

République Algérienne Démocratique et Populaire  
Ministère de L'Enseignement Supérieur et de la Recherche Scientifique



Université Mohamed Seddik Ben-Yahia, jijel

Faculté des sciences et de la Technologie

Département d'Automatique

Projet de fin d'étude

En vue de l'obtention du diplôme

**MASTER**

Filière : Génie électrique

Option : Automatique et traitement de signal

Thème

# Commandes des robots mobiles

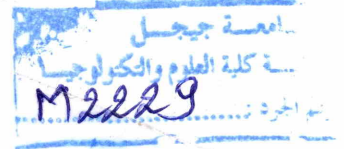
Présenté par :

**Amina Boubellouta**

Proposé et dirigé par :

**Dr. A. Boulkrone**

Promotion 2014/2015



# *Remerciements*

Au moment où on achève ce travail, je remercie Allah tout puissant qui m'a donné la force et la volonté pour finir ce travail.

J'adresse mes grands remerciements au Mr. A.boulekroune pour avoir accepté de m'encadrer, pour ses encouragements et sa patience .je remercie les membres de jury.

Mes plus vifs remerciements à toute ma famille, tous mes amis Je tiens aussi remercier, tous ceux qui m'ont enseigné durant toutes mes années D'études.

*Dédicace*

*je dédie ce mémoire*

*A mes parents,*

*A ma famille*

*et à tous ceux que j'aime.*

# SOMMAIRE

---

# SOMMAIRE

**Introduction générale ..... 1**

## **Chapitre 1**

### **Généralités sur les robots mobiles**

1.1 Introduction.....4  
1.2 Historique .....5  
1.3 Boucle de contrôle.....6  
1.4 Classification des robots.....7  
1.5 Différentes bases mobiles.....7  
    1.5.1 Holonomie.....7  
    1.5.2 Plates-formes différentielles.....8  
    1.5.3 Plates-formes omnidirectionnelles.....8  
    1.5.4 Plates-formes non holonomes.....8  
    1.5.5 Plates-formes à pattes.....8  
    1.5.6 Comparaison des différents types.....9  
1.6 Applications des robots mobiles.....9  
1.7 Conclusion .....10

## **Chapitre 2**

### **Modélisation des robots mobiles**

2.1 Introduction.....11  
2.2 Modélisation d'un robot mobile bicycle .....11  
2.3 Conclusion .....16

## Chapitre 3

### Commande par bouclage linéarisant

|   |    |
|---|----|
| 3.1 Introduction.....   | 17 |
| 3.2 Linéarisation entrée-sortie .....   | 17 |
| 3.3 Commande par bouclage linéarisant appliquée au robot mobile bicycle ..... | 19 |
| 3.4 Simulation.....   | 21 |
| 3.5 Conclusion .....  | 26 |

## Chapitre 4

### Commande par mode glissant des systèmes non linéaires

|   |    |
|---|----|
| 4.1 Introduction .....  | 27 |
| 4.2 Commande par mode glissant.....   | 27 |
| 4.2.1 Fondements théoriques de la commande par mode glissant .....                    | 27 |
| 4.2.2 Conception de la commande à mode glissant pour un système<br>monovariablen..... | 28 |
| 4.2.3 Condition de convergence et d'existence   |    |
| 4.2.3.1 Fonction directe de commutation.....  | 29 |
| 4.2.3.2 Fonction Lyapunov.....  | 29 |
| 4.2.4 Loi de commande .....   | 30 |
| 4.3 Commande à mode glissant pour les systèmes multivariables.....                    | 33 |
| 4.4 Simulation .....  | 37 |
| 4.5 Conclusion .....  | 43 |

## Chapitre 5

### Commande adaptative floue des systèmes non linéaires

|   |           |
|---|-----------|
| 5.1 Introduction.....                                     | 44        |
| 5.2 Systèmes Flous.....                                   | 44        |
| 5.2.1 Logique floue .....                                 | 44        |
| 5.2.2 Notions de Base de la logique floue.....            | 47        |
| 5.2.2.1 Ensembles flous.....                              | 47        |
| 5.2.2.1.1 Ensembles classiques et flous .....             | 47        |
| 5.2.2.1.2 Formalisme Linguistique.....                    | 51        |
| 5.2.2.1.3 Opérations de base sur les ensembles flous..... | 52        |
| 5.2.2.1.4 Interprétation des ensembles flous.....         | 54        |
| 5.2.2.2 Règles floues.....                                | 54        |
| 5.2.2.2.1 Calcul des propositions floues .....            | 56        |
| 5.2.2.2.2 Calcul des relations.....                       | 56        |
| 5.2.2.2.3 Inférence floue .....                           | 57        |
| 5.2.2.2.4 Agrégation des règles.....                      | 58        |
| 5.2.2.2.5 Défuzzification.....                            | 59        |
| 5.2.3 Différents types de modèle.....                     | 59        |
| 5.2.3.1 Modèles de Mamdani (linguistiques).....           | 59        |
| 5.2.3.2 Modèles de TS.....                                | 61        |
| 5.2.4 Théorème des approximateurs universels.....         | 64        |
| 5.3 Commande floue adaptative .....                       | 65        |
| 5.4 Simulation .....                                      | 72        |
| 5.5 Conclusion .....                                      | 77        |
| <br>  |           |
| <b>Conclusion générale.....</b>                           | <b>78</b> |
| <br>  |           |
| <b>Bibliographie .....</b>                                | <b>80</b> |

# Introduction Générale



## Introduction générale

La *robotique mobile* est la discipline qui s'intéresse à la conception de robots mobiles. Elle s'appuie sur d'autres disciplines comme l'automatique, le traitement du signal, la mécanique, l'informatique et l'électronique. La commande des robots mobiles a fait l'objet de nombreuses recherches ces dernières années, en raison de l'utilisation de plus en plus fréquente en milieu dangereux ou inaccessible où l'être humain peut difficilement intervenir. À titre d'exemples, nous pouvons citer l'intervention sur des sites nucléaires dangereux ou l'exploration spatiale.

L'objectif de ce mémoire est de donner un aperçu sur les méthodes de modélisation et de commande des robots mobiles à deux roues. Les robots à deux roues seront modélisés par des *équations d'état*, c'est-à-dire des équations différentielles non linéaires du premier ordre. Ces équations d'état peuvent être obtenues en utilisant les lois de la mécanique. Il n'est pas dans notre objectif l'étude en détail des méthodes de modélisation des robots, mais juste d'en rappeler les principes. Par *modélisation*, nous entendons ici l'obtention des équations d'état. Cette étape est indispensable pour la simulation des robots ainsi que pour la conception de différentes commandes.

Nous allons toutefois illustrer le principe de la modélisation au chapitre 2 sur un robot bicycle. Ce choix a pour objectif d'introduire des concepts importants pour la robotique mobile comme les équations de Lagrange. Les modèles utilisés dans ce contexte sont non linéaires et la propriété de non-holonomie qui caractérise un certain nombre des robots mobiles conduit à l'utilisation de techniques de commande non linéaire pour les piloter.

La commande par linéarisation exacte (connue également sous le nom de « feedback-linearization ») au sens entrée-état ou bien au sens entrée-sortie[25]. Cette commande consiste à linéariser le système par compensation et à appliquer au nouveau système linéarisé une commande linéaire classique, par exemple une commande par retour d'état. Cependant, pour pouvoir appliquer cette méthode, le système doit vérifier des critères mathématiques précis. Cependant, aussi intéressante qu'elle puisse paraître, la commande par linéarisation exacte, n'est pas une commande robuste, et elle

est, de ce fait, sensible aux variations paramétriques. Il est toutefois possible de la stabiliser de manière robuste, en ajoutant un processus adaptatif au contrôleur non linéaire [26] ou bien en augmentant cette commande par un terme robuste à mode glissant .

La commande par mode glissant a largement prouvé son efficacité à travers plusieurs études théoriques, ses principaux domaines d'application sont la robotique et la commande des moteurs électriques [27]. L'avantage que procure une telle commande qui la rend aussi importante, est sa robustesse vis-à-vis des perturbations et des incertitudes du modèle, la stabilité, la simplicité et le temps de réponse très faible. Cependant, ces performances sont obtenues au prix de certains inconvénients telles que:

1) l'apparition du phénomène de chattering ou broutement provoqué par la partie discontinue de la commande qui peut avoir un effet néfaste sur les actionneurs, le système est soumis à chaque instant à une commande élevée afin d'assurer sa convergence vers l'état désiré et ceci n'est pas souhaitable.

2) Dues aux commutations rapides de la commande discontinue, on peut exciter les parties non modélisées de la dynamique haute fréquence du système. Parmi les solutions proposées à ces problèmes, on peut citer:

1) l'utilisation de la commande par mode glissant à bande limite qui consiste à remplacer la fonction de commutation par une fonction de saturation.

2) l'emploi de la commande floue à la place de la commande par mode glissant, pour diminuer l'effet du chattering.

Le troisième type de commande non linéaire considérée dans ce mémoire est la commande adaptative par logique floue. La synthèse de cette commande ne nécessite pas de modélisation préalable, mais seulement des informations émanant de l'expert humain. L'application des systèmes flous dans la commande a été introduite depuis les années 70 par Mamdani et son équipe. Cette commande non linéaire est connue pour sa robustesse vis à vis les incertitudes de modélisation. Comme dans la commande adaptative classique, on peut distinguer deux cas : direct et indirect. Dans le cas de la loi directe, la commande est approximée par un approximateur mis à jour selon une loi d'adaptation déduite de l'étude de la stabilité. Dans le cas indirect, on approxime d'abord les fonctions non linéaires du modèle puis on met en œuvre la loi de commande. Les lois d'adaptation sont également déduites de l'étude de stabilité au sens de Lyapunov.

Ce mémoire est organisé en cinq chapitres comme suit:

*Le premier chapitre* est consacré à un rappel sur la robotique mobile. Nous classons les robots mobiles selon quatre classes principales: robot tricycle, voiture, omnidirectionnel et le robot uni cycle.

*Dans le deuxième chapitre*, on fait la modélisation du robot bicycle en utilisant le formalisme d'Euler-Lagrange. Le modèle dynamique établi sera par la suite exploité dans la conception des différentes commandes et la simulation numérique.

*Dans le troisième chapitre*, nous allons concevoir une commande non-linéaire basée sur la linéarisation exacte entrée-sortie. Cette commande sera ensuite testée sur ce robot mobile en présence et en absence des incertitudes.

*Le quatrième chapitre* est consacré à l'étude et à la synthèse de la commande par mode glissant pour résoudre le problème des systèmes multivariables non linéaires. Cette commande sera validée sur notre robot en considérant les deux cas avec et sans incertitudes.

*Dans le cinquième chapitre*, nous étudions la technique de commande adaptative floue et les notions générales relatives à la logique floue ainsi que aux systèmes flous. Cette commande floue adaptative qui ne dépend pas du modèle sera conçue en utilisant l'approche de Lyapunov et testée sur notre robot mobile bicycle.

A la fin de ce mémoire, nous donnons une générale conclusion pour résumer et rappeler l'essentiel de ce travail et donner ses perspectives envisagées.

# Chapitre 1

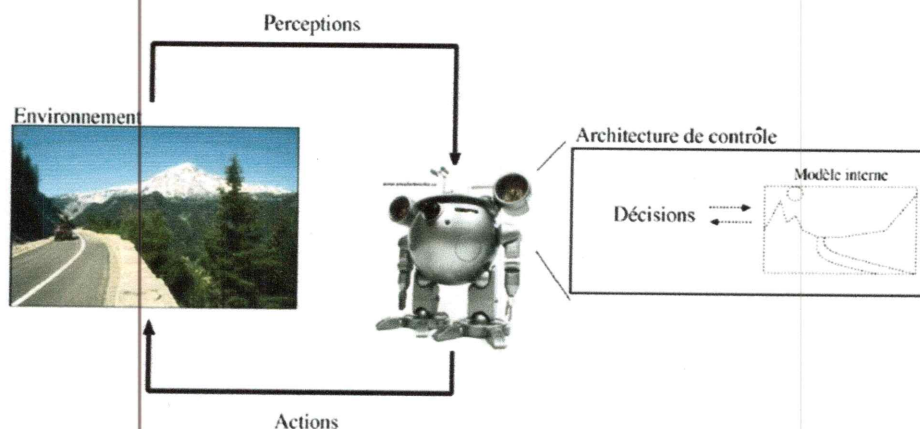
# Chapitre 1

## Généralités sur les robots mobiles

### 1.1 Introduction

La robotique est un très bon exemple de domaine pluridisciplinaire qui implique de nombreuses thématiques telles que la mécanique, la mécatronique, l'électronique, l'automatique, l'informatique ou l'intelligence artificielle. En fonction du domaine d'origine des auteurs, il existe donc diverses définitions du terme robot, mais elles tournent en général autour de celle-ci [1] :

**Un robot** est une machine équipée de capacités de perception, de décision et d'action qui lui permettent d'agir de manière autonome dans son environnement en fonction de la perception qu'il en a.



**Figure 1.1** : Schéma des interactions d'un robot avec son environnement

Selon les approches, un modèle interne de l'environnement peut être utilisé ou non. Cette définition s'illustre par un schéma classique des interactions d'un robot avec son environnement (Figure 1.1).

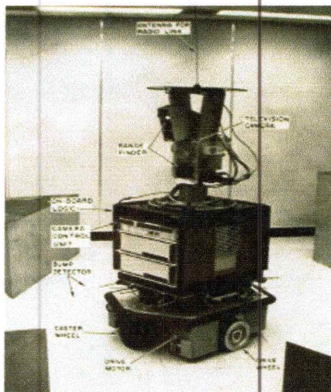
**Origine des termes :** 'Robot ' a été utilisé pour la première fois en 1921 par Karel Capek dans sa pièce R.U.R. : Rossums Universel Robots.

- Il provient du tchèque "robota" qui signifie corvées, travail obligatoire.
- Le terme robotique a été employé pour la première fois par Asimov en 1941.

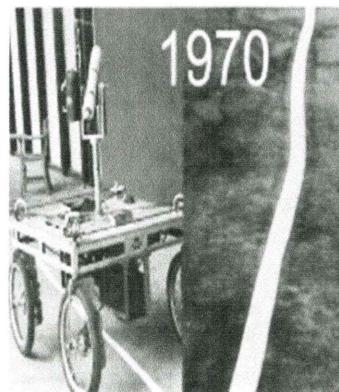
## 1.2 Historique :

Le mot **robot** qui apparaît pour la première fois dans la pièce de théâtre de Karel Capek [2] :

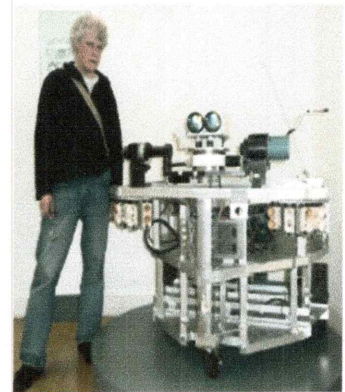
- **1921** : Karel Capek , RUR (Rossum's Universal Robots).
- **1940** : Isaac Asimov écrit un ensemble de nouvelles sur les robots.
- **1960-1980** : premiers robots mobiles :
- **1960-1964** : Ouverture des laboratoires d'Intelligence Artificielle au M.I.T., Stanford Research Institute (SRI), Stanford University, University of Edinburgh.
- Fin des années 60 : Mise en place de "Shakey" premier robot mobile intégrant perception, planification et exécution.
- **1970** : Stanford Cart
- **1977** : premier robot mobile français HILARE au LAAS (CNRSToulouse).



(a)



(b)



(c)

**Figure 1.2** : premier robot mobile français : (a)hilare . (b) shakey . (c) stanford cart

- **1990- 2000** : essor de la robotique mobile
- **1992** : Mise en place de la compétition annuelle AAAI sur la robotique mobile.

- 1997 : premier robot mobile extra planétaire sur Mars.
- 1999 : Lancement d'Aibo.
- Depuis 2000 : exploration .
- 2003 : Projet "Mars Exploration Rover" (Spirit & Opportunity).
- 2009 : projet " Mars Science Laboratory" succédant au projet Rover, envoi prévu de Curiosity fin 2011

### 1.3 Boucle de contrôle

Un robot mobile est commandé par une boucle de contrôle, comme illustré à la figure (1.2). De façon itérative, cette boucle fait une lecture des données reçues par les capteurs, les interprètes, calcule les commandes motrices et les envoie aux actionneurs. Typiquement, cette boucle est exécutée environ dix (10) fois par seconde ; la fréquence peut varier selon les types de capteurs et d'actionneurs utilisés. La boucle de contrôle n'est pas unique ; selon l'architecture utilisée, elle peut être décomposée en plusieurs sous-boucles de contrôle agencées de manières différentes [3].

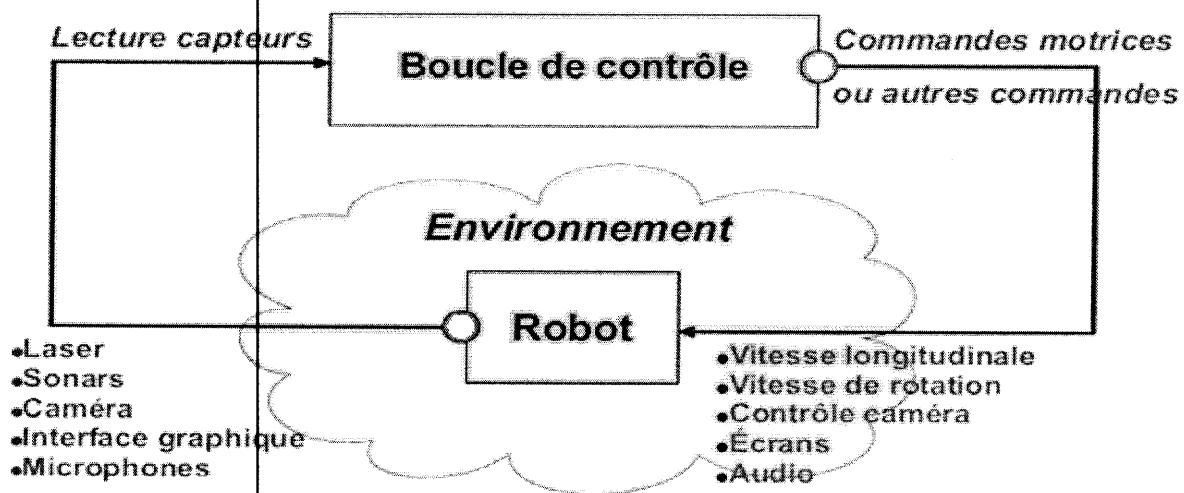


Figure 1.3 : Boucle de contrôle.

## 1.4 Classification des robots

Les systèmes robotiques peuvent être classifiés selon cinq catégories, comme précisé dans :

- **robots marcheurs** désignent l'ensemble des robots équipés de pattes. On peut notamment noter les robots humanoïdes qui sont des robots marcheurs bipèdes.
- **robots manipulateurs** sont généralement des bras mécaniques fixés au sol ou encore sur des rails. Ils sont notamment utilisés en industrie afin d'effectuer rapidement des tâches répétitives et de façon précise.
- **robots aériens**, aussi appelés drones ou **UAV1** désignent l'ensemble des robots volants sans pilote. On peut par exemple noter les avions militaires photographiant des zones dangereuses.
- **robots sous-marins**, ou **UUV2** désignent l'ensemble des robots sous-marins évoluant sans pilote. Ils sont par exemple utilisés pour la détection de mines sous-marines.
- **robots terrestres**, ou **UGV3** correspondent à l'ensemble des véhicules terrestres se déplaçant sans pilote, à l'exception des robots marcheurs. Les robots aspirateurs sont des exemples d'UGV.

Il est important de noter que le terme "robots mobiles" bien que désignant l'ensemble des robots à bases mobiles, est généralement employé pour désigner les robots mobiles terrestres.

## 1.5 Différentes bases mobiles

Les différents bases mobiles sont [1]:

### 1.5.1 Holonomie

En robotique, une plateforme est dite holonome lorsque que le nombre de degrés de libertés contrôlables est égal au nombre total de degrés de liberté. Pour un robot se déplaçant sur un plan, il y a 3 degrés de liberté. A partir d'une position donnée, une plateforme holonome devra donc pouvoir se déplacer en avant, sur le côté et tourner sur elle-même. Cette capacité permet de contrôler très simplement le robot car tous les déplacements imaginables sont réalisables, ce qui simplifie le problème de planification de trajectoire.



### 1.5.2 Plates-formes différentielles

Ce type de plateforme est caractérisée par:

- Deux roues commandées indépendamment.
- Une ou plusieurs roues folles sont ajoutées à l'avant ou à l'arrière du robot pour assurer sa Stabilité.
- Plate-forme simple à commander : spécifier la vitesse des roues motrices permettant au robot de tourner sur place.

Ce type de plate-forme peut également être utilisé avec des chenilles ce qui fournit une capacité de franchissement de petits obstacles.

### 1.5.3 Plates-formes omnidirectionnelles

Les plates-formes omnidirectionnelles permettent de découpler de manière plus nette le contrôle de la rotation et de la translation d'un robot et sont donc quasiment holonomes. Il existe différents types de plateformes omnidirectionnelles. Le premier utilise trois ou quatre roues qui tournent à la même vitesse pour fournir une translation et un mécanisme qui permet d'orienter simultanément ces roues dans la direction du déplacement souhaitée. Une deuxième catégorie de plateformes utilise des roues dites "suédoises", qui n'offrent pas de résistance au déplacement latéral. La plateforme comporte trois roues dont les axes sont fixes.

### 1.5.4 Plates-formes non holonomes

Des plates-formes non holonomes, telles que les voitures, sont également utilisées en robotique mobile. C'est plus particulièrement le cas dans le domaine des véhicules intelligents. Ces plates-formes sont toutefois plus difficiles à commander car elles ne peuvent pas tourner sur place et doivent manœuvrer, ce qui peut être difficile dans des environnements encombrés. La commande de ces plates-formes pour réaliser un déplacement particulier est un problème à part entière que nous n'aborderons pas dans ce mémoire. Par contre, il est possible de prendre en compte ces contraintes de manière relativement simple dans la planification.

### 1.5.5 Plates-formes à pattes

Des plates-formes à deux, quatre ou six pattes peuvent également être utilisées. Elles ont l'avantage théorique de pouvoir se déplacer sur des terrains assez complexes, même si en pratique la plupart de ces plates-formes ne fonctionnent que sur des sols plans.

### 1.5.6 Comparaison des différents types

Nous pouvons observer dans le tableau ci-dessous un récapitulatif des avantages et des inconvénients des différents types de robots à roues [4] [5] :

**Tableau 1.01 : Comparaison des différents types**

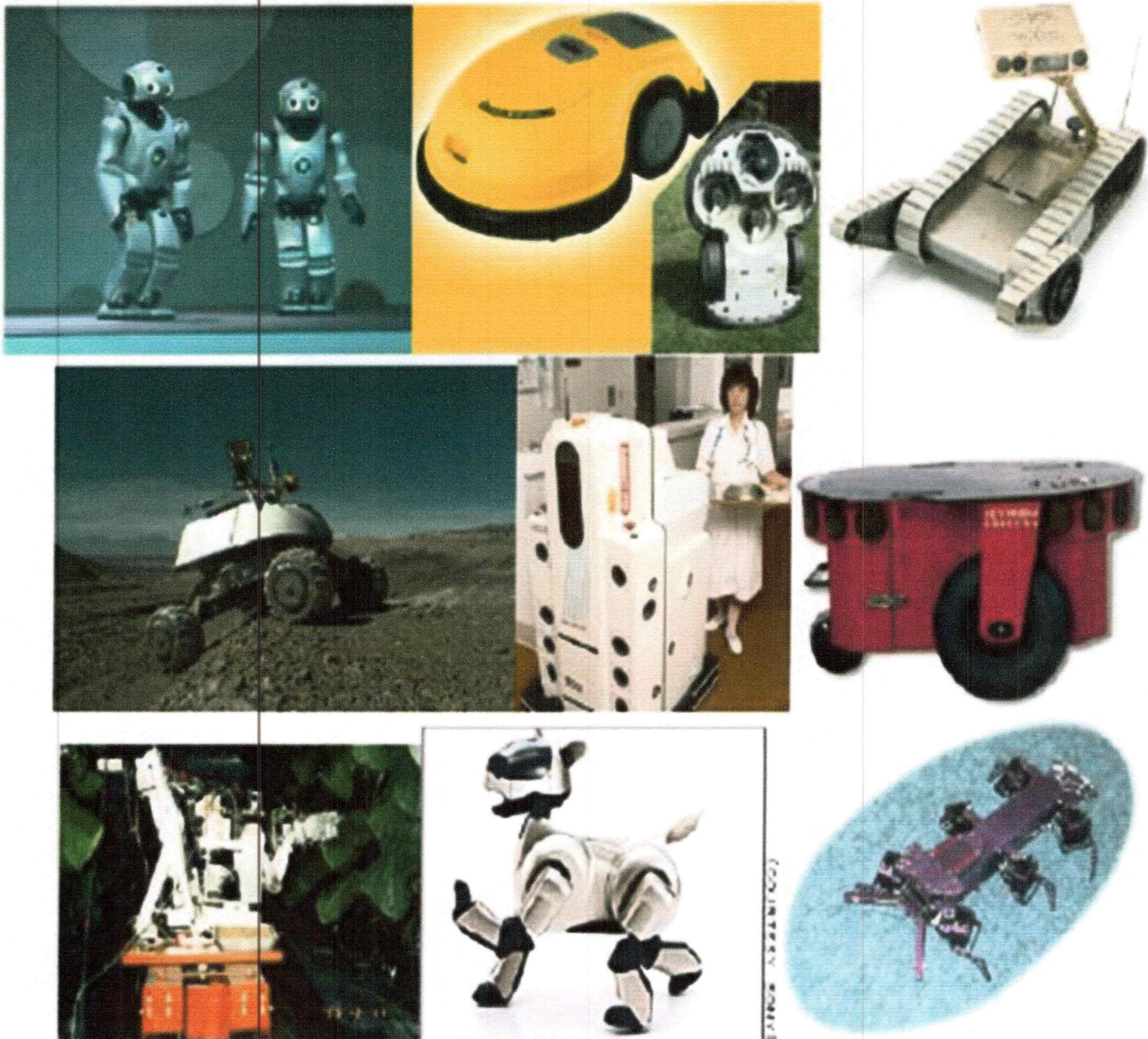
| Robot                  | Avantages  | Inconvénients  |
|------------------------|--|--|
| Robot uni cycle        | Stable<br>Rotation sur soi-même<br>Complexité mécanique faible | Non-holonome   |
| Robot tricycle         | Complexité mécanique modérée                                   | Non-holonome<br>Peu stable<br>Pas de rotation sur soi-même |
| Robot voiture          | Stable<br>Complexité mécanique modérée                         | Non-holonome<br>Pas de rotation sur soi-même               |
| Robot omnidirectionnel | Holonome<br>Stable<br>Rotation sur soi-même                    | Complexité mécanique importante                            |

### 1.6 Applications des robots mobiles

Les applications des robots peuvent se trouver dans de nombreuses activités "ennuyeuses, salissantes ou dangereuses", mais également pour des applications ludiques ou de service, comme l'assistance aux personnes âgées ou handicapées. Parmi les domaines concernés, citons :

- La robotique de service (hôpital, bureaux)
- La robotique de loisir (robot 'compagnon')
- La robotique industrielle ou agricole (entrepôts, récolte de productions agricoles, mines)

- La robotique industrielle ou agricole (entrepôts, récolte de productions agricoles, mines)
- La robotique en environnement dangereux (spatial, industriel, militaire)



**Figure 1.4:** Exemples de robots utilisés dans différentes applications

## 1.7 Conclusion

Dans ce chapitre, nous avons introduit quelques définitions et généralités sur les robots mobiles ainsi que leur principe de fonctionnement. Nous avons essayé de les classer en quatre grandes catégories, et nous avons fait une comparaison entre les différents types de plates-formes robotiques.

# Chapitre 2

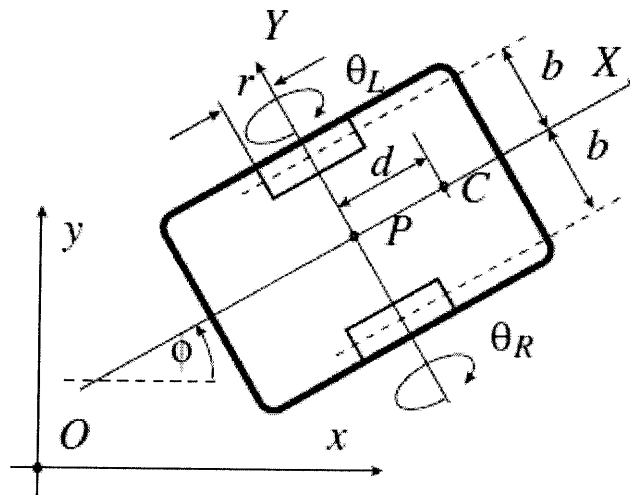
## *Chapitre 2*

### *Modélisation des robots mobiles*

#### 2.1 Introduction

Ce chapitre est consacré à la modélisation des robots mobiles à roues. Notre objective est de rechercher un modèle qui décrit avec exactitude le comportement d'un robot mobile à roues. Ces systèmes robotiques sont en effet un assemblage complexe d'une multitude d'éléments mécaniques, mais aussi électriques. Nous allons modéliser le robot bicycle.

