

الجمهورية الجزائرية الديمقراطية الشعبية
République Algérienne démocratique et populaire

وزارة التعليم العالي و البحث العلمي
Ministère de l'enseignement supérieur et de la recherche scientifique

جامعة البليدة 1
Université de BLIDA 1

كلية التكنولوجيا
Faculté de Technologie

قسم الإلكترونيك
Département d'Électronique



Mémoire de Master

présenté par

MELIK Billel

&

BOUHABZA Naima

Pour l'obtention de diplôme master en électronique option automatique

Commande par logique floue de type 2 et commande neuro-floue appliquées à un pendule inversé

Proposé par : Mr : FAS Mohamed Lamine

Année Universitaire 2015-2016

Remerciements

Tout d'abord, nous remercions Allah qui nous a donné la volonté, la force, et la patience durant toutes ces années, et nous aide à terminer ce modeste travail.

Nous tenons à exprimer notre profonde gratitude à nos familles, et spécialement nos parents qui nous accompagnés tout au long de la vie, et qui nous ont toujours soutenues.

Nous adressons le grand remerciement à notre encadreur M^r M.L.FAS pour avoir d'abord proposé ce thème, pour ces conseils du début jusqu'à la fin de ce mémoire.

Nos vifs remerciements vont aussi aux membres du jury qui ont accepté de juger ce modeste travail.

Nous adressons aussi nos remerciements à tous nos amis et collègues.

Enfin nous ne pourrons terminer ces remerciements sans penser à l'ensemble de nos enseignants qui sont à l'origine de tous nos savoirs.

Merci

Dédicace

Avant tout ; nous tenons à remercier le bon dieu, et l'unique qui nous offre le courage.

Nous dédions ce modeste travail :

A nos parents la source du succès.

A nos frères.

A nos sœurs.

A toute nos familles.

A nos collègues de la promotion 2016 « Automatique ».

A nos féaux amis.

ملخص :

يعتبر الذكاء الاصطناعي من بين الأدوات القيمة في ميادين التحكم لمختلف الأنظمة خاصة منها غير خطية و متعددة المتغيرات. لقد قمنا في هذا العمل بدراسة التحكم في الناوس المقلوب بواسطة تقنيات الذكاء الاصطناعي. بعد وصف وتقديم الجهاز قمنا بدراسة تدقيقية للجهاز التي سمحت لنا بالحصول على نموذج متعدد المتغيرات وغير خطي. ثلاثة طرق لتحكم في هذا النظام تم تقديمها: متحكم غامض نوع 1، نوع 2 والعصب الغامض. كما أننا قمنا بإجراء مقارنة لهذه التقنيات الثلاثة، بعد الدراسة و المحاكاة بواسطة Matlab أظهرت النتائج فعالية المتحكم الغامض من النوع 2 بالنسبة لباقي التقنيات الأخرى .

كلمات المفاتيح : ناوس مقلوب، المنطق الغامض، العصب الغامض.

Résumé :

L'intelligence artificielle par sa définition, elle exprime le savoir-faire humain, elle est devenue actuellement un outil précieux de commande dans divers domaines.

Ce mémoire présente la commande d'un pendule inversé par des techniques d'intelligence artificielle. Après avoir décrit et présenter le pendule inversé, nous avons développé la partie modélisation de ce procédé, qui a permet d'obtenir un modèle multi-variables, non-linéaire par la représentation d'état. Après une étude théorique des techniques d'intelligence artificielle, trois structures de commande ont été présenté qui sont régulateurs flou type-1, type-2 et neuro-flou, ainsi nous avons fait une étude comparative entre ces trois techniques.

Les résultats de simulation par Matlab seront présentés, ils ont montrés l'efficacité du régulateur flou type-2 para port les autres commandes.

Mots clés : pendule inversé, logique floue, neuro-flou.

Abstract :

Artificial intelligence by definition, expresses the expertise human, it has now become a valuable tool control in various fields.

This paper presents the control of an inverted pendulum by artificial intelligence techniques. After describing and presenting the inverted pendulum, we have developed the modeling part of this process, which enables a multivariate model, non- linear by state representation. After a theoretical study of artificial intelligence techniques, three control structures have been presented that are fuzzy controllers Type-1, Type-2 and neuro-fuzzy , and we made a comparative study of these three techniques.

The simulation results will be presented by Matlab , they have shown the effectiveness of the fuzzy controller Type -2.

Keywords : inverted pendulum, fuzzy logic, neuro-fuzzy.

Listes des acronymes et abréviations

Symboles

m : masse du pendule.

b : frottement de déplacement du chariot.

M : masse du chariot.

l : demi longueur du pendule.

$F(t)$: force exercée sur le chariot.

g : intensité de pesanteur.

d : frottement du pendule.

T : énergie cinétique.

V : énergie potentielle.

\mathcal{E} : degré de liberté (dans notre cas $x(t)$ et $\theta(t)$).

D : énergie dissipé par frottement.

F_j : force généralisée dans le sens de degré de liberté

T_M : énergie cinétique de chariot

T_m : énergie cinétique de pendule

r_c : position de centre de gravité de pendule

v_c : la vitesse de centre de gravité de pendule

θ : l'angle entre la barre et le verticale.

$\dot{\theta}$: la vitesse de rotation de la barre.

x : la position du chariot.

\dot{x} : la vitesse linéaire du chariot.

Min : minimum

Max : maximum

GN : grand négatif

PN : petit négatif

ZR : zéro

PP : petit positif

GP : grand positif

Acronymes

SIMO : Single Input Multiple Output

FLC : Fuzzy Logic Controller

FLS : Fuzzy Logic System

FOU : Footprint Of Uncertainty

RN : Réseau de Neurone

ANFIS : Adaptive Neuro Fuzzy Inference System

Liste des figures

Figure 1.1 : schéma du système chariot-pendule	6
Figure 2.1 : Exemple d'un variable linguistique	15
Figure 2.2 : Schéma synoptique d'un contrôleur flou type 1	18
Figure 2.3 : Représentation schématique d'un ensemble flou type-2 intervalle	27
Figure 2.4 : Représentation schématique des appartenances secondaires	27
Figure 2.5 : la région de l'incertitude	27
Figure 2.6 : Schéma synoptique d'un contrôleur flou type 2	32
Figure 2.7 : Opérations d'entrée d'un système flou type-2 avec fuzzification type singleton	33
Figure 2.8 : Fonctions d'appartenances des entrées	35
Figure 2.9 : Fonctions d'appartenances de la sortie	35
Figure 3.1 : Neurone biologique	42
Figure 3.2 : Neurone formel	43
Figure 3.3 : Réseau de neurone non bouclé	44
Figure 3.4 : Réseau non bouclé mono couche	45
Figure 3.5 : Réseau non bouclé multi couche.	46
Figure 3.6 : Réseau de neurone bouclé	46
Figure 3.7 : Apprentissage supervisé	47
Figure 3.8 : Apprentissage non supervisé	48
Figure 3.9 : Principe du système Neuro-flou	52
Figure 3.10 : Première architecture Neuro-floue	53
Figure 3.11 : Deuxième architecture des réseaux Neuro-Flou Réalisation en série	53
Figure 3.12 : Réseau de neurones fonctionnant en aval d'un système flou	54
Figure 3.13 : Architecture d'ANFIS	55
Figure 4.1 : définition des classes de l'angle et la vitesse angulaire du pendule	63
Figure 4.2 : définition des classes de la commande	64
Figure 4.3 : résultat de simulation par la commande floue type-1 pour conditions initial [0 0 0.3 0]	66
Figure 4.4 : résultat de simulation par la commande floue type-1 pour conditions initial [0 0 0.8 0]	68

Figure 4.5 : résultat de simulation par la commande floue type-1 pour conditions initial [0 0 -0.5 0]	69
Figure 4.6 : les classes de l'angle et la vitesse angulaire du pendule pour ANFIS	71
Figure 4.7 : résultat de simulation par la commande ANFIS pour conditions initial [0 0 0.3 0]	74
Figure 4.8 : résultat de simulation par la commande ANFIS pour conditions initial [0 0 0.8 0]	76
Figure 4.9 : résultat de simulation par la commande ANFIS pour conditions initial [0 0 -0.5 0]	77
Figure 4.10 : définition des classes de l'angle et la vitesse angulaire du pendule pour le régulateur floue type-2	80
Figure 4.11 : définition des classes de la commande pour le régulateur floue type-2	80
Figure 4.12 : résultat de simulation par la commande floue type-2 pour conditions initial [0 0 0.3 0]	81
Figure 4.13 : résultat de simulation par la commande floue type-2 pour conditions initial [0 0 0.8 0]	83
Figure 4.14 : résultat de simulation par la commande floue type_2 pour conditions initial [0 0 -0.5 0]	84
Figure 4.15 : variation de l'angle θ pour [0 0 0.3 0]	85
Figure 4.16 : variation de l'angle θ pour [0 0 0.8 0]	86
Figure 4.17 : variation de l'angle θ pour [0 0 -0.5 0]	86

Liste des tableaux

Tableau 2.1 : fonctions d'appartenance de la logique floue type 1	17
Tableau 2.2 : Matrice d'inférence floue	20
Tableau 2.3 : Matrice d'inférences floue	22
Tableau 2.4 : fonctions d'appartenance type 2	29
Tableau 2.5 : Matrice d'inférences floue	36
Tableau 3.1 : les fonctions d'activation	44
Tableau 4.1 : la matrice d'inférence du contrôleur flou type-1	65
Tableau 4.2 : paramètre des fonctions d'appartenance de contrôleur ANFIS	71
Tableau 4.3 : Base de règles obtenue après la descente de gradient	73
Tableau 4.4 : matrice d'inférence pour le contrôleur flou type-2	79
Tableau 4.5 : résumé des résultats de simulation pour [0 0 0.3 0]	87
Tableau 4.6 : résumé des résultats de simulation pour [0 0 0.8 0]	87
Tableau 4.7 : résumé des résultats de simulation pour [0 0 -0.5 0]	88

Sommaire

Introduction générale	01
Chapitre 1 Modélisation et description du pendule inversé	04
1.1. Introduction	05
1.2. Description et modélisation du pendule inversé	05
1.2.1. Principe de fonctionnement	05
1.2.2. Particularités du système	06
1.2.3. Coordonnées généralisées du système	06
1.2.4. Modélisation d'un pendule inversé	07
1.2.4.1. Energie cinétique du système en mouvement	07
1.2.4.2. L'énergie potentielle du système	08
1.2.5. Equation de Lagrange	08
1.2.5.1. L'équation de Lagrange pour le degré de liberté $\mathcal{E}(t)=x(t)$	08
1.2.5.2. L'équation de Lagrange pour le degré de liberté $\mathcal{E}(t)=\theta(t)$	08
1.3. Linéarisation du modèle autour des points d'équilibres	09
1.3.1. Position d'équilibre instable ($\theta=0$)	09
1.3.2. Position d'équilibre stable ($\theta=\pi$)	10
1.4. Représentation d'état	10
1.5. Conclusion	11
Chapitre 2 Le contrôleur flou	12
2.1. Introduction	13
2.2. Historique	13
2.3. Domaines d'application	13
2.4. La logique floue type 1	14
2.4.1. Introduction	14
2.4.3. Les concepts de bases	14
2.4.2.1. L'ensemble flou	14
2.4.2.2. Univers de discours	15
2.4.2.3. Les variables et les valeurs linguistiques	15
2.4.2.4. Fonction d'appartenance	16
2.4.2.5. Opérateurs de la logique floue	17
2.4.2.6. Règles linguistiques	18
2.5. Structure d'un régulateur flou type 1	18
2.5.1. Fuzzification	18
2.5.2. Base de règle	19
2.5.3. Mécanisme d'inférence flou	20
2.5.4. Défuzzification	20
2.6. Exemple d'application	21
2.7. Logique floue type 2	25
2.7.1. Introduction	25
2.7.2. Concept de base	25
2.7.2.1. Les ensembles flous	25

2.7.3. Différents types d'ensembles flous type 2	28
2.7.4. Opération sur les ensembles flous type 2	29
2.8. Structure d'un contrôleur flou type 2	32
2.8.1. Fuzzification	32
2.8.2. Mécanisme d'inférence	32
2.8.3. Réduction de type	34
2.8.4. Défuzzification	35
2.9. Exemple d'application	35
2.10. Avantages et inconvénients du régulateur par logique floue	38
2.11. Conclusion	38
Chapitre 3 les réseaux de neurone et la commande neuro-floue	39
3.1. Introduction	40
3.2. Les réseaux de neurones	40
3.2.1. Historique	40
3.2.2. Neurone biologique	42
3.2.3. Le neurone formel	42
3.2.3.1. Définition	42
3.2.4. Architecture des réseaux de neurones formels	44
3.2.4.1. Les réseaux de neurones non bouclés (ou statiques)	44
3.2.4.2. Les réseaux de neurones bouclés (ou dynamiques)	46
3.2.4.3. Les réseaux de neurones à connexion complexes	47
3.2.5. Processus d'apprentissage	47
3.2.5.1. Types d'apprentissages	47
3.2.5.2. Les méthodes d'apprentissage	48
3.2.6. Avantages et Inconvénients des réseaux de neurones	50
3.3. Réseau neuro-flou	51
3.3.1. La combinaison des réseaux neuronaux et de la logique floue	51
3.3.2. Définition	52
3.3.3. Principe de fonctionnement	52
3.3.4. Architectures neuro-floues	52
3.3.4.1. Première architecture	52
3.3.4.2. Deuxième architecture	53
3.3.4.3. Troisième architecture	54
3.3.5. Commande neuro-flou adaptative (Adaptative Network Fuzzy Inference System)	54
3.4. Exemple d'application	57
3.5. Conclusion	60
Chapitre 4 Application à la commande d'un pendule inversé	61
4.1. Introduction	62
4.1.1. Analyse de problème	62
4.2. Application de la commande floue type_1 sur le pendule inversé	62
4.2.1. Enoncé de la méthode	62
4.2.2. Choix des éléments des régulateurs flous	62
4.2.2.1. Choix de la forme des classes	62
4.2.2.2. Définition des classes pour les deux variables d'entrées	63
4.2.2.3. Définition des classes de sortie	64
4.2.2.4. Matrice d'inférence	64

4.2.2.5. Mécanisme (moteur) d'inférence	65
4.2.2.6. La defuzzification	65
4.2.3. Les résultats de simulation	65
4.2.4. Interprétation des résultats de simulation dans le cas de FLC type	70
4.3. Application de la commande Neuro-floue sur le pendule inversé	70
4.3.1. Fonction d'appartenance	70
4.3.2. Algorithme d'apprentissage	72
4.3.2.1. Les différentes étapes de fonctionnement	72
4.3.3. Les résultats de simulation	73
4.3.4. Interprétation des résultats de simulation	78
4.4. Application de la commande floue type 2 sur le pendule inversé	78
4.4.1. Enoncé de la méthode	78
4.4.1.1. Choix des éléments de régulateur flou type-2	78
4.4.1.2. Matrice d'inférence	79
4.4.1.3. Les fonctions d'appartenance des entrées et de la sortie	80
4.4.2. Les résultats de simulation	81
4.5. Comparaison des résultats obtenus	85
4.6. Conclusion	89
Conclusion Générale	90
Bibliographie	92

Introduction Générale

Les méthodes classiques de l'automatique ont été largement appliquées dans de nombreux problèmes de régulation industrielle. Cependant, la plupart des systèmes physiques présentent des non-linéarités et leurs paramètres sont souvent mal connus et/ou variables dans le temps. Pour la commande de telles classes de systèmes, les méthodes conventionnelles de l'automatique ont montré leurs limites en termes de stabilisation et performances. Suite aux développements des calculateurs numériques, les automaticiens commencent à s'intéresser aux nouvelles approches de commande telles que la commande adaptative, la commande prédictive, la commande robuste, ainsi que les techniques basées sur l'intelligence artificielle. Parmi ces dernières, la commande par logique floue. Cette commande est à l'heure actuelle une des préoccupations des chercheurs dans le monde.

Les systèmes d'inférence flous possèdent deux points forts, le premier est qu'ils sont généralement construits à partir de la connaissance humaine, et le deuxième réside dans leur capacité descriptive due à l'utilisation des variables linguistique.

Depuis la première application du formalisme de la logique floue à la commande des systèmes proposée par Mamdani, plusieurs travaux ont montré que le contrôle à logique floue est une méthode adéquate pour la commande des procédés mal définis ou complètement inconnus et qui ne peuvent pas être modélisés mathématiquement d'une manière précise.

La logique floue classique appelée aujourd'hui logique floue type-1 a été généralisée vers une nouvelle logique floue appelée logique floue type-2. Ces dernières années, Mendel et ses collègues ont beaucoup travaillé sur cette nouvelle logique et ils ont bâtis son fondement théorique.

La logique floue type-2 est très efficace dans les circonstances où il est très difficile de déterminer des fonctions d'appartenance exactes pour un système flou, par conséquent, cette nouvelle logique va nous permettre d'incorporer des incertitudes dans les règles, ce qui va agir positivement sur la sortie du système.

Il y a au moins trois sources d'incertitudes dans les systèmes flous type-1 :

- Le sens des mots utilisés dans les règles peut être incertain, c'est-à-dire qu'un mot veut dire des choses différentes à des gens différents.
- Les mesures activant un système flou type-1 peuvent être bruitées, ce qui introduit des incertitudes.
- Les données utilisées pour l'ajustement des paramètres d'un système flou type-1 peuvent être aussi bruitées.

Toutes ces incertitudes mènent à obtenir des incertitudes au niveau des fonctions d'appartenance floues. Les ensembles flous type-1 ne sont donc pas capables de modéliser directement de telles incertitudes, parce que leurs fonctions d'appartenance floues sont définies d'une manière mathématique précise. De l'autre côté, les ensembles flous type-2 sont capables de modéliser de telles incertitudes parce que leurs fonctions d'appartenance sont elles-mêmes floues.

Les fonctions d'appartenance type-1 sont bidimensionnelles, par contre, les fonctions d'appartenance floues type-2 sont tridimensionnelles. La nouvelle (troisième) dimension des ensembles flous type-2 fournit un degré de liberté supplémentaire permettant de prendre en charge la modélisation des incertitudes.

La commande par logique floue est très utilisée dans des cas d'études particuliers, qui sont représentatifs de grandes classes d'applications. De plus, avec l'expérience, la connaissance de ces cas s'est affinée et ils fournissent aujourd'hui une base idéale pour comparer de façon valable les avantages et les inconvénients d'approches différentes. Le pendule inversé est l'un de ces cas typique qui occupe une place importante dans l'industrie comme un moyen de transport par exemple.

La description du pendule inversé muni de son actionneur et ses capteurs ainsi que les divers phénomènes physiques présents lors du fonctionnement, montrent la forte complexité due aux nombreuses non linéarités ainsi que la difficulté à modéliser parfaitement la dynamique du système pendule inverse-capteurs-actionneurs, le contrôle du pendule inversé devient difficile, car la connaissance du système se révèle imprécise et imparfaite.

Notre travail est appliqué à un système pendule inversé, qui est un système non linéaire, instable, SIMO (Single Input Multiple Output) ; sous actionné à plus d'un degré de liberté, très sensible aux retards et aux frottements.

Ce mémoire est organisé comme suit :

Chapitre I :

Ce chapitre est consacré à la description du système, nous élaborons le modèle mathématique d'un pendule inversé on basant sur le formalisme d'Euler-Lagrange suivi d'une linéarisation autour de deux points d'équilibre possible. Ensuite, nous présenterons le système dans l'espace d'état.

Chapitre II :

Dans le deuxième chapitre nous étudions une technique d'intelligence artificielles qui est la logique floue type 1 et type 2, nous présentons toutes les notions et les définitions nécessaire pour comprendre cette technique, ainsi nous définirons les contrôleurs flous, ensuite un exemple explicatif est présenté.

Chapitre III :

Dans le troisième chapitre nous étudions la structure multicouche du réseau de neurones en tenant compte des différentes architectures. Nous nous concentrons en particulier sur la projection des systèmes flous dans un réseau de neurone afin de former un système neuro-flou en tirant profit de la logique floue et de réseaux de neurones.

Chapitre IV :

Dans ce chapitre, nous appliquons ces différentes méthodes déjà citées sur la commande de pendule inversé, en faisant une étude comparative.

Enfin, nous terminons notre travail par une conclusion générale résumant les différents résultats obtenus et quelques perspectives envisagées.

Chapitre 1

Modélisation et description du pendule inversé

I.1. Introduction	5
I.2. Description et modélisation du pendule inversé	5
I.3. Linéarisation du modèle autour des points d'équilibres	9
I.4. Représentation d'état	10
I.5. Conclusion	11

I.1. Introduction

Nous présentons dans ce chapitre en premier lieu, un modèle non linéaire d'un système classique souvent utilisé en automatique, plus particulièrement dans les applications de commande : le pendule inversé qui est une plate-forme d'essai qui pose un problème d'instabilité à l'angle $\theta = 0$, il nous servira notamment, plus loin dans les chapitres suivants pour les tests des différentes commandes synthétisées.

Par la suite, nous développerons un modèle dynamique qui sera présenté sous forme d'équations différentielles déduites à partir du formalisme d'Euler Lagrange qui constitue une approche systématique simple à mettre en œuvre.

I.2. Description et modélisation du pendule inversé

Le pendule inversé est un système instable, SIMO (Single Input Multiple Output) et possède des non-linéarités non négligeables dues à sa structure dynamique et aux forces de friction. C'est un outil didactique et un problème classique utilisé en automatique c'est pourquoi il est souvent utilisé pour tester les performances et la robustesse de nouvelles lois de commande. [19]

Le pendule inversé étudié ici est constitué d'un chariot mobile en translation sur un axe horizontal de longueur de deux mètres. Le pendule, tout en étant fixé sur le chariot, est libre en rotation.

En exerçant une force horizontale $F(t)$ sur le chariot délivrée par un moteur électrique à courant continu effectué un signal variant entre 0 v et +5 v, une translation de X mètres de celui-ci est obtenue ainsi qu'une rotation de θ radians du pendule.

Le pendule de longueur $2*l$ et de masse m dont l'axe de suspension est lié au chariot de masse M , peut se déplacer sur un axe horizontal. Initialement le système est au repos, les frottements dus au déplacement du chariot sont notés b , les frottements situés au niveau de l'axe de rotation notés d .

I.2.1. Principe de fonctionnement

Le principe de fonctionnement du système est très simple en théorie : quand le pendule penche vers la gauche, le chariot doit le rattraper en effectuant un mouvement vers la gauche. Et quand le pendule penche vers la droite, le chariot doit le rattraper en effectuant un mouvement vers la droite, jusqu'à la stabilisation du pendule à l'angle $\theta = 0^\circ$.

La difficulté c'est de régler l'intensité et la forme de la réaction du chariot en fonction de l'angle que le pendule fait avec la verticale.

I.2.2. Particularités du système

- Instable.
- Non-linéaire.
- Modèle de connaissance incomplet.
- Un seul actionneur (sous actionné).
- Deux grandeurs à asservir (multi variables).

I.2.3. Coordonnées généralisées du système

L'ensemble du chariot pendule à deux degrés de liberté qui sont représentées par deux coordonnées généralisées, x pour le déplacement horizontal du chariot, θ pour la rotation du pendule. La direction positive de x est le sens à droite en mètre et celui de l'angle est le sens aiguilles d'une montre en radian. [10]

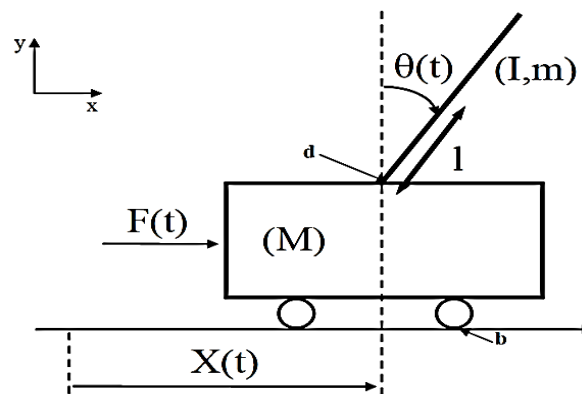


Figure 1.1 : schéma du système chariot-pendule

Soit :

m : masse du pendule.

b : frottement de déplacement du chariot.

M : masse du chariot.

$x(t)$: position du chariot.

l : demi longueur du pendule.

$\theta(t)$: l'angle du pendule.

$F(t)$: force exercée sur le chariot.

g : intensité de pesanteur.

d : frottement du pendule.

Les variables et les paramètres du système chariot pendule sont [10]:

M	m	g	l	I	b	d
2.4kg	0.23kg	9.81m/s ²	0.36m	0.099kg.m ²	0.05 Ns/m	0.005 Nms/rad

Tableau 1.1 : Les paramètres du système du pendule inversé

La méthode utilisée pour déterminer le modèle est celle des équations du mouvement données par les équations d'Euler-Lagrange.

I.2.4. Modélisation d'un pendule inversé

En utilisant la méthode de Lagrange qui est définie par l'équation suivante :

$$L = T - V$$

$$\frac{d}{dt} \left(\frac{\partial L}{\partial \dot{\epsilon}_j} \right) - \frac{\partial L}{\partial \epsilon_j} + \left(\frac{\partial D}{\partial \dot{\epsilon}_j} \right) = F_j \quad (1.1)$$

Tel que :

T : énergie cinétique.

V : énergie potentielle.

\mathcal{E} : degré de liberté (dans notre cas $x(t)$ et $\theta(t)$).

D : énergie dissipé par frottement.

F_j : force généralisée dans le sens de degré de liberté. [11]

I.2.4.1. Énergie cinétique du système en mouvement

✓ L'énergie cinétique du chariot :

$$T_M = \frac{1}{2} M \dot{x}^2 \quad (1.2)$$

✓ L'énergie cinétique du pendule :

$$T_m = \frac{1}{2} m v_c \times v_c + \frac{1}{2} I \omega^2 \quad (1.3)$$

La position de centre de gravité de pendule à partir de ces coordonnées :

$$r_c = (x + l \sin \theta) \hat{i} + l \cos \theta \hat{j} \quad (1.4)$$

D'où la vitesse de centre de gravité est donnée par :

$$V_c = \frac{dr_c}{dt} = (\dot{x} + l \cos \theta \dot{\theta}) \hat{i} - (l \sin \theta \dot{\theta}) \hat{j} \quad (1.5)$$

En remplace (1.5) dans (1.3) on obtient :

$$T_m = \frac{1}{2} m (\dot{x}^2 + 2 \dot{x} l \cos \theta \dot{\theta} + l^2 \cos^2 \theta \dot{\theta}^2 + l^2 \sin^2 \theta \dot{\theta}^2) + \frac{1}{2} I \dot{\theta}^2 \quad (1.6)$$

L'énergie cinétique du pendule devient alors :

$$T_m = \frac{1}{2} m (\dot{x}^2 + 2 \dot{x} l \cos \theta \dot{\theta} + l^2 \dot{\theta}^2) + \frac{1}{2} I \dot{\theta}^2 \quad (1.7)$$

Et l'énergie cinétique de l'ensemble chariot et pendule est exprimée par :

$$T = T_M + T_m = \frac{1}{2} M \dot{x}^2 + \frac{1}{2} m (\dot{x}^2 + 2 \dot{x} l \cos\theta \dot{\theta} + l^2 \dot{\theta}^2) + \frac{1}{2} I \dot{\theta}^2 \quad (1.8)$$

I.2.4.2. L'énergie potentielle du système

L'énergie potentielle de centre de gravité de la barre est donnée par :

$$V = m g l \cos\theta \quad (1.9)$$

I.2.5. Equation de Lagrange

$$\frac{d}{dt} \left(\frac{\partial L}{\partial \dot{\epsilon}_j} \right) - \frac{\partial L}{\partial \epsilon_j} + \left(\frac{\partial D}{\partial \dot{\epsilon}_j} \right) = F_j \quad (1.10)$$

Le Lagrangien de système en utilisant (1.8) et (1.9) : $L = T - V$

$$L = \frac{1}{2} M \dot{x}^2 + \frac{1}{2} m (\dot{x}^2 + 2 \dot{x} l \cos\theta \dot{\theta} + l^2 \dot{\theta}^2) + \frac{1}{2} I \dot{\theta}^2 - m g l \cos\theta \quad (1.11)$$

I.2.5.1. L'équation de Lagrange pour le degré de liberté $\mathcal{E}(t) = x(t)$

$$D = \frac{1}{2} b \dot{x}^2 \quad (1.12)$$

$$\frac{d}{dt} \left(\frac{\partial L}{\partial \dot{x}} \right) - \frac{\partial L}{\partial x} = F - b \dot{x} \quad (1.13)$$

La première équation de Lagrange :

$$(M + m) \ddot{x} + m l \cos\theta \ddot{\theta} - m l \sin\theta \dot{\theta}^2 = F - b \dot{x} \quad (1.14)$$

I.2.5.2. L'équation de Lagrange pour le degré de liberté $\mathcal{E}(t) = \theta(t)$

$$\frac{d}{dt} \left(\frac{\partial L}{\partial \dot{\theta}} \right) - \frac{\partial L}{\partial \theta} = -d \dot{\theta} \quad (1.15)$$

La deuxième équation de Lagrange :

$$(m l^2 + I) \ddot{\theta} + m l \ddot{x} \cos\theta + m l \dot{x} \sin\theta \dot{\theta} - m l \dot{x} \sin\theta \dot{\theta} - m g l \sin\theta = -d \dot{\theta} \quad (1.16)$$

Le modèle de connaissance du système chariot-pendule est donné par le système d'équation suivant :

$$\begin{cases} (M + m) \ddot{x} + b \dot{x} + m l \cos\theta \ddot{\theta} - m l \sin\theta \dot{\theta}^2 = F \\ m l \ddot{x} \cos\theta + (m l^2 + I) \ddot{\theta} + d \dot{\theta} - m g l \sin\theta = 0 \end{cases} \quad (1.17)$$

I.3. Linéarisation du modèle autour des points d'équilibres

Il existe deux points d'équilibre, la première instable ($\theta = 0$) et l'autre stable ($\theta = \pi$), on fait la linéarisation des équations différentielles obtenues autour de ces points d'équilibres.

