

République algérienne démocratique et populaire
Ministère de l'enseignement supérieur et de la recherche scientifique

Université de Batna
Faculté des Sciences de l'ingénieur
Département d'Electronique

Mémoire

En vue de l'obtention du diplôme de :

Magistère en électronique

Option : Contrôle industriel

Présenté par :

CHEBBI Smail

Ingénieur d'état en électronique

Thème

Etude des algorithmes de commande d'un robot manipulateur

Soutenu le : 11/11/2012

Devant le jury composé de :

Dr .AOUGHLLANET Said	M.C Univ.Batna	Président
Dr. MELAAB Djamel	M.C Univ.Batna	Rapporteur
Dr. KHIREDINE MD Salah	M.C Univ.Batna	Examineur
Dr. ATHAMENA Nouredine	M.C Univ.Batna	Examineur
Dr. DEBILOU Abderrazak	M.C Univ.Biskra	Examineur

Remerciements

Je tiens à remercier tout d'abord Dieu le tout puissant qui nous a donné durant toutes ces années la santé, le courage et la foi pour arriver à ce jour.

Je tiens à exprimer ma sincère reconnaissance à mon encadreur monsieur MELAAB Djamel, maître de conférences au département d'électronique à l'université de Batna pour sa confiance, son aide, son encouragement et ses conseils durant la réalisation de ce travail.

Je tiens à remercier vivement monsieur AOUGHLLANET Said, maître de conférences au département d'électronique de l'université de Batna, d'avoir accepté de présider le jury de ce mémoire.

Je suis très reconnaissant à monsieur ATHAMENA Noureddine, Maître de Conférences au département d'électronique de l'université de Batna, d'avoir accepté d'examiner mon mémoire avec attention et rigueur, et dont la participation à la soutenance en tant que examinateur m'honore beaucoup.

J'adresse mes vives remerciements à monsieur KHIREDDINE MD Salah maître de conférences à l'université de Batna, d'avoir accepté de participer dans ce jury et d'examiner ce travail.

Je suis très sensible au fait que monsieur DEBILOU Abderrazak, Maître de Conférences à l'université de Biskra, a pris la peine de se déplacer pour valoriser mon travail en temps que membre du jury en qualité d'examineur.

Je ne saurais oublier mes collègues et amis, avec lesquels j'ai partagé de très beaux moments, merci et bonne chance à tous.

Enfin, je remercie tous ceux qui ont contribué de près ou de loin à l'aboutissement de ce mémoire.

Sommaire

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

Chapitre I

Généralités sur la modélisation et la commande des robots manipulateurs

I.1 Introduction.....	3
I.2 Modélisation.....	4
I.2.1 Modèle géométrique.....	5
I.2.1.1 Modèle géométrique direct.....	5
I.2.1.2 Modèle géométrique inverse	8
I.2.2 Modèle cinématique.....	9
I.2.2.1 Modèle cinématique directe	9
I.2.2.2 Modèle cinématique inverse.....	10
I.2.3 Modèle dynamique.....	11
I.2.3.1 Modèle dynamique inverse.....	11
I.2.3.2 Modèle dynamique direct.....	14
I.2.3.3 Représentation d'états.....	14

I.3 Commande	14
I.3.1 Stabilité.....	15
1.3.1.1 Le point d'équilibre.....	15
1.3.1.2 Plan de phase.....	15
1.3.1.3 Définition mathématique précise de la stabilité.....	16
I.3.2 Méthode de Lyapunov.....	18
I.3.2.1 Méthode indirect de Lyapunov.....	18
I.3.2.2 Méthode directe de Lyapunov.....	18
I.3.3 Quelques stratégies de commande.....	20
I.3.3.1 Commande par contrôleur PID.....	20
I.3.3.2 Commande par linéarisation entrée-sortie.....	20
I.3.3.3 Commande non linéaire.....	21
I.3.3.4 Commande basé sur l'intelligence artificielle.....	21
I.4. Conclusion.....	21

Chapitre II

Commande par Mode glissant et par Backstepping

II.1 Introduction.....	22
II.2 Commande par mode glissant.....	22
II.2.1 Structures de base.....	23
II.2.1.1 Structure par commutation d'une contre réaction d'état.....	23

II.2.1.2	Structure par commutation au niveau de l'organe de commande.....	23
II.2.1.3	Structure par commutation au niveau de l'organe de commande, avec ajout de la commande équivalente.....	24
II.2.2	Principe de la commande par modes glissants.....	25
II.2.2.1	Définition du régime glissant.....	25
II.2.2.2	Choix de la Surface de glissement.....	26
II.2.3.3	Construction de la commande équivalente par la méthode Phillipov.....	27
II.2.3	Synthèse de la loi de commande.....	27
II.2.4	Robustesse des modes glissants.....	28
II.3.	Commande par backstepping.....	30
II.3.1	Conception de la loi de commande par backstepping.....	30
II.3.2	Backstepping et perturbation.....	34
II.4	Application sur le bras manipulateur.....	35
II.4.1	Description du manipulateur utilisé	35
II.4.1.1	Représentation d'états	36
II.4.2	Application du mode glissant.....	37
II.4.3	Application du backstepping.....	39
II.5	Simulation et résultats.....	41
II.5.1	Commande par mode glissant.....	41
II.5.2	Commande par backstepping.....	44
II.6.	Conclusion.....	48

Chapitre III

Commande par Mode glissant flou et flou adaptatif

III.1 Introduction.....	49
III.2 La logique floue.....	50
III.2.1 Systèmes et contrôleurs flous.....	50
III.2.1.1 Ensembles flous.....	50
III.2.1.2 Variable linguistique.....	52
III.2.1.3 le Contrôleur flou.....	53
III.3 Commande par mode glissant flou..	55
III.3.1 Conception du contrôleur(FSMC).....	56
III.3.1.1 Fuzzification.....	57
III.3.1.2 La base de règle.....	58
III.3.1.3 Inférence et défuzzification.....	59
III.3.2 Analyse de stabilité.....	60
III.4 Commande par mode glissant flou adaptatif.....	60
III.4.1 Conception du contrôleur(AFSMC).....	61
III.4.1.1 Fuzzification.....	61
III.4.1.2 La base de règles.....	63
III.4.1.3 Inférence et défuzzification.....	63
III.5 Simulation et résultats.....	64
III.5.1 Commande par mode glissant flou (FSMC).....	64

III.5.2 Commande par mode glissant flou adaptatif (AFSMC).....	68
III.6 Conclusion.....	71

Chapitre IV

Commande hybride Backstepping et Mode glissant

IV.1 Introduction.....	72
IV.2 Commande hybride backstepping -mode glissant.....	72
IV.2.1 Conception du contrôleur.....	73
IV.3 Présentation d'une commande hybride backstepping-mode glissant flou....	74
IV.3.1 Conception du contrôleur.....	75
IV.3.2 Analyse de stabilité.....	76
IV.4 Simulation et résultats.....	78
IV.4.1 Commande hybride backstepping-mode glissant	78
IV.4.2 Commande hybride backstepping-mode glissant flou	81
IV.5 Conclusion.....	85

Conclusion générale.....	86
--------------------------	----

Références

Annexe

Introduction générale

Les robots manipulateurs sont des systèmes fortement non linéaires, d'une dynamique qui peut être mal définis, ils ont des paramètres variables et sont soumis à des perturbations externes. La modélisation de ces systèmes relève très souvent d'une approximation des phénomènes physiques mis en jeu [16, 25]; Or, c'est à partir de cette représentation approximative que l'on souhaite construire une commande pour le système réel, cette commande doit être alors robuste dans le sens où elle devra assurer une faible sensibilité aux incertitudes des paramètres, à leurs variations et aux perturbations externes. L'objectif de ce travail consiste à étudier quelque commande robuste.

Lorsque le système est soumis à des fortes perturbations, les commande classique ne peuvent pas achever les performances voulues [15], on est souvent amené à chercher d'autre alternatives, or l'intelligence artificielle représente une approche intéressante qui a été le sujet de plusieurs recherches. En exploitant les capacités d'apprentissage, d'approximation et d'optimisation, l'intelligence artificielle offre des outils très puissants pour la commande, afin de concevoir des algorithmes bien meilleurs.

La logique floue est une technique de l'intelligence artificielle basée sur des théories mathématiques dont l'objectif principal est la modélisation des notions vagues et incertaines du langage naturel et à l'approximation des fonctions [13]. Ces propriétés font de la logique floue une approche inestimable pour résoudre les problèmes de la commande des systèmes incertains et perturbés.

Parmi les différents types des robots manipulateurs, dans ce mémoire on s'intéresse plutôt aux bras manipulateurs fixe à chaîne ouverte simple, en explorant les méthodes de commande qui peuvent assurer une certaine marge de robustesse vis-à-vis à la présence d'incertitudes et de bruits. Les méthodes qu'on va étudier dans ce travail se basent sur deux approches :

La commande adaptative backstepping qui est une procédure récursive qui utilise la théorie de Lyapunov dans la recherche de la loi de commande stabilisante et la commande par mode glissant qui est une commande robuste basée sur la théorie des structures variables.

Dans le but de chercher d'éventuelles améliorations apportées aux commandes par mode glissant et par backstepping, et pour explorer les potentiels de la logique floue, on va étudier des algorithmes hybrides qui ont pour fin le rassemblement des avantages de toutes ces techniques. On commence par une architecture à mode glissant flou, et flou adaptatif, ensuite, on passe à la combinaison du mode glissant et du backstepping et on finit par une commande backstepping-mode glissant flou.

L'étude de ces méthodes va être validée en l'appliquant sur un bras manipulateur à deux degrés de liberté de modèle incertain et perturbé pour résoudre le problème de la commande de position.

Ce travail s'organise en quatre chapitres :

Le chapitre I, présente des généralités sur la modélisation et la commande des robots manipulateurs, en présentant quelque notion de base.

Le chapitre II, concerne l'étude de deux méthodes de commande; le mode glissant et le backstepping, en montrant leurs caractéristiques.

Le chapitre III, traite l'utilisation de la « logique floue » pour améliorer les performances de la commande par mode glissant.

Le chapitre IV, présente une autre alternative qui va combiner quelque stratégie de commande évoquée dans les chapitres précédents, voir le backstepping, le mode glissant et la logique floue.

Chapitre I

Généralités sur la modélisation et la commande des robots manipulateurs

I.1 Introduction :

Actuellement les robots représentent un outil indispensable pour l'être humain et un grand soutien dans sa vie quotidienne, car ils font d'énormes tâches, que même l'homme ne peut pas les faire. Les robots manipulateurs présentent une partie importante des différents types de robots et leur évolution n'a pas cessé d'augmenter. Ils ont permis d'exécuter des tâches plus complexes et précises. La plus grande partie de ces robots manipulateurs est constituée de manipulateurs sériels [11] dont la morphologie peut facilement être comparée à celle de l'être humain.

Pour pouvoir exploiter les capacités de ces robots, il faut élaborer des commandes qui peuvent assurer les performances voulues et pour atteindre cet objectif, il est indispensable de connaître leur mécanisme. Ainsi plusieurs niveaux de modélisation sont possibles et dépendent des spécifications du cahier des charges de l'application envisagée : modèles géométriques, différentiels ou dynamiques.

Dans ce chapitre on présente un rappel sur la modélisation suivi d'une introduction sur la commande.

I.2 Modélisation :

Un manipulateur est constitué de deux sous-ensembles distincts : un (ou plusieurs) organe terminal et une structure mécanique articulée [25]. L'organe terminal est le dispositif d'interaction, fixé à l'extrémité mobile de la structure mécanique. Il est désigné par différentes appellations : préhenseur, outil, effecteur, organe terminal (OT) ou pince lorsque il s'agit d'une pince. La structure cinématique articulée est une chaîne cinématique de corps, généralement rigides, assemblées par des articulations. Le rôle de la structure mécanique articulée est d'amener l'organe terminal à un emplacement de position et d'orientation donnée, selon des requêtes spécifique de vitesse et d'accélération.

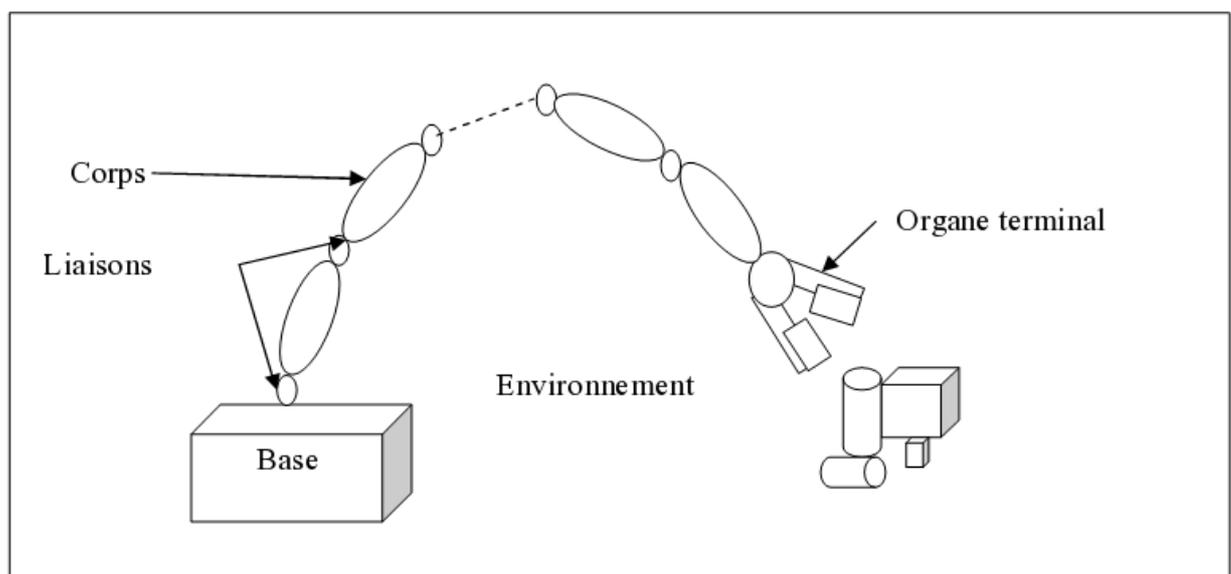


Figure I.1 : Robot à chaîne ouverte simple

❖ Le degré de liberté [25] :

L'un des caractéristiques essentielles du bras manipulateur est le degré de liberté, qui représente le nombre de variables de positions (articulation) indépendantes. Le nombre de d'articulation détermine le degré-de-liberté (ddl) du manipulateur. Pour pouvoir couvrir l'espace du travail le manipulateur devra posséder au moins six ddl, 3 pour le positionnement et 3 pour l'orientation. Avec moins de six ddl le manipulateur ne peut pas effectuée toute les positions et orientations nécessaires dans son espace de travail. La difficulté de contrôler un manipulateur augmente rapidement avec le nombre d'articulations. Un manipulateur ayant plus de six ddl (six articulations) est désigné sous le nom d'un manipulateur cinématiquement redondant.

❖ Espace articulaire et opérationnel [25] :

1-On appelle espace de configuration articulaire d'un robot manipulateur l'état du robot représentant la situation de ses différents corps. Les variables ou coordonnées articulaires sont aussi appelées coordonnées généralisées.

2-L'espace opérationnel est celui dans lequel est représenté la situation de l'organe terminal. Les variables ou coordonnées au nombre minimum permettant de définir la situation de l'OT sont appelées les coordonnées opérationnelles.

I.2.1 Modèle géométrique :

I.2.1.1 Modèle géométrique direct :

Le modèle géométrique direct (MGD) est un modèle mécanique utilisé pour déterminer la configuration (position, orientation) de l'outil terminal du robot en fonction de la configuration de ses liaisons, il est représenté par l'équation [25]:

$$x = f(q) \quad (I.1)$$

❖ Convention de Denavit-Hartenberg [25, 29] :

Pour décrire la géométrie des robots manipulateur on utilise généralement la convention de Denavit-Hartenberg pour la sélection des formes de référence. Dans cette convention, chaque transformation homogène T_i^{i-1} qui lie entre l'espace opérationnel et articulaire est représentée comme produit de quatre transformations de base :

$$T_i^{i-1} = Trans_{z_{i-1}, d_i} Rot_{z_{i-1}, \theta_i} Trans_{x_{i-1}, \alpha_i} R_{x_{i-1}, \alpha_i} \quad (I.2)$$

$$= \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & d_i \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} \cos\theta_i & -\sin\theta_i & 0 & 0 \\ \sin\theta_i & \cos\theta_i & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 1 & 0 & 0 & a_i \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & \cos\alpha_i & -\sin\alpha_i & 0 \\ 0 & \sin\alpha_i & \cos\alpha_i & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (I.3)$$

$$= \begin{bmatrix} \cos\theta_i & -\sin\theta_i \cos\alpha_i & \sin\theta_i \sin\alpha_i & a_i \cos\theta_i \\ \sin\theta_i & \cos\theta_i \cos\alpha_i & -\cos\theta_i \sin\alpha_i & a_i \sin\theta_i \\ 0 & \sin\alpha_i & \cos\alpha_i & d_i \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (I.4)$$

$$= \begin{bmatrix} R_i^{i-1} & o_i^{i-1} \\ 0 & 1 \end{bmatrix} \quad (I.5)$$

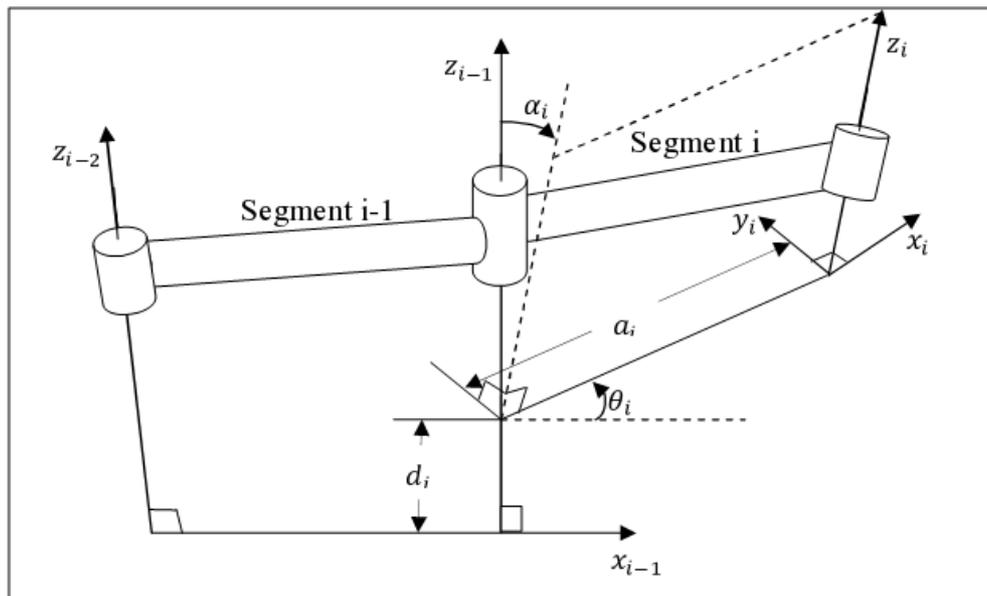


Figure I.2 : Paramètres de DH pour la liaison i et le segment i

Les paramètres de Denavit-Hartenberg sont alors :

- d_i : représente une translation le long de l'axe z_{i-1} entre les axes x_{i-1} et x_i .
- θ_i : représente l'angle de rotation autour de l'axe z_{i-1} entre les axes x_{i-1} et x_i .
- a_i : représente la distance minimale entre les axes z_{i-1} et z_i mesurer sur x_i .
- α_i : représente l'angle de rotation autour de l'axe x_i entre les axes z_{i-1} et z_i .

La matrice R_i^{i-1} 3x3 formée des trois premières lignes et de trois premières colonnes représente l'orientation du repère R_{i-1} par rapport au repère R_i et le vecteur o_i^{i-1} constitué des trois premiers coefficients de la dernière colonne représente la position de l'origine O_{i-1} du repère R_{i-1} par rapport à l'origine O_i du repère R_i .

Exemple [29] :

C'est on prend comme exemple un manipulateur à deux degrés de liberté on aura pour la convention Denavit-Hartenberg les paramètres suivantes :

N° du repère	a_i	α_i	d_i	θ_i
1	l_1	0	0	θ_1
2	l_2	0	0	θ_2

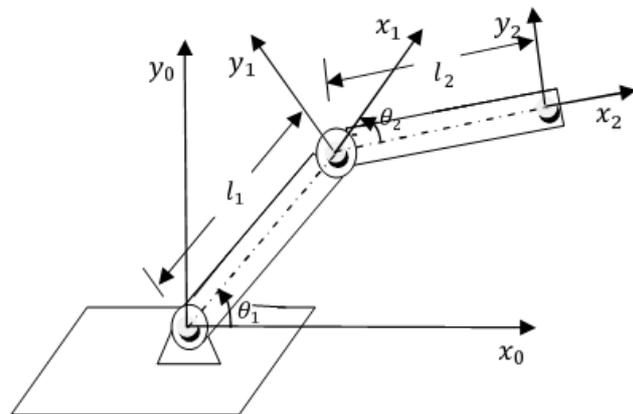


Figure I.3 : Paramètres de DH pour un bras à 2 ddl

On aura les transformées suivantes :

$$T_1^0 = \begin{bmatrix} \cos\theta_1 & -\sin\theta_1 & 0 & l_1\cos\theta_1 \\ \sin\theta_1 & \cos\theta_1 & 0 & l_1\sin\theta_1 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (I.6)$$

$$T_2^1 = \begin{bmatrix} \cos\theta_2 & -\sin\theta_2 & 0 & l_2\cos\theta_2 \\ \sin\theta_2 & \cos\theta_2 & 0 & l_2\sin\theta_2 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (I.7)$$

$$T_2^0 = T_1^0 T_2^1 = \begin{bmatrix} \cos(\theta_1 + \theta_2) & -\sin(\theta_1 + \theta_2) & 0 & l_1\cos\theta_1 + l_2\cos(\theta_1 + \theta_2) \\ \sin(\theta_1 + \theta_2) & \cos(\theta_1 + \theta_2) & 0 & l_1\sin\theta_1 + l_2\sin(\theta_1 + \theta_2) \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (I.8)$$

Ainsi les coordonnées du point de l'extrémité par rapport au repère lié à la base sont :

$$\begin{bmatrix} x \\ y \end{bmatrix} = \begin{bmatrix} l_1\cos\theta_1 + l_2\cos(\theta_1 + \theta_2) \\ l_1\sin\theta_1 + l_2\sin(\theta_1 + \theta_2) \end{bmatrix} \quad (I.9)$$

I.2.1.2 Modèle géométrique inverse :

Le modèle géométrique direct permet le passage depuis les coordonnées articulaires du robot à la position de l'organe terminal de celui-ci. C'est plutôt l'opération inverse qui est recherchée lors de la commande d'un bras manipulateur, c'est-à-dire disposant de la position où devrait être ramené l'organe terminal, quelles sont les coordonnées articulaires correspondant. La réponse à cette question est donnée par le modèle géométrique inverse.

A partir des coordonnées opérationnelles définies dans l'espace de la tâche, on utilise le MGI pour définir les coordonnées articulaires. Le modèle géométrique inverse est défini par la relation :

$$q = f^{-1}(x) \quad (I.10)$$

Après la détermination des coordonnées de la cible dans le repère de base on calcule les valeurs des variables généralisées $q_1 = \theta_1, q_2 = \theta_2$ qui amènent O_i au point visé. Il faut inverser les équations représentées par le système (I.1).

Le positionnement de OT peut rencontrer trois types de problèmes [11, 16].

❖ Problème géométrique

Si l'objet ne se situe pas dans le volume atteignable par le robot, il est évident que le système d'équations (I.1) est sans solution.

❖ Problème mécanique

Pour des raisons liées à la construction mécanique du robot les variables articulaires ne peuvent pas effectuer des rotations de 360° , donc le volume atteignable par le robot sera ainsi réduit.

❖ Problème mathématique

Pour une position O_i placée dans le champ atteignable par le manipulateur, il existe plusieurs valeurs des variables généralisées q_1, q_2 qui remplissent les équations du système (I.9) (différentes configurations pour une seule position). Seule et une seule configuration est à retenir.

Si on prend comme exemple une position quelconque pour le système (I.9), dans le champ qui définit les positions de placement, on va trouver deux configurations différentes permettant d'accéder à cette position (Figure I.3).

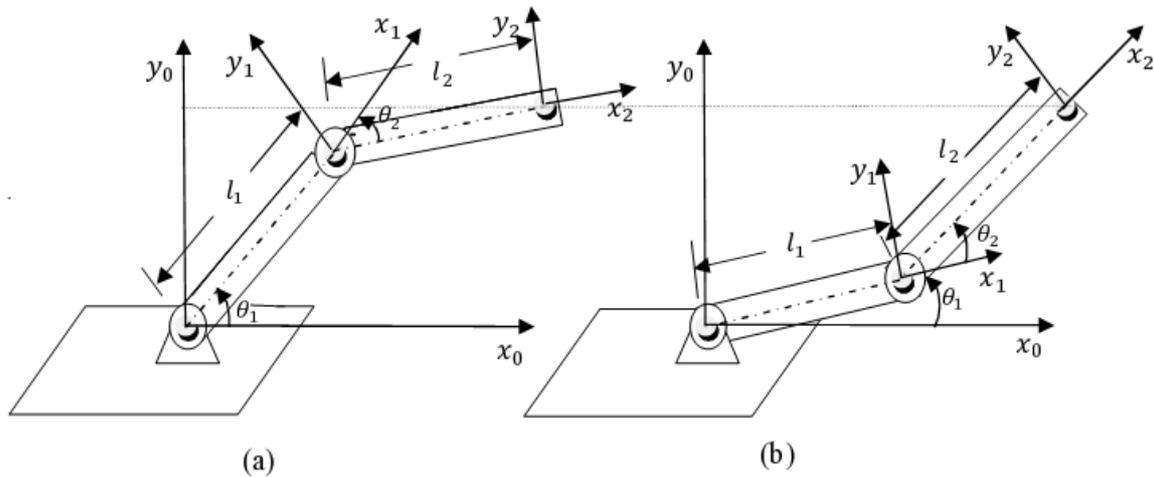


Figure I.4 : Deux configurations articulaires différentes pour une même position de l'OT

I.2.2 Modèle cinématique :

I.2.2.1 Modèle cinématique directe :

Le modèle cinématique directe exprime la relation entre les vitesses opérationnelles \dot{X} et les vitesses généralisées \dot{q} du bras manipulateur :

$$\dot{X} = J(q)\dot{q} \quad (I.11)$$

En général, on prend la notation [29] :

$$\dot{X} = [\dot{X}_n \ \dot{Y}_n \ \dot{Z}_n \ \omega_{X_n}^0 \ \omega_{Y_n}^0 \ \omega_{Z_n}^0] \quad (I.12)$$

$$= \begin{bmatrix} v_n^0 \\ \omega_n^0 \end{bmatrix} = \begin{bmatrix} J_D \\ J_R \end{bmatrix} \dot{q} \quad (I.13)$$

Où :

$-v_n^0$ et ω_n^0 sont la vitesse linéaire et angulaire de O_n dans le mouvement de R_n par rapport à R_0

$-J(q)$ est la matrice Jacobienne de la fonction $f(q)$ définie dans le repère (X, Y, Z).

$-J_D$ et J_R sont la matrice Jacobienne de déplacement et de rotation.

Pour un manipulateur à n ddl on aura :

$$T_n^0 = \begin{bmatrix} R_n^0 & o_n^0 \\ 0 & 1 \end{bmatrix} \quad (I.14)$$

$$J_D = \begin{bmatrix} \frac{\partial o_{Xn}^0}{\partial q_1} & \frac{\partial o_{Xn}^0}{\partial q_2} & \dots & \frac{\partial o_{Xn}^0}{\partial q_n} \\ \frac{\partial o_{Yn}^0}{\partial q_1} & \frac{\partial o_{Yn}^0}{\partial q_2} & \dots & \frac{\partial o_{Yn}^0}{\partial q_n} \\ \frac{\partial o_{Zn}^0}{\partial q_1} & \frac{\partial o_{Zn}^0}{\partial q_2} & \dots & \frac{\partial o_{Zn}^0}{\partial q_n} \end{bmatrix} \quad (I.15)$$

$$J_R = \begin{bmatrix} \frac{\partial \omega_{Xn}^0}{\partial q_1} & \frac{\partial \omega_{Xn}^0}{\partial q_2} & \dots & \frac{\partial \omega_{Xn}^0}{\partial q_n} \\ \frac{\partial \omega_{Yn}^0}{\partial q_1} & \frac{\partial \omega_{Yn}^0}{\partial q_2} & \dots & \frac{\partial \omega_{Yn}^0}{\partial q_n} \\ \frac{\partial \omega_{Zn}^0}{\partial q_1} & \frac{\partial \omega_{Zn}^0}{\partial q_2} & \dots & \frac{\partial \omega_{Zn}^0}{\partial q_n} \end{bmatrix} \quad (I.16)$$

Tel que :

$$\omega_n^0 = \dot{R}_n^0 R_n^{0T} \quad (I.17)$$

Donc on peut obtenir le modèle cinématique par la dérivation du modèle géométrique directe.

La matrice Jacobéenne peut être obtenue aussi en appliquant la loi de composition des vitesses [25, 29].

I.2.2.2 Modèle cinématique inverse :

Le modèle cinématique inverse permet de déterminer la vitesse des variables articulaires en fonction de la vitesse des variables opérationnelles. Pour les manipulateurs non redondants ($n=m$), le modèle s'écrit [25, 29] :

$$\dot{q} = J^{-1}(q)\dot{X} \quad (I.18)$$

I.2.3 Modèle dynamique [25, 29] :

I.2.3.1 Modèle dynamique inverse :

Le modèle dynamique inverse d'un robot sert à répondre à la question : quelle sont les couples et/ou forces nécessaires pour suivre une certaine trajectoire voulue. Ainsi le modèle dynamique inverse exprime la fonction qui lie le vecteur τ des forces et couples articulaires exercer sur les actionneurs avec la position et l'orientation de l'effecteur, autrement dit avec l'espace des coordonnées généralisées en termes de positions $q(t)$, vitesses $\dot{q}(t)$ et accélérations $\ddot{q}(t)$ articulaires.

Pour établir le modèle dynamique inverse d'un robot manipulateur, on utilise le plus souvent les équations d'Euler-Lagrange pour le définir comme :

$$\frac{d}{dt} \frac{\partial L}{\partial \dot{q}_i} - \frac{\partial L}{\partial q_i} = \tau_i \quad (I.19)$$

Avec :

- $L = K - V$ qui est appelé Lagrangien du système, K est l'énergie cinétique du système et V est l'énergie potentielle du système.
- q_i représente la i ème coordonnée généralisée du système.
- τ_i est la force généralisée appliquée à l' i ème élément du système. τ_i est un couple si l'articulation est rotoïde ou une force si l'articulation est prismatique.

L'énergie cinétique est donnée par :

$$K = \sum_{i=0}^n \frac{1}{2} m_i v_i^T v_i + \frac{1}{2} \omega_i^T I_i \omega_i \quad (I.20)$$

Avec :

- I_i : Matrice d'inertie du corps i .
- m_i : Masse du corps i .
- v_i : vitesse linéaire du centre de gravité du corps i .
- ω_i : vitesse angulaire du corps i .

Dès équations (I.11-I.17) l'énergie cinétique du système peut s'écrire sous la forme :

$$K = \frac{1}{2} \dot{q}^T \sum_{i=1}^n [m_i J_{Di}^T(q) J_{vi}(q) + J_{\omega i}^T(q) R_{0,i} I_i R_{0,i}^T J_{Ri}(q)] \dot{q} \quad (I.21)$$

La vitesse de rotation est exprimée dans le même repère que celui qui a servi à la détermination de la matrice d'inertie I_i .

L'équation (I.21) peut s'écrire sous forme :

$$K = \frac{1}{2} \dot{q}^T M(q) \dot{q} \quad (I.22)$$

L'énergie potentielle est donnée par :

$$V = g^T \sum_{i=1}^n O_{gi} m_i \quad (I.23)$$

Avec :

g : Vecteur de gravité

O_{gi} : Coordonnées de centre de gravité du corps i dans le repère de base.

On peut déduire les équations d'Euler-Lagrange en définissant le lagrangien comme suit :

$$L = K - V = \frac{1}{2} \sum_{i,j} d_{i,j}(q) \dot{q}_i \dot{q}_j - V(q) \quad (I.24)$$

On a :

$$\frac{\partial L}{\partial \dot{q}_k} = \sum_j d_{k,j}(q) \dot{q}_j \quad (I.25)$$

$$\frac{d}{dt} \frac{\partial L}{\partial \dot{q}_k} = \sum_j d_{k,j}(q) \ddot{q}_j + \sum_j \frac{d}{dt} d_{k,j}(q) \dot{q}_j \quad (I.26)$$

$$\frac{d}{dt} \frac{\partial L}{\partial \dot{q}_k} = \sum_j d_{k,j}(q) \ddot{q}_j + \sum_{i,j} \frac{\partial d_{k,j}}{\partial q_i} \dot{q}_i \dot{q}_j \quad (I.27)$$

De même :

$$\frac{\partial L}{\partial q_k} = \frac{1}{2} \sum_{i,j} \frac{\partial d_{i,j}}{\partial q_k} \dot{q}_i \dot{q}_j - \frac{\partial V}{\partial q_k} \quad (I.28)$$

Ainsi, les équations d'Euler-Lagrange deviennent :

$$\sum_j d_{k,j}(q) \ddot{q}_j + \sum_{i,j} \left[\frac{\partial d_{k,j}}{\partial q_i} - \frac{1}{2} \frac{\partial d_{i,j}}{\partial q_k} \right] \dot{q}_i \dot{q}_j + \frac{\partial V}{\partial q_k} = \tau_k \quad (I.29)$$

En utilisant le fait que :

$$\sum_{i,j} \frac{\partial d_{k,j}}{\partial q_i} \dot{q}_i \dot{q}_j = \frac{1}{2} \sum_{i,j} \left[\frac{\partial d_{k,j}}{\partial q_i} + \frac{\partial d_{k,i}}{\partial q_j} \right] \dot{q}_i \dot{q}_j \quad (I.30)$$

On obtient :

$$\sum_{i,j} \left[\frac{\partial d_{k,j}}{\partial q_i} - \frac{1}{2} \frac{\partial d_{i,j}}{\partial q_k} \right] \dot{q}_i \dot{q}_j = \sum_{i,j} \frac{1}{2} \left[\frac{\partial d_{k,j}}{\partial q_i} + \frac{\partial d_{k,i}}{\partial q_j} - \frac{\partial d_{i,j}}{\partial q_k} \right] \dot{q}_i \dot{q}_j \quad (I.31)$$

On note :

$$c_{i,j,k} = \frac{1}{2} \left[\frac{\partial d_{k,j}}{\partial q_i} + \frac{\partial d_{k,i}}{\partial q_j} - \frac{\partial d_{i,j}}{\partial q_k} \right] \quad (I.32)$$

Et

$$\phi_k = \frac{\partial V}{\partial q_k} \quad (I.33)$$

Pour k fixe on a :

$$c_{i,j,k} = c_{j,i,k} \quad (I.34)$$

Finalement les équations d'Euler-Lagrange deviennent :

$$\sum_j d_{k,j}(q) \ddot{q}_j + \sum_{i,j} c_{i,j,k}(q) \dot{q}_i \dot{q}_j + \phi_k(q) = \tau_k \quad (I.35)$$

Ce qui peut s'écrire sous forme matricielle :

$$M(q) \ddot{q} + C(q, \dot{q}) \dot{q} + G(q) = \tau \quad (I.36)$$

Où :

$M(q)$: Matrice d'inertie du manipulateur.

