

REPUBLIQUE ALGERIENNE DEMOCRATIQUE ET POPULAIRE

ECOLE NATIONALE POLYTECHNIQUE

DEPARTEMENT : GENIE ELECTRIQUE

المدرسة الوطنية المتعددة التقنيات  
BIBLIOTHEQUE — المكتبة  
Ecole Nationale Polytechnique

SPECIALITE : AUTOMATIQUE

PROJET DE FIN D'ETUDES

THEME

**LA COMMANDE INVERSE  
PAR RESEAUX DE NEURONES  
DE LA MACHINE ASYNCHRONE**

PROPOSE ET DIRIGE PAR:

M<sup>er</sup> H. CHEKIREB.

ETUDIE PAR:

Mlle CHITOUR FATIMA

Mlle KAHLOUCHE AICHA

PROMOTION JUILLET 1997

REPUBLIQUE ALGERIENNE DEMOCRATIQUE ET POPULAIRE

ECOLE NATIONALE POLYTECHNIQUE

DEPARTEMENT : GENIE ELECTRIQUE



SPECIALITE : AUTOMATIQUE

THEME

**LA COMMANDE INVERSE  
PAR RESEAUX DE NEURONES  
DE LA MACHINE ASYNCHRONE**

PROPOSE ET DIRIGE PAR:  
M. H. CHEKIREB.

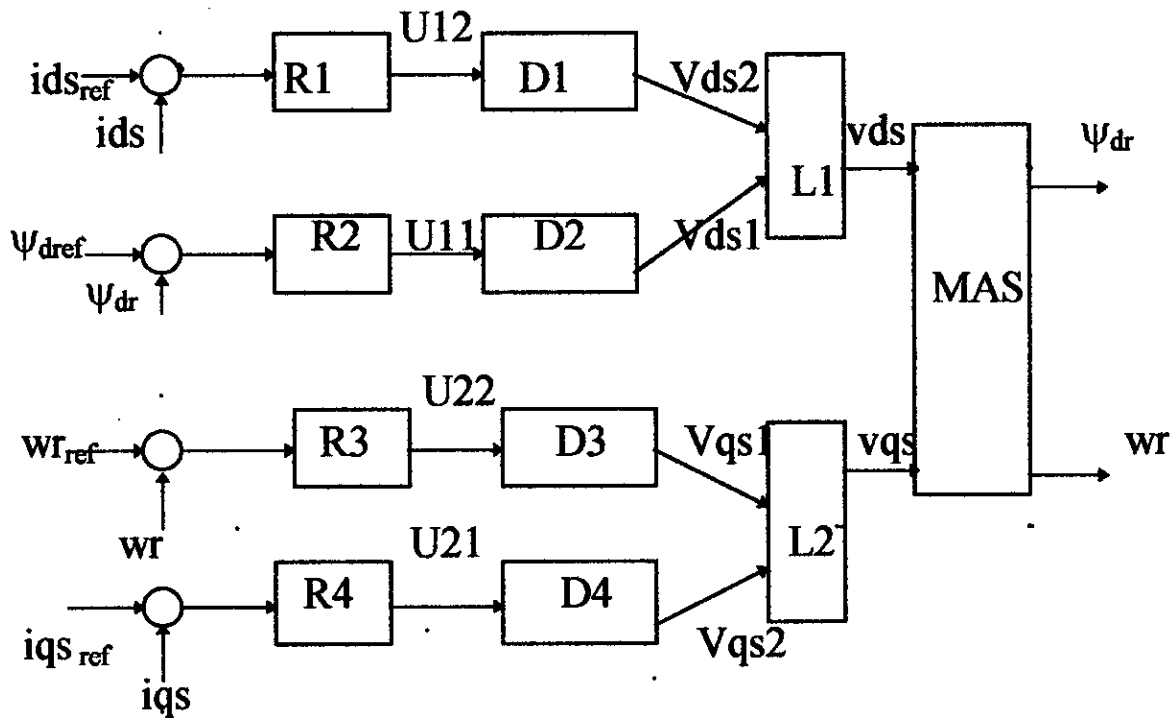
ETUDIE PAR:  
Mlle CHITOUR FATIMA  
Mlle KAHLOUCHE AICHA

PROMOTION JUILLET 1997

• page 13

dans notre cas un choix de poids synaptiques.

• page 41



المدرسة الوطنية المتعددة التقنيات  
BIBLIOTHEQUE — المكتبة  
Ecole Nationale Polytechnique

# DEDICACES

بِسْمِ اللَّهِ وَالْحَمْدُ لِلَّهِ وَالصَّلَاةُ عَلَى رَسُولِ اللَّهِ .

إلى أمي .

إلى أبي .

إلى إخوتي وأخواتي .

إلى عائلتي وكل عائلتيها .

إلى كل صديقاتي خاصة: سميرة وسعيدة .

إلى أمشي، إلى أخيارها وشرافاتها .

إلى الذين مضوا في صمت .

إليهم جميعا أهدي هذا العمل  
فالهمة .

**A MON TRES CHER PERE, J'OFFRE LE FRUIT DE SON SACRIFICE  
A MA TRES CHERE MERE, QUI MA SOUTENUE DURANT TOUTE  
MA VIE ESTUDIANTINE.**

**A MES FRERES .**

**A MES BEAUX FRERES.**

**A MES SOEURS.**

**A TOUTE LA FAMILLE CHITOUR SPECIALEMENT FATIMA.**

**A TOUTE MA FAMILLE ET MES AMIS.**

**JE DEDIE CE MODESTE TRAVAIL.**

**AICHA.**

المدرسة الوطنية المتعددة التخصصات  
BIBLIOTHEQUE — المكتبة  
Ecole Nationale Polytechnique

# REMERCIEMENT

## REMERCIEMENT

Nos remerciements s'adressent à tout les enseignants qui ont contribué à notre formation ainsi qu'à notre encadreur,  
Monsieur H.CHEKIREB.

Nous tenons à remercier:

Monsieur D.BOUKHETALA. NAIMA et YACINE KHALEF  
pour leurs aide précieuse.

SABINA, HOURIA, KRIMO, SALAH, ainsi que tous le personnel  
de la bibliothèque pour nous avoir faciliter notre recherche  
bibliographique.

A tous ceux qui ont participé de près ou de loins à l'élaboration  
de ce mémoire.

Nous finissons en saluant la promotion d'automatique de  
JUILLET 1997

المدرسة الوطنية المتعددة الفنون  
BIBLIOTHEQUE — المكتبة  
Ecole Nationale Polytechnique

# SOMMAIRE



## Sommaire :

المدرسة الوطنية المتعددة التقنيات  
المكتبة — BIBLIOTHEQUE  
Ecole Nationale Polytechnique

### Introduction générale

#### Chapitre 1

#### La théorie des réseaux de neurones

1- Introduction	1
2- Le neurone biologique	1
3- Les éléments constitutifs d'un réseau de neurones	2
4- Le modèle formel de neurone	3
5- Les synapses	4
6- La fonction d'activation	5
7- Les propriétés des réseaux de neurones	8
8- Les différents modèles des réseaux de neurone	8
9- L'architecture d'un réseau de neurone	9
10- L'apprentissage des réseaux de neurones	11
11- Différents types d'apprentissage des réseaux de neurones	12
12-1- La technique du descente de gradient	13
12-2- L'optimisation aléatoire MOA	14
13- Apprentissage des réseaux de neurones statiques	14
13-1- Rétro-propagation de l'erreur	16
13-2- La technique du Momentum	18
14- Apprentissage des réseaux de neurones dynamiques	19
15- Problèmes d'apprentissage	20
16- Apprentissage et généralisation	21
17- Réalisation	22
18- Applications industrielles	23
Conclusion	25

#### Chapitre 2

#### Modélisation de la MAS

1- Introduction	26
1- Modélisation de la MAS	26
1-1 Mise en équations	26
1-2 Transformation de PARK	28
1-3 Mise sous forme d'état	30
1-4 Choix du référentiel	30
1-5 Théorie du flux orienté	31
Conclusion	32

**Sommaire :**

<b>Chapitre 3</b>	<b>Commande inverse de la MAS à orientation parfaite du champ</b>	
Introduction		34
1- Identification inverse		34
1-1 La 1 <sup>er</sup> structure		34
1-2 La 2 <sup>eme</sup> structure		35
1-3 La 3 <sup>eme</sup> structure		36
2- La commande inverse		36
3- La commande inverse de la MAS à orientation parfaite du champ rotorique		35
3-1 Commande par bouclage linéarisant		37
3-2 Dimensionnement du réseau approxinant l'inverse de la MAS		44
3-3 Synthèse des régulateurs neuronaux		44
3-4 Simulation et commentaires		46
3-5 Tests de robustesse		51
Conclusion		51
<b>Chapitre 4</b>	<b>Commande inverse de la MAS à orientation réelle du champ</b>	
Introduction		56
1- Modèle de la MAS alimentée en tension dans le repère synchrone		56
2- Synthèse du réseau ws		57
3- Structure de commande		57
4- Résultats de simulation		58
5- Tests de robustesse		58
Conclusion		67
<b>Conclusion générale</b>		
<b>Annexe 1</b>		
<b>Annexe 2</b>		
<b>Bibliographie</b>		

المدرسة الوطنية المتعددة التقنيات  
BIBLIOTHEQUE — المكتبة  
Ecole Nationale Polytechnique

# INTRODUCTION GENERALE

## **Introduction générale**

---

**L'**idée des réseaux de neurones consiste à modéliser le cerveau humain, en se basant sur l'architecture et le fonctionnement de ce dernier, dans le but de subvenir aux besoins grandissant de traitement automatique de l'information.

L'approche neuronale n'est qu'une combinaison entre les principes du neurones biologique d'une part, et les lois de la physique et les mathématiques d'autres part.

Notre travail portera sur l'application de la commande inverse par réseaux de neurones à la machine asynchrone alimentée en tension, vu l'importance et l'aspect non linéaire du modèle mathématique de cette dernière.

Notre choix de la machine asynchrone est dû à sa grande utilisation dans différents domaines de l'industrie, caractérisée par l'absence de collecteurs, elle peut fonctionner soit en génératrice soit en moteur, avec une grande robustesse électrique et mécanique.

Ce travail est partagé en quatre chapitres organisés comme suit :  
Dans le premier chapitre, la théorie des réseaux de neurones est présentée, avec ces différentes applications industrielles.

Dans le second, une modélisation de la machine asynchrone est présentée dans différents référentiels, le découplage entre le couple et le flux est réalisé par l'orientation du flux.

Le troisième chapitre, est consacré à l'application de la commande inverse à la MAS à orientation parfaite du champ, des résultats de simulation ainsi que des tests de robustesse sont présentés.

Dans le dernier chapitre, de nouveau, la commande inverse est appliquée à la MAS, mais cette fois dans le but de réaliser l'orientation réelle du champ.

Enfin, on termine notre travail par une conclusion générale.

---

# CHAPITRE . I

## 1-INTRODUCTION

**L'**étude des réseaux de neurones date des années 1940-50 . Ces réseaux sont des assemblages fortement connectés des neurones .

Le premier modèle de neurone est dû à deux neurobiologistes Américains , Warren McCulloch et Walter Pitts qui supposèrent que l'impulsion nerveuse était le résultat d'un calcul simple effectué par chaque neurone et que grâce à l'effet collectif d'un réseau de neurones interconnectés que naît la pensée.

Au même moment Donald Hebb proposait une hypothèse qui associait l'activité des neurones à l'efficacité des synapses qui les relie [11].

Les travaux sur les réseaux de neurones formels ont connu des débuts promoteurs vers la fin des années 50, mais un certain manque d'approfondissement de la théorie, et le mouvement concurrent de l'intelligence artificielle , ont conduit à une période de relative inactivité sur le connexionnisme . Assez peu d'articles furent publiés entre 70 et 80 en raison de la faible quantité de chercheurs intéressés .

Depuis le début des années 80 , le connexionnisme connaît un regain d'activité qui est dû à plusieurs facteurs dont :

- La découverte de modèle neuromimétique de John Hopfield .
- La découverte de modèle neuromimétique capable de dépasser les limites du perceptron , de l'Adaline et de la machine de Boltzman .
- Le développement extraordinaire des moyens de l'informatique.

Aujourd'hui le réseau de neurones fait l'objet principal d'étude et outils applicatifs , il joue un rôle dans la recherche et dans l'industrie.

L'intérêt d'un réseau de neurones réside en sa capacité de résoudre les problèmes pratiques.

### 2- Le neurone biologique [9],[11] :

Le neurone est l'élément de base du système nerveux qui possède à peu près 100 à 1000 milliards de neurones . C'est une cellule constituée de trois parties , voir fig1-1 :

- Le corps cellulaire.
- Les dendrites .
- L'axone

#### Le corps cellulaire :

On l'appelle aussi soma . Il fait quelques microns de diamètres , contient le noyau de neurone , sa forme est pyramidale ou sphérique .

---

**Les dendrites :**

Ce sont des fines extensions tubulaires de quelques dizaines de microns de longueurs et de quelques dixièmes de microns de diamètres . Leurs rôle est de recevoir les signaux .

**L'axone :**

C'est la fibre nerveuse, son diamètre est de l'ordre de quelques millièmes de millimètres , c'est un moyen de transport pour les signaux émis par les neurones .

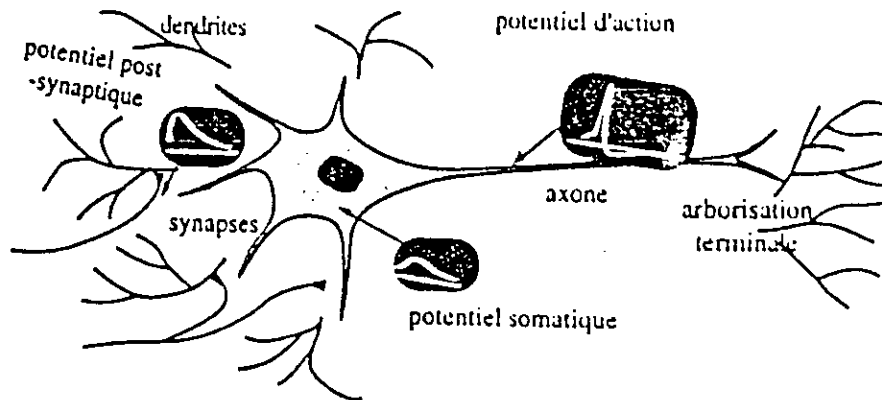


Fig.1.1 : Cellule nerveuse.

Le neurone reçoit des signaux des neurones voisins , il intègre ces signaux , engendre un influx nerveux , le conduit et le transmet à un autre qui peut le recevoir .

L'influx nerveux est une impulsion électrique , et les signaux dendritiques et somatiques sont des variations de potentiel électrique .

La transmission entre deux neurones se fait à l'aide d'un médiateur chimique appelé neurotransmetteur .

Le soma recueille l'ensemble des informations reçues par les dendrites et fait leurs somme . Si le potentiel somatique dépasse un certain seuil , il y'a émission d'un potentiel d'action . Ce signal très bref (1ms) est transmis le long de l'axone sans atténuation , et réparti sur les synapses des neurones cibles grâce à l'arborisation terminale [9].

En général , le système nerveux est formé par des neurones qui sont connectés les uns aux autres suivant des répartitions spatiales complexes . Les connexions entre deux neurones se font en des endroits appelés synapses où ils sont séparés par un petit espace synaptique de l'ordre d'un centième de microns .

**3-Les éléments constitutifs d'un réseau de neurones [9] :**

Un réseau de neurones est un graphe orienté et pondéré . Les noeuds de ce graphe sont des automates simples nommés neurones , et dotés d'un état interne , l'activation . Cette dernière se propage dans le graphe le long d'arc pondérés appelés liens synaptiques .

La règle qui détermine l'activation d'un neurone est appelée fonction d'activation .

**4-Le modèle formel de neurone [9],[11],[6] :**

Le modèle théorique du neurone trouve ces origines dans la neurobiologie .

C'est en 1943 que McCulloch et Pitts ont émis l'idée simplificatrice du neurone formel, c'est-à-dire un opérateur binaire interconnecté à ses semblables par des synapses excitatrices ou inhibitrices. Une assemblée de tels opérateurs a des propriétés collectives , capable de certains calculs que chacun d'eux séparément est incapable d'exécuter .

Le neurone formel est modélisé par deux opérateurs ( fig1-2) :

- **Opérateur de sommation :** il calcule un potentiel p qui est la somme de ses entrées pondérées par des coefficients appelés poids synaptiques .

$$p = \sum_{i=1} W_i X_i \quad (1-1)$$

- **Un opérateur qui calcule la sortie y du neurone en fonction de son potentiel p .**

$$y = f(p) \quad (1-2)$$

f : fonction de seuillage .

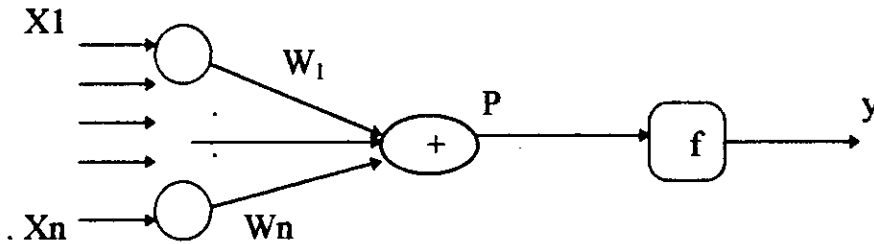


fig1-2 modèle formel de neurone

avec 
$$F(p) = \begin{cases} 1 & \text{si } p > \beta \\ 0 & \text{si } p < \beta \end{cases} \quad (1-3)$$

$\beta$  : seuil de la fonction d'activation .

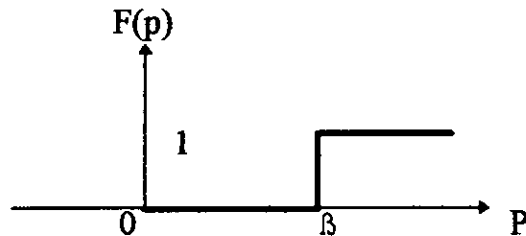


fig 1-3 Fonction de seuillage du neurone



4-1 Modélisation générale : (voir fig 1-4)

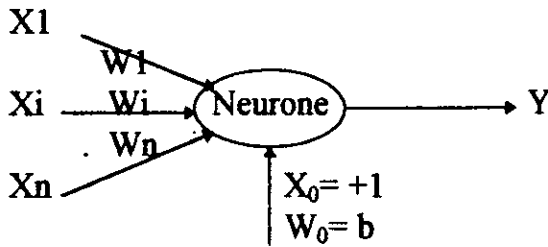


fig1-4 Neurone formel avec biais

avec :

- ( X ) : les entrées du neurone.
- ( W ) : les poids synaptiques du neurone.

On note  $p = H( X_0, X_1, X_2, \dots, X_n )$  : entrée totale du neurone, dans la plus part des cas :

$$H = \sum_{i=0}^n W_i X_i \tag{1-4}$$

b est appelé biais du neurone ,  $W_0=b, X_0=1$ .

5-Les synapses [9],[11] :

La synapse est un mot grec qui signifie point de jonction . Dans le système nerveux on estime en moyen 1000 à 100.000 synapses par neurones .

Hebb a proposé en 1949 , une règle qui montre que le renforcement synaptique intervient lorsqu'il y'a activité du neurone pré-synaptique (cause) et activité du neurone post-synaptique (effet) .

Cette règle est la suivante :

$$W_{ij}( t + \delta t ) = W_{ij}(t) + \eta . A_i . A_j \quad \eta > 0 \tag{1-5}$$

$W_{ij}(t)$  : poids de la connexion entre le neurone j et le neurone i .

$A_i, A_j$  : les activations des neurones i et j .

$\eta$  : paramètre reflétant l'intensité de l'apprentissage .

La plus part des modèles de neurones sont capable de modifier leurs poids synaptiques automatiquement, ils sont dotés de règles d'apprentissage .

En général, on note ,  $W_{ji}$

- j : indice correspondant au neurone cible.
- i : indice correspondant au neurone émetteur .

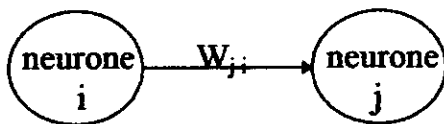


fig 1-5- Notation des poids synaptiques entre deux neurones.

## 6- La fonction d'activation [8],[10],[11] :

Après une présentation du neurone , on va voir la fonction d'activation qui est responsable du comportement du neurone , elle transforme le signal  $X_i$  non borné à l'instant  $t$  en un signal borné  $Y_i$ .

$$Y_i = f( X_i )$$

### 6-1- Les caractéristiques de la fonction d'activation :

- La fonction d'activation peut être à valeurs continues ou à valeurs discrètes : les fonctions d'activation continues sont les plus utilisées , car le réseau traite plus d'informations à la fois dans un seul neurone. ceci augmente la puissance de calcul du réseau, en plus , des fois ,la dérivabilité des fonctions continues est utile .
- la fonction d'activation est en général monotone ,croissante : le fait que la fonction est monotone implique que les neurones sont semi-linéaires .si on choisit la fonction linéaire ,les calculs seront plus faciles ,mais l'inconvénient est que ce type de neurones n'est pas robustes. Par contre ,si la fonction est non linéaire ,le réseau peut alors approximer des fonctions complexes mais l'analyse sera aussi plus complexe .
- la fonction d'activation peut être dérivable ,c'est le cas des fonctions lisses ,et non dérivable pour les fonctions dures , elle est souvent dérivable ,ceci est nécessaire pour les fonctions d'apprentissage .
- fonction à mémoire où sans mémoire (une fonction peut ou non tenir compte de ses valeurs passées pour le calcul de ses valeurs futures).
- fonction déterministes où fonctions stochastiques .
- seuillage : la fonction d'activation possède une valeur ,au delà de cette dernière les autres valeurs sont négligeables .ceci donne au neurone une certaine résistance au bruit.
- saturation :la fonction d'activation peut avoir une valeur maximale au dessus de laquelle sa réponse est fixe. Ceci évite la propagation des grandes valeurs dans le réseau.
- si la fonction d'activation possède des valeurs importantes loin de l'origine ,c'est qu'elle ressemble à un échelon. Dans le cas contraire c-à-d qu'elle possède des valeurs importantes autour de zéro , elle ressemble à un pic .
- si la fonction d'activation est positive ,elle variera entre 0 et 1 ,si par contre elle varie entre -1et 1 sa moyenne est nulle.

### 6-2- Exemples des fonctions d'activation :

on définit :

- $A_i^t$  :l'activation du neurone  $i$  au temps  $t$ .
- $net_i^t$  : son activation pondérée .
- $\theta_i$  : son seuil .

**6-2-1-fonction linéaire :**

c'est la fonction la plus simple, cette simplicité à pour effet la limitation des capacités de calcul du réseau.

$$\begin{cases} A_i = \lambda Net_i \\ \lambda = 1 \end{cases}$$

(1-6)

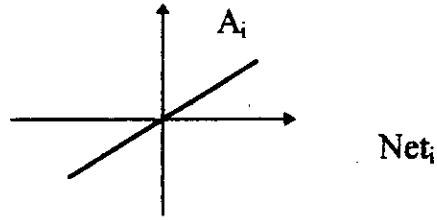


fig 1-6-1 Fonction linéaire

**6-2-2-fonction seuil :**

Elle introduit une non-linéarité dans les neurones .

$$A_i = \begin{cases} 1 \text{ si } Net_i \geq \theta_i \\ 0 \text{ si } \text{autre } Net_i \end{cases} \quad (1-7)$$

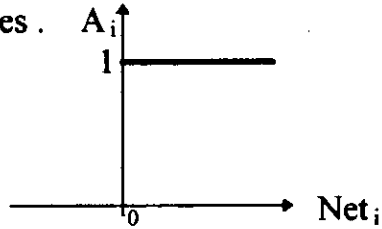


fig 1-6-2 Fonction seuil

**6-2-3 : Fonction linéaire bornée :**

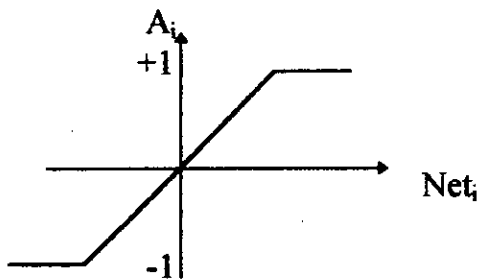


fig 1-6-3 Fonction linéaire bornée

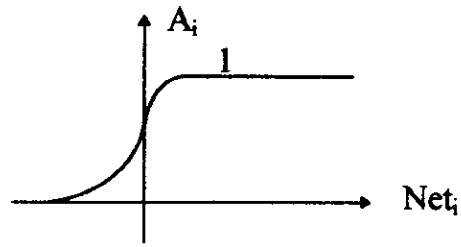
$$A_i = \begin{cases} Max_i \text{ si } Net_i > \theta_i^+ \\ Net_i \text{ si } \theta_i^- \leq Net_i \leq \theta_i^+ \\ Min_i \text{ si } Net_i < \theta_i^- \end{cases}$$

$$\theta_i^+ = Max = +1 \quad (1-8)$$

$$\theta_i^- = Min = -1$$

**6-2-4 : Fonction sigmoïde exponentielle :**

$$A_i = \frac{1}{1 + e^{-Net_i}} \quad (1-9)$$



C'est une fonction continue, sa dérivée est :

fig 1-6-4 Fonction sigmoïde exponentielle

$$\frac{dA_i}{dNet_i} = A_i(1 - A_i) \quad (1-10)$$

**6-2-5 : Fonction sigmoïde tangentielle :**

Utilisée surtout pour les réseaux multicouches.

$$A_i = \tanh(Net_i)$$

$$\tanh(Net_i) = \frac{e^{Net_i} - e^{-Net_i}}{e^{Net_i} + e^{-Net_i}} \quad (1-11)$$

sa dérivée est :

$$\frac{dA_i}{dNet_i} = (1 - A_i^2)$$

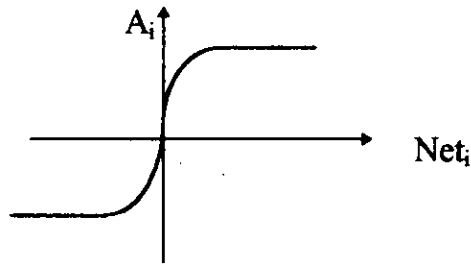


fig 1-6-5- Fonction sigmoïde tangentielle

**6-2-6 : Fonction gaussienne :**

Cette fonction d'activation n'est ni monotone , ni croissante .

Elle est utilisée pour des réseaux spéciaux comme les réseaux RBF( Radial Basis Function ) .

$$A_i = e^{-\frac{Net_i^2}{2 \cdot \sigma^2}} \quad (1-12)$$

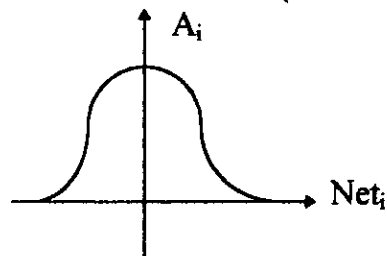


fig 1-6-6 Fonction gaussienne

**6-2-7 : Fonction stochastique :**

Dans ce cas ce n'est pas l'activation de neurone qui est spécifiée , mais plutôt la probabilité que l'activation d'un neurone soit à 1 .

$$p[A_i] = \frac{1}{1 + e^{-\frac{Net_i}{T}}} \quad (1-13)$$

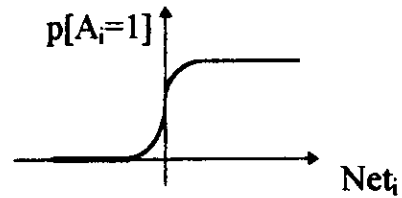


fig 1-6-7 Fonction stochastique ( T = 1)

T : température ( détermine l'importance de l'effet de  $Net_i$  sur le comportement ) .  
En général , la fonction la plus utilisée c'est la fonction sigmoïde qui peut approximer n'importe qu'elle fonction continue.

### 6-3- La propagation de l'activation :

Le comportement d'un neurone dépend du choix de sa fonction d'activation.  
L'activation calculée par cette fonction se propage d'un neurone à l'autre à travers les connexions .

## 7- LES PROPRIETES DES RESEAUX DE NEURONES[10] :

L'intérêt qu'a le réseau de neurones est dû aux propriétés qu'il possède telles que:

- **Le parallélisme:**

Le réseau de neurones permet le traitement en parallèle des problèmes, il permet aussi une rapidité de calcul du faite de l'interconnexion de plusieurs neurones.

- **Capacité d'application:**

C'est la capacité d'apprentissage qui permet au réseau de prendre en considération des nouvelles données.

- La capacité d'apprentissage d'approximer n'importe quelle fonction.

- Les réseaux de neurones sont multivariables. Il sont applicables aux systèmes MIMO .

## 8- LES DIFFERENTS MODELES DES RESEAUX [11] :

En 1958 , Frank Rosenblatt a décrit le premier modèle opérationnel de réseau de neurones: le Perceptron . C'est un réseau mono couche inspiré du système visuel , utilisé pour le calcul des fonctions logiques en modifiant ces connexions synaptiques. Dans ce cas , la couche d'entrée c'est la rétine, la seconde couche est l'air d'association enfin la couche de sortie c'est l'air de réponse( voir fig1-7) .

