

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



Université Mohamed Seddik BENYAHIA de Jijel

Faculté des Sciences et de la Technologie

Département d'Automatique

Mémoire de fin d'études pour l'obtention du diplôme de

Master en Automatique

Option : Automatique et Systèmes

Thème

Commande L1 Adaptative Floue Appliquée à
un Robot Manipulateur

Jury :

Pr. LABIOD Salim (Président)
Dr. BOUBAKIR Ahsene (Rapporteur)
Pr. BOUAKRIF Farah (Examineur)
Dr. SOUKKOU Ammar (Examineur)

Réalisé par :

SIFOUANE Fatih
BENSEGHIER Omar

Promotion juin 2018

« Remerciement »

Nous tenons à remercier tout d'abord « Allah », le tout puissant, qui nous a donné le courage, la force et la patience pour bien mener ce modeste travail.

Nous exprimons nos profonds remerciements à notre encadreur, "Mr. BOUBAKIR Ahcene" pour la confiance qui nous a accordé, pour sa patience, son entière disponibilité, ses encouragements et ses précieux conseils pour structurer le travail, nous le remercions vivement.

Nous lui disons

« Merci Oustad ».

Nous tenons à exprimer notre grande gratitude à toute personne qui, de près ou de loin, a contribué à la réalisation de ce travail.

Finalement, nos remerciements affectueux et chaleureux vont aussi à nos parents pour leur patience, et à toutes nos familles.

« *Dédicace* »

Je dédie ce modeste travail :

- ❖ *A ma très chère Mère et à la mémoire de mon cher et inoubliable père.*
- ❖ *A tout ma famille.*
- ❖ *A tous mes amis.*
- ❖ *A tous mes amis de la promotion 2018.*

FATIH

« *Dédicace* »

Je dédie ce modeste travail :

- ❖ *A mes très chers parents pour leur aide inestimable.*
- ❖ *A tout ma famille.*
- ❖ *A tous mes amis.*
- ❖ *A tous mes amis de la promotion 2018.*

OMAR

Tables des matières

Table des figures.....	iv
Liste des Tableaux.....	vi

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

Chapitre I : Généralités sur la logique floue

I.1. Introduction	4
I.2. Logique classique et logique flou.....	5
I.3. Les sous-ensembles flous.....	5
I.4. Variable linguistique	6
I.5. Opérations de base de la logique floue	7
I.6. Raisonnement flou.....	8
I.7. Système flou	9
I.7.1. Fuzzification	9
I.7.2. Base de règle floue	10
I.7.3. Moteur d'inférence	10
I.7.4. Défuzzification	10
I.8. Les différents types de la commande à base de la logique floue.....	10
I.8.1. La commande floue de type Mamdani	10
I.8.2. La commande floue de type Takagi-Sugeno	11
I.9. Approximation des fonctions par les systèmes flous	12
I.10. Conclusion.....	13

Chapitre II: Conception de la commande L_1 adaptative

II.1. Introduction.....	14
II.2. Principe de la commande adaptative	15
II.2.1 Commande adaptative indirecte.....	16
II.2.2 Commande adaptative directe	16
II.2.3 Méthode de Lyapunov	19

II.2.4. Commande adaptative directe avec estimateur d'état	20
II.3. Exemple d'application	22
II.3.1. Application de la commande adaptative sans estimateur d'état.....	23
II.3.2. Résultats et simulation	24
II.3.3 Application de la commande adaptative à modèle de référence avec estimateur d'état	24
II.4. Concept de la commande L1 adaptative	28
II.4.1. Architecture de la commande L1	28
II.5. Exemple de simulation.....	30
II.6. Conclusion	31

Chapitre III: Commande L_1 adaptative appliquée à un robot manipulateur

III.1. Introduction	32
III.2. Conception d'une commande L1 adaptative floue	32
III.2.1. Formulation du problème.....	32
III.2.2. Conception d'un approximateur flou	36
III.3. Structure de la commande L1 adaptative floue.....	37
III.4. Application de la commande L1 adaptative floue à un robot manipulateur ..	38
III.4.1. Résultats de simulation	40
III.4.3 Interprétation des résultats	42
III.5. Conclusion.....	45

Chapitre IV: Etude comparative

IV.1. Introduction	46
IV.2. Commande par mode de glissement SMC.....	46
IV.2.1. Commande par modes glissants d'ordre 1	46
IV.2.2. Synthèse de la loi de commande.....	46
IV.2.2.1. Choix de la surface de glissement.....	47
IV.2.2.2. Condition d'existence du mode de glissement.....	47
IV.2.2.3. Loi de commande.....	48
IV.2.2.4. Avantages et inconvénients de la commande par mode glissant	48
IV.2.3. Commande par mode de glissement appliquée à un robot manipulateur	49
IV.2.3.1. Synthèse des lois de commande SMC pour le robot manipulateur.....	50
IV.2.4. Résultats de simulation.....	52

IV.2.4.1. Interprétation des résultats de simulation.....	54
IV.3.2. Principe de la commande backstepping.....	55
IV.3.3. Fonctions de Lyapunov.....	55
IV.3.4. Commande backstepping appliquée à un robot manipulateur	56
IV.3.4.1. Synthèses des lois de commandes pour le robot manipulateur	56
IV.3.5. Résultats de simulation.....	60
IV.3.6. Interprétation des résultats de simulation	61
IV.4. Étude comparative.....	61
IV.5. Conclusion.....	65
Conclusion générale.....	66
Références bibliographiques.....	67

Tables des figures

Figure I.1: Classification des températures d'une pièce en deux ensembles.	5
Figure I .2 : Représentation graphique d'un ensemble classique et d'un ensemble flou.	6
Figure I .3 : Structure générale d'un système flou.	9
Figure II.1 : Principe des systèmes de commande adaptative.	15
Figure II.2 : Architecture de la commande adaptative indirecte.	16
Figure II.3 : Architecture de la commande adaptative directe.	17
Figure II.4 : Structure de la MRAC en boucle fermée sans estimateur d'état.	19
Figure II.5 : Structure de la MRAC en boucle fermée avec estimateur d'état.	22
Figure II.6 : Système en BF avec MRAC contrôleur de type intégrateur.	22
Figure II.7: Résultats de simulation en boucle fermée sans estimateur d'état.	24
Figure II.8 : résultats de simulation en boucle fermée avec estimateur d'état.	26
Figure II.9 : Diagramme de Nyquist des fonctions de transfert.	26
Figure II.10 : Système en boucle BF avec le contrôleur L1 adaptatif.	27
Figure II.11 : L'effet du gain d'adaptation sur la marge de la stabilité.	27
Figure II.12 : Structure de la commande adaptative en BF.	29
Figure II.13 : Résultats de simulation.	30
Figure III.1 : Robot manipulateur de deux degrés de libertés.	39
Figure III.2 : Contrôleur L1 adaptatif flou appliqué à un robot manipulateur.	40
Figure III.3 : Position articulaires et position désirées.	41
Figure III.4 : Les commandes appliquées au robot manipulateur.	41
Figure III.5 : Les erreurs de poursuite.	42
Figure III.6 : Les erreurs filtrées et leurs prédictions.	42
Figure.III.7 : Positions articulaires et positions désirées.	43
Figure.III.8 : Les commandes appliquées au robot manipulateur.	43
Figure.III.9 : Les erreurs de poursuite.	44

Figure III.10 : Les erreurs filtrées et leurs prédictions.	44
Figure IV.1 : Différents modes de convergence pour la trajectoire d'état.	46
Figure IV.2 : Représentation de la commande discontinue.	48
Figure.IV.3: Plan de phase réel.....	49
Figure.IV.4 : Position articulaires et position désirées.	53
Figure.IV.5 : Les erreurs de poursuite.....	53
Figure.IV.6 : Les commandes appliquées au robot manipulateur.	54
Figure.IV.7 : Les surfaces de glissement.....	54
Figure.VI.8 : Position articulaires et position désirées.	60
Figure.IV.9 : Les erreurs de poursuite.....	60
Figure.IV.10 : Les commandes appliquées au bras manipulateur.	61
Figure.VI.11 : Position articulaires et position désirées (selon l'angle q_1)	63
Figure.VI.12 : Position articulaires et position désirées (selon l'angle q_2).	63
Figure.VI.13 : Les erreurs de poursuite.....	64
Figure.IV.14 : Les commandes appliquées au robot manipulateur : commande par mode de glissement	64
Figure.IV.15 : Les commandes appliquées au robot manipulateur : commande Backstepping.	64
Figure.IV.16 : Les commandes appliquées au robot manipulateur : commande L1 adaptative.	65

Liste des tableaux

Tableau .IV.1 : Étude comparative entre les commandes développées pour le robot manipulateur sans test de robustesse.....	62
Tableau .IV.2 : Étude comparative entre les commandes développées pour le robot manipulateur avec test de robustesse.....	62



Introduction générale

Introduction générale

La commande adaptative est introduite dans le but de maintenir les performances de commande d'un système performante acceptable malgré l'effet de perturbations et des incertitudes. La notion de commande adaptative regroupe un ensemble de techniques utilisant les signaux qui sont disponibles dans le système en boucle fermée afin d'ajuster automatiquement et en temps réel les paramètres du contrôleur. Cependant, les méthodes de conception de la commande adaptative standard sont limitées aux systèmes non linéaires qui peuvent être linéairement paramétrisés (i.e. les non linéarités du système non linéaire connue par un vecteur de paramètres inconnu). Malheureusement, il est souvent très difficile voire impossible d'obtenir cette forme de paramétrisation linéaire notamment pour un système physique complexe incertain [1].

La commande adaptative à modèle de référence avec et sans estimateur joue un rôle très important dans la commande des systèmes. Dans un système de commande adaptatif, une adaptation rapide diminue la robustesse du contrôleur. Récemment, une approche de commande adaptative baptisée commande L_1 adaptative est introduite. Cette approche permet de faire une adaptation rapide tout en préservant la robustesse. Dans ce mémoire, premièrement, nous présentons les différents concepts de cette approche de commande avec des exemples d'application, ensuite nous présentons une approche de commande L_1 adaptative floue robuste.

La méthode de commande L_1 adaptative a attiré beaucoup d'attention au cours des dernières années en raison de son caractère robuste et capacité à fournir une haute performance de contrôle aussi bien en régime transitoire qu'en régime permanent, malgré la présence des incertitudes et des dynamiques non modélisée et des perturbations [2]. En fait, la commande L_1 adaptative est une amélioration de la commande adaptative à modèle de référence (MRAC) [3] avec prédicteur et en utilisant un filtre passe bas, qui découple la vitesse d'adaptation de la robustesse. Par conséquent, et contrairement au schéma de contrôle MRAC, la commande L_1 adaptative permet une adaptation rapide sans sacrifier la robustesse, et fournit une réponse transitoire uniformément limitée, en plus une stable suivie des signaux d'entrée et de sortie. Son principe a été introduit en 2006 par Cao et Hovakimyan [4] [5]. Depuis, le développement de contrôleurs basés sur la méthodologie de la commande L_1 adaptative a fait l'objet des nombreuses publications. Cao et Hovakimyan [6][7] ont proposé deux contrôleurs L_1 adaptatifs avec retour d'état pour les systèmes linéaires, avec une incertitude constante et avec des paramètres variables inconnus dans le temps et des perturbations bornées. Ils ont également considéré une formulation avec retour de sortie pour un système linéaire (SISO) avec perturbations variant dans le temps [8]. Dans [9], Cao et Hovakimyan ont discuté la stabilité relative et la robustesse de l'architecture L_1 adaptative. Une étude comparative intéressante entre le contrôleur L_1 adaptatif et le MRAC direct et indirect, en utilisant les outils de domaine fréquentiels, est donnée par Kharisov et al [10]. Quelques articles de recherche de Cao et Hovakimyan [4] [6] [8] [9] ont été regroupé dans un livre, publié en 2010, qui détaille le

principe de la commande L_1 adaptative et son application aux classes des systèmes linéaires et non linéaires avec des études de performance. Le choix de la structure et les coefficients du filtre rétroaction joue un rôle important dans l'efficacité d'un contrôleur L_1 adaptatif, ceci est largement discuté dans la littérature [2] [11].

Dans ce mémoire, un contrôleur L_1 adaptatif qui utilise une approximation floue comme moyen supplémentaire pour réaliser une bonne estimation de l'incertitude et améliorer les performances résultantes est introduit pour une classe des systèmes MIMO non linéaires affines. Le fait d'incorporer une approximation floue dans l'architecture de la commande L_1 adaptative, au but d'estimer la non-linéarité inconnue, permet de réduire le nombre paramètres de conception, qui dépendent du choix de gain d'adaptation, tout en gardant des bonnes performances. Généralement, dans la conception de la commande L_1 adaptative, le prédicteur d'état utilisé est une fonction d'approximation des incertitudes. De plus, un système flou peut être conçu pour être une approximation universelle des fonctions continues [12] [13], une approximation basée sur un système flou est plus efficace, ce qui conduit à une petite erreur de prédiction et donc une meilleure performance de commande.

Contrairement aux approches de commande L_1 adaptative qui utilisent seulement des lois d'adaptation de type projection rapide et nécessitant un petit pas d'intégration, ce qui impose des contraintes rigoureuses sur la vitesse du processeur pour sa mise en œuvre [14], le contrôleur L_1 adaptatif flou présenté peut atteindre les mêmes performances avec un gain d'adaptation et un pas d'intégration moyen. La commande présentée garantit la stabilité du système en boucle fermée et assure le suivi des signaux d'entrée et de sortie d'un système de référence virtuelle stable. La capacité du contrôleur proposé est examinée en simulation dans la commande d'un bras manipulateur.

Organisation du mémoire

Ce mémoire est divisé en quatre chapitres résumés ci-dessous.

Chapitre 1 :

Ce chapitre est consacré principalement à la présentation des notions de base de la logique floue et des systèmes flous. Aussi, nous essayons de rappeler le théorème d'approximation universelle relatif aux systèmes flous.

Chapitre 2 :

Ce chapitre présente deux techniques différentes de commande adaptative, dans un premier lieu nous parlerons de la structure de commande adaptative à modèle de référence sans et avec estimateur. Ensuite, nous allons présenter le principe de la technique de commande L_1 adaptative. À la fin de ce chapitre, une étude comparative en simulation sur un système scalaire est réalisée entre la commande adaptative à modèle de référence et la commande L_1 adaptative et ceci pour montrer l'efficacité de cette dernière.

Chapitre 3 :

Dans ce chapitre nous présentons la commande L1 adaptative pour une classe des systèmes MIMO non linéaires affines à la commande. Ensuite, nous examinons les performances de cette commande sur un bras manipulateur.

Chapitre 4 :

Dans ce dernier chapitre nous présentons deux méthodes classiques très connues et utilisées, la commande par mode de glissement et la commande par backstepping. Ensuite, on va appliquer ces deux commandes pour une étude de simulation sur le bras manipulateur. Finalement, nous allons faire une étude comparative entre la commande L1 adaptative et les deux commandes citées précédemment et ceci pour mettre en évidence l'efficacité et la robustesse de la commande L1 adaptative floue présentée.

Conclusion :

Nous finirons par une conclusion générale, on parle en générale sur les résultats obtenus et les perspectives pour ce travail.



Chapitre .I

Généralités sur la logique floue

Chapitre .I

Généralités sur la logique floue

Chapitre I. Généralités sur la logique floue

I.1. Introduction

De nos jours, la logique floue (fuzzy logic) est un axe de recherche important sur lequel se focalisent de nombreux scientifiques. Des retombées technologiques sont d'ores et déjà disponibles, tant dans le domaine grand public (appareils photos, machines à laver, fours à micro-onde), que dans le domaine industriel (classification, aide à la décision, réglage et commande de processus, complexes liés à l'énergie, aux transports, à la transformation de la matière, à la robotique, aux machines-outils).

Le terme d'ensemble flou apparaît pour la première fois en 1965 lorsque le professeur Lotfi A. Zadeh, de l'université de Berkeley aux USA, publie un article intitulé « Ensembles flous » (Fuzzy sets). Il a introduit la notion de sous ensemble flou pour fournir un moyen de représentation et de manipulation des connaissances imparfaitement décrites, vagues ou imprécises. À cette époque, la théorie de la logique floue n'a pas été prise au sérieux excepté par quelques experts. Dès 1975, Mamdani et Assilian publient les premiers résultats permettant une exploitation de cette théorie dans des systèmes de réglage. En utilisant une structure de contrôleur relativement simple, ils ont obtenu de meilleurs résultats lors de la commande de certains processus que ceux fournis par un régulateur standard de type PID [15].

C'est au Japon, où la recherche n'est pas seulement théorique mais également très applicative, que la logique floue connaît son véritable essor. À la fin des années 1980, c'est d'un véritable boum qu'il faut parler. Les produits grand public, machines à laver, appareils photographiques et autres caméscopes estampillés « fuzzy logic » ne se comptent plus. Dans l'industrie, le traitement des eaux, les grues portuaires, les métros, les systèmes de ventilation et de climatisation sont touchés. Enfin, des applications existent dans des domaines très différents tels que la finance ou le diagnostic médical. À partir de 1990, c'est en Allemagne que des applications apparaissent en grand nombre ainsi qu'à une moindre échelle aux USA. Enfin en France, la logique floue devient aujourd'hui une réalité [16].

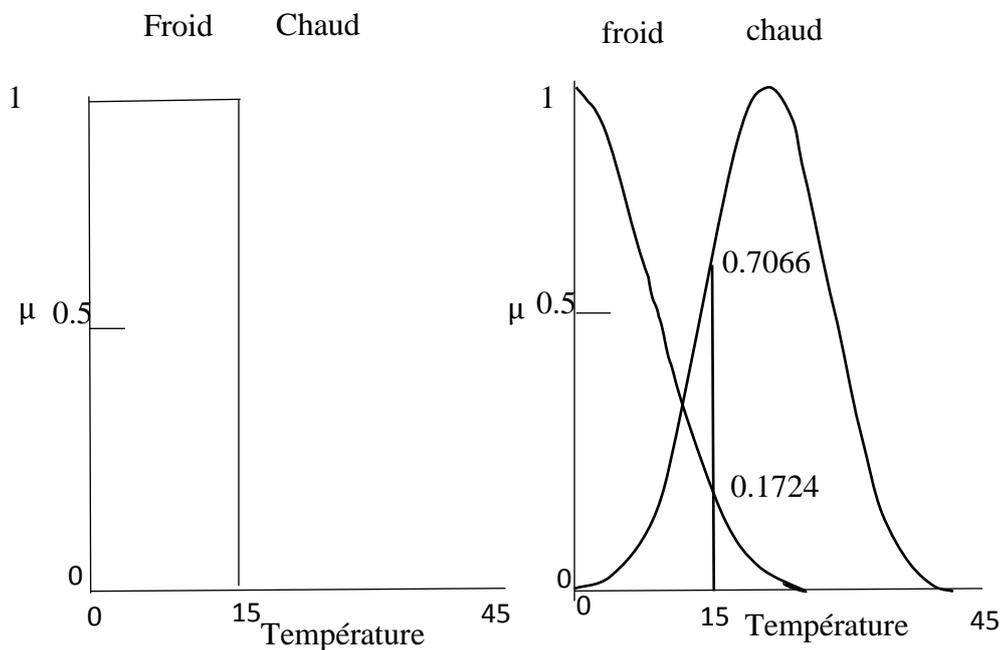
L'utilisation de la commande floue est particulièrement intéressante lorsqu'on ne dispose pas de modèle mathématique précis du processus à commander ou lorsque ce dernier présente de fortes non linéarités ou imprécisions. Les systèmes flous permettent d'exploiter et de manipuler efficacement les informations linguistiques émanant de l'expert humain grâce à un fondement théorique important [17] [18]. En plus, le système mis en œuvre peut être intégré facilement dans une boucle de commande ou d'identification.

Dans ce chapitre que nous intitulons généralités sur la logique floue, nous exposons un bref rappel sur la théorie des ensembles flous et de la logique floue. Ensuite, nous présentons la structure des systèmes flous en général.

I.2. Logique classique et logique floue

En logique floue, un ensemble flou est la généralisation de la notion de l'ensemble en logique classique (binaire). Ils sont caractérisés par des fonctions d'appartenance. Dans le cadre de la logique classique, une proposition est soit vraie, soit fausse (1 ou 0).

Par exemple, la logique classique peut facilement partitionner la température d'une pièce en deux sous-ensembles, «moins de 15 degrés» et «15 degrés ou plus». La figure 1.1a montre le résultat de cette partition. Toutes les températures de moins de 15 degrés sont alors considérées comme appartenant à l'ensemble «moins de 15 degrés». On leur affecte une valeur de 1. Toutes les températures atteignant 15 degrés ou plus ne sont pas considérées comme appartenant à l'ensemble «moins de 15 degrés». On leur attribue une valeur de 0. Cependant, le raisonnement humain s'appuie fréquemment sur des connaissances ou des données inexactes, incertaines ou imprécises. Une personne placée dans une pièce dont la température est soit de 14.95 degrés soit de 15.05 degrés, ne fera certainement pas de distinction entre ces deux valeurs. Cette personne sera pourtant capable de dire si la pièce est «froide» ou «chaude», sans pour cela utiliser de température limite ni de mesure précise [15].



(a) Deux ensembles selon la logique classique (b) Deux ensembles selon la logique floue

Figure I.1: Classification des températures d'une pièce en deux ensembles.

I.3. Les sous-ensembles flous

La logique floue repose sur la théorie des ensembles flous, qui est une généralisation de la théorie des ensembles classiques comme il est représenté sur figure I.2. Par abus de langage, suivant les utilisations de la littérature, nous utiliserons indifféremment les termes sous-ensembles flous et ensembles flous. Les ensembles classiques sont également appelés ensemble nets, par opposition à flous, et de même la logique classique est également appelée logique booléenne ou binaire [19].

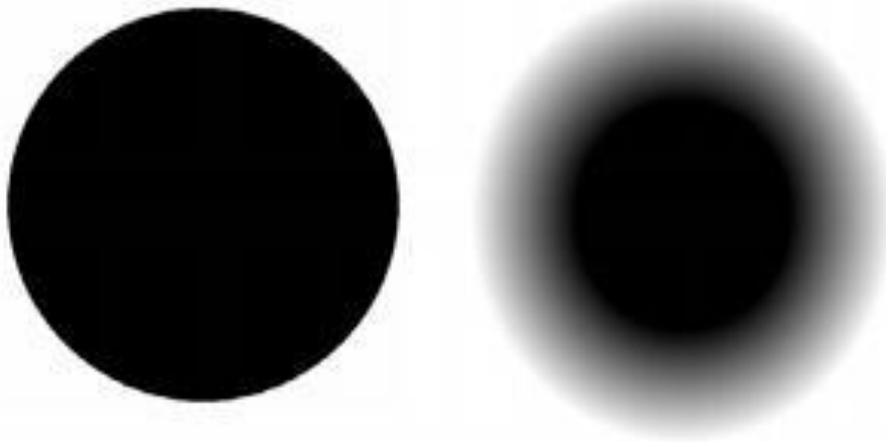


Figure I.2 : Représentation graphique d'un ensemble classique et d'un ensemble flou.

I.4. Variable linguistique

Une variable linguistique, comme son nom l'indique, est une variable dont les valeurs sont des mots ou des expressions dans un langage naturel ou artificiel.

Si on considère la couleur d'un objet comme une variable. Ses valeurs (rouge, vert, etc...) peuvent être interprétées comme des étiquettes des sous-ensembles d'un univers d'objets. Il est important de noter que la caractérisation de la variable couleur par une étiquette telle que rouge est beaucoup moins précise que la valeur numérique de la longueur d'onde d'une couleur particulière.

En général les valeurs d'une variable linguistique peuvent être générées à partir d'un terme primaire (eg. Chaud), de son antonyme (i.e., froid), d'une collection de modificateurs (eg. Très ou moins, moyens, etc...) et des connectifs « et » et « ou ». La variable linguistique est souvent caractérisée par le triplet suivant : $(x, L(x), X)$ où :

- ✓ **x** : le nom de la variable linguistique (ex ; âge, température, erreur, etc...).
- ✓ **L(x)** : l'ensemble des valeurs ou variables linguistique que peut être prendre la variable x (ex ; pour la variable température ; $L(x) = \{\text{très chaud, chaud, tiède, froid, très froid}\}$).
- ✓ **X** : univers de discours de x est un sous ensemble des réels, regroupe l'intervalle numérique que peut correspondre à la variable linguistique.

I.5. Opérations de base de la logique floue

La représentation formelle des ensembles flous par des fonctions d'appartenance a permis de généraliser les opérateurs des ensembles classiques au cas flou.

Soit A et B deux ensembles flous définis dans l'univers de discours X par les fonctions d'appartenance $\mu_A(x)$ et $\mu_B(x)$ respectivement.