$C(q, \dot{q})$: Vecteur de Coriolis et des couples centrifuges.

$G(q)$: Vecteur de gravité.

Dans $C(q, \dot{q})$ les termes impliquant un produit q_i^2 sont appelés centrifuges. Et ceux impliquant un produit $q_i q_j$ avec $i \neq j$ sont les termes de Coriolis.

I.2.3.2 Modèle dynamique direct :

Le modèle dynamique direct exprime la position, la vitesse et l'accélération des articulations en fonction du couple appliqué. Il est obtenu par inversion du modèle précédant, il s'écrit :

$$\ddot{q} = M(q)^{-1}[\tau - C(q, \dot{q})\dot{q} - G(q)] \quad (I.37)$$

I.2.3.3 Représentation d'états :

Pour obtenir la représentation d'état, on fait le changement de variables suivants :

On pose $x_1 = q$, $x_2 = \dot{q}$ on aura le système :

$$\begin{cases} \dot{x}_1 = x_2 \\ \dot{x}_2 = M(q)^{-1}[\tau - C(q, \dot{q})\dot{q} - G(q)] \end{cases} \quad (I.38)$$

I.3 Commande :

La commande des systèmes non linéaires constitue un domaine de recherche très actif. La diversité des comportements des phénomènes dynamiques qui peuvent être observés pour cette classe de systèmes explique la grande variété d'approches développées. Confronté à un tel défi, il est clair que nous ne pouvons pas nous attendre à ce qu'une procédure particulière s'applique à tous les systèmes non linéaires [15]. Il est également peu probable que la conception d'un contrôleur puisse être basée sur un outil particulier. Ce dont l'automaticien a besoin est un ensemble d'outils d'analyse et de procédures qui couvrent une large gamme de situations. Or pour une application particulière, il faut employer les outils qui sont les plus appropriés pour le problème à disposition.

Dans cette partie on va évoquer le problème de la stabilité en présentant un outil puissant pour son analyse, ensuite on va citer quelque stratégie de commande utilisée pour les bras manipulateurs.

I.3.1 Stabilité :

La notion de stabilité correspond à l'idée d'un comportement qui dure dans le temps et permet de formaliser la question suivante : supposons que l'on initialise le système dynamique $\dot{x} = f(x)$ en un point voisin d'un point d'équilibre x_0 , que devient-il de la trajectoire solution ?

Cette question est d'importance car dans la pratique les conditions initiales présentent des incertitudes ; il serait souhaitable que deux conditions initiales voisines conduisent à des trajectoires solutions voisines pour tout temps et ceci même pour des temps infiniment longs. Une manière naturelle d'aborder cette question consisterait à résoudre l'équation différentielle et à examiner le comportement des solutions [15,9]. Cependant, en général, on ne sait pas résoudre les équations différentielles.

La réponse à la question nécessite donc une description qualitative des trajectoires du système. C'est le mathématicien russe Lyapounov qui a établi en 1892, dans son mémoire intitulé "Problème général de la stabilité du mouvement" les fondements de la théorie moderne de la stabilité [9]. Les démonstrations utilisent des fonctions auxiliaires appelées aujourd'hui fonction de Lyapounov.

1.3.1.1 Le point d'équilibre :

L'état x_e est appelé état d'équilibre ou point d'équilibre pour le système (I.38) si lorsque $x(t_0) = x_e$ alors $x(t) = x_e$ pour tout $t \geq t_0$. En d'autres termes, x_e vérifie l'équation $f(x_e) = 0$.

On peut toujours se ramener au cas où le point d'équilibre est l'origine 0 puisque si x_e vérifie $f(x_e) = 0$, il suffit de considérer le changement de coordonnées $z = x - x_e$, la dérivée de z est donnée par $\dot{z} = \dot{x} = f(x) = f(z + x_e)$, par définition $= g(z)$ et $g(0) = 0$. L'origine est bien un point d'équilibre du système $\dot{z} = g(z)$. [15]

1.3.1.2 Plan de phase [15]:

La représentation dans le plan de phase d'un système est une méthode qui exprime l'ensemble des courbes qui définissent l'évolution de l'état du système dans le temps (Figure I.4). Quoique cette représentation doit toute fois passer par la résolution de l'équation

différentielle, ce qui n'est pas toujours évident. Cette méthode, connue depuis longtemps, connaît un regain d'intérêt lié aux performances des calculateurs actuels qui rendent possible le calcul des trajectoires solutions par intégration numérique.

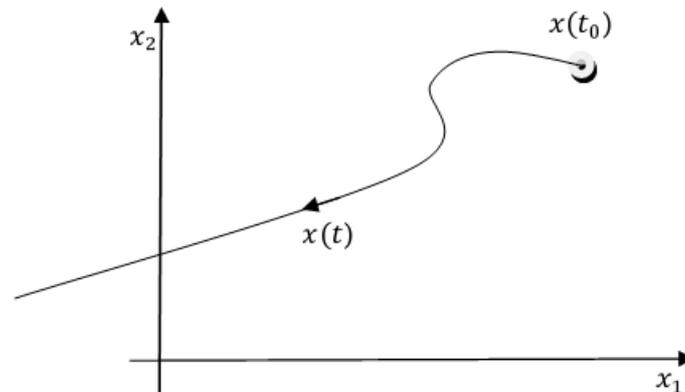


Figure I.5 : Trajectoire d'un système dans le plan de phase

I.3.1.3 Définition mathématique précise de la stabilité [15,26]:

Définition 1 : Le point d'équilibre x_e est dit stable si $\forall \rho > 0$, il existe $r(\rho) > 0$ tel que si $\|x_0 - x_e\| \leq r$ alors $\|x(t) - x_e\| \leq \rho$, sinon le point d'équilibre est dit instable.

Cette définition signifie que, quelle que soit la boule d'exigence de rayon ρ , il est toujours possible de choisir une certaine sous-boule de rayon r telle que pour toute condition initiale comprise dans cette sous-boule, la trajectoire résultante sera, en tout temps, comprise dans la boule d'exigence de rayon ρ . Dans un langage plus imagé, un petit déséquilibre initial n'entraîne qu'un petit déséquilibre au cours du temps, déséquilibre qui peut très bien être permanent.

Il existe plusieurs notions de stabilité :

- Stabilité asymptotique.
- Stabilité exponentielle.
- Stabilité locale et globale.

Définition 2 : (Stabilité asymptotique) Le point d'équilibre $x_e = 0$ est asymptotiquement stable s'il est stable et attractif, i.e., s'il existe $r > 0$ tel que $\forall x_0 \in B(x_e, r)$, $\lim_{t \rightarrow \infty} x(t) = 0$.

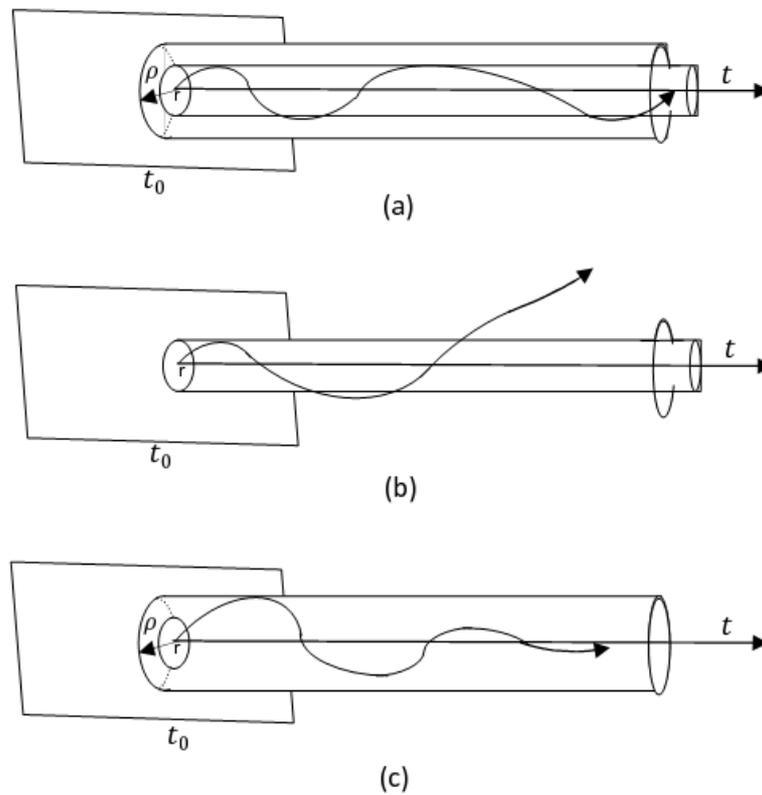


Figure I.6: (a)Stabilité, (b), Instabilité, (c)Stabilité asymptotique

Définition 3 : (Stabilité exponentielle) Le point d'équilibre $x_e = 0$ est exponentiellement stable s'il existe deux scalaires strictement positifs k et α tels que

$$\forall t \geq 0, \|x(t)\| < k \|x_0\| \exp(-\alpha t) .$$

La définition de la stabilité présente certains désavantages importants [9,28]:

- Il est nécessaire de pouvoir calculer de manière explicite chaque solution correspondant à chacune des conditions initiales.
- Le maniement de la définition est fastidieux.

Par conséquent, des résultats permettant de déterminer la stabilité sans devoir intégrer les équations différentielles seraient les bienvenues.

I.3.2 Méthode de Lyapunov :

On a vu que les critères précédents qui définissent la stabilité des systèmes nécessitent la résolution des équations de la dynamique pour tous les états initiaux possibles afin de pouvoir donner un avis sur la stabilité du système, ce qui est très difficile pour une très grande variété de systèmes.

I.3.2.1 Méthode indirecte de Lyapunov :

Une approximation locale de la dynamique du système autour du point d'équilibre permet, dans certains cas, de déduire la stabilité locale du système complet. Il s'agit de la méthode indirecte de Lyapunov [15]. Considérons le système (I.38).

Soit $x = 0$ un point d'équilibre du système où $f : D \rightarrow R_n$ est une fonction de classe $C^1(D)$ et D un voisinage de zéro. Soit

$$A = \left. \frac{\partial f}{\partial x}(x) \right|_{x=0} \quad (I.39)$$

$$\frac{\partial f}{\partial x} = \begin{bmatrix} \frac{\partial f_1}{\partial x_1} & \frac{\partial f_1}{\partial x_2} \\ \frac{\partial f_2}{\partial x_1} & \frac{\partial f_2}{\partial x_2} \end{bmatrix} \quad (I.40)$$

Alors :

1. L'origine est asymptotiquement stable si $\text{Re}(\lambda_i) < 0$ pour toutes les valeurs propres λ_i de A .
2. L'origine est instable si $\text{Re}(\lambda_i) > 0$ pour au moins une valeur propre λ_i de A .

I.3.2.2 Méthode directe de Lyapunov :

Fonction de Lyapunov :

La notion de fonction de Lyapunov constitue d'une certaine manière une généralisation de l'énergie. Etant donnée une fonction définie positive, l'idée directrice des théorèmes de Lyapunov consiste à évaluer l'évolution de cette fonction sur les trajectoires du système afin de conclure à la décroissance de l'énergie. [15, 28]

Lorsque on prend comme exemple le mouvement d'une bille dans un bol, et en l'examinant, on constate que le comportement stable ou instable (rester dans le bol ou sortir) de celle-ci est liée à la fois à la caractéristique et à l'évolution de sa fonction d'énergie. La présence d'un maximum ou minimum d'énergie potentielle possède une influence critique. De plus, la présence de frottement est responsable de la décroissance de l'énergie complète (cinétique et potentielle) et influence donc la stabilité. La bille possède donc une fonction d'énergie E qui comporte une part d'énergie potentielle E_p et une part d'énergie cinétique E_c .

On a $E = E_c + E_p$. Le comportement est stable lorsque [28]:

- L'énergie E diminue et est minimum au point d'équilibre.
- L'énergie E est conservée et E est minimum à l'équilibre

Par contre, le comportement est instable lorsque :

- L'énergie E augmente.
- L'énergie E est conservée mais elle ne correspond pas à un minimum à l'équilibre.

Théorème [15] :

Etant donné le système $\dot{x} = f(x)$, avec un point d'équilibre à l'origine $f(0) = 0$, et on considère un voisinage N au tour de l'origine d'un diamètre ε tel que $N = \{x, \|x\| \leq \varepsilon\}$

Ainsi :

Stabilité : l'origine est stable au sens de Lyapunov, si $\forall x \in N$ il existe une fonction scalaire $V(x)$ d'une dérivé partielle continue tel que

- a) $V(x)$ est définit positif
- b) \dot{V} est négatif semi définit

Stabilité asymptotique : l'origine est asymptotique stable si (a) est vérifié et \dot{V} définit négatif.

Stabilité asymptotique global : l'origine est asymptotique globalement stable si (a) est vérifié et \dot{V} définit négatif pour tout $x \in R^n$, c.-à-d. $N = R^n$.

I.3.3 Quelques stratégies de commande :

Le domaine de la commande des robots manipulateurs est très vaste, il est encore en pleine évolution. Plusieurs algorithmes de commande ont été développés pour satisfaire les objectifs exigés par les cahiers de charge. Pour avoir une idée sur les procédures existantes on va présenter dans cette partie quelques méthodes de commande dites basiques.

I.3.3.1 La commande par contrôleur PID [16]:

Cette commande est l'une des commandes les plus utilisées dans le domaine de l'automatisme à cause de sa simplicité et de son efficacité. Elle consiste à construire une loi de commande à partir de l'erreur de suivi, sa dérivée et son intégrale tel que $u = k_p e + k_i \int_0^t e \, dr + k_d \dot{e}$, où k_p, k_i, k_d représente le gain proportionnel, intégral et dérivé respectivement et e représente la différence entre l'état actuel et l'état désiré.

I.3.3.2 Commande par linéarisation entrée-sortie [15, 26, 34]:

Les robots manipulateurs sont des systèmes non linéaires, et la commande linéaire PID classique ne peut pas satisfaire des bonnes performances dans tous les cas. Ainsi pour exploiter les outils puissants de la commande linéaire, on élabore des commandes qui linéarisent le système afin de pouvoir utiliser ces outils. Si on considère le modèle (I.36) on peut choisir la commande linéarisante suivante :

$$\tau = \ddot{q}_d + M(q)^{-1}[C(q, \dot{q})\dot{q} + G(q) + u] \quad (I.41)$$

On appelle ce type de commande pour les bras manipulateurs commande par couple calculé. Pour avoir le modèle du système bouclé on substitue (I.41) dans (I.37) on obtient :

$$\ddot{e} + u = 0 \quad (I.42)$$

Maintenant on choisit u qui assure la stabilité du système en utilisant les outils des systèmes linéaires.

I.3.3.3 Commande non linéaire [15, 26, 34] :

Le problème que posent les commandes linéarisante est de supprimer les non linéarité des systèmes qui peuvent être utiles, ainsi on fait souvent recoure aux commandes non linéaire qui on prouvées leur efficacité, parmi eux on site :

La commande par mode glissant : qui est une commande robuste basé sur la théorie des structures variable qu'on va expliquée en détaille dans le chapitre suivant.

La commande backstepping : c'est une commande adaptatif, qui utilise la théorie de Lyapunov pour assure la stabilité, cette méthode aussi va être expliqué dans le chapitre suivant.

La commande passive : Cette commande se base sur le principe de passivité, qu'on peut définir pour un système par l'existence d'une certaine fonction de stockage V positive qui assure l'expression : $u^T y > \dot{V}$ où u représente l'entrée et y représente la sortie.

I.3.3.4 Commande basée sur l'intelligence artificielle [26, 36]:

La commande basée sur l'intelligence artificielle représente un domaine de recherche très active et vaste à cause des avantages qu'offrent ces outils surtout pour les systèmes non linéaires. Elle est fondée sur l'exploitation des capacités d'apprentissage, d'approximation et d'optimisation qui caractérise ces outils. Parmi Ces outils on cite les réseaux de neurones artificiels, la logique floue et les algorithmes génétiques.

❖ Il y a aussi la combinaison de plus d'une méthode pour obtenir des algorithmes hybrides qui englobent les différents avantages de ces méthodes.

I.4 Conclusion :

Dans ce chapitre nous avons présenté dans la première partie les différentes méthodes de modélisation des bras manipulateurs, en notant les problèmes rencontrés lors de l'utilisation de ces modèles. La deuxième partie présente une introduction à la théorie de commande tout en expliquant le concept de la stabilité des systèmes en faisant un petit rappel sur la théorie de Lyapunov, et présentant quelque commande fréquente des robots manipulateurs.

Chapitre II

Commande par Mode glissant et par Backstepping

II.1 Introduction :

Un modèle mathématique constitue souvent une description approchée de la réalité physique, et la loi de commande ne pourrait être construite que sur ce dernier. Ainsi la commande choisie devra être robuste dans le sens où elle devra garantir une faible sensibilité aux erreurs et aux incertitudes sur les paramètres, à leurs variations et aux perturbations. Ainsi deux approches de commandes complémentaires sont traitées dans ce chapitre pour remédier à ces problèmes : La première approche est la commande par mode glissant qui appartient aux commandes robustes et la deuxième est la commande adaptative backstepping. Ces deux méthodes vont être étudiées et appliquées sur un bras manipulateur à 2 ddl en observant leur comportement vis-à-vis l'existence d'incertitudes et de perturbations.

II.2 Commande par mode glissant :

La commande par mode glissant (sliding mode control dans la littérature anglaise) est une technique issue des recherches menées sur la théorie des systèmes à structure variables [2]. Les premiers travaux sur ce type de commande ont apparus au début des années soixante grâce à la résolution de l'équation différentielle à second membre discontinu par le mathématicien Russe A. Fillipov, suivie des recherches de S. Emelyanov en 1967 et de V. Utkin en 1977. [3, 4]

Cette technique se base sur l'évidente idée, qu'il est beaucoup plus facile de commander des systèmes de 1^{er} ordre (c.-à-d., systèmes décrits par des équations différentielles de 1^{er} ordre) avec leur non linéarité et leur incertitudes que de commander des systèmes d'ordre n. Par conséquent, une simplification d'écriture va être introduite, ce qui permet à des problèmes d'ordre n d'être remplacés par des problèmes équivalents de 1^{er} ordre [18] dont leur commande est beaucoup plus facile.

II.2.1 Structures de base [3]:

Dans les systèmes à structure variable utilisant la commande par mode de glissement, on peut distinguer trois configurations de base pour la synthèse des différentes commandes. La première correspond à un changement de structure par commutation d'une contre réaction d'état variable, La seconde change la structure au niveau de l'organe de commande, et enfin, la troisième configuration change aussi la structure au niveau de l'organe de commande mais avec ajout d'une commande dite « commande équivalente ».

II.2.1.1 Structure par commutation d'une contre réaction d'état :

La configuration de la structure par commutation d'une réaction d'état est représentée à la Figure II.1(a). Selon la position du commutateur, le vecteur d'état x est mis en contre-réaction d'état soit par $-k_1$ soit par $-k_2$. Ceci se fait à l'aide de la loi de commutation $s(x)$.

$$\begin{cases} u = -k_1^T(x) \text{ si } s(x) > 0 \\ u = -k_2^T(x) \text{ si } s(x) < 0 \end{cases} \quad (II.1)$$

Le comportement dynamique du système Σ est déterminé par $s(x) = 0$.

II.2.1.2 Structure par commutation au niveau de l'organe de commande :

Dans ce cas de configuration, la loi de commutation est donnée par :

$$u = \begin{cases} u_{max} \text{ si } s(x) > 0 \\ u_{min} \text{ si } s(x) < 0 \end{cases} \quad (II.2)$$

En mode de glissement (ou régime glissant), la dynamique du système Σ est donnée par $s(x) = 0$. Cette configuration s'adapte bien pour la commande de convertisseurs électriques dont l'organe de commande est un interrupteur.

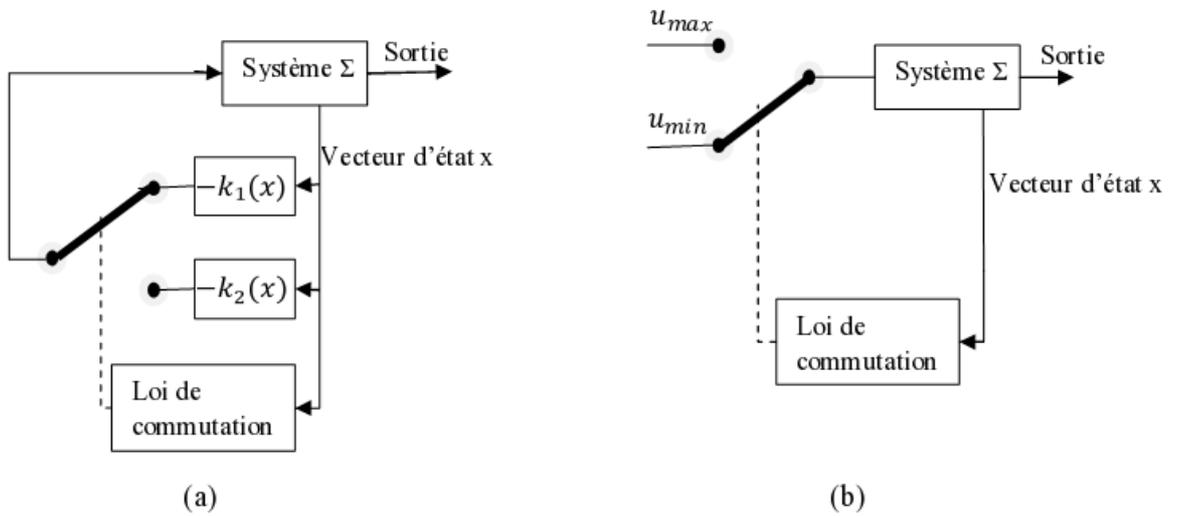


Figure II.1 : (a) Structure par commutation d'une contre réaction d'état, (b) Structure par commutation au niveau de l'organe de commande

II.2.1.3 Structure par commutation au niveau de l'organe de commande, avec ajout de la commande équivalente :

Le schéma d'une telle structure est représenté à la Figure II.2. Cette structure de commande est simple à réaliser et elle a été utilisée dans beaucoup d'applications. L'ajout de la commande équivalente permet de pré-positionner le système dans un état désiré stable et en plus de réduire le phénomène de réticence.

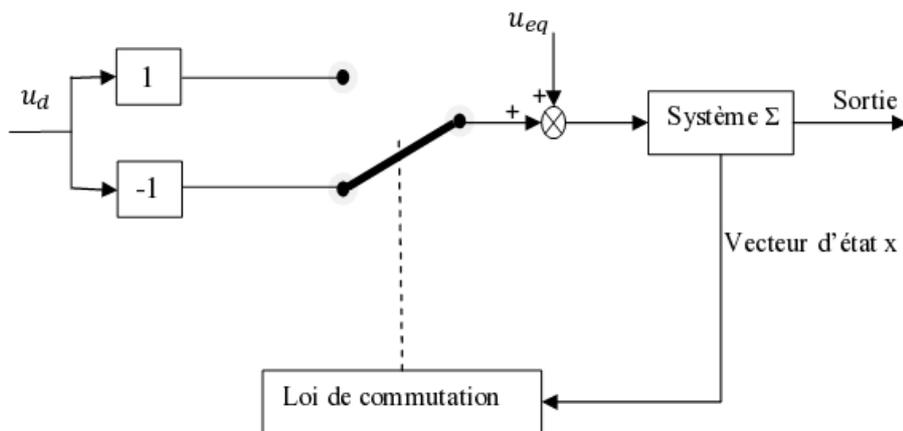


Figure II.2 : commutation au niveau de l'organe de commande, avec ajout de la commande équivalente

Le terme de commutation u_d assure principalement la convergence des trajectoires du système vers l'état désiré et assure son maintien. La loi de commutation est donnée par :

$$u = \begin{cases} u_{eq} + u_d & \text{si } s(x) > 0 \\ u_{eq} - u_d & \text{si } s(x) < 0 \end{cases} \quad (II.3)$$

II.2.2 Principe de la commande par modes glissants :

Le principe de la commande par mode glissant est de forcer en premier temps la dynamique du système à atteindre une surface prédéfinie, appelée surface de glissement, une fois arrivée à cette surface, elle y reste. La première phase est appelée mode de convergence et la deuxième c'est le mode de glissement [18] comme le montre la figure (II.3).

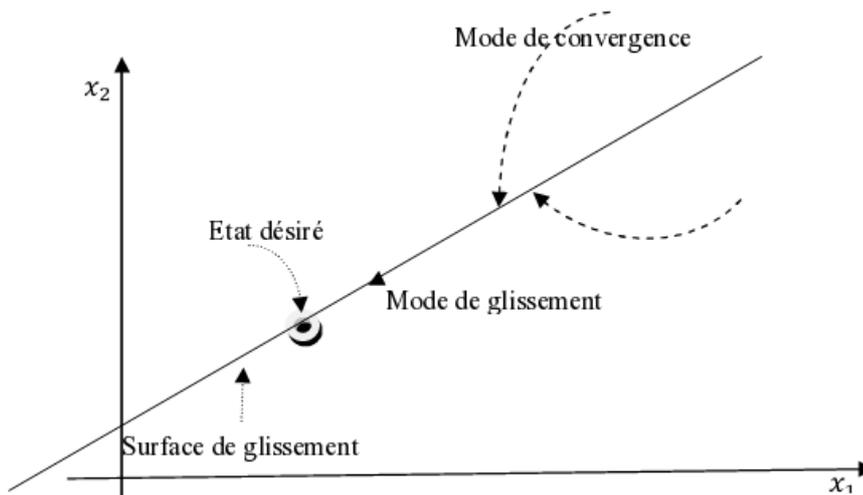


Figure II.3 : Convergence vers la surface de glissement

II.2.2.1 Définition du régime glissant :

Le régime glissant est représenté par le mouvement qui se produit le long de la surface de glissement après avoir été attiré par celle-ci. D'où la condition d'existence du mode de glissement est l'attractivité de la surface de commutation $s(x) = 0$.

❖ Conditions d'existence :

Nous considérons une classe de système non-linéaire décrite par l'équation différentielle suivante :

$$\dot{x} = f(x) + g(x)u \quad (II.4)$$

Où $x = [x_1, x_2, \dots, x_n]^T$ représente le vecteur d'état de dimension n , u est le vecteur d'entrée de commande, et $f(x)$ et $g(x)$ sont des champs de vecteur suffisamment différentiables.

Soit s une fonction continue de x , tel que $s(x) = 0$. On appelle la fonction $s(x)$ fonction de glissement ou fonction de commutation. Elle sépare l'espace d'état en deux parties disjointes $s(x) > 0$ et $s(x) < 0$.

On peut dire que le mode glissant existe et stable, si on peut choisir une fonction de surface s , tel que la fonction de Lyapunov

$$V(x) = \frac{1}{2}s(x)^2 \quad (II.5)$$

Assure la stabilité du système, c'est à dire que la fonction :

$$\dot{V}(x) = s(x)\dot{s}(x) < 0 \quad (II.6)$$

est négative, d'où elle assure la convergence de la trajectoire vers la surface s .

L'équation (II.5) explique que le carré de la distance entre un point donné du plan de phase et la surface de glissement exprimé par $s(x)^2$ diminue tout le temps [4], contraignant la trajectoire du système à se diriger vers la surface à partir des deux côtés de cette dernière. Cette condition suppose un régime glissant idéal ou la fréquence de commutation est infinie.

II.2.2.2 Choix de la Surface de glissement :

La surface de glissement est une fonction scalaire telle que la variable à régler glisse sur cette surface et tend vers l'origine du plan de phase. Le choix de ses paramètres dépend des performances désirées. Elle est construite de façon à faire apparaître la commande u explicitement dans la dérivée de la surface $s(x)$ par rapport au temps. Ainsi plusieurs formes d'écriture de la surface ont été proposées, la plus répandue est celle de Soltine [18], qui a proposé la forme générale :

$$s = \left(\frac{d}{dt} + \lambda\right)^{n-1} e(t) \quad (II.7)$$

Où : λ est une constante positive, n est le degré relatif, égale au nombre de fois qu'il faut dériver la sortie pour faire apparaître la commande

II.2.2.3 Construction de la commande équivalente par la méthode de Fillipov [18]:

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

$$s(x) = 0 \text{ et } \dot{s}(x) = 0 \quad (II.8)$$

On résolvant le système (II.8) pour une certaine entrée de commande, on va obtenir une expression appelée la commande équivalente, qu'on peut définir comme étant la commande qui maintient l'état sur la surface de glissement $s(x) = 0$. Elle est déterminée de façon unique, on mettait $\dot{s}(x) = 0$, lorsque tous les paramètres du modèle sont connus, avec l'inexistence de perturbation extérieur. Par exemple on prend un système de la forme [18]

$$\ddot{x} = f + u \quad (II.9)$$

$$e = x - x_d \Rightarrow s = \left(\frac{d}{dt} + \lambda \right)^1 e(t) = \dot{e} + \lambda e \quad (II.10)$$

$$\Rightarrow \dot{s} = \ddot{e} + \lambda \dot{e} = \ddot{x} - \ddot{x}_d + \lambda \dot{e} = f + u - \ddot{x}_d + \lambda \dot{e} \quad (II.11)$$

En appliquant la condition (II.8) qu'elle s'appelle condition d'invariance de la surface de glissement on obtient :

$$\dot{s} = 0 \Rightarrow f + u_{eq} - \ddot{x}_d + \lambda \dot{e} = 0 \quad (II.12)$$

Ainsi :

$$u_{eq} = \ddot{x}_d - f - \lambda \dot{e} \quad (II.13)$$

Physiquement, la commande équivalente représente la valeur moyenne de la commande u [4], qui maintient l'état du système sur la surface de glissement $s(x) = 0$.

II.2.3 Synthèse de la loi de commande :

La synthèse de la loi de commande est fondée sur les deux étapes qui constituent la réponse du mode glissant représentées par la phase de convergence et la phase de glissement. Pour cela la théorie des structures variables propose la loi suivante :

$$u = \begin{cases} u^+(x) & \text{si } s(x) > 0 \\ u^-(x) & \text{si } s(x) < 0 \end{cases} \quad (II.14)$$

Cette loi est composée de deux termes :

$$u = u_{eq} + u_d \quad (II.15)$$

Le terme u_{eq} est une fonction continue qui va maintenir la dynamique du système sur la surface de glissement. On appelle ce fonctionnement mode de glissement. On conçoit cette commande suivant les étapes défini dans le paragraphe II.3.2.3.

La partie de la commande u_d est de nature discontinue [18], c'est elle qui va contraindre les trajectoires du système à atteindre la surface de glissement et d'y rester au voisinage de celle-ci malgré la présence de perturbations. En d'autres termes, elle va rendre cette surface localement attractive en assurant la phase de convergence. Différentes formes sont proposées dans la littérature pour exprimer u_d , la forme généralement utilisée est $u_d = -k \text{sign}(s)$ où k est une constante positive et sign est la fonction signe classique,

$$\text{sign}(s) = \begin{cases} +1 & \text{si } s > 0 \\ 0 & \text{si } s = 0 \\ -1 & \text{si } s < 0 \end{cases} \quad (II.16)$$

II.2.4 Robustesse des modes glissants :

Pour étudier la robustesse du mode glissant, on considère comme exemple le système perturbé suivant :

$$\begin{cases} \dot{x}_1 = x_2 \\ \dot{x}_2 = f(x) + g(x)u + b \end{cases} \quad (II.17)$$

Où u est l'entrée de la commande, x est le vecteur d'état, f et g représente la dynamique du système et p représente les imperfections de la modélisation et les perturbations qu'on suppose bornées tel que $|b| < b_m, \forall x \in R^n$.

Pour stabilisé le système autour d'une sortie définie $x_1 = x_d$, on choisit une surface $s = 0$, comme présenté dans le paragraphe II.2.2.2 avec $e = x - x_d$:

$$s = \left(\frac{d}{dt} + \lambda \right) e = \dot{e} + \lambda e \quad (II.18)$$

$$\Rightarrow \dot{s} = \ddot{e} + \lambda \dot{e} = f(x) + g(x)u + b - \ddot{x}_d + \lambda \dot{e} \quad (II.19)$$

Dans le cas où les imperfections de la modélisation et les perturbations sont nulles $b = 0$, le système peut être stabilisé ($\dot{s} = 0$) par la commande équivalente :

$$u_{eq} = g(x)^{-1}(\ddot{x}_d - f(x) - \lambda x_2) \quad (II.20)$$

Cependant si les perturbations existent, le système ne peut pas se maintenir sur la surface, ainsi on choisit une loi de commande qui stabilise le système $\dot{s} = 0$ et le rend attractif. En prenant la fonction de Lyapunov $V = \frac{1}{2} s^2$, ce qui donne :

$$\dot{V} = s\dot{s} = s(f(x) + g(x)u + b - \ddot{x}_d + \lambda \dot{x}) \quad (II.21)$$

Et en choisissant $u = u_{eq} + u_c$, et substituant (II.20) dans (II.21) on aura :

$$\dot{V} = s(b + u_c) \quad (II.22)$$

On choisit la loi de commande discrète $u_c = -k \text{sign}(s)$:

$$\dot{V} = s(b - k \text{sign}(s)) \leq |s|(b_m - k) \quad (II.23)$$

Si on choisit $k > b_m$ alors la surface deviendra attractive et le système est stable.

