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

Université Mohammed Seddik Ben yahia

Faculté des Sciences et de la Technologie
Département d'Automatique

جامعة محمد الصديق بن يحيى
قسم الآليات
كلية العلوم والتكنولوجيا

Projet de fin d'étude

MR 518

Option : Automatique et traitement du signal

Thème

Commande adaptative floue des systèmes non linéaires avec saturation

Proposé par :
Mme ZIBRA Aicha

Réalisés par :
BOUCHOUKH Ahmed Ridha
SOUYAD Nour el-houda

2015/2016

Remerciements

Nous tenons à remercier Md Zibra Aicha pour la proposition du sujet de ce mémoire, ses conseils et ses qualités scientifiques ainsi que tous nos enseignants qui nous suivi tout au long de nos études.

Nos remerciements vont aux membres de jury pour nous avoir honorés en acceptant de juger notre travail.

Nos vifs remerciements à nos familles qui nous ont témoignés un soutien moral et matériel inégalable lors de la réalisation de ce travail.

DEDICAS

Je dédie mon travail :

A ma très chère famille,

A mes amies,

Au symbole de tolérance mon binôme Ridha,

A tous que j'ai la chance

De les rencontrer,

Et enfin je le dédie à toute la promotion

Master 2 automatique 2015/2016

Nour

DEDICAS

Je dédie mon travail :

A ma très chère famille,

A mes amis,

Au symbole de tolérance mon binôme Nour,

A tous que j'ai la chance

De les rencontrer,

Et enfin je le dédie à toute la promotion

Master 2 automatique 2015/2016

Ridha

Contents

1 Les systèmes flous **1**
- 1.1 Introduction . . . 1
- 1.2 Logique floue . . . 2
  - 1.2.1 Définition . . . 2
  - 1.2.2 Quelques avantages de la logique floue . . . 2
  - 1.2.3 Les inconvénients . . . 3
  - 1.2.4 Application de la logique floue . . . 3
- 1.3 Les Concepts de la Logique Floue . . . 3
  - 1.3.1 Variables linguistiques et ensembles flous . . . 3
  - 1.3.2 Sous-ensembles flous . . . 4
  - 1.3.3 Caractéristique d'un sous-ensemble flou . . . 4
  - 1.3.4 Fonction d'appartenance . . . 5
  - 1.3.5 Variable linguistique . . . 6
  - 1.3.6 Opérations de base sur les sous ensembles flous . . . 7
- 1.4 La structure d'un système flou . . . 8
  - 1.4.1 Interface de fuzzification . . . 8
  - 1.4.2 Base de règles floues . . . 8
  - 1.4.3 Moteur d'inférence . . . 8
  - 1.4.4 Interface de Déffuzification . . . 9
- 1.5 Types de système flou . . . 9
  - 1.5.1 Développement du modèle de Takagi-Sugéno . . . 10
- 1.6 Conclusion . . . 11

2 Commande adaptative indirecte floue **13**
- 2.1 Introduction . . . 13
- 2.2 Stabilité au sens de Lyapunov . . . 14
  - 2.2.1 Méthode directe de Lyapunov . . . 14
  - 2.2.2 Théorème de stabilité par la méthode directe . . . 15
  - 2.2.3 Lemme de Barbalat . . . 15
- 2.3 Commande Adaptative . . . 16
- 2.4 Commande adaptative floue . . . 17
  - 2.4.1 Commande adaptative floue directe . . . 18
  - 2.4.2 Commande adaptative floue indirecte . . . 18
- 2.5 Conception de la commande adaptative floue indirecte . . . 19
  - 2.5.1 Position du problème . . . 19
  - 2.5.2 Objectif . . . 20
  - 2.5.3 Calcul de l'erreur . . . 20
  - 2.5.4 Approximation des fonctions non linéaires par les systèmes flous 21
  - 2.5.5 Loi de commande . . . 22

2.5.6 Conception de la loi de commande proposée 23
2.5.7 Démonstration de la stabilité 24
2.6 Résultats de simulation 27
2.7 Conclusion 29

3 Commande adaptative indirecte floue avec saturation 30
3.1 Introduction 30
3.2 La saturation 31
3.3 Position du problème 34
3.3.1 Objectif 35
3.3.2 Calcul de l'erreur 36
3.3.3 Approximation des fonctions non linéaires par les systèmes flous 36
3.3.4 Loi de commande 38
3.3.5 Conception de la loi de commande proposée 39
3.3.6 Démonstration de la stabilité 40
3.4 Résultats de simulation 41
3.4.1 Sans prise en compte de saturation 41
3.4.2 Avec solution de saturation 44
3.4.3 Conclusion 45

4 Commande adaptative directe floue avec saturation 47
4.1 Introduction 47
4.2 Position du problème 48
4.2.1 Objectif 49
4.2.2 Calcule de l'erreur 49
4.2.3 Conception de la loi de commande adaptative directe 50
4.2.4 Démonstration de la stabilité 52
4.3 Résultats de simulation 53
4.3.1 Sans prise en compte de saturation 53
4.3.2 Avec solution de saturation 55
4.4 Conclusion 56

A Annexe A 60
A.1 Théorème : 60

Bibliography 61

List of Figures

1.1 Fonction caractéristique d'appartenance pour la logique booléenne et la logique floue. 4
1.2 Fonction triangulaire 6
1.3 Fonction trapézoïdale 6
1.4 Fonction gaussienne 6
1.5 Fonction sigmoïdale 6
1.6 Structure de base d'un système flou. 8

2.1 Principe de commande adaptative. 16
2.2 Commande adaptative directe floue. 18
2.3 Commande adaptative indirecte floue 18
2.4 Premier état du systeme ; réelle (trait bleu), consigne (trait vert). . . 28
2.5 Deuxième état du systeme ; réelle (trait bleu), consigne (trait vert). . 28
2.6 signal de commande u. 28

3.1 La fonction de saturation 32
3.2 Effets de saturation[25] 32
3.3 Saturation d'amplitude et de vitesse. 33
3.4 Premier état du systeme ; réelle (trait bleu), consigne (trait vert). . . 43
3.5 Deuxième état du systeme ; réelle (trait bleu), consigne (trait vert). . 43
3.6 signal de commande ; u. 43
3.7 Premier état du systeme ; réelle (trait bleu), consigne (trait vert). . . 44
3.8 Deuxièeme état du systeme ; réelle (trait bleu), consigne (trait vert). 45
3.9 signaux de commande ; u (trait bleu),$\sigma(u)$(trait vert). 45

4.1 Premier état du systeme ; réelle (trait bleu), consigne (trait vert). . . 54
4.2 Deuxième état du systeme ; réelle (trait bleu), consigne (trait vert). . 54
4.3 signal de commande ; u. 55
4.4 Premier état du systeme ; réelle (trait bleu), consigne (trait vert). . . 55
4.5 Deuxième état du systeme ; réelle (trait bleu), consigne (trait vert). . 56
4.6 signaux de commande ; u (trait bleu),$\sigma(u)$ (trait vert). 56

Introduction générale

Au cours des deux dernières décennies, un grand nombre de publication a été dédié au problème de commande des systèmes non linéaires. Un problème qui présente beaucoup de chalenges puisque les systèmes non linéaires, contrairement aux systèmes linéaires pour lesquels l'automatique fournit une panoplie de méthodes pour la synthèse de la commande. Ceci est dû au fait que les systèmes non linéaires possèdent des structures extrêmement variées, des dynamiques complexes et peuvent présenter toutes sortes de comportement étrange. A l'inverse des systèmes linéaires, les non linéaires ne possèdent pas des outils ou des méthodes générales directes pour la synthèse des lois de commande.

Au début, la commande des systèmes non linéaires était classique, elle reposait sur la théorie de l'automatique linéaire. En fait, elle consiste à linéariser la dynamique non linéaire du système (approximation du premier ordre) autour d'un point de fonctionnement de telle sorte que les outils de la commande linéaire puissent être exploités pour la synthèse d'une loi de commande assurant les performances recherchés. Grâce à sa facilité d'implantation et de mise en œuvre, cette approche a remporté un grand succès auprès des industriels. Cependant, la nécessité d'atteindre de hautes performances dans des domaines de fonctionnement relativement larges, a imposée la prise en compte de la dynamique globale non linéaire des processus dans la synthèse de la commande. Par la suite, de nombreuse recherches on été fait à-propos de la commande des systèmes non linéaires et plusieurs schémas de commande non linéaires on été proposés on essayant de résoudre a chaque fois les problèmes de commande de ce type des systèmes. Parmi ces schémas de commande on trouve la commande adaptative qui vient pour résoudre le problème de synthèse en cas des systèmes incertains ou variables dans le temps.

La naissance de la logique floue était l'une des solutions les plus intéressantes dans la commande adaptative, toute en exploitant ses propriétés d'approximations universelles dans l'estimation des non linéarités inconnues.

Les premières utilisations du flou dans la commande adaptative remonte aux travaux de Wang dans [15] et [9], suivie de plusieurs travaux après lui, mais sans prendre en considération un problème qui apparait par la suite dans la pratique, celui de la saturation. La saturation est une caractéristique rencontrée sur les actionneurs des systèmes réels c'est l'un des principaux problèmes rencontrés durant la conception d'une boucle de régulation. Cette caractéristique est pratiquement inévitable dans les systèmes réels. Elle inclut des contraintes sur l'amplitude ou la vitesse de la variable d'entrée. Ces contraintes peuvent être dues à des restrictions volontairement placées sur l'actionneur pour éviter la détérioration du système et/ou des limitations physiques sur les actionneurs eux mêmes.

La négligence de la saturation, pendant la mise en œuvre d'un contrôleur, conduit à une dégradation significative dans les performances de la boucle de commande, elle peut même causer l'instabilité dans le cas des systèmes instables en boucle ouverte. Cela, revient au fait que la saturation impose des limitations sur la capacité du shéma de commande.

Plusieurs méthodes et outils ont été développés pour faire face aux différents problèmes de la saturation consacrée à la commande des systèmes linéaires et/ou non linéaires avec un modèle dynamique parfaitement connu. Dans le cas des dynamiques inconnues ou on fait appel aux techniques basées sur la commande adaptative [2], [16], Le problème de la saturation s'avère de plus en plus délicat et surtout dans le cas d'une adaptation continue durant la saturation. De ce fait, on peut avoir des dégradations néfastes dans les performances et même dans la stabilité du système de commande. Par conséquent, la conception d'un schéma de commande adaptatif en présence de la saturation peut être une tache délicate.

Ainsi, le travail présenté dans ce projet s'intéresse au problème de la commande adaptative floue avec saturation de l'entrée de commande pour une classe de systèmes non linéaires incertains et penchera sur des approches et solution proposées, pour remédier au problème de la saturation comme la modification de la loi de commande de telle sorte que le signal généré par le contrôleur reste toujours dans la zone linéaire, i.e. non saturé [27], et la modification du signal d'erreur utilisé dans l'adaptation par élimination de la composante due a la saturation [24]. Ce mémoire est organisé en quatre chapitres:

Le premier chapitre présente quelques notions théoriques sur la logique floue et la structure des systèmes flous. L'accent est mis particulièrement sur les systèmes flous de Takagi-Sugeno.

Le deuxième chapitre se commence par un petit rappel sur la stabilité de Lyapunov, puis quelques définitions de la commande adaptative et la commande adaptative floue. Puis une présentation du développement de la commande adaptative indirecte floue stable, pour une classe de systèmes non linéaires monovariables.Les systémes flous ici sont utilisés pour l'approximation des dynamiques inconnus.Le problème de division par zéro est résolu, par l'utilisation d'un algorithme de projection.

Dans le troisième chapitre, la commande adaptative indirecte floue est présentée où le problème de saturation est résolu par une combinaison entre deux solutions où dans la première, la loi de commande est modifiée de façon à fournir un signal de commande non saturé. Dans la seconde solution, des modifications sont apportées aux lois d'adaptation paramétriques afin d'obtenir une adaptation stable insensible à l'erreur causée par la saturation du signal de commande.

Dans le quatrième chapitre, la commande adaptative directe floue est développée.Les systémes flous ici sont utilisés pour approximer directement la loi de commande . Le problème de saturation est évité par la modification des lois d'adaptation paramétriques. Et on termine par une conclusion.

CHAPTER 1

Les systèmes flous

Sommaire

1.1 Introduction **1**

1.2 Logique floue **2**

1.2.1 Définition 2

1.2.2 Quelques avantages de la logique floue 2

1.2.3 Les inconvénients 3

1.2.4 Application de la logique floue 3

1.3 Les Concepts de la Logique Floue **3**

1.3.1 Variables linguistiques et ensembles flous 3

1.3.2 Sous-ensembles flous 4

1.3.3 Caractéristique d'un sous-ensemble flou 4

1.3.4 Fonction d'appartenance 5

1.3.5 Variable linguistique 6

1.3.6 Opérations de base sur les sous ensembles flous 7

1.4 La structure d'un système flou **8**

1.4.1 Interface de fuzzification 8

1.4.2 Base de règles floues 8

1.4.3 Moteur d'inférence 8

1.4.4 Interface de Déffuzification 9

1.5 Types de système flou **9**

1.5.1 Développement du modèle de Takagi-Sugéno 10

1.6 Conclusion **11**

1.1 Introduction

Au départ, La logique floue s'affirme comme une technique opérationnelle. Utilisée à côté d'autres techniques de contrôle avancé, elle fait une entrée mais appréciée dans les automatismes de contrôle industriel La logique floue ne remplace pas nécessairement les systèmes de régulation conventionnels, elle est complémentaire. Ses avantages viennent notamment de ses capacités: formaliser et simuler l'expertise

d'un opérateur ou d'un concepteur dans la conduite et le réglage d'un procédé; donner une réponse simple pour les procédés dont la modélisation est difficile; prendre en compte sans discontinuité des cas ou exceptions de natures différentes, et les intégrer au fur et à mesure dans l'expertise; prendre en compte plusieurs variables et effectuer de la « fusion pondérée » des grandeurs d'influences.

L'utilisation de la logique floue dans la commande des systèmes nécessite d'avoir des connaissances approfondies sur cette théorie et son utilisation dans la commande, d'où la nécessité de présenter quelques bases générales de la logique floue: principe de la logique floue, Théorie des ensembles flous, les relations floues ainsi que la structure des systèmes flous.

Dans ce qui nous suit, nous allons tenter de passer sur la pluparts des outils et principes de la logique floue, en détaillant la structure des systèmes flous, notamment le système de Takagi-Sugéno utilisé dans notre travail pour l'estimation des dynamiques inconnues.

1.2 Logique floue

La logique floue est à l'image de logique binaire, clairement et logiquement étayée à l'aide de théorie mathématique. Elle est apparue en 1965 avec la publication par Lotfi Zadeh d'un article intitulé «les ensembles flous» [1].

La logique floue permet de manipuler des symboles et d'inférer des actions en utilisant des règles logiques à partir des prémisses imprécises ou incertaines.

En 1975 Mamdani a développé à partir de quelques principes de la logique floue une stratégie de contrôle de procède, par la suite ces travaux ont été repris, et les techniques utilisées dans les réalisations récentes sont issues [5].

1.2.1 Définition

Logique floue : « logique qui substitue à la logique binaire une logique fondée sur des variables pouvant prendre, outre les valeurs « vrai » ou« faux », les valeurs intermédiaires « vrai » ou « faux » avec une certaine probabilité » [2].

1.2.2 Quelques avantages de la logique floue

- Nous n'avons pas besoin d'établir le modèle mathématique. Cela évite des années de travail sur des modèles.
- Traitement des systèmes à comportement complexe (même si on a un exemple simplifié, cela reste un système complexe en mathématique pure).
- Nous pouvons aujourd'hui servir des solutions matérielles comme des microprocesseurs uniquement dédiés aux calculs flous, ce qui permet d'accroître le rendement puisque cela réduit la programmation.

- *L'ajout ou le retrait des règles ne change quasiment rien au principe de* la logique floue, alors qu'en logique bivalente (logique classique), cela peut changer du tout au tout.
- On peut augmenter facilement le nombre d'entrées sans réellement compliquer les calculs (alors qu'en logique simple la taille des calculs est exponentielle).

1.2.3 Les inconvénients

- performances en boucle fermée dépendent de l'expertise.
- il n'existe pas de théorie générale qui caractérise rigoureusement la stabilité, la robustesse (difficulté de certification dans le transport, espace).

1.2.4 Application de la logique floue

Les domaines d'applications de la logique floue concernant principalement les problèmes où les données ne peuvent être formulées de manière explicite, ainsi que des techniques de contrôle et de réglages, lorsque les moyens classiques atteignent leurs limites (exemples systèmes non linéaires, etc.).

La logique floue est bien connue par des automaticiens pour ses applications dans le contrôle et commande de procédés, appelé alors couramment «contrôle flou». Tout comme un contrôleur (ou correcteur) classique [ ], Son but est de traiter les problèmes de commande de processus avec une approche différente de l'automatique classique. La logique floue est aussi connue par sa capacité d'approximer n'importe quel fonction d'où son utilisation, pour approximer les dynamiques inconnues des systèmes incertains ou difficile à modéliser.

1.3 Les Concepts de la Logique Floue

Ce rapport permet de considérer des classes d'objets dont les frontières ne sont pas clairement déterminées, par l'introduction d'une fonction caractéristique (fonctions d'appartenance des objets à la classe) prenant des valeurs courantes entre 0 et 1, contrairement aux ensembles «booléens», dont la fonction caractéristique ne prend que deux valeurs possibles 1 et 0.

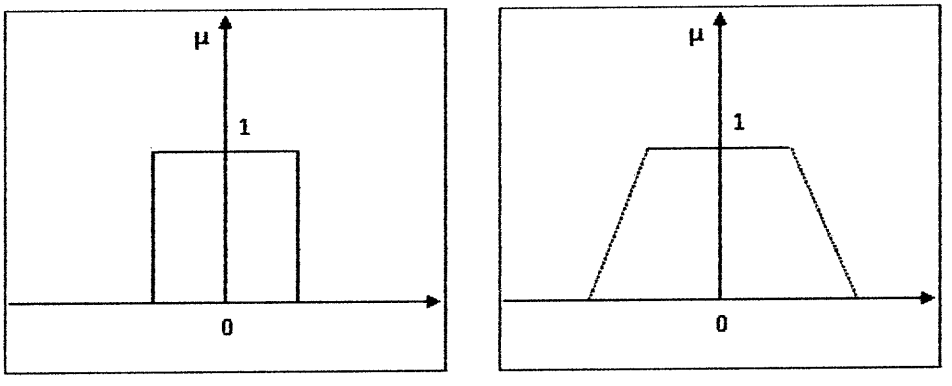
1.3.1 Variables linguistiques et ensembles flous

La description imprécise d'une certaine situation, d'un phénomène ou d'une grandeur physique ne peut se faire que par des expressions relatives ou floues. Ces différentes classes d'expressions floues dites ensembles flous forment ce qu'on appelle

des variables linguistiques. Afin de pouvoir traiter numériquement ces variables linguistiques qui sont normalisées généralement sur un intervalle bien déterminé appelé univers de discours, il faut les soumettre à une définition mathématique à base de fonctions d'appartenance qui montrent le degré de vérification de ces variables linguistiques relativement aux différents sous-ensembles flous de la même classe.