- **Égalité floue**

Deux ensembles flous A et B sont égaux ($A=B$) si et seulement si

$$\forall x \in X : \mu_A(x) = \mu_B(x) \quad (\text{I.1})$$

- **Complément floue**

Le complément \bar{B} de l'ensemble flou B est défini par la fonction d'appartenance :

$$\forall x \in X : \mu_{\bar{B}}(x) = 1 - \mu_B(x) \quad (\text{I.2})$$

- **Union floue**

L'intersection de deux ensembles flous A et B est un ensemble flou ($A \cup B$) de fonction d'appartenance :

$$\forall x \in X : \mu_{A \cup B}(x) = \mu_A(x) + \mu_B(x) \quad (\text{I.3})$$

avec :

$$\mu_A(x) + \mu_B(x) = \text{MAX}(\mu_A(x), \mu_B(x)) \quad (\text{I.4})$$

- **Intersection floue**

L'intersection de deux ensembles flous A et B est un ensemble flou ($A \cap B$) de fonction d'appartenance :

$$\forall x \in X : \mu_{A \cap B}(x) = \mu_A(x) * \mu_B(x) \quad (\text{I.5})$$

Le symbole « * » représente la norme triangulaire [21]. La norme triangulaire caractérise le plus petit des degrés avec lesquels il appartient à A et à B . les normes les plus utilisés dans le domaine de la commande sont :

$$\mu_A(x) * \mu_B(x) = \min(\mu_A(x), \mu_B(x)) \quad (\text{I.6})$$

$$\mu_A(x) * \mu_B(x) = (\mu_A(x) \times \mu_B(x)) \quad (\text{I.7})$$

- **Produit cartésien**

Soit C_1, \dots, C_n des ensembles flous respectivement définis sur X_1, \dots, X_n , le produit cartésien de $C_1 \times \dots \times C_n$ est un ensemble flou défini sur $X_1 \times \dots \times X_n$ et d'une fonction d'appartenance :

$$\mu_{C_1 \times \dots \times C_n}(x_1, x_2, \dots, x_n) = \min(\mu_{C_1}(x_1), \dots, \mu_{C_n}(x_n)) = (\mu_{C_1}(x_1) * \dots * \mu_{C_n}(x_n)) \quad (I.8)$$

- **Relation floue**

Une relation floue représente le degré de présence, ou d'absence d'une association entre les éléments de deux ou plusieurs ensembles flous. Une relation floue d'ordre n est un ensemble flou défini sur $X_1 \times \dots \times X_n$ par l'expression suivante :

$$R_{x_1 \times \dots \times x_n} = \{((X_1, \dots, X_n), \mu_R(x_1, \dots, x_n)) / (x_1, \dots, x_n) \in X_1, \dots, X_n\} \quad (I.9)$$

- **Composition des règles floues**

Soit R et S deux relation floues définies respectivement dans $U \times V$ et $V \times W$. La composition de R et S est un ensemble flou, symbolisé par $R \circ S$, de fonction d'appartenance :

$$\mu_{R \circ S}(u, w) = \{[(u, w), \sup(\mu_R(u, v) * \mu_S(v, w))], u \in U, v \in V, w \in W\} \quad (I.10)$$

- **Implication floue**

L'implication floue est un opérateur qui permet d'évaluer le degré de vérité d'une règle de la forme :

Si x est A Alors y est B

À partir des valeurs de la prémisse d'une part, et de celle de la conclusion d'autre part. Ce degré de vérité est évalué à partir des degrés d'appartenance de x à A et y à B comme suit :

$$\mu_R(x, y) = \text{imp}(\mu_A(x), \mu_B(y)) \quad (I.11)$$

Les opérateurs les plus utilisés en commande floue sont les implications de Mamdani et de Larsen

➤ Implication de Mamdani :

$$\mu_R(x, y) = \min(\mu_A(x), \mu_B(y)) \quad (I.12)$$

➤ Implication de Larsen :

$$\mu_R(x, y) = \mu_A(x) \times \mu_B(y) \quad (I.13)$$

I.6. Raisonnement flou

On rappelle qu'en logique classique le modus ponens permet, à partir de la règle « Si x est A alors y est B » et du fait « x est A », de conclure le fait « y est B », qui sera ajouté à la base des faits. Zadeh a étendu ce principe au cas flou, le principe que l'on appelle alors modus

ponens généralisé [20] [21]. Le modus ponens et le modus généralisé se résument comme suit :

Modus ponens

Fait : x est A

Règle : SI x est A ALORS y est B

Conclusion : y est B

Modus ponens généralisé

x est A'

y est B'

À partir de la règle « SI A ALORS B » et du fait « A' », on déduit un nouveau fait « B' » qui est caractérisé par un ensemble flou dont la fonction d'appartenance est donnée par :

$$\mu_{B'}(y) = \text{SUP}_x(\mu_{A'}(x) * \mu_R(x, y)) \quad (\text{I.14})$$

Les fonctions d'appartenance $\mu_{A'}(x)$ et $\mu_R(x, y)$ caractérisent respectivement le fait « A' » et la règle.

I.7.1 Système flou

Un système flou est un système MISO qui se compose de quatre blocs essentiels, la fuzzification, base de règle, moteur d'inférence (l'inférence) et la défuzzification. La figure suivante illustre la configuration de base d'un système flou :

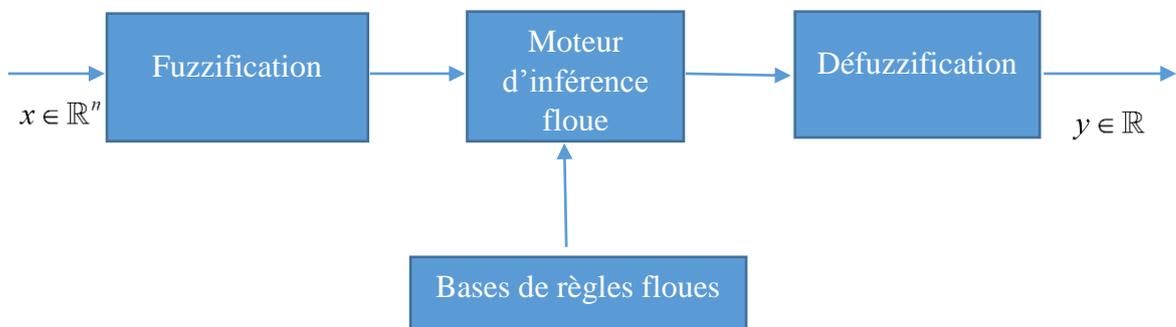


Figure I.3 : Structure générale d'un système flou.

I.7.2. Fuzzification

La fuzzification transforme la grandeur physique d'entrée en une grandeur floue. Elle est réalisée dans l'interface d'entrée du contrôleur flou. Durant cette phase, les informations issues du système sont tout d'abord normalisées. Ensuite, les données normalisées sont transformées en qualifications linguistiques, en utilisant des règles sémantiques définies par un expert. Durant la phase de normalisation, chaque mesure issue du système est modifiée pour fournir une valeur appartenant à un univers du discours relativement simple [15].

I.7.2. Base de règle floue

La base de règles floues, ou base de connaissances, contient les règles floues décrivant le comportement du système ; elle est le cœur du système entier dans le sens où tous les autres composants sont utilisés pour interpréter et combiner ces règles pour former le système final. Ces règles peuvent être fournies par un expert ou peuvent être extraites de données numériques. Dans les deux cas, les règles prennent la forme « si prémisse Alors conclusion ». D'une manière générale, la base de règles d'un système flou doit respecter les conditions de complétude et de consistance afin d'assurer le bon fonctionnement de ce dernier. Une base de règles d'un système flou est dite complète si, pour chaque vecteur d'entrée, il existe au moins une règle floue activée. Afin d'assurer cette propriété, les fonctions d'appartenance doivent couvrir tout l'espace des variables d'entrée. Une base de règles floues ayant la même prémisse mais des conclusions différentes. La propriété de consistance permet d'éviter les contradictions dans une base de règles [22].

I.7.3. Moteur d'inférence

Cette partie est le moteur d'inférence qui se chargera d'appliquer chacune des règles d'inférences. Ces règles d'inférences représentant les connaissances que l'on a du système dû à l'expertise humaine. Chaque règle générera une commande de sortie.

Le moteur d'inférence floue transforme, à l'aide des techniques de raisonnement flou, la partie floue issue de la fuzzification en une nouvelle partie floue.

I.7.4. Défuzzification

C'est l'étape qui permet de fusionner les différentes commandes générés par le moteur d'inférence pour lui donner qu'une seule commande de sortie et de transformer cette variable linguistique qualitative de sortie en donnée numérique quantitative, ce qui revient donc à trouver la meilleure valeur quantitative en fonction des fonctions d'appartenances des variables linguistiques.

I.8. Les différents types de la commande à base de la logique floue

En se basant sur les différentes règles de la commande floue et de leurs méthodes de génération [23], les approches de commande en utilisant la logique floue peuvent être classées dans les catégories suivantes :

I.8.1. La commande floue de type Mamdani

Dans un système flou de Mamdani (Conventionnel) les règles sont de type :

$$R_{(i,j,k)} : \text{Si } x \text{ est } A_i \text{ et } y \text{ est } B_j \text{ Alors } z \text{ est } C_k \quad (\text{I.15})$$

où A_i, B_j est C_k représentent des termes linguistiques auxquels est attribuée une signification floue que l'on suppose normalisée.

Pour des entrées numériques précises x_0 et y_0 un sous-ensemble flou de sortie, noté F , est généré selon la relation :

$$\mu_F(z) = \max_{(i,j,k) \in I} \min(\mu_{A_i}(x_0), \mu_{B_j}(y_0), \mu_{C_k}(z)) \quad (\text{I.16})$$

μ est appelée fonction d'appartenance.

L'opération $\max_{(i,j,k) \in I}$ correspond à l'application de l'opérateur max sur l'ensemble des règles actives. La valeur numérique de sortie délivrée par le système flou est obtenue par défuzzification, classiquement implantée par la méthode du centre de gravité :

$$z = \frac{\int z \mu_F(z) dz}{\int \mu_F(z) dz} \quad (\text{I.17})$$

Dans ce type de systèmes flous, la prémisse et la conclusion sont floues. Après l'inférence, le résultat est un ensemble flou caractérisé par sa fonction d'appartenance. Afin d'obtenir la valeur réelle de la commande à appliquer il faut passer par une étape de « défuzzification ».

Les règles de la commande floue « Si-Alors » sont obtenues à partir de l'expertise d'un opérateur. Quoique la performance d'une telle technique de commande soit généralement satisfaisante en pratique le non garanti de la stabilité du système de commande en boucle fermée est souvent critiquée.

Plusieurs approches ont été proposées pour étudier le problème de stabilité. L'idée principale de ces approches est de considérer le contrôleur flou comme un contrôleur non linéaire et d'utiliser la théorie des systèmes non linéaire pour l'analyse de la stabilité.

Les avantages du système flou de MAM peuvent être résumés par les points suivants :

- L'avantage majeur de ces modèles est que les prémisses et les conclusions des règles sont de nature entièrement linguistique. Ceci les rend facilement compréhensibles, et donne une interprétation sémantique claire au modèle.
- Autre avantage réside dans la flexibilité dans l'implémentation : Ceci est dû à la liberté dans le choix des formes d'implication floues, du fuzzificateur et du défuzzificateur.

En revanche, le système flou de MAM possède les inconvénients suivants:

- Un grand nombre de règles est souvent nécessaire pour approcher le comportement global d'un système. Par conséquent, le nombre de paramètres à identifier est élevé.
- Le deuxième inconvénient, du moins du point de vue purement théorique, réside dans la mise en œuvre heuristique, ne prenant en compte aucun critère de stabilité ou de robustesse de la théorie de la commande.

I.8.2. La commande floue de type Takagi-Sugeno

Les contrôleurs flous de Takagi–Sugeno sont, comme ceux de Mamdani, construits à partir d'une base de règles “Si ... Alors ...”. Les prémisses sont toujours exprimées linguistiquement et donc similaires à celles utilisées dans un contrôleur flou de Mamdani alors que les conclusions sont de nature numérique. Un système flou de type Takagi-Sugeno (T-S) utilise des règles écrites de la manière suivante :

$$R_{(i,j,k)} : \text{si } x \text{ est } A_i \text{ et } y \text{ est } B_j \text{ Alors } z = F(x) \quad (\text{I.18})$$

où f_k représente une fonction réelle quelconque.

Les fonctions de sortie f_k , peuvent être en principe des fonctions arbitraires des entrées, mais d'une manière générale elles sont choisies telles qu'elles soient une combinaison linéaire des entrées.

La particularité d'un modèle T-S est que la logique floue est seulement utilisée dans la partie prémisse des règles. La partie conclusion est décrite par des valeurs numériques.

Pour les valeurs d'entrée précises x_0, y_0 et z_0 , la sortie z est évaluée selon le mécanisme suivant :

$$z = \frac{\sum_{(i,j,k) \in I} w_{i,j} f_k(x_0, y_0)}{\sum_{(i,j,k) \in I} w_{i,j}} \quad \text{Avec } w_{i,j} = \mu_{A_i}(x_0), \mu_{B_j}(y_0) \quad (\text{I.19})$$

Ce type de modèle est aussi très intéressant pour la représentation de systèmes non linéaires tels que les systèmes mécaniques ou chaotiques.

Ces modèles sont actuellement les plus utilisés dans la conception des contrôleurs flous car ils présentent les avantages suivants :

- Ils permettent de décrire des situations où la structure physique du système est bien connue, par exemple, une fonction linéaire exprime la relation entrée-sortie.
- Ils permettent d'approcher le comportement d'un système complexe par une collection de modèles locaux, généralement linéaires.
- Les modèles de type T-S ont une capacité de représentation importante. En effet, le nombre de règles nécessaires pour approcher un système à un certain degré de précision est généralement plus faible que dans le cas des modèles linguistiques.
- Les systèmes flous de type de T-S permettent d'exploiter un certain nombre de techniques numériques issues de l'automatique conventionnelle (méthodes moindres carrés récursifs, filtrage de Kalman, commande adaptative, commande prédictive, ...etc.) et d'établir l'analyse de la stabilité et la robustesse.
- Il a été démontré dans la littérature que les systèmes flous de type T-S sont des bon approximateurs universels car ils sont capables d'approximer toute fonction non linéaire continue sur un ensemble compact avec une précision arbitraire définie.

Toutefois, l'inconvénient majeur de ces modèles réside dans la nature fonctionnelle des conclusions, c'est-à-dire, les règles n'ont pas de signification sémantique claire comme dans le cas des modèles linguistiques.

- Autre inconvénient réside dans que les contrôleurs de Takagi-Sugeno peuvent être considérés comme une classe particulière de contrôleurs non linéaires.

I.9. Approximation des fonctions par les systèmes flous

Deux raisons principales amènent à utiliser les systèmes flous comme élément de base des contrôleurs adaptatifs. Premièrement, ce type de système flous a la propriété d'approximation universelle de fonctions continues avec un degré de précision quelconque à condition

d'utiliser un problème suffisant de règles floues. Deuxièmement, les systèmes flous sont construits à partir de règles floues de la forme Si-alors, de ce fait, les informations linguistiques disponibles, issues d'une expertise, peuvent éventuellement être incorporées dans le contrôleur ce qui n'est pas le cas avec d'autres approximateurs universels tels que les réseaux de neurones et fonctions radiales...etc.

I.10. Conclusion

La logique floue est un domaine approprié pour modéliser les connaissances imprécises et incertaines, possédant la capacité d'approximation.

Un système flou est défini mathématiquement comme une relation non linéaire entre un espace d'entrée et un espace de sortie, il est constitué de quatre parties principales : la fuzzification, la base des règles, le système d'inférence et de défuzzification. La base des règles floues distingue essentiellement la nature des systèmes flous, symbolique, système flou de Mamdani, est algébrique pour celui de Takagi-sugeno.

Nous avons présenté dans ce chapitre des généralités sur la logique floue et des systèmes flous. L'architecture d'un système flou est déterminée par une meilleure compréhension des ensembles flous et opérateurs flous.

Dans le prochain chapitre, nous allons présenter le concept de la commande L1 adaptative et nous essaierons de mettre en évidence ses avantages par rapport à la commande adaptative classique notamment en termes de robustesse.



Chapitre II.

Concept de la commande L1 adaptative

Chapitre II.

Concept de la commande L1 adaptative

Chapitre II. Conception de la commande L1 adaptative

II.1. Introduction

Le début des recherches sur la commande adaptative date des années 1950. Le défi des automaticiens de faire remédier au handicap des régulateurs classiques a fait l'objet de l'apparition de plusieurs travaux de recherche, parmi ces travaux nous citons la commande adaptative. En 1958, Whittaker, Yamron et Kezer ont été les premiers à l'origine du schéma de base des systèmes de commande adaptative à modèle de référence (MRAC), l'architecture de cette commande a été développée d'une façon classique pour contrôler les systèmes linéaires en présence d'incertitudes paramétriques [1]. Au fil des ans, deux approches de techniques de contrôle adaptatif sont apparues à savoir : la méthode directe où seulement les paramètres de commande sont estimés et la méthode indirecte où les paramètres de processus sont estimés et ensuite utilisés dans le calcul de la loi de commande.

Dans un schéma de commande adaptative, pour améliorer les performances d'estimation il faut augmenter la vitesse d'adaptation, néanmoins une adaptation assez rapide peut diminuer significativement la robustesse du système de commande. La théorie de la commande L1 adaptative est adressée précisément à cette question en mettant une architecture en place pour que l'adaptation soit découplée de la robustesse. La vitesse d'adaptation dans ces architectures n'est limitée que par le matériel disponible, tandis que la robustesse est résolue par des méthodes de contrôle classiques et robustes. L'architecture de la commande L1 adaptative doit assurer des performances au régime transitoire et garantit la robustesse en présence d'une adaptation rapide. Sans introduction ou l'application de l'excitation persistante, sans aucune programmation de gain dans les paramètres du contrôleur, et sans avoir recours à la rétroaction à grand gain. Avec la commande L1 adaptative dans la boucle fermée sa commande peut être prédite à priori ce qui réduit considérablement la quantité d'analyse nécessaire pour la vérification et la validation de tels systèmes. Ces caractéristiques de la théorie de commande L1 adaptative ont été vérifiées avec la théorie dans un grand nombre d'essais en simulation [24].

Nous présentons dans ce chapitre deux techniques différentes de la commande adaptative, dans un premier lieu nous exposons la structure de la commande adaptative à modèle de référence sans et avec estimateur. Ensuite, nous allons présenter la technique de la commande L1 adaptative et son schéma général. À la fin de ce chapitre une étude comparative sur un système linéaire scalaire en boucle fermée est réalisée pour mettre en évidence les performances et l'efficacité de ces techniques.

II.2. Principe de la commande adaptative

La commande adaptative est un ensemble de techniques utilisées pour l'ajustement automatique en ligne des paramètres du régulateur afin de réaliser ou de maintenir un certain niveau de performances quand les paramètres du procédé à commander varient dans le temps. Les paramètres du régulateur sont adaptés de manière à poursuivre le système dans son évolution [26].

En principe, un système de commande adaptative mesure un certain indice de performance du système à commander à partir de l'écart entre l'indice de performance désiré et l'indice de performance mesuré. Le mécanisme d'adaptation commande certains paramètres du système ajustable ou introduit un signal supplémentaire de commande d'après une certaine stratégie afin de minimiser l'indice de performance, la figure (II.1) représente le principe général d'un système dans une plage donnée de commande adaptative [25].

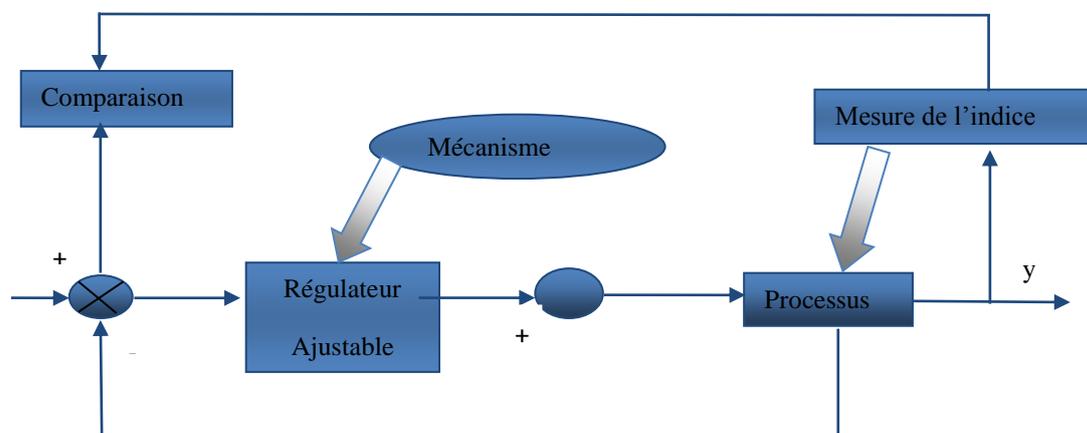


Figure II.1 : Principe des systèmes de commande adaptative.

En général, le système à contrôler possède des paramètres incertains au début de l'opération de commande. Bien que l'effet de cette incertitude paramétrique puisse disparaître en temps fini par un certain mécanisme d'ajustement, elle peut causer une instabilité du système. Dans d'autre cas, un phénomène inverse se produit, on démarre avec des paramètres connus et certains, mais au cours de fonctionnement, ces paramètres perdent leurs valeurs initiales et deviennent incertains, donc sans une mise à jour continue du régulateur, le régulateur initial s'avère inefficace [25].

Dans de nombreuses situations, les systèmes de commande classique sont basés sur les modèles à paramètres fixes, car ils ne peuvent prendre en charge les variations lentes ou rapides de la dynamique des systèmes, d'où la nécessité d'introduire la notion de systèmes de commande adaptative afin d'assurer les performances désirées. Pour cela, les paramètres du régulateur (contrôleur) sont ajustés automatiquement sur la base des informations recueillies du système.

La conception d'un contrôleur adaptatif implique généralement les étapes suivantes [1] :

- Choisir une loi de commande des paramètres variables ;
- Choisir une loi d'adaptation pour ajuster les paramètres ;
- Analyser les propriétés de convergence du système de contrôle résultant.

La commande adaptative se divise en deux grandes techniques à mettre en œuvre et largement utilisées à ce jour en pratique. Ces deux grandes techniques sont :

- Commande adaptative indirecte.
- Commande adaptative directe.

II.2.1 Commande adaptative indirecte

La commande indirecte MIAC, encore appelée la commande auto-ajustable, a été introduite en 1958 par Kalman. Cette stratégie se base sur les principes de séparation et d'équivalence.

Un régulateur adaptatif est conçu selon ce principe de séparation. Le schéma synoptique de la commande indirecte MIAC est représenté par la figure II.2. [27].

Le principe de cette méthode consiste à estimer les paramètres du modèle dynamique utilisé lors de la synthèse de la loi de commande. L'estimation du modèle du système est effectuée séparément de la conception du régulateur ; c'est-à-dire seul le critère sur la commande intervient pour la synthèse de ce dernier sans considération de performances de l'estimation. Autrement dit toute erreur d'identification nécessairement présente n'est pas prise en compte pour la phase de synthèse de la loi de commande (séparation totale des deux étapes), justifiant ainsi la terminologie de commande adaptative indirecte.

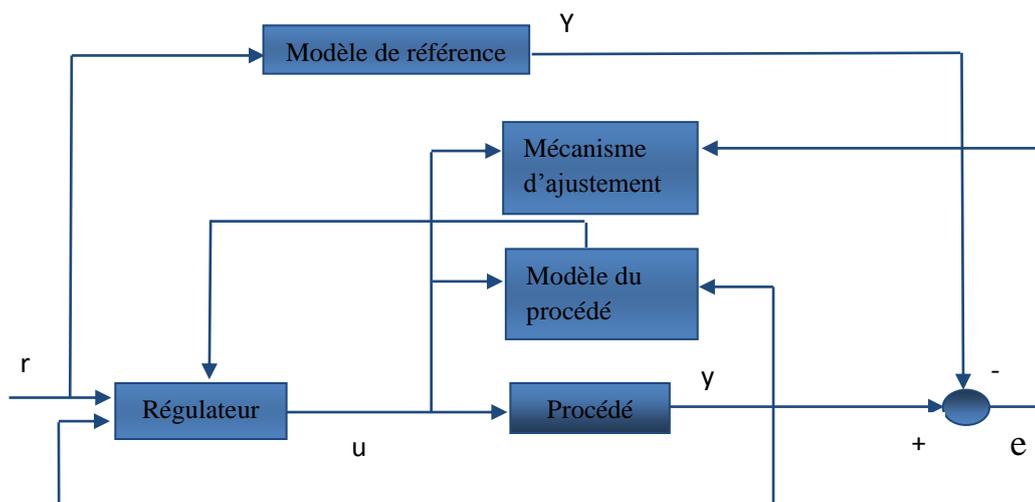


Figure II.2 : Architecture de la commande adaptative indirecte.

II.2.2 Commande adaptative directe

La commande adaptative directe dit aussi (MRAC) qui sera connue plus tard sous la règle de conception du MIT, a été proposée pour la première fois en 1961. Elle spécifie la forme désirée de la réponse du processus à un signal de commande par l'intermédiaire d'un modèle de référence, image donc des performances souhaitées en boucle fermée. Cette stratégie se base sur la minimisation d'un indice de performances. Un mécanisme d'adaptation élaboré à

partir de la sortie du système et celle du modèle et un jeu de paramètres est conçu de telle sorte que la différence entre ces deux sorties tend vers zéro [27]. La structure de la loi de commande est telle que le comportement du système bouclé soit stable et équivalent au modèle de référence, malgré la présence des dynamiques non modélisées et d'incertitudes paramétriques [28].

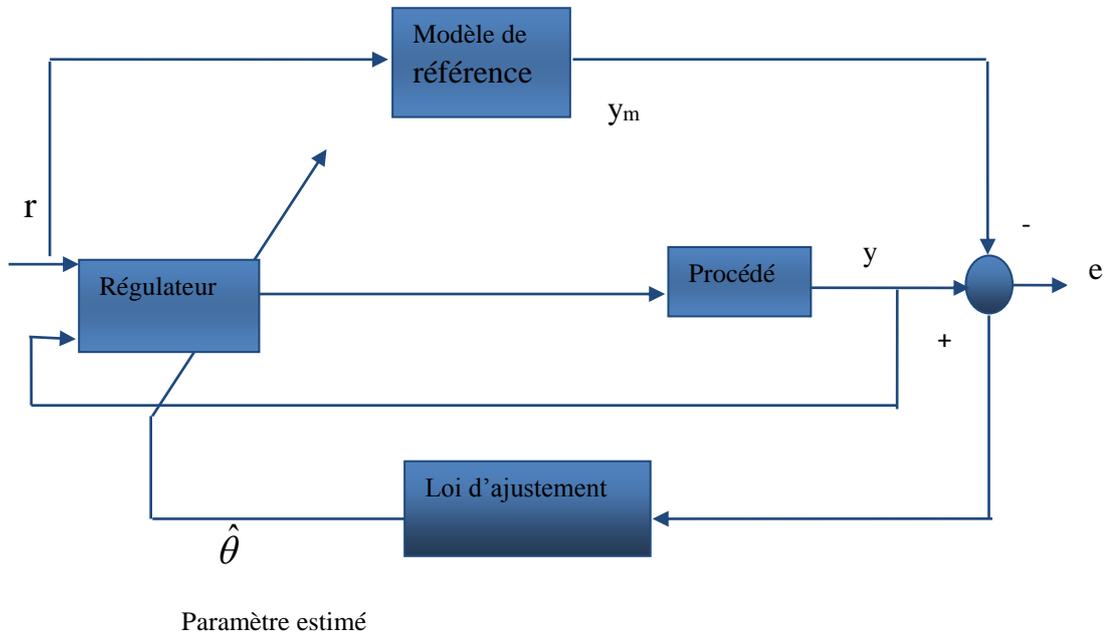


Figure II.3 : Architecture de la commande adaptative directe.

On considère la dynamique du système gouvernée par l'équation différentielle suivante [24]

$$\begin{aligned} \dot{x}(t) &= A_m x(t) + b(u(t) + k_x^T x(t)), x(0) = x_0 \\ y(t) &= c^T x(t) \end{aligned} \quad (\text{II.1})$$

Où $x(t) \in \mathbb{R}^n$ est l'état du système mesuré, $A_m \in \mathbb{R}^{n \times n}$ est une matrice de Hurwitz connue qui définit la dynamique désirée du système en boucle fermée, $(b, c \in \mathbb{R}^n)$ sont des vecteurs constants connus, $k_x \in \mathbb{R}^n$ est un vecteur de paramètres constants inconnus, $u(t) \in \mathbb{R}$ est la commande, et $y(t) \in \mathbb{R}$ est la sortie à commander. L'objectif est de concevoir une loi de commande en boucle fermée pour l'entrée $u(t)$ tel que $y(t)$ suit une trajectoire $r(t)$ avec des spécifications désirées, et tous les signaux en boucle fermée soient bornés.

