

REPUBLIQUE ALGERIENNE DEMOCRATIQUE ET POPULAIRE

MINISTERE DE L'ENSEIGNEMENT SUPERIEUR ET DE LA RECHERCHE SCIENTIFIQUE



Université Kasdi Merbah Ouargla
Faculté des Sciences Appliquées
Département de Génie électrique



THÈSE

Présentée pour l'obtention du diplôme de

DOCTORAT en SCIENCES

Spécialité : Génie électrique

Par : BENHELLAL Belkheir

Thème

**Commande d'un système non linéaire
par une méthode hybride**

Soutenue le : 09/01/2019 devant le jury composé de :

Pr. KELFAOUI Fethi	Président	Professeur.	Univ Ouargla
Dr. BELHOCINE Mahmoud	Examineur	Directeur de recherche	CDTA
Dr. MERAOUMIA Abdallah	Examineur	MCA	Univ Tebessa
Dr. BOULESBAA Mohammed	Examineur	MCA	Univ Ouargla
Dr. LOUAZEN Med Lakhdar	Examineur	MCA	Univ Ouargla
Dr. HAMERLAIN Mustapha	Directeur de thèse	Directeur de recherche	CDTA

بِسْمِ اللَّهِ الرَّحْمَنِ الرَّحِيمِ

بِسْمِ اللَّهِ الرَّحْمَنِ الرَّحِيمِ

صَدَقَ اللَّهُ الْعَظِيمُ

Dédicaces

*A la mémoire de mon père, a ma mère que dieu la protège,
A mes frères, mes sœurs et mon épouse,
A tous mes amis,
A tous ceux que j'aime et qui m'aiment, où qu'ils soient,
A tous ceux qui m'ont aidé et encouragé.*

Belkheir BENSALLA

Remerciements

Je remercie tout d'abord DIEU le tout puissant, de m'avoir permis d'atteindre ce modeste niveau scientifique et de m'avoir donné le courage et la patience afin de mener à bien le travail réalisé dans cette thèse.

*Mes sincères remerciements à monsieur **Mustapha Hamerlain**, mon directeur de thèse, Directeur de recherche au Centre de Développement des Technologie Avancées 'CDTA', Alger, pour son appui scientifique, sa disponibilité, ses orientations judicieuses et dont les compétences intellectuelles, l'expérience, la modestie et la patience ont grandement contribué à l'aboutissement de cette thèse. Qu'il trouve, ici, l'expression de mon profond respect.*

*Je tiens aussi à remercier monsieur **Pierre Sicard**, Professeur à l'université de Quebec à trois rivières, pour ces précieux conseils durant mon stage à L'UQTR et ses conseils pratiques qui m'ont aidé à mieux aborder et concevoir le sujet de la thèse.*

Je tiens à exprimer toute ma gratitude aux membres de jury :

***Pr. Fethi Khelfaoui**, qui ma fait l'honneur de présider ce jury.*

***Dr Abdallah Meraoumia**, **DR Mohammed Boulesbaa**, **Dr Mohammed Lakhdar Louazen** et **Dr Mahmoud Belhocine**, qui ont eu la gentillesse d'accepter mon invitation à juger ce travail.*

J'adresse aussi mes sincères reconnaissances à tous les enseignants du département d'électronique et des communications de l'université Kasdi Merbah à Ouargla pour leurs aides et leurs conseils, sans oublier tout le staff administratif de la Faculté des Nouvelles Technologies de l'Information.

Enfin, mes remerciements vont à tous ceux qui ont contribué d'une quelconque manière à l'aboutissement de ce travail.

Résumé

Le déplacement d'une charge suspendue le long d'une trajectoire pré-spécifiée n'est pas une tâche facile quand des spécifications strictes sur l'angle de balancement doivent être satisfaites. Cependant, afin de minimiser l'angle de balancement, les lois de contrôle adaptatif sont essentielles et en particulier dans le cas où la dynamique des systèmes est soumise à des incertitudes. Dans ce travail, nous proposons un contrôleur neuro-flou pour le contrôle en off-line du Robot Motoman en simulation ainsi qu'un contrôleur découplé adaptatif neuro-flou basé sur la théorie du mode glissant pour le contrôle du système de grue 3D. Avec ce dernier, nous avons utilisé deux structures du contrôleur hybride; la première est un contrôleur neuro-floue de type-1 et la seconde est un contrôleur neuro-flou intervalle type-2.

Le système de grue 3D considéré entraîne une charge dans un mouvement dans le plan en conjonction avec un mouvement de levage. Il dispose de trois entrées de commande uniquement (forces de chariot et de levage) avec cinq variables contrôlées (la position du chariot dans le plan XOY, la longueur du câble de levage, et les deux angles de basculement). Les interactions entre chaque sous-système de contrôle ne sont pas prises en compte explicitement, mais sont considérées comme des perturbations dans le contrôle de chaque sous-système individuel.

Dans l'approche adaptative proposée, un contrôleur conventionnel (PD) est utilisé en parallèle avec le contrôleur neuro-flou. Le contrôleur (PD) assure la stabilité asymptotique dans l'espace compact, les paramètres d'adaptations des règles du réseau neuro-flou sont dérivés, et les théorèmes de l'algorithme d'apprentissage en ligne sont démontrés en utilisant la méthode de la stabilité de Lyapunov. Les résultats expérimentaux sont donnés pour le cas de problème de contrôle de la position de la charge ainsi que dans le cas d'une perturbation avec minimisation de basculement du système de grue 3D du laboratoire.

Abstract

Moving a suspended load along a pre-specified path is not an easy task when strict specifications on the swing angle need to be satisfied. However, to minimize the swing angle, adaptive control laws are essential and especially in case the dynamics of systems subject to uncertainties. In this work we propose a neuro-fuzzy controller for off-line control of the Motoman robot in simulation as well as a decoupled controller neuro-fuzzy controller based on the sliding mode theory for the control of 3D crane system. With the latter, we used two structures of the hybrid controller; the first is the neuro-fuzzy-type 1 and the second is the neuro-fuzzy type-2 controller. The considered 3D crane system involves a plan movement in conjunction with a lifting movement. It has three control inputs only (trolley and hoisting forces) with five controlled variables (the trolley position in the XOY plane, the length of the lifting cable, and the two angles of swing). The interactions between each control subsystem are not taken into account explicitly, but are considered to be disturbances in control of each individual subsystem. In the proposed approach, a conventional controller (PD) is used in parallel with the neuro-fuzzy controller, the PD controller ensures the asymptotic stability in compact space, the parameter update rules of the fuzzy neural network are derived, and the proof of the online learning algorithm is verified by using the Lyapunov stability method. Experimental results are given to solve the crane position control problem of 3D crane system laboratory equipment.

Abstract

Moving a suspended load along a pre-specified path is not an easy task when strict specifications on the swing angle need to be satisfied. However, to minimize the swing angle, adaptive control laws are essential and especially in case the dynamics of systems subject to uncertainties. In this work we propose a neuro-fuzzy controller for off-line control of the Motoman robot in simulation as well as a decoupled controller neuro-fuzzy controller based on the sliding mode theory for the control of 3D crane system. With the latter, we used two structures of the hybrid controller ; the first is the neuro-fuzzy-type 1 and the second is the neuro-fuzzy type-2 controller. The considered 3D crane system involves a plan movement in conjunction with a lifting movement. It has three control inputs only (trolley and hoisting forces) with five controlled variables (the trolley position in the XOY plane, the length of the lifting cable, and the two angles of swing). The interactions between each control subsystem are not taken into account explicitly, but are considered to be disturbances in control of each individual subsystem. In the proposed approach, a conventional controller (PD) is used in parallel with the neuro- fuzzy controller, the PD controller ensures the asymptotic stability in compact space, the parameter update rules of the fuzzy neural network are derived, and the proof of the online learning algorithm is verified by using the Lyapunov stability method. Experimental results are given to solve the crane position control problem of 3D crane system laboratory equipment.

Table des matières

Résumé	iv
Liste des Figures	x
Liste de Tableaux	xiii
Abbreviations	xiv
1 Introduction générale	1
2 Ensembles flous	6
2.1 Ensembles flous de type-1	7
2.1.1 Caractéristiques d'un sous ensemble floue type-1	8
2.1.2 Opérations sur les ensembles flous type-1	9
2.1.3 Règle floue et raisonnement flou	10
2.2 Ensembles flous de type-2 (EFT-2)	11
2.2.1 Représentation d'un ensemble flou type-2 (EFT-2)	11
2.2.2 Types d'ensemble flous type-2	13
2.3 Systèmes flous	15
2.3.1 Système flou de type-1	15
2.3.2 Système flou de type-2	16
2.4 Conclusion	22
3 Systèmes neuro-flous	23
3.1 Modèles de neurone	25
3.2 Les réseaux de neurones	27
3.2.1 Le perceptron multicouche (PMC)	28
3.2.2 Structure d'interconnexion	28
3.2.3 Apprentissage	29
3.3 Algorithme d'apprentissage	30
3.4 Réseaux à fonction de base radiale (RBF)	32
3.5 Avantages et inconvénients des réseaux de neurones	34
3.5.1 Avantages des réseaux de neurones	34

3.5.2	Inconvénients des réseaux de neurones	34
3.6	Systèmes neuro-flous	35
3.6.1	Systèmes neuro-flous coopératifs et concurrents	35
3.6.2	Systèmes neuro-flous fusionnés	36
3.7	Architecture d'ANFIS	37
3.8	Résultats de simulation	39
3.9	Conclusion	42
4	Commande par mode glissant et adaptative	43
4.1	Commande Adaptative	43
4.1.1	Structure à gain programmé	44
4.1.2	Commande adaptatives à modèle de référence (MRAC)	45
4.1.3	Commande adaptative auto-ajustable (STR : Self Tuning Regulator)	46
4.2	Commande par mode glissant	46
4.2.1	Structures de base	48
4.2.2	Principe du réglage par mode glissant	49
4.2.3	Phénomène de Chattering	53
4.3	Résultats de simulation	54
4.4	Conclusion	57
5	Commande adaptative neuro-floue mode glissant	58
5.1	Commande adaptative neuro-floue par mode glissant	59
5.2	Réseau neuro-flou	59
5.3	L'algorithme d'apprentissage par mode glissant	62
5.4	Réseau neuro-flou intervalle type-2	65
5.5	Conclusion	69
6	Résultats expérimentaux	70
6.1	Le système 3DCrane	70
6.2	Le modèle mathématique du system « 3DCrane »	71
6.3	Résultats expérimentaux	73
6.4	Tests des performances	77
6.4.1	Perturbation externe	77
6.4.2	Changement de la trajectoire de référence	77
6.5	Commande adaptative neuro-floue intervalle type-2 mode glissant	80
6.6	Conclusion	84
7	Conclusion générale et perspectives	85
	Bibliographie	88

Annexe A. Algorithme de Karnil et Mendel	93
Annexe B. Modélisation du pendule inversé	95
Annexe C. Opérations sur les ensembles flous type-2	97

Liste des figures

1.1	Grue en bois.	2
1.2	Le système 3D Crane.	5
2.1	Caractéristiques d'un sous ensemble flou	8
2.2	Blurring d'une MF.	11
2.3	Représentation à 2.5D d'un ensemble flou type-2.	12
2.4	Ensemble flou type-2 gaussien.	13
2.5	Représentation à 2.5D d'un ensemble flou type-2 triangulaire.	14
2.6	Ensemble flou type-2 intervalle.	14
2.7	Structure d'un ensemble flou type-1.	16
2.8	Structure de base d'un système flou type-2.	17
3.1	Le neurone biologique.	25
3.2	Le neurone artificiel.	26
3.3	Modèle de base d'un neurone artificiel.	26
3.4	Le perceptron multicouche.	28
3.5	Classification des architectures des réseaux de neurones.	29
3.6	Réseau RBF avec un noyau gaussien.	33
3.7	Modèle neuro-flou coopératif.	36
3.8	Modèle neuro-flou concurrent.	36
3.9	Architectures des systèmes neuro-flous.	36
3.10	Modèle flou de Sugeno d'ordre un à deux entrées.	37
3.11	Architecture d'ANFIS équivalent au modèle flou de Sugeno d'ordre un à deux entrées et deux règles.	38
3.12	Autre architecture d'ANFIS pour le modèle flou de Sugeno à deux entrées et deux règles.	39
3.13	Poursuite de trajectoire.	40
3.14	Erreur de poursuite.	40
3.15	Signaux de commande.	40
3.16	Poursuite de trajectoire.	41
3.17	Erreur de poursuite.	41
3.18	Signaux commande.	41

4.1	Structure d'une commande daptative.	45
4.2	Structure de commande adaptative à gain programmé.	45
4.3	Commande adaptative à modèle de référence.	46
4.4	Commande adaptative auto-ajustable.	46
4.5	Choix de la structure de la commande adaptative.	47
4.6	Régulation par commutation au niveau de la contre réaction d'état.	48
4.7	Régulation commutation au niveau de l'organe de commande.	48
4.8	Structure de régulation par ajout de la commande équivalente.	49
4.9	Différents modes de convergence pour la trajectoire d'état.	50
4.10	Condition de glissement.	51
4.11	Construction de la dynamique équivalente d'après Filipov.	52
4.12	Phénomène de Chattering.	53
4.13	Boucle de contrôle avec boucle secondaire d'observateur.	54
4.14	Courbe de référence et de sortie du système.	55
4.15	Ecart entre la sortie et la référence.	55
4.16	Signal de commande.	55
4.17	Réponse dans le plan de phase.	55
4.18	Courbe de référence et de sortie du système.	56
4.19	Ecart entre la sortie et la référence.	56
4.20	Signal de commande.	56
4.21	Réponse dans le plan de phase.	56
5.1	Schéma de commande adaptative neuro-flou.	59
5.2	Architecture de contrôleur neuro-floue.	60
5.3	Architecture de contrôleur neuro-floue intervalle type-2.	65
6.1	Le système 3DCrane de laboratoire.	71
6.2	Représentation des coordonnées et forces du système 3DCrane.	72
6.3	Poursuite de trajectoire par rapport l'axe X.	74
6.4	Poursuite de trajectoire par rapport l'axe Y.	74
6.5	Poursuite de trajectoire par rapport l'axe Z.	74
6.6	Balancement des angles α et β en rad.	75
6.7	Erreur de poursuite sur l'axe X.	75
6.8	Erreur de poursuite sur l'axe Y.	75
6.9	Erreur sur l'axe Z.	76
6.10	Signal de commande dans l'axe X.	76
6.11	Signal de commande dans l'axe Y.	76
6.12	Signal de commande dans l'axe Z.	77
6.13	Poursuite de trajectoire par rapport aux axes (x, y, z).	77
6.14	Balancement des angles α et β	78

6.15	Signal de commande dans l'axe X.	78
6.16	Signal de commande dans l'axe Y.	78
6.17	Signal de commande dans l'axe Z.	79
6.18	Poursuite de trajectoire par rapport aux axes (x, y, z).	79
6.19	Balancement des angles α et β	79
6.20	Signaux de commande des axes (x, y, z).	80
6.21	Poursuite de trajectoire par rapport l'axe X.	81
6.22	Poursuite de trajectoire par rapport l'axe Y.	81
6.23	Poursuite de trajectoire par rapport l'axe Z.	82
6.24	Balancement de l'angle α	82
6.25	Balancement de l'angle β	82
6.26	Balancement de l'angle α	83
6.27	Balancement de l'angle β	83
A.1	Pendule inversé.	95

Liste des tableaux

2.1	Les opérations en logique floue	9
2.2	Fonctions d'appartenances	9
3.1	Les fonctions d'activation usuelles	27

Abbreviations

A2C1	: Antécédent type-2, Conséquence type-1.
A2C0	: Antécédent type-2, Conséquence type-0.
A2C2	: Antécédent type-2, Conséquence type-2.
ANNFLC	: Commande Adaptative Neuro-Flou, 'Adaptive Neural Network Fuzzy Logic Control'.
ANIT2FLC	: Commande Adaptative Neuro-Flou Type-2 Intervalle, 'Adaptive Neuro Interval Type-2 Fuzzy Logic Control'.
ANFIS	: Système d'Inférence Flou Adaptatif, 'Adaptive Neuro-Fuzzy Inference Systems'.
AFSMC	: Commande Adaptative par Mode Glissant Flou, 'Adaptive fuzzy sliding mode control'.
CNF	: Contrôleur Neuro-Flou, 'NFC, Neuro-Fuzzy Controller'.
DENFIS	: Système d'Inférence Neuro-Flou à Evolution Dynamique, 'Dynamic Evolving Neural-Fuzzy Inference System'.
EFT-2	: Ensembles Flous de Type-2.
EFT-1	: Ensembles Flous de Type-1.
FA	: Fonction d'Appartenance.
FAT-2	: Fonction d'Appartenance Type-2.
FALCON	: Réseau de Commande Flou Apprentissage Adaptif, 'Fuzzy Adaptive Learning Control Network'.
FL	: Logique Floue, 'Fuzzy Logic'.
FOU	: Empreinte de l'Incertitude, 'Footprint Of Uncertainty'.
FSMC	: Commande par Mode Glissant Flou, 'Fuzzy Sliding Mode Control?.'
FIS	: Système d'Inférence Flou, 'Fuzzy Inference System'.
FLEXNFIS	: Système d'Inférence Neuro-Flou Flexible, 'Flexible Neuro-Fuzzy Inference Systems'.
GA	: Algorithmes Génétiques, 'Genetic Algorithm'.
GARIC	: Commande Intelligente Généralisée par Renforcement Adaptatif, 'Generalised Adaptive Reinforced Intelligent Control'.
HEA	: Algorithme Evolutionnaire hybride, 'Hybrid evolutionary algorithm'.
IT2FLS	: System Flou de Type-2 Intervalle, Interval Type-2 Fuzzy Logic System'.
LMF	: Fonction d'Appartenance Inférieur, 'Lower Membership Function'.
MIMO	: Entrées Multiples et Sorties Multiples 'Multi-Input Multi-Output'.
MLP	: Perceptron Multi Couche PMC, 'Multi Layer Perceptron'.
MRAC	: Commande Adaptative à Modèle de Référence 'Model Reference Adaptive Control'.

NF	:	Neuro-Flou, 'Neuro-Fuzzy'.
NFC	:	Contrôleur Neuro-Flou, 'Neuro-Fuzzy Control'.
NEFCLASS	:	Classification Neuro-Floue, 'Neuro-Fuzzy Classification'.
NEFPROX	:	Approximation de Fonction Neuro-Floue, 'Neuro-Fuzzy Function Approximator'.
NFIT2	:	Neuro-Flou Type-2 Intervalle, 'Neuro-Fuzzy Interval type-2'.
PD	:	proportionnel dérivé.
RNN	:	Réseau Neuronal Récurrent, 'Recurrent Neural Network'.
RNF	:	Réseaux Neuro-Flous.
RNA	:	Réseaux de Neurones Artificiels.
RBF	:	Fonction De Base Radiale, 'Radial Basis Function'.
RN	:	Réseau de Neurone.
STA	:	'Super-Twisting Algorithm'.
SIRMs	:	Modules de Règles d'Entrée Unique, 'The Single-Input-Rule-Modules'.
SIRMFLC	:	Contrôleur Flou à Modules de Règles d'Entrée Unique, 'Single-Input-Rule-Modules Fuzzy Logic Control'.
SANFIS	:	Auto-Adaptatif Système d'Inférence Neuro-Flou, 'Self-Adaptive Neuro-Fuzzy Inference System'.
STR	:	'Self Tuning Regulator'.
SSV	:	Système à Structure Variable.
T2FLC	:	Contrôleur Flou de Type-2, 'Type-2 Fuzzy Logic Control'.
T2FLS	:	Contrôleur Flou de Type-2, 'Type-2 Fuzzy Logic System'.
TSKFLS	:	System Flou de Takagi-Sugeno-Kang, 'Takagi-Sugeno-Kang Fuzzy Logic System'.
UMF	:	Fonction d'Appartenance Supérieur, 'Upper Membership Function'.

Chapitre 1

Introduction générale

Les systèmes de grues sont principalement utilisés pour le levage lourd et le déplacement de charges au-delà de la capacité normale d'un homme, pour cette raison, le contrôle de ces systèmes, qui n'est pas une tâche simple en raison de la complexité de leur modèle (systèmes MIMO, fortement couplé et non linéaires), joue un rôle important dans les applications industrielles à cause de la haute performance qu'ils doivent offrir.

Les propriétés dynamiques fortement non linéaires ainsi que le manque d'entrée de commande réelle pour le mouvement de balancement pourrait provoquer des oscillations indésirables et significatives, en particulier à la phase de démarrage et d'arrivée. En outre, les systèmes portiques ont la particularité d'avoir un nombre limité d'entrée pour contrôler plusieurs sorties. Dans ce cas, les oscillations incontrôlables peuvent causer des problèmes de sécurité et de stabilité graves, et limiteraient largement l'efficacité de l'opération. Cependant, ces phénomènes indésirables rendent également les stratégies de contrôle classiques ne parviennent pas à atteindre l'objectif.

Dans la plupart des applications d'ingénierie de contrôle, la performance du contrôleur est directement liée à la précision du modèle mathématique obtenu pour le système contrôlé. Durant les dernières décennies, les techniques informatiques intelligentes utilisant la logique floue et les réseaux de neurones ont été étudiées afin de surmonter les difficultés de modélisation existantes. Le concept de l'incorporation de la logique floue dans le réseau neuronal a émergé et est devenu un domaine de recherche populaire. Les réseaux neuro-flous (RNFs) combinent les avantages des deux techniques. Comme les systèmes flous et les réseaux de neurones, il a été prouvé dans la littérature que les RNFs sont des approximateurs universels.

Les systèmes de grues sont largement utilisés dans les usines modernes, les entrepôts, les ports, les aéroports, les installations nucléaires, en général dans toute structure où on a besoin de manipuler des objets de masse importante. Son essence est une sorte de dispositifs électromécaniques utilisés pour transférer les objets d'une position A à une position cible B avec le caractère de stabilité et de précision. Due à l'architecture du système lui-même et les

facteurs externes, il n'est pas facile pour la grue de réaliser cet objectif. Dans certains cas, et en raison de la mauvaise stabilité, ou un manque de précision, des accidents très graves se produisent fréquemment. L'effet des oscillations dans un système de grue peut provoquer des chocs avec d'autres objets qui peut engendrer d'autres dégâts très importantes et même la coupure du câble porteur de la charge.

Dans la littérature les premiers systèmes de grues ont été conçus en bois avec un système de commande manuelle très simple, ils fonctionnaient par l'intervention humaine.

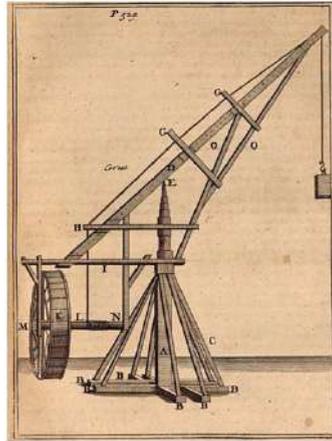


FIGURE 1.1: Grue en bois.

Ces Grues étaient composées de plusieurs pièces de bois Assemblés de façon à réaliser une structure sous forme de flèche capable de transférer des objets lourds on réalisant un mouvement rotationnelle autour de l'axe A. [1].

Dans le domaine de l'industrie, il existe différents modèles de grues qui diffèrent dans leurs structures, selon l'application pour laquelle ils sont conçus, parmi ces modèles on peut citer :

- ◆ Grues de bord.
- ◆ Grues mobiles.
- ◆ Grues portique secteur conteneur.
- ◆ Grue portique vrac.

D'autres dispositifs de contrôle peuvent être intégrés selon les besoins, Cependant et avec tout ces dispositifs le problème de basculement de la charge ne peut être éliminé, et reste toujours un domaine où les chercheurs doivent investir pour minimiser les oscillations d'une charge portée par une grue et pour la faire atteindre un objectif désiré. Autrement dit une augmentation des performances en précision et stabilité dans ces systèmes.

Dans tous les modèles de grues qui se trouvent dans l'industrie, la caractéristique d'avoir un nombre de variable en sortie supérieurs au nombre de variable d'entrée est commune, ce qui fait qu'un certain nombre de variables est commandé indirectement. Les systèmes avec cette caractéristique sont appelés des systèmes mécaniques sous actionnés, l'accélération nécessaire

pour déplacer le chariot induit de grands oscillations dans la charge, donc ces systèmes sont d'une dynamique complexe non linéaire et fortement couplés.

Pendant les dernières décennies, de nombreuses approches en matière de contrôle de la grue ont été développées et rapportées dans la littérature, parmi les approches qui sont basées sur le mode glissant on trouve, A. Nowacka-Leverton et al [2] ont proposé une méthode de commande basée sur le mode glissant d'ordre 1, pour le contrôle d'un mouvement point à point d'une grue à 3D. Carlos Vazquez et al [3] ont développé une commande par mode glissant d'ordre 1 basée sur un algorithme appelé 'Super-Twisting Algorithm, STA', dans leur travail l'apport principal a été d'éviter l'effet de résonance paramétrique et d'atténuer les oscillations de la charge utile dans un système de grue soumis à des variations périodiques dans le support de base. Le Anh Tuan et al [4] ont proposé dans leur étude une commande par mode glissant d'ordre 2, qu'ils ont appliquée à une grue 3D dans des conditions de fonctionnement très compliquées. Carlos Vazquez et al [5] ont aussi proposé une commande par mode glissant d'ordre 2 mais cette dernière est basée sur l'algorithme STA. Cette commande a été appliquée à un système de grue en 1D avec une plate forme en mouvement et des perturbations externes sur la charge portée. D'autres approches de commande des systèmes de grues sont orientées vers la commande basée sur la logique floue, on cite par exemple les travaux de Dai Shijie et al [6] qui ont développé une méthode de contrôle basée sur la linéarisation partielle du système avec la logique floue, cette méthode a minimisé le basculement et a réalisé une localisation précise.

Sung-Kun Cho, Ho-Hoon Lee [7] eux ont proposé un asservissement en position qui est utilisé pour contrôler la position de la grue et de la longueur de la corde en parallèle avec une commande par logique floue, l'approche est utilisée pour supprimer les oscillations de la charge.

D'autres auteurs ont développé des stratégies de commandes basées sur la logique floue mais qui réduisent considérablement le nombre de règles floues, cette méthode appelée SIRM 'The single-input-rule-modules' ou module de règle à une seule entrée. Dans [8], le SIRM a été utilisé avec la logique floue type-2 pour concevoir un contrôleur par logique floue de type-2 basé sur les modules de règle à une seule entrée, cette commande a été appliquée pour la stabilisation d'un système de grue. Dans [9] les auteurs ont conçu un contrôleur flou type-1 basé sur les SIRM modèle d'inférence flou dynamiquement connecté, et ont appliqué ce dernier à un système de grue, les auteurs dans [10] ont mis au point les conditions qui permettent de concevoir un SIRMFLC, pour la logique floue type-1 et type-2, et pour démontrer l'efficacité de la méthode proposée, ils l'ont appliquée à la commande de stabilisation du système de grue.

Pour augmenter les performances des systèmes de grues des auteurs se sont orientés vers la combinaison des stratégies de contrôle pour concevoir des méthodes adaptatives, l'utilisation de la logique floue avec le mode glissant permet de prouver la stabilité du système. Les auteurs dans [11] ont utilisé une transformation du modèle non linéaire de la grue en

deux sous systèmes indépendant, un sous système de positionnement et un sous système anti-basculement, et ils ont appliqué une commande adaptative floue par mode glissant pour chaque sous système. Mun-Soo Park et al ont développé une commande floue par mode glissant et une commande adaptative floue par mode glissant 'FSMC', 'AFSMC' pour la poursuite de trajectoire sans basculement robuste d'une charge portée par la grue [12]. Dans [13] les auteurs ont conçu une structure de réseau neuronal récurrent (RNN) évolué par un algorithme évolutionnaire pour le contrôle de la grue. Ensuite afin d'améliorer la performance de l'évolution du RNN, un algorithme évolutionniste hybride (HEA) a été utilisé avec le réseau.

Afin de surmonter les difficultés de modélisation existantes. Plusieurs chercheurs se sont orientés vers le concept de l'incorporation de la logique floue dans le réseau neuronal, pour développer des approches de commandes neuro-flous, avec ces derniers les concepts du mode glissant et de la commande adaptative sont utilisés pour former des structures de commandes complexes bénéficiant de tous les avantages de ces derniers. Une autre alternative très prometteuse a été utilisée par quelques chercheurs est celle de ces combinaisons d'approches mais avec la logique floue type-2, et ceci pour surmonter le problème des incertitudes aux niveaux des fonctions d'appartenances qui sont exactes dans la logique floue de type-1.

Les méthodes de commande classiques ont été utilisées avec différentes forme pour la commande des systèmes de grue, cependant l'évolution de ces derniers, et la demande de spécifications pour qu'elles puissent être utilisées dans des conditions critiques et difficiles les rend de plus en plus exigeant en matière de précision et stabilité. Sur ce contexte est basé le choix de la commande adaptative neuro-floue mode glissant en parallèle avec un régulateur classique PD pour le réglage en position du systèmes de grue 3D avec minimisation de basculement de la charge, le système est supposé découplé, et à chaque sous système est appliquer la même structure de la loi de commande.

Le réseau neuro-flou NFC choisi avec cette structure de commande est de type Takagi–Sugeno avec cinq couches, une deuxième approche de commande avec la même structure est utilisée pour la commande de système basée sur la logique floue type-2, l'objectif est d'augmenter la précision et la stabilité du système du fait que dans ce type les incertitudes sont aussi modéliser.

La commande hybride neuro-flou est appliquée en simulation sur le robot Motoman SV3, pour cela nous avons utiliser les boites à outils robotic toolbox et fuzzy logic toolbox.

Les lois de commandes adaptative neuro–flous mode glissant avec les systèmes flous de type-1 et les systèmes flous de type-2 sont développer dans l'environnement Matlab Simulink et implémentées sur une plateforme expérimentale 3Dcrane système de INTECO. Cette plate forme représente un modèle réduit d'une grue à 3 dimension, mais elle conserve les caractéristiques fondamentales du système réel. La position de la charge utile est décrite par deux angles α et β .



FIGURE 1.2: Le système 3D Crane.

Cette thèse est structurée comme suit :

Le deuxième chapitre présente les concepts de bases de la logique floue type-1, et les principales caractéristiques des systèmes flous de type-1, par la suite les outils essentiels de la logique floue type-2 sont exposés, ainsi que la structure de base des systèmes flous type-2, et spécialement ceux de type Takagi Sugeno.

Dans le chapitre trois les notions de base des réseaux de neurones et ceux des systèmes neuro-flous sont présentées, les différentes structures de ces systèmes sont citées, à la fin de ce chapitre on présente l'algorithme de rétro propagation de l'erreur pour les systèmes AN-FIS, des résultats de simulation de cette commande appliquée sur le système de grue sont présentés.

Le quatrième chapitre porte sur la commande par mode glissant et adaptative, dans ce dernier quelques rappels des principes de bases des deux types de commande sont présentés brièvement, dans la dernière partie de ce chapitre on présente les résultats de simulations qui montrent l'effet du phénomène de Chattering sur un simple système.

Le cinquième chapitre présente le développement de la commande adaptative neuro-flou mode glissant, et les paramètres d'adaptations, ainsi que la justification par le théorème de Lyapunov de la stabilité du système par l'application de ces paramètres.

Dans le même chapitre le développement de la loi de commande adaptative neuro-flou intervalle type-2 mode glissant est présentée, la structure de la commande appliquée au système est donnée, ainsi que la démonstration de la stabilité par le théorème de Lyapunov.

Le dernier chapitre comporte les résultats expérimentaux de l'application des deux lois de commandes sur le système de grue. Les résultats des tests de performance des commandes sont présentés ainsi que l'interprétation de ces résultats. Enfin on termine par une conclusion générale et quelques perspectives.

Chapitre 2

Ensembles flous

DANS les sources de la littérature, on peut trouver différents types de justification de la théorie des systèmes flous. La connaissance humaine devient aujourd'hui de plus en plus importante, nous l'acquérirons en expérimentant le monde dans lequel nous vivons et nous utilisons notre capacité à raisonner et à mettre de l'ordre dans la masse d'informations (par exemple, pour formuler la connaissance humaine d'une manière systématique). Puisque nous sommes tous limités dans notre capacité à percevoir le monde et de raisonnement profond, nous trouvons partout confrontés à l'incertitude qui est le résultat du manque d'information (impression lexical, l'incomplétude), en particulier, l'inexactitude des mesures. L'autre facteur de limitation dans notre désir de précision est un langage naturel utilisé pour décrire le partage des connaissances, la communication, etc. Nous comprenons significations fondamentales de parole et sommes en mesure de communiquer avec précision à un niveau acceptable, mais en général, nous ne pouvons pas précisément d'accord entre nous sur le seul mot ou des mots de sens de bon sens. En bref, les langues naturelles sont vagues [14].

La logique floue a été proposée par Zadeh 1965 et est basée sur le concept des ensembles flous. La théorie des ensembles flous fournit un moyen pour représenter les incertitudes. En général, la théorie des probabilités est le principal outil pour analyser l'incertitude, et suppose que l'incertitude est un processus aléatoire. Toutefois, pas toute incertitude est aléatoire, et la théorie des ensembles flous est utilisée pour modéliser le genre d'incertitude lié à l'imprécision, et le manque d'information [15].

En 1975 le professeur Lofti Zadeh a également introduit le concept des ensembles flous de type-2 comme une extension de la notion des ensembles flous ordinaires (ensemble flou de type-1). La logique floue de type-2 permet d'intégrer de l'incertitude dans les réponses et ainsi de considérer simultanément les différents avis des experts.

Le système d'inférence floue apparu pour la première fois dans les années soixante dix avec des applications dans le domaine de contrôle des processus, est un cadre populaire de calcul basé sur les concepts de la théorie des ensembles flous, les règles floues si-alors, et le

raisonnement flou. En raison de sa nature multidisciplinaire, le système d'inférence floue est connu par de nombreux autres noms, tels que "système expert flou" (Kandel, 1992), "modèle flou" (Sugeno et Kang, 1988), "mémoire associative floue" (Kosko, 1992), et tout simplement "système flou". La structure de base d'un système d'inférence floue de type-1 se compose de trois composantes conceptuelles : une "base de règles", qui contient une sélection de règles floues ; une "base de données" (ou "dictionnaire"), qui définit les fonctions d'appartenances utilisées dans les règles floues, et un "mécanisme de raisonnement", qui effectue l'inférence des règles et les faits donnés pour dériver une sortie ou une conclusion raisonnable.

En général, nous pouvons dire que le système d'inférence floue met en œuvre une cartographie non linéaire à partir de son espace d'entrée à l'espace de sortie. Cette cartographie est réalisée par un certain nombre de règles floues SI-ALORS, dont chacun décrit le comportement local de la cartographie. En particulier, l'antécédent d'une règle floue définit une région dans l'espace d'entrée, tandis que la conséquence spécifie la sortie dans la zone floue. Aujourd'hui, les applications des systèmes d'inférences flous sont très nombreuses outre la commande, ils sont largement utilisés pour la modélisation, le diagnostic et la reconnaissance de formes.

Les systèmes flous de type-2 sont constitués de règles floues si-alors, qui contiennent des ensembles flous de type-2. Nous pouvons dire que la logique floue de type-2 est une généralisation de la logique floue classique (de type-1) dans le sens que l'incertitude est non seulement limitée aux variables linguistiques, mais est également présente dans la définition des fonctions d'appartenance [16].