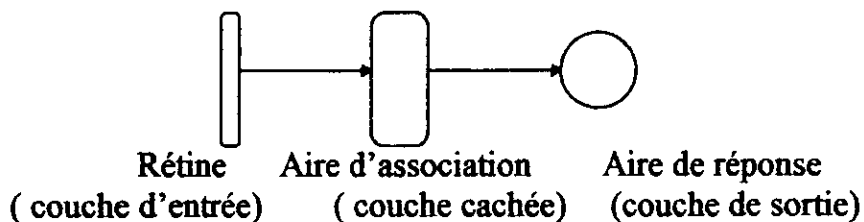


fig 1-7 Le Perceptron de F.Rpsenblatt

L'un des modèles le plus connu de F-Rosenblatt avait pour tâche de reconnaître les lettres de l'alphabet, que l'on présentait à la caméra .

En suite ,le perceptron a été utilisé par Widrow pour construire la machine Adaline en se basant sur l'interconnexion de plusieurs éléments.

Il existe différents types d'architecture pour les réseaux de neurones suivant la connectivité des neurones . Elle peut être totale ( c'est-à-dire tous les neurones sont reliés entre eux )ou locale( c'est-à-dire que les neurones sont reliés à leurs plus proches voisins) .

### 9-L'ARCHITECTURE D'UN RESEAU DE NEURONES :

L'architecture d'un réseau joue un rôle important dans son comportement . Elle spécifie le nombre de couches utilisées .

On distingue deux types de réseaux :

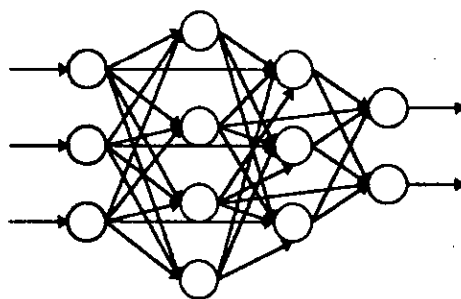
- Les réseaux statiques ( ou les réseaux à couches).
- Les réseaux dynamiques ( ou les réseaux entièrement connectés).

#### 9-1 Les réseaux statiques [6] :

Un réseau statique contient une couche d'entrée , une couche de sortie et en plus des couches cachées ( fig 1-8). Chaque fois qu'on lui présente la même entrée il donne la même sortie .

Dans ce type de réseau , un neurone dans une couche reçoit ses entrées des neurones situés plus en amont , dans le sens entrée -sortie, mais pas spécialement la couche immédiatement précédente. Les neurones qui appartient à une même couche ne sont pas connectés entre eux .

Les réseaux à couches sont utilisés pour la modélisation , le contrôle et l'identification des systèmes .



couche    couches    couche  
d'entrée    cachées    de sortie

fig 1-8 Architecture d'un réseau statique .

#### 9-2 Les réseaux dynamiques ( réseaux récurrent ) [8],[12] :

Dans leurs architecture , il y'a des bouclages .Chaque neurone est relié à tous les autres neurones , et il a même un retour sur lui même . L'évolution de ces réseaux est dépendante du temps .Ce type de réseau est dit entièrement connecté .

Il est régit par l'équation suivante :

$$\begin{cases} \dot{x}(t) = F(x(t), u(t), \theta) & (1-14) \\ y(t) = G(x(t), \theta) & (1-15) \end{cases}$$

où :

- x : c'est l'état .
- u : les entrées externes .
- $\theta$  : vecteur de paramètre du réseau .
- F : fonction qui représente la structure du réseau .
- G : fonction qui représente la relation entre les états et la sortie .

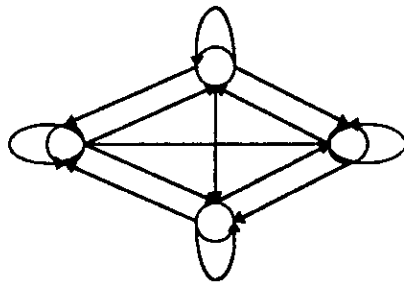


fig 1-9 Architecture d'un réseau dynamique.

Un réseau dynamique peut donner une sortie différente en lui présentant la même entrée à des instants différents. Ceci dépend des entrées qui étaient présentées précédemment .

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

**9-2-1 Réseau de Hopfield [22],[6] :**

C'est un réseau dynamique. Il a été l'objet de plusieurs travaux fondamentaux .Ce pendant , dans la pratique , il n'a pas suscité beaucoup d'intérêt .

Dans le modèle d'origine de Hopfield , un neurone est décrit par une variable  $\sigma$  qui peut prendre deux états, +1 ou - 1 . Donc l'état d'un réseau de N neurones est décrit par un vecteur binaire  $\sigma \in \{-1, +1\}^N$  .

Il y'a  $N^2$  connexions portant des poids. Les interconnexions entre les neurones sont symétriques .

$$W_{ij} = W_{ji}$$

La matrice des poids est donnée par la relation suivante :

$$W = \sum_{i=1}^n x_i x_i^T \tag{1-16}$$

L'inconvénient dans les réseaux de Hopfield est l'oubli , le réseau peut apprendre k exemples sans problèmes , et il les oublis tous dès qu'on souhaite lui apprendre un de plus .

## 10-L'APPRENTISSAGE DES RESEAUX DE NEURONES :

Le choix des poids synaptiques correct est un problème difficile à cause de leurs nombres et de leurs rapport avec le comportement globale du réseau . C'est pourquoi les réseaux de neurones disposent d'une règle d'apprentissage qui leur permet d'adapter leurs poids automatiquement . On commence par donner la procédure d'entraînement d'un réseau , puis nous présentons les trois différents types d'apprentissage :

- Apprentissage non supervisé .
- Apprentissage supervisé .
- Apprentissage semi-supervisé « reinforcement learning » .

### 10-1 La procédure d'apprentissage [11],[22] :

Les informations à traiter sont codées sous forme d'un vecteur appelé exemple d'entrée .

En général , l'apprentissage se fait sur une période relativement longue . Il comprend 4 étapes de calcul :

- **1<sup>er</sup> étape :**

Initialisation des poids synaptiques du réseau , le choix exact de ces valeurs n'est pas critique , et la pratique courante est d'initialiser les poids du réseau à des petites valeurs aléatoires au début de l'apprentissage .

- **2<sup>eme</sup> étape :**

Présentation des exemples d'entrées et propagation d'activation .

- **3<sup>eme</sup> étape :**

Calcul de l'erreur : pour chaque neurone de réseau une valeur d'erreur est calculée à partir de son activation et de celui des neurones qui lui sont reliés .

- **4<sup>eme</sup> étape :**

Calcul du vecteur de correction : à partir des valeurs d'erreur , on détermine la correction à apporter aux poids synaptiques .

Cette correction peut se faire après chaque présentation d'exemples .

### 10-2 Les techniques d'apprentissage :

#### 10-2-1 La technique (on-line) :

L'idéal est que l'apprentissage d'un réseau développe lui même un comportement adéquat en le plaçant en situation . Cet apprentissage est appelé interactif « on-line » . Cette technique est relativement peu employée , car l'apprentissage n'est pas encore une technique suffisamment maîtrisée pour pouvoir se dispenser d'intervention humaine[11] .

#### 10-2-2 La Technique (off-Line) :

C'est la technique la plus utilisée, dans ce cas l'apprentissage est effectué à part sur un échantillon d'exemples présélectionné.

---



Durant l'apprentissage, ces exemples seront chacun présenté plusieurs fois au réseau . Une présentation de tous les exemples s'appelle cycle d'apprentissage.

Le critère employé lors de la sélection d'un exemple est alors soit déterministe (l'ordre de présentation est fixe ), soit stochastique ( il est aléatoire ). Lors de cet apprentissage, il y'a le problème du choix des exemples , il faut qu'ils soient représentatifs de processus réel [11].

## 11-DIFFIRENTS TYPES D'APPRENTISSAGE :

### 11-1 Apprentissage Non Supervisé [11] :

Il modifie les poids du réseau en fonction d'un critère interne, indépendant de l'équation entre le comportement du réseau et la tâche qu'il doit effectuer.

la 1<sup>re</sup> règle d'apprentissage fut non supervisée vers les années 40 « Hebb ».

L'hypothèse ici est qu'une synapse améliore son efficacité seulement quand l'activité de ses deux neurones est corrélée, mathématiquement parlant :

$$W_{ij} = \eta \cdot \overline{A_i A_j} \quad (1-17)$$

$\eta$  : constante  $0 < \eta \ll 1$ .

$W_{ij}$  : poids synaptiques entre les neurones i et j .

$A_i A_j$  : C'est la corrélation entre l'activation des neurones i et j du réseau .

Plusieurs modèles ont été proposé avec l'apprentissage Hebbien « Dennis et Wiles 1993 ; Hopfield, 1982 ; Judouim et Menuim , 1990 ».

Cependant, c'est un critère fort limité, car il ne tient pas compte de la nature de la tâche que le réseau doit effectuer.

D'autre auteurs ont étudié un apprentissage anti-hebbien, qui décorrèle l'entrée et la réponse du réseau « Barleau , 1989; Foldiak , 1992 » .

$$W_{ij} = -\eta \cdot \overline{A_i A_j} \quad (1-18)$$

Ce type d'apprentissage est utilisé pour les processus rapide ou quand il y'a un manque d'information .

### 11-2 Apprentissage supervisé [8],[11] :

Il dispose d'un comportement de référence vers lequel il fait converger le réseau .

On suppose que pour chaque exemple d'entrée est associé une sortie désirée ; l'apprentissage se déroule de la manière suivante :

Les exemples d'entrées sont présentés aux neurones d'entrée , puis l'activation propagée à travers le réseau . La réponse des neurones de sorties est alors comparée aux valeurs désirées, ce qui donne l'erreur du réseau pour les exemples donnés.

Cette erreur est répartie a chaque poids du réseau selon le rôle qu'il a dans la production de la réponse erronée.

$$\Delta W = f_w(W, E, \eta) \tag{1-19}$$

$\Delta W$  : C'est la fonction qui calcule la modification à apporter aux poids  $W$  en fonction de l'erreur  $E$  et une constante  $\eta$  ( pas d'apprentissage ).

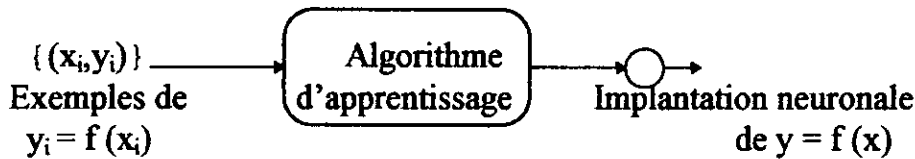


fig 1-10 Apprentissage à partir d'exemples .

Dans les calculs, ce type d'apprentissage est plus complexe que le précédent .

**11-3 Apprentissage semi-supervisé ( reinforcement learning ) [8],[21] :**

Dans ce cas l'information est implicite ( apprentissage par renforcement ), sous forme d'une simple appréciation « bon ou mauvais », sans mesure d'erreur [11].

Ce type d'apprentissage est basé sur l'adaptation d'un réseau auxiliaire appelé réseau critique, la sortie de ce dernier va mesurer la qualité de la sortie du réseau .

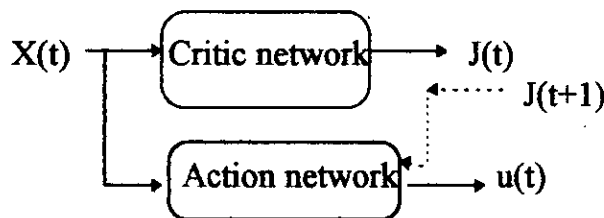


fig 1-11 Schéma de l'apprentissage semi-supervisé .

$J(t)$  mesure la qualité de la sortie  $u(t)$  par rapport à la sortie désirée .

• **Les inconvénients de cette méthode [8] :**

L'ajustement des poids du réseau critique en fonction de  $J(t + 1)$ , et du réseau .

Werbos, dans son article a montré qu'on peut le faire avec la programmation dynamique stochastique ( H DP). En général les règles d'apprentissage supervisé sont des formes de descente de gradient .

**12-1- LA TECHNIQUE DU DESCENTE DE GRADIENT [11] :**

C'est une technique d'optimisation issue de la recherche opérationnelle , son but est de trouver par itération les valeurs d'ensemble de paramètres (dans notre cas un choix de cas synoptiques ), et elle donne la possibilité d'évaluer le coût ( l'erreur  $E$  ).

Le principe d'une optimisation par descente de gradient est de progresser de solution en solution en minimisant l'erreur à chaque pas.

- **Inconvénient de la technique:**

Les surfaces de coût peuvent comprendre des minima locaux.

Les recherches par descente de gradient sont itératives et généralement procèdent par une succession de modification discrètes ( par pas ) sur une surface continue, c'est le cas pour le réseau des neurones, on cherche un ensemble de poids synaptiques dans un domaine de valeurs continues . Alors se pose le problème de la taille de ces pas, un pas trop grand peut empêcher la découverte d'une solution optimale .Un trop petit augmente inutilement le temps de calcul [11].

### 12-2- OPTIMISATION ALEATOIRE MOA:

L'optimisation aléatoire de Matyas assure une convergence vers un minimum globale. Elle est utilisée quand la dimension du réseau est importante.

Les étapes de cette méthode sont les suivantes, où on va optimiser aléatoirement la fonction  $f(x)$ :

1- Initialiser le vecteur  $x$  aléatoirement à  $x(0)$ , fixer le nombre maximal l'itérations  $m$   $b(0)$ , donner une valeur à  $V$  mettre  $k=0$ .

2- Générer 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 :

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

$$f(x(k) + \varepsilon(k)) > f(x(k)) \text{ et } f(x(k) - \varepsilon(k)) < f(x(k))$$

alors :  $x(k+1) = x(k) - \varepsilon(k)$  et  $b(k+1) = b(k) - 0.4\varepsilon(k)$

sinon  $x(k+1) = x(k)$  et  $b(k+1) = 0.5b(k)$

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

### 13- APPRENTISSAGE DES RESEAUX DE NEURONES STATIQUES:

Le but de l'apprentissage supervisé est de minimiser l'erreur quadratique

$$e^2(t) = \frac{1}{2} (y_d - y)^2 \quad (1-20)$$

pour cela on peut utiliser deux approches :

#### 1- F.F.N. pattern [8],[21] :

Découvert en 1974 par Werbos, puis redécouvert en 1986 par Rumelhart , Hunton et Williams .

$$E_p = \frac{1}{2} \sum_j (y_{d_j} - y_j)^2 \quad (1-21)$$

$$E = \sum_p E_p$$

$E_p$  : l'erreur sur le  $p^{ème}$  exemples .

$E$  : l'erreur global sur tous les exemples .

Le but de cet approche est de minimiser  $E_p$  suivant les étapes ci-dessous:

1. Initialiser les poids .
2. Présenter l'entrée et la sortie désirée .
3. calculer la sortie du réseau .
4. Ajuster les poids de réseau.

**2- F.F.N. batch [8],[21] :**

Dans ce cas, l'erreur a minimisé est  $E = \sum_p E_p$  ( l'erreur globale ).

Pour cet approche on utilise la dérivée de Werbos, c'est une dérivée ordonnée.  
soit :

$$x_i = f(x_{i-1}, \dots, x_n), \quad i = 1, \dots, n+1 .$$

La dérivée ordonnée de  $x_{n+1}$  est :

$$\frac{\partial^+ x_{n+1}}{\partial x_i} = \sum_{j>i}^{n+1} \frac{\partial^+ x_{n+1}}{\partial x_j} \frac{\partial f_j}{\partial x_i} \tag{1-22}$$

(1-22) est une équation réursive .

Dans ce cas d'erreur est :

$$E = \frac{1}{2} \sum_{t=1}^m (\hat{y}(t) - y(t))^T (\hat{y}(t) - y(t)) \tag{1-23}$$

$y(t)$  : c'est la sortie désirée .

Les poids sont ajustés suivant la relation :

$$A_{ij}(k+1) = A_{ij}(k) - \eta_k \frac{\partial E}{\partial A_{ij}} \tag{1-24}$$

$k$  : le  $k$ -ieme passage sur tous les exemples .

avec :

$$\frac{\partial E}{\partial A_{ij}} = \frac{\partial E}{\partial y_i} \frac{\partial y_i}{\partial S_i} \frac{\partial S_i}{\partial A_{ij}} = \frac{\partial E}{\partial y_i} g'(S_i) y_j \tag{1-25}$$

$S_i$  : la somme pondérée de toutes les sorties des neurones situés dans la couche précédente.

L'algorithme de F.F.N. Batch est le suivant :

$$\begin{aligned} \frac{\partial^* E}{\partial y_i} &= \sum A_{ij} g'(S_i) \frac{\partial^* E}{\partial y_j} + \varepsilon_i \\ \varepsilon_i &= \frac{\partial E}{\partial y_i} \\ \frac{\partial E}{\partial A_{ij}} &= \sum g'(S_i) \frac{\partial^* E}{\partial y_i(t)} y_j(t) \\ A_{ij}(k+1) &= A_{ij}(k) - \eta_k \frac{\partial E}{\partial A_{ij}} \end{aligned} \quad (1-26)$$

La procédure est la même que dans le cas précédent .

En faisant une comparaison entre les deux méthodes ,on voit que ,approximativement FFN pattern implante FFN batch après un cycle de calcul (passage de tous les exemples d'entraînement),pour des petites valeurs du pas d'apprentissage la différence entre les deux approches converge vers zéro après T cycles[19],

### 13- 1 Rétro-propagation de l'erreur (back-propagation) [5],[13],[21] :

C'est l'algorithme le plus utilisé dans l'apprentissage supervisé, repose sur la méthode du " Descente du gradient " , il réajuste les poids en propageant l'erreur à travers les couches du réseau.

Cet algorithme est basé sur le calcul de la dérivée partielle d'une fonction multivariable , lorsque toutes les variables varient. Ce travail a été fait en 1974 par WERBOS de l'université de HARDWARD, dans sa thèse de doctorat, mais il est resté inconnu jusqu'à 1982 où PARKER l'a de nouveau étudié, et il l'a utilisé dans les réseaux de neurones.

Cet algorithme est appelé back - propagation ,il a été publié en Avril 1985 par MIT .

Au début de 1986, RUMELHRT , HINTON et WILLIAMS ont présentés cette méthode avec plus de précision . Applicable seulement pour les réseaux multicouches .

#### 13-1-1 Présentation de l'algorithme [5] :

Tout d'abord on initialise les poids d'une manière aléatoire et on suit les étapes suivantes:

- 1- Présenter le vecteur d'entraînement x .
- 2- Calculer les entrées des couches cachées :

$$Net_p^h = \sum_{i=1}^n W_{ij} X_{pi} + \theta_j$$

avec :

- p : le p<sup>ème</sup> exemple.
- $\theta_j$  : le biais .
- h : indice de la couche cachée

3 - Calculer les sorties des couches cachées :

$$I_{pj} = f_j(Net_{pj}^h)$$

f: fonction d'activation de la sortie.

4 - Calculer les entrées de la couche de sortie :

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

$Net^0$  : l'entrée de la couche de sortie

5 - Calculer les sorties du réseau :

$$O_{pk} = f(Net_{pk}^0).$$

6 - Calculer le terme d'erreurs pour la couche de sortie , et il existe 2 approches :

6-a- Calculer l'erreur commise sur un passage complet des exemples :

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

6-b- Calculer l'erreur commise sur un passage d'un seul exemple :

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

Le terme d'erreur associé à chaque neurone de la couche de sortie , pour les deux approches est respectivement:

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

$$\delta_{pk}^0 = (y_{pk} - O_{pk}) f'(Net_{pk}^0)$$

$f'$  : la dérivée de f.

7 - Modifier les poids de la couche de sortie :

$$W_{kj}^0 = W_{kj}^0 + \eta \cdot \delta_{pk}^0 I_{pj}$$

$\eta$ : le pas d'apprentissage

8 - Calculer le terme d'erreur pour chaque neurone des couches cachées :

$$\delta_{pj}^h = f'(Net_{pj}^h) \sum_{k=1}^m \delta_{pk}^0 W_{kj}$$

9 - Modifier les poids des couches cachées :

$$W_{ji}^h = W_{ji}^h + \eta \cdot \delta_j^h \cdot X_{ji}$$

10 - Calculer la sortie du réseau en utilisant les nouveaux poids ; comparer la sortie du réseau à la sortie désirée.

11 - Si on utilise la deuxième approche, on refait les étapes précédentes jusqu'à  $E \leq E_0$  avec  $E_0$  : Erreur acceptable pour chaque exemple .

Pratiquement , la première approche converge plus rapide que la seconde.

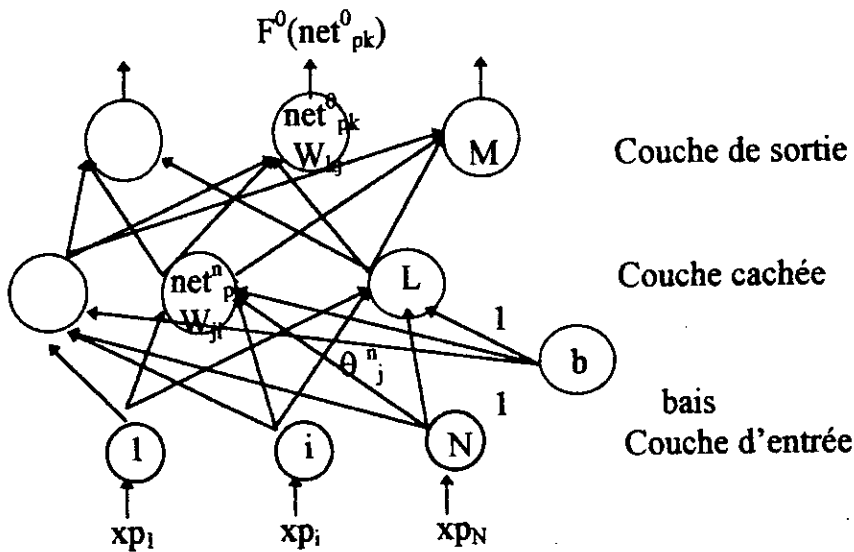


fig -1-12

**13-2 la technique du MOMENTUM [4] :**

On utilise un terme d'inertie dans la connexion des poids , dans lequel on tient compte de la connexion précédente. Son rôle est d'accélérer l'apprentissage et pour pouvoir sortir des minimums locaux de la surface d'erreur [5].

La relation utilisée pour la correction des poids est :

$$W_{ij}(t+1) = W_{ij}(t) + \Delta W_{ij}(t+1) \tag{1-27}$$

$\Delta W_{ij}(t+1)$  dépend de la couche sur laquelle porte les corrections avec la technique du momentum , il est donné par :

$$\Delta W_{ij}(t+1)_{\text{avec momentum}} = Y \Delta W_{ij}(t) + \Delta W_{ij}(t+1)_{\text{sans momentum}}$$

Y: C'est le momentum, il est variable.

on définit un rapport fixe appelé rapport d'erreur, noté C tel que :

- si l'erreur(t) > C l'erreur(t-1), alors Y=0
  - si l'erreur(t) ≤ l'erreur(t-1) alors Y = c<sup>te</sup> < 1
- en général Y est proche de 1

#### 14- APPRENTISSAGE DES RESEAUX DE NEURONES DYNAMIQUES:

Les réseaux de neurones qui ont la même architecture et qui sont entraînés avec différents algorithmes, n'évoluent pas de la même manière.

##### 14-1 Fixed point learning [12] :

C'est la 1<sup>ère</sup> technique utilisée pour l'apprentissage d'un réseau de neurone dynamique, elle impose un point d'équilibre au réseau dynamique. Il existe deux approches :

##### 14-1-1 La minimisation de l'erreur instantanée:

Appliquée à un réseau de trois couches. La couche cachée est entièrement connectée [12].

$$\dot{X} = -X + W \cdot g(X) + b \cdot z(k) \quad (1-28)$$

$$Y(k) = h^T X^* \quad (1-29)$$

$X^*$  : point d'équilibre stable de l'équation (1-28).

$z(k)$  : échantillon d'entrée à l'instant k.

L'algorithme d'apprentissage est :

$$\begin{aligned} h_i(k+1) &= h_i(k) - \eta_1 \frac{\partial E(k)}{\partial h_i} \\ W_{ij}(k+1) &= W_{ij}(k) - \eta_2 \frac{\partial E(k)}{\partial W_{ij}} \\ b_{ij}(k+1) &= b_{ij}(k) - \eta_3 \frac{\partial E(k)}{\partial b_{ij}} \end{aligned} \quad (1-30)$$

$\eta_1, \eta_2$  et  $\eta_3$  sont les pas d'adaptation.

avec:

$$\begin{aligned} E(k) &= (y_d(k) - y(k))^2 = (y_d(k) - h^T X^*)^2 \\ h_i(k+1) &= h_i(k) - \eta_1 (y_d(k) - y(k)) X_i^* \quad i = 1, \dots, q \end{aligned}$$



$$W_{ij}(k+1) = W_{ij}(k) - \eta_2 h_i(k)(y_d(k) - y(k))g_i(X_i^*) \quad i, j = 1, \dots, q \quad i \neq j$$

$$b_j(k+1) = b_j(k) - \eta_3 h_i(k)(y_d(k) - y(k))z_j(k) \quad i = 1, \dots, q, \quad j = 1, \dots, p$$

Ce type de réseaux est utilisé pour l'identification et la commande des systèmes dynamiques. il est rapide en apprentissage.

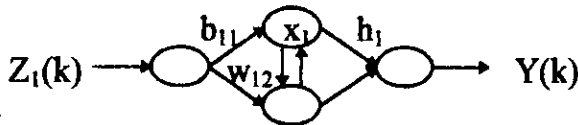


fig 1-13 Réseau récurrent

**14-1-2 Récurrent back-propagation [13] :**

Il minimise l'erreur globale :

$$E = \frac{1}{2} \sum_{t=1}^p (y_\infty^t - y^t)^T (y_\infty^t - y^t) \quad (1-31)$$

$y_\infty^k$  : les vecteur des points d'équilibre désirés .

Dans cet algorithme on calcule les points d'équilibre  $X_i^*$ , puis on calcule les points d'équilibre  $Z_i^*$  enfin les poids sont adaptés en utilisant l'équation :

$$\dot{A} = \alpha . g(s_i(\infty)) X_i^* Z_i^* \quad (1-32)$$

**14-2 Trajectory learning [21] :**

C'est la seconde technique d'apprentissage , la dynamique du réseau est imposée dans ce cas

$$E = \frac{1}{2} \sum_{\tau=0}^n [(d(\tau) - y(\tau))^T (d(\tau) - y(\tau))] \quad (1-33)$$

$d(\tau)$  : vecteur de la trajectoire désirée.

**15- PROBLEMES D'APPRENTISSAGE [11],[22] :**

• **Minima locaux :**

La majorité des méthodes d'apprentissage supervisé utilisent la descente de Gradient qui peut conduire à une solution sous optimale .Et en général, la forme de la surface d'erreur n'est pas connue , la solution dans ce cas est de choisir des poids initiaux différents pour éviter le minimum local ,ou bien changer la structure du réseau ( le nombre de neurones dans la couche).

- **Mauvais choix de paramètres :**

**Pas d'apprentissage :**

cette valeur affecte la taille du vecteur de correction . Théoriquement , le pas d'entraînement doit être très petit , mais malheureusement ceci entraîne un temps de calcul énorme , également un pas assez grand peut conduire à des oscillations compromettant ainsi la convergence.

L'utilisation d'un coefficient d'apprentissage adaptatif est un compromis de la rapidité de convergence et la stabilité du réseau [4].

Si l'erreur à l'instant  $t$  dépasse celle de l'instant  $(t-1)$  d'un certain rapport fixe appelé rapport d'erreur noté  $C$ , alors le facteur  $\eta$  est atténué par multiplication par une constante inférieure à 1 . dans le cas contraire , le pas  $\eta$  est augmenté par multiplication par un facteur supérieur à 1.

- si l'erreur( $t$ )  $> C$  l'erreur( $t-1$ ) , alors  $\eta = \eta^* (\eta\_déc)$  , avec ( $\eta\_déc < 1$ ).

- si l'erreur( $t$ )  $\leq$  l'erreur( $t-1$ ) , alors  $\eta = \eta^* (\eta\_inc)$  , avec ( $\eta\_inc > 1$ ).

Avec cette procédure , on accélère le temps d'apprentissage du réseau.

**16 - APPRENTISSAGE ET GENERALISATION [22] :**

Après que la phase d'apprentissage est terminée , nous souhaitons utiliser le réseau dans une véritable tâche . Pour cela , il faut être en mesure d'estimer d'abord le degré de confiance que l'on peut placer dans le réseau . Tout d'abord on effectue l'apprentissage à l'aide des exemples , puis on test l'aptitude du réseau à reconnaître des exemples non appris .

Le taux d'erreur mesuré en généralisation est nécessairement plus grand que l'erreur résiduelle de l'apprentissage . Il faut respecter ces conditions :

- Tous les exemples utilisés soient représentatifs des classes à reconnaître . De plus l'erreur de généralisation est directement liée au rapport entre le nombre d'exemples et le nombre total de connexions dans le réseau .

Il faut que le rapport du nombre d'exemples d'apprentissage au nombre de connexions soit supérieur à un . Plus ce rapport est élevé , plus l'erreur de généralisation est plus basse ! Si le nombre d'exemples est insuffisant , cela conduit à une généralisation très mauvaise .

- La taille appropriée de la couche cachée :

Si le nombre de cette couche est insuffisant , on peut pas atteindre une erreur d'apprentissage suffisamment faible . Et si la taille augmente , cette erreur peut être rendue de plus en plus petite au prix d'un apprentissage lourd . Au même temps , l'erreur de généralisation diminue passe par un minimum, puis recommence à croître . La figure (1-14) montre cet effet .