1.3.2 Sous-ensembles flous

Si on désigne par G un sous-ensemble flou dans un univers de discours U ,ce sous ensemble flou et caractérisé par une fonction d'appartenance qui prend des valeurs entre 0 et 1 à la différence pour le sous ensemble booléen qui prend deux valeurs 0 ou 1 (figure(1.1)).



(a) Logique booléenne (b) Logique floue

Figure 1.1: Fonction caractéristique d'appartenance pour la logique booléenne et la logique floue.

Pour la logique floue, la fonction d'appartenance peut êtres écrite sous la forme:

$$\mu_G : U \rightarrow [0\ 1] \tag{1.1}$$

$$G = \{(u, \mu_G(u))\ |u \in U)\} \tag{1.2}$$

1.3.3 Caractéristique d'un sous-ensemble flou

Les caractéristiques d'un sous ensemble flou G de l'univers U les plus utilisés pour décrire sont celles qui montrent à quel point il diffère d'un sous ensemble ordinaire de U.

a-Support:

Le support de G, noté supp(G), est la partie de U sur la quelle la fonction d'appartenance de G n'est pas nulle :

$$Supp(G) = \{x/ \ \mu_G(x) > 0\} \tag{1.3}$$

b-Hauteur:

La hauteur, notée hgt (G), du sous ensemble flou G de U est la plus grande valeur prise par sa fonction d'appartenance :

$$hgt(G) = Supp\mu_G(x) \tag{1.4}$$

Le sous ensemble flou G de U est normalisé si sa hauteur hgt(G) est égale à 1.

c-Noyau:

Le noyau de G, noté cor(G), est le sous ensemble des éléments de U pour lesquels la fonction d'appartenance de G vaut 1.

$$cor(G) = \left\{ \frac{x}{\mu_G(x)} = 1 \right\} \tag{1.5}$$

d-Cardinalité :

Lorsque U est fini, on caractérise également le sous ensemble flou G de U par sa cardinalité, qui indique le degré global avec lequel les éléments de U appartiennent à G.

1.3.4 Fonction d'appartenance

Chaque sous ensemble flou peut être représenté par sa fonction d'appartenance. Si le sous ensemble de référence est discret, les fonctions d'appartenance sont des valeurs discrètes de l'intervalle [0,1]. Si l'univers du discours est un sous ensemble continu, nous pouvons représenter ces valeurs analytiquement comme fonction d'appartenance. Les fonctions d'appartenance peuvent avoir différentes formes, parmi eux on peut citer [1, 7, 8, 9, 11] :

- Fonction triangulaire :

$$\mu(x) = max\left[min(\frac{x-a}{b-a}, \frac{c-x}{c-b}), 0\right] \tag{1.6}$$

- Fonction trapézoïdale :

$$\mu(x) = max\left[min(\frac{x-a}{b-a}, 1, \frac{d-x}{d-c}), 0\right] \tag{1.7}$$

- Fonction gaussienne :

$$\mu(x) = exp\left[-\frac{(x-a)^2}{2\sigma^2}\right] \tag{1.8}$$

- Fonction sigmoïde :

$$\mu(x) = \frac{1}{1+exp(-a(x-c))} \tag{1.9}$$

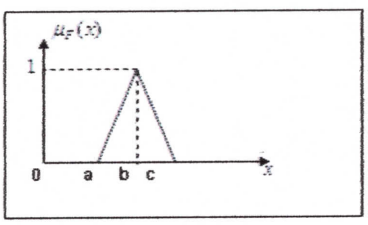


Figure 1.2: Fonction triangulaire

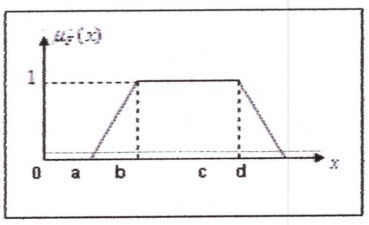


Figure 1.3: Fonction trapézoïdale

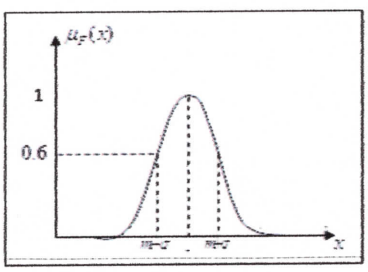


Figure 1.4: Fonction gaussienne

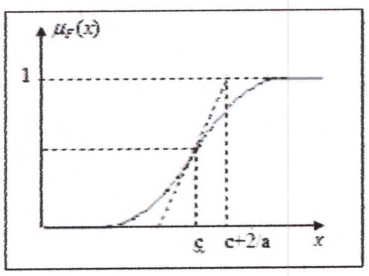


Figure 1.5: Fonction sigmoïdale

جامعة محمد الصديق بن يحيى جيجل
مكتبة الكلية
كلية العلوم والتكنولوجيا

1.3.5 Variable linguistique

Le sous ensemble de référence d'un mot d'un langage naturel s'appelle l'univers du discours. L'univers de discours d'un mot est un sous ensemble de termes qui évoquent le même concept mais à degrés différents.

Une variable linguistique représente un état du système à régler ou d'une variable de réglage dans un contrôleur flou. Sa valeur est définie dans des termes linguistiques qui peuvent être des mots ou des phrases d'un langage naturel.

Chaque variable linguistique est caractérisée par un sous ensemble tel que x , T(x), X ou

- x est nom de la variable.
- T(x) est le sous ensemble des valeurs linguistiques que peut prendre x.
- X est l'univers du discours avec la valeur de base.

1.3.6 Opérations de base sur les sous ensembles flous

Ces opérateurs permettent d'écrire des combinaisons logiques entre notions floues, c'est-à-dire de faire des calculs sur des degrés de vérité [12]. Comme pour la logique classique, on peut définir des opérateurs ET, OU, négation...etc.

On suppose A et B deux sous ensembles flous d'un même univers de discours X.

- Opérateur NON (compliment) B est le compliment d'un sous ensemble flou A si seulement si :

$$\forall x \in X,\ \mu_B = 1 - \mu_A(x) \tag{1.10}$$

- Opérateur ET (l'intersection)

$$Soit\ A \cap B = C\ on\ a\ \mu_C(x) = min(\mu_A(x), \mu_B(x)) \tag{1.11}$$

- Opérateur OU(l'union)

$$\forall x \in X,\ A \cup B = C,\ \mu_C(x) = max(\mu_A(x), \mu_B(x)) \tag{1.12}$$

- Inclusion

$$A \subset B \iff \forall x \in X,\ \mu_A(x) \leq \mu_B(x) \tag{1.13}$$

- Produit cartésien

Soit $A_1, A_2, \cdots, A_n$ des sous ensembles flous, dans les univers de discours $X_1, X2, ...X_n$ respectivement. le produit cartésien noté $A_1 \times A_2 \times \cdots \times A_n$ est un sous ensemble flou défini sur $X_1 \times X_2 \times \cdots \times X_n$ dont sa fonction d'appartenance est exprimée par :

$$\mu_{A_1 \times A_2 \times \cdots \times A_n}(x_1, x_2, \cdots, x_n) = \mu_{A_1}(x_1) \times \mu_{A_2}(x_2) \times \cdots \times \mu_{A_n}(x_n)$$

$$= min\{\mu_{A_1}(x_1), \mu_{A_2}(x_2), \cdots, \mu_{A_n}(x_n)\} \tag{1.14}$$

- Egalité flou

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

$$\forall x \in X\ : \mu_A(x)\ = \mu_B(x) \tag{1.15}$$

1.4 La structure d'un système flou

La structure conventionnelle d'un système floue est présentée par figure (1.6), il est composé de quatre blocs distincts dont les définitions son données ci-dessous

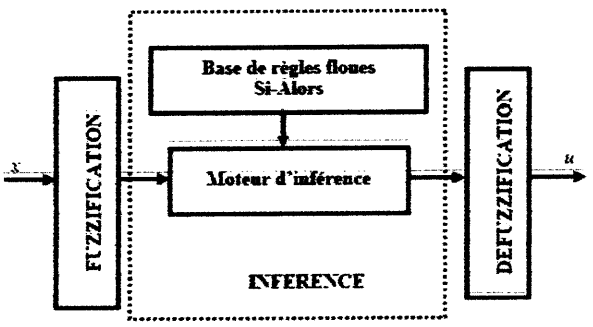


Figure 1.6: Structure de base d'un système flou.

1.4.1 Interface de fuzzification

La fuzzification est réalisée dans l'interface d'entrée du système 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.

1.4.2 Base de règles floues

Les règles floues représentent le cœur du système, elle rassemble l'ensemble des règles floues de type «Si-Alors» décrivant en termes linguistiques basés sur la connaissance d'un expert le comportement dynamique du système.

$$R_i : Si\ x_1 est\ A_1^i\ et \ldots et\ x_n\ A_n^i\ Alors\ u_i\ est\ B^i$$

Avec :

$(x_1, \ldots, x_n)$ et u_i sont respectivement les entrées et la sortie du système floue.

1.4.3 Moteur d'inférence

L'inférence floue est le processus de formulation de la relation entre les entrées et les sorties par logique floue. Cette relation offre une base avec laquelle la décision est

prise par le système flou. L'inférence floue fait appel alors aux concepts expliqués dans les sections précédents. A savoir: Fonctions d'appartenance, les opérateurs flous et les règles floues. Nous distinguons une variété importante d'inférences floues, L'inférence de Max-Min (Mamdani), Max-produit, Somme-produit et de Sugéno.

Les quatre méthodes d'inférences sont différentes par la manière de déterminer les sorties. Parmi les quartes méthodes on vas présenter celle utilisée dans notre travail:

Méthode d'inférence de Sugéno:

Sugeno a proposé une méthode d'inférence floue qui garantit la continuité de la sortie [3]. Cette méthode d'inférence s'avère très efficace dans des applications faisant intervenir à la fois des techniques linéaires, d'optimisation et adaptatives. Dans l'inférence de Sugeno, les règles floues sont exprimées de la façon suivante :

$$\text{Si } (x_1 \ est \ A_{i1}) et \ldots et (X_m \ est \ A_{im}) \ Alors \ y = f_i(x_1, \ldots, x_m)$$

Dans laquelle $x_1, \ldots, x_m$ et y sont des éléments de l'univers du discours $x_1, \ldots, x_m$ et $A_{i1}, \ldots, A_{im}$ sont des termes linguistiques sur ces mêmes univers du discours. y est une fonction de $x_1, \ldots, x_m$.

1.4.4 Interface de Déffuzification

Pour avoir a la sortie du système flou un signal précis, il faudra donc transformer la fonction d'appartenance résultante obtenue à la sortie du moteur d'inférence en une valeur précise. Cette opération est appelée déffuzification. Parmi les méthodes utilisées dans la littérature [1, 5, 6], on peut citer : la méthode de la hauteur, la méthode de la hauteur modifiée, la méthode de la valeur maximum ainsi que le centre de gravité que l'on vas l'utiliser dans ce travaille.

1.4.4.1 Méthode du centre de gravité

Cette méthode est la plus souvent utilisée et donne généralement les meilleurs résultats. Elle consiste à prendre comme solution l'abscisse du centre de gravité des solutions donnée par :

$$y^* = \frac{\int_{y \in R} y \mu_B(x) dy}{\int_{y \in R} y \mu_B(x) dy} \tag{1.16}$$

1.5 Types de système flou

On peut classer les systèmes flous en deux grandes catégories :

- Système flou a conclusion symbolique appelés, système de Mamdani [10] et dite aussi systèmes flous standards.
- Systèmes flous a conclusions fonctionnelle appelés, systèmes flous de Takagi-Sugéno (TS) , [13, 14] .

1.5.1 Développement du modèle de Takagi-Sugéno

Soit $x = [x_1, \ldots, x_n]^T$ l'entrée du système flou, y la sortie, le système est représenté par l'ensemble des règles suivantes :

$$R_k : \; si \; x_1 \; est \; \tilde{A}_1^k \; et \ldots et \; si \; x_n \; est \; \tilde{A}_n^k \; Alors \; y = f_k(x) \tag{1.17}$$

avec $(k = 1, \ldots, N)$. Pour chaque x_i associé m_i ensembles flous $\tilde{A}_i^j$ dans X_i tel que pour $x_i \in X_i$ il existe au moins un degré d'appartenance $\mu_{\tilde{A}_i^j}(x_i) \neq 0$ où $i = 1, \ldots, n_i$ et $j = 1, \ldots, m_i$ la base de règles du système flou comporte N règles.$f_k(x)$ est une fonction numérique dans l'espace de sortie, elle peut être une fonction polynomiale en fonction des variables d'entrées, comme elle peut prendre le rôle d'une fonction arbitraire tant qu'elle puisse décrire convenablement le comportement du système étudier. Prenons le cas où $f_k(x)$ est linéaire :

$$f_k(x) = a_0^k + \sum_{i=1}^{n} a_i^k x_i \tag{1.18}$$

Il s'agit dans ce cas d'un système flou de Takagi-Sugéno d'ordre un (TS1) pour $f_k(x) = a_0^k$ c'est un système d'ordre zéro (TS0), la sortie totale y du système flou est obtenue directement sans passer par la défuzzification en calculant la moyenne pondérée suivante :

$$y(x) = \frac{\sum_{k=1}^{N} \mu_k(x) \;\; f_k(x)}{\sum_{k=1}^{N} \mu_k(x)} \tag{1.19}$$

Avec $R_k(x)$ est le degré de confiance ou d'activation de la règle R_k représentée par :

$$\mu_k(x) \; = \prod_{i=1}^{n} \mu_{\tilde{A}_i^k}(x_i), \;\; \tilde{A}_i^k \in \; \left\{ \tilde{A}_i^1, \ldots, \tilde{A}_i^{m_i} \right\} \tag{1.20}$$

On peut simplifier la sortie à :

$$y(x) = \; w^T(x).\theta \tag{1.21}$$

Avec : $\theta = \left[a^1, \ldots, a^N\right]^T$: Vecteur constitué des paramètres de la partie conclusion des règles floues

$w(x) = \ [w_1(x), \ldots, w_N(x)]^T$: Vecteur des fonctions floues de base, ou chacune des composantes est donnée par :

$$w_k(x) = \frac{\mu_k(x)}{\sum_{k=1}^{N} \mu_k(x)}, k = 1, \ldots, N \tag{1.22}$$

Remarque :

Lorsque $f_k(x)$ est un polynôme d'ordre un. Le système flou résultant est dit système flou de Takagi-Sugeno un (TS1). Si $f_k(x)$ est un polynôme d'ordre zéro (une constante), on a donc un système flou d'ordre zéro (TS0). Sa sortie peut s'écrit comme suit:

$$y_k(x) = \frac{\sum_{k=1}^{N} \mu_k(x).a^k}{\sum_{k=1}^{N} \mu_k(x)} \tag{1.23}$$

Dans notre cas on s'intéresse aux modèles flous de Takagi-Sugéno à conclusion constante, i.e., les systèmes flous de Takagi-Sugéno d'ordre zéro. Dans le reste de ce travail on restreint la sortie du système floue de TS0 (1.23) à la forme réduite (1.21). Des fonctions d'appartenances gaussiennes sont utilisées pour caractériser les labels A_i^j , ce choix revient à leurs expression mathématique compacte et la facilité de leurs manipulation, on a :

$$\mu_{A_i^j}(x) = exp\left\{-\frac{1}{2}\left[\frac{x_i - c_i^j}{v_i^j}\right]\right\} \tag{1.24}$$

Où c_i^j la moyenne et v_i^j la variance.

1.6 Conclusion

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

Les systèmes flous se distingué essentiellement par la nature de la conclusion de leurs règles, Symbolique, système flou de Mamdani, est algébrique pour celui de Takagi-Sugéno.

Malheureusement la manipulation de règles non précises peut générer un nombre d'erreurs non négligeable. La mise en place d'un système flou nécessite donc une attention particulière lors de la phase de teste de manière à détecter Les éventuelles aberrations du système.

CHAPTER 2

Commande adaptative indirecte floue

Sommaire

2.1 Introduction **13**
2.2 Stabilité au sens de Lyapunov **14**
2.2.1 Méthode directe de Lyapunov 14
2.2.2 Théorème de stabilité par la méthode directe 15
2.2.3 Lemme de Barbalat 15
2.3 Commande Adaptative **16**
2.4 Commande adaptative floue **17**
2.4.1 Commande adaptative floue directe 18
2.4.2 Commande adaptative floue indirecte 18
2.5 Conception de la commande adaptative floue indirecte . . . **19**
2.5.1 Position du problème 19
2.5.2 Objectif 20
2.5.3 Calcul de l'erreur 20
2.5.4 Approximation des fonctions non linéaires par les systèmes flous 21
2.5.5 Loi de commande 22
2.5.6 Conception de la loi de commande proposée 23
2.5.7 Démonstration de la stabilité 24
2.6 Résultats de simulation **27**
2.7 Conclusion **29**

2.1 Introduction

La commande adaptative a débutée dans les années 50 comme solution pour contrôler les processus fonctionnant sous des conditions et environnement variables dans le temps .Dans les années 60 plusieurs contributions de théorie de la commande ont été introduites dans le développement de la commande adaptative, comme par exemple l'approche d'état et les théories de stabilité. Au début des années 70 les

différentes méthodes d'estimation ont été introduites dans la commande adaptative.L'utilisation de la commande adaptative a commencé au début des années 80 en parallèle avec une rapide évolution en micro-électronique qui permet d'implémenter des Régulateurs adaptatifs sur des systèmes a microprocesseurs.

La naissance de l'exploitation du flou dans cette commande remonte aux travaux de Wang [15] et [9], par la suite plusieurs approches ont été proposés [17, 18] et plusieurs schémas de commande floue adaptative ont été proposés pour les systèmes non linéaires, Dans le but d'obtenir de meilleures performances, et en se basant sur la propriété d'approximation universelle des systèmes flous. On peut distinguer deux type de commande floue adaptative : la commande adaptative floue directe et la commande adaptative floue indirecte. Dans la commande directe, un système flou est employé pour décrire directement la loi de commande et les paramètres du système flou sont directement ajustés pour atteindre les objectifs de commande. Par contre, Dans la commande indirecte, des systèmes flous sont utilisés pour estimer la dynamique du système et la loi de commande est obtenue à partir de ces estimées et elle repose sur le principe d'équivalence certaine.