Pour arriver à l'architecture de MRAC, nous considérons le contrôleur nominal suivant :

$$u_{nom}(t) = -k_x^T x(t) + k_g r(t) \quad (\text{II.2})$$

où

$$k_g \triangleq 1/c^T A_m^{-1} b \quad (\text{II.3})$$

Ce contrôleur nominal assure une annulation parfaite de l'incertitude dans (II.1) et mène au système de référence désiré (idéal)

$$\begin{aligned} \dot{x}_m &= A_m x_m(t) + b k_g r(t), \\ y_m &= c^T x_m(t) \end{aligned} \quad x_m(0) = x_0 \quad (\text{II.4})$$

où, $x_m(t) \in R^n$ est l'état du système de référence, le choix de k_g comme dans (II.3) assure que y_m suit une entrée de référence constante avec une erreur statique nulle.

Le contrôleur adaptatif direct à modèle de référence est donné par :

$$u(t) = -\hat{k}_x^T(t)x(t) + k_g r(t) \quad (\text{II.5})$$

où $\hat{k}_x(t) \in R^n$ est l'estimation de k_x . En remplaçant (II.5) dans (II.1), on trouve la dynamique du système en boucle fermée ci-après :

$$\begin{aligned} \dot{x}(t) &= (A_m - b\tilde{k}_x^T(t))x(t) + b k_g r(t) \\ y(t) &= c^T x(t) \end{aligned} \quad x(0) = x_0$$

avec $\tilde{k}_x(t) \triangleq \hat{k}_x(t) - k_x$, est l'estimateur paramétrique.

soit $e(t) \triangleq x_m(t) - x(t)$ l'erreur de poursuite, l'erreur de poursuite dynamique peut s'écrire alors

$$\dot{e}(t) = A_m e(t) + b\tilde{k}_x^T(t)x(t) \quad e(0) = 0 \quad (\text{II.6})$$

La nouvelle loi des paramètres estimés est donné par :

$$\dot{\hat{k}}_x(t) = -\Gamma x(t)e^T(t) P b \quad \hat{k}_x = k_{x0} \quad (\text{II.7})$$

où $\Gamma \in R^+$ est le gain d'adaptation et $p = p^T > 0$ est une matrice solution de l'équation $A_m^T P + P A_m = -Q$.

Pour chaque matrice constante $Q = Q^T > 0$, le schéma bloc du système en boucle fermée est donné par la figure (III.4)

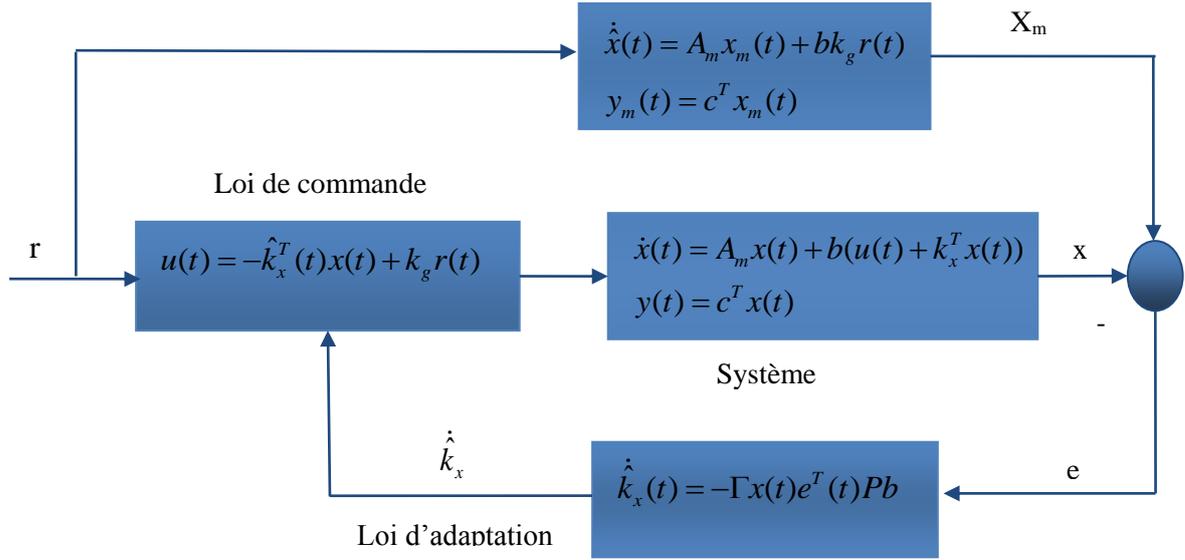


Figure II.4 : Structure de la MRAC en boucle fermée sans estimateur d'état.

II.2.3 Méthode de lyapunov

Le concept de la commande adaptative est basé sur la théorie de Lyapunov. Le but est de construire de proche en proche une loi de commande ramenant le système vers des états désirés. En d'autres termes, on souhaite faire de l'état désiré un état d'équilibre stable en boucle fermée [29].

La méthode de Lyapunov permet d'étudier la stabilité des systèmes complexes qui sont décrits par des systèmes différentiels. La stabilité d'un système est la capacité de ce dernier à revenir à sa position d'équilibre lors qu'il en est ponctuellement écarté. Il existe deux méthodes pour démontrer la stabilité du système, directe et indirecte [30].

Considérons la fonction de Lyapunov suivante :

$$V(e(t), \tilde{k}_x(t)) = e^T(t) P e(t) + \frac{1}{\Gamma} \tilde{k}_x^T(t) \tilde{k}_x(t) \quad (\text{II.8})$$

Sa dérivée temporelle au long des trajectoires du système (II.6) et (II.7) est donnée par :

$$\begin{aligned} \dot{V}(t) &= -e^T(t) Q e(t) + 2e^T(t) P b \tilde{k}_x^T(t) x(t) + \frac{2}{\Gamma} \tilde{k}_x^T(t) \dot{\tilde{k}}_x(t) \\ &= -e^T(t) Q e(t) + 2\tilde{k}_x^T(t) \left(\frac{1}{\Gamma} \dot{\tilde{k}}_x(t) + x(t) e^T(t) P b \right) \\ &= -e^T(t) Q e(t) \leq 0 \end{aligned}$$

d'où le point d'équilibre de (II.6)-(II.7) est stable au sens de Lyapunov, c'est-à-dire les signaux $e(t)$ et $\tilde{k}_x(t)$ sont bornés. Comme $x(t) = x_m(t) - e(t)$ et $x_m(t)$ est l'état d'un modèle de référence stable, alors $x(t)$ est borné. Pour montrer que le comportement de l'erreur de

converge asymptotiquement vers zéro, nous calculons la deuxième dérivée de $V(e(t), \tilde{k}_x(t))$ comme suit :

$$\ddot{V}(t) = -2e^T(t)Q\dot{e}(t)$$

De (II.6) nous déduisons que $e(t)$ est uniformément borné, et d'où $\ddot{V}(t)$ est borné, en utilisant le fait que $\dot{V}(t)$ est uniformément continu. L'application du lemme de Barbalat conduit à

$$\lim_{t \rightarrow \infty} \dot{V}(t) = 0$$

Par conséquent, $e(t) \rightarrow 0$ lorsque $t \rightarrow \infty$. Donc $x(t)$ converge asymptotiquement vers $x_m(t)$, ce qui implique que $y(t) = c^T x_m(t)$, il suit donc $r(t)$ avec les spécifications désirées.

Notons que la convergence asymptotique des erreurs d'estimation vers zéro n'est pas garantie. Seulement la bornitude des erreurs d'estimation est garantie.

II.2.4. Commande adaptative directe avec estimateur d'état

Dans ce qui suit, on considère une reparamétrisation de l'architecture précédente qui utilise un estimateur de l'état (ou identificateur) [24], donné par :

$$\begin{aligned} \dot{\hat{x}}_m(t) &= A_m \hat{x}(t) + b(u(t) + \hat{k}_x^T(t)x(t)), & \hat{x}(0) &= x_0 \\ \hat{y}(t) &= c^T \hat{x}(t), \end{aligned} \quad (\text{II.9})$$

avec $\hat{x}(t) \in \mathbb{R}^n$ est l'état estimé. Le système dans (II.9) a la même structure du système (II.1) avec k_x un paramètre inconnu remplacé par son estimation $\hat{k}_x(t)$. Par la soustraction de (II.1) de (II.9), on obtient la dynamique de l'erreur d'estimation (ou l'identification de la dynamique de l'erreur), indépendamment du choix de la commande.

$$\dot{\tilde{x}}(t) = A_m \tilde{x}(t) + b k_x^T(t)x(t), \quad \tilde{x}(0) = 0$$

où $\tilde{x}(t) \triangleq \hat{x}(t) - x(t)$ et $\tilde{k}_x(t) \triangleq \hat{k}_x(t) - k_x$. Notons que cette dynamique de l'erreur est identique à la dynamique de l'erreur dans (II.6).

Ensuite, considérons une loi d'adaptation pour $\hat{k}_x(t)$ donner par :

$$\dot{\hat{k}}(t) = -\Gamma x(t)\tilde{x}(t)Pb, \quad \hat{k}_x(0) = k_{x0} \quad (\text{II.10})$$

où $\Gamma \in \mathbb{R}^+$ est le gain d'adaptation et $A_m^T P + P A_m = -Q$, $Q = Q^T > 0$. Cette loi d'adaptation est similaire à (II.7) dans sa structure, mais l'erreur de la poursuite $e(t)$ est remplacée par son paramètre estimé $\tilde{x}(t)$. Le choix de la fonction de lyapunov candidate est le suivant

$$V(\tilde{x}(t), \tilde{k}_x(t)) = \tilde{x}^T(t)P\tilde{x}(t) + \frac{1}{\Gamma} \tilde{k}_x^T(t)\tilde{k}_x(t)$$

ce qui donne

$$\dot{V}(t) = -\tilde{x}^T(t)Q\tilde{x} \leq 0$$

donc les erreurs $\tilde{x}(t)$ et $\tilde{k}_x(t)$ sont uniformément bornées. Cependant, sans introduire le signal de commande $u(t)$ on ne peut pas appliquer le lemme de Barbalat pour conclure la convergence asymptotique de $\tilde{x}(t)$ vers zéro. Aussi, $\tilde{x}(t)$ et $\hat{x}(t)$ peuvent diverger en même temps, $\tilde{x}(t)$ reste uniformément borné.

Si on utilise (II.5) dans (II.9), prenons en considération la formule (II.10), l'estimateur d'état en boucle fermée reproduit le système de référence borné de (II.4)

$$\begin{aligned} \dot{\hat{x}}(t) &= A_m \hat{x}(t) + b k_g r(t), \quad \hat{x}(0) = x_0 \\ \hat{y}(t) &= c^T \hat{x}(t) \end{aligned}$$

d'où, le lemme de Barbalat peut être utilisé pour conclure que $\tilde{x}(t) \rightarrow 0$ quand $t \rightarrow \infty$.

Le schéma bloc du système en boucle fermée avec l'estimateur est donné dans la figure (II.5). Les figures (II.4) et (II.5) illustrent la différence fondamentale entre la commande MRAC directe et celle basée sur l'utilisation d'un estimateur d'état. Dans la figure (II.5), le signal de commande est fourni comme étant l'entrée pour les deux systèmes, le système et l'estimateur, par contre dans la figure (II.4) le signal de commande sert seulement comme entrée du système.

Le diagramme du bloc du système en boucle fermée avec l'estimateur d'état est donné dans la figure (II.5)

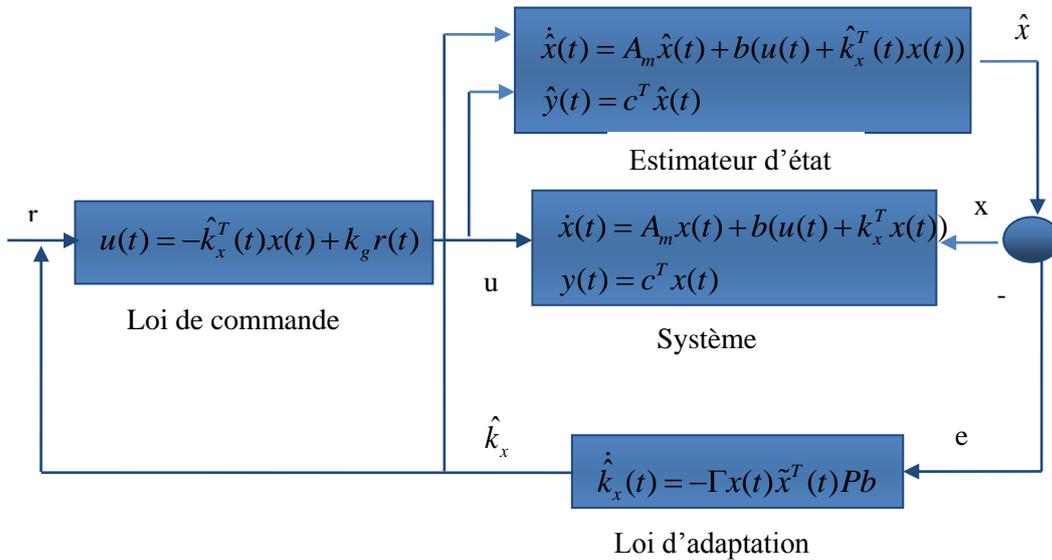


Figure II.5 : Structure de la MRAC en boucle fermée avec estimateur d'état.

II.3. Exemple d'application

Dans cette partie, nous présentons l'idée essentielle du contrôleur **L1 adaptatif**, qui permet une adaptation rapide et garanti la robustesse. On commence par un système simple, scalaire et stable avec une perturbation constante, faisant appel aux outils de commande classique pour l'analyser. On constate que dans ce cas, le MRAC sera réduit à un contrôleur linéaire intégrateur [24].

On considère le système dynamique linéaire du premier ordre donné par l'équation différentielle suivante :

$$\dot{\hat{x}}(t) = -5\hat{x}(t) + \hat{\theta}(t) + u(t), \quad x(0) = x_0 \tag{II.11}$$

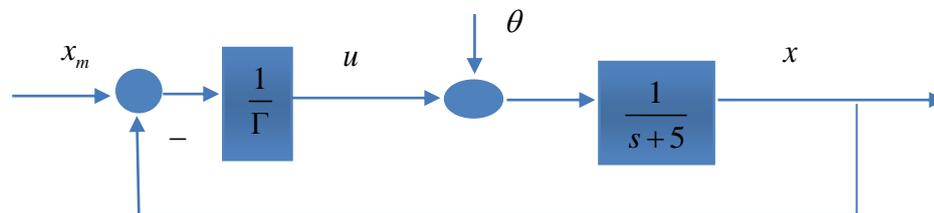


Figure II.6 : Système en BF avec MRAC contrôleur de type intégrateur.

où θ représente le paramètre constant inconnu rejeté par la commande d'entrée $u(t)$. Soit l'objectif est la stabilisation de l'origine. Pour ce système, l'architecture de la MRAC décrit dans (II.4) et (II.5) est réduite à un contrôleur intégrateur de la structure

$$u(t) = -\hat{\theta}(t) \tag{II.12}$$

où, $\hat{\theta}$ est le paramètre estimé de θ , donné par

$$\dot{\hat{\theta}} = -\Gamma(x_m(t) - x(t)), \quad \hat{\theta}(0) = \theta_0, \quad \Gamma > 0 \quad (\text{II.13})$$

et $x_m(t)$ est le signal de référence, généré par le système

$$\dot{x}_m = -x_m(t), \quad x_m(0) = x_0$$

On constate que ce système de référence est obtenu à partir du système d'origine (II.11) par substitution du contrôleur idéal nominal $u_{nom}(t) = -\theta$ dedans, assumant ainsi l'annulation parfaite du paramètre incertain θ dans le système (II.11). Le schéma du système en boucle fermée est représenté sur la figure (II.6).

Pour l'application de cet exemple, on considère les deux cas suivants :

- L'application de la commande adaptative à modèle de référence sans estimateur.
- L'application de la commande adaptative à modèle de référence avec estimateur.

II.3.1. Application de la commande adaptative sans estimateur d'état

Pour ce système, l'architecture de la MRAC décrit dans (II.2) et (II.5) est réduite à un contrôleur intégrateur de la structure dans (II.11)

La fonction de transfert de ce système est la suivante

$$L_1(s) = \frac{\Gamma}{s(s+5)} \quad (\text{II.14})$$

Parce que le système en boucle fermée reste linéaire et invariable dans le temps, on peut utiliser des outils standards du modèle classique pour analyser les marges de stabilité du système. Généralement, les deux marges de stabilité les plus utilisées sont le gain et la marge de phase. D'après la figure (II.9) (a), il est évident que le schéma de Nyquist de $L_1(s)$ ne franchit jamais la partie négative de la ligne réelle ; par conséquent le système en boucle fermée à la marge de gain infini ($g_m = \infty$). La fréquence de gain croisement peut être calculé à partir de

$$|L_1(jw_{gc})| = \frac{\Gamma}{w_{gc} \sqrt{w_{gc}^2 + 5}} = 1$$

Ce qui mène à la marge de phase

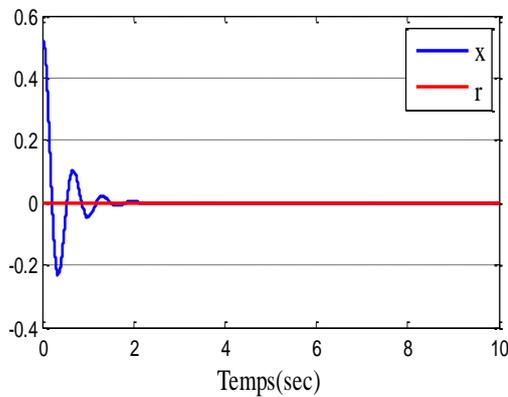
$$\phi_m = \pi + \angle L_1(jw_{gc}) = \arctan\left(\frac{1}{w_{gc}}\right)$$

Dans les simulations nous avons pris

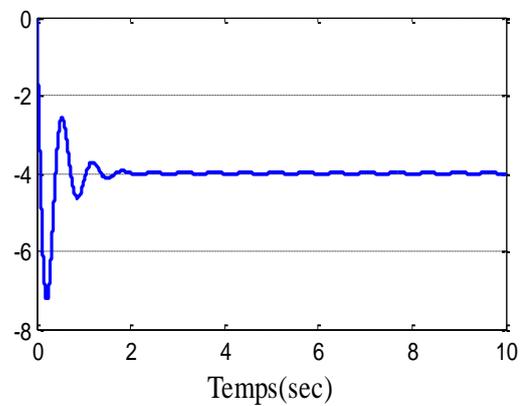
- Les états initiaux choisies : $x(0) = 0.5$ et $\theta(0) = 0$;
- Les paramètres de synthèse sont choisis comme suit $\Gamma = 100$, $k_g = 5$, $A_m = 5$ et $r = 0$;
- La durée de simulation du système de commande est 10s.

II.3.2. Résultats et simulation

Les résultats de simulation sont donnés respectivement, dans les figures (a), (b), (c) :



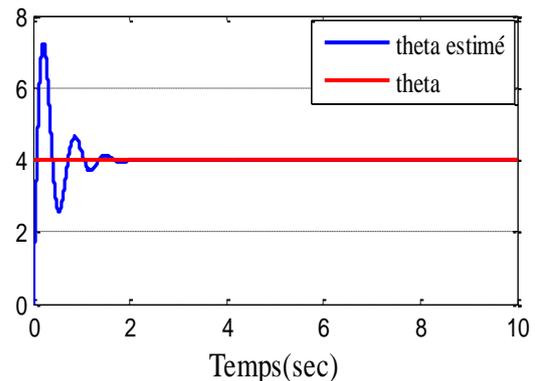
(a) la consigne (r) et la sortie du système



(b) signal de commande u(t)

D'après les résultats obtenus on voit que :

- La sortie suit celle de la consigne et tous les signaux en boucle fermée sont bornés.
- On observe que le paramètre estimé $\hat{\theta}$ tend vers θ .
- La grandeur de commande est bornée.



(c) paramètre estimé θ

Figure II.7: Résultats de simulation en boucle fermée sans estimateur d'état.

II.3.3 Application de la commande adaptative à modèle de référence avec estimateur d'état

Pour ce système, l'architecture de la MRAC décrit dans (II.11) et (II.12) est réduite à un contrôleur intégrateur de la structure

$$\dot{\hat{x}}(t) = -5\hat{x}(t) + \hat{\theta}(t) + u(t), \quad \hat{x}(0) = x_0 \quad (\text{II.15})$$

Qui mène à la dynamique d'erreur de prédiction suivante, indépendamment du choix de la commande

$$\dot{\tilde{x}}(t) = -5\tilde{x}(t) + \tilde{\theta}(t), \quad \tilde{x}(0) = x_0 \quad (\text{II.16})$$

où $\tilde{x}(t) \triangleq \hat{x}(t) - x(t)$ et $\tilde{\theta}(t) \triangleq \hat{\theta}(t) - \theta$. Les paramètres sont estimés comme suit

$$\dot{\hat{\theta}}(t) = -\Gamma \tilde{x}(t), \theta(0) = \theta_0, \Gamma > 0 \quad (\text{II.17})$$

Ensuite, au lieu de choisir le contrôleur adaptatif comme $u(t) = -\hat{\theta}(t)$. Nous utilisons un filtre passe bas tel que

$$u(s) = -C(s)\hat{\theta}(s) \quad (\text{II.18})$$

avec

$$C(s) = \frac{w_c}{w_c + s} \quad (\text{II.19})$$

Cependant, des résultats semblables peuvent être obtenus en utilisant des filtres plus complexes. La fonction de transfert de la boucle de ce système est

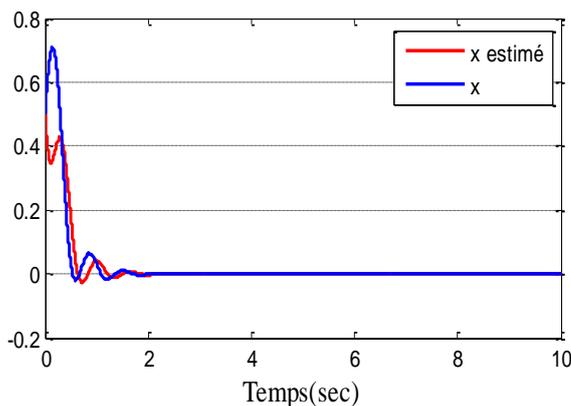
$$L_2(s) = \frac{\Gamma C(s)}{s(s+5) + \Gamma(1-C(s))} \quad (\text{II.20})$$

Dans les simulations, nous avons pris

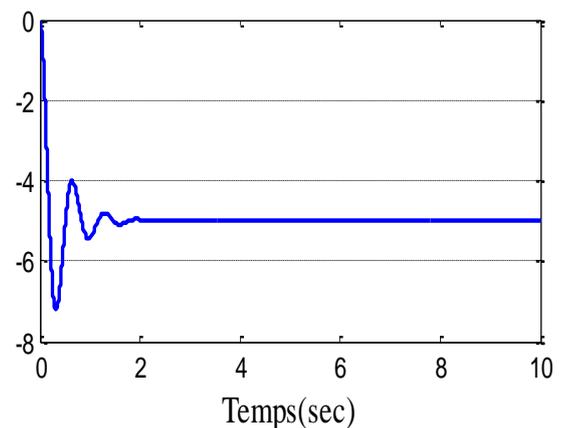
- Les états initiaux choisis : $x(0) = 0.5$ et $\theta(0) = 0$;
- Les paramètres de synthèse sont choisis comme suit : $\Gamma = 100, k_g = 5, A_m = 5$ et $r = 0$.
- La durée de simulation du système de commande est de 10s.

Résultats et simulation

Les résultats de simulation sont donnés respectivement, dans les figures (a), (b), (c) :



(a) l'état estimé et l'état réel



(b) signal de commande $u(t)$

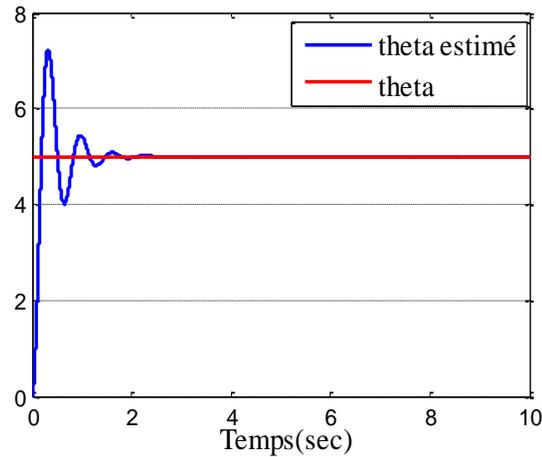
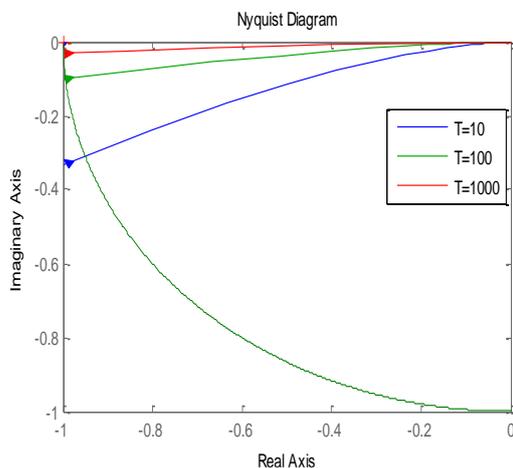
(c) Paramètre estimé θ

Figure II.8 : résultats de simulation en boucle fermée avec estimateur d'état.

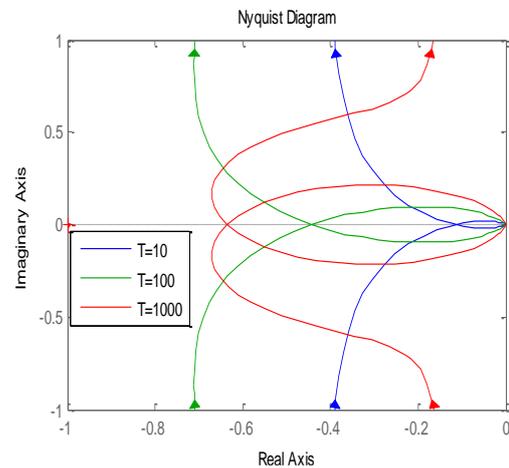
Interprétation des résultats

Les résultats obtenus montrent le comportement du système en boucle fermée. On remarque bien que la sortie réelle suit bien la sortie désirée et nous avons une bonne estimation du paramètre θ , tout cela signifie que la commande assure la stabilité du système.