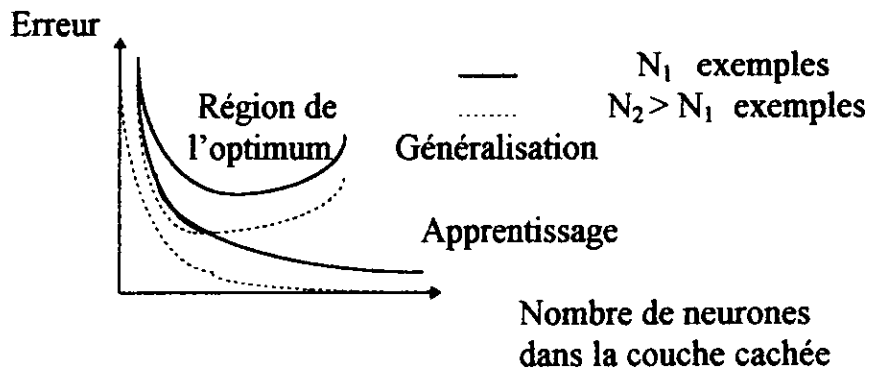


fig 1-14 Evolution de l'erreur d'apprentissage et de généralisation en fonction de la couche cachée et de nombre d'exemples.

## 17- REALISATIONS [22] :

La réalisation efficace de réseau de neurones a été d'abord une préoccupation dans les laboratoires de recherche, et plusieurs machines et circuits intégrés spécialisés ont vu le jour.

### 1-Logiciels :

Certains sont d'origine universitaire et diffusés dans le domaine public. D'autres sont proposés par des sociétés commerciales.

Il y a aussi des ensembles ordinateur-logiciel dotés de langages de description particuliers qui peuvent servir à la résolution de problèmes réels.

### 2-Simulateurs : logiciel et matériels associés :

Plusieurs fabricants ont créé des cartes accélératrices spécialisées dans les calculs d'apprentissage comme la rétropropagation, qui se place dans un micro-ordinateur de type PC.

### 3- Machines câblées spécialisées :

Ce sont des montages qui utilisent des composants du commerce, mémoires, LCA (Logical Cellular Arrays), FPGA (Field Programmable Gate Array)...pour leur construction.

### 4- circuits intégrés :

Dans les circuits intégrés, l'apprentissage n'est généralement pas implanté directement, mais a lieu sur ordinateur extérieur. Après les calculs, on charge les coefficients synaptiques.

**5-Implémentation analogique des réseaux de neurones [4] :**

Elle se fait à l'aide des circuits électroniques de base tel que :

**L'amplificateur opérationnel :** il fait la somme des courants « en parallèle » sur son noeud d'entrée . Sa tension de sortie est la somme des tensions d'entrées pondérées par des coefficients.

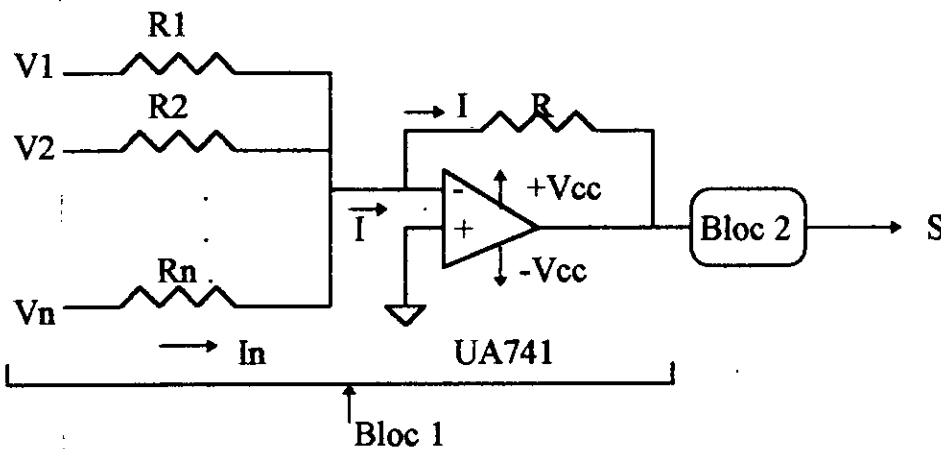


fig 1-15 Neurone artificiel réalisé à partir des amplificateurs opérationnels

Le bloc 2 pour le seuillage du neurone .

$$i_d = 0 \text{ implique } v_s = \sum_{k=1}^n \frac{-R.V_k}{R_k}$$

$(-V_k)$  : les entrées du réseau.

$\left(\frac{R}{R_k}\right)_{k=1,2,...,n}$  : les poids synaptiques.

La tension de transition V est constante , elle est égale à la valeur du biais associe au neurone .

**18- APPLICATIONS INDUSTRIELLES [22] :**

Plusieurs réalisations connues sont plus souvent dans un état de prototype qu'en exploitation véritable.

**1- Reconnaissance de forme statiques :**

Statique c'est-à-dire indépendante du temps . Le traitement d'images fait partie aussi de la reconnaissance de formes .

Des applications dans le domaine civil tel que la reconnaissance des visages , ou dans le domaine militaire, la reconnaissance de cibles .

L'une des applications les plus étudiées est la reconnaissance de signes manuscrits, spécialement les chiffres pour l'utiliser dans la reconnaissance de codes postaux. Les laboratoires AT et T de Holmdel ont mis au point un système efficace, dont le cœur est un réseau à cinq couches, précédé d'un système de prétraitement et de normalisation. Une partie de ce réseau est constituée par un circuit intégré qui contient quatre des cinq couches et 130.000 connexions. Les performances atteignent 1000 caractères par seconde pour la puce seule.

### **2-Détection d'anomalie :**

Ce réseau est capable d'indiquer un état anormal, car il a appris le fonctionnement normal du système. On l'applique pour la surveillance de disjoncteurs à très haute tension, aussi dans les machines qui distribuent les billets de banque.

### **3- la classification :**

Dans ce type de problème, les entrées sont l'information recueillie par un ou plusieurs capteurs (caméra, micro,...), le but ici est d'associer à l'objet perçu une forme connue. [Gallinari et al 1991, Hinton 1989, Kohonen].

### **4-transformation :**

Il existe des réseaux qui doivent apprendre une transformation entre l'entrée et la sortie par exemple en traitement d'images, pour calculer les contours d'objets [Larcelle, 1989], la compression des données [Nauphol et al, 1991]

### **5-Traitement dépendant du temps**

#### **• prédiction :**

Dans ce cas, on utilise les réseaux récurrents. Beaucoup de travaux sont consacrés à la prédiction de données financières et boursières.

- Identification et commande de processus.
  - Application militaires.
  - Application bancaires et financières, (pour la détection de fraude sur les cartes de crédit).
  - Traitement de la parole.
  - Robotique.
-

## CONCLUSION :

Dans ce chapitre , nous avons commencé par donner des définitions concernant le réseau de neurones , ce dernier n'est rien d'autre qu'un approximateur de fonctions , formé de l'interconnexion de plusieurs noeud appelé neurone  
les réseaux de neurones peuvent être classer en deux grandes catégories :

- réseaux statiques
- réseaux dynamique

Les procédures d'apprentissage , peuvent se subdiviser, elles aussi en trois classes :

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

Nous avons également parlé du fameux algorithme de back-propagation dont le principe et la procédure d'utilisation ont été présenté . Cet algorithme sera utilisé dans la suite de notre travail .

Ensuite, nous avons cité les différents problèmes rencontrés lors de l'étude des réseaux de neurones, ainsi que les solutions proposées .

On a terminé ce chapitre par présenter les domaines d'application des réseaux de neurones .

Le développement de ces dernier a permet aux chercheurs de passer de la simulation numérique à leurs implémentation sous forme de logiciel ou circuits intégré ...afin de mieux profiter de leurs avantages.

---

# CHAPITRE . II

## INTRODUCTION :

La simplicité de commander le moteur à courant continu a permis à ce dernier d'être le meilleur variateur de vitesse. Mais il nécessite une maintenance souvent trop coûteuse pour les performances désirées, à cause de la présence du collecteur mécanique.

Alors on a pensé à utiliser des machines à courant alternatif telle que la machine asynchrone. C'est une machine dont la vitesse en charge et la fréquence du réseau auquel elle est reliée ne sont pas dans un rapport constant (phénomène du glissement). Le moteur asynchrone présente plusieurs avantages tel que :

- Faible coût.
- Entretien réduit.
- Facilité de construction.
- Grande plage de puissance.
- Robustesse.

Néanmoins, on trouve des difficultés pour le commander qui réside dans le fait que :

- Le modèle dynamique du système est non linéaire.
- Les deux composantes du flux rotorique sont deux variables d'état non observables.
- La résistance rotorique varie considérablement quand la machine s'échauffe, ce qui affecte la dynamique du système.

### 1- MODELISATION DE LA MACHINE ASYNCHRONE [2],[3] :

La machine étudiée dans ce travail, est une machine qui comporte un stator et un rotor dont les armatures magnétique sont toutes les deux cylindriques, séparées par un entrefer constant, et munies chacune d'un enroulement triphasé.

Vu la grande complexité qui présente le moteur asynchrone, son étude est facilitée par les hypothèses simplificatrices suivantes :

- Le circuit magnétique est non saturé et parfaitement feuilleté au stator et au rotor.
- On prend en considération que le premier harmonique d'espace de la distribution de la force magnétique créée par chaque phase du stator et rotor.

#### 1-1 Mise en équations :

La machine est représentée par les trois phases statorique a b c, et par les trois phases rotoriques A B C. L'angle  $\theta$  définit la position relative entre une phase statorique et une autre rotorique. Voir fig 2-1.

Tenant compte des hypothèses précédentes, on obtient les équations sous forme matricielle.



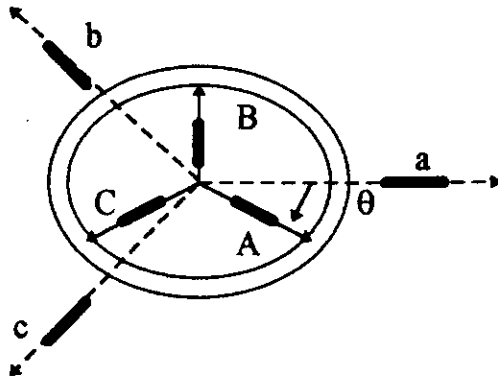


fig 2-1 Représentation d'une machine asynchrone triphasée.

$$\begin{aligned}
 [u_{abc}] &= Rs[iabc] + \left[ \frac{d\Psi_{abc}}{dt} \right] \\
 [0] &= Rr[iABC] + \left[ \frac{d\Psi_{ABC}}{dt} \right] \\
 [\Psi_{abc}] &= [Ls][iabc] + [Lsr][iABC] \\
 [\Psi_{ABC}] &= [Lr][iABC] + [Lsr][iabc]
 \end{aligned}
 \tag{2-1}$$

$$Lsr(\theta) = L_{aA} \cos \theta_r$$

Avec :

- Rs : Résistance d'une phase statorique.
- Rr : Résistance d'une phase rotorique.

$$[u_{abc}] = \begin{bmatrix} u_a \\ u_b \\ u_c \end{bmatrix}, \quad [iabc] = \begin{bmatrix} ia \\ ib \\ ic \end{bmatrix}, \quad [\Psi_{abc}] = \begin{bmatrix} \Psi_a \\ \Psi_b \\ \Psi_c \end{bmatrix}$$

[Ls],[Lr] : sont les matrices d'inductances statoriques et rotoriques .

[Lsr] : la matrice des inductances mutuelles.

$L_{aA}$  : l'inductance mutuelle entre phases statoriques et rotoriques.

$$[Ls] = \begin{bmatrix} L_{aa} & L_{ab} & L_{ab} \\ L_{ab} & L_{ab} & L_{ab} \\ L_{ab} & L_{ab} & L_{ab} \end{bmatrix}$$

$$[L_{sr}] = L_{sd} \begin{bmatrix} \cos\theta & \cos(\theta + \frac{2\pi}{3}) & \cos(\theta - \frac{2\pi}{3}) \\ \cos(\theta - \frac{2\pi}{3}) & \cos\theta & \cos(\theta + \frac{2\pi}{3}) \\ \cos(\theta + \frac{2\pi}{3}) & \cos(\theta - \frac{2\pi}{3}) & \cos\theta \end{bmatrix}$$

$\theta$  : l'angle électrique entre l'axe de référence du stator et du rotor.

$$[u_{abc}] = R_s[i_{abc}] + [L_s] \frac{d[i_{abc}]}{dt} + \frac{d[L_{sr}][i_{ABC}]}{dt} \quad (2-2)$$

$$[0] = R_r[i_{ABC}] + [L_r] \frac{d[i_{ABC}]}{dt} + \frac{d[L_{sr}][i_{abc}]}{dt}$$

Le couple électromagnétique :

$$C_m = p[i_{abc}]^T \frac{d[L_s][i_{ABC}]}{d\theta} \quad (2-3)$$

la matrice des inductances mutuelles est en fonction de  $\theta$ , mais cette dernière est variable, ce qui rend la résolution du système précédent impossible.

A ce niveau, on remarque que l'étude du système tel qu'il est réellement pose beaucoup de problèmes du fait qu'on a un système d'équations à coefficients variable. D'où la nécessité de trouver un modèle mathématique équivalent au modèle réel, où les coefficients seraient constants. Alors on est obligé de recourir à certaines transformations usuelles. Dans notre travail on utilisera la transformation de PARK.

### 1 -2 Transformation De PARK :[3]

Elle permet le passage d'un système triphasé réel (a, b, c) à un autre système orthogonale (d, q, o) qui lui est équivalent.

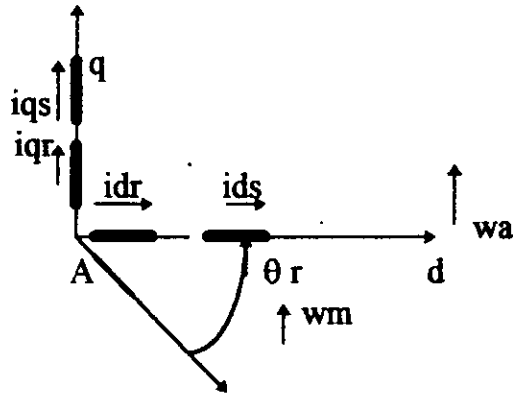
Elle consiste à transformer les enroulements statoriques et rotoriques en enroulements orthogonaux équivalents Voir fig 2-2 .

La transformation de PARK permet la conservation des puissances et des valeurs efficaces .

Le passage du système (a, b, c) au système (d, q) se fera donc par l'intermédiaire de la matrice de PARK généralisée :

$$[A] = \frac{2}{3} \begin{bmatrix} \cos\theta & \cos(\theta - \frac{2\pi}{3}) & \cos(\theta + \frac{2\pi}{3}) \\ -\sin\theta & -\sin(\theta - \frac{2\pi}{3}) & -\sin(\theta + \frac{2\pi}{3}) \\ \frac{1}{2} & \frac{1}{2} & \frac{1}{2} \end{bmatrix} \quad (2-4)$$

fig 2- 2  
Transformation de PARK  
appliquée à la MAS



La transformation des grandeurs statoriques est définie par :

$$\begin{aligned}
 [idq]_s &= [As][iabc] \\
 [u_{dq}]_s &= [As][u_{abc}] \\
 [\Psi_{dq}]_s &= [As][\Psi_{abc}]
 \end{aligned}$$

La même transformation s'obtient pour les grandeurs rotoriques en remplaçant les indices s par r et a ,b,c par A ,B et C.

On note  $wa$  : vitesse du repère de PARK.

L'application de PARK aux courants,aux flux et aux tensions donne les équations suivantes :

$$\begin{aligned}
 v_{qs} &= R_s \cdot i_{qs} + \frac{d\Psi_{qs}}{dt} + wa\Psi_{ds} \\
 v_{ds} &= R_s \cdot i_{ds} + \frac{d\Psi_{ds}}{dt} + wa\Psi_{qs} \\
 v_{qr} &= R_r \cdot i_{qr} + \frac{d\Psi_{qr}}{dt} + (wa - wr)\Psi_{dr} \\
 v_{dr} &= R_r \cdot i_{dr} + \frac{d\Psi_{dr}}{dt} - (wa - wr)\Psi_{qr}
 \end{aligned} \tag{2-5}$$

$$\begin{aligned}
 \Psi_{qs} &= L_s \cdot i_{qs} + L_{sr} \cdot i_{qr} \\
 \Psi_{ds} &= L_s \cdot i_{ds} + L_{sr} \cdot i_{dr} \\
 \Psi_{qr} &= L_r \cdot i_{qr} + L_{sr} \cdot i_{qs} \\
 \Psi_{dr} &= L_r \cdot i_{dr} + L_{sr} \cdot i_{ds}
 \end{aligned} \tag{2-6}$$

L'équation du couple électromagnétique est donnée par :

$$C_{em} = p \cdot (\Psi_{ds} \cdot i_{qs} - \Psi_{qs} \cdot i_{ds}) \tag{2-7}$$

avec :

$$\begin{aligned} L_s &= L_{aa} - L_{ab} \\ L_r &= L_{AA} - L_{AB} \\ L_{sr} &= \frac{3}{2} L_{Aa} \end{aligned}$$

### 1-3 Mise sous forme d'état :

On prend comme vecteur d'état le vecteur :  $X = [ \text{ids} \quad \text{iqs} \quad \Psi_{dr} \quad \Psi_{qr} \quad \text{wr} ]^T$

et le vecteur (vds, vqs) comme vecteur de commande.

en posant :

$$X = [ x_1 \quad x_2 \quad x_3 \quad x_4 \quad x_5 ]^T, \quad Tr = \frac{L_r}{R_r}, \quad Ts = \frac{L_s}{R_s}, \quad \text{et } \sigma = \frac{1 - L_{sr}^2}{L_s L_r}, \quad M = L_{sr}$$

Dans un repère arbitraire, alimentée en tension, la machine se modélise comme suit :

$$\begin{cases} \dot{x}_1 = -\left(\frac{1}{\sigma Ts} + \frac{1-\sigma}{\sigma Tr}\right)x_1 + w_a \cdot x_2 + \frac{1-\sigma}{\sigma M Tr} x_3 + \frac{1-\sigma}{\sigma M} x_4 x_5 + \frac{1}{\sigma L_s} v_{ds} \\ \dot{x}_2 = -\left(\frac{1}{\sigma Ts} + \frac{1-\sigma}{\sigma Tr}\right)x_2 - w_a \cdot x_1 + \frac{1-\sigma}{\sigma M Tr} x_4 - \frac{1-\sigma}{\sigma M} x_3 x_5 + \frac{1}{\sigma L_s} v_{qs} \\ \dot{x}_3 = \frac{M}{Tr} x_1 - \frac{1}{Tr} x_3 + (w_a - x_5)x_4 \\ \dot{x}_4 = \frac{M}{Tr} x_2 - \frac{1}{Tr} x_4 - (w_a - x_5)x_3 \\ \dot{x}_5 = \frac{p^2 M}{J L_r} (x_3 x_2 - x_4 x_1) - \frac{K_f}{J} x_5 - \frac{p}{J} C_r \end{cases} \quad (2-7)$$

### 1-4 Choix du référentiel:

#### 1-4-1 Dans un repère synchrone :

Dans ce cas,  $w_a = w_s$ , et le modèle devient :

$$\begin{cases} \dot{x}_1 = -\left(\frac{1}{\sigma Ts} + \frac{1-\sigma}{\sigma Tr}\right)x_1 + w_s \cdot x_2 + \frac{1-\sigma}{\sigma M Tr} x_3 + \frac{1-\sigma}{\sigma M} x_4 x_5 + \frac{1}{\sigma L_s} v_{ds} \\ \dot{x}_2 = -\left(\frac{1}{\sigma Ts} + \frac{1-\sigma}{\sigma Tr}\right)x_2 - w_s \cdot x_1 + \frac{1-\sigma}{\sigma M Tr} x_4 - \frac{1-\sigma}{\sigma M} x_3 x_5 + \frac{1}{\sigma L_s} v_{qs} \\ \dot{x}_3 = \frac{M}{Tr} x_1 - \frac{1}{Tr} x_3 + (w_s - x_5)x_4 \\ \dot{x}_4 = \frac{M}{Tr} x_2 - \frac{1}{Tr} x_4 - (w_s - x_5)x_3 \\ \dot{x}_5 = \frac{p^2 M}{J L_r} (x_3 x_2 - x_4 x_1) - \frac{K_f}{J} x_5 - \frac{p}{J} C_r \end{cases} \quad (2-8)$$

**1-4-2 Dans un repère du stator :**

Dans un référentiel lié au stator on a :  $\omega_a = 0$ , on obtient les équations suivantes :

$$\begin{cases} \dot{x}_1 = -\left(\frac{1}{\sigma T_s} + \frac{1-\sigma}{\sigma T_r}\right)x_1 + \frac{1-\sigma}{\sigma M T_r}x_3 + \frac{1-\sigma}{\sigma M}x_4x_5 + \frac{1}{\sigma L_s}v_{ds} \\ \dot{x}_2 = -\left(\frac{1}{\sigma T_s} + \frac{1-\sigma}{\sigma T_r}\right)x_2 + \frac{1-\sigma}{\sigma M T_r}x_4 - \frac{1-\sigma}{\sigma M}x_3x_5 + \frac{1}{\sigma L_s}v_{qs} \\ \dot{x}_3 = \frac{M}{T_r}x_1 - \frac{1}{T_r}x_3 - x_5x_4 \\ \dot{x}_4 = \frac{M}{T_r}x_2 - \frac{1}{T_r}x_4 + x_5x_3 \\ \dot{x}_5 = \frac{p^2 M}{J L_r}(x_3x_2 - x_4x_1) - \frac{K_f}{J}x_5 - \frac{p}{J}C_r \end{cases} \quad (2-9)$$

**1-5 Théorie du flux orienté [1],[16]:**

La commande de la machine asynchrone est difficile du fait qu'il existe un couplage complexe entre ses variables tels que flux, couple.

En 1971 BLASHKE a proposé une nouvelle théorie de commande par flux orienté qui permet d'assimiler la machine asynchrone à une machine à courant continu.

**1-5-1 Principe du flux orienté :**

Dans l'expression du couple électromagnétique, on voit bien qu'il y'a une interaction entre les flux et les courants statoriques et rotoriques.

$$C_{em} = p.(\Psi_{\sigma} \cdot i_{qs} - \Psi_{\sigma r} \cdot i_{ds}) \quad (2-10)$$

Le flux peut être décomposé dans un référentiel orthogonal (d,q), une composante directe selon l'axe d , et une composante transversale selon l'axe q.

L'orientation du flux consiste à maintenir la composante directe constante et à annuler la composante en quadrature. Ceci est possible si on choisit un référentiel lié au champ tournant , afin que l'axe d coïncide avec la direction désirée du flux.

On veut que le flux  $\Psi_r$  soit commandé par le courant  $i_{ds}$  , et le couple  $C_{em}$  par le courant  $i_{qs}$  voir fig 2-3.

$$\begin{aligned} \Psi_r &= \Psi_{\sigma} \\ \Psi_{\sigma r} &= 0 \\ C_{em} &= \frac{p^2 M}{L_r} \Psi_{\sigma} \cdot i_{qs} \\ \dot{x}_3 &= \frac{M}{T_r}x_1 - \frac{1}{T_r}x_3 \\ \omega_s &= x_5 + \frac{M}{T_r} \cdot \frac{x_2}{x_3} \end{aligned} \quad (2-11)$$

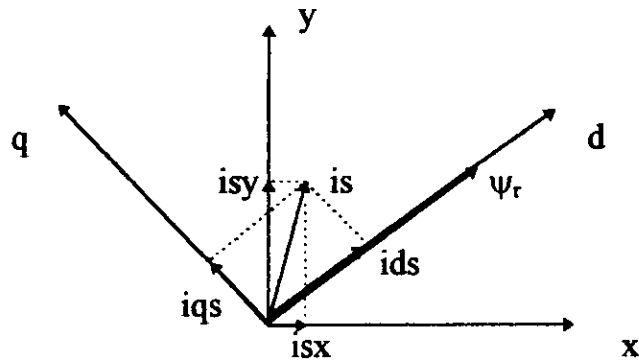


fig 2-3 Orientation du flux rotorique

D'après les équations précédentes ,on a l'expression de la pulsation statorique :

$$\omega_s = x_5 + \frac{M}{Tr} \frac{x_2}{x_3}$$

**1-5-2 Modèle de la machine asynchrone a flux orienté ,alimentée en tension:**

En remplaçant  $\omega_s$  par sa valeur on obtient le modèle dynamique suivant :

$$\begin{cases} \dot{x}_1 = -\left(\frac{1}{\sigma T_s} + \frac{1-\sigma}{\sigma T_r}\right)x_1 + \frac{1-\sigma}{\sigma M T_r}x_3 + x_5x_2 + \frac{Mx_2^2}{Trx_3} + \frac{1}{\sigma L_s}v_{ds} \\ \dot{x}_2 = -\left(\frac{1}{\sigma T_s} + \frac{1-\sigma}{\sigma T_r}\right)x_2 - \frac{1-\sigma}{\sigma T_r}x_3x_5 - x_5x_1 - \frac{Mx_1x_2}{Trx_3} + \frac{1}{\sigma L_s}v_{qs} \\ \dot{x}_3 = \frac{M}{Tr}x_1 - \frac{1}{Tr}x_3 \\ \dot{x}_5 = \frac{p^2 M}{JLr}x_3x_2 - \frac{Kf}{J}x_5 - \frac{p}{J}Cr \end{cases} \quad (2-12)$$

avec :  $X = [x_1, x_2, x_3, x_5]^T$  vecteur d'état

On remarque que le flux rotorique dépend seulement de la composante directe du courant statorique  $i_{ds}$  , et que le couple dépend de la composante en quadrature  $i_{qs}$  si le flux rotorique est maintenu constant.

Donc, on a décomposé le courant statorique en deux termes  $i_{ds}$  et  $i_{qs}$  pour commander le flux et le couple indépendamment l'un à l'autre.

**CONCLUSION :**

Après avoir modéliser la machine dans un repère biphasés en utilisant la transformée de PARK , on a obtenu une configuration du système qui est non linéaire multivariable et couplé .

L'orientation du flux rotorique est le meilleur choix pour avoir un découplage entre les différentes variables de la machine.

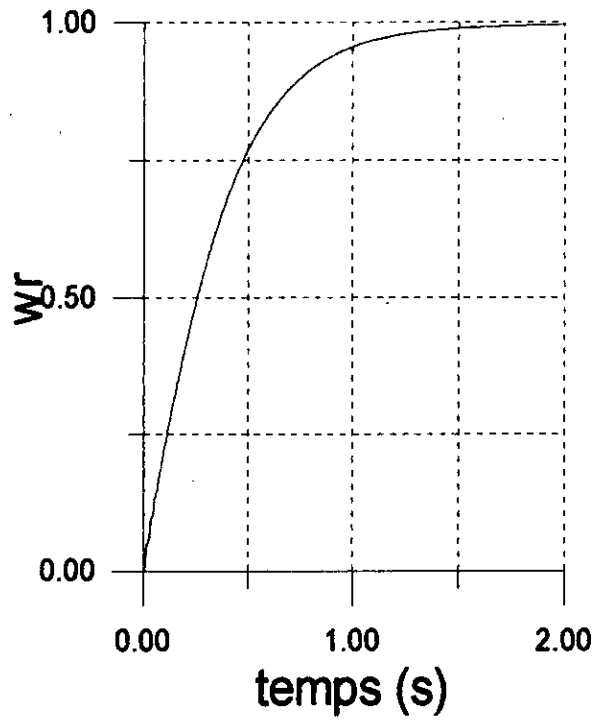
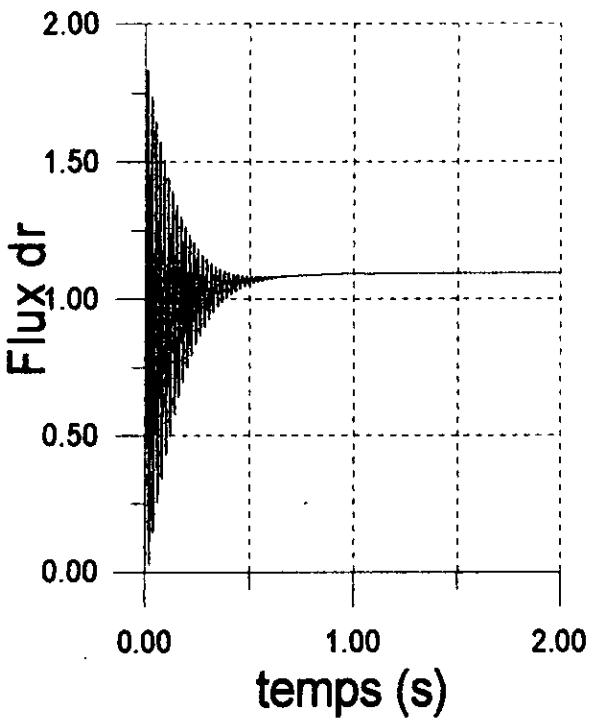
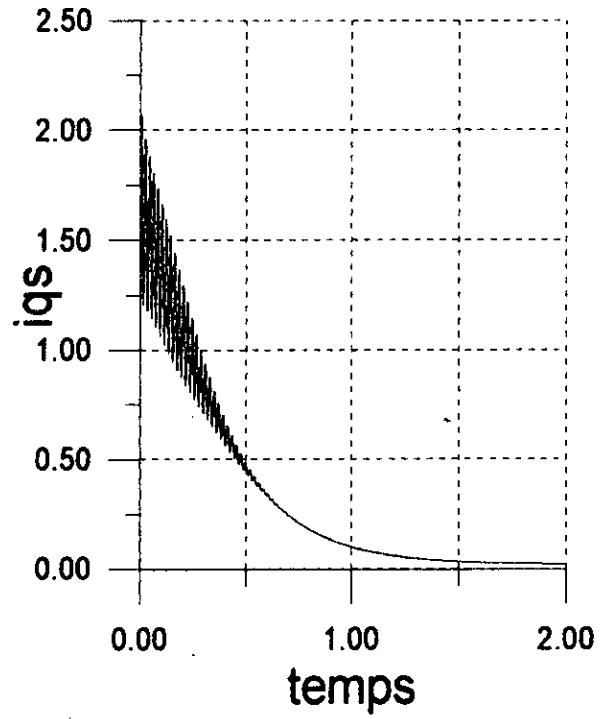
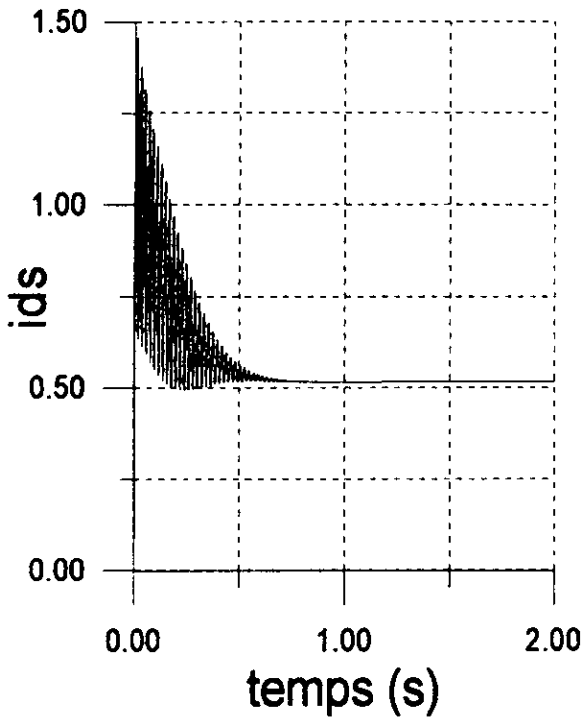


Planche-1- simulation en boucle ouverte de la MAS

## INTRODUCTION

**A** cause de leurs caractéristiques non-linéaire ,les réseaux de neurones sont largement utilisés dans la commande des systèmes. Plusieurs travaux ont montré la capacité des réseaux de neurones d'approximer différentes fonctions linéaires ou non linéaires ,en se servant d'une seule couche cachée dont chaque neurone a une fonction d'activation continue et non linéaire (sigmoïde)[12].

