

République Algérienne Démocratique et Populaire
Ministère de l'Enseignement Supérieur et de la Recherche Scientifique
Université de Batna -2-
Faculté de Technologie
Département d'Électronique



THÈSE

Présentée pour l'obtention du diplôme de
DOCTORAT en SCIENCES
Spécialité: **Électronique**
Option: **Robotique**

Par

ZIDANI Ghania

Ingénieur d'état (1995), Magister en électronique (2009) de l'université de Batna
Maître assistant A, université de Batna 2

Thème

Commande Robuste d'un Robot Mobile à Roues

Soutenue le 04/07/2017

Devant le jury:

CHAFAA Kheiredine	Prof.	Université Batna -2-	Président
DRID Said	Prof.	Université Batna -2-	Rapporteur
ARAR Djemai	Prof.	Université Batna -2-	Co-rapporteur
GOLEA Amar	Prof.	Université Biskra	Examineur
BETKA Achour	Prof.	Université Biskra	Examineur
BENAICHA Samira	M.C.A.	Université Sétif	Examineur

Commande robuste d'un robot mobile à roues

ZIDANI Ghania

Université Batna 2, Département d'électronique, Faculté de technologie, Batna, Algérie.

Laboratoire des Systèmes de Propulsion-Induction Electromagnétique LSP-IE de l'université de Batna 2.

E-mail: gh.zidani@gmail.com

LISTE DES TRAVAUX

Revue Internationale

- G. Zidani, S. Drid, L. C. Alaoui, D. Arar and P. Bussy, “Robust Nonlinear Control of a Mobile Robot,” *Journal of Electrical Engineering & Technology*, Vol. 11 N°4, pp. 1921-1928, 2016.

Conférences Internationales

- G. Zidani, S. Drid, L. C. Alaoui, A. Benmakhlouf and S. Chaouch, “Backstepping Controller for a Wheeled Mobile Robot,” 4th International Conference on Systems and Control (ICSC'15), Sousse, Tunisia, April 28-30, 2015.
- G. Zidani, S. Drid, L. C. Alaoui and D. Arar, “Nonlinear Tracking Control of a Wheeled Mobile Robot,” the 15th International conference on Sciences and Techniques of Automatic control & computer engineering, STA'2014, Hammamet, Tunisia, 2014.
- G. Zidani, A. Louchene, A. Benmakhlouf and D. Djarah, “Exécution de trajectoire pour robot mobile par réseaux de neurones,” The 2nd International Conference on Electronics and Oil : From Theory to Application ICEO'13, Ouargla, Algérie, 05-06, Mars, 2013.
- A. Benmakhlouf, D. Djarah and G. Zidani, “A new fuzzy path following controller for a mobile robot,” International Conference on Systems and Information Processing ICSIP'09, Guelma, Algérie, 2-4, Mai, 2009.

Ouvrages

- Ghania Zidani, Ahmed Louchene, “Exécution de trajectoire pour robot mobile par réseaux de neurones,” Editions Universitaires Européennes, ISBN-13: 978-3-639-60787-1, 06-02-2017.
- Ghania Zidani, Ahmed Slamati, Batoul Fatima Tom, “Etude et simulation des régulateurs P, PI et PID, Application aux systèmes -1er ordre et 2ième ordre-,” Editions Universitaires Européennes, ISBN-13: 978-3-8417-7831-4, 19-02-2016.

REMERCIEMENTS

Je tiens à remercier mon promoteur Monsieur Drid Said, professeur au département de génie électrique à l'université de Batna, pour m'avoir proposé et dirigé ce thème, pour ses remarques, ses conseils, son aide et son soutien, qu'il trouve ici l'expression de ma reconnaissance la plus sincère. Mes remerciements vont également à Monsieur Arar Djemai, professeur au département d'électronique à l'université de Batna, pour avoir accepté d'en être un des rapporteurs.

Je remercie très spécialement mes deux collègues Monsieur Djarah Djalal et Monsieur Benmakhlouf Abdeslam, enseignants au département de génie électrique à l'université de Ouargla, pour toute l'aide qu'ils m'ont apporté, je leur suis reconnaissante pour avoir répondu toujours présents aux moments difficiles. Qu'ils trouvent ici l'expression de ma profonde gratitude.

J'adresse mes sincères remerciements à Madame Zidani Fatiha, professeur au département de génie électrique à l'université de Batna, son suivi, ses compétences scientifiques et son aide précieuse ont beaucoup apporté au développement de ce travail. Quoique je dise, je ne pourrais certainement pas te remercier assez ma sœur.

J'adresse également mes remerciements à Madame Chaouch Souad, professeur à la faculté de technologie à l'université de Batna, pour son aide, son soutien inconditionnel, son amitié, et ses conseils qui ont été pour moi une aide si précieuse.

Je tiens à remercier vivement Monsieur Chafaa Kheiredine, professeur au département d'électronique à l'université de Batna, pour l'honneur qu'il me fait en acceptant de présider le jury de cette thèse.

Qu'ils soient tous remerciés d'avoir accepté et participer à ce jury de la présente thèse, à savoir :

- Monsieur Golea Amar, professeur à l'université de Biskra.
- Monsieur Betka Achour, professeur à l'université de Biskra.
- Mademoiselle Benaicha Samira, maître de conférences à l'université de Sétif.

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, mes sœurs et mes frères.

Mes remerciements s'adressent aussi à toutes mes amies.

Merci à tous ceux qui de près ou de loin m'ont aidé.

A mes chers parents

A mes sœurs et frères

A mes nièces et neveux

A tous ceux qui me sont chers

Commande robuste d'un robot mobile à roues

Résumé

Le travail présent dans ce mémoire est l'élaboration des lois de commande robuste pour résoudre le problème de poursuite des trajectoires des robots mobiles non holonomes. L'objectif est de garantir le comportement correct du système et d'assurer une bonne poursuite de trajectoire même en présence des variations paramétriques du système. La fonction de poursuite consiste à guider le robot à travers les points intermédiaires pour arriver à la destination finale, le robot doit rejoindre son but dans un temps prédéfini en essayant de minimiser l'erreur "en distance" et "en orientation". Le modèle utilisé pour démontrer les lois de commande dans ce travail est basé sur le modèle cinématique et dynamique du robot mobile de type unicycle. On s'est orienté vers différentes approches, la première approche est basée sur l'utilisation d'une commande adaptative non linéaire par backstepping, la deuxième approche concerne la technique par mode de glissement. Un contrôleur basé sur la théorie de Lyapunov associé au mode glissant est conçu pour assurer la fonction de poursuite de la trajectoire du robot mobile. Afin d'améliorer les performances statique et dynamique, la logique floue est introduite dans le but d'adapter les gains du contrôleur. Enfin une implémentation pratique est réalisée pour assurer la poursuite d'une trajectoire désirée, cette dernière est basée sur le suivi d'un objet dans une séquence d'images afin de localiser la position et la vitesse du robot (Lego EV3) en temps réel dans une vidéo.

Mots clés — robot mobile à roues, commande non linéaire, contrôle cinématique, contrôle dynamique, robot mobile non holonome, poursuite, adaptation des gains, suivi visuel, séquence d'images.

Robust control of a wheeled mobile robot

Abstract

The present work in the development of robust control laws is to solve the trajectory tracking problem of non holonomic mobile robots. The aim is to guarantee the correct behavior of the system and ensure a good trajectory tracking even in the presence of parametric variations of the system. The trajectory tracking consists of guiding the robot through the intermediate points to reach the final destination. The robot must reach its goal in a determined time by trying to minimize the "distance" and "orientation" errors. In this work the model used to demonstrate the control laws is based on the kinematic and dynamic model of the unicycle type mobile robot. We are oriented to various approaches, the first approach is based on the use of a non linear adaptive command by backstepping, the second one concerns sliding mode technique. A controller based on the Lyapunov theory associated to the sliding mode is designed to ensure the trajectory tracking of the wheeled mobile robot. In order to improve static and dynamic performance, fuzzy logic is introduced in order to adapt the gains of the controller. Finally, a practical implementation is made to ensure the tracking of a desired trajectory; the latter is based on the following of an object in a sequence of images in order to locate the position and velocity of the robot (Lego EV3) over time in a video.

Keywords —wheeled mobile robot, nonlinear control, kinematic control, dynamic control, non holonomic mobile robot, tracking, adaptation gains, visual tracking, images sequence.

ملخص

موضوع هذه المذكرة يتمحور حول تطوير قوانين مراقبة قوية لمعالجة مشكلة تتبع الروبوتات المتحركة اللاهلونومية لمسار معين. الهدف من ذلك هو ضمان السلوك الصحيح للنظام وضمان مسار سليم للروبوت حتى في وجود تغييرات بارامترية في النظام. تتبع مسار هو توجيه الروبوت من خلال نقاط وسيطية للوصول إلى الوجهة النهائية، يجب على الروبوت الوصول إلى هدفه في غضون فترة زمنية محددة مسبقاً في محاولة لتقليل الخطأ في "المسافة" و"الاتجاه". النموذج المستخدم في هذا العمل من أجل إدراج قوانين الرقابة و التحكم يستند على نموذج الحركي والديناميكي للروبوت المتحرك مع عجلات من نوع الدراجة الأحادية. عدة طرق اعتمدت للتحكم في مسار الروبوت، تستند الطريقة الأولى على استخدام سيطرة التكيف غير الخطية عن طريق التحكم التراجعي. أما الطريقة الثانية تم فيها تصميم وحدة تحكم على أساس نظرية ليابونوف مرتبطة بتقنية النمط الانزلاقي، بعدها تم تطوير نظام التكيف لثوابت وحدة التحكم على أساس المنطق الضبابي لتحسين الخصائص الثابتة والديناميكية للنتائج المتحصل عليها. تطبيق عملي أدرج في هذا العمل كمرحلة أخيرة يضمن تحقيق المسار المطلوب، ويستند هذا الجزء التجريبي على رصد تحرك جسم في تسلسل الصور لتحديد موقع وسرعة الروبوت (EV3) مع مرور الوقت في شريط فيديو.

الكلمات المفتاحية — الروبوت المتحرك بالعجلات، السيطرة غير الخطية، التحكم الحركي، التحكم الديناميكي، الروبوت المتحرك اللاهلونومي، التتبع، تكيف الثوابت، التتبع البصري، تسلسل الصور.

RESUME ET MOTS CLES	I
SOMMAIRE	II
NOTATIONS ET SYMBOLES	V

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

CHAPITRE 1

ETAT DE L'ART DES ROBOTS MOBILES

1.1. Définition	4
1.2. Introduction	5
1.3. Robot mobile et vision	8
1.4. L'holonomie et la non holonomie	11
1.4.1. L'holonomie	12
1.4.2. La non holonomie	13
1.5. Classification des robots selon la structure mécanique	13
1.5.1. Mobiles à chenilles	14
1.5.2. Mobiles à pattes	14
1.5.3. Mobiles à roues	15
1.6. Modélisation des robots mobile à roues de type unicycle	15
1.6.1. Modélisation cinématique	16
a. Définition	16
b. Roulement sans glissement	17
c. Disposition des roues et centre instantané de rotation	18
d. Hypothèses	19
e. Modélisation cinématique du robot mobile de type unicycle	20
1.6.2. Modélisation dynamique	22
a. Modèle dynamique du robot mobile à roue	22
1.7. Commandabilité d'un robot mobile	23
1.8. Problèmes de commande	26
1.8.1. Suivi de chemin	26
1.8.2. Stabilité de trajectoires	26
1.8.3. Stabilité de configurations fixes	27
1.9. Techniques de commande en robotique mobile	27

1.9.1. Méthode de linéarisé tangent	27
1.9.2. Méthode directe de Lyapunov	28
1.9.3. Théorème de Lasalle et du lemme de Barbalat	28
1.9.4. Le retour d'état linéarisant	28
1.10. Conclusion	29

CHAPITRE 2

COMMANDE PAR BACKSTEPPING DU ROBOT MOBILE

2.1. Introduction	30
2.2. Généralités sur les systèmes non linéaires	31
2.3. Commande par backstepping	34
2.3..1. Introduction	34
2.3.2. Principe de backstepping	34
2.4. Modèle d'état du robot mobile type unicycle	37
2.5. Méthode de poursuite de trajectoire	38
2.6. Conception d'un contrôleur pour la poursuite de trajectoire	40
2.7. Résultats de simulation	44
2.8. Conclusion	47

CHAPITRE 3

COMMANDE A STRUCTURE VARIABLE DU ROBOT MOBILE

3.1. Introduction	48
3.2. Bref historique	48
3.3. Principe de la commande par mode de glissement	49
3.4. Conception de l'algorithme de commande par mode de glissement	51
3.4.1. Choix de la surface de glissement	51
3.4.2. Condition de convergence et d'existence	52
3.4.3. Détermination de la loi de commande	53
3.5. Application à la commande du robot mobile	54
3.5.1. Contrôleur cinématique	55
3.5.2. Contrôleur dynamique	56
3.6. Résultats de simulation	58
3.7. Conclusion	61

CHAPITRE 4

APPORT DE LA LOGIQUE FLOUE A LA COMMANDE DU ROBOT MOBILE

4.1. Introduction	62
-------------------	----

4.2. Bref historique	63
4.3. Concept de base de la logique floue	64
4.3.1. Théorie de l'ensemble flou	64
a. Définition d'un ensemble flou	64
b. Fonction d'appartenance	64
c. Opérations sur les ensembles flous	65
d. Règles d'interférence	65
4.3.2. Variables linguistiques	66
4.4. Structure d'un contrôleur flou	66
4.4.1. Base de connaissances	67
4.4.2. Interface de fuzzification	67
4.4.3. Interface de défuzzification	67
4.4.4. Mécanisme d'interface	68
4.5. Conception du bloc à logique floue	69
4.6. Résultats de simulation	73
4.7. Synthèse des trois approches	76
4.8. Conclusion	76

CHAPITRE 5

IMPLEMENTATION PRATIQUE ET RESULTATS EXPERIMENTAUX

5.1. Introduction	77
5.2. Méthodologie retenue	78
5.3. Description de la plateforme expérimentale	79
5.4. Schéma global vision-commande	80
5.5. Système de vision	82
5.5.1. Détection du mouvement	82
a. Système d'acquisition de vidéo	83
b. Détection des régions en mouvement	83
c. Conversion, filtrage et seuillage	84
d. Suivi de mouvement (motion tracking)	84
5.5.2. Implémentation pratique et résultats	85
5.6. Système de commande	88
5.7. Conclusion	91
CONCLUSION GENERALE	92
ANNEXES	94
REFERENCES BIBLIOGRAPHIQUES	98

NOTATIONS ET SYMBOLES

- $2L$: Distance entre les deux roues
- r : Rayon des roues
- D : Longueur du robot
- v_1 : Vitesse linéaire de la roue droite
- v_2 : Vitesse linéaire de la roue gauche
- u_1 : Accélération de la roue droite
- u_2 : Accélération de la roue gauche
- v : Vitesse linéaire du robot
- w : Vitesse de rotation du robot autour de son Centre Instantané de Rotation (CIR)
- x et y : Coordonnées du robot
- P : Centre de la roue
- Q : Point de contact de la roue avec le sol
- φ : Angle de rotation propre de la roue
- θ : Angle entre le plan de la roue
- \vec{v}_Q : Nullité de la vitesse relative
- \vec{v}_P : Vitesse du centre P de la roue
- $\vec{\omega}$: Vecteur de vitesse de rotation de la roue
- m : Masse du robot
- I : Moment d'inertie robot
- τ_d : Couple appliqué à la roue droite
- τ_g : Couple appliqué à la roue gauche
- θ : Orientation du robot
- ρ : Rayon de courbure
- $J(.)$: Jacobéen
- V : Vecteur de vitesses linéaire et de rotation du robot réel
- V_r : Vecteur de vitesses de référence linéaire et de rotation du robot virtuel

-
- V_c : Vecteur de commande en vitesse
 q : Vecteur des coordonnées généralisées du robot réel
 q_r : Vecteur des coordonnées généralisées du robot de référence
 \tilde{q} : Vecteur d'erreur
 T_e : Matrice de transformation
 q_e : Vecteur d'erreur dans le repère lié au châssis du robot
 e_v : Vecteur d'erreur de vitesse linéaire et de vitesse de rotation du robot
 s : Surface de glissement
 sgn : Fonction signe
 Δf : Paramètre d'incertitudes et de perturbations externes
 μ_A : Fonction d'appartenance
 $M(\cdot)$: Matrice d'inertie symétrique définie positive
 $V_m(\cdot)$: Matrice centripète et Coriolis
 $G(\cdot)$: Vecteur de gravitation
 τ_p : Perturbations aléatoires inconnues bornées
 $B(\cdot)$: Matrice de transformation de l'entrée
 λ : Multiplicateur de Lagrange associé aux contraintes
 $I(x, y, t)$: Valeur d'intensité du pixel (x, y) , sur l'image " I " au temps t
 $\mu(x, y, t)$: Valeur d'intensité du pixel (x, y) , sur l'image arrière-plan au temps t
 σ : Seuil de seuillage
 v_d : Vitesse linéaire réelle de la roue droite
 v_g : Vitesse linéaire réelle de la roue gauche
 v_{dr} : Vitesse linéaire de référence de la roue droite
 v_{gr} : Vitesse linéaire de référence de la roue gauche
 u_d : consigne de la roue droite
 u_g : consigne de la roue gauche

Introduction générale

INTRODUCTION GENERALE

Dans les années 80, on a vu émerger un domaine dans la science de l'automatique : la robotique. Le livre de Daniel Ichbiah, *Robots, genèse d'un peuple artificiel*, propose la définition suivante en préambule : « Un superordinateur logé dans un corps mobile, capable de fonder ses actions de manière raisonnée sur ce qu'il perçoit du monde extérieur ». Le robot mobile permet d'étendre les capacités d'action des humains dans des environnements variés.

Un robot mobile peut être considéré comme un synonyme de véhicule puisque ce dernier est composé de la structure matérielle et mécanique qui permet au robot d'effectuer un déplacement [1]. D'autres disciplines, telles que l'automatique, sont toutefois impliquées et peuvent présenter un éclairage différent, mais ne changent pas fondamentalement les problèmes qui restent à résoudre.

“La courte histoire de la robotique montre que, pour être à l' hauteur de l'ambition de départ, les chercheurs ont dû faire preuve d'ouverture, les amenant à se spécialiser et à contribuer à des disciplines connexes (l'étude des systèmes non holonomes en commande non linéaires est née des contraintes de roulement sans glissement des robots mobiles à roues), voire à dégager de nombreux champs de recherches maintenant reconnus comme disciplines à part entière” [2].

L'une des problématiques élémentaires dans le domaine de la robotique mobile est l'exécution de trajectoire. Le robot mobile, doté dans certains cas d'un bras manipulateur pour intervenir sur des sites hostiles, doit d'abord atteindre l'endroit où se trouvent les objets à manipuler [3]. Pour ce faire, tout robot mobile doit posséder la faculté de planifier et de poursuivre des trajectoires pour succéder ses missions. La planification est le procédé par

lequel le robot mobile va essayer de trouver le chemin le plus court et le plus surmenant au but à partir des données disponibles sur son environnement (modèle global ou local) [3-6]. La planification globale est une méthode globale supposant connu l'ensemble de l'univers d'évolution du mobile, la planification locale est une méthode locale qui génère un déplacement à partir des informations recueillies par le mobile sur son environnement proche [7-8]. Des recherches importantes ont été consacrées à la planification de mouvement. Dont l'objectif est de transférer un système à partir d'un état initial spécifié à un état final spécifié lorsque la commande de mouvement est de résoudre les problèmes de navigation de base; suivre une trajectoire de référence, la stabilité sur une position souhaitée.

La question qui se pose alors est: comment faire réaliser ces déplacements par le système physique, via la commande des actionneurs dont il est équipé ? Dans ce mémoire on va essayer d'apporter des éléments de réponse à cette question par l'exploitation des commandes simples et efficaces.

Commander un système non holonome pour l'amener d'une configuration de départ à une configuration d'arrivée n'est pas chose aisée, ce problème a attiré l'attention des automaticiens pour la robotique d'où la possibilité d'appliquer la commande non linéaire [8].

La poursuite de trajectoire consiste à guider le robot à travers des points intermédiaires pour arriver à la destination finale. Ce guidage se fait sous une contrainte temporelle, autrement dit, le robot doit rejoindre son but dans un délai prédéfini. Dans la littérature, ce problème est traité comme étant la poursuite d'un robot de référence (robot virtuel) qui se déplace sur la trajectoire désirée avec une certaine allure. Le robot réel doit suivre ce robot virtuel avec précision et essayer de minimiser l'erreur en distance, en variant ses vitesses linéaire et de rotation.

Généralement la conception d'un contrôleur pour la poursuite de trajectoire est divisée en deux niveaux, le niveau haut ou le contrôle cinématique basé sur les propriétés cinématiques du robot. Ce type de contrôle génère les vitesses de référence de translation et de rotation en tenant compte de la position actuelle du robot vis-à-vis la situation désirée (position et vitesse du robot de référence) en faisant abstraction de la dynamique du robot. Au niveau bas, un contrôleur dynamique contrôle les vitesses des roues motrices du robot pour atteindre les vitesses de consigne générées par le contrôleur cinématique.

Le présent manuscrit s'articulera autour de cinq chapitres ; à savoir :

Le premier chapitre, sera consacré à l'état de l'art des robots mobiles et leur application, On exposera une synthèse de lois et de techniques de commande appliquées aux robots mobiles à roues. Les différents types des robots mobiles, leurs structures, les classes de robots mobiles à roues rencontrés le plus fréquemment seront présentés suivi d'une modélisation cinématique et dynamique d'un robot mobile type unicycle.

Le deuxième chapitre sera consacré à la présentation théorique de la technique de backstepping qui est une méthode de commande récursive et représentant un outil pour l'étude de la stabilité dynamique. On exposera le principe de cette commande qui permettra l'exécution de trajectoire du robot mobile à roues. Les résultats de simulation seront présentés et commentés à la fin du chapitre.

Dans le troisième chapitre on traitera les notions de base de l'approche par mode de glissement. On présentera le contrôleur conçu par mode de glissement destiné à la poursuite de trajectoire par un robot mobile à roues de type unicycle. Les résultats de simulation seront également illustrés et interprétés.

Les fondements théoriques de la logique floue seront abordés dans le quatrième chapitre. Un mécanisme d'adaptation des gains basé sur la logique floue sera développé afin d'améliorer les caractéristiques statiques et dynamiques des réponses obtenues.

Le dernier chapitre sera destiné à la validation expérimentale La commande d'un lego EV3 sera élaborée et réalisée. Cette configuration expérimentale sera établie en intégrant le module de traitement d'image en temps réel, le module de communication sans fil et le module de décision de stratégie. On s'intéressera au traitement de la vidéo et au suivi d'objet mobile dans une scène perçue par une caméra couleur fixe.

Ce travail sera clôturé par une conclusion générale à travers laquelle, on analysera les principaux résultats obtenus et on citera les perspectives à envisager comme suite à ce travail.

Chapitre 1

Etat de l'art des robots mobiles

CHAPITRE 1

ETAT DE L'ART DES ROBOTS MOBILES

1.1. Définition

Qu'est-ce que la robotique ? Une discipline qui fait partie des sciences de l'artificiel (Herbert Simon, Prix Nobel) [9].

La robot est une machine équipée de capacités de perception, de décision et d'action qui lui permet d'agir de manière autonome dans son environnement en fonction de la perception qu'il en a et de ses objectifs [10-11]. Le robot est parfois capable d'améliorer ses propres performances par apprentissage automatique ou supervisé par des humains [9]. IL est capable de naviguer de manière autonome dans un environnement complexe, parfois évolutif, partiellement connu ou inconnu, et d'exécuter les tâches programmées sans intervention humaine ou avec une intervention réduite [6]. La robotique est un très bon exemple de domaine pluridisciplinaire qui implique de nombreuses thématiques, fortement liées: la conception mécanique, l'électronique et l'informatique « mécatronique », l'action et le mouvement, et notamment la commande et les contrôles sensori-moteurs la perception « multi-sensorielle » et la représentation de l'environnement, la prise de décision, le raisonnement et l'apprentissage et, enfin, l'interaction et la coopération avec d'autres robots ou avec l'homme [9] et [11].

Un robot est un ensemble de mobiles associés dont les mouvements sont commandés numériquement et synchronisés. Chaque mouvement élémentaire est un axe de commande numérique. Cependant un robot n'est qu'une machine programmable qui ne fait qu'exécuter ce que l'homme lui a appris [12]. Un robot est capable d'extraire de l'information à partir de son environnement et d'utiliser ses connaissances pour décider comment agir [13].

De manière générale, on regroupe sous l'appellation robots mobiles l'ensemble des robots à base mobile, par opposition notamment aux robots manipulateurs. L'usage veut

néanmoins que l'on désigne le plus souvent par ce terme les robots mobiles à roues. Les autres robots mobiles sont en effet le plus souvent désignés par leur type de locomotion, qu'ils soient marcheurs, sous-marins ou aériens. On peut estimer que les robots mobiles à roues constituent le gros des robots mobiles [6].

1.2. Introduction

Historiquement, Shakey, le premier robot mobile à comporter un planificateur d'actions qui fut développé au Stanford Research Institute(SRI), est considéré comme support de recherches conduites en intelligence artificielle qu'est né en 1967 [10] et [2]. Ce robot utilise des télémètres à ultrason et une caméra et sert de plate forme pour la recherche en intelligence artificielle, qui à l'époque travaille essentiellement sur des approches symboliques de la planification [14]. C'est avec la robotique mobile que la robotique s'établit comme une discipline à part entière. Quelle technique mettre en œuvre pour une machine, agissant sur le monde réel, soumise aux lois de la physique qui le régissent, condamnée au calcul, puisse comprendre et exécuter les tâches qui lui sont demandées ? C'est le défi que lançaient les chercheurs du SRI avec Shakey. C'est ce défi que Georges Giralt portait en France, dès 1977, avec le robot mobile Hilare au LAAS-CNRS [2].



Figure (1.1) : Robot Shakey de SRI (Stanford Research Institute).

Une étape importante est à signaler au début des années 1990 avec la mise en œuvre de la robotique réactive, présentée notamment dans les travaux de Rodney Brooks. Cette nouvelle approche de la robotique, qui met la perception au centre de la problématique, a permis de passer de gros robots très lents à de petits robots plus rapides, figure (1.2), beaucoup plus réactifs et adaptés à leur environnement. Ces robots n'utilisent pas ou peu de modélisation du monde, problématique qui s'est avérée être extrêmement complexe [9].



Figure.(1.2) : Genghis, développé par Rodney Brooks au MIT.

Les avancées technologiques réalisées à la fin du siècle dernier et jusqu'à nos jours ont donné naissance à de nouveaux matériaux, des ordinateurs plus rapides, plus petits et moins chers, ...etc. Ceci a propulsé à la recherche en robotique mobile. Ainsi, les robots se déplacent maintenant sur le sol, dans l'air, sous l'eau et même dans l'espace [10].

Le problème de la planification de mouvement et le contrôle des robots mobiles a attiré l'intérêt des chercheurs en vue de ses défis théoriques à cause de leur pertinence évidente dans les applications. Du point de vue de contrôle, la nature particulière de la cinématique non holonomes et complexité dynamique du robot mobile fait que la stabilisation de la rétroaction à une posture donnée ne peut être atteinte via un contrôle en temps invariant lisse (Oriolo et al., 2002). A la fin des années 1970, la planification de mouvement devient un thème de recherche à part entière et c'est à cette époque que Lozano-Pérez a employé le concept de l'espace de configuration en robotique (Loz 83), tel que l'espace de configuration d'un système mécanique est considéré comme un espace des paramètres qui caractérisent le système, cela permet de situer ce système dans l'espace [2].

En général, de nombreux systèmes réels, y compris les robots non holonomes montrent des perturbations externes, non-linéarités et perturbations paramétriques [2]. Dans le cas des systèmes non linéaires, les lois de contrôle classiques peuvent être insuffisantes parce qu'ils ne sont pas forts en particulier lorsque les exigences sur les performances et les caractéristiques dynamiques du système sont strictes [15]. Le théorème de Brockett a montré la non existence de retour d'état satisfaisant pour la stabilisation asymptotique des configurations fixes et il a classé les robots mobiles à roues comme des systèmes non linéaires qui signifie que la théorie de la commande linéaire ne peut pas être appliquée aux problèmes de contrôle des systèmes non holonomes ainsi ils ne peuvent pas être transformés en problèmes de contrôle linéaire [16-17]. Un autre problème qui se pose, la modélisation de robot mobile à roues sous la forme d'un modèle mathématique simple, en réalité, il est difficile, en raison des incertitudes dans sa

modélisation, telles que les erreurs de localisation, les limites des actionneurs électriques, l'inertie [17].

Ces dernières années ont vu un intérêt dans la conception et le développement de diverses approches de contrôle automatique pour haute précision de contrôle de planification de chemin pour robots mobiles à roues, tels que la logique floue et mode glissant... etc. Dans la littérature, plusieurs types d'algorithmes de contrôle basés sur la logique floue ont été adoptés pour résoudre le problème de poursuite de trajectoire (Deiste et Fourie, 1993; Lakehal, Amirat et Pontnau, 1995; Hung Chung, 2006; Antonelli, Chiaverini, et Fusco, 2007; Susnea, Filipescu Vasiliu et Filipescu, 2008), ces travaux ont traité le problème de contrôle robots mobiles sans étudier la stabilité du système, de sorte que l'exécution de la tâche en cours n'est pas garantie théoriquement. Parmi les œuvres qui ont fait face aux incertitudes du modèle, il existe (Tang et al., 2006; Fierro et Lewis, 1998; Xianhua et al., 2005; Chiu et al., 2005; Das et Kar, 2006; Khooban et al, 2012, 2013;. Khooban et Soltan pour, 2013; Casillas et al., 2003; Lughofer, 2013) [18-21].

D'un autre côté certains chercheurs ont traité le problème basant sur la commande en mode glissant. Les premiers travaux sur ces systèmes de contrôle ont été proposés et mis au point au début des années 50 par Emelyanov, puis par le chercheur Utkine à partir des résultats des études du mathématicien Filipov. Différentes recherches ont été faites, sur la base de cette approche ; (Bloch et Drakunov, 1994) ont traité le problème de la stabilisation de robot mobile à roues non holonome exprimée sous forme chaînée, (Utkine et Guldner, 1994, 1995) ont élaboré des lois de contrôle basées sur la théorie Lyapunov pour le mouvement des robots mobiles, (Hamel et al., 1996) a abordé le problème de stationnement du robot mobile, (Dans, 1998) a développé un contrôleur robuste pour la stabilisation exponentielle, (Chih et al., 2009) ont mis en œuvre une adaptation de contrôleur en mode glissant avec lequel les robots mobiles convergent pour référencer les trajectoires souhaitées [22-24].

Au cours des dernières années, un grand intérêt a été consacré à la stabilisation et la poursuite des robots mobiles. Dans le domaine de la robotique mobile, il est une pratique acceptée de travailler avec des modèles dynamiques pour obtenir des lois de control de mouvement stables pour la poursuite de la trajectoire (Fierro et Lewis, 1997). Dans le cas du contrôle d'un modèle dynamique de robots mobiles, les chercheurs généralement utilisent des vitesses linéaires et angulaires du robot (Fierro et Lewis, 1997;. Fukao et al, 2000) ou des

couples (Rajagopalan et Barakat, 1997; Topalov et al, 1998) en tant que vecteur de commande d'entrée [2] et [25].

D'autres chercheurs ont résolu le problème de la stabilité du robot mobile en utilisant l'algorithme non linéaire de backstepping (Tanner et Kyriakopoulos, 2003) avec des paramètres constants (Fierro et Lewis, 1997), ou avec les fonctions connues (Oriollo et al., 2002). Dans (Tanner et Kyriakopoulos, 2003) une loi de commande combinée cinématique-couple est développée en utilisant l'algorithme backstepping et la stabilité est garantie par la théorie de Lyapunov [25-26].

L'objectif dans les travaux de (Jasmin et Al., 2008) [25] était de présenter les méthodes de contrôle non linéaires avancées pour résoudre le problème de poursuite de trajectoire ainsi que la convergence des conditions de stabilité, des contrôleurs de position ont été développés, un contrôleur de backstepping (Velagic et al., 2006) et un contrôleur logique floue (Lacevic et al., 2007) et les paramètres optimaux des deux contrôleurs sont ajustés à l'aide d'algorithmes génétiques [25].

Les contrôleurs de réseaux de neurones sont proposés dans certaines méthodes de poursuite de trajectoire [27]. (Omid et al. 2012) [17] ont présenté dans leur travail une nouvelle conception d'un contrôleur adaptatif de poursuite de trajectoire pour un robot mobile à roues non holonome avec des paramètres inconnus et dynamiques incertains, la capacité d'apprentissage des réseaux de neurones est utilisée pour concevoir un contrôleur de backstepping adaptatif robuste qui ne nécessite pas la connaissance de la dynamique du robot. Les gains de contrôleur cinématiques sont réglés en ligne afin de minimiser l'erreur de vitesse et d'améliorer les caractéristiques de suivi de trajectoire [17].

1.3. Robot mobile et vision

Comme le robot doit exécuter des tâches en milieu interne ou extérieur, cela impose plusieurs contraintes qui doivent être surmontées ou au moins atténuées : aucun contrôle sur l'illumination, conditions météorologiques variables, grande variabilité des scènes selon le temps, la saison, le lieu..., vibrations dans les systèmes d'acquisition, transmission de données limitée ou impossible vers une station centrale, etc. [28]. Le récent développement dans les domaines des capteurs de vision et du traitement d'images ainsi que l'évolution concernant les ordinateurs ont permis d'intégrer des systèmes complexes de vision dans le domaine de la robotique pour la navigation autonome.

En utilisant la vision couleur, un robot peut réaliser des tâches complexes telles que le contrôle des moteurs pour exécuter des déplacements, l'évitement d'obstacles, la coalisation par rapport à des amers naturels, le suivi d'objets dynamiques... ainsi, le domaine d'application du robot devient plus vaste et varie, mais en même temps l'intégration de ces fonctionnalités vont multiplier les problèmes de synchronisation de données, de fusion entre données acquises à des instants différents, de compacité des représentations construites avec ces données et de l'exploitation de ces représentations pour planifier et exécuter des déplacements ou des tâches [28].

Différents travaux dans ce domaine sont menés et mettent en lumière différents problèmes. Papanikolopoulos dans (Pap, 91) effectue une étude comparative de différents modes des contrôles qui sont régulation proportionnelle intégrale PI, boucle fermée à placement de pôles, version modifiée du contrôleur LQG (Lieaire Quadratic Gaussian) où le vecteur de perturbation et le vecteur d'état sont estimés par un filtre de Kalman, les résultats de cette étude montrent qu'un contrôleur PI est utilisable si les mesures visuelles sont précises et qu'il vaut mieux utiliser un contrôle stochastique LQG pour les données bruitées. Pour la modélisation, il utilise les équations du flot optique auxquelles est ajoutée, pour les raisons de précisions, une technique d'appariement nommée SDD optical flow (somme des différences carrées du flot optiques) [29].

En Europe, concernant la robotique en milieu extérieur non structuré, des travaux ont été effectués au centre national d'études spatiales à Toulouse. Le robot de fabrication russe Marsokhod, équipé d'un système de navigation basé sur la stéréovision a permis la validation de plusieurs études sur la robotique d'exploration planétaire. Les équipes réunies au sein du réseau VIRGO (Paletta, 00) (Vision-based Robot Navigation Network), ont lancé quelques machines automatisées surprenantes basées sur le concept de vision active et intentionnelle (exploration des aspects de l'environnement qui lui sont nécessaires à un moment donné). Le réseau VIRGO a pour objectif la coordination de la recherche européenne dans le développement des systèmes robotiques intelligents dotés de la capacité de navigation sur des environnements partiellement inconnus ou dynamiques [28].

Dans les travaux de (Roger Pissard Gibollet, Patrick Rives, 91), intitulé "asservissement visuel appliqué à un robot mobile : état de l'art et modélisation cinématique", une approche a été présentée pour asservir visuellement un robot mobile, le capteur embarqué sur le robot est une caméra, leur étude concerne une commande cinématique d'une base mobile (version char, deux roues motrices et directrices et deux roues folles) dotée d'un bras

manipulateur, ils ont montrés qu'il est possible de contrôler la trajectoire en tenant compte le caractère non holonome de la base [30]. L'objectif de (Alaa Dib, 2011) était de concevoir et de mettre en œuvre des lois d'asservissement permettant à un robot mobile d'exécuter une tâche ou de réaliser une mission donnée, d'où la nécessité de déterminer les mouvements de déplacement du robot par rapport à son environnement local [31]. L'utilisation d'un contrôleur de bas niveau et un planificateur de chemin PI-logique combiné avec un algorithme de surveillance de vitesse a été l'objectif de travail (Ehsan Hashemia et al, 11) pour entraîner des sorties plus précises et appropriées pour le mouvement des robots mobiles omnidirectionnels à quatre roues [32]. Les chercheurs (Leonardo Marin et a., 13) ont présenté une technique de fusion de capteur locale avec une correction de position globale basée sur des événements pour améliorer la localisation d'un robot mobile à ressources informatiques limitées, la mesure de position obtenue à partir d'une caméra zénithale, la solution est testée à l'aide d'un robot LEGO Mindstorm NXT [32].

Le modèle de l'environnement peut être représenté de différentes manières et avec différents niveaux de détail; ces propriétés du modèle dépendent principalement de la tâche à réaliser par le robot, du type d'environnement, ainsi que de la résolution et du champ de vue des capteurs soit montés sur le robot ou fixes, un robot mobile peut disposer de plusieurs capteurs par exemple les capteurs ultrasons, les télémètres laser, les odomètres..., mais les caméras représentent les capteurs de vision qui fournissent au robot le plus d'informations sur l'environnement [28-30].