#### 2.2 Modélisation d'un robot mobile bicycle



**Figure 2.1 :** Robot mobile à deux roues.

Considérons un robot mobile à deux roues se déplaçant dans un plan comme représenté sur la Figure (2.1). Le robot se caractérise par ses coordonnées  $(x, y, \phi)$ , qui sont mesurées conformément à un référentiel local ayant son origine au centre géométrique  $P$  de la base mobile. Les angles de rotation

des roues sont perpendiculaires au plan du mouvement, ayant les valeurs respectives  $(\theta_L, \theta_R)$ . Chaque roue est actionnée indépendamment et tourne sans glissement. Soit  $m$  la masse totale du robot,  $m_c$  la masse de la plate forme (sans roues), et  $m_w$  la masse de chaque roue, de sorte que  $m = m_c + 2m_w$ . Soit  $J$  le moment d'inertie total par rapport à l'axe vertical passant par  $P$ , et  $J_w$  le moment d'inertie de chaque roue par rapport à son axe de rotation. Alors,  $J = J_c + m_c d^2 + 2m_w b^2$  où  $J_c$  est le moment d'inertie sans roues par rapport au centre de masse  $C$  [6]. Les coordonnées et la vitesse du point  $C$  sont :

$$P_c = P + d(\cos \phi, \sin \phi), \quad V = V_c + \dot{\phi}d(-\sin \phi, \cos \phi) \quad (2.1)$$

Le Lagrangien est :

$$L = \frac{1}{2} m \langle V_c, V_c \rangle + \frac{1}{2} J \dot{\phi}^2 + \frac{1}{2} J_w (\dot{\theta}_L^2 + \dot{\theta}_R^2) \quad (2.2)$$

où  $\langle \cdot, \cdot \rangle$  indique le produit scalaire. Il existe trois contraintes. La première est que le robot mobile ne peut pas se déplacer dans la direction latérale [6], c'est-à-dire

$$\dot{y} \cos \phi - \dot{x} \sin \phi = 0 \quad (2.3)$$

Les deux autres contraintes sont que les deux roues motrices roulent sans glissement :

$$\begin{aligned} \dot{x} \cos \phi + \dot{y} \sin \phi + b \dot{\phi} &= r \dot{\theta}_L \\ \dot{x} \cos \phi + \dot{y} \sin \phi - b \dot{\phi} &= r \dot{\theta}_R \end{aligned} \quad (2.4)$$

A partir des équations (2.3) et (2.4), nous pouvons obtenir deux contraintes non-holonomes et une contrainte holonome. Pour obtenir la contrainte holonome, on soustrait la première équation de (2.4) du deuxième de celle-ci, on obtient  $\dot{\phi} = r / (2b)(\dot{\theta}_L - \dot{\theta}_R)$ . En intégrant cette dernière et choisissant correctement l'état initial de  $(\phi, \theta_L, \theta_R)$ , nous obtenons  $\phi = c(\theta_L - \theta_R)$  avec  $c = r / (2b)$ ,

qui représente clairement une équation de contrainte holonome [6]. Les deux contraintes non holonomes sont :

$$\begin{aligned} \dot{y} \cos \phi - \dot{x} \sin \phi &= 0 \\ \dot{x} \cos \phi + \dot{y} \sin \phi &= cb(\dot{\theta}_L + \dot{\theta}_R) \end{aligned} \quad (2.5)$$

En notant les variables de la configuration par  $q = (x, y, \theta_L, \theta_R)^T$ , les contraintes (2.5) peuvent être présentées sous la forme suivante:

$$A(q)\dot{q} = \begin{pmatrix} -\sin \phi & \cos \phi & 0 & 0 \\ \cos \phi & \sin \phi & -cb & -cb \end{pmatrix} \dot{q} = 0 \quad (2.6)$$

Les équations du mouvement sont obtenues à partir des équations de Lagrangien comme suit :

$$\frac{d}{dt} \left( \frac{\partial L}{\partial \dot{q}} \right) - \frac{\partial L}{\partial q} + A(q)^T \lambda = E(q)\tau \quad (2.7)$$

où  $\lambda = [\lambda_1, \lambda_2]^T$  sont appelés les multiplicateurs de Lagrange associés aux contraintes cinématique

(2.6),  $\tau = [\tau_R, \tau_L]^T$  sont les couples d'entraînement, et  $E(q) = \begin{bmatrix} 0 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 \end{bmatrix}^T$ . Ainsi, les équations du

mouvement sont :

$$M(q)\ddot{q} + V(q, \dot{q}) + A(q)^T \lambda = E(q)\tau \quad (2.8)$$

où

$$M(q) = \begin{pmatrix} m & 0 & -mcd \sin \phi & mcd \sin \phi \\ 0 & m & mcd \cos \phi & -mcd \cos \phi \\ -mcd \sin \phi & mcd \cos \phi & J_1 & J_2 \\ mcd \sin \phi & -mcd \cos \phi & J_2 & J_1 \end{pmatrix} \quad (2.9)$$

avec

$$J_1 = J_w + c^2(J + md^2), J_2 = -c^2(J + md^2), \text{ et}$$

$$V(q, \dot{q}) = \begin{pmatrix} -c^2 dm \dot{\phi}^2 \cos \phi \\ -c^2 dm \dot{\phi}^2 \sin \phi \\ 0 \\ 0 \end{pmatrix} \quad (2.10)$$

Définissons, maintenant :

$$N_A(q) = \begin{pmatrix} cb \cos \phi & cb \cos \phi \\ cb \sin \phi & cb \sin \phi \\ 1 & 0 \\ 0 & 1 \end{pmatrix} \quad (2.11)$$

De telle sorte que  $A(q)N_A(q) = 0$ . La pré-multiplication de l'équation (2.8) par  $N_A(q)^T$  donne :

$$N_A(q)^T M(q) \ddot{q} + N_A(q)^T V(q, \dot{q}) = N_A(q)^T E(q) \tau \quad (2.12)$$

Etant donné que  $A(q)N_A(q) = 0$ , il existe un vecteur  $\eta = [\eta_1, \eta_2]^T$  tel que  $\dot{q} = N_A(q)\eta$ . A partir de  $N_A(q)$ , on peut facilement vérifier que  $\eta = [\dot{\theta}_L, \dot{\theta}_R]^T$ . Ensuite, le calcul de la dérivée temporelle de  $\dot{q} = N_A(q)\eta$  donne [6]:



$$\ddot{q} = \dot{N}_A(q)\dot{\eta} + N_A(q)\ddot{\eta} \quad (2.13)$$

Considérant maintenant une partition adéquate telle que  $q = (P, \Theta)$  avec  $P = (x, y)$  et  $\Theta = (\theta_L, \theta_R)$ , on peut écrire (2.13) comme :

$$\ddot{q} = \dot{N}_A(q)\dot{\Theta} + N_A(q)\ddot{\Theta} \quad (2.14)$$

Le remplacement de (2.14) dans (2.12) donne:

$$\ddot{\Theta} = f_\theta(q, \dot{q}) + g_\theta(q, \dot{q})\tau \quad (2.15)$$

où

$$\begin{aligned} f_\theta(q, \dot{q}) &= -(N_A^T M N_A)^{-1} (N_A^T M \dot{N}_A \dot{\Theta} + N_A^T V) \\ g_\theta(q, \dot{q}) &= (N_A^T M N_A)^{-1} N_A(q)^T E(q) \end{aligned} \quad (2.16)$$

Et puisque  $q = (P, \Theta)$ , on obtient également:

$$\ddot{P} = f_P(q, \dot{q}) + g_P(q, \dot{q})\tau \quad (2.17)$$

L'ensemble des équations (2.15) et (2.17) représente les équations dynamiques du mouvement. En définissant  $Y = P_c = P + d(\cos \phi, \sin \phi) = h(q)$  comme fonction de sortie, les équations du système deviennent [6]:

$$\begin{aligned} \ddot{P} &= f_P(q, \dot{q}) + g_P(q, \dot{q})\tau \\ \ddot{\Theta} &= f_\theta(q, \dot{q}) + g_\theta(q, \dot{q})\tau \\ Y &= h(q) \end{aligned} \quad (2.18)$$

avec

$$f_p(q, \dot{q}) = \begin{pmatrix} c_1 \dot{\phi}^2 \cos \phi - c_2 (\dot{\theta}_L^2 - \dot{\theta}_R^2) \sin \phi \\ c_1 \dot{\phi}^2 \sin \phi + c_2 (\dot{\theta}_L^2 - \dot{\theta}_R^2) \cos \phi \end{pmatrix}, \quad g_p(q, \dot{q}) = c_0 \begin{pmatrix} \cos \phi & \cos \phi \\ \sin \phi & \sin \phi \end{pmatrix}, \quad f_\theta(q, \dot{q}) = c_3 \dot{\phi} \begin{pmatrix} c_4 \dot{\theta}_L - c_5 \dot{\theta}_R \\ c_3 \dot{\theta}_L - c_4 \dot{\theta}_R \end{pmatrix}$$

$$g_\theta(q, \dot{q}) = \frac{1}{2} \begin{pmatrix} c_6 & c_7 \\ c_7 & c_6 \end{pmatrix}, \quad c_0 = \frac{bc}{J_w + 2b^2c^2m}, \quad c_1 = \frac{2b^2c^2md}{J_w + 2b^2c^2m}, \quad c_2 = bc^2,$$

$$c_3 = \frac{2bc^2dm}{(J_w + 2b^2c^2m)(J_w + 2c^2(J + md^2))}, \quad c_4 = c^2(J + m(d^2 - b^2)), \quad c_5 = J_w + c^2(J + m(d^2 + b^2))$$

$$c_6 = \frac{1}{(J_w - 2b^2c^2m)} + \frac{1}{(J_w + 2c^2(J + md^2))}, \quad c_7 = \frac{1}{(J_w - 2b^2c^2m)} - \frac{1}{(J_w + 2c^2(J + md^2))}$$

### 2.3 Conclusion

Dans ce chapitre, nous avons établi le modèle dynamique du robot mobile bicycle, en utilisant le formalisme de Lagrange-d'Alembert. Ces modèles seront employés pour la synthèse des lois de commande dans le chapitre suivant.

# Chapitre 3

## ***Chapitre 3***

### ***Commande par bouclage linéarisant***

#### **3.1 Introduction**

Les robots sont considérés comme des systèmes fortement non linéaires. Dans ce chapitre, nous allons concevoir des commandes non linéaires afin de forcer le vecteur d'état du robot à suivre une trajectoire lisse fixée à l'avance. Il n'existe en effet pas de méthode générale et standard pour stabiliser globalement les systèmes non linéaires [7]. Le but de ce chapitre est d'en présenter une assez représentative des méthodes théoriques existantes, il s'agit de la méthode de linéarisation par bouclage ou par bouclage linéarisant (BL). Le système choisi ici est le robot bicycle dont le modèle est décrit dans le chapitre 2.

#### **3.2 Linéarisation entrée-sortie**

Un système non linéaire est un système qui ne satisfait pas le principe de superposition. Les problèmes non linéaires sont d'intérêt pour les automaticiens, physiciens et mathématiciens, car la plupart des systèmes physiques sont intrinsèquement non linéaires dans la nature.

Les outils et techniques de conception pour la commande des systèmes non linéaires peuvent être classés en deux grandes catégories. Dans la première catégorie, La conception de la loi de commande passe essentiellement par deux étapes, dans une première étape il s'agit de transformer, par l'introduction d'une commande dite linéarisante, le système non linéaire en un système linéaire à travers des algorithmes de linéarisation spécifiques, la deuxième étape repose sur l'utilisation des techniques de commande linéaire pour le système linéarisé. Cette technique est connue sous le nom de feedback linéarisation.

La deuxième catégorie de commande utilise des algorithmes et des techniques plus adaptées aux systèmes non linéaires, il s'agit de concevoir des contrôleurs non linéaires sans passer par une

linéarisation. Différentes approches existent, telles que l'approche adaptative, l'approche floue, l'approche neuronale, l'approche backtepping, etc.

**Principe de la méthode de commande par bouclage linéarisant:** Nous cherchons à concevoir la commande par BL décrite au paragraphe précédent. Considérons le système non linéaire suivant [8]:

$$\begin{cases} \dot{x} = f(x) + g(x)u \\ y = h(x) \end{cases} \quad (3.1)$$

Où  $x$  est le vecteur d'état,  $u$  le vecteur d'entrée et  $y$  le vecteur des mesures,  $f(x)$  et  $g(x)$  fonction non linéaire continue inconnue. dont le nombre d'entrées et de sorties sont tous les deux égaux à  $m$ . L'idée de la méthode par bouclage linéarisant est de boucler le système par une commande non linéaire de type  $u = r(x, v)$ , où  $v$  est la nouvelle entrée, aussi de dimension  $m$ . Cette opération nécessite que l'état soit complètement accessible.

Pour effectuer ce bouclage, il nous faut exprimer les dérivées successives de chacun des  $y_i$  en fonction de l'état et de l'entrée. On s'arrête de dériver  $y_i$ , dès que les entrées commencent à apparaître dans l'expression de la dérivée. Nous disposons ainsi d'une équation de type:

$$\begin{pmatrix} y_1^{(k_1)} \\ \vdots \\ y_m^{(k_m)} \end{pmatrix} = G(x)u + F(x) \quad (3.2)$$

où  $k_i$  désigne le degré relatif de chaque sous-système.

Sous l'hypothèse que la matrice  $G(x)$  soit inversible, la commande par bouclage linéarisant est donnée par:

$$u = G^{-1}(x)(v - F(x)) \quad (3.3)$$

où  $v$  est notre nouvelle entrée. Après l'application de (3.3) au système (3.2), on obtient le système linéaire  $S_L$  de  $m$  entrées à  $m$  sorties décrit par les équations différentielles suivantes :

$$S_L : \begin{cases} y_1^{(k_1)} = v_1 \\ \vdots \\ y_m^{(k_m)} = v_m \end{cases} \quad (3.4)$$

Ce système est linéaire et complètement découplé (c'est-à-dire que chaque entrée  $v_i$  agit sur une et une seule sortie  $y_i$ ). Il est donc très facile de concevoir  $v_i$  par les techniques linéaires classiques. Ici, comme le système à commander est constitué de chaînes d'intégrateurs découplés, nous utiliserons  $m$  régulateurs de types PID (Proportionnel-Intégral-Dérivée). Notons que pour utiliser de tels régulateurs, il est nécessaire de disposer des dérivées de sorties. Comme nous avons supposé avoir accès à toutes les variables l'état  $x_i$  du système, une expression formelle de ces dérivées en fonction des  $x_i$  s'obtient facilement en utilisant les équations d'états.

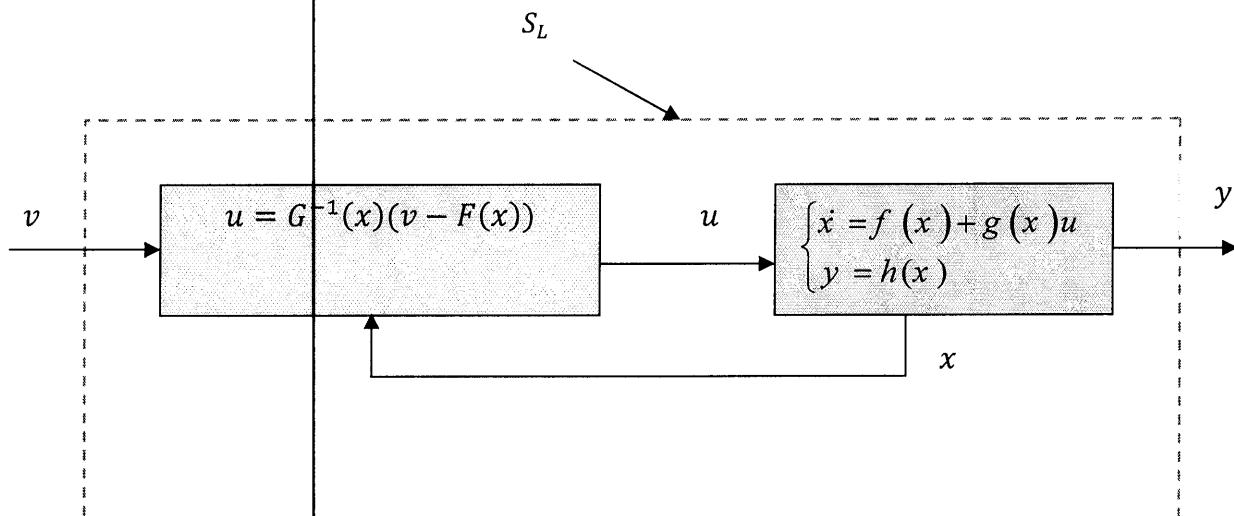


Figure 3.1 : Principe de la commande linéarisante par bouclage.

### 3.3 Commande par bouclage linéarisant appliquée au robot mobile bicycle

Dans cette section, nous concevons une loi de commande linéarisante par bouclage pour le robot mobile bicycle en nous basant sur son modèle dynamique donnée dans le chapitre 2. Pour vérifier si le système (2.18) est E/S linéarisable, avec  $Y$  comme sortie, nous calculons les dérivées de  $Y$  jusqu'à ce que les entrées  $\tau$  apparaissent dans les équations. En adoptant le formalisme de Lie [9], nous obtenons :

$$\ddot{Y} = F(q, \dot{q}) + G(q, \dot{q})\tau \quad (3.5)$$

Les expressions détaillées de  $F(q, \dot{q})$  et  $G(q, \dot{q})$  sont données comme:

$$F(q, \dot{q}) = \begin{pmatrix} -c_8 \dot{\phi} \cos \phi - c_9 (\dot{\theta}_L^2 - \dot{\theta}_R^2) \sin \phi \\ -c_8 \dot{\phi} \sin \phi + c_9 (\dot{\theta}_L^2 - \dot{\theta}_R^2) \cos \phi \end{pmatrix} \text{ Et } G(q, \dot{q}) = \begin{pmatrix} c_{10} \cos \phi - c_{11} \sin \phi & c_{10} \cos \phi + c_{11} \sin \phi \\ c_{10} \cos \phi + c_{11} \sin \phi & -c_{10} \cos \phi + c_{11} \sin \phi \end{pmatrix} \quad (3.6)$$

Où

$$c_8 = \frac{dJ_w}{(J_w + 2b^2 c^2 m)}, \quad c_9 = \frac{c^2 b (J_w + 2c^2 J)}{J_w + 2c^2 (J + d^2 m)}, \quad c_{10} = \frac{bc}{J_w + 2b^2 c^2 m} \quad \text{et} \quad c_{11} = \frac{cd}{J_w + 2c^2 (J + d^2 m)}.$$

Si  $G(q, \dot{q})$  est inversible, et si nous choisissons la commande comme  $\tau = G^{-1}(q, \dot{q})(v - F(q, \dot{q}))$ , nous pouvons réduire la dynamique non linéaire (3.5) à un double intégrateur en cascade:

$$\ddot{Y} = v \quad (3.7)$$

Considérons une trajectoire de référence  $Y_r$ , donc l'erreur de poursuite est  $e = Y_r - Y$ . En appliquant la commande  $\tau = G^{-1}(q, \dot{q})[v - F(q, \dot{q})]$  au système, on obtient:

$$\ddot{e} = \ddot{Y}_r - \ddot{Y} = \ddot{Y}_r - v \quad (3.8)$$

Si l'on choisit

$$v = \ddot{y}_r + K_p e + K_d \dot{e} + K_i \int_0^t e d\tau \quad (3.9)$$

où  $K_p$  et  $K_d$  et  $K_i$  sont des matrices de conception diagonales et définies positives. Elles sont aussi choisies telle que la dynamique suivante est stable.

$$\ddot{e} + K_p e + K_d \dot{e} + K_i \int_0^t e d\tau = 0 \quad (3.10)$$

### 3.4 Simulation

Pour les illustrer les performances de la méthode du commande par bouclage linéarisant, nous faisons des simulations numériques sur ce robot à deux roues. Pour la résolution numérique des équations différentielles, nous avons adopté la méthode de RK4. La trajectoire de références est choisie comme suit:

$$\begin{aligned}x_r &= \sin(\alpha t) \\ y_r &= \rho \cos(\beta t)\end{aligned}\tag{3.11}$$

avec  $\alpha = 2/3$ ,  $\beta = 1/3$  et  $\rho=2$ . Dans le plan  $x$ - $y$ , ce chemin de référence prend la forme d'un *huit*. Les paramètres du robots sont choisis comme :  $m_c = 94 \text{ kg}$ ,  $m_w = 5 \text{ kg}$ ,  $J_c = 6.61 \text{ kgm}^2$ ,  $J_w = 0.01 \text{ kgm}^2$ ,  $r = 0.075 \text{ m}$ ,  $b = 0.171 \text{ m}$ , et  $d = 0.08 \text{ m}$ .

Pour tester la robustesse de cette commande, nous avons considéré deux cas: i.e. avec perturbation  $\Delta = 0.15[c_4\dot{\theta}_l - c_5\dot{\theta}_r; c_5\dot{\theta}_l - c_4\dot{\theta}_r]^T$  et sans perturbation ( $\Delta = [0 \ 0]^T$ ). Les paramètres de conception sont choisis comme suit :

$$k_i = 2, k_d = 100, k_p = 500$$

Les figures (3.2)-(3.5) montrent les résultats de simulation obtenus en appliquant la commande par bouclage linéarisant, pour le cas sans perturbations. Ces résultats montrent que les performances de poursuite sont très bonnes et tous les signaux du système en boucle fermée sont bornés et les erreurs de poursuite convergent vers zéro.

Les résultats de simulation pour le cas avec perturbation sont représentés sur les Figures (3.6)-(3.9). Nous remarquons d'après ces figures que les performances en terme de poursuite sont dégradées. Donc, cette commande n'est pas robuste vis-à-vis les incertitudes paramétriques et les perturbations ajoutées.



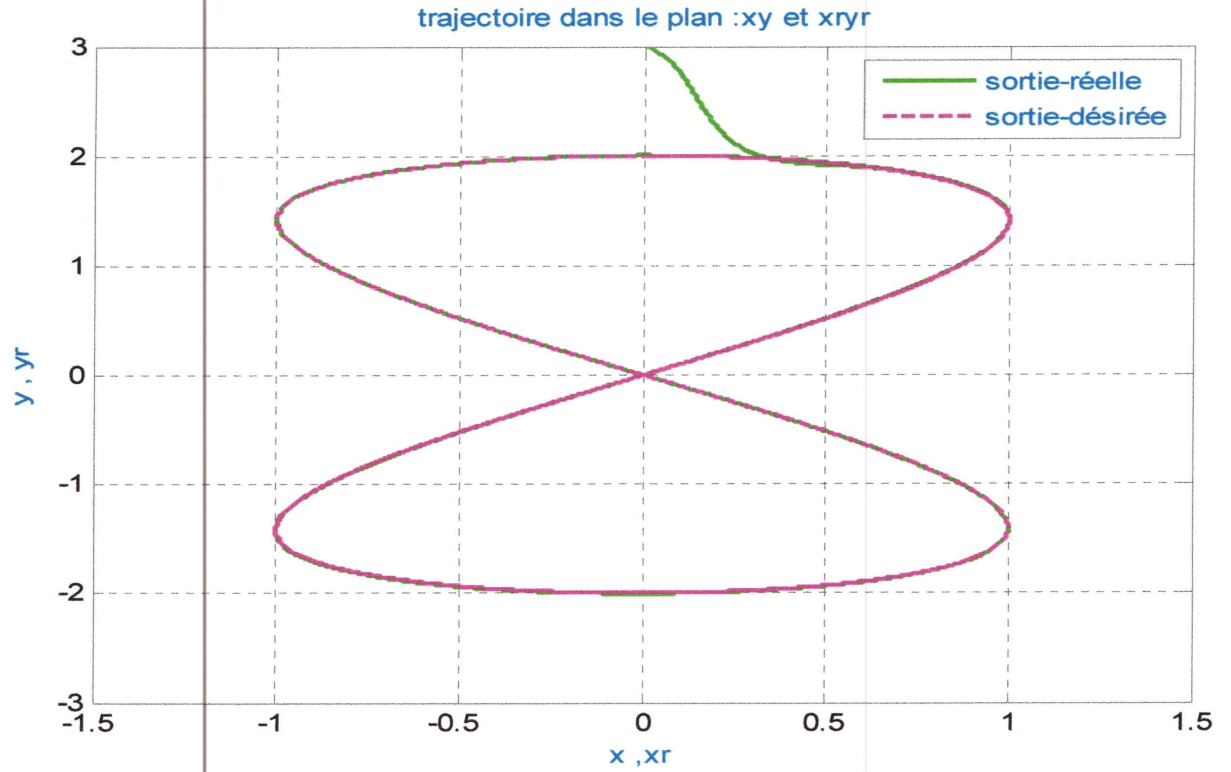


Figure 3.2: Trajectoire tracée par le robot mobile (pour le cas sans perturbations).

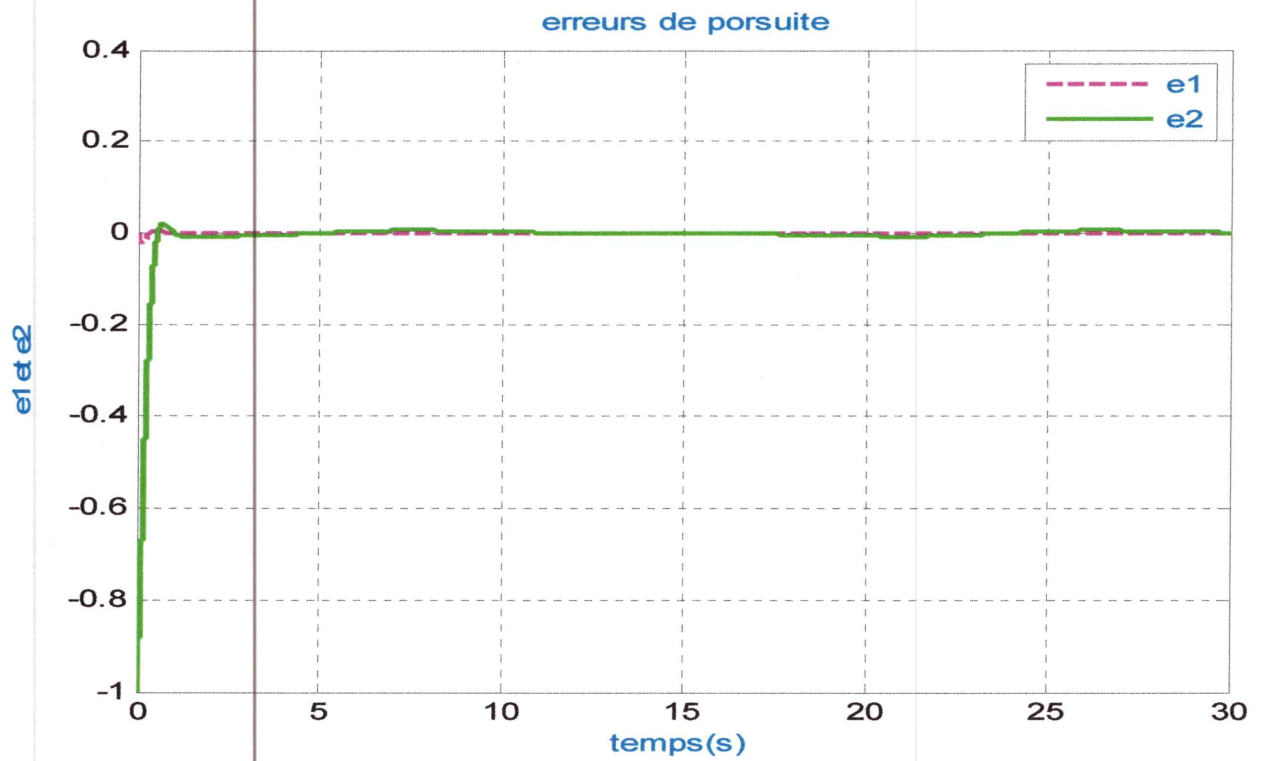


Figure 3.3: Signaux d'erreurs de poursuite (pour le cas sans perturbations).

