REPUBLIQUE ALGERIENNE DEMOCRATIQUE ET POPULAIRE
Ministère de l'Enseignement Supérieur et de la Recherche Scientifique

Ecole Nationale Polytechnique

Département: Génie Electrique

Mémoire de Magister

Filière: Génie Electrique
Spécialité: Automatique

Présentée par : BENBOUCHAMA Cherrad
Ingénieur d'état en Automatique
E.M.P (ex. E.N.I.T.A)

Thème

COMMANDE NEURO-LINGUISTIQUE DE LA SUSPENSION ACTIVE D'UN VEHICULE ROULANT

Soutenue le 16 / 11 / 2000 devant le jury composé de :

Président	:	*M A.ZERGUERRAS*	*Professeur, E.N.P*
Rapporteurs de thèse :		*M C.LARBES*	*Ph.D, E.N.P*
		M N.LOUAM	*Professeur, E.N.P*
Examinateurs	:	*M E.BERKOUK*	*Maître de conférence, E.N.P*
		M M.TADJINE	*Docteur d'état, E.N.P*
		M M.S.AIT CHEIKH	*Chargé de recherche et de cours, E.N.P*

Ecole Nationale Polytechnique 10, Avenue Hassen Badi, El-Harrach, ALGER

A

Ma mère Yamina et mon père Chadli pour tout ce qu'ils ont fait pour moi.

Ma femme Souhila et mon fils Ahmed Hakim.

Tous les amis.

Remerciements

En premier lieu, je tiens à remercier Monsieur Chérif LARBES, Chargé de cours à l'Ecole Nationale Polytechnique d'Alger, pour l'intérêt qu'il a porté à ce travail en tant que directeur de thèse, ainsi que Monsieur Nadjib LOUAM, professeur à l'E.N.P, qui m'a guidé au début de ce travail.

Je remercie aussi Messieurs A.BELHADJ Directeur de l'Ecole Militaire Polytechnique de Bordj-El-Bahri (ex.E.N.I.T.A) et H.REZINE chef de l'U.E.R.Automatique - E.M.P.

Que tous les membres de jury, présidé par le professeur A.ZERGUERRAS, trouvent ici l'expression de ma profonde gratitude pour avoir accepté de juger ce travail.

J'exprime également ma gratitude à tout le corps enseignant du département Génie électrique à l'Ecole Nationale Polytechnique.

Que soient aussi remerciés ici tous les amis, surtout Monsieur Chérif GHEBACHE.

Ministère de l'Enseignement Supérieur et de la Recherche Scientifique
Ecole Nationale Polytechnique
Département de génie électrique
Option : Automatique

Résumé de thèse de Magister
Thème

COMMANDE NEURO-LINGUISTIQUE DE LA SUSPENSION ACTIVE D'UN VEHICULE ROULANT

Présenté par :
C.BENBOUCHAMA
Ingénieur d'état, E.M.P (ex. E.N.I.T.A)

Directeurs de thèse :
C.LARBES : PhD, chargé de cours, E.N.P
N.LOUAM : Professeur, E.N.P

ملـــخـــص:

تتضمن هذه الأطروحة دراسة تقنية تحكم جديدة مطبقة على نظام امتصاص الاهتزازات الإيجابي لسيارة ، و هدا لتحسين الراحة في حالة طريق معلوم. استعمال نظام التحكم العصبي-اللغوي يؤدي إلى تحكم مقبول دون معرفة مسبقة للنموذج، وهذا كله بضمان متانة بالنسبة للتغيرات الوسطية و إشارات التشويش.

كلمات مفتاحيه : التحكم بالمنطق الغامظ، الشبكات العصبية الاصطناعية، التحكم العصبي-اللغوي، نظام امتصاص الاهتزازات الإيجابي.

Abstract :

The present work concerns a novel control technique applied to a vehicle-active suspension, in order to reduce considerably vibrations and to enhance comfort, in the case of known road profile. The neuro-linguistic controller maintains a proper control without the exact knowledge of the model and ensures robustness with regard to parametric variations and diturbances.

Keywords : Fuzzy logic control, neural networks, neuro-linguistic control, active suspension.

Résumé :

Le présent travail concerne une nouvelle technique de commande appliquée à la suspension active d'un véhicule roulant, dans le but de réduire considérablement les vibrations et accroître le confort dans le cas d'un profil de la route connu. Le contrôleur neuro-linguistique assure une commande sans la connaissance à priori du modèle, tout en garantissant une robustesse face à des variations paramétriques et des entrées de perturbations.

Mots clés : Commande par la logique floue, réseaux de neurones, commande neuro-linguistique, suspension active.

Nomenclature

$\mu_A(x)$: *Fonction d'appartenance de la variable linguistique x dans l'ensemble flou A.*
N : *Nombre de neurones de la couche d'entrée.*
L : *Nombre de neurones de la couche cachée.*
M : *Nombre de neurones de la couche de sortie.*
x_p : *Vecteur d'entrée à la couche d'entrée.*
w^c : *Poids constants de la couche cachée.*
w^s : *Poids constants de la couche de sortie.*
net^c : *Entrées des neurones de la couche cachée.*
net^s : *Entrées des neurones de la couche de sortie.*
i : *Sorties des neurones de la couche cachée.*
o : *Sorties des neurones de la couche de sortie.*
δ^c : *Erreur de la couche cachée.*
δ^s : *Erreur de la couche de sortie.*
E : *Erreur en sortie du réseau de neurones.*
e : *Première variable linguistique.*
de : *Dérivée de e et deuxième variable linguistique.*
a, b, c : *Les trois paramètres de la fonction triangulaire.*
y_1 : *Déplacement de l'axe de la roue(avant).*
y_2 : *Déplacement du châssis(avant).*
y_3 : *Déplacement de l'axe de la roue arrière.*
y_4 : *Déplacement du châssis arrière.*
$\dot{y}_1$: *Vitesse de déplacement de l'axe de la roue(avant).*
$\dot{y}_2$: *Vitesse de déplacement du châssis(avant).*
$\dot{y}_3$: *Vitesse de déplacement de l'axe de la roue arrière.*
$\dot{y}_4$: *Vitesse de déplacement du châssis arrière.*
y_a : *Composante verticale de la surface de la route(avant).*
y_b : *Composante verticale de la surface de la route arrière.*
f : *Vecteur de commande.*
f_1 : *Force de l'actionneur avant.*
f_2 : *Force de l'actionneur arrière.*
M : *Masse du châssis du véhicule.*
J : *Moment d'inertie du châssis.*
m : *Masse de la roue(quart du véhicule).*
h : *Raideur du pneu(quart du véhicule).*
k : *Raideur du ressort de la suspension conventionnelle(quart du véhicule).*
d : *Amortisseur de la suspension conventionnelle(quart du véhicule).*
M_1 : *Masse de l'essieu et la roue avant(moitié du véhicule).*
M_3 : *Masse de l'essieu et la roue arrière(moitié du véhicule).*
h_1 : *Raideur du pneu avant(moitié du véhicule).*
h_3 : *Raideur du pneu arrière(moitié du véhicule).*
k_1 : *Raideur du ressort de la suspension conventionnelle avant.*
k_3 : *Raideur du ressort de la suspension conventionnelle arrière.*
d_1 : *Amortisseur de la suspension conventionnelle avant.*
d_3 : *Amortisseur de la suspension conventionnelle arrière.*

Table des matières

Chapitre 1

INTRODUCTION

L'automatique, dans le but d'élargir son champ d'application, ressent actuellement un besoin d'intégrer de nouveaux concepts regroupés sous le terme de commande intelligente.

L'objectif est d'introduire de nouveaux mécanismes permettant une commande plus souple, capable de s'adapter à des variations de l'environnement et démontrant des capacités d'apprentissage, telles que les efforts et les interventions de l'homme, tant dans les phases de conception que de conduite proprement dite, en soient significativement réduits.

Parmi ces concepts, on peut citer la commande "Neuro-linguistique" basée sur l'association de deux autres nouveaux concepts.

Le premier est la technique linguistique floue qui permet la représentation de la connaissance et le raisonnement humains, et avec laquelle on peut conçevoir des approximateurs universels.

Le second est le pouvoir d'apprentissage et de généralisation des réseaux de neurones formels.

L'objet de ce travail est l'application de ce nouveau concept de commande à un système qui attire, ces dernière années, l'attention de nombreux chercheurs, et qui n'est autre que la suspension active de véhicules roulants.

Effectivement, les premières recherches conçernant les suspensions actives de véhicules étaient basées sur les théories conventionnelles de la commande des systèmes [Tho 76] [Tom 76] [Hac 85] [Yue 89] [Yos 90] [Yos 91] [Hro 91] [Ray 92]. Tous ces travaux supposaient que le modèle était linéaire, que l'indice de performance comportait les accélérations verticales du châssis, les décollements dynamiques des roues, les espaces de fonctionnement et la force de l'actionneur.

Une condition nécéssaire dans la conception de la commande de la suspension active est que le modèle doit être précis. Une condition difficile à obtenir puisqu'en réalité le modèle est plus compliqué, comportant des non-linéarités

et des incertitudes paramétriques, et que le profil de la route ne peut être connu d'une façon précise.

Dans le but de trouver une solution à ce problème, d'autres travaux ont été réalisés, parmi lesquels l'application de la commande floue à la suspension active qui a été développée et a donné des résultats satisfaisants [Lin 93] [Yeh 94] [Yos 96.a] [Yos 96.b]et[Rao 97].

Les systèmes d'inférence floue sont des approximateurs universels[Cas 95] [Wan 92]et sont capables de représenter une connaissance humaine, mais ces deux propriétés ont montré des limites dans certaines circonstances:

- La connaissance d'un expert ne peut être entièrement verbalisée.

- La précision dans l'approximation se paye par l'augmentation considérable de la base de règles.

Alors, pour remédier à l'ensemble des problèmes rencontrés par les techniques de commande classiques et celle utilisant les systèmes d'inférence floue, dans ce travail une solution est proposée, elle consiste en l'utilisation d'un système de commande hybride ou neuro-linguistique.

Cette thèse se présente comme suit. Dans le deuxième chapitre, un rappel sur la théorie de la logique floue, ainsi que son utilisation dans la commande des systèmes sont présentés.

Le troisième chapitre est consacré à la présentation des notions élémentaires des réseaux de neurones formels, ainsi qu'une application dans laquelle est mise à profit leur puissance de calcul pour régler un problème de commande.

Dans le quatrième chapitre, on présente la méthode de synthèse d'un contrôleur neuro-linguistique basée sur l'association des deux techniques neuronale et floue.

Le cinquième chapitre est consacré à la formulation du problème de la suspension active de véhicules roulants.

Dans le sixième chapitre, on applique le contrôleur neuro-linguistique déjà conçu à la commande du système de suspension.

Enfin, le dernier chapitre contient les conclusions et les recommandations pour effectuer, éventuellement, de futurs travaux dans le même axe de recherche.

Chapitre 2

LOGIQUE ET COMMANDE FLOUES

2.1 *INTRODUCTION*

La logique floue est née, effectivement, en 1965 des travaux de Lotfi A. Zadeh Professeur à l'université de Californie à Berkeley, internationalement reconnu pour ses travaux en automatique et théorie des systèmes, qui a éprouvé le besoin de formaliser la représentation et le traitement de connaissances imprécises ou approximatives, afin de pouvoir traiter des systèmes d'une grande complexité dans lesquels sont, par exemple présents des facteurs humains. La logique floue intervient dans la manipulation de connaissances imparfaites.

La commande floue est un domaine d'application de la théorie des ensembles flous qui a été abordée trés tôt, proposée par L.A.Zadeh[Zad 73]et développée par d'autres chercheurs. Son principe a été introduit en Angleterre par Mamdani et Assilian[Ass 74], [Mam 75]en prenant l'exemple d'une machine à vapeur expérimentale. Il a ensuite été exploité pour la commande d'une usine d'eau chaude[Lem 76]et un échangeur de chaleur[Ost 77], pour la régulation de la vitesse d'un moteur en France[Wil 77]et pour la réalisation industrielle d'un four à ciment au Danemark[Hol 82]. La commande floue a été largement developpée au Japon à partir du début des années 80, où il existe actuellement de nombreuses réalisations industrielles.

Dans ce chapitre, on commence par une présentation des bases générales de la logique floue, ensuite on présente la méthode de synthèse d'un contrôleur flou.

2.2 LA LOGIQUE FLOUE

2.2.1 Ensemble flou

La notion d'ensemble flou provient du constat établi par Zadeh en 1965: " trés souvent les classes d'objets rencontrées dans le monde physique ne possédent pas de critéres d'appartenance bien définis ".

Mathématiquement, un ensemble flou A sera défini sur un référentiel U par une fonction d'appartenance, notée μ_A , qui appliquée à un élément u de U retourne un degré d'appartenance $\mu_A(u)$ de u à U.

Quand U est discret, A peut être représenté par:

$$A = \sum_{u_i \in U} \mu(u_i)/u_i \tag{2.1}$$

et quand U est continu, A est représentée par:

$$A = \int_U \mu(u)/u \tag{2.2}$$

Les ensembles flous ont le grand avantage de constituer une représentation mathématique de labels linguistiques largement utilisés dans l'expréssion de connaissances expertes et qualitatives.

2.2.2 Opérateurs des ensembles flous

Disposant d'une représentation formelle de classes floues sous la forme de fonctions d'appartenances, il est naturel de chercher à faire avec les ensembles flous ce que l'on fait avec les ensembles classiques. Zadeh a ainsi généralisé les opérateurs d'ensembles classiques au cas d'ensembles flous.

Ces opérateurs appliqués aux ensembles flous sont définis à partir de leurs fonctions d'appartenance.

Soit A et B deux ensembles flous définis dans le référentiel U avec les fonctions d'appartenance μ_A et μ_B respectivement:

2.2.2.1 Union

La fonction d'appartenance $\mu_{A \cup B}$ de l'union $A \cup B$, notée aussi A OR B, est définie pour tout $u \in U$ par:

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

2.2.2.2 Intersection

La fonction d'appartenance $\mu_{A\cap B}$ de l'intersection $A\cap B$, notée aussi A AND B, est définie pour $u\in U$ par:

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

2.2.2.3 Complémentation

La fonction d'appartenance $\mu_{\bar{A}}$ du complément de l'ensemble A notée $\bar{A}$ ou NOT A est définie pour tout $u\in U$ par:

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

2.2.2.4 Produit cartésien

Soit $A_1, A_2, ..., A_n$ des ensembles flous respectivement définis sur $U_1, ..., U_n$ et de fonction d'appartenance:

$$\mu_{A_1\times...\times A_n}(u_1, ..., u_n) = \min(\mu_{A_1}(u_1), ..., \mu_{A_n}(u_n)) \tag{2.6}$$

ou

$$\mu_{A_1\times...\times A_n}(u_1, ..., u_n) = \mu_{A_1}(u_1) \times ... \times \mu_{A_n}(u_n) \tag{2.7}$$

2.2.2.5 Relation floue

Une relation floue de n-uplet est un ensemble flou défini sur $U_1, ..., U_n$ par l'expression suivante:

$$R_{U_1,...,Un} = [((U_1, ..., U_n), \mu_R(u_1, ..., u_n))(u_1, ..., u_n) \in U_1\times... \times U_n] \tag{2.8}$$

Notons qu'il existe d'autres définitions pour les opérateurs flous AND et OR dans la littérature sous les noms " Normes Triangulaires " et "Co-normes Triangulaires" respectivement.

2.2.3 Variable linguistique

Une variable linguistique est représentée par un triplet (x,T(x),U) dans lequel x est le nom de la variable linguistique (vitesse,position,erreur,...), T(x) est l'ensemble des symboles linguistiques qui sont utilisés pour caractériser x et définissant des restrictions de valeurs que prend x dans U. Par exemple, si la

position est considérée comme variable linguistique, définie sur le référentiel U=[-10,+10], ses symboles linguistiques peuvent être:

T(position)={ Négative(N) , Zéro(Z) , Positive(P)}

Ces symboles linguistiques peuvent être considérés comme des ensembles flous dont les fonctions d'appartenances sont représentées sur la figure2.1:

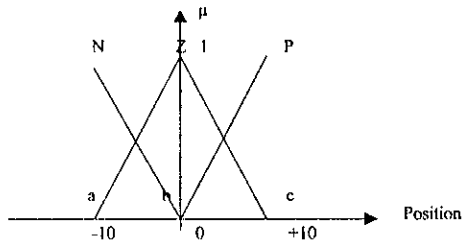


Figure 2.1: *Variable linguistique (position).*

2.2.4 *Fonction d'appartenance*

Afin de permettre un traitement numérique des variables linguistiques dans la prise de décisions floues sur calculateurs, une définition des variables linguistiques à l'aide de fonctions d'appartenance s'impose. Dans ce contexte, on attribue à chaque valeur de la variable linguistique x une fonction d'appartenance $\mu_A(x)$, dont la valeur varie entre 0 et 1, tandis que A indique l'ensemble conçerné. Une valeur précise pour $\mu_A(x)$ sera désignée par le degré ou le facteur d'appartenance.

Le plus souvent, on utilise pour les fonctions d'appartenance les fonctions suivantes:

2.2.4.1 Fonction triangulaire

Elle est définie par trois paramètres {a, b, c} qui détèrminent les coordonnées des trois coins (figure 2.2):

$$\mu(x) = \max\left(\min\left((x-a)/(b-a), (c-x)/(c-b)\right), 0\right) \qquad (2.9)$$

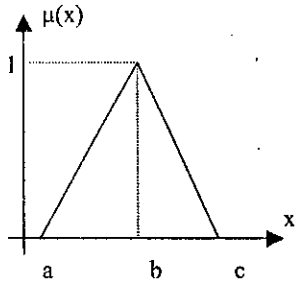


Figure 2.2: *Fonction triangulaire.*

2.2.4.2 Fonction trapézoïdale

Elle est définie par quatre paramètres {a, b, c, d} comme suit (figure 2.3):

$$\mu(x) = \max\left(\min\left((x-a)/(b-a), 1, (d-x)/(d-c)\right), 0\right) \ . \qquad (2.10)$$

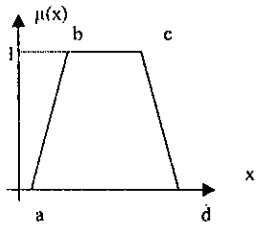


Figure 2.3: *Fonction trapèzoïdale.*

2.2.4.3 Fonction Gaussiènne

Elle est définie par deux paramètres {m, σ} (figure 2.4):

$$\mu(x) = \exp\left[-(x-m)^2/\sigma^2\right] \qquad (2.11)$$

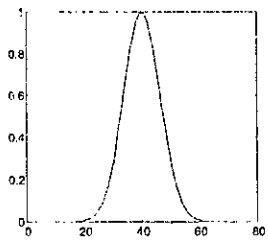


Figure 2.4: *Fonction Gaussiènne ($m = 40$ et $\sigma = 10$).*

2.2.4.4 Fonction sigmoïde

Elle est définie par (figure 2.5):

$$\mu(x) = 1/[1 + \exp(-a(x - c))] \tag{2.12}$$

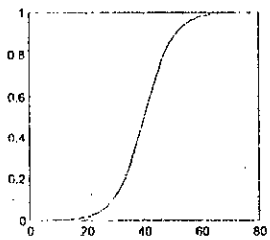


Figure 2.5: *Fonction sigmoïde($a = 0.2$ et $c = 40$).*

2.2.5 *Implication floue*

L'implication floue est un opérateur qui permet d'évaluer un degré de vérité d'une règle R de la forme ' Si x est A alors y est B ' à partir des valeurs de la prémisse d'une part, et de celle de la conclusion d'autre part:

$$\mu_R(x, y) = I(\mu_A(x), \mu_B(y)) \tag{2.13}$$

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

- Implication de Mamdani:

$$\mu_M(x, y) = \min(\mu_A(x), \mu_B(y)) \tag{2.14}$$

- *Implication de Larsen:*

$$\mu_L(x,y) = \mu_A(x) \times \mu_B(y) \tag{2.15}$$

2.2.6 *Raisonnement flou*

En logique classique, à partir de la règle ' Si x est A alors y est B ' et du fait ' x est A ', on peut conclure le fait ' y est B ' qui sera ajouté à la base des faits. Zadeh a étendu ce principe au cas flou:

- Logique classique:
Fait: x est A
Règle: si x est A alors y est B
.....
Déduction: y est B
-Logique floue:
Fait: x est A'
Règle: si x est A alors y est B
.....
Déduction: y est B'

A partir de la règle ' si A alors B ' et du fait A', on déduit un nouveau fait B' qui est caractérisé par un ensemble flou dont la fonction d'appartenance est :

$$\mu_{B'}(y) = \sup_{x \in X} \min(\mu_{A'}(x), \mu_R(x,y)) \tag{2.16}$$

Les fonctions d'appartenance $\mu_{A'}$ et μ_R caractérisent respectivement le fait A' et la règle.

2.3 *La commande par la logique floue*

2.3.1 *Structure d'un contrôleur flou*

Un contrôleur flou est un système à base de connaissances particulier. Il est constitué de trois modules principaux(figure 2.6):

– Fuzzification: transforme la grandeur d'entrée en une partie floue.

- L'inférence: transforme, à l'aide d'un jeu de règles, la partie floue issue de la fuzzification en une nouvelle partie floue.

-La défuzzification: transforme la partie floue issue de l'inférence en une grandeur physique réelle applicable au processus.

