



Université ABBES LAGHROUR Khenchela
Faculté des Sciences et de la Technologie
Département de Génie Industriel
جامعة عباس لغرور خنشلة
كلية العلوم والتكنولوجيا قسم
الهندسة الصناعية



N° Série :

Mémoire de fin d'étude

Pour l'obtention du diplôme de Master

Filière : Automatique

Spécialité : Automatique et informatique industrielle

THEME

**Modélisation et élaboration d'une
commande adaptative neuronale -
mode glissant d'un bras manipulateur**

*Réalisé par : - BOUDJIJ Rania
- GHOZLANE Hanin*

Soutenu le 15 / 09 / 2020 Devant le jury composé de:

Mr. A.BEDIAF

Mr. A.MALLEM

Madame. S.AOURAR

Président

Encadreur

Examinatrice

Université Abbes Laghrour-Khenchela

Université Abbes Laghrour-Khenchela

Université Abbes Laghrour-Khenchela

Promotion 2019/2020

Dédicace

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

Je dédie ce travail en signe de respect et de reconnaissance à :

Ma très chère mère et mon père,

Ma sœur et mes frères,

A mon encadreur Mr. MALLEM Ali,

A mes amis qui m'ont aidé Dounia, Wafaa, Sabah, Younes,

A tous mes amis,

A mes camarades de la section Master 2 Automatique.

Enfin à tous ceux qui me sont chers.

BOUDJIDJ Rania

Dédicace

Je dédie ce modeste travail à :

*A mon cher père, le plus noble qui par son courage a
consacré tous ses efforts, et ses moyens pour m'aider à accomplir*

Ce mémoire, est pour me faire réussir.

La personne la plus chère dans le monde, la lumière de

Ma vie, la source de tendresse, celle qui a sacrifié et a souffrent

Les plus belles années de sa vie pour me voir réussir.

‘‘Ma chère mère

A mon encadreur ‘MALLEM ALI ‘

A mon chère frère ‘SABER.’

A mes chères sœurs.

A tous ma famille, source d’espoir et motivation.

A tout mes chères amies, tout particulièrement ‘SABRINA’

Et ‘MARWA’.

A mon amie ‘CHERRATT NIDHALE’.

Chère amie avant d’être binôme’ RANIA BOUDJIJ ‘.

A vous cher lecteur.

Ghozlane Hanine

Résumé

Le travail présenté dans ce mémoire consiste à élaborer différentes techniques de commande pour un bras manipulateur à 2 ddl. La stratégie proposée a pour objectif de présenter et comparer les techniques conventionnelles (PID et mode glissant classique) avec des techniques hybrides (mode glissant neuronal). Nous utilisons un système de commande robuste capable de faciliter la tâche et de surmonter les incertitudes présentées dans le monde réel et d'aider le robot à suivre des trajectoires prédéfinies. On s'est orienté vers une approche de commande par mode neuronal pour le suivi de trajectoires en présence des perturbations, en se basant sur une approche combine la commande par mode glissant pour assurer une convergence rapide des erreurs et les réseau de neurones artificiels pour approximer la fonction non linéaire du système et les paramètres de la loi de commande et pour stabiliser les erreurs, en présence des perturbations.

Abstract

The work presented in this thesis consists in developing different control techniques for a 2 DOF. The proposed strategy aims to present and compare conventional techniques (PID and classic sliding mode) with hybrid techniques (neural sliding mode). We use a robust control system capable of making the job easier and overcoming uncertainties presented in the real world and helping the robot to follow predefined paths. We moved towards a neural mode control approach for tracking trajectories in the presence of perturbations, based on an approach that combines sliding mode control to ensure rapid convergence of errors and artificial neural networks to approximate the non-linear function of the system and the parameters of the control law and to stabilize the errors, in the presence of disturbances.

ملخص

ينكون العمل المقدم في هذه الأطروحة من تطوير تقنيات تحكم مختلفة لذراع مناوِر، حيث تهدف الإستراتيجية المقترحة إلى تقديم ومقارنة التقنيات التقليدية (المنظم ووضع الانزلاق الكلاسيكي) مع التقنيات الهجينة (وضع الانزلاق العصبي). حيث نستخدم نظام تحكم قوياً قادراً على تسهيل المهمة والتغلب على أوجه عدم اليقين المقدمة في العالم الحقيقي ومساعدة الروبوت على اتباع مسارات محددة مسبقاً. انتقلنا نحو نهج التحكم في الوضع العصبي لتتبع المسارات في حالة وجود اضطرابات، بناءً على نهج يجمع بين التحكم في الوضع المنزلق لضمان التقارب السريع للأخطاء والشبكات العصبية الاصطناعية لتقريب الوظيفة غير الخطية للنظام ومعلمات قانون التحكم وتثبيت الأخطاء، في حالة وجود اضطرابات.

SOMMAIRE

Remerciement	
Dédicace	
Résumé	
Sommaire	
Abréviations	
Liste des figures	
Liste des tableaux	
Introduction générale	01

CHAPITRE I

MODELISATION DES ROBTS MANIPULATEURS

I.1 Introduction	03
I.2 Matrices de transformations homogènes	03
I.2.1 Coordonnées homogènes	03
I.2.2 Transformations homogènes	04
I.2.3 Représentation d'un plan	04
I.2.4 La matrice de transformation homogène	06
I.3-Modèle Géométrique	07
I.3.1- Modèle géométrique direct (MGD)	07
I.3.2- Modèle géométrique inverse (MGI)	11
I.4-Modèle cinématique	13
I.4.1-Modélisation cinématique direct (MCD)	14
I.4.2-Modélisation cinématique inverse (MCI)	17
I.4.2- I.5-Modèle Dynamique	18
I.5.1 Formalisme de Lagrange	19
I.6-Conclusion	24

CHAPITRE II

LA COMMANDE PAR MODE GLISSANT

II.1 Introduction	25
II.2 Introduction aux systèmes de réglage à structure variable	26
II.2.1 Principe de la commande par mode glissant	26
II.2.2 Différentes structures du contrôle par mode glissement	26
II.3 Conception de la commande par mode glissement	28
II.3.1 Choix des surfaces de glissement	28
II.3.2 Condition de convergence	30
II.3.2.1 Fonctions directes de communication	30
II.3.2.2 Fonctions de Lypunov	30
II.3.3 Calcul de la loi de commande	30
II.3.3.1 Définitions des grandeurs de la commande	32
II.3.3.2 Applications de la commande	32
II.3.4 Élimination du phénomène de Chattering	33
II.3.4.1 Fonction SAT	35
II.3.4.2 Fonction SMOUTH	35
II.4 Concept générale de la commande par mode glissant terminal	36
II.4.1 Convergence en temps fini	37
II.4.2 Conception de la commande par mode glissant terminal	37
II.4.2.1 Description du système	37
II.4.2.2 Conception de la surface de glissement	37
II.4.3 Calcul de la loi de commande	38
II.5 Commande par mode glissant terminal rapide du système SIFO (FTSM)	39
II.5.1 Étude de la stabilité	41
II.6 Conclusion	43

CHAPITRE III

COMMANDE PAR MODE GLISSANT D'UN BRAS MANIPULATEUR

III.1. Introduction	44
III.2. Rappel sur les équations du mouvement	44
III.3. Commande de PID d'un bras manipulateur 2 degrés de liberté (2DDL)	45
III.3.1. Commande de PID dans l'espace articulaire	46
III.3.2. Commande de PID dans l'espace cartésien	49
III.4. Commande par mode glissant d'un bras manipulateur	51
III.4.1. Principe de la commande par modes glissants	51

III.4.2. Conception de la commande par mode de glissement	52
III.4.3. Application de la commande par mode de glissement pour la suite de trajectoire d'un bras à 2 DDL	57
III.5. Résultats et Simulation	59
II.6 Conclusion	67

CHAPITRE IV

COMMANDE MODE GLISSANT NEURONALE APPLIQUEE POUR LA POURSUIT D'UN RAS MANIPULATEUR

IV.1 Introduction	68
IV.2 Le neurone formel	68
IV.3 Le perceptron multicouche (Multi-Layer Perceptron, MLP)	69
IV.4 Apprentissage	70
IV.4.1 Définitions	70
IV.4.2 Type d'apprentissage	71
IV.4.3 Règles d'apprentissage	72
IV.5 Réseaux de neurones à base radiale (RBF)	73
IV.5.1 Introduction	73
IV.5.2 Architecture	73
IV.5.3 Apprentissage	75
IV.6 Avantage et limites des RNAs	76
IV.7 Application des RNAs	77
IV.8 L'approximation de fonctions par le RBF	78
IV.9 Application d'une commande mode glissant-neuronale adaptative pour un bras manipulateur à 2DDL	79
IV.9.1 Énoncé du problème	79
IV.9.2 Contrôle du mode siliding par rapport à l'approximation de $f(x)$	81
IV.9.2.1 Conception du contrôleur	81
IV.9.2.2 Analyse de stabilité	82
IV.10. Résultats et simulation	83
IV.11 Conclusion	86
Conclusion générale	87
Références bibliographiques	104

LISTE DES ABREVIATIONS :

MGD : Modèle géométrique direct.

MGI : Modèle géométrique inverse.

SMA : The state machine-based anti-spoofing.

MCD : Modélisation cinématique direct.

MCI : Modélisation cinématique inverse.

MDD : Modèle dynamique direct.

MDI : Modèle dynamique inverse.

PID : Proportionnelle intégral dérivé.

2DDL : Deux degré de liberté.

MC : Le mode convergence.

MG : Le mode glissement.

MRP : Le mode du régime permanent.

CSV : La commande à structure variable.

VSS : Un système à structure variable.

SAT : Fonction de saturation.

TSM : Synthèse sur la commande par mode glissant terminal.

FTSM : Synthèse sur la commande par mode glissant terminal rapide.

RNA : Réseau de neurone artificiel.

RBF : Radial Basis Fonctions.

MLP : Multi layer perceptron.

LISTE DES FIGURES

Chapitre I :

Figure I.1 Changement de repères	04
Figure I.2 Transformation d'un vecteur	06
Figure I.3 Structure d'un robot manipulateur à chaîne ouverte simple	09
Figure I.4 Paramètre de Denavit-Hartenberg	10
Figure I.5 Transformations entre l'organe terminal et le repère atelier	12
Figure I.6 La relation entre le MGD et le MGI	14
Figure I.7 Cas d'articulation prismatique	16
Figure I.8 Cas d'articulation <i>rotoïde</i>	16

Chapitre II

Figure II.1 Convergence du système glissant	25
Figure II.2 Configuration de la structure par communication d'une contre réaction d'état variable	28
Figure II.3 Linéarisation exacte de l'écart	29
Figure II.4 Trajectoire de l'état vis-à-vis de la surface de glissement	31
Figure II.5 Signale de commande équivalente	33
Figure II.6 Définition de la commande signe	35
Figure II.7 Représentation de la fonction SAT	36
Figure II.8 Représentation de la fonction SMOOTH	38

Chapitre III

Figure III.1 Schéma bloc d'un système asservi avec correcteur	47
Figure III.2 Schéma classique d'une commande PID dans l'espace articulaire	49

Figure III.3 Schéma d'une commande PID dans l'espace cartésien	52
Figure III.4 Différents modes des trajectoires d'état	54
Figure III.5 La commande équivalente	58
Figure III.6 La fonction signe	59
Figure III 7. Position articulaire de la jonction 1	62
Figure III 8. Position articulaire de la jonction 2	62
Figure III 9. Vitesse articulaire de la jonction 1	63
Figure III 10. Vitesse articulaire de la jonction 2	63
Figure III.11. Signaux de commande (N/m)	63
Figure III.12. Génération d'une trajectoire carrée	64
Figure III.13. Génération d'une trajectoire circulaire	65
Figure III 15. Position articulaire de la jonction 2	65
Figure III 16. Vitesse articulaire de la jonction 1	66
Figure III 17. Vitesse articulaire de la jonction 2	66
Figure III.18. Signaux de commande (N/m)	66
Figure III 19. Position de poursuite des jonctions 1 et 2	67
Figure III 20. Vitesse de poursuite des jonctions 1 et 2	68
Figure III 21. Signaux de commande des jonctions 1 et 2	68

Chapitre IV

Figure IV.1 Un neurone formel	71
Figure IV.2 Un réseau de 03 couches	72
Figure IV.3 Architecture d'un réseau de neurone RBF	77
Figure IV 5. Position articulaire des jonctions 1 et 2	87

Figure IV. 6. Vitesse de poursuite des jonctions 1 et 2	87
Figure IV .7. Torque des jonctions 1 et 2	88
Figure IV .8. Signaux de la fonction $f(x)$ et leur fonction estimée	88

LISTE DES TABLEUX

Tableau IV.4 : Correspondance type de RNA-Domaine d'application	81
---	----

INTRDUCTION GENERALE

Le robot est une machine capable d'effectuer des tâches et de manipuler des objets selon un programme de façon automatique. Ils sont généralement utilisés pour remplacer les humains dans des situations où ces derniers sont incapables d'effectuer le travail, des situations plus dangereuses, de haute précision ou répétitives. [1]

Un robot manipulateur est considéré comme étant un système mécanique articulé, actionné et commandé. La modélisation du robot consiste à représenter son comportement par des équations algébriques, soit du point de vue des positions c'est le modèle géométrique, soit de point de vue des vitesses c'est le modèle cinématique, ou encore en considérant les efforts mis en jeu c'est le modèle dynamique.

La génération de mouvements pour un robot désigne la fonction de calcul des consignes désirées (articulaires ou cartésiennes) en fonction du temps afin de réaliser une tâche décrite par une trajectoire composée de positions successives de l'outil du robot et de contraintes cinématiques ou dynamiques. [2]

La plupart des robots industriels actuels, le mécanisme est considéré comme un système linéaire et chacune de ses articulations est asservie par une commande décentralisée. [3]

Dans ce travail, nous nous sommes intéressées par l'identification et la commande d'un bras manipulateur de deux degrés de liberté avec les commandes PID (dans l'espace cartésien et l'espace articulaire), la commande mode glissant tout en absence des perturbations, et une commande mode glissant neuronale adaptative en présence des perturbations.

La commande PID (proportionnel, intégral, dérivé) est permettant d'améliorer les performances sur sa valeur requise sans perturber le fonctionnement du système, C'est le régulateur le plus utilisé dans l'industrie où ses qualités de correction s'appliquent à de multiples grandeurs physiques. [4]

La commande par modes glissants est une commande à structure variable (CSV). Elle est par nature une commande non linéaire. La caractéristique principale des systèmes à structure variable est que la loi de commande se modifie d'une manière discontinue. [5]

Les réseaux de neurones sont des approximateurs universels constituent une nouvelle méthode d'approximation de systèmes complexes particulièrement utile lorsque c'est systèmes sont difficiles à modaliser à l'aide des méthodes statistique classique [6], l'objectif de notre travail est l'identification et commande d'un bras manipulateur on utilisant les

réseaux de neurones. Le robot avec son contrôleur et identificateur est simulé sous MATLAB/SIMULINK.

Nous avons présentée dans :

- Dans le premier chapitre nous avons présenté les différents types des modélisations d'un robot manipulateur,
- Le deuxième chapitre aborde les notions générales sur la commande mode glissant,
- Le troisième chapitre donne un aperçu sur les commande PID et mode glissant
- Le quatrième chapitre présente les réseaux de neurones artificiels ainsi une commande neuronale mode glissant.

CHAPITRE I

MODELISATION DES ROBOTS

MANIPULATEURS

Introduction

Plusieurs niveaux de modélisation sont possibles selon les objectifs, les contraintes de la tâche et les performances recherchées : modèles géométriques, cinématiques et dynamiques ; ces modèles, qu'ils soient utilisés en simulation ou en commande, doivent représenter le comportement du robot de façon réaliste. Les techniques proposées ont été développées dans ce souci et conduisent aux modèles les plus performants.

La conception et la commande des robots nécessitent le calcul de certains modèles mathématiques, tels que : les modèles de transformation entre l'espace opérationnel (dans lequel est définie la situation de l'organe terminal) et l'espace articulaire (dans lequel est définie la configuration du robot). [7]

On distingue trois modèles :

- les modèles géométriques direct et inverse, qui expriment la situation de l'organe terminal en fonction des variables articulaires du mécanisme et inversement.
- les modèles cinématiques direct et inverse qui expriment la vitesse de l'organe terminal en fonction des vitesses 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 et accélérations des articulations. [8]

Matrices de transformations homogènes

Coordonnées homogènes

- Un point (p) est représenté par :

Les coordonnées homogènes d'un point de l'espace projectif de dimension n sont écrites habituellement comme (P_x, P_y, P_z, P_w) .

Où : P_{σ} dans la robotique est égale à 1.

$$P = \begin{bmatrix} Px \\ Py \\ Pz \\ 1 \end{bmatrix} \quad (I.1)$$

- Représentation d'une direction (vecteurs libre) :

$$U = \begin{bmatrix} Ux \\ Uy \\ Uz \\ 0 \end{bmatrix} \quad (I.2)$$

- Représentation d'un plan :

le plan $\alpha x + \beta y + \gamma z = \delta$ est représenté par un vecteur $Q = [\alpha, \beta, \gamma, \delta]$; pour tout point appartenant à Q, $Q \cdot P = 0$. [9]

$$Q \cdot P = [\alpha, \beta, \gamma, \delta] \cdot \begin{bmatrix} Px \\ Py \\ Pz \\ 1 \end{bmatrix} = 0 \quad (I.3)$$

Transformation homogène

a. changement de repère

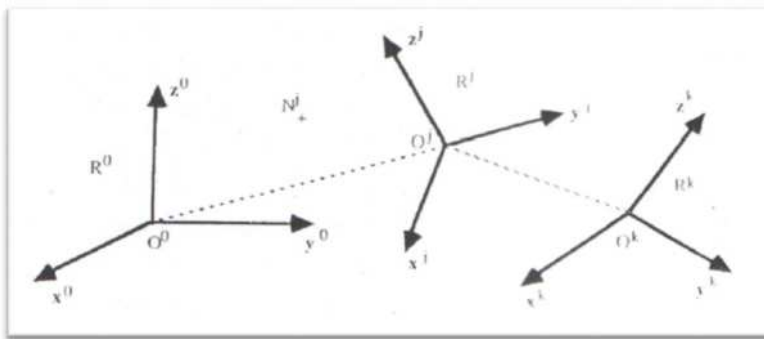


Figure I.1 : Changements de repères.

On définit la matrice de transformation homogène par : $T_j = [S_j, a_j^j P_j]$

$${}^i T_j = [{}^i s_j, {}^i n_j, {}^i a_j, {}^i p_j] = \begin{bmatrix} s_x & n_x & a_x & p_x \\ s_y & n_y & a_y & p_y \\ s_z & n_z & a_z & p_z \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (I.4)$$

${}^i s_j$, ${}^i n_j$ et ${}^i a_j$ sont les vecteurs unitaires des axes X_j , Y_j et Z_j du repère R_j exprimé dans R_i vecteur exprimant l'origine du repère R_j dans le repère R_i .

On écrit aussi :

$${}^i T_j = \begin{bmatrix} {}^i A_j & {}^i P_j \\ 0 & 0 & 0 & 1 \end{bmatrix} = \begin{bmatrix} {}^i s_j & {}^i n_j & {}^i a_j & {}^i p_j \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (I.5)$$

- La matrice A représente la matrice de rotation ou d'orientation du repère R_i par rapport à R_j .
- La matrice ${}^i T_j$ représente la transformation permettant de passer du repère R_i au repère R_j .
- Elle peut être interprétée comme étant un repère R_j (trois axes orthonormés et l'origine) exprimé dans R_i .
- La colonne P représente la translation du repère R_i par rapport au repère R_j . Dans le cas d'une translation pure $A = I_3$, tel que I est la matrice unité. [10]

Propriétés :

- La matrice A est orthogonale : $A^{-1} = A^T$.
- ${}^j T_i = ({}^i T_j)^{-1}$.
- $\text{Rot}(u, \theta)^{-1} = \text{Rot}(u, -\theta) = \text{Rot}(-u, \theta)$.
- $\text{Trans}(u, d)^{-1} = \text{Trans}(-u, d) = \text{Trans}(u, -d)$.

Transformation de vecteurs :

Soit un vecteur ${}^j p_1$ définissant le point P_1 dans le repère R_j (FigI.2). Compte tenu de la définition des coordonnées homogènes, on calcule les coordonnées du point P_1 dans le repère R_i grâce à l'équation suivante :

$${}^i p_1 = ({}^i O P_1) = {}^i s_j p_{1x} + {}^i n_j p_{1y} + {}^i a_j p_{1z} + {}^i p_j = {}^i T_j^j p_1 \quad (I.6)$$

La matrice ${}^i T_j$ permet donc d'exprimer dans le repère R_i les coordonnées d'un point données dans le repère R_j .

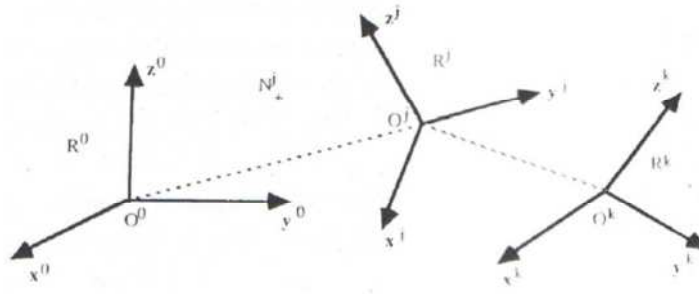


Figure I.2 : Transformation d'un vecteur.

les coordonnées homogènes

Les coordonnées homogènes d'un plan Q de l'équation :

$${}^i\alpha x + {}^i\beta y + {}^i\gamma z + {}^i\delta = 0 \quad (I.7)$$

Selon le repère R_i , sont données par le vecteur ligne :

$${}^iQ = [{}^i\alpha \quad {}^i\beta \quad {}^i\gamma \quad {}^i\delta]. \quad (I.8)$$

Si, un point P appartient à un plan Q, le produit matriciel ${}^iQ \cdot {}^iP$ est nul.

$${}^iQ \cdot {}^iP = [{}^i\alpha \quad {}^i\beta \quad {}^i\gamma \quad {}^i\delta] \begin{bmatrix} {}^iP_x \\ {}^iP_y \\ {}^iP_z \\ 1 \end{bmatrix} = 0. \quad (I.9)$$

I.2.4 la matrice de transformation homogène

La transformation d'un repère R_i au repère R_j est représentée par la matrice de transformation homogène iT_j de dimension (4×4) :

$${}^iT_j = [{}^i s_j \quad {}^i n_j \quad {}^i a_j \quad {}^i p_j] = \begin{bmatrix} s_x & n_x & a_x & p_x \\ s_y & n_y & a_y & p_y \\ s_z & n_z & a_z & p_z \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (I.10)$$

Ou :

- ${}^i s_j, {}^i n_j, {}^i a_j$: contiennent les composants des vecteurs unitaires x_j, y_j, z_j du repère R_j exprimées dans le repère R_i .
- ${}^i p_j$: représente les coordonnées de l'origine O_j de repère R_j exprimées dans R_i .

La matrice de transformation s'écrit aussi sous la forme :

$${}^i T_j = \begin{bmatrix} {}^i A_j & {}^i P_j \\ 0 & 0 & 0 & 1 \end{bmatrix} = \begin{bmatrix} {}^i s & {}^i n & {}^i a & {}^i p \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (\text{I.11})$$

La transformation :

Soit $Trans(a, b, c)$ la transformation d'une translation pure

Ou : a, b, c sont les translations le long des axes x, y, z . [8]

la matrice de transformation de cette translation à la forme :

$${}^i T_j = Trans(a, b, c) = \begin{bmatrix} 1 & 0 & 0 & a \\ 0 & 1 & 0 & b \\ 0 & 0 & 1 & c \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (\text{I.12})$$

Soit $Rot(x, \theta)$ la transformation correspondante à une rotation pure d'angle θ autour de l'axe

x .

➤ la matrice de transformation de cette rotation s'écrit :

$${}^i T_j = Rot(x, \theta) = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & \cos\theta & -\sin\theta & 0 \\ 0 & \sin\theta & \cos\theta & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} = \begin{bmatrix} Rot(x, \theta) & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (\text{I.13})$$

Ou, $rot(x, \theta)$ désignant la matrice d'orientation de dimension (3×3) .

➤ Les matrice de transformation des rotations autour des axes y :

$${}^i T_j = Rot(y, \theta) = \begin{bmatrix} \cos\theta & 0 & \sin\theta & 0 \\ 0 & 1 & 0 & 0 \\ -\sin\theta & 0 & \cos\theta & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} = \begin{bmatrix} Rot(y, \theta) & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (\text{I.14})$$

➤ Les matrice de transformation des rotations autour des axes z :

$${}^i T_j = Rot(z, \theta) = \begin{bmatrix} \cos\theta & -\sin\theta & 0 & 0 \\ \sin\theta & \cos\theta & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} = \begin{bmatrix} Rot(z, \theta) & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (I.15)$$

Modélisation Géométrique

La modélisation géométrique est l'ensemble des outils mathématiques, numériques, et informatiques qui combinés permettent de construire modèle virtuel (ou modèle informatique) d'un objet réel :

- plus ou moins complexe
- plus ou moins schématisé

L'objet peut :

- être le fruit de l'imagination, d'une tendance, ...
- une solution plus ou moins exacte d'un problème physique donné
- une combinaison des deux.

On distingue deux types de modèle géométrique :

- Modèle géométrique direct.
- Modèle géométrique inverse.

Modèle géométrique direct (MGD)

Le modèle géométrique direct est donc une fonction qui permet d'exprimer l'attitude de l'organe terminal en fonction des coordonnées articulaires

$$p(q) = \begin{bmatrix} T_q \\ \theta_q \end{bmatrix} = G_d(q). \quad (I.16)$$

Pour calculer l'équation géométrique directe d'un bras manipulateur à chaîne ouverte simple, une méthode systématique et générale doit être appliquée pour définir la position et l'orientation relative de deux corps consécutifs, Les méthodes de « Denavit-Hartenberg, Khalil, et Klein_nger » sont les plus répandus.

Le problème est comment déterminer deux repères attachés aux deux corps ? et comment calculer les transformations entre eux ? Cependant, il est commode de placer quelques hypothèses pour la définition des repères du corps. [11]

a. Convention de Denavit_Hartenberg

Cette méthode permet de normaliser, simplifier, et rationaliser la modélisation géométrique d'un robot.