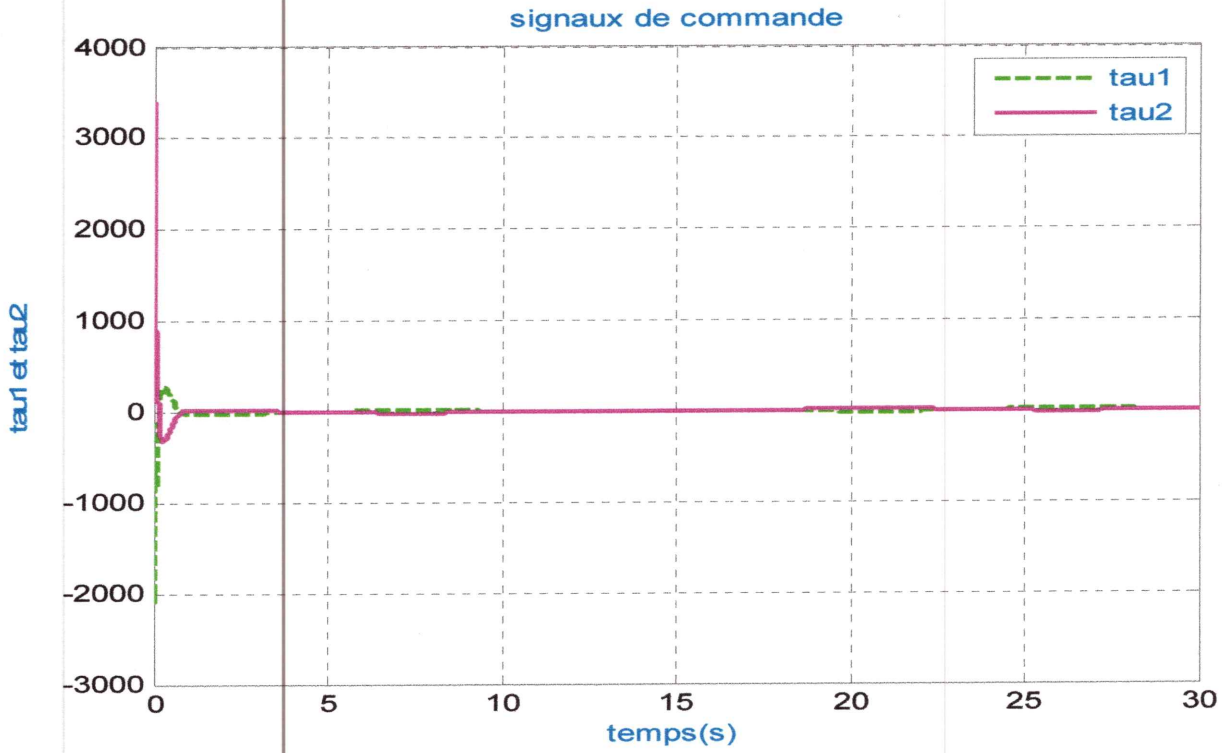


Figure 3.4: Signaux de commande (pour le cas sans perturbations).

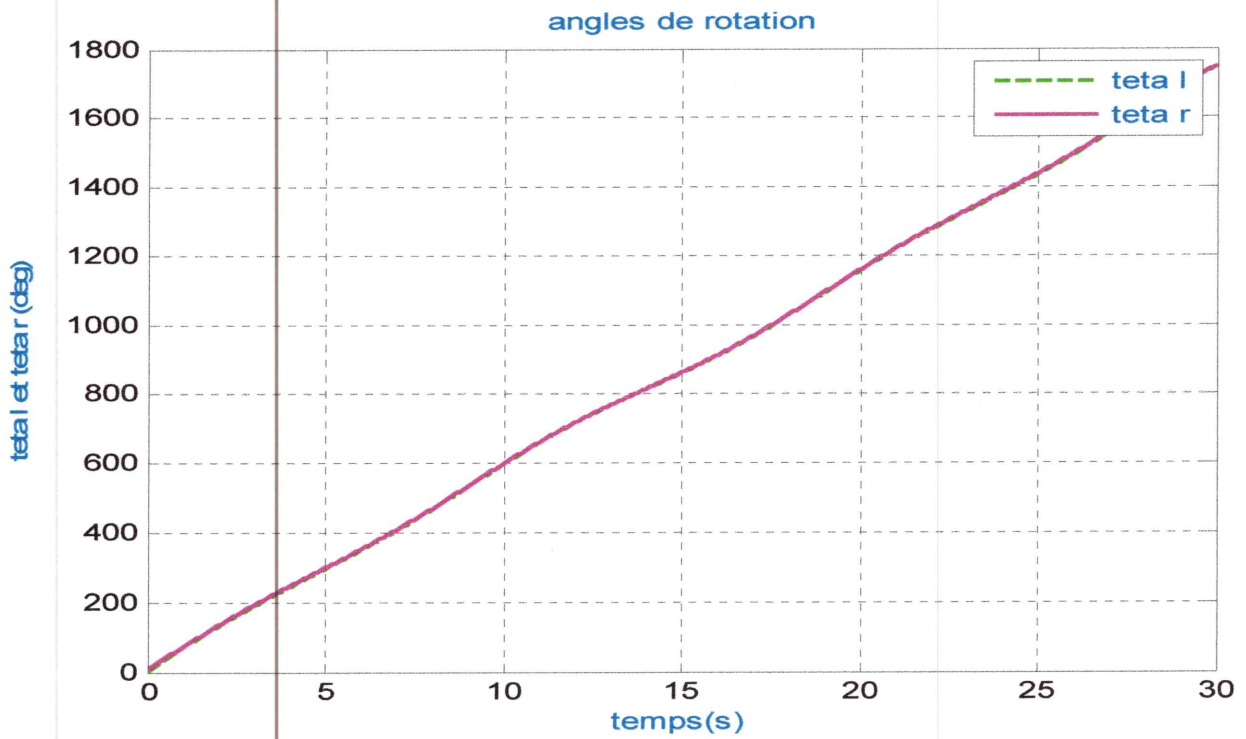


Figure 3.5: Angles de rotation (pour le cas sans perturbations).

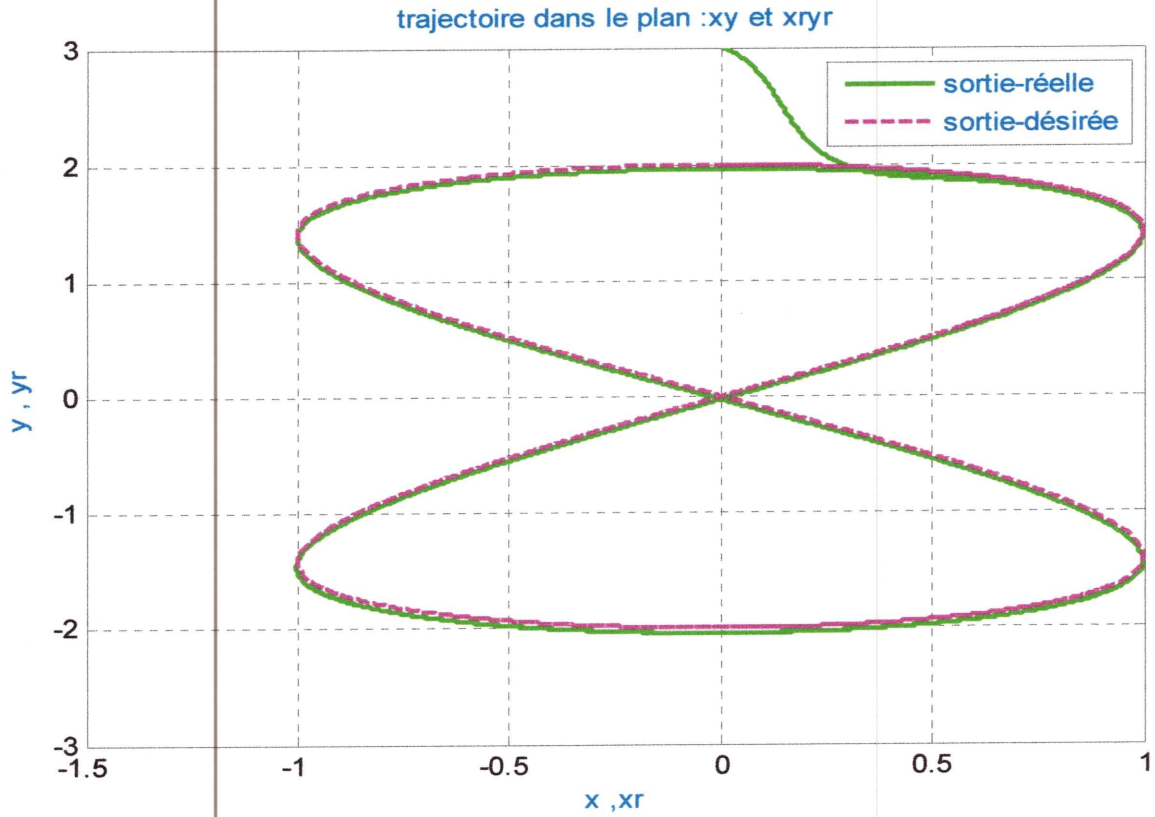


Figure 3.6: Trajectoire tracée par le robot mobile (pour le cas avec perturbation).

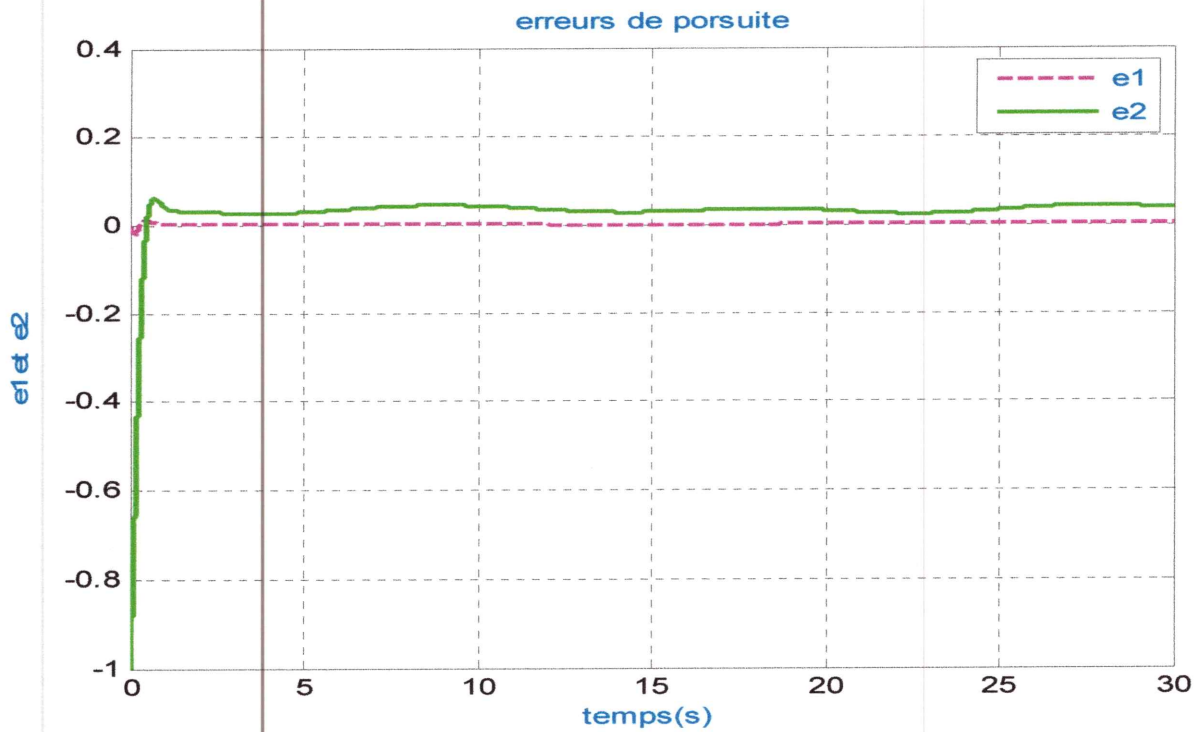


Figure 3.7: Signaux des erreurs de poursuite (pour le cas avec perturbation).

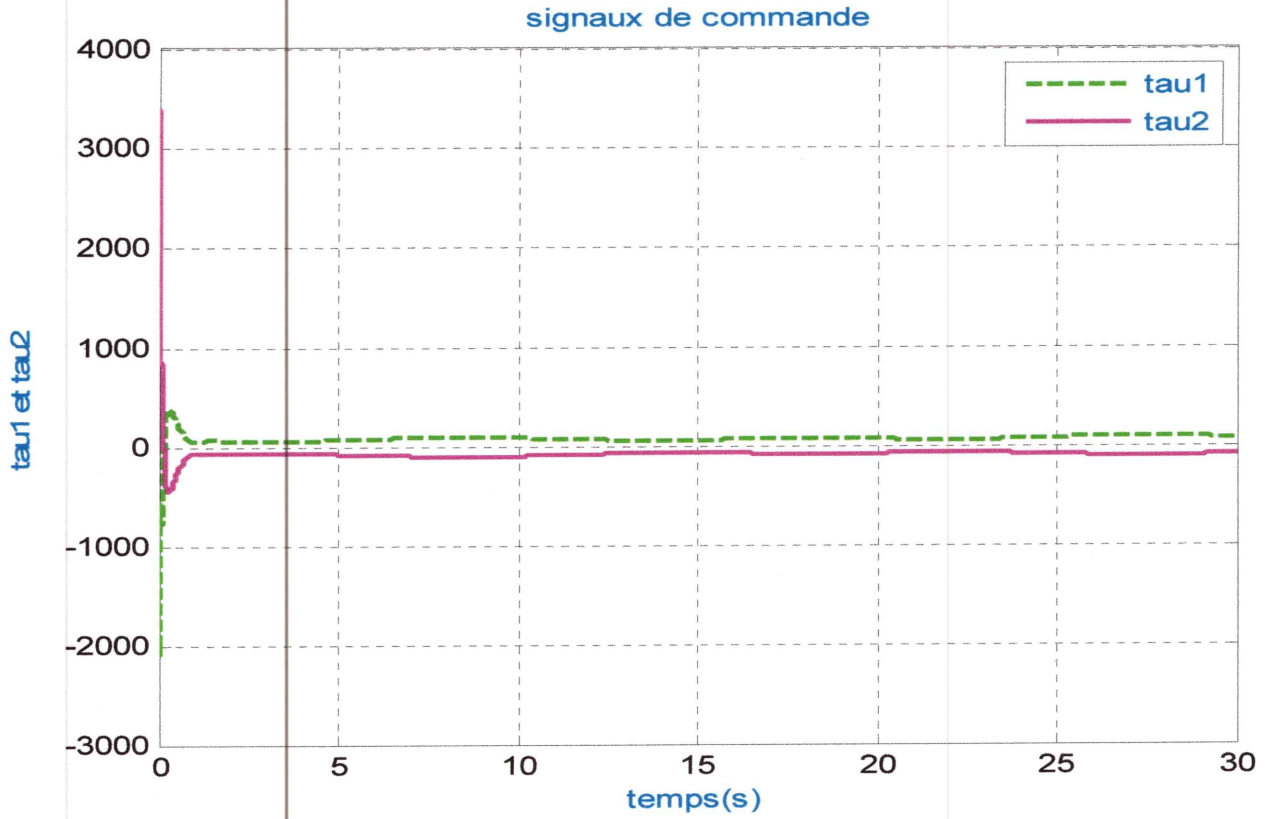


Figure 3.8: Signaux de commande (pour le cas avec perturbation).

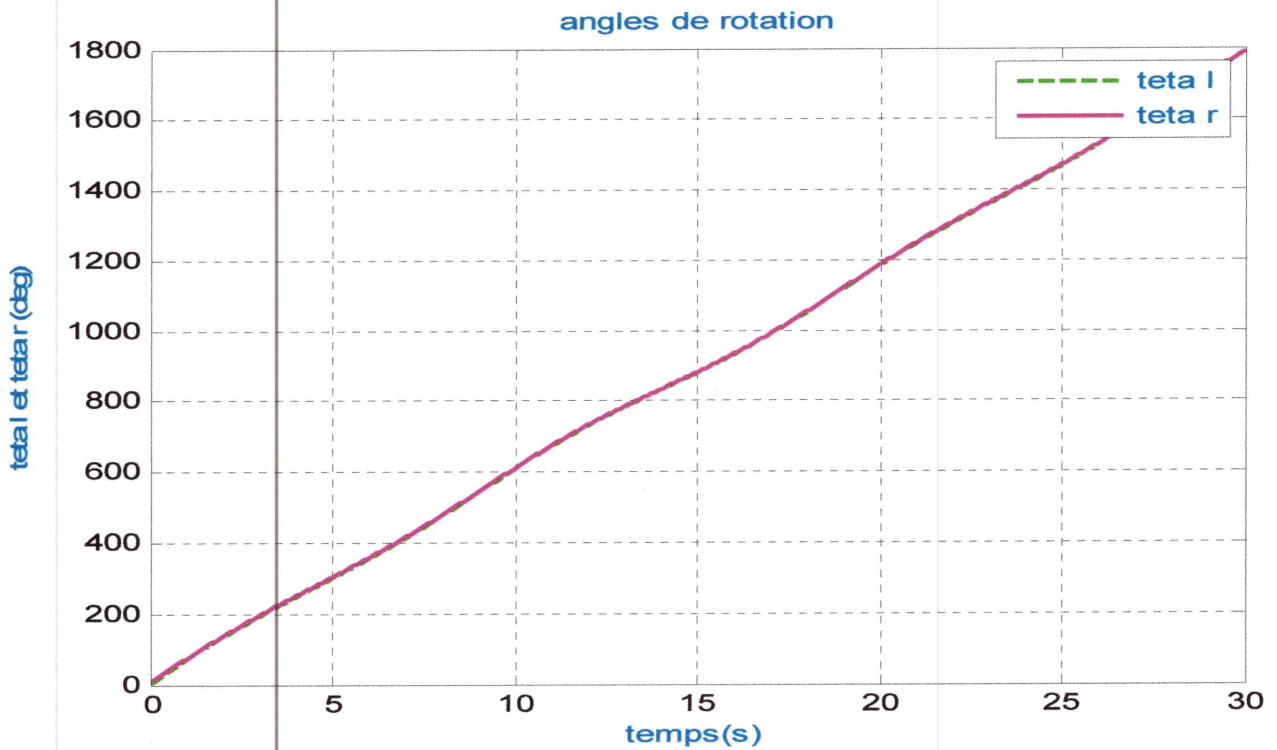


Figure 3.9: Angles de rotation (pour le cas avec perturbation)

### 3.5 Conclusion

Dans ce chapitre, la technique de commande par bouclage linéarisant a été utilisée dans le cadre des systèmes non linéaires. En appliquant cette commande au robot mobile à deux roues, la trajectoire de référence est facilement reproductible par le système. Cependant, cette commande, puisque elle est basée sur le modèle, est moins robuste et très sensible aux perturbations et aux variations paramétriques.

# Chapitre 4

# Chapitre 4

## Commande par mode glissant des systèmes non linéaires



### 4.1 Introduction

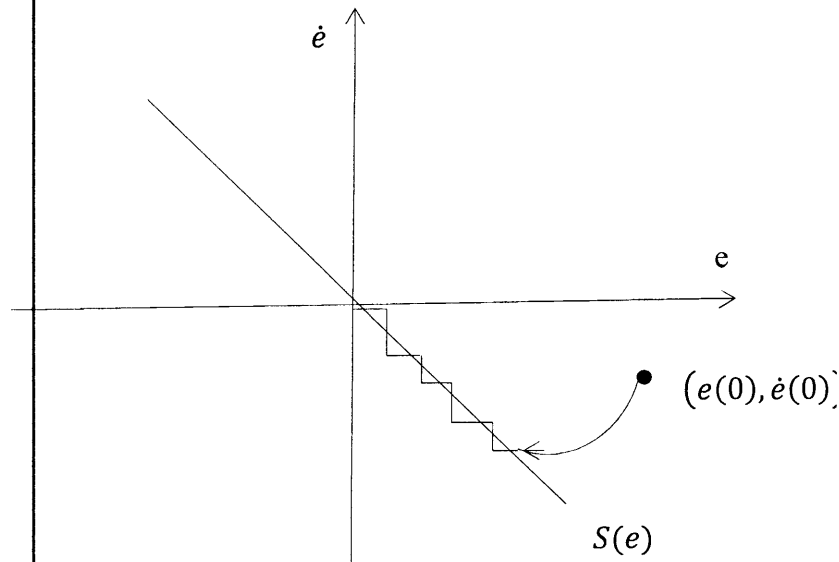
Les lois de commande linéaire donnent de bons résultats dans le cas des systèmes linéaires à paramètre constants. Pour les systèmes non linéaires ou linéaires mais ayant des paramètres non constants, ces lois de commande classique peuvent être insuffisantes car elles sont non robustes, surtout lorsque les exigences sur la précision et autres caractéristiques dynamiques du système sont strictes. On doit faire appel à des lois de commande robuste insensibles aux variations des paramètres, aux perturbations et aux dynamiques incertaines.

Les lois de commande par mode de glissement constituent une bonne solution à ces problèmes d'incertitudes. Ce type de commande présente plusieurs avantages tels que robustesse, précision importante, stabilité, simplicité et temps de réponse très faible. Ceci lui permet d'être particulièrement adaptée pour traiter les systèmes ayant des modèles incertains.

### 4.2. Commande par mode glissant

#### 4.2.1 Fondements théoriques de la commande par mode glissant

La commande par mode glissant consiste à concevoir une loi de commande qui puisse guider le vecteur d'état d'un système donné vers un hyperplan  $S=0$  communément appelé surface de glissement. Une fois ce sous-espace d'état atteint, le vecteur d'état possédera alors une dynamique stable qui dépendra directement du choix de la surface de glissement, et qui fera en sorte que le vecteur d'état convergera vers le point d'équilibre. La figure (4.1) schématise ce processus dans le plan de phase.



**Figure 4.1:** Schématisation du mode de glissement dans le plan de phase

La conception de cette commande peut être divisée en trois étapes principales. Ces étapes concernent [10] :

1. Le choix de la surface de glissement.
2. L'établissement de condition d'existence.
3. La détermination de la loi de commande.

#### 4.2.2 Conception de la commande glissant pour un système monovariante

Considérons la classe des systèmes monovariante suivante:

$$\dot{x} = f(x) + g(x)u + \Delta(x) \quad (4.1)$$

où  $x = [x, \dot{x}, \dots, x^{n-1}]^T$  est le vecteur d'état,  $u$  est l'entrée du système.  $\Delta(x)$  présente les incertitudes étant supposées bornées par une constante connue  $\bar{\Delta}$ , i.e.  $|\Delta(x)| \leq \bar{\Delta}$ .

On définit l'erreur de poursuite par le vecteur  $e = x - x_d = [e, \dot{e}, \dots, e^{n-1}]^T$ . La surface de glissement est une fonction scalaire telle que l'erreur sur la variable à régler glisse sur cette surface et tend vers l'origine du plan de phase. Ainsi, la surface de glissement représente le comportement dynamique désiré.



Dans notre travail, on s'intéresse à une surface de la forme linéaire donnée par:

$$S(t) = \left(\frac{d}{dt} + \lambda\right)^{n-1} e = e^{(n-1)} + \alpha_{n-1} e^{(n-2)} + \dots + \alpha_1 e = 0 \quad (4.2)$$

Avec  $e(x)$  est l'écart sur la variable à régler.  $\lambda$  est une constante positive de conception.  $n$  est le degré relatif qui est égal au nombre de fois qu'il faut dériver la sortie pour faire apparaître la commande.

Par exemple, pour  $n = 3$ , on peut choisir la surface comme  $S(x) = \ddot{e}(x) + 2\lambda\dot{e}(x) + \lambda^2 e(x)$ . L'objectif de cette commande est de garder la surface égale à zéro. Si  $S(x) = 0$ , on peut facilement montrer que  $e(x)$  converge exponentiellement à zéro, si  $\lambda$  est choisi strictement positif.

### 4.2.3 Condition de convergence et d'existence

Les conditions d'existence et de convergence sont les critères qui permettent aux dynamiques du système de converger vers la surface de glissement d'y rester indépendamment à la perturbation. Il y a deux considérations correspondantes au mode de convergence de l'état du système [11].

#### 4.2.3.1 Fonction directe de commutation

La première condition de convergence est proposée et étudiée par Emilyanov et Utkin. Il s'agit de donner à la surface une dynamique convergente vers zéro. Elle s'exprime sous la forme [12] :

$$S(x)\dot{S}(x) < 0 \quad (4.3)$$

#### 4.2.3.2 Fonction de Lyapunov

La fonction de Lyapunov est une fonction scalaire positive ( $V(x) > 0$ ) pour les variables d'état du système. La loi de commande doit faire décroître cette fonction ( $\dot{V}(x) < 0$ ).

L'idée est de choisir une fonction scalaire  $S(x)$  pour garantir l'attraction de la variable à contrôler vers sa valeur de référence et de construire une commande  $u$  tel que le carré de la surface correspond à une fonction de Lyapunov. En définissant la fonction de Lyapunov par :

$$V(x) = \frac{1}{2} S^2(x) \quad (4.4)$$

La dérivée de cette fonction est :

$$\dot{V}(x) = \dot{S}(x)S(x) \quad (4.5)$$

Pour que la fonction  $V(x)$  décroisse, il suffit de s'assurer que sa dérivée est négative.

**Remarque 4.1:** Pour assurer une convergence en temps fini vers la surface de glissement, la condition d'attractivité définie plus haut doit être remplacée par une condition plus fine, dite condition de  $\eta$ -attractive, donnée par:  $\dot{V} = \dot{S}S \leq -\eta|S| - h(S)S$  où  $h(S) = \psi S$ , et  $\eta$  et  $\psi$  sont des constantes strictement positives.

#### 4.2.4 Loi de commande

Une fois la surface de glissement est déterminée, le prochain pas est de concevoir une loi de commande qui rend la surface de glissement attractive à l'état du système, d'une autre manière, la loi de commande doit être capable de pousser l'état du système vers la surface de glissement. Si l'état du système reste sur la surface  $S$ , la stabilité asymptotique de la dynamique de glissement assurera la convergence vers zéro de l'erreur de poursuite [13].

##### a) Expression analytique de la commande

Maintenant, nous nous intéressons au calcul d'une loi de commande attractive du système définie par l'équation (4.1).

Si le système (4.1) est en boucle fermée sous la loi de commande

$$u = g^{-1}(x) (-f(x) + v + K_0 S(x) + K_1 \text{sign}(S)) \quad (4.6)$$

où  $K_0$  et  $K_1 > 0$  sont des constantes de conception, et la fonction signe est définie comme suit:

$$\text{sign}(S) = \begin{cases} 1 & \text{si } S > 0 \\ 0 & \text{si } S = 0 \\ -1 & \text{si } S < 0 \end{cases} \quad (4.7)$$

Si le gain de commande  $g(x)$  est différent de zéro tout au long de la trajectoire d'état du système, la surface  $S$  convergera exponentiellement vers zéro.

**Démonstration**

Soit la fonction de Lyapunov  $V = \frac{1}{2}S^2(x)$ , sa dérivée temporelle est donnée par:

$$\dot{V} = S(x)\dot{S}(x) \quad (4.8)$$

De (4.2), la dérivée temporelle de la surface  $S$  peut s'écrire comme suit:

$$\dot{S} = x_d^{(n)} - x^{(n)} + \alpha_{n-1}e^{(n-1)} + \dots + \alpha_1\dot{e} \quad (4.9)$$

Mettons

$$v = x_d^{(n)} + \alpha_{n-1}e^{(n-1)} + \dots + \alpha_1\dot{e} \quad (4.10)$$

Donc, (4.9) devient:

$$\dot{S} = -x^{(n)} + v \quad (3.11)$$

ou encore

$$\dot{S} = -f(x) - g(x)u + v - \Delta(x) \quad (4.12)$$

En utilisant (4.12),  $\dot{V}$  devient:

$$\dot{V} = S(x)\dot{S}(x) \leq S(x)(-f(x) - g(x)u + v) + \bar{\Delta}|S| \quad (4.13)$$

Si l'on introduit la loi de commande  $u(t)$  donnée par (4.6) dans (4.13), on obtient:

$$\dot{V} \leq -K_0S^2 + (-K_1 + \bar{\Delta})|S| \quad (4.14)$$

où  $K_1 \geq \bar{\Delta} + \eta$ .

Cela implique que

$$\dot{V} \leq -\eta|S| - K_0S^2 \quad (4.15)$$

La dynamique de  $\dot{V}$  est rendue négative ( $\forall S \neq 0$ ), si les coefficients sont choisies positifs. Le choix de la constante  $K_1$  est très influant, car si  $K_1$  est très petit, le temps de réponse est trop long et si elle est trop grande, le broutement peut avoir lieu.

