



Université ABBES LAGHROUR Khenchela

Faculté des Sciences et de la Technologie

Département de Génie Industriel



جامعة عباس لغرور خنشلة
كلية العلوم والتكنولوجيا
قسم الهندسة الصناعية

N° Série :

Mémoire de fin d'étude

Pour l'obtention du diplôme de Master

Filière : Automatique

Spécialité: Automatique et informatique industrielle

THEME

**Conception d'un contrôleur basé sur
les réseaux de neurones pour le suivi
de trajectoire d'un robot mobile**

Réalisé par :

- Mohssen Mabrouk Abdulljabbar

Soutenu le / 07/2021 Devant le jury composé de:

BERKANI Abdelhakim

Président

Université Abbes Laghrou-Khenchela

FRIHA Souad

Examineur

Université Abbes Laghrou-Khenchela

MALLEM Ali

Encadreur

Université Abbes Laghrou-Khenchela

Promotion 2020/2021

Remerciement

Tout d'abord, je remercie le Bon Dieu le tout puissant pour la bonne santé, la volonté et la patience qu'il m'a donné pour accomplir ce travail.

J'exprime ma gratitude et mes sincères remerciements à Monsieur MAALEM ALI, qui a assuré la direction de mon mémoire de master, pour son suivi et ses conseils judicieux et pour le temps qu'il a su me consacrer.

Mes remerciements vont également à tous les enseignants ainsi qu'à tous les responsables de notre département qui ont veillé au bon déroulement de notre formation.

J'exprime ma profonde reconnaissance aux membres du jury qui ont accepté de juger ce mémoire.

Merci à vous Papa, Maman, et toute ma famille, Merci pour votre soutien, vos encouragements.

Je souhaite aussi dire un merci à mon amie Boudjij Rania que je m'aider, et à tous mes amis pour leur soutien.

Dédicace

Avant tous, je remercie dieu le tout puissant de m'avoir donné le courage et la patience pour réaliser ce travail malgré toutes les difficultés rencontrées « Covid-19 ».

Je dédie ce travail :

A Mon père, et Ma mère,

A Mes frère,

A Toute la famille,

A Tous mes amis.

Et À mon encadreur Mr. MALLEM Ali.

Mohssen Mabrouk

Résumé

Dans notre travail, des différentes techniques de commande d'un robot mobile sont proposées pour le suivi de trajectoire des robots mobiles, dont la dynamique du robot est prise en considération. Des commandes cinématiques sont appliquées où les résultats obtenus montrent l'efficacité de ces dernières, la commande en mode glissant rapide (FTSM) permet au robot de suivre sa trajectoire en temps fini, la commande backstepping présente une bonne convergence avec un bon choix des gains. La commande dynamique basée sur un contrôleur neuronal montre la robustesse de la commande en présence des perturbations qui influent sur la convergence du système. L'objectif de ce travail est de mettre le robot suivre sa trajectoire dans un temps fini et de garder la stabilité du système, la fonction de Lyapunov est proposée pour assurer cette dernière. Le bon choix des surfaces de glissements et les gains de la commande backstepping nous permettent d'obtenir des bons résultats.

Mot-clé :

Robot mobile- modèle cinématique- modèle dynamique -mode glissant rapide-commande backstepping –Lyapunov.

Abstract

In our work, different techniques of control of a mobile robot are proposed for the trajectory tracking of mobile robots, whose dynamics of the robot is taken into consideration. Kinematic controls are applied where the results obtained show the effectiveness of the latter, The Fast Sliding Mode (FTSM) control allows the robot to follow its trajectory in finite time; the backstepping control presents a good convergence with a good choice of gains. Dynamic control based on a neuronal controller shows the robustness of the control in the presence of disturbances that affect the convergence of the system.

The objective of this work is to put the robot to follow its trajectory in a finite time and to keep the stability of the system; the Lyapunov function is proposed to ensure it. The right choice of sliding surfaces and the gains of the backstepping command allow us to obtain good results.

Keywords:

Mobile robot- kinematic model- dynamic model -fast-sliding-command- mode backstepping –Lyapunov.

SOMMAIRE

Dédicace	
Remerciement	
Résumé	
Sommaire	
Abréviations	
Liste des figures	
Liste des tableaux	
Introduction générale	02

CHAPITRE I MODELISATION DES ROBOTS MOBILES

I.1 Introduction	04
I.2. Modélisation d'un robot mobile de type unicycle	04
I.2.1 Modélisation Cinématique	05
I.2.2 Modélisation Dynamique	10
I.3 Modélisation d'un robot mobile de type tricycle	16
I.3.1 Modélisation Cinématique	17
I.3.2 Modélisation Dynamique	20
I.4 Modélisation d'un robot mobile de type voiture	23
I.4.1 Modélisation Cinématique	24
I.4.2 Modélisation Dynamique	25
I.5 Conclusion	27

CHAPITRE II ELABORATION DES DEFERENTES COMMANDES APPLIQUEES A LA POURSUITE DE TRAJECTOIRE

II.1 Introduction	28
II.2 Conception du contrôleur pour le suivi de trajectoire	28
II.3 Commande par mode glissant (CMG)	29

II.3.1 Principe de la commande par modes de glissement	29
II.3.2 Conception de la commande par mode glissement	30
II .3.2.1 Choix de la surface de glissement	30
II .3.2.2 Principe d'invariance	31
II .3.2.3 Conditions de convergence et d'existence	31
II.3.3 Synthèse de la commande	32
II.3.3.1 Méthode de Philippov	33
II.3.3.2 Méthode de la commande équivalente d'Utkin	33
II.3.4 Calcul de la commande	34
II.4 Commande cinématique basée sur la commande par mode de glissement en temps fini	36
II.4.1 Problème de suivre la commande	38
II.4.2 la stratégie de suivi la commande	38
II.4.3 la conception de commande en cascade	38
II.4.4 Conception de loi de commande angulaire basée sur la technique de commande en temps fini	40
II.4.5 Conception de la loi de commande de la vitesse à base de commande en mode glissant et en temps fini	42
II.4.6 L'analyse de la stabilité	44
II.4.7 Les Résultats et simulations	46
II.5. Commande dynamique appliquée à la poursuite de trajectoire	48
II.5.1 Commande dynamique PID –backstepping	48
II.5.1.1 Analyse de la stabilité du Lyapunov	51
II.5.1.2 Résultats de simulation	52
II.5.2 Commande dynamique PID –mode glissant rapide (PID-FTSM)	55
II.5.2.1 Conception d'un contrôleur en mode glissant rapide	55
II.5.2.2 Commande dynamique basée sur un contrôleur PID	57
II.5.2.3 Résultats de simulation	58

II.6 Conclusion	61
-----------------	----

CHAPITRE III

LES RESEAUX DE NEURONES

III.1 Introduction	62
III.2 Définition d'un réseau de neurones	62
III.3 Neurone formel	62
III.3.1 Description mathématique	63
III.3.2 Architecture des réseaux de neurones	64
III.3.2.1 Réseaux de neurones multicouches (MLP)	64
III.3.2.2 Réseaux de neurones à fonction de base radiale (RBF)	65
III.4 Apprentissage des réseaux de neurones	68
III.4.1 Les Règles d'apprentissage	68
III.4.2 Les méthodes d'apprentissage	69
III.5 La Propriété fondamentale des réseaux de neurones	70
III.6 L'application des réseaux de neurones	70
III.7 Conclusion	70

CHAPITRE IV

CONCEPTION D'UN CONTROLEUR BASE SUR LES RESEAUX DE NEURONES

IV.1 Introduction	71
IV.2 Conception d'un contrôleur neuronal	72
IV.3 Application d'une commande NN-GFSM-backstepping pour la poursuite de trajectoires pour un robot mobile	73
IV.3.1. Contrôleur cinématique	74
IV.3.2. Contrôleur dynamique	74
IV.3.2.1. Contrôle en absence de perturbations	75
IV.3.2.2. Contrôle en présence de perturbations	76
IV.3.3. Résultats de simulations	83

IV.4 Conclusion	86
Conclusion générale	90
Bibliographie	

LISTE DES ABREVIATIONS :

- PID :** Proportionnelle intégral dérivé.
- MC:** Le mode convergence.
- MG:** Le mode glissement.
- CSV:** La commande à structure variable.
- TSM:** la commande par mode glissant terminal.
- FTSM :** la commande par mode glissant terminal rapide.
- RNA:** Réseau de neurone artificiel.
- RBF:** Radial Basis Fonctions.
- MLP:** Multi layer perceptron.

Chapitre I :

Figure I.1 Robot mobile de type unicycle	05
Figure I.2 Robot mobile de type unicycle dans les repères fixe	05
Figure I.3 Caractérisation du roulement sans glissement	06
Figure I.4 Robot mobile de type tricycle	17
Figure I.5 Robot mobile de type tricycle	17
Figure I.6 Roue classique	18
Figure I.7 Robot mobile	23
Figure I.8 Robot mobile de type voiture	24

Chapitre II

Figure II.1 Convergence du système glissant	30
Figure II.2 l'erreur de posture de la coordonnée pour un robot mobile	37
Figure II.3 Commande d'entrée v	46
Figure II.4 Commande d'entrée w	47
Figure II.5 Poursuite de trajectoire circulaire	47
Figure II.6 Les erreurs de position et l'erreur angulaire	47
Figure II.7 Stratégie de contrôle PID-backstepping	48
Figure II.8 Schéma fonctionnel de simulation	52
Figure II.9 Les erreurs de position et l'erreur angulaire	53
Figure II.10 Suivi de trajectoire circulaire	53
Figure II.11 Control d'entré v	54
Figure II.12 Control d'entré ω	54
Figure II.13 Control d'entré T_1 (torque1)	54
Figure II.14 Control d'entré T_2 (torque2)	55
Figure II.15 Stratégie de commande PID- FTSM	58
Figure II.16 Les erreurs de position et les erreurs angulaires	59
Figure II.17 Le suivi d'une trajectoire circulaire	59
Figure II.18 La vitesse linéaire v	60
Figure II.19 La vitesse angulaire ω	60
Figure II.20 La loi de commande (torque)	61

Chapitre III

Figure III.1 Neurone Formel	63
Figure III.2 Modèle général d'un neurone	63

Figure III.3 Structure d'un réseau de neurones MLP à 03 couches	66
Figure III.4 Architecture d'un Réseau de Neurone RBF	66

Chapitre IV

Figure IV.1 Système mobile de contrôle des mouvements du robot	72
Figure IV.2 un réseau neuronal à deux couches	73
Figure IV.3 Stratégie de la commande dynamique basée sur l'approche GFSM	75
Figure IV.4 Stratégie de la commande dynamique basée sur NN-GFSM	76
Figure IV.5 Suivi de trajectoire circulaire	79
Figure IV.6 Erreurs x_e, y_e, θ_e	79
Figure IV.7 Vitesses dynamiques v et ω	80
Figure IV.8 Erreurs de vitesses	80
Figure IV.9 Couples générés	80
Figure IV.10 Suivi de trajectoires avec perturbations	81
Figure IV.11 Couples τ_1 et τ_2	82
Figure IV.12 Suivi de trajectoire circulaire	82
Figure IV.13 Erreurs de position x_e et y_e	83
Figure IV.14 Erreur angulaire θ_e	83
Figure IV.15 Vitesses angulaires ω et ω_c	84
Figure IV.16 Vitesses linéaires v et v_c	84
Figure IV.17 Couples générés	85
Figure IV.18 Fonctions f et \hat{f}	85

LISTE DES TABLEUX

Tableau 1 : Paramètres des roues	19
----------------------------------	----

INTRDUCTION GENERALE

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 [1].

Les robots mobiles autonomes dirigeants dans un environnement structuré et non structuré doivent être capables d'être utilisés dans des conditions d'imprécision et d'incertitude présentes dans de tels environnements [2].

Afin que le robot puisse être autonome, le dispositif doit posséder de nombreuses capacités, notamment :

- ✓ Percevoir son environnement et se localiser dans celui-ci.
- ✓ Se déplacer d'un point à l'autre en trouvant des chemins efficaces et sécuritaires afin d'éviter toute collision.
- ✓ Identifier les objets présents dans son espace [3].

Différentes techniques ont été étudiées pour démontrer ces lois d'approches en particulier la commande discontinue proposée par Utkin [4], Young et Utkin [5] qui ne peut prendre que deux valeurs suivant le signe de la surface de glissement $S(x)$. La caractéristique principale de cette technique est sa structure variable avec la commutation autour de la surface choisie, engendrant du coup des fréquences de commutation élevées des organes de commande qui peuvent provoquer une détérioration anticipée de l'organe de commande ou exciter des dynamiques hautes fréquences non considérées dans la modélisation du système. Ceci crée un phénomène de broutement (chattering en anglais) autour de la surface de glissement qui peut endommager les actionneurs de commande dans la mesure où la commande va osciller entre deux valeurs suivant le signe de $S(x)$ d'une part et ne permet généralement pas de glissement sur la surface d'autre part.

Dans le but de résoudre le phénomène de réticence (chattering) et de l'éliminer complètement, des méthodes ont été étudiées par Slotine [6], Sastry et Slotine [7], Asada et Slotine [8], Hung et al [9], Habbi et al [10], Harashima et al [11], et Levant [12] en agissant sur la dynamique d'approche de la surface de glissement.

Dans ce travail, nous nous sommes intéressées par l'identification et la commande d'un robot mobile (la commande Back stepping, commande Feed back linéarisation, la commande mode glissant, et la commande basée sur les réseaux de neurones). Avec des simulations, sera réalisée grâce au logiciel MATLAB.

Le présent travail est composé et organisé en quatre chapitres comme suit :

Le premier chapitre est consacré à quelques généralités concernant la robotique mobile en général ainsi que les modélisations cinématique et dynamique des trois types des robots mobiles " robot mobile de type unicycle, robot mobile de type tricycle, et robot mobile de type voiture", et une étude de leur architecture.

Le second chapitre présente différents commandes appliquées à la poursuite de trajectoire « Back stepping, Feed back linéarisation, Mode glissant », avec son contrôleur et identificateur est simulé sous MATLAB/SIMULINK.

Le troisième chapitre donne les différents modèle et types d'architectures de réseaux de neurones, la base théorique portant sur les principales structures de réseaux de neurones, leurs méthodes d'apprentissages ainsi que leurs domaines d'utilisation notamment en tant que réseau de neurones contrôleur.

Dans le dernier chapitre, nous présentons les détails de la conception d'un contrôleur de réseau de neurone pour suivi de trajectoire de système de robot mobile, avec la simulation de ce contrôleur sous l'environnement MATLAB/SIMULINK.

CHAPITRE I

MODELISATION DES ROBOTS MOBILES

I.1 Introduction

La modélisation d'un robot, considéré comme étant un système mécanique articulé, actionné et commandé, consiste à en établir un modèle mathématique. Outre une fonction générale d'aide à la conception, elle a de multiples utilisations pour, la prédiction des mouvements, l'adaptation des actionneurs, la planification des tâches, l'établissement des lois de commande, l'incorporation du robot dans des simulations informatiques...etc.

Deux types de modèles sont généralement utilisés lors de la commande, à savoir : le modèle cinématique et le modèle dynamique ;

- ✓ Les contraintes cinématiques, qui portent sur la géométrie des déplacements possibles du robot.
- ✓ Les contraintes dynamiques, liées aux effets du mouvement (accélération bornées, vitesse bornées, présence de forces d'inertie ou de friction). [13]

On rencontre plusieurs types de robots à savoir : les robots de type unicycle, les robots de type tricycle, et les robots de type voiture.

I.2 Modélisation d'un robot mobile de type unicycle

Un robot de type unicycle est actionné par deux roues indépendantes, il possède éventuellement des roues folles pour assurer sa stabilité. Son centre de rotation est situé sur l'axe reliant les deux roues motrices. [14]

Caractéristiques : Stables, Complexité mécanique faible, non-holonome.

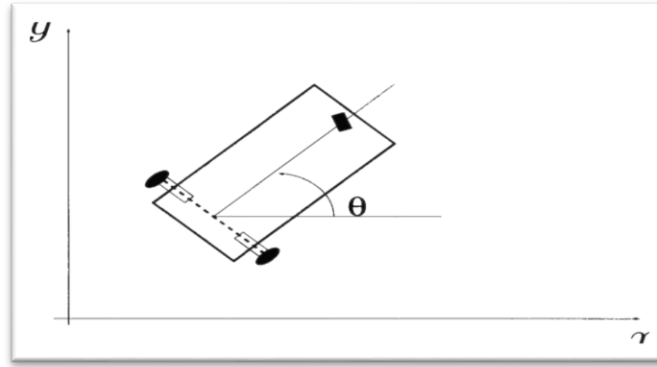


Figure I.1 : Robot mobile de type unicycle [13].

I.2.1 Modélisation Cinématique

Notons $\{X_I, Y_I\}$ un repère fixe quelconque et $\{X_r, Y_r\}$ un repère mobile lié au robot. Soient $q^i = [x^i, y^i, \theta^i]^T$ un point du repère $\{X_I, Y_I\}$ et $q^r = [x^{ri}, y^{ri}, \theta^r]^T$ un point du repère $\{X_r, Y_r\}$.

Les points q^i et q^r sont liés par la matrice orthogonale $R(\theta)$.

$$q^i = R(\theta)q^r \quad (1.1)$$

Avec :

$$R(\theta) = \begin{pmatrix} \cos \theta & -\sin \theta & 0 \\ \sin \theta & \cos \theta & 0 \\ 0 & 0 & 1 \end{pmatrix} \quad (1.2)$$

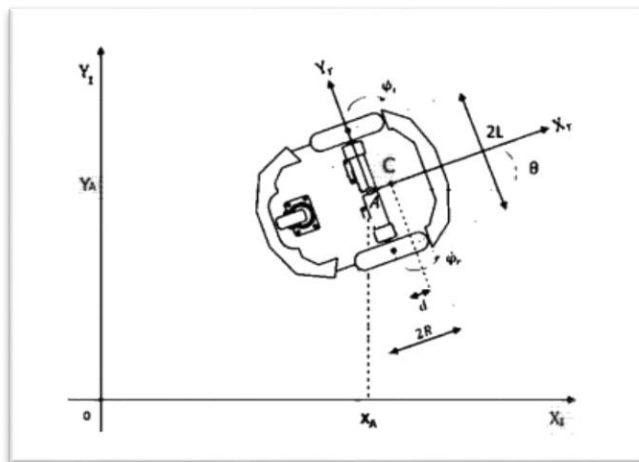


Figure I.2. Robot mobile de type unicycle dans les repères fixe.

Avec :

- A : le point milieu de l'axe des roues ;
- 2R : représente le diamètre de la roue du robot ;
- 2L : représente la largeur du robot ;

Chapitre I : Modélisation des robots mobiles

- $\dot{\varphi}_r$ et $\dot{\varphi}_l$: représentent respectivement la vitesse de rotation de la roue droite et de la roue gauche
- θ : est l'angle d'orientation du robot
- d : est la distance entre le point A et le point C

Le mouvement du robot est caractérisé par deux contraintes non-holonomes qui sont obtenues par deux hypothèses.

Si le robot peut instantanément se déplacer en avant ou en arrière mais il ne peut pas se déplacer à droite et à gauche sans que les roues glissent, on dit qu'il possède une contrainte non holonome à savoir :

- Aucun glissement latéral.
- Roulement sans glissement. [15].

Aucun glissement latéral:

Le robot peut se déplacer uniquement en avant et en arrière, et la vitesse est nulle le long de l'axe

latéral, $\dot{y}_A^r = 0$

Alors l'expression de la vitesse du robot est :

$$\begin{pmatrix} \dot{x}_A^i \\ \dot{y}_A^i \\ \dot{\theta}_A^i \end{pmatrix} = \begin{pmatrix} \cos\theta & -\sin\theta & 0 \\ \sin\theta & \cos\theta & 0 \\ 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} \dot{x}_A^r \\ 0 \\ \dot{\theta}_A^r \end{pmatrix} \Rightarrow \begin{cases} \dot{x}_A^i = \dot{x}_A^r \cdot \cos\theta \\ \dot{y}_A^i = \dot{x}_A^r \cdot \sin\theta \end{cases} \quad (1.3)$$

$$\Rightarrow -\dot{x}_A^i \sin\theta + \dot{y}_A^i \cos\theta = 0 \quad (1.4)$$

Roulement sans glissement:

La contrainte de roulement sans glissement représente le fait que chaque roue maintient un point en contact avec le sol. [16]

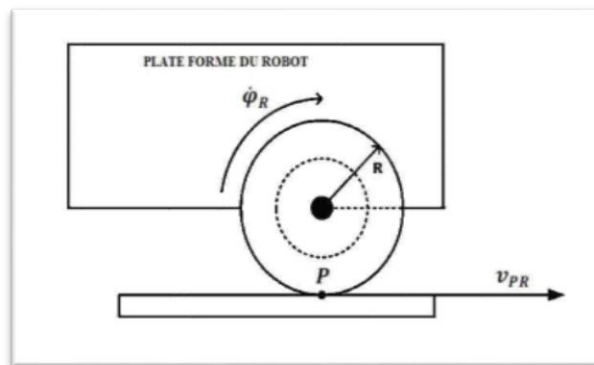


Figure I.3: Caractérisation du roulement sans glissement.

Chaque roue du robot au point de contact P à la vitesse linéaire :

$$\begin{cases} v_{PL} = R \dot{\varphi}_L \\ v_{PR} = R \dot{\varphi}_R \end{cases} \quad (1.5)$$

Avec :

- v_{PL} : la vitesse linéaire de la roue gauche.
- v_{PR} : la vitesse linéaire de la roue droite. [16]

Les expressions des positions et des vitesses généralisées dans le repère fixe en fonction des coordonnées du point A sont données par :

- Roue droite

$$\begin{cases} x_{PR}^I = x_A^I + L \sin \theta \\ y_{PR}^I = y_A^I - L \cos \theta \end{cases} \Rightarrow \begin{cases} \dot{x}_{PR}^I = \dot{x}_A^I + L \dot{\theta} \sin \theta \\ \dot{y}_{PR}^I = \dot{y}_A^I - L \dot{\theta} \cos \theta \end{cases} \quad (1.6)$$

- Roue gauche

$$\begin{cases} x_{PL}^I = x_A^I - L \sin \theta \\ y_{PL}^I = y_A^I + L \cos \theta \end{cases} \Rightarrow \begin{cases} \dot{x}_{PL}^I = \dot{x}_A^I - L \dot{\theta} \sin \theta \\ \dot{y}_{PL}^I = \dot{y}_A^I + L \dot{\theta} \cos \theta \end{cases} \quad (1.7)$$

En utilisant la matrice de rotation R(e) et en l'appliquant à la roue droite on a:

$$\begin{pmatrix} \dot{x}_{PR}^I \\ \dot{y}_{PR}^I \\ \dot{\theta}^I \end{pmatrix} = \begin{pmatrix} \cos \theta & -\sin \theta & 0 \\ \sin \theta & \cos \theta & 0 \\ 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} \dot{x}_{PR}^r \\ \dot{y}_{PR}^r \\ \dot{\theta}^r \end{pmatrix} \quad (1.8)$$

Avec :

$\dot{y}_{PR}^r = 0$ signifie que la vitesse au point P de la roue droite est nulle (aucun glissement latéral).

$$\Rightarrow \begin{pmatrix} \dot{x}_{PR}^I \\ \dot{y}_{PR}^I \\ \dot{\theta}^I \end{pmatrix} = \begin{pmatrix} \dot{x}_{PR}^r \cos \theta \\ \dot{x}_{PR}^r \sin \theta \\ \dot{\theta}^r \end{pmatrix} \quad (1.9)$$

Ou

$$\vec{v}_{PR} = \dot{x}_{PR}^r \vec{i} + \dot{y}_{PR}^r \vec{j} = R \dot{\varphi}_R \vec{i} \quad (1.10)$$

$$\Rightarrow v_{PR} = \dot{x}_{PR} = R \dot{\varphi}_R \quad (1.11)$$

L'équation (1.9) dévie :

$$\begin{cases} \dot{x}_{PR} \cos \theta = \dot{x}_{PR} \cos^2 \theta & (1) \\ \dot{y}_{PR} \sin \theta = \dot{y}_{PR} \sin^2 \theta & (2) \end{cases} \quad (1.12)$$

En sommant (1) et (2), on obtient:

$$\dot{x}_{PR} \cos \theta + \dot{y}_{PR} \sin \theta = R \dot{\varphi}_R \quad (1.13)$$

Et le même pour la roue gauche :

$$\dot{x}_{PL} \cos \theta + \dot{y}_{PL} \sin \theta = R \dot{\varphi}_L \quad (1.14)$$

Donc le système d'équation des deux roues se la forme suivante [14]:

$$\begin{cases} \dot{x}_{PR} \cos \theta + \dot{y}_{PR} \sin \theta = R \dot{\varphi}_R \\ \dot{x}_{PL} \cos \theta + \dot{y}_{PL} \sin \theta = R \dot{\varphi}_L \end{cases} \quad (1.15)$$

En introduisant les équations (1.6) et (1.7) dans (1.15) on obtient :

$$\begin{cases} -\dot{x}_{PR} \sin \theta + \dot{y}_{PR} \cos \theta = 0 \\ \dot{x}_{PR} \cos \theta + \dot{y}_{PR} \sin \theta = R \dot{\varphi}_R \\ \dot{x}_{PL} \cos \theta + \dot{y}_{PL} \sin \theta = R \dot{\varphi}_L \end{cases} \quad (1.16)$$

On peut écrire l'équation (1.16) :

$$A(q) \dot{q} = 0 \quad (1.17)$$

A(q) est la matrice de contraintes non-holonomes donnée par:

$$A(q) = \begin{pmatrix} -\sin \theta & \cos \theta & 0 & 0 & 0 \\ \cos \theta & \sin \theta & L & -R & 0 \\ \cos \theta & \sin \theta & -L & 0 & -R \end{pmatrix} \quad (1.18)$$

\dot{q} Représente la dérivé de la coordonnée généralisée q, donnée par [17] :

$$\dot{q} = \begin{bmatrix} \dot{x}_A \\ \dot{y}_A \\ \dot{\theta} \\ \dot{\varphi}_R \\ \dot{\varphi}_L \end{bmatrix}^T \quad (1.19)$$

On peut obtenir l'expression des vitesses linéaires des roues droite et gauche au point de contact P par L'équation (1.16) :

$$\begin{cases} v_{PR} = v_A + L\dot{\theta} \\ v_{PL} = v_A - L\dot{\theta} \end{cases} \quad (1.20)$$

En posant:

$$\begin{cases} v = v_A \\ \theta = \varpi \end{cases} \text{ et } \begin{cases} v_{PR} = v_R \\ v_{PL} = v_L \end{cases} \quad (1.21)$$

On obtient l'expression de la vitesse linéaire v et la vitesse angulaire ϖ du robot mobile en fonction des vitesses de rotation de la roue gauche $\dot{\varphi}_L$ et de la roue droite $\dot{\varphi}_R$.

$$\begin{cases} v = \frac{(v_R + v_L)}{2} = \frac{R(\dot{\varphi}_R + \dot{\varphi}_L)}{2} \\ \omega = \frac{(v_R - v_L)}{2L} = \frac{R(\dot{\varphi}_R - \dot{\varphi}_L)}{2L} \end{cases} \quad (1.22)$$

Dans le repère mobile les coordonnées du point A sont :

$$\begin{cases} \dot{x}_A^r = v = \frac{R(\dot{\varphi}_R + \dot{\varphi}_L)}{2} \\ \dot{y}_A^r = 0 \\ \dot{\theta}_A^r = \omega = \frac{R(\dot{\varphi}_R - \dot{\varphi}_L)}{2L} \end{cases} \quad (1.23)$$

Devant l'équation (1.2), on peut écrire :

$$\begin{pmatrix} \dot{x}_A^i \\ \dot{y}_A^i \\ \dot{\theta}_A^i \end{pmatrix} = \begin{pmatrix} \cos\theta & -\sin\theta & 0 \\ \sin\theta & \cos\theta & 0 \\ 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} \dot{x}_A^r \\ \dot{y}_A^r \\ \dot{\theta}_A^r \end{pmatrix} \quad (1.24)$$

En remplaçant l'équation (1.23) dans (1.24), on obtient:

$$\begin{pmatrix} \dot{x}_A^I \\ \dot{y}_A^I \\ \dot{\theta}_A^I \end{pmatrix} = \begin{pmatrix} \cos\theta & -\sin\theta & 0 \\ \sin\theta & \cos\theta & 0 \\ 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} R \begin{pmatrix} \dot{\varphi}_R + \dot{\varphi}_L \end{pmatrix} \\ 2 \\ 0 \\ R \begin{pmatrix} \dot{\varphi}_R - \dot{\varphi}_L \end{pmatrix} \\ 2L \end{pmatrix} \quad (1.25)$$

$$\Rightarrow \begin{pmatrix} \dot{x}_A^I \\ \dot{y}_A^I \\ \dot{\theta}_A^I \end{pmatrix} = \begin{pmatrix} \frac{R}{2} \cos\theta & \frac{R}{2} \cos\theta \\ \frac{R}{2} \sin\theta & \frac{R}{2} \sin\theta \\ \frac{R}{2L} & -\frac{R}{2L} \end{pmatrix} \begin{pmatrix} \dot{\varphi}_R \\ \dot{\varphi}_L \end{pmatrix} \quad (1.26)$$

En combinant les équations (1.22) et (1.26), soit encore à partir de l'équation (1.25), on obtient le modèle cinématique du robot mobile uni cycle :

$$\dot{q}_A^I = \begin{pmatrix} \dot{x}_A^I \\ \dot{y}_A^I \\ \dot{\theta}_A^I \end{pmatrix} = \begin{pmatrix} \cos\theta & 0 \\ \sin\theta & 0 \\ 0 & 1 \end{pmatrix} \begin{pmatrix} v \\ \omega \end{pmatrix} \quad (1.27)$$

1.2.2 Modélisation Dynamique

Les équations du modèle dynamique qui décrivent le mouvement du robot sont données soit par la méthode d'Euler - Lagrange, soit par la méthode de Newton - Euler.