Pour des petites variations de θ autour de point d'équilibre θ_0 . [11]

$$\begin{cases} \theta = \theta_0 + \varepsilon \\ \dot{\theta} = \dot{\varepsilon} \end{cases}$$

Le développement en série de Taylor du premier ordre d'une fonction de θ est donné par :

$$f(\theta) \approx f(\theta_0) + \varepsilon \frac{df}{d\theta}(\theta_0)$$

Les termes du haut ordre sont négligés : $\dot{\varepsilon}^2 \approx 0$

I.3.1. Position d'équilibre instable ($\theta = 0$)

Ce point d'équilibre instable correspondre à la position verticale de la barre.

$$\text{Pour } \theta_0 = 0 \text{ on a : } \begin{cases} \cos\theta \approx \cos(0) + \theta[-\sin(0)] = 1 \\ \sin\theta \approx \sin(0) + \theta[\cos(0)] = \theta \\ \dot{\theta}^2 = 0 \end{cases}$$

En substituant ces linéarisations dans le système d'équation (1.17) et en négligeant les termes du haut ordre, on trouve le système d'équation linéarisé suivant :

$$\begin{cases} (M + m)\ddot{x} + b\dot{x} + m l \ddot{\theta} = F \\ ml\ddot{x} + (ml^2 + I)\ddot{\theta} + d\dot{\theta} + m g l \theta = 0 \end{cases} \quad (1.18)$$

En appliquant la transformée de Laplace sur le système d'équation, on trouve :

$$\begin{cases} (M + m) s^2 X(s) + b s X(s) + m l s^2 \Theta(s) = F(s) \\ m l s^2 X(s) + (m l^2 + I) s^2 \Theta(s) + d \Theta(s) - m g l \Theta(s) = 0 \end{cases} \quad (1.19)$$

En substituant pour éliminer $X(s)$ ou $\Theta(s)$ dans (1.19), on trouve les deux fonctions de transfert suivantes :

$$G_1(S) = \frac{X(S)}{F(S)} = \frac{(ml^2 + I)s^2 + ds - mgl}{((M+m)(ml^2 + I) - m^2 l^2)s^4 + [(M+m)d + (ml^2 + I)b]s^3 + [-(M+m)mgl + db]s^2 - mglbs}$$

$$G_2(S) = \frac{\Theta(S)}{F(S)} = \frac{-mls}{((M+m)(ml^2 + I) - m^2 l^2)s^3 + [(M+m)d + (ml^2 + I)b]s^2 + [-(M+m)mgl + db]s - mglb}$$

I.3.2. Position d'équilibre stable ($\theta = \pi$)

$$\text{Pour } \theta_0 = \pi \text{ on a : } \begin{cases} \cos\theta \approx \cos(\pi) + (\pi - \theta)[- \sin(\pi)] = -1 \\ \sin\theta \approx \sin(\pi) + (\pi - \theta)[\cos(\pi)] = -\theta \\ \dot{\theta}^2 = 0 \end{cases}$$

En substituant ces linéarisations dans le système d'équation (1.17) et en négligeant les termes du haut ordre, on trouve le système d'équation linéarisé suivant :

$$\begin{cases} (M + m)\ddot{x} + b\dot{x} + m l \ddot{\theta} = F \\ -m l \ddot{x} + (m l^2 + I)\ddot{\theta} + d \dot{\theta} + m g l \theta = 0 \end{cases} \quad (1.20)$$

En appliquant la transformée de Laplace sur le système d'équation, on trouve :

$$\begin{cases} (M + m) s^2 X(s) + b s X(s) + m l s^2 \Theta(s) = F(s) \\ -m l s^2 X(s) + (m l^2 + I) s^2 \Theta(s) + d s \Theta(s) + m g l \Theta(s) = 0 \end{cases} \quad (1.21)$$

En substituant pour éliminer $X(s)$ ou $\Theta(s)$ dans (1.21), on trouve les deux fonctions de transfert suivantes :

$$G_1(S) = \frac{X(S)}{F(S)} = \frac{(m l^2 + I) s^2 + d s + m g l}{((M+m)(m l^2 + I) - m^2 l^2) s^4 + [(M+m) d + (m l^2 + I) b] s^3 + [(M+m) m g l + d b] s^2 + m g l b}$$

$$G_2(S) = \frac{\Theta(S)}{F(S)} = \frac{m l s}{((M+m)(m l^2 + I) - m^2 l^2) s^3 + [(M+m) d + (m l^2 + I) b] s^2 + [(M+m) m g l + d b] s + m g l b}$$

I.4. Représentation d'état

La représentation d'état des systèmes linéaires est de la forme suivante :

$$\begin{cases} \dot{x} = Ax + BF \\ y = Cx \end{cases}$$

On pose le vecteur d'état : $x = [x_1 \ x_2 \ x_3 \ x_4]^T = [x \ \dot{x} \ \theta \ \dot{\theta}]^T$

θ : l'angle entre la barre et le verticale.

$\dot{\theta}$: la vitesse de rotation de la barre.

x : la position du chariot.

\dot{x} : la vitesse linéaire du chariot.

Tel que $\dot{x}_1 = x_2$ et $\dot{x}_3 = x_4$

$$\dot{x}_1 = x_2$$

$$\dot{x}_2 = \ddot{x} = \frac{-b(ml^2+I)-m^2 l^2 g \sin x_3 \cos x_3 + m l d \cos x_3 + m l \sin x_3 (m l^2 + I) x_4^2 + F(ml^2+I)}{(ml^2+I)(M+m) - m^2 l^2 \cos^2 x_3}$$

$$\dot{x}_3 = x_4$$

$$\dot{x}_4 = \ddot{\theta} = \frac{-m l \cos x_3 F + m l b \cos x_3 x_2 - m^2 l^2 \sin x_2 \cos x_2 x_4^2 - d(M+m) x_4 + m g l \sin x_3 (M+m)}{(ml^2+I)(M+m) - m^2 l^2 \cos^2 x_3}$$

Notre travail consiste à réaliser des contrôleurs en appliquant les équations obtenues de la représentation d'état qui représentent le système étudié.

- La commande de ce système doit réaliser :
 - ✓ La régulation sur l'angle θ , en partant d'une condition initiale comprise dans l'intervalle $[-1, +1]$
 - ✓ La régulation et la poursuite sur la position X , en partant d'une condition initiale comprise dans l'intervalle $[-1m, +1m]$.

I.5. Conclusion

Dans ce chapitre nous avons présenté le système sur lequel nous avons travaillé, c'est-à-dire le pendule inversé, ainsi que nous avons fait une modélisation de ce procédé pour obtenir des équations qui représentent ce dernier.

L'objectif de notre travail étant de commander le pendule inversé à l'aide des différentes méthodes heuristiques.

Nous allons maintenant introduire une méthode de commande par la logique floue type 1 et type 2 qui font l'objet du chapitre suivant.

Chapitre 2

Le contrôleur flou

II.1. Introduction	13
II.2. Historique	13
II.3. Domaines d'application	13
II.4. La logique floue type1	14
II.5. Structure d'un régulateur flou type 1	18
II.6. Exemple d'application	21
II.7. Logique floue type 2	25
II.8. Structure d'un contrôleur flou type 2	32
II.9. Exemple d'application	35
II.10. Avantages et inconvénients du régulateur par logique flou	38
II.11. Conclusion	38

II.1. Introduction

Dans ce chapitre, nous allons présenter une nouvelle méthode de commande qui est souvent utilisée et en cours de développement, il existe beaucoup de recherches sur cette technologie qui peuvent changer le monde de l'automatique, on parle de la logique floue de type 1 et type 2.

La commande floue est le domaine d'application de la logique floue le plus utilisé dans le cadre de la commande des systèmes non linéaires complexes. L'approche utilisée repose sur la propriété d'approximation universelle des systèmes flous. Cette approche de commande floue a pour objectif le traitement des problèmes de commande de processus qui sont, soit difficilement automatisables par une approche classique, soit les sources d'information sont jugées non précises ou incertaines, et cela à partir des connaissances des experts ou d'opérateurs qualifiés travaillant sur le processus. La particularité de cette commande est de reproduire le comportement humain plutôt que de réaliser un modèle mathématique du système, et le contrôleur flou peut ainsi être vu comme un algorithme qui peut convertir une stratégie formelle de commande basée sur les connaissances d'un expert en une stratégie automatique de commande. Cet algorithme de commande se base sur une collection de règles floues appelée base de règles. L'ensemble des règles de la commande sont reliées par les concepts d'implication, de composition floue et des règles d'inférence floue.

II.2. Historique [22] [27]

- 1965, naissance du concept flou avec le Pr. Zedeh Lofti (Californie) « Un contrôleur électromécanique doté d'un raisonnement humain serait plus performant qu'un contrôleur classique. » Théorie des « sous-ensembles flous ».
- 1973, Zadeh introduit la notion de variables linguistiques.
- 1974, Mamdani (Londres) réalise un contrôleur flou pour moteur à vapeur.
- 1987, explosion du flou au Japon et qui atteint son apogée en 1990.
- 1995 J. S. R. Jang Logique floue élargie aux systèmes à réseaux de neurones et l'Intelligence Artificielle.

II.3. Domaines d'application

Les domaines d'applications pratiques déjà connues sont innombrables, nous en citons quelques-uns [28]:

- l'intelligence artificielle.
- la robotique.

- le contrôle des machines.
- l'informatique.
- le traitement des données.
- la communication homme-machine.
- la reconnaissance des formes.
- la médecine.
- les transports.

II.4. La logique floue type1

II.4.1. Introduction

La théorie de la logique floue « fuzzy logic FL» a été développée par Lotfi Zadeh en 1960. C'est une méthodologie de calcul basée non pas sur des valeurs numériques mais sur des variables linguistiques prenant des valeurs linguistiques appartenant au langage humain.

Cette théorie a des applications dans plusieurs domaines les plus populaires sont dans le domaine automatique pour la commande est la modélisation des systèmes. Le premier contrôleur à la logique floue a été proposé par Mamdani en 1974, mais l'essor véritable de la commande à la logique floue commencé au Japon en début des années 1980 travaux de Takagi Sugeno et Tongin. [9]

II.4.2. Les concepts de bases

II.4.2.1. L'ensemble flou

La théorie des ensembles flous est une théorie mathématique dont l'objectif principal est la modélisation des notions vagues et incertaines du langage naturel. Cette théorie permet d'exprimer l'idée d'une appartenance partielle d'un élément à un ensemble [10].

En théorie classique des ensembles, un sous ensemble A de B est défini par une fonction d'appartenance $\mu_A(x)$ qui caractérise tout élément x appartenant à B . cette fonction prend la valeur 1 si x appartient à A et la valeur 0 dans le cas contraire.[19]

$$\mu_A(x) = \begin{cases} 1 & \text{si } x \in A \\ 0 & \text{si } x \notin A \end{cases} \quad (2.1)$$

En logique floue, un sous ensemble A de B est défini par une fonction d'appartenance $\mu_A(x)$ pouvant prendre différentes valeurs comprise entre 0 et 1, selon le degré d'appartenance de l'élément x au sous-ensemble A .

$$\mu_A(x) \in [0 \ 1] \quad (2.2)$$

II.4.2.2. Univers de discours

C'est l'ensemble des valeurs réelles (l'intervalle numérique) qui peut prendre la variable floue, en général l'univers de discours est noté par des lettres en majuscule « U, W, \dots ».

Les valeurs linguistiques seront alors projetées dans l'univers de discours pour définir le sous-ensemble associé à chaque valeur linguistique. [20]

II.4.2.3. Les variables et les valeurs linguistiques

La logique floue se repose sur des variables floues dites variables linguistiques à valeurs linguistiques dans l'univers de discours U .

Une variable linguistique représente un état dans le système à régler ou une variable de réglage dans un contrôleur flou. Chaque valeur constitue un ensemble flou de l'univers de discours.

Une variable linguistique est généralement caractérisée par [16] :

- Un nom de la variable linguistique.
- Un ensemble des valeurs linguistique qui peut prendre la variable linguistique.
- L'univers de discours, c'est-à-dire l'ensemble des valeurs numérique sur lesquelles est défini.

Exemple :

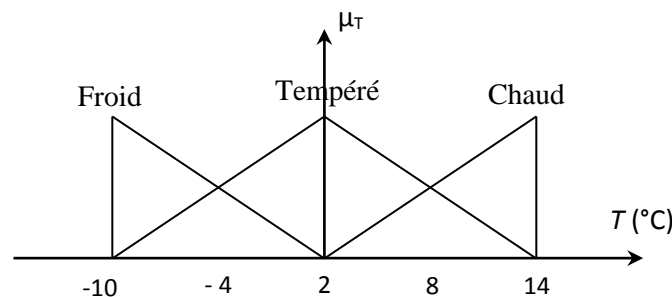


Figure 2.1 : Exemple d'un variable linguistique.

- Univers de discours : Gamme de température de -10°C à 14°C .
- Variable linguistique : La température.
- Valeurs linguistiques : « chaud » « tempéré » « froid ».

II.4.2.4. Fonction d'appartenance

Il s'agit d'établir une relation entre le degré de vérité de la variable floue et la grandeur d'entrée correspondante, les sous-ensembles flous sont ceux caractérisés par une fonction d'appartenance notée $\mu_A(x)$ définie sur $[0,1]$.

Les fonctions d'appartenance les plus utilisées sont représentées dans le tableau suivant : [2] [12]

Fonction	Forme algébrique	Forme graphique
Fonction triangulaire	$\mu_A(x) = \begin{cases} 0 & x < a \\ \frac{x-a}{b-a} & a \leq x < b \\ \frac{c-x}{c-b} & b \leq x \leq c \\ 0 & x > c \end{cases}$	
Fonction trapézoïdale	$\mu_A(x) = \begin{cases} 0 & x < a \\ \frac{x-a}{b-a} & a \leq x < b \\ 1 & b \leq x < c \\ \frac{d-x}{d-c} & c \leq x \leq d \\ 0 & x > d \end{cases}$	
Fonction gaussien	$\mu_A(x) = \exp\left(-\frac{1}{2}\left(\frac{x-m}{\sigma}\right)^2\right)$	

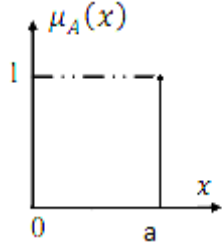
Fonction singleton	$\mu_A(x) \begin{cases} 1 & x = a \\ 0 & x \neq a \end{cases}$	
--------------------	--	---

Tableau 2.1 : fonctions d'appartenance de la logique floue type 1

Le choix des formes des fonctions d'appartenance est arbitraire. Des études comparatives ont montré qu'avec les différentes formes des fonctions d'appartenance, les résultats sont pratiquement similaires en boucle fermée. La forme la plus fréquemment utilisée en commande floue est la forme triangulaire. Le nombre de fonctions d'appartenance est généralement impair et se répartissent autour de zéro. En général, on introduit pour une variable linguistique trois, cinq ou sept ensembles flous. Le choix du nombre dépend de la précision souhaitée. Les fonctions d'appartenance peuvent être symétriques, non symétriques et équidistantes ou non équidistantes.[26]

II.4.2.5. Opérateurs de la logique floue

Soit de ensemble flou A et B définies sur des univers de discours V et W , avec une fonction d'appartenance μ_A et μ_B respectivement : [10]

- **Union de l'ensemble flou (Union flou)**

La fonction d'appartenance d'union μ_A et μ_B est définie par :

$$\mu_{A \cup B} = \max(\mu_A, \mu_B) \tag{2.3}$$

- **L'intersection floue**

La fonction d'appartenance de l'intersection μ_A et μ_B est donnée par :

$$\mu_{A \cap B} = \min(\mu_A, \mu_B) \tag{2.4}$$

- **Le complément flou**

Le complément \bar{A} de A de l'ensemble flou incluse dans V avec la fonction d'appartenance définie par :

$$\mu_{\bar{A}} = 1 - \mu_A \tag{2.5}$$

II.4.2.6. Règles linguistiques

L'idée principale des systèmes basés sur la logique floue est d'exprimer la connaissance humaine sous la forme de règles linguistiques de forme si, alors. Chaque règle à deux parties [9]:

1. Partie antécédente (prémisse ou condition), exprimée par Si...
2. La partie conséquente (conclusion), exprimée par alors...
 - La Partie antécédente est la description de l'état du système.
 - La partie conséquente exprime l'action que l'opérateur qui contrôle le système doit exécuter. Chaque règle floue est basée sur l'implication floue.

Zedah a été le premier à introduire la notion de règle floue sous la forme :

Règle : Si x est A alors y est B .

II.5. Structure d'un régulateur flou type 1

Il existe trois modules pour un régulateur à logique floue [2] :

1. La fuzzification des entrées et sorties
2. L'inférence floue selon une base de règle
3. La défuzzification des sorties.

L'architecture d'un contrôleur flou est donnée par le schéma suivant :

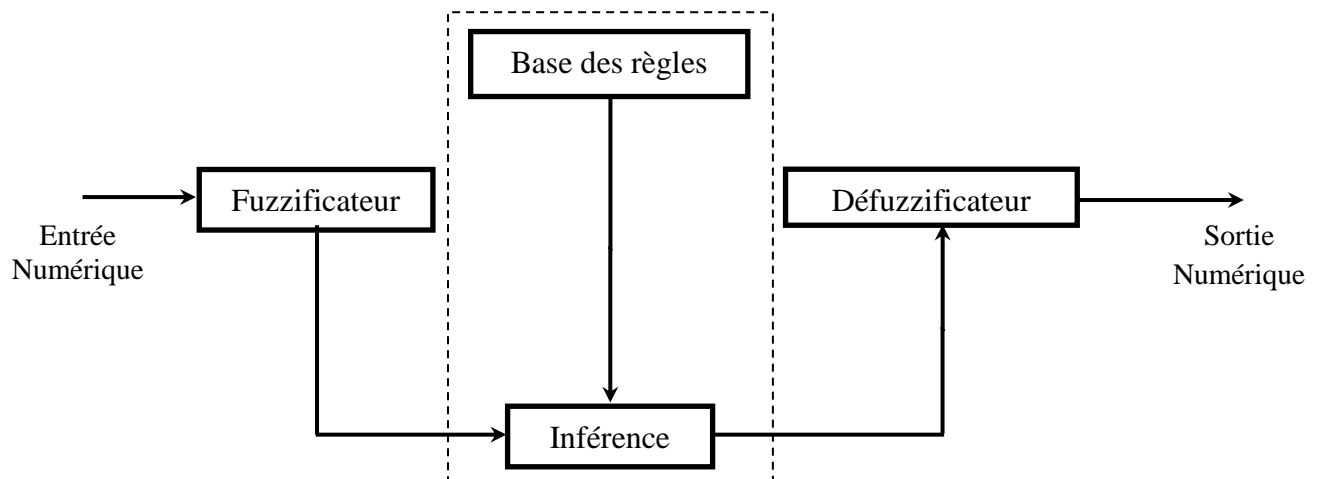


Figure 2.2 : Schéma synoptique d'un contrôleur flou type 1

II.5.1. Fuzzification

C'est le mécanisme réalisant l'interface "Numérique - linguistique". Les variables d'entrée et de sortie choisies pour modéliser ou commander un système sont des grandeurs numériques. L'étape

de fuzzification consiste à transformer ces grandeurs réelles en variables linguistiques en vue d'un traitement d'inférence. Ainsi, à chaque variable d'entrée et de sortie est associé des ensembles caractérisant les termes linguistiques pris par ces variables. Ces termes seront utilisés pour écrire les règles d'inférence. [26]

Le bloc de fuzzification effectue les fonctions suivantes :

- Définition des fonctions d'appartenance de toutes les variables d'entrées ;
- Transformation des grandeurs physiques (réelles ou numériques) à des grandeurs linguistiques ou floues ;
- Représentation d'échelle transférant la plage des variables d'entrées aux univers de discours correspondants.

II.5.2. Base de règle

Ce bloc est une base de connaissance qui correspond à l'expertise ou au savoir-faire de l'opérateur sur le comportement du système. Elle est composée de l'ensemble des informations et des connaissances dans le domaine d'application et le résultat de commande prévu. Elle permet de déterminer le signal de sortie du contrôleur flou et exprime la relation qui existe entre les variables d'entrées transformées en variables linguistiques et les variables de sortie converties également en variables linguistiques [20]. Ainsi, elle est constituée par une collection de règles données sous la forme «Si...Alors...». D'une manière générale, on peut exprimer la $i^{\text{ème}}$ règle floue par la relation suivante :

- Modèle de Mamdani :

$$R^i: \text{Si } x_1 \text{ est } F_1^i \text{ et } x_2 \text{ est } F_2^i \text{ et } \dots x_n \text{ est } F_n^i \text{ Alors } y_j \text{ est } G_j$$

- Modèle de Takagi Sugeno :

$$R^i: \text{Si } x_1 \text{ est } F_1^i \text{ et } x_2 \text{ est } F_2^i \text{ et } \dots x_n \text{ est } F_n^i \text{ Alors } y_i = a_0^i + a_1^i x_1 + \dots + a_n^i x_n$$

Où : $i = 1, \dots, m$.

- m est le nombre total des règles.
- F_1, F_2, \dots, F_n sont les ensembles flous (valeurs linguistique) des entrées (x_1, x_2, \dots, x_n)
- G_j est l'ensemble flou correspondant à la sortie y_j .
- les $a_0^i, a_1^i, \dots, a_n^i$ sont paramètres ajustables des conséquences de la règle R^i .

On peut écrire les règles d'inférence sous forme d'une matrice appelée Matrice d'inférence, qui est généralement antisymétrique. A titre d'exemple, si on considère un contrôleur flou à deux entrées

caractérisées par trois ensembles flous et une sortie, alors la matrice d'inférence peut prendre la forme suivante :

$x_2 \backslash x_1$	F_1^1	F_1^2	F_1^3
F_2^1	G_1	G_4	G_7
F_2^2	G_2	G_5	G_8
F_2^3	G_3	G_6	G_9

Tableau 2.2 : Matrice d'inférence floue.

II.5.3. Mécanisme d'inférence flou

L'inférence floue ou la logique de prise de décision est le cœur du contrôleur flou qui possède la capacité de simuler les décisions humaines et de déduire (inférer) les actions de commande floue à l'aide de l'implication floue et des règles d'inférence de la logique floue. Elle utilise les variables floues transformées par la fuzzification et les règles d'inférence pour créer et déterminer les variables floues de sortie, en se basant sur des opérations floues appliquées aux fonctions d'appartenance. [28]

Comme on l'a mentionné, il existe plusieurs possibilités pour réaliser les opérateurs flous qui s'appliquent aux fonctions d'appartenance.

On introduit la notion de mécanisme ou méthode d'inférence, qui dépend des relations utilisées pour réaliser les différents opérateurs dans une inférence, permettant ainsi un traitement numérique de cette dernière.

Pour le réglage par logique floue, on utilise en général l'une des trois méthodes suivantes :

- Méthode d'inférence Max - Min (Méthode de Mamdani) ;
- Méthode d'inférence Max - Produit (Méthode de Larsen) ;
- Méthode d'inférence Somme - Produit.

II.5.4. Défuzzification

La dernière étape pour avoir un système flou opérationnel s'appelle la défuzzification, c'est une interface "linguistique - Numérique". Lors de la seconde étape, on a généré un tas de commandes sous la forme de variables linguistiques (une commande par règle).

Le but de la défuzzification est de fusionner ces commandes et de transformer les paramètres

résultants en donnée numérique. L'étape de défuzzification se déroule en deux temps : [19] [28]

D'abord, il faut fusionner les variables linguistiques communes à l'aide d'un opérateur de la logique floue choisi par le concepteur du système.

Si on a plusieurs règles d'inférences qui génèrent plusieurs valeurs de la même variable linguistique, on peut choisir un opérateur pour combiner les valeurs de la variable. Cet opérateur sera dans la grande majorité des cas, le OU logique utilisant l'opérateur de maximalité.

Dans un second temps, nous pouvons réellement entamer la partie délicate de la défuzzification.

On a une série de variables linguistiques qui caractérisent une seule et même donnée, défuzzifier la donnée revient donc à trouver la meilleure valeur quantitative en fonction des fonctions d'appartenances des variables linguistiques.

Il existe plusieurs méthodes de défuzzification, la plus utilisée est la méthode du centre de gravité.[30]

$$\bar{y} = \frac{\sum_{i=1}^{N_c} \mu_{ci}(y) \omega_i}{\sum_{i=1}^{N_c} \mu_{ci}(y)} \quad (2.6)$$

Où :

- \bar{y} : la valeur numérique de sortie.
- N_c : le nombre de valeur linguistique de conséquence.
- ω_i : les sommets des ensembles flous de conséquence.
- μ : la valeur d'appartenance.

II.6. Exemple d'application

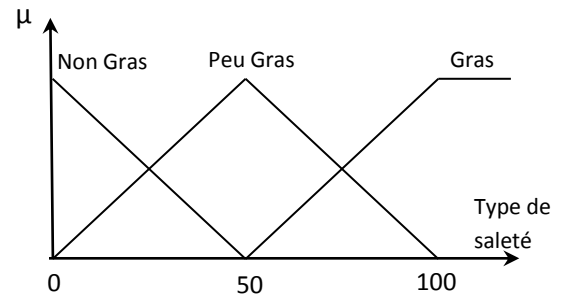
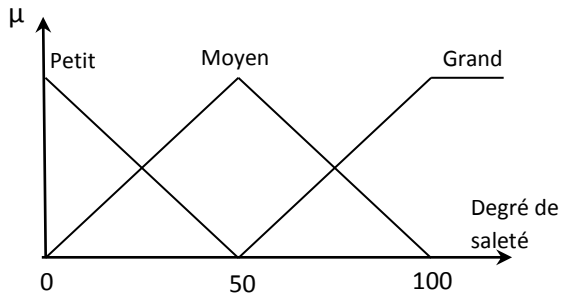
Notre objectif est de commander la durée de lavage (en minute) d'une machine à laver selon le degré et le type de saleté de vêtement à laver. Dans notre application on va considérer les variables E/S de commande suivantes :

Les entrées : • Degré de saleté DS (Petit (P) , Moyen(M) , Grand (G))

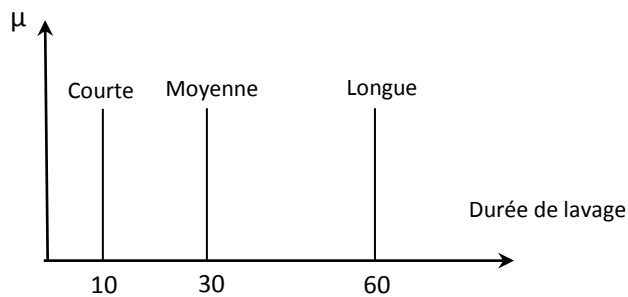
- Type de saleté TS (Non Gras (NG) , Peu Gras (PG) , Gras (G))

La sortie : Durée de Lavage DL (Courte(C), Moyenne (M), Longue (L))

Après avoir défini les variables de commande du processus (Degré de saleté, Type de saleté et Durée de lavage), on doit spécifier les sous-ensembles flous associés à ces variables ainsi que leurs fonctions d'appartenances. On divise l'univers de discours de chaque variable en trois sous-ensembles flous. Les figures suivantes montrent les fonctions d'appartenances des sous-ensembles flous de chaque variable linguistique « Degré de saleté », « Type de saleté » et « Durée de lavage ».



Les sous-ensembles flous de la sortie « Durée de lavage » : courte, moyenne, longue sont de type singletons.



Le tableau ci-dessous représente la base de règles (Matrice d'inférences) en donnant les valeurs de la sortie pour les différentes valeurs linguistiques d'entrées.

Degrés de saleté \ Type de saleté	P	M	G
NG	C	C	M
PG	C	M	L
G	C	L	L

Tableau 2.3: Matrice d'inférences floue

La logique de décision utilisée de type MIN-MAX.

➤ **Question**

Utiliser un contrôleur FLC de type MAMDANI pour déterminer la durée de lavage en minute pour laver un vêtement de type et degré de saleté est égales à 50 %.

➤ **Solution :**

1. Synthèse de contrôleur de type MAMDANI

Etape 1 : La fuzzification

D'après les figures des fonctions d'appartenances de DS, TS.

Pour le degré de saleté

$$\mu_P (DS = 50\%) = 0$$

$$\mu_M (DS = 50\%) = 1$$

$$\mu_G (DS = 50\%) = 0$$

Pour type de saleté

$$\mu_{NG} (TS = 50\%) = 0$$

$$\mu_{PG} (TS = 50\%) = 1$$

$$\mu_G (TS = 50\%) = 0$$

Etape 2 : L'interprétation de chaque règle :

- R1 : si DS est P et TS est NG Alors DL est C

$$\mu_{R1} = \min (\mu_P (DS) , \mu_{NG} (TS)) = \min (0,0) = 0$$

- R2 : si DS est P et TS est PG Alors DL est C

$$\mu_{R2} = \min (\mu_P (DS), \mu_{PG} (TS)) = \min (0,0) = 0$$

⋮
⋮
⋮

- R9 : si DS est G et TS est G Alors DL est L

$$\mu_{R9} = \min (\mu_G (DS) , \mu_G (TS)) = \min (0,0) = 0$$

Etape 3 : La fusion des règles (Agrégation)

$$\mu_C = \max (\mu_{R1} , \mu_{R2} , \mu_{R3} , \mu_{R4}) = 0$$

$$\mu_M = \max (\mu_{R5} , \mu_{R7}) = 1$$

$$\mu_L = \max (\mu_{R6} , \mu_{R8} , \mu_{R9}) = 0$$

Etape 4 : La défuzzification (méthode de centre de gravité)

$$y = \frac{\mu_C \times 10 + \mu_M \times 30 + \mu_L \times 60}{0 + 1 + 0} = 30 \text{min}$$

Donc la durée effective de lavage est 30 min (moyenne durée).

La figure suivante présente en détaille les différentes étapes de structure générale de contrôleur flou.

II.7. Logique floue type 2

II.7.1. Introduction

La notion des ensembles flous de type-2 a été proposée par Lotfi Zadeh comme extension aux ensembles flous types-1.

Un ensemble flou type-2 est caractérisé par une fonction d'appartenance, c.à.d la valeur d'appartenance de chaque élément de l'ensemble est un nombre flou dans $[0,1]$. Les ensembles flous type-2 est utilisés dans les situations où l'incertitude se présente (par exemple l'incertitude sur la forme des fonctions d'appartenances).

Cette partie sera consacrée à la présentation des principes de base et le fondement théorique de la théorie de la logique floue type-2.

II.7.2. Concept de base

II.7.2.1. Les ensembles flous

II.7.2.1.1. Définition 1

Un ensemble flou type-2, noté \tilde{A} est caractérisé par une fonction d'appartenance tridimensionnelle $\mu_{\tilde{A}}(x, u)$ c.à.d :

$$\tilde{A} = \int_{x \in X} \int_{u \in J_x} \mu_{\tilde{A}}(x, u) / (x, u) \quad J_x \subseteq [0 \ 1] \quad (2.7)$$

Avec \int dénote l'union de tous les éléments du produit cartésien sur x et u avec :

$$0 \leq \mu_{\tilde{A}}(x, u) \leq 1$$

A chaque point fixe x de X , J_x est l'appartenance primaire de x , et x appelé variable primaire.[21]

II.7.2.1.2. Définition 2

A chaque valeur de x , notée $x = x'$, le plan dimensionnel dont les axes sont u et $\mu_{\tilde{A}}(x', u)$ est appelé tranche verticale (vertical slice) de $\mu_{\tilde{A}}(x, u)$. Une fonction d'appartenance secondaire est une tranche verticale de $\mu_{\tilde{A}}(x, u)$.

Donc pour $x' \in X$ et $\forall u \in J_{x'} \subseteq [0 \ 1]$ on a :

$$\mu_{\tilde{A}}(x = x', u) = \mu_{\tilde{A}}(x') = \frac{\int_{f_{x'}(u)} f_{x'}(u)}{u} \quad J_{x'} \subseteq [0 \ 1] \quad (2.8)$$

Avec : $0 \leq f_{x'}(u) \leq 1$.

Puisque $\forall x'$, ce x' va appartenir à X , c.à.d $x' \in X$, alors nous notons la fonction d'appartenance secondaire par $\mu_{\tilde{A}}(x)$ qui est une fonction d'appartenance flou type 1.

En se basant sur le concept des ensembles secondaires, on peut réinterpréter un ensemble flou type-2 comme l'union de tous les ensembles secondaire, c.à.d en utilisant l'équation (2.8). [21][14]

On peut écrire \tilde{A} sous la forme suivante :

$$\tilde{A} = \{(x, \mu_{\tilde{A}}(x)) / \forall x \in X\} \quad (2.9)$$

Ou comme : $\tilde{A} = \int_{x \in X} \mu_{\tilde{A}}(x) / x = \int_{x \in X} \left[\int_{u \in J_x} f_x(u) / u \right] / x \quad J_x \subseteq [0 \ 1]$

II.7.2.1.3. Définition 3

Le domaine de la fonction d'appartenance secondaire est appelé appartenance primaire de x noté J_x telle que $J_x \subseteq [0 \ 1] \forall x \in X$

II.7.2.1.4. Définition 4

L'amplitude de la fonction d'appartenance secondaire est appelée degré d'appartenance secondaire notée $f_x(u)$.

