

16/04

REPUBLIQUE ALGERIENNE DEMOCRATIQUE ET POPULAIRE
MINISTERE DE L'ENSEIGNEMENT SUPERIEUR ET DE LA RECHERCHE
SCIENTIFIQUE

ECOLE NATIONALE



POLYTECHNIQUE

El Harrache-Alger

الدرسة الوطنية المتعددة التقنيات
BIBLIOTHEQUE — المكتبة
Ecole Nationale Polytechnique

Projet de Fin d'Etudes

En vue de l'obtention d'un diplôme d'ingénieur d'état en Automatique

Thème

**Modélisation et commande
par réseaux de neurones
d'une suspension active**

Réalisé par : Younes MOUSSOUS

Dirigé par : Mr. D. BOUKHETELLA
Mr. F. BOUDJEMAA

Promotion 2003/2004

REMERCIEMENT

Mes sincères remerciements pour mes deux promoteurs
Mr BOUHKETTALA et Mr BOUDJEMAA, pour la confiance qu'ils m'ont
accordée, en me confiant ce sujet, pour leur soutien moral et leurs orientations
leur aide en documentation, m'a été le grand soin pendant toute la période de
déroulement de ce travail.

Mes remerciements se portent également, a tout les enseignants du département
d'Automatique, en particulier Mr Chekireb, Mr Tadjine et Mr Hamici.

En fin, je remercie tous ceux qui ont participé de près ou de loin à la réalisation
de ce travail.

DEDICACE

Je dédie ce modeste travail

A ceux qui m'ont donné la vie symbole de
fierté et patience.

A ceux qui m'ont offert le bonheur : Mes très chers parents ;

A

Mon cher frère Brahim ;

Mes chères sœurs, Zineb, Fatima, et la meilleure Khira ;

Tous mes collègues de Poly techniques, en particuliers :

Smail.B, Mouh.B, Kamatchou, Chergou et Boudjela.

Tous mes amis de la cité :

Ahcéne, Hakim et Nabil

Et tous les amis du quartier, notamment :

Rezki, Hocine, Farid, Makhoulf et mouh.

Younes Moussous.

إن العمل المقدم من خلال هذه المذكرة يتمثل في تأليف قانون قيادة متدلية سيارة باستعمال تقنيات التحكم عن طريق الشبكة العصبية. لهذا الغرض قمنا في أول الأمر باستخراج مختلف النماذج التي تشرح نظام عمل متدلية سيارة. تعرضنا من خلال هذا العمل إلى النموذجين الخطي و اللخطي للنظامين ربع و نصف السيارة. القسم الثاني من المذكرة كان مخصصا لشرح أنظمة عمل الشبكة العصبية و ذكر مختلف النمط المستعملة و بعض تقنيات التعلم و التحكم التي توفرها مثل هذه الأنظمة. في القسم الثالث و الأخير من هذه المذكرة قمنا باستخراج أربعة أنظمة تحكم لكل من النموذجين الخطي و اللخطي للنمطين ربع و نصف سيارة باستعمال طريقتي الباكمتيبينيق و رجوع الحالة المرشح. ثم تركيب هذه القوانين باستعمال شبكة عصبية مناسبة.

Le travail présenté dans ce mémoire, concerne la synthèse d'une loi de commande neuronale, assurant la stabilité en boucle fermée de la suspension active d'un véhicule. Pour se faire, on a commencé par la présentation des différentes modélisations décrivant le fonctionnement d'une suspension automobile. On a cité les deux types de modélisation, à savoir la modélisation linéaire et non linéaire, pour les système quart et demi véhicule.

Le deuxième axe de notre travail est dédié à l'étude des réseaux de neurones. On avait donc, cité les différents types de réseaux existant dans la littérature, les divers types et algorithmes d'apprentissage neuronal, et quelque méthode de la commande par réseaux de neurones.

En troisième et dernière partie, nous avons synthétisé quatre lois de commande pour les quatre modèles cités, en utilisant l'approche de backstepping et des bouclages filtrés pour les modèles quart et demi véhicule respectivement. Les lois de commande ainsi synthétisées, sont supervisées par un réseau de neurones convenable. Pour tous les cas, des testes de simulation ont été faits à fin de comparer les performances.

The work presented in this memory, relates to the synthesis of a law of neuronal order, ensuring stability in closed loop of the active suspension of a vehicle. To be done, one started with the presentation of various modelings describing the operation of an automobile suspension. One quoted the two types of modelling, namely modelling linear and nonlinear, for the system quarter and half conveys.

The second axis of our work is dedicated to the study of the neurone networks. One had thus quoted the various types of networks existing in the literature, the various types and algorithms of neuronal training, and some method of the order by neurons networks.

In third and last part, we synthesized four laws of order for the four quoted models, by using the approach of back stepping and of the looping filtered for the models quarter and half conveys respectively. A suitable neurons network supervises the laws of order thus synthesized.

Sommaire

	Page
Introduction générale.....	1
Chapitre 2. Modélisation des suspensions automobiles.....	7
I. Modélisation linéaire.....	7
I. 1. Modèle quart de véhicule.....	7
I. 2. Modèle demi véhicule.....	8
II. Modélisation non linéaire.....	11
II. 1. Modèle quart de véhicule.....	11
II. 2. Modèle demi véhicule.....	13
II. 3. Simulation en boucle ouverte.....	17
Conclusion.....	22
Historiques des réseaux de neurones	23
Chapitre 2. Réseaux de neurones artificiels.....	27
I. . Le neurone formel.....	27
I. 1. Le neurone formel de base.....	27
I. 2. Le neurone formel à fonction d'activation.....	27
I. 3. Le réseau de neurones.....	29
I. 3.1. Les réseaux de neurones non bouclés.....	31
I. 3.2. Les réseaux de neurones bouclés.....	32
I. 4. Les réseaux de neurones, approximateurs universels.....	33
II. . Apprentissage des réseaux de neurones.....	34
II. 1. Principe d'apprentissage.....	35
II. 2. Procédure d'apprentissage.....	35
II. 3. La procédure de la validation croisée.....	37
II. 4. Types d'apprentissage.....	38
II. 4.1 Apprentissage non supervisé.....	38
a) Règle de HEEB.....	38
a) Cartes auto adaptatives de KOHONEN.....	39
II. 4.2. Apprentissage supervisé.....	40
a) Règle du perceptron.....	40

b) Algorithme de rétro propagation.....	41
II.4.3 Apprentissage semi supervisé.....	45
II. .5. Problèmes d'apprentissage.....	45
Conclusion.....	46
Chapitre.3 Réseaux de neurones et la commande des processus.....	48
I. 1. Le processus à commander.....	48
I. 2. Conception d'un organe de commande.....	49
a. Modélisation du processus.....	49
b. Identification du processus physique.....	49
c. Conception de l'organe de commande.....	49
d. Estimation des paramètres du correcteur.....	50
I. 3. Techniques de commande par réseaux de neurones.....	50
I. 3.1. Commande supervisée.....	50
I. 3.2. Commande inverse.....	51
a. Méthode d'apprentissage indirect.....	51
b. Méthode d'apprentissage généralisé.....	52
c. Méthode d'apprentissage spécialisé.....	53
I. 3.3. Commande adaptative.....	54
a. Commande adaptative directe.....	54
b. Commande adaptative indirecte.....	55
Conclusion.....	55
Chapitre 4 Application des commandes et simulation.....	57
I. Critères de performances pour les suspensions.....	57
1. Objectifs de performances.....	57
1.1. Confort.....	57
1.2. Tenue de route.....	57
1.3. Débattement de la suspension.....	58
2. Critères pour le modèle quart de véhicule.....	58
3. Critères pour le modèle demi véhicule.....	59
II. Stratégie de commande.....	60
II.1. Modèle quart de véhicule.....	60
1. Modélisation linéaire.....	61
2. Modélisation non linéaire.....	65
II.2. Modèle demi véhicule.....	72

I. Introduction générale :

En général, la suspension est un système conçu pour faire suspendre le corps du véhicule à travers les essieux et les roues, de façon à ce que celui-ci, soit isolé des différents chocs, qui se produisent quand le véhicule circule sur une surface inégale. La suspension a une influence sur le confort, sur les performances et sur la sécurité du véhicule. Elle constitue l'ensemble des organes déformables, généralement élastiques, introduits dans la conception d'un véhicule. Les tâches essentielles d'une suspension sont :

- supporter le poids du véhicule et lui donner une direction fixe sur la route ou la piste.
- Absorber les vibrations provenant de la chaussée pour éviter de les faire subir aux usagers.

La première tâche nécessite, en général, un système de suspension rigide (raide), alors que la seconde, à l'opposé, fait appel à un système de suspension souple (tendre), d'où l'apparition d'un dilemme confort sécurité. La suspension doit donc gérer des compromis entre les objectifs de confort et de tenue de route.

Il est difficile de concevoir une suspension assurant à la fois, un bon confort de roulement et de bonnes caractéristiques de tenue de route. Une voiture possédant une suspension souple, favorise un roulement en douceur et confortable, mais sacrifie un peu les capacités d'accélération, de freinage et de virage. Inversement, les voitures de sport ou les véhicules utilitaire de loisir, possèdent, dans leur version standard, une suspension ferme qui assure une bonne tenue de route, au détriment partiel du confort de roulement. Le but de l'automatique, appliquée à la suspension, est de gérer les relations entre les sollicitations et l'état du système. De plus, la recherche sur la commande des suspensions permet une amélioration constante du confort et de la sécurité des véhicules à tout niveau.

Le confort et la sécurité de conduite d'un véhicule, dépendent intrinsèquement des caractéristiques de son système de suspension qui lie les roues et le corps de véhicule. Les systèmes de suspension sont classés en ; des systèmes passifs (PSS : Passive Suspension Systems), des systèmes semi actifs (SASS : Semi Active Suspension Systems) et des systèmes actifs (ASS : Active Suspension Systems).

Dans le cas des PSS, qui se composent, typiquement, de ressorts et amortisseurs passifs, Les caractéristiques du confort et de tenue de route, sont déterminées lors de leur conception (à savoir la raideur et le coefficient d'amortissement du système), et ne varient qu'avec l'usure de ses éléments constitutifs. Dans ce cas rien n'est commandé.

Lorsque le système est dit du type SASS, seul Le coefficient d'amortissement du système est commandé. Dans ce cas, la force développée peut être commandée dans un ensemble ou un intervalle limité de coefficients.

Dans le cas des ASS, les éléments sont des pompes ou des pistons, qui fournissent alors une force indépendante du débattement de la suspension ou de sa vitesse, l'ASS maintient, donc, l'équilibre du véhicule contre n'importe quel bruit externe. Par conséquent, l'ASS fournit de meilleurs confort et sécurité de conduite par rapport aux systèmes conventionnels de suspension.

Notre contribution à travers ce travail, porte sur la modélisation et la commande des suspensions actives d'automobiles.

Plusieurs travaux ont été élaborés dans ce cadre, on cite entre autres, le travail de *Ramerez-Mendoza* publiés en 1997, [8,9], sur la modélisation et la commande de véhicule automobile, portant principalement sur le contrôle du comportement latérale du véhicule, le travail de *Alvares-Salas*, [8,10], lors de son DEA sur la modélisation de suspension automobile, en 1998 et portant notamment, sur le modèle demi véhicule de type essieu, la thèse de doctorat de *Damien Sammier*, [8], présentée en 2001 portant sur la modélisation et la commande de véhicules automobiles, le travail de *Gobbi et al* en 1999, [8], qui présentent une méthode pour optimiser un modèle complet de véhicule prenant en compte, notamment, les modèles de pneu, les couples moteur ou de freinage sur chaque roue, la géométrie des suspension ainsi que le siège du conducteur, et bien d'autres qu'on présentera ultérieurement en citant les différents modèles de suspensions.

Plusieurs techniques de commandes, utilisées pour le contrôle des suspensions automobiles, existent dans la littérature, on cite entre autres [8] :

- La commande non linéaire, utilisée par *Fialho & Balas* en 2002, pour développer un correcteur non linéaire, visant, soit le débattement de la suspension, soit le mouvement de la caisse, selon que l'écrasement de la suspension est proche ou non de ses limites.
- La commande Skyhook, synthétisée par *Emura et al* en 1995, pour contrôler l'amortisseur semi actif d'une suspension, en utilisant un modèle quart de véhicule à 3ddl. A leur part, *Autran et al*, ont utilisé cette approche à fin de synthétiser une loi de commande pour les modèles, quart de véhicule à 1ddl et véhicule complet à 3ddl (les mouvements des roues sont négligés).
- La commande LQG : développée par *Elbeheiry et al* en 1996, pour contrôler les suspensions d'un modèle complet de véhicule à 7ddl, synthétisée aussi, par *Thompson et Chaplin* en 1996, pour le contrôle des modèles quart de véhicule. A ça part, *Ramirez Mendoza* (1997) a utilisé la commande LQG, pour contrôler la suspension d'un modèle quart de véhicule à 2ddl.
- La commande par placement de pôles, utilisée pour contrôler les oscillations longitudinales d'un véhicule, par *Richard et al* en 1997.
- La commande par logique floue, appliquée par *Cherry et Jones* en 1995, au contrôle d'un modèle quart de véhicule d'une suspension semi active (la commande est appliquée à l'amortisseur actif), elle est aussi développée, pour le contrôle d'une suspension active d'un modèle quart de véhicule, par *Titli et Boverie* en 1995.

Quant à notre travail, il se décompose, alors en quatre chapitres :

- Dans le premier chapitre, nous donnons la présentation des modèles linéaires et non linéaires, utilisés pour représenter la suspension d'un véhicule, dans un objectif de commande. Nous exposons alors, les deux modèles (quart et demi véhicule). Pour chaque type de modèle de véhicule, nous présentons les modèles des éléments qui le constituent, et les équations les décrivant.
- Le deuxième chapitre de ce mémoire, est dédié à l'étude des propriétés des réseaux neuronaux, et présentation de quelques travaux dans la littérature des RNA, ainsi que pour

la description des différents types d'apprentissage neuronal, nous décrivons, pour chaque type les différentes caractéristiques, et algorithmes utilisés à cette fin.

- Le troisième chapitre concerne, la présentation de quelques techniques de la commande neuronale, en précisant leurs avantages et leurs reproches.
- Dans le quatrième et dernier chapitre, nous introduisons la technique de commande par Backstepping, pour laquelle, nous synthétisons une loi de commande permettant d'assurer la stabilité en boucle fermée, des modèles quart de véhicule linéaire et non linéaire. Pour le cas non linéaire, et vu que la représentation d'état du système n'est pas en cascade, un bouclage non linéaire, a été introduit pour faciliter la synthèse de la commande par backstepping. Les commandes ainsi élaborées, ont été supervisées par des réseaux de neurones à fin de comparer les performances obtenues.

Pour les modèles demi véhicule, deux approches différentes ont été adoptées pour les cas linéaire et non linéaire. Deux bouclages avec filtres, sont utilisés pour fournir les lois de commande, satisfaisant les critères de performances fixées pour le modèle demi véhicule linéaire, et qui sont, ensuite, supervisées par un réseau de neurones. Pour le cas non linéaire, nous avons synthétisé deux régulateurs, pour contrôler les parties avant et arrière du modèle indépendamment. Ces deux régulateurs sont ensuite, supervisés par un seul correcteur neuronal.

Pour tous les cas, des résultats de simulations ont été donnés pour valider les approches proposées.

En fin, on terminera par une conclusion générale et commentaires.

Chapitre 1

*Modélisation des
suspensions
automobiles*

Chapitre 1 :

Modélisation des suspensions automobiles

Introduction :

Dans ce chapitre nous présentons quatre modèles différents de suspension de véhicule, cités dans l'ordre croissant du nombre de mouvements pris en compte, et de leur complexité. Ces modèles seront utilisés, ultérieurement, pour la synthèse de lois de commande. Le choix d'un modèle de suspension de véhicule, dépend des mouvements auxquels on s'intéresse. Les principaux mouvements généralement pris en compte, sont les déplacements verticaux, latéraux, longitudinaux, le roulis, le lacet et le tangage des différents éléments constituant le système de suspension d'un véhicule.

I. Modélisation linéaire :

Pour obtenir un modèle linéaire, on choisit des valeurs constantes des raideurs et des amortissements de la suspension et du pneumatique, on ne prend pas en compte, aussi, la dynamique non linéaire des actionneurs.

I. 1. Modèle quart de véhicule ou mono roue : [8], [2]

Deux modèles peuvent être définis selon le but recherché (confort seul, ou confort et tenue de route). Le modèle le plus simple est le modèle quart de véhicule à un degré de liberté. Il prend en compte seulement le mouvement vertical de la caisse, et pas celui de la roue. Ce modèle, appelé modèle à (1ddl) est décrit par *Hrovat* (1997) [8], comme étant un modèle de référence du comportement désiré de la caisse du véhicule. Il peut être modélisé, par une masse représentant le quart de la masse de la caisse, relié au sol par une suspension (ressort et amortisseur et actionneur commandé).

Le modèle quart de véhicule à deux degrés de liberté, prend en compte le mouvement vertical de la caisse et celui de la roue, c'est un modèle plus complexe mais plus représentatif du système. Ce modèle à (2ddl) est décrit, par exemple, par (*Thompson & Chaplin* en 1996), par (*Liberzon et al* en 1998) [8] et par (*Supavut Chantranuwathana et Huei Peng*) en 2000 [2]. Il est modélisé par une masse suspendue, représentant le quart de la caisse, relié à une masse non suspendue, représentant la roue et une partie de l'essieu.

Dans la figure (1) les variables $m_s, m_{us}, k_s, k_{us}, C_s$ et C_{us} représentent, respectivement, la masse, la raideur et l'amortissement, correspondant aux masses suspendue et non suspendue. Les variables X_c, X_w et X_r sont les déplacements de la caisse, de la roue et le profil de la route (perturbation), F_a représente la force générée par l'actionneur hydraulique.

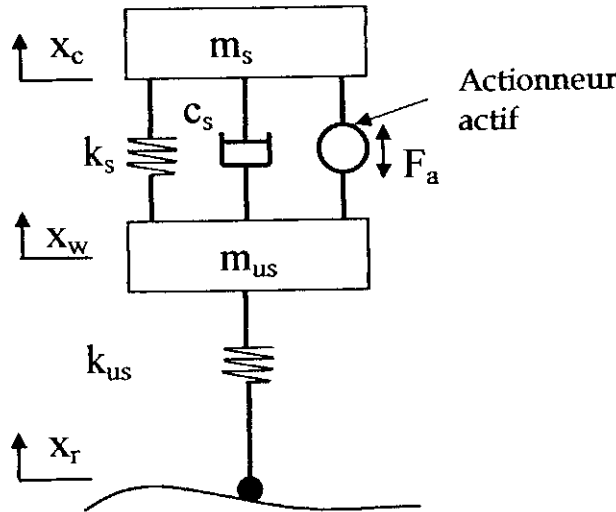


Fig.1 Modèle quart de véhicule à 2ddl

La dynamique linéaire de ce modèle est décrite par les équations suivantes :

$$m_s \ddot{X}_c = k_s (X_w - X_c) + c_s (\dot{X}_w - \dot{X}_c) + F_a \tag{1.1}$$

$$m_{us} \ddot{X}_w = k_{us} (X_r - X_w) - k_s (X_w - X_c) - c_s (\dot{X}_w - \dot{X}_c) - F_a \tag{1.2}$$

La représentation d'état associée aux équations précédentes est donc, la suivante:

$$\dot{X} = A.X + b.u + b_d d \tag{1.3}$$

Où $X = [X_c, \dot{X}_c, X_w, \dot{X}_w]^T$ et $U = F_a$, ou F_a est la force générée par l'actionneur et $d = X_r$ est le profil de la route. Les matrices d'état sont, alors données par :

$$A = \begin{bmatrix} 0 & 1 & 0 & 0 \\ -\frac{k_s}{m_s} & -\frac{c_s}{m_s} & \frac{k_s}{m_s} & \frac{c_s}{m_s} \\ 0 & 0 & 0 & 1 \\ \frac{k_s}{m_{us}} & \frac{c_s}{m_{us}} & -\frac{k_s + k_{us}}{m_{us}} & -\frac{c_s}{m_{us}} \end{bmatrix}, b = \begin{bmatrix} 0 \\ 1 \\ 0 \\ -1 \end{bmatrix}, b_d = \begin{bmatrix} 0 \\ 0 \\ 0 \\ \frac{k_{us}}{m_{us}} \end{bmatrix}$$

I. 2. Modèle demi véhicule : [8], [5]

Deux modèles peuvent être définis, selon les mouvements pris en compte ;

Le modèle demi véhicule de type essieu, qui permet de prendre en compte, en plus des mouvements verticaux, les mouvements de roulis de la caisse et de l'essieu (dans le cas d'un essieu mono bloc). Ce type de modèles est peu représenté dans la littérature, on cite par exemple [8] le modèle de *Dorling* (1996) développé pour les camions. Ce modèle a été adapté, aux véhicules de tourisme, par *Ramirez Mendoza et al.*

Le deuxième type de modèles est le modèle demi véhicule de type bicyclette, qui est utilisé pour tenir compte, en plus des mouvements verticaux, du mouvement de tangage de la caisse du véhicule. C'est ce type de modèle qu'on va présenter ci-dessous.

Les mouvement linéaires (verticaux) et angulaires (tangage) de la caisse, et des roues peuvent être étudiés, en utilisant le modèle demi véhicule de type bicyclette de la figure (2) ;

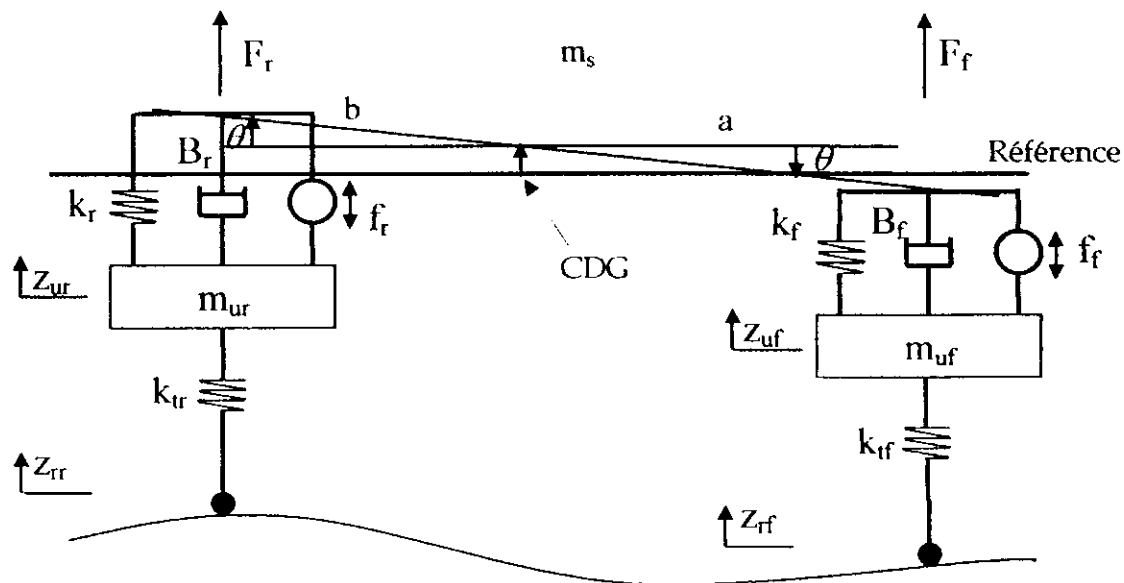


Fig.2 Modèle demi véhicule du type bicyclette.

Pour simplifier les calculs, on considère que les angles de tangage sont assez petits. La masse de la caisse est m_s , son moment d'inertie est J_y . Les masses non suspendues avant et arrière sont respectivement, m_{uf} et m_{ur} , Z_{rr} et Z_{rf} sont les profils de la route (perturbations affectant les roues arrière et avant).

A partir de la figure (2), on peut voir que les déplacements de la caisse sont donnés par :

$$\text{Partie avant ; } Z_{sf} = Z - a \cdot \sin(\theta) \approx Z - a \cdot \theta \quad (1.4)$$

$$\text{Partie arrière ; } Z_{sr} = Z + b \cdot \sin(\theta) \approx Z + b \cdot \theta \quad (1.5)$$

Les forces équivalentes des deux cotés sont les suivantes ;

Partie avant ;

$$F_f = -k_f \cdot (Z_{sf} - Z_{wf}) - B_f \cdot (\dot{Z}_{sf} - \dot{Z}_{wf}) + f_f = -k_f \cdot (Z - a \cdot \theta - Z_{wf}) - B_f \cdot (\dot{Z} - a \cdot \dot{\theta} - \dot{Z}_{wf}) + f_f \quad (1.6)$$

Partie arrière ;

$$F_r = -k_r \cdot (Z_{sr} - Z_{wr}) - B_r \cdot (\dot{Z}_{sr} - \dot{Z}_{wr}) + f_r = -k_r \cdot (Z + b \cdot \theta - Z_{wr}) - B_r \cdot (\dot{Z} + b \cdot \dot{\theta} - \dot{Z}_{wr}) + f_r \quad (1.7)$$

En appliquant la deuxième loi de Newton, et en considérant la position d'équilibre statique, comme origine des déplacements linéaires et angulaires du centre de gravité, on aura ;

$$\ddot{Z} = \frac{1}{m_s} \left\{ \begin{array}{l} -(k_f + k_r) \cdot Z - (B_f - B_r) \cdot \dot{Z} + (a \cdot k_f - b \cdot k_r) \cdot \theta + (a \cdot B_f - b \cdot B_r) \cdot \dot{\theta} \\ + k_f \cdot Z_{wf} + k_r \cdot Z_{wr} + B_f \cdot \dot{Z}_{wf} + B_r \cdot \dot{Z}_{wr} + f_f + f_r \end{array} \right\} \quad (1.8)$$

Pour le mouvement angulaire, on aura :

$$\begin{aligned} J_y \cdot \ddot{\theta} &= -F_f \cdot a \cdot \cos(\theta) + F_r \cdot b \cdot \cos(\theta) = -l_f \cdot a + l_r \cdot b \\ &= \left(\begin{array}{l} a \cdot k_f \cdot (Z - a \cdot \theta - Z_{wf}) + a \cdot B_f \cdot (\dot{Z} - a \cdot \dot{\theta} - \dot{Z}_{wf}) - b \cdot k_r \cdot (Z + b \cdot \theta - Z_{wr}) \\ - b \cdot B_r \cdot (\dot{Z} + b \cdot \dot{\theta} - \dot{Z}_{wr}) - a \cdot f_f + b \cdot f_r \end{array} \right) \end{aligned}$$

Posons $J_y = m_s \cdot r_y^2$ ou r_y est le rayon de giration.

$$\ddot{\theta} = \frac{1}{m_s \cdot r_y^2} \left(\begin{array}{l} (a \cdot k_f - b \cdot k_r) \cdot Z + (a \cdot B_f - b \cdot B_r) \cdot \dot{Z} - (a^2 \cdot k_f + b^2 \cdot k_r) \cdot \theta - (a^2 \cdot B_f + b^2 \cdot B_r) \cdot \dot{\theta} \\ - a \cdot k_f \cdot Z_{wf} + b \cdot k_r \cdot Z_{wr} - a \cdot B_f \cdot \dot{Z}_{wf} + b \cdot B_r \cdot \dot{Z}_{wr} - a \cdot f_f + b \cdot f_r \end{array} \right) \quad (1.9)$$

En appliquant la deuxième loi de Newton aux masses non suspendues (roues), page ci-après, on aura ;

Roue avant ;

$$\begin{aligned}
 m_{uf} \ddot{Z}_{uf} &= B_f (\dot{Z}_{sf} - \dot{Z}_{uf}) + k_f (Z_{sf} - Z_{uf}) - f_f - k_{yf} (Z_{uf} - Z_{rf}) \\
 &= B_f (\dot{Z} - a\dot{\theta} - \dot{Z}_{uf}) + k_f (Z - a\theta - Z_{uf}) - f_f - k_{yf} (Z_{uf} - Z_{rf}) \\
 &= B_f \dot{Z} - aB_f \dot{\theta} - B_f \dot{Z}_{uf} + k_f Z - a k_f \theta - (k_f + k_{yf}) Z_{uf} + k_{yf} Z_{rf} - f_f
 \end{aligned} \quad (1.10)$$

Roue arrière ;

$$\begin{aligned}
 m_{ur} \ddot{Z}_{ur} &= B_r (\dot{Z}_{sr} - \dot{Z}_{ur}) + k_r (Z_{sr} - Z_{ur}) - f_r - k_{yr} (Z_{ur} - Z_{rr}) \\
 &= B_r (\dot{Z} + b\dot{\theta} - \dot{Z}_{ur}) + k_r (Z + b\theta - Z_{ur}) - f_r - k_{yr} (Z_{ur} - Z_{rr}) \\
 &= B_r \dot{Z} + bB_r \dot{\theta} - B_r \dot{Z}_{ur} + k_r Z + b k_r \theta - (k_r + k_{yr}) Z_{ur} + k_{yr} Z_{rr} - f_r
 \end{aligned} \quad (1.11)$$

On définit l'espace d'état comme suit ;

$$\begin{aligned}
 X_1 &= Z. & X_2 &= \dot{Z}. & X_3 &= \theta. & X_4 &= \dot{\theta}. \\
 X_5 &= Z_{sf} - Z_{uf}. & X_6 &= \dot{Z}_{uf}. & X_7 &= Z_{sr} - Z_{ur}. & X_8 &= \dot{Z}_{ur}.
 \end{aligned}$$

La représentation d'état associée aux équations précédentes, est la suivante ;

$$\begin{aligned}
 \dot{X}_1 &= X_2. \\
 \dot{X}_2 &= -\frac{B_f + B_r}{m_s} X_2 + \frac{aB_f - bB_r}{m_s} X_4 - \frac{k_f}{m_s} X_5 + \frac{B_f}{m_s} X_6 - \frac{k_r}{m_s} X_7 + \frac{B_r}{m_s} X_8 + \frac{f_f + f_r}{m_s} \\
 \dot{X}_3 &= X_4. \\
 \dot{X}_4 &= \frac{aB_f - bB_r}{m_s r_y^2} X_2 - \frac{a^2 B_f + b^2 B_r}{m_s r_y^2} X_4 + \frac{a k_f}{m_s r_y^2} X_5 - \frac{a B_f}{m_s r_y^2} X_6 - \frac{b k_r}{m_s r_y^2} X_7 + \frac{b B_r}{m_s r_y^2} X_8 - \frac{a f_f - b f_r}{m_s r_y^2} \\
 \dot{X}_5 &= X_2 - a X_4 - X_6 \\
 \dot{X}_6 &= -\frac{k_{yf}}{m_{uf}} X_1 + \frac{B_f}{m_{uf}} X_2 + \frac{a k_{yf}}{m_{uf}} X_3 - \frac{a B_f}{m_{uf}} X_4 + \frac{k_f + k_{yf}}{m_{uf}} X_5 - \frac{B_f}{m_{uf}} X_6 - \frac{f_f}{m_{uf}} + \frac{k_{yf}}{m_{uf}} Z_{rf} \\
 \dot{X}_7 &= X_2 + b X_4 - X_8. \\
 \dot{X}_8 &= -\frac{k_{yr}}{m_{ur}} X_1 + \frac{B_r}{m_{ur}} X_2 + \frac{b k_{yr}}{m_{ur}} X_3 + \frac{b B_r}{m_{ur}} X_4 + \frac{k_r + k_{yr}}{m_{ur}} X_7 - \frac{B_r}{m_{ur}} X_8 - \frac{f_r}{m_{ur}} + \frac{k_{yr}}{m_{ur}} Z_{rr}.
 \end{aligned} \quad (1.12)$$

$$\text{On obtient ; } \dot{X} = AX + BU + B_d d \quad (1.13)$$

Avec, $U = \begin{bmatrix} f_f \\ f_r \end{bmatrix}$ est l'entrée de commande

$d = \begin{bmatrix} Z_{rf} & Z_{rr} \end{bmatrix}$ est le profil de la route

II. Modélisation non linéaire :

Dans un deuxième lieu de ce chapitre, nous reprendrons les modèles quart et demi véhicule présentés précédemment, mais en tenant compte des non linéarités du système, et ceci en introduisant dans la modélisation, les dynamiques des actionneurs utilisés pour générer les commandes cherchées, il sera donc, nécessaire de connaître les concepts de base de la dynamique de l'actionneur utilisé.