### b) Définition des grandeurs de la commande

La structure d'un tel contrôleur comporte deux parties, une première concernant la linéarisation exacte et une deuxième stabilisante. Cette dernière est très importante dans la technique de commande non linéaire car elle est utilisée pour éliminer les effets d'imprécision du modèle et des perturbations extérieures [12], [13].

Nous posons:

$$u = u_{eq} + u_n \quad (4.16)$$

avec

$$u_{eq} = g^{-1}(x)(-f(x) + v) \quad (4.17)$$

et

$$u_n = g^{-1}(x)(K_0 S + K_1 \text{sign}(S)) \quad (4.18)$$

où  $u_{eq}$  correspond à la loi de commande équivalente proposée par Filipov et Utkin [13]. Cette commande est considérée comme la commande la plus directe et la plus simple. Elle est calculée en connaissant que le comportement du système durant le mode de glissement est décrit par  $\dot{S} = 0$ .

La commande  $u_n$  est déterminée pour garantir l'attractivité de la variable  $x$  à contrôler vers la surface. En d'autre terme, elle définit le comportement dynamique du système durant le mode de convergence.

Cette loi de commande est certes robuste vis-à-vis des perturbations paramétriques et externes, elle est caractérisée par des propriétés implicites de robustesse. Mais présente quelques inconvénients majeurs tels que:

- L'utilisation du terme  $\text{sign}(S)$  dans le signal de commutation provoque le phénomène de broutement (chattering) qui peut exciter les hautes fréquences et détériorer le système commandé.

- La mise en œuvre du signal de commutation nécessite la détermination de la constante qui dépend des perturbations paramétriques et externes, ce qui est difficile si ce n'est pas impossible.

Pour résoudre le problème du chattering, la fonction signe a été remplacée par des autres fonctions plus lisses (Tanh, Arctan, Sat,...etc). D'autres auteurs ont proposé d'utiliser un système flou pour avoir une approximation de la fonction lisse.

### 4.3 Commande à mode glissant pour les systèmes multivariables

Considérons la classe suivante des systèmes non linéaires multivariables décrite par :

$$\begin{aligned} y_1^{(r_1)} &= f_1(x) + \sum_{j=1}^p g_{1j}(x)u_j + \Delta_1(x), \\ &\vdots \\ y_p^{(r_p)} &= f_p(x) + \sum_{j=1}^p g_{pj}(x)u_j + \Delta_p(x). \end{aligned} \quad (4.19)$$

Où  $x = [y_1, \dot{y}_1, \dots, y_1^{(r_1-1)}, \dots, y_p, \dot{y}_p, \dots, y_p^{(r_p-1)}]^T \in R^r$  est le vecteur d'état complet qui est supposé disponible pour la mesure et  $r_1 + \dots + r_p = r$ ,  $u = [u_1, \dots, u_p]^T \in R^p$  est le vecteur d'entrée de commande,  $y = [y_1, \dots, y_p]^T \in R^p$  est le vecteur de sortie, et  $f_i(x), i = 1, \dots, p$  sont des fonctions non linéaires continues inconnues, et  $g_{ij}(x), i, j = 1, \dots, p$  sont des fonctions non linéaires continues inconnues de classe  $C^1$ .  $\Delta_j(x)$  Sont les incertitudes du système.

Notons

$$y^{(r)} = [y_1^{(r_1)} \dots y_p^{(r_p)}]^T, \quad F(x) = [f_1(x) \dots f_p(x)]^T, \quad G(x) = \begin{bmatrix} g_{11}(x) & \dots & g_{1p}(x) \\ \vdots & \ddots & \vdots \\ g_{p1}(x) & \dots & g_{pp}(x) \end{bmatrix}$$

$$\text{Et } \Delta(x) = [\Delta_1(x), \dots, \Delta_p(x)]^T.$$

Donc, le système (4.19) peut être réécrit sous la forme compacte suivante

$$y^{(r)} = F(x) + G(x)u + \Delta(x) \quad (4.20)$$

où  $F(.) \in R^p$  et  $G(.) \in R^{p \times p}$ .

Notre objectif consiste à concevoir une loi de commande  $u$  qui permet de forcer les sorties  $y_1, \dots, y_p$  à suivre les trajectoires de références prédéterminées  $y_{d1}, \dots, y_{dp}$ , tout en gardant la bornitude de tous les signaux du système en boucle fermée.

#### Hypothèse 4.1 :

1) Le vecteur d'état  $x$  est supposé disponible pour la mesure.

2) Le vecteur des trajectoires désirées,  $x_d = [y_{d1}, \dot{y}_{d1}, \dots, y_{d1}^{(r_1)}, \dots, y_{dp}, \dot{y}_{dp}, \dots, y_{dp}^{(r_p)}]^T$ , est supposé être continu, borné et disponible pour la mesure. Donc,  $x_d \in \Omega_{x_d} \subset R^{r+p}$ , avec  $\Omega_{x_d}$  est un ensemble compact connu.

Définissons les erreurs de poursuite comme suit :

$$\begin{aligned} e_1 &= y_{d1} - y_1 \\ &\vdots \\ e_p &= y_{dp} - y_p \end{aligned} \quad (4.21)$$

Et le vecteur des erreurs filtrées comme

$$S = [S_1, \dots, S_p]^T \quad (4.22)$$

Avec

$$S_i = \left[ \frac{d}{dt} + \lambda_i \right]^{r_i-1} e_i, \text{ pour } \lambda_i > 0, \quad \forall i = 1, \dots, p. \quad (4.23)$$

Ou encore, (4.23) peut s'écrire comme suit

$$S_i = \lambda_i^{r_i-1} e_i + (r_i - 1) \lambda_i^{r_i-2} \dot{e}_i + \dots + (r_i - 1) \lambda_i e_i^{(r_i-2)} + e_i^{(r_i-1)}, \quad (4.24)$$

Avec  $i = 1, \dots, p$ .

Notons que si nous choisissons  $\lambda_i > 0$ , avec  $i = 1, \dots, p$ , alors les zéros du polynôme

$H_i(s) = \lambda_i^{r_i-1} + (r_i - 1) \lambda_i^{r_i-2} s + \dots + (r_i - 1) \lambda_i s^{r_i-2} + s^{r_i-1}$  correspondant à l'équation caractéristique de  $S_i = 0$  sont tous dans le demi-plan gauche ouvert.

Par conséquent, le vecteur  $S$  des surfaces prend la forme suivante :

$$S = C^T E \quad (4.25)$$

Avec

$$C^T = \text{diag}[C_1^T C_2^T \dots C_p^T]_{(p \times r)}, \quad E = [E_1^T E_2^T \dots E_p^T]^T_{(r \times 1)} \quad \text{où} \quad E_i = [e_i \dot{e}_i \dots e_i^{(r_i-2)} e_i^{(r_i-1)}]^T \quad \text{et}$$

$$C_i^T = [\lambda_i^{r_i-1} (r_i-1)\lambda_i^{r_i-2} \dots (r_i-1)\lambda_i \ 1], \quad S_i = C_i^T E_i.$$

Et la dynamique  $S_i$  est décrite par :

$$\dot{S}_i = C_{ri}^T E_i + e_i^{(r_i)}, \quad \text{et } i = 1, \dots, p \quad (4.26)$$

Où

$$C_{ri}^T = [0 \ \lambda_i^{r_i-1} \ (r_i-1)\lambda_i^{r_i-2} \dots \ 0.5(r_i-1)(r_i-2)\lambda_i^2 \ (r_i-1)\lambda_i] \quad (4.27)$$

La dynamique de  $S$  peut s'écrire sous la forme compacte suivante

$$\dot{S} = C_r^T E + y_d^{(r)} - y^{(r)} \quad (4.28)$$

Avec

$$C_r^T = \text{diag}[C_{r1}^T C_{r2}^T \dots C_{rp}^T]_{(p \times r)}, \quad C_{ri}^T = [0 \ \lambda_i^{r_i-1} \ (r_i-1)\lambda_i^{r_i-2} \dots \ 0.5(r_i-1)(r_i-2)\lambda_i^2 \ (r_i-1)\lambda_i], \quad \text{et}$$

$$y_d^{(r)} = [y_{d1}^{(r_1)} \ y_{d2}^{(r_2)} \ \dots \ y_{dp}^{(r_p)}]^T.$$

En utilisant (4.20), la dynamique (4.28) devient

$$\dot{S} = C_r^T E + y_d^{(r)} - F(x) - G(x)u - \Delta(x) \quad (4.29)$$

Pour stabiliser exponentiellement la dynamique (4.29), on peut concevoir la commande suivante

$$u = u_{eq} + u_n \quad (4.30)$$

avec

$$u_{eq} = G^{-1}(x) [-F(x) + C_r^T E + y_d^{(r)}] \quad (4.31)$$

et

$$u_n = G^{-1}(x)[-K_1 S - K_0 \text{signe}(S)] \quad (4.32)$$

où  $u_{eq}$  est la loi de commande équivalente.  $u_n$  est la commande non linéaire avec  $K_0 = \text{Diag}[k_{01}, \dots, k_{0p}]$ , où  $k_{0i} > \bar{\Delta}_i \geq |\Delta_i(x)|, i = 1, \dots, p$ , sont des constantes positives qui seront par la suite définies.  $K_1 = \text{Diag}[k_{11}, \dots, k_{1p}]$ , avec  $k_{1i} > 0, i = 1, \dots, p$ , sont des constantes libres de conception.

**Théorème 4.1 :** *Considérons le système (4.19). Si l'Hypothèse 4.1 est satisfaite, alors la loi de commande idéale définie par (4.31)-(4.32) garantit les propriétés suivantes :*

- *Tous les signaux du système en boucle fermée sont bornés.*
- *Les erreurs de poursuite et leurs dérivées convergent asymptotiquement vers zéro, i.e.  $e_i^{(j)}(t) \rightarrow 0$  quand  $t \rightarrow \infty$  pour  $i = 1, \dots, p$  et  $j = 0, 1, \dots, r_i - 1$ .*

**Preuve du Théorème 4.1 :**

Nous pouvons définir une fonction candidate de Lyapunov comme suit :

$$V = \frac{1}{2} S^T S \quad (4.33)$$

La dérivée temporelle de  $V$  est donnée par

$$\dot{V} = S^T \dot{S} \quad (4.34)$$

En utilisant (4.29)-(4.32),  $\dot{V}$  peut être bornée comme suit:

$$\dot{V} \leq -S^T K S \quad (4.35)$$

A partir de (4.29), et en utilisant le théorème de Barbalat, nous pouvons facilement démontrer comme que  $s_i \rightarrow 0$ , quand  $t \rightarrow \infty$ . Donc, toutes les erreurs de poursuite et leurs dérivées convergent vers zéro, i.e.  $e_i^{(j)}(t) \rightarrow 0$  quand  $t \rightarrow \infty$  pour  $i = 1, \dots, p$  et  $j = 0, 1, \dots, r_i - 1$ .



#### 4.4 Simulation

Pour les illustrer les performances de la méthode de commande à mode glissant, nous faisons des simulations numériques sur ce robot à deux roues. Pour la résolution numérique des équations différentielles, nous avons adopté la méthode de RK4. La trajectoire de références est choisie comme suit:

$$\begin{aligned} x_r &= \sin(\alpha t) \\ y_r &= \rho \cos(\beta t) \end{aligned} \quad (4.36)$$

avec  $\alpha = 2/3$ ,  $\beta = 1/3$  et  $\rho = 2$ . Dans le plan  $x$ - $y$ , ce chemin de référence prend la forme d'un huit. Les paramètres du robot sont choisis comme dans les chapitres 3.

L'objectif de la commande est de forcer les sorties du système  $y = P + d(\cos(\phi), \sin(\phi))$  à suivre la trajectoire de référence (4.36).

Pour tester la robustesse de la commande proposée comme dans les chapitres précédents, nous avons considéré deux cas: i.e. avec perturbation ( $\Delta = 0.15 [c_4 \dot{\theta}_l - c_5 \dot{\theta}_r; c_5 \dot{\theta}_l - c_4 \dot{\theta}_r]^T$ ) et sans perturbation ( $\Delta = [0 \ 0]^T$ ).

Les paramètres de la commande à mode glissant sont choisis comme suit:  $\lambda_i = 2$ ,  $K_0 = \text{diag}(10,10)$ ,  $K_1 = \text{diag}(2,2)$ .

Les figures (4.2)-(4.6) montrent les résultats de simulation obtenus en utilisant la commande à mode glissant (4.30), pour le cas sans perturbations. Ces résultats montrent que les performances de poursuite sont très bonnes et tous les signaux du système en boucle fermée sont bornés et les erreurs de poursuite et les surfaces de glissement convergent rapidement vers zéro.

Les résultats de simulation pour le cas avec perturbation sont représentés sur les Figures (4.7)-(4.11). Nous remarquons d'après ces figures que les performances en terme de poursuite sont très bonnes et ne sont pas dégradées malgré la présence des perturbations externes, et les commandes sont bornées et admissibles. Ces résultats de simulation obtenus ont montré de grande robustesse vis-à-vis les incertitudes paramétriques et les perturbations ajoutées.

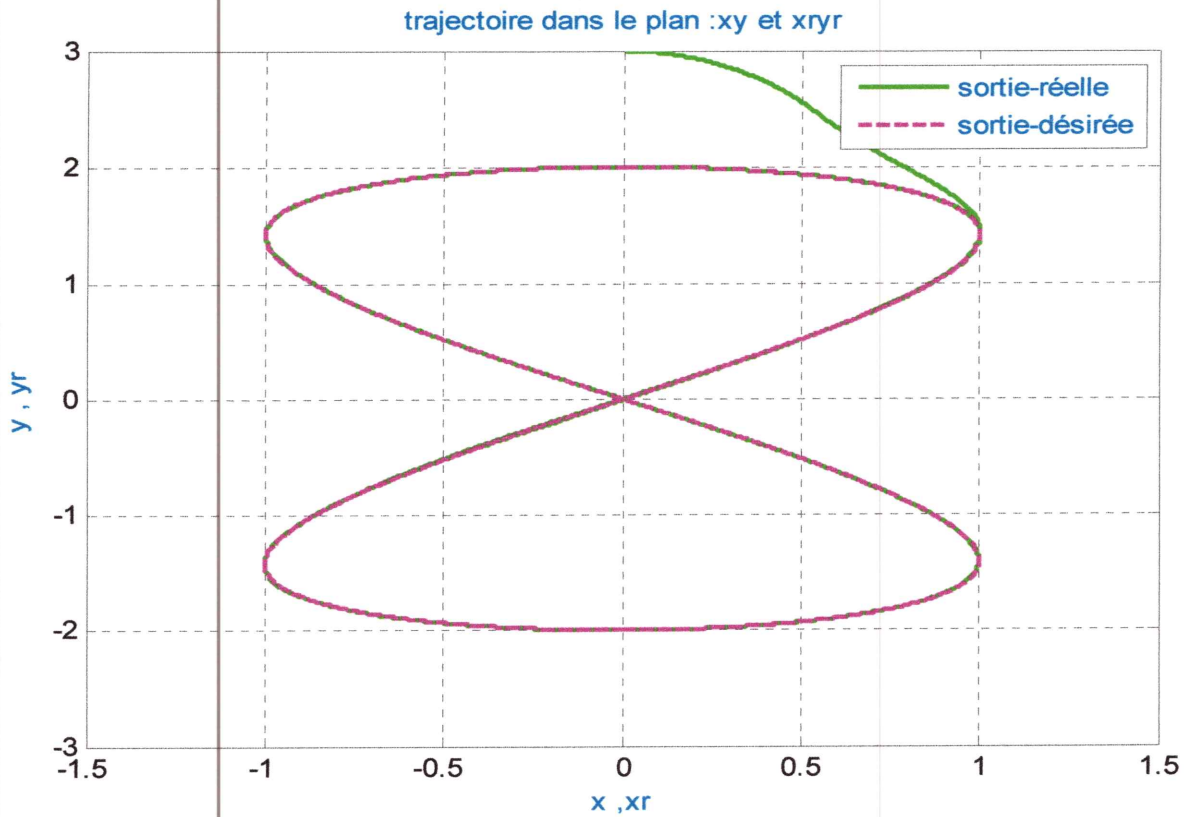


Figure 4.2 : Trajectoire tracée par le robot mobile (pour le cas sans perturbations)

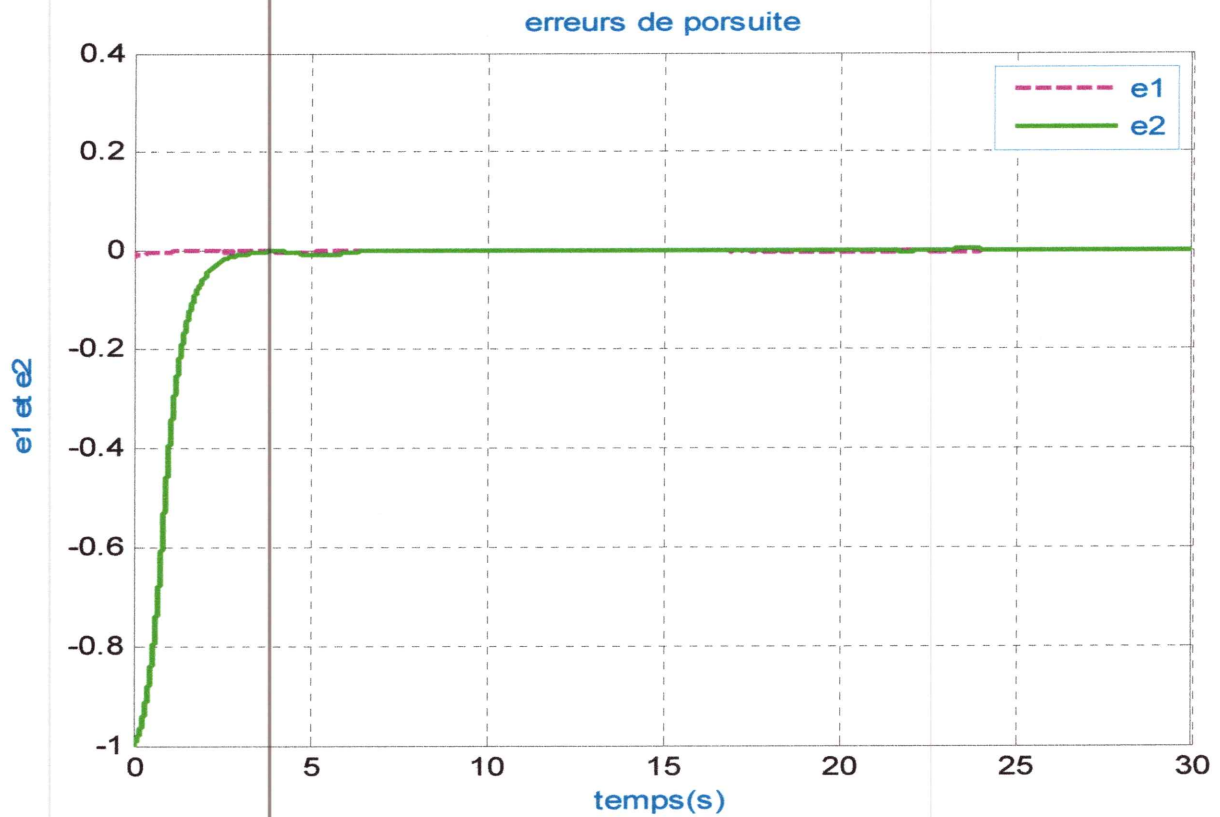


Figure 4.3: Signaux d'erreurs de poursuite (pour le cas sans perturbations).

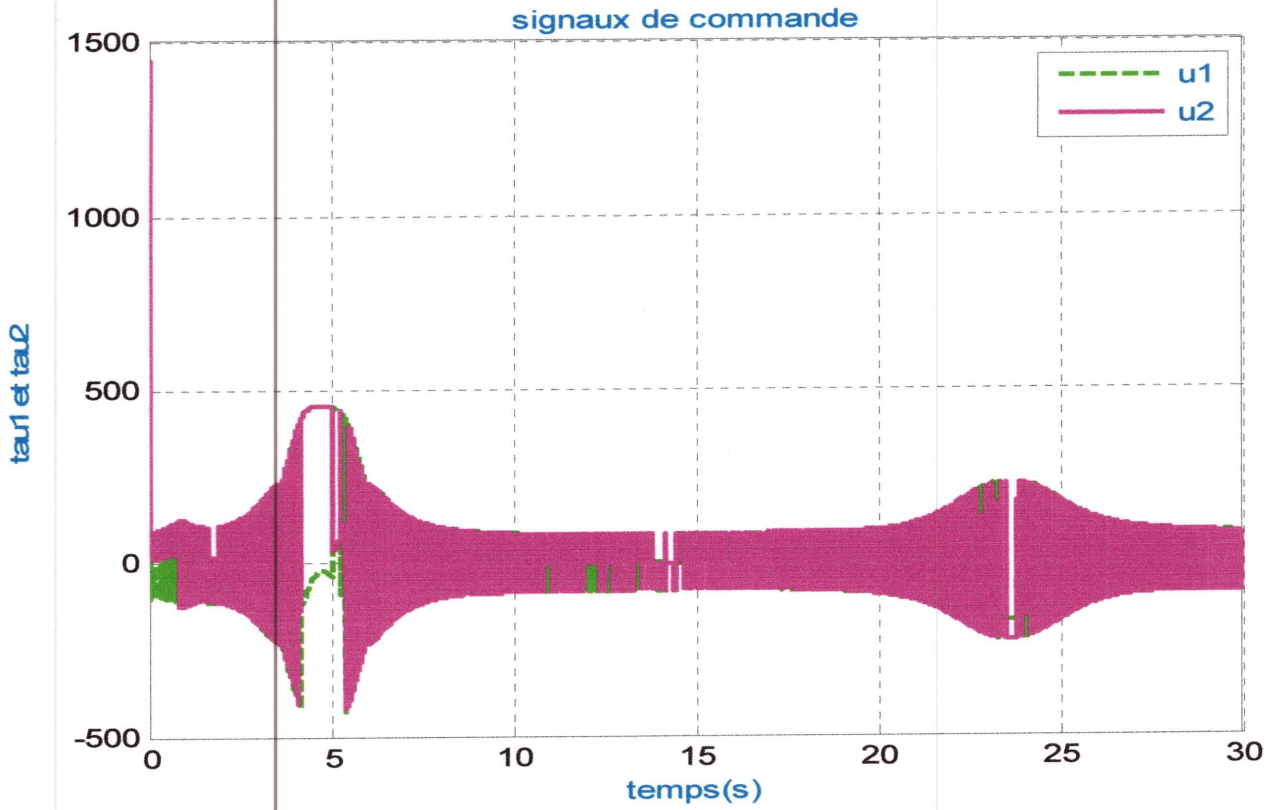


Figure 4.4: Signaux de commande (pour le cas sans perturbations).

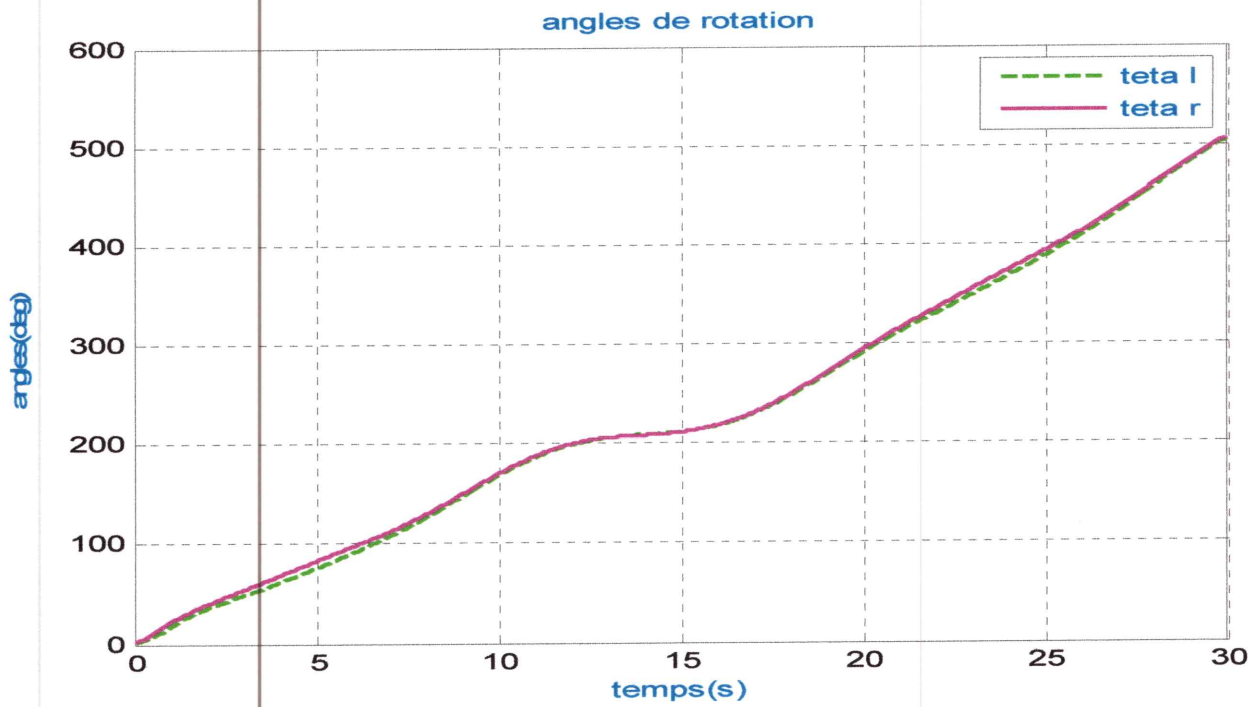


Figure 4.5: Angles de rotation (pour le cas sans perturbations).

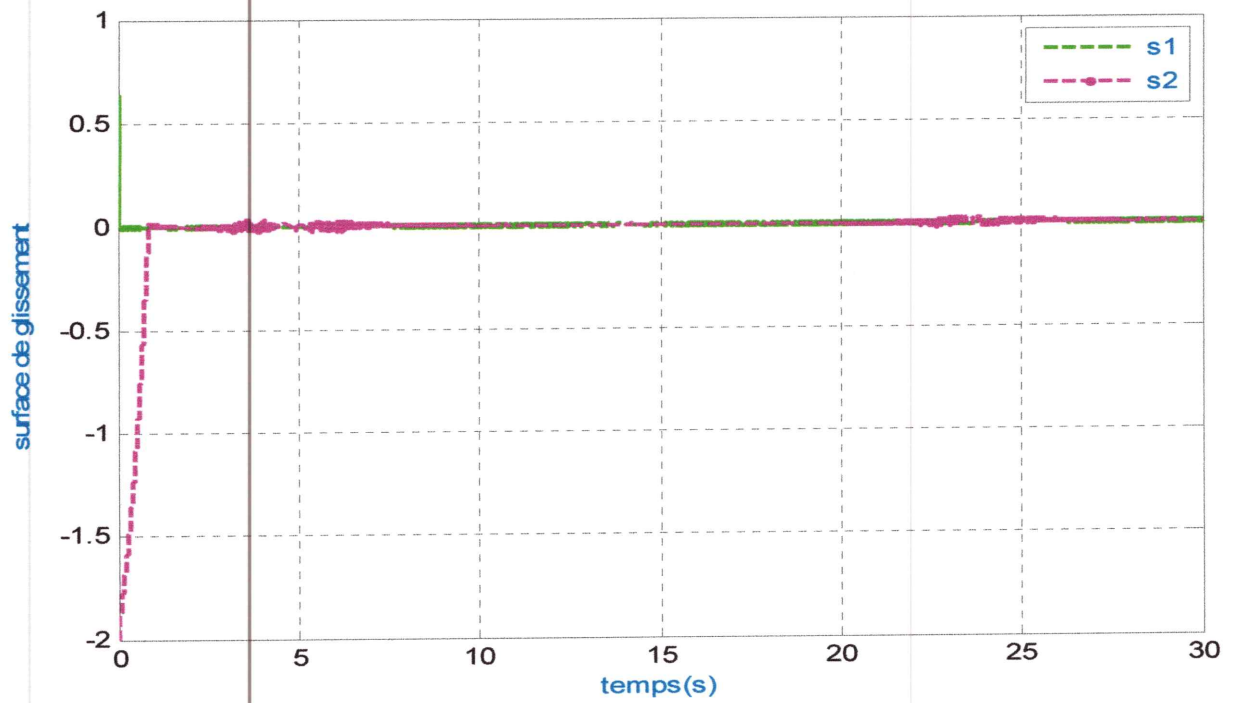


Figure 4.6: Surfaces de glissement (pour le cas sans perturbations).

