



République Algérienne Démocratique et Populaire
Ministère de l'Enseignement Supérieure et la
Recherche Scientifique



Université Echahid Hamma Lakhdar d'El-Oued

FACULTE DE TECHNOLOGIE

DEPARTMENT DE GENIE

MECANIQUE

Mémoire De Fin d'étude

Présenté pour l'obtention du diplôme de

MASTER ACADEMIQUE

Domaine : Science Technologie

Filière : Electromécanique

Spécialité : Electromécanique

Thème :

**IDENTIFICATION ET COMMANDE PAR
LES RESEAUX DE NEURONES D'UNE
PARABOLE DE RADAR DE POURSUITE**

Présenté par :

SEBAA Mokim

LADJIMI Mohammed

HAFIAN Ahmed Tidjani

CHELLIG Abdelali

Dirigé par :

Djokhrab Ala Eddine

Meneceur Redha

2022-2023

REMERCIEMENTS

Avant toute chose, on remercie Dieu le tout puissant de nous avoir donné le courage, la patience et la force durant toutes ces années d'études.

Nous sommes très reconnaissants à Monsieur Djokhrab Ala Eddine et à Monsieur Meneceur Redha et nos remerciements pour nous avoir encadré et en particulier pour son soutien, sa gentillesse, ses encouragements, ses conseils et sa patience pendant la période de la réalisation de ce travail

Nos remerciements vont aussi aux enseignants du département de génie mécanique de la spécialité Electromécanique qui ont contribué à notre formation.

Un remerciement spécial aux étudiants de la 2ème année Master de la spécialité Electromécanique .

Nous remercions vivement toutes les personnes que ont contribué de près ou de loin , à la réalisation de ce travail .

SOMMAIRE

INTRODUCTION GENERALE.....	01
----------------------------	----

Chapitre 01

Les réseaux de neurones

I.1. INTRODUCTION	2
I.2. HISTORIQUE	2
I.3. ELEMENT DE BASE	3
I.3.1. Modélisation biologique	3
I.3.2. Structure de neurones	3
I.3.2.1. Le corps cellulaire	3
I.3.2.2. Les dendrites	3
I.3.2.3. L'axone	4
I.3.2.4. Les synapses	4
I.4. FONCTIONNEMENT DE NEURONE	4
I.5. RESEAUX DE NEURONE ARTIFICIAL	5
I.4.1. Modélisation	5
I.5.1.1 Nature des entrées	6
I.5.1.2. La fonction d'entrée totale	6
I.5.3. La fonction d'activation	6
I.5.4. La fonction de sortie	8
I.5.5. Les automates booléens	8
I.5.6. Les automates à seuil	8
I.5.7. Les automates linéaires	8
I.5.1.8. Les automates à saturation	8
I.5.1.9. Les automates continus	8
I.5.1.10. Les automates probabilistes	8
I.6. PROPRIETES DES RESEAUX DE NEURONES	9
I.6.1. Le parallélisme	9
I.6.2. La résistance aux pannes	9

I.6.3. La capacité d'adaptation	9
I.6.4. La généralisation	9
I.6.5. Structure de connexion	9
I.7. TOPOLOGIE DES RESEAUX DE NEURONES	9
I.7.1. Les réseaux statiques ou réseau à couche (FEED FORWARD)	9
I.7.2. Les réseaux dynamiques (récurrents)	11
I.8. LE PERCEPTRON	12
I.8.1. Rétine	12
I.8.2. Couche de cellules d'association	12
I.8.3. Couche de cellule de décision	13
I.9. PERCEPTRON SIMPLE	13
I.10. L'APPRENTISSAGE DES RESEAUX DE NEURONES	14
I.10.1. Apprentissage supervisé	15
I.10.2. Apprentissage non supervisé	15
I.11. LES PRINCIPALES REGLES D'APPRONTISSAGE	15
I.11.1. Les mécanismes de classification	17
I.11.2. Les mémoires associatives	17
I.11.3. Mémoires auto associatives et hétéro associatives	18
I.12. ALGORITHME DE RETRO PROPAGATION	18
I.12.1. Introduction	18
I.12.2. Présentation	19
I.12.3. Position de problème	19
I.12.4. Modèle et équation du réseau	19
I.12.5. Principe de rétro propagation	20
I.12.6. Adaptation des poids	20
I.12.7. Les 'étapes d'algorithme de rétro propagation	22
I.12.8. Algorithme de rétro propagation	23
I.13. DOMAINES D'APPLICATION DES RESEAUX DE NEURONES	24
I.14. LES CARACTERISTIQUES D'UNE BONNE APPLICATION	25
I.15. CONCLUSION	26

Chapitre 02

Identification et commande par les RNA

II.1.1 Introduction	27
II.2. Identification par les réseaux de neurones	27
II.2.1. Modèle directe	28

II.2.1.1. Identification série parallèle	29
II.2.1.2. Identification parallèle	30
II.2.2. Modèle inverse	31
II.2.2.1. Apprentissage généralisé	31
II.2.2.2. Apprentissage spécialisé	33
II.3. Commande par les réseaux de neurones	34
II.3.1. Les étapes de la conception d'un organe de commande	35
II.3.2. Principe de la commande	35
II.3.3. Les structures de commande	35
II.3.3.1. Commande par modèle inverse	36
II.3.3.1.1. Contrôleur Feed-forward	36
II.3.3.1.2. Contrôleur Feed-back	37
II.3.3.2. Commande par modèle de référence	38
II.3.3.2.1. Structure de commande direct	38
II.3.3.2.2. Structure de commande indirect	39
II.4 Conclusion	40

Chapitre 03

Application les techniques de commande neuronal sur la parabole de radar de poursuite

III.1. INTRODUCTION	41
III.2. Fonctionnement du système	41
III.2.1 Analyse et performance du système asservi	43
III.2.2 Fonction de transfert	44
III.3. Application le RNA sur la parabole de radar de poursuite	46
III.3.1 Identification du modèle direct	46
III.3.2. Identification du modèle inverse	52
III.3.3. Commande par modèle inverse	56
III.3.4. Commande par modèle de référence	58
III.4 Conclusion	60
CONCLUSION GENERALE	61
Bibliographie	
Résume	

Liste des figures

Chapitre I

- Figure (I.1) Le neurone biologique
- Figure (I.2) Schéma d'un neurone formel
- Figure (I.3) Les différentes formes de la fonction d'activation
- Figure (I.4) Forme d'un réseau non bouclé
- Figure (I.5) Forme d'un réseau bouclé
- Figure (I.6) Le modèle de Kohonen
- Figure (I.7) Réseau de Hopfield
- Figure (I.8) Schéma d'un Perceptron
- Figure (I.9) Schéma de principe de l'Adaline
- Figure (I.10) Illustration de l'apprentissage supervisé
- Figure (I.11) Illustration de l'apprentissage non supervisé
- Figure (I.12) Architecture de réseaux multi couche
- Figure (I.13) Courbe explicative du phénomène du minimum local
- Figure (I.14) Evolution du N_{bre} de compagnies proposant des produits connexionnistes

Chapitre II

- Figure (II.1) Structure d'identification série-parallèle
- Figure (II.2) Structure d'identification parallèle
- Figure (II.3) Structure d'apprentissage généralisé
- Figure (II.4) Structure d'apprentissage spécialisé
- Figure (II.5) Structure de contrôle par modèle inverse (feed-forward)
- Figure (II.6) Structure de contrôle par modèle inverse (feed-back)
- Figure (II.7) Principe de l'apprentissage d'une commande par modèle de référence
- Figure (II.8) Structure de contrôle direct
- Figure (II.9) Structure de contrôle indirect

Chapitre III

- Figure (III.1) Définitions des grandeurs
- Figure (III.2) Organisation matérielle
- Figure (III.3) La dynamique de réducteur
- Figure (III.4) Schéma bloc en boucle ouverte
- Figure (III.5) Schéma bloc en boucle fermée
- Figure (III.6) Entrée du système
- Figure (III.7) Sortie du système
- Figure (III.8) Identification du modèle direct
- Figure (III.9) Erreur entre le modèle et le système
- Figure (III.10) Test 1 du modèle direct
- Figure (III.11) Erreur
- Figure (III.12) Test 2 du modèle direct
- Figure (III.13) Erreur
- Figure (III.14) Identification du modèle inverse

Figure (III.15) Erreur
Figure (III.16) Test 1 du modèle inverse
Figure (III.17) Erreur
Figure (III.18) Test 2 du modèle inverse
Figure (III.19) Erreur
Figure (III.20) Sortie du système pour une consigne échelon
Figure (III.21) Sortie du système pour une consigne rampe
Figure (III.22) Sortie du système pour une consigne
Figure (III.23) Sortie du système pour une consigne échelon
Figure (III.24) Sortie du système pour une consigne rampe
Figure (III.35) Sortie du système pour une consigne



**INTRODUCTION
GENERALE**

INTRODUCTION GENERALE:

La théorie de contrôle fournit des outils d'analyse et de synthèse parfaitement adaptés aux systèmes linéaires. Cependant en pratique, ces méthodes ne s'avèrent pas applicables à cause des non linéarités des systèmes réels et parce qu'il n'est pas toujours possible de linéariser le système à commander. Il existe des méthodes générales pour le traitement des systèmes non linéaires, des nouvelles méthodes qui doivent prendre en compte les caractéristiques particulières de tels systèmes s'avèrent nécessaires.

Les propriétés d'approximation, d'apprentissage et de traitement parallèle des réseaux de neurones peuvent fournir des solutions plus efficaces aux problèmes de contrôle des systèmes non linéaires. Les possibilités d'approximation permettent sans doute une meilleure modélisation de sa linéarité, la possibilité d'apprentissage peut réduire l'effort humain lors de la conception des contrôles et permet de découvrir des structures de contrôle plus efficaces que celles déjà connues. La propriété de traitement parallèle fournit au contrôleur à base de réseaux de neurones l'habilité d'une réponse rapide aux variations complexes de l'environnement.

En générale, les réseaux de neurones ont des applications puissantes dans tous les niveaux d'une structure de commande hiérarchisée qui permet au système d'avoir un degré élevé d'autonomie.

L'objectif du présent travail est d'apporter les bases nécessaires pour envisager l'exploitation des réseaux de neurones dans l'identification et la commande d'un système qu'est la parabole de radar de poursuite. Ce mémoire contient trois chapitres organisés comme suit :

Le premier intitulé les réseaux de neurones artificiels, dans ce chapitre on donne un aperçu général sur les réseaux de neurones ainsi que leur algorithme d'apprentissage et un exposé sur la méthode de rétropropagation et en fine quelque application.

Le second est l'identification et commande par les réseaux de neurones, ce chapitre trait les méthodes de modélisation de la dynamique direct et inverse des systèmes non linéaires

Le troisième présente la parabole de radar de poursuite qui est l'objet de la commande et les résultats de simulation.

En fin, une conclusion générale qui exprime le bilan de ce travail.

CHAPITRE I :

Les réseaux de neurones

I.1. INTRODUCTION :

Les réseaux de neurones formels sont des structures la plupart du temps simulés par des algorithmes exécutés sur les ordinateurs d'usage général, parfois sur des machines ou même des circuits spécialisés, qui prennent leurs inspirations à partir du fonctionnement élémentaire des systèmes nerveux. Ils sont utilisés essentiellement pour résoudre des problèmes de classification, de reconnaissance de formes, d'association, d'extraction de caractéristique, et d'identification,... etc.

I.2. HISTORIQUE:

C'est en 1943 que, dans un article resté fameux, MC CULLOCH et PITTS ont émis l'idée simplificatrice du neurone formel, c'est-à-dire une opération binaire interconnectée à ses semblables par des « synapses » excitatrices ou inhibitrices. Une assemblée de tels opérateurs en interaction devait avoir des propriétés collectives émergentes, c'est-à-dire capables de certain « calcul » que chacun d'eux séparément est capable d'exécuter.

En 1949, D.O. HEBB, dans un livre resté lui aussi fameux introduisit la notion de « plasticité synaptique », c'est-à-dire le mécanisme de modification progressive des couplages inter neurone responsables de changement permanent de leurs propriétés collectives, ce que l'on peut appeler « l'apprentissage ».

Son hypothèse, limitée à l'augmentation du coefficient de couplage entre deux neurones réels qui sont excités simultanément, a été étendue aux neurones artificiels comme une règle par laquelle les couplages se modifient proportionnellement aux corrélations entre neurones, que ces corrélations soient positives « activation » ou négatives « inhibition ».

Les travaux sur les réseaux de neurones formels ont connu des débuts prometteurs vers la fin des années 50. C'est en 1958 que ROSENBLATT décrit le premier modèle opérationnel de réseaux de neurones, mettant en œuvre les idées de HEBB, MC CULLOCH et PITTS.

En 1969, MINSKY et PEPART, démontrèrent les limites théoriques du perceptron, qui est capable d'apprendre à calculer certaines fonctions logiques, mais un certain manque d'approfondissement de la théorie et le mouvement concurrent de l'intelligence artificielle, ont conduit à une mise sommeil assez prolongée des travaux sur le connexionnisme.

Assez peu d'articles furent publiés entre 1970 et 1980, avec un peu de ralentissement en raison du faible nombre de chercheurs intéressés. Ils ont cependant permis une certaine continuité et surtout ont servi de germes à une renaissance frappante de l'activité dans le début des années 80. S'il faut citer des travaux de cette période, c'est celui de HOPFIELD en 1982, qui détaille dans un article le bon usage du réseaux de neurones formels, en s'appuyant sur l'organisation en couches et la simulation numérique sur ordinateur et ceux de WIDROW et HOFF sur les algorithmes adaptatifs.

En 1986, Rumelhart, Hinton et Williams publient, l'algorithme de la rétropropagation de l'erreur qui permet d'optimiser les paramètres d'un réseau de neurones à plusieurs couches.

À partir de ce moment, la recherche sur les réseaux de neurones connaît un essor fulgurant et les applications commerciales de ce succès académique suivent au cours des années 90.

Actuellement, les chercheurs s'intéressent beaucoup plus à l'exploitation des structures d'identification et de contrôle mono et multivariable de la commande adaptative avec les réseaux de neurones [1].

I.3. ELEMENTS DE BASE :

I.3.1. Modélisation biologique :

Le cerveau humain, est le meilleur modèle de la machine, polyvalente incroyablement rapide et surtout douée d'une incomparable capacité d'auto organisation. Son comportement est beaucoup plus mystérieux que le comportement de ses cellules de base. Il est constitué d'un grand nombre d'unités biologiques élémentaires (1000 à 10000 synapse par neurone)[2].

