

11/95

RÉPUBLIQUE ALGÉRIENNE DÉMOCRATIQUE ET POPULAIRE

MINISTÈRE DE L'ENSEIGNEMENT SUPÉRIEUR ET DE LA  
RECHERCHE SCIENTIFIQUE

ÉCOLE NATIONALE POLYTECHNIQUE

DÉPARTEMENT  
OPTION

المدرسة الوطنية المتعددة التقنيات  
المكتبة —  
BIBLIOTHEQUE —  
Ecole Nationale Polytechnique

GÉNIE ÉLECTRIQUE  
AUTOMATIQUE

PROJET DE FIN D'ÉTUDES

THÈME

ÉTUDE DE LA COMMANDE PAR  
RÉSEAUX DE NEURONES  
APPLICATION A UN  
TRAIN DE LAMINOIR

DIRIGÉ PAR  
R.ILLOUL

ÉTUDIÉ PAR  
M.DAMA  
A.RABHI

PROMOTION  
JUN 95

E.N.P 10, Avenue Hacén Badi El-harrach. ALGER

**RÉPUBLIQUE ALGÉRIENNE DÉMOCRATIQUE ET POPULAIRE**

**MINISTÈRE DE L'ENSEIGNEMENT SUPÉRIEUR ET DE LA  
RECHERCHE SCIENTIFIQUE**

**ÉCOLE NATIONALE POLYTECHNIQUE**

**DÉPARTEMENT  
OPTION**

المدرسة الوطنية المتعددة التقنيات  
المكتبة — BIBLIOTHEQUE  
Ecole Nationale Polytechnique

**GÉNIE ÉLECTRIQUE  
AUTOMATIQUE**

**PROJET DE FIN D'ÉTUDES**

**THÈME**

**ÉTUDE DE LA COMMANDE PAR  
RÉSEAUX DE NEURONES  
APPLICATION A UN  
TRAIN DE LAMINOIR**

**DIRIGÉ PAR  
R.ILLOUL**

**ÉTUDIÉ PAR  
M.DAMA  
A.RABHI**

**PROMOTION  
JUN 95**

**E.N.P 10, Avenue Hacem Badi El-harrach. ALGER**

## Remerciements

Nous tenons à exprimer nos vifs remerciements à notre promoteur Monsieur R.ILLOUL qui a dirigé ce travail et nous a prodigué de nombreux conseils.

Nous voudrions également exprimer notre gratitude à tous les enseignants de l'Ecole Nationale Polytechnique et en particulier ceux du département de Génie Electrique.

## Sommaire

<b>Introduction générale</b> .....	1
<b>Chapitre I Réseaux de neurones</b>	
1 Introduction .....	3
2 Le neurone artificiel .....	4
3 Architecture du réseau .....	6
4 Les propriétés des réseaux de neurones .....	8
5. Apprentissage .....	10
6. Optimisation aléatoire MOA.....	11
7. Rétro-propagation BP .....	12
<b>Chapitre II Back-propagation</b>	
1. Introduction .....	13
2. Principe .....	13
3. Présentation de l'algorithme .....	14
4. Problèmes posés par l'algorithme .....	17
5. Considérations pratiques .....	18
<b>Chapitre III Commande neuronal des systèmes</b>	
1. Introduction .....	20
2. Principe fondamental .....	20
3. Les structures de commande .....	21
4. Conclusion .....	29
<b>Chapitre IV Train de laminoir</b>	
1. Introduction .....	30
2. Les différents types de laminoirs .....	30
3. Conditions que doit remplir un laminoir .....	33

4. Le tandem quatre à cages .....	35
5. Modèle mathématique .....	37

## Chapitre V      **Simulation**

1. Introduction .....	41
2. Régulation d'épaisseur .....	41
3. Commande anticipatrice .....	43
4. Commande par réseaux de neurones .....	50
5. Test de robustesse .....	69

<b>Conclusion générale</b> .....	<b>75</b>
----------------------------------	-----------



## *Chapitre 0*

### **Introduction générale**

## Introduction générale

Actuellement l'industrie, fait face au problème de contrôle des systèmes non linéaires de plus en plus complexes, qui présentent des modèles mathématiques inconnus ou difficile à introduire. Les techniques utilisées ne satisfaisaient pas les exigences pointus de l'industrie, on remarque d'ailleurs que la linearisation autour d'un point de fonctionnement et la commande adaptative sont délicates à établir et parfois même inapplicables.

Cet échec, a donné lieu à l'immergence d'une nouvelle technique qui constitue le moyen efficace pour surmonter ces problèmes, cette technique est la commande par réseaux de neurones, qui est le fruit de longues recherches, dans le but de l'affiner et de la rendre apte à acquérir les avantages suivants :

- L'activité parallèle en temps réel de nombreux composants.
- La présentation distribuée des connaissances.
- L'apprentissage par modification des poids.
- La connaissance du modèle du système à commander n'est pas nécessaire.

Dans notre travail nous avons introduit cette commande pour réguler l'épaisseur à la sortie d'un train de laminoir, en utilisant plusieurs structures de commande.

Nous avons jugé utile d'étaler notre étude sur cinq chapitres, qui traduisant notre démarche logique dans le traitement de notre sujet d'investigation.

En effet, nous avons consacré le premier chapitre à la présentation des réseaux de neurones et leur classification en introduisant la notion d'apprentissage.

Dans le deuxième chapitre, l'accent est mis sur l'algorithme d'apprentissage de retropropagation qui sera utilisé ultérieurement.

Les structures de commande par les réseaux de neurones sont exposées dans le troisième chapitre, où les techniques Feedforward, feedback et la commande à modèle interne sont étudiées.

Dans le quatrième chapitre, nous avons étudié le processus industriel ( le laminoir ), en considérant son fonctionnement, son modèle mathématique ainsi que ses différents types.

Le cinquième chapitre, constitue la synthèse entre les structures mentionnées dans le chapitre III et ses applications dans le processus envisager, dans le sens où on y simule les réponses du systèmes et on donne l'interprétation des résultats obtenus.

On achèvera notre étude, par la conclusion générale qui constitue le couronnement de notre travail.



*Chapitre I*

**Réseaux de neurones**

## 1 Introduction

Les réseaux de neurones sont apparus au début des années quarante dans une tentative de concevoir un processeur qui simule le fonctionnement du cerveau humain.

Dans ce sens, **MC CULLOCH & PITTS** ont établi un modèle de neurone biologique appelé " **Neurone formel** ", en suite **ROSENBLATT** a développé le modèle pour aboutir au " **Perceptron** " qui a été utilisé par **WIDROW** pour construire la machine " **Adaline** " en se basant sur la capacité d'interconnexion de plusieurs éléments.

Après plusieurs années de développement le travail des chercheurs a abouti à ce qu'on appelle les réseaux de neurones artificiels (**ANN**), qui sont de plus en plus utilisés par l'automaticien dans l'identification et la commande, surtout pour les systèmes non linéaires ou ceux à modèles inconnus ou difficilement modelisables, et cela grâce à leurs propriétés de :

- Activités parallèles et en temps réel de plusieurs composants.
- Représentation distribuée des connaissances .
- L'apprentissage par modification des connexions .

Une définition résume tous ce qu'on a dit sur les réseaux de neurones donnée par **HECHT NELSEN** est la suivante :

“ Un réseau de neurones est une structure de traitement d'informations parallèles et distribuées, constituée d'unités de calcul interconnectées par des canaux unidirectionnels appelés “ connexions ”. Chaque unité de traitement n'a qu'une seule connexion de sortie, qui peut être dupliquée en autant d'exemplaires que désiré .

Les traitements effectués par chaque unité peuvent être définies de manière arbitraire, pourvu qu'ils soient complètement locaux.”

## 2 Le neurone artificiel [2]

Le neurone formel de **MC CULLOCH & PITTS** est un automate à seuil, qui reçoit  $n$  signaux  $X_i$  à son entrée, et dont la sortie  $Y$  est obtenue en appliquant une fonction seuil, à la combinaison linéaire entre les entrées  $X_i$  et les poids des connexions qui lient chaque signal au neurone d'entrée.

$$Y = \begin{cases} 1 & \text{si } \sum_{i=1}^n W_i X_i \geq \theta \\ 0 & \text{si non} \end{cases}$$

$\theta$  : le seuil

Plus généralement , on définit un neurone artificiel un automate linéaire ou quasi linéaire comme étant l'application d'une fonction  $F$  dite fonction d'activation de sortie ou de transition ou produit scalaire entre le vecteur d'entrées  $X=(1,X_1,X_2,\dots,X_n)$  et vecteur des poids  $W=(W_0,W_1,W_2,\dots,W_n)$  .

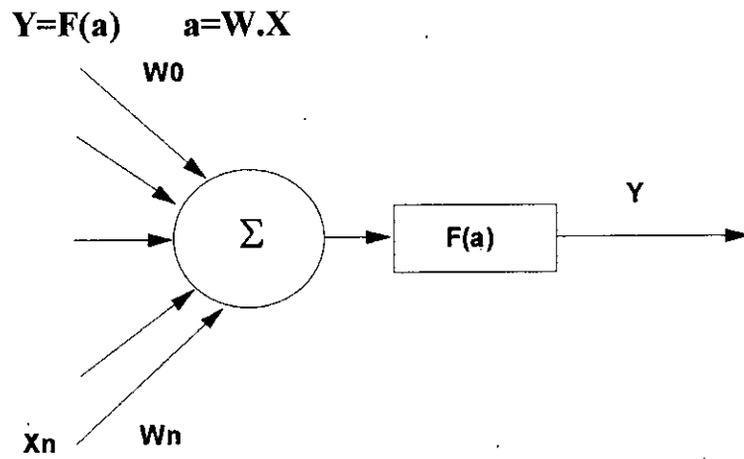


Fig (I.1) Schéma d'un neurone artificiel

La fonction d'activation de sortie P peut être à :

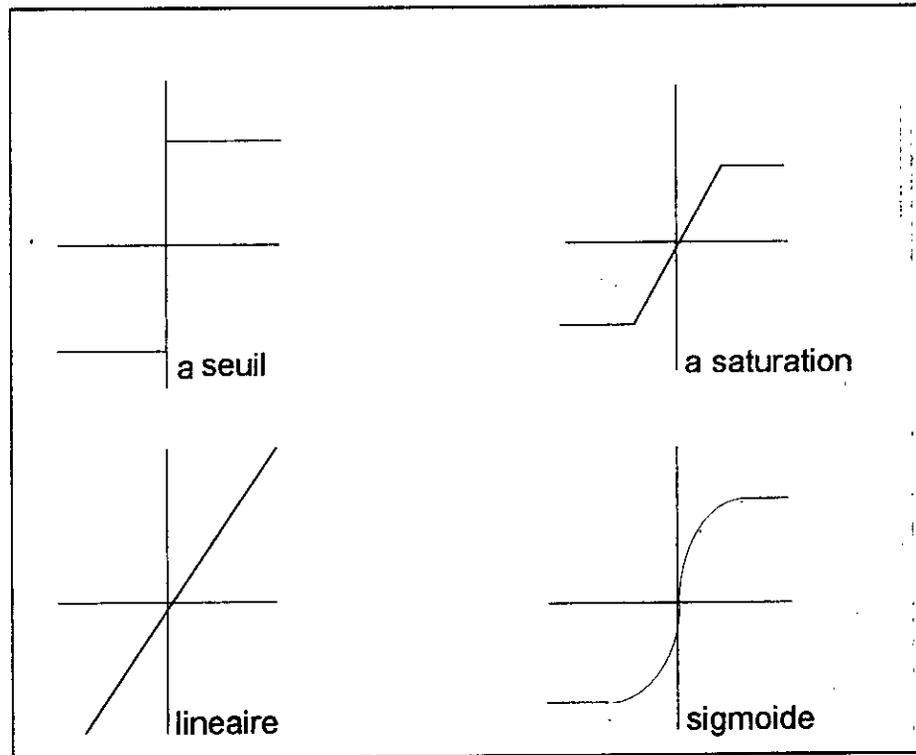


Fig (I.2) Fonctions d'activation de sortie

*Remarque*

La fonction d'activation  $F$  est saturable pour éviter des valeurs de sortie trop élevées qui peuvent déstabiliser le réseau .

### 3 Architecture du réseau [1]

L'architecture du réseau spécifie :

- Le nombre de couches
- La nature des couches (leur fonction d'activation de sortie )
- Le graphe d'interconnexion des couches et des neurones dans la même couche .

On distingue deux grands types de réseaux :

- les réseaux dynamiques
- les réseaux à couches .

#### 3.1 Les réseaux dynamiques

Appelés aussi les réseaux entièrement connectés leurs architecture à la forme suivante :

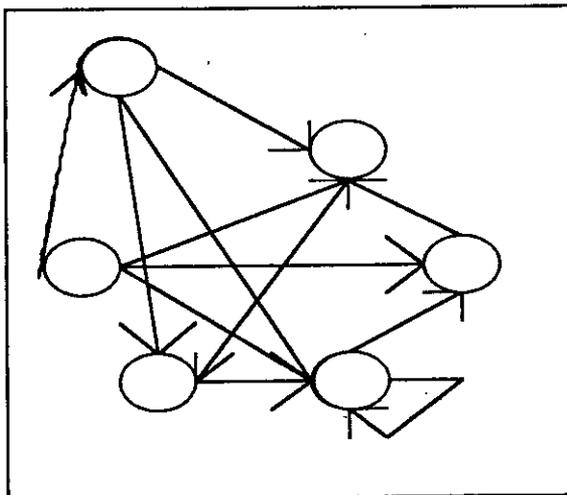


Fig (I.3) Réseau Hopfield

Chaque neurone est relié presque à tous les autres ,neurones , même la sortie dépend d'elle même ( possède un retour ).

Ces réseaux sont utilisés surtout pour la reconnaissance des formes, l'identification des systèmes dynamiques et les systèmes variant dans le temps .

le modèle dynamique le plus connu est celui de **HOPFIELD**.

### **3.2 les réseaux à couches**

Ce type de réseaux est utilisé dans la modélisation, le contrôle et l'identification des systèmes.

Ils sont formés de plusieurs couches, de plusieurs neurones chacune, tel que chaque neurone possède plusieurs entrées qui ne peuvent recevoir les signaux que des couches précédentes, et une seule sortie qui transmet le signal de sortie aux couches qui viennent après.

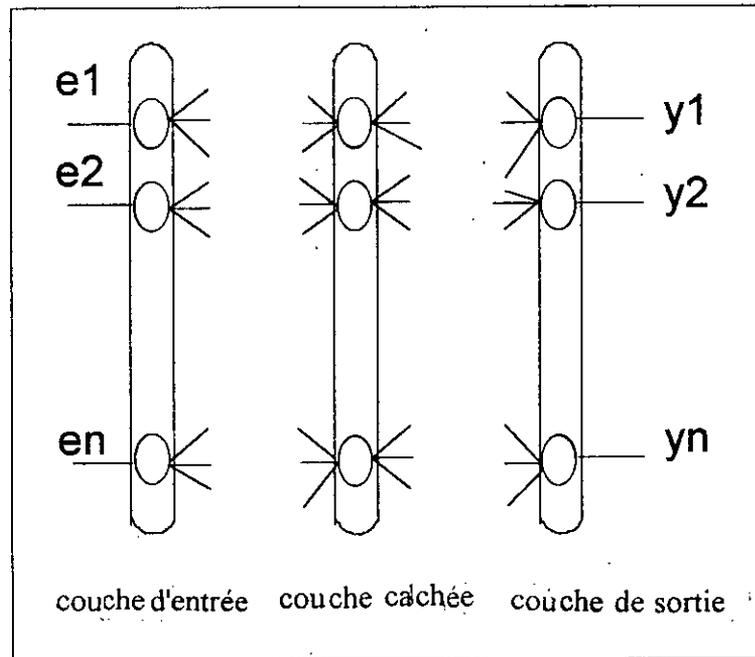


Fig (I.4) Réseau à couches

## 4 Les propriétés des réseaux de neurones [2]

L'intérêt porté aujourd'hui aux réseaux de neurones tient sa justification des propriétés qu'ils possèdent et qui permettent de dépasser les limitations de l'informatique traditionnelle parmi ses propriétés :

### 4.1 Le parallélisme

Un réseau de neurone est un ensemble d'entités élémentaires qui transitent simultanément, l'intérêt d'une telle conception de traitement des données a été mise en évidence, par l'échec des méthodes séquentielles pour le traitement des problèmes qui nécessitent des quantités énormes de données aboutissaient à une explosion combinatoire.

Le parallélisme permet une rapidité de calcul supérieure mais exige de penser et de poser différemment les problèmes à résoudre, par l'étude du fonctionnement des réseaux de formalisation des problèmes qui permettrait de les traiter en parallèle.

#### **4.2. La Capacité d'Application**

Est la capacité d'apprentissage qui permet au réseau de tenir compte des nouvelles données ou informations supplémentaires sur le système.

Cette capacité présente un intérêt déterminant pour tous les problèmes évolutifs; pour les résoudre il faut tenir compte des situations non encore connues.

#### **4.3. Domaines d'Application**

Les caractéristiques des problèmes adoptés ont une résolution par les réseaux de neurones.

- 1- l'inexistence de solution technologique courantes
- 2- les problèmes qui nécessitent une grande rapidité de traitement
- 3- les problèmes qui font intervenir les données bruitées
- 4- un système très difficile à le formuler
- 5- les systèmes non linéaires ou inconnus

Si on les cite suivant les domaines d'application privilégiés on les ordonne comme suite :

- *Reconnaissance des formes.*
- *Traitement de signal.*
- *Vision, parole.*
- *Prévision et modélisation.*
- *Aide à la décision.*
- *Robotique.*

Pour la modélisation et la commande des systèmes, les réseaux les plus adaptés sont les réseaux à couches. Pour cela, nous présentons l'apprentissage des réseaux de neurones .

## **5.Apprentissage [1] [10]**

L'apprentissage , pour les biologistes est l'évolution du réseau nerveux et cela est le résultat de l'interaction entre l'environnement, et le programme génétique .

Pour les réseaux de neurones artificiels l'apprentissage consiste à modifier les poids reliant les neurones , pour arriver à l'état désiré en minimisant l'erreur entre la sortie du réseau et la sortie désiré .

il existe deux modes d'apprentissage :

### **5.1.Apprentissage Supervisé**

Il est effectué en présentant au réseau des couples d'entrées/sorties, du système réel qui sont appelé vecteur d'entraînement.

Dans le but de minimiser l'erreur entre la réponse du réseau et la sortie du système réel pour tous les couples d'entraînement sur les poids .

## 5.2.Apprentissage non supervisé

On présente au réseau des informations groupées , sous forme de classes , à l'intérieur desquelles les données présentent des caractéristiques connus de l'apprentissage est base sur la méthode de minimisation des fonctions à plusieurs ,variables , il y'a plusieurs techniques de changements des poids , deux algorithmes seront présentés dans ce qui suit :

- La rétro propagation
- L'optimisation aléatoire

## 6.Optimisation aléatoire MOA

L'optimisation aléatoire de **MATYAS**, assure, une convergence vers un minimum globale, avec une probabilité égale à 1 .

Cette méthode est utilisée lorsque la dimension du réseau devient importante , matyas utilise dans sa méthode la théorie d'optimisation aléatoire d'une fonction  $f(x)$  .

cette technique comporte les étapes suivantes :

1- initialiser le vecteur  $x$  aléatoirement à  $x(0)$ , mettre  $k=0$  et fixer le nombre maximal d'itération  $m$ ,  $b(0)$  , donner une valeur à  $V$  .

2- génère un vecteur aléatoire gaussien  $\varepsilon(k)$  de moyenne  $b(k)$  et de variance  $V$  .

3- Si  $F(x(k)+\varepsilon(k)) < F(x(k))$  alors

$$x(k+1) = x(k) + \varepsilon(k) \text{ et}$$

$$F(x(k)+\varepsilon(k)) > F(x(k)) \text{ et}$$

$$F(x(k)-\varepsilon(k)) < F(x(k)) .$$

$$\text{alors } x(k+1) = x(k) - \varepsilon(k) \quad \text{et} \quad b(k+1) = b(k) - 0.4\varepsilon(k) .$$

$$\text{sinon } x(k+1) = x(k) \text{ et } b(k+1) = 0.5 b(k) .$$

si  $k=m$  stopper les calculs sinon aller à 2 .

## 7. Rétro-propagation BP

Cette technique est la plus utilisée, elle repose sur la méthode dite "*Descente du Gradient*", cet algorithme réajuste les poids en propageant l'erreur à travers les couches du réseau .

Elle est traitée dans le chapitre suivant et c'est telle qu'on va utiliser au cours de notre travail .