II. 1. Modèle quart de véhicule: [1], [4]

Le modèle quart de véhicule d'une suspension active et son actionneur, sont représentés sur la figure ci après, avec les mêmes significations pour les variables décrivant les différents éléments du système.

La dynamique linéaire de ce modèle, est décrite par les équations suivantes :

$$m_s \cdot \ddot{X}_c = k_s \cdot (X_w - X_c) + c_s \cdot (\dot{X}_w - \dot{X}_c) + F_a$$

$$m_{us} \cdot \ddot{X}_w = k_{us} \cdot (X_r - X_w) - k_s \cdot (X_w - X_c) - c_s \cdot (\dot{X}_w - \dot{X}_c) - F_a$$

L'actionneur actif utilisé, est une servovalve hydraulique. Elle est représentée par la figure (3) ci dessous :

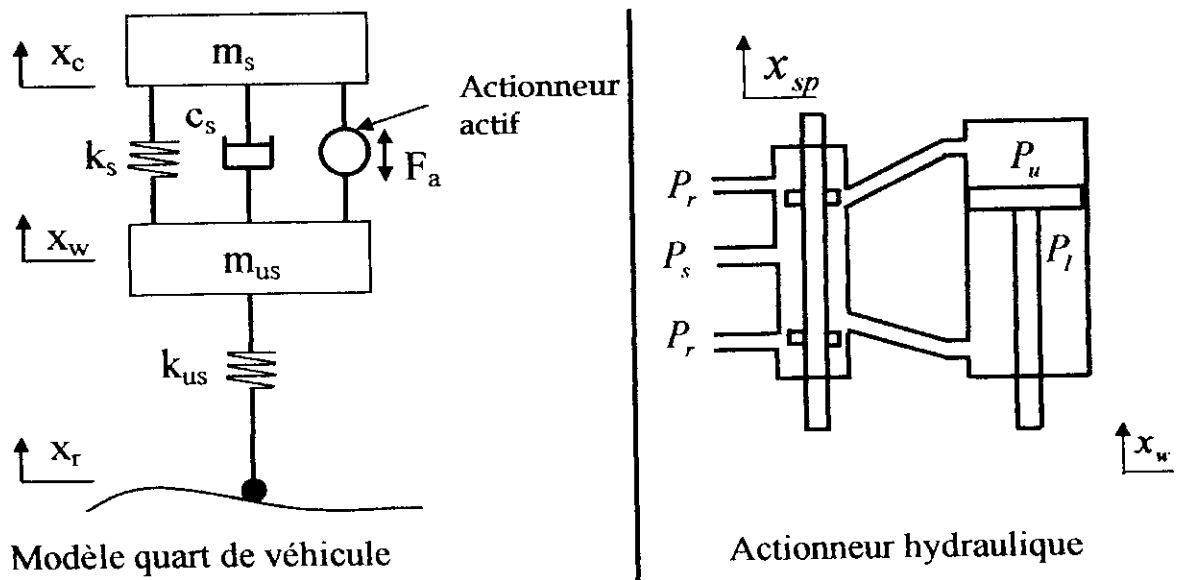


Fig.3.

La commande générée par l'actionneur est donnée par

$$U_a = A.P_L \tag{1.14}$$

Où :

A est la surface du piston et P_L est la chute de pression dans le piston.

D'après [11], le flux hydraulique du liquide utilisé, et la chute de pression dans le piston, satisfont les équations suivantes :

$$\frac{V_t}{4.\beta_e}.\dot{P}_L = Q - C_{ip}.P_L - A.(x_s - x_w) \tag{1.15}$$

$$Q = \text{sgn}[P_s - \text{sgn}(x_v).P_L] C_{d.v}.\omega.x_v.\sqrt{\frac{1}{\rho}|P_s - \text{sgn}(x_v).P_L|} \tag{1.16}$$

Où ;

V_t est le volume totale de l'actionneur, B_e et C_d sont des caractéristiques du fluide utilisé. Q est le flux hydraulique du liquide, ω est le gradient de surface de la valve, x_v est le déplacement du piston de la valve, ρ est la densité du liquide, P_s et P_r sont les pressions d'approvisionnement, et de renvoi allant dans et hors de la soupape.

Finalement, le déplacement de la soupape est contrôlé par l'entrée de la valve, qui peut être un courant ou une tension. La dynamique de la valve est approximée par un filtre linéaire avec une constante de temps τ . C'est une approximation justifiée [11], si la fréquence n'est pas trop élevée, et elle est régulièrement utilisée dans l'industrie.

$$\dot{x}_v = \frac{1}{\tau}.(u - x_v) \tag{1.17}$$

En tenant compte de ces relations, et des équations précédentes, et en posant :

$$\alpha = \frac{4\beta_e}{V_t}, \beta = \alpha.C_{ip}, \gamma = \alpha.C_{d.v}.\omega.\sqrt{\frac{1}{\rho}},$$

$$X_1 = X_C, \quad \text{déplacement de la caisse.} \quad X_2 = \dot{X}_C, \quad \text{Vitesse de la caisse.}$$

$$X_3 = X_w, \quad \text{déplacement de la roue.} \quad X_4 = \dot{X}_w, \quad \text{Vitesse de la roue.}$$

$$X_5 = P_L \quad \text{Pression développée par le piston hydraulique}$$

$$X_6 = X_{sp} = X_v. \quad \text{déplacement de la soupape.} \quad U = k_{sv}.I_{sv} \quad \text{entrée de commande.}$$

On obtient le système d'état suivant :

$$\begin{aligned}
 \dot{X}_1 &= X_2 \\
 \dot{X}_2 &= -\frac{1}{m_s} \{k_s \cdot (X_1 - X_3) + C_s \cdot (X_2 - X_4) - A \cdot X_5\} \\
 \dot{X}_3 &= X_4 \\
 \dot{X}_4 &= \frac{1}{m_{US}} \{k_s \cdot (X_1 - X_3) + C_s \cdot (X_2 - X_4) - k_{US} \cdot (X_3 - X_r) - A \cdot X_5\} \\
 \dot{X}_5 &= -\beta \cdot X_5 - \alpha \cdot A \cdot (X_2 - X_4) + \gamma \omega_3 \cdot X_6 \\
 \dot{X}_6 &= \frac{1}{\tau} (u - X_6) \\
 \text{avec } \omega_3 &= \text{sgn}[P_s - \text{sgn}(X_6) \cdot X_5] \sqrt{|P_s - \text{sgn}(X_6) \cdot X_5|}
 \end{aligned} \tag{1.18}$$

II. 2. Modélisation d'un demi véhicule: [1], [4], [5], [6]

De la même manière que le modèle quart de véhicule, l'introduction des dynamiques des actionneurs, donnent lieu à des non linéarités dans la modélisation du système considéré. L'actionneur actif utilisé est tous jour une servovalve hydraulique. On reprend donc le schéma précédent du modèle demi véhicule pour extraire les équations décrivant le système.

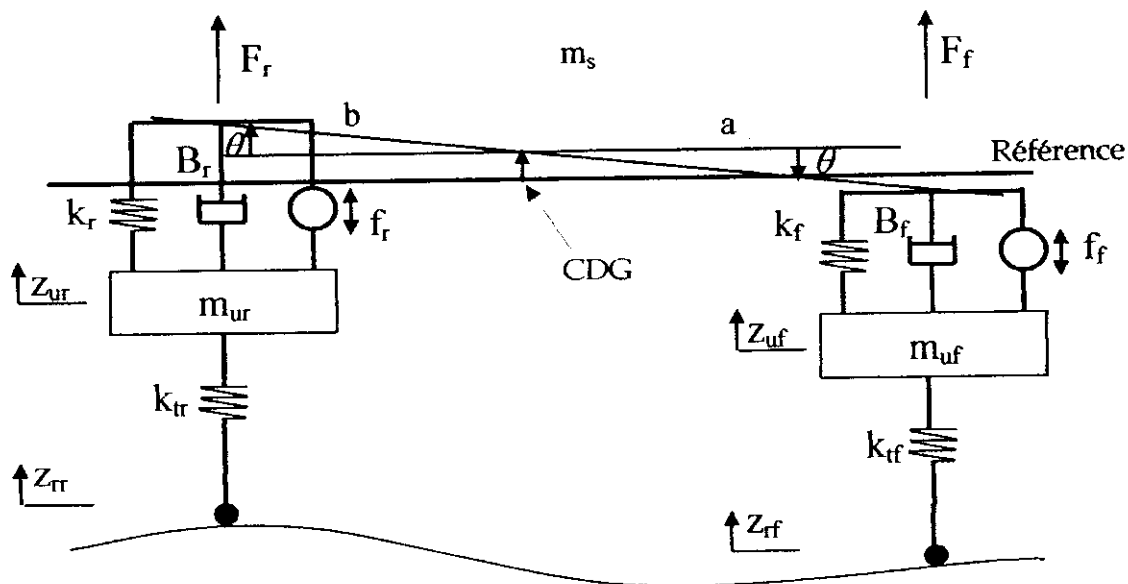


Fig.4 Modèle demi véhicule du type bicyclette.

A partir de la figure (4), on aura :

$$\text{Partie avant ; } Z_{sf} = Z - a.\sin(\theta) \quad (1.19)$$

$$\text{Partie arrière ; } Z_{sr} = Z + b.\sin(\theta) \quad (1.20)$$

Les forces équivalentes des deux cotés sont les suivantes ;

Partie avant ;

$$\begin{aligned} F_f &= -k_f.(Z_{sf} - Z_{uf}) - B_f.(\dot{Z}_{sf} - \dot{Z}_{uf}) + f_f \\ &= -k_f.(Z - a.\sin(\theta) - Z_{uf}) - B_f.(\dot{Z} - a\dot{\theta}.\cos(\theta) - \dot{Z}_{uf}) + f_f \end{aligned} \quad (1.21)$$

Partie arrière ;

$$\begin{aligned} F_r &= -k_r.(Z_{sr} - Z_{ur}) - B_r.(\dot{Z}_{sr} - \dot{Z}_{ur}) + f_r \\ &= -k_r.(Z + b.\sin(\theta) - Z_{ur}) - B_r.(\dot{Z} + b\dot{\theta}.\cos(\theta) - \dot{Z}_{ur}) + f_r \end{aligned} \quad (1.22)$$

En appliquant la deuxième lois de Newton, et en considérant la position d'équilibre statique comme origine des déplacements linéaires et angulaires du centre de gravité, on aura ;

$$\ddot{Z} = \frac{1}{m_s} \left\{ \begin{aligned} &-(k_f + k_r).Z - (B_f - B_r).\dot{Z} + (a.k_f - b.k_r).\sin(\theta) + (a.B_f - b.B_r).\dot{\theta}.\cos(\theta) \\ &+ k_f.Z_{uf} + k_r.Z_{ur} + B_f.\dot{Z}_{uf} + B_r.\dot{Z}_{ur} + f_f + f_r \end{aligned} \right\} \quad (1.23)$$

Pour le mouvement angulaire, on aura :

$$\begin{aligned} J_y.\ddot{\theta} &= -F_f.a.\cos(\theta) + F_r.b.\cos(\theta) \\ &= \left(\begin{aligned} &a.k_f.(Z - a.\sin(\theta) - Z_{uf}).\cos(\theta) + a.B_f.(\dot{Z} - a\dot{\theta}.\cos(\theta) - \dot{Z}_{uf}).\cos(\theta) \\ &- b.k_r.(Z + b.\sin(\theta) - Z_{ur}).\cos(\theta) - b.B_r.(\dot{Z} + b\dot{\theta}.\cos(\theta) - \dot{Z}_{ur}).\cos(\theta) \\ &- a.\cos(\theta).f_f + b.\cos(\theta).f_r \end{aligned} \right) \end{aligned}$$

Posons $J_y = m_s.r_y^2$ ou r_y est le rayon de giration.

$$\ddot{\theta} = \frac{1}{m_s.r_y^2} \left(\begin{aligned} &(a.k_f - b.k_r).Z.\cos(\theta) + (a.B_f - b.B_r).\dot{Z}.\cos(\theta) - (a^2.k_f + b^2.k_r).\sin(\theta).\cos(\theta) \\ &- (a^2.B_f + b^2.B_r).\dot{\theta}.\cos^2(\theta) - a.k_f.\cos(\theta).Z_{uf} + b.k_r.\cos(\theta).Z_{ur} \\ &- a.B_f.\cos(\theta).\dot{Z}_{uf} + b.B_r.\cos(\theta).\dot{Z}_{ur} - a.\cos(\theta).f_f + b.\cos(\theta).f_r \end{aligned} \right) \quad (1.24)$$

En appliquant la deuxième loi de Newton aux masses non suspendues (roues), on aura ;

Roue avant ;

$$m_{uf} \ddot{Z}_{uf} = (B_f \cdot (\dot{Z} - a \dot{\theta} \cos(\theta) - \dot{Z}_{uf}) + k_f \cdot (Z - a \sin(\theta) - Z_{uf}) - f_f - k_{rf} \cdot (Z_{uf} - Z_{rf})) \quad (1.25)$$

Roue arrière ;

$$m_{ur} \ddot{Z}_{ur} = (B_r \cdot (\dot{Z} + b \dot{\theta} \cos(\theta) - \dot{Z}_{ur}) + k_r \cdot (Z + b \sin(\theta) - Z_{ur}) - f_r - k_{rr} \cdot (Z_{ur} - Z_{rr})) \quad (1.26)$$

En tenant compte des dynamiques non linéaires des actionneurs (on considère les mêmes actionneurs que le cas d'un quart de véhicule), et en choisissant les variables d'états suivantes,

$$\begin{aligned} X_1 = Z, & \quad X_2 = \dot{Z}, & X_3 = \theta, & \quad X_4 = \dot{\theta}, & X_5 = Z_{uf}, & \quad X_6 = \dot{Z}_{uf}, \\ X_7 = Z_{ur}, & \quad X_8 = \dot{Z}_{ur}, & X_9 = P_{L1}, & \quad X_{10} = P_{L2}, & X_{11} = X_{v1}, & \quad X_{12} = X_{v2}, \end{aligned}$$

Où :

X_{v1} et X_{v2} Représentent les positions des deux pistons des deux actionneurs, et P_{L1} et P_{L2} les chutes de pressions dans les deux pistons des roues avant et arrière.

On aura :

$$\begin{aligned} \dot{X}_1 &= X_2, \\ \dot{X}_2 &= \left(-\frac{k_f + k_r}{m_s} X_1 - \frac{B_f + B_r}{m_s} X_2 + \frac{ak_f - bk_r}{m_s} \sin(X_3) + \frac{aB_f - bB_r}{m_s} \cos(X_3) \cdot X_4 \right. \\ &\quad \left. + \frac{k_f}{m_s} X_5 + \frac{B_f}{m_s} X_6 + \frac{k_r}{m_s} X_7 + \frac{B_r}{m_s} X_8 + \frac{Ap \cdot (X_9 - X_{10})}{m_s} \right) \end{aligned}$$

$$\begin{aligned} \dot{X}_3 &= X_4, \\ \dot{X}_4 &= \left(\frac{ak_f - bk_r}{m_s \cdot r_y^2} X_1 \cdot \cos(X_3) + \frac{aB_f - bB_r}{m_s \cdot r_y^2} X_2 \cdot \cos(X_3) - \frac{a^2 k_f + b^2 k_r}{2 \cdot m_s \cdot r_y^2} \sin(2 \cdot X_3) \right. \\ &\quad \left. - \frac{a^2 B_f + b^2 B_r}{m_s \cdot r_y^2} X_4 \cdot \cos^2(X_3) - \frac{aK_f}{m_s \cdot r_y^2} X_5 \cdot \cos(X_3) - \frac{aB_f}{m_s \cdot r_y^2} X_6 \cdot \cos(X_3) \right. \\ &\quad \left. + \frac{bk_r}{m_s \cdot r_y^2} X_7 \cdot \cos(X_3) + \frac{bB_r}{m_s \cdot r_y^2} X_8 \cdot \cos(X_3) - Ap \cdot \frac{aX_9 - bX_{10}}{m_s \cdot r_y^2} \cdot \cos(X_3) \right) \end{aligned}$$

$$\begin{aligned}
 \dot{X}_5 &= X_6, \\
 \dot{X}_6 &= \left(\begin{array}{l} \frac{k_f}{m_{uf}} X_1 + \frac{B_f}{m_{uf}} X_2 - \frac{ak_f}{m_{uf}} \sin(X_3) - \frac{aB_f}{m_{uf}} X_4 \cdot \cos(X_3) - \frac{k_f + k_{gf}}{m_{uf}} X_5 - \frac{B_f}{m_{uf}} X_6 \\ -\frac{A_p \cdot X_9}{m_{uf}} + \frac{k_{gf}}{m_{uf}} Z_{rf} \end{array} \right) \\
 \dot{X}_7 &= X_8, \\
 \dot{X}_8 &= \left(\begin{array}{l} \frac{k_r}{m_{ur}} X_1 + \frac{B_r}{m_{ur}} X_2 + \frac{bk_r}{m_{ur}} \sin(X_3) + \frac{bB_r}{m_{ur}} X_4 \cdot \cos(X_3) - \frac{k_r + k_{gr}}{m_{ur}} X_7 - \frac{B_r}{m_{ur}} X_8 \\ -\frac{A_p \cdot X_{10}}{m_{ur}} + \frac{k_{gr}}{m_{ur}} Z_{rr} \end{array} \right)
 \end{aligned} \tag{1.27}$$

$$\begin{aligned}
 \dot{X}_9 &= -\beta_f \cdot X_9 - \alpha_f \cdot A_p \cdot (X_2 - a \cdot X_4 \cdot \cos(X_3) - X_6) + \omega_f \cdot \gamma_f \cdot X_{11}, \\
 \dot{X}_{10} &= -\beta_r \cdot X_{10} - \alpha_r \cdot A_p \cdot (X_2 + b \cdot X_4 \cdot \cos(X_4) - X_8) + \omega_r \cdot \gamma_r \cdot X_{12}, \\
 \dot{X}_{11} &= \frac{1}{\tau} \cdot (u_f - X_{11}), \\
 \dot{X}_{12} &= \frac{1}{\tau} \cdot (u_r - X_{12}).
 \end{aligned}$$

$$\text{avec } \omega_f = \text{sgn} \left[P_{sf} - \text{sgn}(X_{11}) \cdot X_9 \right] \sqrt{|P_{sf} - \text{sgn}(X_{11}) \cdot X_9|}, \tag{1.28}$$

$$\omega_r = \text{sgn} \left[P_{sr} - \text{sgn}(X_{12}) \cdot X_{10} \right] \sqrt{|P_{sr} - \text{sgn}(X_{12}) \cdot X_{10}|}.$$

$U = \begin{bmatrix} u_f \\ u_r \end{bmatrix}$ est l'entrée de commande.

$d = \begin{bmatrix} Z_{rf} & Z_{rr} \end{bmatrix}$ est la sollicitation de la route.

Les tests temporels en boucle ouverte pour les différents modèles vus précédemment, sont représentés pour chaque type de modèle.

Les résultats suivants, sont obtenus pour le modèle quart de véhicule, avec prise en compte de la dynamique de l'actionneur (courbes en rouge sur les figures), et sans considération de cette dynamique (courbes en bleu sur les figures).

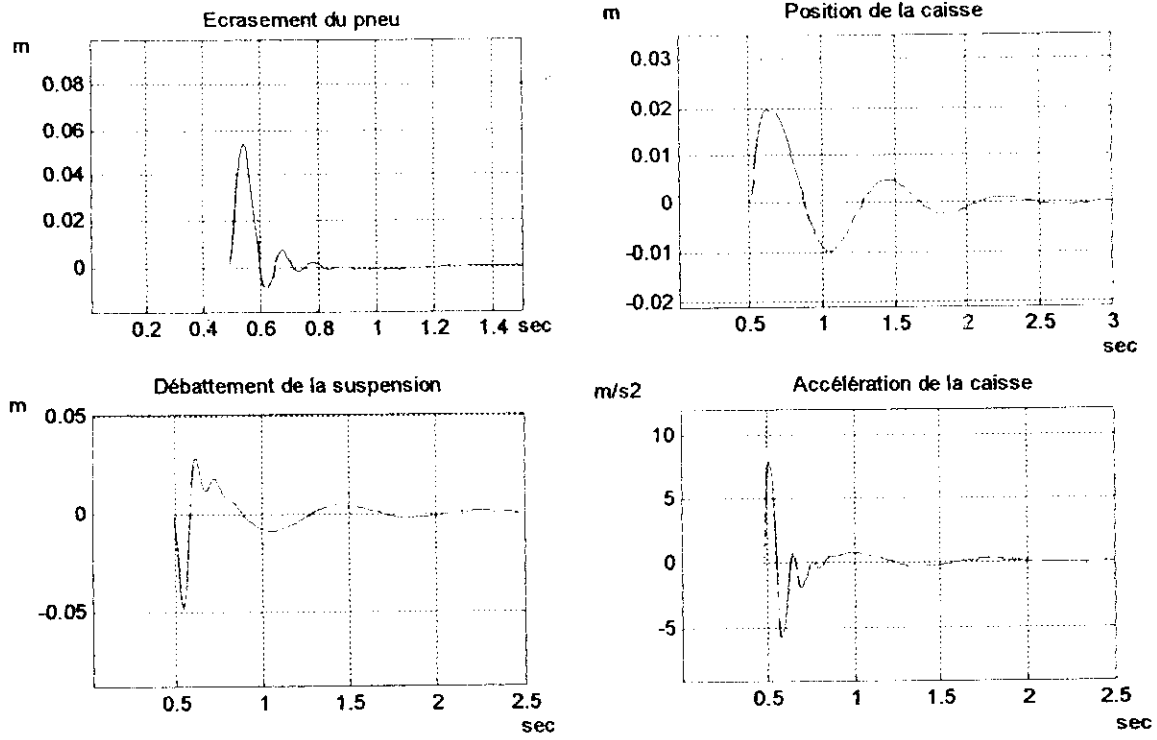


Fig.1. Passage sur une bosse de 05 cm, à une vitesse de 35 Km/h.

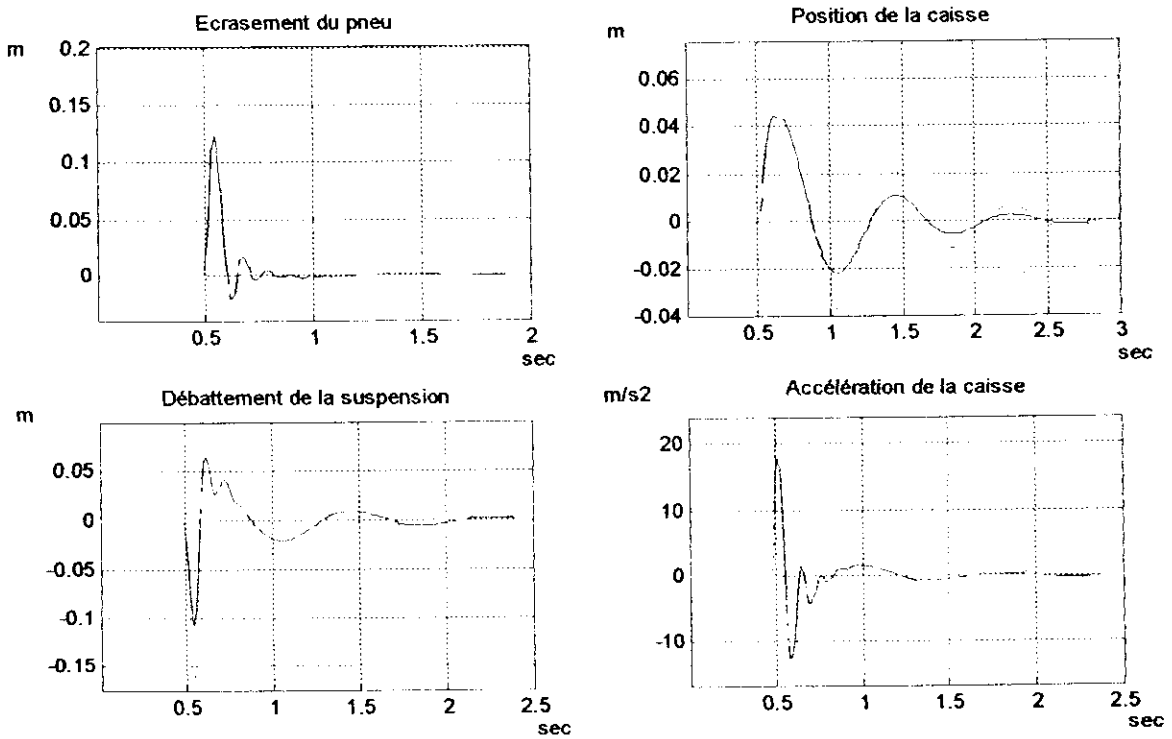


Fig.2. Passage sur une bosse de 11 cm, à une vitesse de 35 Km/h.

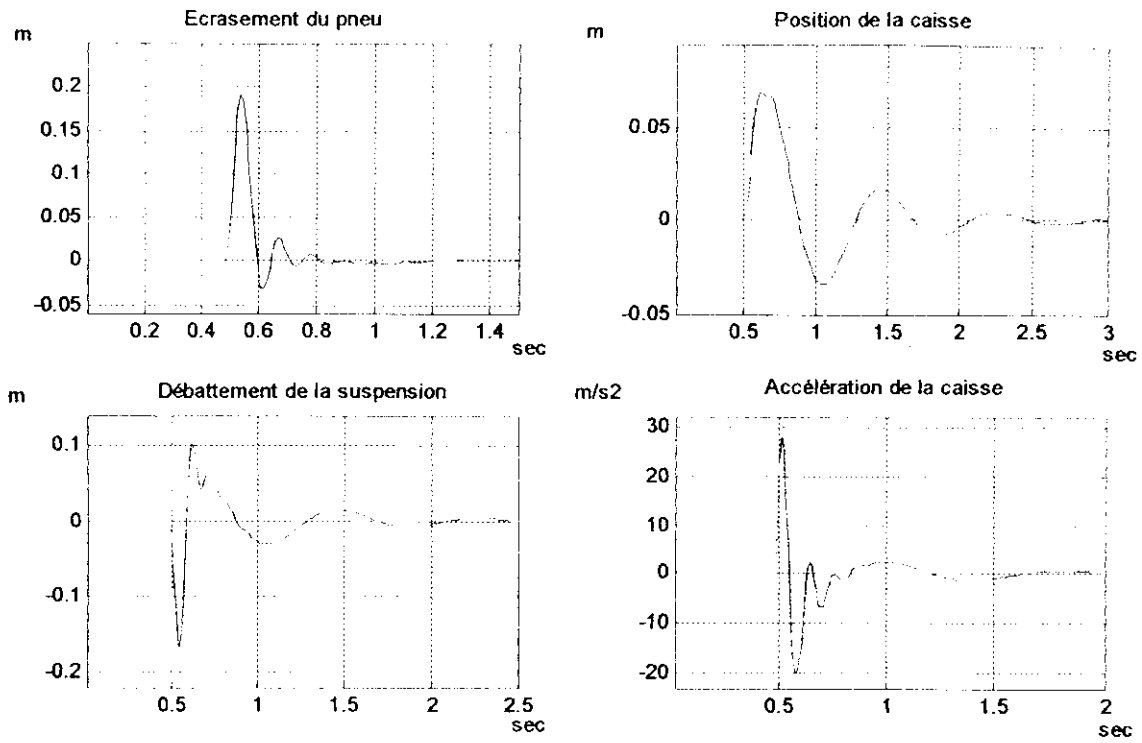


Fig.3. Passage sur une bosse de 15 cm, à une vitesse de 35 Km/h.