### 1-IDENTIFICATION INVERSE [6]:

Le réseau de neurone est entraîné de différentes méthodes ,pour apprendre l'inverse du modèle à identifier.

#### 1-1 La 1<sup>ère</sup> structure [10]:

C'est la méthode directe qui utilise l'apprentissage généralisé (fig 3-1)

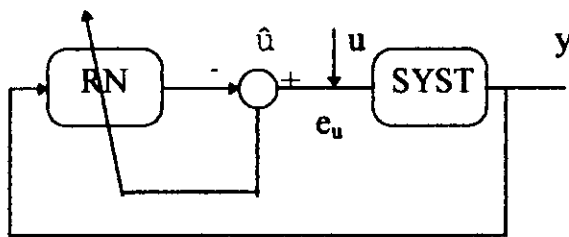


fig 3-1 Apprentissage généralisé

la sortie du système est injectée dans le réseau pour produire l'approximé de l'entrée. L'erreur de comparaison  $e_u$  entre l'entrée du système  $u$  et la sortie du réseau  $\hat{u}$  est utilisée pour l'entraînement de ce dernier.

Ainsi ,cette structure force le réseau de neurones à approximer l'inverse du système , néanmoins elle présente un inconvénient major.

Lors de la commande ,il est nécessaire que les signaux délivrés par le réseau de neurones puissent imposer au système de suivre les références exigées .Or, pendant la phase d'identification ,c'est le signal de sortie du système  $y$  qui a servi comme entrée pour le réseau de neurones et non la sortie désirée  $y_d$ .

L'utilisation d'un grand nombre d'exemples d'entraînement est indispensable dans ce cas, ceci est dû au concept de l'excitation persistante ,où les entrées doivent apporter toutes les informations concernant le système [10].



### 1-2 La 2<sup>ème</sup> structure [ 6 ],[14] :

Dans ce cas le réseau de neurones produira sa sortie à partir de la sortie désirée (référence).

L'apprentissage est dit spécialisé, l'erreur entre la sortie du système et la sortie désirée est rétropropagée à travers le système, puis elle est utilisée pour entraîner le réseau.

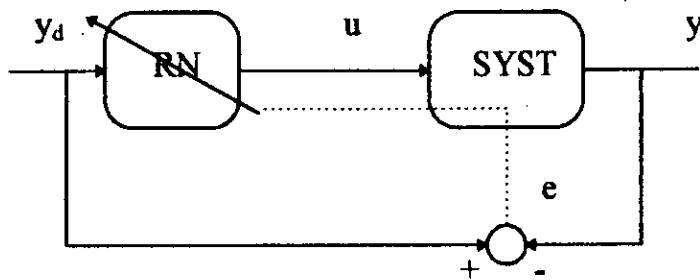


fig 3-2 L'apprentissage spécialisé

soit  $J$  le critère à minimiser tel que :

$$j = \frac{1}{2} (y_d - y)^2$$

$$\frac{\partial J}{\partial u} = (y_d - y) \frac{\partial y}{\partial u}$$

Le calcul de  $\partial y / \partial u$  se fait par plusieurs méthodes.

- La dérivée partielle est approximée par :

$$\frac{\partial y(k+1)}{\partial u(k)} = \frac{y(k+1) - y(k)}{u(k) - u(k-1)}$$

- Variation de l'entrée au tour d'un point de fonctionnement :

$$\frac{\partial y(k+1)}{\partial u(k)} = \frac{f(u + \delta \cdot u) - f(u)}{\delta \cdot u} \quad f : \text{la sortie}$$

- Utilisation d'un modèle neuronal directe :

un réseau est placé en parallèle pour approximer la dynamique directe, l'erreur  $(y_d - y)$  est rétropropagée à travers ce modèle, l'utilisation de l'algorithme de la rétropropagation de l'erreur facilite le calcul de  $\frac{\partial y_m}{\partial u}$ , avec  $y_m$  est la sortie du modèle directe (fig 3-3)

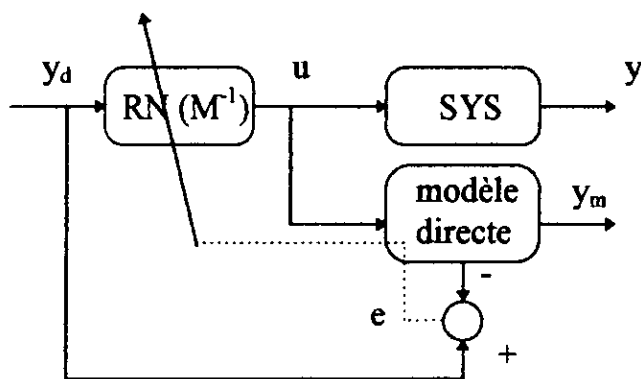


fig 3-3

L'utilisation de ce modèle directe apparaît efficace ,dans le cas où les sorties du système sont bruitées [10].

1-3 3<sup>ème</sup> Structure :l'apprentissage indirecte [8],[14] :

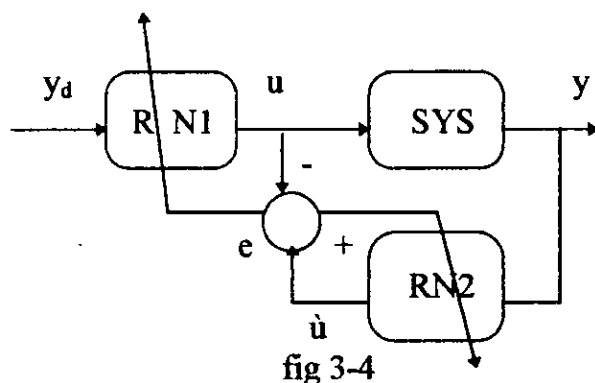


fig 3-4

$RN1$  est entraîné pour minimiser  $(u-\hat{u})$ , dans le but de minimiser  $(y_d-y)$ , malheureusement se n'est pas dans tous les cas que la minimisation de  $(u-\hat{u})$  entraîne la minimisation de  $(y_d-y)$ .

2- LA COMMANDE INVERSE :

La commande inverse consiste dans la mise en cascade du modèle inverse ,ainsi identifié, avec le système à commander afin de réaliser une fonction identité entre la sortie de ce dernier et la référence[10].

Cette approche est sensiblement liée à la fidélité du modèle inverse, La commande inverse est moins performante ,et moins robuste à cause de l'absence de bouclage. Ce problème peut être affranchi en utilisant l'apprentissage spécialisé (on-line) ,mais ce dernier présente un problème d'instabilité au début - le réseau commande directement le système - pour cela ,nous devons préparer les poids initiaux en utilisant un entraînement off-line (apprentissage généralisé), une fois le réseau a entré dans la zone de convergence ,il est installé en cascade avec le système à commander [10],[6].

### 3- LA COMMANDE INVERSE DE LA MAS A ORIENTATION PARFAITE DU CHAMP ROTORIQUE :

La recherche du sous espace des commandes utiles est essentielle lors de la commande inverse. Afin de délimiter cet espace nous allons tout d'abord effectuer la commande par bouclage linéarisant .

#### 3-1 Commande par bouclage linéarisant [18] :

Le modèle de la machine asynchrone à orientation du champ , alimenté en tension est un modèle non linéaire , cette machine présente un problème concernant le courant statorique qui atteint en régime transitoire des valeurs inacceptables , pour cela nous avons proposé une structure de commande appropriée qui permet de limiter le courant statorique, voir fig 3-5.

Une représentation de la MAS par un modèle linéaire et couplé est possible par le retour d'état linéarisant et une transformation difféomorphique . nous ne donnons pas ici l'étude théorique par ce que ce n'est pas notre but ,mais nous envoyons le lecteur à [18] pour plus de détails .

la commande par bouclage linéarisant se fait en 3 étape :

1. une transformation mathématique est appliquée au système ( MAS ) pour passer de son modèle non linéaire à une autre représentation d'état où ce dernier apparaît linéaire.
2. pour le modèle linéaire , la commande par retour d'état est ainsi appliquée.
3. le calcul de la commande est repris , mais cette fois pour le modèle non linéaire (découplage) .

Notons que Le degré relatif d'un système représente le nombre de fois qu'il faut dériver la sortie afin de faire apparaître l'entrée.

Si, la somme des degrés relatifs de toutes les sorties est égale au nombre d'états de système la linéarisation est dite totale, si non elle est dite partielle.

Soit le vecteur  $(x_1 \ x_2 \ x_3 \ x_4)$ , le vecteur d'état ,et  $(v_{ds}, v_{qs})$  le vecteur de commande tel que :

$$x_1 = i_{ds}, \quad x_2 = i_{qs}, \quad x_3 = \psi_{dr}, \quad x_4 = \omega_r.$$

le modèle de la machine à orientation du champ et le suivant :

$$\begin{aligned} \dot{x}_1 &= f_1 + \frac{v_{ds}}{L_s \sigma} \\ \dot{x}_2 &= f_2 + \frac{v_{qs}}{L_s \sigma} \\ \dot{x}_3 &= f_3 \\ \dot{x}_4 &= f_4 \end{aligned} \quad (3-1)$$

$$\begin{cases} f1 = -\left(\frac{1}{Ls\sigma} + \frac{1-\sigma}{Ts\sigma}\right)x1 + x4x2 + \frac{1-\sigma}{MTr\sigma}x3 + \frac{M}{Trx3}x2^2 \\ f2 = -\left(\frac{1}{Ts\sigma} + \frac{1-\sigma}{Tr\sigma}\right)x2 - x1x4 - \frac{1-\sigma}{M\sigma}x4x3 - \frac{M}{Trx3}x1x2 \\ f3 = \frac{M}{Tr}x1 - \frac{1}{Tr}x3 \\ f4 = \frac{MP^2}{JLr}x3x2 - \frac{Kf}{J}x4 - \frac{P}{J}Cr \end{cases} \quad (3-2)$$

où sous forme vectorielle :

$$\begin{aligned} \dot{x} &= f(x) + g1(x) \cdot vds + g2(x) \cdot vqs \\ y1 &= \begin{bmatrix} 0 & 0 & 1 & 0 \end{bmatrix} x^T \\ y2 &= \begin{bmatrix} 0 & 0 & 0 & 1 \end{bmatrix} x^T \\ g1(x) &= \begin{bmatrix} \frac{1}{Ls\sigma} & 0 & 0 & 0 \end{bmatrix}^T \\ g2(x) &= \begin{bmatrix} 0 & \frac{1}{Ls\sigma} & 0 & 0 \end{bmatrix}^T \end{aligned} \quad (3-3)$$

- calculons le degré relatif de la 1<sup>ère</sup> sortie  $y1=x3$

Soit :

$$z1 = h1(x) = x3$$

$$\frac{dz1}{dt} = \frac{\partial h1}{\partial x} \cdot \frac{dx}{dt}$$

d'où  $\dot{z1}$  devient :

$$\dot{z1} = f3(x).$$

on pose :

$$\frac{dz1}{dt} = z2$$

$$\frac{dz2}{dt} = \frac{\partial f3(x)}{\partial x} \cdot x$$

$$\text{d'où :} \quad \frac{dz2}{dt} = -\frac{1}{Tr} f3(x) + \frac{M}{Tr} f1(x) + \frac{M}{LsTr\sigma} vds$$

Le degré relatif de  $y_1$  est donc d'ordre 2, la nouvelle commande  $U_{11}$  est donnée par :

$$U_{11} = \frac{dz_2}{dt} = \frac{-1}{Tr} f_3(x) + \frac{M}{Tr} f_1(x) + \frac{M}{LsTr\sigma} v_{ds} \quad (3-4)$$

Ainsi la commande réelle est la suivante :

$$v_{ds} = v_{ds1} = [U_{11} + \frac{1}{Tr} f_3 - \frac{M}{Tr} f_1] \frac{M}{LsTr\sigma} \quad (3-5)$$

• Le degré relatif de la 2<sup>ème</sup> sortie  $y_2 = x_4$  :

soit :  $z_3 = h_2(x) = x_4$

$$\frac{dz_3}{dt} = \frac{\partial h_2(x)}{\partial x} \cdot x$$

d'où :  $\frac{dz_3}{dt} = f_4$

On pose :  $z_4 = \frac{dz_3}{dt}$  , Et de la même manière on établit  $\frac{dz_4}{dt}$  .

d'où :

$$\frac{dz_4}{dt} = \frac{MP^2}{LrJ} x_3 \cdot f_2 + \frac{MP^2}{LrJ} x_2 \cdot f_3 - \frac{Kf}{J} f_4 - \frac{MP^2}{LrLsJ\sigma} x_3 \cdot v_{qs}$$

Le degré relatif donc est de 2, et  $U_{22}$  est la nouvelle commande tel que :

$$\frac{dz_4}{dt} = U_{22}$$

$$U_{22} = \frac{MP^2}{LrJ} x_3 \cdot f_2 + \frac{MP^2}{LrJ} x_2 \cdot f_3 - \frac{Kf}{J} f_4 - \frac{MP^2}{LsLrJ\sigma} x_3 \cdot v_{qs} \quad (3-6)$$

La commande réelle  $v_{qs}$  est donnée par :

$$v_{qs} = v_{qs1} = \frac{LrLsJ\sigma}{MP^2 x_3} [U_{22} - \frac{MP^2}{LrJ} x_3 \cdot f_2 - \frac{MP^2}{LrJ} x_2 \cdot f_3 + \frac{Kf}{J} f_4] \quad (3-7)$$

- Le degré relatif de la composante iqs du courant statorique :

On pose

$$z5 = h3(x) = x2$$

$$\frac{dz5}{dt} = \frac{\partial h3(x)}{\partial x} \frac{dx}{dt}$$

$$\frac{dz5}{dt} = \dot{x}2 = f2 + \frac{1}{Ls\sigma} vqs$$

Ainsi le degré relatif est d'ordre 1 ,la commande délivrée par le régulateur de iqs est U21 tel que :

$$U21 = \frac{dz5}{dt} \tag{3-8}$$

La commande réelle est :

$$vqs = vqs2 = Ls\theta \cdot U21 - Ls\theta \cdot f2 \tag{3-9}$$

- Le degré relatif de la composante ids du courant statorique :

On pose :

$$z6 = h4(x) = x1$$

$$\frac{dz6}{dt} = \frac{\partial h4(x)}{\partial x} \frac{dx}{dt}$$

$$\frac{dz6}{dt} = \dot{x}1 = f1 + \frac{1}{Ls\sigma} vds$$

Ainsi le degré relatif est d'ordre 1 ,la commande délivrée par le régulateur de ids est U12 tel que :

$$U12 = \frac{dz6}{dt} \tag{3-10}$$

La commande réelle est :

$$vds = vds2 = Ls\sigma \cdot U12 - Ls\sigma \cdot f1 \tag{3-11}$$

Lorsque les courants ids,iqs sont en court de réglage , les commandes de la MAS sont, par ordre, vds2 et vqs2.

- La dynamique du système se met sous la forme suivante :

$$\frac{dz6}{dt} = U12 \quad \text{dynamique linéaire.}$$

$$z1 = f3 \quad \text{dynamique du zéros.}$$

$$\frac{dz5}{dt} = U21 \quad \text{dynamique linéaire.}$$

$$z3 = f4 \quad \text{dynamique du zéros.}$$

La transformation utilisée est celle ci-dessous :

$$\begin{aligned} z1 &= x3 \\ z6 &= x1 \\ z4 &= x4 \\ z5 &= x2 \end{aligned}$$

Une fois la limitation est réalisée ,les commandes commutent à vds1 et vqs1 et la dynamique du système se mettra alors sous forme de deux sous système :

$$\begin{cases} \dot{z1} = z2 \\ \frac{dz2}{dt} = U11 \end{cases} \quad \text{1}^{\text{er}} \text{ sous système linéaire .}$$

$$\begin{cases} \dot{z3} = z4 \\ \frac{dz4}{dt} = U22 \end{cases} \quad \text{2}^{\text{eme}} \text{ sous système linéaire}$$

La transformation utilisée est celle ci-dessous :

$$\begin{aligned} z1 &= x3 \\ z2 &= f3 \\ z3 &= x4 \\ z4 &= f4 \end{aligned}$$

D1 , D2 ,L 1 , L2 sont respectivement les blocs de découplage et de limitation

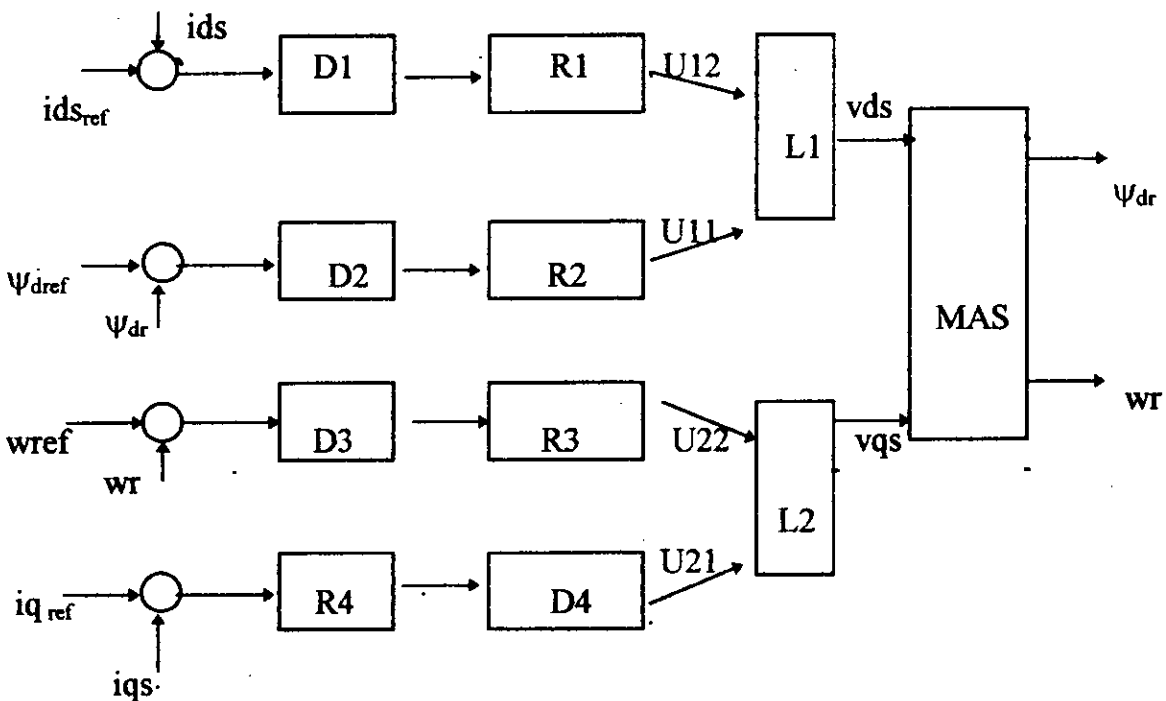


fig 3-5 Schéma de commande

**Synthèse des régulateurs :**

Les signaux de commandes  $U_{11}$ ,  $U_{12}$ ,  $U_{22}$ ,  $U_{21}$  sont synthétisés en utilisant les différents sous systèmes linéaire par la technique du retour d'état, d'où on écrit :

$$U_{11} = K_{1_{ref}} \psi_{ref} - K_{1z1} - K_{2z2}$$

$$U_{12} = K_{2_{ref}} id_{ref} - K_3 z_6$$

(3-19)

$$U_{22} = K_{3_{ref}} w_{ref} - K_{4z3} - K_{5z4}$$

$$U_{21} = K_{4_{ref}} iq_{ref} - K_{6z5}$$

tel que :

$\psi_{ref}$  : la référence du flux

$id_{ref}$  : la référence du courant ids

$w_{ref}$  : la référence de la vitesse

$iq_{ref}$  : la référence du courant iqs

$K_{1_{ref}}$ ,  $K_{2_{ref}}$ ,  $K_{3_{ref}}$ ,  $K_{4_{ref}}$ ,  $K_1$ ,  $K_2$ ,  $K_3$ ,  $K_4$ ,  $K_5$ ,  $K_6$ , sont les coefficients de la contre réaction d'état .

**Résultats de simulation :**

Nous avons choisi l'algorithme de RUNGE KUTTA d'ordre 4 pour simuler la MAS , et un pas d'échantillonnage de  $10^{-3}$  .

En effectuant la simulation , on a constaté que durant le réglage de ids le flux présente une dynamique satisfaisante.

Mais lors de la commutation des oscillations sévères apparaissent dans les réponses des courants ainsi la dynamique du flux se dégrade .

Pour cette raison nous avons supprimé le bloc de commutation L1 et le signal de commande vds est assuré par le régulateur R1 .

En faisant un placement de pôles approprié , les coefficients du retour d'état sont les suivants :

$$K_{3_{ref}} = K_4 = 1080 , K_5 = 100$$

$$K_{2_{ref}} = K_3 = 650 , K_{4_{ref}} = K_6 = 690 .$$

**Commentaire (planche -2-):**

La dynamique du flux est très rapide avec un temps de réponse de 0.0415 (s), et une erreur statique nulle.

Le courant ids suit sa consigne parfaitement , la réponse est sous forme d'échelon

La vitesse présente une dynamique très satisfaisante avec un temps de réponse de 0.681(s), et une erreur statique nulle.

La perturbation est appliquée sous forme d'un échelon de couple résistant dont la valeur est indiquée dans l'annexe.



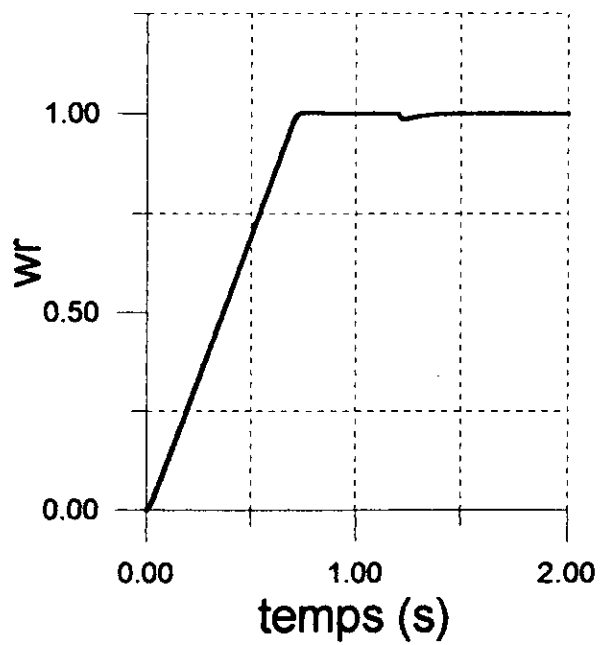
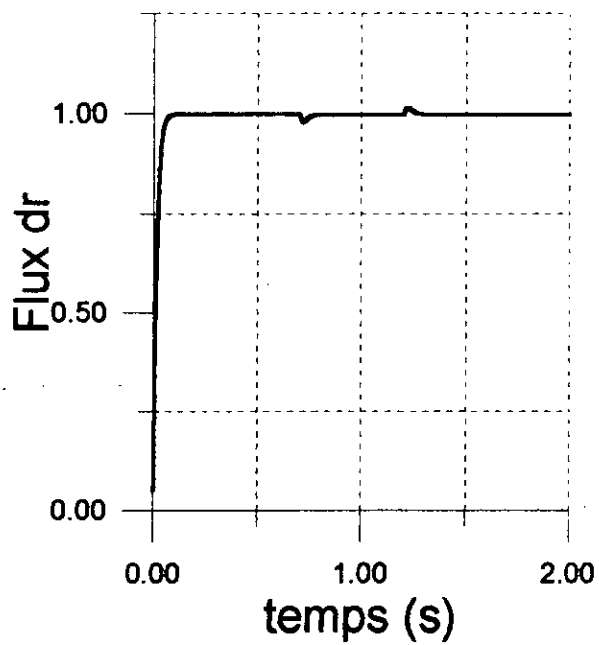
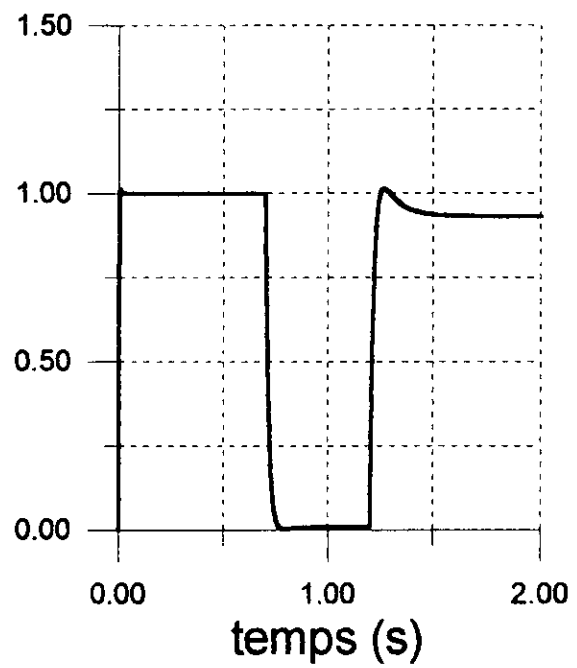
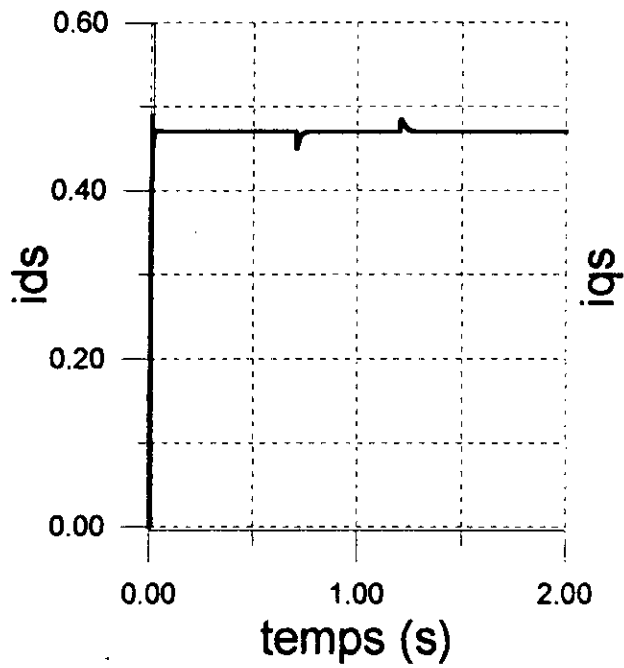


Planche -2- simulation de la commande par bouclage linéarisant de la MAS

L'introduction du couple ne perturbe vraiment pas le flux ,et le courant ids , par contre la vitesse chute de sa valeur de consigne , mais elle la rejoint rapidement

Le courant iqs lui aussi suit parfaitement la consigne ,et lors de la commutation , il commence à tendre vers 0 ,il garde cette valeur jusqu'à l'application de la perturbation puis il croit rapidement pour compenser la charge introduite .

