terminal rapide peut faire converger les états du système vers zéro en temps fini. La convergence asymptotique des états dans le mode glissant normal est surmontée. La caractéristique de la convergence du mode glissant terminal rapide est supérieure à celle du mode glissant normal. De plus, il n'y a pas de fonctions de commutation pour le mode glissant terminal, par conséquent le phénomène de broutement est évitable.

Pour la conception d'un contrôleur basé sur la commande par mode glissant rapide, on distingue deux types de surface de glissement :

#### A. Surface de glissement terminale traditionnelle

Park et al [71], ont proposé une surface de glissement terminale rapide qui est définie comme suit:

$$s = \dot{x} + \beta x^{(q/p)} = 0 \quad (3.26)$$

A partir de l'équation (3.26) on obtient :

$$\frac{dx}{dt} = -\beta x^{q/p} \quad (3.27)$$

$$dt = -\frac{1}{\beta} x^{-q/p} dx$$

Alors:

$$\int_0^t dt = \int_{x_0}^0 -\frac{1}{\beta} x^{-q/p} dx \quad (3.28)$$

L'intervalle de temps que l'état initial  $x(0) \neq 0$  prend pour atteindre l'équilibre  $x = 0$  le long de la surface de glissement (3.27) est:

$$t_s = \frac{P}{\beta(p-q)} |x(0)|^{(p-q)/p} \quad (3.29)$$

L'équilibre  $x = 0$  est appelé l'attracteur terminal. La vitesse de convergence augmente à mesure que la partie non linéaire  $\beta x^{q/p}$  est introduite. Cependant, le temps de convergence de la commande par mode de glissement terminal peut ne pas être optimal. En effet, la vitesse de convergence de la surface de glissement non linéaire est plus lente que celle de la loi linéaire ( $p = q$ ) lorsque l'état se rapproche de l'équilibre.

### **B. Surface de glissement terminale globale**

Une nouvelle surface de glissement non linéaire proposée par Park qui est appelée surface de glissement terminale rapide et globale est définie comme suit:

$$s = \dot{x} + \alpha x + \beta x^{q/p} = 0 \quad (3.30)$$

où  $\alpha$  représente une constante positive.

L'équation (3.30) donne :

$$\dot{x} = -\alpha x - \beta x^{q/p} \quad (3.31)$$

Dévisant l'équation (3.31) par  $x^{q/p}$  :

$$\frac{-q}{x^p} dx \frac{dx}{dt} + \alpha x^{1-\frac{q}{p}} = -\beta \quad (3.32)$$

On pose :  $y = x^{1-\frac{q}{p}}$  alors:

$$\frac{dy}{dt} = \frac{p-q}{p} x^{-\frac{q}{p}} \frac{dx}{dt} \quad (3.33)$$

Selon l'équation différentielle linéaire de premier ordre :

$$\frac{dy}{dx} + p(x)y = \varphi(x) \quad (3.34)$$

Cette dernière donne :

$$y = \exp\left(\int p(x) dx\right) \left( \int \varphi(x) \exp\left(\int p(x) dx\right) dx + c \right) \quad (3.35)$$

Lorsque  $t=0$ ,  $c=y(0)$  la solution est :

$$\frac{\beta + \alpha y(0)}{\beta} = \exp\left(\frac{p-q}{p} \alpha t_s\right) \quad (3.36)$$

Par conséquent, à partir de l'équation (3.36), le temps de convergence peut être obtenu à partir de l'état initial arbitraire.

$$t_s = \frac{p}{\alpha(p-q)} \ln \frac{\alpha x(0)^{\frac{p-q}{p}} + \beta}{\beta} \quad (3.37)$$

Lorsque l'état  $x$  est loin de l'origine, le temps de convergence est déterminé par l'attracteur terminal rapide  $\dot{x} = -\beta x^{q/p}$ , lorsque l'état  $x$  se rapproche de l'origine  $x=0$ , le temps de convergence est déterminé par l'équation  $\dot{x} = -\alpha x$ . Exponentiellement  $x$  converge vers zéro. Par conséquent, l'attracteur terminal introduit dans l'équation de surface de glissement (3.30) fait converger l'état vers zéro dans un temps fini. De plus, la vitesse de la surface de glissement linéaire est garantie. Par conséquence, l'état peut converger vers l'équilibre rapidement et précisément. L'équation de surface de glissement est appelée surface de glissement rapide globale.

Une autre surface de glissement rapide globale est proposée par Zhang dans [72] où l'état est défini comme une surface. La surface est définie comme suit :

$$s = \dot{s} + \alpha s + \beta s^{q/p} \quad (3.38)$$

### III.6. Conception d'un contrôleur basé sur la commande par mode glissant rapide pour le suivi de trajectoires d'un robot mobile

La commande par mode glissant est l'une des techniques qui sont utilisées pour résoudre le problème de suivi de trajectoires pour les robots mobiles. La commande par mode glissant rapide d'un robot mobile assure la convergence du robot vers la trajectoire de référence en un temps fini.

Pour la commande des robots mobiles, on distingue deux types de commande selon le modèle du robot utilisé :

- Commande cinématique
- Commande dynamique

### III.6.1. Commande cinématique

La commande cinématique du robot mobile exige le modèle cinématique de ce dernier. Dans le premier chapitre on a étudié les modèles cinématiques des différents types de robots mobiles, dans cette section du chapitre on s'intéresse à un robot mobile de type unicycle.

#### III.6.1.1. Modélisation cinématique

Comme défini au premier chapitre, le modèle cinématique du robot mobile unicycle est défini comme suit :

$$\dot{p} = \begin{pmatrix} \dot{x} \\ \dot{y} \\ \dot{\theta} \end{pmatrix} = \begin{bmatrix} \cos \theta & 0 \\ \sin \theta & 0 \\ 1 & 0 \end{bmatrix} q \quad (3.39)$$

$$\text{avec } q = (v, w)^T, S(q) = \begin{bmatrix} \cos \theta & 0 \\ \sin \theta & 0 \\ 1 & 0 \end{bmatrix}$$

Pour le suivi de trajectoires, le robot mobile se déplace de la posture  $p = (x, y, \theta)^T$  vers la posture de référence  $p_r = (x_r, y_r, \theta_r)^T$  comme il est mentionné sur la figure (3.5), alors l'erreur de posture est donnée par  $p_e = (x_e, y_e, \theta_e)^T$ .

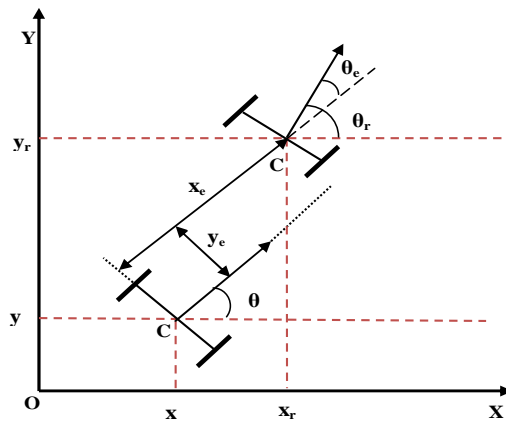


Figure 3.5: Présentation de l'erreur de posture

Selon la transformation des coordonnées, l'équation d'erreur de posture du robot mobile est décrite comme dans [73]:

$$p_e = \begin{bmatrix} x_e \\ y_e \\ \theta_e \end{bmatrix} = \begin{bmatrix} \cos\theta & \sin\theta & 0 \\ -\sin\theta & \cos\theta & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x_r - x \\ y_r - y \\ \theta_r - \theta \end{bmatrix} \quad (3.40)$$

avec  $T = \begin{bmatrix} \cos\theta & \sin\theta & 0 \\ -\sin\theta & \cos\theta & 0 \\ 0 & 0 & 1 \end{bmatrix}$  une matrice de transformation.

Les dérivées des erreurs du suivi de trajectoires qui sont définies dans l'équation (3.40) peuvent être écrites comme :

$$\begin{aligned} \dot{x}_e &= (\dot{x}_r - \dot{x})\cos\theta + (\dot{y}_r - \dot{y})\sin\theta - (x_r - x)\dot{\theta}\sin\theta + (y_r - y)\dot{\theta}\cos\theta \\ &= y_e\omega - v + \dot{x}_r \cos(\theta_r - \theta) + \dot{y}_r \sin(\theta_r - \theta) = y_e\omega - v + v_r \cos\theta_e \end{aligned} \quad (3.41)$$

$$\begin{aligned} \dot{y}_e &= -(\dot{x}_r - \dot{x})\sin\theta + (\dot{y}_r - \dot{y})\cos\theta - (x_r - x)\dot{\theta}\cos\theta - (y_r - y)\dot{\theta}\sin\theta \\ &= -x_e\omega - \dot{x}_r \sin(\theta_r - \theta) + \dot{y}_r \cos(\theta_r - \theta) = -x_e\omega + v_r \sin\theta_e \end{aligned} \quad (3.42)$$

$$\dot{\theta}_e = \dot{\theta}_r - \dot{\theta} = \omega_r - \omega \quad (3.43)$$

A partir des équations (3.41), (3.42) et (3.43) on obtient :

$$\dot{p}_e = \begin{bmatrix} \dot{x}_e \\ \dot{y}_e \\ \dot{\theta}_e \end{bmatrix} = \begin{bmatrix} y_e\omega - v + v_r \cos\theta_e \\ -x_e\omega + v_r \sin\theta_e \\ \omega_r - \omega \end{bmatrix} \quad (3.44)$$

### III.6.1.2. Conception du contrôleur cinématique pour le suivi de la trajectoire

La stratégie du contrôleur cinématique pour le suivi de la trajectoire d'un robot mobile est schématisée par la figure ci-dessous :



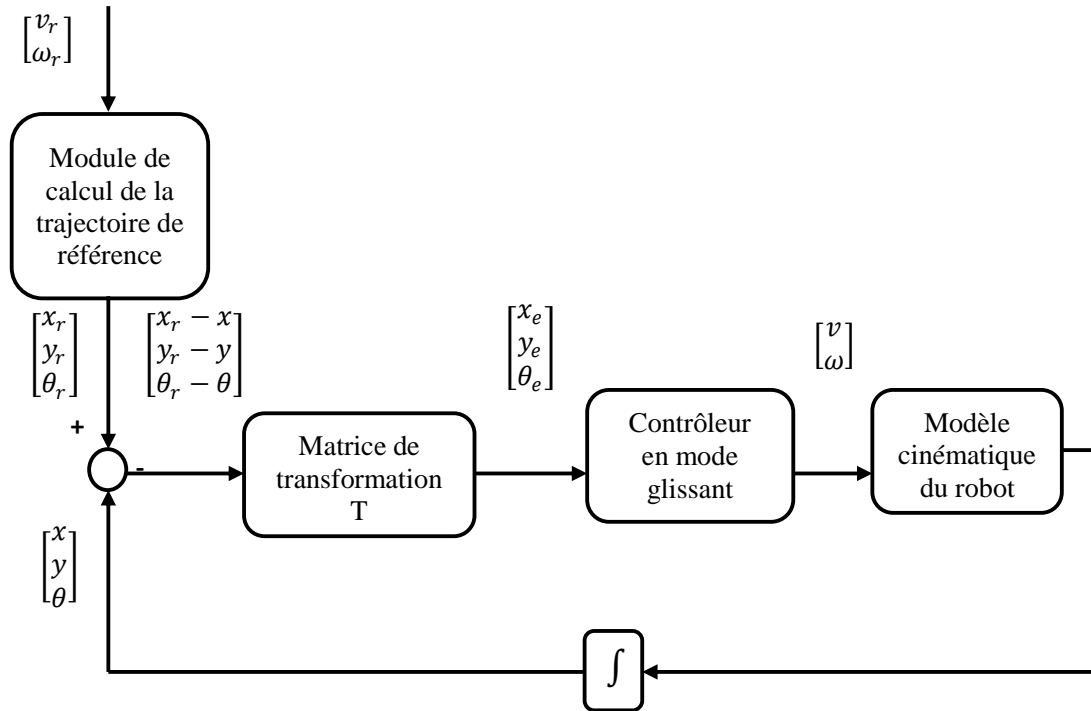


Figure 3.6: Stratégie de commande par mode glissant terminal rapide

Le processus de contrôle par mode glissant peut être divisé en deux étapes:

1. Le choix d'une surface de glissement appropriée de sorte que si la trajectoire du système est confinée sur elle, alors le système présente le comportement souhaité.
2. La détermination d'une loi de contrôle discontinue sur la droite de glissement, capable de forcer la trajectoire du système à atteindre cette dernière et à y rester.

L'équation (3.40) pour le robot mobile représente un système non linéaire à entrées multiples. Par conséquent, la conception de la surface de glissement est un problème difficile. Pour simplifier le problème, le mode de glissement terminal rapide et global est considéré pour le contrôleur proposé.

#### A. Choix des surfaces de glissement

Avant de choisir le type de la surface de glissement, il faut d'abord chercher le nombre de surfaces qu'il faut choisir. Le nombre de surfaces de glissement dépend des lois de commande, pour le contrôleur proposé on choisit deux surfaces de glissement :

- La première surface en fonction de l'erreur angulaire ( $\theta_e$ ), qui est une surface terminale rapide et globale proposée dans l'équation (3.30) :

$$s_1 = \dot{\theta}_e + \alpha_1 \theta_e + \beta_1 \theta_e^{q/p} = 0 \quad (3.45)$$

- La deuxième en fonctions des erreurs de position ( $x_e$  et  $y_e$ ), qui est une surface rapide et globale proposée dans l'équation (3.38) :

$$s_2 = \dot{s}_2 + \alpha_2 s_2 + \beta_2 s_2^{q/p} \quad (3.46)$$

### **B. Détermination des lois de commande**

Pour le contrôleur proposé dans cette partie, on distingue deux lois de commande : la loi de la vitesse angulaire et la loi de la vitesse linéaire.

A partir de l'équation (3.45) on obtient :

$$\dot{\theta}_e = -\theta_e - \alpha_1 \theta_e - \beta_1 \theta_e^{q/p} \quad (3.47)$$

Remplaçant l'équation (3.43) dans l'équation (3.47) on obtient une loi de contrôle de la forme :

$$\omega = \omega_r + \alpha_1 \theta_e + \beta_1 \theta_e^{q/p} \quad (3.48)$$

qui représente la loi de commande de la vitesse angulaire.

A partir de l'équation (3.47), on choisit la fonction de Lyapunov comme :

$$V_1 = \frac{1}{2} \theta_e^2 \quad (3.49)$$

La dérivée de la fonction de Lyapunov peut être obtenue comme :

$$\begin{aligned} \dot{V}_1 &= \theta_e \dot{\theta}_e = \left( -\theta_e - \alpha_1 \theta_e - \beta_1 \theta_e^{q/p} \right) \theta_e \\ &= -\theta_e^2 - \alpha_1 \theta_e^2 - \beta_1 \theta_e^{q+p} \end{aligned} \quad (3.50)$$

Sachant que les termes  $p$ ,  $q$ ,  $\alpha$  et  $\beta$  sont positifs, alors  $\dot{V}_1 \leq 0$ . Donc le système est stable.

Lorsque  $\theta_e$  converge vers zéro, alors  $\omega_r = \omega$ . Deux autres conceptions de contrôle d'état sont considérées en ce moment:

$$\dot{x}_e = y_e - v + v_r \quad (3.51)$$

$$\dot{y}_e = -\omega_r x_e \quad (3.52)$$

Pour la deuxième surface de glissement, la fonction de commutation est désignée comme suit :

$$s_2 = x_e - y_e \quad (3.53)$$

L'objectif de la deuxième loi de commande est d'assurer la convergence de  $s_2$  vers 0, donc  $x_e$  converge vers  $y_e$  afin d'obtenir la convergence de  $x_e$  et  $y_e$  vers 0.