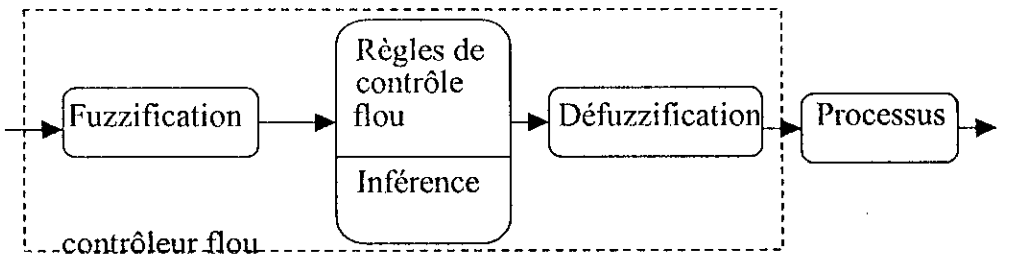


Figure 2.6: *Architecture de base d'un contrôleur flou.*

Selon que la sortie d'un contrôleur conçerne la commande ou sa variation, on obtiendra des équivalents structuraux des contrôleurs classiques PD, PI et PID(figure 2.7):

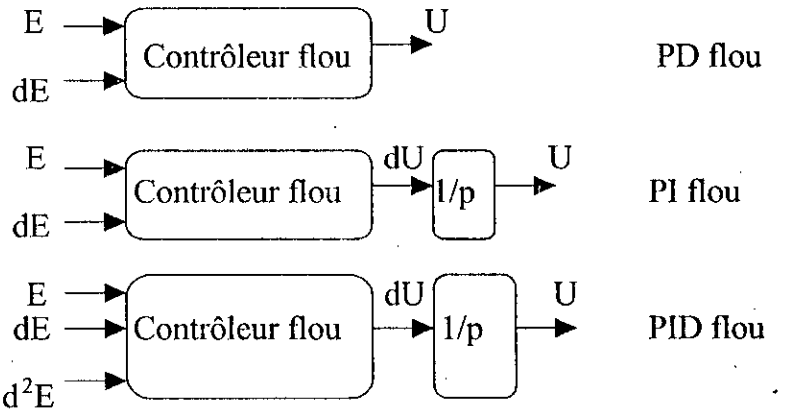


Figure 2.7: *Equivalents flous des contrôleurs classiques.*

2.3.2 *La fuzzification*

La fuzzification consiste en une opération de projection de variables physiques réelles sur des ensembles flous caractérisant les valeurs linguistiques prises par ces variables. La fuzzification d'une mesure précise d'une variable phy-

sique réelle permet de caractériser le degré avec lequel cette mesure appartient à un ensemble flou donné en utilisant des fonctions d'appartenances.

Les grandeurs physiques sont souvent normalisées entre -1 et +1 par un facteur d'échelle qui doit être choisi sur la base de l'étude du système.

La forme des fonctions d'appartenances et le nombre d'ensembles flous (nombre des valeurs linguistiques)ont une influence sur la caractéristique du régulateur flou.

2.3.2.1 Définition des fonctions d'appartenances pour les variables d'entrées

La fuzzification proprement dite consiste à définir les fonctions d'appartenances pour les différentes variables, notamment pour les variables d'entrées. On réalise, ainsi, le passage des variables physiques aux variables floues qui peuvent, par la suite, être traitées par les inférences. En général, on introduit pour une variable un nombre impair(trois, cinq, sept, ...)d'ensembles flous, représentés par des fonctions d'appartenances(figure 2.8), généralement, de formes triangulaires ou trapézoïdales.

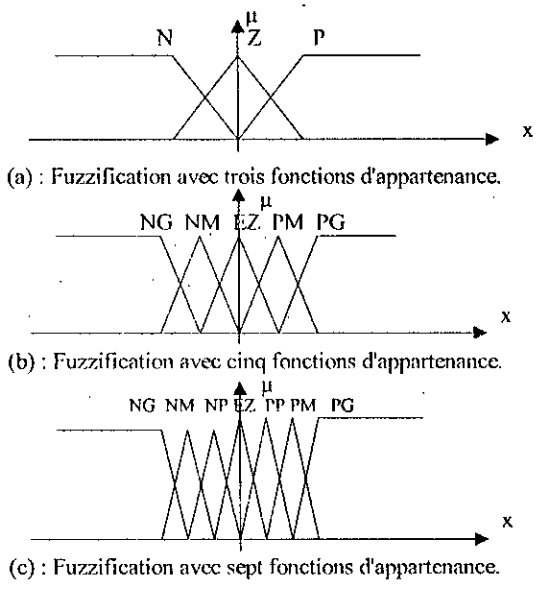


Figure 2.8: *Fuzzification de la variable d'entrée.*

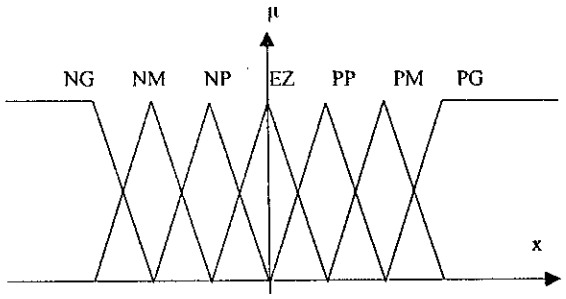


Figure 2.9: *Défuzzification de la variable de sortie.*

2.3.2.2 Définition des fonctions d'appartenances pour la variable de sortie

Il est également indispensable de fuzzifier la variable de sortie du contrôleur. Les ensembles flous obtenus seront utilisés au niveau de la formulation des règles et lors de la défuzzification.

Un exemple typique pour la dèfuzzification de la variable de sortie, où l'on a employé sept fonctions d'appartenences, est illustré par la figure 2.9. La variable de sortie est normalisée entre -1 et +1, ce qui nécéssite l'introduction d'un facteur d'échelle adéquat.

2.3.3 *L'inférence*

L'inférence transforme, à l'aide du jeu de règles, la partie floue issue de la fuzzification en une nouvelle partie floue qui caractérisera la sortie du contrôleur. Dans ce qui suit, nous étudierons la manière d'établissement des régles utilisées dans l'inférence et on présentera les méthodes de traitement numérique des inférences.

2.3.3.1 Jeu de règles

Un contrôleur flou est toujours associé à un jeu de règles où chaque règle a la forme suivante: Si prémisse alors conclusion.

On peut distinguer deux types de règles selon que la conclusion est symbolique, comme dans la méthode de Mamdani (si l'erreur est grande alors la commande est moyenne), ou qu'elle est numérique, comme dans la méthode de Sugeno (si l'erreur est grande alors $U = 3 \times e + 5$ *avec e la valeur numérique de l'erreur).*

2.3.3.2 Ecriture des règles

Pour la commande des systèmes complexes, l'écriture des règles peut se faire:

- Par l'utilisation de l'expertise et de l'expérience des opérateurs, formulées sous forme de règles floues.

- Par l'observation et la modélisation du comportement et des actions des opérateurs.

- Par l'utilisation de méthodes d'apprentissage, notamment avec l'introduction des réseaux de neurones.

2.3.3.3 Représentation des règles

On choisit, par exemple, un système quelconque à deux grandeurs physiques d'entrées mesurées et convenablement transformées en variables linguistiques x_1 et x_2. Elles forment les variables d'entrées du contrôleur. La variable de sortie est désignée par u et est exprimée, également, comme variable linguistique. Le jeu de règles du contrôleur peut être décrit, soit d'une façon linguistique:

si x_1 est négatif et x_2 est zéro alors u est positif OU
si x_1 est zéro et x_2 est zéro alors u est zéro OU
si x_1 est positif et x_2 est zéro alors u est négatif OU
si x_1 est zéro et x_2 est négatif alors u est positif OU
si x_1 est zéro et x_2 est positif alors u est négatif .

ou d'une manière symbolique:

si x_1 est N et x_2 est Z alors u est P OU
si x_1 est Z et x_2 est Z alors u est Z OU
si x_1 est P et x_2 est Z alors u est N OU
si x_1 est Z et x_2 est N alors u est P OU
si x_1 est Z et x_2 est P alors u est N.

La description symbolique est plus compacte et, donc, de meilleur clarté.

Une simplification de la description des règles s'obtient à l'aide d'une représentation graphique, appelée matrice d'inférence(figure 2.10).

Si toutes les positions de la matrice d'inférence sont remplies, on parle de règles d'inférences complètes. Dans le cas contraire, il s'agit de règles d'inférences incomplètes. Cependant, cette représentation devient complexe lorsqu'il y a plus de trois variables et un nombre élevé d'ensembles.

Une autre possibilité de représentation est celle d'un tableau d'inférence (figure 2.11). Elle se prête particulièrement bien aux systèmes MIMO.

u		x_1		
		N	Z	P
x_2	N		P	
	Z	P	Z	N
	P		N	

Figure 2.10: *Matrice d'inférence pour deux variables linguistiques.*

Règle N°	x_1	x_2	x_3
1	N	Z	P
2	Z	Z	Z
3	P	Z	N
4	Z	N	P
5	Z	P	N

Figure 2.11: *Tableau d'inférence.*

2.3.3.4 Traitement numérique des inférences

Dans le jeu de règles du contrôleur flou interviennent les opérateurs logiques tels que ET(AND)et OU(OR). L'opérateur AND s'applique aux variables à l'intèrieur d'une règle, tandis que l'opérateur OR lie les différentes règles.

Il existe plusieurs possibilités pour réaliser ces opérateurs qui s'appliquent aux fonctions d'appartenance. On introduit, alors, la notion de méthode d'inférence qui détermine la réalisation des différents opérateurs.

Les principales méthodes d'inférence qui sont utilisées sont:

- La méthode de Mamdani.

- La méthode de Larsen.

- La méthode de Tsukamoto.

- La méthode de Sugeno.

a-La méthode de Mamdani: *Elle repose sur l'utilisation de l'opérateur min pour la combinaison des prémisses et pour l'implication. L'agrégation des règles est réalisée par l'opérateur max. La figure 2.12 illustre le principe de cette méthode pour les deux règles suivantes:*

R_1: si x est A_1 et y est B_1 alors u est C_1 OU

R_2: si x est A_2 et y est B_2 alors u est C_2

Pour ces deux règles la combinaison des prémisses et l'agrégation des résultats sont données par:

$$\begin{aligned} a_i &= \min\left(\mu_{A_i}(x_0), \mu_{B_i}(y_0)\right) \\ \mu_{C_i'}(u) &= \min(a_i,\ \mu_{C_i}(u)) \\ \mu_{C'}(u) &= \max\left(\mu_{C_1'}(u), \mu_{C_2'}(u)\right) \end{aligned} \tag{2.17}$$

Par la suite, l'opération de défuzzification permet d'obtenir une valeur physique réelle de sortie.

b-La méthode de Larsen: *Cette méthode est basée sur l'utilisation du produit pour l'implication, dans ce cas la $i^{\text{ème}}$ règle donne la décision:*

$$\mu_{C_i'}(u) = a_i \times\ \mu_{C_i}(u) \tag{2.18}$$

Par conséquent , la fonction d'appartenance résultante de l'exemple précédent est donnée par:

$$\mu_C(u) = \max\left(a_1.\mu_{C_1}(u), a_2.\mu_{C_2}(u)\right) \tag{2.19}$$

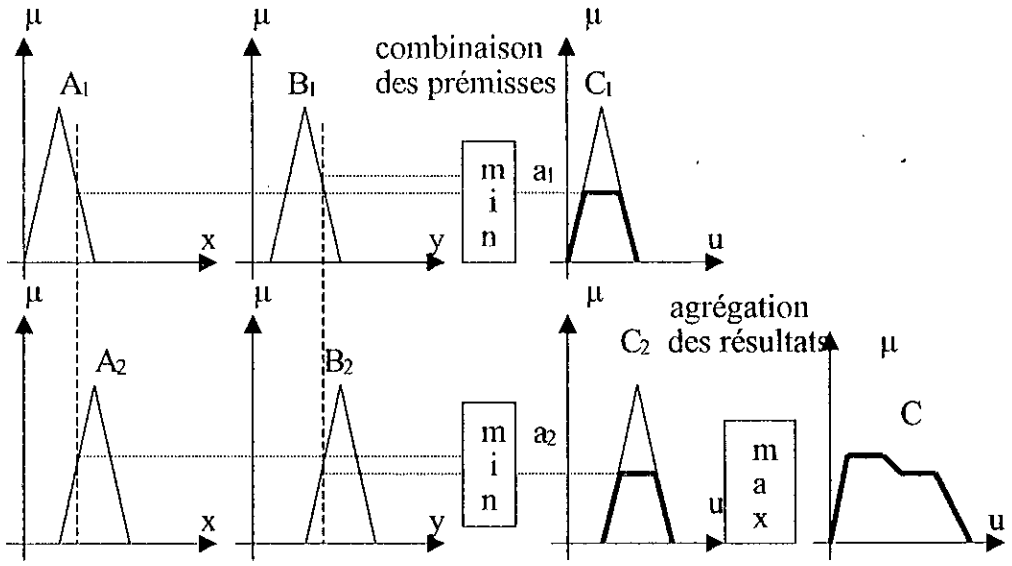


Figure 2.12: *Principe de la méthode de Mamdani.*

La méthode est illustrée par la figure 2.13.

La sortie concrète du contrôleur est déterminée, par la suite, par la défuzzification.

c-La méthode de Tsukamoto: *C'est une méthode simplifiée basée sur la méthode de Mamdani dans laquelle les fonctions d'appartenance de la variable de sortie sont monotones(figure 2.14).*

Dans cette méthode, les résultats obtenus de la première règle a_1 *tel que* $a_1 = C_1(u_1)$ *et de la deuxième règle* a_2 *tel que* $a_2 = C_2(u_2)$ *leurs correspond une action réelle donnée par:*

$$\bar{u} = \frac{a_1 u_1 + a_2 u_2}{a_1 + a_2} \tag{2.20}$$

d-La méthode de Sugeno: *Cette règle utilise des règles à conclusion numérique de la forme:*

R_i: *si* x *est* A_1, ... , *et* y *est* B_1 *alors* $u = f(x, ..., y)$

où x,..., y *et* u *sont des variables linguistiques.*

Pour simplifier le principe de la méthode, considérons les règles suivantes:

R_1: *si* x *est* A_1, *et* y *est* B_1 *alors* $u_1 = f_1(x, y)$

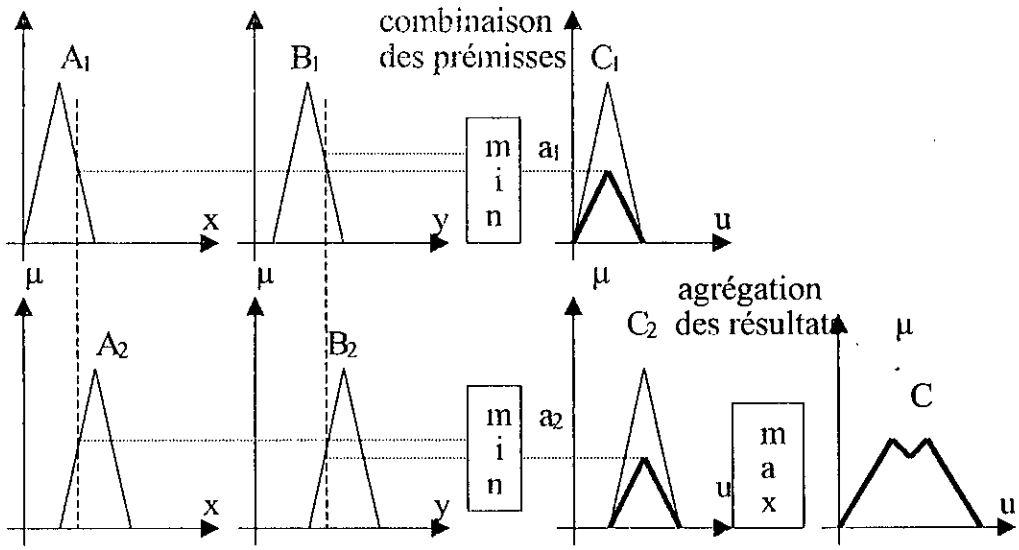


Figure 2.13: *Principe de la méthode de Larsen.*

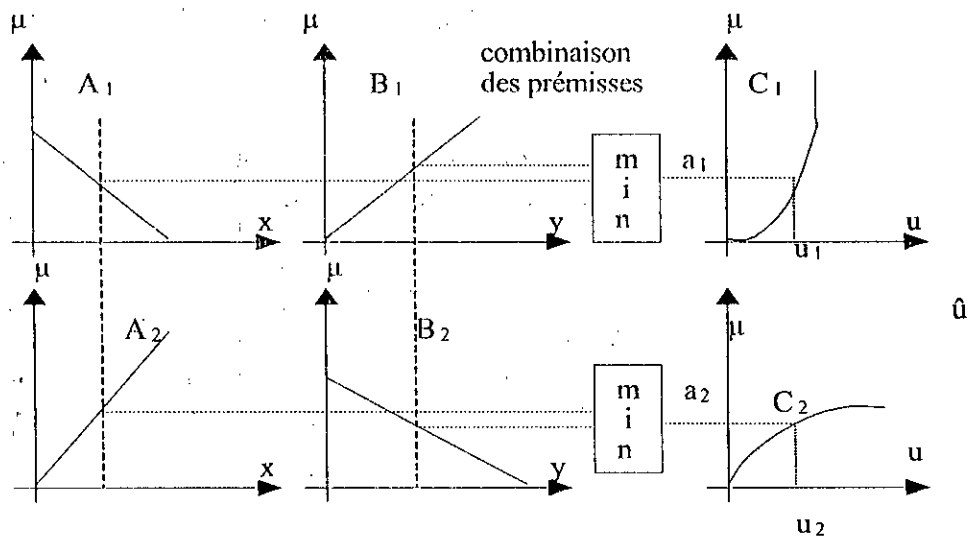


Figure 2.14: *Principe de la méthode de Tsukamoto.*

R_2: si x est A_2, et y est B_2 alors $u_2 = f_2(x, y)$

La valeur obtenue de la première règle est $a_1 \times f_1(x, y)$, de la deuxième règle est $a_2 \times f_2(x, y)$. La valeur de la sortie engendrée par les entrées x_0 et y_0, est illustrée sur la figure 2.15 et donnée par l'expression:

$$\hat{u} = \frac{a_1 \times f_1(x_0, y_0) + a_2 \times f_2(x_0, y_0)}{a_1 + a_2} \tag{2.21}$$

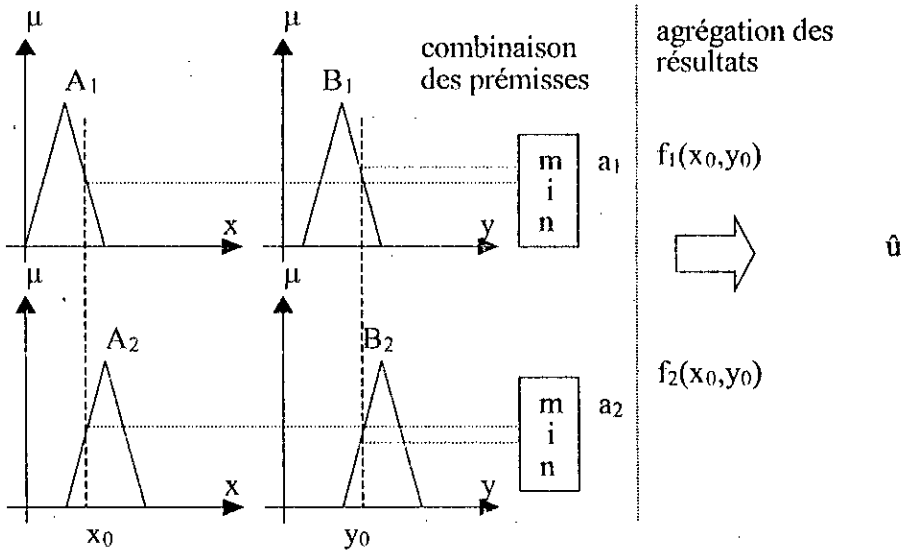


Figure 2.15: *Principe de la méthode de Sugeno.*

2.3.4 *La défuzzification*

Le rôle de la défuzzification est de transformer la partie floue issue de l'inférence en une grandeur numérique de commande. Malheureusement, il n'y a pas une procédure systématique pour choisir la stratégie de défuzzification. Comme on s'intéresse à l'application de la logique floue en commande, un des critères pour le choix d'une méthode de défuzzification est la simplicité du calcul. Ce critère a conduit aux différentes méthodes suivantes.

2.3.4.1 La méthode du maximum

Cette méthode examine l'ensemble flou C issu de l'inférence et choisit comme sortie la valeur u pour laquelle $\mu_C(u)$ est maximale.

2.3.4.2 La méthode des moyennes des maximums

Cette méthode examine l'ensemble flou C issu de l'inférence et détèrmine en premier temps les valeurs u_i pour lesquelles $\mu_C(u_i)$ est un maximum. Ensuite, la valeur moyenne de ces valeurs sera considérée comme résultat de défuzzification.

2.3.4.3 La méthode du centre de gravité

Cette méthode calcule le centre de gravité $\bar{u}$ de l'ensemble flou C résultant et considére cette valeur comme résultat de la défuzzification:

$$\bar{u} = \frac{\int_S u\mu_C(u)du}{\int_S \mu_C(u)du} \tag{2.22}$$

où S représente le support de $\mu_C(u)$.
La version discrète est de la forme:

$$\bar{u} = \frac{\sum_{i=1}^{I} u_i\mu_C(u_i)}{\sum_{i=1}^{I} \mu_C(u_i)} \tag{2.23}$$

Notons que cette méthode exige, en général, une envergure de calcul assez importante.

2.3.4.4 La méthode des hauteurs pondérées

Dans cette méthode la valeur réelle de sortie est donnée par la relation suivante:

$$\bar{u} = \frac{\sum_{i=1}^{m} \alpha_i\bar{u}_i}{\sum_{i=1}^{m} \alpha_i} \tag{2.24}$$

où
α_i: le facteur d'appartenance des conditions dans la règle R_i.
$\bar{u}_i$: le centre de gravité de l'ensemble flou de la variable de sortie associée à la règle R_i.

2.3.5 Mise en oeuvre du contrôleur flou

La mise en oeuvre du contrôleur flou ne nécéssite pas la connaissance du modèle mathématique du système à commander, mais uniquement son comportement qualitatif: le sens des variations de ses sorties en fonctions des variations de ses entrées. Dans le cas des systèmes MIMO, il est nécéssaire de savoir si une entrée affecte toutes les sorties du système[Pro 79]. La mise en oeuvre du contrôleur passe par les étapes suivantes[Lee 90].

2.3.5.1 Définition des variables linguistiques

Ceci consiste à choisir les grandeurs qui serviront de variables linguistiques pour le contrôleur. Ces grandeurs doivent refléter suffisamment l'état du système pour en déduire la commande à appliquer.

Le choix des variables linguistiques a un effet considérable sur les performances du système. En général, on utilise comme variables linguistiques l'erreur entre la sortie et la consigne, et la variation de cette erreur.

2.3.5.2 Partition des variables en classes

Dans cette étape, on définie les valeurs linguistiques que peuvent prendre les variables linguistiques choisies en découpant l'intervalle d'intêret en un nombre fini de classes. Pour cela, il n'existe pas de règles et le choix est donc subjectif.

Le nombre de classes doit être suffisamment grand pour avoir une bonne approximation et le choix des classes a une influence essentielle sur la finesse de la commande.