Dans le domaine de la navigation de robots mobiles, la distinction de l'environnement est basée sur sa structuration [8] et [28] :

- Les environnements structurés peuvent être représentés par des primitives géométriques simples (détection de lignes droites, surfaces planes, couloirs, portes...).
- Les environnements non structurés sont considérés pour des applications en milieu vraiment naturel (site planétaire, polaire, forestier...), en milieu terrestre, ce sont de riches sources d'information contextuelle, de couleur et de texture.
- Les environnements semi-structurés sont en essence des environnements naturels qui ont subi une modification partielle de l'homme, typiquement les sentiers ou chemins laissés par des passages fréquents de l'homme ou des animaux, par exemple dans le cadre d'activités agricoles.

Comme on trouve dans la littérature de la navigation des robots mobile deux grandes classes d'application, cette division classique vient principalement du fait que les problèmes pour les fonctions visuelles d'interprétation et de suivi dans chaque environnement sont très différents. Mais, il existe toujours quelques recouvrements entre ces deux problématiques : par exemple, la reconnaissance ou le suivi d'une affiche sur le mur d'un couloir, ou d'un panneau de signalisation au bord d'une route, peuvent exploiter les mêmes méthodes [28-29].

- La navigation en milieu intérieur : la navigation en milieu intérieur concerne principalement des environnements humains hautement structurés, tel que des bâtiments publics ou des ateliers. Afin de naviguer dans ces environnements, il existe plusieurs possibilités pour les représentations utilisées dans ce modèle du monde, parmi lesquelles on peut citer :
 - une représentation purement géométrique, généralement éparse (par exemple, un ensemble de segments laser),
 - une représentation discrète de type carte d'occupation, pour décrire l'espace libre dans chaque pièce,
 - une représentation topologique qui décrit les connexions entre couloirs et pièces.
- La navigation en milieu extérieur : Pour les milieux naturels d'extérieur, la construction du modèle de l'environnement est nettement plus compliquée (niveau de structuration faible : extraction plus complexe de primitives géométriques), à plus forte raison quand les scènes changent dynamiquement (à cause de la météo, de la saison, des conditions d'illumination, etc.) ou quand d'autres agents dynamiques partagent le même environnement (piétons, autres robots, véhicules...).

1.4. L'holonomie et la non holonomie

Beaucoup de systèmes mécaniques sont sujets à des contraintes de position ou de vitesse. Ces contraintes sont dites holonomes s'il est possible d'intégrer les relations entre les positions ou les vitesses des différents points du système et qui doivent être satisfaites pendant tout le mouvement et qui doivent aboutir à des relations algébriques liant les paramètres de configuration. Le système est dit holonome dans le cas où ces relations peuvent être éliminées par un changement de variables approprié. Par contre le système est dit non holonome si l'élimination n'est plus possible dans le cas de contraintes non intégrables [6] et [33-35].

Dans un cas idéal un robot peut se déplacer dans n'importe quelle direction c'est à dire que le robot possède autant de degrés de liberté que de paramètres de configuration. Le

nombre de degrés de liberté représentant l'ensemble des mouvements que peut réaliser le robot [36].

1.4.1. L'holonomie

Les systèmes holonomes sont beaucoup plus rares dans notre vie quotidienne. Ils ont une structure mécanique complexe qui leur permet de se déplacer dans toutes les directions sans manœuvre.

Dans un cas idéal un robot peut se déplacer dans n'importe quelle direction ce que signifie que le robot possède autant de degrés de liberté que de paramètres de configuration. Le nombre de degré de liberté d'un robot mobile est défini comme le nombre de mouvements indépendants que ce robot peut faire par rapport à un système de coordonnées déterminé. Un robot holonome ou omnidirectionnel est un robot qui possédant trois degrés de liberté [33] et [36] :

- une translation selon X,
- une translation selon Y,
- une rotation selon Z.

Tout mouvement effectué dans un plan peut être décomposé en un maximum de deux mouvements de translation purs effectués respectivement par rapport aux axes X et Y, plus un mouvement de rotation pur autour de l'axe perpendiculaire au plan. Le robot holonome est capable de se déplacer dans n'importe quelle direction quelle que soit son orientation. La société Nomadic disparue en l'an 2000 ; a conçu un système mobile holonome : le XR4000, il dispose de 4 roues motrices et directrices montées comme des roues de chariot. La synchronisation des 8 axes (2 par roue, rotation et orientation) est assurée par une carte dédiée basée sur le microcontrôleur Motorola 68332 et des circuits FPGAs et la structure mécanique est composée d'engrenages coniques, comme le montre la figure suivante [29] et [36].



Figure (1.3) : Le robot XR4000 de la société Nomadic.

1.4.2. La non holonomie

Les systèmes non holonomes sont des systèmes fortement non linéaires avec des caractéristiques qui les rendent difficiles à commander dans certaines situations. En particulier, les robots non holonomes sur roues sont sujets à des contraintes sur les vitesses qui relèvent de la contrainte mécanique de roulement sans glissement des roues. D'un point de vue pratique, cette contrainte se traduit par le fait que les robots non holonomes à roues ne peuvent pas se déplacer latéralement de manière instantanée [34].

Les systèmes mobiles dit non holonomes sont ceux que l'on rencontre le plus dans la vie courante (voiture particulière, camion...etc.). Ces systèmes se caractérisent par une structure mécanique relativement simple (des roues motrices, des roues directrices et des roues libres). Une roue peut avoir une, deux ou trois fonctions. Les robots mobiles à roues font partie de cette catégorie [33] et [37-39].

Ces robots mobiles n'ont que deux degrés de liberté : leurs structures ne permettent qu'une translation sur l'axe X ou Y et une rotation. La figure suivante montre un exemple de ce type de robots.



Figure (1.4) : Le robot EV3.

1.5. Classification des robots selon la structure mécanique

Elle rassemble les éléments mécaniques et électriques permettant d'assurer la fonction de mobilité. Le mouvement est obtenu en associant la propulsion et l'appui sur le milieu, la propulsion, très liée à l'application du robot, est assurée par des techniques variées (électrique, thermique, pneumatique ou hydraulique) en fonction de sa taille, sa masse et de son lieu d'évolution. Le milieu dans lequel évolue le mobile détermine en général le mode d'appui. On dénombre deux grands types de milieux naturels : le milieu solide et le milieu fluide. La majorité des applications des robots relève du premier cas, le second, plus restreint, concerne les robots sous-marins ou robots volants. Nous nous intéressons plutôt au premier

cas qui ouvre des voies d'applications plus nombreuses. Différents mécanismes de locomotion sont utilisés sur sol solide [40].

1.5.1. Mobiles à chenilles

Les chenilles assurent à un mobile, une meilleure adhérence au sol et une faculté de franchissement d'obstacles. Elles sont employées lorsque le sol est perturbé, essentiellement en environnement extérieur, sol accidenté ou de mauvaise qualité de l'adhérence (présence de boue, herbe...). La commande est réalisée en imposant une différence de vitesse aux chenilles droites et gauches [37] et [40].



Figure (1.5) : Robot mobile à chenilles

1.5.2. Mobiles à pattes

La structure précédente n'est plus adaptée dans les cas d'applications sur des terrains avec de grandes différences d'amplitudes, comme un sol jonché de rochers. Dans ce contexte, les mobiles ayant les points d'appuis discrets, figure (1.6), sont la solution au problème. La solution consiste à discrétiser les points de contact entre le mobile et le sol par des pattes, à l'image des animaux terrestres, qui pour la majorité, disposent de deux, quatre, six, voire plus de points d'appuis. La réalisation d'un tel système et l'établissement de sa commande sont complexes [27] et [35].



Figure (1.6) : Robot mobile à pattes

1.5.3. Mobiles à roues

La très grande majorité de robots mobiles développés jusqu'à ce jour sont équipés de roues plutôt que de jambes. Les principales raisons de ce choix technologique sont : les roues sont plus faciles à contrôler, elles procurent une meilleure stabilité au véhicule, elles dissipent moins d'énergie et elles permettent au véhicule de se déplacer plus rapidement. Ces roues peuvent avoir plusieurs fonctions à savoir : roues motrices, roues directrices, roues motrices-directrices, roues libres (roues folles). Les différentes combinaisons des roues assurent une diversité des commandes de la locomotion. La plupart des combinaisons entraîne un déplacement contraint : la non holonomie. Cette caractéristique impose un déplacement uniquement selon la tangente au mouvement des roues. Certaines configurations des roues associées à des structures mécaniques particulières permettent de s'affranchir de ce problème [2] et [6].

Par contre, les roues comportent aussi un certain nombre d'inconvénients par rapport aux mécanismes à jambes. Entre autres, elles ne sont utilisables que sur des terrains relativement plats et dont la surface est dure. Sur des terrains accidentés, les roues doivent être plus grandes que les obstacles qu'elles risquent de rencontrer. Dans le cas de surfaces moins stables, les roues peuvent glisser ou s'enliser, ce qui pose des difficultés de contrôle importantes.

Il existe plusieurs classes de robots à roues déterminées, principalement, par la position et le nombre de roues utilisées. Nous pouvons les classer en quatre classes principales, robot unicycle, robot tricycle, robot voiture et robot omnidirectionnel.

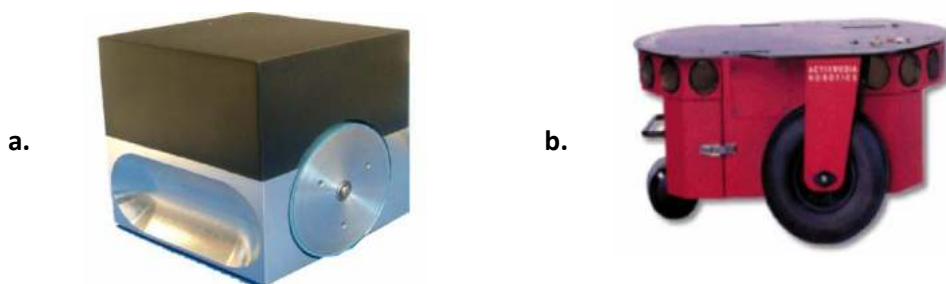


Figure (1.7) : (a) Robot mobile MIABOT (b) Robot mobile Pioneer P3-DX

1.6. Modélisation du robot mobile à roues de type unicycle

Les robots mobiles sont différents par leurs structures et leurs modes de locomotion. D'où la différence dans leurs modèles cinématiques et dynamiques. Notre intérêt dans cette

section est focalisé sur la modélisation cinématique et dynamique d'un robot mobile à roues de type unicycle (à commande différentielle).

Un robot de type unicycle est actionné par deux roues indépendantes, il possède éventuellement des roues libres (folles) pour assurer sa stabilité. Son centre instantané de rotation (CIR) est situé sur l'axe reliant les deux roues motrices [25]. Son déplacement d'un point à un autre peut être réalisable par une suite de rotations et de translations sur des lignes droites [39-40].

La cinématique est l'étude mathématique de mouvements des corps solides sans prendre en compte les forces qui influent sur les corps. Le modèle cinématique d'un robot mobile décrit les mouvements du robot. Afin de concevoir un robot mobile pour des tâches données et élaborer des algorithmes de contrôle pour la commande du robot, nous avons besoin de comprendre le comportement mécanique du robot. La principale question dans ce contexte est la suivante : «Compte tenu d'un certain angle de braquage et d'un certain nombre de tours des roues, où va le robot se retrouver et de quelle manière seront-ils soulignés? Ce problème traite des relations géométriques qui influent sur le système et la relation entre les paramètres de contrôle et le comportement du robot dans son espace d'évolution.

La dynamique est une discipline de la mécanique classique qui étudie les corps en mouvement sous l'influence des actions mécaniques qui leur sont appliquées. Elle traite la masse, la force et la vitesse associées à ces mouvements. Les mouvements d'un robot mobile sont limités par des contraintes dynamiques imposées par la structure du robot et le type d'actionneurs utilisés. Par exemple, un centre de gravité élevé limite le rayon de braquage d'un robot rapide en raison du risque de renversement. Tenant compte de la structure du robot et du type de roues utilisées, chaque roue impose des contraintes sur le mouvement du robot.

1.6.1. Modélisation cinématique

a. Définitions

On note $\mathcal{R} = (O, \vec{x}, \vec{y}, \vec{z})$ un repère fixe quelconque, dont l'axe \vec{z} est vertical et $\mathcal{R}' = (O', \vec{x}', \vec{y}', \vec{z}')$ un repère mobile lié au robot. Le point O' est le centre de l'axe des roues motrices, comme illustré sur la figure (1.8).

On appelle position ou souvent posture du robot le vecteur [6] et [41]:

$$q = \begin{bmatrix} x \\ y \\ \theta \end{bmatrix} \quad (1.1)$$

Où x et y sont respectivement l'abscisse et l'ordonnée du point O dans R et θ l'angle (\vec{x}, \vec{x}') . La posture du robot est donc définie sur un espace M de dimension $m = 3$. La configuration d'un système mécanique est connue quand la position de tous ses points dans un repère donné est connue. La configuration du robot mobile est définie par un vecteur de n coordonnées appelées coordonnées généralisées. [6], [10] et [41]:

$$q = \begin{bmatrix} q_1 \\ q_2 \\ \vdots \\ q_n \end{bmatrix} \quad (1.2)$$

On appelle coordonnées généralisées associées à un système mécanique S , un ensemble de variables dont la donnée, combinée avec les constantes décrivant la géométrie de S , permet de connaître sans ambiguïté la position de chacun des points de S . Un modèle pour S se présentera donc sous la forme d'un système d'équations différentielles reliant les coordonnées généralisées, leurs dérivées et les commandes de S [6], [35] et [41].

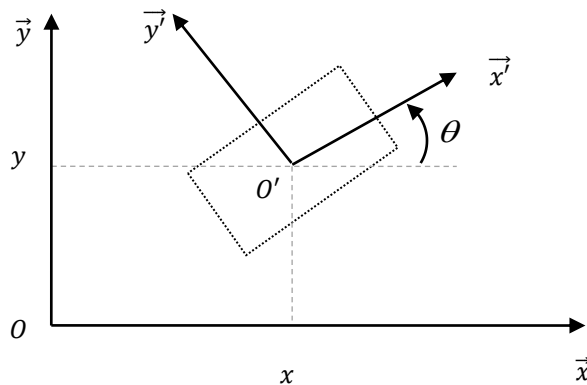


Figure (1.8) : Repérage d'un robot mobile

b. Roulement sans glissement

Considérons une roue qui roule sans glisser sur un sol plan, comme illustré sur la figure (1.9). Soit P le centre de la roue, Q le point de contact de la roue avec le sol, φ l'angle de rotation propre de la roue et θ l'angle entre le plan de la roue et le plan (O, \vec{x}, \vec{z}) . La nullité de la vitesse relative \vec{v}_Q roue/sol au point de contact permet d'obtenir une relation vectorielle entre la vitesse \vec{v}_P du centre P de la roue et le vecteur de vitesse de rotation $\vec{\omega}$ de la roue [35], [40] et [42] :

$$\vec{v}_Q = \vec{v}_P + \vec{\omega} \wedge \overrightarrow{PQ} = \vec{0} \quad (1.3)$$

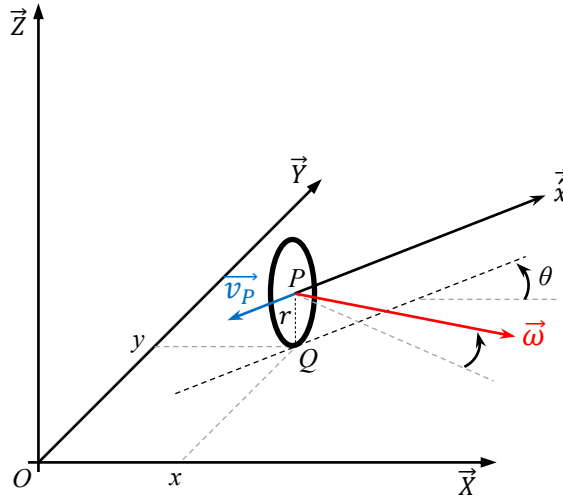


Figure (1.9) : Caractérisation du roulement sans glissement

Les points P et Q ont pour coordonnées respectives $(x \ y \ r)^T$ et $(x \ y \ 0)^T$. Il vient alors :

$$\dot{x}\vec{x} + \dot{y}\vec{y} + (\dot{\theta}\vec{z} + \dot{\phi}(\sin\theta\vec{x} - \cos\theta\vec{y})) \wedge (-r\vec{z}) = \vec{0} \quad (1.4)$$

$$(\dot{x} + r\dot{\phi}\cos\theta)\vec{x} + (\dot{y} + r\dot{\phi}\sin\theta)\vec{y} = \vec{0} \quad (1.5)$$

Il en résulte les deux contraintes cinématiques indépendantes

$$\dot{x} + r\dot{\phi}\cos\theta = 0 \quad (1.6)$$

$$\dot{y} + r\dot{\phi}\sin\theta = 0 \quad (1.7)$$

Que l'on peut transformer pour faire apparaître les composantes de vitesse dans le plan de la roue d'une part et perpendiculairement à la roue d'autre part :

$$-\dot{x}\sin\theta + \dot{y}\cos\theta = 0 \quad (1.8)$$

$$\dot{x}\cos\theta + \dot{y}\sin\theta = -r\dot{\phi} \quad (1.9)$$

Ces contraintes traduisent le fait que le vecteur \vec{v}_P soit dans le plan de la roue et ait pour module $r\dot{\phi}$ [6] et [41].

c. Disposition des roues et centre instantané de rotation

C'est la combinaison du choix des roues et de leur disposition qui confère à un robot son mode de locomotion propre. Sur les robots mobiles, on rencontre principalement trois types de roues, voir figure (1.10) [3], [6] et [42] :

- Les roues fixes dont l'axe de rotation, de direction constante, passe par le centre de la roue;
- Les roues centrées orientables, dont l'axe d'orientation passe par le centre de la roue;

- Les roues décentrées orientables, souvent appelées roues folles, pour lesquelles l'axe d'orientation ne passe pas par le centre de la roue.

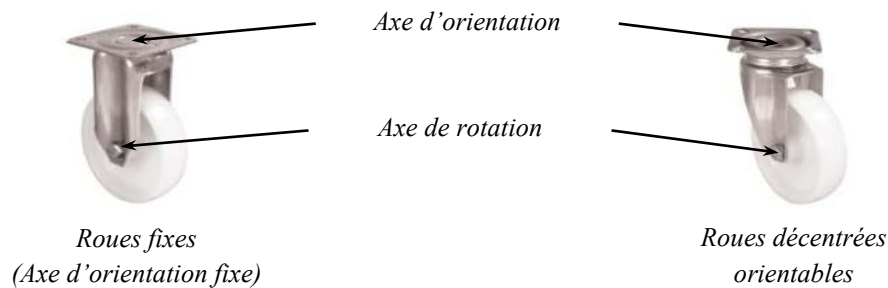


Figure (1.10) : Les principaux types de roues

De manière anecdotique on rencontrera aussi des systèmes particuliers, tels que les roues suédoises, les roues à plusieurs directions de roulement... etc.

Bien évidemment, pour un ensemble de roues donné, toute disposition ne conduit pas à une solution viable. Un mauvais choix peut limiter la mobilité du robot ou occasionner d'éventuels blocages. Par exemple, un robot équipé de deux roues fixes non parallèles ne pourrait pas aller en ligne droite. Pour qu'une disposition de roues soit viable et n'entraîne pas de glissement des roues sur le sol, il faut qu'il existe pour toutes ces roues un unique point de vitesse nulle autour duquel tourne le robot de façon instantanée. Ce point, lorsqu'il existe, est appelé centre instantané de rotation (CIR). Les points de vitesse nulle liés aux roues se trouvant sur leur axe de rotation, il est donc nécessaire que le point d'intersection des axes de rotation des différentes roues soit unique. Pour cette raison, il existe en pratique trois principales catégories de robots mobiles à roues [6] et [42].

d. Hypothèses

Généralement pour la commande cinématique de robots mobiles, un modèle de commande en vitesse est utilisé plutôt qu'un modèle de commande en couple. Les principales raisons de ce choix sont les suivantes :

- Le calcul de la commande est plus simple pour un modèle cinématique que pour un modèle dynamique ;
- Il n'y a pas de paramètres géométriques ou inertiels compliqués à identifier pour un modèle cinématique.

Pour ces raisons, nous considérons dans la suite de cette partie des modèles cinématiques en prenant en compte les hypothèses simplificatrices suivantes :

- Le robot mobile est considéré comme un véhicule rigide évoluant dans un plan horizontal ;
- Les roues conventionnelles sont supposées indéformable, de rayon noté ci-dessous ;
- Chaque contact roue/sol est réduit à un point ;
- Les roues roulent sans glisser sur le sol [21] ;

Nous restreignons notre étude aux robots mobiles de type unicycle.

e. Modélisation cinématique du robot mobile de type unicycle

Pour la modélisation cinématique du robot mobile, nous avons besoin de déterminer les équations de quelques paramètres du robot. Un robot mobile unicycle est 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 par la figure (1.11).

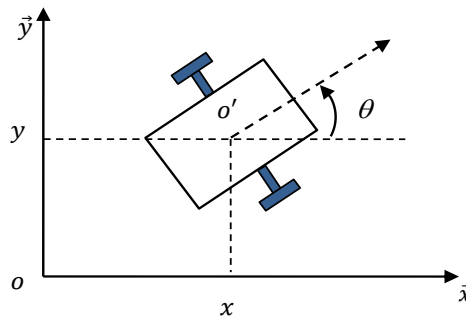


Figure (1.11) : Robot mobile de type unicycle

- Centre instantané de rotation

Les roues motrices ayant même axe de rotation, le CIR du robot est un point de cet axe. Soit ρ le rayon de courbure de la trajectoire du robot, c'est à dire la distance du CIR au point O' , voir figure (1.12). Soit L l'entre-axe et ω la vitesse de rotation du robot autour du CIR. Alors les vitesses des roues droite et gauche, respectivement notées v_d et v_g et définies dans la figure (1.10), vérifient :

$$v_d = -r\dot{\varphi}_d = (\rho + L)\omega \quad (1.10)$$

$$v_g = r\dot{\varphi}_g = (\rho - L)\omega \quad (1.11)$$

Ce qui permet de déterminer ρ et ω à partir des vitesses des roues :

$$\rho = L \frac{\dot{\varphi}_d - \dot{\varphi}_g}{\dot{\varphi}_d + \dot{\varphi}_g} \quad (1.12)$$

$$\omega = -\frac{r(\dot{\varphi}_d + \dot{\varphi}_g)}{2L} \quad (1.13)$$

L'équation (1.12) permet de situer le CIR sur l'axe des roues. Par ailleurs ces équations expliquent deux propriétés particulières du mouvement des robots de type unicycle :

- Si $\dot{\varphi}_d = \dot{\varphi}_g$, le robot se déplace en ligne droite,
- Si $\dot{\varphi}_d = -\dot{\varphi}_g$, alors le robot effectue une rotation sur lui-même.

L'utilisation de ces deux seuls modes de locomotion, bien que limitée, permet de découpler les mouvements et de fournir une solution simple pour amener le robot d'une posture à une autre. C'est sans doute une des raisons du succès de ce type de robots. Pour élaborer une stratégie plus fine de déplacement, il est cependant intéressant de savoir comment la posture du robot est reliée à la commande de ses roues [6].

En ce qui concerne la commande, si l'on se contente de traiter le cas cinématique, on peut considérer que celle ci est donnée, au plus bas niveau, par les vitesses de rotation des roues. Ceci étant, on préfère généralement exprimer cette commande par la vitesse longitudinale du robot, notée v (en O') et sa vitesse de rotation $\dot{\theta}$ (autour de O'). Il y a en effet équivalence entre les deux représentations. D'une part, on a :

$$v = \frac{v_d + v_g}{2} \tag{1.14}$$

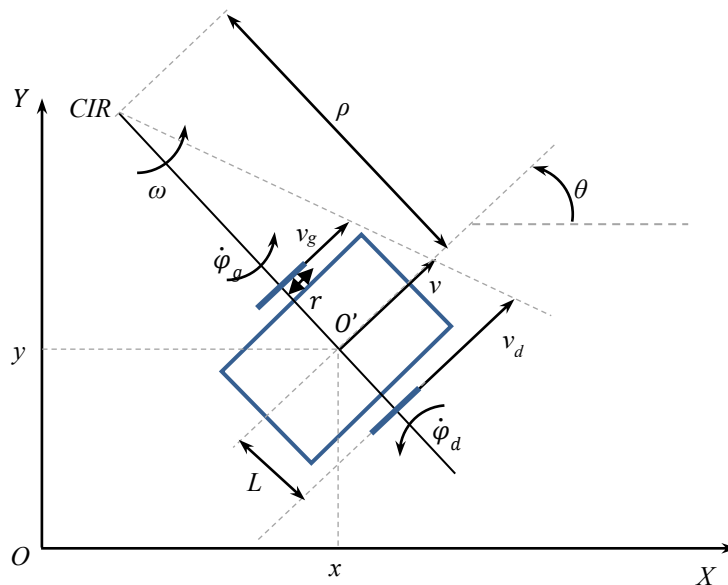


Figure (1.12) : Centre instantané de rotation d'un robot de type unicycle

D'autre part, la vitesse de rotation du robot est égale à la vitesse de rotation autour du CIR :

$$\omega = \dot{\theta} = -\frac{r(\dot{\varphi}_d + \dot{\varphi}_g)}{2L} \tag{1.15}$$

Conformément à l'équation (1.14). On montre que ces relations sont parfaitement inversibles et qu'il y a ainsi équivalence entre les couples $(\dot{\phi}_d, \dot{\phi}_g)$ et (v, ω) . Désormais on utilise plutôt ce dernier couple de grandeurs, plus parlantes, quitte à calculer ensuite les angles ou vitesses de consigne des asservissements des roues.

- Modèle cinématique en posture

Relier la dérivée de la posture à la commande $V = (v \ \omega)^T$ est facile. Une simple considération géométrique donne :

$$\begin{cases} \dot{x} = v \cos\theta \\ \dot{y} = v \sin\theta \\ \dot{\theta} = \omega \end{cases} \quad (1.16)$$

Ce qui s'écrit, sous forme matricielle :

$$\begin{bmatrix} \dot{x} \\ \dot{y} \\ \dot{\theta} \end{bmatrix} = \begin{bmatrix} \cos\theta & 0 \\ \sin\theta & 0 \\ 0 & 1 \end{bmatrix} \begin{bmatrix} v \\ \omega \end{bmatrix} = J(\theta).V \quad (1.17)$$

$$J(\theta) = \begin{bmatrix} \cos\theta & 0 \\ \sin\theta & 0 \\ 0 & 1 \end{bmatrix} \quad (1.18)$$

avec, $J(\theta)$ et V sont, respectivement, la matrice de Jacobéen et le vecteur de vitesse linéaire et de vitesse de rotation du robot. Ce modèle est appelé modèle cinématique en posture du robot mobile [6] et [43-44].

1.6.2. Modélisation dynamique

a. Modèle dynamique du robot mobile à roues

Le modèle dynamique d'un robot mobile avec n coordonnées généralisées q tel que $q \in \mathcal{R}^{n \times 1}$, et r entrées avec $r = n - m$ ($m = 3$), peut être décrit comme suit [43-44] :

$$M(q)\ddot{q} + V_m(q, \dot{q})\dot{q} + F(\dot{q}) + G(q) + \tau_p = B(q)\tau - A^T(q)\lambda \quad (1.19)$$

Où $M(q) \in \mathcal{R}^{n \times m}$ est une matrice d'inertie symétrique définie positive, $V_m(q, \dot{q}) \in \mathcal{R}^{n \times n}$ est la matrice centripète et Coriolis, $F(\dot{q}) \in \mathcal{R}^{n \times 1}$ désigne le coefficient de frottement, $G(q) \in \mathcal{R}^{n \times 1}$ est le vecteur de gravitation, $\tau_p \in \mathcal{R}^{n \times 1}$ désigne des perturbations aléatoires inconnues bornées dont la dynamique est non modélisable, $B(q) \in \mathcal{R}^{n \times (n-m)}$ est la matrice de transformation de l'entrée, $\tau \in \mathcal{R}^{(n-m) \times 1}$ est un vecteur d'entrée de commande, $A \in \mathcal{R}^{m \times n}$ est une matrice associée aux contraintes non holonomes, $\lambda \in \mathcal{R}^{m \times 1}$ est un multiplicateur de Lagrange associé aux contraintes, et \dot{q} et \ddot{q} désignent les vecteurs de vitesse et d'accélération, respectivement. Les variables dans l'équation (1.18) sont définies comme suit [43-44] :

$$M(q) = \begin{bmatrix} m & 0 & 0 \\ 0 & m & 0 \\ 0 & 0 & I \end{bmatrix}, A^T(q) = \begin{bmatrix} -\sin\theta \\ \cos\theta \\ 0 \end{bmatrix}, G(q) = 0, B(q) = \frac{1}{r} \begin{bmatrix} \cos\theta & \cos\theta \\ \sin\theta & \sin\theta \\ L & -L \end{bmatrix}, \tau = \begin{bmatrix} \tau_1 \\ \tau_2 \end{bmatrix}$$

et :

$$V_m(q, \dot{q}) = 0,$$

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

Où m est la masse du robot mobile à roue, I est le moment d'inertie du robot autour de son centre de masse, et $2L$ et r sont respectivement la distance entre les deux roues motrices et le rayon de la roue. Les entrées de commande τ_1 et τ_2 sont des couples générées par les moteurs à courant continu entraînant les roues droite et gauche, respectivement, il serait plus approprié d'exprimer les équations de la dynamique du mouvement en termes de vitesses internes. En substituant (1.16) et sa dérivée dans l'équation (1.19) pré-multipliée par $J^T(\theta)$, on peut obtenir :

$$\bar{M}(q)\dot{V} + \bar{V}_m(q, \dot{q})V + \bar{F}(\dot{q}) + \bar{\tau}_p = \bar{B}(q)\tau \quad (1.20)$$

Où ; $\bar{M} = J^T M J \in R^{2 \times 2}$, $\bar{V}_m = J^T [M J + V m J] \in R^{2 \times 2}$, $\bar{F} = J^T F \in R^{2 \times 1}$, $\bar{\tau}_p = S^T \tau_p$ et $\bar{B} = J^T B$

Etant donné que la distance entre le centre de masse du robot mobile et le centre du repère lié au robot mobile est égal à zéro, alors \bar{V}_m peut être éliminé de l'équation (1.20). Les variables restantes dans l'équation (1.20) sont définies comme :

$$\bar{M}(q) = \begin{bmatrix} m & 0 \\ 0 & I \end{bmatrix}, \bar{B}(q) = \frac{1}{r} \cdot \begin{bmatrix} 1 & 1 \\ L & -L \end{bmatrix}$$

En supposant nulles tous les coefficients de frottements, les perturbations externes et les incertitudes de modélisation, le modèle dynamique de l'équation (1.20) est réduit à un modèle dynamique simplifié d'un robot mobile à roues de type unicycle et est donné par :

$$\dot{V}(t) = E \cdot \tau(t) \quad (1.21)$$

Où la matrice système E est :

$$E = \bar{M}^{-1}(q)\bar{B}(q) = \frac{1}{m \cdot r \cdot I} \begin{bmatrix} I & I \\ Lm & -Lm \end{bmatrix}$$

1.7. Commandabilité d'un robot mobile

Dans cette section, nous rappelons la définition de certains concepts de base de la théorie de la contrôlabilité et les résultats importants liés au niveau de modélisation utilisés en planification de mouvement.

Soit n une variété de dimension et U une classe de fonctions dépendant du temps, prenant leurs valeurs dans un compact K de R^m , le système est sous forme :

$$\dot{q} = X_1(q)u + X_2(q) \quad (1.22)$$

Où u est la commande du système, la $i^{\text{ème}}$ colonne de la matrice $X_1(q)$ est un champ de vecteur noté X_i et $X_2(q)$ représente la dérive. Une trajectoire admissible est une solution du système différentiel vérifiant les conditions initiales et finales imposées et telle que $u \in U$ [2] et [45]. La terminologie de Sussmann permet de donner la définition suivante.

Définition : Le système est localement commandable à partir de q si l'ensemble des points accessibles à partir de q le long d'une trajectoire admissible contient un voisinage de q . Il est localement commandable en temps petit à partir de q si l'ensemble des points accessibles à partir de q le long d'une trajectoire admissible et avant un temps T contient un voisinage de q pour tout T [2] et [45].

La commandabilité locale en petit temps implique la commandabilité locale, l'inverse est faux.

L'analyse de la structure de l'algèbre engendrée par les champs de vecteur définissant le système et munie de l'opérateur crochet de Lie est nécessaire pour établir les propriétés de commandabilité d'un système. Le crochet de Lie de deux champs de vecteur X_1 et X_2 est le champ de vecteur noté $[X_1, X_2]$, la $k^{\text{ème}}$ coordonnée de $[X_1, X_2]$ est :

$$[X_1, X_2][k] = \sum_{i=1}^n \left(X_1[i] \frac{\partial}{\partial q_i} X_2[k] - X_2[i] \frac{\partial}{\partial q_i} X_1[k] \right) \quad (1.23)$$

Théorème : Un système symétrique sans dérive est commandable en temps petit à partir de q si et seulement si la dimension de l'espace vectoriel engendré par la famille des champs de vecteur X_{1i} et de tous leurs crochets de Lie itérés est n au point q .

Ce théorème procure un résultat puissant pour tester la commandabilité en temps petit des systèmes. Il est connu sous le nom de condition du rang de l'algèbre de Lie. L'exemple suivant montre l'utilité de ce théorème.

Robot mobile à deux roues motrices [2], [40] et [45] :

La figure suivante illustre le modèle de ce type de robot, les deux roues sont parallèles, commandées par deux moteurs indépendants, la roue folle assure la stabilité du robot. Le point A , situé au milieu de l'axe des roues motrices, est considéré comme point de référence et θ représente la direction du robot.

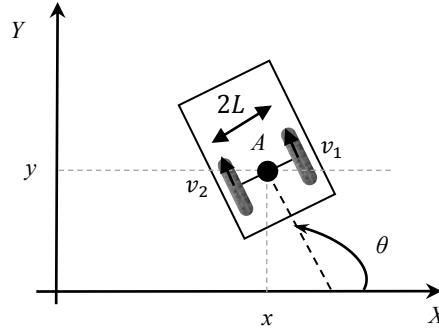


Figure (1.13) : Robot mobile à deux roues motrices

Le modèle de ce dernier est donné par l'équation (1.24)

$$\begin{bmatrix} \dot{x} \\ \dot{y} \\ \dot{\theta} \\ \dot{v}_1 \\ \dot{v}_2 \end{bmatrix} = \begin{bmatrix} \frac{1}{2}(v_1 + v_2)\cos\theta \\ \frac{1}{2}(v_1 + v_2)\sin\theta \\ \frac{1}{L}(v_1 - v_2) \\ 0 \\ 0 \end{bmatrix} + \begin{bmatrix} 0 \\ 0 \\ 0 \\ 1 \\ 0 \end{bmatrix} u_1 + \begin{bmatrix} 0 \\ 0 \\ 0 \\ 0 \\ 1 \end{bmatrix} u_2 \quad (1.24)$$

Où, $2L$ représente la distance entre les deux roues motrices, v_1 et v_2 sont les vitesses des roues et elles sont bornées. u_1 et u_2 sont des accélérations des roues, elles sont également bornées avec $|u_1| \leq u_{1,max}$ et $|u_2| \leq u_{2,max}$. Ces bornes apparaissent comme des obstacles à éviter dans cet espace (dimension 5), le système n'est pas commandable en temps petit, ceci est dû à la présence d'une dérive et au fait que les accélérations sont bornées.

Pour obtenir le modèle cinématique du robot qui s'exprime comme un système de dimension 3, on pose $v = \frac{1}{2}(v_1 + v_2)$ et $w = \frac{1}{2L}(v_1 - v_2)$. Le modèle cinématique du robot mobile peut être réécrit sous la forme suivante :

$$\dot{q} = X_1(q)v + X_2(q)w \quad (1.25)$$

Où, $q = [x \ y \ \theta]^T$, $X_1(q) = [\cos\theta \ \sin\theta \ 0]^T$ et $X_2(q) = [0 \ 0 \ 1]^T$

Les bornes sur v_1 et v_2 induisent des bornes v_{min} et v_{max} sur les nouvelles commandes v et w . Les bornes $u_{1,max}$ et $u_{2,max}$ sont ignorées à ce niveau de modélisation au stade de la recherche d'un chemin. Le nouveau système est symétrique et sans dérive, on peut lui appliquer la condition du rang. Les coordonnées du champ de vecteur $[X_1, X_2]$, crochet de Lie des champs de vecteur X_1 et X_2 , sont $[-\sin\theta \ \cos\theta \ 0]^T$.

On vérifie facilement que la famille $\{X_1(q), X_2(q), [X_1, X_2](q)\}$ est une base de l'espace tangent en q , et ceci pour q . Le modèle cinématique de ce robot est commandable en

petit temps, on montrerait de même que celui de la voiture l'est aussi. La conséquence de cette propriété est l'existence de retours d'états pour commander le système.

1.8. Problèmes de commande

L'un des aspects importants des systèmes de robots mobiles est lié à son mouvement ou le contrôle de la navigation. La question du problème de contrôle ne dépend pas seulement de la cinématique et la dynamique du système de robot mobile, mais aussi les réels éléments individuels du contrôle lui-même. Sans un bon système de contrôle, un robot mobile est pratiquement inutile et inefficace. Par conséquent, le développement d'un robot mobile est significativement influencé par la conception appropriée du système de commande.

Un certain nombre de techniques de contrôle appliquées aux robots ont été proposées par certains chercheurs, dans le livre intitulé « la robotique mobile » sous la direction de J. P. Laumond, les auteurs ont résumé, dans le 2^{ème} chapitre, les problèmes de commande comme suit [2].