**3-2 Dimensionnement du réseau approximant l'inverse de la MAS :**

Dans le but de dimensionner les réseaux de neurones ,et à cause de la non-linéarité de la MAS, et l'absence d'un modèle échantillonné ,nous avons développé une approximation de ce modèle [Annexe II].

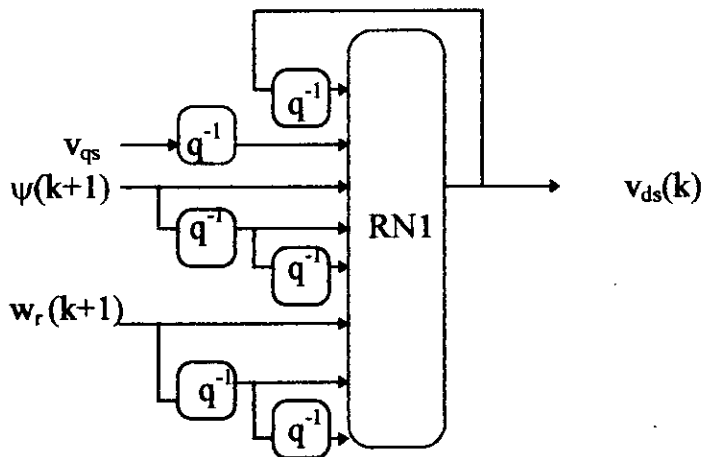
E-Slotine [20],donne aussi une méthode pour déterminer le nombre des entrées d'un réseau de neurones conçu pour approximer l'inverse du système.

Si n est le nombre des états mesurables d'un tel système non-linéaire à modèle inconnu, les entrées du réseau de neurones approximant l'inverse de ce système sont le vecteur dont les éléments sont toutes les états de ce système ainsi que leurs anciennes valeurs jusqu'à l'ordre n.

Dans notre cas de la MAS, il y'a 4 grandeurs d'états (  $i_{ds}, i_{qs}, \psi_{dr}, w_r$  ),mais pour minimiser le nombre des entrées du réseau, seuls le flux et la vitesse sont utilisés .

**3-3 Synthèse des régulateurs neuronaux .:**

Pour établir le modèle inverse tout en ayant un fonctionnement satisfaisant ,deux réseaux ont été proposés ,le premier approxime la commande  $v_{ds}$  ,et l'autre donne la commande  $v_{qs}$  (fig 3-5)



$$v_{ds}(k) = f'(\psi(k+1), w_r(k+1), \psi(k), w_r(k), \psi(k-1), w_r(k-1), v_{ds}(k-1), v_{qs}(k-1))$$

f' : fonction réalisée par RN1.

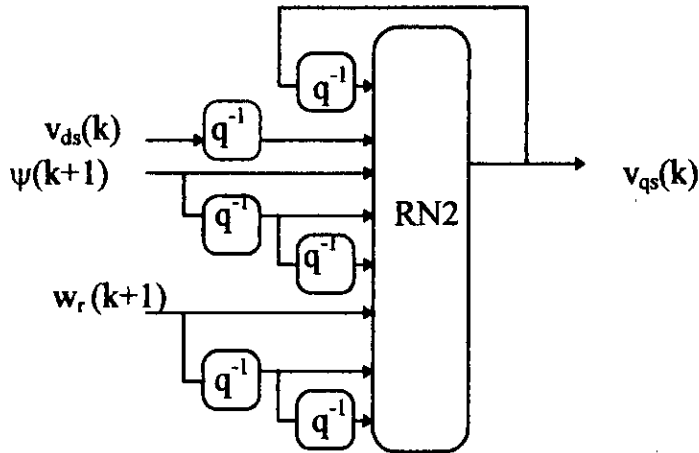


fig 3-5

$v_{ds}(k) = f^{**}(\psi(k+1), w_r(k+1), \psi(k), w_r(k), \psi(k-1), w_r(k-1), v_{ds}(k-1), v_{qs}(k-1))$   
 $f^{**}$  : fonction réalisée par RN1.

Le réseau RN1 est un réseau statique appartenant à la classe  $R_{8.15.20.1}$ , il en est de même pour le réseau RN2, notons ici que l'utilisation des valeurs passées introduit la dynamique nécessaire au réseau statique [10], [17].

Pour l'adaptation des poids nous avons utilisé l'algorithme de back-propagation, avec des pas d'apprentissage variables pour toutes les couches des réseaux (suivant la technique indiquée au chapitre 1).

les exemples d'entraînement sont pris à partir des résultats obtenus par la commande par bouclage linéarisant en faisant une variation de la consigne de la vitesse, ces exemples ont été sélectionnés en balayant le régime transitoire de la vitesse, et du flux ainsi que la zone où la perturbation a été appliquée.

Au début, le choix était fixé sur 150 exemples, mais après la phase de l'apprentissage les réseaux de neurones, n'ont pas pu généraliser, alors on a augmenté le nombre des exemples à 260.

Les deux réseaux sont entraînés off-line (apprentissage généralisé) jusqu'à ce que l'erreur atteigne la valeur  $10^{-2}$ . Par la suite, ils sont mis en cascade avec la MAS pour réaliser la commande inverse (fig 3-6).

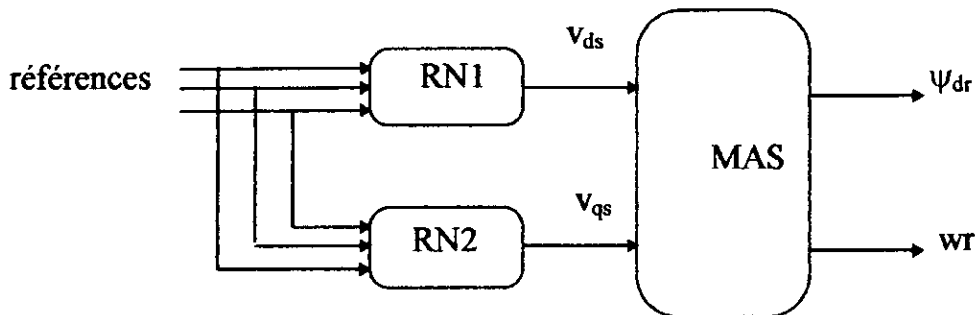


fig 3-6

### 3-4 Simulations et commentaires :

La simulation de la MAS alimentée en tension à orientation parfaite du champ rotorique est réalisée en utilisant l'algorithme de Range Kutta d'ordre 4, avec une période d'échantillonnage de  $10^{-3}$ .

- **Performance en régulation :**

Une charge nominale est introduite à l'instant  $t=0.9s$

la dynamique de la vitesse est assez rapide, avec un temps de réponse de  $0.673s$ , et une erreur permanente nulle.

Lors de l'application de la perturbation, la vitesse chute de  $0.4\%$  de sa valeur statique, mais cette chute s'annule rapidement.

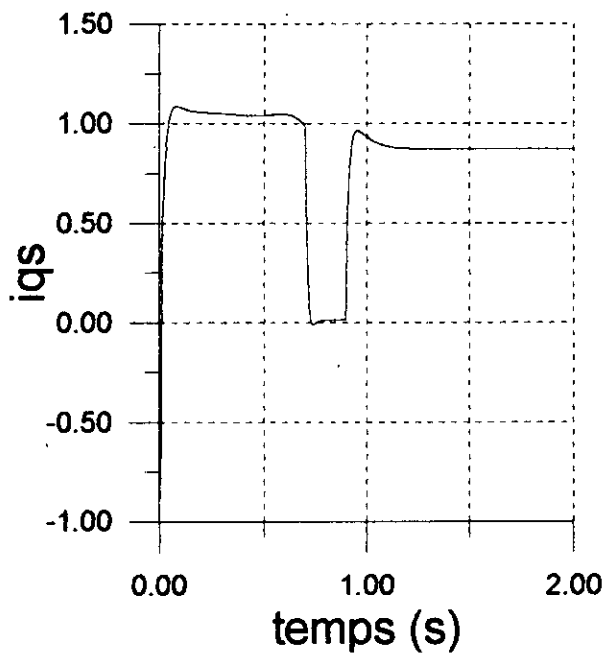
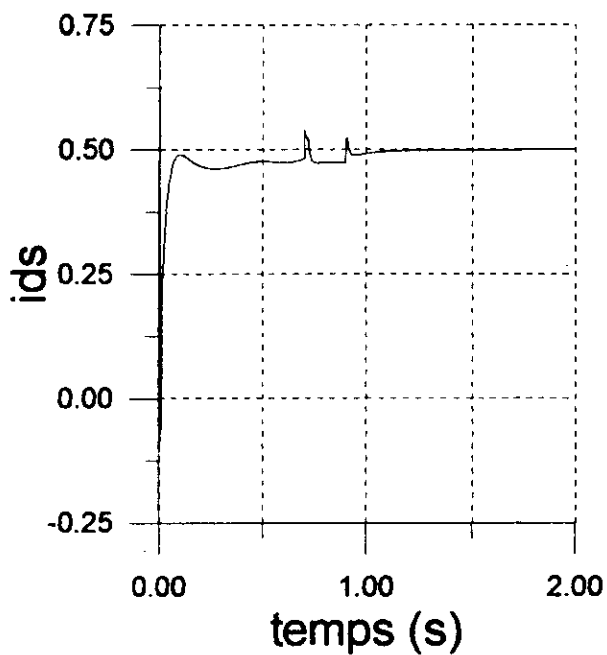
La réponse du flux est moins rapide que celle de la référence, avec un temps de réponse de  $0.077s$  et une erreur permanente de  $5.45\%$

On signale la présence d'un pic dans la réponse du flux ainsi que dans celle du courant  $i_{ds}$  dû à la commutation du courant  $i_{qs}$ , ce dernier passe à zéro après que la vitesse atteint son régime permanent, et il en reste, jusqu'à l'introduction de la perturbation, puis il croît rapidement pour compenser la charge introduite (planche 3).

- **Performance en poursuite :**

Pour une consigne de vitesse variante entre  $0.9$  à  $0.95$  puis à  $1$  de sa valeur nominale, la réponse de la vitesse est rapide, avec un temps de réglage de  $0.609s$  et une erreur permanente nulle.

Le flux a une dynamique très rapide, avec un temps de réponse de  $0.065s$ , et une erreur permanente de  $1\%$  (planche 5).



--- référence

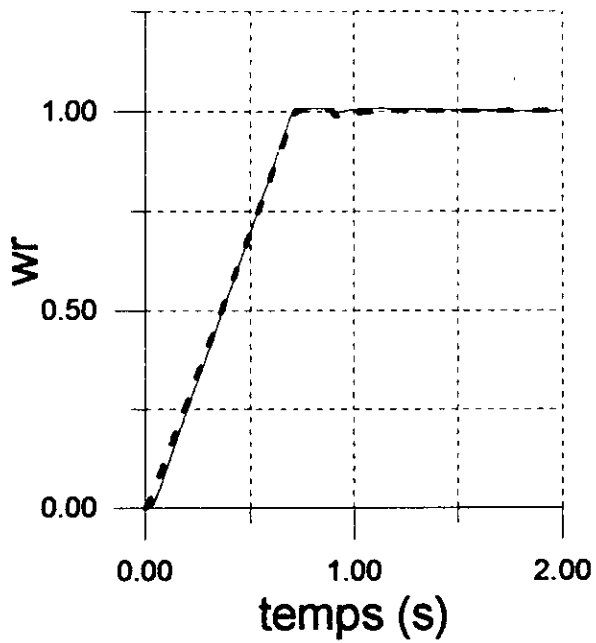
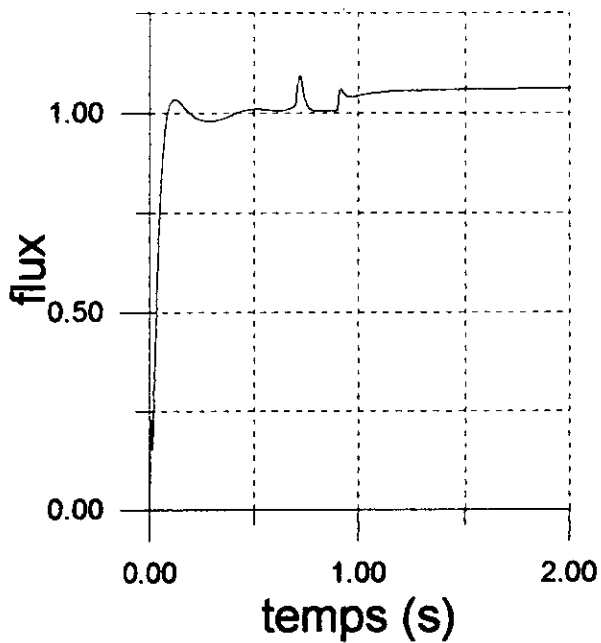


Planche-3- résultats de simulation de la commande inverse de la MAS à orientation parfaite du champ

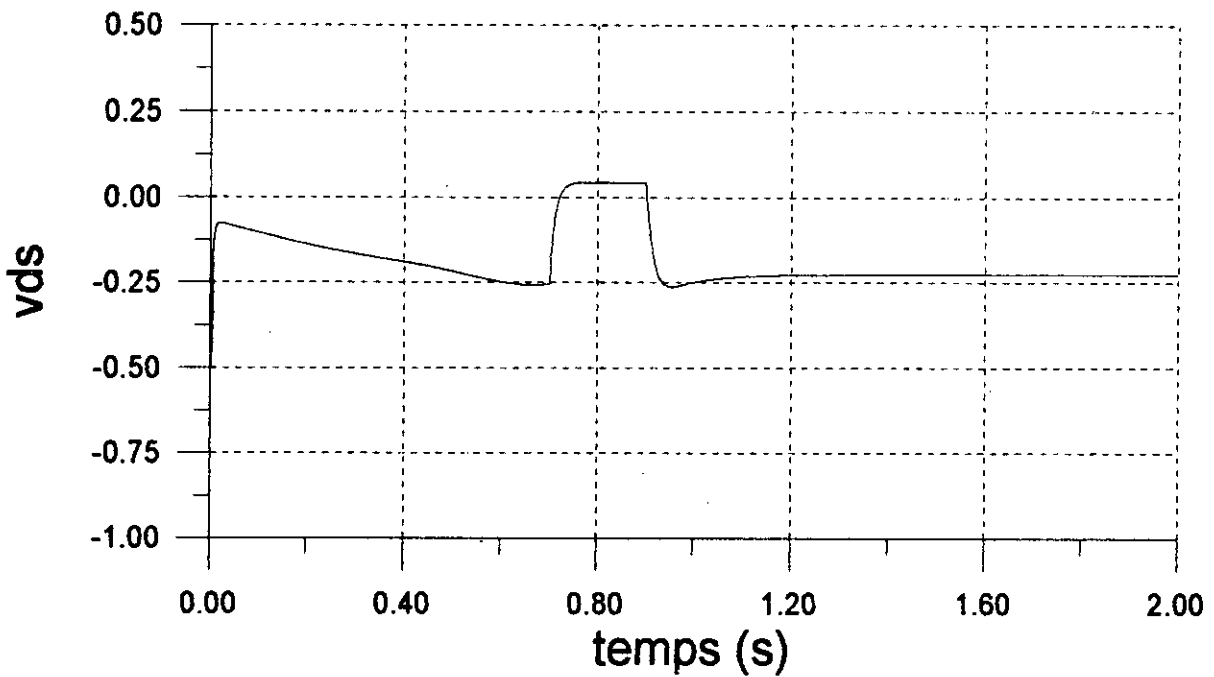
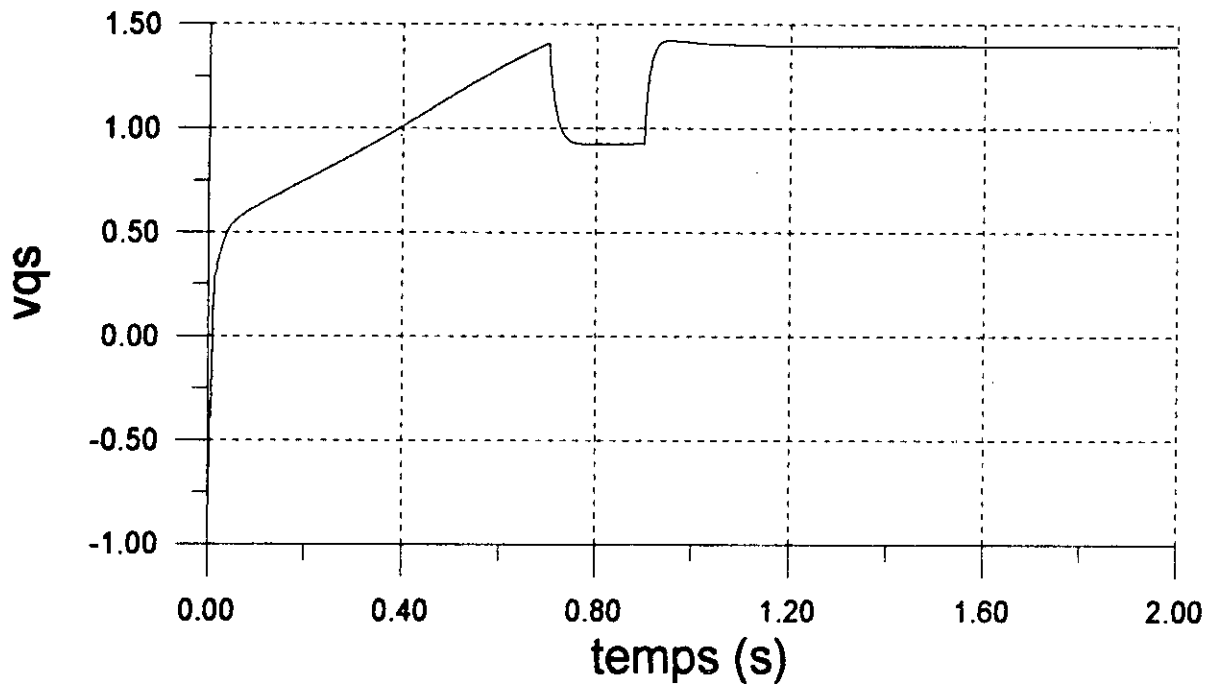
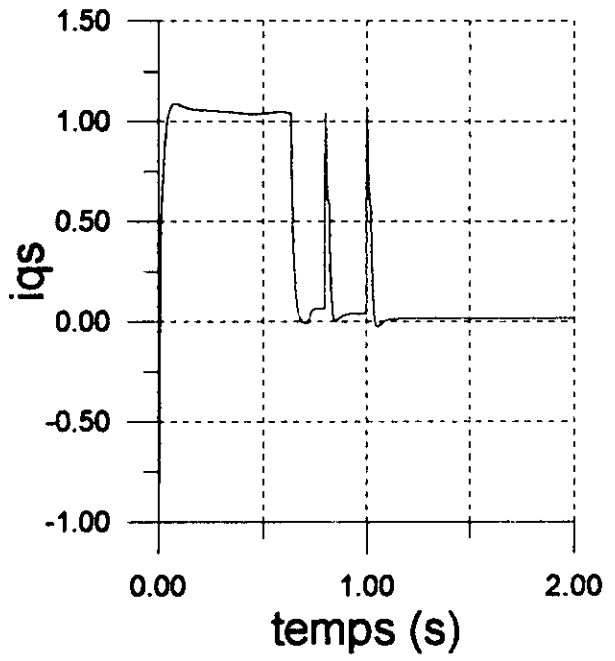
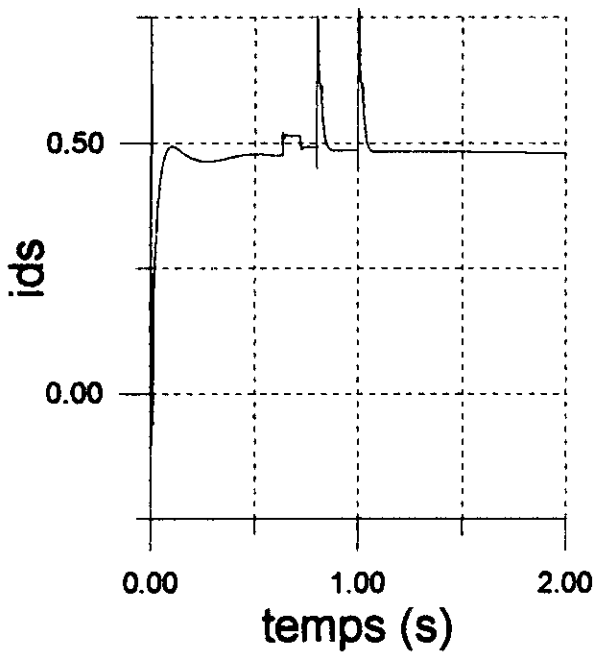


Planche-4-Les commandes délivrées par les régulateurs neuronaux RN1,RN2



--référence

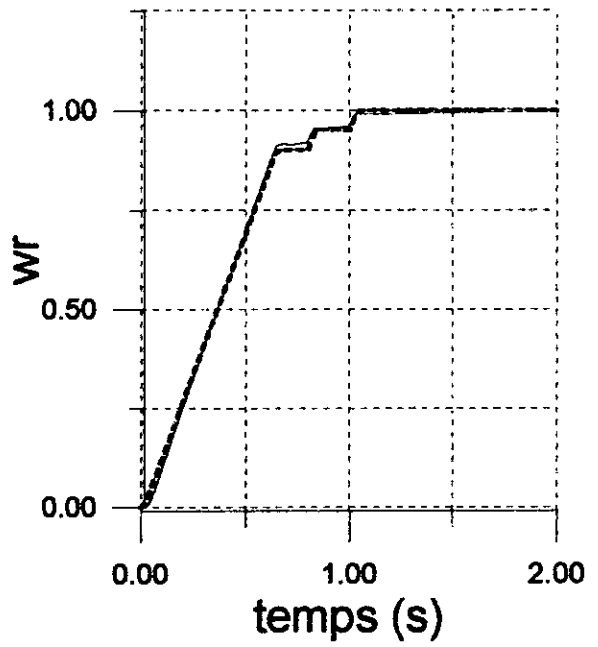
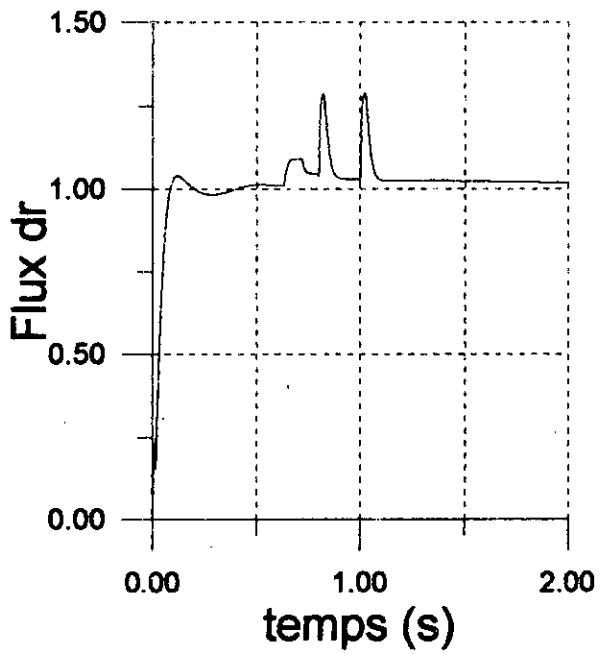


Planche -5 - variation de la consigne de la vitesse de 0.9-0.95-1 de  $w_{ref}$

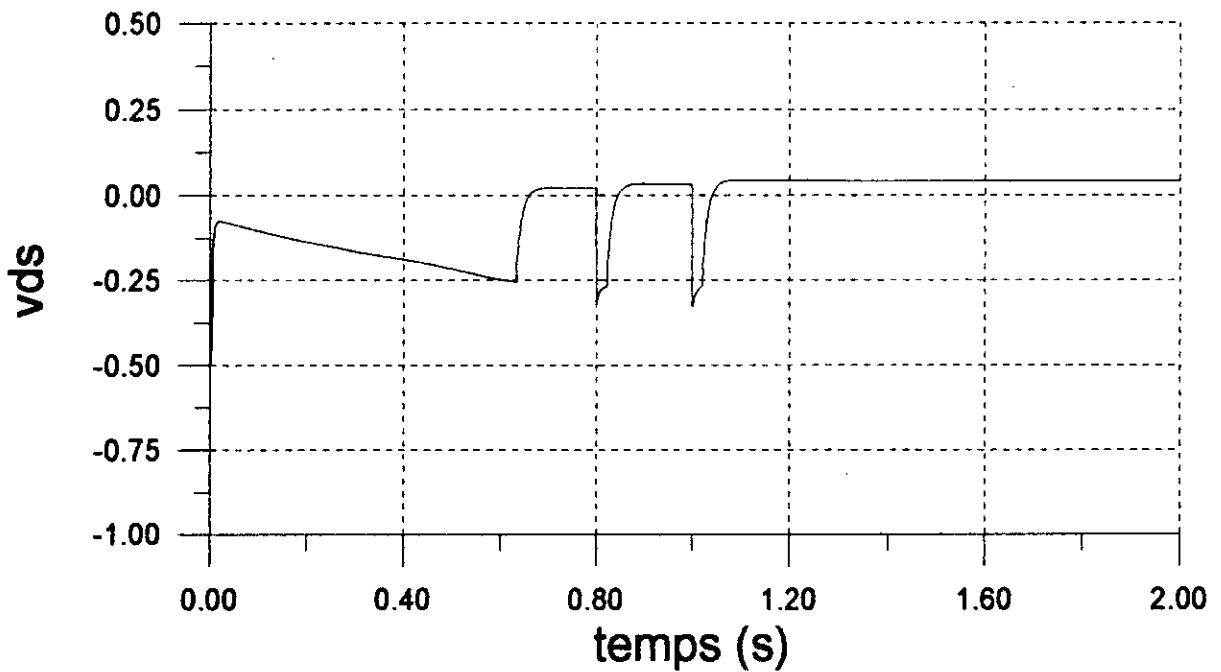
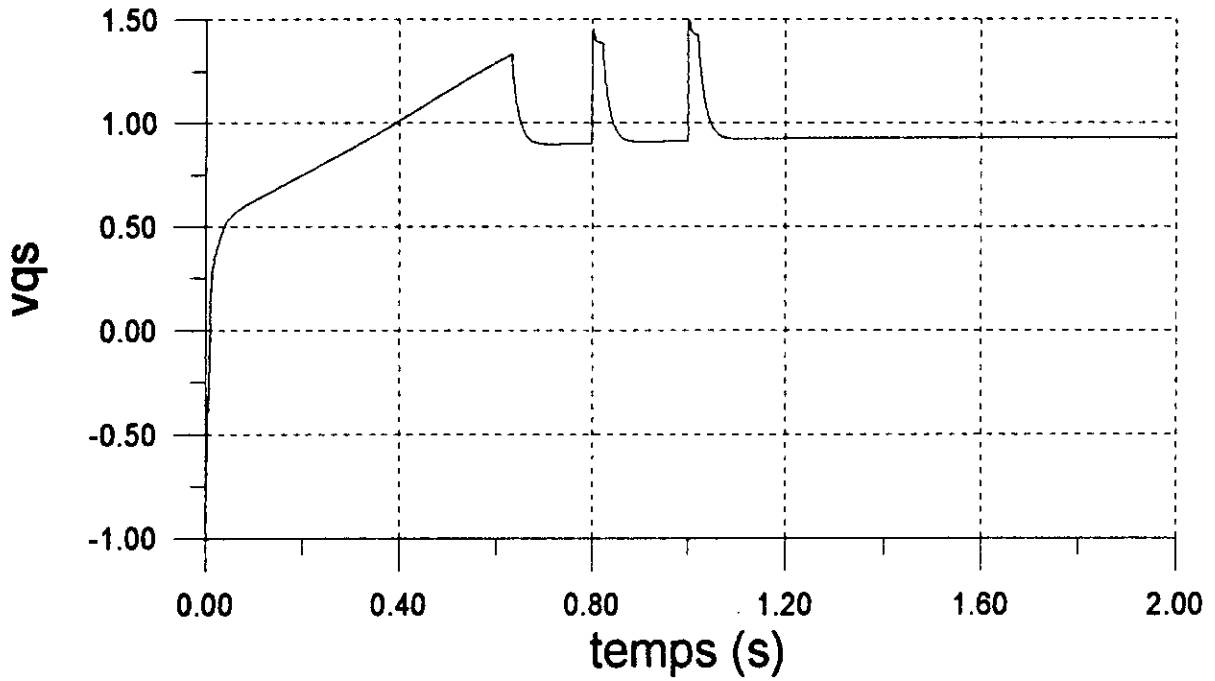


Planche-6- les commandes délivrées par les réseaux de neurones RN1,RN2



### 3-5 Tests de robustesse :

Nous avons effectué plusieurs tests de robustesse vis-à-vis les résistances statorique et rotoriques, et des inductances propres cycliques  $L_s$ ,  $L_r$ , les résultats obtenus sont les suivants :

- **vis-à-vis  $R_s$  :**

La variation de  $R_s$  était faite dans la plage  $R_{so} < R_s < 2R_{so}$ , nous avons remarqué des légères variation (atténuation) dans le régime transitoire du flux, par contre la réponse de la vitesse reste pratiquement inchangée (planche-7-).