Les cellules nerveuses, appelées neurones, sont les éléments de base du système nerveux central. Celui-ci en posséderait environ cent milliards. Les neurones possèdent de nombreux points communs dans son organisation générale et leur système biologique avec les autres cellules. Ils présentent cependant des caractéristiques qui leur sont propres et se retrouvent au niveau de cinq fonctions spécialisées qu'ils assurent [2] [3].

- Recevoir des signaux en provenance de neurones voisins.
- Intégrer ces signaux.
- Engendrer un influx nerveux.
- Le conduire.
- Le transmettre à un autre neurone capable de le recevoir.

I.3.2. Structure de neurone :

Un neurone est constitué de trois parties :

I.3.2.1. Le corps cellulaire :

Il contient le noyau du neurone et effectue les transformations biochimiques nécessaires à la synthèse des enzymes et des autres molécules qui assurent la vie de neurone. Sa forme est pyramidale ou sphérique dans la plupart des cas. Elle dépend souvent de sa position dans le cerveau, ce corps cellulaire fait quelques microns de diamètre [4].

I.3.2.2. Les dendrites :

Chaque neurone possède une « chevelure » de dendrites. Celles-ci sont de fines extensions tubulaires, de quelques dixièmes de microns de diamètre et d'une longueur de

quelques dizaines de microns. Elles sont les récepteurs principaux du neurone qui servent à capter les signaux qui lui parviennent [4].

I.3.2.3. L'axone :

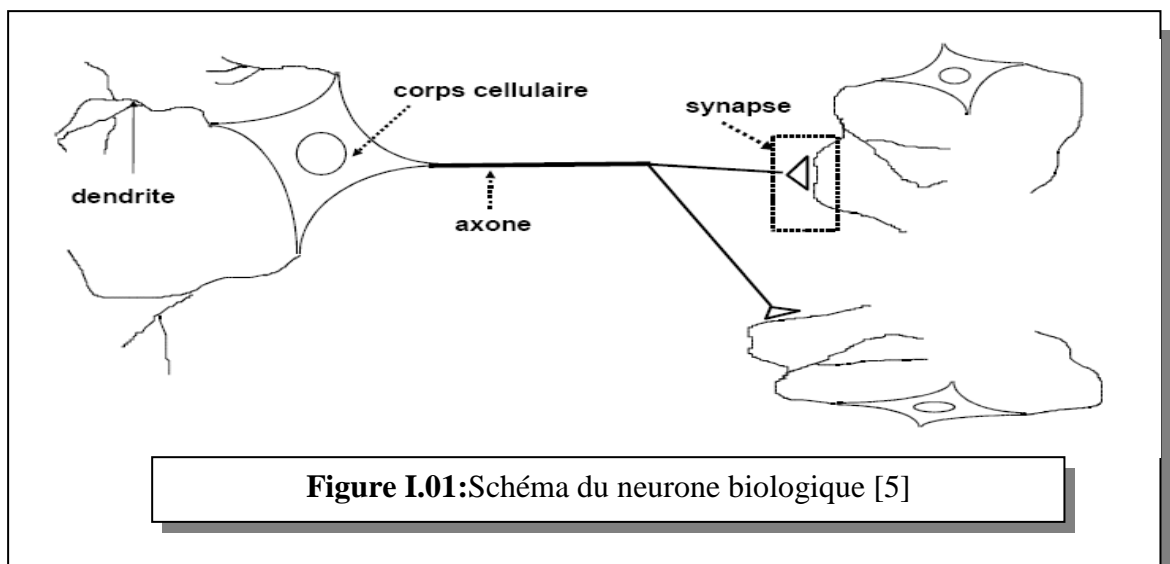
L'axone, qui est à proprement parler la fibre nerveuse, sert de moyen de transport pour les signaux émis par le neurone. Il se distingue des dendrites par sa forme et par les propriétés de sa membrane externe. En effet, il est généralement plus long que les dendrites, et se ramifie à son extrémité, là où il communique avec les autres neurones, alors que les ramifications des dendrites se produisent plutôt près du corps cellulaire.

Pour former le système nerveux, les neurones sont connectés les uns aux autres suivant des répartitions spatiales complexes. Les connexions entre deux neurones se font à des endroits appelés synapses ou ils sont séparés par un petit espace synaptique de l'ordre d'un centième de micron [4].

I.3.2.4. Les synapses :

Le rôle des synapses est fondamental pour permettre aux cellules nerveuses de communiquer entre elles. Les signaux qui se propagent dans les neurones sont de nature électrique. Cependant, il n'existe pas de liaisons directes entre deux cellules nerveuses. Celles-ci sont séparées par un espace appelé fente synaptique que l'influx électrique ne peut traverser.

Le relais s'effectue à ce niveau par l'intermédiaire d'un médiateur chimique [4].



I.4. FONCTIONNEMENT DES NEURONES :

D'une façon générale, le soma de neurone traite les courants électriques (information) qui lui proviennent de ses dendrites et qu'il transmet le courant électrique résultant de ce traitement aux neurones auxquelles il est connecté par l'intermédiaire de son axone.

D'après le modèle classique, le soma effectuant une sommation des influx nerveux transmise par les dendrites, Si la sommation dépasse un seuil, le neurone répond par un influx nerveux au potentiel d'action qui se propage le long de son axone. Si la sommation est inférieure à ce seuil, le neurone reste inactif.

Lorsqu'un potentiel d'action est parvenu au synapse, provoque à travers la membrane de libération d'un médiateur chimique celle-ci se diffuse jusqu'à la membrane du dendrite qui provoque naissance un potentiel [6].

I.5. RESEAU DE NEURONE ARTIFICIEL :

L'idée générale des réseaux de neurones artificiels et de trouver une nouvelle procédure de calcul pour produire une nouvelle génération d'ordinateur. Cette idée, inspirée de la biologie du cerveau humain, est différente de la méthode conventionnelle utilisée sur les ordinateurs.

Cette nouvelle technique semble plus humaine que mécanique. L'homme à toujours rêver de la machine qu'est parle, pense, oublie et se rappelle. Le but n'est pas de produire une machine humaine ce que impossible, mais de copier et imiter l'approche naturelle de cerveau humaine au profite de la science. Un RNA peut être considéré comme une boîte noire, qui reçoit des signaux d'entrée et produit des signaux de sortie c'est un modèle mathématique composé d'un grand nombre d'éléments de calculs organisée sous forme de couches interconnectées [7].

D'autre définition sont donnés comme suite :

Les RNA sont des réseaux fortement connectés de processeurs élémentaires fonctionnant en parallèle. Chaque processeur calcule une sortie unique sur la base des informations qu'il reçoit. Toute structure hiérarchique de réseaux est évidemment un réseau [8].

I.5.1. Modélisation :

La première modélisation d'un neurone date des années quarante. Elle a été présentée par Mac CULLOCH et PITTS. S'inspirant de leurs travaux sur les neurones biologiques, ils ont proposé le modèle suivant : un neurone formel fait une somme pondérée des potentiels d'action qui lui parviennent (chacun de ces potentiels est une valeur numérique qui représente l'état du neurone qui la émis), puis s'active suivant la valeur de cette sommation pondérée. Si cette somme dépasse une certain seuil, le neurone est activé et transmet une réponse (sous forme de potentiel d'action) dont la valeur est celle de son activation, si le neurone n'est pas activé il ne transmet rien.

La fonction de transfert est un simple effet de seuil. Alors, si une cellule « i » est reliée à « N » autres cellules, le neurone formel donné par la figure (02), effectue une somme pondérée des « N » entrées, augmentée d'une entrée externe désignant le seuil de neurone « i ».

Le tout est suivi d'un élément de décision qui représente la fonction d'activation non linéaire qui détermine la sortie du neurone. La fonction de décision peut avoir les formes les plus utilisées données par la figure (03) [9].

Si on désigne par « X » le vecteur d'entrées du neurone « i », « W_i » représente la force de la connexion simulant les poids synaptiques des neurones, tels que :

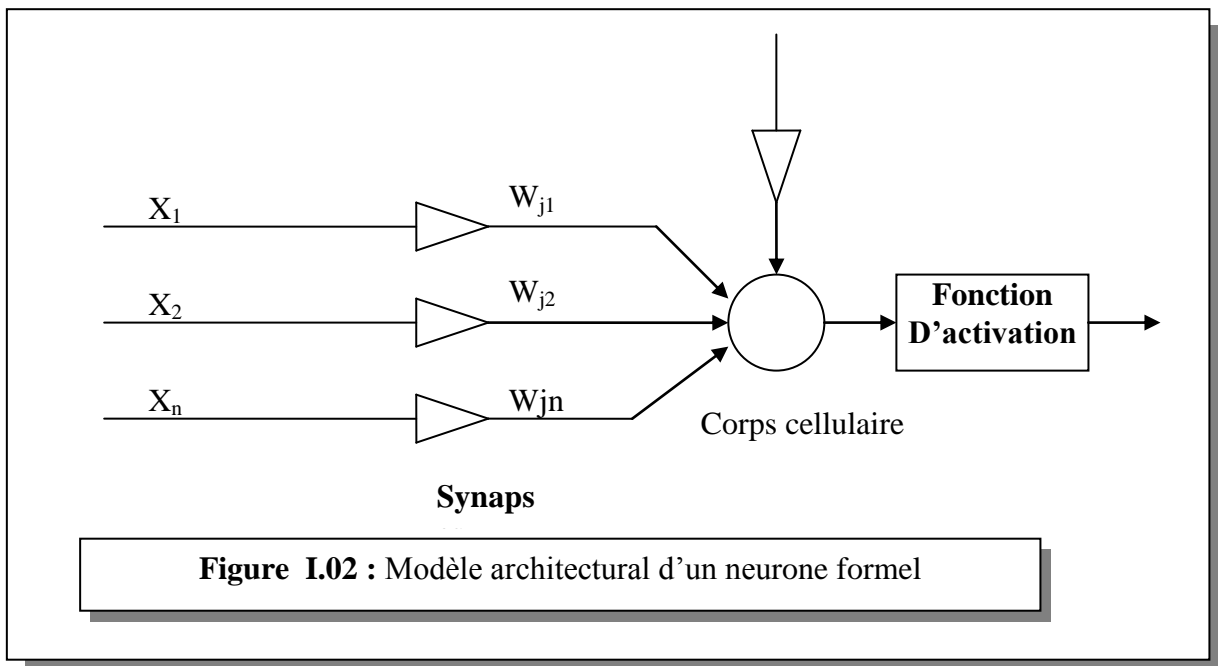
$$X=[X_1, X_2 \dots\dots, X_n]$$

$$W_j=[W_{j1}, W_{j2} \dots\dots W_{jn}]$$

La sortie intermédiaire « S_i » est donnée par : $S_i = W_i * (X) - \theta_i$ (I.1)

Et la sortie de neurone « O_i » est donnée par : $O_i = f(s_i)$. (I.2)

Où les « θ_i » sont les seuils et « f » est la fonction d'activation (ou de décision).



En général, un réseau de neurone artificiel peut être défini par les éléments suivants :

I.5.1.1. Nature des entrées :

Elle peuvent être soit binaires (-1, +1) ou (0, 1), ou bien réelles.

I.5.1.2. La fonction d'entrée totale :

Elle définit le prétraitement effectué sur les entrées.

I.5.1.3. La fonction d'activation :

La fonction de transfert est en général, une fonction non linéaire monotone croissante ; Par ailleurs, les fonctions de transfert sont de qualités diverses : elles peuvent être déterministes, continues, discontinues ou aléatoires [10]. La figure03 donne les modèles de fonctions d'activation utilisées.

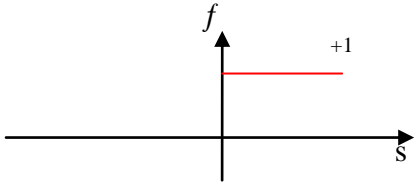
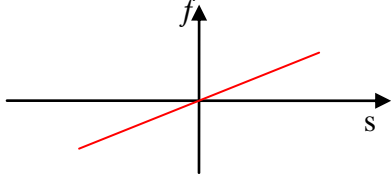
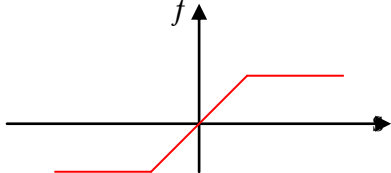
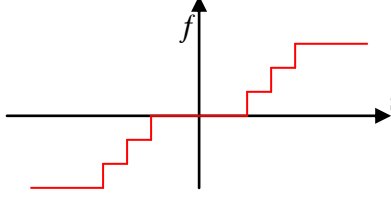
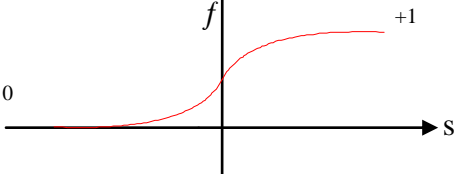
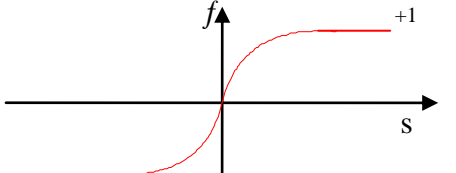
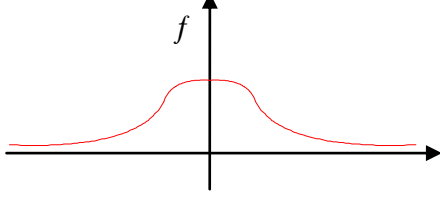
<p>Fonction de Heaviside</p>	
<p>Fonction linéaire sans saturation</p>	
<p>Fonction linéaire avec seuil</p>	
<p>Fonction a seuils multiples</p>	
<p>Fonction sigmoïde (1)</p> $f_1(s) = \frac{1}{1 + e^{-s}}$	
<p>Fonction sigmoïde (2)</p> $f_2(s) = \frac{1 - e^{-s}}{1 + e^{-s}}$	
<p>Fonction de stochastique</p>	

Figure I.03 : Les modèles de fonctions d'activation.

Le choix de la fonction d'activation dépend de l'application. S'il faut avoir des sorties binaires c'est la première fonction que l'on choisit habituellement. Une entrée spéciale est pratiquement toujours introduite pour chaque neurone. Cette entrée, normalement appelée biais, sert pour déplacer le pas de la fonction d'activation sur l'axe S.

I.5.1.4. La fonction de sortie :

Elle calcule la sortie de neurone en fonction de son état d'activation, en général cette fonction est considérée comme la fonction identité, par la suite confondrons toujours activation et sortie du neurone.