Ce chapitre présente dans la première partie, les concepts de base, la notation et les opérations dans les ensembles flous de type-1. Les ensembles flous de Type-2 ainsi que leurs opérations seront discutés dans la partie suivante de ce chapitre. Nous introduisons également la définition des variables linguistiques et expliquons comment les utiliser dans les règles floues type-1, qui sont un outil efficace pour la modélisation quantitative des mots ou des phrases dans une langue naturelle ou artificielle. En interprétant les règles floues que les relations floues, nous décrivons les différentes méthodes de raisonnement flou, où les procédures d'inférences basées sur le concept de la règle de composition de l'inférence sont utilisées pour tirer des conclusions à partir d'un ensemble de règles floues et les faits connus. Les Règles floues et raisonnement flou sont les éléments de base des systèmes d'inférence floue, qui sont le plus important outil de modélisation, basés sur la théorie des ensembles flous.

2.1 Ensembles flous de type-1

Le concept de la théorie des sous-ensembles flous, s'appuie sur la notion de degré d'appartenance d'un élément à un sous-ensemble flou. Tandis que les ensembles traditionnels sont

caractérisés par une fonction notée χ (appelée fonction caractéristique) définie sur $0,1$. Si X est une collection d'objets désignés de façon générique par x , un "ensemble flou" A dans X est défini comme un ensemble de paires ordonnées :

$$A = \{(x, \mu_A(x)) | x \in X\} \quad (2.1)$$

Où $\mu_A(x)$ est appelée « fonction d'appartenance » pour l'ensemble flou A . La fonction d'appartenance fait correspondre à chaque élément de X à un degré d'appartenance entre 0 et 1.

A peut être aussi définie par

$$A = \int_{x \in X} \mu_A(x) / x \quad (2.2)$$

Où \int représente l'union de toutes les valeurs x admissibles.

$$\mu_A(x) = \begin{cases} 0 < b \leq 1 & x \in a \\ 0, & \text{sinon.} \end{cases} \quad (2.3)$$

2.1.1 Caractéristiques d'un sous ensemble flou type-1

Un sous-ensemble A sur un univers de discours U , est représenté comme dans la figure (2.1) à travers sa fonction caractéristique $\mu_A(x)$. Il peut également être décrit par un certain nombre de caractéristiques comme :

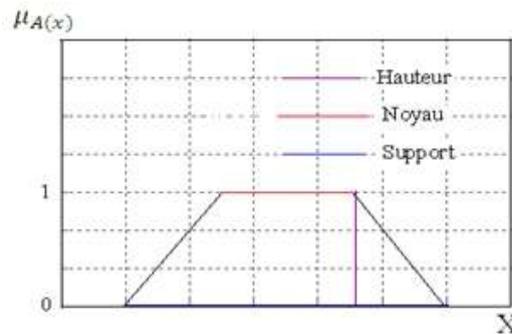


FIGURE 2.1: Caractéristiques d'un sous ensemble flou

◦ Support :

$$Support(A) = \{x \in U / \mu_A(x) \neq 0\} \quad (2.4)$$

◦ Hauteur :

$$Hauteur(A) = sup\{\mu_A(x)\} \quad (2.5)$$

◦ Noyau :

$$Noyau(A) = \{x \in U / \mu_A(x) = 1\} \quad (2.6)$$

2.1.2 Opérations sur les ensembles flous type-1

Les ensembles flous ont des opérations similaires aux opérations sur les ensembles ordinaires, l'union, intersection et complément ont été définis dans un article fondateur de Zadeh [17].

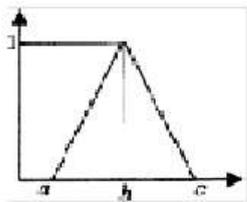
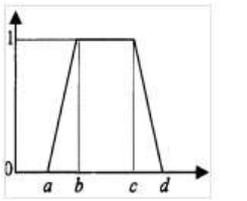
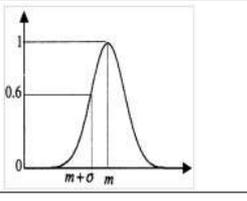
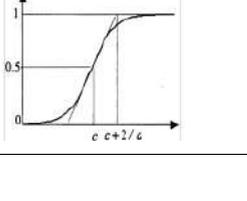
Les opérations sur les ensembles flous de type-1 sont résumées dans le tableau Tab 2.1 :

TABLE 2.1: Les opérations en logique floue

Opération	La forme en logique flou
inclusion	$A \subset B \Leftrightarrow \mu_A(x) \leq \mu_B(x)$.
Complément	$\mu_{\bar{A}}(x) = 1 - \mu_A(x)$.
Union	$\mu_{A \cup B}(x) = \max(\mu_A(x), \mu_B(x)), x \in X$
Intersection	$\mu_{A \cap B}(x) = \min(\mu_A(x), \mu_B(x)), x \in X$
Produit cartésien	$\mu_{A_1 \times \dots \times A_n}(x_1, \dots, x_n) = \min(\mu_{A_1}(x_1), \dots, \mu_{A_n}(x_n))$ ou $\mu_{A_1 \times \dots \times A_n}(x_1, \dots, x_n) = \mu_{A_1}(x_1) \times \dots \times \mu_{A_n}(x_n)$

Un ensemble flou est entièrement caractérisé par sa fonction d'appartenance. Un moyen pratique pour définir cette dernière est de l'exprimer comme une formule mathématique. Les types de fonctions à une dimension les plus utilisés sont présentés dans le tableau Tab 2.2 [18] :

TABLE 2.2: Fonctions d'appartenances

Fonction	La forme algébrique	La forme géométrique
Fonction triangulaire	les trois paramètres a,b,c qui déterminent les coordonnées des trois coins $\mu(x) = \begin{cases} 0 & x \leq a \\ \frac{x-a}{b-a} & a \leq x \leq b. \\ \frac{c-x}{c-b} & b \leq x \leq c \\ 0, & c \leq x. \end{cases}$	
Fonction trapézoïdale	elle est définie par quatre paramètres a,b,c,d comme suit : $\mu(x) = \begin{cases} 0 & x \leq a \\ \frac{x-a}{b-a} & a \leq x \leq b. \\ 1, & a \leq x \leq c. \\ \frac{d-x}{d-c} & c \leq x \leq d. \\ 0, & d \leq x. \end{cases}$	
Fonction gaussienne	elle est définie par les deux paramètres m,σ $\mu(x) = \exp\left(-\frac{(x-m)^2}{2(\sigma)^2}\right)$	
Fonction sigmoïde	elle est définie par les deux paramètres a,c $\mu(x) = \frac{1}{1+\exp(-a(x-c))}$	

2.1.3 Règle floue et raisonnement flou

Les règles forment la base de la logique floue pour obtenir une sortie floue. Le système à base de règles est différent du système expert de la manière que les règles comprenant le système à base de règles proviennent de sources autres que celles des experts humains. La forme à base de règles utilise des variables linguistiques comme ses antécédents et conséquents. Les antécédents expriment une condition ou une inégalité, qui doit être satisfaite.

Les conséquents sont ceux dont nous pouvons en déduire, et présente la sortie si l'inégalité de l'antécédent est satisfaite. Le système à base de règles floues utilise des règles SI-ALORS, donnée par 'SI antécédent, ALORS conséquent'.

On appelle proposition floue élémentaire, une proposition de type 'X est A' où (X,U,Tx) est une variable linguistique et A un sous ensemble de Tx. Une telle proposition possède un degré de vérité égal à $\mu_A(x)$ où x est une valeur réelle de X. D'une manière générale, on peut combiner ces propositions élémentaires à l'aide des opérateurs logiques de conjonction et de disjonction ('et' et 'ou') mis en œuvre respectivement par des T-normes et T-conormes [19].

Le degré de vérité des nouvelles propositions obtenues peut être calculé entre autre par les équations suivantes :

Conjonction : (X est A) ET (Y est B)

$$\text{minimum } (\mu_A(x), \mu_B(y))$$

$$\text{produit } \mu_A(x) \times \mu_B(y)$$

Disjonction : (X est A) OU (Y est B)

$$\text{maximum } (\mu_A(x), \mu_B(y))$$

$$\text{somme } \mu_A(x) + \mu_B(y) + \mu_A(x) \times \mu_B(y)$$

L'opérateur d'implication permet d'introduire la notion de règle floue qui caractérise les relations de dépendance entre plusieurs propositions floues :

$$(X1 \text{ est } A1) \text{ ET } (X2 \text{ est } A2) = (Y \text{ est } B) \quad (2.7)$$

Où X1, X2 et Y sont des variables linguistiques et A1 et A2 et B sont des sous ensembles flous. Une telle règle se trouve habituellement dans les systèmes flous avec une formulation légèrement différente :

$$\text{Si } (X1 \text{ est } A1) \text{ ET } (X2 \text{ est } A2) \text{ Alors } (Y \text{ est } B) \quad (2.8)$$

Dans cette dernière formulation la partie (X1 est A1) ET (X2 est A2) est appelée prémisse de la règle et la partie (Y est B) est appelée conclusion (conséquent) [20].

2.2 Ensembles flous de type-2 (EFT-2)

Un EFT-2 est une extension de l'ensemble flou Type-1, il est caractérisé par une fonction d'appartenance floue, c'est-à-dire, le degré d'appartenance de chaque élément de l'ensemble est lui-même un ensemble flou dans $[0, 1]$.

Un EFT-2 est un ensemble dans lequel nous avons également l'incertitude au niveau de la fonction d'appartenance.

Nous pouvons dire aussi que la logique floue de type-2 (LFT-2) est une généralisation de la logique floue classique (de type-1) (LFT-1) dans le sens que l'incertitude n'est pas seulement limitée dans les variables linguistiques, mais elle est également présente dans la définition des fonctions d'appartenance [16].

La FA d'un EFT-2 est donc tridimensionnel, et c'est cette nouvelle troisième dimension qui fournit de nouveaux degrés de liberté dans la conception pour traiter l'incertitude. De tels ensembles sont conseillés dans le cas où nous avons une incertitude au niveau de la valeur de l'appartenance elle-même.

Exemple : Supposons que l'on veuille déterminer une fonction d'appartenance triangulaire et que l'on soit indécis par rapport à la position exacte des deux points gauche et droit extrêmes de cette fonction. Supposons que 100 experts ont spécifié 100 points différents pour chacune des extrémités. Une meilleure façon de faire, serait de positionner ces points extrêmes à la moyenne de tous ces points, P_g et P_d , calculer l'écart-type de ces points, puis faire un "Blurring" comme montré sur la figure 2.2. [21].

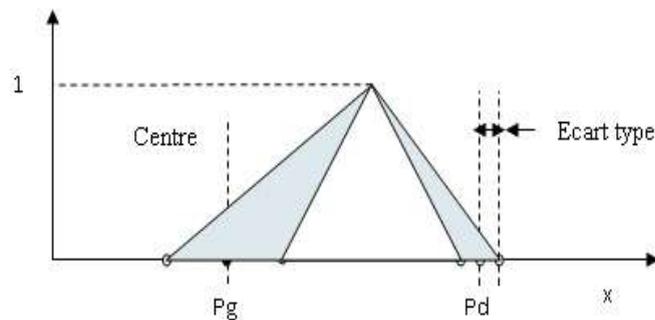


FIGURE 2.2: Blurring d'une MF.

2.2.1 Représentation d'un ensemble flou type-2 (EFT-2)

Une FAT-2 peut être vue comme une fonction à deux variables. Pour chaque x de l'univers du discours et degré d'appartenance primaire μ_1 , nous aurons une appartenance secondaire μ_2 qui est un nombre ordinaire dans l'intervalle $[0, 1]$. Ainsi, une FAT-2 peut être définie par

[23] :

$$\mu_2(x, \mu_1) = X \times [0, 1] \rightarrow [0, 1] \quad \forall x \in X \quad (2.9)$$

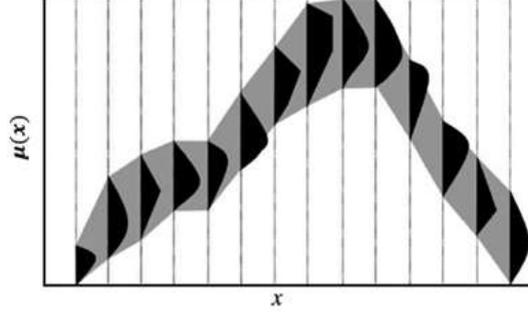


FIGURE 2.3: Représentation à 2.5D d'un ensemble flou type-2.

Définition 1 : L'EFT-2 \tilde{A} peut être représenté par la relation

$$\tilde{A} = ((x, u), \mu_{\tilde{A}}(x, u)) \mid \forall x \in X \quad \forall u \in J_x \subseteq [0, 1] \quad (2.10)$$

Où $\mu_{\tilde{A}}(x, u)$ est la fonction d'appartenance type-2 caractérisée par $0 \leq \mu_{\tilde{A}}(x, u) \leq 1$.

L'EFT-2 \tilde{A} peut être aussi définie par la relation :

$$\tilde{A} = \int_{x \in X} \int_{u \in J_x} \mu_{\tilde{A}}(x, u) / (x, u) \quad J_x \subseteq [0, 1] \quad (2.11)$$

Où \int désigne l'union de toutes x et u admissibles.

J_x est nommé la valeur d'appartenance primaire de x , en plus, il existe une valeur d'appartenance secondaire correspondant à chaque valeur d'appartenance primaire qui définit la possibilité pour l'adhésion primaire [21, 22].

Définition 2

A chaque valeur de x , soit $x = x'$, le plan 2D dont les axes sont u et $\mu_{\tilde{A}}(x', u)$ est appelé la partie verticale de $\mu_{\tilde{A}}(x, u)$. Une FA secondaire est une partie verticale de $\mu_{\tilde{A}}(x, u)$.

$$\mu_{\tilde{A}}(x = x', u) \text{ pour } x' \in X \text{ et } u \in J_{x'} \subseteq [0, 1]. \quad (2.12)$$

$$\mu_{\tilde{A}}(x = x', u) = \mu_{\tilde{A}}(x') = \int_{u \in J_{x'}} f_{x'}(u) / u \quad J_{x'} \subseteq [0, 1]. \quad (2.13)$$

où $0 \leq f_{x'}(u) \leq 1$. Nous référons à $\mu_{\tilde{A}}(x)$ comme une FA secondaire, c'est un EF type-1 "ensemble secondaire" [21, 22]. En se basant sur le concept des ensembles secondaires, on peut interpréter un EF type-2 comme l'union de tous les ensembles secondaires :

$$\tilde{A} = \{(x, \mu_{\tilde{A}}(x)) / \forall x \in X\} \quad (2.14)$$

$$\tilde{A} = \int_{x \in X} \mu_{\tilde{A}}(x)/x = \int_{x \in X} \left[\int_{u \in J_x} f_x(u)/u \right] /x \quad J_{x'} \subseteq [0, 1] \quad (2.15)$$

Le domaine d'une FA secondaire est appelé l'appartenance primaire de x . dans 2.15 J_x est l'appartenance primaire de x , ou $J_x \subseteq [0, 1] \quad \forall x \subseteq X$.

L'amplitude d'une FA secondaire est appelée un degré secondaire. Dans 2.15, $f_x(u)$ est un degré secondaire.

L'incertitude dans les appartenances primaires d'un EF Type-2, \tilde{A} , consiste en une région limitée que nous appellerons l'empreinte de l'incertitude (Footprint Of Uncertainty (FOU)). C'est l'union de toutes les appartenances primaires [21, 22] :

$$FOU(\tilde{A}) = \cup_{x \in X} J_x \quad (2.16)$$

2.2.2 Types d'ensemble flous type-2

Ensemble flou type-2 gaussien

Dans ce type d'ensembles, le degré d'appartenance de chaque point est un ensemble type-1 Gaussien dont le domaine de définition est inclus dans l'intervalle $[0, 1]$. Notons qu'il n'est pas nécessaire que la fonction d'appartenance principale soit aussi Gaussienne [16].

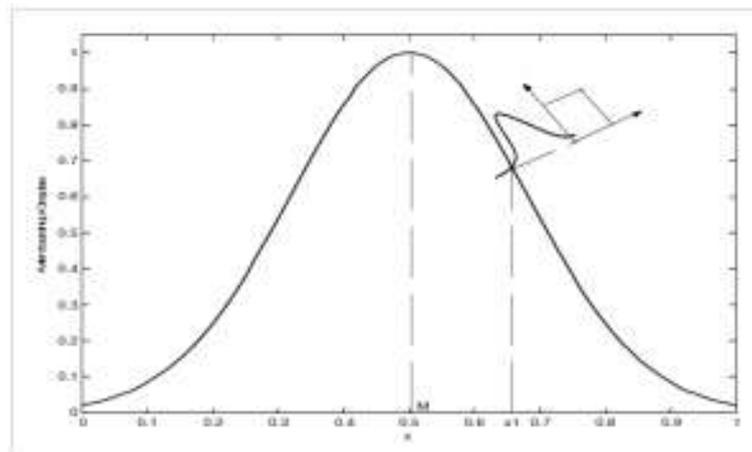


FIGURE 2.4: Ensemble flou type-2 gaussien.

Ensemble flou type-2 triangulaire

Dans ce type d'ensembles, le degré d'appartenance de chaque point est un ensemble type-1 triangulaire dont le domaine de définition est inclus dans l'intervalle $[0, 1]$ [23–25].

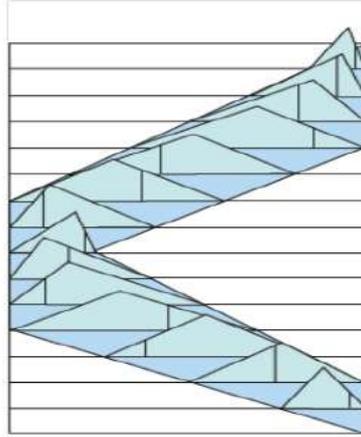


FIGURE 2.5: Représentation à 2.5D d'un ensemble flou type-2 triangulaire.

Ensemble flou type-2 intervalle

Une fonction d'appartenance floue de Type-2 intervalle peut être créée à partir de deux de MF type-1. Une fonction d'appartenance supérieure (UMF) qui représente la valeur maximale et une fonction d'appartenance inférieure (LMF), qui représente la valeur minimale de μ pour chaque x . L'incertitude U est représentée par la zone située entre la UMF et LMF. Cette région est appelée empreinte de l'incertitude (FOU) et est illustrée dans la Fig 2.6 [26].

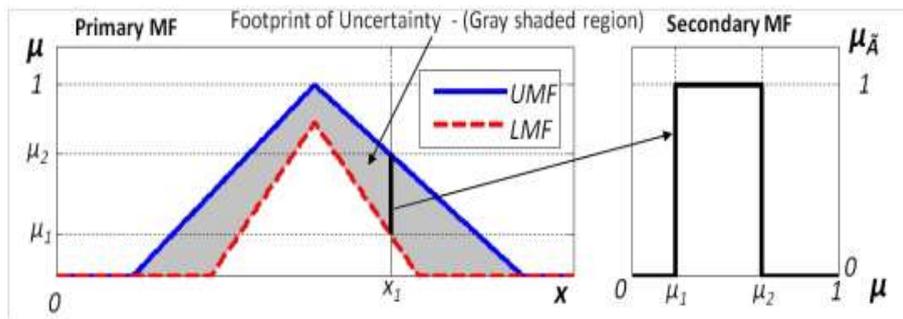


FIGURE 2.6: Ensemble flou type-2 intervalle.

Cette Figure représentation une fonction d'appartenance de type-2 intervalle lorsque tous $\mu_{\tilde{A}}(x, u)$ sont égales à 1, alors \tilde{A} est un intervalle T2FS. Dans ce cas particulier de l'équation 2.11 peut être définie par :

$$\tilde{A} = \int_{x \in X} \int_{u \in J_x} 1/(x, u) J_x \subseteq [0, 1] \quad (2.17)$$

Les chercheurs sont familiers à la complexité de calcul de T2FLS générales. Par conséquent, l'intervalle T2FLSs sont couramment utilisés dans la littérature. Les deux fonctions d'appartenance floues générales et intervalle types-2 sont à trois dimensions. Comme on peut le voir sur les figures précédentes, la seule différence entre elles est que la valeur d'appartenance secondaire de la fonction d'appartenance intervalle type-2 est toujours égale à 1.

2.3 Systèmes flous

La logique floue 'FL' est une extension de la logique conventionnelle (Boolean), elle a été étendue pour traiter des données permettant l'adhésion partielle à un ensemble plutôt que appartenance ou non appartenance à un ensemble.

La FL est une méthodologie de résolution de problème de contrôle de système. Elle fournit un moyen simple pour arriver à une conclusion définitive sur la base des informations d'entrée vagues, ambigus, imprécis, bruités, ou manquante.

Un système flou 'FLS' est un système expert qui utilise un ensemble de fonctions d'appartenances floues (FAs) et des règles pour traiter les données. Des règles Si-alors sont utilisées pour formuler les énoncés conditionnels compris dans le system.

Les règles dans les systèmes flous sont exprimées sous la forme suivante :

$$SI (x_1 \text{ est } A_1) \text{ et } (x_2 \text{ est } A_2) \text{ ALORS } (y \text{ est } B) \quad (2.18)$$

Ou x_1 et x_2 sont les variables linguistiques d'entrées, y et la variable linguistique de sortie. A_1, A_2 sont les valeurs linguistiques définies par les ensembles flous dans l'univers de discours U_1, U_2 . La partie $(x_1 \text{ est } A_1)$ et $(x_2 \text{ est } A_2)$ dans la règle est appelée antécédent ou prémisses, tandis que la partie $(y \text{ est } B)$ est appelée conséquent ou conclusion. B peut être un sous ensemble flou ou une fonction mathématique.

Un système flou qui est complètement décrit par des ensembles flous de type-1 est appelé un système flou de type-1, tandis qu'un FLS qui est décrit en utilisant au moins un ensemble flou de type-2 est appelé un FLS de type-2.

Les FLSs de Type-1 ne peuvent pas gérer directement les incertitudes de la règle, car ils utilisent des ensembles flous type-1 dont les fonctions d'appartenances sont certaines. Les FLSs de Type-2, d'autre part, sont très utiles dans des circonstances où il est difficile de déterminer une fonction d'appartenance exacte pour un ensemble flou ; par conséquent, ils peuvent être utilisés pour gérer les incertitudes de règles et même des incertitudes de mesure.

2.3.1 Système flou de type-1

Deux types de systèmes d'inférences floues ont été largement utilisés dans les applications. La différence entre ces deux systèmes d'inférence floue se trouve dans les conséquents de leurs règles floues, et donc leurs procédures d'agrégation des règles et de défuzzification.

Le système d'inférence floue de type Mamdani [27] a été proposé comme la première tentative de contrôler une machine à vapeur par un ensemble de règles linguistiques obtenues à partir des opérateurs humains expérimentés. Le modèle flou de Type Takagi-Sugeno-Kang TSK FLSs [28], se compose de règles avec des antécédents flous et de fonction mathématique dans la partie conséquente. Habituellement, la fonction de conclusion est en forme d'équation

linéaire dynamique. Les antécédents divisent l'espace d'entrée en un ensemble de régions floues, tandis que les conséquences décrivent le comportement du système dans ces régions.

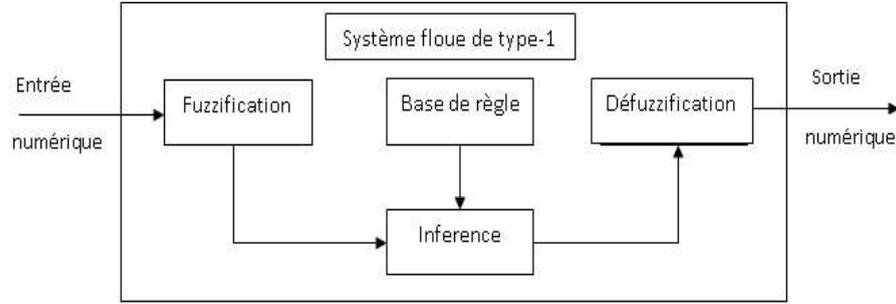


FIGURE 2.7: Structure d'un ensemble flou type-1.

Système flou de type-1 de TSK :

Un système flou de TSK de type-1 d'ordre 1 est décrit par des règles d'inférence SI-ALORS dont la forme est la suivante :

$$SI (x_1 \text{ est } A_{j1}) \text{ et } (x_2 \text{ est } A_{j2}) \text{ et...et}(x_n \text{ est } A_{jn}) \text{ ALORS } u_j = \sum_{i=1}^n w_{ij}x_i + b_j \quad (2.19)$$

Ou x_1, x_2, x_n sont les variables d'entrées, u_j sont les variables de sorties, A_{ij} sont les fonctions d'appartenance de type-1 pour la règle d'indice j et l'entrée d'indice i . les paramètres dans la partie conséquente des règles sont w_{ij} et b_j ($i = 1..N$ et $j = 1..M$). La sortie finale du système est obtenue comme suit :

$$u = \frac{\sum_{j=1}^M f_j u_j}{\sum_{j=1}^M f_j} \quad (2.20)$$

Ou f_j est donnée par :

$$f_j(x) = \mu_{A_{j1}}(x_1) * \dots * \mu_{A_{jn}}(x_n) \quad (2.21)$$

Dont $*$ représente l'opérateur de la T-norm [29].

2.3.2 Système flou de type-2

La structure de base d'un système flou de type-2, qui est représentée par la figure 2.8, est tout à fait similaire à un FLS de type-1, la seule différence étant que le bloc défuzzification dans les FLS de type-1 est remplacé par le bloc traitement de sortie, ce dernier contient deux blocs, un bloc 'Réduction de type' plus un bloc 'défuzzification'. Donc la structure d'un FLS de type-2 est composée de trois éléments à savoir : l'interface de fuzzification, le mécanisme d'inférence et le module de traitement de la Sortie.

Fuzzification

L'interface de fuzzification fait correspondre à une entrée déterministe (non floue) un

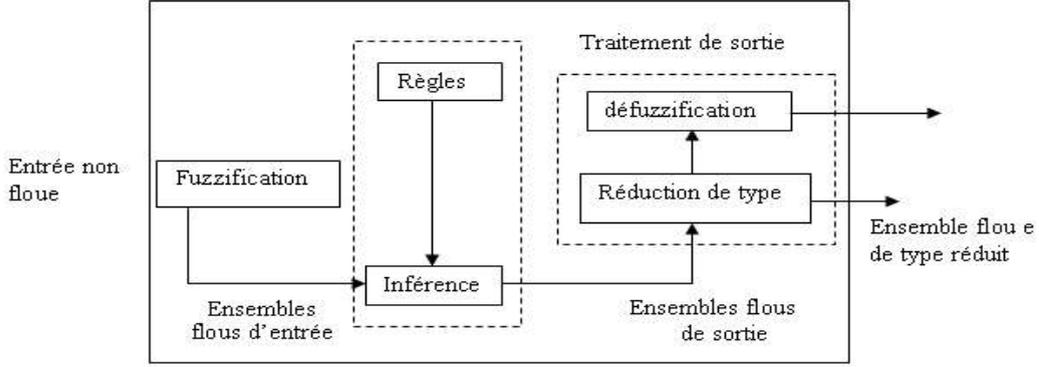


FIGURE 2.8: Structure de base d'un système flou type-2.

ensemble flou qui peut être généralement un EFT-2.

Règle d'inférence

La structure de la l^{ieme} règle dans un SFT-2 à la forme suivante :

$$R^l : si (x_1 \text{ est } \tilde{A}_1) \text{ et } (x_2 \text{ est } \tilde{A}_2) \text{ et...et}(x_p \text{ est } \tilde{A}_p) \text{ ALORS } y \text{ est } \tilde{G}^l \quad (2.22)$$

Ou x_i sont les entrées, \tilde{A}_i sont les ensembles antécédents ($i=1...p$), y est la sortie et \tilde{G}^l est l'ensemble conséquent. Dans cette dernière les (ou quelques) fonctions d'appartenance sont de type-2.

Inférence

Pour un FLS type-2, ayant p entrées $x_1 \in X_1, \dots, x_p \in X_p$, et une sortie $y \in Y$ et M règles, tel que la l^{ieme} règle a la forme R^l , cette règle représente une relation entre l'espace d'entrée $X_1 \times \dots \times X_p$ et l'espace de sortie Y du FLS. L'inférence consiste en première lieu de calculer l'intervalle d'activation associé au j^{ieme} ensemble flou de sortie :

$$F(\underline{x}) = \prod_{i=1}^p \mu_{\tilde{F}_i^j}(x_i) \quad (2.23)$$

$\mu_{\tilde{F}_i^j}(x_i)$ est l'intervalle d'activation associée à la variable x_i . Ensuite, si on note par \tilde{B}^l l'ensemble flou de sortie correspondant à la composition de la l^{ieme} règle R^l et l'ensemble flou d'entrée \tilde{X}^l , l'ensemble $F^l(\underline{x}')$ est combiné avec l'ensemble flou conséquent \tilde{G}^l de la l^{ieme} règle en utilisant les deux implications, le minimum ou le produit (équivalent à l'opération meet avec T-norm minimum ou produit dans le cas du type-2) pour obtenir l'ensemble flou de sortie correspondant à la l^{ieme} règle :

$$\mu_{\tilde{B}^l}(y) = \mu_{\tilde{G}^l}(y) \cap \mu_{\tilde{F}^l}(x') \quad (2.24)$$

En utilisant une fuzzification singleton, c'est-à-dire le degré d'appartenance pour l'ensemble flou \tilde{X}' n'a une valeur qui est unitaire que lorsque $x = x'$ alors :

$$\mu_{\tilde{B}'}(y) = \mu_{\tilde{G}'}(y) \cap \prod_{i=1}^n \mu_{\tilde{F}'}(x') \quad (2.25)$$

Réduction de type et défuzzification

La sortie du système d'inférence est un ensemble flou de type-2, d'après la structure d'un T2FLS la défuzzification ne se réalise qu'après l'opération de réduction de type. La formule générale de l'ensemble flou réduit par la méthode des centres d'ensembles est la suivant :

$$Y = (Y_1, \cdot, Y_M, F_1, \cdot, F_M) = \int_{y_1} \cdot \int_{y_M} \int_{f_1} \cdot \int_{f_M} \mathfrak{S}_{l=1}^M \mu_{Y_1}(y_1) * \mathfrak{S}_{l=1}^M \mu_{F_1}(f_1) / \frac{\sum_{l=1}^M f_l y_l}{\sum_{l=1}^M f_l} \quad (2.26)$$

où \mathfrak{S} et $*$ indiquent les t-norms utilisées (prod ou min), $f_l \in F_l$ et $y_l \in Y_l$ pour $l = 1, 2, \dots, M$. La majorité des travaux utilisent les IT2FLS au lieu des systèmes flous générales T2FLS a cause de la complexité de calcul dans ces derniers, les opérations se réduisent énormément lorsque les fonctions d'appartenance secondaires sont de type intervalle [30], Dans ce cas nous n'allons pas considérer toutes les combinaisons de y_l et de f_l .

Pour un système type-2 intervalle, chaque y_l et f_l ($l = 1, 2, \dots, M$) dans (2.26) est un ensemble type-1 intervalle, par conséquent nous avons $\mu_{Y_1}(y_1) = \mu_{F_1}(f_1) = 1$, donc (2.26) devient :

$$Y_{cos} = (Y_1, \cdot, Y_M, F_1, \cdot, F_M) = \int_{y_1} \cdot \int_{y_M} \int_{f_1} \cdot \int_{f_M} 1 / \frac{\sum_{l=1}^M f_l y_l}{\sum_{l=1}^M f_l} = [y_l, y_r] \quad (2.27)$$

Par conséquent il suffit de calculer l'expression :

$$\frac{\sum_{l=1}^M f_l y_l}{\sum_{l=1}^M f_l} \quad (2.28)$$

Puisque tous les degrés d'appartenances dans un ensemble type-1 intervalle sont ordinaires, donc chaque intervalle est représenté par ses limites gauche et droite [l, r], ou par son centre et sa largeur $\frac{l+r}{2}$ et $\frac{r-l}{2}$ respectivement. Pour le calcul de y_l, y_r Karnil et Mendel ont proposé un algorithme itératif 'KM algorithm' dont le détail est présenté en annexe1 [32]. En appliquant le centre de gravité au type réduit, la sortie numérique sera calculée par l'expression suivante :

$$Y = \frac{y_l + y_r}{2}$$

Avec y_l et y_r peuvent être données sous la forme

$$y_l = \frac{\sum_{j=1}^M f_l^j y_l^j}{\sum_{j=1}^M f_l^j} = \sum_{j=1}^M y_l^j \zeta_l^j = \underline{y}_l^T \underline{\zeta}_l(\underline{x})$$

$$y_r = \frac{\sum_{j=1}^M f_r^j y_r^j}{\sum_{j=1}^M f_r^j} = \sum_{j=1}^M y_r^j \zeta_r^j = \underline{y}_r^T \underline{\zeta}_r(\underline{x})$$

Ou f_l^j, f_r^j sont les degrés d'activation des règles flous type-2 contribuant au points extrêmes de gauche et droite.

Système flou de type-2 de TSK :

Les systèmes flous de TSK sont introduits par T. Takagi, M. Sugeno et K.T. Kang en 1984. Ces derniers sont caractérisés par l'utilisation d'une fonction des variables d'entrées dans la partie conséquente des règles d'inférences au lieu d'un ensemble flou comme dans le cas des systèmes de Mamdani. Les systèmes flous de type-2 de TSK présentent l'extension des systèmes flous de TSK de type-1, Liang et Mendel ont proposé l'architecture de trois modèles de ces systèmes qui diffèrent dans la nature des prémisses (A, Antécédent) et des conclusions (C)[31, 34].

Modèle I

Le modèle IT2FLS de TSK est décrit par des règles IF-then dont la partie prémisse est des ensembles flous de type-2 et la partie conséquente est des ensembles flous de type-1. Ces modèles sont représentés dans la littérature par A2-C1. La règle d'inférence du modèle A2-C1 est énoncée comme suit :

$$SI (x_1 \text{ est } \tilde{A}_{j1}) \text{ et } (x_2 \text{ est } \tilde{A}_{j2}) \text{ et...et}(x_n \text{ est } \tilde{A}_{jn}) \text{ ALORS } y_j = \sum_{i=1}^n w_{ij}x_j + B_j \quad (2.29)$$

Ou x_1, x_1, \dots, x_n sont les variables d'entrées.

y_j sont les variables de sorties.

\tilde{A}_{ji} est l'ensemble flou type-2 de la j^{eme} règle et la i^{eme} entrée.

Les paramètres de conséquence w_{ij} et B_j ($i = 1..N, j = 1..M$) sont les ensembles flous type-1.

La sortie globale du modèle est donnée par la relation :

$$Y(Y_1, \dots, Y_M, F_1, \dots, F_M) = \int_{y_1} \cdot \int_{y_M} \int_{f_1} \cdot \int_{f_M} \tau_{j=1}^M \mu_{Y_1}(y_1) * \tau_{j=1}^M \mu_{F_1}(f_1) / \frac{\sum_{j=1}^M f_j y_j}{\sum_{j=1}^M f_j} \quad (2.30)$$

Avec M est le nombre de règle mis en jeu.

$y_j \in Y_j, f_j \in F_j$ et τ indique la T-norm.