Se reportant à la figure (Figure I.3), la structure est composée de n corps C_i , $i = 1$ à n , plus la base (C_0) reliés entre eux par des liaisons rotoïde ou prismatique, l'articulation i relie le corps C_i au corps C_{i-1} .

Afin de définir la relation entre les positions des corps ont associé à chaque corps C_i le repère $R_i (O_i ; X_i ; Y_i ; Z_i)$ tels que :

- L'axe X_i est porté par la perpendiculaire commune aux axes Z_i et Z_{i+1} .
- L'axe Y_i est formé par la règle droite pour compléter les coordonnées du système ($X_i ; Y_i ; Z_i$).
- L'axe Z_i est porté par l'axe de l'articulation i .

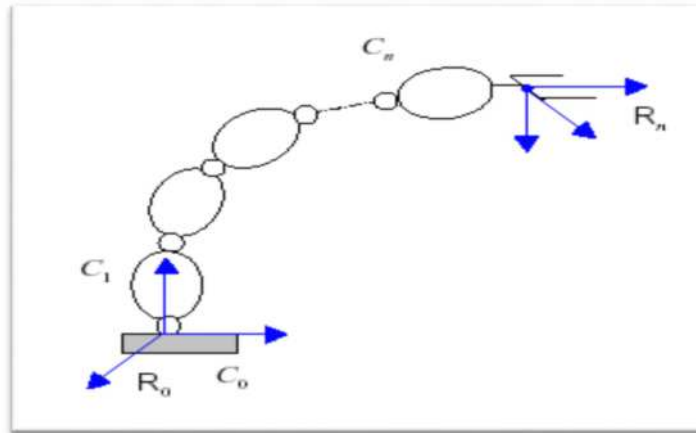


Figure I.3 : Structure d'un robot manipulateur à chaîne ouverte simple.

La matrice de transformation du repère R_{i-1} au repère R_i s'exprime en fonction des paramètres (de Denavit-Hartenberg) suivants (Figure I.4) :

- α_i : L'angle de rotation entre l'axe Z_{i-1} et l'axe Z_i autour de X_{i-1} .
- d_i : La distance entre l'axe Z_{i-1} et l'axe Z_i le long de l'axe Z_{i-1} .
- θ_i : L'angle de rotation entre les axes X_{i-1} et X_i autour de Z_i .
- r_i : la distance entre les axes Z_{i-1} et X_i le long de l'axe Z_i .

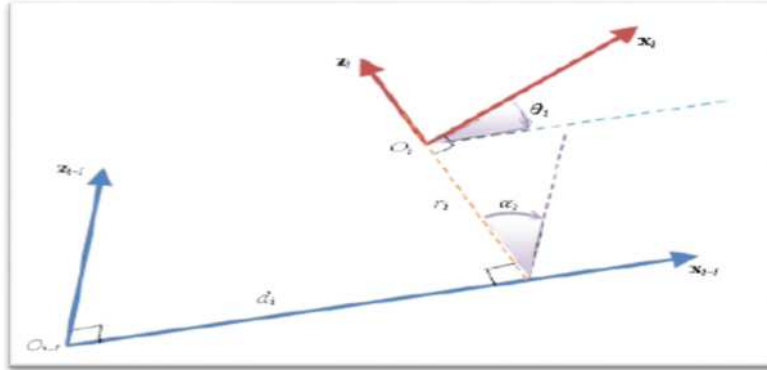


Figure I.4 : Paramètres de Denavit-Hartenberg.

La variable q_i de l'articulation i , définissant l'orientation ou la position relative entre les articulations $i-1$ et i , est soit θ_i , soit r_i , selon le type d'articulation est rotoïde ou prismatique respectivement. [12]

Ceci est défini par la relation suivante :

$$\overline{q}_i = \overline{\sigma}_i \theta_i + \overline{\sigma}_i r_i. \quad (\text{I.17})$$

Avec :

- $\sigma_i = 0$, si l'articulation i est rotoïde.
- $\sigma_i = 1$, si l'articulation i est prismatique.
- $\overline{\sigma}_i = 1 - \sigma_i$.

On définit la variable q_i par l'équation suivante :

$$\overline{q}_i = \overline{\sigma}_i \theta_i + \overline{\sigma}_i r_i. \quad (\text{I.18})$$

Et la matrice de transformation définissant R_i dans R_{i-1} est donnée par :

$${}^{i-1}T_i = Rot(x; \alpha_i) Trans(x; d_i) Rot(z; \theta_i) Trans(z; r_i). \quad (\text{I.19})$$

$${}^{i-1}T_i = \begin{bmatrix} \cos\theta_i & -\sin\theta_i & 0 & d_i \\ \cos\alpha_i \sin\theta_i & \cos\alpha_i \cos\theta_i & -\sin\alpha_i & -r_i \sin\alpha_i \\ \sin\alpha_i \sin\theta_i & \sin\alpha_i \cos\theta_i & \cos\alpha_i & r_i \cos\alpha_i \\ 0 & 0 & 0 & 1 \end{bmatrix}. \quad (\text{I.20})$$

Ou :

$${}^{i-1}T_i = \begin{bmatrix} & & d_i & \\ & {}^{i-1}A_i & -r_i \sin \alpha_i & \\ & & r_i \cos \alpha_i & \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (I.21)$$

Tell que : la matrice de rotation :

$${}^{i-1}A_i = \text{Rot}(x; \alpha_i) \text{Rot}(z; \theta_i). \quad (I.22)$$

La matrice de transformation définissant R_{i-1} dans R_i est donnée par :

$${}^i T_{i-1} = \begin{bmatrix} & & -d_i \cos \theta_i & \\ & {}^{i-1}A_i \hat{T} & d_i \sin \theta_i & \\ & & -r_i & \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (I.23)$$

Pour un robot à chaîne ouverte simple le MGD est défini par la matrice de transformation

0T_n :

$$T_n = {}^0T_1(q_1) {}^1T_2(q_2) {}^2T_3(q_3) \dots {}^{n-1}T_n(q_n) \quad (I.24)$$

Le MGD peut aussi être représenté par la relation :

$$X = f(q). \quad (I.25)$$

Où q est le vecteur des variables articulaires tels que :

$$q = [q_1 q_2 q_3 \dots q_n] \hat{T}. \quad (I.26)$$

Modèle géométrique inverse (MGI)

Le modèle géométrique inverse (MGI) consiste à déterminer les variables articulaires correspondant à une situation spécifique de l'organe terminal. [13]

Lorsqu'elles existent Trois méthodes de calcul de MGI sont répandues :

- La méthode de Paul, qui convient pour la plupart des robots industriels.
- La méthode de Pieper, qui permet de résoudre le problème pour les robots à six degrés de liberté avec trois articulations rotoïde ou trois articulation prismatiques. [13]
- La méthode de raghavan et Roth, donnant la solution générale des robots à six articulations à partir d'un polynôme de degré au plus égale 16. [14]

Remarque :

Quand le modèle géométrique inverse ne peut pas être obtenu, des techniques numériques peuvent être employées ; ces techniques emploient la méthode de Newton Raphson ou des méthodes fondés sur la transposé de la matrice Jacobienne.

Soit ${}^{i-1}T_E^d$ la matrice de transformation homogène représenté la position désiré du repère outil R_E par rapport au repère atelier R_f . En général on peut exprimer ${}^{i-1}T_{dE}$ sous la forme :

$${}^f T^d_E = Z \circ T_n(q)E. \quad (I.27)$$

Tels que :

- Z est la matrice de transformation définissant le repère de base du robot dans le repère atelier R_f .
- 0T_n est la matrice de transformation du repère terminal dans le repère R_0 .
- E est la matrice de transformation du repère outil R_E dans le repère terminal R_n .

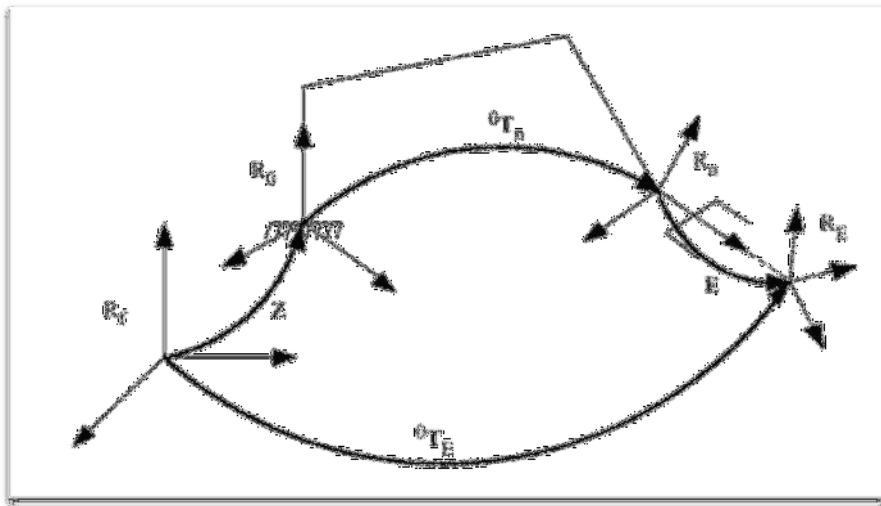


Figure I.5 : Transformations entre l'organe terminal et le repère atelier.

Mettant tous les termes connus dans le membre gauche, on obtient :

$$U_0 = {}^0T_n(q). \quad (I.28)$$

Avec :

$$U_0 = Z {}^{f-1}T^d_E E^{-1}. \quad (I.29)$$

Le MGD donnant :

$$X = f(q). \quad (I.30)$$

$$q = [q_1 q_2 \dots q_n]^T. \quad (I.31)$$

$$X = [X_1 X_2 \dots X_m]^T. \quad (I.32)$$

Tels que :

- n est le nombre de coordonnées opérationnelles
- m le nombre de coordonnées articulaires.

Le problème au-dessus (I.27) s'agit de résoudre un système de m équations à n inconnues, ce système étant non linéaire.

Le nombre de solutions dépend de l'architecture du robot manipulateur et de l'amplitude des articulations.

Trois cas se présentent pour calculer le MGI :

1. Solutions en nombre fini.
2. Absence de solution, lorsque la position de l'organe terminal désirée est en dehors de la zone accessible du robot.
3. Infinité de solutions lorsque :
 - le robot est redondant vis-à-vis la tâche.
 - le robot se trouve dans certaines configurations singulières.

Lorsqu'il est possible de calculer toutes les configurations permettant d'atteindre une situation donnée d'un robot manipulateur ce dernier est dit résoluble.

Méthode de Paul :

Soit la matrice de transformation homogène d'un robot manipulateur (I.24) :

$${}^0T_n(q) = {}^0T_1(q_1) {}^1T_2(q_2) {}^2T_3(q_3) \dots {}^{n-1}T_n(q_n) \quad (I.33)$$

Soit U_0 la situation désirée telle que :

$$U_0 = \begin{bmatrix} sx & nx & ax & px \\ sy & ny & ay & py \\ sz & nz & az & pz \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (I.34)$$

On cherche à résoudre le système d'équations suivant :

$${}^0T_n(q) = {}^0T_1(q_1) {}^1T_2(q_2) {}^2T_3(q_3) \dots {}^{n-1}T_n(q_n) \quad (I.35)$$

Pour résoudre le système (1.23), Paul a proposé une méthode (Méthode de Paul) qui consiste à pré-multiplier successivement les deux membres de l'équation par les matrices ${}^i T_{i-1}$

pour i de 1 à $n-1$. Ces opérations permettent d'isoler les variables d'articulations l'une après l'autres. [2]

$$\begin{aligned}
 U_0 &= {}^0T_1 {}^1T_2 {}^2T_3 \dots {}^{n-1}T_n \\
 {}^1T_0 U_0 &= {}^0T_1 {}^1T_2 {}^2T_3 {}^3T_4 {}^4T_5 \dots {}^{n-1}T_n. \\
 {}^2T_1 U_1 &= {}^2T_3 {}^3T_4 {}^4T_5 \dots {}^{n-1}T_n. \\
 {}^3T_4 U_2 &= {}^3T_4 {}^4T_5 \dots {}^{n-1}T_n. \\
 &\parallel \\
 {}^{n-1}T_{n-2} U_{n-2} &= {}^{n-1}T_n.
 \end{aligned}$$

Avec :

$$U_{j+1} = {}^{j+1}T_n = {}^{j+1}T_j U_j, \text{ pour } j = 0, \dots, n. \quad (I.36)$$



Figure I.6 : Mode de changement d'espace.

Modélisation cinématique

La cinématique complète la modélisation géométrique en établissant les relations entre les vitesses des paramètres articulation (q) et les vitesses des variables articulation.

Ces relations sont désignées par la notation $\dot{X} = J(q) \cdot \dot{q}$, où J est une matrice qui dépend de la configuration du SMA. La propriété évidente du modèle cinématique est sa linéarité par rapport aux vitesses ; il est donc a priori plus aisé à manipuler que le modèle géométrique.

On distingue deux types de modélisation cinématiques :

- Modèle cinématique direct.
- Modèle cinématique inverse.

Modélisation cinématique directe (MCD)

Le modèle cinématique directe d'un robot manipulateur décrit les vitesses des coordonnées opérationnelles (X°), en fonction des vitesses articulaires (q°).

$$X^\circ = J(q).q^\circ \tag{I.37}$$

Où: $J(q)$ désigne la matrice jacobienne de dimension $(m \times n)$, égale à $\frac{\partial X}{\partial q}$ « à partir du modèle différentiel direct $dX=J(q).dq$ ».

Le calcul de la matrice Jacobienne peut se faire en dérivant le MGD, $X = f(q)$ à partir de la relation suivante :

$$J_{ij} = \left(\frac{\partial f_i(q)}{\partial q_j} \right) i=1, \dots, m; j=1, \dots, n. \tag{I.38}$$

Où, J_{ij} est l'élément (i, j) de la matrice Jacobienne J .

Cette méthode est pratique pour des robots simples avec un nombre réduit de degrés de liberté.

Le calcul de la matrice Jacobienne de base, connu sous le nom de « matrice Jacobienne cinématique », est plus pratique pour un robot général de degré de liberté égale n .

La matrice Jacobienne obtenue relie les vecteurs des vitesses de translation et de rotation V_n et ω_n et les vitesses articulaires :

$$\begin{bmatrix} V_n \\ \omega_n \end{bmatrix} = J_n.q^\circ. \tag{I.39}$$

Où, V_n et ω_n sont les vitesses linéaire et angulaire du repère R_n respectivement.

La vitesse q°_k de l'articulation k produit une vitesse linéaire ($V_{k,n}$) et une vitesse angulaire ($\omega_{k,n}$) sur le repère terminal R_n .

Deux cas se présentent :

1- Si l'articulation k est prismatique $\sigma_k = 1$, (Figure I.7) :

$$\begin{cases} V_{k,n} = a_k q_k^\circ \\ \omega_{k,n} = 0. \end{cases}$$

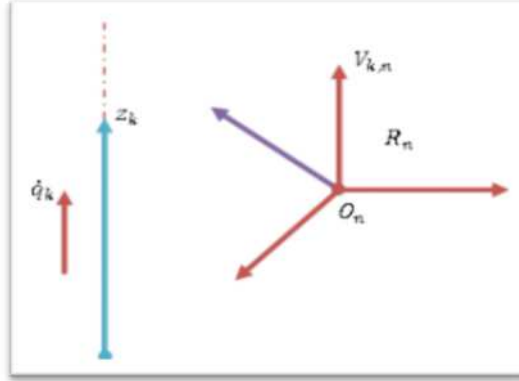


Figure I.7: Cas d'articulation prismatique.

Où, a_k est le vecteur unitaire porté par l'axe z_k de l'articulation k.

2- Si l'articulation k est rotoïde $\sigma_k = 0$ (Figure I.8) :

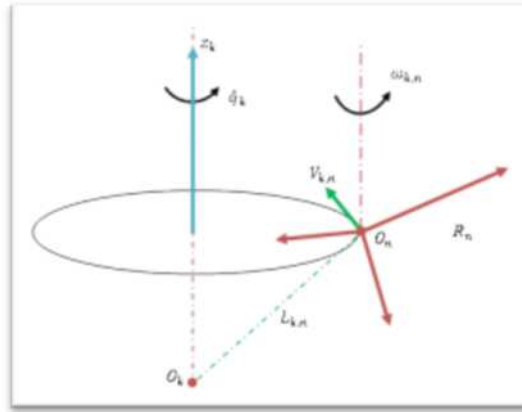


Figure I.8 : Cas d'articulation rotoïde.

$$\begin{cases} V_{k,n} = a_k q_k^\circ \times L_{k,n} (a_k \times L_{k,n}) q_k^\circ \\ \omega_{k,n} = a_k q_k^\circ \end{cases} \quad (I.40)$$

Le terme $L_{k,n}$ désigne le vecteur $\overrightarrow{O_k O_n}$.

La forme générale des vecteurs $V_{k,n}$ et $\omega_{k,n}$ s'écrivent:

$$\begin{cases} V_{k,n} = [\sigma_k a_k + \bar{\sigma}_k (a_k \times L_{k,n})] \dot{q}_k \\ \omega_{k,n} = \bar{\sigma}_k a_k \dot{q}_k \end{cases} \quad (I.41)$$

Les vitesses linéaires et angulaires de l'organe terminal peuvent être écrites comme :

$$\begin{cases} V_n = \sum_{k=1}^n V_{k,n} = \sum_{k=1}^n \sigma_k [a_k + \bar{\sigma}_k (a_k \times L_{k,n})] \dot{q}_k \\ \omega_n = \sum_{k=1}^n \omega_{k,n} = \sum_{k=1}^n \bar{\sigma}_k a_k \dot{q}_k \end{cases} \quad (I.42)$$

Se référant les vecteurs de J_n par rapport au repère R_i , on obtient la matrice jacobienne ${}^i J_n$, telle que :

$$\begin{bmatrix} V_n \\ \omega_n \end{bmatrix}_i = \mathcal{F}_n^i \dot{q}_n \quad (I.43)$$

L'écriture de l'équation (I.43) sous forme d'une matrice en utilisant l'équation (I.42), donne :

$$J_n = \begin{bmatrix} \sigma_1 a_1 + \bar{\sigma}_1 (a_1 \times L_{1,n}) & \dots & \sigma_n a_n + \bar{\sigma}_n (a_n \times L_{n,n}) \\ \bar{\sigma}_1 a_1 & \dots & \bar{\sigma}_n a_n \end{bmatrix} \quad (I.44)$$

En général, on calcule V_n et ω_n dans R_n et R_0 , la matrice jacobienne correspondante est ${}^n J_n$ ou ${}^0 J_n$ respectivement ; ces matrices peuvent être aussi calculées en utilisant une matrice ${}^i J_n$, $j=0, \dots, n$, grâce à l'expression suivante :

$${}^s J_n = \begin{bmatrix} {}^s A_i & 0_3 \\ 0_3 & {}^s A_i \end{bmatrix} \quad (I.45)$$

Où : ${}^s A_i$ est la matrice d'orientation du repère R_i dans R_s .

On note que les matrices ${}^i J_n$ ayant les mêmes positions singulières. [15]

Calcul de la matrice ${}^i J_n$:

Le produit $(a_k \times K_{k,n})$ peut être calculé par $(\hat{a}_k L_{k,n}^2)$, la $k^{\text{ème}}$ colonne de la matrice ${}^i J_n$ notée par ${}^i J_{n:k}$ devienne :

$${}^i j_{n:k} = \begin{bmatrix} \sigma_k^i a_k + \overline{\sigma}_k^i A_k^k \hat{a}_k^k L_{k,n} \\ \overline{\sigma}_k^i a_k \end{bmatrix} \quad (\text{I.46})$$

on pose ${}^k a_k = [0 \ 0 \ 1]^T$, et ${}^k L_{k,n} = {}^k P_n$, donc on obtient :

$${}^i j_{n:k} = \begin{bmatrix} \sigma_k^i a_k + \overline{\sigma}_k^i (-{}^k p_{ny} {}^i s_k + {}^k p_{nz} {}^i n_k) \\ \overline{\sigma}_k^i a_k \end{bmatrix} \quad (\text{I.47})$$

Où : ${}^k P_{ny}$ et ${}^k P_{nz}$ sont les composants du vecteur ${}^k P_n$.

A partir de cette expression, la k^{ime} colonne de la matrice ${}^i J_n$ est :

$${}^i j_{n:k} = \begin{bmatrix} \sigma_k^i a_k + \overline{\sigma}_k^i \hat{a}_k^i ({}^i p_n - {}^i p_k) \\ \overline{\sigma}_k^i a_k \end{bmatrix} \quad (\text{I.48})$$

qui donne pour $i = 0$:

$${}^0 j_{n:k} = \begin{bmatrix} \sigma_k^0 a_k + \overline{\sigma}_k^0 \hat{a}_k^0 ({}^0 p_n - {}^0 p_k) \\ \overline{\sigma}_k^0 a_k \end{bmatrix} \quad (\text{I.49})$$

Dans ce cas on est besoin de calculer les matrices ${}^0 T_k$ pour $k=1, \dots, n$.

Modélisation cinématique inverse (MCI)

Le modèle cinématique inverse (MCI) donne les vitesses articulaires q° correspondants à une vitesse désirée X° de l'organe terminal. Le modèle cinématique inverse s'obtient par la solution d'un système d'équations linéaires soit analytiquement, soit numériquement.

Les solutions analytiques diminuent le nombre d'opérations de façon remarquable par rapport aux solutions numériques, mais il faut traiter les cas singuliers séparément.

Les solutions numériques sont plus générale set traitent tous les cas de la même façon.

Soit $X = \begin{bmatrix} X_p^T & X_r^T \end{bmatrix}$ une représentation de la situation du repère R_n dans le repère R_0 .

Où X_p^T et X_r^T désignent respectivement la position et l'orientation opérationnelles.

Les relations entre les vitesses X_p° et X_r° et les vecteurs ${}^0 V_n$ et ${}^0 \omega_n$ sont :

$$\begin{bmatrix} X_p^\circ \\ X_r^\circ \end{bmatrix} = \begin{bmatrix} \Omega_p & 0_3 \\ 0_3 & \Omega_r \end{bmatrix} \begin{bmatrix} V_n^\circ \\ \omega_n^\circ \end{bmatrix} = \Omega \begin{bmatrix} V_n^\circ \\ \omega_n^\circ \end{bmatrix} \quad (I.50)$$

Remarque : La matrice \hat{a} est définie par :

$$\hat{a} = \begin{bmatrix} 0 & -a_z & a_y \\ a & & -a_x \\ z & & \\ -a_y & a_x & 0 \end{bmatrix} \quad (I.51)$$

A partir de l'équation (I.37) le MCD s'écrit sous la forme :

$$X^\bullet = Jq^\bullet \quad (I.52)$$

La méthode la plus générale consiste à calculer J^{-1} la matrice inverse de J, qui permet de déterminer les vitesses articulaires q^\bullet grâce à la relation :

$$q^\bullet = J^{-1} X^\bullet \quad (I.53)$$

Lorsque la matrice J a la forme :

$$j = \begin{bmatrix} A & 0 \\ B & C \end{bmatrix} \quad (I.54)$$

Tel que : les matrices A et C étant carrées inversibles, il est facile de montrer que l'inverse de cette matrice s'écrit :

$$j^{-1} = \begin{bmatrix} A^{-1} & 0 \\ -C^{-1}BA^{-1} & C^{-1} \end{bmatrix} \quad (I.55)$$

La résolution du problème se ramène donc à l'inversion de la matrice J. [7]

Modélisation Dynamique :

Le modèle dynamique est la relation entre les couples (et/ou forces) appliqués aux actionneurs et les positions, vitesses et accélérations articulaires. [16]

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

$$\Gamma = f(q, \dot{q}, \ddot{q}, fe).$$

Avec :

- Γ : vecteur des couples/forces des actionneurs, selon que l'articulation est rotoïde ou prismatique. Dans la suite, on écrira tout simplement couples.
- q : vecteur des positions articulaires.
- \dot{q} : vecteur des vitesses articulaires.
- \ddot{q} : vecteur des accélérations articulaires.
- f_e : vecteur représentant l'effort extérieur (forces et moments) qu'exerce le robot sur l'environnement.

Le modèle dynamique direct (MDD) est celui qui exprime les accélérations en fonction des positions, vitesses, et couples des articulations ; il est alors représenté par la relation :

$$\ddot{q} = g(q, \dot{q}, G, f_e). \quad (\text{I.56})$$

Et on convient d'appeler modèle dynamique inverse (MDI), ou tout simplement modèle dynamique, la relation de la forme :

$$\Gamma = f(q, \dot{q}, \ddot{q}, f_e) \quad (\text{I.57})$$

Le modèle dynamique joue un rôle important dans la conception et le fonctionnement des robots. Pour la conception, le modèle dynamique inverse peut être utilisé pour choisir les actionneurs, alors que le modèle dynamique direct est utilisé pour effectuer des simulations, afin de tester les performances du robot. En ce qui concerne les fonctionnalités du robot, le modèle dynamique inverse est utilisée pour calculer les couples actionneurs, qui sont nécessaires pour réaliser un mouvement souhaité. Il est également utilisé pour identifier les paramètres dynamiques qui sont nécessaires à la fois pour le contrôle et la simulation.

Plusieurs approches sont proposées pour obtenir le modèle dynamique des robots. Les plus souvent utilisé dans la robotique sont le formalisme de Lagrange, et le formalisme de Newton-Euler. L'approche de Newton-Euler est basée sur les forces et les moments agissent entre les liens. [17]

La formulation de Newton-Euler peut être considérée comme une approche basée sur l'équilibre des forces, la formulation de Lagrange est une approche basée sur l'énergie. Bien sûr, pour le même manipulateur, les deux donnent les mêmes équations du mouvement.

Dans cette étude on présente le formalisme de Lagrange et on considère que les robots à chaîne ouverte simple.

I.5.1 Formalisme de Lagrange

On considère un système idéal sans frottement ou élasticité, exerçant ni des forces ni des moments sur l'environnement.