- **vis-à-vis  $L_s$  :**

Au fur et à mesure que  $L_s$  diminue, la réponse de la vitesse s'amplifie, tout en gardant les mêmes allures pour les autres grandeur (planche-10-).

- **vis-à-vis  $R_r$  :**

de même que  $R_s$ , nous avons varié  $R_r$  de  $R_{so}$  à  $1.5R_{so}$ , au fur et à mesure que  $R_r$  croit, la vitesse sera atténuée en régime permanent, arrivant à la valeur  $1.5R_{so}$ , les performances de réglage se dégradent complètement, en régime transitoire des pics considérables apparaissent dans les réponses des courants, ainsi que le flux (planche-9)

- **vis-à-vis  $L_r$  :**

Pour une variation de l'inductance propre cyclique de  $0.7L_r$  à  $L_r$ , la réponse de la vitesse suit la consigne, cependant elle diverge après injection du couple résistant, la réponse du flux est de plus en plus atténuée (planche-8-)

### Conclusion :

La commande inverse de la MAS représentée par son modèle alimenté en tension à orientation parfaite de champ rotorique donne des résultats assez satisfaisantes pour la poursuite, avec un rejet de perturbation très rapide.

Mais lors des variation des paramètres de la machine, les régulateurs neuronaux ne réagissent pas de la manière souhaitable pour s'adapter avec les nouveaux paramètres.

Ceci revient à l'absence d'un feedback dans la structure de la commande inverse, où le réseau est isolé de la machine.

---

$-R_s = 1.5R_{s0}$

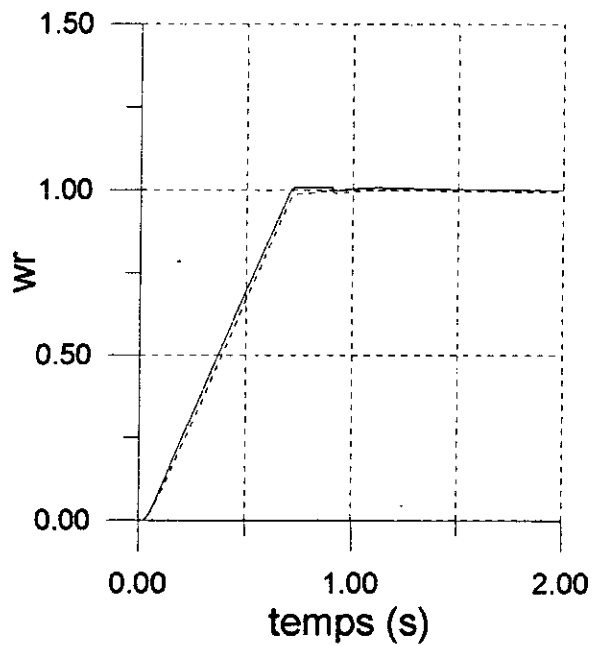
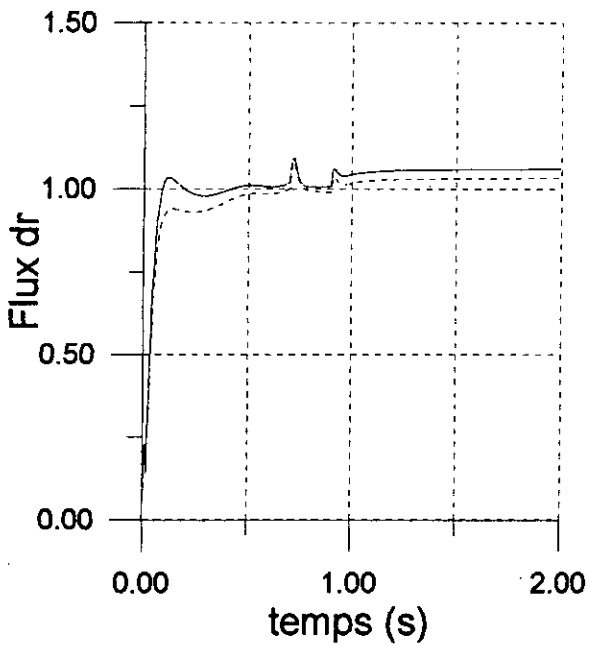
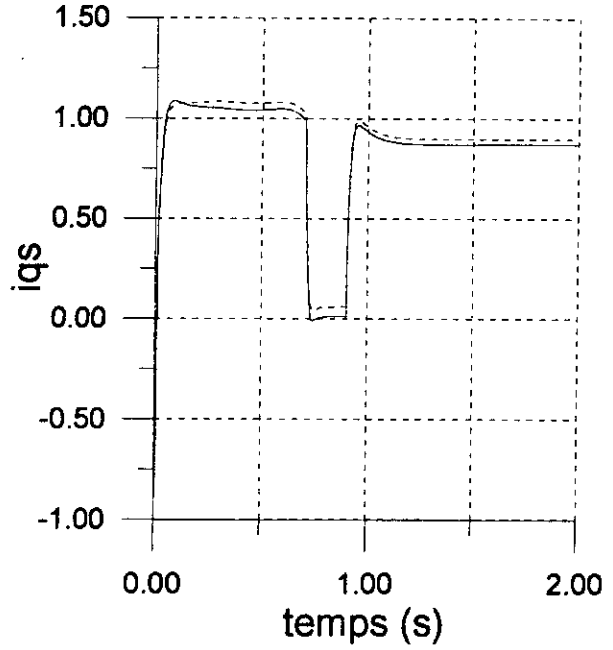
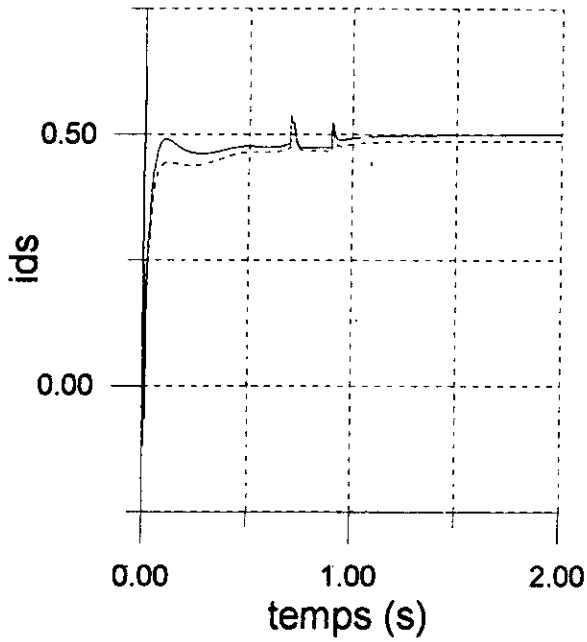


Planche-7- Test de robustesse vis-à-vis  $R_s$

---  $L_r = 0.8L_{ro}$

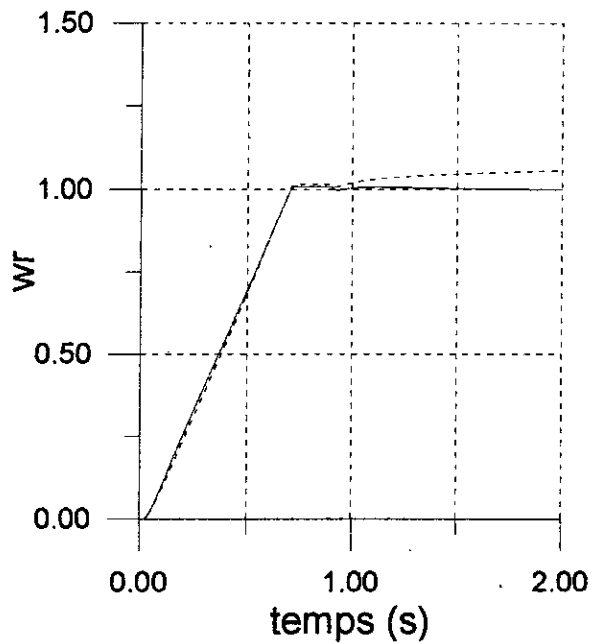
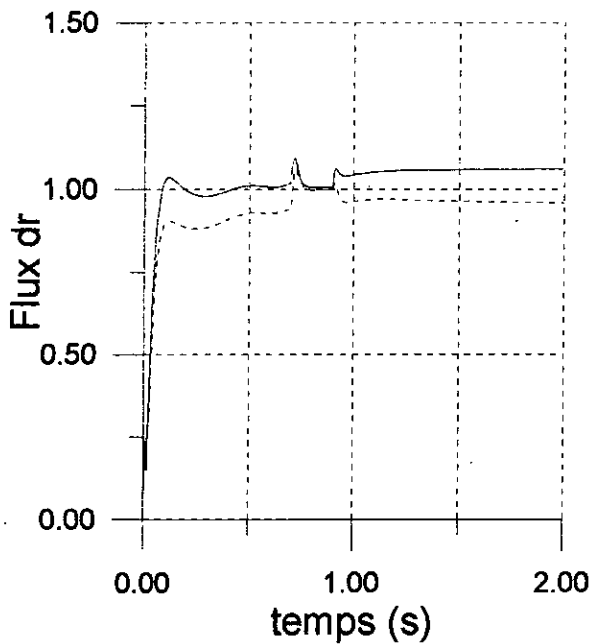
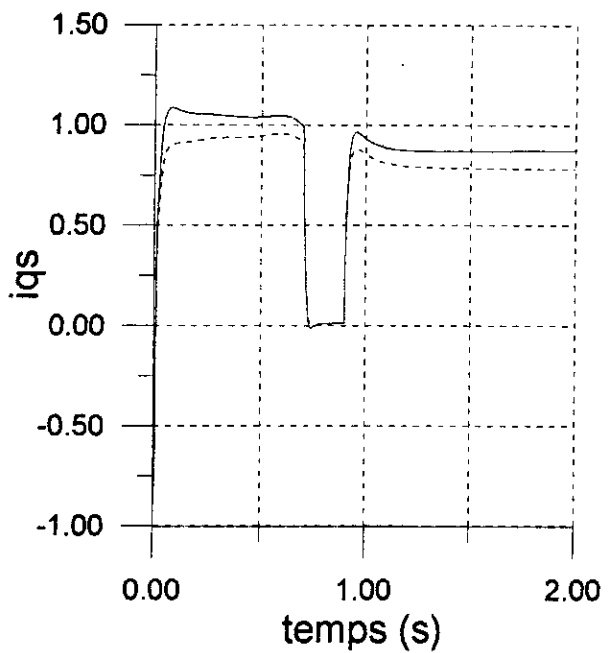
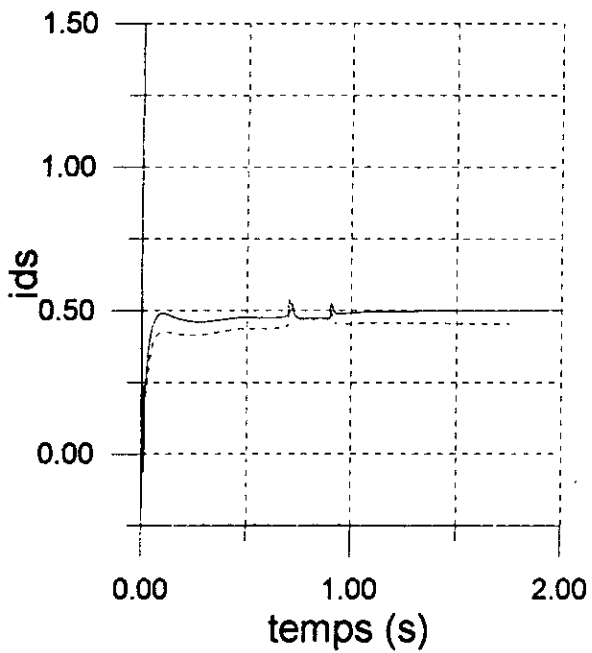


Planche -8-Test de robustesse vis-à-vis  $L_r$ .

—  $R_r = 1.5R_{r0}$

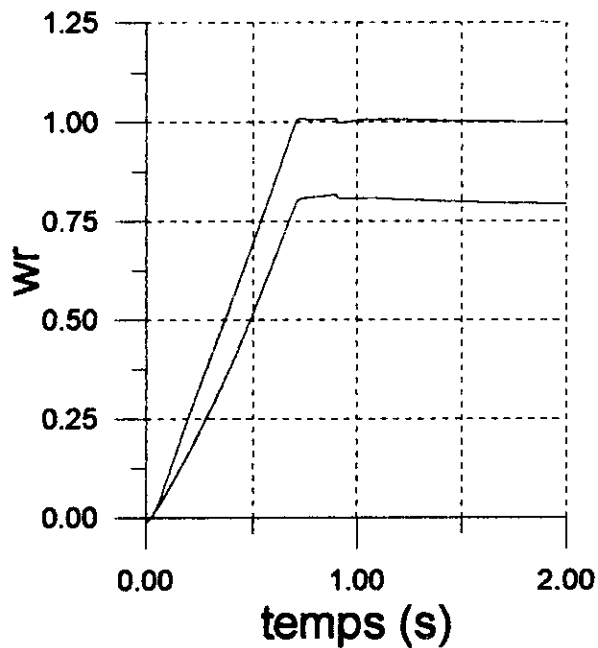
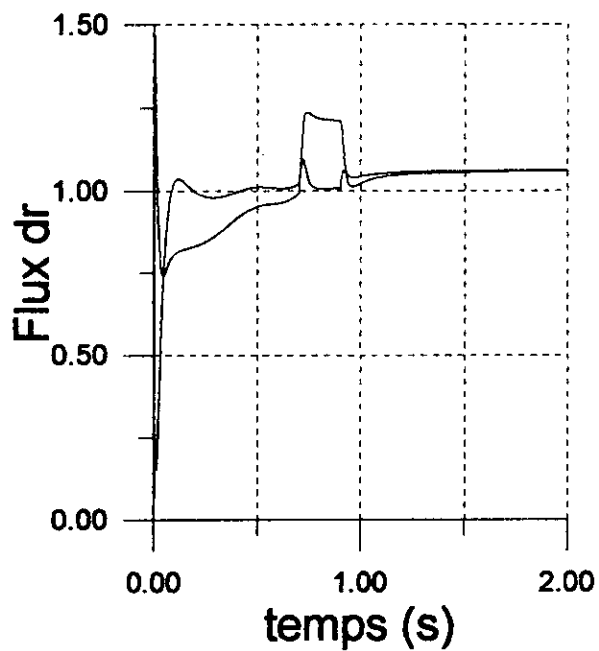
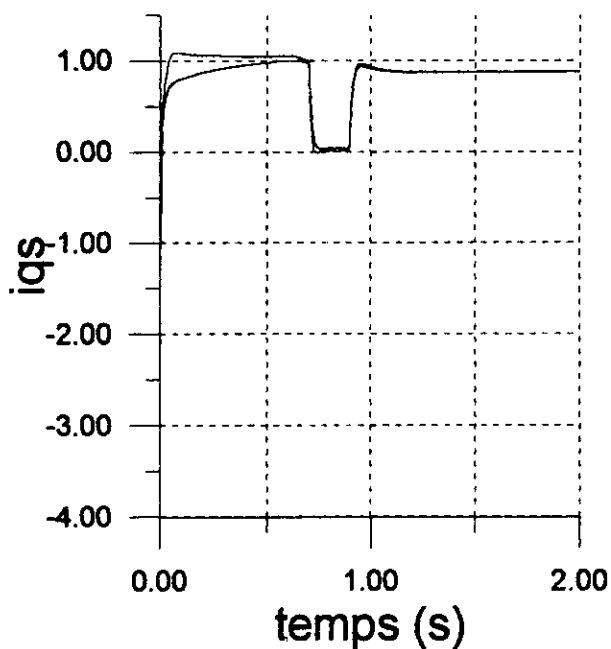
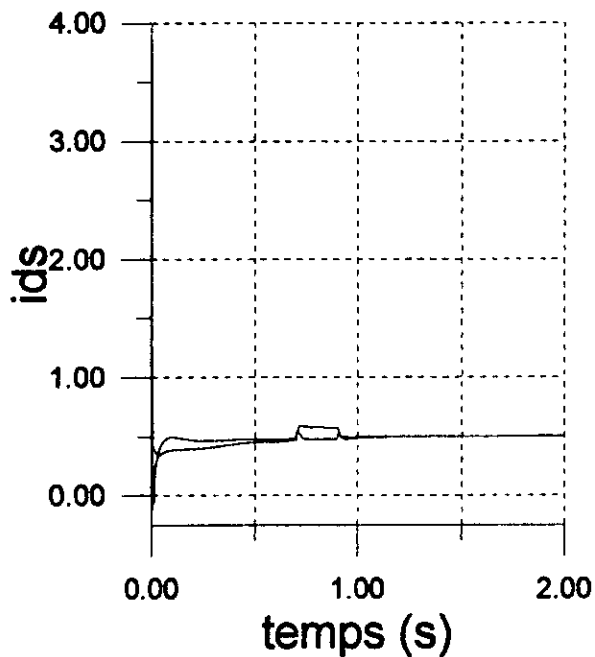


Planche-9-Test de robustesse vis-à-vis  $R_r$

-  $L_s = 1.5L_{s0}$

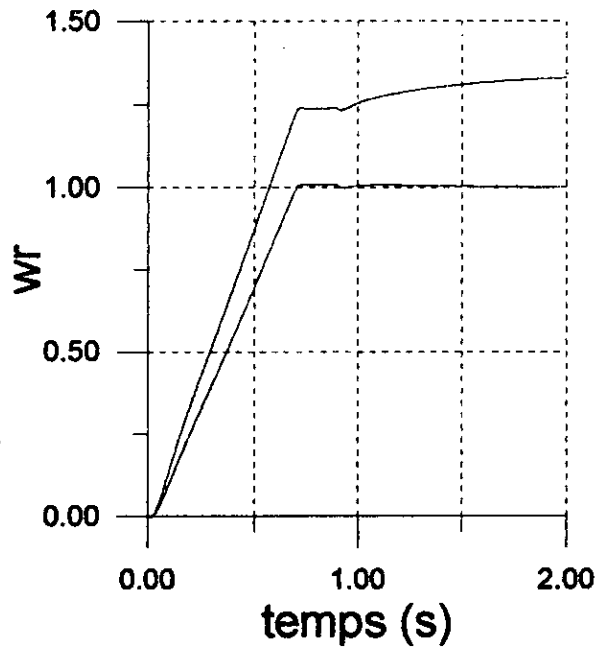
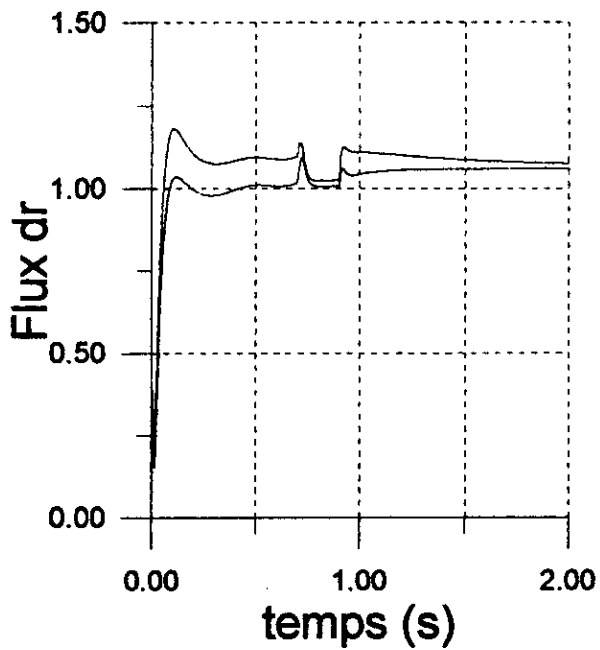
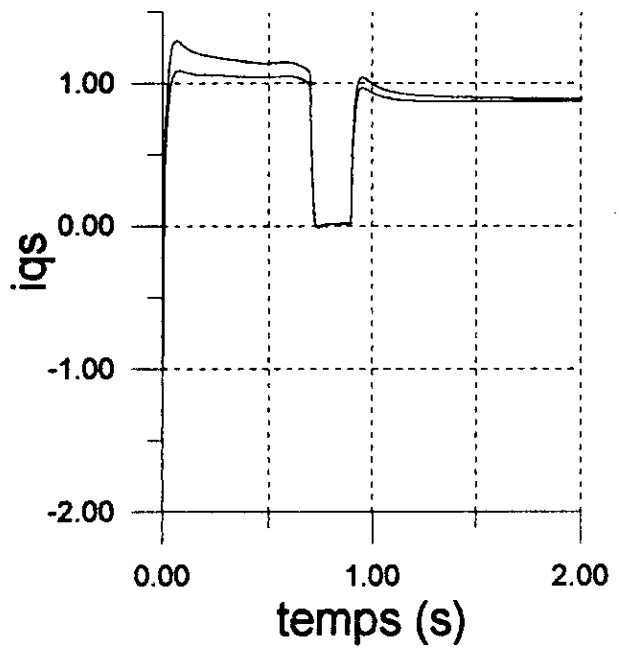
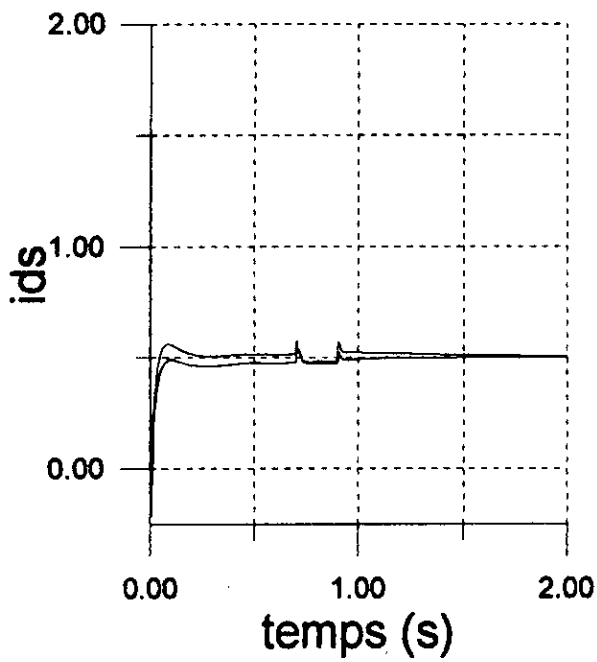


Planche-10- Test de robustesse vis-à-vis  $L_s$

# CHAPITRE . IV

**INTRODUCTION :**

En réalité, la MAS représentée par son modèle dans le repère synchrone lié au champ tournant rotorique, nécessite en plus des commandes en tension la commande de la pulsation statorique, où bien la vitesse de glissement  $w_{sl}$  :

$$w_{sl} = w_s - w_n \quad (\text{fig 4-1}).$$

Dans ce chapitre, on propose une structure neuronale qui permet de déterminer  $w_s$  à partir des commandes produites par le modèle à orientation parfaite du champ rotorique.



fig 4-1

**1- MODELE DE LA MAS ALIMENTEE EN TENSION DANS LE REPERE SYNCHRONE :**

On admet les grandeurs  $i_{qs}$ ,  $i_{ds}$ ,  $\psi_{dr}$ ,  $\psi_{qr}$ ,  $w_r$  comme vecteur d'état, et  $v_{ds}$ ,  $v_{qs}$ ,  $w_s$  comme vecteur de commande :

$$\begin{cases} \dot{x}_1 = -\left(\frac{1}{\sigma \cdot T_s} + \frac{1-\sigma}{\sigma \cdot T_r}\right)x_1 + x_2 \cdot v_3 + \frac{1-\sigma}{\sigma \cdot M \cdot T_r}x_3 + \frac{1-\sigma}{\sigma \cdot M}x_4x_5 + \frac{1}{\sigma L_s}v_1. \\ \dot{x}_2 = -\left(\frac{1}{\sigma \cdot T_s} + \frac{1-\sigma}{\sigma \cdot T_r}\right)x_2 - x_1 \cdot v_3 + \frac{1-\sigma}{\sigma \cdot M \cdot T_r}x_4 - \frac{1-\sigma}{\sigma \cdot M}x_3x_5 + \frac{1}{\sigma \cdot L_s}v_2. \\ \dot{x}_3 = \frac{M}{T_r}x_1 - \frac{1}{T_r}x_3 + (v_3 - x_5)x_4. \\ \dot{x}_4 = \frac{M}{T_r}x_2 - \frac{1}{T_r}x_4 - (v_3 - x_5)x_3. \\ \dot{x}_5 = \frac{M \cdot p^2}{J \cdot L_r}(x_3x_2 - x_4x_1) - \frac{K_f}{J}x_5 - \frac{p}{J}C_r. \end{cases}$$

$$X = [i_{qs}, i_{ds}, \psi_{dr}, \psi_{qr}, w_r], \quad U = [v_{ds}, v_{qs}, w_s]$$

**2 Synthèse de réseau de ws :**

L'expression de la pulsation statorique , qui réalise l'orientation du champ rotorique , est la suivante

$$ws = wn + \frac{M}{Tr} \cdot \frac{iqs}{\Psi dr}$$

Dans ce qui suit , ws est approximé par un réseau statique appartenant à la classe  $R_{5 \ 10 \ 10 \ 1}$  .

Pendant la phase d'apprentissage , on a utilisé 260 exemples pour ajuster les poids des réseaux en utilisant l'algorithme de la rétropropagation de l'erreur avec des pas variables .

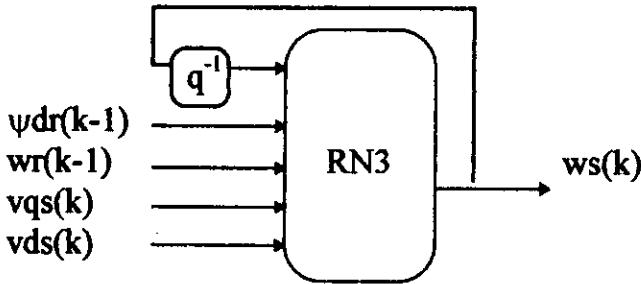


fig -4 -2

$$ws(k) = g( vds(k), vqs(k), wr(k-1), \psi_{dr}(k-1), ws(k-1))$$

**3-Structure de commande :**

Pour avoir les commandes , vds et vqs , on garde la structure précédente(chap3) L'injection de la commande ws dans le modèle réel impose à ce dernier le comportement de celui à orientation parfaite du champ rotorique .

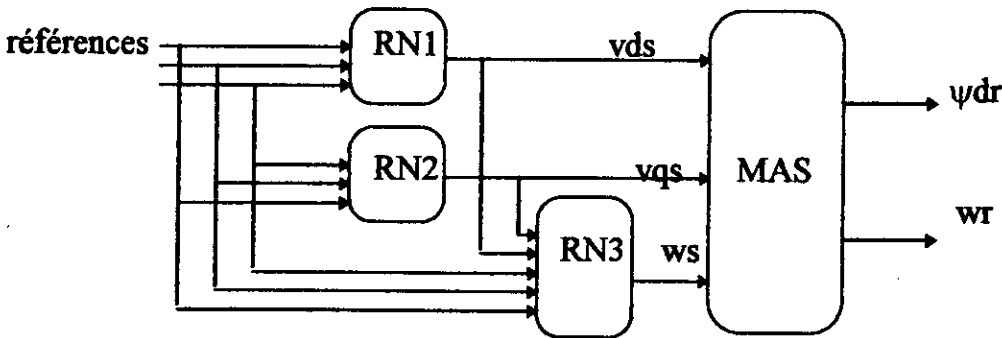


fig4-3



#### 4-Résultats de simulation

##### Performances en poursuite :

Comme pour le cas de la MAS à orientation parfaite du champ , nous avons effectué une variation de la consigne de vitesse de 0.9 à 0.95 puis à 1 wref.

Les performances dynamiques sont intéressantes , le temps de réponse de la vitesse est de 0.618(s) , sans dépassement , et une erreur statique nulle .

La réponse du flux est très lente par rapport au cas parfait , nous avons constaté que le flux monte en oscillant , son temps de réponse est de 0.216(s) , avec une erreur statique de 5.5% .(planche-13-)

La composante quaratique du flux rotorique ( $\psi_{qr}$ ) a une allure oscillatoire autour de zéro, donc l'orientation du champ est réalisée .(planche-14-)

##### Performances en régulation :

Après l'application de la perturbation , le flux reste au voisinage de la référence avec une erreur permanente de 7.37% .

La réponse de la vitesse suit parfaitement sa référence (planche-11).

#### 5- Tests de robustesse :

- vis-à-vis  $R_s$  :

La variation de  $R_s$  était faite dans la plage :  $R_{s0} < R_s < 2R_{s0}$  .

La réponse de la vitesse varie légèrement dans le régime permanent . Les autres grandeurs restent inchangées (planche-15-)

- vis-à-vis  $L_s$  :

$$0.7L_{s0} < L_s \leq L_{s0}$$

Au fur et à mesure que  $L_s$  diminue , les réponses du flux , de vitesse et des courants s'amplifient (planche-18-).