Les fonctions d'entrée et d'activation sont choisies en fonction des types d'automates utilisés, c'est-à-dire tout dépend de la nature des entrées et des sorties. En général, ils sont au nombre de six [11]:

I.5.1.5. Les automates booléens :

- Les entrées et les sorties sont booléennes.
- La sortie est une fonction booléenne des entrées.

I.5.1.6. Les automates à seuil :

- La sortie est binaire alors, que les entrées peuvent être binaires ou réelles.
- La fonction d'entrée totale est affine.
- La fonction d'activation est la fonction signe ou la fonction de heaviside.

I.5.1.7. Les automates linéaires :

- Les entrées et les sorties sont réelles.
- La fonction d'entrée totale est linéaire.
- La fonction de sortie est l'identité.

I.5.1.8. Les automates à saturation :

- Les entrées et les sorties sont comprises dans un intervalle $[u, v]$.
- La fonction d'entrée totale est linéaire.
- La fonction d'activation est la fonction saturée.

Si les entrées et les sorties sont entières, on obtient un automate multi seuils.

I.5.1.9. Les automates continus :

- Les entrées et les sorties sont réelles.
- La fonction d'entrées est linéaire ou affine.
- La fonction d'activation est stochastique.

I.5.1.10. Les automates probabilistes :

- Les sorties sont binaires et les entrées sont quelconques.
- La fonction d'entrée est affine ou linéaire.
- La fonction d'activation est stochastique.

I.6. PROPRIETES DES RESEAUX DE NEURONES :

Un réseau de neurone est un ensemble d'éléments de traitement de l'information, avec une topologie spécifique d'interconnexions entre ces éléments et une loi d'apprentissage pour adapter les points de connexions. D'une manière générale, un réseau de neurones possède les propriétés suivantes [6]:

I.6.1. Le parallélisme :

Cette notion se situe à la base de l'architecture des réseaux de neurones considérés comme ensembles d'entrées élémentaires qui travaillent simultanément.

I.6.2. La résistance aux pannes :

A cause de l'abondance des entrées et la structure du réseau. Les données bruitées ou les pannes locales dans un certain nombre de ses éléments n'affectent pas ses fonctionnalités. Cette propriété résulte de fonctionnement collectif et simultané des neurones qui les composent.

I.6.3. La capacité d'adaptation :

Celle-ci se manifeste tout d'abord dans les réseaux de neurones par la capacité d'apprentissage qui permet au réseau de tenir compte des nouvelles contraintes ou des nouvelles données du monde extérieur. De plus, ils se caractérisent par leur capacité d'auto organisation qui assure leur stabilité en tant que système dynamique.

I.6.4. La généralisation :

La capacité de généralisation d'un réseau de neurone est son aptitude de donner une réponse satisfaisante à une entrée qui ne fait pas partie des exemples à partir desquels il a appris

I.6.5. Structure de connexion :

Les connexions entre les neurones qui composent le réseau décrivent la "topologie" du modèle. Elles sont très variées, le nombre de connexions étant énorme. Cette topologie fait apparaître une certaine régularité de l'arrangement des neurones.

I.7. TOPOLOGIE DES RESEAUX DE NEURONES :

On peut classer les réseaux de neurones en deux grandes catégories, selon la dépendance de l'évolution de ceux-ci en fonction explicite du temps.

I.7.1. Les réseaux statiques ou réseau à couche (FEED FORWARD) :

C'est le cas de réseaux statiques, ou le temps n'est pas un paramètre significatif. En d'autres termes, la modification d'entrée n'entraîne que la modification stable de la sortie, mais elle n'entraîne pas le retour de l'information de cette entrée.

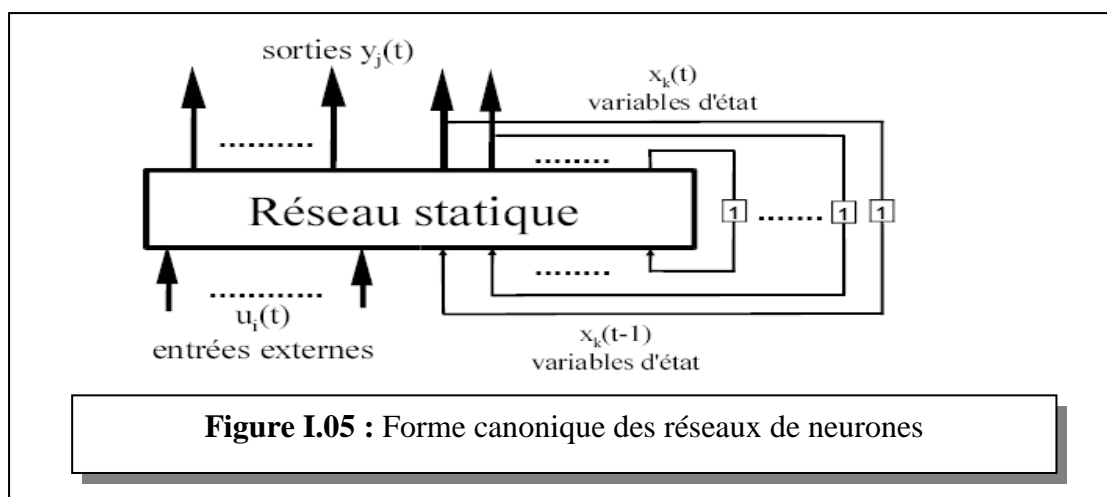
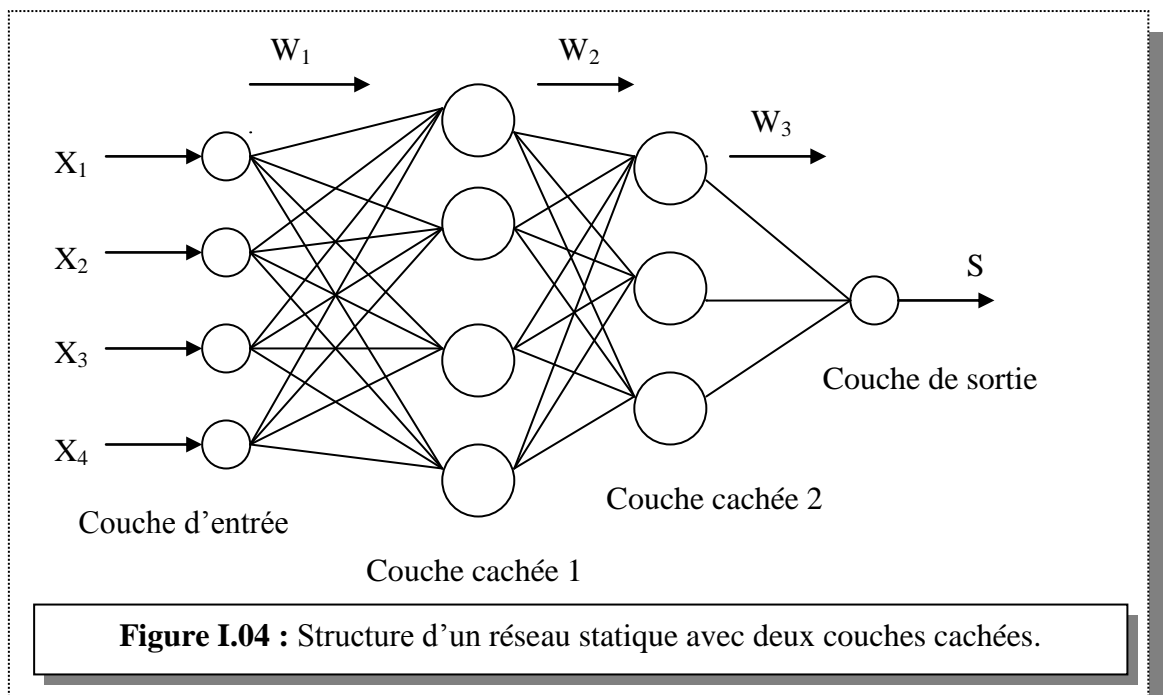
Les réseaux statiques (FEED FORWARD) sont des réseaux à couches, constitués d'une couche d'entrée, une couche de sortie et entre les deux au moins une couche composée de nombreux éléments de traitements non linéaires, appelée couche cachée.

Les signaux des entrées se propagent de la première couche à la couche de sortie en passant par les couches cachées, Il n'y a pas des communications entre les unités de la même couche, d'où le nom de feedforward. Les liens dirigés connectant les neurones sont appelés les interconnexions [1].

On distingue des réseaux à deux couches tel que le perceptron et l'adaline (adaptive linear neurone) qui sont caractérisés par :

- la simplicité de réglage d'apprentissage.
- la facilité de détermination de l'influence d'un neurone d'entrée sur l'erreur d'un neurone de sortie d'en déduire les modifications à apporter au lien qui les relie.
- La limitation au calcul de fonction très simple.

Ces réseaux ne pouvaient résoudre que des problèmes simples de classification. Pour des problèmes complexes, une solution consiste à organiser le réseau en plusieurs couches[12].



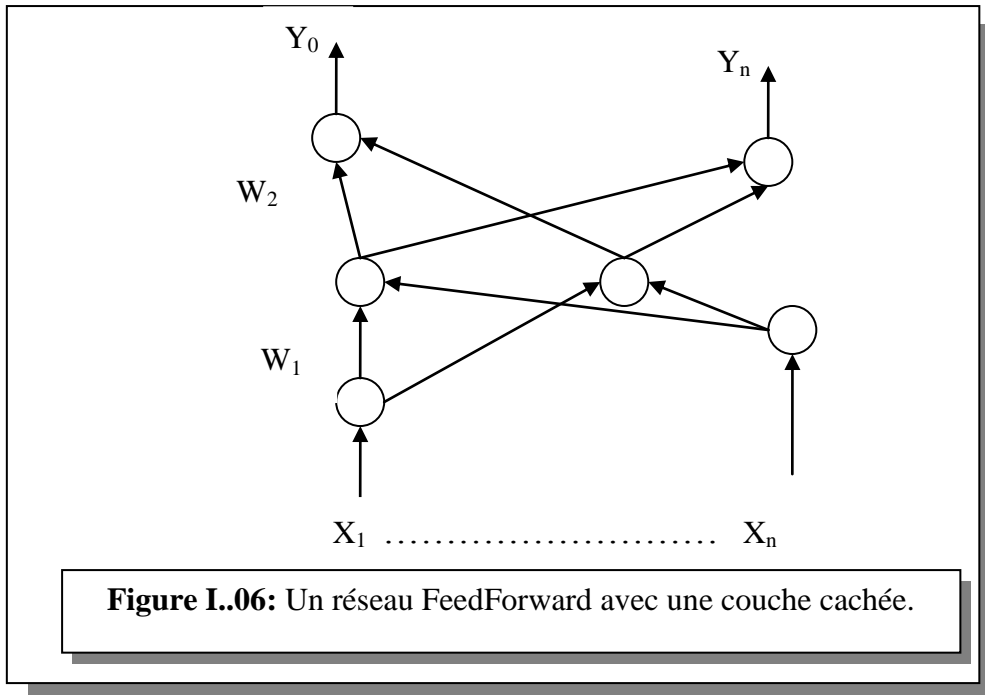


Figure I.06: Un réseau FeedForward avec une couche cachée.

I.7.2. Les réseaux dynamiques (récurrents):

Comme leurs noms l'indique, contiennent des rebouclages partiels ou totaux entre neurones, ils représentent donc une évolution dépendante du temps.

Il faut bien distinguer la dépendance théorique, pour laquelle l'état du réseau à un certain instant dépend de son état à l'instant ou aux instants précédents, du temps nécessaire à obtenir une réponse, dans le cas d'une réalisation matérielle ou d'une simulation sur ordinateur.

La théorie de ces réseaux fut publiée par HOPFIELD dans les années 1980 et sont souvent appelées « **HOPFIELD nets** ». Ils sont principalement utilisés pour l'implémentation des mémoires associatives « BAM : bidirectional associative memory »[1].

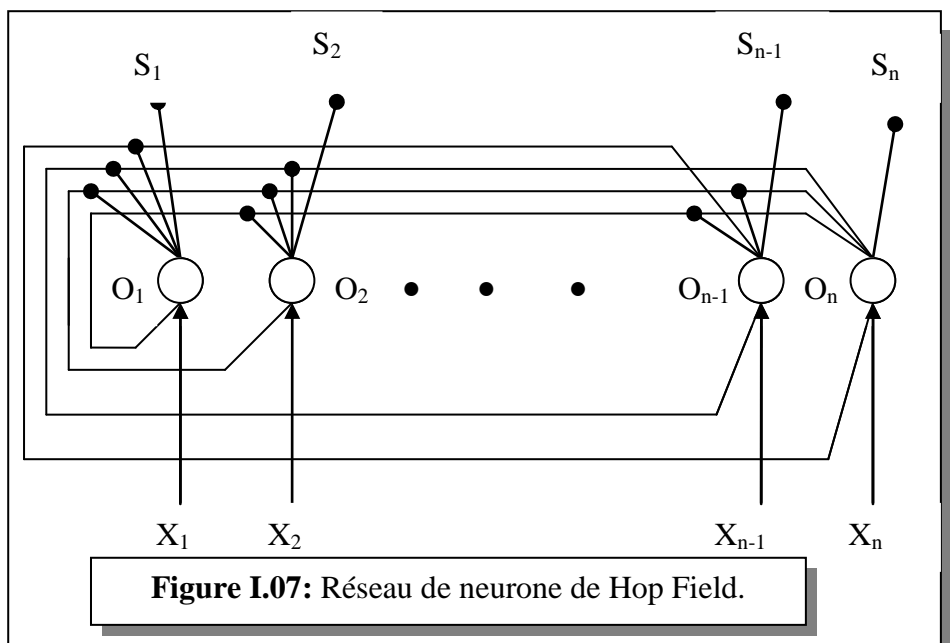


Figure I.07: Réseau de neurone de Hop Field.

Le perceptron multicouches ordinaire ou la carte auto organisatrice sont des réseaux statiques. Par contre, le réseau du HOPFIELD ou le perceptron avec rebouclage est des réseaux dynamiques.

Les critères motivant les choix d'un type de réseau sont la simplicité de mise en œuvre et l'efficacité des algorithmes d'adaptation appelés à répondre aux performances désirées du système, quelle que soit sa complexité [1].

L'opérateur non linéaire réalisé par un réseau, bouclé ou non, dépend des valeurs des coefficients de pondération du réseau.

Pour qu'un réseau effectue une tâche donnée, il faut donc ajuster la valeur des ses coefficients. Une tâche est définie par un ensemble d'exemples, ou couples (valeurs des entrées et valeurs des sorties désirées correspondantes), tels les couples (forme classe) en classification, ou les couples (commande sortie mesurée du processus) en modélisation ces couples constituent l'ensemble d'apprentissage.

