

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

Université Mouloud MAMMERI de Tizi-Ouzou



Faculté de Génie électrique et D'informatique
DEPARTEMENT D'AUTOMATIQUE

**Mémoire de Fin d'Etudes
De MASTER ACADEMIQUE**
Spécialité : **Commande des systèmes**

Présentés par
LAGAB Salima
TONKIN Karima

Mémoire dirigée par **KHERRAZ KHedoudja**

Thème

**Commande par mode glissant
Neuro-flou d'un bras manipulateur**

Mémoire soutenu publiquement le 02 juillet 2014 devant le jury composé de:

M Prénom NOM

Grade, Lieu d'exercice, Président

M Prénom NOM

Grade, Lieu d'exercice, Rapporteur

M Prénom NOM

Grade, Lieu d'exercice, Examineur

M Prénom NOM

Grade, Lieu d'exercice, Examineur

REMERCIEMENT

Nous remercions avant tout le bon Dieu de nous avoir donné la force et la volonté pour achever ce travail.

Nous tenons à exprimer nos vifs remerciement à notre promotrice **M^{me} KHARRAZ** pour avoir dirigé ce travail pour son suivi, son conseil et sa disponibilité à chaque instant.

Enfin, je remercie toutes les personnes de près ou de loin qui m'ont aidé à finir ce travail.

DÉDICACES

Je tiens à dédier ce mémoire :

A ma très chère Mère et à mon cher père, en témoignage et en gratitude de leurs dévouements, de leurs soutiens permanents durant toutes mes années d'études, leurs sacrifices illimités, leurs réconfort moral.

A mes chères sœurs et mon petit frère Menad, et ma grande famille.

A mes camarades de la section Master 2 Automatique tous ceux de Faculté de Génie électrique et d'Informatique et tous mes enseignants de l'Université Mouloud MAMMERI de Tizi-Ouzou.

Je dédie ce travail également à mon binôme et à toute sa famille.

L.SALIMA

DÉDICACES

Je tiens à dédier ce mémoire :

A ma chère maman et à mon cher papa, en témoignage et en gratitude de leurs dévouements, de leurs soutiens permanents durant toutes mes années d'études.

A ma chère grand-mère et mon cher grand-père.

A ma chère sœur Safia et mes chers frères Zidane, Nabil et Boudjemaa.

En prenant le risque d'oublier Mustapha : vive la patience.

A mes amies Nassima et Wafia et Celia a mes camarades de la section Master 2 Automatique tous ceux de Faculté de Génie électrique et d'Informatique et tous mes enseignants de l'Université Mouloud MAMMERI de Tizi-Ouzou.

Je dédie ce travail également à mon binôme Salima et à toute sa famille.

T.KARIMA

Introduction générale.....	1
----------------------------	---

Chapitre I : Généralités et modélisation des robots manipulateurs

I.1 Introduction	3
I.2 Généralités sur les robots	3
I.2.1 Définitions générales.....	3
I.2.2 Historique	4
I.2.3 Constitution et Géométrie des Robots Manipulateurs	4
I.3 Mouvement de robot	5
I.3.1 Les degrés de liberté (d .d .l).....	6
I.3.2 Espace des coordonnées.....	6
I.3.2.1 Coordonnées opérationnelles (géométrique)	6
I.3.2.2 Coordonnées généralisées (articulaires).....	6
I.4 Modélisation et commande d'un bras manipulateur	6
I.4.1 Modélisation géométrique	7
I.4.2 Modélisation cinématique	7
I.4.3 Modélisation dynamique	7
I.5 Détermination du modèle dynamique.....	7
I.6 Conclusion	13

Chapitre II : Théorie de la commande par mode glissant

II.1 Introduction	14
II.2 Principe de fonctionnement	14
II.3. Définition du régime glissant	15
II.4. Propriétés de la convergence en temps fini.....	17
II.5 Choix de la surface de glissement	17
II.6 Propriétés de la robustesse	18
II.7 Le phénomène du chattering	18
II.8 Solution pour la suppression de chattering	19
II.9 Conclusion	21

Chapitre III : Les systèmes flous et réseaux de neurones

III.1 Introduction	22
III.2 Les systèmes d'inférence floue.....	22
III.2.1 Introduction	22
III.2.2 Historique	22
III.2.3 Principe de la logique floue.....	23
III.2.4 Les bases de la logique floue	24
III.2.4.1 Les Fonctions d'appartenance.....	24
III.2.4.2 Univers de discours	24
III.2.4.3 Les opérateurs	24
III.2.5 Structure interne d'un système d'inférence flou	25
III.2.5.1 Fuzzification	25
III.2.5.1.1 Fuzzification numérique	25
III.2.5. Fuzzification symbolique	26
III.2-5-2-Base des règles floues.....	26
III.2-5-3-Inférence Floue	26
III.2.5.4 Défuzzification	27
III.2.5.4.1 Défuzzification numérique	27
III.2.5.4.2 Défuzzification symbolique	28
III.2.6 Les modèle d'inférence floue	28
III.2.6.1 Modèle de MAMDANI	28
III.2.6.2 Modèle de TAKAGI-SUGENO	28
III.2.6.3 Modèle TSUKAMOTO.....	29
III.3 Les réseaux de neurones artificiels	29
III.3.1 Introduction.....	29
III.3.2 Historique	29
III.3.3 Principes de base des RNA	30
III.3.4 Le neurone biologique	30
III.3.5 Le neurone formel	31
III.3.6 Fonction d'activation	32
III.3.7 Différents types de réseaux de neurones	32
III.3.7.1 Les réseaux de neurones non bouclés	32
III.3.7.2 Les réseaux de neurones bouclés	33
III.3.8 L'apprentissage des réseaux de neurones	34

III.3.8.1 L'apprentissage non supervisé	34
III.3.8.2 L'apprentissage supervisé	35
III.3.9 Systèmes Neuro-flous	36
III.3.9.1 Modèles Neuro-flous hybrides	37
III.3.9.1.1 FALCON et GARIC	37
III.3.9.1.2 NEFCLASS	38
III.3.9.1.3 Système d'inférence Neuro-flou adaptative(ANFIS)	38
III.3.9.1.3.1 Structure d'ANFIS	38
III.3.10 Conclusion	41

Chapitre IV : Commande par mode glissant neuro-flou

IV.1 Introduction	42
IV.2 Stratégie de la commande par mode glissant	42
IV.3 Simulation et interprétation.....	43
IV.4 Stratégie de la commande hybride.....	48
IV.4.1 Régulateur Neuro-flou	48
IV.5 Simulation et interprétation	51
IV.6 Conclusion.....	63
Conclusion général.....	64

Liste des figures

Figure I.1 : chaine cinématique simple ouverte.

Figure I.2 : Liaisons (Rotoïde) et (Prismatique) considérées en robotique.

Figure I.3: Robot manipulateur à 2dd

Figure II.1 : Différents mode de convergence pour la trajectoire d'état

Figure II.2 : Phénomène de chattering

Figure III.1 : Structure interne d'un système flou

Figure III.2 : Un neurone avec son arborisation dendritique

Figure III.3 : neurone artificiel

Figure III.4 : fonctions d'activation les plus utilisées

Figure III.5 : rétro-propagation du gradient

Figure III.6 : Architecture ANFIS

Figure(IV.1) : Trajectoires désirée et réelle (articulation 1)

Figure(IV.2) : Trajectoires désirée et réelle (articulation 2)

Figure(IV.3) : Profil des erreurs de position et de vitesse (articulation 1 et 2)

Figure(IV.4) : Evolution des commandes (articulation 1 et 2)

Figure(IV.5) : Evolution des commandes (articulation 1 et 2)

Figure(IV.6) : Evolution des surfaces de glissement (articulation 1, 2)

Figure(IV.7) : Profil des erreurs de positions et de vitesses (articulation 1 et 2)

Figure(IV.8) : Evolution des commandes (articulation 1 et 2)

Figure(IV.9) : Evolution des surfaces de glissement (articulation 1 et 2)

Figure(IV.10) : Le régulateur de type segeno sous forme de RNA(articulaion 1)

Figure(IV.11) : Le régulateur de type segeno sous forme de RNA(articulaion 1)

Figure (IV. 12) : Schéma (commande par mode glissant avec le neuro-flou)

Figure(IV.13) : Trajectoires désirée et réelle (articulation 1)

Figure(IV.14) : Trajectoires désirée et réelle (articulation 1)

Figure(IV.15) : Trajectoires désirée et réelle (articulation 2)

Figure(IV.16) : Trajectoires désirée et réelle (articulation 2)

Figure(IV.17) : Profil des erreurs de position et de vitesse (articulation 1 et 2)

Figure(IV.18) : Evolution des commandes (articulation 1 et 2)

Figure(IV.19) : Evolution des surfaces de glissement (articulation 1 et 2)

Figure(IV.20) : Profil des erreurs de position et de vitesse (articulation 1 et 2)

Figure(IV.21) : Evolution de la commande (articulation 1 et 2)

Figure(IV.22) : Evolution de la surface (articulation 1 et 2)

Figure(IV.23) : trajectoires désirée et réel (articulation 1)

Figure(IV.24) : Trajectoires désirée et réelle (articulation 2)

Figure(IV.25) : Profil des erreurs de position et de vitesse (articulation 1 et 2)

Figure(IV.26) : Evolution des commandes (articulation 1 et 2)

Figure(IV.27) : Evolution des surfaces de glissement (articulation 1 et 2)

Figure(IV.28) : trajectoires désirée et réelle (articulation 1)

Figure(IV.29) : trajectoires désirée et réelle (articulation 2)

Figure(IV.30) : Profil des erreurs de position et de vitesse (articulation 1 et 2)

Figure(IV.31) : Evolution des commandes (articulation 1 et 2)

Figure(IV.32) : Evolution des surfaces de glissement (articulation 1 et 2)

Liste des tableaux

Tableau III.1 : les opérateurs en logique floue

Introduction générale

Quand on parle de robotique, plusieurs idées viennent à l'esprit de chacun de nous. Dans les procédés hautement automatisés, les robots manipulateurs sont souvent employés pour augmenter le volume de production et améliorer la qualité du produit.

En robotique, il est important toujours de considérer le robot comme un dispositif réel, avec une architecture et des contraintes mécaniques spécifiques, dans le but d'accomplir des tâches réelles dans un monde réel. Un robot peut effectuer de nombreuses tâches comme l'assemblage et la saisie d'objets, ou l'exploration [B.N]. Indépendamment de la tâche, le robot doit effectuer une séquence d'actions, tel que: bouger un bras, fermer des pinces ou se propulser. Puis, chaque action conduit à un mouvement. Ainsi, afin d'accomplir avec succès une tâche, un robot doit être capable de planifier, c'est-à-dire trouver par lui-même et à l'avance, la séquence d'action à exécuter.

C'est avec les développements scientifiques, spécifiquement de l'électronique et de l'informatique, mais aussi automatique, mathématique, mécanique, matériaux, que la technologie robotique a progressé. Les robots actuels sont dotés d'une « intelligence » qui leur donne une certaine autonomie qui va leur permettre de se diffuser dans de nouveaux domaines. Tous ces critères à satisfaire montrent la nécessité d'une bonne loi de commande plus performante et plus robuste qui répond aux exigences [GA].

Lorsque la partie commandée du processus est faiblement perturbée, les commandes classiques, par exemple à action proportionnelle intégrale dérivée, peuvent s'avérer suffisantes si les exigences en précision et en performance du système ne sont pas trop sévères. Dans le cas contraire, il faut concevoir des algorithmes de commande assurant une robustesse du système. Dans ce travail, nous avons choisi la technique des modes glissants connue par sa grande robustesse.

La caractéristique principale d'une commande par modes glissants est sa structure variable avec commutation sous certaines conditions prédéfinies lors de la synthèse de part et d'autre d'une surface choisie à priori, appelée surface de glissement. Le système commandé est dit alors en régime glissant et sa dynamique devient moins sensible aux variations paramétriques, aux erreurs de modélisation et à certaines perturbations externes. La mise en œuvre d'une telle commande est relativement simple et ne nécessite pas une représentation exacte du processus, ce qui est un avantage considérable.

Or, la nature incertaine des systèmes non linéaires rend difficile si ce n'est impossible de disposer d'une description analytique de la dynamique du système. Le second inconvénient réside dans l'utilisation de la fonction signe dans la loi de commande pour assurer le passage

Introduction générale

de la phase d'approche à celle du glissement. Ceci donne lieu au phénomène de broutement qui consiste en des variations brusques et rapides du signal de commande.

En effet, pour le premier inconvénient plusieurs travaux ont été focalisés sur la combinaison des modes glissants avec une autre commande non linéaire où la dynamique du système incertain est approximée à l'aide d'un réseau de neurones, d'un système flou ou une combinaison des deux (système neuro-flou) [Wan, 97], [Yoo,98], [Man, 03], [Hsu, 04].

La logique floue et les réseaux de neurones appartiennent à la famille de l'intelligence artificielle (IA), une discipline majeure du génie informatique. L'intelligence artificielle trouve ses applications dans une vaste gamme de domaines, incluant la commande.

Le contenu de notre mémoire est organisé en quatre chapitres principaux.

L'objectif de notre travail est de remédier au phénomène de chattering de la commande par mode glissant. Pour cela, nous avons remplacé la partie discontinue qui est responsable de ce problème par une partie neuro-floue. Cette commande est appliquée sur un bras manipulateur pour le suivi de trajectoire en position et en vitesse.

Dans le premier chapitre, nous avons donné brièvement quelques notions de base à propos de notre bras manipulateur qui est à 2 degrés de liberté. Ensuite, nous sommes basées sur la modélisation dynamique à l'aide de formalisme Euler- Lagrange.

Dans le deuxième chapitre, nous avons décrit la commande par mode glissant qui est une technique de commande non linéaire. Elle est caractérisée par la discontinuité de la commande aux passages par une surface de commutation appelée surface de glissement.

Dans le troisième chapitre, nous avons abordé les réseaux de neurones et la logique floue qui sont des systèmes qui appliquent les principes du réseau neuronal au raisonnement flou.

Dans le quatrième chapitre, nous avons donné les résultats de simulations de la commande par mode glissant suivi par la commande mode glissant-neuro-flou appliquées sur un bras manipulateur à deux axes.

I.1 Introduction

Un système robotique, quel qu'il soit, doit être capable de déterminer et de réaliser les mouvements nécessaires pour accomplir une tâche donnée. En robotique, on appelle planification de mouvement ce problème particulier du calcul préalable du mouvement à accomplir. C'est un problème fondamental en robotique qui a fait et continue de faire l'objet de nombreux travaux de recherche depuis la fin des années 60.

Dans sa forme la plus générale, la planification de mouvement se définit de la façon suivante : étant donné un modèle du système robotique et de son environnement, planifier un mouvement consiste à calculer le mouvement que doit effectuer le système robotique pour atteindre un objectif fixé à priori.

Dans ce chapitre, nous allons donner quelques généralités sur les robots et les différentes modélisations utilisées.

I.2 Généralités sur les robots

I.2.1 Définitions générales

Le petit Larousse définit un robot comme étant un appareil automatique capable de manipuler des objets, ou d'exécuter des opérations selon un programme fixe ou modifiable.

En fait, l'image que chacun s'en fait est généralement vague, souvent un robot est défini comme un manipulateur automatique à cycles programmables. Un robot est un système mécanique poly-articulé muni par des actionneurs et commandé par un ordinateur qui est destiné à effectuer une grande variété de tâches.

Pour « mériter » le nom de robot, un système doit posséder une certaine flexibilité, caractérisée par les propriétés suivantes :

La versatilité : Un robot doit avoir la capacité de pouvoir exécuter une variété de tâches, ou la même tâche de différente manière.

L'auto-adaptabilité : Un robot doit pouvoir s'adapter à un environnement changeant au cours de l'exécution de ses tâches.[10]

I.2.2 Historique

1947 : premier manipulateur électrique téléopéré.

1954 : premier robot programmable.

1961 : apparition d'un robot sur la chaîne de montage de General Motors, premier robot avec contrôle en effort.

1963 : utilisation de la vision pour commander un robot.

1973 : premier robot mobile à roues.

I.2.3 Constitution et géométrie des robots manipulateurs

Un robot manipulateur est un ensemble des corps assemblés à l'aide de liaisons munies d'actionneurs et possédant deux corps particuliers. Le premier appelé base est le socle du robot; le second appelé terminal est le support d'un préhenseur ou d'un outil.

Les robots manipulateurs auxquels on s'intéresse dans le cadre de ce travail sont dotés d'une structure de chaîne cinématique simple ouverte Figure (I.1).

Autrement dit, tous les corps possèdent deux liaisons. Les corps sont supposés rigides et les seules liaisons considérées sont, soit rotoïdes (notées R), soit prismatiques (notées P).

Elles autorisent donc chacune un seul degré de liberté correspondant à un mouvement élémentaire, soit de rotation, soit de translation. La figure 2 représente la schématisation usuelle de ces deux types de liaisons. On nommera les chaînes cinématiques par l'énumération des liaisons qui les composent en partant de la base; par exemple RRP pour une chaîne comportant trois liaisons, les deux premières étant rotoïdes et la dernière est prismatique. On se place, en outre, dans le cas des robots manipulateurs munis d'un actionneur par liaison.[9]

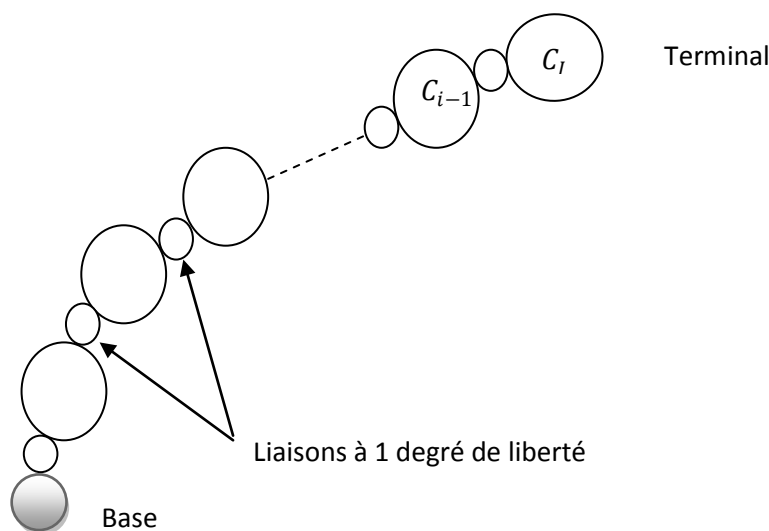


Figure I.1 : chaîne cinématique simple ouverte.

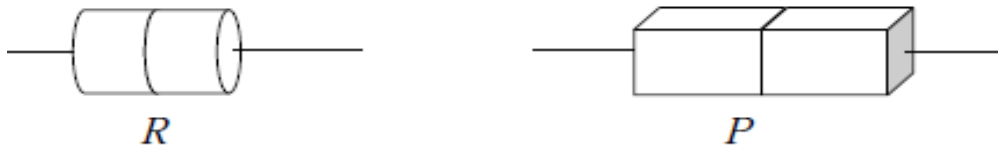


Figure I.2 : Liaisons (Rotoïde) et (Prismatique) considérées en robotique.

La géométrie du mécanisme est alors définie de manière systématique par l'introduction de repères affins orthonormés particuliers liés à chaque corps du mécanisme.

I.3 Mouvement de robot

I.3.1 Les degrés de liberté (d .d .l)

En robotique, le nombre de degré de liberté (N.D.L) d'un manipulateur est le nombre de paramètres indépendants nécessaire pour décrire la situation de l'organe terminal. [10]

I.3.2 Espace des coordonnées

Pour définir correctement le mouvement de l'organe terminal (trajectoire, vitesse et accélération), il faut choisir un repère de coordonnées en rapport avec le nombre et le type des (d .d .l). On distingue deux types de coordonnées :

I.3.2.1 Coordonnées opérationnelles (géométriques)

Elles permettent de caractériser et de mesurer les composantes de l'état (position et orientation) d'un repère orthonormé direct lié à l'organe terminal, par rapport à un repère fixe lié à la base du robot.

I.3.2.2 Coordonnées généralisées (articulaires)

On appelle coordonnées généralisées la quantité définissant le mouvement relatif du corps (segments) C_i par rapport au corps C_{i-1} , on note la variable articulaire q , elle peut être un angle autour d'un axe (liaison de rotation). L'ensemble q_i , permet de définir les différentes configurations d'un robot manipulateur.

I.4 Modélisation d'un bras manipulateur

Depuis quelques décennies, la recherche dans la robotique s'est concentrée presque entièrement sur la commande des robots manipulateurs. Pour la synthèse d'une commande ou pour simuler le comportement d'un système mécanique articulé (robot), on doit disposer d'un

modèle. Plusieurs niveaux de modélisation sont possibles selon les objectifs, les contraintes de la tâche et les performances recherchées. Les modèles mathématiques sont :

Les modèles géométriques directs et inverses qui expriment les positions de l'organe terminal en fonction des variables articulaires et inversement.

Les modèles cinématiques directs et inverses qui expriment les vitesses de l'organe terminal en fonction des variables articulaires et inversement.

Les modèles dynamiques définissant les équations du mouvement du robot qui permettent d'établir les relations entre les couples ou forces exercées par les actionneurs et les positions, vitesses, accélérations des articulations.[10]

I.4.1 Modélisation géométrique

Le modèle géométrique d'un robot constitue une représentation mathématique en le considérant comme une structure arborescente comportant n corps rigides, sans masse, articulés entre eux. Chaque corps est réduit à sa plus simple expression c'est-à-dire à son orientation et sa position par rapport au précédent.

I.4.2 Modélisation cinématique

Dans le modèle géométrique on prend en considération seulement la position et l'orientation des corps et de l'organe terminal, dans le modèle cinématique il y a apparition des, celles de l'organe terminal et des actionneurs.

I.4.3 Modélisation dynamique

L'utilisation des robots est devenue de plus en plus importante dans le domaine de l'automatisme. Dans les modèles précédents, nous avons supposé que les corps du SMA (système mécanique articulé) sont rigides et l'équilibre statique est réalisé pour toute configuration prise par le robot au cours de son déplacement. Au fur et à mesure que la vitesse augmente, les modèles géométriques et cinématique vont s'éloigner du modèle réel.

Cependant l'ultra rapidité et la haute précision de la trajectoire sont de plus en plus exigées pour atteindre les performances et la perfection désirée du robot. Pour satisfaire les exigences des utilisateurs, il est nécessaire de tenir compte des phénomènes dynamiques (forces inertielles, centrifuges, coriolis et de gravité, Frottement et perturbations), afin d'avoir un modèle plus proche de la réalité. Le comportement dynamique d'un système mécanique articulé est décrit par la variation dans le temps des moments des articulations exercés par les actionneurs

I.5 Détermination du modèle dynamique

Le modèle dynamique d'un SMA exprime la relation entre les variables d'états du mécanisme et les forces extérieures agissant sur chaque corps (effet des moteurs, des frottements et de pesanteur etc.), sous forme d'équations différentielles.