2.3.5.3 Normalisation

Cette étape consiste à rapporter le domaine de variation de chaque variable linguistique, qu'elle soit en entrée ou en sortie, à un intervalle normalisé $[-1, +1]$. *Cela ne peut se faire indépendamment du processus à commander.*

2.3.5.4 Le choix de la fonction d'appartenance

Le choix de la fonction d'appartenance est trés subjectif et affecte sensiblement les performances du contrôleur. En particulier, si les entrées du contrôleur sont bruitées, des fonctions d'appartenances choisies assez larges permettent de réduire sa sensibilité au bruit.

2.3.5.5 Complétude

Le contrôleur flou doit être capable de déduire la commande à appliquer quelque soit l'état du système. Il faut que chaque élément de l'intervalle d'intêret appartienne à au moins une des classes résultantes du découpage de celui-ci. Ceci garantit que le contrôleur sera complet.

2.3.5.6 Source et dérivation des règles floues

Les règles floues pour la prise de décision peuvent être dérivées en utilisant:

a-L'expérience et la connaissance des systèmes: *On peut constituer un ensemble de règles floues à partir d'une bonne connaissance qualitative du système et en utilisant une certaine expérience dans la commande du processus. Une autre approche consiste à interroger des spécialistes expérimentés dans la gestion d'un processus ou des opérateurs qui, souvent à l'aide de manipulations basées sur l'expérience, arrivent à commander avec succés un processus dont ils ignorent le modèle mathématique.*

b-Un apprentissage: *Cette technique consiste à utiliser la puissance de mémorisation et de généralisation des réseaux de neurones(voir chapitre suivant)à partir d'exemples appris pour la génération de la commande, et c'est cette technique qui sera utilisée dans la suite de ce travail.*

2.4 *CONCLUSION*

Le but de la commande floue est, comme en automatique classique, de traiter des problèmes de commande de processus, c'est à dire de gérer un processus en fonction d'une consigne donnée, par action sur les variables qui décrivent le processus, mais son approche est différente de celle de l'automatique classique. Elle se sert le plus souvent des connaissances des experts ou d'opérateurs qualifiés travaillant sur le processus. Donc, contrairement aux méthodes de synthèses classiques, la logique floue permet d'élaborer une commande sans la connaissance du modèle mathématique du système à commander. Toutefois, le choix des variables linguistiques, des fonctions d'appartenance, des règles floues, des classes et de leur nombre est purement subjectif.

Chapitre 3

LES RESEAUX DE NEURONES

3.1 *INTRODUCTION*

L'unité centrale de l'être humain est son cerveau qui est l'organe le plus complexe. Cet organe lui permet d'analyser et de comprendre les différents phénomènes qui l'entourent. Cette merveilleuse machine est composée de $36^{1679616}$ *entités élémentaires appelées neurones. Chaque neurone est reliée à d'autres par des synapses(36000 par neurone)[Ren 95]. C'est l'interconnexion des neurones qui permet au cerveau de réaliser les différentes fonctions telles que la réflexion, la vision, l'audition, etc.... On appelle cette interconnexion "un réseau de neurones", et c'est en essayant de comprendre son fonctionnement que l'homme s'est trouvé en train de modéliser le cerveau et c'est ce qui a permis aux "réseaux de neurones artificiels" de voir le jour.*

Ces derniers sont des modèles mathématiques et informatiques, des assemblages d'unités de calculs appelés, aussi, neurones formels, et dont l'inspiration originelle était un modèle de la cellule nerveuse humaine. Le souci de maintenir une certaine correspondance avec le système nerveux humain a animé et continue d'animer une part importante des recherches dans le domaine.

Malgré cet héritage, l'essentiel des travaux d'aujourd'hui ont pour objet le réseau de neurones formels et non son corrélat neurobiologique. Vus comme des systèmes de calcul, les réseaux de neurones possèdent plusieurs propriétés qui les rendent intéressants d'un point de vue théorique, et fort utiles en pratique(optimisation, reconnaissance, classification, identification, commande, etc...).

Dans ce chapitre, on commence par une présentation des notions élémen-

taires du domaine, ensuite on présente une application dans laquelle est mise à profit la puissance de calcul des réseaux de neurones pour régler un problème de commande de systèmes.

3.2 LES RESEAUX DE NEURONES

3.2.1 Le neurone

Un neurone η est une fonctionnelle caractérisée par les éléments suivants (figure 3.1):

a- Un sommateur pondéré.
b- Un systéme dynamique linéaire SISO.
c- Une fonction non-linéaire statique(fonction d'activation).

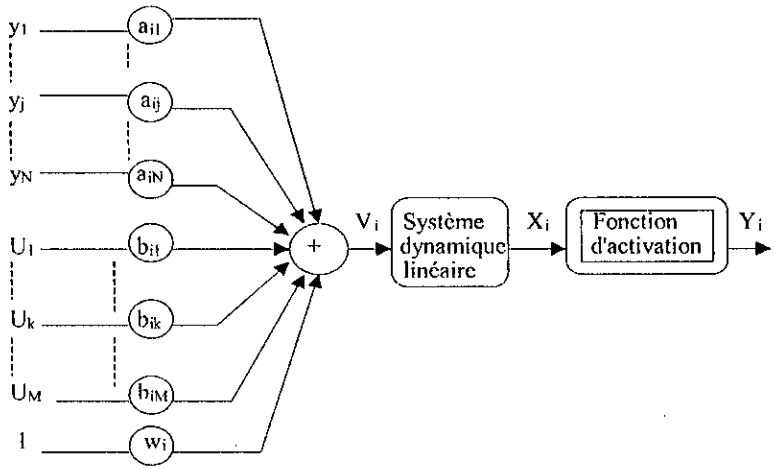


Figure 3.1: *Modèle d'un neurone.*

3.2.1.1 Le sommateur pondéré

Il est décrit par l'équation suivante:

$$v_i(t) = \sum_{j=1}^{N} a_{ij} y_j(t) + \sum_{k=1}^{M} b_{ik} u_k(t) + w_i \tag{3.1}$$

ou sous la forme matricielle:

$$V(t) = A.Y(t) + B.U(t) + W \tag{3.2}$$

avec:
V(t): vecteur colonne de dimension N×1.
Y(t): vecteur colonne (N×1) représentant les N sorties y_j.
U(t): vecteur colonne (M×1) représentant les M entrées u_k.
W: vecteur colonne (N×1) représentant les N constantes w_i.
A: matrice (N×N) dont l'élément ij est a_{ij}.
B: matrice (N×M) dont l'élément ik est b_{ik}.

3.2.1.2 Le système dynamique

C'est un système linéaire SISO, possédant v_i *comme entrée et* x_i *comme sortie. Donc, on a:*

$$x_i(s) = H(s).v_i(s) \tag{3.3}$$

Cinq choix de H(s) existent:

- $H(s) = 1 \longrightarrow h(t) = \delta(t) \longrightarrow x_i(t) = v_i(t)$
- $H(s) = \frac{1}{S} \longrightarrow h(t) = \left\{ \begin{array}{l} 0 \text{ si } t < 0 \\ 1 \text{ si } t \geq 0 \end{array} \right\} \longrightarrow \dot{x}_i(t) = v_i(t)$
- $H(s) = \frac{1}{1+sT} \longrightarrow h(t) = \frac{1}{T} e^{(\frac{-t}{T})} \longrightarrow T\dot{x}_i(t) + x_i(t) = v_i(t)$
- $H(s) = \frac{1}{\alpha_0 + \alpha_1 s} \longrightarrow h(t) = \frac{1}{\alpha_1} e^{(-(\frac{\alpha_0}{\alpha_1})t)} \longrightarrow \alpha_1 \dot{x}_i(t) + \alpha_0 x_i(t) = v_i(t)$
- $H(s) = e^{-sT} \longrightarrow h(t) = \delta(t - T) \longrightarrow x_i(t) = v_i(t\text{-}T)$

3.2.1.3 *La fonction d'activation*

La fonction d'activation g(.) transforme le signal x_i *non borné à l'instant t en un signal* y_i*:* $y_i = g(x_i)$ *[Kos 92]. Cette fonction est, en général, monotone et non-décroissante. Ainsi, l'augmentation de l'entrée ne peut que faire augmenter la sortie ou la garder constante, elle ne peut jamais faire décroitre le signal.*

La monotonicité de la fonction d'activation implique que les neurones sont non-linéaires mais pas trop. Le choix d'une fonction d'activation linéaire facilite les calculs mais au détriment de la robustesse; tandis que les neurones dont la fonction d'activation est non-linéaire augmentent la capacité du réseau à approximer des fonctions complexes, facilitant la suppression des bruits, mais les calculs et l'analyse risquent d'être compliqués. En plus, ceci favorise l'instabilité du réseau de neurones considéré comme étant un système dynamique[Kos 92].

La fonction d'activation peut avoir l'un des choix doubles:
- Dérivable / non-dérivable.
- Ressemble à un pic / ressemble à un échelon.

- *Positive / moyenne nulle.*

Le premier choix permet de distinguer entre les fonctions lisses et les fonctions dures. Les fonctions d'activations non-dérivables sont utilisées lorsqu'on désire avoir des réponses possédant deux valeurs (la fonction signum par exemple).

Le second choix permet de distinguer entre les fonctions d'activations possédant des valeurs importantes autour de zéro et celles possédant des valeurs importantes loin de zéro.

Le troisième choix distingue entre les fonctions qui varient entre 0 et 1, et celles qui varient entre -1 et 1.

Les fonctions d'activations les plus usuelles, sont:

- $g(x^{k+1}) = \left\{ \begin{array}{c} 1 \text{ si } x^{k+1} > T \\ g(x^k) \text{ si } x^{k+1} = T \\ 0 \text{ si } x^{k+1} < T \end{array} \right\}$
- *Tangente hyperbolique:* $g(x) = tanh(cx) = \frac{e^{cx}-e^{-cx}}{e^{cx}+e^{-cx}}$
- *La fonction linéaire seuillée:* $g(x) = \left\{ \begin{array}{l} 1 \text{ si } cx \geq 1 \\ 0 \text{ si } cx < 0 \\ cx \text{ ailleurs} \end{array} \right\} = \min(1, \max(0, cx))$
- *L'exponentielle seuillée:* $g(x) = \min(1, e^{cx})$
- *La distribution exponentielle:* $g(x) = max\ (\ 0, 1\text{-}e^{-cx})$
- *Une fonction rationnelle:* $g(x) = max\ (\ 0, \frac{x^n}{c+x^n});\ n > 1$

3.2.2 *Les connexions*

Dans le domaine des réseaux de neurones, on parle surtout de champs de neurones pour spécifier un regroupement topologique de plusieurs neurones[Kos 92]. En général, un réseau de neurones contient plusieurs champs de neurones. Le regroupement topologique des neurones dans un champ de neurones est souvent défini par la proximité des neurones par rapport à d'autres neurones qui seront considérés comme origine du champ.

Le cas le plus simple, qu'on va considérer, est le cas où les neurones ne seront pas topologiquement ordonnés. Ils ne seront reliés que par des connexions. Ce manque de structure topologique dans un champ de neurones est appelé "topologie d'ordre zéro". On dénotera par F_x *le champ de neurones par défaut. Un second champ de neurones sera noté par* F_y*, un troisième par* F_z*, etc...*

L'interconnexion de plusieurs neurones sera décrite par un système d'é-

quations différentielles du premier ordre, ou par un système d'équations aux différences qui régissent l'évolution, au cours du temps, des sorties de chaque neurone en fonction de ses entrées. Si nous considérons les champs F_x et F_y, les équations qui régissent le réseau de neurones seront décrites par:

$$\begin{aligned} \dot{x}_1 &= g_1(F_x, F_y, ...) \\ &\vdots \\ \dot{x}_n &= g_n(F_x, F_y, ...) \\ \dot{y}_1 &= h_1(F_x, F_y, ...) \\ &\vdots \\ \dot{y}_p &= h_p(F_x, F_y, ...) \end{aligned} \tag{3.4}$$

Sous la forme vectoctorielle les équations précédentes deviennent:

$$\begin{aligned} \dot{X} &= g(F_x, F_y, ...) \\ \dot{Y} &= h(F_x, F_y, ...) \end{aligned} \tag{3.5}$$

Les variables x_i et y_j sont respectivement les entrées du $i^{ème}$ élément du champ F_x et du $j^{ème}$ neurone du champ F_y. Les arguments des fonctions g_i et h_j incluent toutes les informations conçernant les connexions et les entrées.

On peut noter l'absence de la variable temporelle des équations précédentes, ainsi le modèle du réseau est un système dynamique autonome.

A titre d'exemple, considérons le cas d'un réseau statique($H(s) = 1$), ainsi les équations précédentes deviennent un système d'équations algébriques:

$$\begin{aligned} x(t) &= Ay(t) + Bu(t) + w \\ y(t) &= g(x(t)) \end{aligned} \tag{3.6}$$

Où x est un vecteur de n éléments x_i et $g(x)$ est un vecteur dont les éléments sont $g(x_i)$. Si par exemple, le neurone se comportait comme un filtre passe-bas alors $H(s) = \frac{1}{1+Ts}$, dans ce cas le champ de neurones sera régit par les équations différentielles suivantes:

$$\begin{aligned} T\,\dot{x}\,(t) + x(t) &= Ay(t) + Bu(t) + w \\ y(t) &= g(x(t)) \end{aligned} \tag{3.7}$$

On peut facilement remarquer que les solutions de(3.6)constituent le régime permanent de(3.7).

Une version discrète de(3.7)est:

$$\begin{aligned} Tx(t+1) + (1-T)x(t) &= Ay(t) + Bu(t) + w \\ y(t) &= g(x(t)) \end{aligned} \tag{3.8}$$

Le comportement de ces champs de neurones dépend clairement de la matrice d'interconnexions A et de la forme de H(s).

3.2.3 *Les réseaux de neurones statiques*

L'interconnexion de plusieurs neurones définit un champ de neurones qui peut avoir plusieurs architectures. Ceci dépend du choix de la dynamique du neurone utilisé comme élément de base.

Un réseau de neurones statique est un réseau tel que $H(s) = 1$, c'est un réseau dont la topologie de la connexion ne contient pas de boucles synaptiques fermées. La matrice A est telle que le réseau de neurones puisse être décomposé en plusieurs champs de neurones élémentaires appelés "couches". Les couches différentes des couches d'entrées ou de sortie sont appelés "couches cachées". Un neurone dans une couche ne reçoit ses entrées que des neurones situés plus en amont, dans le sens entrée-sortie, mais pas forcément seulement des neurones situés sur la couche qui le précède immédiatement(figure 3.2).

Par exemple, dans un réseau de neurones qui possède quatre couches, chaque couche contenant N neurones, on peut écrire les vecteurs X, Y, U et W de l'équation(3.6)comme suit:

$$\begin{bmatrix} x^1(t) \\ x^2(t) \\ x^3(t) \\ x^4(t) \end{bmatrix} = A. \begin{bmatrix} y^1(t) \\ y^2(t) \\ y^3(t) \\ y^4(t) \end{bmatrix} + B. \begin{bmatrix} u^1(t) \\ u^2(t) \\ u^3(t) \\ u^4(t) \end{bmatrix} + \begin{bmatrix} w^1(t) \\ w^2(t) \\ w^3(t) \\ w^4(t) \end{bmatrix} \tag{3.9}$$

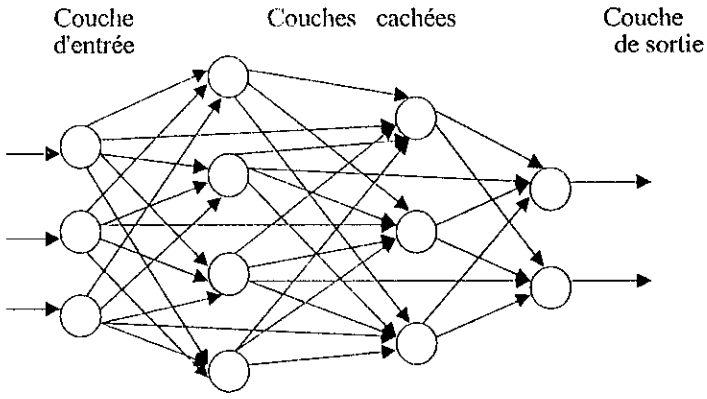


Figure 3.2: *Architecture d'un réseau statique.*

Les indices supèrieurs indiquent le numéro de la couche. La structure des matrices A et B est la suivante:

$$A = \begin{bmatrix} 0_{NN} & 0_{NN} & 0_{NN} & 0_{NN} \\ A^2 & 0_{NN} & 0_{NN} & 0_{NN} \\ 0_{NN} & A^3 & 0_{NN} & 0_{NN} \\ 0_{NN} & 0_{NN} & A^4 & 0_{NN} \end{bmatrix} ; \quad B = \begin{bmatrix} B^1 & 0_{NN} & 0_{NN} & 0_{NN} \\ 0_{NN} & 0_{NN} & 0_{NN} & 0_{NN} \\ 0_{NN} & 0_{NN} & 0_{NN} & 0_{NN} \\ 0_{NN} & 0_{NN} & 0_{NN} & 0_{NN} \end{bmatrix}$$

0_{NN}: *est la matrice nulle de dimension* ($N \times N$).
0_{NM}: *est la matrice nulle de dimension* ($N \times M$).
A^2, A^3, A^4: *matrices des poids dont les dimensions sont* $N \times N$.
B^1: *matrice des poids de dimension* $N \times M$.
Pour la première couche, on a:

$$\begin{aligned} x^1(t) &= B^1 u^1(t) + w^1 \\ y^1(t) &= g(x^1(t)) \end{aligned} \tag{3.10}$$

Pour les couches 2, 3 et 4, on aura:

$$\begin{aligned} x^i(t) &= A^i y^{i-1}(t) + w^i \\ y^i(t) &= g(x^i(t)) \end{aligned} \tag{3.11}$$

avec $i = 2, 3, 4$.

3.2.4 *Les réseaux de neurones dynamiques*

Les réseaux de neurones dynamiques ou récurrents sont différents des réseaux de neurones statiques car leur architecture contient un bouclage(la topologie des connexions est bouclée). En général, la sortie de chaque neurone est réinjectée en entrée de chaque neurone grâce à des poids variables. Un réseau de neurones composé de ces éléments, dont tous les poids sont en général non nuls, est dit entièrement connecté. Le réseau, étant fondamentalement dynamique, ne contient qu'une seule couche(figure 3.3).

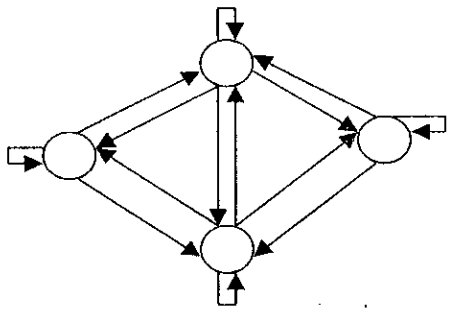


Figure 3.3: *Architecture d'un réseau récurrent.*

L'introduction de ce bouclage produit un réseau de neurones dynamique avec différents points d'équilibres. Il est régit par l'équation suivante:

$$\begin{aligned} \dot{x}(t) &= F(x(t), u(t), \theta) \\ y(t) &= G(x(t), \theta) \end{aligned} \tag{3.12}$$

avec:

x: l'état.

u: les entrées externes.

θ: vecteur des paramètres du réseau.

F: fonction qui représente la structure du réseau.

G: fonction qui représente la relation entre les états et les sorties.

Contrairement à un réseau de neurones statique qui donne la même sortie chaque fois qu'on lui présente la même entrée, un réseau de neurones dynamique peut donner une sortie différente en lui présentant la même entrée à des instants différents. Ceci dépend des entrées qu'on lui a présentées précédemment. Ainsi, l'ordre de la présentation des entrées-sorties est tout aussi important que les exemples eux mêmes.

3.3 *Apprentissage des réseaux de neurones*

On appelle apprentissage, l'opération par laquelle le réseau de neurones acquiert la capacité de faire certaines tâches en modifiant ses paramètres internes en utilisant un algorithme d'adaptation paramétrique appelé "algorithme d'apprentissage". On dit que le réseau a appris la paire entrée-sortie (x_i, y_i) *s'il répond avec* y_i *si* x_i *est présentée comme entrée. La paire*(x_i, y_i) *représente un échantillon d'une fonction* f*:* $R^n \longrightarrow R^p$*, la fonction* f *associe* p *vecteurs* y *à* n *vecteurs* x*. Le réseau a appris la fonction* f *s'il répond par* y*, avec* $y=f(x)$*, si* x *varie dans le domaine de définition de* f*. On dit que le réseau a partiellement appris s'il répond par* y'*, qui est proche de* $y = f(x)$*, lorsqu'on lui présente* x' *qui est proche de* x*.*

On peut subdiviser le processus d'apprentissage en trois classes[Kos 92]:
- *Apprentissage supervisé " supervised learning ".*
- *Apprentissage non-supervisé " unsupervised learning ".*
- *"Reinforcement learning".*

3.3.1 *Apprentissage non-supervisé*

L'apprentissage non-supervisé représente la capacité du réseau de neurones à apprendre sans connaître la sortie désirée. Ce type d'apprentissage possède souvent une moindre complexité dans les calculs en comparaison avec l'apprentissage supervisé. Il apprend rapidement, parfois un seul passage sur des données bruitées suffit. Il est particulièrement souhaitable dans les processus rapides où on n'a pas suffisamment de temps ou d'informations[Kos 92].

3.3.2 *Apprentissage supervisé*

Dans l'apprentissage supervisé la sortie désirée est connue, et c'est en minimisant l'erreur entre la sortie désirée et la sortie du réseau qu'on modifie les paramètres internes du réseau de neurones. Dans ce cas, on devrais avoir un ensemble de paires entrées-sorties désirées qu'on appelle exemples (figure 3.4).