## *Chapitre II*

# **Back-Propagation**

## 1. Introduction

La méthode de back-propagation est la plus connue des méthodes d'apprentissage supervisée, elle est basée sur le calcul de la dérivée partielle d'une fonction multi-variables, lorsque toutes les variables varient.

Ce problème a été résolu en 1974 par **WERBOS**, de l'université de HARVARD, dans sa thèse de doctorat, mais malheureusement ce travail est resté presque inconnu de la communauté scientifique.

En 1982 **PARKER** a retrouvé de nouveau la solution du problème, et il a utilisé la technique de la minimisation de l'erreur de sortie d'un réseau de neurone et qui est appelé par la suite back-propagation.

Cette méthode est publiée en avril 1985 par le MIT.

Au début de l'année 1986, **RUMELHART, HINTON ET WILLIAMS** ont présenté cette méthode d'une façon claire et précise.

Cette méthode est applicable seulement pour les réseaux multicouches.

## 2. Principe [1] [2] [12]

On transmet des données au réseau en forçant le niveau d'activation des neurones de la première couche.

Les signaux se propagent ensuite successivement de couche en couche jusqu'à la sortie, les activations des neurones de cette dernière couche constituent la sortie du réseau, elle est comparée avec la sortie désirée et l'erreur calculée détermine les nouveaux poids de la dernière couche avec la couche d'entrée, une fois les poids modifiés on essaie une autre fois et on calcule l'erreur jusqu'au moment où on obtient une erreur suffisamment petite.

### 3. Présentation de l'algorithme

Notre but est de déterminer les poids synaptiques qui définissent les connexions entre les neurones pour produire la sortie désirée avec une certaine tolérance.

Avant de commencer l'algorithme d'apprentissage, il faut tout d'abord initialiser aléatoirement les poids, puis suivre les étapes suivantes

#### PHASE AVANT

- 1- Appliquer les vecteurs d'entrée X
- 2- Calculer les entrées des neurones des couches cachées

$$Net_{pj} = \sum_{i=1}^n W_{ji} X_{pi} + \theta_j$$

p : le p<sup>eme</sup> exemple

$\theta_j$  : le biais

- 3- Calculer les sorties des neurones des couches cachées

$$I_{pj} = F_j(Net_{pj})$$

F : fonction d'activation de sortie.

4- calculer les entrées des neurones de sortie .

$$Net_{pk}^0 = \sum_{j=1}^n W_{kj} I_{pj} + \theta_k$$

$^0$  : output

5- calculer la sortie du réseau

$$out_{pk} = F(Net_{pk}^0)$$

#### PHASE ARRIERE

6- calculer le terme d'erreurs pour la couche de sortie et il existe deux approches :

La première est de calculer l'erreur commise sur un passage complet de l'ensemble des exemples .

$$E = \sum_{p=1}^T E(t) = \sum_{p=1}^T \sum_{k=1}^m \frac{1}{2} (out_{pk} - Y_{pk})^2$$

La deuxième est de calculer l'erreur commise sur un passage d'un seul exemple .

$$E = \sum_{k=1}^m \frac{1}{2} (out_{pk} - Y_{pk})^2$$

Et le terme d'erreur pour les poids pour les deux approches est respectivement.

$$\delta_{pk}^0 = \sum_{p=1}^T (y_{pk} - out_{pk}) F'(Net_{pk}^0)$$

$$\delta_{pk}^0 = (Y_{pk} - out_{pk}) F'(Net_{pk}^0)$$

$F'$  : est la dérivée de la fonction  $F$  .

7- Modification des poids de la couche de sortie .

$$\text{New } W_{kj} = W_{kj} + \mu \delta^o_{pk} I_{pj}$$

$n$  : le pas d'apprentissage .

8 Calcul le terme d'erreur des couches cachées

$$\delta_{pj} = F'(Net_{pj}) \sum_{k=1}^n \delta^o_{pk} W_{kj}$$

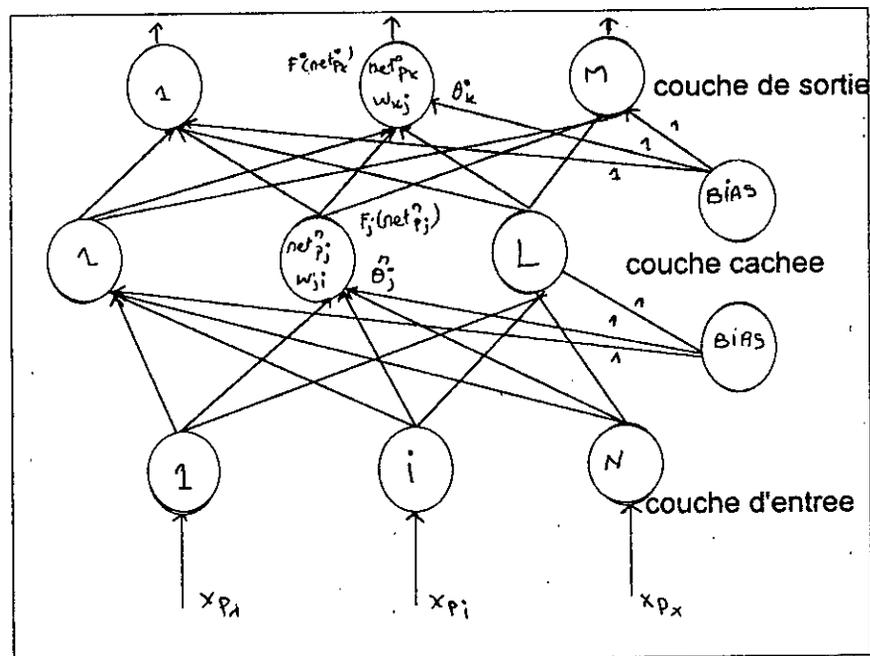
9- Modification des poids des couches cachées.

$$\text{New } W_{ji} = W_{ji} + \mu \delta_{pj} X_{pi}$$

10- Si on est dans la deuxième approche , il faut refaire les étapes précédentes jusqu'à  $E \leq E_0$  ( $E_0$  erreur acceptable ) pour chaque exemple , mais pour la première approche jusqu'à ce que l'erreur globale soit  $\leq E_0$  .

La pratique a montrée que la première approche converge plus vite que la deuxième.

La figure suivante vous permet de bien comprendre les étapes .



Une autre version ajoute un terme en plus a formule de correction des poids. Ce terme est appelé " momentum " et a pour avantage d'accélérer la convergence vers le minimum global .

$$\text{New } W_{ij} = W_{ij} + \Delta W_{ij}$$

$$\text{avec } \Delta W_{ij} = n \delta_{pj} X_{pj} + mm \Delta W_{ij}$$

#### **4.Problemes poses par l'algorithme [2] [4]**

Bien que cet algorithme ait prouvé son efficacité , il présente néanmoins quelques problèmes :

##### **4.1.Architecture du réseau**

Il n'existe pas de règles générales pour déterminer la structure du réseau , c'est-à-dire le nombre de couche cachée et le nombre de neurones dans chaque couche cachée, sachant que le problème critique pendant l'apprentissage est de trouver un réseau assez large pour apprendre plus, mais assez petit pour pouvoir généraliser .

##### **4.2.Temps de calcul**

En augmentant le nombre de couples d'apprentissage le temps de calcul augmente, ce qui diminue la vitesse de convergence .

##### **4.3.Pas de correction**

Un pas trop grand conduit le réseau à osciller, un pas trop petit augmente le temps d'apprentissage .

#### 4.4. Minimum local

Du fait que cette méthode utilise la descente du gradient, la recherche d'un minimum global de la surface d'erreur, dans le domaine des poids peut présenter un problème, si cette surface possède des minimum locaux qui peuvent emprisonner l'algorithme .

#### 4.5. Les poids initiaux

Si les poids initiaux sont grands , la sortie devient élever et se rapproche de la zone de saturation .

pour résoudre ces problèmes, il faut prendre en compte les considérations pratiques .

### 5. Considerations pratiques

#### 5.1. Dimension du réseau

Le nombre de sorties et entres du réseau , sont ceux du système et sont aussi le nombre de neurones dans la couche de sortie et d'entrée .

le nombre des neurones dans la couche cachée est obtenu par le calcul suivant :

$h$  : le nombre minimal de neurones dans la couche cachée

$$h = \frac{case}{10(n+m)}$$

case : représente le nombre de ligne ou colonnes dans le fichier d'exemple .

$n$  : nombre d'entrée et  $m$  : nombre de sortie .

En ce qui concerne le nombre de couche cachée, une règle dont son énoncé est “ le nombre de couches cachées est égale au nombre de relations qui existent entre les neurones de la couche d'entrée et les neurones de la couche de sortie ” .

Ces règles ne sont basées que sur des faits pratiques, l'approche réponde est d'utiliser le minimum de couches et le minimum de neurones dans les couches cachées .

### **5.2.Pas de correction**

En général , il est choisi de tel façon à ce qu'il décroît lentement , mais pas trop ou rapidement mais pas trop, et il doit vérifier la condition .

### **5.3.Poids initiaux**

les poids doivent être initialisés aléatoirement à des petites valeurs comprise entre -0.5 et 0.5 .



*Chapitre III*

**Commande neuronale  
des systèmes**

## 1.Introduction

Depuis leur apparition , les réseaux de neurones et en particulier les réseaux à couches ont été appliquées dans plusieurs domaines , vue leur capacité de commander les systèmes non linéaires, et les systèmes dont les modèles sont peu connus, ou ayant des paramètres variables dans le temps, et de plus ils présentent une bonne résistance au bruit en entrée .

Plusieurs manières existent pour intégrer le réseau de neurone dans la commande des systèmes , cela dépend essentiellement de la structure de réglage et de l'apprentissage choisi .

Le but de ce chapitre est de présenter les différentes structures d'apprentissage des réseaux à couches .

Les inconvénients qu'elles présentent et les techniques utilisées pour répondre à ces insuffisances , ainsi que les différentes théories de commande utilisées par la suite .

## 2.Principe fondamental [1],[2],[3]

La commande par réseau de neurones consiste essentiellement à remplacer le régulateur par un réseau de neurones, ce dernier préalablement entraîné va synthétiser la commande appropriée au système à commander .

L'entraînement du réseau (**contrôleur**) dépend de la structure du réglage utilisée .

### 3. Les structures de commande [3],[9]

Avant d'étudier les structures d'apprentissage, nous présentons les différentes structures de commande ( voir les figures (III.1, III.2, III.3) ).

- Le contrôleur **FEEDFORWARD** : qui fournit une commande à partir du signal de consigne .
- Le contrôleur **FEEDBACK** : qui fournit une commande à partir du signal d'erreur entre la consigne et la sortie du système .
- La structure de commande à modèle interne qui fournit la commande à partir du signal de consigne modifié .

#### 3.1. Contrôleur FEEDFORWARD

Ce contrôleur est appelé " *contrôleur à modélisation inverse* " car, il identifie le modèle inverse du système à contrôler. c'est-à-dire qu'il fournit la commande en utilisant la consigne, il est entraîné de deux façons différentes :

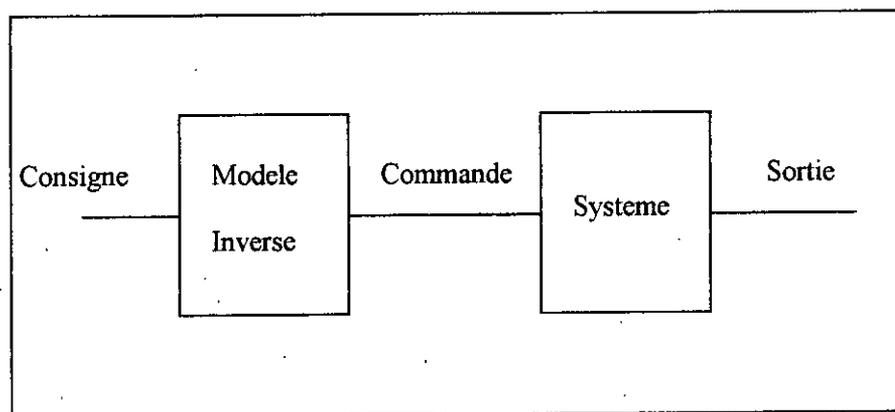


Fig (III.1) Contrôleur Feedforward

a- Apprentissage généralisé (OFF-line).

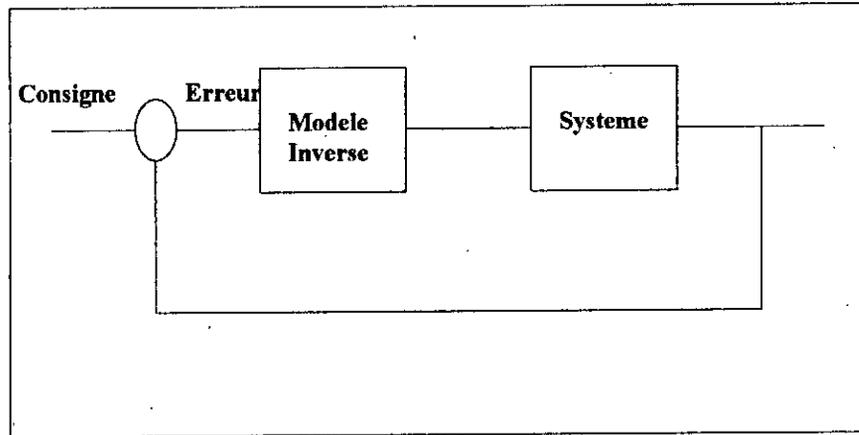


Fig (III.2) Contrôleur Feedback

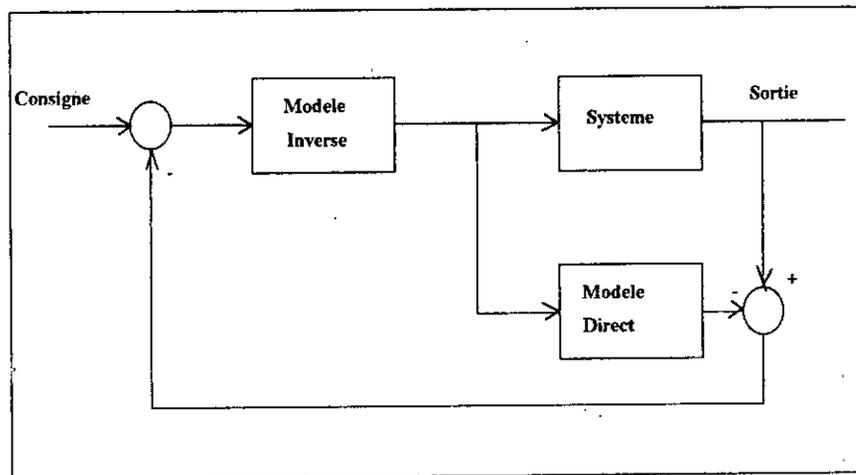


Fig (III.3) Contrôleur à modèle interne

Nommé aussi modélisation inverse direct . Dans cette structure, la sortie du système excitée par un signal de commande synthétique est appliquée au réseau qui délivre une commande qui est comparée à la commande synthétique et l'erreur est utilisée dans l'apprentissage du réseau .

**b- Apprentissage spécialisé (ON line).**

Dans cette structure, le réseau précède le système et reçoit le signal de consigne, génère une commande qui est appliquée au système, dont la sortie est comparée à la consigne et l'erreur donnée est utilisée dans l'algorithme d'apprentissage comme le montre la figure (III.6).

La différence entre cette structure et la structure précédente, est que celle là utilise la consigne comme entrée, et le modèle inverse obtenu est plus fiable en supposant qu'il n'a pas de bruit.

Ses inconvénients sont dans le fait que :

$$E = \frac{1}{2} (yk - ck) \Rightarrow \frac{\delta E}{\delta W} = \frac{\delta E}{\delta \mu} \cdot \frac{\delta Y}{\delta \mu} \cdot \frac{\delta \mu}{\delta w}$$

$$\frac{\delta E}{\delta W} = (y - c)y'(u) \cdot \frac{\delta \mu}{\delta w}$$

qui nécessite la connaissance de  $Y'(u)$ , ce qui n'est pas toujours possible.

- Au début de l'apprentissage, la commande est aléatoire ( dépend des valeurs initiés des poids ) ce qui peut entraîner l'instabilité du système.

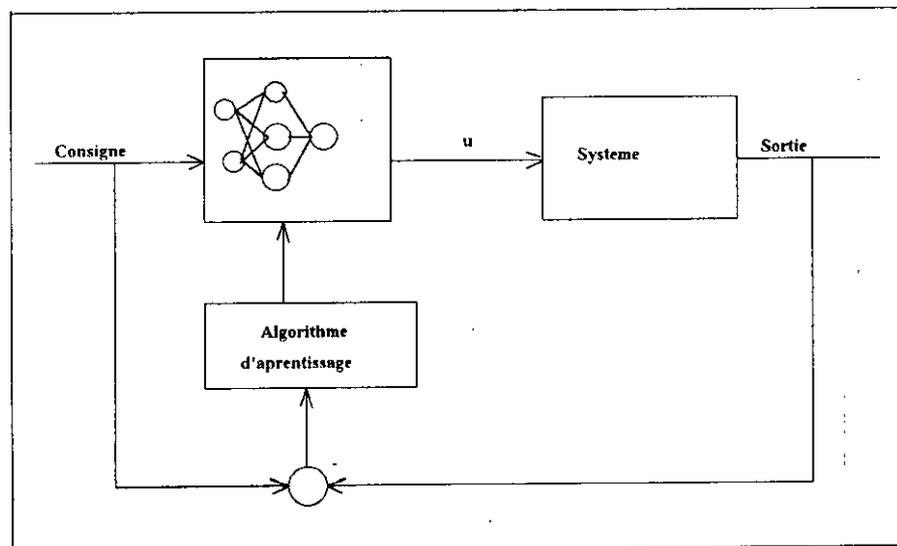


Fig (III.6) Apprentissage spécialisé

### 3.3. Contrôleur FEEDBACK

Ce contrôleur est utilisé surtout comme contrôleur dynamique des systèmes linéaires et est caractérisé par sa robustesse.

Son utilisation est parallèle à celle des régulateurs connus dans la commande des systèmes linéaires, c'est-à-dire son entrée et le signal d'erreur ( consigne - mesure ) ou la sortie d'un régulateur classique PI et son signal de sortie est la commande à appliquer au système (voir figure(III.7) ).

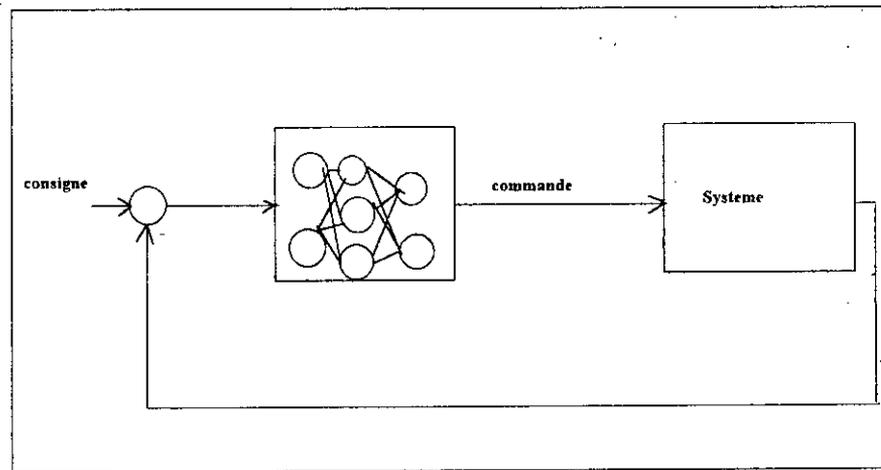


Fig (III.7.a) Contrôleur Feedback

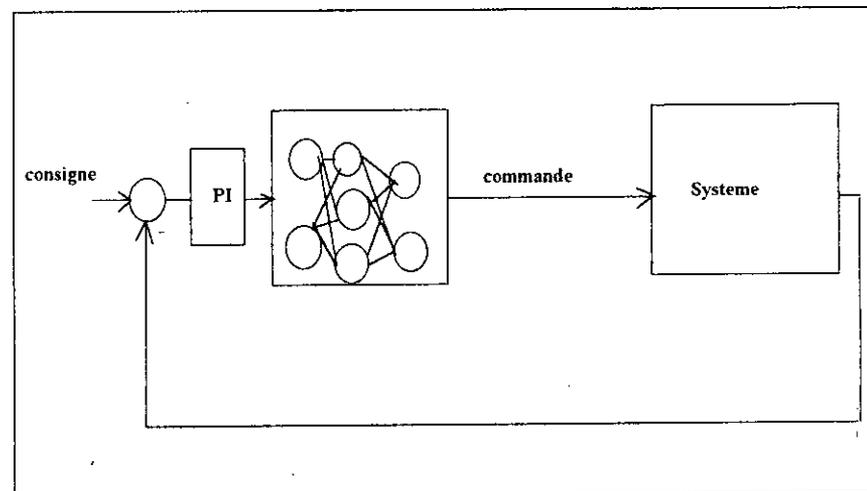


Fig (III.7.b) Contrôleur Feedback avec PI

La différence entre ces deux structures réside dans l'entraînement des réseaux, le premier est entraîné par l'erreur ( consigne - mesure ) et le deuxième par la sortie du système .