A partir des équations (3.46), (3.51), (3.52) et (3.53) on obtient :

$$\omega_r y_e - v + v_r + \omega_r x_e = -s_2 - \alpha_2 s_2 - \beta_2 s_2^{q/p} \quad (3.54)$$

Donc la loi de commande de la vitesse angulaire peut être obtenue comme suit :

$$v = v_r + \omega_r x_e + \omega_r y_e - s_2 - \alpha_2 s_2 - \beta_2 s_2^{q/p} \quad (3.55)$$

La fonction candidate de Lyapunov est définie comme :

$$V_2 = \frac{1}{2} s_2^2 \quad (3.56)$$

La dérivée de la fonction de Lyapunov est :

$$\begin{aligned} \dot{V}_2 &= s_2 \dot{s}_2 = s_2 (-s_2 - \alpha_2 s_2 - \beta_2 s_2^{(q/p)}) \\ &= -s_2^2 - \alpha_2 s_2^2 - \beta_2 s_2^{\frac{(q+1)}{p}} \end{aligned} \quad (3.57)$$

On remarque que  $\dot{V}_2 < 0$  avec  $\alpha > \beta$ , donc le système est stable.

### III.6.1.3. Résultats de simulations

Selon la loi de contrôle établie dans les sections précédentes, les simulations sont appliquées sur un robot mobile en utilisant Matlab/Simulink. Le modèle à contrôler est la différence entre les trajectoires désirées et réelles du robot mobile. Les simulations sont validées par différents types de trajectoires (circulaire, sinusoïdale, carré).

#### A. Trajectoire circulaire

Pour une trajectoire circulaire, on considère les valeurs suivantes :

$$v_r = 1 \text{ m/s}, \omega_r = 1 \text{ rad/s et } r = \frac{v_r}{\omega_r}.$$

$\alpha_1$	4	$\alpha_2$	4
$\beta_1$	8	$\beta_2$	2
$q_1$	5	$q_2$	3
$p_1$	7	$p_2$	5

Tableau 3.1: Valeurs des paramètres

La posture de référence est  $p_r = (x_r, y_r, \theta_r)^T$

$$\begin{cases} x_r = r \cos(\omega_r t) = \cos(t) \\ y_r = r \sin(\omega_r t) = \sin(t) \\ \theta_r = \omega_r t = t \end{cases} \quad (3.58)$$

Les valeurs des erreurs initiales sont  $(x_e, y_e, \theta_e) = (-1.5, 0.5, -\pi/4)$

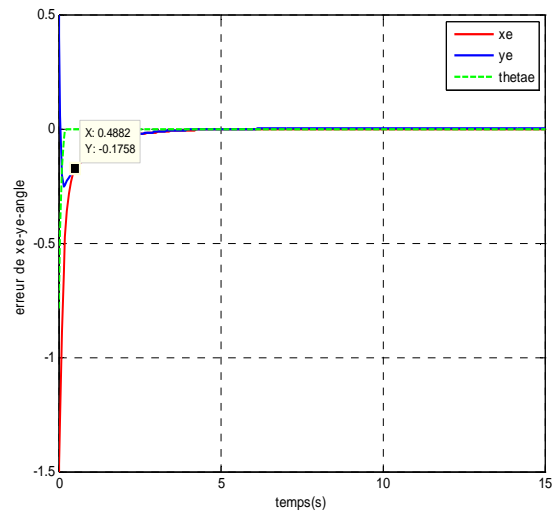
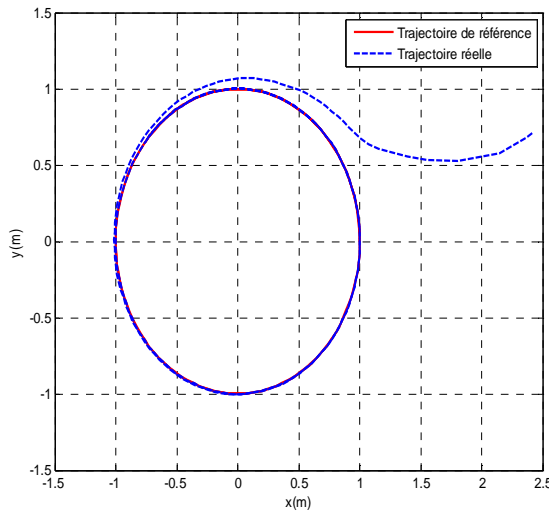


Figure 3.7: Suivi de trajectoire circulaire

Figure 3.8: Erreurs de position et orientation

La figure (3.7) montre que le robot mobile suit la trajectoire de référence dans un temps limité. Dans la figure (3.8), l'erreur  $\theta_e$  converge vers zéro en une fraction de seconde, mais les erreurs  $y_e$  et  $x_e$  convergent vers zéro dans un temps qui est supérieur à deux secondes ( $> 2s$ ). On remarque que  $x_e$  converge vers  $y_e$  dans un temps limité (0.488 s) qui est le temps de convergence de la deuxième surface vers zéro, ce qui traduit l'efficacité de la commande proposée pour atteindre la trajectoire de référence en un temps fini. D'après la figure (3.8) on peut confirmer le temps de convergence de la deuxième surface obtenu en simulation à partir de l'équation (3.37) :

$$t_s = \frac{5}{4(5-3)} \ln \frac{4(2)^{\frac{2}{5}} + 2}{2} = 0.52 \text{ s qui est presque égal au temps de convergence pratique.}$$

A partir des résultats obtenus, on peut aussi calculer l'erreur quadratique moyenne MSE (Mean Square Error) qui est définie par :

$$MSE = \sum_{n=0}^k \frac{\sqrt{x_{er}^2(n) + y_{er}^2(n)}}{k} \quad (3.59)$$

Avec :  $x_{er-tot} = \frac{\sum_{n=0}^k x_{er}(n)}{k}$  ,  $y_{er-tot} = \frac{\sum_{n=0}^k y_{er}(n)}{k}$  ,  $\theta_{er-tot} = \frac{\sum_{n=0}^k \theta_{er}(n)}{k}$

Les résultats numériques obtenus sont présentés dans le tableau suivant :

$x_{er-tot}$	0.023
$y_{er-tot}$	0.028
$\theta_{er-tot}$	0.0021
$MSE$	0.015

Tableau 3.2: Erreur quadratique moyenne

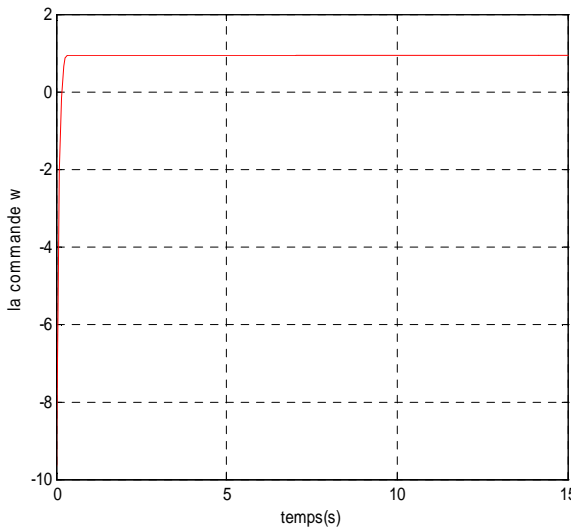


Figure 3.9: Vitesse angulaire  $\omega$

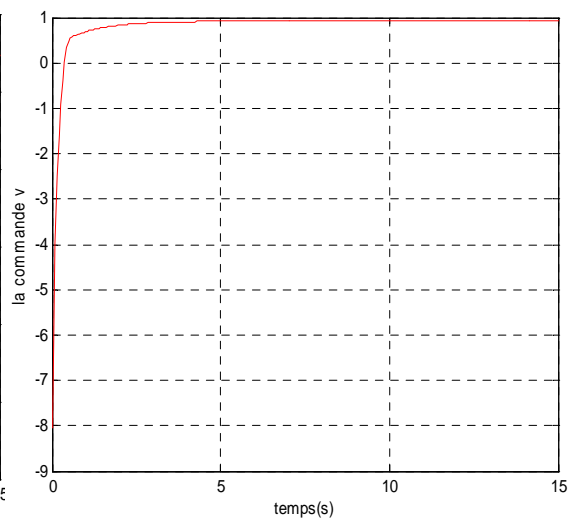


Figure 3.10: Vitesse linéaire  $v$

Les figures (3.9) et (3.10) représentent les variations de la vitesse linéaire et angulaire, on remarque que la vitesse angulaire converge vers la vitesse de référence en un temps très petit puisqu'elle dépend de la convergence de la première surface. Par contre, la vitesse linéaire converge vers la vitesse de référence après un temps de 5 secondes puisqu'elle dépend de la deuxième surface de glissement.

**B. Trajectoire carrée**

Pour une trajectoire carrée, on garde les mêmes valeurs des paramètres et des erreurs initiales.

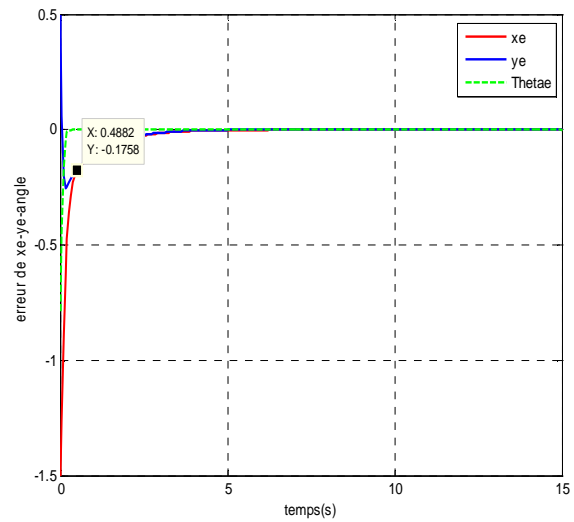
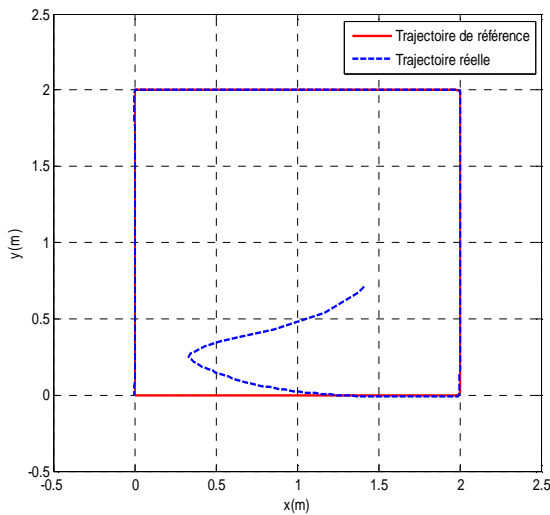


Figure 3.11: Suivi de trajectoire carrée

Figure 3.12: Erreurs de position et orientation

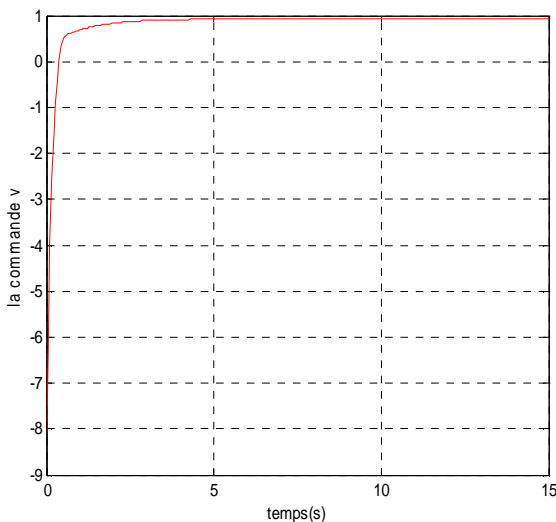


Figure 3.13: Vitesse linéaire  $v$

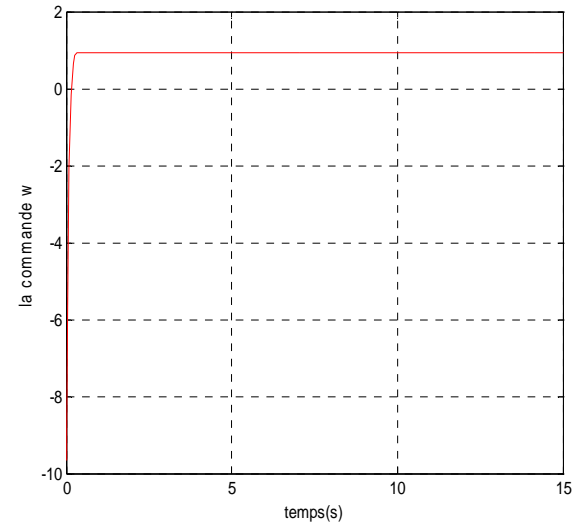


Figure 3.14: Vitesse angulaire  $\omega$

La figure (3.11) montre que le robot mobile converge vers la trajectoire de référence qui est carrée. La figure (3.12), l'erreur  $\theta_e$  converge vers zéro en une fraction de seconde, mais les erreurs  $y_e$  et  $x_e$  convergent vers zéro dans un temps de 3 secondes, ainsi on remarque que  $x_e$  converge vers  $y_e$  dans un temps limité (0.488 s) qui est identique au temps de convergence pour la trajectoire circulaire. Les figures (3.13) et (3.14) représentent la variation de la vitesse linéaire et angulaire. On remarque que les deux vitesses convergent vers les vitesses de référence, en un temps très faible pour la vitesse angulaire et en un plus important pour la vitesse linéaire.

**C. Trajectoire sinusoïdale**

On garde les mêmes valeurs des paramètres et des erreurs initiales.

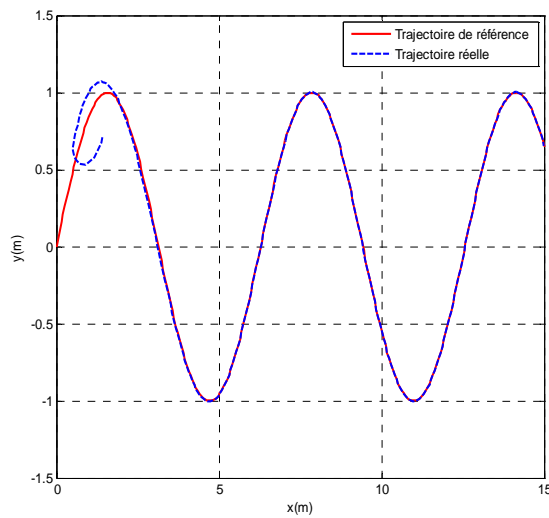


Figure 3.15: Suivi de trajectoire sinusoïdale

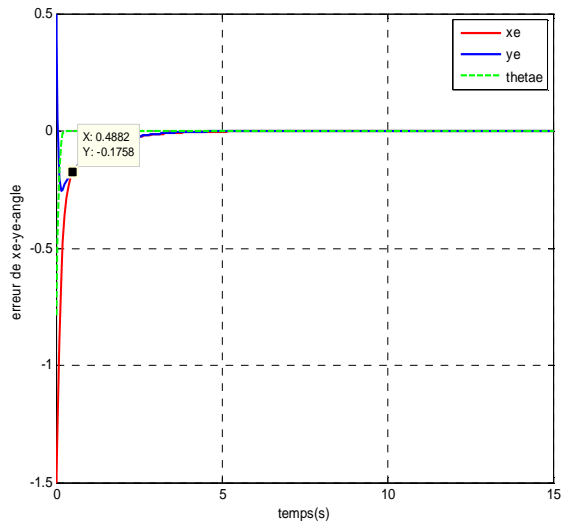


Figure 3.16: Erreurs de  $x_e$ ,  $y_e$  et  $\theta_e$

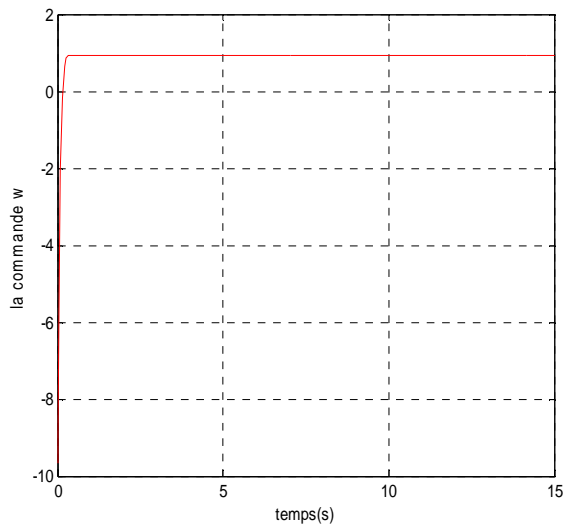


Figure 3.17: La vitesse angulaire  $\omega$

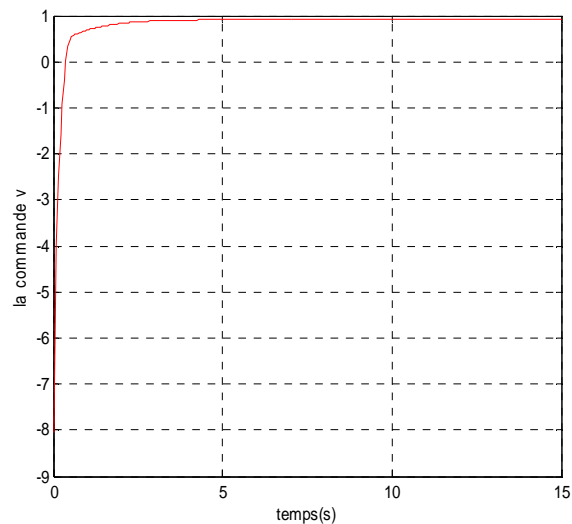


Figure 3.18: La vitesse linéaire  $v$

La figure (3.15) montre que la commande proposée assure la poursuite de la trajectoire sinusoïdale du robot. La figure (3.16) montre la convergence des erreurs en temps fini, ainsi on remarque que  $x_e$  converge vers  $y_e$  dans un temps fini même pour ce type de trajectoire (0.488 s) qui est théoriquement très logique puisqu'on analyse l'expression de temps de convergence qui est présenté par l'équation (3.38), on constate que ce dernier dépend des erreurs initiales et des paramètres  $p$ ,  $q$ ,  $\alpha$ ,  $\beta$ . Les figures (3.17) et (3.18) représentent la variation de la vitesse linéaire et angulaire.