Ainsi le théorème suivant est énoncé :

Théorème [3, 4] :

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

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

II.3 Commande par backstepping:

Le backstepping a été développé par Kanellakopoulos et al, et inspiré par les travaux de Feurer et Morse d'une part et Tsinias et Kokotovii et Sussmann d'autre part [11]. L'idée principale de cette stratégie de commande repose sur la construction d'un algorithme qui permet de concevoir une loi de commande du système globale à partir des lois de commande virtuelle conçue pour les sous-systèmes qui se résultent de la fragmentation du système globale [12, 15]. En premier lieu, on commence par concevoir la loi virtuelle pour le sous-système d'ordre inférieur en élaborant la fonction de Lyapunov qui assure sa stabilité. Ensuite, pour le deuxième sous système on calcule une nouvelle commande virtuelle et une deuxième fonction de Lyapunov et ainsi de suite selon l'ordre du système, puis, à chaque étape l'ordre du système augmente et la partie non stabilisée lors de l'étape précédente est traitée. Enfin, en dernier lieu, on obtient l'expression de la commande qui garantit la stabilité globale du système.

II.3.1 Conception de la loi de commande par backstepping:

Pour montrer comment construire une loi de commande par backstepping on va utiliser l'exemple suivant [15], soit le système non linéaire:

$$\begin{cases} \dot{x}_1 = f(x_1) + g(x_1) x_2 \\ \dot{x}_2 = u \end{cases} \quad (II.24)$$

En supposant que f et g sont connues, ainsi une loi de commande va être construite pour stabiliser l'origine du système $((x_1, x_2) = (0,0))$. Ce système peut être représenté comme une connexion en cascade de deux composantes comme le représente la Figure II.4. (a); la première composante avec comme entrée x_2 , et la 2^{ème} représente un intégrateur.

Pour commencer, on suppose que la première composante peut être stabilisée par une certaine loi de commande $x_2 = \varphi(x_1)$, avec $\varphi(0) = 0$. ce qui donne

$$\dot{x}_1 = f(x_1) + g(x_1)\varphi(x_1) \quad (II.25)$$

Qui est un système asymptotiquement stable, si on peut trouver une fonction de Lyapunov $V(x_1)$ qui satisfait l'inégalité $\dot{V} \leq -W(x_1) \quad \forall x_1 \in D$, dont $W(x_1)$ est défini positive.

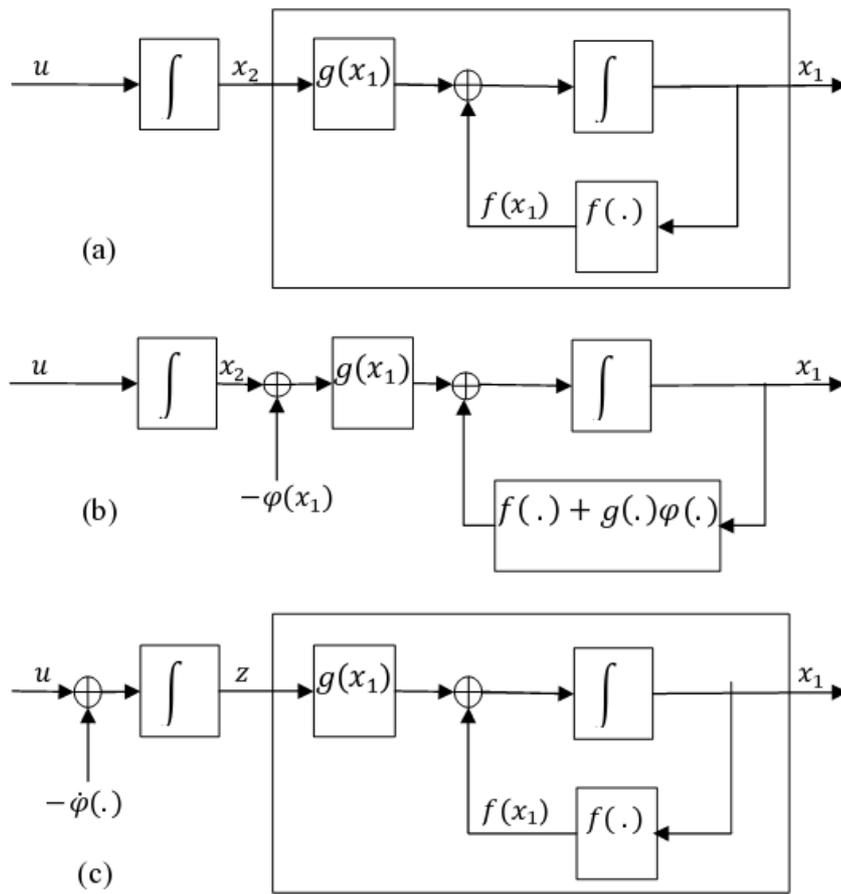


Figure II.4 : (a), (b), (c) Les étapes de conception du backstepping

Ainsi notre système (II.24) peut être représenté par :

$$\begin{cases} \dot{x}_1 = f(x_1) + g(x_1) \varphi(x_1) + g(x_1) (x_2 - \varphi(x_1)) \\ \dot{x}_2 = u \end{cases} \quad (II.26)$$

On fait le changement de variable $z = x_2 - \varphi(x_1)$ ce qui donne :

$$\begin{cases} \dot{x}_1 = f(x_1) + g(x_1) \varphi(x_1) + g(x_1) z \\ \dot{z} = u - \dot{\varphi} \end{cases} \quad (II.27)$$

Par ce que f, g et φ sont connue alors $\dot{\varphi}$ peut être calculé par l'expression

$$\dot{\varphi} = \frac{\partial \varphi}{\partial x_1} [f(x_1) + g(x_1) x_2] \quad (II.28)$$

Prendre $v = u - \dot{\varphi}$ donne :

$$\begin{cases} \dot{x}_1 = f(x_1) + g(x_1) \varphi(x_1) + g(x_1)z \\ \dot{z} = v \end{cases} \quad (II.29)$$

Le système (II.29) est similaire au système initiale, sauf que la première composante est asymptotiquement stable lorsque l'entrée v est nulle. Maintenant, on va utiliser cette propriété pour stabiliser le système global.

On prend $V_t(x_1, x_2) = V(x_1) + \frac{1}{2}z^2$ comme fonction de Lyapunov pour le système globale.

Ce qui donne

$$\dot{V}_t = \frac{\partial V}{\partial x_1} [f(x_1) + g(x_1) \varphi(x_1)] + \frac{\partial V}{\partial x_1} g(x_1) z + zv \leq -W(x_1) + \frac{\partial V}{\partial x_1} g(x_1) z + zv \quad (II.30)$$

$$\text{On choisit } v = -\frac{\partial V}{\partial x_1} g(x_1) - kz, k > 0 \quad (II.31)$$

$$\text{On aura } \dot{V}_t \leq -W(x_1) - kz^2 \quad (II.32)$$

Ainsi on obtient un origine asymptotiquement stable avec la loi de commande :

$$u = \frac{\partial \varphi}{\partial x_1} [f(x_1) + g(x_1) x_2] - \frac{\partial V}{\partial x_1} g(x_1) - k[x_2 - \varphi(x_1)] \quad (II.33)$$

Exemple [15]:

Soit le système:

$$\begin{cases} \dot{x}_1 = x_1^2 - x_1^3 + x_2 \\ \dot{x}_2 = u \end{cases} \quad (II.34)$$

Pour le stabiliser on commence par le premier sous système avec x_2 comme entré, on choisit la loi de commande $x_2 = \varphi(x_1)$ pour stabilisé l'origine $x_1 = 0$ avec $x_2 = \varphi(x_1) = -x_1^2 - x_1$ pour obtenir $\dot{x}_1 = -x_1 - x_1^3$ avec comme fonction de Lyapunov :

$$V(x_1) = \frac{x_1^2}{2} \Rightarrow \dot{V} = -x_1^2 - x_1^4 \leq 0 \quad (II.35)$$

Maintenant, on applique le changement de variable $z = x_2 - \varphi(x_1) = x_2 + x_1^2 + x_1$, notre système devient :

$$\begin{cases} \dot{x}_1 = -x_1 - x_1^3 + z \\ \dot{z} = u + (1 + 2x_1)(-x_1 - x_1^3 + z) \end{cases} \quad (II.36)$$

On choisit :

$$V_t(x) = \frac{1}{2}x_1^2 + \frac{1}{2}z^2 \quad (II.37)$$

$$\Rightarrow \dot{V}_t = x_1(-x_1 - x_1^3 + z) + z[u + (1 + 2x_1)(-x_1 - x_1^3 + z)] \quad (II.38)$$

$$= -x_1^2 - x_1^4 + z[x_1 + (1 + 2x_1)(-x_1 - x_1^3 + z) + u] \quad (II.39)$$

$$\text{Et si on prend } u = -x_1 - (1 + 2x_1)(-x_1 - x_1^3 + z) - z \rightarrow \dot{V}_t = -x_1^2 - x_1^4 - z^2 \quad (II.40)$$

On obtient un origine asymptotiquement stable.

Remarque :

1-Pour des systèmes plus général de la forme :

$$\begin{cases} \dot{x}_1 = f(x_1) + g(x_1) x_2 \\ \dot{x}_2 = f_a(x_1, x_2) + g_a(x_1, x_2)u \quad / \quad g_a(x_1, x_2) \neq 0 \end{cases} \quad (II.41)$$

En appliquant l'entrée $u = g_a(x_1, x_2)^{-1}[u_a - f_a(x_1, x_2)]$ le système va être transformé pour être similaire au système (II.24). [15]

2- le backstepping s'applique d'une manière récursive pour des systèmes plus complexe qui se présentent sous la forme dite paramétrique pure. Les équations d'un tel système sont données par : [15]

$$\left\{ \begin{array}{l} \dot{x} = f_0(x) + g_0(x)z_1 \\ \dot{z}_1 = f_1(x, z_1) + g_1(x, z_1)z_2 \\ \dot{z}_2 = f_2(x, z_1, z_2) + g_2(x, z_1, z_2)z_3 \\ \vdots \\ \vdots \\ \vdots \\ \dot{z}_{k-1} = f_{k-1}(x, z_1, \dots, z_{k-1}) + g_{k-1}(x, z_1, \dots, z_{k-1})z_k \\ \dot{z}_k = f_k(x, z_1, \dots, z_k) + g_k(x, z_1, \dots, z_k)u \end{array} \right. \quad (II.42)$$

II.3.2 Backstepping et perturbation :

Dans cette partie on va voir les contraintes qui s'imposent à la commande par backstepping pour pouvoir être robustes pour un certain type de perturbations et incertitudes [15].

On considère le système :

$$\begin{cases} \dot{x}_1 = f(x_1) + g(x_1)x_2 + \delta_{x_1}(x_1, x_2) \\ \dot{x}_2 = f_a(x_1, x_2) + g_a(x_1, x_2)u + \delta_{x_2}(x_1, x_2) / g_a(x_1, x_2) \neq 0 \end{cases} \quad (II.43)$$

On suppose que f, g, f_a et g_a sont connues et δ_{x_1} et δ_{x_2} sont des termes incertains avec $f(0) = f_a(0,0) = 0$ et les incertitudes sont bornés par [15]:

$$\|\delta_{x_1}(x_1, x_2)\|_2 \leq a_1 \|x_1\|_2 \quad (II.44)$$

$$|\delta_{x_2}(x_1, x_2)| \leq a_2 \|x_1\|_2 + a_3 |x_2| \quad (II.45)$$

On suppose qu'il ya une loi stabilisante $x_2 = \varphi(x_1)$ et $\varphi(0) = 0$, et une fonction de Lyapunov [...] tel que :

$$\frac{\partial V}{\partial x_1} [f(x_1) + g(x_1)\varphi(x_1) + \delta_{x_1}(x_1, x_2)] \leq -b \|x_1\|_2^2 / \quad b > 0 \quad (II.46)$$

Ce qui indique que $x_1 = 0$ est un point d'équilibre asymptotiquement stable pour la 1^{ère} partie du système (II.43)

$$\text{Et supposant aussi que } |\varphi(x_1)| \leq a_4 \|x_1\|_2, \quad \left\| \frac{\partial \varphi}{\partial x_1} \right\| \leq a_5 \quad (II.47)$$

$$\text{Maintenant on considère la fonction de Lyapunov } V_t = V(x_1) + \frac{1}{2} [x_2 - \varphi(x_1)]^2 \quad (II.48)$$

$$\begin{aligned} \dot{V}_t &= \frac{\partial V}{\partial x_1} [f + g\varphi + \delta_{x_1}] + \frac{\partial V}{\partial x_1} g [x_2 - \varphi] \\ &\quad + (x_2 - \varphi) \left[f_a + g_a u + \delta_{x_2} - \frac{\partial \varphi}{\partial x_1} (f + g x_2 + \delta_{x_1}) \right] \end{aligned} \quad (II.49)$$

$$\text{On prend } u = g_a^{-1} \left[\frac{\partial \varphi}{\partial x_1} (f + g x_2) - \frac{\partial V}{\partial x_1} g - f_a - k(x_2 - \varphi) \right], \quad k > 0 \quad (II.50)$$

$$\text{On va avoir } \dot{V}_t \leq -b \|x_1\|_2^2 + (x_2 - \varphi) \left[\delta_{x_2} - \frac{\partial \varphi}{\partial x_1} \delta_{x_1} \right] - k(x_2 - \varphi)^2 \quad (II.51)$$

$$\dot{V}_t \leq -b\|x_1\|_2^2 + 2a_6\|x_1\|_2|x_2 - \varphi| - (k - a_3)(x_2 - \varphi)^2 \quad (\text{II.52})$$

$$= - \begin{bmatrix} \|x_1\|_2 \\ |x_2 - \varphi| \end{bmatrix}^T \begin{bmatrix} b & -a_6 \\ -a_6 & (k - a_3) \end{bmatrix} \begin{bmatrix} \|x_1\|_2 \\ |x_2 - \varphi| \end{bmatrix} \quad (\text{II.53})$$

Pour un certain $a_6 \geq 0$ si on choisit $k > a_3 + \frac{a_6^2}{b} \Rightarrow \dot{V}_t \leq -\sigma[\|x_1\|_2^2 + |x_2 - \varphi|^2] / \sigma > 0$.

Ainsi l'origine du système est stable.

II.4 Application sur le bras manipulateur :

II.4.1 Description du manipulateur utilisé :

Les principes de modélisation des bras manipulateur sont bien connus, ainsi dans notre travail on va prendre le modèle du robot défini par l'équation dynamique d'Euler-Lagrange du modèle suivant [1,26]:

$$M(q)\ddot{q} + C(q, \dot{q})\dot{q} + F_d\dot{q} + F_s(\dot{q}) + \tau_d(q, \dot{q}) = \tau \quad (\text{II.54})$$

Tel que :

q, \dot{q} et \ddot{q} : Vecteur $n \times 1$ de variable articulaire et de ses dérivées.

$M(q)$: Matrice $n \times n$ d'inertie du manipulateur.

$C(q, \dot{q})$: Vecteur $n \times 1$ de Coriolis et des couples centrifuges

$G(q)$: Vecteur $n \times 1$ de gravité.

F_d : Vecteur $n \times 1$ de frottement dynamique.

F_s : Vecteur $n \times 1$ de frottement statique.

τ_d : Vecteur $n \times 1$ de bruit externe.

τ : Vecteur $n \times 1$ de couple et/ou force.

Pour le manipulateur à 2 ddl utilisé dans ce mémoire on prend $n=2$.

Le modèle (II.54) ne peut pas donner la dynamique réelle du robot, par ce que si on suppose qu'il y a des paramètres inconnus ou mal connus, le modèle précédent ne peut pas nous donner des informations nécessaires pour la commande. Ainsi, on va introduire sur notre modèle les incertitudes paramétriques [1]. L'équation (II.54) devient alors:

$$(M(q) + \Delta M)\ddot{q} + (C(q, \dot{q}) + \Delta C)\dot{q} + F_d\dot{q} + F_s(\dot{q}) + \tau_d(q, \dot{q}) + (G(q) + \Delta G) = \tau \quad (\text{II.55})$$

$$\Rightarrow M(q)\ddot{q} + (C(q, \dot{q})\dot{q} + G(q)) + (\Delta M\ddot{q} + \Delta C\dot{q} + \Delta G) + (F_d\dot{q} + F_s(\dot{q}) + \tau_d(q, \dot{q})) = \tau \quad (\text{II.56})$$

$$\Rightarrow M(q)\ddot{q} + n(q, \dot{q}) + b_1 + b_2 = \tau \quad (\text{II.57})$$

Tel que :

$\Delta M, \Delta C, \Delta G$: Incertitudes paramétriques du modèle.

$$n(q, \dot{q}) = C(q, \dot{q})\dot{q} + G(q) \quad (\text{II.58})$$

$$b_1 = (\Delta M\ddot{q} + \Delta C\dot{q} + \Delta G) \quad (\text{II.59})$$

$$b_2 = F_d\dot{q} + F_s(\dot{q}) + \tau_d(q, \dot{q}) \quad (\text{II.60})$$

$$\rightarrow \ddot{q} = M(q)^{-1}\tau - M(q)^{-1}n(q, \dot{q}) - M(q)^{-1}b_1 - M(q)^{-1}b_2 \quad (\text{II.61})$$

II.4.1.1 Représentation d'états :

En effectuant le changement $x_1 = q, x_2 = \dot{q}$, et prendre comme sortie $y = x_1$ on obtient le système :

$$\begin{cases} \dot{x}_1 = x_2 \\ \dot{x}_2 = M(q)^{-1}\tau - M(q)^{-1}n(q, \dot{q}) - M(q)^{-1}b_1 - M(q)^{-1}b_2 \\ y = x_1 \end{cases} \quad (\text{II.62})$$

$$\rightarrow \begin{cases} \dot{x}_1 = x_2 \\ \dot{x}_2 = g_a u + f_a + b(t) \\ y = x_1 \end{cases} \quad (\text{II.63})$$

Tel que :

$$f_a = -M(q)^{-1}n(q, \dot{q}) \quad (\text{II.64})$$

$$g_a = M(q)^{-1} \quad (\text{II.65})$$

$$b(t) = -M(q)^{-1}b_1 - M(q)^{-1}b_2 = -M(q)^{-1}(b_1 + b_2) \quad (\text{II.66})$$

Avec $|b(t)| < \alpha$ / α représente les bornes supérieures d'incertitudes paramétriques et de perturbations, qu'on suppose connaître. [15, 1]

II.4.2 Application du mode glissant :

Dans cette partie, on va appliquer la commande par mode glissant pour stabiliser la position d'un bras manipulateur à 2 ddl sur un vecteur de coordonnées $q = [q_{d1}, q_{d2}]^T$ bien définie. La dynamique du manipulateur utilisée est incertaine et perturbée, définie par l'équation (II.57). L'architecture du système complet (manipulateur, commande) est présentée par la figure (II.5).

La première étape de la conception de la loi de commande est de choisir l'expression de la surface de glissement qui va définir les performances du système, ensuite, on procède à la conception de la loi qui va assurer la convergence de la réponse vers cette surface.

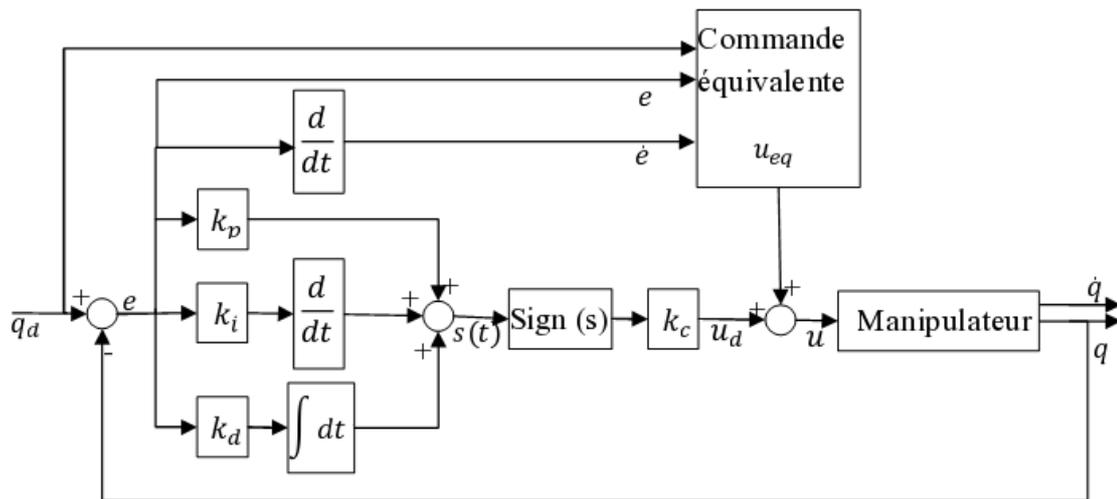


Figure II.5 : schéma block de la commande par mode glissant avec une surface PID.

D'abord on transforme le problème du positionnement en un problème de stabilisation en faisant le changement de variable $e = q - q_d$, et on étudie la stabilité du système sur $s = 0$.

L'expression de la surface choisie est de la forme [1] $s = \left(\frac{d}{dt} + \lambda\right)^{n-1} \left(\int_0^t e(r) dr\right)$, tel que :

$$s(t) = k_p e(t) + k_i \int e(\xi) d\xi + k_d \frac{d}{dt} e(t) \quad (\text{II.67})$$

Où k_p , k_i et k_d ce sont des gains de valeur positive.

Et on continue, en cherchant la dérivé de la surface et étudier la stabilité

$$\dot{s}(t) = k_p \dot{e} + k_i e + k_d \ddot{e} \quad (II.68)$$

$$\dot{s}(t) = k_p \dot{e} + k_i e + k_d [\ddot{q} - \ddot{q}_d] \quad (II.69)$$

$$= k_p \dot{e} + k_i e + k_d [-\ddot{q}_d + g_a u + f_a + b(t)] \quad (II.70)$$

Lorsque il n'y a pas de perturbations ou d'incertitude de modélisation ($b(t) = 0$), la commande qui va maintenir la dynamique sur la surface de glissement ($\dot{s} = 0$) s'appelle la commande équivalente, qu'elle correspond à :

$$u_{eq} = (k_d g_a)^{-1} [-k_p \dot{e} - k_i e - k_d f_a + k_d \ddot{q}_d] \quad (II.71)$$

Cependant, s'il y a apparition de perturbations et/ou d'incertitudes paramétriques, la commande équivalente ne peut pas assurer les performances voulues, ainsi il faut ajouter un signal de commande qui va éliminer l'effet des perturbations, en assurant la stabilité et la robustesse du système. Pour cela, il faut élaborer une fonction de Lyapunov et faire une étude de stabilité.

On prend comme fonction de Lyapunov $V = \frac{1}{2} s^2$, avec la dérivée

$$\dot{V} = s \dot{s} = s [k_p \dot{e} + k_i e + k_d \ddot{e}] \quad (II.72)$$

$$= s [k_p \dot{e} + k_i e + k_d [-\ddot{q}_d + g_a u + f_a + b(t)]] \quad (II.73)$$

On choisit :

$$u = u_{eq} + u_c \quad (II.74)$$

Substituant u_{eq} par sa valeur (II.71) dans (II.73) on obtient :

$$\dot{V} = s [k_d g_a u_c + k_d b(t)] \quad (II.75)$$

$$\leq |s| k_d |b(t)| + s k_d g_a u_c \quad (II.76)$$

Avec $|b(t)| < \alpha$, ce qui donne :

$$\dot{V} \leq |s| k_d \alpha + s k_d g_a u_c \quad (II.77)$$

On prend le signal de commande $u_c = -k_c \text{sign}(s)$, où k_c est un gain positive. On obtient :

$$\dot{V} \leq |s|k_d\alpha - sk_dg_ak_c \text{sign}(s) \quad (II.78)$$

Pour assurer que $\dot{V} \leq 0$, il faut que $k_c \geq g_a^{-1}\alpha$, ainsi le système va être attiré vers la surface s , une fois arriver, il ne peut pas sortir, ensuite, l'étape de glissement commence à travers la surface s , jusqu'à l'origine. Ainsi la loi de commande par mode glissant pour le robot manipulateur est représentée par :

$$\tau = (k_dg_a)^{-1}[-k_p\dot{e} - k_i e - k_d f_a + k_d \ddot{q}_d] - k_c \text{sign}(s) \quad (II.79)$$

$$\tau = (k_dM(q)^{-1})^{-1}[k_d\ddot{q}_d - k_p\dot{e} - k_i e + M(q)^{-1}C(q, \dot{q})\dot{q} + G(q)] - k_c \text{sign}(s) \quad (II.80)$$

II.4.3 Application du backstepping :

Notre commande a pour but de faire suivre au manipulateur une certaine trajectoire $y = q_d$, donc obtenir une erreur de suivi minimale. On va transformer ce problème de poursuite et/ou régulation en un problème de stabilisation autour de zéro au lieu de q_d en faisant un changement de variables sur le système (II.63), tel que :

$$z_1 = x_1 - q_d, \dot{z}_1 = \dot{x}_1 - \dot{q}_d \quad (II.81)$$

On obtient :

$$\begin{cases} \dot{z}_1 = x_2 - \dot{q}_d \\ \dot{x}_2 = f_a(x_1, x_2) + g_a(x_1, x_2)u + b(t) \end{cases} \quad (II.82)$$

On prend $u = g_a(x_1, x_2)^{-1}(\tau_a - f_a(x_1, x_2))$ (voir remarque-1-), on trouve

$$\begin{cases} \dot{z}_1 = x_2 - \dot{q}_d \\ \dot{x}_2 = u_a + b(t) \end{cases} \quad (II.83)$$

Qui est similaire au système (II.24) lorsque $b(t) = 0$.

On continue maintenant en stabilisant la première équation du système en prenant x_2 comme entrée virtuelle. En choisit $x_2 = \varphi(z_1) = -k_1z_1$ pour faire décroître la fonction de Lyapunov V_1 tel que :

$$V_1 = \frac{1}{2}z_1^2 \Rightarrow \dot{V}_1 = z_1 \dot{z}_1 = -k_1z_1^2 \leq 0 \quad (II.84)$$

Ainsi, notre système peut être représenté comme :

$$\begin{cases} \dot{z}_1 = -k_1 z_1 + (x_2 + k_1 z_1 - \dot{q}_d) \\ \dot{x}_2 = u_a + b(t) \end{cases} \quad (II.85)$$

$$\Rightarrow \begin{cases} \dot{z}_1 = -k_1 z_1 + z_2 \\ \dot{x}_2 = u_a + b(t) \end{cases} \quad / \quad z_2 = x_2 + k_1 z_1 - \dot{q}_d \quad (II.86)$$

$$\text{Et on a} \quad \dot{z}_2 = \dot{x}_2 + k_1 \dot{z}_1 - \ddot{q}_d = u_a + b(t) + k_1(x_2 - \dot{q}_d) - \ddot{q}_d \quad (II.87)$$

$$\Rightarrow \begin{cases} \dot{z}_1 = -k_1 z_1 + z_2 \\ \dot{z}_2 = u_a + b(t) + k_1(x_2 - \dot{q}_d) - \ddot{q}_d \end{cases} \quad (II.88)$$

On choisit la fonction de Lyapunov $V_t = \frac{1}{2} z_1^2 + \frac{1}{2} z_2^2$:

$$\Rightarrow \dot{V}_t = z_1 \dot{z}_1 + z_2 \dot{z}_2 = z_1(-k_1 z_1 + z_2) + z_2(u_a + b(t) + k_1(x_2 - \dot{q}_d) - \ddot{q}_d) \quad (II.89)$$

$$\text{On prend } u_a = (-k_1(x_2 - \dot{q}_d) + \ddot{q}_d - k_2 z_2 - z_1) \quad (II.90)$$

Où, k_1 et k_2 sont des gains positive.

$$\Rightarrow \dot{V}_t = -k_1 z_1^2 - k_2 z_2^2 + z_2 b(t) \leq -k_1 z_1^2 - k_2 z_2^2 + |z_2| |b(t)| \quad (II.91)$$

$$\leq -k_1 z_1^2 - k_2 z_2^2 + |z_2| \alpha \quad / \quad |b(t)| \leq \alpha \quad (II.92)$$

$$= -k_1 z_1^2 - \alpha |z_2| \left(\frac{k_2}{\alpha} |z_2| - 1 \right) \quad (II.93)$$

Pour assurer $\dot{V}_t \leq 0$, il faut que $\left(\frac{k_2}{\alpha} |z_2| - 1 \right) > 0 \Rightarrow k_2 > \frac{\alpha}{|z_2|}$. Ça veut dire que si on choisit une certaine valeur k_2 tel que $k_2 > \alpha \beta$, on aura une solution bornée [12] et l'état du système converge vers la solution $z_2 = \frac{1}{\beta}$, une fois arriver la, on peut rien dire car son comportement est influencé par le signe de la somme $(-k_1 z_1^2 - k_2 z_2^2 + z_2 b(t))$.

Enfin le couple total $\tau = u$ égale à :

$$\tau = g_a^{-1}(\tau_a - f_a(z_1, x_2)) \quad (II.94)$$

$$= g_a^{-1}((-k_1(x_2 - \dot{q}_d) + \ddot{q}_d - k_2 z_2 - z_1) - f_a(z_1, x_2)) \quad (II.95)$$

$$= M(q)[(-k_1(x_2 - \dot{q}_d) + \ddot{q}_d - k_2 z_2 - z_1) - M(q)^{-1}(-C(q, \dot{q})\dot{q} - G(q))] \quad (II.96)$$

II.5 Simulation et résultats :

Tout au long de ce mémoire on va simuler les algorithmes de commande étudiés sur un modèle de bras manipulateur à 2 ddl en utilisant le logiciel de simulation Matlab.

Le modèle dynamique avec l'expression des frottements dynamiques, statiques et les perturbations externes utilisées sont présentées dans l'annexe, en intégrant les incertitudes du modèle dans l'expression de τ_d . La sortie des couples est soumis à une fonction de saturation de valeur 250,150 N.m pour le premier et le deuxième bras respectivement.