Les équations différentielles qui gouvernent la réponse dynamique d'un robot ont été établies de plusieurs manières, elles peuvent être obtenues en utilisant l'un des formalismes suivants :

- Les équations de Newton-Euler.
- Le formalisme de Lagrange.
- Le principe des travaux virtuels d'Alembert.

On représente le modèle dynamique par une relation de la forme :

$$\tau = f(q, \dot{q}, \ddot{q}) \quad (I.1)$$

Dans le cas d'un bras manipulateur à n degrés de libertés (n d.d.l)

$\tau = [\tau_1, \tau_2, \dots, \tau_n]^T$ est le vecteur des couples/forces des actionneurs, l'articulation est rotoïde (mouvement de rotation) ou prismatique (mouvement de translation).

$q = [q_1, q_2, \dots, q_n]^T$ est le vecteur des positions articulaires.

\dot{q} est le vecteur des vitesses articulaires ($\dot{q} = \frac{dq}{dt}$).

\ddot{q} est le vecteur des accélérations articulaires ($\ddot{q} = \frac{d^2q}{dt^2}$).

f est une fonction vectorielle multivariable fortement non linéaire.

Dans notre travail, nous avons opté pour un bras manipulateur à deux axes et segments rigides évoluant dans le plan vertical. Ce bras est constitué de deux axes de masses respectives m_1 et m_2 et de longueurs respectives L_1 et L_2 comme le montre la figure suivante :

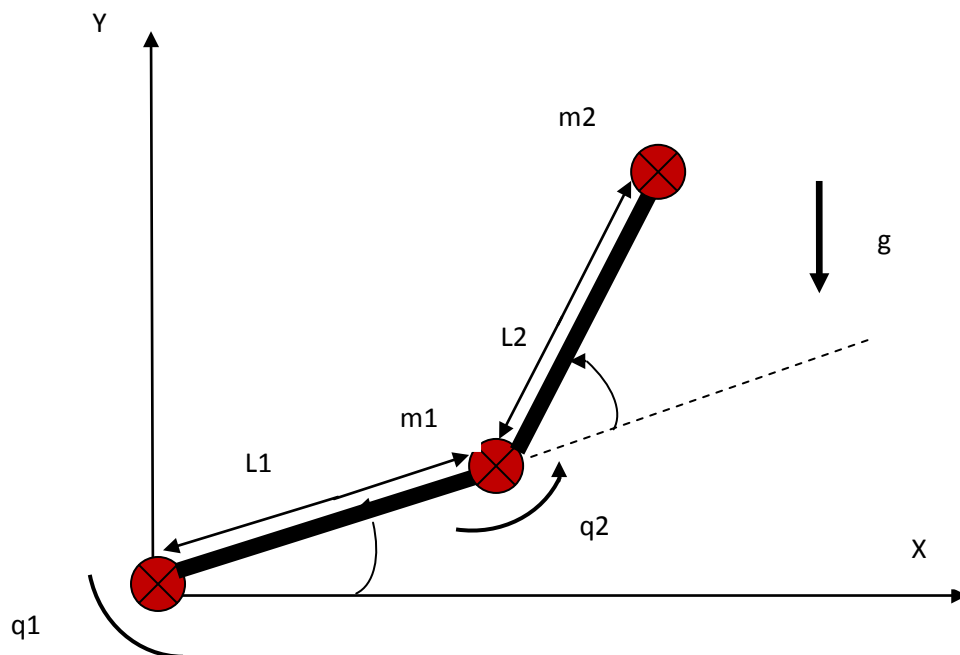


Figure I.3: Robot manipulateur à 2dd

- **Formalisme Euler- Lagrange**

Le formalisme Euler- Lagrange décrit les équations du mouvement en termes de travail et d'énergie du système, il est décrit par :

$$\frac{d}{dt} \left(\frac{\partial l}{\partial \dot{q}_i} \right) - \frac{\partial l}{\partial q_i} = \tau_i \quad (\text{I.2})$$

$$i = 1, \dots, n$$

Avec :

n = nombre de degré de liberté;

L = lagrangien du système égal à $E_c - E_p$;

E_c = énergie cinétique totale du système ;

E_p = énergie potentielle totale du système ;

L'énergie cinétique du système est une fonction quadratique des vitesses articulaires, elle est donnée par :

$$E_c = \sum_{i=1}^n \left(\frac{1}{2} m_i v_{Gi}^2 + \frac{1}{2} I_{Gi} \dot{q}_i^2 \right) \quad (\text{I.3})$$

Où :

m_i , I_{Gi} et v_{Gi} représentent pour le $i^{\text{ème}}$ corps du robot, la masse, le moment d'inertie par rapport au centre de gravité et la vitesse de translation du centre de gravité, respectivement.

Quand à l'énergie potentielle, elle est donnée par :

$$E_p = \sum_{i=1}^n (-m_i g^T p_{Gi}) \quad (\text{I.4})$$

Où :

g , p_{Gi} représentent respectivement, les vecteurs de gravité et de position du centre de gravité du $i^{\text{ème}}$ corps du robot, projetés dans le repère de la base.

Après avoir appliqué le formalisme d'Euler- Lagrange, les équations obtenues (constituant le modèle dynamique de robot) peuvent se mettre sous la forme compacte suivante :

$$M(q)\ddot{q} + V(q, \dot{q}) + G(q) = \tau \quad (\text{I.5})$$

Avec : $V(q, \dot{q}) = V_m(q, \dot{q}) \dot{q}$

Où :

$M(q)$ est une matrice de dimension $n \times n$ symétrique définie positive, appelée matrice d'inertie (inertia matrix),

$V_m(q, \dot{q})$ est le vecteur de dimension $(n \times 1)$ représentant les couples/forces de coriolis et des forces centrifuges,

$G(q)$ est un vecteur de dimension n des couples de gravitation ou couple de maintien, On admettra que les masses des tiges composant le robot sont concentrées en leurs extrémités. Le robot manipulateur est commandé en couples τ_1 et τ_2 fournis par les actionneurs dont on négligera la dynamique.

- **Les coordonnées cartésiennes sont**

$$\text{La } 1^{\text{ère}} \text{ extrémité : } \begin{cases} x_1 = l_1 \cos q_1 \\ y_1 = l_1 \sin q_1 \end{cases} \quad (\text{I.6})$$

$$\text{La } 2^{\text{ème}} \text{ extrémité : } \begin{cases} x_2 = x_1 + l_2 \cos (q_1 + q_2) \\ y_2 = y_1 + l_2 \sin (q_1 + q_2) \end{cases} \quad (\text{I.7})$$

- **Energie cinétique**

L'énergie cinétique du bras manipulateur à 2ddl (figure I.3) est donnée par :

$$E_c = \frac{1}{2} (m_1 + m_2) l_1^2 \dot{q}_1^2 + \frac{1}{2} m_2 l_2^2 (\dot{q}_1 + \dot{q}_2)^2 + m_2 l_1 l_2 \dot{q}_1 (\dot{q}_1 + \dot{q}_2) \cos q_2 \quad (\text{I.8})$$

- **Energie potentielle**

La seule source d'énergie potentielle est la gravitation. L'énergie potentielle du bras manipulateur à 2ddl (figure 3) est donnée comme suit :

$$E_p = (m_1 + m_2)gl_1 \sin q_1 + m_2 gl_2 \sin(q_1 + q_2) \quad (\text{I.9})$$

- **Lagrangien du système**

$$L = \frac{1}{2} (m_1 + m_2) l_1^2 \dot{q}_1^2 + \frac{1}{2} m_2 l_2^2 (\dot{q}_1 + \dot{q}_2)^2 + m_2 l_1 l_2 \dot{q}_1 (\dot{q}_1 + \dot{q}_2) \cos q_2 \\ [(m_1 + m_2)gl_1 \sin q_1 + m_2 gl_2 \sin(q_1 + q_2)] \quad (\text{I.10})$$

L'application de formalisme d'Euler- Lagrange (I.2) sur le système (I.10) amène à l'équation suivante :

$$\begin{cases} \frac{d}{dt} \left(\frac{\partial L}{\partial \dot{q}_1} \right) - \frac{\partial L}{\partial q_1} = \tau_1 \\ \frac{d}{dt} \left(\frac{\partial L}{\partial \dot{q}_2} \right) - \frac{\partial L}{\partial q_2} = \tau_2 \end{cases}$$

Le modèle dynamique du robot manipulateur est donné sous la forme d'équations différentielles non linéaires suivantes :

$$[(m_1 + m_2) l_1^2 + m_2 l_2^2 + 2m_2 l_1 l_2 \cos q_2] \ddot{\theta}_1 + [m_2 l_2^2 + m_2 l_1 l_2 \cos q_2] \ddot{q}_2 - \\ m_2 l_1 l_2 \dot{q}_1 (2\dot{q}_1 + \dot{q}_2) \sin q_2 + (m_1 + m_2) gl_1 \cos q_1 + m_2 gl_2 \cos(q_1 + q_2) = \tau_1 \quad (\text{I.11})$$

$$[m_2 l_2^2 + m_2 l_1 l_2 \cos q_2] \ddot{q}_1 + [m_2 l_2^2] \ddot{q}_2 + m_2 l_1 l_2 \dot{q}_1^2 \sin q_2 + m_2 gl_2 \cos(q_1 + q_2) = \tau_2 \quad (\text{I.12})$$

En réécrivant les équations du système sous forme matricielle, elle devient:

$$\begin{bmatrix} (m_1 + m_2) l_1^2 + m_2 l_2^2 + 2m_2 l_1 l_2 \cos q_2 & m_2 l_2^2 + m_2 l_1 l_2 \cos q_2 \\ m_2 l_2^2 + m_2 l_1 l_2 \cos q_2 & m_2 l_2^2 \end{bmatrix} \begin{bmatrix} \ddot{q}_1 \\ \ddot{q}_2 \end{bmatrix} +$$

$$\begin{bmatrix} -m_2 l_1 l_2 \dot{q}_2 (2\dot{q}_1 + \dot{q}_2) \sin q_2 \\ m_2 l_1 l_2 \dot{q}_1^2 \sin q_2 \end{bmatrix} \begin{bmatrix} \dot{q}_1 \\ \dot{q}_2 \end{bmatrix} +$$

$$\begin{bmatrix} (m_1 + m_2) g l_1 \cos q_1 + m_2 g l_2 \cos(q_1 + q_2) \\ m_2 g l_2 \cos(q_1 + q_2) \end{bmatrix} \begin{bmatrix} q_1 \\ q_2 \end{bmatrix} = \begin{bmatrix} \tau_1 \\ \tau_2 \end{bmatrix} \quad (\text{I.13})$$

On identifie les différents termes de l'équation (I.13) comme suit:

$$M(q) = \begin{bmatrix} (m_1 + m_2) l_1^2 + m_2 l_2^2 + 2m_2 l_1 l_2 \cos q_2 & m_2 l_2^2 + m_2 l_1 l_2 \cos q_2 \\ m_2 l_2^2 + m_2 l_1 l_2 \cos q_2 & m_2 l_2^2 \end{bmatrix} \quad (\text{I.14})$$

$$V(q, \dot{q}) = \begin{bmatrix} -m_2 l_1 l_2 \dot{q}_2 (2\dot{q}_1 + \dot{q}_2) \sin q_2 \\ m_2 l_1 l_2 \dot{q}_1^2 \sin q_2 \end{bmatrix} \quad (\text{I.15})$$

$$V_m(q, \dot{q}) = \begin{bmatrix} -2m_2 l_1 l_2 \dot{q}_1 \sin q_2 & -m_2 l_1 l_2 \dot{q}_2 \sin q_2 \\ m_2 l_1 l_2 \dot{q}_1^2 \sin q_2 & 0 \end{bmatrix} \quad (\text{I.16})$$

$$G(q) = \begin{bmatrix} (m_1 + m_2) g l_1 \cos q_1 + m_2 g l_2 \cos(q_1 + q_2) \\ m_2 g l_2 \cos(q_1 + q_2) \end{bmatrix} \quad (\text{I.17})$$

I.6 Conclusion

Dans ce chapitre nous avons présenté quelques notions sur les robots rigides ainsi qu'une synthèse de différents modèles décrivant leurs dynamiques.

Pour tenir compte des différents phénomènes non linéaires engendrés par la dynamique de la structure mécanique à vitesses élevées, le modèle dynamique est établi à partir de l'équation d'Euler-Lagrange.

II.1. Introduction

La commande par mode glissant est une commande à structure variable VSS (Variable Structure System dans la littérature anglo-saxonne). Les premiers travaux sur ce type de commande ont été menés dans l'ancienne U.R.S.S à partir des années soixante. Elle est basée essentiellement sur la résolution des équations différentielles à second membre discontinu, initiée par le mathématicien soviétique A.G. Filippov.[17]

Le véritable essor de l'utilisation de cette technique date de la parution des livres d'Itkis [18] et d'Utkin [19]. Cette technique de commande a reçu un intérêt sans cesse croissant en raison de sa simplicité d'élaboration et de ses applications dans divers domaines de l'automatique ou de l'électronique de puissance.

La commande par modes glissants a été largement appliquée dans le cas des systèmes linéaires et non linéaires et a été étendue avec un certain succès sur les systèmes d'ordre entier à retards, et récemment, sur les systèmes fractionnaires (d'ordre non entier) avec et sans retards.

Dans ce chapitre, nous présenterons les concepts classiques utilisés en régime glissant d'ordre un, nécessaires à la compréhension de cette technique de commande.

II.2. Principe de fonctionnement

Le principe de la commande par modes glissants est de contraindre le système à atteindre une surface donnée appelée surface de glissement en fonction des objectifs de commande fixant la dynamique en boucle fermée : c'est le mode de convergence, puis par la synthèse d'une commande discontinue qui permet aux trajectoires du système à atteindre et, ensuite, à rester sur cette surface : c'est le mode de glissement.[12]

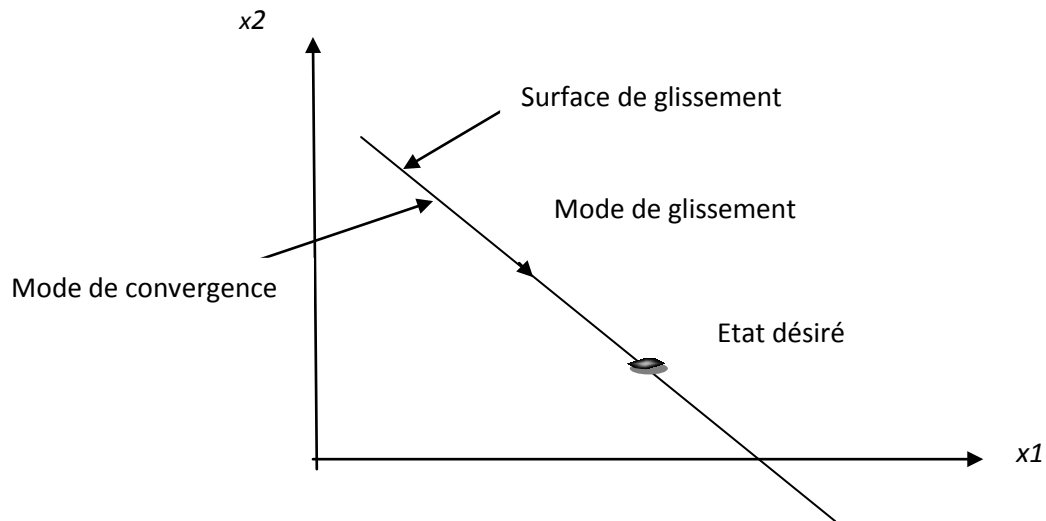


Figure II.1 : Différents mode de convergence pour la trajectoire d'état

II.3. Définition du régime glissant

Considérons le système dynamique suivant :

$$X^{(n)} = f(x) + g(x)u \quad (\text{II.1})$$

Où

X est la sortie du système.

x est le vecteur d'état,

$$x = [x \ \dot{x} \ \dots \ x^{(n-1)}]^T$$

U est le vecteur de commande, dont l'expression est définie par :

$$u_i(x, t) = \begin{cases} u_i^+(x, t), & \text{si } s_i(x) > 0 \\ u_i^-(x, t), & \text{si } s_i(x) < 0 \end{cases} \quad i = 1, \dots, m \quad (\text{II.2})$$

u^+ et u^- sont des fonctions continues.

f est une fonction généralement non linéaire et non exactement connue.

g est une fonction de commande non exactement connue.

Nous voulons faire suivre au vecteur d'état x une trajectoire désirée, définie par le vecteur désiré $x_d(t)$. Pour cela, on pose le vecteur d'erreur :

$$\tilde{x} = x - x_d = [\tilde{x} \quad \dot{\tilde{x}} \quad \dots \quad \tilde{x}^{(n-1)}]^T$$

Par ailleurs, nous allons définir une surface de glissement dans l'espace d'état des erreurs $s(x;t) = 0$,

Le problème de suivre $x = x_d$ est le même que celui de rester sur la surface $s(x; t)$ pour tout le temps $t > 0$. En effet $s \equiv 0$ représente une équation, dont la solution unique est $e(x) = 0$. De cette façon le problème de suivre le vecteur désiré x_d est réduit à retenir s à zéro.

Après la détermination de la surface de glissement, l'étape suivante est de garantir l'existence d'un régime glissant. Il existe un régime glissant, si dans le voisinage de la surface de glissement $s = 0$, les vecteurs de vitesse des trajectoires du système pointent toujours vers la surface. Où d'une autre manière, la distance à la surface de glissement et sa vitesse de changement \dot{s} ont un signe opposé : [20]

$$\lim_{s \rightarrow 0^+} \dot{s} < 0 \quad \text{et} \quad \lim_{s \rightarrow 0^-} \dot{s} > 0 \quad (\text{II.3})$$

L'expression précédente peut être écrite de façon simplifiée :

$$s\dot{s} < 0 \quad (\text{II.4})$$

- **Méthode de la commande équivalente**

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

$$s(x, t) = 0 \quad \text{et} \quad \dot{s}(x, t) = 0 \quad (\text{II.5})$$

La condition (II.5) est appelée condition d'invariance de la surface de glissement pour la suite de notre travail, nous prenons $s(x, t) = s(x(t)) = s(t)$

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

u_{eq} , appelée commande équivalente, est associée au système nominal, elle est déterminée de façon unique par la condition d'invariance (29). En supposant que $\frac{\partial s}{\partial x} g(x) \neq 0$

La commande équivalente est donnée par le scalaire : $u_{eq} = - \left(\frac{\partial s}{\partial x} g(x) \right)^{-1} \frac{\partial s}{\partial x} f(x)$ (II.6)

Physiquement, la commande équivalente représente la valeur moyenne de la commande u qui maintient l'état du système de glissement $s(x)=0$.

L'équation du régime glissant idéal est obtenue en portant l'expression de u_{eq} dans (II.1).

$$\dot{x} = f(x) - g(x) \left(\frac{\partial s}{\partial x} g(x) \right)^{-1} \frac{\partial s}{\partial x} f(x) \quad (\text{II.7})$$

$\frac{\partial s}{\partial x}$ étant non nulle sur x .

II.4 Propriétés de la convergence en temps fini

Cette condition est en fait la condition sous laquelle le mode de glissement existe et sous laquelle la trajectoire d'état va effectivement atteindre la surface de glissement en un temps fini. Soit, la fonction de Lyapunov suivante :

$$V(x) = \frac{1}{2} s^2(x) \quad (\text{II.8})$$

La dérivée de cette fonction est : $\dot{V}(x) = \dot{s}(x)s(x)$ (II.9)

La condition qui garantit la convergence de l'état vers la surface de glissement est donnée par $\dot{V}(x) = \dot{s}(x) s(x) < 0$

Cette équation explique que le carré de la distance vers la surface de glissement mesurée par $s^2(x)$, diminue tout le temps, contraignant les trajectoires du système à se diriger vers la surface des deux cotés.

II.5 Choix de la surface de glissement

Le choix de la surface de glissement concerne le nombre et la forme des fonctions nécessaires. Ces deux facteurs dépendent de l'application et de l'objectif visé.

Pour un système défini par l'équation (II.20), le vecteur de surface s a la même dimension que le vecteur de commande u .

$$\dot{x} = f(x, t)x + g(x, t)u \quad (\text{II.20})$$

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

La forme non linéaire est une fonction de l'erreur sur la variable à réguler x , elle est donnée par :

$$s(x) = \left(\frac{\partial}{\partial t} + \lambda \right)^{r-1} e(x) \quad (\text{II.21})$$

Avec :

$e(x)$: L'écart entre la variable à régler et sa référence.

λ : Une constante positive.

r : Le degré relatif, il représente le nombre de fois qu'il faut dériver la surface pour faire apparaître la commande.

L'expression de la surface est une équation différentielle dont l'unique solution est $e(x)=0$.

L'objectif de la commande est de maintenir la surface à zéro.[8]

II.6 Propriétés de la robustesse

Considérons le système perturbé suivant :

$$\dot{x} = f(x) + g(x)u + p \quad (\text{II.22})$$

Où p représente l'effet des incertitudes paramétriques sur le terme $f(x)$ ou des perturbations externes indépendantes de l'état.

Théorème : Le régime glissant sur s , du système perturbé (II.22), est invariant vis-à-vis de p , si et seulement si le vecteur de perturbation p est engendré par $g(x)$. Cette condition est appelée condition de recouvrement où « matching condition ».

Notons que le système est insensible à de telles perturbations seulement en régime glissant. [12]

II.7 Le phénomène du chattering

En pratique, la commande discontinue peut exciter les dynamiques de hautes fréquences non modélisées, qui entraînent l'apparition de ce qu'on appelle «réticence» où «broutement» connu en anglais sous le nom «chattering» et se caractérise par de fortes oscillations autour de la surface.

Ce phénomène de réticence apparait car la commutation de la commande ne se fait pas à une fréquence infinie à cause des imperfections physique (les retards, limitation

physique des actionneurs ...etc.). Pendant les premières années de son apparition la commande par modes glissants a été entravée par ce phénomène qui peut provoquer une détérioration anticipée de l'organe de commande, augmenter la consommation de l'énergie engendrant l'instabilité du système en excitant des dynamiques hautes fréquences non considérées dans la modélisation.