Donc, le but est de produire un réseau de neurones qui implante la fonction inconnue f*, quand un nombre suffisant d'exemples est à notre disposition. Le processus est automatisé grâce à l'algorithme d'apprentissage. Cette implantation est souvent une approximation de* f*.*

L'un des algorithmes les plus répandus est celui de la "backpropagation" (rétropropagation). Cet algorithme change les poids d'un réseau dont l'architecture est fixée par le superviseur, à chaque fois qu'un exemple $y_i = f(x_i)$

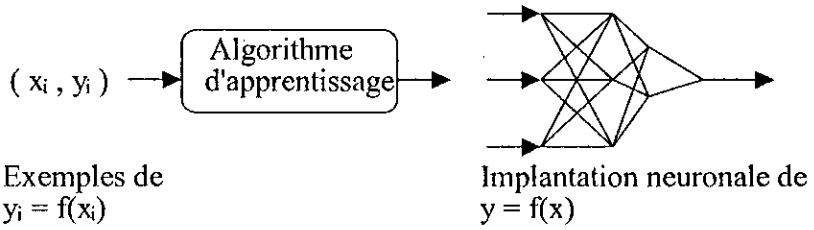


Figure 3.4: *Apprentissage à partir d'exemples.*

est présenté. Ce changement est fait de telle sorte à minimiser l'erreur entre la sortie désirée y_i et la réponse du réseau à une entrée x_i. Ceci est réalisé grâce à la descente du gradient. A chaque itération le signal d'entrée se propage dans le réseau dans le sens entrée-sortie, une sortie est ainsi obtenue, l'erreur entre cette sortie et la sortie désirée est calculée puis rétropropagée dans le sens sortie-entrée. Malheureusement, cet algorithme souffre des problèmes de la méthode de la descente du gradient: les paramètres du réseau se trouvent souvent bloqués dans un minimum local. La figure 3.5 illustre l'algorithme.

L'algorithme "backpropagation" peut être appliqué aux réseaux de neurones statiques comme aux réseaux de neurones dynamiques.

3.3.2.1 Apprentissage des réseaux de neurones statiques

Un réseau de neurones statique associe un vecteur de sorties à chaque vecteur d'entrées. Pour un vecteur d'entrées x, le réseau lui associe un vecteur de sorties $\hat{y}$ suivant la relation:

$$\hat{y} = \eta(x) \tag{3.13}$$

où η est la fonction qui représente le réseau de neurones.
Le but de l'apprentissage supervisé est de minimiser l'erreur quadratique:

$$e^2(t) = (y - \hat{y})^2 \tag{3.14}$$

la méthode utilisée pour cela est la "backpropagation". Plusieurs variantes ont été proposées. Entre autres, on peut citer les algorithmes suivants:

- *"Backpropagation" avec momentum.*
- *"Robust backpropagation", etc...*

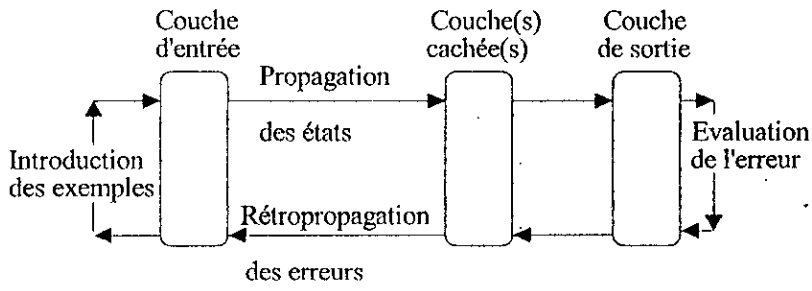


Figure 3.5: *Apprentissage des réseaux de neurones en utilisant l'algorithme "backpropagation".*

3.3.2.2 Présentation de l'algorithme "backpropagation"

Le but de l'apprentissage est de déterminer les poids synaptiques qui définissent les connexions entre les neurones pour produire les sorties désirées. Le déroulement de l'apprentissage nécéssite un ensemble d'entrées et de sorties organisées en paires(entrée, sortie correspondante). Cette ensemble est appelé ensemble des couples d'apprentissage[Fre 92].

L'apprentissage par l'algorithme backpropagation consiste à changer les poids des connexions de façon à minimiser l'erreur carrée moyenne de la couche de sortie.

Pour l'exemple d'un réseau de neurones à trois couches(figure 3.6), l'algorithme se présente comme suit:

- Etape 1: Initialisation aléatoire des pondérations b_{ij}.

- Etape 2: Application du vecteur d'entrée $x_p=(x_{p1},x_{p2},\ldots,x_{pN})^T$ *à la couche d'entrée.*

- Etape 3: Calcul des entrées des neurones de la couche cachée:

$$net^c_{pj}=\sum_{i=1}^{N} b^c_{ji}x_{pi}+w^c_j \tag{3.15}$$

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

$$i_{pj}=f^c_j(net^c_{pj}) \tag{3.16}$$

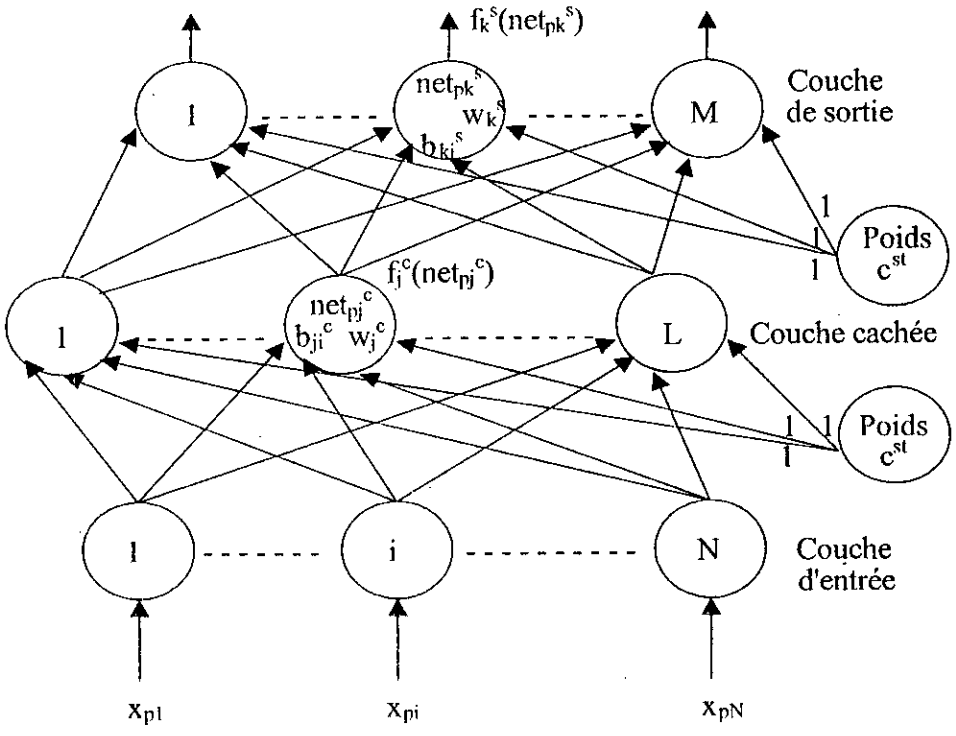


Figure 3.6: *Un réseau de neurones à trois couches.*

- *Etape 5: Passer à la couche de sortie et calculer les entrées de chaque neurone:*

$$net^s_{pk} = \sum_{j=1}^{L} b^s_{kj} i_{pj} + w^s_k \tag{3.17}$$

- *Etape 6: Calcul des sorties des neurones de la couche de sortie:*

$$o_{pk} = f^s_k(net^s_{pk}) \tag{3.18}$$

- *Etape 7: Calcul de l'erreur de la couche de sortie:*

$$\delta^s_{pk} = (y_{pk} - o_{pk}) f'^s_k(net^s_{pk}) \tag{3.19}$$

- *Etape 8: Calcul de l'erreur de la couche cachée:*

$$\delta^c_{pj} = f'^c_j(net^c_{pj}) \sum_k \delta^s_{pk} b^s_{kj} \tag{3.20}$$

L'erreur de la couche cachée est calculée avant la modification des poids de la couche de sortie.

- *Etape 9: Modification des poids de la couche de sortie:*

$$b^s_{kj}(t+1) = b^s_{kj}(t) + \eta \delta^s_{pk} i_{pj} \tag{3.21}$$

- *Etape 10: Modification des poids de la couche cachée:*

$$b^c_{ji}(t+1) = b^c_{ji}(t) + \eta \delta^c_{pj} x_i \tag{3.22}$$

L'ordre de modification des poids dans une même couche n'est pas important.

- *Etape 11: Calcul l'erreur:*

$$E_p = \frac{1}{2} \sum_{k=1}^{M} \delta^2_{pk} \tag{3.23}$$

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

3.3.2.3 Problèmes posés par l'algorithme "backpropagation"

L'apprentissage par l'algorithme "backpropagation" pose plusieurs problèmes[Was 89]. Les principaux, qu'on peut citer, sont comme suit:

a-L'architecture du réseau: *Il n'existe pas de règle gènèrale pour déterminer la structure du réseau(le nombre de couches cachées et le nombre de neurones par couche), sachant que le problème critique pendant l'apprentissage est de trouver un réseau assez large pour apprendre mais assez petit pour gènèraliser.*

b-Le temps d'apprentissage: *Le temps nécéssaire à l'apprentissage augmente avec le nombre de couples d'apprentissage, ce qui diminue la vitesse de convergence.*

c-La convergence de l'algorithme: *Aucune preuve mathématique sur la convergence de cet algorithme vers un minimum global n'existe. Du fait que cette méthode utilise la descente du gradient, la recherche d'un minimum global, sur la surface de l'erreur dans le domaine des poids, peut présenter un problème si cette surface possède des minimums locaux qui peuvent emprisonner l'algorithme.*

d-Le pas de correction des poids: *Si le pas de correction des poids est trés petit l'apprentissage nécéssiterait, alors, un temps trés important. Par contre, si ce même pas est trés grand le réseau devient oscillatoire, ce qui compromet sa convergence. La solution à ce problème est de choisir un pas variable, initialisé à une grande valeur comprise entre 0 et 1, et qui sera diminué jusqu'à une valeur minimale positive fixée au préalable.*

e-La saturation du réseau: *Si les poids prennent de grandes valeurs, les sorties deviennent grandes et se rapprochent de la zone de saturation de la fonction d'activation. La solution à ce problème est un compromis à faire entre:*

- *Le choix d'un pas de correction petit.*
- *L'initialisation des poids à de trés petites valeurs.*

3.4 *Structures de commande*

Le problème qui se pose est la manière d'intégrer le réseau de neurones à une structure de réglage et comment effectuer son apprentissage en vue de sa préparation pour la fonction qui lui est assignée.

Les deux plus importantes structures de commande sont:

- *Le contrôleur feedforward.*
- *Le contrôleur feedback.*

3.4.1 *L'apprentissage du contrôleur feedforward*

3.4.1.1 Apprentissage généralisé(off-line)

La structure est représentée sur la figure 3.7. Son but est d'identifier le modèle inverse du système, c'est à dire que le réseau de neurones fournira la commande qu'il faut appliquer au système par la donnée de la consigne[Ren 95].

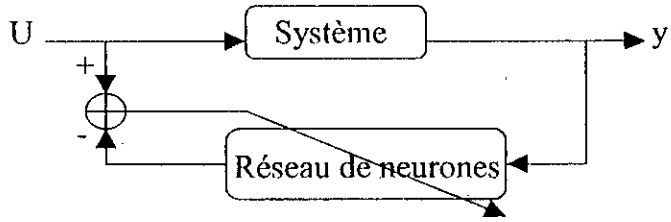


Figure 3.7: *Structure de l'apprentissage généralisé(off-line).*

L'inconvenient de cette technique est le fait de ne pas connaître à priori les commandes correspondantes aux sorties désirées.

3.4.1.2 Apprentissage spécialisé(on-line)

Dans cette structure (figure 3.8), le réseau de neurones précède le système. Il reçoit la consigne c pour délivrer une commande u qu'on applique au système. L'apprentissage est effectué de telle façon à minimiser l'erreur $e = c - y$ [Ren 95].

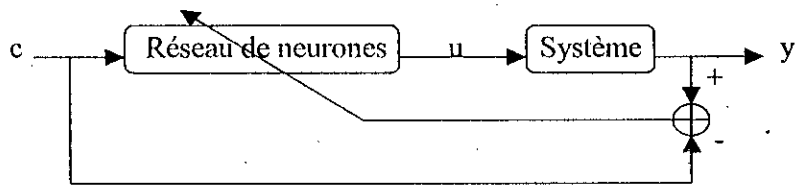


Figure 3.8: *Structure de l'apprentissage spécialisé (on-line).*

3.4.2 *L'apprentissage du contrôleur feedback*

Dans cette structure(figure 3.9), le réseau de neurones est utilisé comme contrôleur dynamique du système, avec comme entrée le signal d'erreur et son signal de sortie la commande à appliquer au système.

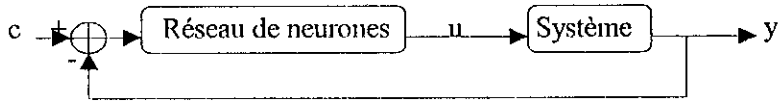


Figure 3.9: *Le contrôleur feedback.*

3.5 *CONCLUSION*

Les possibilités d'apprentissage et de généralisation, qu'ont les réseaux de neurones pour approximer n'importe quelle fonction, donnent de nouvelles solutions, particulièrement, pour les problèmes de commande de systèmes dont le traitement rencontre, de nos jours, des difficultés d'ordre pratique et théorique.

Dans ce chapitre, on a pu voir que les réseaux de neurones sont des approximateurs universels, donc ils peuvent apporter des solutions aux problèmes posés à la commande floue: limitations pratiques, dérivation des règles floues, etc...

Chapitre 4

SYNTHESE DU CONTROLEUR

4.1 *INTRODUCTION*

Les systèmes d'inférence floue sont des approximateurs universels et sont capables de représenter une connaissance humaine. Ces deux propriétés ont leurs limites inhérentes et peuvent devenir antinomiques dans certaines circonstances. Il est souvent nécéssaire de trouver un compromis et de privilégier un aspect au détriment de l'autre. On peut examiner rapidement les limites et les aspects contradictoires des propriétés d'approximation universelle et de représentation des connaissances:

- Limites de la représentation des connaissances:

Les systèmes d'inférence floue traduisent la connaissance humaine sous la forme de règles telles que "si telle situation alors telle conclusion", règles qui constituent une part importante de la connaissance. Mais toute la connaissance d'un expert ne peut être entièrement verbalisée: le savoir-faire et l'intuition ne sont pas transmissibles facilement.

- Limites de l'approximation universelle:

La plupart des systèmes d'inférence floue sont des approximateurs universels [Cas 95], [Wan 92]. Malheureusement, la précision dans l'approximation se paye par l'augmentation considérable de la base de règles: le nombre de règles croit exponentielement avec le nombre d'entrées.

Pour remédier à ce problème, un système hybride ou neuro-linguistique a été proposé.

4.2 ASSOCIATION DU NEURONAL ET DU FLOU

L'association des réseaux de neurones et de la logique floue présente une technique trés puissante pour la commande des systèmes dynamiques[Kos 92], [Dot 90].

Grâce aux grandes capacités d'apprentissage des réseaux de neurones, cette technique est capable de générer, souvent, la bonne commande pour n'importe quelle loi analytique, ce qui constitue un véritable support à l'implémentation des contrôleurs flous.

L'implémentation de la commande peut se faire d'une manière explicite [Hor 90] ou d'une manière implicite. Dans la suite du travail on utilisera la deuxième méthode.

Dans ce cas, le contrôleur est constitué soit de plusieurs réseaux traitant chacun un module du contrôleur flou(la fuzzification, l'inférence et la défuzzification)qu'on appelera dans la suite de notre travail le contrôleur neuro-linguistique 1(CNL 1), soit d'un seul réseau traitant le tout et qu'on appelera le contrôleur neuro-linguistique 2(CNL 2).

4.3 CONCEPTION DU CONTROLEUR NEURO-LINGUISTIQUE

4.3.1 Réalisation de la partie linguistique

Dans la conception d'un contrôleur neuro-linguistique, la première phase consiste à réaliser la partie linguistique, c'est à dire un contrôleur flou. Ceci a été exposé au premier paragraphe de ce document[Rao 97], [Büh 94], [Yos 96.b].

4.3.1.1 Définition des variables linguistiques

La première étape, dans la réalisation d'un contrôleur flou, est le choix des variables linguistiques. On utilise dans ce travail deux variables d'entrées

$$e = y_2 - y_1 \tag{4.1}$$

et sa dérivée

$$de = d(y_2 - y_1) \tag{4.2}$$

avec:

y_1: *le déplacement de l'axe de la roue.*
y_2: *le déplacement du chassis.*

Par contre, la variable de sortie du contrôleur flou est la force f appliquée par l'actionneur.

4.3.1.2 Division des variables linguistiques en classes

On divise chacune des variables linguistiques e, de et f en sept classes(ou ensembles flous)qui sont:

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

Le choix du nombre de classes est guidé par le soucis de faire un compromis entre une commande assez fine et une base de règles pas trés grande.

4.3.1.3 Choix des fonctions d'appartenance

Les fonctions d'appartenance utilisées pour la représentation des variables linguistiques e, de et f sont du type triangulaire(figure 2.2). Ce choix est dans le but de faciliter une éventuelle codification, par réseaux de neurones, des différentes variables linguistiques. La distribution des classes est uniforme et équidistante(figure 4.1).

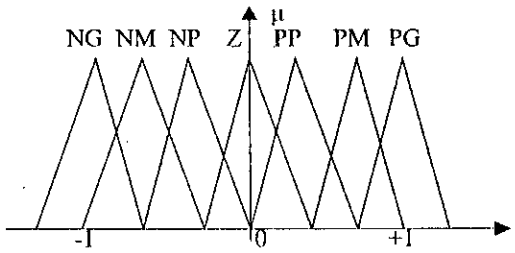


Figure 4.1: *Fonctions d'appartenance pour les variables e , de et f.*

4.3.1.4 Etablissement des règles floues

La loi de commande dépend essentiellement de l'établissement des règles floues, c'est à dire d'une liste de décisions à appliquer dans tous les cas

qui peuvent se présenter au contrôleur. La formulation concrète de ces règles floues dépend du comportement statique et dynamique du système à commander,, ainsi que des objectifs envisagés. L'expérience joue un rôle important.

Etant donné que chaque variable linguistique d'entrée est divisée en sept classes, on peut formuler au maximum quarante neuf règles floues. En général, on travail avec des règles d'inférence complètes où toutes les positions de la matrice d'inférence sont occupées. Cependant, il peut être judicieux de travailler avec des règles d'inférence incomplètes pour réduire le nombre de règles, ce qui réduit le temps de traitement.

La matrice d'inférence utilisée est présentée à la figure 4.2:

f \ e	NG	NM	NP	Z	PP	PM	PG
NG	NG	NG	NG	NG	NM	NP	Z
NM	NG	NG	NG	NM	NP	Z	PP
NP	NG	NG	NM	NP	Z	PP	PM
Z	NG	NM	NP	Z	PP	PM	PG
PP	NM	NP	Z	PP	PM	PG	PG
PM	NP	Z	PP	PM	PG	PG	PG
PG	Z	PP	PM	PG	PG	PG	PG

de

Figure 4.2: *Matrice d'inférence.*

4.3.2 *Conception du CNL 1*

Dans ce cas, le contrôleur neuro-linguistique est constitué de quatre réseaux de neurones traitant chacun un module du contrôleur flou: la fuzzification de la variable d'entrée e, la fuzzification de la variable d'entrée de, l'inférence et la défuzzification.

4.3.2.1 Codification de la variable d'entrée e

Le réseau de neurones utilisé pour la codification de la variable d'entrée e est un réseau statique à une seule couche cachée. Son architecture est présentée

sur la figure 4.3. Les fonctions d'activations utilisées dans la couche d'entrée ainsi que la couche cachée sont de types tangentes hyperboliques. Par contre, dans la couche de sortie les fonctions d'activations sont de types linéaires.

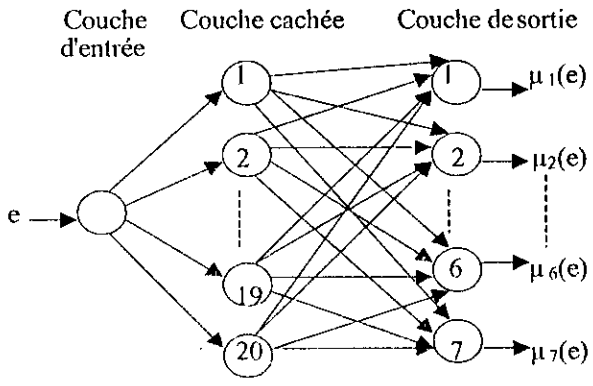


Figure 4.3: *Réseau codifiant la variable d'entrée e.*

a-Distribution des classes: *Puisque les fonctions d'appartenance utilisées pour la représentation de la variable linguistique e sont du type triangulaire, et la distribution des classes est uniforme et équidistante(figure 4.1), alors on peut en déduire les équations des différentes classes:*

- NG: a = -1; b = -1; c = -0.67 →

$$\mu_1(e) = \max(\frac{-0.67 - e}{0.33}, 0) \tag{4.3}$$

- NM: a = -1; b = -0.67; c = -0.33 →

$$\mu_2(e) = \max[\min(\frac{e+1}{0.33}, \frac{-0.33 - e}{0.34}), 0] \tag{4.4}$$

- NP: a = -0.67; b = -0.33; c = 0 →

$$\mu_3(e) = \max[\min(\frac{e+0.67}{0.34}, \frac{-e}{0.33}), 0] \tag{4.5}$$

- Z: a = -0.33; b = 0; c = 0.33 →

$$\mu_4(e) = \max[\min(\frac{e+0.33}{0.33}, \frac{0.33 - e}{0.33}), 0] \tag{4.6}$$

- *PP:* $a = 0;\ b = 0.33;\ c = 0.67 \rightarrow$

$$\mu_5(e) = \max[\min(\frac{e}{0.33}, \frac{0.67 - e}{0.34}), 0] \tag{4.7}$$

- *PM:* $a = 0.33;\ b = 0.67;\ c = 1 \rightarrow$

$$\mu_6(e) = \max[\min(\frac{e - 0.33}{0.34}, \frac{1 - e}{0.33}), 0] \tag{4.8}$$

- *PG:* $a = 0.67;\ b = 1;\ c = 1 \rightarrow$

$$\mu_7(e) = \max(\frac{e - 0.67}{0.33}, 0) \tag{4.9}$$

b-Apprentissage du réseau de neurones: *L'apprentissage du réseau utilisé(figure 4.3)nécéssite des exemples d'apprentissage sous forme de couples de données entrée-sortie du type*$(e, \begin{bmatrix} \mu_1(e) \\ \cdot \\ \cdot \\ \mu_7(e) \end{bmatrix})$*. Le choix de ces exemples d'apprentissage est fait à partir de la figure 4.1. Les fonctions d'appartenance utilisées sont du type triangulaire, donc l'interpolation de chacune d'elles nécéssite au minimum trois points.*

L'apprentissage est arrêté à 100000 itérations, et l'erreur par composante en sortie est 10^{-4}.