F_j est définie par

$$F_j = \mu_{\tilde{A}_{j1}}(x_1) \sqcap \mu_{\tilde{A}_{j2}}(x_2) \dots \sqcap \mu_{\tilde{A}_{jM}}(x_M) \quad (2.31)$$

Ou \sqcap représente l'opérateur meet.

Un cas pratique pour le calcul de la formule 2.29 est lorsque les ensembles type-2 intervalle sont utilisés dans la partie antécédent, et des ensembles flous type-1 intervalle sont utilisés dans la partie conséquente, ce qui signifie que U_i et F_i ($i = 1..M$) sont des ensembles flous type-1 intervalle ce qui veut dire que $\mu_{Y_1}(y_1) = 1$ et $\mu_{F_1}(f_1) = 1$, alors 2.29 se simplifie a :

$$Y(Y_1, \dots, Y_M, F_1, \dots, F_M) = \int_{y_1} \dots \int_{y_M} \int_{f_1} \dots \int_{f_M} 1 / \frac{\sum_{j=1}^M f_j y_j}{\sum_{j=1}^M f_j} \quad (2.32)$$

avec $Y = [y_l, y_r]$, $F_i = [f_l, f_r]$, $Y_i = [y_l^i, y_r^i]$, et ($i = 1..M$).

le centre c_i de F_i est $c_i = (f_l^i + f_r^i)/2$.

On suppose que tous les ensembles flous sont des valeurs exactes.

$$\frac{\sum_{j=1}^M f_j y_j}{\sum_{j=1}^M f_j} \quad (2.33)$$

y_l et y_r sont calculés par l'algorithme de Karnil et Mendel [32, 87].

Modèle II

Ce modèle est identique au précédent au niveau des antécédent, mais au niveau des conséquences les ensembles utilisés sont non flous, ces modèles sont représentés par (A2-C0) La règle R^j est donnée par :

$$SI (x_1 \text{ est } \tilde{A}_{j1}) \text{ et } (x_2 \text{ est } \tilde{A}_{j2}) \text{ et...et}(x_n \text{ est } \tilde{A}_{jn}) \text{ ALORS } y_j = \sum_{i=1}^n w_{ij} x_j + B_j \quad (2.34)$$

Ou x_1, x_2, \dots, x_n sont les variables d'entrées.

y_j sont les variables de sorties.

\tilde{A}_{ji} est l'ensemble flou type-2 de la j_{eme} règle et la i_{eme} entrée.

Les paramètres de conséquence w_{ij} et $B_j (i = 1..n, j = 1..M)$ sont des valeurs non flous. La sortie du modèle est donnée par la relation :

$$Y(F_1, \dots, F_M) = \int_{f_1} \dots \int_{f_M} \tau_{j=1}^M \mu_{F_1}(f_j) / \frac{\sum_{j=1}^M f_j y_j}{\sum_{j=1}^M f_j} \quad (2.35)$$

Avec M est le nombre de règle mis en jeu.

Si on utilise des ensembles flous type-2 intervalle dans la partie prémisse l'équation 2.34 se réduit à la forme suivante :

$$Y(F_1, \dots, F_M) = \int_{f_1 \in [\underline{f}_1, \bar{f}_1]} \dots \int_{f_j \in [\underline{f}_j, \bar{f}_j]} 1 / \frac{\sum_{j=1}^M f_j y_j}{\sum_{j=1}^M f_j} \quad (2.36)$$

ou $\underline{f}_j, \bar{f}_j$ sont données par

$$\underline{f}_j = \underline{\mu}_{\tilde{A}_{j1}}(x_1) * \underline{\mu}_{\tilde{A}_{j2}}(x_2) * \dots * \underline{\mu}_{\tilde{A}_{jn}}(x_n) \quad (2.37)$$

$$\bar{f}_j = \bar{\mu}_{\tilde{A}_{j1}}(x_1) * \bar{\mu}_{\tilde{A}_{j2}}(x_2) * \dots * \bar{\mu}_{\tilde{A}_{jn}}(x_n) \quad (2.38)$$

ou * représente de la T-norm choisis.

La sortie du système flou en boucle fermée est donnée par [31] :

$$Y_{TSK/A2-C0} = \frac{\sum_{j=1}^M \underline{f}_j y_j}{\sum_{j=1}^M \underline{f}_j + \sum_{j=1}^M \bar{f}_j} + \frac{\sum_{j=1}^M \bar{f}_j y_j}{\sum_{j=1}^M \underline{f}_j + \sum_{j=1}^M \bar{f}_j} \quad (2.39)$$

La procédure de calcul de u_l et u_r est basée sur l'algorithme itératif de Karnil-Mende 'KM', cette approche ne peut être formulée en boucle fermée pour l'étude de la stabilité, dans [33] les auteurs ont proposé un nouveau mécanisme d'inférence pour les systèmes A2-C0 de type IT2TSK, cette nouvelle expression peut remplacer la réduction de type.

$$Y_{TSK-2} = \frac{q \sum_{i=1}^M \underline{f}^i y_i}{\sum_{i=1}^M \underline{f}^i} + \frac{(1-q) \sum_{i=1}^M \bar{f}^i y_i}{\sum_{i=1}^M \bar{f}^i} \quad (2.40)$$

Le paramètre q est un facteur de conception qui indique la contribution des valeurs inférieures et supérieures à la sortie finale. q permet aux concepteurs d'ajuster la portion inférieure ou supérieure selon le niveau d'incertitude de leur système.

Modèle III

Le modèle III est aussi un cas particulier du modèle A2-C1 il diffère dans la partie prémisses des règles d'inférence ou des ensembles flous type-1 sont utilisés au lieu des ensembles flous type-2 d'où la nomination (A1-C1). La règle d'inférence R^j dans ce modèle est décrite par :

$$SI (x_1 \text{ est } \tilde{A}_{j1}) \text{ et } (x_2 \text{ est } \tilde{A}_{j2}) \text{ et...et}(x_n \text{ est } \tilde{A}_{jn}) \text{ ALORS } y_j = \sum_{i=1}^n w_{ij} x_i + B_j \quad (2.41)$$

Ou x_1, x_2, \dots, x_n sont les variables d'entrées.

y_j sont les variables de sorties.

\tilde{A}_{ji} est l'ensemble flou type-1 de la j_{eme} règle et la i_{eme} entrée.

Les paramètres de conséquence w_{ij} et $B_j (i = 1..n, j = 1..M)$ sont des valeurs flous type-1.

La sortie du modèle est donnée par la relation :

$$Y(Y_1, \dots, Y_M) = \int_{y_1} \dots \int_{y_M} \tau_{j=1}^M \mu_{Y_1}(y_1) / \frac{\sum_{j=1}^M f_j y_j}{\sum_{j=1}^M f_j} \quad (2.42)$$

ou M est le nombre de règle excitée et $y_j \in Y_j$. f_j est le poids de la règle définie par :

$$f_j = \mu_{\tilde{A}_{j1}}(x_1) * \mu_{\tilde{A}_{j2}}(x_2) * \dots * \mu_{\tilde{A}_{jM}}(x_M) \quad (2.43)$$

2.4 Conclusion

Dans ce chapitre, nous avons présenté les concepts de base de la logique floue de type-1, depuis la définition des ensembles flous jusqu'aux systèmes d'inférences flous qui s'applique actuellement dans de nombreux applications dans divers domaine, cela résulte du pouvoir expressif en terme linguistique de l'être humain à décrire les phénomènes.

Cette expression linguistique est traduite par des règles de la forme SI-ALORS, cela simplifie énormément la description du comportement d'un processus complètement inconnu.

Les blocs de conception d'un système d'inférence flou de type-1 sont aussi présentés dans ce chapitre, la limitation dans la prise en charge des incertitudes dans ces systèmes a mené à une extension qui prend en considération les incertitudes et imprécision même au niveau des fonctions d'appartenances, la logique flou de type-2 basée sur les ensembles flous type-2 fournis la mesure de dispersion de l'incertitude linguistique.

Les outils de base des ensembles flous type-2 sont présentés ainsi que l'architecture des systèmes d'inférence flous type-2, dans laquelle nous avons vu où réside la différence avec celle des systèmes flous de type-1.

Le cas particulier des systèmes d'inférences flous type-2 intervalle permet de simplifier le calcul dans la conception des systèmes flous type-2 sont aussi présentés.

La dernière partie de ce chapitre présente les différents modèles développés par Mendel et Liang des systèmes flous de type TSK, le modèle A2-C0 avec le nouveau mécanisme d'inférence qui est adapté pour l'étude de la stabilité est choisi dans cette thèse pour l'analyse de la stabilité au sens de Liapunov.

La conception de contrôleur par logique flou de type-1 ou de type-2 contient toujours des étapes où il n'existe aucune règle de choix, par exemple la forme des fonctions d'appartenance ou le nombre de règles pour un problème donné, de ce fait les chercheurs se sont orientés vers les méthodes d'apprentissage automatique des contrôleurs flous en leurs introduisant les réseaux de neurones, les algorithmes génétiques, les réseaux d'ondelettes..etc. Dans notre cas les réseaux de neurones sont utilisés pour le réglage des paramètres des systèmes flous, on conçoit ainsi un système neuro-flou, ces systèmes sont introduis dans le chapitre suivant.

Chapitre 3

Systèmes neuro-flous

DANS le chapitre précédent ont été présentés les concepts principaux des systèmes flous de type-1 et de leurs extensions à ceux de type-2. La commande par logique floue est bénéfique pour atteindre les objectifs de réglage désiré en employant des règles linguistiques de la forme SI-ALORS. Cependant, les caractéristiques des outils utilisés pour la conception de cette commande manquent de méthode formelle pour être déterminés (fonctions d'appartenance et règles floues). Dans ce sens, il serait intéressant de disposer d'algorithmes permettant l'apprentissage automatique de ces paramètres. L'une des approches qui permette de répondre à ce besoin est les réseaux de neurones. Donc il est utile de fusionner les réseaux de neurones, et les systèmes flous pour la détermination de leurs paramètres optimaux, en utilisant la capacité d'apprentissage des réseaux de neurones, les systèmes résultants vont bénéficier aussi de la lisibilité et la souplesse des systèmes flous.

Les origines des réseaux de neurones sont très diversifiées. En 1943, McCulloch et Pitts, lors d'essais de modélisation du neurone biologique, ils supposaient que l'impulsion nerveuse était le résultat d'un calcul simple effectué par chaque neurone et que la pensée naissait grâce à l'effet collectif d'un réseau de neurones interconnectés.

En 1949, Hebb, dans une perspective psychophysologique, a proposé la première règle d'apprentissage, son idée était que si deux neurones étaient actifs en même temps alors les synapses entre ces deux seraient renforcées.

En 1958 que Rosenblatt décrivait le premier modèle opérationnel de réseaux de neurones, mettant en oeuvre les idées de Hebb, McCulloch et Pitts : le perceptron, inspiré du système visuel, capable d'apprendre à calculer certaines fonctions logiques en modifiant ses connexions synaptiques. Ce modèle suscita beaucoup de recherche, et sans doute trop d'espoirs.

Les techniques neuronales ont trouvé un nouvel essor grâce aux travaux du groupe de recherche de David Rumelhart à San Diego [35] et ceux de Yann Le Cun en France [36]. Ces chercheurs ont proposé l'algorithme de rétropropagation du gradient pour les réseaux multicouches, ces derniers étant une extension du Perceptron. Depuis une dizaine d'années,

l'intérêt pour les RNA a considérablement augmenté et ils sont sortis du domaine de la recherche pour aller vers des domaines applicatifs [37].

Un réseau de neurones est un réseau d'éléments de traitement interconnecté, les éléments de traitement interagissent le long des chemins sur les forces de connexion variables qui, lorsqu'ils sont convenablement adaptés peuvent produire collectivement le comportement complexe global souhaité.

Les réseaux de neurones sont des outils d'intelligence artificielle, capables d'effectuer des opérations de classification. Leur fonctionnement est basé sur les principes de fonctionnement des neurones biologiques. Leur principal avantage par rapport aux autres outils est leur capacité d'apprentissage et de généralisation de leurs connaissances à des entrées inconnues. Les réseaux de neurones peuvent être également implémentés en circuits électroniques, offrant ainsi la possibilité d'un traitement en temps réel. Le processus d'apprentissage est donc une phase très importante pour la réussite d'une telle opération (classification, identification, reconnaissance de caractères, de la voix, vision, système de contrôle. . .) [38]. Dans la littérature on trouve plusieurs problèmes pratiques où les réseaux de neurones ont été appliqués :

- Radar et la classification de signal du sonar. Les réseaux de neurones peuvent distinguer entre les différents types de déclarations de radar (météo, oiseaux, avions) avec plus de précision que les systèmes conventionnels [39].
- Conversion de la parole en texte. L'auteur dans [40] a conçu un netTALK, célèbre programme qui a appris à lire le texte et de le convertir à la parole.
- Dans les applications biomédicales les réseaux de neurones trouvent de nombreuses utilisations dans la classification et les applications de diagnostic médical [41].
- La reconnaissance de caractères. Security Pacific à Los Angeles utilise un NN pour détecter les modes d'utilisation de la carte de crédit frauduleuse par observation des changements dans les habitudes d'achat [42].
- Anti-virus. IBM utilise un système neuronale pour la détection les nouveaux virus automatiquement grâce à un "artificielle immunitaire" pour les ordinateurs [43].
- Contrôle de processus. La société Texas Eastman a été en mesure de réduire l'utilisation d'un additif coûteux a un tiers à son usine chimique [44].

De nombreux auteurs ont donc tout naturellement cherché à combiner ces deux approches depuis le début des années 90, certaines méthodes d'auto-apprentissage ont été proposés tels que les réseaux neuro-flous [45, 46] à l'aide de l'algorithme de retropropagation [35], l'apprentissage floue en utilisant la fonction à base radiale (RBF) [47, 48], Algorithme génétique (GA) pour décider des formes de fonctions d'appartenance et des règles floues [49, 50], et la méthode de la descente de gradient [51]. Ces méthodes peuvent apprendre plus vite que les réseaux de neurones; Cependant, l'opérateur doit déterminer le nombre et la forme des fonctions d'appartenance avant l'apprentissage.

Ce deuxième chapitre est organisé en deux parties. La première partie est consacrée à une brève présentation des notions générales des réseaux de neurones, nous présentons les

structures neuronales les plus utilisées. Nous abordons également le problème d'apprentissage des paramètres de ces structures, nous nous intéressons particulièrement à l'algorithme de rétro-propagation.

Nous présentons dans la deuxième partie la technique d'hybridation entre les RNs et la logique floue, employée fréquemment pour la commande des systèmes complexes.

3.1 Modèles de neurone

L'unité centrale de l'être humain est son cerveau qui est l'organe le plus complexe. Cet organe lui permet d'analyser et de comprendre les différents phénomènes qui l'entourent. Cette merveilleuse machine est composée des cellules nerveuses hautement spécialisées appelées neurones.

a) Neurone biologique

C'est la cellule de base composant le cerveau, le neurone est une cellule composée d'un corps cellulaire et d'un noyau. Le corps cellulaire se ramifie pour former ce que l'on nomme les dendrites. Celle-ci est parfois si nombreuse que l'on parle de chevelure dendritique ou d'arborisation dendritique. C'est par les dendrites que l'information est acheminée de l'extérieur vers le soma, corps du neurone.

L'information traitée par le neurone chemine ensuite le long de l'axone (unique) pour être transmise aux autres neurones. La transmission entre deux neurones n'est pas directe. La jonction entre deux neurones est appelée la synapse (Fig. 3-1) [52].

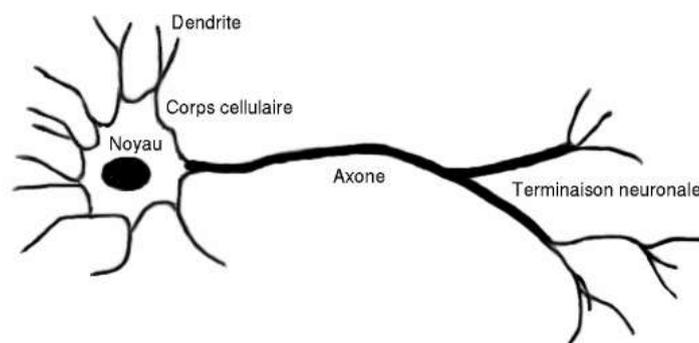


FIGURE 3.1: Le neurone biologique.

b) Neurone formel

La figure ci-dessous montre la structure d'un neurone artificiel, appelé aussi perceptron. Chaque neurone artificiel est un processus élémentaire. Il reçoit un nombre variable d'entrée

en provenance de neurone amont. A chacune de ces entrées est associé un poids w représentatif de la force de la connexion. Chaque processeur élémentaire est doté d'une sortie unique, qui se ramifie ensuite pour alimenter un nombre variable de neurones avals. A chaque connexion est associée un poids [52].

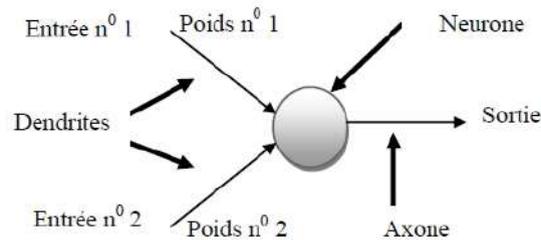


FIGURE 3.2: Le neurone artificiel.

Le modèle de base d'un neurone artificiel est constitué d'un sommateur pondéré et une fonction d'activation (ou transfert) comme le montre la figure 3.3. Cette figure montre un neurone dans la j ème couche.

$$a_j = \sum_{i=1}^N w_{ji}x_i - b_j \quad (3.1)$$

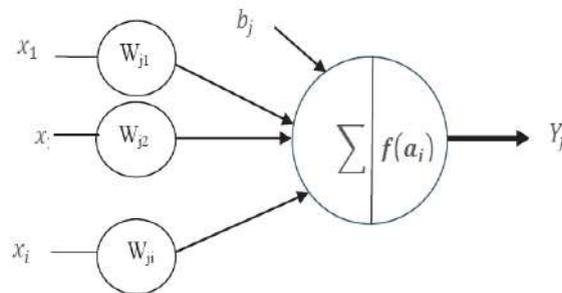


FIGURE 3.3: Modèle de base d'un neurone artificiel.

- a_j : Son état (aussi appelée activation), cet état est généralement choisi comme valeur de sortie du neurone.
- w_{ji} : les poids synoptiques associés aux entrées.
- x_i : les entrées.
- b_j : biais.

A partir de cette valeur, une fonction de transfert calcule la valeur de l'état du neurone.

$$Y_j = f(a_j) \quad (3.2)$$

- Y_j : la sortie.

- f : sa fonction d'activation.

C'est cette valeur qui sera transmise aux neurones avals, la plupart des fonctions de transferts sont continues, offrant une infinité de valeurs possibles comprises dans l'intervalle $[0, 1]$ (ou $[-1, +1]$), elle doit être différentiable. Il existe des nombreuses formes possibles pour la fonction de transfert. Les plus courantes sont présentées sur le tableau suivant.

TABLE 3.1: Les fonctions d'activation usuelles

nom de fonction	Relation d'entrée / sortie
Seuil	$a = 0$ si $n < 0$ $a = 1$ si $n \geq 0$
Seuil symétrique	$a = -1$ si $n < 0$ $a = 1$ si $n \geq 0$
Linéaire	$a = 0$
Linéaire saturée	$a = 0$ si $n < 0$ $a = n$ si $0 \leq n \leq 1$ $a = 1$ si $n > 1$
Linéaire saturée symétrique	$a = -1$ si $n < -1$ $a = n$ si $0 \leq n \leq 1$ $a = 1$ si $n > 1$
Linéaire positive	$a = 0$ si $n < 0$ $a = n$ si $0 \leq n \leq 1$
Sigmoïde	$a = \frac{1}{1+e^{-n}}$
Tangente hyperbolique	$a = \frac{e^n - e^{-n}}{e^n + e^{-n}}$
Compétitive	$a = 1$ si n maximum $a = 0$ si autrement

3.2 Les réseaux de neurones

Les réseaux de neurones sont composés d'éléments simples (ou neurones) fonctionnant en parallèle. Le fonctionnement du réseau de neurone est fortement influencé par la connexions des éléments entre eux. On peut entraîner un réseau de neurone pour une tâche spécifique (reconnaissance de caractères par exemple) en ajustant les valeurs des connexions (ou poids) entre les éléments (neurone). Le premier RNA été le perceptron de Rosenblatt, il représente le réseau le plus simple (Le perceptron monocouche), puisque il ne se compose que d'une couche d'entrée et d'une couche de sortie. Minsky et Papert en 1969 ont démontré mathématiquement que le perceptron simple couche avait des limites fondamentales sur ce qu'il pourrait calculer [53]. Les perceptrons multicouches (MLP, PMC 'multi layer perceptron') introduisent une ou plusieurs couches cachées, dont les noeuds de calcul sont appelés en conséquence neurones cachés.

3.2.1 Le perceptron multicouche (PMC)

C'est une extension du précédent, avec une ou plusieurs couches cachées entre l'entrée et la sortie. Chaque neurone dans une couche est connecté à tous les neurones de la couche précédente et de la couche suivante (excepté pour les couches d'entrée et de sortie) et il n'y a pas de connexions entre les cellules d'une même couche. Les fonctions d'activation utilisées dans ce type de réseaux sont principalement les fonctions à seuil ou sigmoïdes. Il peut résoudre des problèmes de la commande non linéaire.

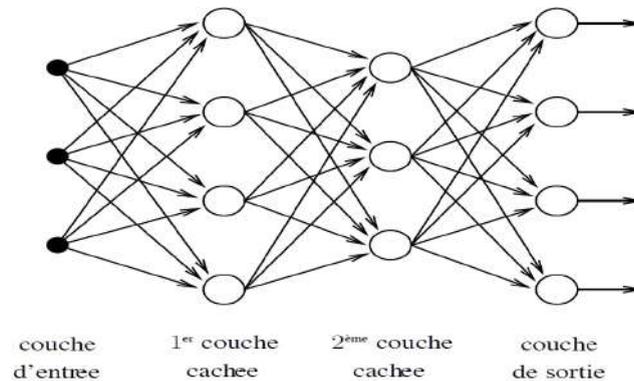


FIGURE 3.4: Le perceptron multicouche.

3.2.2 Structure d'interconnexion

L'architecture d'un réseau de neurone peut être considérée comme un graphe orienté et pondéré dans lequel les neurones artificiels sont les noeuds et les arcs orientés (avec poids) sont des connexions entre les neurones de sorties et les neurones d'entrées.

Basé sur le modèle de connexion (architecture), les RNs peuvent être regroupées en deux catégories fig.3.5

- Les réseaux à connexion locale 'réseaux feed-forward' dans lesquelles l'architecture ne contient aucune boucle.
- Les réseaux récurrents (ou rétroaction), qui contiennent des boucles à cause des connexions de rétroaction.

Les différentes connectivités donnent des comportements différents du réseau. D'une manière générale, les réseaux feed-forward sont des réseaux statiques, ce qui signifie qu'ils produisent un seul ensemble de valeurs de sortie plutôt qu'une séquence de valeurs à partir d'une entrée donnée. Ces réseaux fonctionnent comme des petites mémoires, dans le sens où leur réponse à une entrée est indépendante de l'état précédent du réseau. D'autre part, les réseaux Récurrents sont des systèmes dynamiques. Quand un nouveau échantillon est présenté à l'entrée, les sorties des neurones sont calculées. À cause des boucles de réaction, les entrées de chaque neurone sont alors modifiées, ce qui conduit le réseau à entrer dans un nouvel état [54].

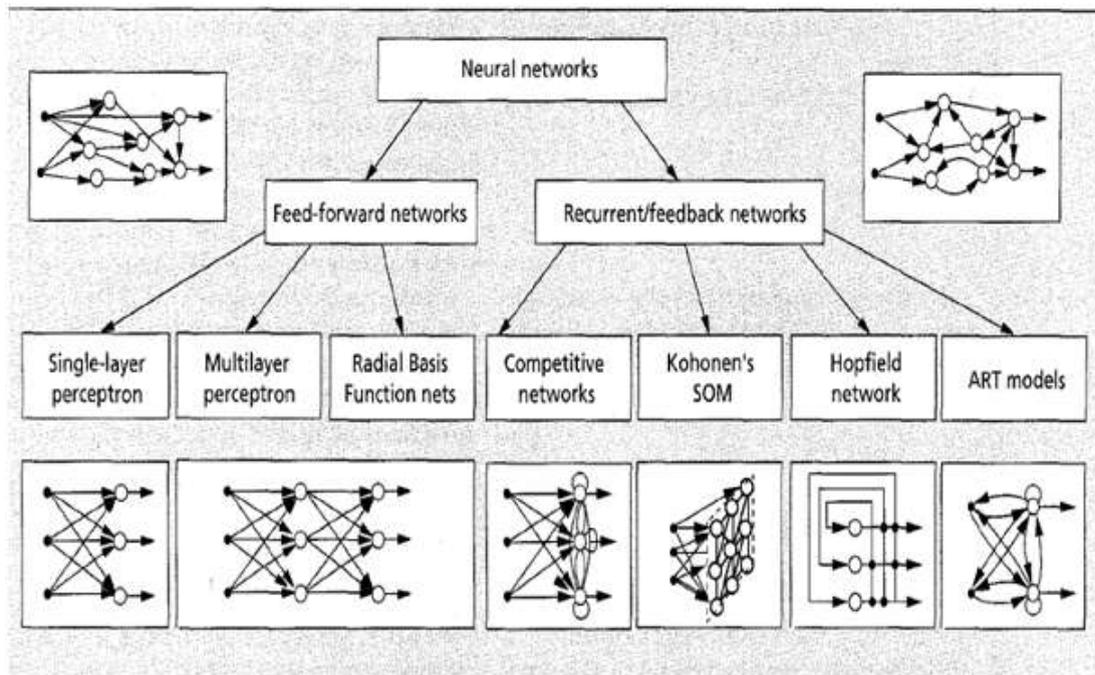


FIGURE 3.5: Classification des architectures des réseaux de neurones.

3.2.3 Apprentissage

L'apprentissage est la phase qui vient juste après la conception de la structure des réseaux de neurone, l'une des définitions de cette opération est donnée dans [52] par :

« *L'apprentissage est une phase du développement d'un réseau de neurones durant laquelle le comportement du réseau est modifié jusqu'à l'obtention du comportement désiré. L'apprentissage neuronal fait appel à des exemples de comportement* ».

A cette opération on fait toujours correspondre un algorithme d'apprentissage. A l'aide de cet algorithme le réseau ajuste ces poids de liaison 'poids synaptiques' à ce que sa réponse soit celle désiré par les exemples fournis au réseau. Selon la forme des données fournies au réseau, on peut distinguer en générale trois catégories de processus d'apprentissage :

- ◆ Apprentissage non supervisé.
- ◆ Apprentissage supervisé.
- ◆ Apprentissage compétitive.

a) Apprentissage non supervisé

Dans ce type d'apprentissage, le réseau peut apprendre sans connaître les sorties désirées. Il modifie ces poids en fonction d'un critère interne. Ce type d'apprentissage est plus rapide et moins complexe en terme de calcul que l'apprentissage supervisé, donc il est adapté pour les systèmes rapides ou dans le cas où l'on a une insuffisance d'information.

b) Apprentissage supervisé

Un superviseur, ou professeur, fournit au réseau des couples d'entrées-sorties. Il fait apprendre au réseau l'ensemble de ces couples, par une méthode d'apprentissage, comme la rétro-propagation du gradient de l'erreur, en comparant pour chacun d'entre eux la sortie effective du réseau et la sortie désirée. L'apprentissage se termine lorsque tous les couples entrées-sorties sont reconnus par le réseau. Ce type d'apprentissage se retrouve, entre autres, dans le perceptron.

b) Apprentissage compétitive

Consiste à faire compétitionner les neurones d'un réseau pour déterminer lequel sera actif à un instant donné. Contrairement aux autres types d'apprentissage ou, généralement, tous les neurones peuvent apprendre simultanément et de la même manière, l'apprentissage compétitif produit un « vainqueur » ainsi que, parfois, un ensemble de neurones « voisins » du vainqueur, et seuls ce vainqueur et, potentiellement, son voisinage bénéficient d'une adaptation de leur poids. On dit alors que l'apprentissage est local car limité à un sous-ensemble des neurones du réseau [55].

3.3 Algorithme d'apprentissage

Les algorithmes d'apprentissage dans le cadre de l'intelligence artificielle sont à classer parmi les méthodes non paramétriques, la méthode d'apprentissage la plus populaire pour l'entraînement des perceptrons multicouches est l'algorithme de rétro-propagation. Ce dernier utilise une méthode itérative de descente de gradient qui minimise l'erreur quadratique moyenne (norme L^2) entre la sortie désirée et la sortie de réseau (apprentissage supervisé), l'algorithme de rétro-propagation du gradient comporte deux phases distinctes :

- ◆ **Propagation** :à chaque étape, on présente au réseau un exemple en entrée. Cette entrée est propagée jusqu'à la couche de sortie.
- ◆ **Correction** :à-coup sûr, le réseau ne fournira pas exactement ce que l'on attendait, on calcule donc une erreur (en général la somme quadratique moyenne des erreurs pour tous les neurones de sortie) que l'on rétro- propage dans le réseau. Ce processus est interrompu dès que l'erreur globale est estimée suffisante .

l'algorithme se présente comme suit :

Etape 1 :initialisation des poids W_{ji} .

Etape 2 :application du vecteur d'entrée $x_p = (x_{p1}, x_{p2}, x_{p3}, \dots, x_{pn})^T$ à la couche d'entrée.

Etape 3 :calcul des entrées des neurones de la couche cachée :

$$a_{pj} = \sum_{i=1}^N W_{ji} X_{pi} + b_j \quad (3.3)$$

Etape 4 :Calcul des sorties des neurones de la couche cachée

$$X_{pj} = f(a_{pj}) \quad (3.4)$$

Etape 5 :passer à la couche de sortie et calculer les entrées de chaque neurone

$$a_{pk} = \sum_{j=1}^L W_{kj} X_{pj} + b_k \quad (3.5)$$

Etape 6 :calcul des sorties des neurones de la couche de sortie

$$y_{pk} = f_k(a_{pk}) \quad (3.6)$$

Etape 7 :calcul de l'erreur de la couche de sortie

$$\delta_{pk} = (y_{pk}^d - y_{pk}) \cdot f'_k(a_{pk}) \quad (3.7)$$

Etape 8 :calcul de l'erreur de la couche cachée

$$\delta_{pj} = f'_j(a_{pj}) \sum_k \delta_{pk} W_{kj} \quad (3.8)$$

L'erreur de la couche cachée est calculée avant la modification des poids de la couche de sortie. Les poids vont être ajustés par une variation dont la direction optimale est donnée par l'opposé du gradient de l'erreur par rapport aux poids.

Etape 9 :modification des poids de la couche de sortie

$$W_{kj}(t+1) = W_{kj}(t) + \eta \delta_{pk} y_{pj} + \alpha \Delta W_{kj}(t) \quad (3.9)$$

Etape 10 :Modification des poids de la couche cachée

$$W_{ji}(t+1) = W_{ji}(t) + \eta \delta_{pj} x_i + \alpha \Delta W_{ji}(t) \quad (3.10)$$

Etape 11 :Calcul de l'erreur globale

$$E_p = \frac{1}{2} \sum_{k=1}^M \delta_{pk}^2 \quad (3.11)$$

Continuer l'algorithme jusqu'à l'obtention de $E_p < E_{pseuil}$, et cela pour chaque paire entrée sortie (x_p, y_p).

Critère d'arrêt

1. Le plus utilisé consiste à fixer un nombre maximum de périodes d'entraînement, ce qui fixe effectivement une limite supérieure sur la durée de l'apprentissage.
2. Un deuxième critère peut être utilisé est celui de fixer une borne inférieure sur l'erreur quadratique moyenne, ou encore sur la racine carrée de cette erreur. (risque de sur apprentissage, ou l'erreur d'apprentissage atteint zéro, mais le réseau ne peut généraliser si on prend d'autres échantillons).

Les deux critères peuvent amener le réseau à un sur apprentissage, pour remédier à ce phénomène on peut utiliser la technique suivante :

3. Une technique appelée technique de validation croisée, cette dernière consiste à utiliser deux ensembles indépendants de données pour entraîner le réseau : un pour l'apprentissage (l'ajustement des poids) et l'autre pour la validation, c'est-à-dire le calcul d'un indice de performance, le critère d'arrêt consiste alors à stopper l'apprentissage lorsque l'indice de performance calculé sur les données de validation cesse de s'améliorer pendant plusieurs périodes d'entraînement.
4. le phénomène de saturation des neurones, qui consiste à ce que les neurones aboutissent à une convergence très lente. L'un des remèdes est de normaliser les données à l'entrée du réseau, ou encore de borner les valeurs initiaux des poids synaptiques.
5. Mise à jour des poids synaptiques après une période complète d'apprentissage.
6. L'une des méthodes d'amélioration de l'algorithme de rétro-propagation est le momentum, qui consiste à ajouter un terme à l'inertie dont le rôle est de filtrer les oscillations dans la trajectoire de la descente du gradient [56]
7. Une autre façon d'améliorer les performances de l'algorithme est de modifier le taux d'apprentissage dynamiquement tout au long de l'entraînement [57].

Ces considérations pratiques peuvent améliorer l'algorithme de rétro-propagation dans certaines situations, mais ça ne représente en aucun cas des règles à appliquer et s'attendre que ça donne toujours de bons résultats [59].

L'algorithme de rétro-propagation est très gros consommateur de temps de calcul sur des problèmes de grande taille. En outre, il n'existe pas de résultat liant la « complexité » du problème au temps d'apprentissage nécessaire.

3.4 Réseaux à fonction de base radiale (RBF)

L'algorithme de rétro-propagation décrit précédemment pour la conception d'un perceptron multicouche peut être considéré comme une forme d'approximation stochastique. Les réseaux à fonctions de base radiale (RBFs) adoptent une approche différente en regardant

la conception d'un réseau neuronal comme un problème d'ajustement de courbe en trouvant un meilleur ajustement aux données d'entraînement dans un espace multidimensionnel. L'utilisation de RBF dans la conception des réseaux de neurones a été introduite par Broomhead et Lowe en 1988 [58].

Le réseau RBF est formé essentiellement de trois couches complètement différentes ; une couche d'entrée, une couche cachée de dimension suffisamment élevée, et une couche de sortie. La transformation de l'unité cachée à l'espace de sortie est linéaire. Chaque nœud de sortie est la somme pondérée des sorties de la couche cachée. Toutefois, la transformation de la couche d'entrée vers la couche cachée est non linéaire.