La formulation de Lagrange décrit le comportement d'un système dynamique en terme d'énergie

Les équations de Lagrange sont généralement écrites sous la forme :

$$\Gamma = \frac{d}{dt} \frac{\partial L}{\partial \dot{q}^o} - \frac{\partial L}{\partial q} \quad (\text{I.58})$$

Où L est le Lagrangien du système, se définit comme la différence entre l'énergie cinétique K et l'énergie potentielle P :

$$L = K - P. \quad (\text{I.59})$$

Pour obtenir l'équation dynamique générale du robot, on détermine les énergies cinétiques et potentielles, et le lagrangien, puis les remplacés dans l'équation de Lagrange. [18]

Dans notre usage, q sera le vecteur des variables articulaires, se composant de l'angle d'articulation θ_i et le déplacement de l'articulation d_i . Alors que Γ est un vecteur qui a comme composants les couples n_i correspondants aux angles θ_i , et les forces f_i correspondantes aux déplacements d_i .

L'énergie cinétique :

L'énergie cinétique k_i de l' $i^{\text{ème}}$ articulation est donnée par l'expression :

$$k_i = \frac{1}{2} m_i v_{i C_i}^T v_{i C_i} + \frac{1}{2} \omega_i^T I_i^{C_i} \omega_i \quad (\text{I.60})$$

Où le premier terme est l'énergie cinétique due à la vitesse linéaire et le deuxième terme est l'énergie cinétique due à la vitesse angulaire. L'énergie cinétique totale du manipulateur est la somme de l'énergie cinétique des différentes articulations

C'est-à-dire :

$$K = \sum_{i=1}^n K_i \quad (\text{I.61})$$

Les vitesses $v_{C_i}^T$ et ${}^i\omega_i^T$ sont des fonctions de q et \dot{q} respectivement, donc on voit que l'énergie cinétique d'un manipulateur peut être décrite par une formule en fonction de la position et de la vitesse $k(q, \dot{q})$. En fait, l'énergie cinétique d'un manipulateur est donnée par :

$$K(q, \dot{q}) = \frac{1}{2} \dot{q}^T M(q) \dot{q}. \quad (\text{I.62})$$

Où $M(q)$ est une matrice ($n \times n$) de l'énergie cinétique appelée matrice d'inertie. Une expression de la forme ci-dessus est appelé la forme quadratique. L'énergie cinétique doit toujours être positive, pour cela la matrice d'inertie du manipulateur doit être une matrice définie positive.

L'énergie potentielle :

L'énergie potentielle de l' $i^{\text{ème}}$ articulation, à l'expression suivante :

$$p_i = m_i {}^0g^T {}^0T_{C_i} + p_{refi} \quad (\text{I.63})$$

Où 0g est le vecteur de gravité (3×1), ${}^0T_{C_i}$ une transformation homogène localisant le centre de la $i^{\text{ème}}$ articulation, et p_{refi} est une constante choisie de sorte que la valeur minimum de p_i soit 0. L'énergie potentielle totale est la somme de l'énergie potentielle dans les différentes articulations :

$$P = \sum_{i=1}^n p_i \quad (\text{I.64})$$

Puisque ${}^0T_{C_i}$ sont en fonction de q_1, q_2, \dots, q_n , l'énergie potentielle est en fonction de la position de l'articulation, $P(q)$.

Donc le Lagrangien sera :

$$L(q, \dot{q}) = \dot{q}K(q, \dot{q}) - P(q) \quad (\text{I.65})$$

Equation du mouvement d'un robot manipulateur :

Se reportant à l'équation (3.1) le Lagrangien a la forme :

$$L(q, \dot{q}) = \frac{1}{2} \dot{q}^T M(q) \dot{q} - P(q) \quad (\text{I.66})$$

Où $M(q)$ est la matrice d'inertie du manipulateur et $P(q)$ est l'énergie potentielle due au gravité. Il est convenable d'écrire l'énergie cinétique sous forme d'une somme :

$$L(q, \dot{q}) = \sum_{i,j=1}^n M_{i,j}(q) \dot{q}_i \dot{q}_j - P(q) \quad (\text{I.67})$$

En substituant dans l'équation de Lagrange, les équations de mouvements sont données par :

$$\Gamma_i = \frac{d}{dt} \frac{\partial L}{\partial \dot{q}_i} - \frac{\partial L}{\partial q_i} \quad (\text{I.68})$$

Où Γ_i sont les forces (couples) agissant sur l'articulation i , en utilisant l'équation (I.67) on a :

$$\frac{d}{dt} \frac{\partial L}{\partial \dot{q}_i} = \frac{d}{dt} \left(\sum_{j=1}^n M_{ij}(q) \dot{q}_j \right) = \sum_{j=1}^n (M_{ij}(q) \ddot{q}_j + \dot{M}_{ij}(q) \dot{q}_j) \quad (\text{I.69})$$

$$\frac{\partial L}{\partial q_i} = \frac{1}{2} \sum_{k,j=1}^n \frac{\partial M_{ij}(q)}{\partial q_i} \dot{q}_k \dot{q}_j - \frac{\partial P(q)}{\partial q_i} \quad (\text{I.70})$$

Le terme $M(q)$ peut être augmenté en termes de dérivés partiels.

$$\Gamma_i = \sum_{j=1}^n M_{ij}(q) \ddot{q}_j + \sum_{j,k=1}^n \left(\frac{\partial M_{ij}(q)}{\partial q_k} \dot{q}_j \dot{q}_k - \frac{1}{2} \frac{\partial M_{kj}(q)}{\partial q_i} \dot{q}_k \dot{q}_j \right) + \frac{\partial P(q)}{\partial q_i} \quad (\text{I.71})$$

Afin de mettre les équations du mouvement sous forme d'un vecteur, on définit la matrice $C(q, \dot{q})$ tel que :

$$C_{ij}(q, \dot{q}) = \frac{1}{2} \sum_{k=1}^n \left(\frac{\partial M_{ij}}{\partial q_k} + \frac{\partial M_{ik}}{\partial q_j} - \frac{\partial M_{kj}}{\partial q_i} \right) \dot{q}_k \quad (\text{I.72})$$

Maintenant l'équation peut être écrite comme :

$$\Gamma = M(q) \ddot{q} + C(q, \dot{q}) \dot{q} + G(q). \quad (\text{I.73})$$

Ou :

- Γ : vecteur des couples appliquées aux articulations.
- $M(q)$: matrice d'inertie du robot.
- $C(q, \dot{q}) \dot{q}$: vecteur des forces de Coriolis et centrifuges.

- $G(q)$: vecteur des forces de gravité.

Pour un système avec frottement ou élasticité, le modèle dynamique est :

$$\Gamma = M(q)\ddot{q} + C(q, \dot{q})\dot{q} + G(q) + Fe \quad (I74)$$

Il s'agit d'une équation différentielle du second ordre pour le mouvement du manipulateur en fonction des couples articulaires appliqués. [19]

Les matrices M et C qui résument les propriétés d'inertie du manipulateur, ont les propriétés :

- La matrice d'inertie $M(q)$ est symétrique et définie positive.
- La matrice $M(q)$ est donnée comme suit :

$$M_m I \leq M(q) \leq M_M I .$$

Avec M_m et M_M des scalaires positives, et I la matrice d'inertie

- L'inverse de la matrice $M(q)$ est borné :

$$\frac{1}{M_M} \leq M^{-1}(q) \leq \frac{1}{M_m} . \quad (I.75)$$

- La matrice $N(q, \dot{q}) = M(q) - 2C(q, \dot{q})$ est antisymétrique.
- La matrice $C(q, \dot{q})$ vérifie la relation suivante :

$$C(q, x)y = C(q, y)x . \quad (I.76)$$

avec x, y les vecteurs de vitesse de dimension $(n \times 1)$.

- la matrice $C(q, \dot{q})\dot{q}$ peut être écrite sous la forme :

$$C(q, \dot{q})\dot{q} = B(q)[q\dot{q}] + C(q)[\dot{q}^2] . \quad (I.77)$$

Où :

- $B(q)$: une matrice de dimension $n \times n(n - 1)$ des coefficients de Coriolis ses éléments sont égaux à :

$$b_{ijk} = \frac{\partial m_{kj}}{\partial q_i} + \frac{\partial m_{ki}}{\partial q_j} - \frac{\partial m_{ij}}{\partial q_k} . \quad (I.78)$$

- $C(q)$: est une matrice de dimension $(n \times n)$ des coefficients centrifuges et $[\dot{q}^2]$ est un vecteur de dimension $(n \times n)$, les éléments de $C(q)$ sont données par :

$$1 \quad C(q) = \frac{1}{2} \begin{pmatrix} \frac{\partial m_{ij}}{\partial q_k} - \frac{\partial m_{kj}}{\partial q_i} - \frac{\partial m_{ki}}{\partial q_j} \end{pmatrix} . \quad (I.79)$$

- La norme de la matrice $C(q, \dot{q})$ vérifiée la relation suivante :

$$\|C(q, \dot{q})\| \leq v_0 \|\dot{q}\|. \quad (\text{I.80})$$

Où, $v_0 > 0$ est indépendant de q .

- La norme du vecteur de gravité est bornée supérieurement ce que veut dire :

$$\|G(q)\| \leq g_b(q). \quad (\text{I.81})$$

Où g_b est une fonction scalaire.

Conclusion

Ce chapitre permet la découverte de quelques notions de base du domaine de robotique, notamment il présente les définitions générales de base et des notions permettant de comprendre la constitution et le fonctionnement d'un robot manipulateur rigide. Une présentation des différents types de modélisation des robots (la modélisation géométrique, la modélisation cinématique, la modélisation dynamique) est consacrée dans ce chapitre.

CHAPITRE II

LA COMMANDE PAR MODE GLISSANT

II.1 Introduction

Les lois de commande classique du type PID sont très efficaces dans le cas des systèmes linéaires à paramètres constants. Pour des systèmes non linéaires ou ayant des paramètres non constants, ces lois de commande peuvent être insuffisantes car elles sont non robustes surtout lorsque les exigences sur la précision et autres caractéristiques dynamiques du système sont strictes. On doit faire appel à des lois de commande insensibles aux variations des paramètres, aux perturbations et aux non linéarités. Les lois de commande dite à structure variable constituent une solution à ces problèmes.

La commande à structure variable (CSV) est, par sa nature, une commande non linéaire, elle est basée sur la commutation de fonctions des variables d'état, utilisées pour créer une variété ou hyper surface, le système de glissement, dont le but est de forcer la dynamique du système à correspondre avec celle définie par l'équation de l'hyper surface. Quand l'état est maintenu sur cette hyper surface, le système se trouve en régime glissant. Sa dynamique est alors insensible à perturbations extérieures et paramétrique tant que les conditions du régime glissant sont assurées.

Dans la pratique, l'utilisation de cette technique de commande a été long temps limitée par les oscillations liées aux commutations de la commande et qui peuvent se manifester sur les grandeurs asservies. Depuis, de nombreuses solutions ont été proposées permettant de réduire ces oscillations : augmentation de la fréquence de commutation, commande continue dans une bande autour de la variété de glissement ou décomposition de la commande en une composante continue de basse fréquence et une commande discontinue de haute fréquence et la méthode de la couche limite où la composante discontinue de la commande est remplacée au voisinage de l'hyper surface de glissement par une fonction continue.

Malheureusement, cette technique de commande conçue ne peut pas assurer une convergence en temps fini. Dans le but d'éviter ce problème c'est à dire obtenir une convergence en temps

fini il est extrêmement approprié d'utiliser une surface de glissement non linéaire, la commande est alors qualifiée de mode glissant terminal.

Dans ce chapitre on commence brièvement par une introduction aux systèmes de réglage à structure variable et la conception de la commande par mode glissement. Ensuite, on présente le problème générale de cette commande (phénomène de Chattering) et des quelque solutions existent. A la fin du chapitre, un résumé sur la conception de la commande mode glissant terminal et mode glissant terminal rapide est donné, ce dernier sera utile dans les prochains chapitres.

II.2 Introduction aux systèmes de réglage à structure variable

II.2.1 Principe de la commande par mode glissant

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

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

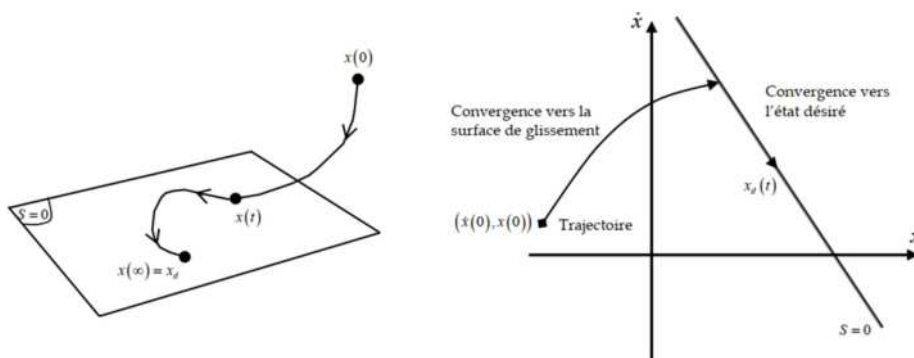


Figure II.1 : Convergence du système glissant.

II .2.2 Différentes structures du contrôle par mode de glissement

Un système à structure variable (VSS) est un système dont la structure change pendant son fonctionnement. Il est caractérisé par le choix d'une structure et d'une logique de commutation. Ce choix permet au système de commuter d'une structure à une autre à tout instant. De plus, un tel système peut avoir de nouvelles propriétés qui n'existent pas dans chaque structure.

➤ Structure par commutation au niveau de l'organe de commande

Dans ce schéma le changement de la structure se fait par commutation au niveau de l'organe de commande, comme bien illustre par la figure ci-dessous (Figure II.2) :

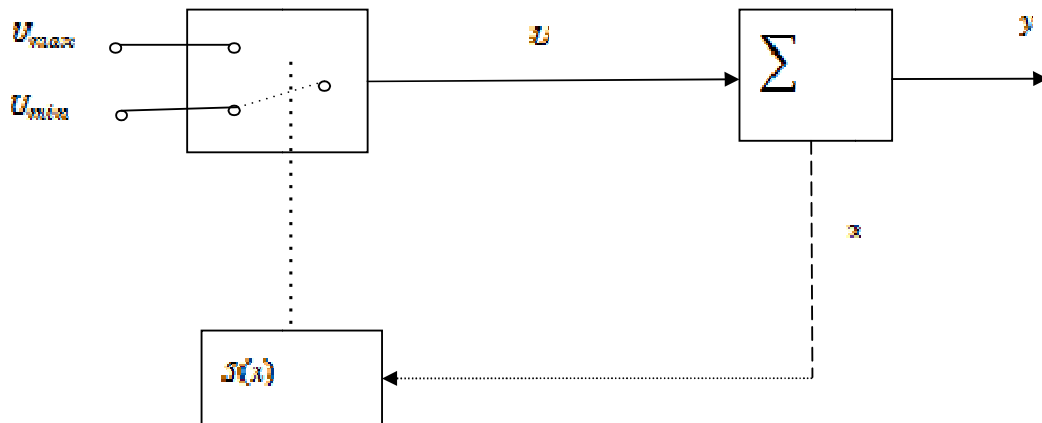


Figure II.2 : Configuration de la structure par commutation au niveau de l'organe de commande.

Dans cette structure, l'origine de commande entre deux valeurs constantes U_{Nas} et U_{Nin} , selon le signe de la fonction $S(x)$, alors la logique de commutation peut être définie comme suit [20] :

$$f(x) = \begin{cases} U = U_{Nas} & \text{pour } S(x) > 0 \\ 0 & \text{pour } S(x) = 0 \\ U = U_{Nin} & \text{pour } S(x) < 0 \end{cases} \quad (\text{II.1})$$

de commutation plus performante. Elle a été utilisée dans une grande majorité d'application aux variateurs de vitesse et aussi pour la commande des moteurs pas-à-pas .

➤ **Structure par commutation au niveau d'une contre réaction d'état**

Dans cette schéma le changement de la structure se fait lors de la commutation de la commande U. L'organe de commande reçoit dans ce cas une tension de commande U_{cn} qui commute rapidement entre deux valeurs variables, U_{cn1} et U_{cn2} ce qui peut provoquer de fortes sollicitations de l'organe de commande, donc, une réalisation pratique impossible. le schéma d'une telle configuration est représenté par la figure (II.3) [20].

Le système fonctionne alors en mode glissement et le comportement dynamique du système est déterminé selon la condition suivante :

$$f(x)=\begin{cases} U = U_{cn1} = -K_1x & \text{eour } \square s(x) > 0 \\ U = U_{cn2} = -K_2x & \text{eour } \square s(x) < 0 \\ 0 & \end{cases} \quad (II.2)$$

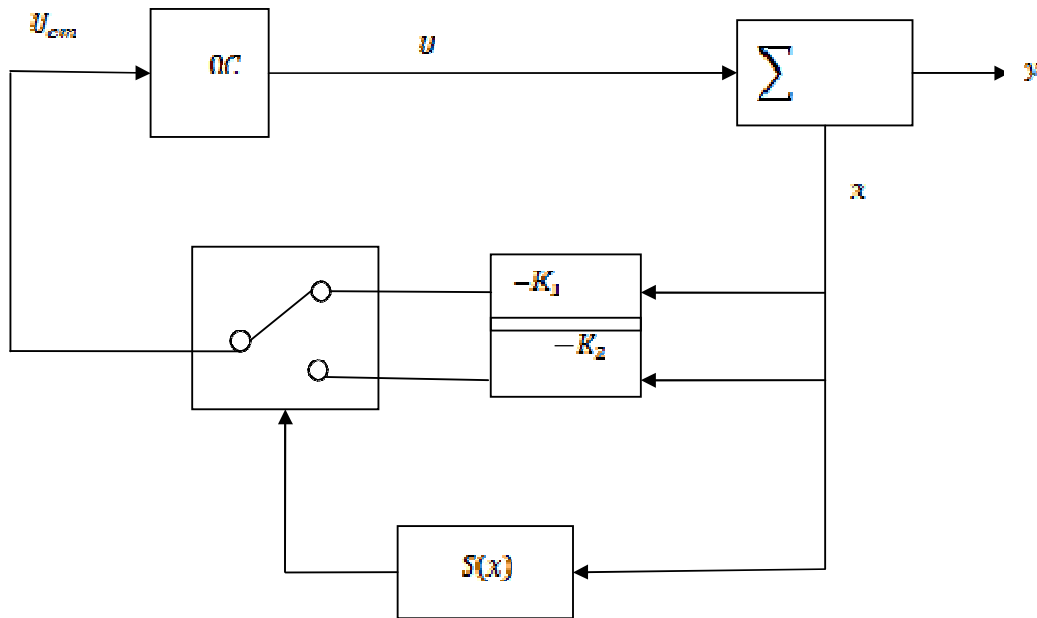


Figure II.2 : Configuration de la structure par commutation d'une contre réaction d'état variable.

II .3. Conception de la commande par mode glissement

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

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

II.3.1 Choix des surfaces de glissement

Le choix des surfaces de glissement concerne leur nombre ainsi que leur formes en fonction de l'application et de l'objectif visé .pour un système défini par l'équation d'état suivante :

$$\dot{x}(t) = A(x, t) + B(x, t)u(t) \quad (\text{II.3})$$

Avec

$X = (x_1, x_2, x_3, \dots, x_n)^T$ est le vecteur d'état de système, $u = (u_1, u_2, u_3, \dots, u_n)^T$ est le vecteur de commande. et A, B sont des champs du vecteur suffisamment différentiables définis sur X

Pour le nombre de la surface, il faut choisir "N" surfaces de glissement pour un vecteur de commande "v" de dimension "N".

En ce qui concerne la forme de la surface, il y a deux possibilités :

- ✓ Dans le plan de phase.
- ✓ Dans l'espace d'état.

Dans notre mémoire, nous nous intéressons à l'analyse par la méthode de traitement dans le plan de phase où la fonction de commutation est une fonction scalaire, telle que la variable à réguler glisse sur cette surface pour atteindre l'origine du plan de phase [13], ainsi, la surface de glissement S(x) représente le comportement dynamique désiré du système.

La forme d'équation générale pour déterminer la surface de glissement, proposée par «

J.J.Slotinie » et assurant la convergence d'une variable x vers sa valeur désirée, est donnée comme suit :

$$S(x) = \left(\frac{6}{6t} + h_s\right)^{r-1} \cdot e(x) \quad (\text{II.4})$$

Avec :

$e(x)$: Écart de la variable à réguler.

h_s : Constante positive choisie par le compteur.

r : Degré relatif égale au nombre de dérivation de la sortie pour faire apparaitre le terme de commande.

Cette dernière équation est différentielle linéaire dont l'unique solution est $e(x)=0$ pour un choix convenable du gain h_s . Ceci revient à un problème de poursuite de trajectoire qui est équivalent à celui du problème qui consiste à amener les points représentatif du système à rester sur la surface de glissement définie par l'équation $s(x)=0$. Ceci est équivalent à une linéarisation exacte de l'écart tout en respectant une condition de convergence.

Notons que la linéarisation exacte de l'écart consiste à forcer l'écart à avoir une dynamique similaire à celle d'un système linéaire autonome comme illustré dans la (figure II, 3)

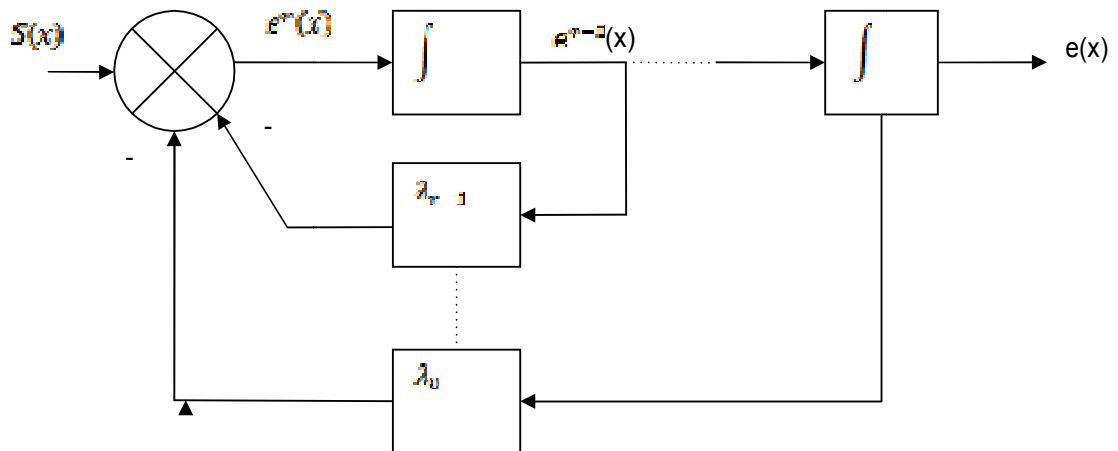


Figure. II.3 : Linéarisation exacte de l'écart

II.3.2 Condition de convergence

La condition de glissement est effectuée afin de contraindre les points représentatifs de

la trajectoire à rester sur la surface de glissement. Nous retenons dans la littérature deux conditions, celles-ci correspondent au mode de convergence de l'état du système.

II.3.2.1 Fonction directes de commutation

C'est la première condition de convergence. Cette condition est proposée par EMILYANOV et UTKIN [12] et elle donnée par :

$$\begin{cases} \dot{s}(x) > 0 & \text{lorsque } s(x) < 0 \\ \dot{s}(x) < 0 & \text{lorsque } s(x) > 0 \end{cases} \quad (\text{II.5})$$

Ce qui est équivalent à :

$$\dot{S}(x) \cdot S(x) < 0 \quad (\text{II.6})$$

Dans cette condition, il faut introduire pour $S(x)$ et sa dérivée $\dot{s}(x)$ les valeurs juste à gauche ou à droite de la droite de commutation. La condition (II.6) exige que la dérivée de $S(x)$ doit aussi changer de signe lorsqu'on passe d'un côté à l'autre de la droite, ceci peut exprimer par l'expression suivante:

$$\text{signe}(S(x)) = -\text{Signe}(\dot{S}(x)) \quad (\text{II.7})$$

II.3.2.2 Fonctions de Lypunov

Il s'agit de formuler une fonction scalaire positive $V(x) > 0$ pour les variable d'état du système, et de concevoir une loi de commutation qui fera décroître cette fonction. Elle est utilisée pour l'étude de la robustesse, et garantir la stabilité des systèmes non linéaires.

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

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

Pour que cette fonction de type Lyapounov $V(x)$ décroisse, il suffit d'assurer que sa dérivée soit négative ceci est vérifié par la condition (II.6).

L'équation (II.8) explique que le carré de la distance vers la surface mesuré par $s(x)^2$, diminue tout le temps, contraignant la trajectoire de système à se diriger vers la surface des deux côtés (figure II.4).

La condition de convergence (II.6) suppose un régime glissant idéal .Dans le cas d'un régime glissant pratique, la condition (I.4) qui ne garantie qu'une convergence asymptotique vers la surface de glissement est remplacée par une condition plus forte donnée par :

$$\frac{1}{2} \frac{d}{dt} S(x)^2 \leq -\eta |S(x)| \quad (II.9)$$

est appelée condition de η -attractivité avec η est une constante positive qui définira le temps de convergence vers la surface de glissement. La satisfaction de la condition (II.9), garantie que si la condition $x(t=0)$ n'est pas exactement vérifiée, ce qui revient à supposer que $S(t \neq 0)$ soit en dehors de $x'(t=0)$, la surface $S(x)$ sera atteinte en un temps T_r fini plus petit que :

$$T_r = \frac{S(t=0)}{\eta} \quad (II.10)$$

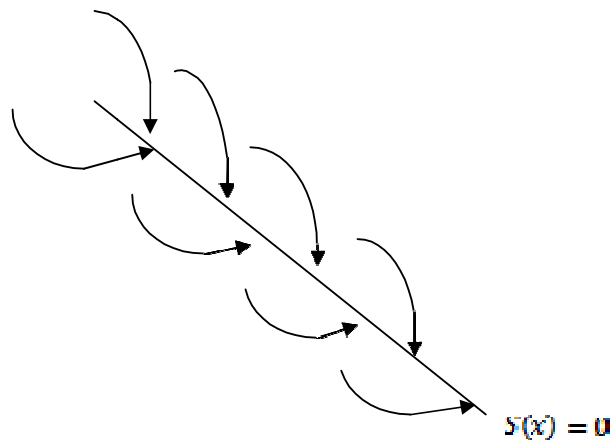


Figure II.4 : trajectoire de l'état vis-à-vis de la surface de glissement.