Le diagramme de Nyquist est donné par la figure (II.9) (b). Il apparaît que les marges de phase et les marges de gain de contrôleur L1 ne sont pas significativement affectés par les grandes valeurs de Γ contrairement au contrôleur intégrateur.



(a) contrôleur intégrateur



(b) contrôleur L1

Figure II.9 : Diagramme de Nyquist des fonctions de transfert.

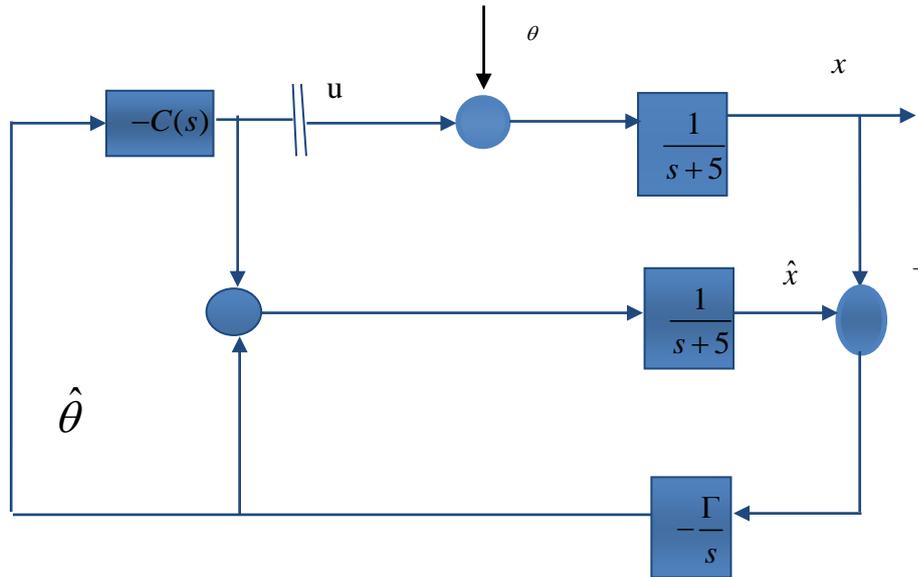


Figure II.10 : Système en boucle BF avec le contrôleur L1 adaptatif.

De plus, si nous supposons que $\Gamma \rightarrow \infty$, nous remarquons que l'expression dans (II.20) mène à la fonction de transfert en boucle fermée suivante

$$L_{2l}(s) = \frac{C(s)}{1 - C(s)} = \frac{w_c}{s} \tag{II.21}$$

Cette fonction de transfert en boucle fermée donne une marge de gain infinie $g_m = \infty$ et une marge de phase $\phi_m = \pi / 2$. Cependant, d'après la figure (II.11) (a), nous remarquons que la marge de gain est toujours finie et converge à $g_m = 6.02dB$ avec l'augmentation de Γ .

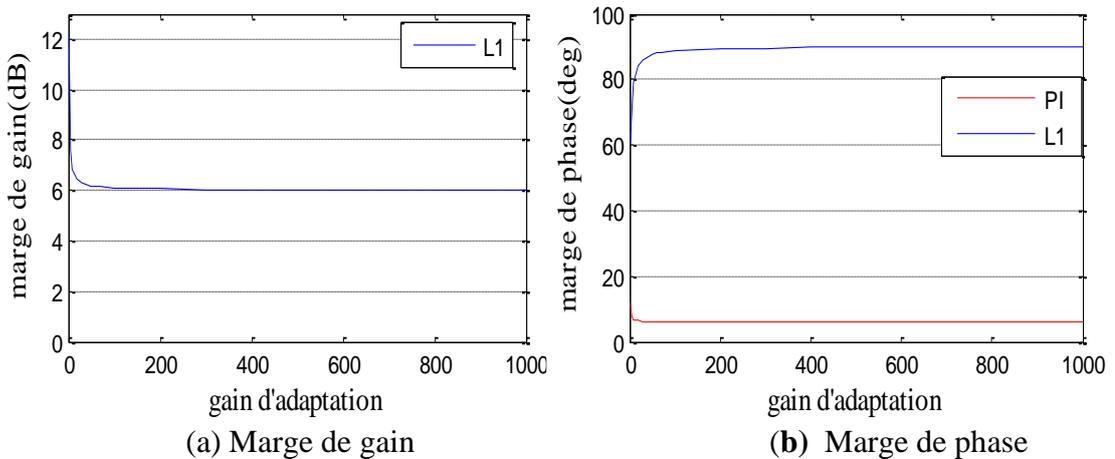


Figure II.11 : L'effet du gain d'adaptation sur la marge de la stabilité.

Dans la partie suivante consacrée à la commande L1 adaptative, nous allons utiliser la loi de commande L1 pour améliorer les performances.

II.4. Concept de la commande L1 adaptative

Dans cette partie, nous présentons une architecture différente appliquée sur des systèmes LTI en présence de paramètres constants inconnus, mais presque équivalente à celle de la partie précédente. Bien que la mise en œuvre est différente, elle mène à la même dynamique d'erreurs avec des mêmes conditions initiales [24].

Le système linéaire à paramètres variables peut être décrit par les équations suivantes :

$$\begin{aligned} \dot{x}(t) &= Ax(t) + b(u(t) + \theta^T x(t)), & x(0) &= x_0 \\ y(t) &= c^T x(t) \end{aligned} \quad (\text{II.22})$$

avec $x(t) \in \mathbb{R}^n$ est le vecteur d'état du système, $A \in \mathbb{R}^{n \times n}$ est la matrice à paramètres constants et connus présente la dynamique du système avec (A, b) contrôlable, $b, c \in \mathbb{R}^n$ sont des matrices à paramètres constants, $u(t) \in \mathbb{R}$ est l'entrée de commande, $y(t) \in \mathbb{R}$ est la sortie à commander, θ est le paramètre inconnu qui appartient à l'ensemble compact donné par $\theta \in \Theta \subset \mathbb{R}^n$.

Notre objectif consiste à synthétiser une loi de commande qui assure la bornitude de toutes les variables du système en boucle fermée et garantit la poursuite pour la sortie $y(t)$ d'une référence $r(t)$.

II.4.1. Architecture de la commande L1

On considère la structure de commande suivante :

$$u(t) = u_m(t) + u_{ad}(t), \quad u_m(t) = -k_m^T x(t) \quad (\text{II.23})$$

où $k_m \in \mathbb{R}^n$ est choisi tel que $A_m \triangleq A - bk_m^T$ soit une matrice de Hurwitz, $u_{ad}(t)$ est une composante adaptative qui sera définie dans ce qui suit. Le gain statique k_m donne le système suivant en boucle fermé

$$\begin{aligned} \dot{x}(t) &= A_m x(t) + b(\theta^T x(t) + u_{ad}(t)), & x(0) &= x_0 \\ y(t) &= c^T x(t) \end{aligned} \quad (\text{II.24})$$

Pour le système linéairement paramétré dans (II.23), on considère l'estimateur d'état suivant

$$\begin{aligned} \dot{\hat{x}}(t) &= A_m \hat{x}(t) + b(\theta^T \hat{x}(t) + u_{ad}(t)), & \hat{x}(0) &= x_0 \\ \hat{y}(t) &= c^T \hat{x}(t) \end{aligned} \quad (\text{II.25})$$

avec $\hat{x}(t) \in \mathbb{R}^n$ est l'état estimé et $\hat{\theta}(t) \in \mathbb{R}^n$ est l'estimation du paramètre θ , généré par la loi adaptative de type projection suivante

$$\dot{\hat{\theta}}(t) = \Gamma \text{Pr oj}(\hat{\theta}(t), -\tilde{x}(t)Pbx(t)), \quad \hat{\theta}(0) = \hat{\theta}_0 \in \Theta \quad (\text{II.26})$$

où $\tilde{x}(t) \triangleq \hat{x}(t) - x(t)$ est l'erreur d'estimation, $\Gamma \in \mathbb{R}^+$ est le gain d'adaptation, et $P = P^T > 0$ est une matrice solution de $A_m^T P + PA_m = -Q$, pour chaque matrice constante $Q = Q^T > 0$, la projection est restreinte à l'ensemble Θ . La transformé de Laplace du signal de commande est donnée par

$$u_{ad}(s) = -C(s)(\hat{\eta}(s) - k_g r(s)) \quad (\text{II.27})$$

avec $r(s)$ et $\hat{\eta}(s)$ est la transformé de Laplace de $r(t)$ et $\hat{\eta}(s) \triangleq \hat{\theta}^T(t)x(t)$ respectivement, $k_g \triangleq -1/(c^T A_m^{-1}b)$, $C(s)$ est une fonction de transfert BIBO-stable et strictement adéquat, avec $C(0)$ un gain, et la réalisation de son espace d'état suppose des conditions initiales nulles, avec $c(s) = \frac{w_c}{s + w_c}$.

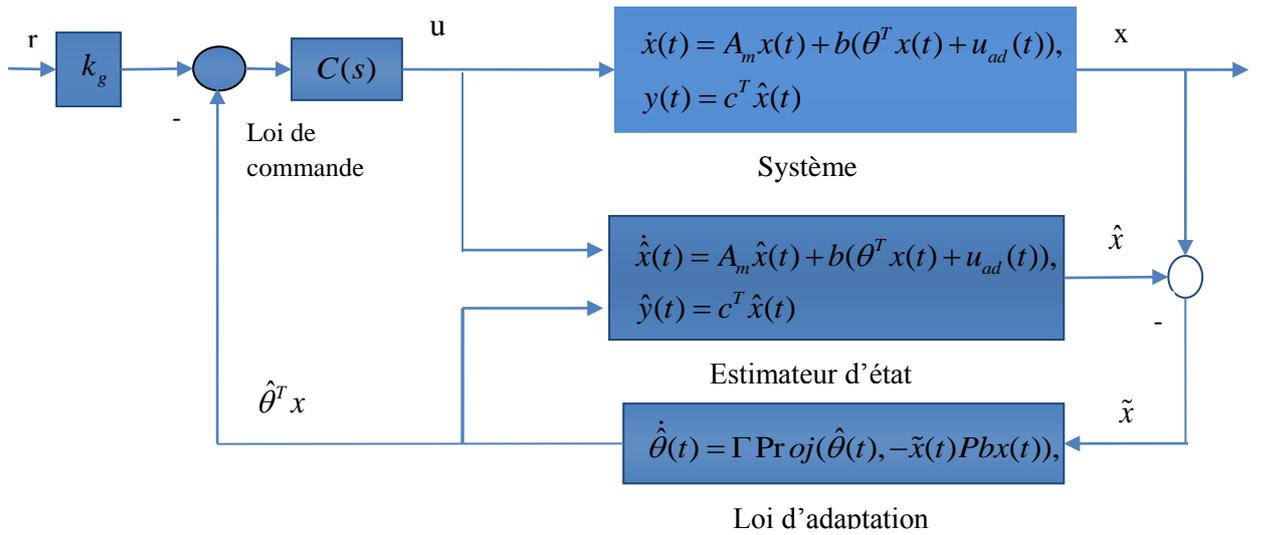


Figure II.12 : Structure de la commande adaptative en BF.

La commande L1 est définie via les équations (II.22), (II.24) et (II.26), avec k_m et $C(s)$ vérifiant de la norme L1

$$\lambda \triangleq \|G(s)\|_1 < 1 \quad (\text{II.28})$$

Tel que

$$G(s) = H(s)(1 - C(s)), H(s) \triangleq (sI - A_m)^{-1}b, \quad L \triangleq \max \|\theta\|_1 \quad (\text{II.29})$$

L'architecture de la commande L1 adaptative et ses éléments principaux sont respectivement dans la figure (II.12).

II.5. Exemple de simulation

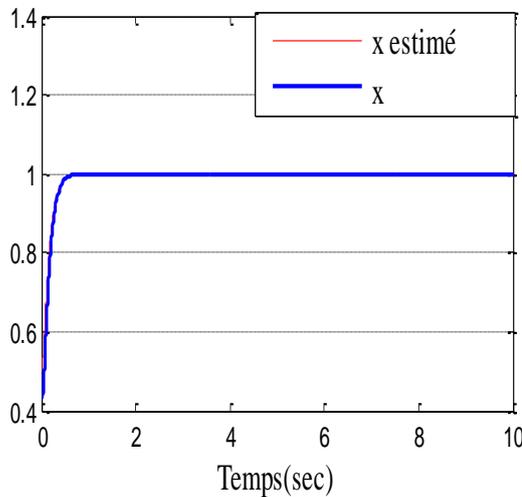
On considère le système dynamique linéaire du premier ordre donné par l'équation différentielle (déjà utilisé dans l'exemple de simulation dans la partie précédente sur lequel on a appliqué la MRAC), pour améliorer les performances du système

$$\dot{x}(t) = -5x(t) + \theta + u(t), \quad x(0) = 0.5$$

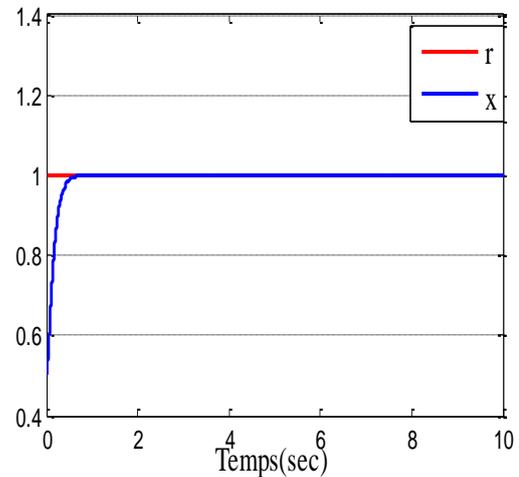
où $\theta = 5$, représente un paramètre constant inconnu. Les conditions initiales sont choisies par $x(0) = 0.5$ et $\theta(0) = 0$.

Les paramètres de synthèse sont choisis comme suit : $\Gamma = 600000$, $k_g = 5$, $A_m = 5$ et $r = 1$

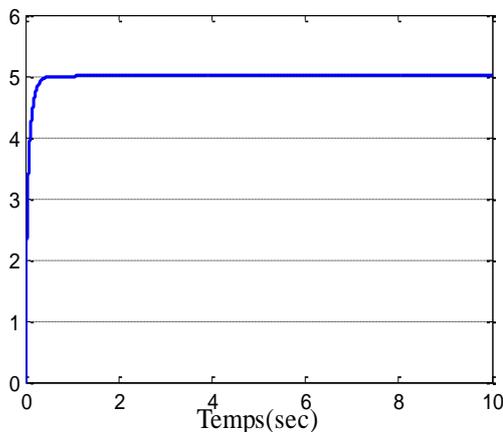
Les résultats de simulation sont donnés respectivement, dans les figures suivantes,



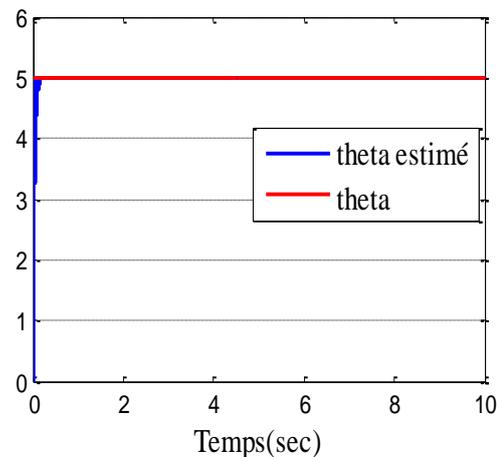
(a) estimateur d'état et la sortie du système (x)



(b) la consigne(r) et le signal de sortie(x)



(c) signal de commande u(t)



(d) paramètre estimé θ

Figure II.13 : Résultats de simulation.

Interprétation des résultats :

D'après ces résultats nous constatons que le signal de commande est lisse, on remarque une bonne estimation des paramètres, qui se fait néanmoins avec des oscillations à haute fréquences. Ces résultats montrent aussi une bonne estimation des états et les sorties réelles convergent vers les sorties désirées.

II.6. Conclusion

Nous avons présenté dans ce chapitre le principe de la commande adaptative avec ses deux architectures directe et indirecte, particulièrement la commande adaptative à modèle de référence sans et avec estimateur d'état. Par la suite, nous avons aussi fait une simulation de ces deux types de commande sur un système scalaire du premier ordre.

Au cours de ce chapitre nous avons présenté les concepts et le principe de la commande L1 adaptative, avec une application sur le même système qu'on a utilisé pour la MRAC avec ces deux approches sans et avec prédicteur. La stabilité du système est prouvée par la méthode de Lyapunov. Les résultats de simulation montrent l'efficacité de la technique commande L1 adaptative pour l'amélioration des performances du système.

Dans le prochain chapitre, nous allons introduire une approche de commande L1 adaptative floue pour une classe des systèmes MIMO avec une application sur un robot manipulateur.



Chapitre .III

Commande L1 adaptative floue appliquée à un robot manipulateur

Chapitre .III

Commande L1 adaptative floue appliquée à un robot manipulateur

Chapitre III. Commande L1 adaptative floue appliquée à un robot manipulateur

III.1. Introduction

Dans ce chapitre, nous présentons une architecture de commande L₁ adaptative floue pour une classe des systèmes non linéaires.

Ce chapitre est constitué de deux parties. D'abord, nous présentons la structure de la commande L1 adaptative floue pour la classe des systèmes considérée avec l'étude de ses performances. La deuxième partie sert à l'application de cette structure de commande L1 adaptative pour commander un bras manipulateur, une étude en simulation des performances de cette structure de commande est réalisée sous Matlab.

III.2. Conception d'une commande L1 adaptative floue

Dans cette partie, nous présentons la conception d'un contrôleur L1 adaptatif flou pour une classe des systèmes non linéaires MIMO (Multi Input Multi Output) affines en la commande.

III.2.1. Formulation du problème

Dans cette étude, nous considérons les systèmes dynamiques non linéaires MIMO (Multi Input Multi Output) Σ composé de sous-systèmes $p = 2, \dots, n$, $i = 1, 2, \dots, n$, représenté sous la forme normale suivante :

$$\Sigma_i \begin{cases} \dot{x}_{i1} = x_{i2}, \dot{x}_{i2} = x_{i3}, \dots, \dot{x}_{i(n_i-1)} = x_{in_i} \\ \dot{x}_{in_i} = f_i(x) + \sum_{j=1}^p g_{ij}(x)u_j \\ y = x_{i1} \end{cases} \quad (\text{III.1})$$

Où $x = [x_1, \dots, x_p]^T \in R^p$ est le vecteur d'état supposé mesurable, avec $u = [u_1, \dots, u_p]^T \in R^p$ est l'entrée de la commande, $y = [y_1, \dots, y_p]^T \in R^p$ est le vecteur de sortie. $f_{ij}(x), g_{ij}(x)$ (avec $i, j = 1, \dots, p$) sont des fonctions non linéaires supposées inconnues.

Nous choisissons la notation suivante

$$y^{(r)} = \begin{bmatrix} y_1^{(r_1)} \\ \vdots \\ y_p^{(r_p)} \end{bmatrix}, f(x) = \begin{bmatrix} f_1(x) \\ \vdots \\ f_p(x) \end{bmatrix}, G(x) = \begin{bmatrix} g_{11}(x) & \dots & g_{1p}(x) \\ \vdots & \ddots & \vdots \\ g_{p1}(x) & \dots & g_{pp}(x) \end{bmatrix} \quad (\text{III.2})$$

La dynamique du système global est donnée par

$$y^{(r)} = f(x) + G(x)u \quad (\text{III.3})$$

Notre objectif consiste à synthétiser une loi de commande qui assure la bornitude de tous les variables dans la boucle fermé et garantit la poursuite de la trajectoire désirée

$$y_d(t) = [y_{d1}(t), \dots, y_{dp}(t)]^T.$$

Avant de passer à la conception du contrôleur L1 adaptatif flou, nous considérons l'hypothèse suivante concernant le système (III.1) et la trajectoire désirée $y_{di}(t)$.

Hypothèse III.1 : La trajectoire désirée $y_{di}(t)$ est une fonction bornée, sa dérivée est connue et bornée, et $y_{di}(t)$ supposé être différentielle r_i temps.

Hypothèse III.2 : La matrice $G(x)$ est définie positive, et il existe $\bar{g}_1 > 0$ et $\bar{g}_0 > 0$, $\bar{g}_0 \in \mathbb{R}$, $\bar{g}_1 \in \mathbb{R}$, telle que : $0 < \bar{g}_0 I_p < G(x) < \bar{g}_1 I_p$, ou I_p est la matrice d'identité de dimension $p \times p$.

Remarque III.1 : Notons que l'hypothèse(III.2) assure que la matrice $G(x)$ est toujours régulière, par conséquent, le système (III.1) est linéarisable par un retour d'état statique. Bien que cette hypothèse limite la classe considérée de systèmes non linéaires MIMO, de nombreux systèmes physiques, tels que les systèmes robotiques, remplissent une telle propriété.

On définit l'erreur de poursuite comme suit

$$\begin{aligned} e_1(t) &= y_{d1}(t) - y_1(t) \\ &\vdots \\ e_p(t) &= y_{dp}(t) - y_p(t) \end{aligned} \quad (\text{III.4})$$

Et les erreurs filtrées par

$$\begin{aligned} \sigma_1(t) &= \left(\frac{d}{dt} + \lambda_1\right)e_1(t), \quad \lambda_1 > 0 \\ &\vdots \\ \sigma_p(t) &= \left(\frac{d}{dt} + \lambda_p\right)e_p(t), \quad \lambda_p > 0 \end{aligned} \quad (\text{III.5})$$

La dérivée des erreurs filtrées peuvent être données sous la forme :

$$\begin{aligned} \dot{\sigma}_1 &= v_1 - f_1(x) - \sum_{j=1}^p g_{1j}(x)u_j \\ &\vdots \\ \dot{\sigma}_p &= v_p - f_p(x) - \sum_{j=1}^p g_{pj}(x)u_j \end{aligned} \quad (\text{III.6})$$

où v_1, \dots, v_p est un signal définie par :

$$\begin{aligned} v_1 &= y_{d1}^{(r_1)} + B_{1,r_1-1} e_1^{(r_1-1)} + \dots + B_{1,1} \dot{e}_1 \\ &\vdots \\ v_p &= y_{dp}^{(r_p)} + B_{p,r_p-1} e_1^{(r_p-1)} + \dots + B_{p,1} \dot{e}_p \end{aligned} \quad (\text{III.7})$$

avec :

$$\beta_{i,j} = \frac{(r_i-1)!}{(r_i-j)!(j-1)!} \lambda_{i,j}, \quad i=1, \dots, p, \quad j=1, \dots, r_i-1. \quad (\text{III.8})$$

On note que

$$\begin{aligned} \sigma &= [\sigma_1 \dots \sigma_p]^T \\ v &= [v_1 \dots v_p]^T \end{aligned}$$

Donc on peut écrire la dérivée de l'erreur filtrée sous la forme compacte générale suivante :

$$\dot{\sigma}_i = v_i - F_i(x) - G(x)u_i, \quad i=1, \dots, p \quad (\text{III.9})$$

Remarque III.2 : L'équation $\sigma_i(t) = 0$ est une équation différentielle qui possède la solution $e_i(t) = 0$. De plus si $\sigma_i(t) \leq \Phi$ avec Φ une constante positive, l'erreur de poursuite vérifiée $|e_i^j(t)| \leq 2^j \lambda_i^{j-1} \Phi$, $j=0, \dots, r_i-1$ et $i=1, \dots, p$.

Notre objectif est de concevoir une loi de commande $u(t)$ pour que la sortie de système $y(t)$ suive son signal de référence.

Pour atteindre cet objectif on peut utiliser la loi de commande suivante :

$$u^* = G^{-1}(x)(-F(x) + v + \alpha \sigma) \quad (\text{III.10})$$

où $\alpha = \text{diag}[\alpha_1 \dots \alpha_p]$, avec $k_i > 0$.

On remplace u par u^* dans l'équation (III.9) on obtient

$$\dot{\sigma} = -\alpha \sigma \quad (\text{III.11})$$

où

$$\begin{aligned} \dot{\sigma}_1 &= -\alpha_1 \sigma_1 \\ &\vdots \\ \dot{\sigma}_p &= -\alpha_p \sigma_p \end{aligned} \quad (\text{III.12})$$

De ces dernières équations il résulte que $\sigma(t) \rightarrow 0$ lorsque $t \rightarrow 0$ donc $e(t) \rightarrow 0$ alors $y_i(t)$ suit son signal de référence $y_{ri}(t)$.

Maintenant on définit l'erreur entre u^* et u comme suit :

$$e_u = u^* - u \quad (\text{III.13})$$

Par addition et substitution de $G(x)u^*$ au deuxième terme de l'équation (III.9) on trouve :

$$\dot{\sigma} = v - f(x) + G(x)u + G(x)u^* - G(x)u^* \quad (\text{III.14})$$

à partir de (III.11) et (III.14), nous avons

$$\dot{\sigma} = -\alpha\sigma + G(x)e_u \quad (\text{III.15})$$

donc

$$\dot{\sigma} = -\alpha\sigma + G(x)u^* - G(x)u = -\alpha\sigma - G(x)u + G(x)u^* \quad (\text{III.16})$$

Pour le terme non linéaire $G(x)u$ on peut toujours trouver $\Omega = \text{diag}[\Omega_1 \dots \Omega_p]$, où $0 < \varpi_{0_i} < \Omega_i < \varpi_1$ est un gain constant inconnu, ϖ_{0_i} et ϖ_{1_i} sont des constantes positives, et $\Delta(x, u) = [\Delta_1(x, u) \dots \Delta_p(x, u)]$, représente l'écart entre $G(x)u$ et Ωu , tel que

$$G(x)u = \Omega u - \Delta(x, u) \quad (\text{III.17})$$

On remplace (III.17) dans (III.16) et on note

$$F(x, u, u^*) = \Delta(x, u) + G(x)u^* \quad (\text{III.18})$$

avec $F(x, u, u^*) = [F_1(x, u, u^*) \dots F_p(x, u, u^*)]$, on peut écrire l'équation (III.16) comme suit

$$\dot{\sigma} = -\alpha\sigma - \Omega u + F(x, u, u^*) \quad (\text{III.19})$$

on considère maintenant une hypothèse sur le terme $F(x, u, u^*)$.