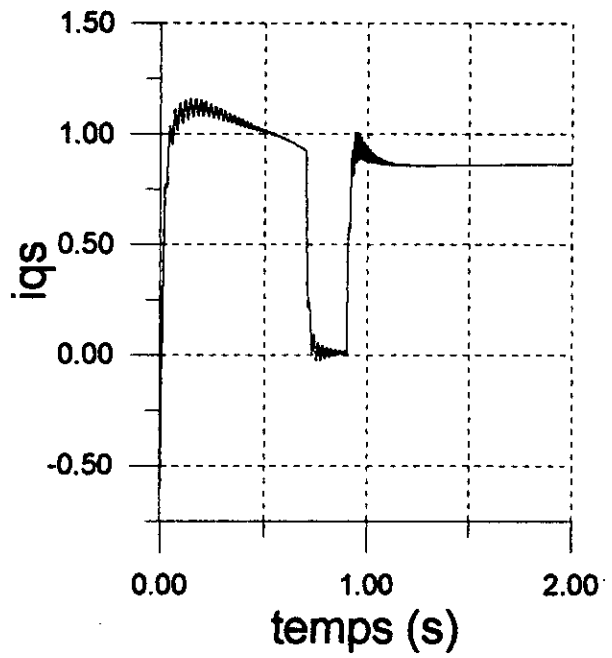
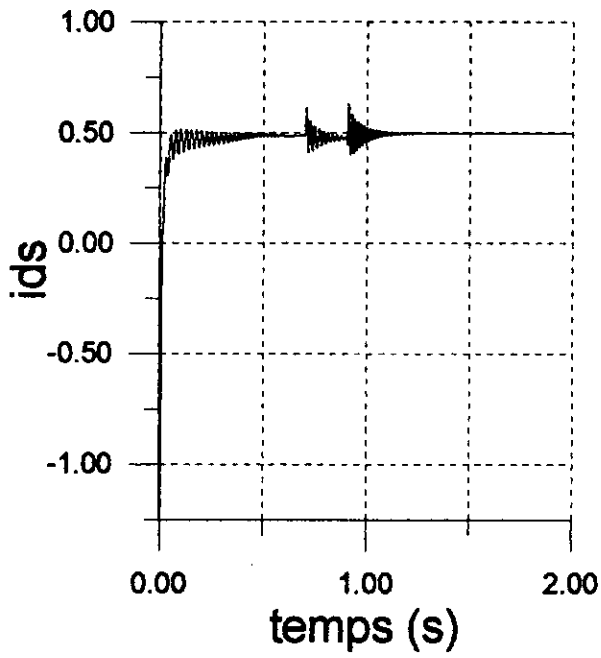
- vis-à-vis  $R_r$  :

$$R_{r0} \leq R_r \leq 2R_{r0}$$

Nous avons constaté une atténuation sur la vitesse,( planche-17-).

- vis-à-vis  $L_r$  :

Les allures des réponses du flux , de vitesse et des courants <sup>ne</sup> changent pas pour toute variation de  $L_r$  ( planche-16-).



—référence

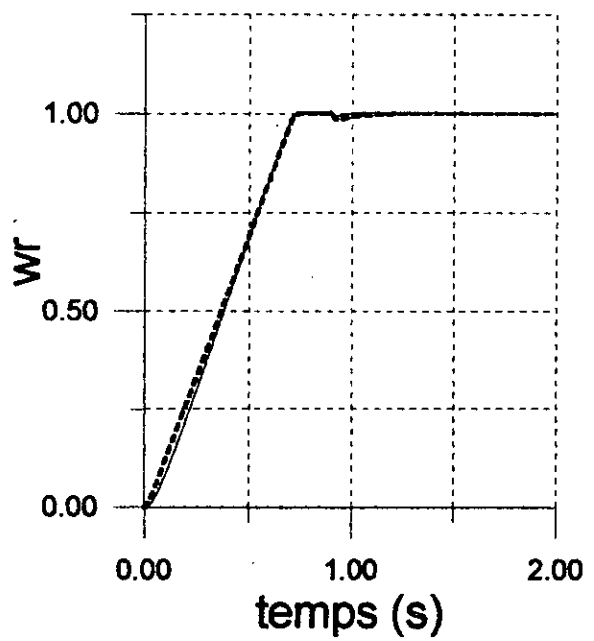
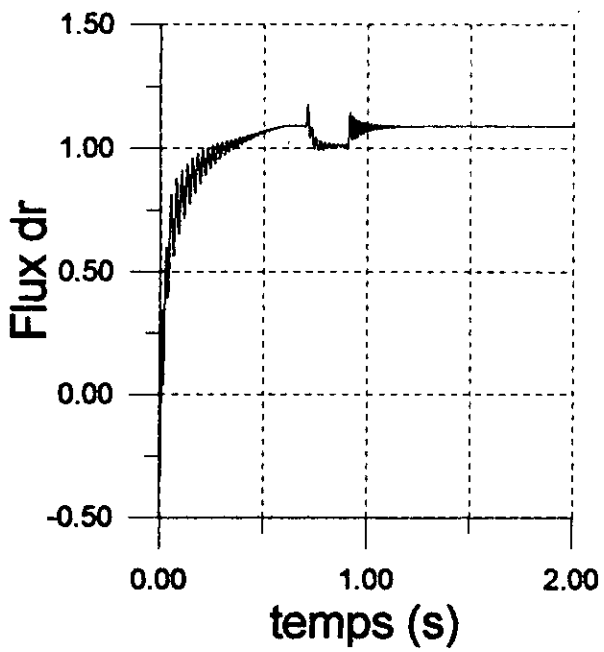
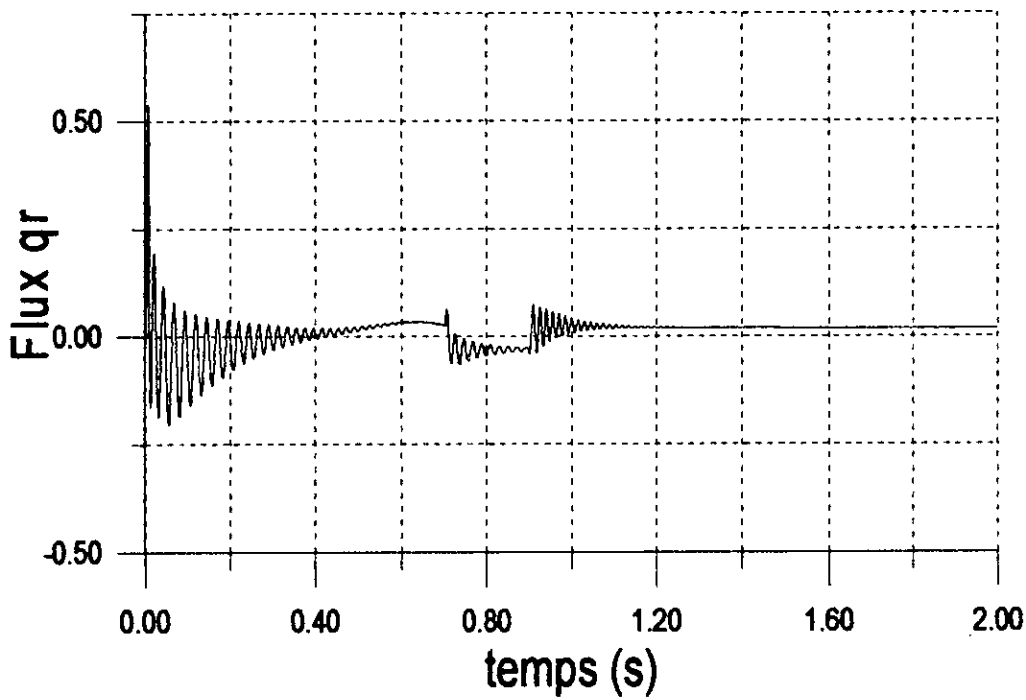


Planche-11- Réponse de simulation à une charge nominale introduite à  $t = 0.9$  (s)



- La composante quadratique du flux rotorique  $\Psi_{qr}$  -

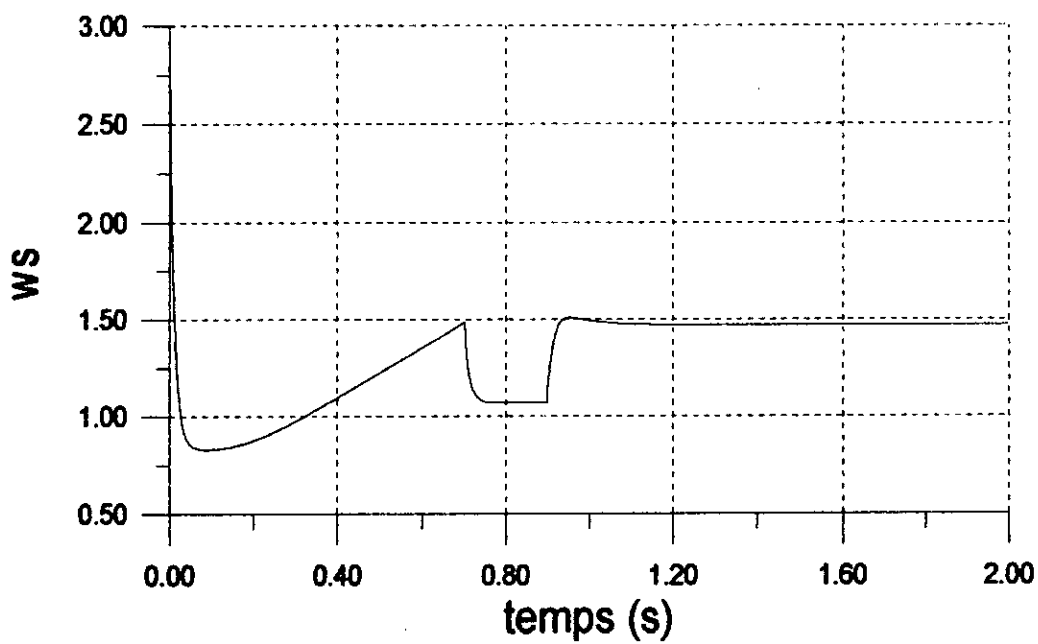


Planche-12- La pulsation statorique délivrée par le réseau de neurone RN3

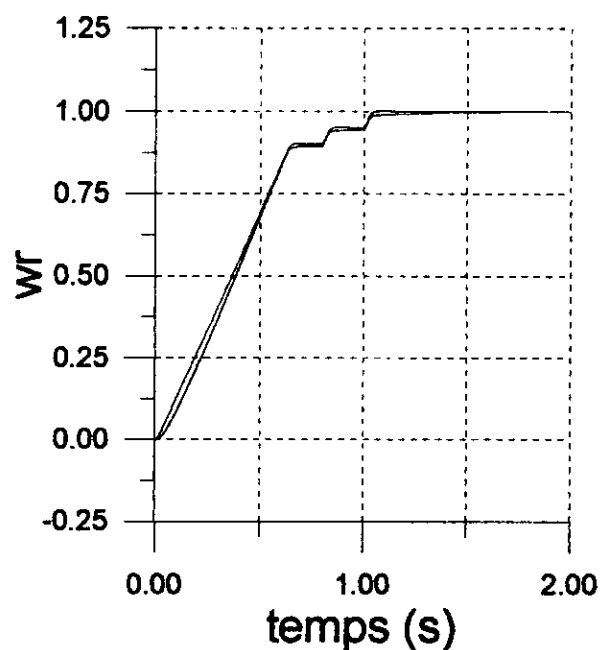
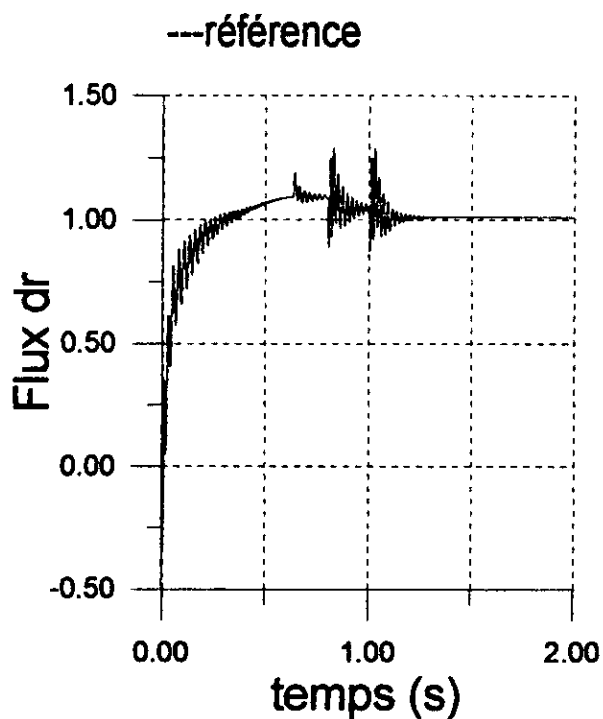
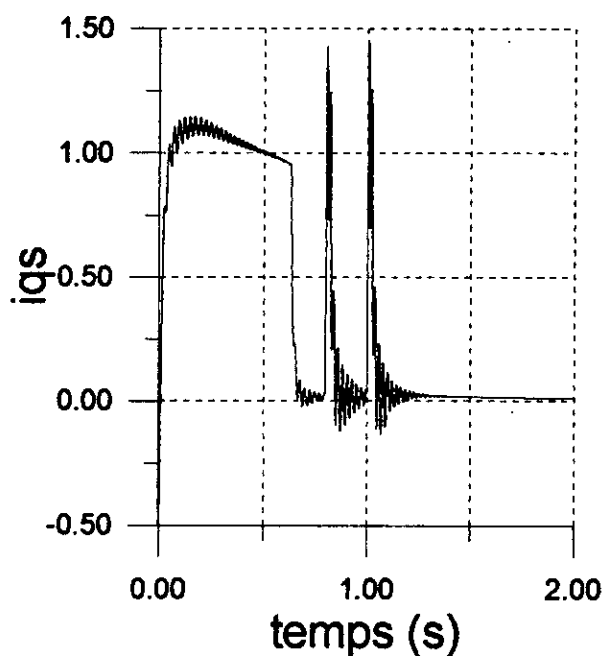
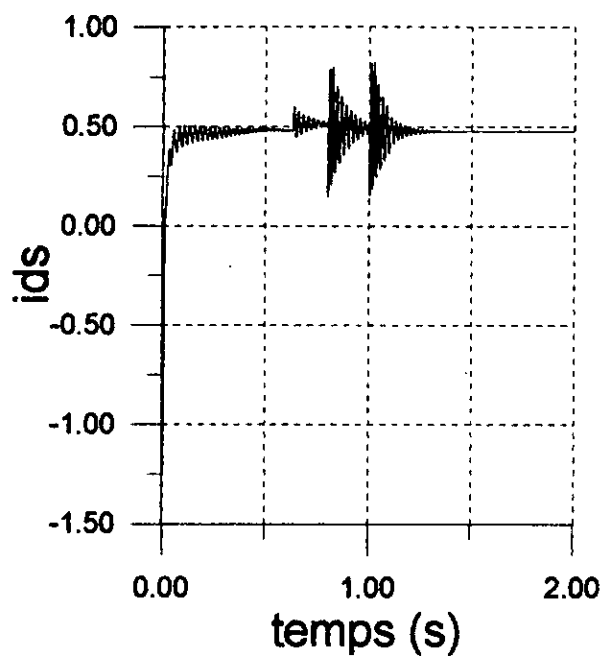


Planche-13- Réponse de la MAS à une variation de la référence de la vitesse de 0.95 à 1wref

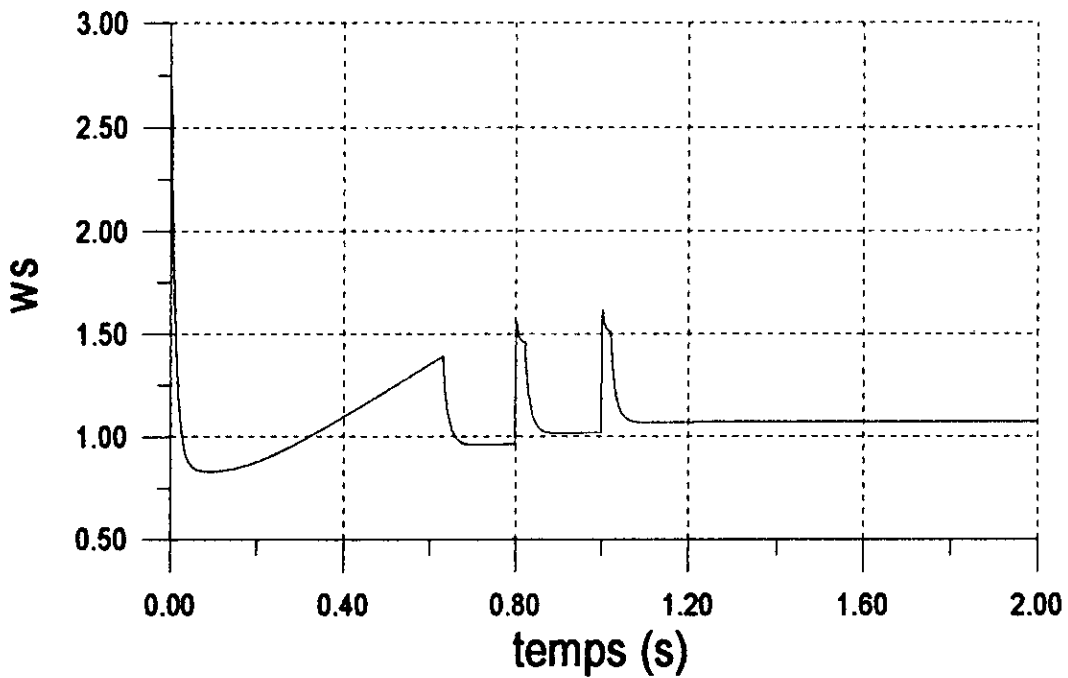
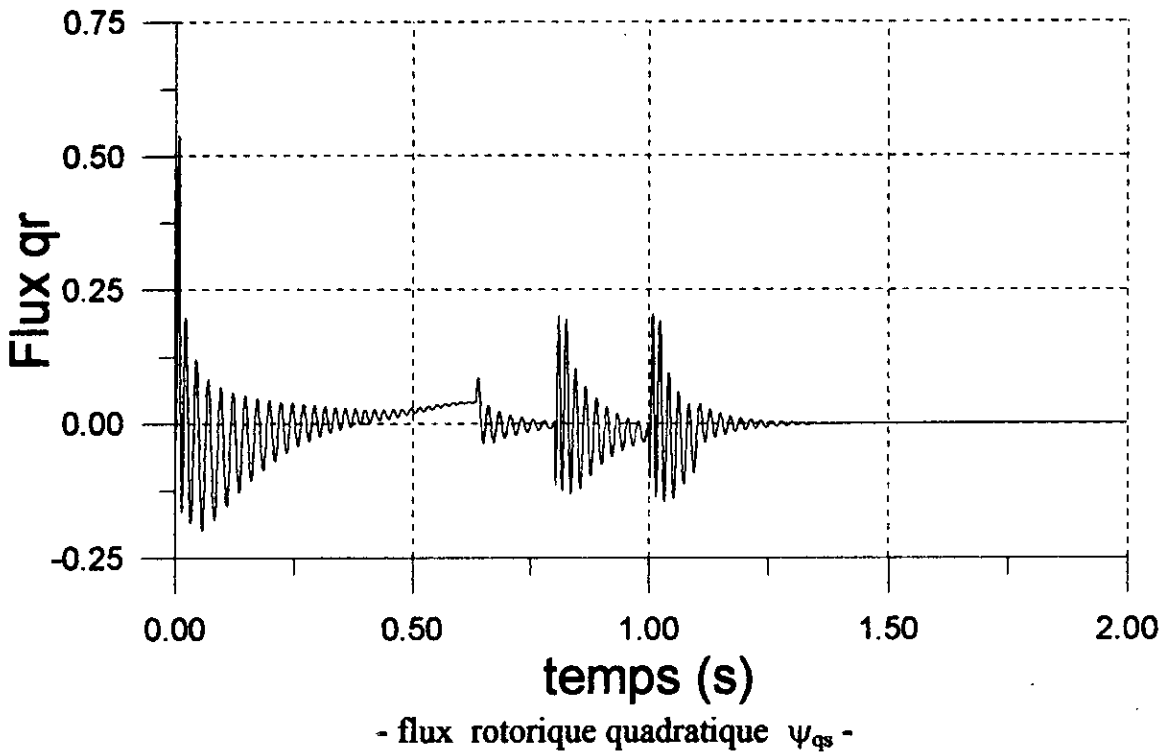


Planche-14- Pulsation statorique délivrée par le régulateur neuronal RN3 -

---  $R_s = 1.5R_s$

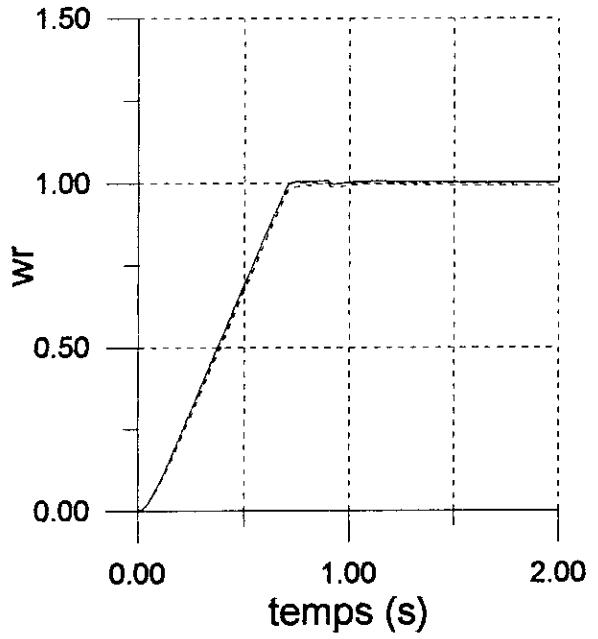
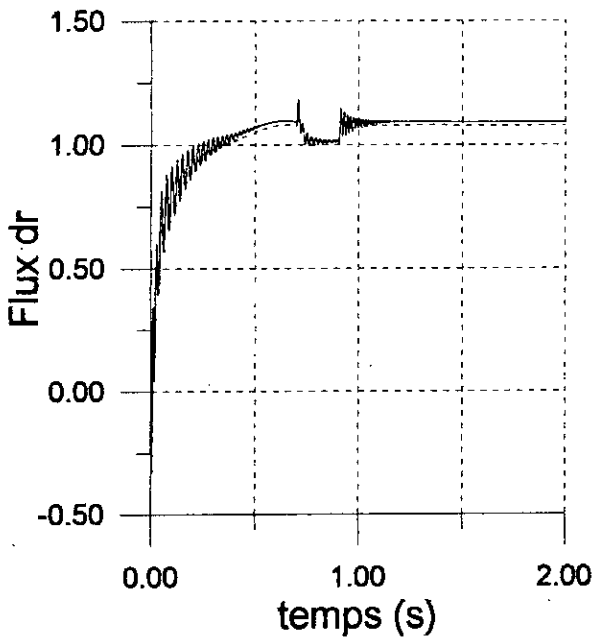
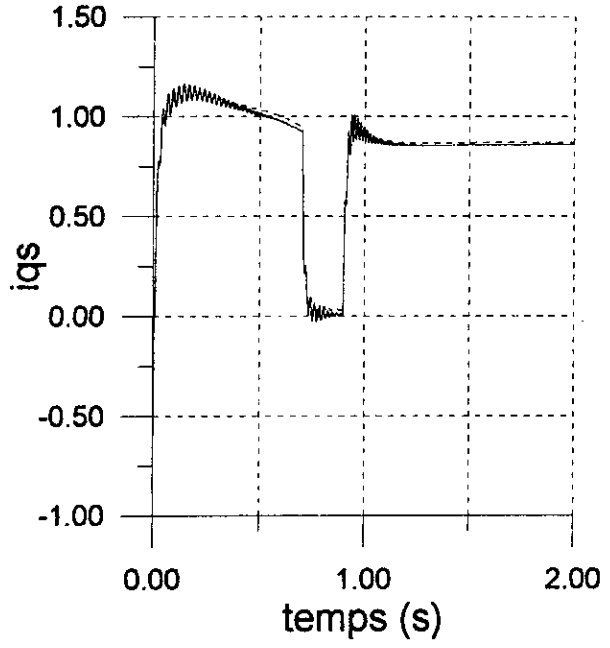
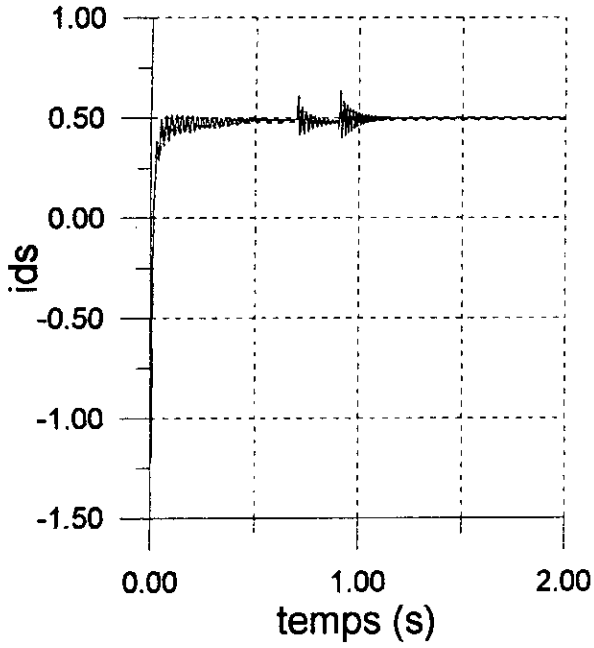


Planche -15- Test de robustesse vis-à-vis  $R_s$

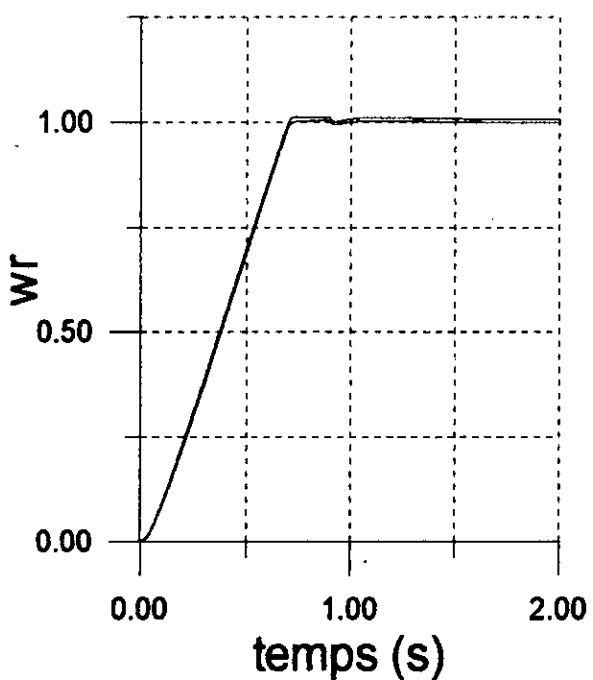
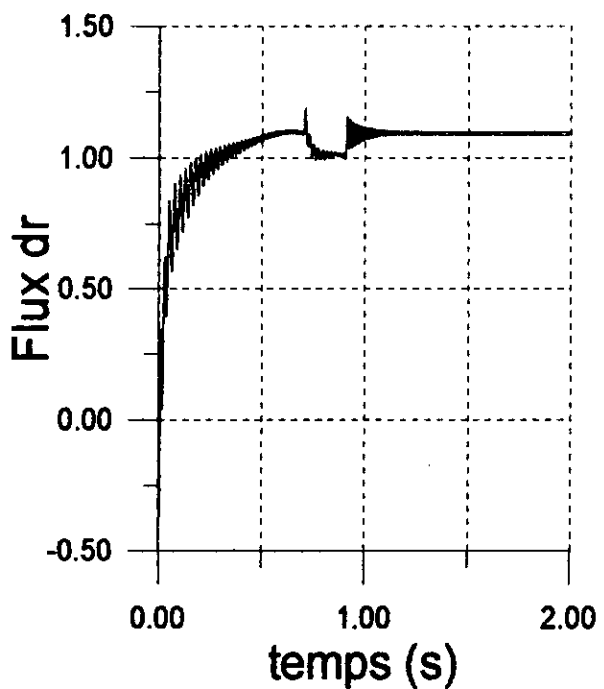
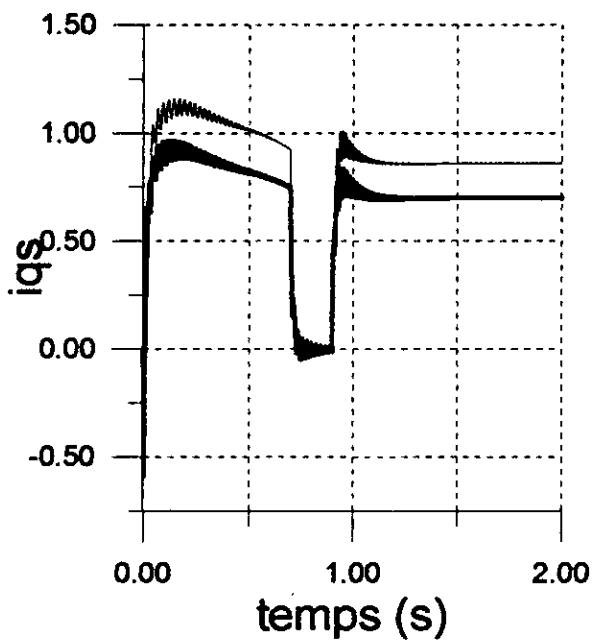
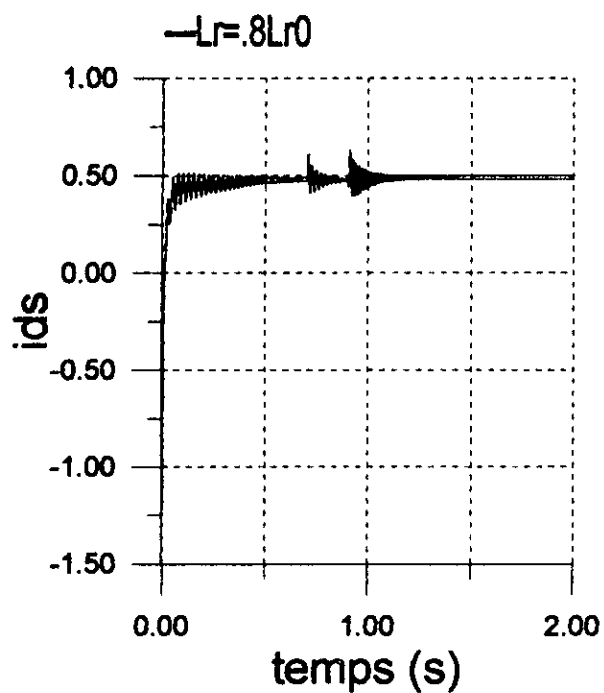