### 3.3. Commande à modèle interne (IMC)

Le principe de base de cette commande est de réaliser une commande du modèle et d'appliquer en même temps au procédé ( système) les actions ainsi obtenues pour le modèle .

Les structures varient avec la façon de déterminer la commande du modèle et éventuellement avec la manière de tenir compte des écarts entre la procédé et son modèle .

Le modèle qui est obtenu avec une identification du système à commander, par les réseaux de neurones .

La figure (III.8) présente le schéma de base d'une identification par réseau de neurones artificiel .

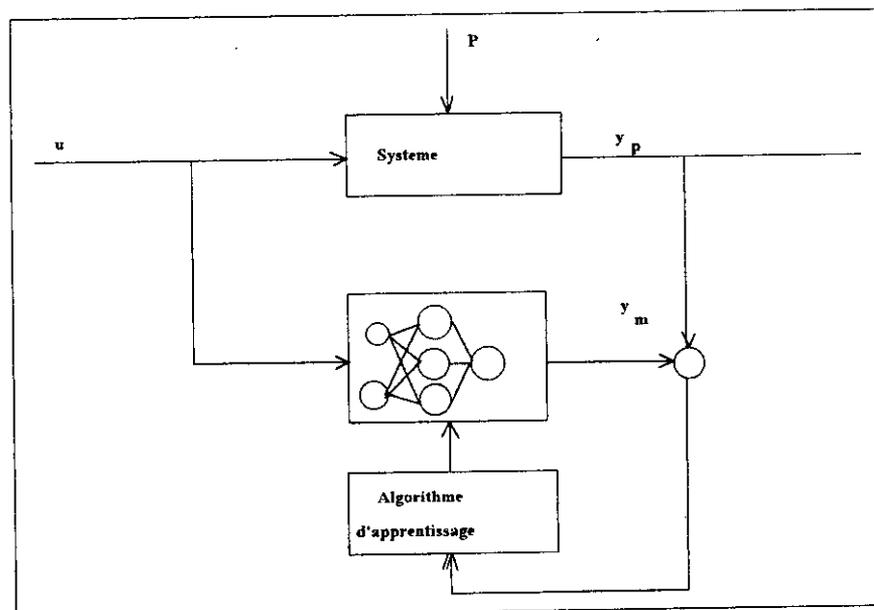


Fig (III.8) Identification du modèle direct avec ANN

On applique une commande  $U$  au système et au réseau et on compare ensuite la sortie du système  $Y_p$  avec celle du réseau  $Y_m$  erreur obtenu et celle utilisée dans l'algorithme d'apprentissage .

### 3.3.1 Schéma de base de la IMC :

Considérons les deux structures précédentes **FEEDFORWARD** et **FEEDBACK** comme cas particuliers de la commande à modèle interne . Si le modèle direct reproduite parfaitement le système c'est-à-dire que l'écart entre la sortie du modèle et la sortie du système est toujours nul . mais les inconvénients de ces deux structures sont :

- le modèle inverse est irréalisable s'il existe des retards purs dans le procédé , l'inverse est une avance .
- signalons également une grande sensibilité aux erreurs de modélisation .
- on n'a pas tenu compte des perturbations non mesurées qui affectent le procédé et modifient la sortie comme le montre la figure (III.9).

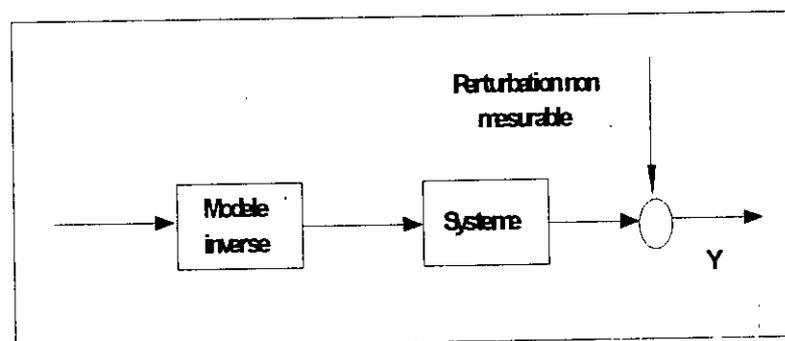


Fig (III.9) Schéma de base de IMC

On peut modifier ce schéma en introduisant la connaissance du modèle du procédé pour obtenir le schéma de la figure ( III.10 ).

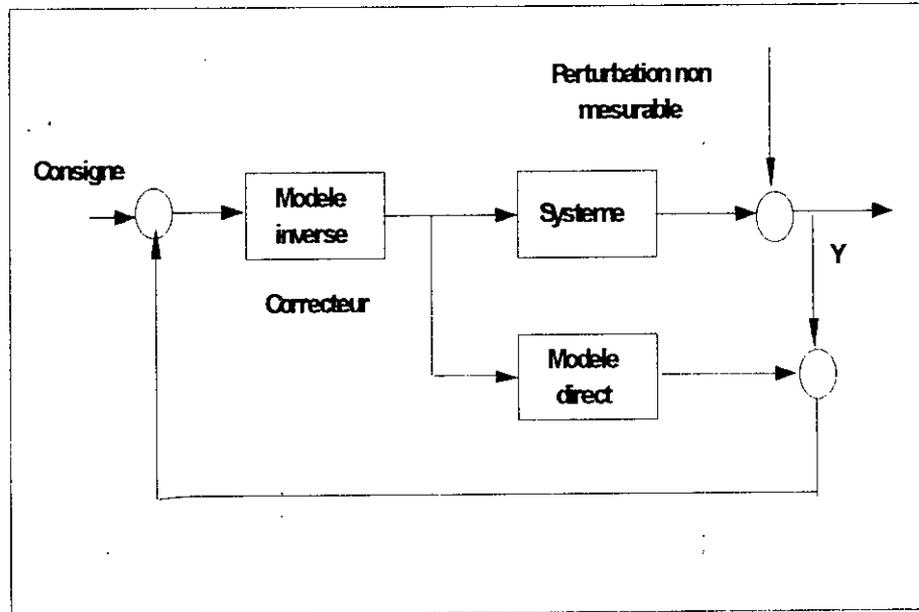


Fig (III.10) Schéma de base de IMC

Cette structure de commande à modèle interne, qui nous permet d'estimer les perturbations qui affectent le procédé et font que sa sortie  $Y$  est différente de celle du procédé  $Y_M$  puisque le signal de retour est précisément  $e = Y - Y_M$ .

On peut présenter les structures de commande à modèle interne comme une commande du modèle dans laquelle on applique au procédé les mêmes actions que celle obtenues pour le modèle et on introduit une modification des consignes en fonction des écarts entre le modèle et le procédé.

En résumé, une structure de commande à modèle interne est essentiellement caractérisée par :

- la présence d'un modèle qui est le résultat d'une identification du système par réseau de neurones.

- Une estimation des perturbations non mesurées par comparaison des sorties du procédé et du modèle .modification de la consigne par bâclage de cette estimation .

#### **4.Conclusion**

On a présenté dans ce chapitre les principales techniques utilisées pour l'apprentissage des réseaux de neurones multicouches en vue de leur application pour la commande des systèmes .

L'apprentissage du correcteur neuronal FEEDFORWARD a été largement pratiqué ces dernières années .

Par contre, les techniques d'apprentissage des contrôleurs FEEDBACK et la commande à modèle interne par réseaux de neurones font actuellement l'objet d'intenses recherches .



*Chapitre IV*

**Train de laminoir**

## 1. Introduction

Le laminage consiste à déformer et allonger un lingot ou un demi produit entre deux cylindres tournant dans le sens inverse .

cette déformation est poursuivie par des passages successifs, soit entre les mêmes cylindres , soit dans une succession de paire de cylindres, de façon à aboutir à une forme géométrique utilisable, rail, poutrelle, ect.....

En outre, avant de commencer le formage proprement dit du profil, il est nécessaire, pour des raisons purement dimensionnelles, de réaliser une réduction de section, en gardant pour celle-ci une forme polygonale convexe carrée ou rectangulaire, octogone ou rond, ce premier laminage s'appellera dégrossissage ( à 1200°c environ ) .

Le premier travail de dégrossissage est commun à tous les produits laminés, il est effectuée sur les laminoirs à demi produit .

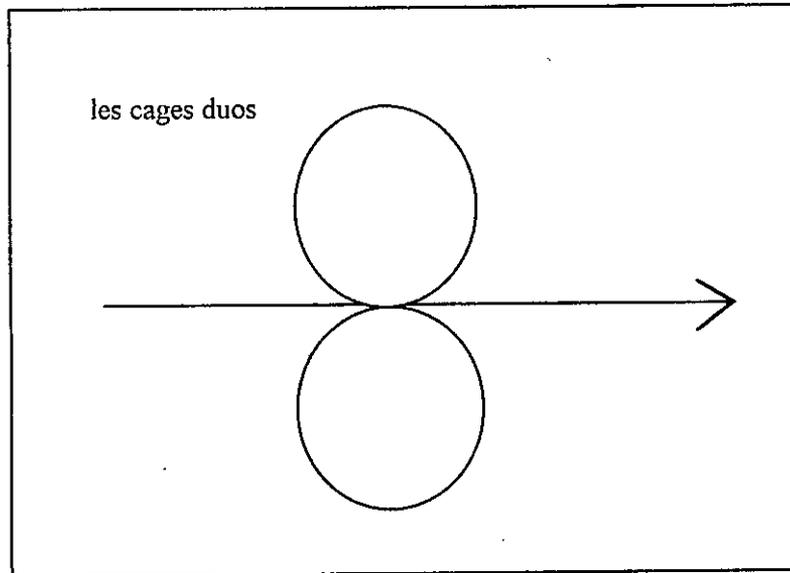
Le demi produit est repris est du une nouvelle réduction au cours de la quelle le profilage

Désiré est obtenu par le passage dans des cammelures successives places sur des cages de laminoirs constituant les laminoirs des produits finis ou laminoirs finisseurs à chaud ou à froid .

## 2. Les différents types de laminoirs [8]

On classifie les laminoirs selon le type de la cage de travail et pour ça on distingue les principaux groupes de cages qui se différent selon le nombre et la disposition des cylindres . on trouve :

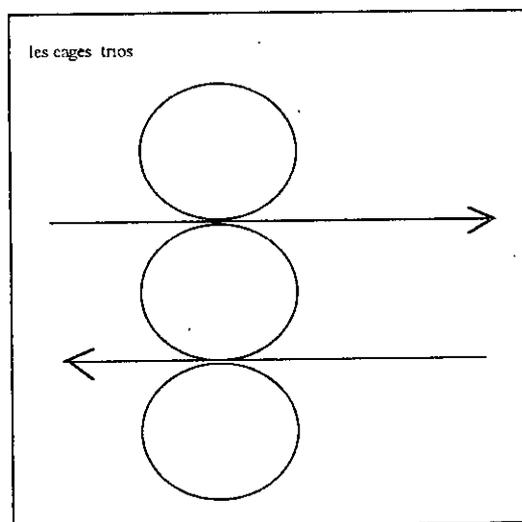
### 2.1. Les cages duos



Fig(IV.1) Schéma d'une cage duo

Les cages duos comportent deux cylindres à sens de rotation constant inverse, on les utilise largement quand le produit ne subit qu'une passe de laminage (trains finisseurs), dans les laminoirs à tôles minces, et également comme cages de dégrossissage puisque le sens de rotation des duos est réversible.

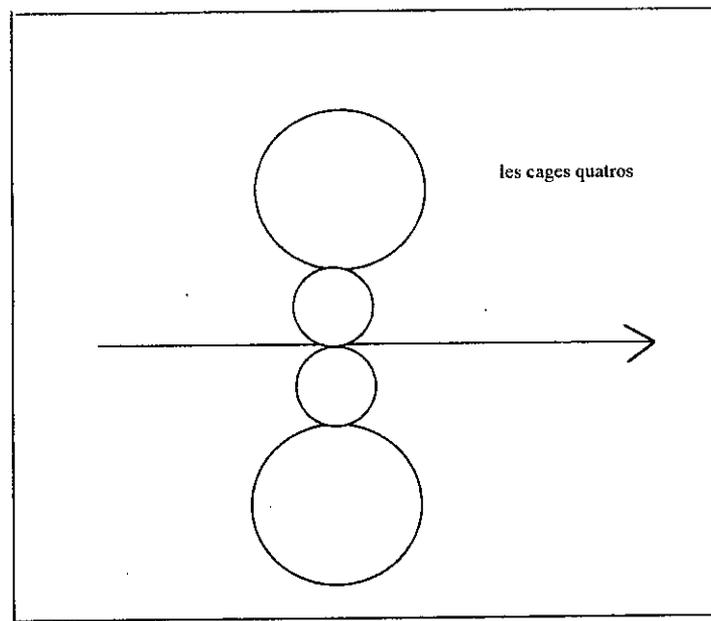
### 2.2. Les cages trios



Fig(IV.2) Schéma d'une cage trio

Chaque cylindre tourne dans le même sens c'est-à-dire qu'ils ne sont pas réversible, les cages de ce type sont largement employées dans les laminoirs à profiles de cage côte à côte .

### 2.3. Les cages quartos



Fig(IV.3) Schéma d'une cage quartos

Elles comportent quatre (04) cylindres placés l'un au dessus de l'autre dans la plan vertical, les deux petits cylindre sont les cylindres de travail, alors que les deux autres, plus grands, forment l'appui; ils reçoivent la pression développée pendant l'opération et réduisent la flexion des cylindres de travail .

Ces cages sont utilisées uniquement pour le laminage des produits plats à chaud ou à froid .

La mise en application du quartas, qui est à l'origine de la réussite des trains continus à tôles, a complètement révolutionné la technique du laminage à plat.

Et il existe d'autres types de cages connues, les cages à cylindres multiples et les cages universelles.

### 3. Conditions que doit remplir un laminoir [8]

Considérons le laminoir le plus simple, c'est-à-dire celui qui est constitué de deux cylindres lisses sans conelures, supposons pour simplifier que l'élargissement est négligeable,

Nous appellerons

$h_e$  et  $h_a$  l'épaisseur du produit avant et après laminage

$p$  l'effort sur les deux cylindres

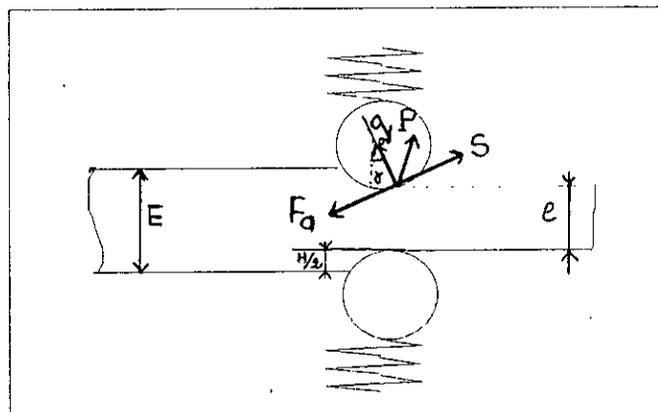
$h$  pression de laminage =  $h_e - h_a$

$r$  les rayons des cylindres et  $d$  est le diamètre

$\alpha$  l'angle d'attaque

$f$  le coefficient de frottement

$q$  et  $s$  forces de décomposition



Fig(IV.4) Schéma d'une cage simple

l'effort  $p$  sur les cylindres se divise en deux composantes :

- l'une  $q$  radiale, passe par l'axe et est reporté finalement sur les tourillons .

- l'autre  $s$ , tangentielle s'oppose à la force d'entraînement du cylindre et est égale à  $f q$  .

pour qu'il y'ait entraînement, il faut que  $t q \geq q \operatorname{tg} \alpha$  .

on peut voir sur la figure ... que

$$H/2 = r - r \cos \alpha$$

en remarquant qu'à l'équilibre  $t = \operatorname{tg} \alpha$  , on a

$$\sin \alpha = f \cos \alpha \Rightarrow \cos \alpha = 1 - f \cos \alpha .$$

$$\Rightarrow \cos \alpha = 1 / f \cos \alpha .$$

$$\cos \alpha = \sqrt{\frac{1}{1+f}}$$

$$H = 2r \left( 1 - \sqrt{\frac{1}{1+f}} \right)$$

Le maximum de pression admissible est donc directement proportionnel au diamètre des cylindres et croît avec les coefficients de frottement .

Nous tiendrons de ce qui précède que deux facteurs principaux ont une influence sur la pression : le diamètre des cylindres et le coefficient du frottement .

De ce qui précède , on remarque que la nature des cylindres joue un rôle très important, mais l'aspect de la surface survient aussi .

En conclusion, la condition exigée pour entraînement d'une barre est moins sévère que l'engagement . un lingot engage par sa section la plus faible passe entier dans les cylindres .

#### 4. Le tandem quatre à cages [7],[6]

Il s'agit du laminoir à froid continu de haute production pour lequel les programmes doivent être en harmonie c'est-à-dire la commande numérique .

On va lui appliquer la commande par réseau de neurones pour la régulation d'épaisseur . Pour cela, il faut tout d'abord présenter les caractéristiques de laminage propres à chaque cage afin d'enrouler à la sortie de la dernière cage une bande dont la section et la régularité d'épaisseur seront conformes aux tolérances imposées .

Le schéma de cette installation est présenté dans la figure . (IV.6)

Les caractéristiques des produits à laminer définissent principalement les puissances entraînement des cages .

Les matières à laminer sont :

- Des bandes d'acier doux à 0.12 % maxi de carbone . les bobines à laminer ont pour caractéristiques .

- Largeurs des bandes 650 à 1525 mm .

- Epaisseur des bandes 2 à 5 mm .

- Poids des bobines : 5 à 28 tonnes .

- Les bandes laminées ont pour épaisseur : 0.50 à 3mm .

- Diamètre des bobines : 610/1830 mm .

La bande qui après avoir été placée automatiquement dans l'axe d'une dérouleuse à double mandrin traverse successivement les cages 1,2,3 et 4 .

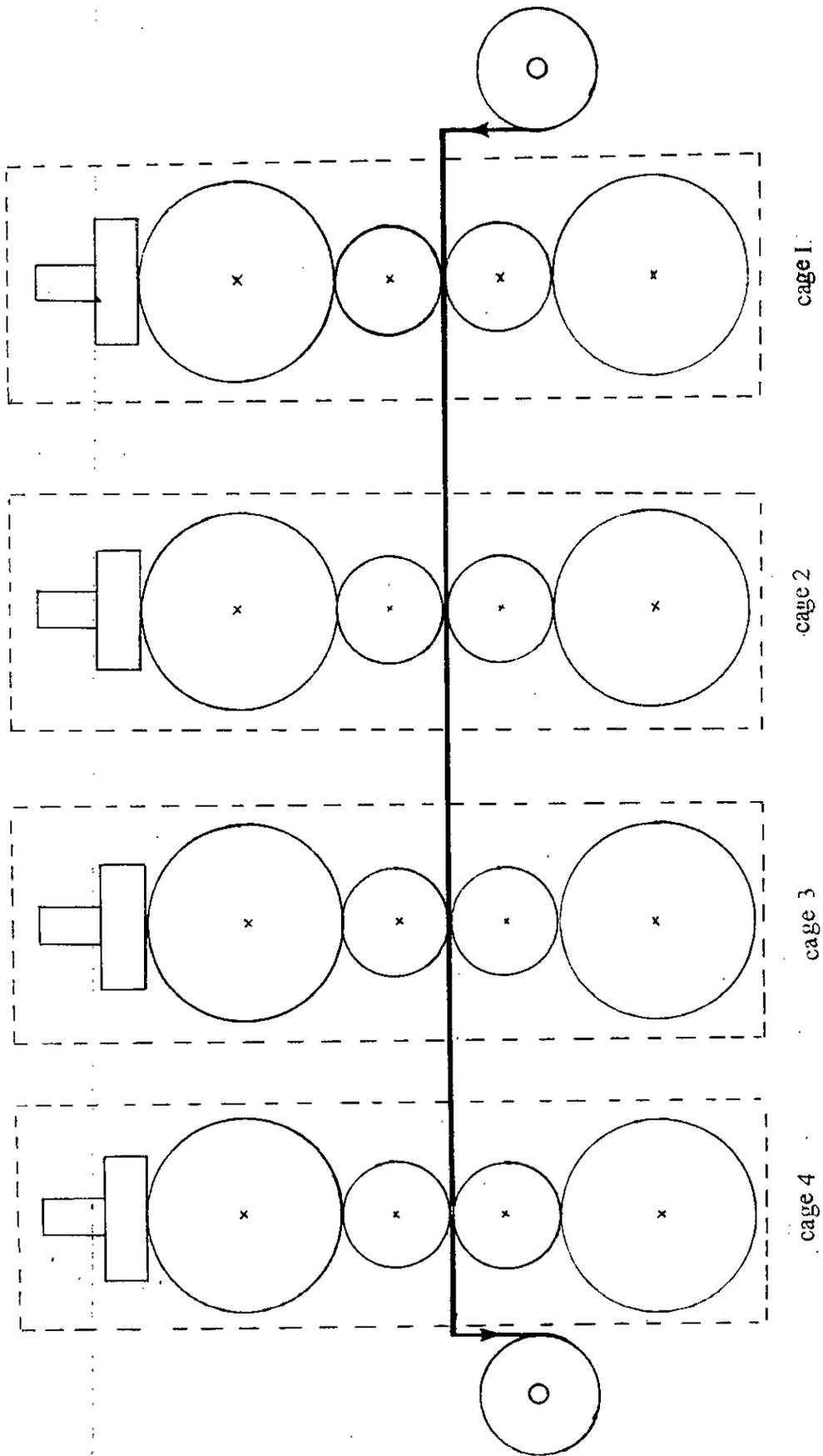


Fig (1v.6)

Puis s'enroule automatiquement sous traction sur le mandrin d'une enrouleuse .