II.3.3 Calcul de la loi de commande

Une fois la surface de glissement choisie, ainsi que le critère de convergence, il reste à déterminer la commande nécessaire pour attirer la variable à réguler vers la surface, ensuite vers son point d'équilibre (origine du plan de phase) en maintenant la condition d'existence du mode glissant.

II.3.3.1 Définitions des grandeurs de la commande

Lorsqu'il y a un régime glissant, la dynamique du système est indépendante de la loi de commande qui n'a pour seul but que de maintenir les conditions de glissement, c'est à dire l'attractivité de la surface de commutation. c'est pourquoi la surface de glissement a pu être déterminée indépendamment de la commande, sur la base du système et les performances désirées comme nous l'avons mentionné, l'obtention d'un régime glissant supposerait une commande discontinue.

De ce fait, si cette composante est indispensable, elle n'empêche nullement, au contraire, qu'une partie continue lui soit adjointe pour diminuer l'amplitude de la discontinuité. la partie continue peut en effet amener à réduire autant qu'on veut la partie discontinue ayant essentiellement pour but de maintenir les conditions d'attractivité. de ce fait, la structure du contrôleur comprend deux parties, une partie concernant la linéarisation exacte et l'autre stabilisante :

$$U = U_{eq} + U_n \quad (II.11)$$

U_{eq} : représente la commande équivalente proposée par FILIPOV et UTKIN, qui est considérée comme la plus simple et la plus directe. Elle est calculée en reconnaissant que le comportement du système durant le mode de glissement est décrit par :

$$S'(x) = 0 \quad (II.12)$$

Notons que la commande équivalente peut être considérée comme étant un retour d'état particulier, jouant le rôle du signal de valeur efficace appliqué sur le système à commander

Ainsi, comme le montre dans la (Figure II.5), la commande équivalente U_{eq} peut être interprétée comme la valeur moyenne de la commande U lors de la commutation à fréquence élevée entre U_{Nas} et U_{Nin} .

U_n : représente la commande discontinue, Il est constitué de la fonction signe "sign" de la surface de glissement S , multipliée par une constante de glissement K_{gsic} . Cette commande est déterminée pour garantir l'attractivité de la variable à contrôler vers la surface et satisfaire la condition (II.6).

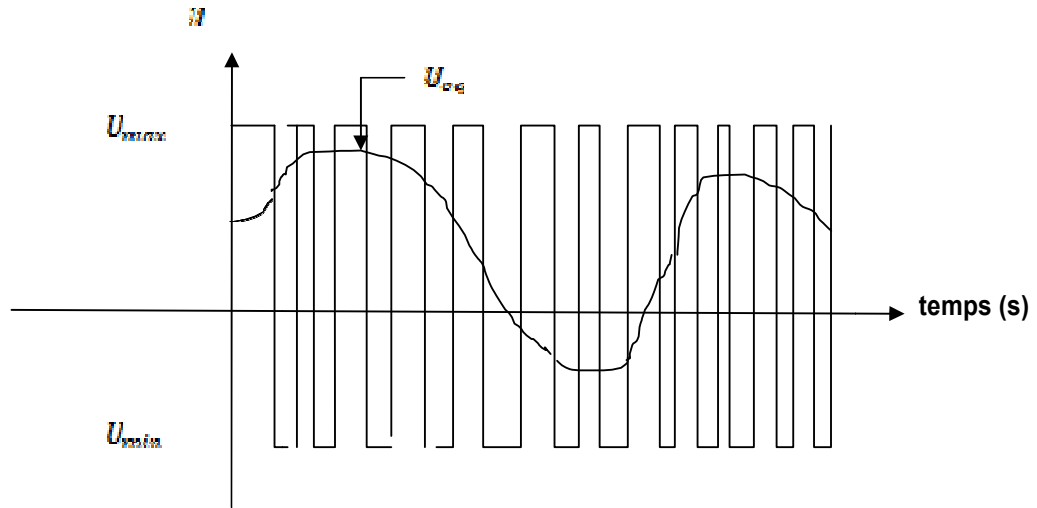


Figure II.5 : Signal de commande équivalente U_{eq}

II.3.3.2 Applications de la commande

Nous considérons le système non linéaire décrit par l'équation (II.3). Nous nous occupons d'abord du calcul de la commande équivalente U_{eq} ensuite de la commande attractive U_n .

La dynamique de système $S'(x)$ est donnée par :

$$S'(x) = \frac{dS}{dX} \frac{dX}{dt} = [A(x, t) + B(x, t) \cdot U_{eq}(t)] + \frac{dS}{dt} [B(x, t) \cdot U_n] \quad (II.13)$$

Durant le mode de glissement dans le régime permanent la dérivée de la surface est nulle, alors on peut trouver, dans ce régime, à partir de (II.13) que :

$$[A(x, t) + B(x, t) \cdot U_{eq}(t)] + \frac{dS}{dt} [B(x, t) \cdot U_n] = 0 \quad (II.14)$$

Ce qui implique

$$U_{eq} = - \left[\frac{dS}{dt} B(x, t) \right]^{-1} \left[\frac{dS}{dt} A(x, t) \right] \quad (II.15)$$

$$U_n = 0$$

Durant le mode de convergence, en remplaçant la commande U_{eq} par sa valeur dans l'équation (II.14), nous obtenons les nouvelles expressions de la dynamique de l'erreur comme suit :

$$\dot{S}(x) = \frac{6c}{6s} B(x, t) \cdot U_n \quad (II.16)$$

Le problème revient à trouver la commande attractive U_n tel que :

$$\dot{S}(x) \cdot S(x) = S(x) \frac{6s}{6x} B(x, t) U_n < 0 \quad (II.17)$$

Le choix le plus simple du terme de commande U_n , qui permet de satisfaire la condition (II.16) est (Figure II.6) :

$$U_n = k \cdot \text{Signe}(S(x)) \quad (II.18)$$

En remplaçant l'expression de U_n dans (II.18), on obtient :

$$\dot{S}(x) \cdot S(x) = \frac{6s}{6s} B(x, t) k |S(x)| < 0 \quad (II.19)$$

Le facteur $\frac{6c}{6s} B(x, t)$ est négatif pour la classe des systèmes que nous considérons. Le gain k est choisi positif pour satisfaire la condition (II.19). Notons que si k est choisi très petit, le temps de réponse sera très long, et s'il est choisi très grand nous aurons de fortes oscillations de l'organe de commande qui peuvent exciter les dynamiques négligées.

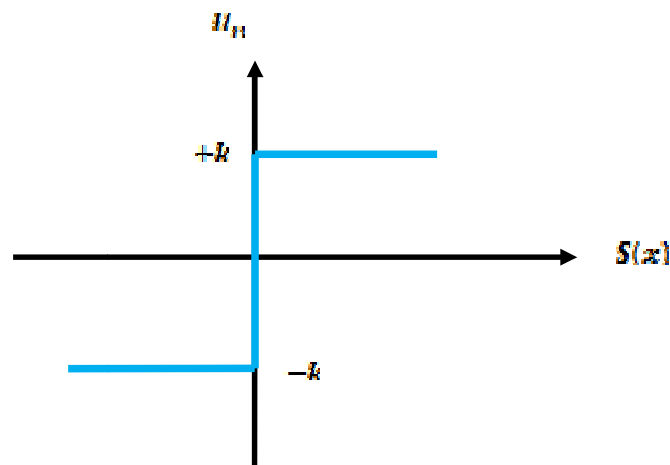


Figure II.6 : Définition de la commande signe

II.3.4 Élimination du phénomène de Chattering

En mode de glissement idéal, les trajectoires du système sont des fonctions continues ne dépendant que de la surface de commutation choisie. Ces trajectoires correspondent au système commandé par un retour statique appelé contrôle équivalent. Cependant, du fait des imperfections technologiques (retard, hystérésis) des actionneurs et des capteurs, le contrôle discontinu ne peut pas osciller idéalement à une fréquence infinie. Alliée aux retards négligés lors de la modélisation du processus physique, la commande discontinue engendre alors une dynamique parasite appelée communément “chattering” en anglais, ou phénomène de réticence en français. Celle-ci se caractérise par des oscillations persistantes et à hautes fréquences de la commande

Dans le but de réduire ou éliminer ce phénomène, plusieurs solutions ont été apportées. Dans cette section, nous proposons de décrire les techniques de limitation de la condition de glissement.

II.3.4.1 Fonction SAT

Les amplitudes des ondulations peuvent être atténuées en remplaçant la fonction signe par une fonction de saturation SAT caractérisée par un ou deux seuils comme illustré dans la figure (II.7) .

- **1^{ère} cas (pour un seuil)**

$$\begin{cases} \text{sat}(S) = +1 & \text{si } S > \varepsilon \\ \text{sat}(S) = \frac{c}{\varepsilon} & \text{si } |S| < \varepsilon \\ \text{sat}(S) = -1 & \text{si } S < -\varepsilon \end{cases} \quad (\text{II.20})$$

Avec ε : est un constant positif très petit

- **2^{ième} cas (pour deux seuils)**

$$\begin{cases} \text{fsat}(S) = \text{signe}(S) & \text{si } |S| > \varepsilon_2 \\ \text{fsat}(S) = \frac{S - \varepsilon_1}{\varepsilon_2 - \varepsilon_1} & \text{si } \varepsilon_1 < |S| < \varepsilon_2 \\ \text{Lsat}(S) = 0 & \text{si } |S| < -\varepsilon_1 \end{cases} \quad (\text{II.21})$$

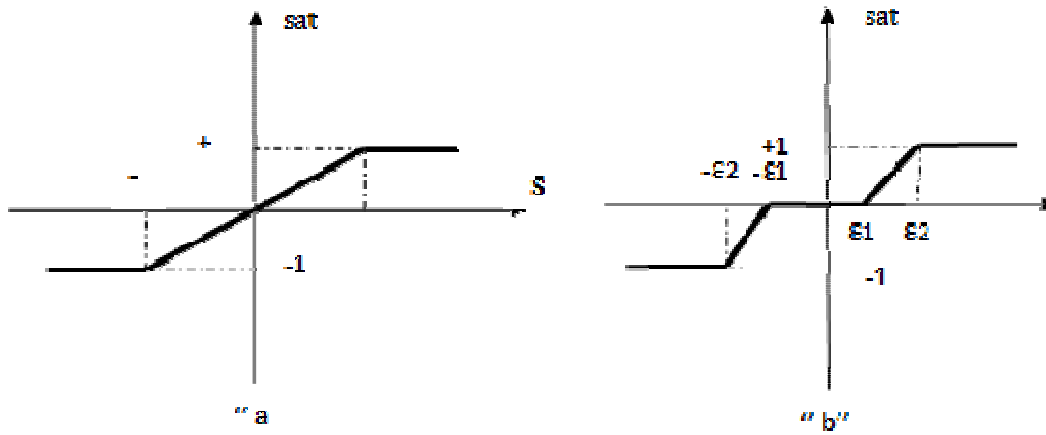


Figure II.7. Représentation de la fonction SAT a : un seuil, b : deux seuils

II.3.4.2 Fonction SMOOTH

On peut aussi remplacer la fonction signe par une fonction continue définie par la forme suivante (la figure (II.7)) :

$$S_{Nout}(s) = \frac{s}{|s| + \epsilon} \quad (II.22)$$

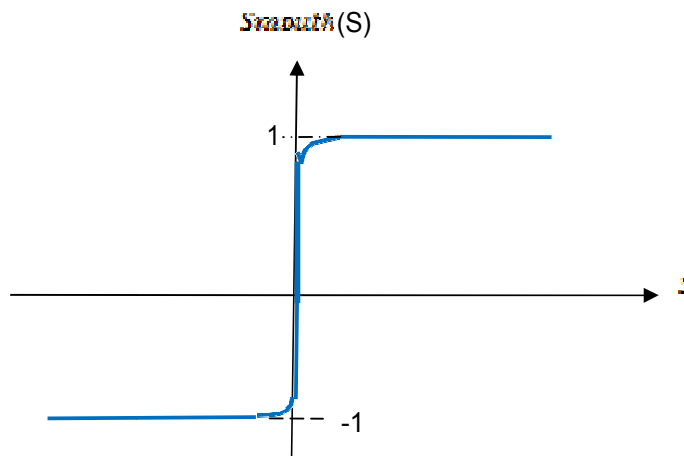


Figure II.8: Représentation de la fonction SMOOTH.

Dans la section suivante, on propose de présente la commande par mode glissant terminal, qui a un rôle très important pour l'amélioration de la performance de la commande

par mode glissant.

II.4. Concept générale de la commande par mode glissant terminal

Un nouvel algorithme de contrôle de type mode de glissant, nommé mode glissant terminal (TSM), a été récemment étudié pour améliorer davantage les performances de commande [20], [21]. En comparaison avec l'hyperplan linéaire à base de mode glissant, TSM offre des propriétés supérieures telles que : éviter d'une façon complète le Chattering, rapidité et convergence en temps fini. Ce contrôleur est particulièrement utile dans le traitement de hautes précisions, car il accélère, dans le cas des systèmes non linéaires incertains, d'ordres deux ou supérieurs, le taux de convergence près du point d'équilibre choisi [20], [22].

II.4.1. Convergence en temps fini

Cette nouvelle algorithme peut converger l'état du système vers zéro dans un temps fini. Le traitement avec une poursuite en temps fini n'a seulement pas pour but, d'améliorer les performances du système, mais il rend aussi les propriétés de stabilité plus fortes. Il n'a donc pas seulement un intérêt théorique premier, mais il fait également appel aux implications pratiques : il assure par exemple que la tâche est faite exactement comme désirée, c'est à dire d'une manière robuste.

II.4.2 Conception de la commande par mode glissant terminal

L'idée principale de la conception de la commande par mode glissant terminal est de choisir une surface de glissement de type non linéaire et de détermination d'une loi permettant de conduire l'état du système à rester sur cette surface.

II.4.2.1 Description du système

Considérons une classe des systèmes non linéaires SISO incertains donnée par son modèle d'état suivant [27] :

$$\begin{cases} \dot{x}_1 = x_2 \\ \dot{x}_2 = f(x, t) + g(x, t)u(t) + d(x, t) \\ y = x_1 \end{cases} \quad (\text{II.23})$$

Où $x = [x_1, x_2]^T$ est le vecteur d'état du système, $f(x, t)$ et $g(x, t)$ sont des fonctions continues non linéaires et $d(x, t)$ est une perturbation externe.

Objectif : développer une loi de commande mode glissant terminal u pour forcer la sortie y à suivre une certaine trajectoire désirée.

II. 4.2.2 Conception de la surface de glissement

Maintenant, on définit une surface de glissement comme suite [23] :

$$S = x_2 + p x_1^{q/p} \quad (\text{II.24})$$

Où $p > 0$, e et q sont des nombres entiers impaires positif et $e > q$.

Avec sa dérivée est :

$$\dot{S} = \dot{x}_2 + p \frac{q}{p-1} x_1^{(q-p)/p} \dot{x}_1 \quad (\text{II.25})$$

II.4.3 Calcul de la loi de commande

La condition suffisante pour l'existence de TSM est [23] :

$$\frac{1}{2} \frac{d}{dt} S^2 < -\delta |S| \quad (\text{II.26})$$

Où $\delta > 0$.

ce qui implique:

$$S\dot{S} < -\eta |S| \quad (\text{II.27})$$

En remplaçant (II.3) dans (II.5), on obtient :

$$S(x_2 + p \frac{q}{p-1} x_1^{(q-p)/p} \dot{x}_1) < -\eta |S| \quad (\text{II.28})$$

Maintenant en utilisant (II.1) dans cette inégalité, on trouve

$$S(f(x, t) + g(x, t)u(t) + d(x, t) + p \frac{q}{p-1} x_1^{(q-p)/p} x_2) < -\eta |S| \quad (\text{II.29})$$

Si $S > 0$ alors :

$$u < \frac{1}{g(s,t)} (-\eta - f(x, t) - d(x, t) - p \frac{q}{p-1} x_1^{(q-p)/p} x_2) \quad (\text{II.30})$$

Si $S < 0$ alors :

$$u > \frac{1}{g(s,t)} (\eta - f(x, t) - d(x, t) - p \frac{q}{p-1} x_1^{(q-p)/p} x_2) \quad (\text{II.31})$$

Pour vérifier les deux inégalités (II.28) et (II.29), il suffit de prendre u dans l'expression

suiivante:

$$U = - \frac{1}{g(s,t)} (f(x, t) - d(x, t) - p \frac{q}{p-1} x_2^{(q-p)/p} + (5 + k) \text{sign}(S)) \quad (\text{II.32})$$

Où k est une constante positive.

A partir de cette dernière équation on peut trouve que si $S(0) \neq 0$, l'état du système atteint le mode glissant $S = 0$ dans un temps fini T_r vérifiant :

$$T_r \leq \frac{|c(0)|}{\gamma} \quad (\text{II.33})$$

Lorsque le régime glissant est attient, la dynamique (II.33) devient alors

$$\dot{x}_1 = -p x_1^{(q)/p} \quad (\text{II.34})$$

Soit donné un état initial $x_1(0) \neq 0$, la dynamique (II.2) atteindra l'état $x_1=0$ dans un temps fini [24].

A partir de cette dernière équation on peut déterminer l'expression de la période t_s prise à partir de l'état initial $x_1(0)$ à 0 comme suit :

$$t_c = -p - 1 \int_{s_1(0)}^0 \frac{ds_1}{s_1^q} = \frac{p}{p(p-q)} |x_1(0)|^{(q-p)/p} \quad (\text{II.35})$$

On peut également démontrer que le point d'équilibre 0 est un attracteur [24], c'est-à-dire .quand l'état x_1 atteint zéro pour toujours. Ceci peut être prouvé en prenant une fonction de lyapunov $V = \frac{1}{2} x_1^2$ Alors sa dérivée peut être exprimée comme suit :

$$\dot{V} = x_1 \dot{x}_1 = -p x_1 p x_1^{(q)/p} = -p x_1^{(p+q)/p} \quad (\text{II.36})$$

Puisque $(e + q)$ est paire, alors, \dot{V} est définie négative, ce qui rend ainsi $x_1 = 0$ stable (mais pas nécessairement asymptotiquement stable)

Le temps t_s exigé pour atteindre l'équilibre 0 et qui est déterminé par (II.13) dépend des paramètres pas paramètres p, q, β , et de la valeur initiale $x_1(0)$. De même que $x_1(0)$ est soit fixé ou appartenant à une région limitée connue, on peut choisir β de telle sorte que t_s soit très petit [24].

II.5 Commande par mode glissant terminal rapide des systèmes SIFO (FTSM)

La surface de glissement de la commande FTSM est formulée comme suit [24], [25] :

$$S = \dot{x}_1 + \alpha x_1 + \beta x_1^{q/d} \quad (II.37)$$

Où $x_1 \in \mathbb{R}^1$, est une variable scalaire $\alpha, \beta > 0$ et e, q sont des nombres entiers impaire positifs vérifiant $e > q$. Il peut être déduit que le temps pour atteindre l'équilibre $x_1=0$ est déterminé par [24], [25] :

$$t_c = \frac{p}{\alpha(p-q)} \ln \frac{\alpha |s_1(0)|^{(p-q)/p+\beta}}{\beta} \quad (II.38)$$

La performance de convergence rapide en comparaison avec le mode de glissement linéaire classique peut être démontrée par l'exemple suivant en utilisant MATLAB considérant $\alpha=1, \beta=1$ et la condition initiale $x_1(0)=1$.

D'abord, on suppose que $e = 3$ et $q = 1$. A partir de (II.38), on peut facilement trouver que le temps d'attendre zéro est $t_c = 1.03972077083992$ sec.

Nous comparons maintenant ce qui précède à la situation où e et q sont mis à 1. Pour $e = 3$ et $q = 1$ on trouve $x_1(t_c) = 0.12500519281775$ sec.

Il est évident que le taux de convergence du mode glissant terminal rapide est beaucoup mieux que son homologue linéaire (réglage $= q$). La raison évidente est quand près de l'équilibre, la vitesse de convergence du mode de glissement linéaire ralentit exponentiellement vers le bas tandis que le taux de convergence de la mode glissant rapide accélère de façon exponentielle [25].

La structure hiérarchique basée sur le concept de mode de glissement rapide pour les systèmes d'ordre supérieur peut être utilisée [24], [25] :

$$S_1 = \dot{S}_0 + \alpha_0 S_0 + \beta_0 S_0^{q_0/p_0} \quad (II.39)$$

$$S_2 = \dot{S}_1 + \alpha_1 S_1 + \beta_1 S_1^{q_1/p_1} \quad (II.40).$$

$$S_{n-1} = \dot{S}_{n-2} + \alpha_{n-2} S_{n-2} + \beta_{n-2} S_{n-2}^{q_{n-2}/p_{n-2}} \quad (II.41)$$

Où $\alpha_i, p_i > 0$ avec $i = 1, 2, \dots, n-2$, et e et q ($p > q$) sont des nombres entiers impairs positifs.

C'est évident que, si S_{n-1} atteint zéro, $S_{n-2}, S_{n-3}, \dots, S_0$ atteindront zéro séquentiellement.

La commande par mode glissant u peut être conçue de sorte que

$$\dot{S}_{n-1} \leq -k |S_{n-1}| \quad k > 0 \quad (\text{II.42})$$

Ce qui représente un état suffisant. Par conséquent $S_{n-1} = 0$ peut être atteint dans un temps fini, ainsi que $x_1 = 0$ [24].

Prenant $S_0 = x_1$, S_i peut être exprimé comme suit [25] :

$$S_i = S_{i-1} + \alpha_{i-1} S_{i-1} + p_{i-1} S_{i-1}^{q_{i-1}/p_{i-1}} \quad (\text{II.43})$$

sa dérivée \dot{S}_i est donnée par :

$$\dot{S}_i^{(S)} = \dot{S}_{i-1}^{(S)} + \alpha_{i-1} \dot{S}_{i-1}^{(S)} + p_{i-1} \frac{d^{(1)}}{dt^{(1)}} S_{i-1}^{q_{i-1}/p_{i-1}} \quad (\text{II.44})$$

Prenant la dérivée d'ordre 1 de S_{n-1} , on obtient l'équation suivante :

$$\dot{S}_{n-1} = \dot{S}_{n-2} + \alpha_{n-2} \dot{S}_{n-2} + p_{n-2} \frac{d}{dt} S_{n-2}^{q_{n-2}/p_{n-2}} \quad (\text{II.45})$$

Ensuite, il peut facilement être induit étape par étape jusqu'à l'équation suivante est arrivée [25] :

$$\dot{S}_{n-1} = \dot{S}_0^{(n)} + \sum_{k=0}^{n-2} \alpha_k \dot{S}_k^{(n-k-1)} + \sum_{k=0}^{n-2} p_k \frac{d^{n-k-1}}{dt^{n-k-1}} S_k^{(q_k/p_k)} \quad (\text{II.46})$$

A partir de (II.45), prenant S_{n-1} comme suivant :

$$\dot{S}_{n-1} = -\gamma S_{n-1} + y S_{n-1}^{q/p} \quad (\text{II.47})$$

Où $\gamma, y > 0$, et e, q sont des nombres entiers impairs positifs vérifiant $e > q$.

Alors pour le système non linéaire (II.1), la loi de commande de type FTSM est définie comme suit [30] :

$$U = -\frac{1}{g(s)} \left(f(x) + \sum_{k=0}^{n-2} \alpha_k \dot{S}_k^{(n-k-1)} + \sum_{k=0}^{n-2} p_k \frac{d^{n-k-1}}{dt^{n-k-1}} S_k^{(q_k/p_k)} + \gamma S_{n-1} + y S_{n-1}^{q/p} \right) \quad (\text{II.48})$$

Avec $S_0 = x_1$, alors les états du système atteindra la surface de glissement $S_{n-1} = 0$ selon

l'attracteur terminal $S_{n-1} = -\delta S_{n-1} - \gamma S_{n-1}^{q/p}$ dans un temps fini, $t_{c_{n-1}}$, où

$$t_{c_{n-1}} = \frac{p}{\delta(p-q)} \ln \frac{\delta |s_1(0)|^{(p-q)/p + \gamma}}{\gamma} \quad (\text{II.49})$$

Avec $\delta, > 0$, et e, q sont des nombres entiers impairs positifs vérifiant $e > q$. Le système suit la structure hiérarchique pour converger vers l'équilibre du système en temps fini.

II.5.1 Étude de la stabilité

Considérons la fonction de type Lypunov suivante [19]:

$$V = \frac{1}{2} S_{n-1}^2 \quad (\text{II.50})$$

La dérivée de V par rapport au temps est donnée par :

$$\dot{V} = S_{n-1} \dot{S}_{n-1} \quad (\text{II.51})$$

Substituant (II.50) dans (II.51), on obtient

$$\dot{V} = -\delta S_{n-1}^2 - \gamma S_{n-1}^{(q+p)/p} \quad (\text{II.52})$$

Puisque p, q sont des nombres entiers impairs positifs, $(p+q)$ est pair, alors, \dot{V} est définie négative. Donc le système est stable [19].

Remarque :

Le mécanisme peut facilement être étendue à la conception du problème de poursuite de sortie si γ est remplacé par $e = y_N - y$, où y_N est le signal de sortie désiré.

II.6 Conclusion

La commande par mode glissant est une commande non linéaire. La caractéristique principale de cette commande est que sa loi de commande change d'une manière discontinue selon des conditions déterminées. Elle contient un terme continu (commande équivalente) pour pré-positionner l'état futur de système et un terme discontinu de fréquences élevée (commande discontinue) pour assurer l'attractivité de la surface et la compensation des perturbations externes ou des erreurs de modélisation. Malgré que ce type de commande par mode glissant classique est caractérisé par sa robustesse et sa facilité de conception mais elle est difficile à mettre en application en raison du phénomène de broutement (chattering) et le

problème de convergence en temps infini. Une solution à ces problèmes consiste à appliquer la commande par mode glissant terminal cette dernière permet d'atteindre une convergence de temps fini et de ce fait réduit le phénomène de broutement.

Dans ce chapitre, Après avoir présenté les idées clés de la commande par modes glissant: systèmes de réglage à structure variable, la surface de glissement, les conditions d'existence et de convergence, les grandeurs de la commande, le problème de chattering, autre partie de ce chapitre a été consacré à une synthèse sur la commande par mode glissant terminal (TSM) et sur la commande par mode glissant terminal rapide (FTSM).