Hypothèse III.3: La fonction non linéaire $F_i(x, u_i, u_i^*(x, v_i(t, \sigma_i)))$ est continue dans ses arguments, lipschitzienne par rapport à σ_i et borné à $\sigma_i = 0$, telle que

$$\left| F_i(x, u, u^*) \Big|_{\sigma_{i2}} - \left| F_i(x, u, u^*) \Big|_{\sigma_{i1}} \right| \leq L_i |\sigma_{i2} - \sigma_{i1}| \quad (\text{III.20})$$

$$\left| F_i(x, u, u^*) \Big|_{\sigma_i=0} \leq L_{i0} \quad (\text{III.21})$$

où L_{i0} et L_i sont des constantes positives.

Notons que la fonction non linéaire $F(x, u, u^*)$ est inconnue. Pour cela, un approximateur flou de cette fonction inconnue est considérée dans la conception du contrôleur.

III.2.2. Conception d'un approximateur flou

Afin de développer le contrôleur L1 adaptatif flou, nous supposons que le terme $F_i(x, u, u^*)$, $i = 1, \dots, p$, peut être approchée sur un ensemble compact $D_x \times \mathbb{R}$ par un système flou jusqu'à la précision désirée :

$$\begin{aligned} F_i(x, u, u^*(x, v(t, \sigma))) &= \omega^T(\pi) \theta_i^* + \varepsilon_i(\pi), \\ |\varepsilon_i(\pi)| &\leq \varepsilon_i^* \end{aligned} \quad (\text{III.22})$$

L'architecture du système flou utilisé dans ce travail sera présentée prochainement. Dans (III.22), $\pi = [x, u, v]^T$ design le vecteur d'entrée du système flou et $\omega(\pi)$ est un vecteur de fonctions de base fixé par le concepteur, $\varepsilon_i(\pi)$ est un vecteur optimal inconnu qui minimise d'erreur d'approximation sur un ensemble compacte opérationnel, $\theta^* \in \Theta$, et ε^* est bande uniforme pour ε .

Hypothèse III.4 : Soit $\theta_i^*(t)$ une fonction dérivable avec une dérivée uniformément bornée

$$\|\dot{\theta}_i^*(t)\| \leq d_{\theta^*} < \infty, \quad \forall t \geq 0 \quad (\text{III.23})$$

Dans ce travail on utilise un système flou de Takagi-Sugeno qui établit un passage à partir d'un vecteur d'entrée $\pi = [\pi_1, \dots, \pi_m]^T \in D_\pi \subset \mathbb{R}^m$ à une variable scalaire de sortie $y_f \in \mathbb{R}$ telle que $D_\pi = D_{\pi_1} \times \dots \times D_{\pi_m}$ et $D_{\pi_1} \subset \mathbb{R}$. Pour chaque entrée π_i , nous définissons M_i ensembles floue nommée F_i^j , $j = 1, \dots, M_i$, alors le système flou se compose d'une séquence de N règles de la forme donnée comme suit :

$$R^k : \text{if } \pi_1 \text{ est } G_1^k \text{ et } \dots \text{ et } \pi_m \text{ est } G_m^k, \text{ alors } y_f \text{ est } y_f^k \quad (k = 1, \dots, N) \quad (\text{III.24})$$

où $G_i^k \in \{F_i^1, \dots, F_i^{M_i}\}$, $i = 1, \dots, m$, y_f^k désigne la sortie de la K th règle. L'utilisation des fonctions d'appartenance de type singleton et le produit pour l'inférence, donne la sortie finale du système flou suivante

$$y_f(\pi) = \frac{\sum_{k=1}^N \mu_k(\pi) y_f^k}{\sum_{k=1}^N \mu_k(\pi)} \quad (\text{III.25})$$

où $\mu_k(\pi) = \prod_{i=1}^m \mu_{G_i^k}(\pi_i)$ avec $G_i^k = \{\mu_{F_i^1}, \dots, \mu_{F_i^{M_i}}\}$ et $\mu_{F_i^j}(\pi_i)$ est la fonction d'appartenance de l'ensemble floue F_i^j . La sortie donnée par l'équation (III.25) peut aussi exprimée dans la forme compacte suivante

$$y_f(\pi) = w^T(\pi) \theta \quad (\text{III.26})$$

Où θ est un vecteur contenant tous les paramètres de conclusion et $w(\pi) = [w_1(\pi), \dots, w_N(\pi)]$ est un ensemble flou des fonctions de base donné comme suit

$w_k(\pi) = \frac{\mu_k(\pi)}{\sum_{k=1}^N \mu_k(\pi)}$, $k = 1, \dots, N$. En d'autre part le système flou est supposée être bien défini

par le concepteur, i.e. $\sum_{k=1}^N \mu_k(\pi) \neq 0$ pour tout $\pi \in \Omega_\pi$. Il est important de rappeler qu'un système flou sous la forme (III.19) peut approximer une fonction non linéaire continue sur un domaine d'intérêt pour toute précision souhaitée.

On substitue (III.22) dans (III.19), on obtient

$$\dot{\sigma} = -\alpha\sigma - \Omega u + w^T(\pi)\theta^* + \varepsilon(\pi) \quad (\text{III.27})$$

avec $\varepsilon^*(\pi) = [\varepsilon^*_{1}(\pi), \dots, \varepsilon^*_{p}(\pi)]$.

III.3. Structure de la commande L1 adaptative floue

Basée sur l'équation (III.19), nous considérons le prédicteur suivant

$$\dot{\hat{\sigma}}_i = -\alpha_i \hat{\sigma}_i - \hat{\Omega}_i u_i + w^T(\pi) \hat{\theta}_i \quad (\text{III.28})$$

Où $\hat{\sigma}_i \in \mathbb{R}$ est la prédiction de σ_i , $\hat{\Omega}_i$ et $\hat{\theta}_i$ sont des estimées de Ω_i et θ_i et gouvernées par les lois d'adaptation suivantes

$$\dot{\hat{\Omega}}_i = \Gamma_i \text{Proj}(\hat{\Omega}_i, \tilde{\sigma}_i u_i) \quad (\text{III.29})$$

$$\dot{\hat{\theta}}_i = \Gamma_i \text{Proj}(\hat{\theta}_i, -\tilde{\sigma}_i w(\pi)) \quad (\text{III.30})$$

où $\tilde{\sigma}_i = \hat{\sigma}_i - \sigma_i$ est l'erreur de prédiction, $\Gamma_i > 0$ est le gain d'adaptation et $\text{Proj}(\cdot, \cdot)$ désigne l'opérateur de projection défini, pour les deux vecteurs $O \in \mathbb{R}^p$ et $\mathcal{G} \in \mathbb{R}^p$, comme suit

$$\text{Proj}(O, \mathcal{G}) = \begin{cases} \mathcal{G} - \frac{\nabla h(O) \nabla h^T(O)}{\|\nabla h(O)\|} \mathcal{G} h(O) & \text{if } h(O) > 0 \text{ et } \nabla h^T(O) \mathcal{G} > 0 \\ \mathcal{G} & \text{sinon} \end{cases} \quad (\text{III.31})$$

où $h(O) = \frac{(\varepsilon_0 + 1)O^T O - O_{\max}^2}{\varepsilon_0 O_{\max}^2}$ avec $O_{\max} \in \mathbb{R}^+$ étant la norme imposée pour le vecteur

O et $\varepsilon_0 \in \mathbb{R}^+$ représente la tolérance de projection de notre choix.

La loi de commande est choisie comme suit

$$u_i = K_i D_i(s) \eta_i(s), \quad i = 1, \dots, p \quad (\text{III.32})$$

où $\eta_i(s)$ est la transformée de Laplace du signal $\eta_i(t) = -\hat{\Omega}_i u_i + w^T(\pi) \hat{\theta}_i$, K_i est un gain de rétroaction positive, et $D(s)$ est une fonction de transfert strictement propre conduit à un filtre stable en boucle fermé donné par

$$C_i(s) = \frac{\Omega_i K_i D(s)}{1 + \Omega_i K_i D(s)} \quad (\text{III.33})$$

où $C(0) = 1$. Avec un choix simple de $D(s)$ tel que $D(s) = \frac{1}{s}$, nous obtenons un filtre du premier ordre de la forme suivante

$$C_i(s) = \frac{\Omega_i K_i}{1 + \Omega_i K_i} \quad (\text{III.34})$$

Le contrôleur L_1 adaptative flou proposé consiste du prédicteur (III.28) et les lois de d'adaptations (III.29) et (III.30) et le contrôleur (III.32) réservé pour la condition de la norme suivante

$$\|B_i(s)\|_{L_1} L_i < 1 \quad (\text{III.35})$$

où $B_i(s) = H_i(s)(1 - C_i(s))$, $H_i(s) = (s + \alpha_i)^{-1}$ et L_i est donnée dans (III.35)

Remarque III.3. Les paramètres de conception positifs λ et α dans (III.5) et (III.11) respectivement sont utilisés pour spécifier la dynamique souhaitée en boucle fermé. Le filtre à passe bande limitée $C(s)$ dans (III.33), défini par le gain de rétroaction K et la fonction de transfert strictement propre $D(s)$ est conçu pour assurer un bon suivi des signaux de référence et pour compenser les effets indésirables des incertitudes dans une gamme de fréquence spécifiée.

III.4. Application de la commande L1 adaptative floue à un robot manipulateur

Dans cette partie, pour illustrer l'efficacité de l'approche de commande présentée, nous avons procédé à une application sur un robot manipulateur type sériel montré dans la figure (III.1). Le modèle dynamique du robot manipulateur est

$$\begin{bmatrix} M_{11} & M_{12} \\ M_{21} & M_{22} \end{bmatrix} \begin{bmatrix} \ddot{q}_1 \\ \ddot{q}_2 \end{bmatrix} + \begin{bmatrix} -h\dot{q}_2 & -h(\dot{q}_1 + \dot{q}_2) \\ h\dot{q}_1 & 0 \end{bmatrix} \begin{bmatrix} \dot{q}_1 \\ \dot{q}_2 \end{bmatrix} = \begin{bmatrix} \tau_1 \\ \tau_2 \end{bmatrix} \quad (\text{III.36})$$

avec :

$$\begin{aligned}
M_{11} &= a_1 + 2a_3 \cos(q_2) + 2a_4 \sin(q_2), \\
M_{22} &= a_2, \\
h &= a_3 \sin(q_2) - a_4 \cos(q_2) \\
M_{21} = M_{12} &= a_2 + a_3 \cos(q_2) + a_4 \sin(q_2), \\
a_2 &= I_e + m_e l_{ce}^2, a_3 = m_e l_1 l_{ce} \sin \delta_e.
\end{aligned} \tag{III.37}$$

Dans la simulation, nous utilisons les paramètres suivants :

$$m_1 = 1, m_e = 2, l_1 = 1, l_{c1} = 0.5, l_{ce} = 0.6, I_1 = 0.12, I_e = 0.25, \delta_e = 30^\circ$$

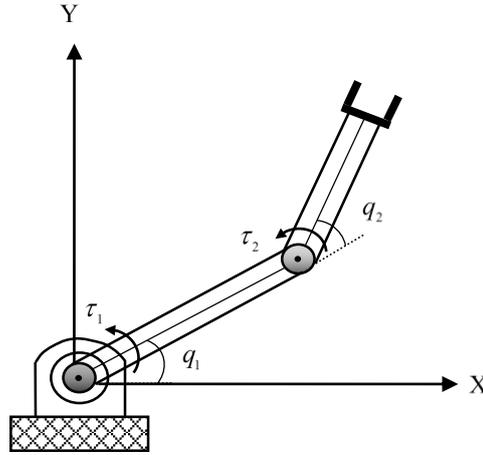


Figure III.1 : Robot manipulateur de deux degrés de liberté.

Choisissons $y = [q_1, q_2]^T$, $u = [\tau_1, \tau_2]$, $x = [q_1, \dot{q}_1, q_2, \dot{q}_2]$ et

$$f(x) = \begin{bmatrix} f_1(x) \\ f_2(x) \end{bmatrix} = -M^{-1} \begin{bmatrix} -h\dot{q}_2 & -h(\dot{q}_1 + \dot{q}_2) \\ h\dot{q}_1 & 0 \end{bmatrix} \begin{bmatrix} \dot{q}_1 \\ \dot{q}_2 \end{bmatrix}$$

$$G(x) = \begin{bmatrix} g_{11}(x) & g_{12}(x) \\ g_{21}(x) & g_{22}(x) \end{bmatrix} = M^{-1} = \begin{bmatrix} M_{11} & M_{12} \\ M_{21} & M_{22} \end{bmatrix}^{-1}$$

Donc, le système du robot donné par (III.19) peut s'écrire comme suit

$$y = f(x) + G(x)u \tag{III.38}$$

L'objectif de commande est de forcer les deux articulations du robot q_1 et q_2 à suivre les deux trajectoires sinusoïdales désirées $y_{d1}(t) = \sin(t)$ et $y_{d2}(t) = \cos(t)$ respectivement.

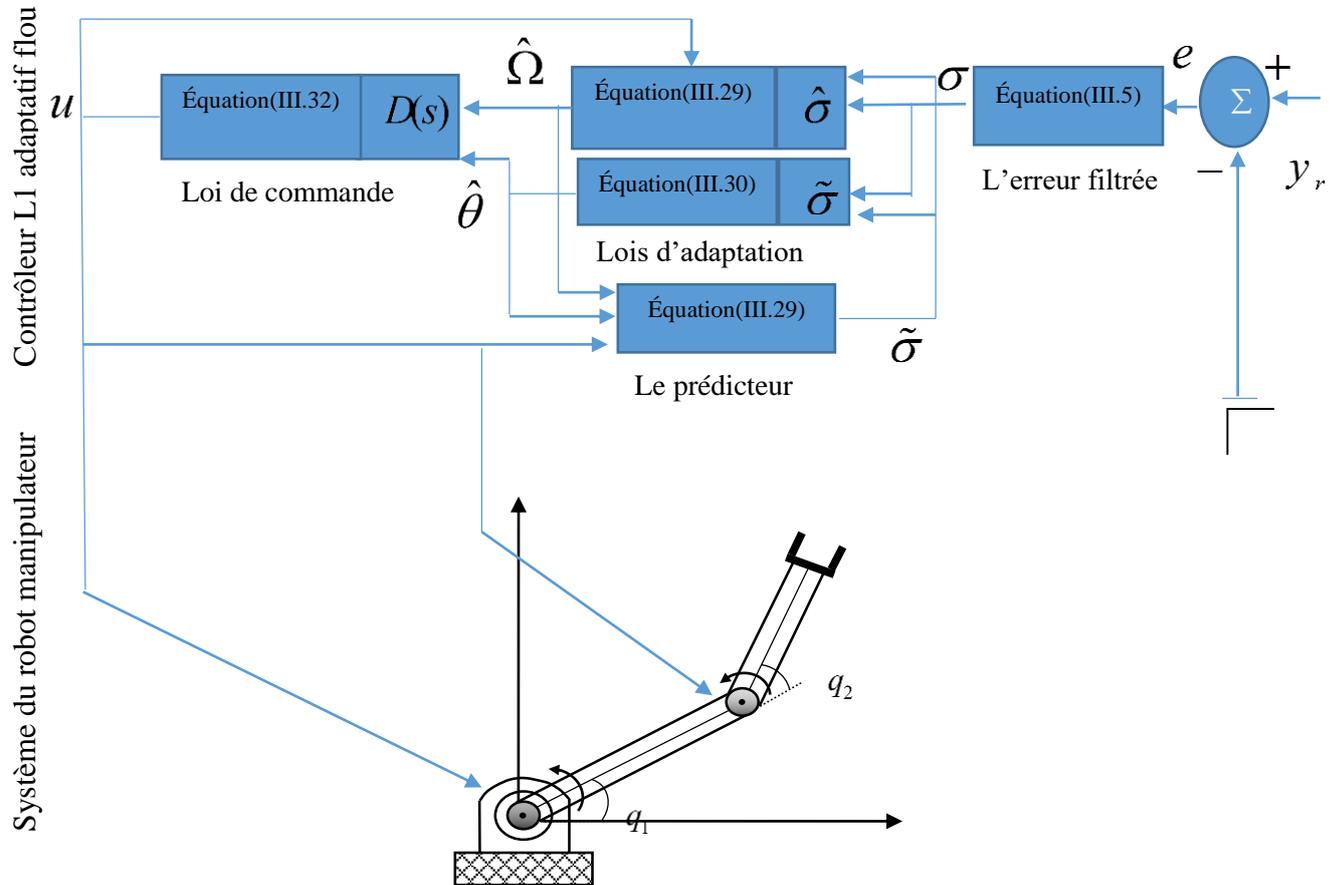


Figure III.2 : Contrôleur L1 adaptatif flou appliqué à un robot manipulateur.

III.4.1. Résultats de simulation

Dans cette simulation, les fonctions non linéaires $f(x)$ et $G(x)$ sont supposées inconnues. Le système de commande ne nécessite pas la connaissance du modèle du système comme dans le modèle des contrôleurs adaptatif conventionnels. En fait, le modèle dynamique du robot est seulement requis pour la simulation. Le robot manipulateur est constitué de deux sous systèmes de degré relatif $r_i = 2$. Donc, pour ce système nous avons $p = 2$ et $n = \sum_{i=1}^2 r_i = 4$.

Paramètres de simulation

Les conditions initiales du robot sont $x_0 = [0.25 \ 0 \ 0.25 \ 0]^T$ et les valeurs initiales paramètres utilisés dans cette simulation sont choisis comme suit : $\lambda_1 = 2, \lambda_2 = 2$, et les gains de commande $k_1 = k_2 = 50$, la résolution de l'équation différentielle a été faite via l'algorithme

Rung-Kutta d'ordre 4 avec un pas de simulation $\Delta t = 0.001\text{sec}$ et un temps de simulation $t_f = 15\text{sec}$.

Les résultats sont présentés dans les figures suivantes :

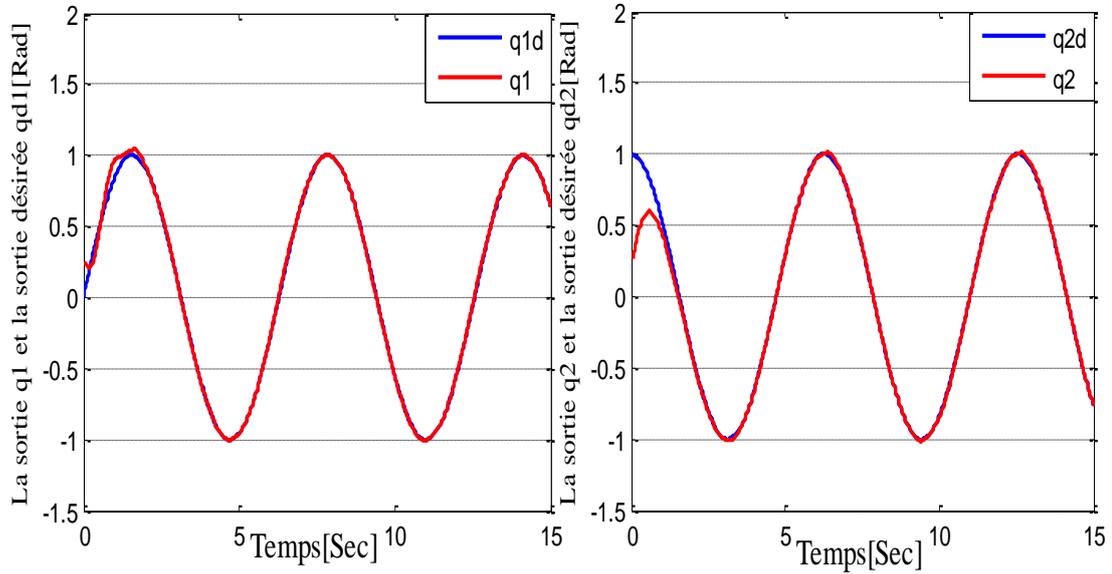


Figure III.3 : Position articulaires et position désirées.

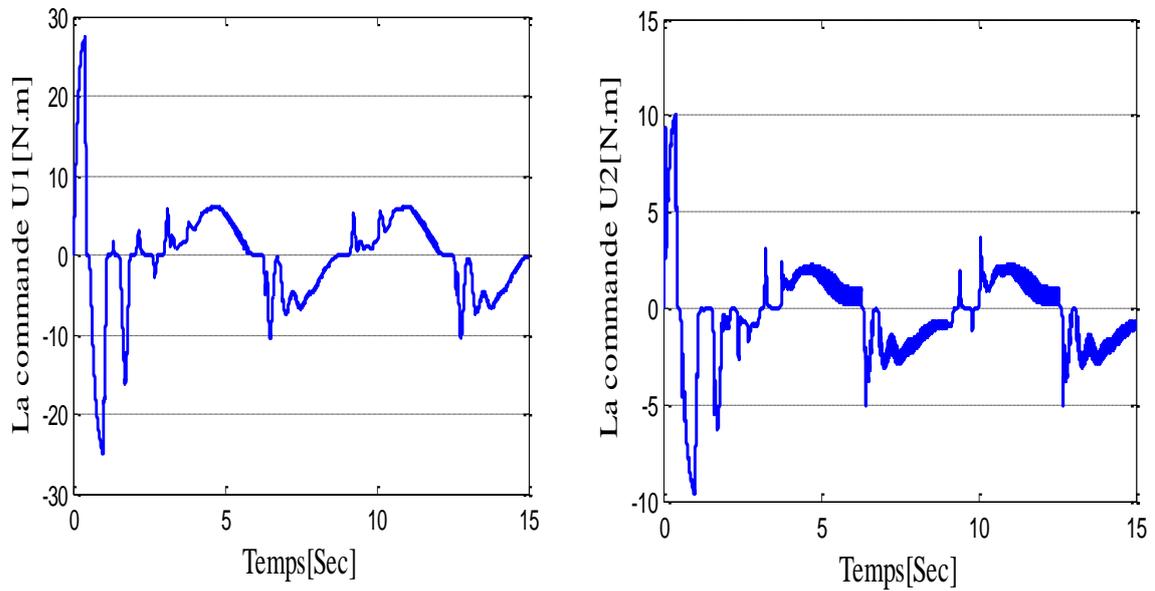


Figure III.4 : Les commandes appliquées au robot manipulateur.

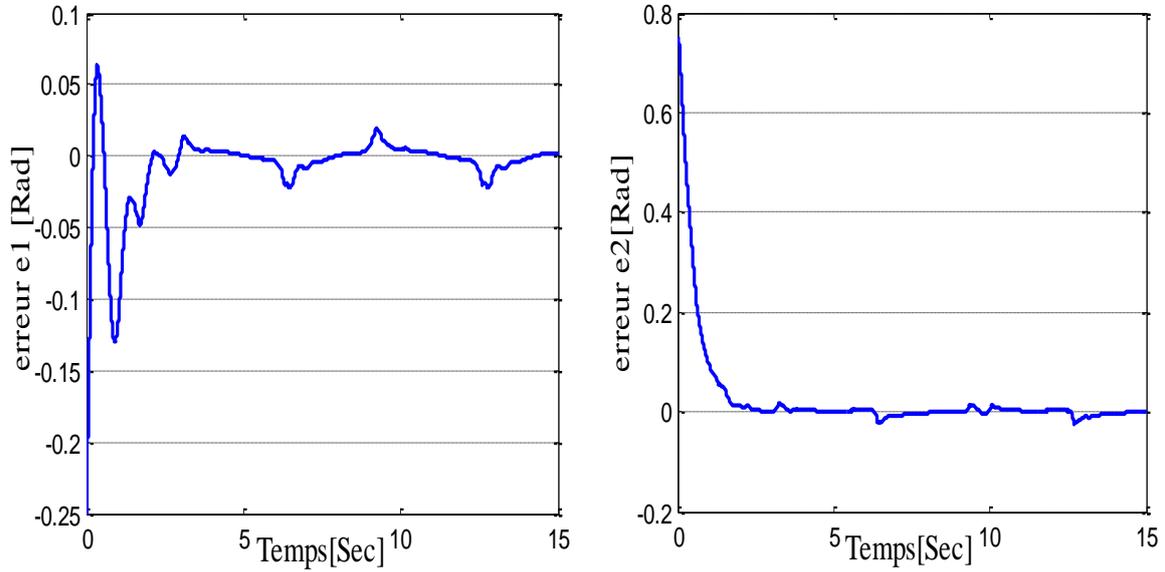


Figure III.5 : Les erreurs de poursuite.

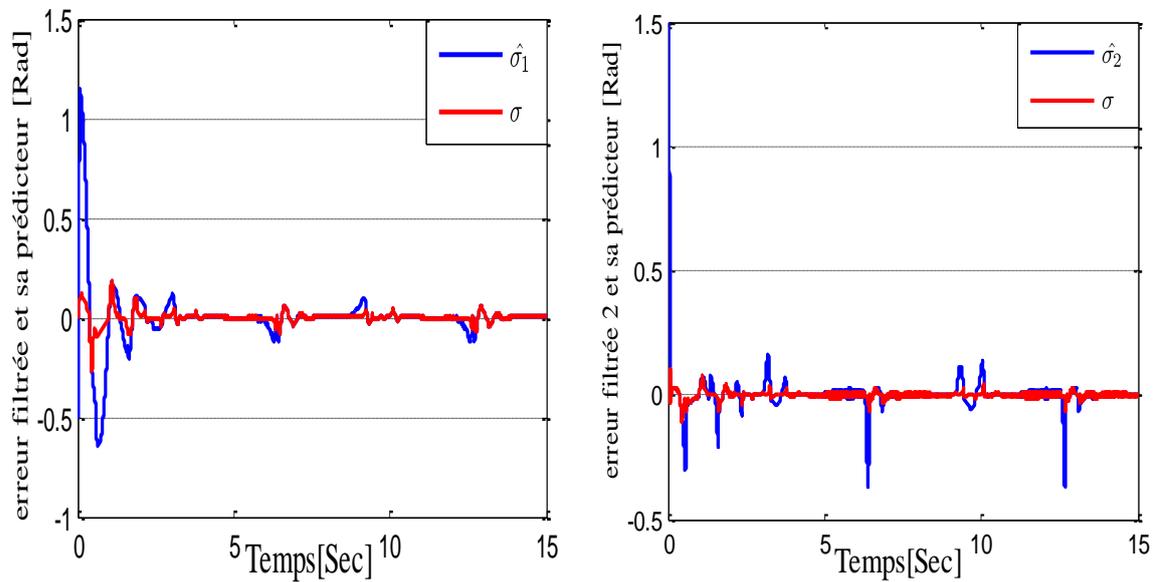


Figure III.6 : Les erreurs filtrées et leurs prédictions.

III.4.3 Interprétation des résultats

Les résultats de simulation obtenus montrent bien les performances de la commande L1 adaptative floue. Il apparaît clairement que les positions articulaires suivent les positions

désirées, avec des erreurs de poursuite qui tendent vers zéro. Nous remarquons aussi que les erreurs filtrées suivent leurs prédictions et convergent vers zéro. La commande et les paramètres estimés sont bornés. Malgré l'utilisation d'un gain d'adaptation très élevé, il est possible d'assurer des bonnes performances.