La conduite automatique de cet ensemble a nécessité l'introduction de quatre niveaux de " commande " .

- régulations élémentaires des serrages c'est-à-dire la distance entre les cylindres .

- régulation des épaisseurs .

- régulation des tractions

- régulation des vitesses .

### **5.Modèle mathématique [6],[7],[8],[9]**

Pour réguler épaisseur à la sortie de laminoir, il faut effectuer un travail de correction épaisseur à la sortie de chaque cage. pour cela, on prend une seule cage et on la modélise pour pouvoir commander épaisseur à la sortie .

La modélisation de la cage détermine un système d'équation non linéaires permettant de calculer épaisseur de sortie en fonction des forces et les autres grandeurs caractéristiques du laminage .

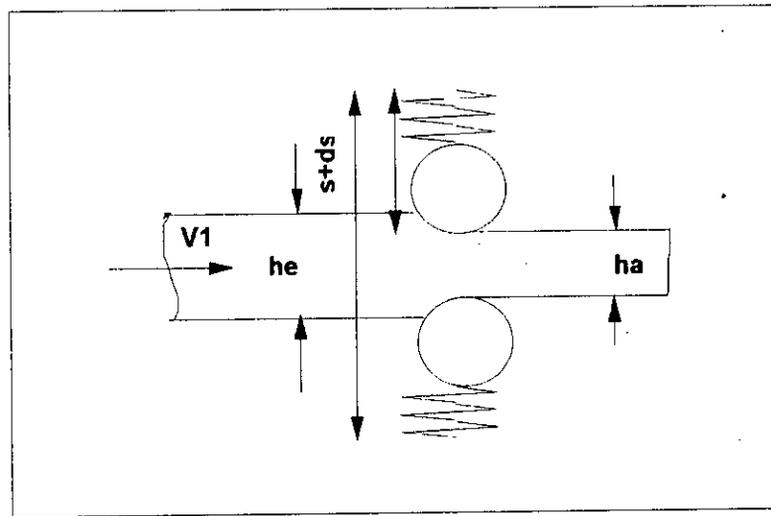
Avant de définir les différentes relations, décrivons brièvement le comportement du processus en rappelant quelques définitions, les principes de base et les hypothèses choisies .

La première hypothèse consiste à dire qu'une tranche de métal conserve sa largeur inchangée quand elle traverse l'emprise ( la surface ou le domaine de contrôle entre les cylindres et la tôle ) des cylindres, celle-ci se définit par le volume délimité d'une part par les surfaces de contact et d'autre part , par la section verticale de la bande .

L'entrée de l'emprise, la vitesse de la bande est constante et supposée inchangée en cours de la première partie de notre travail .

### 5.1. Forces de laminage

Considérons la laminoir schématisé par le schéma de la figure suivante .



Fig(IV.5) Diagramme général d'un laminoir

$h_e$  : l'épaisseur de la bande (brame) à l'entrée

$h_a$  : l'épaisseur à la sortie

$s_0$  : est la distance entre les cylindres au repos

$f_w$  : la force de laminage

$v_1$  : vitesse de la tôle à l'entrée

$v_w$  : vitesse de la tôle à la sortie .

Les équations de base qu'on essaie de les établir entre les différentes sont donnés au point d'équilibre

Lorsqu'une tranche de métal arrivant dans l'emprise des cylindres et la cage soumise à une déformation élastique .

la force de laminage , rapporté à l'unité de largeur de l'emprise s'écrit :

$$f_w = M(h_a - s_0) \dots (4.1)$$

$h_a > s_0$  à cause de la déformation élastique des cylindres .

$M$  est une constante qui exprime la résistance du matériau .

la caractéristique du matériau à laminé est donnée par l'équation suivante :

$$f_w = f(K, \sigma, \mu, R') \sqrt{h_e - h_a} + f_e \quad (4.2)$$

$k$  : est la constante moyenne d'écoulement du métal lamine (  $n/m^2$  )

$r$  : est la contrainte moyenne de tension appliquée .

$\mu$  : coefficient de frottement entre le métal et le cylindre

$R'$  : est le rayon du cylindre déformé élastiquement

$f_e$  : est la force d'élasticité à la sortie .

Dans le cadre de simulation, on peut simplifier l'équation précédente

ou

$$f_w = v_s \sqrt{h_e - h_a} \quad (4.3)$$

ou

$$v_s = v_0 \frac{3v_{i\max}}{2v_{i\max} + v_i} \quad (4.4)$$

Donc , en faisant l'égalité entre l'équation (4.1) et l'équation (4.3)

$$M(h_a - s_0) = v_s \sqrt{h_e - h_a}$$

On met  $h_a - s_0 = ds$  on obtient :

$$ds = -\frac{v_s^2}{2M^2} + \sqrt{\frac{v_s^4}{4M^4} - \frac{v_s^2}{M^2}(s_0 - h_e)} \dots\dots(4.5)$$

$$h_0 = s_0 + ds \dots\dots(4.6)$$

$$f_w = Mds$$

Notre objectif est de maintenir  $h_a$  plus proche possible de  $h_{ref}$ , par la commande  $s_0$ , lorsque le laminoir reçoit des perturbations au niveau de l'épaisseur d'entrée  $h_e$  et de la vitesse d'entrée de la tôle.

Ce modèle est non linéaire et la non linéarité est dans la commande ( $S_0$ ) qui est en racine carrée. Les techniques de commande classiques sont inapplicables sans linéarisation autour d'un point de fonctionnement, et pour garder toutes les performances du système une nouvelle technique de commande qui est la commande par réseaux de neurones va jouer le rôle.



*Chapitre V*

**Simulation**

## 1. Introduction

La simulation donne une image du fonctionnement statique, ou dynamique d'un processus sur la base d'un modèle mathématique. ainsi celui ci peut prévoir l'évolution de la qualité et de la productivité d'une installation industrielle, sans pour autant influencer le processus de production réelle, cela fait gagner du temps et éviter des essais coûteux.

Le but de ce chapitre est d'appliquer les différentes structures de réglage par réseaux de neurones étudiées dans le chapitre III dans la régulation d'épaisseur un train de laminoir.

## 2. Régulation d'épaisseur

Il s'agit d'effectuer un travail de correction d'épaisseur maximale sur les deux premières cages afin de délivrer aux cages suivantes une épaisseur pratiquement indépendante des irrégularités de la bande à l'entrée du laminoir.

La solution que nous avons mise en oeuvre décompose la régulation épaisseur du laminoir en deux parties principales .

-Une régulation d'entrée agit sur le serrage des cages 1 et 2 de façon à aplanir les défauts d'épaisseur a l'entrée toute en gardant la vitesse d'entrée constante.

-Une régulation des deux dernières cages 3 et 4 agit sur la vitesse de la cage 4 de manière à modifier dans les mêmes proportions le débit de métal à la sortie du train .

Une correction de vitesse sur la cage 4 se traduit par une variation de la retenue (traction entre cages 3 et 4) qui sera corrigée par la régulation de traction entre cages agissant sur la vitesse de la cage 3.

Dans ce but on a partagé la simulation en deux parties.

La première considère la vitesse du laminoir constante et la régulation se fait en agissant sur le serrage ( $s_0$ ) la deuxième utilise le modèle complet du train et considère la vitesse comme perturbation d'entrée.

### PARTIE 1

#### Le système à vitesse constante

Remplaçons dans l'équation (4.4),  $V_{1\max}=V_1$

d'où

$$ds = -\frac{v_0^2}{2M^2} + \sqrt{\frac{v_0^4}{4M^4} - \frac{v_0^2}{M^2}(s_0 - he)}$$

et

$$ha = s_0 - \frac{v_0^2}{2M^2} + \sqrt{\frac{v_0^4}{4M^4} - \frac{v_0^2}{M^2}(s_0 - he)}$$

Le système a donc deux entrées  $he$  et  $s_0$

$he$ : épaisseur d'entrée de la tôle (perturbation)

$s_0$ : l'écartement des cylindres (commande)

et une sortie qui est l'épaisseur de sortie  $ha$ .

**PARTIE 2**

Le système à vitesse variable

Les équations sont celles du modèle complet c'est à dire :

$$h\alpha = -\frac{v_s^2}{2M^2} + \sqrt{\frac{v_s^4}{4M^4} - \frac{v_s^2}{M^2}(s_0 - he)}$$

avec

$$v_s = v_0 \frac{3v_{i\max}}{2v_{i\max} + v_i}$$

Une variation de vitesse à l'entrée de la tôle perturbe le système et l'éloigne de la référence, le modèle complet du système concéder la vitesse à l'entrée comme une deuxième perturbation après l'épaisseur d'entrée

La vitesse d'entrée de la tôle est donnée par une période d'accélération, et de décélération (une rampe de 1a 5m/s, constante puis une rampe de 5a 1m/s).

Donc la valeur de  $V_{1\max}=5\text{m/s}$ .

**3.Commande anticipatrice**

La commande anticipatrice ,généralement associée à la correction série on parallèle, et appelée parfois compensation , permet d'atténuer sensiblement les effets de la variation de la grandeur de référence et / ou grandeur perturbatrice .

**3.1.Compensation de la perturbation**

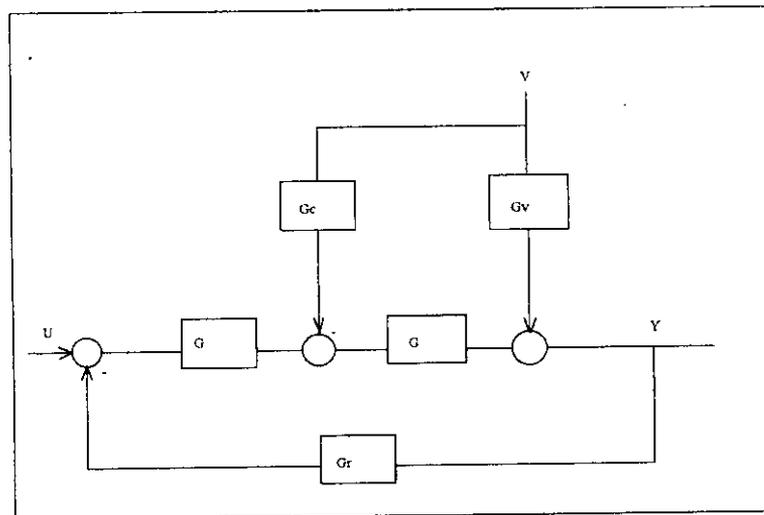
La compensation de perturbation concerne la régulation de maintien ou la consigne  $W=$ constante le problème consiste alors a compenser la perturbation  $V$ , a condition que celle ci soit mesurable .

Théoriquement, il est possible d'éliminer l'influence de la grandeur perturbatrice en plaçant, entre cette perturbation et une située en amont de celle ci, un correcteur (compensateur)

dont la grandeur de est

$$u(t)=f(v(t))$$

Ce type de correction est illustre par la figure (V.1.a)



Fig(V.1.a)

On a

$$Y=-VGv-GcGwV+XeGw$$

$$Y=-(Gv+GcGw)V+GwXe$$

En posant

$$Gc=-Gv/Gw$$

On obtient

$$Y/Xe=Gw$$

D'ou

$$Y=(GwG/(1+GwGGr))W=GfW$$

On constate bien que  $Y(s)=Gf(s)W(s)$  est indépendante de  $V(s)$

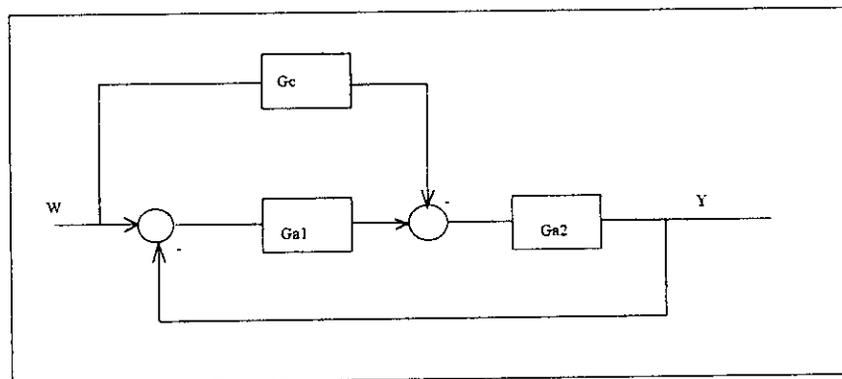
Ce type de correction, qui est théoriquement une compensation parfaite de la perturbation, ne modifie ni la stabilité, ni la précision, vis a vis de la grandeur de référence .

Malheureusement,  $Gc=-1/Gw$  n'est pas réalisable si  $Gv$  est un gain; on se contente en pratique ,d'une forme approximative, la compensation en régime perturbe est donc imparfaite.

### 3.2.Compensation de la grandeur de référence

la compensation de la grandeur de référence concerne la régulation de correspondance en  $V=constante$ . Le problème consiste à annuler à tout instant le signale d'erreur  $e(t)$ .

Ce type de correction est illustrée par la figure (V.1.b)



Fig(V.1.b)

$$Gc=1/Ga1 \quad Y=W$$

On constate que la grandeur de sortie est toujours égale a la grandeur de référence et que le signal d'erreur  $e(t)$  est toujours nul.

Mais le problème de la réalisation de  $G_c=1/G_a2$  se pose toujours.

Dans notre étude on utilise la compensation de la perturbation, puisque notre système possède une perturbation a l'entrée qui n'est pas constante.

Le système ne présente aucune dynamique, alors avec une linearisation autour d'un point de fonctionnement ,le système se réduit a un gain constant.

Le point de fonctionnement choisie pour cette application est:  
 $ha=3mm$ .

$ha$  est en fonction de  $s_0$  et  $he$

$$ha = f(s_0, he)$$

$$\Delta ha = \left. \frac{\partial f}{\partial s_0} \right|_{pt\ fc} \Delta s_0 + \left. \frac{\partial f}{\partial he} \right|_{pt\ fc} \Delta he$$

On obtient :

$$\Delta ha = \Delta s_0 V + \Delta he U$$

Où  $U$  est  $V$  sont des gains constantes.

Puisque  $1/V$  est aussi un gain une compensation de la perturbation  $he$  est réalisable et le schéma correspondant a cette commande est illustrée par la figure (V.2)

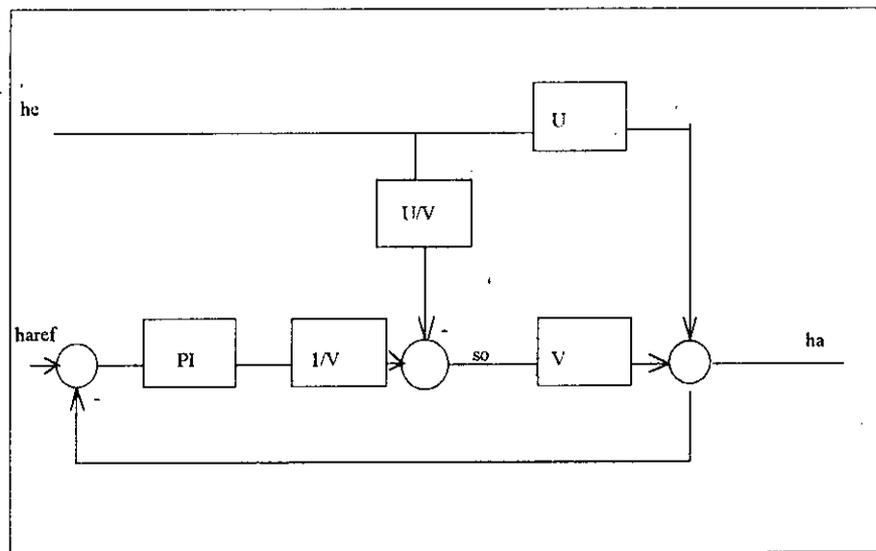
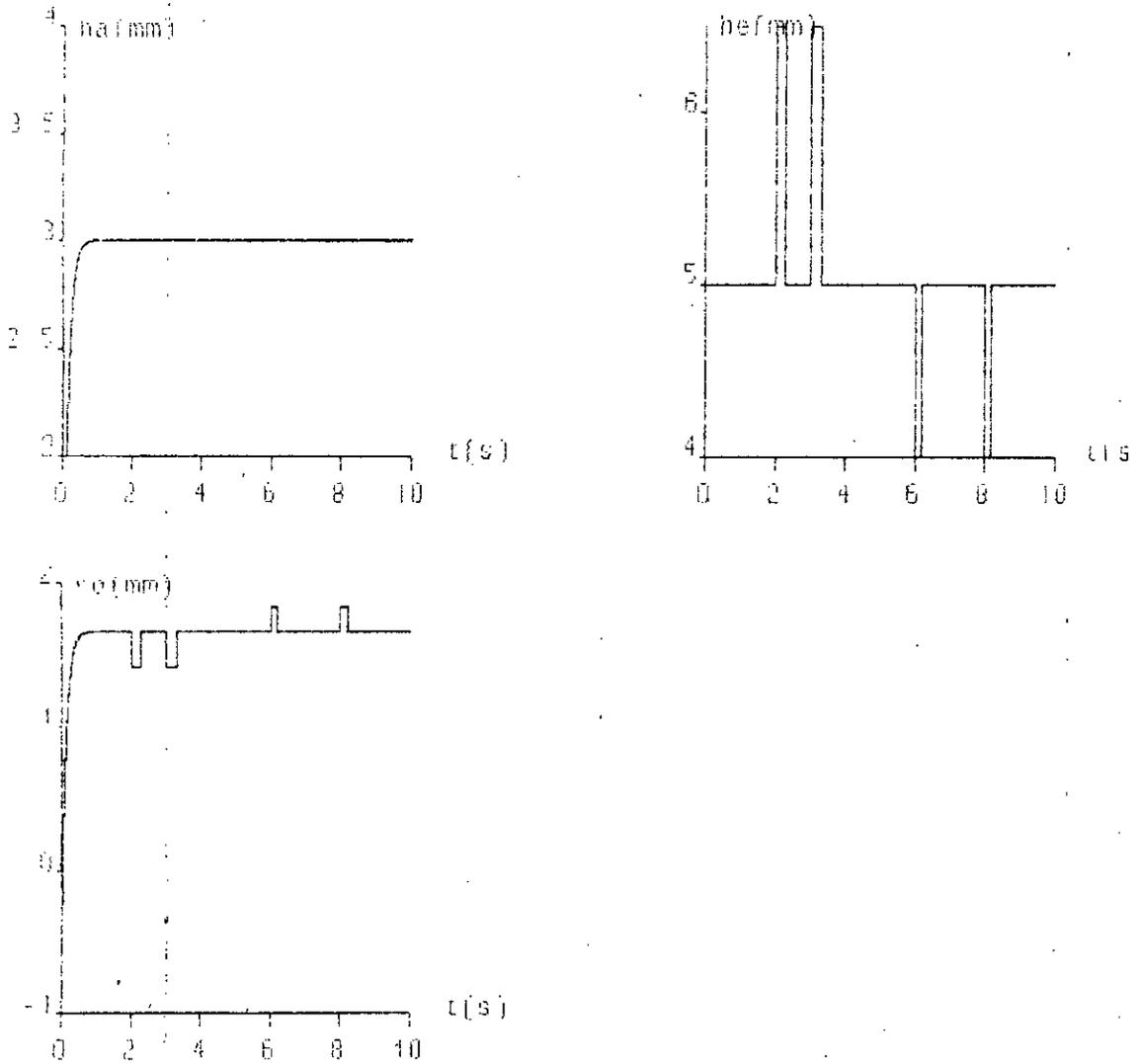


Fig (V.2.a) Contrôleur Feedformard lineair

Un régulateur PI est inséré dans la chaîne d'action pour annuler l'erreur statique.

les résultats de cette commande sont présentés dans la figure (V.2.b)

On constate qu'il y a une compensation de la perturbation produite dans l'épaisseur d'entrée, un régime dynamique s'établit cela est dû à la constante de temps du correcteur PI, néanmoins cette structure n'est utilisable qu'autour d'un point de fonctionnement.