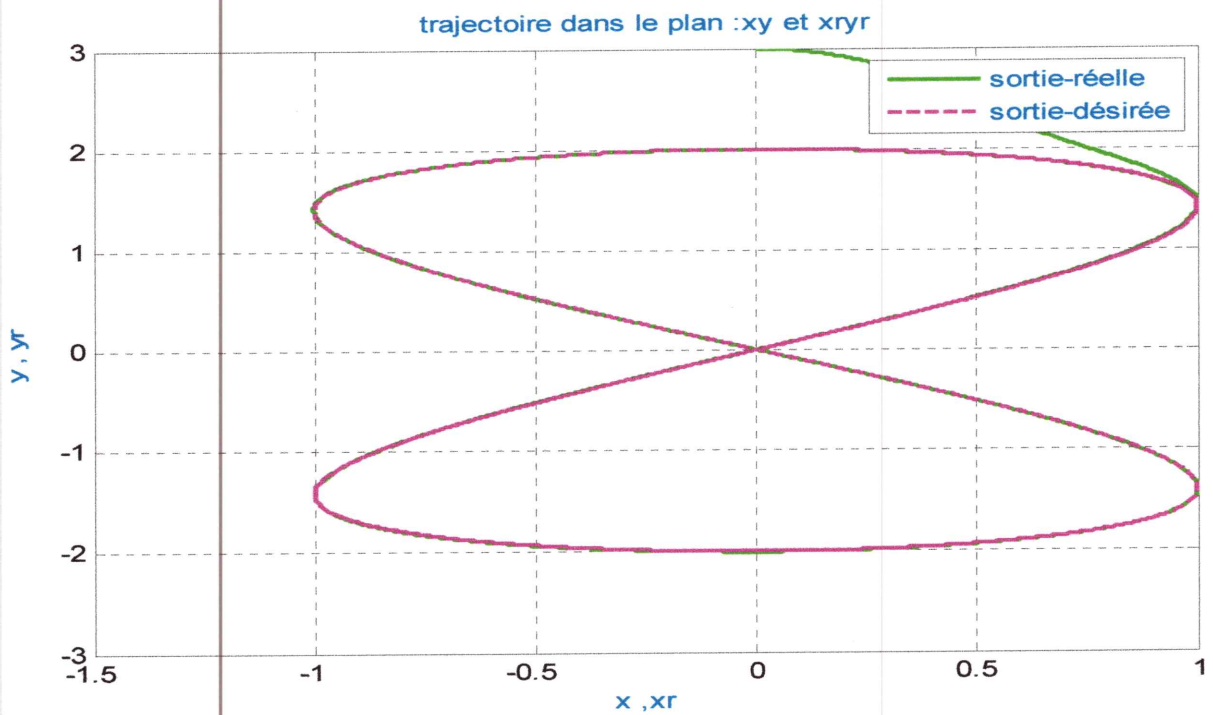


Figure 4.7: Trajectoire tracée par le robot mobile (pour le cas avec perturbation).

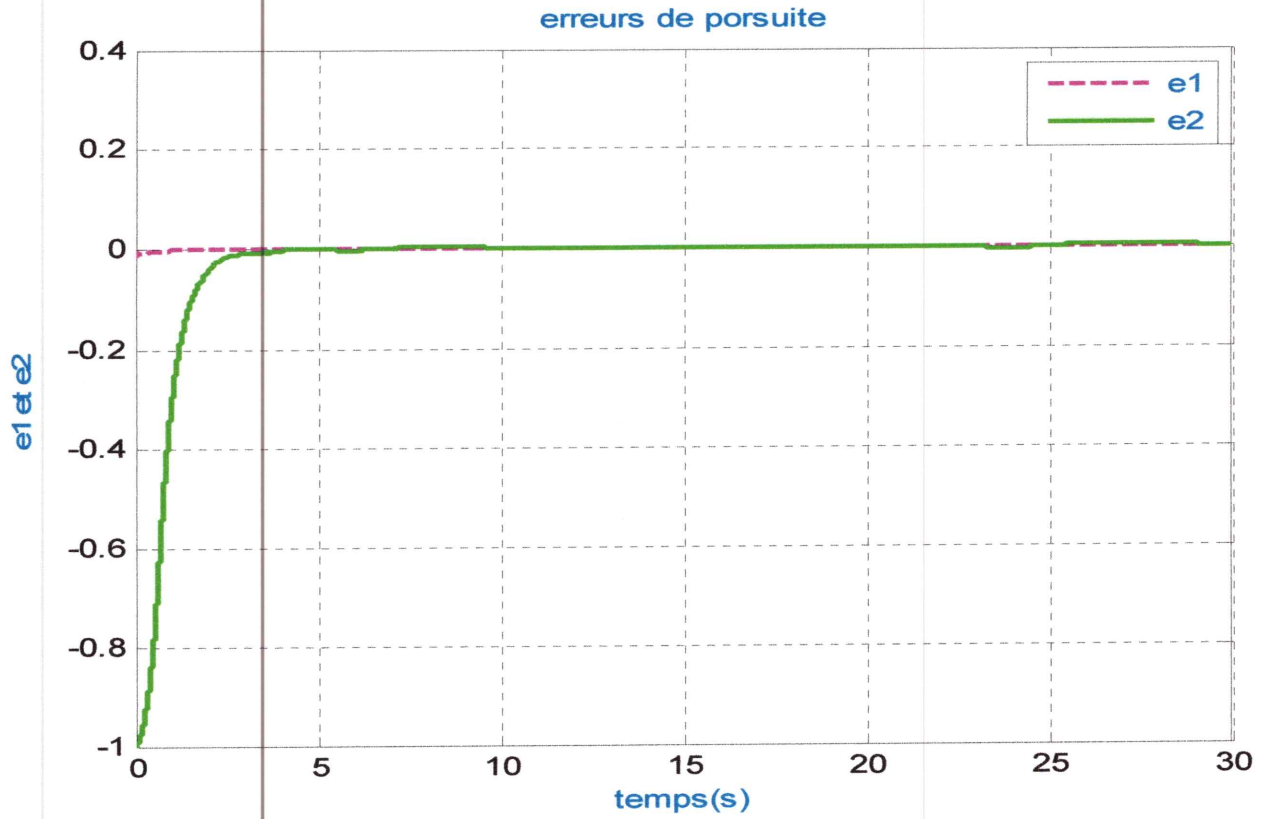


Figure 4.8: Signaux d'erreurs de poursuite (pour le cas avec perturbation).

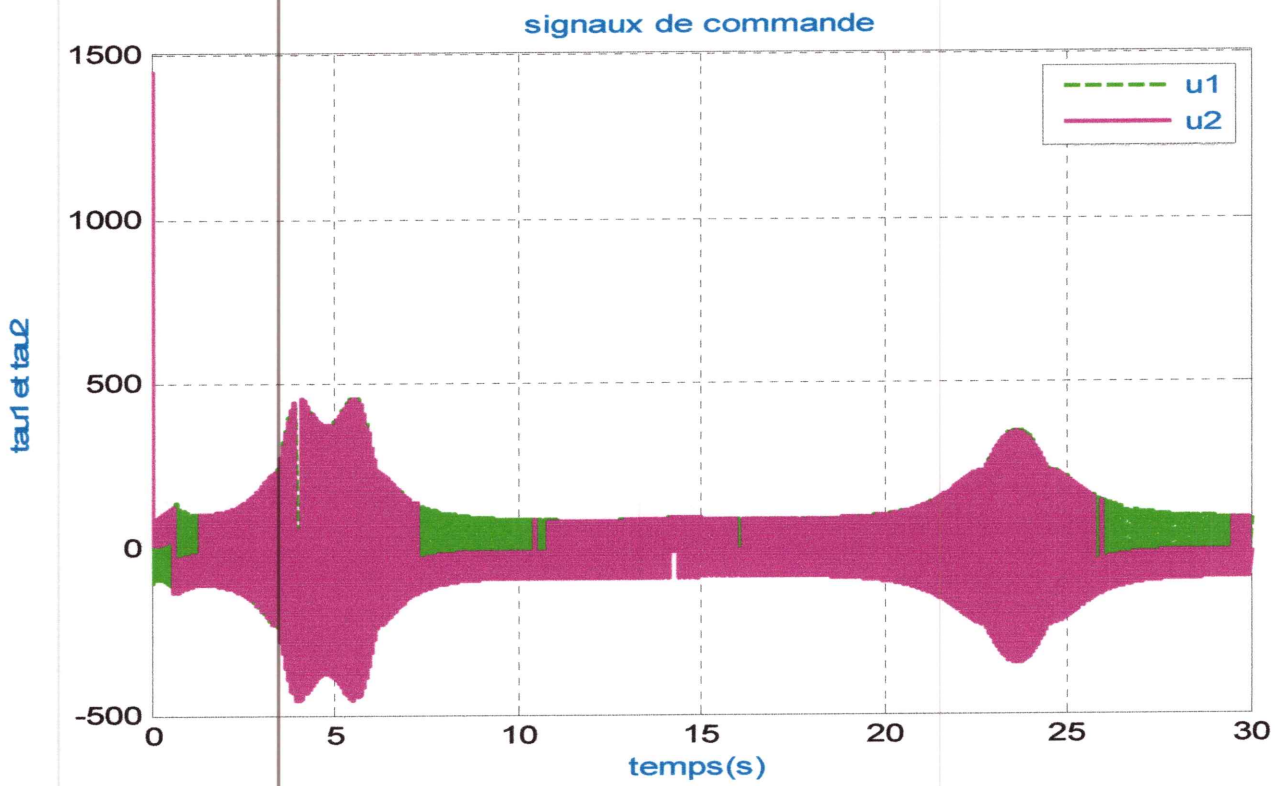


Figure 4.9: Signaux de commande (pour le cas avec perturbation).

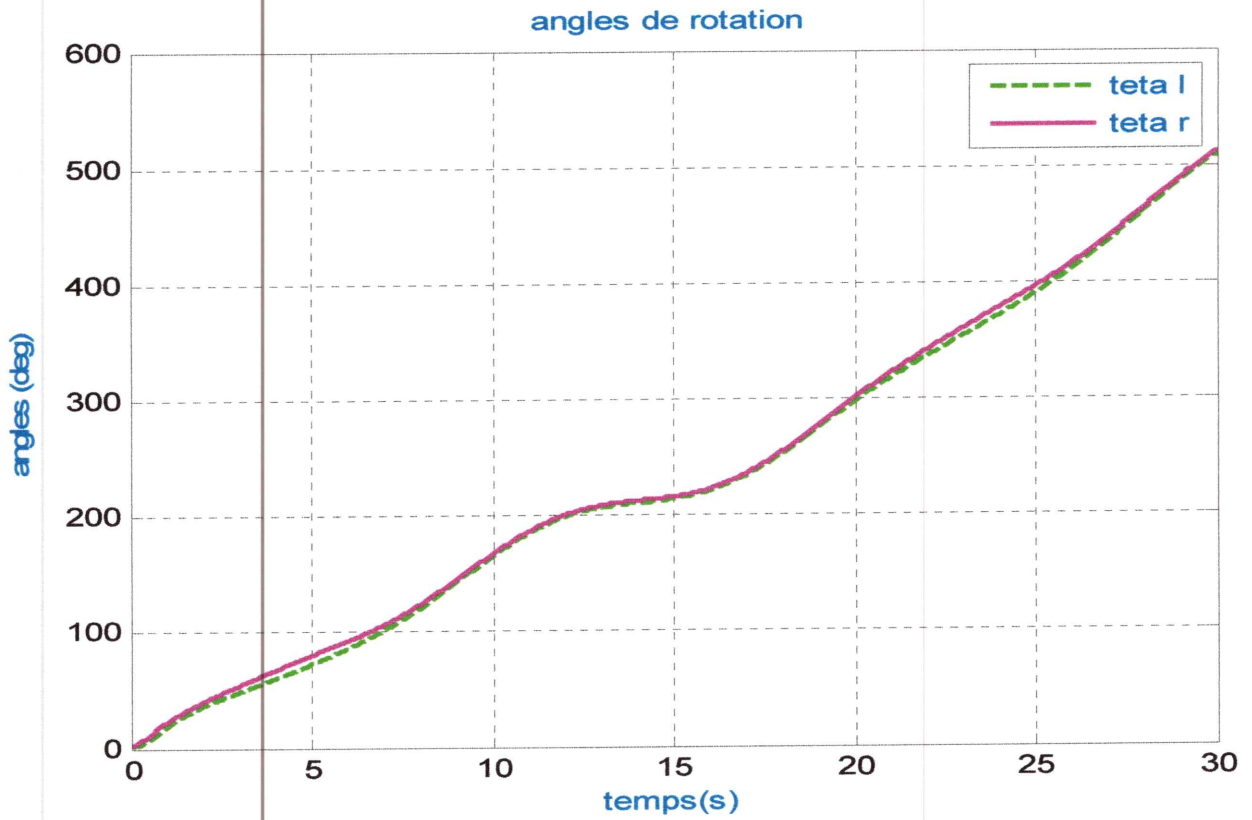


Figure 4.10: Angles de rotation (pour le cas avec perturbation).

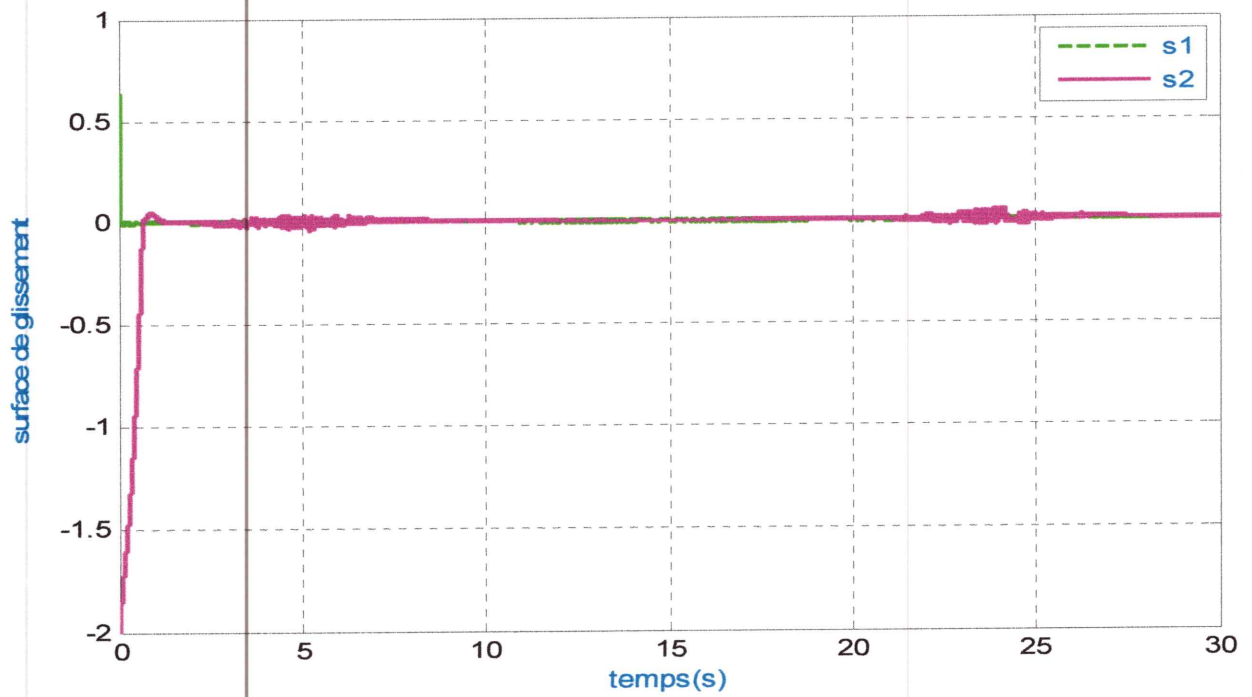


Figure 4.11: Surfaces de glissement (pour le cas avec perturbation).

## 4.5 Conclusion

Dans cette section, nous nous sommes principalement intéressés à l'étude de la commande à mode glissant pour les systèmes monovariables aussi bien que les systèmes multivariables avec perturbations et variations paramétriques et à son application au robot mobile à deux roues.

L'intérêt majeur de cette approche se situe dans la simplicité de mise en œuvre de la loi de commande et sa forte robustesse vis-à-vis les incertitudes paramétriques et les perturbations ajoutées. Cependant, son inconvénient majeur réside dans l'apparition du phénomène de chattering qui se manifeste dans les grandeurs asservies et le signal de commande.

# Chapitre 5



# *Chapitre 5*

## *Commande adaptative floue des systèmes*

### *Non linéaires*

#### **5.1 Introduction**

Les systèmes flous ont été appliqués avec succès pour beaucoup de problèmes de commande. Ils fournissent l'avantage d'être capables d'incorporer pendant la construction et l'apprentissage aussi bien des informations linguistiques que numériques. Il a démontré que les systèmes flous sont capables d'approximer n'importe quelle fonction continue sur un ensemble compact [14]. Basés sur cette propriété importante d'approximation universelle, plusieurs schémas de commande floue adaptative ont été proposés pour les systèmes non linéaires inconnus ou mal connus [15][ 16][ 17]. Leur stabilité a été démontrée à l'aide de la méthode de Lyapunov. Comparés aux schémas de commande adaptative conventionnelle, l'avantage principal des schémas de commande floue adaptative (ou encore les schémas de commande à base d'approximateurs universels en générale) est que la condition sur la paramétrisation linéaire des non-linéarités incertaines n'est plus exigée.

Cette section présente les notions de base de la logique floue et les différents mécanismes et outils mathématiques nécessaires pour son utilisation dans le domaine de la commande des systèmes. Les deux types de systèmes (modèles) flous sont comparés et étudiés en détails. La propriété d'approximation universelle des systèmes flous est aussi discutée. Ensuite, une commande floue adaptative est conçue pour une classe des systèmes non linéaires multivariables. Cette commande est ensuite testée sur notre robot mobile à deux roues.

#### **5.2 Systèmes Flous**

##### **5.2.1 Logique floue**

###### **a) La logique floue aujourd'hui**

La logique floue, dans la plupart des applications actuelles, permet de prendre en compte toutes sortes de connaissances qualitatives de concepteurs et d'opérateurs dans l'automatisation des systèmes. Les nombreuses applications dans l'électroménager et l'électronique grand public réalisées notamment au Japon en ont été l'élément déclenchant. Machines à laver sans réglage, caméscopes anti-bougé et de nombreuses autres innovations ont fait connaître le terme « logique floue » à un large public. Dans l'automobile les transmissions automatiques, les contrôles d'injection et d'anticliquetis, l'air conditionné sont réalisés sur des véhicules de série grâce à la logique floue [18]. Dans le domaine des processus de production, continue et par lots, et dans les automatismes, les applications se sont également multipliées.

La logique floue s'y développe car il s'agit d'une approche essentiellement pragmatique, efficace et générique. On dit parfois qu'elle permet de systématiser ce qui est du domaine de l'empirisme, et donc difficile à maîtriser. La théorie des ensembles flous fournit une méthode pertinente et facilement réalisable dans des applications temps réel; elle permet de transcrire et rendre dynamiques les connaissances des concepteurs ou des opérateurs. Cet aspect adaptable et universel de la logique floue permet de s'attaquer à l'automatisation de procédures telles que : la mise en route, le réglage de paramètres, pour lesquelles peu d'approches existaient auparavant.

### **b) Historique de la logique floue**

**Apparition de la logique floue :** Le terme d'ensemble flou apparaît pour la première fois en 1965 lorsque le professeur Lotfi A. Zadeh, de l'université de Berkeley aux USA, publie un article intitulé « *Ensembles flous* » (*Fuzzy sets*), [19]. Il a réalisé depuis de nombreuses avancées théoriques majeures dans le domaine et a été rapidement accompagné par de nombreux chercheurs développant des travaux théoriques.

**Premières applications :** Parallèlement, certains chercheurs se sont penchés sur la résolution par logique floue de problèmes réputés difficiles. Ainsi en 1975, le professeur Mamdani à Londres développe une stratégie pour le contrôle des procédés et présente les résultats très encourageants qu'il a obtenus sur la conduite d'un moteur à vapeur. En 1978, la société danoise F.L.Smidth réalise le contrôle d'un four à ciment. C'est là la première véritable application industrielle de la logique floue.

**Essor :** C'est au Japon, où la recherche n'est pas seulement théorique mais également très applicative, que la logique floue connaît son véritable essor. A la fin des années 1980, c'est d'un véritable boom qu'il faut parler. Les produits grand public, machines à laver, appareils

photographiques et autres caméscopes estampillés « *fuzzy logic* » ne se comptent plus. Dans l'industrie, le traitement des eaux, les grues portuaires, les métros, les systèmes de ventilation et de climatisation sont touchés. Enfin, des applications existent dans des domaines non techniques tels que : la finance ou le diagnostic médical.

### c) Intérêt et utilisation de la logique floue pour le contrôle

**Intérêt :** La logique floue trouve ses origines dans un certain nombre de constatations [18] :

- La connaissance que l'être humain a d'une situation quelconque est généralement imparfaite,
  - elle peut être incertaine (il doute de sa validité),
  - ou imprécise (il a du mal à l'exprimer clairement).
- L'être humain résout souvent des problèmes complexes à l'aide de données approximatives : la précision des données est souvent inutile ; par exemple pour choisir un appartement, il pourra prendre en compte la surface, la proximité de commerces, la distance du lieu de travail, le loyer, sans pour autant avoir besoin d'une valeur très précise de chacune de ces données.
- Dans l'industrie ou les techniques, les opérateurs résolvent souvent des problèmes complexes de manière relativement simple et sans avoir besoin de modéliser le système. De même, tout le monde sait qu'un modèle mathématique n'est pas nécessaire pour conduire une voiture et pourtant une voiture est un système très complexe.
- Plus la complexité d'un système augmente, moins il est possible de faire des affirmations précises sur son comportement.

De ces constatations viennent naturellement les déductions suivantes :

- plutôt que de modéliser le système, il est souvent intéressant de modéliser le comportement d'un opérateur humain face au système ;
- plutôt que par des valeurs numériques précises, le fonctionnement doit être décrit par des qualificatifs globaux traduisant l'état approximatif des variables.

**Utilisation pour le contrôle :** La logique floue est bien connue des automaticiens pour ses applications dans le contrôle/commande de procédés, appelé alors couramment « contrôle/commande flou(e) ». Tout comme un contrôleur (ou correcteur) classique, le contrôleur flou s'insère dans la boucle de régulation et calcule la commande à appliquer au

procédé (système) suivant une ou plusieurs consignes et une ou plusieurs mesures effectuées sur celui-ci.

Les bases de règles floues sont intéressantes en commande car elles permettent :

- de prendre en compte une expertise existante de nature qualitative,
- de prendre en compte des variables que l'on sait difficilement intégrer dans la boucle,
- d'améliorer le fonctionnement de contrôleurs classiques, par :
  - autoréglage hors ligne ou en ligne des paramètres de ces contrôleurs,
  - modification de leur sortie (feedforward) en fonction d'événements qui ne peuvent pas être pris en compte par une technique classique.

## 5.2.2 Notions de Base de la logique floue

Dans cette partie, nous présentons les notions de base de la logique floue. Nous abordons uniquement les points les plus importants à une bonne compréhension d'un raisonnement flou.

### 5.2.2.1 Ensembles flous

#### 5.2.2.1.1 Ensembles classiques et flous

En logique floue, un ensemble flou est la généralisation de la notion de l'ensemble en logique classique (binaire).

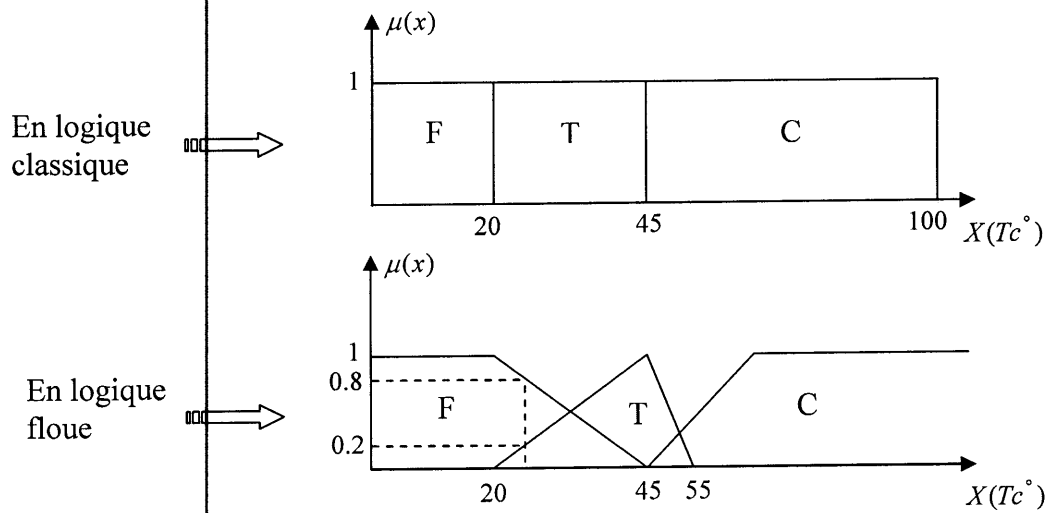
Mathématiquement, un ensemble flou  $F$  est défini sur un référentiel (un univers de discours)  $X$  et par une fonction d'appartenance  $\mu_F(x)$  qui prend des valeurs dans l'intervalle  $[0,1]$ . Il peut être représenté comme un ensemble de paires ordonnées [20] :

$$F = \{(x, \mu_F(x)) | x \in X\} \quad (5.1)$$

La fonction d'appartenance  $\mu_F(x)$  quantifie le degré d'appartenance de chaque élément de  $X$  à  $F$  :

$$\mu_F(x) = \begin{cases} \in ]0,1] & \text{Si } x \in F, \\ 0 & \text{sinon} \end{cases} \quad (5.2)$$

Prenons un exemple simple de la Figure 5.1 d'évaluation de la température de l'eau d'un récipient par les mots : *Froide* :  $F$ , *Tiède* :  $T$ , *Chaude* :  $C$ .



**Figure 5.1 :** Evaluation de la température selon la logique classique et floue.

De cette figure, nous pouvons voir que la logique classique ne peut utiliser que le 0 et le 1, ainsi l'eau est d'abord totalement froide puis tiède et enfin chaude. Alors qu'en logique floue, les ensembles flous sont définis par des fonctions d'appartenances ayant des valeurs *dans l'intervalle [0, 1]*.

Soit la température  $25\text{ c}^\circ$ . Dans le cas de la logique classique, cette température n'appartient pas à l'ensemble booléen  $F$ . Dans le cas de la logique floue, cette température appartient à l'ensemble flou  $F$  avec un degré d'appartenance 0.8.

La notion d'ensemble flou évite l'utilisation arbitraire de limites rigides d'appartenances à des classes ; il serait vraiment aberrant de considérer qu'une température de  $20\text{ c}^\circ$  est froide ( $F$ ), mais qu'une température de  $21\text{ c}^\circ$  ne l'est pas.

**Fonction d'appartenance :** Les fonctions d'appartenances utilisées pour décrire les ensembles flous possèdent quatre caractéristiques principales (Figure 5.7(a)) :

1) **Le type :** Elles peuvent être à titre d'exemple triangulaires, trapézoïdales, gaussiennes ou sigmoïdales, etc.

- **Fonction triangulaire :** Elle est définie par trois paramètres ( $a, b, c$ ) et donnée par :

$$\mu_F(x) = \max \left[ \min \left( \frac{x-a}{b-a}, \frac{c-x}{c-b} \right), 0 \right] \quad (5.3)$$

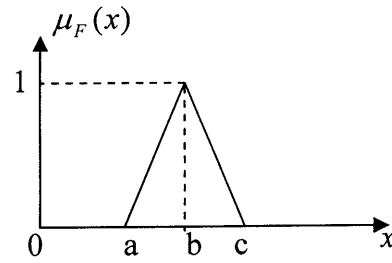


Figure 5.2 : Fonction triangulaire.

- **Fonction trapézoïdale** : Elle est définie par quatre paramètres (a,b,c,d) et donnée par

$$\mu_F(x) = \max \left[ \min \left( \frac{x-a}{b-a}, 1, \frac{d-x}{d-c}, 0 \right) \right] \quad (5.4)$$

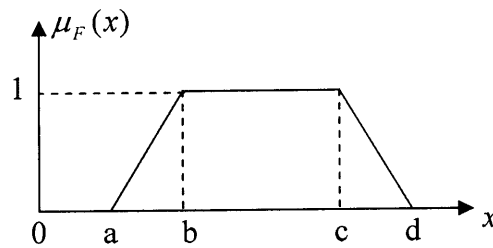


Figure 5.3 : Fonction trapézoïdale.