Pour les modèles linéaire et non linéaire demi véhicule, les résultats de simulation sont représentés sur les figures ci après :

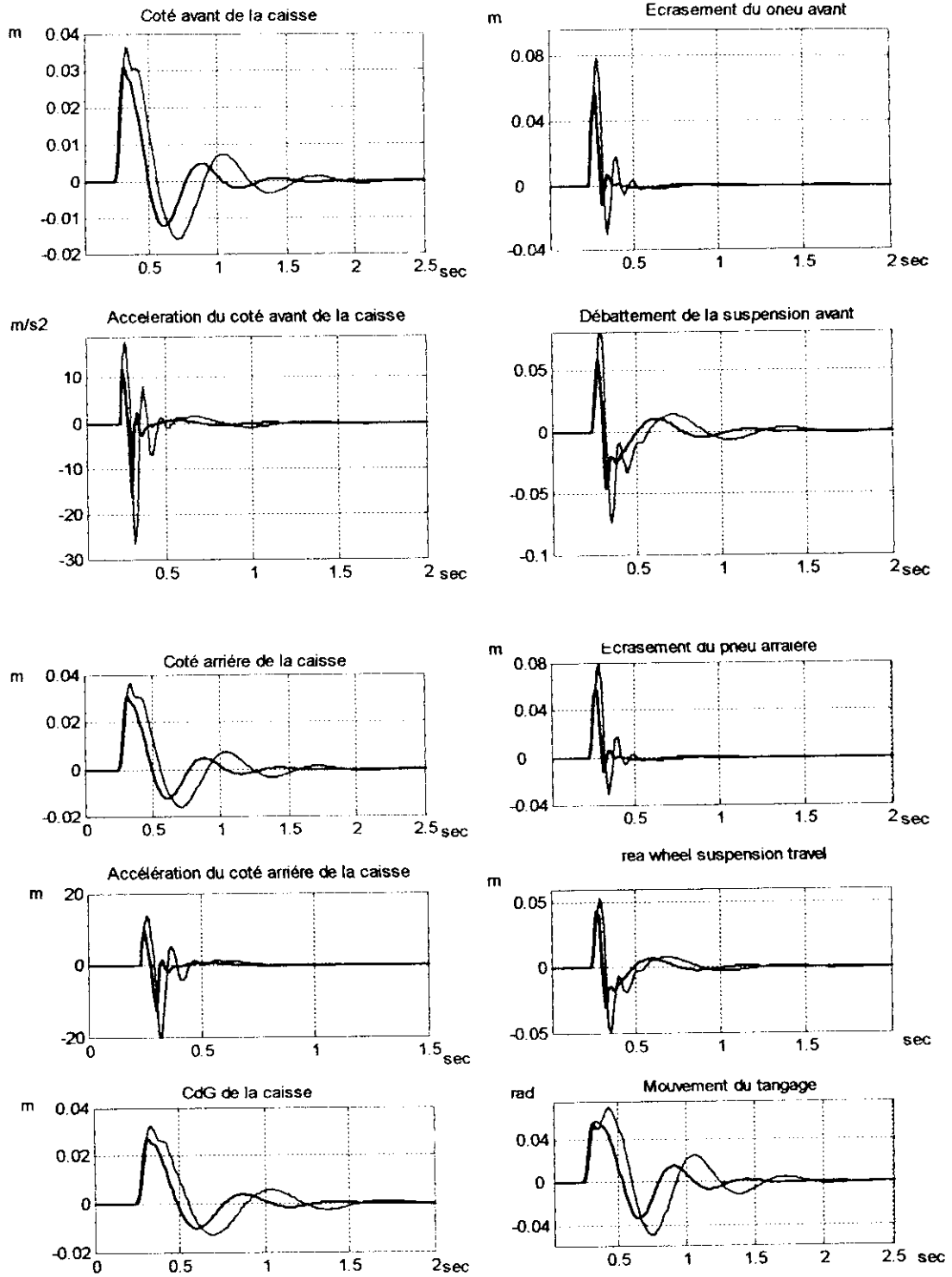


Fig.4. Passage sur une bosse de 05 cm, à une vitesse de 35 Km/h.

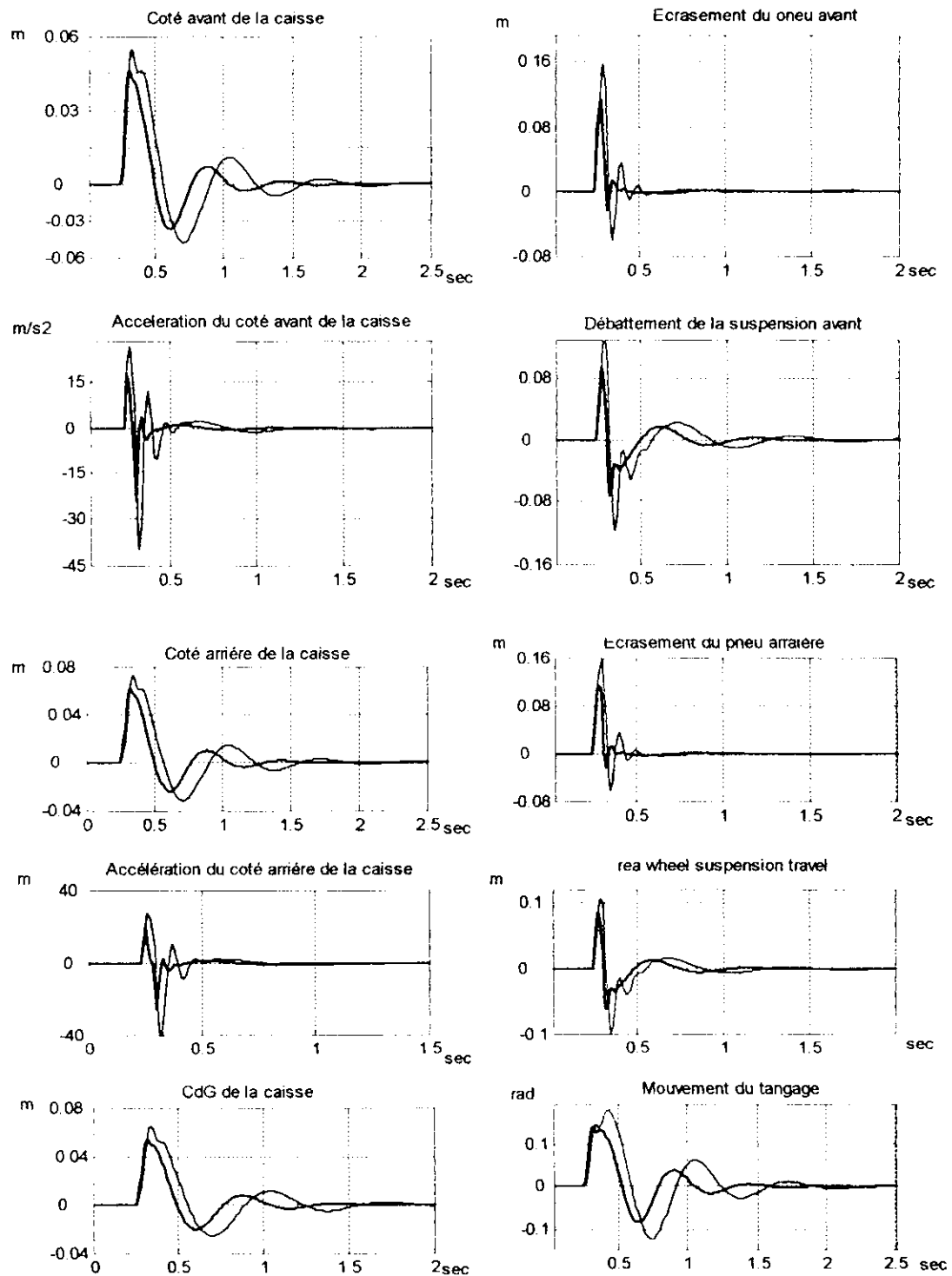


Fig.5. Passage sur une bosse de 10 cm, à une vitesse de 35 Km/h.

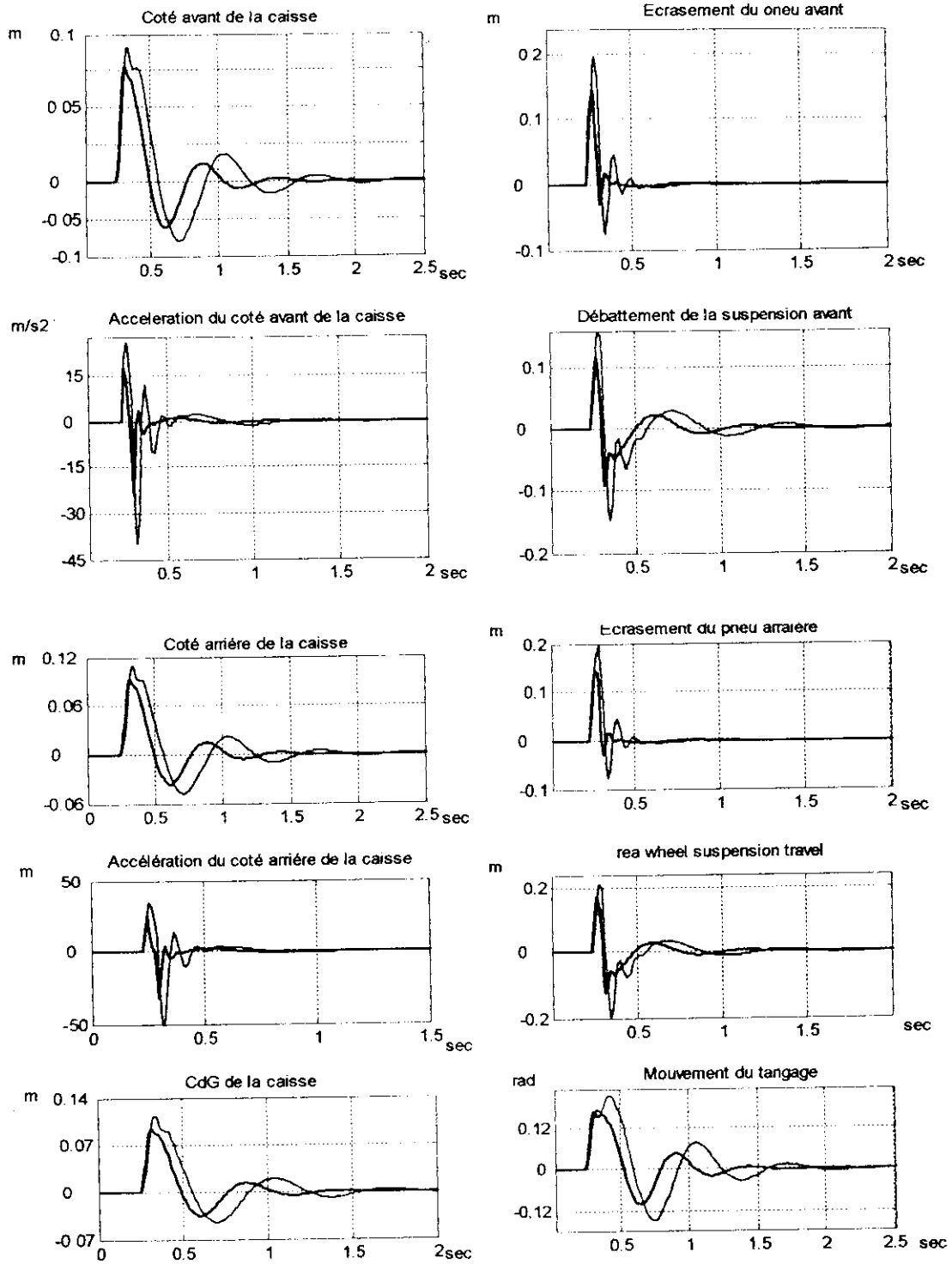


Fig.6. Passage sur une bosse de 15 cm, à une vitesse de 35 Km/h.

Constatation :

A partir des résultats de simulation de la boucle ouverte précédents, on constate que l'introduction des dynamique des actionneurs, quoiqu' elle donne lieu à des calculs plus complexes, mais elle représente au mieux la dynamique du système réel (ceci peut être remarqué, en l'apparition des oscillations, et l'augmentation des amplitude, après la prise en compte de ces dynamiques). La considération de ces dynamiques, lors de la synthèse des lois de commande, améliore donc, certainement les caractéristiques du système réel, et assure, de plus, une meilleure robustesse en stabilité et performance, vis avis, des perturbations, et des incertitudes du système.

Conclusion :

Dans ce preùier chapitre nous avons représenté les deux modèles, quart de et demi véhicule de la suspension d'un véhicule, en négligeant dans un premier temps, et en tenant compte dans un second temps, des dynamiques des actionneurs hydrauliques utilisés. La dynamique des actionneurs quoiqu'elle donne lieu à des équations plus complexes mais elle est plus représentative du système réel. Pour chaque type de modèle de véhicule, nous avons donné les éléments le constituant ainsi que les équations décrivant un modèle de type considéré.

Historique des réseaux de neurones : [11], [12], [13]

Les réseaux de neurones formels sont des systèmes de traitement de l'information, dont la structure s'inspire de celle du système nerveux. Ce terme, regroupe aujourd'hui, un certain nombre de modèles qui tentent à imiter quelques fonctionnalités du cerveau humain. Historiquement, les origines de cette discipline sont très diversifiées, en effet ;

En 1943, *Warren Mc Culloch*, un biologiste, et *Walter Pitts*, un mathématicien, proposaient de simuler le fonctionnement du système nerveux, avec un réseau de neurones formels. Ces «neurones» sont en fait des automates interconnectés entre eux. Ils inventèrent, donc, le premier neurone formel ; un produit scalaire entre un vecteur d'entrée et un vecteur poids, suivi d'un élément à seuil répondant, 0 ou 1, selon que le résultat du produit scalaire est supérieure ou non au seuil.

En 1949, *Hebb* montrera l'importance de l'apprentissage ou couplage synaptique, et proposait le principe des modifications synaptiques comme modèle d'apprentissage.

Après c'est en 1958 que *Frank Rosenblatt*, décrit le premier modèle opérationnelle de réseaux de neurones, le *Perceptron de Frank Rosenblatt*, en se basant sur les idées du *Mc Culloch, Pitts et Hebb* ; c'est un séparateur linéaire qui permet d'apprendre à fournir la bonne décision dans des opérations de classification. Ce qui a donné beaucoup d'espoir, jusqu'à ce qu'en 1969 deux mathématiciens, *Minsky et Papert* démontrèrent les limites mathématiques de ce modèle (le perceptron), et qu'avec, un perceptron à une couche de neurones, on ne pouvait séparer que des données linéairement séparables et que pour des opérations plus générales de classification, il fallait mettre plusieurs couches. Cette période reconnut alors, un désintéressement des chercheurs des réseaux de neurones. Il fallait donc, attendre une quinzaine d'années plus tard pour devenir aujourd'hui l'une des approches les plus en vue de l'intelligence artificielle.

Pendant les années 70 ; période extrêmement riche, les chercheurs comme *Anderson, Kohonen* et bien d'autres ont développée leurs idées, apprenant progressivement à les formaliser, en combinant les méthodes de l'algèbre linéaires et celles de techniques non linéaires.

En 1982, *Hop Field* montrait l'analogie de ces derniers, avec certains systèmes physiques. Il importe donc de définir ce qu'est précisément un réseau de neurones. Il est vrai que le brio des théories de *Hop Field*, a été le moteur d'un nouveau mondial sur les neurones artificiels.

Préambule :

On part tout d'abord, d'un neurone biologique, illustré par la figure ci-après à gauche, pour en créer un neurone formel. Ce neurone formel, essaie de reproduire le fonctionnement d'un neurone biologique à plusieurs entrées d'où proviennent les informations. Elles sont ensuite, analysées au centre, pour finalement donner un résultat à la sortie. La mise en réseau de ces neurones formels, permet de gérer un nombre impressionnant de données à la fois. Les neurones sont reliés entre eux par des liaisons appelées axones.

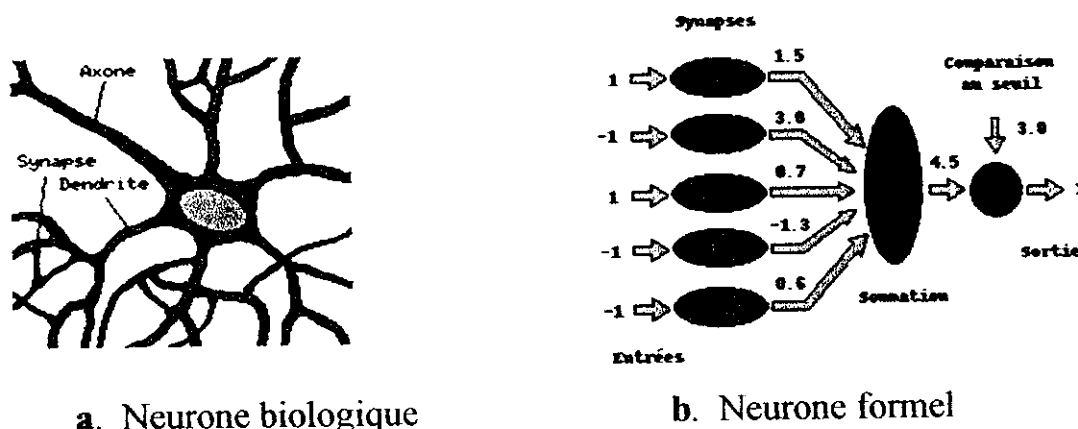


Fig. 5.

Il est à relever que les neurones formels, fonctionnent de manière binaire, étant donné qu'ils n'ont qu'une seule sortie. Suivant les données reçues, ils se mettent en état actif ou inactif, transmettant cette information au neurone suivant, et ainsi de suite.

Il semble toutefois impossible de reproduire le fonctionnement du cerveau humain, sachant que nous possédons plus de 100 milliards de neurones, ce qui représente le traitement de près de 10^5 données.

Les neurones formels constituent une approximation des neurones biologiques, mais ils constituent un moyen intéressant quant à la résolution de nombreux problèmes. Les réseaux de neurones sont en plus évolutifs, c'est-à-dire qu'ils apprennent par l'expérience. Ils représentent l'une des plus importantes révolutions dans le monde de l'informatique depuis l'introduction des ordinateurs numériques. Ce sont des architectures parallèles capables d'apprendre, sans aucune programmation, simplement par apprentissage, ceci en s'inspirant de neurone biologique. Les RN regroupés sont aujourd'hui un certain nombre de modèle dont l'intention est d'imiter certaines fonctions du cerveau humain.

Les RN ont des propriétés fascinantes, comme le parallélisme, la mémoire distribuée, la capacité de généralisation, et la capacité d'adaptation. Toutes ces propriétés proviennent directement de la modélisation des propriétés du système nerveux de l'homme. Les principales applications proposées, sont restées dans des dimensions modestes, et parmi les domaines d'applications, on cite :

- Reconnaissance de formes.
- Synthèse vocale.
- Analyse financière.
- Traitement de la parole.
- Diagnostic médical.
- Commande de processus.

Alors pour résoudre un problème il faut d'abord définir les objectifs et choisir le type de réseaux qui convient.

Chapitre 2

Réseaux de Neurones Artificiels.

Chapitre 2

Réseaux de neurones artificiels

I. Les neurones formels :

I. 1. Le neurone formel de base (L'automate à seuil) : [13], [14]

La première modélisation d'un neurone formel date de 1943. Elle a été présentée par *McCulloch et Pitts*, c'est un automate binaire qui réalise une somme pondérée de ses entrées, le *potentiel*, et compare ce potentiel à un seuil (nul) : s'il est supérieur, la sortie vaut +1 et le neurone est actif (émission d'un spike le long de l'axone), s'il est inférieur, la sortie vaut 0, et le neurone est inactif, ce qu'on peut formaliser en décrivant l'état (ou activité) par une variable binaire :

$$\begin{aligned} V &= 0, (\text{repos}) \\ V &= 1 \text{ emission d'un spike} \end{aligned}$$

L'état du neurone, dépend d'un certain ensemble de N signaux d'entrée ($V_j, j=1, N$). Chaque V_j est soit l'activité d'un autre neurone, soit une valeur importée de l'extérieure. Chaque lien entre une unité d'entrée j , et le neurone formel, représente un contact synaptique entre l'axone issu du neurone j et l'arbre dendrite du neurone considéré. Il est caractérisé par un nombre J_j , appelé poids synaptique, ce poids est positif ou négatif suivant que la synapse est excitatrice ou inhibitrice.

Le neurone calcule son potentiel S , défini comme la somme des signaux d'entrée pondérés par le poids correspondant :

$$S = \sum_j J_j V_j.$$

Si ce potentiel est plus grand qu'un certain seuil h^s , le neurone 'décharge' :

$$\begin{aligned} V &= 1 \text{ Si } S > h^s; \\ V &= 0 \text{ Si non.} \end{aligned}$$

I. 2. Le neurone formel à fonction d'activation [11], [13], [14]

Il existe aujourd'hui d'autres types de neurones, mieux adaptés aux tâches de traitement du signal, de classification, ou de commande, dont la sortie n'est pas le signe de leur potentiel,

mais une fonction f non linéaire dérivable de ce potentiel, cette fonction est dite *fonction d'activation* du neurone (voir Figure 6), on cite par exemples les suivantes :

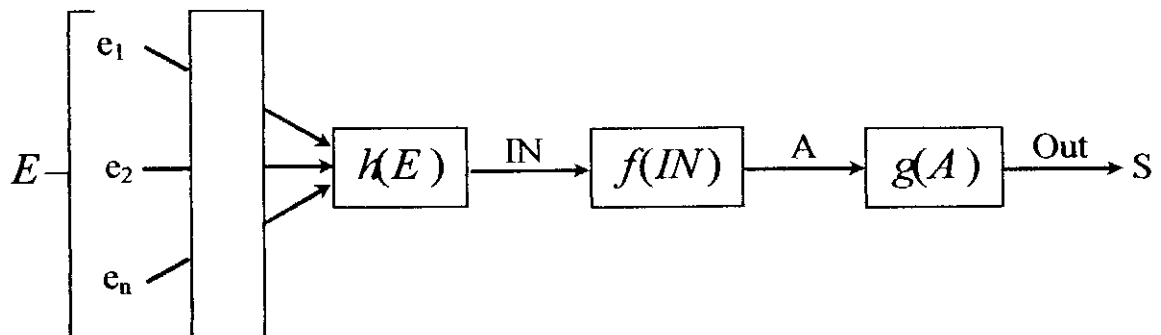
Fonctions à seuil : - Heaviside : $f(x) = \begin{cases} 1 & \text{si } (x) \geq 0 \\ 0 & \text{si } (x) < 0 \end{cases}$

Fonction -signe : $f(x) = \begin{cases} +1 & \text{si } (x) \geq 0 \\ -1 & \text{si } (x) < 0 \end{cases}$

Fonction sigmoïde : $f(x) = \frac{1}{1 + e^{-\theta x}}$

Les fonctions de type sigmoïde, sont introduites par vraisemblance biologique pour prendre en compte l'importance de l'activation du neurone.

La nature des entrées et des sorties, peut être soit binaire, soit réelle. Par extension, le neurone formel étendu est défini ainsi :



-Fig.6- Neurone formel étendu.

Avec :

$E = (e_1, \dots, e_i, \dots, e_n)$: Vecteur d'entrée du neurone

$C = (c_1, \dots, c_i, \dots, c_n)$: Vecteur des poids

$IN = h(E)$. Fonction d'entrée $A = f(IN)$: fonction d'activation

$S = OUT = g(A)$: Fonction de sortie du neurone (identité, ou toute fonction croissante et impaire).

1. 3. Le réseau de neurones : [14], [15]

Un réseau de neurones est un graphe orienté et pondéré. Les nœuds de ce graphe sont des automates simples, appelés neurones formels (ou unité connexionniste), et dotés d'un état interne, l'activation, par laquelle ils influencent les autres neurones du réseau. Cette activation se propage dans le graphe le long des arcs pondérés, appelés liens synaptiques ; par extension, on appelle poids synaptique, la pondération d'un lien synaptique. La règle qui détermine l'activation d'un neurone en fonction de l'influence de ses pairs, est appelée fonction d'activation.

L'état du réseau entier, est composé de l'activation de ses neurones constitutifs.

Si l'on suppose que les N neurones d'un réseau sont indexés, cet état global peut être donné sous la forme d'un vecteur d'activation, dont les composantes sont les valeurs des neurones individuels :

$$Z = (z_1, \dots, z_i, \dots, z_n).$$

De même, les liens synaptiques du réseau, sont utilement décrits, par la matrice des poids synaptiques, matrice carré dans laquelle s'inscrivent les poids des liens :

$$C = \begin{bmatrix} C_{11} & \dots & C_{1n} \\ \vdots & & \vdots \\ C_{ni} & \dots & C_{nm} \end{bmatrix}$$

La connectivité d'un réseau, déterminée lors de sa construction, représente les liens synaptiques que possède le neurone dans le réseau. Elle est rarement complète, en effet ; un neurone n'est relié, en général, qu'un sous ensemble des autres neurones. Les liens manquants sont alors, représentés dans la matrice des poids, par une entrée nulle. On nommera architecture (nombre et type des neurones, conductivité) la structure du réseau. Ainsi la structure du réseau est bien résumée par la matrice des poids.

Chaque neurone est un processeur non linéaire qui, à chaque instant discret k, calcule son potentiel $v_i(k)$ et son activité $Z_i(k)$ de la façon suivante :

$$z_i(k) = f_i(v_i(k)) \quad \text{ou} \quad v_i(k) = \sum_{j \in P_i} \sum_{\tau=0}^{q_{ij}} c_{ij,\tau} z_j(k - \tau)$$

P_i est l'ensemble des indices des unités du réseau propageant leur activité au neurone i . Son potentiel $v_i(k)$ est une somme des valeurs de ces unités, à l'instant k ou à des instants précédents, pondérée par les coefficients C_{ij} , q_{ij} est le retard maximal du neurone i sur l'entrée ou le neurone j . Si $q_{ij} = 0$ pour tout j , le neurone i est statique. La fonction f_i , fonction d'activation du neurone i , est en général non linéaire. Ce peut être la distribution de Heaviside, la fonction tangente hyperbolique ou une sigmoïde, une fonction à base radiale (RBF), ou encore la fonction identité.

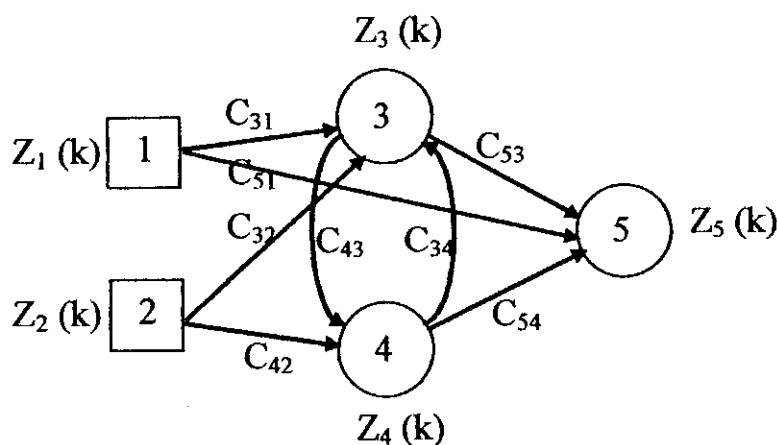


Fig.7. Exemple du réseau de neurones.

Un réseau de neurones est conçu, pour remplir une tâche que le concepteur définit, par une séquence d'entrées, et par une séquence de valeurs désirées correspondantes pour les activités de certains neurones du réseau, les neurones de sortie. Les neurones qui ne sont pas des neurones de sortie sont dits cachés. Le réseau de la figure 7 possède deux entrées, deux neurones cachés, et un neurone de sortie.

L'architecture d'un réseau, est définie par le graphe du réseau, les coefficients de celui-ci, et par les fonctions d'activation des neurones. Le caractère bouclé ou non du réseau, ainsi que les fonctions d'activation, peuvent être fixés en fonction de la tâche que doit remplir le réseau de neurones. Les valeurs des coefficients sont en général déterminées par apprentissage. Ainsi, dans le cas de la modélisation d'un processus, l'architecture peut être partiellement déterminée par des connaissances a priori ; les valeurs de coefficients ayant une signification physique peuvent être fixées préalablement à l'apprentissage. Nous distinguons deux types de réseaux :

I. 3.1. Les réseaux de neurones non bouclés: [15], [16]

Un réseau de neurones non bouclé, est un réseau dans lequel l'information circule des entrées vers les sorties sans bouclage. Son graphe ne possède pas de cycle, ainsi ses unités (entrées et neurones) sont ordonnées, et les connexions ne peuvent aller que d'une unité à un neurone dont l'indice est supérieur. C'est un système statique, utilisé principalement pour effectuer des tâches de classification, ou de modélisation statique de processus. Il réalise donc, de manière générale, une relation *algébrique* non linéaire entre ses entrées et ses sorties. Chaque neurone i du réseau calcule à l'instant k :

$$Z_i(k) = f_i(v_i(k)) \quad \text{avec} \quad v_i(k) = \sum_{j \in P_i} c_{ij} \cdot Z_j(k)$$

Où $v_i(k)$ est le potentiel du neurone i à l'instant k , $z_i(k)$ son activité, P_i est l'ensemble des indices des unités propageant leur activité au neurone i , et c_{ij} est le coefficient de la connexion reliant l'unité j au neurone i .

Un réseau non bouclé, réalise donc une transformation entrée/sortie non linéaire paramétrée par les coefficients C du réseau.

$$Y(k) = \psi_{RN}(I(k); C)$$

Où :

$Y(k) \in R^{NY}$ est le vecteur des sorties à l'instant k , c'est-à-dire des activités des neurones de sortie du réseau à l'instant k , et $I(k) \in R^{NI}$ est le vecteur des entrées externes à l'instant k .

$\psi_{RN}(I(k); C), R^{NI} \rightarrow R^{NY}$, représente la fonction réalisée par les neurones du réseau interconnectés avec les coefficients C .