II.7.2.1.5. Définition 5

Un ensemble flou type-2 intervalle est un ensemble flou type-2 dont toutes les fonctions d'appartenance secondaires sont des ensembles type-1 de formes intervalles, c.à.d : [5]

$$f_x(u) = 1, \forall u \in J_x \subseteq [0 \ 1], \forall x \in X \quad (2.10)$$

Les ensembles flous type-2 intervalle reflètent l'uniformité de l'incertitude au niveau de la fonction d'appartenance primaire, ce type de fonctions d'appartenance est le plus souvent utilisé dans les systèmes flous type-2. Notons que ce type de fonctions d'appartenance est représenté seulement par ses domaines (intervalles), qui peuvent être exprimés en termes de bornes de gauches et de droites $[l, r]$ ou par leurs centres et largeurs $[c - s, c + s]$ où $c = (l + r) / 2$ et $s = (r - l) / 2$.

II.7.2.1.6. Définition 6

Supposons que chaque fonction d'appartenance secondaire d'un ensemble flou type 2 possède un seul degré d'appartenance unitaire. On définit alors une fonction d'appartenance principale comme l'union de tous ces points : [15][5]

$$M_{\text{principale}}(x) = \int_{x \in X} u / x \quad \text{où} \quad f_x(u) = 1. \quad (2.11)$$

Pour un ensemble flou type-2 intervalle, nous définissons la fonction d'appartenance principale comme étant l'union de toutes les valeurs d'appartenance moyennes de la fonction d'appartenance

primaire. Notons que lorsque l'incertitude des fonctions d'appartenance disparaît, la fonction d'appartenance d'un ensemble flou type-2 se réduit à la fonction d'appartenance principale.

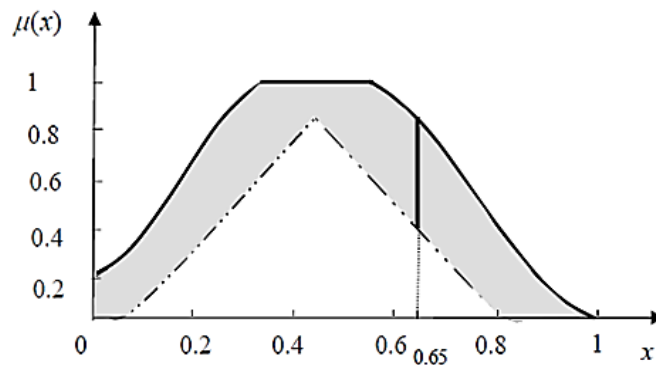


Figure 2.3 : Représentation schématique d'un ensemble flou type-2 intervalle

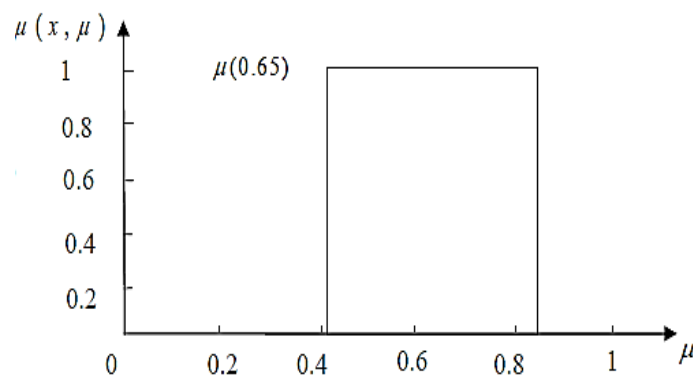


Figure 2.4 : Représentation schématique des appartenances secondaires

On remarque que les appartenances secondaires sont des ensembles flous intervalle type-1.

II.7.2.1.7. Définition 7

L'incertitude dans un ensemble flou type-2 \tilde{A} est représentée par une région bornée appelée «Footprint Of Uncertainty» FOU . C'est l'union de toutes les appartenances primaires :[14][15]

$$FOU(\tilde{A}) = \cup_{x \in X} J_x \tag{2.12}$$

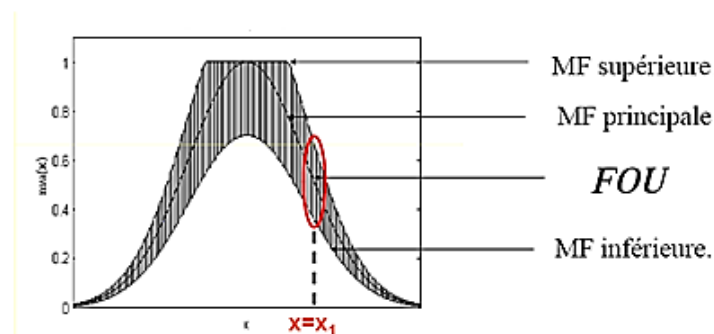


Figure 2.5: la région de l'incertitude

II.7.2.1.8. Définition 8

La fonction d'appartenance supérieure et la fonction d'appartenance inférieure sont des fonctions d'appartenance de type-1 et sont les frontières du FOU d'un ensemble flou \tilde{A} type-2. La fonction d'appartenance supérieure sera notée par $\overline{\mu_{\tilde{A}}}(x)$, et la fonction d'appartenance inférieure sera notée par $\underline{\mu_{\tilde{A}}}(x) \forall x \in X$, alors nous avons :[31]

$$\overline{\mu_{\tilde{A}}}(x) = \overline{FOU(\tilde{A})} \quad \forall x \in X \tag{2.13}$$

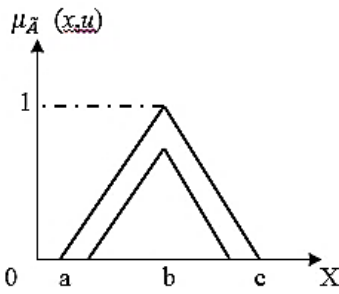
Et

$$\underline{\mu_{\tilde{A}}}(x) = \underline{FOU(\tilde{A})} \quad \forall x \in X \tag{2.14}$$

II.7.3. Différents types d'ensembles flous type-2

- **Ensemble flou type-2 gaussien** : Le degré d'appartenance de chaque point est un ensemble flou type-1 gaussien dont le domaine de définition est inclus dans l'intervalle [0, 1].
- **Ensemble flou type-2 triangulaire** : Le degré d'appartenance de chaque point est un ensemble flou type-1 triangulaire dont le domaine de définition est inclus dans l'intervalle [0, 1].
- **Ensemble flou type-2 intervalle** : Le degré d'appartenance de chaque point est un ensemble ordinaire dont le domaine de définition est inclus dans l'intervalle [0, 1]. Dans ce cas, les appartenances secondaires sont égales à 1.

Les fonctions d'appartenance les plus utilisées sont représentées dans le tableau suivant :

fonction	Forme algébrique	Forme graphique
Fonction triangulaire	$\overline{\mu_{\tilde{A}}}(x, u) = \begin{cases} 0 & x < a \\ \frac{x-a}{b-a} & a \leq x < b \\ \frac{c-x}{c-b} & b \leq x \leq c \\ 0 & x > c \end{cases}$ $\underline{\mu_{\tilde{A}}}(x, u) = \begin{cases} 0 & x < (a + \alpha) \\ (1 - \alpha) \frac{x - (a + \alpha)}{b - (a + \alpha)} & (a + \alpha) \leq x < b \\ (1 - \alpha) \frac{(c - \alpha) - x}{(c - \alpha) - b} & b \leq x \leq (c - \alpha) \\ 0 & x > (c - \alpha) \end{cases}$	

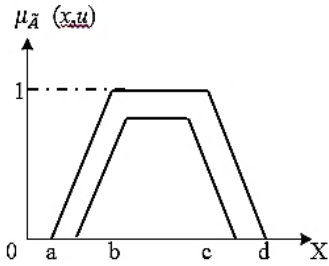
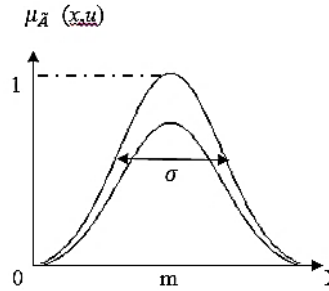
<p>Fonction trapézoïdale</p>	$\overline{\mu_{\tilde{A}}(x, u)} = \begin{cases} 0 & x < a \\ \frac{x-a}{b-a} & a \leq x < b \\ 1 & b \leq x < c \\ \frac{d-x}{d-c} & c \leq x \leq d \\ 0 & x > d \end{cases}$ $\underline{\mu_{\tilde{A}}(x, u)} = \begin{cases} 0 & x < (a + \alpha) \\ (1 - \alpha) \frac{x - (a + \alpha)}{b - (a + \alpha)} & (a + \alpha) \leq x < b \\ (1 - \alpha) & b \leq x < c \\ (1 - \alpha) \frac{(d - \alpha) - x}{(d - \alpha) - c} & c \leq x \leq (d - \alpha) \\ 0 & x > (d - \alpha) \end{cases}$	
<p>Fonction gaussien</p>	$\overline{\mu_{\tilde{A}}(x, u)} = \exp\left(-\frac{1}{2} \left(\frac{x-m}{\sigma}\right)^2\right)$ $\underline{\mu_{\tilde{A}}(x, u)} = (1 - \alpha) \exp\left(-\frac{1}{2} \left(\frac{x-m}{\sigma}\right)^2\right)$	

Tableau 2.4 : fonctions d'appartenance type 2

II.7.4. Opération sur les ensembles flous type 2

Les degrés d'appartenance des ensembles flous type-2 sont des ensembles type-1 ; par conséquent, pour effectuer des opérations telles que l'union et l'intersection sur les ensembles type- 2, nous devons faire appel aux opérations t-conorme et t-norme sur des ensembles de type1. Cela est réalisable en utilisant le principe d'extension de Zadeh [13].

Une opération « * » entre deux nombres ordinaires peut être étendue à deux ensembles type-1:[15]

$$F = \int_v f(v) / v \tag{2.15}$$

Et

$$G = \int_w g(w) / w \tag{2.16}$$

De la façon suivante :

$$F * G = \int_v \int_w [f(v) \cdot g(w)] / (v * w) \tag{2.17}$$

Où « • » est une t-norme, généralement c'est une opération produit ou opération min , par exemple, l'extension de la t-norme aux ensemble de type 1 est donnée par :

$$F \cup G = \int_v \int_w [f(v) \bullet g(w)] / (v \vee w) \quad (2.18)$$

$$F \cap G = \int_v \int_w [f(v) \bullet g(w)] / (v \wedge w) \quad (2.19)$$

En utilisant le principe d'extension, une opération sur n valeurs ordinaires $f(\theta_1 \dots \theta_n)$ peut être étendue à une opération sur n ensembles flous type-1 $F_1 \dots F_n$:

$$f(F_1, \dots, F_n) = \int_{\theta_1} \dots \int_{\theta_n} \mu_{F_1}(\theta_1) * \dots * \mu_{F_n}(\theta_n) / f(\theta_1 \dots \theta_n) \quad (2.20)$$

Où tous les intégrales dénotes des unions logiques, et $\theta_i \in F_i$ pour $i = 1 \dots n$.

Nous allons dans la suite définir le concept de centroïde d'un ensemble flou type-2 en utilisant la formule (2.20), cette notion est très importante dans les systèmes floue type-2.

Rappelons qu'un centroïde d'un ensemble flou A type-1, dont le domaine est discrétisé en N valeurs est donné par :

$$C_A = \frac{\sum_{i=1}^N x_i \mu_A(x_i)}{\sum_{i=1}^N \mu_A(x_i)} \quad (2.21)$$

Similairement, le centroïde d'un ensemble flou \tilde{A} type-2 dont le domaine est discrétisé en N points, peut être défini en utilisant l'équation (2.20) comme suit. Si on pose $D_i = \mu_{\tilde{A}}(x_i)$ alors :

$$C_{\tilde{A}} = \int_{\theta_1} \dots \int_{\theta_n} [\mu_{D_1}(\theta_1) * \dots * \mu_{D_n}(\theta_n)] / \frac{\sum_{i=1}^N x_i \theta_i}{\sum_{i=1}^N \theta_i} \quad (2.22)$$

Où $\theta_i \in D_i$

L'équation veut dire : chaque point x_i de \tilde{A} possède un degré d'appartenance de type-1 (c.à.d son degré d'appartenance est une fonction d'appartenance de type-1) noté $D_i = \mu_{\tilde{A}}(x_i)$. [15][32]

Pour trouver le centroïde, nous considérons chaque combinaison possible $\{\theta_1 \dots \theta_n\}$ tel que $\theta_i \in D_i$. Pour chaque telle combinaison, nous calculons un centroïde de type-1 en utilisant la formule (2.21) en remplaçant chaque $\mu_A(x_i)$ par θ_i , et à chaque point centroïde, nous allons associer un degré d'appartenance égale à la t-norm des degrés d'appartenances des θ_i aux D_i . Si plus d'une combinaison des θ_i nous donne le même point dans le centroïde, on garde le point possédant le plus grand degré d'appartenance. Chaque combinaison $\{\theta_1 \dots \theta_n\}$, ($\theta_i \in D_i$) considérée lorsque nous calculons $C_{\tilde{A}}$, peut être vue comme une fonction d'appartenance d'un certain ensemble flou

A' de type-1 possédant le même domaine que \tilde{A} . Nous appelons A' un ensemble flou type-1 encastré dans \tilde{A} . [32]

Chaque ensemble type-1 encastré possède aussi un point qui est calculé comme la t-norm des degrés d'appartenance secondaires correspondant à $\{\theta_1 \dots \theta_n\}$.

Alors, un ensemble flou type-2 peut être vu comme une large collection d'ensembles type-1 encastrés, dont chacun possède son poids associé. Le centroïde de cet ensemble flou type-2, $C_{\tilde{A}}$ est par conséquent un ensemble type-1 dont les éléments sont les centroïdes de tous les ensembles encastrés de type-1 dans \tilde{A} , leurs degrés d'appartenance sont les poids associés aux ensembles encastrés correspondants. Le calcul du centroïde se simplifie beaucoup lorsque \tilde{A} est un ensemble type-2 intervalle.[23]

Dans ce cas, si A est un ensemble type-2 intervalle, alors l'équation (2.22) sera simplifiée à :

$$C_{\tilde{A}} = \int_{\theta_1} \dots \int_{\theta_1} 1 / \frac{\sum_{i=1}^N x_i \theta_i}{\sum_{i=1}^N \theta_i}$$

Où chaque θ_i appartient à un certain intervalle dans $[0,1]$.

Observons que si le domaine de \tilde{A} ou de $\mu_{\tilde{A}}(x)$ ($x \in \tilde{A}$) est continu, alors le domaine de $C_{\tilde{A}}$ est aussi continu. Le nombre de tous les ensembles type-1 encastrés dans \tilde{A} est infini, par conséquent, les domaines de \tilde{A} et chaque ($x \in \tilde{A}$) doivent être discrétisés pour le calcul de $C_{\tilde{A}}$. Nous remarquons de l'équation (2.22) que si le domaine de chaque D_i est discrétisé en M points, le nombre des combinaisons $\{\theta_1 \dots \theta_n\}$ possibles sera M^N , qui est un nombre très grand même pour M et N petits.[18]

Si, par conséquent les fonctions des D_i ont une structure régulière (exemple : ensemble type-1 intervalle, gaussiennes, triangulaires), on peut obtenir avec exactitude ou approximativement le centroïde sans réaliser tous les calculs.

L'ensemble type-1 réduit d'un ensemble flou type-2 est le centroïde de l'ensemble flou type-2 du *FLS*; par conséquent, chaque élément de l'ensemble de type réduit est un centroïde d'un certain ensemble encastré type-1.

II.8. Structure d'un contrôleur flou type 2

Il existe quatre modules pour un régulateur à logique floue :

1. La fuzzification des entrées et sorties.
2. L'inférence floue selon une base de règle.
3. La réduction de type.
4. La défuzzification des sorties.

L'architecture d'un contrôleur flou est donnée par le schéma suivant :

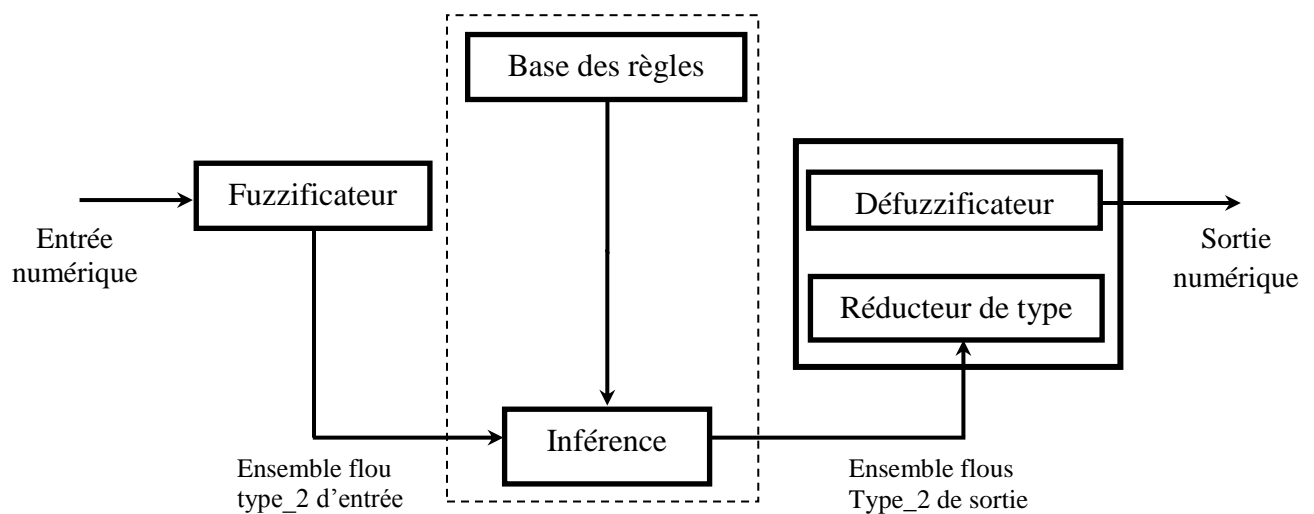


Figure 2.6 : Schéma synoptique d'un contrôleur flou type 2

Les systèmes flous type-2 sont dotés d'un nouveau module appelé réducteur de type, permettant de réduire des ensembles flous de type-2 à des ensembles flous de type-1.

II.8.1. Fuzzification [24]

L'interface de fuzzification fait correspondre à l'entrée déterministe un ensemble flou qui peut être généralement un ensemble flou type 2. Cependant, dans notre travail la fuzzification de type singleton sera utilisée, en d'autres termes, l'entrée floue est un point singulier possédant une valeur d'appartenance unitaire.

II.8.2. Mécanisme d'inférence [29]

La différence entre le type-1 et le type-2 réside seulement dans la nature des fonctions d'appartenance, donc, la structure des règles dans le cas du type-2 va rester exactement la même, la seule différence étant que quelques (ou toutes) les fonctions d'appartenance seront de type-2.

Il n'est pas nécessaire que toutes les fonctions d'appartenance des prémisses et des conséquences soient de type-2. Il suffit qu'une seule fonction d'appartenance dans une prémisse ou dans une conséquence soit de type-2 pour que tout le système le soit aussi.

➤ **Théorème**

Dans un système flou type-2 intervalle, avec une t-norm min ou prod, l'intervalle d'activation $F^l \in [\underline{f}^l, \overline{f}^l]$ de la $i^{ième}$ règle est :

$$\underline{f}^l = \underline{\mu F_1^l}(x_1) * \underline{\mu F_2^l}(x_2) * \dots * \underline{\mu F_p^l}(x_p) \tag{2.23}$$

$$\overline{f}^l = \overline{\mu F_1^l}(x_1) * \overline{\mu F_2^l}(x_2) * \dots * \overline{\mu F_p^l}(x_p) \tag{2.24}$$

Ceci est clairement représenté sur la figure (2.7), où le nombre d'antécédents est $p = 2$. Dans ce cas, le degré d'activation est un ensemble type-1 intervalle $[f^l, \overline{f}^l]$ ou :

$$\underline{f}^l = \underline{f}_1^l * \underline{f}_2^l$$

$$\overline{f}^l = \overline{f}_1^l * \overline{f}_2^l$$

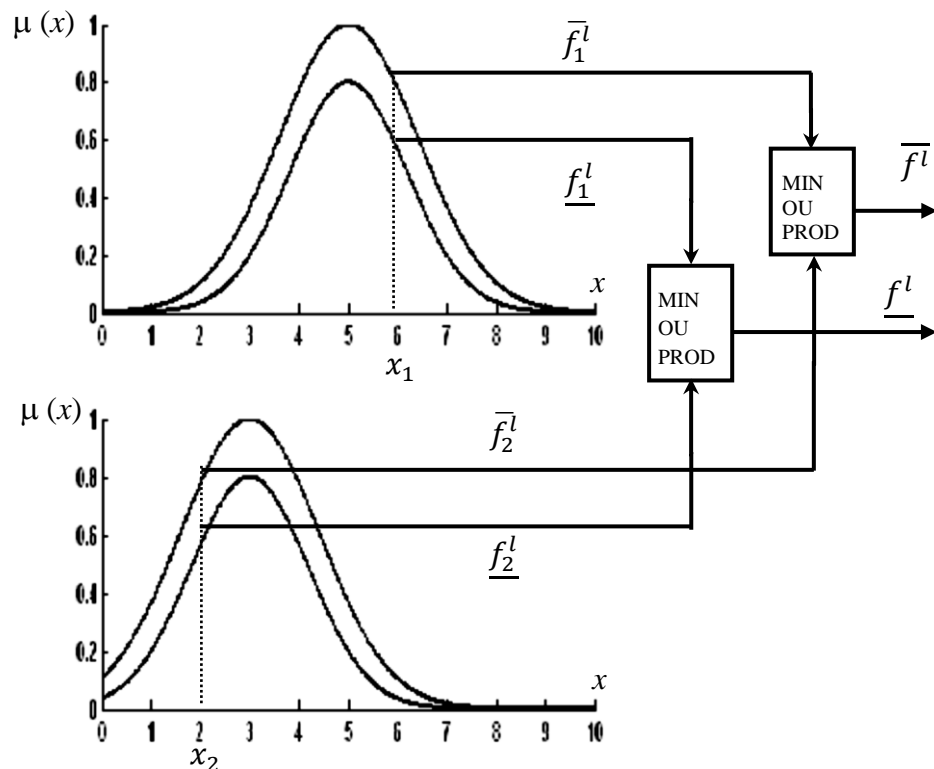


Figure 2. 7 : Opérations d'entrée d'un système flou type-2 avec fuzzification type singleton

II.8.3. Réduction de type [22] [21]

Etant donné que la sortie du système d'inférence est un ensemble flou type-2, son type doit être réduit avant l'étape de défuzzification pour qu'on puisse l'utiliser afin de générer une sortie réelle. C'est la principale différence structurelle entre les systèmes flous type-1 et ceux de type-2. [13]

L'expression de l'ensemble flou type réduit par la méthode des centres d'ensembles est donnée par:

$$Y_{cos}(Y^1, \dots, Y^M, F^1, \dots, F^M) = \int_{y^1} \dots \int_{y^k} \int_{f^1} \dots \int_{f^k} \tau_{l=i}^k \mu_{Y^l}(y^l) * \tau_{l=i}^k \mu_{F^l}(f^l) / \frac{\sum_{i=1}^N f^l y^l}{\sum_{i=1}^N f^l} \quad (2.25)$$

Où τ et $*$ indiquent les t-norme choisies, $y^l \in Y^l = C_{\widetilde{G}^l}$ le centroïde du $l^{\text{ième}}$ ensemble flou conséquent type-2 et $f^l \in F^l(x) = \prod_{i=1}^n \mu_{\widetilde{F}^l_i}(x_i)$ le degré d'activation associé à la $l^{\text{ième}}$ règle pour $l = 1, 2, \dots, k$.

Puisque les ensembles flous utilisés sont type-2 intervalle, alors chaque Y^l et F^l est un ensemble type-1 intervalle, ce qui se traduit par : $\mu_{Y^l}(y^l) = \mu_{F^l}(f^l) = 1$.

L'équation (II.6) peut être réécrite :

$$Y_{cos}(Y^1, \dots, Y^M, F^1, \dots, F^M) = \int_{y^1} \dots \int_{y^k} \int_{f^1} \dots \int_{f^k} 1 / \frac{\sum_{i=1}^N f^l y^l}{\sum_{i=1}^N f^l} \quad (2.26)$$

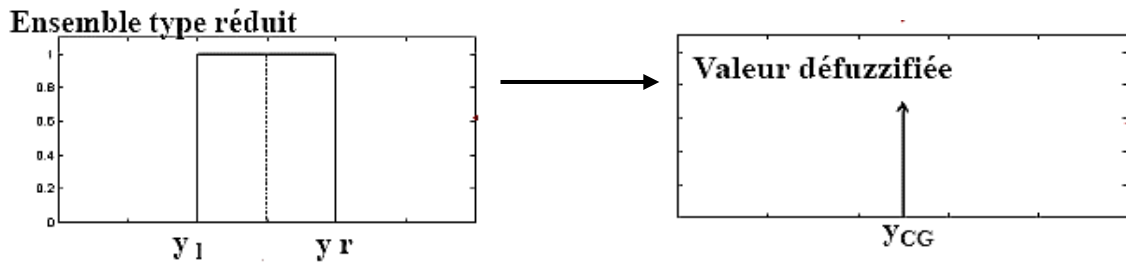
Aussi, étant donné que chaque ensemble dans l'équation (2.26) est un ensemble type-1 intervalle, alors $Y_{cos}(Y^1, \dots, Y^M, F^1, \dots, F^M)$ est aussi un ensemble type-1 intervalle et donc son domaine est un intervalle situé sur l'axe des réels :

$$Y_{cos}(Y^1, \dots, Y^M, F^1, \dots, F^M) = [y_l, y_r] \quad (2.27)$$

II.8.4. Défuzzification

Pour obtenir une sortie ordinaire (non floue) du système flou type-2, nous devons défuzzifier l'ensemble de type réduit. Nous allons transformer l'ensemble flou type-2 en ensemble flou type-1 et la sortie soit [22]:

$$Y_{CG} = \frac{y_l + y_r}{2} \quad (2.28)$$



II.9. Exemple d'application

On prendre le même exemple d'application de type-1

Les Figures suivantes montres les fonctions d'appartenances des sous-ensembles flous de chaque variable linguistique Degré de saleté, Type de saleté et Durée de lavage.

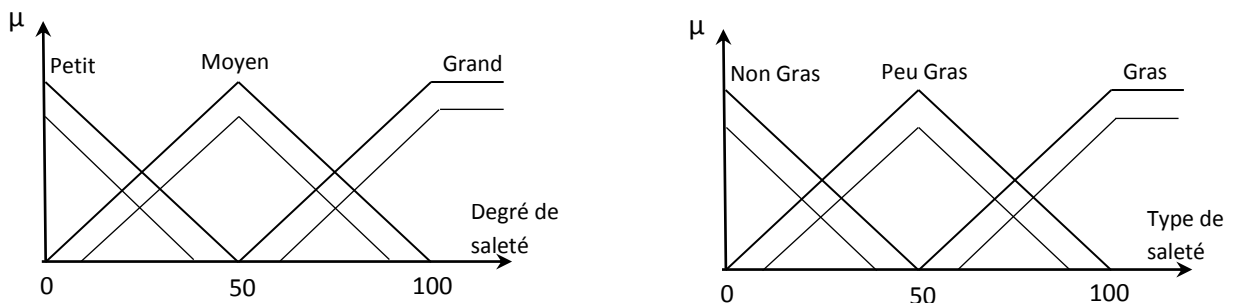


Figure 2.8 : Fonctions d'appartenances des entrées

Les Sous-ensembles flous de la sortie 'Durée de lavage' : courte, moyenne, longue sont de type-2 Singletons.

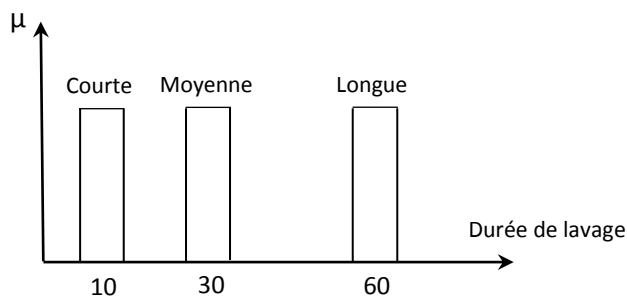


Figure II.9: Fonctions d'appartenances de la sortie

Le tableau ci-dessous représente la base de règles (Matrice d'inférences) en donnant les valeurs de la sortie pour les différentes valeurs linguistiques d'entrées.

Degrés de saleté Type de saleté	P	M	G
NG	C	C	M
PG	C	M	L
G	C	L	L

Tableau 2.5: Matrice d'inférences floue

La logique de décision utilisée de type MIN-MAX.

➤ **Question**

Utiliser un contrôleur FLC type-2 MAMDANI pour déterminer la durée de lavage en minute pour laver un vêtement de type et degré de saleté est égales à 50 %.

➤ **Solution :**

1. Synthèse de contrôleur type-2 MAMDANI

Etape 1 : La fuzzification :

D'après les figures des fonctions d'appartenances de DS, TS.

Pour le Degrée de Saleté	Pour le Type de Saleté
$\overline{\mu_P}(DS = 50\%) = 0$	$\overline{\mu_{NG}}(TS = 50\%) = 0$
$\underline{\mu_P}(DS = 50\%) = 0$	$\underline{\mu_{NG}}(TS = 50\%) = 0$
$\overline{\mu_M}(DS = 50\%) = 1$	$\overline{\mu_{PG}}(TS = 50\%) = 1$
$\underline{\mu_M}(DS = 50\%) = 0.75$	$\underline{\mu_{PG}}(TS = 50\%) = 0.75$
$\overline{\mu_G}(DS = 50\%) = 0$	$\overline{\mu_G}(TS = 50\%) = 0$
$\underline{\mu_G}(DS = 50\%) = 0$	$\underline{\mu_G}(TS = 50\%) = 0$

Etape 2 : L'interprétation de chaque règle :

- R1 : si DS est P et TS est NG Alors DL est C
 $\overline{\mu_{R1}} = \min(\overline{\mu_P}(DS) ; \overline{\mu_{NG}}(TS)) = \min(0 ; 0) = 0$
 $\underline{\mu_{R1}} = \min(\underline{\mu_P}(DS) ; \underline{\mu_{NG}}(TS)) = \min(0 ; 0) = 0$
- R2 : si DS est P et TS est PG Alors DL est C
 $\overline{\mu_{R2}} = \min(\overline{\mu_P}(DS) ; \overline{\mu_{PG}}(TS)) = \min(0 ; 0) = 0$
 $\underline{\mu_{R2}} = \min(\underline{\mu_P}(DS) ; \underline{\mu_{PG}}(TS)) = \min(0 ; 0) = 0$

⋮

- R9 : si DS est G et TS est G Alors DL est L
 $\overline{\mu_{R9}} = \min(\overline{\mu_G}(DS) ; \overline{\mu_G}(TS)) = \min(0 ; 0) = 0$
 $\underline{\mu_{R9}} = \min(\underline{\mu_G}(DS) ; \underline{\mu_G}(TS)) = \min(0 ; 0) = 0$

Etape 3 : La fusion des règles (Agrégation)

$$\mu_{C_{left}} = \max(\underline{\mu_{R1}}, \underline{\mu_{R2}}, \underline{\mu_{R4}}, \underline{\mu_{R7}}) = 0$$

$$\mu_{C_{right}} = \max(\overline{\mu_{R1}}, \overline{\mu_{R2}}, \overline{\mu_{R4}}, \overline{\mu_{R7}}) = 0$$

$$\mu_{M_{left}} = \max(\underline{\mu_{R3}}, \underline{\mu_{R5}}) = 0.75$$

$$\mu_{M_{right}} = \max(\overline{\mu_{R3}}, \overline{\mu_{R5}}) = 1$$

$$\mu_{L_{left}} = \max(\underline{\mu_{R6}}, \underline{\mu_{R8}}, \underline{\mu_{R9}}) = 0$$

$$\mu_{L_{right}} = \max(\overline{\mu_{R6}}, \overline{\mu_{R8}}, \overline{\mu_{R9}}) = 0$$

Etape 4 : La défuzzification (méthode de centre de gravité)

$$y_{left} = \frac{\mu_C \times (10 - \alpha) + \mu_M \times (30 - \alpha) + \mu_L \times (60 - \alpha)}{0 + 0.75 + 0} = 29.75 \text{ min}$$

$$y_{right} = \frac{\mu_C \times (10 + \alpha) + \mu_M \times (30 + \alpha) + \mu_L \times (60 + \alpha)}{0 + 1 + 0} = 30.25 \text{ min}$$

$$y_{cos} = \frac{y_{left} + y_{right}}{2} = 30 \text{ min}$$

Donc la durée effective de lavage est 30 min (moyenne durée).