II.5.1 Commande par mode glissant :

a) régulation :

La trajectoire désirée pour la régulation des deux articulations est:

$$q_{d1,2} = u(t) = \begin{cases} 1 \text{ rad} & \text{pour } t \geq 0 \\ 0 & \text{ailleurs} \end{cases}$$

Les valeurs des paramètres utilisés pour la régulation :

$$k_p = \begin{bmatrix} 65 & 0 \\ 0 & 75 \end{bmatrix}, k_d = \begin{bmatrix} 9 & 0 \\ 0 & 7 \end{bmatrix}, k_i = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}, k_c = \begin{bmatrix} 400 & 0 \\ 0 & 350 \end{bmatrix}$$

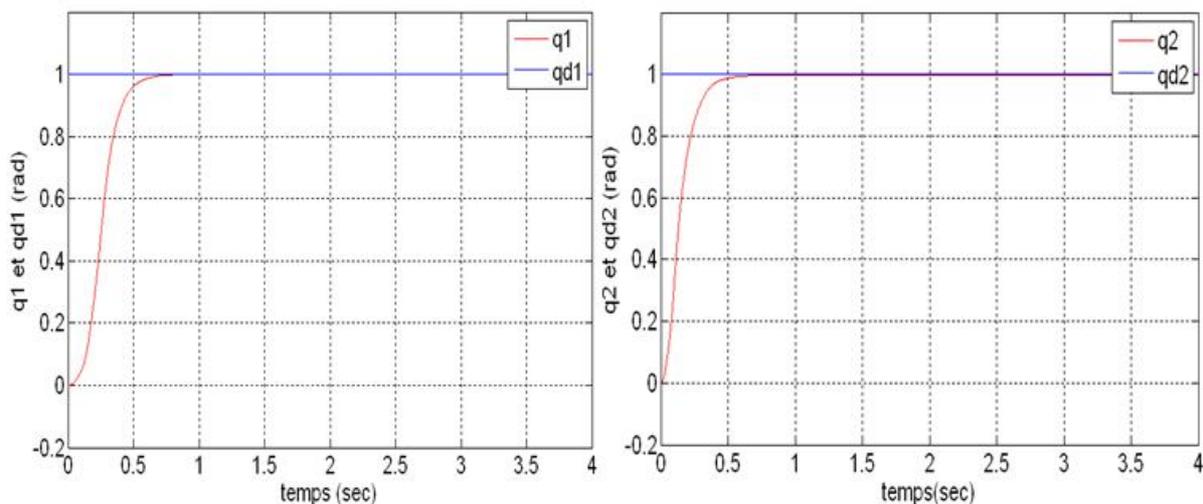


Figure II.6 : Résultat de suivi de la trajectoire désirée pour les deux articulations -cas de régulation-

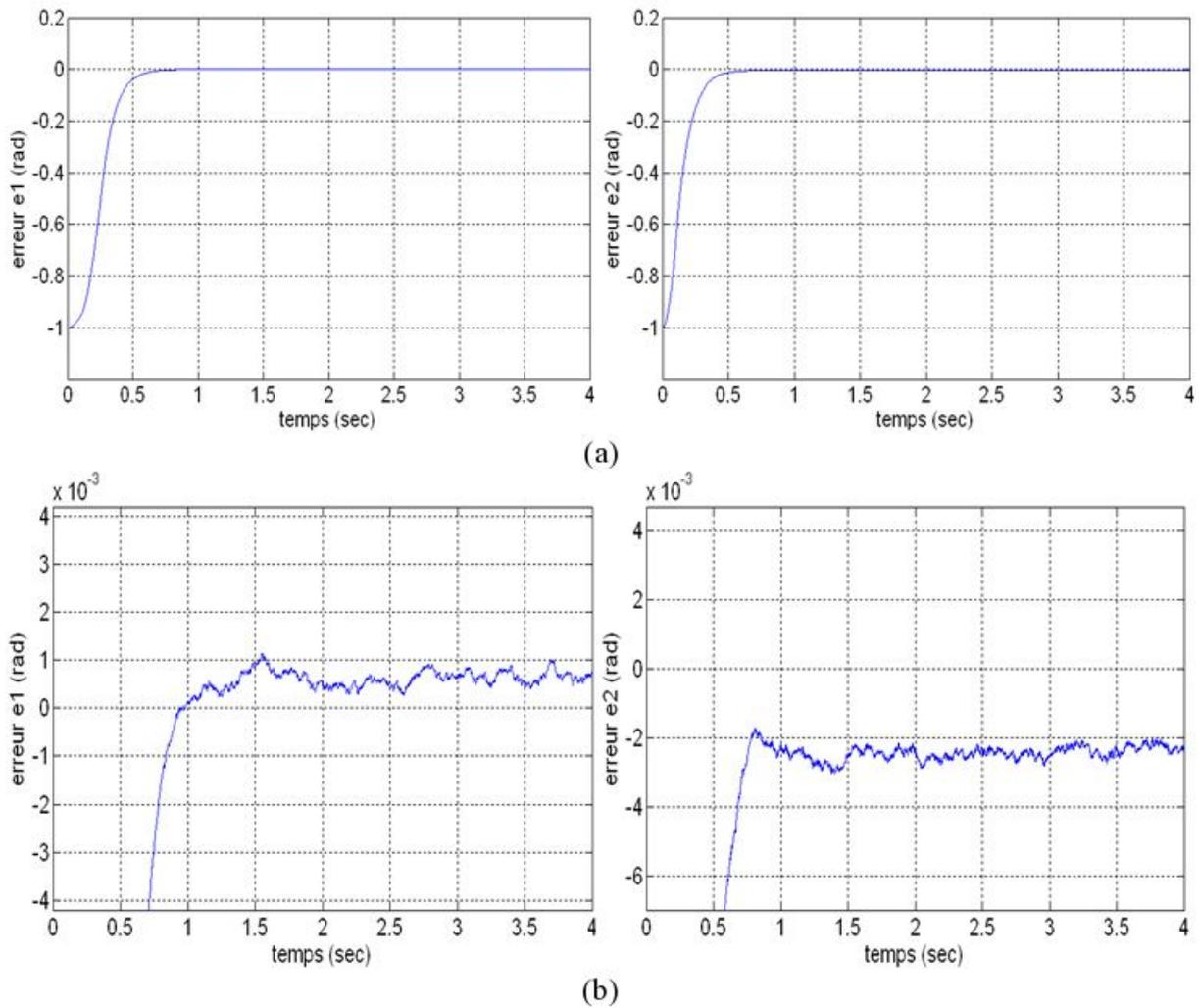


Figure II.7: (a), (b) Résultat de l'erreur de suivi de la trajectoire désirée et son zoom pour les deux articulations -cas de régulation-

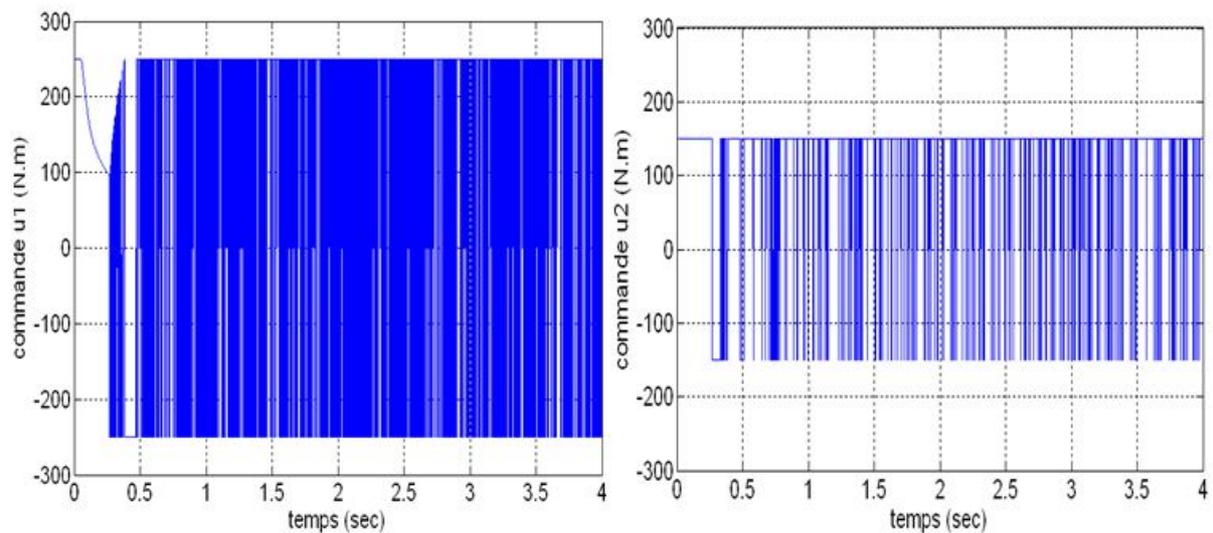


Figure II.8 : Résultat des couples générés pour le suivi de la trajectoire désirée pour les deux articulations -cas de régulation-

b) poursuite :

La trajectoire désirée pour la régulation des deux articulations est:

$$q_{d1,2} = \sin\left(\pi t + \frac{3\pi}{2}\right) \text{ rad.}$$

Les valeurs des paramètres utilisés pour la régulation :

$$k_p = \begin{bmatrix} 160 & 0 \\ 0 & 140 \end{bmatrix}, k_d = \begin{bmatrix} 31 & 0 \\ 0 & 29 \end{bmatrix}, k_i = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}, k_c = \begin{bmatrix} 300 & 0 \\ 0 & 250 \end{bmatrix}$$

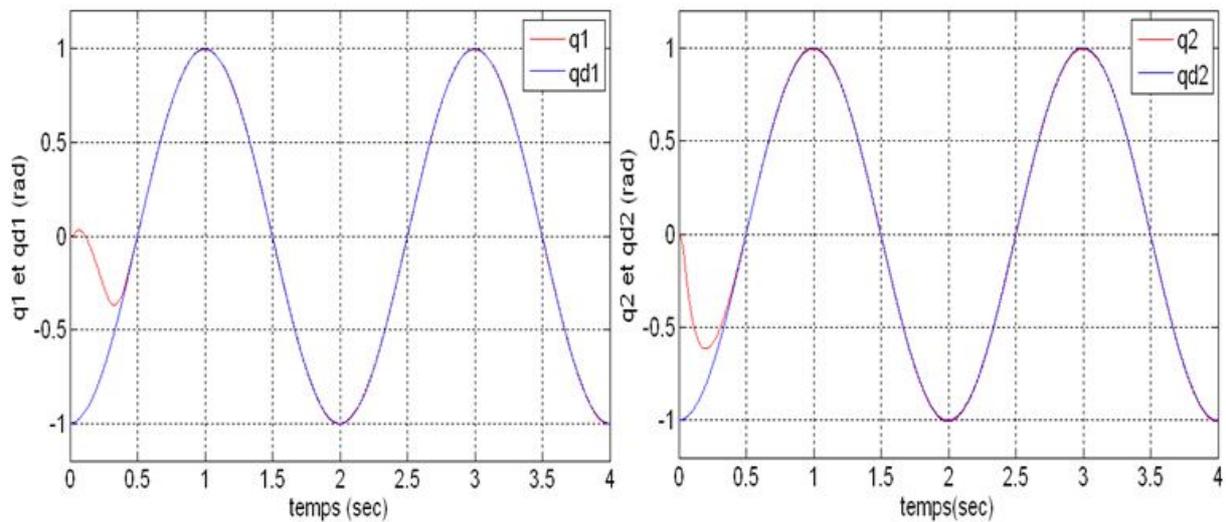


Figure II.9 : Résultat de suivi de la trajectoire désirée pour les deux articulations
-cas de poursuite-

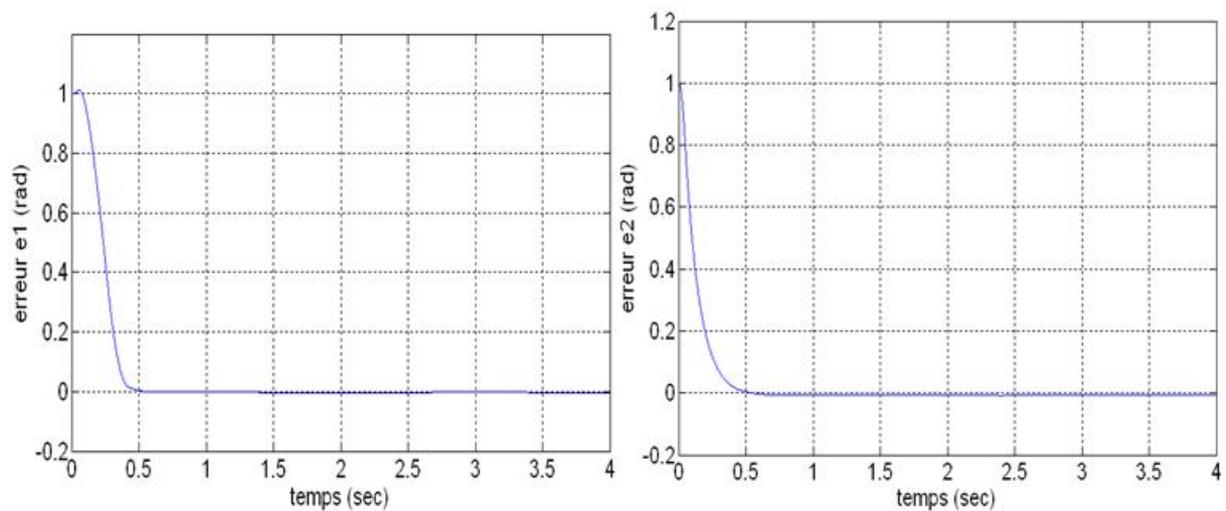


Figure II.10 : Résultat de l'erreur de suivi de la trajectoire désirée pour les deux articulations
-cas de poursuite-

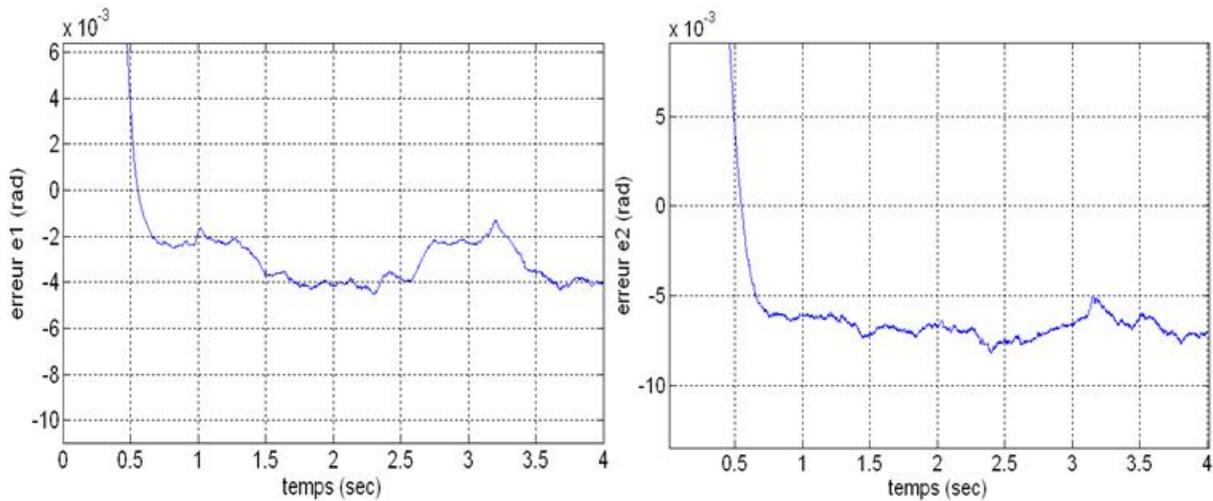


Figure II.11 : Zoom sur l'erreur de suivi de la trajectoire désirée pour les deux articulations
-cas de poursuite-

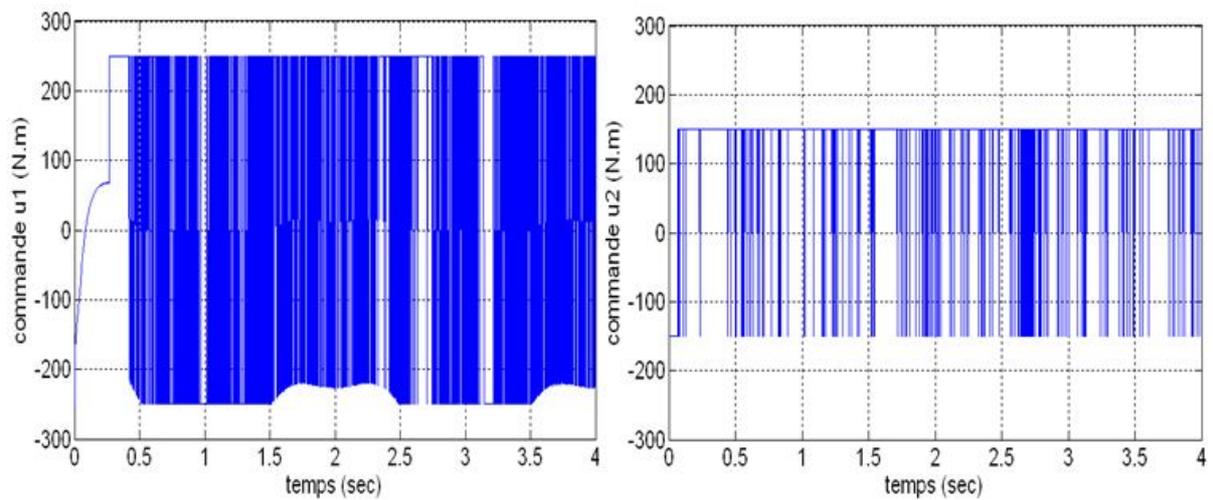


Figure II.12 : Résultat des couples générés pour le suivi de la trajectoire désirée pour les deux articulations - cas de poursuite-

II.5.2 Commande par backstepping :

a) régulation :

La trajectoire désirée pour la régulation des deux articulations est:

$$q_{d1,2} = u(t) = \begin{cases} 1 & \text{rad pour } t \geq 0 \\ 0 & \text{ailleurs} \end{cases}$$

Les valeurs des paramètres utilisés pour la régulation :

$$k_1 = \begin{bmatrix} 30 & 0 \\ 0 & 105 \end{bmatrix}, k_2 = \begin{bmatrix} 21 & 0 \\ 0 & 75 \end{bmatrix}$$

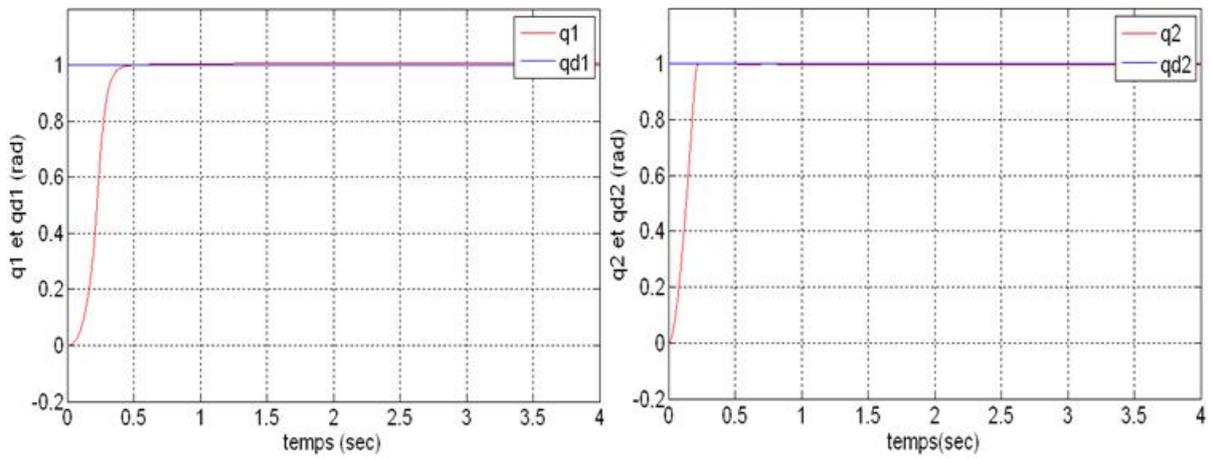
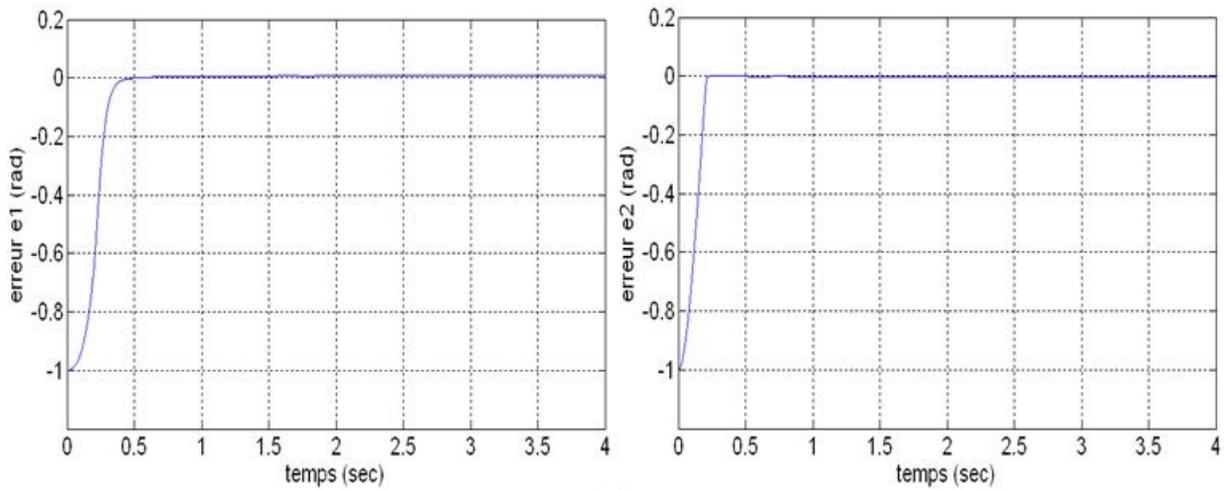
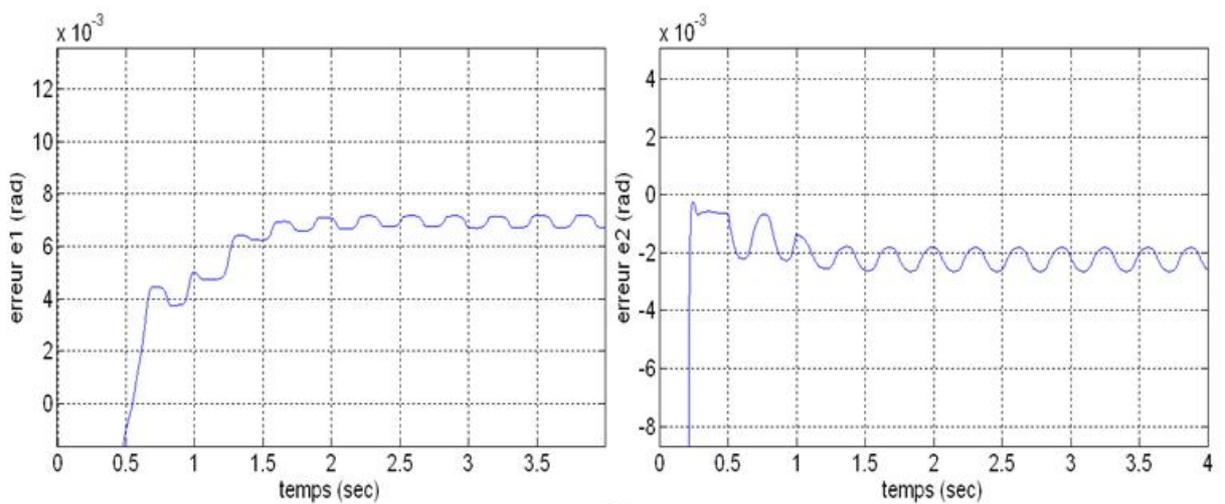


Figure II.13 : Résultat de suivi de la trajectoire désirée pour les deux articulations - cas de régulation-



(a)



(b)

Figure II.14: (a), (b) Résultat de l'erreur de suivi de la trajectoire désirée et son zoom pour les deux articulations -cas de régulation-

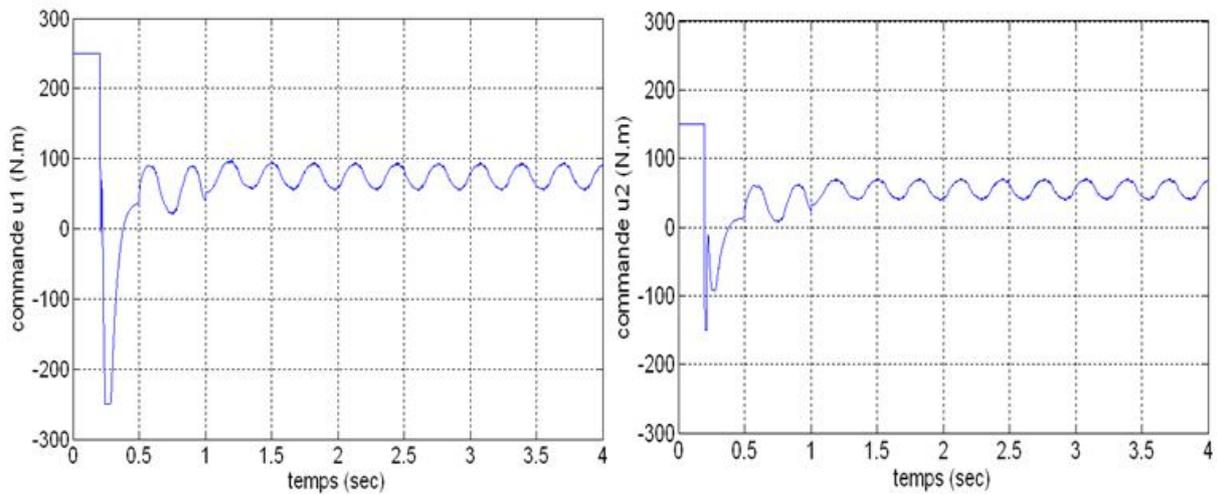


Figure II.15 : Résultat des couples générés pour le suivi de la trajectoire désirée pour les deux articulations -cas de régulation-

b) poursuite :

La trajectoire désirée pour la régulation des deux articulations est:

$$q_{d1,2} = \sin\left(\pi t + \frac{3\pi}{2}\right) \text{ rad.}$$

Les valeurs des paramètres utilisés pour la régulation :

$$k_1 = \begin{bmatrix} 60 & 0 \\ 0 & 100 \end{bmatrix}, k_2 = \begin{bmatrix} 2 & 0 \\ 0 & 10 \end{bmatrix}$$

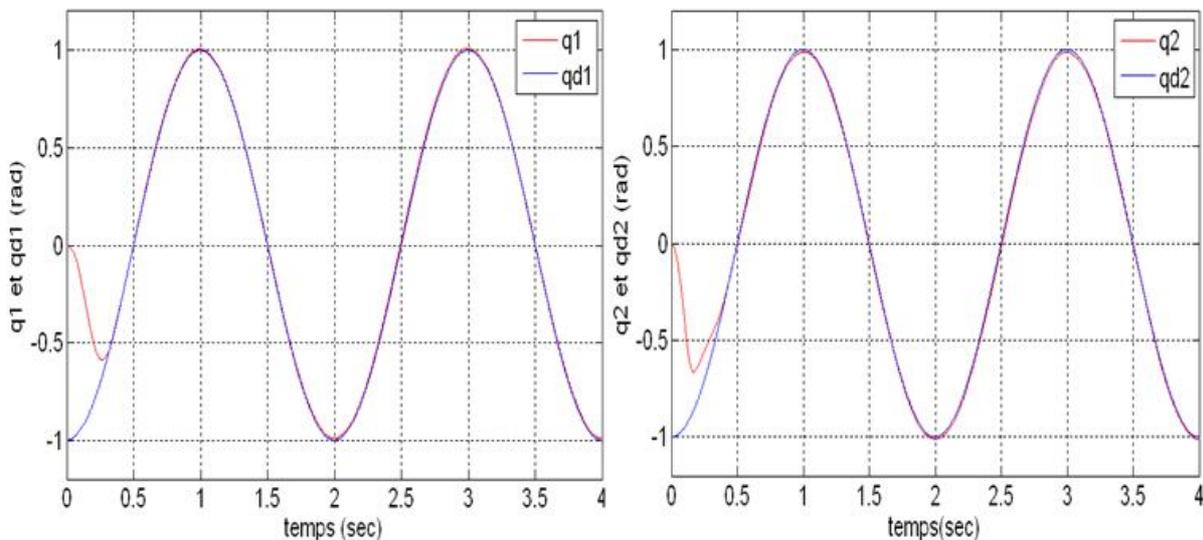


Figure II.16: Résultat de suivi de la trajectoire désirée pour les deux articulations -cas de poursuite-

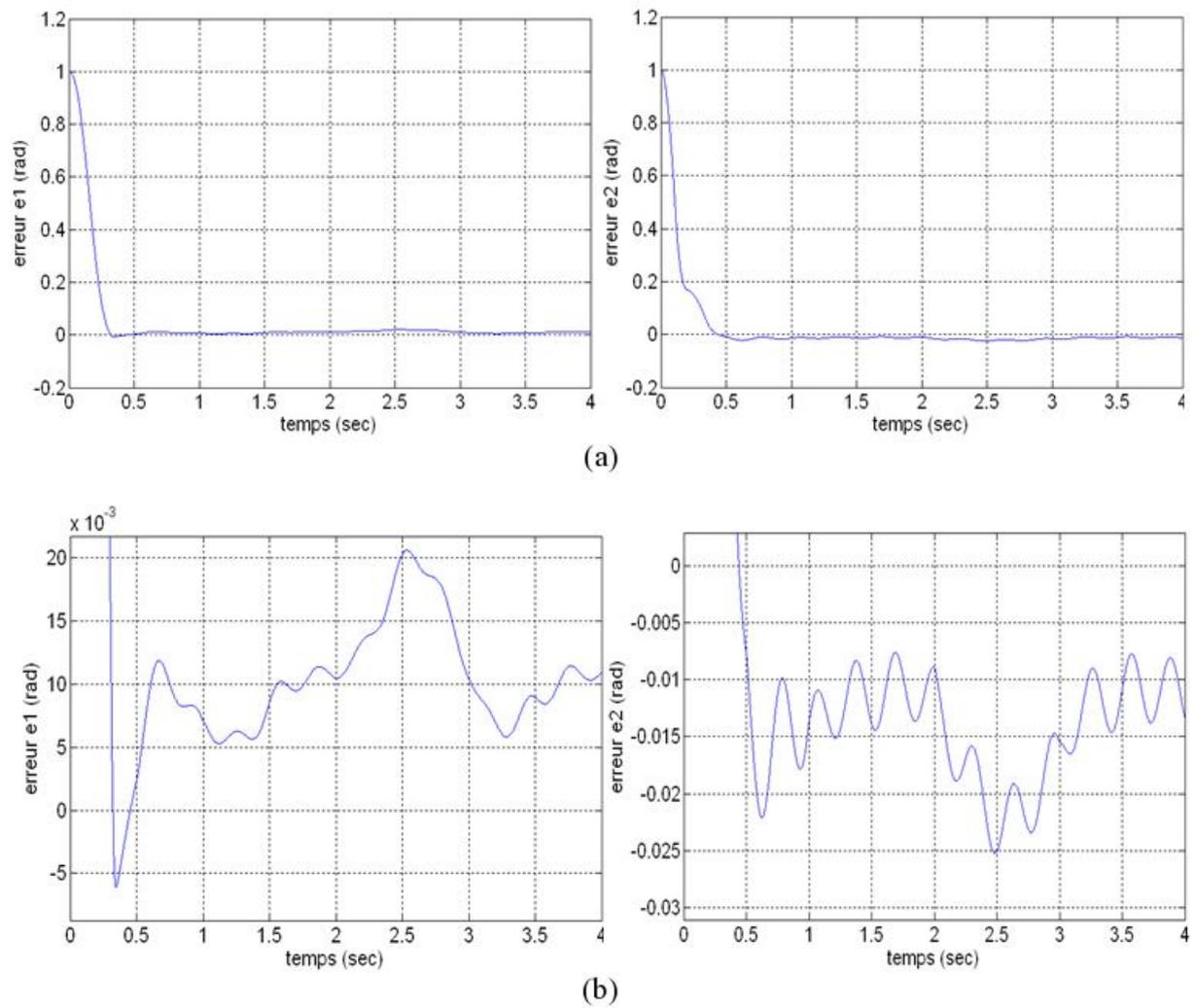


Figure II.17: (a), (b) Résultat de l'erreur de suivi de la trajectoire désirée et son zoom pour les deux articulations -cas de poursuite-

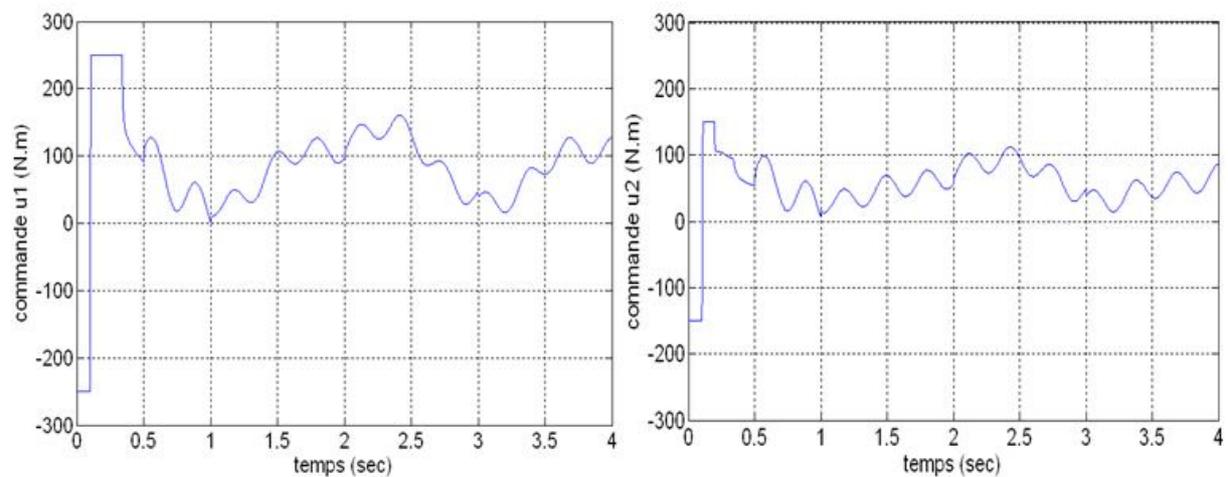


Figure II.18 : Résultat des couples générés pour le suivi de la trajectoire désirée pour les deux articulations -cas de poursuite-

Les figures II.6 à II.12, montrent bien l'efficacité de la commande par mode glissant, qui donne des bons résultats même dans la présence d'incertitudes et de perturbations. On voit clairement que le signal de commande de cette méthode se présente en une allure alternative d'une certaine fréquence, ce phénomène est appelé réticence [15, 18] (chattering en anglais).

Les figures II.13 à II.18, représentent les résultats de simulation de la commande backstepping qui présente un comportement acceptable.

II.6 Conclusion :

Dans ce chapitre, nous avons fait une description théorique des deux algorithmes de commandes, voir le backstepping et la commande par mode glissant, tout en montrant leur robustesse vis-à-vis les perturbations, ensuite on les a appliqué sur un bras manipulateur à 2 ddl avec un modèle incertain et dans un environnement perturbé. Les résultats de simulation des deux commandes ont montré leurs efficacités dans de telles situations, cependant on remarque la présence de quelque inconvénient présentés par le phénomène de réticence « chattering » qui caractérise la commande par mode glissant et qui représente un vrai problème lorsque sa fréquence est élevée, dont il faut minimiser son effet, et pour cela on va introduire quelques outils qu'on va voir dans les chapitres qui suit.

Chapitre III

Commande par Mode glissant flou et flou adaptatif

III.1 Introduction :

La commande basée sur la théorie des structures variables (CSV) est caractérisée par un contrôleur discontinu qui peut changer sa structure automatiquement pour atteindre une surface de glissement afin d'obtenir la dynamique désirée. Issue de la théorie des structures variables, la commande par mode glissant (en anglais sliding mode control « SMC ») est l'une des approches non linéaires effectives de contrôle robuste puisqu'elle fournit à la dynamique du système une propriété d'invariance aux incertitudes, une fois que la dynamique est contrôlée en mode glissant. Cependant, son inconvénient majeur dans les applications pratiques est le problème de réticence [15, 18] « chattering » qui est un zigzag autour de la surface de glissement. Pour minimiser l'effet de ce phénomène beaucoup de propositions ont été élaborées, comme faire remplacer la fonction discontinue singe par des approximations continues comme la fonction de saturation [15] ou la fonction tangente hyperbolique ou sigmoïde. [1]

Devant ces inconvénients et avec l'efficacité de cette méthode, des études sont faites pour améliorer l'algorithme de la commande par mode glissant en introduisant les outils de l'intelligence artificielle. La logique floue a été ainsi utilisée de plusieurs manières [1,14, 22, 39] pour obtenir un contrôleur qui combine la puissance de la technique SMC et de la logique floue.

Dans ce chapitre on va faire introduire quelque concept de la logique floue, ensuite on va procéder à l'élaboration du contrôleur à mode glissant flou et flou adaptatif.

III.2 La logique floue :

La logique floue est un ensemble de concepts mathématiques simples qui permet de résoudre des problèmes complexes là où les outils de résolution conventionnelle sophistiquée peinent. Son succès réside principalement dans sa capacité à résoudre les différents défis existant dans le monde réel, d'où les problèmes de modélisation, de contrôle, de diagnostic ou d'aide à la décision sont des clients potentiels de la logique floue. Ainsi, La logique floue s'est imposée dans des domaines aussi variés que l'industrie, la robotique, l'automatisme, l'espace, la gestion de la circulation routière, le contrôle aérien, l'environnement, la médecine, afin de résoudre les problèmes d'identification, de régulation de processus, d'optimisation, de classification, de détection de défauts ou de prise de décision

III.2.1 Systèmes et contrôleurs flous:

III.2.1.1 Ensembles flous :

La logique floue repose sur la théorie des ensembles flous, qui est une théorie mathématique dont l'objectif principal est la modélisation des notions vagues et incertaines du langage naturel [13,20]. Cette théorie permet d'exprimer l'idée d'une appartenance partielle d'un élément à un ensemble (Figures III.1, (b)). La notion d'appartenance stricte est mal adaptée lorsqu'on manipule:

- Des données incertaines dont la validité est entachée de doute.
- Des données imprécises dont l'expression verbale est difficile.