L'architecture la plus générale d'un réseau statique, est celle du réseau à couches (voir figure 8). Les neurones cachés sont répartis en couches successives, les neurones appartenant à une couche donnée, n'étant commandés que par les neurones de la couche précédente, et ceux de la première couche, n'étant connectés qu'aux entrées externes. Mentionnons que la propriété d'approximation universelle des réseaux de neurones, est valable pour la famille des réseaux possédant, seulement une couche de neurones cachés.

La figure 8 illustre le schéma d'un réseau non bouclé, Ce réseau possède 8 entrées externes, 5 neurones cachés, et un neurone de sortie.

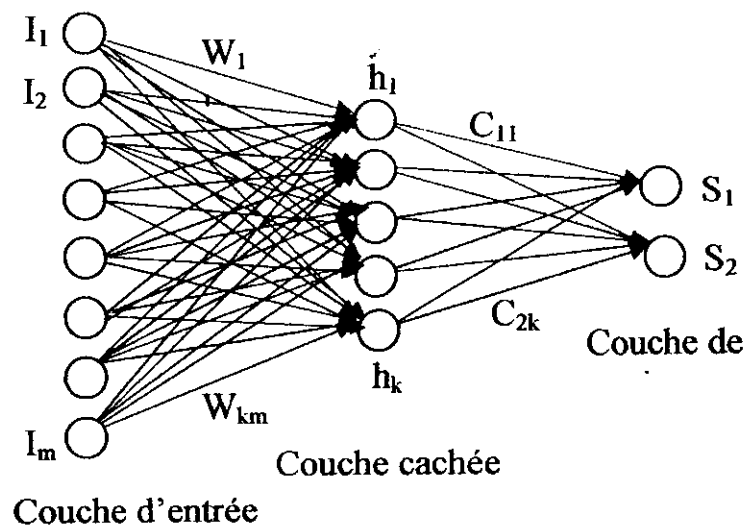


Fig.8 Réseau multi couches statique

I. 3.2. Les réseaux de neurones bouclés : [15], [16]

Un réseau est bouclé, si son graphe possède, au moins un cycle, c'est un système dynamique, utilisés comme filtres non linéaires, ainsi que pour la modélisation et la commande de processus. Tout réseau de neurones bouclé à temps discret d'ordre N_s , peut être représenté par un réseau, dont la dynamique est décrite par N_s équations aux différences couplées d'ordre 1, mettant en jeu, N_s variables d'état, et N_i entrées externes. Cette forme canonique n'est en général pas unique.

Le comportement dynamique d'un réseau de neurones bouclé, peut être décrit, par la représentation d'état paramétrée par les coefficients C :

$$\begin{aligned} S(k+1) &= \varphi_{RN}(S(k), I(k), C) \\ Y(k) &= \psi_{RN}(S(k), I(k), C) \end{aligned}$$

Où : $I(k) \in \mathbb{R}^{N_i}$ est le vecteur des entrées externes, $S(k) \in \mathbb{R}^{N_s}$ le vecteur des variables d'état, $Y(k) \in \mathbb{R}^{N_y}$ le vecteur des sorties, à l'instant k , et $S(k+1) \in \mathbb{R}^{N_s}$ est le vecteur des variables d'état à l'instant $k+1$. φ_{RN} et ψ_{RN} Représentent les fonctions réalisées par le réseau de neurones statiques de la forme canonique interconnectés avec les coefficients C .

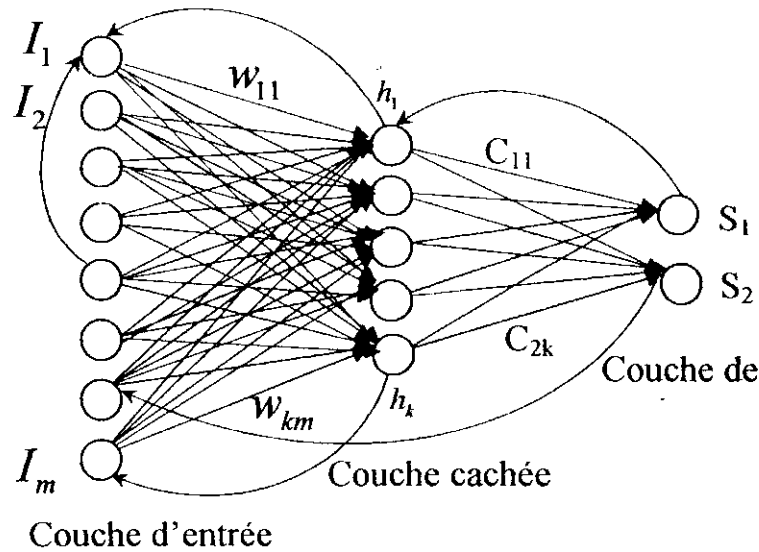


Fig.9 schéma d'un Réseau de neurones bouclé

Un réseau de neurones bouclé, est ainsi défini à partir d'un réseau non bouclé constitué de neurones statiques, possédant, N_i entrées externes, N_s entrées d'état (les variables d'état à l'instant k), N_c neurones cachés et N_y neurones de sortie (neurones pour les activités desquels il existe une valeur désirée). Les sorties du réseau à l'instant k , sont les activités des N_y neurones de sortie, et les variables d'état à l'instant $k+1$, sont les activités de N_s neurones que nous appelons neurones d'état. Ces neurones d'état, sont soit des neurones cachés, soit des neurones de sortie.

I. 4. Les réseaux de neurones, approximateurs universels : [12], [15], [16]

Les résultats qui présentent un intérêt pour la modélisation et la commande de processus, sont ceux qui concernent l'approximation de fonctions à valeurs continues. Nous laissons donc de côté, la possibilité de réaliser des fonctions booléennes, à l'aide de réseaux de neurones, démontrée anciennement par *McCulloch et Pitts*, ainsi que celle de réaliser une frontière de séparation, solution d'un problème de classification.

En effet, plusieurs approches théoriques, ont montré que les réseaux de neurones à 3 couches (deux couches de traitement effectifs), sont capable d'effectuer l'approximation de fonction continues, de R^p dans $[0, 1]$, ou de compacts de R^p dans $[0, 1]$ (ou plus généralement de R^p ou de compacts de R^p dans $[0, 1]^n$). Ces approches sont fondées sur

plusieurs théorèmes de mathématique, théorème de *Stone-Weierstrass*, théorème de *Kolmogorov*, théorème de *Cybenko*, et théorème de *Funabashi*.

De façon un peu simplifiée, on peut résumer les résultats des deux premiers théorèmes, de la manière suivante :

1. Les travaux fondés sur le théorème de *Kolmogorov* prévoient que toute fonction continue de $[0, 1]^p$ dans $[0, 1]^n$, peut être approchée, par un réseau multicouche, constitué d'une première couche (transmission) de p cellules, d'une seconde couche de $(2n + 1)$ cellules, et d'une couche de sortie de p cellules, chaque couche étant totalement connectée à la suivante. Hélas, ce théorème, s'il précise de nombre d'unités du réseau, ne prédit pas la nature des neurones, c'est-à-dire les fonctions non linéaires des neurones qui permettent cette approximation.
2. Les travaux issus du théorème de *Stone-Weierstrass* montrent qu'un réseau de 3 couches de neurones possédant une fonction non linéaire arbitraire, dite saturante ('squashing function'), dont la sigmoïde vérifie les propriétés, peut approximer, toute fonction continue de $[0, 1]^p$ dans R^n , pour vue qu'il y ait suffisamment d'unités dans le réseau.

II. Apprentissage des réseaux de neurones : [14], [16]

La plus part des méthodes d'apprentissage par réseaux de neurones, sont en fait, des méthodes d'estimation paramétriques, ou d'approximation de fonction.

Dans le cas de la modélisation par réseaux de neurones, d'un processus physique, il est tout à fait normal, de supposer que les sorties mesurées sur le processus, obéissent à des lois déterministes, et de chercher une expression mathématique des fonctions à approximer. La propriété d'approximation universelle, est donc une propriété nécessaire pour la modélisation par réseaux de neurones, mais n'est pas suffisante. En effet :

- ✓ d'une part, les bases d'apprentissage {couples entrées-sorties mesurées}, utilisées pour déterminer les fonctions inconnues, ont un ordre fini, et ne permettent pas, donc, de déterminer ces fonctions de façon univoque. le but de l'apprentissage est alors de trouver la solution la plus parcimonieuse, passant par tous les points d'apprentissage, qui, si

l'ensemble d'apprentissage est bien choisi, tendra vers les fonctions f et g , supposées régir le fonctionnement du processus.

- ✓ d'autre part, on est souvent, face à des processus physiques affectés de perturbations aléatoires ; dans ce cas, le but de l'apprentissage ne peut être de passer par tous les points de l'ensemble d'apprentissage, mais il doit ajuster les coefficients du réseau de façon que les fonctions qu'il réalise, tendent vers ces fonctions.

Dans le cas où la tâche du réseau est de réaliser une loi de commande, on fait, fréquemment, appel aux propriétés d'approximation universelle des réseaux de neurones, pour synthétiser automatiquement une fonction de commande.

Dans ce paragraphe, nous donnons les principes d'apprentissage neuronal, et décrivons la mise en œuvre des procédures et algorithmes d'apprentissage.

II. 1 Principe général :

L'architecture du réseau de neurones n'est souvent que partiellement imposée par la tâche à réaliser : les entrées et les sorties du réseau, peuvent être fixées en fonction de celle-ci par le concepteur, ainsi que le type et la connectivité des neurones (comme nous l'avons précisé au paragraphe précédent), mais le nombre de neurones ne peut être fixé a priori, et il est en général ajusté selon le succès de l'apprentissage.

Une fois l'architecture du réseau fixée, le but de l'apprentissage est l'estimation des poids synaptiques, pour remplir au mieux la tâche à laquelle le réseau est destiné.

On appelle apprentissage, l'opération par laquelle le réseau de neurones, acquiert la capacité de faire certaines tâches en modifiant ses paramètres internes, (connections) 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 présenté comme entrée; la paire (x_i, y_i) , représente un échantillon d'une fonction $f \in R^n$.

II. 2. La procédure d'apprentissage :

Les informations d'entraînement, sont codées sous forme d'un vecteur appelé patron d'entrée, qui est communiqué aux neurones d'entrée du réseau. La réponse du réseau est

représentée, par la valeur d'activation de ses neurones de sortie, dont le vecteur s'appelle patron de sortie. Lorsqu'il s'agit d'un apprentissage supervisé, on dispose aussi du comportement désiré que doit apprendre le réseau, exprimé sous forme de patron de référence.

En général l'apprentissage, se fait sur une période relativement longue, durant laquelle les patrons d'entrée (et éventuellement de sorties désirées), peuvent être présentés au réseau, un grand nombre de fois. Cet apprentissage comprend quatre étapes de calcul :

- 1) **initialisation des poids synaptiques du réseau** : étant donné que, dans le cas général, les poids du réseau à la fin de l'apprentissage, dépendent en partie de l'ensemble des poids, dont il disposait au départ, alors le choix exact de ces valeurs initiales, n'est pas critique pour le succès de l'apprentissage. Il est donc courant, sauf cas exceptionnels, d'initialiser les poids du réseau à des (petites) valeurs aléatoires au début de l'apprentissage.
- 2) **Présentation du patron d'entrée et propagation d'activation.**
- 3) **Calcul de l'erreur** : pour chaque neurone du réseau, une valeur d'erreur est calculée à partir de son activation et celles des neurones qui lui sont reliés.
- 4) **Calcul du vecteur de correction** : à partir des valeurs d'erreur, on détermine alors la correction à apporter aux poids synaptiques des connections. La correction des poids, peut se faire après chaque présentation de patron ; alternativement, les vecteurs de correction, peuvent être accumulés pendant un certain temps, avant d'être appliqué au réseau. Le nombre de patrons à présenter au réseau avant d'effectuer la correction, s'appelle **fenêtre de mise à jour ('batch size')**.

L'apprentissage d'un réseau de neurones, peut s'effectuer en le plaçant, directement en situation, ou il développerait automatiquement un comportement adéquat. Cet apprentissage, dit 'on-line', est relativement peu employé, en premier lieu, on évite les effets indésirables du système dus au désapprentissage du réseau. De plus, les règles d'apprentissages, souvent lentes, peuvent ralentir le réseau au point de le rendre inutile en pratique.

La deuxième approche d'apprentissage (qui est en général employée), est dite 'off line' ou différée. Dans ce cas, l'apprentissage s'effectue à part, sur un échantillon de patrons présélectionnés. Durant l'apprentissage, ces patrons seront chacun présenté plusieurs fois au réseau ; une présentation du corpus entier, s'appelle cycle ou époque d'apprentissage. Dans ce cas, il se pose le problème du choix du corpus : dans quelles mesures, les patrons du corpus

sont représentatifs de l'environnement réel, et à quel point, les performances du réseau se dégradent elles dans la situation réelle ? La capacité d'un réseau neuromimétique de réagir correctement face à des données non apprises, définit ses performances en généralisation. Etant, le corpus d'apprentissage ne représente qu'une fraction des cas possibles, les performances réelles du réseau sont fortement liées à ses performances en généralisation.

II. 3 La procédure de validation croisée :

La validation croisée ('cross validation'), est une autre technique d'entraînement neuronal. Elle permet de décider sur le succès de l'apprentissage d'un réseau de neurones, et de tester ses performances en généralisation. Ces tests, seront faits sur le corpus d'apprentissage, où les performances, en général, s'améliorent constamment. Néanmoins, cette mesure n'est pas suffisante, à elle seule, pour estimer les performances réelles du réseau, en effet, il est possible que le réseau, apprenne à traiter spécifiquement les informations contenues dans le corpus d'apprentissage, au détriment de ses performances sur le problème général, ce phénomène se nomme "sur apprentissage", ou apprentissage par cœur.

La mesure de l'évolution des performances sur un second corpus (le corpus test, ou corpus de généralisation), est nécessaire à fin d'éviter ce problème. L'amélioration des performances sur ce corpus, est un bon indicateur sur le succès de l'apprentissage, et que le réseau apprend effectivement à traiter sur le problème général. Dans l'autre cas, si le réseau commence à apprendre par cœur, ses performances en test, se dégradent sur ce corpus. Ainsi le moment judicieux, pour interrompre l'apprentissage, est celui où les performances en test sont optimales.

Un troisième corpus (le corpus de validation), est en fin, présenté au réseau, pour mieux tester ses performances de généralisation à la fin d'apprentissage.

La validation croisée procède donc en trois étapes :

- L'apprentissage même, pendant lequel on évalue le comportement du réseau sur le corpus d'apprentissage.
- Une étape de test, où l'apprentissage est suspendu périodiquement, et où le corpus test est présenté au réseau. Les performances ainsi obtenues, constituent une première indication des performances réelles du réseau.

- Une étape de validation à posteriori, un troisième corpus (dit de validation), est présenté au réseau à la fin de l'apprentissage. Les performances du réseau sur celui-ci, sont employées pour juger du succès ou du l'échec de l'apprentissage

II. 4. Types d'apprentissage :

Les techniques d'apprentissage connexionnistes, se répartissent en trois grandes familles :

- Apprentissage supervisé.
- Apprentissage non supervisé.
- Apprentissage semi supervisé.

II. 4.1 Apprentissage non supervisé : [11], [14]

Dans un apprentissage non supervisé, la règle d'apprentissage du réseau n'est pas fonction du comportement de la sortie du réseau, mais plutôt, du comportement local des neurones, et de l'inter corrélation entre neurones. Plusieurs algorithmes de ce type d'apprentissage, sont développés, on cite, entre autres, les deux algorithmes suivants ;

a - Règle de HEBB :

La première règle d'apprentissage par réseaux de neurones, fut non supervisée. En effet, vers la fin des années 40, *HEBB* propose l'idée que le cerveau s'adapte à son environnement en modifiant l'efficacité des connexions entre neurones. Cette modification, dépend de l'activité simultanée des neurones reliés par la connexion. Son application pour les réseaux de neurones formels, est la suivante :

On considère que si deux neurones connectés entres eux, sont activés au même moment, la connexion qui les relie doit être renforcée. Dans les cas contraires, elle n'est pas modifiée. La règle de *HEBB* stimule, donc, qu'une synapse modifie son efficacité, seulement quand ses deux neurones sont actifs. Cette modification, qu'il décrit, a été reformulée mathématiquement de la façon suivante :

Supposons qu'entre les instants ' t ' et ' $t + \partial t$ ', on ait soumis le réseau à un stimulus, et que ce stimulus ait provoqué l'activation A_i du neurone i , et A_j du neurone j , alors, on aura ;

$$w_{ij}(t + \partial t) = w_{ij}(t) + \mu.A_i.A_j$$

Où

μ ($0 < \mu < 1$) : est une constante quelconque.

w_{ij} : représente le poids synaptique entre les neurones i et j .

A_i, A_j : représente la corrélation entre l'activation des neurones i et j du réseau.

Appliquée au réseau neuromimétique, cette règle simple pousse les neurones, à corréler leurs activations avec les valeurs des patrons d'entrée.

b - Cartes auto-adaptatives de KOHONEN :

Le neurone formel de *KOHONEN*, est similaire aux modèles déjà présentés. Son activation, que nous notons S , ne prend que des valeurs positives. Dans ce cas, la règle de *HEEB*, ne permet pas de réduire la valeur des poids des connexions, quand l'un des deux neurones est inactif. *KOHONEN* propose donc, l'équation suivante comme règle d'apprentissage :

$$\Delta w_i = k \cdot S \cdot e_i - B(S) \cdot w_i, \text{ en général on pose } B(s) = k \cdot S$$

Avec $B(s) \geq 0$ une fonction d'oubli, et généralement $k \geq 0$, e_i entrée n° i et S la sortie.

La règle proposée par *KOHONEN*, se décompose en deux étapes :

- Premièrement, il y a sélection du neurone qui correspond à un type de signal d'entrée donné, autrement dit, chaque cellule du réseau doit correspondre à une entrée spécifique.
- Deuxièmement, il y a augmentation de l'activation de ce neurone, et du groupe de neurones qui l'entoure, lorsque ce signal d'entrée est présenté.

KOHONEN, exprime alors une idée très intéressante ; si l'on veut sélectionner un neurone qui correspond à un vecteur d'entrée donné, pourquoi alors ne pas comparer les vecteurs des poids des connexions, avec le vecteur d'entrée ? Ainsi, lors de la présentation d'un signal E , le réseau, doit trouver le neurone c tel que $\|E - M_c\| = \min(\|E - M_i\|)$ avec $1 < i < n$ (n est le nombre de neurones). C'est-à-dire le neurone c , tel que son vecteur de poids M_c , soit le plus proche du vecteur entré E . De fait, ce type de modèle, est souvent appelé réseau à réponse localisée.

II. 4.2. Apprentissage supervisé : [11], [14]

Dans ce cas, le but de l'apprentissage, est d'inculquer un certain comportement de référence au réseau. On associe alors, à chaque patron d'entrée I' une sortie désirée D' , qui spécifie les valeurs de sortie désirée, l'apprentissage se déroule de la façon suivante : un patron est présenté aux neurones d'entrée, puis l'activation se propage à travers le réseau. La réponse des neurones de sortie, est alors comparée aux valeurs désirées ; ceci détermine l'erreur du réseau E' pour le patron donné :

$$E' = (e_{i1}, e_{i2}, \dots, e_{is})$$

L'on procède alors, à une modification des poids, qui vise à réduire l'erreur ainsi calculée en répartissant l'erreur sur chaque poids du réseau, en fonction de la part qu'il a joué dans la production de la réponse erronée. L'apprentissage supervisé, peut se reformulé, plus précisément, comme la recherche des poids qui minimise l'erreur de sortie pour l'ensemble des patrons. Cette recherche, s'effectue, généralement, par une série d'approximations successives, ou l'ensemble des patrons est présenté plusieurs fois au réseau, et ou les poids synaptiques sont modifiées de façon à améliorer graduellement le comportement global du réseau :

$$D_w = F_w(W, E, n).$$

Où

D_w est la correction à apporter aux poids du réseau.

F_w est une fonction qui calcule la modification à apporter aux poids W , en fonction de l'erreur E , et d'une constante n (souvent appelée pas d'apprentissage) ; en général, les règles d'apprentissage supervisé, sont des formes de descente de gradient. Différents algorithmes sont développés, dont les plus connus sont :

a- règle du perceptron :

L'apprentissage du perceptron, est un apprentissage supervisé qui se fait par correction d'erreur.

Supposons que l'on présente une entrée F à l'entrée du réseau, et que l'on désire avoir la réponse d à la cellule de sortie, la réponse obtenue effectivement, est notée S . Avec ces notations, la règle du perceptron est alors :

$w_i \rightarrow w_i + k(d - S).e_i$, Avec $k > 0$.

Si $d = S$, les poids ne sont pas modifiés.

Si $d - S = 1$, w_i est transformé en $w_i + k.e_i$.

Cette règle, est le plus simple algorithme d'apprentissage supervisé.

b- Algorithme de rétro propagation :

Présentation :

Appelé couramment "back propagation", cet algorithme, n'est en fait, qu'une généralisation de la règle de *Widrow Hoff* pour un réseau multi couches. Sa mise au point a été faite, par (*Fogelman souli, Gallinari, Le cun*) en France, et (*Rumelhart, Hinton, Williams*) aux Etats Unis. Cet algorithme, se base sur l'utilisation d'une fonction dérivable (fonction sigmoïde par exemple), en remplacement de la fonction seuil utilisée dans les neurones linéaires à seuil. Dans cet algorithme, qui permet de propager l'activation provenant des cellules d'entrées vers la couche de sortie, on peut aussi, en suivant le chemin inverse, rétro propager l'erreur commise en sortie, vers les couches internes.

Apprentissage :

Pour l'apprentissage du réseau, on suit le même principe que la règle de *Hoff Widrow*. On procède donc, en deux phases :

- Phase de propagation ; dans cette étape, pour chaque exemple présenté en entrée du réseau, une sortie réelle est calculée par chaque neurone. Ce calcul est effectué de proche en proche de la couche d'entrée à la couche de sortie.
- Phase de rétro propagation ; Dans cette étape, on calcule l'erreur sur chaque cellule de la couche de sortie. Cette erreur est ensuite rétro propagée dans le réseau, en allant de la sortie du réseau vers son entrée, donnant lieu, alors, à une modification de chaque poids. Ce processus est répété, en présentant successivement tout les exemples, jusqu' à ce que l'erreur, devienne inférieure à un seuil choisi, on dit alors que le réseau a convergé. L'apprentissage s'effectue donc, en minimisant l'erreur quadratique, commise sur l'ensemble des exemples, par une approximation d'une descente du gradient.

Formalisation :

Pour un exemple à apprendre donné, on note X le vecteur des entrées, et Y le vecteur des sorties désirées. Si le réseau comporte n neurones en entrée, et m en sortie, on a donc :

$$X = (x_1, x_2, \dots, x_n) \text{ et } Y = (y_1, y_2, \dots, y_m)$$

On note $S = (s_1, s_2, \dots, s_m)$ le vecteur des sorties effectives du réseau. On cherche à minimiser l'erreur quadratique, entre les sorties désirées et les sorties obtenues, cette erreur étant considérée comme une fonction des poids des connections :

$$E(w) = \sum_{i=1, m} (y_i - s_i). \quad (2.1)$$

La règle de modification des poids à la présentation numéro k de l'exemple X est :

$$w_{ij}(k) = w_{ij}(k-1) - e(k) \cdot d_i \cdot O_j \quad (2.2)$$

Où

$$(1) \quad d_i = 2 \cdot (s_i - y_i) \cdot f(I_i), \text{ pour la couche de sortie.} \quad (2.3)$$

$$(2) \quad d_i = M_h \cdot w_{hi} \cdot \dot{f}(I_i) \text{ pour les couches cachées.} \quad (2.4)$$

Notation :

f . est la fonction d'activation d'un neurone, \dot{f} . est sa dérivée.

O_j . est la sortie du neurone j .

I_i . est l'entrée du neurone i . $I_i = \sum w_{ij} \cdot O_j$.

$E(k)$. est le pas du gradient à l'étape k .

Règle de Widrow Hoff (règle de modification des poids dans la rétro propagation):

La règle de *Widrow Hoff*, proposée en 1960, est intéressante à plusieurs titres, par exemple pour le problème de minimisation quadratique.

On considère un réseau de n neurones d'entrée, et m neurones de sortie, avec un nombre quelconque de couches cachées.

On suppose qu'un neurone, n'est connecté qu'à des neurones plus en aval, dans le sens entrée-sortie, mais pas forcément à des neurones situés sur la couche immédiatement suivante.

L'erreur commise sur un passage complet de l'ensemble des exemples est :

$$E(w) = \sum_k E^k(w). \quad (2.5)$$

Où l'on a noté $E^k(w)$, l'erreur commise sur un exemple k donné.

$$E^k(w) = (S^k - Y^k)^2 = \sum_{i=1,m} (s_i^k - y_i^k)^2. \quad (2.6)$$

L'algorithme de rétro propagation, consiste à effectuer une descente de gradient sur E. On effectue, donc une modification sur chaque poids après chaque passage d'un exemple k, suivant la formule :

$$w_{ij}(k) = w_{ij}(k-1) - \epsilon(k) \cdot \frac{\partial E^k}{\partial w_{ij}}. \quad (2.7)$$

Il faut donc, calculer pour tous les poids w_{ij} , le gradient de E^k :

$$\frac{\partial E^k}{\partial w_{ij}} = \frac{\partial E^k}{\partial I_i} \cdot \frac{\partial I_i}{\partial w_{ij}}. \quad (2.8)$$

$$\text{Or : } \frac{\partial I_i}{\partial w_{ij}} = \frac{\partial(\sum_{i,p} w_{i,p} \cdot O_p)}{\partial w_{ij}} = O_j. \quad (2.9)$$

En effet, l'indice p portant sur des neurones appartenant à la couche précédant celle de i, les sorties O_p de ces neurones ne dépendent pas de w_{ij} .

On obtient donc :

$$\frac{\partial E^k}{\partial w_{ij}} = \frac{\partial E^k}{\partial I_i} \cdot O_j.$$

$$\text{Ou, si l'on note } d_i = \frac{\partial E^k}{\partial I_i}. \quad (2.10)$$

$$\text{Alors : } \frac{\partial E^k}{\partial w_{ij}} = d_i \cdot O_j. \quad (2.11)$$

Et donc :

$$w_{ij}(k) = w_{ij}(k-1) - e(k).d_i.O_j$$

Pour un neurone i de la couche de sortie, on a :

$$d_i = \frac{\partial(\sum_j (s_j^k - y_j^k)^2)}{\partial I_j} = 2.(s_j^k - y_j^k). \frac{\partial s_j^k}{\partial I_j} \quad (2.12)$$

Car seul s_i^k dépend de I_i . Comme de plus $s_i^k = f(I_i)$, il vient :

$$d_i = 2.(s_i^k - y_i^k). \dot{f}(I_i) \quad (2.13)$$

Pour les neurones des couches cachées, on a :

$$d_i = \sum_h \frac{\partial E^k}{\partial I_h} \cdot \frac{\partial I_h}{\partial I_i} = \sum_h d_h \cdot \frac{\partial I_h}{\partial I_i} \quad (2.14)$$

Où l'indice h, porte sur les neurones sur lesquels le neurone i envoie des connexions. En effet, les entrées I_h des autres neurones ne dépendent pas de I_i , on a donc :

$$d_i = \sum_h d_h \cdot \frac{\partial I_h}{\partial O_i} \cdot \frac{\partial O_i}{\partial I_i} \quad (2.15)$$

Or

$$\frac{\partial I_h}{\partial O_i} = \frac{\partial(\sum_p w_{hp}.O_p)}{\partial O_p} = w_{hi} \quad (2.16)$$

Car p porte sur les neurones envoyant une connexion vers h, ces neurones situés sur la même couche que i et donc les O_p ne dépendent pas de O_i (pour $p \neq i$).

Comme de plus $O_i = f(I_i)$, on obtient finalement :

$$d_i = \sum_h d_h \cdot w_{hi} \cdot f(I_i) \quad (2.17)$$

La règle de modification des poids à la présentation numéro k de l'exemple X est :

$$w_{ij}(k) = w_{ij}(k-1) - e(k).d_i.O_j$$

Où l'on a :

$$(1) \quad d_i = 2.(s_i - y_i).f(I_i). \text{ pour la couche de sortie.}$$

$$(2) \quad d_i = \sum_h d_h \cdot w_{hi} \cdot f'(I_i). \text{ pour les couches cachées.}$$

Où h porte sur les neurones vers lesquels i envoie une connexion.

II. 4.3. Apprentissage semi supervisé :

Les techniques d'apprentissage semi supervisé (appelé 'renforcement learning'), sont utilisées lorsqu'on ne dispose pas d'un comportement de référence précis, mais d'une évaluation qualitative du comportement du système (par exemple, correct ou incorrect). Cette technique, est étudiée par *G.Samuel*, qui développa, en 1959, un réseau pour jouer aux échecs. Ce jeu, est un problème typique d'apprentissage supervisé, vu que la seule information, dont dispose le système pour corriger l'ensemble des coups qui constituent son comportement, est l'évaluation finale de la partie. Le réseau tient alors, de comprendre les coups qui avaient mené à l'échec, ou au succès des parties qu'il jouait, en employant des fonctions d'évaluation, pour choisir parmi les différents coups possibles. Ces fonctions, comportaient un ensemble de paramètres qui étaient affinés par apprentissage, en fonction des parties gagnées et perdues.

II. 5. Problèmes d'apprentissage : [11], [14]

Les techniques d'apprentissage par réseaux de neurones, bien qu'ils aient prouvé leur efficacité pratique dans de nombreux problèmes, on leur a reproché certaines insuffisances :

Insuffisance de la base d'apprentissage : la question se posant ici, est à quel point, cette base est représentative de la situation réelle, et à quel point les performances du réseau se dégradent elles dans la situation réelle ?

Minima locaux : la majorité des apprentissages supervisés, procèdent par descente du gradient. Or ce type de techniques, peut conduire à une solution sous optimale. En pratique, ce type de problèmes, est particulièrement difficile à éviter. Sachant que les poids initiaux du réseau, conditionnent en partie les résultats de l'apprentissage, plusieurs chercheurs lancent l'apprentissage, à partir de choix des poids initiaux défient, pour se préserver de ce problème.