La figure suivante présente en détaille les différentes étapes de structure générale de contrôleur flou.

II.10. Avantages et inconvénients du régulateur par logique flou

Le réglage par logique floue réunit un certain nombre d'avantages et inconvénients essentiels [17].

➤ **Les avantages sont :**

- Le non nécessité d'une modélisation du système à régler.
- La possibilité d'implémenter des connaissances (linguistique) de l'opérateur de processus,
- La maîtrise du système à régler avec un comportement complexe (fortement non linéaire et difficile à modéliser).
- L'emploi possible aussi pour les processus rapides (grâce à des processeurs dédiés).
- La disponibilité de systèmes de développement efficaces soit pour micro processeurs ou PC (solution logicielle), soit pour circuit intégrés (processeur dédiés, solution matérielle).

➤ **Par contre, les inconvénients sont :**

- Le manque de directive précise pour la conception d'un réglage (choix des grandeurs à mesurer, détermination de la fuzzification, des inférences et de la défuzzification).
- L'approche artisanale et non systématique (implémentation des connaissances de l'opérateur souvent difficile).

II.11. Conclusion

Dans ce chapitre, nous avons abordé principalement deux techniques de commande intelligente : la logique floue type-1 et la logique floue type-2. La logique floue, établie par Zadeh en 1965, permet la représentation et le traitement de connaissances imprécises ou approximatives. Le nombre d'applications basées sur la théorie de la logique floue a augmenté considérablement ces dernières années car cette logique est exprimée usuellement par des règles linguistiques de la forme Si –Alors. Elle est utilisée pour résoudre les problèmes de décision en contrôle ou pour décrire le comportement dynamique d'un système inconnu ou mal défini. La première génération de la logique floue est appelée la logique floue type-1. Une nouvelle logique floue appelée logique floue type-2 est introduite pour mieux représenter l'imprécision et l'incertitude. Les fonctions d'appartenance floues type-1 sont bidimensionnelles, par contre, les fonctions d'appartenance floues type-2 sont tridimensionnelles. La nouvelle (troisième) dimension des ensembles flous type-2 fournit un degré de liberté supplémentaire permettant de prendre en charge la modélisation des incertitudes. De ce fait, les ensembles flous type-2 ont la capacité de modéliser les incertitudes parce que leurs fonctions d'appartenance sont eux-mêmes floues.

Chapitre 3

les réseaux de neurone et la commande neuro-floue

III.1. Introduction	40
III.2. Les réseaux de neurones	40
III.3. Réseau neuro-flou	51
III.4. Exemple d'application	57
III.5. Conclusion	60

II.1. Introduction

Dans ce chapitre, nous présentons une méthode importante dans le domaine d'intelligence artificielle, qui est les réseaux de neurones.

Aujourd'hui, cette méthode est implantée dans divers domaines, tel que l'automatique, Traitement du signal.....

Les réseaux de neurone permettent de copier la nature et de reproduire des modes de raisonnement et de comportement de l'être humain, qui sont constitués des fonctions non-linéaires paramétrées, qui sert à la modélisation de processus et la synthèse des lois de commande. Les réseaux de neurones artificiels sont composés de neurones artificielle simples montés en réseaux, à petites fonctions mathématiques, qui permette, de former des fonctions complexes très utiles.

Nous présentons aussi dans ce chapitre la commande neuro-floue, qui est une combinaison de la logique floue et les réseaux de neurones, sert à prendre les avantages de ces deux méthodes. Donc un système neuro-flou est un système flou formé par un algorithme d'apprentissage inspiré de la théorie des réseaux de neurones.

II.2. Les réseaux de neurones

L'origine des réseaux de neurones vient de l'essai de modélisation mathématique du cerveau humain. Les premiers travaux datent de 1943 et sont l'œuvre de M. Mac Culloch et Pitts. Ils supposent que l'impulsion nerveuse est le résultat d'un calcul simple effectué par chaque neurone et que la pensée née grâce à l'effet collectif d'un réseau de neurones interconnectés. Ils ont connu des débuts prometteurs vers la fin des années 50, mais le manque d'approfondissement de la théorie a gelé ces travaux jusqu'aux années 80.

Les réseaux de neurones forment une famille de fonctions non linéaires, permettant de construire, par apprentissage, une très large classe de modèles et de contrôleurs. Un réseau de neurones est un système d'opérateurs non linéaires interconnectés, recevant des signaux de l'extérieur par ses entrées, et délivrant des signaux de sortie, qui sont en fait les activités de certains neurones [6].

II.2.1. Historique [25] [30]

- **1890** : W. James, célèbre psychologue américain introduit le concept de mémoire associative, et propose ce qui deviendra une loi de fonctionnement pour l'apprentissage sur les réseaux de neurones connue plus tard sous le nom de loi de Hebb.

- **1943** : J. Mc Culloch et W. Pitts laissent leurs noms à une modélisation du neurone biologique (un neurone au comportement binaire). Ceux sont les premiers à montrer que des réseaux de neurones formels simples peuvent réaliser des fonctions logiques, arithmétiques et symboliques complexes (tout au moins au niveau théorique).
- **1949** : D. Hebb, physiologiste américain explique le conditionnement chez l'animal par les propriétés des neurones eux-mêmes. Ainsi, un conditionnement de type pavlovien tel que, un chien nourri tous les jours à la même heure, entraîne chez cet animal la sécrétion de salive à cette heure précise même en l'absence de nourriture. La loi de modification des propriétés des connexions entre neurones qu'il propose, explique en partie ce type de résultats expérimentaux.
- **1957** : F. Rosenblatt développe le modèle du Perceptron. Il construit le premier Neuro ordinateur basé sur ce modèle et l'applique au domaine de la reconnaissance de formes. Notons qu'à cet époque les moyens à sa disposition sont limités et c'est une prouesse technologique que de réussir à faire fonctionner correctement cette machine plus de quelques minutes.
- **1960** : B. Widrow, un automaticien, développe le modèle Adaline (Adaptative Linear Element). Dans sa structure, le modèle ressemble au Perceptron, cependant la loi d'apprentissage est différente. Celle-ci est à l'origine de l'algorithme de rétro propagation de gradient très utilisé aujourd'hui avec les Perceptrons multicouches. Les réseaux de type Adaline restent utilisés de nos jours pour certaines applications particulières.
- **1982** : J.J. Hopfield est un physicien reconnu pour les réseaux de neurones artificiels. Il présente une théorie du fonctionnement et des possibilités des réseaux de neurones.
- **1983** : La Machine de premier modèle connu apte à traiter de manière satisfaisante les limitations recensées dans le cas du perceptron. Mais l'utilisation pratique s'avère difficile, la convergence de l'algorithme étant extrêmement longue (les temps de calcul sont considérables).
- **1985** : La rétro-propagation de gradient apparaît. C'est un algorithme d'apprentissage adapté aux réseaux de neurones multicouches (aussi appelés Perceptrons multicouches). Sa découverte réalisée par trois groupes de chercheurs indépendants indique que "la chose était dans l'air". Dès cette découverte, nous avons la possibilité de réaliser une fonction non linéaire d'entrée/sortie sur un réseau en décomposant cette fonction en une suite d'étapes linéairement séparables. De nos jours, les réseaux multicouches et la rétro propagation de gradient reste le modèle le plus étudié et le plus productif au niveau des applications.

II.2.2. Neurone biologique

Le cerveau humain contient environ 100 milliards de neurones biologique. Les neurones biologique est une cellule vivante spécialisée dans le traitement des signaux électriques. Un neurone typique a plusieurs composants. Les dendrites, le corps cellulaire et l'axone.

Le corps cellulaire, qui fait la somme des influx, qui lui proviennent ; si cette somme dépasse un certain seuil, il envoie lui-même un influx par intermédiaire de l'axone.

L'axone, qui permet de transmettre les signaux émis par le corps cellulaire aux autres neurones.

Les dendrites, qui sont les récepteurs principaux du neurone, captant les signaux qui lui parviennent.

Les synapses, qui permettent aux neurones de communiquer avec les autres via les axones et les dendrites [4].

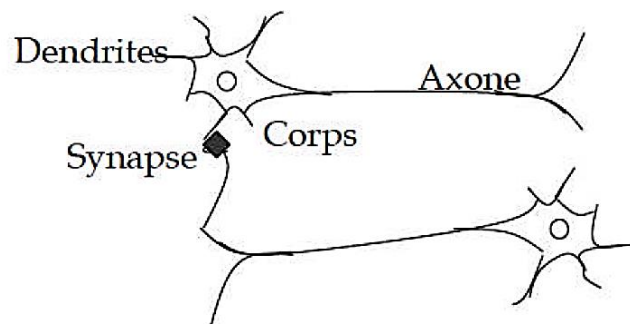


Figure 3.1 : Neurone biologique

II.2.3. Le neurone formel

Un neurone formel est une fonction algébrique non linéaire et bornée. Les modèles les plus utilisés sont basés sur le modèle développé par Mc Culloch & Pitts. Le neurone peut être représenté par une cellule possédant plusieurs entrées et une sortie [25].

II.2.3.1. Définition

C'est un processeur très simple qui calcule une somme pondérée et qui applique à cette somme une fonction de transfert non linéaire (échelon, sigmoïde, gaussienne, ...) [27].

Ainsi, quand des signaux d'entrée x_1, x_2, \dots, x_n arrivent au neurone par des liens de raccordement avec les poids associés w_1, w_2, \dots, w_n , respectivement, l'entrée résultante du neurone, appelée l'entrée nette, est la somme pondérée.

$$\sum_{i=0}^n w_i x_i \quad (3.1)$$

Si le seuil de polarisation est b et la fonction d'activation est $f(z)$ alors la sortie de ce neurone est :

$$y = f\left(\sum_{i=0}^n w_i x_i - b\right) \tag{3.2}$$

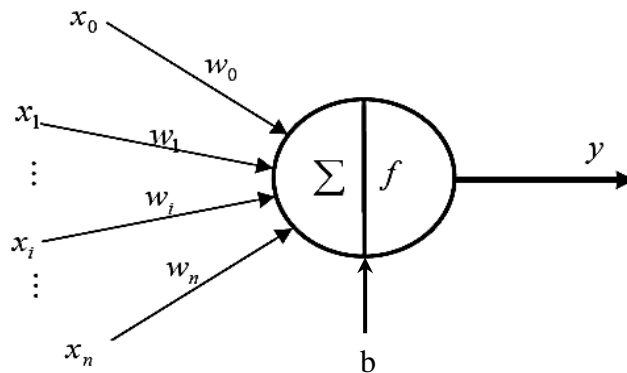


Figure 3.2 : Neurone formel

Les fonctions d'activation les plus utilisées sont les suivantes :

Fonction d'activation	Forme graphique
Tout ou rien	
Fonction signe	
Saturation	
Fonction gaussienne	

Tableau 3.1 : Les fonctions d'activation

III.2.4. Architecture des réseaux de neurones formels

On distingue deux grands types d'architectures de réseaux de neurones : les réseaux de neurones non bouclés et les réseaux de neurones bouclés [4][10].

III.2.4.1. Les réseaux de neurones non bouclés (ou statiques)

Ce sont des réseaux unidirectionnels sans retour arrière (feedforward).

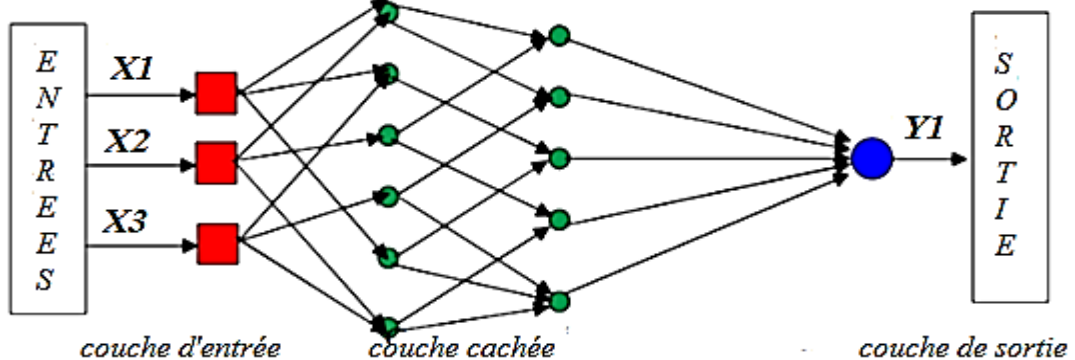


Figure 3.3. Réseau de neurone non bouclé.

Les réseaux de neurones non bouclés sont des objets statiques : si les entrées sont indépendantes du temps, les sorties le sont également. Ils sont utilisés principalement pour effectuer des tâches d'approximation de fonction non linéaire, de classification ou de modélisation de processus statiques non linéaires.

a. Réseau non bouclé Mono - couche

Ce type de réseau possède une couche d'entrée recevant les stimuli à traiter par l'intermédiaire des nœuds sources. Cette couche se projette en une couche de sortie composée de neurones (nœuds de calcul) transmettant les résultats du traitement au milieu extérieur. La figure (3.4) représente un réseau proactif mono-couche de 4 nœuds d'entrée et 3 nœuds de sortie. La désignation mono-couche est attribuée à la couche de sortie.

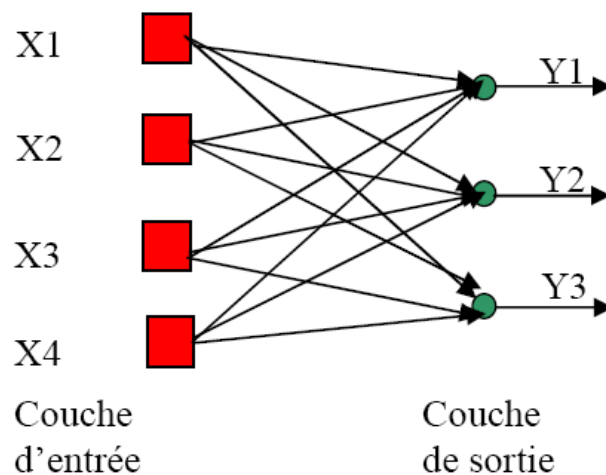


Figure 3.4 : Réseau non bouclé mono couche

b. Les réseaux non bouclés multi-couches

Ce type de réseaux proactifs se caractérise par la présence d'une ou de plusieurs couches cachées, dont les nœuds de calcul correspondants s'appellent neurones cachés. Les couches cachées s'interposent entre l'entrée du réseau et sa sortie. Leur rôle est d'effectuer un prétraitement des signaux d'entrée, reçus par la couche d'entrée en provenance du milieu extérieur, et de transmettre les résultats correspondants à la couche de sortie ou seront déterminées les réponses finales du réseau.[10]

Ce rôle de prétraitement fait que, en ajoutant une ou plusieurs couches cachées, le réseau est capable d'extraire plus de propriétés statistiques que celles extraites d'un réseau similaire ayant moins de couches cachées. Ceci est utile pour réaliser des fonctions plus complexes que de simple séparation linéaires.

La figure (3.5) illustre un réseau à une seule couche cachée comportant 4 unités d'entrées, 3 unités cachées et 2 unités de sortie (réseau 4-3-2)

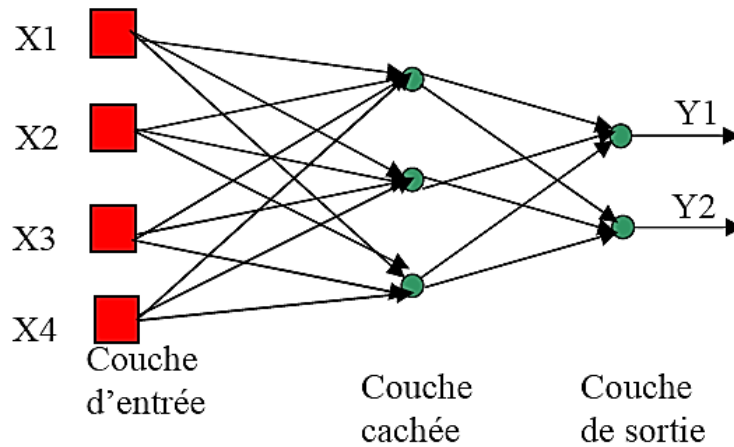


Figure 3.5 : Réseau non bouclé multi couche.

Ce réseau est dit complètement connecté dans le sens que chaque nœud d'une couche est connecté à tous les nœuds de la couche adjacente suivante. Si éventuellement serait dit partiellement connecté.

III.2.4.2. Les réseaux de neurones bouclés (ou dynamiques)

Un réseau est bouclé, ou dynamique, si son graphe possède au moins un cycle. Il s'agit de réseaux de neurones avec retour en arrière (feedback network ou recurrent network).[30]

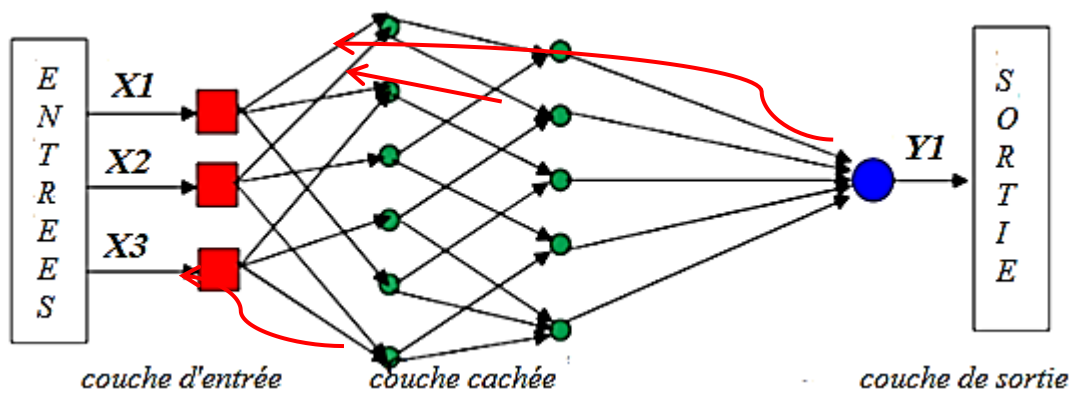


Figure 3.6 : Réseau de neurone bouclé

Les réseaux de neurones bouclés sont utilisés pour effectuer des tâches de modélisation des systèmes dynamiques, de commande de processus, ou de filtrage.

III.2.4.3. Les réseaux de neurones à connexion complexes

Dans la catégorie des réseaux de neurones à connexion complexes se trouvent tous les réseaux qui ne se trouvent pas leur place dans les deux catégories précédemment citées. Nous pouvons mentionner les SOM (Self Organizing Map), LVQ (Learning Vector Quantisation), ART (Adaptative Resonance Theorie) [10].

III.2.5. Processus d'apprentissage

On appelle apprentissage des réseaux de neurones artificiels la procédure itérative qui consiste à estimer les paramètres des neurones du réseau afin que celui-ci remplisse au mieux la tâche qui lui est affectée [6].

L'apprentissage est la caractéristique principale des réseaux de neurones et il peut se faire de différents règles et manières.

III.2.5.1. Types d'apprentissages

Les techniques d'apprentissage peuvent être classées en deux catégories :

a. 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 est terminé lorsque tous les couples entrées-sorties sont reconnus par le réseau. Ce type d'apprentissage se retrouve, entre autres, dans le perceptron. [8]

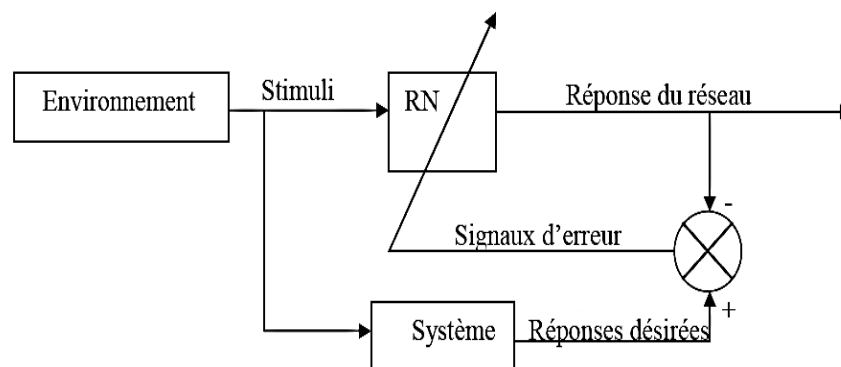


Figure 3.7 : Apprentissage supervisé

b. Apprentissage non supervisé

L'apprentissage est qualifié de non supervisé lorsque, seules les valeurs d'entrée sont disponibles. Il repose sur un critère interne de conformité du comportement du réseau par rapport à des spécifications générales et non sur des observations externes. Dans ce cas, l'apprentissage est basé sur des probabilités. Le réseau va se modifier en fonction des régularités statistiques de l'entrée. L'algorithme d'apprentissage exploite pour se faire une mesure prédéterminée de la qualité de représentation de la connaissance afin d'ajuster les paramètres du réseau. [8]

Ce type d'apprentissage est représenté schématiquement en figure (3.8)

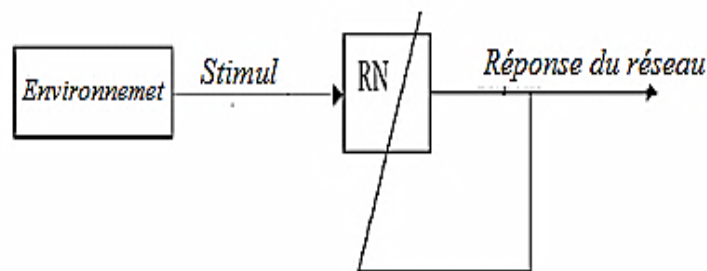


Figure 3.8 : Apprentissage non supervisé

III.2.5.2. Les méthodes d'apprentissage

Dans les systèmes experts, les connaissances de l'expert ont une forme innumérée, elles sont exprimées sous forme de règles. Dans le cas des réseaux de neurones, les connaissances ont une forme distribuée, elles sont codées par les poids des connexions, la topologie du réseau, les fonctions de transfert de chaque neurone, le seuil de ces fonctions, la méthode d'apprentissage utilisée. Il existe un certain nombre de méthodes d'apprentissage. [30]

a. Règle de Hebb

C'est la méthode d'apprentissage la plus ancienne (1949), elle est inspirée de la biologie. Elle traduit le renforcement des connexions liant deux neurones activés. Si un des deux neurones au moins n'est pas activé, le poids de la connexion n'est pas modifié. [10]

b. Rétro-propagation du gradient de l'erreur

Cet algorithme est utilisé dans les réseaux de type feed-forward, ce sont des réseaux de neurones à couches, ayant une couche d'entrée, une couche de sortie, et au moins une couche cachée. Il n'y a pas de récursivité dans les connexions, et pas de connexions entre neurones de la même couche. Le principe de la rétro-propagation consiste à présenter au réseau un vecteur d'entrées, de procéder au

calcul de la sortie par propagation à travers les couches, de la couche d'entrées vers la couche de sortie en passant par les couches cachées. Cette sortie obtenue est comparée à la sortie désirée, une erreur est alors obtenue. A partir de cette erreur, est calculé le gradient de l'erreur qui est à son tour propagé de la couche de sortie vers la couche d'entrée, d'où le terme de rétro-propagation. Cela permet la modification des poids du réseau et donc l'apprentissage. L'opération est répétée pour chaque vecteur d'entrée et cela jusqu'à ce que le critère d'arrêt soit vérifié.

c. Apprentissage par la descente de gradient

La construction d'un modèle à base de neurone à partir de donnée (mesure) se fait par l'apprentissage progressif qu'on appelle aussi l'entraînement. Le calcul de l'erreur guide l'entraînement du RN.

L'entraînement d'un RN consiste alors à déterminer les poids (ω_i), des connexions entre les différentes couches de telle façon que la sortie de RN soit à tout moment proche que possible à la sortie du système à modéliser.

Si $y_m(t)$ la sortie du système à modéliser et $y(t)$ la sortie du RN et si on a T mesure donc l'objectif peut s'écrire

$$J = \frac{1}{2} \sum_{t=1}^T (y(t) - y_m(t))^2 \quad (3.3)$$

Si on note ω l'ensemble de poids de RN, la modélisation neuronale consiste à chercher les poids optimaux pour que l'erreur est très petit où résoudre les problèmes d'optimisation.

Les données sont présentés par paquet le nombre de paquet correspond aux nombre mesure d'entrées.

On peut utiliser l'algorithme du gradient

$$\omega_{i+1} = \omega_i - \alpha \nabla J_i(\omega_i) \quad (3.4)$$

s'il y a pas de contraintes sur le poids ω on peut utilisé un algorithme d'optimisation sans contraintes. La fonction de cout n'est pas nécessairement convexe donc elle peut avoir plusieurs minimums locaux.

➤ L'algorithme de descente de gradient

L'algorithme d'apprentissage par la descente de gradient est définie par :

Entrée : nombre d'itération « T ».

Le pas α ($\alpha = 10^{-2}$)

Initialisation aléatoire des poids ω_i pour i varie entre '1' et 'n'.

Répéter (jusqu'à la convergence ou atteinte du nombre maximal d'itération)

```

Pour tout  $i$ 
|  $\Delta\omega_i = 0$ 
Fin pour

Pour tout  $(x(t), y(t))$ 
|  $y(t) = x(t) \cdot \omega$ 
|   Pour tout  $i$ 
|   |  $\Delta\omega_i = \Delta\omega_i + \alpha (y_m(t) - y(t)) x_i(t)$ 
|   Fin pour
Fin pour

Pour tout  $i$ 
|  $\omega_i = \omega_i + \Delta\omega_i$ 
Fin pour

Fin de répéter

```

III.2.6. Avantages et inconvénients des réseaux de neurones [7]

a. 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 ;
- ✓ Simple à manier, beaucoup moins de travail personnel à fournir que dans l'analyse statistique classique. Aucune compétence en mathématiques, informatique statistique requise ;
- ✓ Comportement moins mauvais en cas de faible quantité de données.

b. 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.

III.3. Réseau neuro-flou

III.3.1. La combinaison des réseaux neuronaux et de la logique floue

Les réseaux de neurones et la logique floue sont deux concepts totalement inspirés du raisonnement humain. Bien que ces deux modèles numériques soient différents du point de vue structure, ils possèdent de nombreux points communs. En effet, l'utilisation de ces deux techniques ne nécessite pas de modèle mathématique bien défini car la résolution se base sur les valeurs numériques des entrées et des sorties pour les réseaux neuronaux et sur la logique floue du système dans le cas de la logique floue. De plus, les résultats sont généralement incertains. En revanche des deux techniques peuvent accomplir des tâches complémentaires. La logique floue fournit des connaissances avec un certain degré d'incertitude et les réseaux de neurones peuvent modéliser et reproduire l'apprentissage humain, d'où l'idée de combiner ces deux techniques pour créer un système artificiel intelligent qui résout les problèmes de manière proche de celle de l'être humain [25].

Les méthodes des réseaux de neurones peuvent être fusionnées à l'intérieur des systèmes flous. Ceci permet de les adapter à de nouvelles situations et de nouvelles règles, lors du fonctionnement en temps réel. De même, les techniques de la logique floue peuvent être introduites dans les réseaux neuronaux afin de perfectionner et d'améliorer l'expressivité et la flexibilité du réseau. Ces combinaisons sont appelées les systèmes hybrides : neuro-flous (fuzzy-neuro system) de telles combinaisons permettent non seulement de s'approcher davantage de la structure de la pensée humaine mais aussi d'obtenir un champ d'utilisation plus large.

Et d'utiliser les avantages de chaque technique, à savoir la simplicité des connaissances floues d'une part et l'habileté d'apprentissage des réseaux de neurones de l'autre. L'utilisation des réseaux hybrides permet, en plus, de diminuer le temps d'apprentissage par la possibilité d'introduire de la connaissance à priori dans le système d'aide au diagnostic. Le choix de la façon d'intégration de ces deux techniques est crucial.

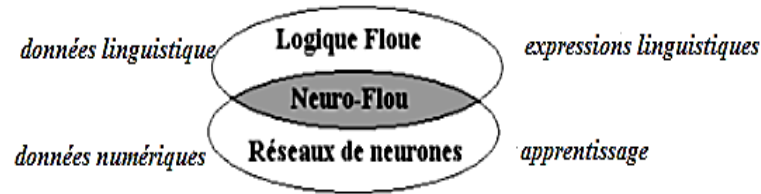


Figure 3.9 : Principe du système Neuro-flou.

III.3.2. Définition

Un système neuro-flou hybride est un réseau de neurone avec un signal flou, des poids flous, une fonction de transfert floue et un module d'adaptation de la structure de RNF par un apprentissage d'un ensemble de données [6].

III.3.3. Principe de fonctionnement

Les réseaux Neuro-flous hybride apprennent des rapports et des modèles en utilisant un algorithme d'apprentissage supervisé qui examine les données dans un ensemble de la formation qui consiste en exemples d'entrées et leurs sorties associées. Pendant la phase d'apprentissage, un réseau Neuro-flou hybride modifie sa structure interne pour refléter le rapport entre les entrées et les sorties dans l'ensemble de la formation (base de connaissance). L'exactitude d'un Réseau Neuro-flou est vérifiée après que le cycle d'apprentissage soit complet en utilisant un ensemble séparé d'entrées et sorties appelé l'ensemble de la validation [10].

III.3.4. Architectures neuro-floues

Plusieurs architectures ont été développées depuis les années quatre-vingt et sont orientées généralement vers la commande des systèmes complexes et les problèmes de classification

En conséquence, il existe trois architectures hybrides Neuro-floues : [30]

III.3.4.1. Première architecture

La première méthode neuro-floue consiste au codage du système d'inférence floue sous la forme d'un réseau de neurones multicouches dans lequel les poids de réseau de neurone correspondent aux paramètres du système. L'architecture du réseau dépend du type de règle et des méthodes d'inférences, d'agrégation et de défuzzification choisies.