Par l'analyse des résultats obtenus pour les différents types de trajectoires, on constate que la commande proposée assure une bonne convergence en un temps fini. Cela est interprété par la convergence des erreurs de position et angulaires vers zéro en un temps très acceptable.

### III.6.2. Commande dynamique

Ce type de commande consiste à concevoir une stratégie de la commande pour la poursuite dynamique du robot mobile. Cette stratégie est devisée en deux contrôleurs, un contrôleur cinématique et un contrôleur dynamique qui exige le modèle dynamique du robot.

Pour le contrôleur cinématique, on propose un contrôleur qui est basé sur la commande par mode glissant terminal pour assurer la convergence en temps fini des erreurs de poursuite vers zéro. Pour le contrôleur dynamique, on propose un contrôleur PID afin qu'il puisse assurer la convergence des vitesses réelles du robot vers les vitesses de commande générées par le contrôleur cinématique en absence de perturbations [74].

#### III.6.2.1. Modélisation dynamique

Le modèle dynamique général du robot mobile, détaillé dans le premier chapitre, est défini par l'équation suivante :

$$M(q)\ddot{q} + V(q, \dot{q})\dot{q} + F(\dot{q}) + G(q) + \tau_d = B(q)\tau - A^T(q)\lambda \quad (3.60)$$

où :

$M(q)$  : est une matrice d'inertie définie symétrique et positive.

$V(q, \dot{q})$  : est la matrice centripète et coriolis.

$F(\dot{q})$  : est la matrice de la surface de friction.

$G(q)$  : est la matrice de gravité.

$\tau_d$  : est le vecteur des perturbations inconnues bornées, y compris des dynamiques non structurées.

$B(q)$  : est la matrice de transformation des entrées.

$\tau$  : est le vecteur des torques.

$A^T(q)$  : est la matrice associée aux contraintes.

$\lambda$  : est le vecteur des forces de contrainte.



Le système défini par l'équation (3.60) peut être transformé en une représentation plus appropriée à des fins de contrôle et de simulation, qui est présentée par l'équation suivante :

$$Mh(q)\ddot{q} + Vh(q, \dot{q})\dot{q} + F(\dot{q}) + \tau_h = Bh(q)\tau \quad (3.61)$$

$G(q) = 0$  parce que le mouvement est contraint sur le sol.

Pour notre cas de simulation on considère que les perturbations et la surface de friction sont nulles, alors l'équation à contrôler (3.61) devient :

$$Mh(q)\ddot{q} + Vh(q, \dot{q})\dot{q} = Bh(q)\tau \quad (3.62)$$

### III.6.2.2. Conception du contrôleur dynamique pour le suivi de trajectoires

La stratégie de la commande dynamique pour le suivi de trajectoires d'un robot mobile est schématisée par la figure ci-dessous :

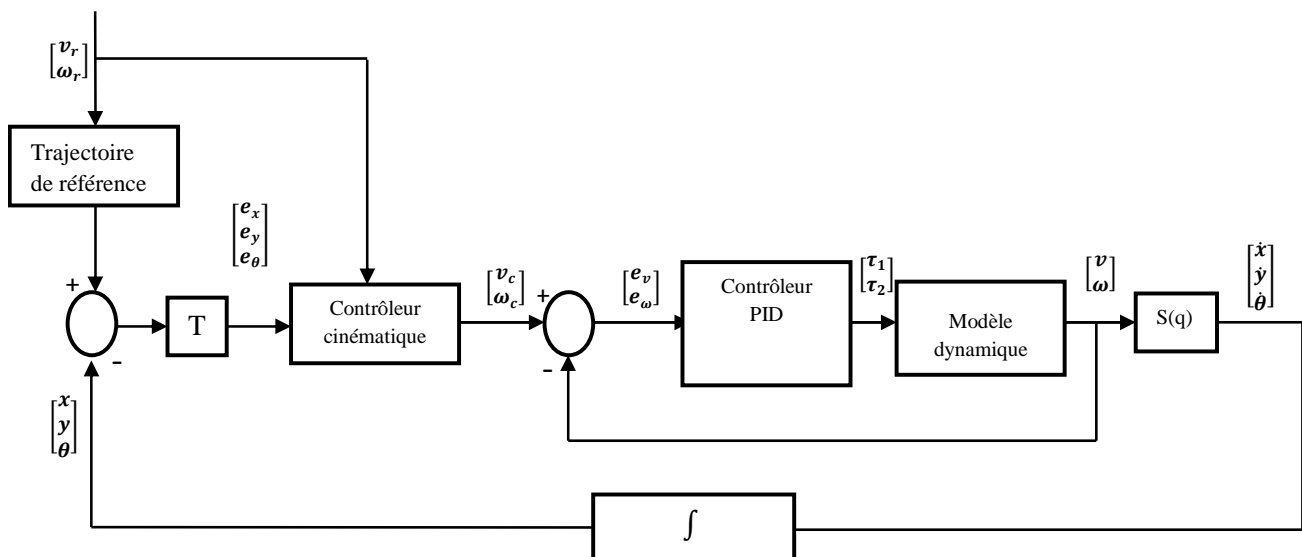


Figure 3.19: Stratégie de la commande dynamique basée sur un contrôleur PID

D'après la figure ci-dessus on distingue deux contrôleurs :

#### A. Contrôleur cinématique

Le contrôleur cinématique est basé sur la commande par mode glissant. Pour la première surface on propose une surface terminale rapide et globale qui est une surface en fonction de l'erreur angulaire  $\theta_e$ . Pour la deuxième surface on propose un autre type de commande par mode glissant qui est le mode glissant basé sur la loi d'arrivée avec une vitesse puissante. La deuxième surface est en fonction des erreurs de position  $x_e$  et  $y_e$ .

L'équation suivante présente les deux surfaces proposées :

$$s = \begin{bmatrix} s_1 \\ s_2 \end{bmatrix} = \begin{bmatrix} \dot{\theta}_e + \alpha\theta_e + \beta\theta_e^{\frac{q}{p}} \\ x_e - y_e \end{bmatrix} \quad (3.63)$$

A partir de l'équation (3.63) on peut établir les lois de commande :

$$\dot{\theta}_e = -\alpha\theta_e - \beta\theta_e^{q/p} \quad (3.64)$$

Combinant les équations (3.44) et (3.64) on trouve la première loi de commande :

$$\omega = \omega_r + \alpha\theta_e + \beta\theta_e^{\frac{q}{p}} \quad (3.65)$$

La stabilité est vérifiée dans l'équation (3.50).

La dérivée de la surface  $s_2$  est donnée par l'équation :

$$\dot{s}_2 = \dot{x}_e - \dot{y}_e \quad (3.66)$$

La loi d'arrivée avec une vitesse puissante est présentée dans l'équation (3.13), on remplace l'équation (3.66) dans l'équation (3.13) on trouve :

$$\dot{x}_e - \dot{y}_e = -k|s_2|^\delta \operatorname{sgn}(s_2) \quad (3.67)$$

A partir des équations (3.51), (3.52) et (3.67) on trouve :

$$w_r y_e - v + v_r + w_r x_e = -k|s_2|^\delta \operatorname{sgn}(s_2) \quad (3.68)$$

La deuxième loi de commande cinématique est exprimée comme suit :

$$v = \omega_r y_e + v_r + \omega_r x_e + k|s_2|^\delta \operatorname{sgn}(s_2) \quad (3.69)$$

La fonction candidate de Lyapunov est choisie comme :

$$V = \frac{1}{2} s_2^2 \quad (3.70)$$

La dérivée de cette équation est donnée par :

$$\dot{V} = s_2 \dot{s}_2 = S_2 \left( -k|s_2|^\delta \operatorname{sgn}(s_2) \right) = -k|s_2|^{\delta+1} \leq 0 \quad (3.71)$$

### ***B. Contrôleur dynamique***

Notre objectif de contrôle s'établit comme suit:

- On propose une trajectoire désirée  $q_d$

- On doit concevoir un contrôleur qui applique un couple adéquat  $\tau$  tel que les positions mesurées  $q$  atteignent la référence souhaitée  $q_d$ :

$$\lim_{t \rightarrow \infty} (\|q_d(t) - q(t)\|) = 0 \quad (3.72)$$

Le contrôleur dynamique proposé est basé sur un contrôleur PID qui a pour but de trouver une entrée de commande  $\tau$  telle que le vecteur de vitesse actuel  $\vartheta$  peut atteindre le vecteur de vitesse  $\vartheta_c$  :

$$\lim_{t \rightarrow \infty} (\|\vartheta_c - \vartheta\|) = 0 \quad (3.73)$$

Avec :  $\vartheta = \begin{bmatrix} v \\ \omega \end{bmatrix}$  et  $\vartheta_c = \begin{bmatrix} v_c \\ \omega_c \end{bmatrix}$

Comme le montre la figure (3.18), le régulateur PID possède deux variables d'entrée correspondantes aux erreurs de vitesse obtenues et deux variables de sortie qui sont les couples d'entraînement et de rotation  $\tau$ .

$$T_o = k_p e_{\vartheta} + \int_0^t k_i e_{\vartheta}(\tau) + k_d \frac{de_{\vartheta}}{dt} \quad (3.74)$$

$$e_{\vartheta} = \begin{bmatrix} e_v \\ e_{\omega} \end{bmatrix} = \begin{bmatrix} v_c - v \\ \omega_c - \omega \end{bmatrix} \quad (3.75)$$

### III.6.2.3. Résultats de simulations

Les simulations sont appliquées sur le robot mobile en utilisant Matlab/Simulink. Le modèle à contrôler est la différence entre les trajectoires désirées et réelles du robot mobile.

On considère les valeurs :  $v_r = 1$  m/s,  $\omega_r = 1$  rad/s,  $\alpha=4$ ,  $\beta=8$ ,  $p=5$ ,  $q=3$ ,  $K=15$ ,  $\delta=0.5$ .

La trajectoire de référence circulaire est définie par l'équation (3.58). Les valeurs des erreurs initiales :  $(x_e, y_e, \theta_e) = (0.1, 0, 0)$ .

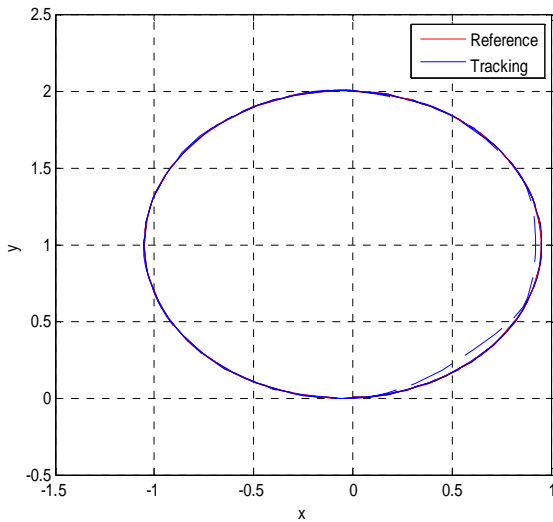


Figure 3.20: Suivi de trajectoire circulaire

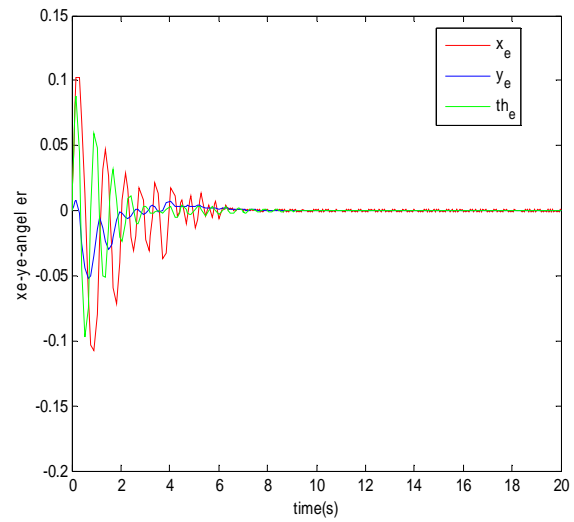


Figure 3.21: Erreurs  $x_e$ ,  $y_e$  et  $\theta_e$

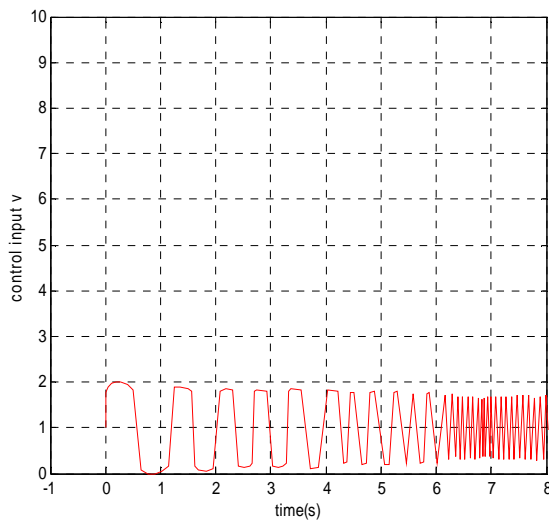


Figure 3.22: Vitesse linéaire  $v$

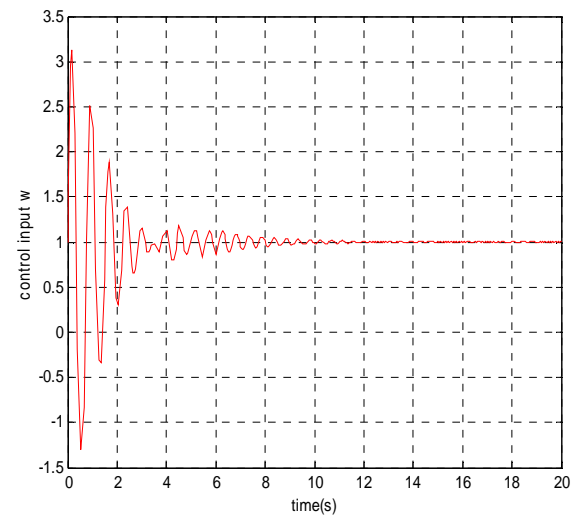


Figure 3.23: Vitesse angulaire  $\omega$

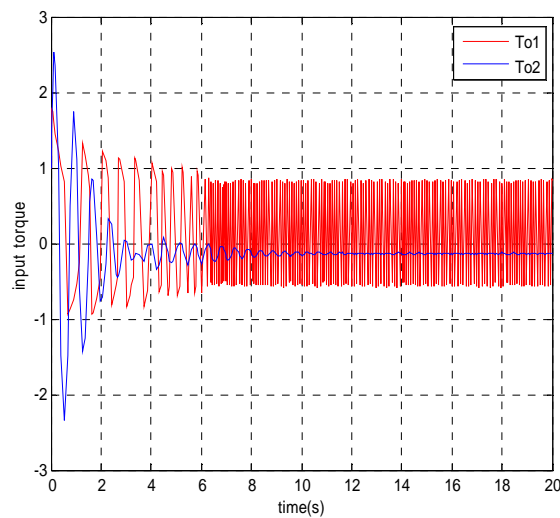


Figure 3.24: Couples  $\tau_1$  et  $\tau_2$

La figure (3.20) montre la poursuite de la trajectoire de référence par le robot mobile. Dans la figure (3.21), l'erreur  $\theta_e$  converge vers zéro en temps fini suite à la commande proposée mais l'erreur  $x_e$  et  $y_e$  convergent vers zéro après un temps très important par rapport aux états initiaux donnés.