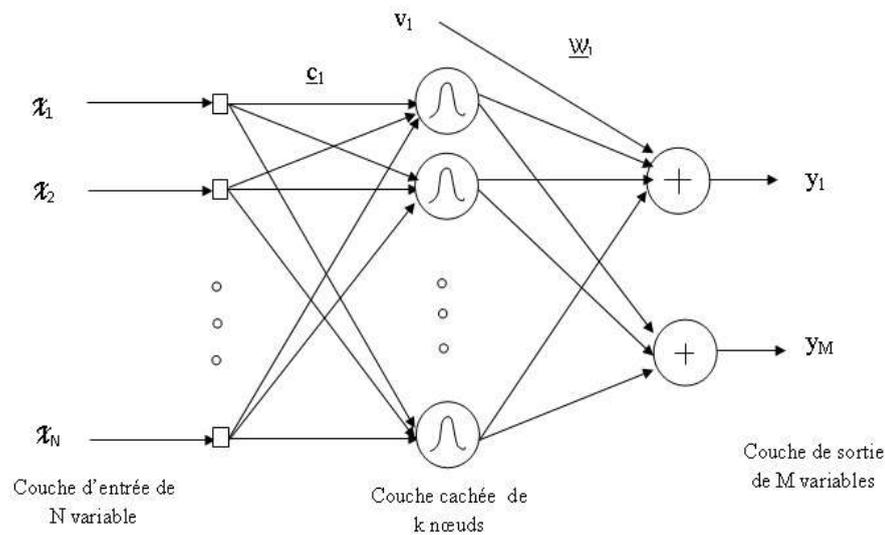


FIGURE 3.6: Réseau RBF avec un noyau gaussien.

Chaque neurone ou un nœud dans la couche cachée forme une combinaison linéaire des fonctions de base (ou noyau) qui produit une réponse localisée par rapport aux signaux d'entrée. Cela revient à dire que le RBF produit une réponse significative différente de zéro que lorsque l'entrée se situe dans une petite région localisée de l'espace d'entrée. La figure 3.6 illustre la structure de réseau de neurones RBF. La base la plus commune des RBFs est une fonction de noyau gaussien de la forme :

$$\varphi_l(\underline{x}) = \exp \left[-\frac{(\underline{x} - \underline{c}_l)^T (\underline{x} - \underline{c}_l)}{2\sigma_l^2} \right] \quad l = 1, 2, K, L \quad (3.12)$$

Où φ_l est la sortie du l^{eme} nœud dans la couche cachée, \underline{x} est l'échantillon d'entrée, \underline{c}_l est le vecteur des poids du l^{eme} nœud dans la couche cachée, c.a.d le centre de la gaussienne du nœud l ; σ_l^2 est le paramètre de normalisation (la mesure de propagation) du l^{eme} nœud, et L est le nombre de nœuds dans la couche cachée. Les sorties sont dans la plage de zéro à un de sorte que le plus proche de l'entrée est au centre de la gaussienne, plus la réponse du

nœud. Le nom RBF vient du fait que ces noyaux gaussiens sont radialement symétriques ; autrement dit, chaque nœud produit une sortie identique pour les entrées qui se trouvent à une distance radiale fixe du centre du noyau c_l .

La sortie du réseau est donnée par la relation suivante :

$$y_i = \underline{w}_i^T \varphi(\underline{x}) \quad i = 1, 2, \dots, K, \dots, M$$

Où y_i est la sortie du i_{eme} nœud, \underline{w}_i est le vecteur de poids de ce nœud, M est le nombre de nœud dans la couche de sortie, et $\varphi(\underline{x})$ est le vecteur des sorties de la couche cachée (auquel est ajouté un biais) [59].

3.5 Avantages et inconvénients des réseaux de neurones

3.5.1 Avantages des réseaux de neurones

- ◆ Capacité de représenter n'importe quelle fonction, linéaire ou pas, simple ou complexe.
- ◆ Faculté d'apprentissage à partir d'exemples représentatifs, par rétro propagation des erreurs. L'apprentissage (ou construction du modèle) est automatique.
- ◆ Résistance au bruit ou au manque de fiabilité des données.
- ◆ Comportement moins mauvais en cas de faible quantité de données.
- ◆ Pour l'utilisateur novice, l'idée d'apprentissage est plus simple à comprendre que les complexités des statistiques multivariées.

3.5.2 Inconvénients des réseaux de neurones

- ◆ L'absence de méthode systématique permettant de définir la meilleure topologie du réseau et le nombre de neurones à placer dans la (ou les) couche(s) cachée(s).
- ◆ Le choix des valeurs initiales des poids du réseau et le réglage du pas d'apprentissage, qui jouent un rôle important dans la vitesse de convergence.
- ◆ Le problème du sur-apprentissage (apprentissage au détriment de la généralisation).
- ◆ La connaissance acquise par un réseau de neurone est codée par les valeurs des poids synaptiques, les réseaux de neurones sont donc des boîtes noires où les connaissances sont inintelligibles pour l'utilisateur.

3.6 Systèmes neuro-flous

La combinaison des réseaux de neurones artificiels (RNAs) et les systèmes d'inférence floue (FIS) a attiré l'attention de plusieurs chercheurs dans différents domaines scientifiques et techniques en raison des besoins croissants des systèmes intelligents pour résoudre les problèmes complexes du monde réel. Les RNAs apprennent les entrées présentées à partir d'une base de données en mettant à jour les interconnexions entre les couches. Le FIS est le modèle le plus courant de calcul basée sur le concept de la théorie des ensembles flous, les règles flous si-alors, et le raisonnement flou.

La combinaison de Neurone et floue pour les applications d'engineering est un modèle populaire pour résoudre les problèmes complexe. Chaque fois qu'il y a une connaissance exprimée dans des règles linguistiques, un FIS peut être modélisée, et si l'information est disponible, ou si les paramètres peuvent être appris par apprentissage, alors un RNA peut être utilisé. Bien que pour la construction d'un FIS, les ensembles flous, opérateurs flous, et la base de connaissances doivent être spécifiés. Pour mettre en œuvre un ANN pour une application spécifique l'architecture et l'algorithme d'apprentissage sont nécessaires. Les inconvénients de ces approches apparaissent complémentaires et par conséquent, il est naturel de considérer la mise en œuvre d'un système intégré combinant les concepts neuro-flous.

L'intégration des RNAs et des FISs peuvent être classé en trois catégories à savoir les modèles concurrentes, les modèles coopératifs, et les modèles entièrement fusionnés.

3.6.1 Systèmes neuro-flous coopératifs et concurrents

Un modèle coopératif peut être considéré comme un préprocesseur dans lequel le mécanisme d'apprentissage du RNA détermine les fonctions d'appartenance de FIS ou règles floues à partir des données d'apprentissage. Une fois ces paramètres sont déterminés, le RNA va à l'arrière-plan. Les règles sont formées par une approche de clustering ou algorithmes de clustering floues.

Les fonctions d'appartenance floues sont approchées par réseau de neurones à partir des données d'entraînement. Dans un modèle concurrent, le RNA assiste le FIS en continu pour déterminer les paramètres nécessaires surtout si les variables d'entrée du contrôleur ne peuvent pas être mesurés directement. Dans certains cas, les sorties de FIS pourraient ne pas être directement applicable au processus. Dans ce cas le RNA peut agir comme un post-processeur de la sorties de FIS. Les figures 3.7 et 3.8 représentent les modèles NF coopératives et concurrent.

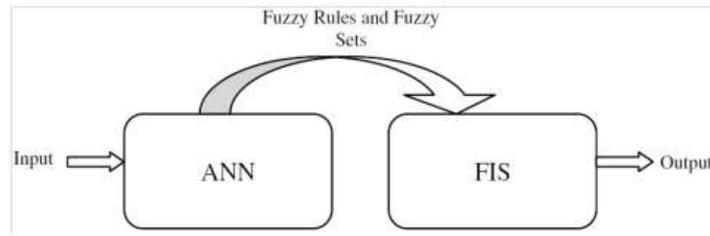


FIGURE 3.7: Modèle neuro-flou coopératif.

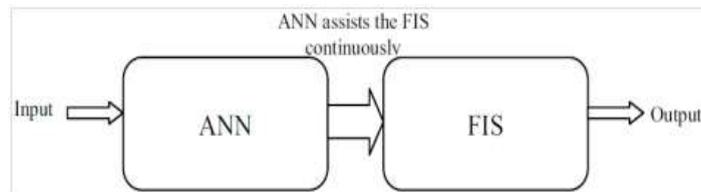


FIGURE 3.8: Modèle neuro-flou concurrent.

3.6.2 Systèmes neuro-flous fusionnés

Dans l'architecture neuro-flou fusionné (NF), les algorithmes d'apprentissage sont utilisés pour trouver les paramètres de FIS. Les systèmes NF fusionnés partagent des structures d'information et la représentation des connaissances. L'algorithme d'apprentissage peut être appliqué à un système flou en interprétant le système flou dans un RNA comme architecture.

Dans la dernière décennie, divers systèmes neuro-flous fusionnés ont été développés, plusieurs d'entre eux sont connus dans la littérature sous des abréviations comme ANFIS [60], DENFIS [61], FALCON [62], GARIC [63], NEFCLASS [64], NEFPROX [65], SANFIS [66], FLEXNFIS [67] et d'autres. Les systèmes neuro-flou sont souvent orientés vers la commande de système complexe et les problèmes de classification. La figure 3.9 montre quelques exemples d'architecture de système neuro-flou fusionnés, ces architectures sont souvent orienter vers des tâches spécifiques, et diffèrent entres elles soit dans la structure elle-même ou dans l'algorithme utilisé pour l'apprentissage [68].

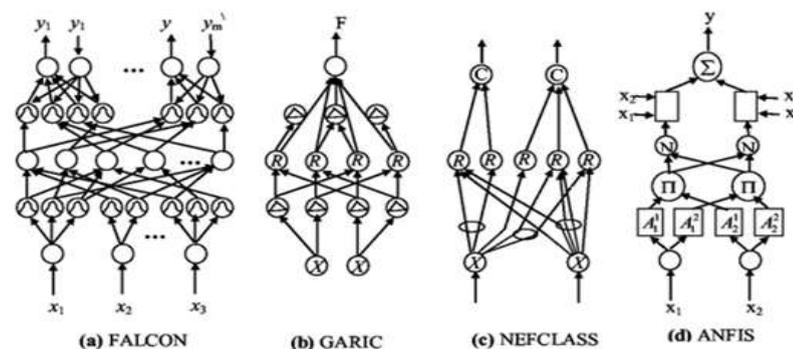


FIGURE 3.9: Architectures des systèmes neuro-flous.

3.7 Architecture d'ANFIS

Pour des raisons de simplicité, nous considérons le modèle flou de Sugeno d'ordre un ayant deux entrées x et y , une sortie f et deux règles :

Règle 1 : si x est A_1 ET y est B_1 ALORS $f_1 = p_1x + q_1y + r_1$

Règle 2 : si x est A_2 ET y est B_2 ALORS $f_2 = p_2x + q_2y + r_2$

Pour ce type de règles, on fait appel à un réseau de neurones qui admet pour entrées x et y et dont les deux couches cachées correspondent respectivement au calcul de la valeur des fonctions d'appartenance $A_i, B_i, i = 1, 2$ et à celui de la valeur prise par la conjonction des conditions de chaque règle utilisant un opérateur adéquat. Les fonctions d'appartenance intervenant dans les règles sont considérées comme des paramètres ajustés par l'intermédiaire des poids entrant dans la première couche cachée, les conclusions f_i des règles sont également des paramètres ajustables par l'intermédiaire des poids à la dernière couche. La figure (3.10) illustre le mécanisme de raisonnement du modèle flou de Sugeno.

L'architecture ANFIS équivalente est donnée par la figure (3.11), où les nœuds d'une même couche ont des fonctions similaires, comme décrit ci-dessous.

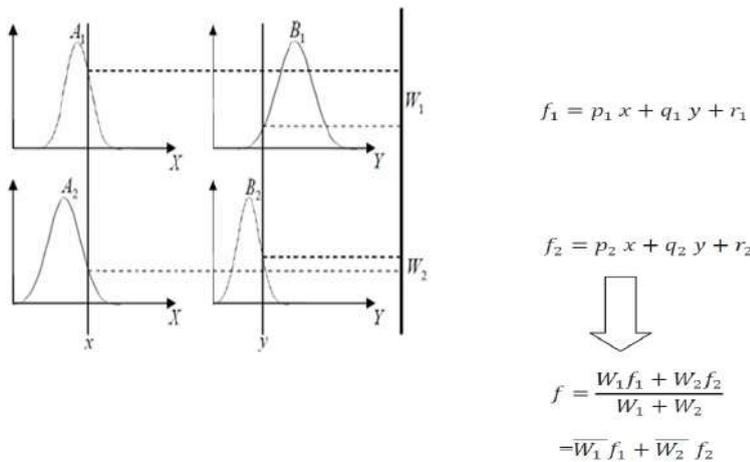


FIGURE 3.10: Modèle flou de Sugeno d'ordre un à deux entrées.

Couche1 :

Dénotons le nœud de sortie i dans la couche l par $O_{l,i}$. Chaque nœud de cette couche est un nœud adaptatif avec un nœud de sortie défini par :

$$O_{1,i} = \mu_{A_i}(x), \quad i = 1, 2 \quad \text{Ou}$$

$$O_{1,i} = \mu_{B_{i-2}}(y), \quad i = 3, 4 \quad (3.13)$$

Où x (ou y) est l'entrée du nœud et A_i (ou B_{i-2}) est l'ensemble flou associé à ce nœud. Autrement dit, les sorties de cette couche sont les degrés d'appartenance des prémisses (x

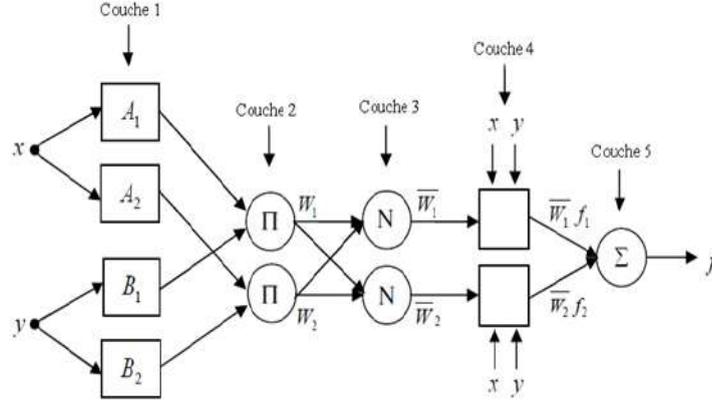


FIGURE 3.11: Architecture d'ANFIS équivalent au modèle flou de Sugeno d'ordre un à deux entrées et deux règles.

et y). Les fonctions d'appartenances choisies pour A_i et B_{i-2} sont de type gaussien :

$$\mu_{A_i}(x) = \exp \left[-\frac{1}{2} \left(\frac{x - c_i}{b_i} \right)^2 \right] \quad (3.14)$$

Où c_i désigne le centre de la fonction et b_i sa variance.

Couche 2 :

$$O_{2,i} = W_i = \mu_{A_i}(x) \times \mu_{B_i}(y), \quad i = 1, 2 \quad (3.15)$$

Couche 3 :

Tout nœud de cette couche est un nœud fixe étiqueté N. Le i_{eme} nœud calcule ce rapport :

$$O_{3,i} = \bar{W}_i = \frac{W_i}{W_1 + W_2}, \quad i = 1, 2 \quad (3.16)$$

Couche 4 :

Chaque nœud i de cette couche est un nœud adaptatif, sa fonction est :

$$O_{4,i} = \bar{W}_i f_i = \bar{W}_i (p_i x + q_i y + r_i) \quad (3.17)$$

Où \bar{W}_i est la sortie de la couche 3 et p_i, q_i, r_i est l'ensemble des paramètres. Les paramètres de cette couche sont les paramètres de la conséquence des règles.

Couche 5 :

Le seul nœud dans cette couche est un nœud fixe étiqueté Σ , qui calcule la sortie globale comme sommation de tous les signaux qui arrivent :

$$O_{5,i} = \text{sortie globale} = \sum_i \bar{W}_i f_i = \frac{\sum_i W_i f_i}{\sum_i W_i} \quad (3.18)$$

Donc, nous avons construit un réseau adaptatif qui a exactement la même fonction qu'un

modèle flou de Sugeno. Notons que la structure de ce réseau adaptatif n'est pas unique, nous pouvons facilement combiner les couches 3 et 4 pour obtenir un réseau équivalent avec seulement 4 couches. Nous pouvons jouer sur la normalisation du poids dans la dernière couche ; la figure (3.12) illustre un ANFIS de ce type.

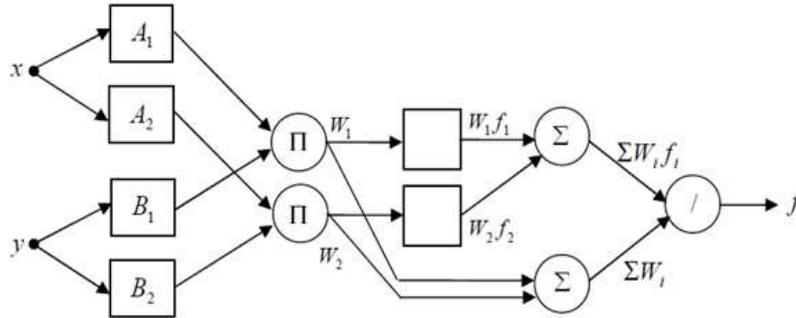


FIGURE 3.12: Autre architecture d'ANFIS pour le modèle flou de Sugeno à deux entrées et deux règles.

Apprentissage de contrôleur neuro-flou

Une fois que le régulateur neuro-floue est connecté, le but de l'apprentissage sera d'optimiser les valeurs des paramètres des fonctions d'appartenance et qui sont les paramètres de la fonction gaussienne ainsi que les poids qui sont les valeurs des conclusions. L'apprentissage est assuré par la méthode de retro propagation ou la méthode hybride qui combine entre l'algorithme de retro propagation et la méthode de moindre carrée.

3.8 Résultats de simulation

Dans cette partie nous présentons les résultats de simulation de la commande neuro-floue 'ANFIS' appliquée au robot Motoman réduit a trois premiers degrés de libertés, pour cette simulation nous avons utilisé les toolbox du logiciel Matlab/Simulink, fuzzy logic toolbox et robotic toolbox [89]. Pour l'apprentissage du contrôleur neuro-flou nous avons utilisé une base de donnée entrée/sortie d'un contrôleur PID, avec des trajectoires de référence sigmoïdes dont les paramètres sont : $q_0 = [0, 0, 0]$, $q_f = [\pi/3, \pi, -\pi]$. Après avoir réalisé l'apprentissage du contrôleur, nous l'avons appliqué au robot Motoman, les graphes de poursuite de trajectoire, de l'erreur de poursuite et de la commande pour les trois axes sont présentés.

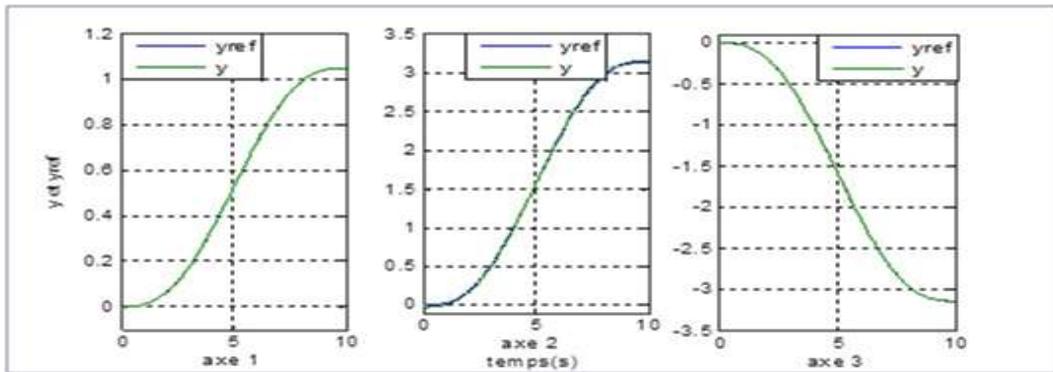


FIGURE 3.13: Poursuite de trajectoire.

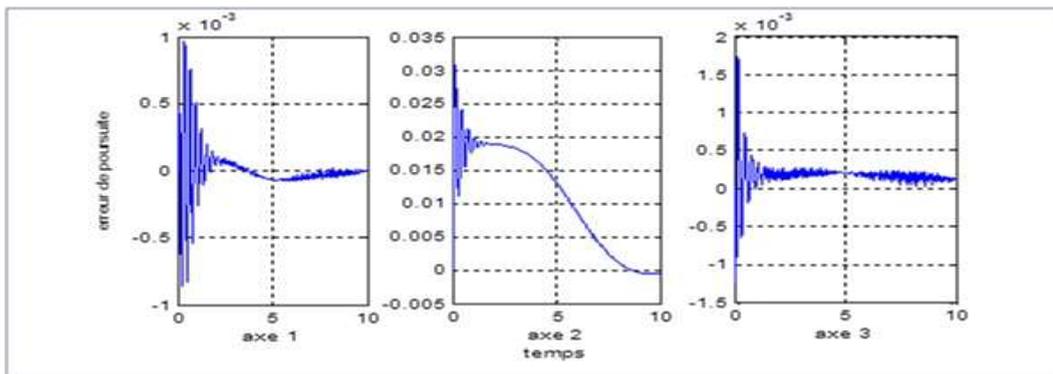


FIGURE 3.14: Erreur de poursuite.

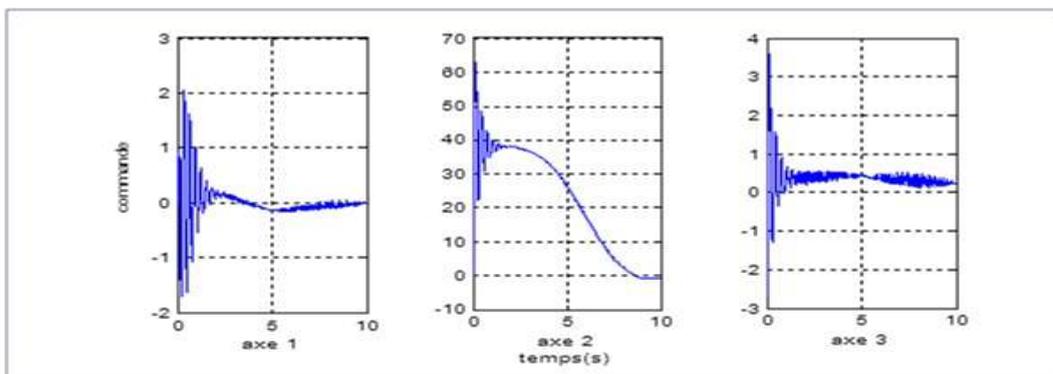


FIGURE 3.15: Signaux de commande.

Les résultats obtenus montrent une bonne poursuite de trajectoire avec une erreur très faible, ce qui indique que le contrôleur neuro-flou peut reprendre la tâche réalisé par le PID. Dans la deuxième partie de cette simulation et pour testé la robustesse et la possibilité de généralisation de la commande nous avons choisi des trajectoires sinusoïdales pour les trois axes, dont les paramètres sont : Amplitude = $[1.5, 1.8, 2]$, Fréquence = $[2, 1.5, 0.5]$ rad/sec.

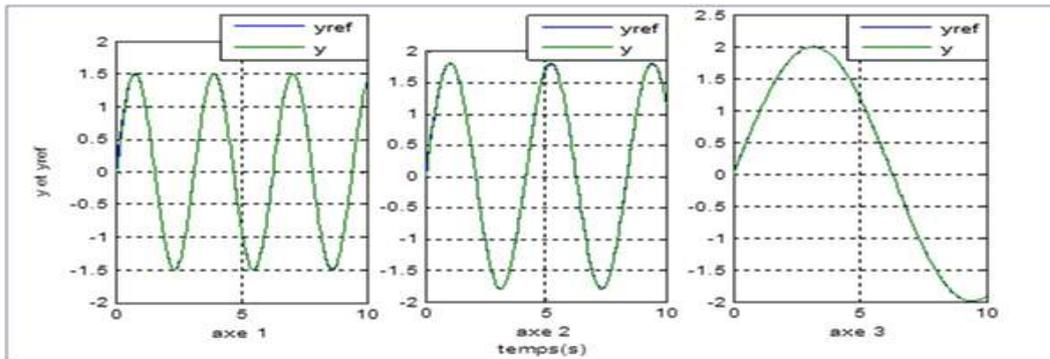


FIGURE 3.16: Poursuite de trajectoire.

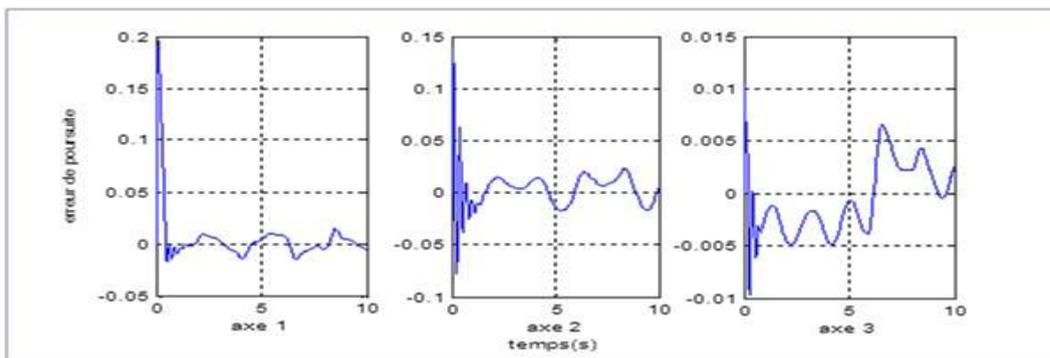


FIGURE 3.17: Erreur de poursuite.

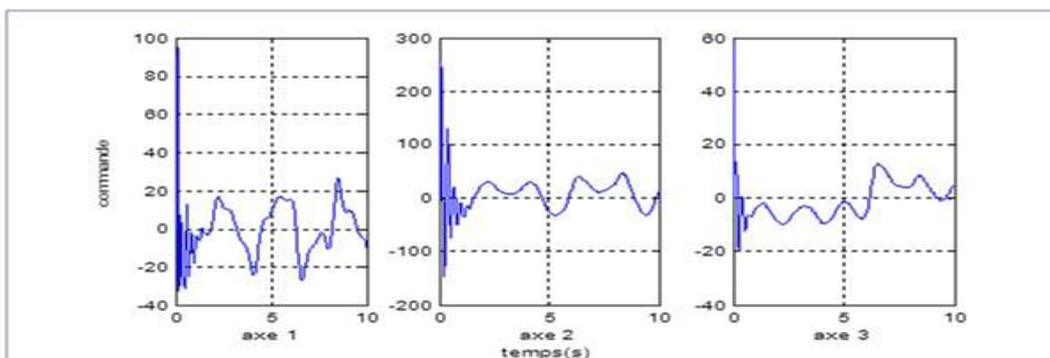


FIGURE 3.18: Signaux commande.

Les résultats obtenus avec des trajectoires de référence sinusoïdales sur les trois axes montrent une poursuite avec une erreur minimale, on peut voir que le contrôleur applique une commande intéressante au démarrage du système ensuite une fois qu'il atteint la position désirée la commande se réduit à des petites valeurs.

3.9 Conclusion

Il a été prouvé théoriquement que la fonction des réseaux de neurones formels sont des approximateurs universels. Dans ces derniers les paramètres les plus importants sont les poids synaptiques. Ce sont eux qui construisent le modèle dit neurone de résolution en fonction des informations fournis au réseau. Le mécanisme d'apprentissage permet de les calculer à partir des grandeurs que l'on peut acquérir du problème. Dans un modèle de réseaux de neurones formels, apprendre, c'est calculer les valeurs des coefficients synaptiques en fonction des exemples disponibles.

Le perceptron a longtemps été le seul réseau de neurone à être utilisé à cause de sa simplicité. Mais avec l'augmentation de la puissance des calculateurs, on commence à rencontrer les RBF et des réseaux plus complexes qui ne sont pas à propagation directe.

Les méthodes à base de logique floue et de réseaux de neurones se retrouvent de plus en plus souvent dans les algorithmes dits de "contrôle intelligent". Les réseaux neuro-flous sont des systèmes qui combinent les capacités d'apprentissage et de généralisation des réseaux de neurones, avec la formalisation de règles imprécises ou entachées d'incertitudes correspondant à une information humaine approximative ou à une information partielle ou partiellement erronée, comme c'est souvent le cas dans les applications industrielles. L'ANFIS (système d'inférence floue paramétré par apprentissage neuronal) présente l'un des systèmes neuro-flou à cinq couches, est bien adapté à la commande des systèmes non linéaires, à été choisis dans notre travail pour une application en simulation sur le robot Motoman SV3 à trois degrés de liberté. Les résultats obtenus ont montré l'efficacité de ce contrôleur, néanmoins l'utilisation de ce dernier en off line, après une phase d'apprentissage est un inconvénient dans le mécanisme de la commande des systèmes, or, on cherche à développer des lois de commande qui s'applique en on line et qui s'adapte aux variations paramétrique du système.

Après avoir présenté dans ce chapitre les principes de base des réseaux de neurones qui sont inspirés de l'étude du cerveau humain, l'algorithme de rétro-propagation, et dans une deuxième partie le réseau neuro-flou, et le réseau ANFIS, et dans le but de développer une commande neuro-flou adaptative, les concepts de la commande adaptative et de la commande par mode glissant sont présentés dans le chapitre suivant.

Chapitre 4

Commande par mode glissant et adaptative

Les techniques de contrôle intelligent tels que logique floue, les réseaux de neurone ont présenté des résultats prometteurs dans divers domaines. En particulier, les techniques flous se sont retrouvées avec succès dans le contrôle des systèmes sous actionnés. Mais très peu de travaux de recherche ont été élaborés sur le réglage des paramètres du contrôleur flou qui peut en générale améliorer les performances du contrôleur. En outre, l'intégration des réseaux de neurones avec les régulateurs flous a prouvé être une combinaison puissante qui élimine les inconvénients des deux concepts aboutissant aux systèmes neuro-flous.

L'implémentation des systèmes NFs pour la commande des systèmes en ligne nécessite une technique de réglage des paramètres du contrôleur en ligne, l'intégration de la commande adaptative avec les systèmes NFs permet de réaliser cette tâche, en plus de la commande par mode glissant offre la possibilité de prouver la stabilité en boucle fermée.

Dans cette partie nous présentons quelques concepts de la commande adaptative ainsi que le principe de base la commande par mode glissant.

4.1 Commande Adaptative

Un système réel dans un milieu industriel est souvent soumis à des conditions environnementales qui influent fortement sur sa dynamique, cet effet rend les paramètres du système variable, ce qui complique la régulation de ces derniers avec des contrôleurs à paramètres fixes.

Dans le langage courant, "adapter" désigne une modification du comportement afin qu'il soit conforme aux nouvelles conditions, intuitivement, un contrôleur adaptatif est donc un dispositif de commande qui peut modifier son comportement en réponse à des changements dans

la dynamique du processus et le caractère de perturbations [69]. En ce sens, un contrôleur adaptatif constitue un moyen de gérer l'incertitude paramétrique.

Donc on peut dire qu'un contrôleur adaptatif est un contrôleur avec des paramètres ajustables et un mécanisme pour l'ajustement de ces derniers. Le contrôleur devient non linéaire en raison du mécanisme d'ajustement des paramètres, il a, cependant, une structure très particulière.

La théorie du contrôle adaptative porte essentiellement sur la recherche des algorithmes d'ajustement des paramètres qui garantissent la stabilité globale et la convergence.

Dans le cas où les paramètres du modèle dynamique du système à contrôler sont inconnus, mais constants (au moins dans une certaine région de fonctionnement). Dans de tels cas, le réglage correct des paramètres de régulation ne peut se faire sans la connaissance de leurs valeurs. Des techniques de contrôle adaptatif peuvent aussi fournir un procédé de réglage automatique en boucle fermée pour les paramètres du régulateur. Dans ce cas, l'effet de l'adaptation disparaît dans le temps. Les changements dans les conditions de fonctionnement peuvent nécessiter un redémarrage de la procédure d'adaptation [70].

Un système de commande adaptative est généralement d'une structure ayant deux boucles. Une boucle de rétroaction normale avec le processus et le contrôleur. L'autre boucle est une boucle de réglage du paramètre. Cette structure est représentée sur la figure 4.1. La boucle d'ajustement des paramètres est souvent plus lente que la boucle de rétroaction normale [69].

Un contrôleur adaptatif est formé en combinant un estimateur de paramètre en ligne, qui fournit des estimations des paramètres inconnus à chaque instant, avec une loi de commande qui est basée sur les paramètres connus. La façon dont l'estimateur de paramètre combiné avec la loi de commande donne lieu à deux approches différentes. Dans une première approche, appelée commande adaptative indirecte, les paramètres de système sont estimés en ligne et utilisés pour calculer les paramètres du régulateur. Cette approche a également été appelée la commande adaptative explicite, parce que la conception est basée sur un modèle de processus explicite.

Dans la seconde approche, appelée commande adaptative directe, les paramètres de régulation sont estimés directement, sans calculs intermédiaires concernant les paramètres estimés du système. Cette approche a été désignée commande adaptative implicite parce que la conception est basée sur l'estimation d'un modèle du système implicite [71].

Il existe essentiellement trois approches de commande adaptative :

4.1.1 Structure à gain programmé

Dans plusieurs cas il est possible de trouver des variables mesurables qui corréleront bien avec les changements de la dynamique du processus. Ces variables peuvent être utilisées pour

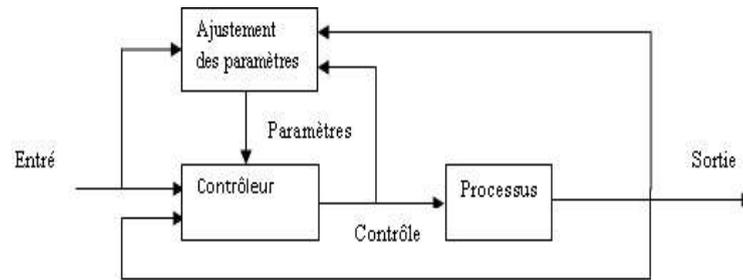


FIGURE 4.1: Structure d'une commande adaptative.

changer les paramètres du contrôleur. Cette approche est appelée gain programmé, puisque le schéma est originalement utilisé pour mesurer le gain et ces variations. Cette approche a l'avantage d'ajuster rapidement les paramètres du correcteur lors de changements rapides de la dynamique du processus.

Un des inconvénients de la programmation de gain est que le mécanisme de réglage des gains de contrôleur est pré calculée (hors ligne). Des changements imprévisibles dans la dynamique du système peuvent conduire à une détérioration de la performance ou même un échec complet. Un autre inconvénient possible de la programmation de gain est les coûts élevés de conception et d'implémentation qui augmentent avec le nombre de points de fonctionnement.

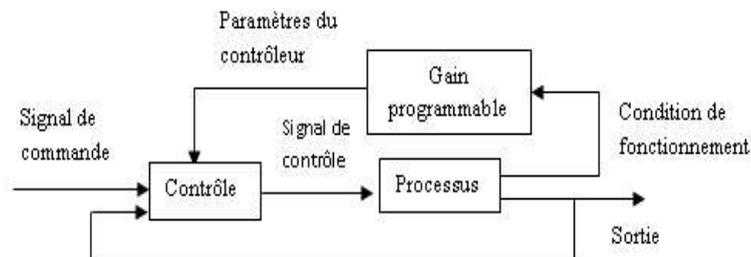


FIGURE 4.2: Structure de commande adaptative à gain programmé.

4.1.2 Commande adaptatives à modèle de référence (MRAC)

Le système adaptatif à modèle de références avait été originalement proposée pour résoudre un problème dans lequel les spécifications des performances sont données sous forme d'un modèle de référence. Ce modèle indique comment la sortie du processus doit répondre idéalement au signal de commande. La structure de cette commande est donnée dans la figure 4.3. On peut voir dans cette structure que la boucle externe ajuste les paramètres du contrôleur de telle sorte que l'erreur, qui est la différence entre la sortie du processus et celle du modèle, soit minimale.