Figure(V-2.a) : Reponse du systeme avec une commande et un PI (des impulsions sur  $h_e$ ).

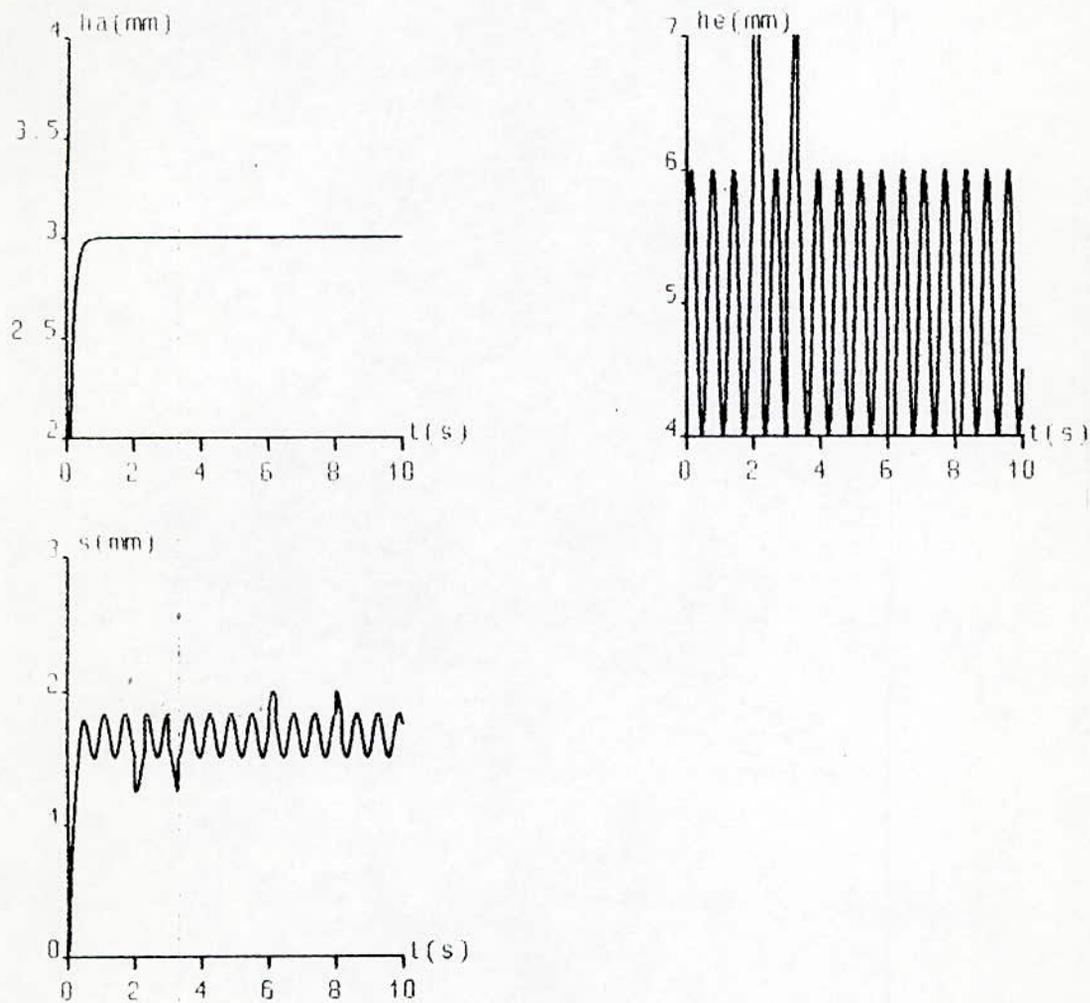


Figure (V-2.b): Reponse du systeme avec une commande anticipatrice avec un PI ( $h_e$  deformee).

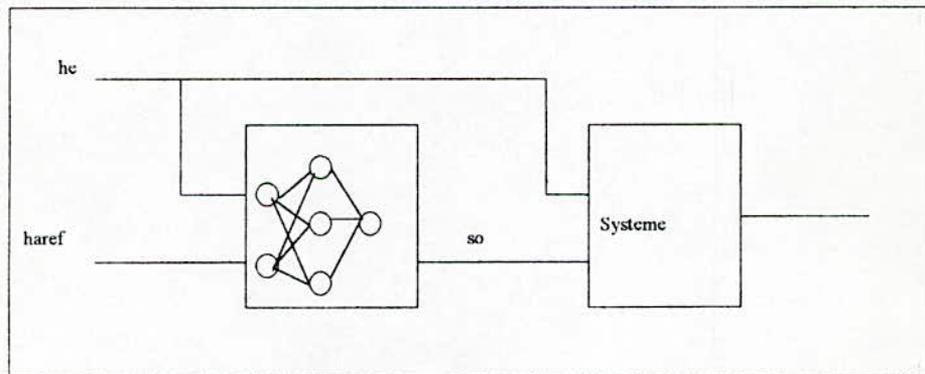
## 4. Commande par réseaux de neurones

On prend le modèle mathématique d'une cage établi dans le chapitre précédant et on essaye de le commander avec les différentes structures de réglage par réseau de neurones citées dans le chapitre III

### 4.1. Contrôleur Feed-Forward

#### 1- Vitesse constante :

Avec un réseau de neurones de deux couches cachée à trois neurones chacune , une sortie qui représente la commande  $so$  , et deux entrées  $ha$  , on identifie le modèle inverse direct du système de laminoir qui sera par la suite utilisé comme contrôleur dans la chaîne d'action . comme il est illustré sur la figure (v.3)



Fig(V.3) Commande en boucle ouverte avec un modèle neuronal inverse du système

On fixe la consigne  $h_{ref}$  à une valeur bien précise et on crée des perturbation au niveau de  $h_e$  qui représente dans la pratique des bosses à l'entrée de la tôle .

Les réponses du système contrôlé en boucle ouverte est données par la fig (V.4) .

## 2 - Vitesse Variable :

Le réseau de neurones qui simulera le modèle inverse non linéaire de ce système, est un réseau a trois entrées, deux entres eux il les considère comme perturbations qui sont la vitesse et l'épaisseur d'entrée , et une sortie qui est la commande ( $s_o$ ), comme il est représenté dans la figure (V.5)

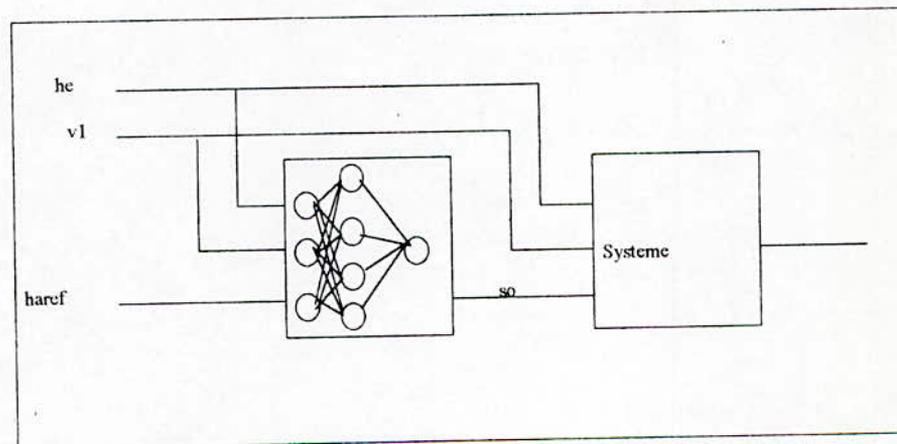
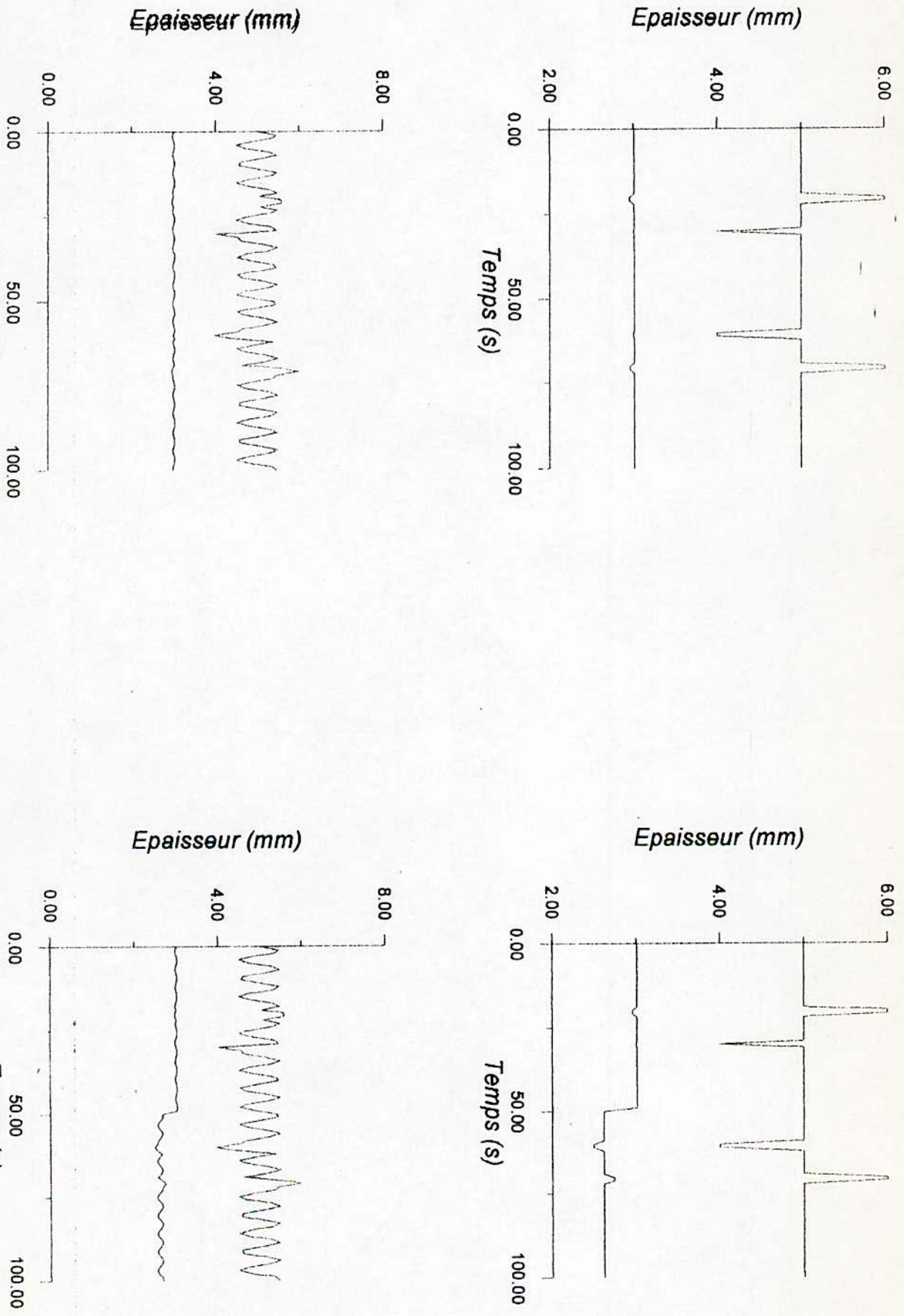


Fig ( V.5)

Les résultats de cette simulation sont donnés par la figure(V.6).

Les résultats montrent évidemment que la structure de commande en boucle ouvert ne peut pas assurer une erreur statique nulle , par ce qu'il n'ya pas de signal du retour .



Fig(V. 4)- Reponses du systeme avec le correcteur neuronal inverse en boucle ouverte

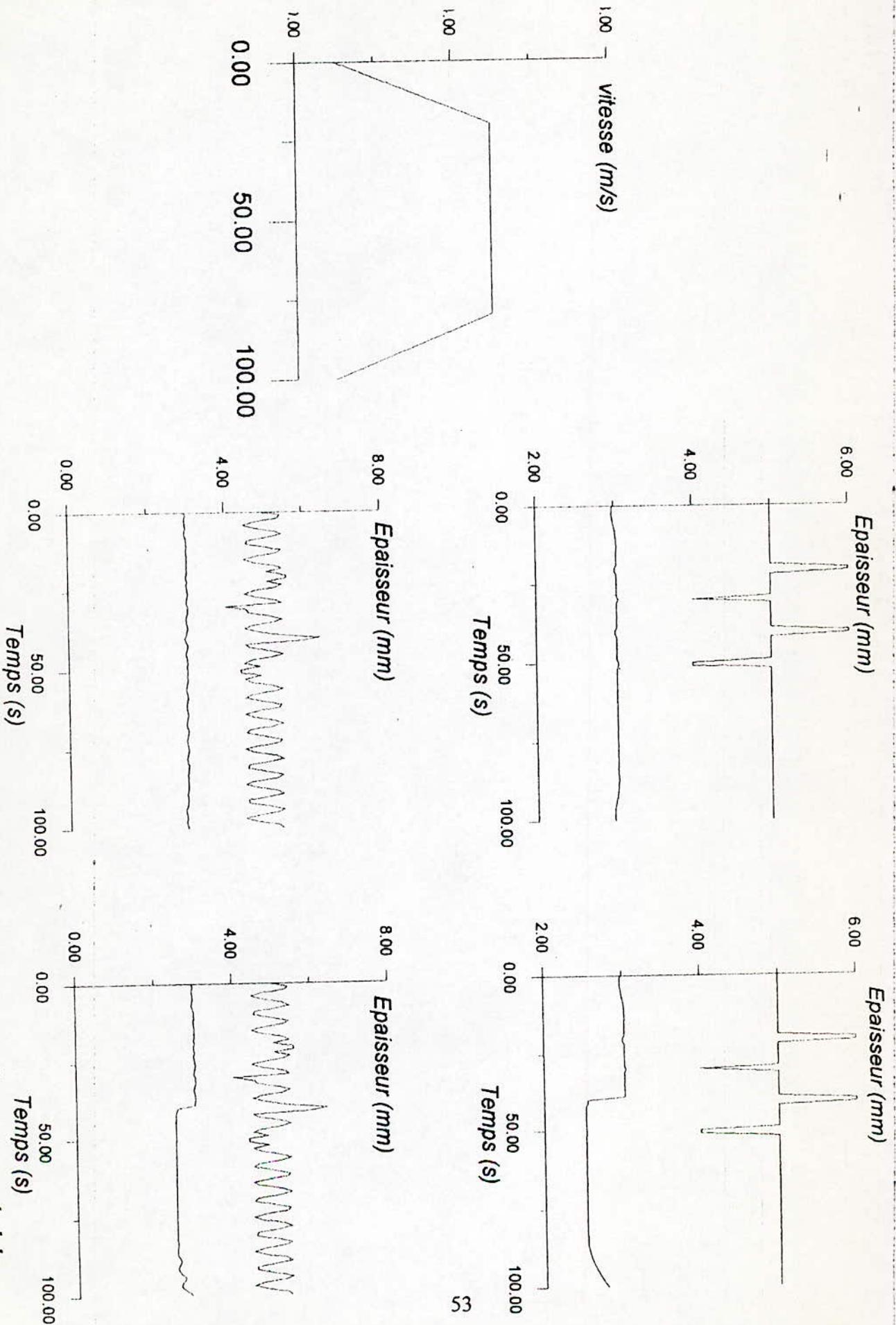


Fig (V.6)-Reponses du systeme en boucle ouverte a vitesse variable

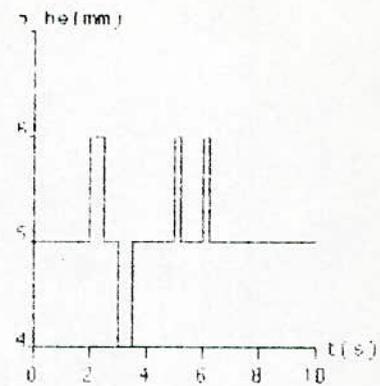
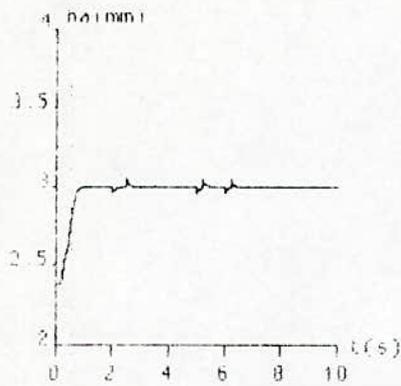


Fig.V- : reponce de systeme avec un PI et model inverse (impulsion dans  $h_e$ )

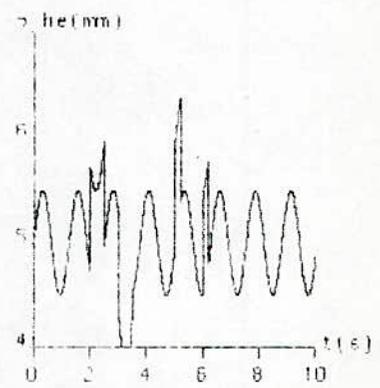
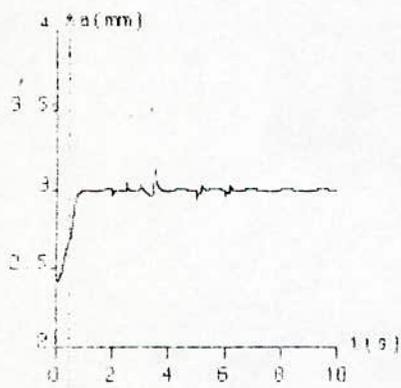
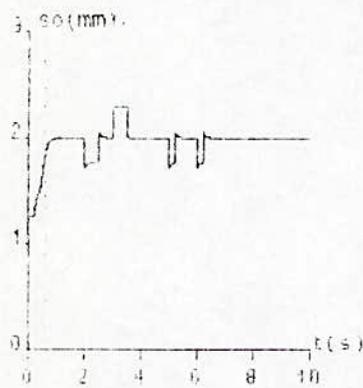
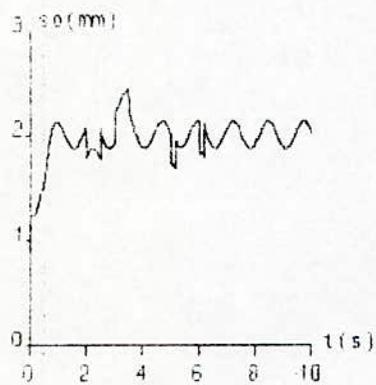


Fig.V- Reponse du systeme avec un PI et modele inverse ( $h_e$  deforme)



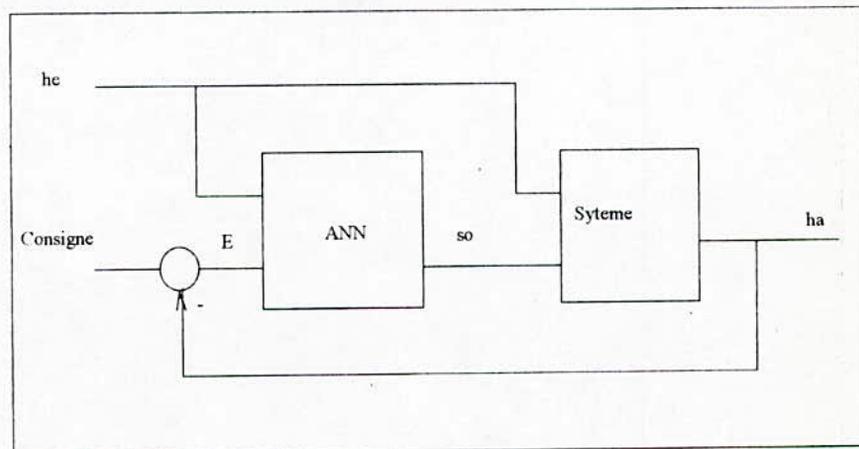
Seulement cette approche peut donner une meilleur idée de l'efficacité de la modélisation par réseaux de neurones que n'importe qu'elle mesure d'erreur, par ce que les résultats montres combien l'erreur de modélisation influe sur la sortie du système .