Considérons un robot non-holonyme avec n coordonnées généralisées (q_1, q_2, \dots, q_n) et soumis à m contraintes.

Le formalisme d'Euler-Lagrange est décrit par l'équation différentielle suivante:

$$\frac{\partial}{\partial t} \left(\frac{\partial L}{\partial \dot{q}_i} \right) + \frac{\partial L}{\partial q_i} = F + A^T(q) \cdot \lambda_k \quad (1.28)$$

Et :

$$L(q, \dot{q}) = T - V \quad (1.29)$$

Avec :

- L : le Lagrangien.
- T : L'énergie cinétique du système ;
- V : L'énergie potentielle du système ;
- F : Le vecteur de force généralisée ;

Chapitre I : Modélisation des robots mobiles

- λ_k : Le vecteur des multiplicateurs de Lagrange ;

- q_i : la coordonnée généralisée et $q = \begin{bmatrix} x_A, y_A, \theta, \varphi_R, \varphi_L \end{bmatrix}^T$ (1.30)

L'énergie cinétique T du système est donnée par :

$$T = T_C + T_{\omega R} + T_{\omega L} \quad (1.31)$$

Avec :

- T_C : est l'énergie cinétique de la plateforme.
- $T_{\omega R}$: est l'énergie cinétique de la roue droite.
- $T_{\omega L}$: est l'énergie cinétique de la roue gauche.

Ou :

$$\begin{cases} T_C = \frac{1}{2} m_c v_c^2 + \frac{1}{2} I_c \dot{\theta}^2 \\ T_{\omega R} = \frac{1}{2} m_\omega v_{\omega R}^2 + \frac{1}{2} I_m \dot{\theta}^2 + \frac{1}{2} I_\omega \dot{\varphi}_R^2 \\ T_{\omega L} = \frac{1}{2} m_\omega v_{\omega L}^2 + \frac{1}{2} I_m \dot{\theta}^2 + \frac{1}{2} I_\omega \dot{\varphi}_L^2 \end{cases} \quad (1.32)$$

Avec :

- m_c : la masse de la plate-forme.
- m_ω : la masse de chaque roue plus la masse du moteur.
- $v_{\omega R}$: la vitesse linéaire de la roue droite.
- $v_{\omega L}$: la vitesse linéaire de la roue gauche.
- I_c : Moment d'inertie de la plate-forme du robot sans les roues, les moteurs, autour de l'axe vertical qui passe par le point C.
- I_m : Moment d'inertie de chaque roue avec le moteur par rapport au diamètre de la roue.
- I_ω : Moment d'inertie de chaque roue avec le moteur par rapport à l'axe de la roue.

Le point C dans le repère fixe a pour coordonnées :

$$\begin{cases} x_c = x_A + d \cdot \cos \theta \\ y_c = y_A + d \cdot \sin \theta \end{cases} \quad (1.33)$$

A partir de l'équation (1.31) et (1.33) on obtient l'énergie cinétique totale sous la forme suivante :

$$T = \frac{1}{2} m (\dot{x}_A^2 + \dot{y}_A^2) - m_c \cdot d \cdot \dot{\theta} (\dot{y}_A \cos \theta - \dot{x}_A \sin \theta) + \frac{1}{2} I_\omega (\dot{\varphi}_R^2 + \dot{\varphi}_L^2) + \frac{1}{2} I \dot{\theta}^2 \quad (1.34)$$

Ou :

$$\begin{cases} m = m_c + 2m_\omega \\ I = I_c + m_c \cdot d^2 + 2m_\omega L^2 + 2I_m \end{cases} \quad (1.35)$$

L'énergie potentielle étant nulle car le robot se déplace sur un plan horizontal.

Alors on a donc le Lagrangien [14] :

$$L = T \quad (1.36)$$

En se servant de l'équation (1.34), l'équation (1.28) devient :

$$\begin{cases} m \ddot{x}_A - md \ddot{\theta} \sin \theta - md \dot{\theta}^2 \cos \theta = C_1 \\ m \ddot{y}_A - md \ddot{\theta} \cos \theta - md \dot{\theta}^2 \sin \theta = C_2 \\ -md \ddot{x}_A \sin \theta + md \ddot{y}_A \cos \theta + I \ddot{\theta} = C_3 \\ I_\omega \ddot{\varphi}_R = \tau_R + C_4 \\ I_\omega \ddot{\varphi}_L = \tau_L + C_5 \end{cases} \quad (1.37)$$

Avec :

- C1, C2, C3, C4 et C5 : sont les coefficients relatifs aux contraintes cinématiques qui peuvent être exprimés en fonction du vecteur de multiplicateurs de Lagrange et de la matrice de contrainte cinématique $A^T(q)$.

Le modèle dynamique du robot mobile à roue peut se mettre sous la forme :

$$M(q) \ddot{q} + V(q, \dot{q}) \dot{q} = B(q)u - A^T(q) \lambda \quad (1.38)$$

Ou :

$$M(q) = \begin{pmatrix} m & 0 & -md \sin \theta & 0 & 0 \\ 0 & m & md \cos \theta & 0 & 0 \\ -md \sin \theta & md \cos \theta & I & 0 & 0 \\ 0 & 0 & 0 & I\omega & 0 \\ 0 & 0 & 0 & 0 & I\omega \end{pmatrix}$$

$$V(q, \dot{q}) = \begin{pmatrix} 0 & -md \dot{\theta} \cos \theta & 0 & 0 & 0 \\ 0 & -md \dot{\theta} \sin \theta & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \end{pmatrix}$$

$$B(q) = \begin{pmatrix} 0 & 0 \\ 0 & 0 \\ 0 & 0 \\ 1 & 0 \\ 0 & 1 \end{pmatrix}$$

$$A^T(q) \cdot \lambda = \begin{pmatrix} -\sin \theta & \cos \theta & \cos \theta \\ \cos \theta & \sin \theta & \sin \theta \\ 0 & L & -L \\ 0 & -R & 0 \\ 0 & 0 & -R \end{pmatrix} \begin{bmatrix} \lambda_1 \\ \lambda_2 \\ \lambda_3 \\ \lambda_4 \\ \lambda_{51} \end{bmatrix}$$

Avec :

- $M(q)$: la matrice d'inertie symétrique définie positive de taille $n \times n$.
- $V(q, \dot{q})$: la matrice des forces centrifuges et des forces de Coriolis.
- $B(q)$: la matrice de transformation d'entrée.
- $A(q)$: la matrice des contraintes non-holonomes.
- u : le vecteur d'entrée, $u = \begin{bmatrix} \tau_R \\ \tau_L \end{bmatrix}$

En définissant $\eta = \begin{pmatrix} \dot{\varphi}_R \\ \dot{\varphi}_L \end{pmatrix}$ comme étant le vecteur de vitesses auxiliaires et en utilisant l'équation

(1.26) on peut écrire que :

$$\begin{pmatrix} \dot{x}_A \\ \dot{y}_A \\ \dot{\theta} \\ \dot{\varphi}_R \\ \dot{\varphi}_L \end{pmatrix} = \begin{pmatrix} \frac{R}{2} \cos \theta & \frac{R}{2} \cos \theta \\ \frac{R}{2} \sin \theta & \frac{R}{2} \sin \theta \\ \frac{R}{2L} & -\frac{R}{2L} \\ 1 & 0 \\ 0 & 1 \end{pmatrix} \begin{pmatrix} \dot{\varphi}_R \\ \dot{\varphi}_L \end{pmatrix} \quad (1.39)$$

$$\Rightarrow \dot{q} = S(q)\eta \quad (1.40)$$

$$\text{Et } \ddot{q} = \dot{S}(q)\eta + S(q)\dot{\eta} \quad (1.41)$$

$S(q)$ est une matrice de rang complet qui satisfait à la condition suivante :

$$S^T(q)A^T(q) = 0 \quad (1.42)$$

En remplaçant (1.39) et (1.40) dans (1.38), on obtient :

$$M(q) = \left[\dot{S}(q)\eta + S(q)\dot{\eta} \right] + V(q, \dot{q})S(q)\eta = B(q)u - A^T(q)\lambda \quad (1.43)$$

En multipliant l'équation (1.43) par $S^T(q)$, on a :

$$S^T(q)M(q)\dot{S}(q)\eta + S^T(q)M(q)S(q)\dot{\eta} + S^T(q)V(q, \dot{q})S(q)\eta = S^T(q)B(q)u - S^T(q)A^T(q)\lambda \quad (1.44)$$

En posant:

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

L'équation (1.43) sous la forme réduite est :

$$\bar{M}(q)\dot{\eta} + \bar{V}(q, \dot{q})\eta = \bar{B}(q)u \quad (1.45)$$

Ou :

$$\bar{M}(q) = \begin{bmatrix} I_\omega + \frac{R^2}{4L^2}(mL^2 + I) & \frac{R^2}{4L^2}(mL^2 - I) \\ \frac{R^2}{4L^2}(mL^2 - I) & I_\omega + \frac{R^2}{4L^2}(mL^2 + I) \end{bmatrix}$$

$$\bar{V}(q, \dot{q}) = \begin{bmatrix} 0 & \frac{R^2}{2L^2}(m_c d \dot{\theta}) \\ -\frac{R^2}{2L^2}(m_c d \dot{\theta}) & 0 \end{bmatrix}$$

$$\bar{B}(q) = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}$$

En insérant (1.23) dans (1.45) on obtient :

$$\begin{bmatrix} (m + \frac{2I_\omega}{R^2}) & 0 \\ 0 & (I + \frac{2L^2}{R^2} I_\omega) \end{bmatrix} \begin{pmatrix} \dot{v} \\ \dot{\omega} \end{pmatrix} + \begin{bmatrix} 0 & -m_c d \omega \\ m_c d \omega & 0 \end{bmatrix} \begin{pmatrix} v \\ \omega \end{pmatrix} = \begin{bmatrix} \frac{1}{R} & 0 \\ 0 & \frac{L}{R} \end{bmatrix} \begin{pmatrix} u_1 \\ u_2 \end{pmatrix} \quad (1.46)$$

Avec :

$$u = \begin{cases} u_1 = \tau_R + \tau_L \\ u_2 = \tau_R - \tau_L \end{cases}$$

Ou :

- (τ_R, τ_L) : est le couple d'entrée exprimé en Newton mètres (N.m).

L'équation (1.46) est la forme non linéaire du modèle dynamique du robot mobile soumis à notre étude ; et son expression linéarisée est:

$$\begin{bmatrix} (m + \frac{2I_\omega}{R^2}) & 0 \\ 0 & (I + \frac{2L^2}{R^2} I_\omega) \end{bmatrix} \begin{pmatrix} \dot{v} \\ \dot{\omega} \end{pmatrix} = \begin{bmatrix} \frac{1}{R} & 0 \\ 0 & \frac{L}{R} \end{bmatrix} \begin{pmatrix} u_1 \\ u_2 \end{pmatrix} \quad (1.47)$$

Ou :

- m : est la masse totale du robot.
- I : est l'inertie totale équivalente du système.
- R : est le rayon de la roue.
- $2L$: est la largeur du robot.

En remplaçant :

- $(m + \frac{2I_\omega}{R^2})$ par m_0 .
- $(I + \frac{2L^2}{R^2} I_\omega)$ par I_0 .

On obtient une expression simplifiée de l'équation (1.47) donnée par:

$$\begin{bmatrix} m_0 & 0 \\ 0 & I_0 \end{bmatrix} \begin{pmatrix} \dot{v} \\ \dot{\omega} \end{pmatrix} = \begin{bmatrix} \frac{1}{R} & 0 \\ 0 & \frac{L}{R} \end{bmatrix} \begin{pmatrix} u_1 \\ u_2 \end{pmatrix} \quad (1.48)$$

I.3 Modélisation d'un robot mobile de type tricycle

Un robot mobile de type tricycle est constitué de deux roues fixes placées sur un même axe et d'une roue centrée orientable placée sur l'axe longitudinal. Le mouvement du robot est donné par la vitesse des deux roues fixes et par l'orientation de la roue orientable. Son centre de rotation est situé à l'intersection de l'axe contenant les roues fixes et de l'axe de la roue orientable [18]. C'est un robot non-holonome. En effet, il est impossible de le déplacer dans une direction perpendiculaire aux roues fixes. Sa commande est plus compliquée. Il est en général impossible d'effectuer des rotations simples à cause d'un rayon de braquage limité de la roue orientable [19].

Caractéristique : Peu stable, Complexité mécanique faible, non-holonome [20].

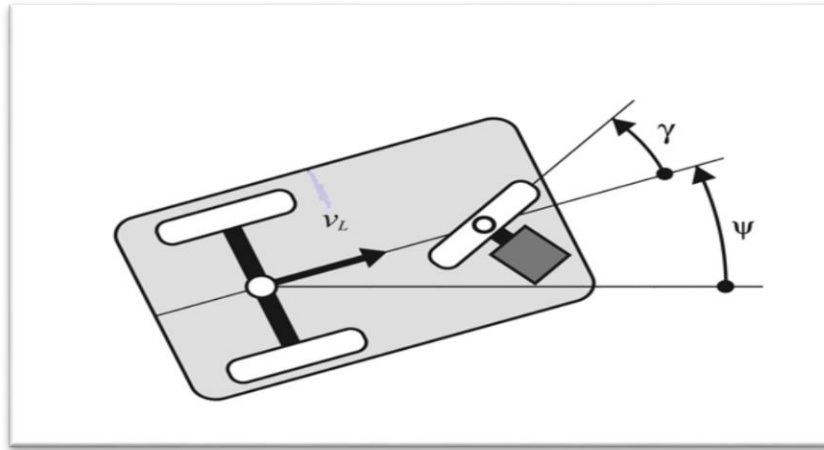


Figure I.4 : Robot mobile de type tricycle

I.3.1 Modélisation Cinématique

Pour l'analyse cinématique nous considérons un robot capable de locomotion sur une surface par l'action de roues montées sur le robot, donc considérons l'hypothèse suivante:

- Le robot mobile sur roues se déplace sur une surface plane horizontale, d'où l'énergie potentielle $u(\xi)$ est constante.
- Les axes de guidage sont orthogonaux surface.
- Il n'existe pas des éléments flexibles sur la structure du robot.
- Le contact entre les roues et le visage est réduit à un seul point (condition antidérapante).

Pour déterminer la position du robot est nécessaire d'obtenir les composants de translation et de rotation du système coordonné attaché du robot, ce le cadre est mobile par rapport au cadre inertiel [21].

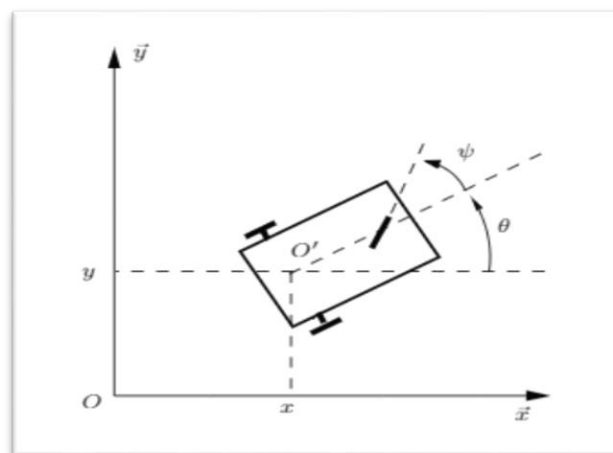


Figure I.5 : Robot mobile de type tricycle [22].

La position du robot mobile de type tricycle est décrit à la Figure (1.5) tous les cadres de

La posture du robot est donnée par :

$$\xi = \begin{bmatrix} x \\ y \\ \theta \end{bmatrix} \quad (1.49)$$

$$R(\theta) = \begin{bmatrix} \cos \theta & \sin \theta & 0 \\ \sin \theta & \cos \theta & 0 \\ 0 & 0 & 1 \end{bmatrix} \quad (1.50)$$

Où :

- $R(\theta) \in SO(3)$ est la matrice de rotation [23].

Une différence essentielle entre les robots mobiles à roues et d'autres systèmes robotiques sont les restrictions cinématiques. Ces restrictions sont non holonomes, donc l'analyse cinématique et dynamique est plus compliquée que les systèmes holonomes [23]- [24].

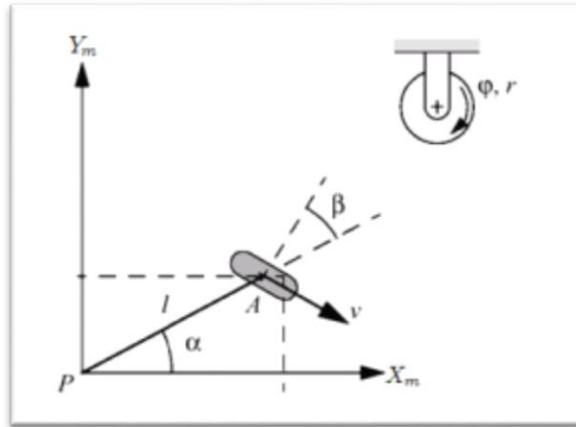


Figure 1.6 : Roue classique. [25]

Nous utilisons la roue conventionnelle comme indiqué dans la Figure (1.6) A indique le centre de la roue qui est un point fixe du cadre Σ_m . La position de A est caractérisée par des coordonnées polaires, c.-à-d. la distance l de P à A et l'angle α .

L'orientation du plan de la roue par rapport à l'axe est représentée par l'angle β . L'angle de rotation de la roue autour de son essieu horizontal est noté par $\varphi(t)$ et le rayon de la roue par r . Avec cette description, les composantes de la vitesse du point de contact sur le plan de la roue et orthogonale au plan de la roue respectivement, sont facilement calculées par :

$$\begin{bmatrix} -\sin(\alpha + \beta) & \cos(\alpha + \beta) & l \cos \beta \end{bmatrix} R(\theta) \dot{\xi} + r \dot{\varphi} = 0 \quad (1.51)$$

$$\begin{bmatrix} \cos(\alpha + \beta) & \sin(\alpha + \beta) & l \sin \beta \end{bmatrix} R(\theta) \dot{\xi} = 0$$

Satisfaire les deux conditions de roulement pur et non-glissement le long du mouvement. Les

Chapitre I : Modélisation des robots mobiles

roues fixes et roues orientables centrées ont les mêmes contraintes, sauf que dans le centre orientable roues l'angle β (t) varie dans le temps.

Le robot mobile à roues en configuration classique tricycle est présenté dans la figure (1.7). Ce robot mobile possède deux roues fixes conventionnelles sur le même essieu et une roue orientable centrée conventionnelle. Les contraintes de tout robot mobile à roues peuvent être écrites sous la forme de matrice générale [26].

$$\begin{aligned} J_1(\beta_c, \beta_{oc})R(\theta)\xi + J_2\varphi &= 0 \\ C_1(\beta_c, \beta_{oc})R(\theta)\xi + C_2\varphi &= 0 \end{aligned} \quad (1.52)$$

Où :

- β_{oc} : est un angle pour une roue orientable conventionnelle décentrée

Dans le cas d'un tricycle et compte tenu des paramètres présentes dans le tableau 1, nous avons que :

$$\begin{aligned} J_1 &= \begin{pmatrix} J_{10} \\ J_{11} \\ J_{12} \end{pmatrix} \\ &= \begin{bmatrix} \cos\beta & \sin\beta & l\cos\beta \\ 0 & 1 & \frac{L}{2} \\ 0 & -1 & \frac{L}{2} \end{bmatrix} \end{aligned} \quad (1.53)$$

$$J_2 = \text{diag}(r)$$

$$\begin{aligned} C_1 &= \begin{pmatrix} C_{10} \\ C_{11} \\ C_{12} \end{pmatrix} \\ &= \begin{bmatrix} \sin\beta & -\cos\beta & l\sin\beta \\ 1 & 0 & 0 \\ -1 & 0 & 0 \end{bmatrix} \end{aligned} \quad (1.54)$$

$$C_2 = 0$$

Tableau 1 : Paramètres des roues [25]

Roue	α	β	l
1c	$\frac{3\lambda}{2}$	-	L
2f	0	0	$\frac{L}{2}$
3f	λ	0	$\frac{L}{2}$

I.3.2 Modélisation Dynamique

La dérivation du modèle dynamique d'un robot joue un rôle important pour la simulation du mouvement, l'analyse des structures du robot et la conception d'algorithmes de contrôle.

La dynamique d'un robot mobile non holonomique en accord avec la méthodologie Euler-Lagrange [21], [27] est donnée pour :

$$\frac{d}{dt} \left(\frac{\partial K}{\partial \dot{q}} \right) - \left(\frac{\partial K}{\partial q} \right) = A(q)\lambda + B(q)\tau \quad (1.55)$$

Où :

- K : est l'énergie cinétique.
- B(q) τ : est l'ensemble des forces généralisées avec B(q) une matrice cinématique à (n*p) et τ la p-vecteur des couples appliqués.
- A(q) : est la matrice associée aux non-contraintesholonomiques.
- λ : est le m-vecteur des multiplicateurs de grange.

L'énergie cinétique K [28] est définie comme suit :

$$K = \frac{1}{2} \dot{q}^T M(q) \dot{q} \quad (1.56)$$

Où :

- M(q) : est la n*n positive symétrique définie. matrice d'inertie.

Le mouvement du robot est donné par les 7 coordonnées suivantes :

$$q = [x \quad y \quad \theta \quad \beta \quad \phi_1 \quad \phi_2 \quad \phi_3]^T \quad (1.56)$$

Les équations dynamiques du tricycle ont la forme suivante :

$$\frac{d}{dt} \left(\frac{\partial K}{\partial \dot{\xi}} \right) - \frac{\partial K}{\partial \xi} = R^T(\theta) [J_1^T(\beta)\lambda + C_{10}^T(\beta)\mu + C_{11}^T v] \quad (1.57)$$

$$\frac{d}{dt} \left(\frac{\partial K}{\partial \dot{\xi}} \right) - \frac{\partial K}{\partial \xi} = B_1 \tau_1 \quad (1.58)$$

$$\frac{d}{dt} \left(\frac{\partial K}{\partial \dot{\xi}} \right) - \frac{\partial K}{\partial \xi} = J_2^T \lambda + B_2 \tau_2 \quad (1.59)$$

Où :

- $\lambda = (\lambda_1, \lambda_2, \lambda_3)$.
- μ et ν : sont les 5 Lagrange multiplicateurs associés aux 5 contraintes cinématiques indépendantes, $B_1=1$ et $B_2=[0 \ 0 \ 1]^T$.

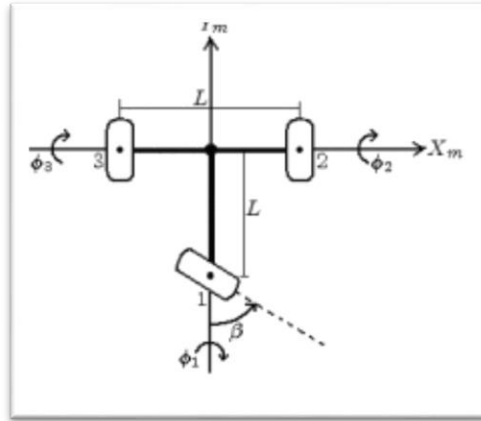


Figure I.7 : robot mobile.

L'énergie cinétique du robot mobile à roues Figure (I.7) est exprimé par la symétrie suivante :

Forme quadratique :

$$K = \frac{1}{2} \dot{q}^T \begin{bmatrix} R^T(\theta)M(\beta)R(\theta) & R^T(\theta)v(\beta) & 0 \\ v^T(\beta)R(\theta) & I_\beta & 0 \\ 0 & 0 & I_\phi \end{bmatrix} \dot{q} \quad (1.60)$$

Ou $M(\beta)$ est une matrice symétrique de (3*3) définie par :

$$M_{11}(\beta) = M_{22}(\beta) = M + \sum_{i=1}^3 m_i$$

$$M_{12}(\beta) = M_{21}(\beta) = 0$$

$$M_{13}(\beta) = M_{31}(\beta) = -Ml_{c2} - \sum_{i=1}^3 m_i l_i \sin \alpha_i$$

$$M_{23}(\beta) = M_{32}(\beta) = -Ml_{c2} - \sum_{i=1}^3 m_i l_i \cos \alpha_i$$

$$M_{33}(\beta) = 0 + M(l_{c1} + l_{c2}) - \sum_{i=1}^3 m_i l_i^2 \quad (1.61)$$

V (β) est un vecteur 3 défini comme suit :

$$V(\beta) = \begin{bmatrix} 0 \\ 0 \\ I_\beta \end{bmatrix}$$

Et I_ϕ ont la forme suivante :

$$I_\phi = \begin{bmatrix} I_{\phi1} & 0 & 0 \\ 0 & I_{\phi2} & 0 \\ 0 & 0 & I_{\phi3} \end{bmatrix}$$

Où :

- M : masse du chariot.
- m_i : masse de la roue i ; $i=1, \dots, 3$.
- l_{c1}, l_{c2} : coordonnées du centre de masse du chariot dans le cadre fixé au chariot Σ_m .
- I_0 : moment d'inertie du chariot autour des axes verticaux passant par son centre de masse.
- I_β : moment d'inertie de la roue orientable centrée.
- $I_{\phi i}$: moment d'inertie de la roue i , autour de son axe de rotation.

En utilisant cette énergie cinétique, le modèle dynamique (1.57), (1.58), (1.59) est réécrit comme suit :

$$R^T(\theta)M(\beta)R(\theta)\ddot{\xi} + f_1 = R^T(\theta)\left[J_1^T(\beta)\lambda + C_{10}^T(\beta)\mu + C_{11}^T v\right]$$

$$v^T(\beta)R(\theta)\ddot{\xi} + f_2 = \tau_1$$

$$I_\phi \ddot{\phi} = J_2^T \lambda + B_2 \tau_2 \quad (1.62)$$

Où :

- f_1 et f_2 : sont respectivement un vecteur 3 et vecteur 1 des fonctions de $(\theta, \dot{\theta}, \beta, \dot{\beta}, \ddot{\beta})$.

Ces fonctions sont donnés par :

$$f_1 = R^T(\theta) \left[\left(2M(\beta)S_t - R(\theta)D(\dot{\xi}, \xi, \dot{\beta}) \right) R(\theta) \dot{\xi} + v(\beta) \ddot{\beta} + S_t^T v(\beta) \dot{\beta} \right]$$

$$f_2 = v(\beta)^T S_t R(\theta) \dot{\xi} + I_\beta \quad (1.63)$$

Ou le dérivé de temps $\dot{R}(\theta) = S_t R(\theta)$, la matrice symétrique S_t est [29]:

$$S_t = \begin{bmatrix} 0 & \dot{\theta} & 0 \\ -\dot{\theta} & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix}$$

Et $D(\dot{\xi}, \xi, \beta)$:

$$D(\dot{\xi}, \xi, \beta) = \begin{bmatrix} [0 & 0 & 0] \\ [0 & 0 & 0] \\ \left[M(\beta) S_p R(\theta) \xi + S_p^T v(\beta) \beta \right]^T \end{bmatrix}$$

Et S_p :

$$S_p = \begin{bmatrix} 0 & 1 & 0 \\ -1 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix}$$

Et il lié à $\frac{\partial R(\theta)}{\partial \theta} = S_p R(\theta)$ Sous forme compacte, le modèle dynamique (1.62) du robot mobile à roues est réécrit comme suit :

$$M(q) \ddot{q} + C(q, \dot{q}) \dot{q} = A^T(q) \lambda + B \tau \quad (1.64)$$

Ou :

- $M(q)$: est la matrice symétrique définie positive ($n \times n$).
- $C(q)$: est la matrice ($n \times n$) centripète et Coriolis torques [25].

I.4 Modélisation d'un robot mobile de type voiture

Un robot de type voiture est semblable au tricycle, il est constitué de deux roues fixes placées sur même axe et de deux roues centrées orientables placées elle aussi sur un même axe; le robot de type voiture cependant plus stable puisqu'il possède un point d'appui supplémentaire. Toutes les autres propriétés de robot voiture sont identique au robot tricycle, le deuxième pouvant être

ramené au premier en remplaçant les deux roues avant par une seule place au centre de l'axe, et ceci de manière à laisser le centre de rotation inchangé. [30]

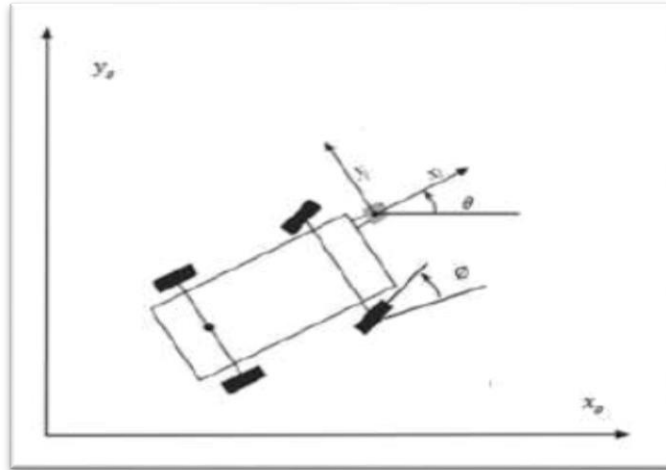


Figure I.8 : Robot mobile de type voiture [31].

I.4.1 Modélisation Cinématique

Le robot mobile de type voiture il est composé de deux roues motrices indépendantes en arrière du corps principal du robot et deux roues en avant pour le guidage (non commandées), Les vitesses linéaires V_R et V_L des roues motrice (droite et gauche) sont donc :

$$V_{\omega R} = \dot{\varphi}_R \cdot r \quad (1.65)$$

$$V_{\omega L} = \dot{\varphi}_L \cdot r \quad (1.66)$$

La vitesse \vec{V}_G du centre de gravité et les vitesses linéaires, des roues droites et gauches sont nécessaires ultérieurement pour établir le modèle dynamique.