Test de robustesse : pour toutes les simulations effectuées dans cette partie nous supposons que chaque articulation de ce robot manipulateur est subie à une perturbation $d = 0.1\sin(2t)e^{-0.1t}$. les figures issues de cette étude sont les suivantes :

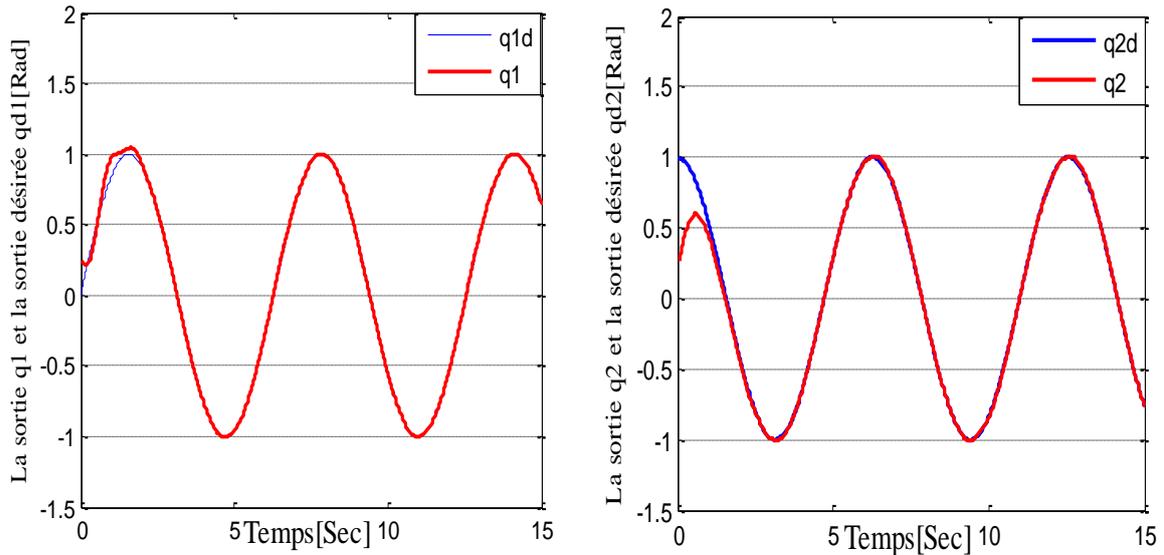


Figure.III.7 : Positions articulaires et positions désirées.

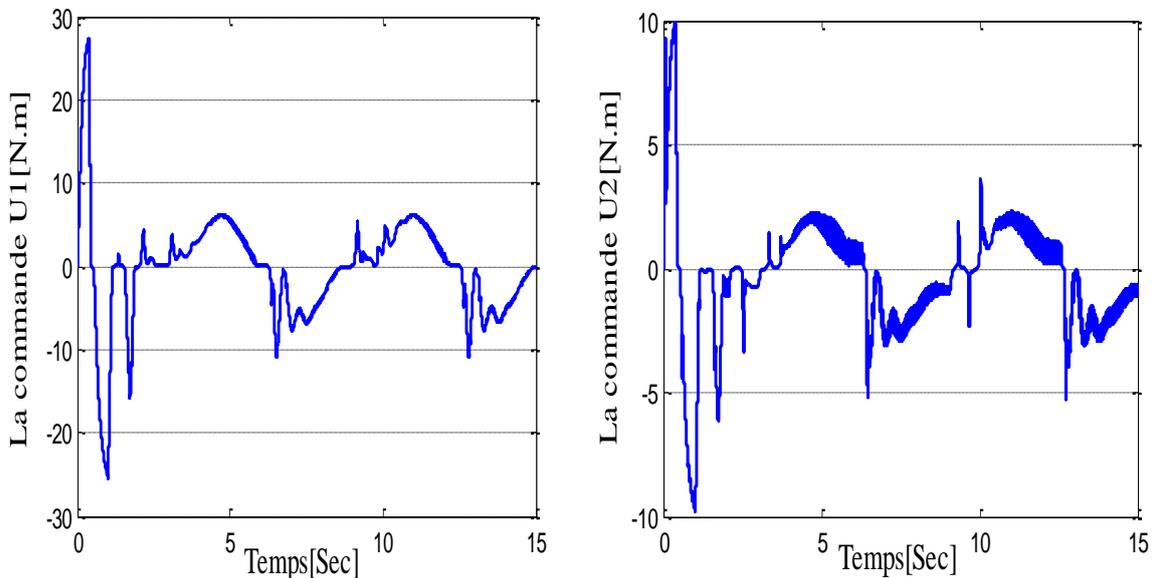


Figure.III.8 : Les commandes appliquées au robot manipulateur.

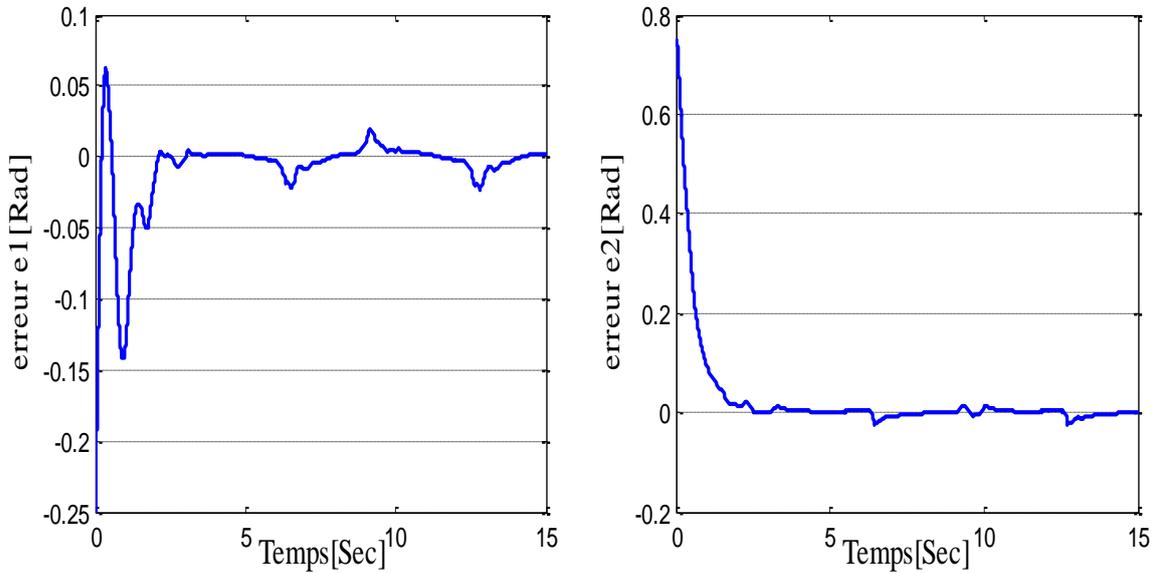


Figure.III.9 : Les erreurs de poursuite.

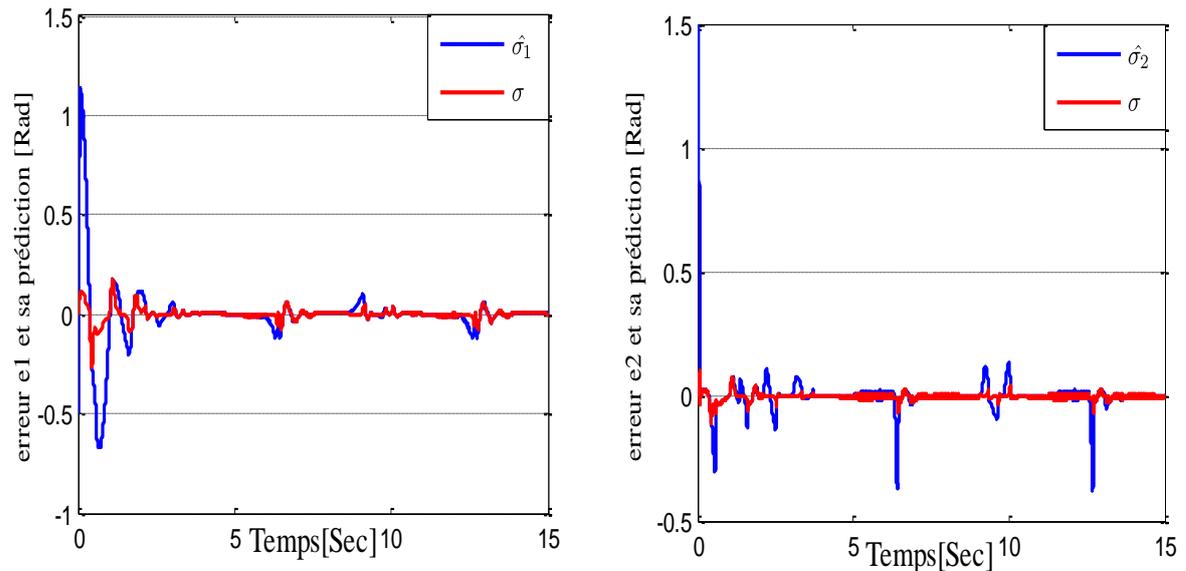


Figure III.10 : Les erreurs filtrées et leurs prédictions.

Interprétation des résultats :

Nous remarquons dans ce test que les positions articulaires suivent les positions désirées et les erreurs de poursuite tendent vers zéro, les commandes sont aussi acceptables et ceci malgré qu'on a subit le système à une perturbation. Donc, on peut constater la robustesse de la commande L1 adaptative floue.

III.5. Conclusion

Dans ce chapitre, nous avons présenté la conception d'une loi de commande L1 adaptative floue pour une classe des systèmes non linéaires affine en entrée. La commande adaptative proposée n'exige pas la connaissance du modèle mathématique du système. Les performances de cette loi de commande sont examinées sur un modèle d'un robot manipulateur, les résultats obtenus montrent bien les performances de cette approche de commande pour commander de tels systèmes.

Dans le prochain chapitre, nous allons faire une étude comparative (qualitative et quantitative) entre la commande L1 adaptative et deux autres commandes classiques : la commande par mode de glissement et la commande backstepping.



Chapitre .IV
Étude comparative

Chapitre .IV
Étude comparative

Chapitre .IV Étude comparative

IV.1. Introduction

Ce chapitre est consacré à une étude comparative des trois commandes appliquées à un robot manipulateur. Cette comparaison est faite dans les mêmes conditions de fonctionnement (références, perturbations, ...) et dans la même configuration de simulation numérique (pas d'échantillonnage, durée de la simulation, ...). Ces trois commandes appliquées au robot manipulateur sont la commande par mode de glissement, la commande backstepping et la commande L1 adaptative floue.

IV.2. Commande par mode de glissement SMC

IV.2.1. Commande par modes glissants d'ordre 1

La commande par mode glissant d'ordre 1 (SMC) est une commande à structure variable pouvant changer de structure et commutant entre deux valeurs suivant une logique de commutation bien spécifique $s(x)$.

Le principe de la commande par modes glissants est de contraindre le système à atteindre une surface donnée appelée surface de glissement et d'y demeurer jusqu'à l'équilibre. Cette commande se fait en deux étapes [33] : La convergence vers la surface et ensuite le glissement le long de celle-ci (**Figure IV.1**).

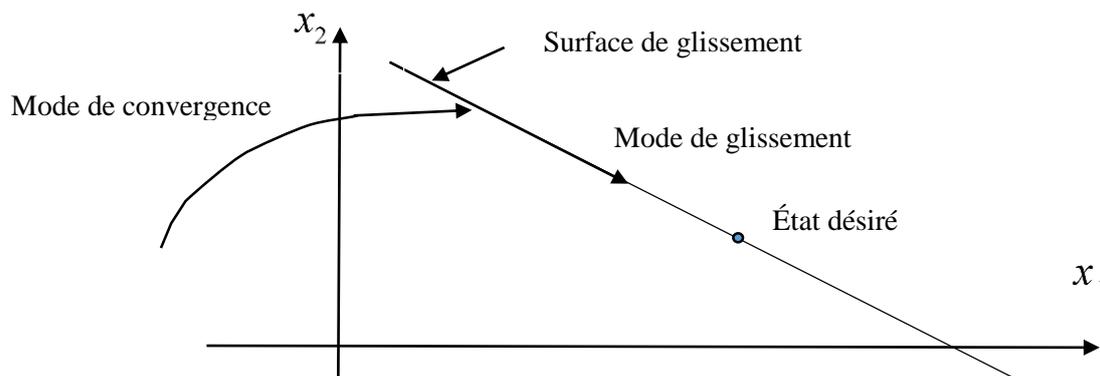


Figure IV.1 : Différents modes de convergence pour la trajectoire d'état.

IV.2.2. Synthèse de la loi de commande

La synthèse de la commande par mode de glissement se fait en trois étapes :

- Le choix de la surface de glissement.
- Établir la condition de convergence.
- Déterminer la loi de commande qui permet d'atteindre la surface et d'y demeurer.

IV.2.2.1. Choix de la surface de glissement

Soit le système décrit par l'équation différentielle suivante :

$$\dot{x} = f(x,t) + g(x,t)u \quad (\text{IV.1})$$

où f et g sont des fonctions non linéaires, g est supposée inversible, u : l'entrée du système, et x l'état du système.

Soit x_d la consigne désirée et e l'erreur de poursuite définie par

$$e = x - x_d \quad (\text{IV.2})$$

La formule générale de la surface de glissement est définie en fonction de l'ordre du système comme suit

$$s(x) = \left(\frac{d}{dt} + \lambda \right)^{n-1} e(x) \quad (\text{IV.3})$$

où n : le degré relatif de système par rapport à la sortie $y(t)$. Il représente le nombre minimum de fois qu'il faut dériver la sortie $y(t)$ par rapport au temps pour y avoir apparaitre l'entrée.

IV.2.2.2. Condition d'existence du mode de glissement

Les conditions d'existence et de convergence sont les critères qui permettent aux différentes dynamiques du système de converger vers la surface de glissement et d'y rester indépendamment de la perturbation. Il existe deux considérations pour assurer le mode de convergence [34].

- C'est la première condition de convergence, elle est proposée et étudiée par ELMYANOV et UTKIN. Il s'agit de donner à la surface une dynamique convergente vers zéro, elle est donnée par:

$$\begin{cases} \dot{s}(x) > 0 & \text{Si } s(x) < 0 \\ \dot{s}(x) < 0 & \text{Si } s(x) > 0 \end{cases} \quad (\text{IV.4})$$

Cette condition peut être formulée comme suit:

$$s(x)\dot{s}(x) < 0 \quad (\text{IV.5})$$

Si on choisit comme fonction de Lyapunov :

$$v(x) = \frac{1}{2} s^2(x) \quad (\text{IV.6})$$

La dérivée de cette fonction est :

$$\dot{v}(x) = s(x).\dot{s}(x) \quad (\text{IV.7})$$

Pour que la fonction $v(x)$ puisse décroître, il suffit d'assurer que sa dérivée soit négative. Ceci n'est vérifiée que si la condition (IV.8) est vérifiée.

$$s(x)\dot{s}(x) < 0 \quad (\text{IV.8})$$

L'équation (IV.6), explique que le carré de la distance entre un point donné du plan de phase et la surface de glissement exprimée par $s^2(x)$ diminue tout le temps, c'est à dire que l'état de système sera attiré vers la surface de glissement, d'où le nom d'attractivité. Cette condition suppose un régime glissant idéal ou la fréquence de commutation est infinie.

IV.2.2.3. Loi de commande

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

Lorsqu'il y a un régime glissant, la dynamique du système est indépendante de la loi de commande. De ce fait, on peut introduire une partie continue pour diminuer l'amplitude de la discontinuité, on aura donc

$$u = u_{eq} + u_c \quad (\text{IV.9})$$

Commande équivalente

La commande équivalente proposée par FILIPOV et UTKIN correspond à la commande du système nominale permettant de satisfaire la condition : $\dot{s}(x, t) = 0$

Commande de commutation

La commande de commutation permettant de garantir la condition d'attractivité et responsable du glissement, la forme la plus simple que peut prendre est celle d'un relais.

$$u_c = \dot{s}(x) = -k \text{sign}(s(x)) \quad (\text{IV.10})$$

où : k est une constante positive qui représente le gain de la commande discontinue (**Figure.IV.2**)

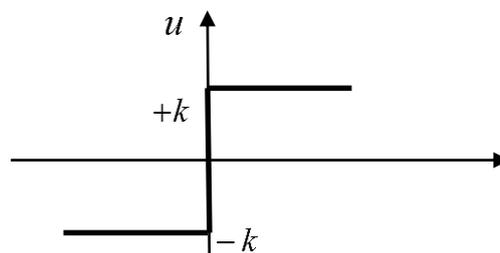


Figure IV.2 : Représentation de la commande discontinue.

IV.2.2.4. Avantages et inconvénients de la commande par mode glissant

Un des principaux avantages résidant dans l'utilisation de la commande par mode glissant est que tout système commandé par cette technique aura des propriétés implicites de

« robustesse », car le système sera contraint à chaque instant à respecter la condition de glissement.

En général, on combine cette technique avec des méthodes de commande à structure variable, dans le but de pouvoir définir de manière explicite la robustesse du système, vis-à-vis des incertitudes paramétriques du modèle et des perturbations du système.

Quand l'état du système se trouve à la limite du changement de structure, des commutations successives à fréquences élevées peuvent se produire. Ce phénomène pouvant produire une sollicitation excessive des actionneurs, est appelé **broutement** (en anglais « chattering »). Pour certaines applications, le broutement n'est pas acceptable pour deux raisons principales :

- Pour des fréquences de broutement élevées, dans certains cas, le système n'est plus modélisé correctement et peut devenir instable en boucle fermée.
- Le broutement produit une dynamique importante sur les actionneurs (moteurs), ce qui peut réduire leurs « temps de vie » et d'être la cause de vibrations dangereuses.

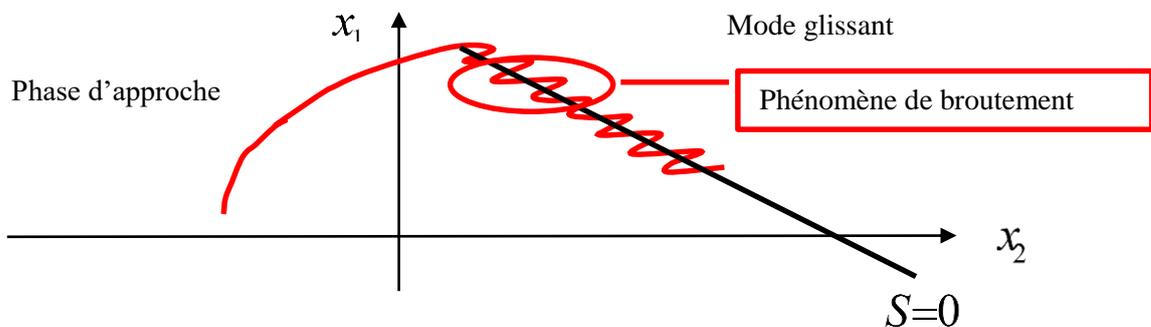


Figure.IV.3: Plan de phase réel.

IV.2.3. Commande par mode de glissement appliquée à un robot manipulateur

La commande par mode de glissement présenté dans cette partie a été appliquée à un robot manipulateur. À partir de son modèle dynamique cité dans le chapitre III à l'équation (III.36) et avec le choix des variables d'états suivant $x_1 = q_1$, $x_2 = \dot{q}_1$, $x_3 = q_2$ et $x_4 = \dot{q}_2$, le modèle mathématique de mouvement de robot manipulateur est donné par :

$$\begin{cases} \dot{x}_1 = x_2 \\ \dot{x}_2 = f_1(x) + g_{11}(x)u_1 + g_{12}(x)u_2 \\ \dot{x}_3 = x_4 \\ \dot{x}_4 = f_2(x) + g_{21}(x)u_1 + g_{22}(x)u_2 \end{cases} \quad (\text{IV.11})$$

avec

$$f(x) = \begin{bmatrix} f_1(x) \\ f_2(x) \end{bmatrix} = -M^{-1} \begin{bmatrix} -h\dot{q}_2 & -h(\dot{q}_1 + \dot{q}_2) \\ h\dot{q}_1 & 0 \end{bmatrix} \begin{bmatrix} \dot{q}_1 \\ \dot{q}_2 \end{bmatrix} \quad (\text{IV.12})$$

$$G(x) = \begin{bmatrix} g_{11}(x) & g_{12}(x) \\ g_{21}(x) & g_{22}(x) \end{bmatrix} = M^{-1} = \begin{bmatrix} M_{11} & M_{12} \\ M_{21} & M_{22} \end{bmatrix}^{-1} \quad (\text{IV.13})$$

IV.2.3.1. Synthèse des lois de commande SMC pour le robot manipulateur

Dans cette section, une structure de commande pour le robot manipulateur basée sur la commande par mode de glissement est développée. Notre objectif est de commander ce robot manipulateur en position suivant les deux angles q_1 et q_2 .

Le robot manipulateur est constitué de deux sous-systèmes de degré relatif $r_i = 2$. donc, pour ce système nous avons $p = 2$ et $\sum_{i=1}^2 r_i = 4$.

• Commande du premier sous-système

Pour la commande en position du premier sous-système, suivant l'angle q_1 , avec la technique de commande par mode de glissement, la loi de commande est constituée de deux parties : la commande équivalente et le terme de correction. A partir de la dynamique du premier sous-système suivant l'angle q_1

$$\begin{cases} \dot{x}_1 = x_2 \\ \dot{x}_2 = f_1(x) + g_{11}(x)u_1 + g_{12}(x)u_2 \\ y_1 = x_1 \end{cases} \quad (\text{IV.14})$$

Pour ce sous-système, le degré relative est $r_1 = 2$, pour cela nous choisissons la surface de glissement comme suit

$$s_1 = \dot{e}_1 + \lambda_1 e_1 \quad (\text{IV.15})$$

avec $\lambda_1 > 0$, $e_1 = q_1 - q_{1d}$, l'erreur entre l'angle q_1 du robot manipulateur et celle désirée.

En utilisant la théorie de la commande par mode de glissement, la loi de commande est donnée par

$$u_1 = u_{1eq} + u_{1gliss} \quad (\text{IV.16})$$

avec u_{1gliss} est le terme de correction défini par

$$u_{1gliss} = -k_1 \text{sign}(s_1), k_1 > 0 \quad (\text{IV.17})$$

et u_{1eq} la commande équivalente calculée lorsque

$$\begin{cases} s_1 = 0 \\ \text{et} \\ \dot{s}_1 = 0 \end{cases} \quad (\text{IV.18})$$

on dérive la surface s_1 par rapport au temps, on trouve

$$\dot{s}_1 = \ddot{e}_1 + \lambda_1 \dot{e}_1 \quad (\text{IV.19})$$

à partir de (IV.18) et (IV.19), on obtient

$$\dot{s}_1 = 0 \Leftrightarrow \ddot{e}_1 + \lambda_1 \dot{e}_1 = 0 \quad (\text{IV.20})$$

On utilisant \dot{e}_1 et \ddot{e}_1 dans les relations de l'équation (IV.14), on trouve

$$\dot{x}_2 - \ddot{x}_{1d} + \lambda_1 (x_2 - x_{1d}) = 0 \quad (\text{IV.21})$$

En utilisant (IV.14), donc l'équation (IV.22) devient

$$f_1(x) + g_{11}(x)u_{1eq} + g_{12}(x)u_{2eq} - \ddot{x}_{1d} + \lambda_1 (x_2 - x_{1d}) = 0 \quad (\text{IV.22})$$

La commande u_{1eq} est calculée dans (IV.33).

• Commande du deuxième sous-système

Pour la commande en position du deuxième sous-système suivant l'angle q_2 avec la technique de commande par mode de glissement, la dynamique du deuxième sous-système de ce robot manipulateur selon cet angle est donnée par

$$\begin{cases} \dot{x}_3 = x_4 \\ \dot{x}_4 = f_2(x) + g_{21}(x)u_1 + g_{22}(x)u_2 \\ y_2 = x_3 \end{cases} \quad (\text{IV.23})$$

Pour le sous-système (IV.24), le degré relative est $r_2 = 2$, pour cela nous choisissons la surface de glissement comme suit

$$s_2 = \dot{e}_2 + \lambda_2 e_2 \quad (\text{IV.24})$$

avec $\lambda_2 > 0$, $e_2 = q_2 - q_{2d}$, l'erreur entre l'angle q_1 et celle désirée.

en utilisant la théorie de la commande par mode de glissement la loi de commande est donnée par :

$$u_2 = u_{2eq} + u_{2gliss} \quad (\text{IV.25})$$

avec u_{2gliss} est le terme de correction défini par

$$u_{2gliss} = -k_2 \text{sign}(s_2), k_2 > 0 \quad (\text{IV.26})$$

et u_{2eq} la commande équivalente calculée lorsque

$$\begin{cases} s_2 = 0 \\ \text{et} \\ \dot{s}_2 = 0 \end{cases} \quad (\text{IV.27})$$

On dérive la surface s_2 par rapport au temps, on trouve

$$\dot{s}_2 = \ddot{e}_2 + \lambda_2 \dot{e}_2 \quad (\text{IV.28})$$

À partir de (IV.27) et (IV.28) on obtient

$$\dot{s}_2 = 0 \Leftrightarrow \ddot{e}_2 + \lambda_2 \dot{e}_2 = 0 \quad (\text{IV.29})$$

en utilisant \dot{e}_2 et \ddot{e}_2 dans les relations de (IV.23), on trouve

$$\dot{x}_4 - \ddot{x}_{3d} + \lambda_2 (x_4 - \dot{x}_{3d}) = 0 \quad (\text{IV.30})$$

en utilisant (IV.29), donc l'équation (IV.30) devient

$$f_2(x) + g_{21}(x)u_{1eq} + g_{22}(x)u_{2eq} - \ddot{x}_{3d} + \lambda_2 (x_4 - \dot{x}_{3d}) = 0 \quad (\text{IV.31})$$

À partir des équations (IV.22) et (IV.31), la commande équivalente est donnée par

$$u = \begin{bmatrix} u_{1eq} \\ u_{2eq} \end{bmatrix} = G^{-1}(x) \left(\begin{bmatrix} \ddot{x}_{1d} - Vx_4 - \lambda_1 (x_2 + Vx_3 - x_{1d}) \\ \ddot{x}_{3d} - \lambda_2 (x_4 - \dot{x}_{3d}) \end{bmatrix} - \begin{bmatrix} f_1(x) \\ f_2(x) \end{bmatrix} \right) \quad (\text{IV.32})$$

Finalement, la commande u_1 et la commande u_2 qui assurent la commande de l'angle q_1 et la commande de l'angle q_2 sont données comme suit

$$\begin{aligned} u_1 &= u_{1eq} - k_1 \text{sign}(s_1) \\ u_2 &= u_{2eq} - k_2 \text{sign}(s_2) \end{aligned} \quad (\text{IV.33})$$

L'objectif de cette commande est de forcer les deux articulations de robot manipulateur à suivre les deux consignes $y_{d1}(t)$ et $y_{d2}(t)$, avec $y_{d1}(t) = \sin(t)$ et $y_{d2}(t) = \cos(t)$.