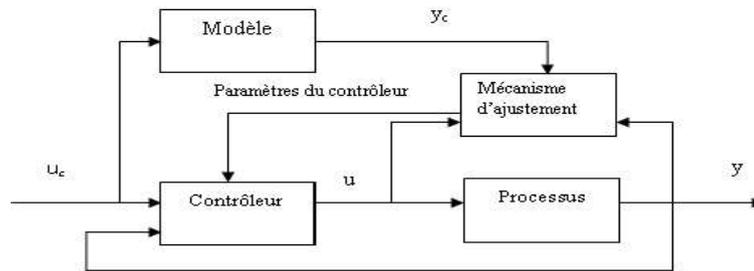


FIGURE 4.3: Commande adaptative à modèle de référence.

4.1.3 Commande adaptative auto-ajustable (STR : Self Tuning Regulator)

Les schémas vus précédemment sont appelés, méthode directes, parce que le mécanisme d'ajustement indique directement comment les paramètres du contrôleur doivent être adaptés. Un schéma différent est obtenu si les paramètres estimés du processus sont ajustés et les paramètres du contrôleur sont obtenus en utilisant les paramètres estimés. Le schéma de cette structure est donné dans la figure 4.4.

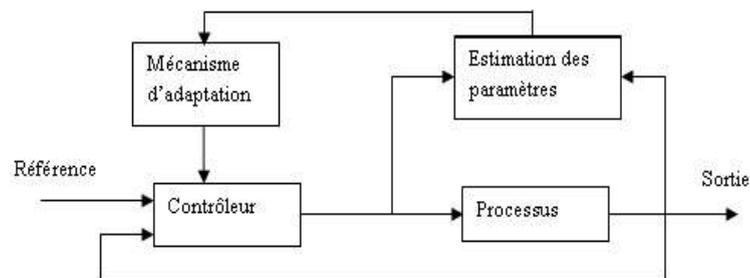


FIGURE 4.4: Commande adaptative auto-ajustable.

Le diagramme représenté dans la figure 4.5 est une méthodologie pour le choix de la structure de commande adaptative selon les caractéristiques et la dynamique du système.

4.2 Commande par mode glissant

La théorie des systèmes à structure variables (SSV) avait son origine dans l'ex-Union soviétique, la caractéristique principale des systèmes à structure variables, comme leur nom l'indique, est que leur structure peut être modifiée intentionnellement en cours de fonctionnement, selon un algorithme ou une loi de sélection à des instants qui dépend de la valeur du signal d'erreur et de ces dérivées.

La méthodologie de commande par mode de glissement est devenue le mode de fonctionnement principal des systèmes de contrôle basé sur les SSV [72]. La majorité des méthodes de conception des SSV sont basées sur l'introduction délibérée des modes glissant qui ont joué un rôle remarquable non seulement dans les développements théoriques, mais aussi dans

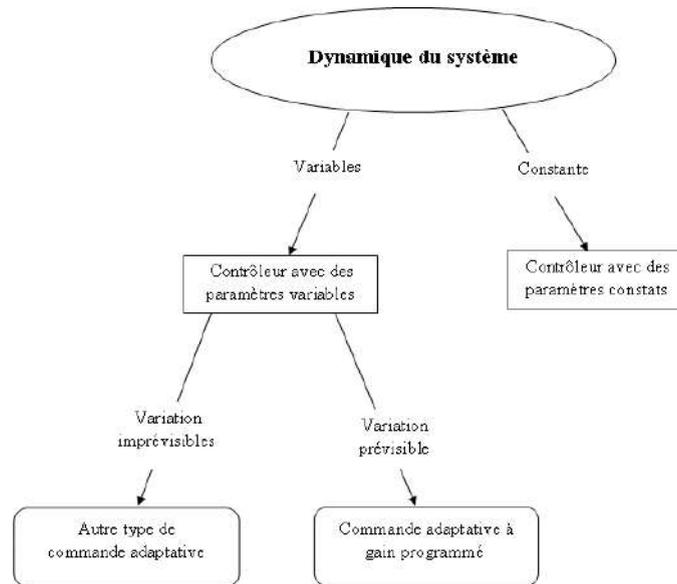


FIGURE 4.5: Choix de la structure de la commande adaptative.

les applications pratiques [73]. La théorie de la commande par mode de glissement couvre les systèmes à entrées et sorties simples ou multiples, les systèmes mono-variables et multi-variables, les systèmes linéaires et non linéaires, ainsi que les systèmes d'ordre élevé.

La commande par mode glissant a connu un essor considérable durant les dernières décennies. Ceci est dû principalement à la propriété de convergence rapide et en temps fini des erreurs, ainsi, que la grande robustesse par rapport aux erreurs de modélisation et certains types de perturbations extérieures [74].

La commande par mode glissant est caractérisée par sa simplicité d'élaboration et par sa performance de contrôle acceptable dans une large gamme de systèmes et elle résoud deux sujets difficiles, les plus importants dans le contrôle qui sont, la stabilité et la robustesse [75].

Les oscillations de haute fréquence liées à la commutation de la commande au voisinage de la surface de glissement « phénomène de chattering » présente l'inconvénient principal de la commande par mode glissant, ce dernier peut exciter des dynamiques non modélisables conduisant à des situations d'instabilité.

Plusieurs solutions ont été proposées dans la littérature pour remédier à cette inconvénient [69, 76].

Au cours des dernières années, les concepts de l'intelligence artificielle sont utilisés dans les systèmes de commande par mode glissant. Réseau neuronal, logique floue, et neuro-flou sont combinés de manière synergique avec le contrôleur par mode de glissement et appliqués aux systèmes non linéaires variant dans le temps, et incertains [74].

4.2.1 Structures de base

Dans les systèmes à structure variable utilisant la commande par mode glissant, on peut distinguer trois configurations de base pour la synthèse des différentes commandes [77].

Structure par commutation d'une contre réaction d'état

Le schéma d'une telle structure est donné dans la figure 4.6, la loi de commutation permet de mettre le vecteur d'état \underline{x} en contre réaction d'état soit par le gain k_1 ou soit par le gain k_2 selon l'état.

$$u = \begin{cases} -k_1^T(x) & \text{si } s(x) > 0 \\ -k_2^T(x) & \text{si } s(x) < 0. \end{cases} \quad (4.1)$$

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

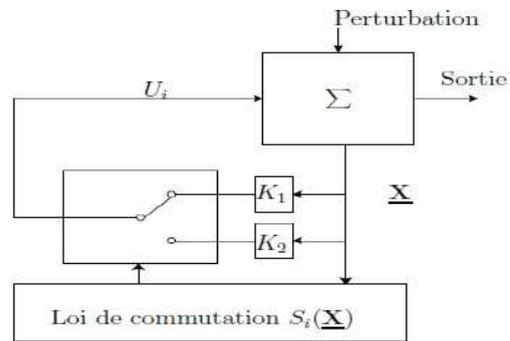


FIGURE 4.6: Régulation par commutation au niveau de la contre réaction d'état.

Structure par Commutation au Niveau de L'organe de Commande

Le schéma d'une structure par commutation au niveau de l'organe de commande est donné sur la figure 4.7. Cette structure de commande est la plus classique et la plus utilisée. Dans

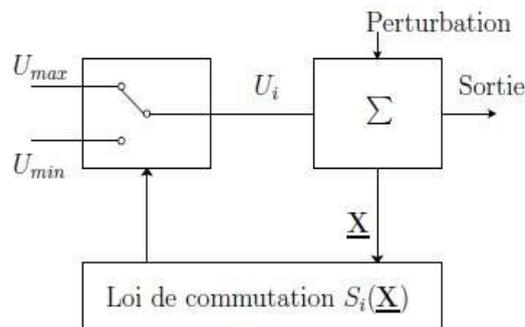


FIGURE 4.7: Régulation commutation au niveau de l'organe de commande.

cette configuration la loi de commutation est donnée par :

$$u = \begin{cases} u_{max} & \text{si } s(x) > 0 \\ u_{min} & \text{si } s(x) < 0. \end{cases} \quad (4.2)$$

En mode de glissement (régime glissant), la dynamique du système est donnée par $s(x)=0$. Cette structure correspond au fonctionnement tout ou rien, donc elle s'adapte au système dont l'organe de commande est un interrupteur.

Structure par Commutation au niveau de l'organe de commande, avec ajout de la commande équivalente

Cette structure permet de pré-positionner le système dans un état désiré stable et de réduire le phénomène de chattering, le terme de commutation assure la convergence vers cet état et l'y maintenir.

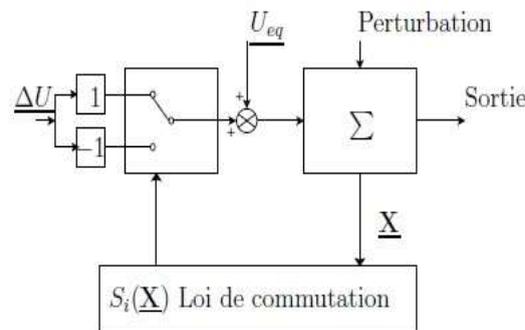


FIGURE 4.8: Structure de régulation par ajout de la commande équivalente.

4.2.2 Principe du réglage par mode glissant

Le principe de la commande par modes glissants est de contraindre l'état du système à atteindre en temps fini une hypersurface (dans l'espace d'état) donnée pour ensuite y rester. Cette hypersurface étant une relation entre les variables d'état du système, elle définit une équation différentielle, et donc détermine totalement la dynamique du système. L'évolution d'un système soumis à une loi de commande qui le fait maintenir sur une hypersurface donnée ne dépend donc plus du système lui-même ou des perturbations auxquelles il peut être soumis, mais uniquement des propriétés de cette dernière. Le système bouclé n'est donc pas seulement robuste vis à vis des incertitudes (propres au système) et perturbations (extérieures au système), mais totalement insensible à ces incertitudes et perturbations, moyennant qu'elles puissent effectivement être rejetées par la commande [75]. Cette commande se fait en deux

étapes : la convergence vers la surface et ensuite le glissement le long de celle-ci (Figure 4.9)
 La conception de cette commande peut se faire en trois étapes :

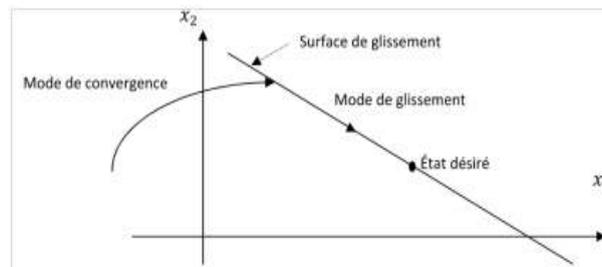


FIGURE 4.9: Différents modes de convergence pour la trajectoire d'état.

1. le choix des surfaces.
2. l'établissement des conditions d'existence et de convergence.
3. la détermination de la loi de commande [77].

Choix des surfaces

La surface de glissement est une fonction scalaire telle que la variable à régler glisse sur cette surface et tend vers l'origine du plan de phase. Cette surface est déterminée indépendamment de la commande. Sa forme non linéaire est une fonction de l'erreur sur la variable à régler x [78] :

Soit le système décrit par l'équation différentielle suivante :

$$\dot{x}^n = f(x, t) + g(x, t).u \quad (4.3)$$

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

L'erreur de poursuite e est définie par :

$$e = x_d - x \text{ Avec } x_d \text{ la consigne désirée.}$$

La formule générale de la surface de glissement est définie par :

$$s(x) = \left(\frac{\partial}{\partial t} + \lambda \right)^{n-1} e(x) \quad (4.4)$$

Où n représente le degré relatif du système par rapport à la sortie $y(t)$. Il indique le nombre minimum de fois qu'il faut dériver à la sortie $y(t)$ par rapport au temps, pour y voir apparaître l'entrée.

λ est une constante strictement positive.

$$\text{Pour } n = 2 : \quad s(x) = \dot{e} + \lambda.e(x)$$

$$\text{Pour } n = 3 : \quad s(x) = \ddot{e} + 2.\lambda.\dot{e}(x) + \lambda^2.e(x)$$

Compte tenu de l'état initial $x_d(0) = x(0)$, le problème de suivi $x \equiv x_d$ est équivalent à celui du maintien sur la surface $s(t)$ pour tout $t > 0$; en effet $s = 0$ représente une équation différentielle linéaire dont l'unique solution est $x = 0$, compte tenu des conditions initiales, le problème du suivi de vecteur à n dimensions x_d peut être réduit à celle du maintien de la valeur scalaire s à zéro [76].

Conditions d'existence et de convergence

La commande u de nature discontinue, va contraindre les trajectoires du système à atteindre la surface de glissement et d'y rester au voisinage de celle-ci malgré la présence de perturbation. En d'autres termes, la commande doit rendre cette surface de glissement localement attractive. un régime glissant existe à chaque fois que :

$$s \cdot \dot{s} < 0 \quad (4.5)$$

Pour une convergence en temps fini, la condition (4.5) qui ne garantit qu'une convergence asymptotique vers la surface de glissement est remplacée par une condition plus restrictive dite d'attractivité et donnée par [76] :

$$s \cdot \dot{s} < \eta \cdot |s| \quad , \eta > 0 \quad (4.6)$$

Cela indique que la "distance" au carré de la surface, tel que mesuré par s^2 , diminue le long de toutes les trajectoires du système. Ainsi, on contraint les trajectoires pour pointer vers la surface $S(t)$, comme illustré sur la figure 4.10. En particulier, une fois sur la surface, les trajectoires du système restent sur cette surface.

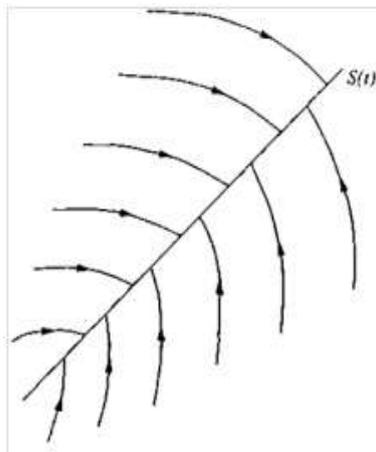


FIGURE 4.10: Condition de glissement.

Détermination de la loi de commande

La structure d'un contrôleur par mode glissant est constituée de deux parties, une concernant la linéarisation exacte u_{eq} et l'autre stabilisante u_c . u_{eq} sert à maintenir la variable à contrôler sur la surface de glissement $s(x, t) = 0$. Elle est déduite lorsque les trajectoires du système évoluent exactement sur la surface : $s(x, t) = 0$.

Pour le système

$$\ddot{x} = f + g.u \quad (4.7)$$

Nous avons

$$u_{eq} = -f + \ddot{x}_d - \lambda.\dot{e} \quad (4.8)$$

Donc la dynamique du système en mode de glissement est :

$$\ddot{x} = f + u_{eq} = \ddot{x}_d - \lambda.\dot{e} \quad (4.9)$$

Géométriquement, la commande équivalente peut être formulée comme suit :

$$u_{eq} = \alpha.u_+ + (1 - \alpha).u_- \quad (4.10)$$

À savoir, en tant que combinaison convexe des valeurs de u sur les deux côtés de la surface $S(t)$. La valeur de α peut être obtenue formellement à partir de $\dot{s} = 0$, ce qui correspond à exiger que les trajectoires du système soient tangente à la surface. Cette construction intuitive est résumée dans la figure 4.11, où $f_+ = [\dot{x}, f + u_+]^T$, $f_- = [\dot{x}, f + u_-]^T$ et $f_{eq} = [\dot{x}, f + u_{eq}]^T$.

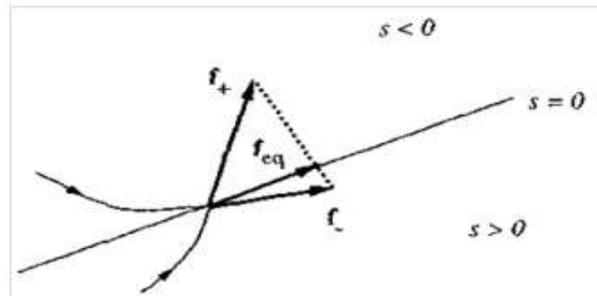


FIGURE 4.11: Construction de la dynamique équivalente d'après Filipov.

Cependant, il faut aussi contraindre le système à suivre la trajectoire imposée, c'est-à-dire obliger que la surface $s(x, t) = 0$ soit attractive. C'est le rôle de la commande discontinue qui est définie comme suit :

$$u_c = -k.sgn(s(x, t)) \quad (4.11)$$

Où k est une constante positive et sgn désigne la fonction signe.

4.2.3 Phénomène de Chattering

Le phénomène de chattering appelé aussi réticence ou broutement, caractérisé par de fortes oscillations autour de la surface de glissement est dû aux discontinuités appliquées à la commande durant le régime glissant, constitue un inconvénient non négligeable pour la commande par mode glissant d'ordre un. Les principales raisons à l'origine de ce phénomène sont les limitations des actionneurs ou les retards de commutation au niveau de la commande. Ces commutations détériorent la précision de la commande et peuvent s'avérer néfastes pour l'organe de commande en provoquant une détérioration prématurée des systèmes mécaniques. Afin de réduire ou d'éliminer le phénomène de chattering, de nombreuses techniques ont été

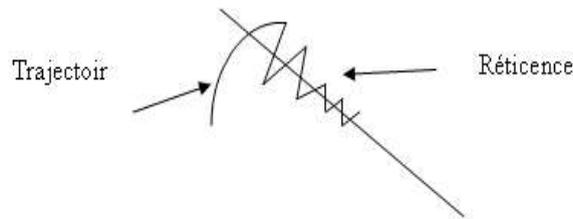


FIGURE 4.12: Phénomène de Chattering.

proposées dans la littérature, les plus utilisées sont les suivant :

La couche limite (Boundary Layer)

Elle consiste à substituer la fonction signe par une approximation à gain élevé uniquement dans un voisinage de la surface S . Parmi les fonctions utilisées nous citerons la fonction de saturation et la fonction signe plus saturation. Il existe d'autres fonctions qui sont moins utilisées car elles sont coûteuses en temps comme la loi d'interpolation de puissance, la fonction pseudo-signe, la fonction arc tangente et la fonction tangente hyperbolique. Avec cette solution le système ne converge plus vers la valeur désirée, mais vers un voisinage de cette dernière, dans ce cas le système est dit en régime pseudo-glissant [75]. Bien que cela permette d'atténuer le phénomène de réticence, la précision par rapport à l'objectif fixé, la robustesse de la commande et le temps de réponse s'en trouvent dépréciés.

En utilisant le même principe de la couche limite Lin et Chen en introduit une bande de transition flou non linéaire [79], cette méthode a permis d'éliminer le phénomène de chattering cependant la commande permettant la phase d'approche reste difficile à calculer.

Solution par mode glissant d'ordre supérieur

Les modes glissants d'ordre supérieur représentent une extension des modes glissants d'ordre 1 à un degré plus élevé, ils permettent de réduire l'effet de chattering au voisinage de

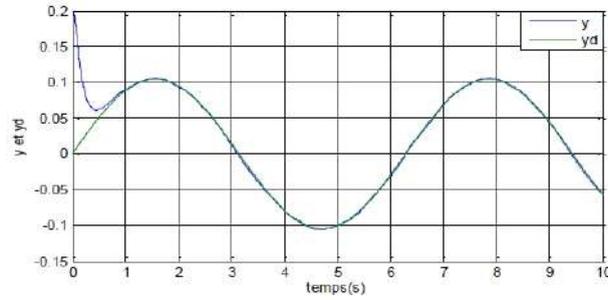


FIGURE 4.14: Courbe de référence et de sortie du système.

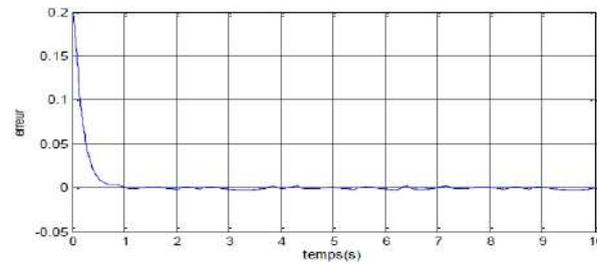


FIGURE 4.15: Ecart entre la sortie et la référence.

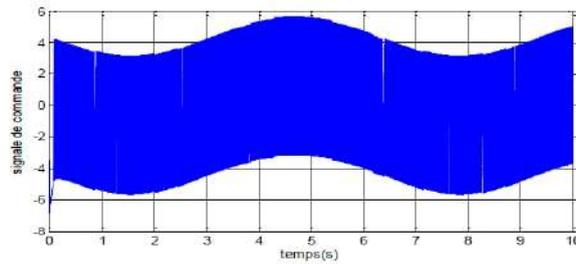


FIGURE 4.16: Signal de commande.

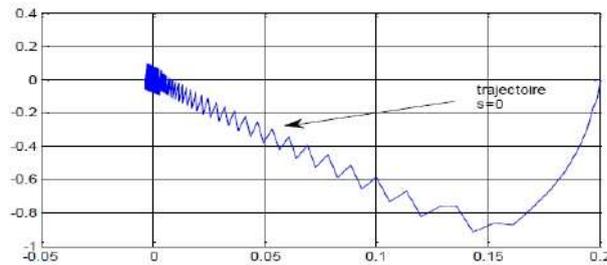


FIGURE 4.17: Réponse dans le plan de phase.

Dans ce cas on remarque une bonne poursuite de trajectoire, néanmoins la commande est trop excitée, ainsi que des oscillations au niveau de la trajectoire indiquant le phénomène de broutement.

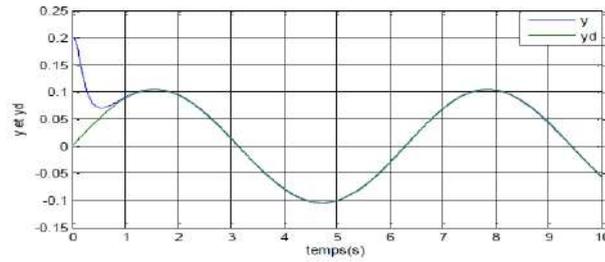


FIGURE 4.18: Courbe de référence et de sortie du système.

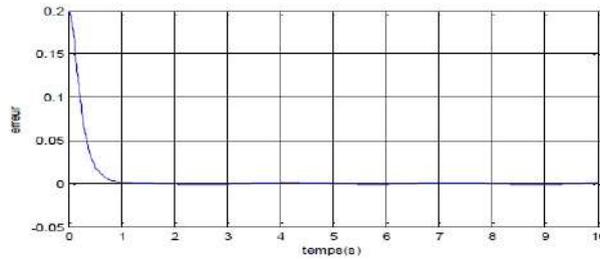


FIGURE 4.19: Ecart entre la sortie et la référence.

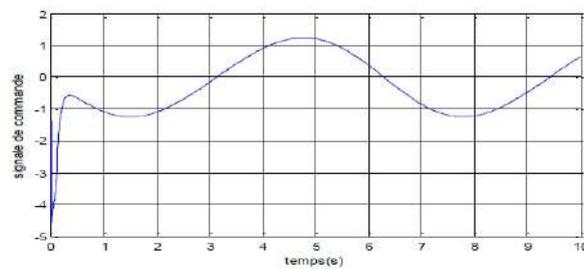


FIGURE 4.20: Signal de commande.

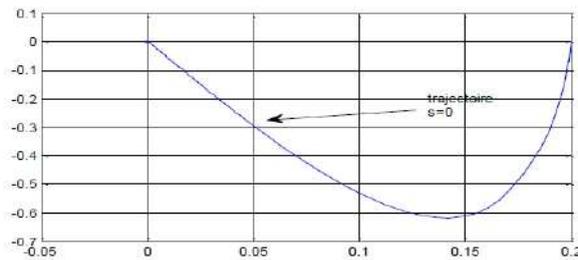


FIGURE 4.21: Réponse dans le plan de phase.

Une meilleure poursuite de trajectoire a été réalisée avec une fonction de saturation, de plus il est nettement remarqué qu'une diminution des oscillations au niveau de la trajectoire, donc une réduction apparente du phénomène de broutement.

4.4 Conclusion

La commande des systèmes dynamique à paramètres variant dans le temps est décrite dans ce chapitre au nom de commande adaptative, dont les paramètres s'adaptent au changement de la dynamique du système contrôlé, cette dernière peut être réalisée sous trois structures différentes, le choix de l'une des structures pour la commande d'un système variant dans le temps est faite selon cette variation de paramètre.

La deuxième partie de ce chapitre porte sur la commande par mode glissant qui représente le mode de fonctionnement principal des SSVs, l'avantage principal de cette loi de commande est sa simplicité de mise en œuvre, les différentes étapes de conception de cette loi de commande ont été présentées. Pour surmonter l'inconvénient de cette loi de commande qui est le phénomène de réticence dû à la partie de la commande discontinue, quelques méthodes appliquées dans la pratique sont présentées.

L'une des solutions de réduction du phénomène de chattering est l'utilisation des méthodes intelligentes en synergie avec la commande par mode glissant, notre objectif est de concevoir une loi de commande adaptative intelligente basée sur le mode glissant, la partie intelligente dans cette loi de commande est choisie sous deux formes, la première est le neuro-flou type-1, la seconde est le neuro-flou type-2.

Dans le chapitre suivant un développement de deux lois de commande est présenté, la commande adaptative neuro-flou type-1 mode glissant et la commande adaptative neuro-flou intervalle type-2 mode glissant.

Chapitre 5

Commande adaptative neuro-floue mode glissant

Les systèmes de contrôle deviennent de plus en plus importants de nos jours. Au début, l'industrie utilise des commandes séquentielles pour résoudre un grand nombre d'applications industrielles dans les systèmes de contrôle, puis les systèmes linéaires nous ont donné une énorme augmentation dans l'application de contrôle automatique linéaire dans le domaine industrielle. Une des méthodes les plus récentes pour contrôler les applications industrielles est le contrôle intelligent, qui est basée sur le comportement humain ou sur un processus naturel.

Les systèmes flous nous permettent de transférer la forme floue vague du raisonnement humain à des systèmes mathématiques. L'utilisation des règles SI-ALORS dans les systèmes flous nous donne la possibilité de comprendre facilement les informations modélisé par le système. Dans la plupart des systèmes flous la connaissance est obtenue à partir des experts humains. Cependant, cette méthode d'acquisition de l'information a un grand désavantage étant donné que pas tous les experts humain peuvent partager leurs connaissances.

Les réseaux de neurones artificiels (RNAs) peuvent apprendre de l'expérience, mais la plupart des topologies ne nous permettent pas de comprendre clairement l'information apprise par les réseaux. Les RNA incorporés dans les systèmes flous pour former des systèmes neuro-flous, peuvent acquérir des connaissances automatiquement par des algorithmes d'apprentissages des réseaux neuronaux. Les systèmes neuro-flous ont l'avantage sur les systèmes flous que les connaissances acquises, qui sont facile à comprendre, sont plus significatif pour les humains.

Comme les systèmes flous et les réseaux de neurones, les RNFs ont été prouvées pour être des approximateurs universels.

Dans ce chapitre, un algorithme d'apprentissage basé sur la théorie de la commande par mode glissant est proposé pour l'entraînement en ligne d'un système neuro-flou, et d'un

système neuro-flou intervalle type-2 en utilisant des fonctions d'appartenances gaussiennes. Dans l'algorithme d'apprentissage proposé, les poids des deux structures neuro-flou et neuro-flou-intervalle type-2 proposés sont réglés par l'algorithme proposé d'une manière à ce que l'erreur soit renforcée pour satisfaire une équation stable.

Les paramètres d'adaptations des règles d'inférences du réseau neuro-flou sont mises à jour par dérivation, et leurs convergence est prouvée par l'utilisation du théorème de stabilité de Lyapunov dans le cas des deux méthodes.

5.1 Commande adaptative neuro-floue par mode glissant

Le schéma de contrôle proposé est représenté sur la figure 5.1. Le contrôleur conventionnel proportionnel dérivée (PD) est utilisé à la fois comme un régulateur par retour d'état ordinaire pour garantir la stabilité asymptotique globale dans l'espace compact et comme modèle de référence inverse de la réponse du système commandé.

Le bloc NFC avec deux entrées et une sortie est un modèle neuro-flou ou sa structure met en oeuvre un modèle de TS tel que présenté sur la figure (5-2) [81, 83]. La loi de commande du régulateur PD est décrite comme suit :

$$\tau_c = K_P e + K_D \dot{e} \quad (5.1)$$

Avec $e(t)$ l'écart entre la consigne et la sortie calculée du système.

$e(t) = x_d - x$, x_d la valeur de la consigne. K_P, K_D sont les gains du régulateur PD.

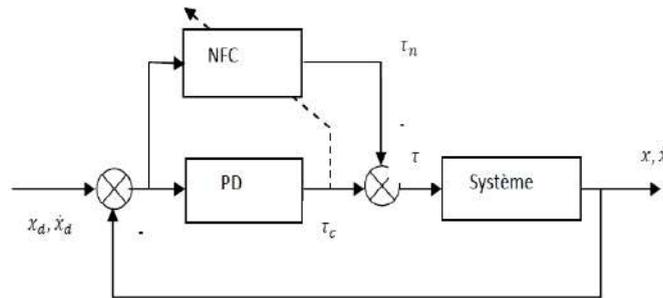


FIGURE 5.1: Schéma de commande adaptative neuro-floue.

5.2 Réseau neuro-flou

Le modèle flou utilisé est celui de Takagi-Sugeno ayant deux entrées $x_1 = e(t)$ et $x_2 = \dot{e}(t)$, et une sortie f_{ij} : La règle correspondante R_{ij} peut être exprimée comme :

$$R_{ij} : \text{SI } x_1 \text{ est } A_i \text{ et } x_2 \text{ est } B_j \text{ Alors } f_{ij} = a_i x_1 + b_j x_2 + d_{ij} \quad (5.2)$$

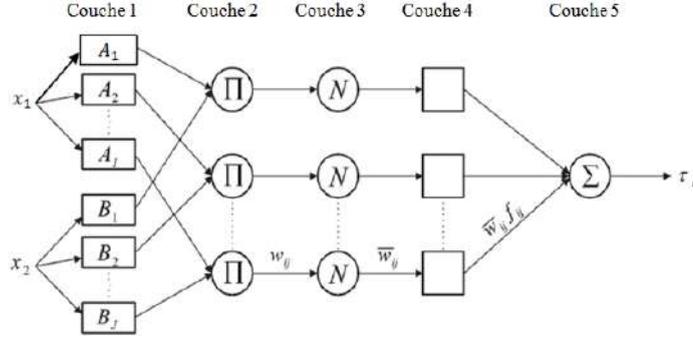


FIGURE 5.2: Architecture de contrôleur neuro-floue.

On le suppose que la sortie de chaque règle floue égale à d_{ij} seulement, qui est une simplification employée couramment

$$R_{ij} : SI \ x_1 \text{ est } A_i \text{ et } x_2 \text{ est } B_j \text{ Alors } f_{ij} = d_{ij} \quad (5.3)$$

Pour ce type de règles, on fait appel à un réseau de neurones qui admet pour entrées x_1, x_2 et dont les deux couches cachées correspondent respectivement au calcul de la valeur des fonctions d'appartenance $A_i, B_j, i=1..I, j=1..J$ (ou I et J représente étant le nombre de fonctions d'appartenance utilisées pour l'entrée 1 et l'entrée 2, respectivement) et à celui de la valeur prise par la conjonction des conditions de chaque règle utilisant un opérateur adéquat. Les fonctions d'appartenance intervenant dans les règles sont considérées comme des paramètres ajustés par l'intermédiaire des poids entrant dans la première couche cachée, les conclusions f_{ij} des règles sont également des paramètres ajustables par l'intermédiaire des poids à la dernière couche [81, 83].

Couche 1 :

La première couche est la couche d'entrée. Cette couche calcule le degré d'appartenance des entrées x_1 et x_2 en utilisant des fonctions d'appartenances gaussienne. Chaque noeud de cette couche est un noeud adaptatif avec un noeud de sortie défini par :

$$O_i^1 = \mu_{A_i}(x_1) \quad i = 1..I \quad (5.4)$$

$$O_j^1 = \mu_{B_j}(x_2) \quad j = 1..J \quad (5.5)$$

Pour $\mu_{A_i}(x_1)$ et $\mu_{B_j}(x_2)$ sont les degrés d'appartenance pour les entrées x_1 et x_2 respectivement Les fonctions gaussiennes $\mu_{A_i}(x_1)$ et $\mu_{B_j}(x_2)$ sont écrites comme suit :

$$\mu_{A_i}(x_1) = \exp \left(- \left(\frac{x_1 - c_{A_i}}{\sigma_{A_i}} \right)^2 \right) \quad (5.6)$$

$$\mu_{B_j}(x_2) = \exp \left(- \left(\frac{x_2 - c_{B_j}}{\sigma_{B_j}} \right)^2 \right) \quad (5.7)$$

Avec c_A et c_B désigne le centre de la fonction et σ_A et σ_B sa variance.

Couche 2 :

Tout les noeuds de cette couche sont des noeuds fixes étiquetés π , est donne en sortie la multiplication des signaux entrant.

$$O_{ij}^2 = w_{ij} = \mu_{A_i}(x_1)\mu_{B_j}(x_2) \quad (5.8)$$

L'équation précédente est réécrite comme suit :

$$O_{ij}^2 = w_{ij} = \mu_{A_i}(x_1) = \exp\left(-\left(\frac{x_1 - .c_{A_i}}{\sigma_{A_i}}\right)^2\right) \quad (5.9)$$

Couche 3 :

Tout les noeuds de cette couche sont des noeuds fixes étiquetés N . ces noeuds calculent le rapport de normalisation :

$$O_{ij}^3 = \bar{w}_{ij} = \frac{w_{ij}}{\sum_{i=1}^I \sum_{j=1}^J w_{ij}} \quad (5.10)$$

Couche 4 :

Chaque noeud de cette couche est un noeud adaptatif, sa fonction est :

$$O_{ij}^4 = \bar{w}_{ij}f_{ij} = \bar{w}_{ij}(a_ix_1 + b_jx_2 + d_{ij}) \quad (5.11)$$

Ou

$$O_{ij}^4 = \bar{w}_{ij}f_{ij} = \bar{w}_{ij}d_{ij} \quad (5.12)$$

Avec \bar{w}_{ij} est la sortie de la couche 3 et $\{a_ix_1 + b_jx_2 + d_{ij}\}$, est l'ensemble des paramètres. Les paramètres de cette couche sont les paramètres de la conséquence des règles.

Couche 5 :

Le seul noeud dans cette couche est un noeud fixe étiqueté \sum , qui calcule la sortie globale τ_n comme sommation de tous les signaux qui arrivent :

$$O^5 = \tau_n = \sum_{i=1}^I \sum_{j=1}^J \bar{w}_{ij}f_{ij} = \sum_{i=1}^I \sum_{j=1}^J \bar{w}_{ij}d_{ij} \quad (5.13)$$

La matrice des signaux de sortie normalisée à partir de la troisième couche est définie comme suit : $\bar{w}(t) = [\bar{w}_{11}(t), \bar{w}_{12}(t), \dots, \bar{w}_{21}(t), \dots, \bar{w}_{IJ}(t)]^T$, $\sigma_A = [\sigma_{A_1}, \sigma_{A_2}, \dots, \sigma_{A_I}]^T$, $\sigma_B = [\sigma_{B_1}, \sigma_{B_2}, \dots, \sigma_{B_J}]^T$ Sont les vecteurs des paramètres définissant la répartition 'l'écart type' des fonctions d'appartenances gaussiennes des deux entrées du réseau.