Les figures (3.22) et (3.23) représentent la variation de la vitesse linéaire et angulaire respectivement, on remarque que la vitesse angulaire converge vers la vitesse de référence qui vaut 1 en temps fini puisqu'elle dépend de la convergence de la commande par mode glissant terminal rapide, par contre la vitesse linéaire ne converge pas vers la vitesse de référence et provoque ainsi un effet de broutement, ce qui traduit les performances limitées de la commande proposée pour la deuxième surface de glissement.

### **III.7 Conclusion**

Nous avons proposé dans ce chapitre la commande par mode glissant notamment en temps fini pour la poursuite de trajectoire d'un robot mobile. Nous avons proposé en premier temps une commande cinématique qui est basée sur une commande terminale rapide où la contrainte du temps limité est prouvée par les différentes simulations. En deuxième temps on a proposé une commande dynamique qui est basée sur deux contrôleurs : un contrôleur cinématique qui s'articule autour de la commande par mode glissant terminal rapide, et un contrôleur dynamique en PID en absence de perturbations.

Les travaux de simulation ont montré l'efficacité des commandes proposées. Pour La commande dynamique, on ne peut pas parler sur l'efficacité et la robustesse de cette dernière sans faire intervenir les perturbations. Dans le chapitre qui suit on va proposer une commande robuste en présence des perturbations.

---

**CHAPITRE 4**

**COMMANDE ROBUSTE APPLIQUEE A LA POURSUITE**

**DE TRAJECTOIRES D'UN ROBOT MOBILE**

---

# COMMANDE ROBUSTE APPLIQUEE A LA POURSUITE DE TRAJECTOIRES D'UN ROBOT MOBILE

### IV.1 Introduction

Pour résoudre les problèmes de suivi de trajectoires, certaines techniques sont déployées dans la littérature. Des techniques de contrôle par mode glissant pour les robots mobiles sont proposées dans [75,76]; un problème similaire avec des perturbations bornées est considéré dans [77]. Les lois de contrôle, qui assurent la convergence en temps fini sont introduites dans [78,79]. En combinant la conception en cascade et l'approche en retour, un contrôleur de suivi est conçu dans [80,81] où la prise en compte de la saturation du couple d'entrée et des perturbations externes est introduite. Une approche basée sur la logique floue pour le suivi du chemin du robot mobile est traitée dans [82].

Le contrôleur neuronal peut traiter les perturbations bornées non modélisables et / ou les dynamiques non structurées du robot mobile. Par conséquent, une structure de contrôle qui rend possible l'intégration d'un contrôleur cinématique et d'un contrôleur dynamique de couple basé sur les réseaux de neurones est présentée dans [83]. Un contrôleur dynamique neuro-fuzzy (NFN) pour les robots mobiles est présenté dans [84], avec une loi combinée de contrôle cinématique / dynamique utilisant la théorie backstepping pour la commande et la théorie de Lyapunov pour la garantie de la stabilité.

Dans ce chapitre, une commande dynamique pour le suivi de trajectoires d'un robot mobile utilisant le mode glissant rapide global et les réseaux de neurones (NN-GFSM) est présentée. La stratégie proposée combine deux approches de contrôle, le contrôle cinématique et le contrôle dynamique. Les lois de contrôle cinématique sont basées sur le GFSM afin de déterminer les vitesses adéquates pour la stabilité du système en temps fini. Pour le contrôleur dynamique deux techniques de contrôle sont combinées, le GFSM pour la stabilité des erreurs de vitesse et le contrôleur neuronal pour l'approximation de la fonction non linéaire et le traitement des perturbations. Ce contrôleur dynamique permet aux robots de suivre la trajectoire désirée même en présence de perturbations.

## IV.2 Réseaux de neurones artificiels (RNA) appliqués dans la commande des systèmes

Le principe des réseaux de neurones artificiels est apparu dans les années 40 à partir d'une analogie avec le système nerveux humain. Les réseaux de neurones artificiels sont modélisés sur un processus biologique pour le traitement de l'information, incluant spécifiquement le système nerveux. Les signaux se propagent sous forme de différences de potentielles entre l'intérieur et l'extérieur des cellules

Les réseaux de neurones sont recommandés pour le contrôle intelligent dans le cadre de structures bien connues [85-87]. Récemment de nombreuses recherches ont été effectuées sur les applications des réseaux de neurones pour la commande non-linéaire des processus dynamiques [88,89]. Ces travaux sont soutenus par deux caractéristiques importantes pour les réseaux de neurones : l'apprentissage et la bonne performance pour l'approximation des fonctions non linéaires [90]. Actuellement la plupart des travaux concernant le contrôle des systèmes, utilisant les réseaux de neurones sont basés sur l'apprentissage avec rétro-propagation pour les réseaux multicouches. Il est montré qu'un réseau de neurones avec une couche cachée comportant un grand nombre de neurones, peut approximer toutes les fonctions continues.

La commande des robots mobiles basée sur les réseaux de neurones a fait l'objet d'intenses recherches [91]. Fierro et Lewis [83] ont développé un modèle neuronal pour le contrôle du robot mobile, basé sur une combinaison de la technique backstepping et d'un contrôleur dynamique pour les couples, en utilisant un réseau neuronal multi-couches. Ce dernier peut apprendre la dynamique du robot mobile par un apprentissage en ligne.

Deux des propriétés les plus importantes des réseaux de neurones sont la classification (pour la reconnaissance de formes) et l'approximation de la fonction non linéaire. La théorie de l'approximation universelle de base dit que toute fonction  $f(x)$  peut être approchée arbitrairement de près par un ensemble compact utilisant un RNA à deux couches avec des poids appropriés. Plus précisément, soit  $f(x)$  une fonction générale lisse. Alors, étant donné un ensemble compact  $S$  et un nombre positif  $\varepsilon_N$ , il existe un RNA à deux couches tel que:

$$f(x) = W^T \sigma(V^T x) + \varepsilon \quad (4.1)$$

Avec  $\|\varepsilon\| = \varepsilon_N$  pour tout  $x$  dans l'ensemble compact  $S$ , pour une valeur suffisamment grande  $L$  de neurones de couches cachées. La valeur  $\varepsilon$  est appelée l'erreur d'approximation de la



fonction, il diminue à mesure que le nombre de neurones de couches cachées  $L$  augmente. Le réseau de neurones agissant comme un approximateur de fonction est montré sur la figure (4.1).

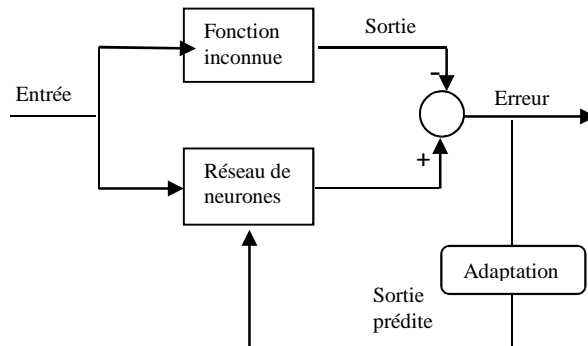


Figure 4.1: Approximation d'une fonction par RNA

#### IV.2.1 Modèle mathématique d'un réseau neuronal

Le modèle mathématique d'un neurone est représenté sur la figure (4.2):

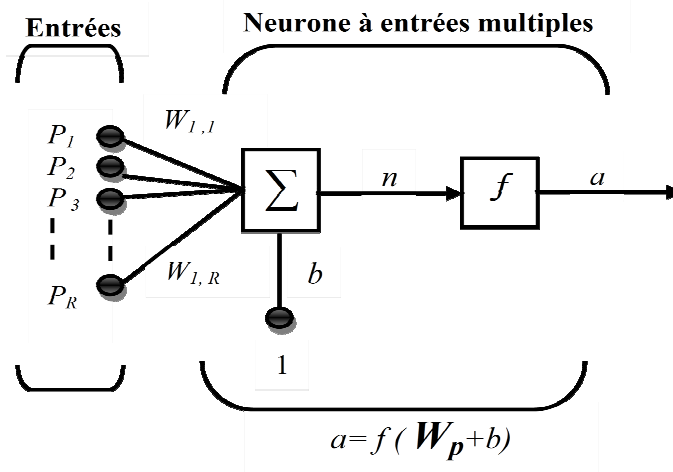


Figure 4.2: Modèle mathématique d'un réseau de neurone

Les poids d'entrée  $w_j$ , le seuil de déclenchement  $b$  (également appelé biais), la somme des entrées pondérées et la fonction d'activation non linéaire  $f$  sont représentés sur cette figure. Si les entrées de la cellule sont  $n$  signaux à l'instant  $k$ ,  $x_1(k)$ ,  $x_2(k)$ ,  $x_3(k)$ , ...  $x_n(k)$  et la sortie est le scalaire  $y(k)$ , l'équation mathématique du neurone peut être écrite comme suit:

$$y(k) = f\left(\sum_{j=1}^n w_j x_j(k) + b\right) \quad (4.2)$$

Les fonctions d'activation sont sélectionnées selon l'application choisie, ces fonctions sont présentées comme suit :

- Fonction à seuil :  $y = x$
- Fonction sigmoïde :  $y = \frac{1}{1+e^{-x}}$
- Fonction tangente hyperbolique :  $y = \frac{e^x - e^{-x}}{e^x + e^{-x}}$

Dans la littérature deux grandes familles de RNA sont présentées. En l'occurrence, il s'agit des réseaux à fonctions de base radiales (RBF : Radial Basis Function) et les réseaux à perceptron multi-couches (MLP : Multi Layer Perceptron).

#### IV.2.2 Réseaux de neurones MLP

L'architecture d'un réseau MLP est formée de plusieurs couches de neurones sans communication à l'intérieur d'une même couche :

- Une couche d'entrée qui représente les entrées auxquelles sont transmises les données à traiter.
- Une ou plusieurs couches cachées effectuant le traitement spécifique du réseau.
- Une couche en sortie qui délivre les résultats.

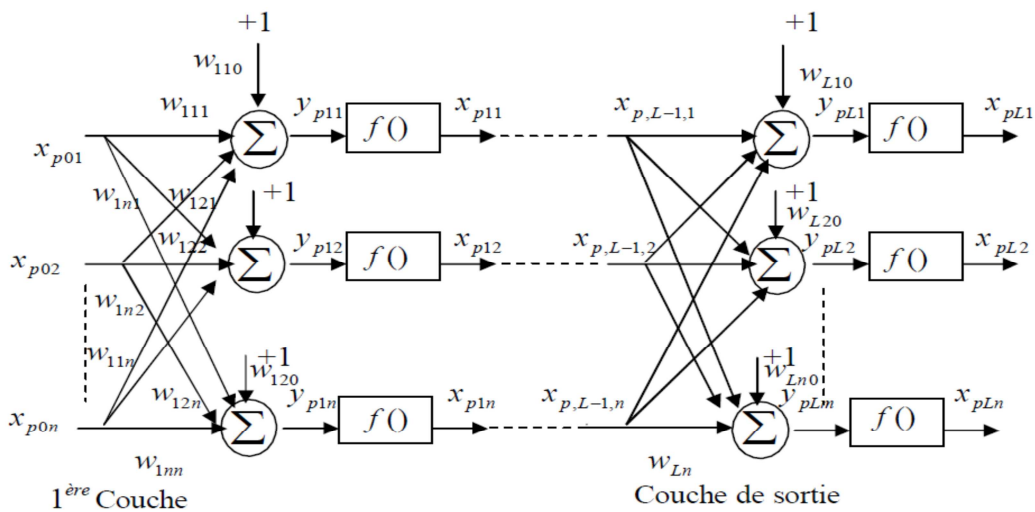


Figure 4.3: Architecture d'un réseau MLP [92].

La figure précédente représente l'architecture d'un réseau MLP avec les définitions suivantes:

$w_{j,k,i}$  : Connexion entre le neurone  $k$  de la couche  $j-1$  et le neurone  $i$  de la couche  $j$ .

$y_{p,j,k}$  : Entrée totale du neurone  $k$  pour l'échantillon  $p$  de la couche  $j$ .

$w_{j,k,0} = \theta_{jk}$  : Poids fictif du neurone  $k$  de la couche  $j$  correspondant à un biais dont l'entrée est fixée à 1.

L'entrée totale du nœud  $k$  pour la couche  $j$  est :

$$y_{p,j,k} = \sum_{i=0}^n w_{j,k,i} x_{p,j-1,i} \quad (4.3)$$

La sortie de ce nœud est :

$$x_{p,j,k} = F(y_{p,j,k}) \quad (4.4)$$

L'utilité des réseaux de neurones dans la commande est généralement basée sur la propriété d'approximation des réseaux MLP (réduire l'erreur minimale à une valeur arbitrairement petite). On emploie les réseaux MLP pour l'approximation des fonctions non linéaires incertaines.

### IV.2.3 Réseaux de neurones RBF

Les réseaux à fonctions de base radiales sont relativement des classes récentes de RNA. Ils se composent d'une entrée, une couche cachée et une couche de sortie. La figure ci-dessous représente un réseau RBF :

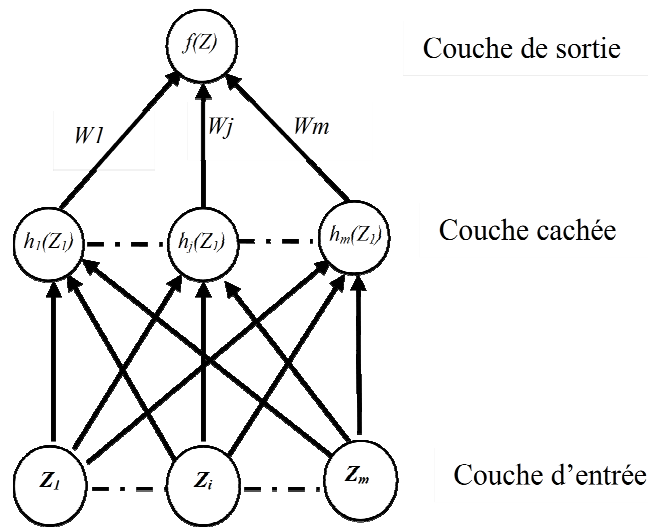


Figure 4.4: Architecture d'un réseau RBF

Les réseaux RBF sont utilisés de manière adaptative dans notre travail pour approximer la fonction non linéaire incertaine. L'algorithme des réseaux à fonction de base radiale (RBF) est défini dans [93] comme:

$$h_j = g(z - c_{ij}^2) / b_j^2 \quad (4.5)$$

et

$$f = W^t h(z) + \varepsilon \quad (4.6)$$

où  $z$  est l'état d'entrée du réseau,  $i$  est le numéro d'entrée du réseau,  $j$  est le nombre de nœuds de couches cachés dans le réseau,  $h=[h_1 \ h_2 \ \dots \ h_n]^T$  est la sortie de la fonction gaussienne et  $W$  est le poids du réseau neuronal avec  $\varepsilon \leq \varepsilon_N$ .