La théorie ordinaire des ensembles considère des collections d'éléments regroupés en ensembles. L'appartenance d'un élément à un ensemble est une question non ambiguë dont la réponse est soit oui soit non (ensemble booléen) voir Figure II.1, (a). La relation d'appartenance est représentée dans ce cas par une fonction μ qui prend des valeurs de vérité dans la paire $[0,1]$. Cependant, dans le cas réel, on rencontre fréquemment des situations où l'ambiguïté est de règle, particulièrement dans le langage courant. Un ensemble flou sera alors utilisé pour modéliser l'incertitude et les imprécisions dans la connaissance.

Un ensemble flou A est défini sur un ensemble de valeur appelé l'univers du discours X par une fonction d'appartenance $\mu_A(x)$ comprise généralement entre 0 et 1 et qui quantifie le degré avec lequel chaque élément x de X appartient à A .

$$\mu_A(x) = 0 \Rightarrow x \notin A$$

$$\mu_A(x) = 1 \Rightarrow x \in A$$

$$A = \{(\mu_A(x) / x : x \in X, \mu_A(x) \in [0,1] \subset R\} \quad (III.1)$$

L'équation (III.1) définit un ensemble flou comme complètement défini par sa fonction d'appartenance $\mu_A(x)$:

$$\mu_A(x): X \rightarrow [0,1] \quad (III.2)$$

Si la fonction d'appartenance ne donne que les valeurs de l'ensemble $\{0,1\}$, alors l'ensemble qu'elle génère n'est pas flou.

Dans le cas d'un ensemble discret U , un ensemble flou A peut être défini par une liste de paires ordonnées : degré d'appartenance / élément de l'ensemble :

$$A = \sum_i \mu_A(x) / x_i \quad (III.3)$$

Où $\mu_A(x_i)$ avec $i = 1, \dots, n$ représente le degré d'appartenance de x_i à l'ensemble flou A .

Dans le cas d'un ensemble continu U , un ensemble flou A peut être représenté par :

$$A = \int \mu_A(x) / x \quad (III.4)$$

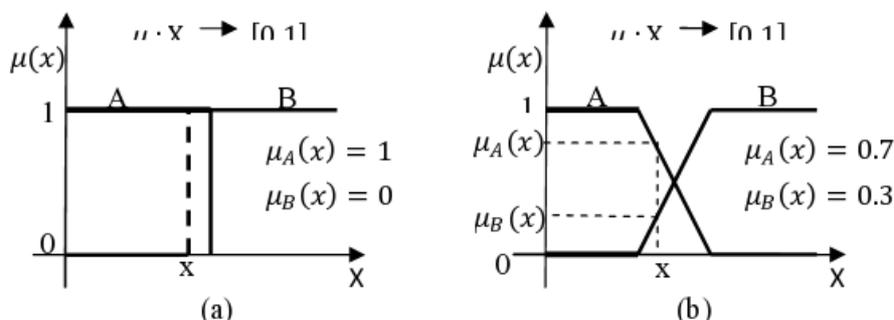


Figure III.1 : (a) ensemble booléen, (b) ensemble flou

Selon le type de la fonction d'appartenance, différents types d'ensembles flous seront obtenus. Les fonctions d'appartenance pourraient être classées en deux groupes : celles qui sont faites de lignes droites, Figure II.2 : (a) et (b) et celles dont la forme est gaussienne, Figure II.2 : (c) et (d).

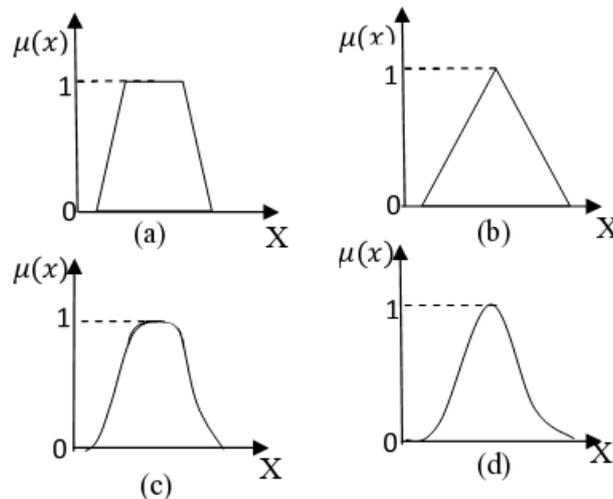


Figure III.2 : Différent type de fonction d'appartenance

(a) Trapezoidal, (b) Triangulaire, (c) Gaussienne, (d) Gaussienne Généralisé

Plusieurs opérations peuvent être appliquées aux ensembles flous. Soient A et B deux ensembles flous de fonctions d'appartenance respectives $\mu_A(x)$ et $\mu_B(x)$ et leur univers de discours.

- **Intersection** : La fonction d'appartenance de l'intersection de deux ensembles est exprimée par : $\mu_{A \cap B}(x) = g(\mu_A(x), \mu_B(x))$, $x \in X$ où g est un opérateur t-norme.
- **Union** : La fonction d'appartenance de l'union de deux ensembles est exprimée par : $\mu_{A \cup B}(x) = f(\mu_A(x), \mu_B(x))$, $x \in X$ où f est un opérateur t-conorme.
- **Complément** : La fonction d'appartenance du complément de l'ensemble flou est définie par : $\forall x \in X : \mu_{\bar{A}}(x) = 1 - \mu_A(x)$

III.2.1.2 Variable linguistique :

Une variable linguistique est une variable dont les valeurs sont des mots ou des phrases exprimés dans un langage naturel. Elle est définie par le triplet suivant [6]: (V, X, T_V) où V est le nom de la variable linguistique, X est le domaine physique associé à la variable V appelée

aussi l'univers de discours et T_V est l'ensemble des caractéristiques floues de la variable V c'est à dire l'ensemble des valeurs linguistiques que peut prendre la variable V . La définition d'une variable linguistique regroupe donc à la fois des informations de nature numérique et linguistique.

III.2.1.3 Le Contrôleur flou [6, 20]:

Dans la théorie de commande classique on emploie toujours des procédés analytiques pour la description du comportement du système et du contrôleur, tandis que les systèmes flous sont convenables à la modélisation de la connaissance vague, par exemple l'expérience de l'être humain. Cette différence principale résulte des méthodologies différentes pour résoudre un problème de contrôle donné. D'où un contrôleur flou est constitué de 4 parties principales: la base de connaissance, le système d'inférence, l'interface de fuzzification et l'interface de défuzzification.

Et le contrôleur flou passe généralement par les étapes suivantes:

- Choix de la stratégie de fuzzification.
- Etablissement de la base de règles.
- Choix de la méthode d'inférence.
- Choix de la stratégie de défuzzification.

❖ Fuzzification : Les variables d'entrée et de sortie choisies pour modéliser ou commander un système sont des grandeurs numériques. L'étape de fuzzification consiste à transformer ces grandeurs réelles en variables linguistiques en vue d'un traitement d'inférence. Ainsi, à chaque variable d'entrée et de sortie on associe des ensembles caractérisant les termes linguistiques pris par ces variables. Ces termes seront utilisés pour écrire les règles d'inférence.

Le choix des formes des fonctions d'appartenance est arbitraire. Des études comparatives ont montré qu'avec les différentes formes des fonctions d'appartenance, les résultats sont pratiquement similaires en boucle fermée [6]. La forme la plus fréquemment utilisée en commande floue est la forme triangulaire. Le nombre de fonctions d'appartenance est généralement impair et se répartissant autour de zéro. En général, on introduit pour une variable linguistique trois, cinq ou sept ensembles flous. Le choix du nombre dépend de la

précision souhaitée. Les fonctions d'appartenance peuvent être symétriques, non symétriques et équidistantes ou non équidistantes.

❖ La base de règles : Une base de règles floues est une collection de règles qui permet de lier les variables floues d'entrée et de sortie. La description de la commande se fait par l'intermédiaire de ces règles qui ont la forme suivante :

$$\text{Si } x_1 \text{ est } A_i \text{ et } x_2 \text{ est } A_2 \text{ Alors } y \text{ est } B \quad (\text{III. 5})$$

Où x_1, x_2 sont les grandeurs physiques caractéristiques du système. A_1, A_2 sont les termes linguistiques. B a deux possibilités, si elle est une valeur linguistique, le contrôleur est dit de type Mamdani et si est une valeur numérique ou une équation mathématique, alors le contrôleur est dit de type Takagi-Sugeno [20]. Le ET de conjonction est réalisé en effectuant le minimum entre les degrés de vérité des propositions floues $x_1 \text{ est } A_i \text{ et } x_2 \text{ est } A_2$.

❖ Méthode d'inférence floue : Elle permet de calculer l'ensemble flou associé à la commande qui se fait par les opérations d'inférence floue et l'agrégation des règles.

L'inférence floue repose sur l'utilisation d'un opérateur d'implication floue pour chaque règle à analyser. Cet opérateur quantifie la force de liaison entre la prémisse et la conclusion de la règle. Soit la règle suivante : *Si x est A Alors y est B* , l'inférence peut être exprimée mathématiquement par l'expression suivante : $\mu'_B(y) = I(\mu_A(x_0), \mu_B(y))$ où désigne I l'opérateur d'inférence.

Il existe d'autres possibilités pour exprimer les inférences, à savoir par description linguistique, par matrice d'inférence ou par tableau d'inférence. Deux approches d'inférence sont couramment utilisées :

- Implication de Mamdani : $\mu'_B(y) = \min(\mu_A(x_0), \mu_B(y))$
- Implication de Larsen : $\mu'_B(y) = \mu_A(x_0) \mu_B(y)$

Pour générer une conclusion à partir de l'ensemble des règles actives, on procède à une agrégation de ces règles par un opérateur disjonctif. Ce qui revient à lier les règles par un opérateur OU. Généralement l'opérateur est utilisé pour agréger un ensemble de n règles :

$$\mu_B(y) = \max_{i=1 \dots n} \mu_{B_i}(y) \quad (\text{III. 6})$$

❖ **Défuzzification** : Le traitement des règles d'inférence fournit une valeur floue. L'étape de défuzzification consiste à transformer l'ensemble flou résultant de l'agrégation des règles en une grandeur de commande précise à appliquer au processus. Dans la littérature, il existe plusieurs stratégies pour réaliser cette opération telle que la moyenne des maxima, le centre des aires, le centre des maxima. La méthode de défuzzification par le centre de gravité est la méthode la plus utilisée en commande floue du fait qu'elle fournit intuitivement la valeur la plus représentative de l'ensemble flou issu de l'agrégation des règles. Elle consiste à calculer le centre de gravité de la surface formée par la fonction d'appartenance résultante. Dans le cas particulier de règles de Takagi-Sugeno où les conclusions sont polynomiales :

$$R^l : \text{Si } x_1 \text{ est } A_1 \text{ et } x_2 \text{ est } A_2 \text{ et } \dots \text{ et } x_n \text{ est } A_n \text{ Alors } y = f^l(x_1, x_2, \dots, x_n) \quad (III. 7)$$

$$l = 1, 2, \dots, K$$

La commande u est obtenue par une simple moyenne pondérée selon les niveaux d'activation w^l de chacune des règles.

$$u = \frac{\sum_{l=1}^K w^l f^l(x_1, x_2, \dots, x_n)}{\sum_{l=1}^K w^l} \quad (III. 8)$$

Avec $w^l = T(\mu_{A_1^l}(x_1), \mu_{A_2^l}(x_2), \dots, \mu_{A_n^l}(x_n))$ où T est une t-norme choisit très souvent égale à l'opérateur produit

III.3 Commande par mode glissant flou:

La commande floue est le domaine d'application de la logique floue le plus utilisé dans le cadre de la commande des systèmes non linéaires complexes. L'approche qu'on va utiliser (fuzzy sliding mode control en anglais) repose sur la propriété d'approximation universelle des systèmes flous [13,20]. Cette approche de commande floue a pour objectif de se bénéficier des avantages de la logique floue et de la commande par mode glissant (SMC) afin de garantir un comportement robuste du robot manipulateur. Le deuxième avantage de l'arrangement proposé est d'améliorer les performances du système en minimisant l'effet du chattering. [1, 14, 22, 39]

III.3.1 Conception du contrôleur (FSMC) :

En se basant sur la particularité de la logique floue de modéliser les notions vagues et incertaines du langage naturel, on va convertir la stratégie de commande par mode glissant en relations entre des ensembles flous en faisant une approximation du comportement de la partie discrète de l'algorithme du mode glissant classique. Ainsi un mécanisme d'inférence floue est utilisé et la méthodologie du contrôleur par mode glissant flou (FSMC) est définie. [1]

L'architecture (manipulateur, contrôleur) du système complet est montrée dans la figure (III.3), la commande se divise en deux parties, une commande équivalente identique à la commande équivalente du mode glissant classique (II.65) et une autre partie représentant la loi de commande qui assure le mode de convergence, elle est de la forme [1, 22]:

$$u_c = k_f u_f \quad (III.9)$$

Où k_f est le facteur de normalisation de la variable de sortie de dimensions 2×2 et u_f est le vecteur de sortie du FSMC de dimensions 2×1 .

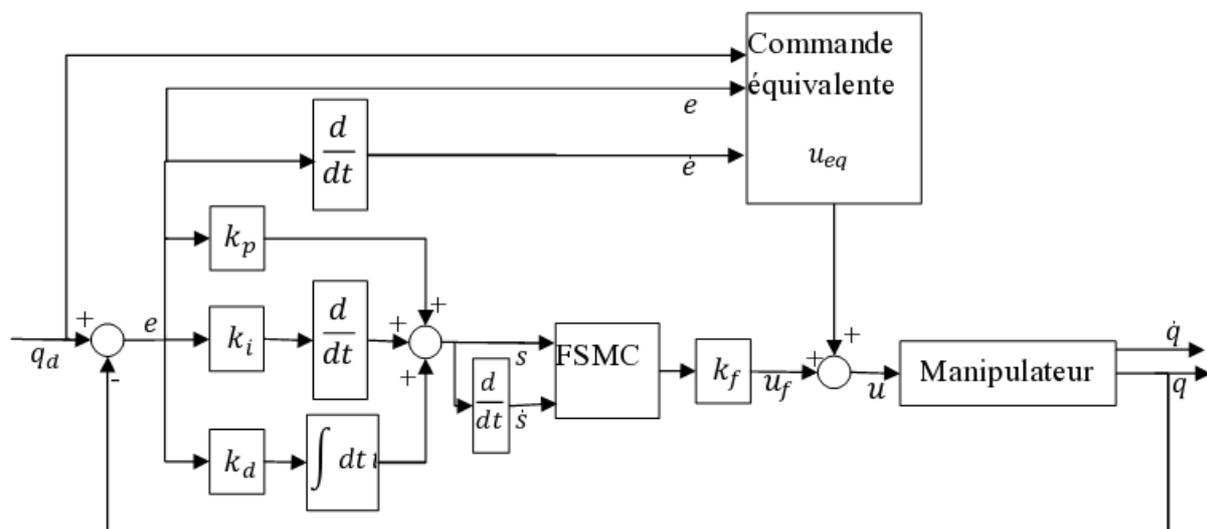


Figure III.3 : schéma block de la commande par mode glissant flou adaptatif (FSMC) avec une surface PID.

L'expression de la surface utilisée dans ce contrôleur est la même que celle utilisée dans le chapitre II (équation II.61). L'expression de notre contrôleur flou est alors: [22]

$$u_f = FSMC(s, \dot{s}) = [u_{f1}, u_{f2}, \dots, u_{fn}] \quad (III.10)$$

$$u_f = [FSMC(s_1, \dot{s}_1), FSMC(s_2, \dot{s}_2), \dots, FSMC(s_n, \dot{s}_n)] \quad (III.11)$$

Et chaque surface s_i correspond à une variable généralisée q_i , ça veut dire qu'on a n surfaces pour n variables généralisées contrôlées par n contrôleurs flou. Pour un bras manipulateur à 2 ddl on aura $n=2 \Rightarrow u_f = [FSMC(s_1, \dot{s}_1), FSMC(s_2, \dot{s}_2)]$. La conception de la partie floue du contrôleur suit les étapes suivantes :

III.3.1.1 Fuzzification :

Le contrôleur proposé a deux entrées et une sortie, ce sont s, \dot{s} et le signal de commande u_f , respectivement. On a utilisé les mêmes variables linguistiques pour les trois ensembles flous $\{(s, \mu(s)), (\dot{s}, \mu(\dot{s})), (u_f, \mu(u_f))\}$. [1, 22]

Les variables linguistiques utilisés sont :

NG: négatif grand, NM : négatif moyen, NP : négatif petit, Z : zéros, PP : positif petit
PM : positif moyen, PG : positif grand.

Les fonctions d'appartenances des entrées linguistiques pour chaque ensemble de variables s, \dot{s} et u_f sont normalisées [1, 22] comme le représente les Figures III.4 et III.5, en utilisant les constantes a_1 et a_2 comme facteur de mise à l'échelle pour s et \dot{s} respectivement.

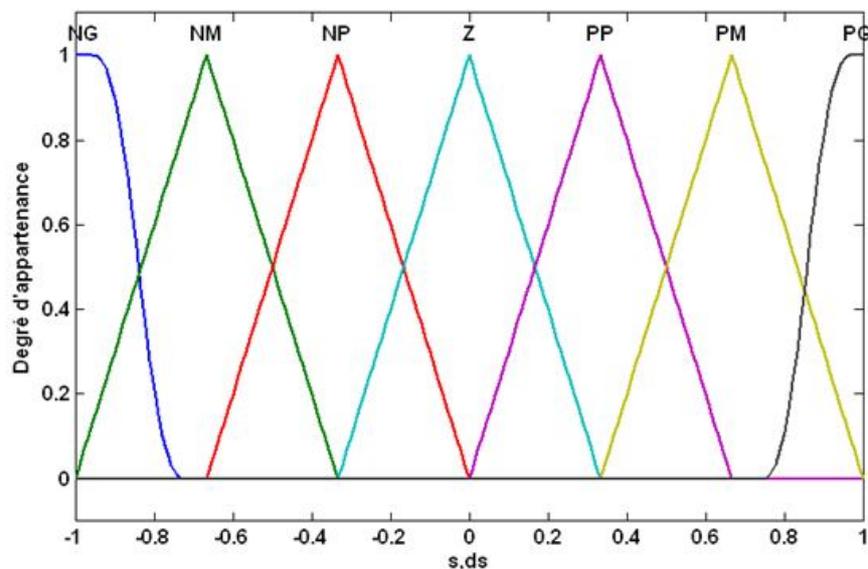


Figure III.4 : Les fonctions d'appartenances des variables linguistiques de s, \dot{s} .

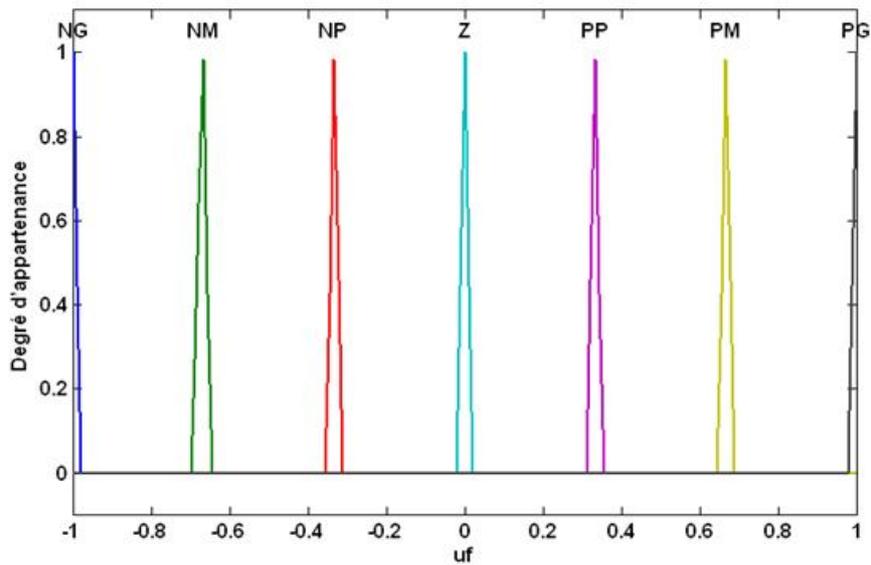


Figure III.5: Les fonctions d'appartenance des variables linguistiques de u_f .

III.3.1.2 La base de règles:

Les règles du contrôle flou peuvent être représentées en tant qu'ensembles de relations qui lient les variables linguistiques d'entrée s et \dot{s} aux variables linguistiques de sortie u_f comme suit :

$$R^{(l)} :: \text{Si } s(t) \text{ est } A_1^l \text{ et } \dot{s}(t) \text{ est } A_2^l \text{ ALORS } u_f \text{ est } B^l \quad (\text{III.12})$$

Tel que A_1^l, A_2^l et B^l représente les fonctions d'appartenance des ensembles flous.

La base de règles du tableau III.1 [22] a été conçue de façon à respecter la condition suivante :

lorsque la multiplication de s et de \dot{s} est positive, la fonction de sortie u_f va être réglé de telle manière que son signe soit l'opposé de celui de s ; ce qui donne : $su_f \leq -|s|$.

Les règles floues contiennent les relations qui définissent la stratégie de commande, elles sont extraites de telle manière que la stabilité du système soit satisfaite [1, 22]. Chaque entrée du contrôleur a sept ensembles flous de sortie, ce qui nous donne 49 règles floues.

		s_i						
		PG	PM	PP	Z	NP	NM	NG
\dot{s}_i	PG	NG	NG	NG	Z	Z	Z	Z
	PM	NG	NG	NG	Z	Z	Z	PP
	PP	NG	NG	NM	Z	Z	PP	PM
	Z	NG	NM	NP	Z	PP	PM	PG
	NP	NM	NP	Z	Z	PM	PG	PG
	NM	NP	Z	Z	Z	PG	PG	PG
	NG	Z	Z	Z	Z	PG	PG	PG

Tableau III.1 [22]

III.3.1.3 Inférence et défuzzification :

Un mécanisme d'inférence de Mamdani est utilisé et la fonction minimum d'intersection a été employée pour l'implication floue. Dans la défuzzification c'est la méthode du centre de gravité qui a été employée pour calculer les valeurs de sortie.

Le signal de commande globale est :

$$u = u_{eq} + u_c = u_{eq} + k_f u_f \tag{III.13}$$

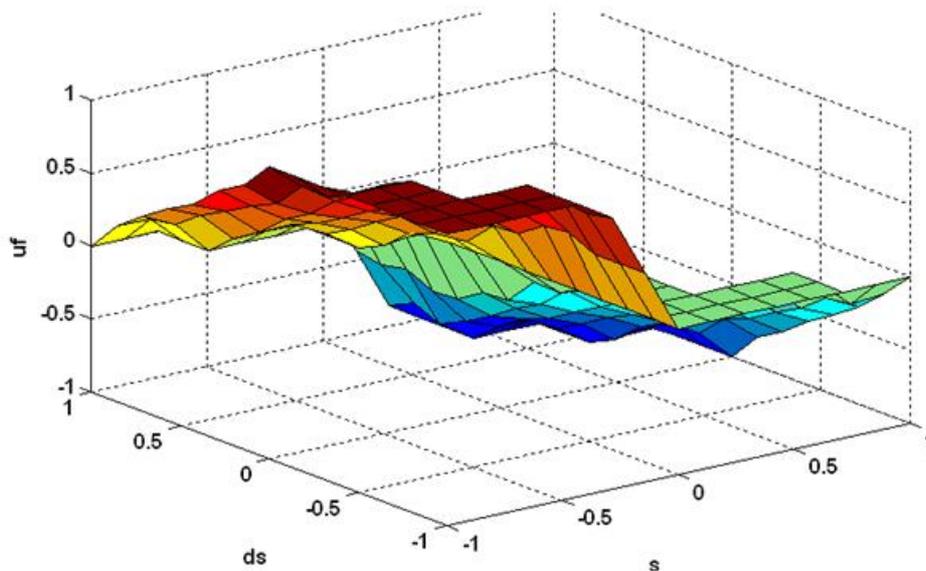


Figure III.6: La surface commande de u_f

II.3.2 Analyse de stabilité :

On considère le modèle présenté par l'équation (II.57), et on choisit la surface de glissement s tel que :

$$s(t) = k_p e(t) + k_i \int e(\xi) d\xi + k_d \frac{d}{dt} e(t) \quad (III.14)$$

$$\dot{s}(t) = k_p \dot{e} + k_i e + k_d \ddot{e} \quad (III.15)$$

$$\dot{s}(t) = k_p \dot{e} + k_i e + k_d [\ddot{q}_d - \ddot{q}] \quad (III.16)$$

$$= k_p \dot{e} + k_i e + k_d [\ddot{q}_d - g_a u - f_a - b(t)] \quad (III.17)$$

On choisit comme fonction de Lyapunov, la fonction $V = \frac{1}{2} s^2$, d'où sa dérivé est :

$$\dot{V} = s\dot{s} = s[k_p \dot{e} + k_i e + k_d \ddot{e}] \quad (III.18)$$

$$\dot{V} = s[k_p \dot{e} + k_i e + k_d [\ddot{q}_d - g_a u - f_a - b(t)]] \quad / u = u_{eq} + u_c \quad (III.19)$$

$$= s[-k_d g_a u_f - k_d b(t)] \quad / u_{eq} = (k_d g_a)^{-1} [k_p \dot{e} + k_i e - k_d f_a + k_d \ddot{q}_d] \quad (III.20)$$

$$= s[-k_d g_a k_f FSMC(s, \dot{s}) - k_d b(t)] \quad (III.21)$$

$$\leq -k_d g_a k_f s FSMC(s, \dot{s}) + k_d |b(t)| |s| \leq -k_d g_a k_f |s| + k_d \alpha |s| \quad (III.22)$$

Avec α représente l'extrémité des bornes d'incertitudes et de perturbations : $|b(t)| < \alpha$

Si on choisit $k_f > g_a^{-1} \alpha$, alors la commande va assurer la convergence de l'état vers la surface de glissement, ainsi le système bouclé est asymptotiquement stable, et l'erreur va converger vers zéros.

III.4 Commande par mode glissant flou adaptatif :

Le gain fixe de la méthode de commande par mode glissant flou (FSMC) impose d'avoir une certaine connaissance sur les bornes des incertitudes et des perturbations qui peuvent survenir, ce qui est difficile ; Pour résoudre ce problème un superviseur flou a été ajouté [1]

au contrôleur précédent pour ajuster le gain k_f suivant la valeur de l'erreur pour alléger les contraintes posées sur la connaissance des bornes .

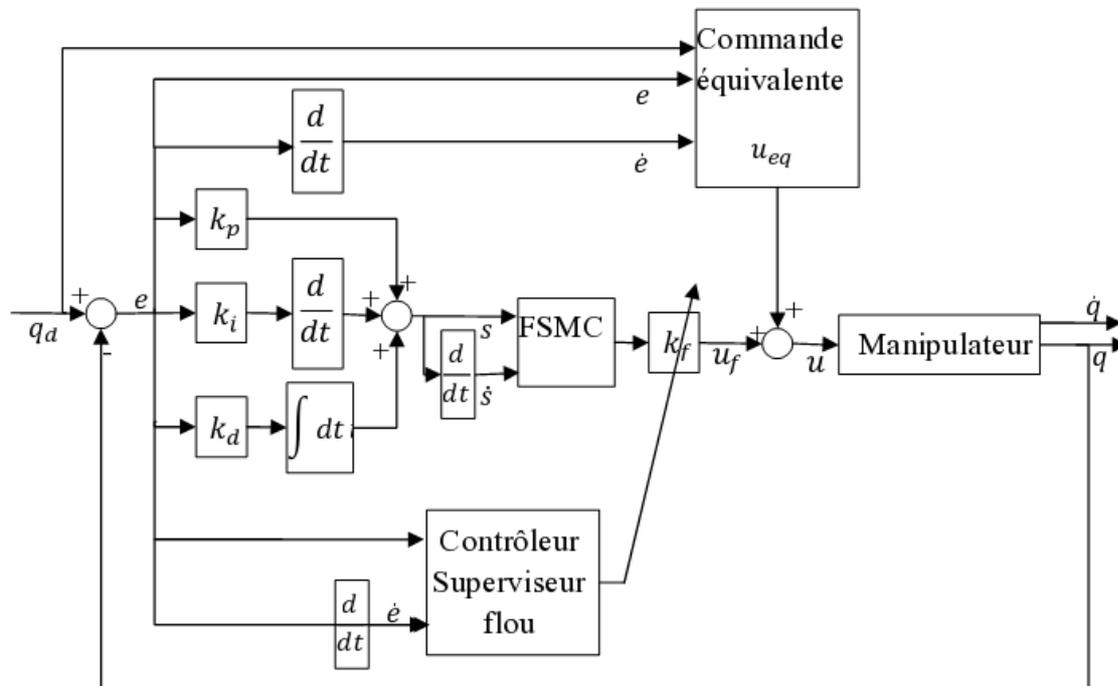


Figure III.7 : schéma block de la commande par mode glissant flou adaptatif (AFSMC) avec une surface PID.

III.4.1 Conception du contrôleur (AFSMC) :

En partant de l'architecture précédente, le contrôleur par mode glissant flou adaptatif (adaptive fuzzy sliding mode control) conserve le même contrôleur flou avec l'ajout d'un système de surveillance flou [1], afin d'ajuster de manière adaptative le gain de la commande de convergence k_f dans le but d'améliorer le rendement du Contrôleur. La structure générale du Contrôleur proposé est donnée dans (Figure III.7). La conception du superviseur déroule comme suite :

III.4.1.1 Fuzzification :

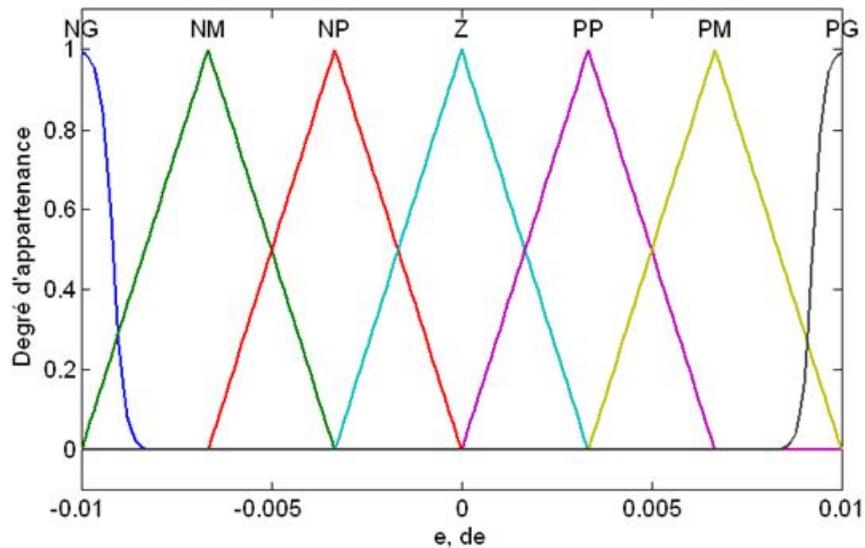
Le superviseur proposé a deux entrées et une sortie: e, \dot{e} et le signal de commande k_f , respectivement. Les variables linguistiques proposées sont exprimés comme [1]:

1) - pour les entrées :

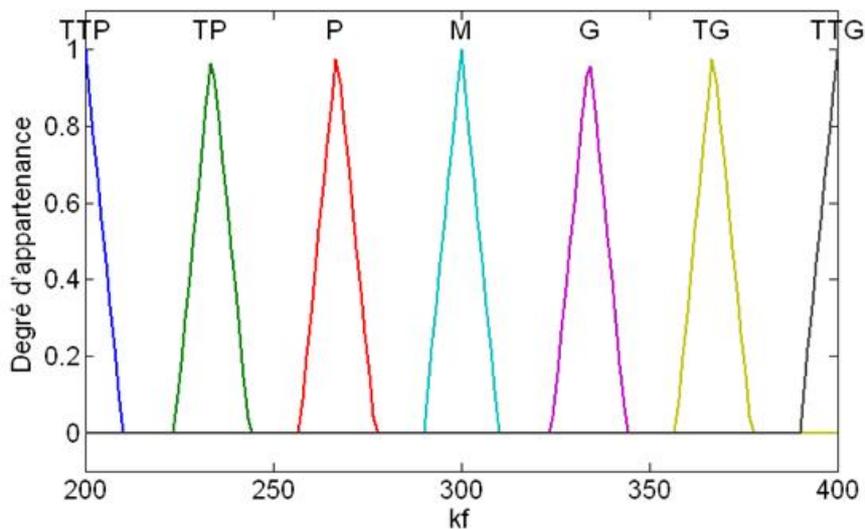
NG : négatif grand, NM : négatif moyen, NP : négatif petit, Z : zéros, PP : positif petit
 PM : positif moyen, PG : positif grand.

2) - pour la sortie :

TTP : très très petit, TP : très petit, P : petit, M : moyen, G: grand, TG : très grand
 TTG : très très grand.



(a)



(b)

Figure III.8: (a), (b) Les fonctions d'appartenances des variables linguistiques e , \dot{e} et u_f

III.4.1.2 La base de règles :

Les règles de contrôle du système Degré d'appartenance superviseur flou sont conçues à base du vecteur de l'erreur e et sa dérivé \dot{e} , et $k_f = \{k_{f1}, k_{f2}\}$, avec l'expression des règles comme suite :

$$R^{(i)} :: SI e(t) \text{ est } E_1^i \text{ et } \dot{e}(t) \text{ est } E_2^i \text{ ALORS } k_f \text{ est } G^i \quad (III. 23)$$

Tel que E_1^i , E_2^i , G^i representent les ensembles d'appartenance de $e(t)$, $\dot{e}(t)$ et k_f respectivement. Les règles linguistiques sont présentées dans le tableau III.2 [1].

k_f		s_i						
		NG	NM	NP	Z	PP	PM	PG
\dot{s}_i	NG	M	P	TP	TTP	TP	P	M
	NM	G	M	P	VP	P	M	G
	NP	TG	G	M	P	M	G	TG
	Z	TTG	VG	G	M	G	TG	TTG
	PP	TG	G	M	p	M	G	TG
	PM	G	M	P	VP	P	M	G
	PG	M	P	VP	TTP	TP	P	M

Tableau III.2 [1]

III.4.1.3 Inférence et défuzzification :

Un mécanisme d'inférence de Mamdani est utilisé avec l'emploi de la fonction minimum pour l'intersection et la méthode du centre de gravité pour défuzzifier la sortie.