Mauvais choix des paramètres : le comportement du réseau, est gouverné par un ensemble de paramètres architecturaux, pour lesquels, un mauvais choix de valeurs peut compromettre

l'apprentissage. Parmi ceux-ci, le nombre de neurones cachés du réseau. Cette valeur est particulièrement importante, par c'elle détermine le nombre de variables libres du réseau, et par c'elle détermine la capacité du calcul du réseau. Ainsi un nombre insuffisant de paramètres cachés, peut compromettre la capacité du réseau à résoudre la tâche donnée. Inversement, trop de neurones cachés, permettent au réseau d'apprendre les particularités du corpus d'apprentissage, au détriment des performances de généralisation.

Sur-apprentissage : quand l'apprentissage d'un réseau est prolongé, ou le réseau est surdimensionné, son choix de poids, reflète trop près les particularités du corpus d'apprentissage, au détriment de la tâche réelle, ce genre de problèmes peut être évité, par exemple, par l'emploi de la validation croisée.

Conclusion :

Ce chapitre est un aperçu sur les réseaux de neurones ; en effet nous avons cité tout d'abord les différents types de réseaux existant dans la littérature neuronale dans l'ordre chronologiques de leur apparition, présenté leurs caractéristiques de base, et précisé le principe fondamental de leur fonctionnement (approximation de fonction).

Nous avons, ensuite, présenté la procédure d'apprentissage neuronal, cité relativement en détail, les différents type et algorithmes d'apprentissage neuronal. Nous avons défini les critères du choix pour chaque algorithme' et en fin, donné quelques problèmes reconnus dans la réalité

La caractéristique de l'approximation universelle, fait des réseaux de neurones de bons candidats, dans les domaines de la modélisation et de la commande des processus. Leur utilisation pour la conduite des processus sera détaillée ultérieurement.

Chapitre 3

*Réseaux de Neurones
et commande des
processus*

Chapitre 3

I. Réseaux de neurones et la commande des processus:

I. 1. Le processus à commander : [15], [16]

Un processus de commande, est tout système physique évoluant dans le temps, sous l'effet des grandeurs internes, appelées variables d'états, et des grandeurs externes appelées variables d'entrée. Les variables d'entrée, sont de deux types : les commandes, sur lesquelles on peut agir, et les perturbations, sur lesquelles on n'a pas d'action. Parmi ces dernières, on distingue les perturbations mesurées et les perturbations non mesurées.

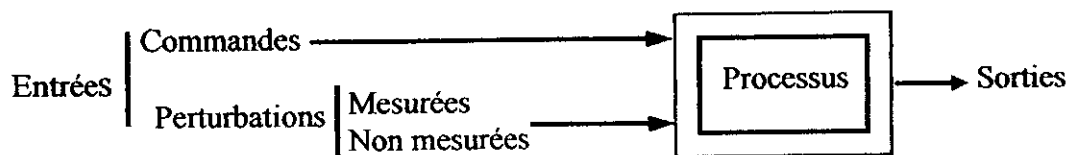


Fig. 10.

Notre processus appartient à la famille des processus dynamiques, c'est-à-dire des processus dont la valeur des sorties à un instant donné, ne dépend pas uniquement des valeurs de ses entrées à ce même instant, mais aussi, de leurs valeurs passées. Ces processus, seront représentés par des modèles mathématiques, qu'on supposera décrivant convenablement le système réel. De tels modèles, sont caractérisés par des variables d'état, qui constituent l'information minimale nécessaire au calcul de l'évolution des sorties, si toutes les entrées sont connues.

La mise au point d'un système de commande, veut dire déterminer les commandes à appliquer au système, de manière à assurer aux variables d'état et aux sorties qui nous intéressent, un comportement désiré. Pour se faire, on conçoit un *organe de commande* ; la mise en boucle du processus et de son organe de commande constitue *le système de commande*.

I. 2. Conception d'un organe de commande : [15], [16]

Plusieurs étapes sont nécessaires pour la conception d'un organe de commande

a. Modélisation du processus :

La modélisation du processus, est une étape nécessaire pour la synthèse de l'organe de commande. Modéliser, consiste à rassembler les connaissances disponibles du comportement dynamique du processus, par une application des lois physique sur les phénomènes mis en jeu, et par une analyse des données expérimentales. Ces analyses, conduisent à la définition des grandeurs caractérisant le processus, c'est-à-dire ses entrées, ses variables d'état, ses sorties, et aussi les perturbations, mesurables ou non, auxquelles il est soumis. En général, la démarche de modélisation conduit à plusieurs modèles hypothèses concurrentes.

b. Identification du processus physique :

Le but de cette étape, est de configurer, et sélectionner le meilleur modèle parmi différents modèles de la structure de modèles hypothèse, sur la base d'un critère de performances. Dans le cas où le but de cette démarche est la conception d'un organe de commande, alors le meilleur modèle, est celui qui conduit aux meilleures performances du système de commande, au sens du cahier des charges. Le meilleur modèle, est défini comme celui dont l'erreur de prédiction est la plus faible. L'estimation des paramètres d'un modèle, est donc, effectuée de manière à minimiser l'erreur de prédiction, à partir de mesures effectuées sur le processus (ensemble d'apprentissage)

c. Conception de l'organe de commande :

Le choix de l'architecture de l'organe de commande, sera fait, dans cette étape, en fonction du modèle du processus, et du cahier des charges, qui spécifie les performances désirées pour le système de commande en régulation et en poursuite. Le calcul de la commande à appliquer au processus, à partir de la consigne et de l'état du processus, se fait par un élément essentiel dans la structure de l'organe de commande, appelé correcteur. Un modèle de référence, un observateur, ou encore un " modèle interne ", sont d'autres éléments pouvant existés dans la structure de commande. La structure du correcteur est choisie en fonction de la tâche, définie par le cahier des charges, que doit remplir le système de

commande, elle peut avoir la structure d'un PID, ou celle d'un correcteur par retour d'état linéaire ou non... etc.

d. Estimation des paramètres du correcteur.

La configuration du correcteur, après avoir choisir un modèle convenable, pour qu'il remplisse au mieux la tâche définie dans le cahier des charges, est le but de cette étape. Etant donné que, la famille des modèles utilisée dans notre cas, est la famille des réseaux de neurones, alors l'estimation des coefficients du correcteur, correspond à la phase d'apprentissage du réseau de neurones. En réalité, la conception d'un organe de commande, est une procédure itérative, surtout lorsque celui-ci n'est pas adaptatif. Car un réseau modèle ne peut être définitivement rejeté ou validé, qu'en fonction des performances du système de commande élaboré à partir de ce modèle.

I. 3. Techniques de commande par RNA : [14], [17]

On parle ici des quelques techniques de commandes par RNA. On commencera par la commande supervisée, on parlera ensuite de la commande inverse, qui sera réalisée grâce à quatre techniques ; apprentissage indirect, apprentissage généralisé, apprentissage spécialisé, ainsi de la commande adaptative avec modèle de référence.

I. 3.1. Commande supervisée :

Cette approche, appelée aussi par imitation, ou parfois "copying an existing control", consiste, comme son nom l'indique, à imiter un régulateur existant (qui pourrait être un opérateur humain) en utilisant les propriétés d'approximation et d'apprentissage sur la base d'exemples de réseau de neurones. Les sorties désirées du réseau pour une entrée donnée, sont les valeurs de la commande générées par le contrôleur existant (Fig.11).

La question concernant l'utilité de cette méthode, possède plusieurs réponses ; premièrement, le régulateur peut être un élément impraticable lors de l'utilisation (par exemple, un être humain), deuxièmement, le réseau de neurone étant un approximateur de fonctions, on peut donc, avoir une forme explicite (sous forme d'équation mathématiques) du régulateur. En plus, les RNA possédant la capacité de calcul parallèle et distribué, permettra une meilleur implantation du contrôleur existant.

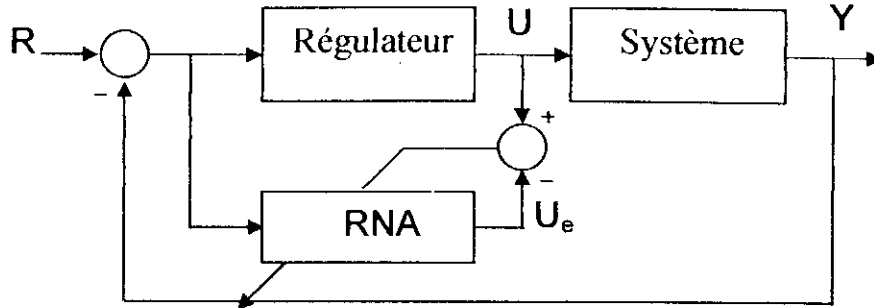


Fig.11. Schéma de la commande supervisée.

I. 3.2. Commande inverse :

L'idée de cette technique, se base sur la détermination du modèle inverse du système à commander. La mise en cascade du système et de son modèle inverse, donne lieu à un système équivalent, identité entre l'entrée et la sortie du système (Fig.12).

Sur la figure 12, les sorties désirées, sont notées par y_d , et les sorties réelles du système, sont notées par y .

Le régulateur neuronal, doit approximer l'inverse du système, et produira, ainsi, à partir de la réponse désirée y_d le signal de commande U , afin que y soit proche de y_d .

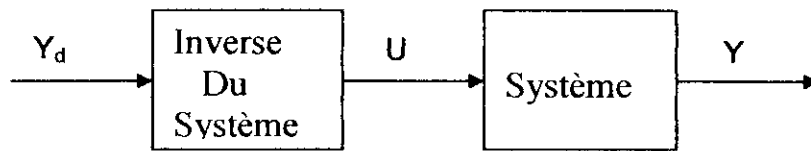


Fig. 12. Utilisation d'un réseau de neurone pour la commande inverse.

Pour approximer l'inverse du système, plusieurs approches d'entraînement du réseau de neurone, existent :

a. Méthode d'apprentissage indirect :

La première approche, consiste simplement, à entraîner le réseau de neurones par l'adaptation de ses poids, afin de minimiser l'erreur, $e_1 = u - t$, en utilisant l'architecture

présentée, sur la figure 13. Dans ce cas la minimisation de $e = y_d - y$ entraîne la minimisation de e_1 .

L'avantage de cette méthode, est qu'elle permet d'entraîner le réseau uniquement dans le domaine de fonctionnement désiré, puisqu'on commande par la sortie désirée y_d , et tout les autres signaux, sont générés à partir d'elle. Notons qu'il est avantageux, d'adapter les poids pour minimiser l'erreur $e = y_d - y$ à la sortie du système.

Malheureusement, cette méthode, n'est pas tous jours une procédure valide, parce que minimiser e_1 , ne revient pas nécessairement à minimiser e , il existe des systèmes, où le réseau de neurones tend à converger vers une solution qui fait correspondre à toutes les entrées, une valeur unique $u = u_0$, pour laquelle $e_1 = u - t$ est nulle, mais évidemment e ne l'est pas.

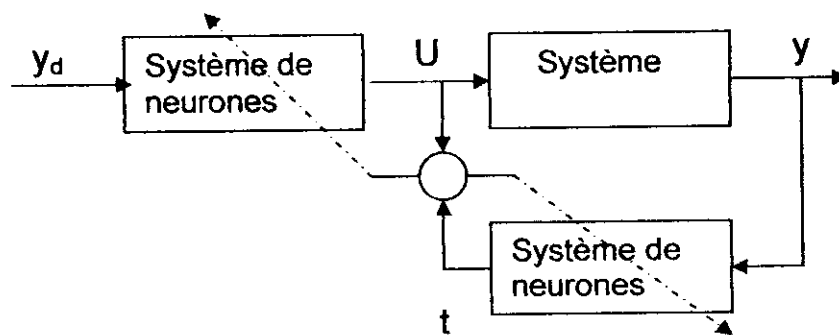


Fig. 13. Apprentissage indirect.

b. Méthode d'apprentissage généralisé :

La deuxième approche, appelée ' apprentissage généralisé ', fournit une méthode d'entraînement des réseaux de neurones qui minimise l'erreur e^2 . La procédure d'entraînement est la suivante :

Dans un premier temps, on cherche à découvrir un modèle inverse du processus, en utilisant la sortie $y(t)$ du processus (résultant d'une entrée $u(t)$ donnée), comme entrée d'un réseau neuronal ; ce réseau est entraîné par rétro propagation, à fin de produire une sortie la plus proche de $u(t)$. Dans un second temps, le réseau est placé devant le processus pour le conduire en boucle ouverte. Le réseau de neurones entraîné, doit pouvoir produire les valeurs appropriées de u pour un signal de référence $y_d(t)$ donné. Cette méthode ne sera efficace que

si la référence $y_d(t)$ est suffisamment proche de l'un des signaux d'entraînement (i.e des $y(t)$ utilisées).

On en voit directement les reproches de cette méthode : l'apprentissage se fait hors ligne, les $u(t)$ choisies pendant l'apprentissage ne conviendront peut être pas du tout pour l'objectif recherché, alors, le succès de cette méthode est lié à l'aptitude des RNA de généraliser aux cas non appris, et de répondre correctement dans le domaine de fonctionnement désiré. L'élargissement de la base d'apprentissage dans l'espace des entrées du système, peut aider à cette fin. Dans ce cas, la méthode d'apprentissage généralisé ne peut être, tous jours, efficace puisque le réseau est entraîné pour apprendre les réponses du système sur un espace trop large plus que le nécessaire. Les deux étapes de cette méthode sont fusionnées sur le schéma ci-dessous :

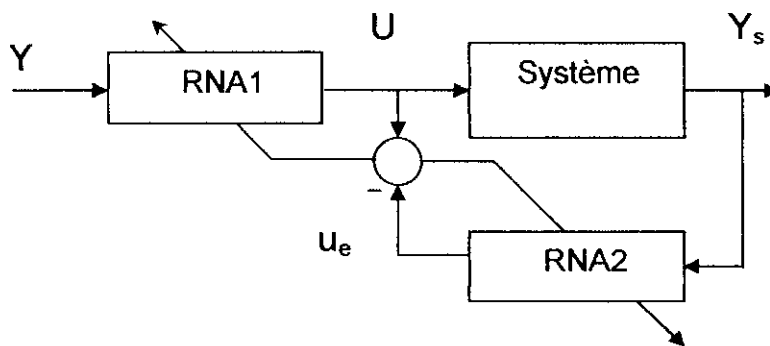


Fig. 14. Apprentissage généralisé.

C. Méthode d'apprentissage spécialisé :

Cette méthode présente une architecture pour entraîner un réseau de neurones, pour opérer proprement dans le domaine de fonctionnement seulement.

Dans cette approche, on utilise la réponse désirée y_d comme entrée pour l'apprentissage du réseau. Le réseau est entraîné pour trouver l'entrée du système, $u(t)$, qui conduit la sortie du système, $y(t)$, aux valeurs désirées $y_d(t)$. Cette opération est accomplie par l'utilisation de l'erreur entre la réponse désirée et les sorties réelles du système pour ajuster les poids du réseau de neurones par l'algorithme de 'back propagation'. A chaque itération, les poids du réseau sont ajustés pour minimiser l'erreur. L'avantage de cette méthode est qu'elle permet d'entraîner le réseau dans le domaine de fonctionnement

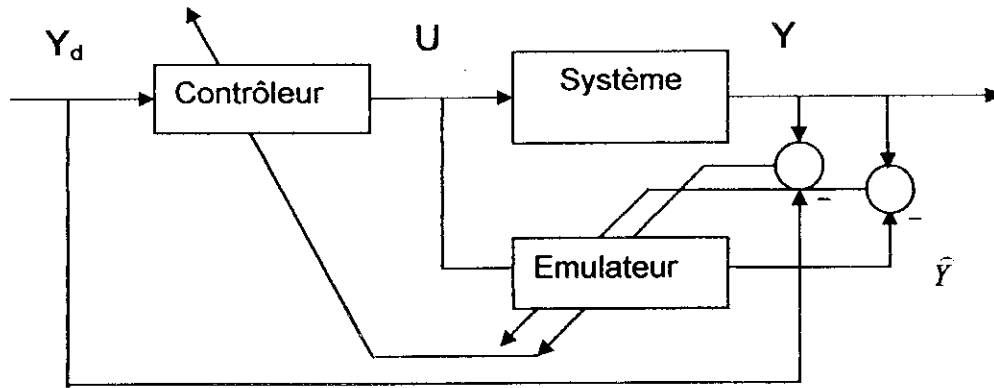


Fig. 15. Apprentissage spécialisé.

I. 3.3. Commande adaptative :

Par fois les paramètres physique du système, varient continuellement dans le temps, dans ce cas, la lois de commande synthétisée, sur la base d'une valeur donnée du vecteur des paramètres, peut ne pas fonctionner dans la pratique, on doit, donc, adapter la lois de commande, à chaque instant. C'est ce qu'on désigne par la commande adaptative.

Durant les vingt dernières années, deux approches ont été utilisées en commande adaptative. La première est la commande directe ("direct control"), la deuxième est la commande indirecte ("indirect control").

a. commande adaptative directe :

En commande adaptative directe, les paramètres du régulateur, sont ajustés directement de telle façon, à minimiser une certaine norme de l'erreur de sortie. Dans ce cas, les méthodes d'ajustement des paramètres du régulateur, sont basées sur la mesure de l'erreur de sortie.

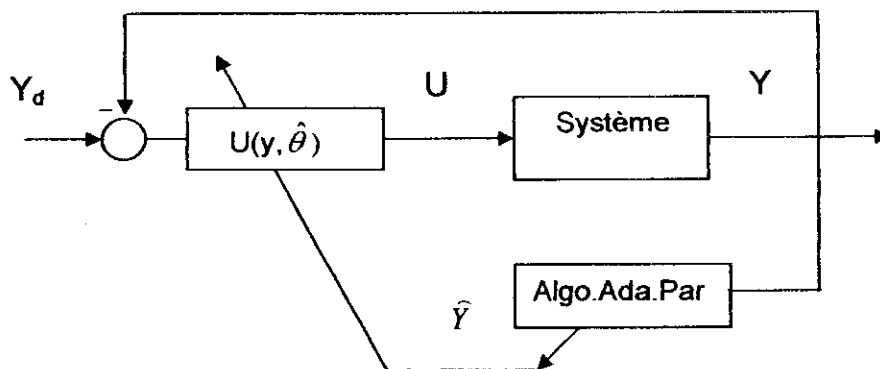


Fig. 16. Commande adaptative indirecte.

b. Commande adaptative indirecte :

En commande adaptative indirecte, les paramètres du système, sont estimés à chaque instant, et les paramètres du régulateur, sont calculés en supposant que les paramètres du modèle, sont les mêmes que ceux du système.

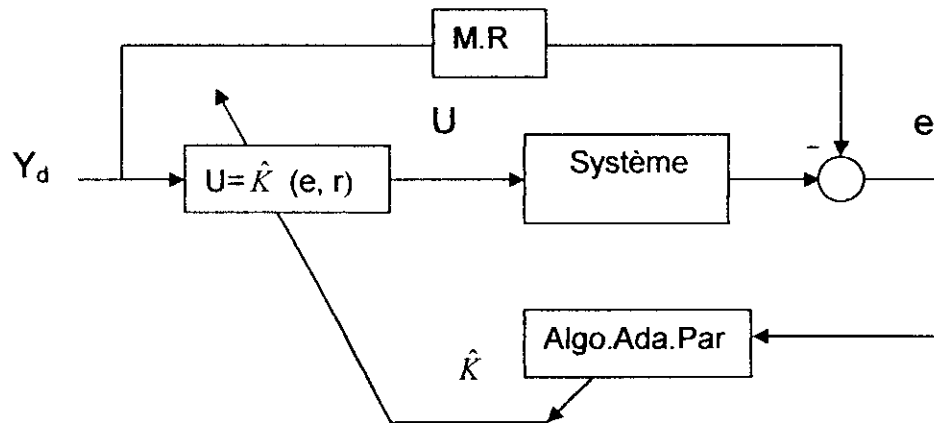


Fig. 17. Commande adaptative directe.

Quand la commande indirecte, est utilisée pour la commande des systèmes non linéaires, le système est paramétré en utilisant l'un des modèles décrits précédemment, les paramètres du modèle sont ajustés, par l'erreur d'identification et les paramètres du régulateur sont ajustés par la rétro propagation de l'erreur entre les sorties du modèle de référence et les sorties du modèle.

Les paramètres du régulateur, et ceux du modèle d'identification, peuvent être ajustés à chaque instant ou après un intervalle du temps. Dans le cas de la présence des bruits, il est judicieux, d'ajuster les paramètres du régulateur avec une cadence moins rapide que celle d'ajustement des poids du modèle d'identification.

Conclusion :

Dans ce chapitre, nous avons décrit le processus de commande de processus, en définissant, les étapes de construction d'un système de commande et ses éléments constitutifs. Nous avons, aussi, décrit, dans des aspects trop généraux, quelques méthodes de la commande des processus par réseaux de neurones. Etant donné que, les réseaux de neurones, sont classés comme des approximateurs universels, alors leur utilisation dans le domaine de la conduite des systèmes, notamment les commandes supervisée et inverse qui se basent complètement sur l'approximation de fonctions, est trop favorable.

Chapitre 4

*Application des
commandes et
simulation*

Chapitre 4

Application des commandes et simulations

I. Critères de performance pour les suspensions

Introduction :

A fin de déterminer la qualité d'une suspension, il est nécessaire de définir des critères à satisfaire. Nous fixons, donc, pour chaque modèle de suspensions les critères qui seront utilisés pour analyser les performances des suspensions.

La première démarche, consiste à vérifier la stabilité nominale du système. En sachant que la suspension est un système stable en boucle ouverte (c a d la suspension passive) ce critère est alors rempli.

Le confort du passager et sa sécurité, étant les principaux critères de performances à satisfaire, ces critères sont spécifiques aux suspensions.

1. Objectifs de performances :

1.1. Confort :

Parler du confort dans un véhicule, est associée au bien-être des passagers dans le véhicule. Selon l'accélération à laquelle est soumis le passager, ce dernier peut ou non ressentir des malaises. En effet, à une vitesse constante (accélération nulle), le corps humain ne ressent aucune force agissant sur lui, alors que, si l'accélération varie à une certaine fréquence, le passager ressent des malaises. La durée d'exposition aux vibrations, augmente, aussi, la sensation des malaises. Pour améliorer le confort, on est amené à réduire :

- ✓ L'amplitude et la durée de l'accélération verticale de la caisse du véhicule.
- ✓ Les mouvements de roulis et de tangages sont aussi des sources de malaise, la diminution de ces mouvements améliore alors le confort.

1.2. Tenue de route :

La tenue de route, caractérise la capacité du véhicule à adhérer au sol et donc, à répondre aux sollicitations du conducteur. Ceci se traduit par la capacité du système, à fournir une force

longitudinale appropriée (accélération ou freinage) et/ou latérale (virage) entre le pneu et la route.

Dans notre cas, il n'y a que les profils verticaux sur les roues, qui sont prises en compte, cela revient donc, à vouloir que la roue suive le profil de la route.

1.3. Débattement de la suspension :

Dans la réalité, les éléments constituant la suspension, possèdent un débattement limité. Il est donc, important de prendre en compte, le débattement de la suspension dans les critères de performances.

2. Critères pour le modèle quart de véhicule :

Étant donné, que la seule entrée de perturbation prise en compte, dans notre cas, est le profil de la route, les critères à remplir sont :

- ✓ Le confort en basses fréquences étudié à partir du déplacement vertical de la caisse.
- ✓ Le confort en hautes fréquences étudié à partir de la trépidation qui concerne l'accélération verticale de la caisse.
- ✓ Le dernier critère concerne l'adhérence au sol étudiée à partir du déplacement verticale de la roue.

Les spécifications associées à ces critères, sont définies par :

1. Un bon confort en basses fréquences, obtenue en limitant au maximum, le déplacement de la caisse Z_{caisse} (nul dans le cas idéal).
2. un meilleur confort en hautes fréquences, obtenu en diminuant l'accélération de la caisse \ddot{Z}_{caisse} , et l'allongement de cette accélération.
3. une meilleure adhérence, vérifiée en minimisant au maximum, le débattement de la suspension, et l'écrasement du pneu, on cherche alors à annuler les écarts $Z_{caisse} - Z_{roue}$ et $Z_{roue} - Z_{sol}$ respectivement.

On compare pour chaque modèle, les résultats obtenus par la synthèse de la loi de commande par réseaux de neurones, avec d'autres résultats obtenus par d'autres lois de commandes.

3 Critères pour le modèle demi véhicule :

Étant donné, qu'on est dans le cas d'un modèle demi véhicule du type bicyclette, le confort passe, alors, par la limitation de deux mouvements de la caisse à savoir, l'accélération verticale de la caisse, et celle du tangage. Les critères sont alors, en ajoutant ces derniers, les mêmes que le modèle quart du véhicule.

De la même manière que le modèle quart de véhicule, les spécifications associées aux critères répondent aux mêmes exigences à savoir :

1. le confort en basses fréquences passe à minimiser les deux déplacements de la caisse, vertical Z_{caisse} et angulaire θ_{caisse} .
2. meilleur confort en hautes fréquences en minimisant l'allongement des accélérations verticale \ddot{Z}_{caisse} et angulaire $\ddot{\theta}_{caisse}$ de la caisse.
3. une meilleure adhérence, vérifiée en minimisant au maximum le débattement de la suspension et l'écrasement du pneu pour les deux roues avant et arrière, on cherche alors à annuler les écarts $Z_{caisse} - Z_{roue}$ et $Z_{roue} - Z_{sol}$ respectivement pour les deux roues.

On compare aussi, pour chaque modèle nos résultats, avec d'autres résultats obtenus pour d'autres lois de commandes.

Les paramètres du système qui seront utilisés dans toutes les simulations sont :

La masse de la caisse du demi véhicule $m_s = 730 \text{ kg}$.

La masse du coté avant de la caisse $m_f = 40 \text{ kg}$.

La masse du coté arrière de la caisse $m_r = 35 \text{ kg}$.

Le moment d'inertie $I_y = 2460 \text{ kg.m}^2$.

Le coefficient d'amortissement de la suspension avant $b_f = 1290 \text{ N/m}$.

Le coefficient d'amortissement de la suspension arrière $b_r = 1620 \text{ N/m}$.

La raideur du ressort avant $k_f = 19960 \text{ N}$.

La raideur du ressort arrière $k_r = 17500 \text{ N}$.

La raideur du pneu avant $k_{ff} = 175500 \text{ N}$.

La raideur du pneu arrière $k_{fr} = 175500 \text{ N}$.

La distance entre le CdG et l'essieu avant $L_f = 1.001 \text{ m}$.

La distance entre le CdG et l'essieu arrière $L_r = 1.001 \text{ m}$.

II. Stratégies de commande :

II.1. Modèle quart de véhicule ou mono roue

1. Modélisation linéaire :

Le système de suspension est représenté dans ce cas, par le modèle linéaire étudié dans le premier chapitre. Etant donné que l'approche de commande adoptée est du type supervisé, nous introduisons donc, la technique de commande par Backstepping, pour synthétiser une loi de commande permettant de stabiliser le système en boucle fermée. Cette loi de commande sera supervisée par un réseau de neurones approprié.

Approche de Backstepping :

La représentation d'état du système est la suivante (chapitre 1) :

$$\begin{cases} \dot{x}_1 = x_2 \\ \dot{x}_2 = -\frac{1}{m_s} [k_s(x_1 - x_3) + C_s(x_2 - x_4) - u_a] \\ \dot{x}_3 = x_4 \\ \dot{x}_4 = \frac{1}{m_{us}} [k_s(x_1 - x_3) + C_s(x_2 - x_4) - k_{us}(x_3 - r) - u_a] \end{cases} \quad (4.1)$$

Si on veut minimiser le débattement de la suspension, tout en ayant une commande stable, la variable de régulation choisie est le débattement de la suspension. Etant donné que le système n'est pas en cascade, on pose :

$$z_1 = x_1 - \tilde{x}_3 \quad (4.2)$$

Où : $\tilde{x}_3 = \frac{\varepsilon}{s + \varepsilon} x_3$ est la version filtrée de x_3 (choisie pour avoir un signal de commande

stable). On procède alors comme suit :

Étape 1 :

$$\begin{aligned} \dot{z}_1 &= \dot{x}_1 - \dot{\tilde{x}}_3 \\ &= x_2 + \varepsilon \cdot (\tilde{x}_3 - x_3) \\ &= x_2 + \varepsilon \cdot (x_1 - z_1 - x_3) \\ &= x_2 + \varepsilon \cdot (x_1 - x_3) - \varepsilon \cdot z_1 \end{aligned}$$

On choisie donc, la première commande virtuelle :

$$\alpha_1 = -c_1 z_1 - \varepsilon(x_1 - x_3) \quad (4.3)$$

Où : c_1 est une constante positive. La deuxième variable est :

$$z_2 = x_2 - \alpha_1 \quad (4.4)$$

Et l'erreur sera alors, donnée par l'équation : $\dot{z}_1 = -(c_1 + \varepsilon)z_1 + z_2$.

Etape 2 :

On a : $z_2 = x_2 - \alpha_1$

$$\dot{z}_2 = \dot{x}_2 - \dot{\alpha}_1$$

$$= \frac{1}{m_s} [k_s(x_1 - x_3) + C_s(x_2 - x_4) - u_a] - [-c_1(-c_1 z_1 - \varepsilon z_1 + z_2) - \varepsilon(x_2 - x_4)]$$

On choisit donc, le signal de commande :

$$u_a = m_s [-(c_1 - c_2)z_2 + (c_1^2 - 1 + c_1 \cdot \varepsilon)z_1 - \varepsilon(x_2 - x_4) + k_s(x_1 - x_3) + C_s(x_2 - x_4)] \quad (4.5)$$

Où : c_2 est une constante positive, telle que la dérivée de la fonction de Lyapunov

$$V = \frac{1}{2}z_1^2 + \frac{1}{2}z_2^2, \text{ qui est donnée par } \dot{V} = -(c_1 + \varepsilon)z_1^2 - c_2z_2^2 \text{ soit négative.}$$

On obtient alors les erreurs :

$$\dot{z}_1 = -(c_1 + \varepsilon)z_1 + z_2. \quad (4.6)$$

$$\dot{z}_2 = -c_2z_2 - z_1 \quad (4.7)$$

Qui tendent exponentiellement vers zéro.