L'approximation de la fonction  $f$  est utilisée. La sortie du réseau RBF est:

$$\hat{f}(z) = \hat{W}^T h(z) \quad (4.7)$$

La fonction d'activation utilisée pour le RBF est généralement la fonction gaussienne qui est définie comme suit :

$$h(z) = \exp\left(-\frac{r^2}{2\sigma^2}\right) \quad (4.8)$$

Pour approximer une fonction non linéaire, on se base sur la propriété d'approximation des réseaux RBF [88].

### IV.3 Application d'une commande neuronale robuste pour la poursuite de trajectoires d'un robot mobile

La commande dynamique proposée dans ce chapitre a pour but d'améliorer les résultats obtenus dans le chapitre précédent en termes de robustesse par injection des perturbations et en termes de convergence vers la trajectoire désirée en temps fini. Comme nous avons vu précédemment dans le troisième chapitre, le contrôle dynamique proposé pour la poursuite du robot se fait sans prise en compte des perturbations, donc on ne peut pas parler d'une robustesse. Dans ce contexte et dans le but d'apporter une solution à la poursuite en présence de perturbations, un bloc neuronal est ajouté afin de permettre une commande robuste avec approximation de la fonction non linéaire du système affecté par des perturbations [94].

La stratégie de contrôle proposée dans ce chapitre est conçue avec deux contrôleurs, un contrôleur cinématique et un contrôleur dynamique.

#### IV.3.1. Contrôleur cinématique

Le contrôleur cinématique proposé est basé sur ce qui est démontré dans le chapitre précédent, où les lois de commandes  $w_c$  et  $v_c$  sont présentées par les équations (3.48) et (3.55) respectivement :

$$w = w_r + \alpha_1 \theta_e + \beta_1 \theta_e^{\frac{q}{p}}$$

$$v = v_r + \omega_r x_e + \omega_r y_e - s_2 - \alpha_2 s_2 - \beta_2 s_2^{q/p}$$

Les fonctions de Lyapunov choisies sont présentées par les équations (3.49) et (3.56) :

$$V_1 = \frac{1}{2} \theta_e^2$$

$$V_2 = \frac{1}{2} s_2^2$$

où la stabilité a été prouvée.

### IV.3.2. Contrôleur dynamique

Pour le contrôle dynamique, on va présenter deux cas :

- Le cas où les perturbations ne sont pas considérées où un contrôleur basé sur la commande GFSM est proposé.
- Le cas où les perturbations sont considérées où un contrôleur basé sur la commande NN-GFSM est proposé.

#### IV.3.2.1. Contrôle en absence de perturbations

Dans ce cas, la stratégie de contrôle est donnée par la figure suivante :

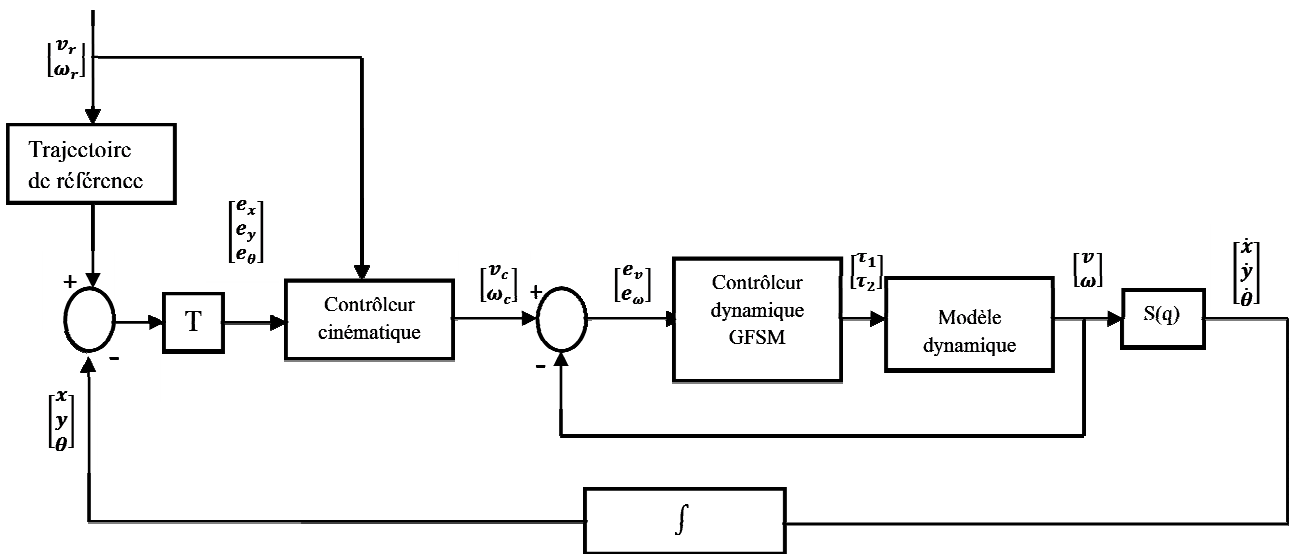


Figure 4.5 : Stratégie de la commande dynamique basée sur l'approche GFSM

Le modèle dynamique du robot mobile sans considération des perturbations est donné par l'équation (3.62) :

$$Mh(q)\ddot{q} + Vh(q, \dot{q})\dot{q} = Bh(q)\tau$$

Les erreurs des vitesses sont définies par l'équation suivante :

$$e_{\vartheta} = \vartheta_c - \vartheta = \begin{bmatrix} e_v \\ e_{\omega} \end{bmatrix} = \begin{bmatrix} v_c - v \\ \omega_c - \omega \end{bmatrix} \quad (4.9)$$

La dérivée de l'équation (4.9) est obtenue comme :

$$\dot{e}_{\vartheta} = \dot{\vartheta}_c - \dot{\vartheta} \quad (4.10)$$

A partir de l'équation (4.10), l'équation (3.62) peut réécrire comme suit :

$$\begin{aligned} Mh(q)(\dot{\vartheta}_c - \dot{e}_{\vartheta}) + Vh(q, \dot{q})(\vartheta_c - e_{\vartheta}) &= Bh(q)\tau \\ Mh(q)\dot{\vartheta}_c - Mh(q)\dot{e}_{\vartheta} + Vh(q, \dot{q})\vartheta_c - Vh(q, \dot{q})e_{\vartheta} &= Bh(q)\tau \end{aligned} \quad (4.11)$$

La fonction non linéaire du système est définie comme suit :

$$f(z) = Mh(q)\dot{\vartheta}_c + Vh(q, \dot{q})\vartheta_c \quad (4.12)$$

Avec  $z = [\vartheta_c^T \ \dot{\vartheta}_c^T \ \vartheta^T]^T$

Remplaçant l'équation (4.12) dans (4.11), on trouve :

$$-Mh(q)\dot{e}_{\vartheta} - Vh(q, \dot{q})e_{\vartheta} + f(z) = Bh(q)\tau \quad (4.13)$$

Pour obtenir la loi de commande dynamique, on choisit la surface de glissement comme :

$$s_3 = e_{\vartheta} \quad (4.14)$$

A partir de l'équation (3.38) qui présente la surface globale rapide et l'équation (4.14), l'équation (4.15) est établie comme suit :

$$\dot{e}_{\vartheta} = -(\alpha_3 + 1)e_{\vartheta} - \beta_3 e_{\vartheta}^{q_3/p_3} \quad (4.15)$$

On remplace l'équation (4.15) dans (4.13) :

$$Mh(q)((\alpha_3 + 1)e_{\vartheta} + \beta_3 e_{\vartheta}^{q_3/p_3}) - Vh(q, \dot{q})e_{\vartheta} + f(z) = Bh(q)\tau \quad (4.16)$$

Alors, la loi de commande obtenue est donnée par :

$$\tau = Bh^{-1}(C_v e_{\vartheta} + f(z) + \beta_3 Mh(q)e_{\vartheta}^{q_3/p_3}) \quad (4.17)$$

Avec :

$$C_v = Mh(q)(\alpha_3 + 1) - Vh(q, \dot{q}) \quad (4.18)$$

### IV.3.2.2. Contrôle en présence de perturbations

Dans ce cas un contrôleur neuronal est introduit, la stratégie de contrôle est donnée par la figure suivante :

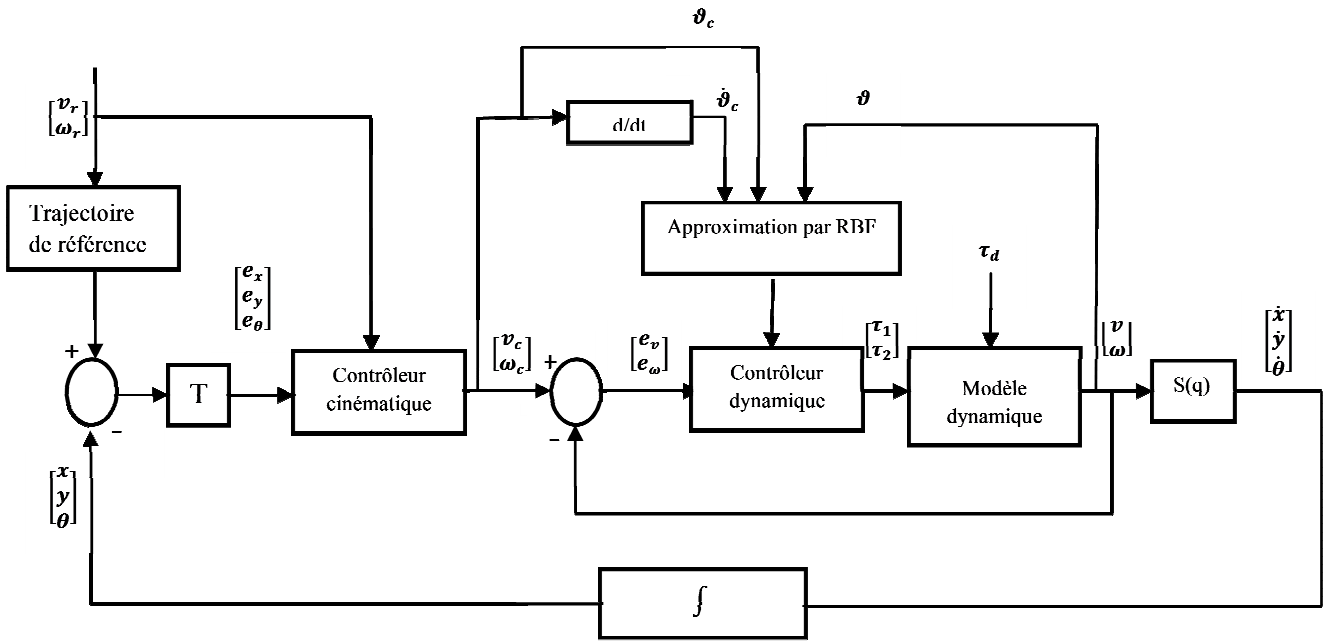


Figure 4.6: Stratégie de la commande dynamique basée sur NN-GFSM

L'équation dynamique du robot en présence des perturbations est donnée par l'équation (3.61) :

$$Mh(q)\ddot{q} + Vh(q, \dot{q})\dot{q} + F(\dot{q}) + \tau_h = Bh(q)\tau$$

Donc l'équation (4.13) peut se réécrire comme suit :

$$-Mh(q)\dot{e}_v - Vh(q, \dot{q})e_v + f(z) - \tau_d = Bh(q)\tau \quad (4.19)$$

La loi de contrôle désignée dans l'équation (4.17) devient :

$$\tau = Bh^{-1}(C_v e_v + \hat{f}(z) + \beta_3 Mh(q) e_v^{q_3/p_3}) \quad (4.20)$$

où  $\hat{f}(z)$  est la sortie du réseau RBF, et  $\hat{f}(z)$  approxime  $f(z)$ .

Le réseau RBF peut être adopté pour approximer  $f(z)$ . L'algorithme d'approximation du réseau RBF est donné comme :

$$\phi_i = g(\|z - c_i\|^2 / b_i^2) \quad i=1,2,\dots,n \quad (4.21)$$

$$\zeta = W^{*T} \varphi(z) \quad (4.22)$$

$$f(z) = W^{*T} \varphi(z) + \varepsilon \quad (4.23)$$

où  $z$  est l'état des entrées du réseau,  $\varphi(z) = [\varphi_1 \varphi_2 \dots \varphi_n]^T$  est la fonction non linéaire,  $\varepsilon$  est l'erreur d'approximation et  $W^*$  est le vecteur de poids du réseau RBF.

On remplace l'équation (4.20) dans (4.19) on trouve :

$$Mh(q) \dot{e}_\vartheta = -(Vh(q, \dot{q}) + C_v) e_\vartheta + \mu_0 - \beta_3 Mh(q) e_\vartheta^{q_3/p_3} \quad (4.24)$$

avec  $\mu_0 = \tilde{f}(z) - \tau_d$  et  $\tilde{f}(z) = f(z) - \hat{f}(z)$ .

La sortie du réseau RBF est donnée comme:

$$\hat{f}(z) = \hat{W}^T \varphi(z) \quad (4.25)$$

On choisit  $\tilde{W} = W^* - \hat{W}$ ,  $\|\tilde{W}\| \leq W_{\max}$ , alors:

$$\mu_0 = \tilde{f}(z) - \tau_d = \tilde{W}^T \varphi(z) + \varepsilon - \tau_d \quad (4.26)$$

La loi de contrôle désignée dans l'équation (4.19) devient :

$$\tau = Bh^{-1}(C_v e_\vartheta + \hat{f}(z) + \beta_3 Mh(q) e_\vartheta^{q_3/p_3}) - \xi \quad (4.27)$$

où  $\xi$  est l'élément robuste introduit pour éliminer l'erreur d'approximation du réseau  $\varepsilon$  et les perturbations  $\tau_d$ .