Dans ce chapitre, considérons une approche de commande adaptative indirecte floue. Dans cette approche, les systèmes flous seront utilisés pour approcher les non-linéarités inconnues et la loi de commande est synthétisée à partir de la méthode de linéarisation par bouclage [2], [28]. Le problème de la possibilité de division par zéro et la singularité de la loi de commande est évité par l'utilisation des algorithmes de projection pour l'adaptation des paramètres des systèmes flous [29] et le problème de la présence des erreurs d'approximation floues est traité par des termes de commande robustes construits par la technique des modes glissants [2].

Pour cette approche de commande développée, des résultats de simulation effectuée sur un système sont données pour montrer les performances de cette approche.

2.2 Stabilité au sens de Lyapunov

Si le problème de stabilité se présente d'une manière remarquablement simple dans le cas des systèmes linéaires, tant en ce qui concerne le définition de la stabilité que les conditions de stabilité, il n'en est pas de même pour les systèmes non linéaires.la théorie de Lyapunov, à la quelle l'emploi des variables d'état permet de donner une forme particulièrement satisfaisante, constitue la seule théorie générale de la stabilité des systèmes dynamiques.

2.2.1 Méthode directe 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 un 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 $x(x \in R^n)$ est toujours dirigée vers l'intérieur, ce point finira par arriver à l'origine ; dans le cas contraire, le point x s'en écartera davantage. Dans quelques classes de systèmes physiques, la fonction V peut être choisie comme étant l'énergie du système.

2.2.2 Théorème de stabilité par la méthode directe

Considérons le système suivant :

$$\begin{cases} \dot{x} = f(x) \\ x(0) = x_0 \end{cases} \tag{2.1}$$

Théorème 2.1 : Supposons que l'origine est un point singulier de (2.1) et posons $V : U \to R$ une fonction de classe C^1 définie dans un voisinage U de l'origine, telle que :

1. $V(0) = 0$
2. $V(x) > 0$ pour $x \neq 0 \;\; x \in U$
   si
3. $\frac{dV(x)}{dt} \leq 0$ dans $U - \{0\}$ l'origine est donc stable.
4. $\frac{dV(x)}{dt} < 0$ dans $U - \{0\}$ l'origine est asymptotiquement stable.
5. $\frac{dV(x)}{dt} > 0$ dans $U - \{0\}$ l'origine est instable.

Dans ces conditions, la fonction V est appelée fonction de Lyapunov ; elle est définie positive si 1) et 2) sont vérifiées. Notons qu'avec ces deux hypothèses, la fonction V est seulement une fonction candidate "fonction de signe définie".

2.2.3 Lemme de Barbalat

Pour la démonstration de convergence des erreurs de poursuite vers zéro. On aura besoin du lemme de Barbalat [25] qui est comme suit :

lemme 2.1 : *si la fonction $f : R^+ \rightarrow R$ est uniformément continue pour et si la* limite de l'intégrale

$$\lim_{x\rightarrow\infty} \int_0^t |f(\tau)| d\tau \tag{2.2}$$

existe et finie, alors :

$$\lim_{x\rightarrow\infty} f(t) = 0 \tag{2.3}$$

Théorème 2.2 : Si une fonction scalaire$V(x)$ de classe $C^1 vrifie$:

1. $V(0) = 0$
2. $V(x) > 0 \quad \forall x \neq 0$
3. $\lim_{||x||\rightarrow\infty} V(x) = \infty$ (c'est-à-dire, $V(x)$ n'est pas bornée radialement) .
4. $\dot{V}(x) \leq 0 \quad \forall x$
5. $\dot{V}(x)$ est uniformément continue (cette condition est généralement remplacée par la condition suffisante$\ddot{V}(x)$ existe est bornée).

Alors,$\dot{V}(x) \rightarrow$ 0 quand $t \rightarrow \infty$.

2.3 Commande Adaptative

La commande adaptative est un ensemble de techniques utilisées pour l'ajustement automatique en temps réel des régulateurs des boucles de commande afin de réaliser ou maintenir un certain niveau de performances quand les paramètres du procédé à commander sont soit inconnus soit variantes dans le temps [19].

En principe, un système de commande adaptative mesure un certain indice de performance (IP) 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'IP La figure.2.1 représente le principe général de la commande adaptative [19].

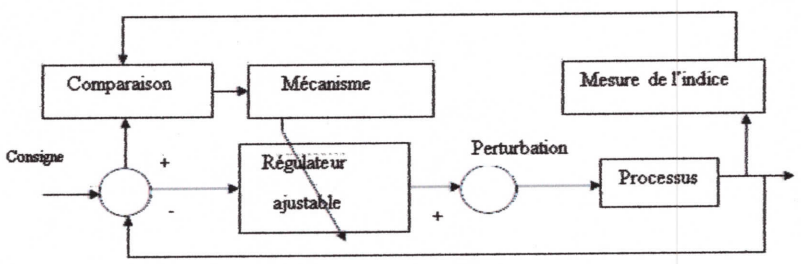


Figure 2.1: Principe de commande adaptative.

Plusieurs schémas de commande adaptative sont proposés dans la théorie , ces stratégies de commande peuvent être classés en trois grande famille :

- Approximation des stratégies de commande optimale stochastique «Duale ».
- Système de commande auto - ajustable (self-tunning control).
- Commande adaptative à modèle de référence (MRAC) [20].

La synthèse d'adaptation et le problème principal qui se pose lors de la réalisation des systèmes adaptatifs en général. Plusieurs études théoriques ont été développées dans le but de fournir des méthodes de synthèse pour ce mécanisme, en faisant appel à la théorie de base de l'automatique. Les premiers algorithmes étaient construits à partir des méthodes de gradient (optimisation) [12]. Les stratégies résultantes conduisent a deux grandes techniques adaptatives à mettre en œuvre [13]. Ces deux grandes techniques sont:

- Commande adaptative directe : les paramètres du régulateur sont ajustés en une seule étape.
- Commande adaptative indirecte : l'adaptation des paramètres se fait en deux étapes, estimation des dynamiques inconnue du système puis le calcul du régulateur à partir de cette estimation.

Le développement de la théorie de la logique floue a donné naissance à un axe de recherche trés important qui est la commande adaptative floue où on essié d'exploiter toutes les informations linguistiques concernant le modele du système pour l'approximer (cas indirecte) ou bien d'exploiter les informations linguistiques concernant la commande (cas directe). Les premiers travaux sur la commande adaptative floue revient aux travaux de Wang [15], [9], dans notre travail on s'intéresse a la commande adaptative floue, pour cela dans la suite de ce chapitre on va développer un schéma de commande basé sur cette technique.

2.4 Commande adaptative floue

La commande adaptative floue est un modèle de commande adaptative où l'on exploite les propriétés d'approximation universelle des systèmes flous (voir annexe A), pour approcher les non linéarités inconnus. Plusieurs techniques d'optimisation et d'apprentissage ont été développées pour mieux déterminer les paramètres du système flou. Il éxiste deux types de commande adaptative flous, directs et indirects.

2.4.1 Commande adaptative floue directe

Pour ce type de commande, les approximateurs sont utilisées pour approcher directement le contrôleur, sans passer par l'identification du modèle du système, ainsi dans ce cas c'est l'approximateur (système flou) qui joue le rôle du contrôleur.

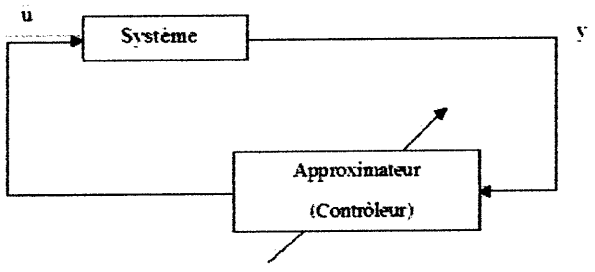


Figure 2.2: Commande adaptative directe floue.

2.4.2 Commande adaptative floue indirecte

Dans ce cas, c'est l'ajustement et l'approximation des paramètres et fonctions inconnus du système, par la suit ces approximations seront utilisées dans la loi de commande à la place des fonctions réel du système.

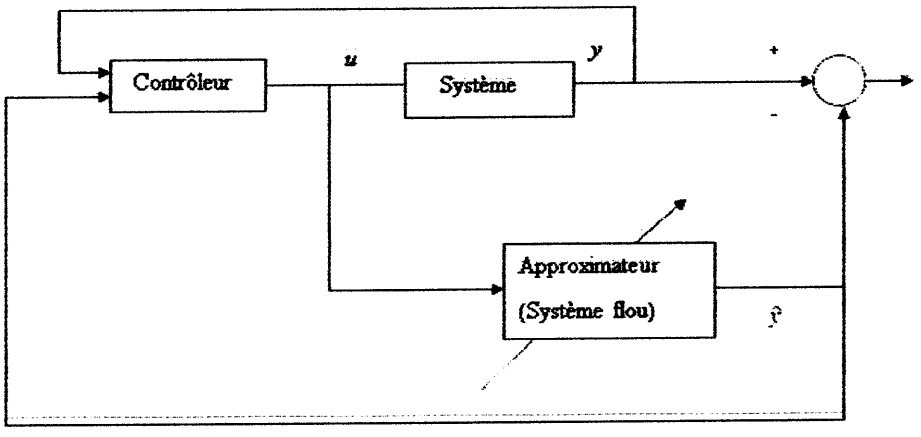


Figure 2.3: Commande adaptative indirecte floue

2.5 Conception de la commande adaptative floue indirecte

2.5.1 Position du problème

Considérons le système non linéaire monovariable régit par les équations suivantes :

$$\begin{cases} \dot{x}_i = x_{i+1}, i = 1, ..., n-1 \\ \dot{x}_n = f(x) + g(x)u \\ y = x_1 \end{cases} \tag{2.4}$$

Pour faciliter la synthèse d'une loi de commande, il est souhaitable de passer d'une représentation d'état vers une représentation entrée-sortie de façon à mettre en évidence la relation directe entre la sortie et la commande. Pour ce faire, il est nécessaire de dériver la sortie du système jusqu'à l'apparition de l'entrée, on obtient alors :

$$y^{(n)} = f(y) + g(x)u \tag{2.5}$$

Avec:

- $x = [x_1, ..., x_n]^T \in R^n$ est le vecteur d'état supposé mesurable.
- $u \in R$ est l'entrée de commande.
- $y \in R$ est la sortie du système.
- $f(x)$ et $g(x)$ sont des fonctions non linéaires continues incertaines.

جامعة محمد الصديق بن يحيى - جيجل
مكتبة الكلية
كلية العلوم و التكنولوجيا

Afin de développer notre contrôleur adaptatif, les suppositions suivantes sont nécessaires par la suite.

Supposition 2.1 : le gain de la commande $g(x) \neq 0$,pour tout x est de signe connu. Sans perte de généralité, il est supposé que $g(x) \geq \underline{g} > 0$ avec $\underline{g}$ une constante donnée.

Supposition 2.2 : Le vecteur d'état x est mesurable.

Supposition 2.3 : La trajectoire désirée $y_d(t)$ et ses dérivées jusqu'à l'ordre n sont connues et bornées.

2.5.2 Objectif

Notre objectif est d'assurer :

1. La poursuite telle que la sortie$y(t)$ poursuit la trajectoire désirée $y_d(t)$.
2. La stabilité en boucle fermée, dans le sens que tous les signaux soient bornés,est cela malgré les non linéarités inconnus.

2.5.3 Calcul de l'erreur

Afin d'assurer la convergence vers la trajectoire désirée, nous commençons par la définition de l'erreur de poursuite :

$$e(t) = y_d(t) - y(t) \tag{2.6}$$

La résolution des équations de Lyapunov et de Riccati,nécessaires pour l'implantation des lois de commande, peut poser des problèmes numériques, notamment pour les systèmes de grande dimension. Afin d'éviter ce problème, la formulation de l'erreur de poursuite peut être remplacée par une expression du premier ordre d'une surface de glissement appelée aussi l'erreur filtré [30] :

$$s = \left(\frac{d}{dt} + \lambda\right)^{n-1} e(t), \lambda > 0 \tag{2.7}$$

La forme de l'erreur filtrée est intéressante dans le sens ou le problème de poursuite de $y_d(t)$, peut être remplacé par celui de stabilisation du premier ordre en s, or pour $s = 0$, la formule devient une équation différentielle linéaire avec $e = 0$ comme solution unique, ainsi que pour ses dérivées jusqu'à l'ordre $(n-1)$, de plus, la bornitude du signal $s(t)$ par une constante Φ, entraîne la bornitude de l'erreur $e(t)$ et ses dérivées par [2] :

$$|e^{(j)}(t)| \leq 2^j \lambda^{j-n+1} \Phi, j = 0, ..., n-1 \tag{2.8}$$

La dérivée de (2.7) le long de (2.5) peut être écrite sous la forme :

$$\dot{s} = v - f(x) - g(x)u \tag{2.9}$$

avec

$$v = y_d^{(n)} + \alpha_{n-1} e^{(n-1)} + ... + \alpha_1 \dot{e} \tag{2.10}$$

où

$$\alpha_j = \frac{(n-1)!}{(n-j)!(j-1)!} \lambda^{n-1}, j = 1, ..., n-1 \tag{2.11}$$

2.5.4 Approximation des fonctions non linéaires par les systèmes flous

Dans cette partie, notre objectif est d'utiliser les systèmes flous pour approcher les fonctions non linéaires inconnues $f(x)$ et $g(x)$. A partir de ces approximations floues, nous proposons des lois de commande avec leurs lois d'adaptation pour atteindre les objectifs de commande et garantir la bornitude de tous les signaux du système bouclé.

Les systèmes flous de la forme de (1.21) sont utilisés pour identifier en ligne les fonctions $f(x)$et $g(x)$.On peut écrire :

$$\hat{f}(x,\theta_f) = \omega_f^T(x)\theta_f \tag{2.12}$$

$$\hat{g}(x,\theta_g) = \omega_g^T(x)\theta_g \tag{2.13}$$

avec:
• $\omega_f(x)$ et $\omega_g(x)$ sont des vecteurs de fonctions floues de base supposés convenablement fixés en avance par l'utilisateur.
• θ_f et θ_g sont les vecteurs des paramètres à estimer.

Les fonctions $f(x)$ et $g(x)$ peuvent être exprimées en fonction des approximations floues de la manière suivante :

$$f(x) = \hat{f}(x,\theta_f^*) + \varepsilon_f(x) \tag{2.14}$$

$$g(x) = \hat{g}(x,\theta_g^*) + \varepsilon_g(x) \tag{2.15}$$

avec
• ε_f et ε_g sont les erreurs d'approximations floues qui résultent de la représentation des fonctions $f(x)$ et $g(x)$ par des systèmes flous.
• θ_f^* et θ_g^* sont respectivement les paramètres optimaux de θ_f et θ_g.Ces paramètres optimaux satisfont :

$$\theta_f^* = arg\ \underset{\theta_f}{min}\{\underset{x}{Sup}|f(x) - \hat{f}(x,\theta_f)|\} \tag{2.16}$$

$$\theta_g^* = arg\ \underset{\theta_g}{min}\{\underset{x}{Sup}|g(x) - \hat{g}(x,\theta_g)|\} \tag{2.17}$$

Ces derniers sont des constantes artificielles inconnues introduites uniquement pour faire l'étude théorique de la stabilité de l'algorithme de commande. en réalité, la connaissance de leurs valeurs n'est pas nécessaire pour l'implantation des lois de commande adaptatives.

Supposition 2.4 : on suppose que les erreurs d'approximations $\varepsilon_f(x) et \varepsilon_g(x)$ sont respectivement bornées par $\bar{\varepsilon}_f$ et $\bar{\varepsilon}_g$ (entités positives inconnues) tel que :

$$|\varepsilon_f(x)| < \bar{\varepsilon}_f, |\varepsilon_g(x)| < \bar{\varepsilon}_g$$

$\bar{\varepsilon}_f$ et $\bar{\varepsilon}_g$ représentent les bornes supérieurs des erreurs d'approximation floues atteignables en utilisant les systèmes flous choisis [17].

$$\tilde{\theta}_f = \theta_f^* - \theta_f \tag{2.18}$$

$$\tilde{\theta}_g = \theta_g^* = \theta_g \tag{2.19}$$

$\tilde{\theta}_f$ et $\tilde{\theta}_g$ sont les erreurs d'estimation paramétrique, et qui seront utilisés pour déterminer l'erreur entre les fonctions $f(x)$ et $g(x)$ et leurs approximations floues $\hat{f}(x, \theta_f)$ et $\hat{g}(x, \theta_g)$.

$$f(x) - \hat{f}(x, \theta_f) = w_f^T(x)\tilde{\theta}_f + \varepsilon_f(x) \tag{2.20}$$

$$g(x) - \hat{g}(x, \theta_g) = w_g^T(x)\tilde{\theta}_g + \varepsilon_g(x) \tag{2.21}$$

2.5.5 Loi de commande

Une loi de commande parfaite peut être conçue à base d'une méthode conventionnelle dite méthode de la linéarisation par bouclage. Cette dernière repose sur le principe de la linéarisation du système non linéaire par la neutralisation de ces non-linéarités. La mise en place d'une telle loi de commande nécessite la bonne connaissance des fonctions non linéaires $f(x)$ et $g(x)$. Malheureusement, dans notre travail ce n'est pas le cas où ces deux fonctions sont complètement inconnues. Pour cela, elles, seront remplacer par leurs approximations floues $\hat{f}(x, \theta_f)$ et $\hat{g}(x, \theta_g)$.

Considérons maintenant la loi de commande suivante :

$$u = \frac{1}{\hat{g}(x, \theta_g)} \left(-\hat{f}(x, \theta_f) + v + ks \right) \tag{2.22}$$

Malheureusement, une telle loi de commande ne peut pas garantir à elle seule la stabilité du système bouclé. Ceci est dû :

- d'une part, à l'existence des erreurs d'approximation ; En pratique, le nombre de règles est toujours limité alors les erreurs ne peuvent êtres parfaitement nulles.
- d'autre part, il y-a le problème de singularité où la loi de commande n'est pas défini lorsque $\hat{g}(x, \theta_g) = 0$.

  Les solutions proposées sont :
- Pour traiter le problème des erreurs d'approximation, on ajoute un terme de robustification en mode de glissement.

- *Pour éviter la division par zéro (singularité de commande) lors du processus* d'adaptation des paramètres θ_g , on modifie les lois d'adaptations en utilisant l'algorithme d'adaptation avec projection [13].

Dans ce qui suit, nous montrons comment modifier la loi de commande et/ou les lois d'adaptation pour surmonter ces problèmes.