1.8.1. Suivi de chemin

Etant donnée une courbe « ζ » du plan, une vitesse d'avancement v (non nulle) pour le robot mobile, ainsi qu'un point « C » fixe sur ce robot, on souhaite que le point « C » suive la courbe « ζ » lorsque le robot roule à la vitesse v . La variable que l'on doit réguler à zéro est donc la distance du point « C » à la courbe. Ce type de problème correspond à l'exécution d'une trajectoire indépendamment du temps, comme dans l'exemple de la conduite sur route lorsque le conducteur cherche à maintenir une distance constante par rapport à des marquages au sol.

1.8.2. Stabilisation de trajectoires

Ce problème se distingue du précédent par le fait que la vitesse d'avancement du véhicule n'est plus fixée à l'avance, du fait que l'on cherche également à réguler la distance parcourue le long de la courbe « ζ ». Cet objectif nécessite de doter la courbe « ζ » d'une loi horaire, c'est à dire de la paramétrer par la variable temporelle « t ». Cela revient à définir une trajectoire $t \rightarrow (x_r(t), y_r(t))$ par rapport à un repère de référence R . On souhaite alors réguler à zéro le vecteur d'erreur $(x_r(t) - x(t), y_r - y(t))$, où $(x(t), y(t))$ désignent les coordonnées du point « C » dans R , à l'instant t . Le problème peut également être interprété comme celui consistant à asservir le véhicule à un véhicule de référence, dont la trajectoire est donnée par $t \rightarrow (x_r(t), y_r(t))$.

1.8.3. Stabilisation de configurations fixes

Pour ce dernier problème, un repère de référence R_0 est donné, ainsi qu'un repère lié au véhicule R_1 . L'objectif est alors de réguler à zéro le vecteur position $((x(t), y(t)))$, exprimé dans R_0 , d'un point « C » lié au véhicule, ainsi que l'orientation $\theta(t)$ de R_1 par rapport à R_0 . Ce problème est celui pour lequel les propriétés de non holonomie interviennent de la façon la plus contraignante, et pour lequel les méthodes classiques de l'automatique linéaire se montrent insuffisantes. Les principales applications de ce problème sont liées aux manœuvres de parking (créneau...) et plus généralement aux applications nécessitant une immobilisation du véhicule s'accompagnant d'un positionnement précis.

1.9. Quelques techniques de commande en robotique mobile

On trouve dans la littérature plusieurs outils de commande des systèmes qui peuvent être appliqués aux problèmes de contrôle des robots mobiles [3].

- La méthode du linéarisé tangent ;
- La technique directe de Lyapunov ;
- L'utilisation du théorème de LaSalle et du lemme de Barbalat ;
- Le retour d'état linéarisant ;
- Commande non linéaire (mode de glissement, backstepping, etc.) ;
- La commande à base de l'intelligence artificielle.

Un résumé sur les quatre premières techniques sera présenté dans cette partie et en ce qui concerne les deux dernières catégories de commande, nous allons voir par la suite l'exploitation de ces deux approches pour la commande du robot.

1.9.1. Méthode de linéarisé tangent

Le principe de la méthode consiste à développer en série de Taylor, au voisinage d'un point d'équilibre x^* , la fonction qui représente la dynamique du système, en tenant compte la possibilité de considérer l'approximation linéaire du système au voisinage d'un point d'équilibre. Puis, en ne considérant que le terme linéaire de la série, on approxime la dynamique de l'erreur de configuration de l'état par rapport à x^* . Finalement, on en déduit une loi de commande stabilisante à partir de la théorie des systèmes linéaires. La commande ainsi obtenue ne peut évidemment être utilisée que pour des positions initiales se trouvant près de x^* et en considérant seulement de petites perturbations sur le système. Bien que l'approche du

linéarisé tangent apparaisse très simple, elle présente l'inconvénient de ne définir la loi de commande que localement autour du point d'équilibre, et ne donne pas un résultat global.

1.9.2. Méthode directe de Lyapunov

Cette approche consiste à concevoir une loi de commande en fonction de l'état, et éventuellement du temps, de manière à construire une fonction de potentiel avec un minimum global à l'origine de coordonnées. Les théorèmes de stabilité, au sens de Lyapunov, garantissant que lorsque ces fonctions énergie décroissent au cours du temps, le système converge à l'origine. L'annexe A précise le théorème de Lyapunov.

1.9.3. Théorème de LaSalle et lemme de Barbalat

Dans le cas où l'analyse de Lyapunov ne permet pas de garantir la stabilité asymptotique d'un système. Le théorème de LaSalle et le Lemme de Barbalat peuvent être utilisés pour l'assurer. Des exemples sur l'application du lemme de Barbalat, ou bien du théorème de LaSalle, peuvent être trouvés dans (Slotine, 91).

1.9.4. Le retour d'état linéarisant

D'un autre point de vue, la linéarisation par retour d'état peut être interprétée comme une nouvelle modélisation du système. Il ne s'agit pas d'une approximation du système mais d'une transformation exacte, pour cette raison, cette technique donne des résultats précis. Néanmoins, à cause de la nature non linéaire du système, la méthode par linéarisation exacte ne permet pas dans tous les cas d'obtenir une modélisation globale, c'est à dire le résultat de la transformation n'est valable que dans une région de l'espace. Le principal intérêt d'une telle représentation est de pouvoir utiliser la théorie des systèmes linéaires qui est beaucoup plus riche que celle des systèmes non linéaires, les travaux de (Isidori, 89; Slotine, 91) présentent des détails sur cette technique.

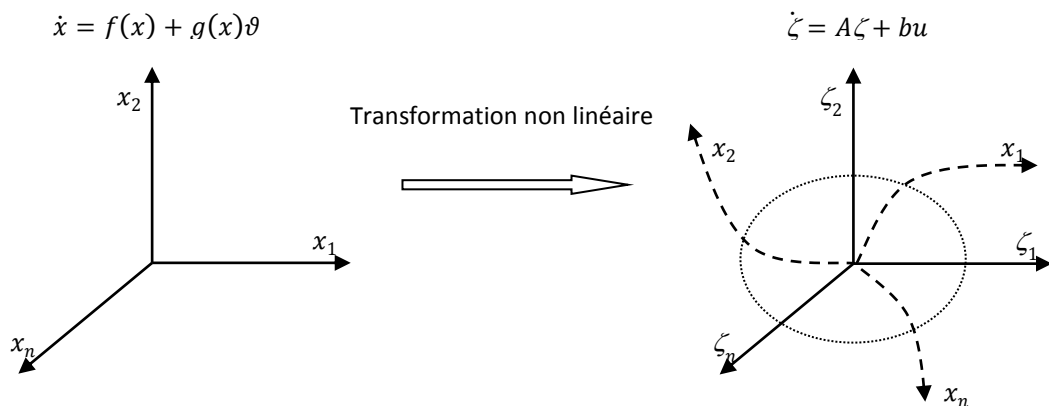


Figure (1.14) : Principe du retour d'état linéarisant

1.10. Conclusion

Un état de l'art non exhaustif sur la robotique mobile et son application a été présenté dans ce premier chapitre. Les modèles adoptés dans ce domaine ont été également examinés, on s'est intéressé spécialement au robot mobile de type unicycle. Ce modèle sera exploité dans les chapitres qui suivent afin d'élaborer plusieurs techniques de commande, en l'occurrence la commande par backstepping, la commande par mode glissant et la commande hybride "mode glissant-floue".

Chapitre 2

Commande par backstepping du robot mobile

CHAPITRE 2

COMMANDE PAR BACKSTEPPING

DU ROBOT A ROUES

2.1. Introduction

Pendant les deux dernières décennies, beaucoup de progrès ont été faits dans le domaine de la commande des systèmes non linéaires avec la première version de la linéarisation entrée-sortie [46-48], les outils développés dans ce domaine ont été appliqués à l'étape de la conception et à l'exécution des unités de commande dans les systèmes non linéaires, ces derniers sont devenus mieux développés plus structurés par rapport aux anciens outils [49-50]. La technique du backstepping fait partie de ces nouvelles percées dans ce domaine, elle est relativement récente, elle donne un outil pour la conception récursive de la loi de commande [40-50].

(M. Krstić et P. V. Kokotović, 1995) ont introduit des méthodes utilisant des changements de variables récursifs appelés backstepping, sur des classes de systèmes triangulaires non linéaires paramétrés [46-48]. Le backstepping a été développé par (Kanellakopoulos et al., 1991) et inspiré par les travaux de (Feurer et Morse, 1978) d'une part et (Tsinias, 1989) et (Kokotović et Sussmann, 1989) d'autre part. De façon générale, les lois de commande proposées satisfont de bonnes propriétés de robustesse et d'atténuation de perturbations, mais ne s'appliquent qu'à des classes restreintes de systèmes et n'utilisent que des contrôleurs statiques. En introduisant un changement de variables dynamiques et en utilisant une fonction de Lyapunov, des contrôleurs et des lois d'adaptation simples ont été obtenues pour des classes plus générales de systèmes non linéaires en 1980 [51].

Dans le présent chapitre, la commande non linéaire par backstepping d'un robot mobile de type unicycle sera étudiée et simulée.

2.2. Généralités sur les systèmes non linéaires

Un système physique est linéaire si la relation entre les grandeurs d'entrée et la ou les grandeurs de sortie est une équation différentielle linéaire à coefficients constants, là où on peut appliquer le théorème de superposition. Les systèmes non linéaires, par opposition aux systèmes linéaires, sont les systèmes physiques qui sont régis par des équations différentielles à coefficients variables (équations non linéaires), décrivant l'évolution temporelle des variables constitutives du système sous l'action d'un nombre fini de variables indépendantes appelées entrées ou variables de commande, ou simplement commandes, que l'on peut choisir librement pour réaliser certains objectifs [50] et [52]. Les entrées peuvent être choisies en boucle ouverte, c'est à dire ne dépendant que du temps, ou en boucle fermée, c'est à dire comme des fonctions des variables mesurées, appelés observations, qui rendent compte de l'état du système à chaque instant [52]. Un système est non linéaire s'il ne vérifie pas le principe de superposition, les conditions de proportionnalité et d'additivité ne s'appliquent plus aux systèmes non linéaires [50] et [52].

Lors de l'étude des systèmes non linéaires on se heurte à plusieurs difficultés [52].

- L'analyse par des fonctions de transfert est impossible ;
- La notion des pôles disparaît ;
- Un système non linéaire possède en général plusieurs points d'équilibre et l'étude de leur stabilité est plus complexe que dans le cas linéaire pour lequel le concept de stabilité est global.

Si le système à commander présente des faibles non linéarités, dans ce cas il suffit de les ignorer et les traitant comme des perturbations affectant un modèle linéaire du système. Cette approche peut s'appliquer à une grande classe de systèmes et elle permet de profiter de l'arsenal de la théorie du contrôle des systèmes linéaires qui est maintenant assez complet et maîtrisé [50] et [52]. En outre, certains phénomènes sont difficilement linéarisables (saturations, seuils, etc.), et seule une approche purement non linéaire permet d'expliquer des comportements globaux.

La forme la plus utilisée pour la représentation des systèmes non linéaires est la suivante [50]:

$$\dot{x} = f(x(t), u(t), t), \quad \forall t \geq 0 \quad (2.1)$$

Où t est le temps, $x(t) \in R^n$ est le vecteur d'état, $u(t) \in R^m$ est le vecteur de commande ou d'entrée.

$f(x(t), u(t), t): R^n \times R^m \times R^+ \rightarrow R^n$: est une fonction non linéaire.

a. Système autonome :

Le système non linéaire défini par l'équation (2.1) est dit autonome si la fonction f ne dépend pas explicitement du temps t , c'est à dire le système peut être écrit sous la forme :

$$\dot{x} = f(x(t), u(t)), \quad \forall t \geq 0 \tag{2.2}$$

Parfois, on utilise le terme de (invariant dans le temps ou stationnaire) au lieu d'autonome. Dans le cas contraire le système (2.1) est dit non autonome, ou (variant dans le temps. [52].

b. Points d'équilibre :

Le point $x_e \in R^n$ est dit point d'équilibre du système non linéaire non forcé :

$$\dot{x} = f(x(t), t), \quad \forall t \geq 0 \tag{2.3}$$

Si

$$\dot{x} = f(x_e, t) = 0 \quad \forall t \geq 0 \tag{2.4}$$

Si x_e est un point d'équilibre du système (2.4) alors l'équation différentielle :

$$\dot{x} = f(x(t), t), \quad \forall t \geq t_e, \quad x(t_e) = x_e \tag{2.5}$$

admet une solution unique :

$$x(t) = x_e, \quad \forall t \geq t_e \tag{2.6}$$

c. Linéarisation autour d'un point de fonctionnement

Soit un système non linéaire représenté mathématiquement par une relation $y = f(x)$, si ce système est disponible sous forme graphique par exemple comme celui présenté dans la figure (2.1), la linéarisation de ce modèle mathématique autour d'un point de fonctionnement ou point d'opération est réalisé en considérant un ensemble de tangentes à ces points x_0 caractérisé par un ensemble d'intervalles $[x_{i-1} x_{i+1}]$ [53-54].

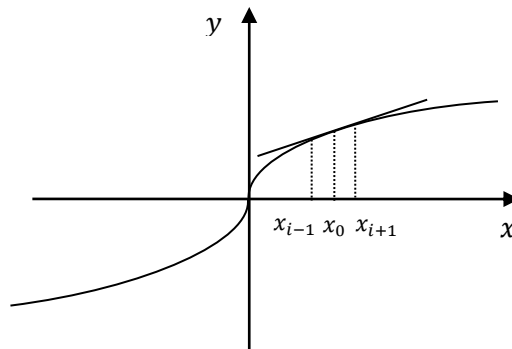


Figure (2.1) : Principe de linéarisation autour d'un point de fonctionnement

d. Linéarisation par retour d'état

Cette méthode est basée sur la théorie de la géométrie différentielle pour la commande des systèmes non linéaires. Elle consiste à trouver un état et à transformer partiellement ou complètement les systèmes dynamiques non linéaires en systèmes dynamiques linéaires d'une manière algébrique. Par conséquent cette nouvelle approche permet l'utilisation des techniques classiques de commande linéaire connues [53].

e. Système non linéaire sous forme triangulaire

La forme générale de cette classe de signaux est comme suit :

$$\begin{aligned}
 \dot{x}_1 &= x_2 + \varphi_1(x_1)^T \theta \\
 \dot{x}_2 &= x_3 + \varphi_2(x_1, x_2)^T \theta \\
 &\vdots \\
 &\vdots \\
 &\vdots \\
 \dot{x}_{n-1} &= x_n + \varphi_{n-1}(x_1, \dots, x_{n-1})^T \theta \\
 \dot{x}_n &= \beta(x)u + \varphi_n(x)^T \theta \\
 y &= x_1
 \end{aligned}
 \tag{2.7}$$

La figure suivante représente le diagramme général d'un système à commander par la méthode de backstepping, il s'agit d'un système sous forme triangulaire et d'ordre n avec $\beta(x) = 1$.

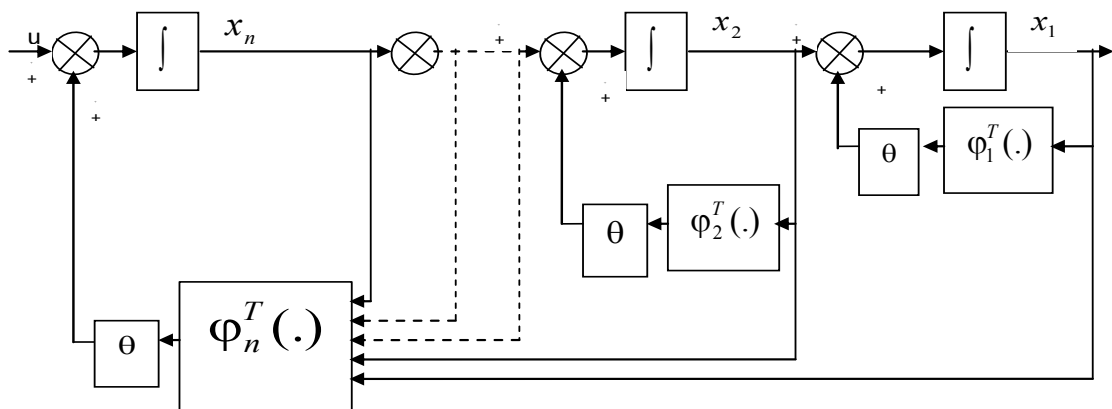


Figure (2.2) : Schéma du système sous forme triangulaire et d'ordre n

Afin d'illustrer la procédure récursive de la méthode de backstepping, on considère que la sortie du système $y = x_1$ désirée suit le signal de référence y_{ref} . Le système étant d'ordre n , la mise en œuvre s'effectue en n étapes [52] et [56].

Etape 1

On commence par la première équation du système (2.8), où x_2 sera considérée comme une commande virtuelle intermédiaire. La première référence désirée est notée :

$$(x_1)_d = \alpha_0 = y_{ref} \quad (2.9)$$

Ce qui conduit à l'erreur de régulation suivante :

$$z_1 = x_1 - \alpha_0 \quad (2.10)$$

Ainsi sa dérivée est :

$$\dot{z}_1 = \dot{x}_1 - \dot{\alpha}_0 = f_1(x_1) + g_0(x_1)x_2 - \dot{\alpha}_0 \quad (2.11)$$

Pour un tel système, nous construisons d'abord la fonction de Lyapunov V_1 sous une forme quadratique :

$$V_1 = \frac{1}{2} z_1^2 \quad (2.12)$$

Sa dérivée temporelle est :

$$\dot{V}_1 = z_1 \dot{z}_1 = z_1 [f_1(x_1) + g_0(x_1)x_2 - \dot{\alpha}_0] \quad (2.13)$$

Un choix judicieux de x_2 rendrait \dot{V}_1 négative et assurerait la stabilité pour la dynamique de (2.11), pour cela, prenons : $x_2 = \alpha_1$ telle que :

$$f_1(x_1) + g_0(x_1)x_2 - \dot{\alpha}_0 = -k_1 z_1 \quad (2.14)$$

Où $k_1 > 0$ est une constante de conception.

Ainsi, la loi de commande pour le système (2.11) sera donnée par :

$$\alpha_1 = \frac{1}{g_0(x_1)} [-k_1 z_1 + \dot{\alpha}_0 - f_1(x_1)] \quad (2.15)$$

Ce qui implique

$$\dot{V}_1 = -k_1 z_1^2 < 0 \quad (3.16)$$

Etape 2

Maintenant, la nouvelle référence désirée sera la variable de commande pour le sous système précédent (2.11) :

$$(x_2)_d = \alpha_1 \quad (2.17)$$

D'où l'erreur de régulation :

$$z_2 = x_2 - \alpha_1 \quad (2.18)$$

Sa dérivée est :

$$\dot{z}_2 = \dot{x}_2 - \dot{\alpha}_1 = f_2(x_1, x_2) + g_1(x_1, x_1)x_3 - \dot{\alpha}_1 \quad (2.19)$$

Pour le système (2.19), la fonction de Lyapunov étendue est :

$$V_2 = V_1 + \frac{1}{2}z_2^2 = \frac{1}{2}[z_1^2 + z_2^2] \quad (2.20)$$

Dont la dérivée est :

$$\dot{V}_2 = \dot{V}_1 + z_2\dot{z}_2 = -k_1z_1^2 + z_2[f_2(x_1, x_2) + g_1(x_1, x_1)x_3 - \dot{\alpha}_1] \quad (2.21)$$

Le choix de x_3 qui stabilisera la dynamique du système (2.18) et rendra \dot{V}_2 négative est : $x_3 = \alpha_2$, telle que :

$$f_2(x_1, x_1) + g_1(x_1, x_1)x_3 - \dot{\alpha}_1 = k_2z_2 \quad (2.22)$$

Où $k_2 > 0$ est une constante de conception. Ainsi, la loi de commande pour le système (2.19) sera donnée par :

$$\alpha_2 = \frac{1}{g_1(x_1, x_2)}[-k_2z_2 + \dot{\alpha}_1 - f_2(x_1, x_2)] \quad (2.23)$$

avec ;

$$\dot{\alpha}_1 = \frac{g_0(x_1)[-k_1z_1 + \ddot{\alpha}_0 - \dot{f}_1(x_1)] - [-k_1z_1 + \dot{\alpha}_0 - f_1(x_1)]\dot{g}_0(x_1)}{g_0^2(x_1)} \quad (2.24)$$

Un tel choix implique que :

$$\dot{V}_2 = -k_1z_1^2 - k_2z_2^2 < 0 \quad (2.25)$$

Etape n

De la même façon, pour cette étape la référence à suivre sera :

$$(x_n)_d = \alpha_{n-1} \quad (2.26)$$

D'où l'erreur de régulation :

$$z_n = x_n - \alpha_{n-1} \quad (2.27)$$

Sa dérivée est :

$$\dot{z}_n = \dot{x}_n - \dot{\alpha}_{n-1} = f_n(x_1, \dots, x_n) + g_n(x_1, \dots, x_n)u - \dot{\alpha}_{n-1} \quad (2.28)$$

Pour le système (2.28), la fonction de Lyapunov étendue est :

$$V_n = V_1 + V_1 + \frac{1}{2}z_n^2 = \frac{1}{2} [z_1^2 + \dots + z_n^2] \quad (2.29)$$

Sa dérivée est :

$$\begin{aligned} \dot{V}_n &= \dot{V}_1 + \dots + z_n \dot{z}_n \\ &= -k_1 z_1^2 + \dots + z_n [f_n(x_1, \dots, x_n) + g_n(x_1, \dots, x_n)u - \dot{\alpha}_{n-1}] \end{aligned} \quad (2.30)$$

Dans cette dernière étape, on est arrivé à déduire la loi de commande pour le système entier, un bon choix doit satisfaire :

$$f_n(x_1, \dots, x_n) + g_n(x_1, \dots, x_n)u - \dot{\alpha}_{n-1} = -k_n z_n \quad (2.31)$$

Où $k_n > 0$ est une constante de conception.

Ainsi, la loi de commande pour le système entier sera donnée par :

$$u = \frac{1}{g_n(x_1, \dots, x_n)} [-k_n z_n + \dot{\alpha}_{n-1} - f_n(x_1, \dots, x_n)] \quad (2.32)$$

Ce qui garantit la négativité de la dérivée de la fonction de Lyapunov étendue :

$$\dot{V}_n = -k_1 z_1^2 - \dots - k_n z_n^2 < 0 \quad (2.33)$$

2.4. Modèle état du robot mobile type unicycle

Dans le chapitre précédent, une étude de la modélisation du robot à roues a été faite et nous avons vu que le robot mobile à roues peut être décrit par le modèle cinématique et le modèle dynamique. Dans ce qui suit, une modélisation globale du robot mobile type unicycle sera donnée, elle regroupe les deux modèles (cinématique et dynamique), ce qui nous permettra, par la suite, de passer à la commande backstepping du robot (voir section 1.6).

Modèle dynamique :

On définit les variables x_1 et x_2 la vitesse linéaire et la vitesse de rotation du robot, respectivement, ainsi l'équation (1.21) peut être écrite sous forme équation matricielle comme suit :

$$M \begin{bmatrix} \dot{x}_1 \\ \dot{x}_2 \end{bmatrix} = B \begin{bmatrix} \tau_1 \\ \tau_2 \end{bmatrix} \Rightarrow \begin{bmatrix} \dot{x}_1 \\ \dot{x}_2 \end{bmatrix} = M^{-1} B \begin{bmatrix} \tau_1 \\ \tau_2 \end{bmatrix} = \begin{bmatrix} E_{11} & E_{12} \\ E_{21} & E_{22} \end{bmatrix} \begin{bmatrix} \tau_1 \\ \tau_2 \end{bmatrix} \quad (2.34)$$

Avec ; $E = \begin{bmatrix} E_{11} & E_{12} \\ E_{21} & E_{22} \end{bmatrix}$

Modèle cinématique :

On définit les variables x_3 , x_4 et x_5 les coordonnées cartésiennes et l'orientation du robot, respectivement, ainsi :

$$q = \begin{bmatrix} x \\ y \\ \theta \end{bmatrix} = \begin{bmatrix} x_3 \\ x_4 \\ x_5 \end{bmatrix} \tag{2.35}$$

$$\dot{q} = J(\theta)V \Rightarrow \dot{q} = \begin{bmatrix} \dot{x}_3 \\ \dot{x}_4 \\ \dot{x}_5 \end{bmatrix} = \begin{bmatrix} \cos\theta & 0 \\ \sin\theta & 0 \\ 0 & 1 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} \tag{2.36}$$

Modèle complet:

La modélisation du robot est simplifiée dans la représentation d'état suivante :

$$\begin{bmatrix} \dot{x}_1 \\ \dot{x}_2 \\ \dot{x}_3 \\ \dot{x}_4 \\ \dot{x}_5 \end{bmatrix} = \begin{bmatrix} 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & \cos x_5 & 0 \\ 0 & 0 & 0 & \sin x_5 & 0 \\ 0 & 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ x_3 \\ x_4 \\ x_5 \end{bmatrix} + \begin{bmatrix} E_{11} & E_{12} \\ E_{21} & E_{22} \\ 0 & 0 \\ 0 & 0 \\ 0 & 0 \end{bmatrix} \begin{bmatrix} \tau_1 \\ \tau_2 \end{bmatrix} \Rightarrow \dot{X} = AX + BU \tag{2.37}$$

D'où la forme matricielle est :

$$\dot{X} = AX + BU = [A_{11} + A_{12}\cos x_5 + A_{13}\sin x_5]X + BU \tag{2.38}$$

avec ;

$$A_{11} = \begin{bmatrix} 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 \end{bmatrix}; A_{12} = \begin{bmatrix} 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \end{bmatrix}; A_{13} = \begin{bmatrix} 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 \end{bmatrix}; B = \begin{bmatrix} E_{11} & E_{12} \\ E_{21} & E_{22} \\ 0 & 0 \\ 0 & 0 \\ 0 & 0 \end{bmatrix} \text{ et } U = \begin{bmatrix} \tau_1 \\ \tau_2 \end{bmatrix}$$

2.5. Méthode de poursuite de trajectoire

Pour formuler le problème de commande de la poursuite de trajectoire, nous définissons un robot mobile de référence (virtuel) qui génère une trajectoire désirée à suivre par le robot réel, figure (2.3), [58]. Cette trajectoire est donnée par :

$$\dot{q}_r = J(\theta_r)V_r \tag{2.39}$$

avec :

- $J(\theta_r)$: est le jacobéen défini dans le premier chapitre, équation (1.18).
- $q_r(t) = [x_r(t) \ y_r(t) \ \theta_r(t)]^T \in R^3$: désigne le vecteur de coordonnées généralisées du robot de référence.

- $V_r(t) = [v_r(t) \ w_r(t)]^T \in R^2$: représente le vecteur des vitesses de référence linéaire et de rotation.

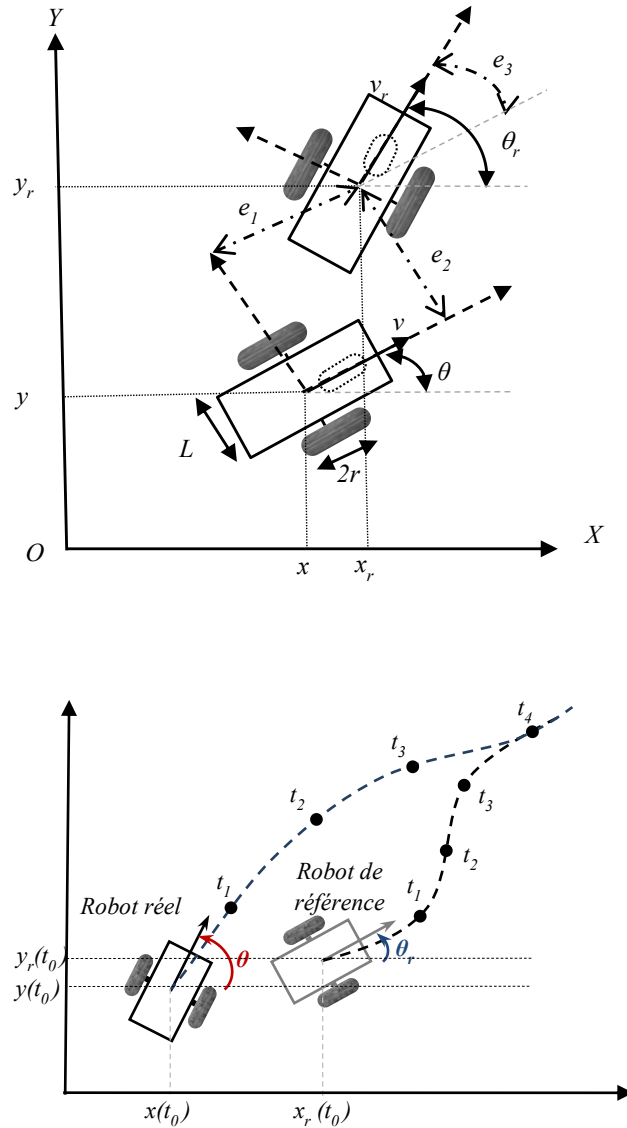


Figure (2.3) : Caractérisation de la poursuite de trajectoire

La trajectoire de référence est générée en respectant les contraintes de non holonomie pour qu'elle soit faisable. En pratique, le vecteur $V_r(t)$ est dérivable et borné.

La solution du problème de poursuite de trajectoire consiste à trouver le vecteur de commande en vitesse $V_c = [v_c, w_c]^T$, tel que $q \rightarrow q_r$ quand $t \rightarrow \infty$. En général, le problème de poursuivre une trajectoire désirée consiste à poursuivre un robot mobile de référence connue par sa posture (configuration) instantanée donnée par $[x_r(t) \ y_r(t) \ \theta_r(t)]$. Par conséquent, nous définissons les erreurs entre les positions et les orientations réelles et désirées par :

$$\tilde{q} = q_r - q = [(x_r - x)(y_r - y)(\theta_r - \theta)]^T \quad (2.40)$$

Nous définissons aussi le vecteur q_e , vecteur d'erreur de poursuite, qui est l'expression du vecteur de référence q_r dans le repère lié au châssis du robot :

$$q_e = \begin{bmatrix} e_1 \\ e_2 \\ e_3 \end{bmatrix} = \begin{bmatrix} \cos \theta & \sin \theta & 0 \\ -\sin \theta & \cos \theta & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x_r - x \\ y_r - y \\ \theta_r - \theta \end{bmatrix} \quad (2.41)$$

On appelle T_e la matrice de transformation des coordonnées du repère fixe dans le repère lié au robot:

$$T_e = \begin{bmatrix} \cos \theta & \sin \theta & 0 \\ -\sin \theta & \cos \theta & 0 \\ 0 & 0 & 1 \end{bmatrix} \quad (2.42)$$

En remplaçant la matrice T_e dans l'équation (2.41), nous obtenons :

$$q_e = T_e(q_r - q) \quad (2.43)$$

Le vecteur des variations d'erreurs est écrit sous forme :

$$\begin{bmatrix} \dot{e}_1 \\ \dot{e}_2 \\ \dot{e}_3 \end{bmatrix} = v \begin{bmatrix} -1 \\ 0 \\ 0 \end{bmatrix} + w \begin{bmatrix} e_2 \\ -e_1 \\ -1 \end{bmatrix} + \begin{bmatrix} v_r \cos e_3 \\ v_r \sin e_3 \\ w_r \end{bmatrix} \quad (2.44)$$

Le but des lois de commande à concevoir est de faire tendre le vecteur q_e à zéro.

2.6. Conception d'un contrôleur pour la poursuite de trajectoire

L'architecture du contrôleur conçu pour la commande d'un robot mobile de type unicycle est représentée sur la figure (2.4). La consigne est donnée par le vecteur V_r qui représente les vitesses de référence linéaire et de rotation [59-61]. Le bloc trajectoire de référence a pour objectif la génération de la trajectoire désirée en intégrant le vecteur V_r . La sortie du bloc trajectoire de référence est le vecteur de posture de référence instantanée q_r qui est comparée avec le vecteur de posture actuelle du robot q pour donner le vecteur d'erreur \tilde{q} exprimé dans le repère lié à l'espace d'évolution du robot [61-64].

Pour commencer la conception du contrôleur, le modèle donné par l'équation (1.37) est reformulé en termes d'erreurs de poursuite.

A l'aide de la transformation de coordonnées, avec $(x_1, x_2)^T \in \mathfrak{R}^2$ est le vecteur d'état et $(u_v, u_w)^T \in \mathfrak{R}^2$ est le vecteur de contrôle :

$$\begin{cases} x_1 = v \\ x_2 = w \\ u_v = \tau_1 + \tau_2 \\ u_w = \tau_1 - \tau_2 \end{cases} \quad (2.45)$$

Ce qui nous donne

$$\begin{cases} \dot{x}_1 = \frac{1}{mr} u_v \\ \dot{x}_2 = \frac{L}{rI} u_w \end{cases} \quad (2.46)$$

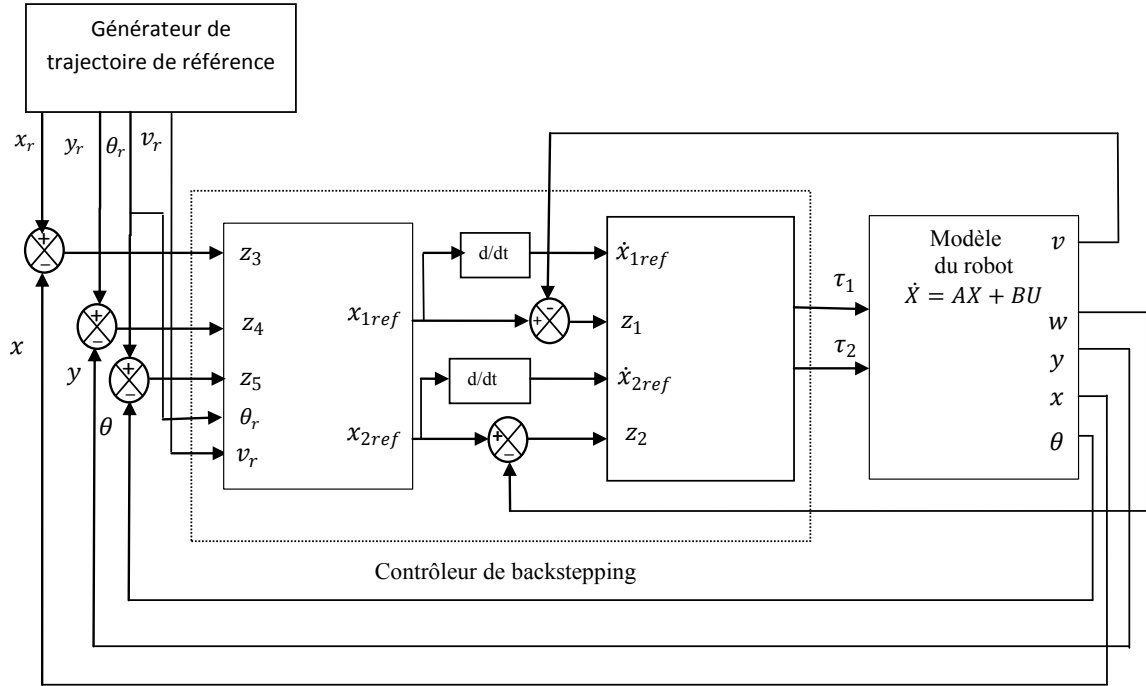


Figure (2.4) : Architecture de contrôleur du robot à base de backstepping

Pour cette approche, nous procédons en quatre étapes. L'idée de base est de choisir récursivement quelques fonctions appropriées d'état comme des entrées virtuelles de commande pour des sous systèmes du premier ordre du système global. Ce qui implique, dans chaque étape, une fonction Lyapunov étendue est associée afin de garantir la stabilité du système entier.

Etape 1:

On va commencer par définir l'erreur de suivi et sa dérivée par :

$$z_1 = x_{1ref} - x_1 \quad (2.47)$$

$$\dot{z}_1 = \dot{x}_{1ref} - \dot{x}_1 = \dot{x}_{1ref} - \frac{1}{mr} u_v \quad (2.48)$$

La première fonction de Lyapunov et sa dérivée sont déterminées comme suit :

$$V_1 = \frac{1}{2} z_1^2 \quad (2.49)$$

$$\dot{V}_1 = z_1 \dot{z}_1 = z_1 \left[\dot{x}_{1ref} - \frac{1}{mr} u_v \right] = -k_a z_1^2 \quad (2.50)$$

Pour avoir $\dot{V}_1 \leq 0$, on choisit la commande virtuelle comme :

$$u_v = mr [\dot{x}_{1ref} + k_a z_1] \quad (2.51)$$

Et on obtient :

$$\dot{V}_1 = -k_a z_1^2 \leq 0 \quad (2.52)$$

Où k_a est une constante positive, ce qui assure la négativité de la dérivée de la fonction de Lyapunov.

Étape 2 :

On définit encore une fois l'erreur de suivi et sa dérivée :

$$z_2 = x_{2ref} - x_2 \quad (2.53)$$

$$\dot{z}_2 = \dot{x}_{2ref} - \dot{x}_2 = \dot{x}_{2ref} - \frac{L}{rI} u_w \quad (2.54)$$

La nouvelle fonction de Lyapunov et sa dérivée sont :

$$V_2 = \frac{1}{2} z_1^2 + \frac{1}{2} z_2^2 \quad (2.55)$$

$$\dot{V}_2 = z_1 \dot{z}_1 + z_2 \dot{z}_2 = -k_a z_1^2 + z_2 \left[\dot{x}_{2ref} - \frac{L}{rI} u_w \right] \quad (2.56)$$

Si on choisit :

$$u_w = \frac{2rI}{L} [\dot{x}_{2ref} + k_b z_2] \quad (2.57)$$

On obtient :

$$\dot{V}_2 = -k_a z_1^2 - k_b z_2^2 \leq 0 \quad (2.58)$$

et

$$\begin{aligned} \tau_{1c} &= \frac{1}{2} [u_v + u_w] \\ \frac{1}{2} &\left[mr [\dot{x}_{1ref} + k_a z_1] + \frac{rI}{L} [\dot{x}_{2ref} + k_b z_2] \right] \\ \tau_{2c} &= \frac{1}{2} [u_v - u_w] \\ &= \frac{1}{2} \left[mr [\dot{x}_{1ref} + k_a z_1] - \frac{rI}{L} [\dot{x}_{2ref} + k_b z_2] \right] \end{aligned} \quad (2.59)$$

D'où k_b est une constante positive, afin d'assurer l'attraction de la variable de contrôle à sa valeur de référence et de satisfaire la stabilité de Lyapunov, $\dot{V}_2 < 0$.