Le réseau de neurones utilisé pour fournir ce signal de commande, possède l'architecture suivante :

- ❖ Première couche possédant 6 neurones ;
- ❖ Une couche cachée possédant 6 neurones ;
- ❖ Couche de sortie possédant un seul neurone.

La procédure de l'apprentissage est résumée par le schéma ci après :

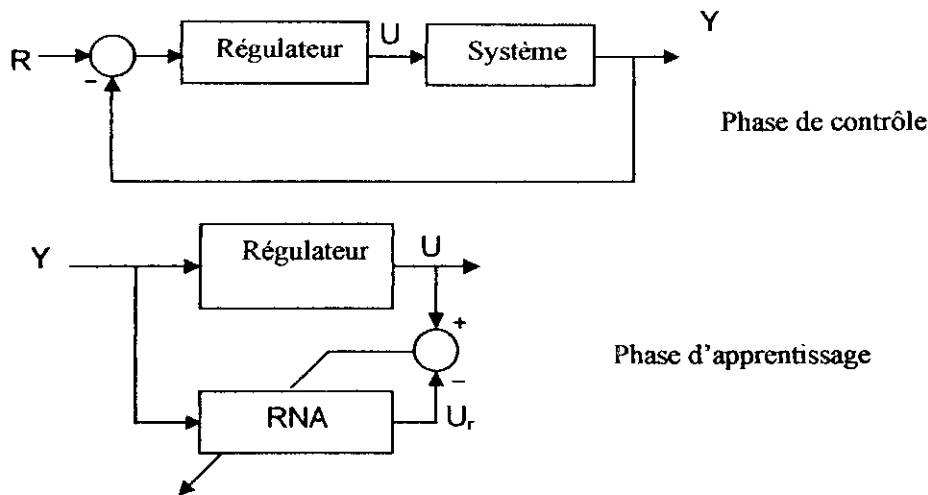


Fig.1. Procédure d'apprentissage du réseau de neurones

Simulation :

La perturbation affectant le processus, est le profil de la route, elle est donnée par :

$$X_r(t) = \mu \cdot \sin\left(\frac{2 \cdot \pi \cdot v}{\lambda} \cdot t\right) \tag{4.8}$$

Où : $w_r = \frac{2 \cdot \pi \cdot v}{\lambda}$ est la pulsation de la perturbation.

Les paramètres de simulation sont les suivants :

- ✓ Temps de simulation est de 3 secondes.
- ✓ Amplitude μ et vitesse v de la perturbation variables.

Les résultats obtenus sont :

- ✓ Suspension passive en vert (trait pointé).
- ✓ Active commandée par Backstepping en rouge (trait discontinu).
- ✓ Active commandée par RNA en bleu (trait continu).

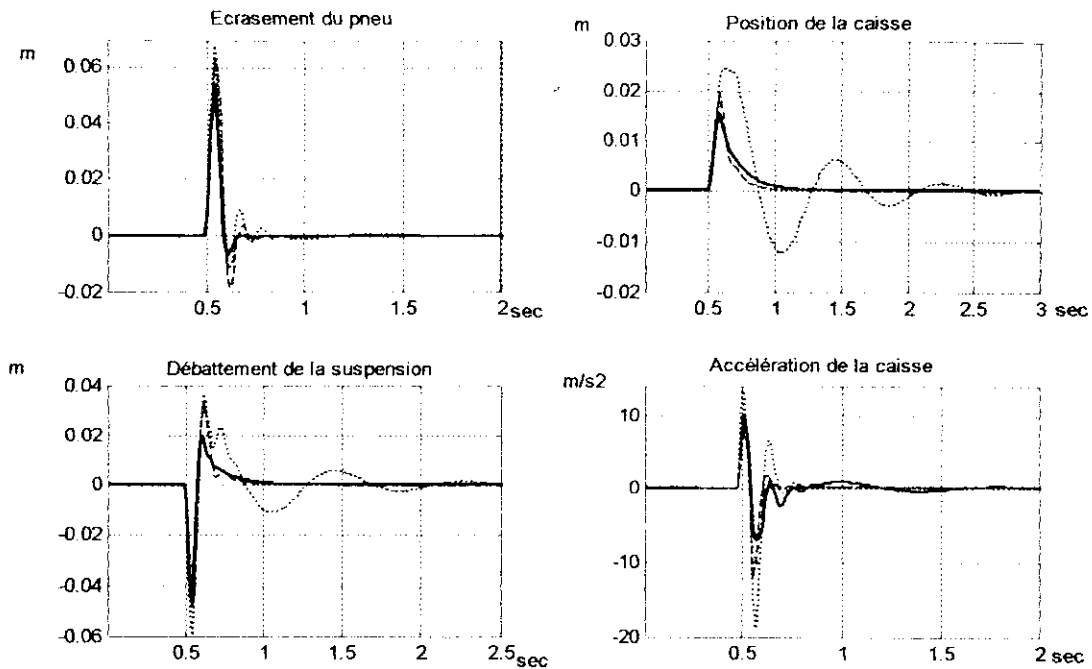


Fig.2 Réponse du véhicule roulant à une vitesse de 40 Km/h pour un passage sur une bosse de 05 cm, pour trois types de suspension.

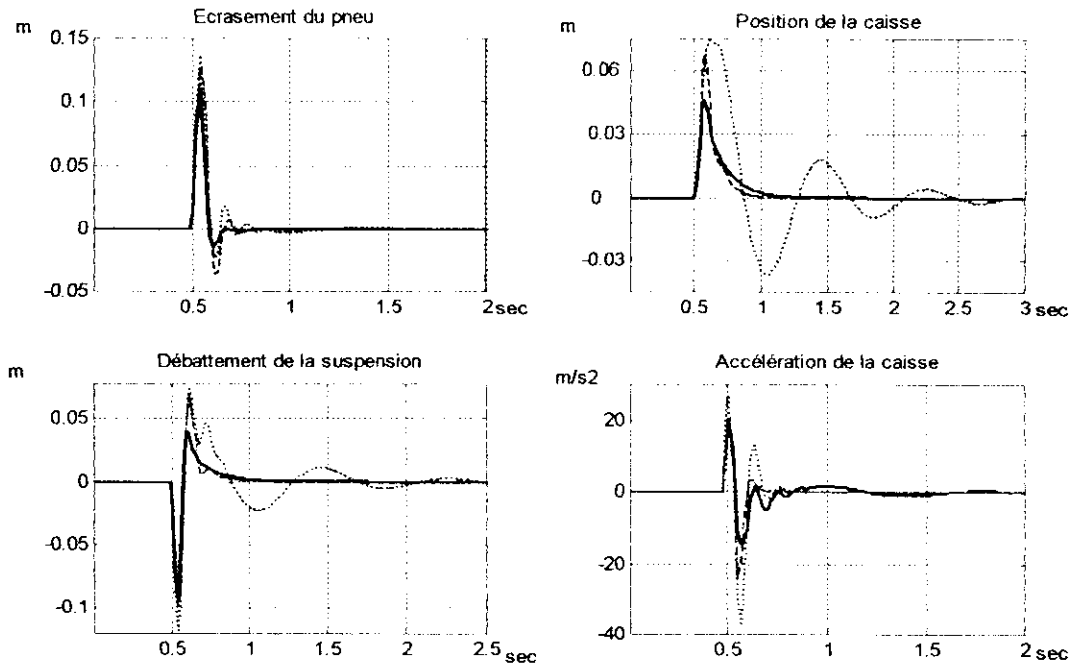


Fig.3 Réponse du véhicule roulant à une vitesse de 40 Km/h pour un passage sur une bosse de 10 cm, pour trois types de suspension.

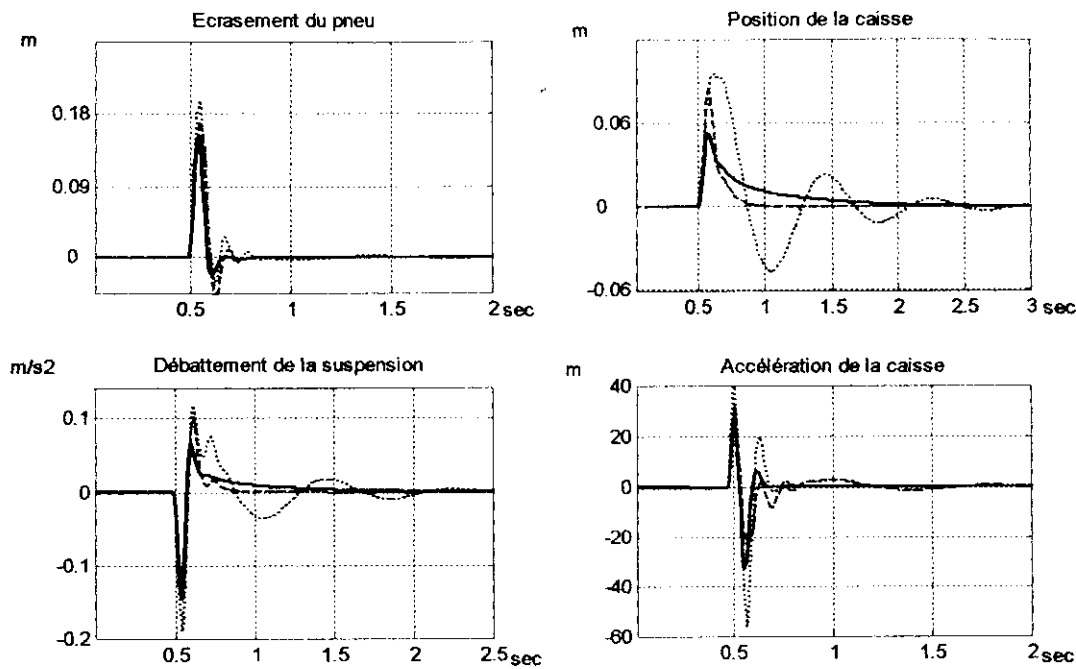


Fig.4 Réponse du véhicule roulant à une vitesse de 40 Km pour un passage sur une bosse de 15 cm, pour trois types de suspension.

Analyse des résultats :

Les figures (2 à 4), représentent les réponses temporelles de différents types de suspensions, pour un profil de route inégal, on voit clairement que la commande améliore les performances du système. L'introduction de la suspension active, et donc apport d'énergie externe, résout le dilemme ; confort-stabilité existant pour les suspensions passives. Etant donnée que, la suspension utilisée ici, est dure, on voit que le critère de sécurité (la tenue de route du véhicule) est satisfait par les deux types de suspensions (active et passive). En ce qui concerne le confort dans le véhicule, une amélioration constante est apportée par la suspension active (la caisse s'éloigne moins de sa position d'équilibre et y revient plus rapidement), à l'inverse, la suspension passive sacrifie ce critère. De même, les amplitudes maximales des autres variables (accélération, débattement de la suspension) sont moins importantes pour une suspension active qu'une suspension passive. Par conséquent, la suspension active fournit un meilleur, confort et sécurité du véhicule par apport aux autres systèmes conventionnels de suspensions.

La deuxième constatation à faire, concerne la robustesse des réseaux de neurones, en effet ; les caractéristiques obtenues par un correcteur neuronal sont, relativement meilleurs que celles obtenues par un correcteur développé par l'approche de Backstepping. Ces résultats

montrent, donc, la capacité d'approximation neuronale, et justifient leur utilisation pour les modèles linéaires.

2. Modélisation non linéaire :

Le système de suspension est représenté, par le modèle non linéaire étudié dans le premier chapitre. Nous adoptons, tous jour, la technique de la commande supervisée. Nous introduisant aussi la technique de commande par Backstepping. Etant donnée, les non linéarités apportées par l'introduction de l'actionneur, un bouclage non linéaire est introduit pour simplifier le calcul de la commande.

Calcul de la commande :

La représentation d'état du système est donnée par

$$\begin{cases} \dot{x}_1 = x_2 \\ \dot{x}_2 = -\frac{1}{m_s} [k_s \cdot (x_1 - x_3) + C_s \cdot (x_2 - x_4) - A \cdot x_5] \\ \dot{x}_3 = x_4 \\ \dot{x}_4 = \frac{1}{m_{us}} [k_s \cdot (x_1 - x_3) + C_s \cdot (x_2 - x_4) - k_{us} \cdot (x_r - x_2) - A x_5] \\ \dot{x}_5 = -\beta \cdot x_5 - \alpha \cdot A \cdot (x_2 - x_4) + \gamma \cdot \omega_3 \cdot x_6 \\ \dot{x}_6 = \frac{1}{\tau} \cdot (u - x_6) \end{cases} \quad (4.9)$$

$$\text{Avec : } \omega_3 = \text{sgn}[P_s - \text{sgn}(x_6) \cdot x_5] \sqrt{|P_s - \text{sgn}(x_6) \cdot x_5|}$$

Le système est sous la forme :

$$\dot{X} = f(X) + g(X) \cdot u$$

Où :

$$f(X) = \begin{bmatrix} x_2 \\ -\frac{1}{m_s} [k_s \cdot (x_1 - x_3) + C_s \cdot (x_2 - x_4) - A \cdot x_5] \\ x_4 \\ \frac{1}{m_{us}} [k_s \cdot (x_1 - x_3) + C_s \cdot (x_2 - x_4) - k_{us} \cdot (x_3 - r) - A \cdot x_5] \\ -\beta \cdot x_5 - \alpha \cdot A \cdot (x_2 - x_4) + \gamma \cdot \omega_3 \cdot x_6 \\ -\frac{1}{\tau} x_6 \end{bmatrix} \quad \text{et} \quad g(X) = \begin{bmatrix} 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ \frac{1}{\tau} \end{bmatrix} \quad (4.10)$$

On pose $h(x) = x_1$, (4.11)

On calcule, alors, les crochés de Lie jusqu'à avoir $L_g L_f^r h(x) = 0$;

$$L_g h(x) = \frac{\partial h}{\partial X} g = 0 \tag{4.12}$$

$$L_f h(x) = \frac{\partial h}{\partial X} = x_2 \tag{4.13}$$

$$L_g L_f h(x) = \frac{\partial(L_f h(X))}{\partial X} = 0 \tag{1.14}$$

$$L_f^2 h(x) = -\frac{1}{m_s} [k_s(x_1 - x_3) + C_s(x_2 - x_4) - Ax_5]. \tag{1.15}$$

$$L_g L_f^2 h(x) = 0. \tag{4.16}$$

$$L_f^3 h(x) = \left[-\frac{k_s}{m_s} \quad -\frac{C_s}{m_s} \quad \frac{k_s}{m_s} \quad \frac{C_s}{m_s} \quad \frac{A}{m_s} \quad 0 \right] \times f(X).$$

$$L_f^3 h(x) = \left(\frac{k_s \cdot C_s}{m_s^2} + \frac{k_s \cdot C_s}{m_s \cdot m_{us}} \right) x_1 + \left(\frac{C_s^2}{m_s^2} - \frac{k_s}{m_s} + \frac{C_s^2}{m_s \cdot m_{us}} - \frac{\alpha \cdot A^2}{m_s} \right) x_2 - \left(\frac{k_s \cdot C_s}{m_s^2} + \frac{k_s \cdot C_s}{m_s \cdot m_{us}} + \frac{k_{us} \cdot C_s}{m_s \cdot m_{us}} \right) x_3 - \left(\frac{C_s^2}{m_s^2} - \frac{k_s}{m_s} + \frac{C_s^2}{m_s \cdot m_{us}} - \frac{\alpha \cdot A^2}{m_s} \right) x_4 - \left(\frac{AC_s}{m_s^2} + \frac{AC_s}{m_s \cdot m_{us}} + \frac{A \cdot \beta}{m_s} \right) x_5 + \frac{A \cdot \gamma \cdot \omega_3}{m_s} x_6 + \frac{k_{us} \cdot C_s}{m_s \cdot m_{us}} x_r$$

$$L_g L_f^3 h(x) = \frac{\partial}{\partial x_6} \left(\frac{A \cdot \gamma \cdot \omega_3}{m_s} \cdot x_6 \right) \cdot \frac{1}{\tau} = \frac{1}{\tau} \cdot \frac{A \cdot \gamma}{m_s} \left[x_6 \cdot \frac{\partial \omega_3}{\partial x_6} + \omega_3 \right]. \tag{4.17}$$

$L_g L_f^3 h(x) = \frac{A \cdot \gamma}{\tau \cdot m_s} \omega_3 \neq 0$ Alors $r = 4$ est l'ordre du sous système linéaire.

$$L_f^4 h(x) = \left(\frac{k_s \cdot C_s}{m_s^2} + \frac{k_s \cdot C_s}{m_s \cdot m_{us}} \right) x_2 + \left(\frac{C_s^2}{m_s^2} - \frac{k_s}{m_s} + \frac{C_s^2}{m_s \cdot m_{us}} + \frac{\alpha \cdot A^2}{m_s} \right) \left\{ -\frac{1}{m_s} [k_s(x_1 - x_3) + C_s(x_2 - x_4) - Ax_5] \right\} - \left(\frac{k_s \cdot C_s}{m_s} + \frac{k_s \cdot C_s}{m_s \cdot m_{us}} + \frac{k_{us} \cdot C_s}{m_s \cdot m_{us}} \right) x_4 - \left(\frac{C_s^2}{m_s^2} - \frac{k_s}{m_s} + \frac{C_s^2}{m_s \cdot m_{us}} - \frac{\alpha \cdot A^2}{m_s} \right) * \left\{ \frac{1}{m_{us}} [k_s(x_1 - x_3) + C_s(x_2 - x_4) - k_{us}(x_3 - x_r) - Ax_5] \right\} - \left(\frac{AC_s}{m_s^2} + \frac{AC_s}{m_s \cdot m_{us}} + \frac{A \cdot \beta}{m_s} - \xi \right) * (-\beta \cdot x_5 - \alpha \cdot A(x_2 - x_4) + \gamma \cdot \omega_3 x_6) - \frac{A \cdot \gamma}{m_s \cdot \tau} \omega_3$$

Où :

$$\xi = \frac{A\gamma}{m_s} x_6 \cdot \frac{\partial \omega_3}{\partial x_5} \text{ Avec :}$$

$$\frac{\partial \omega_3}{\partial x_5} = \text{sgn}[P_s - \text{sgn}(x_6) \cdot x_5] \frac{(P_s - \text{sgn}(x_6) \cdot x_5 < 0) \cdot \text{sgn}(x_6) - (P_s \text{sgn}(x_6) > 0) \cdot \text{sgn}(x_6)}{\sqrt{|P_s - \text{sgn}(x_6) \cdot x_5|}}$$

En posant : les changement de variables :

$$\left. \begin{aligned} Z_1 &= h(x) \\ Z_2 &= L_f h(x) \\ Z_3 &= L_f^2 h(x) \\ Z_4 &= L_f^3 h(x) \\ Z_5 &= \lambda_1(x) \\ Z_6 &= \lambda_2(x) \end{aligned} \right\} \Rightarrow \left\{ \begin{aligned} \dot{Z}_1 &= Z_2 \\ \dot{Z}_2 &= Z_3 \\ \dot{Z}_3 &= Z_4 \\ \dot{Z}_4 &= V \end{aligned} \right. \quad (4.18)$$

Approche de Backstepping:

Etape 1 :

Soit la fonction de Lyapunov $V_1 = \frac{1}{2} \cdot Z_1^2 \Rightarrow \dot{V}_1 = Z_1 \cdot Z_2$

On choisit la première commande virtuelle $\alpha_1 = Z_2 = -k_1 Z_1$ (4.19)

Ce qui donne $\dot{V}_1 = -k_1 \cdot Z_1^2 < 0$

Etape 2 :

On prend la deuxième fonction de Lyapunov, $V_2 = V_1 + \frac{1}{2} \cdot (\alpha_1 - Z_2)^2$

Ce qui donne $\dot{V}_2 = \dot{V}_1 + (\alpha_1 - Z_2)(\dot{\alpha}_1 - \dot{Z}_2)$

On choisit la deuxième commande virtuelle : $\alpha_2 = Z_3 = \dot{\alpha}_1 - k_1(\alpha_1 - Z_2)$ (4.20)

On assure alors que $\dot{V}_2 < 0$

Etape 3 :

On prend la troisième fonction de Lyapunov $V_3 = V_2 + \frac{1}{2} \cdot (Z_3 - \alpha_2)^2$

Ce qui donne $\dot{V}_3 = \dot{V}_2 + (Z_3 - \alpha_2)(Z_4 - \dot{\alpha}_2)$

On choisit la troisième commande virtuelle : $\alpha_3 = Z_4 = \dot{\alpha}_2 - k_3(Z_3 - \alpha_2)$ (4.21)

On assure donc que $\dot{V}_3 < 0$

Etape 4 :

On choisit la quatrième fonction de Lyapounov $V_4 + V_3 + \frac{1}{2}(Z_4 - \alpha_3)^2$

Et donc ; $\dot{V}_4 = \dot{V}_3 + (Z_4 - \alpha_3)(W - \dot{\alpha}_3)$

On obtient, en fin la commande : $V = \alpha_3 - k_4(Z_4 - \alpha_3)$ (4.22)

Les termes $k_1, k_2, k_3,$ et k_4 sont des constantes positives.

La loi de commande finale est donnée par : $u = \frac{L_f^4 h(x) + V}{L_x L_f^3 h(x)}$ (4.23)

Le réseau de neurones utilisée, pour superviser cette loi de commande, possède la même L'architecture que le cas précédent.

La perturbation affectant le processus est tous jour, la sollicitation verticale de la route, qui donnée donc, par :

$$X_r(t) = \mu \cdot \sin\left(\frac{2 \cdot \pi \cdot v}{\lambda} t\right).$$

Ou : $w_r = \frac{2 \cdot \pi \cdot v}{\lambda}$ est la pulsation de la perturbation

Les résultats obtenus sont :

- ✓ Suspension passive en vert (trait pointé).
- ✓ Active commandée par Backstepping en rouge (trait discontinu).
- ✓ Active commandée par RNA en bleu (trait continu).

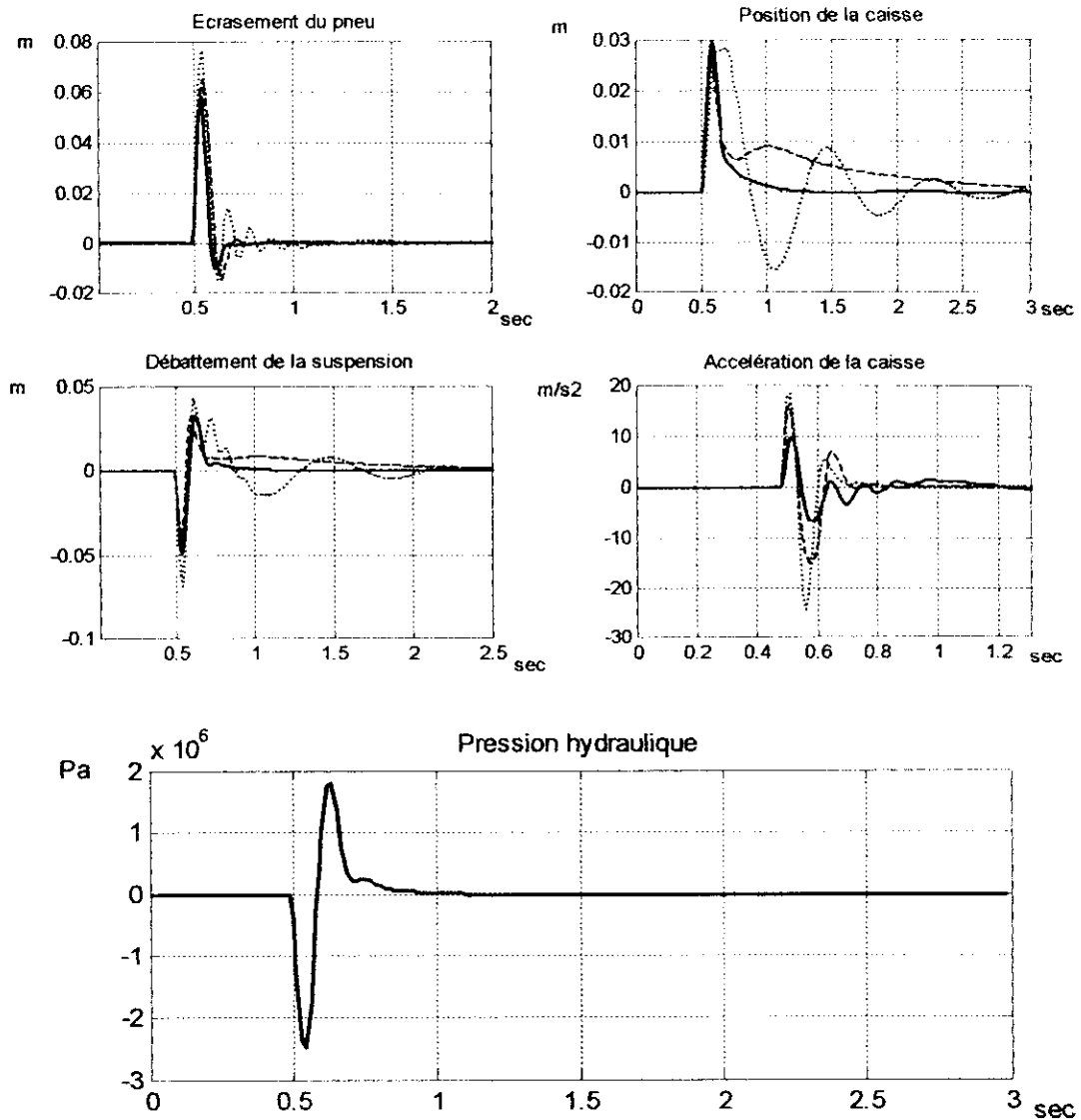


Fig.5. Passage du véhicule sur une bosse de 05 cm à une vitesse de 40 Km/h

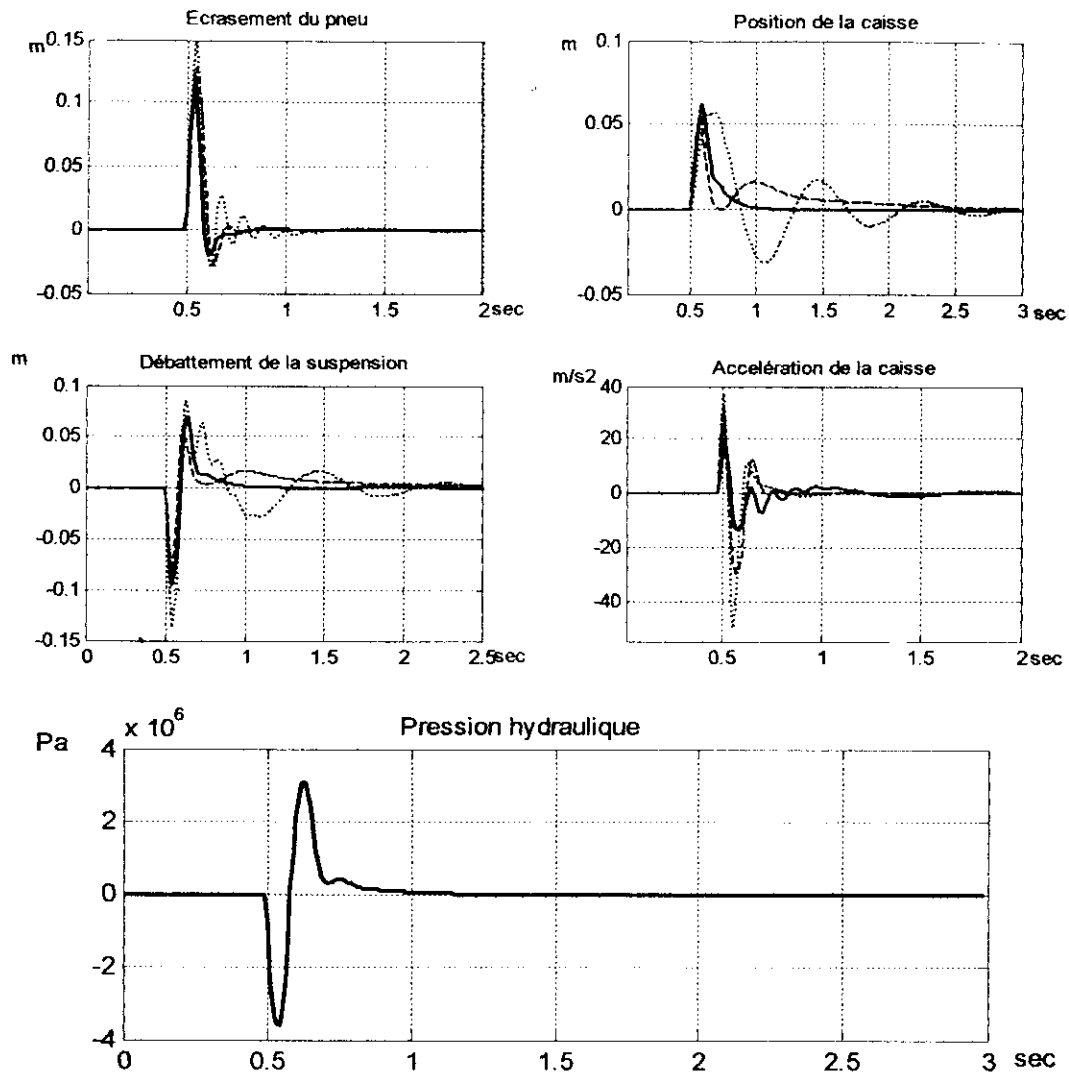


Fig.6. Passage sur une bosse de 10 cm à une vitesse de 40 Km/h.