Vitesse du centre de gravité du châssis du robot

Le vecteur de vitesse du centre de gravité du châssis du robot est donné par [30]:

$$\vec{V}_G = (\dot{x}_G \cos\theta + \dot{y}_G \sin\theta)\vec{i}_1 + (-\dot{x}_G \sin\theta + \dot{y}_G \cos\theta)\vec{j}_1 \quad (1.67)$$

Vitesse de la roue droite du robot

La loi de composition de vitesse au centre de la roue on obtient :

$$\vec{V}_{\omega R} = (\dot{x}_G \cos\theta + \dot{y}_G \sin\theta + b\dot{\theta})\vec{i}_1 + (-\dot{x}_G \sin\theta + \dot{y}_G \cos\theta - L_r \dot{\theta})\vec{j}_1 \quad (1.68)$$

La vitesse du point de contact de la roue droite est donnée par :

$$\vec{V}_{\omega R}^{cl} = (\dot{x}_G \cos\theta + \dot{y}_G \sin\theta + b\dot{\theta} - r\dot{\varphi}_R)\vec{i}_1 + (-\dot{x}_G \sin\theta + \dot{y}_G \cos\theta - L_r \dot{\theta})\vec{j}_1 \quad (1.69)$$

Vitesse de la roue gauche du robot

La vitesse au centre de la roue on obtient :

$$\vec{V}_{\omega_L} = (\dot{x}_G \cos\theta + \dot{y}_G \sin\theta - b\dot{\theta})\vec{i}_1 + (-\dot{x}_G \cos\theta + \dot{y}_G \sin\theta - L_r \dot{\theta})\vec{j}_1 \quad (1.70)$$

La vitesse du point de contact de la roue gauche est donnée par :

$$\vec{V}_{\omega_L}^{c2} = (\dot{x}_G \cos\theta + \dot{y}_G \sin\theta + b\dot{\theta} - r\dot{\varphi})\vec{i}_1 + (-\dot{x}_G \cos\theta + \dot{y}_G \sin\theta - L_r \dot{\theta})\vec{j}_1 \quad (1.71)$$

I.4.2 Modélisation Dynamique

On appelle alors Lagrangien de ce système, le scalaire noté $L(q, \dot{q})$ définie par :

$$L(q, \dot{q}) = T(q, \dot{q}) - V(q) \quad (1.72)$$

Ou :

- $T(q, \dot{q})$: est l'énergie cinétique.
- $V(q)$: est l'énergie potentielle.

L'équation d'Euler-Lagrange est donnée par :

$$\frac{d}{dt} \left(\frac{\partial L}{\partial \dot{q}_n} \right) - \left(\frac{\partial L}{\partial q_n} \right) = F - A^T \cdot \lambda_n \quad (1.73)$$

Avec :

- $F = E(q) \cdot \tau$
- q_n : Coordonnées généralisées.
- $q_n = [x_G, y_G, \varphi_R, \varphi_L]^T$.
- λ_n : Les multiplicateurs de Lagrange.

On peut calculer $T(q, \dot{q})$ par :

$$T(q, \dot{q}) = \frac{1}{2} \dot{q}^T M(q) \dot{q} \quad (1.74)$$

Nous obtenons que les équations d'Euler-Lagrange s'écrivent sous la forme :

$$M(q) \ddot{q} + V(q, \dot{q}) = B(q)\tau - A(q)^T \quad (1.75)$$

Avec :

- $B(q)$: matrice de projection.
- $M(q)$: matrices d'inertie.
- λ : multiplicateur de Lagrange.
- $A(q)^T$: Matrice des contraintes cinématiques non holonomes.

- $V(q, \dot{q})$: Matrice de Coriolis.

Un système mécanique non holonôme avec n coordonnées généralisées (q_1, q_2, \dots, q_n) et soumis à m contraintes peut être décrit généralement par les équations de mouvement suivantes :

$$M(q)\ddot{q} + V(q, \dot{q}) + F(\dot{q}) + G(q) + \tau_d = B(q)\tau - A^T \lambda \quad (1.76)$$

Avec :

- $M(q)$: matrice d'inertie ($n \times n$) symétrique définie positive.
- $V(q, \dot{q})$: matrice de Coriolis.
- $F(\dot{q})$: matrice de frottement impliquant la viscosité et le frottement dynamique.
- $G(q)$: le vecteur gravitationnel.
- τ_d : le vecteur de perturbations incluant la dynamique non-modélisée.
- $B(q)$: la matrice de transformation.
- τ : le vecteur couple.
- A^T : la matrice associée aux contraintes cinématiques.
- λ : le vecteur des multiplicateurs de Lagrange. [32] [33]

Pour trouver le modèle dynamique en utilisant l'approche de Lagrange est de calculer les énergies cinétiques et potentielles qui régissent le mouvement du robot mobile.

On suppose que les robots mobiles sont rigides et qu'ils se déplacent sur un sol horizontal, Par conséquent, leur énergie potentielle est nulle $V(q) = 0$, et les énergies cinétiques totales sont la somme de l'énergie cinétique de la plate-forme du robot sans roues augmentées des énergies cinétiques des roues et des actionneurs.

$$L = T = T_{rot} + T_{tr} \quad (1.77)$$

Avec :

- T_{tr} : est l'énergie cinétique de translation du corps définie par :

$$T_{tr} = \frac{1}{2} m_c \cdot V_G^2 + \frac{1}{2} m_{\omega 1} \cdot V_{\omega 1}^2 + \frac{1}{2} m_{\omega 2} \cdot V_{\omega 2}^2 \quad (1.78)$$

Ou :

- m_c : la masse du châssis.
- V_G : la vitesse linéaire du centre de gravité.
- $m_{\omega i}$: la masse de la roue i .
- $V_{\omega i}$: la vitesse linéaire par rapport au centre de la roue.

L'énergie cinétique de rotation T_{rot} , est définie par :

$$T_{rot} = \frac{1}{2} \cdot \vec{\Omega}^t \cdot I_G \vec{\Omega} + \sum_{i=1}^2 \frac{1}{2} I_{\omega_i} \cdot \dot{\varphi}_{\omega_i}^2 \quad (1.79)$$

Avec :

- $\vec{\Omega}^t$: le vecteur ligne représente les composantes du vecteur de rotation suivant l'axe de rotation OZ, est représentée par $\vec{\Omega}^t = (0, 0, \dot{\theta})$
- I_G : Moment d'inertie du châssis calculée par rapport au centre de gravité G.
- I_{ω_i} : Moment d'inertie des roues par rapport au centre des roues.
- $\dot{\varphi}_{\omega_i}$: vitesse de rotation de la roue i.

Les coordonnées généralisées donnant la position instantanée est représenté par :

$$q = [X_G \quad Y_G \quad \varphi_R \quad \varphi_L]^T \quad (1.80)$$

Tout calcul fait le Lagrangien du système est égale [27] :

$$L = T = T_{rot} + T_{tr} = \frac{1}{2} m (\dot{x}_G^2 + \dot{y}_G^2) - 2m_\omega L_r \dot{\theta} (-x_G \sin \theta + y_G \cos \theta) + \frac{1}{2} I \dot{\theta}^2 + \frac{1}{2} I_\omega (\dot{\varphi}_{\omega_R}^2 + \dot{\varphi}_{\omega_L}^2) \quad (1.81)$$

Avec :

$$\begin{cases} m = m_c + 2m_\omega \\ I = I_G + 2I_m + sm_\omega (b^2 + L_r^2) \end{cases} \quad (1.82)$$

1.5 Conclusion

Dans ce chapitre, nous avons étudié la modélisation du trois types des robots mobiles (le robot mobile de type unicycle, tricycle, et voiture), nous avons présenté la modélisation cinématique, et la modélisation dynamique de chaque type de robot mobile.

Le modèle dynamique et le modèle cinématique sont deux modèles essentiels sur lesquels est basée la commande.

CHAPITRE II

ELABORATION DES EFFIRENTECOMMANDES APPLIQUEES ALA POURSUIT DE TRAJECTOIRE

II.1 Introduction

La poursuite de trajectoire permet au robot mobile d'être autonome dans le suivi d'une trajectoire prédéfinie à des instants précis sans l'intervention de l'être humain. Et pour respecter un itinéraire et des horaires précis la synthèse d'une commande en poursuite nécessite la connaissance de la trajectoire désirée. Cette dernière doit être exprimée en fonction des états d'un robot virtuel de même type que le robot à commander. Les états virtuels permettent alors le calcul du modèle d'erreurs de poursuite, ce dernier est indispensable pour l'élaboration de la loi de commande [34].

Plusieurs stratégies de commande ont été utilisées afin d'accomplir la tâche de poursuite d'un robot mobile à roues, dans ce chapitre on va présenter les deux commandes suivantes :

- Backsteeping.
- Mode glissant.

II.2 Conception du contrôleur pour le suivi de trajectoire

Dans la littérature, le problème de suivi non holonomique est simplifié en négligeant la dynamique du robot et en ne considérant le système de direction, donc plusieurs approches sont abordées.

Le problème de suivi de trajectoire d'un robot mobile non holonomique peut être décrit et formulé comme suit :

Qu'il y ait une trajectoire de référence prescrite :

$$\dot{x}_r = v_r \cos \theta_r \quad (2.1)$$

$$\dot{y}_r = v_r \sin \theta_r \quad (2.2)$$

$$\dot{\theta}_r = \omega_r \quad (2.3)$$

$$q_r = [x_r \quad y_r \quad \theta_r]^T \quad (2.4)$$

$$v_r = [v_r \quad \omega_r]^T \quad (2.5)$$

Avec $v_r > 0$ tous un contrôle de vitesse lisse $v_c = f(e_p, v_r, K)$ tel que $\lim_{t \rightarrow \infty} (q_r - q) = 0$, où e_p , v_r et K sont respectivement l'erreur de position de suivi, le vecteur de vitesse de référence et le vecteur de gain de contrôle. Calculer ensuite l'entrée de couple pour l'équation dynamique telle que $v \rightarrow v_c$ comme $t \rightarrow \infty$. Ce chapitre détaille les étapes et les différentes parties de la conception de cet algorithme de contrôle. Deux types de commandes sont présentées, une commande cinématique basée sur le mode glissant afin d'assurer une convergence des vitesses des robots vers les vitesses de référence, une commande dynamique basée sur deux contrôleurs ; un contrôleur cinématique (commande par mode glissant et la commande backstepping) et un contrôleur dynamique qui est un contrôleur PID.

II.3 Commande par mode glissant (CMG)

La commande par modes de glissement est une commande non linéaire, elle est caractérisée par la discontinuité de la commande aux passages d'une surface de commutation appelée surface de glissement. Dans cette étude, nous donnons une définition du concept de système à structure variable et nous exprimons les caractéristiques essentielles de cette loi de commande à partir de différents outils théoriques de l'automatique linéaire et non linéaire.

Le principe de la commande par mode glissant est, à l'aide d'une commande discontinue, de contraindre le système à atteindre une surface donnée pour ensuite y rester. La synthèse d'une CMG se déroule en deux temps :

- une surface est déterminée en fonction des objectifs de commande et des propriétés statiques et dynamiques désirées pour le système bouclé,
- une commande discontinue est synthétisée de manière à contraindre les trajectoires d'état du système à atteindre et, ensuite, à rester sur cette surface en dépit d'incertitudes, de variations de paramètres [35]. . .

II.3.1 Principe de la commande par modes de glissement

L'idée de base de la commande par régime glissant est premièrement d'attirer les états du système dans une région convenablement sélectionnée, puis de concevoir une loi de commande qui maintiendra toujours le système dans cette région. On peut résumer la commande par mode glissant en les deux étapes essentielles suivantes (voir Figure II.1):

- Synthétiser une surface $S(x, t)$, telle que toutes les trajectoires du système aient le comportement désiré (poursuite, régulation et stabilité).
- Définir une loi de commande (commutation) $U(x, t)$ qui est capable d'attirer toutes les trajectoires d'état vers la surface de glissement et [36].

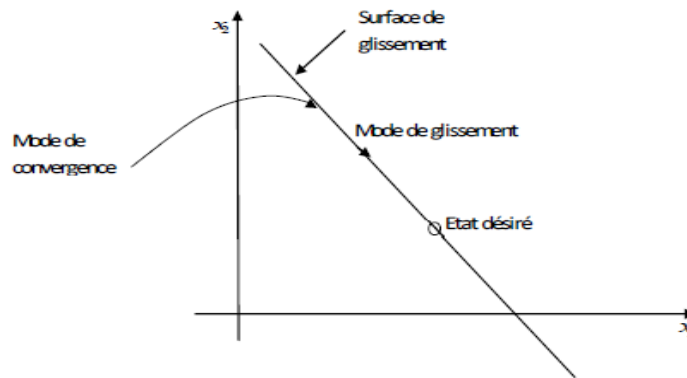


Figure II.1 : Convergence du système glissant.

II.3.2 Conception de la commande par mode glissement

Le développement des régulateurs de type mode de glissement prend en charge systématiquement le problème de la stabilité et le problème des performances désirées. La mise en œuvre de cette commande nécessite les trois étapes suivantes :

- Le choix des surfaces de glissement.
- L'établissement des conditions d'existence et de convergence.
- La détermination de la loi de commande [36].

II .3.2.1 Choix de la surface de glissement

La surface de glissement peut être linéaire ou non linéaire, elle est construite de telle sorte que le système ait toujours une dynamique désirée et est généralement choisie avec un degré relatif égal à un (i.e. la dérivée de la surface $S(x)$ par rapport au temps fait apparaître explicitement la commande U) [37].

Soit un système décrit par l'équation différentielle suivante:

$$\dot{x}(t) = f(x, t) + g(x, t)u \tag{2.6}$$

Où :

- $x = (x_1, x_2, x_3, \dots, x_n)^T$: appartient à X est le vecteur d'état de système.
- $u = (u_1, u_2, u_3, \dots, u_n)^T$: est le vecteur de commande.
- f, g : sont des champs du vecteur suffisamment différentiables définis sur x .

La surface est donnée par :

$$S_j(x) = \sum_{i=1}^n c_{ji} \cdot e_i \tag{2.7}$$

Où :

- $S_i(x) = 0$: est la fonction de commutation, elle sépare l'espace d'état en deux parties disjointes

$$S(x, t) < 0 \text{ et } S(x, t) > 0$$

- $S(x) = (S_1(x), S_2(x), S_3(x), \dots, S_n(x))$: Vecteur de la surface de glissements.

- $x_{refi} = [x_{ref1}, x_{ref2}, \dots, x_{refn}]^T$: Vecteur de référence (l'état désiré).

- $e_i = x_i - x_{refi}$: L'erreur entre la variable à commander et sa référence.

- $e_i = [e_1, e_2, \dots, e_n]^T$: Vecteur de l'écart.

- c_{ij} : Paramètres de la surface de glissement.

- c : Matrice d'éléments de $(m * n)$ paramètres de la surface (matrice de pente de la droite de surface).

Le choix des paramètres de la surface se fait de façon satisfaisant les performances désirées.

Slotine propose une forme d'équation générale pour déterminer la surface de glissement qui assure la convergence d'une variable vers sa valeur désirée.

$$S(x) = \left(\frac{\partial}{\partial t} + \lambda \right)^{r-1} . e(x) \quad (2.8)$$

Avec :

- λ : Une constante positive.
- r : degré relatif, égal au nombre de fois qu'il faut dériver la sortie pour faire apparaître la commande.
- $s(x) = 0$ est une équation différentielle linéaire dont l'unique solution est :
 $e(x) = 0$, ceci implique que l'état du système est égal à l'état désiré.

II .3.2.2Principe d'invariance

Cette propriété nécessite toutefois que certaines hypothèses appelées (*matching conditions*) soient vérifiées par les perturbations [38].

$$\dot{x}(t) = (A + \Delta A).x(t) + B.u(t) + f(t) \quad (2.9)$$

Où :

- ΔA et $f(t)$: sont respectivement le terme d'erreurs de modélisation et une perturbation externe.

La signification physique de cette hypothèse est que l'on considère des incertitudes de modélisation ou une perturbation attaquant le système par la matrice d'entrée.

II .3.2.3Conditions de convergence et d'existence

Les conditions d'existence et de convergence sont les critères qui permettent aux différentes dynamiques du système de converger vers la surface de glissement et d'y rester,

Chapitre II : ELABORATION DES EFFIRENTECOMMANDES APPLIQUEES ALA POURSUIT DE TRAJECTOIRE indépendamment de la perturbation. Il existe deux considérations pour assurer le mode de convergence.

a. Fonction directes de la commutation

C'est la première condition de convergence. Cette condition est proposée par EMILYANOV et UTKIN et elle donnée par :

$$\begin{cases} \dot{s}(x) > 0 & \text{lorsque } s(x) < 0 \\ \dot{s}(x) < 0 & \text{lorsque } s(x) > 0 \end{cases} \quad (2.10)$$

Ce qui est équivalent à :

$$S'(x) \cdot S(x) < 0 \quad (2.11)$$

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

b. La Fonction de Lyapunov

Il s'agit de formuler une fonction scalaire positive $V(x) > 0$ pour les variable d'état du système La loi de commande doit faire décroître cette fonction

$V(x) > 0$. L'idée est de choisir une fonction scalaire $S(x)$ pour garantir l'attraction de la variable à contrôler vers sa valeur de référence et de concevoir une commande u telle que le carré de la surface correspond à une fonction de Lyapunov.

Pour la commande mode de glissant, on considère généralement la fonction de Lyapounov suivante :

$$V(x) = \frac{1}{2} S^T(x) S(x) \quad (2.12)$$

La dérivée de cette fonction est:

$$\dot{V}(x) = S(x) \dot{S}(x) \quad (2.13)$$

Pour que la fonction puisse décroître $V(x) > 0$, il suffit d'assurer que sa dérivée soit négative. Ceci n'est vérifiée que si la condition (2.12) est vérifiée.

II.3.3 Synthèse de la commande

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

II.3.3.1 Méthode de Philippov

L'une des premières approches est la méthode de Philippov. Cette approche est basée sur les résultats des travaux effectués par ce dernier sur les équations différentielles à second membre discontinu. Il propose une loi de commande qui prend deux valeurs suivant le signe de la surface de glissement [39]:

$$u = \begin{cases} u^+ \dots si \dots S(x,t) > 0 \\ u^- \dots si \dots S(x,t) < 0 \end{cases} \quad (2.14)$$

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

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

II.3.3.2 Méthode de la commande équivalente d'Utkin

Cette méthode décrit la dynamique du système sur la surface de glissement. Elle permet de garder la variable contrôlée sur cette surface en imposant la valeur désirée en régime permanent de la grandeur d'entrée. Pour un système contrôlé donné, la première étape pour la réalisation d'une structure de régulation par ajout de la commande équivalente consiste à trouver un vecteur d'entrée équivalent de sorte que la trajectoire d'état du système à contrôler reste sur la surface de glissement prédéfinie [40].

Lorsque le système est en régime glissant, la trajectoire restera sur la surface de commutation. Cela peut être exprimé par [41] :

$$S(x,t) = 0 \text{ et } \dot{S}(x,t) = 0 \quad (2.15)$$

La commande équivalente est une fonction continue qui sert à maintenir la variable à contrôler sur la surface de glissement.

La condition (2.15) est appelée la condition de l'invariance de la surface de glissement. Nous prenons : $S(x,t) = S(x(t)) = s(x)$

$$\dot{S} = 0 \Rightarrow \frac{\partial S}{\partial x} (f(x) + g(x)u_{eq}) = 0 \quad (2.16)$$

u_{eq} : est appelée commande équivalente est associée au système nominale.

Elle est déterminée d'une façon unique par la condition de l'invariance (2.15). En supposant que

$$\frac{\partial S}{\partial x} g(x) \neq 0$$

La commande équivalente est donnée par le scalaire :

$$u_{eq} = \left(\frac{\partial S}{\partial x} g(x) \right)^{-1} \frac{\partial S}{\partial x} g(x) \quad (2.17)$$

La commande équivalente représente la valeur moyenne de u qui maintient l'état du système sur la surface de glissement $S(x) = 0$. L'équation du régime glissant idéal est obtenue en portant l'expression u_{eq} dans (2.6) :

$$\dot{x} = f(x) - g(x) \left(\frac{\partial S}{\partial x} g(x) \right)^{-1} \frac{\partial S}{\partial x} f(x) \quad (2.18)$$

$\frac{\partial S}{\partial x}$: étant non nul sur X , ceci implique m états en fonction de $(m-n)$ autres. Ainsi, en régime glissant, la dynamique du système évoluant sur l'espace d'état est réduite de dimensions $(m-n)$. On aura alors m valeurs propres nulles et $(m-n)$ valeurs propres non nulles.

II.3.4 Calcul de la commande

La commande $u(t)$ à déterminer est fonction de la vitesse linéaire $v(t)$ et de la vitesse angulaire $w(t)$ du robot. Rappelons que l'équation différentielle de notre système est l'équation (2.6)

En considérant que notre système est linéaire et sans perturbation, l'équation (2.6) devient [14] :

$$\dot{x}(t) = g(x,t)u(t) = \begin{bmatrix} \frac{1}{m_0 R} & 0 \\ 0 & \frac{1}{m_0 L} \end{bmatrix} u(t) \quad (2.19)$$

En définissant la surface de glissement $S(t)$ comme suit :

$$S(t) = \begin{bmatrix} S_1(t) \\ S_2(t) \end{bmatrix} = e_c(t) + \beta \int e_c(t) dt \quad (2.20)$$

$\beta > 0$

Ou :

- $e_c(t) = [e_v \quad e_w]^T = x_c(t) - x(t)$: est la différence entre la vitesse produite par le contrôleur cinématique (vitesse désirée) et la vitesse réelle du robot.
- $x = \begin{pmatrix} v \\ \omega \end{pmatrix}$: est la vitesse réelle du robot.
- $x_c = \begin{pmatrix} v_c \\ \omega_c \end{pmatrix}$: est la sortie du contrôleur cinématique.

indépendamment des perturbations, il faut que dans $\dot{S}(t) = 0$ la commande s'y trouve.

Alors le système est dans la surface de glissement implique :

$$S(t) = 0 \text{ et } \lim e_c(t) \rightarrow 0$$

En dérivant S (t) on obtient :

$$\dot{S}(t) = \dot{e}_c(t) + \beta e_c(t) \quad (2.21)$$

Pour trouver la commande équivalente $u_{eq}(t)$ il faut résoudre l'équation suivante:

$$\dot{S}(t) = \dot{e}_c(t) + \beta e_c(t) = 0 \quad (2.22)$$

En remplaçant \dot{e}_c par son expression, l'équation (2.22) devient:

$$\begin{cases} \dot{S}_1(t) = \dot{v}_c(t) - \dot{v}(t) + \beta e_v(t) = 0 \\ \dot{S}_2(t) = \dot{\omega}_c(t) - \dot{\omega}(t) + \beta e_\omega(t) = 0 \end{cases} \quad (2.23)$$

En introduisant l'équation de modèle dynamique linéaire simplifié,

$$\begin{cases} \dot{v}(t) = \frac{1}{m_0 R} u_1(t) \\ \dot{\omega}(t) = \frac{1}{I_0 R} u_2(t) \end{cases}$$

Dans (2.66) on obtient:

$$\begin{cases} \dot{v}_c(t) - \frac{1}{m_0 R} u_1(t) + \beta e_v(t) = 0 \\ \dot{\omega}_c(t) - \frac{1}{I_0 R} u_2(t) + \beta e_\omega(t) = 0 \end{cases} \quad (2.24)$$

La commande équivalente obtenue est :

$$u_{eq}(t) = \begin{cases} u_{eq_1}(t) = m_0 R \left[\dot{v}_c(t) + \beta e_v(t) \right] \\ u_{eq_2}(t) = \frac{I_0 R}{L} \left[\dot{\omega}_c(t) + \beta e_\omega(t) \right] \end{cases} \quad (2.25)$$

En remplaçant $m_0 R$ par γ et $\frac{I_0 R}{L}$ par α , l'équation (2.25) devient:

$$u_{eq}(t) = \begin{cases} u_{eq1}(t) = \gamma \left[\dot{v}_c(t) + \beta e_{\dot{v}}(t) \right] \\ u_{eq2}(t) = \alpha \left[\dot{\omega}_c(t) + \beta e_{\dot{\omega}}(t) \right] \end{cases} \quad (2.26)$$

$u_{eq}(t)$: est la commande permettant au système de rester sur la surface de glissement.

En tenant compte des perturbations extérieures, l'équation dynamique (2.19) du système à commander devient :

$$\begin{cases} \dot{v}(t) = \frac{1}{\gamma} u_1(t) + d_1 \\ \dot{\omega}(t) = \frac{1}{\alpha} u_2(t) + d_2 \end{cases} \quad (2.27)$$

Notons que l'équation (2.27) représente la forme linéaire du système soumis à des perturbations, Où :

$d(t) = \begin{pmatrix} d_1 \\ d_2 \end{pmatrix}$: est la perturbation qui s'ajoute à la commande. D'étant une perturbation non connue mais bornée.

Ainsi la nouvelle loi de commande tenant compte de ces perturbations est :

$$u(t) = u_{eq}(t) + u_d(t) \quad (2.28)$$

Où :

$$u_d(t) = \begin{cases} u_{d1}(t) = k_1 \operatorname{sgn}(S_1) \\ u_{d2}(t) = k_2 \operatorname{sgn}(S_2) \end{cases} \quad (2.29)$$

La commande $u(t)$ obtenue par mode glissant classique est donnée par :

$$u(t) = \begin{cases} u_1(t) = \gamma \left[\dot{v}_c(t) + \beta e_{\dot{v}}(t) \right] + k_1 \operatorname{sgn}(S_1) \\ u_2(t) = \alpha \left[\dot{\omega}_c(t) + \beta e_{\dot{\omega}}(t) \right] + k_2 \operatorname{sgn}(S_2) \end{cases} \quad (2.30)$$

Notons que la loi de commande donnée dans l'équation (2.30), est une commande non adaptative car on a supposé que les paramètres du système, soient la masse du robot et l'inertie, sont connus.

II.4 Commande cinématique basée sur la commande par mode de glissement en temps fini:

Notre robot mobile considéré est constitué de trois roues. Deux roues motrices à l'avant du robot mobile sont parallèles, entraînés par deux moteurs indépendants. Une autre est une roue motrice à l'arrière

du robot mobile (Figure II.2). Soit D et r représentent la longueur de l'axe de la roue et le rayon des roues motrices, respectivement. Les vitesses V_l et V_r représentent les vitesses de la roue gauche et la roue droite, respectivement. Les variables de contrôle v et w désignent l'avant la vitesse et la vitesse angulaire du robot mobile, respectivement, ce qui peut être décrit comme

$$\begin{bmatrix} v \\ w \end{bmatrix} = \begin{bmatrix} 1/2 & 1/2 \\ -1/D & 1/D \end{bmatrix} \begin{bmatrix} v_l \\ v_r \end{bmatrix} \quad (2.31)$$

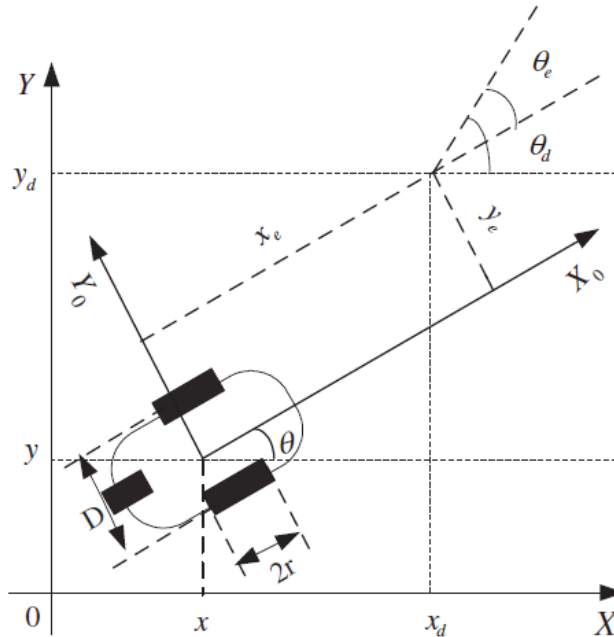


Figure II.2 : l'erreur de posture de la coordonnée pour un robot mobile.

Le modèle cinématique du robot mobile dans le plan $X - Y$ est donnée comme suit :

$$\dot{q} = \begin{bmatrix} \dot{x} \\ \dot{y} \\ \dot{\theta} \end{bmatrix} = \begin{bmatrix} \cos \theta & 0 \\ \sin \theta & 0 \\ 0 & 1 \end{bmatrix} u \quad (2.32)$$

Où $q = [x, y, \theta]^T$ désigne la position de coordonnées dans le centre de la masse du robot mobile. (x, y) Est défini comme étant les coordonnées cartésiennes, et θ est l'angle d'orientation entre la direction de cap et de l'axe x . Et u l'entrée de commande est :

$$u = [v \ w]^T \quad (2.33)$$

Supposons qu'il n'y ait pas d'effet de glissement, et le modèle cinématique du robot mobile satisfait la contrainte non holonome. Pour les robots mobiles sous les hypothèses ci-dessus, l'équation de contrainte non holonome peut être exprimée comme :

$$\dot{x} \sin \theta = \dot{y} \cos \theta \quad (2.34)$$

II.4.1 Problème de suivre la commande

L'objectif de suivi de commande est de concevoir les lois de la vitesse d'avancement v et la vitesse angulaire w , de telle sorte que la trajectoire de référence est suivie par le robot mobile. Dans la (Figure II.2) le vecteur d'état $q_d = (x_d y_d \theta_d)^T$ désigne la posture de référence et le vecteur d'état $q = (x, y, \theta)^T$ représente la position réelle. Le vecteur d'état $q_e = (x_e y_e \theta_e)^T$ est défini comme étant l'erreur de posture. Accord à la relation géométrique, un moyen efficace de transformation de coordonnées global est décrit comme suit [14]

$$q_e = \begin{bmatrix} x_e \\ y_e \\ \theta_e \end{bmatrix} = \begin{bmatrix} \cos \theta & \sin \theta & 0 \\ -\sin \theta & \cos \theta & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x_d - x \\ y_d - y \\ \theta_d - \theta \end{bmatrix} \quad (2.35)$$

En prenant la dérivée de l'équation (3.35), la dynamique d'erreur du système peuvent être obtenus sous forme de :

$$\dot{x}_e = y_e w - v + v_d \cos \theta_e \quad (2.36)$$

$$\dot{y}_e = -x_e w + v_d \sin \theta_e \quad (2.37)$$

$$\dot{\theta}_e = w_d - w \quad (2.38)$$

Considérons $-\frac{\pi}{2} < \theta < \frac{\pi}{2}$, $|v_d| \leq v_d^{max}$ and $w_d \neq 0$, où v_d^{max} sont des constantes appropriées, et $|\cdot|$ désigne le signe de valeur absolue. Dans cet article, la trajectoire de suivi problème pour les robots mobiles est de concevoir un variant dans le temps l'état-rétroaction des lois de la forme :

$$w = w(t, x_e, y_e, \theta_e), \quad (2.39)$$

$$v = v(t, x_e, y_e, \theta_e) \quad (2.40)$$

De telle sorte que trois erreurs d'état x_e , y_e et θ_e converge vers zéro en un temps fini.