Étape 3 :

Le changement de variables adopté est décrit par les expressions :

$$\begin{aligned} z_3 &= x_r - x \\ z_4 &= y_r - y \end{aligned} \quad (2.60)$$

La dynamique des erreurs s'exprime par :

$$\begin{aligned} \dot{z}_3 &= \dot{x}_r - \dot{x} = v_r \cos \theta_r - v \cos \theta \\ \dot{z}_4 &= \dot{y}_r - \dot{y} = v_r \sin \theta_r - v \sin \theta \end{aligned} \quad (2.61)$$

La fonction de Lyapunov est :

$$V_3 = \frac{1}{2} z_1^2 + \frac{1}{2} z_2^2 + \frac{1}{2} (z_3^2 + z_4^2) \quad (2.62)$$

Sa variation dynamique est déterminée comme suit :

$$\begin{aligned} \dot{V}_3 &= z_1 \dot{z}_1 + z_2 \dot{z}_2 + (z_3 \dot{z}_3 + z_4 \dot{z}_4) \\ &= -k_a z_1^2 + z_2 \left[\dot{x}_{2ref} - \frac{L}{r_l} u_w \right] + [(\dot{x}_r - v \cos \theta) z_3 + (\dot{y}_r - v \sin \theta) z_4] \\ &= -k_a z_1^2 + z_2 \left[\dot{x}_{2ref} - \frac{L}{r_l} u_w \right] + (v_r \cos \theta_r - v \cos \theta) z_3 + (v_r \sin \theta_r - v \sin \theta) z_4 \end{aligned} \quad (2.63)$$

Pour atteindre la négativité de \dot{V}_3 , on choisit la commande comme :

$$\begin{cases} v \cos \theta = v_r \cos \theta_r + k_c z_3 \\ v \sin \theta = v_r \sin \theta_r + k_d z_4 \end{cases} \quad (2.64)$$

avec ;

$$v_c^2 = (v_r \cos \theta_r + k_c z_3)^2 + (v_r \sin \theta_r + k_d z_4)^2 \quad (2.65)$$

Ce qui permet de déterminer

$$v_c = [v_r^2 + (k_c z_3)^2 + (k_d z_4)^2 + 2v_r(k_c z_3 \cos \theta_r + k_d z_4 \sin \theta_r)]^{1/2} \quad (2.66)$$

k_c et k_d sont des constantes positives.

Étape 4 :

On pose l'erreur :

$$z_5 = \theta_r - \theta \quad (2.67)$$

Donc ;

$$\dot{z}_5 = w_r - w \quad (2.68)$$

La fonction de Lyapunov et sa dérivée sont décrites par les deux relations suivantes :

$$V_4 = V_3 + \frac{1}{2}z_5^2 \quad (2.69)$$

$$\dot{V}_4 = \dot{V}_3 + z_5\dot{z}_5 = \dot{V}_3 + (w_r - w)z_5 \quad (2.70)$$

Il est évident de choisir

$$w_c = w_r + k_e z_5 \quad (2.71)$$

avec $k_e > 0$

2.7. Résultats de simulation

Pour montrer l'efficacité du contrôleur proposé, des simulations ont été réalisées en utilisant l'environnement Simulink-Matlab. Les trajectoires de poursuite désirées sont la trajectoire circulaire et la trajectoire carrée. Ces dernières sont données sous forme de vitesse de consignes qui seront intégrées pour générer les trajectoires à suivre.

Les paramètres du robot simulé sont donnés en annexe "B".

Pour évaluer les performances de cette approche, deux tests de robustesse sont effectués, un changement de la masse de 50% durant l'intervalle [35s 45s] et une distorsion du rayon de la roue de 10% durant l'intervalle [60s 70s]

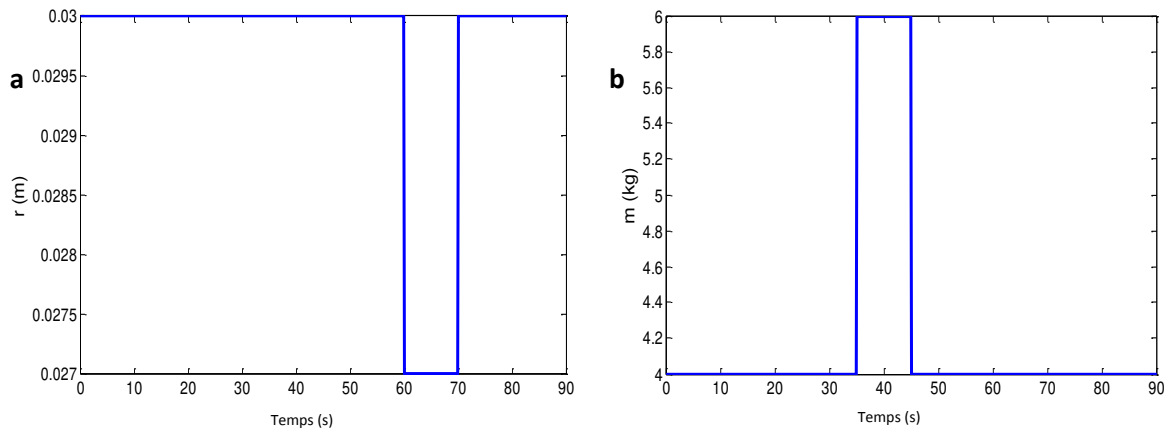


Figure (2.5) : Variation des paramètres : (a) Rayon du robot, (b) Masse du robot

a. Trajectoire circulaire

La trajectoire circulaire est générée à partir du choix des vitesses de référence suivantes :
 $v_r(t) = 0,5 \text{ m/s}$, $\omega_r(t) = 0,1 \text{ rad/s}$.

L'état initial du robot de référence est défini par :

$$q_r(0) = [x_r(0) \ y_r(0) \ \theta_r(0)]^T = [0 \ 0 \ 45^\circ]^T.$$

La position initiale réelle du robot mobile est

$$q(0) = [x(0) \ y(0) \ \theta(0)]^T = [3 \ 1 \ 180^\circ]^T.$$

Les résultats de simulation sont illustrés par les figures suivantes.

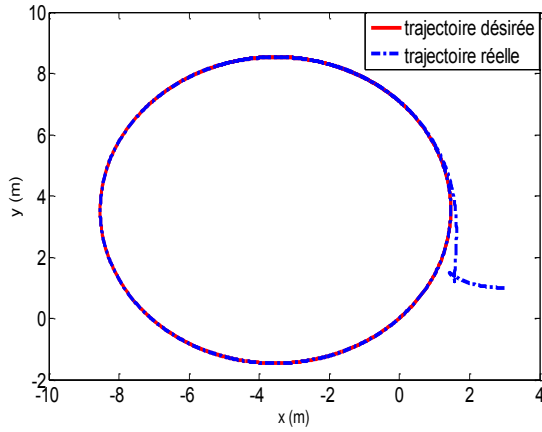


Figure (2.6) : Trajectoire circulaire

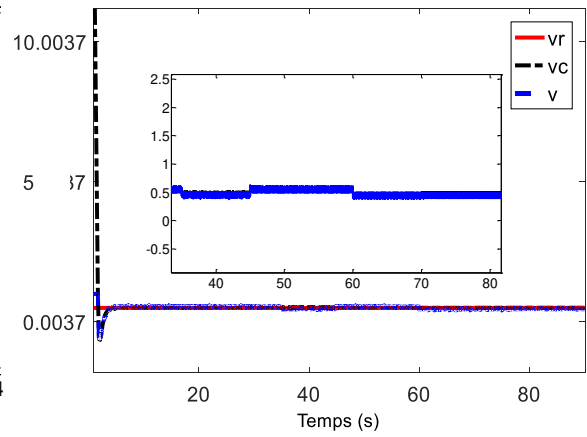


Figure (2.7) : Vitesses linéaires

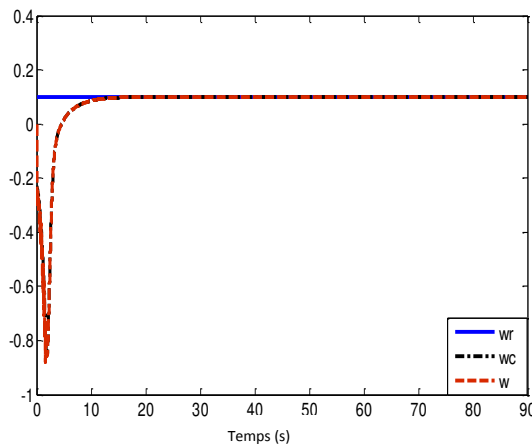


Figure (2.8) : Vitesses de rotation

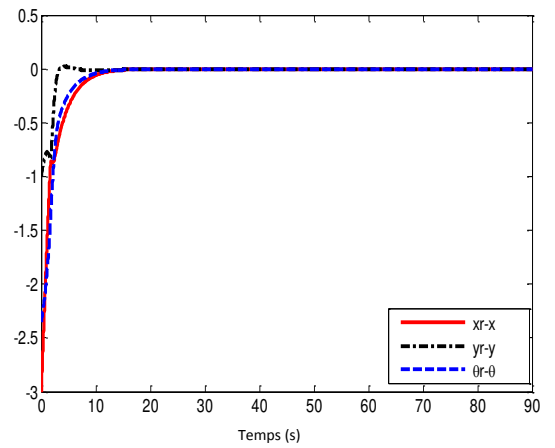


Figure (2.9) : Erreur en distance et en orientation (x_r-x , v_r-v , $\theta_r-\theta$)

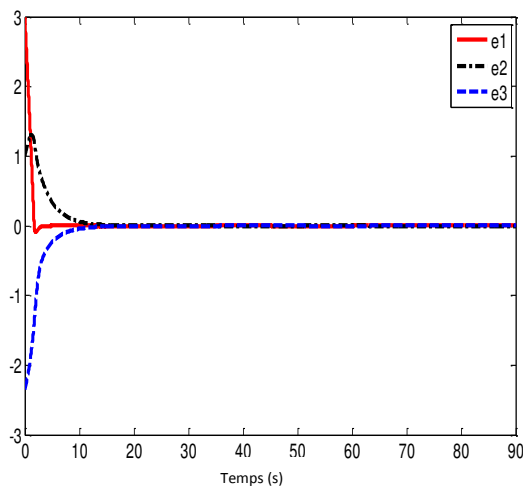


Figure (2.10) : Erreur de poursuite (e_1 , e_2 , e_3)

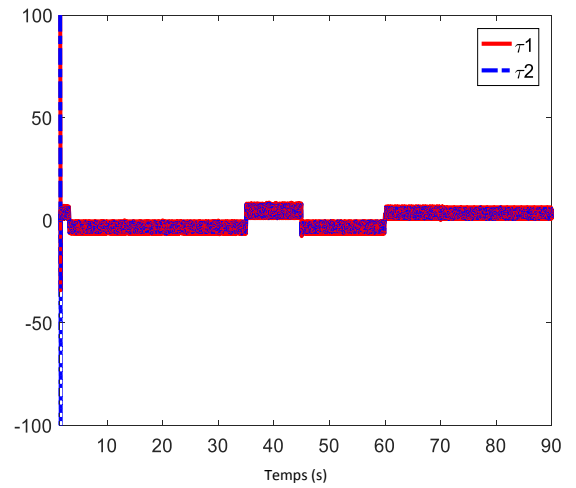


Figure (2.11) : Couple (τ_1 , τ_2) (N*m)

b. Trajectoire carrée

Dans cette section de simulation, la trajectoire de carrée est générée comme suit :

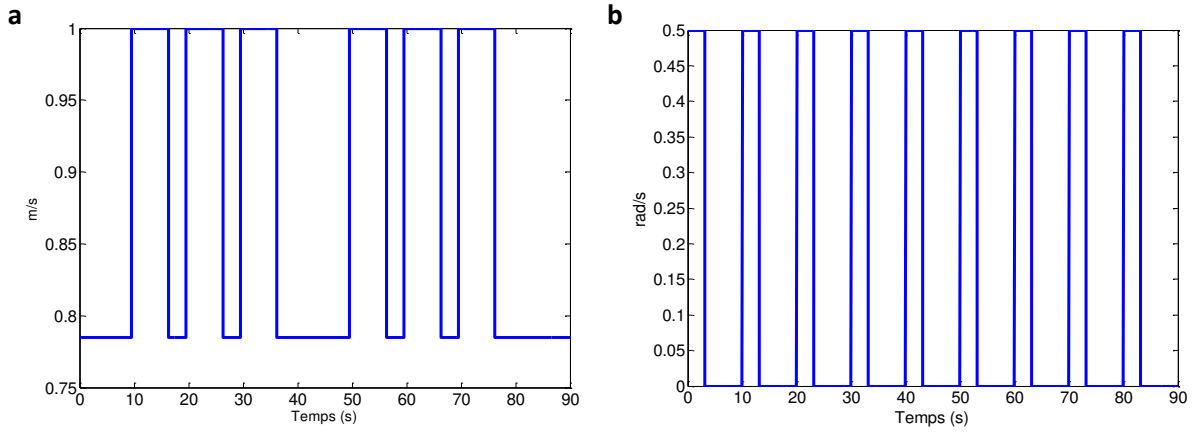


Figure (2.12) : Vitesses de référence V_r (a) Vitesse linéaire v_r , (b) Vitesse de rotation w_r

Les résultats de simulation sont présentés par les figures suivantes :

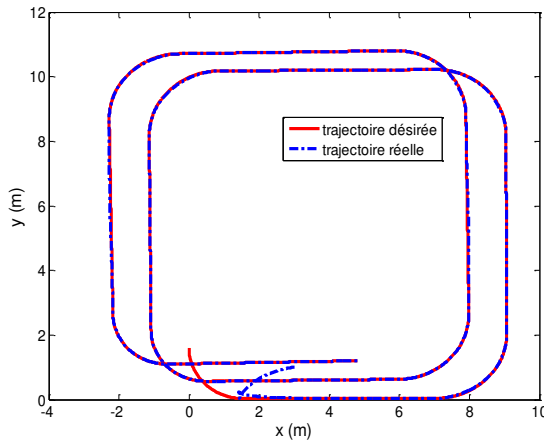


Figure (2.13) : Trajectoire carrée

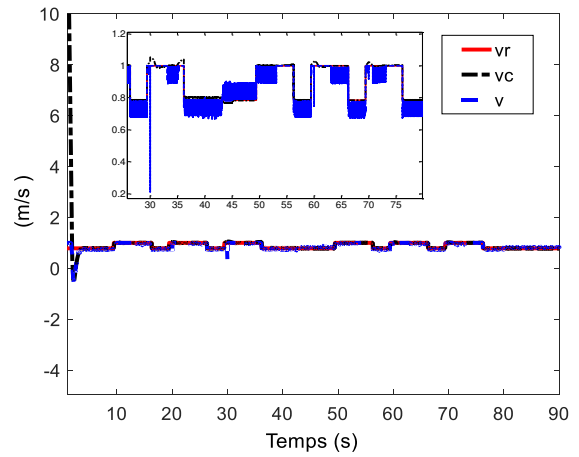


Figure (2.14) : Vitesses linéaires

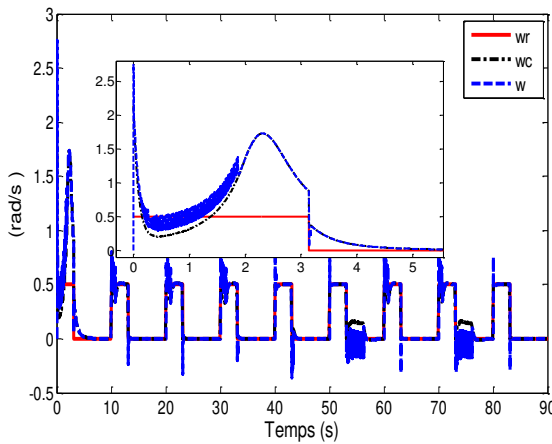


Figure (2.15) : Vitesses de rotation

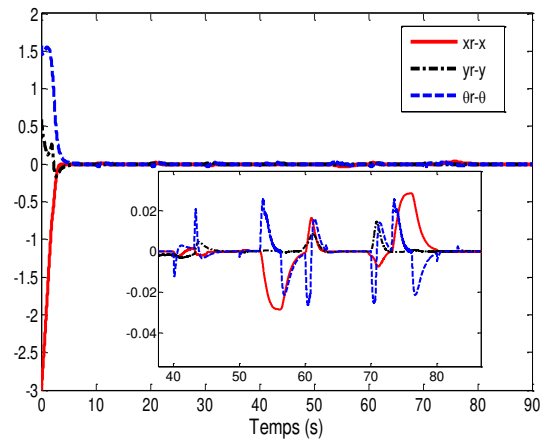
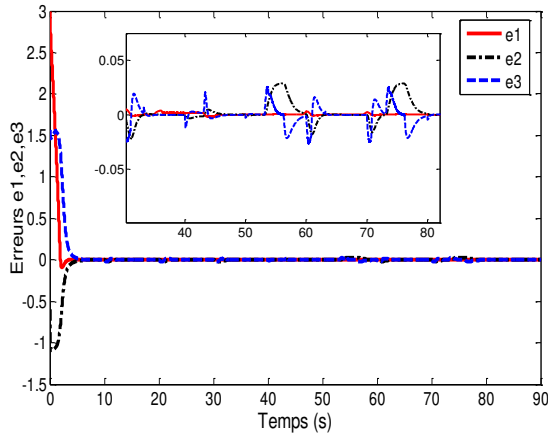
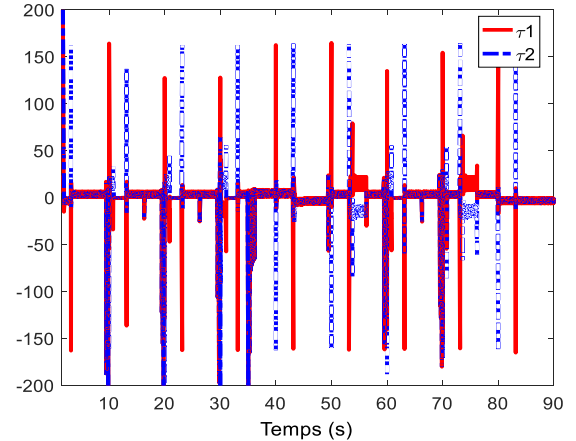


Figure (2.16) : Erreur en distance et en orientation ($x_r - x$, $y_r - y$, $\theta_r - \theta$)

Figure (2.17) : Erreur de poursuite (e_1 , e_2 , e_3)Figure (2.18) : Couple (τ_1 , τ_2) (N*m)

Les résultats de simulation sont présentés dans les figures (2.6)-(2.11) et (2.13)-(2.18). Les figures (2.6) et (2.13) indiquent que le robot suit la trajectoire de référence, en dépit des variations de la trajectoire carrée. Les figures (2.10) et (2.17) montrent que les erreurs de poursuite convergent vers zéro, les figures (2.7), (2.8), (2.14) et (2.15) montrent que les commandes de vitesses linéaires et de rotation sont lisses en cas d'une trajectoire circulaire contrairement à celles de la trajectoire carrée qui présentent des oscillations (ripples) dues à la nature de la trajectoire poursuite imposée. En analysant les différentes figures, on peut constater que les résultats obtenus attestent la robustesse de la commande backstepping. Une bonne poursuite est acquise et ce malgré la variations paramétriques.

2.8. Conclusion

Dans ce chapitre, le problème du contrôle de poursuite pour robot mobile a été traité, la conception du contrôleur est basé sur l'approche de backstepping. Les résultats numériques montrent l'efficacité des résultats théoriques proposés et la bonne performance du contrôleur en termes d'erreur de poursuite et le temps d'exécution de la trajectoire. Afin d'améliorer ces performances une autre méthode sera appliquée dans le chapitre suivant.

Chapitre 3

Commande à structure variable du robot mobile

CHAPITRE3

COMMANDE A STRUCTURE VARIABLE

DU ROBOT MOBILE

3.1. Introduction

Les contraintes non holonomes qui se posent dans les robots mobiles à roues, les incertitudes des paramètres et les perturbations présentent des difficultés et des problèmes dans la poursuite de trajectoire des robots mobiles, ce qui limite l'espace d'application de ces derniers. Pour résoudre ce problème, plusieurs méthodes ont été développées telle que la commande à structure variable (VSS : Variable Structure System).

Dans le présent chapitre, nous allons aborder les concepts de base de la commande par mode glissant (CMG), son principe, ses propriétés de robustesse ainsi que les étapes de conception, ensuite un contrôleur non linéaire basé sur la théorie de Lyapunov associée à la technique par mode glissant sera développé pour la poursuite de la trajectoire du robot mobile.

3.2. Bref historique

La théorie des systèmes à structure variable, c'est à dire les systèmes à commande commutante entre plusieurs lois de commande différentes (en général linéaires) a fait l'objet de multiples études depuis une cinquantaine d'années [65-66]. Les premiers travaux sur ce type de systèmes sont ceux d'Anosov, de Tzypkin et d'Emel'yanov dans l'ancienne URSS, ou ceux d'Hamel en France, sur la commande à relais [65-66]. La théorie du mode glissant est l'extension de l'étude des systèmes à structure variable, la recherche d'Utkin est l'une des premières recherches qui ont traitées et élaborées des commandes basées sur cette théorie. Par la suite, ces travaux ont été repris, soit pour compléter l'étude théorique, soit pour étudier quelques applications. Aux Etats-Unis, cette théorie est introduite par Slotine et au Japon par Young, Harasma et Hashimoto. Il a fallu attendre les années 80, pour que le concept de cette

théorie trouve ses applications dans plusieurs domaines notamment dans les systèmes de commande et serait parmi les techniques de contrôle non linéaire les plus efficaces et les plus robustes [62-63].

Le principe de cette technique est de forcer le système à atteindre et ensuite rester sur une surface donnée (représentant un ensemble de relations statiques entre les variables d'état), la surface considérée est alors désignée comme étant la surface de glissement ou de commutation. Le comportement dynamique résultant, appelé régime glissant idéal, est complètement déterminé par les paramètres et les équations définissant la surface [66]. Les trajectoires de système sont alors confondues avec la surface de glissement durant l'utilisation des commandes à une grande vitesse de commutation. Cette commande a deux principaux avantages, premièrement, le comportement dynamique du système au voisinage de la surface de glissement est essentiellement conditionné par cette dernière, on dit que le système est en régime glissant, deuxièmement, ce comportement est non seulement robuste, mais aussi insensible aux incertitudes pour lesquelles la commande a été réglée [65].

Une fois, le régime glissant établi, la discontinuité de la commande engendre des oscillations à hautes fréquences dans un voisinage de la surface (dont l'épaisseur est de l'ordre de la période d'échantillonnage ou de commutation) [65-66]. Ce phénomène est connu sous le nom de broutement ou "chattering en anglais" et est un des inconvénients majeurs de cette technique, car il peut en outre exciter des dynamiques non modélisées et conduire à l'instabilité [65-66].

3.3. Principe de la commande par mode de glissement

Le contrôle par mode glissant consiste à amener la trajectoire d'état d'un système vers la surface de glissement et de la faire commuter à l'aide d'une logique de commutation autour de celle-ci jusqu'au point d'équilibre, d'où le phénomène de glissement [66]. Parmi les propriétés des modes glissants :

- Le processus de glissement est d'ordre réduit en comparaison au système original.
- La dynamique du système en mode de glissement est déterminée uniquement par le choix des coefficients de la surface de glissement.
- La robustesse vis-à-vis de la variation de certains types de paramètres.
- La trajectoire dans le plan de phase est constituée de trois parties distinctes, figure (3.1).

- Le Mode de convergence « MC » : C’est le mode durant lequel la variable à régler se déplace à partir de n’importe quel point initial dans le plan de phase et tend vers la surface de commutation $s(x, y) = 0$. Ce mode est caractérisé par la loi de commande et le critère de convergence.
- Le Mode de glissement « MG » : C’est le mode durant lequel la variable d’état a atteint la surface de glissement et tend vers l’origine du plan de phase. La dynamique de ce mode est caractérisée par le choix de la surface de glissement $s(x, y) = 0$.
- Le Mode du régime permanent « MRP » : Ce mode est ajouté pour l’étude de la réponse du système autour de son point d’équilibre (origine du plan de phase), il est caractérisé par la qualité et les performances de la commande [64].

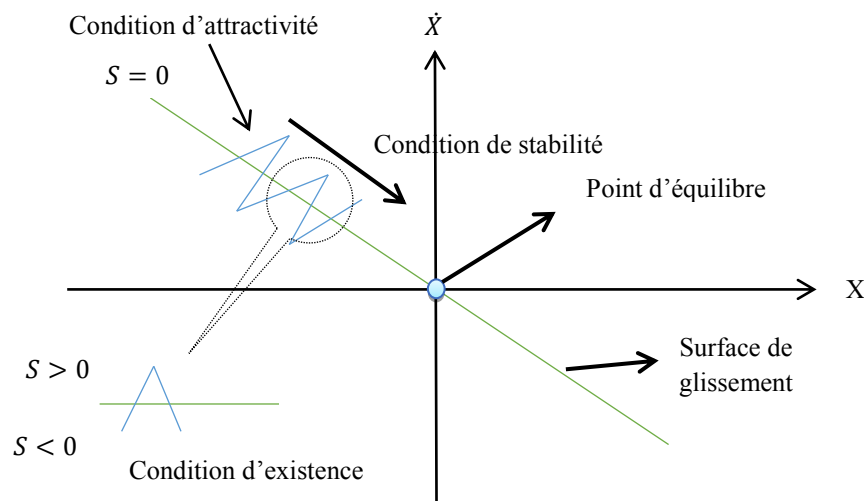


Figure (3.1) : Différents modes pour la trajectoire dans le plan de phase

Le comportement des systèmes possédant des discontinuités peut être décrit formellement par l'équation :

$$\dot{X} = f(X, t, U) \tag{3.1}$$

Où U : représente à la fois la discontinuité et la commande.

Historiquement, les premiers régulateurs bâtis sur ce modèle ont été de simples relais, faciles à mettre en œuvre, ils ont représenté les prémices du contrôle par structure variable. Ils ont ainsi amené les automaticiens à développer une théorie qui puisse décrire un tel fonctionnement. Les bases d'une telle théorie ont été posées : il suffit de dire que le comportement du système est décrit par deux équations différentielles distinctes, suivant que l'équation d'évolution du système soit supérieure ou inférieure à une surface dénommée hyper surface de commutation où

$$S(X, t) = (S_1 X \dots S_m X) \tag{3.2}$$

On a alors :

$$U(X) = \begin{cases} U^+(X) & \text{si } S(X) > 0 \\ U^-(X) & \text{si } S(X) < 0 \end{cases} \tag{3.3}$$

En se plaçant dans un espace à deux dimensions, le principe de discontinuité de la commande peut être illustré par la figure suivante [66].

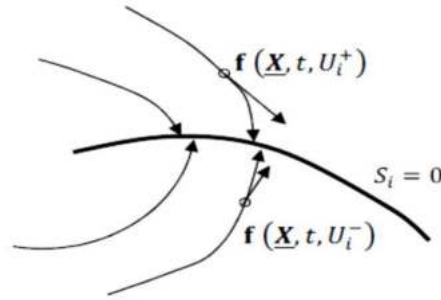


Figure (3.2) : Convergence de la trajectoire vers la surface de commutation grâce à la discontinuité de la de phase

3.4. Conception de l’algorithme de commande par mode de glissement

La conception de l’algorithme de commande par mode de glissement prend en compte les problèmes de stabilité et de bonnes performances de façon systématique dans son approche, qui s’effectue principalement en trois étapes complémentaires définies par [65-66] :

- Choix des surfaces de glissement ;
- Définition des conditions d’existence et de convergence du régime glissant ;
- Détermination de la loi de commande.

3.4.1. Choix de la surface de glissement

Le choix de la surface de glissement concerne non seulement le nombre nécessaire de ces surfaces, mais également leurs formes en fonction de l’application et de l’objectif visé. Généralement, le nombre des surfaces de glissement est choisi égal à la dimension du vecteur de commande [62]. Le procédé le plus judicieux et le plus simple consiste à choisir une surface de commutation égale au vecteur d’erreur [67], du vecteur d’état :

$$S = \Delta X = X_{ref} - X \tag{3.4}$$

La surface de glissement est une fonction scalaire telle que la variable à régler glisse sur cette surface et tend vers l’origine du plan de phase. Plusieurs formes de la surface de

glissement ont été proposées dans la littérature, chacune présente des meilleures performances pour une application donnée [62].

La surface la plus utilisée pour obtenir le régime de glissement qui garantit la convergence de l'état vers sa référence « donné par Slotine », est définie par :

$$S(X) = \left(\frac{d}{dt} + \lambda \right)^{n-1} (X_{ref} - X) \quad (3.5)$$

où :

n : est le degré relatif au nombre qu'il faut dériver la sortie pour faire apparaître la commande;

λ : est une constante positive.

L'objectif de cette commande est de garder la surface qui tend à zéro. Cette dernière est une équation différentielle linéaire dont l'unique solution est $S(X) = 0$, pour un choix convenable du paramètre λ . Ceci revient à un problème de poursuite de trajectoire, ce qui est équivalent à une linéarisation exacte de l'écart, tout en respectant la condition de convergence.

3.4.2. Conditions de convergence et d'existence

Les conditions d'existence et de convergence sont les critères qui permettent aux dynamiques du système de converger vers la surface de glissement et d'y rester indépendamment de la perturbation. Pour cela nous retenons deux conditions [67] :

Fonction directe de commutation : C'est l'approche la plus ancienne et directe, a été proposée et étudiée par Emilyanov et Utkin. Elle est globale mais ne garantit pas, en revanche un temps d'accès fini. Elle est exprimée par :

$$S(X)\dot{S}(X) < 0 \quad (3.6)$$

Dans cette condition, il faut introduire pour $S(X)$ et sa dérivée, les valeurs justes à gauche et à droite de commutation.

Fonction de Lyapunov : L'approche de Lyapunov est une condition globale d'accès au mode glissant. Il s'agit de formuler une fonction scalaire positive $V(X) > 0$ pour les variables d'état du système, et de choisir la loi de commutation qui fera décroître cette fonction c'est à dire $\dot{V}(X) < 0$. Cette fonction est généralement utilisée pour garantir la stabilité des systèmes non linéaires [62-63] et [67].

En définissant la fonction de Lyapunov par :

$$V(X) = \frac{1}{2} S^2(X) \quad (3.7)$$

L'idée est de choisir cette fonction pour garantir l'attraction de la variable à contrôler vers sa valeur de référence, et de concevoir une commande telle que le carré de la surface correspond à une fonction de Lyapunov. La dérivée de la fonction $V(X)$ donne :

$$\dot{V}(X) = S(X)\dot{S}(X) \quad (3.8)$$

Pour que la fonction candidate de Lyapunov puisse décroître et converger vers zéro « pour garantir l'existence du mode de glissant », il suffit d'assurer que : $S(X)\dot{S}(X) < 0$.

Cette équation montre que le carré de la distance vers la surface, mesurée par $S^2(X)$, diminue tout le temps, contraignant la trajectoire du système à se diriger vers la surface des deux côtés [67].

3.4.3. Détermination de la loi de commande

La structure d'un contrôleur en mode glissant comporte deux parties : La première concerne la linéarisation exacte (U_{eq}) et la deuxième est stabilisante (U_n). Cette dernière est très importante dans le réglage par mode glissant, elle permet d'éliminer les effets d'imprécisions du modèle et de rejeter les perturbations extérieures [65-67].

$$U = U_{eq} + U_n \quad (3.9)$$

U_{eq} Correspond à la commande proposée par Filipov, elle sert à maintenir la variable à contrôler sur la surface de glissement $s(X) = 0$. La commande équivalente est déduite, en considérant que la dérivée de la surface est nulle $\dot{S}(X) = 0$.

Elle peut être interprétée comme étant un retour d'état particulier jouant le rôle d'un signal de commande appliqué sur le système à commander. Elle peut être aussi interprétée autrement comme étant une valeur moyenne que prend la commande lors de la commutation rapide entre les valeurs U_{max} et U_{min} , figure (3.3). La commande discrète U_n est déterminée pour vérifier la condition de convergence en dépit de l'imprécision sur les paramètres du modèle du système.

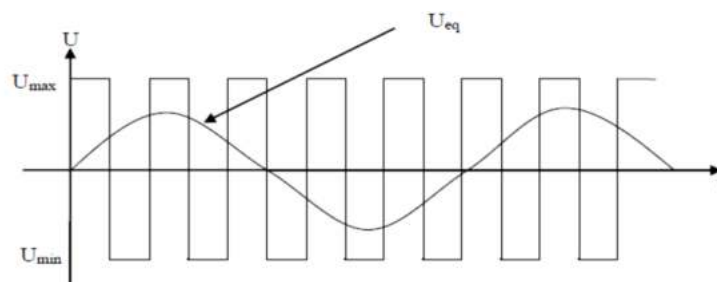


Figure (3.3) : Interprétation de U_{eq}

Cette commande présente les caractéristiques suivantes :

- Elle est robuste, rejetant les perturbations ;
- Le choix de la surface de commutation est assez libre ;
- La commande est adoucie par la présence de la commande équivalente.

3.5. Application à la commande du robot mobile

Le robot mobile à roues peut être décrit par le modèle cinématique et le modèle dynamique ce qui donnent la possibilité de classer les problèmes de recherche en deux axes, les problèmes de contrôle de poursuite comme des problèmes cinématiques ou dynamiques [61]. Mais la difficulté réside dans le non intégrabilité des contraintes non holonomes, ceci indique que le problème est non linéaire, ce qui rend le problème du contrôle de poursuite du robot mobile difficile à gérer [68]. Il est donc nécessaire de concevoir un dispositif de commande qui prend en compte les deux modèles simultanément [68-69].

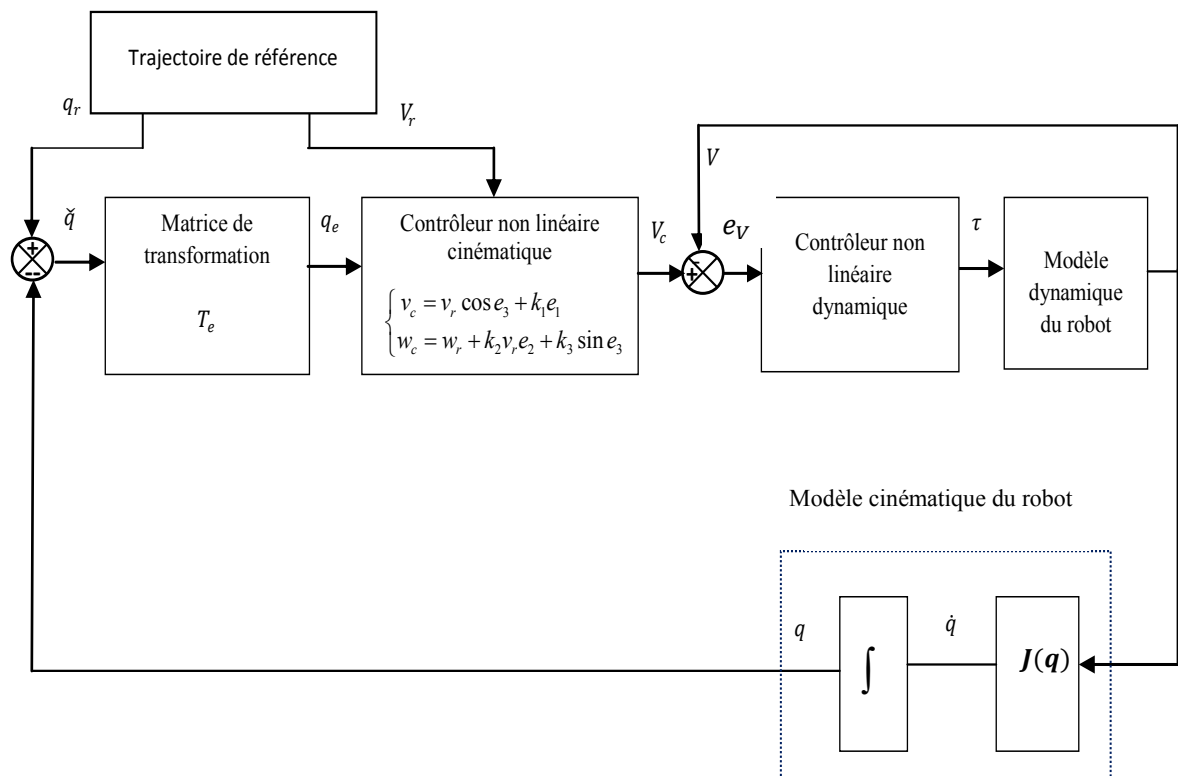


Figure (3.4) : Architecture de contrôleur du robot.

Dans cette conception, l'unité de contrôle cinématique a pour rôle de produire des sorties de vitesse pour le robot qui permet de converger l'erreur entre la trajectoire réelle et la trajectoire de référence à zéro. Ensuite, la technique basée sur la théorie de Lyapunov associée au mode glissant est utilisée pour la conception de la commande du couple en

fonction de la dynamique du système et de rendre la vitesse du robot mobile converge vers la vitesse désirée produite [26] et [69-70].