2.5.6 Conception de la loi de commande proposée

Nous proposons la loi de commande suivante :

$$u = u_a + u_r \tag{2.23}$$

Avec :

- u_a :un terme de commande adaptative, utilisé pour compenser les non-linéaités du système.
- u_r :un terme de robustification, introduit pour pallier le problème des erreurs d'approximation.

$$u_a = \frac{1}{\hat{g}(x,\theta_g)}\Big(- \hat{f}(x,\theta_f) + v + ks\Big) \tag{2.24}$$

$$u_r = \frac{1}{\underline{g}}\Big(\hat{\varepsilon}_f + \hat{\varepsilon}_g|u_a|\Big)sgn(s) \tag{2.25}$$

avec $\hat{\varepsilon}_f$ et $\hat{\varepsilon}_g$ sont, respectivement, les estimées de $\bar{\varepsilon}_f$ et $\bar{\varepsilon}_g$,et $sgn(s)$ est la fonction signe définie comme suit :

$$sgn(s) = \begin{cases} +1, & s > 0 \\ 0, & s = 0 \\ -1, & s < 0 \end{cases} \tag{2.26}$$

Nous proposons l'utilisation des lois d'adaptation avec projection pour éviter le problème de singularité de (2.24) quelle n'est pas définie lorsque $\hat{g}(x,\theta_g) = 0$ et garantir la bornitude des paramètres ajustés .Aussitôt, nous introduisons une supposition sur les paramètres optimaux θ_f^* et θ_g^* .

Supposition 2.5 : Les paramètres optimaux θ_f^* et θ_g^* sont bornés comme suit :

$$|\theta_f^*| \leq M_f$$

$$g_{li} < \theta_{gi}^* < g_{ui}, i = 1, ..., P_g$$

avec M_f,g_{li},g_{ui} des paramètres donnés.

Les lois d'adaptation des paramètres des systèmes flous sont déterminées de façon à minimiser l'erreur de poursuite entre la sortie du procédé et la trajectoire

désirée et aussi afin d'assurer la borgnitude des paramètres estimés.
la loi d'adaptation avec projection des paramètres est donnée par [29] :

$$\dot{\theta}_f = -\eta_f \omega_f(x) s - \Phi_f \tag{2.27}$$

avec $\eta_f > 0$ le gain d'adaptation et Φ_f est définie comme suit :

$$\Phi_f = \begin{cases} 0 & \text{Si} \quad ||\theta_f|| < M_f \\ \eta_f \rho_f \frac{|\omega_f^T(x)\theta_f s|}{\theta_f^T \theta_f} \theta_f & \text{Sinon} \end{cases} \tag{2.28}$$

avec $\rho_f \geq 1$.

Pour θ_g , l'algorithme de projection est choisi afin de garantir la condition $\hat{g}(x, \theta_g) \neq 0$ .Il prend la forme suivante [29] :

$$\dot{\theta}_g = -\eta_g \omega_g(x) u_a s + \Phi_g \tag{2.29}$$

avec $\eta_g > 0$ le gain d'adaptation et Φ_g est définie comme suit :

$$\Phi_{gi} = \begin{cases} 0 & \text{Si} \quad g_{li} < \theta_{gi} < g_{ui} \\ \eta_g \rho_g sgn(\tilde{\theta}_{gi}) |\omega_{gi}(x) u_a s| & \text{Sinon} \end{cases} \tag{2.30}$$

avec: $\rho_g \geq 1$.
Finalement, les bornes des erreurs d'approximation sont calculées à partir des lois :

$$\dot{\hat{\varepsilon}}_f = \gamma_f |s| \tag{2.31}$$

$$\dot{\hat{\varepsilon}}_g = \gamma_g |s u_a| \tag{2.32}$$

2.5.7 Démonstration de la stabilité

En utilisant (2.23), l'équation (2.9) peut être réécrite comme suit :

$$\dot{s} = v - f(x) - g(x)u$$

$$\dot{s} = v - f(x) - g(x)(u_a + u_r)$$

$$\dot{s} = v - f(x) - g(x)u_a - g(x)u_r$$

$$\dot{s} = v - f(x) - (g(x) - \hat{g}(x, \theta_g))\, u_a - \hat{g}(x, \theta_g) u_a - g(x) u_r$$

En utilisant l'équation (2.24), $\dot{s}$ devient :

$$\dot{s} = -ks - \left(f(x) - \hat{f}(x, \theta_f)\right) - (g(x) - \hat{g}(x, \theta_g))\, u_a - g(x) u_r \tag{2.33}$$

A l'aide de (2.20) et (2.21), $\dot{s}$ devient :

$$\dot{s} = -ks - w_f^T(x)\tilde{\theta}_f - w_g^T(x)\tilde{\theta}_g u_a - g(x) u_r - \varepsilon_f(x) - \varepsilon_g(x) u_a \tag{2.34}$$

Pour démontrer que $||\theta_f|| \leq M_f$, considérons la fonctions candidate suivante :

$$V_f = \frac{1}{2}\theta_f^T\theta_f \tag{2.35}$$

D'où

$$\dot{V}_f = \theta_f^T\dot{\theta}_f \tag{2.36}$$

En utilisant (2.27), (2.36) $\dot{V}_f$ devient :

$$\dot{V}_f = -\eta_f\theta_f^T w_f(x)s - \theta_f^T\Phi_f \tag{2.37}$$

Lorsque $||\theta_f|| = M_f$, en utilisant (2.28) il vient :

$$\dot{V}_f \leq -\eta_f(\rho_f - 1)|\theta_f^T w_f(x)s| \tag{2.38}$$

et avec $\rho_f \geq 1$, il vient $\dot{V}_f \leq 0$. On assure donc que $||\theta_f|| \leq M_f$ est toujours vérifiée. Pour démontrer que les paramètres estimés θ_{gi} vérifient toujours $g_{li} \leq \theta_{gi} \leq g_{ui}$, considérons la fonction suivante :

$$V_{gi} = \frac{1}{2}\tilde{\theta}_{gi}^2 \tag{2.39}$$

d'où:

$$\dot{V}_{gi} = -\tilde{\theta}_{gi}\dot{\theta}_{gi} \tag{2.40}$$

Lorsque θ_{gi} n'est pas entre les bornes prévues, i.e., ne vérifie pas l'inégalité $g_{li} \leq \theta_{gi} \leq g_{ui}$,en utilisant (2.29) et (2.30), (2.40) devient :

$$\dot{V}_{gi} \leq -\eta_g(\rho_g - 1)|\tilde{\theta}_{gi}||w_{gi}(x)u_a s| \tag{2.41}$$

Puisque $\rho_f \geq 1$, on a donc $\dot{V}_{gi} \leq 0$ et on peut conclure que $g_{li} \leq \theta_{gi} \leq g_{ui}$ est satisfaite.

Pour démontrer la stabilité du système bouclé, nous considérons la fonction de Lyapunov suivante :

$$V = \frac{1}{2}s^2 + \frac{1}{2\eta_f}\tilde{\theta}_f^T\tilde{\theta}_f + \frac{1}{2\eta_g}\tilde{\theta}_g^T\tilde{\theta}_g + \frac{1}{2\gamma_f}\tilde{\varepsilon}_f^2 + \frac{1}{2\gamma_g}\tilde{\varepsilon}_g^2 \tag{2.42}$$

avec $\tilde{\varepsilon}_f = \bar{\bar{\varepsilon}}_f - \hat{\varepsilon}_f$ et $\tilde{\varepsilon}_g = \bar{\bar{\varepsilon}}_g - \hat{\varepsilon}_g$.
Sa dérivée temporelle est donnée par :

$$\dot{V} = s\dot{s} - \frac{1}{\eta_f}\tilde{\theta}_f^T\dot{\theta}_f - \frac{1}{\eta_g}\tilde{\theta}_g^T\dot{\theta}_g - \frac{1}{\gamma_f}\tilde{\varepsilon}_f\dot{\hat{\varepsilon}}_f - \frac{1}{\gamma_g}\tilde{\varepsilon}_g\dot{\hat{\varepsilon}}_g \tag{2.43}$$

En utilisant (2.34), $\dot{V}$ devient :

$$\dot{V} = -ks^2 + \dot{V}_\theta + \dot{V}_2 \tag{2.44}$$

avec

$$\dot{V}_\theta = -\tilde{\theta}_f^T\left[sw_f(x) + \frac{1}{\eta_f}\dot{\theta}_f\right] - \tilde{\theta}_g^T\left[sw_g(x)u_a + \frac{1}{\eta_g}\dot{\theta}_g\right] \tag{2.45}$$

$$\dot{V}_2 = -sg(x)u_r - s\varepsilon_f(x) - s\varepsilon_g(x)u_a - \frac{1}{\gamma_f}\tilde{\varepsilon}_f\dot{\hat{\varepsilon}}_f - \frac{1}{\gamma_g}\tilde{\varepsilon}_g\dot{\hat{\varepsilon}}_g \tag{2.46}$$

Si on introduit les lois d'adaptation données par (2.27) et (2.29) dans l'expression de $\dot{V}_\theta$, celle-ci sera réduite à la forme :

$$\dot{V}_\theta = \tilde{\theta}_f^T\Phi_f - \tilde{\theta}_g^T\Phi_g$$

En utilisant supposition 2.4, $\dot{V}_2$ peut être bornée comme suit :

$$\dot{V}_2 \leq -sg(x)u_r + |s|\bar{\varepsilon}_f + |su_a|\bar{\varepsilon}_g - \frac{1}{\gamma_f}(\bar{\varepsilon}_f - \hat{\varepsilon}_f)\dot{\hat{\varepsilon}}_f - \frac{1}{\gamma_g}(\bar{\varepsilon}_g - \hat{\varepsilon}_g)\dot{\hat{\varepsilon}}_g$$

A partir des équations (2.31) et (2.32), $\dot{V}_2$ devient :

$$\dot{V}_2 \leq -sg(x)u_r + |s|\hat{\varepsilon}_f + |su_a|\hat{\varepsilon}_g$$

En utilisant (2.25) et le fait $g(x) \geq \underline{g} > 0$, il vient:

$$\dot{V}_2 \leq -(g(x)/\underline{g})(|s|\hat{\varepsilon}_f + |su_a|\hat{\varepsilon}_g) + |s|\hat{\varepsilon}_f + |su_a|\hat{\varepsilon}_g \leq 0$$

Des résultats ci-dessus, (2.44) devient :

$$\dot{V} \leq -ks^2 + \tilde{\theta}_f^T\Phi_f - \tilde{\theta}_g^T\Phi_g \tag{2.47}$$

Démontrons maintenant que $\tilde{\theta}_f^T\Phi_f \leq 0$.Lorsque $||\theta_f|| < M_f$,$\Phi = 0$,la conclusion est évidente. Quand $||\theta_f|| = M_f$, puisque $||\theta_f^*|| < M_f$, on a :

$$2\theta_f^T\theta_f = ||\theta_f^*||^2 - ||\theta_f||^2 - ||\theta_f^* - \theta_f||^2 \leq 0 \tag{2.48}$$

D'où :

$$\tilde{\theta}_f^T\Phi_f = \tilde{\theta}_f^T\theta_f\frac{|w_f^T(x)\theta_f s|}{\theta_f^T\theta_f} \leq 0 \tag{2.49}$$

De la même manière, on peut démontrer que :

$$\tilde{\theta}_g^T\Phi_g \geq 0 \tag{2.50}$$

A partir de l'analyse ci-dessus, (2.47) devient :

$$\dot{V} \leq -ks^2 \tag{2.51}$$

Nous concluons, d'après le théorème de Barbalat [2],que $s(t) \to 0$ quand $t \to \infty$. Par conséquent, l'erreur de poursuite et ses dérivées jusqu'à l'ordre $n-1$ convergent vers zéro et les objectifs de commande sont alors atteints.

2.6 Résultats de simulation

Pour illustrer les performances des méthodes de commande proposées, nous considérons la commande en poursuite d'un système non linéaire incertain. Son modèle dynamique est donné par :

$$\begin{cases} \dot{x}_1 = x_2 \\ \dot{x}_2 = f(x) + g(x)u \\ y = x_1 \end{cases} \tag{2.52}$$

avec :

$$f(x) = 1.5(1 - x_1^2)x_2 - x_1 \tag{2.53}$$

$$g(x) = 1 + 0.5\cos(x_1 x_2) \tag{2.54}$$

Notons que les fonctions non linéaires $f(x)$ et $g(x)$ sont supposées inconnues. Notre objectif est de forcer la sortie du système à suivre la trajectoire :$y_d(t) = \sin(t)$.

Chacune des fonctions (2.53) et (2.54) est représentée par un système flou, et chaque système flou a pour entrée les variables x_1 et x_2 , nous utilisons un système flou avec les entrées x_1 et x_2 car la forme de la fonction est inconnue. Pour chaque variable d'entrée, on définit cinq fonctions d'appartenance Gaussiennes comme suit:

$\mu_{F_i^1}(x_i) = exp\left\{-\frac{1}{2}\left(\frac{x_i+1}{0.25}\right)^2\right\}$

$\mu_{F_i^2}(x_i) = exp\left\{-\frac{1}{2}\left(\frac{x_i+0.5}{0.25}\right)^2\right\}$

$\mu_{F_i^3}(x_i) = exp\left\{-\frac{1}{2}\left(\frac{x_i}{0.25}\right)^2\right\}$

$\mu_{F_i^4}(x_i) = exp\left\{-\frac{1}{2}\left(\frac{x_i-0.5}{0.25}\right)^2\right\}$

$\mu_{F_i^5}(x_i) = exp\left\{-\frac{1}{2}\left(\frac{x_i-1}{0.25}\right)^2\right\}$, $i = 1, 2$.

D'où, chaque système flou a une base de règles de 25 règles.

Les paramètres de synthèse sont choisis comme suit : $\lambda = 10$, $k = 10$, $\eta_f = 50$, $\eta_g = 50$, $\rho_f = 1.1$, $\rho_g = 1.1$, $\gamma_f = 0.001$, $\gamma_g = 0.001$, $g_{li} = 0.1$, $g_{ui} = 5$, $M_f = 10$, $\alpha = 1$, $\underline{g} = 0.1$.

Les conditions initiales choisies sont : $x_1(0) = 0.5$, $x_2(0) = 0$, $\theta_f(0) = 0$,$\theta_g(0) = 0.5$, $\hat{\varepsilon}_f = 0$, $\hat{\varepsilon}_g = 0$.

Les résultats de simulation sont donnés respectivement Figures 2.4 et 2.5. Le signal de commande u est montré dans Figure 2.6.

On remarque que les trajectoires réelles convergentes vers les trajectoires désirées.

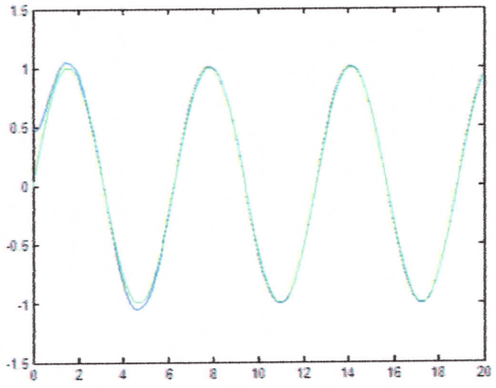


Figure 2.4: Premier état du systeme ; réelle (trait bleu), consigne (trait vert).

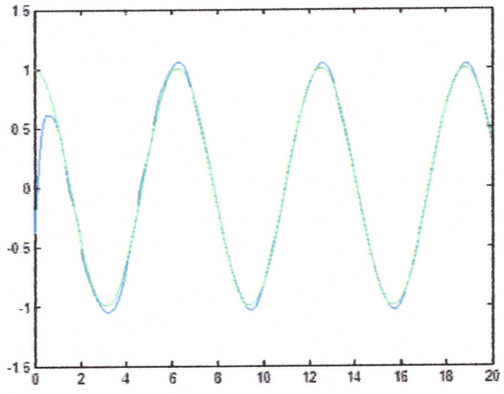


Figure 2.5: Deuxième état du systeme ; réelle (trait bleu), consigne (trait vert).

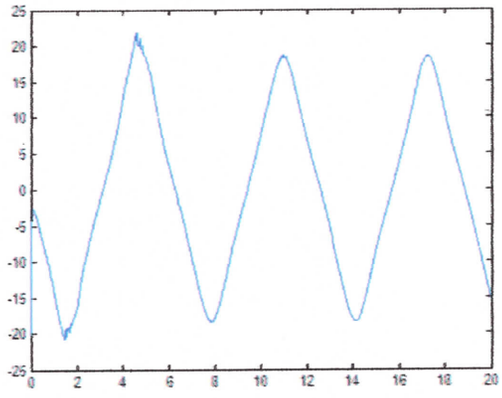


Figure 2.6: signal de commande u.

2.7 Conclusion

Dans ce chapitre, nous avons donné un aperçu sur la stabilité au sens de Lyapunov qui va être utilisée dans la preuve de stabilité des schémas de commande dans ce travail.

En deuxième lieu, nous avons définis la commande adaptative et l'utilisation du flou dans cette commande, pour développer par la suite une approche de commande adaptative floue indirecte pour une classe des systèmes SISO où les systèmes flous sont utilisés pour identifier en ligne la dynamique inconnue du système. Dans cette approche, on a proposés une loi de commande augmentée d'un terme de commande robuste afin de compenser les erreurs d'approximations floues. Pour éviter le problème de singularité de la loi de commande on a utilisé un algorithme de projection pour l'adaptation des paramètres du système flous.

Les résultats de la simulation effectuée sur un système SISO non linéaire montrent les bonnes performances de poursuite de cette méthode.

CHAPTER 3

Commande adaptative indirecte floue avec saturation

Sommaire

3.1 Introduction **30**
3.2 La saturation **31**
3.3 Position du problème **34**
3.3.1 Objectif 35
3.3.2 Calcul de l'erreur 36
3.3.3 Approximation des fonctions non linéaires par les systèmes flous 36
3.3.4 Loi de commande 38
3.3.5 Conception de la loi de commande proposée 39
3.3.6 Démonstration de la stabilité 40
3.4 Résultats de simulation **41**
3.4.1 Sans prise en compte de saturation 41
3.4.2 Avec solution de saturation 44
3.4.3 Conclusion 45

3.1 Introduction

La saturation est une caractéristique statique et sans mémoire, elle est inévitable dans la plupart des actionneurs. Les catégories des non-linéarités de type saturation incluent les contraintes sur l'amplitude et la vitesse des entrées de l'actionneur. Ces contraintes peuvent être dues à des restrictions délibérément placées sur les actionneurs pour éviter d'endommager le système et/ou des limitations physiques sur les actionneurs eux-mêmes. Dû à son caractère non linéaire, à ses paramètres inconnus et à la non mesurabilité de la sortie de l'actionneur, la conception d'une loi de commande adaptative pour un système non linéaire avec une saturation à l'entrée constitue un vrai défi.

Autant prendre en considération le problème de saturation de vitesse et d'amplitude des actionneurs qui surgie souvent dans la pratique, et elle est à l'origine des dégradations de performance du système, des dégradations qui peuvent aller jusqu'à

l'instabilité. Il est à noter que ce modèle de commande où l'adaptation est en ligne et continue lors de la saturation des signaux de commande, rend les performances plus vulnérables, par conséquent, il faut prendre en compte la saturation dans l'analyse et la synthèse de la loi de commande adaptative.