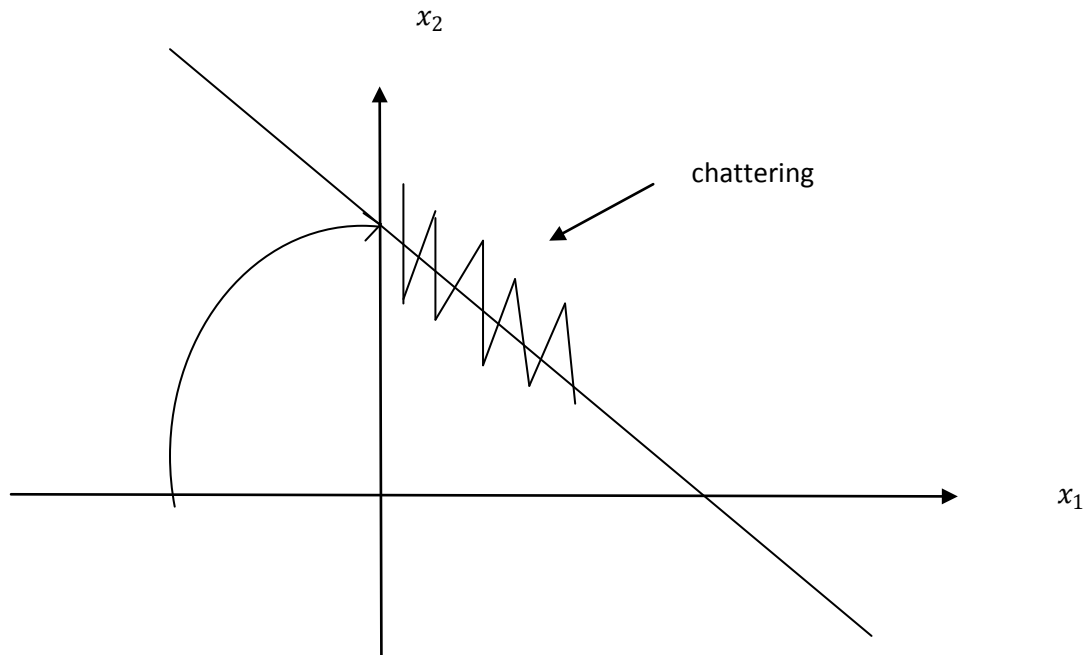


Figure II.2 : Phénomène de chattering

II.8 Solution pour la suppression de chattering

- **Solution de couche limite :**

Cette solution est connue sous le nom « boundary layer ». Elle est basée sur la définition d'une zone autour de la surface, à l'intérieur de laquelle une condition de glissement moins stricte que la condition signe est appliquée. La fonction $sign(s(t))$ dans la loi de commande est remplacée par une fonction plus lisse dans la couche limite de $s(t) = 0$. Parmi ces fonctions utilisées, nous citons la fonction de saturation.

- **Solution par des lois d'approche**

Par ailleurs, une approche intéressante consiste à concevoir des lois de convergence non linéaires qui permettent une adaptation dynamique d'une composante de la commande en fonction de la variation de la fonction de glissement. Ainsi, plus le vecteur d'état est loin de la

surface de glissement, plus cette composante est grande et tend à ramener le vecteur vers la surface, et inversement. Par conséquent, il est théoriquement possible de réduire les commutations hautes fréquences en régime établi, sans toutefois affecter le temps de convergence ni même l'erreur de poursuite du système. Gao et Hung ont proposé notamment deux lois d'atteinte possibles pour réduire les commutations. La première loi d'atteinte contient un terme proportionnel à la fonction de glissement qui permet au système d'atteindre plus rapidement la surface de glissement lorsque celui-ci en est éloigné. De plus, le terme proportionnel allège le travail de la partie discontinue $\text{sign}(s)$ et permet par conséquent une réduction des commutations sur la commande.

Cette loi d'atteinte est donnée par :

$$\dot{s} = -\alpha \cdot \text{sign}(s) - Q \cdot s \quad (\text{II.23})$$

Dans la deuxième loi de convergence figure une puissance fractionnaire de la fonction de glissement qui multiplie le signe de celle-ci, comme suit :

$$\dot{s} = -\alpha |s|^p \text{sign}(s) \quad (\text{II.24})$$

Où p est un réel strictement compris entre 0 et 1 il est par ailleurs démontré par Gao et Hung que la loi de convergence (II.24) engendre un temps d'atteinte fini de la surface de glissement.[13]

- **Solution par des modes glissants d'ordre supérieur :**

Les modes glissants d'ordre supérieur ont été introduits pour pallier au problème du chattering tout en gardant les propriétés de convergence en temps fini et de robustesse des commandes par modes glissants classiques. Dans cette approche, le terme discontinu n'apparaît plus directement dans l'expression de la commande synthétisée mais dans une de ses dérivées supérieures ce qui a le mérite de réduire le Chattering.[14][15][16]

II.8 Conclusion

Dans ce chapitre, nous avons présenté d'une manière générale les concepts de la théorie des systèmes à structure variable et les modes glissants associés. La caractéristique principale de ces systèmes est que leur loi de commande change d'une manière discontinue selon une logique de commutation déterminée. La commutation de la commande s'effectue selon le signe de la surface de glissement afin de générer la commande d'ouverture ou de fermeture, dont le but est de forcer la dynamique du système à atteindre l'état désiré en un temps fini.

Nous avons aussi présenté les concepts de base de la commande par modes glissants, à savoir : choix de la surface de glissement, conditions d'existence et de convergence et quelques solutions pour réduire le phénomène de chattering.

III.1 Introduction

L'utilisation des systèmes flous et des réseaux de neurones pour la synthèse des systèmes de commande a connu un essor important au cours de ces dernières années.

La Logique floue et les réseaux de neurones appartiennent à la famille de l'intelligence artificielle (IA), une discipline majeure du génie informatique.

Le chapitre est organisé en trois parties, La première partie fait l'objet d'une étude détaillée sur l'emploi des systèmes flous. Après une présentation générale de quelques notions sur les ensembles flous, nous présenterons en détail la structure interne d'un contrôleur flou. Ce dernier est considéré comme l'une des applications les plus importantes des systèmes flous.

La deuxième partie sera consacrée aux réseaux de neurones pour la commande. Après une brève présentation de quelques notions générales sur les réseaux de neurones, Nous aborderons également le problème d'apprentissage des paramètres de ces structures, nous nous intéresserons particulièrement à l'algorithme de rétropropagation. Ensuite, dans la troisième partie de ce chapitre nous présenterons l'interaction entre la logique floue et les réseaux de neurones pour former les systèmes neuro-flous.

III.2 Les systèmes d'inférence floue

III.2.1 Introduction

Les systèmes d'inférence floue (SIF) peuvent être considérés comme des systèmes logiques qui utilisent des règles linguistiques pour établir des relations entre leurs variables d'entrée et de sortie. Aujourd'hui, les applications des SIF sont très nombreuses outre la commande, ils sont largement utilisés pour la commande, la modélisation, le diagnostic et la reconnaissance de formes. Pour une meilleure compréhension, nous présentons quelques notions de base de ces systèmes ainsi que leurs types et leurs caractéristiques.

III.2.2 Historique

1965 : Le Prof. L. A. Zadeh de l'Université de Berkeley (Californie) pose les bases théoriques de la logique floue.

1973 : L. A. Zadeh propose d'appliquer la logique floue aux problèmes de réglage.

1974 : Première application du réglage par la logique floue appliquée à une turbine à vapeur. Suivie en 1980 par une application sur un four à ciment, et en 1983 sur un épurateur d'eau.

1985 : Premiers produits industriels (Japon) utilisant le principe de la logique floue appliqué à des problèmes de réglage et de commande.

Développement de processeurs dédiés à des applications de réglage par la logique floue.[21]

III.2.3 Principe de la logique floue

La logique floue est une branche des mathématiques et, à ce titre, toute une série de notions fondamentales sont développées. Ces notions permettent de justifier et de démontrer certains principes de base. Dans ce qui suit, on ne retiendra que les éléments indispensables à la compréhension du principe du réglage par la logique floue. Ces éléments sont :[22]

Les variables linguistiques

Les règles d'inférences

- **Les variables linguistiques**

La notion de variable linguistique permet de modéliser les connaissances imprécises ou vagues sur une variable dont la valeur précise est inconnue. Une variable linguistique, ou variable floue, est donc une variable dont les valeurs floues appartiennent à des ensembles flous pouvant représenter des mots du langage naturel par exemple : «petite , moyenne, grande ».

- **Les règles d'inférences**

On appelle règles d'inférence l'ensemble des différentes règles reliant les variables floues d'entrée d'un système aux variables floues de sortie de ce système. Ces règles se présentent sous la forme :

Si condition 1 et/ou condition 2 (et/ou...) alors action sur les sorties

Si condition 3 et/ou condition 4 (et/ou...) alors action sur les sorties

.

.

Si condition n et/ou condition n+1(et/ou...) alors action sur les sorties [23]

III.2.4 Les bases de la logique floue

III.2.4.1 Les Fonctions d'appartenance

Un ensemble flou est défini par sa « fonction d'appartenance », qui correspond à la notion de « fonction caractéristique » en logique classique, et fonction d'appartenance en logique floue. Il permet de définir le degré de vérité de cette variable floue en fonction de sa grandeur d'entrée.

On peut évidemment choisir n'importe quelle forme pour les fonctions d'appartenance. Cependant, en pratique, on utilise les formes trapézoïdales . [24]

III.2.4.2 Univers de discours

Pour décrire un paramètre (exemple température), plusieurs mots ou prédicats peuvent être utilisés : chaud, froid, tiède. Chacune de ces prédicats possède une fonction d'appartenance. Elle couvre l'ensemble des valeurs prises par une variable.[25]

III.2.4.3 Les opérateurs

Ces règles permettent de relier les variables floues d'entrée aux variables floues de sortie à l'aide de différents opérateurs qui sont ET, OU et NON, qui s'appliquent aux variables floues. Dans le cas de la logique binaire ces opérateurs sont définis de façon simple et univoque. Dans le cas de la logique floue, la définition de ces opérateurs n'est plus univoque et on utilise le plus souvent les relations présentées dans le tableau suivant :

Opérateur	Opérateur sur le degré de vérité des variables
ET	minimum
	produit
OU	Maximum
	valeur moyenne
NON	Complément à 1

Tableau III.1 : les opérateurs en logique floue

Les opérations minimum et maximum présentent l'avantage de la simplicité lors du calcul, par contre, elles privilégient l'une des deux variables. Les opérations de produit et valeurs moyennes sont plus complexes à calculer mais elles produisent un résultat qui tient compte des valeurs des deux variables.

III.2.5 Structure interne d'un système d'inférence flou

Un système flou est un système à base de connaissance particulières, dont l'architecture de base, illustrée par la figure (III.2) se compose essentiellement de quatre modules.

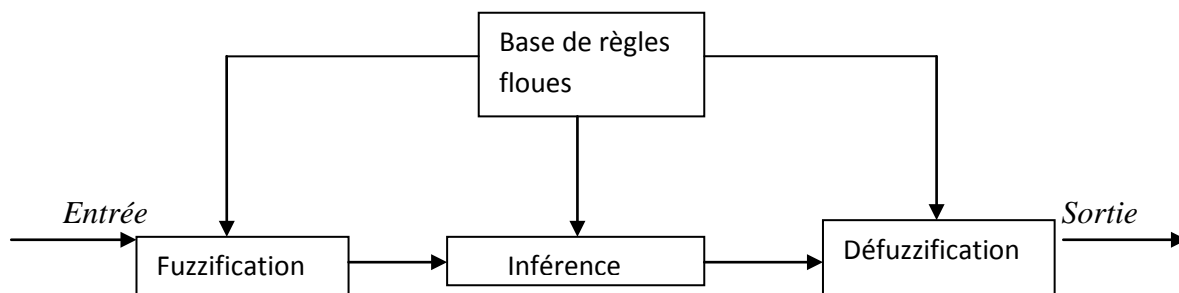


Figure III.2 : Structure interne d'un système flou

III.2.5.1 Fuzzification

C'est le mécanisme réalisant l'interface " Numérique – Linguistique " qui permet de transformer une grandeur générée par un capteur à l'entrée en partie floue définie sur un espace de représentation lié à l'entrée. Du choix de cet espace de représentation (numérique ou linguistique) dépend la nature de la partie E résultante de la partie fuzzification.

Cette dernière peut être un sous ensemble numérique (élément de $F(x)$) ou un sous ensemble flou symbolique (élément $F(L(x))$). On distingue alors deux types de fuzzification respectivement appelés fuzzification numérique et fuzzification symbolique.[26]

III.2.5.1.1 Fuzzification numérique

Une fuzzification numérique transforme toute valeur numérique x de l'entrée en valeur floue numérique, c'est-à-dire un élément de $F(x)$, en utilisant la fuzzification singleton ou non singleton.

- **La fuzzification singleton :** cette démarche est utilisée lorsque la valeur fournie par le capteur est supposée précise, sa valeur est $x = x_0$, la partie floue numérique $N(x_0)$ est définie par la fonction d'appartenance suivante [25]

$$\mu_{(x_0)}(x) = \begin{cases} 1 & \text{Si } x = x_0 \\ 0 & \text{Si } x \neq x_0 \end{cases} \quad (\text{III.1})$$

- **La fuzzification non singleton** : elle est utilisée lorsque la valeur délivrée par le capteur est imprécise. La fonction d'appartenance de la partie floue numérique $N(x_0)$ peut être définie par une fonction gaussienne.

$$\mu_{N(x_0)}(X) = \exp\left(-\frac{1}{2}\left(\frac{x-x_0}{\beta}\right)^2\right) \quad (\text{III.2})$$

Où β est un paramètre qui caractérise la largeur de la fonction d'appartenance.[25]

III.2.5. Fuzzification symbolique

La fuzzification symbolique transforme la valeur d'entrée x délivrée par le capteur en une partie floue linguistique. La description de la mesure s'effectue dans l'ensemble $L(X)$ des valeurs linguistiques des prémisses. Par exemple, on peut utiliser l'ensemble $L(x) = \{N, Z, P\}$, où N est l'abréviation de négatif, Z celle de zéro et P celle de positif.[26]

III.2-5-2-Base des règles floues

C'est une base de connaissances qui contient les règles floues décrivant le comportement du système. L'ensemble des règles se présente sous la forme d'une énumération de type:

Si condition 1 et/ou condition 2 (et/ou...) alors action sur les sortie

Si condition 3 et/ou condition 4 (et/ou...) alors action sur les sortie

Si condition 5 et/ou condition 6 (et/ou...) alors action sur les sortie

III.2-5-3-Inférence Floue

L'inférence floue est le processus de formulation de la relation entre les entrées et les sorties par logique floue. Cette relation offre une base avec laquelle la décision est prise par le système flou. L'inférence floue fait appel alors aux concepts de fonctions d'appartenance, d'implication floue et les règles floues. Il transforme à l'aide des technique de raisonnement flou, la partie floue issue de la fuzzification en une nouvelle partie floue, afin d'évaluer le degré de vérité d'une règle i à partir des fonctions d'appartenance liées à la prémisse et à la conclusion. L'agrégation des n règles s'effectue en combinant les différentes règles floue à l'aide de l'opérateur OU.

les méthodes d'inférence permettent la réalisation des différents opérateurs ET ,OU , ALORS, intervenant dans les règles d'inférence et s'appliquant aux fonction d'appartenance .

Trois méthodes sont généralement utilisées :

- **Méthode d'inférence max-min**

L'implication floue et l'opérateur ET sont réalisés par la fonction minimum et l'opérateur logique OU est réalisé par la fonction maximum. Elle présente l'inconvénient de nécessiter un temps de calcul trop élevé qui la rend inutilisable pour les applications en temps réel. Cette méthode est également dite « implication de Mamdani ».

La fonction d'appartenance résultante correspond au maximum des deux fonctions d'appartenance partielles puisque les règles sont liées par l'opérateur OU.[27]

- **Méthode d'inférence max-prod**

La seule différence avec la méthode précédente est la réalisation de l'implication floue, le produit est utilisé. Les opérateurs logiques ET et OU sont toujours réalisés respectivement par les fonctions minimum et maximum. La méthode max-prod est également dite implication de Larsen. [27]

- **Méthode d'inférence somme-prod**

La méthode d'inférence somme-prod réalise, au niveau de la condition, l'opérateur OU par la formation somme, et l'opérateur ET par la formation produit. Au niveau de la conclusion, elle réalise également l'opérateur ALORS par la formation produit .[27]

III.2.5.4 Défuzzification

A la fin de l'inférence, l'ensemble flou de sortie est déterminé mais il n'est pas directement utilisable pour donner une information précise à l'opérateur ou commander un actionneur. Il est nécessaire de passer du « monde flou » au « monde réel », c'est la défuzzification[28]

III.2.5.4.1 Défuzzification numérique

La défuzzification transforme la partie floue issue de l'inférence en valeur numérique.

On trouve trois méthodes qui sont généralement utilisées :

- **Méthode de maximum**

Cette méthode consiste à choisir comme valeur de sortie celle correspondant à l'abscisse du maximum de la fonction d'appartenance, Cette méthode est simple, rapide et facile à réaliser mais introduit des ambiguïtés dans certains cas.

Dans le cas où plusieurs sous ensembles auraient la même hauteur, on réalise leur moyenne. Cette méthode est appelée « moyenne de maximas »(on prend la moyenne des abscisses du maximum).[29]

- **La méthode de barycentre**

Cette méthode est la plus couteuse en temps de calcul mais la plus utilisée, elle consiste à prendre comme valeur de sortie l'abscisse du centre de gravité de la surface de la fonction d'appartenance résultante.[29]

- **La méthode de la somme pondérée**

Cette méthode est un compromis entre la méthode du maximum et la méthode du barycentre. On calcul individuellement les sorties relatives à chaque règle selon le principe de la moyenne des maximas. On réalise ensuite la moyenne pondérée de toutes ces sorties relatives.[29]

III.2.5.4.2 Défuzzification symbolique

La défuzzification symbolique transforme un sous ensemble flou symbolique en une valeur numérique.

III.2.6 Les modèles d'inférence flous

III.2.6.1 Modèle de MAMDANI

Dans la méthode de Mamdani, les prémisses et les conclusions des règles sont floues. L'inférence floue est réalisée à l'aide de l'opérateur min (minimum) ou prod (produit) et l'agrégation des règles à l'aide de l'opérateur max (maximum). La défuzzification est généralement effectuée par la méthode du centre de gravité [30]

III.2.6.2 Modèle de TAKAGI-SUGENO

La méthode de TAKAGI-SUGENO, est très efficace du point de vue de calcul car elle privilégie l'analyse mathématique, elle fonctionne bien avec les techniques linéaires. L'inférence floue est réalisée avec l'opérateur min ou prod. Les conclusions des règles sont des valeurs numériques calculées à partir des entrées et la valeur finale est obtenue en effectuant une moyenne pondérée des conclusions [31]

III.2.6.3 Modèle TSUKAMOTO

Dans la méthode de TSUKAMOTO, la fonction d'appartenance de chaque conclusion est une fonction monotone croissante. L'inférence floue est réalisée par l'opérateur min ou prod. La sortie numérique est une moyenne pondérée des valeurs issues de la fonction monotone.[32]

III.3 Les réseaux de neurones artificiels

Les réseaux de neurones ont d'abord été développés pour résoudre des problèmes de contrôle, de reconnaissance de formes ou de mots, de décision, de mémorisation comme une alternative à l'intelligence artificielle, et en relation plus ou moins étroite avec la modélisation de processus cognitifs (capable de connaître ou faire connaître) réels et des réseaux de neurones biologiques.

III.3.1 Introduction

Les réseaux de neurones sont composés d'éléments simples (ou neurones) fonctionnant en parallèle. Ces éléments ont été fortement inspirés par le système nerveux biologique. Comme dans la nature, le fonctionnement du réseau de neurones est fortement influencé par la connexion des éléments entre eux. On peut entraîner un réseau de neurones pour une tâche spécifique (reconnaissance de caractères par exemple) en ajustant les valeurs des connections (ou poids) entre les éléments (neurones).

III.3.2 Historique

En 1943 : les premiers travaux datent de 1943 et sont l'œuvre de Mac Culloch et Pitts qui proposent un modèle simple de neurone formel qui est une abstraction du neurone physiologique.

En 1949, D. Hebb présente dans son ouvrage "The Organisation of Behavior" une règle d'apprentissage. De nombreux modèles de réseaux aujourd'hui s'inspirent encore de la règle de Hebb. En 1958, F. Rosenblatt développe le modèle du Perceptron. C'est un réseau de neurones inspiré du système visuel. Il possède deux couches de neurones : une couche de perception et une couche liée à la prise de décision. C'est le premier système artificiel capable

d'apprendre par expérience. Dans la même période, L'Adaline (ADaptive LINar Element) a été présenté par B.Widrow, chercheur américain à Sanford.

En 1969, M.Minsky et S.Papert publient une critique des propriétés du Perceptron. Cela va avoir une grande incidence sur la recherche dans ce domaine. Elle va fortement diminuer jusqu'en 1972, où T.Kohonen présente ses travaux sur les mémoires associatives et propose des applications à la reconnaissance de formes.

C'est en 1982 que J. Hopfield présente son étude d'un réseau complètement rebouclé, dont il analyse la dynamique.

Aujourd'hui, les réseaux neuronaux sont utilisés dans de nombreux domaines (entre autres, vie artificielle et intelligence artificielle) à cause de leur propriété en particulier, leur capacité d'apprentissage, et qu'ils soient des systèmes dynamiques.[33]

III.3.3 Principes de base des RNA

Les RNA s'inspirent du cerveau humain. Les neurones qui constituent les centres de décisions sont reliés de façon à accomplir un travail bien défini. Chaque neurone reçoit des stimuli qui entraînent la prise d'une décision en relation avec le processus qui lui est propre.

Les RNA constituent une méthode de programmation qui consiste à établir les paramètres de chaque neurone ainsi que les relations entre chacun d'eux. En établissant les paramètres optimums, nous pouvons faire l'approximation d'une fonction de façon précise. Les réseaux de neurones représentent une base algorithmique puissante pour le traitement de signaux grâce à leurs propriétés, principalement la non-linéarité et le parallélisme intrinsèque à la méthode.[34]

III.3.4 Le neurone biologique

Le neurone biologique est une cellule vivante spécialisée dans le traitement des signaux électriques. Les neurones sont reliés entre eux par des liaisons appelées axones.

Ces axones vont eux-mêmes jouer un rôle important dans le comportement logique de l'ensemble. Ces axones conduisent les signaux électriques de la sortie d'un neurone vers l'entrée (synapse) d'un autre neurone. Les neurones font une sommation des signaux reçus en entrée et en fonction du résultat obtenu vont fournir un courant en sortie (Figure III.2).

La structure d'un neurone se compose de trois parties :

La somma : ou cellule d'activité nerveuse au centre du neurone.

L'axone : attaché au somma qui est électriquement actif, ce dernier conduit l'impulsion conduite par le neurone.

Dendrites : électriquement passives, elles reçoivent les impulsions d'autres neurones[35]

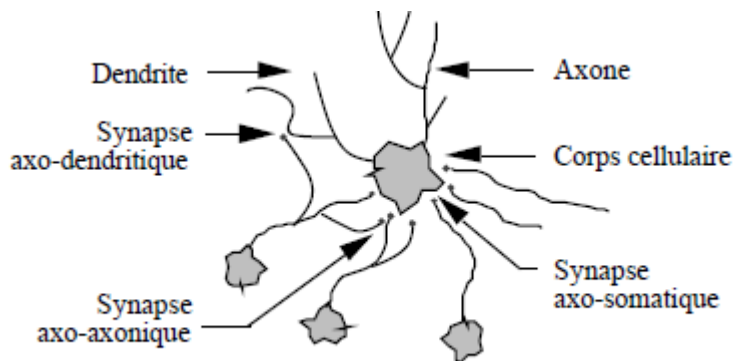


Figure III.2 : Un neurone avec son arborisation dendritique

III.3.5 Le neurone formel

La modélisation des neurones biologiques par des neurones formels a été faite par Mac Culoch et Pitts dans les années quarante.