CHAPITRE III

COMMANDE PAR MODE GLISSANT D'UN BRAS MANIPULATEUR

III.1. Introduction

La commande par modes glissants des systèmes non linéaire a été largement étudiée et développée depuis son introduction. Celle-ci appartient à une classe plus large appelée commande à structure variable. L'objectif de la commande par mode glissant est à l'aide d'une commande discontinue, de contraindre le système à évoluer au bout d'un temps fini et de maintenir sur une surface dite surface de glissement, où le comportement résultant correspond aux dynamiques souhaitées.

La technique des modes glissants consiste à ramener la trajectoire d'un état d'un système vers la surface de glissement et de la faire commuter à l'aide d'une commutation appropriée autour de celle-ci jusqu'au point d'équilibre, d'où le phénomène de glissement. L'avantage de cette commande est qu'elle est robuste aux perturbations et aux incertitudes de modèle.

Dans ce chapitre, nous présenterons les concepts généraux des différentes commandes pour un bras manipulateurs, donc on s'est intéressé par la commande PID et les commandes basées sur le mode de glissement.

Rappelle sur les équations du mouvement

Les équations du modèle dynamique du robot dont la forme générale, pour un robot à n degrés de liberté, est la suivante :

$$\Gamma = A(q)\ddot{\phi} + C(q, \dot{\phi})\dot{\phi} + Q(q) + \text{diag}(\dot{\phi})F_v + \text{diag}(\text{sign}(\dot{\phi}))F_s \quad (\text{III.1})$$

Ou, sous une forme plus compacte :

$$\Gamma = A(q)\ddot{\phi} + H(q, \dot{\phi}) \quad (\text{III.2})$$

Le modèle ayant une forme linéaire par rapport aux paramètres dynamiques :

$$\Gamma = \Phi(q, \dot{\phi}, \ddot{\phi})X \quad (\text{III.3})$$

Avec :

- Γ : est le vecteur (nx1) des couples/forces articulaires.
- $A(q)$: est la matrice (nxn) d'inertie du robot.
- $C(q, \dot{q})\dot{q}$: est le vecteur (nx1) représentant les forces centrifuges et les forces de Coriolis.
- $Q(q)$: est le vecteur des forces de gravité.
- F_v et F_s sont les vecteurs des paramètres de frottement respectivement visqueux et sec.
- X : représente le vecteur des paramètres dynamiques (inertiels et frottement). [2]

Commande de PID d'un bras manipulateur 2 degrés de liberté

(2DDL)

La synthèse d'un asservissement doit toujours répondre à certaines exigences. Ces dernières sont appelées cahier des charges. Un cahier des charges d'une boucle de régulation, impose en boucle fermée: Stabilité, Précision, Rapidité, Robustesse, Dépassement, Rejet de perturbations, ...

Les systèmes peuvent présenter des défauts, par exemples: une précision insuffisante, une stabilité faible (voir une instabilité), un temps de réaction trop lent, un dépassement trop important, etc. Il est souvent nécessaire d'intégrer dans le système asservi un correcteur.

L'objectif est alors d'améliorer un ou plusieurs de ces différents paramètres, sans bien évidemment le faire au détriment des autres.

On considère un système asservi défini par le schéma bloc suivant :

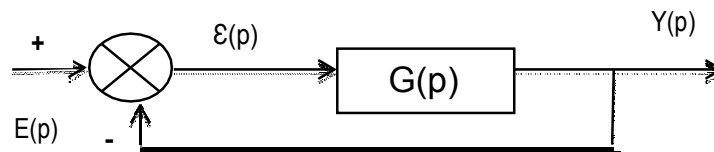


Figure III.1 : Schéma bloc d'un système asservi avec correcteur.

L'objectif de la correction (la commande) consiste à choisir la bonne fonction de transfert $C(p)$ du correcteur de manière à régler chaque performance sur sa valeur requise, sans perturber le fonctionnement du système.

En automatique, il existe trois actions correctives élémentaires qui permettent, individuellement, de corriger telle ou telle performance: l'action Proportionnelle (**P**), l'action Intégrale (**I**), l'action Dérivée (**D**). Ces actions sont relativement simples à réaliser mais,

généralement, elles dégradent d'autres performances. Quand le cahier des charges est peu exigeant, ces actions suffisent largement à satisfaire ce dernier.

Dans le cas contraire, il faut envisager de combiner ces différentes actions au sein d'un correcteur plus complexe de type PID.[26]

Pour élaborer la commande PID (classique), il faut considérer chaque articulation comme un mécanisme indépendant dont le modèle peut être linéarisé dans une zone de fonctionnement. La commande prend le caractère local. Avec sa simplicité et sa facilité d'implémentation, elle tient le monopole dans le domaine industriel.

- Effets du régulateur proportionnel-Intégral-Dérivé (PID):
 - Diminution du temps de montée.
 - Elimination de l'erreur statique.
 - Diminution du temps de stabilisation.
 - Diminution du dépassement.

La loi de commande (t) s'exprime en fonction de l'erreur comme suite :

$$\tau \ddot{q}(t) = k_p q(t) - k_v \dot{q}(t) + k_I \int_0^t q(t) \quad (\text{III.4})$$

Avec :

- K_p : le coefficient du Proportionnel.
- K_d : le coefficient de la Dérivée.
- K_I : le coefficient de l'Intégral.

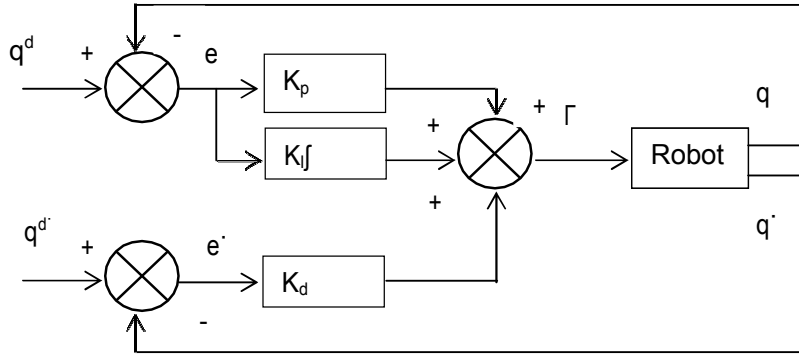
Remarque :

- En cas d'erreur de position il faut augmenter le gain proportionnel et le gain dérivé.
- Quand les gains proportionnel et dérivé sont optimisés, le gain intégral est ajusté en augmentant sa valeur tant que l'erreur de position statique est suffisamment réduite...

Commande de PID dans l'espace articulaire

Le modèle dynamique décrit un système de n équations différentielles du second ordre non linéaires et couplées, n'étant le nombre d'articulations. Pourtant, dans une commande classique, qui est celle de la plupart des robots industriels actuels, le mécanisme est considéré comme un système linéaire et chacune de ses articulations est asservie par une commande décentralisée de type PID à gains constants. Ses avantages sont la facilité d'implantation et le faible coût en calcul. En contrepartie, la réponse temporelle du robot variant selon sa

configuration, on constate des dépassements de consigne et une mauvaise précision de suivi dans les mouvements rapides. Dans beaucoup d'applications, ces inconvénients ne représentent pas un gros handicap. En pratique, une telle commande est réalisée selon le schéma de la figure(III.2). [27].



FigureIII.2 Schéma classique d'une commande PID dans l'espace articulaire.

La loi de commande est donnée par :

$$\Gamma = k_p(q^d - q) + k^d(q^{\bullet d} - q^{\bullet}) + k_I \int_{t_0}^t (q^d - q)d\tau . \quad (III.5)$$

Où :

- $q^{\bullet d}(t)$ et $q^d(t)$: désignent les vitesses et position désirées dans l'espace articulaire.
- $k_p, k_d,$ et k_I : sont des matrices diagonales définies positives, de dimension $(n \times n)$, dont les éléments génériques sont respectivement les gains proportionnels k_{pj} dérivés k_{dj} et intégraux k_{Ij} .

Le calcul des gains k_{pj}, k_{dj} , et k_{Ij} est effectué en considérant le modèle de l'articulation j représenté par le système linéaire du deuxième ordre à coefficients constants suivant :

$$\Gamma_j = a_j \ddot{q}_j + F_{vj} \dot{q}_j + \gamma_j \quad (III.6)$$

L'équation dans laquelle $a_j = A_{jj\max}$ désigne la valeur maximale de l'élément A_{jj} de la matrice d'inertie du robot et γ_j représente un couple perturbateur.

La fonction de transfert en boucle fermée pour $g = 0$ est alors donnée par :

$$\frac{q_j(s)}{q^d_j(s)} = \frac{k_{dj}s^2 + k_{pj}s + k_{lj}}{a_j s^3 + (k_{dj} + F_{vj})s^2 + k_{pj}s + k_{lj}} \quad (III.7)$$

Et l'équation caractéristique s'écrit :

$$\Delta(s) = a_j s^3 + (k_{dj} + F_{vj})s^2 + k_{pj}s + k_{lj} \quad (III.8)$$

La solution la plus courante en robotique consiste à choisir les gains de manière à obtenir un pôle triple réel négatif, ce qui donne une réponse rapide sans oscillations. Par conséquent, l'équation caractéristique se factorise de la façon suivante :

$$\Delta(s) = a_j (s + \varpi_j)^3 \quad (III.9)$$

Telle que : $\varpi_j > 0$.

On en déduit pour les gains :

$$\begin{cases} k_{pj} = 3a_j \varpi_j^2 \\ k_{dj} + F_{vj} = 3a_j \varpi_j \\ k_{lj} = a_j \varpi_j^3 \end{cases} \quad (III.10)$$

Remarque :

- ϖ_j est choisi le plus grand possible « la pulsation ne devra pas être supérieure à la pulsation de résonance ϖ_{rj} » afin de ne pas déstabiliser le système ; $\varpi_j = \varpi_{rj} / 2$.
- Le terme d'anticipation k_{dj} de l'équation permet de réduire les erreurs de suivi du mouvement désiré.
- Les performances d'une telle méthode sont d'autant plus acceptables que le rapport de réduction est important (augmentant ainsi la partie invariante de A_{jj}), que les vitesses sont faibles et que les gains proportionnels et dérivés sont grands. [28]

Lorsque les effets de la gravité sont compensés, par construction mécanique comme pour le robot SCARA ou grâce à la commande, on montre qu'une loi de type PD est asymptotiquement stable pour une position désirée q^d fixe. [29]

La démonstration s'appuie sur la définition d'une fonction de Lyapunov ayant la forme suivante :

$$V = \frac{1}{2} \dot{q}^T A(q) \dot{q} + \frac{1}{2} e^T K_p e \quad (III.11)$$

Où :

- e : désigne l'erreur de consigne ($e = q^d - q$).
- q^d : est constant.

Alors la loi de commande PD est comme suit :

$$\Gamma = K_p e - K_d \dot{q} + Q(q) \quad (\text{III.12})$$

A partir des équations (III.1) et (III.12), on obtient en négligeant les frottements :

$$K_p e - K_d \dot{q} = Aq'' + Cq' \quad (\text{III.13})$$

En dérivant la fonction V (III.11), on trouve :

$$\dot{V} = \frac{1}{2} \dot{q}^T A \dot{q} + \dot{q}^T Aq'' - e^T K_p \dot{q} \quad (\text{III.14})$$

Et, après substitution de Aq'' par sa valeur à partir de l'équation (III.13), on obtient :

$$\dot{V} = \frac{1}{2} \dot{q}^T (A - 2C) \dot{q} - \dot{q}^T K_d \dot{q} \quad (\text{III.15})$$

Telle que :

- La matrice $(A - 2C)$ étant antisymétrique.
- Le terme $\dot{q}^T (A - 2C) \dot{q}$ est nul.

Donc on en conclut que :

$$\dot{V} = -\dot{q}^T K_d \dot{q} \leq 0. \quad (\text{III.16})$$

Remarque :

- V diminue tant que $\dot{q} \neq 0$ est ce qui n'est pas suffisant pour démontrer la stabilité asymptotique. En effet, on doit s'assurer que lorsque $\dot{q} = 0$ le robot n'atteint pas une configuration dans laquelle $q \neq q_d$. Pour montrer que ce n'est pas le cas, on utilise le principe d'invariance de La Salle.
- Pour que $\dot{V} = 0$ il faut nécessairement que $\dot{q} = 0$ et par conséquent que
 - $q'' = 0$.
- A partir de l'équation du mouvement (III.13) on obtient alors que $e = 0$. L'équilibre
 - $(e = 0, \dot{q} = 0)$ est donc l'ensemble invariant le plus grand inclus dans V et on en déduit que la stabilité est asymptotique au point d'équilibre.
- La stabilité du système est assurée si l'on utilise dans l'équation (III.12) le terme $Q(q_d)$, constant pour q_d donné, qui correspond aux couples de gravité sur la position finale, plutôt que le terme $Q(q)$. [30]
- La stabilité est aussi assurée si l'on prend $K_{pj} > \|\partial Q(q)/\partial q\|$, terme qui représente la norme 2 de la matrice jacobienne des couples de gravité par rapport au vecteur des variables articulaires q. [31]

Commande de PID dans l'espace cartésien

Lorsque le mouvement est défini dans l'espace cartésien (opérationnel), une des deux solutions suivantes peut être choisie pour réaliser la commande du système :

1. On transforme le mouvement défini dans l'espace opérationnel en un mouvement dans l'espace articulaire, puis on met en œuvre la commande dans l'espace articulaire. Le signal d'erreur minimisé est alors exprimé dans l'espace articulaire ;
2. On spécifie directement la commande dans l'espace opérationnel.

Pour une commande PID dans l'espace cartésien, la loi de commande est obtenue en remplaçant q par X dans l'équation (III.5) et en multipliant l'erreur dans l'espace opérationnel par J^T pour l'exprimer dans l'espace articulaire.

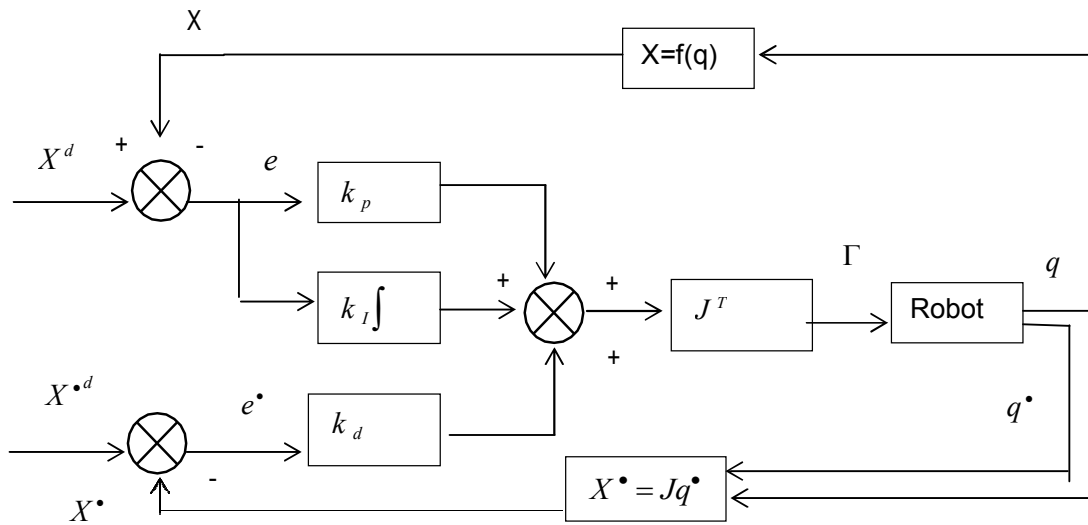


Figure III.3 Schéma d'une commande PID dans l'espace cartésien.

$$\Gamma = J^T \begin{bmatrix} K_p X^d - X + K_d X^{\cdot} - X + K_I \int_{t_0} X^d \\ - X^d \end{bmatrix} \quad (III.17)$$

Pour décrire le mouvement désiré dans l'espace articulaire lorsque celui-ci est spécifié dans l'espace cartésien, il y'a deux solutions sont possibles :

- soit seul le modèle géométrique inverse est utilisé, puis avec une procédure numérique, on dérive la position désirée pour obtenir la vitesse et l'accélération désirées.
- soit on calcule les positions, vitesses et accélérations articulaires :
 - A partir du modèle géométrique inverse (MGI) pour les positions articulaires :

$$q^d = MGI (X^d) .$$

- A partir du modèle cinématique inverse dans les positions régulières :

$$q^{d \bullet} = J (q^d)^{-1} X^{d \bullet} .$$

Remarque :

Dans les positions singulières ou pour les robots redondants, on remplace J^{-1} par une inverse généralisée telle que le pseudo inverse.

- A partir du modèle cinématique inverse du deuxième ordre :

$$\ddot{q}^d = J^{-1} (\ddot{X}^d - \dot{J} \dot{q}^d) \tag{III.18}$$

Avec :

$$\dot{J} (q^d, \dot{q}^d) = \frac{d}{dt} J (q^d) \tag{III.19}$$

Commande par mode glissant d'un bras manipulateur

Un modèle mathématique est souvent une description approximative de la réalité physique, et la loi de contrôle ne peut être construite que sur elle. La commande choisie doit être robuste en ce sens qu'elle doit garantir une faible sensibilité aux erreurs et incertitudes des paramètres, à leurs variations et aux perturbations.

Le contrôle par modes glissants qui n'est autre qu'un cas particulier de la théorie des systèmes à structures variables et multifonctions. Fondé sur la résolution des équations différentielles avec un second membre discontinu. [31]

Principe de la commande par modes glissants

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.

Le principe de la commande par modes glissants est de contraindre l'état du système à atteindre en temps fini une surface (dans l'espace d'état) donnée pour ensuite y rester. Cette surface étant une relation entre les variables d'état du système, elle définit une équation

différentielle, et donc détermine totalement la dynamique du système, pourvu que le système reste sur cette surface. [32]

L'évolution d'un système soumis à une loi de commande qui le fait rester sur une surface donnée ne dépend donc plus du système lui-même ou des perturbations auxquelles il peut être soumis, mais uniquement des propriétés de cette surface. Le système bouclé n'est donc pas seulement robuste vis-à-vis des incertitudes (propres au système) et des perturbations (extérieurs au système), mais totalement insensible à ces incertitudes et perturbations, moyennant qu'elles puissent effectivement être rejetées par la commande.

La trajectoire dans le plan de phase est constituée de trois parties distinctes :

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

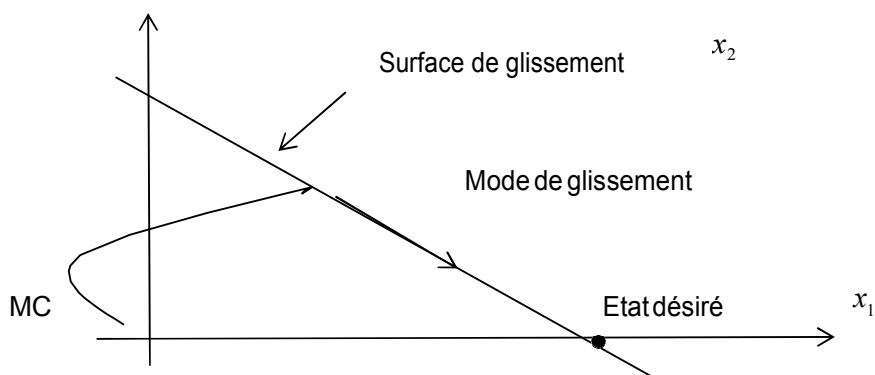


Figure III.4 Différents modes des trajectoires d'état.

Conception de la commande par mode de glissement

La conception des régulateurs par les modes glissants prend en charge les problèmes de stabilité et des performances désirées d'une façon systématique. La mise en œuvre de cette méthode de commande nécessite principalement trois étapes :

a. Choix des surfaces de glissement

Le choix de la surface de glissement concerne non seulement le nombre nécessaire de ces surfaces mais également leur forme, en fonction de l'application et de l'objectif visé.

En général, pour un système défini par l'équation d'état suivante :

$$\dot{x} = \begin{cases} f(x) + g(x)u \\ h(x) \end{cases} \quad (\text{III.20})$$

Il faut choisir (m) surfaces de glissement, pour un vecteur [U] de dimension (m). Pour ce qui est de la forme de la surface. Nous proposons une forme d'équation générale pour déterminer la surface de disc.

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

Tel que :

$$e(x) = y(x) - y_{ref}(x) \quad (\text{III.22})$$

Avec :

- λ_x : Constante positive.
- r : Degré relatif, égale au nombre de fois qu'il fait dériver la sortie pour faire

Apparaître la commande, c'est à dire le plus petit entier positif tel que : $\frac{\partial s(x)}{\partial u(x)} \neq 0$.

Pour :

- $r=1 : S(x) = e(x)$
- $r=2 : S(x) = \lambda_x e(x) + \dot{e}(x)$
- $r=3 : S(x) = \lambda_x^2 e(x) + 2\lambda_x \dot{e}(x) + \ddot{e}(x)$

Pour $r > 0$, $S(x) = 0$ est une équation différentielle linéaire dont la réponse $e(x)$ tend vers zéro pour un choix correct du gain λ_x .

En d'autres termes, la difficulté revient à un problème de poursuite de trajectoire dont l'objectif est de garder $S(x)$ à zéro. Ceci est équivalent à une linéarisation exacte de l'écart tout en respectant une condition de convergence. [33]

a. Condition d'existence et de convergence

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

1. La loi de commutation directe

C'est la première condition de convergence .elle a été proposée par Emelyanov [41] et Utkin [34] et elle est donnée par :

$$S(x) \cdot \dot{S}(x) < 0. \tag{III.23}$$

□ La fonction de Lyapunov

Il s'agit de formuler une fonction scalaire positive $V(x) > 0$ pour les variables d'état du système et de choisir une loi de commande qui fera décroître cette fonction.

$$\dot{V}(x) < 0 \tag{III.24}$$

La fonction de Lyapunov est donnée comme suit :

$$V(x) = \frac{1}{2} S^2(x). \tag{III.25}$$

Sa dérivée sera :

$$\dot{V}(x) = S(x) \cdot \dot{S}(x). \tag{III.26}$$

Pour que la surface $s(x) = 0$ soit attractive sur tout le domaine de fonctionnement, il suffit que

$$V^{\bullet}(x) = S(x).S^{\bullet}(x) < 0. \quad (\text{III.27})$$

□ Calcul de la commande

Une fois la surface de glissement est choisie et la condition de convergence est vérifiée, on construit une loi de commande pour assurer l'attraction des trajectoires d'état vers la surface de glissement $s(x)=0$.

La loi de commande en mode de glissement est composée de deux grandeurs :

- La commande équivalente u_{eq} .
- La commande discontinue u_d .

La loi de commande globale est donnée par la relation suivante :

$$U = U_{eq} + U_d$$

- La commande équivalente

La commande équivalente est une fonction linéaire, peut être interprétée comme étant la valeur moyenne (continue) que peut prendre la commande u lors de la commutation rapide entre u^- et u^+ comme le montre la figure derrière.

Elle est déterminée en résolvant l'équation du comportement dynamique du système durant le mode de glissement :

$$S^{\bullet}(x) = 0$$

$$S^{\bullet} = \frac{\partial s}{\partial x} \frac{\partial x}{\partial t} = 0 \quad (\text{III.28})$$

Alors $S^{\bullet}(x)$ s'écrit par :

$$S^{\bullet}(x) = \frac{\partial s}{\partial x} (f(x) + g(x)u_{eq}) = 0 \quad (\text{III.29})$$

On tire l'équation de u_{eq} à partir de l'équation (III.29) :

$$u_{eq} = - \left(\frac{\partial s}{\partial x} g(x) \right)^{-1} \frac{\partial s}{\partial x} f(x) \quad (III.30)$$

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

L'équation du régime glissant idéal est obtenue en remplaçant l'expression de u_{eq} dans :

$$\dot{x} = f(x) - g(x) \left(\frac{\partial s}{\partial x} g(x) \right)^{-1} \frac{\partial s}{\partial x} f(x) \quad (III.31)$$

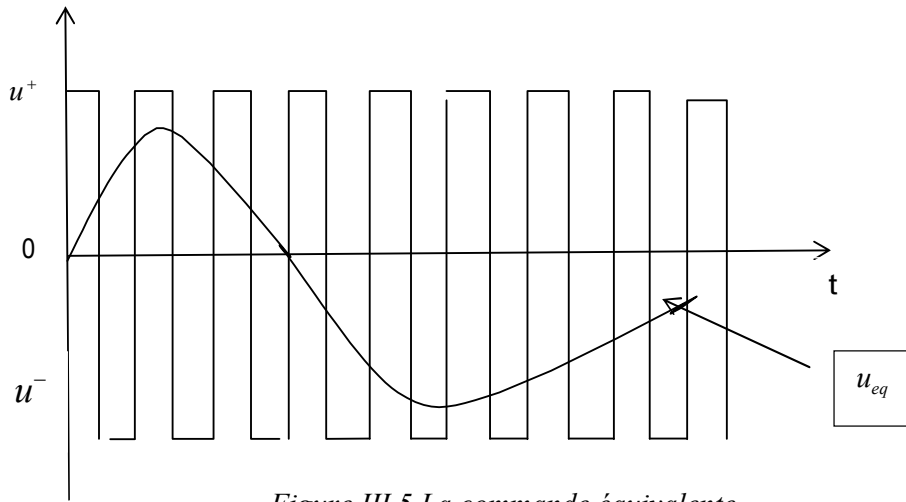


Figure III.5 La commande équivalente.

2. La commande discontinue

La loi de commande discontinue est déterminée pour satisfaire les conditions de convergence. Cette commande force les dynamiques à converger vers la surface et assure l'insensibilité du système vis-à-vis des incertitudes et perturbations par exemple pour des paramètres mal connus, le système ne glisse pas parfaitement sur la surface il quitte celle-ci, mais le terme discontinu l'y ramène car la surface est attractive. C'est pour cette raison que la commande par modes glissants est dite robuste.