Le contrôleur cinématique reçoit comme entrées les vecteurs V_r et q_e , ce contrôleur représente la commande niveau haut du robot mobile, son rôle est de générer le vecteur des vitesses de consigne servant d'entées pour le contrôleur dynamique. Le contrôleur dynamique constitue le niveau bas de commande, il est responsable de la génération des couples à appliquer aux roues du robot pour le guider à poursuivre la trajectoire désirée, figure (3.4).

3.5.1. Contrôleur cinématique

La loi de commande au niveau cinématique est basée sur la méthode de poursuite stable fréquemment utilisée et mentionnée dans la littérature traitant le problème de poursuite de trajectoire [61] qui sera détaillée dans cette section.

La relation suivante entre les variables mesurées (x, y, θ) et les variables décrivant la trajectoire désirée (x_r, y_r, θ_r) , représente le retour d'état appliqué sur le système réel, le but du contrôleur est d'élaborer une loi de contrôle (v_r, w_c) qui permet d'annuler la posture d'erreur $(q_e \rightarrow 0)$ [59-60] et [71].

Le vecteur d'erreur q_e , est défini comme étant (voir l'équation (2.41)) :

$$q_e = \begin{bmatrix} e_1 \\ e_2 \\ e_3 \end{bmatrix} = \begin{bmatrix} \cos\theta & \sin\theta & 0 \\ -\sin\theta & \cos\theta & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x_r - x \\ y_r - y \\ \theta_r - \theta \end{bmatrix} = T_e \tilde{q}$$

Les entrées (v_c, w_c) , qui font convertir les erreurs e_1, e_2 et e_3 à zéro, sont donnés par la relation suivante :

$$\begin{cases} v_c = v_r \cos e_3 + k_1 e_1 \\ w_c = w_r + k_2 v_r e_2 + k_3 \sin e_3 \end{cases} \quad (3.10)$$

Démonstration :

La fonction candidate de Lyapunov est définie comme :

$$V_0 = \frac{1}{2}(e_1^2 + e_2^2) + \frac{1 - \cos e_3}{k_2} > 0 \quad (3.11)$$

La dérivée de l'équation précédente donne ce qui suit :

$$\dot{V}_0 = \dot{e}_1 e_1 + \dot{e}_2 e_2 + \dot{e}_3 \frac{\sin e_3}{k_2} \quad (3.12)$$

où ;

$$\begin{cases} \dot{x} = v \cos \theta \\ \dot{y} = v \sin \theta \\ \dot{\theta} = w \end{cases} \text{ et } \begin{cases} \dot{x}_r = v_r \cos \theta_r \\ \dot{y}_r = v_r \sin \theta_r \\ \dot{\theta}_r = w_r \end{cases}$$

Ce qui permet d'écrire :

$$\begin{aligned} \dot{e}_1 &= (\dot{x}_r - \dot{x}) \cos \theta - \dot{\theta} (x_r - x) \sin \theta + (\dot{y}_r - \dot{y}) \sin \theta + \dot{\theta} (y_r - y) \cos \theta \\ &= \dot{x}_r \cos \theta - \dot{x} \cos \theta + \dot{y}_r \sin \theta - \dot{y} \sin \theta + w e_2 \dot{e}_1 = v_r \cos(\theta_r - \theta) - v + w e_2 \end{aligned} \quad (3.13)$$

$$\begin{aligned} \dot{e}_2 &= -(\dot{x}_r - \dot{x}) \sin \theta - \dot{\theta} (x_r - x) \cos \theta + (\dot{y}_r - \dot{y}) \cos \theta - \dot{\theta} (y_r - y) \sin \theta \\ &= -\dot{x}_r \sin \theta + \dot{x} \sin \theta + \dot{y}_r \cos \theta - \dot{y} \cos \theta - w e_1 \dot{e}_2 = v_r \sin(\theta_r - \theta) - w e_1 \end{aligned} \quad (3.14)$$

$$\dot{e}_3 = \dot{\theta}_r - \dot{\theta} = w_r - w \quad (3.15)$$

donc;

$$\dot{q}_e = \begin{bmatrix} \dot{e}_1 \\ \dot{e}_2 \\ \dot{e}_3 \end{bmatrix} = \begin{bmatrix} -1 \\ 0 \\ 0 \end{bmatrix} v + \begin{bmatrix} e_2 \\ -e_1 \\ -1 \end{bmatrix} w + \begin{bmatrix} v_r \cos e_3 \\ v_r \sin e_3 \\ w_r \end{bmatrix} \quad (3.16)$$

Par substitution dans l'équation (3.12), on obtient :

$$\dot{V}_0 = (-v + v_r \cos e_3) e_1 + (v_r k_2 e_2 + \frac{w_r}{k_2} - \frac{w}{k_2}) \frac{\sin e_3}{k_2} \quad (3.17)$$

La proposition de l'équation (3.10) donne :

$$\dot{V}_0 = -k_1 e_1^2 - \frac{k_3 \sin^2 e_3}{k_2} \quad (3.18)$$

Pour satisfaire la condition de stabilité, $\dot{V}_0 < 0$, les paramètres k_1 , k_2 et k_3 devraient être positifs.

3.5.2. Contrôleur dynamique

Cette étape consiste à concevoir un dispositif de commande de poursuite dynamique basé sur la théorie de Lyapunov associé au mode glissant qui permet aux vitesses réelles du robot de converger vers les vitesses de commande générées par le dispositif de commande cinématique.

Considérant la surface de glissement particulière suivante :

$$s(t) = k e + \dot{e}, \quad k > 0 \quad (3.19)$$

où;

$$e_v = \begin{bmatrix} e_v \\ e_w \end{bmatrix} = \begin{bmatrix} v_c - v_r \\ w_c - w_r \end{bmatrix}, \dot{e} = \begin{bmatrix} \dot{e}_v \\ \dot{e}_w \end{bmatrix} = \begin{bmatrix} \dot{v}_c - \dot{v}_r \\ \dot{w}_c - \dot{w}_r \end{bmatrix} \quad (3.20)$$

L'erreur de poursuite de vitesse et ses dérivées sont définies dans l'équation précédente, pour concevoir l'entrée de couple et d'estimer les vitesses souhaitées V_c .

Il est souhaitable de conduire le système en boucle fermée vers les surfaces de glissement $\dot{s} = 0$, lorsque le système est sur cette surface de glissement, l'erreur de poursuite $e(\infty) \rightarrow 0$.

Proposition :

Pour assurer la stabilité du système et vérifier l'inégalité $s^T \dot{s} < 0$, nous proposons comme entrée pour le contrôleur dynamique la commande :

$$\begin{cases} \tau_1 = \frac{1}{E_{11}} (-E_{12}\tau_2 + \dot{v}_r + k_a e_v) \\ \tau_2 = \frac{1}{E_{22}} (-E_{21}\tau_1 + \dot{w}_r + k_b e_w) \end{cases} \quad (3.21)$$

et

$$\dot{V}(t) = E\tau(t) + \Delta f(t) \quad (3.22)$$

où $\Delta f(t)$ présente les paramètres d'incertitudes et de perturbations externes.

Preuve :

Soit la fonction candidate de Lyapunov

$$V_1 = \frac{1}{2} (e_v^2 + e_w^2) > 0 \quad (3.23)$$

Telle que sa dérivée est :

$$\dot{V}_1 = \dot{e}_v e_v + \dot{e}_w e_w = (\dot{v}_c - \dot{v}_r) e_v + (\dot{w}_c - \dot{w}_r) e_w \quad (3.24)$$

La substitution de (1.21) dans l'équation (3.24) donne le résultat suivant :

$$\dot{V}_1 = (E_{11}\tau_1 + E_{12}\tau_2 - \dot{v}_r) e_v + (E_{21}\tau_1 + E_{22}\tau_2 - \dot{w}_r) e_w \quad (3.25)$$

Compte tenu de la proposition de l'équation (3.21), la condition de la stabilité de Lyapunov est vérifiée, et nous obtenons :

$$\dot{V}_1 = k_a e_v^2 + k_b e_w^2 < 0 \quad (3.26)$$

avec; k_a et k_b sont des constantes négatives, $k_a = -100$ et $k_b = -1000$.

à partir de l'équation (3.22) :

$$\begin{cases} \dot{v}_c = E_{11}\tau_1 + E_{12}\tau_2 + \Delta f_v \\ \dot{w}_c = E_{21}\tau_1 + E_{22}\tau_2 + \Delta f_w \end{cases} \quad (3.27)$$

La nouvelle dérivée de Lyapunov est :

$$\begin{aligned} \dot{V}_2 &= (\dot{v}_c - \dot{v}_r)e_v + (\dot{w}_c - \dot{w}_r)e_w \\ &= (E_{11}\tau_1 + E_{12}\tau_2 + \Delta f_v - \dot{v}_r)e_v + (E_{21}\tau_1 + E_{22}\tau_2 + \Delta f_w - \dot{w}_r)e_w \end{aligned} \quad (3.28)$$

Si nous mettons :

$$\begin{cases} \tau_1 = \frac{1}{E_{11}}(-E_{12}\tau_2 + \dot{v}_r + k_a e_v + k_c \operatorname{sgn}(e_v)) \\ \tau_2 = \frac{1}{E_{22}}(-E_{21}\tau_1 + \dot{w}_r + k_b e_w + k_d \operatorname{sgn}(e_w)) \end{cases} \quad (3.29)$$

Nous trouvons :

$$\dot{V}_2 = (k_a e_v^2 + k_b e_w^2) + (k_c \operatorname{sgn}(e_v) + \Delta f_v)e_v + (k_d \operatorname{sgn}(e_w) + \Delta f_w)e_w \quad (3.30)$$

La substitution de l'équation (3.26) dans l'équation (3.30) donne :

$$\dot{V}_2 = \dot{V}_1 + (k_c \operatorname{sgn}(e_v) + \Delta f_v)e_v + (k_d \operatorname{sgn}(e_w) + \Delta f_w)e_w \quad (3.31)$$

Donc pour assurer $\dot{V}_2 \leq 0$, il est nécessaire que :

$$\begin{cases} (k_c \operatorname{sgn}(e_v) + \Delta f_v)e_v \leq 0, \quad \forall e_v \\ (k_d \operatorname{sgn}(e_w) + \Delta f_w)e_w \leq 0, \quad \forall e_w \end{cases} \quad (3.32)$$

avec ; $\begin{cases} k_c > |\Delta f_v| \\ k_d > |\Delta f_w| \end{cases}$, où $k_c = -10$ et $k_d = -0,5$.

3.6. Résultats de simulation

Dans le but d'aboutir à une étude comparative et de mettre en exigence l'apport de ce type de contrôleur "par mode glissant", les mêmes conditions de fonctionnement (chapitre deux) (trajectoires, variations paramétriques, etc.) sont utilisés.

a. Trajectoire circulaire

Les résultats de simulation sont donnés par les figures suivantes :

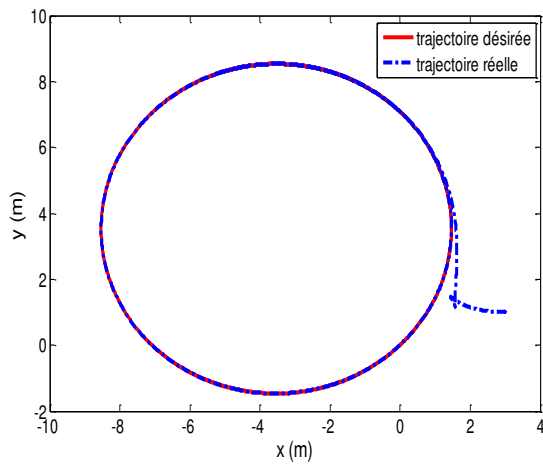


Figure (3.5) : Trajectoire circulaire

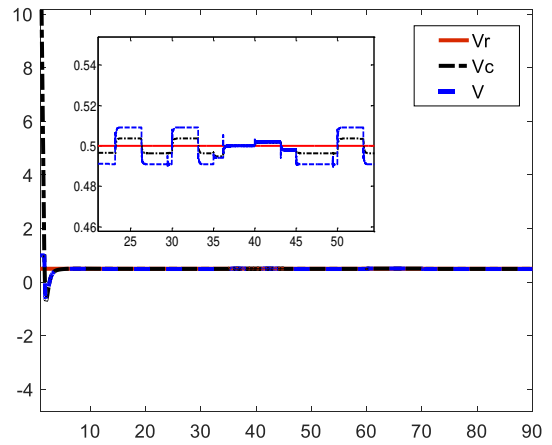


Figure (3.6) : Vitesses linéaires

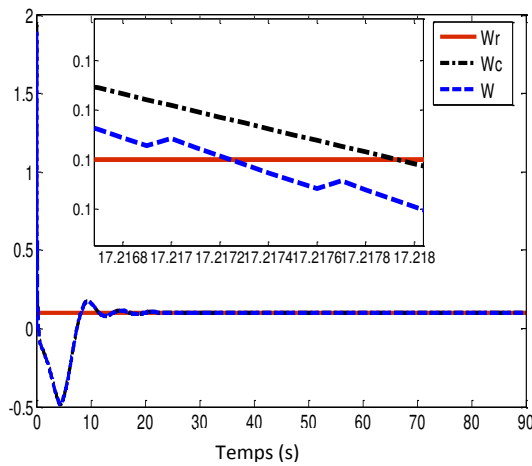


Figure (3.7) : Vitesses de rotation

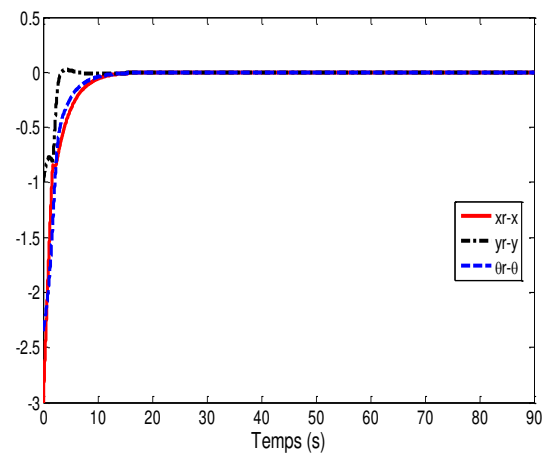


Figure (3.8) : Erreur en distance et en orientation (x_r-x , y_r-y , $\theta_r-\theta$)

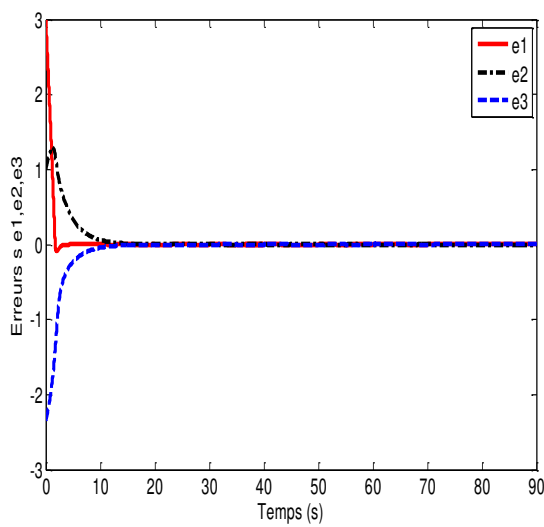


Figure (3.9) : Erreur de poursuite (e_1 , e_2 , e_3)

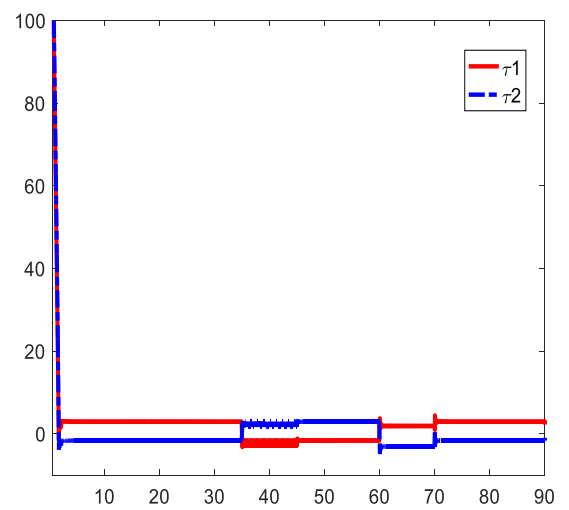


Figure (3.10) : Couple (τ_1 , τ_2) (N*m)

b. Trajectoire carrée

Les résultats de simulation obtenus sont illustrés par les figures suivantes :

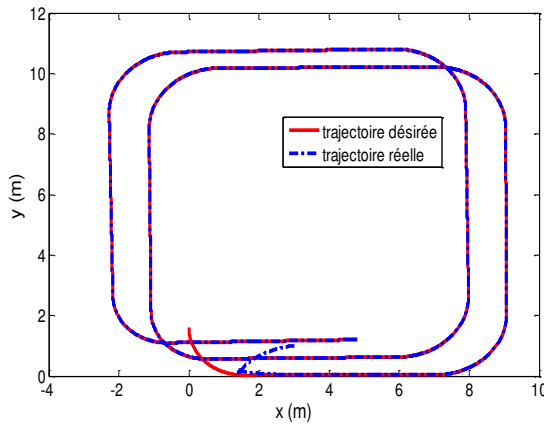


Figure (3.11) : Trajectoire carrée

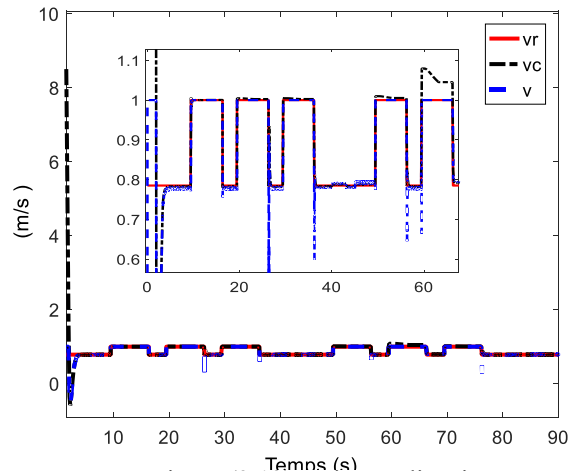


Figure (3.12) : Vitesses linéaires

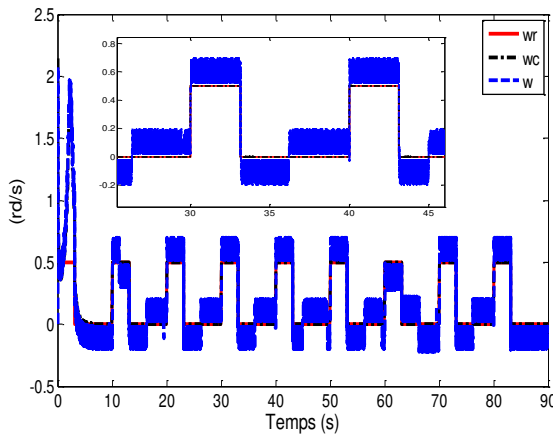


Figure (3.13) : Vitesses de rotation

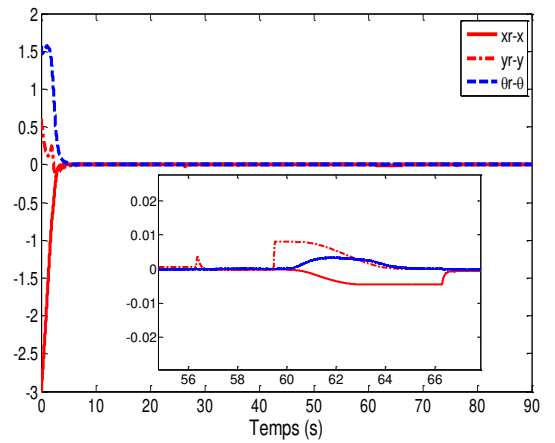


Figure (3.14) : Erreur en distance et en orientation (x_r-x , y_r-y , $\theta_r-\theta$)

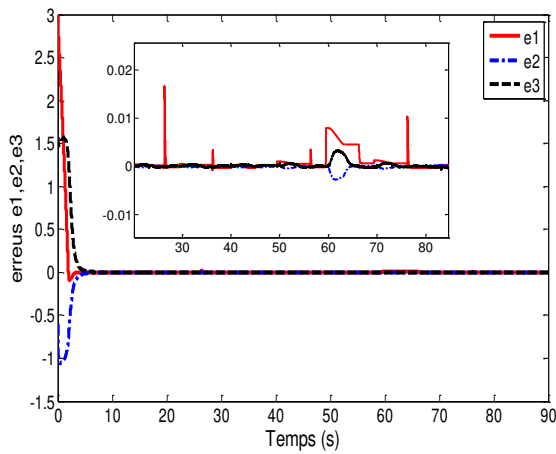


Figure (3.15) : Erreur de poursuite (e_1 , e_2 , e_3)

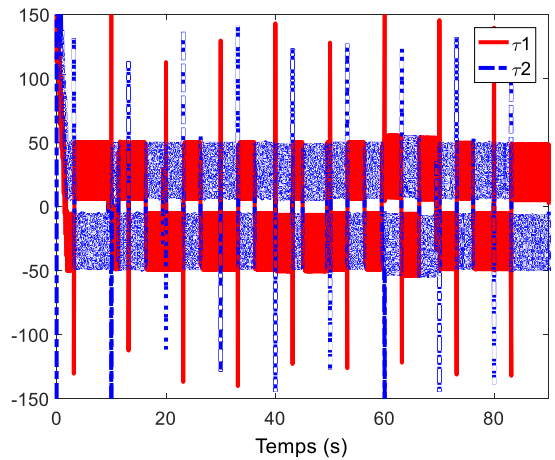


Figure (3.16) : Couple (τ_1 , τ_2) (N*m)

L'analyse des résultats de simulation présentés par les figures (3.5)-(3.10) et (3.11)-(3.16) atteste la bonne poursuite du robot des différentes trajectoires (carrée et circulaire). Le constat de l'évolution des erreurs cartésiennes et de rotation, figures (3.8) et (3.14), montre la convergence des grandeurs contrôlées vers un régime permanent stable. On note la présence du phénomène de "chattering" dû à la nature du contrôleur à structure variable (mode glissant).

3.7. Conclusion

Dans ce chapitre un contrôleur non linéaire basé sur la théorie de Lyapunov associée au mode glissant est proposé pour résoudre le problème de commande de mouvement de robot mobile non holonome à roues avec changement des paramètres (m et r). Les résultats de simulation montrent l'efficacité du contrôleur proposé. Le problème qui se pose est le phénomène de chattering, dans le chapitre suivant, la logique floue sera exploitée afin de minimiser ce phénomène.-

Chapitre 4

Apport de la logique floue à la la commande du robot mobile

CHAPITRE 4

APPORT DE LA LOGIQUE FLOUE A LA COMMANDE DU ROBOT MOBILE

4.1. Introduction

La plupart des techniques de commande reposent sur l'hypothèse que le modèle du système à contrôler ou à réguler est suffisamment bien connu pour y appliquer les résultats théoriques de la régulation, c'est évident qu'il s'agit là d'une vision excessivement optimiste des problèmes industriels, c'est pourquoi les automaticiens ont cherché des moyens pour contourner cette difficulté [72]. C'est dans ce cadre qu'une place privilégiée est dévolue à la commande floue, cette dernière donne une approche plutôt pragmatique permettant d'inclure aussi les expériences acquises par les opérateurs du processus à commander [72-75].

La logique floue diffère de la logique classique parce qu'elle permet des définitions partielles ou (floues) de règles de contrôle, elle demande une bonne connaissance du formalisme mathématique de la logique floue et aussi une bonne intuition et une bonne connaissance du système à régler [74-75]. L'intérêt de la commande floue (bien que cette expression soit abusive puisque la commande est parfaitement définie) est d'introduire l'expertise de l'opérateur dans la commande du process [75].

La stratégie du contrôleur flou en boucle fermée basée sur la connaissance de l'opérateur du processus à contrôler est d'écrire ce phénomène de façon linguistique, puis de représenter le par un nombre de règles, le contrôleur ainsi envisagé nécessite un algorithme permettant la conversion de la stratégie de contrôle linguistique en stratégie de contrôle automatique [74-75] et [77]. Les connaissances dans un système flou sont contenues dans les règles et dans les ensembles flous, qui contiennent des descriptions générales des propriétés du phénomène en question [74-75].

Un domaine d'application de la logique floue qui devient fréquent est celui de la commande des robots mobiles à roues. Ce chapitre présente les fondements de la logique floue en vue de l'appliquer par la suite pour un auto-réglage des gains du contrôleur cinématique du robot mobile. On présentera les résultats de simulation des différents tests qui ont été effectués dans le but de mettre en relief l'efficacité de l'approche proposée.

4.2. Bref historique

Les racines de la logique floue se trouvent dans le principe d'incertitude de Heisenberg. Dans les années 20, les physiciens ont introduit la troisième valeur $\frac{1}{2}$ dans le système logique bivalent $[0,1]$, Pourquoi $\frac{1}{2}$? Parce que c'est la valeur de vérité de tous les paradoxes et la logique classique interdit tous les paradoxes par ses axiomes [78].

En 1965 Lotfi A. Zadeh, Professeur à l'université de Californie à Berkeley, automaticien de réputation internationale a contribué à la modélisation des phénomènes sous forme floue à travers la publication de son article intitulé (les ensembles flous) [73-75] et [78]. Il a été le premier à formaliser les règles floues et à suggérer une approche (système expert) au réglage automatique où la règle de commande est remplacée par des règles floues [78]. En 1974, M. Mamdani a publié les premiers résultats de la logique floue dans la régulation d'un processus industriel (chaudière à vapeur), ce qui a encouragé différentes activités en Europe [73-75] et [78-80].

Grace au chercheur Japonais M. Sugeno, la logique floue était introduite au Japon dès 1985. L'industrie a lancé de nombreux produits basés sur cette technique, notamment des appareils électroménagers et du matériel audio visuel [73-75] et [81]. Actuellement, les principales directions de recherche sont la combinaison de la logique floue et d'autres techniques de régulation (commande hybride).

4.3. Concept de base de la logique floue

La définition de Zadeh : *“La logique floue est déterminée comme un ensemble de principes mathématiques pour la représentation de connaissances basées sur les degrés d'appartenance plutôt que sur l'appartenance nette à la logique binaire classique”* [77].

La logique booléenne classique ne permet que deux états : vrai ou faux, la logique floue permet d'exprimer différents niveaux, plutôt que seulement 1 ou 0 [73-75].

4.3.1. Théorie de l'ensemble flou

a. Définition d'un ensemble flou [75] et [81]

Si U est une collection d'objets ou des valeurs notées par x , alors un ensemble flou A dans U est défini par l'ensemble des couples ordonnées, soit :

$$A = \{(x, \mu_A(x)) / x \in U\}$$

A est défini par une fonction d'appartenance μ_A qui associe à chaque élément x de U le degré d'appartenance $\mu_A(x)$, qui prend ses valeurs dans $[0,1]$.

$$\mu_A(x) : U \rightarrow [0,1]$$

x : Élément générique de l'univers de discours U ;

A : Ensemble de U ;

μ_A : Degré d'appartenance de x à A .

Généralement, A est sous deux formes :

$$A = \int \frac{\mu_A(x)}{x} \quad \text{si } U \text{ est continu}$$

$$A = \sum_{x_i \in U} \frac{\mu_A(x_i)}{x_i} \quad \text{si } U \text{ est discontinu}$$

b. Fonction d'appartenance

Un fait certain aura une fonction d'appartenance égale à 1 pour le point de fonctionnement considéré, un fait incertain aura une fonction d'appartenance inférieure ou égale à 1, lorsque le fait certain correspond à l'énoncé de la valeur d'une variable $x = x_0$, on aura $\mu_{x_0}(x_0) = 1$ pour $x = x_0$ et $\mu_{x_0}(x_0) = 0$ pour $x \neq x_0$, on a alors un singleton [75]. Un fait incertain tel que x à peu près égal à x_0 aura une fonction d'appartenance en forme de triangle, l'affirmation x à peu près égal à x_2 et x_3 correspond à une fonction d'appartenance trapézoïdale [82].

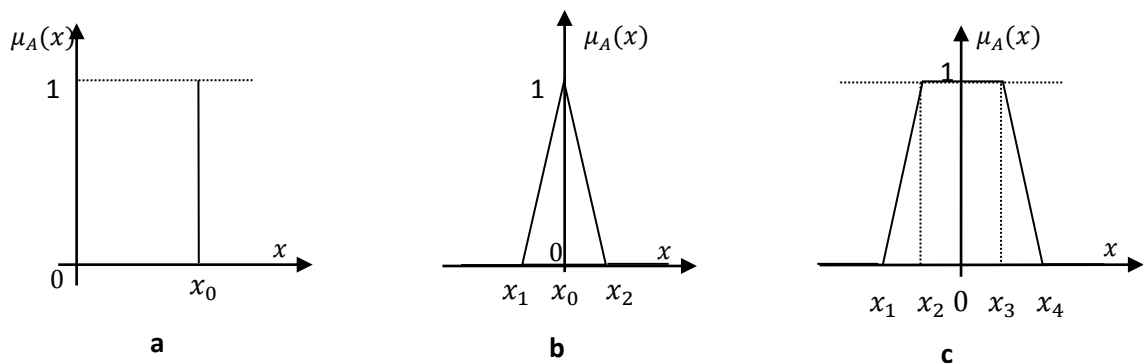


Figure (4.1) : (a) : Fonction singleton, (b) : Fonction triangulaire, (c) : Fonction trapézoïdale

Dans les applications de la logique, les fonctions d'appartenances $\mu_A(x)$ sont associées avec les termes qui apparaissent dans les antécédents et les règles. Les fonctions d'appartenances peuvent prendre plusieurs formes selon l'expérience de l'utilisateur.

c. Opérations sur les ensembles flous [81] et [83]

Il s'agit de la généralisation des opérateurs négation, intersection et union de la théorie des ensembles ordinaires, la figure (4.2) résume ces opérations de façon graphique.

- L'opérateur NON ; $\mu_{\bar{A}}(x) = 1 - \mu_A(x)$
- L'opérateur ET, $\mu_{A \cap B}(x) = \min(\mu_A(x), \mu_B(x))$
- L'opérateur OU, $\mu_{A \cup B}(x) = \max(\mu_A(x), \mu_B(x))$

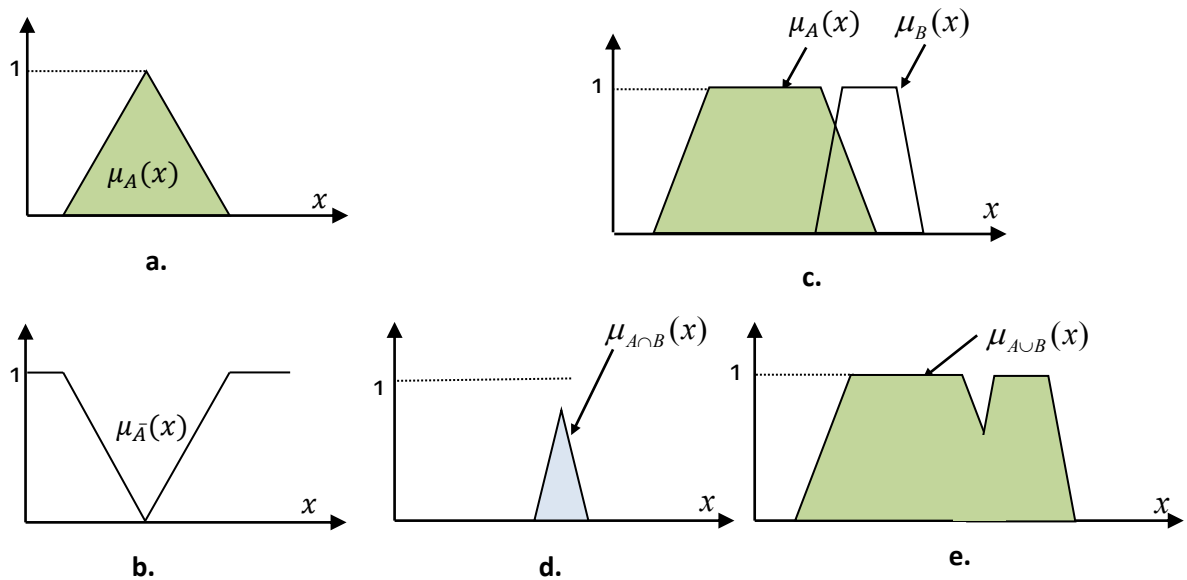


Figure (4.2) : Opérateurs logiques (a) :A, (b) : \bar{A} , (c) : A et B, (d) : $A \cap B$, (e) : $A \cup B$

d. Règles d'inférence [81-83]

On appelle règles d'inférence l'ensemble des différentes règles reliant les variables linguistiques d'entrées d'un système aux variables linguistiques de sorties par des opérateurs flous. La syntaxe est identique à celle de la logique binaire : **si** <condition> **alors** <conséquence>, la condition est une affirmation simple ou composée.

D'une façon générale les règles peuvent être exprimées sous forme :

Si	"antécédent 1"	ET/OU	"conséquence 1"
	"antécédent 2"	ET/OU	"conséquence 2"
	"antécédent n"	Alors	"conséquence n"

- Les antécédents sont les "conditions " alors que les conséquences correspondent aux "actions".
- Les antécédents correspondent aux entrées floues (degrés d'appartenances) déterminés lors de la fuzzification.

4.3.2. Variable linguistique

La description d'une certaine situation, d'un phénomène ou d'un procédé contient en général des qualificatifs flous tels que :peu, beaucoup, énormément, rarement, fréquemment, souvent froid, tiède, chaud, petit, moyen, grand etc. [74]. Ces mots ou phrases en langage naturel ou formel forment des variables linguistiques, parce que les mots généralement moins précis que le nombre, le concept de la variable linguistique donne éventuel rapprochement pour décrire ces phénomènes. Donc les variables linguistiques représentent un état dans le système à régler ou une variable de réglage dans un contrôleur flou [74] et [77].

Une variable linguistique est un triplet $(x, U, T(x))$ dans lequel :

x : variable définie sur l'ensemble U ;

U : Univers de discours associe avec la valeur de base ;

$T(x)$: Ensemble des valeurs linguistiques que peut prendre la variable x dans un univers de discours normalisé U .

4.4. Structure d'un contrôleur flou

Pour appliquer les règles floues et développer un signal de commande, la méthode d'inférence de Mamdani pour la conception d'un contrôleur flou est présentée dans cette section. La figure (4.3) illustre la configuration de base d'un contrôleur flou, elle est constituée de quatre blocs [73-75].

- Une interface de fuzzification à l'entrée ;
- Une base de connaissance ;
- Une logique de prise de décision ;
- Une interface de défuzzification à la sortie du système contrôle.

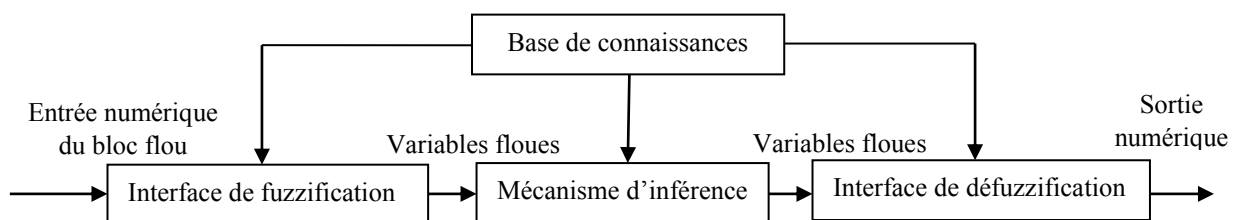


Figure (4.3) : Structure d'un contrôleur flou

4.4.1. Base de connaissances

Le bloc base de connaissances comporte des connaissances dans le domaine d'application et les résultats de commande prévus, il est composé de base de données et de base de règles linguistiques [74].

- La base de données regroupe des définitions qui sont nécessaires pour établir les règles de commande et manipuler les données floues dans le contrôle (univers de discours, partition floue, choix des opérateurs...etc.) ;
- La base de règles représente la stratégie de commande des experts du domaine et le but désiré par le biais des règles de commande linguistiques.

4.4.2. Interface de fuzzification

Le rôle de ce bloc est d'accomplir les fonctions :

- La définition des fonctions d'appartenance pour les variables d'entrées ;
- Le passage des grandeurs physiques en variables linguistiques qui peuvent être ainsi traitées par les règles floues.

L'opérateur de fuzzification sert à associer à une mesure x_0 de la variable x une fonction d'appartenance $\mu_{x_0}(x)$ [74] et [82].

4.4.3. Interface de défuzzification

La défuzzification est le processus de convertir une valeur floue en une valeur nette (sous forme numérique) [72-73]. Autrement dit la défuzzification est le traitement qui réalise une catégorie d'échelle convertissant la plage des variables de sortie aux univers de discours appropriés et aussi il permet de définir une correspondance entre le résultat de l'inférence et la grandeur continue fournie en sortie. Quelques méthodes existent, comme [73-75] et [83-85] :

- Méthode du Maximum ;
- Méthode de la moyenne des Maxima ;
- Méthode du centre de gravité.

Dans ce chapitre c'est la méthode de centre de gravité qui est utilisée.