Le neurone formel reçoit et émet des signaux binaires (0,1), la somme pondérée de toute ses entrées est comparée à un seuil θ ; si ce seuil est dépassé, le neurone s'active, si non il ne transmet aucun signal [36]

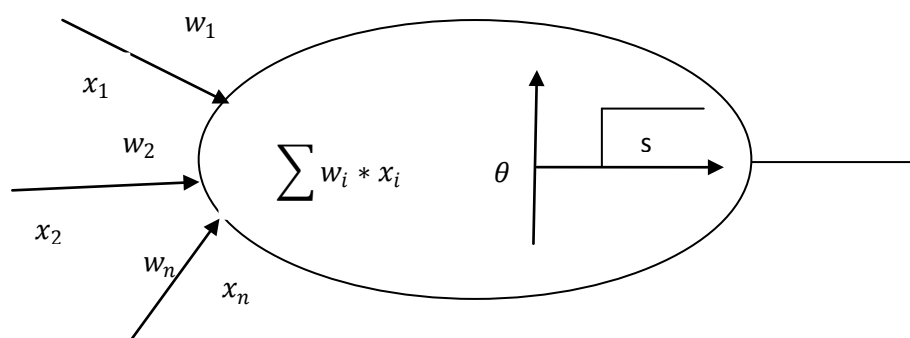


Figure III.3 : neurone artificiel

La somme pondérée des signaux d'entrée appelés (stimuli) constituant l'activation du neurone, se transforme en sortie après son passage par une fonction de seuillage ou de transfert.

$$s = f(w_i x_i) = \begin{cases} 1 & \text{si } w_i x_i < 0 \\ 0 & \text{si } w_i x_i \geq 0 \end{cases} \quad (\text{III.3})$$

III.3.6 Fonction d'activation

Cette fonction permet de définir l'état interne du neurone en fonction de son entrée totale, citons à titre d'exemple quelques fonctions souvent utilisées :

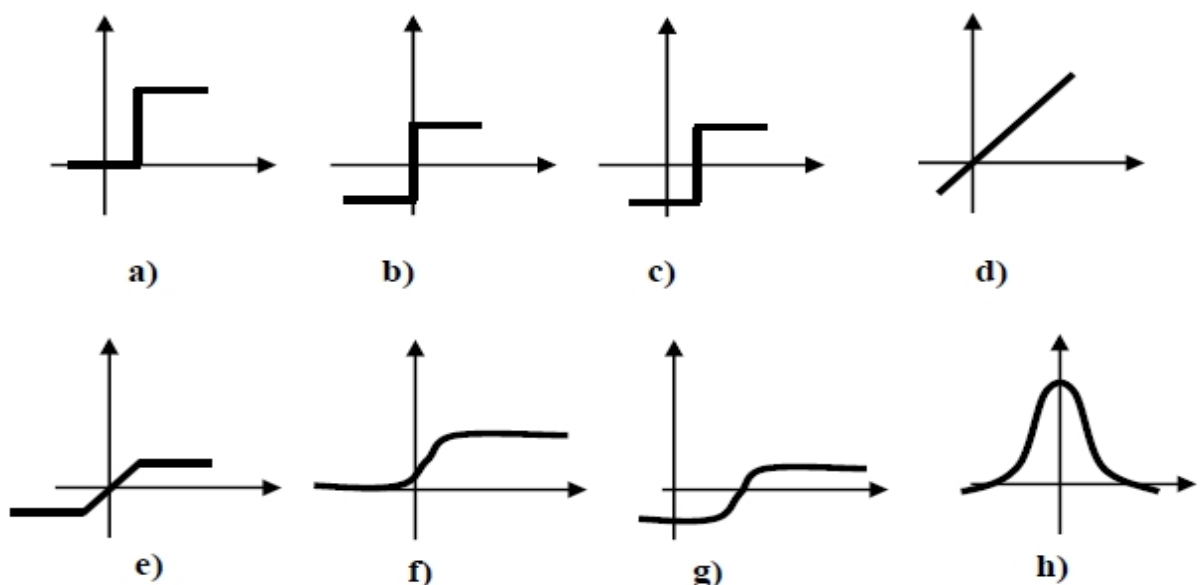


Figure III.4 : fonctions d'activation les plus utilisées

Avec :

- | | |
|--------------------------|--------------------------|
| a) Tout ou rien | e) Saturation |
| b) Fonction signe | f) Sigmoide |
| c) Plus au moins à seuil | g) Fonction arc tangente |
| d) Fonction affine | h) Fonction gaussienne |

III.3.7 Différents types de réseaux de neurones

III.3.7.1 Les réseaux de neurones non bouclés

Un réseau de neurones non-bouclé est représenté comme un graphe dont les nœuds sont les neurones. L'information circule des entrées vers les sorties sans retour en arrière. Si

l'on se déplace dans le réseau à partir d'un neurone quelconque en suivant les connexions, on ne peut pas revenir au neurone de départ.

- **Le perceptron monocouche**

C'est un réseau simple, il ne contient qu'une couche d'entrée et une couche de sortie. Sa principale limite est qu'il ne peut résoudre que des problèmes linéairement séparables.

- **Le perceptron multicouche**

Ce type de réseaux proactif se caractérise par la présence d'une ou de plusieurs couches cachées, dont les nœuds de calcul correspondants s'appellent neurones cachés. Les couches cachées s'interposent entre l'entrée du réseau et sa sortie. Leurs rôle est d'effectuer un prétraitement des signaux d'entrée, reçus par la couche d'entrée en provenance du milieu extérieur, et de transmettre les résultats correspondant à la couche de sortie ou seront déterminées les réponses finales du réseau. Ce rôle de prétraitement fait que, en ajoutant une ou plusieurs couches cachées, le réseau est capable d'extraire plus de propriétés statistiques. Ce réseau est dit complètement connecté dans le sens que chaque nœud d'une couche est connecté à tous les nœuds de la couche voisine.[37]

III.3.7.2 Les réseaux de neurones bouclés

Les réseaux de neurones récurrents (récurrent networks) possèdent une structure similaire à celle des réseaux unidirectionnels mais complétée par des connexions entre éléments de la même couche ou vers des couches amont. Ces réseaux sont assez puissants, car leur fonctionnement est séquentiel et adopte le comportement dynamique. On distingue les réseaux de Hopfield, les réseaux de Kohonen et les réseaux ART (Adaptative Resonance Theory).[38]

- **Modèle de Hopfield**

Le modèle de Hopfield fut présenté en 1982. Ce modèle très simple est basé sur le principe des mémoires associatives. C'est d'ailleurs la raison pour laquelle ce type de réseau est dit associatif (par analogie avec le pointeur qui permet de récupérer le contenu d'une case mémoire). Le modèle de Hopfield utilise l'architecture des réseaux entièrement connectés et récurrents (dont les connexions sont non orientées et où chaque neurone n'agit pas sur lui-même). Les sorties sont en fonction des entrées et du dernier état pris par le réseau.[34]

- **Modèle Kohonen**

Ce modèle a été présenté par Kohonen en 1982 en se basant sur des constatations biologiques. Il a pour objectif de présenter des données complexes et appartenant généralement à un espace discret de grandes dimensions dont la topologie est limitée à une ou deux dimensions. Les cartes de Kohonen sont réalisées à partir d'un réseau à deux couches, une en entrée et une en sortie. Les neurones de la couche de sortie sont placés dans un espace d'une ou de deux dimensions en général, chaque neurone possède donc des voisins dans cet espace. Et qu'enfin, chaque neurone de la couche de sortie possède des connexions latérales récurrentes dans sa couche (le neurone inhibe les neurones éloignés et laisse agir les neurones voisins).[39]

- **Modèle ART**

Les réseaux ART (Adaptive Resonance Theory) sont basés sur une théorie d'inspiration biologique assez complexe sur laquelle nous n'insisterons pas. Elles se manifestent par différentes implantations spécifiques dont celles nommées. «ART1», «ART2», «ART3», «fuzzy ART», Le fuzzy ART est un réseau compétitif à deux couches de neurones : La première couche, sert à coder les stimuli d'entrée avec un encodage dit «Complémentaire». La deuxième couche est une couche compétitive semblable à celle du Kohonen.[40]

III.3.8 L'apprentissage des réseaux de neurones

L'apprentissage dans le contexte des réseaux de neurones, est le processus de modification des poids de connexions ou plus rarement du nombre de couches et de neurones (Man et Halang, 1997), afin d'adapter le traitement effectué par le réseau à une tâche particulière. On distingue deux familles d'apprentissage :

III.3.8.1 L'apprentissage non supervisé

Dans cet apprentissage, les données ne contiennent pas d'information sur la sortie désirée. Il n'y a pas de superviseur. La tâche du réseau consiste, par exemple dans ce cas, à créer des regroupements de données selon des propriétés communes (classification).[41]

III.3.8.2 L'apprentissage supervisé

Dans ce cas, un superviseur (ou expert humain) fournit une valeur ou un vecteur de sortie (appelé cible ou sortie désirée), que le réseau de neurones doit associer à un vecteur d'entrée. L'apprentissage consiste dans ce cas à ajuster les paramètres du réseau afin de minimiser l'erreur entre la sortie désirée et la sortie réelle du réseau.

Il ya plusieurs méthodes et algorithmes d'apprentissage, parmi ces règles on distingue la rétro propagation (back propagation).[42]

- **La rétro-propagation du gradient de l'erreur**

L'algorithme de rétro-propagation du gradient de l'erreur a été crée en généralisant les règles d'apprentissage de Widrow -Holff, aux réseaux multicouche à fonction de transfert non linéaire. La rétro-propagation du gradient de l'erreur est utilisée pour ajuster les poids et les biais du réseau afin de minimiser l'erreur quadratique entre la sortie du réseau et la sortie réelle. A chaque couple entrée/ sortie, une erreur est calculée, le gradient, ou réitère ces calculs jusqu'à l'obtention du critère d'arrêt.

Soit une réponse spécifique désirée D_p à chaque vecteur d'entrée X_p . La mise à jours des coefficients synaptiques W s'effectuent progressivement jusqu'à ce que l'erreur entre les sorties du réseau et les résultats désirés soit minimisée. Pour ce faire, considérons le problème de minimisation de la fonction cout défini pour n exemple $(X_p; D_p)$ et Y_p la sortie du réseau, par l'erreur qui peut être quadratique de la forme :[28]

$$E(w) = \sum_p (Y_p - D_p)^2 \quad (\text{III.4})$$

L'apprentissage revient à déterminer $W = \text{argMin}(E)$ (III.5)

En général on utilise des méthodes de gradient partiel, total ou stochastique :

$$W(t + 1) = w(t) - \lambda \text{grad}_w(E) \quad (\text{III.6})$$

$$\text{Gradient partiel : } w(t + 1) = W(t) + \lambda(D_p - f(WX_p))f'(WX_p)X_p \quad (\text{III.7})$$

$$\text{Gradient total : } W(t + 1) = w(t) - \lambda \text{grad}_w(E(t)) \quad (\text{III.8})$$

$$\text{Gradient stochastique : } W(t + 1) = w(t) - \lambda \text{grad}_w(E_p(t)) \quad (\text{III.9})$$

Ou $E_p = \sum_p (Y_p - D_p)^2$ erreur calculée sur un seul exemple que l'on tire au hasard chaque t.²

f : fonction d'activation

$\lambda(t)$: pas de gradient (fixe, décroissant, ou adaptatif)

Pour calculer $\frac{dE_p}{dW_{ij}}$, on exploite la rétro-propagation de gradient qui est une méthode astucieuse de calcul de gradient de la fonction cout en utilisant la dérivation composée pour rétro-propager l'erreur, comme le montre la figure suivante :[43]

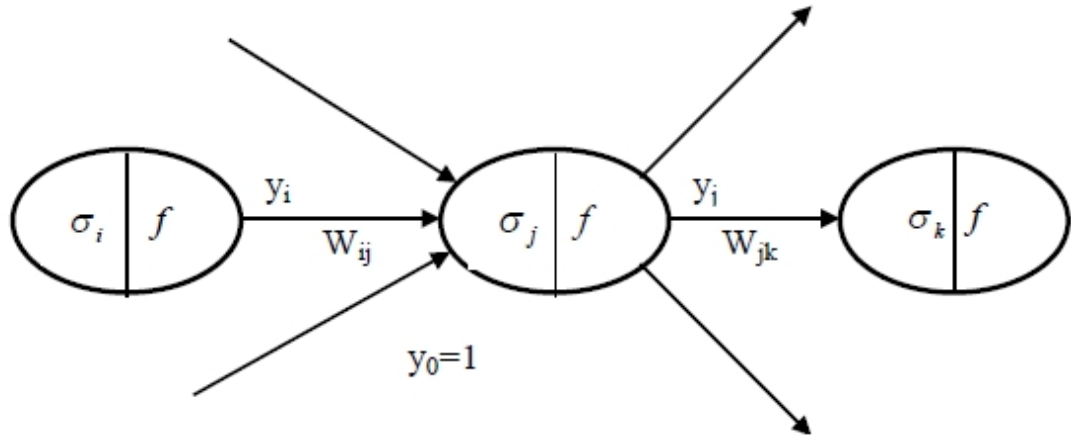


Figure III.5 : rétro-propagation du gradient

$$\frac{dE_p}{dW_{ij}} = \frac{dE_p}{d\sigma_j} \frac{d\sigma_j}{dW_{ij}} = \left(\frac{dE_p}{d\sigma_j} \right) y_i \quad (\text{III.10})$$

On pose $\delta_j = \frac{dE_p}{d\sigma_j}$ d'où

$$W_{ij}(t+1) = W_{ij}(t) - \lambda y_i \delta_j \quad (\text{III.11})$$

$$\text{Mais } \delta_j = \frac{dE_p}{d\sigma_j} = \sum_k \left(\frac{dE_p}{d\sigma_k} \right) \left(\frac{d\sigma_k}{d\sigma_j} \right) = \sum_p \delta_k \left(\frac{dE_p}{d\sigma_j} \right) = \sum_p \delta_k W_{jk} \left(\frac{dE_p}{d\sigma_k} \right) \quad (\text{III.12})$$

D'où $\delta_j = (\sum_k \delta_k W_{jk}) \dot{f}(\sigma_j)$ si neurone j est caché

$$\text{et } \delta_j = \left(\frac{dE_p}{d\sigma_j} \right) = \left(\frac{dE_p}{dy_j} \right) \left(\frac{dy_j}{d\sigma_j} \right) \quad (\text{III.13})$$

$$\text{D'où } \delta_j = 2(y_j - D_j) \dot{f}(\sigma_j) \quad (\text{III.14})$$

Cependant δ_j se calcul de proche en proche par rétro-propagation de l'erreur.

III.3.9 Systèmes neuro-flous

Les principaux avantages d'un système flou sont l'approche naturelle de la modélisation et la bonne interprétabilité de la description, en employant des règles linguistiques. Cependant, il n'y a aucune méthode formelle pour déterminer ses paramètres (fonctions d'appartenance et règles floues). Dans ce sens, il serait intéressant de disposer d'algorithmes permettant l'apprentissage automatique de ces paramètres. L'une des approches qui permettent de répondre à ce besoin est les réseaux de neurones connus pour leur algorithme d'apprentissage et leur précision dans l'ajustement numérique en employant des échantillons entrée/sortie. De nombreux auteurs ont donc tout naturellement cherché à combiner ces deux approches depuis le début des années 90 et ceci de plusieurs manières : coopérative, concurrente et hybride. Nous nous portons ici notre attention sur la dernière approche (hybride) qui permet de représenter sous forme d'un réseau de neurones, les différentes composantes d'un système flou. La structure du réseau ainsi obtenue dépend du type de règles floues, de méthodes d'inférence et de défuzzification employées par le système flou. Les paramètres du système flou (fonction d'appartenance et règles floues) peuvent ensuite être modifiés par un algorithme d'apprentissage conçu initialement pour l'ajustement des paramètres d'un réseau de neurones.

III.3.9.1 Modèles neuro-flous hybrides

Les approches neuro-floues modernes sont de cette forme. Un réseau neuronal et un système flou sont combinés dans une architecture homogène. Le système peut être interprété comme un réseau neuronal spécial avec des paramètres flous ou comme un système flou mis en application sous une forme distribuée parallèle. Les différentes architectures des systèmes neuro-flous hybrides sont : FALCON , GARIC , ANFIS et NEFCLASS.

Selon l'interaction entre la logique floue et les réseaux de neurones, et compte tenu de nos objectifs, nous avons retenu un type de systèmes d'inférence neuro-flou adaptatif (ANFIS) [44]

III.3.9.1.1 FALCON et GARIC

IL s'agit de modèle à 5 couches (Lin et Lee, 1991; Berenji et Khedkar, 1992) utilisant la fuzzification en entrée et la défuzzification en sortie. Ceci correspond à une interprétation

juste de la méthode de Mamdani. La précision accrue des résultats provoque une lenteur dans l'exécution du système. Ce modèle est rarement utilisé en pratique mais il donne de meilleurs résultats en commande.[44]

III.3.9.1.2 NEFCLASS

Modèle utilisé généralement pour la classification, il est constitué de 3 couches: une couche d'entrée avec les fonctions d'appartenance, une couche cachée représentée par l'ensemble des règles et une couche de sortie définissant l'ensemble des classes. Ce modèle est facile à mettre en application, il évite l'étape de défuzzification, tout en étant précis dans le résultat final, avec une rapidité supérieure aux autres modèles.[44]

III.3.9.1.3 Système d'inférence neuro-flou adaptative(ANFIS)

ANFIS (Adaptive-Network-based Fuzzy Inference System) est un système flou mis en application dans le cadre des réseaux adaptatifs. Il a été proposé par Jang (Jang, 1993). Pour les utilisateurs de MATLAB, ANFIS est implanté dans la boîte à outils 'Neuro-fuzzy' accompagné de son algorithme d'apprentissage hybride basé sur la rétro propagation et la méthode des moindres carrés. Ce modèle donne de très bons résultats d'approximation de fonctions non linéaires.

III.3.9.1.3.1 Structure d'ANFIS

ANFIS met en application un SIF du type Takag Sugeno et a une architecture composée de cinq couches comme représentée ci-dessous. Nous considérons un système à deux entrées x_1 , x_2 et une sortie y . Considérons aussi un modèle flou de type TSK de ce système, composé des deux règles suivantes:

$$\text{Si } x_1 \text{ est } A_1 \text{ et } x_2 \text{ est } B_1 \text{ alors } y_1 = f_1(x_1, x_2) = a_1 x_1 + b_1 x_2 + c_1 \quad (\text{III.15})$$

$$\text{Si } x_1 \text{ est } A_2 \text{ et } x_2 \text{ est } B_2 \text{ alors } y_2 = f_2(x_1, x_2) = a_2 x_1 + b_2 x_2 + c_2 \quad (\text{III.16})$$

Jang [45]a proposé de représenter cette base de règles par le réseau adaptatif suivant:

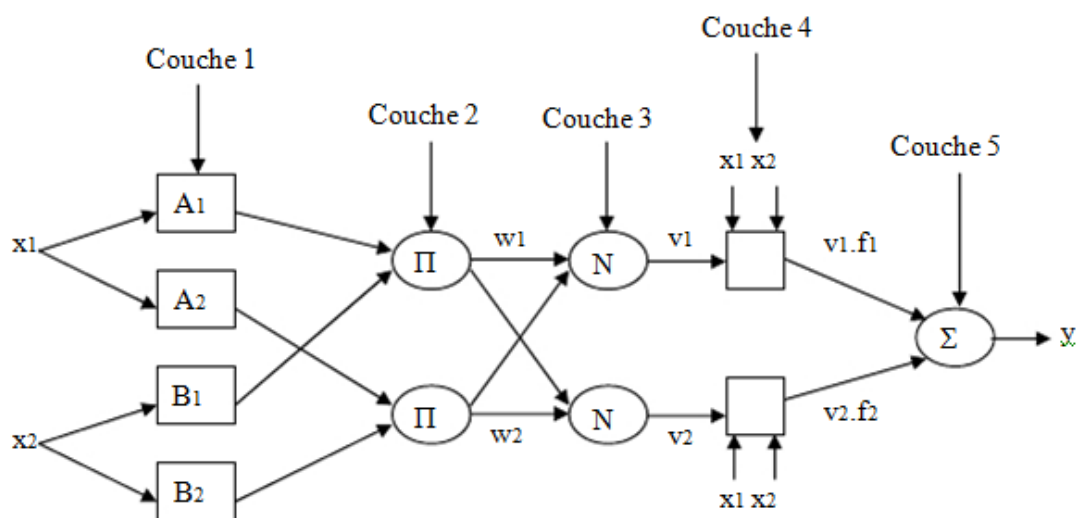


Figure III.6 : Architecture ANFIS

Le réseau adaptatif ANFIS est un réseau multicouches dont les connexions ne sont pas pondérées, ou ont toutes un poids de 1. Les nœuds sont de deux types différents selon leur fonctionnalité: les nœuds carrés (adaptatifs) contiennent des paramètres, et les nœuds circulaires (fixes) n'ont pas de paramètres.

Toutefois, chaque nœud (carré ou circulaire) applique une fonction sur ses signaux d'entrées. La sortie o_i^k du nœud i de la couche k (appelée nœud (i, k)) dépend des signaux provenant de la couche $(k-1)$ des paramètres des nœuds (i, k) , c'est-à-dire :[46]

$$o_i^k = f(o_1^{k-1}, \dots, o_{n_{k-1}}^{k-1}, a, b, c, \dots) \quad (\text{III.17})$$

Où n_{k-1} est le nombre de nœuds dans la couche $k-1$, et a, b, c, \dots sont les paramètres du nœud (i, k) . Pour un nœud circulaire ces paramètres n'existent pas.