En remplaçant l'expression de u_{eq} dans ($u = u_{eq} + u_d$) on obtient :

$$u = - \left(\frac{\partial s}{\partial x} g(x) \right)^{-1} \frac{\partial s}{\partial x} f(x) + u_d \quad (III.32)$$

En remplaçant l'expression de u dans $S^*(x)$, on obtient :

$$S^{\bullet}(x) = \frac{\partial s}{\partial x} \left(f(x) + g(x) \left(\frac{\partial s}{\partial x} g(x) \right)^{-1} \frac{\partial s}{\partial x} f(x) + g(x) u_d \right) \quad (III.33)$$

Après un bref calcul on aura :

$$S^{\bullet}(x) = \frac{\partial s}{\partial x} g(x) + u_d \quad (III.34)$$

Pour assurer la condition de l'attractivité :

$$S^{\bullet}(x)S(x) < 0 \Rightarrow S(x) \left(\frac{\partial s}{\partial x} g(x) + u_d \right) < 0 \quad (III.35)$$

Il suffit que u_d soit de signe opposé à $S(x) \frac{\partial s}{\partial x} g(x)$:

Pour le terme u_d , différentes formes sont proposées dans la littérature. [43]

Toutes les formes proposées donnent un terme discontinu. La forme simple qui est généralement utilisée est : $u_d = -k \text{signe}(s)$

Où : k est une constante positive et signe est la fonction signe classique.

$$\text{signe}(s) = \begin{cases} +1 & s(x) > 0 \\ 0 & s(x) = 0 \\ -1 & \text{si } s(x) < 0 \end{cases} \quad (III.36)$$

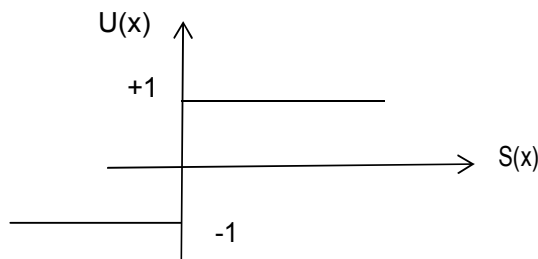


Figure III.6 La fonction signe.

Application de la commande par mode de glissement pour la suite de trajectoire d'un bras à 2DDL

L'équation dynamique d'un bras manipulateur est donnée comme suit :

$$M(q)\ddot{q} + c(q, \dot{q})\dot{q} + G(q) = \tau \quad (\text{III.37})$$

Les erreurs sont définies comme suit :

$$e = q - q_d \quad \Rightarrow \quad \begin{cases} \dot{e} = \dot{q} - \dot{q}_d \\ \ddot{e} = \ddot{q} - \ddot{q}_d \end{cases} \quad (\text{III.38})$$

La surface de glissement est choisie comme :

$$s = \dot{e} + ce \quad / C : \text{est défini positif.}$$

Le signal d'erreur de la poursuite est donné comme suit :

$$\dot{q}_r = \dot{q}_d - ce \quad (\text{III.39})$$

Avec :

- q_d : est la position de sure e^* .

A partir de l'équation de la surface de glissement on trouve :

$$\begin{aligned} s = \dot{e} + ce &= \dot{q} - \dot{q}_d + ce = \dot{q} - \dot{q}_r \\ &= \dot{q} - \dot{q}_d + ce = \dot{q} - \dot{q}_r \end{aligned} \quad (\text{III.40})$$

On a :

$$\dot{s} = \ddot{q} - \ddot{q}_r$$

Alors :

$$\begin{aligned} M\dot{s} &= M\ddot{q} - M\ddot{q}_r \\ &= \tau - c\dot{q} - G - H\ddot{q}_r \end{aligned} \quad (\text{III.41})$$

On a aussi :

$$\dot{s} = csng(s) \quad (\text{III.42})$$

Alors la loi de commande est obtenue comme suit :

$$\tau = cq' + G + Mq''_r - Mcsign(s) \quad (III.43)$$

Etude de la stabilité :

La fonction de Lyapunov est choisie comme suit :

$$V = \frac{1}{2} S^T S \quad (III.44)$$

$$\dot{V} = \frac{1}{2} S^T \dot{S} \quad (III.45)$$

Avec :

$$\dot{S} = \dot{\phi} - \dot{\phi}_r \quad (III.46)$$

$$\dot{V} = S^T (\dot{\phi} - \dot{\phi}_r) \quad (III.47)$$

$$\dot{\phi} = M^{-1}(\tau - c\dot{\phi} - G)$$

Alors :

$$\dot{V} = S^T (M^{-1}(\tau - c\dot{\phi} - G) - \dot{\phi}_r) \quad (III.48)$$

$$= S^T (M^{-1}(cq' + G + Mq''_r - Mcsign(s) - (c\dot{\phi} - G) - \dot{\phi}_r))$$

$$\dot{V} = S^T csign(s) \quad (III.49)$$

On a

- C : est défini positif.

Alors : $\dot{V} \leq 0$.

Résultats et Simulation

Les résultats de simulation sont présentés sous MATLAB/SIMULINK, où les simulations sont subdivisées en trois parties :

- Commande PID dans l'espace articulaire
- Commande PID dans l'espace cartésien

➤ Commande par mode glissant.

1. La commande PID dans l'espace articulaire

La loi de commande est donnée par :

$$\Gamma = k_p(q^d - q) + k^d(q^{\bullet d} - q^{\bullet}) + k_f \int_{t_0}^t (q^d - q) d\tau .$$

Les gains sont choisis comme : $K_p1=1000; K_v1=20; K_p2=1000; K_v2=20$.

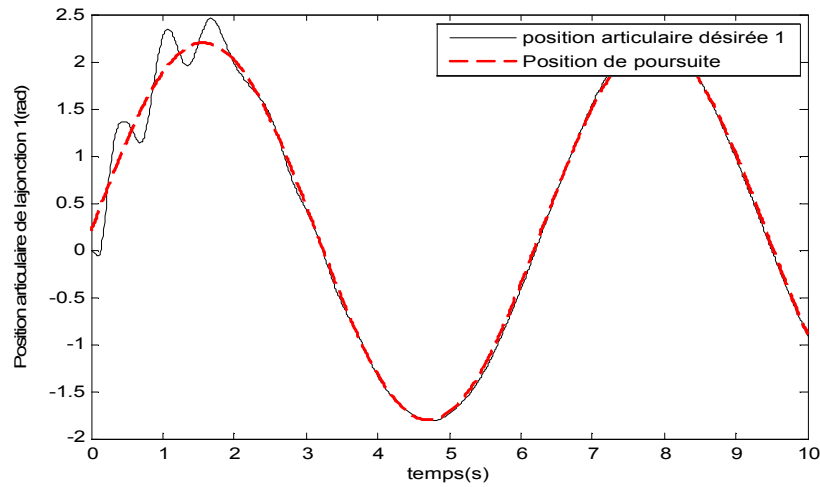


Figure III.7. Position articulaire de la jonction 1.

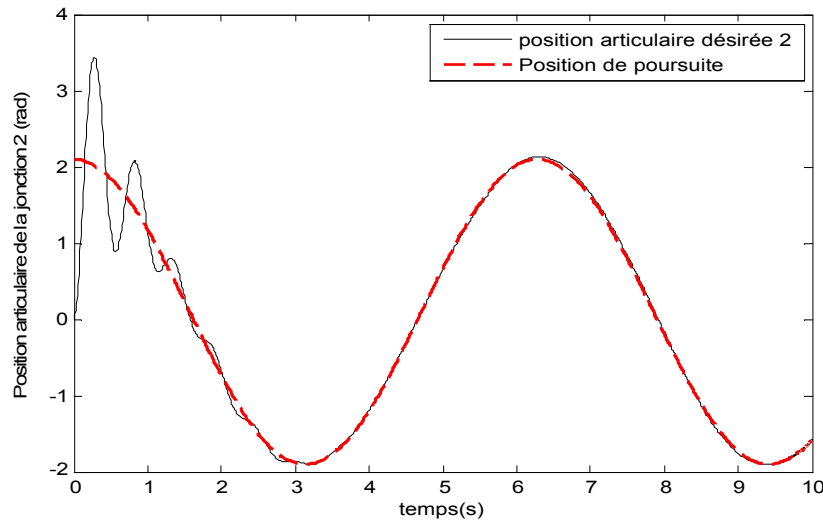


Figure III.8. Position articulaire de la jonction 2.

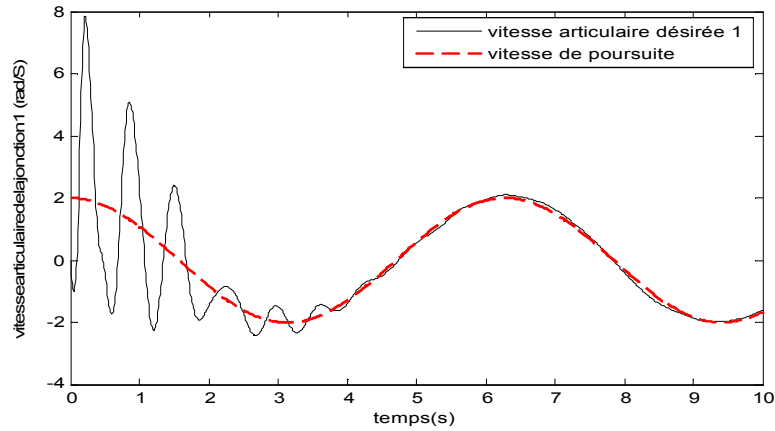


Figure III.9. Vitesse articulaire de la jonction 1.

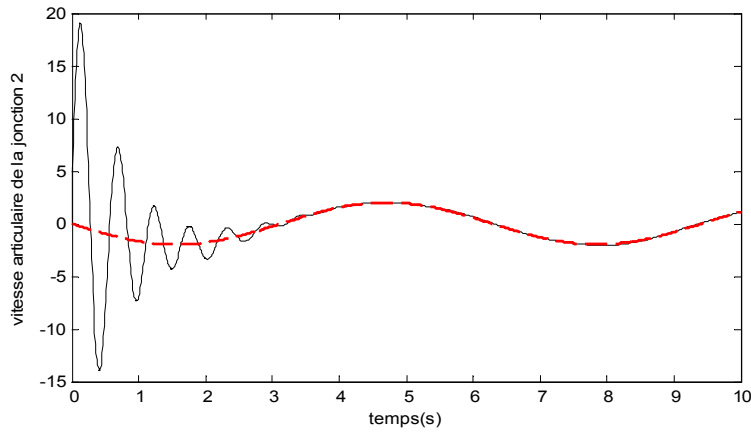


Figure III.10. Vitesse articulaire de la jonction 2.

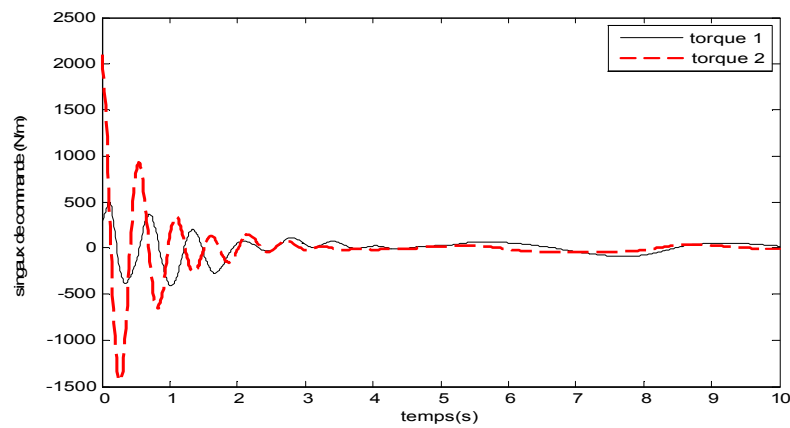


Figure III.11. Signaux de commande (N/m).

Les figures III.7 et III.8 présentent les positions articulaires de la première jonction et la deuxième jonction respectivement où la commande proposée montre une bonne poursuite et une convergence vers les positions articulaires de référence proposées.

Les figures III.9 et III.10 présentent les vitesses articulaires des deux jonctions où elles sont convergées vers les vitesses articulaires de référence.

La figure III.11 présente les torques des deux jonctions qui représentent les signaux de commande, dont on remarque que les torques convergent vers zéro une fois les positions articulaires convergentes vers leurs positions désirées.

2. La commande PID dans l'espace cartésien

La loi de commande est :

$$\Gamma = J \left[\begin{array}{c} K_p X - X + K_d \dot{X} \\ K_p X - X + K_d \dot{X} + K_i \int_{t_0}^t X - X_d dt \end{array} \right] \tau$$

Les gains sont choisis comme suit :

$$K_p1=5000;K_v1=200;k_d1=0;K_p2=7000;K_v2=200;k_d2=0.$$

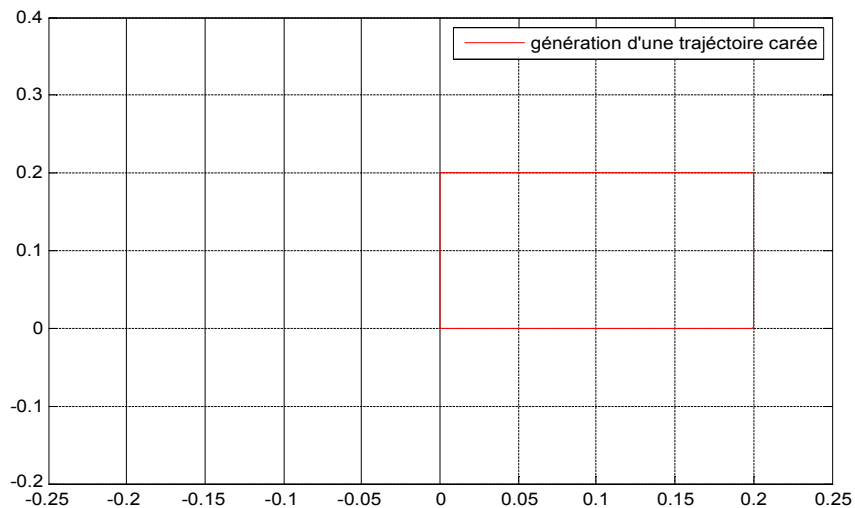


Figure III.12. Génération d'une trajectoire carrée.

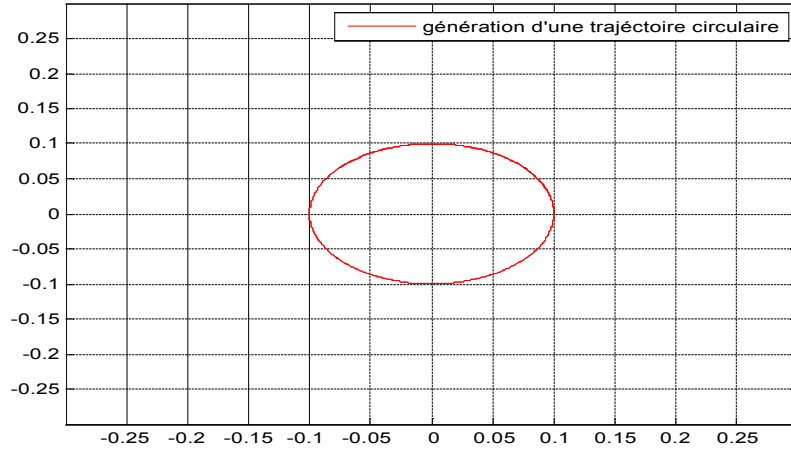


Figure III.13. Génération d'une trajectoire circulaire.

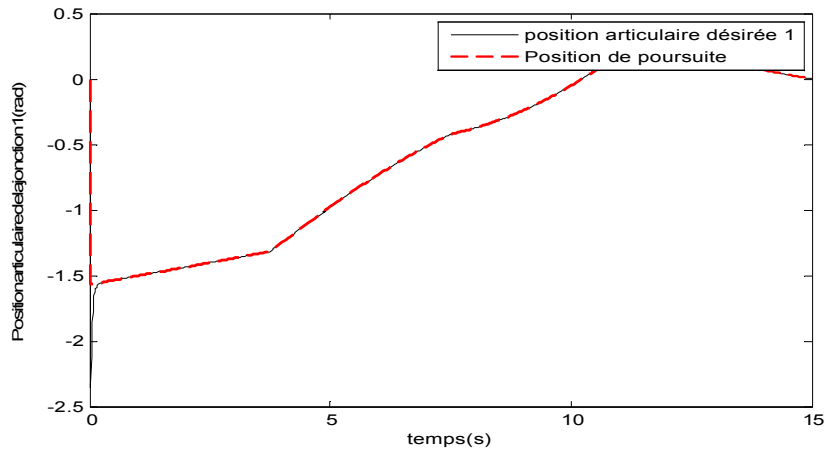


Figure III.14. Position articulaire de la jonction 1.

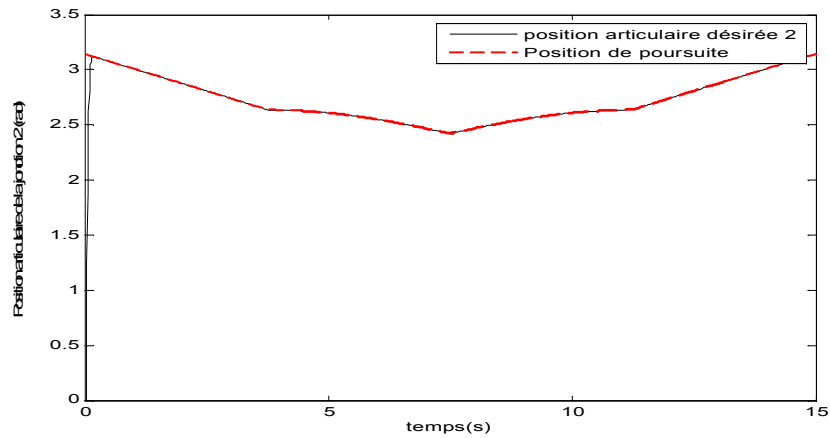


Figure III.15. Position articulaire de la jonction 2.

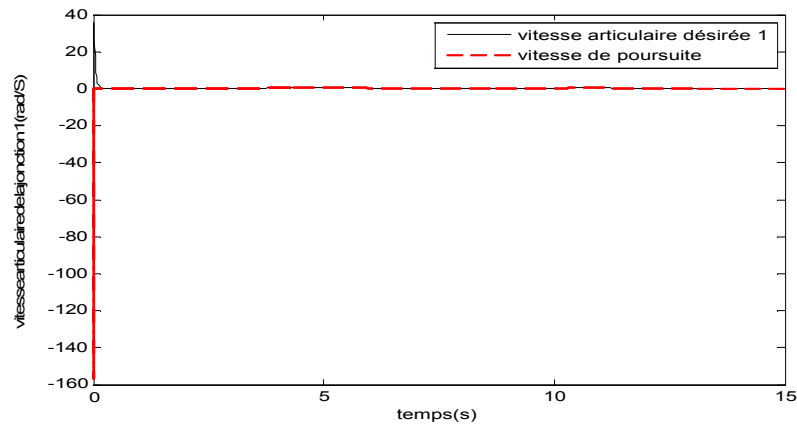


Figure III.16. Vitesse articulaire de la jonction 1.

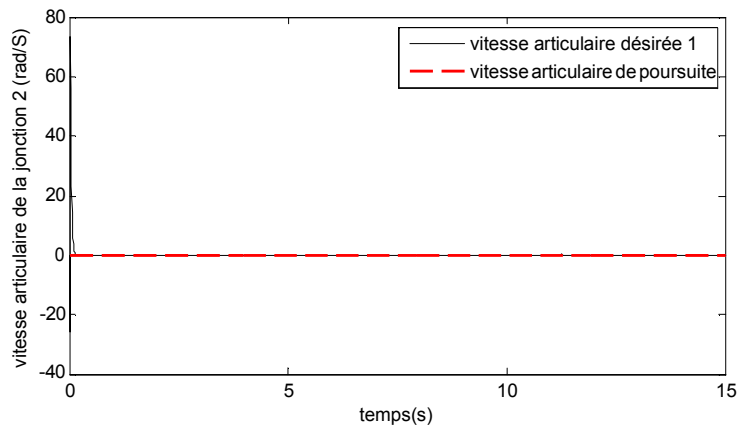


Figure III.17. Vitesse articulaire de la jonction 2.

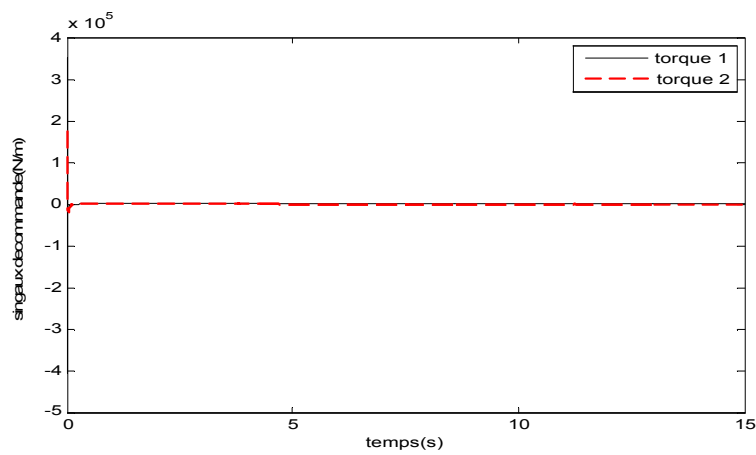


Figure III.18. Signaux de commande (N/m).

Les figures III.14 et III.15 présentent les positions articulaires de la première jonction et la deuxième jonction respectivement où éventuellement la commande PID montre une bonne poursuite et une convergence vers les positions articulaires de référence proposées.

Les figures III.16 et III.17 présentent les vitesses articulaires des deux jonctions où elles sont convergées vers les vitesses articulaires de référence.

La figure III.18 présente les torques des deux jonctions qui représentent les signaux de commande.

D'après les résultats obtenus par la commande PID soit dans l'espace articulaire où cartésien on remarque que cette technique présente des bonnes performances en absence des perturbations.

3. La commande mode glissant

L'équation dynamique d'un bras manipulateur est :

$$M(q)\ddot{q} + c(q, \dot{q})\dot{q} + G(q) = \tau$$

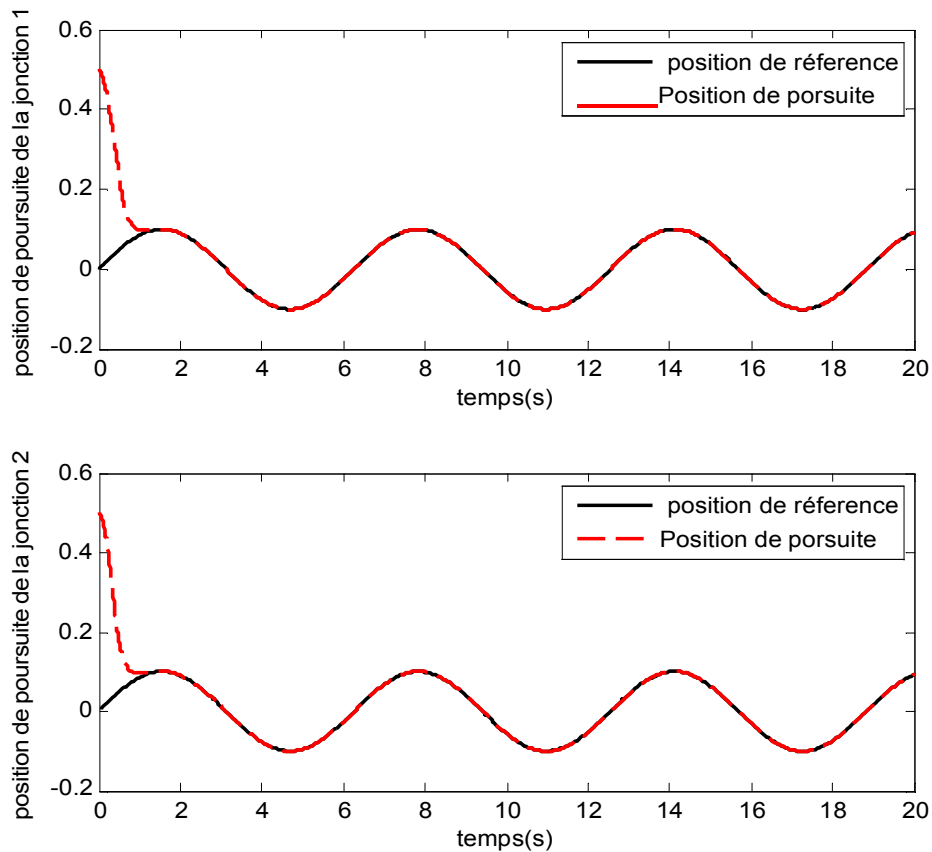


Figure III.19. Position de poursuite des jonctions 1 et 2.

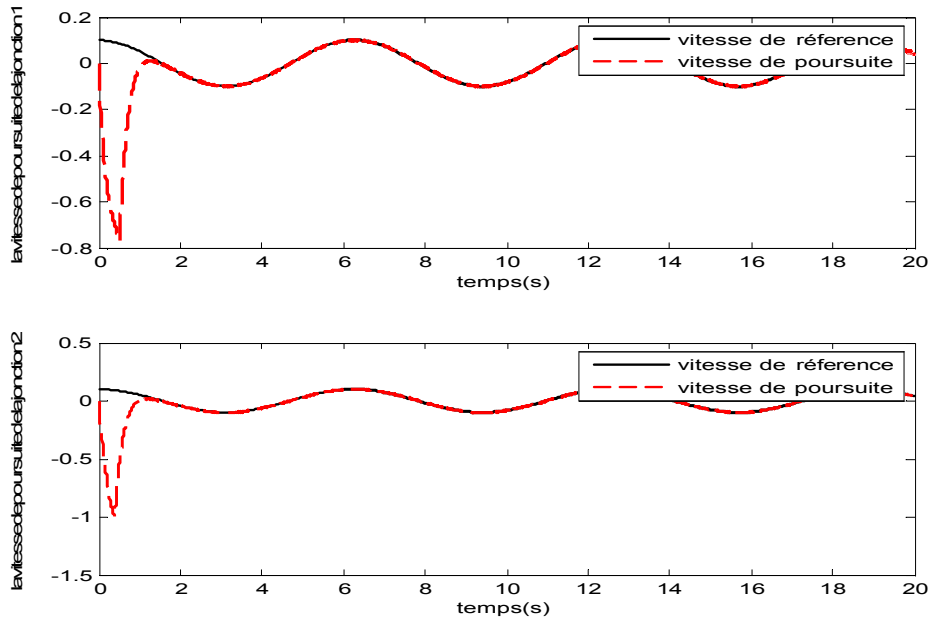


Figure III.20. Vitesse de poursuite des jonctions 1 et 2.