I.8. LE PERCEPTRON :

A la fin des années **50**, la neurobiologie ne disposait pas encore des techniques modernes de l'électrochimie pour étudier les structures du cerveau. L'idée de certains chercheurs fut alors d'utiliser les modélisations des neurones et de l'évolution des synapses pour simuler des réseaux de neurones. Ceux-ci pourraient exhiber des comportements proches de ceux du cerveau et ainsi fournir des explications sur le fonctionnement de celui-ci.

Le premier modèle fut présenté par ROSEBLATT c'était le perceptron ce modèle a pour propriétés [7] :

- D'être spécifié en termes suffisamment précis pour permettre le test des performances annoncées.
- D'être suffisamment complexe pour que l'on puisse espérer des comportements intéressants.
- D'être suffisamment simple pour que ses performances puissent être prédites et analysées.
- Enfin et surtout d'être en accord avec les faits biologiques

Un perceptron comprend trois éléments principaux :

I.8.1. Rétine :

Elle est constituée de cellules, sur lesquelles s'inscrit le stimulus.

I.8.2. Couche de cellules d'association :

Chacune de ces cellules peut être connectée à des cellules de la rétine, et à d'autres cellules de décision.

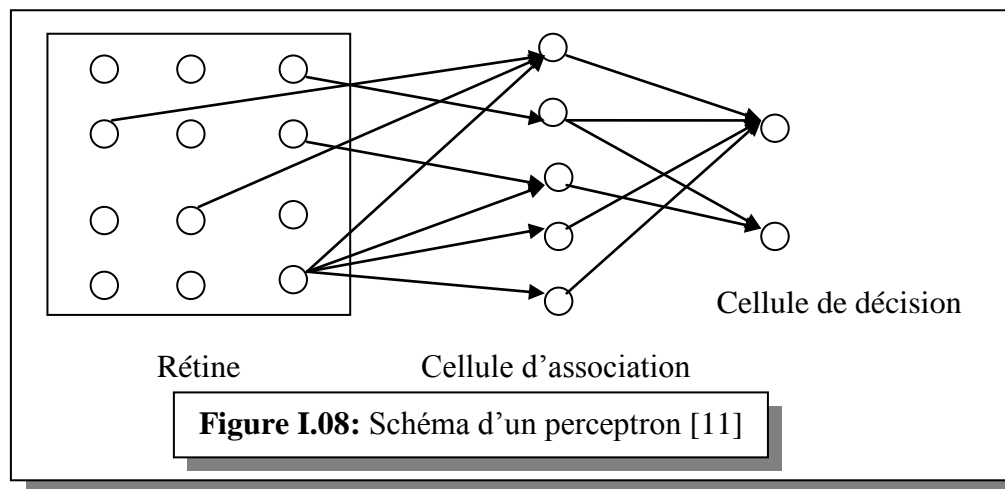
Elle fait la somme des impulsions qui lui viennent et des cellules aux quelles elle est connectée.

I.8.3. Couche de cellule de décision :

Ces cellules fonctionnent comme les cellules d'association, elles reçoivent leurs entrées des cellules d'association ou d'autres cellules de décision. Elles représentent la sortie du perceptron. Le sens des connexions entre les cellules d'associations et les cellules de décisions est bi directionnel, ce qui un FEED BACK de la sortie sur le réseau.

Remarque :

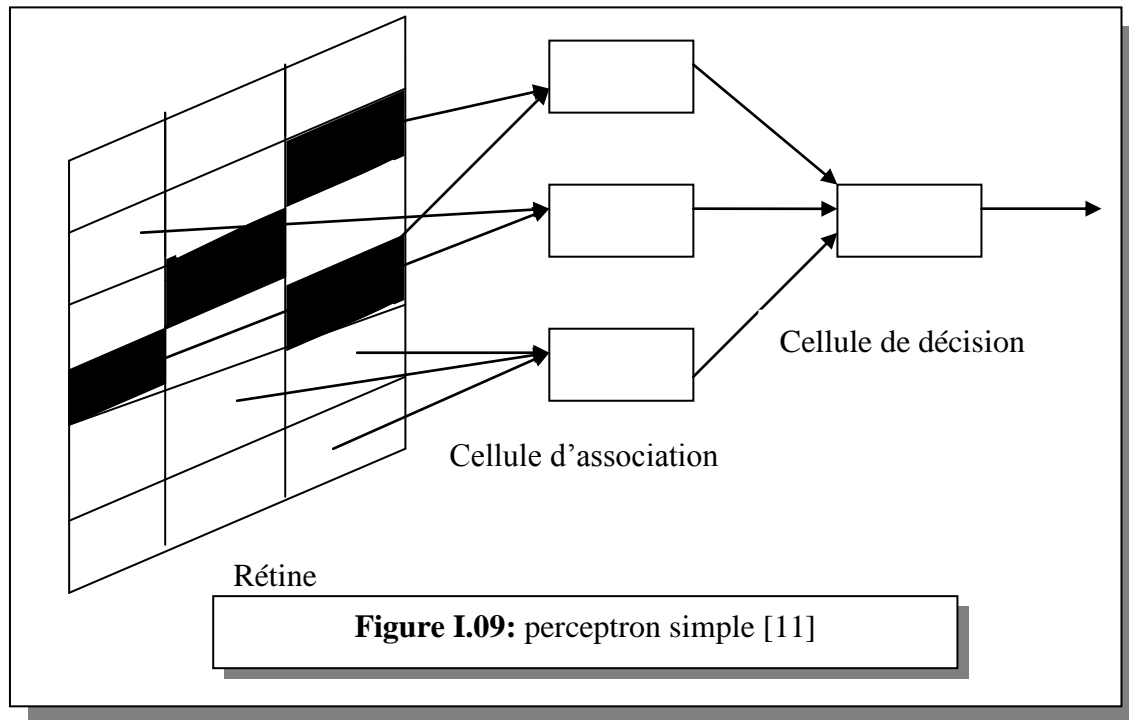
Les neurones constituant le perceptron sont des réseaux linéaires à seuil.



I.9. PERCEPTRON SIMPLE :

Pour faciliter l'étude des comportements du perceptron, il est courant d'utiliser un perceptron d'une forme plus simple que celle définie par le modèle générale. Dans ce cas :

- Les valeurs de sorties de toutes les cellules (rétine, association, décision) sont binaires en 0 et 1.
- Les fonctions réalisées par les cellules d'associations sont des fonctions booléennes et les cellules de décision sont des automates à seuil.
- Il n'existe pas de connexion entre deux cellules d'association, ni entre deux cellules de décisions.
- Les connexions entre cellules d'association et cellules de décision sont à sens unique association décision.



Un tel perceptron va être utilisé pour essayer de reconnaître des formes qui lui seront présentées sur sa rétine. Si l'on n'utilise qu'une cellule de décision, celle-ci devra répondre 1 lorsque la forme qui lui sera présentée correspond à une forme qu'on veut lui faire reconnaître et 0 dans tous les autres cas.

L'apprentissage du perceptron va se faire suivant le perceptron de HEBB, c'est-à-dire en modifiant ses connexions [7].

I.10. L'APPRENTISSAGE DES RESEAUX DE NEURONES :

L'apprentissage se conduit avec une méthodologie précise. La première, est la plus simple, connue sous le nom de règle de HEBB, date de 1949. Elle indique que lorsque deux neurones sont excités en même temps, il faut modifier les coefficients synaptiques pour renforcer cette excitation simultanée. La règle de WIDROW HOFF, établie en 1960, indique que la modification de ses coefficients est proportionnelle à l'erreur entre le résultat souhaité et le résultat réel et aux valeurs d'entrée. La phase d'apprentissage consiste à modifier les poids jusqu'à obtention d'une stabilisation du réseau. C'est-à-dire, jusqu'à ce que le poids ne se modifie plus que d'une façon minime. L'apprentissage mathématique, basé sur ce concept, sert à minimiser une fonction de coûts formulée autour de l'erreur de sortie. Alors l'adaptation commence, par les neurones de la couche de sortie, forcés de la bonne valeur, puis on fait varier légèrement les poids des neurones des couches précédentes :

Il y a autant d'algorithmes d'apprentissage que de types de réseau. Mais la question qui se pose, est ce que on est absolument sûr de la stabilité du réseau ? Question classique qui revient à assurer la convergence des algorithmes d'apprentissage utilisés.

Les procédures d'apprentissage peuvent se subdiviser, elles aussi, en deux grandes catégories [1].

I.10.1. Apprentissage supervisé :

Dans ce cas, la connaissance a priori de la sortie désirée est nécessaire. On présente au réseau le vecteur d'entrée puis on calcule sa sortie, cette dernière sera comparée avec la sortie désirée, ensuite les poids sont ajustés de façon à réduire l'écart entre elles. Cette procédure est répétée jusqu'à ce qu'un critère de performance soit satisfait. L'apprentissage supervisé détermine les poids synaptiques à partir d'exemples étiquetés de formes auquel un professeur a associé des réponses ou des cibles désirées également étiquetées et grâce à une stratégie spécifique. Après la phase d'apprentissage et la convergence des calculs, il n'est plus nécessaire, en général, de mémoriser le jeu d'apprentissage. Ces algorithmes sont particulièrement dédiés aux problèmes de classification des données bruitées ou incomplètes [1].

I.10.2. Apprentissage non supervisé :

Ne nécessite pas «d'enseignant» externe et se base uniquement sur les valeurs d'entrée sans fournir de réponse désirée. Le réseau s'auto organise de façon à optimiser une fonction de coût [13].

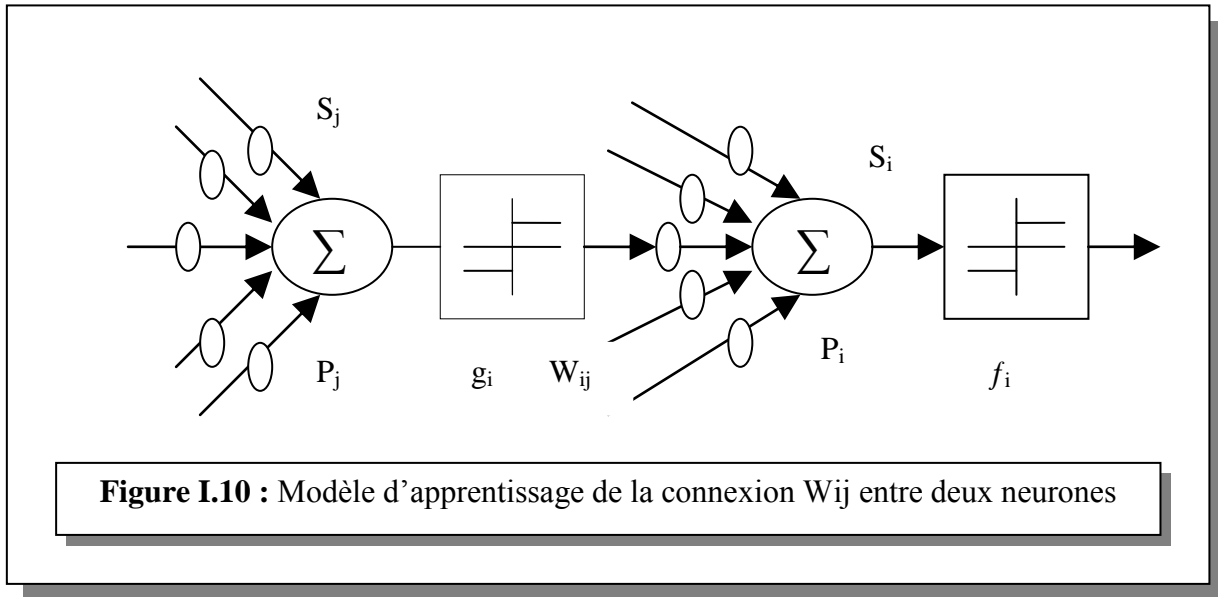
L'apprentissage non supervisé copie le fonctionnement du cerveau humain qui retrouve les informations par association. On présente à l'entrée du réseau des exemples connus et le réseau s'organise lui-même autour d'attracteurs qui correspondent à des configurations stables du modèle dynamique non linéaire associé au réseau. L'apprentissage est accompli à l'aide de règles qui changent ou adaptent le poids des coefficients synaptiques en fonction des exemples présentés à l'entrée et dans certains cas en fonction des sorties désirées.

I.11. LES PRINCIPALES REGLES D'APPRENTISSAGE :

Les stratégies des modifications des poids synaptiques sont dérivées des règles générales suivantes :

- La règle de HEBB ou les poids des connexions entre deux processeurs élémentaires est renforcé si les deux processeurs élémentaires sont activés simultanément
- La règle de WIDROW HOFF ou règle delta règle ou les poids synaptique est adaptée pour obtenir la diminution de l'erreur entre la sortie réelle du processeur élémentaire et la sortie désirée.
- L'apprentissage compétitif dans lequel tous les processeurs élémentaires sont en compétition. La sélection est réalisée à l'aide d'un processus de compétition basé sur une mesure de similarité entre l'exemple de référence associé à chaque processeur et l'exemple d'entrée.

La figure (10) représente la cellule émettrice P_j connectée à la cellule réceptrice P_i par l'intermédiaire du poids synaptique W_{ij}



Rumelhart [RUME 86] a proposé une règle générale de modification de poids synaptique ΔW_{ij} valable pour de nombreux mécanismes d'apprentissage dans le cas d'une cellule P_i avec éventuellement une entrée « professeur » t_i .

$$\Delta W_{ij} = \alpha f_i (S_i, t_i) g_j (S_i, W_{ij}) \tag{I.3}$$

Avec : α : le pas d'adaptation

f_i et f_j : fonctions d'activations de P_i et P_j .

S_i et S_j : valeurs d'activations de P_i et P_j .

t_i : entrée professeur de P_i .

Dans la règle de HEBB, il n'y a pas de professeur et les fonctions f_i sont linéaires : le mécanisme d'adaptation se réduit à :

$$\Delta W_{ij} = \alpha S_i S_j \tag{I.4}$$

La règle de WIDROW HOFF, connue également sous le nom de Delta règle (ou Δ règle) nécessite un professeur et se formule en présence de fonctions d'activation linéaire sous la forme :

$$\Delta W_{ij} = \alpha (t_i - S_i) S_j \tag{I.5}$$

Le modèle compétitif de type KOHONEN met en œuvre l'expression :

$$\Delta W_{ij} = \alpha (t_i - W_{ij}) \tag{I.6}$$

Après la phase d'apprentissage, toute la mémorisation des connaissances est gravée dans le réseau. C'est la phase la plus longue en calcul connexionniste. Par contre, en phase de reconnaissance, si la forme à reconnaître a déjà été apprise, le réseau fournira instantanément la bonne sortie.