4.3.2.2 Codification de la variable d'entrée de

Le réseau de neurones utilisé pour la codification de la variable d'entrée de est un réseau statique à une seule couche cachée. Son architecture est présentée sur la figure 4.4. Les fonctions d'activations utilisées dans la couche d'entrée ainsi que la couche cachée sont de types tangentes hyperboliques. Par contre, dans la couche de sortie les fonctions d'activations sont de types linéaires.

a-Distribution des classes: *Puisque les fonctions d'appartenance utilisées pour la représentation de cette variable linguistique sont du type triangulaire, et la distribution des classes est uniforme et équidistante(figure 4.1), alors on peut en déduire les équations des différentes classes:*

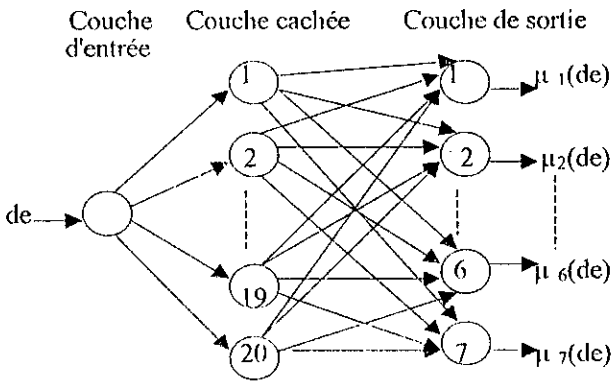


Figure 4.4: *Réseau codifiant la variable d'entrée de.*

- *NG: a = -1; b = -1; c = -0.67* $\rightarrow$

$$\mu_1(de) = \max(\frac{-0.67 - de}{0.33}, 0) \tag{4.10}$$

- *NM: a = -1; b = -0.67; c = -0.33* $\rightarrow$

$$\mu_2(de) = \max[\min(\frac{de + 1}{0.33}, \frac{-0.33 - de}{0.34}), 0] \tag{4.11}$$

- *NP: a = -0.67; b = -0.33; c = 0* $\rightarrow$

$$\mu_3(de) = \max[\min(\frac{de + 0.67}{0.34}, \frac{-de}{0.33}), 0] \tag{4.12}$$

- *Z: a = -0.33; b = 0; c = 0.33* $\rightarrow$

$$\mu_4(de) = \max[\min(\frac{de + 0.33}{0.33}, \frac{0.33 - de}{0.33}), 0] \tag{4.13}$$

- *PP: a = 0; b = 0.33; c = 0.67* $\rightarrow$

$$\mu_5(de) = \max[\min(\frac{de}{0.33}, \frac{0.67 - de}{0.34}), 0] \tag{4.14}$$

- *PM: a = 0.33; b = 0.67; c = 1* $\rightarrow$

$$\mu_6(de) = \max[\min(\frac{de - 0.33}{0.34}, \frac{1 - de}{0.33}), 0] \tag{4.15}$$

- *PG: a = 0.67; b = 1; c = 1* $\rightarrow$

$$\mu_7(de) = \max(\frac{de - 0.67}{0.33}, 0) \tag{4.16}$$

b-Apprentissage du réseau de neurones: *L'apprentissage du réseau utilisé(figure 4.4)nécéssite des exemples d'apprentissage sous forme de couples de données entrée-sortie du type(de,* $\begin{bmatrix} \mu_1(de) \\ \cdot \\ \cdot \\ \mu_7(de) \end{bmatrix}$ *). Le choix de ces exemples d'apprentissage est fait à partir de la figure 4.1. Les fonctions d'appartenance utilisées sont du type triangulaire, donc l'interpolation de chacune d'elles nécéssite au minimum trois points.*

L'apprentissage est arrêté à 100000 itérations, et l'erreur par composante en sortie est 10^{-4}.

4.3.2.3 Codification de la variable de sortie f

Le réseau de neurones utilisé pour la codification de la variable de sortie f est un réseau statique à une seule couche cachée. Son architecture est présentée sur la figure 4.5. Les fonctions d'activations utilisées dans la couche d'entrée ainsi que la couche cachée sont de types tangentes hyperboliques. Par contre, dans la couche de sortie les fonctions d'activations sont de types linéaires.

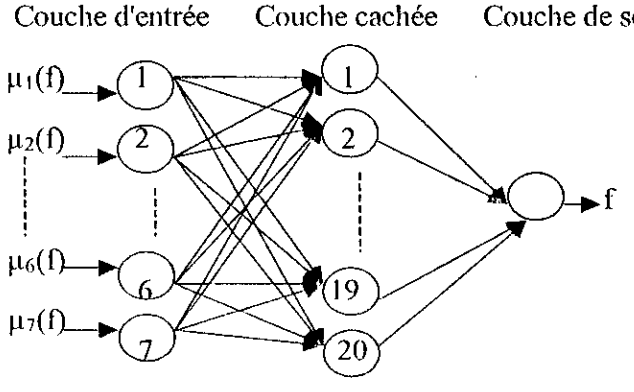


Figure 4.5: *Réseau codifiant la variable de sortie f.*

a-Distribution des classes: *Puisque les fonctions d'appartenance utilisées pour la représentation de la variable linguistique f sont du type trian-*

gulaire, et la distribution des classes est uniforme et équidistante(figure 4.1), alors on peut en déduire les équations des différentes classes:

- NG: a = -1; b = -1; c = -0.67 →

$$\mu_1(f) = \max(\frac{-0.67 - f}{0.33}, 0) \tag{4.17}$$

- NM: a = -1; b = -0.67; c = -0.33 →

$$\mu_2(f) = \max[\min(\frac{f+1}{0.33}, \frac{-0.33 - f}{0.34}), 0] \tag{4.18}$$

- NP: a = -0.67; b = -0.33; c = 0 →

$$\mu_3(f) = \max[\min(\frac{f+0.67}{0.34}, \frac{-f}{0.33}), 0] \tag{4.19}$$

- Z: a = -0.33; b = 0; c = 0.33 →

$$\mu_4(f) = \max[\min(\frac{f+0.33}{0.33}, \frac{0.33 - f}{0.33}), 0] \tag{4.20}$$

- PP: a = 0; b = 0.33; c = 0.67 →

$$\mu_5(f) = \max[\min(\frac{f}{0.33}, \frac{0.67 - f}{0.34}), 0] \tag{4.21}$$

- PM: a = 0.33; b = 0.67; c = 1 →

$$\mu_6(f) = \max[\min(\frac{f - 0.33}{0.34}, \frac{1 - f}{0.33}), 0] \tag{4.22}$$

- PG: a = 0.67; b = 1; c = 1 →

$$\mu_7(f) = \max(\frac{f - 0.67}{0.33}, 0) \tag{4.23}$$

b-*Apprentissage du réseau de neurones:* *L'apprentissage du réseau utilisé(figure 4.5)nécéssite des exemples d'apprentissage sous forme de couples de données entrée-sortie du type(* $\begin{bmatrix} \mu_1(f) \\ . \\ . \\ \mu_7(f) \end{bmatrix}$ *, f). Le choix de ces exemples d'apprentissage est fait à partir de la figure 4.1. Les fonctions d'appartenance utilisées sont du type triangulaire, donc l'interpolation de chacune d'elles nécéssite au minimum trois points.*

L'apprentissage est arrêté à 50000 itérations, et l'erreur par composante en sortie est 10^{-5}.

4.3.2.4 Réseau de neurones représentant la matrice d'inférence

Le réseau de neurones utilisé pour la représentation de la matrice d'inférence(figure 4.2)est un réseau statique à deux couches cachées. Son architecture est présentée sur la figure 4.6. Les fonctions d'activations utilisées dans la couche d'entrée ainsi que les deux couches cachées sont de types tangentes hyperboliques. Par contre, dans la couche de sortie les fonctions d'activations sont de types linéaires.

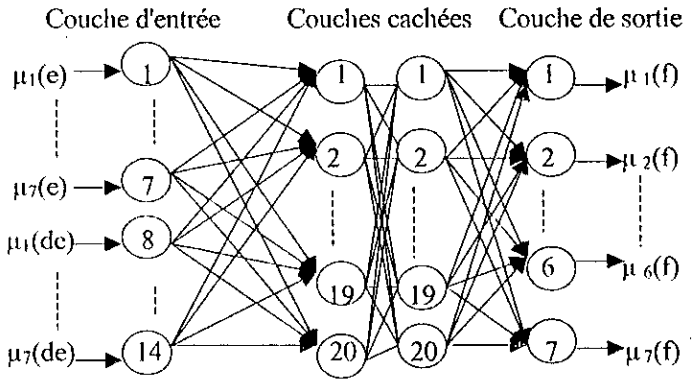


Figure 4.6: *Réseau représentant la matrice d'inférence.*

Apprentissage du réseau de neurones: *L'apprentissage du réseau utilisé(figure 4.6)nécéssite des exemples d'apprentissage sous forme de couples de données entrée-sortie du type(* $\begin{bmatrix} \mu_1(e) \\ . \\ . \\ \mu_7(e) \\ \mu_1(de) \\ . \\ . \\ \mu_7(de) \end{bmatrix}, \begin{bmatrix} \mu_1(f) \\ . \\ . \\ \mu_7(f) \end{bmatrix}$ *). Le choix de ces exemples d'apprentissage est fait à partir de la matrice d'inférence(figure 4.2).*

Pour chaque régle, on choisit au moins trois exemples représentatifs de toute la classe pour chaque variable linguistique.

Par exemple, pour la régle suivante:

si e est NM et de est PP alors f est NP

On prend, au moins, trois exemples de NM pour e, le même nombre d'exemples de PP pour de et le même nombre d'exemples de NP pour f. On établit, ensuite, les différentes combinaisons d'exemples correspondants à cette régle.

On répète la même procédure pour chaque régle de la matrice d'inférence pour obtenir l'ensemble des couples de données entrée-sortie nécéssaires pour l'apprentissage du réseau de neurones(figure 4.6).

L'apprentissage est arrêté à 5×10^6 itérations, et l'erreur par composante en sortie est 10^{-4}.

4.3.3 *Conception du CNL 2*

Dans ce cas, le contrôleur neuro-linguistique est constitué d'un seul réseau de neurones. Ses entrées sont de type continu et sont fournies par la structure de commande, alors que sa sortie est la commande à appliquer au système.

Le réseau de neurones utilisé pour cela est un réseau statique à une seule couche cachée. Son architecture est présentée sur la figure 4.7. Les fonctions d'activations utilisées dans la couche d'entrée ainsi que la couche cachée sont de types tangentes hyperboliques. Par contre, dans la couche de sortie la fonction d'activation est de type linéaire.

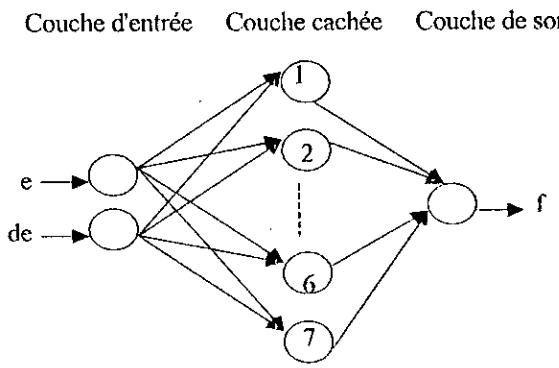


Figure 4.7: *Réseau représentant le contrôleur flou.*

Apprentissage du réseau de neurones: *L'apprentissage du réseau utilisé(figure 4.7)nécéssite des exemples d'apprentissage sous forme de couples de données entrée-sortie du type(* $\begin{bmatrix} e \\ de \end{bmatrix}$ *, f). Le choix de ces exemples d'apprentissage est fait à partir de la structure de commande.*

L'apprentissage est arrêté à 337 itérations, et l'erreur en sortie est 10^{-5}.

4.4 *CONCLUSION*

Dans ce chapitre, nous nous sommes intéréssés à la synthèse des contrôleurs neuro-linguistiques. Durant cette conception, nous avons opté pour des choix qui pourraient, éventuellement, être changés, tels que:

- Le nombre et le choix des variables linguistiques.

- Les fonctions d'appartenance.

- Le nombre et la distribution des classes.

- La méthode d'inférence.

- La méthode de représentation de la partie linguistique par des réseaux de neurones.

- La structure du réseau de neurones utilisé.

- L'algorithme d'apprentissage.

Il en découle un nombre important de contrôleurs neuro-linguistiques qui pourraient être réalisés.

Chapitre 5

FORMULATION DU PROBLEME DU VEHICULE

5.1 *INTRODUCTION*

Une situation commune pour les systèmes roulants est qu'une partie est soumise à une perturbation, c'est à dire à la variation du profil de la route, et une autre partie du même système, comprennant le compartiment passager ou bien celui du fret, doit être isolée des vibrations pour obtenir un certain niveau de confort ou pour protéger les chargements fragiles.

Durant plusieurs années, les suspensions passives ont été utilisées pour remédier à ce problème. Ces dérnières consistent en l'utilisation de ressorts et d'amortisseurs.

Actuellement, une autre solution technologique consiste en l'utilisation des suspensions actives. L'utilisation d'éléments actifs nécéssite la génération d'un signal de commande pour créer des forces entre les différentes parties du système roulant. Ce signal de commande doit être fonction de différentes mesures.

Dans le présent chapitre, nous présenterons le modèle, à utiliser, de la suspension active d'un véhicule.

5.2 *PROBLEME DU QUART DU VEHICULE*

Le modèle du quart du vehicule étudié est donné par la figure 5.1[Lou 88].

La masse M correspond au quart de la masse totale du vehicule. La masse m représente la masse de la roue. Les deux masses sont reliées entre elles par un actionneur ayant une dynamique supposée idéale.

La force de commande f, développée par l'actionneur, est supposée être appliquée avec la même intensité mais de sens opposé aux deux masses M et m. Le pneu est représenté par un ressort linéaire de raideur h et est supposé assurer un contact ferme avec la route pour satisfaire l'exigence de sécurité.

Sous l'excitation de la composante verticale de la surface de la route, y_a, on suppose que les deux masses m et M ne peuvent effectuer que des déplacements verticaux, respectivement y_1 et y_2, à partir des positions initiales d'équilibre.

Le système, supposé linéaire et invariant dans le temps[Lou 88], est décrit par:

$$\begin{aligned} \dot{x}\,(t) &= A.x(t) + B.f(t) + Dw(t) \\ y(t) &= C.x(t) \\ x(0^+) &= x_0 \text{ et } x(0^-) = 0 \end{aligned} \tag{5.1}$$

avec:

$x = [\, y_1 \;\; y_2 \;\; \dot{y}_1 \;\; \dot{y}_2]^T$: *vecteur d'état.*

f: la commande.

$w = [y_a]^T$: *vecteur des perturbations représentant le profil de la route.*

Dans le cas d'une suspension active pure(figure 5.1), les matrices A, B, C et D sont comme suit:

$$A = \begin{bmatrix} 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \\ \frac{-h}{m} & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix},\; B = \begin{bmatrix} 0 \\ 0 \\ \frac{-1}{m} \\ \frac{1}{M} \end{bmatrix},\; C = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \end{bmatrix} \; et \; D = \begin{bmatrix} 0 \\ 0 \\ \frac{h}{m} \\ 0 \end{bmatrix}.$$

Par contre, dans le cas où la suspension active est associée en parallèle à une suspension conventionnelle formée d'un ressort de raideur k et d'un amortisseur d(figure 5.2), les matrices A, B, C et D sont données par:

$$A = \begin{bmatrix} 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \\ \frac{-(h+k)}{m} & \frac{k}{m} & \frac{-d}{m} & \frac{d}{m} \\ \frac{k}{M} & \frac{-k}{M} & \frac{d}{M} & \frac{-d}{M} \end{bmatrix},\; B = \begin{bmatrix} 0 \\ 0 \\ \frac{-1}{m} \\ \frac{1}{M} \end{bmatrix},\; C = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \end{bmatrix} \; et \; D = \begin{bmatrix} 0 \\ 0 \\ \frac{h}{m} \\ 0 \end{bmatrix}$$

Lors de la conception d'une suspension active pour un véhicule roulant,

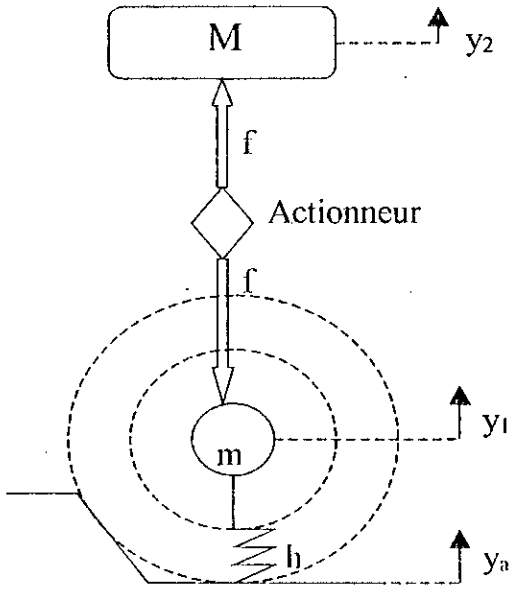


Figure 5.1: *Suspension active pure d'un quart de véhicule.*

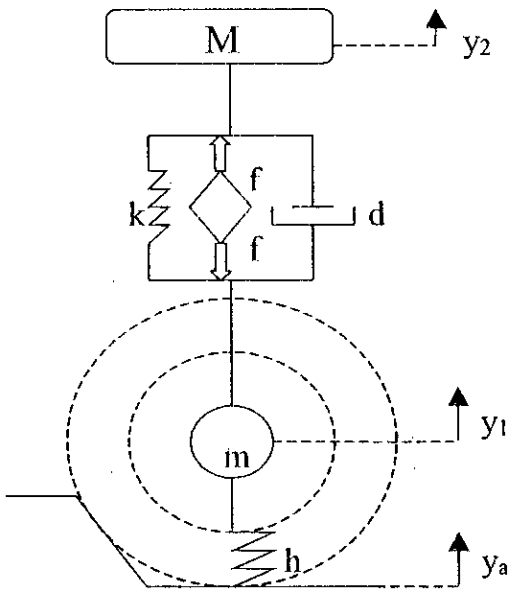


Figure 5.2: *Suspension active avec suspension conventionnelle.*

deux principaux critéres sont souvent considérés: le confort et la tenue de route(la sécurité de conduite)[Tho 79].

Une indication du confort peut être obtenue à partir de l'accélération verticale $\ddot{y}_2$de la masse M. Si M est maintenue constante alors l'accélération $\ddot{y}_2$est proportionnelle à la force f.

Pour la sécurité de conduite et la tenue de route, une indication peut être la mesure du décollement dynamique de la roue (y_1- y_a), ainsi que (y_1- y_2).

5.3 *PROBLEME DE LA MOITIE DU VEHICULE*

Le modèle de la moitié du vehicule étudié est donné par la figure 5.3[Lou 92].

La masse M correspond à la moitié de la masse totale du vehicule et J est le moment d'inertie du châssis par rapport au centre de gravité.

L'essieu et la roue avant sont représentés par une seule masse M_1, de même que l'essieu et la roue arrière sont représentés par une seule masse M_3. Les deux masses M_1 et M_3 sont reliées à la masse M à travers des actionneurs ayant une dynamique supposée idéale.

Les forces de commande f_1 et f_3, développées par les deux actionneurs, sont supposées être appliquées, avec la même intensité mais de sens opposé, au châssis et aux essieux.

Les pneus avant et arrière sont représentés par des ressorts linéaires de raideurs k_1 à l'avant et k_3 à l'arrière, et sont supposés assurer un contact ferme avec la route pour satisfaire l'exigence de sécurité.

Sous l'excitation des composantes verticales de la surface de la route, y_a à l'avant et y_b à l'arrière, on suppose que les masses M_1 et M_3 ne peuvent effectuer que des déplacements verticaux, respectivement y_1 et y_3, à partir des positions initiales d'équilibre.

Le système, supposé linéaire et invariant dans le temps[Lou 92], est décrit par:

$$\begin{aligned} \dot{x}(t) &= A.x(t) + B.f(t) + Dw(t) \\ y(t) &= C.x(t) \\ x(0^+) &= x_0 \\ x(0^-) &= 0 \end{aligned} \tag{5.2}$$

avec:
$x = [\, y_1\ y_2\ y_3\ y_4\ \dot{y}_1\ \dot{y}_2 \dot{y}_3\ \dot{y}_4]^T$: *vecteur d'état.*
$f = [f_1\ f_3]^T$: *vecteur de commande.*

$w = [y_a\ y_b]^T$: vecteur des perturbations représentant le profil de la route.