Planche -16-Test de robustesse vis-à-vis  $L_r$

$-R_r = 1.5R_{r0}$

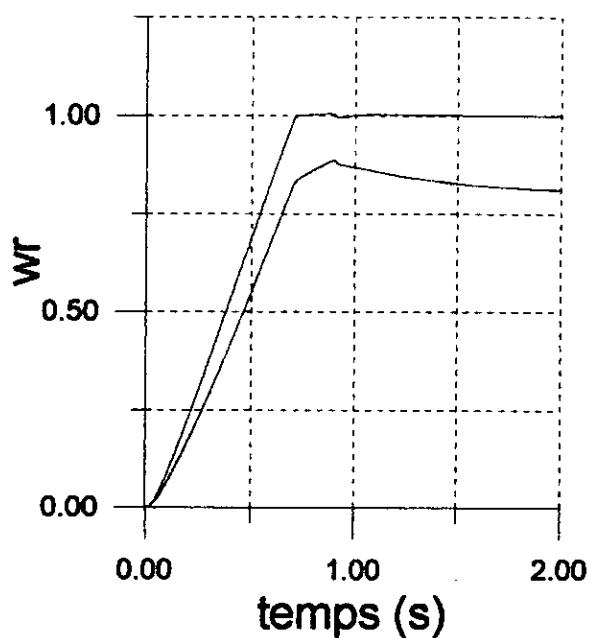
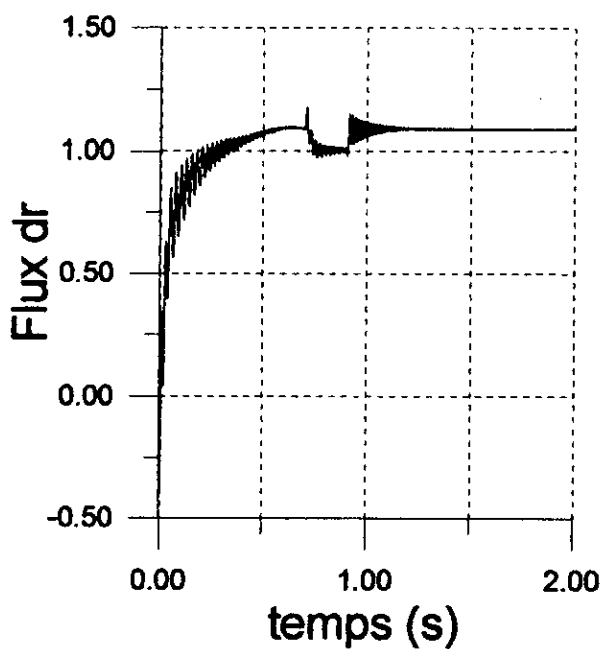
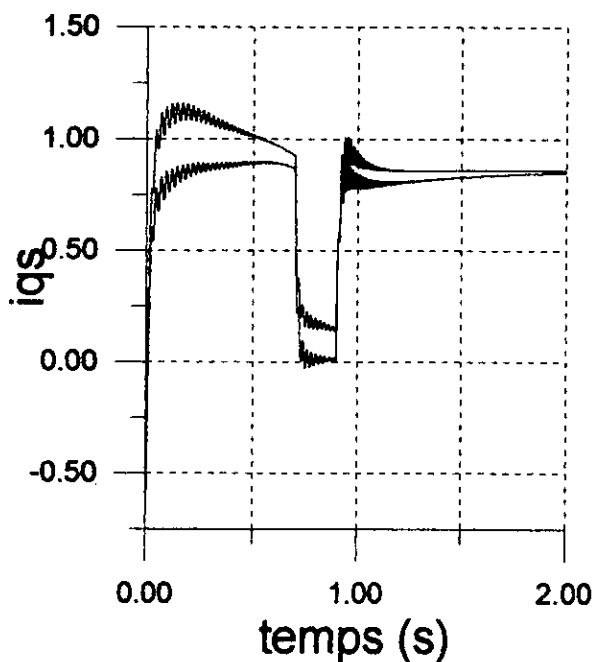
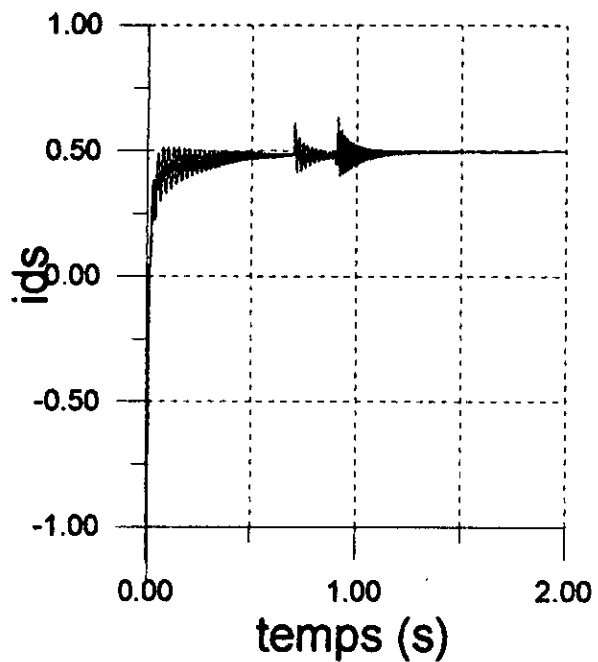


Planche -17-Test de robustesse vis-à-vis  $R_r$



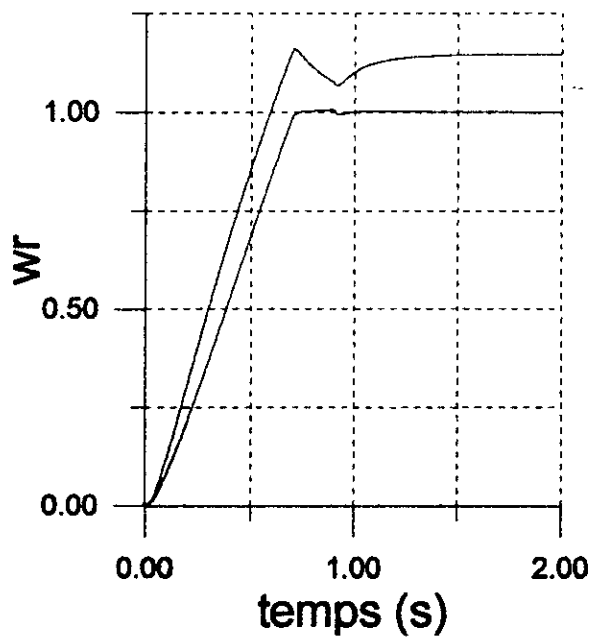
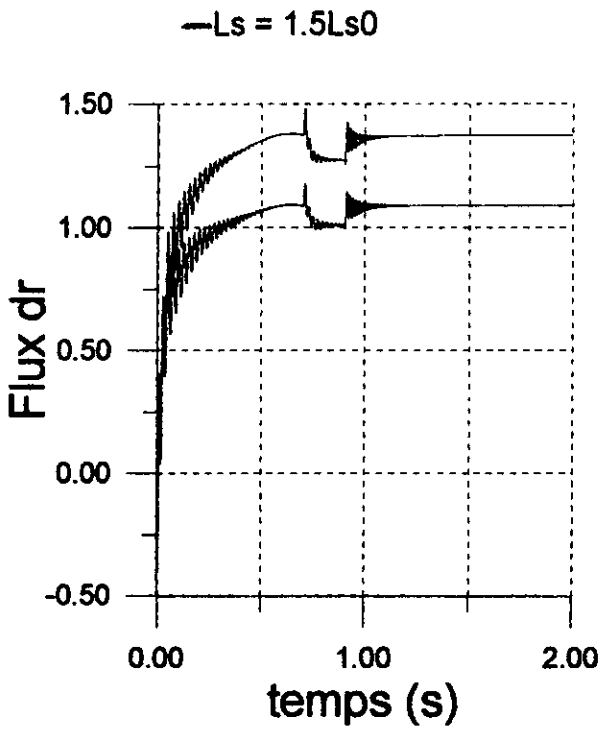
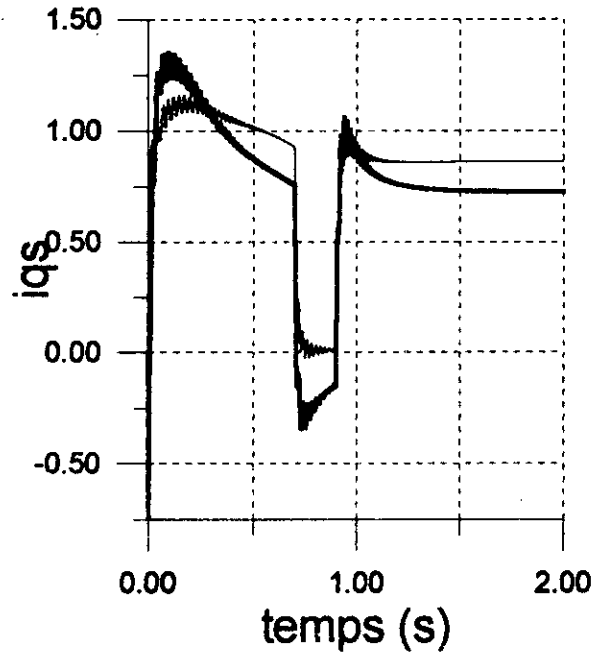
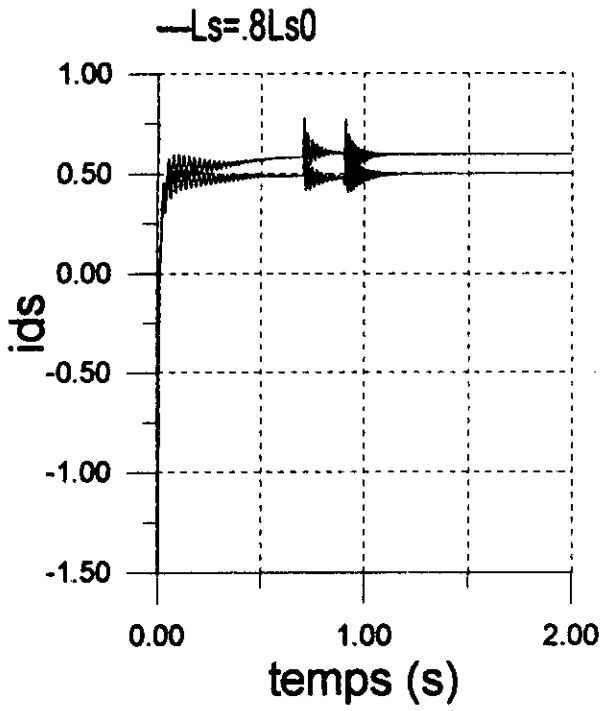


Planche-18- Test de robustesse vis-à-vis  $L_s$

**Conclusion :**

L'application de la commande inverse à la MAS représentée par son modèle réel à orientation du champ rotorique , donne des performances satisfaisantes .

La réponse du flux est moins rapide que dans le cas parfait , le flux  $\psi_{dr}$  a une allure oscillatoire , et le flux  $\psi_{qr}$  oscille autour de zéro .

---

**CONCLUSION**

**GENERALE**

## **CONCLUSION GENERALE**

---

**L**a commande inverse par réseaux de neurones appliquée à la machine asynchrone, à fait l'objet de notre travail.

L'utilisation des réseaux de neurones ,vu leurs propriétés citées dans le chapitre 1 apparaît indispensable pour le cas où le système à commander est non-linéaire, inconnu, ou difficile à formuler son modèle.

La MAS se prête bien à cette application, à cause de son modèle non-linéaire couplé ,le découplage a été réalisé par l'orientation du champ.

La commande par bouclage linéarisant a été appliquée à la MAS représentée par son modèle à orientation parfaite du champ, et a servi par la suite pour la sélection des exemples d'entraînement des réseaux de neurones.

la commande inverse a été appliquée par la suite à ce modèle..

Une structure de commande appropriée a été proposée ,en vu de réaliser la commande inverse, de la MAS représentée dans le repère synchrone ,dans le but de réaliser l'orientation réelle du champ.

Appliquée à la MAS à orientation parfaite du champ ,la commande inverse a donné plein de satisfaction pour la régulation ainsi que pour la poursuite, Les performances de réglage était moins satisfaisantes pour le cas réel.

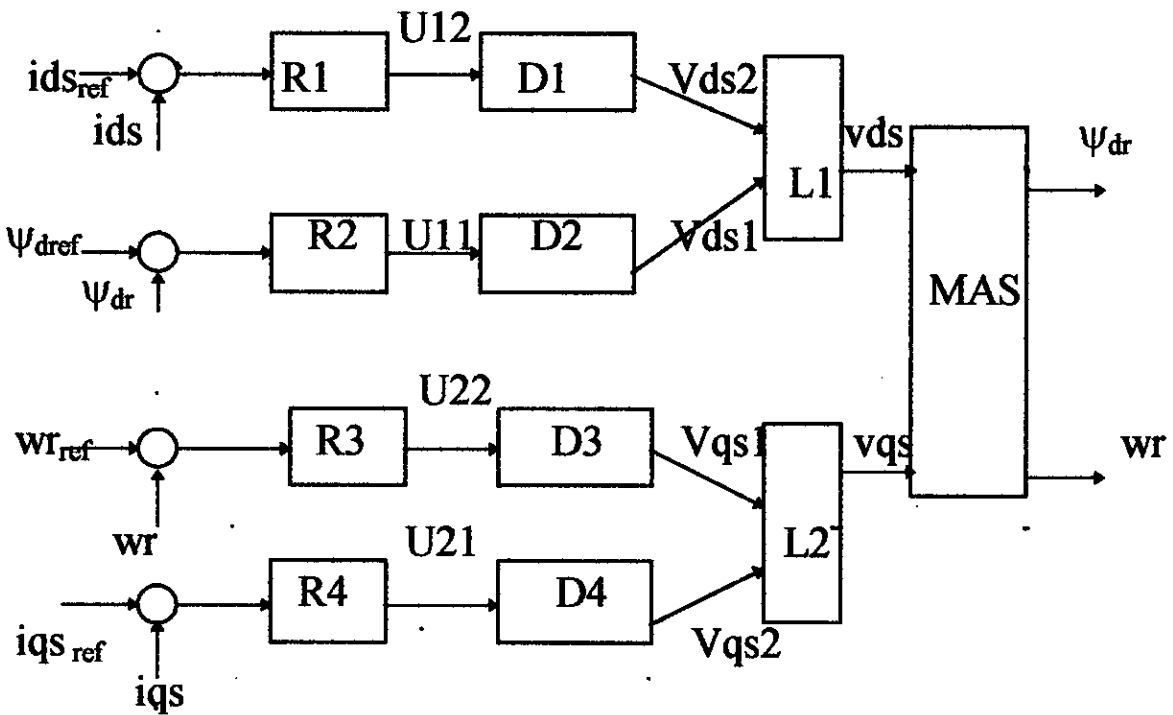
Cependant cette commande est sensible aux variations des paramètres de la machine. Ceci était deviné ,à cause de l'absence d'une chaîne de retour dans la structure de la commande inverse.

On propose de faire un apprentissage spécialisé pour les réseaux, à fin d'éviter la sensibilité aux variations des paramètres ,la commande inverse ainsi réalisée se confond avec la commande adaptative à modèle de référence.

• page 13

dans notre cas un choix de poids synaptiques.

• page 41



# ANNEXE . I

## Annexe 1

---

La machine asynchrone utilisée dans notre travail possède les caractéristiques suivantes :

- Paramètres électriques :

$$R_s = 1.12 \Omega$$

$$R_r = 1.12 \Omega$$

$$L_s = 0.17 \text{ H}$$

$$L_r = 0.015 \text{ H}$$

$$M = 0.048 \text{ H}$$

$$P = 2$$

$$\sigma = 0.096$$

- Paramètres mécaniques :

$$J = 0.135 \text{ Kg.m}^2$$

$$K_f = 0.00182 \text{ N.m.s}$$

$$C_r = 28 \text{ N.m}$$

---

# ANNEXE . II



## ANNEXE 2

---

Le modèle de la Mas à orientation parfaite du champ ainsi établi dans le chapitre 2 est le suivant :

$$\begin{cases} \dot{x}_1(k) = f_1(x_1(k), x_2(k), x_3(k), x_4(k), u_1(k)) \\ \dot{x}_2(k) = f_2(x_1(k), x_2(k), x_3(k), x_4(k), u_2(k)) \\ \dot{x}_3(k) = f_3(x_1(k), x_2(k), x_3(k), x_4(k)) \\ \dot{x}_4(k) = f_4(x_1(k), x_2(k), x_3(k), x_4(k)) \end{cases}$$

tel que :

$$\begin{cases} f_1(x(k)) = a_1 x_1(k) + x_2(k)x_4(k) + b_1 x_3(k) + \frac{d_1 x_1(k)^2}{x_3(k)} + e_1 \cdot u_1(k) \\ f_2(x(k)) = a_2 x_2(k) + x_1(k)x_4(k) + \frac{d_2 x_1(k) \cdot x_2(k)}{x_3(k)} + b_2 x_3 x_4 + e_2 \cdot u_1(k) \\ f_3(x(k)) = a_3 x_1(k) + b_3 x_3(k) \\ f_4(x(k)) = b_4 x_4(k) + a_4 x_2(k)x_3(k) + c_4 \cdot Cr \end{cases}$$

avec :

$$a_1 = a_2 = -\left(\frac{1}{\sigma T_s} + \frac{1-\sigma}{\sigma T_r}\right)$$

$$b_1 = \frac{1-\sigma}{\sigma \cdot M \cdot T_r}$$

$$b_2 = -\frac{1-\sigma}{\sigma \cdot M}$$

$$d_1 = -d_2 = \frac{M}{T_r}$$

$$e_1 = e_2 = \frac{1}{\sigma \cdot L_s}$$

$$a_3 = \frac{M}{T_r}, \quad b_3 = -\frac{1}{T_r}, \quad a_4 = \frac{M p^2}{J L_r}, \quad b_4 = -\frac{K_f}{J}, \quad c_4 = -\frac{p}{J}$$

En utilisant le développement au 2<sup>ème</sup> ordre on obtient :

$$x_1(k+1) = x_1(k) + h f_1(x) + \frac{h^2}{2} f_1'(x)$$

$$x_2(k+1) = x_2(k) + h f_2(x) + \frac{h^2}{2} f_2'(x)$$

$$x_3(k+1) = x_3(k) + h f_3(x) + \frac{h^2}{2} f_3'(x)$$

$$x_4(k+1) = x_4(k) + h f_4(x) + \frac{h^2}{2} f_4'(x)$$


---

## ANNEXE 2

---

Après développement de calcul on trouve :

$$x_1(k+1) = x_1 + (h+a1\frac{h^2}{2})f_1 + (x_4 + 2d1\frac{x_2}{x_3})f_2 - 2d1\frac{x_2}{x_3}f_3 + x_2f_4 + \frac{e1}{h}(u_1 - u_1(k-1))$$

$$x_2(k+1) = x_2(k) + \frac{h^2}{2}(x_4 + d2\frac{x_2}{x_3})f_1(x) + (h + \frac{h^2}{2}(a2 + d2\frac{x_1}{x_2}))f_2 + \frac{h^2}{2}(b2x_4 - \frac{x_1x_4}{x_3})f_3 + \frac{h^2}{2}(x_1 + b2x_3)f_4 + e2\frac{h^2}{2}(u_2(k) - u_2(k-1)) \quad (C)$$

$$x_3(k+1) = x_3 + a3\frac{h^2}{2}f_1 + (h + b3\frac{h^2}{2})f_3$$

$$x_4(k+1) = x_4(4) = a4\frac{h^2}{2}x_3f_2 + a4\frac{h^2}{2}x_2f_3 + (h + b4\frac{h^2}{2})f_4 + c4\frac{h^2}{2}Cr$$

On doit déterminer  $x_1(k) = f(x_3, x_4, u_1)$  et  $x_2(k) = g(x_3, x_4, u_2)$ , on trouve :

$$x_1(k) = \alpha 3 x_3(k+1) - \gamma 3 x_4 x_2 - \beta 3 \frac{x_2^2}{x_3} - \lambda 3 u_1 + \varphi 3 x_3$$

avec :

$$\alpha 3 = \frac{1}{\frac{h^2}{2}a1 \cdot a3 + a3(h + \frac{h^2}{2}b3)}, \quad \gamma 3 = \frac{h^2 a3 \cdot d1}{\frac{h^2}{2}a1 \cdot a3 + a3(h + \frac{h^2}{2}b3)}$$

$$\beta 3 = \frac{\frac{h^2}{2}a3}{\frac{h^2}{2}a1 \cdot a3 + a3(h + \frac{h^2}{2}b3)}, \quad \varphi 3 = \frac{1 + b3(h + \frac{h^2}{2}b3)}{\frac{h^2}{2}a1 \cdot a3 + a3(h + \frac{h^2}{2}b3)}$$

$$\lambda 3 = \frac{\frac{h^2}{2}a3 \cdot e1}{\frac{h^2}{2}a1 \cdot a3 + a3(h + \frac{h^2}{2}b3)}$$

$$x_4(k+1) = \alpha 4 x_2 x_1 + \beta 4 x_3 x_4 x_1 + \gamma 4 x_3 x_2 + \lambda 4 x_4 x_3 + \psi 4 x_4 + \rho 4 Cr + \eta 4 \cdot x_3 u_2 \quad (D)$$


---

## ANNEXE 2

avec :

$$\alpha 4 = \frac{h^2}{2} a4(a3+d2), \quad \beta 4 = \frac{h^2}{2} a4, \quad \gamma 4 = \left( \frac{h^2}{2} a2 \cdot a4 + \frac{h^2}{2} b3 \cdot a4 + \left( h + \frac{h^2 b4}{2} \right) \right)$$

$$\lambda 4 = \frac{h^2}{2} b2, \quad \psi 4 = \left( 1 + b4 \cdot \left( h + \frac{h^2}{2} b4 \right) \right), \quad \varphi 4 = \left( h + \frac{h^2}{2} b4 \right) \cdot c4$$

$$\eta 4 = \frac{h^2}{2} a4 e_2$$

Il en est de même pour  $x_2$  :

$$\alpha 1 \frac{x_2^3}{x_3} + \beta 1 \cdot x_2^2 + \gamma 1 \cdot x_2 + \varphi 1 = 0 \quad (A)$$

$$\begin{aligned} \alpha 1 &= \alpha 4 \cdot \beta 3 & \beta 1 &= (\alpha 4 \cdot \gamma 3 \cdot x_4 + \beta 4 \cdot \beta 3 \cdot x_4) \\ \gamma 1 &= \beta 4 \cdot \gamma 3 \cdot x_3 \cdot x_4^2 - \alpha 4 \cdot \alpha 3 \cdot x_3 \cdot (k+1) - \alpha 4 \cdot \varphi 3 \cdot x_3 - \gamma 4 \cdot x_2 + \alpha 4 \cdot \lambda 3 \cdot u_1 \\ \varphi 1 &= x_4 \cdot (k+1) - \beta 4 \cdot \alpha 3 \cdot x_3 \cdot x_4 \cdot x_3 \cdot (k+1) - \lambda 4 \cdot x_3 - \varphi 4 \cdot x_4 - \varphi 4 \cdot Cr \\ &+ \beta 4 \cdot \lambda 3 \cdot x_3 \cdot x_4 \cdot u_1 - \eta 4 \cdot x_3 \cdot u_2 \end{aligned}$$

La résolution de l'équation (A) par rapport à  $x_2(k)$  donne :

$$x_2(k) = f_2^*(x_3(k+1), x_3(k), x_4(k+1), x_4(k), u_1(k), u_2(k))$$

$$x_1(k) = f_1^*(x_3(k+1), x_3(k), x_4(k+1), x_4(k), u_2(k), u_2(k))$$

Or  $x_3(k+1)$  est connu, on détermine  $x_3(k+2)$  en utilisant les équations du (C) :

$$\begin{aligned} x_3(k+2) &= f_3^*(x_3(k+1), x_3(k), x_4(k+1), x_4(k), u_1(k), u_2(k-1), u_2(k), u_2(k-1)) \\ &+ \lambda 3 u_1(k+1) \end{aligned}$$

De la relation (D), on écrit  $x_4(k+2)$  et de même que pour  $x_3(k+1)$  on trouve :

$$\begin{aligned} x_4(k+2) &= f_4^*(x_3(k+1), x_3(k), x_4(k+1), x_4(k), u_1(k), u_1(k-1), u_2(k), u_2(k-1)) \\ &+ \eta 4 \cdot x_3(k+1) \cdot u_2(k+1) \end{aligned}$$

Ou bien :

$$\begin{aligned} x_3(k+1) &= f_3^*(x_3(k), x_3(k-1), x_4(k), x_4(k-1), u_1(k-1), u_1(k-2), u_2(k-1), u_2(k-2)) \\ &+ \lambda 3 \cdot u_1(k) \end{aligned}$$

## ANNEXE 2

---

$$x_4(k+1) = f_4^*(x_3(k), x_3(k-1), x_4(k), x_4(k-1), u_1(k-1), u_1(k-2), u_2(k-1), u_2(k-2)) \\ + \eta_4 \cdot x_3(k) \cdot u_2(k)$$

avec :

$$\lambda_3 = \frac{h^2}{2} a_3 \cdot e_1$$

$$\eta_4 = \frac{h^2}{2} a_4 \cdot e_2$$

# BIBLIOGRAPHIE

## **BIBLIOGRAPHIE**

---

- [1] L. Barazane ; " Commande Vectorielle d'un moteur Asynchrone Alimenté en courant " Thèse de Magister,1993.
  - [2] P. Barret ; " Régime Transitoire Des Machines Tournantes Electriques " ; Ed Eyrolles,1982.
  - [3] J. Chatelain ; " Machines Electriques " ; tome 1 ,Ed Dunod , 1983.
  - [4] H .Demuth ,M .Beale ; "Neural Network Toolbox :user's guide" ;The MathWorks Inc, Juin,1992 .
  - [5] J.A.Freeman, D.M.Skapura ; "Neural Networks application and programming techniques" ;Ed Adison Wesley,Houston,1990.
  - [6] T.Fukuda and T.Shibata ; "theory and application of Neural Networks for industrial control systems" ;IEEE Trans ;On Industrial Electronic ;39,473,489,1992.
  - [7] K.Funahashi, "On the approximate réalisation by Neural Network " ; 2,183,192,1989.
  - [8] B.Hamzi ,S.Labiod ; "Identification et commande des systèmes dynamiques par réseaux de neurones " ;PFE ,ENP ,1995.
  - [9] J.Hérault ; "Réseaux neuronaux et traitement du signal" ;Ed Hermès,Paris,1994.
  - [10] K.Hunt, J.Sbarbaro ; "Neurol Networks for control systems -A servey\*" Automatica , 28,1083,112,1992.
  - [11] J.F.Jodouim ; "Les Réseaux de Neurones , Principes et définitions" ;Ed Hermès,
  - [12] A.Karakasoglu ; "Identification and Decentralized Adaptive Control Using Dynamical Neural Networks with Application to Robotic Manipulators" ;IEEE Tran,4,6,1993.
  - [13] N.B.Karayiannis and A.N.Venetsapoulov ; "Artificial Neural Networks ,Learning Algorithms,performances, Evaluation and Application" ;Klauer AcademicsPress,1993.
  - [14] S.Y.Kung and J.N.Hwang ;"Neural Network Architectures for Robotic Applications" ;IEEE Trans, On Robotics and Automation,5,641-657,1989.
-

## **BIBLIOGRAPHIE**

---

- [15] Y .Le Cun ; "A Thoretical Framework for Back-Propagation"  
Toronto,Canada,1988.
  - [16] M.Mahmoudi ; " Modélisation de la MAS En Vue de sa commande "  
Article, ENP.
  - [17] K.S.Narendra and Parthasarathy, "Identification and Control of Dynamical  
systems Using Neural Networks" ;IEEE Trans, On Neural Networks ;1,4-27,1990.
  - [18] M. Nibouche ;" Application de la Commande Non Linéaires Au Moteur  
Asynchrone " Thèse de Magister,1994.
  - [19] S-Z-Qin,H-T-Su ;" Comparison of Four Neural Net Learning Methods for  
Dynamic System Identification" ;IEEE,3-1,122-127,1992
  - [20] E.Slotine ; "Neural Network Control of Unknown Nonlinear  
Systems" ;Cambridge,USA ,
  - [21] P.J.Werbos ; "Back-Propagation Through Time, What it does and How to  
do it ? " ;Proc of IEEE, 78,1550-1560.
  - [22] M.Weinfeld ; "Réseaux de neurones" ;Techniques de l'ingénieur, traité  
Informatique,Doc H1, 1990.
-