II.4.2 la stratégie de suivi la commande

Dans cette section, un système de contrôle combinant trois stratégies différentes est proposé. Tout d'abord, le système compliqué est décomposé en deux sous-systèmes de conception de commande en cascade. Pour les deux sous-systèmes une méthode de commande au temps fini et un temps fini de commande de mode de glissement sont développés respectivement.

II.4.3 la conception de commande en cascade

Considérons le système en cascade $\dot{z} = f(t, z)$ qui peut être exprimée en tant que :

$$\dot{z}_1 = f_1(t, z_1) + g(t, z_1, z_2) \quad (2.41)$$

$$\dot{z}_2 = f_2(t, z_2) \quad (2.42)$$

Où : $z_1 = (x_e, y_e) \in R^2, z_2 = \theta_e \in R$ la fonction $f_1(t, z_1)$ est continûment différentiable par rapport à $(t, z_1), f_2(t, z_2)$ et $g(t, z_1, z_2)$ sont continués dans leurs arguments.

En supposant $z_2 = 0$, il résulte de l'équation (2.41) cette :

$$\dot{z}_1 = f_1(t, z_1)$$

Donc l'équation (2.41) peut être vu un sous-système :

$$\Sigma_1: \dot{z}_1 = f_1(t, z_1) \tag{2.43}$$

Qui est perturbé par le sous-système suivant :

$$\Sigma_2: \dot{z}_2 = f_2(t, z_2) \tag{2.44}$$

Définition 1 : [43] Une fonction continue $\alpha: [0, \infty) \rightarrow [0, \infty)$ est dit être une fonction de classe K seulement si elle est strictement croissante et $\alpha(0) = 0$.

Définition 2 : [44,45] Considérons le système en cascade (2.41) et (2.42) Si les hypothèses suivantes A1-A3 tenir, le système (2.41) et (2.42) est globalement uniformément stable (GUS) :

- A1. Supposons que le sous-système $\Sigma_1: \dot{z}_1 = f_1(t, z_1)$ est GUS et qu'il existe une fonction de Lyapunov définie positive auxiliaire candidat $V(t, z_1): R \geq 0 \times R^n \rightarrow R \geq 0$ tel que :

$$\left\| \frac{\partial V}{\partial z_1} \right\| \|z_1\| \leq cV(t, z_1) \quad \forall \|z_1\| \geq \eta \tag{2.45}$$

Où $c > 0$ et $\eta > 0$ des constants, en outre $\frac{\partial V}{\partial z_1}(t, z_1)$ est délimitée de manière uniforme dans l'ensemble pour $t \|z_1\| \leq \eta$. En d'autres termes, il existe un constant $c_1 > 0$ et $t \geq t_0 \geq 0$ tel que :

$$\left\| \frac{\partial V}{\partial z_1} \right\| \leq c_1 \quad \forall \|z_1\| \leq \eta \tag{2.46}$$

- A2. Le sous-système $\Sigma_2: \dot{z}_2 = f_2(t, z_2)$ est globalement uniformément asymptotiquement stable.
- A3. Pour le système (2.41), il existe une fonction bornée $w(t, z_1): R \geq 0 \times R^2 \rightarrow R \geq 0$ définie positive, adéquate et radialement sans bornes, qui satisfait :

$$\dot{W}(t, z_1)|_{(3.10)} \leq \gamma_1(W(t, z_1)), \quad \forall t \geq t_0 \geq 0 \tag{2.47}$$

Où $\dot{W}(t, z_1)|_{(\#)}$ désigne la dérivée temporelle de $W(t, z_1)$ ainsi que les solutions de l'équation différentielle(#). $\gamma_1: R \geq 0 \rightarrow R \geq 0$ Est une fonction croissante satisfaisant à la condition suivante pour une constante $a > 0$

$$\gamma_1(a) \geq 0; \int_a^\infty \frac{d\eta}{\gamma_1(\eta)} = \infty \tag{2.48}$$

Pour prouver que le système en cascade (2.41) et (2.42) c'est des GUS, nous devons montrer que toutes les solutions de (2.41) et (2.42) sont uniformément bornée et uniformément attrayant pour l'origine

Remarque 1 : Une idée importante de la conception de contrôle en cascade simplifie la conception des lois [45,46], ce qui est souvent importante dans certains cas [44]. Dans notre travail, l'idée de cascade sont utilisés pour simplifier les dynamiques cinématiques de robots mobiles afin de dériver une classe de nouvelles lois sur le contrôle.

II.4.4 Conception de loi de commande angulaire basée sur la technique de commande en temps fini

Selon les idées du système en cascade, nous pouvons voir (2.38), comme un sous-système de premier ordre Σ_2 :

$$\dot{\theta}_e = w_d - w \quad (2.49)$$

Théorème 1 : Considérons un système linéaire de premier ordre :

$$\dot{x} = u \quad (2.50)$$

Qui peuvent être stabilisées par la loi de commande suivante dans un temps fini :

$$u = -x - \alpha x - \beta x^{q_0/p_0} \quad (2.51)$$

Théorème 1 : Considérons un système linéaire de premier ordre :

$$\dot{x} = u \quad (2.52)$$

Qui peuvent être stabilisées par la loi de commande suivante dans un temps fini :

$$u = -x - \alpha x - \beta x^{q_0/p_0} \quad (2.53)$$

Où $x \in \mathbb{R}$ désigne des variables d'état, $\alpha > 0, \beta > 0, p_0 > 0, et q_0 > 0$ sont des nombres entiers impairs $q_0/p_0 < 1$.

Substituer (2.53) dans (2.52) on obtient :

$$\dot{x} = -x - \alpha x - \beta x^{q_0/p_0} \quad (2.54)$$

De (2.54), nous avons :

$$x^{-q_0/p_0} \frac{dx}{dt} + (1 + \alpha)x^{1-\frac{q_0}{p_0}} = -\beta \quad (2.55)$$

On met $y = x^{\frac{1-q_0}{p_0}}$. Nous avons $\frac{dy}{dt} = \frac{p_0 - q_0}{p_0} x^{\frac{-q_0}{p_0}} \frac{dx}{dt}$

Tel que :

$$\frac{dy}{dt} + \frac{p_0 - q_0}{p_0} (1 + \alpha)y = -\frac{p_0 - q_0}{p_0} \beta \quad (2.56)$$

On notera que la solution $\frac{dy}{dx} + P(x)y = Q(x)$ est donné par :

$$y = e^{-\int P(x)dx} \left(\int Q(x) e^{\int P(x)dx} dx + C_0 \right) \quad (2.57)$$

On a équation (2.81), la solution d'équation (3.20) peut être écrite comme :

$$y = e^{\int_0^t \frac{p_0 - q_0}{p_0} (1+\alpha) dt} \left(\int_0^t -\frac{p_0 - q_0}{p_0} \beta e^{\frac{p_0 - q_0}{p_0} (1+\alpha) dt} dt C_0 \right) \quad (2.58)$$

On met : $C_0 = y(0)$, on trouve :

$$\begin{aligned} y &= e^{\frac{-p_0 - q_0}{p_0} (1+\alpha)t} \left(\int_0^t -\frac{p_0 - q_0}{p_0} \beta e^{\frac{-p_0 - q_0}{p_0} (1+\alpha)t} dt + y(0) \right) \\ &= -\frac{\beta}{1+\alpha} + \frac{\beta}{1+\alpha} e^{\frac{p_0 - q_0}{p_0} (1+\alpha)t} + y(0) e^{\frac{-p_0 - q_0}{p_0} (1+\alpha)t} \end{aligned} \quad (2.59)$$

Suppose que :

$t = t_s$, Puis $x = 0$. Notez que $y = 0$ quand $x = 0$. De (2.59) on trouve :

$$e^{\frac{p_0 - q_0}{p_0} (1+\alpha)t_s} = \frac{\beta + (1+\alpha)y(0)}{\beta} \quad (2.60)$$

De l'état initial $x(0) \neq 0 \rightarrow 0$, le temps de convergence peut être exprimée par :

$$t_s = \frac{p_0}{(1+\alpha)(p_0 - q_0)} \ln \frac{(1+\alpha)x(0)^{\frac{p_0 - q_0}{p_0}} + \beta}{\beta} \quad (2.61)$$

La preuve est terminée.

Considérons le premier ordre sous-système (2.57) nous concevons une loi de commande finie à temps, en utilisant Théorème 1. La loi de commande peut être réalisée sous la forme

$$w = w_d + \theta_e + \alpha_1 \theta_e - \beta_1 \theta_e^{q_1/p_1} \quad (2.62)$$

Lorsque $\theta_e \in R$ désigne des variables d'état, $1 > 0, \beta_1 > 0, p_1 > 0$ et $q_1 > 0$ sont des nombres entiers impairs, $q_1/p_1 < 1$ De (2.51), la loi sur la commande de la vitesse angulaire est réalisée sous la forme suivante :

$$\dot{\theta}_e = -\theta_e - \alpha_1 \theta_e - \beta_1 \theta_e^{q_1/p_1} \quad (2.63)$$

Donc, le système (2.49) atteindra $\theta_e = 0$ au moment t_1 , qui est donnée par :

$$t_1 = \frac{p_1}{(1+\alpha)(p_1 - q_1)} \ln \frac{(1+\alpha_1)x(0)^{\frac{p_1 - q_1}{p_1}} + \beta_1}{\beta_1} \quad (2.64)$$

Remarque 2 : Si θ_e est loin de zéro, la loi de commande (2.63) peut être approchée par son terme linéaire $\theta_e = -\theta_e - \alpha_1 \theta_e$ dont la convergence rapide quand il est loin de zéro est bien comprise. Si θ_e est proche de zéro, la loi de commande (2.63) peut être approchée par le terme non-linéaire.

Théorème 2 : Considérons le système (2.61), il existe la loi de commande (2.62) de telle sorte que le système (2.61) est finie à temps stable.

Selon que $\theta_e - \alpha_1 \theta_e - \beta_1 \theta_e^{\frac{q_1}{p_1}} = 0$ nous obtenons que la solution $\theta_e = 0$. D'autre part, pour analyser la stabilité du système de commande donné par (2.84) et (2.86) nous choisissons la fonction Lyapunov décrite par :

$$V = \frac{1}{2} \theta_e^2 \quad (2.65)$$

En utilisant l'équation (2.63), nous pouvons avoir

$$\begin{aligned} \dot{v} = \dot{\theta}_e \theta_e &= \left(-\theta_e - \alpha_1 \theta_e - \beta_1 \theta_e^{\frac{q_1}{p_1}} \right) \theta_e = -\theta_e^2 - \alpha_1 \theta_e^2 - \beta_1 \theta_e^{1 + \frac{q_1}{p_1}} \\ &= \theta_e^2 - \alpha_1 \theta_e - \beta_1 \theta_e^{\frac{p_1 + \alpha_1}{p_1}} \end{aligned} \quad (2.66)$$

En notant que $p_1 > 0$ et $q_1 > 0$ sont des nombres entiers impairs, $\alpha_1 > 0$ et $\beta_1 > 0$, on obtient $V \leq 0$ ($V = 0$ Si et seulement si $\theta_e = 0$)

Selon la définition 2, le système en boucle fermée est finie à temps stable.

De l'analyse de la stabilité du système de commande (2.49) et (2.62), la fonction de Lyapunov V est définie positive ($V(t, 0) = 0$ et $V(t, \theta_e) > 0$ pour tous $\theta_e \neq 0$), son dérivé \dot{V} est définie négative ($V(t, 0) = 0$ et $V(t, \theta_e) < 0$ pour tous $\theta_e \neq 0$). Par conséquent, le système en boucle fermée (2.49) et (2.62) est-temps fini mondial stable

Remarque 3 : La loi de commande de la vitesse angulaire est conçue pour les robots mobiles basés sur la méthode de contrôle temps fini. Le système de premier ordre par correspondant est stabilisé par cette loi de commande à temps fini de la vitesse angulaire de telle sorte que l'erreur d'angle se termine à zéro en un temps fini. Avec cela, la dynamique d'erreur (2.49) - (2.48) sont convertis en un système de second ordre sur la base de l'approche en cascade. Différents du temps fini des lois de contrôle dans [41, 44, 45], nous introduisons deux termes linéaires dans la conception des lois de commande finie à temps fini afin d'obtenir le taux de convergence plus rapide.

II.4.5 Conception de la loi de commande de la vitesse à base de commande en mode glissant et en temps fini

On Note que la loi de commande (2.62) garantit que l'erreur angulaire converge vers zéro. Ainsi, le mode d'erreur de suivi de piste non linéaire (2.48) - (2.49) peut être réduite au système linéaire suivant variant dans le temps où $\theta_e = 0$.

$$\begin{bmatrix} \dot{x}_e \\ \dot{y}_e \end{bmatrix} = \begin{bmatrix} 0 & w_d t \\ -w_d t & 0 \end{bmatrix} \begin{bmatrix} x_e \\ y_e \end{bmatrix} + \begin{bmatrix} 1 \\ 0 \end{bmatrix} [v_d(t) - v] \quad (2.67)$$

$$x_1 = \begin{bmatrix} x_e \\ y_e \end{bmatrix}, A_1 = \begin{bmatrix} 0 & w_d t \\ -w_d t & 0 \end{bmatrix}, B_1 = \begin{bmatrix} 1 \\ 0 \end{bmatrix}, \text{ et } u_1 = v_d(t) - v, \text{ puis}$$

$$\dot{x}_1(t) = A_1(t)x_1(t) + B_1 u_1(t) \quad (2.68)$$

Nous sommes maintenant en position de prouver la contrôlabilité du système (2.68). Selon le critère de contrôlabilité, on peut obtenir :

$$M_0(t) = B_1 = \begin{bmatrix} 1 \\ 0 \end{bmatrix}$$

$$M_1(t) = -A_0 M_0(t) + \frac{d}{dt} M_0(t) = \begin{bmatrix} 0 \\ w_0(t) \end{bmatrix}$$

On a aussi $Rank[M_0(t) : M_1(t)] = Rank \begin{bmatrix} 1 & 0 \\ 0 & w_d(t) \end{bmatrix} = 2$: Dans lequel $w_d(t) \neq 0$. Par conséquent, le système (2.68) peut être commandé.

Basé sur la théorie de la conception de commande en cascade, si $(t) \neq 0$, on considère (2.65), le sous-système Σ_1 , et (2.65) peut être transformés en :

$$\dot{x}_e = w_d y_e - v + v_d \quad (2.69)$$

$$\dot{y}_e = -w_d x_e \quad (2.70)$$

Nous considérons en outre le sous-système donné par (2.70) et la conception d'un retour d'état dexe d'une manière à stabiliser solidement le d'étaty_e = 0.

Théorème 3 : Considérons le système (2.70), où $sgn(.)$ est la fonction signe. On note que $w_d \neq 0$, k_0 est une constante et $k_0 > 0$. Il existe $x_e = k_0 y_e sgn(w_d)$ de telle sorte que le système (2.70) uniformément stable.

Pour résoudre (2.70) pour le point d'équilibre, on a $w_d x_e = 0$, et $x_e = 0$. Par conséquent, le sous-système (2.70) ne comporte qu'un seul point d'équilibre. Afin d'analyser la stabilité du système (2.70), nous considérons la fonction candidate de Lyapunov $V(y_e) = \frac{1}{2} y_e^2$. Le dérivé de Lyapunov $\dot{V}(y_e)$ est donné par :

$$\dot{V}(y_e) = y_e \dot{y}_e = y_e (-w_d x_e) = -k_0 y_e^2 w_d sgn(w_d) \quad (2.71)$$

On note que $sgn(w_d) > 0$ quand $w_d > 0$ et $sgn(w_d) < 0$ quand $w_d < 0$, alors $w_d sgn(w_d) > 0$. $\dot{V}(y_e) \leq 0$ ($V(y_e) = 0$ si et seulement si $y_e = 0$), ce qui implique que la solution zéro $x_e = 0$ à (2.68) est globalement asymptotiquement stable.

Remarque 4 : Dans le théorème 3, nous pouvons utiliser une classe de fonctions continues telles que $\frac{w_d}{|w_d| + \delta}$, $1 - e^{-k_1 w_d}$ et $e^{-k_2 w_d}$ de remplacer le terme (w_d) , où, 1, 2 sont des constants et $\delta > 0$, $k_1 > 0$, $k_2 > 0$, le théorème 3 détient toujours. Le collecteur de glissement est défini par :

$$s = x_e - k_e y_e sgn(w_d) \quad (2.72)$$

En utilisant $x_e = k_e y_e sgn(w_d)$, On obtient :

$$s = x_e - k_e y_e sgn(w_d) = 0 \quad (2.73)$$

Pour garantir l'existence le collecteur de glissement s , à savoir $\dot{s} < 0$, s est spécifiée par :

$$\dot{s} = -s - \alpha_2 s - \beta_2 s^{p_2} \quad (2.74)$$

Où $\alpha_2 > 0$, $\beta_2 > 0$, p_2 est un nombre entier impair, $q_2 p_2 < 1$.

D'après le théorème 1, le système (2.74) atteindra $s = 0$ à t_2 qui est écrit par :

$$t_2 = \frac{p_2}{(1 + \alpha_2)(p_2 - q_2)} \ln \frac{(1 + \alpha_2)s(0)^{(p_2 - q_2)/p_2 + \beta_2}}{\beta_2} \quad (2.75)$$

La commande en mode glissant est conçue en utilisant (2.68) – (2.71) – (2.74). Nous pouvons obtenir :

$$\dot{s} = -\alpha_2 s - \beta_2 s^{\frac{q_2}{p_2}} = \dot{x}_e - k_0 \dot{y}_e \operatorname{sgn}(w_d) = w_d y_e - v + v_d - k_0 (-w_d y_e) \operatorname{sgn}(w_d) \quad (2.76)$$

Ensuite, il résulte de (2.76) que :

$$v = v_d + w_d y_e + k_0 w_d x_e \operatorname{sgn}(w_d) + s + \alpha_2 s + \beta_2 s^{\frac{q_2}{p_2}} \quad (2.77)$$

Considérer le sous-système (2.69) et (2.70), nous obtenons la loi de commande de la vitesse vers l'avant qui garantit uniforme mondiale la stabilité de cette boucle fermée sous-système en exploitant le mode de glissement en temps fini.

Remarque 5 : En choisissant le collecteur coulissant représenté sur l'équation (2.72), l'état x_e va converger vers $k_0 y_e \operatorname{sgn}(w_d)$ en temps fini par la conception de la loi pour atteindre (2.74). Évidemment, si l'état $x_e = 0$, alors l'état $y_e = 0$. De théorème 3, nous savons que y_e va converger vers zéro éventuellement avec le changement de temps de convergence. Il en découle que x_e converge à zéro aussi longtemps que y_e converge à zéro

Remarque 6 : Dans [47, 48, 49, 50], la vitesse angulaire désirée w_d pour les robots mobiles doit satisfaire $0 < w_d^{\min} \leq |w_d| \leq w_d^{\max}$, lorsque w_d^{\min} et w_d^{\max} sont des constantes appropriées. En revanche, la vitesse angulaire souhaitée w_d a seulement besoin de satisfaire $w_d \neq 0$. Dans notre travail. Par conséquent la conception des lois de contrôle dans ce travail se détend des contraintes strictes sur les vitesses souhaitées rapportées dans les littératures existantes.

Remarque 7 : De (2.74) il est remarquable que la conception de la loi de commande introduise un terme non linéaire pour le système de second ordre (2.69) et (2.70). Dans [51, 52, 53, 54, 55] les états arrivent au niveau du collecteur coulissant dans l'infini du temps de prise.

II.4.6 L'analyse de la stabilité

Dans cette section, nous analysons la stabilité du système en boucle fermée (2.75), (2.73) et (2.77) en utilisant le lemme 1.

Théorème 4: Considérons le système (2.48) - (2.49), il existe des lois de commande (2.73) et (2.77) de telle sorte que le système (2.74) - (2.75) est uniforme globalement stable. On note que les collecteurs de glissement converge vers zéro en un temps fini, nous pouvons obtenir $x_e = k_e y_e \operatorname{sgn}(w_d)$, et (2.77) sera remplacé par :

$$v = v_d + w_d y_e + k_0 w_d x_e \operatorname{sgn}(w_d) \quad (2.78)$$

La substitution des équations (2.73) et (2.53) dans (2.49) - (2.50), nous avons :

$$\begin{bmatrix} \dot{x}_e \\ \dot{y}_e \end{bmatrix} = \begin{bmatrix} -k_0 w_d x_e \operatorname{sgn}(w_d) \\ -w_d x_e \end{bmatrix} + \begin{bmatrix} y_e \sigma(\theta_e) + v_d (\cos \theta_e - 1) \\ v_d \sin \theta_e - x_e \sigma(\theta_e) \end{bmatrix} \quad (2.79)$$

$$\dot{\theta}_e = -\sigma(\theta_e) \quad (2.80)$$

Chapitre II : ELABORATION DES EFFICIENTES COMMANDES APPLIQUÉES À LA POURSUITE DE TRAJECTOIRE
Lorsque, De toute évidence, la fonction $\sigma(\cdot) \in K$. En utilisant la définition 2, nous obtenons :

$$f_1(t, z_1) = \begin{bmatrix} -k_0 w_d x_e \operatorname{sgn}(w_d) \\ -w_d x_e \end{bmatrix} \quad (2.81)$$

$$g(t, z_1, z_2) = \begin{bmatrix} y_e \sigma(\theta_e) + v_d (\cos \theta_e - 1) \\ v_d \sin \theta_e - x_e \sigma(\theta_e) \end{bmatrix} \quad (2.82)$$

$$f_1(t, z_1) = -\sigma(\theta_e) \quad (2.83)$$

Considérons une transformation d'état défini par :

$$z_1 = \begin{bmatrix} x_e \\ y_e \end{bmatrix}, z_2 = \theta_e \quad (2.84)$$

Les dérivés des z_1 et z_2 sont :

$$\dot{z}_1 = f_1(t, z_1) + g(t, z_1, z_2) \quad (2.85)$$

$$\dot{z}_2 = -\sigma(\theta_e) \quad (2.86)$$

Vérification de A1 : Considérons le système $\dot{z}_1 = f_1(t, z_1) + g(t, z_1, z_2)$, nous choisissons la fonction Lyapunov décrite par :

$$V = \frac{1}{2} x_e^2 + \frac{1}{2} y_e^2 \quad (2.87)$$

Ensuite, on calcule la dérivée temporelle (2.87) sur les sous-systèmes (2.69) et (2.70), nous avons :

$$\begin{aligned} \dot{V} &= \dot{x}_e x_e + \dot{y}_e y_e = (w_d y_e - v + v_d) x_e + (-w_d x_e) y_e \\ &= (-k_0 w_d x_e \operatorname{sgn}(w_d)) x_e + (-w_d x_e) y_e \\ &= -k_0 w_d y_e^2 \operatorname{sgn}(w_d) = -k_0 w_d x_e^2 \operatorname{sgn}(w_d) \end{aligned} \quad (2.89)$$

On note $w_d \operatorname{sgn}(w_d) > 0$ et $k_0 > 0$, on obtient :

$$\dot{V} \leq 0 (\dot{V} = 0 \text{ si et seulement si } x_e = 0 \text{ et } y_e = 0)$$

Donc $\dot{z}_1 = f_1(t, z_1) + g(t, z_1, z_2)$ est global uniformément stable et satisfait le reste d'A1.

Vérification d'A2 : Considérons le système $\dot{z}_2 = -\sigma(\theta_e)$, de la démonstration du théorème 2, nous pouvons tirer une conclusion que ce système satisfait A2

Vérification de A3 : Considérons le système (2.85), nous choisissons la fonction de Lyapunov décrite par

$$W = \frac{1}{2} x_e^2 + \frac{1}{2} y_e^2 \quad (2.90)$$

Ce qui est évidemment une fonction décroissante. En prenant la dérivée (2.90) le long du système (2.87), nous obtenons :

$$\begin{aligned} \dot{W} &= \dot{x}_e x_e + \dot{y}_e y_e \\ &= -k_0 w_d x_e^2 \operatorname{sgn}(w_d) - k_0 w_d y_e^2 \operatorname{sgn}(w_d) + v_d [x_e (\cos \theta_e - 1) + y_e \sin \theta_e] \\ &\leq v_d [x_e (\cos \theta_e - 1) + y_e \sin \theta_e] \leq 2 v_d^{\max} \|z_1\| \end{aligned}$$

On prend $\gamma_1(\eta) = \eta$, nous pouvons vérifier que le système (2.86) satisfait A3. Par conséquent,

II.4.7 Les Résultats et simulations

Pour démontrer l'efficacité des lois de contrôle proposées, nous simulons le modèle d'erreur différentiel cinématique de robots mobiles.

Les lois de commande

Les surfaces de glissements sont choisies comme :

$$s_2 = x_e - k_0 y_e \operatorname{sgn}(w_d) \quad (2.92)$$

Donc les lois de commandes sont :

- La vitesse angulaire

$$w = w_d + \theta_e + \alpha_1 \theta_e + \beta_1 \theta_e^{q_1/p_1} \quad (2.93)$$

- La vitesse linéaire

$$v = v_d + w_d y_e + k_0 w_d x_e \operatorname{sgn}(w_d) + \alpha_2 s + \beta_2 s^{q_2/p_2} \quad (2.94)$$

Durant la première partie de la simulation on va choisir $\alpha_2 = 8$

Trajectoire circulaire

- Le suivi d'une ligne courbe avec des vitesses souhaitées indiquées par $v_d(t) = 1, w_d(t) = 1$

La position désirée $q_d = (x_d, y_d, \theta_d)^T$ est spécifié comme :

$$\begin{cases} x_d(t) = \cos\left(\frac{\pi t}{15}\right) \\ y_d(t) = \sin\left(\frac{2\pi t}{15}\right) \\ \theta_d(t) = w_d(t) = t \end{cases} \quad (2.95)$$

Nous employons les lois de commande, où tous les paramètres sont donnés par :

$\alpha_1 = 4, \beta_1 = 8, p_1 = 7, q_1 = 5, p_2 = 5, q_2 = 3, k_0 = 1, \beta_2 = 2$. Dans la simulation, nous prenons les erreurs de position initiale.

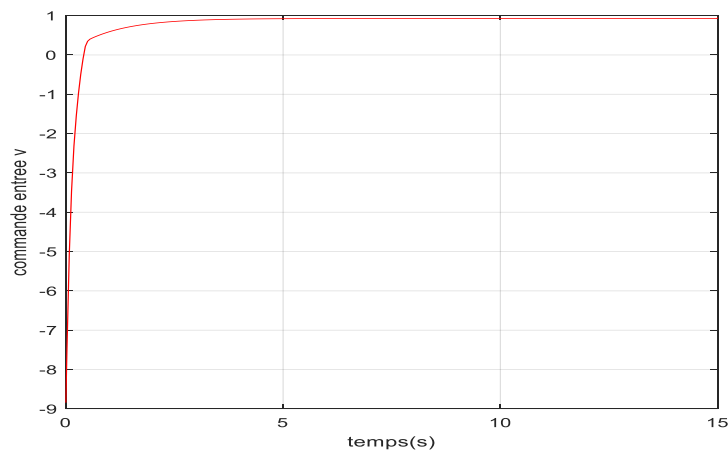


Figure II.3 : commande d'entrée v.

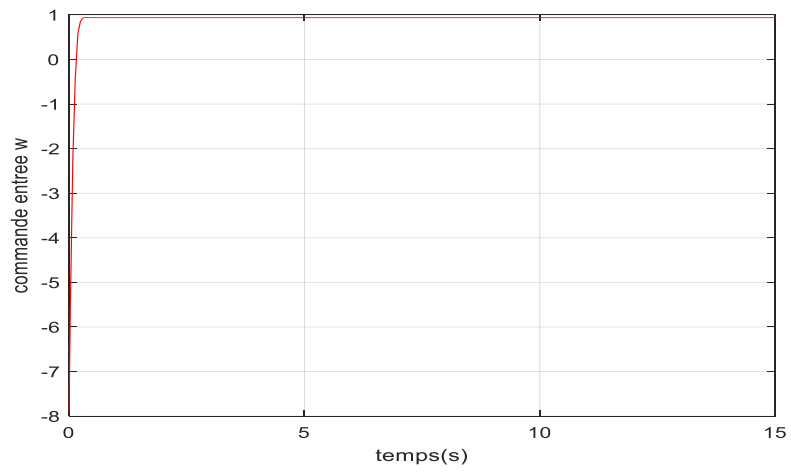


Figure II.4 : commande d'entréew.