Certains réseaux dédiés aux problèmes de classification automatique ont été conçus pour rejeter de façon certains les exemples qui n'ont pas été mémorisés par le réseau [10]

I.11.1. Les mécanismes de classification :

La cellule de décision avec fonction de seuil se comporte comme un adaptateur linéaire qui réalise la séparation des exemples en deux classes w_1 et w_2 à l'aide de la fonction discriminante linéaire :

$$g(x) = w^{\text{tr}}x \quad (\text{I.7})$$

$w = (W_{1j}, W_{2j}, \dots, W_{nj}, W_{0j})^{\text{tr}}$, vecteur des poids synaptiques.

$X = (X_1, X_2, \dots, X_n, 1)^{\text{tr}}$.

Cette fonction permet la séparation de l'ensemble des vecteurs x en deux classes w_1 et w_2 :

$$x \in w_1 \text{ si } g(x) \geq 0.$$

$$x \in w_2 \text{ si } g(x) < 0.$$

La (figure 11) montre l'interprétation géométrique du mécanisme de classification [10].

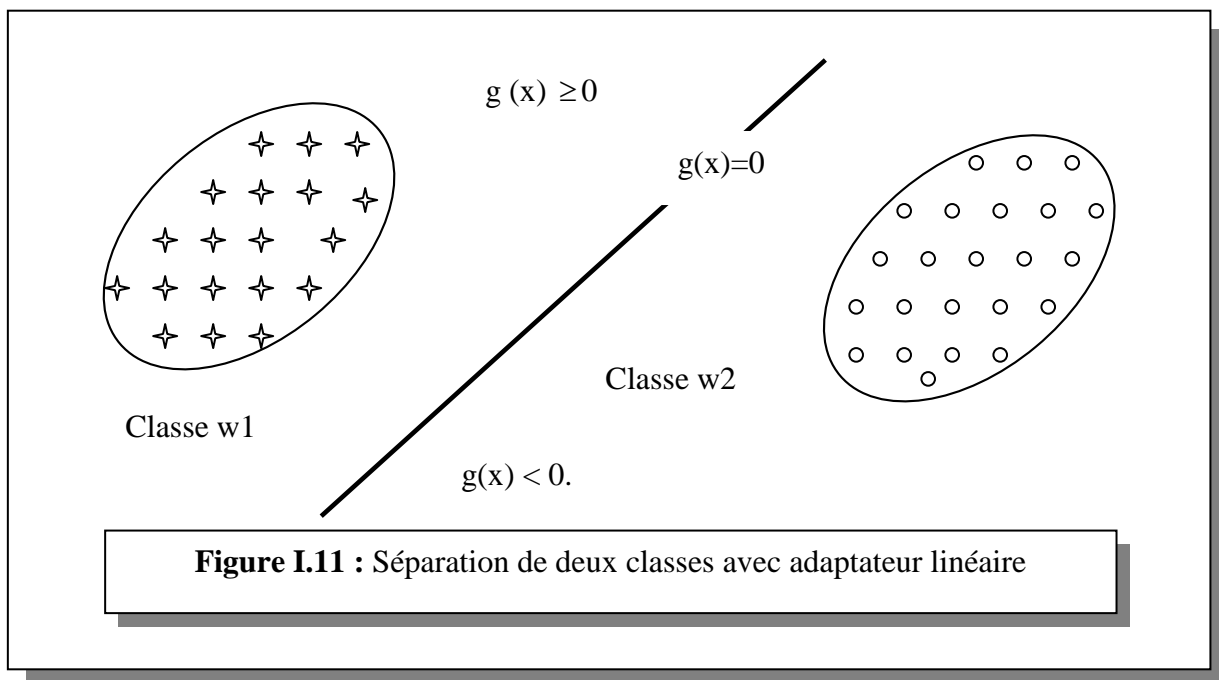


Figure I.11 : Séparation de deux classes avec adaptateur linéaire

I.11.2. Les mémoires associatives :

Les ordinateurs classiques de type Neumann traitent de façon séquentielle des informations qui sont codées sous forme binaire et rangées dans des mémoires accessibles par leur adresse indépendamment de leur contenu. La défaillance d'une seule mémoire provoque l'interruption de la chaîne de traitement séquentiel des opérations. Une caractéristique importante des réseaux de neurones réside dans la manière avec laquelle sont stockées les informations et dans la façon de les associer.

En règle générale, la mémoire des réseaux de neurones est distribuée et toute ou partie des poids de connexion du réseau constitue la mémoire du réseau. Ainsi, un exemple de la base des exemples d'apprentissage sera mémorisé par un sous ensemble particulier des cellules qui contient également la mémoire de certains autres exemples de la base d'apprentissage. La majorité des réseaux de neurones utilisés en diagnostic industriel pour la classification des défauts sont appelés des mémoires associatives ou classifieurs. On distingue deux familles de réseaux de neurones : les réseaux à base de neurones appelés souvent neurones « grand-mère » et les mémoires auto associatives et les mémoires hétéro associatives

Dans ce type de réseau, les neurones répondent spécifiquement à un seul exemple de la base d'apprentissage. Le traitement de l'information est réalisé par plusieurs neurones indépendants les uns des autres. Le surnom de tels réseaux repose sur l'hypothèse que le cerveau humain stocke l'image de notre grand-mère dans un neurone particulier placé dans un endroit précis du cerveau. Ces réseaux sont particulièrement adaptés aux problèmes de classification [10].

I.11.3. Mémoires auto associatives et hétéro associatives :

Dans une mémoire associative, les informations ne sont pas localisées à des emplacements déterminés, mais la mémorisation de chaque exemple est distribuée sur l'ensemble de la structure du réseau qui constitue ainsi la mémoire. Les seules sorties possibles sont également des contenus des mémoires. Les mémoires auto associatives ont pour caractéristique essentielle de reproduire en sortie, l'image apprise de l'entrée à partir d'un exemple incomplet ou bruité. Les mémoires hétéro associatives apprennent une relation entre un exemple donné et une sortie désirée différente de l'exemple. Si un nouvel exemple présenté est proche de l'exemple appris, il restituera la sortie désirée. Ces mémoires sont particulièrement performantes pour la reconnaissance de caractères avec des données incomplètes ou bruitées [10].

I.12. ALGORITHME DE RETRO PROPAGATION :

III.12.1. Introduction :

Le rétro propagation est le paradigme des **R N A** le plus utilisé. Le terme se réfère à un algorithme pour ajuster les poids de connexions en un **R N A** multi couches, ce paradigme a été appliqué avec succès dans différents domaines tel que le domaine militaire, médicale, synthèse de la parole, traitement de signal...etc.

Le rétro propagation est basé sur des principes mathématiques. Cette méthode a donné de bons résultats dans beaucoup d'applications. Pour l'appliquer il suffit d'avoir des données d'entrées et des données de sortie.

Cependant, toute information que possède un réseau de neurone est représentée par les poids d'interconnexions, cette information est acquise durant la phase d'apprentissage.

Cette propriété d'apprendre a permis l'application des tels réseaux dans plusieurs domaines, contrôle, identification, diagnostic,...etc [7].

I.12.2. Présentation :

Cet algorithme que l'on désigne couramment par «Back propagation » est une généralisation de la règle de « WIDROW HOFF » pour un réseau multi couches.

Il a été mis au point simultanément par deux équipes indépendantes en France «FEGELAM N SAULIE, GALLINARI, LECUN » et aux Etats-Unis «RUMELHART, HITON, WILLIAMS ».

L'idée simple qui est à la base de cet algorithme, et qui permet de lever la difficulté du « crédit assignent problème » est l'utilisation d'une fonction dérivable (fonction sigmoïde) en remplacement de la fonction de seuil utilisée dans le neurone linéaires a seuil.

Mathématiquement, cet algorithme utilise simplement les règles de dérivation composée et ne présente aucune difficulté particulière.

Le principe de cet algorithme est que, de même que l'on est capable de propager un signal provenant de cellules d'entrée vers la couche de sortie, on peut, en suivant le chemin inverse, rétro propager l'erreur commise en sortie vers les couches internes.

I.12.3. Position de problème :

L'apprentissage supervisé consiste à ajuster les coefficients synaptiques pour que les sorties du réseau soient les plus proches possibles des sorties de l'ensemble d'entraînement. Donc il faut spécifier une règle d'apprentissage pour l'adaptation de ces paramètres.

Pour remédier à ce problème, on utilise la méthode de rétro propagation de l'erreur pour l'apprentissage des réseaux statiques multi couches.

I.12.4. Modèle et équation du réseau :

Le réseau utilisé est un réseau multi couches, comportant une couche d'entrée qui correspond à la rétine, une couche de sortie qui correspond à la décision, et un certain nombre de couches dites cachées. Ces couches cachées constituent de véritable représentation interne des problèmes [7].

Le neurone utilisé dans le réseau est fondamentalement de même nature que le neurone linéaire à seuil du perceptron, il applique une fonction à la somme pondérer de ses entrées, cette fonction est une version lissée de la fonction à seuil on utilise en général fonction sigmoïde qui s'écrit :

$$f(S) = 1/(1+e^{-S}) \quad (I.8)$$

Les états des différents neurones dans un réseau ayant « L » couches (couches cachées et couches de sortie) données ayant « N » entrées et « M » sorties, sont données par les équations suivantes :

$$O_{pi} = f(S_{pi}) \quad (I.9)$$

Avec :

$$S_{pi} = \sum W_{ij} \cdot O_{pi} \quad (I.10)$$

Telle que

O_{pi} : La sortie de neurone i pour l'exemple (entrée, sortie désirée) p .

S_{pi} : Le potentiel somatique du neurone i pour l'exemple (entrée, sortie désirée) p .

W_{ij} : Coefficient synaptique (poids) de la j^{eme} entrée du neurone i .

I.12.5. Principe de rétro propagation :

La rétro propagation est basée sur l'adaptation des coefficients synaptiques dites encore des coefficients de pondération dans le but de minimiser une fonction de coût (performance) donnée par :

$$E(W) = \sum_{P=1}^P E_p(W) \quad (I.11)$$

$$E_p(W) = \frac{1}{2} \sum_{i=1}^n (t_{pi} - O_{pi})^2 \quad (I.12)$$

Ou : t_{pi} et O_{pi} Représentant respectivement la sortie désirée (Target) et la sortie (output) du réseau.

P : Le nombre d'exemple ou de la longueur de l'ensemble d'entraînement.

La minimisation de l'erreur se fait par une approximation d'une descente de gradient comme dans la règle de Widrow Hoff (règle delta). Toute la difficulté pour effectuer cette descente dans un réseau multi couches était de pouvoir calculer la dérivée de l'erreur quadratique par rapport à un poids donné.

I.12.6. Adaptation des poids :

L'adaptation (ajustement, mise en forme) des coefficients synaptique, se fait par la méthode du gradient basée sur la formule itérative suivante :

$$W_{ij}(n+1) = W_{ij}(n) + \Delta W_{ij}(\text{époque}) \quad (I.13)$$

$$\Delta_p W_{ij} = -\eta \cdot \partial E_p / \partial W_{ij} \quad (I.14)$$

n : Représente le numéro d'itération.

η : Représente la pas d'apprentissage. Ce facteur influe sur la vitesse de convergence du réseau.

La dérivée partielle de E (W) par rapport à chaque poids du réseau, pour cela et on utilisant la règle de delta généralisée. Nous devons charger notre poids liés à :

$$\Delta_p W_{ij} = \eta \cdot \delta_{pi} \cdot O_{pj} \quad (I.15)$$

Avec :

O_{pj} : Sortie du neurone j pour un exemple p .

δ_{pi} : L'erreur commise à la sortie du neurone i pour l'exemple p , posant maintenant :

$$\delta_{pi} = -\partial E_p / \partial S_{pi} = (t_{pi} - O_{pi}) \quad (I.16)$$

L'astuce est de calculer δ_{pi} pour chaque neurone dans le réseau. Le résultat intéressant que nous avons dérivé maintenant, est qu'il y a un simple calcul récursif de ces δ qui peuvent être implémentés par la rétro propagation du signal d'erreur à travers le réseau.

Pour calculer : $\delta_{pi} = -\partial E_p / \partial S_{pi}$.

Nous appliquons la décomposition en chaîne pour exprimer dérivation partielle, nous obtenons :

$$\delta_{pi} = -\partial E_p / \partial S_{pi} = -\partial E_p / \partial O_{pi} * \partial O_{pi} / \partial S_{pi}.$$

Calculons le deuxième facteur de l'équation (15) nous voyons que :

$$\partial O_{pi} / \partial S_{pi} = f'(S_{pi}).$$

Où : $f'(S_{pi})$: Est la dérivée de la fonction d'activation du neurone i évaluée à S_{pi} .

Pour calculer le premier facteur nous considérons deux cas :

Supposant que le neurone U_i est un neurone de sortie du réseau, dans ce cas, il paraît de la définition de E_p que : $\partial E_p / \partial O_{pi} = - (t_{pi} - O_{pi})$.

Substituons les deux facteurs dans l'équation (16) nous obtenons :

$$\delta_{pi} = (t_{pi} - O_{pi}) F'_i(S_{pi}) \quad (I.17)$$

Pour tout neurone de sortie U_i .

Si U_i n'est pas un neurone de sortie, nous utilisons la décomposition en chaîne pour écrire :

$$\begin{aligned} \sum \partial E_p / \partial S_{pk} * \partial S_{pk} / \partial O_{pi} &= \sum \partial E_p / \partial S_{pk} * \partial / \partial O_{pi} \sum W_{ki} O_{pi} = \\ &= \sum \partial E_p / \partial S_{pk} W_{ki} = \sum \delta_{pk} W_{ki}. \end{aligned}$$

Dans ce cas, substituons les deux facteurs dans l'équation (16) nous obtenons :

$$\delta_{pi} = f'(S_{pi}) \sum \delta_{pk} * W_{ki} \quad (I.18)$$

Où : U_i est un neurone appartenant aux couches cachées

Les équations (17) et (18) donnent une procédure récursive pour calculer les δ pour tout neurone dans le réseau liée à l'équation (14), cette procédure constitue «la règle delta généralisée» pour un réseau statique, aussi pour le neurone non linéaire.

Pour minimiser l'erreur total sur l'ensemble d'entraînement, les poids du réseau peuvent être ajustés par la présentation de l'ensemble d'apprentissage en entier.