- **Couche 1 :** génération de degré d'appartenance

Les neurones de cette couche réalisent les ensembles flous qui serviront dans les antécédents des règles. Dans le modèle de Jang, les fonctions d'appartenance sont des gaussiennes.

$$o_i^1 = f_i^1(x) = \mu_{A_i}(x) \quad (\text{III.18})$$

Où x est l'entrée du nœud i , et A_i le terme linguistique associé à sa fonction. En d'autres termes, o_i^1 est le degré d'appartenance de x à A_i . Les paramètres d'un nœud de cette couche sont ceux de la fonction d'appartenance correspondante

- **Couche 2** : génération du poids de la règle i

Chaque neurone dans cette couche correspond à une règle floue Sugeno. Il reçoit les sorties des neurones de fuzzification et calcule son activation. La conjonction des antécédents est réalisée avec l'opérateur produit.

$$w_i = \mu_{A_i}(x_1) \cdot \mu_{B_{i-2}}(x_2) \quad , \quad i = 1,2 \quad (\text{III.19})$$

- **Couche 3** : normalisation du poids de la règle i :

Chaque neurone calcule le degré de vérité normalisé d'une règle floue donnée. La valeur obtenue représente la contribution de la règle floue au résultat final.

$$v_i = \frac{w_i}{w_1 + w_2} \quad (\text{III.20})$$

- **Couche 4** : Calcule de la sortie des règles

Chaque neurone i de cette couche est relié à un neurone de normalisation correspondant et aux entrées initiales du réseau. Il calcule le conséquent pondéré de la règle.

$$O_i^4 = v_i \cdot f_i = v_i(a_i x_1 + b_i x_2 + c_i), i = 1..2 \quad (\text{III.21})$$

Où v_i est la sortie de la couche 3, et $\{a_i, b_i, c_i\}$ est l'ensemble des paramètres de sortie de la règle i .

- **Couche 5** : Faire la somme de toutes les entrées à partir de la couche 4

Comprend un seul neurone qui fournit la sortie de ANFIS en calculant la somme des sorties de tout le nœud de défuzzification c'est-à-dire,

$$O_1^5 = y = \sum_i v_i f_i \quad (\text{III.22})$$

III.3.10 Conclusion

Nous avons présenté dans ce chapitre une description générale des réseaux de neurones, la théorie floue, et les systèmes neuro-flous. Nous nous sommes intéressées principalement aux systèmes ANFIS, alors nous pouvons dire que : L'utilisation conjointe des méthodes neuronales et floues permet de tirer l'avantage des qualités de l'une et de l'autre. Principalement, des capacités d'apprentissage des premières et de la lisibilité et la souplesse des dernières. L'architecture du système neuro-flou adopté dans ce chapitre a montré des bonnes performances dans les simulations des systèmes non linéaires.

IV.1 Introduction

La commande par mode glissant est une technique de commande non linéaire. Elle est caractérisée par sa robustesse vis-à-vis des incertitudes paramétriques du modèle et des perturbations extérieures. L'inconvénient majeur de cette technique est le phénomène de réticence qui se caractérise par de fortes oscillations autour de la surface de glissement. Ce phénomène peut exciter des dynamiques non modélisées et conduire à l'instabilité. L'objectif principal de ce chapitre est de pallier à cet inconvénient, en remplaçant la partie discontinue de la commande par mode glissant, par une partie neuro-floue, afin d'atteindre et de maintenir avec précision une position et une vitesse désirées d'un bras manipulateur à deux axes.

IV.2 Stratégie de la commande par mode glissant

Soit un robot manipulateur ayant pour équation du mouvement :

$$M(q)\ddot{q} + V(q, \dot{q}) + G(q) = \tau \quad (\text{IV.1})$$

Posant le changement de variables suivant :

$x_1 = q_1$: Position angulaire de l'articulation 1

$x_2 = q_2$: Position angulaire de l'articulation 2

$\dot{x}_1 = \dot{q}_1$: Vitesse angulaire de l'articulation 1

$\dot{x}_2 = \dot{q}_2$: Vitesse angulaire de l'articulation 2

Notre objectif est donc d'établir une loi de commande qui permet de commander le système en position et en vitesse.

$$\text{On pose :} \quad \begin{cases} e_1(t) = x_1(t) - x_{1d}(t) \\ e_2(t) = x_2(t) - x_{2d}(t) \end{cases} \quad (\text{IV.2})$$

$e_1(t)$ et $e_2(t)$: Les erreurs des positions

$$\begin{cases} \dot{e}_1(t) = \dot{x}_1(t) - \dot{x}_{1d}(t) \\ \dot{e}_2(t) = \dot{x}_2(t) - \dot{x}_{2d}(t) \end{cases} \quad (\text{IV.3})$$

$\dot{e}_1(t)$ et $\dot{e}_2(t)$: les erreurs des vitesses

L'équation(IV.1) peut être écrite sous la forme suivante :

$$x^{(n)} = f(x) + g(x)u \quad (\text{IV.4})$$

La surface de glissement est donnée par :

$$s = \lambda e + \dot{e} \quad (\text{IV.5})$$

La loi de commande en mode glissant est donnée par :

$$u = u_{eq} + \Delta u \quad (\text{IV.6})$$

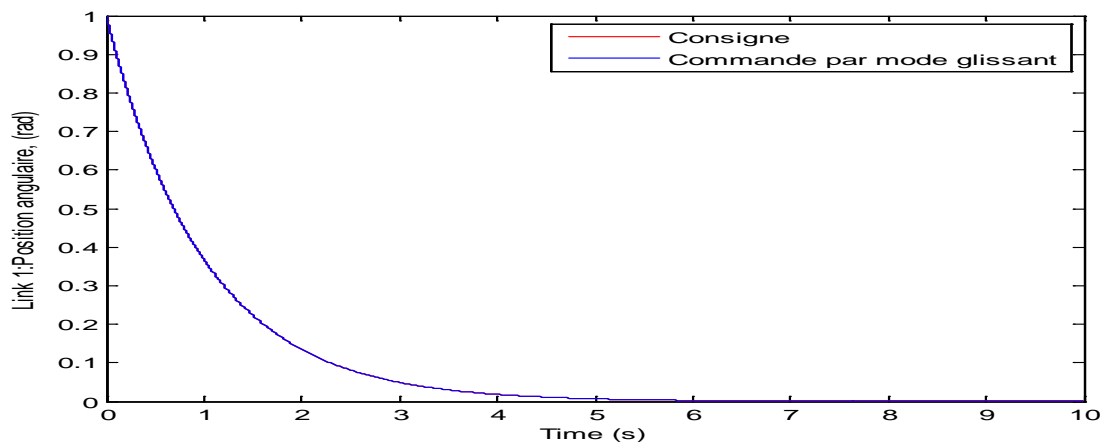
Avec u_{eq} est définie par $\dot{s} = 0$ (IV.7)

$$u_{eq} = - \left[\frac{\partial s}{\partial x} g(x) \right]^{-1} \frac{\partial s}{\partial x} f(x) \quad (\text{IV.8})$$

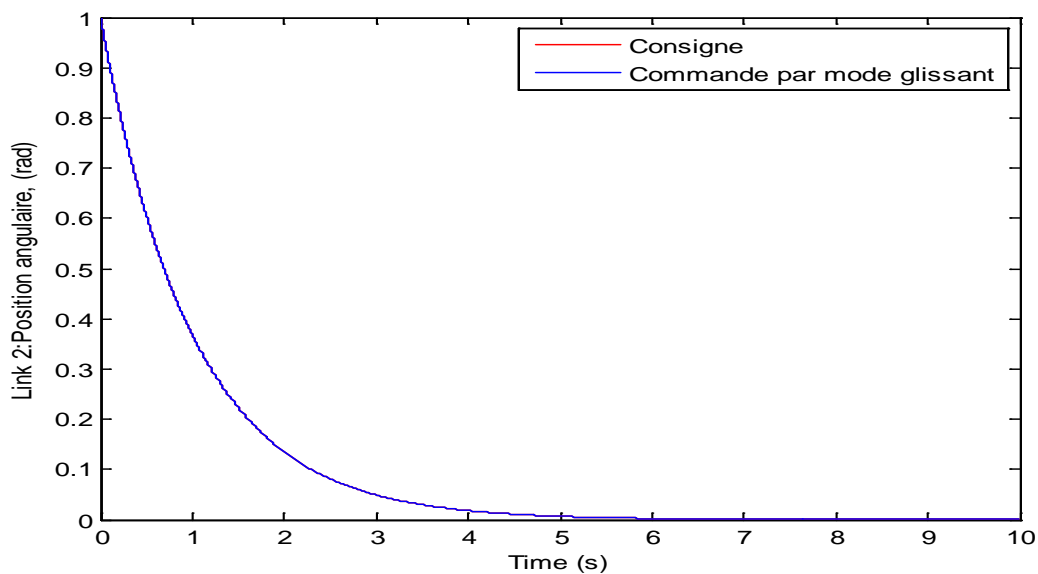
$$\Delta u = -k \text{sign}(s) \quad (\text{IV.9})$$

IV.3 Simulation et interprétation

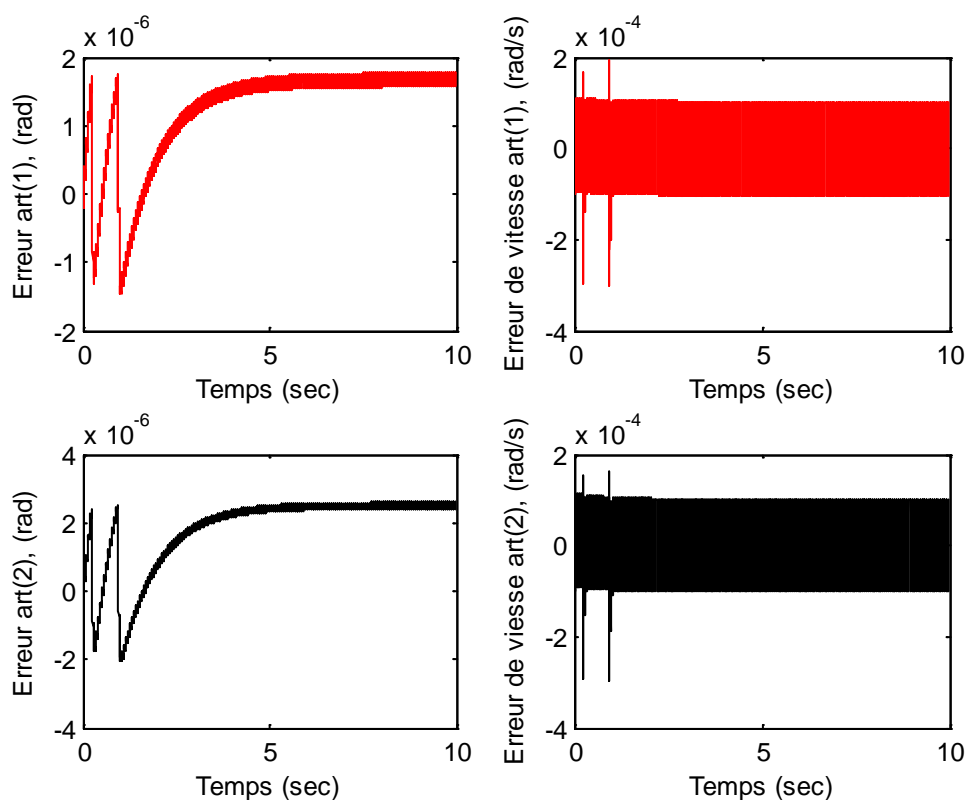
- Sans perturbation



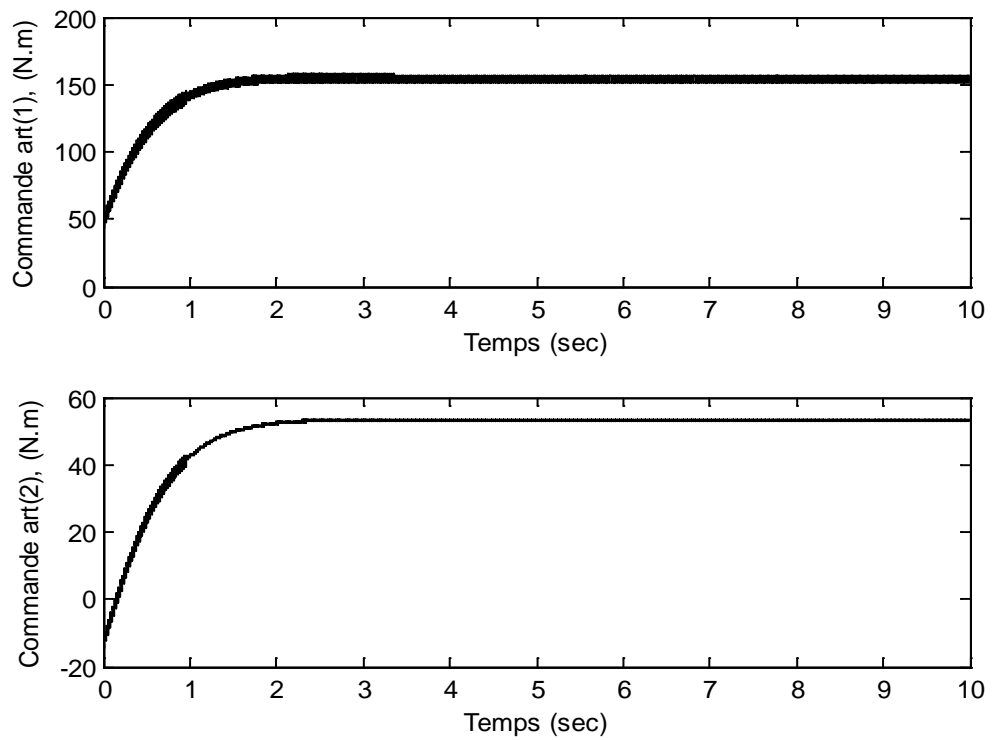
Figure(IV.1) : Trajectoires désirée et réelle (articulation 1)



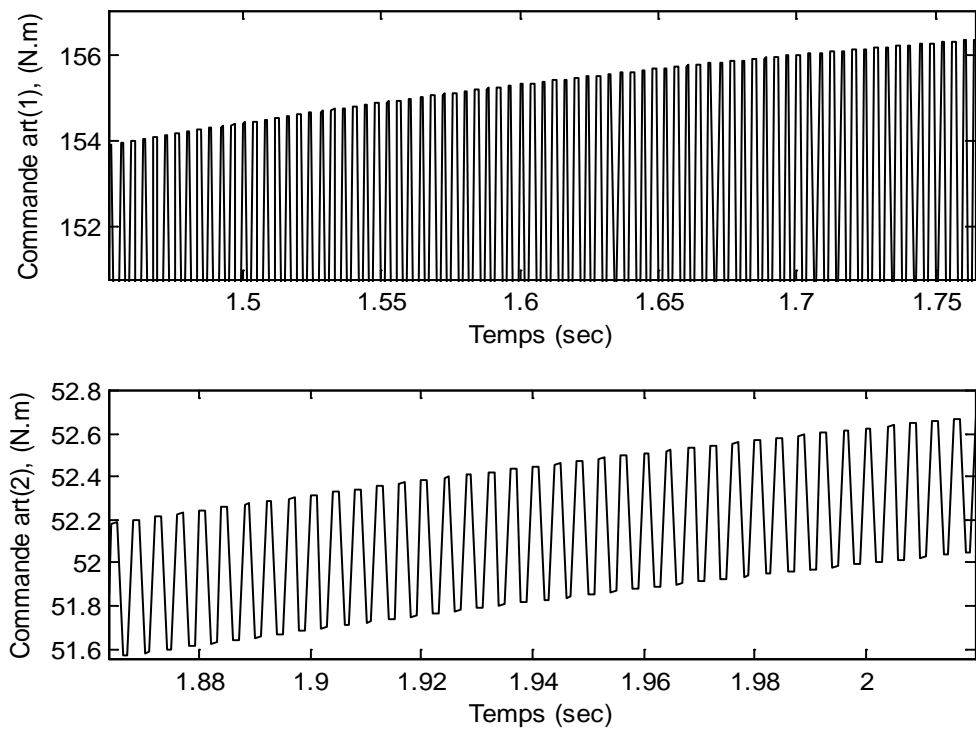
Figure(IV.2) : Trajectoires désirée et réelle (articulation 2)



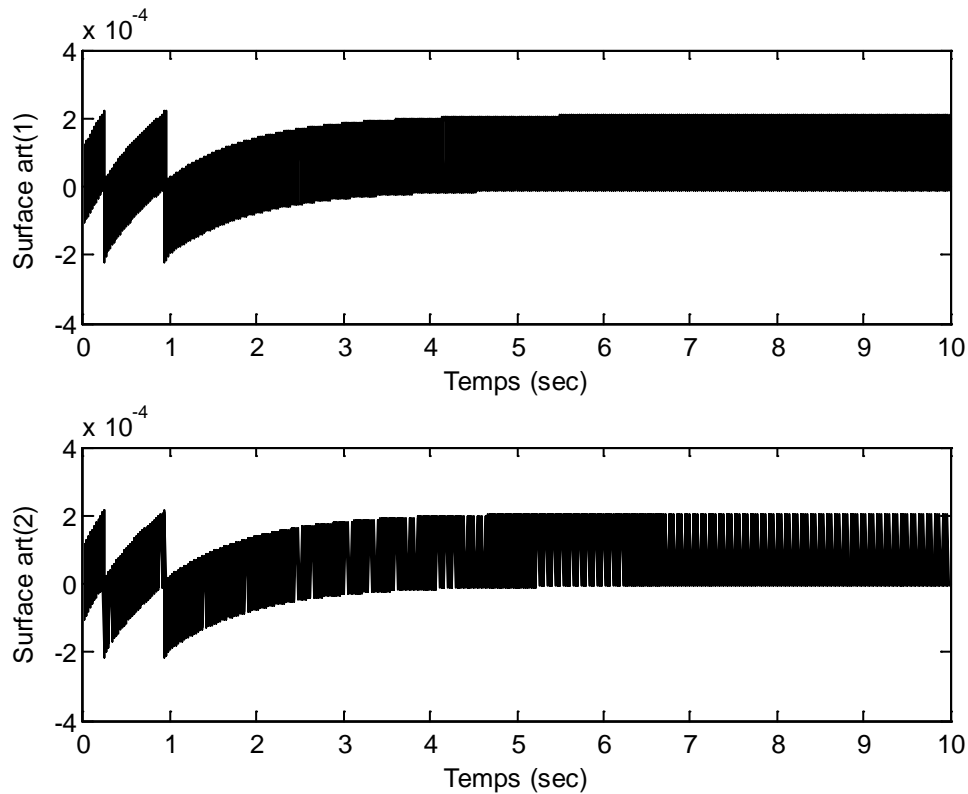
Figure(IV.3) : Profil des erreurs de position et de vitesse (articulation 1 et 2)



Figure(IV.4) : Evolution des commandes (articulation 1 et 2)



Figure(IV.5) : Evolution des commandes (articulation 1 et 2)



Figur(IV.6): Evolution des surfaces de glissement (articulation1, 2)

Les figures (IV.1,IV.2,IV.3 et IV.4) illustrent les résultats obtenus avec la commande par mode glissant sans perturbations.

Elles décrivent l'évolution des trajectoires désirées (position angulaire), les erreurs en positions et en vitesses, les surfaces de glissements et les commandes des deux articulations .

Nous constatons que le suivi de la trajectoire de position est fidèle à la consigne pour les deux articulations.

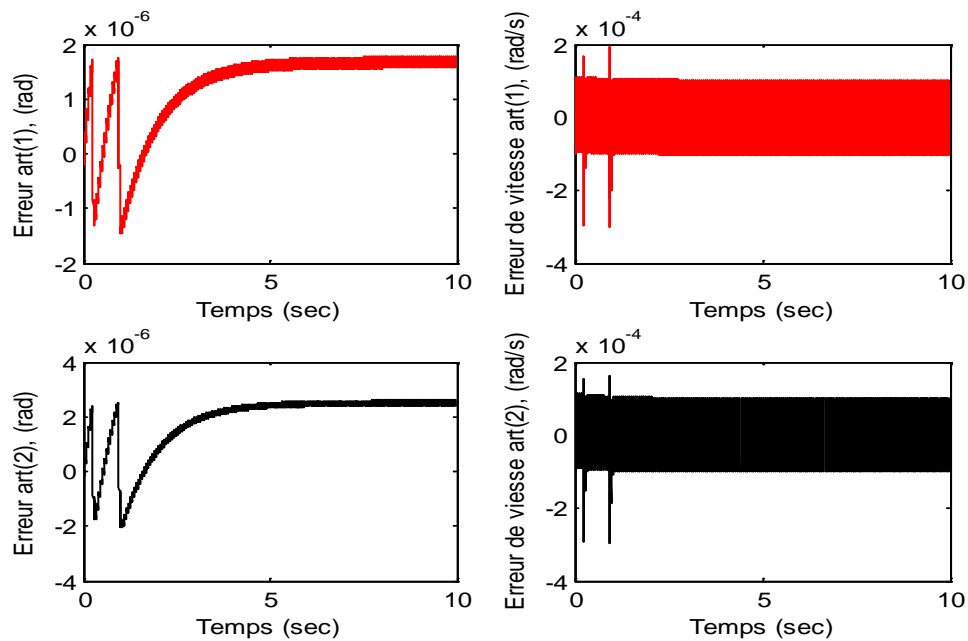
les erreurs en positions sont bornées par de très faibles valeurs , elles sont de l'ordre de 1.8×10^{-6} rd pour l'articulation 1 et de 2.8×10^{-6} rd pour l'articulation 2. Nous notons aussi que les erreurs de vitesses sont aussi très faibles . l'erreur maximale pour l'articulation 1 est de 2×10^{-4} rd/s et celle de l'articulation 2 atteint 1.8×10^{-4} rd/s.

Les figures (IV5, IV6) montrent que les deux surfaces convergent vers zéro, et la figure 4 montre que la commande appliquée sur l'articulation 1 est d'amplitude 152 N.m et celle appliquée sur l'articulation 2 est d'amplitude 57 N.m.

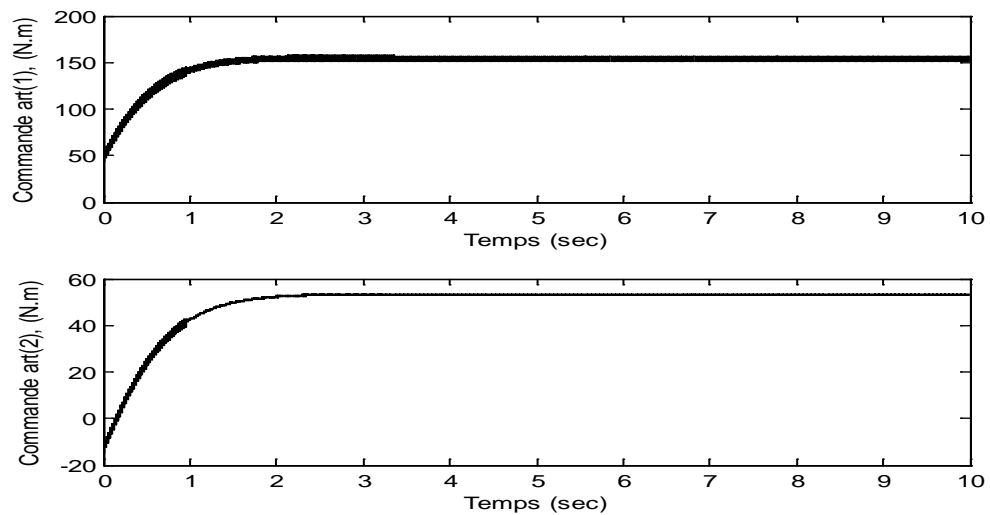
La plus importante remarque qu'il faut noter c'est que les allures de toutes les mesures sont relativement hachées, ce qui correspond au phénomène de chattering, l'inconvénient majeur de la commande par mode glissant.