Par exemple, pour des règles de la forme : **si x_1 est A_i et x_2 est B_j alors y est C_k**

on fait appel à un réseau de neurones qui admet pour entrées les valeurs x_1 et x_2 prises par les variables X_1 et X_2 et dont les deux couches cachées correspondent respectivement au calcul de la valeur des fonctions d'appartenances A_i pour x_1 et B_j pour x_2 , et à celui de la valeur prise par la conjonction des conditions de chaque règle utilisant un opérateur de conjonction adéquat. Cette situation est schématisée sur la figure suivante :

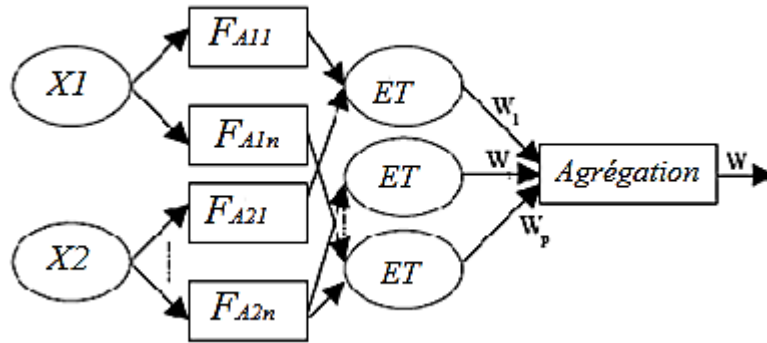


Figure 3.10 : Première architecture Neuro-floue

III.3.4.2. Deuxième architecture

Cette méthode utilise des réseaux de neurones et des systèmes flous associés en série ou en parallèle. Plusieurs variantes sont ainsi possibles :

Le réseau de neurones fonctionne en amont du système flou (figure 3.11). Les variantes d'entrées du système flou sont déterminées à partir des sorties du réseau de neurones (dans le cas où elles ne sont pas mesurables directement) ou encore un réseau de neurones effectue une tâche de classification ou de reconnaissance de formes, suivie d'un système flou d'aide à la décision.

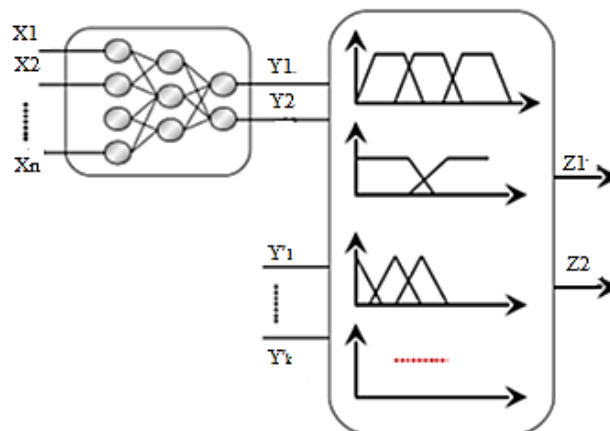


Figure 3.11 : Deuxième architecture des réseaux Neuro-Flou, réalisation en série

Un réseau de neurones qui fonctionne en aval du système flou (figure 3.12), dans le but d'ajuster les sorties d'un système de commande floue à de nouvelles connaissances obtenues, les variables de sorties étant les erreurs sur les variables de sortie du système flou.

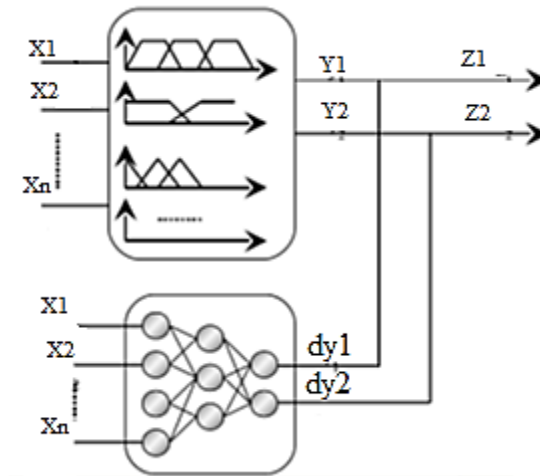


Figure 3.12 : Réseau de neurones fonctionnant en aval d'un système flou

III.3.4.3. Troisième architecture

Ce type d'association entre réseau de neurones et systèmes flous, correspond à l'utilisation des réseaux de neurones pour remplacer toutes ou quelques composantes d'un système flou. De tels réseaux servent à l'apprentissage des fonctions d'appartenance, au calcul de l'inférence, à la réalisation de la phase d'agrégation et de défuzzification. Ils peuvent réaliser l'extraction des règles floues, en analysant la corrélation qui existe entre les entrées et les sorties du réseau de neurones.[10]

Ces approches ont une grande importance car elles sont capables de résoudre des problèmes importants :

- La détermination et l'apprentissage des fonctions d'appartenances.
- La détermination des règles.
- L'adaptation à l'environnement du système.

III.3.5. Commande neuro-flou adaptative (ANFIS : Adaptative Network Fuzzy Inference System)

Le système ANFIS est une classe de réseau adaptatif proposé par JANG en 1992. Il peut être vu comme un RN non bouclé (RN statique) pour lequel chaque couche est un composant d'un

système neuro-flou. Le modèle ANFIS est le modèle le plus utilisé en pratique, des applications dans le traitement du signal, commande des machines compliquées et le filtrage adaptatif [6].

Considérons un système d'inférence flou de type TAKAGI-SUGENO, nous supposons qu'il y a deux variables linguistiques d'entrées x_1 et x_2 et une sortie y , et supposons que la base des règles contient 2 règles de types :

$$R1 : \text{si } x_1 \text{ est } A_1 \text{ et } x_2 \text{ est } B_1 \text{ alors } y_1 = p_1 x_1 + q_1 x_2 + r_1$$

$$R2 : \text{si } x_1 \text{ est } A_2 \text{ et } x_2 \text{ est } B_2 \text{ alors } y_2 = p_2 x_1 + q_2 x_2 + r_2$$

Où x_1 et x_2 sont les variables d'entrées.

A_1, A_2, B_1 et B_2 sont des ensembles flous

y_i les sorties de tous les neurones de défuzzification p_i, q_i et r_i sont des paramètres du conséquent de la règle i .

JANG a proposé de représenter cette base de règle par le réseau adaptatif présenté par la figure suivante :

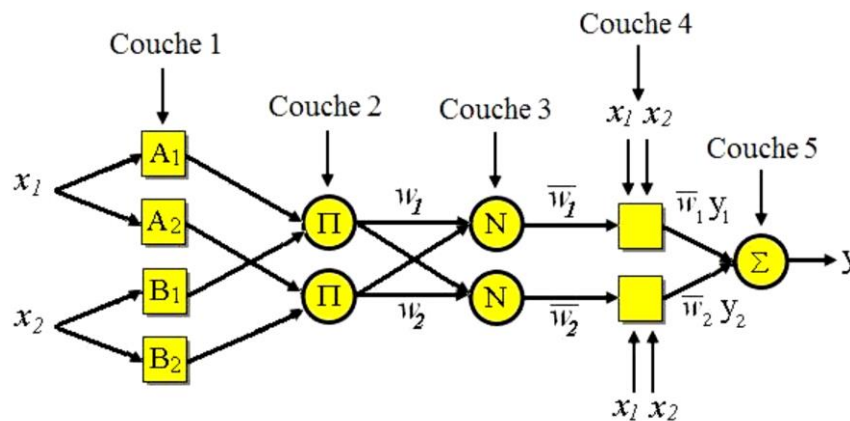


Figure 3.13 : Architecture d'ANFIS

Le réseau ANFIS est un réseau multicouches dont les connexions ne sont pas pondérées, ou ont toutes un poids égale à 1. Les nœuds sont de deux types différents selon leur fonctionnalité, ceux qui contiennent des paramètres (nœuds carrés) et ceux qui ne contiennent pas (noeuds circulaires). La sortie O_i^k du nœuds i de la couche k (appelé nœuds (i,k)) dépend des signaux provenant de la couche $k-1$ et des paramètres du noeud (i,k) , c'est-à-dire : [20]

$$O_i^k = f(O_i^{k-1}, \dots, O_{n-1}^{k-1}, p, q, r \dots) \tag{3.5}$$

Où n_{k-1} est le nombre de nœud dans la couche $k-1$.

• **Première couche (Fuzzification) :**

Chaque nœud de cette couche ait une fonction telle que :

$$O_i^1 = \mu_{A_i}(x_1) \text{ pour } i = 1,2 \tag{3.6}$$

$$O_i^1 = \mu_{B_{i-2}}(x_2) \text{ pour } i = 3,4 \tag{3.7}$$

Où x_1, x_2 : sont les entrées des nœuds respectifs (1,2) et (3,4) ;

A_i, B_{i-2} les termes linguistiques associées aux fonctions d'appartenance μ_{A_i} et $\mu_{B_{i-2}}$.

Les sorties O_i^1 de la première couche représente donc les degrés d'appartenance des variables d'entrée x_1, x_2 aux ensemble flous A_i, B_{i-2} .

Dans le modèle de JANG, les fonctions d'appartenance sont des gaussiennes, données par :

$$\mu_{A_i}(x) \text{ Ou } \mu_{B_i}(x) = \exp\left[-\frac{(x_i - c_i)^2}{a_i}\right] \tag{3.8}$$

Pour les quelles les paramètres (a_i, b_i, c_i) selon le choix sont désignés par le nom de paramètres de la prémisse (ou paramètres locaux du réseau) et définissent les fonctions d'appartenance correspondantes. En fait, toutes les fonctions continues et dérivables, telles les fonctions triangulaire ou trapézoïdale, peuvent être employées.

• **Deuxième couche (Règles floues) :**

La deuxième couche est formée d'un nœud pour chaque règle floue et génère les poids synaptiques. Ces nœuds de type fixe sont notés Π et chacun d'eux engendre en sortie le produit (opérateur ET de la logique floue) de ses entrées, qui ce qui correspond au degré d'appartenance de la règle considérée :

$$O_i^2 = w_i = \mu_{A_i}(x) \times \mu_{B_i}(x) \text{ pour } i = 1,2 \tag{3.9}$$

• **Troisième couche (Normalisation) :**

Les nœuds de cette couche sont également fixes et réalise la normalisation des poids des règles floues selon la relation :

$$O_i^3 = \bar{w}_i = \frac{w_i}{w_1 + w_2} \quad i = 1,2 \tag{3.10}$$

Chaque nœud i de cette couche est un nœud circulaire appelé N . La sortie du nœud i est le degré d'activation normalisé de la règle i .

- **Quatrième couche (Conséquence ou Défuzzification)**

Chaque nœud de cette couche est adaptatif et calcule les sorties des règles en réalisant la fonction :

$$O_i^4 = \bar{w}_i f_i = \bar{w}_i (p_i x_1 + q_i x_2 + r_i) \quad i = 1, 2 \quad (3.10)$$

Les paramètres $\{p, q, r\}$ sont les paramètres de sortie de la règle i .

- **Cinquième couche (Somme)**

La cinquième comprend un seul neurone qui fournit la sortie de ANFIS en calculant la somme des sorties de la couche précédente. Sa sortie qui est également celle du réseau est déterminée par la relation suivante :

$$O_i^5 = y = \sum_{i=1} \bar{w}_i f_i = \frac{\sum_{i=1} w_i f_i}{w_i} \quad i = 1, 2 \quad (3.11)$$

L'architecture d'ANFIS montre l'existence de deux couches adaptatives, la Première et la Quatrième.

La Première couche présente trois paramètres modifiables $\{a_i, b_i, c_i\}$ liées aux fonctions d'appartenance d'entrées, appelés paramètres des prémisses.

La Quatrième couche, contient également trois paramètres modifiables $\{p_i, q_i, r_i\}$ appelés paramètres conséquents.

III.4. Exemple d'application

On va refaire le même exemple que la logique floue type 1 et 2 avec une commande neuro-floue sans apprentissage.

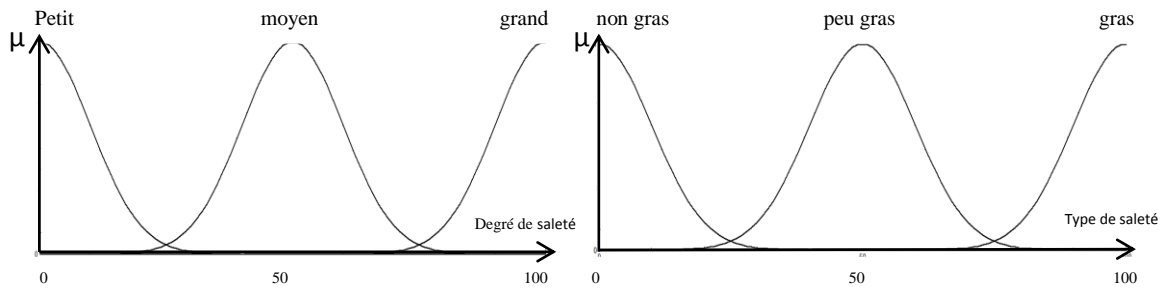
Donc les variables E/S de commande sont :

- Les entrées : Degré de saleté DS (Petit (P), Moyen (M), Grand (G))
Type de saleté TS (Non Gras (NG), Peu Gras (PG), Gras (G))
- La sortie : Durée de Lavage DL (Courte (C), Moyenne (M), Longue (L))

Après avoir défini les variables de commande du processus Degré de saleté, Type de saleté et Durée de lavage, on doit spécifier les sous-ensembles flous associés à ces variables ainsi que leurs fonctions d'appartenance.

On divise l'univers de discours de chaque variable en trois sous-ensembles flous.

Les figures suivantes montrent les fonctions d'appartenance des sous-ensembles flous de chaque variable linguistique sous la forme gaussienne.



La sortie doit être sous forme de fonctions linéaires pour la simplification on va les choisir comme des valeurs $y_1=10$, $y_2=30$, $y_3=60$.

Le tableau ci-dessous représente la base de règles (Matrice d'inférences)

Degré de saleté Type de saleté	P	M	G
NG	y_1	y_1	y_2
PG	y_1	y_2	y_3
G	y_1	y_3	y_3

➤ **Question**

Utiliser un contrôleur hybride neuro-flou pour déterminer la durée de lavage en minute pour laver un vêtement de type et degré de saleté est égales à 50 %.

➤ **Solution :**

1^{ère} couche : (fuzzification)

Pour le degré de saleté Pour Le type de saleté

$$O_1^1 = \mu_P (DS = 50\%) = 0$$

$$O_4^1 = \mu_{NG} (TS = 50\%) = 0$$

$$O_2^1 = \mu_M (DS = 50\%) = 1$$

$$O_5^1 = \mu_{PG} (TS = 50\%) = 1$$

$$O_3^1 = \mu_G (DS = 50\%) = 0$$

$$O_6^1 = \mu_G (TS = 50\%) = 0$$

2^{ème} couche : (Règles floues)

$$R_1 : \text{is DS est P et TS est NG Alors DL} = \mathbf{y1} \Rightarrow O_1^2 = \omega_1 = \mu_P (DS) \times \mu_{NG} (TS) = 0$$

$$R_2 : \text{is DS est P et TS est PG Alors DL} = \mathbf{y1} \Rightarrow O_2^2 = \omega_2 = \mu_P (DS) \times \mu_{PG} (TS) = 0$$

⋮

$$R_9 : \text{is DS est G et TS est G Alors DL} = \mathbf{y3} \Rightarrow O_9^2 = \omega_9 = \mu_G (DS) \times \mu_G (TS) = 0$$

3^{eme} couche : Normalisation

$$O_1^3 = \overline{\omega}_1 = \frac{\omega_1}{\sum_{i=1}^9 \omega_i} = 0$$

⋮

$$O_9^3 = \overline{\omega}_9 = \frac{\omega_9}{\sum_{i=1}^9 \omega_i} = 0$$

4^{eme} couche : défuzzification

$$O_1^4 = \overline{\omega}_1 * f_1 = 0 \times 10 = 0$$

⋮

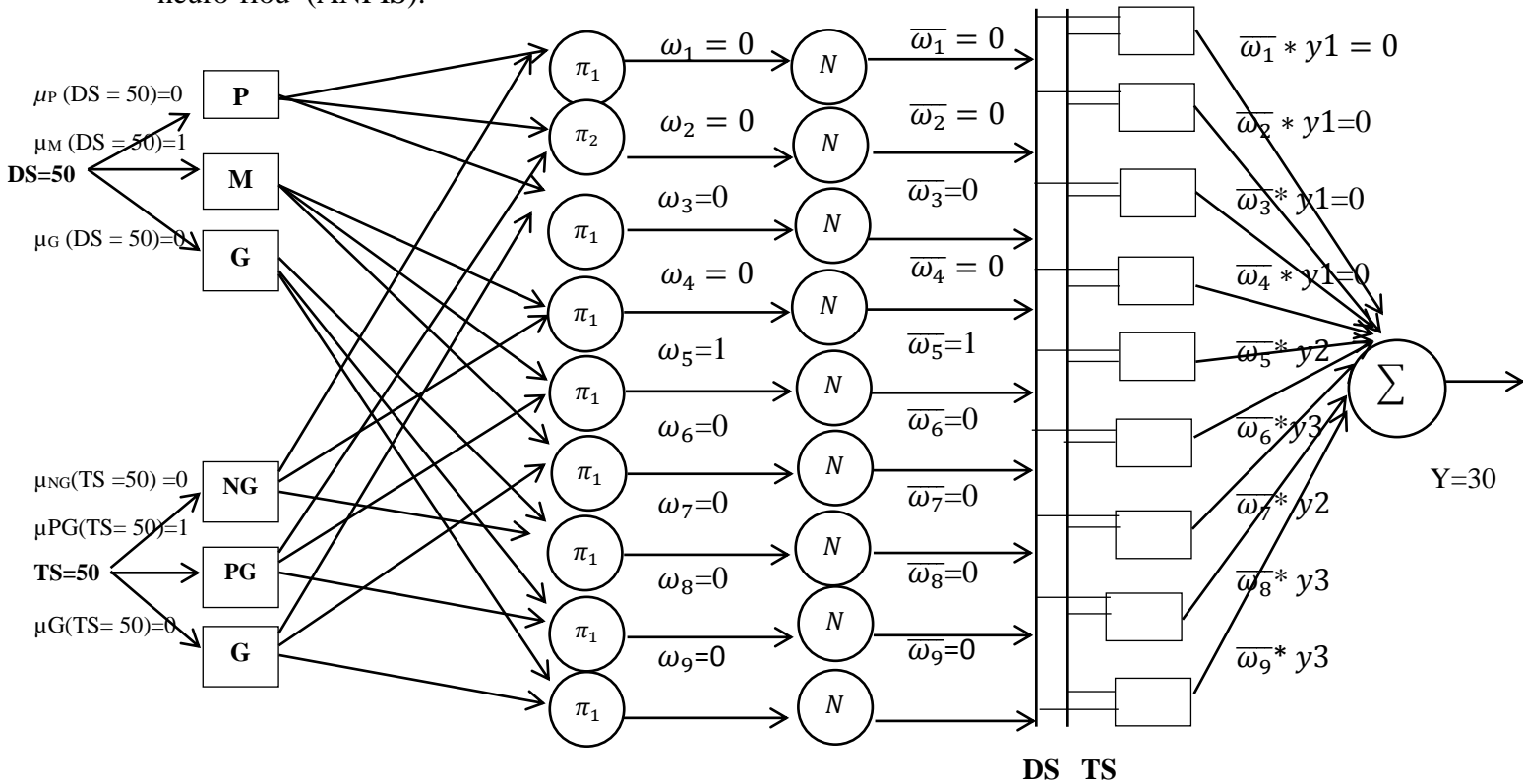
$$O_9^4 = \overline{\omega}_9 * f_3 = 0 \times 60 = 0$$

5^{eme} couche : sommation

$$O_1^5 = y = \sum_{i=1}^9 \overline{\omega}_i * f_i$$

Donc la durée effective de lavage est 30 min (moyenne durée)

La figure suivante présente en détaille les différentes étapes de structure générale de contrôleur neuro-flou (ANFIS).



III.5. Conclusion

Dans ce chapitre nous avons présenté les systèmes neuro-flous issus de la combinaison de la logique floue et les réseaux de neurones.

Les systèmes hybrides Neuro-floues sont utilisés principalement lorsque le modèle mathématique du système physique est difficile à élaborer, ils exploitent des règles floues tirées d'une expertise humaine pour modéliser le comportement dynamique du système. La commande à base de neuro-floue permet de définir une loi de commande numérique à partir des capacités d'apprentissage du réseau de neurone et de la lisibilité et la souplesse de la logique floue.

Le contrôleur neuro-flou peut être appliqué à une large classe de systèmes non linéaire, il combine les avantages de deux techniques, le réseau de neurone et la logique flou.

Chapitre 4

Application à la commande d'un pendule inversé

IV.1. Introduction	62
IV.2. Application de la commande floue type_1 sur le pendule inversé	62
IV.3. Application de la commande Neuro-floue sur le pendule inversé	70
IV.4. Application de la commande floue type 2 sur le pendule inversé	78
IV.5. Comparaison des résultats obtenus	85
IV.6. Conclusion	89

IV.1. Introduction

Le pendule inversé est un système instable, compliqué, et souvent utilisé dans l'automatique pour tester les lois de commande.

Dans ce chapitre, on a appliqué trois contrôleurs différents qui sont le contrôleur flou type-1, type-2, et ANFIS sur ce système pour comparer le plus efficace entre eux, et qui nous donne des bons résultats, nous avons utilisé dans notre contrôleurs deux entrées (l'angle et la vitesse angulaire) et une seule sortie (la commande de moteur) et nous présentons les résultats obtenus.

IV.1.1. Analyse de problème

L'objet de notre travail consiste à assurer la stabilité verticale du pendule inverse. Pour cela, il est nécessaire de gérer l'équilibre vertical du bras du pendule en tenant compte de la position du chariot sur son rail de guidage dont les dimensions mécaniques sont fixées.

En conséquence, il importe d'effectuer une commande du moteur en fonction de l'angle du bras du pendule donc assurer l'équilibre du pendule.

Les contrôleurs sont caractérisés par deux entrées (l'angle et la vitesse angulaire) et une seule sortie (la commande du moteur) [10].

Les dimensions de rail de guidage sont compris entre -100 cm et 100 cm.

La plage de variation de la vitesse linéaire est limitée entre -1m/s et 1m/s.

IV.2. Application de la commande floue type-1 sur le pendule inversé

IV.2.1. Enoncé de la méthode

La commande floue repose sur l'élaboration de règles décrivant du pendule. Un raisonnement physique appliqué à l'équilibre du bras du pendule, nécessite l'utilisation de deux informations, à savoir :

- ✓ La déviation de l'angle ;
- ✓ La vitesse de la déviation de l'angle.

Dans ce raisonnement, la position du chariot n'est pas prise en compte.

IV.2.2. Choix des éléments des régulateurs flous

IV.2.2.1. Choix de la forme des classes

Pour permettre une commande progressive et non pas par paliers, la forme triangulaire semble préférable pour toutes les classes des variables d'entrée et de sortie, ce qui permet également de simplifier les calculs.

IV.2.2.2. Définition des classes pour les deux variables d'entrées ($\theta, \dot{\theta}$)

Dans le but d'effectuer la transformation numérique symbolique, il est nécessaire de connaître la plage de variation réelle de chaque grandeur d'entrée afin d'y faire correspondre son ensemble flou ce dernier étant lui-même divisé en sous-ensembles flous. Nous avons donc défini 5 sous-ensembles flous(ou classes) respectivement nommés :

- Grand Négatif (GN)
- Petit Négatif (PN)
- Zéro(ZR)
- Petit Positif(PP)
- Grand Positif(GP)

➤ **Ecart angulaire**

Il est calculé par différence entre la consigne angulaire et l'inclinaison prise par le bras du pendule. Les limites physiques de la mesure de l'angle sont fixées entre -1 et +1 rad [10].

➤ **Vitesse angulaire**

Après des expériences de mesure sur le système utilisé, les limites de la vitesse angulaire été fixé, soit une vitesse angulaire entre -1 et +1 rad/s [10].

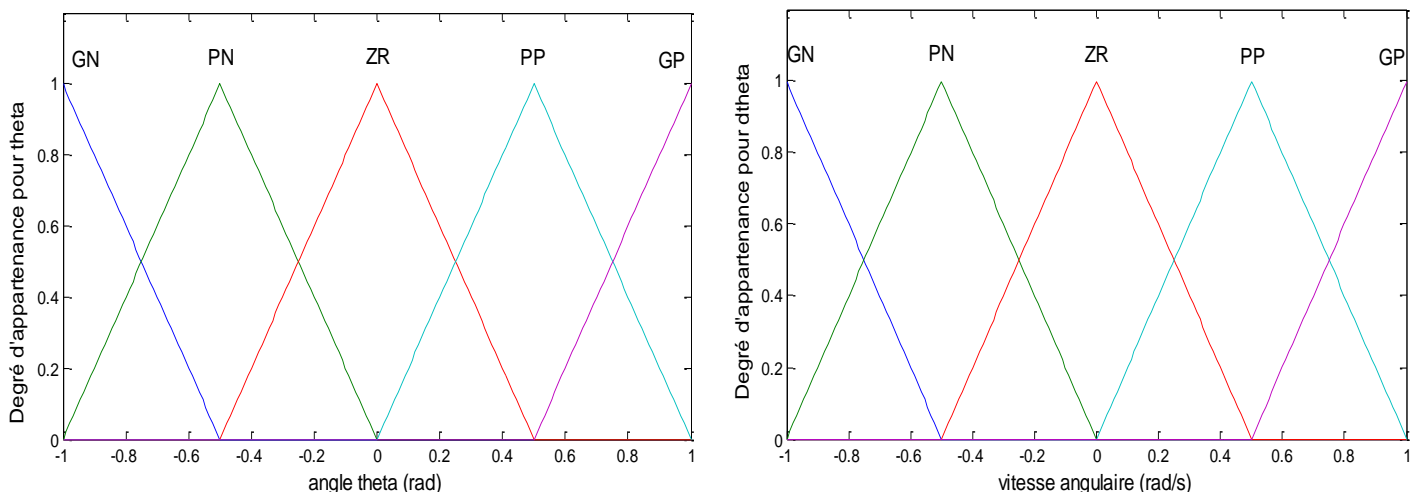


Figure 4.1 : définition des classes de l'angle et la vitesse angulaire du pendule

➤ Tension de commande du moteur

L'ensemble flou de cette grandeur doit s'adapter à la caractéristique du moteur utilisé, soit une tension évoluant entre 0 et +5 V.

IV.2.2.3. Définition des classes de sortie

La grandeur de sortie, exprimée sous forme symbolique à convertir en valeur numérique est la tension de commande de moteur fournie par le régulateur.

La forme de classes sera triangulaire et trapézoïdale, même le nombre de sous-ensembles flous reste à cinq (voire figure 4.2).

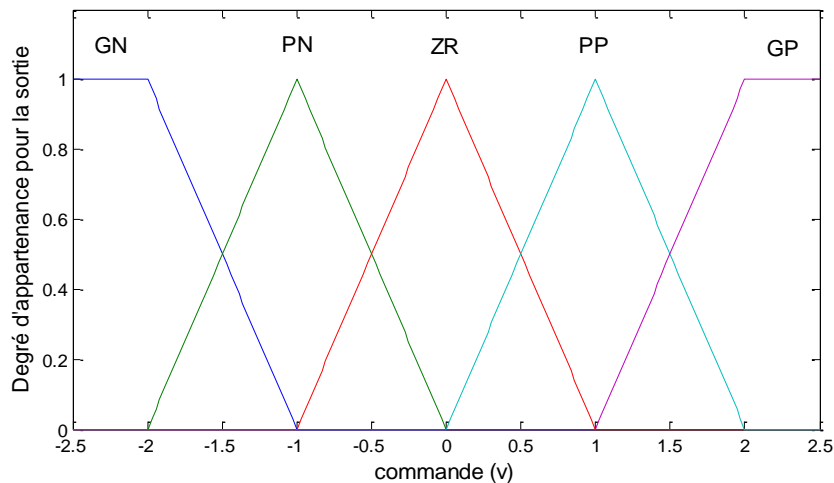


Figure 4.2 : Définition des classes de la commande

IV.2.2.4. Matrice d'inférence

La base de connaissance de contrôleur flou peut être synthétisée sous forme d'un tableau de règles où chaque règle associe sous forme conditionnelle les états des variables d'entrées à une caractéristique de la variable de sortie. Comme chacune des deux entrées est fuzzifiée en cinq classes, on obtient un jeu de 25 règles.

Les règles du contrôleur gérant la stabilisation de pendule sont simples à élaborer. Par exemple, si le bras tombe vers la droite très rapidement alors le chariot doit se déplacer très vite vers la droite pour rétablir l'équilibre. Le tableau suivant présente les règles de régulateur :

$\dot{\theta} \backslash \theta$	GN	PN	ZR	PP	GP
GN	GN	GN	GN	PN	ZR
PN	GN	GN	PN	ZR	PP
ZR	GN	PN	ZR	PP	GP
PP	PN	ZR	PP	GP	GP
GP	ZR	PP	GP	GP	GP

tableau 4.1 : la matrice d'inference du controleur flou type-1 [10]

Ceci peut se traduire par une règle exprimée sous la forme conditionnelle suivante :

Si l'angle θ est très positif entraînant un écart angulaire très négatif et que les variation angulaire est aussi très positive alors la force appliquée au chariot doit être très grande et dirigée vers la droite donc très positive.

IV.2.2.5. Mécanisme (le moteur) d'inférence

La loi de composition MAX-MIN d'une part pour sélectionner les règles au moyen du tableau correspondant et d'autre part faire l'agrégation des résultats flous de la variable de sortie associée à ce tableau.

IV.2.2.6. La défuzzification

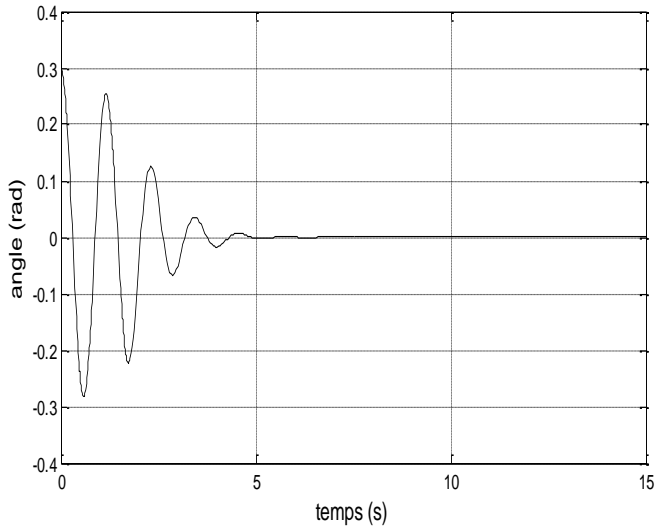
La méthode de centre de gravité simplifiée a été choisie pour sa rapidité de calcul sans risquer de la précision du résultat tolérable.

A partir de ces étapes du travail, nous pouvons présenter les résultats de simulation de la méthode de commande floue type-1 du pendule.

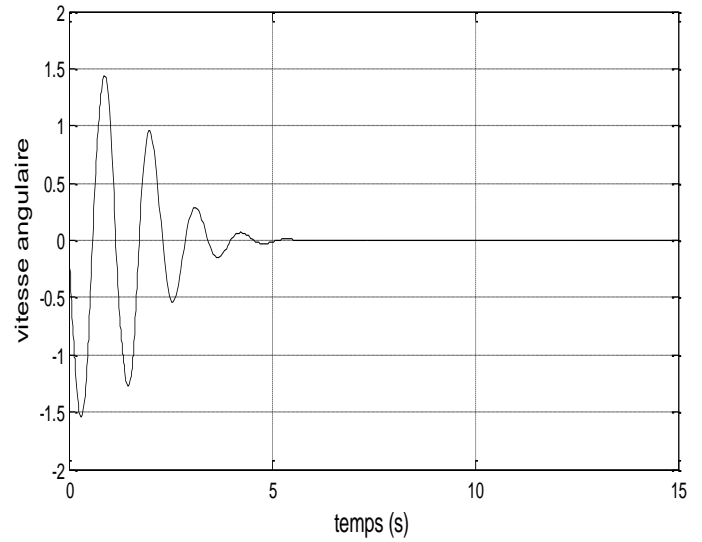
IV.2.3. Résultats de simulation

Les figures suivantes représentent la variation de la position x et de la vitesse linéaire \dot{x} du chariot ainsi que de l'angle θ et de la vitesse angulaire $\dot{\theta}$ du pendule par rapport à la verticale à conditions initiales données, pour la commande floue type-1.