Dans ce chapitre, Nous nous intéressons à la conception de contrôleur adaptatif indirecte flou d'une classe des systèmes non linéaires monovariables avec saturation de vitesse et d'amplitude à l'entrée. Le phénomène de la saturation du signal de commande est tout d'abord présenté. Puis, on passe au développement d'une loi de commande adaptative floue où le problème de saturation est résolu par modification de la loi de commande ainsi que les lois d'adaptation.

Il est à noter que les solutions proposées pour remédier au problème de la saturation sont inspirées des travaux de [24].

3.2 La saturation

Il est bien connu qu'en pratique tous les systèmes dynamiques sont soumis à des limitations sur leurs entrées dues à des contraintes physiques inhérentes au système ou bien à des limitations sur les actionneurs. Ces limitations sont modélisées par des non-linéarités de type saturation.

La saturation a été toujours un problème important pour les systèmes de commande, du fait que tous les actionneurs se saturent à un certain niveau. Lorsqu'un actionneur atteint ces limites, on dit qu'il est saturé et aucune tentative visant à accroître le signal de commande ne donne un résultat. Mathématiquement, la saturation peut être décrite par :

$$sat(u) = \begin{cases} u_{max} sign(u) & \text{Si} \quad \mid u \mid > u_{max} \\ u & \text{Si} \quad \mid u \mid \leq u_{max} \end{cases} \tag{3.1}$$

Remarque: dans la suite de ce travail la fonction de saturation $sat(u)$ sera notée $\sigma(u)$.
La figure suivante explique la fonction de saturation :

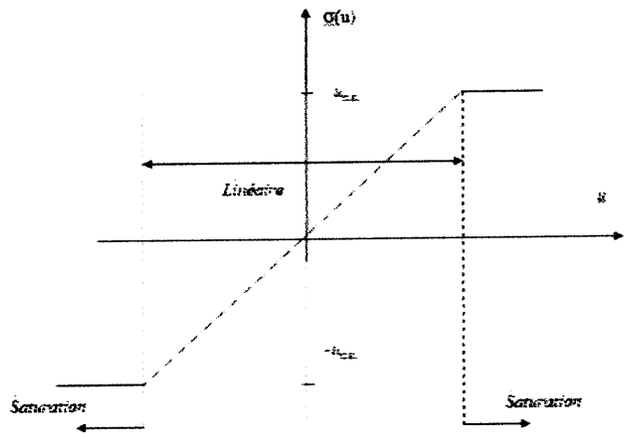


Figure 3.1: La fonction de saturation

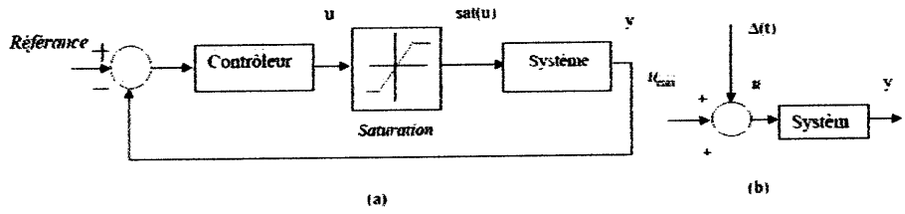


Figure 3.2: Effets de saturation[25]

Dans la littérature, il ya deux types de saturations des actionneurs :

- Saturation sur la vitesse d'actionneur : dans ce type de saturation, l'actionneur est soumis à des contraintes sur le taux de variation de la commande [21] :

$$-\Delta u_{max} \leq \Delta u(t) \leq \Delta u_{max} \tag{3.2}$$

- Saturation sur l'amplitude d'actionneur : dans ce cas, l'actionneur est soumis à une saturation sur l'amplitude :

$$-u_{max} \leq u(t) \leq u_{max} \tag{3.3}$$

La saturation d'amplitude et de vitesse peut être obtenue a l'aide du filtre représenté par la figure suivante :

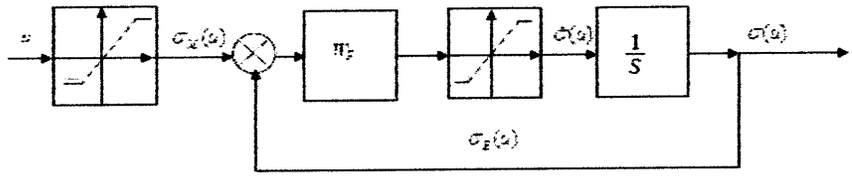


Figure 3.3: Saturation d'amplitude et de vitesse.

Mathématiquement la dérivé de $\sigma(u) = sat(u)$ peut s'écrit comme suit :

$$\dot{\sigma}(u) = \sigma_R W_p(\sigma_M(u) - \sigma(u)) \tag{3.4}$$

Où

$$\sigma_{R,M}(u) = \begin{cases} u_{Rmax.Mmax} sign(u) & \text{Si} \quad \mid u \mid \geq u_{Rmax.Mmax} \\ u & \text{Si} \quad \mid u \mid < u_{Rmax.Mmax} \end{cases} \tag{3.5}$$

avec u_{Rmax} et u_{Mmax} sont respectivement les limites de saturation de vitesse et d'amplitude et W_p est une constante.

Effets et solutions: Les effets indésirables qui peuvent causer la saturation sur les systèmes de commande sont la dégradation des performances de commande comme les grands dépassements et les cycles limite. Dans le pire des cas, elle peut causer une instabilité du schéma de commande. Dans la figure 3.2, quand la saturation se produit on remarque que le système est entraîné en boucle ouverte par le signal de commande saturé jusqu'à ce que ce dernier retourne de nouveau dans la zone linéaire non saturée. Alors, au cours de cette période (saturation), il n'y a pas d'action de commande de plus si on a une saturation de vitesse alors plus que le système est entrainé en boucle ouverte il a une limitation sur le taux de variation de signal de commande ce qui dégrade d'avantage les performances et l'erreur augmente, ce qui implique que la saturation de l'entrée pour un système instable produirait une grande défaillance du schéma de commande, même pour les systèmes stables, elle peut causer des dépassements ou encore l'instabilité sous des grandes perturbations [25].

Le problème de saturation s'avère plus délicat dans la commande adaptative, parce que le fait de continuer l'adaptation durant la saturation de l'entrée peut causer facilement une instabilité du système. Dans les approches de commande adaptatives, l'adaptation des paramètres s'effectue par l'utilisation de l'erreur de poursuite. Dans cette erreur, on trouve la contribution des conditions initiales, des erreurs paramétriques et de la saturation du signal de commande. Alors, pour avoir une adaptation stable, le signal d'erreur à utiliser dans l'adaptation ne doit pas

contenir la composante due à la saturation du signal de commande.

L'inclusion de la partie due à la saturation va provoquer une mauvaise adaptation et peut déstabiliser le système bouclé. A partir de ce constat, les solutions utilisées pour remédier aux effets indésirables de la saturation dans le cadre d'une commande adaptative sont :

- L'arrêt de l'adaptation durant la saturation, cette méthode malgré son efficacité est souvent confrontée au problème de démonstration de la stabilité [22].
- La modification de la loi de commande de telle sorte que le signal généré par le contrôleur reste toujours dans la zone linéaire, i.e. non saturé [23], [25].
- La modification du signal d'erreur utilisé dans l'adaptation par élimination de la composante due à la saturation [24].

Dans notre travail on opte à traiter le problème de saturation de vitesse et d'amplitude par la combinaison des deux dernières méthodes où la commande est modifiée pour avoir un signal non saturé et la composante d'erreur utilisée dans les loi d'adaptation est modifiée pour assurer la stabilité.

3.3 Position du problème

Soit le système non linéaire monovariable régit par la représentation d'état suivantes:

$$\begin{cases} \dot{x}_i = x_{i+1}, i = 1, ..., n-1 \\ \dot{x}_n = f(x) + g(x)\sigma(u) \\ y = x_1 \end{cases} \tag{3.6}$$

Pour le passage d'une représentation d'état à une relation entrée sortie, on dérive n fois la sortie du système de telle sorte à faire apparaitre l'entrée on aura

$$y^{(n)} = f(x) + g(x)\sigma(u) \tag{3.7}$$

Avec:

- $x = [x_1, ..., x_n]^T \in R^n$ est le vecteur d'état supposé mesurable.
- $u \in R$ est l'entrée de commande.
- $y \in R$ est la sortie du système.
- $f(x)$ et $g(x)$ sont des fonctions non linéaires continues incertaines.
- $\sigma(u)$ est la fonction de saturation définie par (3.1),(3.4) et (3.5).

Afin de résoudre la complexité du schéma de commande avec une saturation de vitesse on essaye de simplifier l'équation (3.5) par une simple transformation en une saturation d'amplitude comme suit :

$$-u_{Rmax} \leq \dot{u}(t) \leq u_{Rmax}$$

La dérivée de u peut être réécrite comme suit :

$$-u_{Rmax} \leq \frac{u(t) - u(t-\Delta)}{\Delta} \leq u_{Rmax}$$

avec, Δ le pas d'intégration, finalement on aura :

$$-\Delta u_{Rmax} + u(t-\Delta) \leq u(t) \leq \Delta u_{Rmax} + u(t-\Delta) \tag{3.8}$$

D'après (3.8), la fonction de saturation de vitesse et d'amplitude peut se reforme de la manière suivante:

$$\sigma(u) = \begin{cases} u_p & \text{Si} \quad u \geq u_p \\ u & \text{Si } u_d < u < u_p \\ u_d & \text{Si} \quad u \leq u_d \end{cases} \tag{3.9}$$

avec

$$u_p = min(u_{Mmax}, u(t-\Delta) + \Delta u_{Rmax})$$

$$u_d = max(-u_{Mmax}, u(t-\Delta) - \Delta u_{Rmax})$$

Supposition 3.1 : $g(x) \neq 0$, borné et de signe connu. Sans perte de généralité, on suppose que $g(x) > 0$.

Supposition 3.2 : La trajectoire désirée $y_d(t)$ et ses dérivées jusqu'à l'ordre n sont connues et bornées.

3.3.1 Objectif

L'objectif est la synthèse d'une loi de commande adaptative à base des systèmes flous de telle sorte que malgré les non-linéarités inconnues du système et la saturation d'amplitude et de vitesse du signal de commande, on assure les propriétés suivantes:

1. La poursuite telle que la sortie$y(t)$ poursuit la trajectoire désirée $y_d(t)$.
2. La stabilité en boucle fermée, dans le sens que tous les signaux soient bornés.

3.3.2 Calcul de l'erreur

Définissons l'erreur de poursuite par:

$$e(t) = y_d(t) - y(t) \tag{3.10}$$

et une erreur filtrée par:

$$s = \left(\frac{d}{dt} + \lambda\right)^{n-1} e(t), \lambda > 0 \tag{3.11}$$

A partir de l'équation (3.11), on déduit que pour $s(t) = 0$, nous avons une équation différentielle linéaire dont la solution implique la convergence vers zéro de $e(t)$ et de ses dérivées jusqu'à l'ordre $n-1$ [2]. De plus, la bornitude du signal $s(t)$ par une constante Φ , entraîne la bornitude de l'erreur $e(t)$ et de ses dérivées par [2] :

$$|e^{(j)}(t)| \leq 2^j \lambda^{j-n+1} \Phi, j = 0, ..., n-1 \tag{3.12}$$

De ce fait, l'objectif de la commande se simplifie à la synthèse d'un contrôleur garantissant la convergence vers zéro du signal $s(t)$.
La dérivée de (3.11) peut être écrite sous la forme:

$$\dot{s} = v - f(x) - g(x)\sigma(u) \tag{3.13}$$

avec

$$v = y_d^{(n)} + \alpha_{n-1} e^{(n-1)} + ... + \alpha_1 \dot{e} \tag{3.14}$$

où

$$\alpha_j = \frac{(n-1)!}{(n-j)!(j-1)!} \lambda^{n-1}, j = 1, ..., n-1 \tag{3.15}$$

3.3.3 Approximation des fonctions non linéaires par les systèmes flous

$\hat{f}(x, \theta_f)$ et $\hat{g}(x, \theta_g)$ sont les approximation floue des deux fonctions non linéaires $f(x)$ et $g(x)$, l'approximation floue est la seule information supposée disponible lors de la synthèse de la commande. $\hat{f}(x, \theta_f)$ et $\hat{g}(x, \theta_g)$ sont représentées par des systèmes flous du type Takagi-Sugeno à conclusion constante.

Pour synthétiser la commande on estime d'abord les fonctions $f(x)$ et $g(x)$ par les systèmes flous $\hat{f}(x, \theta_f)$ et $\hat{g}(x, \theta_g)$ de la même forme que le chapitre précédent :

$$\hat{f}(x, \theta_f) = \omega_f^T(x)\theta_f \tag{3.16}$$

$$\hat{g}(x, \theta_g) = \omega_g^T(x)\theta_g \tag{3.17}$$

avec:
• $\omega_f(x)$ et $\omega_g(x)$ sont des vecteurs de fonctions floues de base supposés convenablement fixés en avance par l'utilisateur.
• θ_f et θ_g sont les vecteurs des paramètres à estimer.

En plus de l'approchement flou, on peut définir les valeurs initiales des vecteurs des paramètres θ_f et θ_g en utilisant les connaissances prédéfinit par l'expert de telle que $\theta_f(0)$ et $\theta_g(0)$ seront respectivement proche des vecteurs optimaux θ_f^* et θ_g^* , ces derniers sont introduits uniquement pour faire l'étude théorique de la stabilité de l'algorithme de commande et la connaissance de leurs valeurs n'est pas nécessaire pour l'implantation des lois de commande adaptatives.

avec:

$$\theta_f^* = arg \min_{\theta_f}\{\sup_x |f(x) - \hat{f}(x, \theta_f)|\} \tag{3.18}$$

$$\theta_g^* = arg \min_{\theta_g}\{\sup_x |g(x) - \hat{g}(x, \theta_g)|\} \tag{3.19}$$

Les fonctions réelles $f(x)$ et $g(x)$ peuvent êtres exprimées en fonction des approximations floues de la manière suivante :

$$f(x) = \hat{f}(x, \theta_f^*) + \varepsilon_f(x) \tag{3.20}$$

$$g(x) = \hat{g}(x, \theta_g^*) + \varepsilon_g(x) \tag{3.21}$$

avec $\varepsilon_f(x)$ et $\varepsilon_g(x)$ sont les erreurs d'approximations floues.

Supposition 3.3 : les erreurs d'approximations sont bornées par des constantes positives inconnues.

$$|\varepsilon_f(x)| < \bar{\varepsilon}_f, |\varepsilon_g(x)| < \bar{\varepsilon}_g$$

A partir de cette supposition et du fait que le signal de commande généré par l'actionneur est borné on peut faire la supposition suivante :

$$|\bar{\varepsilon}_f + \bar{\varepsilon}_g u| \leq \bar{\varepsilon}$$

avec $\bar{\varepsilon}$ constate positive connue.

Cette supposition est raisonnable puisque nous supposons que les systèmes flous utilisés pour l'approximation des fonctions inconnues possèdent la propriété d'approximation universel.

Du fait que les vecteurs des paramètres sont ajustés en ligne par des lois d'adaptation convenablement choisis comme on va le voir dans la suite de ce chapitre , des vecteurs des erreurs paramétriques peuvent être décrits comme suit :

$$\tilde{\theta}_f = \theta_f^* - \theta_f \tag{3.22}$$

$$\tilde{\theta}_g = \theta_g^* - \theta_g \tag{3.23}$$

et qui vont être utilisés pour déterminer l'erreur entre les fonctions $f(x)$ et $g(x)$ et leurs approximations floues $\hat{f}(x, \theta_f)$ et $\hat{g}(x, \theta_g)$:

$$f(x) - \hat{f}(x, \theta_f) = w_f^T(x)\tilde{\theta}_f + \varepsilon_f(x) \tag{3.24}$$

$$g(x) - \hat{g}(x, \theta_g) = w_g^T(x)\tilde{\theta}_g + \varepsilon_g(x) \tag{3.25}$$

3.3.4 Loi de commande

Dans notre cas le système est soumi à une saturation à l'entrée (saturation de vitesse et d'amplitude) plus la méconnaissance des non linéarités $f(x)$ et $g(x)$, dans ce cas la mise en place d'une loi de commande ne peut être faite par les méthodes ordinaire de linéarisations (linéarisation par bouclage [28]), pour les non linéarités inconnues on adopte le remplacement des fonctions $f(x)$ et $g(x)$ par leurs approximations flous $\hat{f}(x, \theta_f)$ et $\hat{g}(x, \theta_g)$, et on écrit :

$$u = \frac{1}{\hat{g}(x, \theta_g)} \left(-\hat{f}(x, \theta_f) + v + ks \right), k > 0 \tag{3.26}$$

Malheureusement, une telle loi de commande ne peut pas assurer la stabilité et la robustesse du schéma de commande que sous ces trois conditions:

1. L'erreur d'approximation floue est nulle, cette condition n'est pas satisfaite sauf par l'augmentation du nombre de règles floues utilisées. En pratique, le nombre de règles est toujours limité.

2. L'approximation floue $\hat{g}(x, \theta_g)$ de la fonction $g(x)$ ne passe plus par zéro durant le processus d'adaptation du vecteur des paramètres θ_g , afin d'éviter la singularité de la loi de commande.

3. L'absence de la saturation du signal de commande.

la loi de commande (3.26) est à modifier pour assurer la stabilité et la robustesse du système bouclé en présence de ces problèmes. Pour résoudre ces problème nous proposons les solutions suivantes :

1. l'augmentation de la loi de commande par un terme de robustification capable d'atténuer l'effet des erreurs d'approximations floues sur les performances de commande.

2. Pour le problème de singularité de la commande, on opte pour la modification des lois d'adaptation par l'utilisation d'un algorithme d'adaptation avec projection [29].

3. Finalement pour le problème de saturation qui ne peut être négligé, nous proposons une solution qui est en faite une combinaison de deux solutions existantes en littérature:

- la modification de la loi de commande pour générer un signal de commande non saturé.
- la modification du signal d'erreur utilisé dans l'adaptation des paramètres libres des systèmes flous par l'isolation de la composante de l'erreur due à la saturation.