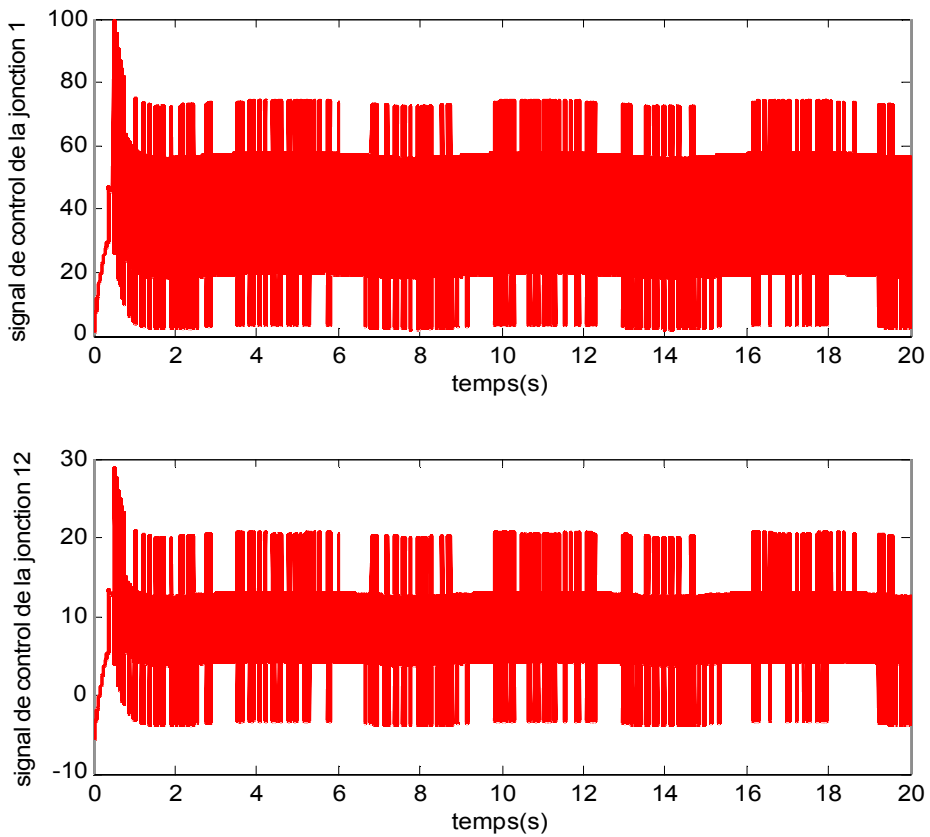


Figure III.21. Signaux de commande des jonctions 1 et 2.

La figure III.19 présente les positions articulaires des deux jonctions où les positions reçoivent les positions de référence dans un temps limité (inférieur à 1.5 s) qui est l'une des caractéristiques de la commande par mode glissant.

La figure III.20 présente les vitesses articulaires qui reçoivent les vitesses de référence après un temps limité.

La figure III.21 présente les signaux de commande qui sont les torques des articulations. L'inconvénient majeur de cette commande qui est le phénomène de chattering est apparu sur les torques. Pour éviter ce phénomène plusieurs commandes peuvent être introduites, où on peut citer :

- Quasi mode glissant
- Mode glissant terminal
- Mode glissant basé sur les lois d'arrivée.

III.5. Conclusion

Dans ce chapitre, nous avons étudié quelques techniques de commandes classiques soit la commande PID où la commande par mode glissant conventionnelle. La commande PID appliquée sur deux stratégies de contrôle (espace articulaire et cartésien) qui nous a donné des résultats acceptables en vue de la convergence des erreurs en absence des perturbations. La commande par mode glissant proposée dans ce chapitre présente des performances de rapidité de la convergence des erreurs qui est l'une des caractéristiques de cette technique, mais l'inconvénient de cette technique qui est le phénomène de broutement (chattering) fait apparaître dans les signaux de commande (les torques). Malgré le problème de chattering les résultats obtenus présentent une efficacité de la commande par mode glissant pour un bras manipulateur.

Les simulations proposées sont développées sous MATLAB/SIMULINK qui nous permet de visualiser et présenter nos résultats de simulations.

CHAPITRE IV

COMMANDE MODE GLISSANT NEURONALE APPLIQUEE POUR LA POURSUITE D'UN BRAS MANIPULATEUR

Introduction

Le principe des réseaux de neurones artificiels est né dans les années 40 à partir d'une analogie avec le système nerveux humain. Le terme désigne aujourd'hui un très grand nombre de modèles, dont beaucoup n'ont plus grand chose à voir avec le fonctionnement des neurones biologiques, et doit donc être pris comme une métaphore. Ces différents modèles ont en commun l'utilisation d'automates, appelés neurones ou unités, capables de réaliser chacun un traitement très simple et d'échanger des informations entre eux. On associe généralement aux réseaux un algorithme « d'apprentissage » permettant de modifier de manière plus ou moins automatique le traitement effectué afin de réaliser une tâche donnée. Nous n'aborderons ici les réseaux qu'avec la vision que pourrait en avoir un ingénieur en oubliant les aspects liés aux sciences cognitives ou à la neurobiologie. Notons que l'utilisation de réseaux de neurones artificiels (RNA) pour la réalisation de systèmes de commande a connu un essor important au cours des dernières années.

Dans ce chapitre nous tentons de présenter les réseaux de neurones à travers des définitions et leurs propriétés. On présente notamment plusieurs types et règles d'apprentissage. On propose aussi dans ce chapitre de connaître les réseaux RBF (Radial Basis Functions). On y retrouve l'architecture de ces réseaux, leur fonctionnement ainsi que leur apprentissage. A la fin du chapitre, un résumé sur l'approximation de fonctions non linéaires par le RBF est donné, ce dernier sera utile dans la conception des techniques de commande adaptative.

Le neurone formel

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

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

La figure IV.1 représente un neurone formel appliquant une fonction de seuil sur la somme pondérée de ses différentes entrées.

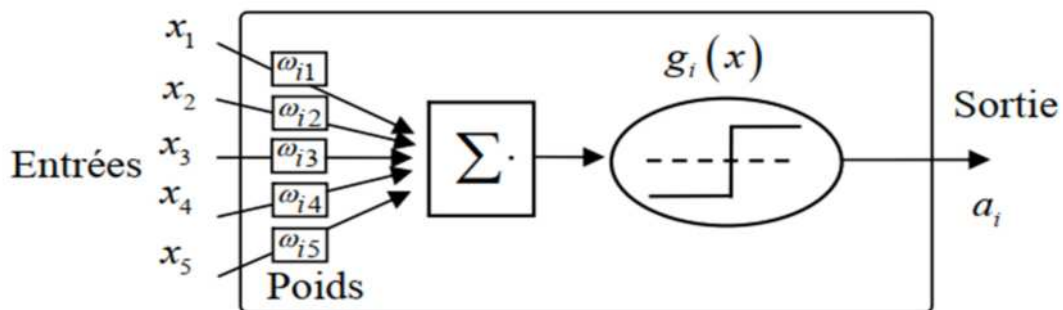


Figure IV.1 : Un neurone formel.

Le perceptron multicouche (Multi-Layer Perceptron, MLP)

Un des premiers chercheurs en connexionnisme, FRANK ROSENBLATT [36], s'intéressait la possibilité d'expliquer les capacités d'association des êtres humains par des capacités équivalentes dans des modèles neuromimétiques. Ainsi, le Perceptron devait modéliser le fonctionnement de la perception humaine. Relié à une caméra, ce réseau de neurones à seuil recevait sur sa couche d'entrée (la rétine) une représentation digitalisée du champ de vision de la caméra. Cette première couche était connectée à une seconde couche (l'aire d'association) qui avait la responsabilité de repérer un ensemble de traits élémentaires. Enfin, la couche de sortie (aire de réponse) devait reconnaître l'objet sous l'œil de la caméra.

Les MLP, ou réseaux à couches, forment la très grande majorité des réseaux. Ils sont intemporels (ce sont des réseaux statiques et non dynamiques).

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

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

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

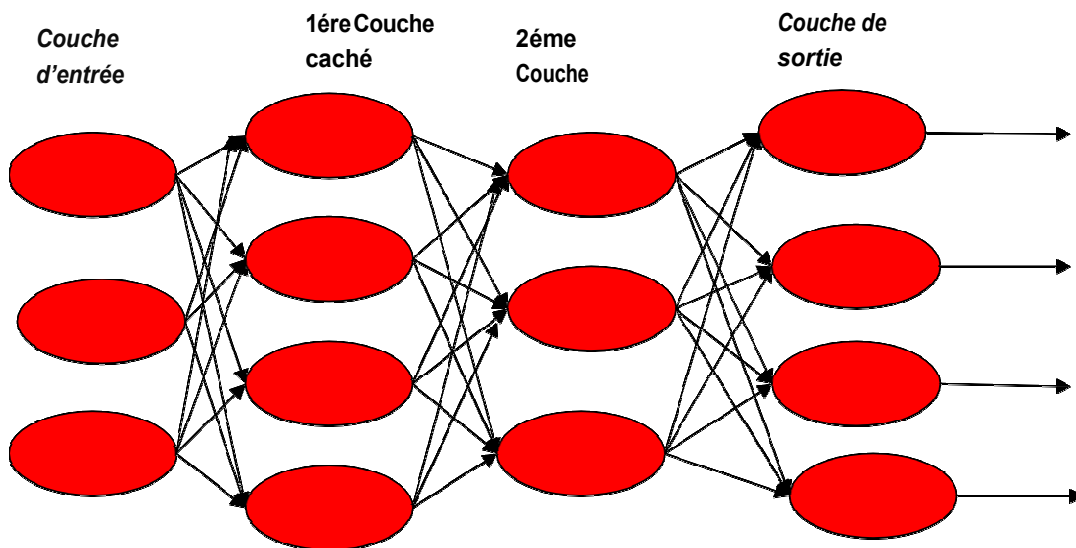


Figure IV.2 : Un réseau de 03 couches.

IV .4. Apprentissage

La notion d'apprentissage est claire et intuitive pour les humains ou les animaux : c'est une procédure cognitive qui doit faire en sorte que l'individu réalise de manière autonome une tâche donnée, typiquement à partir d'exemples.

Pour un réseau de neurones, l'apprentissage peut être considéré comme les problèmes de la mise à jour des poids des connexions au sein du réseau afin de réussir la tâche qui lui est demandée.

IV .4.1. Définitions

L'apprentissage est une phase de développement d'un réseau de neurones durant laquelle une modification du comportement du réseau se fait jusqu'à l'obtention d'un comportement désiré.

Durant cette phase les poids synaptiques seront modifiés, donc d'une manière plus simple, on peut dire que l'apprentissage est une action de mise à jour des poids des connexions dans le but d'atteindre la sortie du réseau pour les exemples présentés en entrée. L'apprentissage est peut se faire de différentes manières et selon différentes règles.

Type d'apprentissage

Au niveau des algorithmes d'apprentissage, il existe trois types selon l'intervention ou non d'un superviseur :

a) Apprentissage supervisé :

Ce type d'apprentissage est le plus courant, où dans ce type d'apprentissage, le réseau s'adapte par comparaison entre les résultats qu'il a calculé, en fonction des entrées fournies, et la réponse attendue en sortie. Ainsi, le réseau va se modifier jusqu'à ce qu'il trouve la bonne sortie, c'est à dire celle attendue, correspondant à une entrée donnée.

Cette apprentissage est dit supervisé, car il est nécessaire de connaître la sortie attendue pour chaque jeu d'entrées, le réseau de neurones va ajuster ses paramètres afin de minimiser l'erreur entre la sortie désirée et sa sortie réelle.

b) Apprentissage semi-supervisé :

Cet apprentissage est identique au précédent dans la mesure où il se base sur la présence d'un concepteur, cependant la valeur exacte de la sortie n'est pas disponible, en général la seule information disponible est un signal d'échec ou de succès.

c) Apprentissage non supervisé :

Ce type d'apprentissage est moins intuitif, où dans ce mode, l'apprentissage est basé sur des probabilités. Le réseau va se modifier en fonction des régularités statistiques de l'entrée et établir des catégories, en attribuant et en optimisant une valeur de qualité, aux catégories reconnues. Donc cet apprentissage repose sur un «critère interne» de conformité du comportement du réseau par rapport à des spécifications générales et non sur des observations.

IV .4.3. Règles d'apprentissage

L'apprentissage d'un Réseau de Neurones Artificiels se base sur des règles d'apprentissage. Nous citons ci-après les règles les plus répandues [37], [38], [39], [40]:

❖ **La règle de HEBB**

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

❖ **La règle delta**

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

La règle d'apprentissage compétitif qui ne concerne qu'un seul neurone. Son principe est de regrouper les données en catégorie. Les neurones similaires vont donc être rangés dans une

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

❖ *La rétro propagation du Gradient (back propagation)*

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

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

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

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

❖ *La règle de corrélation en cascade*

Cette règle est une technique d'apprentissage, ajoute progressivement des neurones cachés au réseau jusqu'à ce que l'effet bénéfique de ces nouveaux neurones ne soit plus perceptible. Cette règle suit les deux étapes suivantes:

1. On entraîne le système par un apprentissage classique qui s'effectue Premièrement dans un petit réseau sans couche cachée
2. On entraîne par la suite un petit groupe des neurones supplémentaires qui doit diminuer l'erreur résiduelle du réseau. La règle d'apprentissage utilisée modifie les

poins de ces neurones. Le neurone qui réussit le mieux est retenu, et intégré au réseau. L'étape 1 est relancée encore une fois, pour permettre au réseau de s'adapter à la nouvelle ressource

Réseaux de neurones à base radiale (RBF)

Il existe plusieurs modèles de RNA tel que RNA multicouches, RNA a fonctions de base radiales et RNA de type CMAC dans ce chapitre nous présentons RNA a fonctions de base radiales.

Introduction

Les réseaux RBF sont des réseaux à couche à couche qui ont comme origine une technique d'interpolation nommée méthode d'interpolation RBF, employée pour la première fois dans le contexte des réseaux neurométriques par BROOMHEAD et LOWE [41].

IV.5 .2 Architecture

Le réseau RBF est un réseau à deux couches, qui diffère du PMC par la fonction d'activation des cellules cachées qui est de forme gaussienne. Les neurones de la couche de cachée sont reliés aux entrées et ont chacun deux paramètres : un vecteur prototype "C" (centre des gaussiennes) et un coefficient d'étalement "a" strictement positif.

Les neurones de la couche de sortie sont généralement animés par une fonction d'activation linéaire.

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

$$f_1(\mathbf{x}) = e^{\frac{-\|\mathbf{x}-\mathbf{c}\|^2}{2\sigma^2}} \quad (\text{IV.1})$$

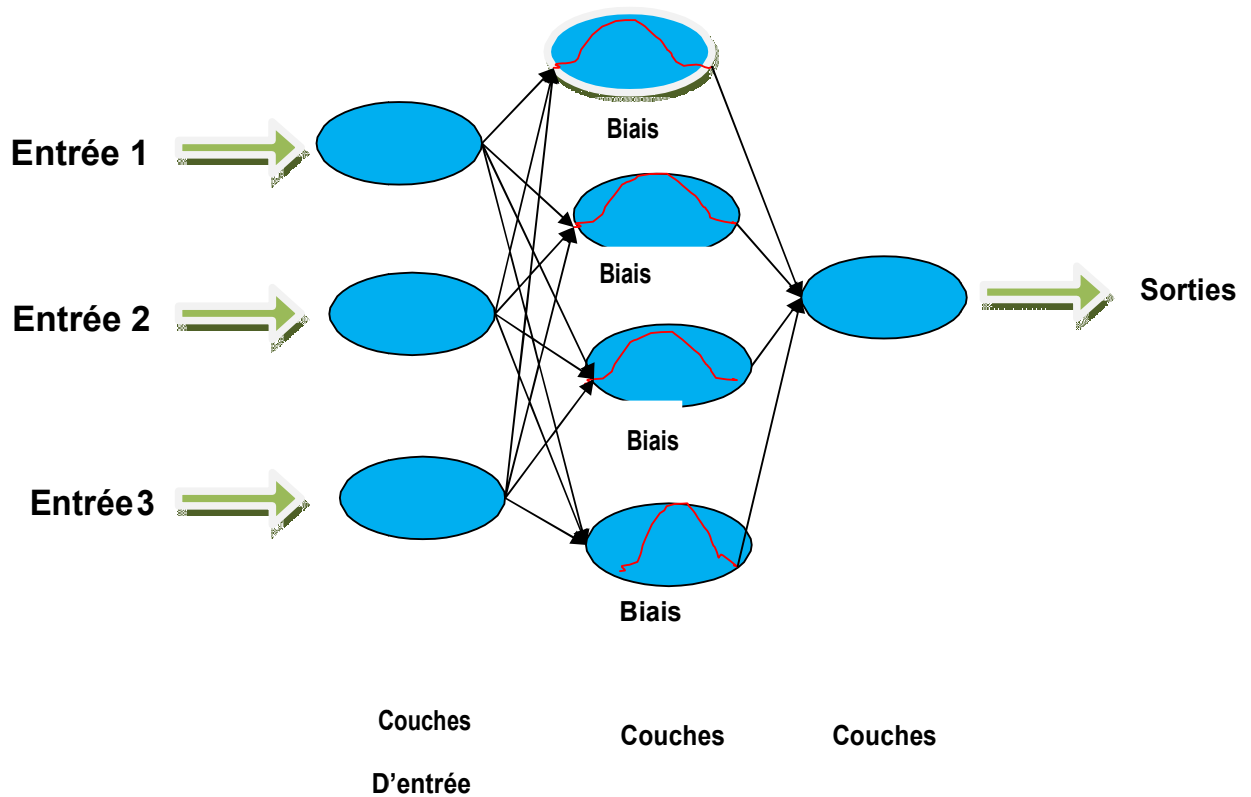


Figure IV.3 : Architecture d'un réseau de neurone RBF.

Avec C définit un point de l'espace d'entrée. La sortie du neurone est égale à 1 pour Une entrée x égale à C , puis décroît vers 0 lorsque l'entrée S 'éloigne de C . La vitesse de décroissance est réglée par a : Plus le coefficient est petit et plus la fonction sera concentrée autour du point C_i et proche de ailleurs. Les neurones de la seconde couche quant à eux calculent la sortie d'un réseau. En effectuant une combinaison linéaire des sorties de ceux de la première couche et un biais est ajouté au total. La fonction qu'ils réalisent est la suivante

$$f_2(X) = XW + b \quad (IV.2)$$

Où : X est un vecteur composé des sorties de tous les neurones de la première couche, W , est un vecteur de poids et b est le biais. W et b sont ajustés lors de l'apprentissage. Chaque sortie du réseau est donnée par la formule:

$$S_i(x) = \sum_{j=1}^N w_{ij} \exp\left(-\frac{\|s-c_j\|^2}{2a_j^2}\right) + e_i \quad (IV.3)$$

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

N : est le nombre de neurones de la première couche,

C_j : est le vecteur prototype du neurone numéro j de la première couche,

σ_j : Son coefficient d'étalement,

w_{ij} : ($j = 1, \dots, N$) sont les poids du neurone de sortie i et b_i son biais.

Apprentissage

Lors de l'apprentissage d'un réseau RBF, deux problèmes se posent : la constitution de la première couche (choix du nombre de neurones, choix des centres C_i et de coefficients d'étalement σ_j) et la détermination des poids et biais de la seconde couche.

a. Répartition des prototypes

Le choix du nombre de neurones sur la première couche et de leurs prototypes correspondants dépend fortement de la répartition des entrées, c'est à dire des vecteurs x_k de la base d'apprentissage. Pour établir ce choix, différentes méthodes sont applicables.

On peut répartir les prototypes uniformément ou aléatoirement dans l'espace d'entrée, ou bien sélectionner aléatoirement quelques vecteurs de la base d'apprentissage. Ces méthodes sont très simples, mais ne donnent pas de bons résultats car généralement la répartition obtenue n'est pas adaptée à la base. On peut alors utiliser des certains algorithmes afin de mieux les répartir dans l'espace d'entrée.

b. Apprentissage de la seconde couche

Une fois les prototypes placés, il reste à déterminer les paramètres de la seconde couche, c'est à dire les poids w_i et les biais b_i qui minimisent la fonction de performance du réseau. On peut calculer la sortie de la première couche pour chaque exemple de la base d'apprentissage :

$$Z_{k,j} = \exp\left(-\frac{\|s_k - c_j\|^2}{2\sigma_j^2}\right) \quad (\text{IV.4})$$

Où k est le numéro de l'exemple et j celui de la composante de z_k calculée, c'est à dire le numéro du neurone de la première couche dont on calcule la sortie. Le critère à minimiser est l'erreur quadratique moyenne du réseau sur la base d'apprentissage et a donc pour équation :

$$E = \frac{1}{p} \sum_{k=1}^p \sum_{i=1}^r (s_i(x_k) - y_{k,i})^2 = \frac{1}{p} \sum_{k=1}^p \sum_{i=1}^r (z_{k,i} \cdot w_{k,i} - y_{k,i})^2 \quad (IV.5)$$

Où $s_i(x_k)$ est la sortie du neurone i de la seconde couche, $y_{k,i}$ est la $i^{\text{ème}}$ composante de y_k (c'est à dire la sortie voulue pour l'exemple k sur le neurone i de la seconde couche) et r est le nombre de sorties. p est le nombre d'éléments dans la base d'apprentissage (x_k, y_k) . Un algorithme des moindres carrés peut être appliqué afin de trouver les paramètres w_i et b_i qui minimisent ce critère.

IV .6 Avantage et limites des RNAs

Les Réseaux de Neurones Artificiels ont plusieurs avantages dont on peut citer [42] :

- ❖ **Réutilisabilité:** Un réseau de neurones n'est pas programmé pour une application mais pour une classe de problèmes : après une phase d'apprentissage adéquate, il peut traiter de nombreuses tâches.
- ❖ **Robustesse:** Les couches cachées du réseau de neurone forment une représentation abstraite des données (concepts), qui permettent de savoir catégoriser des données non traitées lors de l'apprentissage (non prévues).
- ❖ **Parallélisme:** L'architecture des réseaux permet théoriquement à un grand nombre d'éléments de calcul simples de travailler d'une façon concurrente, ce qui facilite l'obtention des résultats très rapides et aide à l'implantation des applications ayant notamment des contraintes temps-réel.
- ❖ **Logique floue:** Les réseaux de neurone se sont inspirés du fonctionnement du cerveau humain, ils savent utiliser des notions imprécises, modélisent des systèmes dynamiques et non linéaires, le réseau établit lui-même ses connaissances, à partir d'exemples.

Bien que les réseaux de neurones soient capables d'effectuer beaucoup de tâches, ils souffrent néanmoins de certaines limites dont on peut citer [45], [46] :

- **Choix des attributs** : Pour permettre de travailler avec les réseaux de neurones, il est nécessaire de choisir soigneusement la représentation des données. Les attributs ne peuvent être que numériques.
- **Processus d'apprentissage** : Lorsque la durée d'apprentissage est très longue, la possibilité de perdre la capacité de généralisation par le RN augmente c.-à-d. apprentissage au détriment de la généralisation (Le problème du sur-apprentissage).
- **Architecture du réseau** : Le nombre de nœuds dans les couches d'entrée et de sortie sont généralement fixés par l'application, mais comment optimiser le nombre de niveaux cachés et le nombre de nœuds dans ces niveaux ? Il n'existe pas des règles claires dans ce sens.
- **Exploitabilité** : Il existe une grande difficulté pour expliquer les résultats obtenus par le réseau de neurones, car ce dernier fonctionne comme une boîte noire et peut découvrir des règles et l'exploitent pour résoudre des problèmes, mais il ne permet pas la possibilité d'extraire des lois ou des formules depuis ces règles.

IV .7. Application des RNAs

Depuis leur apparition, les réseaux de neurones ont été largement utilisés dans plusieurs domaines. On peut citer [47], [48]

- **Industrie**: contrôle de qualité, diagnostic de panne, analyse de signature ou d'écriture manuscrite, contrôle de procédés industriels.
- **Finance**: prévision et modélisation du marché (cours de monnaies...), sélection d'investissements, attribution de crédits
- **Télécommunications et informatique** : analyse du signal, élimination du bruit, reconnaissance de formes (bruits, images, paroles), compression de données...

- **Environnement:** évaluation des risques, analyse chimique, prévisions et modélisation météorologiques, gestion des ressources...
- **Médical:** Diagnostique automatisé des maladies, et traitement automatique des informations issues des imageries médicales...
- **Militaire:** Guidage des missiles, Drones et avions sans pilotes, ...

Le tableau 1.2, représente la correspondance entre chaque domaine d'application et le type de Réseau de Neurones Artificiels le plus approprié :

Domaine d'application	Type de RNA
Optimisation	HOPEFIELD, ART
Approximation de fonctions	MLP, RBF
Traitement d'images	HOPEFIELD
Classification	MLP, KOHONEN, RBF, ART

Tableau IV.4 : Correspondance type de RNA-Domaine d'application

L'approximation de fonctions par le RBF

Dans le réseau RBF, chaque unité dans la couche cachée a son propre centroïde, et pour chaque entrée $x = (x_1, \dots, x_r)$, le réseau calcule la distance entre x et son centroïde. Les sorties sont des fonctions linéaires de cette distance.

Ainsi chaque nœud dans le réseau RBF calcule une sortie qui dépend d'une fonction radiale symétrique.

Supposons qu'il y a "r" nœuds d'entrées et "N" nœuds de sortie. La fonction de réponse globale a la forme suivante :

$$S_i(x) = \sum_{j=1}^N w_{ij} k\left(\frac{x-c_j}{a_i}\right) = \sum_{j=1}^N w_{ij} g\left(\frac{\|x-c_j\|}{a_i}\right) \quad (IV.6)$$

$M \in \mathbb{N}$: ensemble de nombres naturels, est le nombre de nœuds dans la couche cachée; $w_i \in \mathbb{R}_N$: est le vecteur des poids du $i^{\text{éNe}}$ nœud vers les nœuds sortant ; x est un vecteur d'entrée (1'élément de \mathbb{R}) ; K est une fonction radialement symétrique d'une unité dans la couche cachée ; c_j et a_i sont le centroïde et le facteur d'activation. La fonction gaussienne est souvent utilisée comme fonction d'activation, et les facteurs de lissage peuvent être les mêmes ou varier à travers les nœuds.