D'autre part l'erreur statique non nulle n'est pas acceptable pour cette application, pour cela la commande en boucle fermée est préférable .

#### 4.2. Contrôleur Feed-back

##### 1- Vitesse constante :

Commençons d'abord par le retour sans correcteur PI, la figure (V.7) représente le schéma bloc de cette commande



Fig(V.7) Contrôleur Feedback à vitesse constante

Cette fois ci le réseau à couches est entraîné préalablement avec l'erreur et non pas avec la sortie ha

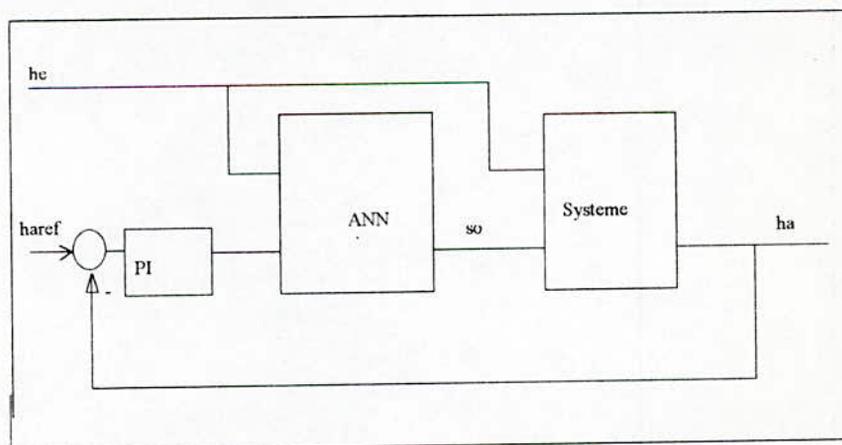
Les résultats de cette approche sont représentés dans la figure (V.8)

La figure (V.8) montre bien qu'il y a une erreur statique non nulle qui est dûe aux erreurs de modélisation , et lorsqu'on change le point de fonctionnement on voit bien que le contrôleur devient sensible aux variations produites sur l'épaisseur d'entrée , cela est claire lorsque le système reçoit à son entrée une tôle avec une épaisseur complètement déformée.

Essayons maintenant , d'insérer un régulateur PI dans la chaîne directe .

Dans cette structure, l'idée principale est d'introduire la relation non linéaire entre les variables du système ,en introduisant dans la boucle le modèle non linéaire inverse du système .

Dans ce sens le régulateur PI voit le système comme un système linéaire avec un gain unité , si le modèle inverse n'est pas parfait le régulateur PI aide à produire une erreur statique nulle la structure générale est dans la figure (V.9)



Fig(V.9) contrôleur PI avec un modèle neuronal inverse du système

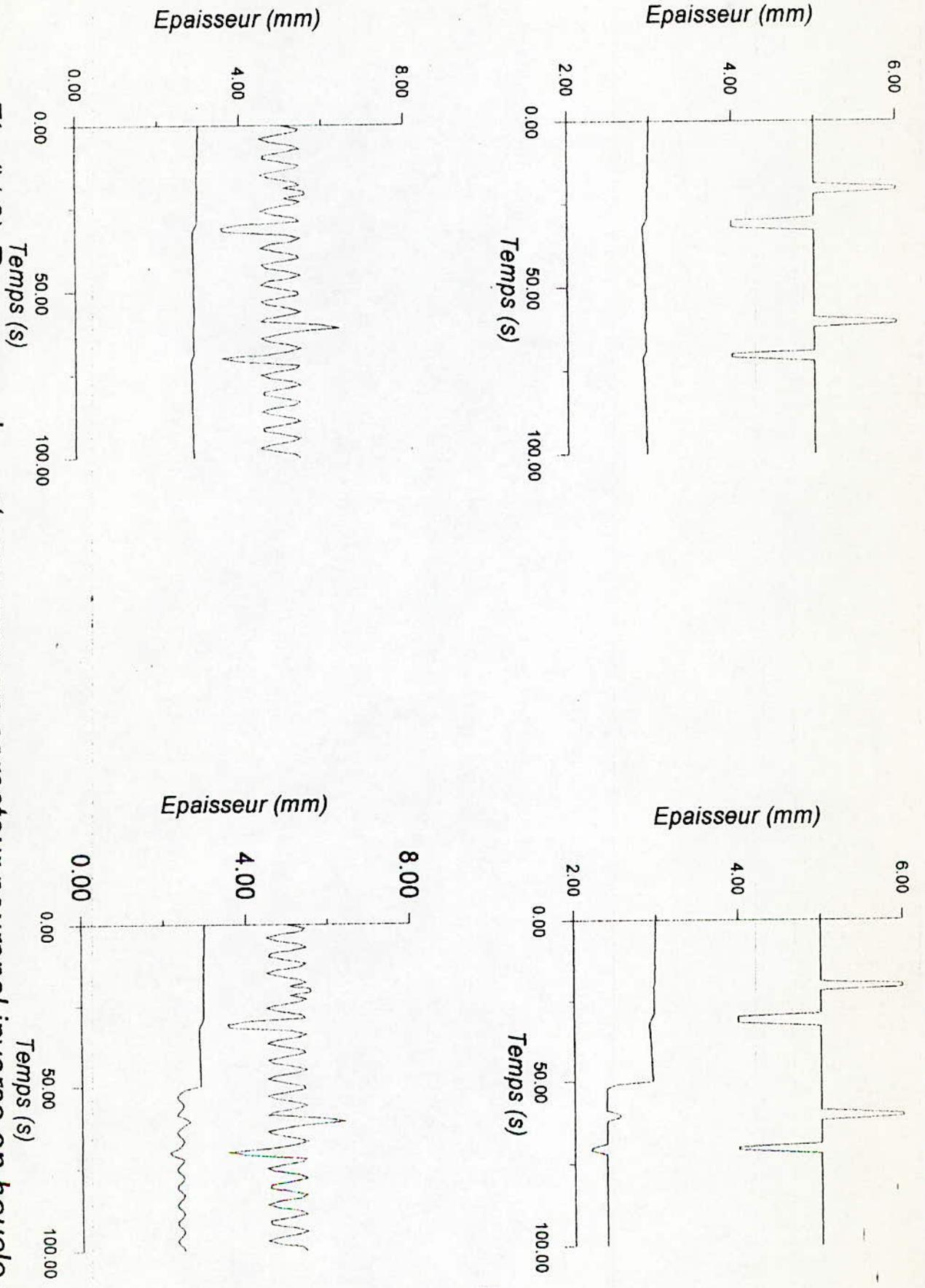


Fig (V.8)-Reponses du systeme avec un correcteur neuronal inverse en boucle ferm.

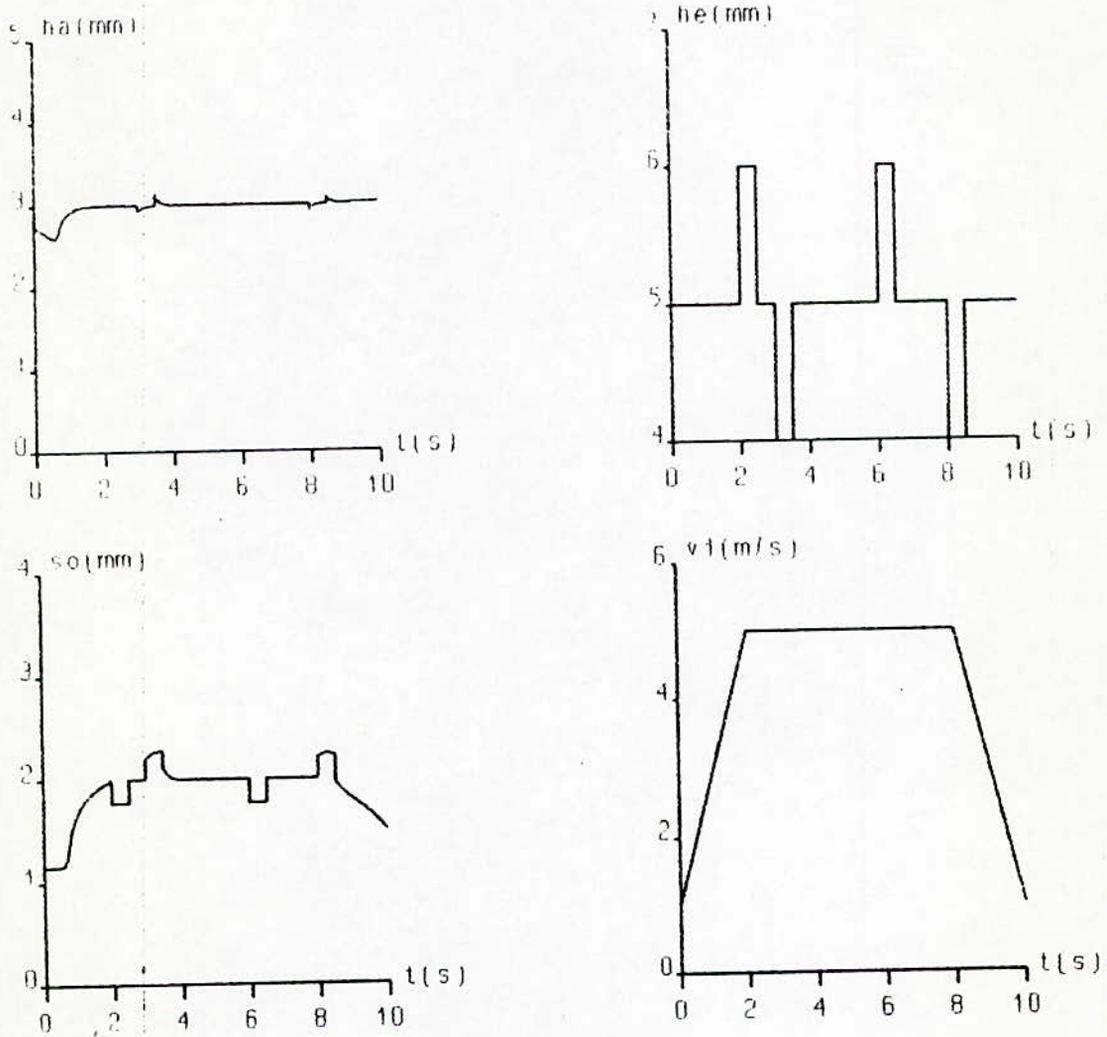


Figure (V-14.a): Reponse du svsteme avec un modele inverse et un correcteur PI (des impulsions sur  $h_e$ ).

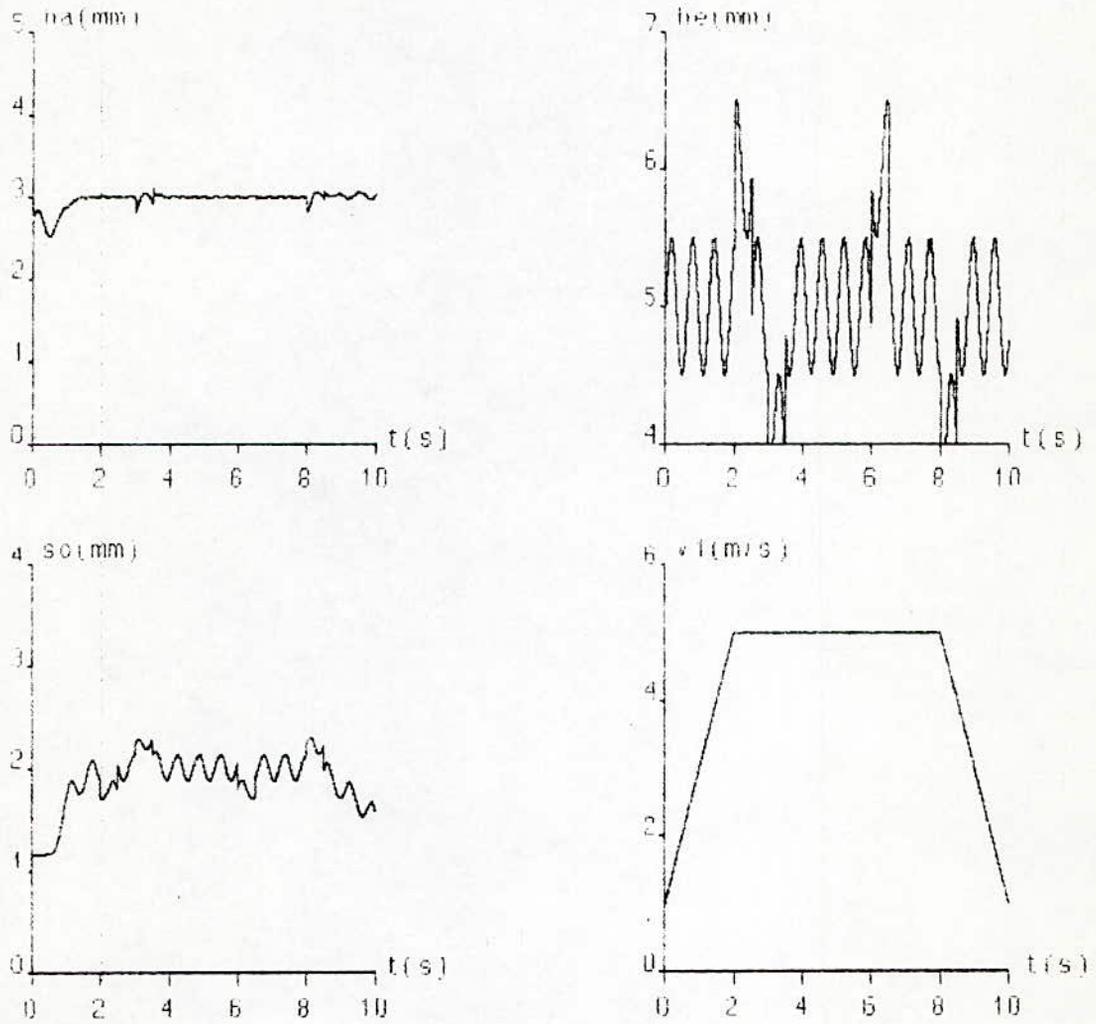
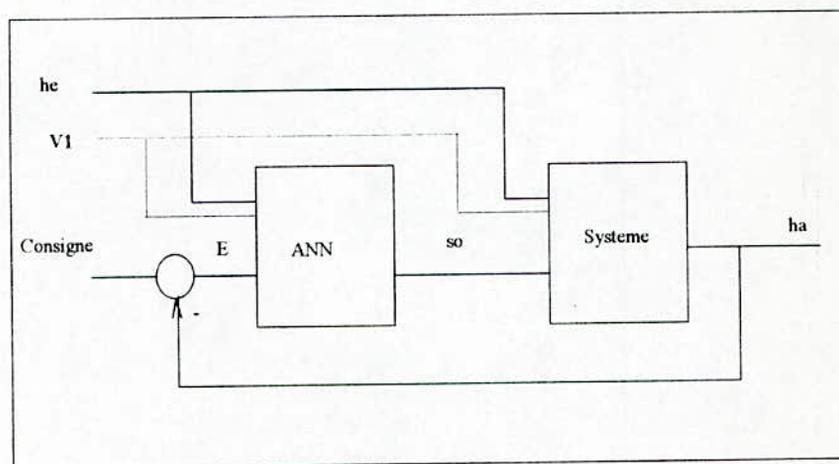


Figure (V-14.b): Reponse du svsteme avec un modele inversee et un correcteur PI (he deformee).

## 2- Vitesse Variable :

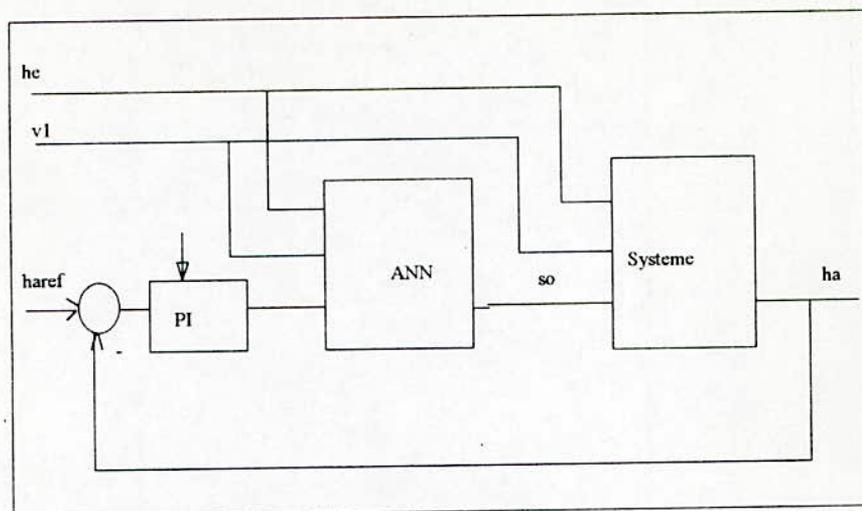
La sortie du système est comparée avec la consigne, et l'erreur est injectée dans le réseau de neurones qui est entraîné au préalable avec l'erreur et les deux perturbations .

le schéma de cette commande est représenté dans la figure (V.10).



Fig(V.10) Feedback à vitesse variable

Cette structure donne de mauvais résultats, chose qui nous oblige à placer un correcteur PI après le comparateur et utiliser le réseau de neurones du correcteur feedforward comme il est montré dans la figure(V.11).