3.3.5 Conception de la loi de commande proposée

Nous allons avoir dans ce qui suit la conception d'une loi de commande adaptative floue robuste qui aboutie à la stabilité, en utilisant les solutions données ci-dessus pour la singularité et les erreurs d'approximations plus la combinaisons des deux solutions proposées pour la saturation (modifiant la loi de commande pour que le signal généré par le contrôleur ne soit jamais saturé et reste dans la zone linéaire, plus la modification du signal d'erreur). Nous proposons la loi de commande suivante :

$$u = \frac{1}{\hat{g}(x,\theta_g)}\Big(-\hat{f}(x,\theta_f) + v + ks + u_r + u_{sat}\Big) \tag{3.27}$$

avec u_{sat}terme additive assure la borgnitude du signal de commande définie comme suit :

$$u_{sat} = \begin{cases} (\hat{g}(x,\theta_g))(u_p - u_n) & Si \quad u_n > u_p \\ 0 & Si \quad u_d < u_n < u_p \\ (\hat{g}(x,\theta_g))(u_d - u_n) & Si \quad u_n < u_d \end{cases} \tag{3.28}$$

Où

$$u_n = \frac{1}{\hat{g}(x,\theta_g)}\Big(-\hat{f}(x,\theta_f) + v + ks + u_r\Big) \tag{3.29}$$

u_r terme de robustification conçu dans le but d'éliminer l'effet des erreurs d'approximation floues. Ce terme sera définit dans la suite du chapitre.

Afin d'assurer la stabilité de meilleurs performances en présence de saturation, on introduit « z » une nouvelle grandeur qui présente la version filtrée de l'effet de saturation ou de la composante de l'erreur due à la saturation [25] tel que :

$$\dot{z} = -kz - u_{sat} \tag{3.30}$$

Puis on utilise la grandeur χ comme nouveau signal d'erreur qui sera par la suite utilisé dans l'adaptation et porte une information sur l'erreur par « s » et sur la saturation par « z », soit :

$$\bar{\chi} = s - z \tag{3.31}$$

on utilise toujours la même forme des algorithmes de projection utilisé dans le chapitre précédent pour l'estimation des paramètres θ_f et θ_g des systèmes flous, afin de garantir la bornitude des paramètres estimés et, bien sûr, la non singularité de la loi de commande. Les lois d'adaptation utilisées sont :

$$\dot{\theta}_f = -\eta_f \omega_f(x)\bar{\chi} - \Phi_f \tag{3.32}$$

$$\dot{\theta}_g = -\eta_g \omega_g(x) u \bar{\chi} + \Phi_g \tag{3.33}$$

avec $\eta_f > 0$ le gain d'adaptation et Φ_f est définie comme suit :

$$\Phi_f = \begin{cases} 0 & \text{Si} \quad ||\theta_f|| < M_f \\ \eta_f \rho_f \frac{|\omega_f^T(x)\theta_f \bar{\chi}|}{\theta_f^T \theta_f} \theta_f & \text{Sinon} \end{cases} \tag{3.34}$$

avec $\rho_f \geq 1$.Et $\eta_g > 0$ le gain d'adaptation et Φ_g est définie comme suit :

$$\Phi_{gi} = \begin{cases} 0 & \text{Si} \quad g_{li} < \theta_{gi} < g_{ui} \\ \eta_g \rho_g sgn(\tilde{\theta}_{gi}) |\omega_{gi}(x) u \bar{\chi}| & \text{Sinon} \end{cases} \tag{3.35}$$

avec $\rho_g \geq 1$.
La borne des erreurs d'approximation est ajustée par :

$$\dot{\hat{\varepsilon}} = \gamma |\bar{\chi}|, \gamma > 0 \tag{3.36}$$

finalement le terme robuste peut être formuler comme suit:

$$u_r = \hat{\varepsilon} sign(\bar{\chi}) \tag{3.37}$$

3.3.6 Démonstration de la stabilité

Pour la démonstration de la stabilité de la loi de commande proposée dans (3.27), nous choisissons la fonction positive de Lyapunov suivante :

$$V = \frac{1}{2}\bar{\chi}^2 + \frac{1}{2\eta_f}\tilde{\theta}_f^T \tilde{\theta}_f + \frac{1}{2\eta_g}\tilde{\theta}_g^T \tilde{\theta}_g + \frac{1}{2\gamma}\tilde{\varepsilon}^2 \tag{3.38}$$

avec $\tilde{\varepsilon} = \bar{\varepsilon} - \hat{\varepsilon}$.
En dérivant cette fonction par rapport au temps, on obtient :

$$\dot{V} = \bar{\chi}\dot{\bar{\chi}} - \frac{1}{\eta_f}\tilde{\theta}_f^T \dot{\theta}_f - \frac{1}{\eta_g}\tilde{\theta}_g^T \dot{\theta}_g - \frac{1}{\gamma}\tilde{\varepsilon}\dot{\hat{\varepsilon}} \tag{3.39}$$

La dynamique de $\bar{\varepsilon}$ est donnée par :

$$\dot{\bar{\chi}} = \dot{s} - \dot{z} \tag{3.40}$$

En utilisant (3.13) et (3.30) l'équation (3.40) peut être réécrite comme suit :

$$\dot{\bar{\chi}} = kz + u_{sat} + v - f(x) - g(x)\sigma(u) \tag{3.41}$$

Du fait qu'on utilise dans la loi de commande un terme additif qui assure que le signal de commande reste non saturé on peut dire que $\sigma(u) = u$ alors l'équation (3.41) peut se réécrire :

$$\dot{\bar{\chi}} = kz + u_{sat} + v - f(x) - g(x)(u) \tag{3.42}$$

On ajoutant et on retranchant le terme $\hat{g}(x,\theta_g)u$ on aura :

$$\dot{\bar{\chi}} \equiv k\bar{z} + u_{sat} + v - f(x) - (g(x) - \hat{g}(x,\theta_g))u - \hat{g}(x,\theta_g)u \tag{3.43}$$

En remplaçant (3.27), (3.24) et (3.25) dans (3.43) on obtient :

$$\dot{\bar{\chi}} = -k\bar{\chi} + u_r - \omega_f^T(x)\tilde{\theta}_f - \omega_g^T(x)\tilde{\theta}_g u - \varepsilon(x) \tag{3.44}$$

avec $\varepsilon(x) = \varepsilon_f(x) + \varepsilon_g(x)u$.
ainsi, (3.39) devient :

$$\dot{V} = -k\bar{\chi}^2 + u_r\bar{\chi} - \varepsilon(x)\bar{\chi} - \frac{1}{\gamma}\tilde{\varepsilon}\dot{\hat{\varepsilon}} + \dot{V}_\theta \tag{3.45}$$

avec

$$\dot{V}_\theta = -\tilde{\theta}_f^T\left[\omega_f(x)\bar{\chi} + \frac{1}{\eta_f}\dot{\theta}_f\right] - \tilde{\theta}_g^T\left[\omega_g(x)\bar{\chi}u + \frac{1}{\eta_g}\dot{\theta}_g\right] \tag{3.46}$$

Si on introduit les lois d'adaptation données par (3.32) et (3.33) dans l'expression de $\dot{V}_\theta$, on trouve (voir démonstration chapitre 2) :

$$\dot{V}_\theta \leq 0 \tag{3.47}$$

Alors $\dot{V}$ devient :

$$\dot{V} \leq -k\bar{\chi}^2 - u_r\bar{\chi} - \varepsilon(x)\bar{\chi} - \frac{1}{\gamma}\tilde{\varepsilon}\dot{\hat{\varepsilon}} \tag{3.48}$$

On remplace (3.36) dans (3.48) on obtient :

$$\dot{V} \leq -k\bar{\chi}^2 - u_r\bar{\chi} - \varepsilon(x)\bar{\chi} - \tilde{\varepsilon}|\bar{\chi}| \tag{3.49}$$

De (3.37) et la supposition 3.3, $\dot{V}$ peut être bornée comme suit :

$$\dot{V} \leq -k\bar{\chi}^2 - \hat{\varepsilon}|\bar{\chi}| + \bar{\varepsilon}|\bar{\chi}| - \bar{\varepsilon}|\bar{\chi}| + \hat{\varepsilon}|\bar{\chi}| \tag{3.50}$$

Alors:

$$\dot{V} \leq -k\bar{\chi}^2 \tag{3.51}$$

De (3.51) et via le lemme de Barbalat [2], on peut conclure que $\chi(t) \to 0$ quand $t \to 0$ .Si la poursuite de la trajectoire de référence est réalisée par un signal de commande non saturé i.e. $\chi(t) \to s(t)$ lorsque $t \to \infty$. Dans ce cas, l'erreur de poursuite et ses dérivées jusqu'à l'ordre $n-1$ convergent vers zéro quand $t \to \infty$.

3.4 Résultats de simulation

3.4.1 Sans prise en compte de saturation

Nous appliquons maintenant le schéma de commande proposé sur un système non linéaire SISO :

$$\begin{cases} \dot{x}_1 = x_2 \\ \dot{x}_2 = f(x) + g(x)\sigma(u) \\ y = x_1 \end{cases} \tag{3.52}$$

avec :

$$f(x) = 1.5(1 - x_1^2)x_2 - x_1 \tag{3.53}$$

$$g(x) = 1 + 0.5\cos(x_1 x_2) \tag{3.54}$$

Les fonctions non linéaires (3.53) et (3.54) sont supposées inconnues. Chaque fonction est approximée par un système flou constitué d'une collection de règles floues de la forme «Si-alors» ayant comme entrée, le vecteur d'état $x = [x_1, x_2]$. Pour chaque variable x_i , $i = 1, 2$,on définit cinq fonctions d'appartenance gaussiennes de la forme (1.8) uniformément distribuées entre ± 1 . Ces fonctions d'appartenance gaussiennes sont définit comme suit:

$$\mu_{F_i^1}(x_i) = exp\left\{-\frac{1}{2}\left(\frac{x_i+1}{0.25}\right)^2\right\}$$
$$\mu_{F_i^2}(x_i) = exp\left\{-\frac{1}{2}\left(\frac{x_i+0.5}{0.25}\right)^2\right\}$$
$$\mu_{F_i^3}(x_i) = exp\left\{-\frac{1}{2}\left(\frac{x_i}{0.25}\right)^2\right\}$$
$$\mu_{F_i^4}(x_i) = exp\left\{-\frac{1}{2}\left(\frac{x_i-0.5}{0.25}\right)^2\right\}$$
$$\mu_{F_i^5}(x_i) = exp\left\{-\frac{1}{2}\left(\frac{x_i-1}{0.25}\right)^2\right\}, i = 1, 2$$

Pour illustrer les effets de saturation de vitesse et d'amplitude, nous considérons de nouveau la commande en poursuite vue dans le chapitre précédent ou le contrôleur est conçu sans considération de cette dernière et on place une saturation avec les bornes $u_{Rmax} = 2$ et $u_{Mmax} = 1$. Les résultats de simulation sont illustrés sur les figures ci-dessous. Les Figures (3.4) et (3.5) présentent respectivement la position et la vitesse du système avec leurs trajectoires de références. La Figure 3.6 présente le signal de commande u.

D'après ces figures, on constate que les trajectoires réelles ne convergent pas vers les trajectoires désirées à cause d'une saturation de vitesse et d'amplitude qui n'est pas prise en compte, d'où on voit bien le besoin de la prise en considération de ce phénomène lors de calcul de contrôleur.

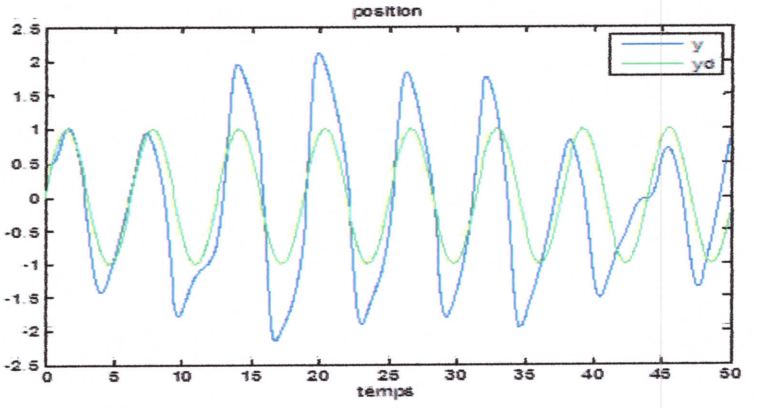
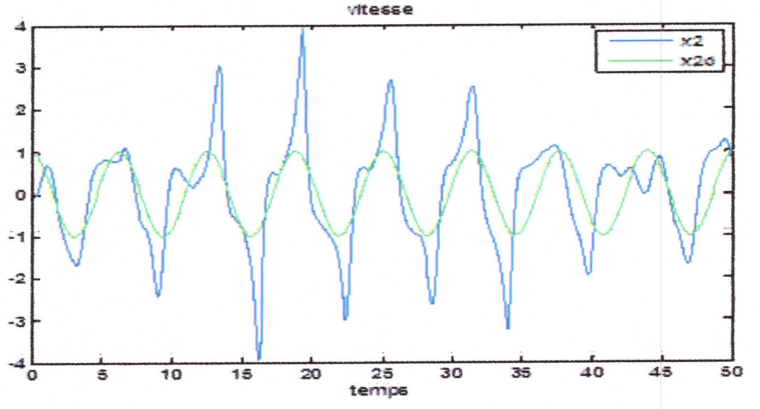


Figure 3.4: Premier état du systeme ; réelle (trait bleu), consigne (trait vert).



مكتبة الكلية

Figure 3.5: Deuxième état du systeme ; réelle (trait bleu), consigne (trait vert).

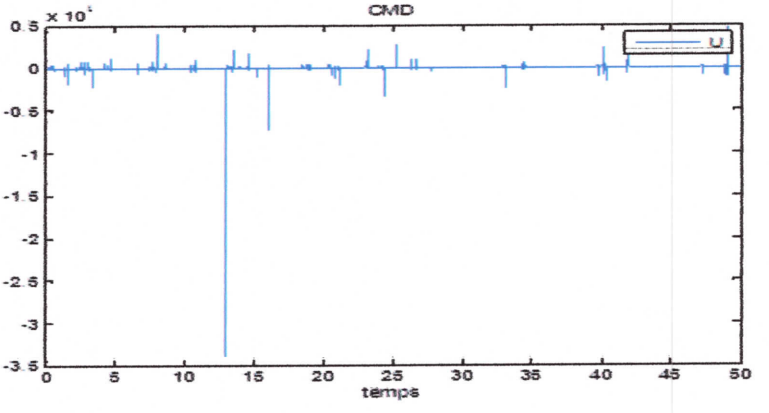


Figure 3.6: signal de commande ; u.

3.4.2 Avec solution de saturation

Pour tester la validité du schéma de commande proposé, nous considérons de nouveau le système non linéaire défini par les équations (2.52)-(2.54).

Notre objectif est la poursuite d'une trajectoire de référence $y_d(t) \equiv sin(t)$. Pour l'approximation des fonctions inconnues on utilise les mêmes systèmes flous définis dans la simulation précédente avec les mêmes entrées et les mêmes caractéristiques. Les paramètres de synthèse sont choisis comme suit : $\lambda = 10$, $k = 5$, $\eta_f = 50$, $\eta_g = 50$, $\rho_f = 1.1$, $\rho_g = 1.1$, $\gamma = 0.000001$, $g_{li} = 0.1$, $g_{ui} = 5$, $M_f = 10$,$W_p = 200$ et $\alpha = 1$.

La fonction de saturation de vitesse et d'amplitude sont caractérisées respectivement par : $u_{Rmax} = 2$ et $u_{Mmax} = 1$. Les conditions initiales choisies sont : $x_1(0) = 0.5$, $x_2(0) = 0$, $\theta_f(0) = 0$,$\theta_g(0) = 0.5$, $\hat{\varepsilon}(0) = 0$, et $z(0) = 0$.

Les résultats de simulation obtenus sont illustrés sur les figures données ci-dessous. Les Figures 3.7 et 3.8 montrent respectivement la position et la vitesse du système avec leurs trajectoires à suivre, la Figure 3.9 représente la commande floue adaptative est le signal de commande appliqué au système.

De ces figures, on remarque que les trajectoires réelles convergent vers les trajectoires de références. On peut par ailleurs remarquer que le signal de commande généré par le contrôleur est toujours maintenu entre les limites de saturation.

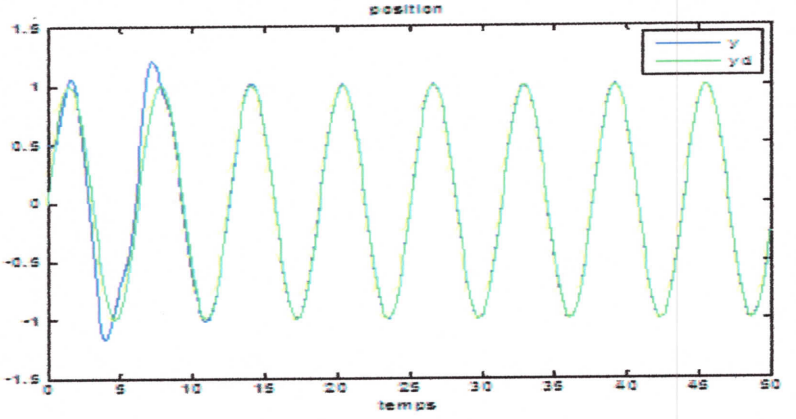


Figure 3.7: Premier état du systeme ; réelle (trait bleu), consigne (trait vert).

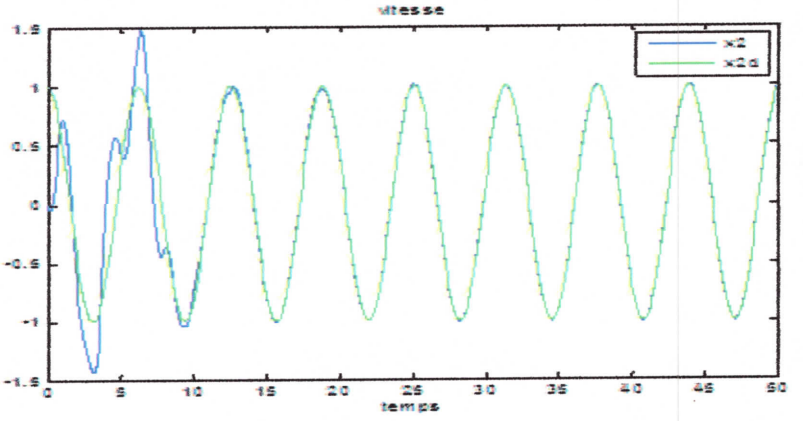


Figure 3.8: Deuxièeme état du systeme ; réelle (trait bleu), consigne (trait vert).

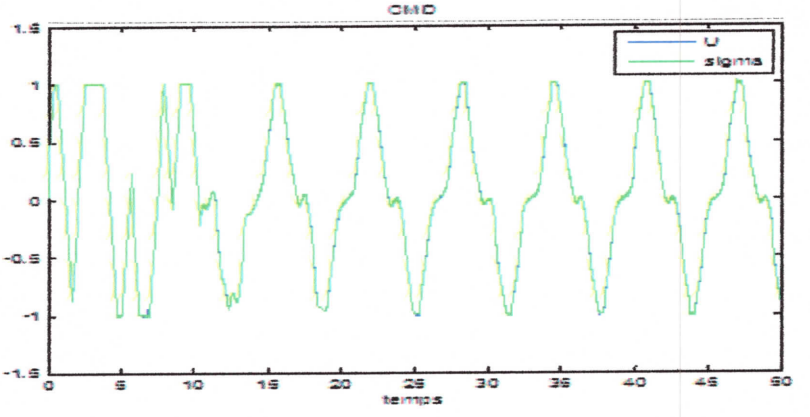


Figure 3.9: signaux de commande ; u (trait bleu),$\sigma(u)$(trait vert).