Le signal de sortie globale du contrôleur est comme suit :

$$u = u_{eq} + u_c = u_{eq} + k_f u_f = (k_d g_a)^{-1} [k_p \dot{e} + k_i e - k_d f_a + k_d \ddot{q}_d] + k_f u_f \quad (III. 24)$$

Tel que :

$$u_f = FSMC(s, \dot{s}) \text{ et } k_f = f(e, \dot{e})$$

Remarque :

On ce qui concerne la stabilité du système, elle est assurée comme il a été prouvé dans la partie III.2.2 tant que la condition $k_f > g_a^{-1}\alpha$ est vérifiée, ce qui nous donne des valeurs de bornes d'incertitudes et de perturbation plus flexibles.

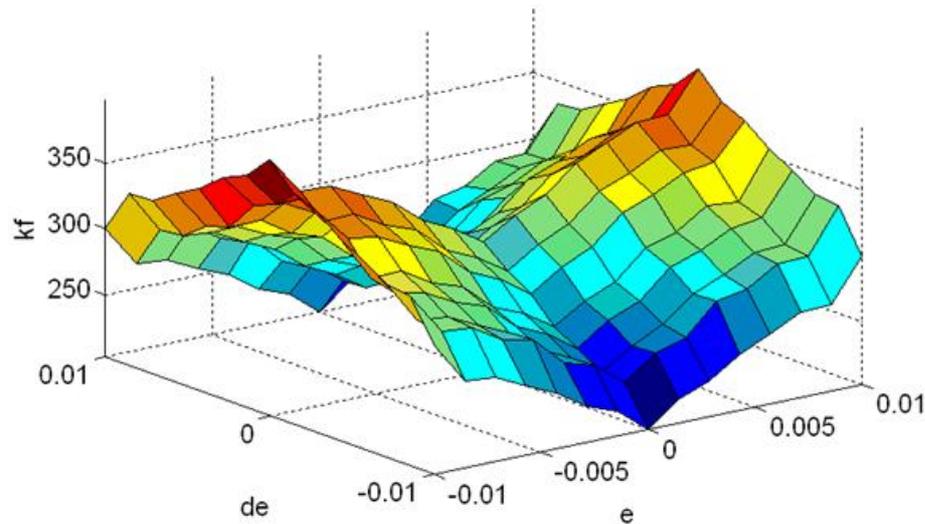


Figure III.9: La surface commande de k_f

III.5 Simulation et résultats :

III.5.1 Commande par mode glissant flou (FSMC):

1) Régulation :

La trajectoire désirée pour la régulation des deux articulations est:

$$q_{d1,2} = u(t) = \begin{cases} 1 \text{ rad} & \text{pour } t \geq 0 \\ 0 & \text{ailleurs} \end{cases}.$$

Avec les valeurs des paramètres utilisées pour la régulation :

$$k_p = \begin{bmatrix} 65 & 0 \\ 0 & 75 \end{bmatrix}, k_d = \begin{bmatrix} 9 & 0 \\ 0 & 7 \end{bmatrix}, k_i = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}, k_f = \begin{bmatrix} 400 & 0 \\ 0 & 400 \end{bmatrix}$$

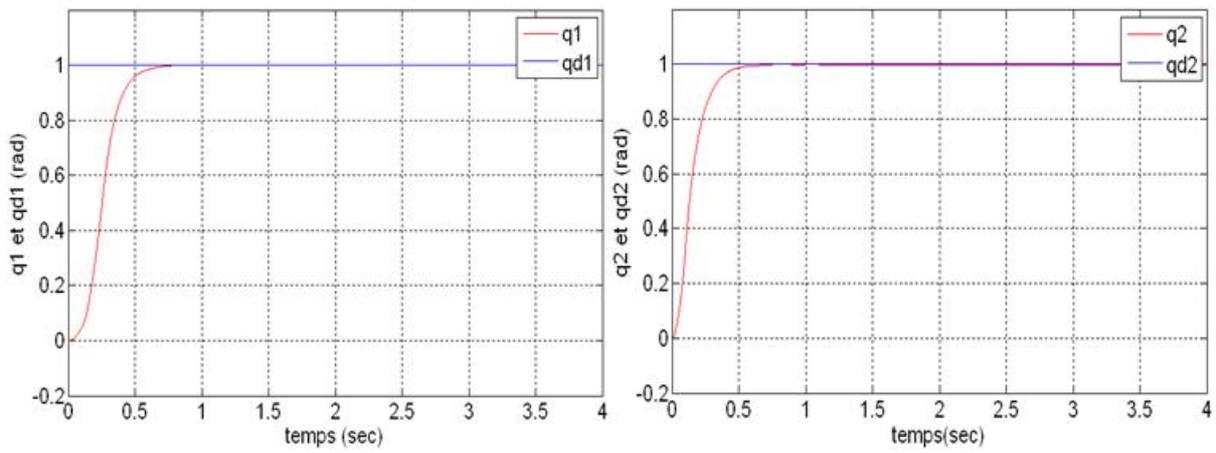
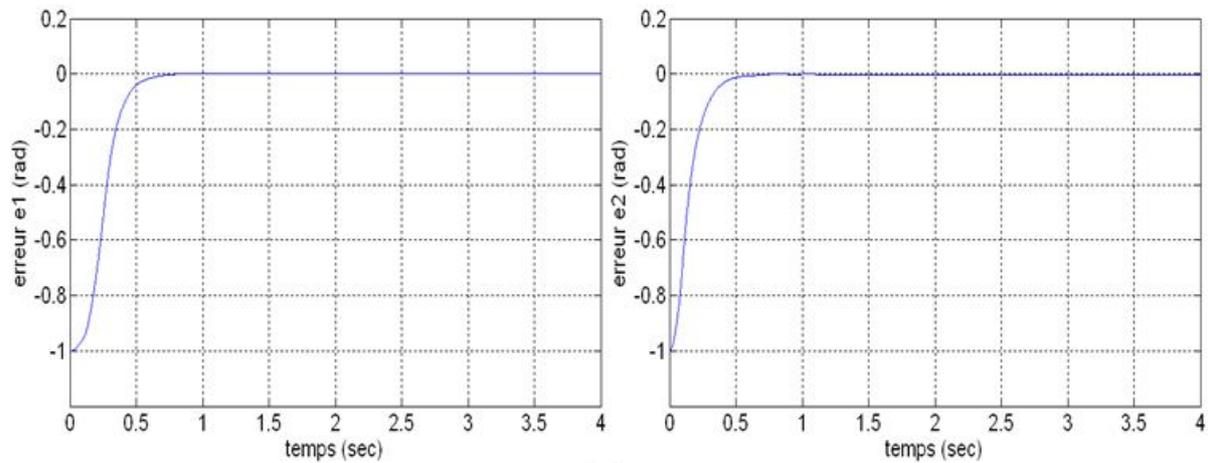
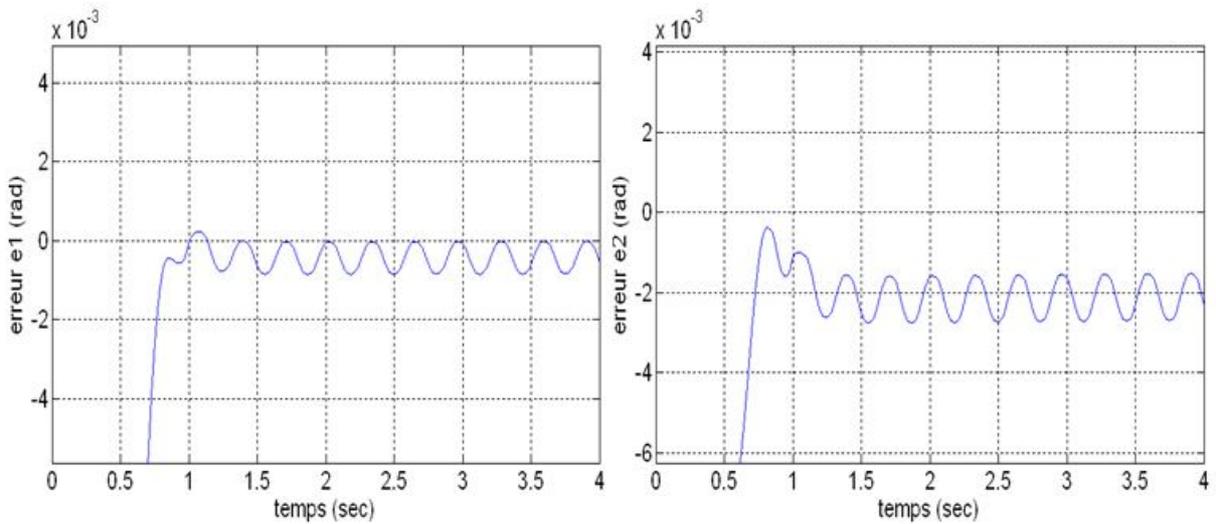


Figure III.10: Résultat de suivi de la trajectoire désirée pour les deux articulations -cas de régulation-



(a)



(b)

Figure III.11: (a), (b) Résultat de l'erreur de suivi de la trajectoire désirée et son zoom pour les deux articulations -cas de régulation-

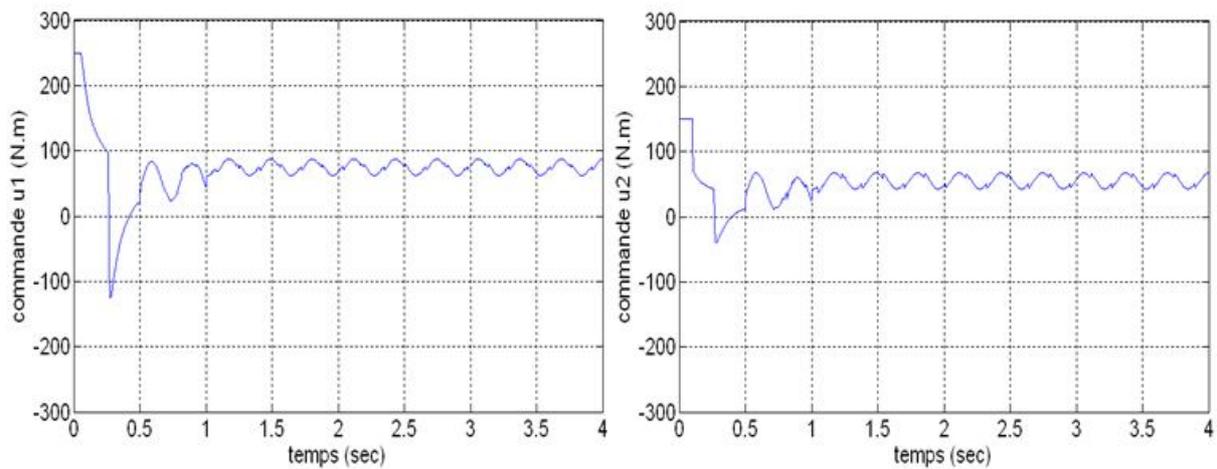


Figure III.12 : Résultat des couples générés pour le suivi de la trajectoire désirée pour les deux articulations -cas de régulation-

2) Poursuite :

La trajectoire désirée pour la régulation des deux articulations est:

$$q_{d1,2} = \sin\left(\pi t + \frac{3\pi}{2}\right) \text{ rad.}$$

Les valeurs des paramètres utilisées pour la régulation :

$$k_p = \begin{bmatrix} 160 & 0 \\ 0 & 140 \end{bmatrix}, k_d = \begin{bmatrix} 31 & 0 \\ 0 & 29 \end{bmatrix}, k_i = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}, k_f = \begin{bmatrix} 400 & 0 \\ 0 & 400 \end{bmatrix}$$

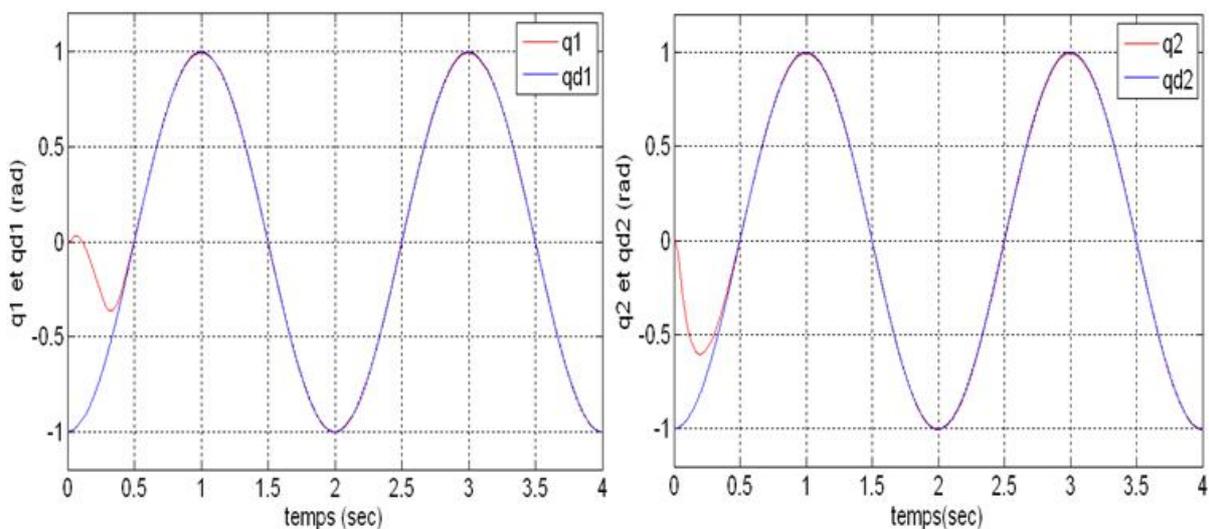


Figure III.13 : Résultat de suivi de la trajectoire désirée pour les deux articulations -cas de poursuite-

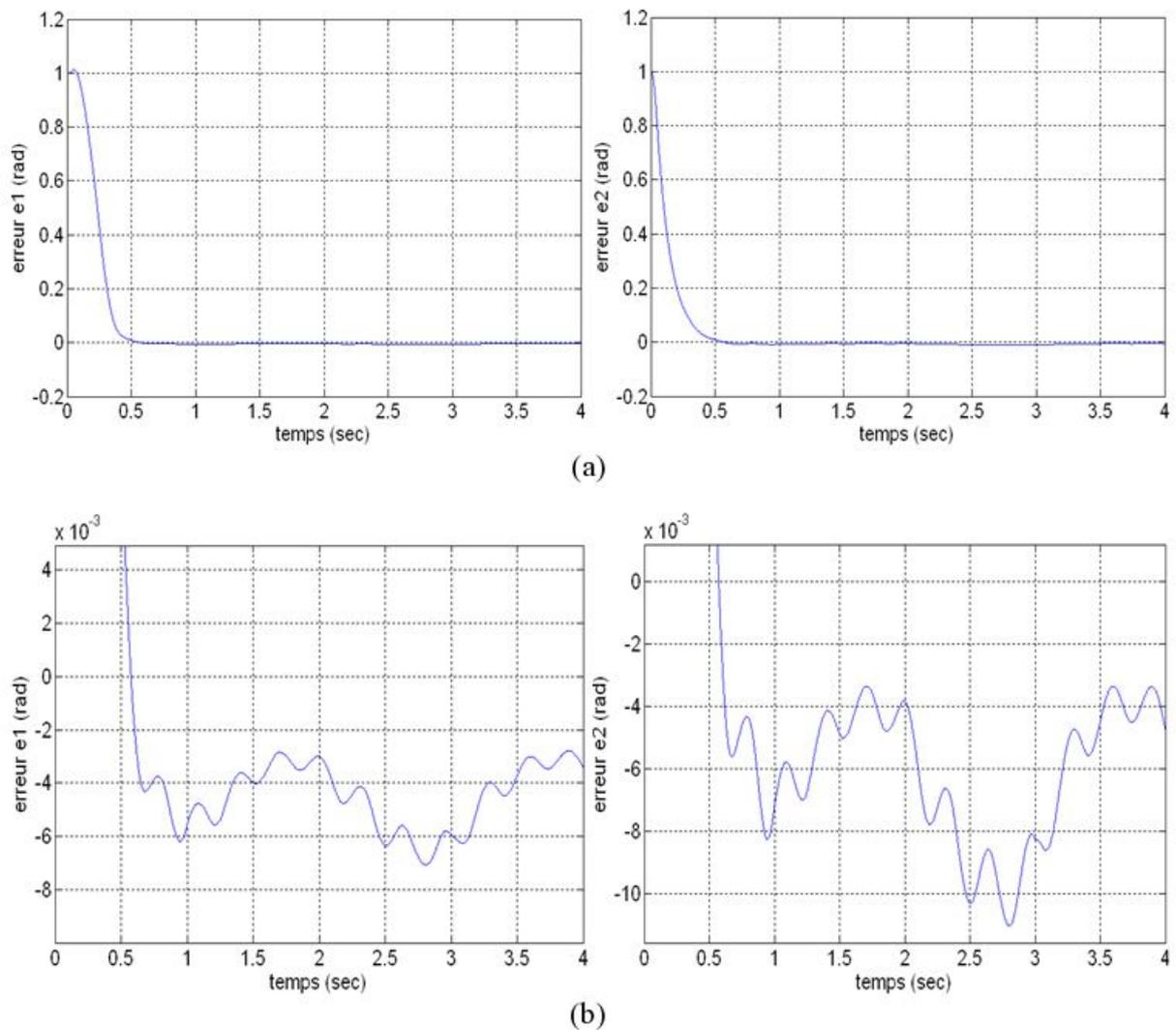


Figure III.14: (a), (b) Résultat de l'erreur de suivi de la trajectoire désirée et son zoom pour les deux articulations -cas de poursuite-

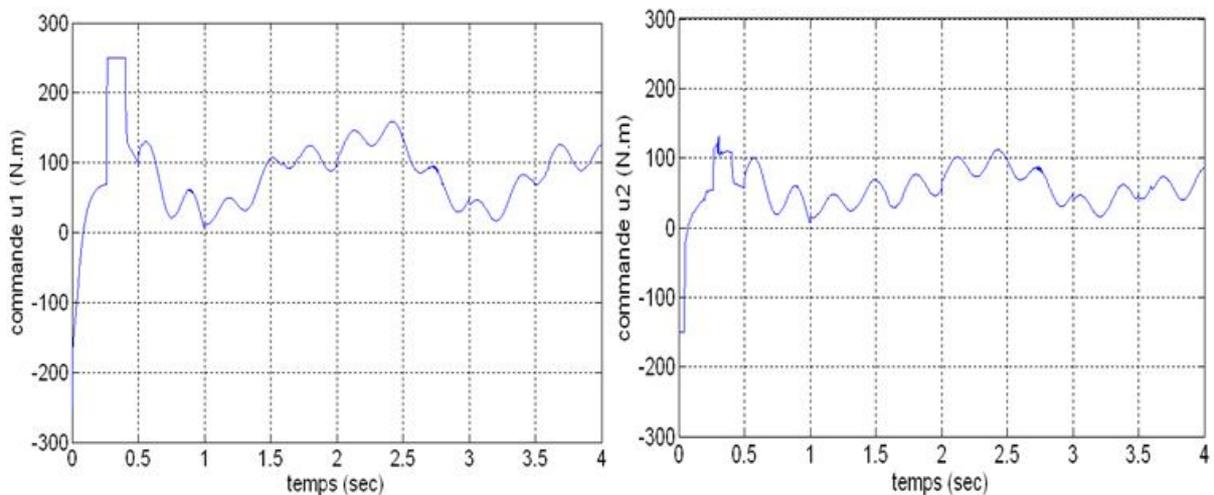


Figure III.15 : Résultat des couples générés pour le suivi de la trajectoire désirée pour les deux articulations -cas de poursuite-

III.5.2 Commande par mode glissant flou adaptatif (AFSMC) :

III.5.2.1 Régulation :

La trajectoire désirée pour la régulation des deux articulations est:

$$q_{d1,2} = u(t) = \begin{cases} 1 \text{ rad} & \text{pour } t \geq 0 \\ 0 & \text{ailleurs} \end{cases}.$$

Les valeurs des paramètres utilisées pour la régulation :

$$k_p = \begin{bmatrix} 65 & 0 \\ 0 & 75 \end{bmatrix}, k_d = \begin{bmatrix} 9 & 0 \\ 0 & 7 \end{bmatrix}, k_i = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}, k_f = \{300, 500\}$$

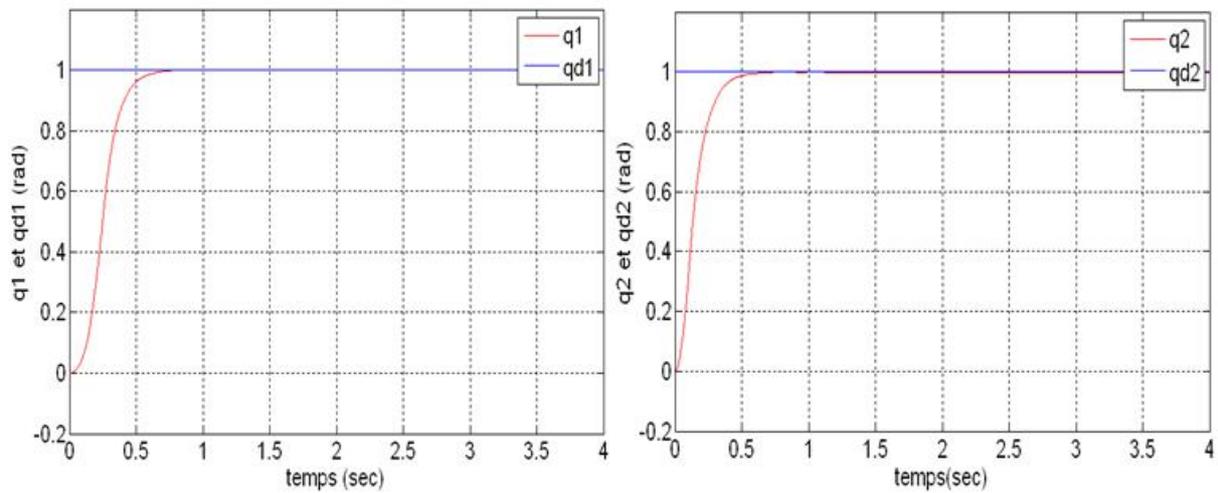


Figure III.16 : Résultat de suivi de la trajectoire désirée pour les deux articulations
-cas de régulation-

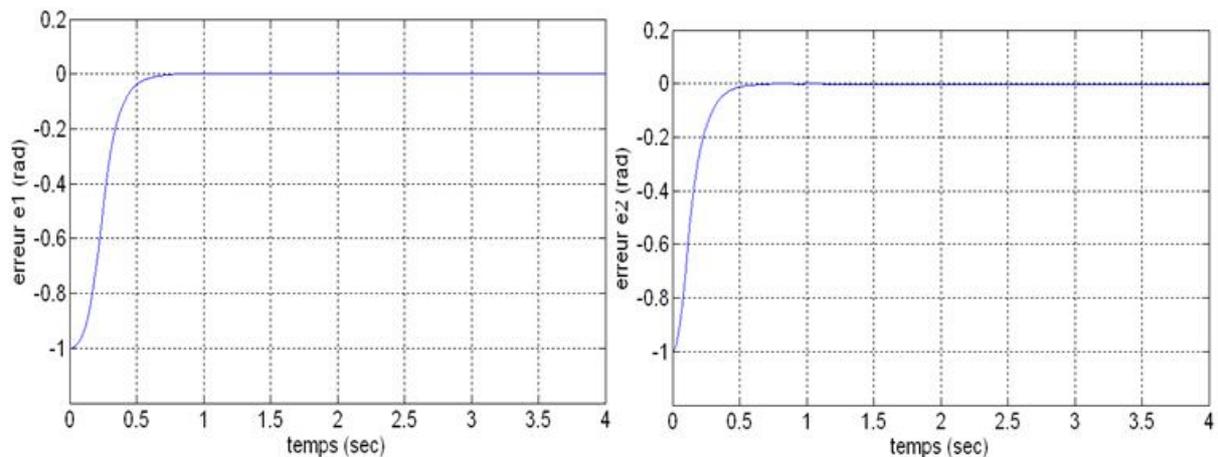


Figure III.17 : Résultat de l'erreur de suivi de la trajectoire désirée pour les deux
articulations -cas de régulation-

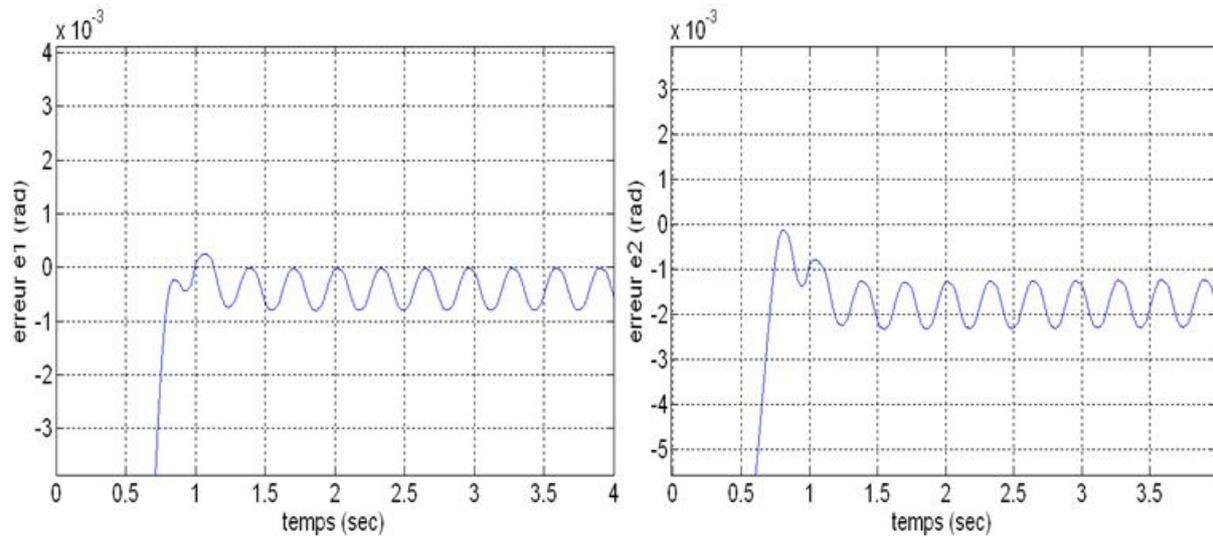


Figure III.18 : Zoom sur l'erreur de suivi de la trajectoire désirée pour les deux articulations -cas de régulation-

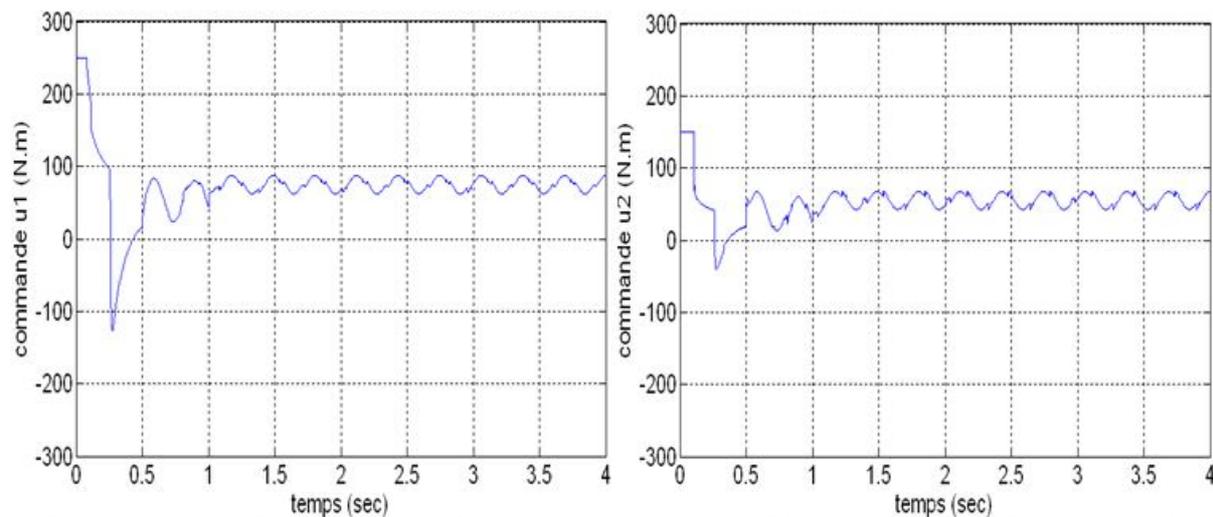


Figure III.19 : Résultat des couples générés pour le suivi de la trajectoire désirée pour les deux articulations -cas de régulation-

III.5.2.2 Poursuite :

La trajectoire désirée pour la régulation des deux articulations est:

$$q_{d1,2} = \sin\left(\pi t + \frac{3\pi}{2}\right) \text{ rad}.$$

Les valeurs des paramètres utilisées pour la régulation :

$$k_p = \begin{bmatrix} 160 & 0 \\ 0 & 140 \end{bmatrix}, k_d = \begin{bmatrix} 31 & 0 \\ 0 & 29 \end{bmatrix}, k_i = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}, k_f = \{300, 500\}$$

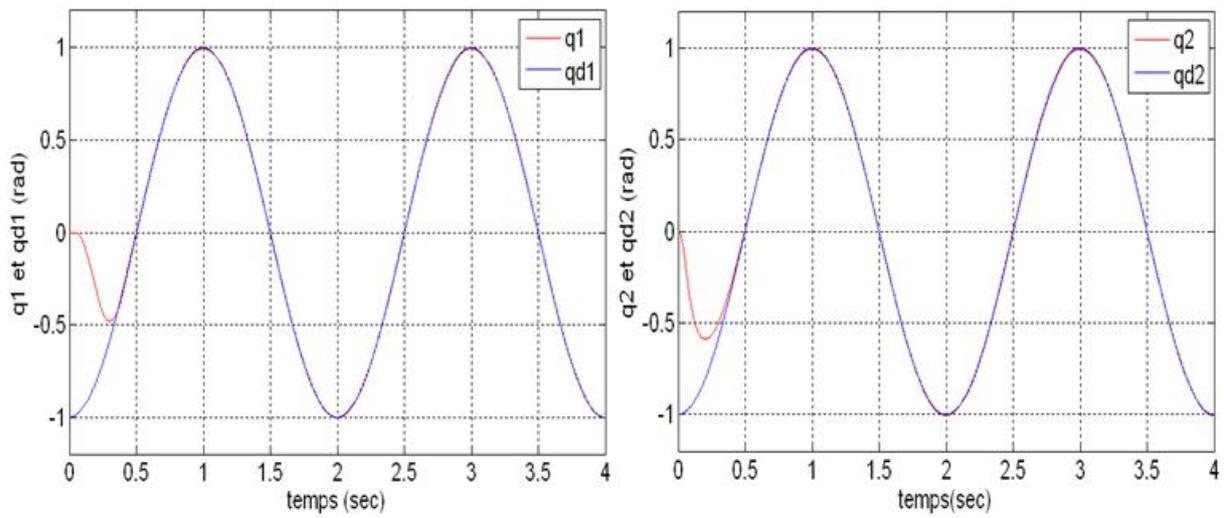


Figure III.20: Résultat de suivi de la trajectoire désirée pour les deux articulations -cas de poursuite-

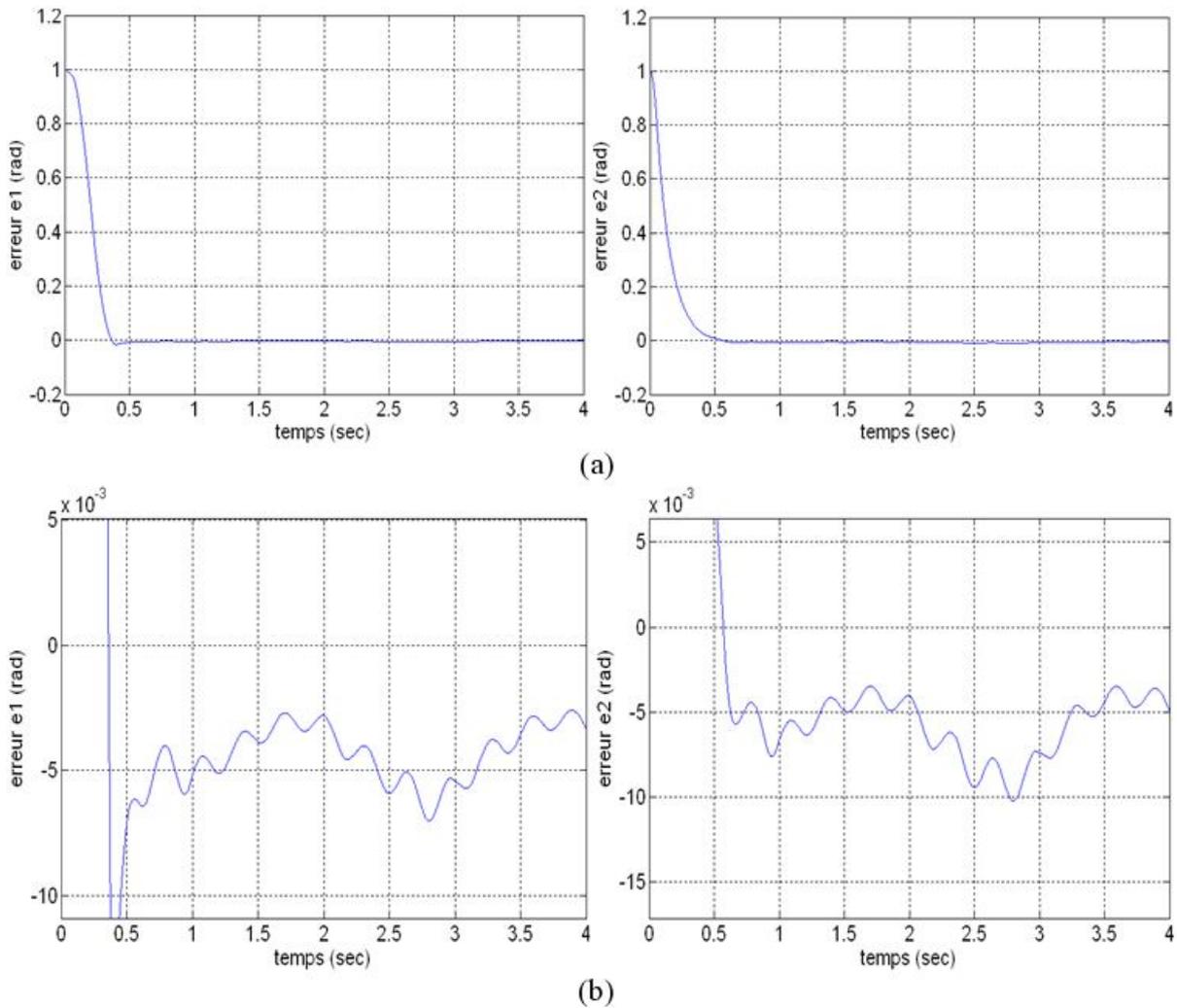


Figure III.21: (a), (b) Résultat de l'erreur de suivi de la trajectoire désirée et son zoom pour les deux articulations -cas de poursuite-

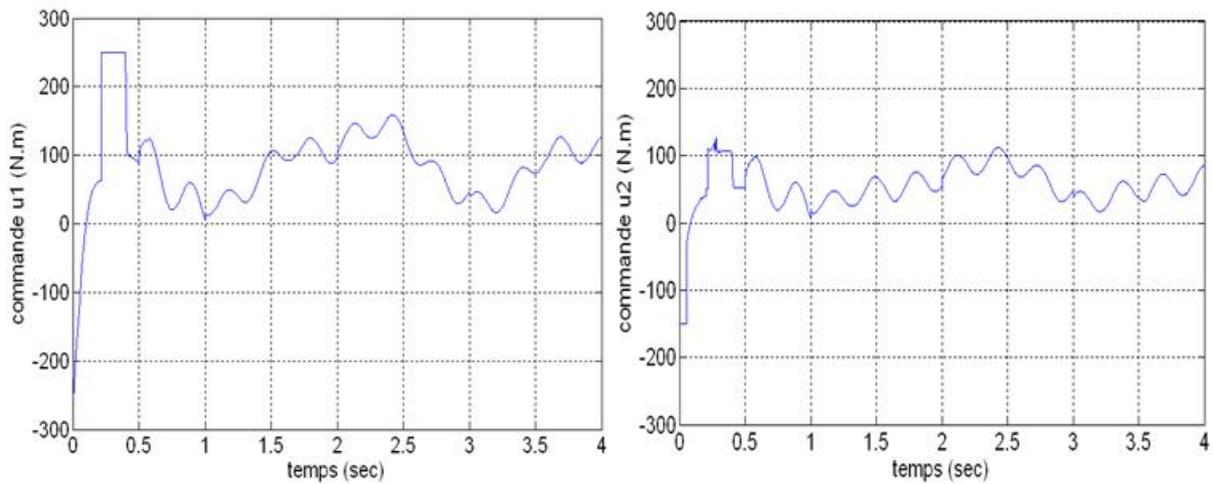


Figure III.22 : Résultat des couples générés pour le suivi de la trajectoire désirée pour les deux articulations -cas de poursuite-

Les figures ci-dessus, montrent un bon comportement de l'algorithme du mode glissant flou (FSMC) et du mode glissant flou adaptatif (AFSMC) du point de vue erreur de suivi et minimisation de l'effet du chattering.