Dans le cas d'une suspension active pure(figure 5.3), les matrices A, B, C et D sont comme suit:

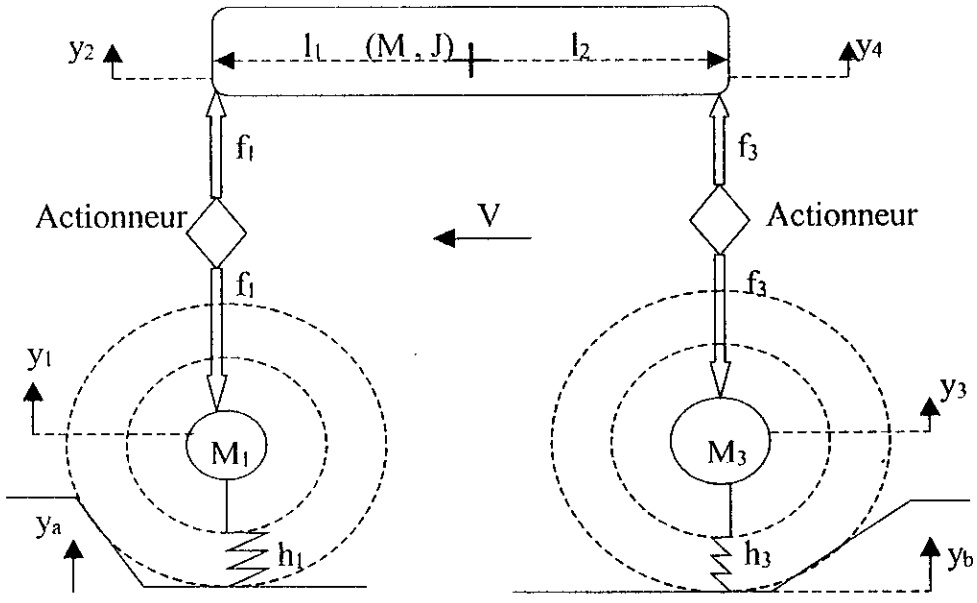


Figure 5.3: *Suspension active pure de la moitié d'un véhicule.*

$$A = \begin{bmatrix} 0 & 0\ 0 & 0\ 0\ 1\ 0\ 0\ 0 \\ 0 & 0\ 0 & 0\ 0\ 0\ 1\ 0\ 0 \\ 0 & 0\ 0 & 0\ 0\ 0\ 0\ 1\ 0 \\ 0 & 0\ 0 & 0\ 0\ 0\ 0\ 0\ 1 \\ \frac{-h_1}{M_1} & 0\ 0 & 0\ 0\ 0\ 0\ 0\ 0 \\ 0 & 0\ 0 & 0\ 0\ 0\ 0\ 0\ 0 \\ 0 & 0\ \frac{-h_3}{M_3} & 0\ 0\ 0\ 0\ 0\ 0 \\ 0 & 0\ 0 & 0\ 0\ 0\ 0\ 0\ 0 \end{bmatrix}, B = \begin{bmatrix} 0 & 0 \\ 0 & 0 \\ 0 & 0 \\ 0 & 0 \\ \frac{-1}{M_1} & 0 \\ (\frac{1}{M} + \frac{l_1^2}{J}) & (\frac{1}{M} - \frac{l_1.l_2}{J}) \\ 0 & \frac{-1}{M_3} \\ (\frac{1}{M} - \frac{l_1.l_2}{J}) & (\frac{1}{M} + \frac{l_2^2}{J}) \end{bmatrix}$$

$$C = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \end{bmatrix} \quad et \quad D = \begin{bmatrix} 0 & 0 \\ 0 & 0 \\ 0 & 0 \\ 0 & 0 \\ \frac{h_1}{M_1} & 0 \\ 0 & 0 \\ 0 & \frac{h_3}{M_3} \\ 0 & 0 \end{bmatrix}$$

Le cas où chaque suspension active est associée en parallèle à une suspension conventionnelle formée d'un ressort de raideur $k_1(k_3)$et d'un amortisseur $d_1(d_3)$est présenté à la figure 5.4.

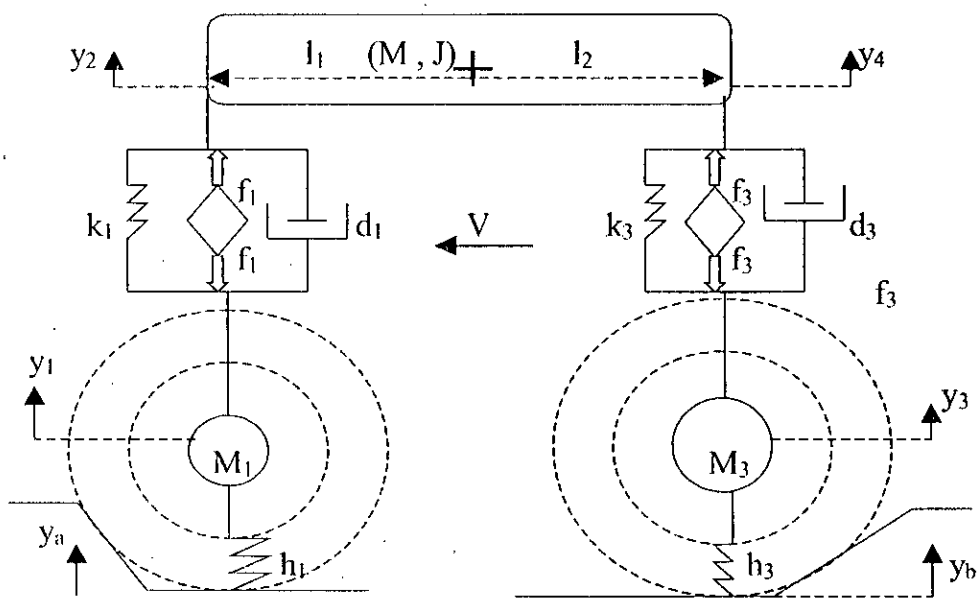


Figure 5.4: *Suspension active avec suspension conventionnelle.*

Comme pour le cas du quart de véhicule, une indication du confort peut être obtenue à partir des accélérations verticales $\ddot{y}_2$et $\ddot{y}_4$du châssis[Tho 79].

Pour la sécurité de conduite et la tenue de route, une indication peut être

la mesure des décollements dynamiques des roues (y_1- y_a) et (y_3- y_b), ainsi que (y_1- y_2) et (y_3- y_4).

5.4 *CONCLUSION*

Le modèle de la suspension active d'un véhicule considéré est supposé linéaire et invariant dans le temps.

Les actionneurs considérés sont supposés non inertiels, c'est à dire ayant des temps de réponse nuls.

Les performances du système ont été définies en tenant compte du confort, de la sécurité et de la tenue de route. L'indicateur du confort est considéré obtenu à partir de l'accélération verticale $\ddot{y}_2(\ddot{y}_4)$qui est proportionnelle à la force générée par l'actionneur.

Pour la sécurité de conduite et la tenue de route, une indication peut être la mesure des décollements dynamiques des roues (y_1- y_a) et (y_3- y_b), ainsi que (y_1- y_2) et (y_3- y_4).

Chapitre 6

APPLICATION A LA SUSPENSION ACTIVE

6.1 *INTRODUCTION*

Dans le présent chapitre, on s'intérèsse à l'application du contrôleur neuro-linguistique, déjà conçu dans le quatrième chapitre, et cela à la commande du système de suspension active d'un véhicule roulant défini dans le chapitre précédent(quart et moitié du véhicule).

Dans ce qui suit, on considère le modéle de la suspension active linéaire et invariant dans le temps. Les actionneurs utilisés sont supposés non inertiels. Le profil de la route est considéré connu d'une façon précise.

6.2 *QUART DU VEHICULE*

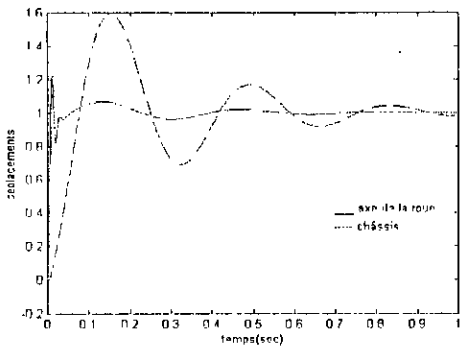
Dans le cas o *la suspension active est associée en parallèle à une suspension conventionnelle formée d'un ressort de raideur k et d'un amortisseur d(figure 5.2)(premier modèle de[Lou 96]), deux types de contrôleurs neuro-linguistiques sont utilisés(CNL 1 et CNL 2).*

6.2.1 *Cas du CNL 1*

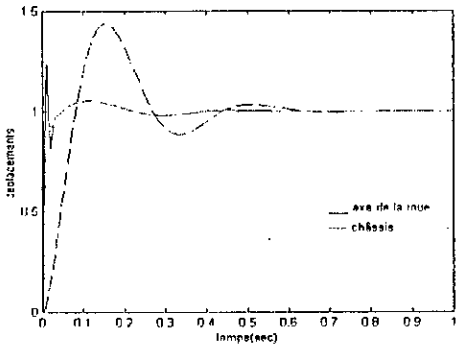
Le contrôleur neuro-linguistique est constitué de quatre réseaux de neurones traitant chacun un module du contrôleur flou: la fuzzification de la variable d'entrée e(figure 4.3), la fuzzification de la variable d'entrée de(figure 4.4), l'inférence(figure 4.6) et la défuzzification(figure 4.5).

6.2.1.1 Apprentissage du *contrôleur neuro-linguistique*

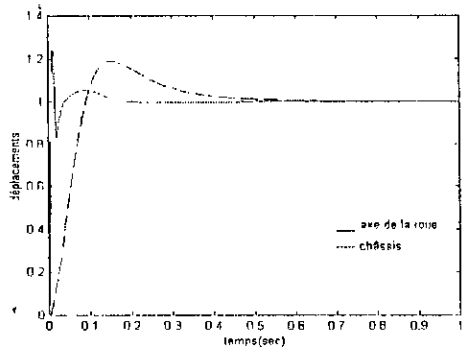
L'apprentissage du contrôleur neuro-linguistique, pour le calcul des 867 paramètres du réseau de neurones(réseau représentant la matrice d'inférence), est effectué jusqu'à 5×10^6 itérations(figures 6.1)où le contrôleur le plus performant est obtenu(figure 6.1.d). L'amplitude de la force de commande f développée par l'actionneur est, alors, représentée sur la figure 6.1.e.



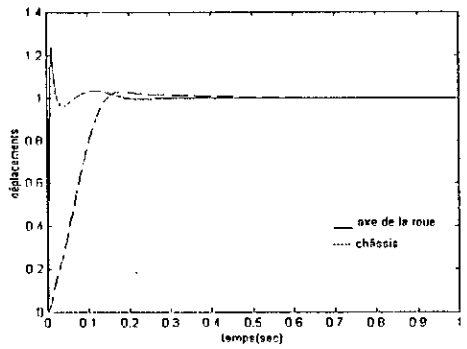
a: *Apprentissage jusqu'à* 10^3 itérations.



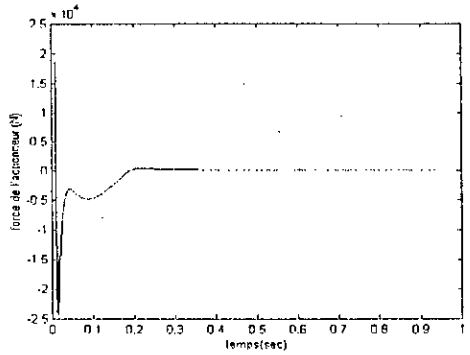
b: *Apprentissage jusqu'à* 10^4 itérations.



c: *Apprentissage jusqu'à* 10^5 itérations.



d: *Apprentissage jusqu'à* 5×10^6 itérations.

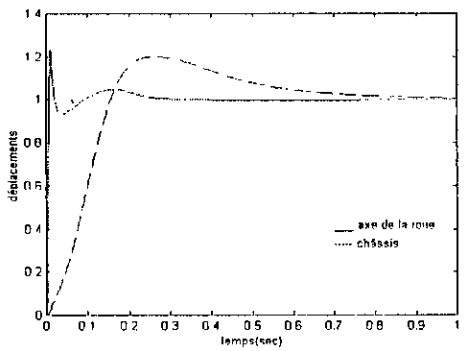


e: *Force de l'actionneur*(5×10^6 itérations).

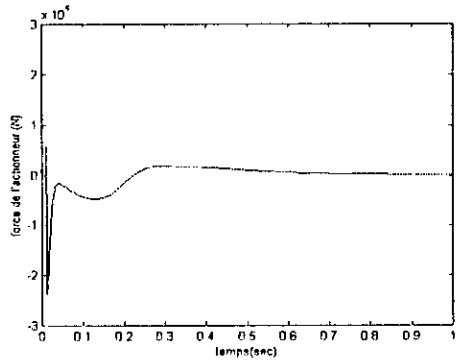
Figures 6.1: Apprentissage du CNL 1.

6.2.1.2 Test de variations paramétriques

En introduisant une variation paramétrique des différents paramètres suivants du système(M, m, k, d, et h), on constate la robustesse du contrôleur(figures 6.2, 6.3, 6.4, 6.5 et 6.6).

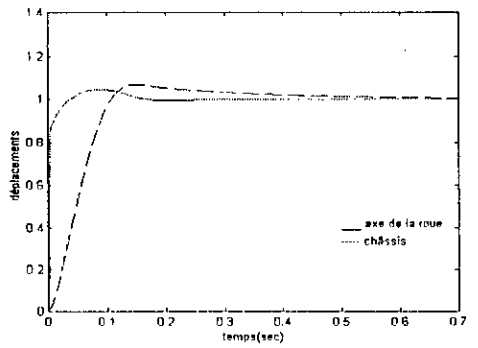


a: *Déplacements.*

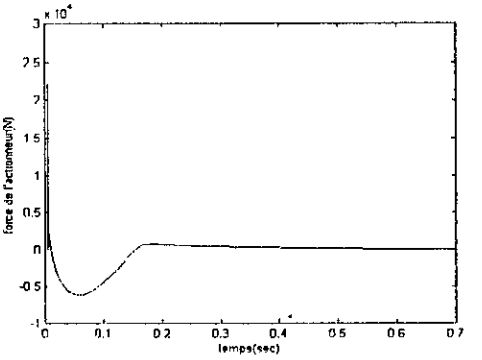


b: *Commande.*

Figures 6.2: Variation de 100% de M

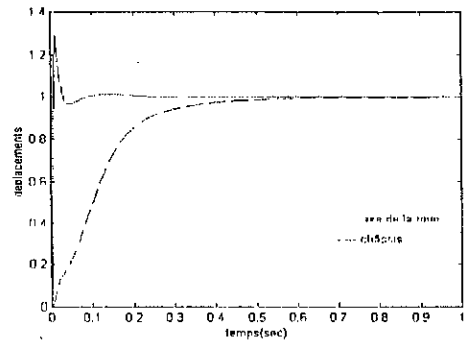


a: *Déplacements.*

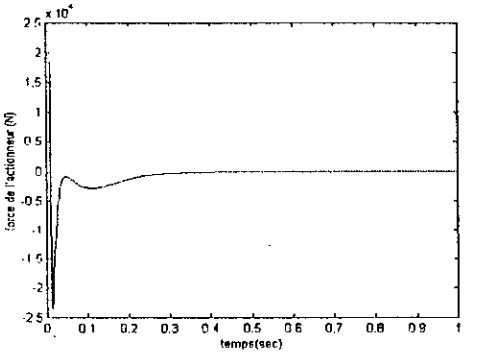


b: *Commande.*

Figures 6.3: Variation de 90% de m

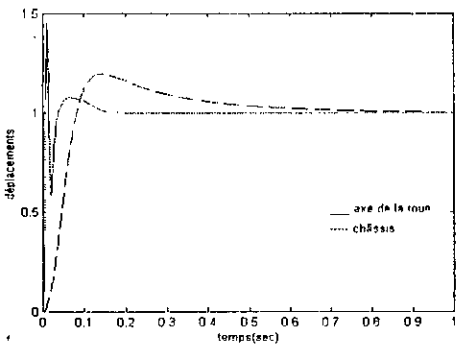


a: *Déplacements.*

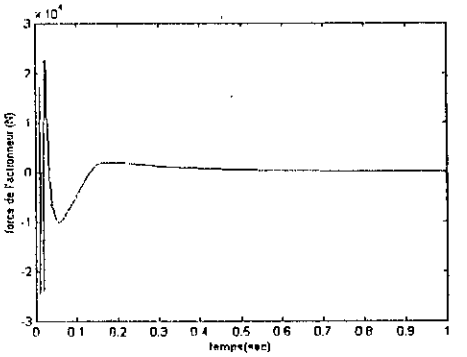


b: *Commande.*

Figures 6.4: Variation de 50% de k

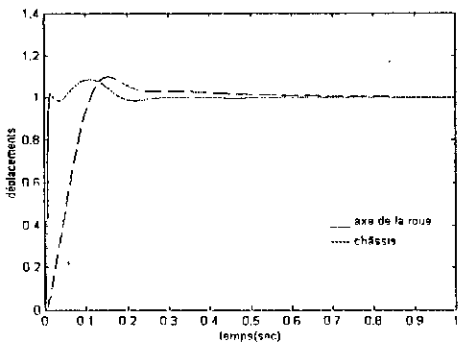


a: *Déplacements.*

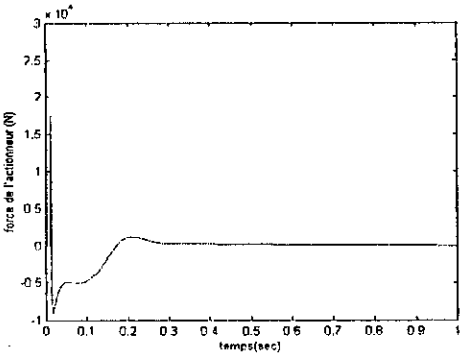


b: *Commande.*

Figures 6.5: *Variation de 50% de d*



a: *Déplacements.*



b: *Commande.*

Figures 6.6: *Variation de 50% de h*

6.2.1.3 Cas d'existence de bruits de mesures

Pour considérer un cas plus pratique, on suppose que la mesure des différentes grandeurs est affectée par un bruit de distribution normale. On remarque, alors, la robustesse du contrôleur(figure 6.7).

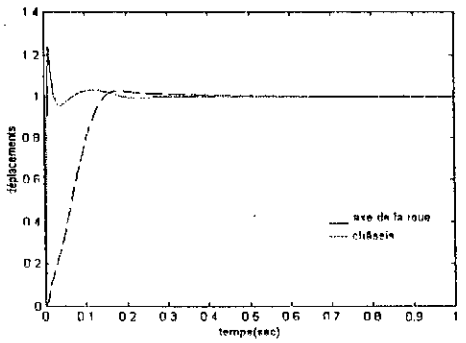
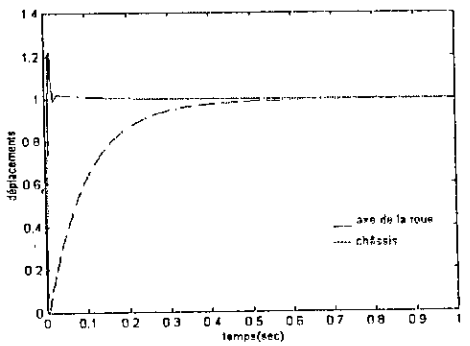
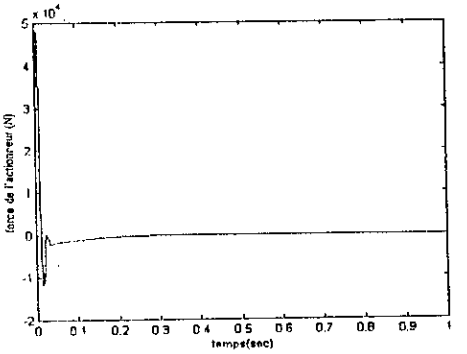


Figure 6.7: *Cas d'existence de bruits de mesures.*

6.2.1.4 Cas d'une suspension active pure

Dans le cas de l'utilisation d'une suspension active pure, on remarque moins d'oscillations, un dépassement moins important, surtout pour le châssis du véhicule, ainsi qu'un travail développé par l'actionneur moins important(figures 6.8).



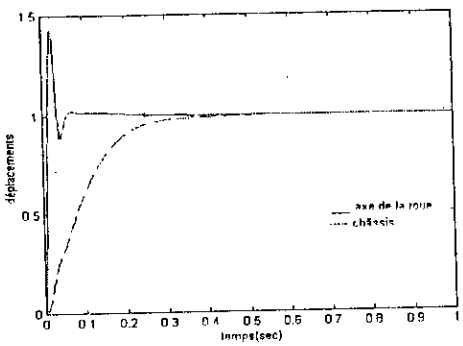


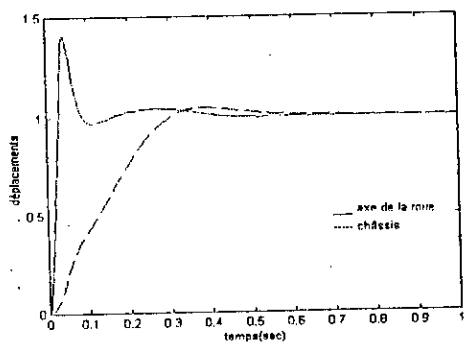
a: Déplacements. *b: Commande.*

Figures 6.8: Application du CNL 1 à une suspension active pure.

6.2.1.5 Application du contrôleur à différents modèles

Pour montrer la possibilité de la commande d'un système de suspension active sans connaissance à priori du modèle, on procède au test qui consiste à changer le modèle, utilisé dans la synthèse du contrôleur, par deux autres modèles: deuxième modèle de [Lou 96] et modèle de [Tho 76]. Les résultats obtenus (figures 6.9) montrent que le contrôleur réagit parfaitement.





a: deuxième modèle de [Lou 96]. *b: modèle de [Tho 76].*

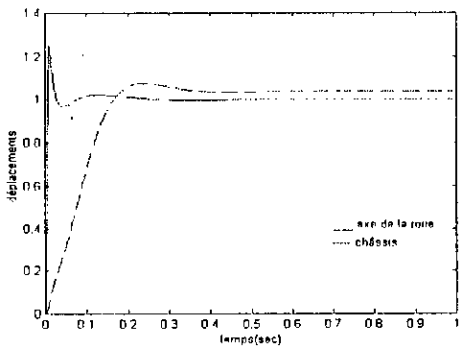
Figures 6.9: Application du CNL 1 à différents modèles.

6.2.2 Cas du CNL 2

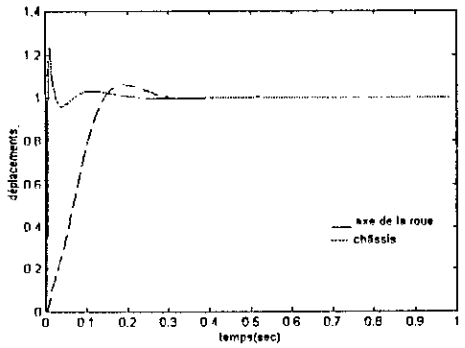
Le contrôleur neuro-linguistique est constitué d'un seul réseau de neurones.

6.2.2.1 Apprentissage du *contrôleur neuro-linguistique*

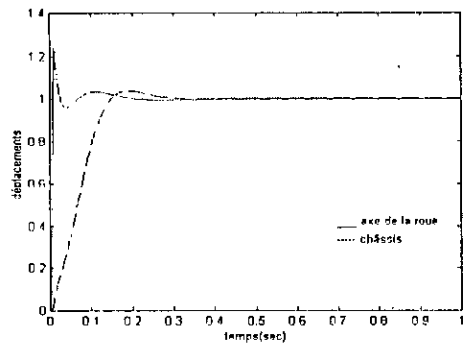
L'apprentissage du contrôleur neuro-linguistique, pour le calcul de 43 paramètres du réseau de neurones, est effectué jusqu'à 337 itérations(figures 6.10)où le contrôleur le plus performant est obtenu(figure 6.10.d). L'amplitude de la force de commande f développée par l'actionneur est, alors, représentée sur la figure 6.10.e.