- **Fonction gaussienne** : Elle est définie par deux paramètres (σ, m) :

$$\mu_F(x) = \exp \left[ -\frac{(x-m)^2}{2\sigma^2} \right] \quad (5.5)$$

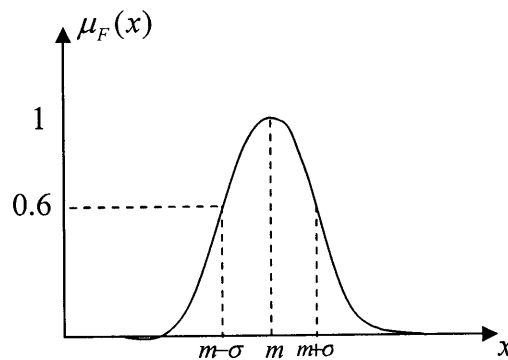


Figure 5.4 : Fonction gaussienne.

- **Fonction sigmoïde** : Elle est définie par deux paramètres (a, c)

$$\mu_F(x) = \frac{1}{1 + \exp(-a(x - c))} \quad (5.6)$$

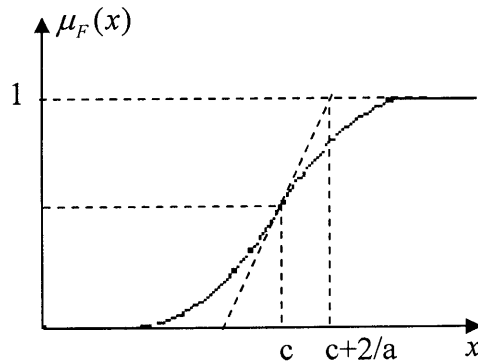


Figure 5.5 : Fonction sigmoïdale.

Il y a également deux ensembles flous spéciaux :

- *Un ensemble flou singleton* : il est équivalent à un nombre flou et défini par :

$$\mu_F(x) = \begin{cases} 1, & \text{si } x = x_0, \\ 0, & \text{ailleurs.} \end{cases} \quad (5.7)$$

La Figure 5.6 montre l'aspect de cette fonction d'appartenance.

- *Un ensemble flou universel* : il est défini par :

$$\mu_F(x) = 1, \quad \forall x \in X \quad (5.8)$$

i.e. tous les degrés d'appartenance sont égaux à 1.

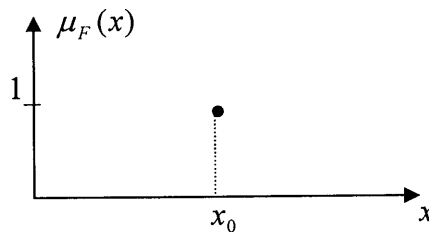


Figure 5.6 : Fonction singleton.

**2) Le support :** Il définit l'ensemble des éléments de  $X$  qui appartiennent au moins un peu à  $F$ . Il est noté  $S(F)$ . Autrement dit, c'est la partie de  $X$  sur laquelle le degré d'appartenance à  $F$  n'est pas nul :

$$S(F) = \{x \in X / \mu_F(x) \neq 0\} ; \quad (5.9)$$

**3) Le noyau :** L'ensemble d'éléments appartenant de façon absolue à  $F$  (i.e. ayant un degré 1) est appelé le noyau de  $F$ , noté par  $N(F)$  :

$$N(F) = \{x \in X / \mu_F(x) = 1\} ; \quad (5.10)$$

Si  $F$  est un ensemble classique sur  $X$ , donc il est normalisé et identique à son support et à son noyau.

**4) La hauteur :** notée  $H(F)$ , c'est le plus fort degré avec lequel un élément de  $X$  appartient à  $F$ . Autrement dit, c'est la valeur maximale prise par sa fonction d'appartenance :

$$H(F) = \sup_{x \in X} \mu_F(x) ; \quad (5.11)$$

On dit que  $F$  est normalisé si sa hauteur  $H(F)$  est égale à 1.

#### 5.2.2.1.2 Formalisme Linguistique

Dans le formalisme défini par Zadeh [19], les propriétés définies par les ensembles flous se nomment les termes (labels) linguistiques ou les termes flous (par exemple : *Froide, Tiède et Chaude*). Chaque terme représente un sous-ensemble de valeurs numériques et caractérise ainsi la variable floue (ou linguistique). Le domaine sur lequel ces termes et ces variables sont définis constitue *l'univers de discours*. Le découpage de cet univers de discours par les termes linguistiques est appelé *une partition floue*. Lorsque la somme des degrés d'appartenance sur la totalité de l'univers de discours est égale à 1, nous parlons alors de *partition floue forte* (Figure 5.7 (b)).



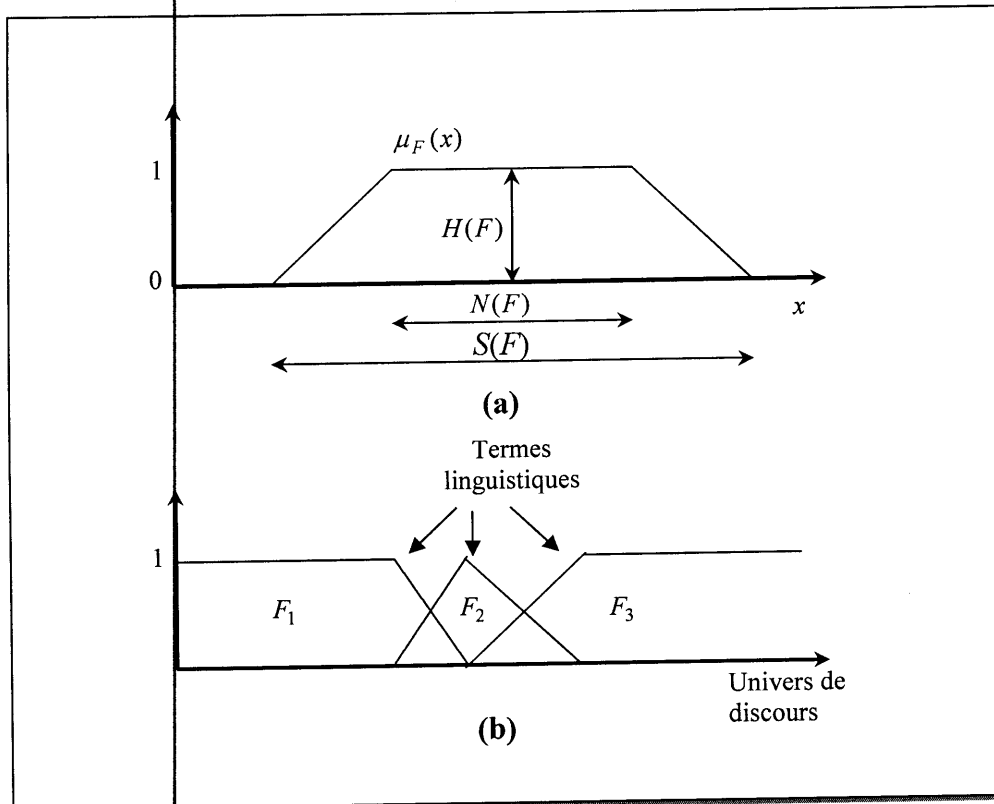


Figure 5.7 : Caractéristiques d'un ensemble flou et partition floue forte.

### 5.2.2.1.3 Opérations de base sur les ensembles flous

La théorie mathématique sur les ensembles flous définit de nombreux opérateurs sur ces ensembles et sur les fonctions d'appartenance.

Soient  $F_1$  et  $F_2$  deux ensembles flous définis sur l'ensemble  $X$  :

- **Complément** : Le complément de  $F_1$  noté  $\bar{F}_1$  est défini par :

$$\forall x \in X, \mu_{\bar{F}_1}(x) = 1 - \mu_{F_1}(x) ; \quad (5.12)$$

Un exemple est donné dans la Figure 5.8.

- **Intersection** : Elle est réalisée par n'importe quelle T-norme (norme triangulaire notée T). La plus usitée est l'opérateur min :

$$\forall x \in X, \mu_{F_1 \cap F_2}(x) = T(\mu_{F_1}(x), \mu_{F_2}(x)) = \min(\mu_{F_1}(x), \mu_{F_2}(x)); \quad (5.13)$$

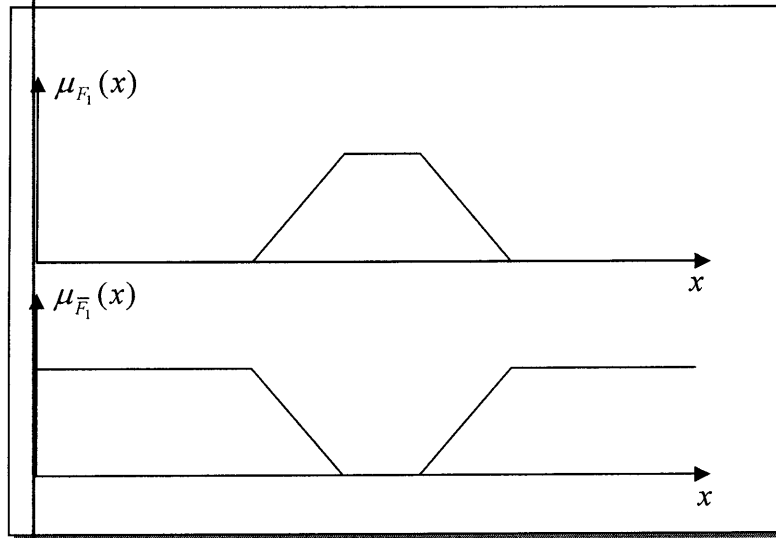


Figure 5.8 : Complément d'un ensemble flou.

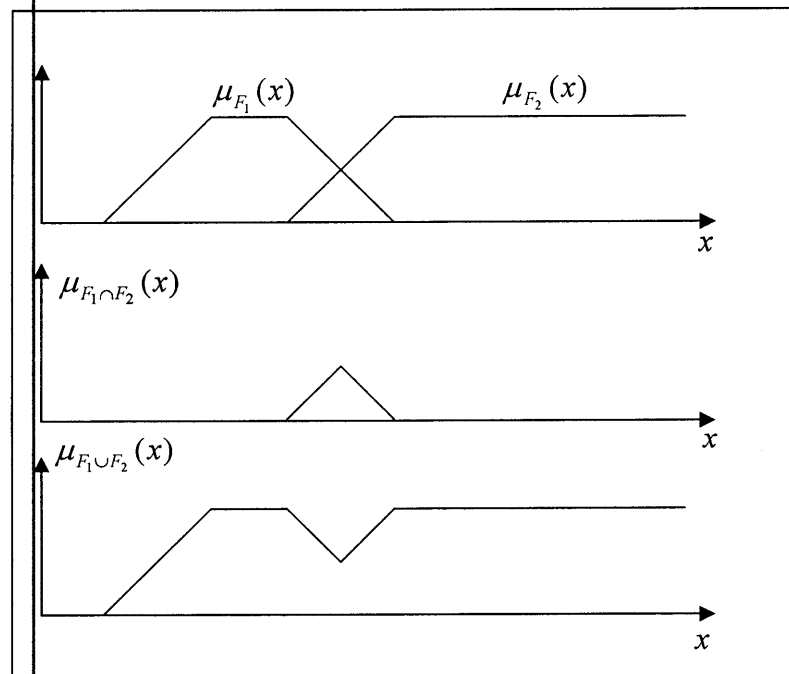


Figure 5.9 : Intersection et Union de deux ensembles flous.

- **Union** : Elle est réalisée par n'importe quelle T-conorme (conorme triangulaire notée  $\perp$ ). La plus usitée est l'opérateur max :

$$\forall x \in X, \mu_{F_1 \cup F_2}(x) = \perp(\mu_{F_1}(x), \mu_{F_2}(x)) = \max(\mu_{F_1}(x), \mu_{F_2}(x)) ; \tag{5.14}$$

L'exemple de la Figure 5.8 illustre les opérations de l'intersection et de l'union.

- **Cardinalité** : Elle évalue le degré globale avec lequel les éléments de  $X$  appartiennent à  $F_1$ . Elle est définie par :

$$|F_1| = \sum_{x \in X} \mu_F(x). \quad (5.15)$$

La cardinalité d'un ensemble classique est égale au nombre d'éléments qui le composent.

#### 5.2.2.1.4 Interprétation des ensembles flous

Il existe deux interprétations possibles des sous-ensembles flous. La première concerne le passage du numérique au symbolique et la seconde le passage du symbolique au numérique [20] :

- ❖ **Numérique  $\Rightarrow$  Symbolique (Fuzzification)** : Dans ce cas de figure, il s'agit du passage d'une valeur numérique à un terme linguistique. Soit  $x$  une variable numérique définie sur  $X$ ,  $F$  un terme flou et  $a$  le degré d'appartenance de  $x$  à  $F$  :

$$\mu_F(x) = \begin{cases} a > 0, & x \text{ vérifie en partie la propriété définie par } F; \\ a = 1, & \text{il est vrai que " } x \text{ est } F \text{";} \\ a = 0, & \text{il est faux que " } x \text{ est } F \text{".} \end{cases} \quad (5.16)$$

- ❖ **Symbolique  $\Rightarrow$  Numérique (Défuzzification)** : Il s'agit ici d'exprimer une connaissance vague, par le biais d'un terme linguistique, sur une échelle numérique.  $\mu_F(x)$  représente alors la distribution de possibilité attachée à la variable  $x$ ,  $\mu_F(x_0)$  représente alors le degré de possibilité de " $x$  est  $x_0$ " :

$$\mu_F(x_0) = a \begin{cases} a = 1, & \text{il est possible que } x \text{ soit } x_0; \\ a = 0, & \text{il est impossible que } x \text{ soit } x_0. \end{cases} \quad (5.17)$$

L'aspect graduel de la possibilité s'interprétant de la manière suivante : si  $\mu_F(x_0) > \mu_F(x_1)$ , alors la valeur  $x_0$  est préférée à  $x_1$ .

#### 5.2.2.2 Règles floues

Les règles floues, appelées aussi règles d'inférence, sont de la forme "**Si prémisses Alors conclusions**". Elles sont très proches du langage naturel. Nous avons, e.g.

**Si la vitesse est grande et le sol est sec alors le freinage est puissant ;**

Plus généralement, un système d'inférence floue est composé de  $N$  règles floues de type :

$R_i$  : **Si**  $x_1 \in X_1^i$  **et ... et**  $x_n \in X_n^i$  **alors**  $y$  est  $Y^i$

avec :

- “**Si**  $x_1 \in X_1^i$  **et ... et**  $x_n \in X_n^i$  ” la partie prémisse de la règle  $i$  ;
- “ $y$  est  $Y^i$ ” la conclusion (conséquence) associée ;
- $X_j^i$  une caractéristique floue de la variable  $x_j$  dont la fonction d'appartenance est notée  $\mu_{X_j^i}$ .

Les prémisses et les conclusions constituent des propositions floues exprimées par “ $x$  est  $X$ ” ou par une conjonction ou disjonction de prédicats de cette forme. Un système d'inférence flou est alors une collection de relations entre propositions floues, ayant comme particularité la déduction de conclusions même quand les prédicats en prémisse ne sont qu'en partie satisfaits. La règle d'inférence utilisée en logique floue est la généralisation du Modus Ponens (MP) qui prend en considération l'aspect graduel inhérent aux ensembles flous :

Règle floue  $R_i$  : si  $x$  est  $X^i$  alors  $y$  est  $Y^i$

$\mu_{X^i}$                        $\mu_{Y^i}$

Fait observé :  $x$  est  $X^i$

$\mu_{X^i}$

---

Conclusion conduite :  $y$  est  $Y^i$

$\mu_{Y^i}$

La conclusion de chaque règle  $i$  est obtenue en déterminant la fonction d'appartenance  $\mu_{Y^i}$ .

Ce calcul est réalisé à partir de la règle  $R_i$  et de  $\mu_{X^i}$ .

La conclusion finale est formée par l'agrégation des conclusions de chaque règle. Pour le calcul de cette conclusion, quatre étapes sont nécessaires :

1. calcul des propositions,
2. calcul des relations,
3. compositions des règles avec les faits observés, et
4. agrégations des conclusions des règles.

Une cinquième opération appelée la défuzzication est indispensable, si le résultat souhaité est non flou.

### 5.2.2.1 Calcul des propositions floues

Le calcul d'une proposition floue consiste à déterminer sa fonction caractéristique  $\mu_P$ . Lorsque la proposition est composée de conjonctions ou de disjonctions de plusieurs prédicats, il est indispensable de définir les opérateurs correspondants. Nous notons trois implémentations principales pour ces opérateurs logiques.

Soient  $X_1^i$  et  $X_2^i$  deux termes flous opérant sur les univers de discours  $X_1$  et  $X_2$  et qui composent la proposition  $P$  :

- ❖ la conjonction :  $P = "x_1 \text{ est } X_1^i \text{ ET } x_2 \text{ est } X_2^i"$  est définie par  $\forall x_1 \in X_1, \text{ et } x_2 \in X_2$  :

$$\mu_P = \text{ET}(X_1^i, X_2^i) = \begin{cases} \min(\mu_{X_1^i}(x_1), \mu_{X_2^i}(x_2)), & \text{Mamdani,} \\ \mu_{X_1^i}(x_1) \times \mu_{X_2^i}(x_2), & \text{Larsen,} \\ \max(0, \mu_{X_1^i}(x_1) + \mu_{X_2^i}(x_2) - 1), & \text{Lukasiewicz;} \end{cases} \quad (5.18)$$

- ❖ la disjonction :  $P = "x_1 \text{ est } X_1^i \text{ OU } x_2 \text{ est } X_2^i"$  est définie par  $\forall x_1 \in X_1, \text{ et } \forall x_2 \in X_2$  :

$$\mu_P = \text{OU}(X_1^i, X_2^i) = \begin{cases} \max(\mu_{X_1^i}(x_1), \mu_{X_2^i}(x_2)), & \text{Mamdani,} \\ \mu_{X_1^i}(x_1) + \mu_{X_2^i}(x_2) - (\mu_{X_1^i}(x_1) \cdot \mu_{X_2^i}(x_2)), & \text{Larsen,} \\ \min(1, \mu_{X_1^i}(x_1) + \mu_{X_2^i}(x_2)) & \text{Lukasiewicz.} \end{cases} \quad (5.19)$$

Après le calcul de toutes les propositions floues, l'étape suivante consiste à calculer la contribution de chaque règle afin de pouvoir, dans un troisième temps, composer la proposition correspondante aux faits observés avec chaque règle.

### 5.2.2.2 Calcul des relations

Une règle floue  $R$  définit une relation entre  $X$  et  $Y$ . Cette relation constitue une distribution conditionnelle qui associe à chaque  $x \in X$  l'ensemble flou  $R(x)$  des images

possibles de  $x$  dans  $Y$  et est caractérisée par sa fonction d'appartenance  $\mu_{R(x,y)}$ . Elle peut être modélisée selon trois classes de fonctions d'implication [20] :

$$\mu_{R(x,y)} = I(\mu_X(x), \mu_Y(y)) \text{ avec } I : [0,1] \times [0,1] \rightarrow [0,1] ; \tag{5.20}$$

❖ **S-implication** :  $I_S(u, v) = S(n(u), v)$  où  $S$  est une T-conorme et  $n$  une fonction de négation définie sur  $[0,1]$  ( $n(u) = 1 - u$  par exemple). Nous pouvons citer trois exemples pour une telle implication :

$$I_S(u, v) = \begin{cases} \min(1, 1 - u + v), & \text{Lukasiewicz,} \\ \max(1 - u, v), & \text{Kleen - Dienes,} \\ 1 - u + uv, & \text{Reinchenbach.} \end{cases} \tag{5.21}$$

❖ **R-implication**: Ces implications proviennent du formalisme de la logique intuitionniste. Nous notons par exemple :

$$I_R(u, v) = \begin{cases} \min(1, 1 - u + v), & \text{Lukasiewicz,} \\ 1 \text{ si } u \leq v \text{ et } u \text{ sinon} & \text{Godel - Brouwer,} \\ 1 \text{ si } u \leq v \text{ et } u/v \text{ sinon} & \text{Menger - Goguen.} \end{cases} \tag{5.22}$$

❖ **Implication T-norme** : Bien que ce modèle ne vérifie pas toutes les propriétés de l'implication, il est possible de l'utiliser tout en prenant garde au choix de l'opérateur d'agrégation (paragraphe 5.2.2.2.4):

$$I_T(u, v) = \begin{cases} \min(u, v), & \text{Mamdani,} \\ u \times v, & \text{Menger - Goguen.} \end{cases} \tag{5.23}$$

### 5.2.2.2.3 Inférence floue

L'inférence floue permet de combiner la connaissance que nous avons de la règle  $R$ , par exemple  $\mu_R(x, y), \forall x \in X, \forall y \in Y$ , avec la donnée concernant la variable  $x$ . Cette composition définie par le MPG (Modus Ponens Généralisé, cf. paragraphe 5.2.2.2) est la règle compositionnelle d'inférence sup-T, où  $T$  est la norme triangulaire. La fonction d'appartenance ainsi inférée est donnée par :

$$\forall y \in Y, \mu_{y'}(y) = \sup_{x \in X} T(\mu_{X'}(x), \mu_R(x, y)). \tag{5.24}$$

La norme triangulaire doit être choisie de telle sorte que le MPG soit compatible avec le Modus Ponens classique i.e.  $\mu_{Y^i} = \mu_{Y^i}$  si  $\mu_{X^i} = \mu_{X^i}$ . Trois opérateurs sont classiquement choisis :

$$T(u, v) = \begin{cases} \max(u + v - 1, 0); \\ \min(u, v) \\ u \cdot v \end{cases} \quad (5.25)$$

Après avoir déduit les conclusions des règles du système d'inférence floue, l'étape suivante consiste à les agréger pour calculer la fonction d'appartenance finale.

#### 5.2.2.2.4 Agrégation des règles

Après le calcul des conclusions intermédiaires fournies par chaque règle de la base du système d'inférence floue, la conclusion finale est le résultat de la combinaison de toutes ces conséquences locales. Elle peut être effectuée de trois manières :

**1- Par intersection :** l'ensemble flou final  $Y'$  est défini comme étant l'intersection de tous les ensembles inférés, soit :

$$\forall y \in Y, \mu_{Y'}(y) = T(\mu_{Y^1}(y), \dots, \mu_{Y^N}(y)), \quad (5.26)$$

où  $T$  est généralement la T-norme min. Cette méthode ne peut pas être appliquée conjointement avec les implications T-norme (cf. paragraphe 5.2.2.2.2).

**2- Par union :** l'ensemble flou final  $Y'$  est issu de l'union des ensembles flous locaux, soit :

$$\forall y \in Y, \mu_{Y'}(y) = \perp (\mu_{Y^1}(y), \dots, \mu_{Y^N}(y)), \quad (5.27)$$

où  $\perp$  est généralement la T-conorme max.

Cette implémentation est particulièrement adaptée au choix des implications T-norme (paragraphe 5.2.2.2.2).

**3- Par certaines méthodes de défuzzification** [20], ou par le calcul du barycentre des conclusions inférées (quand celles-ci sont de nature précise, comme dans le cas du système flou de Takagi-Sugeno, cf. paragraphe 5.2.3.2).

### 5.2.2.2.5 Défuzzification

Le résultat de l'agrégation est généralement un ensemble flou. Or, si l'on désire une sortie non floue, il faut transformer cette partie floue issue de l'inférence en une valeur physique. Cette opération est appelée défuzzification. Plusieurs stratégies de défuzzification existent. Les plus répandues sont :

- la méthode du centre de gravité,
- la méthode du maximum,
- la méthode de la hauteur, et
- la méthode de l'aire, ...etc.

### 5.2.3 Différents types de modèle

Un modèle flou (système flou) est une représentation du comportement du système par les ensembles flous. Il peut remplacer un système d'équations différentielles ou une fonction non linéaire analytique utilisés dans la modélisation classique. D'un point de vue structurel, les modèles flous peuvent être classés en deux classes essentielles :

- les modèles linguistiques (les modèles de Mamdani),
- les modèles à conséquences fonctionnelles (les modèles de Takagi-Sugeno (TS)).

#### 5.2.3.1 Modèles de Mamdani (linguistiques)

Le système flou de Mamdani se caractérise par le choix de l'opérateur min, que cela soit pour la conjonction dans le calcul des propositions ou pour l'implication T-norme (cf. paragraphe 5.2.2.2.2).

Soit un système flou de Mamdani d'une base de  $N$  règles et  $n$  variables linguistiques. Les règles correspondantes sont de cette forme :

$$\text{Si } x_1 \text{ est } X_1^i \text{ et } \dots \text{ et } x_n \text{ est } X_n^i, \text{ alors } y = B^i \text{ ou } i = 1, \dots, N \quad (5.28)$$

où  $B_i$  est un sous-ensemble flou. En général, les  $B_i$  forment une partition de l'espace de sortie.

En utilisant le minimum comme opérateur de conjonction et d'implication, l'inférence floue correspond aux étapes suivantes :



1- calcul du degré d'appartenance de chaque entrée aux différents sous-ensembles,  $\mu_{x_j^i}$ , pour  $j = 1, \dots, n$  et  $i = 1, \dots, N$ .

2- calcul de la valeur d'activation de chaque règle, pour  $i = 1, \dots, N$  :

$$\alpha_i(x) = \min_j(\mu_{x_j^i}(x_j)), \quad j = 1, \dots, n ; \quad (5.29)$$

3- calcul de la contribution de chaque règle, pour  $i = 1, \dots, N$  :

$$\mu_i(y) = \min(\alpha_i(x), \mu_{B^i}(y)) ; \quad (5.30)$$

4-agrégation des règles :

$$\mu(y) = \max_i(\mu_i(y)) ; \quad (5.31)$$

Le résultat de l'agrégation est donc un ensemble flou caractérisé par sa fonction d'appartenance. Pour l'obtention d'une conclusion de nature précise  $y_0$  à partir d'un ensemble flou résultant de l'agrégation, il faut *défuzzifier*. La méthode du centre de gravité donne alors :

$$y_0 = \frac{\int x\mu(x)dx}{\int \mu(x)dx} ; \quad (5.32)$$

En discrétisant l'espace de sortie, la formule précédente devient :

$$y_0 = \frac{\sum_k x_k \mu(x_k)}{\sum_k \mu(x_k)} ; \quad (5.33)$$

Cette implémentation est parfois appelée *min-max, barycentre*, du nom des opérateurs utilisés dans le calcul de la contribution de chaque règle, l'agrégation et la défuzzification. Il existe plusieurs variantes de la méthode de Mamdani. On trouve aussi, par exemple, des variantes de type:

- min-produit, barycentre,
- Produit-somme, barycentre.

D'autres portent sur les méthodes de défuzzification : méthode des maxima ou des hauteurs. Il faut noter que beaucoup de méthodes d'optimisation ou d'adaptation des paramètres actuelles, en particulier celles qui exigent la dérivabilité du système flou, rendent caduques ces différentes variantes ou au moins en réduisant leur importance.

Ces modèles flous ont les avantages suivants [21], [22] :

- ❖ Les prémisses et les conclusions des règles sont de nature entièrement linguistique. Ceci les rend facilement compréhensibles, et donne une interprétation sémantique claire au modèle.
- ❖ Il est tout à fait possible de combiner les règles identifiées à partir des données numériques par une méthode d'apprentissage, avec des règles formulées par un expert.
- ❖ L'interpolation dépend de la forme des fonctions d'appartenance, des opérateurs d'inférence et de la méthode de défuzzification. Elle peut être améliorée par un choix adéquat de ces éléments. Donc, l'interpolation dans ces modèles peut être quantifiée et contrôlée.

Malgré ses avantages, ce type de modèle flou (système flou) a les limitations suivantes [21], [22] :

- ❖ Leur nature non structurée se traduit par une capacité de représentation limitée. En d'autres termes un grand nombre de règles est souvent nécessaire pour approcher le comportement global d'un système. Par conséquent, le nombre de paramètres à identifier est élevé.
- ❖ Le grand nombre de paramètres exige une base d'information importante et complète (dans le cas de l'apprentissage hors-ligne) pour déterminer tous les paramètres.

### 5.2.3.2 Modèles de TS

Les modèles linguistiques ne permettent pas d'intégrer directement des connaissances objectives sur le système qui ne sont pas exprimées par des ensembles flous. Ainsi, des connaissances exprimées sous forme analytique décrivant la structure physique du système, par exemple les équations de conservation d'énergie ou du bilan de masses ne peuvent pas être prises en compte directement. Sugeno et ses collaborateurs [23], [24] ont proposé une classe de modèles permettant de représenter ce type d'information. Les modèles de cette classe sont appelés *les modèles de Takagi-Sugeno (TS)*.

La base de  $N$  règles pour un vecteur d'entrée  $x$  de dimension  $n$  est de la forme :

$$\text{Si } x_1 \text{ est } X_1^i \text{ et } \dots \text{ et } x_n \text{ est } X_n^i \text{ alors } y^i = F_i(x), \quad i = 1, \dots, N \quad (5.34)$$

où  $F_i(x)$  peut être une fonction analytique non linéaire quelconque.