On remplace l'équation (4.26) dans (4.24) :

$$Mh(q) \dot{e}_\vartheta = -(Vh(q, \dot{q}) + C_v) e_\vartheta - \beta_3 Mh(q) e_\vartheta^{q_3/p_3} + \hat{W}^T \varphi(z) + \varepsilon - \tau_d + \xi \quad (4.28)$$

On pose :

$$\mu_1 = \hat{W}^T \varphi(z) + \varepsilon - \tau_d + \xi \quad (4.29)$$

Introduisant l'équation (4.29) dans l'équation (4.28), on trouve :

$$Mh(q) \dot{e}_\vartheta = -(Vh(q, \dot{q}) + C_v) e_\vartheta - \beta_3 Mh(q) e_\vartheta^{q_3/p_3} + \mu_1 \quad (4.30)$$

L'élément robuste est désigné comme :

$$\xi = -(\varepsilon_N + b_d) \text{sign}(e_\vartheta) \quad (4.31)$$



où:  $\|\varepsilon\| \leq \varepsilon_N, \|\tau_d\| \leq b_d$

La fonction candidate de Lyapunov est choisie comme suit :

$$\rho = \frac{1}{2} e_{\vartheta}^T Mh(q) e_{\vartheta} + \frac{1}{2} \text{tr}(\tilde{W}^T F_w^{-1} \tilde{W}) \quad (4.32)$$

La dérivée de la fonction de Lyapunov est définie comme :

$$\dot{\rho} = e_{\vartheta}^T Mh(q) \dot{e}_{\vartheta} + \frac{1}{2} e_{\vartheta}^T \dot{M}h(q) e_{\vartheta} + \text{tr}(\tilde{W}^T F_w^{-1} \dot{\tilde{W}}) \quad (4.33)$$

A partir de l'équation (4.28) on trouve :

$$\dot{\rho} = -e_{\vartheta}^T Mh(q) (\alpha+1) e_{\vartheta} - e_{\vartheta}^T \beta Mh(q) e_{\vartheta}^{q_3/p_3} + \text{tr} \tilde{W}^T (F_w^{-1} \tilde{W} + \varphi(z) e_{\vartheta}^T) + e_{\vartheta}^T (\varepsilon - \tau_d + \xi) \quad (4.34)$$

On choisit :  $\dot{\tilde{W}} = -F_w \varphi(z) e_{\vartheta}^T$

La loi d'adaptation du réseau est donnée par :

$$\dot{\hat{W}} = F_w \varphi(z) e_{\vartheta}^T \quad (4.35)$$

Alors :

$$\dot{\rho} = -e_{\vartheta}^T Mh(q) (\alpha+1) e_{\vartheta} - e_{\vartheta}^T \beta Mh(q) e_{\vartheta}^{q_3/p_3} + e_{\vartheta}^T (\varepsilon - \tau_d + \xi) \quad (4.36)$$

Considérant le terme :

$$e_{\vartheta}^T (\varepsilon - \tau_d + \xi) = e_{\vartheta}^T (\varepsilon - \tau_d) + e_{\vartheta}^T \xi = e_{\vartheta}^T (\varepsilon - \tau_d) - \|e_{\vartheta}\| (\varepsilon_N + b_d) \leq 0$$

Sachant que la matrice  $Mh$  est définie positive,  $\beta_3$  et  $\alpha_3$  sont des positives, donc  $\dot{\rho} \leq 0$ .

### IV.3.3. Résultats de simulation

Dans cette section, les simulations sont subdivisées en deux parties. Dans la première partie, le contrôleur dynamique GFSM est proposé en l'absence et en la présence des perturbations. Dans la seconde partie, le NN-GFSM est introduit pour traiter les perturbations. La simulation utilisant matlab / simulink est appliquée sur le robot mobile.

Les valeurs des matrices du modèle dynamique proviennent de [95].

La trajectoire circulaire est considérée avec :  $v_r = 1$  m/s,  $\omega_r = 1$  rad/s. Les perturbations sont données avec :  $\tau_d = [0.1 \sin(t) \ 0.1 \cos(t)]$ . La trajectoire circulaire est considérée avec les valeurs suivantes :  $\alpha_1=4, \beta_1=1, p_1=7, q_1=5, \alpha_2=0.5, \beta_2=2, p_2=7, q_2=5, \alpha_3=1, \beta_3=2, p_3=7, q_3=5, \varepsilon_N=0.2, b_d=0.1$ .

**a. Contrôleur dynamique GFSM :**

En premier temps, on ne considère pas les perturbations.

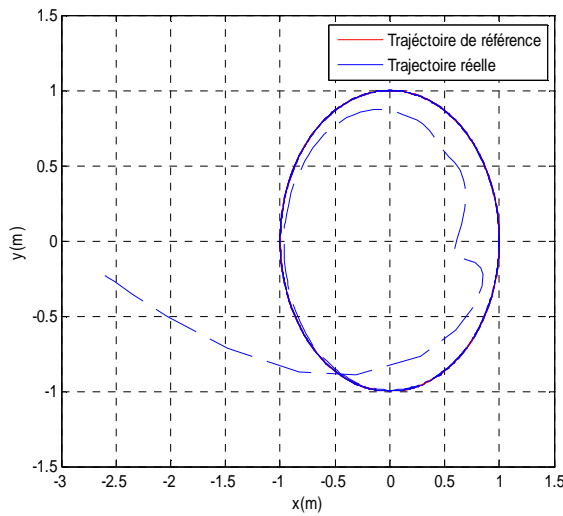


Figure 4.7: Suivi de trajectoire circulaire

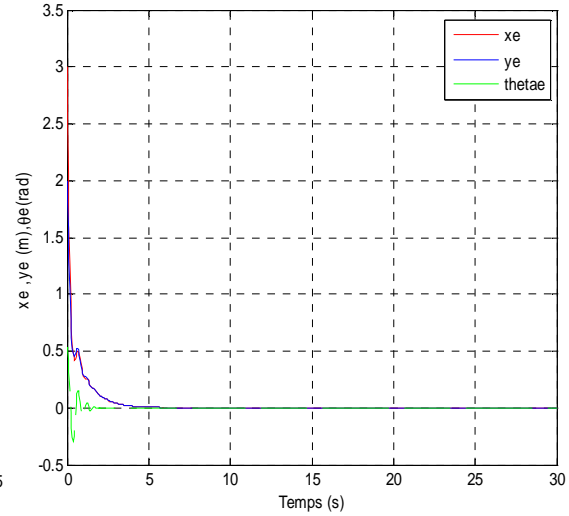


Figure 4.8: Erreurs  $x_e$ ,  $y_e$ ,  $\theta_e$

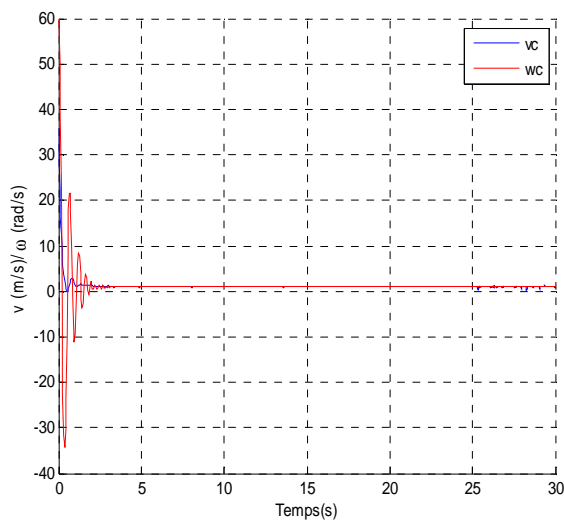


Figure 4.9: Vitesses dynamiques  $v$  et  $\omega$

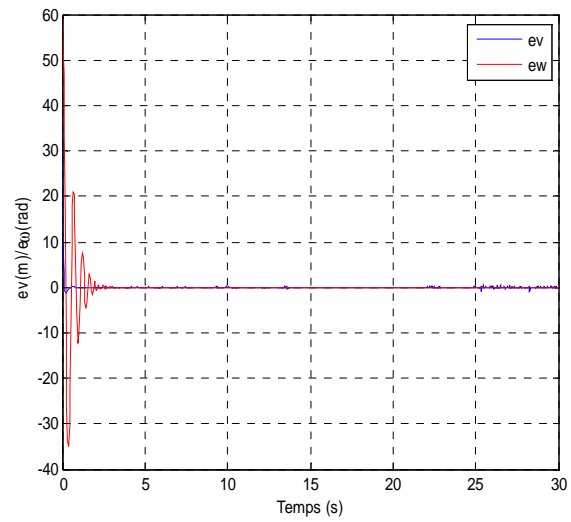


Figure 4.10: Erreurs de vitesses

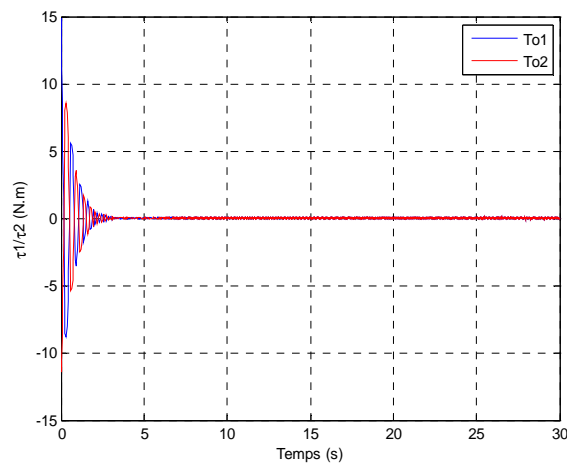


Figure 4.11: Couples générés

La figure (4.7) montre que le robot mobile suit la trajectoire de référence après un temps assez important à cause des erreurs initiales. Dans la figure (4.8), l'erreur  $\theta_e$  converge vers zéro en un temps fini, mais les erreurs  $y_e$  et  $x_e$  convergent vers zéro dans un temps supérieur à 4s. D'après cette figure on remarque que  $x_e$  converge vers  $y_e$  dans un temps limité qui est le temps de convergence de la deuxième surface vers zéro qui traduit l'efficacité de la commande proposée pour atteindre le suivi en temps fini.

La figure (4.9) présente les vitesses (linéaire et angulaire) du contrôleur dynamique  $v$  et  $w$  qui convergent vers les vitesses de référence qui valent 1. La figure (4.10) présente les erreurs des vitesses linéaire et angulaire qui représentent les différences entre les vitesses fournies par le contrôleur cinématique et celles fournies par le contrôleur dynamique, et montre la convergence des erreurs vers la valeur zéro. La figure (4.11) présente les couples appliqués sur les actionneurs du robot.

En deuxième temps, on va injecter les perturbations.

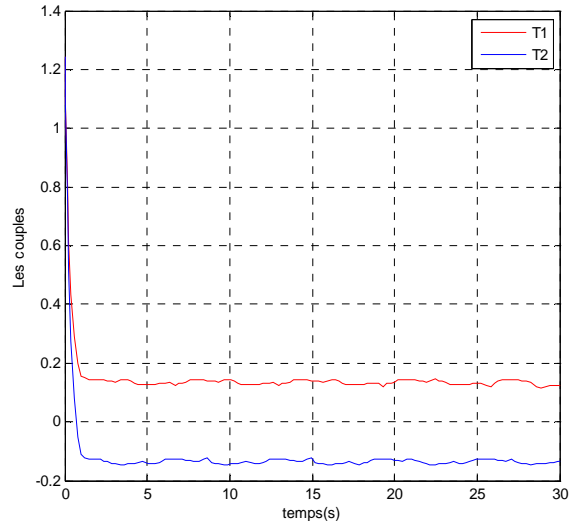
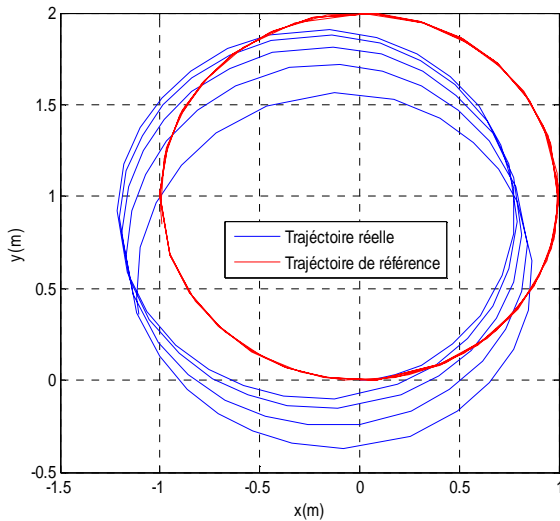


Figure 4.12: Suivi de trajectoires avec perturbations

Figure 4.13: Couples  $\tau_1$  et  $\tau_2$

La figure (4.12) présente le suivi d'une trajectoire circulaire en présence des perturbations et montre que le robot mobile diverge de la trajectoire de référence. Par conséquent, la commande GFSM stabilise le système en temps fini en absence de perturbations, mais ne peut pas le stabiliser en présence de perturbations. La figure (4.13) montre les couples générés par les actionneurs.

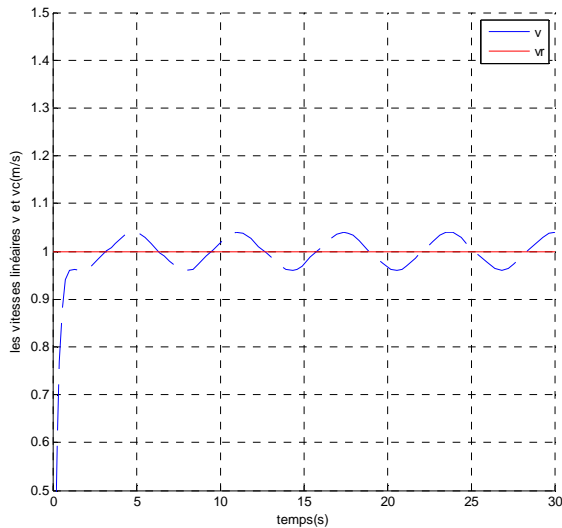


Figure 4.14: Vitesses linéaires  $v_r$  et  $v$

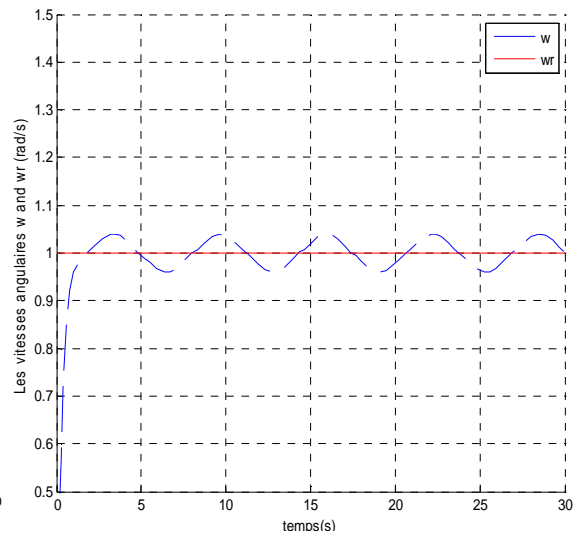


Figure 4.15: Vitesses angulaires  $\omega$  et  $\omega_r$

La figure (4.14) et la figure (4.15) montrent que les vitesses linéaires et angulaires réelles de la commande proposée ne pouvaient pas suivre les vitesses souhaitées en présence de perturbations.

### b. Contrôleur dynamique NN- GFSM

La trajectoire circulaire est considérée avec les mêmes valeurs des paramètres. Le réseau de neurones RBF est choisi avec sept couches cachées, la matrice de poids initiale est sélectionnée avec 0.1 et  $b=10$ .