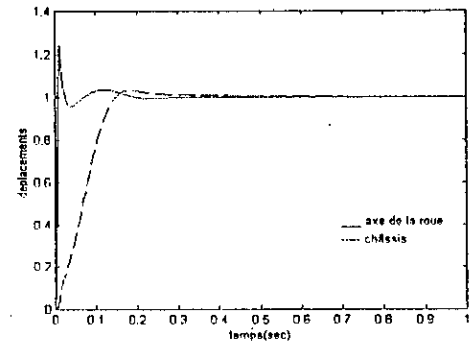
a: Apprentissage jusqu'à 10 itérations.



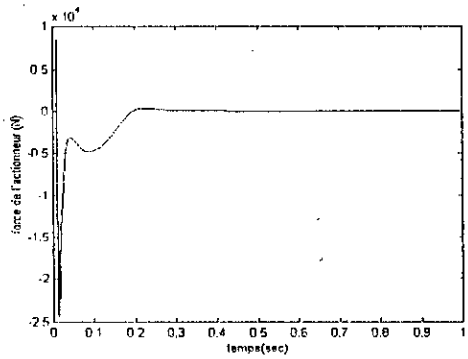
b: Apprentissage jusqu'à 50 itérations.



c: Apprentissage jusqu'à 200 itérations.



d: Apprentissage jusqu'à 337 itérations.

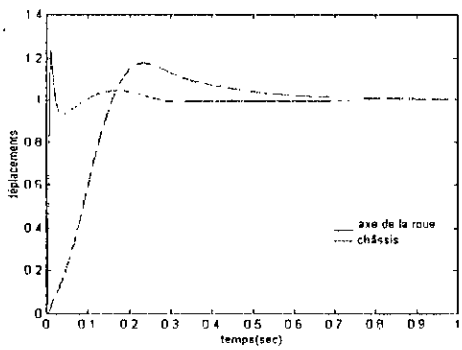


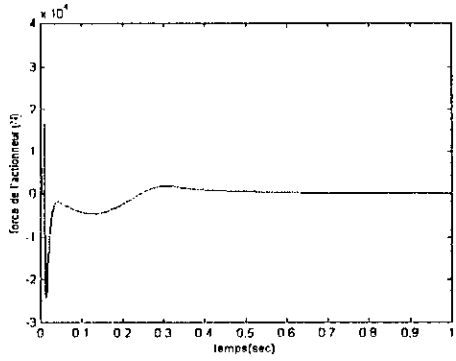
e: Force de l'actionneur(337 itérations).

Figures 6.10: Apprentissage du CNL 2.

6.2.2.2 Test de variations paramétriques

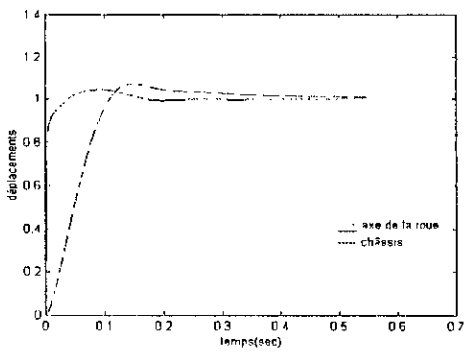
En introduisant une variation paramétrique des différents paramètres suivants du système(M, m, k, d, et h), on constate la robustesse du contrôleur(figures 6.11, 6.12, 6.13, 6.14 et 6.15).

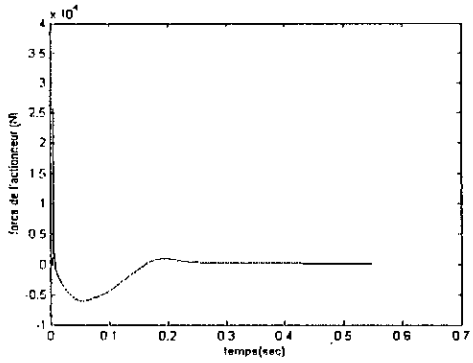




a: Déplacements. *b: Commande.*

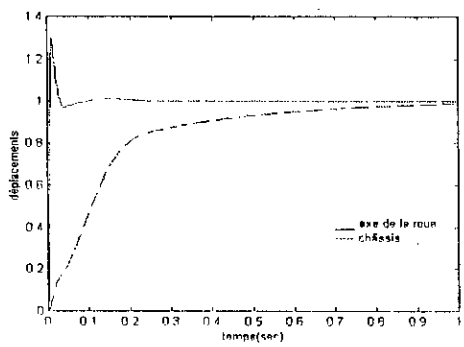
Figures 6.11: Variation de 100% de M

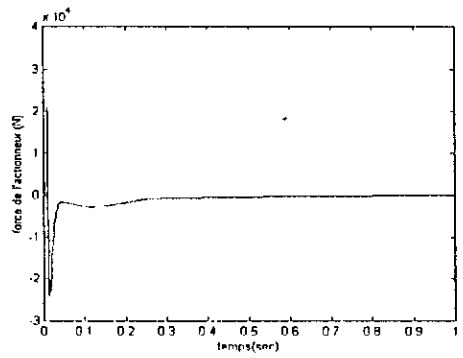




a: Déplacements. *b: Commande.*

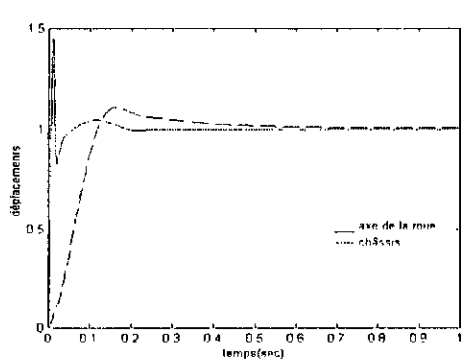
Figures 6.12: Variation de 90% de m

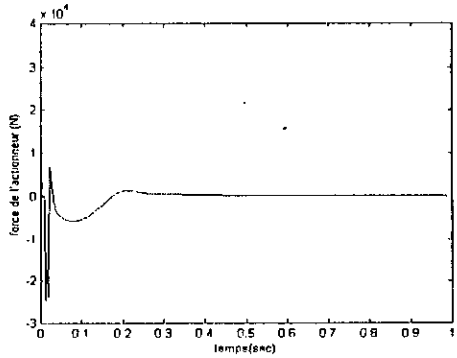




a: Déplacements. *b: Commande.*

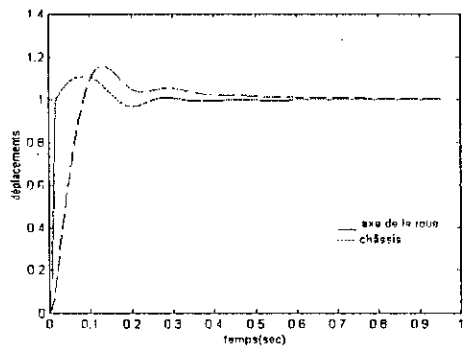
Figures 6.13: Variation de 50% de k

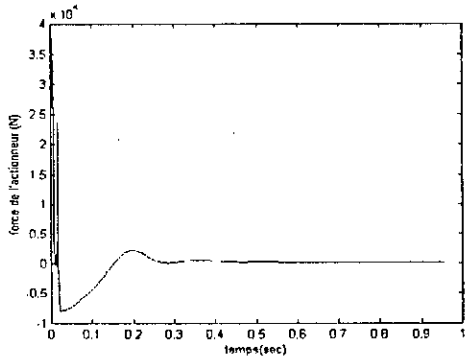




a: Déplacements. *b: Commande.*

Figures 6.14: Variation de 50% de d





a: Déplacements. *b: Commande.*

Figures 6.15: Variation de 50% de h

6.2.2.3 Cas d'existence de bruits de mesures

On suppose que la mesure des différentes grandeurs est affectée par un bruit de distribution normale. On remarque, alors, la robustesse du contrôleur (figure 6.16).

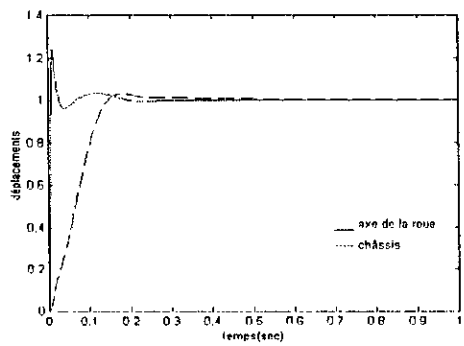
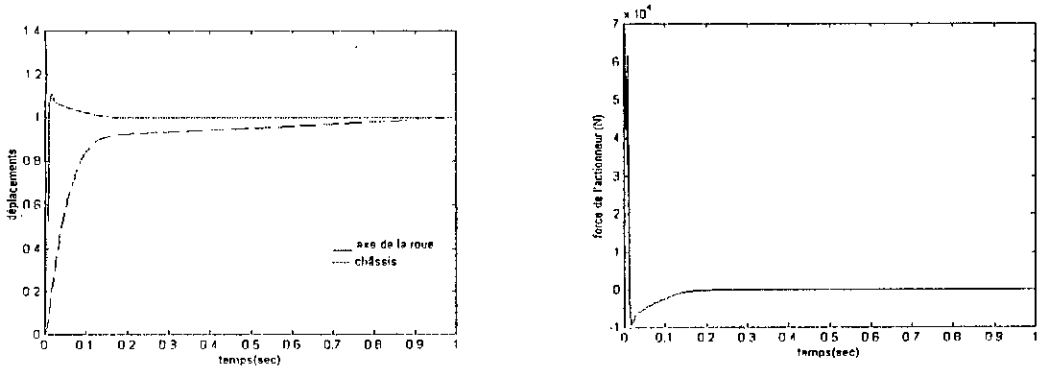


Figure 6.16: Cas d'existence de bruits de mesures.

6.2.2.4 Cas d'une suspension active pure

Dans le cas de l'utilisation d'une suspension active pure, on remarque moins d'oscillations, un dépassement moins important, surtout pour le châssis du véhicule, ainsi qu'une force développée par l'actionneur moins importante (figures 6.17).

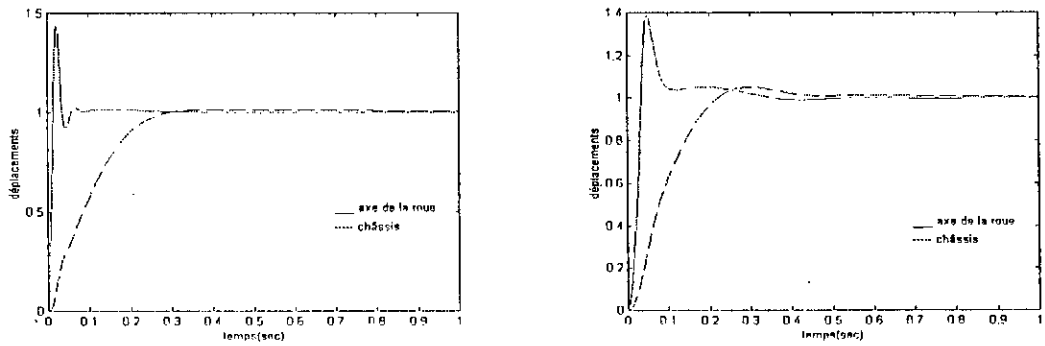


a: Déplacements. *b: Commande.*

Figures 6.17: Application du CNL 2 à la suspension active pure.

6.2.2.5 Application du contrôleur à différents modèles

Pour montrer la possibilité de la commande d'un système de suspension active sans connaissance à priori du modèle, on procède au test qui consiste à changer le modèle, utilisé dans la synthèse du contrôleur, par deux autres modèles: deuxième modèle de [Lou 96] et modèle de [Tho 76]. Les résultats obtenus (figures 6.18) montrent que le contrôleur réagit parfaitement.



a: deuxième modèle de [Lou 96]. *b: modèle de [Tho 76].*

Figures 6.18: Application du CNL 2 à différents modèles.

6.3 *MOITIE DU VEHICULE*

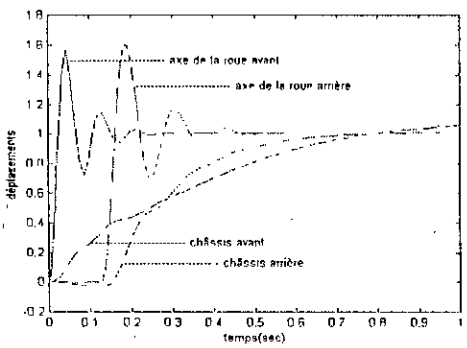
Dans le cas de deux suspensions actives pures(avant et arrière)(figure 5.3), deux types de contrôleurs neuro-linguistiques sont utilisés pour chaque suspension.

6.3.1 *Cas du CNL 1*

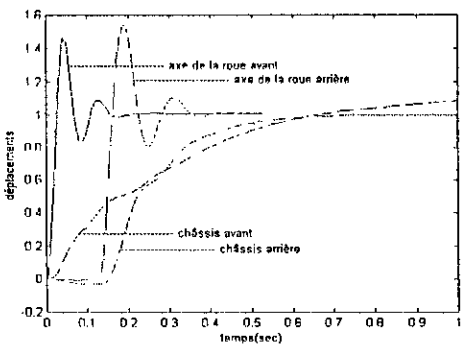
Pour chaque suspension, le contrôleur neuro-linguistique(étant le même) est constitué de quatre réseaux de neurones traitant chacun un module du contrôleur flou: la fuzzification de la variable d'entrée e(figure 4.3), la fuzzification de la variable d'entrée de(figure 4.4), l'inférence(figure 4.6) et la défuzzification(figure 4.5).

6.3.1.1 Apprentissage du *contrôleur neuro-linguistique*

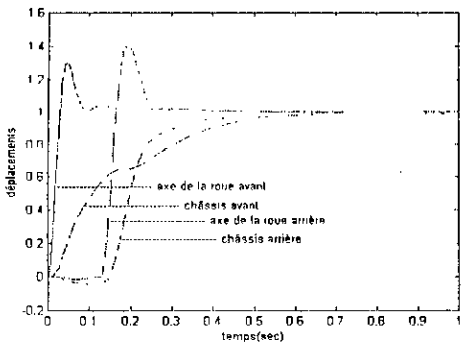
L'apprentissage du contrôleur neuro-linguistique, pour le calcul de 867 paramètres du réseau de neurones(réseau représentant la matrice d'inférence), est effectué jusqu'à 5×10^6 itérations(figures 6.19)où le contrôleur le plus performant est obtenu(figure 6.19.d). Les amplitudes des forces de commande f_1 et f_2 développées par les deux actionneurs sont, alors, représentées sur la figure 6.19.e.



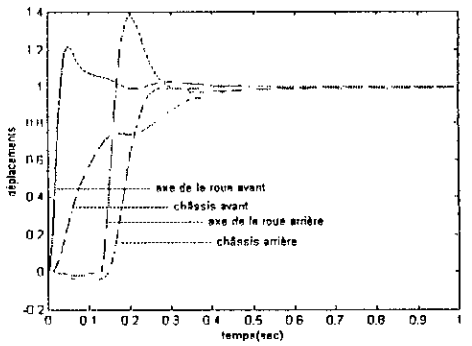
a: Apprentissage jusqu'à 10^3 itérations.



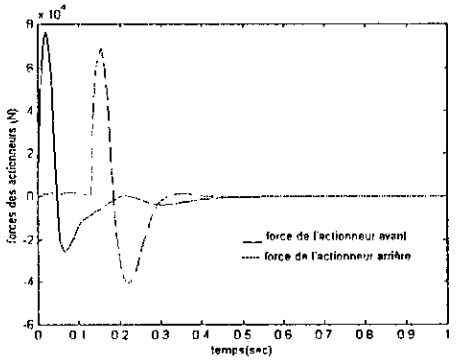
b: Apprentissage jusqu'à 10^4 itérations.



c: Apprentissage jusqu'à 10^5 itérations.



d: Apprentissage jusqu'à 5×10^6 itérations.

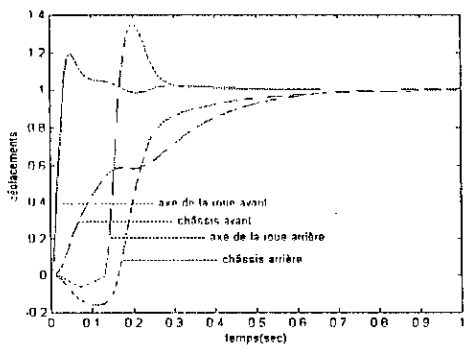


e: Forces des actionneurs(5×10^6 itérations).

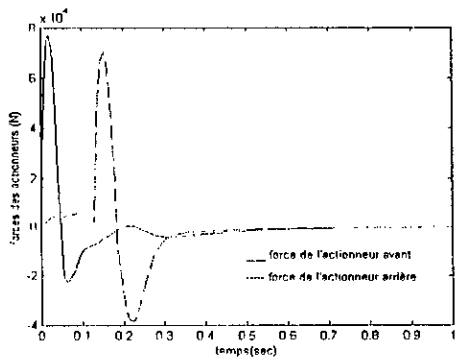
Figures 6.19: Apprentissage du CNL 1

6.3.1.2 Test de variations paramétriques

En introduisant une variation paramétrique des différents paramètres suivants du système(M, h_1, h_3 *et J), on constate la robustesse du contrôleur (figures 6.20, 6.21, 6.22, et 6.23).*

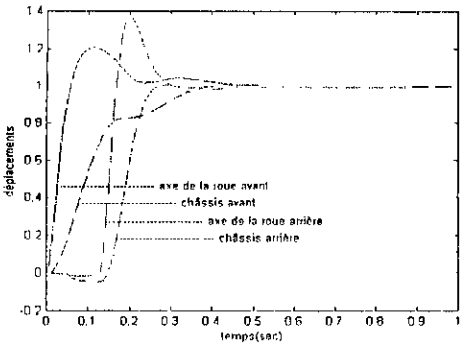


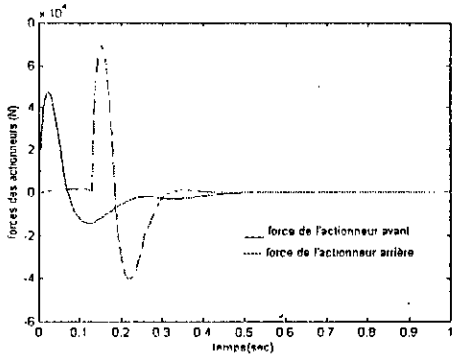
a: Déplacements.



b: Commandes.

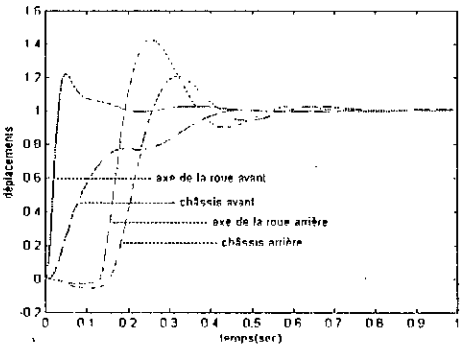
Figures 6.20: Variation de 100% de M

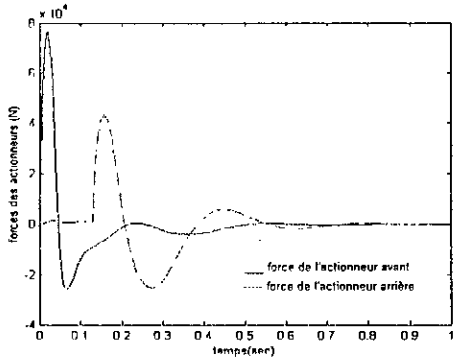




a: Déplacements. *b: Commandes.*

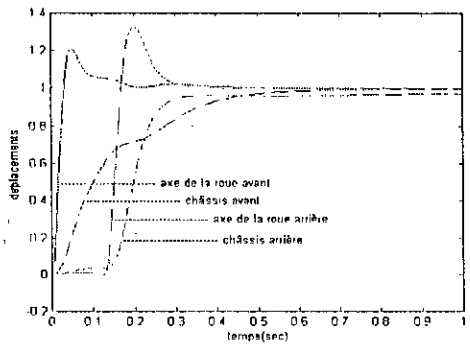
Figures 6.21: Variation de 50% de h_1

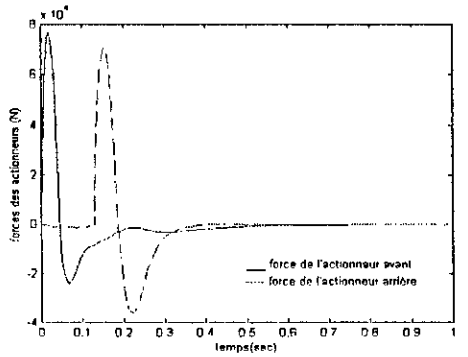




a: Déplacements. *b: Commandes.*

Figures 6.22: Variation de 50% de h_3





a: Déplacements. *b: Commandes.*

Figures 6.23: Variation de 50% de J

6.3.1.3 Cas d'existence de bruits de mesures

Pour considérer un cas plus pratique, on suppose que la mesure des différentes grandeurs est affectée par un bruit de distribution normale. On remarque,

alors, la robustesse du contrôleur (figure 6.24).

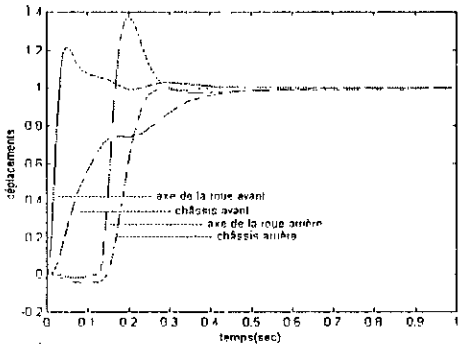


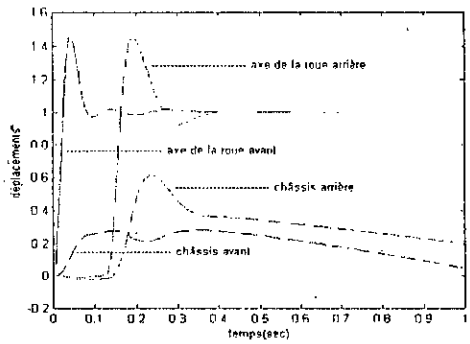
Figure 6.24: Cas d'existence de bruits de mesures.