Fig(V.11) contrôleur PI avec un modèle inverse neuronal

La figure (V.12) montre la réponse du système avec cette structure

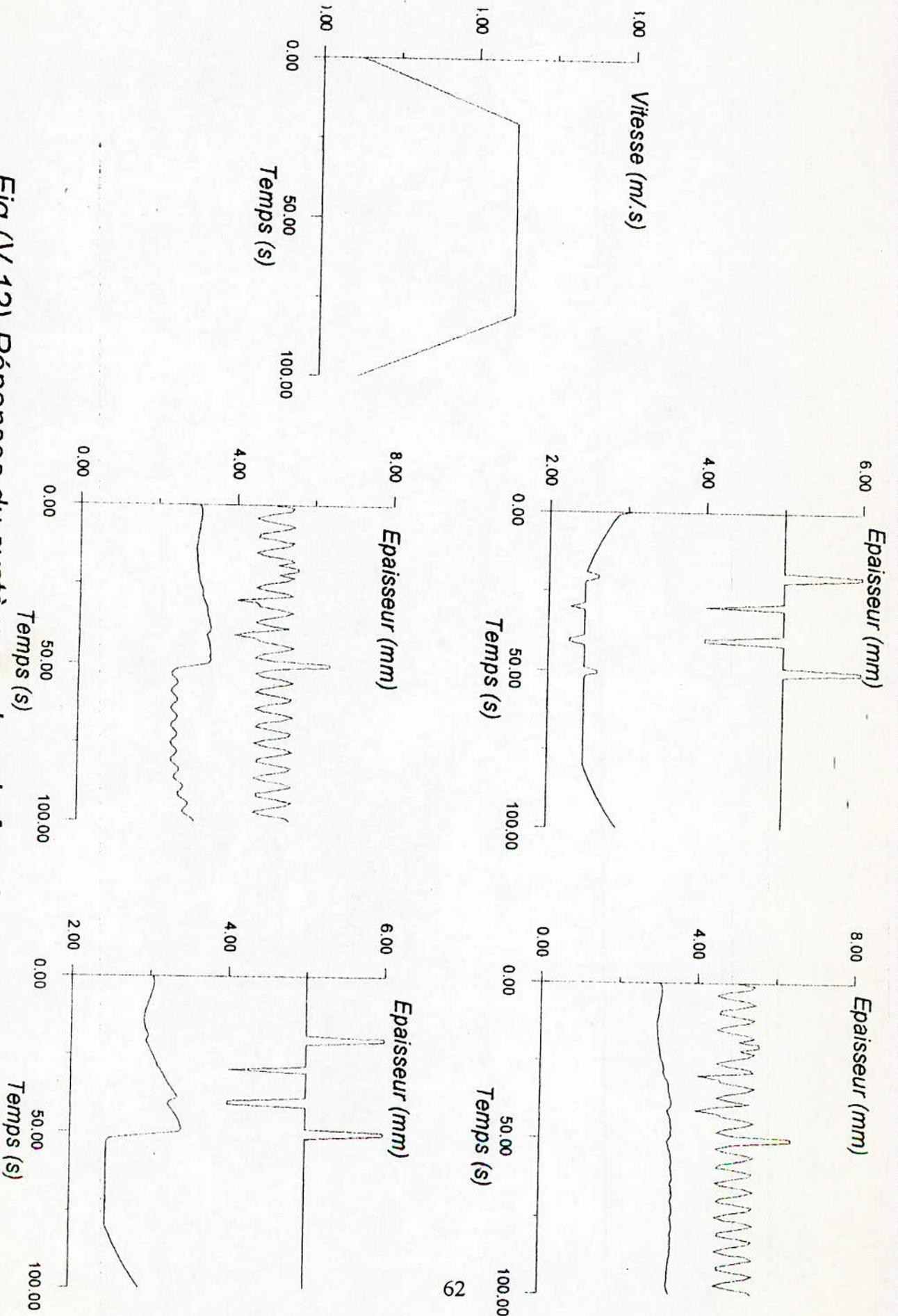


Fig (V.12)-Réponses du système en boucle fermée à vitesse variable

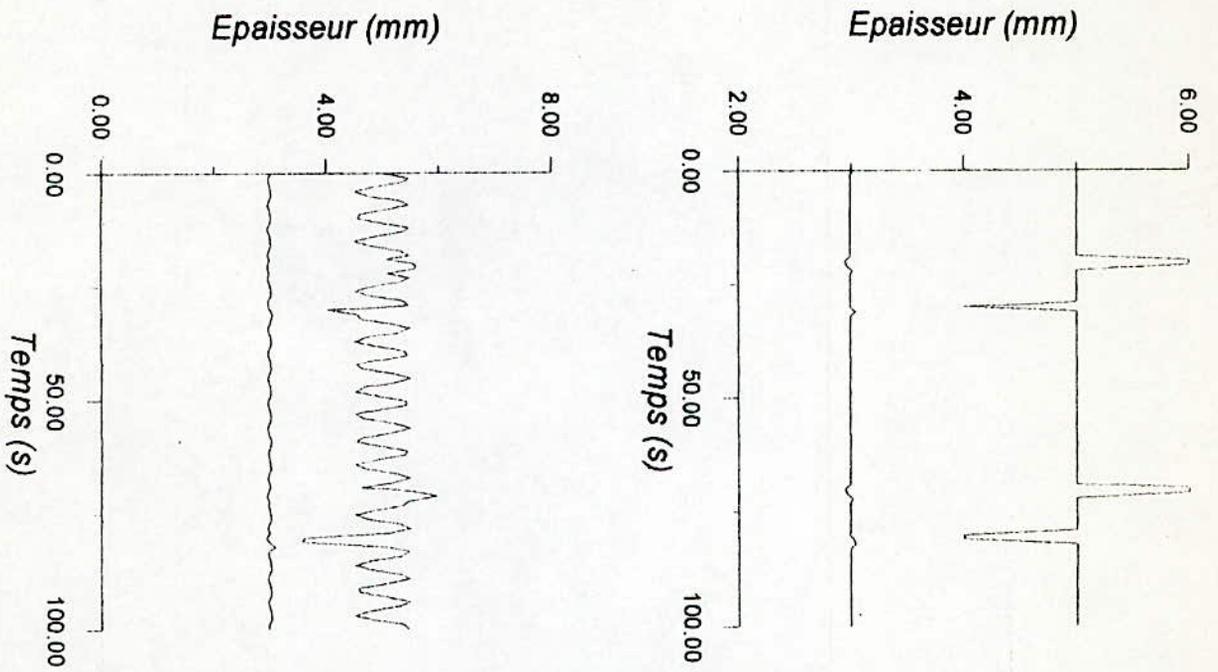
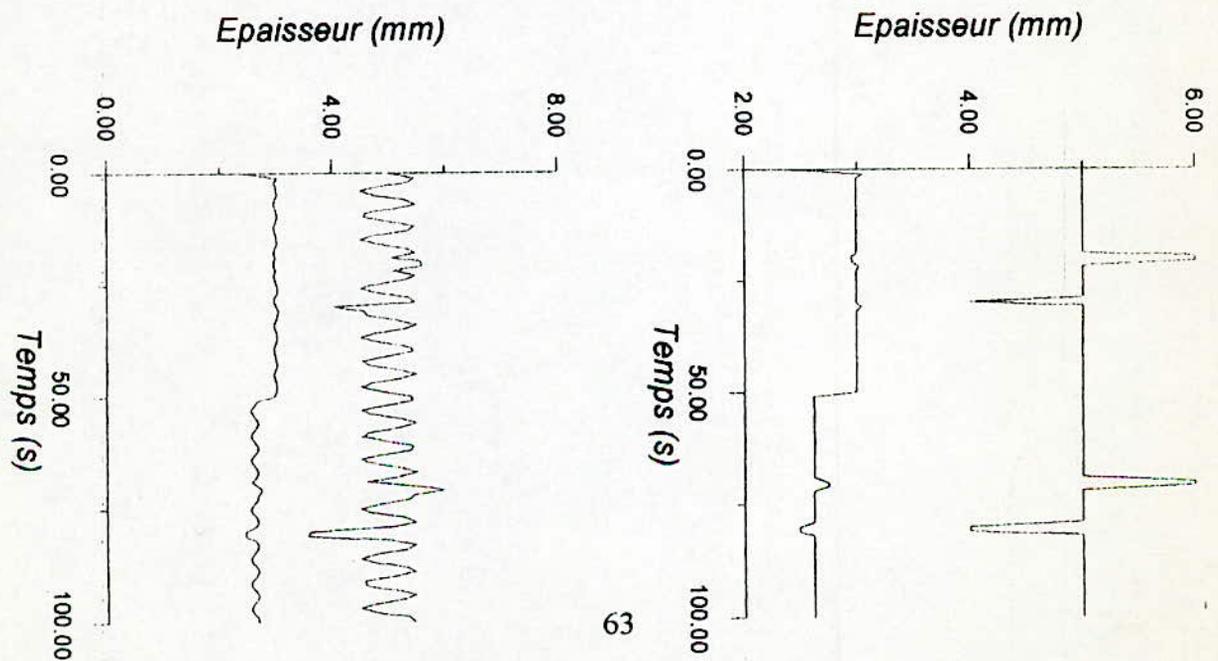
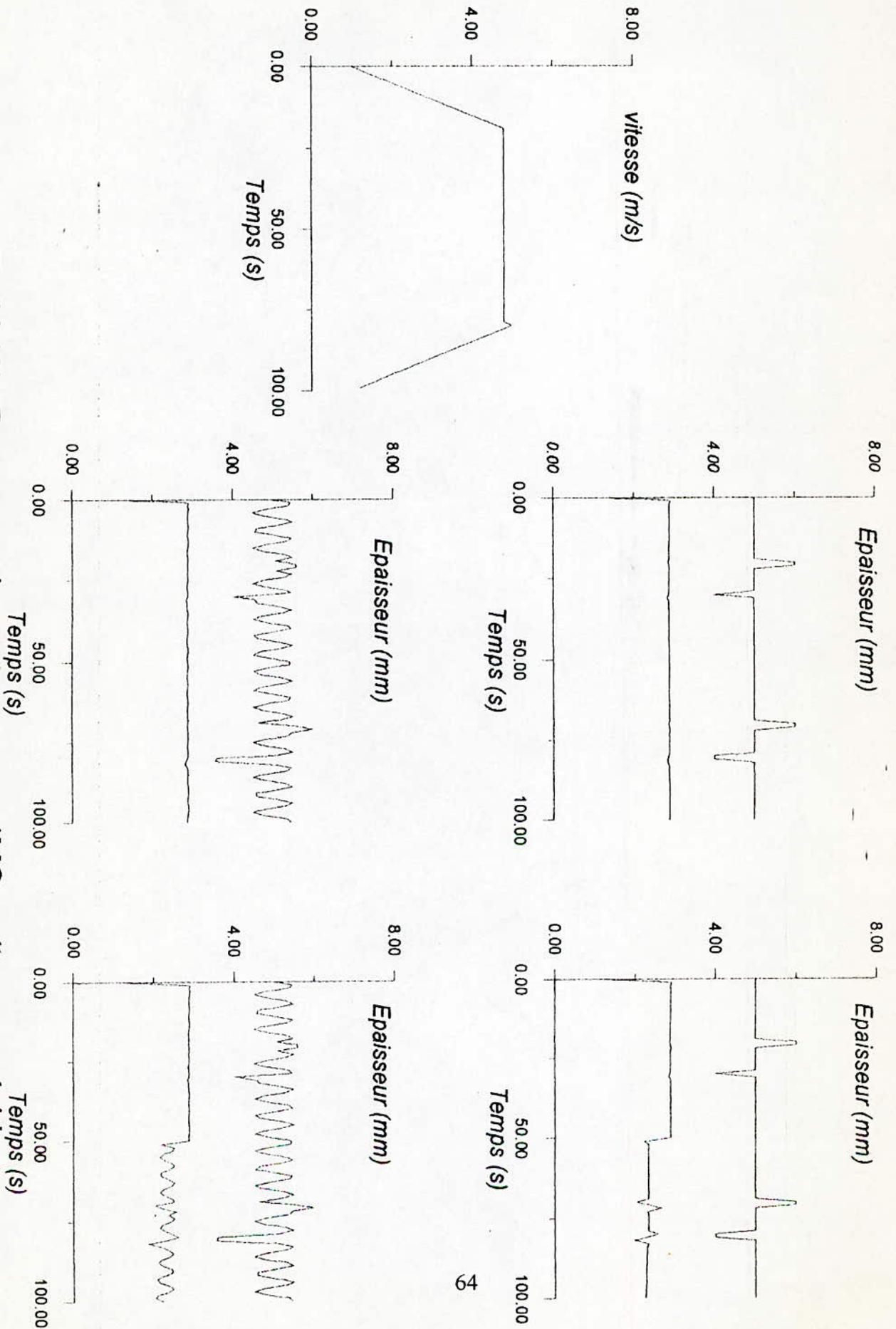


Fig (V. 16)- Reponses du systeme avec IMC





Fig(V. 18) -Reponses du systeme avec IMC a vitesse variable

### 4.3. Commande à modèle interne

#### 1- Vitesse constante

La structure de la commande à modèle interne est convenable pour la commande des systèmes non linéaires grâce à ces importantes propriétés .

La structure de base de IMC est montrée par la figure (V.13).

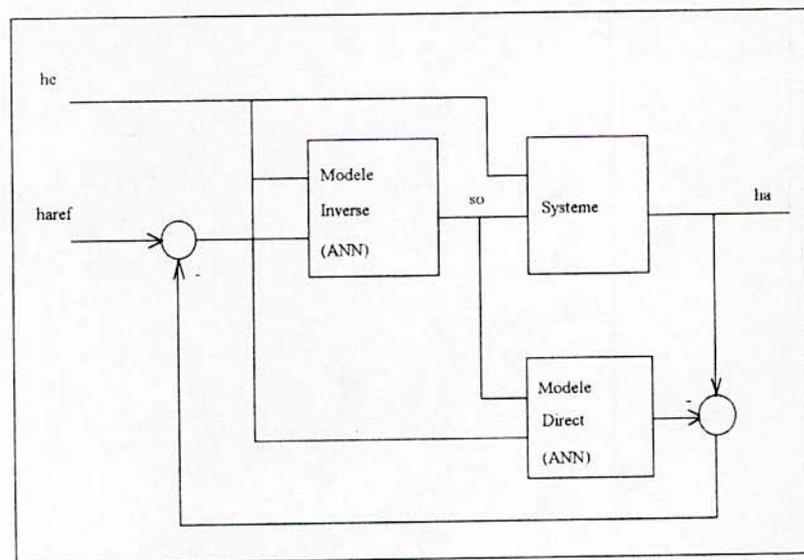


Fig (V.13) Structure de la commande à modèle interne à vitesse constante

Le Modèle est obtenu par une identification du système par le réseau de neurones .

Ce modèle est utilisé pour comparer sa sortie avec celle du système à chaque instant et générer une consigne avec la quelle le système atteint la valeur de référence malgré les perturbations non mesurées qui agissent sur lui.

Les résultats donnés par cette structure sont montrés à la figure (V.14).

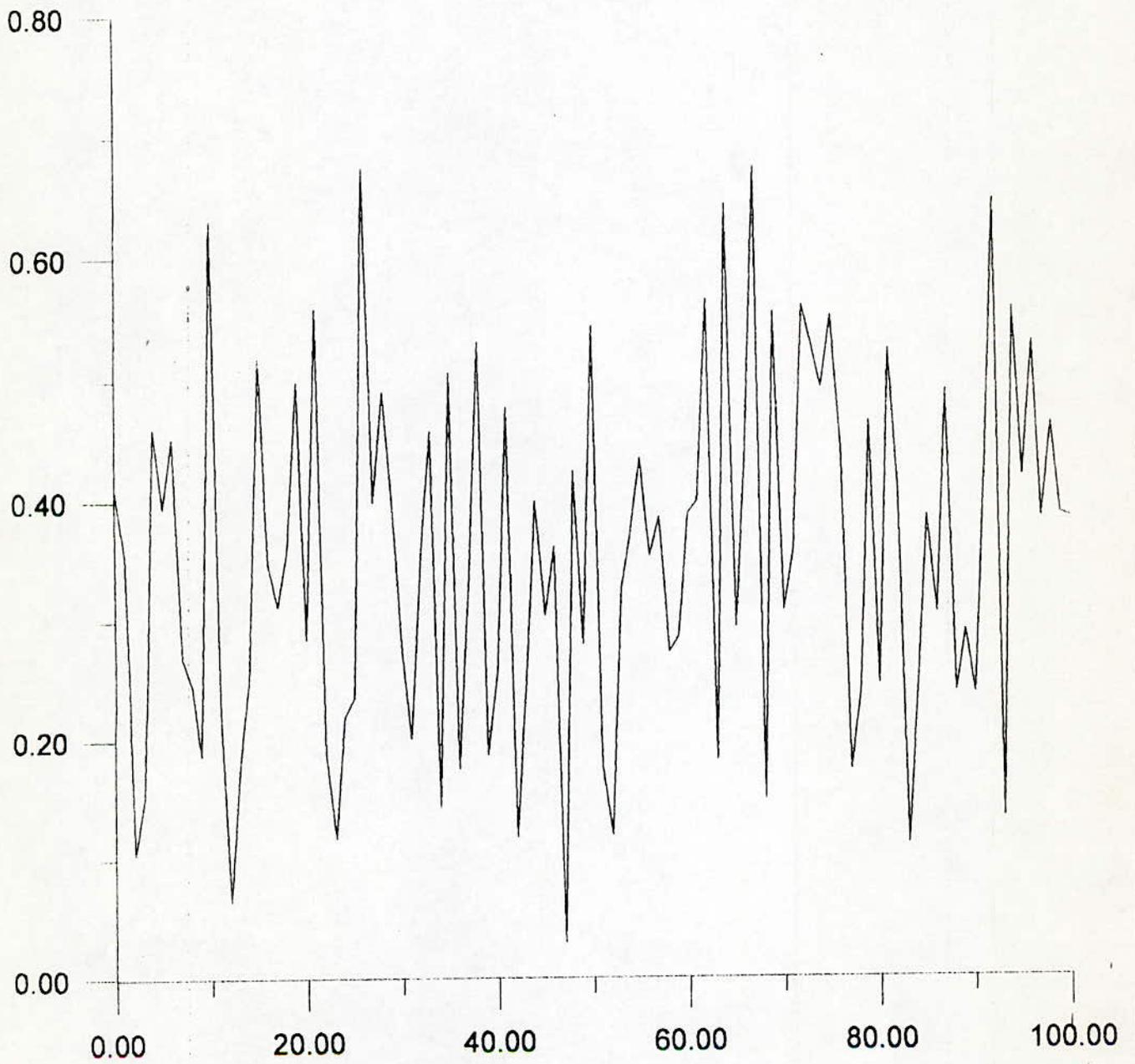


Fig (V.19)- Bruit blanc gaussien (pseudo-blanc)

## 2-Vitesse variable:

On garde la même structure que la précédente ;et on remplace les réseaux précédants par deux autres réseaux ,l'un correspond au modèle direct et l'autre au modèle inverse ,car une nouvelle perturbation est prise en considération ,donc un nouveau apprentissage est effectué cette structure est illustrée dans la figure (V.15).

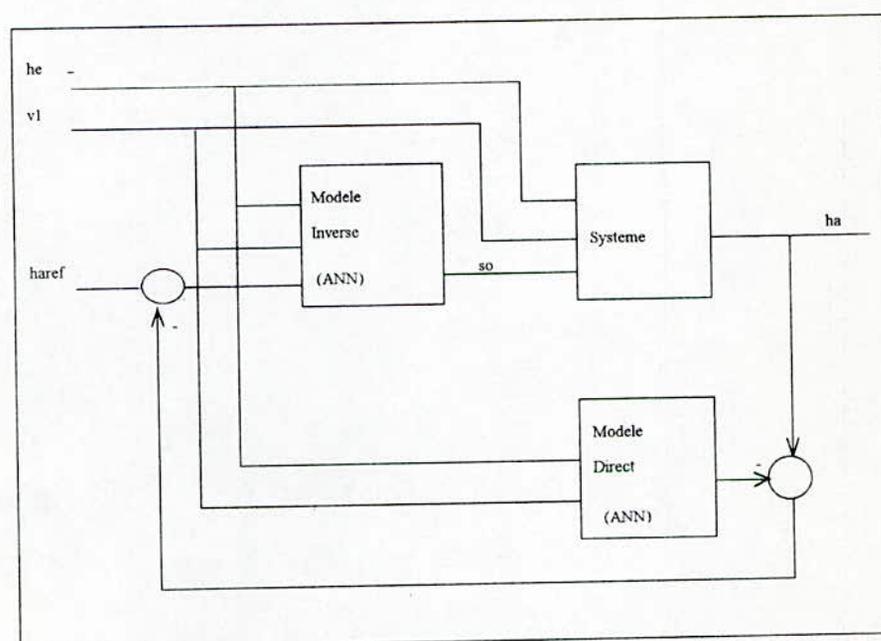


Fig (V.15) Structure de la commande à modèle interne à vitesse variable

La simulation de cette structure a donné des résultats qui sont représentés dans la figure(V.16).

Ces résultats montrent que la commande à modèle interne est plus robuste et donne des résultats meilleurs du point de vue de stabilité, néanmoins elle présente une erreur statique car le modèle inverse n'est pas parfait.

## 5. Test de robustesse

Les appareils de mesure électriques ou électroniques sont souvent soumis à des perturbation qui proviennent de l'extérieur et donnent au correcteur des informations fausses sur l'évolution du système .

Le degré d'influence des perturbations sur chaque structure définit la robustesse de celle ci.

Le signal qui représente mieux les perturbations est le bruit blanc, mais malheureusement ce bruit n'est pas réalisable, on se contente en simulation, d'un signal pseudo-blanc ayant la distribution de gausse de moyenne  $m$  et de variance  $\sigma$  .

la génération de ce signal et sa formule sont détaillés dans l'annexe B.

La figure (V.19) représente un signal aléatoire pseudo-blanc avec une moyenne de zéro et une variance de 0.2

On applique ce signal ( bruit pseudo-blanc ) aux entrées et a la sortie du système pour chaque commande, et on observe l'influence de ce bruit sur la sortie du système commandé.

Les résultats de ce test sont représentés par les figures (V.20) à (V.24).