La variation des poids $\Delta_p W_{ij}(n)$ peut alors s'écrire ainsi :

$$\Delta_p W_{ij}(n) = \eta \partial E_p(W) / \partial W_{ij}(n) \quad (\text{I.19})$$

I.12.7. Les 'étapes d'algorithme de rétro propagation :

L'algorithme de rétro propagation est représenté comme suit :

Etape **01** : Initialiser les poids W_{ij} et les seuils internes des neurones à de petites valeurs aléatoires.

Etape **02** : Présenter l'ensemble des couches (entrée, sortie désirée).

Etape **03** : Présenter le première couché (entrée, sortie désirée) de l'ensemble.

Etape **04** : Calculer :

- 1- La somme des entrées des neurones de la couche cachée par l'équation (I.10)
- 2- Les sorties des neurones de la couche cachée par l'équation (I.9)
- 3- La somme des entrées de la couche de sortie par l'équation (I.10).
- 4- Les sorties de réseau par l'équation (I.9).

Etape **05** : Calculer :

- 1- Les termes de l'erreur pour les neurones de la couche de sortie par l'équation (I.17).
- 2- Les termes de l'erreur pou les neurones de la couche cachée par l'équation (I.18).

Etape **06** : Calculer la variation des poids par l'équation (I.15).

Etape **07** : Calculer l'erreur E_p par l'équation (I.12).

Etape **08** : Présenter un autre couple (entrée, sortie désirée) et aller à l'étape quatre.

Etape **09** : Si tout l'ensemble des couple est présenté, calculer la variation total des poids par époque, en utilisant l'expression :

$$\Delta W_{ij}^{(\text{époque})} = \sum_{P=1}^P \Delta_p W_{ij}.$$

Etape **10** : Ajuster les poids par l'équation (I.13).

Etape **11** : Calculer l'erreur totale par l'équation (I.11).

Etape **12** : Comparer l'erreur totale avec une erreur prédéterminée (E_{PR}).

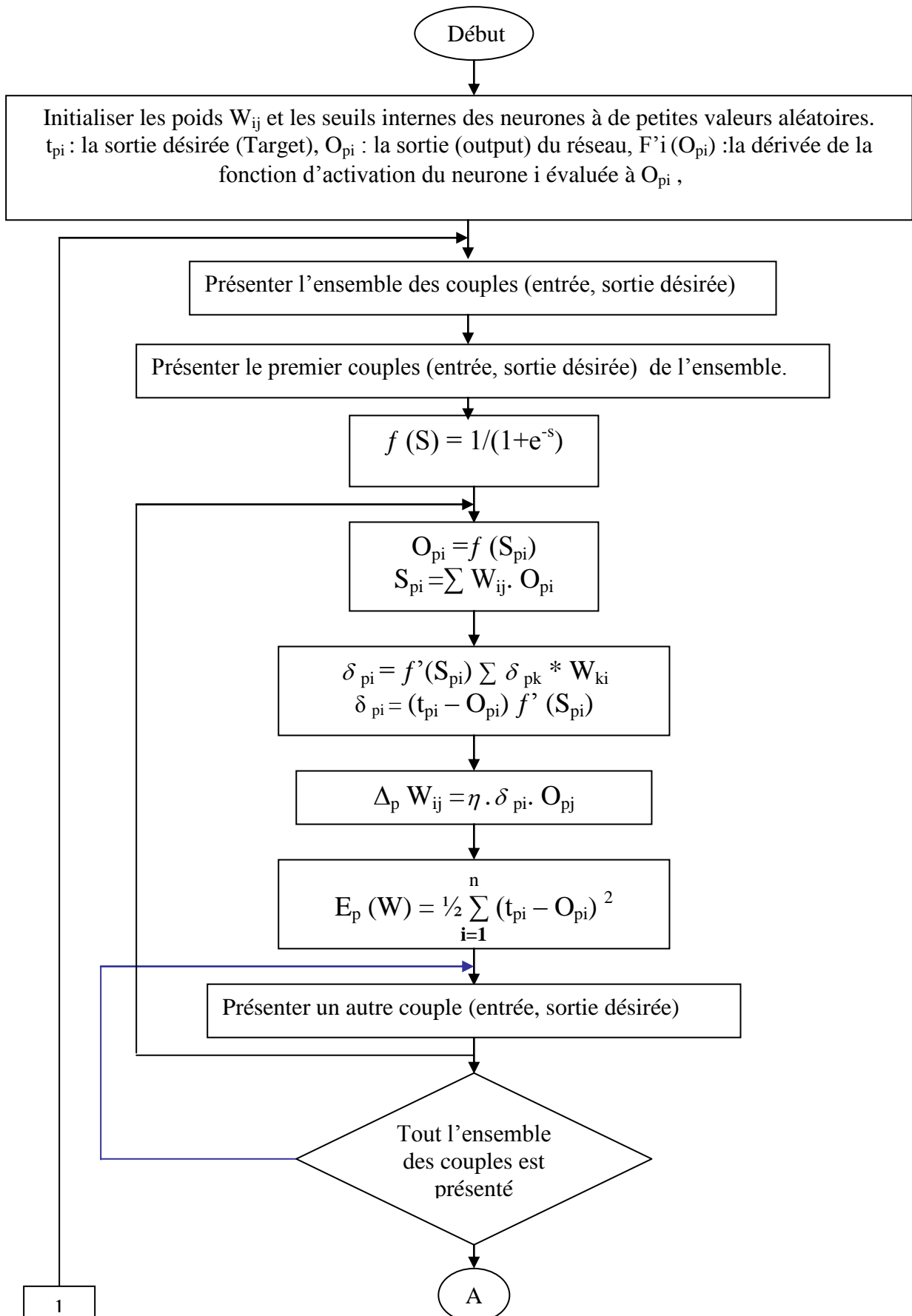
$$[E - E_{PR} < \varepsilon].$$

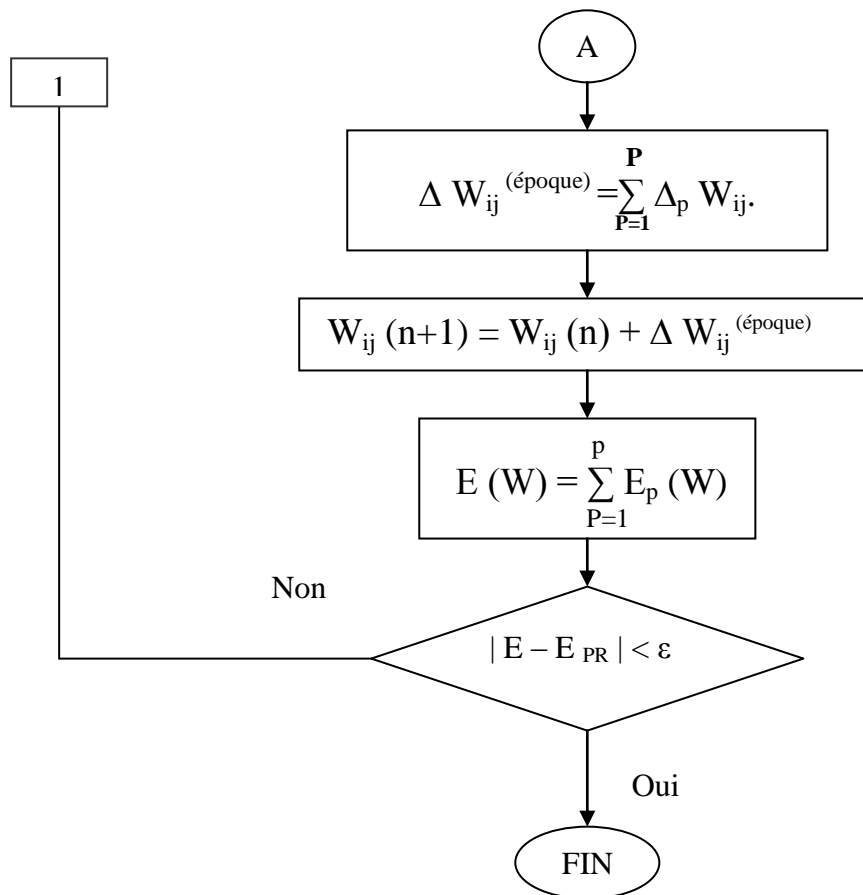
Si la condition est vérifiée aller à l'étape **13**.

Si non aller à l'étape **03**.

Etape **13** : Fin

I.12.8. Algorithme de rétro propagation :





I.13. DOMAINES D'APPLICATION DES RESEAUX DE NEURONES :

Les propriétés d'application et d'approximation des réseaux de neurones ont permis une importante application de ces derniers dans les différents domaines pratiques, notamment, les domaines suivants :

I.13.1. Classification des signaux :

Classer des signaux dans les catégories différentes en fonction des caractéristiques de forme (amplitude, fréquence, phase, etc. ...) peut être réalisé aisément par l'utilisation d'un réseau de neurones.

Supposons qu'on a deux catégories de signaux A et B et qu'on souhaite classer de différents formes dans ces deux catégories en fonction des caractéristiques de ces formes.

On peut simplement définir la fonction non linéaire F qui vaut +1 pour les formes de la classe A et -1 pour celles de la classe B, le problème de classification revient à approximation de la fonction non linéaire F qui peut être résolue par un réseau de neurones qui possède des entrées et une seule sortie pour fournir la décision.

I.13.2. Identification des processus :

Identifier un processus ; c'est d'essayer de donner un modèle plus proche au processus réel qui permet de reproduire avec la précision voulue, la sortie en fonction des entrées du processus, c'est typiquement un problème d'approximation qui peut être résolu par un réseau de neurones.

I.13.3. Contrôle des systèmes :

Contrôler un système; c'est donner un ensemble d'entrées adéquates qui permettent d'avoir la sortie désirée (position, vitesse, etc....), et comme étant la fonction liant la sortie du systèmes avec l'ensemble de commande est non linéaire, cette fonction peut être approchée avantageusement par un réseau de neurones.

I.13.4. L'application au diagnostic :

Les réseaux de neurones sont bien adaptés à la résolution des problèmes de diagnostic, utilisant la classification automatique des signaux et des formes. Dans ce contexte, on distingue plusieurs applications des réseaux de neurones pour le diagnostic des défaillances et en particulier, pour le diagnostic des pannes des machines électriques.

I.14. LES CARACTERISTIQUES D'UNE BONNE APPLICATION :

Nous pouvons déterminer quelques unes des caractéristiques des problèmes bien adaptés à une résolution par les réseaux de neurones :

- 1- Les règles qui permettraient de résoudre le problème sont inconnues ou très difficiles à expliciter ou à formaliser. Cependant, on dispose d'un ensemble d'exemples qui correspondent à des entrées du problème et aux solutions qui leur sont données par des experts.
- 2- Le problème fait intervenir des données bruitées.
- 3- Le problème peut évaluer, par exemple en faisant varier son champ de conditions initiales.
- 4- Le problème nécessite une grande rapidité de traitement, il doit par exemple être traité en temps réel.
- 5- Il n'existe pas de solutions technologiques courantes. Cette approche amène à dresser la liste suivante des domaines d'applications privilégiés :
 - Reconnaissance des formes.
 - Traitement de signal.
 - Vision, parole.
 - Prévision et modélisation.
 - Aide à la décision.
 - Robotique.

Ces quelques domaines possèdent pratiquement toutes les caractéristiques exposées précédemment, c'est pourquoi ils constituent le cœur des applications des réseaux de neurones.

Il ne faut cependant pas oublier le rôle que peuvent jouer les réseaux de neurones formels dans l'étude du cerveau en tant qu'instrument de simulation, et dans la recherche sur les neuro-ordinateurs et les nouvelles architectures d'ordinateurs en tant qu'instruments de formalisation et de simulation [4].

I.15. CONCLUSION :

Dans ce chapitre, nous avons exposé les réseaux de neurones, en spécifiant leurs définitions, les concepts de base de cette technique, l'apprentissage des réseaux de neurones, et ses différents domaines d'utilisation.

Nous avons présenté une étude sur l'algorithme du rétro propagation qui a prouvé son efficacité dans la pratique avec ses capacités d'apprentissage et d'identification.

Pour ces raisons, cet algorithme sera la base du travail qui sera présenté dans le prochain chapitre qui traite l'identification et la commande par les réseaux de neurones.

CHAPITRE II :

Identification et commande par les RNA

II.1. INTRODUCTION :

Comme approximateurs universels, les réseaux de neurones ont des capacités de modélisation très intéressantes. L'application des techniques neuronales pour l'identification des systèmes non linéaires peut fournir de nouvelles solutions pour ce problème.

L'identification d'un système non linéaire dont la structure peut être inconnue consiste à trouver un ensemble d'équations non linéaires qui représentent d'une manière adéquate son comportement dynamique.

Plusieurs approches ont été proposées pour la modélisation des systèmes non linéaires à structure inconnue telles que :

- la modélisation par les séries de Volterra et Wiener ;
- la modélisation par les modèles poly nominales d'Ivakhninko (GMDH : Group Method of Data Handing) ;
- la modélisation par des règles floues ;

Si bien qu'il existe plusieurs méthodes d'identification des systèmes linéaires et même pour les systèmes non linéaires à structure connue comme les méthodes de gradient et les filtres non linéaires, des méthodes générales et efficaces qui peuvent être appliquées pour l'identification des systèmes non linéaires à structure inconnue ne sont pas encore disponibles.

Avec leurs propriétés remarquables d'approximation, d'adaptativité et de généralisation, les réseaux de neurones peuvent fournir des solutions prometteuses pour ce problème. Cette approche a été adoptée par plusieurs chercheurs, son efficacité et ses performances ont été démontrées à travers plusieurs applications.

II.2. Identification par les réseaux de neurones:

L'entraînement d'un réseau de neurones en utilisant les données entrées – sorties d'un système non linéaire peut être considéré comme un problème d'approximation de fonctions non linéaires. A partir du théorème de Weierstrass, il est connu que les polynômes et d'autres structures d'approximation peuvent approcher une fonction continue quelconque avec une précision arbitraire. Récemment, l'application des techniques mathématiques pour l'étude des capacités d'approximation des réseaux de neurones était l'objet de plusieurs travaux.

Les résultats obtenus montrant les perceptrons multi-couches peuvent approcher n'importe quelle fonction continue, et précisent qu'un réseau à une seule couche cachée dont la fonction d'activation est la fonction sigmoïde peut approcher une fonction continue définie sur un sous ensemble compact de R_n . L'utilisation d'autres fonctions autres que les sigmoïdes est possible et non constantes.