Si  $F_i(x)$  est une fonction linéaire, dans ce cas le modèle est de type TS d'ordre 1 (TS1) :

$$F_i(x) = \sum_{j=1}^n a_j^i x_j + b^i \quad (1.35)$$

Lorsque  $F_i(x) = b^i$  le modèle est de type TS d'ordre zéro (TS0).

Chacune des règles représente un modèle local sur une région floue d'entrée ou sur un sous-espace d'entrée. Dans chaque région, le modèle flou est défini par la fonction  $F_i$  qui relie les entrées à la sortie numérique. Le modèle global est constitué par interpolation des modèles locaux (Figure 5.10).

L'inférence floue, dans ce modèle, est composée de trois étapes :

1- calcul des degrés d'activation de chaque règle avec l'opération de conjonction implémentée par le produit :

$$w_i(x) = \prod_{j=1}^n \mu_{x_j^i}(x_j), (i = 1, \dots, N), \quad (i=1, \dots, N) ; \quad (5.36)$$

2- calcul des sorties individuelles : la sortie de chaque règle est donnée par sa conséquence, c'est à dire :

$$y^i = F_i(x) ; \quad (5.37)$$

3- agrégation des sorties individuelles : la valeur finale de la sortie résultante de l'ensemble des règles est donnée par la moyenne des sorties individuelles pondérées par le degré d'activation des règles, soit :

$$y = \frac{\sum_{i=1}^N w_i(x) \cdot F_i(x)}{\sum_{i=1}^N w_i(x)} \quad (5.38)$$

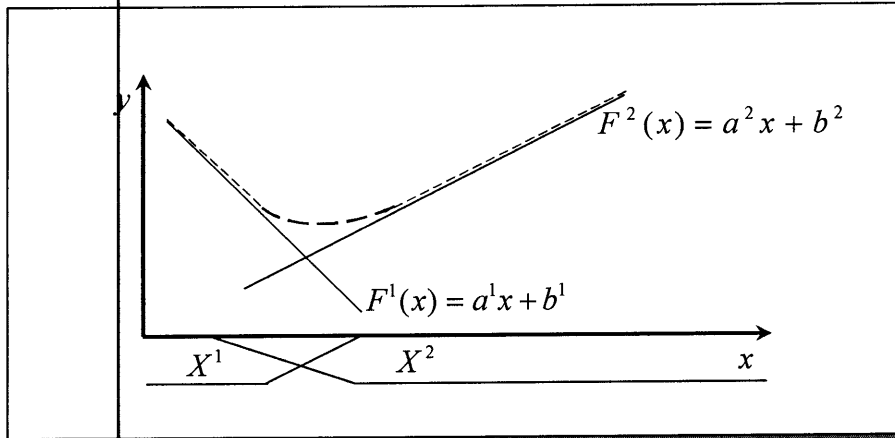
L'expression (5.38) peut être reformulée comme suit

$$y = \sum_{i=1}^N v_i \cdot F_i(x) \quad (5.39)$$

où  $v_i = \frac{w_i}{\sum_{i=1}^N w_i}$ . Notons que  $\sum_{i=1}^N v_i = 1$ .

Les avantages intrinsèques au système flou de TS1 sont les suivants [21],[ 22] :

- ❖ Ils permettent de décrire des situations où la structure physique du système est bien connue, par exemple, une fonction linéaire exprimant la relation entrée-sortie.
- ❖ Ils permettent d'approcher le comportement d'un système complexe par une collection des modèles locaux linéaires.



**Figure 5.10 :** *Interprétation dans un système de TS1.*

- ❖ Les modèles TS ont une capacité de représentation importante. Le nombre de règles nécessaires pour approcher un système à certain degré de précision est généralement plus faible que dans le cas des modèles linguistiques. L'identification du modèle à partir de données (dans le cas hors ligne) est facile et nécessite moins de données que pour les modèles linguistiques.
- ❖ La relation (5.38) relie directement la variable de sortie aux variables d'entrée. Ceci permet une application facile des techniques d'optimisation et d'adaptation pour identifier les différents paramètres du modèle.

Toutefois, ces modèles ont aussi des inconvénients :

- ❖ De part la nature fonctionnelle des conclusions, les règles n'ont pas de signification sémantique claire comme dans le cas des modèles linguistiques.
- ❖ L'interpolation entre les modèles locaux peut engendrer une erreur importante qui réduit la précision globale du modèle.
- ❖ L'intégration de la connaissance humaine formulée par des règles linguistiques n'est pas aisée.

Tout au long de cette thèse, comme la majorité des travaux relatifs à la commande adaptative floue, c'est le système flou de TS0 qui sera utilisé pour estimer la commande idéale ou les fonctions non linéaires inconnues.

En introduisant la notion des fonctions floues de base [15], la sortie du système flou de TS0 peut être écrite comme suit :

$$y(x) = \theta^T \psi(x) \quad (5.40)$$

avec  $\theta = [b^1, \dots, b^N]^T$  est le vecteur des paramètres de la partie conclusion, et  $\psi(x) = [v_1(x), \dots, v_N(x)]^T$  est le vecteur des fonctions floues de base, dont chacune des composantes est donnée par :

$$v_i = \frac{w_i(x)}{\sum_{i=1}^N w_i(x)} \quad (5.41)$$

Il est important de noter que si nous choisissons une partition floue forte pour les variables d'entrée, le vecteur des fonctions floues de base se réduit à  $v_i = w_i(x)$ . Dans le cas où la partition floue forte n'est pas choisie, nous devons assurer tout le temps que  $\sum_{i=1}^N w_i(x) \neq 0$ , i.e. que le système flou est toujours bien défini.

#### 5.2.4 Théorème des approximateurs universels

Deux raisons principales amènent à incorporer les systèmes flous dans un contrôleur adaptatif :

- Ils ont la propriété d'approximation universelle,
- La possibilité d'intégrer dans les contrôleurs des informations linguistiques ou mathématiques disponibles issues d'une expertise.

Dans la littérature floue, on dispose d'un nombre important de travaux montrant que les systèmes flous sont bien des approximateurs universels [15].

Dans [15], Wang a montré que les systèmes flous sont des approximateurs universels s'ils respectent certains critères. Il voit son résultat comme un théorème d'existence d'un système flou optimal pour une large variété d'applications.

La commande floue proposée dans cette section est basée sur le théorème suivant dit *théorème d'approximateurs universels*.

**Théorème 5.1 :** Soit  $f(x)$  une fonction non linéaire continue définie sur un ensemble compact  $\Omega_x$ , et pour toute constante positive  $\varepsilon$ , il existe un système flou  $y(x)$  de la forme (5.40) tel que :

$$\text{Sup}_{x \in \Omega_x} |f(x) - \theta^T \psi(x)| < \varepsilon \quad (5.42)$$

La démonstration de ce théorème est donnée dans [15].

Dans cette thèse, nous supposons que la structure du système flou et les fonctions floues de base sont convenablement spécifiées à priori par l'utilisateur. Cela signifie que la décision de l'utilisateur est nécessaire pour déterminer la structure du système flou à savoir : les entrées pertinentes, le nombre de fonctions d'appartenance pour chaque entrée, les paramètres des fonctions d'appartenance et le nombre de règles. Quant aux paramètres des conclusions, i.e.  $\theta$ , ils seront déterminés en ligne via un algorithme adéquat d'adaptation.

### 5.3 Commande floue adaptive

Considérons la classe suivante des systèmes non linéaires multi variables décrite par:

$$\begin{aligned} y_1^{(r_1)} &= f_1(x) + \sum_{j=1}^p g_{1j}(x)u_j + d_1(t, x), \\ &\vdots \\ y_p^{(r_p)} &= f_p(x) + \sum_{j=1}^p g_{pj}(x)u_j + d_p(t, x), \end{aligned} \quad (5.43)$$

où  $x = [y_1, \dot{y}_1, \dots, y_1^{(r_1-1)}, \dots, y_p, \dot{y}_p, \dots, y_p^{(r_p-1)}]^T \in R^r$  est le vecteur d'état complet qui est supposé disponible pour la mesure et  $r_1 + \dots + r_p = r$ ,  $u = [u_1, \dots, u_p]^T \in R^p$  est le vecteur d'entrée de commande,  $y = [y_1, \dots, y_p]^T \in R^p$  est le vecteur de sortie, et  $f_i(x), i=1, \dots, p$  sont des fonctions non linéaires continues inconnues, et  $g_{ij}(x), i, j=1, \dots, p$  sont des fonctions non linéaires continues inconnues de classe  $C^1$ .  $d_i(t, x)$ , pour  $i=1, \dots, p$ , sont des perturbations externes bornées.

Notons

$$y^{(r)} = [y_1^{(r_1)} \dots y_p^{(r_p)}]^T, \quad F(x) = [f_1(x) \dots f_p(x)]^T, \quad G(x) = [g_{ij}(x)], \quad \text{et} \quad D(t, x) = [d_1(t, x) \dots d_p(t, x)]^T.$$

Donc, le système (5.43) peut être réécrit sous la forme compacte suivante

$$y^{(r)} = F(x) + G(x)u + D(t, x) \quad (5.44)$$

où  $F(\cdot) \in R^p$ , et  $G(\cdot) \in R^{p \times p}$ .

Notre objectif consiste à concevoir une loi de commande  $u$  qui permet de forcer les sorties  $y_1, \dots, y_p$  à suivre les trajectoires de références prédéterminées  $y_{d1}, \dots, y_{dp}$ , tout en gardant la bornitude de tous les signaux du système en boucle fermée.

### Hypothèse 5.1 :

1) Le vecteur d'état  $x$  est supposé disponible pour la mesure.

2) Le vecteur des trajectoires désirées,  $x_d = [y_{d1}, \dot{y}_{d1}, \dots, y_{d1}^{(r_1)}, \dots, y_{dp}, \dot{y}_{dp}, \dots, y_{dp}^{(r_p)}]^T$ , est supposé être continu, borné et disponible pour la mesure. Donc,  $x_d \in \Omega_{x_d} \subset R^{r+p}$ , avec  $\Omega_{x_d}$  est un ensemble compact connu.

Définissons les erreurs de poursuite comme suit :

$$\begin{aligned} e_1 &= y_{d1} - y_1 \\ &\vdots \\ e_p &= y_{dp} - y_p \end{aligned} \quad (5.45)$$

et le vecteur des erreurs filtrées comme

$$S = [S_1, \dots, S_p]^T \quad (5.46)$$

Avec

$$S_i = \left[ \frac{d}{dt} + \lambda_i \right]^{r_i-1} e_i, \text{ pour } \lambda_i > 0, \forall i = 1, \dots, p. \quad (5.47)$$

Ou encore, (5.47) peut s'écrire comme suit

$$S_i = \lambda_i^{r_i-1} e_i + (r_i - 1) \lambda_i^{r_i-2} \dot{e}_i + \dots + (r_i - 1) \lambda_i e_i^{(r_i-2)} + e_i^{(r_i-1)}, \quad (5.48)$$

Avec  $i = 1, \dots, p$ .

Notons que si nous choisissons  $\lambda_i > 0$ , avec  $i = 1, \dots, p$ , alors les zéros du polynôme

$H_i(s) = \lambda_i^{r_i-1} + (r_i - 1) \lambda_i^{r_i-2} s + \dots + (r_i - 1) \lambda_i s^{r_i-2} + s^{r_i-1}$  correspondant à l'équation caractéristique de  $S_i = 0$  sont tous dans le demi-plan gauche ouvert.

La relation (5.48) peut être écrite sous la forme compacte suivante

$$S_i = C_i^T E_i \quad (5.49)$$

avec

$$E_i = [e_i \dot{e}_i \dots e_i^{(r_i-2)} e_i^{(r_i-1)}]^T \quad (5.50)$$

$$C_i^T = [\lambda_i^{r_i-1} (r_i-1)\lambda_i^{r_i-2} \dots (r_i-1)\lambda_i \ 1] \quad (5.51)$$

Par conséquent, le vecteur  $S$  prend la forme suivante :

$$S = C^T E \quad (5.52)$$

où

$$C^T = \text{diag}[C_1^T C_2^T \dots C_p^T]_{(p \times r)} \quad (5.53)$$

$$E = [E_1^T E_2^T \dots E_p^T]^T_{(r \times 1)} \quad (5.54)$$

et la dynamique  $S_i$  est décrite par :

$$\dot{S}_i = C_{r_i}^T E_i + e_i^{(r_i)}, \text{ et } i = 1, \dots, p \quad (5.55)$$

où

$$C_{r_i}^T = [0 \ \lambda_i^{r_i-1} \ (r_i-1)\lambda_i^{r_i-2} \dots \ 0.5(r_i-1)(r_i-2)\lambda_i^2 \ (r_i-1)\lambda_i] \quad (5.56)$$

La dynamique de  $S$  peut s'écrire sous la forme compacte suivante

$$\dot{S} = C_r^T E + e^{(r)} \quad (5.57)$$

où

$$C_r^T = \text{diag}[C_{r_1}^T C_{r_2}^T \dots C_{r_p}^T]_{(p \times r)} \quad (5.58)$$

$$e^{(r)} = [e_1^{(r_1)} \ e_2^{(r_2)} \ \dots \ e_p^{(r_p)}]^T \quad (5.59)$$

et

$$e^{(r)} = y_d^{(r)} - y^{(r)} \quad (5.60)$$

où  $y^{(r)} = [y_1^{(r_1)} \ y_2^{(r_2)} \ \dots \ y_p^{(r_p)}]^T$  est précédemment défini, et

$$y_d^{(r)} = [y_{d1}^{(r_1)} \ y_{d2}^{(r_2)} \ \dots \ y_{dp}^{(r_p)}]^T, \quad (5.61)$$

De (5.44) et (5.60), nous pouvons écrire (4.57) comme suit

$$\dot{S} = C_r^T E + y_d^{(r)} - y^{(r)} \quad (5.62)$$

Par la suite, (5.62) sera employé dans le développement des contrôleurs flous adaptatifs et l'analyse de la stabilité.



### ❖ Conception de la commande floue adaptative

Dans cette section, nous développons une commande adaptative floue pour le système non linéaire (5.62) dont la matrice de commande est supposée connue.

Par la suite, l'hypothèse suivante est nécessaire.

#### Hypothèse 5.2

a)  $G(x)$  est une matrice non-singulière de fonctions connues et

$$\text{b) } \|d_i(t, x)\| \leq \bar{d}_i, \quad (5.63)$$

où  $\bar{d}_i$  est une borne connue.

Notons

$$\alpha(x) = [\alpha_1(x), \dots, \alpha_p(x)]^T = C_r^T E + y_d^{(r)} - F(x) \quad (5.64)$$

En utilisant (5.64), (5.62) devient:

$$\dot{S} = \alpha(x) - G(x)u - D(t, x) \quad (5.65)$$

Par la suite, l'approximation floue de  $\alpha(x)$  sera utilisée pour développer un contrôleur adaptatif avec une loi d'adaptation permettant de répondre aux objectifs de commande et garantir la bornitude de tous les signaux du système en boucle fermée.

La non-linéarité  $\alpha_i(x)$  peut être approchée, sur un ensemble  $D_x$ , par les systèmes flous (5.40) comme suit :

$$\hat{\alpha}_i(x, \theta_i) = \theta_i^T \psi_i(x), \quad i = 1, \dots, p, \quad (5.66)$$

où  $\psi_i(\bar{x})$  est le vecteur des fonctions floues de base étant supposé convenablement choisi a priori, et  $\theta_i$  est le vecteur des paramètres ajustés du système flou.

Définissons

$$\theta_i^* = \arg \min_{\theta_i} \left[ \sup_{x \in D_x} |\alpha_i(x) - \hat{\alpha}_i(x, \theta_i)| \right] \quad (5.67)$$

comme un vecteur des paramètres optimaux de  $\theta_i$ .

Noter que  $\theta_i^*$  est une quantité constante artificielle introduite seulement pour un but d'analyse, et la connaissance de sa valeur n'est pas nécessaire en mettant en application le contrôleur.

Dénotons par  $\tilde{\theta}_i = \theta_i - \theta_i^*$ , avec  $i = 1, \dots, p$  les erreurs d'estimation paramétrique, et

$$\varepsilon_i(x) = \alpha_i(x) - \hat{\alpha}_i(x, \theta_i^*) \quad (5.68)$$

les erreurs d'approximation floue, où  $\hat{\alpha}_i(x, \theta_i^*) = \theta_i^{*T} \psi_i(x)$ .

L'erreur d'approximation floue est supposée être bornée pour tout  $x \in D_x$ , i.e.  $|\varepsilon_i(x)| \leq \bar{\varepsilon}_i$ ,  $\forall x \in D_x$ , où  $\bar{\varepsilon}_i$  est une constante positive inconnue.

Maintenant, Notons

$$\hat{\alpha}(x, \theta) = \theta^T \psi(x) = [\hat{\alpha}_1(x, \theta_1) \dots \hat{\alpha}_p(x, \theta_p)]^T, \quad \varepsilon(x) = [\varepsilon_1(x) \dots \varepsilon_p(x)]^T, \quad \bar{\varepsilon} = [\bar{\varepsilon}_1 \dots \bar{\varepsilon}_p]^T.$$

De l'analyse ci-dessus, nous avons

$$\begin{aligned} \hat{\alpha}(x, \theta) - \alpha(x) &= \hat{\alpha}(x, \theta) - \hat{\alpha}(x, \theta^*) + \hat{\alpha}(x, \theta^*) - \alpha(x), \\ &= \hat{\alpha}(x, \theta) - \hat{\alpha}(x, \theta^*) - \varepsilon(x), \\ &= \tilde{\theta}^T \psi(x) - \varepsilon(x). \end{aligned} \quad (5.69)$$

où  $\tilde{\theta}^T \psi(x) = [\tilde{\theta}_1^T \psi_1(x), \dots, \tilde{\theta}_p^T \psi_p(x)]^T$ , et  $\tilde{\theta}_i = \theta_i - \theta_i^*$ ,  $i = 1, \dots, p$ .

En utilisant (4.31), la loi de commande proposée est:

$$u = G^{-1}(x)(\hat{\alpha}(x, \theta) + K_1 S + K_0 \text{Sign}(S)) = G^{-1}(x)(\theta^T \psi(x) + K_1 S + K_0 \text{Sign}(S)) \quad (5.70)$$

où  $\hat{\alpha}(x, \theta) = \theta^T \psi(x) = [\theta_1^T \psi_1(x), \dots, \theta_p^T \psi_p(x)]^T$ , et  $K_0 = \text{Diag}[k_{01}, \dots, k_{0p}]$ , avec  $k_{0i} > 0, i = 1, \dots, p$ , sont des constantes positives qui seront par la suite définies.  $K_1 = \text{Diag}[k_{11}, \dots, k_{1p}]$ , avec  $k_{1i} > 0, i = 1, \dots, p$ , sont des constantes libres de conception.

En appliquant le contrôleur (5.70) au système (5.43) (i.e. En remplaçant (5.70) dans la dynamique de l'erreur filtrée de poursuite (5.65)), et en utilisant l'expression (5.69), nous obtenons

$$\dot{S} = -K_1 S - K_0 \text{Sign}(S) - \tilde{\theta}^T \psi(x) + \varepsilon(x) - D(t, x) \quad (5.71)$$

En multipliant (5.71) par  $S^T$ , il résulte

$$S^T \dot{S} = -S^T K_1 S - K_0 |S| - \sum_{i=1}^p \tilde{\theta}_i^T \psi_i(x) S_i + S^T \varepsilon(x) - S^T D(t, x) \quad (5.72)$$

L'estimation des paramètres  $\theta_i$  suit la loi intégrale suivante

$$\theta_i = \int [-\sigma_i \gamma_i |S_i| \theta_i + \gamma_i S_i \psi_i(\bar{x})] dt, \quad (5.73)$$

où  $\gamma_i, \sigma_i > 0$  sont des constantes de conception. Le terme  $\sigma_i \gamma_i |S_i| \theta_i$  dans (5.73), dit *e-modification*, est introduit afin d'assurer aussi bien la bornitude des paramètres flous ajustés que la convergence des erreurs de poursuite vers zéro.

Il est aussi important de noter que le terme de commande  $K_1 S$  dans (5.73) est un terme linéaire. Quant au terme  $K_0 \text{Sign}(S)$ , c'est un terme non linéaire qui est utilisé pour compenser les inévitables erreurs d'approximation floue et les éventuelles perturbations externes bornées et aussi éliminer l'effet du terme  $0.5 \sigma_i \|\theta_i^*\|^2$ ,  $i = 1, \dots, p$ , dû à l'utilisation de *e-modification* dans la loi d'adaptation (5.73).

Maintenant, on peut énoncer le résultat fondamental suivant :

**Théorème 4.2 :** *Considérons le système (5.43). Supposons que les Hypothèses 5.1-5.2 sont satisfaites. Alors, la loi de commande définie par (5.70) avec sa loi d'adaptation (5.73) garantit les propriétés suivantes :*

- *Tous les signaux du système en boucle fermée sont bornés.*
- *Les erreurs de poursuites et leurs dérivées convergent vers zéros, i.e.  $e_i, e_i^{(j)}(t) \rightarrow 0$  quand  $t \rightarrow \infty$  pour  $i = 1, \dots, p, j = 0, 1, \dots, r_i - 1$ . quand  $t \rightarrow \infty$  pour  $i = 1, \dots, p$  et  $j = 0, 1, \dots, r_i - 1$ .*

#### Preuve du Théorème 5.1 :

Considérons la fonction de Lyapunov candidate suivante

$$V = \frac{1}{2} S^T S + \frac{1}{2} \sum_{i=1}^p \frac{1}{\gamma_{0i}} \tilde{\theta}_i^T \tilde{\theta}_i \quad (5.74)$$

Sa dérivée temporelle est donnée par

$$\dot{V} = S^T \dot{S} + \sum_{i=1}^p \frac{1}{\gamma_{0i}} \tilde{\theta}_i^T \dot{\theta}_i \quad (5.75)$$

En utilisant (5.72) et les Hypothèses 5.2,  $\dot{V}$  peut être bornée comme suit

$$\dot{V} \leq -S^T K_1 S - \sum_{i=1}^p k_{0i} |S_i| - \sum_{i=1}^p \tilde{\theta}_i^T \psi_i(x) S_i + S^T \varepsilon(\bar{x}) - S^T D(t, x) + \sum_{i=1}^p \frac{1}{\gamma_{0i}} \tilde{\theta}_i^T \dot{\theta}_i$$

$$\leq -S^T K_1 S - \sum_{i=1}^p k_{0i} |S_i| - \sum_{i=1}^p \tilde{\theta}_i^T \psi_i(x) S_i + \sum_{i=1}^p (\bar{d}_i + \bar{\varepsilon}_i) |S_i| + \sum_{i=1}^p \frac{1}{\gamma_{0i}} \tilde{\theta}_i \dot{\theta}_i \quad (5.76)$$

La substitution de la loi d'adaptation des paramètres (5.73) dans (5.76) donne

$$\dot{V} \leq -S^T K_1 S - \sum_{i=1}^p k_{0i} |S_i| + \sum_{i=1}^p (\bar{d}_i + \bar{\varepsilon}_i) |S_i| - \sigma_i |S_i| \tilde{\theta}_i^T \theta_i \quad (5.77)$$

On peut montrer la relation suivante:

$$\sum_{i=1}^p -2\tilde{\theta}_i^T \theta_i \leq -\sum_{i=1}^p \|\tilde{\theta}_i\|^2 + \sum_{i=1}^p \|\theta_i^*\|^2 \quad (5.78)$$

En choisissant  $k_{0i} \geq \bar{d}_i + \bar{\varepsilon}_i + 0.5\sigma_i \|\theta_i^*\|^2$  and remplaçant

Si nous choisissons  $k_{0i} \geq \bar{d}_i + \bar{\varepsilon}_i + 0.5\sigma_i \|\theta_i^*\|^2$  et nous remplaçons (5.78) dans (5.77), nous obtenons:

$$\dot{V} \leq -\sum_{i=1}^p k_{1i} S_i^2 \quad (5.79)$$

De (5.79),  $\dot{V}$  est toujours négative. Cela implique que les signaux  $S_i$  et  $\tilde{\theta}_i$  sont bornés.

Puisque  $\tilde{\theta}_i$ ,  $\theta_i^*$ ,  $\psi_i(x)$  et  $S_i \in L_\infty$ , nous pouvons facilement montrer que  $\theta_i$  et  $u_i \in L_\infty$ .

La bornitude de  $S_i$  implique celle de  $x$ . Le fait que  $V$  est une fonction non croissante en fonction du temps et inférieurement bornée, la limite  $\lim_{t \rightarrow \infty} V(t) = V(\infty)$  existe. En intégrant (5.79) de 0 à  $\infty$ , nous avons

$$\int_0^\infty \sum_{i=1}^p k_{1i} S_i^2(t) dt \leq V(0) - V(\infty) < \infty \quad (5.80)$$

ceci implique que  $S_i \in L_2$ . A partir de (5.71), on peut conclure concernant la bornitude de  $\dot{S}_i$ .

Finalement, puisque  $S_i \in L_2 \cap L_\infty$  et  $\dot{S}_i \in L_\infty$ , en utilisant le lemme de Barbalat, nous concluons que  $S_i(t) \rightarrow 0$  quand  $t \rightarrow \infty$ . Par conséquent, les erreurs de poursuite et leurs dérivées convergent vers zéro, i.e.  $e^{(j)}_i(t) \rightarrow 0$  quand  $t \rightarrow \infty$  pour  $i = 1 \dots p$  et  $j = 0, 1, \dots, r_i - 1$ .

### 5.4 Simulation

Pour illustrer les performances de cette commande floue adaptative, nous faisons des simulations numériques sur ce robot à deux roues. Pour la résolution numérique des équations différentielles, nous avons adopté la méthode de RK4. La trajectoire de références est choisie comme suit:

$$\begin{aligned} x_r &= \sin(\alpha t) \\ y_r &= \rho \cos(\beta t) \end{aligned} \quad (5.80)$$

avec  $\alpha = 2/3$ ,  $\beta = 1/3$  et  $\rho=2$ . Dans le plan  $x$ - $y$ , ce chemin de référence prend la forme d'un huit. Les paramètres du robot sont choisis comme dans les chapitres 3 et 4.

L'objectif de la commande est de forcer les sorties du système  $y = P + d(\cos(\phi), \sin(\phi))$  à suivre la trajectoire de référence (5.80).

Pour tester la robustesse de la commande proposée comme dans les chapitres précédents, nous avons considéré deux cas: i.e. avec perturbation ( $\Delta = 0.15[c_4\dot{\theta}_l - c_5\dot{\theta}_r; c_5\dot{\theta}_l - c_4\dot{\theta}_r]^T$ ) et sans perturbation ( $\Delta = [0 \quad 0]^T$ ).

Deux systèmes flous ( $\theta_1^T \psi_1(x)$ ,  $\theta_2^T \psi_2(x)$ ) sont employés pour implémenter le contrôleur et ont les variables  $y, \dot{y}, \phi, \dot{\phi}$  comme des entrées. Pour chaque variable d'entrée des systèmes flous, nous définissons trois fonction d'appartenance de type (triangulaire au centre et trapézoïdales sur les cotés de l'univers de discours).

Les paramètres de conception sont choisis comme suit:  $\gamma_1 = \gamma_2 = 500$ ,  $\sigma_1 = \sigma_2 = 0.0001$ ,  $\lambda_1 = \lambda_2 = 2$ ,  $\lambda_1 = \lambda_2 = 2$ ,  $k_{01} = k_{02} = 2$  et  $k_{11} = k_{22} = 0.5$ . Les conditions initiales sont fixées comme  $\theta_1(0) = \underline{0}$  et  $\theta_2(0) = \underline{0}$ .

Les figures (5.11)-(5.14) montrent les résultats de simulation obtenus par le contrôleur adaptatif floue (5.70), pour le cas sans perturbations. Ces résultats montrent que les performances de poursuite sont suffisamment bonnes et tous les signaux du système en boucle fermée sont bornés et les erreurs de poursuite sont faibles.

Les résultats de simulation pour le cas avec perturbation sont représentés sur les Figures (5.15)-(5.18). Nous remarquons d'après ces figures que les performances en terme de poursuite sont très bonnes et ne sont pas dégradées malgré la présence des perturbations externes, et les commandes sont bornées et admissibles.