IV.2.4. Résultats de simulation

Les simulations qu'on a fait sur le modèle dynamique du robot manipulateur ont été réalisées avec les surfaces de glissements $s_i = \dot{e}_i + \lambda_i e_i$ pour $\lambda_i = 5$, $i = 1, 2$, et les gains de commutation $k_i = 20$, $i = 1, 2$ pour la commande discontinue.

Pour résoudre les équations différentielles, nous avons utilisé l'algorithme de Runge-kutta d'ordre 4 avec un pas de simulation $\Delta t = 0.001 \text{sec}$, et un temps final $t_f = 15 \text{sec}$.

Les paramètres du robot manipulateur utilisés en simulation sont donnés comme suit :

$m_1 = 1$, $m_e = 2$, $l_1 = 1$, $l_{c1} = 0.5$, $l_{ce} = 0.6$, $I_1 = 0.12$, $I_e = 0.25$, $\delta_e = 30^\circ$. L'état initial du robot est $x_0 = [-0.5 \ 0 \ 0.25 \ 0]^T$.

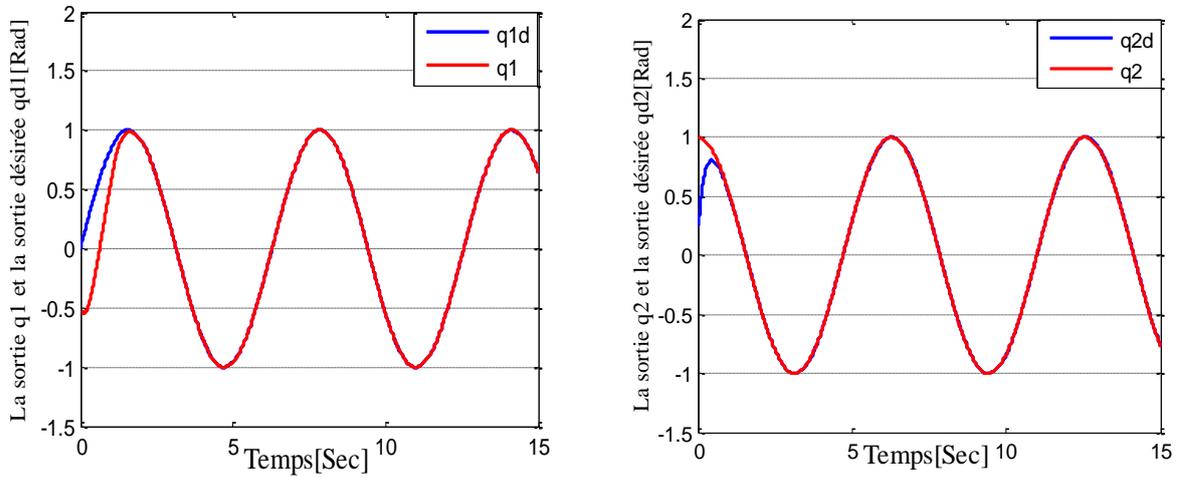


Figure.IV.4 : Position articulaires et position désirées.

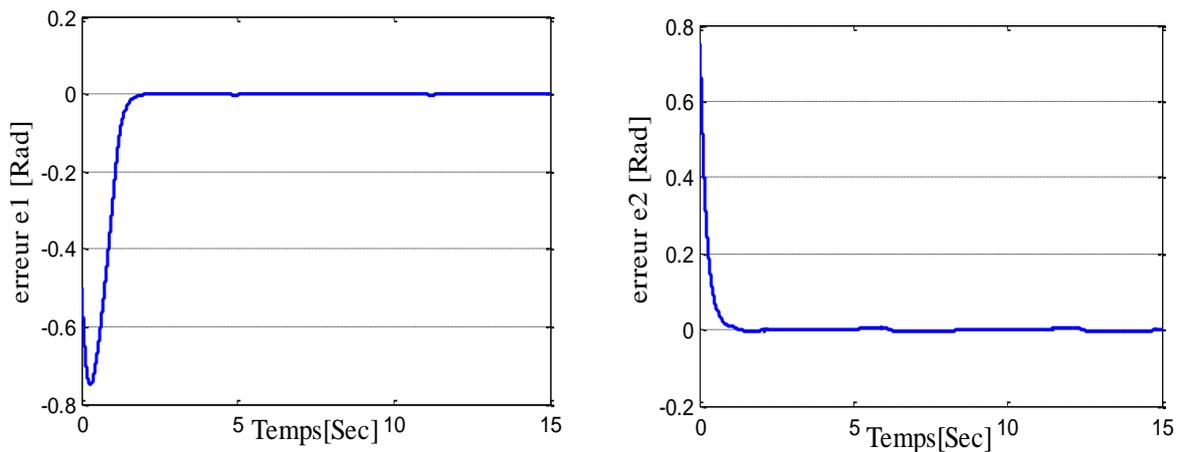


Figure.IV.5 : Les erreurs de poursuite.

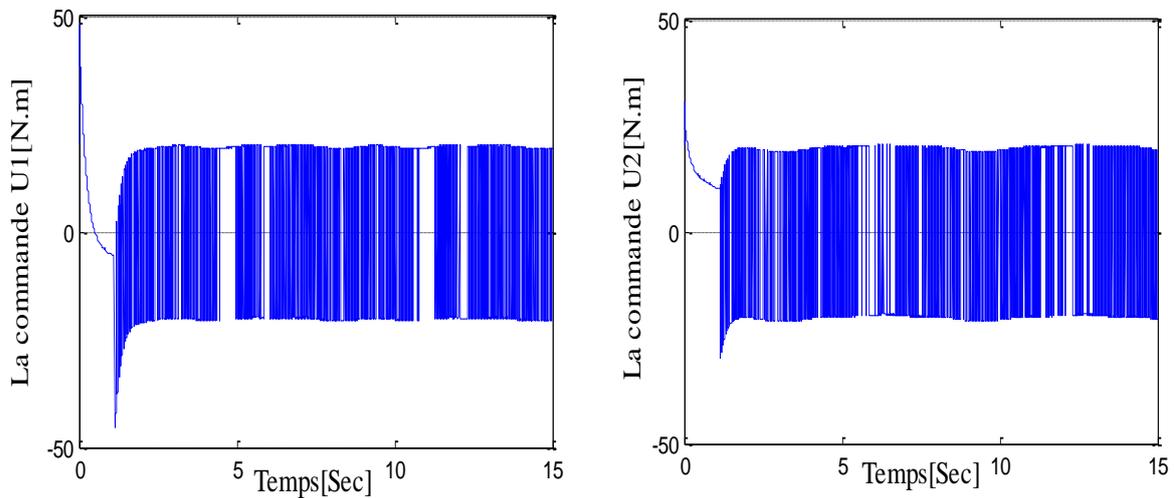


Figure.IV.6 : Les commandes appliquées au robot manipulateur.

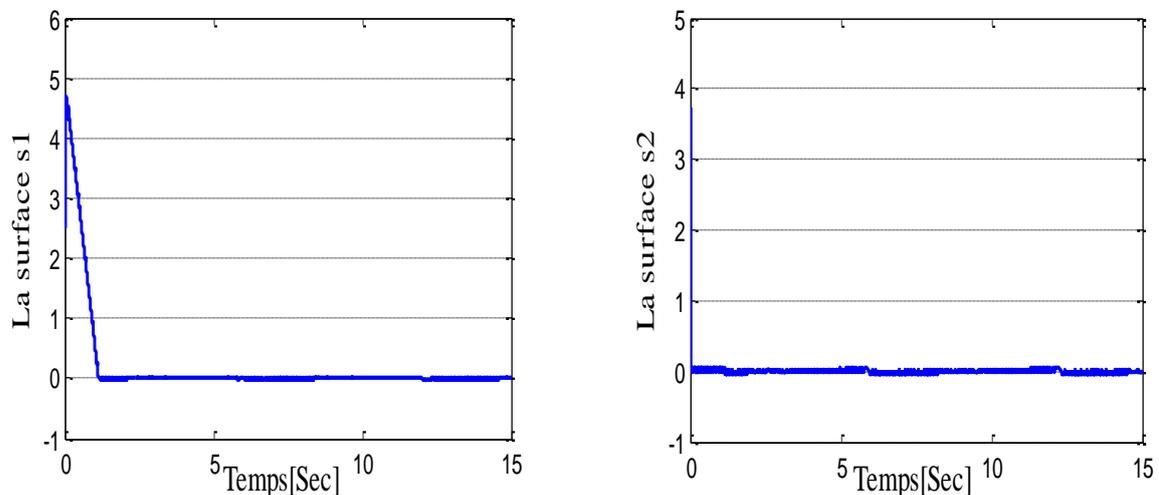


Figure.IV.7 : Les surfaces de glissement.

Les résultats de simulation obtenus par l'application de la commande par mode de glissement sur le robot sont illustrés par les figures IV.4-IV.7. Ceux-ci représentent l'évolution des signaux caractérisant le système en fonction de temps, à savoir : les erreurs, les sorties et les sorties désirés ainsi que la commande appliquée au système.

IV.2.4.1. Interprétation des résultats de simulation

Dans la partie précédente, nous avons présenté une loi de commande par mode de glissement et ceci pour commander en position le robot manipulateur de deux degrés de liberté. Les résultats de simulation illustrés par les figures IV.4-IV.7, permettent de mettre en évidence les avantages dans telle commande.

Nous remarquons une convergence asymptotique des positions articulaires vers celles désirées. Il est clair que les erreurs de poursuite tendent vers zéro. Ces erreurs sont tolérables, par exemple, le temps de réponse est de 1 sec dans la première articulation et environ de 0.6 sec

dans la deuxième articulation. De plus, nous voyons une convergence aussi asymptotique pour les surfaces de glissement vers l'origine. Les signaux de commandes sont lissés et physiquement acceptable.

IV.3. Commande backstepping

IV.3.1. Généralités sur la commande backstepping

La technique de backstepping a été développée au début des années 90. L'arrivée de la commande par backstepping a donné un nouveau souffle à la commande des systèmes non linéaires, qui malgré les grands progrès réalisés, il manquait des approches générales. Cette technique est une méthode systématique et récursive de synthèse de lois de commande non linéaires qui utilise le principe de stabilité de Lyapunov et qui peut s'appliquer à un grand nombre de systèmes non linéaires.

IV.3.2. Principe de la commande backstepping

L'idée de base de la commande de type Backstepping est de rendre les systèmes bouclés équivalents à des sous-systèmes d'ordre un en cascade stable au sens de Lyapunov, ce qui leur confère des qualités de robustesse et une stabilité globale asymptotique. En d'autres termes, c'est une méthode multi étapes. À chaque étape du processus, une commande virtuelle est ainsi générée pour assurer la convergence du système vers son état d'équilibre. Cela peut être atteint à partir des fonctions de Lyapunov qui assurent pas à pas la stabilisation de chaque étape de synthèse [35].

Cette technique s'applique aux systèmes triangulaires inférieurs, appelés aussi systèmes à boucle de retour stricte (strict feedback). Ces systèmes possèdent la spécificité de décrire la dérivée de chaque composante du vecteur d'état.

IV.3.3. Fonctions de Lyapunov

Pour l'étude de la stabilité non linéaire, la méthode la plus classique est basée sur la linéarisation et l'utilisation des valeurs propres du système linéarisé, Lyapunov a proposé une seconde méthode, en s'inspirant de l'idée de l'énergie mécanique de Lagrange qui a formulé le principe de stabilité des systèmes mécaniques « un système qui est dans un état où son énergie potentielle possède un minimum isolé est dans état d'équilibre stable ».

Cette méthode, appelée aussi méthode directe de Lyapunov, est basée sur la recherche d'une fonction scalaire de signe défini à valeurs réelles. Quand sa dérivée par rapport au temps est définie de signe opposé, la vitesse du point x ($x \in \mathbb{R}^n$) est toujours dirigée vers l'intérieur, ce point finira par arriver à l'origine ; dans le cas contraire, le point x s'en écartera d'avantage. Dans quelques classes de systèmes physiques, la fonction V peut être choisie comme étant l'énergie du système.

IV.3.4. Commande backstepping appliquée à un robot manipulateur

Dans cette partie, nous prenons le modèle du robot manipulateur (IV.13) pour la conception de la commande par backstepping, sachant que le nombre d'étapes de construire la loi de commande est dépendre de degré relatif du système à commander.

IV.3.4.1. Synthèses des lois de commandes pour le robot manipulateur

Notre objectif est de commander en position ce modèle du robot manipulateur:

- En premier articulation (selon l'angle q_1).
- En deuxième articulation (selon l'angle q_2).

• Commande du premier sous-système

À partir du modèle du système (IV.11), la dynamique du système suivant l'angle q_1 est donnée par :

$$\begin{cases} \dot{x}_1 = x_2 \\ \dot{x}_2 = f_1(x) + g_{11}(x)u_1 + g_{12}(x)u_2 \\ y_1 = x_1 \end{cases} \quad (\text{IV.34})$$

Donc, on doit synthétiser la loi de commande par les deux étapes suivantes :

Étape 1

On considère le premier sous-système :

$$\dot{x}_1 = x_2 \quad (\text{IV.35})$$

On définit l'erreur e_1 entre x_1 et x_{1d} comme suit :

$$e_1 = x_1 - x_{1d} \quad (\text{IV.36})$$

La dérivée de l'erreur e_1 décrit par :

$$\dot{e}_1 = x_2 - \dot{x}_{1d} \quad (\text{IV.37})$$

On choisit la première fonction de Lyapunov de la forme :

$$V(e_1) = \frac{1}{2} e_1^2 \quad (\text{IV.38})$$

La dérivée de la fonction de Lyapunov est :

$$\dot{V}_1(e_1) = e_1 \dot{e}_1 = e_1(x_2 - \dot{x}_{1d}) \quad (\text{IV.39})$$

Pour assurer la stabilité de e_1 , il faut que $\dot{V}_1(e_1) \leq 0$, pour cela on choisit comme commande virtuel \tilde{x}_2 avec :

$$\tilde{x}_2 = -k_1 e_1 + \dot{x}_{1d}, \quad k_1 > 0 \quad (\text{IV.40})$$

Ce choix permet d'obtenir :

$$\dot{V}_1(e_1) = -k_1 e_1^2 \quad (\text{IV.41})$$

Étape 2

Comme la commande virtuelle ne peut pas prendre instantanément sa valeur désirée, on sert à stabiliser l'erreur entre la commande virtuelle et la fonction stabilisante. Le deuxième sous système est défini par

$$\dot{x}_2 = f_1(x) + g_{11}(x)u_1 + g_{12}(x)u_2 \quad (\text{IV.42})$$

La nouvelle variable d'erreur e_2 , entre la commande virtuelle et la fonction stabilisante, est donnée par

$$e_2 = x_2 - x_{2d} \quad (\text{IV.43})$$

$$e_2 = x_2 - \dot{x}_{1d} + k_1 e_1 \quad (\text{IV.44})$$

En utilisant (IV.44) et (IV.34), sa dérivée est

$$\dot{e}_2 = \dot{x}_2 - \ddot{x}_{1d} + k_1 \dot{e}_1 \quad (\text{IV.45})$$

En utilisant (IV.37) et (IV.45), on trouve

$$\begin{cases} \dot{e}_1 = e_2 - k_1 e_1 \\ \dot{e}_2 = \dot{x}_2 - \ddot{x}_{1d} + k_1 (e_2 - k_1 e_1) \end{cases} \quad (\text{IV.46})$$

La nouvelle fonction de Lyapunov du système augmenté est définie comme suit

$$V(e_1, e_2) = V_1 + \frac{1}{2} e_2^2 = \frac{1}{2} e_1^2 + \frac{1}{2} e_2^2 \quad (\text{IV.47})$$

Sa dérivée est donnée par

$$\dot{V}(e_1, e_2) = e_1 \dot{e}_1 + e_2 \dot{e}_2 \quad (\text{IV.48})$$

On remplace \dot{e}_1 et \dot{e}_2 dans (IV.48), on aura

$$\dot{V}(e_1, e_2) = e_1 (e_2 - k_1 e_1) + e_2 (\dot{x}_2 - \ddot{x}_{1d} + k_1 (e_2 - k_1 e_1)) \quad (\text{IV.49})$$

Le choix de la loi de commande dépend de la dérivée négative de la fonction de Lyapunov, $\dot{V}(e_1, e_2) \leq 0$ comme suit

$$\dot{V}(e_1, e_2) = -k_1 e_1^2 - k_2 e_2^2 \quad \text{avec } k_1 \text{ et } k_2 > 0 \quad (\text{IV.50})$$

à partir de (IV.49) et (IV.50), on obtient

$$-k_2 e_2 = e_1 + f_1(x) + g_{11}(x)u_1 + g_{12}(x)u_2 - \ddot{x}_{1d} + k_1(e_2 - k_1 e_1) \quad (\text{IV.51})$$

La commande u_1 qui assure la poursuite est calculée ci-dessus.

• Commande de deuxième sous-système :

À partir du modèle du système (IV.11), la dynamique de système suivant l'angle q_2 est donnée par :

$$\begin{cases} \dot{x}_3 = x_4 \\ \dot{x}_4 = f_2(x) + g_{21}(x)u_1 + g_{22}(x)u_2 \\ y_2 = x_3 \end{cases} \quad (\text{IV.52})$$

La synthèse de la commande u_2 se fait en deux étapes

Étape 1

On a la première sous système

$$\dot{x}_3 = x_4 \quad (\text{IV.53})$$

On définit l'erreur e_3 entre q_1 et q_{1d} par

$$e_3 = x_3 - x_{3d} \quad (\text{IV.54})$$

La dérivée de l'erreur e_3 est

$$\dot{e}_3 = x_4 - \dot{x}_{3d} \quad (\text{IV.55})$$

Ici, notre objectif est que l'erreur e_3 doit converger vers zéro, pour cela on choisit la première fonction de Lyapunov comme suit

$$V_3(e_3) = \frac{1}{2} e_3^2 \quad (\text{IV.56})$$

En utilisant (IV.55), la dérivation de cette fonction par rapport au temps nous donne

$$\dot{V}(e_3) = e_3 \dot{e}_3 = e_3(x_4 - \dot{x}_{3d}) \quad (\text{IV.57})$$

pour assurer la stabilité de e_3 il faut que $\dot{V}(e_3) \leq 0$, pour cela on prend comme commande virtuelle \tilde{x}_4 avec

$$\tilde{x}_4 = -k_3 e_3 + \dot{x}_{3d}, \quad k_3 > 0 \quad (\text{IV.58})$$

ce choix permet d'obtenir

$$\dot{V}(e_3) = -k_3 e_3^2 \quad (\text{IV.59})$$

Étape 2

Le deuxième sous-système est donnée par

$$\dot{x}_4 = f_2(x) + g_{21}(x)u_1 + g_{22}(x)u_2 \quad (\text{IV.60})$$

La nouvelle variable d'erreur est choisie comme suit

$$e_4 = x_4 - \tilde{x}_4 \quad (\text{IV.61})$$

$$e_4 = x_4 - \dot{x}_{3d} + k_3 e_3 \quad (\text{IV.62})$$

Sa dérivé est donnée par

$$\dot{e}_4 = \dot{x}_4 - \ddot{x}_{3d} + k_3 \dot{e}_3 \quad (\text{IV.63})$$

A partir de (IV.55) , (IV.58) et (IV.61) on trouve

$$\begin{cases} \dot{e}_3 = e_4 - k_3 e_3 \\ \dot{e}_4 = \dot{x}_4 - \ddot{x}_{3d} + k_3 (e_4 - k_3 e_3) \end{cases} \quad (\text{IV.64})$$

La nouvelle fonction de lyapunov est définie comme suit

$$V(e_3, e_4) = \frac{1}{2} e_3^2 + \frac{1}{2} e_4^2 \quad (\text{IV.65})$$

sa dérivée sera

$$\dot{V}(e_3, e_4) = e_3 \dot{e}_3 + e_4 \dot{e}_4 \quad (\text{IV.66})$$

on remplace \dot{e}_3 et \dot{e}_4 dans (IV.65), on trouve

$$\dot{V}(e_3, e_4) = e_3 (e_4 - k_3) + e_4 (\dot{x}_4 - \ddot{x}_{3d} + k_3 (e_4 - k_3 e_3)) \quad (\text{IV.67})$$

Le choix de la loi de commande doit mener à une dérivée négative de la fonction de lyapunov

$\dot{V}(e_3, e_4) \leq 0$, comme suit

$$\dot{V}(e_3, e_4) = -k_3 \dot{e}_3 - k_4 \dot{e}_4 \quad \text{avec } k_3 \text{ et } k_4 > 0 \quad (\text{IV.68})$$

Employant l'équation (IV.66), on trouve que

$$-k_4 e_4 = e_3 + f_2(x) + g_{21}(x)u_1 + g_{22}(x)u_2 - \ddot{x}_{3d} + k_3 (e_4 - k_3 e_3) \quad (\text{IV.69})$$

À partir de (IV.51) et (IV.69), nous obtenons la loi de commande u suivante

$$u = G^{-1}(x) \left[\begin{array}{c} \ddot{x}_{1d} - k_2 e_2 - e_1 - k_1 (e_2 - k_1 e_1) \\ \ddot{x}_{3d} - k_4 e_4 - e_3 - k_3 (e_4 - k_3 e_3) \end{array} \right] - \left[\begin{array}{c} f_1(x) \\ f_2(x) \end{array} \right] \text{ avec } u = [u_1, u_2]^T \quad (\text{IV.70})$$

L'objectif de cette commande est de forcer les deux articulations de robot manipulateur à suivre les deux consignes $y_{d1}(t)$ et $y_{d2}(t)$, avec $y_{d1}(t) = \sin(t)$ et $y_{d2}(t) = \cos(t)$.

IV.3.5. Résultats de simulation

Dans cette partie, on va présenter les résultats de simulation résultent de l'application de la commande backstepping sur le robot manipulateur. Les paramètres de simulations sont :

$k_1 = 10$ et $k_2 = 12, k_3 = 10$ et $k_4 = 12$, avec l'état initiale $X_0 = [-0.5, 0, 0.25, 0]$.

Pour résoudre les équations différentielles, nous utilisons l'algorithme de Runge-kutta d'ordre 4 avec un pas de simulation $\Delta t = 0.001 \text{sec}$ et un temps final $t_f = 15 \text{sec}$.

Les figures (IV.8-IV.9-IV.10) : Ceux-ci représentent l'évolution des signaux caractérisant le système en fonction de temps, à savoir : les erreurs, les sorties et les sorties désirés ainsi que la commande appliquées au système.

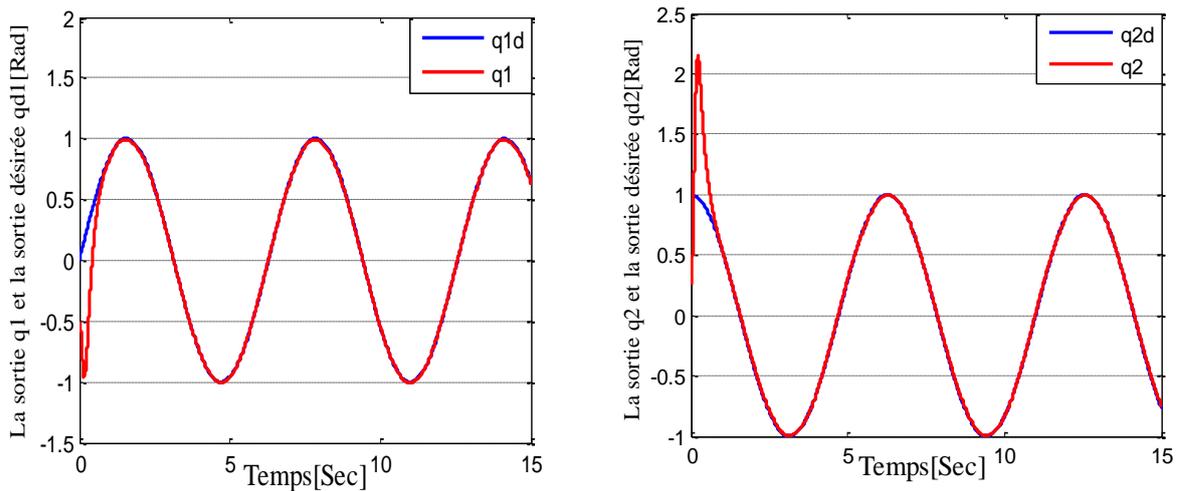


Figure.VI.8 : Position articulaire et position désirées.

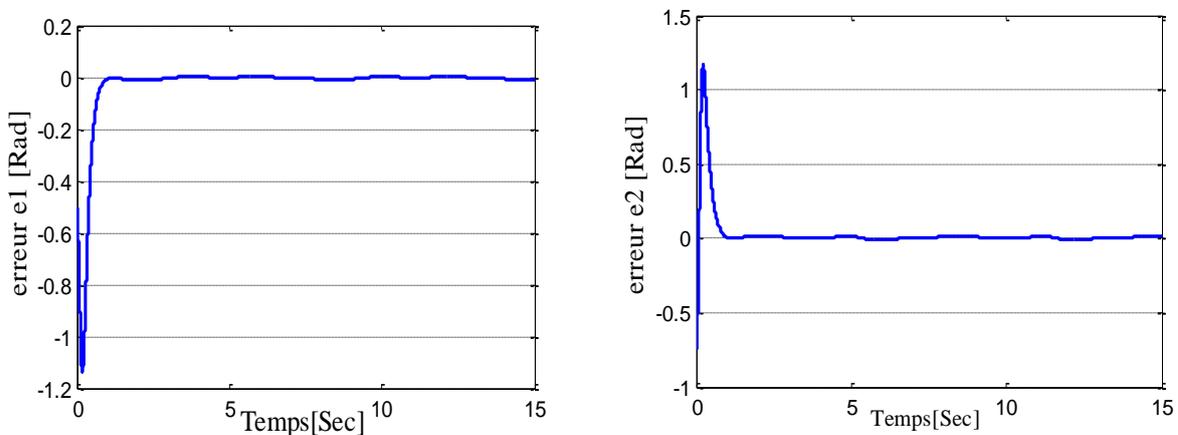


Figure.IV.9 : Les erreurs de poursuite.