Si bien que ces résultats semblent être intéressants, ils ne sont pas suffisants pour caractériser les propriétés d'approximation des réseaux. En plus ces résultats supposent que ces réseaux contiennent un nombre suffisant de neurones cachés et que les poids des connexions

sont correctement choisis. La possibilité de détermination des poids appropriés en utilisant les algorithmes d'apprentissage existants est un problème ouvert. Une étude mathématique plus approfondie est nécessaire pour permettre la définition du type et de la structure convenable pour un problème donné.

L'idée de base de l'approche connexionniste est de substituer aux modèles paramétriques classiques des modèles neuronaux dont les paramètres sont adaptés par une procédure d'apprentissage appropriée en utilisant les données entrées-sorties du système. En l'absence de résultats théoriques concrets concernant la stabilité et l'identifiabilité du système c'est -à- dire la capacité de la structure du modèle choisie à représenter le système d'une manière adéquate, on suppose que tous les systèmes qu'on est entrain d'étudier appartiennent à la classe de systèmes que le réseau choisi est capable de les représenter.

II.2.1. Modèle directe:

La modélisation directe consiste à entraîner un réseau de neurones pour représenter la dynamique directe du système. Dans ce cas le modèle est placé en parallèle avec le système. L'erreur entre les sorties est utilisée pour l'adaptation des paramètres du modèle. Cette structure correspond à un problème d'apprentissage supervisé classique, dans lequel le «professeur» (dans ce cas le système) génère les sorties désirées. Dans le cas des réseaux multi-couche, on utilise généralement la méthode de rétropropagation pour l'adaptation des poids.

La modélisation des caractéristiques dynamiques du système peut être faite par l'introduction de ces caractéristiques dans le réseau lui-même, par l'utilisation de réseaux récurrents, ou par l'introduction d'un comportement dynamique dans les neurones.

L'approche la plus évidente consiste à augmenter la dimension du vecteur d'entrée du réseau en ajoutant les signaux correspondant aux mesures précédentes des entrées et des sorties. supposons que le système à identifier est gouverné par l'équation aux différences non linéaire suivante :

$$Y(t+1) = f[Y(t), \dots, Y(t-n_y+1); U(t), \dots, U(t-n_u+1)] + e(t+1) \quad (\text{II-1})$$

$$\text{Avec: } Y(t) = [Y(t), \dots, Y_m(t)]^T \quad (\text{II-2})$$

$$U(t) = [U_1(t), \dots, U_r(t)]^T \quad (\text{II-3})$$

$$e(t) = [e_1(t), \dots, e_m(t)]^T \quad (\text{II-4})$$

Où $f(\cdot)$ est une fonction non linéaire, $Y(t)$, $U(t)$ et $e(t)$ sont les vecteurs de sortie, d'entrée et du bruit de mesure. A l'instant $t+1$, la sortie du système dépend (dans le sens définit par la fonction non linéaire f) des n_y valeurs précédentes de la sortie et des n_u valeurs précédentes de l'entrée. Le but de l'approche connexionniste est d'approcher la fonction f avec un réseau de neurones.

En s'inspirant des techniques d'identification classiques, il est possible d'utiliser deux structures d'identification.

II.2.1.1. Identification série parallèle

Dans la première structure, représentée par la figure (II.1), on utilise des réseaux non récurrents (statiques) dont la structure entrée-sortie est donnée par :

$$Y(t) = f[X^m(t)] \tag{II-5}$$

$X^m(t)$ et $Y^m(t)$ sont les vecteurs d'entrée et de sortie du réseau. L'entrée à un instant t est :

$$X^m(t) = [Y^t(t-1), \dots, Y^t(t-n_y); U^t(t-1), \dots, U^t(t-n_u)] \tag{II-6}$$

La dimension du vecteur d'entrée :

$$n_1 = m \cdot n_y + r \cdot n_u \tag{II-7}$$

Le but de la procédure d'apprentissage est d'entraîner le réseau pour que la transformation non linéaire \tilde{f} soit une approximation de la fonction f .

Dans cette structure, l'entrée du modèle comporte les valeurs décalées de la réponse du système réel, elle correspond au modèle d'identification série-parallèle de l'identification classique.

Comme dans le cas des méthodes classiques, il apparaît une contrainte imposée a priori sur la connaissance de l'ordre du système (ou d'une borne supérieure sur cet ordre).

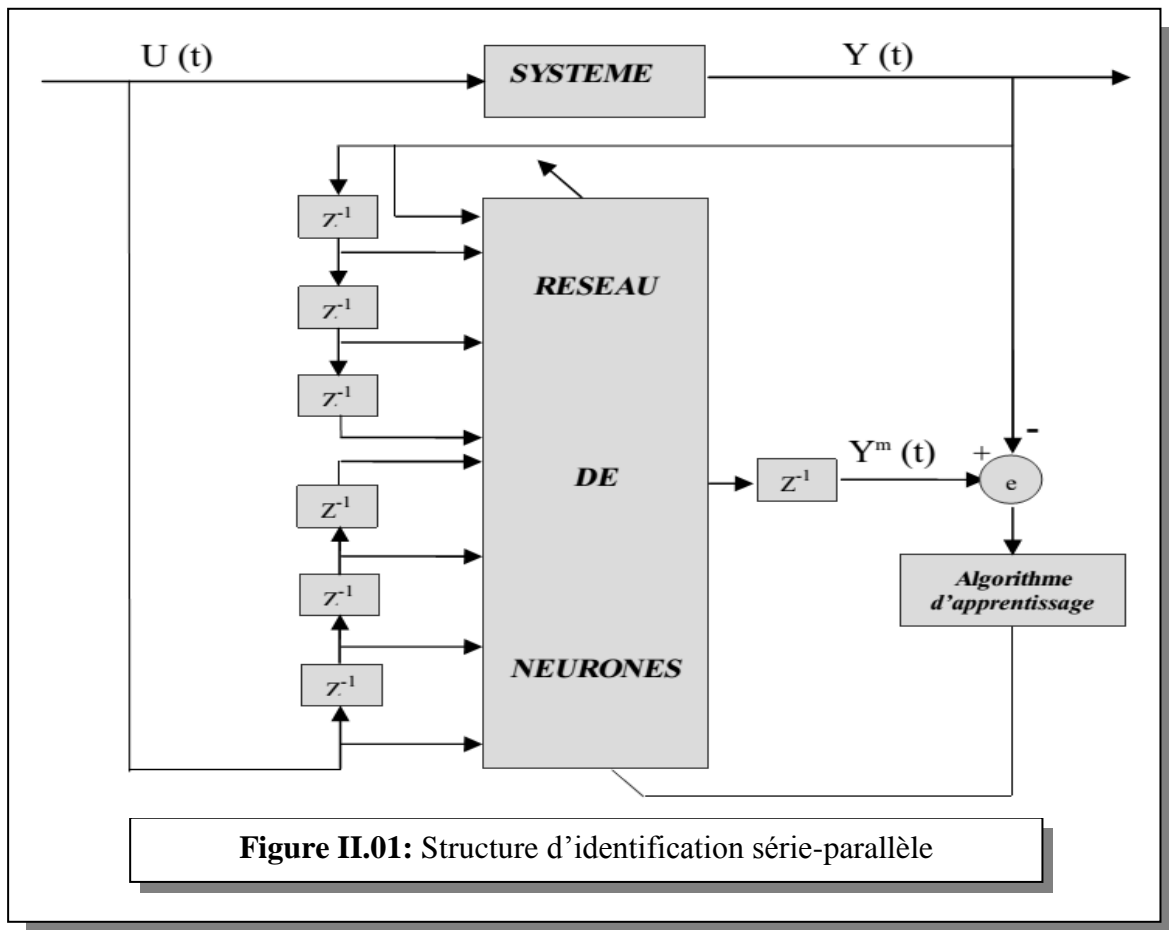


Figure II.01: Structure d'identification série-parallèle

II.2.1.2. Identification parallèle

Dans la deuxième structure, représentée par la figure (II.2), on utilise des réseaux récurrents, l'entrée comporte les valeurs décalées de la réponse du modèle. Elle correspond au modèle d'identification parallèle dans ce cas, la sortie du modèle neuronal est donnée par :

$$Y^m(t) \hat{f}[X^m(t)] \tag{II-8}$$

Avec: $X(t) = [Y^{mT}(t-1), \dots, Y^{mT}(t-n); U(t-1), \dots, U(t-n_u)]^T$ (II-9)

Cependant il n'y a aucune garantie de convergence si la structure parallèle est utilisée. Malgré plusieurs années de recherche, les conditions de convergence du modèle parallèle ne sont pas encore établies même dans le cas des systèmes linéaires. Grâce à de ses propriétés de stabilité et de convergence l'utilisation de la structure série-parallèle est préférable.

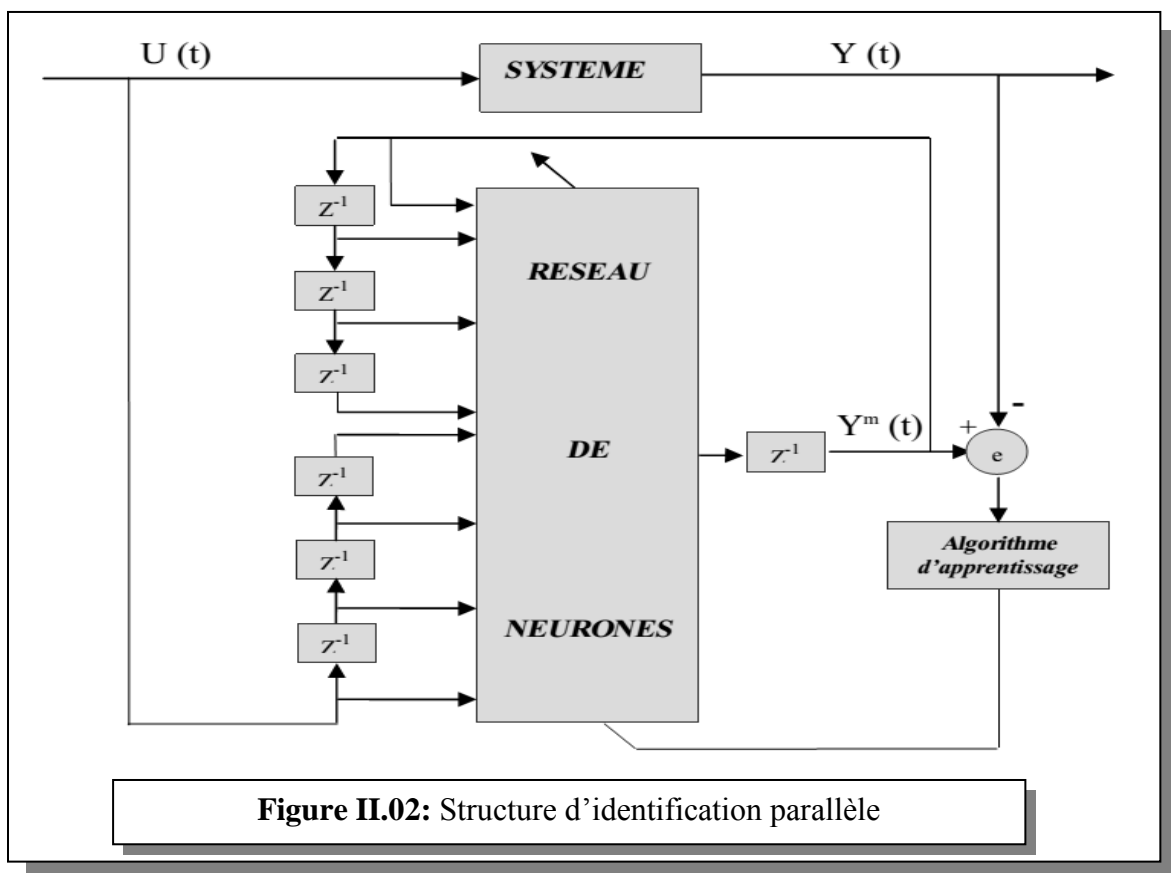


Figure II.02: Structure d'identification parallèle

L'avantage du modèle parallèle est qu'il présente des performances meilleures dans le cas de perturbations puisqu'il permet d'éviter les problèmes du biais causés par les bruits sur la sortie du système réel.

Généralement, lors d'une procédure d'identification, on commence par une identification série-parallèle. En supposant qu'après une période d'entraînement convenable le modèle neuronal représente d'une manière adéquate le comportement du système (les sorties du modèle sont suffisamment proches de celles du système), le modèle série parallèle peut être remplacé par un modèle parallèle. Ceci à un avantage pratique si le modèle d'identification est utilisé par exemple pour des procédures d'entraînement hors ligne d'un contrôleur neurone.

II.2.2. Modèle inverse:

Les modèles inverses des systèmes dynamiques jouent un rôle très important dans les structures de contrôle. La modélisation inverse par réseaux de neurones consiste à entraîner un modèle neuronal pour qu'il génère les commandes nécessaires permettant de déplacer le système de son état actuel $Y(t)$ à un état désiré $Y^d(t+1)$.