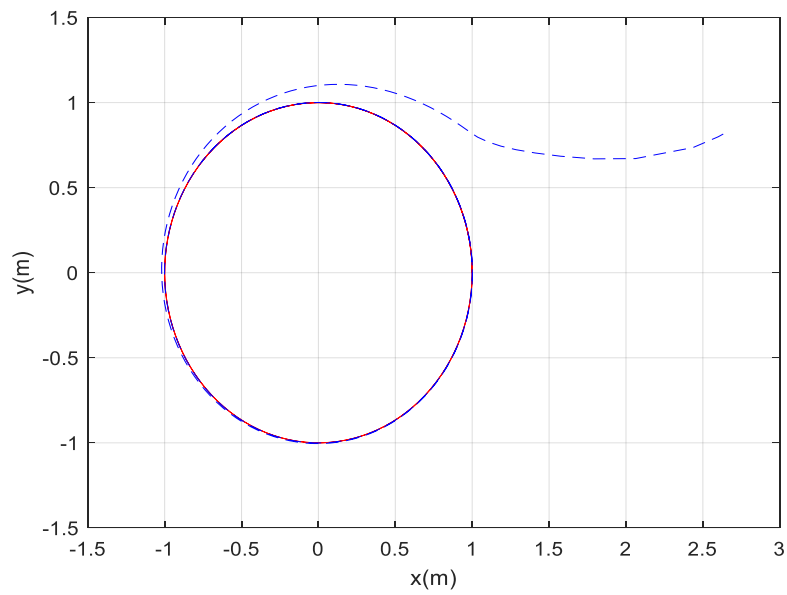


Figure II.5 : Poursuite de trajectoire circulaire

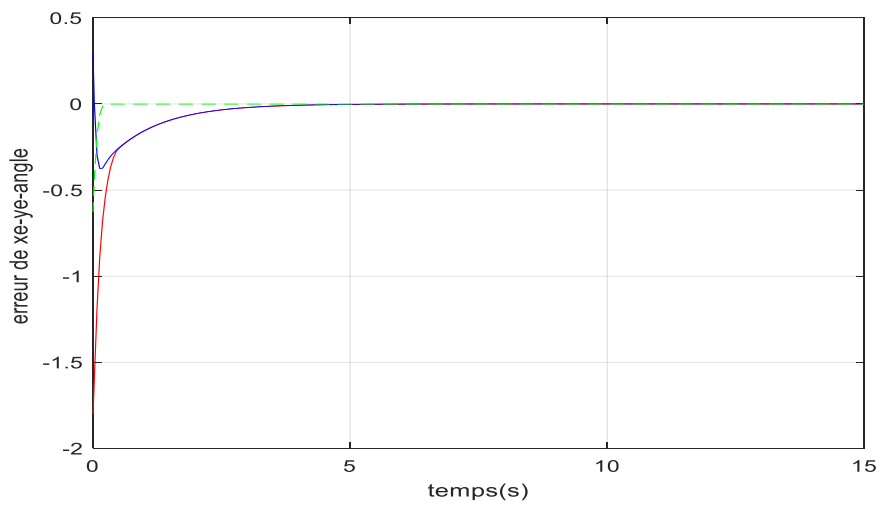


Figure II.6 : Les erreurs de position et l'erreur angulaire

Les figures (II.3) et (II.4) représentent la vitesse linéaire et la vitesse angulaire qui rejoignent les vitesses de référence. La figure (II.5) représente le suivi de trajectoire circulaire dont le robot suit son trajectoire de référence. La figure (II.6) présente les erreurs de position et l'erreur angulaire qui converge vers zéro après un temps très limite qui peut traduire par l'efficacité de la commande proposée.

II.5. Commande dynamique appliquée à la poursuite de trajectoire

Deux commandes sont proposées dans ce chapitre où la commande cinématique qui se change. La première stratégie est basée sur la commande backstepping pour la commande cinématique et un contrôleur PD en dynamique, la deuxième est basée sur la commande mode glissant rapide en cinématique et un contrôleur PD en dynamique.

II.5.1 Commande dynamique PID -backstepping

Le contrôleur cinématique pour un robot mobile non holonomique est proposé pour la première fois en 1992 par Kanayama [22] et est utilisé par tant d'autres chercheurs dans ce domaine. Une règle de contrôle stable de la trajectoire pour un robot mobile non holonomique qui néglige la dynamique du véhicule et repose sur le système de direction est décrite. Dans ce système de contrôle, deux postures pour le robot vont être utilisées : la posture de référence $q_r = [x_r \ y_r \ \theta_r]^T$ et une posture actuelle $q_c = [x_c \ y_c \ \theta_c]^T$. La posture de référence est la posture de but et la posture actuelle est la posture réelle en ce moment. Le schéma fonctionnel de cette structure de commande est illustré à la figure (II.7) :

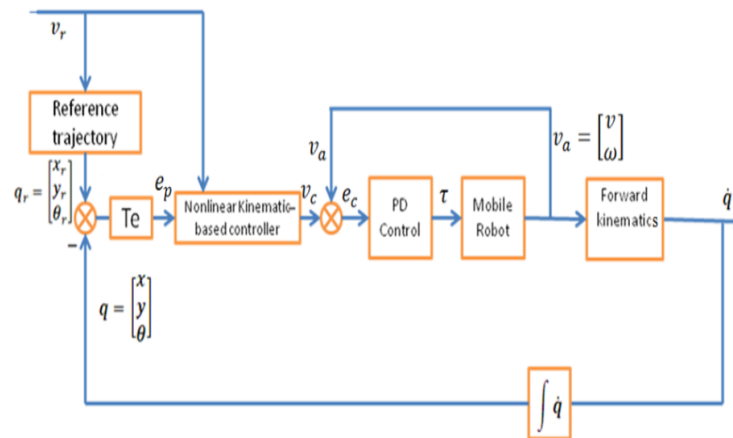


Figure II.7: stratégie de contrôle PID-backstepping.

Nous définissons une posture d'erreur ou l'erreur de suivi e_p dans la base de la trame liée à la plateforme mobile ou à la trame locale comme suit :

$$e_p = \begin{bmatrix} e_x \\ e_y \\ e_\theta \end{bmatrix} = T_e(q_r - q) \tag{2.96}$$

$$\begin{bmatrix} e_x \\ e_y \\ e_\theta \end{bmatrix} = \begin{bmatrix} \cos\theta & \sin\theta & 0 \\ -\sin\theta & \cos\theta & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x_r - x \\ y_r - y \\ \theta_r - \theta \end{bmatrix} \quad (2.97)$$

Le problème de contrôle dans ce cas sera de calculer une règle de contrôle pour le véhicule, qui a calculé les vitesses cibles $v_c = f(e_p, v_r, K)$ qui rend le système asymptotiquement stable. La règle de contrôle cinématique proposée est la suivante :

$$v_c = \begin{bmatrix} v_r \cos e_\theta + K_x e_x \\ \omega_r + K_y v_r e_y + K_\theta v_r \sin e_\theta \end{bmatrix}$$

$$v_c = f(e_p, v_r, K) \quad (2.98)$$

$$K = (K_x, K_y, K_\theta)$$

Où : K_x, K_y et K_θ sont des constantes positives.

La modélisation dynamique en général est l'étude du mouvement du système dans laquelle les forces sont modélisées et elle peut inclure les énergies et les vitesses associées aux mouvements. La principale différence entre la modélisation dynamique et cinématique est qu'en cinématique, nous étudions le mouvement sans tenir compte des forces qui affectent le mouvement et nous traitons uniquement des relations géométriques qui régissent le système.

Le modèle dynamique général du robot mobile peut être décrit par l'équation suivante :

$$M(q)\ddot{q} + V(q, \dot{q})\dot{q} + F(\dot{q}) + G(q) + \tau_d = B(q)\tau - A^T(q)\lambda \quad (2.99)$$

$M(q)$ est la matrice d'inertie symétrique définie positive ; (q, \dot{q}) est la matrice centripète et de Coriolis, $F(\dot{q})$ est la matrice de frottement de surface, $G(q)$ est le vecteur gravitationnel, τ_d Noté inconnu borné perturbations incluant des dynamiques non structurées non modélisées, $B(q)$ est la matrice de transformation d'entrée, est le vecteur d'entrée, $A^T(q)$ est la matrice associée aux contraintes, est le vecteur des forces de contraintes.

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

$$v_a(t) = \begin{bmatrix} v \\ \omega \end{bmatrix} = \begin{bmatrix} v_1 \\ v_2 \end{bmatrix} \quad (2.100)$$

$$S(q) = \begin{bmatrix} \cos\theta & -\sin\theta \\ \sin\theta & \cos\theta \\ 0 & 1 \end{bmatrix} \quad (2.101)$$

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

Par les définitions appropriées suivantes, nous pouvons réécrire l'équation dynamique comme suit :

$$\bar{M}(q)v_a(t) + \bar{V}_m(q, \dot{q})v_a(t) + \bar{F}(\dot{q}) + \bar{G}(q) + \bar{\tau}_d = \bar{B}(q)\tau \quad (2.103)$$

Avec :

$$\bar{M}(q) = S^T(q)M(q)S(q) \quad (2.104)$$

$$\bar{F}(\dot{q}) = S^T(q)F(\dot{q}) = 0 \quad (2.105)$$

$$\bar{G}(q) = S^T(q)G(q) = 0 \quad (2.106)$$

$$\bar{\tau}_d = S^T(q)\tau_d \quad (2.107)$$

$$\bar{B}(q) = S^T(q)B(q) \quad (2.108)$$

$$S^T(q)A^T(q)\lambda = 0 \quad (2.109)$$

$$\bar{V}_m = S^T(q)M(q)\dot{S}(q) + S^T(q)V_m^T(q, \dot{q})S \quad (2.110)$$

Notre objectif de contrôle est établi comme suit : Étant donné une trajectoire q_d et une orientation souhaitées du robot mobile, nous devons concevoir un contrôleur qui applique un couple adéquat tel que les positions mesurées q atteignent la référence q_d souhaitée représentée par l'équation suivante:

$$\lim_{t \rightarrow \infty} (\|q_d(t) - q(t)\|) = 0 \quad (2.111)$$

Le but du contrôleur PID est de trouver une entrée de commande telle que le vecteur vitesse actuel v atteigne le vecteur vitesse v_c ceci est noté dans l'équation suivante :

$$\lim_{t \rightarrow \infty} (\|v_c - v\|) = 0 \quad (2.112)$$

Comme le montre la figure de stratégie de commande, fondamentalement, le contrôleur PID a 2 variables d'entrée correspondant aux erreurs de vitesse obtenues (notées e_v et e_ω : erreurs de vitesse linéaire et angulaire respectivement), et 2 variables de sortie, les couples d'entrée d'entraînement et de rotation.

$$\dot{V} = k_p e_v + \int_0^t k_i e_v(\tau) + k_d \frac{de_v}{dt} \quad (2.113)$$

Où :

$$e_v = \begin{vmatrix} e_v \\ e_w \end{vmatrix} = \begin{vmatrix} v_c - v \\ w_c - w \end{vmatrix}$$

Le contrôleur PD est un contrôleur linéaire qui est responsable de convertir la vitesse de sortie du contrôleur en entrées de couple pour le robot. La stabilité de la règle de contrôle ci-dessus sera démontrée en utilisant la méthode de stabilité Lyapunov dans la section suivante.

II.5.1.1 Analyse de la stabilité du Lyapunov

L'analyse de stabilité Lyapunov de la règle de contrôle dans l'équation (2.98) est décrite comme suit:

1- Selon l'équation (2.97), nous avons :

$$\begin{bmatrix} \dot{e}_x \\ \dot{e}_y \\ \dot{e}_\theta \end{bmatrix} = \dot{e}_p = f(t, e_p) = \begin{bmatrix} \omega(e_p, q_r) e_y - v(e_p, q_r) + v_r \cos e_\theta \\ -\omega(e_p, q_r) e_x + v_r \sin e_\theta \\ \omega_r - \omega(e_p, q_r) \end{bmatrix} \quad (2.114)$$

En utilisant les équations cinématiques du robot et l'équation d'erreur de suivi (2.112), nous avons :

$$e_p = \begin{bmatrix} \cos\theta & \sin\theta & 0 \\ -\sin\theta & \cos\theta & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x_r - x \\ y_r - y \\ \theta_r - \theta \end{bmatrix} = \begin{bmatrix} e_x \\ e_y \\ e_\theta \end{bmatrix} \quad (2.115)$$

$$\begin{aligned} \dot{e}_x &= (\dot{x}_r - \dot{x}) \cos\theta + (\dot{y}_r - \dot{y}) \sin\theta - (x_r - x) \dot{\theta} \sin\theta + (y_r - y) \dot{\theta} \cos\theta = e_y \omega - v + \\ \dot{x}_r \cos(\theta_r - \theta) + \dot{y}_r \sin(\theta_r - \theta) &= e_y \omega - v + v_r \cos e_\theta \end{aligned} \quad (2.116)$$

$$\begin{aligned} \dot{e}_y &= -(\dot{x}_r - \dot{x}) \sin\theta + (\dot{y}_r - \dot{y}) \cos\theta - (x_r - x) \dot{\theta} \cos\theta - (y_r - y) \dot{\theta} \sin\theta = -e_x \omega - \\ \dot{x}_r \sin(\theta_r - \theta) + \dot{y}_r \cos(\theta_r - \theta) &= -e_x \omega + v_r \sin e_\theta \end{aligned} \quad (2.117)$$

$$\dot{e}_\theta = \dot{\theta}_r - \dot{\theta} = \omega_r - \omega \quad (2.118)$$

En substituant v et ω par $v(e_p, q_r)$ et $\omega(e_p, q_r)$

Si nous utilisons la règle de contrôle dans l'équation (4.8), $e_p = 0$ est un point d'équilibre stable si la vitesse de référence $v_r > 0$.

Preuve : Proposons une fonction scalaire v comme candidat de fonction Lyapunov pour le système ci-dessus :

$$V = \frac{1}{2} (e_x^2 + e_y^2) + \frac{(1 - \cos(e_y))}{K_y} \quad (2.119)$$

Clairement $V \geq 0$ et $V = 0$ si $e_p = 0$, donc la fonction v ci-dessus est une fonction positive définie. De plus, en utilisant loi 2, nous avons :

$$\dot{V} = \dot{e}_x e_x + \dot{e}_y e_y + \frac{\dot{e}_\theta \sin(e_\theta)}{K_y} = \left[\left(\omega_r + v_r (K_y e_y + K_\theta \sin(e_\theta)) \right) e_y - K_x e_x \right] e_x + \left[- \left(\omega_r + v_r (K_y e_y + K_\theta \sin(e_\theta)) \right) e_x + v_r \sin(e_\theta) \right] e_y + \frac{[-v_r (K_y e_y + K_\theta \sin(e_\theta))] \sin(e_\theta)}{K_y} = -K_x e_x^2 - \frac{v_r K_\theta \sin^2 e_\theta}{K_y} \leq 0 \tag{2.120}$$

Par conséquent, la dérivée de la fonction proposée de Lyapunov V est une fonction définie négative qui démontre que le point $e_p = 0$ est uniformément asymptotiquement stable dans les conditions où v_r et ω_r sont continus et où v_r, ω_r, K_x, K_y et K_θ sont limités. L'analyse de stabilité Lyapunov ci-dessus est utilisée dans la référence [22] pour prouver la stabilité du contrôleur proposé.

Nous avons démontré que le système est stable pour toute combinaison de K_x, K_y et K_θ . Cependant, comme nous avons besoin d'une réponse non oscillatoire, mais pas trop lente du robot, nous devons trouver un paramétrage optimal pour ce contrôleur. A ce stade, les gains appropriés pour chaque trajectoire de référence doivent être trouvés par le réglage.

II.5.1.2 Résultats de simulation

La règle de contrôle basée sur la cinématique dans l'équation (2.98) est ajoutée au modèle complet du robot avec la dynamique pour voir ses performances sur le système réel. Le modèle Matlab Simulink utilisé pour simuler ce contrôleur est illustré à la figure (II.8) :

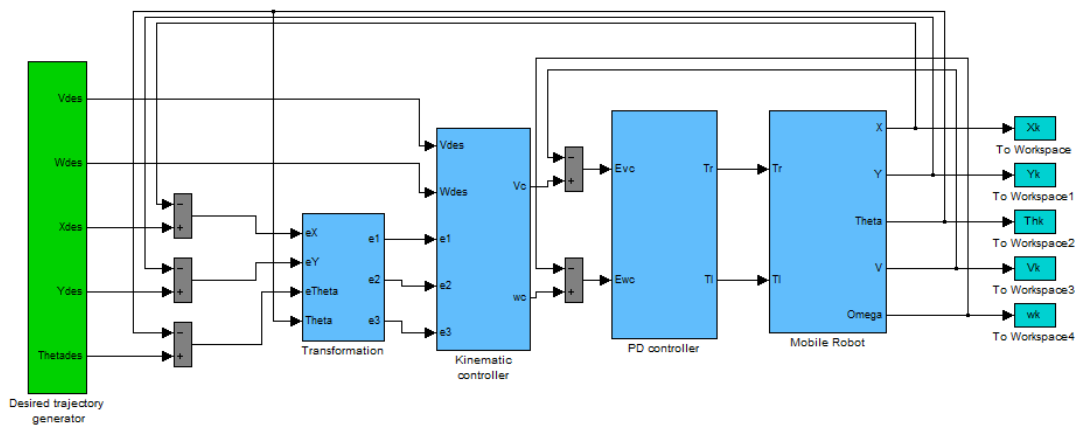


Figure II.8 : Schéma fonctionnel de simulation

En utilisant le schéma de la figure (II.8) et le bloc générateur de trajectoire en fonction de temps de simulation, nous pouvons tester les commandes proposées pour une trajectoire circulaire qui est trajectoire de référence :

Gains du contrôleur cinématique : $K_x = 50$ $K_y = 50$ $K_\theta = 100$

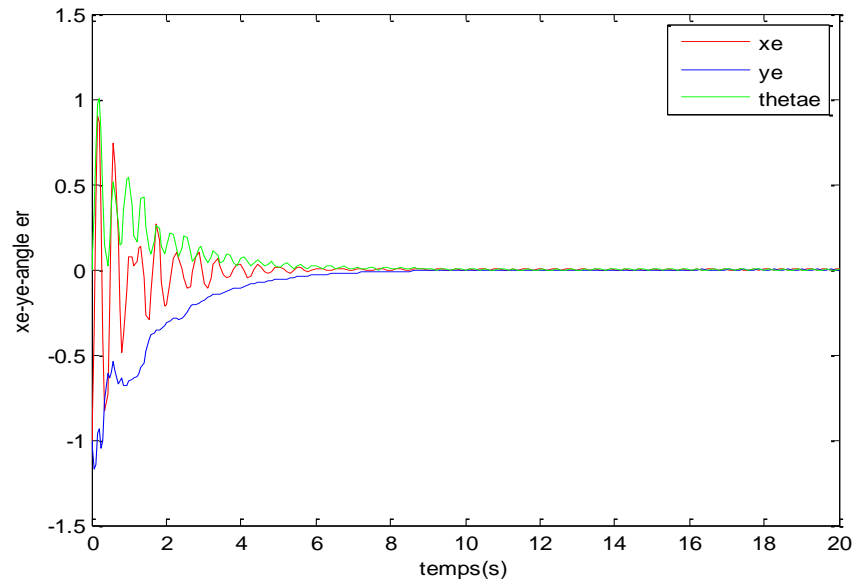


Figure II.9 : Les erreurs de position et l'erreur angulaire

La figure (II.9) montre la trajectoire du robot et la trajectoire de référence dans le plan x-y. Comme on peut le voir, la réponse du robot est lente et l'erreur de trajectoire est d'environ zéro après plus de 7 secondes. L'erreur de trajectoire illustrée est calculée comme suit :

$$e_{tr} = \sqrt{e_x^2 + e_y^2} = \sqrt{(x - x_r)^2 + (y - y_r)^2} \quad (2.121)$$

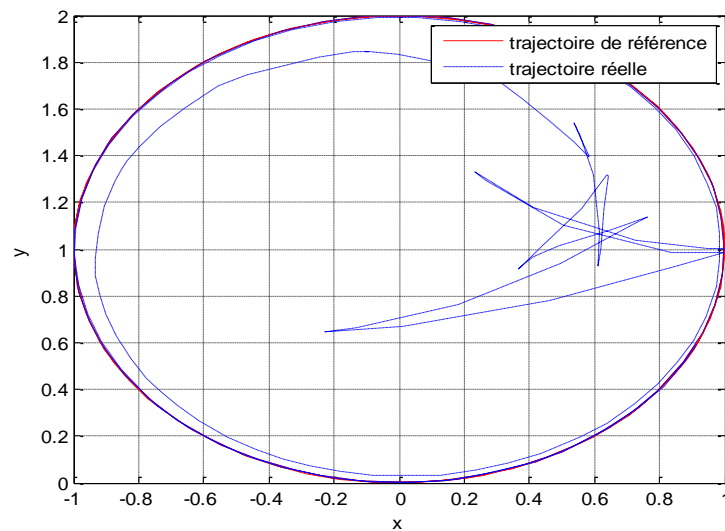


Figure II.10 : Suivi de trajectoire circulaire

La figure précédente présente le suivi de trajectoire circulaire, dont le robot suit sa trajectoire de référence.

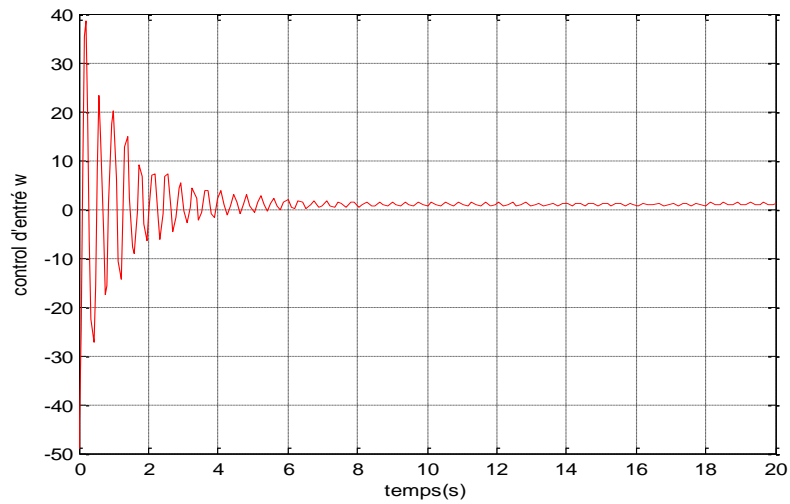


Figure II.11 : Control d'entré v

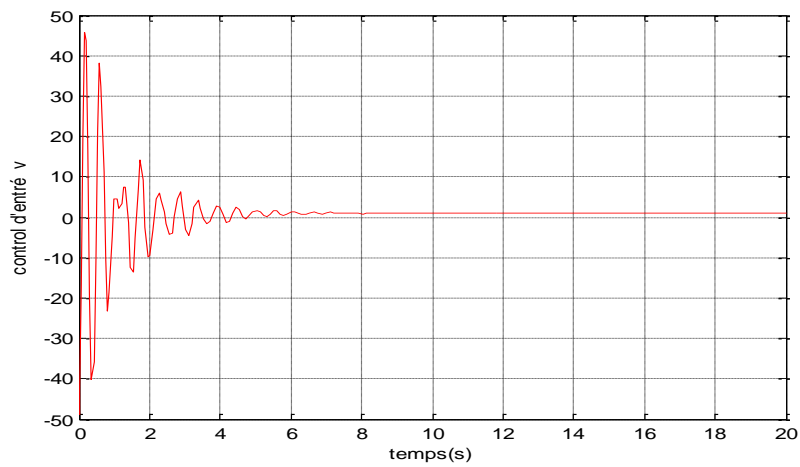


Figure II.12 : Control d'entré ω

Les deux figures ci-dessus présentent les lois de control qui sont la vitesse linéaire et la vitesse angulaire qui rejoignent les vitesses de référence qui valent 1.

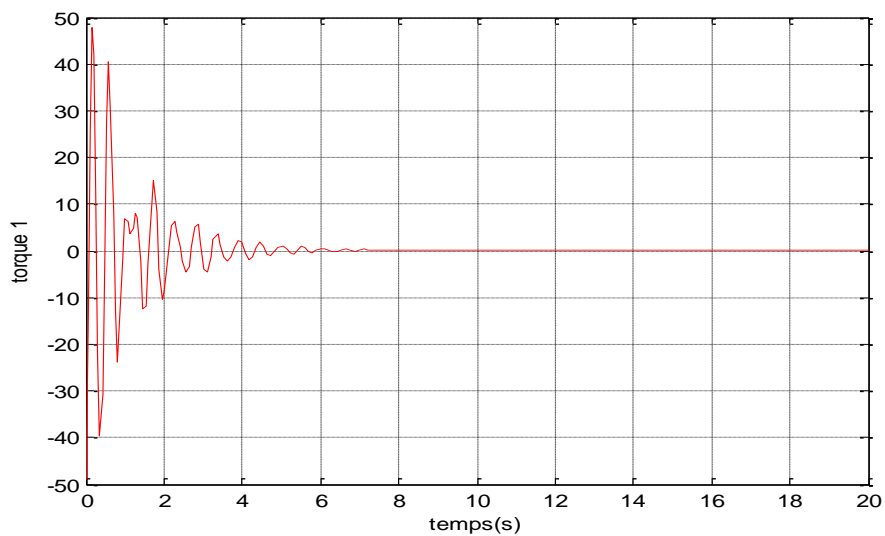


Figure II.13 : Control d'entré T_1 (torque1)

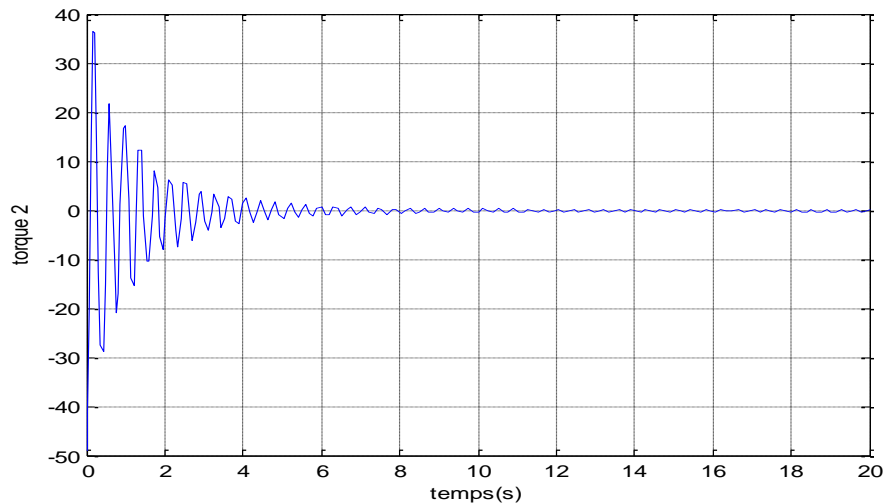


Figure II.14: Control d'entrée T2 (torque2)

Les figures (II.13), (II.14) présentent les torques qui rejoignent zéro une fois que le robot converge vers la trajectoire de référence.

Les résultats de simulation montrent l'efficacité de la commande proposée.

II.5.2 Commande dynamique PID –mode glissant rapide (PID-FTSM)

Notre objectif de contrôle est établi comme suit: Compte tenu d'une trajectoire souhaitée q_d et de l'orientation du robot mobile, nous devons concevoir un contrôleur qui applique un couple τ adéquat de sorte que les positions mesurées q atteignent la référence souhaitée q_d .

II.5.2.1 Conception d'un contrôleur en mode glissant rapide

Nous avons basé sur le mode glissant terminal rapide (FTSMC), l'idée derrière cette stratégie est d'utiliser l'approche de contrôle du mode glissant terminal pour assurer la convergence en temps fini des erreurs de suivi à zéro. FTSMC est défini parce que le phénomène de bavardage (chattering) est évitable, l'approche est basée sur la loi d'atteinte du taux de puissance car elle réduit le taux lorsque l'état est proche du collecteur.

La commande TSM (Terminal sliding mode) ajoute des fonctions non linéaires dans la conception du plan supérieur coulissant; la commande TSM existante peut ne pas fournir les mêmes performances de convergence lorsque l'état du système est loin de l'équilibre. De plus, il n'y a pas de fonction de commutation dans le contrôle du mode de glissement terminal, par conséquent, le phénomène de bavardage est évitable. Le modèle FTSM (fast terminal sliding mode) est capable de combiner les avantages de la commande TSM et de la commande classique en mode coulissant (hyper plane linéaire) ensemble de sorte que la convergence transitoire rapide (temps fini) à la fois à distance et à une plage proche de l'équilibre peut être obtenue.

Considérant les surfaces de glissement linéaires et terminales, une nouvelle surface de glissement terminale rapide globale proposée par Park et al comme suit:

$$s = \dot{x} + \alpha x + \beta x^{q/p} = 0 \quad (2.122)$$

Où : $x \in \mathbb{R}$ est l'état, $\alpha, \beta > 0$, p, q ($p > q$) sont positifs des nombres impairs.

Équation (2.122) donne :

$$\dot{x} = -\alpha x - \beta x^{q/p} \quad (2.123)$$

Pour résoudre le problème de contrôle de suivi pour le modèle cinématique, nous avons suivi la procédure proposée par Fierro et Lewis [33]. Supposons que la trajectoire de référence q_r satisfasse

$$\dot{q}_r = \begin{bmatrix} \dot{x}_r \\ \dot{y}_r \\ \dot{\theta}_r \end{bmatrix} = \begin{bmatrix} \cos\theta_r & -\sin\theta_r \\ \sin\theta_r & \cos\theta_r \\ 0 & 1 \end{bmatrix} \begin{bmatrix} v_r \\ \omega_r \end{bmatrix} \quad (2.124)$$

Où θ_r est l'orientation désirée, v_r et ω_r désignent respectivement les vitesses linéaire et angulaire désirées.