$c_A = [c_{A_1}, c_{A_2}, \dots, c_{A_I}]^T$, $c_B = [c_{B_1}, c_{B_2}, \dots, c_{B_J}]^T$ Sont les vecteurs des paramètres définissant les centres 'moyenne' des fonctions d'appartenances gaussiennes des deux entrées de réseau.

Les hypothèses suivantes sont utilisées pour garantir la stabilité :

Les deux signaux d'entrée $x_1(t)$, $x_2(t)$ et leur dérivées $\dot{x}_1(t)$, $\dot{x}_2(t)$ doivent être bornés comme suit :

$$|x_1(t)| \leq B_x \text{ et } |x_2(t)| \leq B_x \quad \forall t \quad (5.14)$$

$$|\dot{x}_1(t)| \leq B_{\dot{x}} \text{ et } |\dot{x}_2(t)| \leq B_{\dot{x}} \quad \forall t \quad (5.15)$$

Avec B_x et $B_{\dot{x}}$ des constantes positives connues. Les vecteurs définissant la répartition et les centres de fonctions d'appartenance gaussiennes sont supposés bornés comme suit :

$$\|\sigma_A\| \leq B_\sigma, \quad \|\sigma_B\| \leq B_\sigma \quad \forall t \quad (5.16)$$

$$\|c_A\| \leq B_c, \quad \|c_B\| \leq B_c \quad \forall t \quad (5.17)$$

Avec B_σ et B_c sont des constantes positives connues. On suppose qu'en raison de contraintes physiques, les coefficients variables de pondération de temps pour les connexions entre les neurones de la quatrième couche et le neurone de sortie sont bornés comme suit :

$$\|f_{ij}\| \leq B_f \quad \forall t \quad (5.18)$$

Pour une constante positive B_f .

de (5.10) on peut voir que $0 \leq \bar{w}_{ij} \leq 1$ et que $\sum_i^I \sum_j^J \bar{w}_{ij} = 1$.

L'entrée de commande du système τ est déterminée comme suit :

$$\tau = \tau_c - \tau_n \quad (5.19)$$

il sera également adopté que τ et $\dot{\tau}$ sont des signaux bornés :

$$\tau(t) \leq B_\tau, \quad |\dot{\tau}| \leq B_{\dot{\tau}} \quad \forall t \quad (5.20)$$

Avec B_τ et $B_{\dot{\tau}}$ sont des constantes positives connues.

5.3 L'algorithme d'apprentissage par mode glissant

En utilisant les principes de la théorie de commande par mode glissement. La valeur zéro de l'erreur d'apprentissage de coordonnées $\tau_c(t)$ peut être définie comme surface de glissement variant dans le temps.

$$S_c(\tau_c(t), \tau) = \tau_c(t) - \tau_n(t) + \tau(t) = 0. \quad (5.21)$$

Qui est la condition pour que le réseau Neuro-flou soit entraîné à ce qu'il devient un régulateur non linéaire pour atteindre la réponse désirée pendant le mouvement de convergence d'erreur de poursuite de trajectoire par compensation de la non linéarité du système commandé.

La surface de glissement pour le système non linéaire sous contrôle $S_P(e, \dot{e})$ est définie comme [82] :

$$S_p(e, \dot{e}) = \dot{e} + \chi e \quad (5.22)$$

Avec χ est une constante déterminant la pente de la surface de glissement.

Définition :

Un glissement aura lieu sur une surface de glissement $S_c(\tau_n, \tau) = \tau_c(t) = 0$ après un temps t_h si la condition : $S_c \dot{S}_c(t) = \tau_c(t) \dot{\tau}_c(t) < 0$ est satisfaite pour tout t dans un intervalle de temps de la forme $[t, t_h) \subset (-\infty, t_h)$.

L'objectif est de concevoir un mécanisme d'adaptation dynamique par retour d'état, ou l'algorithme d'apprentissage soit en ligne pour les paramètres de réseau neuro-flou tels que l'état du mode de glissement de la définition ci-dessus est appliquée.

Théorème 1 : Si les lois d'adaptations pour les paramètres du réseau neuro-flou considéré sont choisies respectivement comme suit [83]

$$c_{A_i} \dot{=} x_1 - \frac{\alpha}{S_A \sigma_A} \text{sign}(\tau_c) \quad (5.23)$$

$$c_{B_j} \dot{=} x_2 - \frac{\alpha}{S_B \sigma_B} \text{sign}(\tau_c) \quad (5.24)$$

$$\sigma_{A_i} \dot{=} \frac{\alpha}{S_A^2} \text{sign}(\tau_c) \quad (5.25)$$

$$\sigma_{B_j} \dot{=} \frac{\alpha}{S_B^2} \text{sign}(\tau_c) \quad (5.26)$$

$$f_{ij} \dot{=} -\frac{\bar{w}_{ij}}{\bar{w}^T \bar{w}} \text{sign}(\tau_c) \quad (5.27)$$

Pour $S_A = [S_{A_1}, S_{A_2}, \dots, S_{A_I}]_T$, $S_B = [S_{B_1}, S_{B_2}, \dots, S_{B_J}]_T$; $S_{A_i} = x_1 - c_{A_i}$, $S_{B_j} = x_2 - c_{B_j}$. et α est une constante positive satisfaisant l'inégalité :

$$B_{\dot{\tau}} < \alpha$$

Ensuite une condition initiale arbitraire $\tau_c(0)$ est donnée, alors l'erreur d'apprentissage $\tau_c(t)$ converge vers zéro pendant un temps fini t_h et le glissement est maintenu sur $\tau_c(t) = 0$ pour tous $t < t_h$.

Preuve :

Considérer la fonction de Lyapunov suivante :

$$V_c = \frac{1}{2} \tau_c^2(t)$$

Le dérivé dans le temps de V_c est donné par :

$$\dot{V}_c = \tau_c \dot{\tau}_c = \tau_c (\dot{\tau}_n + \dot{\tau})$$

On peut facilement montrer que :

$$\dot{V}_c = \tau_c \left(\sum_i^I \sum_j^J (f_{ij} \bar{w}_{ij} + f_{ij} \dot{\bar{w}}_{ij}) + \dot{\tau} \right)$$

$$\dot{\bar{w}}_{ij} = -\bar{w}_{ij}\dot{K}_{ij} + \bar{w}_{ij} \sum_i^I \sum_j^J \bar{w}_{ij}\dot{K}_{ij}$$

Pour :

$$\begin{aligned} \dot{K}_{ij} &= 2(A\dot{A} + B\dot{B}); \quad A\dot{A} + B\dot{B} = 0; \quad A = \frac{x_1 - c_{A_i}}{\sigma_{A_i}}; \quad B = \frac{x_2 - c_{B_j}}{\sigma_{B_j}} \\ \dot{A} &= \frac{(\dot{x}_1 - \dot{c}_{A_i})\sigma_{A_i} - \dot{\sigma}_{A_i}(x_1 - c_{A_i})}{\sigma_{A_i}^2} \quad \dot{B} = \frac{(\dot{x}_2 - \dot{c}_{B_j})\sigma_{B_j} - \dot{\sigma}_{B_j}(x_2 - c_{B_j})}{\sigma_{B_j}^2} \end{aligned}$$

On peut montrer que :

$$\begin{aligned} A\dot{A} &= 0 = \frac{(x_1 - c_{A_i})(\dot{x}_1 - \dot{c}_{A_i})\sigma_{A_i} - \dot{\sigma}_{A_i}(x_1 - c_{A_i})^2}{\sigma_{A_i}^3} \\ B\dot{B} &= 0 = \frac{(x_2 - c_{B_j})(\dot{x}_2 - \dot{c}_{B_j})\sigma_{B_j} - \dot{\sigma}_{B_j}(x_2 - c_{B_j})^2}{\sigma_{B_j}^3} \end{aligned}$$

Alors \dot{V}_c peut être exprimé comme suit :

$$\begin{aligned} \dot{V}_c &= \tau_c \left(\sum_i^I \sum_j^J \dot{f}_{ij} \bar{w}_{ij} + \dot{\tau} \right) = \tau_c (-\alpha \text{sign}(\tau_c) + \dot{\tau}) \\ &\leq (-\alpha |\tau_c| + \dot{\tau} |\tau_c|) \end{aligned} \quad (5.28)$$

$$\leq (-\alpha |\tau_c| + B_{\dot{\tau}} |\tau_c|) < 0 \quad \text{si} \quad B_{\dot{\tau}} < \alpha$$

L'inégalité (5.29) montre que les trajectoires contrôlées de l'erreur d'apprentissage $\tau_c(t)$ converge vers zéro de façon stable. La relation entre la surface de glissement S_p et le niveau zéro d'erreur d'apprentissage adaptatif S_c , si χ est prise comme $\chi = \frac{K_P}{K_D}$, est déterminée par l'équation suivante :

$$S_c = \tau_c = K_P e + K_D e = K_D \left(e + \frac{K_P}{K_D} e \right) = K_D S_p \quad (5.29)$$

Les performances de poursuite du système de contrôle de position peuvent être analysées en introduisant la fonction de Lyapunov candidate suivante :

$$V_p = \frac{1}{2} S_p^2 \quad (5.30)$$

Théorème 2 :

Si les paramètres d'adaptation du contrôleur neuro-flou 'NFC' sont choisis comme en (5.23 – 5.27), alors la dérivée par rapport au temps de la fonction de Lyapunov dans (5.31) est définie négative.

Preuve :

On calcule la dérivée de la fonction de Lyapunov 5.31, on obtient :

$$\begin{aligned} \dot{V}_p &= \dot{S}_p S_p = \frac{1}{K_D^2} \tau_c \dot{\tau}_c \\ \dot{V}_p &= \frac{1}{K_D^2} \dot{V}_c < 0, \quad \forall S_c; S_p \neq 0 \end{aligned}$$

Remarque :

Le résultat obtenu signifie qu'en supposant que la commande par mode glissant est réalisable, en utilisant τ_c comme erreur d'apprentissage pour le NFC conjointement avec les lois d'adaptation (5.23 – 5.27) impose le mode de poursuite du régime de glissement désiré pour le

système sous contrôle.

5.4 Réseau neuro-flou intervalle type-2

La structure du réseau neuro-flou intervalle type-2 utilisé est une structure à deux entrées $x_1 = e(t), x_2 = \dot{e}(t)$ et une sortie, dans laquelle le système flou proposé est un système flou de type-2 de TSK avec des ensembles flous de type-2 comme antécédents, et des valeurs fixées dans la partie conséquents (A2-C0). Le réseau neuro-flou intervalle type-2 'NFIT2' est implémenté dans le schéma de commande comme un dispositif de rétroaction. Le schéma de commande utilisé dans cette partie est le même schéma utilisé dans la partie précédente. Chaque règle R_{ij} peut être exprimée comme suit [84, 85] :

$$R_{ij} : \text{if } x_1 \text{ is } \tilde{A}_{1i} \text{ and } x_2 \text{ is } \tilde{A}_{2j} \text{ then } f_{ij} = d_{ij} \quad (5.31)$$

Avec $i = 1..I$ et $j = 1..J$

Ou $x_1 = e_1$ et $x_2 = \dot{e}_2$, sont les entrées du réseau et f_{ij} la sortie.

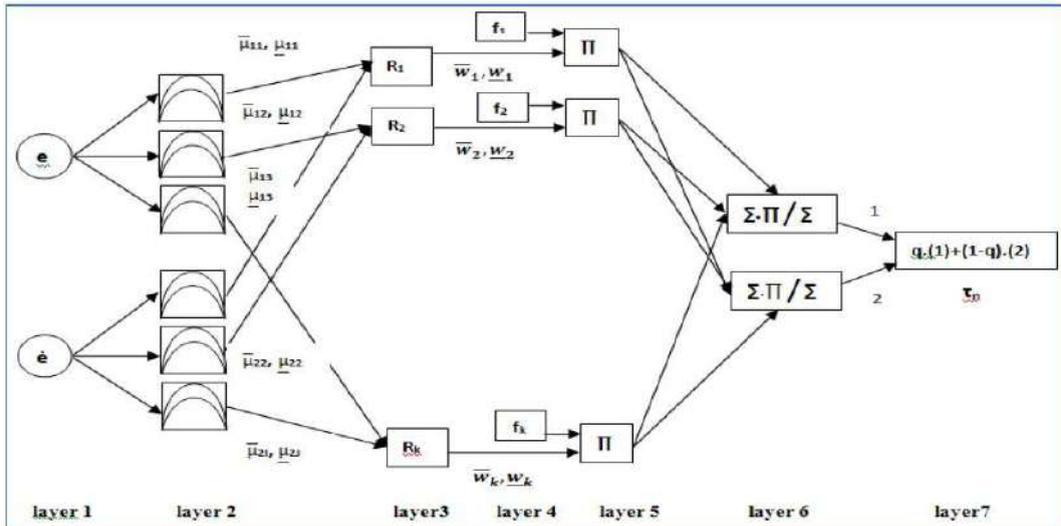


FIGURE 5.3: Architecture de contrôleur neuro-flou intervalle type-2.

Les paramètres de chaque fonction d'appartenance gaussienne (le centre c et l'écart type σ) sont les paramètres ajustables de la structure neuro-flou intervalle type-2.

$$\bar{\mu}_{1i}(x_1) = \exp \left[-\left(\frac{x_1 - \bar{c}_{1i}}{\bar{\sigma}_{1i}} \right)^2 \right] \quad (5.32)$$

$$\underline{\mu}_{1i}(x_1) = \exp \left[-\left(\frac{x_1 - \underline{c}_{1i}}{\underline{\sigma}_{1i}} \right)^2 \right] \quad (5.33)$$

$$\bar{\mu}_{2j}(x_2) = \exp \left[-\left(\frac{x_2 - \bar{c}_{2j}}{\bar{\sigma}_{2j}} \right)^2 \right] \quad (5.34)$$

$$\mu_{2j}(x_2) = \exp \left[-\left(\frac{x_2 - c_{2j}}{\sigma_{2j}} \right)^2 \right] \quad (5.35)$$

Dans la première couche les signaux x_1 et x_1 sont fuzzifiés en utilisant des fonctions Gaussiennes comme fonctions d'appartenance supérieure et aussi comme fonctions d'appartenances inférieures notés par $\bar{\mu}$ et $\underline{\mu}$. Le poids de chaque règle R_{ij} est obtenu dans la deuxième couche par l'application de l'opérateur produit de la T-norme des fonctions d'appartenances de la partie prémisse.

$$\bar{w}_{ij} = \bar{\mu}_{1i}(x_1) \cdot \bar{\mu}_{2j}(x_2) \quad (5.36)$$

$$\underline{w}_{ij} = \underline{\mu}_{1i}(x_1) \cdot \underline{\mu}_{2j}(x_2) \quad (5.37)$$

$$\tilde{w}_{ij} = \frac{\bar{w}_{ij}}{\sum_{i=1}^I \sum_{j=1}^J \bar{w}_{ij}} \quad (5.38)$$

$$\tilde{w}_{ij} = \frac{\underline{w}_{ij}}{\sum_{i=1}^I \sum_{j=1}^J \underline{w}_{ij}} \quad (5.39)$$

$$\tau_n = \frac{q \cdot \sum_{i=1}^I \sum_{j=1}^J \underline{w}_{ij} f_{ij}}{\sum_{i=1}^I \sum_{j=1}^J \underline{w}_{ij}} + \frac{(1-q) \cdot \sum_{i=1}^I \sum_{j=1}^J \bar{w}_{ij} f_{ij}}{\sum_{i=1}^I \sum_{j=1}^J \bar{w}_{ij}} \quad (5.40)$$

$$\tau_n = q \cdot \sum_{i=1}^I \sum_{j=1}^J (\tilde{w}_{ij} f_{ij}) + (1-q) \cdot \sum_{i=1}^I \sum_{j=1}^J (\tilde{w}_{ij} f_{ij}) \quad (5.41)$$

il est évident que $0 < \underline{w}_{ij} < 1$, et $0 < \tilde{w}_{ij} < 1$.

En outre on peut voir que : $\sum_{i=1}^I \sum_{j=1}^J (\tilde{w}_{ij}) = 1$, et $\sum_{i=1}^I \sum_{j=1}^J (\tilde{w}_{ij}) = 1$.

Les hypothèses suivantes ont été utilisées :

$$|\tau(t)| \leq B_\tau, \quad |\dot{\tau}(t)| \leq B_{\dot{\tau}} \quad \forall t \quad (5.42)$$

Avec B_τ et $B_{\dot{\tau}}$ sont des constantes positives. Nous sommes basés sur le même algorithme d'apprentissage par mode glissant utilisé dans la section précédente pour atteindre le résultat donné sous forme du théorème suivant :

Théorème 3 :

Si les lois d'adaptations pour les paramètres du réseau Neuro-flou considéré sont choisies respectivement comme suit [85]

$$\dot{c}_{1i} = \dot{x}_1 - \frac{1}{2} \alpha \frac{\sigma_{1i}^2}{(x_1 - c_{1i})} \text{sign}(\tau_c) \quad (5.43)$$

$$\dot{\bar{c}}_{1i} = \dot{x}_1 - \frac{1}{2} \alpha \frac{\bar{\sigma}_{1i}^2}{(x_1 - \bar{c}_{1i})} \text{sign}(\tau_c) \quad (5.44)$$

$$\dot{c}_{2j} = \dot{x}_2 - \frac{1}{2} \alpha \frac{\sigma_{2j}^2}{(x_2 - c_{2j})} \text{sign}(\tau_c) \quad (5.45)$$

$$\dot{\bar{c}}_{2j} = \dot{x}_2 - \frac{1}{2}\alpha \frac{\bar{\sigma}_{2j}^2}{(x_2 - \bar{c}_{2j})} \text{sign}(\tau_c) \quad (5.46)$$

$$\dot{\bar{c}}_{1i} = -\frac{1}{2}\alpha \frac{\bar{\sigma}_{1i}^3}{(x_1 - \bar{c}_{1i})^2} \text{sign}(\tau_c) \quad (5.47)$$

$$\dot{\bar{\sigma}}_{1i} = -\frac{1}{2}\alpha \frac{\bar{\sigma}_{1i}^3}{(x_1 - \bar{c}_{1i})^2} \text{sign}(\tau_c) \quad (5.48)$$

$$\dot{\bar{c}}_{2j} = -\frac{1}{2}\alpha \frac{\bar{\sigma}_{2j}^3}{(x_2 - \bar{c}_{2j})^2} \text{sign}(\tau_c) \quad (5.49)$$

$$\dot{\bar{\sigma}}_{2j} = -\frac{1}{2}\alpha \frac{\bar{\sigma}_{2j}^3}{(x_2 - \bar{c}_{2j})^2} \text{sign}(\tau_c) \quad (5.50)$$

$$\dot{d}_{ij} = -\frac{q\tilde{w} + (1-q)\tilde{\bar{w}}}{(q\tilde{w} + (1-q)\tilde{\bar{w}})^T(q\tilde{w} + (1-q)\tilde{\bar{w}})} \alpha \text{sign}(\tau_c) \quad (5.51)$$

α est une constante positive satisfaisant l'inégalité :

$$\alpha > B_{\dot{\tau}}$$

Ensuite une condition initiale arbitraire $\tau_c(0)$ est donnée, alors l'erreur d'apprentissage $\tau_c(t)$ converge vers zéro pendant un temps fini t_h et le glissement est maintenu sur $\tau_c(t) = 0$ pour tous $t > t_h$

Preuve :

Considérer la fonction de Lyapunov suivante :

$$V_c = \frac{1}{2}\tau_c^2(t)$$

Le dérivé dans le temps de V_c est donné par :

$$\dot{V}_c = \tau_c \dot{\tau}_c = \tau_c(\dot{\tau}_n + \dot{\tau})$$

$$\tau_n = q \sum_{i=1}^I \sum_{j=1}^J (\tilde{w}_{ij} f_{ij}) + (1-q) \sum_{i=1}^I \sum_{j=1}^J (\tilde{\bar{w}}_{ij} f_{ij}) \quad (5.52)$$

$$\dot{\tau}_n = q \sum_{i=1}^I \sum_{j=1}^J (\dot{f}_{ij} \tilde{w}_{ij} + f_{ij} \dot{\tilde{w}}_{ij}) + (1-q) \sum_{i=1}^I \sum_{j=1}^J (\dot{f}_{ij} \tilde{\bar{w}}_{ij} + f_{ij} \dot{\tilde{\bar{w}}}_{ij}) \quad (5.53)$$

$$\dot{\tilde{w}}_{ij} = \frac{\dot{w}_{ij}}{\sum_{i=1}^I \sum_{j=1}^J w_{ij}} - \frac{\tilde{w}_{ij} (\sum_{i=1}^I \sum_{j=1}^J \dot{w}_{ij})'}{\sum_{i=1}^I \sum_{j=1}^J w_{ij}} \quad (5.54)$$

$$\dot{\tilde{\bar{w}}}_{ij} = \frac{\dot{\bar{w}}_{ij}}{\sum_{i=1}^I \sum_{j=1}^J \bar{w}_{ij}} - \frac{\tilde{\bar{w}}_{ij} (\sum_{i=1}^I \sum_{j=1}^J \dot{\bar{w}}_{ij})'}{\sum_{i=1}^I \sum_{j=1}^J \bar{w}_{ij}} \quad (5.55)$$

$$\dot{\tilde{w}}_{ij} = -\tilde{w}_{ij} \dot{L}_{ij} + \tilde{w}_{ij} \sum_{i=1}^I \sum_{j=1}^J \tilde{w}_{ij} \dot{L} \quad (5.56)$$

$$\dot{\tilde{\bar{w}}}_{ij} = -\tilde{\bar{w}}_{ij} \dot{L}_{ij} + \tilde{\bar{w}}_{ij} \sum_{i=1}^I \sum_{j=1}^J \tilde{\bar{w}}_{ij} \dot{L} \quad (5.57)$$

$$\dot{\underline{L}}_{ij} = 2(A_{1i}\dot{A}_{1i} + A_{2j}\dot{A}_{2j}) \quad (5.58)$$

$$\dot{\bar{L}}_{ij} = 2(B_{1i}\dot{B}_{1i} + B_{2j}\dot{B}_{2j}) \quad (5.59)$$

$$A_{1i} = \frac{x_1 - c_{1i}}{\sigma_{1i}} \quad A_{2j} = \frac{x_2 - c_{2j}}{\sigma_{2j}} \quad (5.60)$$

$$B_{1i} = \frac{x_1 - \bar{c}_{1i}}{\bar{\sigma}_{1i}} \quad B_{2j} = \frac{x_2 - \bar{c}_{2j}}{\bar{\sigma}_{2j}} \quad (5.61)$$

$$\dot{\underline{L}}_{ij} = \dot{\bar{L}}_{ij} = 4\alpha \text{sign}(\tau_c) \quad (5.62)$$

$$\dot{\tau}_n = q \sum_{i=1}^I \sum_{j=1}^J (\dot{f}_{ij} \tilde{w}_{ij}) + (1-q) \sum_{i=1}^I \sum_{j=1}^J (\dot{f}_{ij} \underline{w}_{ij}) \quad (5.63)$$

$$\dot{V}_c = \tau_c \left(\sum_i \sum_j \dot{f}_{ij} (q \underline{w}_{ij} + (1-q) \tilde{w}_{ij}) + \dot{\tau} \right) \quad (5.64)$$

$$\dot{V}_c = \tau_c (-\alpha \text{sign}(\tau_c) + \dot{\tau}) \quad (5.65)$$

$$< (-\alpha|\tau_c| + B_{\dot{\tau}}|\tau_c| < 0 \quad (5.66)$$

La relation (5.67) montre que la dérivée de la fonction de Lyapunov \dot{V}_c est définie négative, cela démontre la convergence de $\tau_c(t)$ vers zéro d'une manière stable. Si γ est pris comme $\gamma = \frac{k_P}{k_D}$ alors S_p 'la ligne de glissement et S_c 'le niveau zéro de l'erreur d'apprentissage adaptatif' sont donnés par l'équation suivante :

$$S_c = \tau_c = K_P \dot{e} + K_D e = K_D \left(\dot{e} + \frac{K_P}{K_D} e \right) = K_D S_p \quad (5.67)$$

La fonction candidate Lyapunov suivante est choisie pour l'étude des performances de suivi de la trajectoire du système sous contrôle :

$$V_p = \frac{1}{2} S_p^2 \quad (5.68)$$

Théorème 4

Si les paramètres d'adaptation du 'NIT2FLC' sont choisis comme en (5.44-5.52), alors la dérivée par rapport au temps de la fonction de Lyapunov dans (5.69) est définie négative.

preuve

On calcule la dérivée de la fonction de Lyapunov (5.31), on obtient :

$$\begin{aligned} \dot{V}_p &= \dot{S}_p S_p = \frac{1}{K_D^2} \tau_c \dot{\tau}_c \\ \dot{V}_p &= \frac{1}{K_D^2} \dot{V}_c < 0, \quad \forall S_c; S_p \neq 0 \end{aligned}$$

Le résultat obtenu montre qu'avec τ_c utilisé comme erreur d'apprentissage pour le contrôleur NIT2FLC et sur la base des lois d'adaptation données dans le théorème 3, le système contrôlé atteint la surface de glissement et y reste.

5.5 Conclusion

La structure de commande sur laquelle on s'est basé pour concevoir un contrôleur adaptatif neuro-flou mode glissant et un contrôleur neuro-flou intervalle type-2 mode glissant a été présenté dans ce chapitre, cette structure contenait un régulateur PD en parallèle avec un système neuro-flou dans une première partie et dans la deuxième partie le régulateur DP est placé en parallèle avec un système neuro-flou intervalle type-2. Dans les deux méthodes le contrôleur PD est utilisé comme un régulateur par retour d'état ordinaire pour garantir la stabilité asymptotique globale et comme un modèle de référence inverse de la réponse du système commandé.

Nous avons développé la structure neuro-flou en couche jusqu'à atteindre la sortie τ_n . La sortie globale de la commande du système est une combinaison de la sortie du régulateur PD avec la sortie du système neuro-flou dont les paramètres sont ajustés en ligne.

La détermination des paramètres d'adaptation a été basée sur le théorème du mode glissant, et pour prouver la convergence nous avons utilisé le théorème de Lyapunov.

Ainsi pour prouver les performances des deux lois de commande mis au point dans ce chapitre, nous allons les implémentés sur une plateforme expérimentale, le système de grue 3D. Dans le chapitre qui suit nous allons présenter les résultats expérimentaux de l'implémentation des deux lois de commande, les résultats de tests de performances comme perturbation externe et changement de trajectoire de référence seront aussi présentés.

Chapitre 6

Résultats expérimentaux

DANS de nombreux environnements industriels, les systèmes de grue sont largement utilisés pour le transfert rapide et précis de charges lourdes sur de longues distances. Toutefois, l'accélération du chariot de la grue, nécessaire pour le mouvement, induit toujours des oscillations indésirables de la charge suspendue. Ces oscillations de la charge réduisent considérablement la sécurité et augmentent le risque de chute et de collision, de ce fait la conception de systèmes de contrôle automatique pour les grues est devenue une tâche primordiale. Cependant, les systèmes de contrôle de grue 3D sont très complexes, car ils ont la caractéristique d'avoir un nombre de variables (sorties) commandées supérieur au nombre de variables de commande (entrées).

Dans cette section nous allons appliquer les deux stratégies de contrôle ANNFLC et ANIT2FLC, développées dans le chapitre précédent, sur le système de grue 3D de INTECO, les résultats expérimentaux seront présentés.

6.1 Le système 3DCrane

Le système de grue est un système électromécanique, qui est utilisé pour soulever et abaisser des matériaux et de les déplacer à partir d'un endroit à un autre. Il est principalement utilisé pour soulever des choses lourdes. Il existe plusieurs types de systèmes de grue, dans notre travail le système « 3D Crane » présente la plateforme expérimentale utilisée, dans ce dernier, la charge utile est attachée à l'extrémité d'un fil et peut se déplacer librement dans les trois dimensions.

Le modèle comporte trois moteurs à courant continu de commande et cinq codeurs de mesure de position angulaire. Le système 3D Crane est intégré à MATLAB / Simulink et

fonctionne en temps réel. Le logiciel permet le prototypage rapide d'algorithmes de contrôle en temps réel.

Le modèle mathématique de « 3DCrane » est assez compliqué. De nombreuses tentatives ont été faites pour introduire des modèles simplifiés ou d'utiliser des techniques de contrôle pour faire face aux incertitudes de la modélisation du système.

L'objectif principal dans le contrôle du système de grue 3D est de positionner la charge à un endroit désiré avec un minimum d'oscillations [86].



FIGURE 6.1: Le système 3DCrane de laboratoire.

6.2 Le modèle mathématique du system « 3DCrane »

Un élément important dans la construction du modèle mathématique est le choix approprié du système de coordonnées. Malgré le fait que le système cartésien est simple d'interprétation, il n'est pas pratique pour la description de la dynamique du mouvement de rotation, de sorte que nous choisissons le système sphérique. Comme nous pouvons le voir sur la figure (6.2), la position de la charge utile est décrite par deux angles α et β . La position de la charge utile est décrite par les équations suivantes :

$$x_c = x_w + R \cdot \sin \alpha \cdot \sin \beta \quad (6.1)$$

$$y_c = y_w + R \cdot \cos \beta \quad (6.2)$$

$$z_c = -R \cdot \sin \alpha \cdot \cos \beta \quad (6.3)$$

Où x_w représente la distance du rail avec le chariot du centre du cadre de la construction [m], et y_w est la distance entre le chariot du centre du rail [m], R est la longueur du fil de levage [m], α est l'angle entre l'axe y et le fil levage [rad], β est l'angle entre la direction négative sur l'axe z et la projection de fil de levage sur le plan xz [rad]. De manière similaire,

la dynamique de la grue peut être obtenue comme suit :

$$m_c \cdot \ddot{x}_c = -S_x \quad (6.4)$$

$$m_c \cdot \ddot{y}_c = -S_y \quad (6.5)$$

$$m_c \cdot \ddot{z}_c = -S_z - m_c \cdot g \quad (6.6)$$

$$(m_w + m_s) \cdot \ddot{x}_w = F_x - T_x + S_x \quad (6.7)$$

$$m_w \cdot \ddot{y}_w = F_y - T_y + S_y \quad (6.8)$$

Où S_x , S_y et S_z sont les composant du vecteur S, avec S la force de réaction dans le fil de levage agissant sur le chariot, F_x est la force agissant sur la rail avec chariot [N], F_y est force agissant sur le chariot le long du rail [N], m_c est la masse de la charge portée par la grue [kg], m_w est la masse du chariot [kg] et m_s représente la masse de la rail mobile [kg].

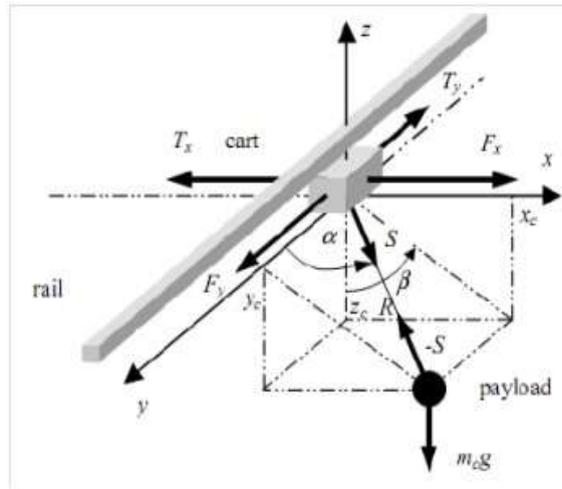


FIGURE 6.2: Représentation des coordonnées et forces du système 3DCrane.

$$S_x = S \cdot \sin \alpha \cdot \sin \beta \quad (6.9)$$

$$S_y = S \cdot \cos \alpha \quad (6.10)$$

$$S_z = -S \cdot \sin \alpha \cdot \cos \beta \quad (6.11)$$

Le modèle mathématique réduit du système de grue 3D est complètement déterminé par ces équations [86].

6.3 Résultats expérimentaux

Dans cette section, l'implémentation de la commande adaptatif neuro-floue basée sur la théorie de mode de glissement proposé est effectuée sur la plateforme expérimentale de la grue 3D présentée dans la figure 6.1. Les algorithmes de contrôle sont implémentés sur le logiciel Matlab/Simulink.

Les figures présentés dans cette section montrent les résultats expérimentaux obtenus par l'application de la commande adaptative neuro-floue type-1 mode glissant sur système 3DCrane. Les paramètres utilisés dans l'implémentation de l'algorithme sont les suivants :

Les signaux de référence sont des signaux rectangulaires avec les fréquences suivantes :

$$f_x = f_z = 0.05hz ; f_y = 0.04hz.$$

les amplitudes sont :

$$A_x = 0.3m, A_y = 0.37m, A_z = 0.32m.$$

Les valeurs initiales des paramètres de la loi de commande, et les gains du régulateur PD utilisés sont les suivants :

Pour I=3 et J=3, le nombre de règles d'inférence pour chaque contrôleur avec deux entrées est 9 règles.

Pour l'axe x :

$$c_{Ax} = \begin{pmatrix} 2.99 \\ 3.99 \\ 4.99 \end{pmatrix} c_{Bx} = \begin{pmatrix} 2.34 \\ 4.34 \\ 6.34 \end{pmatrix} \sigma_{Ax} = \begin{pmatrix} 1.95 \\ 23.14 \\ 2.18 \end{pmatrix} \sigma_{Bx} = \begin{pmatrix} 1.92 \\ 49.89 \\ 2.05 \end{pmatrix} f_x = \begin{pmatrix} -0.384 & -0.247 & -0.12 \\ -0.022 & 0.078 & 0.173 \\ -0.117 & 0.216 & -0.101 \end{pmatrix}$$

$$k_{PX} = 3, k_{DX} = 2$$

Pour l'axe y :

$$c_{Ay} = \begin{pmatrix} 2.39 \\ 3.39 \\ 4.39 \end{pmatrix} c_{By} = \begin{pmatrix} 1.67 \\ 3.67 \\ 5.67 \end{pmatrix} \sigma_{Ay} = \begin{pmatrix} 2.06 \\ 20.17 \\ 2.23 \end{pmatrix} \sigma_{By} = \begin{pmatrix} 1.84 \\ 49.9 \\ 2.09 \end{pmatrix} f_y = \begin{pmatrix} -0.385 & -0.161 & -0.155 \\ 0.035 & 0.065 & 0.162 \\ 0.121 & 0.251 & -0.118 \end{pmatrix}$$

$$k_{PY} = 3, k_{DY} = 2$$

Pour l'axe z :

$$c_{Az} = \begin{pmatrix} -1.02 \\ -0.02 \\ -0.98 \end{pmatrix} c_{Bz} = \begin{pmatrix} -3.1 \\ -1.1 \\ 0.9 \end{pmatrix} \sigma_{Az} = \begin{pmatrix} 2.06 \\ 33.42 \\ 2.14 \end{pmatrix} \sigma_{Bz} = \begin{pmatrix} 2.01 \\ 5.72 \\ 2.09 \end{pmatrix} f_z = \begin{pmatrix} -0.7 & 0.358 & -0.208 \\ -0.006 & 0.108 & 0.208 \\ 0.073 & 0.027 & 0.127 \end{pmatrix}$$

$$k_{PZ} = 10, k_{DZ} = 1.$$

c_{Ax}, σ_{Ax} sont les vecteurs des paramètres des fonctions gaussiennes de la première entrée du contrôleur de l'axe x.

c_{Bx}, σ_{Bx} sont les vecteurs des paramètres des fonctions gaussiennes de la deuxième entrée du contrôleur de l'axe x.

f_x est la matrice des sorties des règles d'inférences pour le contrôleur de l'axe x.

le pas de simulation utilisé est : $t = 5.10^{-4}s$.