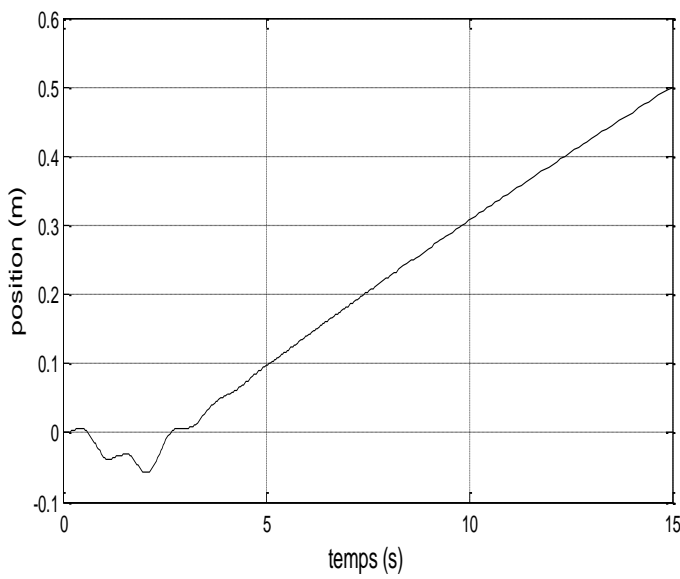
IV.2.3.1. Pour les conditions initiales $[x \dot{x} \theta \dot{\theta}] = [0 \ 0 \ 0.3 \ 0]$



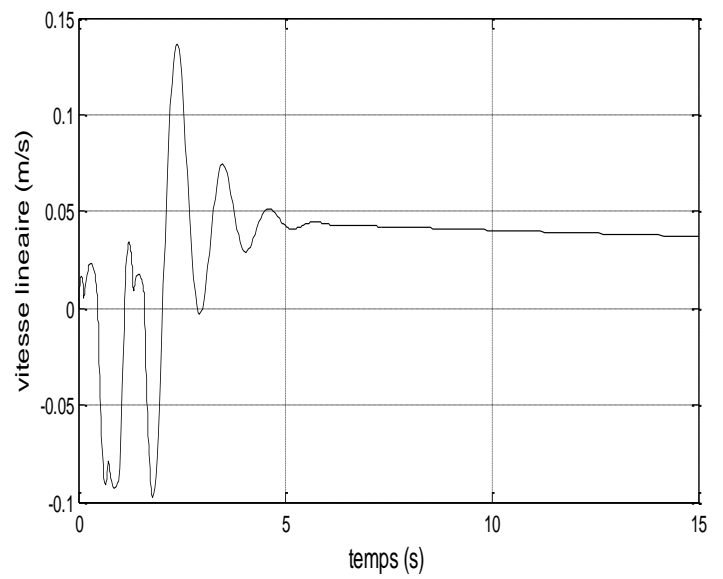
(a)



(b)



(c)



(d)

Figure 4.3 : Résultat de simulation par la commande floue type-1 pour des conditions initiales $[0 \ 0 \ 0.3 \ 0]$

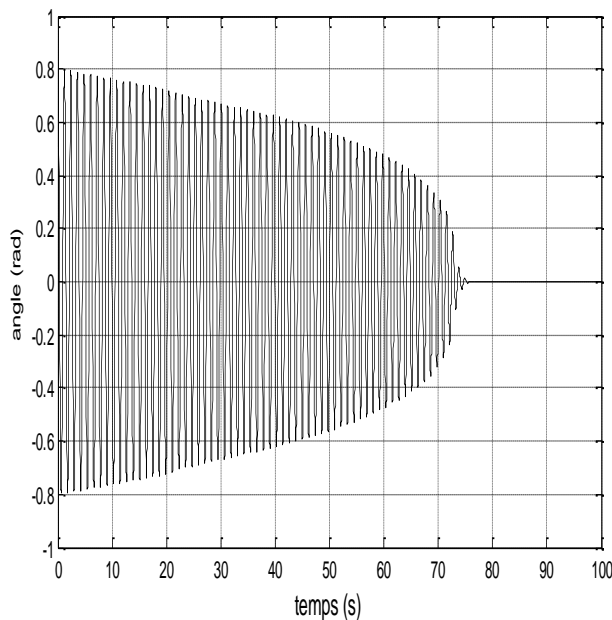
D'après les figures précédentes, on peut constater qu'avec un angle initial positif petit dans ce cas égale à 0.3 rad, le chariot doit faire un déplacement vers la droite pour rattraper le pendule, pour cela on remarque dans la figure « a » l'angle varie de 0.3 rad à -0.28 rad et continue à osciller autour de la référence 0 rad jusqu'à la stabilité avec un temps de réponse égale à 3 s avec un erreur statique nul.

La figure « c » représente le déplacement du chariot sur le rail de guidage, on remarque qu'il ne dépasse pas les limites physiques, et quand le pendule soit stable à 3s le chariot continue son mouvement mais lentement pour garder la position de pendule en état stable.

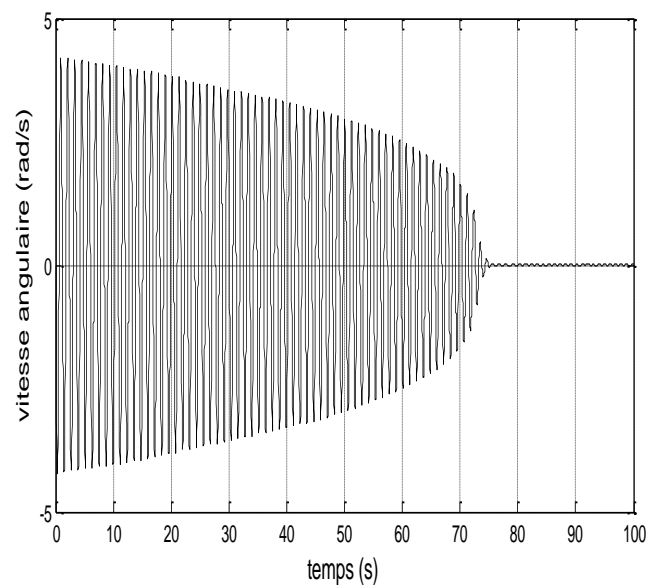
D'après ces résultats on peut dire que le contrôleur flou type-1 donne des bons résultats dans le cas où l'écart initial prend des petites valeurs.

La figure suivante représente la réaction de notre contrôleur flou type-1 face à une grande valeur d'écart initial.

IV.2.3.1. Pour les conditions initiales $[x \ \dot{x} \ \theta \ \dot{\theta}] = [0 \ 0 \ 0.8 \ 0]$



(a)



(b)

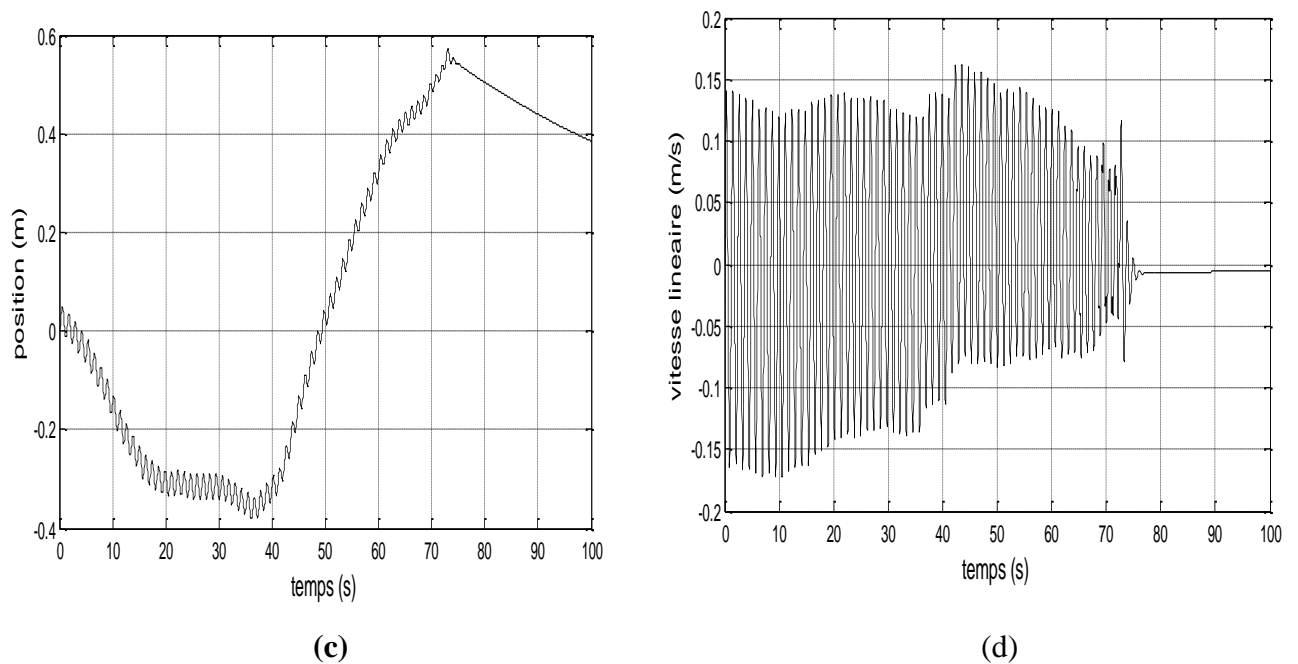


Figure 4.4 : Résultat de simulation par la commande floue type-1
pour des conditions initiales $[0 \ 0 \ 0.8 \ 0]$

D'après les résultats dans la figure 4.4 on remarque que :

- Temps de réponse est de 74s.
- Erreur statique égale à 0.
- La première oscillation est de 0.8 rad
- Le chariot ne dépasse pas les limites physiques de rail de guidage.

Dans la figure 4.4 (a) qui représente la variation de l'angle pour une condition initiale de $\theta = 0.8$ rad, et puisque on a limité la plage de variation de l'angle entre -1rad et 1rad dans notre contrôleur, cette valeur considéré grande, et cela interprété l'augmentation de temps de réponse par rapport au précédent.

On remarque que le nombre d'oscillations est très grand cela revient que le chariot fait plusieurs déplacement à droite et à gauche pour que le pendule atteindre sa stabilité.

Ces résultats de simulation montrent que le contrôleur flou type-1 dans le cas ou l'écart de l'angle est grand ne donne pas des bons résultats, grand nombre d'oscillations et temps de réponse important.

IV.2.3.3. Pour les conditions initiales $[x \ \dot{x} \ \theta \ \dot{\theta}] = [0 \ 0 \ -0.5 \ 0]$

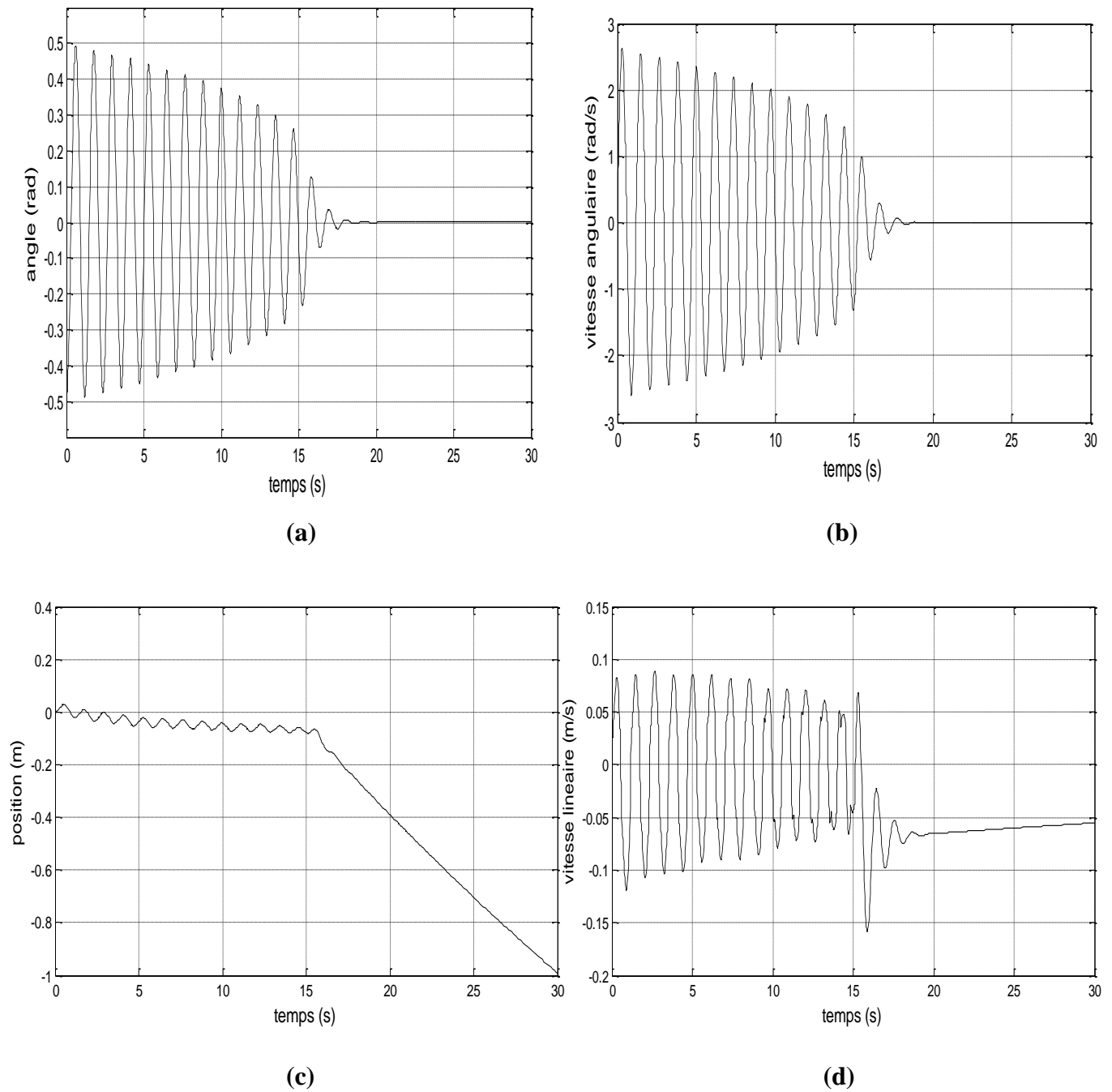


Figure 4.5: Résultat de simulation par la commande floue type-1 pour des conditions initiales $[0 \ 0 \ -0.5 \ 0]$

La figure 4.5 montre la bonne réponse de notre contrôleur avec un angle initial négative égale à -0.5 rad, on peut remarquer que la réaction du contrôleur aux valeurs initiales θ négatives est satisfaisante, avec un temps de réponse de 16s et un nombre d'oscillation acceptable (figure 4 (a)) par rapport au l'angle initial qui représente une valeur au milieu de plage de variation de l'angle θ .

La figure 4.5 (c) confirme que le chariot ne dépasse pas les limites physiques de rail de guidage si on applique un angle initial négatif.

On peut dire d'après ces résultats que le contrôleur flou type-1 est un bon choix dans les cas d'angles initiaux positifs.

IV.2.4. Interprétation des résultats de simulation dans le cas de FLC type 1

Après les résultats obtenus précédemment (écart initial petit, grand et négative), on peut conclure que la commande floue type-1 est très efficace notamment dans des petites valeurs de l'angle initial, et nous donne des résultats satisfaisants. Mais quand on augmente l'écart initial, cette commande devienne moins efficace avec un temps de réponse important de 74 s et nombre d'oscillation grand.

Lorsque l'angle θ prend des valeurs initiales négatives, le contrôleur flou type-1 donne des résultats acceptables au niveau de temps de réponse et la valeur de la première oscillation.

Dans la partie suivante on va passer au deuxième type de contrôleur c'est le contrôleur neuro-flou adaptatif (ANFIS).

IV.3. Application de la commande Neuro-floue

Dans cette partie, nous avons appliqué la commande neuro-floue sur le pendule inversé, ainsi nous avons exploité la capacité d'apprentissage du réseau de neurones, afin de produire la base de connaissances d'une manière automatique. Pour cela on a utilisé, un régulateur neuro-flou de type SUGENO, qui présente un avantage majeur (temps de calcul de la commande est réduit) par rapport à celui de type MAMDANI.

IV.3.1. Fonction d'appartenance

Dans notre cas on a deux entrées pour le régulateur, l'angle $\theta(t)$ et la vitesse angulaire $\dot{\theta}(t)$ du pendule, et une sortie qui est la commande du moteur (la tension).

La commande par ANFIS nécessite des entrées avec des fonctions d'appartenances gaussiennes, dont ils sont définis par leurs centres et écarts types. Par contre les conséquences sont des fonctions de forme $y = p\theta + q\dot{\theta} + r$.

Exemple :

Les règles de TS sont exprimées comme suit :

$$\text{Si } \theta \text{ est } A_1 \text{ et } \dot{\theta} \text{ est } B_1 \text{ alors } y_1 = p_1\theta + q_1\dot{\theta} + r_1.$$

$$\text{Si } \theta \text{ est } A_2 \text{ et } \dot{\theta} \text{ est } B_2 \text{ alors } y_2 = p_2\theta + q_2\dot{\theta} + r_2.$$

Les paramètres des fonctions d'appartenances des entrées de notre contrôleur neuro-flou sont généralement estimés par la méthode des moindres carrés qui est très compliquée et difficile. C'est pour ça, nous avons appliqué une autre astuce simple et efficace. Nous avons utilisé le ToolBox « fuzzy » du Matlab, pour convertir les fonctions d'appartenances triangulaires déjà utilisées dans la commande par logique floue type 1 vers des fonctions gaussiennes pour garder le même comportement du contrôleur.

Dans cette idée les centres des fonctions gaussiennes sont les sommets des fonctions triangulaires du FLC type_1 et l'écart type à une valeur fixe donnée par Matlab qui est 0.21 pour tous les ensembles flous.

Le tableau suivant représente les paramètres des fonctions d'appartenance des entrées :

Les paramètres	GN		PN		ZR		PP		GP	
	c	δ	c	δ	c	δ	c	δ	c	δ
θ	-0.66	0.21	-0.33	0.21	0	0.21	0.33	0.21	0.66	0.21
$\dot{\theta}$	-0.66	0.21	-0.33	0.21	0	0.21	0.33	0.21	0.66	0.21

Tableau 4.2 : Paramètre des fonctions d'appartenance de contrôleur ANFIS

La figure suivante représente les fonctions d'appartenance des entrées θ et $\dot{\theta}$

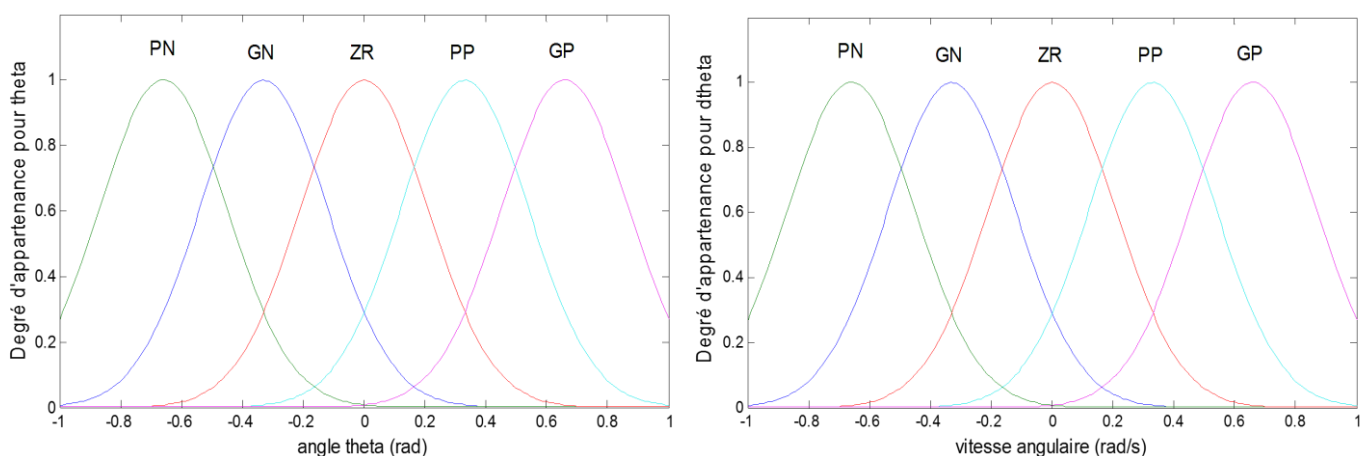


Figure 4.6 : Les classes de l'angle et la vitesse angulaire du pendule pour ANFIS

IV.3.2. Algorithme d'apprentissage

Nous avons utilisé l'algorithme de la descente de gradient uniquement dans la dernière couche.

IV.3.2.1. Les différentes étapes de fonctionnement

- La sortie de première couche représente le degré d'appartenance de chaque entrée, dans notre cas nous avons 10 degrés d'appartenances.
- La deuxième couche représente la phase d'interprétation de chaque règle, nous obtenons 25 sorties.
- La troisième couche représente la partie de normalisation en divisant chaque poids sur la somme de tous les poids.
- La sortie de la quatrième couche est exprimé par : $O_i^4 = \overline{\omega}_i(p_i\theta + q_i\dot{\theta} + r_i)$ alors dans cette étape nous avons 25 poids et la sortie de ce réseau est la somme de ces 25 poids.
- La sortie de la cinquième couche est la sortie du réseau qui représente la somme des 25 poids obtenus de la couche précédente.

L'apprentissage est donc fait dans la dernière couche comme suit :

Les paramètres des prémisses sont supposés fixes, donc O_i^5 peut être exprimé sous la forme

$$O_i^5 = \sum_{i=1}^{25} (\overline{\omega}_i \theta) p_i + (\overline{\omega}_i \dot{\theta}) q_i + \overline{\omega}_i r_i$$

Donc on a :

$$O_i^5 = (\overline{\omega}_1 \theta) p_1 + (\overline{\omega}_1 \dot{\theta}) q_1 + \overline{\omega}_1 r_1 + \dots + (\overline{\omega}_{25} \theta) p_{25} + (\overline{\omega}_{25} \dot{\theta}) q_{25} + \overline{\omega}_{25} r_{25}$$

C'est une combinaison linéaire des paramètres de conséquent modifiable p_i , q_i et r_i pour $i = 1, \dots, 25$.

Nous avons donc une somme de 75 termes, alors la dernière couche est un nœud de 75 entrées et une sortie.

C'est pour ça qu'on n'est pas besoin d'utiliser l'algorithme de rétro-propagation de gradient, parce qu'on a un seul nœud avec 75 entrées, pour faire l'apprentissage nous avons utilisé l'algorithme de la descente de gradient.

Les paramètres des conséquences obtenues après l'apprentissage sont représentées dans le tableau suivant :

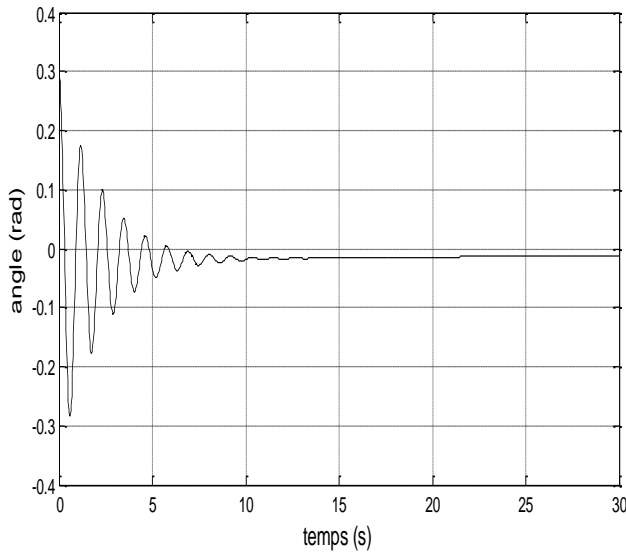
Règles	θ	$\dot{\theta}$	a	b	c
1	GN	GN	0.3819	0.5643	0.5282
2	GN	PN	-1.0990	0.9036	-0.3076
3	GN	ZR	0.0375	-0.2100	0.4020
4	GN	PP	-1.22384	1.2754	-0.2871
5	GN	GP	-0.1550	0.7130	0.8397
6	PN	GN	-0.3562	0.6147	-0.0110
7	PN	PN	-0.2726	0.2397	-0.0472
8	PN	ZR	0.2389	-0.0889	0.1119
9	PN	PP	0.3798	0.5519	0.1797
10	PN	GP	0.8637	0.6258	0.2077
11	ZR	GN	-0.6847	1.0613	-0.0703
12	ZR	PN	0.0319	-0.4731	0.1814
13	ZR	ZR	-0.9350	0.3343	-0.8489
14	ZR	PP	0.5132	0.5593	0.8888
15	ZR	GP	0.3603	0.3775	0.3822
16	PP	GN	-0.2939	-0.1990	0.2667
17	PP	PN	0.0990	-0.1461	0.1275
18	PP	ZR	0.1049	0.0452	0.4170
19	PP	PP	1.5562	0.8578	0.4061
20	PP	GP	0.2563	0.6218	0.0918
21	GP	GN	-0.1096	-0.0101	-0.0462
22	GP	PN	-0.1032	1.0483	-0.5471
23	GP	ZR	0.9238	0.7215	0.5852
24	GP	PP	0.0615	0.7278	0.3912
25	GO	GP	0.2687	0.2978	-0.2713

Tableau 4.3 : Base de règles obtenue après la descente de gradient

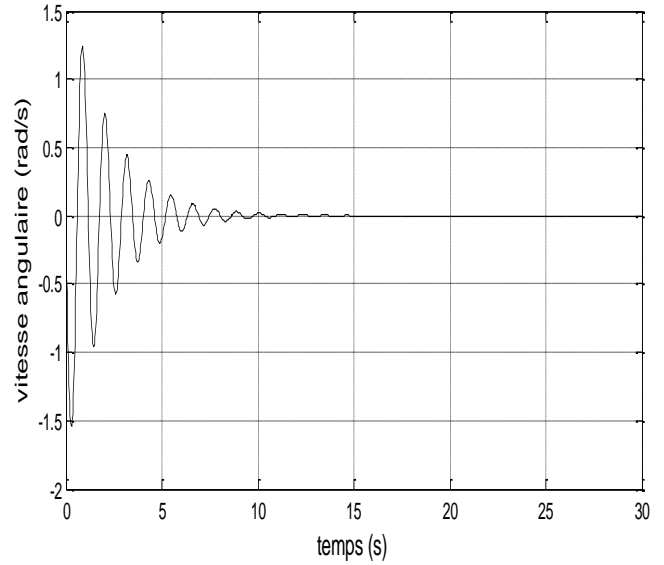
IV.3.3. Résultats de simulation

Les figures suivantes représentent la variation de la position et de la vitesse linéaire du chariot ainsi que de l'angle et de la vitesse angulaire du pendule par rapport à la verticale à conditions initiales données, pour la commande neuro-flou (ANFIS).

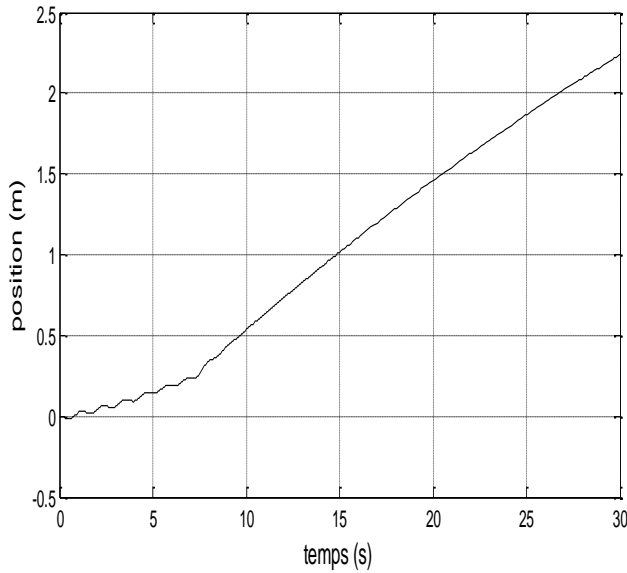
IV.3.3.1. Pour les conditions initiales $[x \dot{x} \theta \dot{\theta}] = [0 \ 0 \ 0.3 \ 0]$



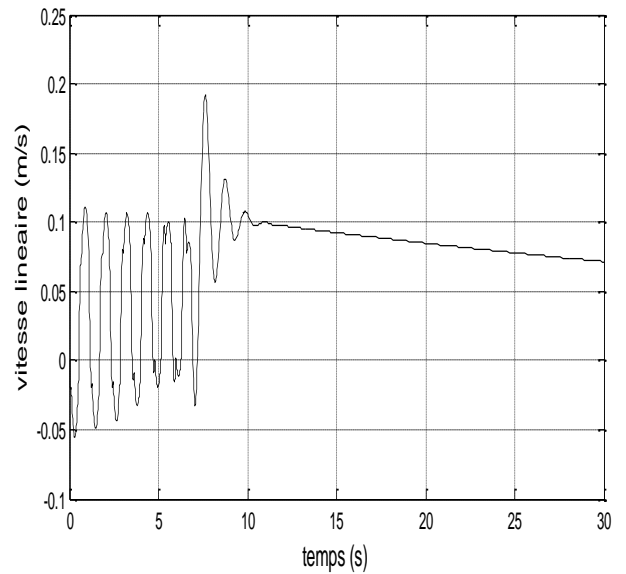
(a)



(b)



(c)



(d)

Figure 4.7 : Résultat de simulation par la commande ANFIS pour des conditions initiales $[0 \ 0 \ 0.3 \ 0]$

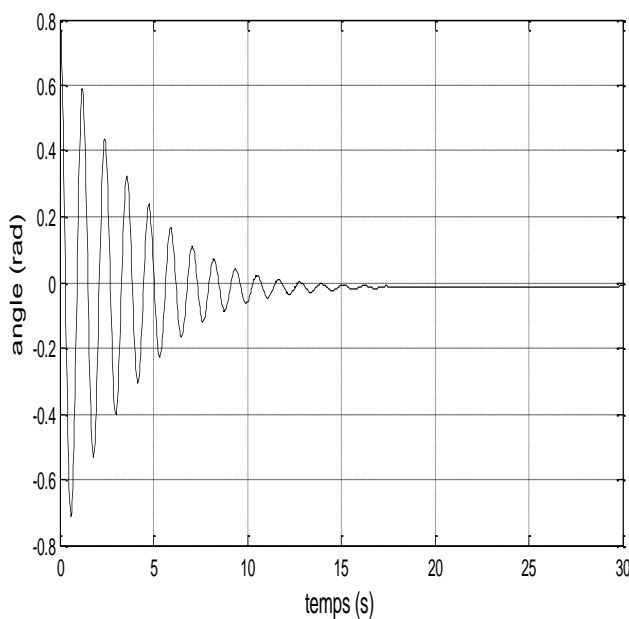
D'après les figures précédentes, on remarque que le contrôleur ANFIS montre la bonne performance avec un petit écart de l'angle θ [0 0 0.3 0], ceci est interpréter par le temps de réponse qui égale à 4.3 s et la première oscillation qui égale à 0.27 rad et un erreur statique nul (figure 4.7 (a)).