Le robot mobile passe de la posture $p = (x, y, \theta)^T$ à la posture $p_r = (x_r, y_r, \theta_r)^T$.

$$p_e = (x_e, y_e, \theta_e)^T$$

L'équation cinématique du robot mobile représente un système non linéaire à entrées multiples. Par conséquent, la conception de la fonction de commutation est un problème difficile. Pour simplifier le problème, le mode de glissement de terminal rapide est considéré. À partir de l'équation (2.123) nous avons :

$$\dot{\theta}_e = -\alpha\theta_e - \beta\theta_e^{q/p} \quad (2.125)$$

la loi de contrôle peut être obtenue comme:

$$w = w_r + \alpha\theta_e + \beta\theta_e^{q/p} \quad (2.126)$$

Notez que lorsque θ_e converge vers 0 alors $w_r = w$, nous considérons une autre conception de contrôle à deux états cette fois:

$$\dot{x}_e = w_r y_e - v + v_r \quad (2.127)$$

$$\dot{y}_e = -w_r x_e \quad (2.128)$$

La fonction de commutation peut être conçue comme suit:

$$s = x_e - y_e \quad (2.129)$$

En concevant la loi de contrôle du mode coulissant que $s \rightarrow 0$, cette réalisation x_e converge vers y_e afin d'obtenir $x_e \rightarrow 0$ et $y_e \rightarrow 0$.

$$s = \begin{bmatrix} s_1 \\ s_2 \end{bmatrix} = \begin{bmatrix} \dot{\theta}_e + \alpha\theta_e + \beta\theta_e^{\frac{q}{p}} \\ x_e - y_e \end{bmatrix} \quad (2.130)$$

L'étape suivante consiste à concevoir un contrôleur coulissant, qui peut faire $s_1 \rightarrow 0$ et $s_2 \rightarrow 0$ pour atteindre 0, et $x_e \rightarrow y_e$. La loi d'atteinte peut être définie comme:

$$\dot{s} = -K|s|^\delta \text{sgn}(s) \quad (2.131)$$

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

Cette loi d'atteinte augmente la vitesse d'atteinte lorsque l'état est éloigné du collecteur de commutation. Cependant, il réduit le taux lorsque l'état est proche du collecteur. Le résultat est un mode d'atteinte rapide et faible.

En remplaçant (2.131) par (2.127), il peut être dérivé comme:

$$\dot{s}_2 = -k|s_2|^\delta \text{sgn}(s_2) \quad (2.132)$$

$$= w_r y_e - v + v_r + w_r x_e \quad (2.133)$$

$$q = \begin{bmatrix} v \\ w \end{bmatrix} = \begin{bmatrix} w_r y_e + v_r + w_r x_e + k|s_2|^\delta \text{sgn}(s_2) \\ w_r + \alpha\theta_e + \beta\theta_e^{\frac{q}{p}} \end{bmatrix} \quad (2.134)$$

En utilisant cette loi de contrôle, l'erreur de posture du robot mobile $p_e = (x_e, y_e, \theta_e)^T$ converge asymptotiquement vers 0.

L'entrée auxiliaire de commande de vitesse qui réalise le suivi de l'équation (2.134) est donnée par l'équation (2.135):

$$v_c = \begin{bmatrix} v_c \\ \omega_c \end{bmatrix} = \begin{bmatrix} w_r y_e + v_r + w_r x_e + k|s_2|^\delta \text{sgn}(s_2) \\ w_r + \alpha\theta_e + \beta\theta_e^{\frac{q}{p}} \end{bmatrix} \quad (2.135)$$

II.5.2.2 Commande dynamique basée sur un contrôleur PID

Le but du contrôleur PID est de trouver une entrée de commande τ telle que le vecteur vitesse courant v pour atteindre le vecteur vitesse v_c ceci soit noté dans l'équation (2.135):

$$\lim_{t \rightarrow \infty} (\|v_c - v\|) = 0 \quad (2.136)$$

Comme il est montré dans la Figure .2 fondamentalement, le contrôleur PID a 2 variables d'entrée correspondant aux erreurs de vitesse obtenues de (36) (notées e_v et e_ω : erreurs de vitesse linéaire et angulaire respectivement), et 2 variables de sortie, les couples d'entrée d'entraînement

$$\dot{V} = k_p e_v + \int_0^t k_i e_v(\tau) + k_d \frac{de_v}{dt} \quad (2.137)$$

Où:

$$e_v = \begin{bmatrix} e_v \\ e_w \end{bmatrix} = \begin{bmatrix} v_c - v \\ w_c - w \end{bmatrix} \quad (2.138)$$

$$\dot{V} = \begin{bmatrix} \dot{v} \\ \dot{w} \end{bmatrix} \quad (2.139)$$

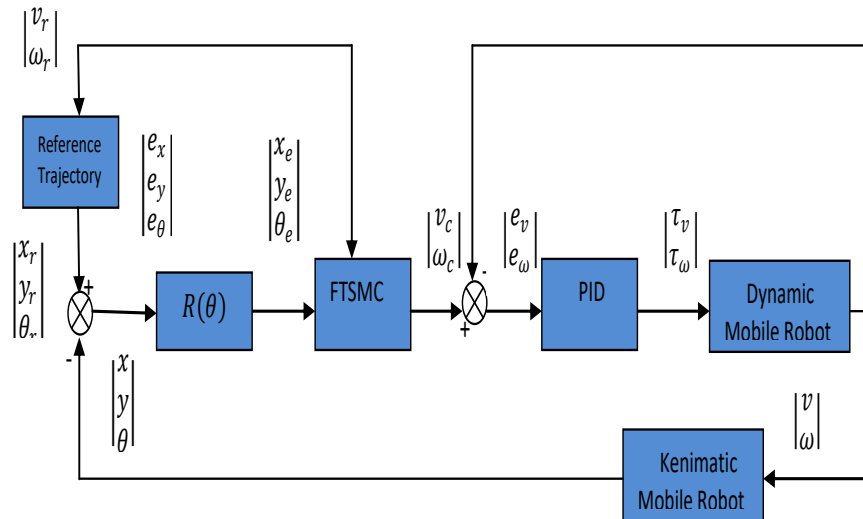


Figure (II.15) : Stratégie de commande PID- FTSM

II.5.2.3 Résultats de simulation

Selon la loi de contrôle établie, la simulation utilisant MATLAB est appliquée sur le système d'un robot mobile. Dans cette section, nous évaluons, à travers une simulation réalisée sur SIMULINK, la capacité des contrôleurs proposés à stabiliser le robot mobile. Donc nous sélectionnons une trajectoire circulaire, et nous choisis: $\begin{bmatrix} v_r \\ \omega_r \end{bmatrix} = \begin{bmatrix} 1 \\ 1 \end{bmatrix}$

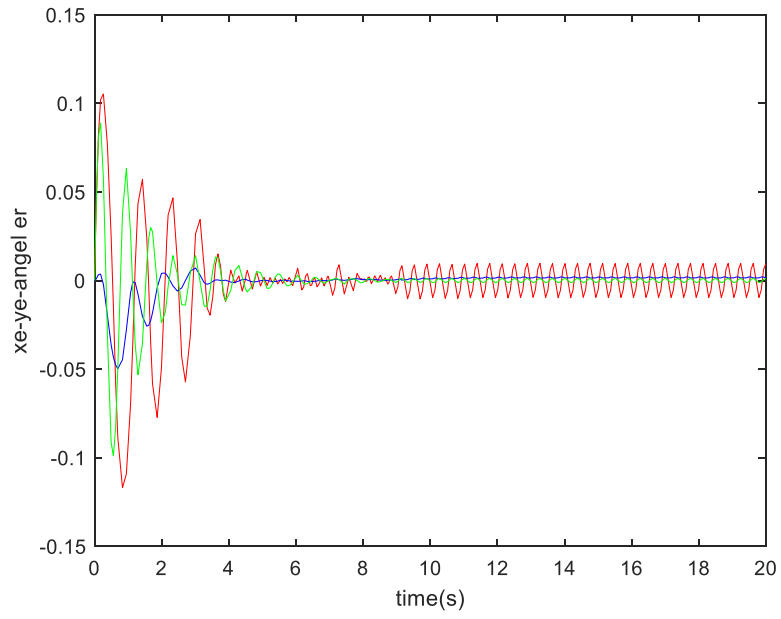


Figure (II.16) : Les erreurs de position et les erreurs angulaires

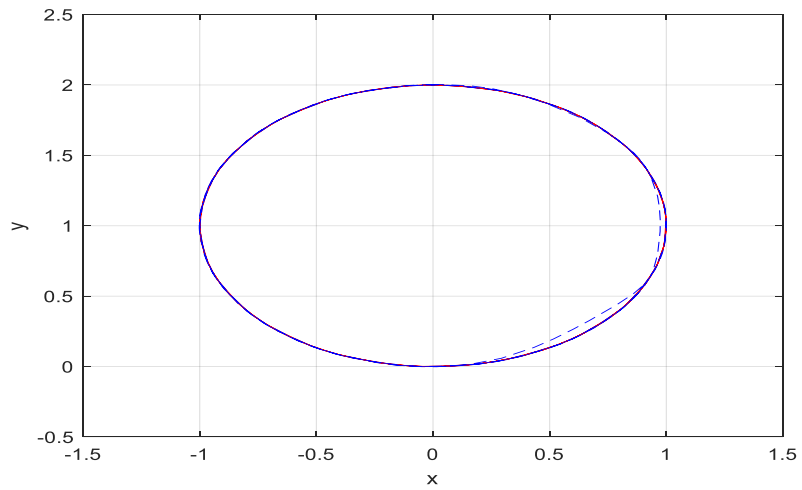


Figure (II.17) : Le suivi d'une trajectoire circulaire

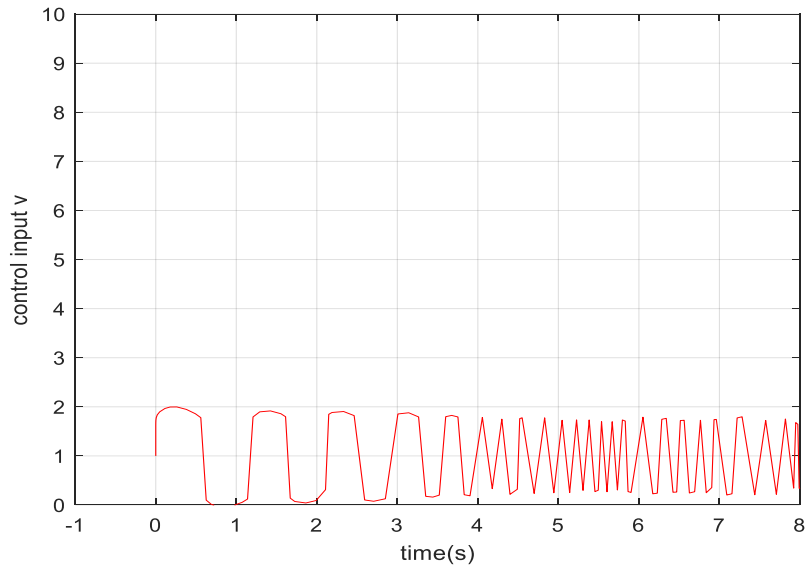


Figure (II.18) : La vitesse linéaire v

La figure (II.16) présente les erreurs de position et l'erreur angulaire qui convergent vers zéro après 4 secondes qui traduit par l'efficacité de la commande proposée.

La figure (II.17) présente le suivi de trajectoire circulaire dont le robot reçoit sa trajectoire de référence.

La figure (II.18) présente la loi de contrôle v qui est la vitesse linéaire.

La figure ci-dessous présente la loi de contrôle ω qui est la vitesse angulaire.

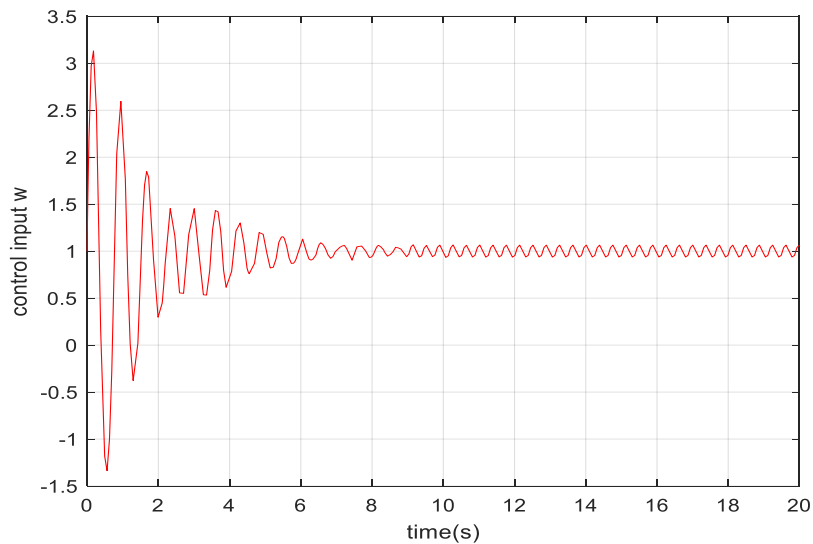


Figure (II.19) : La vitesse angulaire ω

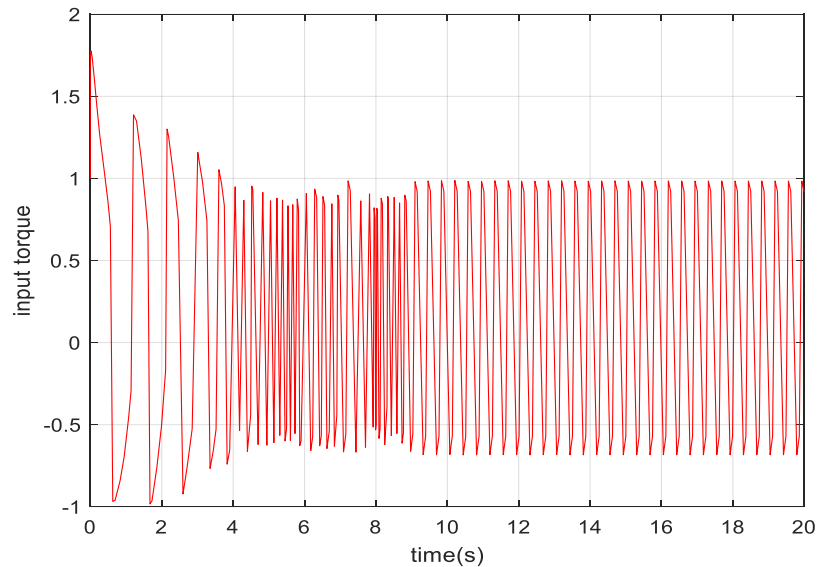


Figure (II.20) : La loi de commande (torque)

II.6 Conclusion

Au niveau de ce deuxième chapitre nous nous sommes efforcés de montrer les différentes formes de lois de commande « commande backstepping, et commande par mode glissant ».

Nous avons proposé une commande cinématique où on néglige la dynamique du robot qui est basée sur la commande par mode glissant, cette dernière prouve leur efficacité en termes de la convergence en temps fini.

Une commande dynamique est abordée en tenant compte la dynamique du robot. Deux types de commande sont proposés, une commande backstepping-PID qui montre une bonne convergence des erreurs de position et l'erreur angulaire vers zéro mais après un temps considérable. Une commande FTSM-PID qui montre ainsi la convergence rapide et après un temps très limité vers zéro.

Si on compare les deux approches proposées on trouve que la commande PID-FTSM présente une convergence rapide mais elle n'est pas efficace avec des erreurs initiales considérables, par contre la commande PID-backstepping présente une efficacité de convergence même avec des erreurs initiales importants mais en temps assez grand.

CHAPITRE III

LES RESEAUX DENEORONES

III.1 Introduction

Les réseaux de neurones, à base de structures cellulaires artificielles, constituent une approche permettant d'aborder sous des angles nouveaux les problèmes de perception, de mémoire, d'apprentissage et de raisonnement. Ils se révèlent aussi des alternatives très prometteuses pour contourner certaines limitations des méthodes classiques. Grâce à leur traitement parallèle de l'information et à leurs mécanismes inspirés des cellules nerveuses (*neurones*), ils infèrent des propriétés émergentes permettant de solutionner des problèmes jadis qualifiés de complexes [56].

Ce premier chapitre est une base théorique portant sur les principales structures de réseaux de neurones, leurs méthodes d'apprentissages ainsi que leurs domaines d'utilisation notamment en tant que réseau de neurones contrôleur.

III.2 Définition d'un réseau de neurones

Un réseau de neurones artificiel est un assemblage de cellules interconnectés chacune d'elles appelée neurone. Chaque neurone fonctionne indépendamment par rapport aux autres d'une manière parallèle ; de telle sorte que l'ensemble forme un système qui permet de réaliser des fonctions bien déterminer où des tâches complexes dans des différents types d'applications. On traite les informations qu'il reçoit par l'entraînement du réseau grâce à des méthodes d'apprentissages et non par sa programmation [57].

III.3 Neurone formel

Le neurone formel qui représente la brique de base des **RNA** est un automate dont le modèle s'inspire de celui d'un neurone biologique. On peut le décrire par les éléments suivants (pour un neurone d'indice i):

- ✓ Son état (aussi appelé activation) a_i , qui peut être une valeur réelle ou booléenne. Cet état est généralement choisi comme valeur de sortie du neurone;
- ✓ Ses connexions d'entrée auxquelles sont associés des poids m_{ij} (j est l'indice du neurone partageant la connexion);
- ✓ Sa fonction d'entrée réalisant un prétraitement (généralement une somme pondérée) des

- ✓ Sa fonction d'activation (ou de transfert) $g(x)_i$ qui calcule à partir du résultat de la fonction d'entrée et l'activation du neurone.

Le premier neurone formel est apparu en 1943 par les professeurs Mac Culloch et Pitts, leur modèle peut être représenté par la figure suivante :

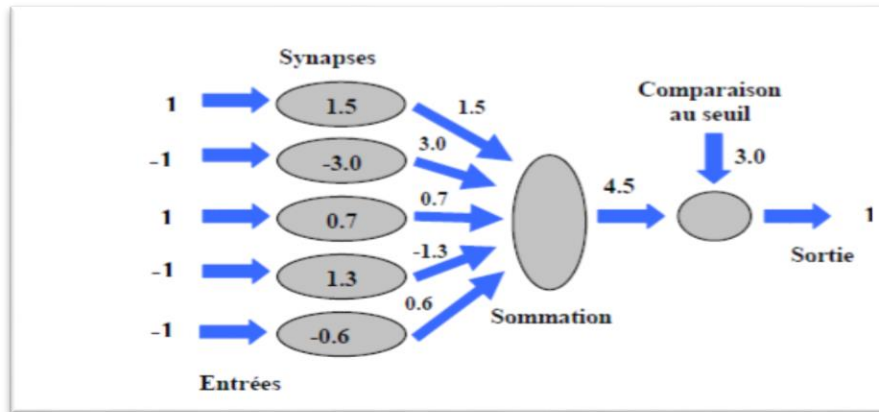


Figure III.1 : Neurone Formel.

La structure d'un neurone formel est présentée en tant qu'unité fonctionnelle élémentaire ; il est alimenté par un vecteur d'entrée \vec{X} de dimension D , $\vec{X} = (x_1, x_2, \dots, x_D) \rightarrow X =$ dont un poids de connexion ω_i (weight) est associé à chaque attribut du vecteur.

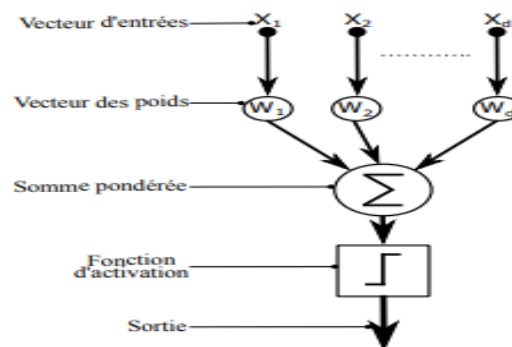


Figure III.2 : Modèle général d'un neurone.

III.3.1 Description mathématique

On considère le cas général d'un neurone formel à m entrées, auquel on doit donc soumettre les m grandeurs numériques ou signaux, notées x_1 à x_m . Un modèle de neurone formel est une règle de calcul qui permet d'associer aux m entrées une sortie : c'est donc une fonction à m variables et à valeurs réelles. Chaque entrée m est associée un poids synaptique, c'est-à-dire une valeur numérique notée de w_1 pour l'entrée 1 jusqu'à w_m pour l'entrée m . La première opération réalisée par le neurone formel consiste en une somme des grandeurs reçues en entrées, pondérées par les coefficients synaptiques, c'est-à-dire la somme :

$$W_1x_1 + W_2x_2 + \dots + W_mx_m = \sum_{j=1}^m W_jx_j \quad (3.1)$$

Cette grandeur est comparée à un seuil θ . Le résultat est alors transformé par une fonction d'activation non linéaire F . La sortie associée aux entrées x_1 à x_m est ainsi donnée par :

$$F\left(\sum_{j=1}^m W_jx_j - \theta\right) \quad (3.2)$$

Le neurone formel est l'unité élémentaire des réseaux de neurones artificiels (RNA) dans lesquels il est associé à ses semblables pour calculer des fonctions arbitrairement complexes, utilisées pour diverses applications.

l'équation mathématique du neurone peut être écrit comme suit :

$$y(k) = f(\sum_{j=1}^n w_jx_j(k) + b) \quad (3.3)$$

Les fonctions d'activation sont sélectionnées spécifiquement pour l'application des choix sont les suivants :

$$y = x \quad (3.4)$$

$$y = \frac{1}{1+e^{-x}} \quad (3.5)$$

$$y = \frac{1-e^{-x}}{1+e^{-x}} \quad (3.6)$$

$$y = \frac{e^x - e^{-x}}{e^x + e^{-x}} \quad (3.7)$$

Le but de la fonction d'activation est de modéliser le comportement non linéaire de la cellule où il n'y a pas de sortie en dessous de certaines valeurs de l'argument. Les fonctions sigmoïdes sont générales classe de fonctions monotoniement non décroissantes prenant des valeurs limitées, Pour de nombreux de réseaux de neurones algorithmes de formation, y compris l'algorithme de rétro-propagation, la dérivée de l'activation la fonction est nécessaire, donc la fonction d'activation sélectionnée devrait être différentiable.

III.3.2 Architecture des réseaux de neurones

Dans un réseau, chaque sous-groupe fait un traitement indépendant des autres et transmet le résultat de son analyse au sous-groupe suivant. L'information donnée au réseau va donc se propager couche par couche, de la couche d'entrée vers la couche de sortie, en passant par une ou plusieurs couches intermédiaires. Chaque neurone dans une couche est connecté à tous les neurones de la couche suivante et de la couche précédente. Selon la structure de connexion des neurones et le type de la fonction d'activation on peut distinguer plusieurs architectures de réseaux de neurones.

III.3.2.1 Réseaux de neurones multicouches (MLP)

Les MLP, ou réseaux à couches, forment la très grande majorité des réseaux. Ils sont

intemporels (ce sont des réseaux statiques et non dynamiques).

Un réseau de neurones consiste en un ensemble de neurones reliés entre eux par des connexions pondérées. Il se caractérise principalement par le type des unités utilisées et par sa topologie. On distingue souvent deux types de neurones particuliers dans un réseau : les neurones d'entrée recevant les données du monde extérieur, et les neurones de sortie fournissant le résultat du traitement effectué. Les autres unités sont généralement qualifiées de cachées. Cette distinction n'est toutefois pas obligatoire et tous les neurones peuvent très bien communiquer dans les deux sens avec l'extérieur.

Il est fréquent de différencier les réseaux suivant la présence ou non de cycles dans le graphe orienté des connexions entre les neurones. On parle dans le cas positif de réseaux récurrents. Il est à noter que ces connexions cycliques, dont les valeurs dépendent des activations passées des unités du réseau, permettent de mieux traiter des problèmes comportant un aspect temporel. L'apprentissage est cependant généralement assez complexe dans ces réseaux, et leurs propriétés sont souvent moins bien connues que celles réseaux non récurrents.

Un réseau à trois couches (deux couches cachées et une de sortie) et à trois entrées peut donc se représenté ainsi:

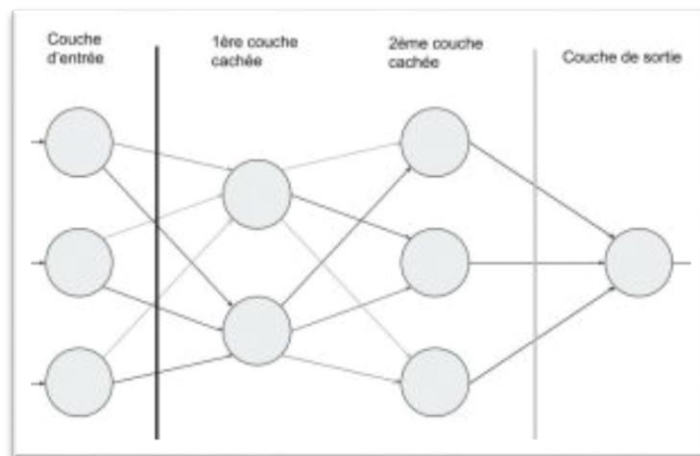


Figure III.3 : Structure d'un réseau de neurones MLP à 03 couches [13].

III.3.2.2 Réseaux de neurones à fonction de base radiale (RBF)

Les réseaux de neurones à base radiale sont une classe particulière des réseaux de neurones multicouches.

Les fonctions à base radiale (ou RBF), sont apparus à la fin des années 80 comme de nouvelles variantes des réseaux de neurones. Divers types de fonctions peuvent être utilisées comme noyau ou fonction de base, la fonction gaussienne reste cependant la plus utilisée [59].

L'idée générale des réseaux **RBF** dérive de la théorie d'approximation des fonctions, ces réseaux sont une architecture « *Feedforward* ». Ce type de réseaux a été introduit pour la

première fois par **Hardy**, et la théorie correspondante a été développée par **Powell**. La raison de son application vient du fait que le réseau utilise des fonctions gaussiennes standard qui sont à symétrie radiale. Les réseaux de neurones **RBF**, sont principalement utilisés pour résoudre des problèmes d'approximation de fonctions dans des espaces de grandes dimensions. Ils sont plus adaptés, en raison d'apprentissage local. Ce type d'apprentissage peut rendre le processus d'entraînement bien plus rapide que dans le cas d'un **MLP**, qui apprend de façon globale. On peut énumérer les caractéristiques suivantes [56]

➤ *Le nombre des couches cachées :*

Un réseau RBF ne peut contenir qu'une seule couche cachée, son architecture est fixée pour tous les problèmes à étudier.

➤ *La fonction d'activation :*

Le réseau RBF utilise toujours une fonction dite à base radiale centrée d'un point et munie d'un rayon.

➤ *Les poids synaptiques :*

Les poids entre la couche d'entrée et la couche cachée dans les modèles neuronaux de type RBF sont toujours d'une valeur d'unité, c'est-à-dire que l'information inscrite sur la couche d'entrée sera retransmise sans distorsion vers les neurones de la couche cachée. En ce qui concerne les ressemblances entre un réseau RBF et un PMC, on peut mentionner quelques points :

✓ *La fonction de sortie :*

Généralement une simple fonction linéaire qui renvoie une sommation pondérée des valeurs calculées par les neurones de la couche cachée. Bien sûr, ce n'est pas toujours le cas, parfois l'utilisation d'autres fonctions pourrait être plus adéquate dans un problème donné.

✓ *Le sens des connexions :*

Suivent le même sens, on peut dire qu'elles ne sont pas récurrentes, et chaque neurone est entièrement connecté vers les neurones de la couche suivante.

✓ *L'apprentissage :*

Pour calculer les poids de la couche de sortie, on utilise un apprentissage supervisé pour les deux types de réseaux.

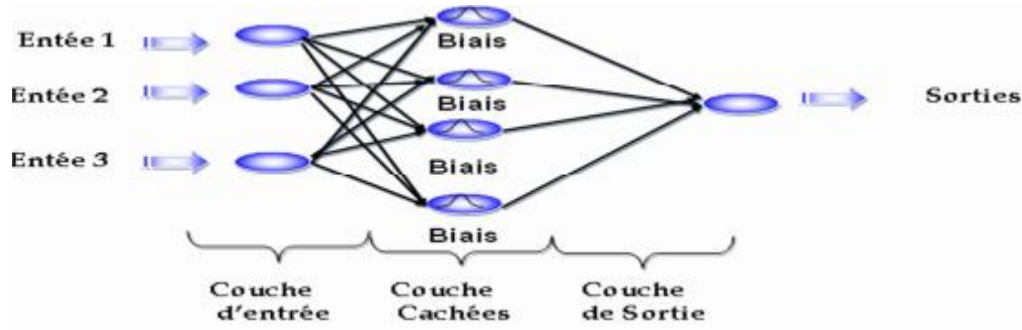


Figure III.4 : Architecture d'un Réseau de Neurone RBF.

Les neurones de la couche de cachée sont reliés aux entrées et ont chacun deux paramètres : un vecteur prototype "C" (centre des gaussiennes) et un coefficient d'étalement "a" strictement positif ; Les neurones de la couche de sortie sont généralement animés par une fonction d'activation linéaire.

La fonction réaliser par la première couche d'un réseau RBF est :

$$f_1(x) = e^{\frac{-\|x-c\|^2}{2a^2}} \quad (3.8)$$

Avec :

- C définit un point de l'espace d'entrée.
- La sortie d'un neurone est égale à 1 pour une entrée x égale à C, puis décroît vers 0 lorsque l'entrée s'éloigne de C.
- La vitesse de décroissance est réglée par : Plus le coefficient est petit et plus la fonction sera concentrée autour du point C_i et proche d'ailleurs.
- Les neurones de la seconde couche quant à eux calculent la sortie d'un réseau.

En effectuant une combinaison linéaire des sorties de ceux de la première couche et un biais est ajouté au total. La fonction qu'ils réalisent est la suivante :

$$f_2(x) = XW + b \quad (3.9)$$

Où :

- ✓ X : est un vecteur composé des sorties de tous les neurones de la première couche ;
- ✓ W : est un vecteur de poids et b est le biais.
- ✓ W et b sont ajustés lors de l'apprentissage.

Chaque sortie du réseau est donnée par la formule:

$$S_i(x) = \sum_{j=1}^N w_{ij} \exp\left(\frac{-\|s-c_j\|^2}{2a_j^2}\right) + e_i \quad (3.10)$$

Où :

- ✓ i : est le numéro de la sortie (c'est à dire le numéro du neurone de la seconde couche dont on calcule la sortie),

- ✓ m : est le nombre de neurones de la première couche,
- ✓ C_i : est le vecteur prototype du neurone numéro j de la première couche,
- ✓ σ_j : Son coefficient d'étalement,
- ✓ w_{ij} ($j=1, \dots, N$) : sont les poids du neurone de sortie,

III.4 Apprentissage des réseaux de neurones

L'apprentissage est une phase du développement d'un réseau de neurones durant laquelle le comportement du réseau est modifié jusqu'à l'obtention du comportement désiré [60].