- Avec variation paramétrique

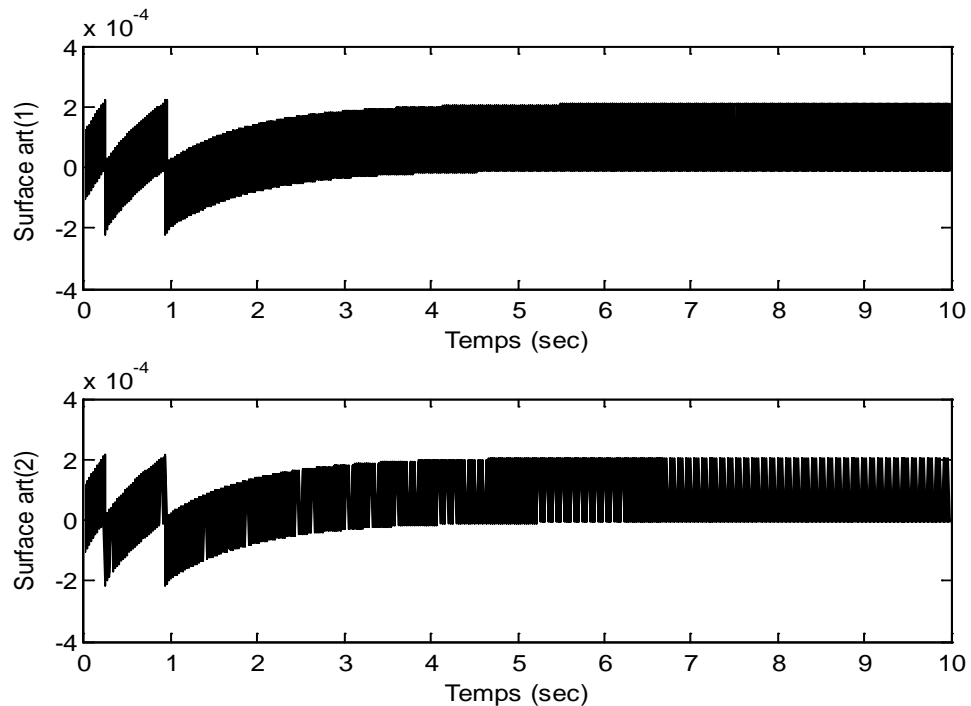
Pour tester la robustesse de la commande par mode glissant, nous avons introduit à l'instant $t=4s$ une variation paramétrique sur la matrice d'inertie, en la multipliant par 800.



Figure(IV.7) : Profil des erreurs de positions et de vitesses (articulation 1 et 2)



Figure(IV.8) : Evolution des commandes (articulation 1 et 2)



Figure(IV.9) :Evolution des surfaces de glissement (articulation 1 et 2)

Atravers ces résultats, nous observons que les erreurs de position et de vitesse restent toujours faibles et les surfaces de glissements convergent toujours vers zéro, avec présence de broutement. Cette perturbation paramétrique n'a pas affecté les performances de poursuite de la commande par mode glissant. Ceci se traduit par la robustesse de cette approche vis-à-vis les variations paramétriques. Pour remédier au problème de chattering, nous remplaçons la partie discontinue (responsable de chattering) de la commande par mode glissant, par une commande neuro-floue.

IV.4 Stratégie de la commande par les réseaux de neurones et de la logique floue

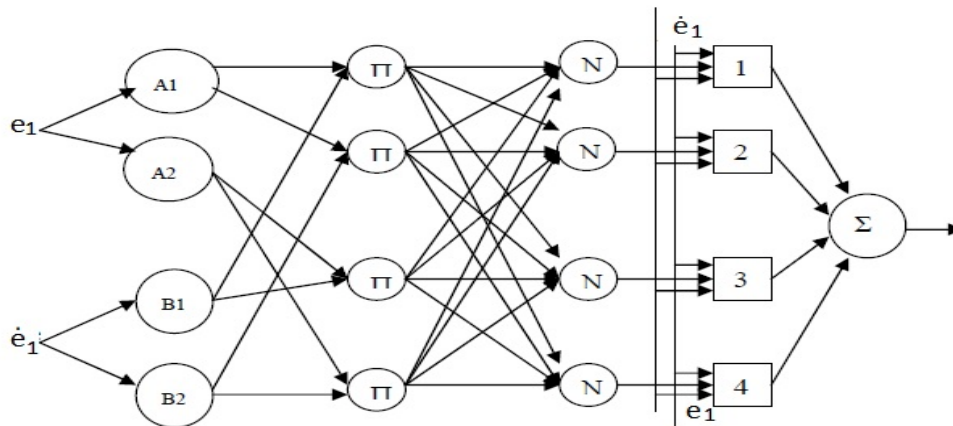
IV.4.1 Régulateur Neuro-flou

Nous avons utilisé la commande neuro-floue (ANFIS) pour remplacer la partie discontinue de la commande par mode glissant, afin de remédier au phénomène de réticence, et d'atteindre avec précision la position et la vitesse désirées.

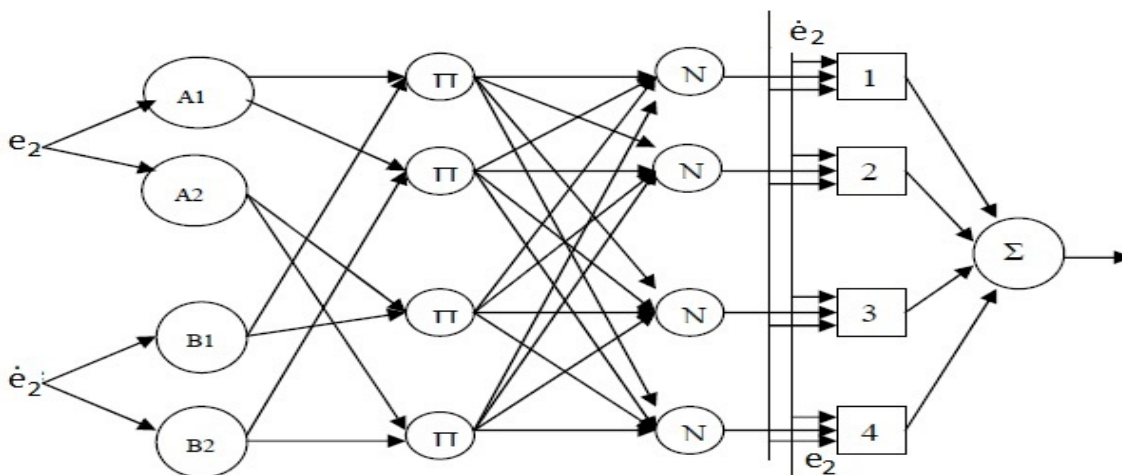
Soit : $V_{ei} = [e_i \dot{e}_i]$ le vecteur d'entrée pour chaque articulation i à deux composantes, erreur de position et erreur de vitesse.

Comme ensembles flous, nous avons utilisé trois : N (négatif), Z (zéro) et P (positif)). Pour les fonctions d'appartenance de l'erreur et sa dérivée, nous avons opté pour forme triangulaire.

La structure du régulateur sous forme d'un réseau adaptatif est présentée dans la figure suivante [46]



Figure(IV.10) : Le régulateur de type segeno sous forme de RNA(articulaion 1)



Figure(IV.11) : Le régulateur de type segeno sous forme de RNA(articulaion 2)

Pour la structure de contrôleur (ANFIS), nous l'avons décrite dans le chapitre précédent. Nous avons combiné les réseaux de neurones et le système flou de type SUGENO.

Les trois paramètres de la couche 4 sont ajustés par l'algorithme de rétro propagation.

- **Algorithme d'apprentissage**

Pour estimer les paramètres du réseau de neurones, nous avons utilisé le filtre KALMAN étendu comme un algorithme d'apprentissage. Le régulateur est caractérisé par un vecteur de paramètre ϕ . Notre objectif est de trouver les valeurs de vecteur ϕ en minimisant l'erreur suivante :

$$e(k) = \frac{1}{2} (h_d(k) - h(k))^2 \quad (\text{IV.10})$$

$k=1..2$

L'approche du filtre de KALAMAN étendu consiste à linéariser à tout instant la sortie y autour du vecteur estimé $\hat{\phi}$. ceci revient à écrire :

$$h_d(k) = h(k) + \Psi_i^T(k)(\phi_i - \hat{\phi}_{i-1}) \quad (\text{IV.11})$$

$$\Psi_i^T(k) = \frac{\partial h(k)}{\partial \phi_i} \quad (\text{IV.12})$$

Par conséquent, les paramètres sont ajustés selon la relation suivante :

$$\phi_i = \phi_{i-1} + P_i \Psi_i^T e_i \quad (\text{IV.13})$$

$$\begin{cases} e_i = h_d - h \\ \Psi_i^T = [O_{3,i} O_{3,i} e_i O_{3,i} e_i] \\ P_i = \frac{\alpha_1}{\alpha_2 + \Psi_i^T \Psi_i} \end{cases} \quad (\text{IV.14})$$

$P(k)$ est le gain de l'algorithme d'estimation, α_1 et α_2 fixés à 1 sont des gains d'adaptation pour modifier la vitesse de convergence.

Nous remarquons que cette méthode nécessite le calcul du gradient qui n'est rien d'autre que la méthode de retro propagation utilisée dans les réseaux de neurones.

$$\Psi = \frac{\partial h}{\partial \phi} \quad (\text{IV.15})$$

Pour le cas étudié, le vecteur des paramètres est :

$$\phi_i = [r_i p_i \delta_i]^T \quad (\text{IV.16})$$

Par conséquent nous avons

$$\Psi_i = \frac{\partial y}{\partial \phi_i} = \left[\frac{\partial h}{\partial r_i} \quad \frac{\partial h}{\partial P_i} \quad \frac{\partial h}{\partial \delta_i} \right]^T \tag{IV.17}$$

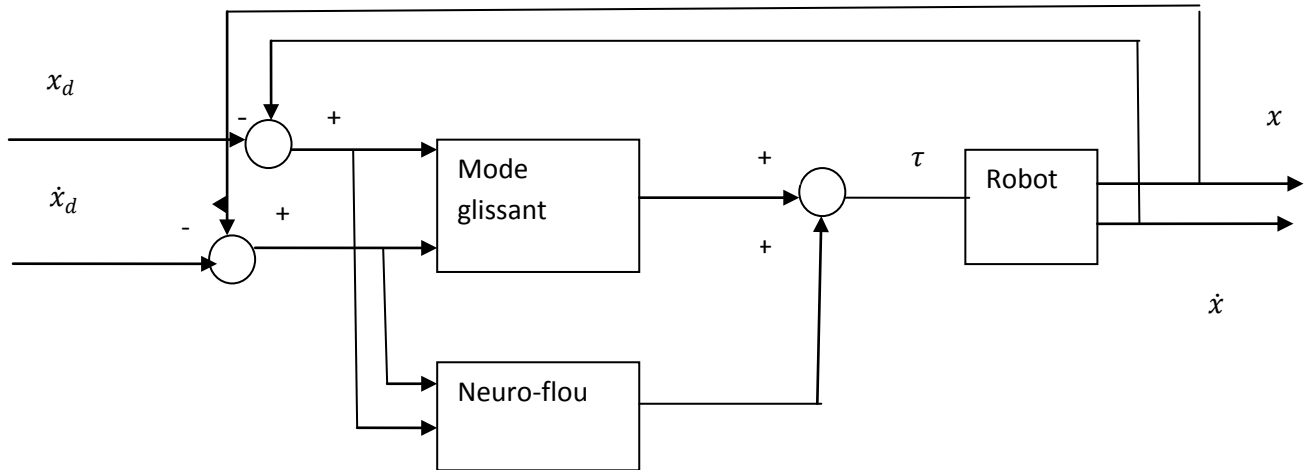
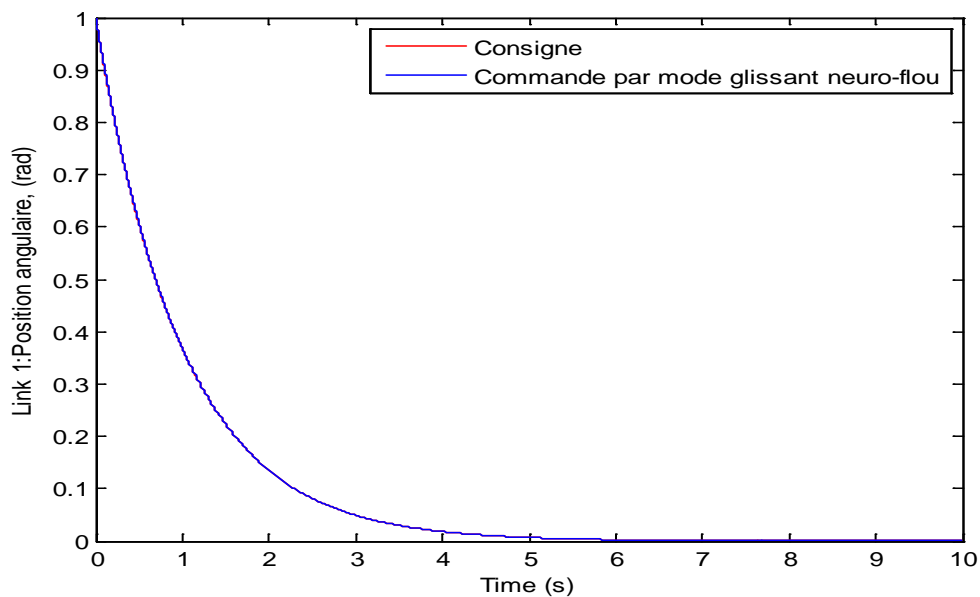


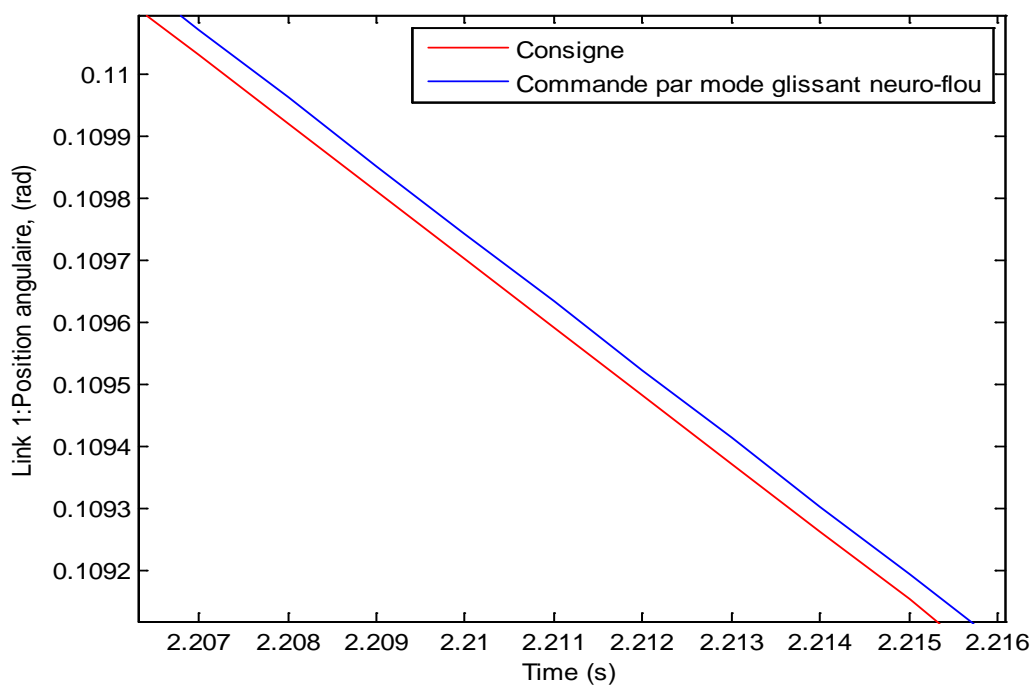
Figure (IV. 12) : Schéma (commande par mode glissant avec le neuro-flou)

IV.2 Simulation et interprétation

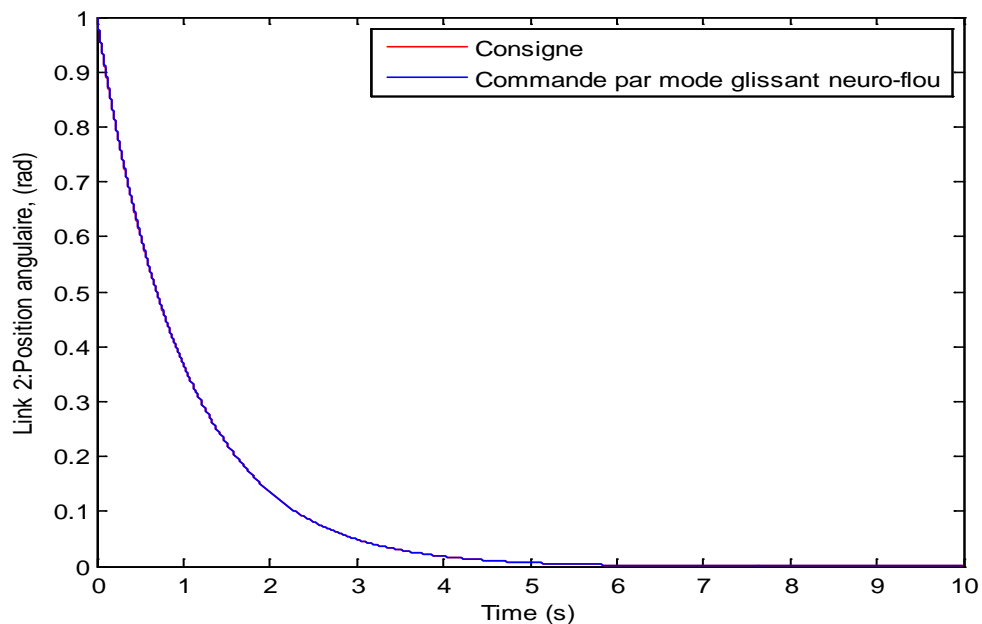
- sans perturbation



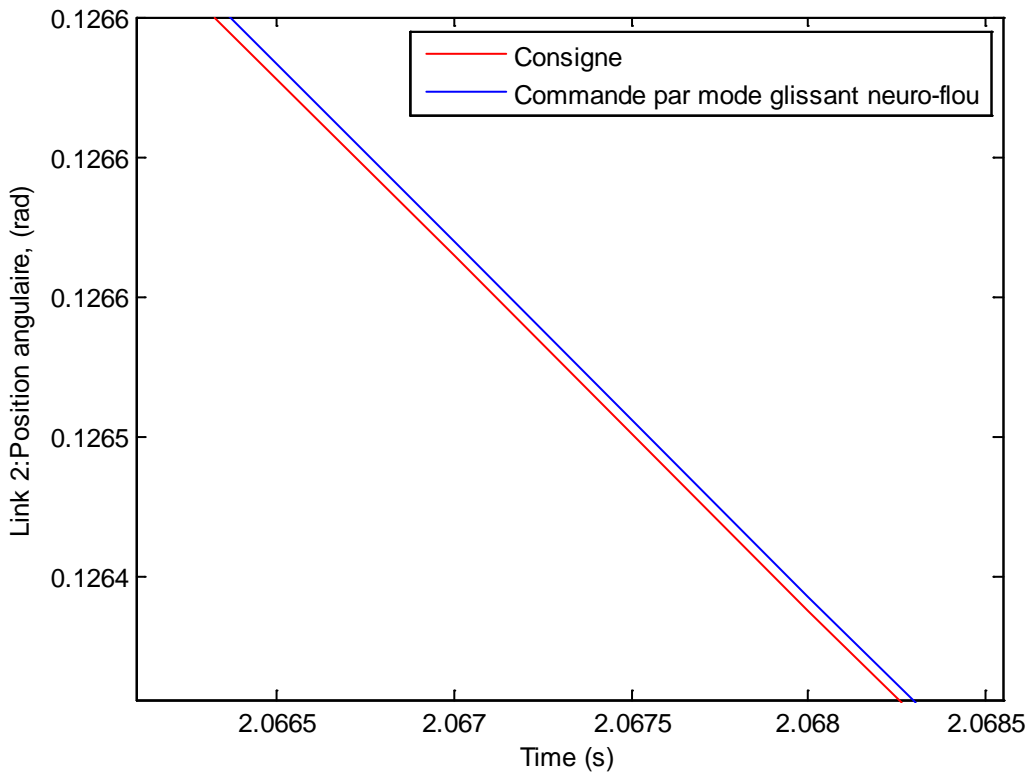
Figure(IV.13) : Trajectoires désirée et réelle (articulation 1)



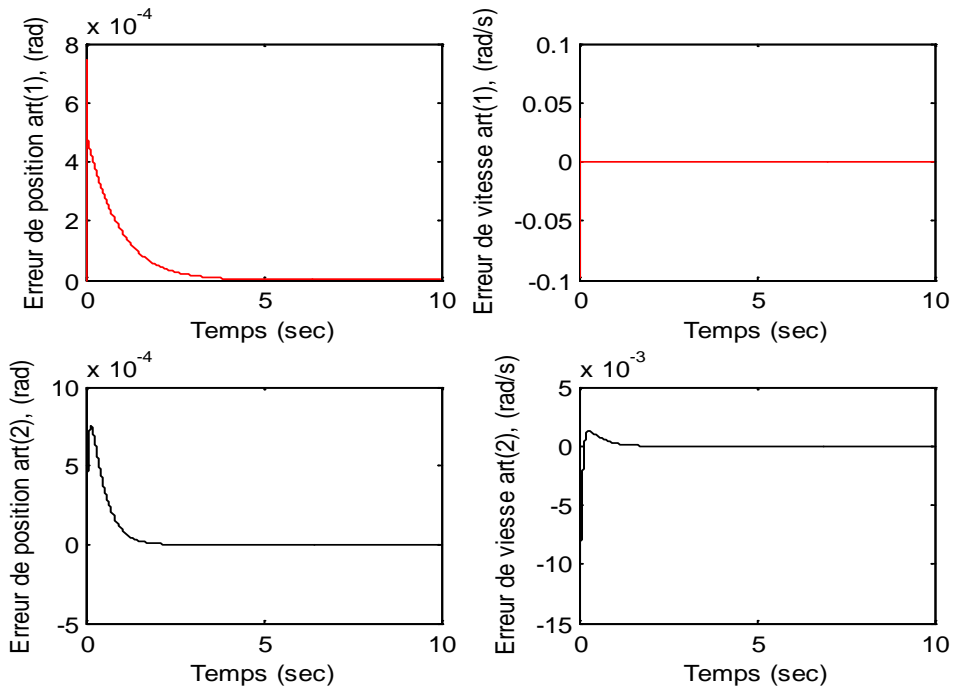
Figure(IV.14) : Trajectoires désirée et réel (articulation 1)