Propriété d'approximation : [50]

Pour toute fonction réelle non linéaire $f(x)$ définie et continue sur un ensemble compact $\{c_i\} \subseteq \mathcal{E}$, il existe un système RBF $y(x)$ tel que :

$$\sup_s |f(x) - y(x)| < s \quad (\text{IV.7})$$

Avec : s est une constante positive.

Il a été montré que le réseau RBF est un approximateur universel, c'est à dire que le réseau est capable d'approximer n'importe quelle fonction avec une précision donnée pourvu que l'on fournisse un nombre suffisant de neurones, et que l'on utilise un algorithme d'apprentissage adéquat.

Ces réseaux approximent localement les fonctions. En effet, les neurones de la première couche ne produisent pas de valeurs significatives en sortie que dans une certaine zone de l'espace d'entrée (c'est à dire que la sortie est proche de 1 pour une entrée proche de leur prototype est proche de 0 ailleurs).

La seconde couche permet d'associer une certaine sortie à chaque zone. Les réseaux RBF sont particulièrement adaptés lorsque les vecteurs des entrées sont regroupés par zones, tel que dans le cas de l'algorithme des K-means. Dans cette situation, les RBF ont généralement de meilleures performances que les autres solutions neuronales, ou des performances équivalentes pour une complexité moindre.

Application d'une commande mode glissant- neuronale adaptative pour un bras manipulateur à 2DDL :

IV.9 .1. Énoncé du problème

Considérez la dynamique d'un manipulateur articulaire comme suite :

$$H(q) \ddot{q} + C(q, \dot{q}) \dot{q} + G(q) = \tau - F(\dot{q}) - \tau_d \quad (IV.8)$$

Où :

$H(q)$ est une matrice inertielle définie positive $n \times n$, $C(q, \dot{q})$ est un vecteur inertielle établie positive $n \times n$, $G(q)$ est un vecteur inertiel $n \times 1$, $F(\dot{q})$ est la force de frottement, v_d est la perturbation inconnue et τ est l'entrée de commande.

Désignons l'erreur de suivi comme suite :

$$e(t) = q_d(t) - q(t)$$

Sélectionner la variable glissante comme suite :

$$S = \dot{e} + Ae \quad (IV.9)$$

Où : A est une matrice constante symétrique positive définie et $A = A^t > 0$,

Donc on a :

$$\begin{aligned} \dot{q} &= -s + \dot{q}_d + Ae \\ Hs &= H(\ddot{q}_d - \ddot{q} + Ae) = H(\ddot{q}_d + Ae) \\ H\dot{q} & \\ &= H(\ddot{q}_d + Ae) + C\dot{q} + G + F + v_d - \tau \\ &= H(\ddot{q}_d + Ae) - Cs + C(\dot{q}_d + Ae) + G + F + v_d - \tau \\ &= -Cs - \tau + f + v_d \end{aligned} \quad (IV.10)$$

On a : $f(x) = H(\ddot{q}_d + Ae) + C(\dot{q}_d + Ae) + G + F$.

En ingénierie, $f(x)$ est inconnue et par conséquent, il est nécessaire d'approcher $f(x)$. L'entrée réseau est sélectionnée en fonction de l'expression de $f(x)$:

$$X = [e^T \quad \dot{e}^T \quad q_d^T \quad \dot{q}_d^T \quad \ddot{q}_d^T]$$

Le contrôleur est conçu comme : $\tau = \hat{f}(x) + k_v s$ (IV.11)

Où k_v est une matrice constante symétrique positive définie, $f^{\wedge}(x)$ est la sortie du réseau RBF. $f^{\wedge}(x)$ approche $f(x)$.

D'eqs : (IV.11) et(IV.10), nous avons :

$$\begin{aligned} Hs' &= -Cs - f^{\wedge}(x) - k_v s + f(x) + v_d \\ &= -(k_v + C) s + \tilde{f}(x) + v_d = -(k_v + C) s + \zeta_0 \end{aligned} \quad (IV.12)$$

Où : $\tilde{f}(x) = f(x) - f^{\wedge}(x)$, $\zeta_0 = \tilde{f}(x) + v_d$.

Sélectionner la fonction LYPUNOV comme :

$$L = \frac{1}{2} s^T H s$$

Donc,

$$\begin{aligned} L' &= s^T H s' + \frac{1}{2} s^T H' s = -s^T k_v s + \frac{1}{2} s^T (H' - 2C) s + s^T \zeta_0 \\ L' &= s^T \zeta_0 - s^T k_v s \end{aligned}$$

Il indique que, avec k_v , la stabilité du système de contrôle dépend de ζ_0 , c'est-à-dire que la précision d'approximation et magnitude de v_d .

Le réseau RBF peut être adopté à environ $f(x)$. l'algorithme de réseau RBF est :

$$\phi_i = g\left(\frac{\|x - c_i\|^2}{a_i^2}\right), \quad i=1,2,\dots,n$$

$$Y = w^{\square T} \phi(x), \quad f(x) = w^{\square T} \phi(x) + \mathcal{E}$$

Où x est l'état d'entrée du réseau, $\phi(x) = [\phi_1 \ \phi_2 \ \dots \ \phi_n]^T$, \mathcal{E} est l'erreur d'approximation du réseau neuronal. w^{\square} Est le vecteur de poids du réseau RBF souhaité.

IV.9 .2. Contrôle du mode siliding par rapport à l'approximation de $f(x)$

IV.9 .2.1.conception du contrôleur

RBF network est adopté pour approximer $f(x)$, par conséquent, la sortie du réseau RBF est :

$$\hat{f} = \hat{w}^T \phi(x) \quad (IV.13)$$

Sélectionner

$$\tilde{w} = w^* - \hat{w} \quad ||w^*||_F \leq w_{Nas}$$

Par conséquent, nous avons :

$$\zeta_0 = \tilde{f}(x) + v_d = \tilde{w}^T \mathbf{r}(x) + \mathcal{E} + v_d$$

Le contrôleur est conçu comme :

$$\tau = \hat{f}(x) + k_v s - v \quad (IV.14)$$

Où v est l'élément robuste requis pour surmonter l'erreur d'approximation de réseau \mathcal{E} et la perturbation v_d .

À partir des équations (IV.14) et (IV.10) nous avons :

$$Hs' = (k_v + C) s + \tilde{w}^T \phi(x) + (\mathcal{E} + v_d) + v = -(k_v + C) s + \zeta_l \quad (IV.15)$$

Où : $\zeta_l = \tilde{w}^T \phi(x) + (\mathcal{E} + v_d) + v$.

L'élément robuste v est conçu comme :

$$V = -(s_N + b_d) \text{sgn}(s) \quad (IV.16)$$

Où $||\mathcal{E}|| \leq s_N$, $|v_d| \leq b_d$.

.2.2. Analyse de stabilité

Sélectionnez la fonction LYPONOV comme

$$L = \frac{1}{2} s^T H s + \frac{1}{2} \text{tr}(\tilde{w}^T F_w^{-1} \tilde{w})$$

Où H et F_w sont des matrices positives .par conséquent, nous avons :

$$\dot{L} = s^T H s + \frac{1}{2} s^T H s + \text{tr}(\tilde{w}^T F_w^{-1} \dot{\tilde{w}})$$

À partir de l'équation (IV.15), nous avons :

$$\dot{L} = -s^T k_v s + \frac{1}{2} s^T (\dot{H} - 2C) s + \text{tr} \tilde{w}^T (F_w^T \tilde{w} r s^T) + s^T (s + v_d + v)$$

On sait que le manipulateur a la caractéristique de $s^T (\dot{H} - 2C) s = 0$.select

$\tilde{w} = -F_w r s^T$, c'est-à-dire que la règle adaptative du réseau est :

$$\dot{\hat{w}} = F_w r s^T \quad (\text{IV.17})$$

Donc :

$$\dot{L} = -s^T k_v s + s^T (s + v_d + v)$$

Car :

$$s^T (s + v_d + v) = s^T (s + v_d) + s^T v = s^T (s + v_d) - \|s\| (s_N + b_d) \leq 0$$

On a :

$$\dot{L} \leq 0$$

.1.Résultats et simulation

Les résultats de simulation sont présentés sous MATLEB /SIMULINK, où les simulations abordent les points suivants :

- Position articulaire des jonctions.
- Vitesse de poursuite des jonctions.
- Torque des jonctions.
- Signaux de la fonction $f(x)$ et leur fonction estimée.

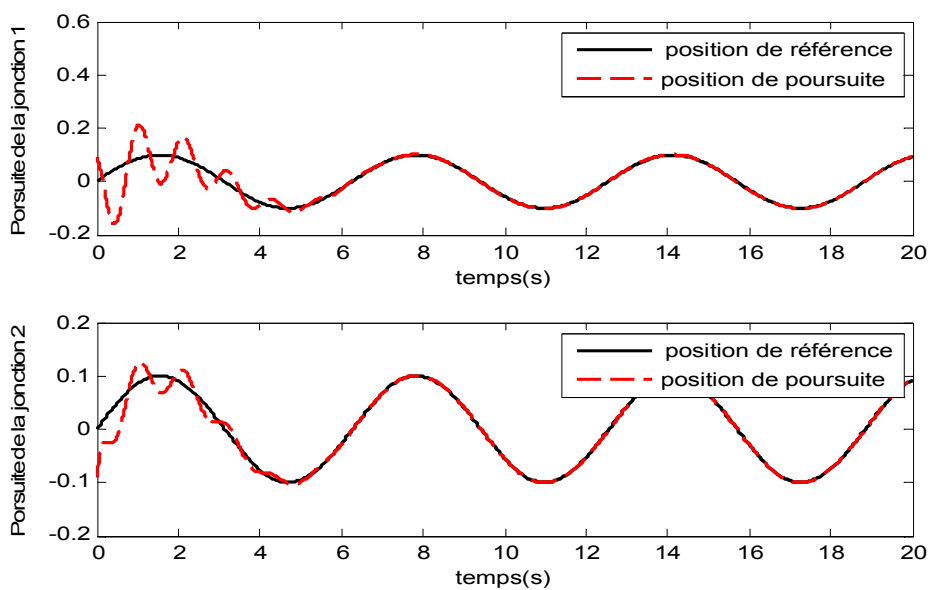


Figure IV .5. Position articulaire des jonctions 1 et 2.

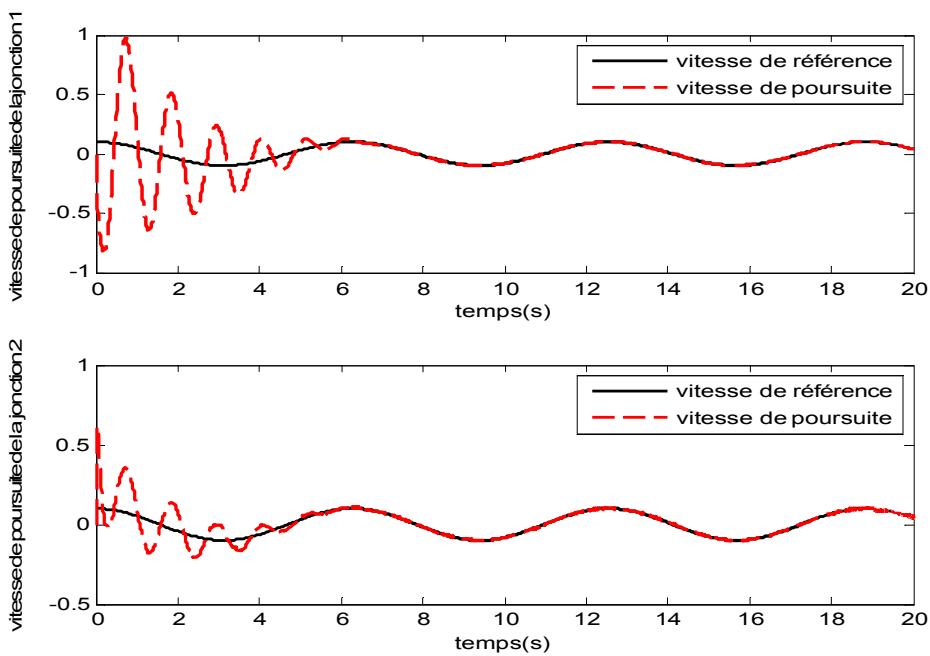


Figure IV. 6. Vitesse de poursuite des jonctions 1 et 2

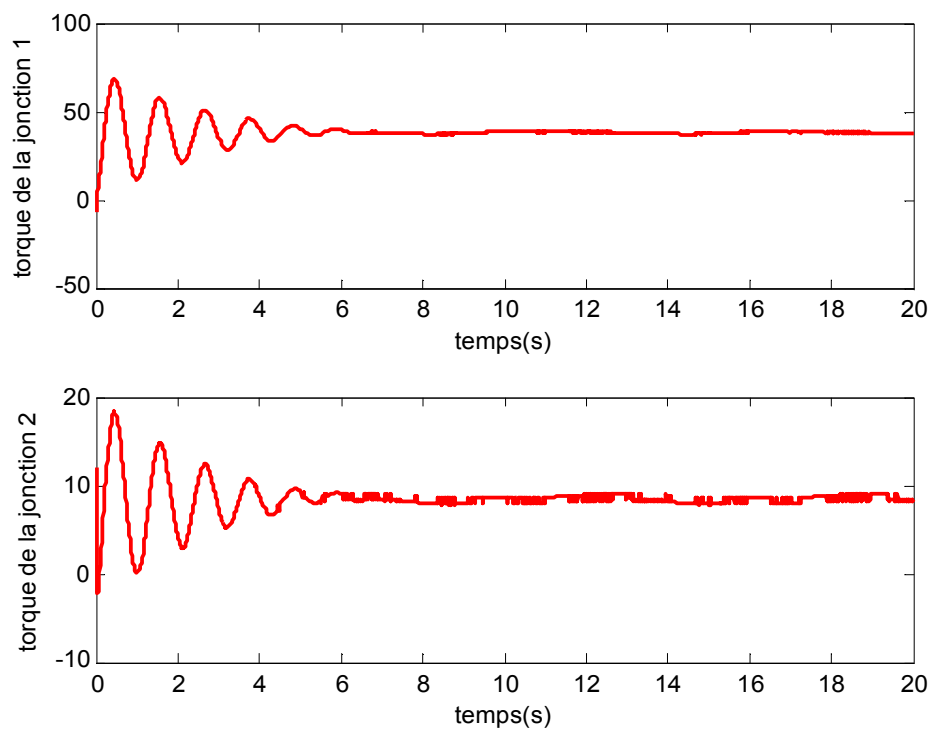


Figure IV .7. Torque des jonctions 1 et 2.

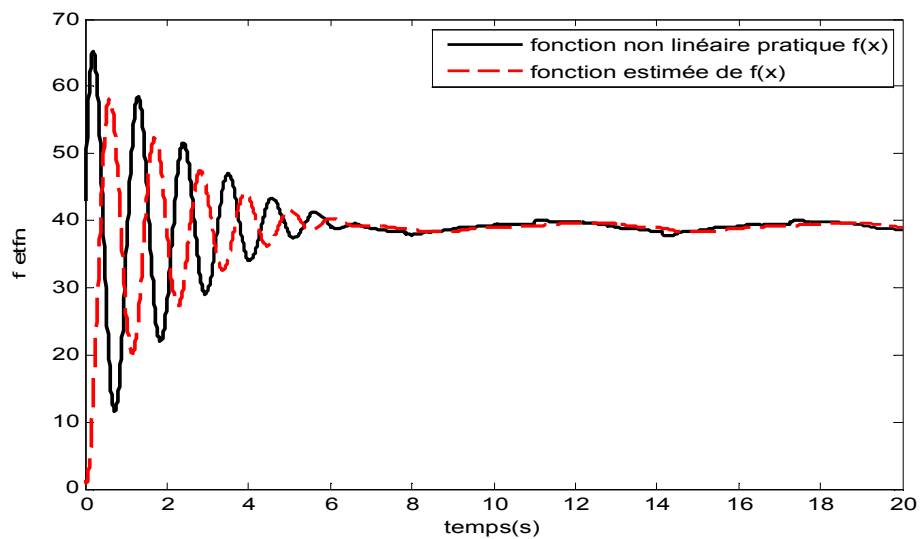


Figure IV .8. Signaux de la fonction $f(x)$ et leur fonction estimée

La figure IV .1 présente les positions de référence et la poursuite dont on remarque la convergence des positions articulaire vers les positions de référence en présence des perturbations qui peuvent être traduite par les performances et la robustesse de la commande proposée.

La figure IV .2 présente les vitesses articulaires de référence et les vitesses de poursuite dont on remarque que les vitesses articulaire reçoivent leurs vitesses de référence même en présence des perturbations.

La figure IV .3 montre les couples moteur (torques) qui convergent vers zéro une fois la convergence des positions est assurée.

La figure IV .4 présente la fonction non linéaire et la fonction estimée dont la fonction délivrée par les réseaux de neurones suit la fonction non linéaire définit le système proposée.

Conclusion

Dans ce chapitre une commande robuste basée sur l'approche mode glissant et une approche neuronale est présentée, dont les réseaux de neurones ont l'objectif d'approximer la fonction non linéaire du système qui est affectée par les perturbations injectées dans le système. L'approche mode glissant est introduite pour assurer une bonne convergence vers l'état d'équilibre. les résultats de simulation obtenus ont démontré l'efficacité de la commande proposée .le temps de convergence peut être amélioré avec une implantation des approches mode glissant récentes (mode glissant terminal rapide et mode glissant à ordre supérieur).

CONCLUSION GENERALE

Nous avons présenté dans ce travail modélisation des robots manipulateurs, où nous avons présenté les outils mathématiques utilisés dans la robotique pour décrire les trois méthodes de modélisation : la modélisation géométrique (directe et inverse), cinématique (direct et inverse), et dynamique.

Après, nous présentons la conception de la commande par mode glissement, les éléments fondamentaux du formalisme de la commande à structure variable. Tout d'abord, nous présenterons les différentes structures de contrôle par les modes glissants, après, nous expliquons brièvement le principe de cette commande, et le choix de la surface de glissement. Cette technique de commande robuste est une classe particulière à structure variable.

Ensuite, nous discutons brièvement des différentes techniques de commande des bras manipulateurs, nous présentons la commande décentralisée PID dans l'espace articulaire et dans l'espace cartésien, et la commande par mode glissant, pour tester l'efficacité de ces commandes, nous les simulons sur MATLAB.

En fin, nous présentons les réseaux de neurones à travers leurs priorités et nous proposons aussi les réseaux RBF (Radial Basis Fonctions), qui sont des approximateurs universels.

Les résultats des simulations obtenus montrent l'efficacité et la robustesse de la commande mode glissant neuronale qui nous a donné une bonne convergence des positions articulaires même en présence des perturbations. La commande par mode glissant conventionnelle présente des résultats très acceptables même en présence des perturbations mais l'inconvénient majeur de la commande qui est le phénomène de chatterring est apparu sur les commandes. La commande PID montre la limite de sa robustesse en cas de présence des perturbations.

Nous envisageons à travers ce travail à introduire une commande mode glissant terminal rapide avec une approximation neuronale afin qu'on puisse avoir une commande robuste face aux perturbations et une convergence rapide des positions articulaires vers leurs références.

REFERENCES BIBLIOGRAPHIQUES

- [1] Mekki Djamel, Commande par mode glissant flou d'un bras manipulateur, 2017.
- [2] Wissama & Dombre, Bases de la modélisation et de la commande des robots manipulateurs de type série, 2012.
- [3] WISSAMA Khalil & DOMBRE Etienne, Bases de la modélisation et des robots de la robots-manipulateurs de type série commande, p37.
- [4] Webographie.
- [5] Commande des mouvements d'un robot mobile par logique floue, 2014.
- [6] Canadian journal of civil engineering 26(3),293-304, 1999.
- [7] Khalil W., Dombre E., Modelisation, identification and control of robots, Hermes Penton Science, London,ISBN 1-90399-613-9, 2002, 480 p.
- [8] Wissama Khalil.Modeling, Identification and control of robots. 2004.
- [9] Abdallah Ghoul. Modélisation des robots manipulateurs19 février 2016.
- [10] Khalil, Dombre, Modélisation, identification et commande des robots, Hermès, 1999.
- [11] Khalil W., Kleinfinger J.-F., "A new geometric notation for open and closed-loop robots", Proc. IEEE Int. Conf.on Robotics and Automation, San Francisco, avril 1986, p. 1174-1180.
- [12] Paul R.C.P., Robot manipulators: mathematics, programming and control, MIT Press, Cambridge, 1981.
- [13] Pieper D.L., "The kinematics of manipulators under computer control", Ph. D. Thesis, Stanford University, 1968.
- [14] Raghavan M., Roth B., "Inverse kinematics of thegeneral 6R manipulator and related linkages", *Trans. of the ASME, J. of Mechanical Design*, Vol. 115, 1990, p. 502-508.
- [15] Dombre Etienne. Analyse et modélisation des robots manipulateurs. LAVOISIER, 2001.
- [16] Kerraci Abdelkader. Synthèse des commandes robustes des robots manipulateurs rigides. Thèse de magistère, Université d'Oran, 2004.
- [17] J.Craig John. Introduction to robotics, Mechanics and control. Pearson Education International, 2005.
- [18] Sciavicco Lorenzo and Siciliano Bruno. Modelling and Control of Robot Manipulators.Springer-Verlag London, 1999.
- [19] M.Murray Richard, Zexiang Li, S.Shankar Sastry, and S.Shankara Sastry. A Mathematical Introduction to Robotic Manipulation. CRC Press, 1994.
- [20] C.Edwards&S.KSpurgeon, "Slidingmodecontrol,theoriesandapplication,Ed.Taylorand Francis1998.
- [21] J.J.Slotine, 'Applied Non Linear Control', Prentice Hall,1991.

- [22] Z. Man, A.P. Paplinski, H. Wu. A robust MIMO terminal sliding mode control scheme for rigid robot manipulators [J]. IEEE Transactions on Automatic Control, 39:2464–2469, 1994.
- [23] K.Y. Zhuang, H.Y. Su, K.Q. Zhuang, J. Chu. Adaptive terminal sliding mode control for high order nonlinear dynamic systems [J]. Journal of Zhejiang University, 4:58–63, 2003.
- [24] Neila Mezghani Ben Romdhane and Tarak Damak. Terminal sliding mode feedback linearization control, International Journal of Sciences and Techniques of Automatic control & computer engineering, IJ-STA, Volume 4, N°1, pp.1174-1187, July 2010.
- [25] X. Yuan, J.-X. Xu (Eds.) Variable Structure Systems: Towards the 21st Century, LNCIS 274, pp.109–127, 2002. Springer-Verlag Berlin Heidelberg 2002.
- [26] Livre de Wisama KHALIL, Etienne DOMBRE, 08 mai 2012.
- [27] Samson C., "Problèmes en identification et commande de systèmes dynamiques", Thèse d'Etat, Rennes, France, 1983.
- [28] Arimoto S., Miyazaki F., "Stability and robustness of PID feedback control for robots manipulators of sensory capability", *The 1st Int. Symp. of Robotics Research*, MIT Press, Cambridge, 1984.
- [29] Korrami F., Özgüner U., "Decentralized control of robot manipulators via state and proportional-integral feedback", *Proc. IEEE Int. Conf. on Robotics and Automation*, Philadelphia, USA, April 1988, p. 1198-1203.
- [30] Kelly R., "A tuning procedure for stable PID control of robot manipulators", *Robotica*, Vol. 13, 1995, p. 141-148.
- [31] FILIPOV, Differential equations with discontinuous right-hand side. Amer. Math. soc trans. 1960.
- [32] SI AMMOUR Amar, « contribution à la commande par modes glissants d'ordre fractionnaire ». Thèse de Doctorat. Université Mouloud Mammeri Tizi-ouzou. 2011.
- [33] Commande par mode glissant flou d'un bras manipulateur, 2014.
- [34] EMELYANOV. LEVANTOVSKY. Drift algorithm in control of uncertain processes. 1986.
- [35] FILIPOV, Differential equations with discontinuous right-hand side. Amer. Math. soc trans. 1960.
- [36] R. Rosenblatt, Principles of Neurodynamics. Spartan Books. New-York, 1958.
- [37] L.P.J. Veelen, Analysis and Applications of Artificial Neural Networks, Book PrnticeHall, Edition, 1995.
- [38] K. Hornik: Approximation capabilities of multilayer feedforward networks. Neural networks, 4(2), p251-257. 1991.
- [39] V. Venugopal et W. Baets: Neural networks and statistical techniques in marketing research. Marketing intelligence and planning, vol. 12, no 7, p30-38. 1994.
- [40] K. Hornik, M. Stinchcombe et H. White: Multilayer Feed forward Networks are Universal Approximators. Neural Networks, vol. 2, p359-366, 1989.

- [41] D.S.Broomhead et D.Lowe: Multivariate function al interpolation and adaptative networks. Complex systems, 2, p321-355, 1988.
- [42] H.Abdiet D. Valentin, Modèles Neuronaux Connexion nistes et Numériques pour la Mémoire des Visages, It Psychologie Française 39(4), pp357-392, 1994.
- [43] Marc Parizeau "Reseau de Neurones", Automne 2004, presse de l'université de LAVAI. 2004.
- [44] L.P.J.Veelenturf, Analysis and Applications of Artificial Neural Networks, Book Prntice Hall, Edition, 1995.
- [45] M.T.Khadir, Les Réseaux de Neurones Artificiels, 2005.
- [46] R.A.Jacobs. "Increased rates of convergence through learning rates adaptation," Neural Networks, vol. 1, pp. 295-350, 1988.
- [47] K.Kara, "Application des réseaux de neurones à l'identification des systèmes non linéaires," Thèse de magistère en électronique, Université de Constantine, 1995.
- [48] L.Fausett, fundamentals of Neural Networks Architectures, Algorithms and Applications, Addison Wesley, Publishing Company, Inc, 1991.
- [49] J.Park and I.W.Sandberg: Universal approximation using radial basis function networks. Neural computation 3(2), 246-257. 1991.
- [50] V.N.Vapnik and A.Y.Cervonenkis: On the uniform convergence of relative frequencies of events to their probabilities. Theory of probability and its applications, 16(2), 264-280. 1971.