Dans la plupart des architectures des réseaux de neurones artificiels, l'apprentissage se traduit par un changement de valeurs des poids qui relient les neurones d'une couche à l'autre. Soit le poids w_{ij} reliant le neurone i à son entrée j au temps t , un changement $\Delta w_{ij}(t)$ de poids peut s'exprimer par l'équation suivante :

$$\Delta w_{ij}(t) = \Delta w_{ij}(t+1) - w_{ij}(t) \quad (3.11)$$

Par conséquent :

$$w_{ij}(t+1) = w_{ij}(t) + \Delta w_{ij}(t) \quad (3.12)$$

Où :

$w_{ij}(t)$: représente l'ancienne valeur de poids w_{ij} .

$w_{ij}(t+1)$: représente la nouvelle valeur de poids w_{ij} .

Il existe un ensemble de règles bien définies qui permet de réaliser l'adaptation des poids constitue des algorithmes d'apprentissage du réseau. A titre d'exemple on peut citer : apprentissage par la règle de Hebb, apprentissage par la correction d'erreur, par règle *delta*...

III.4.1 Les Règles d'apprentissage

➤ *La règle de HEBB*

Cette règle vue dans le livre « Organisation of BEHAVIOR » en 1949. Elle permet de modifier la valeur des poids synaptiques en fonction de l'activité des unités qui les relient. Le but principal est le suivant : si deux unités s'activent en même temps la connexion qui les lie est renforcée (c'est une connexion excitatrice) sinon elle est affaiblie (c'est une connexion inhibitrice).

➤ *La règle delta*

Cette règle basée sur le calcul de la différence entre la valeur de la sortie et la valeur désirée pour ajuster les poids synaptiques. Elle emploie une fonction d'erreur, nommée « la moindre carré moyen », basée sur les différences utilisées pour l'ajustement des poids. Alors son but est de faire évoluer le réseau vers le minimum de cette fonction d'erreur

La règle d'apprentissage compétitif qui ne concerne qu'un seul neurone. Son principe est

de regrouper les données en catégorie. Les neurones similaires vont donc être rangés dans une même classe en se basant sur des corrélations des données et seront représentés par un seul neurone. L'architecture d'un tel réseau possède une couche d'entrée et une couche de compétition. Une forme est présentée à l'entrée du réseau. Elle est projetée ensuite, sur chacun des neurones de la couche compétitive. Le neurone gagnant est celui qui possède un vecteur de poids le plus proche de la forme présentée en entrée. Chaque neurone de sortie est connecté aux neurones de la couche d'entrée et aux autres cellules de sortie (c'est une connexion inhibitrice) ou à elle-même (c'est une connexion excitatrice). La sortie dépend alors de la compétition entre les connexions inhibitrices et excitatrices.

➤ **La rétro-propagation du Gradient**

Cette règle consiste simplement en une descente de gradient, qui est une méthode d'optimisation universelle. On cherche à minimiser une fonction d'erreur (qui représente l'erreur entre la sortie désirée et la sortie obtenue), en suivant les lignes de plus grande pente. Une fonction d'erreur rapportée à une dimension peut se représenter ainsi :

On peut se représenter la descente du gradient comme une bille que l'on poserait sur la courbe, et qui descendrait logiquement la pente (le gradient représente la pente selon chaque dimension).

L'inconvénient de cette méthode est qu'elle va s'arrêter dans le premier minimum local rencontré.

C'est pourquoi diverses améliorations ont été apportées. Diverses solutions permettent de sortir des minima locaux, visant à modifier l'allure locale de l'hyper surface d'erreur seront détaillées dans la section qui suit.

III.4.2 Les méthodes d'apprentissage

On distingue toutefois trois méthodes d'apprentissage distinctes :

➤ **l'apprentissage supervisé**

L'algorithme s'entraîne sur un ensemble de données étiquetées et se modifie jusqu'à être capable de traiter le data set pour obtenir le résultat souhaité.

➤ **l'apprentissage non-supervisé**

Les données ne sont pas étiquetées. Le réseau de neurones analyse l'ensemble de données, et une fonction-coût lui indique dans quelle mesure il est éloigné du résultat souhaité. Le réseau s'adapte alors pour augmenter la précision de l'algorithme.

➤ **l'apprentissage renforcé**

Le réseau de neurones est renforcé pour les résultats positifs et sanctionné pour les résultats négatifs. C'est ce qui lui permet d'apprendre au fil du temps, de la même manière qu'un

humain apprend progressivement de ses erreurs.

III.5 La Propriété fondamentale des réseaux de neurones

Les réseaux de neurones formels, tels que nous les avons définis, possèdent une propriété remarquable qui est à l'origine de leur intérêt pratique dans des domaines très divers : ce sont des approximateurs universels.

La propriété d'approximation peut être énoncée de la manière suivante : « toute fonction bornée suffisamment régulière peut être approchée avec une précision arbitraire, dans un domaine fini de l'espace de ses variables, par un réseau de neurones comportant une couche de neurones cachés en nombre fini, possédant tous la même fonction d'activation, et un neurone de sortie linéaire ».

Contrôler un processus, c'est imposer à celui-ci un comportement défini à l'avance en fonction des signaux de commande, l'ensemble (commande, processus) peut donc être considéré comme un système qui réalise une fonction non linéaire qu'un réseau de neurones peut approcher.

III.6 L'application des réseaux de neurones

Les réseaux de neurones servent dans aujourd'hui à toutes sortes d'applications dans divers domaines. On peut citer par exemples :

- ✓ Autopilotage des avions.
- ✓ Système de guidage des automobiles.
- ✓ Lecture automatique des chèques bancaires et d'adresses postales.
- ✓ Production des systèmes de traitement signal et pour la synthèse de la parole.
- ✓ Les réseaux de neurones ils sont utilisés aussi pour les systèmes de vision par ordinateur.
- ✓ Ils sont utilisés en robotique et en télécommunication.
- ✓ Ils sont aussi utilisés dans les domaines de finance.

III.7 Conclusion

Nous venons aborder dans ce chapitre des différents modèles de réseaux de neurones, c'est-à-dire une vue globale sur les réseaux de neurones. Nous nous sommes intéressés ensuite à la description des réseaux de neurones à base radiale RBF, les différents types d'architectures et de modèle qui existent. Nous avons aussi présenté une définition de l'apprentissage des réseaux de neurones, puis les domaines d'utilisation des réseaux de neurones.

CHAPITRE IV

CONCEPTION D'UN CONTRÔLEUR BASE SUR LES RESEAUX DE NEORONES

IV.1 Introduction

Grâce à leur propriété d'approximation universelle, les réseaux de neurones ont été utilisés dans des nombreuses applications, dans l'identification, la planification et la commande des robots. Ils sont capables d'approximer n'importe quelle fonction non linéaire avec n'importe quel précision souhaitée.

Le contrôle par réseau de neurones est l'étude des systèmes de commande à base de réseaux de neurones. Un réseau de neurones contrôleur est un réseau dont la sortie est le signal de commande $u(t)$: *fonction du temps*.

L'intérêt de ce type de contrôleurs est dû à la capacité d'apprentissage des réseaux de neurones. Un réseau de neurones avec des poids initiaux aléatoires doit apprendre la dynamique inverse du système en exploitant ses entrées-sorties à travers le temps. En générale, les performances exigées sont atteintes par ces structures après un lent entraînement et par conséquent un temps considérable [55].

Le contrôleur neuronal des robots mobiles a récemment fait l'objet d'intenses recherches. Il est habituel de travailler avec des modèles cinématiques de robot mobile pour obtenir stable lois de contrôle des mouvements pour suivre la trajectoire ou atteindre l'objectif [56], [57]. La plupart des auteurs ont résolu le problème du contrôle et de la stabilité des mouvements des robots mobiles à l'aide d'un algorithme de recul non linéaire. Fierro et Lewis [58] ont développé un modèle basé sur le réseau neuronal en combinant la technique de suivi de recul et un contrôleur de couple, en utilisant un réseau neuronal d'alimentation multicouche, où le réseau neuronal peut apprendre la dynamique du robot mobile par son apprentissage en ligne. Mais l'algorithme de contrôle et l'algorithme d'apprentissage du réseau neuronal sont très compliqués et coûtent cher en calcul. Dans un contrôleur neuronal a une seule couche pour robot est proposé. Cette approche ne comprend pas de cinématique non holonomique, il ne peut donc pas être utilisé pour les robots ayant des contraintes cinématiques. Cependant, le problème du contrôle des robots mobiles a suscité l'intérêt des chercheurs face à ses défis théoriques. En fait,

ces systèmes sont un exemple typique de mécanismes non holonomiques en raison des contraintes de roulement parfaites (pas de glissement longitudinal ou latéral des roues).

Le système de contrôle de réseau neural proposé est illustré dans la figure (IV.1).

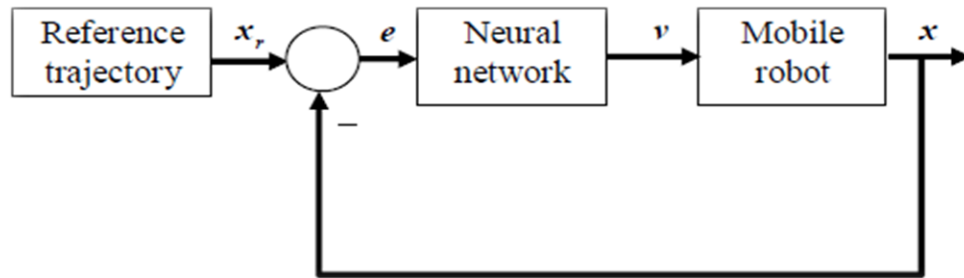


Figure IV.1 : Système mobile de contrôle des mouvements du robot.

IV.2 Conception d'un contrôleur neuronal

Avant de commencer à expliquer les détails de la conception d'un contrôleur neuronal pour le suivi de trajectoire du robot mobile, un bref historique des réseaux neuronaux qui comprend différentes topologies réseau neural, les propriétés, la sélection du poids et l'entraînement et réseau neurone l'application dans les problèmes de contrôle est présentée dans la section suivante.

L'équation du réseau est donnée par :

$$y_l(k) = f(\sum_{j=1}^n w_{lj}x_j(k) + b_l) \quad (4.1)$$

La matrice des poids et seuils comme peuvent être écrites comme suit :

$$W^T = \begin{bmatrix} w_{11} & w_{12} & \dots & w_{1n} \\ w_{21} & w_{22} & \dots & w_{2n} \\ \vdots & \vdots & \vdots & \vdots \\ w_{L1} & w_{L2} & \dots & w_{Ln} \end{bmatrix} \quad (4.2)$$

$$b_w = \begin{bmatrix} b_1 \\ b_2 \\ \vdots \\ b_L \end{bmatrix} \quad (4.3)$$

Nous pouvons écrire le vecteur de sortie $y(t) = [y_1, y_2, y_3, \dots, y_n]^T$ comme :

$$y(t) = f(w^T x + b_w) \quad (4.4)$$

Un Réseau neuronal à deux couches qui a deux couches de neurones dont une couche a des neurones L qui se nourrissent le second ayant des neurones m est illustré à la figure (IV.2) :

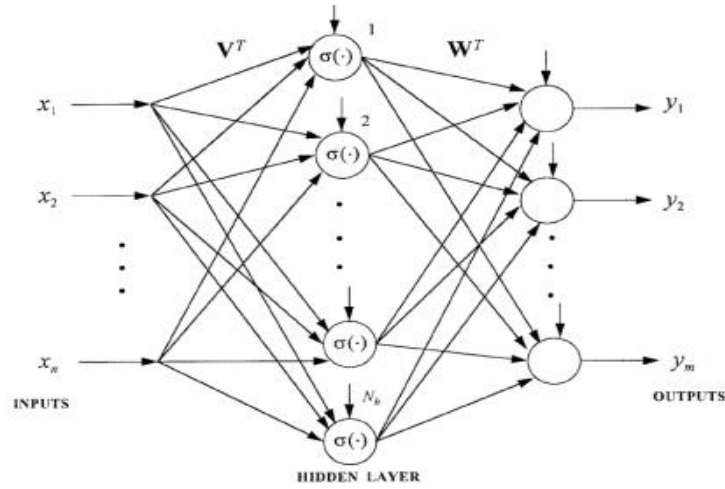


Figure IV.2 : un réseau neuronal à deux couches [13]

La première couche est appelée couche cachée et la seconde couche est appelée couche de sortie, Un réseau neuronal avec plusieurs couches est appelé un perceptron multicouche et son la puissance de calcul est considérablement améliorée sur le réseau à une couche. La sortie du réseau de deux couches ci-dessus est donnée par l'équation suivante :

$$y_i = \sigma(\sum_{l=1}^L w_{il} \sigma(\sum_{j=1}^n v_{lj} x_j + v_{l0}) + w_{i0}) \quad (4.5)$$

Si nous définissons la sortie de la couche cachée comme z_l alors nous pouvons écrire l'équation ci-dessus comme :

$$z_l = \sigma(\sum_{j=1}^n v_{lj} x_j + v_{l0}) \quad l = 1, 2, \dots, L \quad (4.6)$$

$$y_i = \sigma(\sum_{l=1}^L w_{il} z_l + w_{i0}) \quad i = 1, 2, \dots, m \quad (4.7)$$

Les deux équations ci-dessus peuvent être écrites sous forme de matrice comme les équations (4.3) et(4,4) ; Il est important de mentionner que la saisie des poids des calques cachés sera sélectionnée.au hasard tandis que les poids cachés à la couche de sortie seront ajustés en utilisant approprié méthodes de formation, Cela minimisera la complexité de calcul de l'utilisation de réseau de neurone dans les applications de contrôle.

Une structure de contrôle qui intègre le contrôleur backstepping et un réseau neuronal contrôleur de couple calculé pour les robots mobiles non holonomiques est proposé dans cette section. A la loi combinée de contrôle cinématique/couple est développée en utilisant le backstepping et la stabilité est prouvée en utilisant l'approche Lyapunov. Le contrôleur proposé dans ce travail peut traiter avec des perturbations limitées non codées et une dynamique non structurée non codée dans robot mobile.

IV.3 Application d'une commande NN-GFSM-backstepping pour la poursuite de trajectoires pour un robot mobile

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

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

Une structure de contrôle qui intègre le contrôleur backstepping et un réseau neuronal contrôleur de couple calculé pour les robots mobiles non holonomiques est proposé dans cette section. A la loi combinée de contrôle cinématique/couple est développée en utilisant le backstepping et la stabilité est prouvée en utilisant l'approche Lyapunov. Le contrôleur proposé dans ce travail peut traiter avec des perturbations limitées non codées et une dynamique non structurée non codée dans robot mobile.

IV.3.1. Contrôleur cinématique

Le contrôleur cinématique proposé est basé sur ce qui est démontré dans le chapitre deuxième chapitre qui est la commande backstepping, où les lois de commandes w_c et v_c sont présentées par les équations suivantes :

$$v_c = \begin{bmatrix} v_r \cos e_\theta + K_x e_x \\ \omega_r + K_y v_r e_y + K_\theta v_r \sin e_\theta \end{bmatrix}$$

$$v_c = f(e_p, v_r, K)$$

$$K = (K_x, K_y, K_\theta)$$

Où : K_x, K_y et K_θ sont des constantes positives.

IV.3.2. Contrôleur dynamique

Pour le contrôle dynamique, on va présenter deux cas :

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

IV.3.2.1. Contrôle en absence de perturbations

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

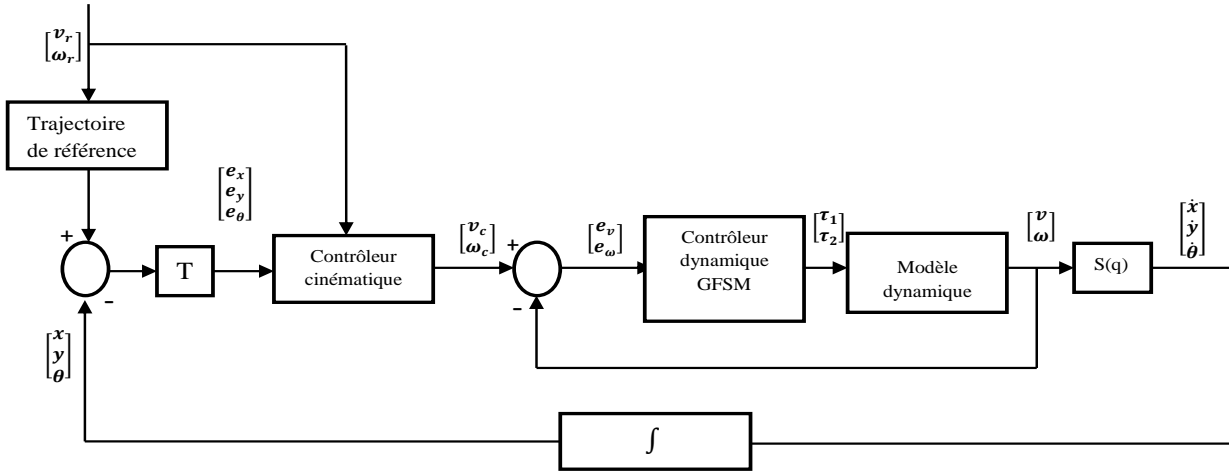


Figure IV.3 : Stratégie de la commande dynamique basée sur l'approche GFSM

Le modèle dynamique du robot mobile sans considérer les perturbations est donné par l'équation suivante :

$$Mh(q)\ddot{q} + Vh(q, \dot{q})\dot{q} = Bh(q)\tau \quad (4.8)$$

Les erreurs des vitesses sont définies par l'équation suivante :

$$e_g = \mathcal{G}_c - \mathcal{G} = \begin{bmatrix} e_v \\ e_\omega \end{bmatrix} = \begin{bmatrix} v_c - v \\ \omega_c - \omega \end{bmatrix} \quad (4.9)$$

La dérivée de l'équation (4.9) est obtenue comme :

$$\dot{e}_g = \dot{\mathcal{G}}_c - \dot{\mathcal{G}} \quad (4.10)$$

A partir de l'équation (4.10) :

$$\begin{aligned} Mh(q)(\dot{\mathcal{G}}_c - \dot{e}_g) + Vh(q, \dot{q})(\mathcal{G}_c - e_g) &= Bh(q)\tau \\ Mh(q)\dot{\mathcal{G}}_c - Mh(q)\dot{e}_g + Vh(q, \dot{q})\mathcal{G}_c - Vh(q, \dot{q})e_g &= Bh(q)\tau \end{aligned} \quad (4.11)$$

La fonction non linéaire du système est définie comme suit :

$$f(z) = Mh(q)\dot{\mathcal{G}}_c + Vh(q, \dot{q})\mathcal{G}_c \quad (4.12)$$

Avec $z = [\mathcal{G}_c^T \ \dot{\mathcal{G}}_c^T \ \mathcal{G}^T]^T$

Remplaçant l'équation (4.12) dans (4.11), on trouve :

$$-Mh(q)\dot{e}_g - Vh(q, \dot{q})e_g + f(z) = Bh(q)\tau \quad (4.13)$$

Pour obtenir la loi de commande dynamique, on choisit la surface de glissement comme :

$$s_3 = e_g \quad (4.14)$$

A partir de l'équation (3.38) qui présente la surface globale rapide et l'équation (4.14), l'équation (4.15) est établie comme suit :

$$\dot{e}_g = -(\alpha_3 + 1)e_g - \beta_3 e_g^{q_3/p_3} \quad (4.15)$$

On remplace l'équation (4.15) dans (4.13) :

$$Mh(q)((\alpha_3 + 1)e_g + \beta_3 e_g^{q_3/p_3}) - Vh(q, \dot{q})e_g + f(z) = Bh(q)\tau \quad (4.16)$$

Alors, la loi de commande obtenue est donnée par :

$$\tau = Bh^{-1}(C_v e_g + f(z) + \beta_3 Mh(q) e_g^{q_3/p_3}) \quad (4.17)$$

Avec :

$$C_v = Mh(q)(\alpha_3 + 1) - Vh(q, \dot{q}) \quad (4.18)$$

IV.3.2.2. Contrôle en présence de perturbations

Dans ce cas un contrôleur neuronal est introduit, la stratégie de contrôle est donnée par la figure suivante :

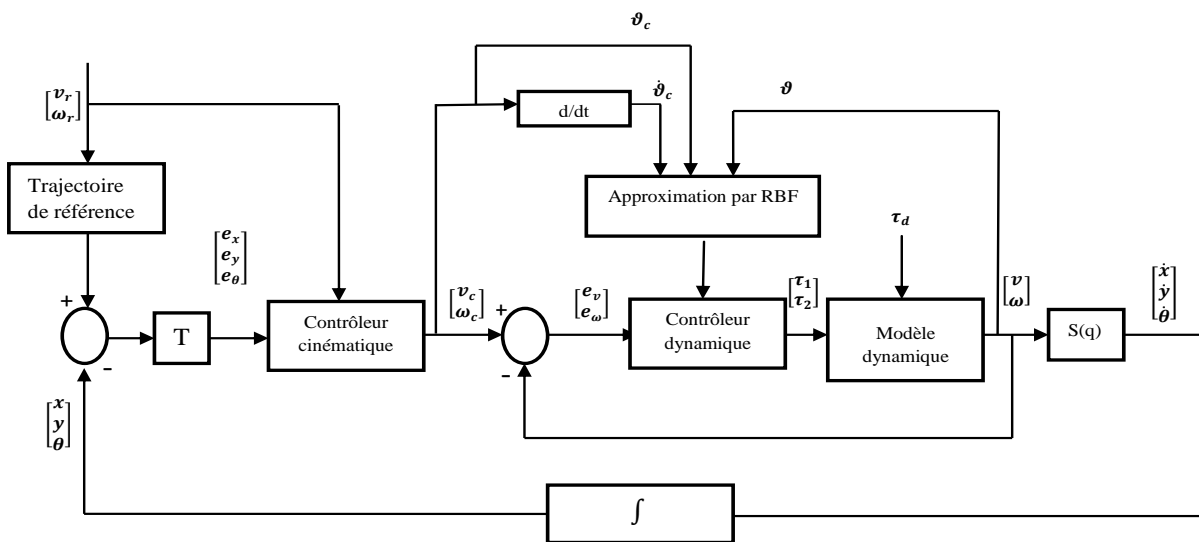


Figure IV.4: Stratégie de la commande dynamique basée sur NN-GFSM

L'équation dynamique du robot en présence des perturbations est donnée par l'équation (3.61) :

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

Donc l'équation (4.13) peut réécrire comme suit :

$$-Mh(q)\dot{e}_g - Vh(q, \dot{q})e_g + f(z) - \tau_d = Bh(q)\tau \quad (4.19)$$

La loi de contrôle désigné dans l'équation (4.17) devient :

$$\tau = Bh^{-1}(C_v e_g + \hat{f}(z) + \beta_3 Mh(q) e_g^{q_3/p_3}) \quad (4.20)$$

où $\hat{f}(z)$ est la sortie du réseau RBF, et $\hat{f}(z)$ approxime $f(z)$.

Le réseau RBF peut être adopté pour approximer $f(z)$. L'algorithme d'approximation du réseau RBF est donné comme :

$$\varphi_i = g(\|z - c_i\|^2 / b_i^2) \quad i=1, 2, \dots, n \quad (4.21)$$

$$\zeta = W^{*T} \varphi(z) \quad (4.22)$$

$$f(z) = W^{*T} \varphi(z) + \varepsilon \quad (4.23)$$

Où z est l'état d'entrée du réseau, $\varphi(z) = [\varphi_1 \varphi_2 \dots \varphi_n]^T$ est la fonction non linéaire, ε est l'erreur d'approximation et W^* est le vecteur de poids du réseau RBF.

On remplace l'équation (4.20) dans (4.19) on trouve :

$$Mh(q)\dot{e}_g = -(Vh(q, \dot{q}) + C_v) e_g + \mu_0 - \beta_3 Mh(q) e_g^{q_3/p_3} \quad (4.24)$$

Avec $\mu_0 = \tilde{f}(z) - \tau_d$ et $\tilde{f}(z) = f(z) - \hat{f}(z)$.

La sortie du réseau RBF est donnée comme:

$$\hat{f}(z) = \hat{W}^T \varphi(z) \quad (4.25)$$

On choisit $\hat{W} = W^* - \hat{W}$, $\|\hat{W}\| \leq W_{\max}$, alors:

$$\mu_0 = \tilde{f}(z) - \tau_d = \hat{W}^T \varphi(z) + \varepsilon - \tau_d \quad (4.26)$$

La loi de contrôle désignée dans l'équation (4.19) devient :

$$\tau = Bh^{-1}(C_v e_g + \hat{f}(z)) + \beta_3 Mh(q) e_g^{\frac{q_3}{p_3}} - \xi \quad (4.27)$$

Où ξ est l'élément robuste introduit pour éliminer l'erreur d'approximation du réseau ε et les perturbations τ_d .

On remplace l'équation (4.26) dans (4.24) :

$$Mh(q) \dot{e}_g = -(Vh(q, \dot{q}) + C_v) e_g - \beta_3 Mh(q) e_g^{\frac{q_3}{p_3}} + \hat{W}^T \varphi(z) + \varepsilon - \tau_d + \xi \quad (4.28)$$

On pose :

$$\mu_1 = \hat{W}^T \varphi(z) + \varepsilon - \tau_d + \xi \quad (4.29)$$

Remplaçant équation (4.29) dans l'équation (4.28), on trouve :

$$Mh(q) \dot{e}_g = -(Vh(q, \dot{q}) + C_v) e_g - \beta_3 Mh(q) e_g^{\frac{q_3}{p_3}} + \mu_1 \quad (4.30)$$

L'élément robuste est désigné comme :

$$\xi = -(\varepsilon_N + b_d) \text{sign}(e_g) \quad (4.31)$$

Où: $\|\varepsilon\| \leq \varepsilon_N$, $\|\tau_d\| \leq b_d$

La fonction candidate de Lyapunov est choisie comme suit :

$$\rho = \frac{1}{2} e_g^T Mh(q) e_g + \frac{1}{2} \text{tr}(\hat{W}^T F_w^{-1} \hat{W}) \quad (4.32)$$

La dérivée de la fonction de Lyapunov est définie comme :

$$\dot{\rho} = e_g^T Mh(q) \dot{e}_g + \frac{1}{2} e_g^T \dot{M}h(q) e_g + \text{tr}(\tilde{W}^T F_w^{-1} \tilde{W}) \quad (4.33)$$

A partir de l'équation (4.28) on trouve :

$$\dot{\rho} = -e_g^T Mh(q) (\alpha + 1) e_g - e_g^T \beta Mh(q) e_g^{\frac{q_3}{p_3}} + \text{tr} \tilde{W}^T (F_w^{-1} \tilde{W} + \varphi(z) e_g^T) + e_g^T (\varepsilon - \tau_d + \xi) \quad (4.34)$$

On choisit : $\tilde{W} = F_w \varphi(z) e_g^T$

La loi d'adaptation du réseau est donnée comme :

$$\dot{\hat{W}} = F_w \varphi(z) e_g^T \quad (4.35)$$

Alors :

$$\dot{\rho} = -e_g^T Mh(q)(\alpha+1)e_g - e_g^T \beta Mh(q)e_g^{q_3/p_3} + e_g^T (\varepsilon - \tau_d + \xi) \quad (4.36)$$

Considérant le terme :

$$e_g^T (\varepsilon - \tau_d + \xi) = e_g^T (\varepsilon - \tau_d) + e_g^T \xi = e_g^T (\varepsilon - \tau_d) - \|e_g\| (\varepsilon_N + b_d) \leq 0$$

Sachant que la matrice Mh est définie positive, β_3 et α_3 sont des positives, donc $\dot{\rho} \leq 0$

IV.3.3. Résultats de simulations

Dans cette section, les simulations sont subdivisées en deux parties. Dans la première partie, le contrôleur dynamique GFSM avec un contrôleur cinématique basé sur la commande backstepping est proposé en l'absence et en la présence des perturbations. Dans la seconde partie, le NN-GFSM-backstepping est introduit pour traiter les perturbations. La simulation utilisant matlab / simulink est appliquée sur le robot mobile.

La trajectoire circulaire est considérée avec : $v_r = 1$ m/s, $\omega_r = 1$ rad/s. Les perturbations sont données avec : $\tau_d = [0.1 \cdot \sin(t) \ 0.1 \cdot \cos(t)]$. La trajectoire circulaire est considérée avec les valeurs suivantes : $\alpha_1=4, \beta_1=1, p_1=7, q_1=5, \alpha_2=0.5, \beta_2=2, p_2=7, q_2=5, \alpha_3=1, \beta_3=2, p_3=7, q_3=5, \varepsilon_N=0.2, b_d=0.1$.

Contrôleur dynamique GFSM :

En premier temps, on ne considère pas les perturbations.