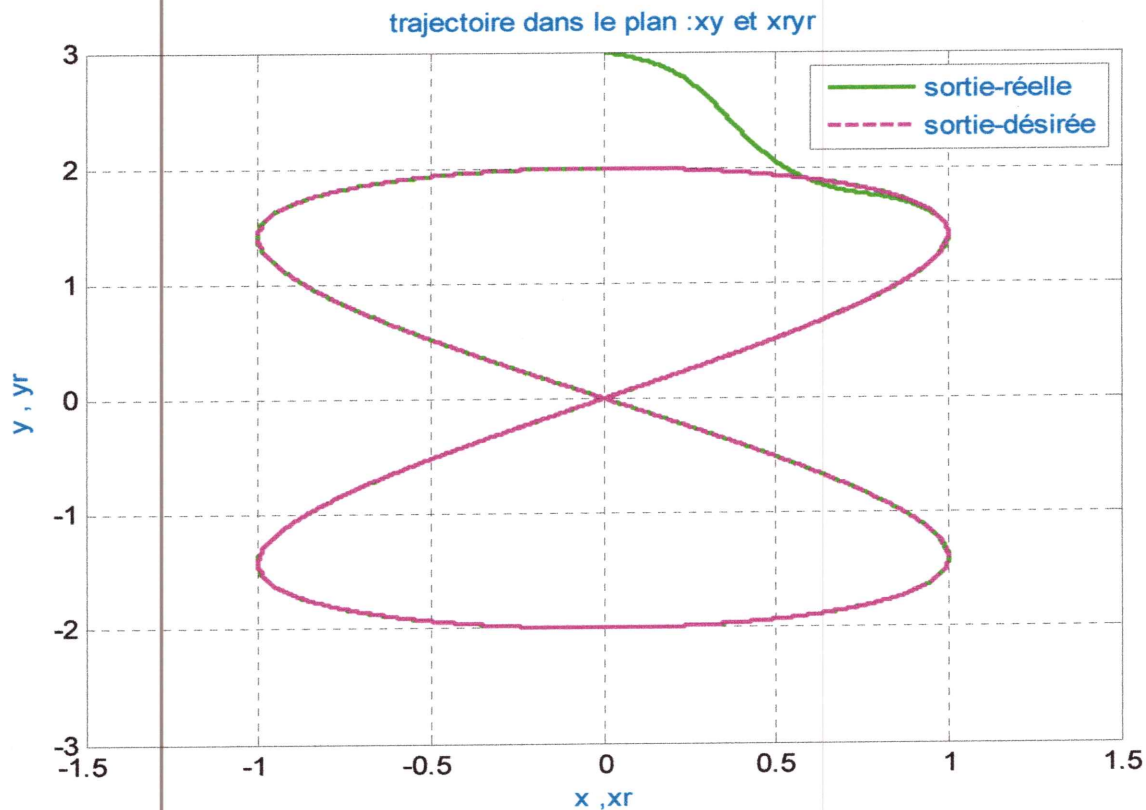


Figure 5.11 : Trajectoire tracée par le robot mobile (pour le cas sans perturbations).

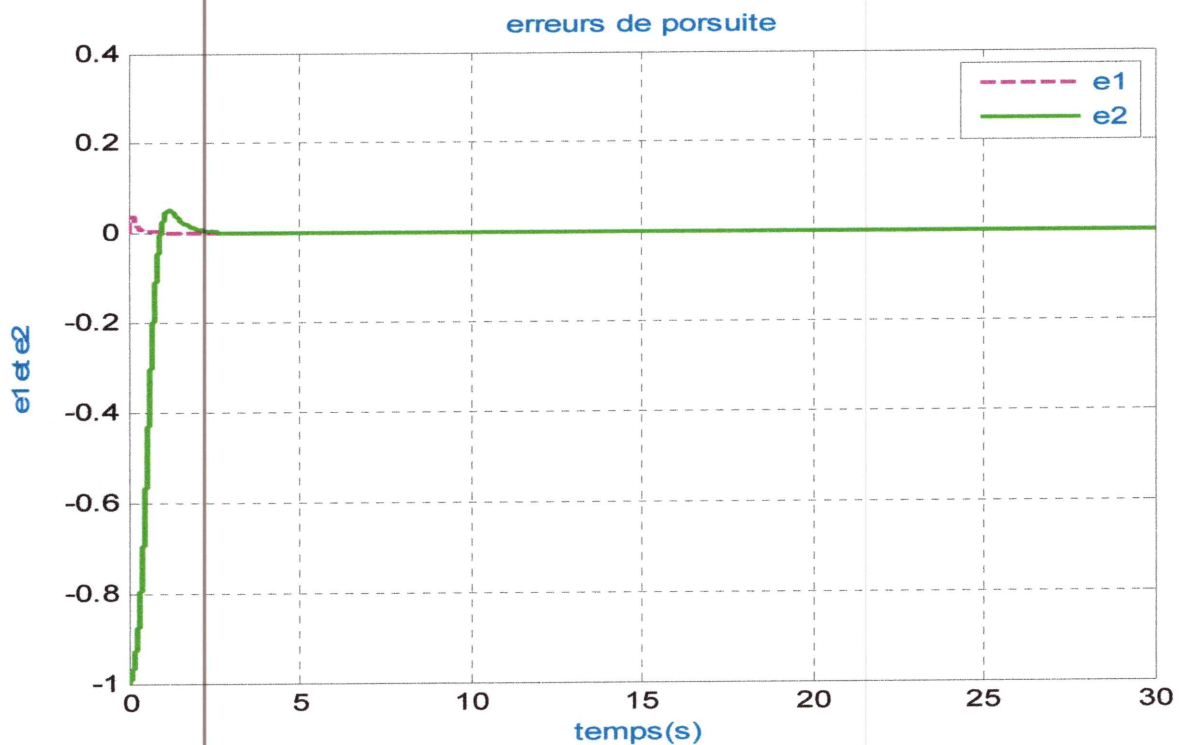


Figure 5.12: Signaux d'erreurs de poursuite (pour le cas sans perturbations).

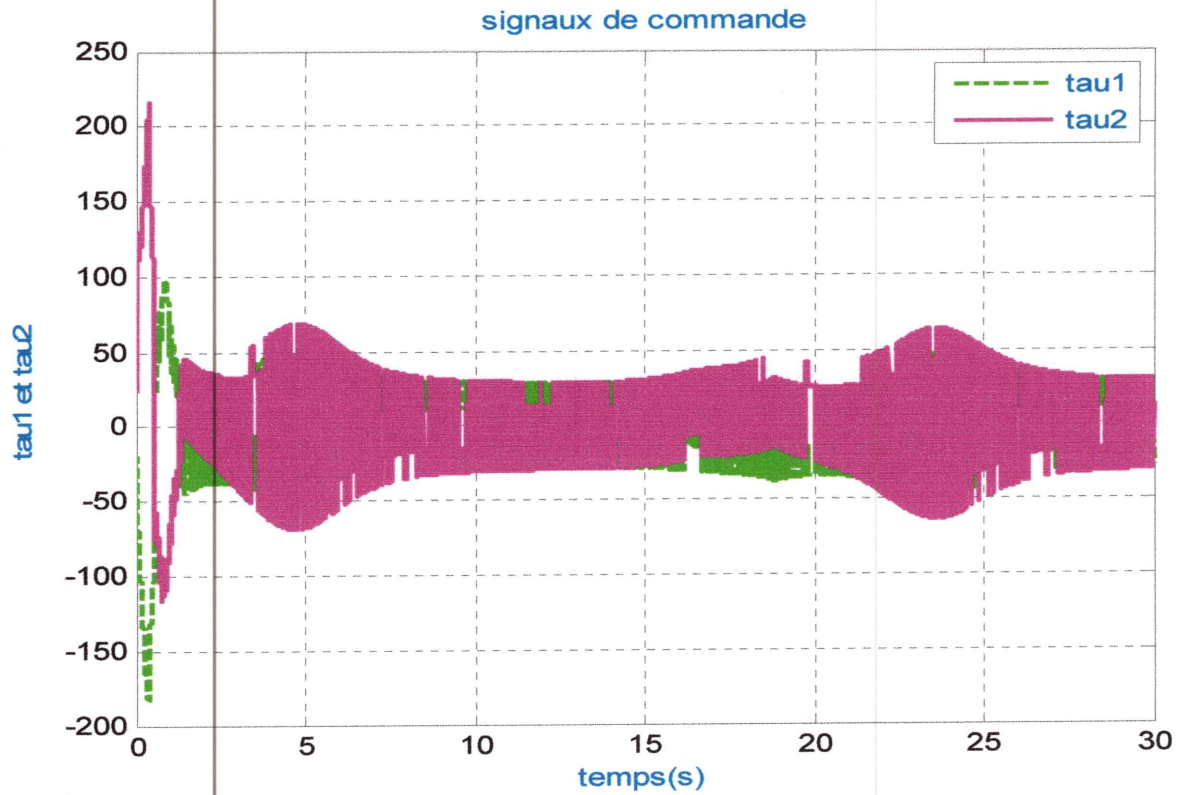


Figure 5.13: Signaux de commande (pour le cas sans perturbations).

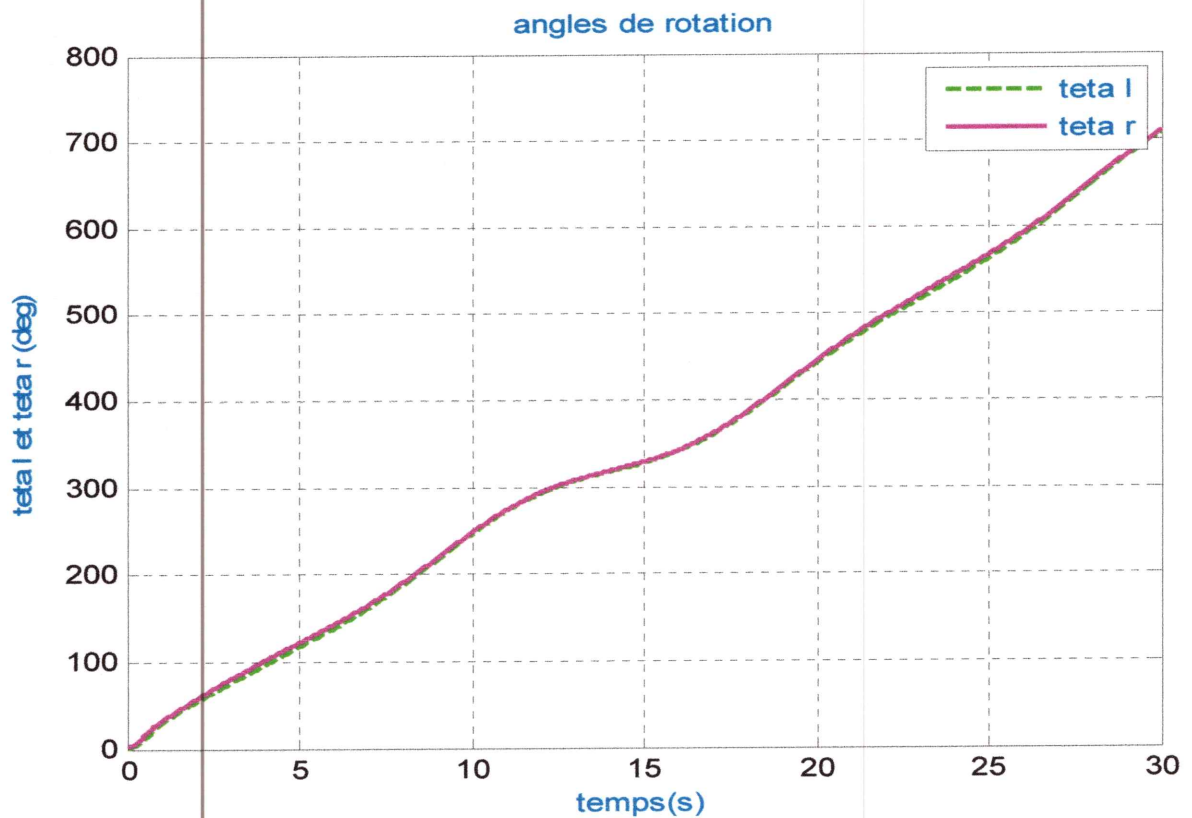


Figure 5.14: Angles de rotation (pour le cas sans perturbations).

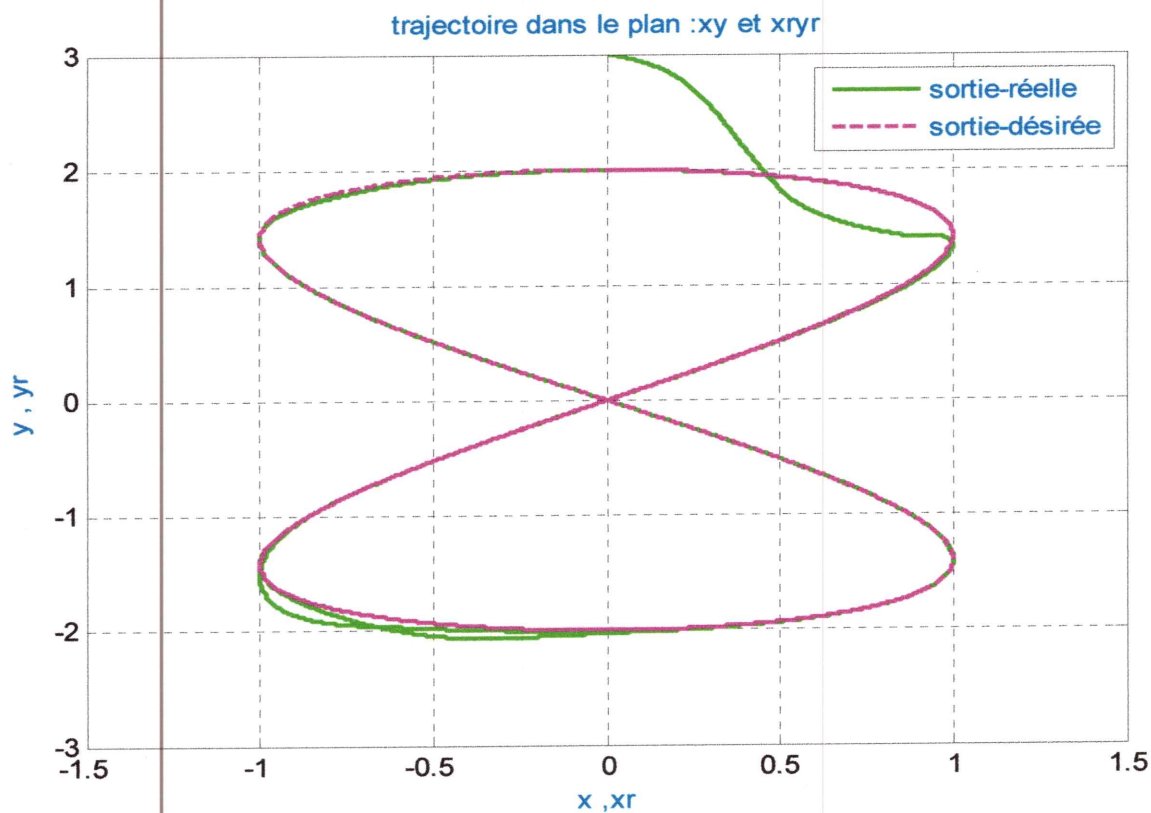


Figure 5.15: Trajectoire tracée par le robot mobile (pour le cas avec perturbation).



Figure 5.16: Signaux d'erreurs de poursuite (pour le cas avec perturbation).



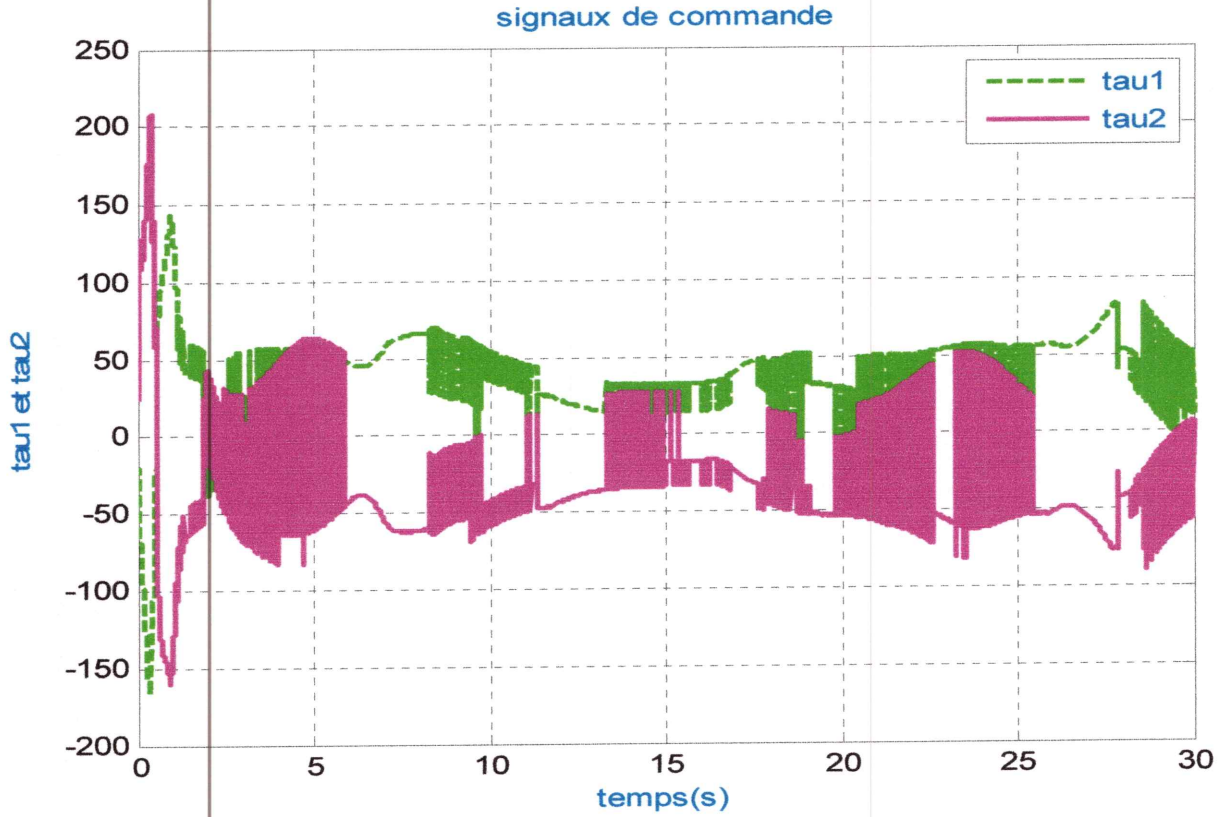


Figure 5.17: Signaux de commande (pour le cas avec perturbation).

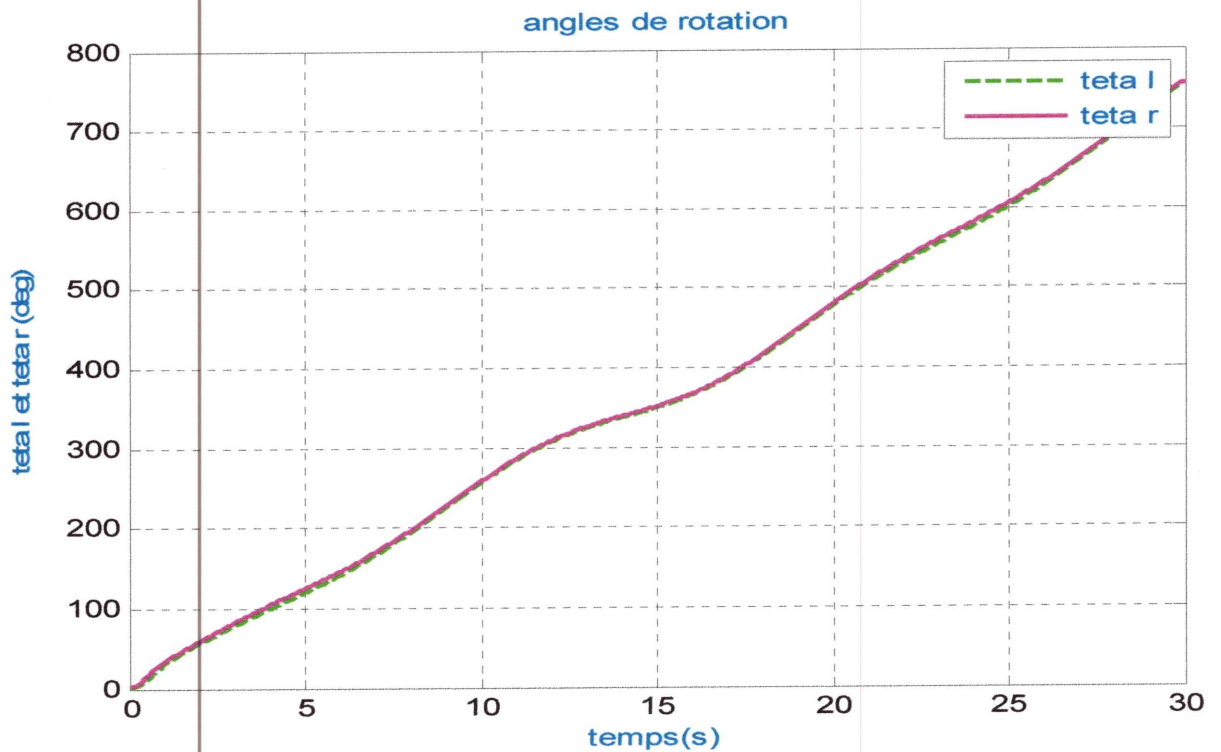


Figure 5.18: Angles de rotation (pour le cas avec perturbation).

## 5.5 Conclusion

Dans cette section, nous nous sommes principalement intéressés à l'étude des notions de base de la logique floue et aux différents mécanismes et outils mathématiques nécessaires à la compréhension de cette logique.

Un système flou est une machine de décision composée essentiellement de quatre parties : la fuzzification, la base de règles, le système d'inférence et la défuzzification. Les systèmes flous se distinguent généralement selon la nature de la conclusion de leurs règles: symbolique (système flou de Mamdani) ou algébrique (système de TS). L'inférence dans les deux systèmes flous a été donnée en détails et une comparaison montrant les inconvénients et les avantages de chacun d'eux a été faite. Nous avons aussi montré que les systèmes flous en général ont la propriété d'approximation universelle.

Une commande floue adaptative a été conçue pour une classe des systèmes multivariables non linéaires inconnues. Les systèmes flous sont intégrés dans la commande floue pour estimer les non linéarités inconnues du système. Cette commande est ensuite testée par simulation numérique sur le robot mobile à deux roues.

# Conclusion Générale

## Conclusion générale

Dans ce travail, nous nous sommes particulièrement intéressés à la modélisation et à la conception des commandes non linéaires (à savoir: commande par linéarisation E/S, commande à mode glissant et commande floue adaptative) des robots mobiles non-holonomes à 2 roues. Pour chaque commande développée, un ensemble de testes par simulation numérique a été effectué à fin de vérifier leur applicabilité et efficacité. Notre mémoire a été principalement articulé en cinq chapitres.

*Dans le premier chapitre*, nous avons présenté des définitions et généralités sur les robots ainsi que leur principe de fonctionnement. Nous les avons classés en quatre grandes familles. L'accent a été mis sur les robots mobiles.

*Dans le deuxième chapitre*, nous avons essayé d'établir les modèles dynamiques des robots mobiles non-holonomes bicycles, en utilisant le formalisme d'Euler Lagrange.

*Dans le troisième chapitre*, nous avons conçu une commande non linéaire linéarisante par bouclage. Cette commande, qui dépend étroitement du modèle, a été appliquée au robot mobile bicycle. En fait, en considérant des perturbations dans le modèle, les performances de poursuite se dégradent.

*Dans le quatrième chapitre*, nous avons appliqué pour un robot mobile bicycle une commande à mode glissant. Les performances de cette commande ont été vérifiées via une simulation numérique effectuée sur ce robot mobile bicycle. Les résultats de simulation obtenus, en présence des perturbations externes, ont montré clairement l'efficacité de la commande à mode glissant (en terme de précision, rapidité et rejet de perturbations (i.e. de robustesse)). Cependant, son inconvénient majeur réside dans l'apparition du phénomène de chattering qui se manifeste dans les grandeurs asservies et le signal de commande.

*Dans le dernier chapitre*, une commande floue adaptative pour une classe des systèmes non linéaires multivariables a été développée en se basant sur l'approche de Lyapunov et le théorème d'approximation universelle. Cette commande qui ne dépend pas du modèle a été ensuite testée en présence des perturbations et des dynamiques incertaines sur notre robot mobile.

Ce travail reste perfectible et ouvre d'autres axes de recherches, à savoir:

- 1- L'introduction des observateurs pour l'estimation des états. En fait, toutes les commandes développées dans ce mémoire sont basées sur la mesure complète de l'état.
- 2- Le test de ces commandes sur des robots réels.

# Bibliographie

## *BIBLIOGRAPHIE*

- [1] D. Filiat, Robotique Mobile, École Nationale Supérieure de Techniques Avancées Paris Tech, 3 octobre 2013
- [2] L. Matignon, Introduction à la robotique, GREYC-CNRS Université de Caen, France, 2011/2012
- [3] É. Beaudry, Planification de tâches pour un robot mobile autonome. Thèse de Maître de sciences, Université de Sherbrooke, Québec, Canada. août 2006
- [4] S. Lens, Locomotion d'un robot mobile. Mémoire d'ingénieur civil Informaticien, Université de Liège, Faculté des Sciences Appliquées, Institut Montefiore. Mai 2008.
- [5] B. Bayle, Robotique mobile. Ecole Nationale Supérieure de Physique de Strasbourg Université Louis Pasteur. Année 2008–2009.
- [6] C. Raimundez, Adaptive Tracking in mobile Robots with Input-Output Linearization. Nonlinear Control Group Department of systems et Control Engineering University of Vigo .Spain .Mai 2014
- [7] H. Boursès, Systèmes linéaires de la modélisation à la commande. Hermès-science, Paris, France, 2006.
- [8] L. Jaulin, Robotique mobile. 11 septembre 2014
- [9] F. L. Lewis, C. T. AbdallaH, and D. M. Dawson, Control of Robot Manipulators. New York: MacMillan, 1993.

[10] L. Xiasong, An investigation of adaptive fuzzy Sliding mode control for robotic manipulators. Mémoire de Master. Université de Carleton Ottawa, Ontario. 2007.

[11] B. Idris, B. Said Commande par mode glissant d'ordre deux. Mémoire d'ingénieur. Université de setif .2002.

[12] N. Bounar, commande par apprentissage itératif : application au pendule inversé. Thèse de Magister, Ecole Militaire Polytechnique ,Alger ,2004.

[13] A. Khaldi, commande par les modes glissants non linéaires de la MSAP alimentée par un onduleur de tension dans le cas général à N niveaux. PFE, USTHB, 2002.

[14] L. X. Wang, Stable adaptive fuzzy control of nonlinear systems. IEEE Trans. Fuzzy Systems, vol. 1, pp. 146–155, 1993.

[15] L. X. Wang, Adaptive Fuzzy Systems and Control: Design and Stability Analysis, Prentice-Hall, Englewood Cliffs, NJ, 1994.

[16] S. Labiod, Contribution à la commande adaptative floue des systèmes non linéaires. Thèse de doctorat, ENP, Alger, 2004.

[17] A. Boulkroune, Commande floue adaptative avec et sans observateur des systèmes non linéaires incertains, Thèse de doctorat, ENP, Alger, 2009.



- [18] F. Chevré and F. Guély, La logique floue, *Cahier Technique*, no.191, pp. 1-28, 1998.
- [19] L. A. Zadeh, Fuzzy sets *Information and control*. vol. 8, pp. 338-353, 1965.
- [20] L. Jouffe, Apprentissage de systèmes d'inférence floue par des méthodes de renforcement. Thèse de doctorat, IRISA, Université de Rennes I, 1997.
- [21] Y. Nakoula, Apprentissage des modèles linguistiques flous par un jeu de règles pondérées. Thèse de doctorat, Université de Savoie, France, 1997.
- [22] R. Babuska and H. B Verbruggen, A new identification method for linguistic fuzzy models. in *Proc. of FUZZ-IEEE '95*, Yokohama, Japan, pp. 1897-1902, March 20-24, 1995.
- [23] T. Takagi and M. Sugeno, Fuzzy identification of systems and its applications to modeling and control, *IEEE Trans. Systems, Man, and Cybernetics*, vol. 15, no.1, 1985.
- [24] M. Sugeno and G. T. Kang, Fuzzy modeling and control of multilayer incinerator, *Fuzzy Sets and Systems*, vol. 18, pp.329, 1986.
- [25] J. J. Slotine et W. Li (1991). *Applied Nonlinear Control*, Englewood Cliffs, NJ: Prentice-Hall.

[26] O. Akhrif et G. L. Blackenship (1988). *Robust Stabilization of FeedbackLinearizable Systems*. Proceedings of the 27th conference on decision and control, Austin, Texas, pp. 1714-1719.

[27] Amieur. Toufik (2009) *Commande des Systèmes Non Linéaires par Mode Glissant Flou*. Masters thesis, Université Mohamed Khider - Biskra.