6.3.2 *Cas du CNL 2*

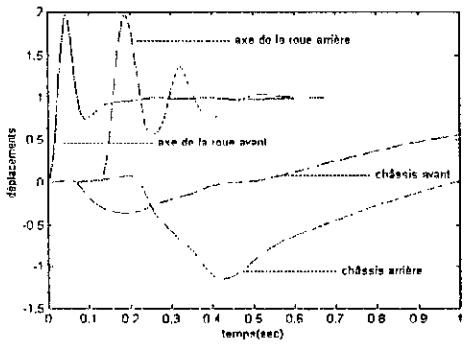
Pour chaque suspension, le contrôleur neuro-linguistique(étant le même)est constitué d'un seul réseau de neurones.

6.3.2.1 Apprentissage du *contrôleur neuro-linguistique*

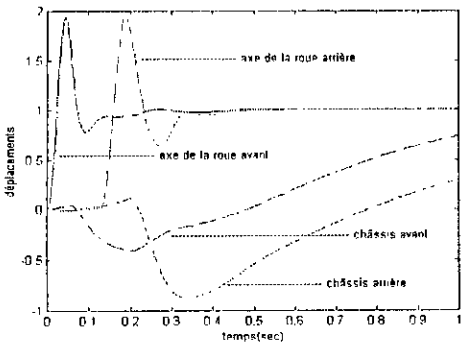
L'apprentissage du contrôleur neuro-linguistique, pour le calcul de 43 paramètres du réseau de neurones, est effectué jusqu'à 337 itérations(figures 6.25)où le contrôleur le plus performant est obtenu(figure 6.25.d). Les amplitudes des forces de commande f_1 et f_3 développées par les deux actionneurs sont, alors, représentées sur la figure 6.25.e.



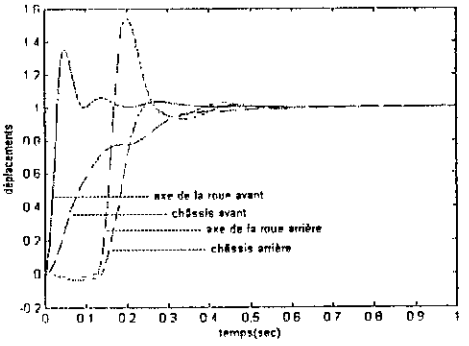
a: Apprentissage jusqu'à 10 itérations.



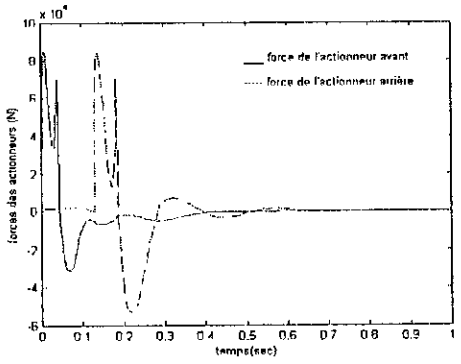
b: Apprentissage jusqu'à 200 itérations.



c: *Apprentissage jusqu'à 250 itérations.*



d: *Apprentissage jusqu'à 337 itérations.*

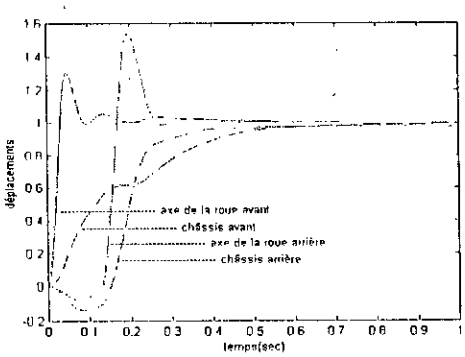


e: *Forces des actionneurs(337 itérations).*

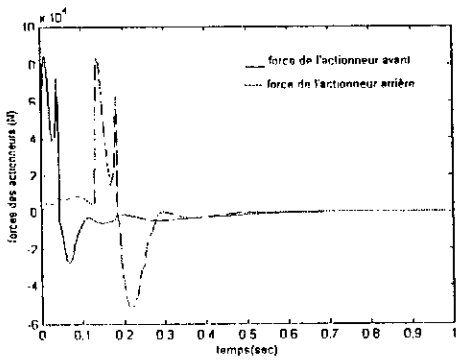
Figures 6.25: Apprentissage du CNL 2.

6.3.2.2 Test de variations paramétriques

En introduisant une variation paramétrique des différents paramètres suivants du système(M, h_1, h_3 et J), on constate la robustesse du contrôleur (figures 6.26, 6.27, 6.28, et 6.29).

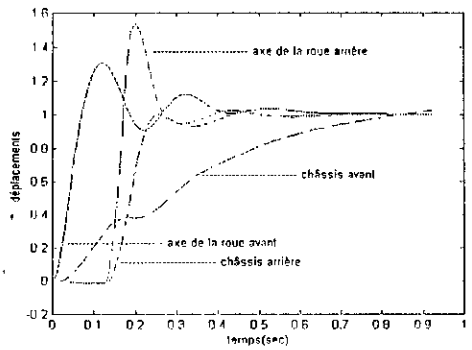


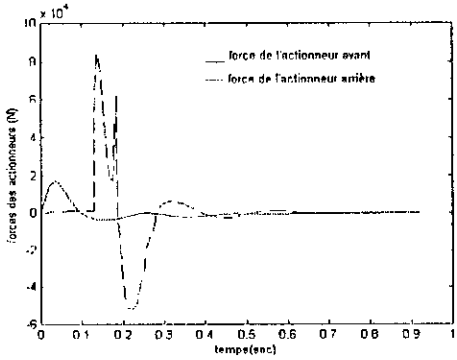
a: *Déplacements.*



b: *Commandes.*

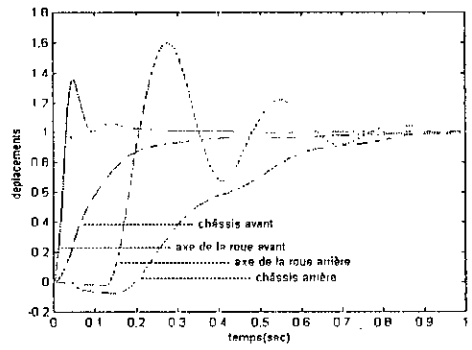
Figures 6.26: Variation de 100% de M

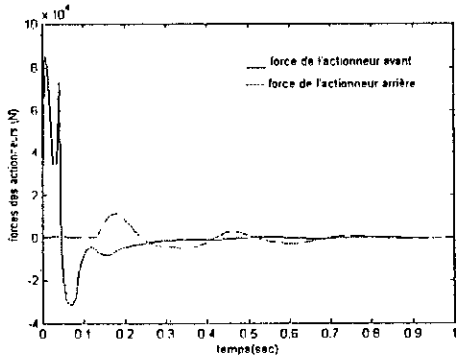




a: Déplacements. *b: Commandes.*

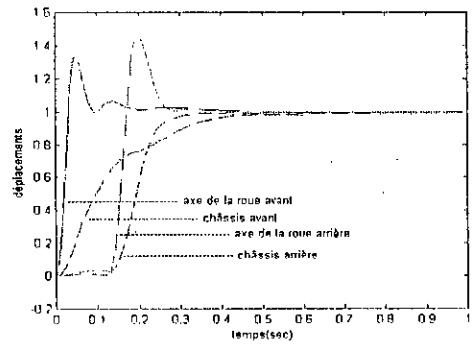
Figures 6.27: Variation de 20% de h_1

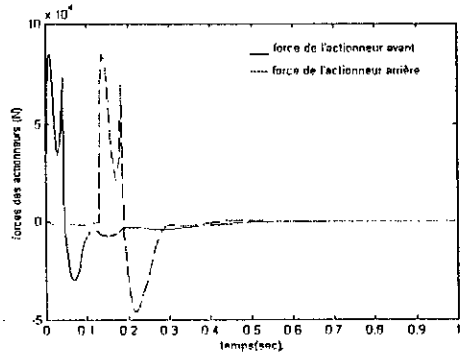




a: Déplacements. *b: Commandes.*

Figures 6.28: Variation de 20% de h_3





a: Déplacements. *b: Commandes.*

Figures 6.29: Variation de 50% de J

Cas d'existence de bruits de mesures

Pour considérer un cas plus pratique, on suppose que la mesure des différentes grandeurs est affectée par un bruit de distribution normale. On remarque,

alors, la robustesse du contrôleur (figure 6.30).

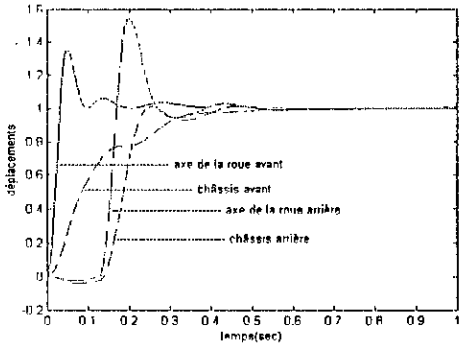


Figure 6.30: Cas d'existence de bruits de mesures.

6.4 *CONCLUSION*

A travers les résultats de simulations présentés dans ce chapitre, on constate que les deux types de contrôleurs neuro-linguistiques, CNL 1 et CNL 2, assurent un bon rejet de perturbation(variation du profil de la route)dans le cas du modèle quart du véhicule ainsi que dans celui de la moitié du véhicule.

Des tests de variations paramétriques montrent la robustesse des deux types de contrôleurs neuro-linguistiques dans des situations pratiques correspondants à une variation de la charge ou de l'inertie du véhicule, une crevaison d'une roue, l'usure de la suspension conventionnelle ou l'usure des pneus.

Les deux types de contrôleurs neuro-linguistiques suppriment les éventuels bruits qui peuvent affecter la mesure des différentes grandeurs.

L'association de la suspension active avec une suspension conventionnelle, ce qui a pour but d'augmenter la fiabilité du système, crée un peu plus d'oscillations et un dépassement un peu plus important surtout au niveau du châssis du véhicule, ainsi que l'augmentation de la force développée par l'actionneur.

Les deux types de contrôleurs neuro-linguistiques(CNL 1 et CNL 2)présentent, pratiquement, les mêmes performances. Du point de vue conception, CNL 1 présente plus de difficultés à cause de sa structure(quatre réseaux de neurones), et la difficulté majeur réside en la conception du réseau représentant la matrice d'inférence. Mais à cause de cette même structure, CNL 1 présente un meilleur avantage de fléxibilité. Le contrôleur neuro-linguistique CNL 2 présente le plus faible temps de calcul.

Les deux types de contrôleurs neuro-linguistiques(CNL 1 et CNL 2)présentent une solution aux problèmes suivants de la commande floue:

- La difficulté de la modélisation des connaissances d'un expert.

- *Les difficultés de synthèse d'un contrôleur flou et la détérioration de ses performances lors d'une augmentation de la base des règles.*
- *Les difficultés de l'implémentation pratique des contrôleurs flous.*

Chapitre 7

CONCLUSION GENERALE

Cette thèse a pour objectif de contribuer à la conception d'un système de suspension active pour véhicule roulant, et cela en lui appliquant un contrôleur neuro-linguistique.

La complexité du modèle réel de la suspension active, ainsi que le fait que le profil de la route ne peut être connu d'une façon précise, mettent en évidence l'intérêt d'une nouvelle approche basée sur l'association de la commande linguistique floue et les réseaux de neurones.

Sur le plan théorique, notre travail a consisté d'abord à rassembler les outils nécéssaires à la réalisation de la partie linguistique du contrôleur, ainsi que ceux utiles pour la partie neuronale, et de formuler le problème de la suspension active d'un véhicule. La recherche de l'architecture possédant la meilleure capacité d'apprentissage a nécéssité de nombreux essais.

Les deux types de contrôleurs neuro-linguistiques obtenus (la représentation par quatre réseaux de neurones CNL 1 et la représentation par un seul réseaux de neurones CNL 2) ont été appliqués à un modèle de suspension active linéaire et invariant dans le temps. Les actionneurs utilisés sont supposés de dynamiques négligeables. Le profil de la route est considéré connu d'une façon précise.

La conception des contrôleurs neuro-linguistiques(surtout CNL 1)a nécéssité un temps de calcul assez considérable pour le choix de l'architecture du réseau de neurones, ainsi que pour l'apprentissage.

Les essais de simulation ont montré que:

-Les deux types de contrôleurs neuro-linguistiques(CNL 1 et CNL 2)présentent, pratiquement, les mêmes performances.

-Du point de vue conception, le contrôleur neuro-linguistique CNL 1 présente plus de difficultés à cause de sa structure.

-Le contrôleur neuro-linguistique CNL 1 présente l'avantage d'une meilleure flexibilité.

-Le contrôleur neuro-linguistique CNL 2 présente le plus faible temps de calcul.

-Les deux types de contrôleurs peuvent commander un système de suspension active sans connaissance à priori du modèle.

-Les deux types de contrôleurs suppriment les éventuels bruits qui peuvent affecter la mesure des différentes grandeurs.

-Les deux types de contrôleurs présentent une robustesse vis à vis des variations paramétriques du systéme qui correspondent à des situations pratiques telles qu'une variation de la charge ou de l'inertie du véhicule, une crevaison d'une roue, l'usure de la suspension conventionnelle ou l'usure des pneus.

Le travail réalisé dans cette thèse n'est qu'une première application de la commande neuro-linguistique à la suspension active. Beaucoup de progrés restent encore à faire. Entre autres, on peut envisager:

-L'étude de l'influence d'autres variantes du contrôleur neuro-linguistique sur ses performances: autres conceptions de la partie linguistique (variables linguistiques, fonctions d'appartenance, nombre de classes, leur distribution, etc...), et autres structures et méthodes d'apprentissage de la partie neuronale.

-L'étude de l'application de la commande neuro-linguistique sur un modèle non-linéaire de suspension active.

-L'association avec cette commande d'une pré-information du profil de la route.

-L'étude de la commande optimale dans le cas du contrôleur neuro-linguistique.

-L'étude de l'influence de la dynamique des actionneurs sur les performances de la commande.

-L'implémentation du contrôleur neuro-linguistique pour une éventuelle utilisation pratique.

Bibliographie

[*Ass 74*] *S.Assilian, E.H.Mamdani: "Learning control algorithms in real dynamic systems". Proc.4th.Int.IFAC / IFIP.Conf.on Digital Computer applications to process control, Zurich, 1974.*

[*Büh 94*] *H.Bühler: "Réglage par logique floue", Presses Polytechniques et Universitaires Romandes, 1994.*

[*Cas 95*] *Castro J.L.: "Fuzzy logic controllers are universal approximators", IEEE Trans.on SMC, vol.25 N°4, Avril 1995.*

[*Dot 90*] *Dote Y.: "Fuzzy and neural network controller", IEEE Conf., pp.1314-1343, 1990.*

[*Fre 92*] *A.Freeman et M.Skapura: "Neural Networks", Addision-Wesley publishing company, 1992.*

[*Hac 85*] *Hac .A.: "Suspension optimization of a 2-D of vehicle using a stochastic optimal control technique", Journal of Sound and Vibration, 100, pp.343-357, 1985.*

[*Hol 82*] *L.P.Holmblad, J.J.Ostergaard: "control of a cement Kiln by fuzzy logic". Fuzzy information and decision processes, North Holland, pp.389-399, 1982.*

[*Hor 90*] *Hrikawa S., Furuhashi T., Okuma S., Uchikawa Y.: "Composition methods of Fuzzy Neural Networks", IEEE Conf., pp.1253-1258, 1990.*

[*Hro 91*] *Hrovat.D.: "Optimal suspension performance for 2-D vehicle models", Journal of Sound and Vibration, 146, pp.93-110, 1991.*

[*Kos 92*] *Kosko.B: "Neural Networks and Fuzzy Systems: a dynamical systems approach to machine intelligence", Prentice Hall, 1992.*

[*Lee 90*] *C.C.Lee: "Fuzzy logic in control systems: fuzzy logic controller - Part 1 and Part 2", IEEE Transactions on Systems, Man and Cybernetics, vol.20, N°2, pp.404-435, Mars/Avril 1990.*

[*Lem 76*] *H.R.Van Nauter Lemke, W.J.M.Kickert: " The application of fuzzy set theory to control a warm water process ". Automatica 12, 4, pp.301-308, 1976.*

[*Lin 93*] *Y.J.Lin, Y.Q.Lu and J.Padovan: "Fuzzy logic control of vehicle suspension systems", Int. J.Vehicle Design, 14 , pp.457-470, 1993.*

[*Lou 88*] *Louam.N, D.A.Wilson and R.S.Sharp: "Optimal control of a vehicle suspension incorporating the time delay between front and rear wheel inputs", Vehicle System Dynamics, Vol.17(6), pp.317-336, 1988.*

[*Lou 92*] *Louam.N, D.A.Wilson and R.S.Sharp: "Optimization and Performance Enhancement of Active Suspensions for Automobiles under Preview of the Road", Vehicle System Dynamics, Vol.21, pp.39-63, 1992.*

[*Lou 96*] *Louam.N: "Conception et optimisation d'un système de suspension active pour un chariot de transport de malade", First Magrebin Congress of Mechanics CMM'96, Ghardaïa, Algeria, pp.307-312, 23-26 Mars 1996.*

[*Mam 75*] *E.H.Mamdani, S.Assilian: "An experiment in linguistic synthesis with a fuzzy logic controller". Int.J.Man-Machine studies 7, pp.1-13, 1975.*

[*Ost 77*] *J.J.Ostergaard: "Fuzzy logic control of a heart exchanger process". Fuzzy Automata and decision processes, North Holland, 1977.*

[*Pro 79*] *J.J.Procyk et E.H.Mamdani: "A linguistic self-organizing process controller", Automatica, vol.15, pp.15-30, 1978.*

[*Rao 97*] *M.V.C.Rao et V.Prahlad: "A tunable fuzzy logic controller for vehicle-active suspension systems", Fuzzy Sets and Systems 85, pp.11-21, 1997.*

[Ray 92] *Ray.L.R.: "Robust linear-optimal control laws for active suspension systems", ASME Journal of Dynamic Systems, Measurement and Control, 114, pp.592-598, 1992.*

[*Ren 95*] *J.M.Renders: "Algorithmes génétiques et réseaux de neurones", Editions HERMES, 1995.*

[*Tho 76*] *Thompson. A.G.: "An active suspension system with optimal linear state feedback", Vehicle System Dynamics, 5 , pp.187-203, 1976.*

[*Tho 79*] *Thompson A.G. and Pearce C.E.M.: "An Optimal Suspension for an Automobile on a Random Road", Society of Automotive Engineers Inc., paper N° 790478, 1979.*

[*Tom 76*] *Tomizuka. M.: "Optimum linear preview control with application to vehicle suspension-revisited", ASME Journal of Dynamic Systems, Measurement and Control, 98, pp.309-315, 1976.*

[*Wan 92*] *Wang L.X.: "Fuzzy systems are universal approximators", Proc. First IEEE Conf.on Fuzzy Systems, pp.1163-1169, San Diego, 1992.*

[*Was 89*] *P.D.Wasserman: "Neural computing: theory and practice", Edition Van Nostrand Reinhold, New York, 1989.*

[*Wil 77*] *D.Willaeys, P.Mangin, N.Malvache: "Use of fuzzy sets for system modelizing and control: application to the speed control of a strongly perturbed motor". Proc.5th IFAC / IFIP Int.Conf.on Digital Computer applications to process control, The Hague, 1977.*

[*Yeh 94*] *Yeh.E.C. and Tsao.Y.J.: "A fuzzy preview control scheme of active suspension for rough road", Int. J.Vehicle Design, 15, pp.166-180, 1994.*

[*Yos 90*] *Yoshimura.T. and Sugimoto.M.: "An active suspension for a vehicle travelling on flexible beams with an irregular surface", Journal of Sound and Vibration, 138, pp.433-445, 1990.*

[*Yos 91*] *Yoshimura.T. and Ananthanarayana.N.: "Stochastic optimal control of vehicle suspension with preview on an irregular surface", International Journal of Systems Science, 22, pp.1599-1611, 1991.*

[*Yos 96.a*] *Yoshimura.T.: "Active suspension of vehicle systems using fuzzy logic", International Journal of Systems Science, 27, pp.215-219, 1996.*

[*Yos 96.b*] *T.Yoshimura and N.Hayashi: "Active control for the suspension of a large-sized buses using fuzzy logic", International Journal of Systems Science, V.27, N°12, pp.1243-1250, 1996.*

[*Yue 89*] *Yue.C, Butsuen.T and Hedrick.J.K.: "Alternative control laws for automotive active suspensions", ASME Journal of Dynamic Systems, Measurement and Control, 111, pp.289-291, 1989.*

[*Zad 73*] *L.A.Zadeh: "Outline of a new approach to the analysis of complex systems and decision process". IEEE Trans.Systems, Man and Cybernetics.3, pp.28-44, 1973.*

Annexe

Paramètres des différents modèles de suspension utilisés:

- *Premier modèle de quart du véhicule de [Lou 96]:*

La masse du quart du châssis du véhicule: M=30 kg
La masse de la roue: m=1 kg
La raideur du pneu: h=100 000 N/m
La raideur du ressort de la suspension conventionnelle: k=12 000 N/m
L'amortisseur de la suspension conventionnelle: d=200 Ns/m

- *Deuxième modèle de quart du véhicule de [Lou 96]:*

La masse du quart du châssis du véhicule: M=35 kg
La masse de la roue: m=3.3 kg
La raideur du pneu: h=80 000 N/m
La raideur du ressort de la suspension conventionnelle: k=3 000 N/m
L'amortisseur de la suspension conventionnelle: d=250 Ns/m

- *Modèle de quart du véhicule de [Tho 76]:*

La masse du quart du châssis du véhicule: M=288.9 kg
La masse de la roue: m=28.58 kg
La raideur du pneu: h=155 900 N/m
La raideur du ressort de la suspension conventionnelle: k=29 038 N/m
L'amortisseur de la suspension conventionnelle: d=247 Ns/m

- *Modèle de la moitié du véhicule de [Lou 92]:*

La masse de la moitié du châssis du véhicule: M=505.10 kg
La masse de l'essieu et la roue avant: M_1=28.58 kg
La masse de l'essieu et la roue arrière: M_3=54.43 kg
Le moment d'inertie du châssis: J=651 kg.m^2
La raideur du pneu avant: h_1=155900 N/m
La raideur du pneu arrière: h_3=155900 N/m
l_1=1.0978 m
l_2=1.4676 m