3.4.3 Conclusion

Dans ce chapitre nous avons posés le problème de saturation et nous avons présentés quelques effets et quelques solutions pour développer par la suite une approche de commande adaptative floue indirecte pour une classe des systèmes non linéaires monovariables avec saturation de vitesse et d'amplitude de l'entrée.

La loi de commande utilisée est celle déduite par la technique de linéarisation par bouclage, telle que les dynamiques inconnues du système sont remplacées par leurs estimations floues le terme additif est conçu pour éliminer les erreurs d'approximation floue, et les paramètres des fonctions floues approchées sont calculés par les algorithmes de projections dans le but d'éviter la singularité de la loi de commande.

Le problème de saturation est traité par la combinaison de la modification de la loi de commande et le signal d'erreur de la loi d'adaptation.

Une simulation numérique est présentée pour démontrer les effets de saturation et par la suite l'efficacité de la solution proposée.

CHAPTER 4

Commande adaptative directe floue avec saturation

Sommaire

4.1 Introduction **47**
4.2 Position du problème **48**
4.2.1 Objectif 49
4.2.2 Calcule de l'erreur 49
4.2.3 Conception de la loi de commande adaptative directe 50
4.2.4 Démonstration de la stabilité 52
4.3 Résultats de simulation **53**
4.3.1 Sans prise en compte de saturation 53
4.3.2 Avec solution de saturation 55
4.4 Conclusion **56**

4.1 Introduction

Dans le chapitre précèdent, nous avons vu que le terme $\hat{g}^{-1}(x, \theta_g)$ pose de problème lorsque $\hat{g}(x, \theta_g) \to 0$ car le signal de commande devient dans ce cas très important et peut ainsi déstabiliser le système en boucle fermée.

Dans ce chapitre, les systèmes flous seront utilisés pour approcher la loi de commande désirée dans son ensemble au lieu des fonctions non linéaires inconnues de façon séparée. Par conséquent,le terme $\hat{g}^{-1}(x, \theta_g)$ n'apparaîtra pas explicitement dans la loi de commande et ainsi le problème de division par zéro est évité. Cette méthode est appelée commande adaptative directe parce que les systèmes flous sont utilisés pour approcher directement une loi de commande stabilisatrice dite idéale.

Dans ce contexte, plusieurs travaux ont été développés[32] mais sans prise en charge du problème de la saturation du signal d'entrée.

Dans ce chapitre, nous présentons une approche de commande adaptative directe floue pour une classe des systèmes non linéaires monovariables avec saturation

d'amplitude et de vitesse de l'entrée. Dans cette approche, le problème de la saturation est résolu par la modification du signal d'erreur utilisé dans la loi d'adaptation des paramètres ajustables. La modification est réalisée de sorte à effectuer une adaptation stable par élimination de l'effet de la composante de l'erreur due à la saturation du signal de commande sur le processus d'adaptation paramétrique. La stabilité de cette approche proposée est analysée par la méthode directe de Lyapunov.

Des résultats de simulation sont donnés pour valider l'approche proposée.

4.2 Position du problème

Considérons la classe des systèmes non linéaires monovariables avec saturation l'entrée décrite par les équations différentielles suivantes :

$$\begin{cases} \dot{x}_i = x_{i+1}, i = 1, ..., n-1 \\ \dot{x}_n = f(x) + g(x)\sigma(u) \\ y = x_1 \end{cases} \tag{4.1}$$

où encore par une représentation entrée-sortie de la forme :

$$y^{(n)} = f(x) + g(x)\sigma(u) \tag{4.2}$$

Avec:

- $x = [x_1, ..., x_n]^T \in R^n$ est le vecteur d'état supposé mesurable.
- $u \in R$ est l'entrée de commande.
- $y \in R$ est la sortie du système.
- $f(x)$ et $g(x)$ sont des fonctions non linéaires continues incertaines.
- $\sigma(u)$ est la fonction de saturation d'amplitude et de vitesse définie dans le chapitre précédant (3.4) et (3.5) et la figure 3.3.

Par la suite, les suppositions suivantes sont nécessaires.

Supposition 4.1 : $g(x) \neq 0$,borné et de signe connu. Sans perte de généralité, on suppose que $g(x) > 0$.

Supposition 4.2 : Le vecteur d'état x est supposé mesurable.

Supposition 4.3 : La trajectoire désirée $y_d(t)$ et ses dérivées jusqu'à l'ordre n sont connues, continues et bornées.

4.2.1 Objectif

L'objectif est la synthèse d'une loi de commande adaptative à base des systèmes flous de telle sorte que malgré les non-linéarités inconnues du système et la saturation du signal de commande, on assure les propriétés suivantes :

1. la sortie $y(t)$ suit la trajectoire désirée $y_d(t)$ si la poursuite est réalisable.
2. La stabilité en boucle fermée, dans le sens que tous les signaux soient bornés.

4.2.2 Calcule de l'erreur

Définissions l'erreur de poursuite par:

$$e(t) = y_d(t) - y(t) \tag{4.3}$$

et une erreur filtrée par

$$s = \left(\frac{d}{dt} + \lambda\right)^{n-1} e(t), \lambda > 0 \tag{4.4}$$

Il apparaît clairement de (4.4) que l'équation différentielle linéaire $s(t) = 0$ possède la solution unique $e(t) = 0$. De plus, la bornitude de $s(t)$ entraîne la bornitude de $e(t)$. La dérivée temporelle de (4.4) est

$$\dot{s} = v - f(x) - g(x)\sigma(u) \tag{4.5}$$

Avec:

$$v = y_d^{(n)} + \alpha_{n-1}e^{(n-1)} + ... + \alpha_1\dot{e} \tag{4.6}$$

où

$$\alpha_j = \frac{(n-1)!}{(n-j)!(j-1)!}\lambda^{n-1}, j = 1, ..., n-1 \tag{4.7}$$

Si les fonctions non linéaires $f(x)$ et $g(x)$ sont supposées connues et que la saturation est absente $(u = \sigma(u))$, les objectifs de commande peuvent être atteints en choisissant la loi de commande idéale suivante :

$$u^* = \frac{v - f(x) + ks}{g(x)}, k > 0 \tag{4.8}$$

Pour démontrer que cette loi de commande garantit les objectives de commande on commence par la fonction positive de Lyapunov suivante :

$$V = \frac{1}{2}s^2 \tag{4.9}$$

Sa dérivée temporelle est :

$$\dot{V} = s\dot{s} \tag{4.10}$$

En supposant que la saturation est absente et par substitution de (4.5) dans (4.10), on obtient :

$$\dot{V} = s(v - f(x) - g(x)u) \tag{4.11}$$

A partir de (4.8), il vient :

$$\dot{V} = -ks^2 \tag{4.12}$$

ce qui implique que $s(t) \to 0$ quand $t \to 0$. Par conséquent, l'erreur de poursuite et ses dérivées jusqu'à l'ordre $n-1$ convergent vers zéro.

Puisque les fonctions non linéaires $f(x)$ et $g(x)$ sont incertaines et la saturation de vitesse et d'amplitude est présente à l'entrée du système, l'implantation de la loi de commande idéale (4.8) est impossible. Dans cette situation, notre but est d'approcher cette loi de commande idéale en utilisant les systèmes flous.

4.2.3 Conception de la loi de commande adaptative directe

Dans cette section, notre but est la construction d'un schéma de commande adaptatif complet avec les lois d'adaptation nécessaires pour atteindre les objectifs de commande. On essaye d'approcher directement la loi de commande idéale définie dans (4.8) on utilisant les systèmes flous.

Selon la propriété d'approximation universelle des systèmes flous, la commande idéale (4.8) peut être approcher par un système flou de la forme de (1.21) comme suit :

$$u^* = \omega^T(z)\theta^* + \varepsilon(z) \tag{4.13}$$

avec:

- z vecteur d'entré du système flou (généralement, une combinaison de l'erreur de poursuite et ses dérivées).
- $\omega(z)$ est un vecteur de fonctions floues de base supposé convenablement fixer en avance par l'utilisateur.
- $\varepsilon(z)$ erreur d'approximation floue.
- θ^* vecteur des paramètres optimaux qui minimise l'erreur $\varepsilon(z)$ défini comme suit :

$$\theta^* = arg \min_{\theta}\{\sup_{z}|u^* - \omega^T(z)\theta|\} \tag{4.14}$$

Supposition 3.4 : on suppose que l'erreur d'approximation floue est arbitrairement petite.

$$\varepsilon(z) \approx 0 \tag{4.15}$$

Avec cette supposition, la loi de commande idéale peut se réécrire comme suit :

$$u^* = \omega^T(z)\theta^* \tag{4.16}$$

Notons que la valeur du vecteur des paramètres optimaux θ^* est une quantité artificielle introduite seulement pour le but d'analyse et sa valeur n'est pas exigée dans la mise en application du contrôleur. De ce fait, on doit estimer cette valeur à l'aide d'une loi d'adaptation convenablement choisie.

Soit θ le vecteur estimé de θ^* et qui sera calculé à partir d'un algorithme d'adaptation convenable. Alors, l'approximation floue de la loi de commande idéale est donnée par :

$$u = \omega^T(z)\theta \tag{4.17}$$

Dans le cas nominal où la saturation à l'entrée du système est négligée, une loi d'adaptation pour l'estimation des paramètres du système flou de la forme suivante peut être convenable :

$$\dot{\theta} = \eta g(x)\omega(z)s \tag{4.18}$$

Seulement en cas d'un signal de commande généré non saturé ou en cas d'absence de contraintes sur l'entrée de commande, Une telle loi d'adaptation peut garantir la stabilité du schéma de commande, malheureusement en pratique, ces conditions ne sont pas toujours vérifiées.

Le système reçoit à son entrée une valeur constante différente de la valeur générée par le contrôleur si la saturation et présente, dés que le signal de commande est saturé. Ceci provoque l'augmentation de l'erreur de poursuite et de l'erreur filtrée de plus si il y a une limitation sur la variation de commande l'erreur va augmenter d'avantage et le signal de commande généré ne peut pas gérer les erreurs. Etant donné que l'erreur filtrée est utilisée dans la loi d'adaptation, le système bouclé peut devenir instable à cause de la saturation.

Nous proposons la modification de l'erreur filtrée utilisée dans la loi d'adaptation (4.18) pour remédier à ce problème, de telle sorte que l'adaptation ne prendra pas en compte l'erreur due à la saturation. La nouvelle grandeur d'erreur notée est définie comme suit :

$$\bar{\chi} = s - z \tag{4.19}$$

Où z représente une vraie mesure de l'effet de la saturation sur la grandeur à contrôler, sa dynamique est donnée par l'équation suivante :

$$\dot{z} = -kz + \gamma(u - \sigma(u)) \tag{4.20}$$

avec $\bar{\gamma}$ une variable où sa dynamique est donnée par.

$$\dot{\gamma} = \eta_\gamma(u - \sigma(u))\bar{\chi} \tag{4.21}$$

avec $\eta_\gamma > 0$.
La loi d'adaptation est alors définie par la formule suivante :

$$\dot{\theta} = \eta\omega(z)\bar{\chi} \tag{4.22}$$

Cette loi d'adaptation garantit la stabilité du schéma de commande et la convergence de l'erreur de poursuite vers zéro, mais à condition que la supposition suivante soit satisfaite :

Supposition 4.5 : on suppose que la dynamique du gain de commande est lentement variable, i.e. :

$$\dot{g}(x) \approx 0 \tag{4.23}$$

4.2.4 Démonstration de la stabilité

Pour la démonstration de la stabilité de la loi de commande proposée, nous choisissons la fonction positive de Lyapunov suivante :

$$V = \frac{1}{2}\bar{\chi}^2 + \frac{g(x)}{2\eta}\tilde{\theta}^T\tilde{\theta} + \frac{1}{2\eta_\gamma}(g(x) - \gamma)^2 \tag{4.24}$$

avec $\tilde{\theta}$ est l'erreur d'approximation paramétrique définie par :

$$\tilde{\theta} = \theta^* - \theta \tag{4.25}$$

En dérivant cette fonction par rapport au temps, on obtient :

$$\dot{V} = \bar{\chi}\dot{\bar{\chi}} - \frac{g(x)}{\eta}\tilde{\theta}^T\dot{\theta} + \frac{\dot{g}(x)}{2\eta}\tilde{\theta}^T\tilde{\theta} + \frac{1}{\eta_\gamma}(\dot{g}(x) - \dot{\gamma})(g(x) - \gamma) \tag{4.26}$$

Par dérivation de (4.19), on obtient :

$$\dot{\bar{\chi}} = \dot{s} - \dot{z} \tag{4.27}$$

La substitution de (4.5) et (4.20) dans (4.27) donne :

$$\dot{\bar{\chi}} = kz + v - f(x) - g(x)\sigma(u) - \gamma(u - \sigma(u)) \tag{4.28}$$

En ajoutant et en retranchant le terme $g(x)(u^* - u)$ à la dérivée de $\bar{\chi}$, on obtient :

$$\dot{\bar{\chi}} = kz + v - f(x) - g(x)\sigma(u) - \gamma(u - \sigma(u)) + g(x)(u^* - u) - g(x)(u^* - u) \tag{4.29}$$

$$\dot{\bar{\chi}} = kz + v - f(x) - g(x)u^* + (g(x) - \gamma)(u - \sigma(u)) + g(x)(u^* - u) \tag{4.30}$$

Avec la commande idéale définie dans (4.8), (4.30) devient :

$$\dot{\bar{\chi}} = -k\bar{\chi} + (g(x) - \gamma)(u - \sigma(u)) + g(x)(u^* - u) \tag{4.31}$$

A partir de (4.16), (4.17) et (4.25), $\dot{\bar{\chi}}$ peut se récrire :

$$\dot{\bar{\chi}} = -k\bar{\chi} + (g(x) - \gamma)(u - \sigma(u)) + g(x)\omega^T(z)\tilde{\theta} \tag{4.32}$$

Remplaçons par cette équation dans la dérivée de la fonction de Lyapunov, on trouve:

$$\dot{V} = -k\bar{\chi}^2 + (g(x)-\gamma)(u-\sigma(u))\bar{\chi} + g(x)\omega^T(z)\tilde{\theta}\bar{\chi} - \frac{g(x)}{\eta}\tilde{\theta}^T\dot{\theta} + \frac{\dot{g}(x)}{2\eta}\tilde{\theta}^T\tilde{\theta} + \frac{1}{\eta_\gamma}(\dot{g}(x)-\dot{\gamma})(g(x)-\gamma) \tag{4.33}$$

En se basant sur la supposition 4.5, (4.33) devient :

$$\dot{V} = -k\bar{\chi}^2 + (g(x)-\gamma)(u-\sigma(u))\bar{\chi} + g(x)\omega^T(z)\tilde{\theta}\bar{\chi} - \frac{g(x)}{\eta}\tilde{\theta}^T\dot{\theta} - \frac{\dot{\gamma}}{\eta_\gamma}(g(x)-\gamma) \tag{4.34}$$

A partir de la loi d'adaptation (4.22) et de la dynamique de γ (4.21), on obtient :

$$\dot{V} = -k\bar{\chi}^2 \tag{4.35}$$

Nous concluons d'après le théorème de Barbalat [2] que $\bar{\chi}(t) \to 0$ quand $t \to \infty$ et que les paramètres estimés sont bornés. Si la poursuite est réalisable sans saturation, i.e., la grandeur z tend vers zéro lorsque $t \to \infty$, l'erreur filtrée et l'erreur de poursuite et ses dérivées jusqu'à l'ordre $n-1$ convergent vers zéro quand $t \to \infty$.

4.3 Résultats de simulation

4.3.1 Sans prise en compte de saturation

Nous procédons maintenant à l'application de la stratégie de commande adaptative directe proposée à la commande en poursuite du système SISO défini par les équations (3.53)-(3.55) dans la section 3.4.1.

Notre objectif est la poursuite d'une trajectoire de référence $y_d(t) = sin(t)$. Pour cela, la loi de commande idéale est approchée par un système flou constitué d'une collection de règles floues de la forme « Si-alors », et possède comme entrée le vecteur $z = [10e, 2\dot{e}]$, Pour chaque Variable z_i , $i = 1, 2$, on définit cinq fonctions d'appartenance gaussiennes de la forme (1.8) uniformément distribuées entre ± 1 .

$$\mu_{A_i^j}(z_i) = exp\left\{-\frac{1}{2}\left[\frac{z_i - c(j)}{0.25}\right]^2\right\}, i = 1, 2, j = 1, \ldots, 5$$

avec $c = [-1, -0.5, 0, 0.5, 1]$.

Pour avoir les effets de saturation on va en premier faire un teste de simulation sans solution i.e. $\gamma = 0$ et $z = 0$, ce qui implique on travail avec $\bar{\chi} = s$ erreur filtrée (cas nominal) et à l'entrée du système on vas placer une saturation d'amplitude et de vitesse utilisée dans le chapitre précédent (voir figure 3.3) avec les bornes:

$u_{Rmax} = 3$ et $u_{Mmax} = 2$ et $W_p = 200$. les conditions initiales sont choisis comme suit : $x_1(0) = 0.5$, $x_2(0) = 0$, $\theta = 0$. Les paramètres de synthèse choisis dans ce cas sont : $\lambda = 15$,k=100,$\eta = 100$.

Les résultats de simulation sont illustrés sur les figures ci-dessous. Les Figures 4.1 et 4.2 présentent respectivement la position et la vitesse du système avec leurs trajectoires de références. La Figure 4.3 présente le signal de commande u.

D'après ces figures, on constate que les trajectoires réelles ne convergent pas vers les trajectoires désirées à cause de la saturation de vitesse et d'amplitude et on voit bien la dégradation de performances si on ne prend pas en compte la saturation lors de la conception des lois de commandes.

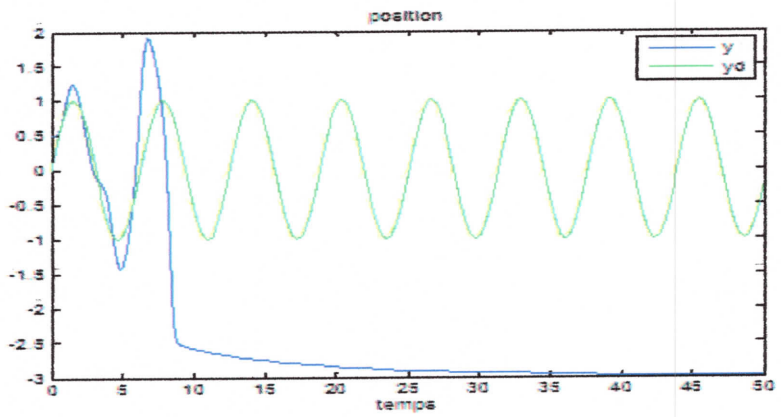


Figure 4.1: Premier état du systeme ; réelle (trait bleu), consigne (trait vert).