Défuzzification par centre de gravité

Cette technique est plus performante et donne les meilleurs résultats, la défuzzification par centre de gravité consiste à calculer l'abscisse du centre de gravité de la fonction

d'appartenance de sortie résultante [75-78]. La valeur de sortie défuzzifiée est calculée en utilisant la formule suivante :

$$x_{0\text{défuzzifié}} = \frac{\sum_{i=1}^n x_i \mu_x(x_i)}{\sum_{i=1}^n \mu_x(x_i)}$$

4.4.4. Mécanisme d'interface

Le bloc inférence est le cœur d'un contrôleur à logique floue. Il possède la capacité de simuler les décisions humaines et de déduire (inférer) les actions de commande floue à l'aide de l'implication floue et des règles d'inférence dans la logique floue [84-85]. On peut décrire l'inférence de manière explicite par la description linguistique à l'aide d'un certain nombre de règles [75] comme :

$$\left\{ \begin{array}{l} R_1 : \text{Si } x \text{ est } A_1 \text{ et } y \text{ est } B_1 \text{ alors } z \text{ est } C_1 \\ R_2 : \text{Si } x \text{ est } A_2 \text{ et } y \text{ est } B_2 \text{ alors } z \text{ est } C_2 \\ \vdots \\ R_n : \text{Si } x \text{ est } A_n \text{ et } y \text{ est } B_n \text{ alors } z \text{ est } C_n \end{array} \right.$$

Les variables d'entrée sont x et y et la variable de sortie est z . Les sous ensembles flous sont les A_i , les B_i et les C_i .

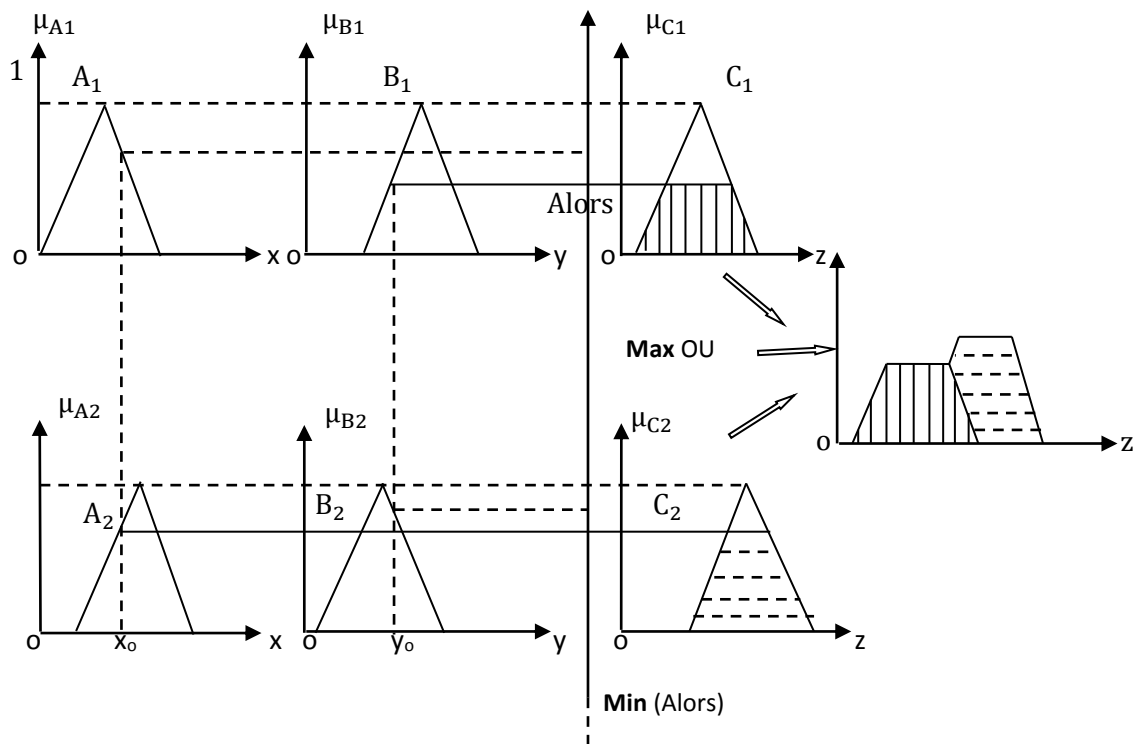


Figure (4.4) : Illustration de la méthode Min-Max

Il existe plusieurs méthodes d'inférence (Max-Min, Max-Produit et Somme-Produit) [75], la plus utilisée est la méthode Max-Min (implication de Mamdani). Cette méthode consiste à réaliser l'opérateur ET par la fonction Min, l'opérateur OU par la fonction Max et l'implication ALORS par la fonction Min [83], figure (4.4).

4.5. Conception du bloc à logique floue

Les régulateurs de normalisation avec des paramètres fixes peuvent être insuffisants dans le contrôle des systèmes tels que les robots, qui sont soumis à de fortes variations d'inertie et de la charge au cours de leurs cycles de fonctionnement. Cependant, les contrôleurs plus sophistiqués sont nécessaires, tels que les régulateurs adaptatifs, les régulateurs hybride, qui possèdent la capacité de modifier leurs caractéristiques afin de maintenir le comportement dynamique souhaité du système même en présence des variations paramétriques [78].

Dans ce chapitre, afin d'améliorer les résultats obtenus dans le chapitre précédent en terme de réduction du phénomène de chattering, un bloc flou est conçu permettant un réglage en temps réel des gains du contrôleur cinématique, figure (4.5). Cette approche floue est développée dans le but d'apporter une amélioration des performances statiques et dynamiques.

Comme nous avons vu précédemment, dans le troisième chapitre, le contrôle de poursuite du robot se fait en ajustant en permanence les couples des deux roues. D'après la modélisation du robot, deux contrôleurs sont conçus, un contrôleur dynamique basé sur le mode glissant est développé pour commander le déplacement du robot et un contrôleur cinématique dont le rôle consiste à générer une loi de commande V_c qui permet d'annuler l'erreur de poursuite q_e [76] et [86]. On a pu constater que la qualité des résultats de simulation obtenus dépend du choix des gains du contrôleur.

Dans ce contexte et dans le but d'apporter une solution au choix des gains, un bloc flou est incorporé permettant un auto-réglage de ces gains.

Le schéma global de la commande est illustré par la figure suivante :

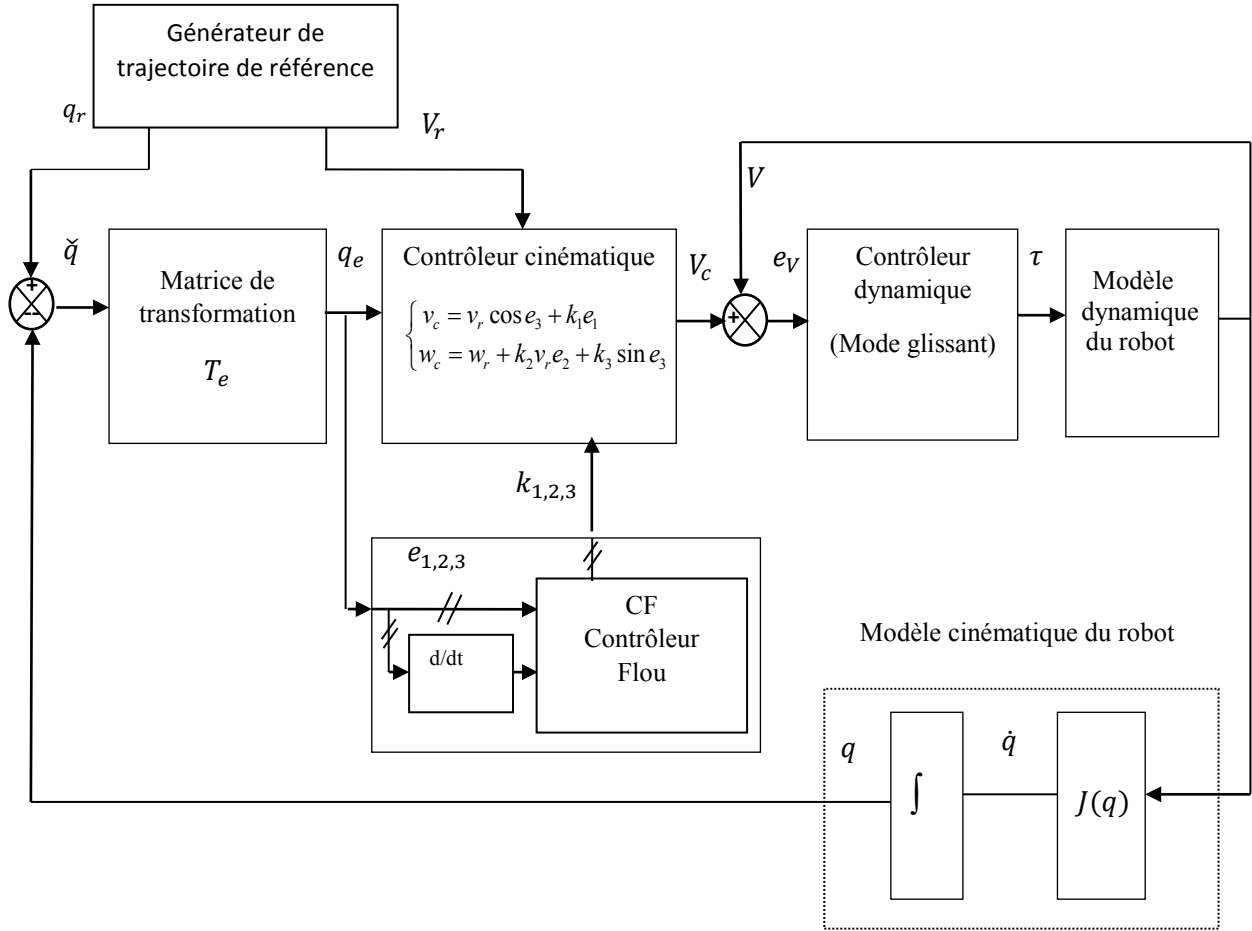


Figure (4.5) : Schéma global de la commande du robot

D'après ce qui a été présenté dans les sections 2.5. et 3.5.1., les vecteurs d'entrée et de sortie du contrôleur cinématique, q_e et V_c , sont donnés par les relations (2.41) et (3.10), respectivement, données par :

$$q_e = \begin{bmatrix} e_1 \\ e_2 \\ e_3 \end{bmatrix} = \begin{bmatrix} \cos\theta & \sin\theta & 0 \\ -\sin\theta & \cos\theta & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x_r - x \\ y_r - y \\ \theta_r - \theta \end{bmatrix} = T_e \tilde{q}$$

$$\begin{cases} v_c = v_r \cos e_3 + k_1 e_1 \\ w_c = w_r + k_2 v_r e_2 + k_3 \sin e_3 \end{cases} ; \quad k_1, k_2, \text{ et } k_3 \in \mathbb{R}^{**}$$

Il a été prouvé, en choisissant la fonction de Lyapunov suivante, la stabilité du système cinématique est théoriquement satisfaite.

$$V_0 = \frac{1}{2}(e_1^2 + e_2^2) + \frac{1 - \cos(e_3)}{k_2} > 0$$

Autrement, l'équation (2.41) peut être écrite sous forme :

$$\dot{e} = Ae ; \text{ avec : } e = q_e = \begin{bmatrix} e_1 \\ e_2 \\ e_3 \end{bmatrix}$$

Démonstration :

En sélectionnant la fonction de Lyapunov suivante :

$$L_1 = \frac{1}{2}(e_1^2 + e_2^2 + e_3^2) > 0 \quad (4.1)$$

Sa dérivée donne :

$$\dot{L}_1 = \dot{e}_1 e_1 + \dot{e}_2 e_2 + \dot{e}_3 e_3$$

$$\dot{L}_1 = (-v_c + w_c e_2 + v_r \cos(e_3))e_1 + (-w_c e_1 + v_r \sin(e_3))e_2 + \frac{1}{k_2}(w_r - w_c)e_3$$

$$\dot{L}_1 \approx (-v_c + v_r \cos(e_3))e_1 + (v_r e_2 + \frac{1}{k_2}(w_r - w_c))e_3 = (-k_1 e_1)e_1 + (-k_3 e_3)e_3 \quad (4.2)$$

Qui donne :

$$\dot{e} = \begin{cases} \dot{e}_1 = -k_1 e_1 + w_c e_2 \\ \dot{e}_2 = v_r e_3 - w_c e_1 \\ \dot{e}_3 = -k_3 e_3 - v_r e_2 \end{cases} \Rightarrow A = \begin{bmatrix} -k_1 & w_c & 0 \\ -w_c & 0 & v_r \\ 0 & -v_r & -k_3 \end{bmatrix} \quad (4.3)$$

On remarque que la matrice A ne dépend pas du gain k_2 (il suffit que $k_2 > 0$)

, selon la relation (4.6).

En réalité deux blocs sont incorporés dans le système de commande, le premier génère l'adaptation du gain k_1 , le deuxième Le but est de concevoir un contrôleur flou qui est capable de générer une loi de commande h ($0 < h \leq 1$) pour assurer le réglage désiré de k_1 et k_3 du contrôleur cinématique concerne le gain k_2 . Les variables d'entrées du bloc flou $CF1$ sont l'erreur e_1 et sa variation. Idem, les variables d'entrées du bloc $CF2$ sont l'erreur e_2 et sa variation. Le bloc $CF1$ fournit la variable d'adaptation h_1 , de même $CF2$ génère la variable d'adaptation h_2 . Ces deux variables fournissent un auto-réglage des gains k_1 et k_3 (utilisés dans le contrôleur cinématique) selon l'équation (4.6).

Les variables entrées sont définies par les équations (4.4) et (4.5).

e_1 et e_3 sont définis par l'équation (2.41)

$$\Delta e_1(k) = e_1(k) - e_1(k-1) \quad (4.4)$$

$$\Delta e_3(k) = e_3(k) - e_3(k-1) \quad (4.5)$$

Les entrées et les sorties de chaque contrôleur flou sont normalisées à l'aide des gains de normalisation, G_e , $G_{\Delta e}$ et G_h respectivement (voir figure (4.6)). Ces facteurs d'échelle

peuvent être des constantes ou des variables [87], ils jouent un rôle important dans la conception du contrôleur flou.

Les fonctions d'appartenance triangulaires avec chevauchement sont utilisées pour les variables entrées/sorties des blocs flous, elles sont représentées par la figure (4.7).

- N (négatif), P (positif), Z (zéro), PE (petit) et G (grand).

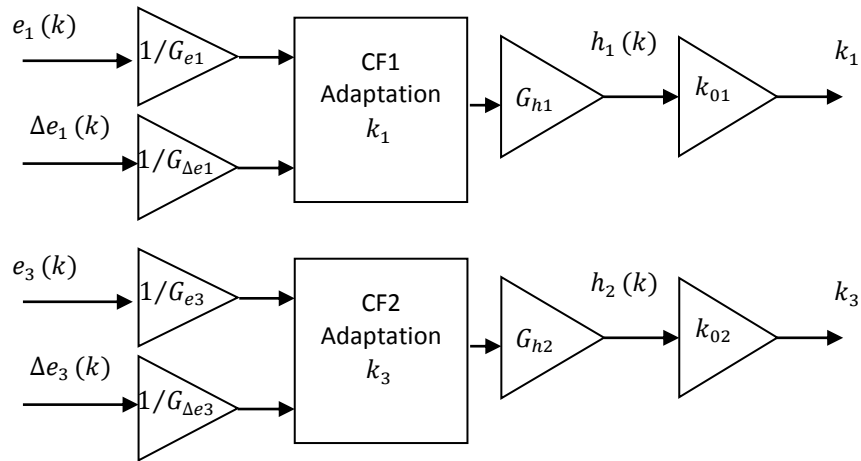


Figure (4.6) : Schéma bloc du contrôleur flou

La variable de sortie de chaque bloc flou est la loi de commande $h_{1,2}$ ($0 < h_{1,2} \leq 1$). Cette loi permet l'adaptation des gains de l'équation (3.10) comme le montre l'équation suivante :

$$k_1 = k_{01} * h_1 \text{ et } k_3 = k_{02} * h_2 ; \quad 0 < h_{1,2} \leq 1 \quad (4.6)$$

Pour la conception des tables de décision tableau 1 et tableau 2, une expertise est extraite à partir des tests de simulation effectués au niveau du chapitre 3.

Exemple explicatif : Table 1

Si l'erreur e_1 est N (et) Δe_1 est N (alors) la variable de la loi d'adaptation " h_1 " doit être faible pour atténuer le dépassement et réduire par conséquence l'erreur.

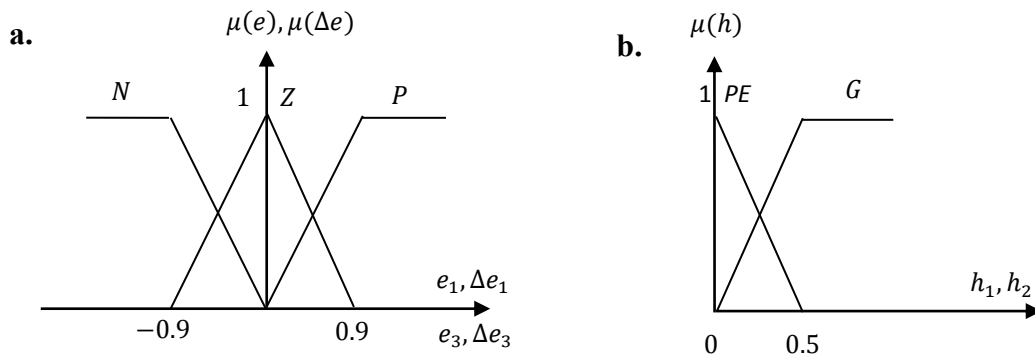


Figure (4.7) : Fonctions d'appartenance (a):Variables d'entrées $e_1, e_3, \Delta e_1$ et Δe_3 , (b):Variables de sorties h_1 et h_2

Table 1: Table de décision du bloc flou CF1

h_1		Δe_1		
		N	Z	P
e_1	N	PE	G	PE
	Z	PE	G	PE
	P	PE	G	PE

Table 2: Table de décision du bloc flou CF2

h_2		Δe_3		
		N	Z	P
e_3	N	G	PE	G
	Z	G	PE	G
	P	G	PE	G

On note que dans cette approche floue on a opté pour le mécanisme d'inférence de Mamdani. Concernant la méthode de défuzzification, la technique du centre de gravité est exploitée.

4.6. Résultats de simulation

On se place toujours dans les mêmes conditions de fonctionnement : trajectoires de poursuite et variations paramétriques, figure (4.8).

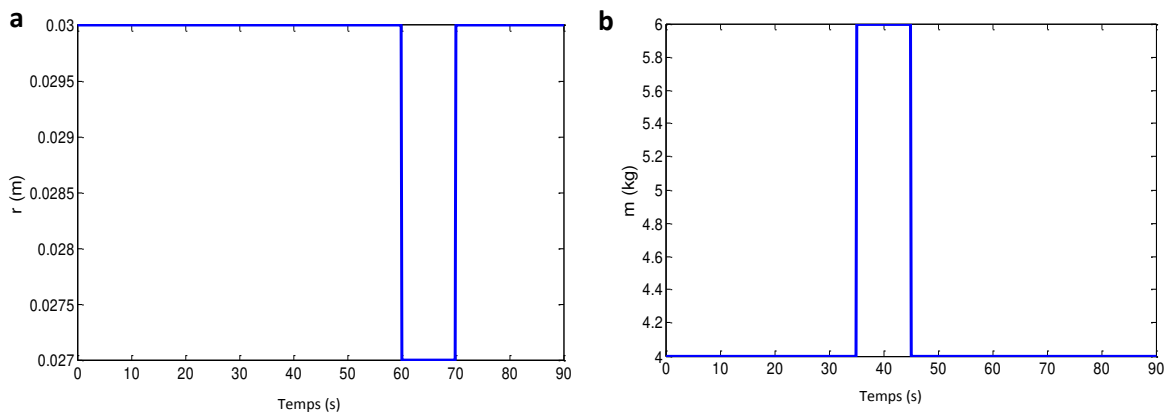


Figure (4.8) : Variation des paramètres (a) Rayon du robot (b) Masse du robot

a. Trajectoire circulaire

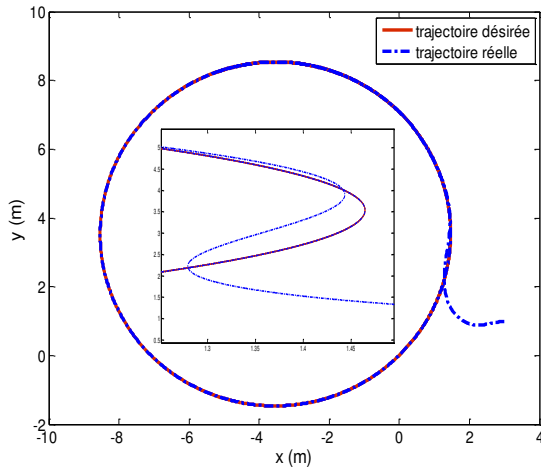


Figure (4.9) : Trajectoire circulaire

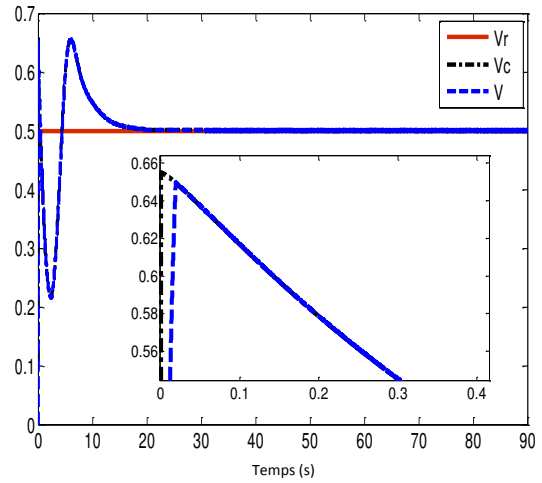


Figure (4.10) : Vitesses linéaires (m/s)

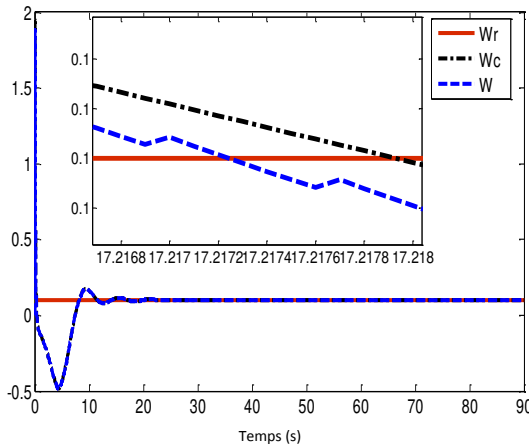


Figure (4.11) : Vitesses de rotation (rd/s)

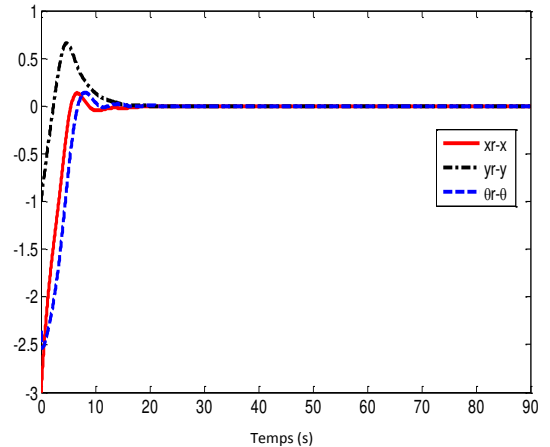


Figure (4.12) : Erreur en distance et en orientation (x_r-x , y_r-y , $\theta_r-\theta$)

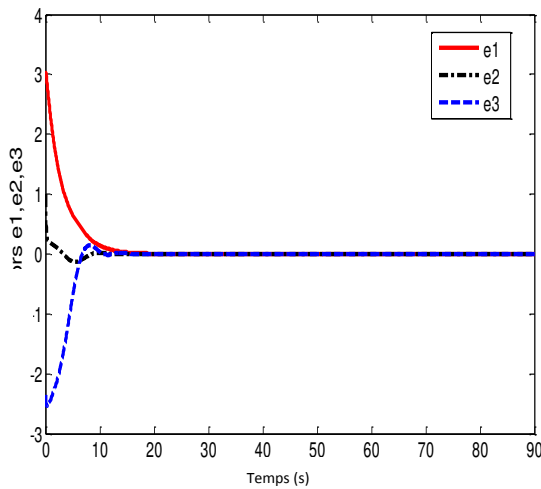


Figure (4.13) : Erreur de poursuite (e_1 , e_2 , e_3)

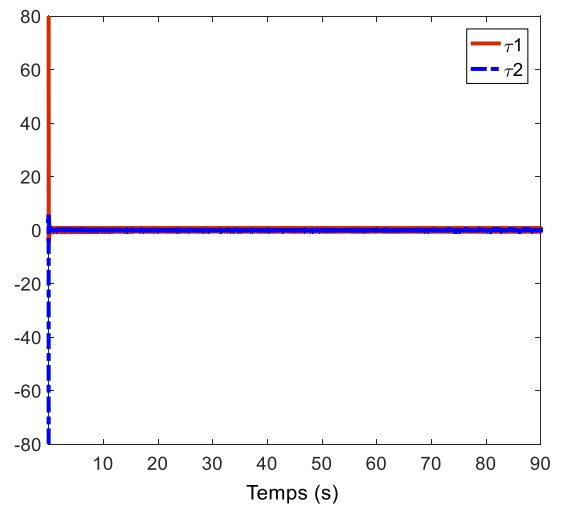


Figure (4.14) : Couple (τ_1 , τ_2) (N*m)

b. Trajectoire carrée

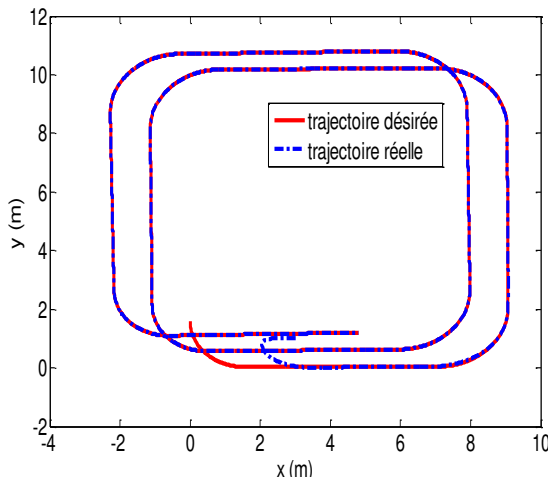


Figure (4.15) : Trajectoire carrée

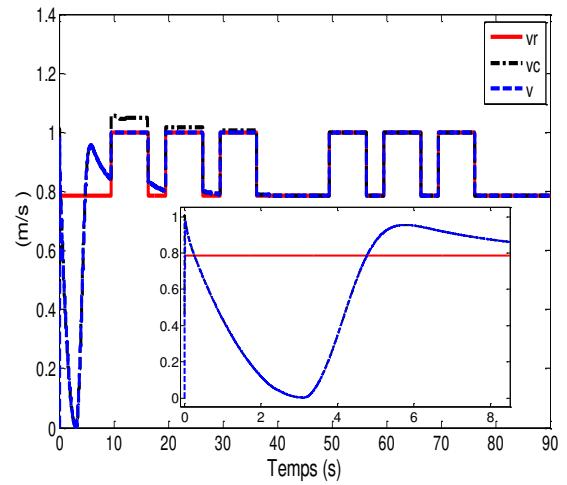


Figure (4.16) : Vitesses linéaires (m/s)

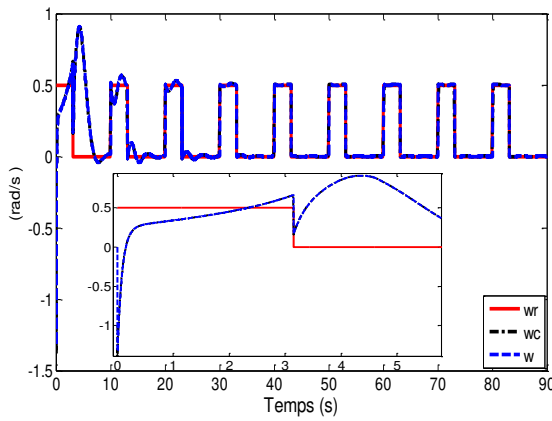


Figure (4.17) : Vitesses de rotation (rd/s)

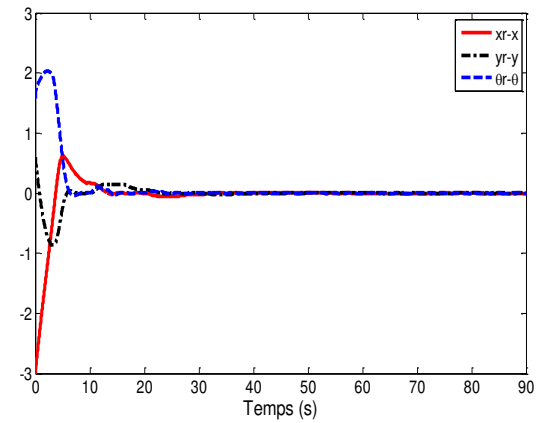


Figure (4.18) : Erreur en distance et en orientation (x_r-x , y_r-y , $\theta_r-\theta$)

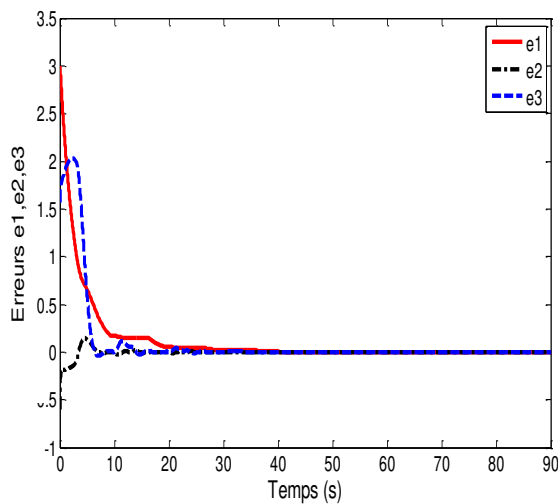


Figure (4.19) : Erreur de poursuite (e_1 , e_2 , e_3)

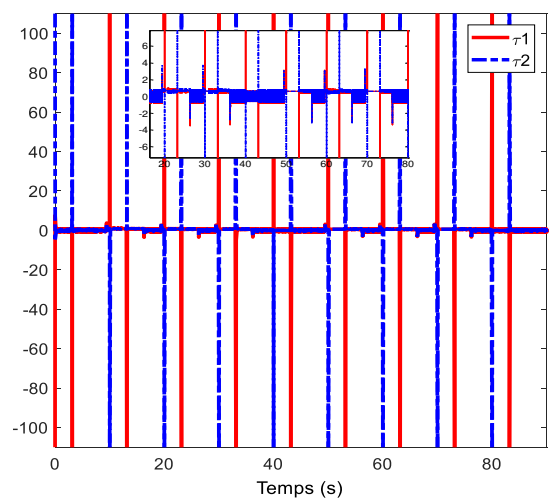


Figure (4.20) : Couple (τ_1 , τ_{21}) (N*m)

Les résultats obtenus sont représentés par les figures (4.9)-(4.14) et (4.15)-(4.20), les figures (4.9) et (4.15) représentent une bonne flexibilité de poursuite de la trajectoire, malgré la variation des paramètres. Les figures (4.10)-(4.11) et (4.16)-(4.17) montrent, respectivement, l'amélioration des caractéristiques dynamiques et statiques de la vitesse linéaire et de rotation. On note, une nette réduction du dépassement de la vitesse linéaire V_c et l'annulation de l'erreur statique de la vitesse de rotation par rapport aux résultats obtenus dans le chapitre précédent. Les réponses des erreurs de poursuite ($x_r - x, y_r - y, \theta_r - \theta$) sont similaires à la réponse d'un système du second ordre. On constate aussi une amélioration dans les caractéristiques dynamiques et statiques des réponses des erreurs (e_1, e_2, e_3) et une réduction du phénomène de chattering au niveau du couple (figure (4.20)) en terme de fréquence est acquise suite à l'introduction de la logique floue.

4.7. Synthèse des trois approches

Afin de mettre en relief l'apport de chaque approche pour l'amélioration des performances dynamiques et statiques, un tableau récapitulatif est donné. Une étude comparatif concernant le temps de réponse des erreurs de poursuite, des dépassements, etc. est établie. Les conclusions sont regroupées dans le tableau ci dessous.

Table 3: Performances dynamiques et statiques des approches

	Temps de réponse des erreurs de poursuite	Dépassement des erreurs de poursuite	Fréquence de phénomène chattering 'couple'	robuste
Approche 1 backstepping	faible	faible	moyenne	acquise
Approche 2 mode glissant	faible	faible	forte	acquise
Approche 3 Hybride glissant-floue	légère augmentation	légère augmentation	réduite	acquise

4.8. Conclusion

Une approche floue est conçue dans ce chapitre afin d'apporter une amélioration des performances acquises au niveau du chapitre 3. Les résultats obtenus sont satisfaisants et reflètent l'apport de la logique floue dans le domaine du contrôle du robot.

Chapitre 5

Implémentation pratique et résultats expérimentaux

CHAPITRE 5

IMPLEMENTATION PRATIQUE ET RESULTATS EXPERIMENTAUX

5.1. Introduction

Le problème de la navigation peut être décomposé en plusieurs tâches, figure (5.1), la planification des trajectoires, l'évitement d'obstacles, etc. [29]. Nous nous intéressons dans cette partie aux fonctionnalités de suivi d'objets nécessaires pour traiter principalement la localisation visuelle d'un robot dans son environnement, nous nous sommes essentiellement intéressés à des entités (objets, amers, primitives visuelles...) **statiques** et **rigides** dans un environnement spécifique "**milieu interne**".

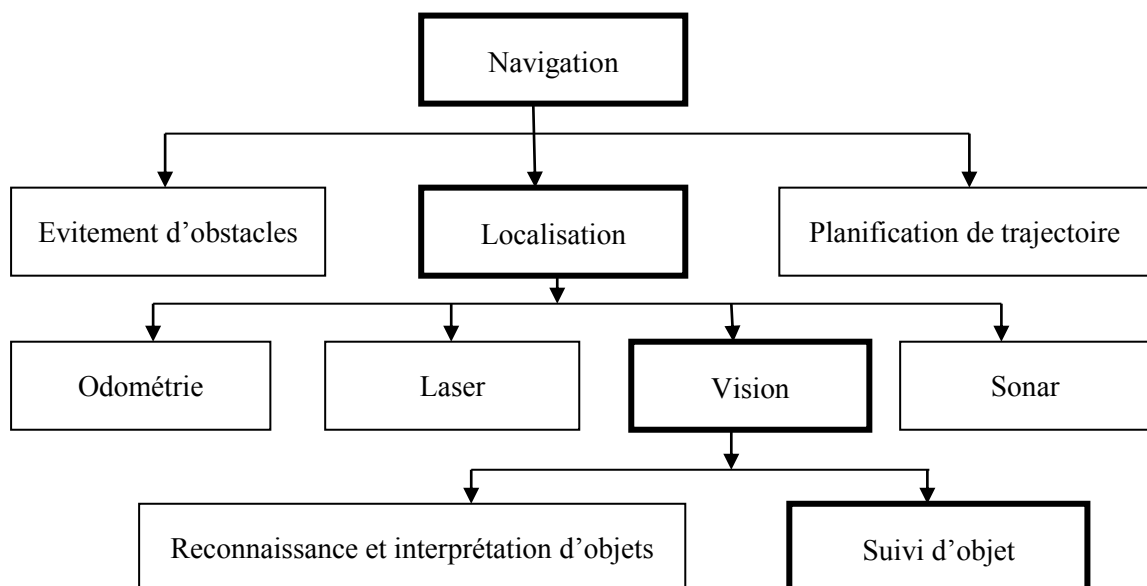


Figure (5.1) : Schéma simplifié de tâches de la navigation autonome

Pour acquérir des informations sur l'environnement du robot, les capteurs extéroceptifs sont indispensables, car les capteurs proprioceptifs (odométrie) fournissent des renseignements sur l'état interne du robot, mais ils ne tiennent pas en compte des problèmes dûs à l'interaction avec l'environnement et par une des conséquences ils ne permettent pas de situer précisément le robot. On distingue plusieurs types de capteurs extéroceptifs [30] :

- Les capteurs tactiles qui préviennent d'un contact avec un élément extérieur ;
- Les capteurs proximétriques qui détectent la présence d'un objet situé à une distance inférieure au mètre ;
- Les capteurs télémétriques pour des objets situés à une distance plus importante ;
- Les capteurs visuels (caméra) qui permettent de percevoir l'environnement.

Notre travail est basé sur un capteur visuel qui est la caméra, cette dernière permet de délivrer des informations pouvant être utilisées au niveau d'action afin de permettre des actions reflexes en intervenant directement au niveau de la boucle d'asservissement du robot.

Le but de ce chapitre est d'asservir visuellement un guidage visuel d'un robot mobile, il s'agit d'une commande en temps réel d'un robot EV3 qui doit se déplacer dans un environnement intérieur et il doit suivre une trajectoire bien définie, en exploitant les informations perçues par une camera fixe.

5.2. Méthodologie retenue

Plusieurs classifications des méthodes de suivi visuel d'objets ont été proposées dans la littérature, elles dépendent autant des auteurs, que du but pour lequel ces méthodes ont été conçues. Selon les auteurs, les méthodes de suivi visuel peuvent être divisées en quatre classes [29] :

- Méthodes de suivi fondées sur des modèles, ces méthodes repèrent des caractéristiques connues dans la scène et les utilisent pour mettre à jour la position de l'objet. Parmi ces méthodes, il y a celles qui exploitent les modèles géométriques fixes et les modèles déformables.
- Méthodes de suivi de régions ou blobs, cette sorte de méthodes se caractérise par la définition des objets d'intérêt comme ceux qui sont extraits de la scène en utilisant des méthodes de segmentation.
- Méthodes de suivi à partir de mesures de vitesse, ces méthodes peuvent suivre les objets en exploitant les mesures de leur vitesse dans l'image, avec des mesures telles que le flux optique ou des équivalents.

- Méthodes de suivi de caractéristiques, ces méthodes suivent certaines caractéristiques de l'objet, comme des points, des lignes, des contours, caractéristiques ou primitives image auxquelles il est possible aussi d'imposer de restrictions globales, ces caractéristiques peuvent être aussi définies par la texture ou la couleur.