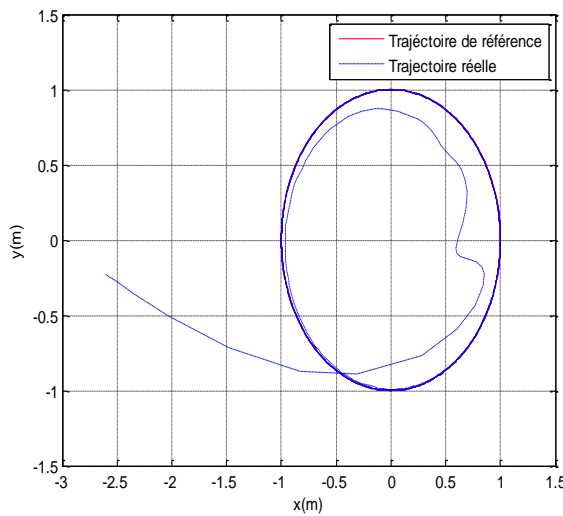


Figure IV.5: Suivi de trajectoire circulaire

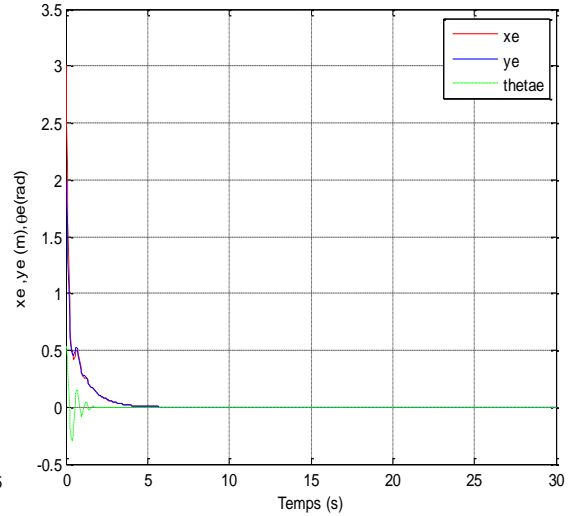


Figure IV.6: Erreurs x_e, y_e, θ_e

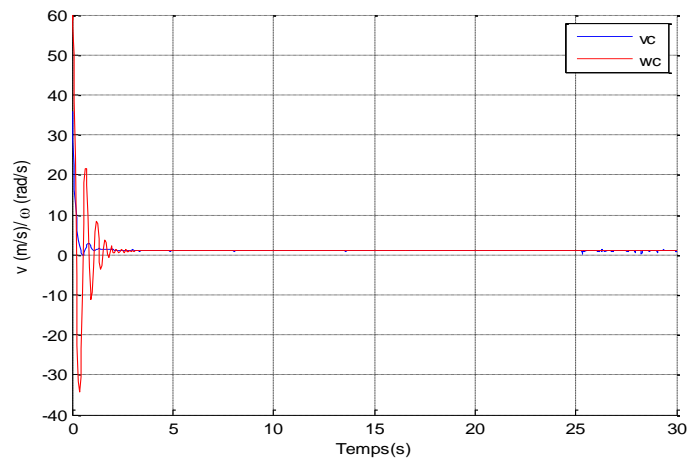


Figure IV.7: Vitesses dynamiques v et ω

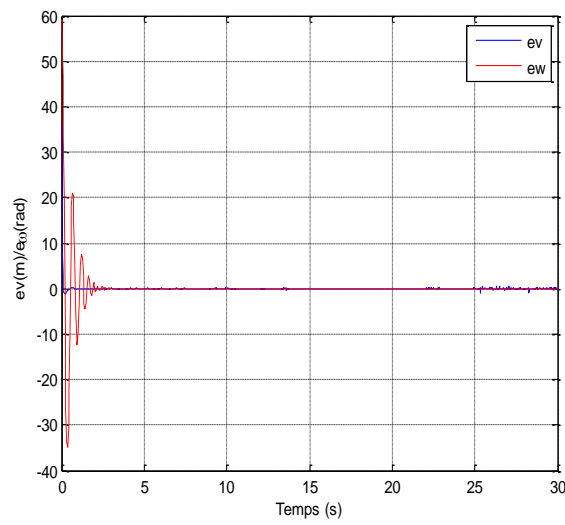


Figure IV.8: Erreurs de vitesses

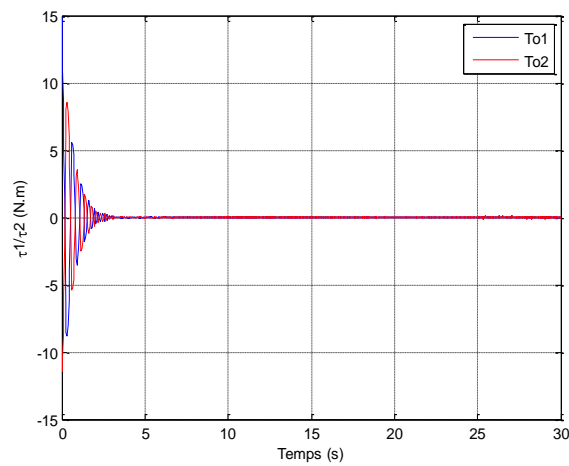


Figure IV.9: Couples générés

La figure (IV.5) montre que le robot mobile suit la trajectoire de référence après un temps considérable à cause des erreurs initiales. Dans la figure (IV.6), l'erreur θ_e converge vers zéro en un temps fini, mais les erreurs y_e et x_e convergent vers zéro dans un temps supérieur à 4s. D'après cette figure on remarque que x_e converge vers y_e dans un temps limité qui est le temps de convergence de la deuxième surface vers zéro qui traduit l'efficacité de la commande proposée pour atteindre le suivi en temps fini.

La figure (IV.7) présente les vitesses (linéaire et angulaire) du contrôleur dynamique v et w qui convergent vers les vitesses de référence qui valent 1. La figure (IV.8) présente les erreurs des vitesses linéaire et angulaire qui représentent les différences entre les vitesses fournies par le contrôleur cinématique et celles fournies par le contrôleur dynamique, et montre la convergence des erreurs vers la valeur zéro. La figure (IV.9) présente les couples appliqués sur les roues du robot.

En deuxième temps, on va injecter les perturbations.

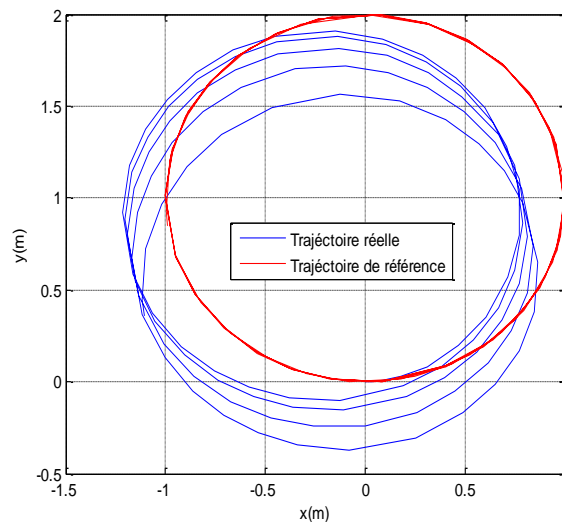


Figure IV.10: Suivi de trajectoires avec perturbations

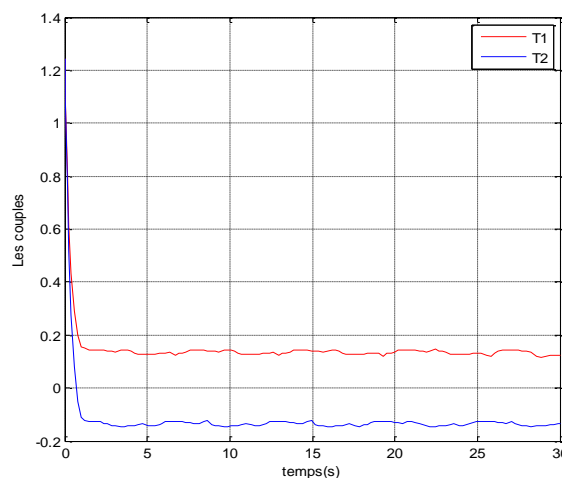


Figure IV.11: Couples τ_1 et τ_2

La figure (IV.10) présente le suivi d'une trajectoire circulaire en présence des perturbations et montre que le robot mobile diverge de la trajectoire de référence. Par conséquent, la commande GFSM stabilise le système en temps fini en absence de perturbations, mais ne peut pas le stabiliser en présence de perturbations. La figure (IV.11) montre les couples générés par les actionneurs.

Contrôleur dynamique NN- GFSM-backstepping

La trajectoire circulaire est considérée avec les mêmes valeurs des paramètres. Le réseau de neurones RBF est choisi avec sept couches cachées, la matrice de poids initiale est sélectionnée avec 0.1 et $b=10$.

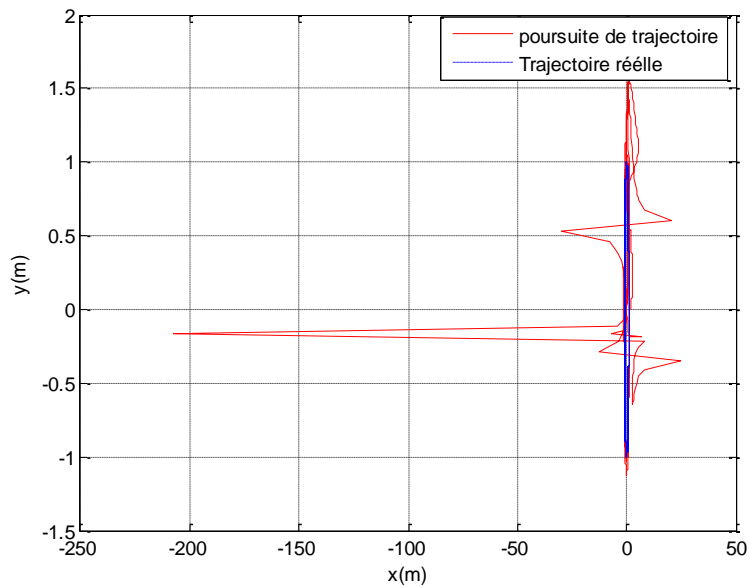


Figure IV.12: Suivi de trajectoire circulaire

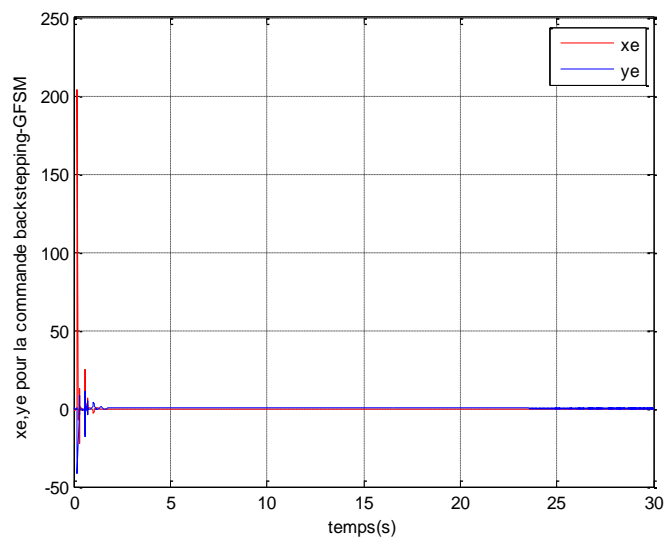


Figure IV.13: Erreurs de position x_e et y_e

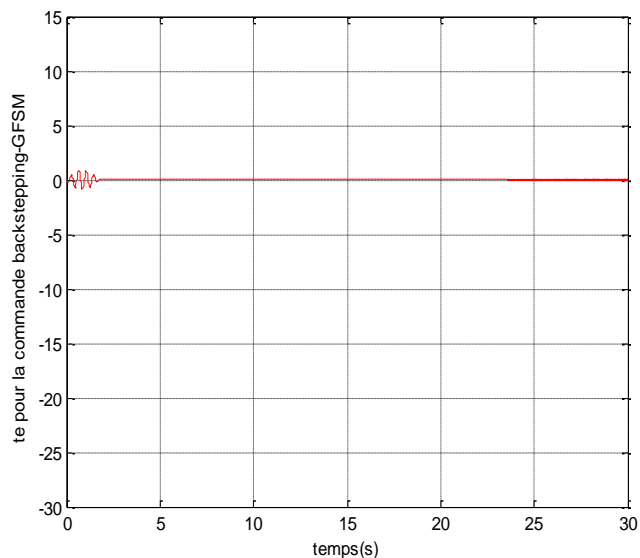


Figure IV.14: Erreur angulaire θ_e

La figure (IV.12) montre le suivi d'une trajectoire circulaire avec le contrôleur NN-GFSM-backstepping où le robot diverge de la trajectoire de référence dans un premier temps, mais il finit par suivre la référence. En comparant les résultats obtenus lors du suivi avec l'approche NN-FGSM-backstepping aux résultats précédents, il est clair que les performances du système ont été améliorées par rapport aux cas précédents.

Les figures (IV.13) et (IV.14) présentent les erreurs de position et l'erreur angulaire respectivement et montrent que ces erreurs convergentes vers zéro en un temps fini qui est inférieur à 2 secondes.

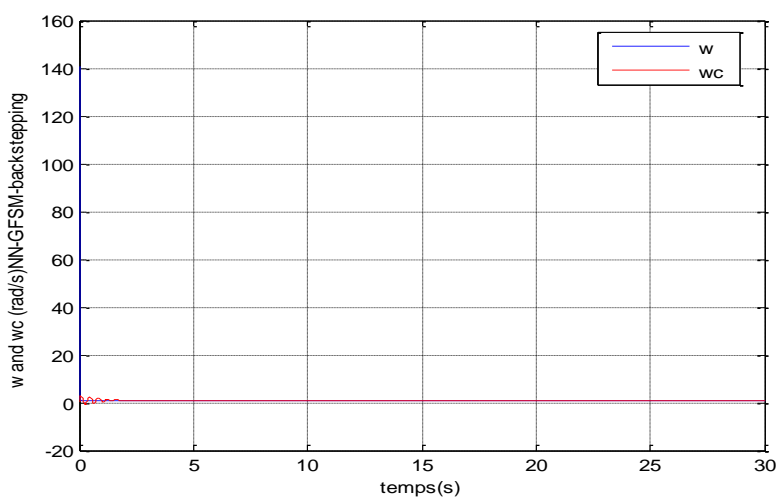


Figure IV.15: Vitesses angulaires ω et ω_c

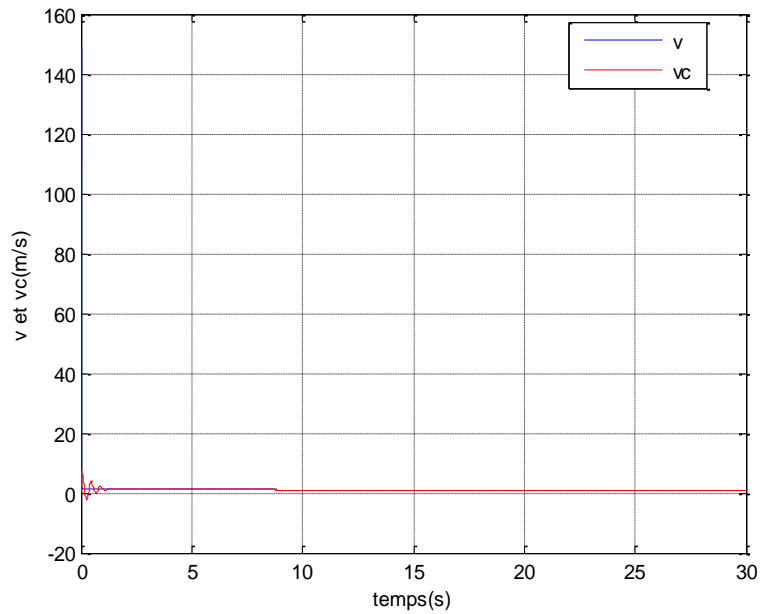


Figure IV.16: Vitesses linéaires v et v_c

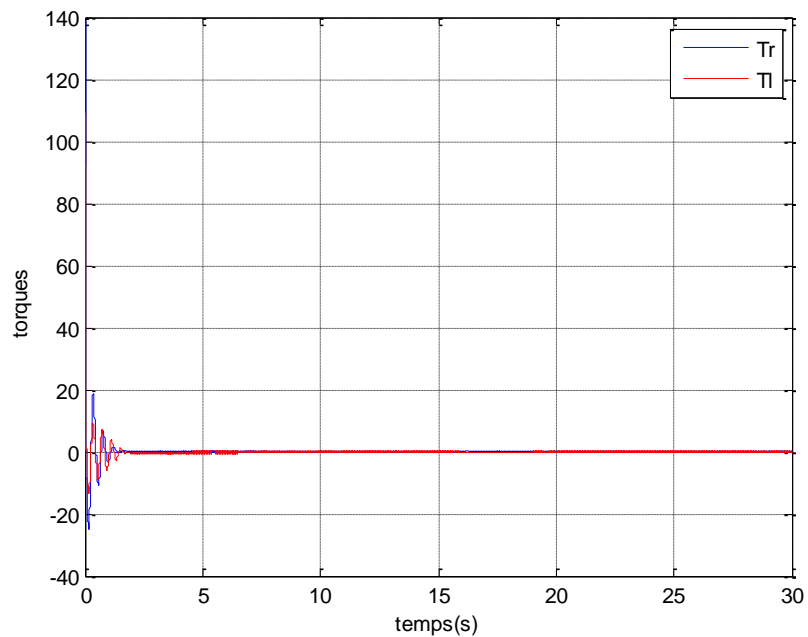


Figure IV.17: Couples générés

La figure (IV.15) montre que la vitesse linéaire v qui est la sortie du contrôleur dynamique converge vers la vitesse linéaire v_c qui est la sortie du contrôleur cinématique, alors l'erreur angulaire e_v converge vers zéro. La figure (IV.16) présente les vitesses angulaires où w converge vers w_c , alors e_w converge vers zéro.

La figure (IV.17) présente les couples appliqués aux roues du robot mobile

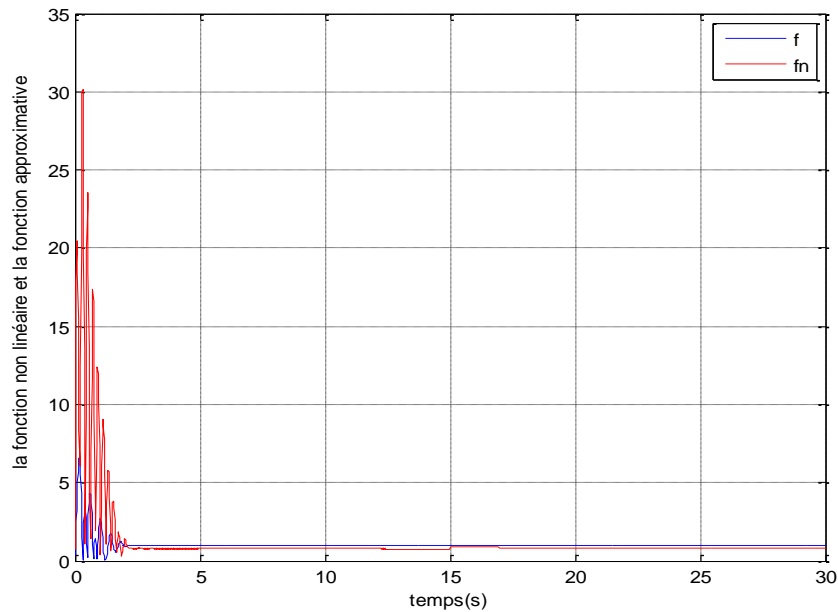


Figure IV.18: Fonctions f et \hat{f}

La figure (IV.18) présente la fonction non linéaire f et sa fonction estimée \hat{f} .

IV.4 Conclusion

Nous avons proposé dans ce chapitre une commande dynamique pour la poursuite de trajectoires d'un robot mobile qui est basée sur le mode glissant et les réseaux de neurones ainsi la commande backstepping pour la commande cinématique. Nous avons proposé en premier temps une commande dynamique basée sur l'approche GFSM-backstepping qui donne une bonne convergence vers la trajectoire de référence en un temps fini, mais cette approche a montré une robustesse limitée en présence des perturbations. En deuxième temps on a proposé une approche NN-GFSM qui associe trois contrôleurs, un contrôleur basé sur l'approche GFSM pour stabiliser les erreurs des vitesses et un contrôleur neuronal pour estimer la fonction non linéaire affectée par les perturbations et un contrôleur cinématique basé sur la commande backstepping. Le contrôleur neuronal a montré une amélioration des performances du système et une bonne robustesse par rapport à l'approche GFSM. Les travaux de simulation ont montré l'efficacité et la robustesse de la commande proposée.

CONCLUSION GENERALE

Nous avons présenté dans ce travail la modélisation des robots mobiles, où nous avons présenté les deux types de modèles sont généralement utilisés lors de la commande, à savoir : le modèle cinématique et le modèle dynamique.

Nous présentons les différents types de commande : la commande backstepping et la commande par mode glissant rapide, dont la commande par mode glissant rapide montre un suivi rapide et en temps fini vers la trajectoire de référence. La commande backstepping présente une bonne convergence mais avec un bon choix des gains. La commande dynamique exige deux contrôleurs : un cinématique et un dynamique, un contrôleur dynamique basé sur la commande par mode glissant rapide présente une robustesse limitée en présence des perturbations qui exige l'intervention des réseaux de neurones.

Ensuite, nous discutons brièvement des principales structures de réseaux de neurones qui sont des approximateurs universels, et les différents types d'architectures et de modèle, leurs méthodes d'apprentissages ainsi que leurs domaines d'utilisation.

En fin, nous présentons la conception d'un contrôleur neuronal qui sont inclus afin d'estimer la fonction non linéaire. Les simulations dans l'environnement MATLAB/SIMULINK prouvent l'efficacité des commandes proposées.

REFERENCES BIBLIOGRAPHIQUES

- [1] Nabil LAMARA, Réalisation d'une commande d'un robot mobile, Mémoire de Fin d'Etudes de MASTER PROFESSIONNEL , 2016.
- [2] Benmechiratima, CONTROLE D'UN SYSTEME FLOU DE TYPE 2, Mémoire de Fin d'Etudes de MASTER, 2011.
- [3] Celia Ait Ali, Commande d'un robot mobile par vision artificielle, Mémoire en vue de l'obtention du diplôme de master,2017.
- [4] V. I. Utkin, "V.S.S. with sliding modes", IEEE transactions on automatic control, volume AC-22, n°2, pp. 212-222, 1977.
- [5] K.K.D. Young and V.I. Utkin, "Methods for constructing discontinuity planes in multidimensional variable structure systems", Automatic Remote Control, volume 39,n°10, pp. 1466-1470, 1978.
- [6] J.J.E Slotine, "Sliding controller design for nonlinear systems", International Journal of Control, volume 40, n°2,pp. 421-434, 1984.
- [7] S.S. Sastry and J.J.E Slotine, "Tracking control of nonlinear systems using sliding surfaces, with application to robot manipulators", International Journal of Control, volume 38, n°2, pp.465- 492, 1983.
- [8] H. Asada and J.J.E. Slotine, "Robot analysis and control", John Wiley and sons, ISBN-10: 0471830291, April 25 1986.
- [9] J.Y. Hung, W.B. Gao and J.C. Hung, "Variable structure control: A survey", IEEE Transactions on Industrial Electronics, volume 40, n°1, pp. 2-21, 1993.
- [10] H. Habbi, M. Zelmat and B, OuldBouamama, "A dynamic fuzzy model for a drumboiler-turbine system", Automatica, volume 39, pp. 1213-1219, 2003.
- [11] H. Harashima, H. Hashimoto and K. Maruyama, "Practical robust control of robot armusing variable structure system", Proceedings of IEEE Conference, pp: 532-539.
- [12] A. Levant, "Principles of 2-Sliding mode design", Automatica, volume 43, pp. 576-586, 2007.

- [13] webographie.
- [14] Stéphane Lens « locomotion d'un robot mobile » mémoire de fin d'études, Université de Liège, Institut Montefiore, MAI 2008.
- [15] R. Dhaouadi and A. A. Hatab, "Dynamic modelling of differential-drive mobile robots using Lagrange and Newton-Euler methodologies: A unified framework," *Advances in Robotics & Automation*, vol. 2, pp. 1-7, 2013.
- [16] R. Dhaouadi and A. A. Hatab, "Dynamic modelling of differential-drive mobile robots using Lagrange and Newton-Euler methodologies: A unified framework," *Advances in Robotics & Automation*, vol. 2, pp. 1-7, 2013.
- [17] BENOIT BELOBO MEVO, Contribution à la commande adaptative et robuste d'un robot mobile de type unicycle avec modèle non-linéaire, Mai 2019.
- [18] Stéphane Lens, "Locomotion d'un robot mobile". Mémoire présenté en vue de l'obtention du grade d'Ingénieur Civil Informaticien Année académique 2007-2008, Université de Liège.
- [19] Belkhadria Khemisti, "commande d'un robot mobile par réseaux de neurones artificiels" Mémoire en vue de l'obtention du diplôme de magister en électronique, Option : Robotique. Université El Hadj Lakhdar Batna.
- [20] Mr. MAZIOUA Toufik, "Etude et réalisation d'un robot à base mobile de type unicycle" mémoire en vue de l'obtention du diplôme de Master en Automatique, Université A. MIRA-BEJAIA.
- [21] Goldstein H., Ch. Poole, and J. Safko, *Classical Mechanics*. Addison Wesley, 2002.
- [22] Bernard BAYLE, "Robotique mobile" Télécom Physique Strasbourg, université de Strasbourg.
- [23] Shabana A., *Dynamics of Multibody Systems*. Cambridge University Press, 1998.
- [24] Marion B. J., *Classical Dynamics of Particles and Systems*. Reverté, 2000.
- [25] ISELA BONILLA, FERNANDO REYES & MARCO MENDOZA, *Modelling and Simulation of a Wheeled Mobile Robot in Configuration Classical Tricycle*.
- [26] Campion G., G. Bastin, and B. D'Andrea-Novel, *Structural Properties and Classification of Kinematic and Dynamic Models of Wheeled Mobile Robots*. *IEEE Transactions on Robotics and Automation*, Vol. 12, No. 1, February 1996, pp. 47-62.
- [27] Campion G., B. d'Andrea-Novel, and G. Bastin, *Modelling and State Feedback Control of Nonholonomic Mechanical Systems*. *IEEE International Conference on Decision and Control*. Brighton, England. December 1991, pp. 1184-1189.
- [28] Spong M. W. and M. Vidyasagar, *Robots Dynamics and Control*. John Wiley & Sons, 1989.
- [29] Sciavicco L. and B. Siciliano, *Modeling and Control of Robot Manipulators*. McGraw-Hill, 1996.
- [30] CHEBILI Zakarya, "Contribution à la modélisation dynamique et la Commande des robots mobiles non holonomes à roues", mémoire en vue de l'obtention du DIPLOME DE MASTER En construction mécanique, 2018.
- [31] Nabil LAMARA, "Réalisation d'une commande d'un robot mobile", mémoire de Fin d'Etudes de

MASTER PROFESSIONNEL, 2016.

- [32] FufaevNewmark JI. Dynamics of nonholonomicsystems : Translations of mathematicalmonographs. American mathematical Society, 1972.
- [33] Baillieul J Marsden J Bloch A, Crouch P. Nonholonomic mechanics and control. IEEE Trans. Indust. Applic.Vol.40, N° .1, 2003.
- [34] SmainAzzeddine, “Stabilisation et Poursuite de Trajectoire de Robots Mobiles de Type Voiture“, THESEPrésentée pour l’obtention du grade de Docteur en sciences, 2017.
- [35] BenaoumeurIbari, “Backstepping Approach for Autonomous Mobile Robot TrajectoryTracking“, 2016.
- [36] Backstepping-prof-
- [37] Michael Defoort, “Contributions à la planification et à la commande pour les robots mobiles coopératifs“, 2007.
- [38] Boujidj Rania, “Modélisation et élaboration d'une commande adaptative neuronale - mode glissant d'un bras manipulateur“, 2020.
- [39] R.A. Decarlo, S.H. Zak and G.P. Matthews, “Variable Structure Control of nonlinear multivariable systems: A tutorial”, IEEE, volume 76, n°3, pp. 212-232, 1988.
- [40] AbdelmalikBachtarzi. “ Commande des systèmes a structure variable applications à un générateur de vapeur”. Thèse de doctorat. Université de Constantine, Algérie.3 juillet 2011.
- [41] BENAZIZA Walid, “Commande par modes glissants du suivi de trajectoires pour un robot mobile“ THESE Présentée pour l’obtention du diplôme de MAGISTER en Électronique.
- [42] F.Yang, M.Paindavoine, 2005.
- [43] A.Seghouane, G.Fleury, 2003.
- [44] B. Krose P. van der Smagt, «An introduction to neuronal networks », Eighth edition,1996.
- [45] Les réseaux de neurone RBF,Rfia2012.
- [46] Introduction au connexionnisme, Claude Touzet Juillet, 1992.
- [47] C, Touzet, « Les réseaux de neurones artificiels introduction au conexionisme », 1992.
- [48] A. Balouli et S. Bouteraa, ‘contrôle en ligne d’un CSTR par réseaux de neurones’,mémoire de fin d’études d’ingénieur d’état en électronique option contrôle, centre universitaire Oum El Bouaghi promotion juillet 2004.
- [49] Z. P. Jiang, E. Lefeber, and H. Nijmeijer, “Saturated stabilization and tracking of a nonholonomic mobile robot,” Systems & Control Letters, vol. 42, pp. 327-332, 2001.
- [50] G. Ramirez et S. Zeghloul, « A New Local Path Planner for Nonholonmic Mobile Robot Navigation in Cluttered Environments », dans Proc. IEEE Int.Conf. on Robotics and Automation, 2000, p. 2058–2063.
- [51] Salah Eddine Ghamri, «Commande Neuronale d’un Robot Mobile Non Holonome avec glissement des roues », Mémoire Présenté pour l’obtention du diplôme de MAGISTER en Électronique, Centre

universitaire Batna2.

- [52] H. AbdElaali « utilisation d'une structure variable et du mode glissant dans un réseau RBF contrôleur », projet d'ingénieur d'état en électronique, Centre universitaire Oum El Bouaghi, promotion juillet 2006.
- [53] R. Fierro, and F. L. Lewis, "Control of a nonholonomic mobile robot using neural networks", IEEE Transactions on Neural Networks, vol. 9, no. 4, pp. 389-400, 1998.
- [54] S. X. Yang, and M. Meng, "Real-time fine motion control of robot manipulators with unknown dynamics," in: Dynamics in Continuous, Discrete and Impulse Systems, Series B, 2001.
- [55] OMID MOHARERI, mémoire de master, 'Mobile robot tracking using neural networks', 2009.
- [56] Z. P. Jiang, E. Lefeber, and H. Nijmeijer, "Saturated stabilization and tracking of a nonholonomic mobile robot," Systems & Control Letters, vol. 42, pp. 327-332, 2001.
- [57] G. Ramirez et S. Zeghloul, « A New Local Path Planner for Nonholonomic Mobile Robot Navigation in Cluttered Environments », dans Proc. IEEE Int. Conf. on Robotics and Automation, 2000, p. 2058–2063.
- [58] R. Fierro, and F. L. Lewis, "Control of a nonholonomic mobile robot using neural networks", IEEE Transactions on Neural Networks, vol. 9, no. 4, pp. 389-400, 1998.