III.6.Conclusion :

Dans ce chapitre, et dans le but d'améliorer les performances de la commande par mode glissant classique citée dans le chapitre II, nous avons remplacé la fonction discrète de l'algorithme classique par une alternative floue qui modélise son comportement, en concevant un contrôleur à mode glissant flou (FSMC), ensuite on lui a ajouté un superviseur flou pour ajuster automatiquement le gain de sortie du contrôleur en obtenant un contrôleur adaptatif à mode glissant flou (AFSMC).

Les résultats de simulation des méthodes étudiées dans cette partie, montrent les performances apportées, où on observe que le phénomène de réticence qui caractérise la commande classique est disparu en conservant une erreur de suivi petite. Cependant on note que le superviseur flou ajouté ne donne pas une grande amélioration au (FSMC), d'où on aura plus de calculs pour des performances proches se qui est inutile.

Chapitre IV

Commande hybride Backstepping et Mode glissant

IV.1 Introduction :

Dans la littérature, on trouve plusieurs algorithmes de commande avec des avantages et des inconvénients pour chacun. Afin d'obtenir des commandes améliorées, il a été pensé utile de faire combiner ces différentes méthodes pour avoir des algorithmes hybrides plus performants. Ainsi la combinaison de la commande par mode glissant et la procédure récursive backstepping est une approche intéressante pour les contrôleurs robustes développés pour les systèmes non linéaires incertains.

Dans ce chapitre on va étudier cette approche hybride, ensuite on va essayer de l'améliorer en introduisant la logique floue pour obtenir une commande backstepping-mode glissant flou qui combine le backstepping, le mode glissant et la logique floue.

IV.2 Commande hybride backstepping-mode glissant :

Dans cette partie, nous présenterons la commande backstepping combinée avec la commande par mode glissant. L'approche est dédiée au problème de poursuite et de régulation d'un bras manipulateur à 2 ddl. La commande traitée ici [37] s'intéresse aux systèmes multi-entrée- multi sortie « MIMO » comme les manipulateurs, elle est issue des travaux développés à la suite du travail de Zinober et de Liu [37], pour les systèmes d'une entrée et une sortie « SISO ».

IV.2.1 Conception du contrôleur :

La nature récursive du contrôleur proposé ressemble à la méthodologie backstepping normale. Cependant, cette architecture emploie le backstepping pour concevoir des contrôleurs virtuels qui contiennent une partie de commande à mode glissant avec une surface de glissement d'ordre zéro à chaque opération récursive [37]. L'avantage de cette approche est que chaque contrôleur virtuel peut compenser les incertitudes et bruits externes dans chaque étape. Une analyse partielle de stabilité sera faite à chaque opération récursive jusqu'à l'arriver à la dernière étape où on va établir la loi de commande globale qui stabilise le système.

On prend le modèle du manipulateur présenté dans (II.56) :

$$\begin{cases} \dot{x}_1 = x_2 \\ \dot{x}_2 = M(q)^{-1}\tau - M(q)^{-1}n(q, \dot{q}) - M(q)^{-1}b_1 - M(q)^{-1}b_2 \end{cases} \quad (IV.1)$$

$$\Rightarrow \begin{cases} \dot{x}_1 = x_2 \\ \dot{x}_2 = g_a u + f_a + b(t) \end{cases} \quad (IV.2)$$

On définit la transformation suivante :

$$s_1 = x_1 - q_d \rightarrow \dot{s}_1 = x_2 - \dot{q}_d \quad (IV.3)$$

On choisit une fonction stabilisante pour s_1 , on prend $x_2 = -k_1 s_1$ on obtient :

$$\dot{s}_1 = -k_1 s_1 + (x_2 - \dot{q}_d - k_1 s_1) \quad (IV.4)$$

$$\dot{s}_1 = -k_1 s_1 + s_2 \quad / \quad s_2 = x_2 - \dot{q}_d - \varphi(s_1) \quad (IV.5)$$

Lorsque l'entrée virtuelle $s_2 = 0$ alors le système est stable, (voir section II.3.2)

Maintenant on stabilise la deuxième partie.

On a :

$$\dot{s}_2 = \dot{x}_2 - \ddot{q}_d + k_1 \dot{s}_1 \quad (IV.6)$$

$$= g_a u + f_a + b(t) - \ddot{q}_d + k_1 \dot{s}_1 \quad (IV.7)$$

On prend comme fonction candidate de Lyapunov la fonction $V = \frac{1}{2}s_1^2 + \frac{1}{2}s_2^2$

$$\Rightarrow \dot{V} = s_1 \dot{s}_1 + s_2 \dot{s}_2 \quad (IV.8)$$

$$\dot{V} = s_1(-k_1 s_1 + s_2) + s_2(g_a u + f_a + b(t) - \ddot{q}_d - k_1(x_2 - \dot{q}_d)) \quad (IV.9)$$

$$\text{On prend } u = g_a^{-1}(-s_1 - k_2 s_2 - f_a + \ddot{q}_d + k_1(x_2 - \dot{q}_d)) + u_c \quad (IV.10)$$

On aura

$$\dot{V} = -k_1 s_1^2 - k_2 s_2^2 + s_2(b(t) + g_a u_c) \quad (IV.11)$$

On choisit :

$$u_c = -k_c \text{sign}(s_2) \quad (IV.12)$$

$$\Rightarrow \dot{V} = -k_1 s_1^2 - k_2 s_2^2 + s_2(b(t) - k_c g_a \text{sign}(s_2)) \quad (IV.13)$$

$$\leq -k_1 s_1^2 - k_2 s_2^2 + |s_2|(|b(t)| - k_c g_a) = -k_1 s_1^2 - k_2 s_2^2 + |s_2|(\alpha - k_c g_a) \quad (IV.14)$$

Pour que le système soit asymptotiquement stable il faut prendre $k_c > \alpha g_a^{-1}$

Ainsi la commande totale est :

$$u = g_a^{-1}(-s_1 - k_2 s_2 - f_a + \ddot{q}_d + k_1(x_2 - \dot{q}_d)) + u_c \quad (IV.15)$$

$$u = g_a^{-1}(-s_1 - k_2 s_2 - f_a + \ddot{q}_d + k_1(x_2 - \dot{q}_d)) - k_c \text{sign}(s_2) \quad (IV.16)$$

$$u = M(q)(-s_1 - k_2 s_2 + M(q)^{-1}n(q, \dot{q}) + \ddot{q}_d + k_1(x_2 - \dot{q}_d)) - k_c \text{sign}(s_2) \quad (IV.17)$$

Où, k_1 , k_2 et k_c sont des gains positive et $s_2 = [s_{2,1}, s_{2,2}]^T$ représente le vecteur de surface de l'articulation 1 et 2 respectivement.

IV.3 Présentation d'une commande hybride backstepping-mode glissant flou :

Dans le but d'améliorer la méthode hybride backstepping-mode glissant, on va introduire la logique floue sur la partie du mode glissant en concevant une commande à mode glissant flou comme on l'a expliqué dans le chapitre III.

IV.3.1 Conception du contrôleur :

L'architecture proposée est similaire à celle de la commande hybride précédente, la différence est de remplacer la partie discontinue du contrôleur par une approximation floue, d'où on obtient le mode glissant flou.

En suivant la procédure récursive du backstepping, on va chercher une fonction de Lyapunov qui va assurer la stabilité du système globale en utilisant le mode glissant flou pour compenser, si nécessaire est, les perturbations à chaque étape récursive.

On a :

$$\begin{cases} \dot{x}_1 = x_2 \\ \dot{x}_2 = M(q)^{-1}\tau - M(q)^{-1}n(q, \dot{q}) - M(q)^{-1}b_1 - M(q)^{-1}b_2 \end{cases} \quad (IV.18)$$

$$\Rightarrow \begin{cases} \dot{x}_1 = x_2 \\ \dot{x}_2 = g_a u + f_a + b(t) \end{cases} \quad (IV.19)$$

On définit la transformation :

$$s_1 = x_1 - q_d \rightarrow \dot{s}_1 = x_2 - \dot{q}_d \quad (IV.20)$$

En suivant les mêmes étapes comme de (IV.4) à (IV.10) on obtient :

$$\dot{V} = -k_1 s_1^2 - k_2 s_2^2 + s_2(b(t) + g_a u_c) \quad (IV.21)$$

Et on prend :

$$u_c = k_f u_f \quad (IV.23)$$

On aura :

$$\dot{V} = -k_1 s_1^2 - k_2 s_2^2 + s_2(b(t) + k_c g_a u_f) \quad (IV.24)$$

Où k_f est le facteur de normalisation de la commande u_c , u_f est la sortie de la partie floue du contrôleur.

Maintenant on cherche la commande u_c qui va stabiliser le système globale en tenant compte des incertitudes et perturbations du système.

❖ Conception de u_f :

On utilise comme partie de commande floue u_f la même que celle utilisée dans le contrôleur du mode glissant flou du chapitre III. Le contrôleur flou a donc deux entrées et une sortie s , \dot{s} et le signal de commande u_f respectivement, avec les mêmes variables linguistiques pour les trois ensembles flous $\{(s, \mu(s)), (\dot{s}, \mu(\dot{s})), (u_f, \mu(u_f))\}$ comme suite :

NG: négatif grand, NM : négatif moyen, NP : négatif petit, Z : zéros, PP : positif petit

PM : positif moyen, PG : positif grand.

On note que l'entrée du contrôleur s et \dot{s} doit être normalisé [1, 22].

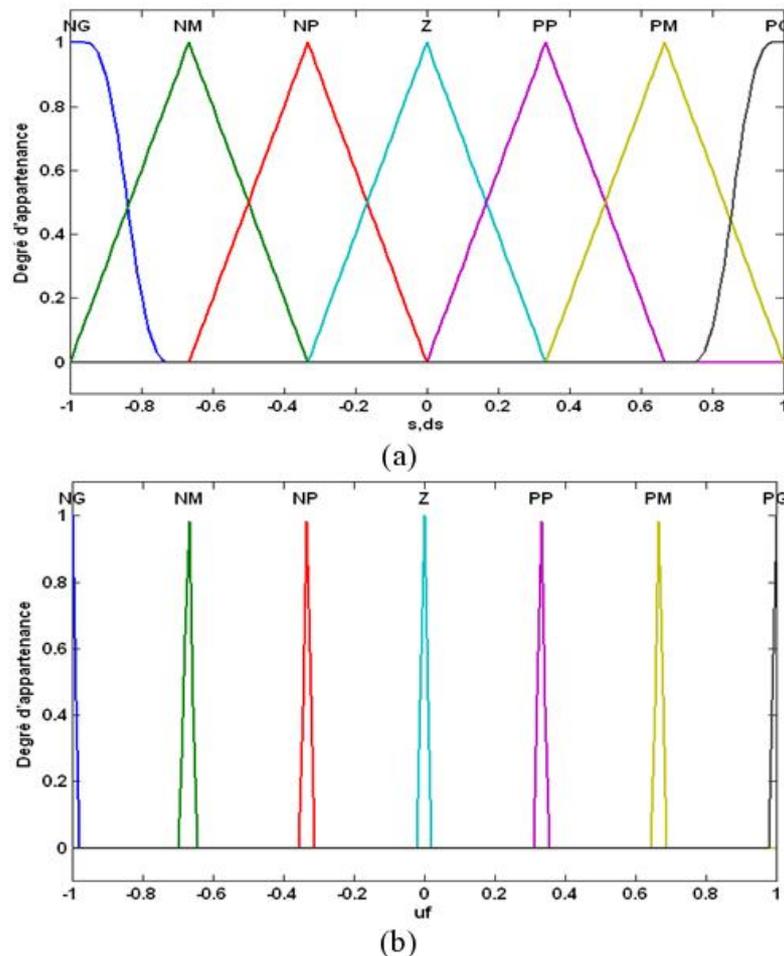


Figure IV.1 : (a), (b) Fonctions d'appartenance pour les variables linguistiques de s , \dot{s} et u_f

❖ Les règles floues sont celles du tableau III.1. Ces règles assurent le fait que lorsque la multiplication de s et de \dot{s} est positive, la fonction d'appartenance d' u_f est réglée de telle manière que son signe soit l'opposé de celui de s , ce qui donne: $s u_f \leq -|s|$ [1, 22].

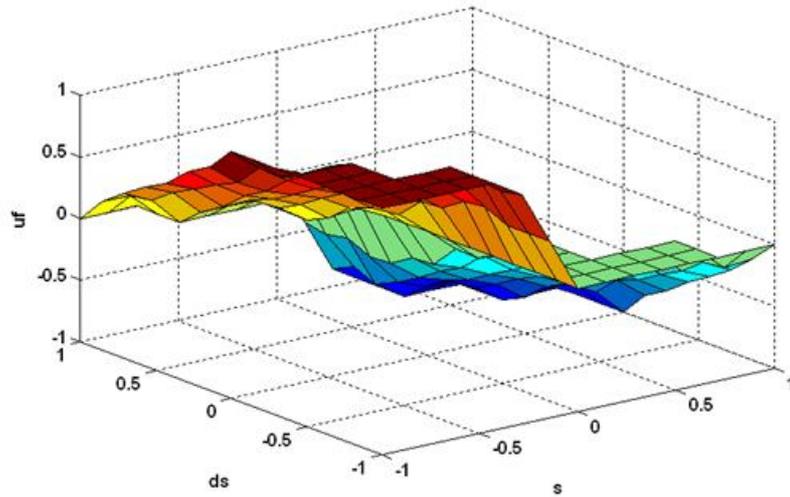


Figure IV.2 : La surface commande de u_f

IV.3.2 Analyse de stabilité :

On continue depuis l'équation (IV.24):

$$\dot{V} = -k_1 s_1^2 - k_2 s_2^2 + s_2(b(t) + k_c g_a u_f) \quad (IV.25)$$

$$= -k_1 s_1^2 - k_2 s_2^2 + s_2(b(t) + k_c g_a FSMC(s_2, \dot{s}_2)) \quad (IV.26)$$

$$\leq -k_1 s_1^2 - k_2 s_2^2 + |s_2|(|b(t)| - k_c g_a) \quad (IV.27)$$

$$\leq -k_1 s_1^2 - k_2 s_2^2 + |s_2|(\alpha - k_c g_a) \quad (IV.28)$$

Si on choisi $k_d > \alpha g_a^{-1}$ alors notre système est asymptotiquement stable.

Ainsi la commande totale $\tau = u$ est :

$$\tau = g_a^{-1}(-s_1 - k_2 s_2 - f_a + \ddot{q}_d + k_1(x_2 - \dot{q}_d)) + u_c \quad (IV.24)$$

$$\tau = g_a^{-1}(-s_1 - k_2 s_2 - f_a + \ddot{q}_d + k_1(x_2 - \dot{q}_d)) - k_c FSMC(s_2, \dot{s}_2) \quad (IV.25)$$

$$\tau = M(q)(-s_1 - k_2 s_2 + M(q)^{-1}n(q, \dot{q}) + \ddot{q}_d + k_1(x_2 - \dot{q}_d)) - k_c FSMC(s_2, \dot{s}_2) \quad (IV.26)$$

IV.4. Simulation et résultats :

IV.4.1 Commande hybride backstepping-mode glissant :

1) Régulation :

La trajectoire désirée pour la régulation des deux articulations est:

$$q_{d1,2} = u(t) = \begin{cases} 1 & \text{rad pour } t \geq 0 \\ 0 & \text{ailleurs} \end{cases}.$$

Les valeurs des paramètres utilisés pour la régulation :

$$k_1 = \begin{bmatrix} 30 & 0 \\ 0 & 105 \end{bmatrix}, k_2 = \begin{bmatrix} 21 & 0 \\ 0 & 75 \end{bmatrix}, k_c = \begin{bmatrix} 80 & 0 \\ 0 & 80 \end{bmatrix}$$

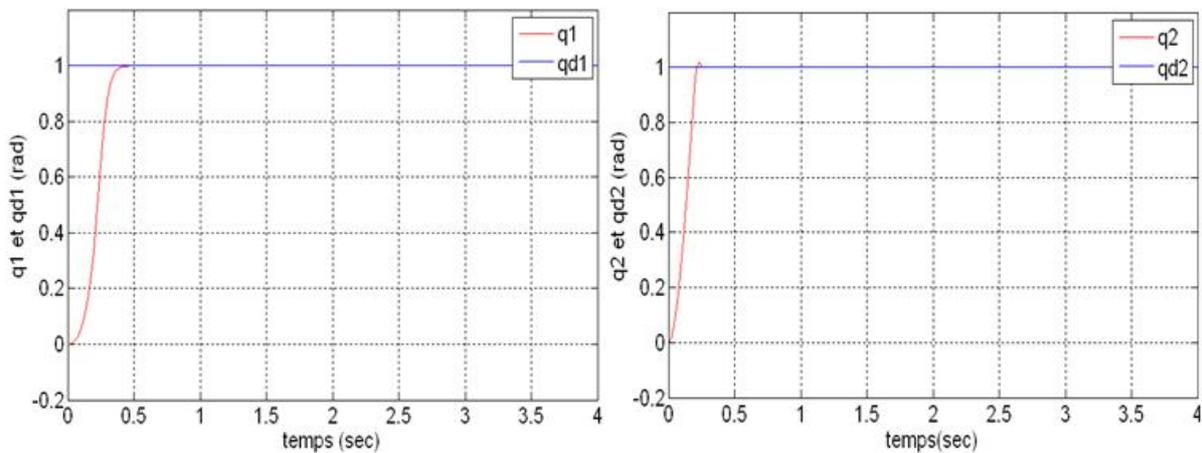


Figure IV.3 : Résultat de suivi de la trajectoire désirée pour les deux articulations
-cas de régulation-

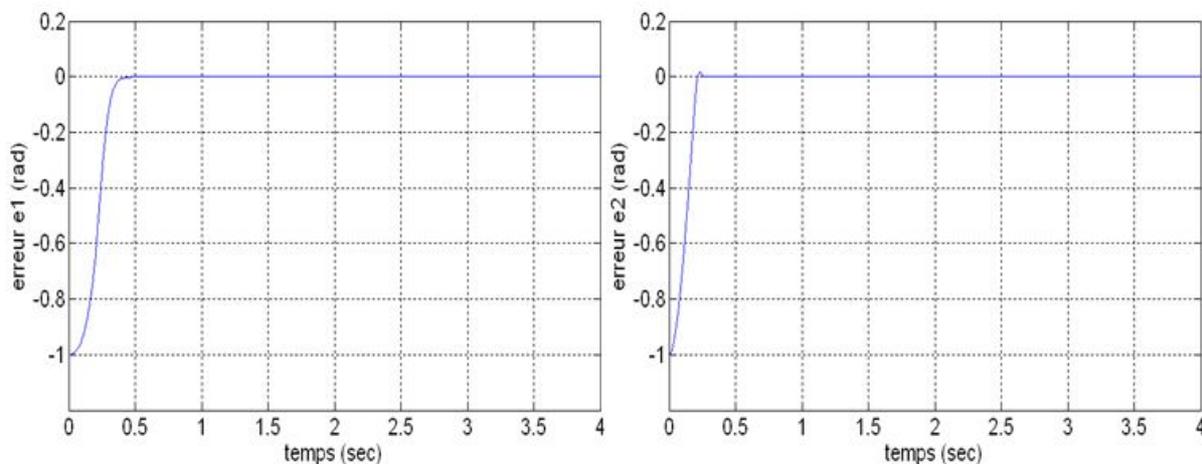


Figure IV.4 : Résultat de l'erreur de suivi de la trajectoire désirée pour les deux articulations
-cas de régulation-

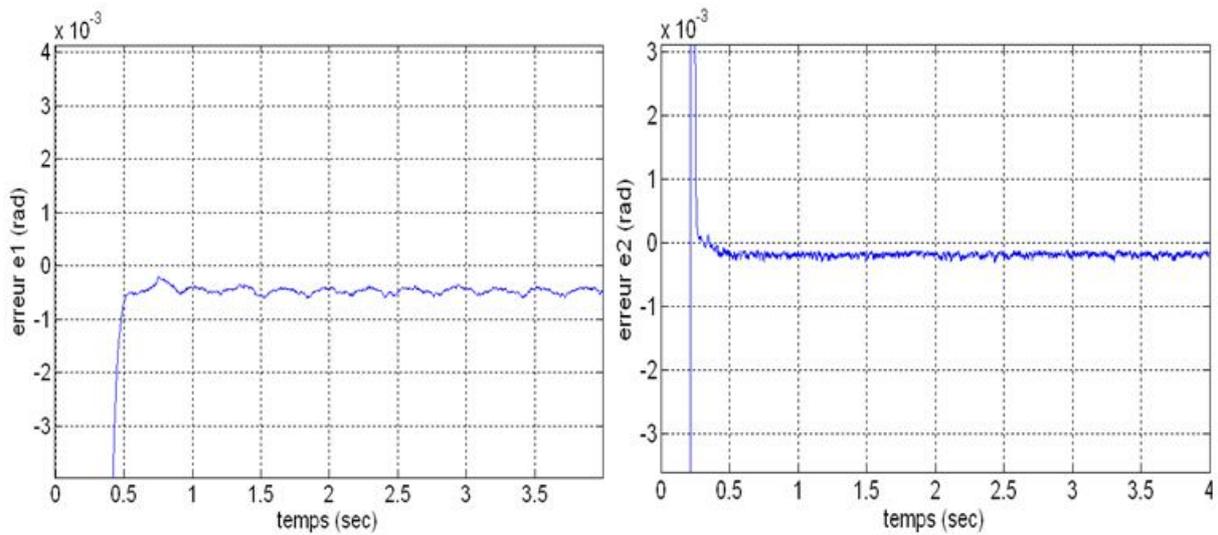


Figure IV.5 : Zoom sur l'erreur de suivi de la trajectoire désirée pour les deux articulations -cas de régulation-

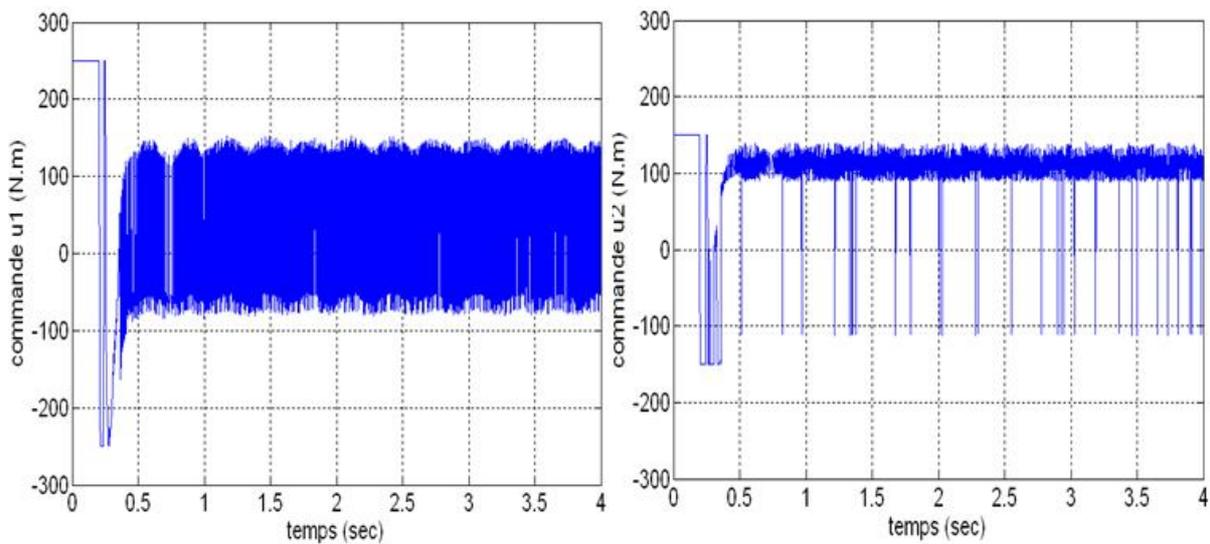


Figure IV.6 : Résultat des couples générés pour le suivi de la trajectoire désirée pour les deux articulations -cas de régulation-

2) Poursuite :

La trajectoire désirée pour la régulation des deux articulations est:

$$q_{d1,2} = \sin\left(\pi t + \frac{3\pi}{2}\right) \text{ rad.}$$

Les valeurs des paramètres utilisés pour la poursuite :

$$k_1 = \begin{bmatrix} 60 & 0 \\ 0 & 100 \end{bmatrix}, k_2 = \begin{bmatrix} 2 & 0 \\ 0 & 10 \end{bmatrix}, k_c = \begin{bmatrix} 80 & 0 \\ 0 & 80 \end{bmatrix}$$

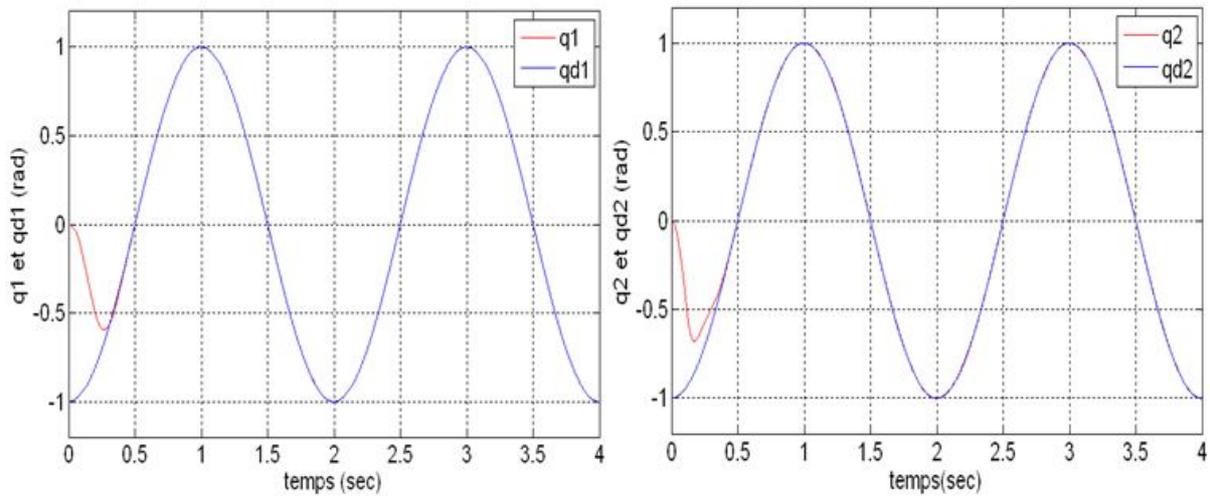
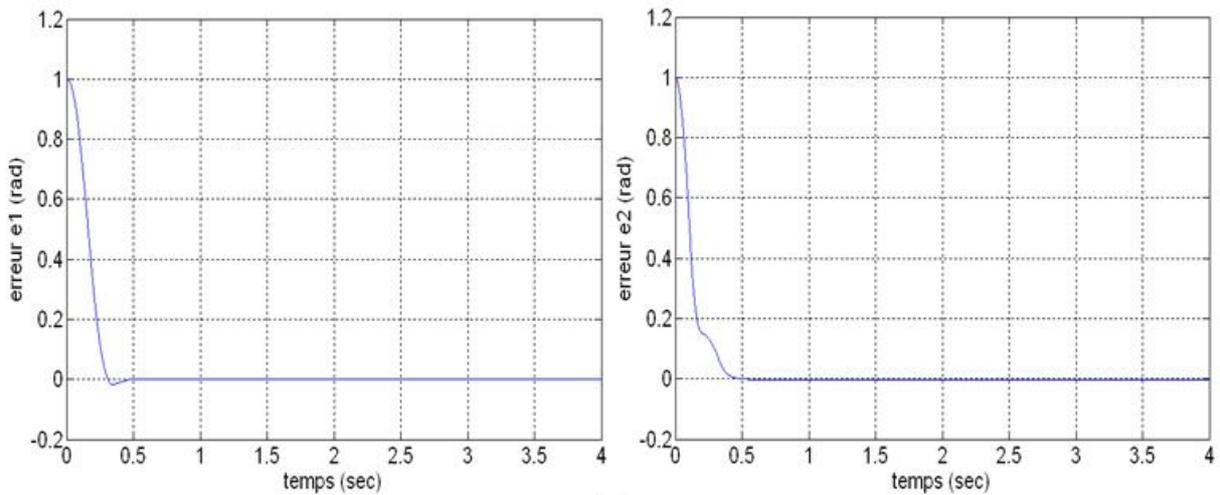
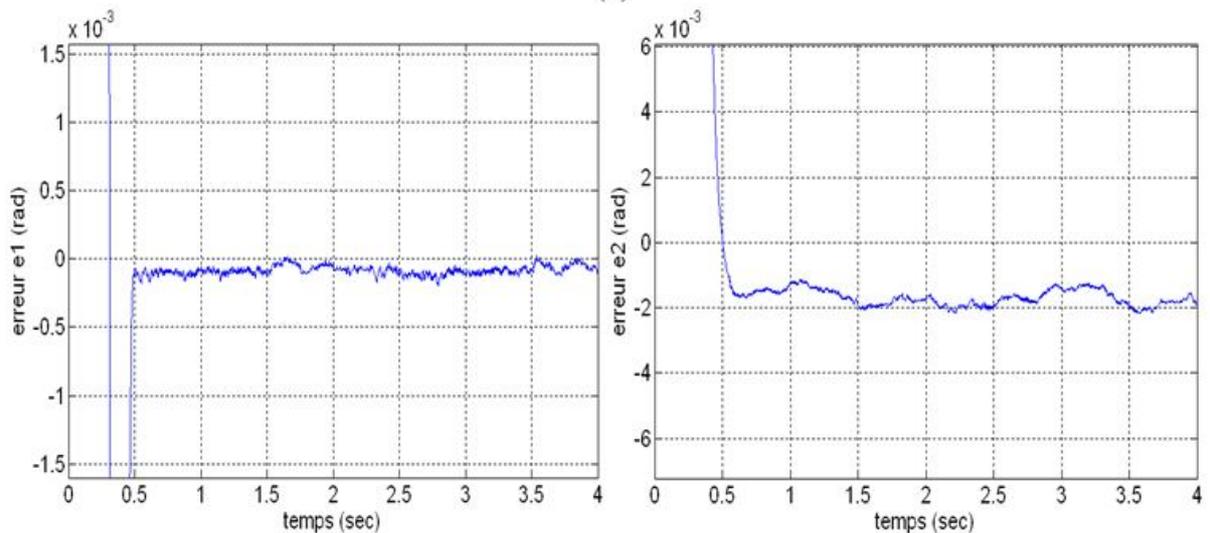


Figure IV.7 : Résultat de suivi de la trajectoire désirée pour les deux articulations -cas de poursuite-



(a)



(b)

Figure IV.8 : (a), (b) Résultat de l'erreur de suivi de la trajectoire désirée et son zoom pour les deux articulations - cas de poursuite-

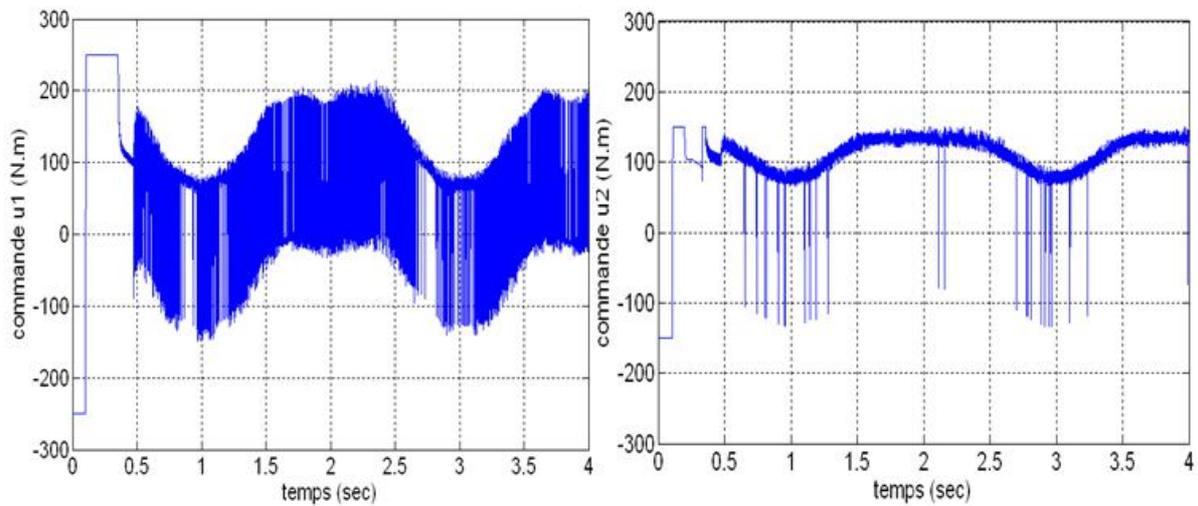


Figure IV.9 : Résultats des couples générés pour le suivi de la trajectoire désirée pour les deux articulations -cas de poursuite-