Cette classification n'est pas exhaustive, et à ce jour, il existe de nombreux recouvrements entre les classes, c'est à dire, des méthodes qui peuvent être classifiées dans deux ou plus de classes.

5.3. Description de la plateforme expérimentale

La composition de l'ensemble est la suivante : un environnement intérieur, un robot mobile EV3, une caméra pour récupérer la position du robot et un module wifi pour communiquer avec les robots depuis l'ordinateur. Le principe de fonctionnement de la plateforme est que l'espace d'évolution du robot est surveillé par une caméra placée verticalement au dessus l'environnement. Cette dernière transmet à l'ordinateur des images qui sont traitées de manière à extraire la position du robot dans l'espace de travail. Les commandes sont transférées aux robots à partir du calculateur par l'intermédiaire d'une connexion wifi, figure (5.2).

Dans cette configuration, la caméra est fixe et observe la scène de mouvement du robot. Dans ce cas, la tâche asservie par vision doit être réalisée dans le champ de vision de la caméra, elle est donc plus adaptée à des tâches locales exécutées dans un petit espace de travail du robot. De plus dans ce type de configuration, le robot sera un système simple, et n'effectuera que peu de tâches. C'est le système de vision qui est très important dans ce cas car c'est lui qui va donner la position et la vitesse du robot [8], [28], [30] et [88-89].



a.



b.

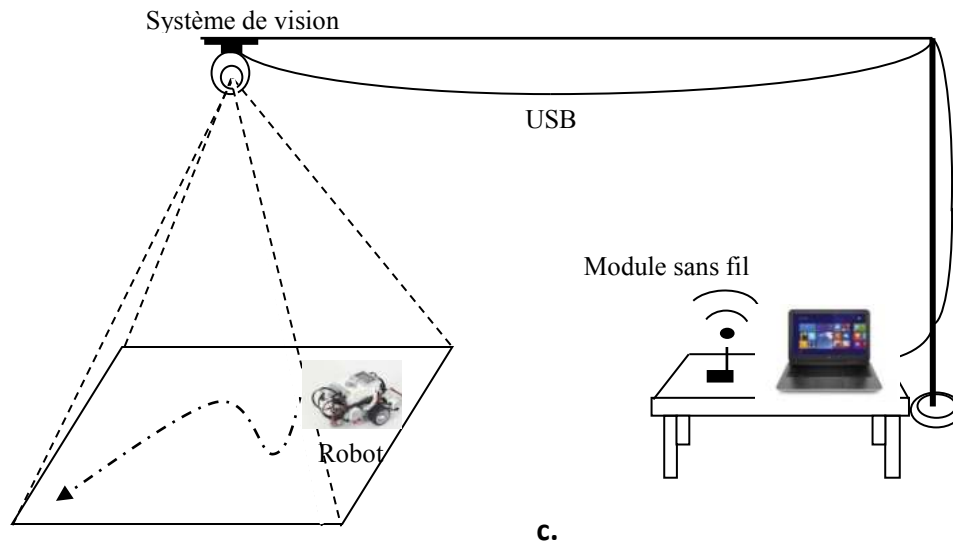


Figure (5.2) : (a) : Conception du robot mobile EV3, (b) : Camera fixe, (c) : Architecture de commande du robot mobile

La spécification de l'ordinateur HP adoptée ici est Intel(R) Core(TM) i5-5200U CPU 2.20 GHz avec une carte graphique non intégrée NVIDIA GEFORCE GF117. La caméra HP Starter basic webcam de résolution 'RGB24_320×420', est utilisée pour capter les images en temps réel et les transmettre à l'ordinateur (RAM). Le modem sans fil est exploité comme support de communication entre l'ordinateur et le robot mobile, où le protocole de transmission, soit l'émission des données de la caméra à l'ordinateur ou l'émission et la réception des données entre le robot et l'ordinateur, se fait dans l'environnement Simulink Matlab 2017. Le contrôleur de niveau haut, configuré sur EV3, prend en charge le contrôle de mouvement. Le robot EV3 se compose, des moteurs à courant continu, des encodeurs, du mécanisme structurel et surtout une brique intelligente, annexe "C".

5.4. Schéma global vision-commande

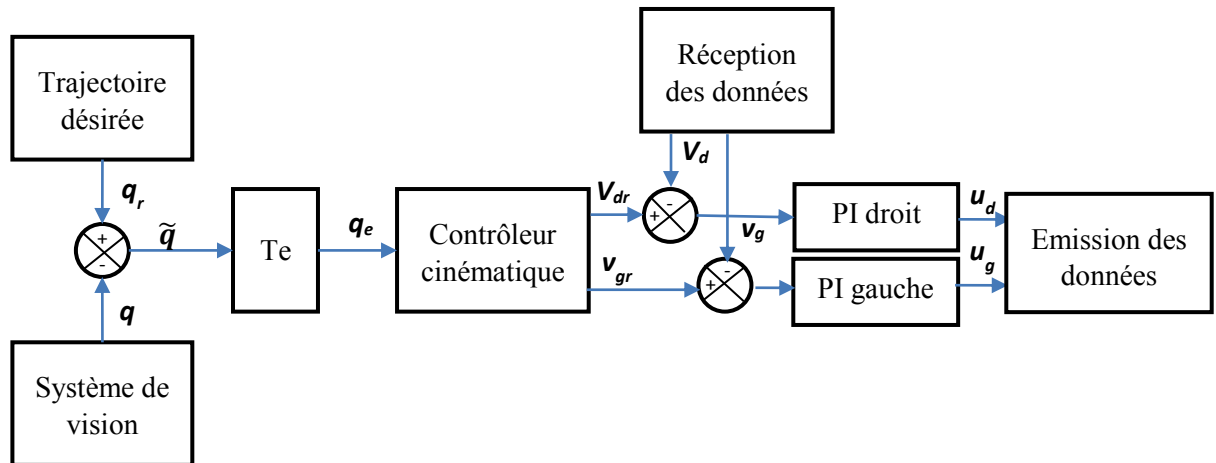
La configuration expérimentale est représentée sur la figure (5.3), elle est établie en intégrant le module de traitement d'image en temps réel, le module de communication sans fil et le module de décision de stratégie. Elle se compose de trois parties :

Partie Pc : assure le protocole de communication entre le PC et le robot.

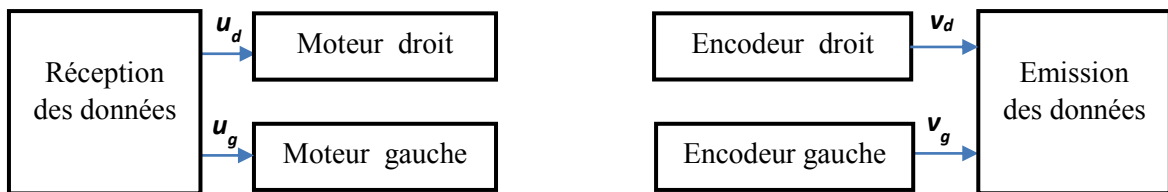
Partie robot : Le programme est chargé dans la brique du robot, qui reçoit les consignes, et transmet les consignes vitesses des roues du robot.

Partie vision : assure le traitement de vidéo, de l'acquisition jusqu'au le calcul des paramètres cinématiques du robot.

a. Partie PC



b. Partie robot



c. Partie vision

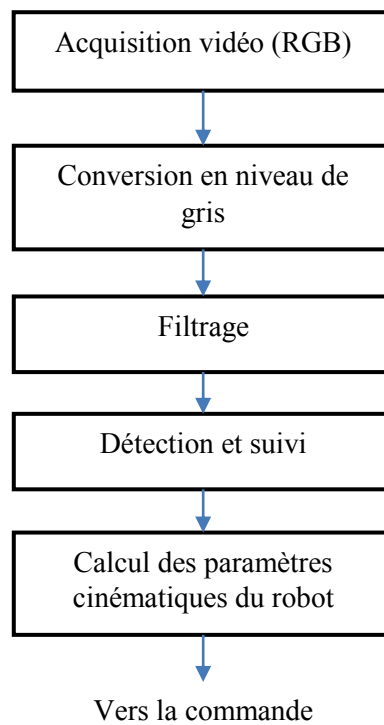


Figure (5.3) : Algorithme de poursuite du robot mobile

5.5. Système de vision

Pour que le robot se déplace dans un environnement réel, il est nécessaire d'obtenir ses paramètres de position et d'orientation. Pour cette tâche, différentes mesures ont été utilisées dans la littérature. Dans ce contexte, une méthode de vision basée sur la caméra est utilisée pour estimer la position et la direction du robot, elle est montée de façon fixe pour fournir une vue holistique de l'emplacement du robot. L'utilisation d'une caméra HP comme dispositif de mesure pour capter des images à partir du milieu local, et en utilisant des algorithmes de traitement d'image et de suivi, permet d'estimer la position et la direction de l'objet d'intérêt qui est le robot EV3. Nous supposons que la caméra est installée au plafond au centre de l'environnement et à une hauteur adéquate afin d'ignorer les effets de perspective aux angles des images captées. Ainsi, un étalonnage linéaire peut être utilisé pour la conversion entre les indices de pixels (i, j) dans l'image et les coordonnées (x, y) du milieu. Cette hypothèse peut introduire des erreurs d'approximation, l'utilisation d'une connaissance préalable de la cinématique du robot dans un cadre de filtrage peut corriger ces erreurs de mesure.

Dans cette section, nous présentons l'algorithme de base pour le traitement de l'image, visant à extraire des informations numériques. Ces dernières contiennent des paramètres qui présentent la scène observée par la caméra.

5.5.1. Détection du mouvement

La détection d'objet représente le fondement de tout algorithme de tracking. L'approche commune de détection des différents objets à suivre n'utilise l'information contenue que dans une image de la séquence à chaque itération. Il est cependant possible d'utiliser une information temporelle calculée à partir d'un ensemble d'images afin de réduire le nombre de fausses détections. On peut distinguer quatre ensembles d'algorithmes de détection : la détection de points dits d'intérêt, la segmentation d'image, la classification supervisée et enfin la soustraction de fond [89].

Nous commençons d'abord par la détection d'objet (robot) en mouvement dans une séquence d'image, par conséquent aux méthodes de séparation des objets mobiles du fond de l'image (l'arrière-plan). Dans ce contexte, la méthode utilisée est la méthode de soustraction de fond qui est basée essentiellement sur la détection des contours en mouvement qui est une technique proposée dans le contexte particulier de la robotique ou de la surveillance [89-90].

La figure suivante illustre les étapes nécessaires afin d'estimer la dynamique d'un objet dans un environnement.

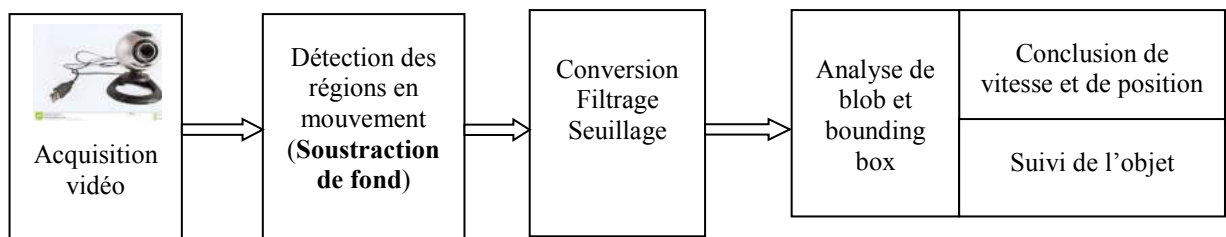


Figure (5.4) : Algorithme général de la détection du robot en mouvement

L'approche proposée doit vérifier les étapes suivantes :

- Acquisition de vidéo en temps réel via une caméra ;
- L'estimation de la partie fixe (fond) ;
- L'obtention d'images de différence de trames ;
- L'application de la valeur de seuil ;
- Avoir une estimation précise de la position et de la vitesse du robot.

a. Système d'acquisition de vidéo

Les séquences d'images de la détection et suivi de l'objet en mouvement (robot) dans une scène sont effectuées à l'aide d'une caméra fixe, sous différentes conditions de changement dans l'éclairage et le bruit, la taille de l'image est de 160×120 pixels.

Une image est un signal discret. Il est donc nécessaire de passer du domaine continu au domaine discret au niveau de la scène et de l'intensité lumineuse émanant de celle-ci. De manière générale, le spectre des intensités lumineuses noir et blanc est quantifié sur 256 niveaux de gris différents si on admet que chaque pixel est codé sur un octet.

b. Détection des régions en mouvement

Afin de détecter un objet, il est possible de construire une représentation de la scène appelée (modèle de fond) et de repérer ensuite les variations par rapport à ce modèle pour chaque nouvelle image, chaque changement significatif vis à vis du modèle représente un objet mobile [89]. C'est une étape critique et difficile car le mouvement d'objets induit des différences temporelles entre images, sa détection s'appuie sur l'étude des variations temporelles de la fonction de luminance, si le capteur est fixe et que l'éclairage de la scène varie peu, alors nous supposons que toute variation temporelle de l'intensité est liée au mouvement d'un objet ou à du bruit, donc cette étape doit être robuste aux variations de la luminosité de la scène et à la présence des ombres [28] et [89-93]. Dans la littérature, il existe

différentes méthodes de détection du mouvement chaque méthode admet plusieurs algorithmes. La méthode qu'on a choisie dans ce travail est la soustraction de l'arrière plan (en anglais Background Subtraction : BGS).

Cette méthode consiste à modéliser l'arrière plan qui est continuellement mis à jour. Initialement une moyenne d'intensité est calculée à partir de n premières images pour chaque pixel et pour chacun des canaux (R : rouge 'Red', G : vert 'Green' et B : bleu 'Blue'). Cette image moyenne sera considérée comme l'arrière plan initial de la scène, la moyenne d'intensité d'un pixel est comme suit :

$$\mu(x, y, t) = \frac{1}{n} \sum_{t=0}^n I(x, y, t) \quad (5.1)$$

où : $I(x, y, t)$: est la valeur d'intensité du pixel (x, y) , sur l'image I au temps t ,

$\mu(x, y, t)$: est la valeur d'intensité du pixel (x, y) , sur l'image arrière-plan au temps t .

On considère que chaque image de la séquence contribue à la mise à jour de l'arrière plan:

$$\mu(x, y, t) = \frac{\mu(x, y, t-1) + I(x, y, t)}{t - (t-1)} \quad (5.2)$$

Si $|I(x, y, t) - \mu(x, y, t)| > \sigma$, alors le pixel (x, y) de I à l'instant t est en mouvement,

Sinon, le pixel (x, y) de I appartient à l'arrière plan.

c. Conversion, filtrage et seuillage

Conversion vers niveau de gris : Pour convertir une image couleur en niveau de gris il faut remplacer, pour chaque pixel, les trois valeurs représentant les niveaux de rouge, de vert et de bleu, en une seule valeur représentant la luminosité.

Filtrage : Cette tâche permet de produire une image complète (avant-plan) contenant seulement les pixels représentant du mouvement.

Seuillage : permet de détecter et éliminer les ombres qui accompagnent les objets en mouvement lors d'opération de détection des objets en mouvement.

d. Suivi de mouvement (motion tracking)

Le suivi des objets est une étape importante, il consiste à faire une estimation de la position des objets mobiles dans le temps et l'espace du champ de vision afin de déterminer la trajectoire des objets mobiles à travers les images successives d'une séquence. Cela se fait par la mise en correspondance d'un objet sur l'image avec le même objet sur l'image successive. Pour cela, le suivi doit résoudre des problèmes tels que l'apparition de nouveaux objets

mobiles dans la scène ou leur disparition, l'occultation partielle ou totale, la fusion et la séparation des blobs.

5.5.2. Implémentation pratique et résultats

Le modèle Simulink représenté par la figure suivante, contient les ensembles de bloc nécessaires pour le traitement d'image.

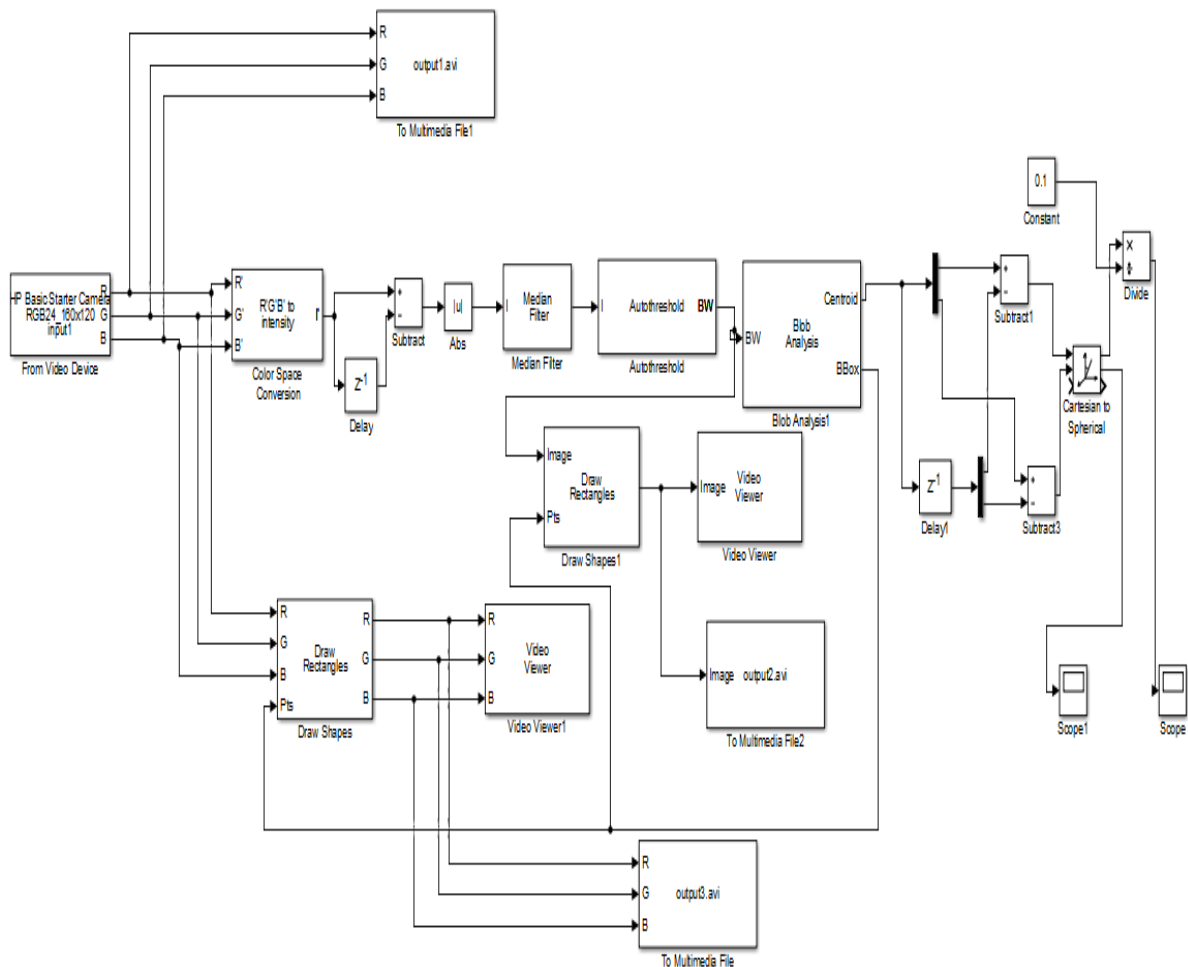


Figure (5.5) : Modèle Simulink proposé pour la vision

Pour la détection et le suivi du robot en mouvement, le modèle proposé contient plusieurs blocs de traitement, le premier bloc (from Video Device) permet d'acquérir en temps réel les données de flux vidéo provenant de dispositif d'acquisition d'image qui est la caméra (mono-vision), Ces données représentent une scène intérieure réelle avec un robot à différentes positions.

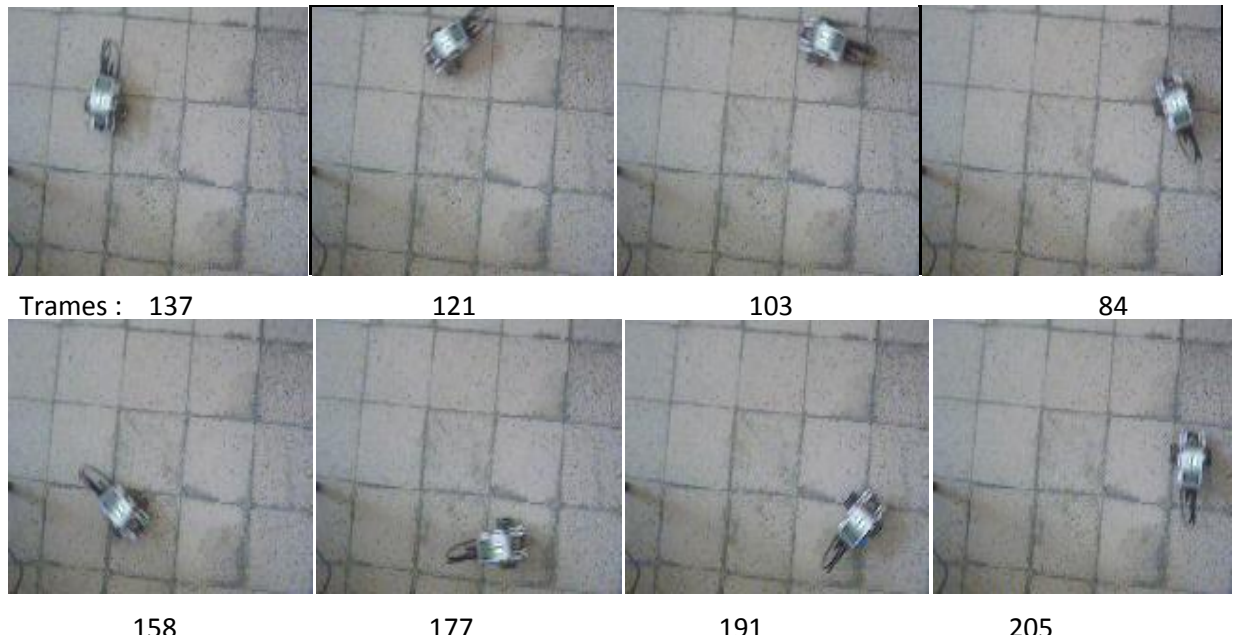


Figure (5.6) : Quelques trames qui représentent l'acquisition de la vidéo

Toutes les trames (images) utilisées dans l'expérimentation sont initialement en RGB, elles passent par un prétraitement, en utilisant le bloc de conversion de l'espace de couleur (color space conversion), ces trames en couleurs sont converties en niveau gris afin d'alléger les calculs en réduisant le nombre de canaux à 1, cela n'influe pas sur les résultats de l'opération d'extraction des points d'intérêt. Chaque couleur a une intensité I codée de 0-255, I , résultante de conversion, elle représente la combinaison de ces couleurs pour avoir un niveau de gris qui varie de 0-255.

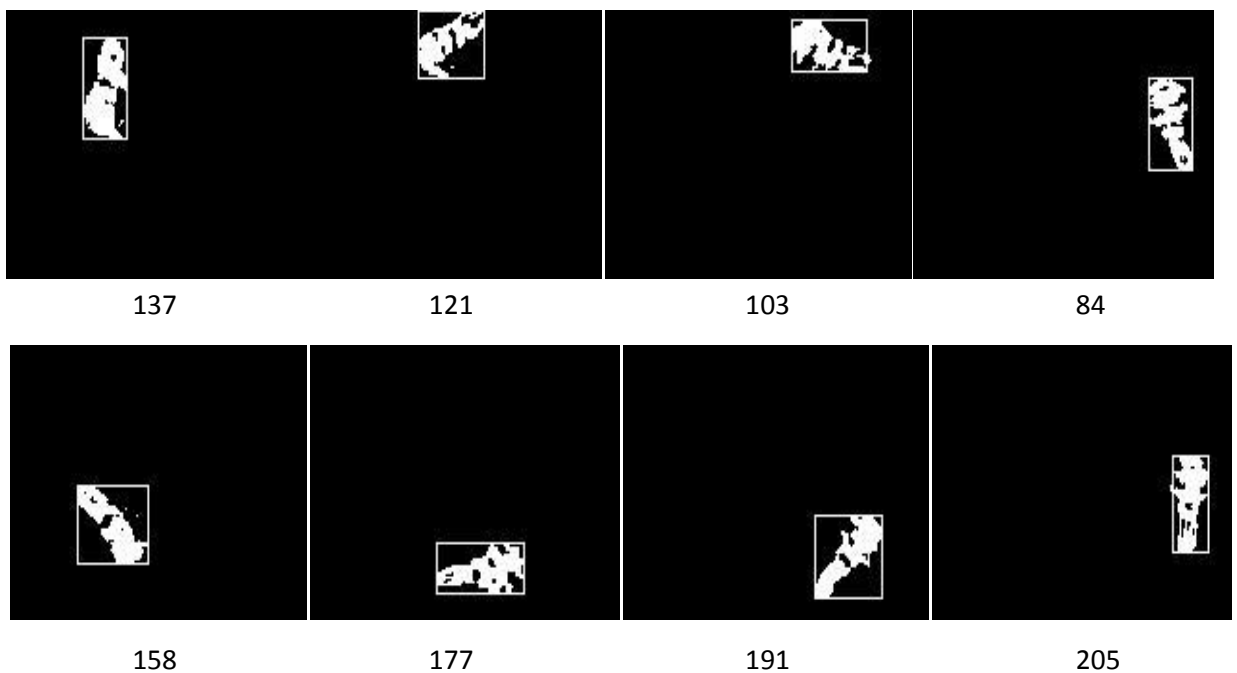


Figure (5.7) : Niveau gris de quelques trames

Les points en mouvement sont séparés des points statiques, les trames converties passent par l'opération de soustraction de fond pour créer une image de référence pour chaque image courante.

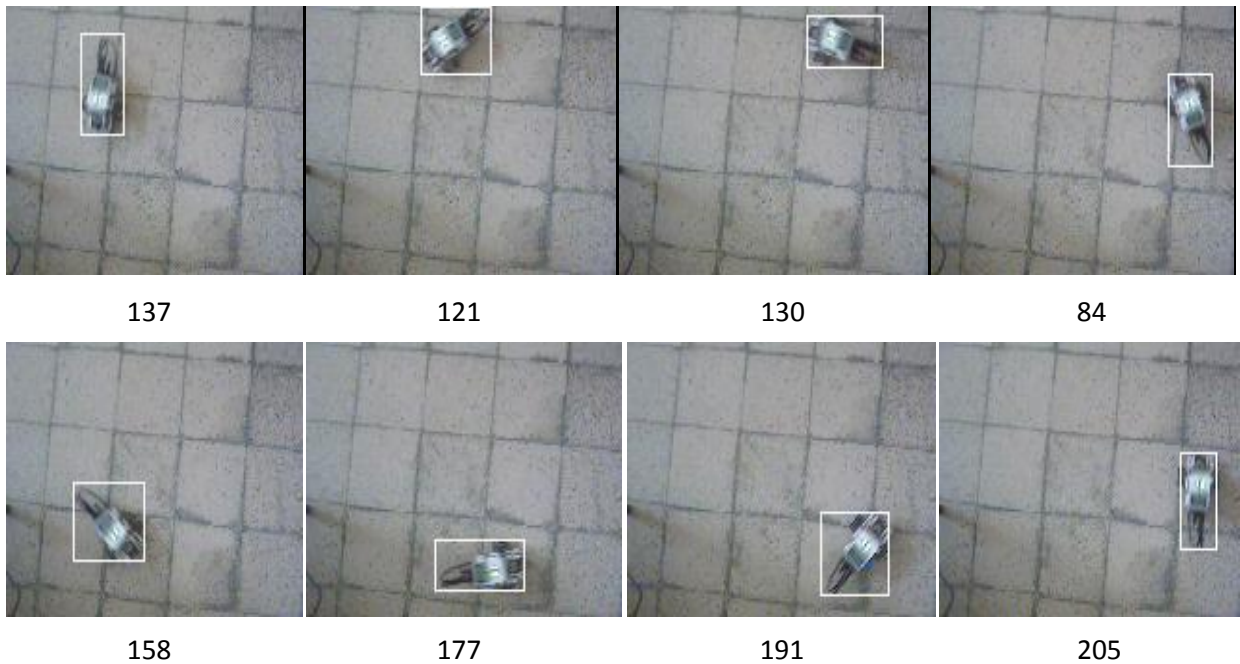


Figure (5.8) : Suivi du robot dans quelques trames

La figure (5.8) présente un exemple de détection utilisant l'image de différence sur la séquence, nous constatons que le robot en mouvement est assez bien détecté.

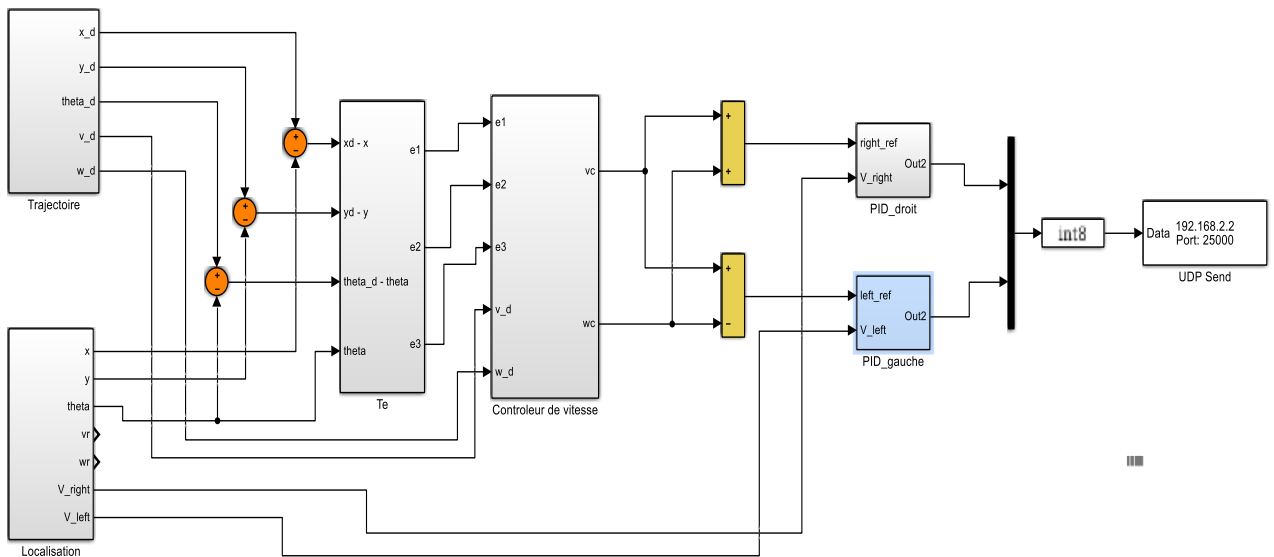
L'étape suivante consiste en un filtrage médian et un seuillage, avec leurs blocs correspondants « median filtre », « Autothreshold ». Ces derniers sont utilisés pour créer un masque binaire sur les images (noire et blanc). Sur cette image binaire, seuls les pixels du robot en mouvement sont représentés en blanc, le reste des pixels de l'image est en noir. Ainsi tout point extrait sur l'image en niveau de gris devrait être classé en mouvement si sa position correspond à un pixel blanc sur le masque, sinon il devrait être statique. C'est un moyen qui nous permet de valider les résultats de l'algorithme de la détection de mouvement.

A cette fin, les groupes de pixels connectés dans les images binaires sont analysés par le bloc (blob analyse) pour calculer les statistiques des régions marquées dans ces images binaires. Les informations sorties de ce bloc sont : le centre de masse, boîte englobante, et le nombre de blob.

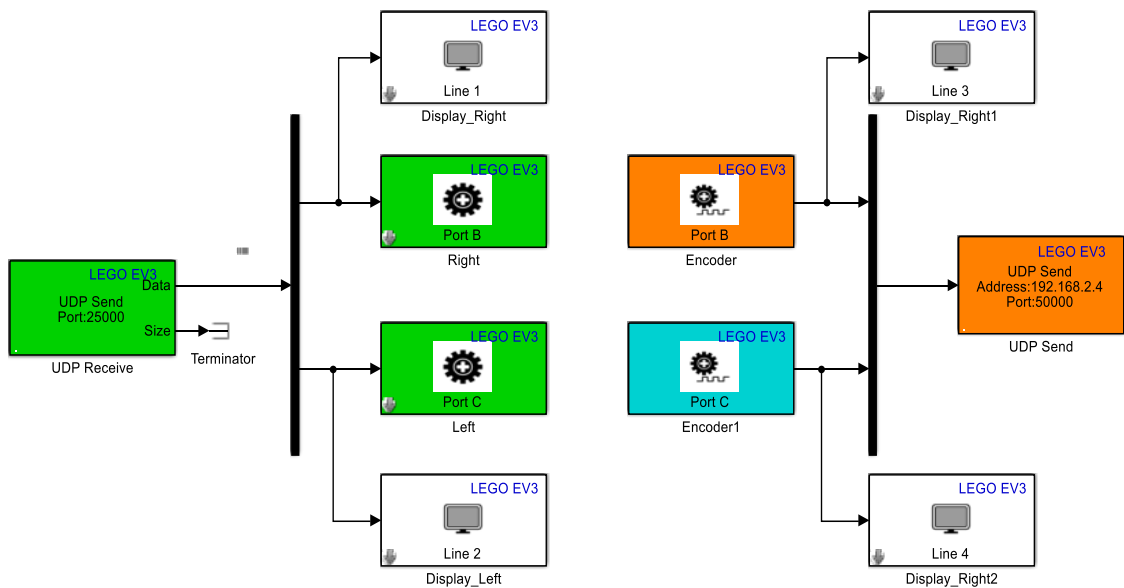
Ces informations sont utiles dans le suivi ou le tracking du robot mobile, afin de générer la trajectoire du robot en mouvement dans le temps en déterminant sa position et sa vitesse dans chaque image d'une séquence vidéo.

5.6. Système de commande

Les sorties du bloc de vision vont servir à chaque pas d'échantillonnage comme entrées au bloc de traitement et de calcul afin de minimiser les signaux des erreurs des deux contrôleurs numériques PI. Ceci permet d'obtenir le signal de commande du robot et exécuter correctement la trajectoire désirée. Figure (5.9) montre les modèles Simulink proposés pour cette partie de commande.



a. Programme PC



b. Programme robot

Figure (5.9) : Modèles Simulink proposés pour la commande

Le protocole de transmission de données entre la partie de vision et la partie de commande est assuré en utilisant les deux blocs de communication UDP Send et UDP Receive.