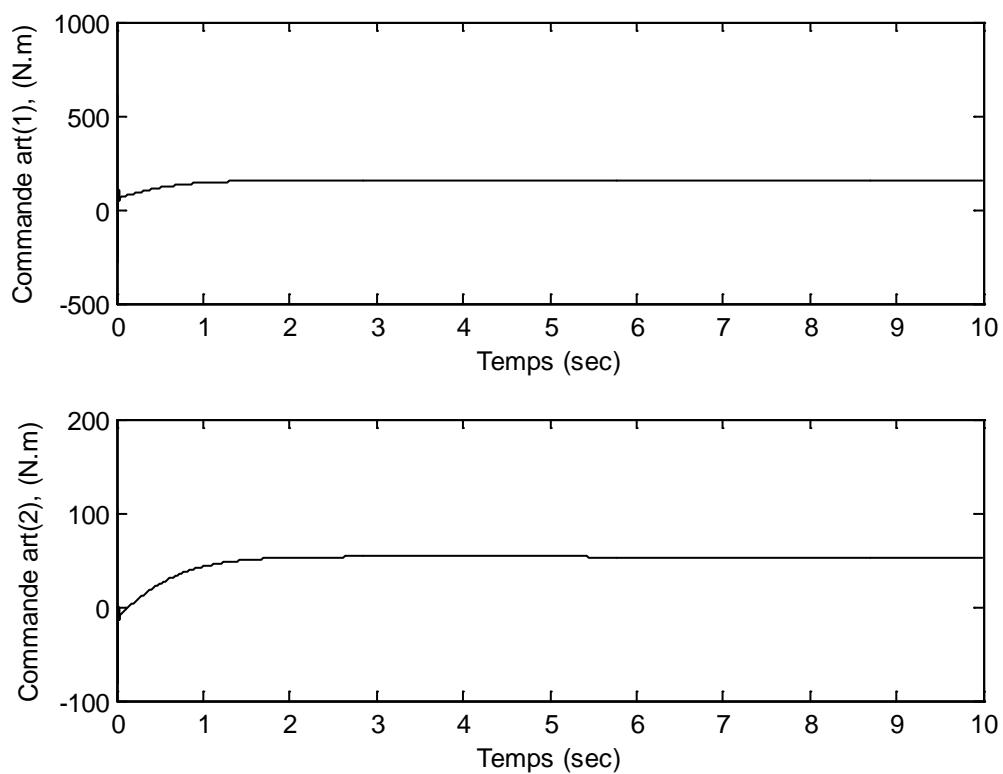
Figure(IV.15) : Trajectoires désirée et réelle (articulation 2)



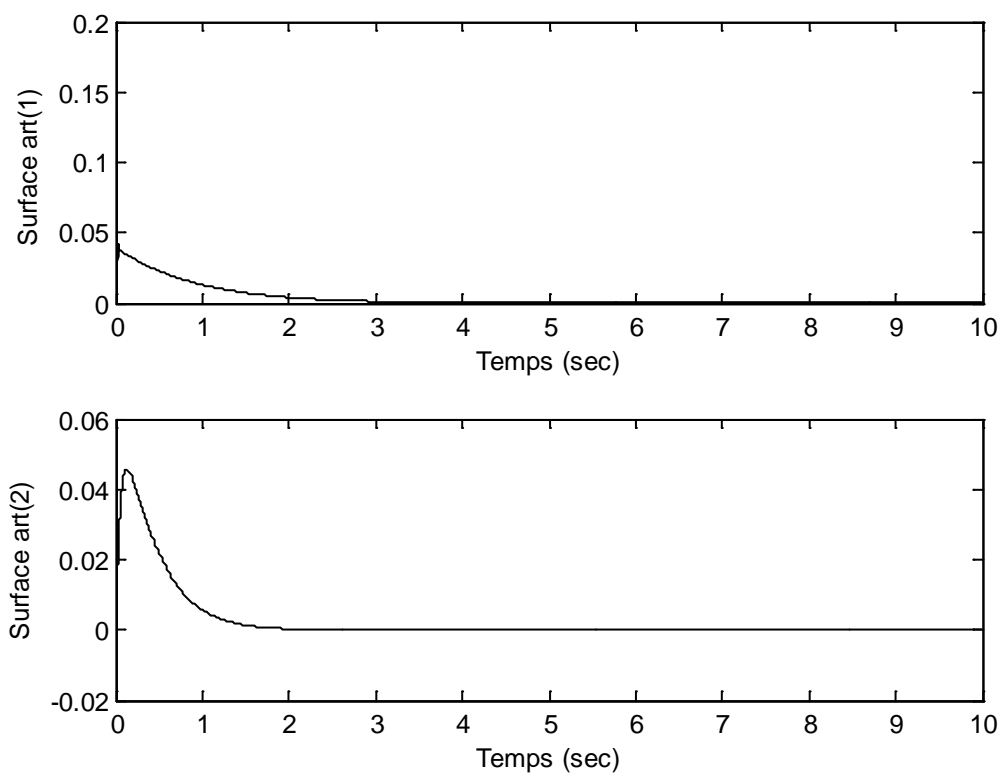
Figure(IV.16) : Trajectoires désirée et réelle (articulation 2)



Figure(IV.17) : Profil des erreurs de position et de vitesse (articulation 1et 2)



Figure(IV.18) : Evolution des commandes (articulation 1 et 2)



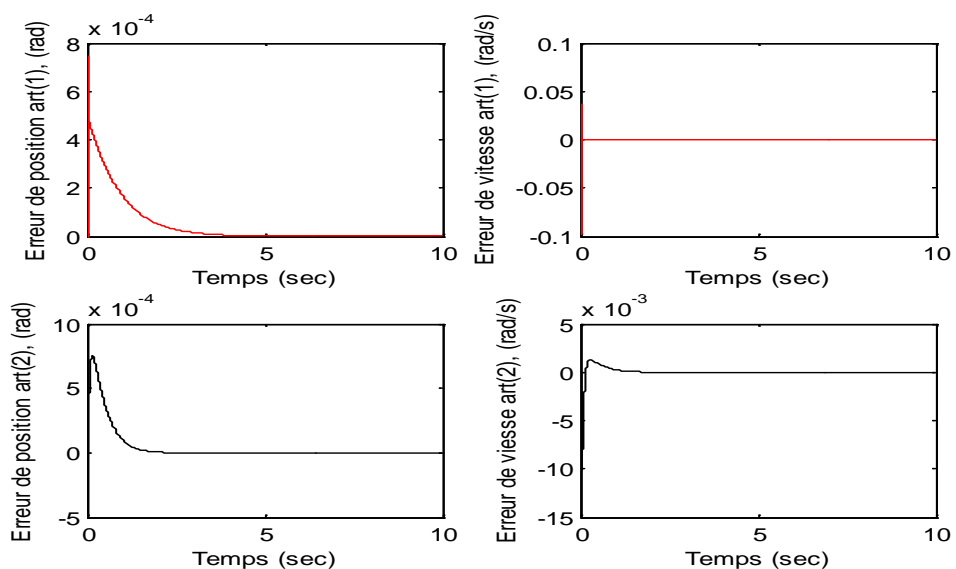
Figure(IV.19) : Evolution des surfaces de glissement (articulation 1 et 2)

Pour le contrôle en position, nous notons que la trajectoire angulaire suit bien la trajectoire désirée, mais avec une précision légèrement inférieure par rapport à la commande par mode glissant.

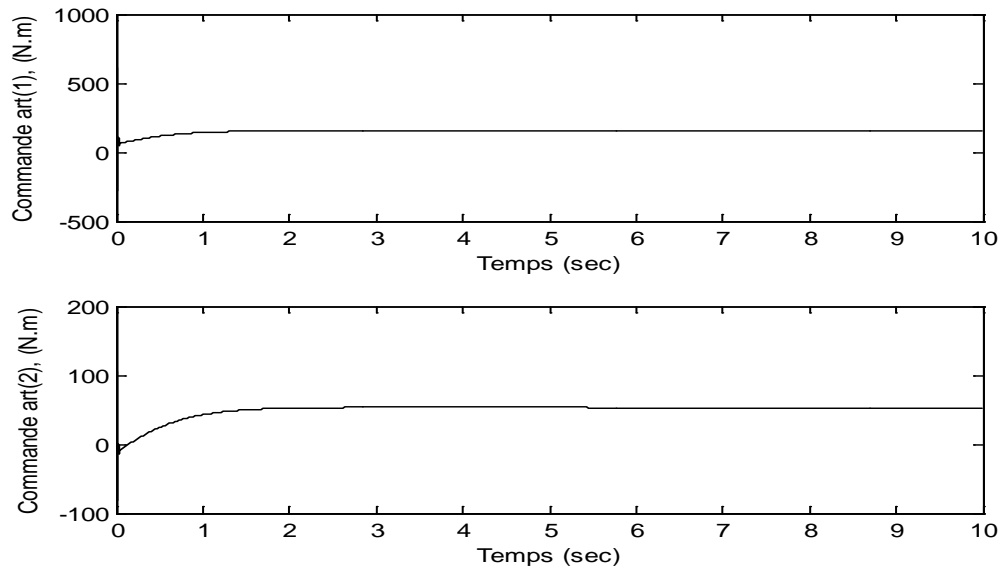
Les erreurs de vitesses sont aussi légèrement supérieures par rapport à celles obtenues avec le mode glissant. Les surfaces de glissement convergent vers zéro et les commandes sont lisses d'amplitudes de 20 N.m pour l'articulation 1 et de 40 N.m pour l'articulation 2.

Le plus important que nous pouvons remarquer est que le suivi de trajectoire est atteint et le chattering n'existe plus.

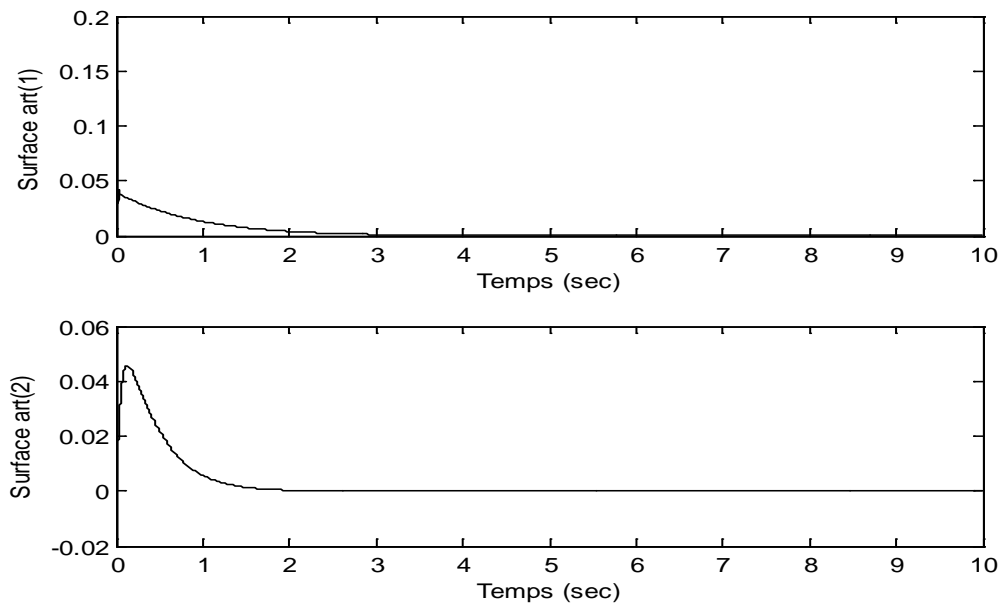
- Avec variation paramétrique



Figure(IV.20) : Profil des erreurs de position et de vitesse (articulation 1 et 2)



Figure(IV.21) : Evolution de la commande (articulation 1 et 2)



Figure(IV.22) : Evolution de la surface (articulation 1 et 2)

La même chose avec le mode glissant, à l'instant $t=4$ s, nous avons appliqué une variation paramétrique sur la matrice d'inertie, en la multipliant par 800. Nous constatons que les performances de poursuite ne sont pas affectées.

Pour confirmer la validité de notre commande (commande par mode glissant neuro-flou) et tester sa robustesse vis-à-vis des perturbations structurées ou non structurées, nous

avons introduit une perturbation externe de valeur importante sur les deux commandes (la commande par mode glissant et la commande hybride).

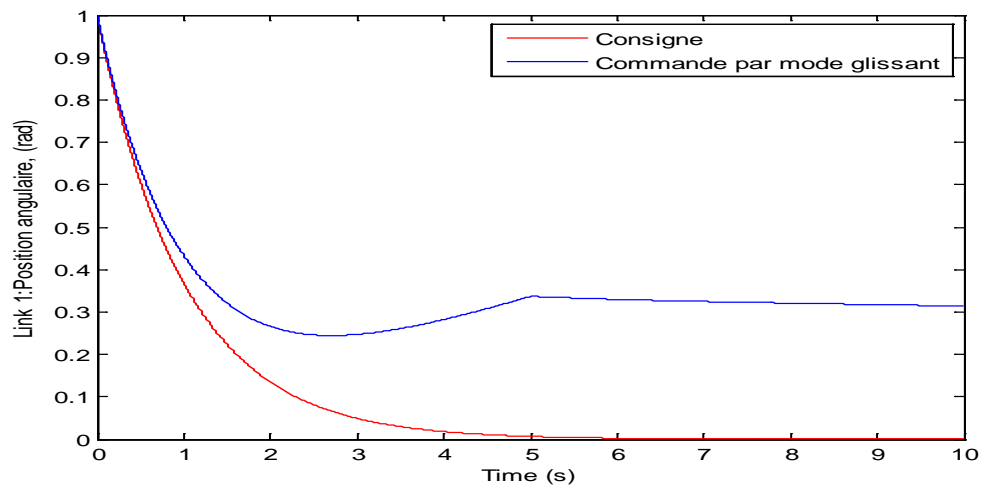
- **Avec perturbation externe**

A l'instant $t=4s$, nous avons introduit une perturbation externe très importante nous l'avons ajoutée au modèle du système.

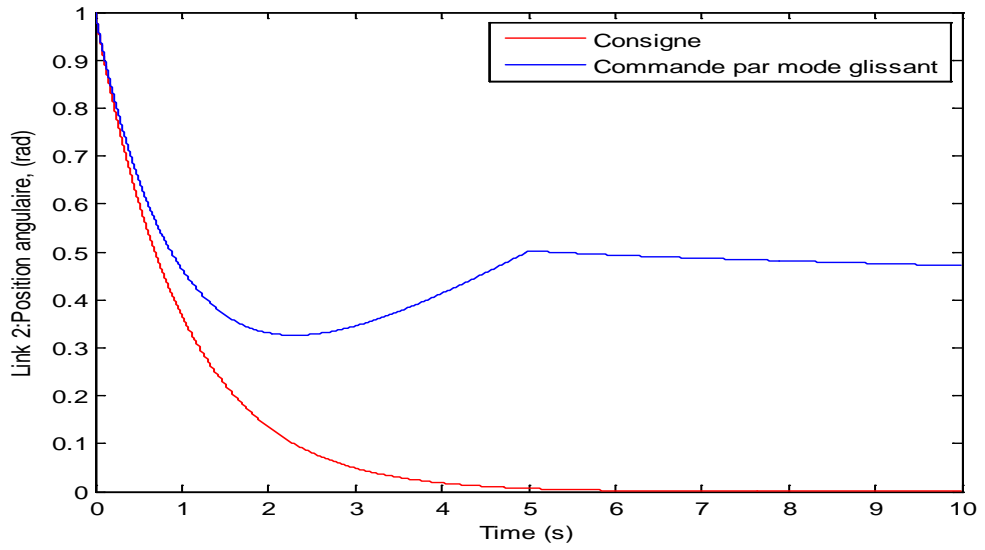
Donc l'équation de notre système devient :

$$\ddot{x} = f(x) + g(x).u + d \quad (IV.18)$$

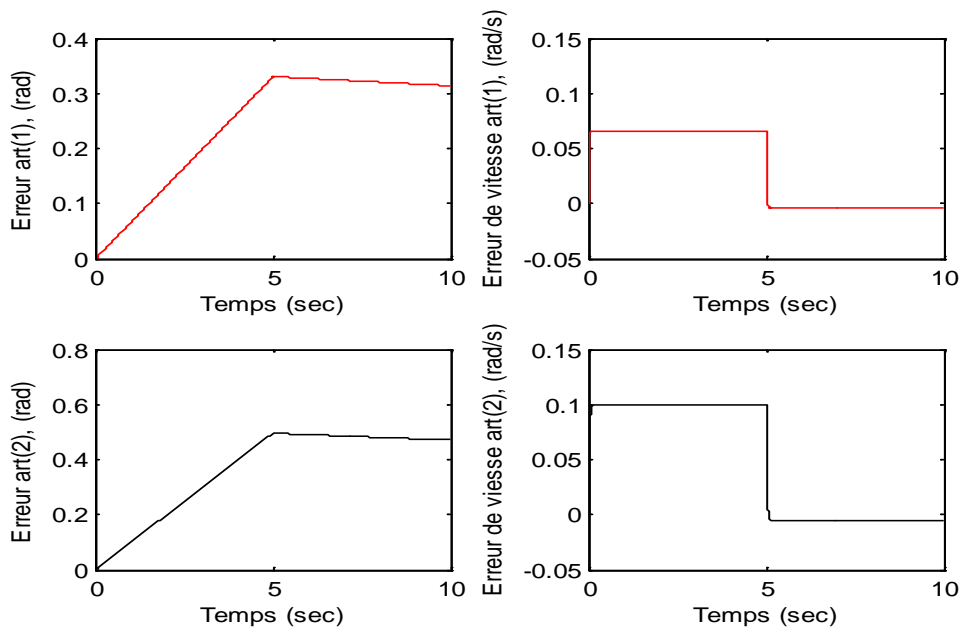
Mode glissant:



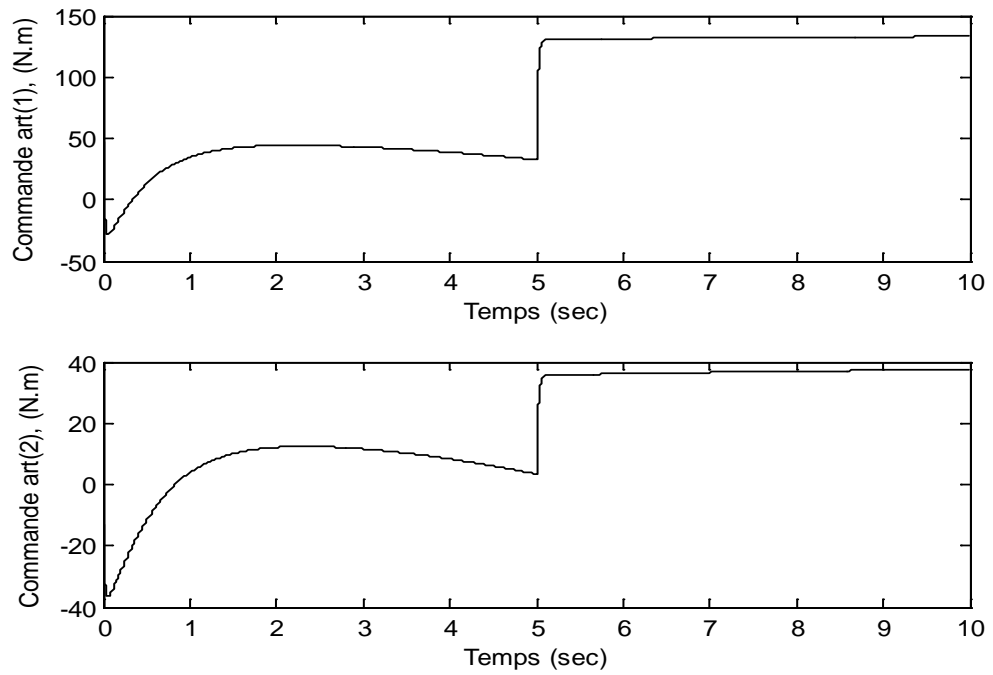
Figure(IV.23) : trajectoires désirée et réel (articulation 1)



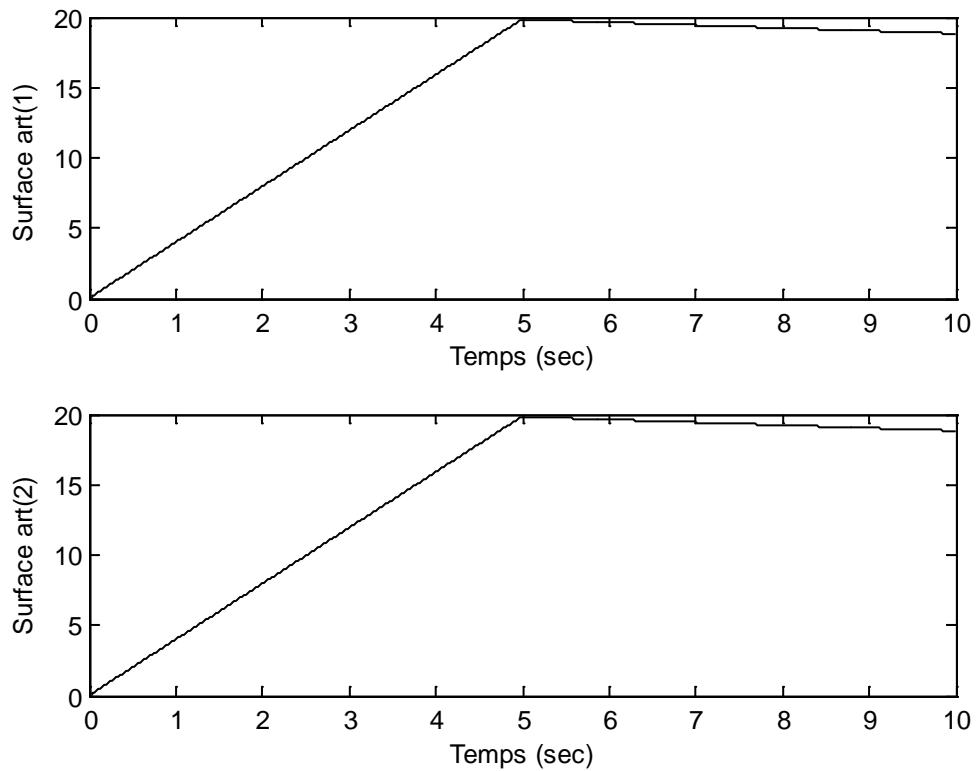
Figure(IV.24) : Trajectoires désirée et réelle (articulation 2)



Figure(IV.25) : Profil des erreurs de position et de vitesse (articulation 1 et 2)