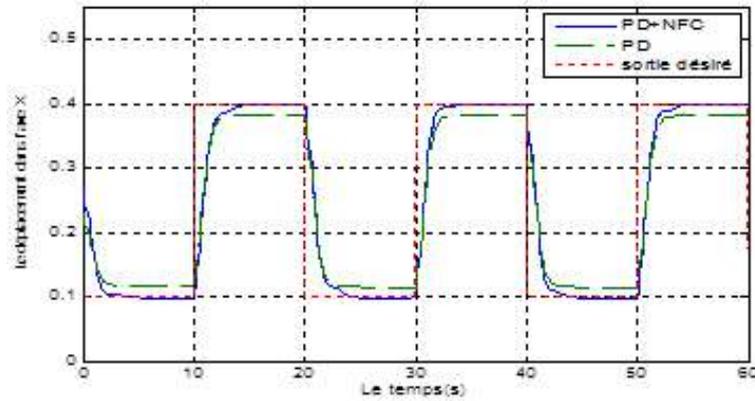


FIGURE 6.3: Poursuite de trajectoire par rapport l'axe X.

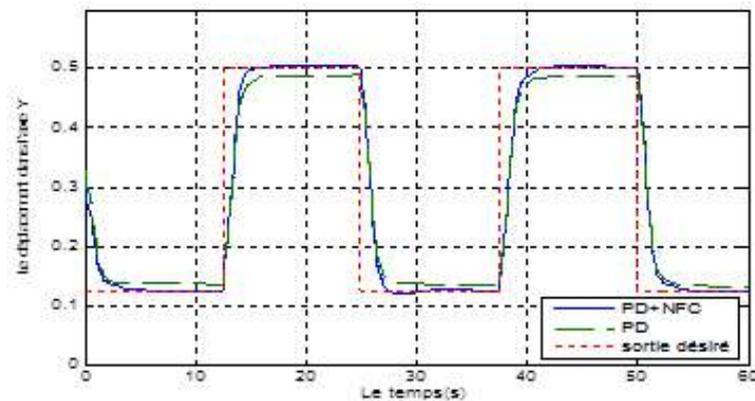


FIGURE 6.4: Poursuite de trajectoire par rapport l'axe Y.

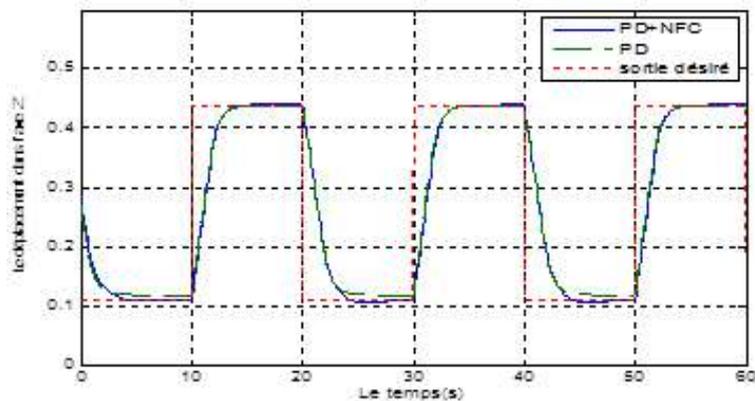


FIGURE 6.5: Poursuite de trajectoire par rapport l'axe Z.

Interprétation des résultats :

Nous avons vu dans les figures (6.3 – 6.5) les résultats expérimentaux des réponses du système par une commande PD et les réponses du système par une commande adaptative neuro-flou mode glissant (PD+NFC). On peut voir qu'avec les paramètres initiaux choisis, le contrôleur adaptatif atteint la trajectoire de référence dans une durée de temps fini avec un minimum de balancement au niveaux des angles α et β comme on peut le voir dans la

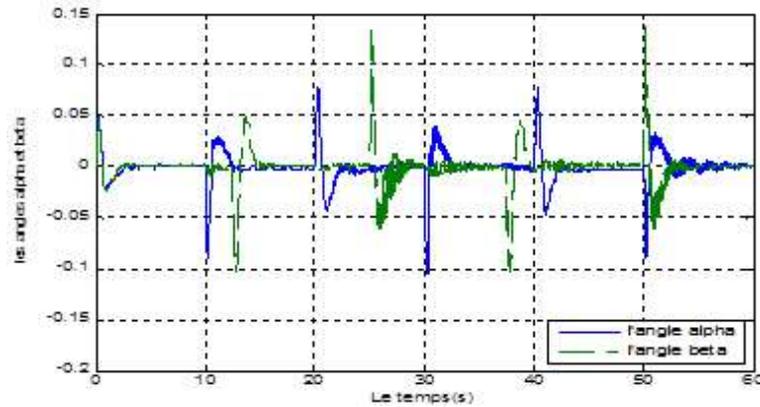
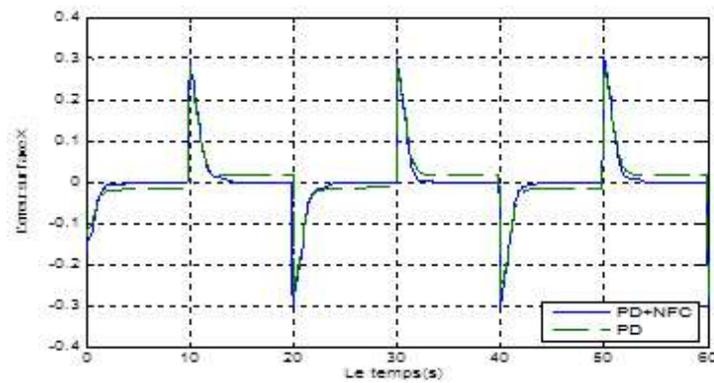
FIGURE 6.6: Balancement des angles α et β en rad.

FIGURE 6.7: Erreur de poursuite sur l'axe X.

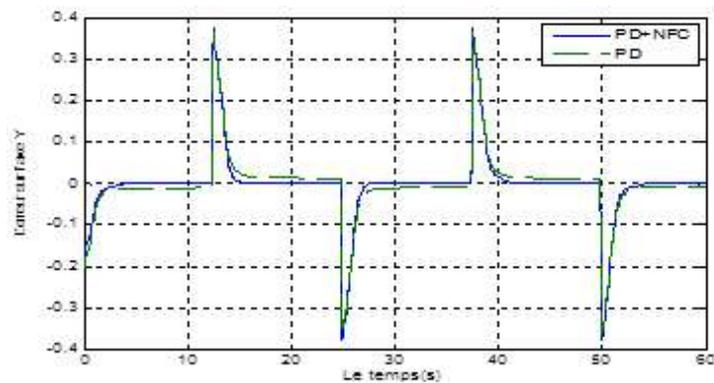


FIGURE 6.8: Erreur de poursuite sur l'axe Y.

figure (6.6), de même on peut observer la correction réalisée par le contrôleur ANFSMC du contrôleur PD.

Dans les figures (6.10-6.12) nous avons vu que les signaux de commande du contrôleur adaptatif utilisé force le systèmes à suivre la trajectoire désiré, même si ces signaux sont perturbés, et s'adapte pour obtenir un signal hybride qui donne une bonne poursuite de sortie désirée, et pour la minimisation d'erreur jusqu'à l'obtention d'une erreur qui tend vers

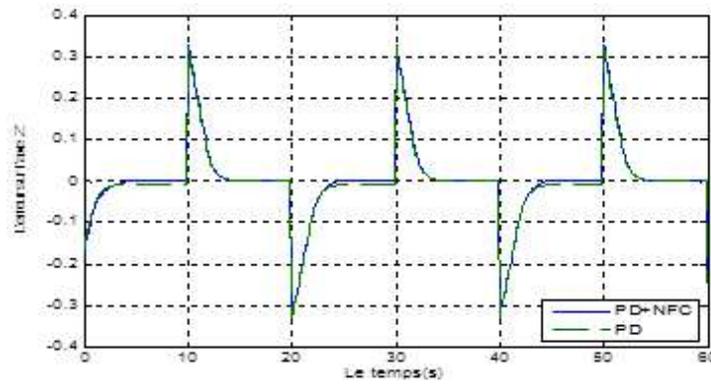


FIGURE 6.9: Erreur sur l'axe Z.

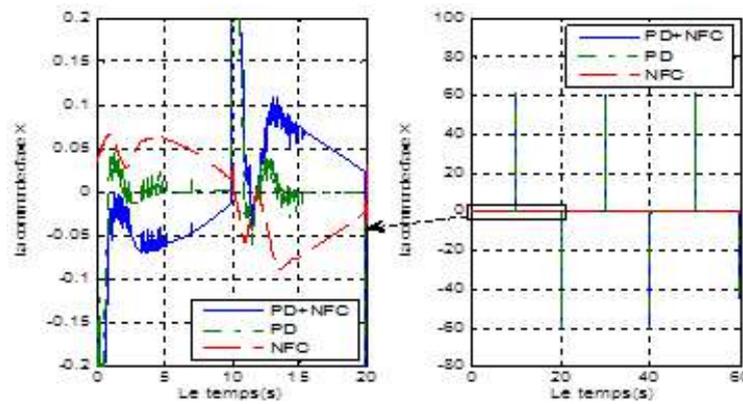


FIGURE 6.10: Signal de commande dans l'axe X.

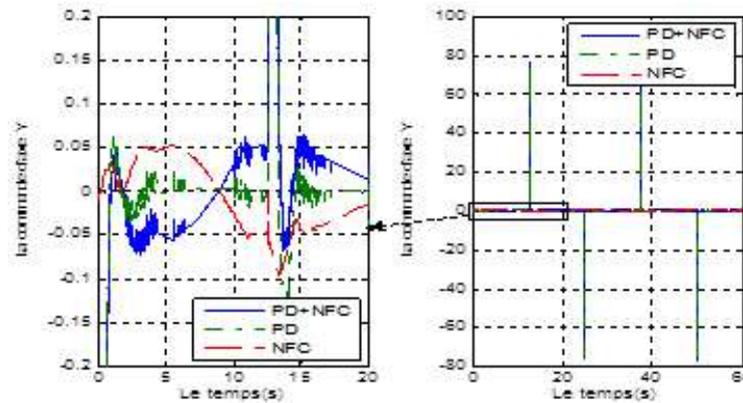


FIGURE 6.11: Signal de commande dans l'axe Y.

zéro dans le régime permanent figures (6.7-6.9).

Ce résultat de commande et d'erreur nous permet d'obtenir la poursuite de trajectoire désirée de la charge transportée par le système 3DCrane avec un minimum de balancement possible (de valeur maximum de l'angle α est de l'ordre de 0.125 rad et l'angle β est de 0.2 rad) ainsi les résultats obtenus peuvent valider l'algorithme de commande utilisé.

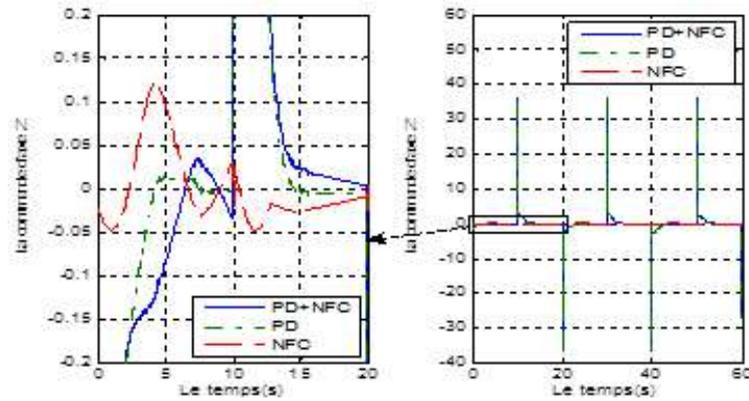


FIGURE 6.12: Signal de commande dans l'axe Z.

6.4 Tests des performances

Comme test de performance de l'algorithme de commande développé nous avons appliqué une perturbation externe sur la charge transporté par la grue, ensuite nous avons utilisé un signal sinusoïdale comme trajectoire de référence, les résultats obtenus pour chacun des tests sont montrés dans la partie qui suit :

6.4.1 Perturbation externe

Dans ce cas nous avons appliqué une perturbation sur la charge portée dans le plan oxy a l'instant $t = 2.3s$.

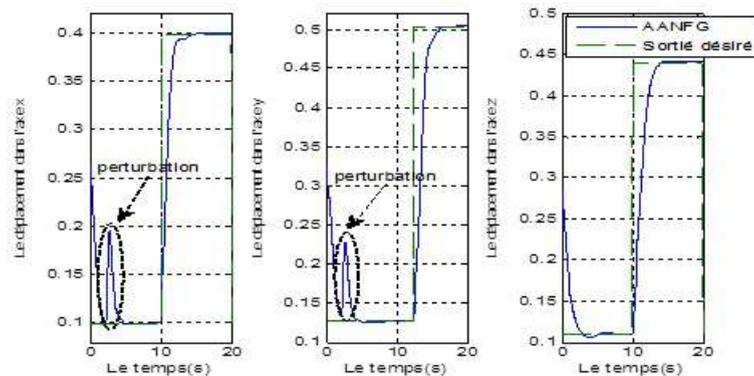


FIGURE 6.13: Poursuite de trajectoire par rapport aux axes (x, y, z).

6.4.2 Changement de la trajectoire de référence

Dans cette partie nous avons utilisé des trajectoires de référence sinusoïdale pour les trois sous systèmes. Les fréquences de ces signaux de référence sont les suivants : $f_x = f_y = 0,05hz$; $f_z = 0,02hz$.

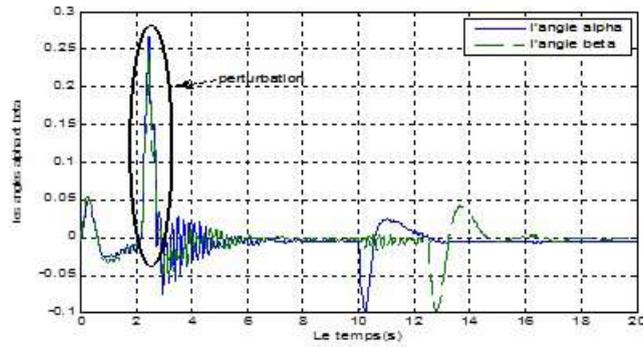


FIGURE 6.14: Balancement des angles α et β .

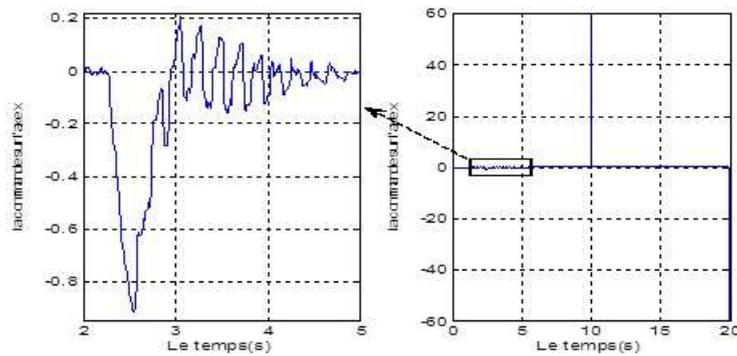


FIGURE 6.15: Signal de commande dans l'axe X.

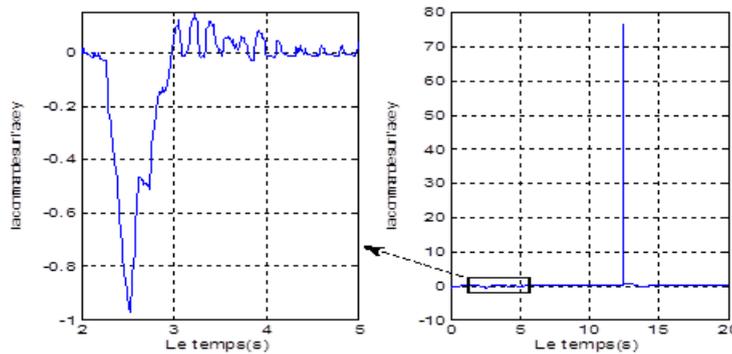


FIGURE 6.16: Signal de commande dans l'axe Y.

Interprétations des résultats :

Dans le cas de perturbation externe, on peut voir dans les figures (6.15-6.17) que la commande réagit fortement au moment de la perturbation et que l'algorithme de commande force le système à revenir vers la trajectoire désirée, et élimine la perturbation en un temps très faible, et donne un très bon résultat de poursuite de trajectoire de référence selon les trois axes comme montré dans la figure (6.13), le balancement provoqué par la perturbation dans les angles α et β est minimisé par cette commande comme indiqué dans la figure (6.14).

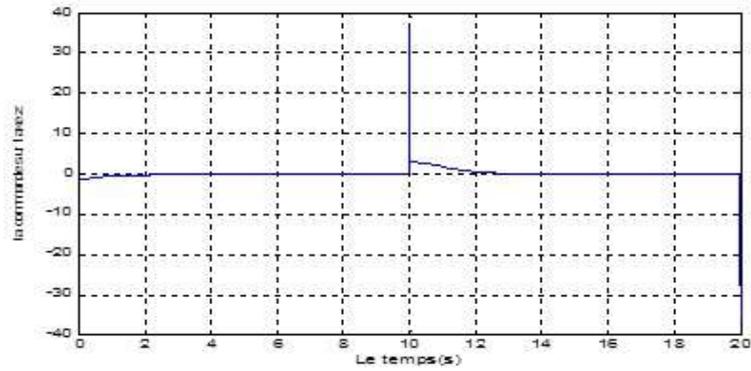


FIGURE 6.17: Signal de commande dans l'axe Z.

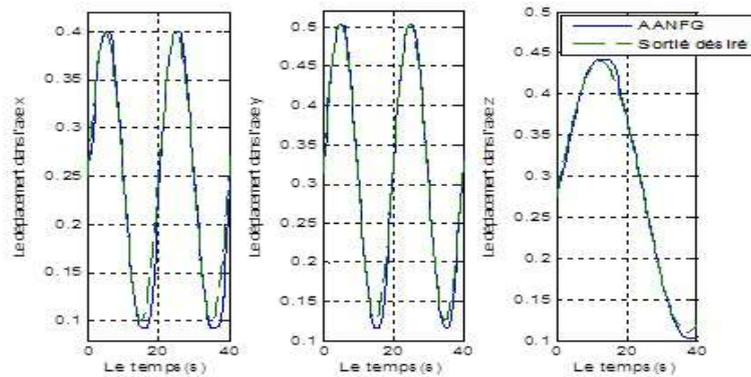


FIGURE 6.18: Poursuite de trajectoire par rapport aux axes (x, y, z).

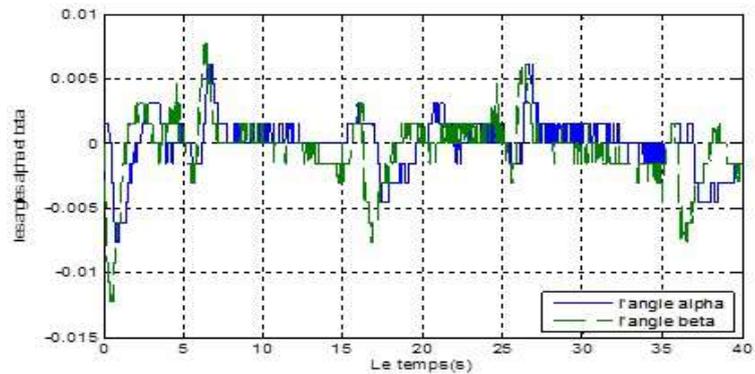


FIGURE 6.19: Balancement des angles α et β .

Pour le cas de modification de trajectoire, nous avons vu dans la figure (6.18) que le système donne un bon résultat de poursuite de trajectoire désirée, en utilisant les mêmes paramètres de commande pour le cas de trajectoire rectangulaire, l'algorithme de commande s'adapte avec le changement de trajectoire pour minimiser le balancement des angles α et β comme montrent les figures (6.19, 6.20).

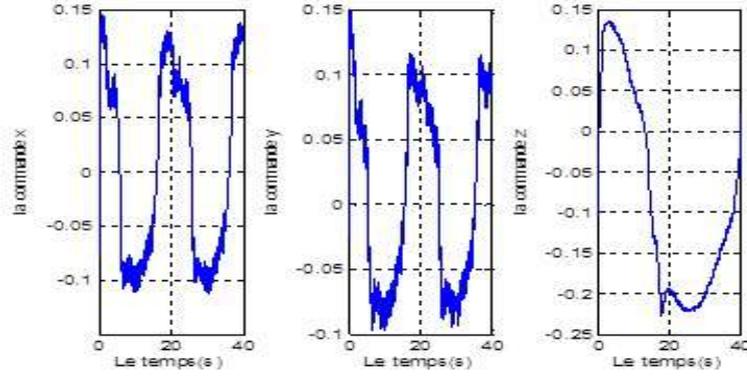


FIGURE 6.20: Signaux de commande des axes (x, y, z).

6.5 Commande adaptative neuro-floue intervalle type-2 mode glissant

Dans cette section, l'implémentation de la commande adaptative neuro-floue intervalle de type-2 basée sur la théorie de mode de glissement proposé est effectuée sur la même plate forme expérimentale.

La commande proposée a été évaluée en poursuite de trajectoire de la charge portée par la grue selon les trois axes x, y et z avec un minimum balancement. La commande proposée a été mise en œuvre par des contrôleurs en temps réel avec une période d'échantillonnage de 10 ms.

Les paramètres initiaux utilisés (les gains des régulateurs PD, les paramètres des fonctions gaussiennes supérieures et inférieures et les sorties des règles d'inférences) avec lesquels nous avons obtenu ces résultats expérimentaux sont les suivants :

$$k_P = (k_{Px}, k_{Py}, k_{Pz}) = (3.7, 10, 10) \text{ et } k_D = (k_{Dx}, k_{Dy}, k_{Dz}) = (0.5, 2, 2)$$

Les paramètres de l'axe X :

$$c_{Ax} = \begin{pmatrix} 2.741 \\ -0.140 \\ 1.409 \end{pmatrix}; c_{Bx} = \begin{pmatrix} 2.339 \\ 4.339 \\ 6.339 \end{pmatrix}; \sigma_{Ax} = \begin{pmatrix} 1.95 \\ 2.145 \\ 2.181 \end{pmatrix}; \sigma_{Bx} = \begin{pmatrix} 1.922 \\ 4.893 \\ 2.053 \end{pmatrix}; c_{Ahx} = \begin{pmatrix} 2.992 \\ 3.992 \\ 4.992 \end{pmatrix}$$

$$c_{Bhx} = \begin{pmatrix} 2.339 \\ 4.339 \\ 6.339 \end{pmatrix}; \sigma_{Ahx} = \begin{pmatrix} 1.95 \\ 2.14 \\ 2.18 \end{pmatrix}; \sigma_{Bhx} = \begin{pmatrix} 1.92 \\ 4.89 \\ 2.05 \end{pmatrix}; f_x = \begin{pmatrix} -0.3842 & -0.2466 & -0.120 \\ -0.0222 & 0.0777 & 0.173 \\ -0.1169 & 0.2162 & -0.101 \end{pmatrix}$$

Les paramètres de l'axe Y :

$$c_{Ay} = \begin{pmatrix} 2.078 \\ 2.319 \\ 3.183 \end{pmatrix}; c_{By} = \begin{pmatrix} 1.67 \\ 3.67 \\ 5.67 \end{pmatrix}; \sigma_{Ay} = \begin{pmatrix} 1.95 \\ 2.145 \\ 2.181 \end{pmatrix}; \sigma_{By} = \begin{pmatrix} 1.922 \\ 4.893 \\ 2.053 \end{pmatrix}; c_{Ahy} = \begin{pmatrix} 2.99 \\ 3.99 \\ 4.99 \end{pmatrix}$$

$$c_{Bhy} = \begin{pmatrix} 2.338 \\ 4.338 \\ 6.338 \end{pmatrix}; \sigma_{Ahy} = \begin{pmatrix} 1.95 \\ 2.145 \\ 2.181 \end{pmatrix}; \sigma_{Bhy} = \begin{pmatrix} 1.92 \\ 4.89 \\ 2.05 \end{pmatrix}; f_y = \begin{pmatrix} -0.308 & -0.140 & -0.154 \\ 0.268 & 0.142 & 0.168 \\ 0.0394 & 0.322 & -0.118 \end{pmatrix}$$

Les paramètres de l'axe Z :

$$c_{Az} = \begin{pmatrix} -0.75 \\ -0.097 \\ -0.109 \end{pmatrix}; c_{Bz} = \begin{pmatrix} -3.1 \\ -1.1 \\ 0.9 \end{pmatrix}; \sigma_{Az} = \begin{pmatrix} 1.95 \\ 2.145 \\ 2.181 \end{pmatrix}; \sigma_{Bz} = \begin{pmatrix} 1.922 \\ 4.893 \\ 2.053 \end{pmatrix}; c_{Ahz} = \begin{pmatrix} 0.378 \\ 0.378 \\ 0.379 \end{pmatrix}$$

$$c_{Bhz} = \begin{pmatrix} 2.338 \\ 4.338 \\ 6.338 \end{pmatrix}; \sigma_{Ahz} = \begin{pmatrix} 1.95 \\ 2.14 \\ 2.18 \end{pmatrix}; \sigma_{Bhz} = \begin{pmatrix} 1.92 \\ 4.89 \\ 2.05 \end{pmatrix}; f_z = \begin{pmatrix} -0.863 & 0.496 & -0.728 \\ -0.224 & 0.023 & -0.142 \\ 0.722 & 0.336 & 0.080 \end{pmatrix}$$

Ou $c_A = \bar{c}_1$, $c_B = \bar{c}_2$, $c_{Ah} = \underline{c}_1$, $c_{Bh} = \underline{c}_2$. $\sigma_A = \bar{\sigma}_1$, $\sigma_B = \bar{\sigma}_2$, $\sigma_{Ah} = \underline{\sigma}_1$, $\sigma_{Bh} = \underline{\sigma}_2$

Sur la base de la méthode d'apprentissage obtenue dans le théorème 3, les résultats présentés de poursuite de trajectoire le long des trois axes ainsi que les angles d'oscillations sont obtenus avec $q = 0,5$.

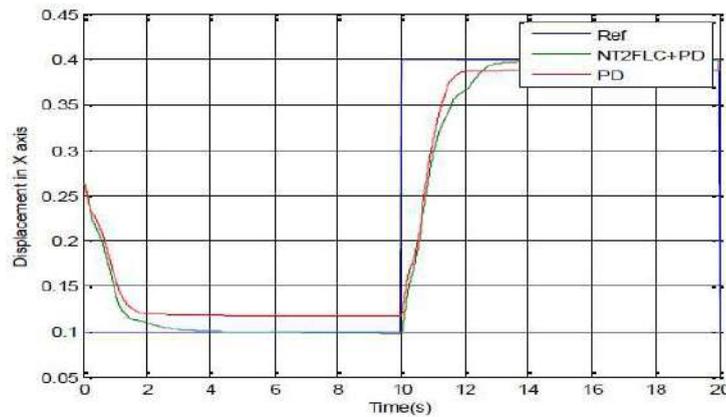


FIGURE 6.21: Poursuite de trajectoire par rapport l'axe X.

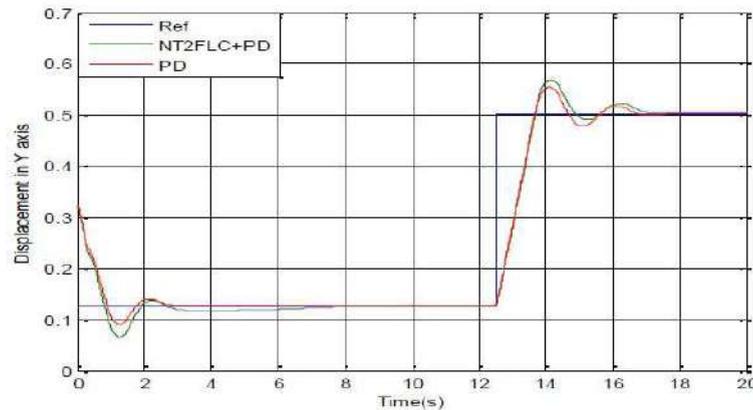


FIGURE 6.22: Poursuite de trajectoire par rapport l'axe Y.

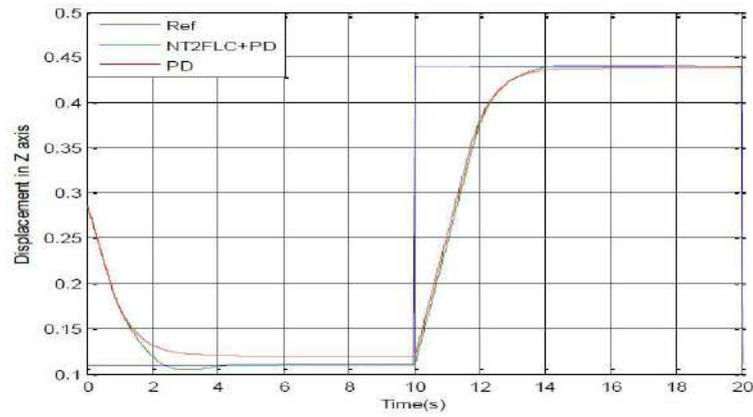


FIGURE 6.23: Poursuite de trajectoire par rapport l'axe Z.

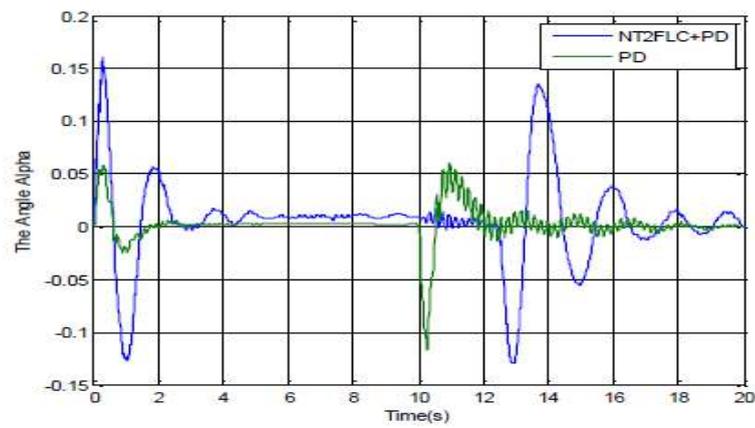


FIGURE 6.24: Balancement de l'angle α .

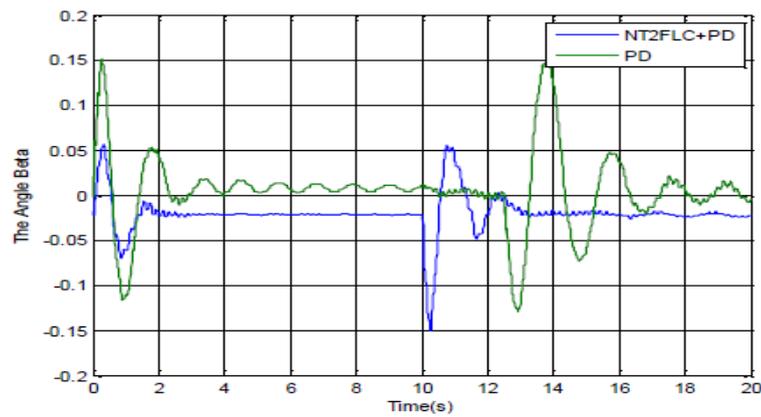


FIGURE 6.25: Balancement de l'angle β .

Les résultats expérimentaux présentés dans les figures (6.21-6.23) montrent la différence entre la réponse du système avec un régulateur PD et le contrôleur découplé neuro-floue intervalle type-2 par mode glissant proposé dans la section 3, et ceux pour le trois axes X, Y et Z. Nous avons observé dans ces résultats avec les paramètres k_p , k_d choisis que l'erreur entre la trajectoire désirée et la trajectoire effectuée par le système a été corrigée par la commande adaptative neuro-floue intervalle type-2 selon les trois axes avec un minimum d'oscillation au niveau des angles alpha et bêta figure 6.24 et 6.25.

On peut clairement voir sur la figure 6.25 que l'oscillation produite par la loi de commande proposée est plus petite que celle produite par un contrôle PD et atteint le minimum plus rapidement.

Afin de tester la robustesse de la commande adaptative neuro-flou intervalle type-2 mode glissant, nous avons appliqué une perturbation externe sur la charge portée par la grue au temps $t=4s$; nous avons remarqué que la commande minimise le balancement de charge par rapport aux deux axes du plan xoy en un temps fini figure 6.26 et 6.27.

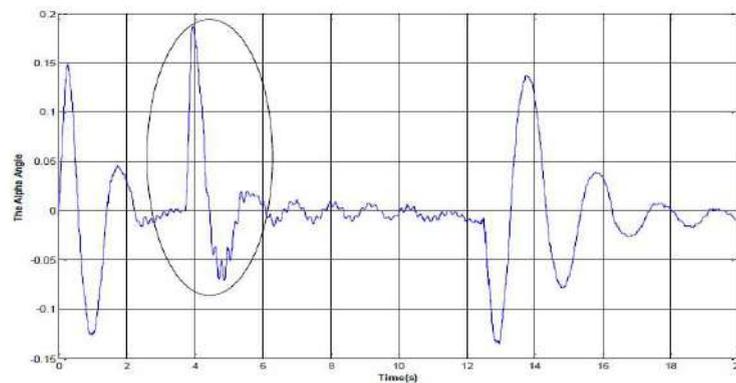


FIGURE 6.26: Balancement de l'angle α .

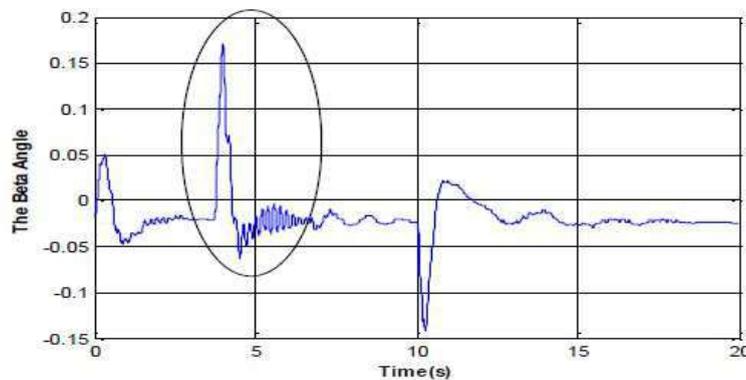


FIGURE 6.27: Balancement de l'angle β .

6.6 Conclusion

Dans ce chapitre, nous avons abordé le problème de suivi de trajectoire en minimisant les oscillations de la charge portée par le système de grue 3D, nous avons aussi réalisé des tests de robustesse des commandes développés.