La figure 4.7 (c) confirme les bonnes résultats de contrôleur ANFIS, parce que le mouvement de chariot est souple et sans déplacement reste dans les limites physiques de raille de guidage.

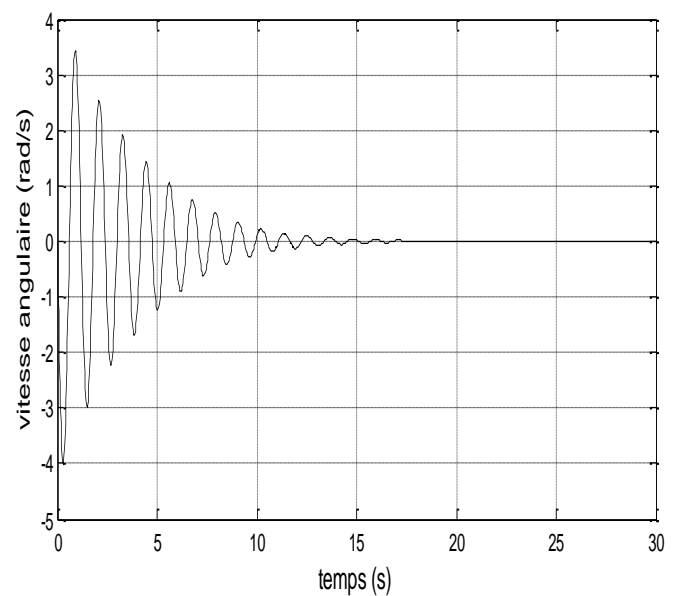
D'après ces résultats on peut dire que le contrôleur neuro-flou adaptatif ANFIS donne des très bons résultats dans le cas ou l'écart initial prend des valeurs petites.

La figure suivante représente la réaction de notre contrôleur neuro-flou adaptatif ANFIS face à une grande valeur d'écart initial.

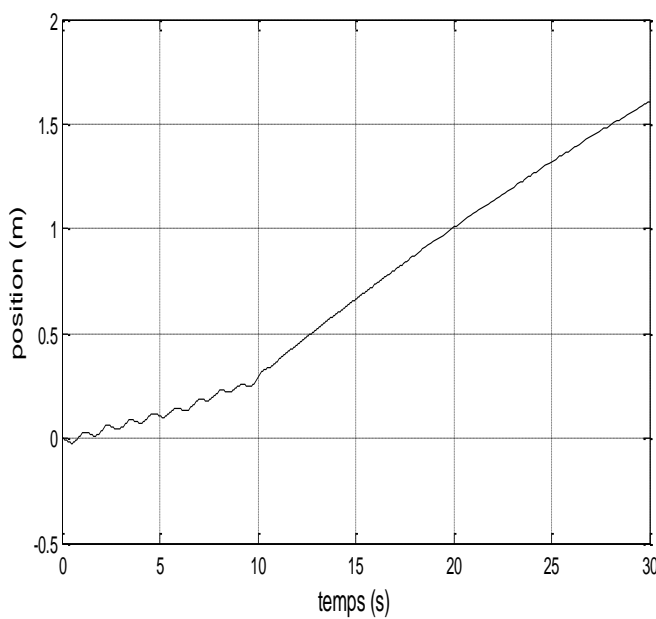
IV.3.3.2. Pour les conditions initiales $[x \dot{x} \theta \dot{\theta}] = [0 \ 0 \ 0.8 \ 0]$



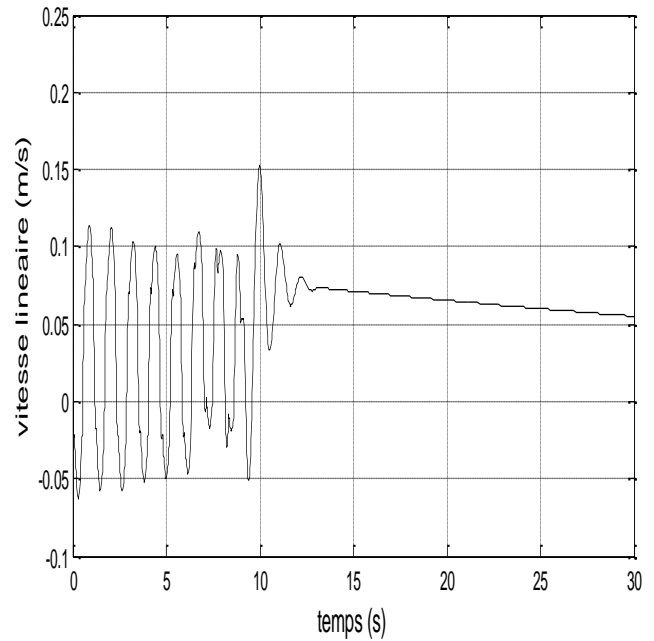
(a)



(b)



(c)



(d)

Figure 4.8 : Résultat de simulation par la commande ANFIS pour des conditions initiales $[0 \ 0 \ 0.8 \ 0]$

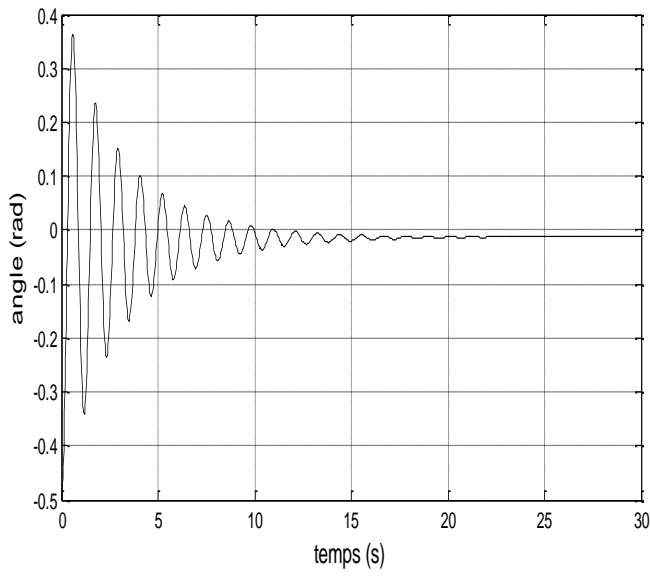
D'après les figures ci-dessus on constate :

- Le temps de réponse : 12s.
- Erreur statique égale à 0.
- La première oscillation : 0.68 rad.
- Le déplacement de chariot dépasse les limites physiques

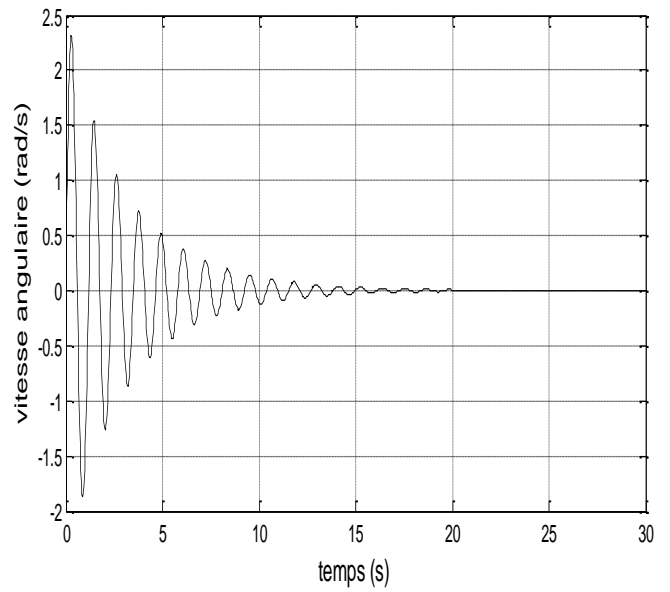
La figure 4.8 montre que le contrôleur ANFIS est un bon choix avec des grands écarts d'angle initial, dans ce cas 0.8 rad, ceci est interprétée par le petit temps de réponse par rapport à la grande erreur d'angle initial, et on peut aussi constater que le dépassement est très bon par rapport au grand écart à rattraper figure 4.8(a).

La figure 4.8(c) Montre qu'on a un dépassement des limites physiques dans le déplacement de chariot sur le rail de guidage et ça peut produire des problèmes au niveau de l'implémentation de contrôleur dans un système réel.

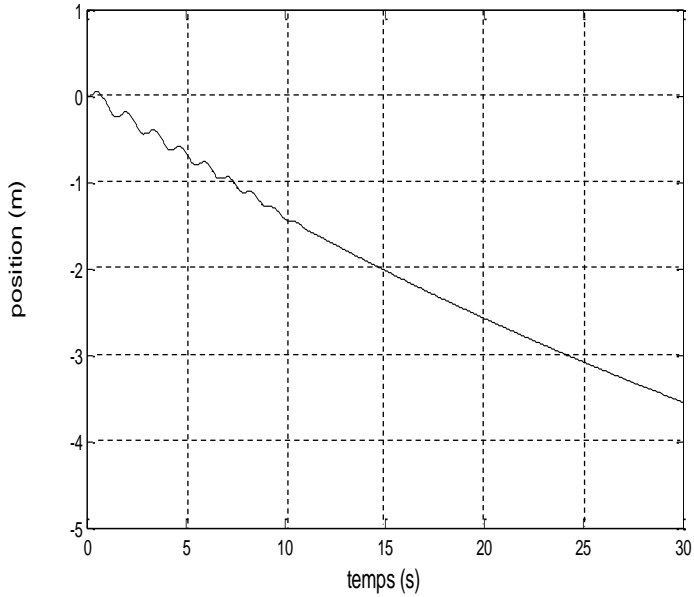
IV.3.3.3. Pour les conditions initiales $[x \ \dot{x} \ \theta \ \dot{\theta}] = [0 \ 0 \ -0.5 \ 0]$



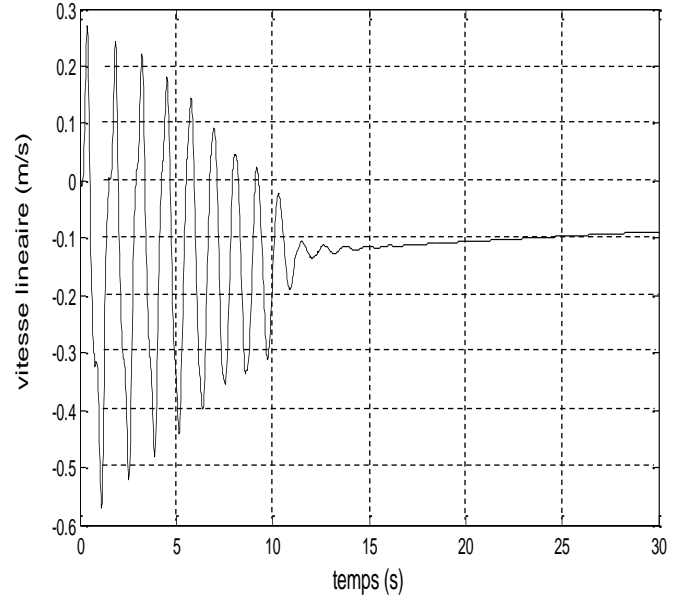
(a)



(b)



(c)



(d)

Figure 4.9: résultat de simulation par la commande ANFIS

pour des conditions initiales $[0 \ 0 \ -0.5 \ 0]$

D'après les figures ci-dessus on constate :

- Le temps de réponse est de 11s.
- Erreur statique égale à 0.
- Le dépassement est de 0.37 rad.
- Le déplacement de chariot ne dépasse pas les limites physiques

La figure 4.9 confirme que le contrôleur ANFIS est réagir parfaitement face aux valeurs négatives d'angle initiale et ça est clairement remarquer d'après les résultats obtenue dans le cas des conditions initiales [0 0 -0.5 0] avec un bon temps de réponse et le nombre d'oscillations réduit qui confirme la précision de notre contrôleur.

La figure 4.9 (c) confirme que le contrôleur ANFIS est un bon choix et ceci interpréter par le déplacement souple de chariot dans le rail de guidage sans dépassement des limites physiques.

IV.3.4. Interprétation des résultats de simulation

Après les résultats obtenus précédemment (écart initial petit, grand et négative), on peut conclure que la commande neuro-flou adaptative « ANFIS » est très efficace aux petites valeurs de l'angle initial, et nous donne des résultats satisfaisants, temps de réponse petit et aussi le bon déplacement de chariot sur le rail de guidage. Mais dans l'écart initial grand, cette commande nous donne des résultats acceptables au niveau de position de pendule, mais pour la position de chariot, on remarque que le chariot dépasse les limites physiques de rail de guidage.

Dans le cas où θ prend des valeurs négatives, le contrôleur ANFIS donne des résultats acceptables avec un petit temps de réponse ainsi que le déplacement de chariot reste sur les limites de rail de guidage.

Dans la partie suivante on va passer au troisième type de contrôleur qui est le contrôleur flou type-2.

IV.4. Application de la commande floue type 2 sur le pendule inversé

IV.4.1. Enoncé de la méthode

Reste identique que celle de type-1 avec une nécessité de deux informations (la déviation de l'angle et la vitesse de déviation de l'angle).

IV.4.1.1. Choix des éléments de régulateur flou type-2

La forme triangulaire reste préférable pour permettre la commande progressive pour les variables d'entrées, ce type prend en considération l'incertitude, et après des essais nous avons choisi la valeur $\alpha = 0.25$ comme intervalle d'incertitude (voir figure 4.10), qui nous donne des résultats acceptables.

Pour les fonctions d'appartenance de sortie nous avons utilisé des ensembles flous type-2 intervalles (voir figure 4.11), en gardant les mêmes centres qui sont les sommets des fonctions triangulaires de variable de sortie de contrôleur flou type-1, et les bornes égale à $[c - \alpha, c + \alpha]$; qui nous a donné des résultats satisfaisantes.

Le nombre des sous-ensembles flous reste cinq :

- Grand Négatif (GN)
- Petit Négatif (PN)
- Zéro(ZR)
- Petit Positif(PP)
- Grand Positif(GP)

IV.4.1.2. Matrice d'inférence

On garde les mêmes règles utilisé précédemment dans le contrôleur type-1 .

$\dot{\theta} \backslash \theta$	GN	PN	ZR	PP	GP
GN	GN	GN	GN	PN	ZR
PN	GN	GN	PN	ZR	PP
ZR	GN	PN	ZR	PP	GP
PP	PN	ZR	PP	GP	GP
GP	ZR	PP	GP	GP	GP

Tableau 4.4 : Matrice d'inférence pour le contrôleur flou type-2

IV.4.1.3. Les fonctions d'appartenance des entrées et de la sortie

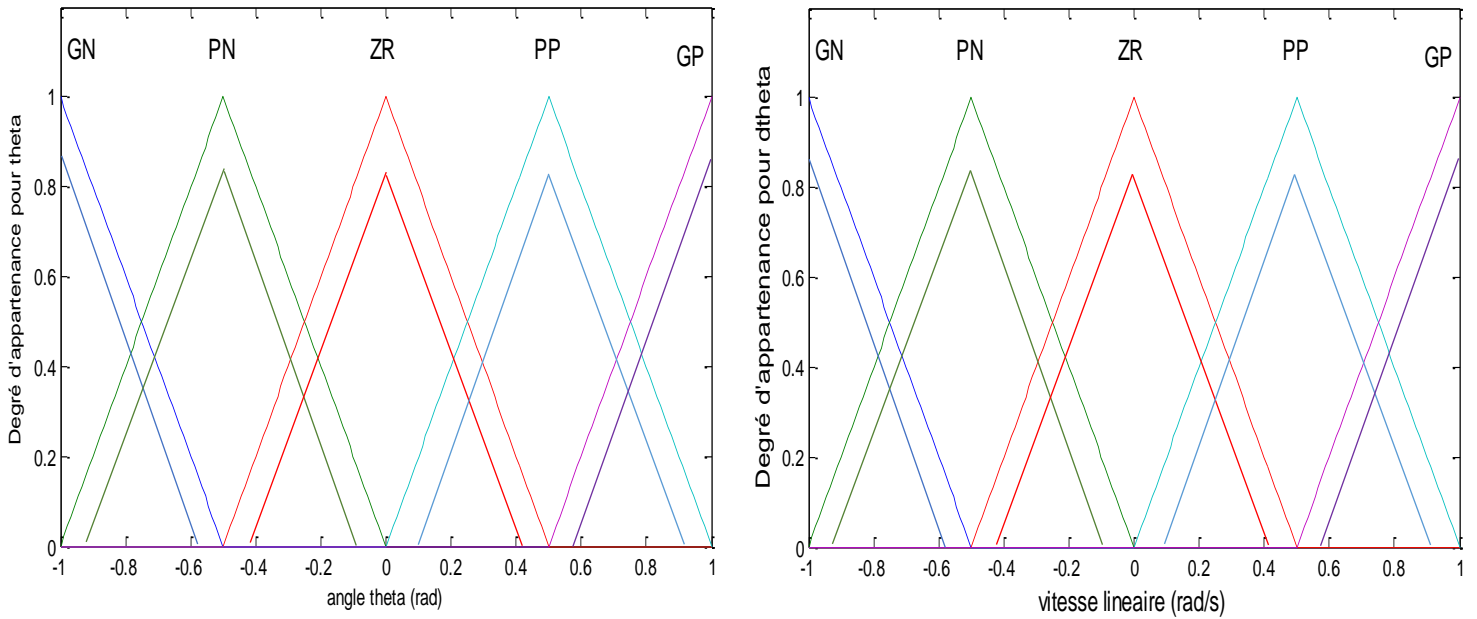


Figure 4.10 : Définition des classes de l'angle et la vitesse angulaire du pendule pour le régulateur flou type-2

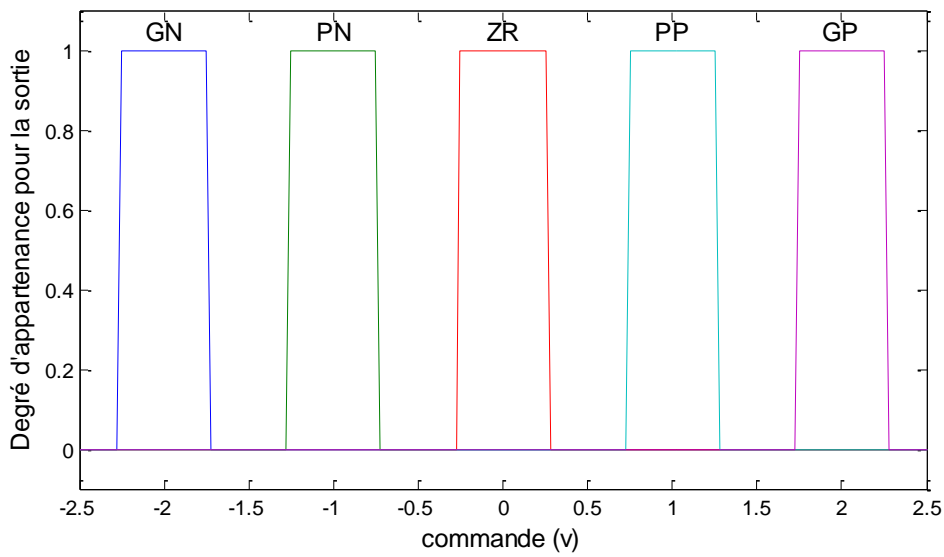
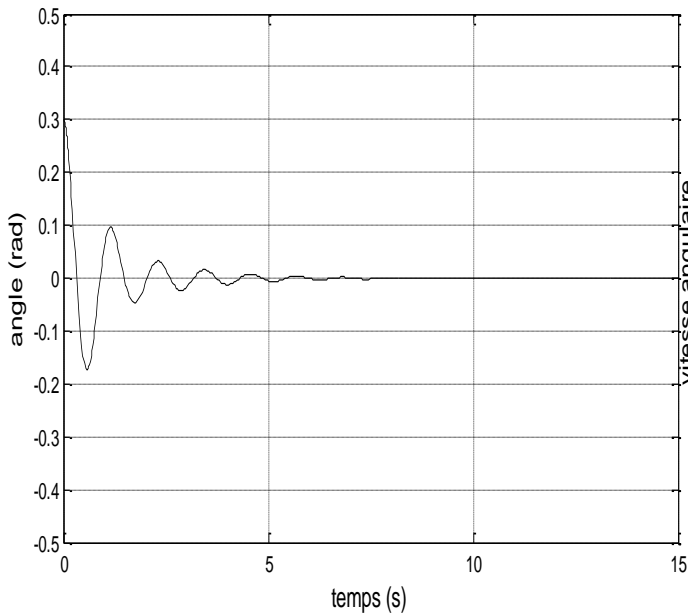


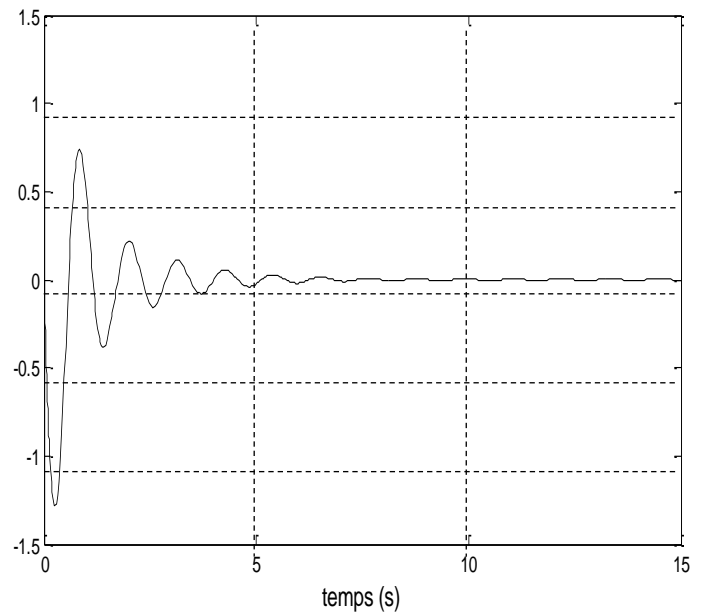
Figure 4.11 : Définition des classes de la commande pour le régulateur flou type-2

IV.4.2. Les résultats de simulation

IV.4.2.1. Pour les conditions initiales $[x \dot{x} \theta \dot{\theta}] = [0 \ 0 \ 0.3 \ 0]$



(a)



(b)

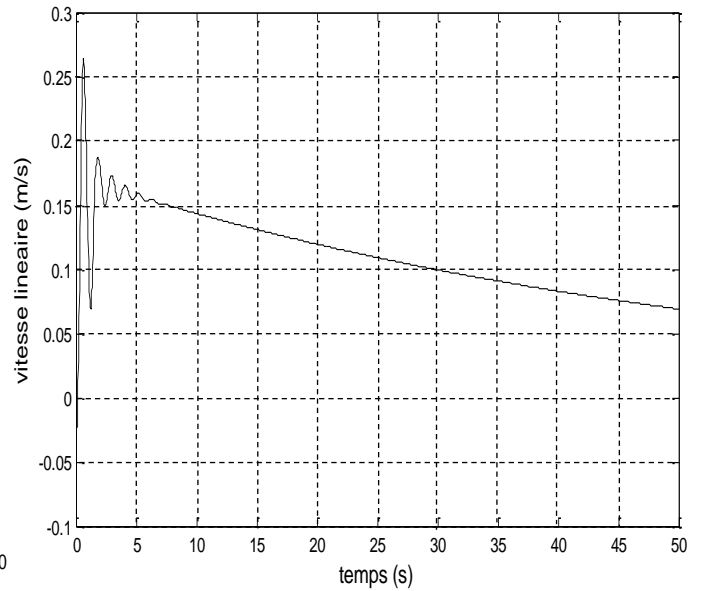
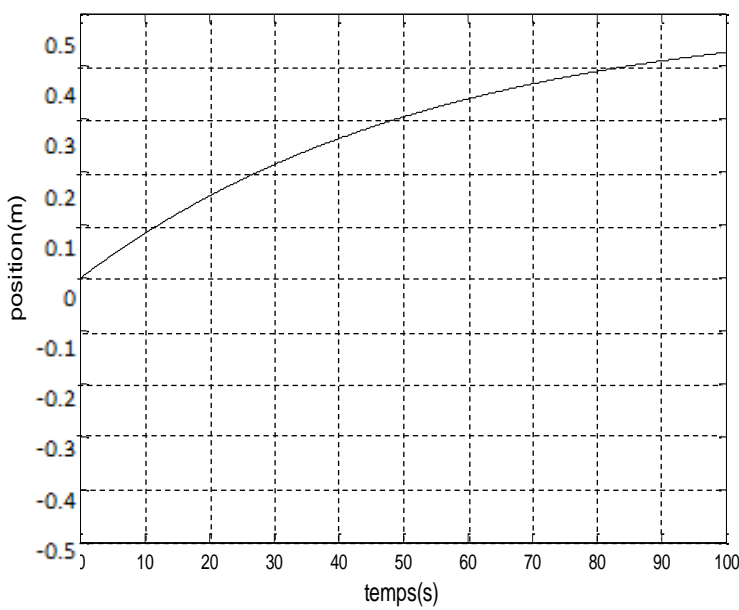


Figure 4.12 : Résultat de simulation par la commande floue type-2 pour des conditions initiales $[0 \ 0 \ 0.3 \ 0]$

D'après ces résultats on remarque que :

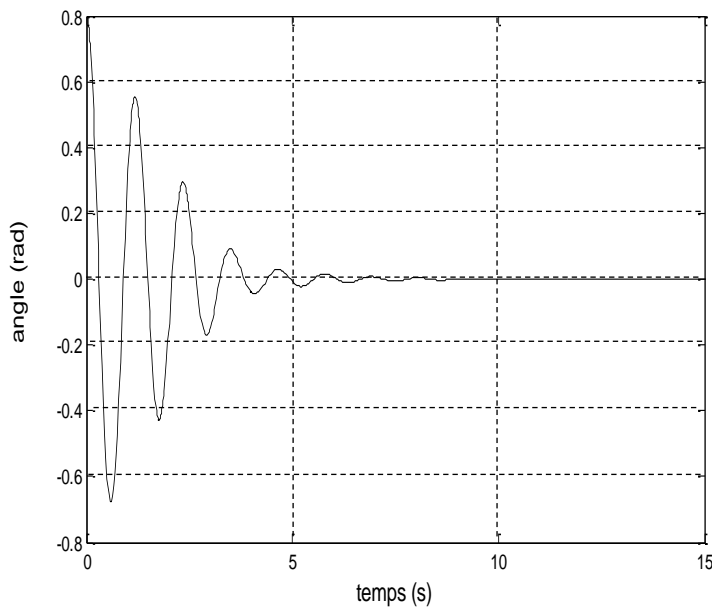
- Temps de réponse est de 1.5s.
- Erreur statique égale à 0.
- La première oscillation est de 0.17 rad.
- Un déplacement de chariot dans les limites physique.

La figure 4.12 représente un très bon résultat de contrôleur flou type 2, avec des conditions initiales $[0 \ 0 \ 0.3 \ 0]$, le contrôleur peut rattraper cet angle en temps de réponse très petit, et un nombre d'oscillation petit qui confirme la meilleur précision de notre contrôleur figure 4.12 (a).

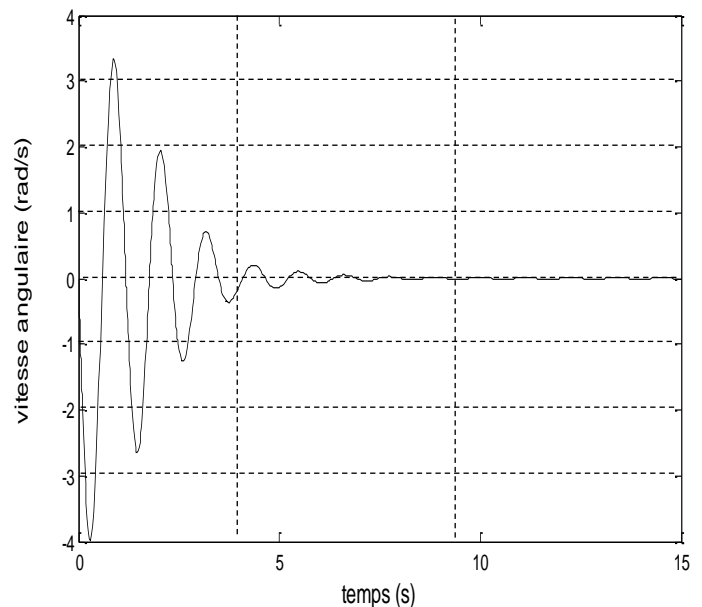
La figure 4.12 (c) confirme que le déplacement de chariot reste dans les limites physiques de rail de guidage avec un petit déplacement interpréter par le petit temps de rattraper la stabilisation du pendule.

D'après ces résultats de simulation on peut clairement remarquer que le contrôleur flou type-2 est un très bon choix quand on est dans le cas d'un petit écart initial.

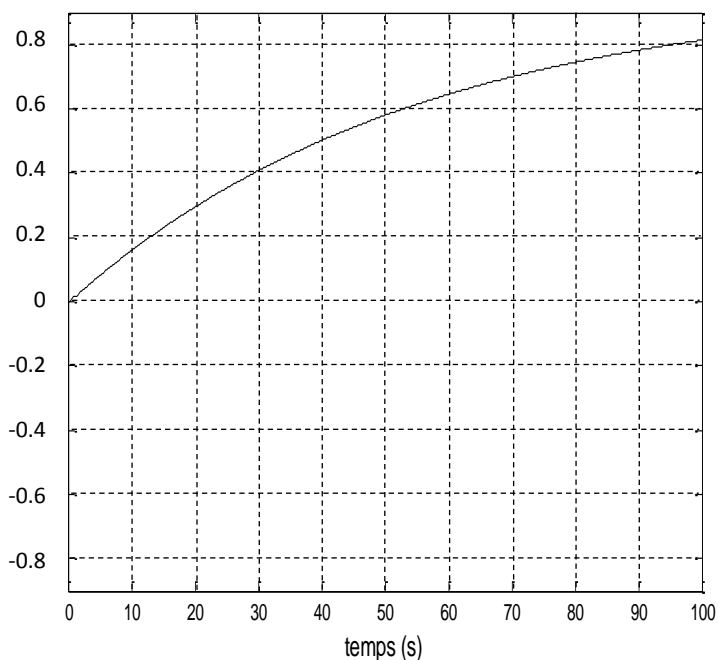
IV.4.2.2. Pour les conditions initiales $[x \ \dot{x} \ \theta \ \dot{\theta}] = [0 \ 0 \ 0.8 \ 0]$



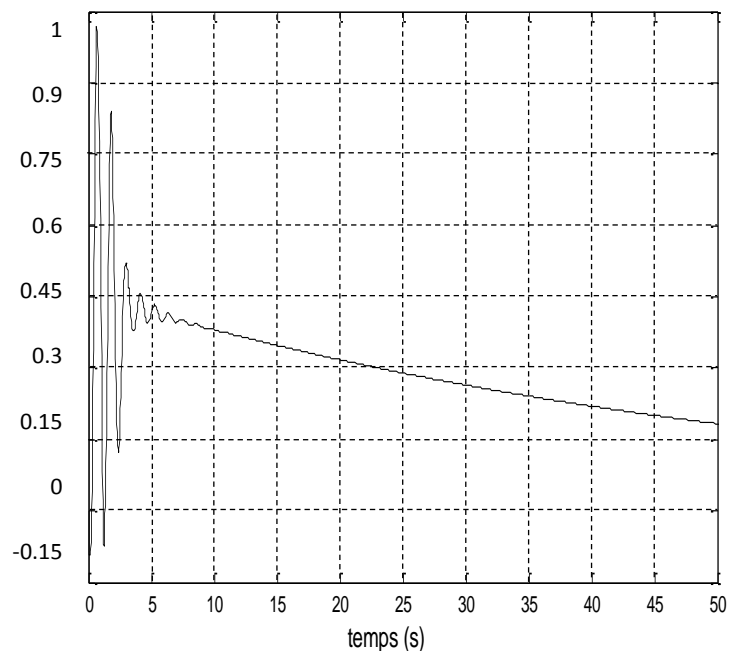
(a)



(b)



(c)



(d)

Figure 4.13 : Résultat de simulation par la commande floue type-2
pour des conditions initiales $[0 \ 0 \ 0.8 \ 0]$

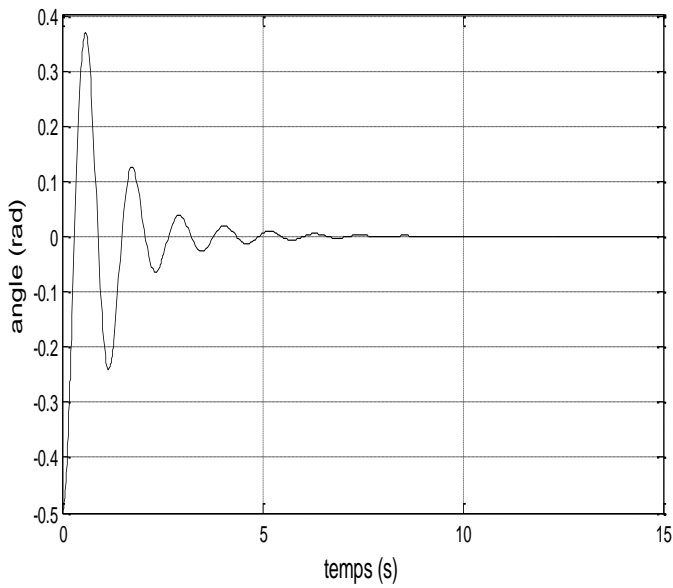
La figure précédente confirme que le contrôleur flou type_2 a réagi parfaitement avec une valeur d'écart initial grande, ceci expliquer par le très petit temps de réponse qui égale a 3.7 s et la première oscillation qui égale a 0.67 rad et un nombre d'oscillations très petit par rapport aux grand écart d'angle a rattraper, avec un erreur statique nul figure 4 (a).