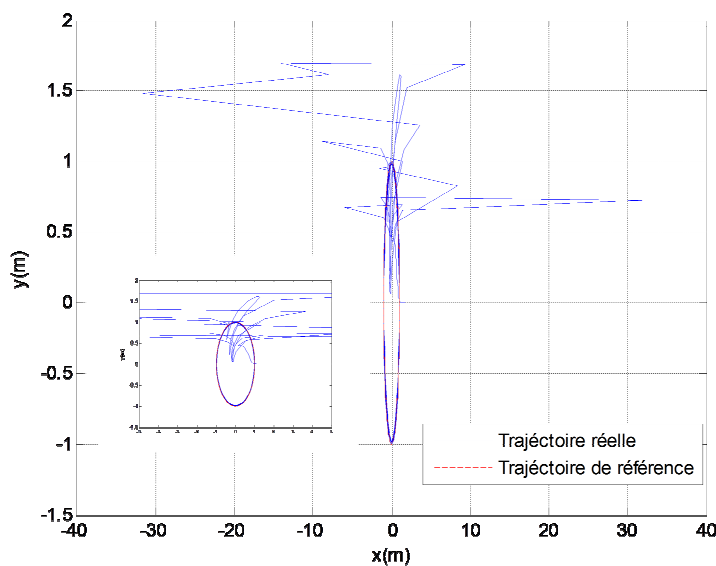


Figure 4.16: Suivi d'une trajectoire circulaire

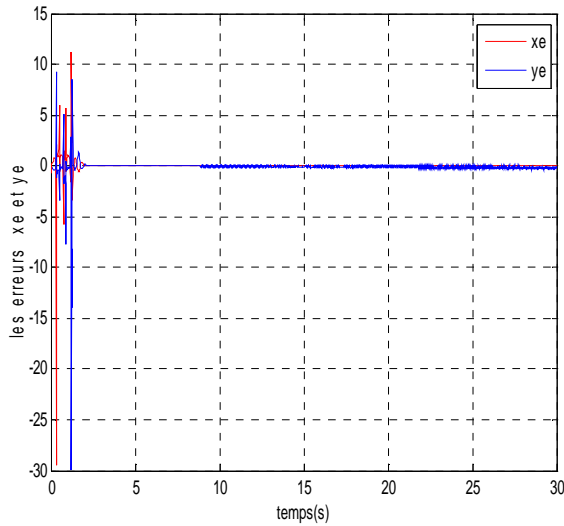


Figure 4.17: Erreurs de position  $x_e$  et  $y_e$

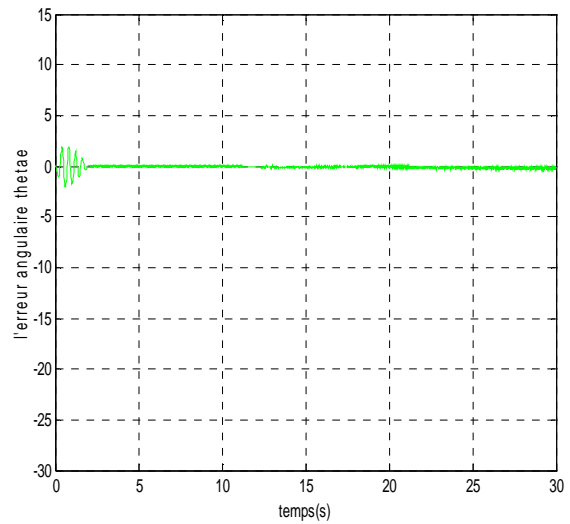


Figure 4.18: Erreur angulaire  $\theta_e$

La figure (4.16) montre le suivi d'une trajectoire circulaire avec le contrôleur NN-FGSM où le robot diverge de la trajectoire de référence dans un premier temps, mais il finit par suivre la référence. En comparant les résultats obtenus lors du suivi avec l'approche NN-FGSM aux résultats précédents, il est clair que les performances du système ont été améliorées par rapport aux cas précédents.

Les figures (4.17) et (4.18) présentent les erreurs de position et l'erreur angulaire respectivement et montrent que ces erreurs convergent vers zéro en un temps fini qui est inférieur à 2 secondes.

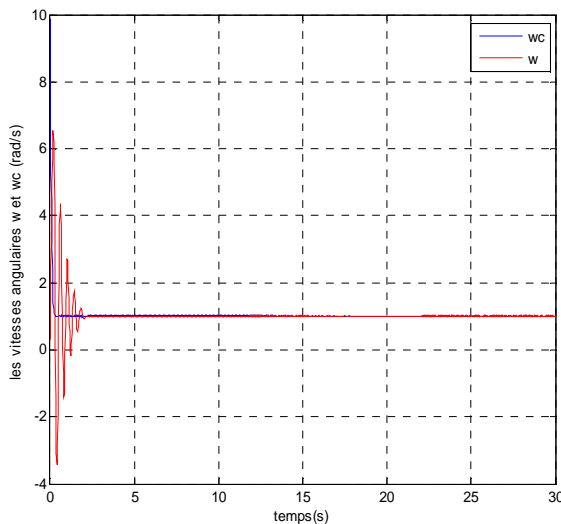


Figure 4.19: Vitesses angulaires  $\omega$  et  $\omega_c$

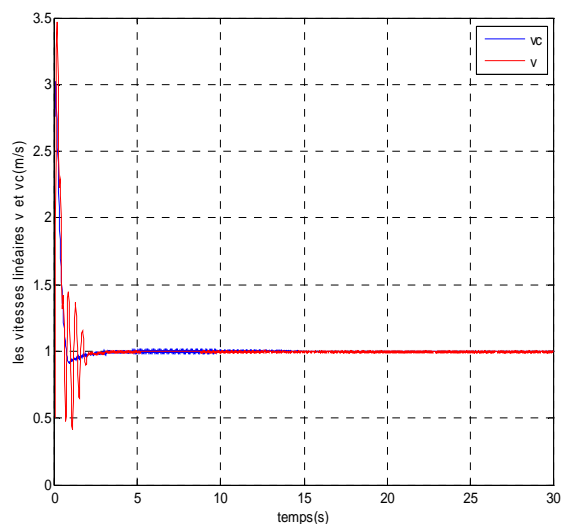


Figure 4.20: Vitesses linéaires  $v$  et  $v_c$

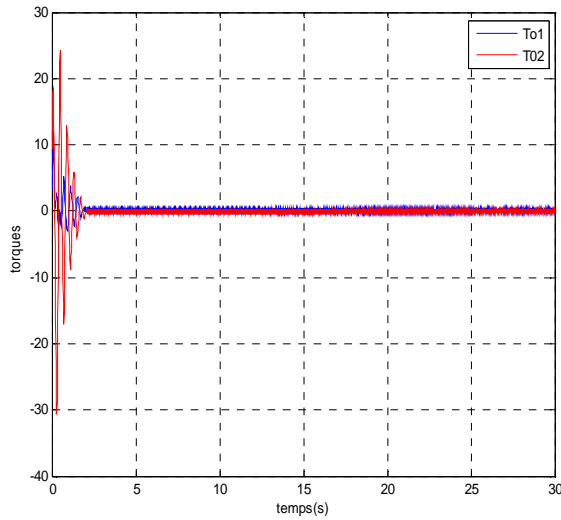


Figure 4.21: Couples générés

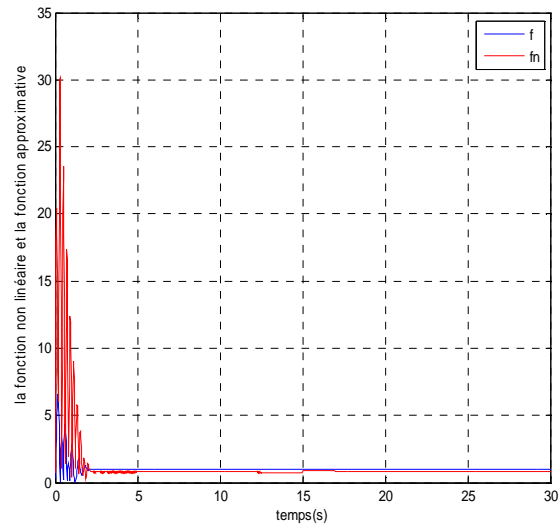


Figure 4.22: Fonctions  $f$  et  $\hat{f}$

La figure (4.19) montre que la vitesse linéaire  $v$  qui est la sortie du contrôleur dynamique converge vers la vitesse linéaire  $v_c$  qui est la sortie du contrôleur cinématique, alors l'erreur angulaire  $e_v$  converge vers zéro. La figure (4.20) présente les vitesses angulaires où  $w$  converge vers  $w_c$ , alors  $e_w$  converge vers zéro.

La figure (4.21) présente les couples appliqués aux actionneurs du robot mobile, la figure (4.22) présente la fonction non linéaire  $f$  et sa fonction estimée  $\hat{f}$ .

#### IV.4 Conclusion

Nous avons proposé dans ce chapitre une commande dynamique pour la poursuite de trajectoires d'un robot mobile qui est basée sur le mode glissant et les réseaux de neurones. Nous avons proposé en premier temps une commande dynamique basée sur l'approche GFSM qui donne une bonne convergence vers la trajectoire de référence en un temps fini, mais cette approche a montré une robustesse limitée en présence des perturbations. En deuxième temps on a proposé une approche NN-GFSM qui associe deux contrôleurs, un contrôleur basé sur l'approche GFSM pour stabiliser les erreurs des vitesses et un contrôleur neuronal pour estimer la fonction non linéaire affectée par les perturbations. Cette dernière approche a montré une amélioration des performances du système et une bonne robustesse par rapport à l'approche GFSM. Les travaux de simulation ont montré l'efficacité et la robustesse de la commande proposée.

---

## **CONCLUSION GENERALE**

---

## CONCLUSION GENERALE

Dans cette thèse, le travail effectué consiste à élaborer les différents algorithmes de commande pour la navigation d'un robot mobile, soit une navigation avec évitement d'obstacles et une navigation avec poursuite de trajectoire. Pour l'évitement d'obstacles les algorithmes proposés ont l'objectif de forcer le robot mobile à atteindre la configuration d'arrivée à partir d'une configuration de départ sans collision avec les obstacles et sans intervention humaine. Les différents types de contrôleurs basés sur la commande par mode glissant sont proposés pour assurer le suivi de la trajectoire du robot mobile et fait converger le système vers la référence dans un temps fini, en tenant compte de la dynamique du robot.

Les algorithmes d'évitement d'obstacles présentés dans ce travail sont basés sur deux approches, celle des champs de potentiels artificiels qui est une simple technique basée sur les forces engendrées par les obstacles et le but à atteindre. La seconde approche repose sur l'utilisation d'un contrôleur flou intelligent pour mettre en œuvre le calcul approximatif de raisonnement nécessaire pour la manipulation du problème d'évitement d'obstacles.

Les différentes approches de poursuite de trajectoire élaborées dans ce présent travail sont basées sur la commande par mode glissant rapide et globale (GFSM). On a proposé deux stratégies de commande : une commande cinématique et une commande dynamique en présence et en absence de perturbations. La commande dynamique élaborée associe un contrôleur neuronal dont le but d'approximer la fonction non linéaire et un contrôleur GFSM. Cette commande a démontré une bonne robustesse et de meilleures performances.

Ce présent travail de thèse s'est articulé autour de quatre chapitres où on a présenté en premier lieu les différents types des robots mobiles à roues ainsi que leurs modèles cinématiques et dynamiques. La modélisation dynamique effectuée repose sur deux approches, Newton-Euler et Lagrange-Euler.



Nous avons proposé dans le deuxième chapitre deux algorithmes de navigation avec évitement d'obstacles d'un robot mobile, basés sur la technique des champs de potentiels artificiels et la logique floue. Les travaux de simulation entrepris permettent d'évaluer ces deux approches tout en montrant leurs avantages et inconvénients. La planification de trajectoires avec évitement d'obstacles d'un robot guidé par les champs de potentiels artificiels est une technique très simple qui a un principe de fonctionnement intuitif basé sur les champs d'énergie. Les contrôleurs flous quant à eux ont démontré leur efficacité et leur simplicité d'implémentation.

Nous avons par la suite proposé une commande par mode glissant notamment en temps fini pour la poursuite de trajectoire d'un robot mobile. Nous avons proposé en premier temps une commande cinématique qui est basée sur une commande terminale rapide où la contrainte du temps limité est prouvée par les différentes simulations. En deuxième temps on a proposé une commande dynamique basée sur deux contrôleurs : un contrôleur cinématique qui s'articule autour de la commande par mode glissant terminal rapide et un contrôleur dynamique PID en absence de perturbations. Les travaux de simulation ont montré l'efficacité des commandes proposées.

Pour une commande dynamique, on ne peut pas parler sur l'efficacité et la robustesse de cette dernière sans faire intervenir les perturbations. Pour cela, en dernier lieu, nous avons proposé une commande dynamique robuste pour la poursuite de trajectoires d'un robot mobile qui est basée sur le mode glissant et les réseaux de neurones. Cette approche a montré une amélioration des performances du système et une bonne robustesse par rapport à l'approche GFSM.

Le travail effectué dans cette thèse s'est étendu sur plusieurs axes de recherche, ce qui justifie que certains aspects n'ont pas été traités d'une façon très profonde. Alors, nous envisageons, dans nos futurs travaux de nous focaliser sur les perspectives suivantes:

- Amélioration des résultats obtenus, par optimisation du temps de convergence en utilisant les techniques d'optimisation (Algorithmes génétiques, PSO,...)

- Développer l'approche de commande en faisant intervenir les observateurs mode glissant pour estimer la surface de friction.
- Concevoir une nouvelle approche de commande dynamique basée sur des approximateurs flous associés à une commande GFSM pour le suivi de trajectoires afin de vérifier la robustesse et l'efficacité.
- Implémentation pratique des algorithmes proposés.

---

## **BIBLIOGRAPHIE**

---

## REFERENCES BIBLIOGRAPHIQUES

- [1] Spyros G. Tzafestas, "Introduction to mobile robot control," Elsevier edition, 2014.
- [2] S. M. LaValle., "Planning algorithms", Cambridge University Press, Cambridge, U.K., 2006.
- [3] H. Choset, K. M. Lynch, S. Hutchinson, G. Kantor, W. Burgard, L. E. Kavraki & S. Thrun, "Principles of robot motion : Theory, algorithms, and implementations.", MIT Press, Cambridge, MA, 2005.
- [4] J. Borenstein, B. Everett & L. Feng, " Navigating mobile robots: Systems and Techniques," A. K. Peters, Ltd., Wellesley, 1996.
- [5] S. Thrun, "Robotic Mapping," A Survey. G. Lakemeyer & B. Nebel, editeurs, Exploring Artificial Intelligence in the New Millenium. Morgan Kaufmann, 2002.
- [6] Joseph L. Jones, Anita M. Flynn, and Bruce A. Seiger, "Mobile Robots: Inspiration to implementation," A K Peters, Natick, Massachusetts, 2nd edition, 1999.
- [7] J. Borenstein and Y. Koren, "Real time obstacle avoidance for fast mobile robots". IEEE Transactions on Systems, Man, and Cybernetics, vol.19, n°5, pp.1179–1187, 1989.
- [8] Salim Belkhou, Adel Azzouz, Maarouf Saad, Chahe Nerguizian, and Vahe Nerguizian. "A novel approach for mobile robot navigation with dynamic obstacles avoidance". Journal of Intelligent Robotic Systems, vol.44, n°3, pp.187–201, 2005.
- [9] R. Kuc and B. Barshan , " Navigating vehicles through an unstructured environment with sonar," In Proc. of the 1989 IEEE International Conference on Robotics and Automation, vol. 3, pp. 1422–1426, Scottsdale, AZ, 1989.
- [10] Oussama Khatib. , "Real time obstacle avoidance for manipulators and mobile robots," The International Journal of Robotics Research, vol. 5, n°1, pp.90–98, 1986.
- [11] Yasushi Yagi, Hiroyuki Nagai, Kazumasa Yamazawa, and Masahiko Yachida, "Reactive visual navigation based on omnidirectional sensing - path following and collision avoidance". Journal of Intelligent and Robotic Systems, vol.31, n°4, pp. 379–395, 2001.
- [12] Jinseok Lee, Yunyoung Nam, Sangjin Hong, Weduke Cho, " New potential functions with random force algorithms using potential field method," Journal of Intelligent & Robotic Systems, vol.66, n°3, pp. 302-319, 2012.
- [13] Javier Minguez and Luis Montano, "Sensor based robot motion generation in unknown, dynamic and troublesome scenarios," Robotics and Autonomous Systems, vol. 52, n°4, pp. 290–311, 2004.
- [14] Zhihua Qu, Jing Wang, and Clinton E. Plaisted, "A new analytical solution to mobile robot trajectory generation in the presence of moving obstacles," IEEE Transactions on Robotics, vol.20, n°6, pp.978–993, 2004.
- [15] Fukao, T., Nakagawa, H. and Adachi, N., " Adaptive tracking control of a nonholonomic mobile robot," IEEE Transactions on Robotics and Automation, vol.16, n° 5, pp. 609–615, 2000.
- [16] Hwang, C. L., "A novel Takagi-Sugeno-based robust adaptive fuzzy sliding-mode controller," IEEE Transactions on Fuzzy Systems, 12 (5), pp. 676–687, 2004.