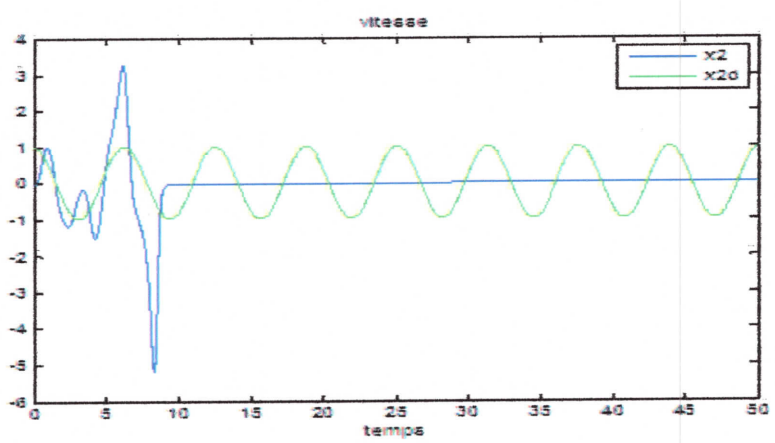


Figure 4.2: Deuxième état du systeme ; réelle (trait bleu), consigne (trait vert).

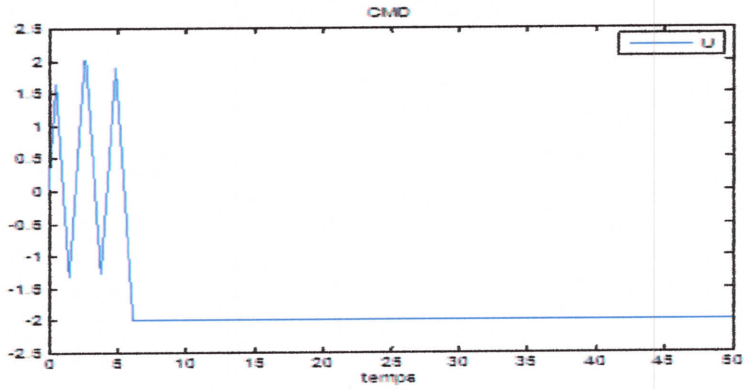


Figure 4.3: signal de commande ; u.

4.3.2 Avec solution de saturation

Pour tester les performances de l'approche de commande adaptative floue directe développée dans cette section, nous utilisons le système donné en section 4.3.1.

Notre objectif est de tester les performances de la technique développée sur un système non linéaire afin de pouvoir suivre la trajectoire désirée $y_d(t) = sin(t)$. Dans ce cas les conditions initiales sont choisies comme suit : $x_1(0) = O.5$, $x_2(0) = O$, $\theta = 0$, $\sigma(0) = 0$, $\gamma(0) = 0$ et $z(0) = 0$. les paramètres de synthèse choisis dans ce cas sont :$\lambda = 15$, $k = 100$, $\eta = 100$, $\eta_\gamma = 100$.

Les résultats de simulation sont illustrés sur les figures ci-dessous. Les figures 4.4 et 4.5 présentent respectivement l'évolution de la sortie du système $y = \theta$ et sa dérivée $\dot{y} = \dot{\theta}$ avec leurs trajectoires de référence. La figure 4.6 montre les signaux de commande u et $\sigma(u)$:

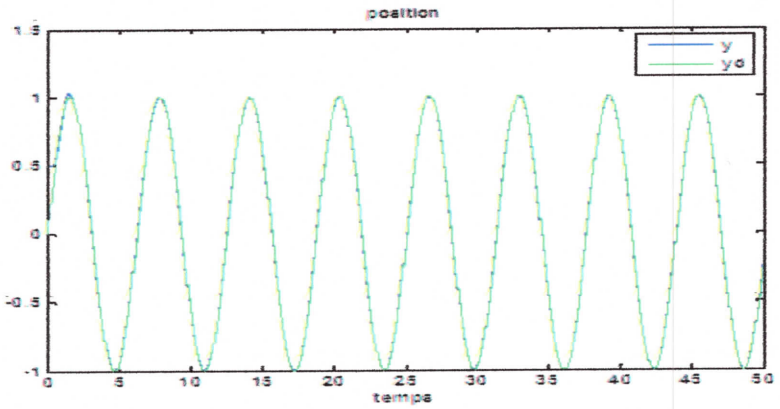


Figure 4.4: Premier état du systeme ; réelle (trait bleu), consigne (trait vert).

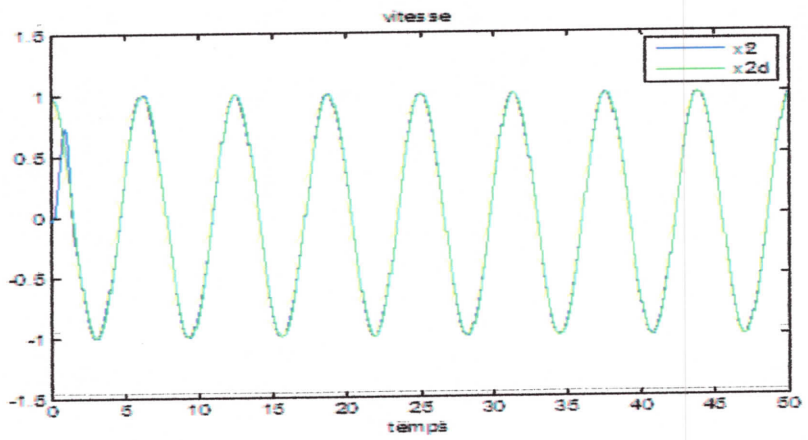


Figure 4.5: Deuxième état du systeme ; réelle (trait bleu), consigne (trait vert).

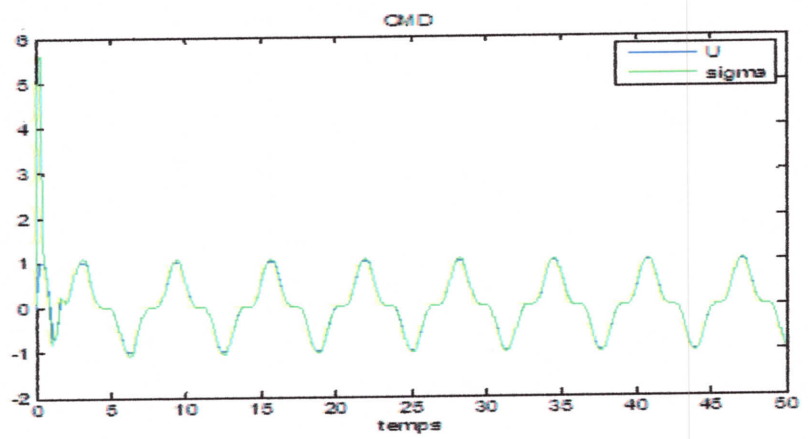


Figure 4.6: signaux de commande ; u (trait bleu),$\sigma(u)$ (trait vert).

D'après ces figures, on constate que les trajectoires réelles convergent vers les trajectoires désirées et que ce schéma de commande arrive à empêcher la saturation de se Produire.

4.4 Conclusion

Ce chapitre a été consacré au développement d'une commande adaptative floue directe par les systèmes flous pour une classe des systèmes non linéaires monovariables avec saturation de l'entrée. Les systèmes flous sont utilisés pour l'approximation directe d'une loi de commande dite idéale.

La synthèse de la loi de commande est effectuée avec la prise en compte de la saturation à l'entrée du système commandé. Nous avons proposés, comme solution, la modification du signal d'erreur utilisé dans la loi d'adaptation des paramètres des systèmes flous afin de réaliser une adaptation paramétrique stable par l'isolation de l'effet de la saturation.

Un schéma de commande est proposé, il exige un gain de commande lentement variable. Ceci limite la classe des systèmes concernés par cette approche.

Cette approche est validée par une simulation effectuée sur le modèle d'un système SISO.

Conclusion générale

Le travail présenté dans ce mémoire concerne le développement de stratégies de commande adaptatives utilisant les systèmes flous pour les systèmes non linéaires de type SISO avec des non linéarités inconnues et en présence de saturation de vitesse et d'amplitude.

Nous nous sommes intéressés particulièrement aux systèmes non linéaires dits linéarisables par bouclage. Le facteur commun de toutes les techniques de commande proposées est l'utilisation des systèmes flous pour l'approximation des dynamiques inconnues des systèmes dans certains cas (approches indirectes), et pour l'approximation des lois de commandes stabilisatrices, dites commandes idéales, dans les autres cas (approches directes). Le problème des erreurs d'approximation, inhérent à ce type de stratégies de commande, est traité par l'utilisation des termes de commande robuste de mode de glissement. La stabilité et la robustesse de toutes les structures de commande sont étudiées par l'approche de Lyapunov. L'avantage de ces stratégies est leur simplicité de mise en œuvre car, aucune information sur le comportement dynamique du système n'est nécessaire pour l'implantation de la loi de commande et la phase d'initialisation des algorithmes n'exige pas de connaissance a priori.

La présence de la saturation présente un vrai challenge pour le concepteur, du fait que si son existence est négligée lors de la synthèse d'un contrôleur, les résultats peuvent être des dégradations remarquables de la stabilité et des performances du système bouclé. La probabilité de l'apparition de ces effets augmente dans le cas de la commande adaptative, en particulier, si on n'arrête pas l'adaptation durant la saturation.

Le travail s'appuie sur une solution qui combine entre deux solutions déjà existantes pour le problème de la saturation. La première repose sur la modification de la loi de commande afin d'avoir un signal de commande toujours non saturé. Dans la deuxième solution, partant du constat que les lois d'adaptation sont les plus touchées par la variation des erreurs de poursuite lors de la saturation, cette solution vise à la modification de ces lois par l'élaboration d'une nouvelle variable d'erreur, excluant la partie de l'erreur due à la saturation. Dans ce qui suit, nous donnons un résumé des résultats obtenus et des perspectives à notre travail.

Dans le premier chapitre, quelques définitions sont données sur le concept de la logique floue et la structure interne des systèmes flous qui étaient la base de notre approche adaptative par leurs propriétés d'approximation universelle.

Dans le deuxième chapitre, nous avons proposé et développé un schéma de commande adaptative indirect pour une classe de systèmes non linéaires monovariables

en utilisant les systèmes flous. Dans ce schéma indirect, une approche de commande est proposée où les systèmes flous sont utilisés pour approcher en ligne la dynamique inconnue du système. A partir de ces approximations, une loi de commande adaptative floue, avec ces lois d'adaptation paramétrique, augmentée d'un terme de commande robuste est proposée. Cette approche indirecte évite le problème de singularité de la loi de commande par l'utilisation d'un algorithme de projection. Ce dernier permet de forcer les paramètres estimés à rester dans une région prédéfinie par l'utilisateur.

Dans le troisième chapitre le problème de saturation est traité dans ce travail par des solutions existantes, avec la commande adaptative indirecte floue cela en proposant une approche composée dont le principe consiste à modifier la loi de commande afin d'avoir un signal de commande toujours non saturé et modifié aussi le signal d'erreur utilisé dans les lois d'adaptation. La modification de la loi de commande est fait par sa diminution avec un terme additif qui intervient à chaque fois que le signal de commande se rapproche des bornes de saturation. La modification du signal d'erreur utilisée dans les lois d'adaptation consiste à isoler la composante de l'erreur due à la saturation.

Dans le quatrième chapitre, un schéma de commande adaptatif direct a été proposée. Dans ce cas, les systèmes flous sont utilisés pour l'approximation directe d'une commande idéale. Ainsi, le problème de singularité de la commande ne se pose pas. La modification de l'erreur utilisée dans les lois d'adaptation est la solution adoptée pour cette approche. On suppose que le gain de commande est constant, ce qui limite la classe des systèmes concernés.

Les simulations effectuées ont permis de relever les performances des différentes méthodes développées. Il est constaté que les systèmes flous utilisés pour l'approximation des non-linéarités inconnues où de la commande idéale ont prouvés leur efficacité et que tous les schémas de commande sont robustes vis-à-vis des erreurs d'approximation et conduisent à des erreurs de poursuite très faibles.

Comme perspective on opte à:

- Refaire les schémas de commande dans le cas ou la saturation est inconnue.
- Appliquer les schémas proposés sur des systèmes réels.

APPENDIX A

Annexe A

A.1 Théorème :

Soit $f(x)$ une fonction non linéaire définie sur un ensemble compact Ω. Pour toute constante positive , il existe un système flou $y(x)$ de la forme (1.21) tel que :

$$sup_{x\in\Omega}|f(x) - y(x)| < \varepsilon$$

La démonstration de ce théorème est donnée dans [9, 26].

Bibliography

[1] H.Buhler, « Réglage par logique floue », Presses polytechniques romandes, 1994. (Cit pages 2, 5 et 9.)

[2] J.J.E. Slotine et W. Li, « Applied Nonlinear Contro », Prentice-Hall, Englewoods, 1991. (Cit pages v, 2, 14, 20, 26, 36, 41 et 53.)

[3] T. Takaki et M. Sugeno, « Derivation of fuzzy control rules from human operatorscontrol action », Marseilles,France,1983. (Cit page 9.)

[4] J.Lvern, « La logique floue : Concepts et définitions », Electronique radio plan N 541,pp11-18,1992. (Cit page 3.)

[5] M.Jamshidi, N.Vadiee et T.J.Ross, « Fuzzy logic and control» , Prentice-Hall, 1993. (Cit pages 2 et 9.)

[6] D.Driankov, H.Hellendoorn et M.Reinfrank « An introduction : fuzzy control», Springer-Verlag, 1996. (Cit page 9.)

[7] J. R. Jang, 'ANFIS: adaptive- network-based fuzzy inference system', IEEE Trans.Syst., Man, Cybern., vol.23, no. 3, pp. 665-685, 1993. (Cit page 5.)

[8] J.S. R. Jang and C.T. Sun, " Neuro-Fuzzy modeling and control, Proc. IEEE, Man,Cybern., vol.83, no. 3, pp. 378-406,Mar. 1995. (Cit page 5.)

[9] L.X. Wang. Adaptive Fuzzy Systems and Control: Design and Stability Analysis. Prentice-Hall, Englewood Clifs, NJ,1994. (Cit pages iv, 5, 14, 17 et 60.)

[10] E. H. Mamdani, " Application of fuzzu logic to approximate reasoning using linguistique synthesis," IEEE Trans. Computers, vol.26, no. 12, pp. 1182-1191,dec. 1977. (Cit page 10.)

[11] J. M. Mendel, " Fuzzy systems for engineering: A tutorial, Proc. IEEE, vol. 83. no. 3, pp. 347-377, Mar. 1995. (Cit page 5.)

[12] T.Hiyama, " Application of rule-based stabilizing controller to electrical power System,IEEE Proc.. vol. 136. pt.c.no.3, may 1989, pp, 175-181. (Cit pages 7 et 17.)

[13] M. Sugeno and G. T. Kang, " Fuzzy identification of systems and its applicationsto modeling and control, IEEE Trans. Syst, Man, Cybern., vol.15, pp, 116-132, 1985. (Cit pages 10, 17 et 23.)

[14] M Sugeno and G T Kang, " Structure identification of fuzzy models, fuzzy sets and systems, vol.28, pp, 15-33, 1988. (Cit page 10.)

[15] L X Wang , " Stable adaptive fuzzy control of nonlinear systems, IEEE Trans, Fuzzy Systems, vol 1, pp 146-155, 1993. (Cit pages iv, 14 et 17.)

[16] M. Krstic, I. Kannellakopoulos and P. Kokotovic, " nonlinear and adaptive control and design. New York: Weley Interscience, 1995 (Cit page v.)

[17] J T Spooner and K M Passino," Stable adaptive control using fuzzy systems and neural networks,IEEE trans fuzzy syst vol 4 pp 339-359 1996. (Cit pages 14 et 22.)

[18] Y. C. Chang, Adaptive fuzzy-based tracking control for nonlinear SISO systems via VSS and H approaches. IEEE Trans. Fuzzy Syst., 9 : 278-292, 2001. (Cit page 14.)

[19] M.Zelmat, « Commande adaptative », 2001. (Cit page 16.)

[20] B. Meunir, «la logique floue et ces applications», addisowesley france , 1995. (Cit page 17.)

[21] Thèse de doctorat en informatique industrielle de Nadia Aouani ingénieur (2006) sur la commande Robuste des Systèmes Linéaires Continus a Paramètres Variant dans le Temps. (Cit page 32.)

[22] S. labiod, T.M. Guerra, direct adaptive fuzzy control of a class of nonlinear systems with input saturation. Third IFAC Workshop on Advanced Fuzzy and Neural Control, 2007. (Cit page 34.)

[23] J. H. Park, G. T. Park, Robust adaptive controller using universal approximators for nonlinear systems under input constraint. ISIE, pp. 1881-1886, 2001. (Cit page 34.)

[24] M. Polycarpou, J. Farrell and M. Sharma, On-line approximation control of uncertain nonlinear systems: issues with control input saturation. In Proc. ACC, pp. 543-548, 2003. (Cit pages v, 31 et 34.)

[25] A. Zibra, " Commande adaptative indirecte floue des systemes non linéaires avec saturation des commande . (Cit pages iii, 15, 32, 33, 34 et 39.)

[26] : J. L. Castro, Fuzzy logic controllers are universal approximators.IEEE. trans. Systems, Man, and Cybernetics, vol. 25, no. 4, pp. 629-635, 1995.

[27] A. Zibra,S Labiod and Thierry Marie Guerra "Adaptative Fuzzy Control for a Class of Uncertain Nonlinear Systems With Input Magnitude and Rate Saturation Constraint .

[28] A. Isidori, " Nonlinear Control systems .Springer-Verlag, Berlin, Heidelberg, 1989.

[29] S. Labiod, M. S. Boucherit, and T. M. Guerra, "adaptative floue d'une classe de systèmes non linéaires monovariables .Actes LFA'03, Tours, pp. 169-176, 2003.

[30] R. Boukazzoula, " Commande floue d'une classe de systèmes non linéaires : application au problème de suivi de trajectoire .Thèse de doctorat, Université de Savoie, Mars. 2000.

[31] S. Labiod, "à la commande adaptative floue des systèmes non linéaires .Thèse de doctorat, ENP, Janvier. 2005.

[32] S. Labiod and M. S. Boucherit, " Direct stable fuzzy adaptive control of a class of SISO nonlinear systems : Archives of control Sciences, 13 (1) : 95-110, 2003.

(Cit page 60.)