La figure 4 (c) confirme la bonne performance de notre contrôleur floue type-2 avec des grandes valeurs d'angle initiale dans ce cas 0.8 rad avec un mouvement souple sans dépassement des limites physiques de rail de guidage.

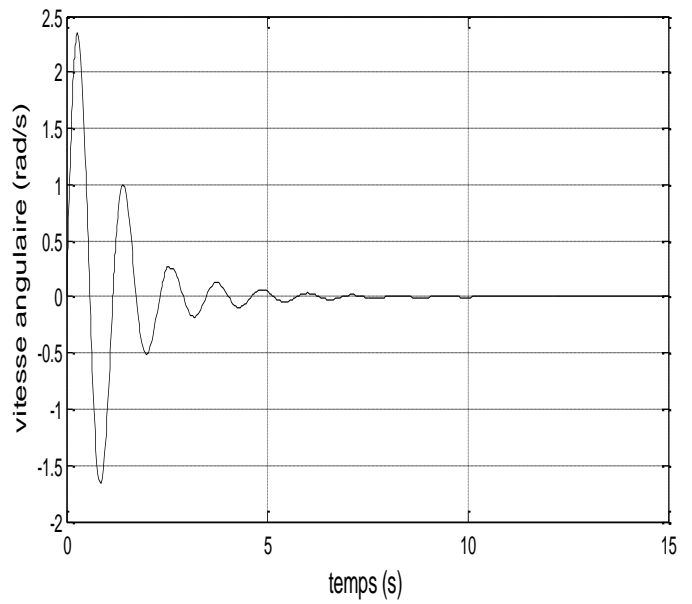
D'après ces résultats de simulation on peut clairement remarquer que le contrôleur flou type_2 est un très bon choix quand on est dans le cas d'un grand écart initial.

La figure suivante représente la réaction de notre contrôleur flou type_2 face à une valeur d'écart initial négative.

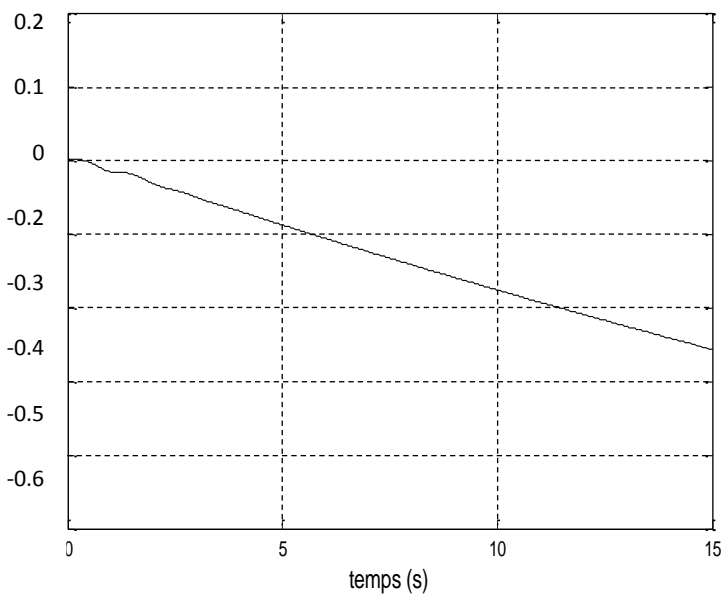
IV.4.2.3. Pour les conditions initiales $[x \dot{x} \theta \dot{\theta}] = [0 \ 0 \ -0.5 \ 0]$



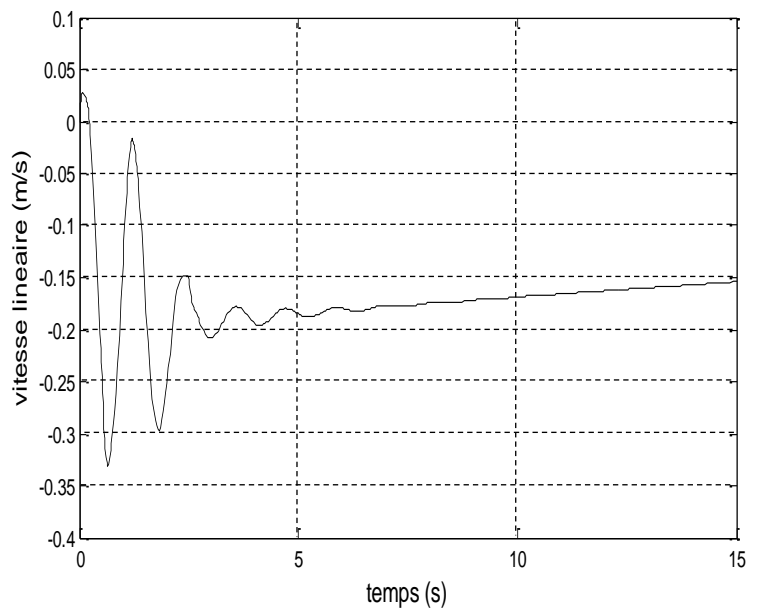
(a)



(b)



(c)



(d)

FigureIV.14 : résultat de simulation par la commande floue type_2

pour des conditions initiales $[0 \ 0 \ -0.5 \ 0]$

D'après ces résultats on remarque que :

- Temps de réponse est de 2.45s
- Erreur statique égale à 0.
- Un déplacement de chariot dans les limites physique.

La figure 4.14 représente la réaction de contrôleur flou type-2, avec un angle initial négatif, on peut dire que le contrôleur nous donne un très bon résultat à cause d'un très petit temps de réponse, et un erreur statique égale à 0, on peut même remarquer que le nombre d'oscillations est très petit par rapport à l'écart initial petit.

La figure 4.14 (c) représente les variations de position de chariot, qui nous montre que le chariot reste dans le rail de guidage avec un déplacement souple qui stabilise dans le côté gauche de rail de guidage.

Ces résultats de simulation confirment que le contrôleur flou type-2 peut s'adapter avec tous les cas des écarts initiaux est nous donne des résultats très satisfaisantes.

IV.5. Comparaison des résultats obtenus

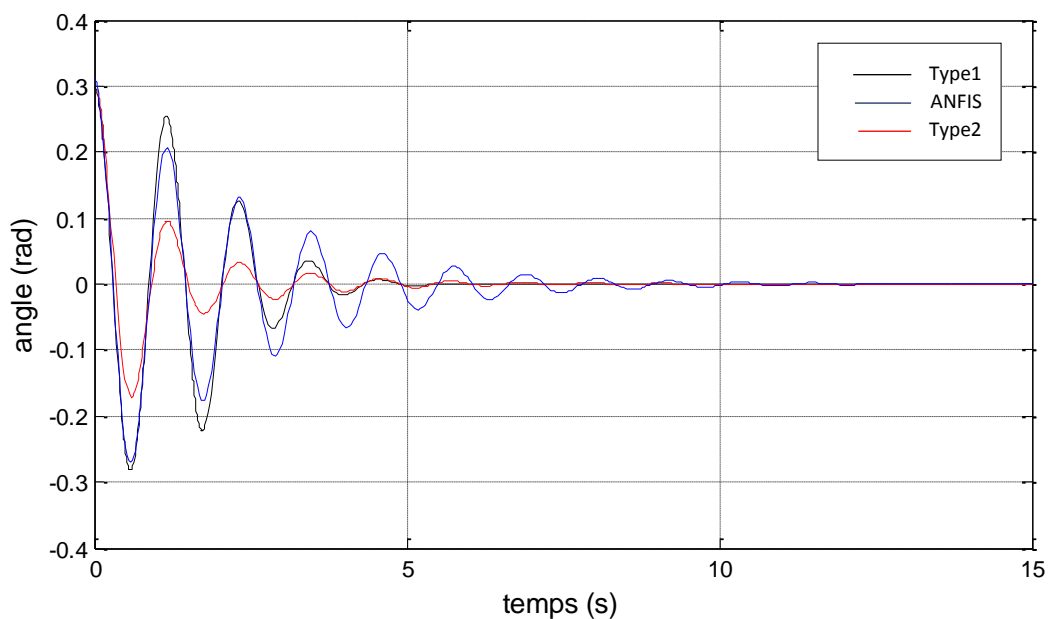


Figure 4.15 : Variation de l'angle θ pour $[0 \ 0 \ 0.3 \ 0]$

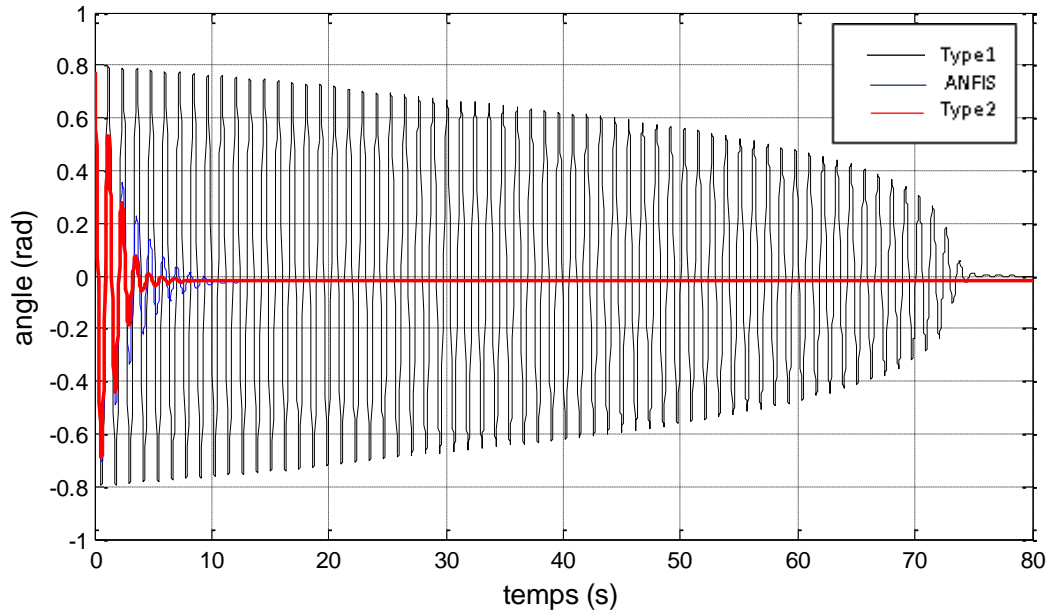


Figure 4.16 : Variation de l'angle θ pour $[0 \ 0 \ 0.8 \ 0]$

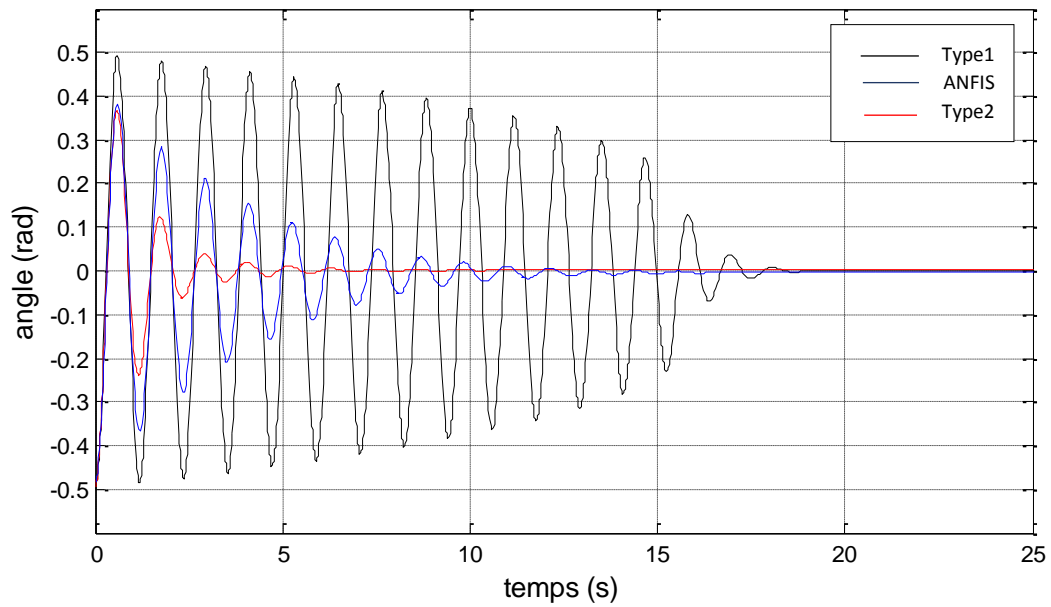


Figure 4.17 : Variation de l'angle θ pour $[0 \ 0 \ -0.5 \ 0]$

D'après les figures précédentes on remarque qu'avec un petit écart initial (figure 4.15), les trois contrôleurs, flou type_1, ANFIS et flou type_2 donne des bons résultats avec un avantage pour le contrôleur flou type_2. mais quand on augmente la valeur de l'écart initial (figure 4.16) le contrôleur flou type_1 donne des mauvaises résultats en comparant avec le neuro-flou et le flou type_2, et ceci interpréter par le grand temps de réponse par rapport aux deux autres contrôleurs, et pour cela on peut dire que le flou type_2 est un très bon choix par rapport aux deux autre contrôleurs. En passant aux valeurs initiaux négatives (figure 4.17), le contrôleur flou type_2 confirme que c'est le bon choix en comparant avec le type_1 et ANFIS, bien que les résultats obtenue avec ANFIS est considérer bonnes, mais les résultats de contrôleur flou type_2 était les plus satisfaisantes.

✓ **Pour les conditions initiales [0 0 0.3 0]**

	Type_1	ANFIS	Type_2
Temps de réponse	3 s	4.3 s	1.5 s
Dépassement	0.28 rad	0.27 rad	0.17 rad
Position de chariot	0.5 m	0.65 m	0.42 m

Tableau 4.5 : Résumé des résultats de simulation pour [0 0 0.3 0]

✓ **Pour les conditions initiales [0 0 0.8 0]**

	Type_1	ANFIS	Type_2
Temps de réponse	74 s	12 s	3.7 s
Dépassement	0.8 rad	0.68 rad	0.67 rad
Position de chariot	0.58 m	1.2 m	0.8 m

Tableau 4.6 : Résumé des résultats de simulation pour [0 0 0.8 0]

✓ Pour les conditions initiales [0 0 -0.5 0]

	Type_1	ANFIS	Type_2
Temps de réponse	16 s	11 s	2.45 s
Dépassement	0.6 rad	0.37 rad	0.37 rad
Position de chariot	-0.9 m	-0.45 m	-0.5 m

Tableau 4.7 : Résumé des résultats de simulation pour [0 0 -0.5 0]

D'après les tableaux précédents, on remarque que les résultats sont presque identiques entre les trois contrôleurs dans le cas d'un petit écart initial (Tableau 4.5) avec un petit avantage de type_2 au niveau des trois critères de comparaison (temps de réponse, première oscillation, position de chariot) et les deux autres contrôleurs donnent des résultats presque identiques avec un petit avantage d'ANFIS au niveau de temps de réponse.

Le Tableau 4.6 résume les résultats des contrôleurs pour grand angle initiale, la différence est claire entre les trois contrôleurs au niveau de temps de réponse, le contrôleur flou type_2 était très rapide à rattraper l'erreur initial, par contre le type_1 était trop long par rapport aux autres au niveau de dépassement, ANFIS et type_2 ont des mêmes résultats qui est meilleurs que type_1. Pour la position de chariot, le type_1 était le meilleur entre les trois contrôleurs avec un déplacement limites aux milieux de côté droite de rail de guidage, pour le ANFIS on remarque un dépassement des limites physique qui est un problème au système réel, pour le type_2, le déplacement est aux l'extrémité de rail de guidage.

Le tableau 4.7 représente le résumé des résultats de trois contrôleurs pour l'écart initial négative, pour une autre fois le contrôleur flou type_2 donne des bonnes résultats dans tous les critères de comparaison pour les amplitudes des oscillations le type_2 était le plus rapide par rapport aux autres avec un avantage de ANFIS en comparant avec le type_1, pour le dépassement, le type_2 et ANFIS était les plus précise et fort à rattraper l'écart en comparant avec le type_1 qui est le plus long entre eux. Pour la position de chariot, les trois contrôleurs ont presque les mêmes résultats.

IV.6. Conclusion

Dans ce chapitre nous avons présenté les trois type de commande qu'on est étudiées dans ce travail.

Nous avons d'abord présenté la méthodologie complète de la construction de la relation entrées sortie à partir des règles floues, ainsi la stratégie de commande floue type-1 utilisé dans notre travail.

Par la suite le modèle neuro-flou ANFIS utilisé été décrit d'une manière détaillées avec l'algorithme d'apprentissage utilisé dans ce travail qui est l'algorithme de la descente de gradient.

Dans la troisième partie nous avons présenté notre contrôleur flou type-2 et la méthodologie complète de construction ainsi les fonctions d'appartenances utilisés.

Enfin nous avons présenté les résultats de simulation obtenue par l'application de la commande floue type-1 et la commande neuro-flou et flou typet-2 respectivement.

Nous avons vu que le contrôleur floue type-2 manifeste de très bonnes performances en comparant avec les deux autre contrôleurs et d'après les résultats obtenue avec trois cas des écarts initiaux, on peut dire que la commande floue type-2 n'est pas juste une extension, mais c'est une amélioration de la commande floue type-1 dans le cadre de simulation.

Conclusion Générale

L'étude présentée dans ce mémoire a pour objectif de comparer trois types de commande d'intelligence artificielle, et montre qu'ils peuvent être utilisés pour commander des systèmes non-linéaires.

Les lois de commande proposées sont appliquées à la stabilisation d'un pendule inversé, qui est un système instable, SIMO, sous actionné, et non-linéaire. C'est un outil didactique et un problème classique souvent utilisé pour tester les caractéristiques des nouvelles lois de commande.

Ce travail est réalisé en trois phases :

Dans le premier chapitre, nous avons commencé par aborder en détail la description et la modélisation de pendule inversé, donc on a abouti à un modèle non-linéaire multi variable.

Dans le deuxième chapitre, nous avons présenté la théorie de base de la logique floue type 1 et 2, où plusieurs points ont été traités : définitions, opérations sur les ensembles flous, ainsi nous avons étudié la structure des contrôleurs flous type1 et type2.

Dans le troisième chapitre, nous avons présenté les réseaux de neurones, ainsi la commande neuro-floue qui réunit la théorie puissante de la logique floue et celle des réseaux de neurones.

Dans le dernier chapitre, nous avons présenté les résultats de simulations des trois contrôleurs. Dans la première partie du chapitre, le régulateur flou type-1 avec des fonctions triangulaire et trapézoïdale, et une taille de base de 25 règles, est appliqué ; il a donné des bons résultats pour des petits angles ; mais pour un angle grand nous n'avons pas pu obtenir le résultat attendu.

Dans la deuxième partie, la commande hybride neuro-floue (ANFIS) a été appliquée ; la combinaison de la logique floue et les réseaux de neurones a donné des résultats meilleurs que ceux de type 1, nous a apporté la rapidité par un meilleur temps de réponse surtout dans le cas où l'angle initial est grand.

Dans la troisième partie, nous avons appliqué une commande par un régulateur flou type-2, toujours avec une taille de base de 25 règles, des fonctions d'appartenance triangulaires avec un intervalle des incertitudes. Cette commande donnée en simulation des résultats satisfaisants surtout en niveau de temps de réponse.

Dans la dernière partie, nous avons fait une étude comparative des trois techniques, qui ont assuré la stabilité du système, mais de point de vue de rapidité, la commande par le régulateur flou type 2 est meilleur par un temps de réponse petit par rapport aux 2 autres techniques.

Donc, les résultats de simulation ont montré la supériorité des régulateurs flous type-2 par rapport aux régulateurs flous type-1 et ANFIS, pour ce type des systèmes.

Les résultats obtenus lors de notre travail, encouragent la poursuite des recherches dans ce sens, nous pouvons aussi envisager les points suivants :

- Application de la commande floue type 2 sur un système réel.
- La commande par neuro-floue type 2.
- Application de la commande floue type 2 sur un pendule inversé doublé.

Références bibliographiques

- [1] Abdelghani El Ougli : ‘Intégration des techniques floues à la synthèse de contrôleurs adaptatifs’, mémoire de doctorat national en Automatique, Signaux et Systèmes, Université Sidi Mohamed Ben Abdellah Fès, 2009.
- [2] Antoine Cornuéjols : ‘Introduction à la logique floue’, présentation, janvier 2008.
- [3] Arnold KAUFMANN : ‘Introduction à la logique floue’, Techniques de l’Ingénieur, traité Informatique industrielle, Aout 2001.
- [4] Borne P et Haggege J : ‘les réseaux de neurones présentation et applications’, Editions Technip, Paris , 2007.
- [5] C.L. Tsung : ‘Observer-based robust adaptive interval type-2 fuzzy tracking control of multivariable nonlinear systems’, Engineering Application of artificial intelligence, vol.23, issue.3, page 386-399, 2010.
- [6] CHEKROUN Soufyane : ‘Commande neuro-floue sans capteur de vitesse d’une machine asynchrone triphasée’, mémoire de magister en électrotechnique, Ecole Normale Supérieure d’Enseignement Technologique d’Oran, Octobre 2009.
- [7] Emira NECHADI : ‘Commande et stabilité d’un système électro- énergétique’, mémoire de Doctorat en sciences, Automatique, Université Ferhat Abbas de Setif, 21 / 03 / 2013.
- [8] Fabien M , ‘Introduction aux réseaux de neurones et leurs application en Automatique’, Ecole des Mines de Paris, Décembre 2003.
- [9] Faucher J , ‘les plans d’expériences pour le réglage et commande à base logique floue’ Thèse de Doctorat Spécialité Génie Electrique et Automatique, Institut national polytechnique de Toulouse.
- [10] Ferhat Lahouazi : ‘ mise en œuvre d’une stratégie de commande neuro floue’, application à un Pendule inversé, mémoire de magister en Automatique, Université Mouloud Mammeri, Tizi-Ouzou, 16.03.2011.
- [11] Fouad Alouani : ‘Commande par Logique Floue Appliquée aux pendules inversés et au Simulateur de Vol d’Hélicoptère : Simulation et Expérimentation’ magister en Electronique, Option Control, Université de Mohamed Boudiaf de M’sila, juin 2006.
- [12] Gacogne . Louis, ‘Eléments de logique floue’, Paris, Hermès, 1997.
- [13] HAMIZI Abdelhakim : ‘Commande Adaptative Floue Type-2 d’un bras Manipulateur’, mémoire de magistère en électronique option robotique, Université de Batna, 2011/2012.

- [14] J.M. Mendel et R.I. Bob John : ‘type-2 Fuzzy Sets Made Simple’, IEEE Transactions, Fuzzy Systems, vol. 10, issue.2, page 117-127, 2002.
- [15] J. M. Mendel, R. I. John et F. Liu : ‘Interval type-2 fuzzy logic systems made simple’, IEEE Transactions, Fuzzy Systems, vol.14, issue.6, page808–821, 2006.
- [16] K.wang et H. Lee : ‘First Course on Fuzzy Theory and Applications’, Springer, Berlin, 2005.
- [17] KHEMIS Abderrahmane : ‘Application de la logique floue type-2 intervalle à la commande adaptative de la machine asynchrone’ mémoire de Magister en Electrotechnique, électricité industrielle, Université Mohamed Khider , Biskra, 10/04/2011.
- [18] L.Tsung, L.L Han, J.K et Ming : ‘Direct adaptive interval type-2 fuzzy control of multivariable nonlinear systems’, Engineering Application of artificial intelligence, vol.22, issue.3, page 420-430, 2009.
- [19] Laurant F, Sylvie G et Andre T. ‘ commande floue 1, de la stabilisation à la supervision, paris, Hermès Sciences publication, 2003.
- [20] M Madaoui : ‘commande floue et neuro-floue d’un convertisseur direct AC/AC monophasé à structure minimale, Application à l’alimentation d’une machine asynchrone monophasé’, Thèse de Magister , ENPAAlger, fevrier 2007.
- [21] N.N. Karnik et J. M. Mendel : ‘Type-2 fuzzy logic systems: Type-reduction’, IEEE International conference, vol.2, page. 2046 – 2051, 1998.
- [22] N.N. Karnik et J.M. Mendel : ‘An introduction to type-2 Fuzzy Logic Systems’, Technical Report, University of Southern California, 1998.
- [23] N.N. Karnik et J.M. Mendel : ‘Operations on type-2 fuzzy sets’, Fuzzy Systems, vol. 122, page 327–348, 2001.
- [24] N.N. Karnik et J.M. Mendel : ‘centroid of a type-2 fuzzy set’, Information sciences, vol.132, issue.1-4, page.195-220, 2001.
- [25] Najet Aouchar Et Chafika Bekhaled : ‘Application des systèmes hybrides neuro-flous au diagnostic des huiles de transformateurs’ mémoire de fin d’étude d’ingénieur d’état en électronique, école nationale polytechnique, juin 2005.
- [26] Nakoula Y : ‘Apprentissage des Modèles linguistiques flous, par jeu de règles pondérées’, Thèse de Doctorat, Université de Savoie, France, juillet 1997.
- [27] Oumaya Mohamed et Limam Mohammed Lakhdar : ‘Commande par réseaux d’ondelette-floue’, mémoire de Master en Automatique, Université Kasdi Merbah, Ouargla, 2011/2012.

- [28] Pierre GABRIEL : ‘Introduction à la Logique Floue et à la Commande Floue’ exposé, 2000-2001.
- [29] Q. Liang et J. M. Mendel : ‘Interval type-2 fuzzy logic systems: Theory and design’, IEEE Transactions, Fuzzy Systems, vol.8, issue.5, page 535-550, 2000.
- [30] R Mellah : ‘contribution la commande adaptative neuro-floue, application à la robotique’ Thèse de Doctorat, Université des Sciences et de la Technologie Houari Boumediene, Alger, Mai 2006.
- [31] S. Coupland et R. John : ‘A new and efficient method for the type-2 meet operation’, IEEE international conference, Fuzzy systems, vol.2, page 959-964, 2004.
- [32] S. Coupland et R. John : ‘Geometric type-1 and type-2 fuzzy logic systems’, IEEE Transactions, Fuzzy Systems, vol.15, issue.1, page 3–15, 2007.

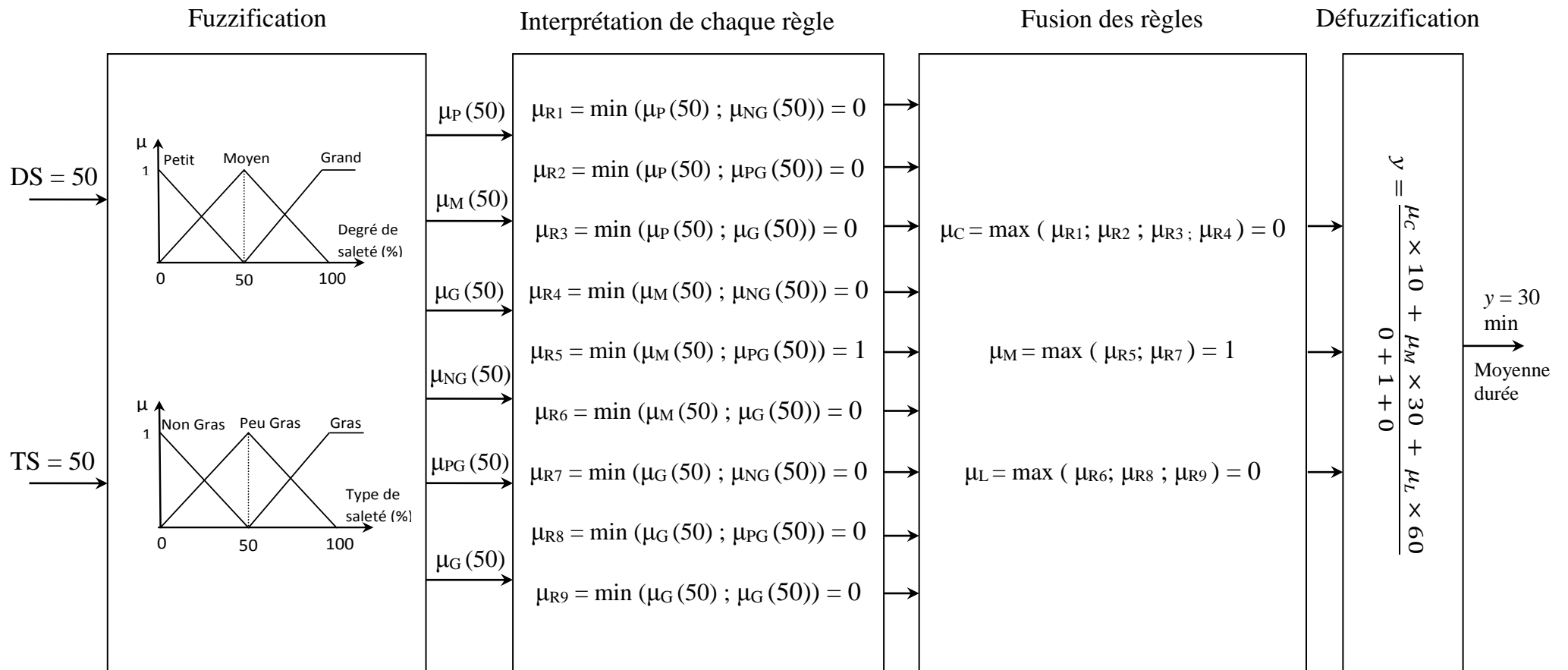


Figure 2.9 : Structure de contrôleur flou de type MAMDANI