III.4.2 Contrôleur hybride backstepping-mode glissant fluide :

1) Régulation :

La trajectoire désirée pour la régulation des deux articulations est:

$$q_{d1,2} = u(t) = \begin{cases} 1 \text{ rad} & \text{pour } t \geq 0 \\ 0 & \text{ailleurs} \end{cases}.$$

Les valeurs des paramètres utilisés pour la régulation :

$$k_1 = \begin{bmatrix} 30 & 0 \\ 0 & 105 \end{bmatrix}, k_2 = \begin{bmatrix} 21 & 0 \\ 0 & 75 \end{bmatrix}, k_f = \begin{bmatrix} 200 & 0 \\ 0 & 200 \end{bmatrix}$$

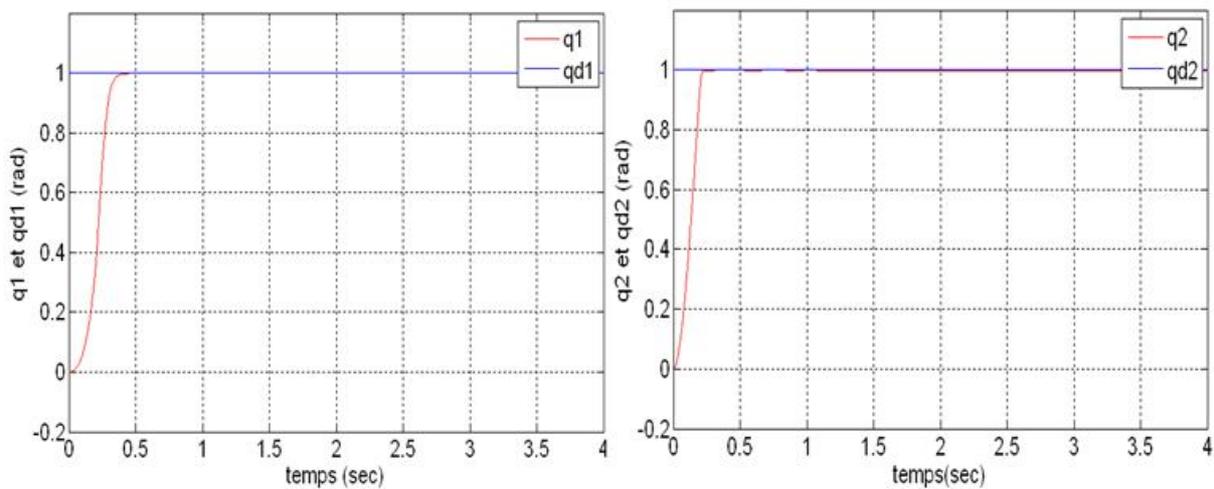


Figure IV.10 : Résultat de suivi de la trajectoire désirée pour les deux articulations -cas de régulation-

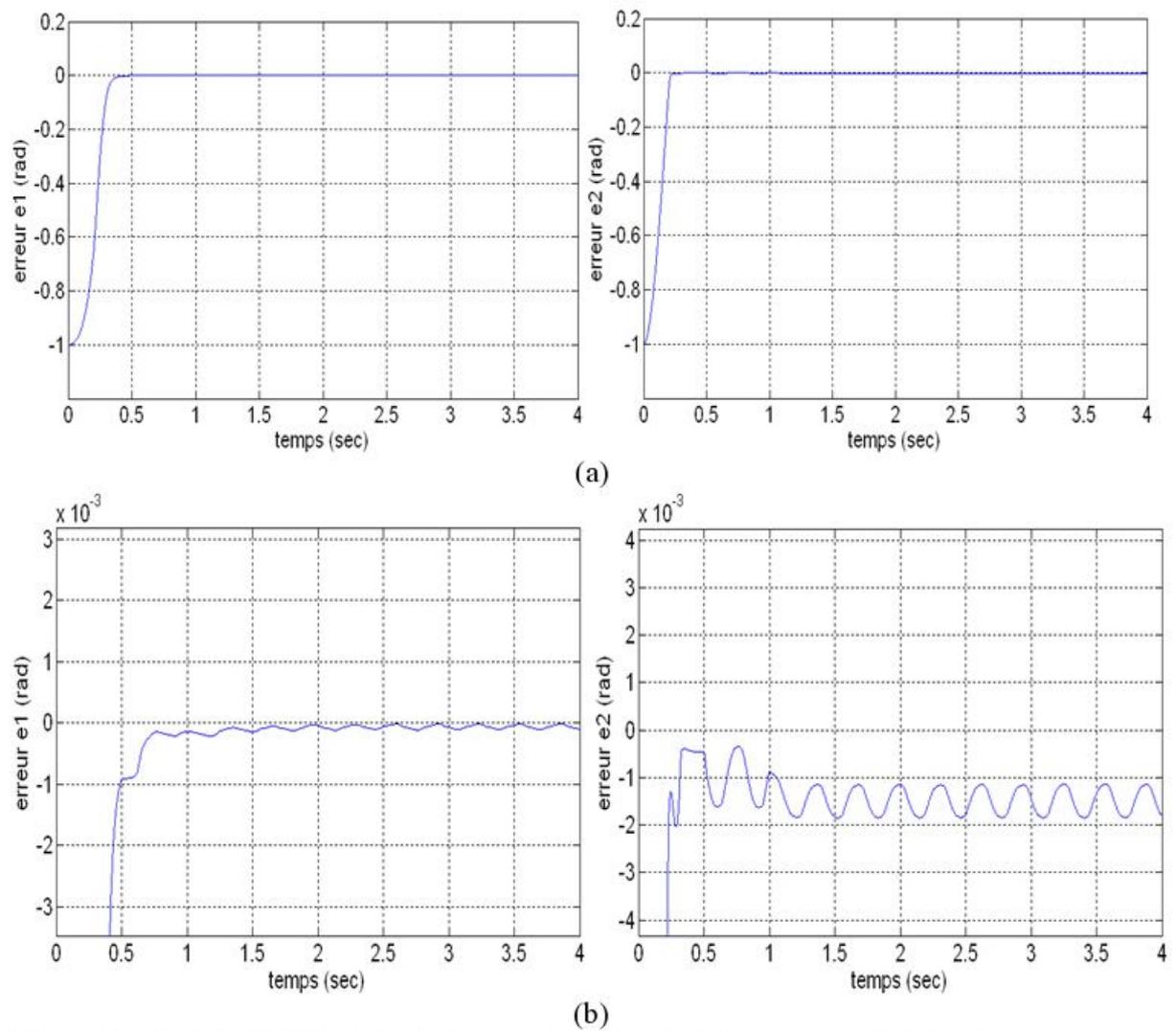


Figure IV.11 : (a), (b) Résultats de l'erreur de suivi de la trajectoire désirée et son zoom pour les deux articulations -cas de régulation-

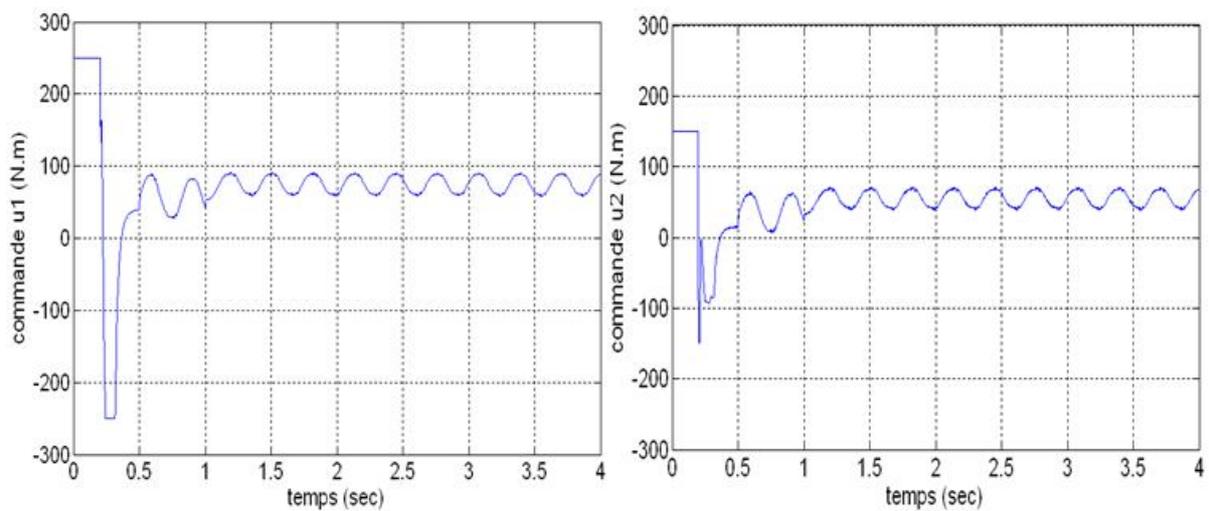


Figure IV.12 : Résultats des couples générer pour le suivi de la trajectoire désirée pour les deux articulations -cas de régulation-

2) Poursuite :

La trajectoire désirée pour la régulation des deux articulations est:

$$q_{d1,2} = \sin\left(\pi t + \frac{3\pi}{2}\right).$$

Les valeurs des paramètres utilisés pour la poursuite :

$$k_1 = \begin{bmatrix} 60 & 0 \\ 0 & 100 \end{bmatrix}, k_2 = \begin{bmatrix} 2 & 0 \\ 0 & 10 \end{bmatrix}, k_f = \begin{bmatrix} 400 & 0 \\ 0 & 400 \end{bmatrix}$$

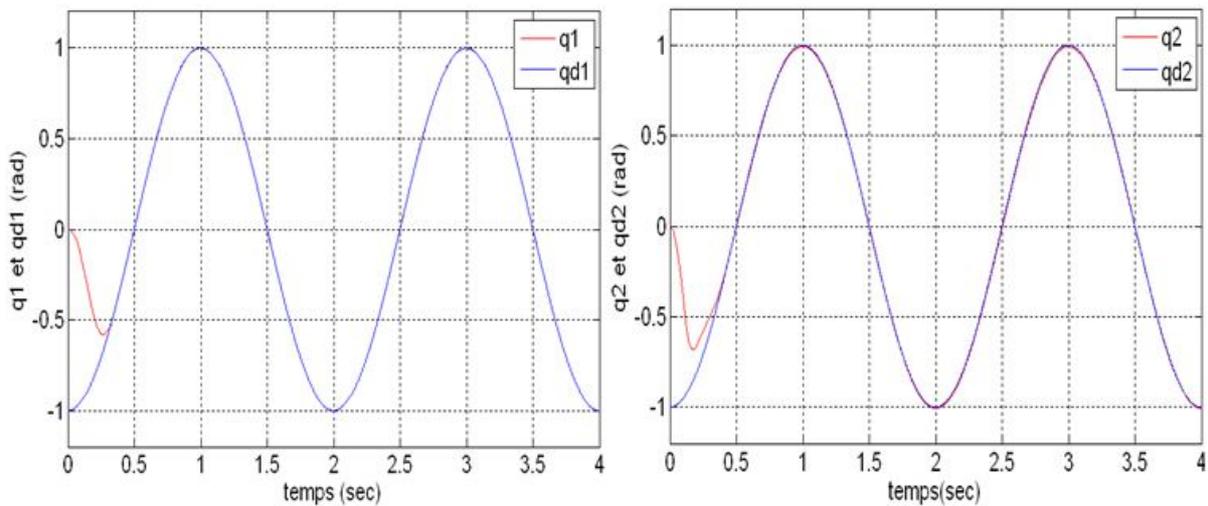


Figure IV.13 : Résultats de suivi de la trajectoire désirée pour les deux articulations -cas de poursuite-

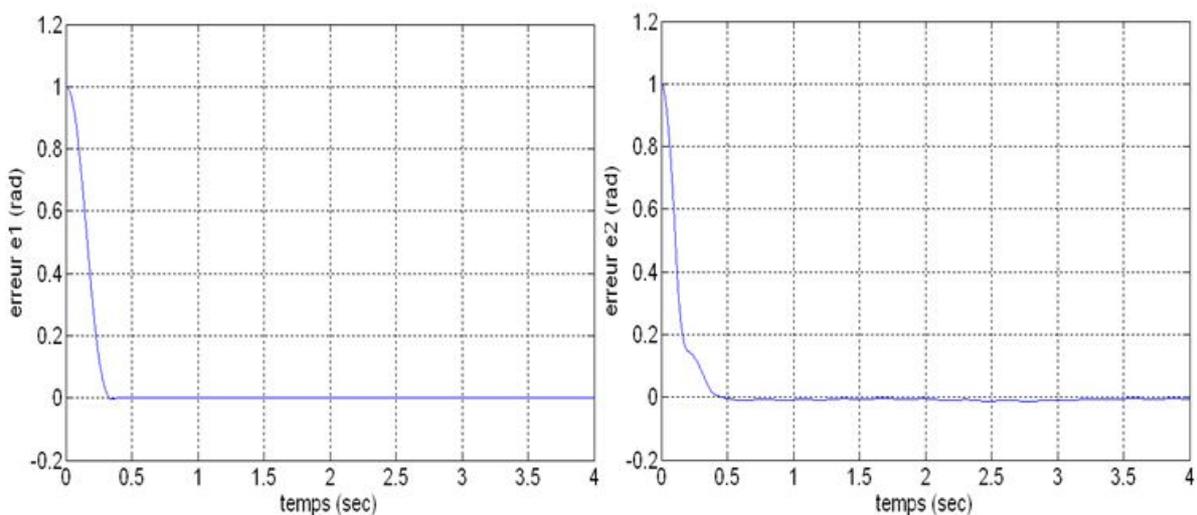


Figure IV.14 : Résultat de l'erreur de suivi de la trajectoire désirée pour les deux articulations -cas de poursuite-

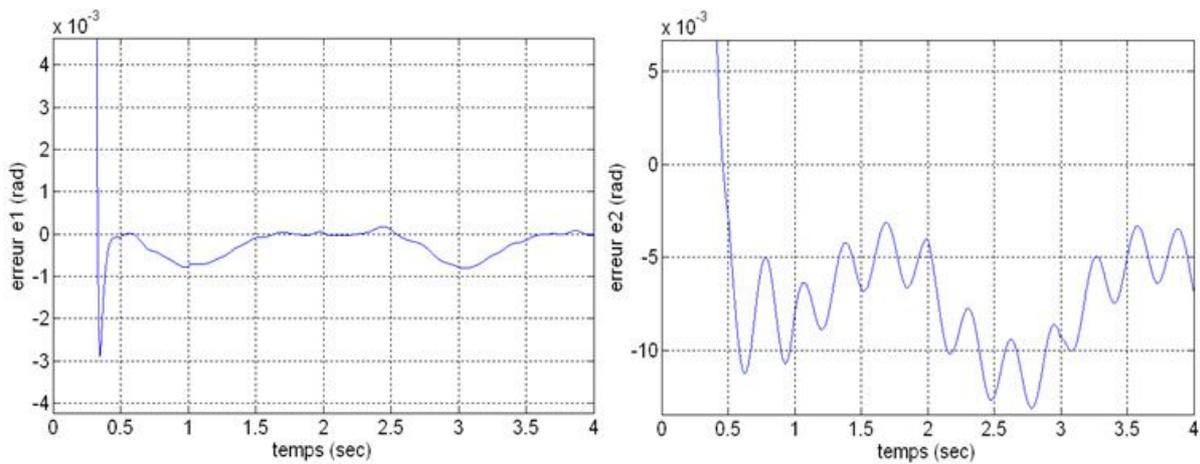


Figure IV.15 : Zoom sur l'erreur de suivi de la trajectoire désirée pour les deux articulations -cas de poursuite-

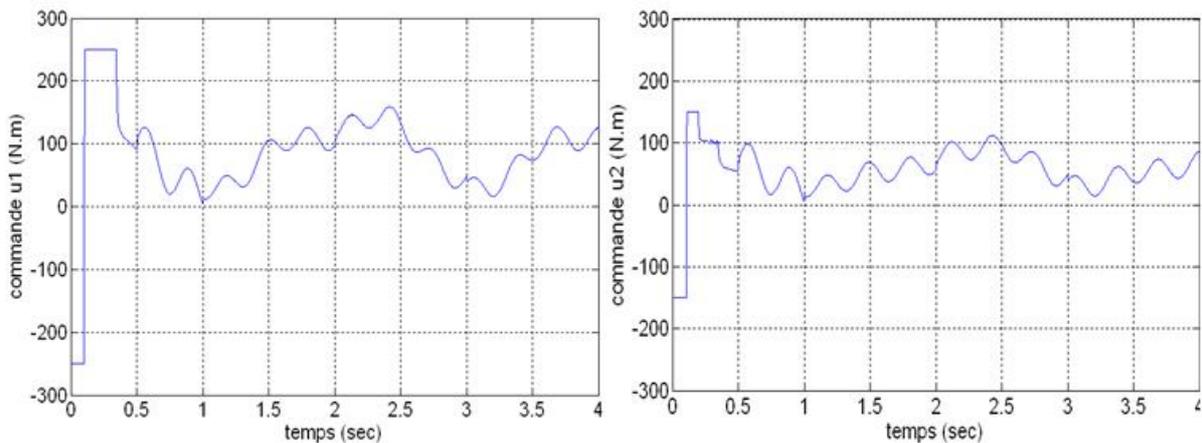


Figure IV.16 : Résultats d'erreur de suivi de la trajectoire désirée pour les deux articulations -cas de poursuite-

Les figures IV.3 à IV.9, montrent les résultats de simulation de l'algorithme hybride qui combine la commande par mode glissant et la commande backstepping. On constate une amélioration apportée aux algorithmes du mode glissant classique du point de vue l'amplitude de l'erreur de suivi et une petite diminution de l'amplitude du phénomène de réticence.

Les figures IV.10 à IV.16, représentent les résultats de simulation de la commande hybride backstepping-mode glissant fluide. La commande montre clairement la bonne conduite de cette approche avec l'élimination du phénomène de réticence.

IV.5 Conclusion :

Dans ce chapitre nous avons fait une combinaison de la commande backstepping avec la commande par mode glissant comme première approche, ensuite on a fait une approximation floue du comportement de la fonction discrète signe en obtenant un contrôleur hybride backstepping-mode glissant flou.

Pour valider ces méthodes, on les a simulées sur un robot manipulateur à 2 ddl avec un modèle incertain et perturbé. Les résultats de simulation montrent les mérites des commandes évoquées en présentant leurs robustesses. On notant que la commande hybride backstepping-mode glissant flou donne des bonnes performances surtout l'annulation du phénomène de réticence en conservant une petite erreur.

Conclusion générale

L'objectif assigné à ce travail était de chercher une commande robuste aux incertitudes de modélisation et perturbation extérieures, dédiée à la commande de position d'un robot manipulateur de type bras sériel de 2 ddl. Plusieurs approches ont été envisagées

La première étape du travail a porté sur l'application d'une approche générale pour la conception d'une commande robuste de position pour le système en utilisant deux approches. Dans la première on a appliqué l'algorithme de commande par mode glissant, et dans la deuxième on a appliqué le backstepping. L'avantage de ces deux commandes est qu'elles imposent un comportement moins sensible aux incertitudes et perturbations, cependant ces deux méthodes présentent quelque inconvénient : La fonction signe du mode glissant génère un phénomène indésirable appelé réticence « chattering ».

Comme deuxième approche et pour remédier au problème énoncé, on a introduit la logique floue sur l'algorithme du mode glissant, pour avoir un algorithme qui prend les performances de la logique floue et la robustesse du mode glissant. Cette approche a donné des performances acceptables et elle a minimisé l'effet du phénomène de réticence rencontré dans le mode glissant ordinaire. On a aussi ajouté à cette commande un superviseur flou pour donner une certaine souplesse à l'approche mode glissant flou en ce qui concerne les bornes des incertitudes.

Une autre approche qu'on a étudiée est la combinaison de la commande par mode glissant avec la structure récursive du backstepping pour assuré une bonne convergence et robustesse de la solution.

Le dernier algorithme traité consiste à changer la partie discrète de la commande hybride backstepping-mode glissant pour minimiser le phénomène de réticence observé dans la commande, d'où on a obtenu une commande hybride backstepping-mode glissant flou.

On a appliqué ces commandes sur un bras manipulateur à 2ddl de modèle incertain et perturbé, et on a pu prouver que la combinaison de ces algorithmes donne des résultats qu'on peut juger meilleurs.

Les objectifs préalablement fixés dans le cadre de ce travail ont été atteints, en présentant un ensemble de commande robuste aux perturbations en étudiant leur comportement face au problème de suivi de trajectoire.

En ce qui concerne les développements ultérieurs envisageables, nous pouvons mentionner les possibilités suivantes :

- Appliquer les méthodes d'optimisation sur les Algorithmes abordées dans ce mémoire afin d'avoir les meilleures performances possibles.
- Utiliser d'autres méthodes d'intelligence artificielle telles que : les réseaux de neurones, la logique floue type 2..etc.
- Appliquer ces méthodes sur des robots réels.

Références

- [1] Ahmed F. Amer, Elsayed A. Sallam, Wael M. Elawady, «Adaptive fuzzy sliding mode control using supervisory fuzzy control for 3 DOF planar robot manipulators », *Applied Soft Computing* 11 (2011) 4943–4953.
- [2] Asif Sabanovic, Leonid M. Fridman, Sarah Spurgeon, « Variable Structure Systems from principles to implementation », édition 2004.
- [3] Amar SI Ammour, « Contribution à la commande par modes glissants d'ordre fractionnaire », thèse de doctorat, spécialité Automatique, Université de Tizi-ouzou, 2011.
- [4] Abdelmalik BACHTARZI, « Commande Des Systemes A Structure Variable », thèse de doctorat, spécialité Electronique, Université de CONSTANTINE, 2011.
- [5] Azoui Nawal, « Commande non linéaire d'un bras manipulateur », thèse de magister en électronique, option : robotique, Université de Batna, 2009.
- [6] Abdelghani El Ougli, « Intégration des techniques floues à la synthèse de contrôleurs adaptatifs », thèse de doctorat en Automatique, Signaux et Systèmes, Fès, Maroc, 2009.
- [7] Cheng-Kai Lin, Tian-Hua Liu and Li-Chen Fu, «Adaptive Backstepping PI Sliding-Mode Control for Interior Permanent Magnet Synchronous Motor Drive Systems», *American Control Conference*, 2011.
- [8] Clarence W. de Silva, « Applications of fuzzy logic in the control of robotic manipulators », *Fuzzy Sets and Systems* 70 (1995) 223 234.
- [9] Cherifa BENNANI, « Stabilisation et Estimation de l'état des Systèmes Dynamiques non Linéaires et Applications », thèse de magistère en math, Option : Recherche Opérationnelle et Optimisation, TIZI-OUZOU, 2011.
- [10] Chun-Yi Su, Tin-Pui Leung, « A Sliding Mode Controller with Bound Estimation for Robot Manipulators», *IEEE Transactions on Robotics and Automation*, vol. 9, no. 2, April 1993.

- [11] Fourar Redha, « Détection de scènes et reconnaissance d'objets dans une opération d'assemblage robotisé », thèse de magister en électronique, Université de Batna, 2009.
- [12] Gangtao, « Adaptive Control: Design and Analysis », edition 2003.
- [13] George J.Klir, Bo Yuan, «Fuzzy Sets and Fuzzy Logic:Theory and Application»,edition 1995
- [14] Her-Terng Yau, Chieh-Li Chen « Chattering-free fuzzy sliding-mode control strategy for uncertain chaotic systems», Chaos, Solitons and Fractals 30 (2006) 709–718
- [15] Hassan K. Khalil, «Nonlinear Systems» 3eme edition, 2002.
- [16] J J Craig, «Introduction to Robotics: Mechanics and Control», 2eme ddition, 1989.
- [17] James Carvajal, Guanrong Chen, Haluk Ogmen, « Fuzzy PID controller: Design, performance evaluation, and stability analysis », Information Sciences 123 (2000) 249 270.
- [18] Jean-Jacques E. Slotine, Weiping Li, «Applied Nonlinear Control », edition 1991.
- [19] H.F. Ho, Y.K. Wong, A.B. Rad,« Robust fuzzy tracking control for robotic manipulators », Simulation Modelling Practice and Theory 15 (2007) 801–816.
- [20] Kai Michels, Frank Klawonn, Rudolf Kruse, Andreas Nürnberger, « Fuzzy Control», edition 2006.
- [21] Khlar Nait Chabane, « Exploitation de la redondance pour la commande coordonnée d'un manipulateur mobile d'assistance aux personnes handicapées », thèse de doctorat en robotique, Evry, FRANCE, 2006.
- [22] M. Roopaei, M. Zolghadri Jahromi, «Chattering-free fuzzy sliding mode control in MIMO uncertain systems», Nonlinear Analysis 71 (2009) 4430_4437.
- [23] Mojtaba Ahmadiéh Khanesar, Mohammad Teshnehlab, Mahdi Aliyari, Shoorehdeli, « Sliding Mode Control of Rotary Inverted Pendulum », Proceedings of the 15thMediterranean Conference on Control and Automation, July 27-29, Athens-Greece.
- [24] Magdy M. Abdelhameed, «Enhancement of sliding mode controller by fuzzy logic with application to robotic manipulators», Mechatronics 15 (2005) 439–458.
- [25] Mark W. Spong, Seth Hutchinson, and M. Vidyasagar,» Robot Dynamics and Control», 2eme edition, 2004.
- [26] Neil Munro, Frank L.Lewis, «Robot Manipulator Control: Theory and Practice»,2eme edition,2004

- [27] Nurkan Yagiz, Yuksel Hacioglu, «Robust control of a spatial robot using fuzzy sliding modes», *Mathematical and Computer Modelling* 49 (2009) 114_127.
- [28] Philippe Mullhaupt, « Introduction à l'Analyse et à la Commande des Systèmes Non Linéaires », édition 2007
- [29] Reza N. Jazar, «Theory of Applied Robotics», 2eme edition, 2010.
- [30] Seung-Bok Choi and Jung-Sik Kim, « A Fuzzy Sliding Mode Controller for Robust Tracking of Robotic Manipulators », *Mechatronics* Vol. 7, No. 2, pp. 199-216, 1997.
- [31] Samir Bouabdallah , Roland Siegwart, « Backstepping and Sliding-mode Techniques Applied to an Indoor Micro Quadrotor», *Proceedings of the 2005 IEEE, International Conference on Robotics and Automation, Spain, 2005.*
- [32] Selami Beyhan, Zsofia Lendek, Robert Babuska, Martijn Wisse, Musa Alci, « Adaptive Fuzzy and Sliding-Mode Control of a Robot Manipulator with Varying Payload», 2011 50th IEEE Conference on Decision and Control and European Control Conference (CDC-ECC), 2011.
- [33] S. Aloui, O. Pages, A. El Hajjaji, A. Chaari, Y. Koubaa, «Improved fuzzy sliding mode control for a class of MIMO nonlinear uncertain and perturbed systems», *Applied Soft Computing* 11 (2011) 820–826.
- [34] Thomas R. Kurfess, « Robotics and Automation Handbook », edition 2005.
- [35] T. C. Kuo, B. W. Hong, Y. J. Huang, C. Y. Chen, « Adaptive Fuzzy Controller Design for Robotic Manipulators with Sliding Mode Control », 2008 IEEE International Conference on Fuzzy Systems.
- [36] W. Li, X.G. Chang, F.M. Wahl, Jay Farrell, « Tracking control of a manipulator under uncertainty by FUZZY P+ID controller », *Fuzzy Sets and Systems* 122 (2001) 125–137.
- [37] Yanxia Zhou, Yuxiang Wu and Yueming Hu, «Robust Backstepping Sliding Mode Control of a Class of Uncertain MIMO Nonlinear Systems», 2007 IEEE International Conference on Control and Automation, Guangzhou, CHINA - May 30 to June 1, 2007.
- [38] Yuzheng Guo, Peng-Yung Woo, «Adaptive Fuzzy Sliding Mode Control for Robotic Manipulators», *Proceedings of the 42nd IEEE, Conference on Decision and Control, 2003.*
- [39] Chieh-Li Chen, Ming-Hui Chang, « Optimal design of fuzzy sliding-mode control: A comparative study », *Fuzzy Sets and Systems* 93 (1998) 37-48.

Modèle dynamique d'un bras manipulateur rigide à 2 ddl :

La structure du robot utilisé dans l'étude faite dans ce mémoire est la suivante :

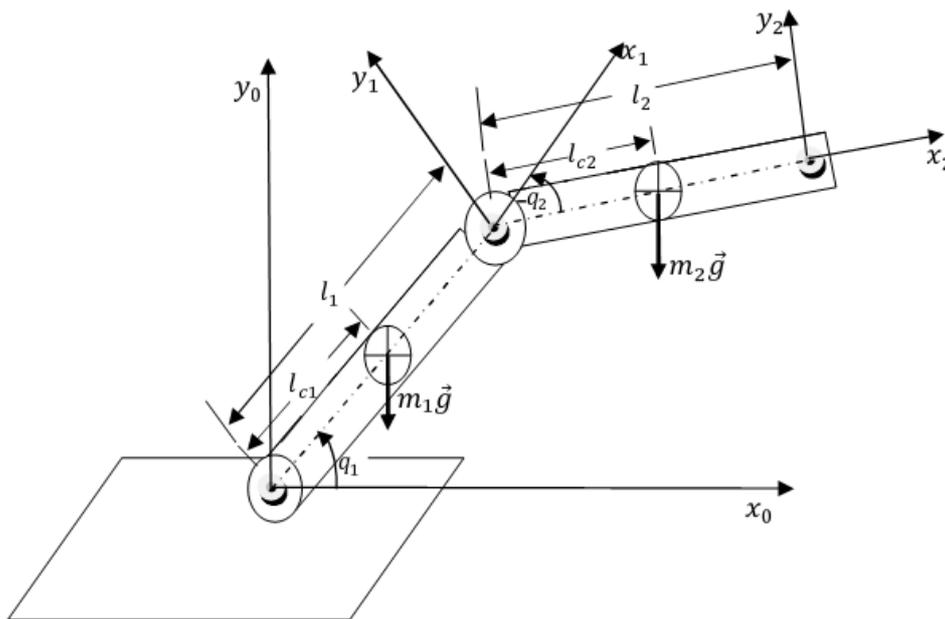


Figure : Structure d'un robot manipulateur à deux degré de liberté

Avec le modèle dynamique suivant :

$$M(q)\ddot{q} + C(q, \dot{q})\dot{q} + F_d\dot{q} + F_s(\dot{q}) + \tau_d(q, \dot{q}) = \tau$$

Ou :

$q = \begin{bmatrix} q_1 \\ q_2 \end{bmatrix}$: Représente le vecteur des variables articulaires

\dot{q}, \ddot{q} : Dérivé premier et seconde de q

$\tau = \begin{bmatrix} \tau_1 \\ \tau_2 \end{bmatrix}$: Le vecteur du couple appliqué aux articulations.

Matrice d'inertie :

$$M(q) = \begin{bmatrix} m_1 l_{c1}^2 + m_2 (l_1^2 + l_{c2}^2 + 2l_1 l_{c2} + 2l_1 l_{c1} c_2) + I_1 + I_2 & m_2 (l_{c2}^2 + l_1 l_{c1} c_2) + I_2 \\ m_2 (l_{c2}^2 + l_1 l_{c1} c_2) + I_2 & m_2 l_{c2}^2 + I_2 \end{bmatrix}$$

Matrice de Coriolis et de centrifuge :

$$C(q, \dot{q}) = \begin{bmatrix} -m_2 l_2 l_{c2} s_2 \dot{q}_2 & -m_2 l_2 l_{c2} s_2 (\dot{q}_1 + \dot{q}_2) \\ m_2 l_2 l_{c2} s_2 \dot{q}_1 & 0 \end{bmatrix}$$

Vecteur des forces de gravité :

$$G(q) = \begin{bmatrix} (m_1 l_{c1} + m_2 l_1) g c_1 + m_2 l_{c2} g c_{12} \\ m_2 l_{c2} c_{12} \end{bmatrix}$$

Vecteur des frottements dynamiques :

$$F_d \dot{q} = \begin{bmatrix} f_d \dot{q}_1 \\ f_d \dot{q}_2 \end{bmatrix} = \begin{bmatrix} 5 \dot{q}_1 \\ 5 \dot{q}_2 \end{bmatrix}$$

Vecteur des frottements statiques :

$$F_s(\dot{q}) = \begin{bmatrix} f_s \text{sign}(\dot{q}_1) \\ f_s \text{sign}(\dot{q}_2) \end{bmatrix} = \begin{bmatrix} 5 \text{sign}(\dot{q}_1) \\ 5 \text{sign}(\dot{q}_1) \end{bmatrix}$$

Vecteur d'incertitudes paramétriques et perturbations externes :

$$\tau_d = \begin{bmatrix} 20 + 20 \sin(20(t-1)) + 30 \sin(10(t-0.5)) + 20u(t-0.5) + 20u(t-1) \\ 20 + 20 \sin(20(t-1)) + 30 \sin(10(t-0.5)) + 20u(t-0.5) + 20u(t-1) \end{bmatrix}$$

Tel que : $c_1 = \cos(q_1)$, $s_2 = \sin(q_2)$ et $c_{12} = \cos(q_1 + q_2)$

Et les valeurs des paramètres utilisés dans notre modèle sont:

$$m_1 = 5kg, m_2 = 5kg, l_1 = 1kg \ m^2, l_2 = 1kg \ m^2, l_{c1} = 0.25m, l_{c2} = 0.25, l_1 = 0.5m, l_2 = 0.5m$$