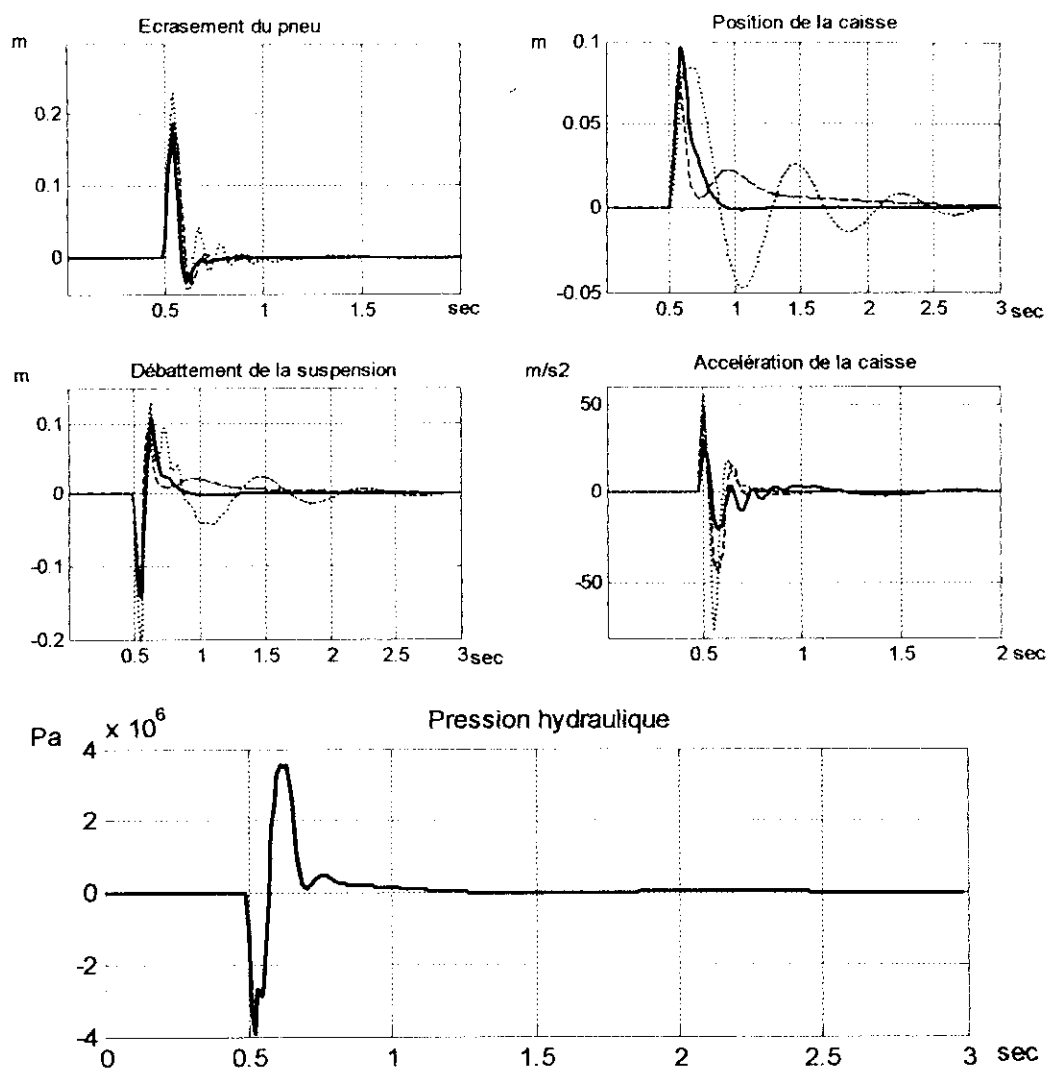


Fig.7. Passage sur une bosse de 15 cm à une vitesse de 40 Km/h

Analyse des résultats :

Les figures (5 à 7) représentent les réponses temporelles des différents types de suspensions sur une route inégale. Les mêmes remarques que le cas linéaire, peuvent être faites dans ce cas aussi, en comparant les réponses du système passif, à celles du système actif, à savoir ; l'amélioration des performances du confort dans le cas de la suspension active (déplacement minimale de la caisse, et retour en douceur, sans oscillations, à la position d'équilibre), l'amélioration des performances de sécurité, (l'écrasement de la roue suit le profil de la route), et en fin, la minimisation du débattement de la suspension (minimisation des amplitudes des pics). Les pressions hydrauliques développées par les suspensions, doivent rester dans le domaine de fonctionnement des actionneurs utilisés ; cette contrainte est, aussi,

vérifié. On peut remarquer aussi, l'apport du correcteur neuronal sur le système de commande, et sa robustesse vis à vis des différentes perturbations. Il apparaît clairement que le rejet de perturbation est complètement fait.

Modèle demi véhicule

1. Modélisation linéaire :

Le système de suspension est représenté dans ce cas, par le modèle linéaire étudié dans le premier chapitre. A fin de synthétiser les lois de commandes pour le système, deux bouclages avec filtres seront utilisés. Le schéma suivant résume cette stratégie de commande.

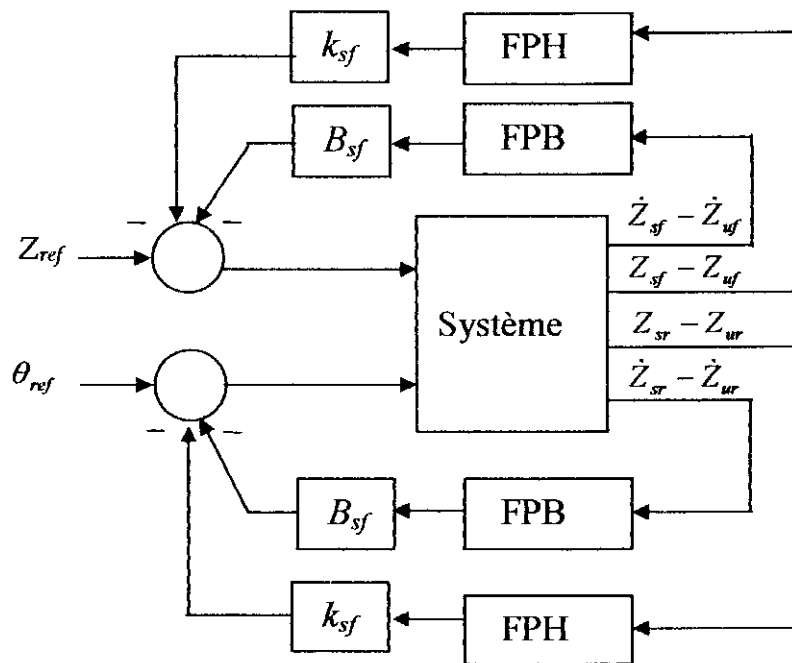


Fig.8. -Représentation du schéma de commande-

Fonctionnement : le régulateur procède de la manière suivante :

L'amplitude et le sens de la force générée par l'actionneur, permettent à la suspension de devenir dure ou moue selon le besoin, en effet :

En dessous de la fréquence de la roue $\omega_0 = \sqrt{\frac{k_u}{m_u}}$, on peut améliorer les performances du

système, en diminuant la raideur de la suspension, ceci peut être fait en introduisant un filtre passe haut dans la boucle de retour sur la raideur du ressort.

En dessus de la fréquence de la roue, l'amélioration des performances est obtenue en diminuant l'amortissement du système. Ceci peut être fait en introduisant un filtre passe bas dans la boucle de retour sur l'amortissement, on procède, donc, comme suit :

On introduit les variables auxiliaires suivantes :

- En hautes fréquences :

$$x_9 = \frac{w_{bf}}{s + w_{bf}} (\dot{z}_{sf} - \dot{z}_{uf}) = \frac{w_{bf}}{s + w_{bf}} (x_2 - a \cdot x_4 - x_5). \quad (4.24)$$

$$x_{10} = \frac{w_{br}}{s + w_{br}} (\dot{z}_{sr} - \dot{z}_{ur}) = \frac{w_{br}}{s + w_{br}} (x_2 + b \cdot x_6 - x_7). \quad (4.25)$$

Sous forme d'état, on aura :

$$\dot{x}_9 = -w_{bf} \cdot x_9 + w_{bf} \cdot (x_2 - a \cdot x_4 - x_5) \quad (4.26)$$

$$\dot{x}_{10} = -w_{br} \cdot x_{10} + w_{br} \cdot (x_2 + b \cdot x_6 - x_7) \quad (4.27)$$

- En basse fréquences :

$$x_{5w} = \frac{s}{s + w_{kf}} (z_{sf} - z_{uf}) = \frac{s}{s + w_{kf}} \cdot x_5 \quad (4.28)$$

$$x_{7w} = \frac{s}{s + w_{kr}} (z_{sr} - z_{ur}) = \frac{s}{s + w_{kr}} \cdot x_7 \quad (4.29)$$

Pour mettre sous la forme d'état, soit ;

$$x_{5w} = \frac{s}{s + w_{kf}} \cdot x_5 = \left[1 - \frac{w_{kf}}{s + w_{kf}} \right] \cdot x_5 = x_5 - w_{kf} \cdot \bar{x}_5.$$

$$x_{7w} = \frac{s}{s + w_{kr}} \cdot x_7 = \left[1 - \frac{w_{kr}}{s + w_{kr}} \right] \cdot x_7 = x_7 - w_{kr} \cdot \bar{x}_7.$$

Posons, alors, $x_{11} = \bar{x}_5$, $x_{12} = \bar{x}_7$ on aura :

$$\begin{cases} \dot{x}_{11} = -w_{kf} \cdot x_{11} + x_5. \\ x_{5w} = -w_{kf} \cdot x_{11} + x_5. \\ \dot{x}_{12} = -w_{kr} \cdot x_{12} + x_7. \\ x_{7w} = -w_{kr} \cdot x_{12} + x_7. \end{cases} \quad (4.30)$$

À partir du schéma de la figure (8), on obtient les signaux de commandes donnés par :

$$f = \begin{bmatrix} f_f \\ f_r \end{bmatrix} = \begin{bmatrix} B_{sf} & k_{sf} & 0 & 0 \\ 0 & 0 & B_{sr} & k_{sr} \end{bmatrix} \begin{bmatrix} x_9 \\ x_{5w} \\ x_{10} \\ x_{7w} \end{bmatrix} \quad (4.31)$$

Le réseau de neurones utilisé pour superviser ces lois de commande possède l'architecture suivante :

- ✓ Six de neurones dans la couche d'entrée.
- ✓ Huit neurones cachés.
- ✓ Deux neurones de sortie.

Le schéma ci-dessous, résume la procédure d'apprentissage :

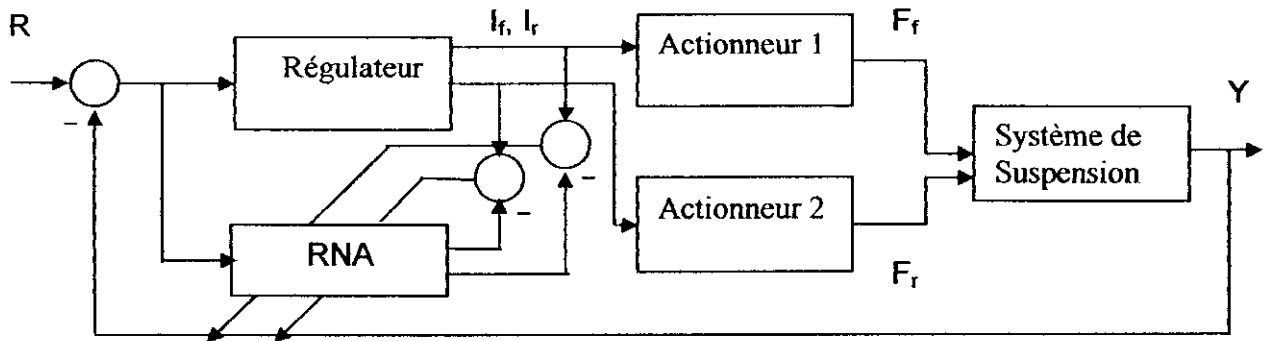


Fig. 9. Procédure d'apprentissage supervisé.

Les perturbations affectant le processus sont les profils verticaux de la route, elles sont données comme suit :

Pour la roue avant ;
$$X_f(t) = \mu \cdot \sin\left(\frac{2 \cdot \pi \cdot v}{\lambda} \cdot t\right).$$

Où ;
$$w_r = \frac{2 \cdot \pi \cdot v}{\lambda}$$
 est la pulsation de la perturbation.

Pour la roue arrière ; la perturbation affectant la roue arrière est la version retardée de celle affectant la roue avant, elle est donc, donnée par :

$$X_r(t) = \mu \cdot \sin\left(\frac{2 \cdot \pi \cdot v}{\lambda} \cdot \left(t + \frac{L}{v}\right)\right).$$

Où ;

L est la distance entre les deux roues.

v est la vitesse du véhicule.

Les résultats de simulation sont représentés sur les figures ci après, pour :

- ✓ Une suspension passive en vert (en pointé).
- ✓ Active commandé par bouclages filtrés, en rouge (trait discontinu).
- ✓ Active commandée par RNA en bleu (trait continu)

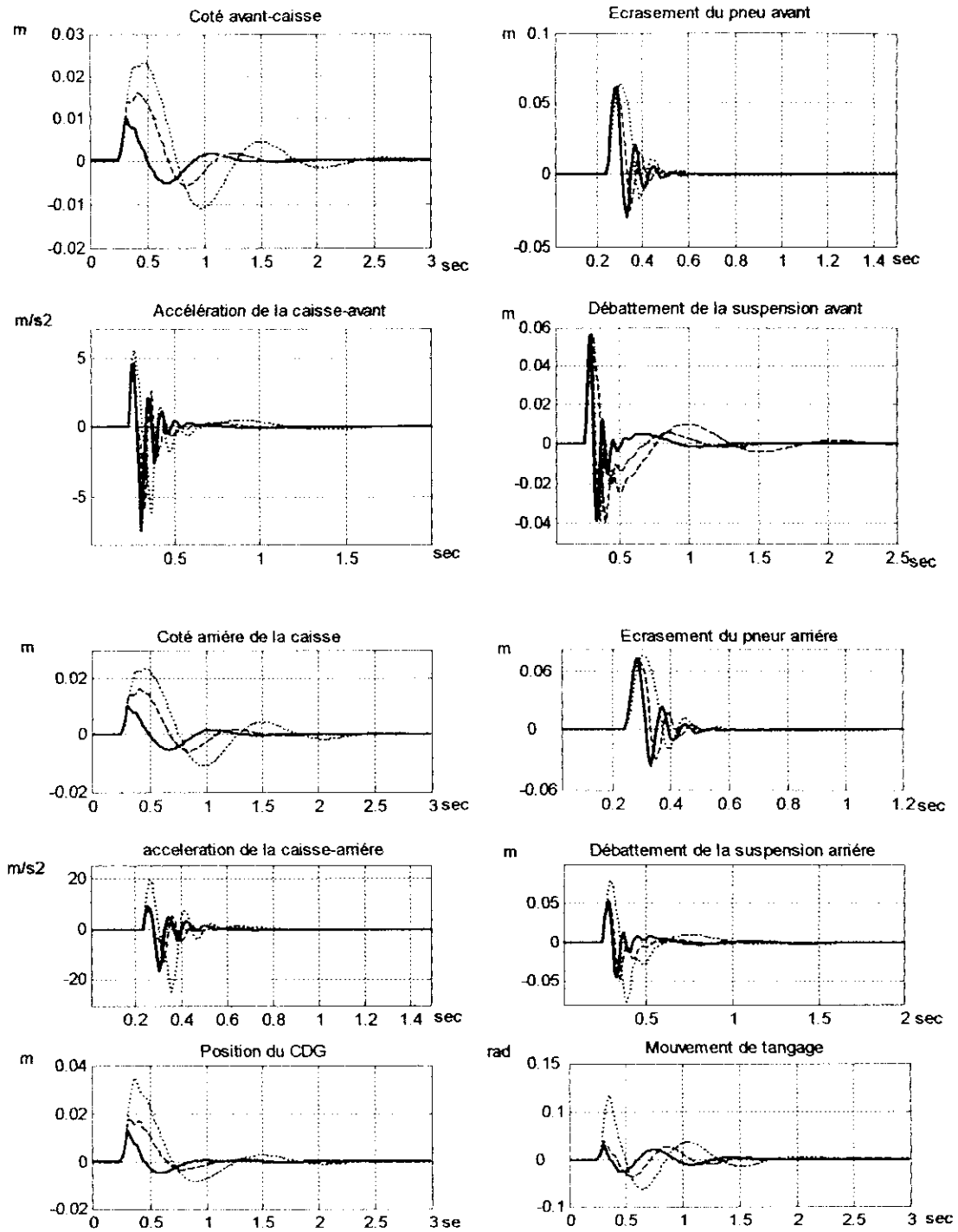


Fig.10- Passage du véhicule sur une bosse de 5 cm avec une vitesse de 45Km/h.

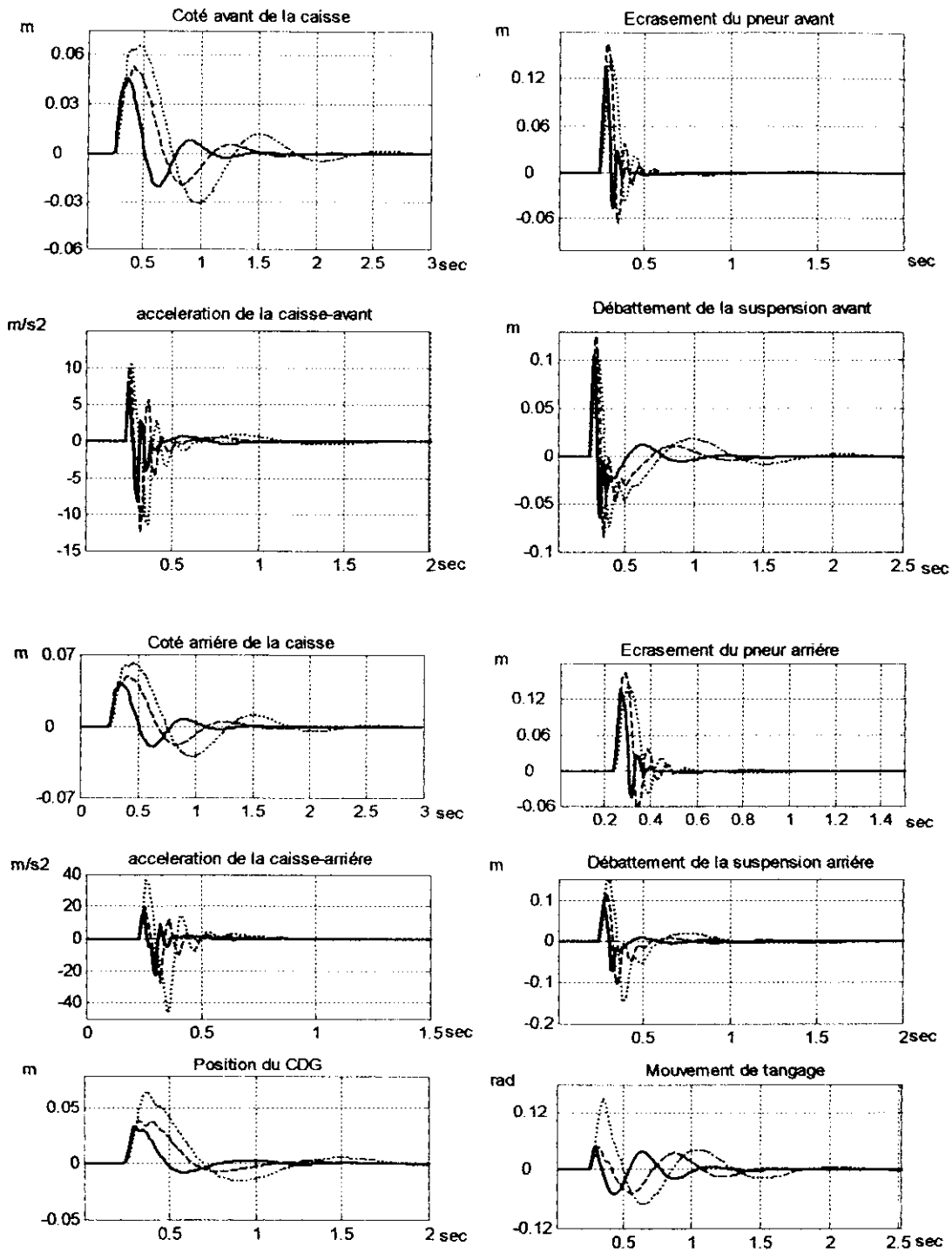


Fig.11 - Passage du véhicule sur une bosse de 10 cm avec une vitesse de 45Km/h

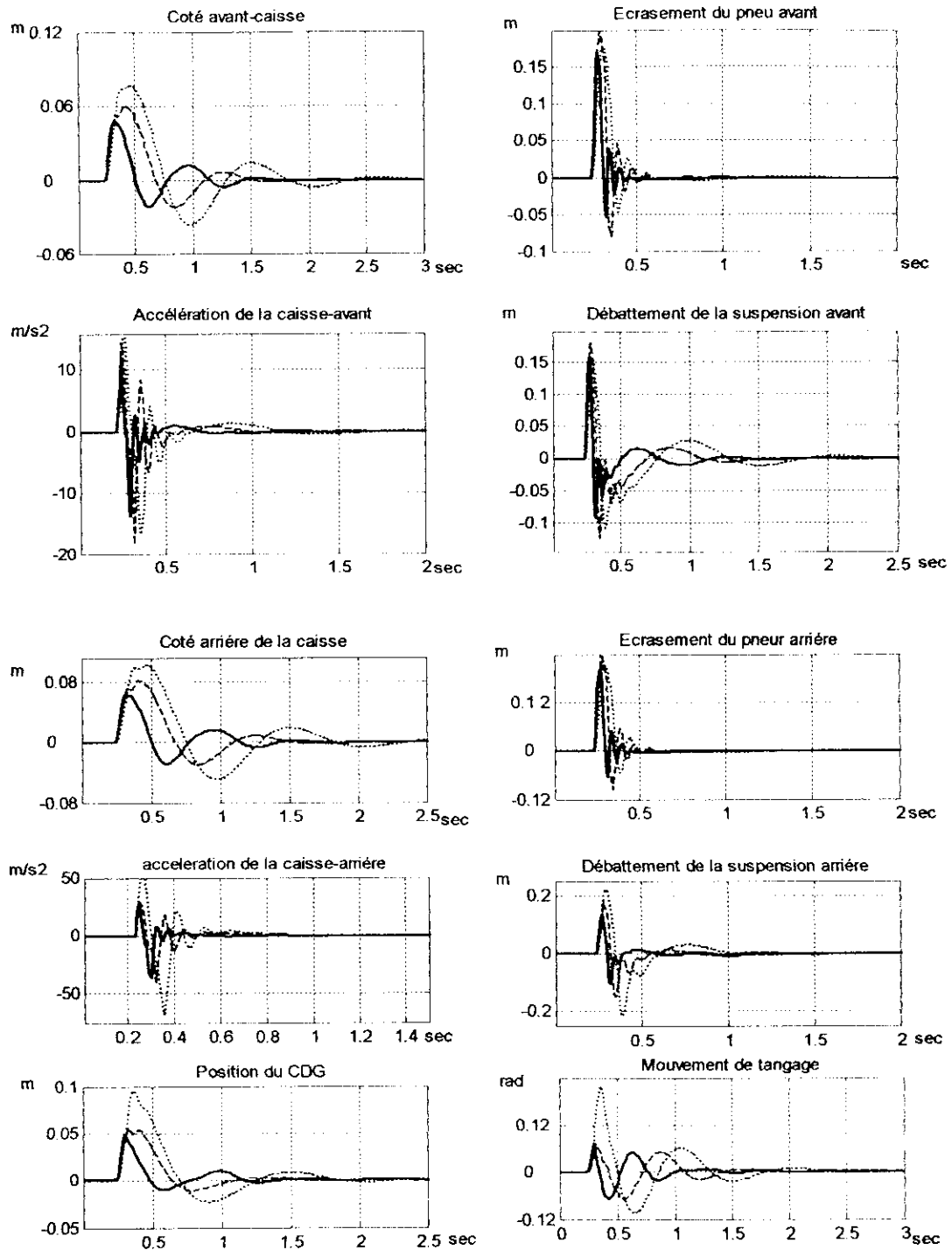


Fig.12. Passage du véhicule sur une bosse de 15 cm avec une vitesse de 50Km/h.

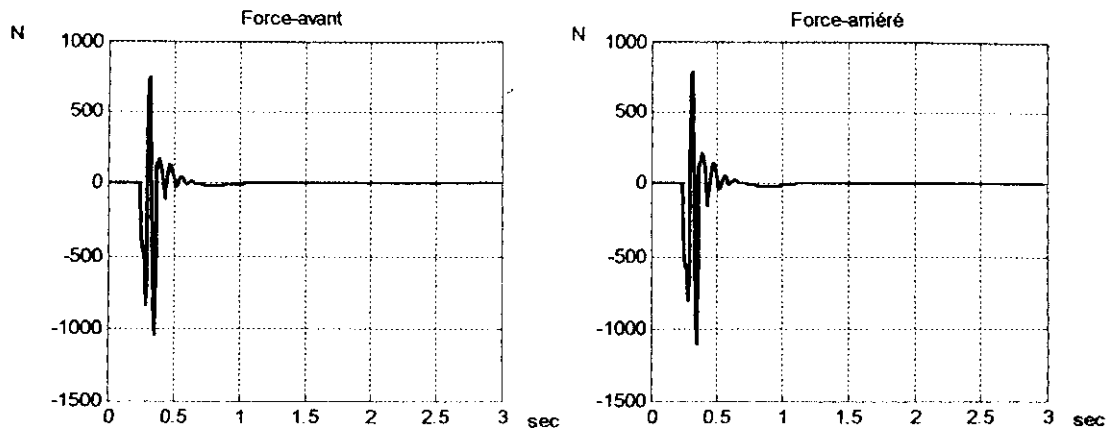


Fig.13. Force développée par l'actionneur pour une bosse de 5cm.

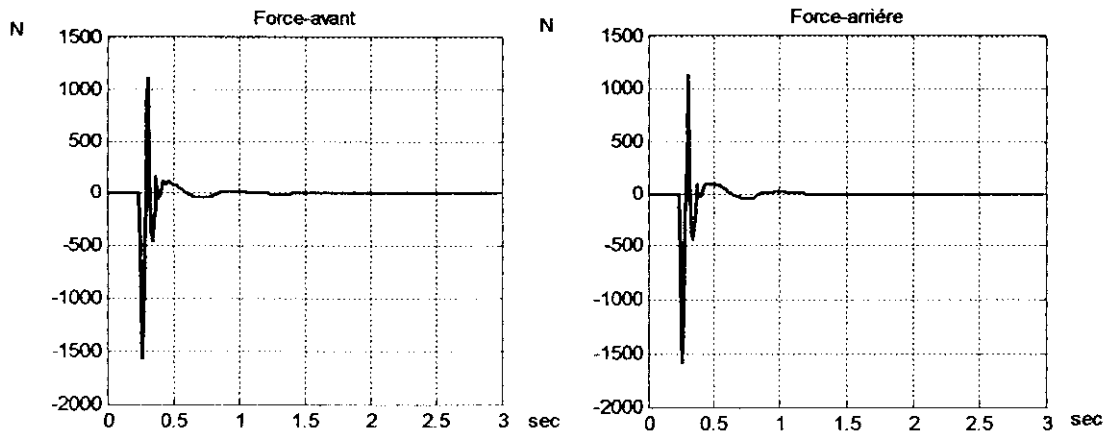


Fig.14. Force développée pour une bosse de 10 cm.

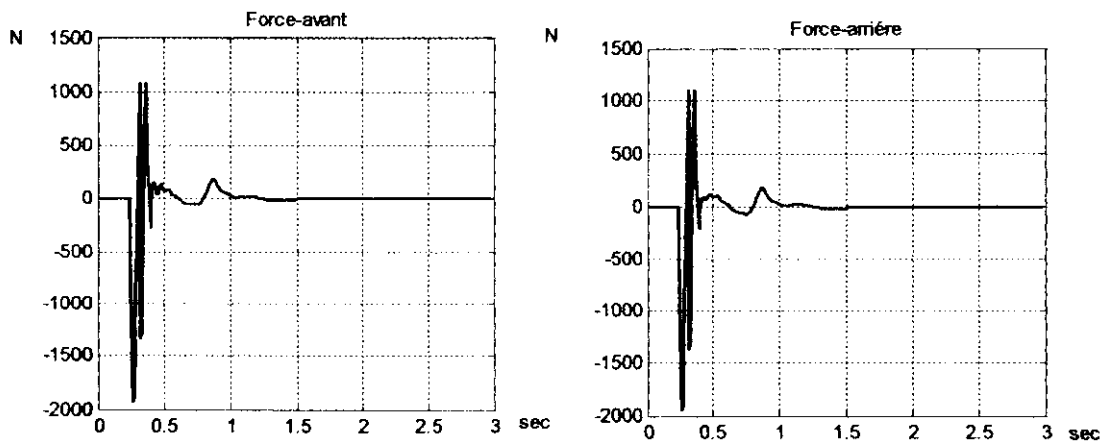


Fig.15. Force développée pour une bosse de 15 cm.

Analyse des résultats :

Les figures (10 à 15) représentent les tests temporels effectués sur les différents types de suspension pour différentes bosses. On peut faire les mêmes constatations que le cas du modèle quart de véhicule, en ce qui concerne l'atténuation des vibrations de la caisse dans le cas actif, suite à une perturbation due au profil de la route, et retour à la position d'équilibre en douceur (sans oscillations), tout en assurant une bonne tenue de route du véhicule (écrasement minimal du pneu), et un rejet rapide de la perturbation. Le mouvement du tangage (oscillation au tour de l'axe transversal du véhicule) est autre critère à prendre en compte, dans le cas du modèle demi véhicule, une amélioration constante est apportée par la suspension active, en comparaison avec la suspension passive. On note, aussi, qu'il n'y a pas de saturation de commande, en effet ; les forces développées par les suspensions, sont restées dans le domaine de fonctionnement des actionneurs utilisés (cette contrainte est d'importance cruciale pour le bon fonctionnement du véhicule).