Pour des raisons de simplification nous avons supposé que les trois axes du système sont découplés, de ce fait nous avons conçu un contrôleur de même structure pour chaque sous système.

Les deux nouvelles approches pour la génération et le maintien du régime de glissement dans le comportement d'un système avec des incertitudes dans sa dynamique sont implémentés sur le système de grue 3D.

Chacun de ces algorithmes de commande est conçu par deux régulateurs montés en parallèle, un régulateur PD conventionnel avec un régulateur neuro-flou pour commander le système. Deux types de systèmes neuro-flous sont utilisés, le système neuro-flou de type-1 avec une structure en cinq couches et le système neuro-flou de type-2 avec une structure en sept couches.

Avec les paramètres initiaux choisis, les résultats de poursuites de trajectoire ont montrés un bon suivi de la trajectoire de référence, un temps de réponse très petit et une erreur statique en régime permanent acceptable.

Ces résultats expérimentaux ont montré que le régime de glissement prédéfini peut être généré et maintenu dans le cas où les paramètres des deux contrôleurs NIT2FLC et ANNFLC sont réglés en utilisant les deux approches proposées.

Les tests appliqués ont montrés la robustesse des deux lois de commande vis-à-vis d'une perturbation externe et dans le cas de changement de la trajectoire de référence on utilisant les mêmes paramètres initiaux.

Par comparaison des résultats obtenus avec les algorithmes proposés et ceux obtenus avec un régulateur PD, ces derniers ont prouvé la supériorité des lois de commandes développés pour compenser des incertitudes très importantes par rapport à un régulateur PD.

On peut en conclure que ce qui est très intéressant, c'est la manière dont s'opèrent la conception, la mise au point et le réglage de ces nouveaux régulateurs qui permettent de s'adapter dans des situations critiques.

Chapitre 7

Conclusion générale et perspectives

Les systèmes sous actionnés, inspirés principalement de la structure du corps humain, admettent des degrés de liberté plus nombreux que les actionneurs. Le manque d'actionneur complique la tâche de commande de ce genre de systèmes. La commande de tels systèmes, lorsqu'on veut la mettre en pratique, se trouve confronté à la présence inévitable de perturbations, notamment celles issues des phénomènes de frottements dont la dynamique est compliquée est souvent mal modélisée.

L'objectif dans le travail présenté dans cette thèse été d'élaborer de nouvelles lois de commandes adaptatives basées sur le mode glissant et les concepts de l'intelligence artificielle qui constituent une alternative aux commandes classiques. Les nouveaux schémas de commande adaptatifs basés sur les réseaux neuro-flous ont été proposés pour des systèmes MIMO non linéaires complexes.

La plateforme expérimentale choisie pour l'implémentation des lois de commandes développées est un système de grue 3D, qui présente l'un des systèmes sous actionné avec deux angles de basculement de la charge portée selon les axes x et y non contrôlable directement.

Les contrôleurs proposés sont conçus pour assurer la stabilité et la robustesse en présence d'incertitudes structurées et non structurées et perturbation externe. Pour commencer, nous avons montré les capacités d'apprentissage et d'approximation des réseaux neuro-flous dans le cadre de la commande du robot Motoman réduit à trois degrés de liberté. La capacité d'apprentissage des RNA a été utilisée pour estimer les paramètres du système flou. La commande hybride résultante a pu reprendre la tâche d'un contrôleur PID.

L'utilisation du modèle flou de type-1 de TS et en se basant sur la théorie des modes glissant nous a permis d'exploiter le théorème de Lyapunov pour démontrer le théorème du choix des paramètres d'adaptations du système neuro-flou ainsi que la stabilité de la loi de commande proposée.

Pour la commande adaptative neuro-flou intervalle type-2, nous avons choisis le modèle de TSK A2-C0 avec l'expression de la sortie globale du système proposée par M.B.Begian, W.W.Melek et J.M.Mendel qui remplace la réduction de type, et en se basant sur la théorie des modes glissant, nous avons démontré le théorème d'adaptation des paramètres de la structure neuro-floue choisi et la stabilité du système commandé par le théorème de Lyapunov.

Le deuxième chapitre propose un bref rappel sur la logique floue qui est basée sur la recopie du savoir faire d'un opérateur humain, les systèmes flous de type-1 basés sur la logique floue de type-1 et les systèmes flous de type-2 basés sur la logique floue de type-2. Les systèmes flous de type-1 de Takagi Sugeno et ceux de type-2 sont présentés, ainsi que les trois modèles A2-C1, A2-C0 et A1-C1.

Dans le chapitre trois différentes structures de systèmes neuro-flous sont présentées, un bref aperçu sur les concepts de base des réseaux de neurones, les différentes structures d'interconnexion, et le mécanisme d'apprentissage. Une simulation est proposée à la fin de ce chapitre qui porte sur la commande en hors ligne neuro-flou d'un robot Motoman SV3. La base d'apprentissage utilisée est celle d'un régulateur PID.

Pour le passage d'une commande neuro-flou en hors ligne a une loi de commande adaptative, nous avons présenté les outils, telles que la commande adaptative et la commande par mode glissant dans le chapitre quatre, avec une application en simulation d'une loi de commande par mode glissant sur un pendule inversé.

Dans cette application nous avons présenté l'une des méthodes qui permet de réduire le phénomène de chattering, qui présente l'inconvénient majeur de cette loi de commande.

Le développement des lois de commandes adaptatives neuro-flou type-1 mode glissant et celles de la commande adaptative neuro-floue intervalle type-2 mode glissant est présenté dans le chapitre cinq. Afin de cerner la problématique énoncée nous avons exploité les algorithmes de commande adaptative neuro-floue type-1 et neuro-floue type-2 basée sur le mode glissant pour contrôler le système de grue 3D. Les résultats de l'implémentation de ces deux structures de commandes ont montrées leurs efficacités par rapport à une commande PD avec les mêmes valeurs des gains.

Les lois de commandes hybrides développés ont pus surmontés l'inconvénient des gains fixe du régulateur PD, cette avantage est obtenu par l'ajout des concepts de la commande adaptative avec les systèmes neuro-flous connus comme approximateur universelle.

Les performances des commandes proposées ont été démontrées pour le suivi de trajectoire du chariot de la grue toute en minimisant le basculement de la charge portée.

Les trajectoires de référence choisies de type rectangulaire ont été réalisées par la charge portée par la grue 3D avec une erreur inférieure à celle réalisée par le contrôleur conventionnel PD en régime permanent et ceux pour les deux algorithmes de commandes développés.

Comme test de robustesse nous avons choisi de changer la trajectoire de référence en premier lieu et d'appliquer une perturbation sur la charge durant le régime permanent. Les résultats obtenus ont montré que les deux lois de commande peuvent s'adapter pour réaliser des trajectoires sinusoïdale avec les mêmes paramètres utilisés avec la trajectoire rectangulaire avec un minimum de basculement des angles α et β .

Avec le deuxième test de robustesse, les résultats obtenus sont satisfaisants et ont montré la robustesse des lois de commandes développées.

Cette thèse contribue à la conception de stratégies de contrôle adaptatives stables utilisant des techniques intelligentes. Dans le prolongement de ces travaux, de nouvelles structures de contrôle peuvent être envisagées pour réduire davantage la complexité des outils intelligents.

En ce qui concerne les réseaux neuro-flous, cette thèse présente les stratégies d'apprentissage basées sur Lyapunov pour les structures RNA le plus populaire, le MLP.

Désormais, de nouvelles stratégies de contrôle adaptatif stables peuvent être envisagées pour d'autres architectures des RNA. De plus, de nouvelles lois d'adaptations stables peuvent également être développées pour les techniques basées sur l'optimisation par les concepts intelligentes, telles que les algorithmes génétiques et les colonies de fourmis.

Une autre perspective s'avère très prometteuse pour la commande des systèmes de grue est le développement d'une nouvelle loi de commande le backstepping.

Le développement de loi de commande prédictive à base de modèle flou appliqué sur le système de grue 3D peut aussi présenter une continuité pour le travail présenté dans cette thèse.

Bibliographie

- [1] M.Ozanane, Dictionnaire Mathématique, ou Idée Générale des Mathématique, 1961.
- [2] Nowacka-Leverton A., Michaek M., Pazderski D., Bartoszewicz A. Experimental verification of SMC with moving switching lines applied to hoisting crane vertical motion control. In : ISA Trans, Vol. 51, No. 2, pp. 682-693, 2012.
- [3] C.Vázquez, J.Collado and L.Fridman. : Super twisting control of a parametrically excited overhead crane. In : J. Frankl. Inst. Vol. 351, No. 4, pp. 2283-2298, 2014
- [4] Le Anh Tuan, Hoang Manh Cuong and Soon-Geul Lee. : Second-order sliding mode control of 3D overhead cranes, In : International Conference on Control, Automation and Information Sciences (ICCAIS), Nha Trang, pp. 341-346, 2013
- [5] C. Vázquez, L. Fridman and J. Collado. : Second Order Sliding Mode Control of an Overhead-Crane in the presence of external perturbations. In : 52nd IEEE Conference on Decision and Control, Firenze, pp. 2876-2880, 2013
- [6] S. Dai, Z. Lv, Z. Liu and x. Guan. : Study of precise positioning and antiswing for the varying rope length in 3D crane systems base on the combination of partial decoupling and fuzzy control, In : IEEE International Conference on Robotics and Biomimetics, Tianjin, pp. 656-661.2010
- [7] Cho SK, Lee HH. :A Fuzzy-Logic Antiswing Controller for Three-Dimensional Overhead Cranes. In ISA Trans. Apr ;41(2) :235-43,2002
- [8] C. Li, J. Yi and D. Zhao. : Control of the TORA system using SIRMs based type-2 fuzzy logic, In : IEEE International Conference on Fuzzy Systems, Jeju Island, pp. 694-699,2009
- [9] Jianqiang Yi, Naoyoshi Yubazaki, Kaoru Hirota. :Anti-Swing and Positioning Control of Overhead Traveling Crane, In : Information Sciences,Vol.155 pp. 19-42, 2003
- [10] Xiaolu Zhang, C. Li and J. Yi. : Prior information driven design of fuzzy logic controllers with application to the overhead crane control, In : International Conference on Fuzzy Systems and Knowledge Discovery (FSKD), Shenyang, pp. 379-38,2013
- [11] Diantong Liu, Jianqiang Yi, Dongbin Zhao, WeiWang. : Adaptive Sliding Mode Fuzzy Control for a Two-Dimensional Overhead Crane, In : Mechatronics, Vol. 15, No. 5, pp. 505-522, 2005
- [12] Mun-Soo Park. : Antisway Tracking Control of Overhead Cranes With System Uncertainty and Actuator Nonlinearity Using an Adaptive Fuzzy Sliding-Mode Control, In : IEEE Trans on Industrial Electronics, Vol. 55, NO. 11, 2008.
- [13] S.C.Duonga, E.Uezato, H.Kinjo, T.Yamamoto. : A Hybrid Evolutionary Algorithm for Recurrent Neural Network Control of a Three-Dimensional Tower Crane, In : Automation in Construction, Vol. 23, pp55-63, 2012
- [14] S. N. Sivanandam. : Introduction to Fuzzy Logic using Matlab, Springer-Verlag Berlin Heidelberg 2007

- [15] Roland S.Burns. : Advanced Control Engineering, Butterworth-Heinemann 2001
- [16] Oscar Castillo and Patricia Melin. : Type-2 Fuzzy Logic : Theory and Applications. : Springer-Verlag Berlin Heidelberg 2008.
- [17] L.A.Zadeh. : Fuzzy Sets, Information and Control, Vol. 8, pp. 338-353, 1965.
- [18] S. Sumathi, Surekha p. : Computational Intelligence Paradigms, Theory and Applications Using Matlab, Taylor and Francis Group 2010.
- [19] T. Whalen. : Superadditive strong implication in fuzzy rule-based interpolation, In : Proceedings of IEEE 5th International Fuzzy Systems, vol.3, pp. 1624-1630, 1996.
- [20] Yuetong Lin. : Modular Construction Of Fuzzy Logic Control Systems Using Wavelet Networks, these de Doctorat Université d'Arizona 2005.
- [21] N.Boumella. : Génération et Optimisation des Contrôleurs Flous Type-2, these de Doctorat université de Batna Algerie 2013.
- [22] J. M. Mendel, R. I. B. John. : Type-2 fuzzy sets made simple, In IEEE Trans on Fuzzy Systems, vol. 10, no. 2, pp. 117-127, 2002.
- [23] J. M. Mendel. : Plotting 2-1/2 D figures for general type-2 fuzzy sets by hand or by PowerPoint, In : IEEE International Conference on Fuzzy Systems, Brisbane, pp. 1-8, 2012.
- [24] N. N. Karnik and J. M. Mendel. : An introduction to type-2 fuzzy logic systems. Technical report, Univ. Southern California, 1998.
- [25] M.Manceur. : Commande Robuste des Systèmes non linéaires complexes, thèse de Doctorat université de Reims Champagne-Ardenne 2012.
- [26] J.S.Martínez. : Modélisation et Supervision des Flux Energétiques à Bord d'un Véhicule Hybride Lourd : Approche par Logique Floue de Type-2, these de Doctorat Université de Franche-Comté Belfort, France 2012.
- [27] E.H. Mamdani, S. Assilian. : An experiment in linguistic synthesis with a fuzzy logic controller, In International Journal of Man-Machine Studies, Vol. 7, No. 1, pp.1-13, 1975.
- [28] T.Takagi, M.Sugeno. : Fuzzy Identification of Systems and Its Application to Modeling and Control, In : IEEE Trans on Systems, Man, and Cybernetics, vol. SMC-15, no. 1, pp. 116-132, 1985.
- [29] E.Kayacan. : Interval Type-2 Fuzzy Logic Systems : Theory and Design, these de Doctorat Université de Bogazici 2011.
- [30] Q.Liang J.M.Mendel. : Interval Type-2 Fuzzy Logic Systems : Theory and Design, In : IEEE Trans on Fuzzy Systems, vol. 8, NO. 5, 2000.
- [31] Q.Liang, J.M.Mendel. : An introduction to type-2 TSK fuzzy logic systems, In : Fuzzy Systems Conference Proceedings, IEEE International, Vol.3, pp. 1534-1539, 1999.
- [32] N. N. Karnik and J. M. Mendel. : Type-2 fuzzy logic systems : type-reduction, In : IEEE International Conference on Systems, Man, and Cybernetics, San Diego, CA, vol.2, pp. 2046-2051, 1998.
- [33] M. B. Begian, W. W. Melek and J. M. Mendel. : Parametric design of stable type-2 TSK fuzzy systems, In : Annual Meeting of the North American Fuzzy Information Processing Society, New York City, NY, pp. 1-6, 2008.

- [34] J.M.Mendel. : Uncertain Rule-Based Fuzzy Logic Systems : Introduction and New Direction, Prentice-Hall, Upper SaddleRiver, NJ, 2001.
- [35] D.E.Rumelhart, J.L.McClelland. : Parallel distributed processing explorations in the microstructure of cognition, University of California, San Diego, and PDP Research Group, Cambridge, Mass : MIT Press,1986.
- [36] Le Cun, Y. : Modèles connexionnistes de l'apprentissage. Thèse de doctorat, Université Paris XI., 1987.
- [37] Puyin Liu, Hongxing Li. : Fuzzy Neural Network Theory and Application, In : Machine Perception and Artificial Intelligence : Vol. 59, 396.pp, Jun 2004.
- [38] P.Ponce-Cruz, F.D. Ramírez-Figueroa,. :Intelligent Control Systems with LabVIEW, Springer-Verlag London Limited 2010.
- [39] S.Haykin, C.Deng. : Classification of radar clutter using neural networks, In IEEE Trans on Neural Networks, vol. 2, no. 6, pp. 589-600, Nov 1991.
- [40] T. Sejnowski, C. Rosenberg, Parallel Networks That Learn to Pronounce English Text. : In : Complex Systems, Vol. 1, No. 1. pp 145-168, 1987.
- [41] M.Donlin, J.Child. : Is Neural Computing The Key To Artificial Intelligence?, In : Computer Design, vol 31, No 10, pp87-104, Oct, 1992.
- [42] H. P. Newquist, ?Smart Cards, ? AI Expert, pp 44-5, Jan. 1995.
- [43] R. C. Johnson. : New Techniques Rewrite Pen Computing, In : Electronic engineering Times, pp 42-44, May 22, 1995.
- [44] D.Hammerstrom. : Neural networks at work, in IEEE Spectrum, vol. 30, no. 6, pp. 26-32, June 1993.
- [45] C. M. Higgins, R. M. Goodman. : Learning fuzzy rule-based neural networks for function approximation, In : International Joint Conference on Neural Networks, Baltimore, vol.1, pp. 251-256, 1992.
- [46] S.Horikawa. : On fuzzy modeling using fuzzy neural networks with the backpropagation algorithm, In : IEEE Trans. Neural Networks, Vol. 3, No. 5, pp. 801-806. 1992.
- [47] S. R. Katayama, Y. Kajitani, K. Kuwata and Y. Nishida. : Self generating radial basis function as neuro-fuzzy model and its application to nonlinear prediction of chaotic time series, In : Second IEEE International Conference on Fuzzy Systems, San Francisco, Vol.1, pp. 407-414, 1993.
- [48] D. A. Linkens and J. Nie. : Fuzzified RBF network-based learning control : structure and self-construction, In : IEEE International Conference on Neural Networks, San Francisco, vol.2, pp. 1016-1021, 1993 .
- [49] M. A. Lee and H. Takagi. : Integrating design stage of fuzzy systems using genetic algorithms, In : Second IEEE International Conference on Fuzzy Systems, San Francisco, Vol.1, pp. 612-617, 1993 .
- [50] D.Whitley, T.Starkweather, C.Bogart. : Genetic Algorithms and Neural Networks : Optimizing Connection and Connectivity, In : Parallel Computing Vol. 14, pp. 347-361. 1990.
- [51] D.Whitley.TStarkweather. CBogartWhitley, D.,et al. : Genetic Algorithms and Neural Networks : Optimizing Connection and Connectivity, in : Parallel Computing Vol. 14, pp. 347-361. 1990.

- [52] C.Touzet. : Les Reseaux De Neurones Artificiels Introduction Au Connexionnisme, Juillet 1992.
- [53] S. Haykin, Neural Networks : A Comprehensive Foundation, Macmillan, New York, NY, 1994.
- [54] A. K. Jain, Jianchang Mao and K. M. Mohiuddin. : Artificial neural networks : a tutorial, in Computer, vol. 29, no. 3, pp. 31-44, Mar 1996.
- [55] P.Borne, M.Benrejeb, J.Haggège. : Les réseaux de neurones : présentation et applications, Edition OPHRYS, 2007.
- [56] Lutaud-Brunet, M. : Identification et contrôle de processus par réseaux neuroflous, Thèse de Doctorat de l'Université de d'Evry Val d'Essonne, 1996.
- [57] P.Picton. : Introduction to neural networks, Ed. Macmillan, 1994.
- [58] D.S. Broomhead, David Lowe and A.R. Webb. : A Sum Rule Satisfied By Optimised Feed-Forward Layered Networks, Controller HMSO, London, 1989.
- [59] Lotfi Baghli. : Contribution à la commande de la machine asynchrone, utilisation de la logique floue, des réseaux de neurones et des algorithmes génétiques, these de doctoratde l'Université Henri Poincaré, Nancy-I, 1999.
- [60] J.Shing, R.Jang. : Anfis : Adaptive-Network-Based Fuzzy Inference System, IEEE Trans On Systems, Man, And Cybernetics, Vol. 23, No. 3, 1993.
- [61] N.K.Kasabov, Q.Song. : DENFIS : Dynamic Evolving Neural-Fuzzy Inference System and Its Application for Time-Series Prediction, In : IEEE Trans On Fuzzy Systems, vol. 10, no. 2, april 2002.
- [62] Cheng-Jian Lin and Chin-Teng Lin. : An ART-based fuzzy adaptive learning control network, in IEEE Trans on Fuzzy Systems, vol. 5, no. 4, pp. 477-496, Nov 1997.
- [63] H. R. Berenji, P. Khedkar. : Learning and tuning fuzzy logic controllers through reinforcements, In : IEEE Trans on Neural Networks, vol. 3, no. 5, pp. 724-740, Sep 1992.
- [64] D. Nauck, U. Nauck, R. Kruse. : NEFCLASS for Java-new learning algorithms, In : 18th International Conference of the North American Fuzzy Information Processing Society, New York, NY, pp. 472-476, 1999.
- [65] D.Nauck, R.Kruse. : Neuro-Fuzzy Systems for Function Approximation, In : Fuzzy Set and Systems, Vol. 101, pp. 261-271, 1999.
- [66] J.S.Wang, C.S.George Lee. : Self-Adaptive Neuro-Fuzzy Inference Systems for Classification Applications, In : IEEE Trans On Fuzzy Systems, vol. 10, no. 6, December 2002.
- [67] L. Rutkowski, K. Cpalka. : Flexible Neuro-Fuzzy Systems, In : IEEE Trans On Neural Networks, vol. 14, no. 3, May 2003.
- [68] O.E.V. Dragomir. : Contribution Au Pronostic De Défaillances Par Réseau Neuro- Flou : Maitrise De L'erreur De Prédiction, Thèse De Doctorat En Automatique Université de Franche-Comté, 2008.
- [69] K.J. Åström, B. Wittenmark. : Adaptive Control, Addison-Wesley Publishing Co.,1995.
- [70] D.Landau, R.Lozano, M.M'Saad, A.Karimi. : Adaptive Control : Algorithms, Analysis and Applications, Springer-Verlag London Limited 2011.
- [71] Petros A. Ioannou, Jing Sun. : robuste apaptive control, Dover Publications, Inc. Mineola, New York 2012.

- [72] V.Utkin, J.Guldner, J.Shi. : Sliding Mode Control in Electromechanical Systems, Taylor and Francis 1999.
- [73] A Hamzaoui, N. Essounbouli, J. Zaytoon. : Fuzzy Sliding Mode Control for Uncertain SISO Systems, Processing of IFAC Conf. on Intelligent Control Systems and Signal ICONS '03, pages 233-238, 2003.
- [74] V.Bregeault. : Quelques Contributions À La Théorie De La Commande Par Modes Glissants, thèse de doctorat Ecole Centrale de Nantes 2010.
- [75] E.Slotine, W.Li. : Applied Nonlinear Control, Prentice-Hall, Inc 1991.
- [76] A.Si.Ammour. : Contribution à la commande par Mode Glissant D'ordre Fractionnaire, thèse de doctorat en automatique, université Mouloud Mammeri de Tizi-Ouzou.
- [77] J.-J.ESlotine. : Sliding Controller Design for Nonlinear Systems, In : *Int.J.Control*, Vol. 40, No. 2, 1984.
- [78] W. S. Lin and C. S. Chen. : Robust adaptive sliding mode control using fuzzy modelling for a class of uncertain MIMO nonlinear systems, In *IEE Proceedings - Control Theory and Applications*, vol. 149, no. 3, pp. 193-201, May 2002.
- [79] H. Bühler. : Réglage par mode de glissement, traité d'électricité, 1ère édition, presses, 1991.
- [80] A. G. Bondarev, S. A. Bondarev, N. E. Kostyleva, V. I. Utkin. : Sliding mode in systems with asymptotic state observers, *Autom. Remote Control*, vol. 46, pp 679-684, 1985.
- [81] A.V. Topalov, E.Kayacan, Y.Oniz, O.Kaynak. : Adaptive Neuro-Fuzzy Control with Sliding Mode Learning Algorithm : Application to Antilock Braking System, In : 7th Asian Control Conference, Hong Kong, China, 2009.
- [82] V. I. Utkin, *Sliding Modes in Control Optimization*. Springer-Verlag, 1992.
- [83] B. Benhellal, M. Hamerlain, R. Ouiguini and Y. Rahmani. : Decoupled Adaptive Neuro-Fuzzy Sliding Mode Control Applied in a 3DCrane System, In : *Jour of Electrical Engineering*, Vol. 14, No. 1, pp. 306-313, 2014.
- [84] E. Kayacan, E. Kayacan, H. Ramon and W. Saeys. : A robust on-line learning algorithm for type-2 fuzzy neural networks and its experimental evaluation on an autonomous tractor, In *IEEE Inter Conference on Systems, Man, and Cybernetics (SMC)*, pp. 1652-1657, Seoul, 2012, .
- [85] B. Benhellal, M. Hamerlain and Y. Rahmani. : Decoupled Adaptive Neuro-Interval Type-2-Fuzzy Sliding Mode Control Applied in a 3DCrane System, In : *Arab J Sci Eng*, Vol. ,No. ,pp. ,2017
- [86] Inteco Ltd., 3D crane, user's manual. Inteco Ltd., Krakow, Poland (2008).
- [87] D. Wu and J. M. Mendel. : Enhanced Karnik-Mendel Algorithms, In *IEEE Transactions on Fuzzy Systems*, vol. 17, no. 4, pp. 923-934, Aug. 2009.
- [88] A. El Ougli. : Intégration des techniques floues à la synthèse de contrôleurs adaptatifs, thèse de doctorat en automatique, Signaux et Systèmes, université Université Sidi Mohamed Ben Abdellah Fès.
- [89] P-Y. Zhang, T-S. Lu, L-B. Song. : RBF networks-based inverse kinematics of 6R manipulator, In : *Int Jour of Advanced Manufacturing Technology*, vol. 26, no. 1-2, pp. 144-147, july. 2005.

Annexe A. Algorithme de Karnil et Mendel

Calcul de y_l [32, 87]

- 1- Discrétiser l'espace de sortie Y en un nombre suffisant de point en choisissant comme segment les centroides des ensembles flous impliqués dans l'activation des règles.
- 2- Trier y_l^i dans un ordre croissant : $y_l^1 \leq y_l^2 \leq \dots \leq y_l^M$.
- 3- Initialiser f_i en prenant comme point de départ :

$$f_j = \frac{\bar{f}_j + \underline{f}_j}{2} \quad (\text{A.1})$$

avec $j=1..M$

Ensuite calculer y_l en utilisant l'équation 2.33.

Et poser $y_l' = y_l$

- 4- Trouver un point de communication k ($1 \leq k \leq M - 1$) qui satisfait $y_l^k \leq y_l \leq y_l^{k+1}$

5- poser

$$f_j = \bar{f}_j \text{ pour } j \leq k$$

et $f_j = \underline{f}_j$ pour $j \geq k + 1$

Ensuite calculer y en utilisant 2.33.

Et poser $y_l'' = y_l$

- 6- Si $y_l'' \neq y_l'$ aller à l'étape 7. Si $y_l'' = y_l'$ arrêter et poser $y_l = y_l''$

7- Poser $y_l = y_l''$ et aller à l'étape 3.

Calcul de y_r [32, 87]

- 1- Discrétiser l'espace de sortie Y en un nombre suffisant de point en choisissant comme segment les centroides des ensembles flous impliqués dans l'activation des règles.
- 2- Trier y_r^i dans un ordre croissant : $y_r^1 \leq y_r^2 \leq \dots \leq y_r^M$.
- 3- Initialiser f_i en prenant comme point de départ :

$$f_j = \frac{\bar{f}_j + \underline{f}_j}{2} \quad (\text{A.2})$$

avec $j=1..M$

Ensuite calculer y_r en utilisant l' équation 2.33.

Et poser $y'_r = y_r$

4- Trouver un point de communication $k(1 \leq k \leq M - 1)$ qui satisfait $y_r^k \leq y_r \leq y_r^{k+1}$

5- poser

$$f_j = \bar{f}_j \text{ pour } j \leq k$$

et $f_j = \underline{f}_j$ pour $j \geq k + 1$

Ensuite calculer y en utilisant 2.33.

Et poser $y''_r = y_r$

6- Si $y''_r \neq y'_r$ aller à l'étape 7. Si $y''_r = y'_r$ arrêter et poser $y_r = y''_r$

7- Poser $y_r = y''_r$ et aller à l'étape 3.

Annexe B. Modélisation du pendule inversé

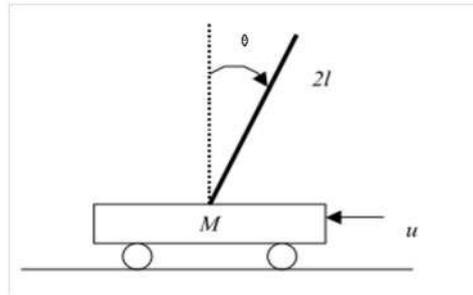


FIGURE A.1: Pendule inversé.

Le schéma de la figure présentée au dessus est celui d'un pendule de masse m et de longueur $2l$, placé sur un chariot de masse M , les équations régissant le système sont les suivantes :

$$\ddot{\theta} = f(\theta, \dot{\theta}) + g(\theta, \dot{\theta}).u + d \quad (\text{B.3})$$

$$y = \theta \quad (\text{B.4})$$

$$f(\theta, \dot{\theta}) = \frac{(m + \Delta m).l.\dot{\theta}^2.\sin(\theta).\cos(\theta) - (m + \Delta m + M + \Delta M).g.\sin(\theta)}{(m + \Delta m).l.\cos^2(\theta) - (\frac{3}{4}).l.(m + \Delta m + M + \Delta M)} \quad (\text{B.5})$$

$$g(\theta, \dot{\theta}) = \frac{-\cos(\theta)}{(m + \Delta m).l.\cos^2(\theta) - (\frac{3}{4}).l.(m + \Delta m + M + \Delta M)} \quad (\text{B.6})$$

Où θ est l'angle du pendule par rapport à la verticale, g la constante de gravité et u l'effort appliqué au chariot, Δ les incertitudes sur les masse et d les perturbations externes.

$f(\theta, \dot{\theta})$ et $g(\theta, \dot{\theta})$ sont des fonctions non linéaires bornées. la forme canonique d'un système non linéaire de second ordre peut être écrite sous la forme suivante :

$$\begin{cases} \dot{x}_1 = \dot{x} \\ \dot{x}_2 = f(X) + g(X) + d \\ y = x_1 \end{cases} \quad (\text{B.7})$$

La sortie du système $y = x_1 = x = \theta$

Les paramètres utilisés pour la simulation sont :

$M=1\text{kg}$, $m=0.1\text{kg}$, $l=0.5\text{m}$, $g=9.8\text{N/kg}$.

Les perturbations et incertitudes sont de la forme : $\Delta m = \mp 0.1m$, $\Delta M = \mp 0.1M \text{ et } d(t) = 0.1 \sin(2t)$.

L'état initial est : $[\theta, \dot{\theta}] = [0.2, 0]^T$

L'erreur de poursuite est donnée par : $e = \theta - \theta_d$

La trajectoire de référence est : $x_d = \frac{\pi}{30} \cdot \sin(t)$

Calcul de la commande :

La surface de glissement est définie comme suit : $s = \dot{e} + \lambda \cdot e$

La commande par mode glissant d'ordre un qui assure la convergence asymptotique de l'erreur vers zéro en un temps fini est donnée par la relation suivante :

$$u = u_{eq} + u_d \quad (\text{B.8})$$

u_{eq} : est la commande équivalente déterminée par la relation : $\dot{s} = 0$

u_d : est la commande discontinue donnée par $u_d = -\alpha \cdot \text{sign}(s)$.

α et λ sont choisis de telle sorte à garantir la convergence de la trajectoire vers la surface de glissement.

La loi de commande globale est donnée par :

$$u = g(x)^{-1} \cdot (\ddot{x}_d - f(x) - \lambda \cdot (\dot{x} - \dot{x}_d) - \alpha \cdot \text{sign}(s)) \quad (\text{B.9})$$

Annexe C. Opérations sur les ensembles flous type-2

Opération sur les ensembles flous type-2

Les opérations les plus importantes des ensembles flous type-2 sont présentées dans cette annexe. Soit $\tilde{A} \in X$ et $\tilde{B} \in X$ deux ensembles flous type-2 et $\mu_{\tilde{A}}(x)$, $\mu_{\tilde{B}}(x)$ leurs fonctions d'appartenance respectives (ensembles flous type-1 dans $J \subseteq [0, 1]$).

$$\mu_{\tilde{A}}(x) = \sum_i f_x(u_i)/u_i; u_i \in J \quad (\text{C.10})$$

$$\mu_{\tilde{B}}(x) = \sum_i g_x(w_i)/w_i; w_i \in J \quad (\text{C.11})$$

Barycentre d'un ensemble flou type-2 : Soit x_i un point de l'ensemble flou type-2 \tilde{A} , le barycentre de l'ensemble \tilde{A} défini sur un domaine discret de N points x_i est un ensemble flou type-1 et est donné par :

$$C_{\tilde{A}} = \int_{\theta} \cdots \int_N [\mu_{D_1}(\theta_1) * \mu_{D_2}(\theta_2) * \cdots * \mu_{D_N}(\theta_N)] / \frac{\sum_{i=1}^N x_i \theta_i}{\sum_{i=1}^N \theta_i} \quad (\text{C.12})$$

où $D_i = \mu_{A}(x_i)$ est un ensemble flou type-1 avec $\theta_i \in D_i$, les x_i sont des nombres ordinaires et $*$ dénote une t-norme. On peut noter qu'un ensemble flou type-2 peut être vu comme une collection d'ensembles flous type-1 [24]. Pour définir les opérations fondamentales : union, intersection et complément des ensembles flous \tilde{A} et \tilde{B} , on utilise le principe d'extension de Zadeh.

Énoncé du principe d'extension monodimensionnel : Le principe d'extension des ensembles flous est essentiellement une identité de base qui permet à un domaine de définition d'une relation à être étendu de quelques points de U à des sous ensembles flous de U . Particulièrement, supposons que f est une relation de U vers V et A est un ensemble flou défini par $A = \int_U \mu_A(u)/u$ cas continu ou $A = \sum_i \mu_A(u_i)/u_i$ cas discret, alors le principe d'extension assert que l'image de A sous la relation f est un ensemble flou $f(A)$ de w défini par $f(A) = f[\int_U \mu_A(u)/u] = \int_W \mu_A(u)/f(u)$ cas continu ou $f(A) = f[\sum_i \mu_A(u_i)/u_i] = \sum_i \mu_A(u_i)/f(u_i)$ cas discret. Par conséquent, l'image de A sous f peut être déduite en connaissant les images de u ou u_i sous f . Par application du principe d'extension aux ensembles flous \tilde{A} et \tilde{B} , on

aura :

Union de \tilde{A} et \tilde{B} :

$$\mu_{\tilde{A} \cup \tilde{B}}(x) = \mu_{\tilde{A}}(x) \cup \mu_{\tilde{B}}(x) = \left(\sum_i f_x(u_i)/u_i \right) \cup \left(\sum_j g_x(w_j)/w_j \right) = \sum_{i,j} (f_x(u_i) \vee g_x(w_j)) / u_i \vee w_j \quad (\text{C.13})$$

Intersection de \tilde{A} et \tilde{B} :

$$\mu_{\tilde{A} \cap \tilde{B}}(x) = \mu_{\tilde{A}}(x) \cap \mu_{\tilde{B}}(x) = \left(\sum_i f_x(u_i)/u_i \right) \cap \left(\sum_j g_x(w_j)/w_j \right) = \sum_{i,j} (f_x(u_i) \wedge g_x(w_j)) / u_i \wedge w_j \quad (\text{C.14})$$

Complément de \tilde{A}

$$\mu_{\bar{\tilde{A}}}(x) = \sum_i f_x(u_i)/(1 - u_i) \quad (\text{C.15})$$

Avec \vee et \wedge indique respectivement les opérations min et max [88].