Dans cette partie expérimentale, nous avons réalisé un contrôleur cinématique (contrôleur haut niveau), il s'agit d'un contrôleur de vitesse pour générer le signal de vitesse V_c . Deux contrôleurs PI numériques sont intégrés en boucle fermée pour commander les deux roues motrices du lego EV3. Le module de traitement d'image est intégré en temps réel et synchronisé avec le module de traitement et calcul pour la décision de stratégie de commande. Des signaux de commande sont fournis aux deux moteurs à courants continus du lego qui agissent sur les deux roues afin de réaliser le déplacement nécessaire assurant la poursuite de la trajectoire. Les figures suivantes illustrent les résultats de poursuite pour deux types de trajectoires, rectiligne et circulaire.

a. Trajectoire rectiligne

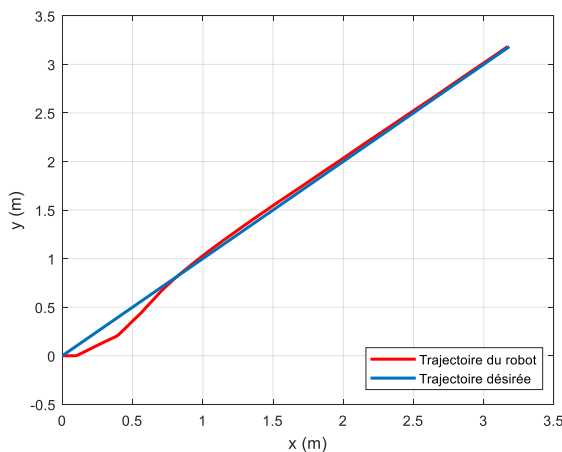


Figure (5.10) : Trajectoire rectiligne

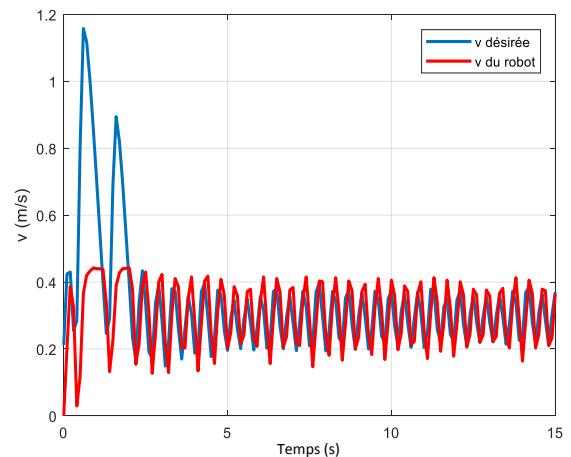


Figure (5.11) : Vitesses linéaires (m/s)

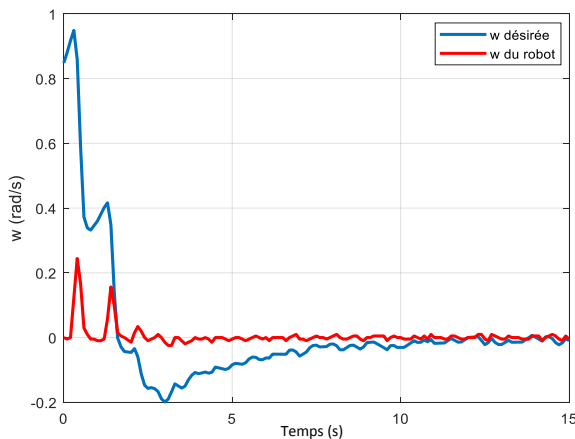


Figure (5.12) : Vitesse de rotation

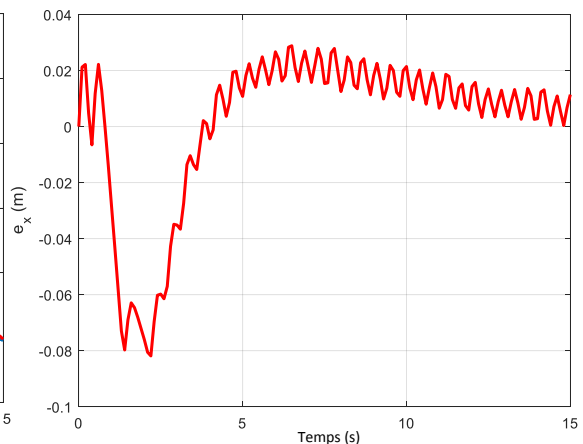


Figure (5.13) : Erreur sur x

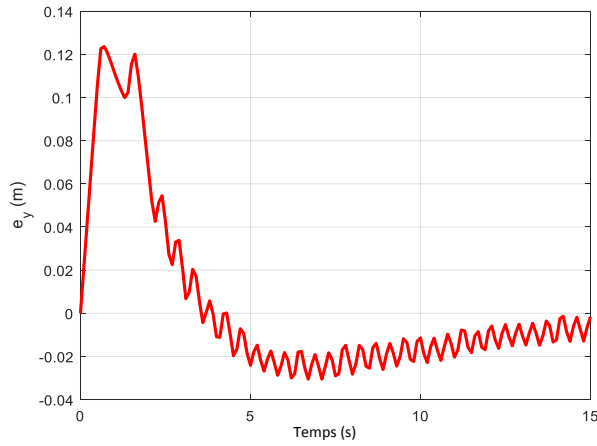


Figure (5.14) : Erreur sur y

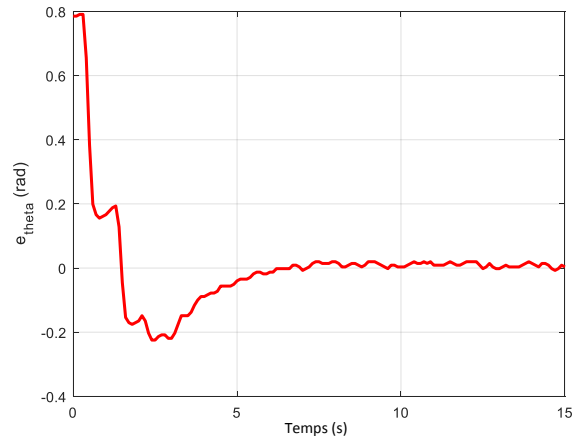


Figure (5.15) : Erreur d'orientation

b. Trajectoire circulaire

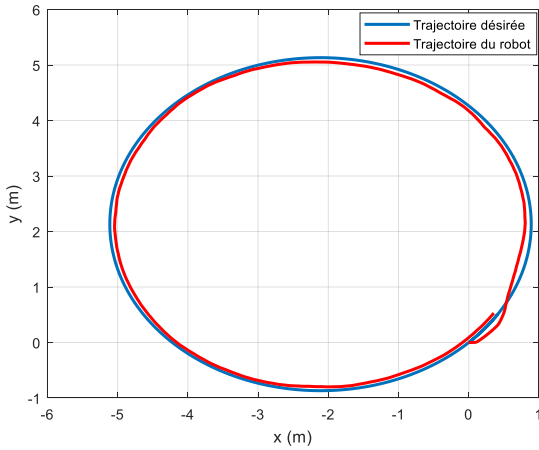


Figure (5.16) : Trajectoire circulaire

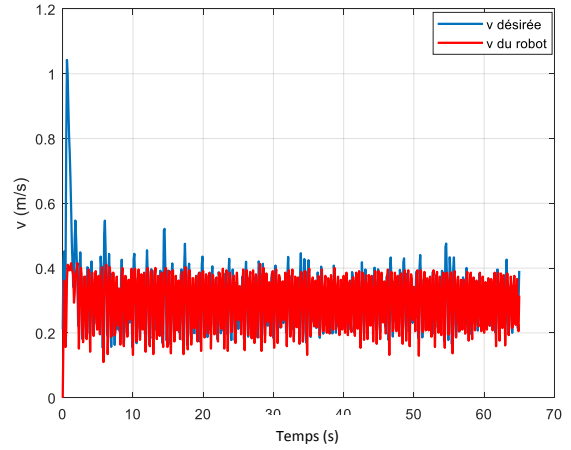


Figure (5.17) : Vitesses linéaires (m/s)

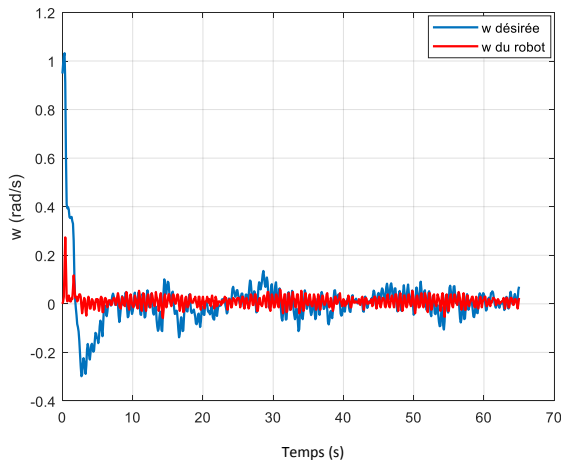


Figure (5.18) : Vitesses de rotation (rd/s)

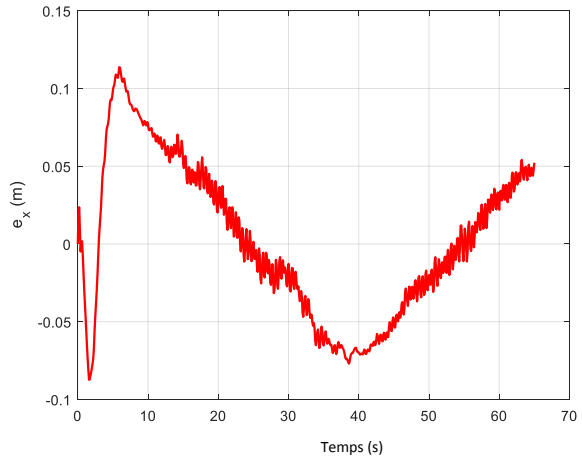


Figure (5.19) : Erreur sur x

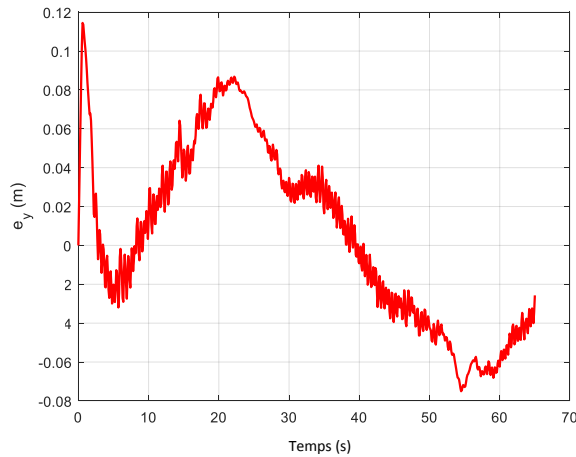


Figure (5.20) : Erreur sur y

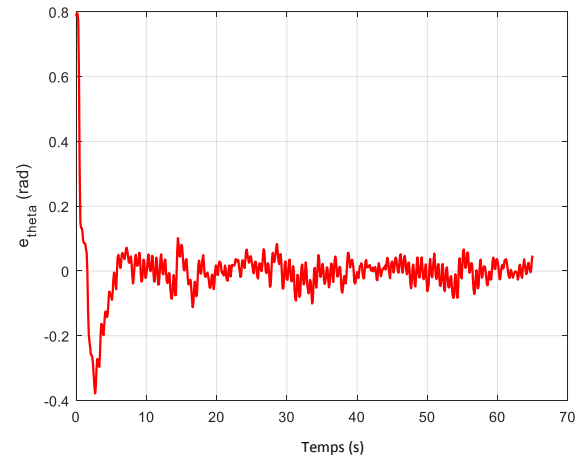


Figure (5.21) : Erreur d'orientation

Les résultats expérimentaux obtenus attestent la convergence de l'erreur de suivi vers zéro avec la structure de commande proposée dans un environnement réel où les mesures sont bruitées. Le robot a pu suivre la trajectoire désirée malgré les erreurs dynamiques. On remarque les bonnes performances de cette approche qui est basée sur le suivi d'objet pour traiter la localisation visuelle et faire déplacer le robot dans son environnement et que cette technique a pu diminuer les erreurs en régime permanent.

5.7. Conclusion

Nous avons proposé dans ce chapitre une approche qui résout le problème de suivi de trajectoire d'un robot mobile non holonome en présence de bruit dans un environnement réel. Cette approche est basée sur l'asservissement visuel. On a présenté une chaîne de traitement de vidéo à partir d'une mono caméra fixe, l'approche développée est basée sur la soustraction de fond pour la détection de mouvement et le suivi de la position des centres de masse, de la vitesse et également la trajectoire du robot. Nous constatons que le système de détection de mouvement peut réagir face aux conditions non contrôlées de luminosité et la détection à l'intérieur de l'environnement, les résultats obtenus sont satisfaisants et encourageants. On constate que l'approche adoptée dans ce travail, basée sur le suivi d'objet, a permis au robot d'exécuter un bon déplacement dans son environnement.

Conclusion générale

CONCLUSION GENERALE

Un robot mobile à roues est l'un des systèmes bien connus avec des contraintes non holonomes. Le travail effectué dans cette thèse a porté essentiellement sur l'élaboration de différents algorithmes de commande robuste pour la poursuite de trajectoire d'un robot mobile non holonome de type unicycle. Cette commande a pour objectif de forcer le robot mobile à atteindre la configuration d'arrivée à partir d'une configuration de départ avec un minimum d'erreurs et sans collision avec les obstacles et sans intervention humaine. Pour l'ensemble des algorithmes, nous sommes arrivés à établir une association de critères de robustesse et de stabilité conduisant à un contrôle simple et efficace d'un robot mobile.

Les approches élaborées se sont basées sur trois techniques, à savoir la méthode de backstepping qui est considérée comme un outil pour la conception récursive de la loi de commande basée sur la théorie de Lyapunov, le mode glissant qui permet d'acquérir un degré de précision, ensuite la logique floue pour son aptitude à prendre en considération la modélisation d'incertitudes.

En effet ce travail s'est articulé autour de cinq chapitres, le premier qui avait pour but de présenter un état de l'art sur la robotique mobile et son application. Une liste non exhaustive des techniques de commandes pour résoudre le problème de suivi de trajectoire des robots mobiles a été dressée dans ce chapitre. Par soucis de simplicité les méthodes de commande ont été développées principalement pour les robots de type unicycle. La plupart des résultats peuvent se généraliser en fait à d'autres robots mobiles, et en particulier au système avec remorques [2].

Dans le deuxième chapitre, nous avons exposé la théorie de commande par backstepping et l'extension de cette technique à la commande d'un robot mobile à roues.

Cette étude a permis l'élaboration d'un algorithme par backstepping appliqué pour contrôler par la suite le robot mobile.

Dans le troisième chapitre, nous avons exposé les concepts de base de la commande par mode glissant ainsi que ses avantages, simplicité d'implantation, robustesse vis à vis les perturbations. Le contrôleur présenté dans cette partie est basé sur une architecture hiérarchique répandue dans le domaine de recherche en robotique mobile. Il est en fait composé de deux contrôleurs assurant la commande du robot mobile à deux niveaux : un niveau haut et un niveau bas. Au niveau haut ou "niveau cinématique", nous avons exploiter la méthode de la poursuite stable vue sa simplicité. Au niveau bas, une conception du contrôleur en mode glissant basé sur la théorie de Lyapunov a été faite pour contrôler la dynamique du robot mobile et assurer un bon asservissement des vitesses des roues.

Nous avons évoqué également le principal inconvénient de ce type de commande à savoir l'existence d'un phénomène du chattering. Nous avons proposé donc dans le quatrième chapitre une commande hybride combinant la technique du mode glissant et l'approche par logique floue. Le choix de ces deux techniques a été motivé par les objectifs de stabilité et de robustesse qu'elles offrent. Notre travail s'est orienté vers une adaptation en temps réel des gains des erreurs de poursuite.

Le dernier chapitre a été consacré à une implémentation pratique d'un asservissement visuel pour la commande du robot mobile. L'application décrite dans ce dernier a porté essentiellement sur la poursuite d'objet dans l'espace, plus précisément la réalisation d'une conception de fonctionnalités visuelles pour un robot EV3 dans un milieu local. Cette étude concerne la commande cinématique du EV3, dont le capteur est une caméra fixe, elle fonctionne vis à vis les mesures visuelles et permet de tenir en compte des erreurs de mesures ou d'estimation de la position du robot et ainsi compenser l'effet de ces erreurs et des perturbations.

Les résultats obtenus montrent les bonnes performances des contrôleurs présentés dans les différents chapitres en termes d'erreur de poursuite et temps d'exécution de la poursuite des trajectoires.

Enfin, pour l'amélioration de ce travail l'implémentation pratique des approches présentées dans ce manuscrit seront des axes de recherches à explorer pour d'autre type de robots pour lesquels l'allée aux variables couple est possible, ce qui n'est pas le cas du robot EV3.

Annexes

ANNEXE A

THEOREME DE LYAPUNOV

Les théorèmes au sens de Lyapunov sont basés sur l'analogie avec les systèmes physiques, l'idée de base est que lorsque l'énergie d'un système décroît continuellement au cours du temps, il arrive en un moment où le système n'a plus d'énergie et s'arrête. Un des exemples très classiques, est celui d'une boule qui roule le long d'une pente, jusqu'à ce que son énergie totale (l'énergie cinétique + l'énergie potentielle) s'annule.

L'idée de Lyapunov consiste à essayer d'associer une fonction énergie à un système quelconque, de sorte que le même raisonnement puisse être appliqué pour déterminer la stabilité. La détermination de la bonne fonction d'énergie dans le cas des systèmes non linéaires demeure un problème difficile à résoudre. Il est important de signaler que si on ne sait pas trouver une fonction de Lyapunov, cela ne signifie pas que le système diverge.

D'une manière générale, un système dynamique est souvent représenté par un ensemble d'équations différentielles de la forme :

$$\dot{x} = F(x, u, t) \quad (\text{A.1})$$

Où $F : R^n \times R^m \times R \rightarrow R^n$ est une fonction continue, $x \in R^n$ représente l'état du système, $u \in R^m$ est la commande et $t \in R$ est le temps.

Pour les systèmes invariants (le temps ne figure pas explicitement dans les équations) :

$$\dot{x} = F(x, u) \quad (\text{A.2})$$

La définition de la loi de commande en fonction de l'état courant $u(x)$ permet d'écrire le système (A.2) comme suit

$$\dot{x} = F(x) \quad (\text{A.3})$$

On note $B_{R_0} = \{x \in R^n / \|x\| < R_0, R_0 > 0\}$: boucle ouverte de rayon R_0 dans R^n .

Théorème 1 (stabilité locale) : Si dans B_{R_0} , il existe une fonction scalaire différentiable $V(x)$ telle que :

1. $\forall x \in B_{R_0} \setminus \{0\}, V(x) > 0, \text{ et } V(0) = 0$
2. $\forall x \in B_{R_0} \setminus \{0\}, \dot{V}(x) \leq 0, \text{ et } \dot{V}(0) = 0$

Alors le point d'équilibre $x = 0$ du système (A.3) est stable.

$$3. \text{ Si } \forall x \in B_{R_0} \setminus \{0\}, \dot{V}(x) < 0$$

alors la stabilité est asymptotique.

Une fonction qui remplit les conditions (1) et (2) est appelée fonction de Lyapunov.

Théorème 2 (stabilité globale) : Supposons qu'il existe une fonction scalaire et différentiable $V(x)$ telle que :

1. $\forall x \in R^n \setminus \{0\}, V(x) > 0$, et $V(0) = 0$
2. $\forall x \in R^n \setminus \{0\}, \dot{V}(x) < 0$, et $\dot{V}(0) = 0$
3. $V(x) \rightarrow \infty$, lorsque $\|x\| \rightarrow \infty$

Alors le point d'équilibre $x = 0$ du système (A.3) est globalement asymptotiquement stable.

La figure (A.1) illustre graphiquement une fonction de Lyapunov, permettant d'assurer la convergence globale d'un système.

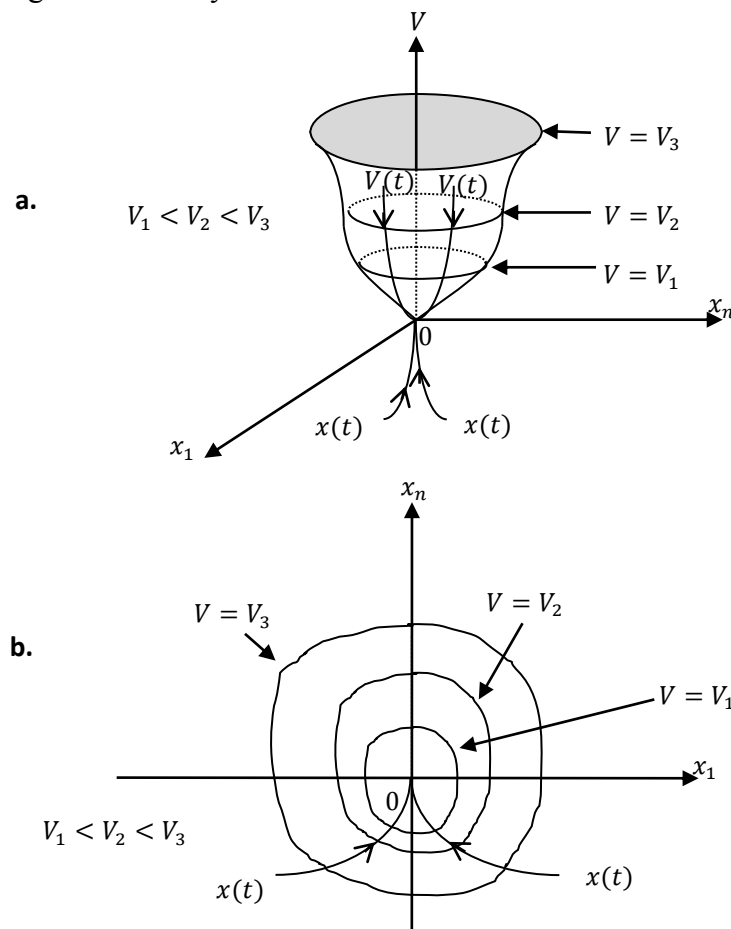


Figure (A.1) : Exemple graphique d'une fonction de Lyapunov

(a) : Représentation de $V(x)$, (b) : Projection de $V(x)$ sur l'espace d'état

ANNEXE B**PARAMETRES DU ROBOT**

Les paramètres du robot utilisé dans la simulation sont les suivants :

Paramètre	Description	Valeur nominale
m	masse du robot mobile à roue	4 kg
I	inertie du robot mobile à roue autour de son centre	2.5 kg m ²
$2L$	distance entre les deux roues motrices	0.15 m
r	rayon des roues	0.03 m

ANNEXE C

ROBOT LEGO MINDSTORMS EV3 THEOREME: LES COMPOSANTS PRINCIPAUX

La brique de commande EV3

Caractéristiques :

- + Système d'exploitation – LINUX
- + Processeur ARM9 300 MHz
- + Mémoire flash – 16 Mo
- + Mémoire vive – 64 Mo
- + Résolution de l'écran – 178x128/noir & blanc

1 : Bouton de sortie
- sortir d'une action
- arrêter le programme
- éteindre la brique

Pour éteindre la brique : 1=>3=>2.

2 : Bouton de centre
- allumer la brique
- sélectionner
- valider

3 : Boutons de navigation

4 Ports de sortie (moteurs)
Connexion usb au PC (télécharg.)

Port USB (Mise en série , clé wifi)
Port carte SD (extension mémoire 32Go max)

Haut-parleur

4 Ports d'entrée (capteurs)

Les composants principaux

Gros moteur
+ Pour programmer les actions du robot avec précision et puissance.

Brique EV3
+ Centre de commande et énergétique du robot.

Capteur tactile
+ Reconnait trois états : enfoncé, heurté et relâché.

Moteur moyen
+ Un moteur qui reste précis, mais qui abandonne un peu de puissance au bénéfice de la compacité et de la rapidité de réaction.

Balise / Télécommande infrarouge
+ Télécommande du robot.

Capteur de couleur
+ Reconnait sept couleurs et mesure la luminosité.

Capteur infrarouge
+ Détecte des objets ; peut suivre et trouver la Balise / Télécommande infrarouge.

Bibliographie

BIBLIOGRAPHIE

- [1] D. Filliat, "Robotique mobile," Fondation UNIT et GDR robotique, 2012.
- [2] J. P. Laumond, "La robotique mobile," Edition Hermes Science Publication, Paris, 2001.
- [3] L. E. A. Melo, "Commande robuste et coordination de mouvements de robots mobiles," Thèse de doctorat, Université Paul Sabatier de Toulouse, 1997.
- [4] P. Alain, "Robotique mobile : la planification de trajectoire (coll. Série robotique)," Edition Hermes Science Publications/Lavoisier, 1996.
- [5] R. Siegwart and I. R. Nourbakhsh, "Introduction to autonomous mobile robots," A Bradford Book, the MIT Press, London, 2004.
- [6] B. Bayle, "Robotique mobile," Cours option ISAV, Télécom Physique Strasbourg 3A.
- [7] F. Morbidi, "Localisation et navigation des robots," Cours M2 EEAI, Université de Picardie Jules Verne, Paris, 2015.
- [8] E. Guechi, "Suivi de trajectoires d'un robot mobile non holonome : approche par modèle flou de Takagi-Sugeno et prise en compte des retards," Thèse de doctorat, Université d'Annaba, Algérie, 2010.
- [9] P. Sabatier, "La Robotique une science aux multiples facettes," Le magazine scientifique, n° 7, pp. 4.
- [10] A. Benzerrouk, "Architecture de contrôle hybride pour les systèmes multi-robot mobiles," Université Blaise Pascal-Clermont II, 2011.
- [11] K. Nauman and A. Mateen, "Emerging Aspects of robotics technology," Int. J. of Computer Application, vol. 27, n° 9, 2011.
- [12] D. Filliat, "Robotique mobile," Cours, Ecole nationale supérieure de techniques avancées, Paris Tech, 2011.
- [13] L. Matignon, "Introduction à la robotique," Cours, GREYC-CNRS, Université de Caen, France, 2012.
- [14] B. Fatma, "Contribution des outils de l'intelligence artificielle dans la robotique mobile," Thèse de doctorat, Université d'Oran, 2014.
- [15] K. Benchaiba, "Commande de la machine asynchrone par mode de glissement," Mémoire de magister, université de Batna, Algérie, 2003.
- [16] J. M. Yang and J. H. Kim, "Sliding mode control for trajectory tracking of nonholonomic wheeled mobile robots," IEEE Trans. on Robotics and automation, vol. 15, n° 3, pp. 578-587, 1999.
- [17] O. Mohareri, R. Dhaouadi and A. B. Rad, "Indirect adaptive tracking control of a nonholonomic mobile robot via neural networks," Neurocomputing 88, pp 54-66, 2012.
- [18] C. Z. Resende, R. Carelli and M. S. Filho, "A nonlinear trajectory tracking controller for mobile robots with velocity limitation via fuzzy gains," Control Engineering Practice, vol. 21, pp. 1302-1309, 2013.
- [19] P. Shahmaleki, M. Mahzoon and V. Shahmaleki, "Designing fuzzy controller and real time experimental studies on a nonholonomic robot," 12th IFAC Symposium on Transportation

- Systems, Redondo Beach, USA, pp. 312-319, 2009.
- [20] D. N. M. Abadi a and M. H. Khooban, “Design of optimal Mamdani-type fuzzy controller for nonholonomic wheeled mobile robots,” *Journal of King Saud University, Engineering Sciences*, pp. 92-100, 2015.
- [21] Y. N. Wang, T. Mai and J. X. Mao, “Adaptive motion/force control strategy for non-holonomic mobile manipulator robot using recurrent fuzzy wavelet neural networks,” *Engineering Applications of Artificial Intelligence*, pp. 137-153, 2014.
- [22] K. K. Young, “Controller design a manipulator using theory of variable structure systems,” *IEEE Trans. Syst. Man. Cybe.*, vol. 8, pp. 101-109, 1978.
- [23] M. R. Azizi and J. Keighobadi, “Robust sliding mode trajectory tracking controller for a nonholonomic spherical mobile robot,” 19th world congress, Int. Federation of Automatic Control, South Africa, 4541-4546, August, 2014.
- [24] J. H. Yang and C. C. Yang, “Adaptive sliding mode control for vehicle brake systems with fuzzy composition,” *Int. J. Modelling, Identification and Control*, vol. 4, n°. 2, 2008.
- [25] J. Velagic, B. Lacevic and N. Osmic, “Nonlinear motion control of mobile robot dynamic model,” University of Sarajevo Bosnia and Herzegovina, pp 529-550, 2008.
- [26] T. Fukao, H. N., and N. Adachi, “Adaptive tracking control of a nonholonomic mobile robot,” *IEEE transactions on robotics and automation*, vol 16, n° 5, October, pp 609-615, 2000.
- [27] C. Angchen, T. Hsengs.Li, Y. Chienhyeh and C. C. Chang, “Design and implementation of an adaptive sliding-mode dynamic controller for wheeled mobile robots,” *Mechatronics*19, pp 156–166, 2009.
- [28] J. G. A. Cervantes, “Navigation visuelle d'un robot mobile dans un environnement d'extérieur semi-structur,” Thèse de doctorat, l'Institut National Polytechnique de Toulouse, 2005.
- [29] A. M. Hernandez, “Vision dynamique pour la navigation d'un robot mobile,” Thèse de doctorat, l'Institut National Polytechnique de Toulouse, 2004.
- [30] Y. Zinbi, “ Detection etcategorisation d'objets en mouvement dans une vidéo,” Thèse de doctorat, Université de Caen basse Normandie, 2009.
- [31] A. Dib, “Commande non linéaire et asservissement visuel de robots autonomes,” Thèse de doctorat, Automatique/Robotique. Supélec, 2011.
- [32] E. Hashemia, M. G. Jadidi and N. G. Jadidi, “Model based PI fuzzy control of four wheeled omni-directional mobile robots,” *Robotics and Autonomous Systems* 59, pp 930–942, 2011.
- [33] S. Noureddine, “Système de localisation pour robots mobiles,” Thèse de doctorat, Université de Batna, Algérie, 2005.
- [34] M. E. M. Mendez, “Commande référencée capteur des robots non holonomes,” Thèse de doctorat, Inria Sophia-Antipolis, 2007.
- [35] B. Thuilot, “Contribution à la modélisation et à la commande de robots mobiles à roues,” Thèse de doctorat, Ecole nationale supérieure des Mines, Paris, 1995.
- [36] R. Mermond, “Planification de chemins pour un robot non holonome sous des contraintes d'incertitudes géométriques,” Thèse de doctorat, Université de Joseph Fourier, Grenoble, 1996.
- [37] M. J. Aldon, “Capteurs et méthodes pour la localisation des robots mobiles,” UMR CNRS, Université de Montpellier II, .
- [38] S. Lens, “Locomotion d'un robot mobile,” Université de Liège, Montèore, 2008.
- [39] T. Peynot, “Sélection et contrôle de modes de déplacement pour un robot mobile autonome

- en environnements naturels,” institut national polytechnique de Toulouse, 2006.
- [40] G. Zidani, “Exécution de trajectoire pour robot mobile d’intérieur -réseaux de neurones-,” Mémoire de magister, université de Batna, Algérie, 2009.
- [41] R. Siegwart and I. R. Nourbakhsh, “Introduction to autonomous mobile robots,” Bradford Book, A MIT Press Cambridge, Massachusetts, London, England, 2004.
- [42] K. Goris, “Autonomous mobile robot mechanical design,” thesis engineer, Vrije University, Brussel, 2005.
- [43] F. Pourboghrat and M. P. Karlsson, “Adaptive control of dynamic mobile robots with nonholonomic constraints,” *Computers and Electrical Engineering*, pp. 241–253, 2002.
- [44] Y. T. Wang, Y. C. Chen and M. C. Lin, “Dynamic object tracking control for a non holonomic wheeled autonomous robot,” *Tamkang Journal of Science and Engineering*, vol. 12, pp. 339-350, 2009.
- [45] M. A. Elkamel, “Stabilisation et régulation des robots mobiles opérant en groupe,” Thèse de doctorat, Université d'Evry-Val d'Essonne, France, 2012.
- [46] M. Krstie and P. V. Kokotovic, “Nonlinear and adaptive control design,” *Proceeding of new York*, Wiley, 1995.
- [47] P. V. Kokotovic, “The joy of feedback : Nonlinear and adaptive,” *IEEE Control Systems Magazine*, vol. 12, pp. 7-17, Jun. 1992.
- [48] A. Benchaib, "Application des modes de glissement pour la commande en temps réel de la machine asynchrone", Thèse de doctorat, Université de Picardie Jules Verne, 1998.
- [49] F. Eshbair, “Modélisation et commande d’un système multi-moteur par la technique de commande backstepping,” Mémoire présenté à l’université du Québec, Aout 2005.
- [50] H. Tédjini, M. Moudjahed, Y. Meslem and M.Tadjine, “Commande non linéaire adaptative par backstepping et observateur d'une machine synchrone à aimant permanent alimentée par un onduleur à trois niveaux,” 4th International Conference on Computer Integrated Manufacturing CIP’2007, November 2007.
- [51] L. Marce, M. Julierz and H. Place, “ Stratégie de contournement d’obstacles pour un robot mobile,” *Rairo*, vol. 15, n°. 1, 1981.
- [52] A. R. Benaskeur, “Aspects de l'application du backstepping adaptatif à la commande décentralisée des systèmes non linéaires,” Thèse de doctorat (Ph.D), Université Laval, Québec, 2000.
- [53] I. Boulkaibet, “Une étude en simulation de stratégie de commande non linéaire,” Thèse de magister, Université Mentouri, Constantine, 2000.
- [54] G. Didier, “Modélisation et diagnostic de la machine asynchrone en présence de défaillances,” Thèse de doctorat, Université Henri Poincaré, Nancy-I, Octobre 2004.
- [55] M. M. M. Ezzat, “Commande non linéaire sans capteur de la machine synchrone à aimants permanents,” Thèse de doctorat, Ecole Centrale de Nantes, 2011.
- [56] F. Mehazzem, “Contribution à la commande d’un moteur asynchrone destiné à la traction électrique,” Thèse de doctorat, Université Paris-Est, 2010.
- [57] F. Mehazzem, A. Reama, Y. Hamam and H. Benalla, “Real time implementation of backstepping controller in indirect field oriented control of induction motor drive,” *Second International Conference on Power Engineering, Energy and Electrical Drives*, Lisbon, Portugal, , March 2009.
- [58] C. Angchen, T. H. Li, Y. Chienhyeh and C. C. Chang, “Design and implementation of an adaptive sliding mode dynamic controller for wheeled mobile robots”, *Mechatronics*, pp

- 156–166, 2009.
- [59] M. Y. Hsiao, C. Y. Chen, S. H. Tsai and S. T. Liu, “Combined interval type-2 fuzzy kinematic and dynamic controls of the weeledmobile robot with adaptive sliding mode technique”, Fuzz-IEEE, Korea, August 20-24, pp 706-711, 2009.
- [60] L. S. Wang, F. R. Chang and P. S. Tsai, “Continuation method of backstepping tracking control for car-like mobile robots”, IEEE International conference on systems, Man and cybernetics, Taipei, Taiwan, October 8-11, pp 2018-2023, 2006
- [61] Y. Kanayama, Y. Kimura, F. Miyazaki and T. Noguchi, “A Stable Tracking Control Method for an Autonomous Mobile Robot”, Mechatronics, pp 384-389, 1990.
- [62] H. Tamrbed et T. Azarobr “Réglage par mode glissant d’un machine asynchrone,” proceedings de la conférence CEE’00, pp.136-141, 2000.
- [63] V. I. Utkin, “Variable structure systems with sliding mode,” IEEE.TAC, vol. AC-22, n° 2, p.212-222, 1977.
- [64] H. Bühler, “Réglage par mode glissant,” Edition presses polytechnique Romandes, 1986.
- [65] M. Manceur, “Commande robuste des systèmes non linéaires complexes,” Thèse de doctorat, Université de Reims, 2011.
- [66] T. Floquet, “Contribution à la commande par modes glissants d’ordre supérieur,” Thèse de doctorat, Université des Sciences et Technologies de Lille, 2000.
- [67] H. Benderradji, “Contribution à la Commande Robuste de la Machine à Induction,” Thèse de doctorat, Université de Batna, 2013.
- [68] G. Zidani, S. Drid, L. C. Alaoui and D. Arar, “Nonlinear Tracking Control of a Wheeled Mobile Robot,” the 15th International conference on Sciences and Techniques of Automatic control & computer engineering, STA’2014, Hammamet, Tunisia, 2014.
- [69] C. Z. Resende, R. Carelli and M. Sarcinelli-Filho, “A nonlinear trajectory tracking controller for mobile robots with velocity limitation via fuzzy gains,” Control Engineering Practice, vol. 21, pp. 1302-1309, 2013.
- [70] T. Fukao, H. Nakagawa, and N. Adachi, “Adaptive tracking control of a nonholonomic mobile robot,” IEEE transactions on robotics and automation, vol 16, n° 5, October, pp 609-615, 2000.
- [71] J. Ye, “Adaptive control of nonlinear PID-based analog neural networks for a nonholonomic mobile robot,” Neurocomputing 71, pp 1561-1565, 2008.
- [72] A. M. Joseph, “La logique floue et ses applications industrielles,” Quaderni, pp. 75-86, 1995.
- [73] B. Meunier, “La Logique Floue et ses Applications,” Edition Addison-Wesley, France, Décembre 1994.
- [74] H. Bühler, “Réglage par logique floue,” Edition Presse Polytechnique et Universitaire Romanie, 1994.
- [75] P. Borne, J. Rozinoer, J. Y. Dieulot et L. Dubois, “Introduction à la commande floue,” Edition Technip, Paris, 1998.
- [76] G. Zidani, S. Drid, L. C. Alaoui, D. Arar and P. Bussy, “Robust nonlinear control of a mobile robot,” Journal of Electrical Engineering & Technology, vol. 11 n°4, pp. 1921-718, 2016.
- [77] Tong-Tong, Jean-Raphaël “La logique floue,” Edition Herms, Paris, 1995.
- [78] J. Godjevac, “Idées nettes sur la logique floue,” Edition Presses Polytechniques et Universitaires Romandes, Suisse 1999.
- [79] J. Godjevac, “Fuzzy logic system for engineering,” Proceedings IEEE, vol. 83, n°3, 1995.

- [80] C. C. Lee, "Fuzzy logic in control system: Fuzzy logic controller," part II, IEEE, Trans. Syst. Man and Cyber, vol. 20, n°2, 1990.
- [81] L. A. Zadeh, "Fuzzy sets," information and Control, vol. 8, pp. 338-353, 1965.
- [82] S. Benaïcha, "Contribution à la commande tolérante aux défauts d'un système à motorisation asynchrone : Apport de l'intelligence artificielle pour l'aide à la supervision et à la décision," Thèse de doctorat, Université de Batna, 2010.
- [83] F. Zidani, "Réglage flou " Cours de doctorat Dépt. électrotechnique, Université de Batna.
- [84] M. Diab, "Contrôle flou des processus biotechnologiques à base d'algorithmes génétiques," thèse de doctorat, Université de Sétif, 2010.
- [85] M. T. Cao, "Commande numérique de machines asynchrones par logique floue," Thèse de doctorat (Ph.D), Université de Laval, 1997.
- [86] G. Zidani, S. Drid, L. C. Alaoui, A. Benmakhlouf and S. Chaouch, "Back-stepping Controller for a Wheeled Mobile Robot," 4th International Conference on Systems and Control (ICSC'15), Sousse, Tunisia, April 28-30, 2015.
- [87] S. Benaïcha, R. N. Said, F. Zidani and M. N. Said, "Fuzzy fault tolerant control of an induction motor," Int. J. Modelling, Identification and Control, vol. 4, n°2, 2008.
- [88] L. Marín, M. Vallés Miquel, A. Soriano Viguera, A. Valera Fernández and P. Albertos, "Event-based localization in ackermann steering limited resource mobile robots," IEEE/ASME Transactions on Mechatronics 19, vol 4, pp. 1171-1182, 2014.
- [89] T. Penne, "Développement d'un système de tracking vidéo sur caméra robotisée," Thèse de doctorat, Université de Blaise Pascal-Clermont-Ferrand II, 2011.
- [90] T. Malavika and M. Poornima, "Moving object detection and velocity estimation using MATLAB," Int. J. of Engineering Research & Technology (IJERT), vol. 2, n°2, pp. 1-4, 2013.
- [91] Y. Raoui, "Indexation d'une base de données images : Application à la localisation et la cartographie fondées sur des radio-étiquettes et des amers visuels pour la navigation d'un robot en milieu intérieur," Thèse de doctorat, Université de Toulouse, 2011.
- [92] L. Biancardini, S. Beucher et L. Letellier, "Extraction des objets en mouvement: une approche mixte contours-régions," CEA LIST Saclay France.
- [93] J. Murgia, C. Meurie et Y. Ruichek, "Vers une meilleure détection des objets mobiles par Codebook et invariance couleur," CORESA 2013, COMpression et REprésentation des Signaux Audiovisuels, Nov 2013, France. 6p, 2013.