- [17] Mohareri, O., Dhaouadi, R., Shirazi, M.M., “Intelligent neural network based controllers for path tracking of wheeled mobile robots: A comparative analysis,” IEEE International Workshop on Robotic and Sensors Environments, pp. 1-6, 2010.
- [18] Kozłowski, K. and Majchrzak, J., “A backstepping approach to control a nonholonomic mobile robot”, In Proceedings of IEEE International Conference on Robotics and Automation, vol.4, 2002, pp. 3972-3977.
- [19] Nganga-Kouya, D. and Okou, F.A., “Adaptive backstepping control of a wheeled mobile robot”, 17th Mediterranean Conference on Control and Automation, 2009, pp. 85-91.
- [20] Song, K.T., Lee, C.H. and Teng, C.C., “Tracking control of unicycle-modeled mobile robots using a saturation feedback controller”, IEEE Transactions on Control Systems Technology, 2001, pp. 305-318.
- [21] Yang, J.M. and Kim, J.H., “Sliding mode control for trajectory tracking of nonholonomic wheeled mobile robots”, IEEE Transactions on Robotics and Automation, vol. 15, n°3, 1999, pp. 578–587.
- [22] V.I. Utkin, “Variable structure systems with sliding modes”, IEEE Trans. Autom. Control 22, 1977, pp. 212–222.
- [23] Y. Orlov, “Discontinuous Systems: Lyapunov Analysis and Robust Synthesis under Uncertainty Conditions”, Springer-Verlag, London, UK, 2010.
- [24] F. Dinuzzo, A. Ferrara, “Higher order sliding mode controllers with optimal reaching,” IEEE Trans. Autom. Control, vol 54 , n°9, 2009, pp. 2126–2136.
- [25] A. Polyakov, “ Nonlinear feedback design for fixed-time stabilization of linear control systems”, IEEE Trans. Autom. Control, vol. 57, n° 8, 2012, pp. 2106–2110.
- [26] Y. Kanayanma “A stable tracking control method for an autonomous mobile robot,” Proc. of IEEE Conference on Robotics and Automation, vol. 1, pp. 384-389, 1990.
- [27] A. Ferrara, and M. Rubagotti, “Second-order sliding-mode control of a mobile robot based on a harmonic potential field,” Control Theory and Applications, IET., vol.2, n°9, pp.807–818, 2008.
- [28] M.S.Park, D.Chwa, and S.K.Hong, “Decoupling Control of A Class of Underactuated Mechanical Systems Based on Sliding Mode Control,” International Joint Conference SICE-ICASE, pp. 806–810, 2006.
- [29] S. Riachy, Y. Orlov, T. Floquet, R. Santiesteban, and J.-P.Richard, “Second-order sliding mode control of underactuated mechanical systems 1: Local stabilization with application to an inverted pendulum,” Int. J. Robust Nonlinear Control, vol. 18, pp. 529–543, Mar. 2008.
- [30] J. Velagic, N. Osmic and B. Lacevic, “ Neural Network controller for mobile robot motion control,” in International journal of intelligent systems and technologies, vol. 21, pp. 470-479, Oct. 2008.
- [31] Angelo A. Robotics: a reference guide to new technology. Boston, MA: Greenwood Press; 2007.
- [32] Muir PF, Neuman CP. “Kinematic modeling of wheeled mobile robots,” Journal of Robotics Systems 1987, vol.4, n°2, pp. 281-329.
- [33] Alexander JC, Maddocks JH. “On the kinematics of wheeled mobile robots,”. International Journal of Robotics Res 1981, vol. 8, n°5, pp.15-27.
- [34] Kim DS, Hyun Kwon W, Park HS.” Geometric kinematics and applications of a mobile robot,”. International Journal Control Automation Systems, 2003, vol.1, n°3, pp. 376-84.
- [35] Rajagopalan R.” A generic kinematic formulation for wheeled mobile robots,”. Journal of Robotics Systems, 1997, vol. 14, pp.77-91.

- [36] Sidek N, Sarkar N. "Dynamic modeling and control of nonholonomic mobile robot with lateral slip,". In: Proceedings of seventh WSEAS international conference on signal processing robotics and automation (ISPRA'08), Cambridge, UK; February 20-22, 2008, pp. 66-74.
- [37] Sidek SN, " Dynamic modeling and control of nonholonomic wheeled mobile robot subjected to wheel slip". PhD Thesis, Vanderbilt University, Nashville, TN, December 2008.
- [38] Ivanjko E, Petrinic T, Petrovic I, " Modeling of mobile robot dynamics,". In: Proceedings of seventh EUROSIM congress on modeling and simulation. Prague, Czech Republic; September 6-9, 2010.pp. 479-86.
- [39] Handy A, Badreddin E, "Dynamic modeling of a wheeled mobile robot for identification, ,navigation and control, " In: Proceedings of IMACS conference on modeling and control of technological systems. Lille, France; 1992. pp. 119-28.
- [40] Kuc, R. and Barshan, B, "Navigating vehicles through an unstructured environment with sonar", Proc. of the 1989 IEEE International Conference on Robotics and Automation, Vol. 3, Scottsdale, AZ, pp. 1422-1426.
- [41] 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.
- [42] G. Blanc, Y. Mezouar, and P. Martinet, " Indoor navigation of a wheeled mobile robot along visual routes, " In Proceedings of the IEEE International Conference on Robotics and Automation, 2005.
- [43] D. Boley, E. Steinmetz, and K. Sutherland, "Robot localization from landmarks using recursive total least squares,"In Proceedings of the IEEE International Conference on Robotics and Automation (ICRA-96), volume 4, pp. 1381–1386, 1996.
- [44] Lamiroux, F., Bonnafous, D. and Lefebvre, O, "Reactive path deformation for nonholonomic mobile robots, " IEEE Transactions on Robotics, vol. 20, n°6, 967–977, 2004.
- [45] Belkhouche, F. and Belkhouche, B, "A method for robot navigation toward a moving goal with unknown maneuvers, " Robotica, vol. 23, n°6, pp.709–720, 2005.
- [46] Yang, S. X. and Meng, M, " Neural network approaches to dynamic collision-free trajectory generation," IEEE Transactions on Systems, Man and Cybernetics -Part B: Cybernetics, vol. 31, n°3, pp. 302–318, 2001.
- [47] Y. Cang, N. H. C. Yung, D. Wang, "A Fuzzy Controller with Supervised Learning Assisted Reinforcement Learning Algorithm for Obstacle Avoidance," IEEE Transaction on Systems, Man, and Cybernetics-Part B: Cybernetics, vol. 33, n°1, pp.1-11, 2003.
- [48] M. Wang, N. K. Liu James, "Fuzzy Logic based Real-time Robot Navigation in Unknown Environment with Dead Ends," Robotics and Autonomous Systems, vol. 56, pp.625-643, 2008.
- [49] Latombe J, " Robot Motion Planning". Kluwer Academic Publishers, Boston, 1991.
- [50] Lee, Leng-Feng, "Decentralized Motion Planning Within an Artificial Potential Framework (APF) for Cooperative Payload Transport by Multi-Robot Collectives ".Mémoire de Maîtrise, Buffalo, New York, State University of New York at Buffalo, 2004.
- [51] L. Cherroun, "Navigation autonome d'un robot mobile par des techniques neuro-floues," Thèse de Doctorat en Automatique, Université de Biskra, Algérie, 2014.
- [52] S.H. Ashikin Mohammad, M. AkmalJeffril, "Mobile robot obstacle avoidance by using Fuzzy Logic technique," IEEE 3rd International Conference on System Engineering and Technology (ICSET), Malaysia ,2013.

- [53] J. Laneurit, "Perception multi sensorielle pour la localisation d'un robot mobile en environnement extérieur, application aux véhicule routiers,"Thèse de Doctorat, Université Blaise Pascal, France, 2012.
- [54] Bartoszewicz, A., Kaynak, O., Utkin, V.I," (eds.) special section on sliding mode control in industrial applications, ". IEEE Transactions on Industrial Electronics, vol. 55, n°11, 2008, pp. 3806- 4074.
- [55] Bartoszewicz, A., Patton, R.J. "(eds.) Sliding mode control", Special issue. Int. Journal of Adaptive Control and Signal Processing 21, 635–822, 2007.
- [56] Emel'yanov, S.V," Method of designing complex control algorithm using an error and its first time derivative only," Automation and Remote Control 18(10), 1957.
- [57] Emel'yanov, S.V., Burovoi, I.A., et al, " Mathematical models of process in technology and development of variable structure control system, " Metallurgy (Moscow) 18(07) 1964.
- [58] Itkis, U."Control Systems of Variable Structure", Wiley, New York ,1976.
- [59] Utkin, V.I., Shi, J,"Integral sliding mode in systems operating under uncertainty conditions, " In: Proceedings of the 35th conference on decision and control, pp. 4591–4596, 1996.
- [60] Castanos, F., Fridman, L," Analysis and design of integral sliding manifolds for systems with unmatched perturbations, " IEEE Trans. on Automatic Control, vol. 51, n°5, pp. 853–858 , 2006.
- [61] Bartoszewicz, A, "Time-varying sliding modes for second-order systems, " IEE Proc.-Control Theory Appl, vol.143, n°5, pp. 455–462 ,1996.
- [62] Choi, S.-B., Park, D.-W., Jayasuriya, S," A time-varying sliding surface for fast and robust tracking control of second order uncertain systems, " Automatica, vol.30(5), pp. 899–904 ,1994.
- [63] Xia, Y., Liu, G.P., Shi, P., Chen, J., Rees, D., Liang, J," Sliding mode control of uncertain linear discrete-time systems with input delay, " IET Control Theory Appl. Vol.1(4), pp. 1169–1175, 2007.
- [64] Yan, M., Shi, Y," Robust discrete-time sliding mode control for uncertain systems with time-varying state delay," IET Control Theory Appl. Vol. 2(8), pp. 662–674 , 2008.
- [65] R.B. Fernandez, J.K. Hedrick, "Control of multivariable nonlinear systems by sliding mode control, " International Journal of Control, volume 46, n°3, pp. 1019- 1040, 1987.
- [66] J.J.E. Slotine and W. Li, "Applied Nonlinear control", Englewood cliffs, New jersey,Prentice Hill, 1991.
- [67] J.Liu. X Wang, "Advanced Sliding Mode Control for Mechanical Systems, " Tsinghua University Press, Beijing and Springer-Verlag Berlin Heidelberg, 2012.
- [68] R.A. Decarlo, S.H. Zak and G.P. Matthews, "Variable Structure Control of nonlinear multivariable systems: A tutorial, " IEEE, volume 76, n°3, pp. 212-232, 1988.
- [69] S.H. Ashikin Mohammad, M. AkmalJeffril,"Mobile robot obstacle avoidance by using Fuzzy Logic technique, " IEEE 3rd International Conference on System Engineering and Technology (ICSET), Malaysia ,2013.
- [70] Feng Y, Yu XH, Man ZH, "Non-singular terminal sliding mode control of rigid manipulators", Automatica, 2002, vol. 38, pp. 2159 -2167
- [71] Park KB, Tsuiji T, "Terminal sliding mode control of second-order nonlinear uncertain Systems", International Journal of Robust and Nonlinear Control, 1999, vol. 9, n°11, pp. 769 -780.
- [72] Yangming Zhang a , Guorong Liu b , Biao Luo," Finite-time cascaded tracking control approach for mobile robots, " Information sciences, vol 284 , pp 31–43, 2014.

- [73] Y. Kanayama, Y. Kimura, F. Miyazaki, "A stable tracking control method for a non-holonomic mobile robot, " Proceedings IROS '91. IEEE/RSJ International Workshop, Japan, 1991.
- [74] A.Mallem, N.Slimane, W.Benaziza, "Mobile robot trajectory tracking using PID fast terminal sliding mod inverse dynamic control, " 4th international conference on control Engineering and information technology (CEIT-2016), Hammamet, Tunisia.
- [75] Chwa. D, " Sliding-mode tracking control of nonholonomic wheeled mobile robots in polar coordinates, " IEEE Trans. Contr. Syst. Technol. Vol.12 n°4, 2004, pp. 637–644.
- [76] Filipescu, A., Minzu, V., Dumitrascu, B., Filipescu, A. and Minca, E., "Trajectory-tracking and discrete-time sliding-mode control of wheeled mobile robots," IEEE International Conference on Information and Automation, 2011, pp. 27-32.
- [77] Park. B.S, Yoo. S.J, Park. J.B., Y.H, Choi, "Adaptive neural sliding mode control of nonholonomic wheeled mobile robots with model uncertainty, " IEEE Trans. Contr. Syst. Technol, vol. 17 n°1, 2009, pp. 207–214.
- [78] Tian, Y.P. Cao. K.C, "Time-varying linear controllers for exponential tracking of non-holonomic systems in chained form," Int. J. Robust Nonlin. Contr, vol. 17 n°7, 2007, pp. 631–647.
- [79] Tian. Y.P, Li. S.H, "Exponential stabilization of nonholonomic dynamic systems by smooth time-varying control," Automatica, vol. 38 n°7, 2002, pp. 1139–1146.
- [80] Fu. J, Chai. T.Y, Su. C.Y, Jin. Y, "Motion/force tracking control of nonholonomic mechanical systems via combining cascaded design and backstepping," Automatica, vol. 49 n°12, 2013, pp. 3682–3686.
- [81] Huang. J.S, Wen. C.Y, Wang. W, Jiang. Z.P, "Adaptive stabilization and tracking control of a nonholonomic mobile robot with input saturation and disturbance, " Syst. Contr. Lett. Vol.62 n°3, 2013, pp. 234–241
- [82] Antonelli, G., Chiaverni, S., Fusco, G., "A fuzzy logic based approach for mobile robot path tracking", IEEE Trans. Fuzzy Syst. Vol.15 n°2, 2007, pp. 211–221.
- [83] R. Fierro and F. L. Lewis, "Control of a Nonholonomic Mobile Robot Using Neural Networks," in IEEE Transactions on Neural Networks, vol.9, 1998, pp. 589-600.
- [84] Jin Oh Jang, "Adaptive neuro-fuzzy network control for a mobile robot", Journal of Intelligent and Robotic Systems, vol.62 ,2011, pp. 567-586.
- [85] A. Lacaze, "A neural network planner that prunes its own tree," Term paper, Knowledge Engineering Department, University of Meryland, 1997.
- [86] A. G. Barto, "Neuronlike adaptive elements that can solve difficult learning control problems," IEEE Transactions on System, Man and Cybernetics, vol. 13, pp. 834-846, 1983.
- [87] Z. Hendzel, "Adaptive Critic Neural Networks for Motion Control of Wheeled Mobile Robot," Nonlinear Dynamics, vol. 50, n° 4, pp. 849- 855, 2007.
- [88] K.S. Narendra, and K. Pathasarathy, "Identification and control of dynamic systems using neural network," IEEE Transaction on Neural Networks, vol. 1, n° 1, pp. 4-27, 1990.
- [89] D. Nguyen, and B. Widrow, "Neural Networks for Self- Learning Control Systems," IEEE Control System Magazine, vol. 10, n° 1, pp. 18-23, Feb. 1990.
- [90] K. Hornik, M. Stinchombe, and H. White, "Universal Approximation of an Unknown mapping and its Derivatives Using Multilayer Feedforward Networks," Neural Networks, vol. 3, 1990.
- [91] M. Corradini, G. Ippoliti, and S. Longhi, "Neural Networks Based Control of Mobile Robots: Development and Experimental Validation", Journal of Robotic Systems, vol. 20, n°. 10, pp. 587-600, 2003.



- [92] F.Douak., "Reconstruction des images compressées en utilisant les réseaux de neurones artificiels et la DCT, " Thèse de magister, département d'électronique, université de Batna, 2008.
- [93] Park J, Sandberg IW," Universal approximation using radial-basis-function networks, "Neural computation, vol. 3 ,1991, pp. 246- 257.
- [94] A.Mallem, N.Slimane, W.Benaziza," Robust control of mobile robot in presence of disturbances using neural network and global fast sliding mode ", Journal of Intelligent And Fuzzy systems, vol. 34, 2018, pp. 4345-4354.
- [95] Duc Do, K., Zhong-Ping J., Pan, J," A global output-feedback controller for simultaneous tracking and stabilization of unicycle-type mobile robots, " IEEE Trans. Automation Control, vol. 30 , n°3, 2004, pp. 589–594.