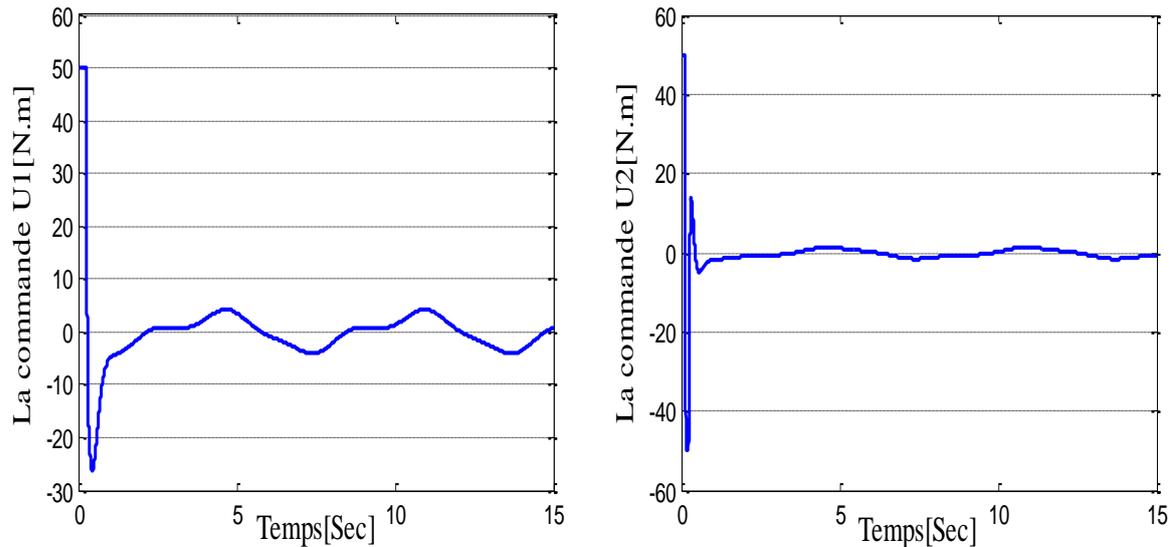


Figure.IV.10 : Les commandes appliquées au bras manipulateur.

IV.3.6. Interprétation des résultats de simulation

Dans la partie précédent, nous avons présenté une loi de commande backstepping et ceci pour commander en position le robot manipulateur de deux degrés de liberté. Les résultats de simulations illustrées par les figures ci-dessous. Nous remarquons une convergence asymptotique des positions articulaires vers celle désirées. Il est clair que les erreurs de poursuite tendent vers zéro. Les signaux de commandes sont lissent et physiquement acceptable.

IV.4. Étude comparative

Pour examiner les différents lois de commande, développées pour le robot manipulateur, nous avons opté pour une étude comparative entre ces différentes commandes. La commande par mode de glissement, la commande backstepping et la commande L1 adaptative floue. Pour cela, nous définissons deux critères, le premier est en fonction de la commande appliquée (considère comme un critère énergétique), et le deuxième est en fonction de l'erreur résultante. Pour bien examiner ces commandes, les résultats ont été pris dans les mêmes conditions, par exemple : la période d'échantillonnage, le temps de simulation, les gains de la commande. Ces résultats sont effectuée avec et sans test de robustesse.

- **Sans test de robustesse :**

critère	Commande développées pour le robot manipulateur		
	Commande par mode de glissement	Commande Backstepping	Commande L1 adaptative
$J_1 = \sum_{k=1}^{p=15000} (u^T \cdot u)$	1.1662 .10 ⁷	1.5341 .10 ⁶	6.6237 .10 ⁵
$J_2 = \sum_{k=1}^{p=15000} (e^T \cdot e)$	454.0530	632.2000	179.4049

Tableau .IV.1 : Étude comparative entre les commandes développées pour le robot manipulateur sans test de robustesse.

- **Avec test de robustesse :**

Critère	Commande développées pour le robot manipulateur		
	Commande par mode de glissement	Commande Backstepping	Commande L1 adaptative
$J_1 = \sum_{k=1}^{p=15000} (u^T \cdot u)$	1.1693 .10 ⁷	1.5524 .10 ⁶	6.8222 .10 ⁵
$J_2 = \sum_{k=1}^{p=15000} (e^T \cdot e)$	448.3546	631.4930	180.9337

Tableau .IV.2 : Étude comparative entre les commandes développées pour le robot manipulateur avec test de robustesse.

Dans les deux tableaux précédents, nous avons présenté une comparaison quantitative entre les différentes approches développées dans ce document. Pour une comparaison très claire, on s'est intéressé à l'évolution temporelle de la commande et de l'erreur résultante (comparaison qualitative).

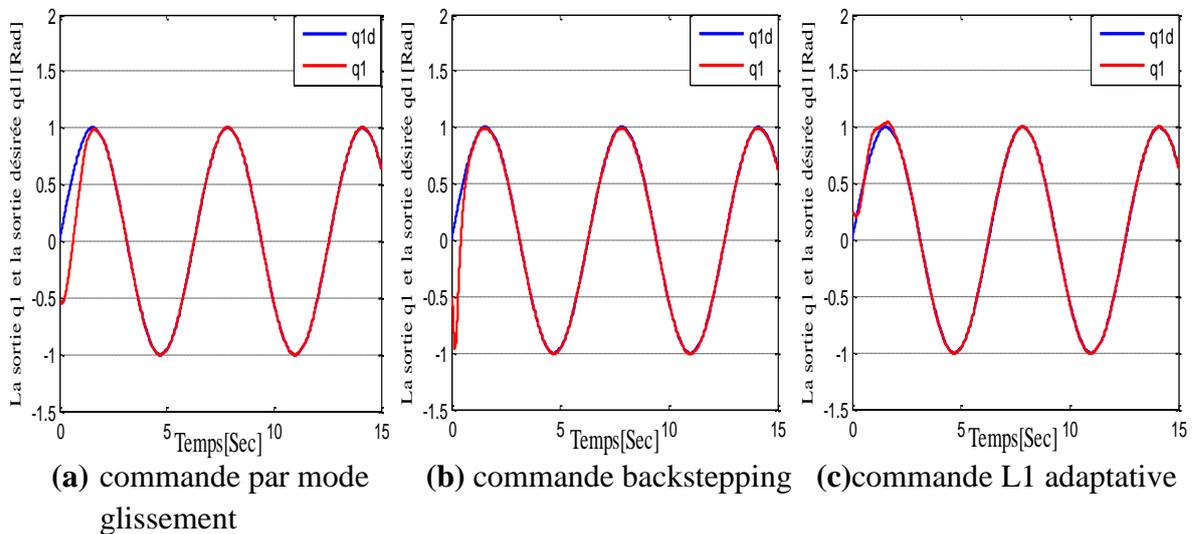


Figure.VI.11 : Position articulaires et position désirées (selon l'angle q_1).

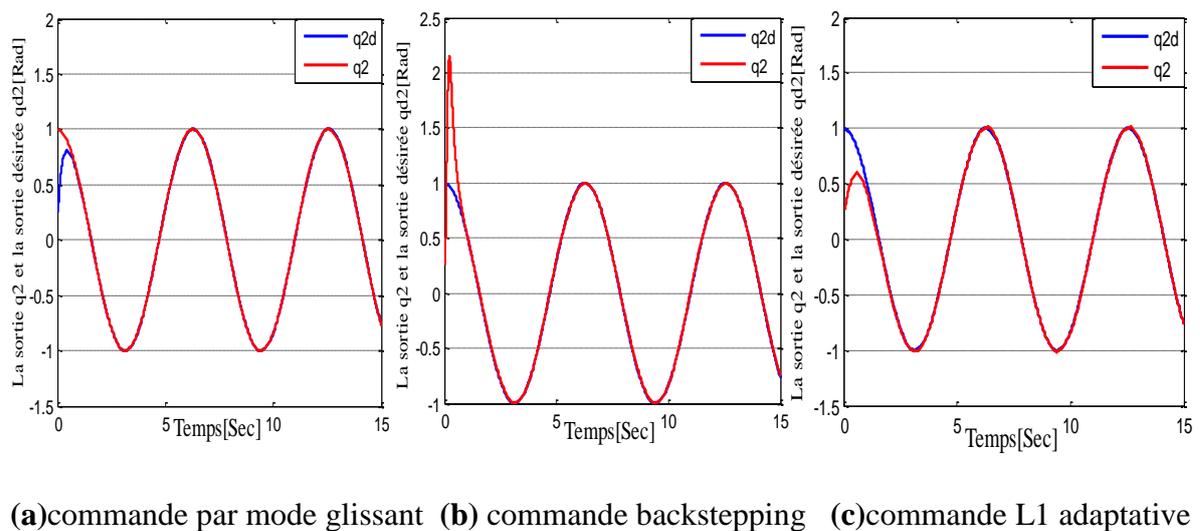
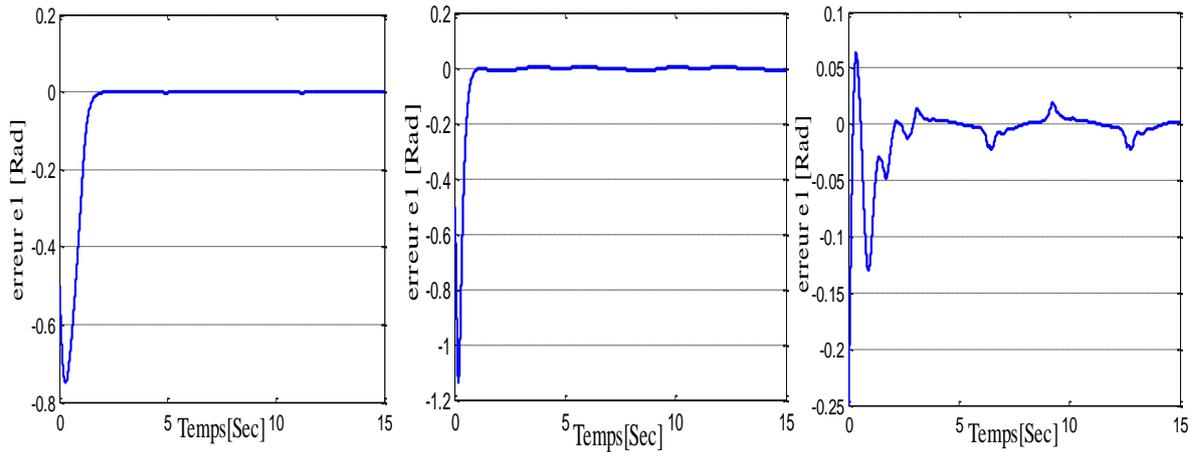


Figure.VI.12 : Position articulaires et position désirées (selon l'angle q_2).



(a) commande par mode glissement (b) commande backstepping (c) commande L1 adaptative

Figure.VI.13 : Les erreurs de poursuite.

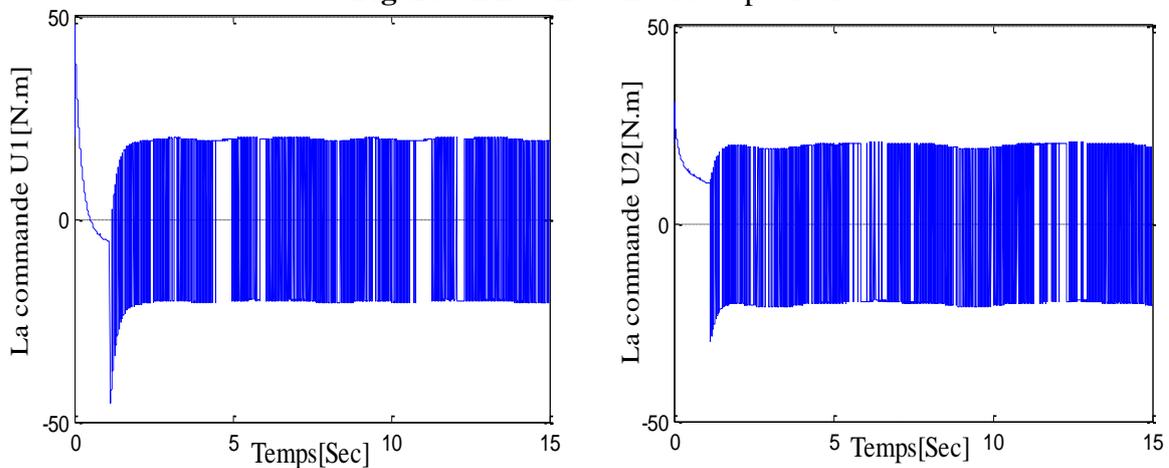


Figure.IV.14 : Les commandes appliquées au robot manipulateur : commande par mode de glissement.

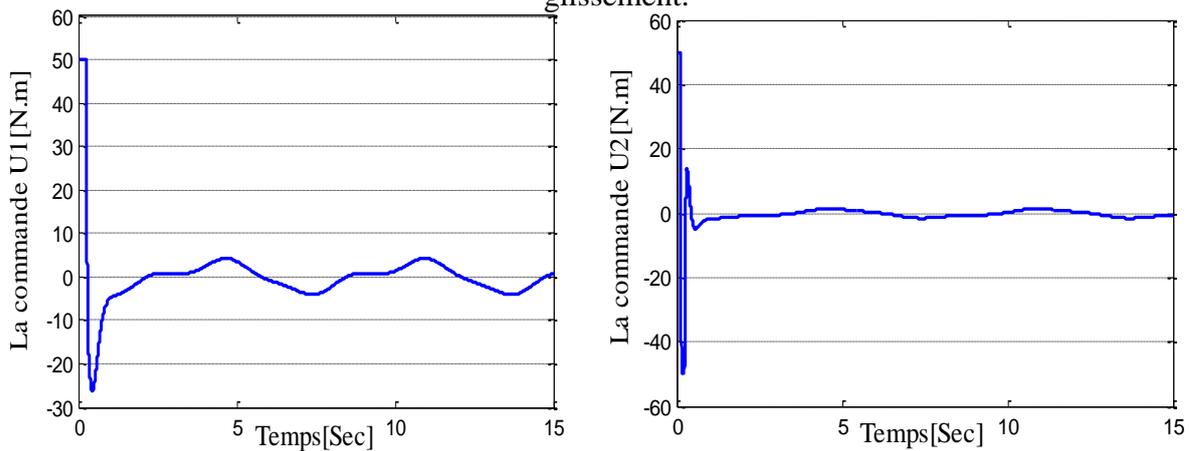


Figure.IV.15 : Les commandes appliquées au robot manipulateur : commande Backstepping.

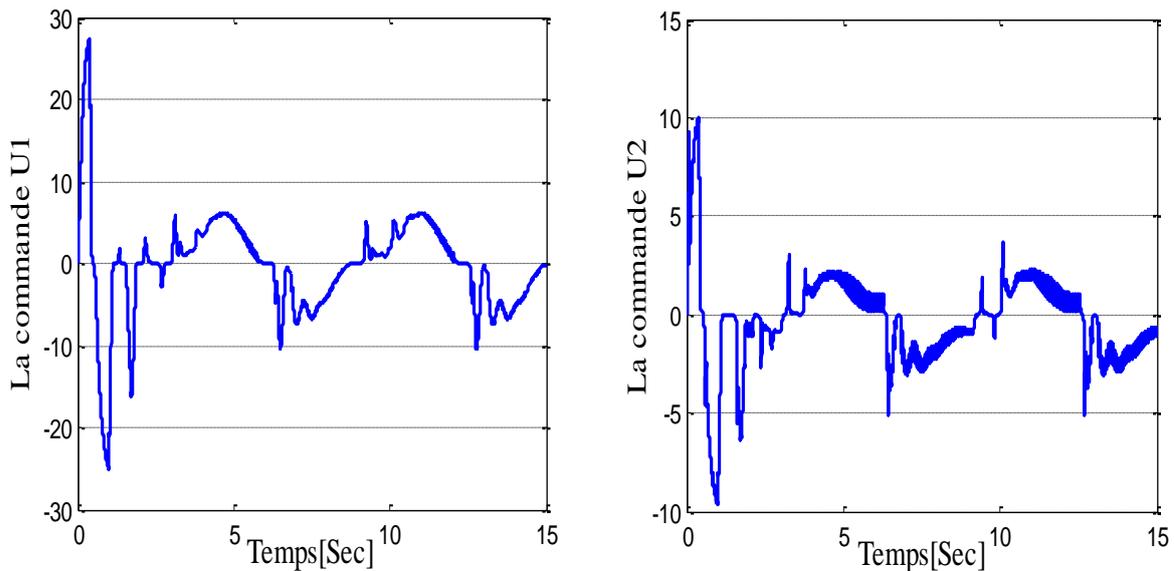


Figure.IV.16 : Les commandes appliquées au robot manipulateur : commande L1 adaptative.

L'objectif de ce chapitre est d'examiner les différentes lois de commandes synthétisées et ceci dans les mêmes conditions. Cette comparaison s'est faite selon deux voies : quantitative et qualitative. Pour mettre en évidence les performances de chaque loi de commande, en termes de l'erreur résultante et de la valeur de la commande nécessaire pour réaliser une telle commande, nous avons adopté les deux critères J_1 et J_2 (Tableau IV.1 et Tableau IV.2). Ensuite nous sommes intéressés à la variation temporelle caractérisant ces deux grandeurs.

Selon le Tableau IV.1, on remarque que pour la commande L1 adaptative flou, le critère J_1 prend la valeur le plus basse ($J_1 = 6.6237 \cdot 10^5$). De même, le critère J_2 prend une bonne valeur pour la commande L1 adaptative flou ($J_2 = 179.4049$) par rapport à celles de la commande par mode de glissement et la commande backstepping.

Selon le Tableau IV.2, on remarque que pour la commande L1 adaptative flou, le critère J_1 prend la valeur le plus basse ($J_1 = 6.8222 \cdot 10^5$). De même, le critère J_2 prend aussi une petite valeur pour la commande L1 adaptative flou (180.9337) par rapport à celles des deux autres commandes.

IV.5. Conclusion

La commande par mode de glissement et la commande backstepping sollicitent une énergie plus grande et importante, cette commande est de hautes fréquences, ce qui est considéré comme un inconvénient néfaste. Le calcul de la commande nécessite la connaissance du système à commander, ce qui est considéré comme une limitation à cette commande tandis que la commande L1 adaptative floue ne nécessite pas la connaissance du système.

On constate que la commande L1 adaptative floue est en général la plus performante par rapport aux autres commandes.



CONCLUSION GÉNÉRALE

Conclusion générale

Comme son nom l'indique on parle de commande adaptative lorsque certains paramètres d'un système sont réadaptés au cours du temps. Cette adaptation est souhaitable pour deux raisons principales. Premièrement, soit pour affiner, au cours du temps, le réglage initial du système de commande. Deuxièmement, soit pour tenir compte des variations des paramètres du système à commander.

Dans ce projet de fin d'étude, notre objectif c'était de présenter et d'étudier une commande L1 adaptative floue pour une classe des systèmes MIMO affines en entrée, avec une étude en simulation des performances de cette commande sur un robot manipulateur.

Nous avons établi un contrôleur L1 adaptatif flou en utilisant un système flou de Takagi-Sugeno. Les résultats de simulation confirment les performances de ce contrôleur. La commande proposée n'exige pas la connaissance du modèle mathématique du système à commander, nous pouvons la considérer comme une commande sans modèle robuste.

Dans le premier chapitre nous avons présenté des généralités sur la logique floue et des notions de base sur les systèmes flous.

Dans le deuxième chapitre, on a parlé de deux techniques de commande adaptative, la commande adaptative à modèle de référence et la commande L1 adaptative. Afin d'étudier leurs performances, nous avons fait une simulation sur un système scalaire. L'étude de l'effet du gain d'adaptation sur la marge de gain et la marge de phase a montré les avantages de la commande L1 adaptative par rapport à la commande MRAC.

Dans le troisième chapitre, on a introduit une architecture de commande L1 adaptative floue pour une classe des systèmes non linéaires MIMO affine en entrée. La simulation de la commande sur le modèle d'un robot manipulateur a donné des résultats très satisfaisants.

Dans le dernier chapitre, nous avons fait une étude comparative quantitative et qualitative entre la commande L1 adaptative floue proposée et deux commandes classiques qui sont : la commande par mode de glissement et la commande backstepping, les résultats de cette étude montrent bien les performances de la commande L1 adaptative floue proposée par rapport à celles des deux commandes classiques.

Comme perspectives pour ce travail, il serait intéressant d'implémenter en temps réel la commande L1 adaptative floue présentée dans ce travail de mémoire ainsi que le développement des observateurs pour reconstruire le vecteur d'état du système à commander.

Références bibliographiques

- [1] J.E. Slotine and W. Li. Applied non linear control. Prentice Hall, Englewood Cliffs, NJ, 1991.
- [2] Hovakimyan N and Cao C. Adaptive control theory: guaranteed robustness with fast adaptation. Philadelphia, PA: Siam - Society for Industrial and Applied Mathematics, 2010.
- [3] Ioannou PA and Sun J. Robust adaptive control. Englewood Cliffs, NJ: Prentice Hall, 1996.
- [4] Cao C and Hovakimyan N. Design and analysis of a novel L1 adaptive controller, Part I: control signal and asymptotic stability. In: American control conference, Minneapolis, MN, USA, 14–16 June 2006, pp.3397–3402. New York: IEEE.
- [5] Cao C and Hovakimyan N. Design and analysis of a novel L1 adaptive control architecture, Part II: guaranteed transient performance. In: American control conference, Minneapolis, MN, USA, 14–16 June 2006, pp.3403–3408. New York: IEEE.
- [6] Cao C and Hovakimyan N. Design and analysis of a novel L1 adaptive control architecture with guaranteed transient performance. IEEE Trans Autom Control 2008; 53(2): 586–591.
- [7] Cao C and Hovakimyan N. Adaptive controller for systems with unknown time-varying parameters and disturbances in the presence of non-zero trajectory initialization error. Int J Control 2008; 81(7): 1147–1161.
- [8] Cao C and Hovakimyan N. L1 adaptive output feedback controller for systems of unknown dimension. IEEE Trans Autom Control 2008; 53(3): 815–821.
- [9] Cao C and Hovakimyan N. Stability margins of L1 adaptive control architecture. IEEE Trans Autom Control 2010; 55(2): 480–487.
- [10] Kharisov E, Hovakimyan N and Astrom KJ. Comparison of architectures and robustness of model reference adaptive controllers and L1 adaptive controllers. Int J Adapt Control Signal Process 2014; 28(7–8): 633–663.
- [11] Hashim HA, El-Ferik S and Abido MA. A fuzzy logic feedback filter design tuned with PSO for adaptive controller. Expert Syst Appl 2015; 42(23): 9077–9085.
- [12] Wang LX. Adaptive fuzzy systems and control: design and stability analysis. Englewood Cliffs, NJ: Prentice Hall, 1994.
- [13] Labiod S and Guerra TM. Direct adaptive fuzzy control for a class of MIMO nonlinear systems. Int J Syst Sci 2007; 38(8): 665–675.
- [14] Cooper J, Che J and Cao C. The use of learning in fast adaptation algorithms. Int J Adapt Control Signal Process 2014; 28(3–5): 325–340.
- [15] Mokeddem D. Contrôle flou des processus biotechnologiques à base d’algorithmes génétiques. Thèse de Doctorat. Université Ferhat Abbas De Setif 2010.

-
- [16] Chevie F and Guély F. Cahier technique. Groupe schneider Edition 1998.
- [17] Ibrahim A, M. Fuzzy logic for embedded systems applications, Elsevier Science, 2004.
- [18] Jantzen J, Foundations of fuzzy control, John Wiley and Sons Ltd, 2007.
- [19] Bencherif F.N Evaluation du risque d'accident vasculaire cérébral à l'aide de la logique floue. Mémoire de Master. Université Abou Bakr Belkaid– Tlemcen 2014.
- [20] Wang L.X, Adaptive fuzzy systems and control : Design and stability Analysis, Englewood Cliffs, NJ : Prentice Hall, 1994.
- [21] Passino K.M and Yurkovich S, Fuzzy control. Addison-Wesley, 1998.
- [22] Ben Abbas O et Louzzani Z. Commande adaptative Floue des systèmes non linéaires et application. Mémoire de Master. Université de Jijel. 2010.
- [23] Amieur T. Commande des Systèmes Non Linéaires par Mode Glissant Flou. PFE. Université de Biskra 2009.
- [24] Cao C and Hovakimyan N. L1 adaptive control theory guaranteed robustness with fast adaptation. University of Illinois ; Urbana, Illinois, et university of connecticut storrs, Connecticut. ISBN 978-0-8987 17-04.
- [25] Graidia S. Commande adaptative floue type-2 par mode de glissant des systèmes chaotiques. Mémoire de Master. Université Kasdi Merbah, Ouergla 2012.
- [26] Khettab K. Techniques avancées pour la synthèse et l'amélioration des performances des systèmes de commande adaptative d'ordre fractionnaire pour les processus incertains. Thèse de doctorat. Université de Skikda 2016.
- [27] Bensafia Y. Utilisation des filtres fractionnaires pour la conception de régulateurs adaptatifs robustes. Thèse de doctorat. Université de Skikda. 2016
- [28] Laroussi K. Développement d'un superviseur flou appliqué à la commande d'un moteur à induction. Thèse de doctorat. Université M'hamed Bougara Boumerdes 2014.
- [29] Zemalache Meguenni K. Commande d'un système sous-actionné : Application à un drone à Quatre Hélices. Thèse de doctorat. Université de d'Evry Val d'Essonne. 2006.
- [30] Selami M. Commande adaptative floue robuste par mode de glissant pour les systèmes non linéaires incertains via backstepping. Mémoire de Master. Université Kasdi Merbah Biskra 2017.
- [31] Boubakir A, Labiod S et Budjema F. A stable self-tuning proportional-integral-derivative controller for a class of multi-input multi-output nonlinear systems. Journal of Vibration and Control 18: 228 originally published online 1 July 2011. 2012.
- [32] Boubakir A, Souanef T, Labiod S, Plestan F et Boudjema F. An L1 fuzzy adaptive controller for a class of SISO nonaffine nonlinear systems: application to the control of an electropneumatic actuator. J Systems and Control Engineering. 2016, Vol. 230(8) 736–748.
- [33] Deghboudj I. Commande des systèmes non linéaires par mode glissant d'ordre supérieur. Mémoire de magister. Université Constantine 1. 2013.
- [34] Abderrezek H. Contribution à la commande robuste des systèmes non linéaires. Mémoire de magister. Université Ferhat Abbas-Setif-1. 2014.

- [35] Ezzat M. Commande non linéaire sans capteur de la machine synchrone à aimants permanents. Thèse de doctorat. École Centrale de Nantes (ECN). 2011.