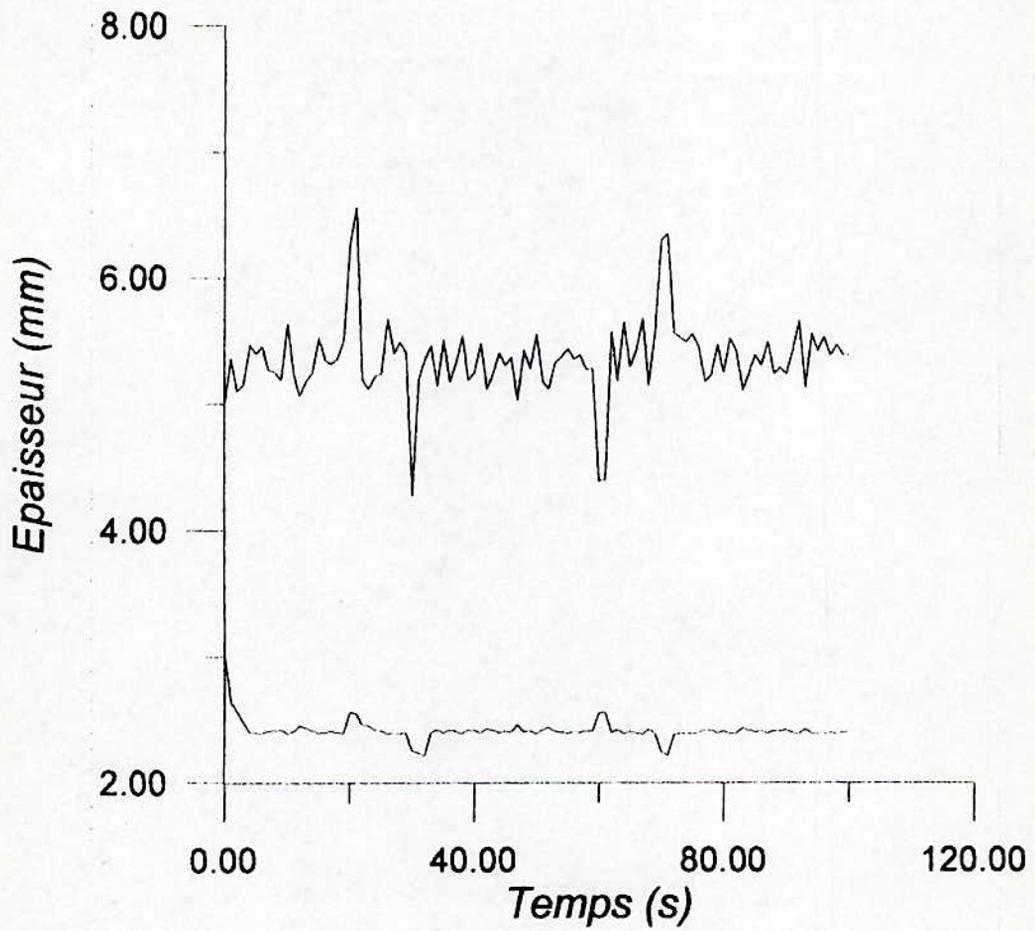


Fig (V.22)-Reponse du systeme en boucle fermée (sortie bruitée)

A partir de ces résultats nous constatons que les réseaux de neurones présentent une bonne robustesse quand il s'agit d'une perturbation aux niveau des entrées du réseau, mais quand il le bruit perturbe la sortie du système certaines structures perdent leurs robustesse, a l'exception de la commande à modèle interne qui arrive toujours à rejeter ce bruit.

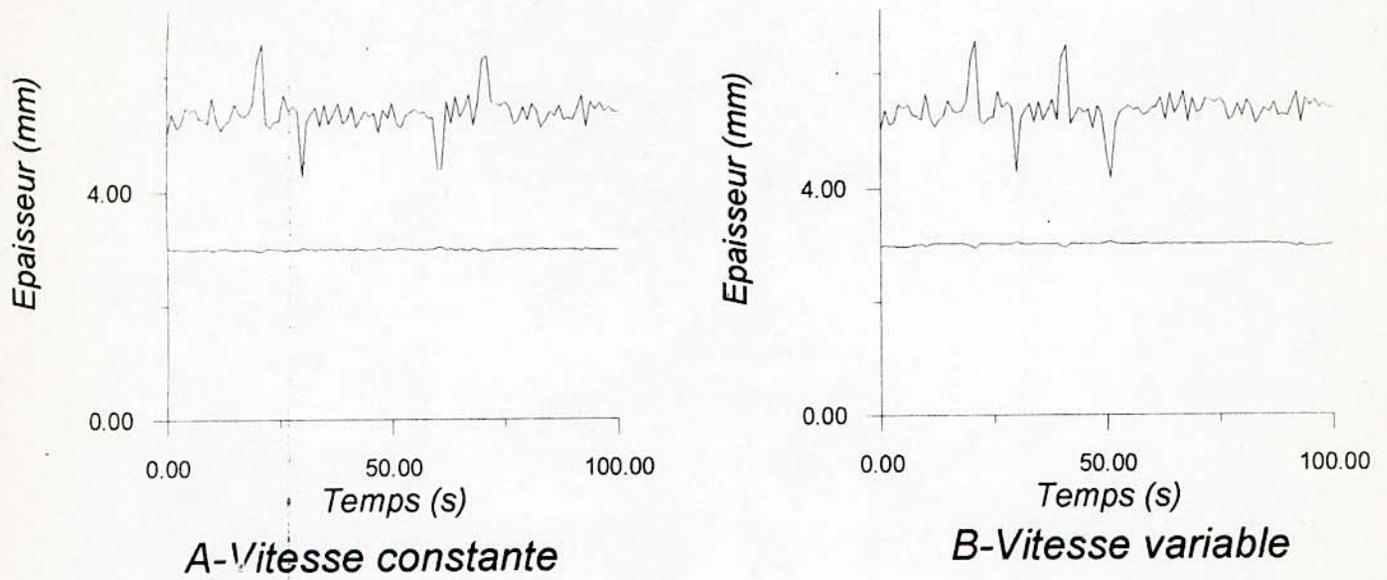


Fig (V.20)- Reponses du systeme on boucle ouverte avec entree bruitee

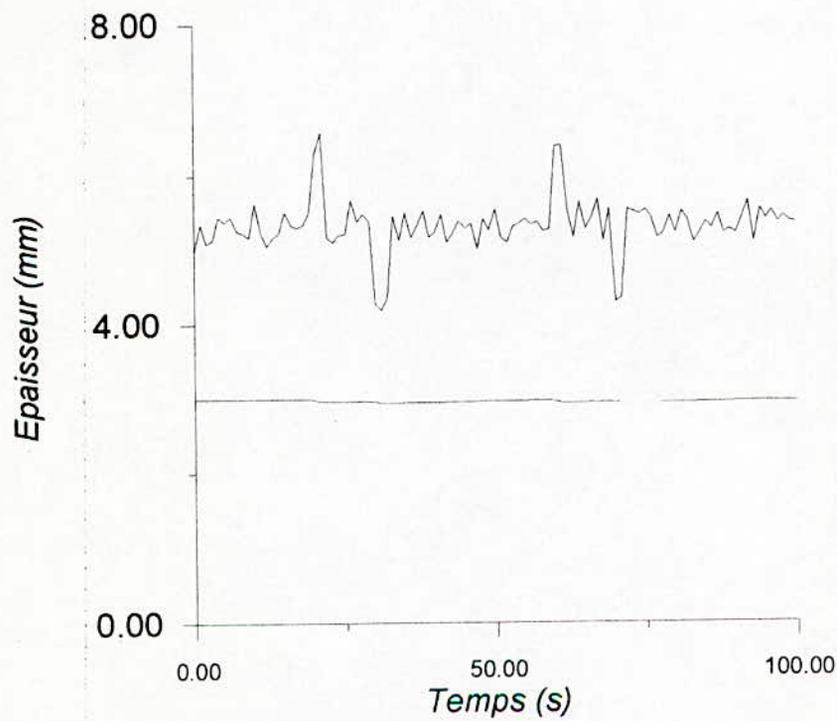


Fig (V.21)- Reponses du systeme on boucle ferme avec entree bruitee

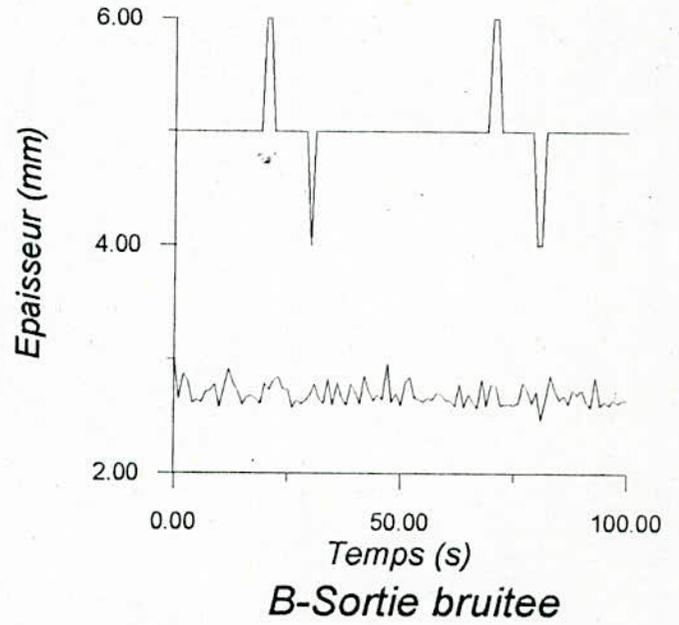
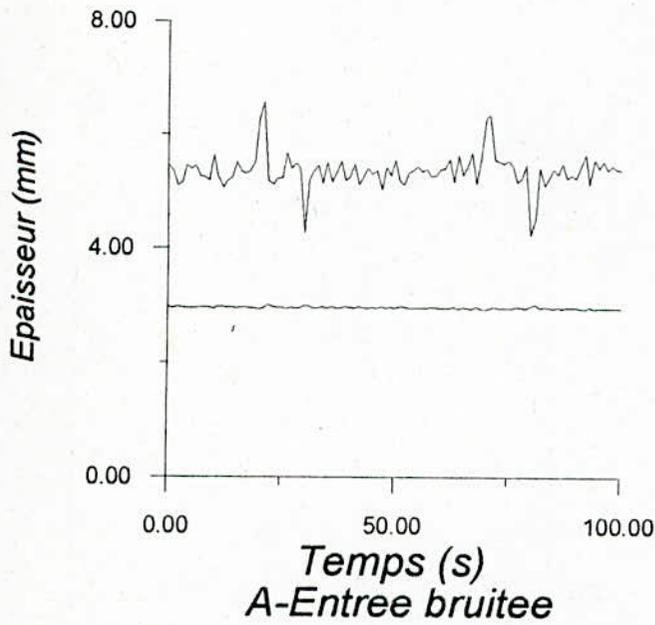


Fig (V.23)-Réponses du systeme a vitesse variable avec IMC

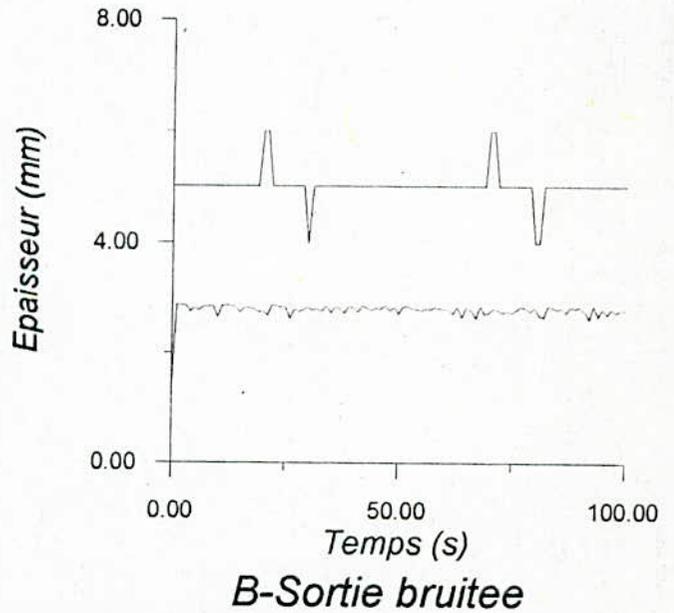
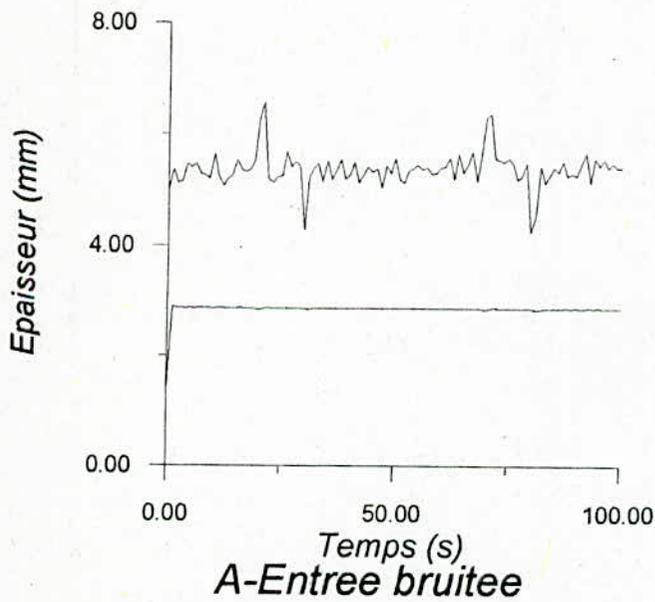
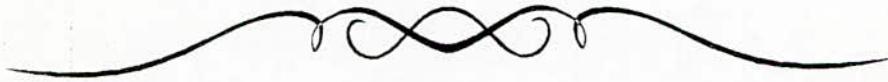


Fig (V.24)-Reponses du systeme a vitesse constante avec IMC



## **Conclusion générale**

## **Conclusion générale**

Dans ce travail, on a orienté notre étude vers la régulation de l'épaisseur d'une tôle à la sortie d'un train de laminoir, et ceci on introduisant la technique de commande par réseaux de neurones.

Cette étude nous a permis de juger sur les performances indéniables des ANN et leurs grande capacité de simuler certaines fonctions non linéaires, de modéliser ces systèmes pour ensuite pouvoir les commander sans pour autant les linaires autour d'un point de fonctionnement.

En outre, les réseaux de neurones nous offert un large éventail de structure de réglage, nous en avons utilisés les structures suivantes : feedforward, feedback et commande à modèle interne. Nous avons pu vérifier leurs robustesse lors d'une perturbation à l'entrée du réseau de neurone, néanmoins le bruit de mesure persiste ce qui a occasionné le rejet de la perturbation, sauf pour la commande à modèle interne qui nous a donné entière satisfaction. En général la technique de commande par réseau de neurones présente l'intérêt d'être plus pratique que la commande classique ( commande anticipatrice ) qui perd toute son efficacité au delà du point de fonctionnement.

En fin, nous avons réussi à introduire la régulation par les ANN à des systèmes complexes, ceci nous a permis d'éviter la tache fastidieuse de synthétiser une commande non linéaire qui s'avère fort délicate et compliquée.



**Annexes**

## ANNEXE A

### MANUEL DU NWP

Le logiciel neuronal network profesional NWP est un logiciel qui permet la conception et la simulation des réseaux de neurones. Il contient tous les methodes et les algorithmes d'apprentissage connus.

Pour constuire un reseaux avec le NWP et le entrainer par la suite les etapes suivantes sont à respecter:

#### Etape I

Avant de lancer le nw.exe , il faut au prealable créer deux fichiers , lun sert à l'apprentissage du reseau et l'autre au test de validation des resultats obtenus du reseau construit .

#### Etape II

Un menu apparait los du lancement du NW.EXE composé des commandes suivantes

- File
- NetWork
- Instrument
- I/O
- Run

### **File**

Cette commande contient tous ce qui concerne le fichier New, Open, Save, Save as, ..., Quit.

### **NetWork**

Cette commande contient tous ce qui concerne le reseau ,parmis ses commandes Back-Propagation

Cette commande permet de préciser sur le reseau que vous créez le

- Le nombre de couches cachées souhaitées dans le reseau
- Le nombre de neurones dans chaque couche cachées
- Le nombre de neurones d'entrées et de sortie
- La methode d'apprentissage et se suivant les fonctions d'activations de sorties (lineaire ,tanghy, ... etc)
- Le mode d'apprentissage
- La selection des fichiers (d'app/test) afin que le reseau permet de comparer entre le resultat itéré et le resultat imposé dans le fichier d'apprentissage/

### **Instrument**

Pour selectionner la sortie du reseau où du fichier par exemple et voir leur évolution en cour de l'apprentissage

### **I/O**

Pour la normalisation des entrées et sorties du fichier d'apprentissage

## Run

Cette commande lance l'apprentissage du reseau apres avoir fixé le nombre d'iterations où l'erreur acceptable

Une fois le reseau est construit et vous voulez modifier l'archetecture du reseau,un deuxieme Menu vous permetra tous se que vous voulez

## Menu 2

Ce menu contient les choses suivantes:

PE, LAYER, GRAPH, etc

### PE

Cette commande concerne les modifications à porter sur les neurones à l'aide des moduls suivants:

Add: ajouter un neurones

Edit: modification d'un neurones (exp: fonction d'activation )

Delete: supprimer un neurone

### LAYER

Les memes moduls que pour le neurone sont permise pour une couche

### GRAPH

La meme chose pour un instrument

Connections entre couche et couche , connection entre un neurone et un neurone ,un neurone et une couche ou bien une couche et un neurone.

A la fin

Une fois les iterations sont terminées,on peut voir les poids attribuées  $W_{ij}$  entre neurones et ce à l'aide du module Edit de la commande PE du deuxieme MENU.

## ANNEXE B

### Generation de signaux à distribution uniforme

$$x(k+1)=[Ax(k)] \text{ Mod } 2^{n-1}$$

où  $x$  et  $A$  sont des entiers

$n$  le nombre de bits du mot binaire du calculateur

Pour generer les  $x(k)$ , il faut choisir  $x(0)$  nombre initial (impair)

$$A=8m+3$$

avec  $m$  entier et  $A$  choisi de l'ordre de  $2^{(n-1)/2}$

Ces conditions assurent la periodicit  maximum de  $2^{n-3}$  de la sequence pseudo-aleatoire

Pour obtenir une variable-aleatoire (generee) uniformement repartie sur  $[0, 1]$ , il faut deviser le resultat par  $2^{n-1}$

exemple:

$$A=131, x(0)=12375, n=16$$

### Generation de sinusoides

Un signal sinusoidal est caracteris  par deux parametres:

- 1/ son amplitude
- 2/ sa frequence

Un signal sinusoidal est s'engendre par la relation

$$s(k)=A\text{Sin}(2\pi (I/M)k)$$

$A$  represente l'amplitude

$I/M$  represente la frequence du signal

**Generation d'un bruit blanc gaussien(pseudo-blanc)**

$$y(k) = \sqrt{-2\sigma^2 \log \frac{1}{x(k)}}$$

$y(k)$ : un signal pseudo-aleatoire ayant la distribution de *Reyleigh*

$$z(k) = y(k) \cos[2\pi x(k+1)] + m$$

où  $x(k)$  est le signal pseudo-aleatoire à distribution uniforme de GAUSS de moyenne  $m$  et de variance  $\sigma$ .

# ERRATA

page	ligne	Correction
- 10	19	Couples .
- 39	18	$\sigma$ : coefficient d'attraction .
- 43	16	série ou parallèle .
- 50	figure V.3.	ha réf : la consigne he : épaisseur d'entrée .

\* Paramètres du système :

$$M = 500 \text{ tonnes / mn}$$

$$J_0 = 250$$

\* Commande anticipatrice :

page 46

ligne 17

$$u = 0.2$$

$$v = 1.2$$

\* Les paramètres du PI :

$$K_p = 0.324 .$$

$$K_I = 10 .$$

## Bibliographie

- [1] A.FREEMAN & M.SKAPURA, " Neural networks ", Addison-Wesley publishing company, 1992.
- [2] P.NAÏM & DAVALLO, " Les réseaux de neurones " Edition Eyrolles, 1990.
- [3] K.J.HUNT & All, " Neural networks for control systems-a survey, "
- [4] K.S.NARENDRA & K.PATHASARATHY, " Identification and control of dynamical systems using neural networks, " IEEE Trans. on neural networks, vol.1, n°1, pp.4-27., Mar.1990.
- [5] W.LI & J.J.E.SLOTINE, "Neural network control of unknown non linear systems ", TA6-9:00
- [6] G.MICHAUD & Y.LECLECQ, " Le tandem à quatre cages ", La metalurgie, vo.105, n°5, Mai.1973.
- [7] M.GHOUALI & All, " Pénalité déplacée en boucle fermée appliquée à l'optimisation du fonctionnement d'un laminoir à froid", APII,1990.
- [8] M.BONAUD, " Les laminoir à chaud" Technique de l'ingenieur 1990.
- [9] D.SBARBARO-HOFER & All, " Neural control of steel rolling mill", IEEE, Jun.1993.
- [10] B.WIDROW & M.A.LEHR, " 30 years of adaptive neural networks: perceptron, madaline and backpropagation, " Proceedings of the IEEE, vol.78, n°9. pp.1415-1440, Sep.1990.
- [11] " Working with neural networks, " IEEE Spectrum, pp.46-53, July.1993: *Dan Hammerstrom*
- [12] P.J.WERBOS, " Backpropagation through time, what it does and how to do it, " Proceedings of the IEEE, vol.78, n°10, pp.1550-1560, Oct.1990.
- [13] A.MADANI & Y.OUSSAR, "Etude de la commande neurolinguistique , application SISO et MIMO.", P.F.E, Ecole Nationale Polytechnique, 1993.