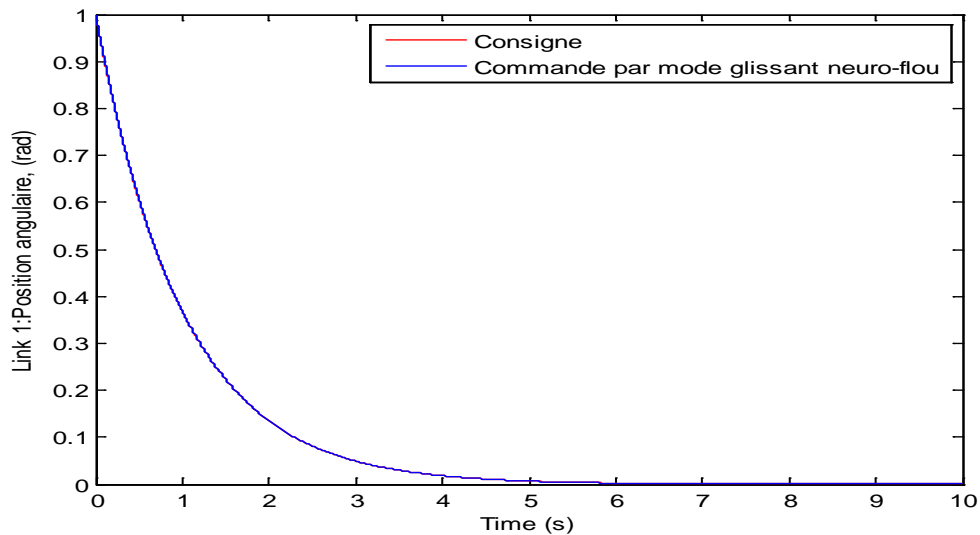
Figure(IV.26) : Evolution des commandes (articulation 1 et 2)



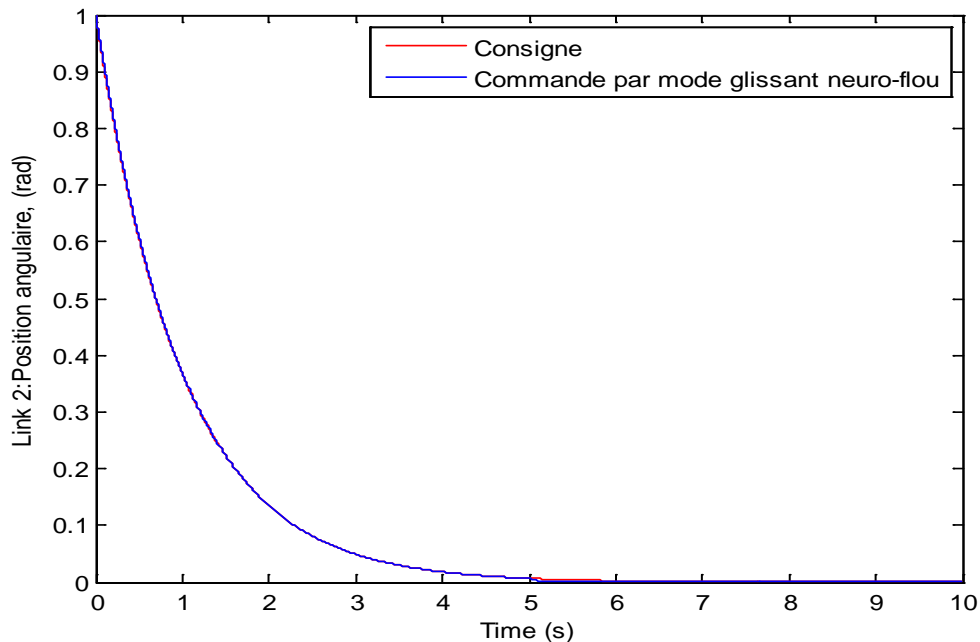
Figure(IV.27) : Evolution des surfaces de glissement (articulation 1 et 2)

Nous remarquons, que les surfaces de glissement ne convergent plus vers zéro, elles divergent complètement, la notion de mode glissant est perdue. Ce qui permet de conclure que la perturbation ne vérifie pas la condition de recouvrement, c-à-d que la perturbation n'agit pas dans les memes directions que la commande.

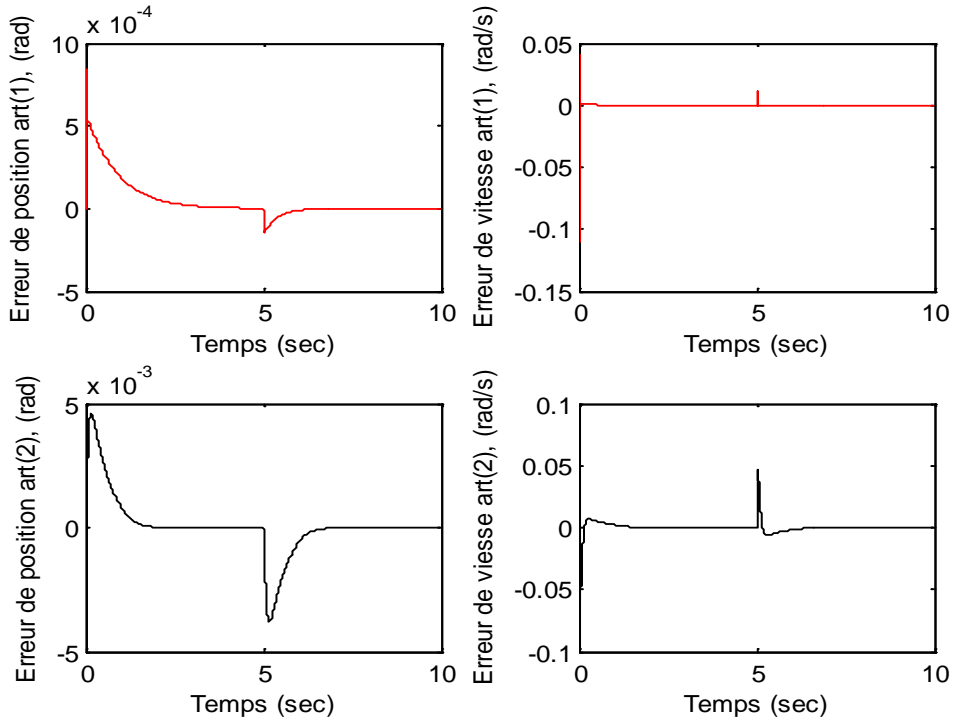
Commande par mode glissant neuro-flou



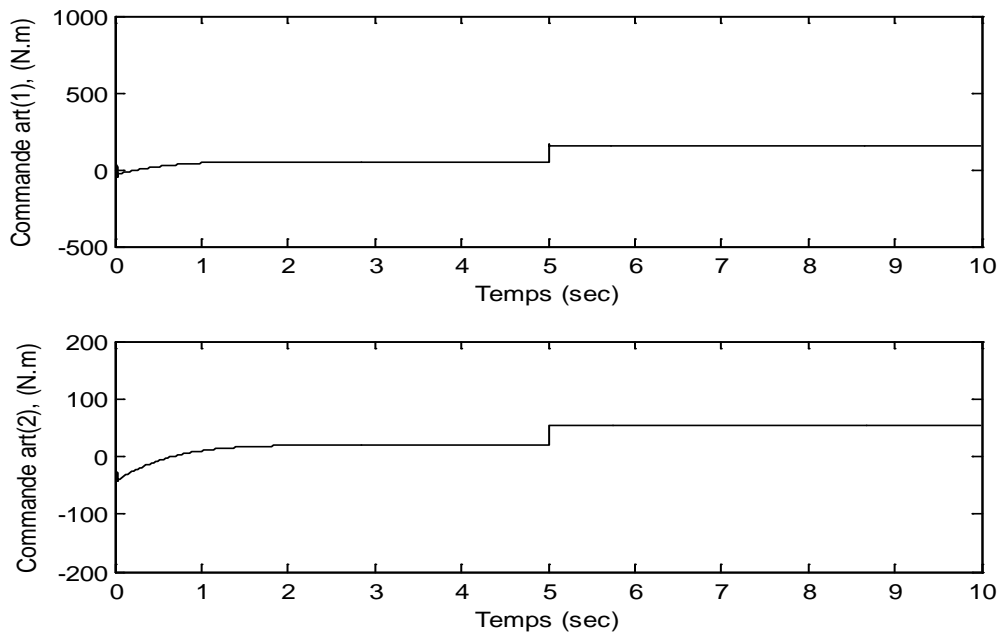
Figure(IV.28) : trajectoires désirée et réelle (articulation 1)



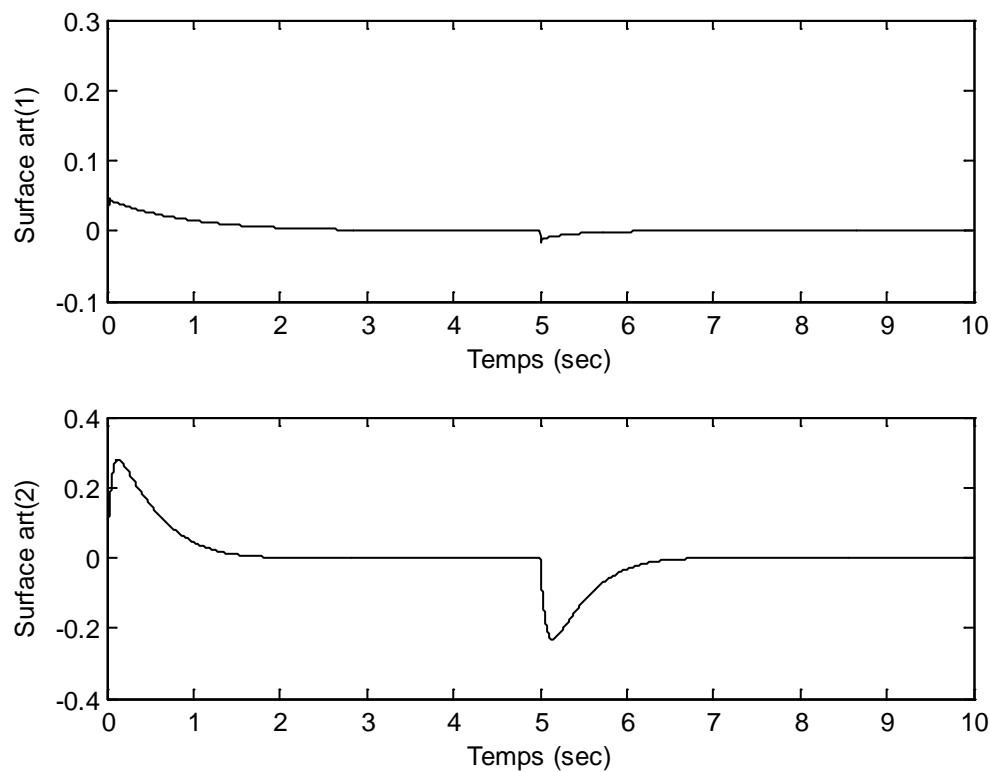
Figure(IV.29) : trajectoires désirée et réelle (articulation 2)



Figure(IV.30) : Profil des erreurs de position et de vitesse (articulation 1et 2)



Figure(IV.31) : Evolution des commandes (articulation 1 et 2)



Figure(IV.32) : Evolution des surfaces de glissement (articulation 1 et 2)

nous constatons que les performances sont très peu affectées par la perturbation. Le suivi de trajectoire est bien atteint. les surfaces de glissements convergent vers zéro. Nous notons aussi des pics qui apparaissent dans les erreurs de positions et de vitesses et sur les surfaces de glissement . Ceci se traduit par le rejet rapide de perturbation. Les deux commandes augmentent leurs amplitudes pour compenser la perturbation.

D'après les résultats obtenus, nous pouvons conclure que la commande par mode glissant neuro-flou est très efficace et robuste .

Conclusion

Dans ce chapitre, nous avons appliqué deux commandes, la commande par mode glissant d'ordre 1 et la commande hybride, pour le suivi de trajectoire d'un bras manipulateur à deux degrés de liberté.

Nous avons considéré le cas d'une simulation sans perturbation puis avec perturbation.

Pour remédier au phénomène de chattering, qui est due à l'existence des fonctions *sign* dans la structure des commandes d'excitation tout en gardant la robustesse, nous avons remplacé la fonction *sign* par une commande neuro-floue, et les résultats obtenus sont satisfaisants.

Dans ce travail, nous nous sommes intéressées aux problèmes de la commande robuste par mode glissant d'un robot manipulateur à deux articulations.

Dans le 1^{er} chapitre, nous avons donné des généralités sur les bras manipulateurs, où nous avons parlé un peu sur leur modèle géométrique et cinématique. Une grande partie a été consacrée à la dynamique de ces bras qui a été déduite par ce qu'on appelle le Lagrangien.

Dans le 2eme chapitre, nous avons exposé les principales caractéristiques de la commande par mode glissant d'ordre un. Nous nous sommes intéressées à présenter la condition d'existence du régime glissant, la notion de commande équivalente qui est une solution de l'équation en mode glissant et la méthode de synthèse de la loi de commande. La stratégie de commande élaborée est présentée pour définir le couple de référence à une commande par mode glissant. Cette stratégie permet d'associer les qualités de robustesse devant des incertitudes et perturbations. Néanmoins, cette commande reste limitée à cause de très fortes oscillations qui provoquent le phénomène de Chattering.

Dans le chapitre 3, nous avons décrit les concepts de la logique floue et des réseaux de neurones.

Dans le chapitre 4, Une nouvelle stratégie a été proposée (la commande neuro-floue), pour remédier au phénomène de chattering. Cette stratégie a été appliquée pour la commande d'un robot manipulateur à deux articulations. Les résultats de simulations ont montré une meilleure performance que ce soit en poursuite de trajectoires ou en robustesse, ce qui favorise largement la possibilité de réaliser une expérimentation implémentant les différentes commandes proposées.

Comme travaux futurs, on propose de poursuivre ce projet en estimant le gain de la commande discontinue par la logique floue.

- [1] **BOUNOUARA NADIA**, « planification de trajectoire pour les bras manipulateur » thèse de Magister, Université de BATNA, promotion 2010/2011
- [2] **GHENNAM Abdellatif**, « Commande Complainte Intelligente d'un Bras Manipulateur Rigide pour des Application de Chirurgie Médicale » thèse de Magister, Université de BATNA, promotion 2012
- [3] **N. Manamanni, A. Hamzaoui et N. Essounbouli**, Sliding mode control with adaptive fuzzy approximator for MIMO uncertain systems, European Control Conference, ECC'03, Cambridge (UK), 2003.
- [4] **L.-X. Wang**, A Course in Fuzzy Systems and Control, Prentice Hall, 1997.
- [5] **B. Yoo et W. Ham**, Adaptive fuzzy sliding mode control of nonlinear system, IEEE Transactions on Fuzzy Systems, vol. 6, no. 2, pp. 315-321, 1998.
- [6] **C.F. Hsu, T.-T. Lee, C.-M. Lin et L.-Y. Chen**, Robust neuro-fuzzy controller design via sliding-mode approach, IEEE International Conference on Fuzzy Systems, Budapest (Hongrie0), vol. 2, pp. 917- 922, 2004.
- [7] **Bechka Larbi**, «Commande Adaptative d'un Robot Manipulateur Flexible A Deux Degrés de Libertés». Thèse de Magister. Université de Batna.2007
- [8] **Benhadj belkacem**, «commande d'un bras manipulateur par mode glissant d'ordre supérieur». Thèse de Master. Université Mouloud Mammeri Tizi-ouzou.2012
- [9] **HAMIZI Abdelhakim**, «Commande Adaptative Floue Type-2 d'un Bras Manipulateur». Thèse de Magister. Université de Batna.2011
- [10] **WISAMA Khalil, ETIENNE Dombre** «Modélisation identification et commande des robots». Edition :HERMES Science publication 1999.
- [11] **Tomas SALGADO JIMENEZ** «Contribution à la commande d'un robot sous-marin autonome de type torpille». Thèse de Doctorat. École Centrale De NANTES.2010
- [12] **SI AMMOUR Amar**, «contribution à la commande par modes glissants d'ordre fractionnaire ». Thèse de Doctorat. Université Mouloud Mammeri Tizi-ouzou.2011
- [13] **Deghboudj Imen**, «Commande Des Systèmes Non Linéaires Par Mode Glissant D'ordre Supérieur». Thèse de Magister. Université Constantine 1. 2013
- [14] **A. Isidori**, "Nonlinear Control Systems", Springer-Verlag Communication and Control Engineering, ISBN 3-540-19916-0 3rd edition, 1995.
- [15] **H. Bühler**, "Réglage par mode de glissement ", traité d'électricité, 1ère édition, presses, 1991.
- [16] **Slotine, J.-J. E., & Li, W.** "Applied Nonlinear Control". London: Prentice-Hall, Inc. 1991

- [17] **Fillipov, AF.** (1960). Differentiel equations with discontinuous right-hand side. Amer.Math. Soc.Trans.62:199-231
- [18] **Itkis, U.**(1976). Control systems of variable structure J.Wiley, New York.
- [19] **Utkin, V.I.**(1978).Sliding modes and their application in variable structure systems.Mir,Moscou
- [20] **J-J. E. Slotine and W. Li.** Applied nonlinear control. Prentice Hall, 1991.
- [21] processurs flous , HE-ARC, BAPTISTE SAVOYE, 26,cH 2610 ST-IMMER, Mai 2008
- [22] P. Brone , J. Rozinore , J . Y .Dubois Introduction a la Logique Floue
<http://elap.montefiore.ulg.ac.be/fuzzy>
- [23] **GERALD Huguenin** ,processurs flous , HE-ARC, BAPTISTE SAVOYE, 26,cH 2610 ST-IMMER, Mai 2008
www.HE.ARC.CH
- [24] **FRANK Dernoncourt, Introduction** a la logique floue
www.openclassroom.com
- [25] **NAKOULA,Y**, Apprentissage des modèles linguistique Apport de la logique floue et Neuro-floue thèse de Doctorat, Université de Savoie, France, Juillet 1997
- [26] **Nakoula,Y**. Apprentissage des modèles floue, par Jeu de Règles pondorées.Thèse de Doctorat, Université de Savoie, France ,Juiillet 1997
- [27] **Hansruedi Bithler** : Réglage par logique floue, presses Polytechnique et Université Romande,Larsane,1994.
- [28] **R.Mellah**, contribution la commande adaptative neuro floue .Application à la robotique, thèse de Doctorat, Université des Sciences et de Technologie Houari Boumediene. Alger. Mai 2006
- [29] **P.Borne,J.Rozinoere.J.Y.Dieulot.L.Dubois**,Fuzzy-logique ou Fuzzy control en terminologie anglosaxonne.introduction a la logique à la logique floue et la commande floue,Pierre Garbriel
<http://elap.montefiore.ulg.ac.be/fuzzy/>
- [30] **Merabti Halim**, Etude des systèmes flous à intervalle, thèse de Magistère Université Mentouri de CONSTANTINE ,03/12/2008
- [31] **T. Takagi, M. Sugeno**, “Fuzzy identification of systems and its applications to modeling and control”, IEEE Transactions on systems Man and cybernetics, Vol. 15, n° 1, 1985, pp. 116-132
- [32] **Berton,S.**Une approche neuronale du contrôle robotique utilisant la vision binoculaire par reconstruire tridimontionnelle,Thèse de Doctorat,Université de haute alsace U.F.R.des Science et Technique Juin 1999.
- [33] **Borne P.Haggege J.**les réseaux de neurone présentation et application , Paris :Edition Technip,2007
- [34] **ORFAN Aboulshamat Boulshamat**, Algorithme De Commande Adaptative Par Mode Glissant A Base De Réseaux De Neurones Pour Une Articulation Flexible Avec Non-Linéarité Dure, Université du Québec à Trois-Rivières, AVRIL 2003
- [35] Cours réseaux de neurone.2006
- [36] **Enic Davola etPatrick Naim** ; Des Réseaux de neurones ; Edition Eyrolles,1990
- [37] Yann Morere,Les Réseaux Neuro-Flou,17 Mai 2001

- [38] **Y. Shi and M. Mizumoto**, An improvement of neuro-fuzzy learning algorithm for tuning fuzzy rules”, *Fuzzy sets and Systems*, vol. 118, 2001, pp. 339-350
- [39] **Kouadri Boudjelthia Mohamed**, Réseau de Kohonen « les Carte Auto-Organisatrices ; Université des Sciences et de la Technologie d’Oran USTO M.B, 2011-2012.
- [40] **Prof,HO Tuong Vinh**, Réseaux de Neurones pour la Reconnaissance Des Forme PHAM Quang Dung Hanoi, 15 juillet 2005
- [41] **Marc Parizeau**. Réseaux de Neurones R ,GIF-21140 et GIF-64326 ,Université LAVAL. Automne 2004.
- [42] **M.T.Hogan, H.B.Demuth, O. de Jesus** ;An Introducton to The Use Of Neural Networks In Control
- [43] **Fabien ;M.** Introduction aux réseaux de neurones et leurs application en Automatique .Ecole des Mines de Paris ,décembre 2003.
- [44] **GUENOUNOU Ouahib** . Méthodologie de conception de contrôleurs intelligents par l'approche génétique- application à un bioprocédé, thèse Doctorat de l’université de TOULOUSE promotion 22 avril 2009
- [45]**Jyh-Shing R.,Chune-Tsai S.**,Neuro-fuzzy Modeling and controle.Proceding of the IEEE.vol.83;N 3 Marche 1995 pp 378-404
- [46] **Nakoula,Y.** Apprentissage des modèle linguistique floue ,par Je de règles Pondérés. Thèse de Doctorat , Université de Savoie,France,Juillet 1997
Hérault : «Réseaux de neurones Méthodologie et application

Résumer

Dans ce travail, nous nous sommes intéressées aux problèmes de la commande robuste par mode glissant d'un robot manipulateur à deux articulations.

Dans le 1^{er} chapitre, nous avons donné des généralités sur les bras manipulateurs, où nous avons parlé un peu sur leur modèle géométrique et cinématique. Une grande partie a été consacrée à la dynamique de ces bras qui a été déduite par ce qu'on appelle le Lagrangien.

Dans le 2eme chapitre, nous avons exposé les principales caractéristiques de la commande par mode glissant d'ordre un. Nous nous sommes intéressées à présenter la condition d'existence du régime glissant, la notion de commande équivalente qui est une solution de l'équation en mode glissant et la méthode de synthèse de la loi de commande.

La stratégie de commande élaborée est présentée pour définir le couple de référence à une commande par mode glissant. Cette stratégie permet d'associer les qualités de robustesse devant des incertitudes et perturbations. Néanmoins, cette commande reste limitée à cause de très fortes oscillations qui provoquent le phénomène de Chattering.

Dans le chapitre 3, nous avons décrit les concepts de la logique floue et des réseaux de neurones.

Dans le chapitre 4, Une nouvelle stratégie a été proposée (la commande neuro-floue), pour remédier au phénomène de chattering. Cette stratégie a été appliquée pour la commande d'un robot manipulateur à deux articulations. Les résultats de simulations ont montré une meilleure performance que ce soit en poursuite de trajectoires ou en robustesse, ce qui favorise largement la possibilité de réaliser une expérimentation implémentant les différentes commandes proposées.

Comme travaux futurs, on propose de poursuivre ce projet en estimant le gain de la commande discontinue par la logique floue.