Dans tous les cas de figures, les capacités d'approximation, et la robustesse des réseaux de neurones peuvent être clairement remarquées, sur les différents résultats, en effet les performances du système de commande de la suspension se sont révélées excellentes, et l'approximation des dynamiques du régulateur déjà existant est parfaitement réalisée, avec une prise en compte du critère de minimisation.

Modélisation non linéaire :

Le système de suspension est représenté par le modèle non linéaire étudié dans le premier chapitre. On utilise tous les jours, un réseau neuronal pour commander le système. La technique de commande utilisée est aussi du type supervisé.

Etant donné que, le système est assez complexe (prise en charge de différentes dynamiques non linéaires du système, et extension du nombre de variables d'état par rapport au cas du modèle quart de véhicule), il est alors, trop difficile et fastidieux de synthétiser les lois de commandes assurant la stabilité du système bouclés, et servant à l'apprentissage du réseau de neurones, en utilisant l'une des approches précédentes. L'idée est ici, de subdiviser le système complet, en deux sous systèmes représentant chacun, le côté avant et arrière du demi véhicule et ensuite, synthétiser les lois de commandes du système décentralisé en utilisant l'approche du Backstepping avec bouclage non linéaire.

Le réseau de neurone utilisé pour superviser ces lois de commande possède la même architecture que le cas linéaire. Les perturbations affectant le système ont la même forme que le cas précédent. On a abouti alors, aux résultats suivants :

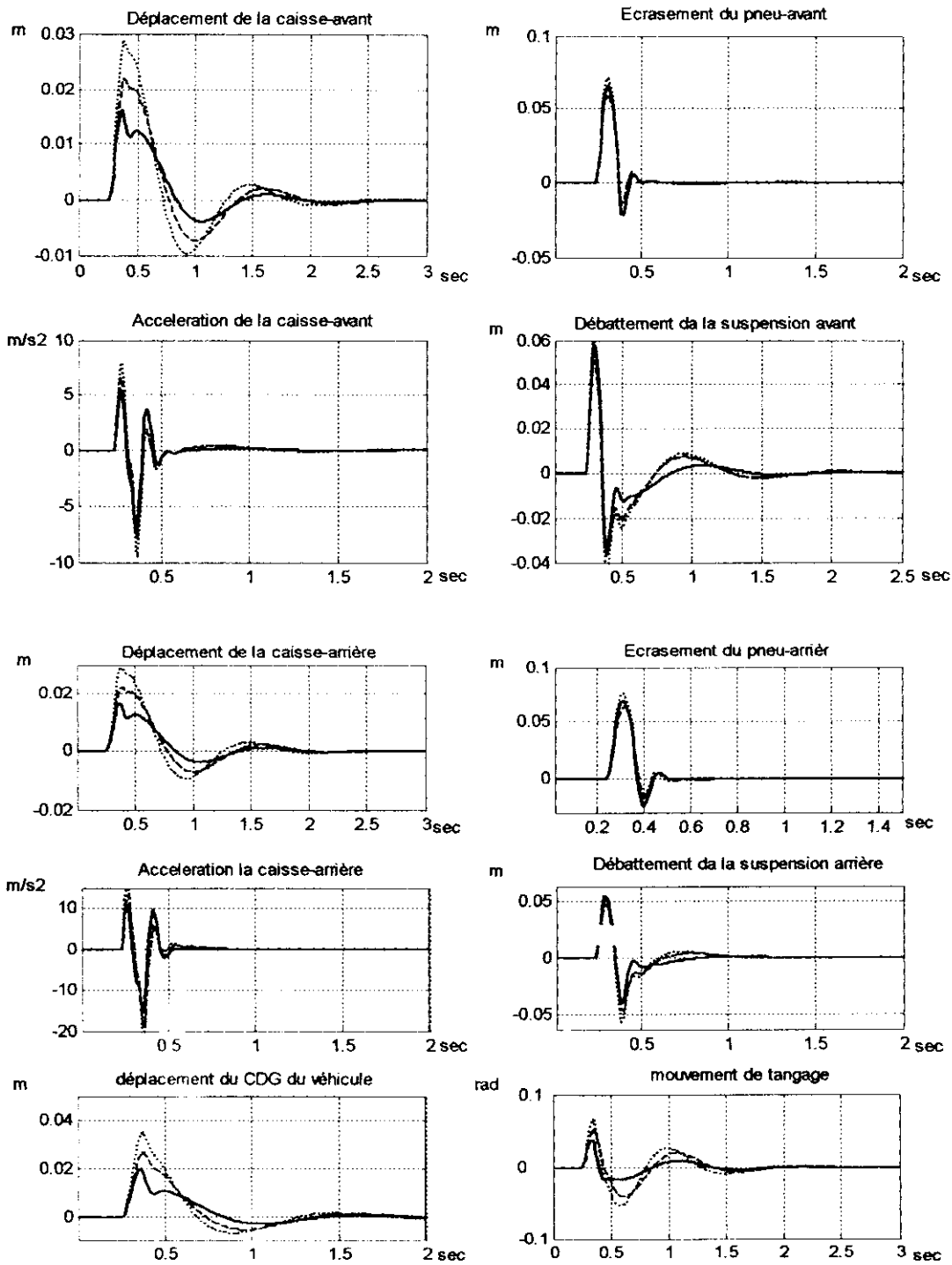


Fig.16 Passage sur une bosse de 05 cm à une vitesse de 45 Km/h.

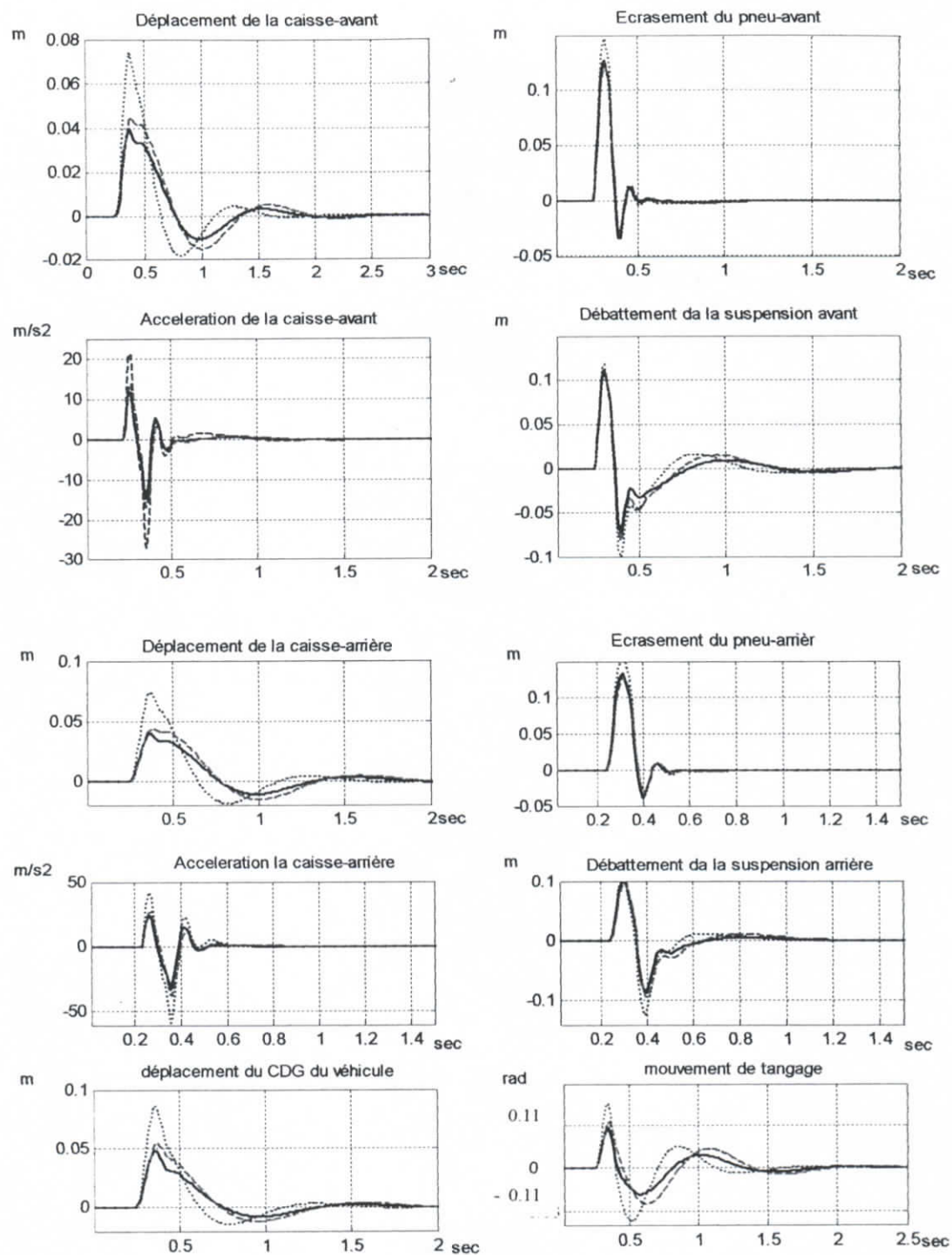


Fig.17 Passage sur une bosse de 10 cm à une vitesse de 45 Km/h

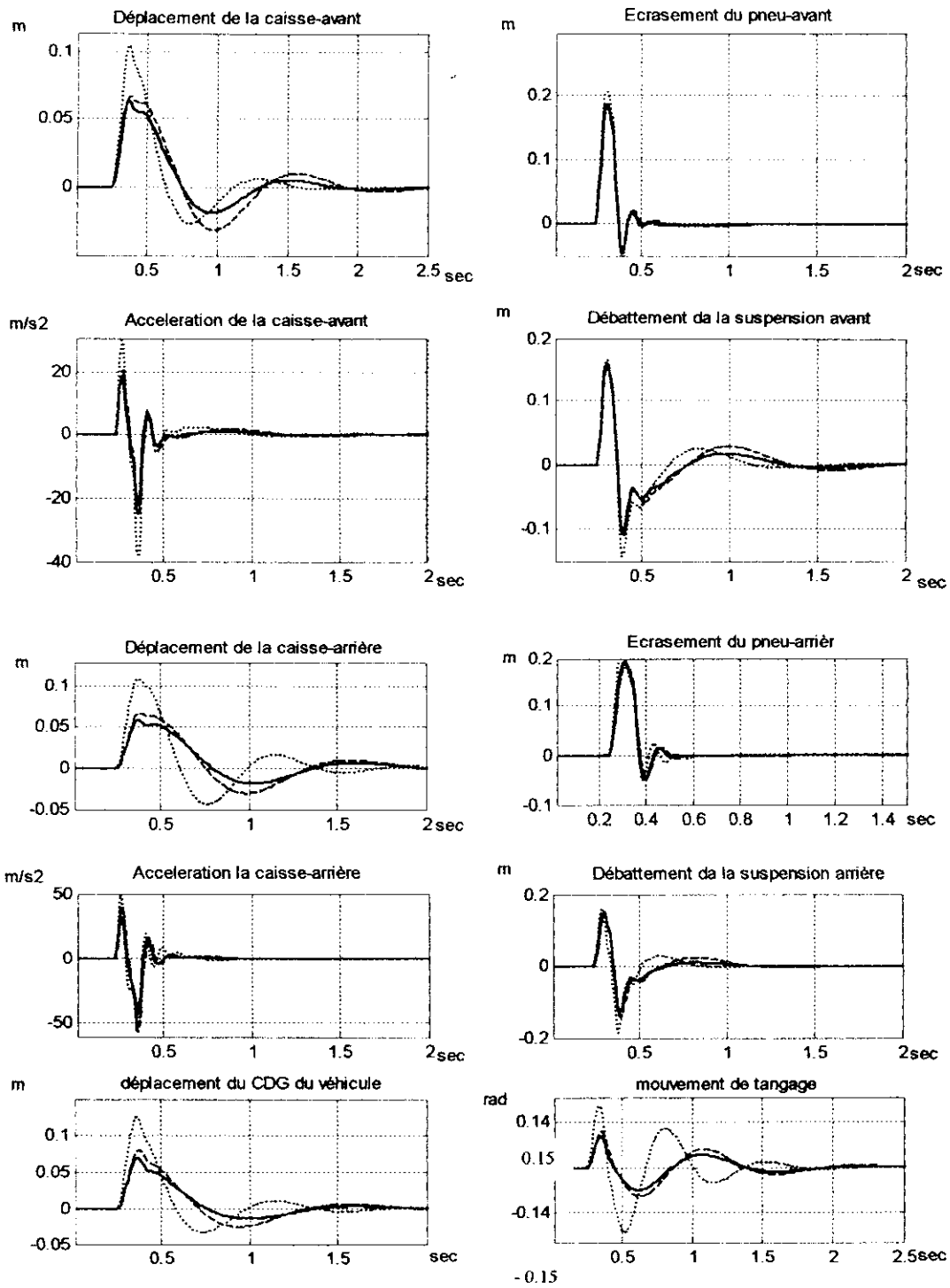


Fig.18 Passage sur une bosse de 15 cm à une vitesse de 40 Km/h

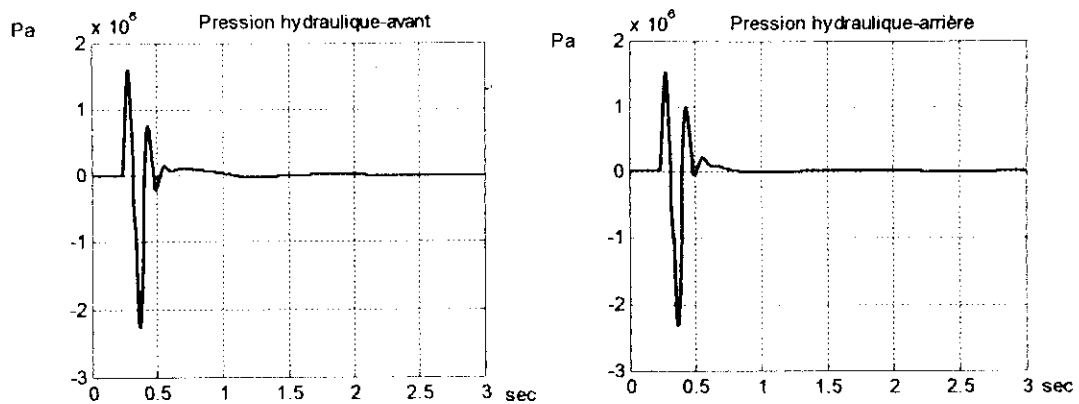


Fig.18. Pression développée pour une bosse de 5 cm.

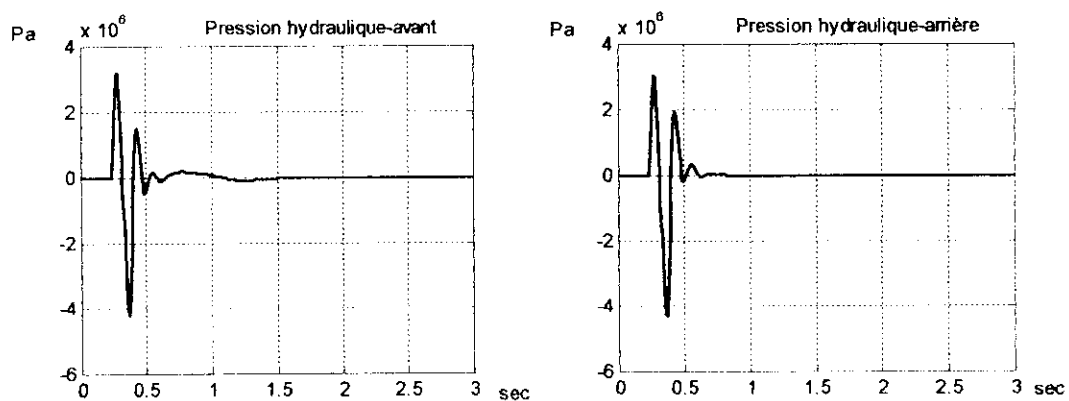


Fig.19. Pression développée pour une bosse de 10 cm.

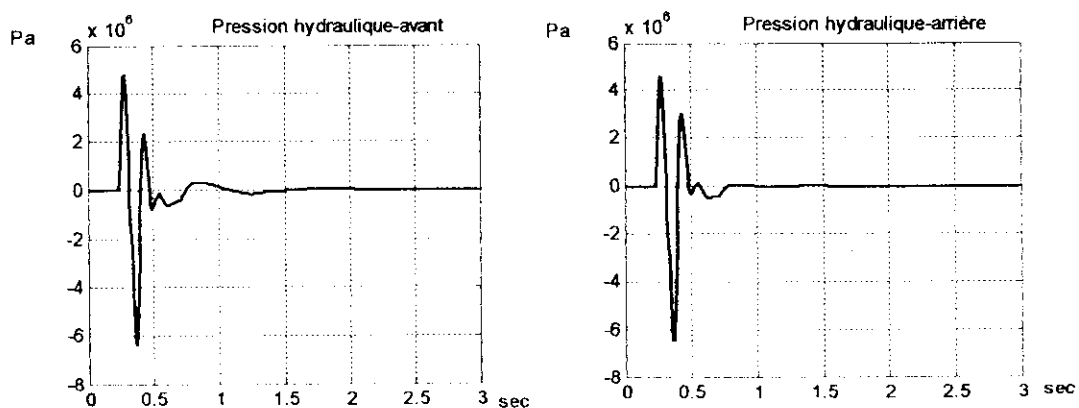


Fig.20. Pression développée pour une bosse de 15cm.

Analyse des résultats :

Les figures (16 à 20) représentent les réponses temporelles des différentes suspensions sur une route inégales. Ces résultats illustrent très nettement l'avantage de la suspension active par rapport à son homologue passive. En effet, les critères de sécurité et du confort sont simultanément satisfaits, chose qui n'est possible dans le cas des suspensions passives.

La comparaison des résultats obtenus avec un correcteur neuronal, à ceux obtenus par les autres approches, est très favorable, et les performances obtenues sont bien meilleurs que les autres correcteurs, ceux-ci démontrent donc, l'intérêt de ces correcteurs très généraux, ainsi que la faisabilité de leur apprentissage.

Conclusion :

Dans cette partie, nous avons synthétisé des lois de commandes pour le contrôle des suspensions de véhicules, en utilisant différentes approches de commande (à savoir l'approche de Backstepping seule, Backstepping avec bouclage non linéaire, et double bouclage avec filtres), pour les différentes modélisations du système (modèles quart et demi véhicule avec et sans pris en compte des dynamiques des actionneurs) et comparé les performances ainsi obtenues, en premier lieu, avec celles des suspensions passives, et en deuxième lieu, avec celles d'une suspension active commandé par un contrôle neuronal. Les résultats obtenus, qui sont représentés par les différentes réponses temporelles, confirment donc, que la suspension active apporte des améliorations constantes par rapport à la suspension passive. Ces résultats confirment aussi, la robustesse des réseaux de neurones, utilisés pour la commande des systèmes, indépendamment des complexités et non linéarités présentes, et démontrent leur intérêt comme des correcteurs généraux et approximateurs universels.

Conclusion générale :

Au cours de ce mémoire, nous avons étudié, sur deux axes fondamentaux, les suspensions actives de véhicules, en particulier leurs modélisations et leurs commandes sur un premier axe, et les réseaux de neurones artificiels, notamment, leurs apprentissages et techniques de commande sur un second axe.

Sur le premier axe, nous avons montré que les suspensions passives étaient limitées, et que l'apport de l'énergie externe, pour la suspension active, améliore substantiellement les performances de celle-ci. Ainsi, pour le cas passif, étant que les suspensions utilisées ici sont dures, l'aspect sécurité du système est amélioré, tant que l'aspect confort est relativement dégradé. Par conséquent, ce type de suspension, ne permet pas une amélioration sur les deux aspects en même temps. La suspension moderne, ou active, améliore substantiellement, l'aspect confort du passager, tout en préservant l'aspect sécurité du véhicule constant.

Cependant, il reste beaucoup de travail à effectuer sur la commande de ces systèmes. En effet, la robustesse en stabilité et en performances vis-à-vis des variations paramétriques peut être étudiée avec les commandes du type adaptatif. De plus les excitations latérales et longitudinales agissant sur la roue (lors d'une accélération ou d'un freinage dans un virage), ne sont pas prises en compte dans notre cas, la considération de ces dynamiques lors de la synthèse de lois de commande améliore certainement les performances du système réel.

Sur le second axe de ce mémoire, nous avons montré, en termes généraux, que les réseaux de neurones reposent, à présent, sur des bases mathématiques solides qui permettent d'envisager leurs capacités d'approximation universelle, à travers les théorèmes cités. Nous avons également, exposé les différents types d'apprentissage et algorithmes associés pour chaque type, et avons montré, à travers les résultats de simulation, le caractère générique de ces algorithmes qui s'applique indépendamment des complexités du système physique. Différentes techniques de commande sont, ensuite, exposées et en particulier la commande supervisée, adaptée dans ce travail.

La robustesse des correcteurs neuronaux, a été aussi démontrée, par les différents résultats de simulation, et comparaison avec d'autres approches de commande. D'autre part, si la résolution de problèmes difficiles nécessite toujours - et nécessitera encore très longtemps -

beaucoup de travail et un éventail étendu de connaissances en statistiques, traitement du signal, automatique, etc., il n'est pas douteux, que les réseaux de neurones peuvent alléger, considérablement la tâche, en permettant une approche efficace et générique des problèmes non linéaire. Bien entendu, leur domaine d'application, ne se limite pas à la modélisation et la commande de processus. Les réseaux de neurones peuvent également être utilisés dans la classification, reconnaissance de formes et bien d'autres domaines. On bénéficie alors de leur grande rapidité de calcul, même exécutés sur des machines traditionnelles telles que des ordinateurs personnels. Toutefois, l'application des réseaux de neurones à l'automatique, peut, toujours, bénéficier des résultats et des méthodes de l'automatique non linéaire classique, en particulier pour la démonstration des propriétés de stabilité et de robustesse des systèmes neuronaux.

Perspectives :

Dans ce travail, nous avons proposé une méthodologie pour l'étude des suspensions, nous pouvons dégager quelques axes de poursuite de ce travail ;

En terme de commande :

Nous avons présenté ici, les techniques, les plus simples d'apprentissage, et de mise en oeuvre des réseaux de neurones, pour la modélisation et la commande de processus, et adapté la technique supervisée, dans ce travail. Il peut être nécessaire, en effet, d'envisager des architectures adaptatives (les coefficients du correcteur, sont alors, ajustés en permanence, pendant son utilisation avec le processus), démarche qui permet, certainement, d'améliorer encore les performances du système.

La démarche hybride, est aussi, d'importance critique. En effet, la 'superposition' d'autres techniques de commandes, comme celles de la logique floue, ou des modes glissant, va nécessairement, améliorer les performances du système, en termes de la stabilité, et de robustesse des lois de commandes.

En termes de modélisation :

Il est intéressant, d'étendre l'étude des modélisations des suspension, à un modèle de véhicule complet, à fin de prendre en compte, les autres mouvement agissant sur le véhicule. La considération des autres dynamiques négligées dans ce travail, telles que les dynamiques latérales et longitudinales du pneu, permet d'aboutir à des modélisations plus réalistes.

Références bibliographiques :

- [1] : JUNG-SHAN LIN & Ioanis KANELLAKOPOULOS ‘ Adaptative nonlinéair control in active suspension ’.
UCLA Electrical Engineering. Los Angeles, CA 90095-1594.
- [2] : JUNG-SHAN LIN & Ioanis KANELLAKOPOULOS ‘ Nonlinéair design of active suspension ’.
UCLA Electrical Engineering. Los Angeles, CA 90095-1594.
- [3] : Supavut CHANTRANUWATHANA & Huei PENG ‘ Pratical adaptative robust controllers for active suspension ’.
Departement of mechanical Engineering and applied mechanics.
University of Michigan, Ann arbor, MI 48109.
- [4] : JUNG-SHAN LIN & Ioanis KANELLAKOPOULOS ‘ Road-Adaptative Nonlinéair design of active suspension ’.
Los Angeles. , Jlin@ee.ucla.edu, ioannis@ee.ucla.edu.
- [5] : J. CAMPOS, L. DAVIS, F. L. LEWIS, S. IKENAGA, S. SCULLY, AND M. EVANS.
‘Active Suspension Control of Ground Vehicle Heave and Pitch Motions ’.
Automation and Robotics Research Institute The University of Texas at Arlington
7300 Jack Newell Blvd. S, Fort Worth, Texas 76118-7115.
- [6] : Antonín STRÍBRSKÝ and Kateřina HYNIOVÁ and Jaroslav HONCŮ and Aleš KRUCZEK. ‘Using fuzzy logic to control active suspension system of one-half-car model ’.
Department of Control Engineering,
Faculty of Electrical Engineering, Czech Technical University
Karlovo namesti. 13, 121 35 Prague, Czech Republic, stibrsk@fel.cvut.cz
- [7] : Wuwei CHEN (Visiting Professor), James K. MILLS and Le WU. ‘ Neurofuzzy Adaptive Control for Semi-Active Vehicle Suspension ’.
Department of Mechanical and Automotive Engineering, Hefei University of Technology, Hefei, Anhui, P. R. China, 230069 Email: wchen@mie.utoronto.ca.
- [8] : Damien SAMMIER ‘ Sur la modélisation et la commande de suspension de véhicules automobiles ’.
Institut National Polytechnique de Grenoble.

- [9] : Jae-Choon , CHON^{*} , Jungwon YOON⁺ , Kyung-Ok KIM^{*} , Young-Kyu YANG^{*} .
 'A New Design of Active Suspension System for Low Energy Consumption and Good Ride Comfort ' .
^{*} : Electronics and Telecommunications Research Institute (ETRI), Korea
⁺ : Department of Mechatronics , KwangJu Institute of Science and Technology(K-JIST), Korea.
- [10] : J. SEONGHARK^{*}, M. DONGHEE and K. JUNGHA^{**} 'Active Control Method of Automotive Suspension System'
^{*} Department of Automotive Engineering, kookmin University, Chongnung-dong, Seongbuk-gu, SEOUL, 861-1 KOREA.
^{**} Department of Automotive Engineering, kookmin University, Chongnung-dong, Seongbuk-gu, SEOUL, 861-1 KOREA.
- [11] : Eric DAVALO & Patrick NAIM ' Der réseaux de neurones ' .
 Eyrolles, 1990.
- [12] : J. HERAULT & C. JUTTEN. ' Réseaux de neurones et traitement de signal ' .
 Traité des nouvelles technologies. Hermes, 1994.
- [13] : Isabelle RIVALS, 'Les réseaux de neurones formels pour le pilotage de robots mobiles ' .
 Laboratoire d'Électronique de l'ESPCI (École Supérieure de Physique et de Chimie Industrielles).
- [14] : ATLAN H. & COHEN I.R ' Theories of Immune Networks', Springer-verlage, Berlin, 1989.
- [15] : I. RIVALS, L. PERSONNAZ, G. DREYFUS. 'Modélisation, classification et commande par réseaux de neurones : Principes fondamentaux, méthodes de conception et illustration industrielles ' .
 Ecole Supérieure de Physique et de Chimie Industrielles de la Ville de Paris
 Laboratoire d'Electronique.
- [16] : I. RIVALS. (1995) ' Modélisation et commande de processus par réseaux de neurones; application au pilotage d'un véhicule autonome ' .
Thèse de Doctorat de l'Université Paris VI.
- [17] : Jean-Michel RENERS. ' Algorithmes génétiques et réseaux de neurones ' .
 Edition HERMIS, Paris 1995.

إن العمل المقدم من خلال هذه المذكرة يتمثل في تأليف قانون قيادة متدلية ميارة باستعمال تقنيات التحكم عن طريق الشبكة العصبية. لهذا الغرض قمنا في أول الأمر باستخراج مختلف النمذجات التي تشرح نظام عمل متدلية ميارة. تعرضنا من خلال هذا العمل إلى النموذجين الخطي و اللخطي للنظامين ربع و نصف الميارة. القسم الثاني من المذكرة كان مخصصا لشرح أنظمة عمل الشبكة العصبية و ذكر مختلف النمط المستعملة و بعض تقنيات التعلم و التحكم التي توفرها مثل هذه الأنظمة. في القسم الثالث و الأخير من هذه المذكرة قمنا باستخراج أربعة أنظمة تحكم لكل من النموذجين الخطي و اللخطي للنمطين ربع و نصف ميارة باستعمال طريقيتي البلاكستيبينق و رجوع الحالة المرشح. ثم تركيب هذه القوانين باستعمال شبكة عصبية مناسبة.

Le travail présenté dans ce mémoire, concerne la synthèse d'une loi de commande neuronale, assurant la stabilité en boucle fermée de la suspension active d'un véhicule. Pour se faire, on a commencé par la présentation des différentes modélisations décrivant le fonctionnement d'une suspension automobile. On a cité les deux types de modélisation, à savoir la modélisation linéaire et non linéaire, pour les système quart et demi véhicule.

Le deuxième axe de notre travail est dédié à l'étude des réseaux de neurones. On avait donc, cité les différents types de réseaux existant dans la littérature, les divers types et algorithmes d'apprentissage neuronal, et quelque méthode de la commande par réseaux de neurones.

En troisième et dernière partie, nous avons synthétisé quatre lois de commande pour les quatre modèles cités, en utilisant l'approche de backstepping et des bouclages filtrés pour les modèles quart et demi véhicule respectivement. Les lois de commande ainsi synthétisées, sont supervisées par un réseau de neurones convenable. Pour tous les cas, des testes de simulation ont été faits à fin de comparer les performances.

The work presented in this memory, relates to the synthesis of a law of neuronal order, ensuring stability in closed loop of the active suspension of a vehicle. To be done, one started with the presentation of various modelings describing the operation of an automobile suspension. One quoted the two types of modelling, namely modelling linear and nonlinear, for the system quarter and half conveys.

The second axis of our work is dedicated to the study of the neurone networks. One had thus quoted the various types of networks existing in the literature, the various types and algorithms of neuronal training, and some method of the order by neurons networks.

In third and last part, we synthesized four laws of order for the four quoted models, by using the approach of back stepping and of the looping filtered for the models quarter and half conveys respectively. A suitable neurons network supervises the laws of order thus synthesized.