A partir de l'équation (II.1) décrivant le comportement dynamique du système, son comportement dynamique inverse peut être décrit par :

$$U(t) = f^{-1}[Y(t+1), \dots, Y(t-n_y+1); U(t-1), \dots, U(t-n_u+1)] \quad (\text{II-10})$$

La fonction inverse f^{-1} nécessite la connaissance de la valeur future $Y(t+1)$, cette valeur est remplacée par la valeur désirée $Y^d(t+1)$ qui est supposée disponible à l'instant t . La relation entrée-sortie du modèle inverse neuronal est donnée par :

$$U^m(t) = \hat{f}^{-1}[X^m(t)] \quad (\text{II-11})$$

$$\text{Avec: } X^t(t) = [Y^t(t), \dots, Y^t(t-n_y+1); U^T(t-n_u+1); Y_d^T(t+1)]^T \quad (\text{II-12})$$

La dimension du vecteur d'entrée dans ce cas est :

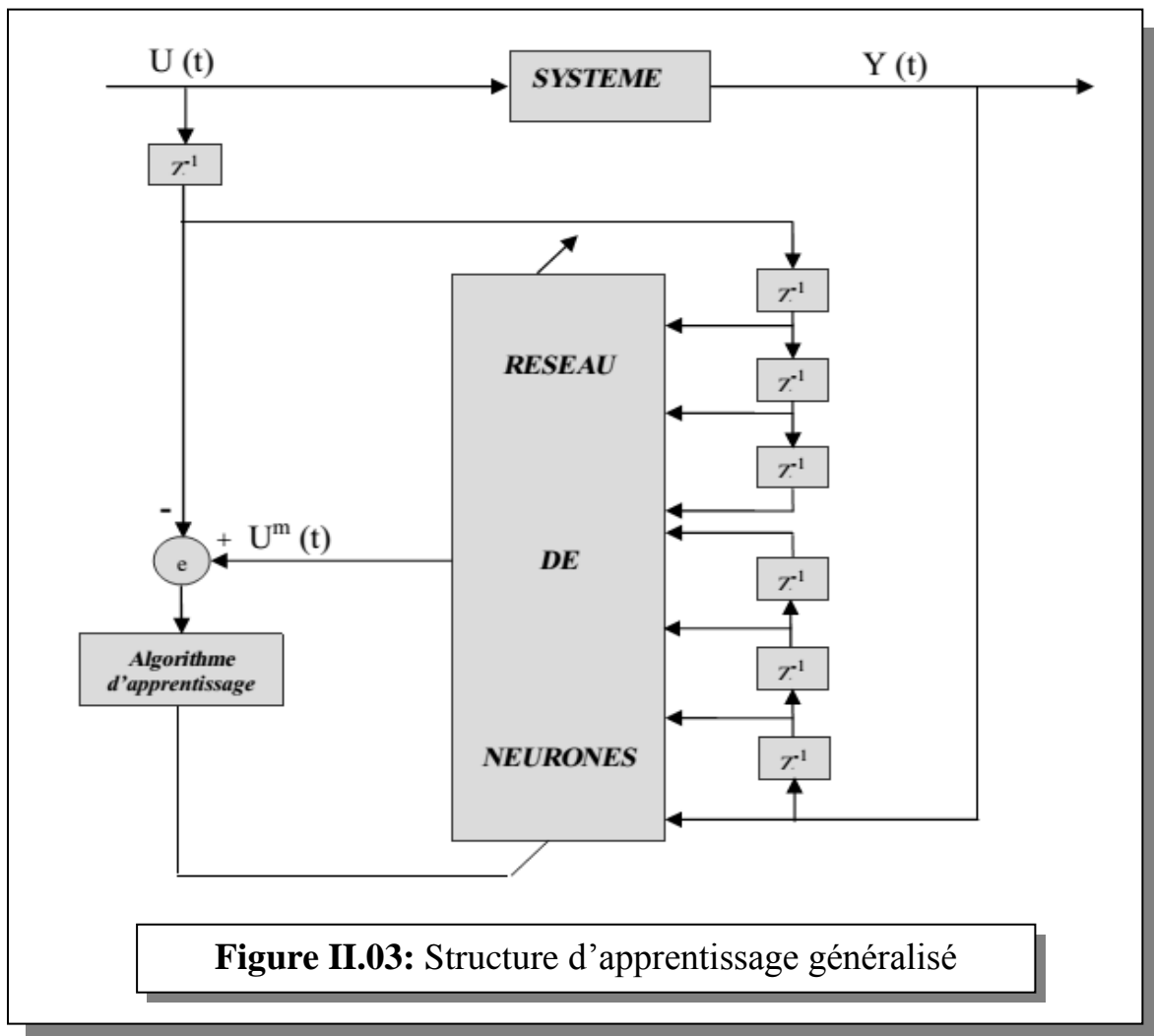
$$n_1 = m \cdot (n_y + 1) + r \cdot n_u \quad (\text{II-13})$$

Le but de la procédure d'entraînement est l'adaptation des paramètres du réseau pour qu'il approche la fonction inverse f^{-1} .

Cependant la commande désirée n'est pas connue a priori. Il faut donc arriver à évaluer l'erreur sur la commande qui sera ensuite utilisée pour l'entraînement du réseau. On peut utiliser pour cela plusieurs structures d'apprentissage.

II.2.2.1. Apprentissage généralisé:

L'approche la plus simple est la modélisation inverse directe illustrée par la figure (II.3), cette structure est appelée structure d'apprentissage généralisé. Dans ce cas, en utilisant la connaissance qualitative que l'on a sur le comportement du système, des signaux d'entraînement compatibles avec son fonctionnement lui sont appliqués. La sortie du système sert alors d'entrée au modèle neuronal dont la sortie est comparée avec le signal d'entraînement. En utilisant cette erreur pour l'entraînement du réseau, cette structure force le réseau à apprendre le comportement inverse du système.



Une fois entraîné, le réseau doit être capable de produire la commande nécessaire assurant de déplacer l'état du système à un état suffisamment proche de l'état désiré. Cependant cette structure souffre de deux inconvénients majeurs :

- La procédure d'apprentissage n'utilise pas des commandes désirées implicites. Puisque les commandes opérationnelles sont difficiles à définir a priori le signal d'entraînement doit être choisi sur un intervalle de l'espace d'entrée du système plus large que celui nécessaire.
- Si le système n'est pas une correspondance un à un on peut obtenir un modèle inverse incorrect.

Le premier point est étroitement lié au concept général d'excitations persistantes qui concerne le choix des entrées utilisées pour l'entraînement. Il faut trouver les conditions assurant les excitations persistantes c'est-à-dire les excitations qui assurent la convergence des paramètres. Pour les réseaux de neurones, les méthodes de caractérisation des excitations persistantes seront très importantes. Une étude préliminaire a été fait par Narendre et Parthasarathy.

Le succès de cette méthode est lié à la capacité du réseau à généraliser et à apprendre à répondre correctement aux entrées pour les quelles il n'a pas été entraîné. Pour surmonter ses difficultés, cette méthode peut être combinée avec une méthode d'apprentissage spécialisée.

II.2.2.2. Apprentissage spécialisé:

Une seconde approche de modélisation inverse dont le but est de dominer les problèmes cités précédemment est connue sous le nom de l'apprentissage spécialisé, sa structure est représentée par la figure (II.4).

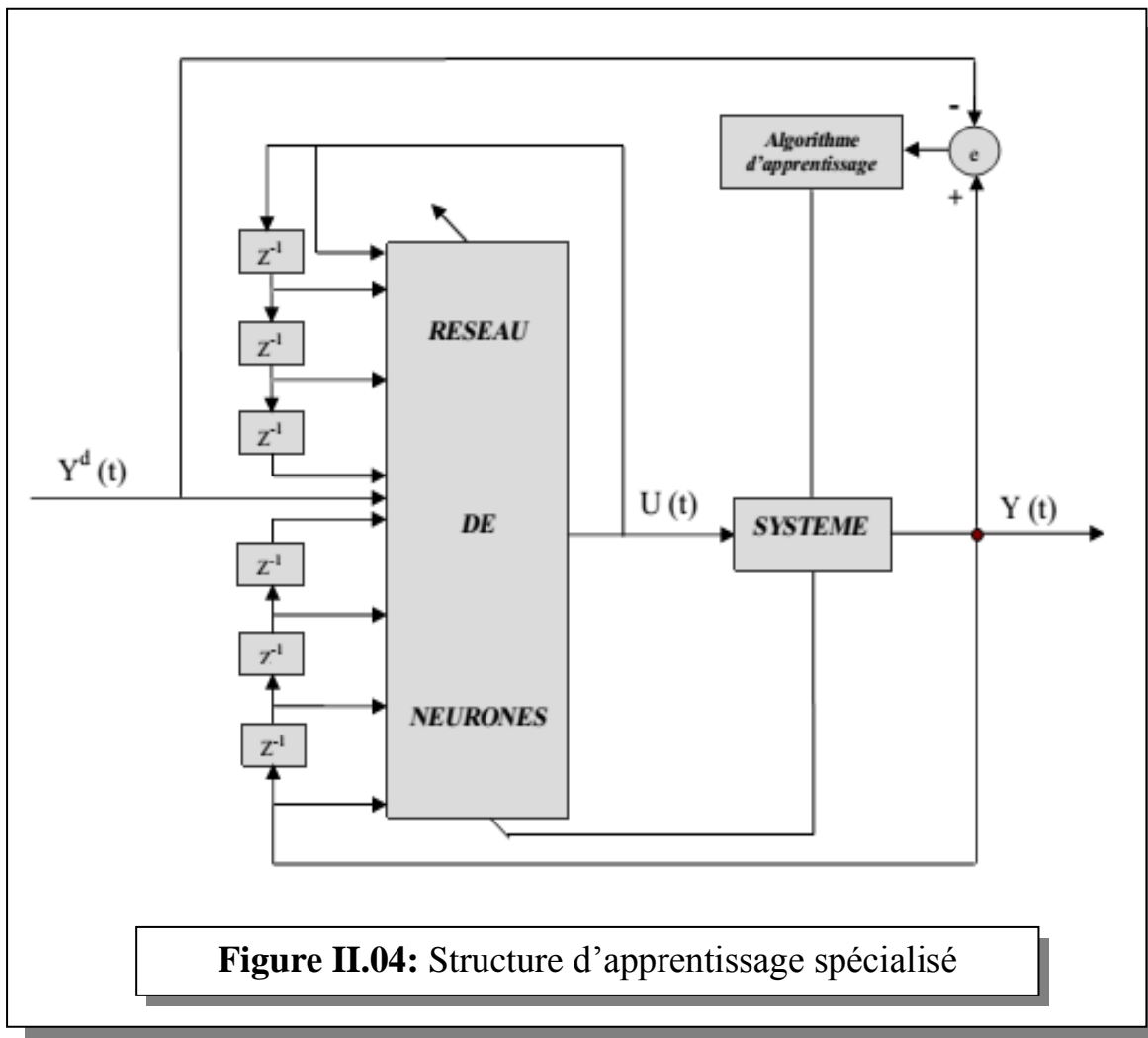


Figure II.04: Structure d'apprentissage spécialisé

Dans cette approche le modèle inverse neuronal est placé en série avec le système, et reçoit à son entrée un signal d'entraînement appartenant à l'espace des sorties opérationnelle désirée du système (le système de référence ou le signal de commande).

Dans le cas d'un entraînement par rétropropagation, le système est considéré comme une couche supplémentaire dont les poids sont fixes. Le but de la procédure d'apprentissage est de minimiser l'erreur E sur les sorties du modèle inverse, qui est donnée par :

$$E = \frac{1}{2} \sum_{i=1}^r (U_i - U_i^d)^2 \tag{II-14}$$

Or les commandes désirées ne sont pas connues, il faut donc évaluer les erreurs sur la commande à partir des erreurs sur la sortie.

L'erreur sur la sortie E' est donnée par :

$$E' = \frac{1}{2} \sum_{i=1}^m (Y_i(t) - Y_i^d(t))^2 \quad (\text{II-15})$$

La dérivée de l'erreur par rapport à un poids W_{ij} est :

$$\frac{\partial E'}{\partial W_{ij}(t)} = \sum_{i=1}^r \frac{\partial E'}{\partial U_i(t)} \frac{\partial U_i(t)}{\partial W_{ij}(t)} \quad (\text{II-16})$$

La dérivée de l'erreur E' par rapport à la commande :

$$\frac{\partial E'}{\partial W_{ij}} = \sum_{i=1}^r \frac{\partial E'}{\partial U_i(t)} \frac{\partial U_i(t)}{\partial W_{ij}(t)} \quad (\text{II-17})$$

L'évaluation de l'erreur nécessite le calcul du Jacobien du système.

Le Jacobien du système peut être obtenu par l'approximation des dérivées partielles des sorties du système par rapport à ses entrées, ou par l'utilisation d'un modèle direct entraîné du système par rapport à ses entrées, ou par l'utilisation d'un modèle direct entraîné du système (un modèle neuronal par exemple) qui sera placé en parallèle avec le système.

Dans ce cas, le signal d'erreur pour l'algorithme d'entraînement peut être la différence entre le signal d'entraînement et la sortie du système, ou la différence entre le signal d'entraînement et la sortie de modèle direct. Jordan et Rumelhart ont montré que l'utilisation de la sortie du système réel peut produire un modèle inverse exact même si le modèle direct est inexact, ce qui n'est pas évidemment vrai lorsqu'on utilise la sortie du modèle direct. Mais l'avantage de l'utilisation de la sortie du modèle est que le système réel n'est pas nécessaire pendant la procédure d'entraînement, ce qui est très important lorsque l'utilisation du système réel n'est pas possible.

L'adaptation des poids est faite par la rétro-propagation de l'erreur à travers le modèle direct et le modèle inverse, seuls les poids du modèle inverse sont ajustés pendant d'entraînement [13].

II.3. Commande par les réseaux de neurones:

La théorie de commande fournit des outils d'analyse et de synthèse adaptés aux systèmes linéaires. En pratique, l'application des techniques analytiques classiques basées sur la théorie des systèmes linéaires souffre de problèmes de robustesse à cause des incertitudes, perturbations et non linéarités.

Le traitement massivement parallèle, les capacités de modélisation et les possibilités d'apprentissage par expérience des réseaux de neurones sont des facteurs motivant pour le développement de système de contrôle à base des modèles connexionnistes.

Depuis leur apparition, les réseaux de neurones et en particulier les réseaux multicouches ont été appliqués dans plusieurs domaines vu leur capacité à commander les systèmes non linéaires et les systèmes dont les modèles sont peu connus ou ayant des paramètres variables dans le temps. De plus ils présentent une bonne résistance au bruit en entrée. Donc les applications des techniques neuronales peuvent fournir des solutions plus intelligentes avec des modérés.

Il existe plusieurs manières d'intégrer les réseaux de neurones dans les structures de commande des systèmes. Cela dépend essentiellement du schéma de réglage et de l'apprentissage choisi [13][14].

II.3.1. Les étapes de la conception d'un organe de commande:

Commander un processus, c'est déterminer les commandes à lui appliquer, de manière assurer aux variables d'état ou aux sorties qui nous intéressent un comportement précisé par un cahier des charges. Ces commandes sont délivrées par un organe de commande ; le processus et son organe de commande constituent le système de commande. L'élaboration de l'organe de commande s'articule en plusieurs étapes.

- 1 Choix d'un modèle du processus
- 2 Estimation des paramètres du modèle (identification)
- 3 Conception de l'organe de commande
- 4 Estimation des paramètres du correcteur

II.3.2. Principe de la commande:

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

Dans certaines situations, lorsqu'il est impossible de construire un contrôleur automatique en utilisant les techniques standard de la théorie de contrôle, les actions de contrôle sont délivrées par un opérateur humain pour réaliser des tâches particulières.

Dans quelque applications, il est peut être désirable de réaliser un contrôleur automatique qui imite les actions de l'opérateur humain. Un réseau de neurones peut être utilisé pour réaliser de telle tâche .le problème se réduit ainsi à un problème d'apprentissage supervisé.

Dans ce cas, les entrées du réseau sont les informations que reçoit l'opérateur humain sur l'état du système. Les sorties désirées sont les actions délivrées par l'opérateur [14].

II.3.3. Les structures de commande:

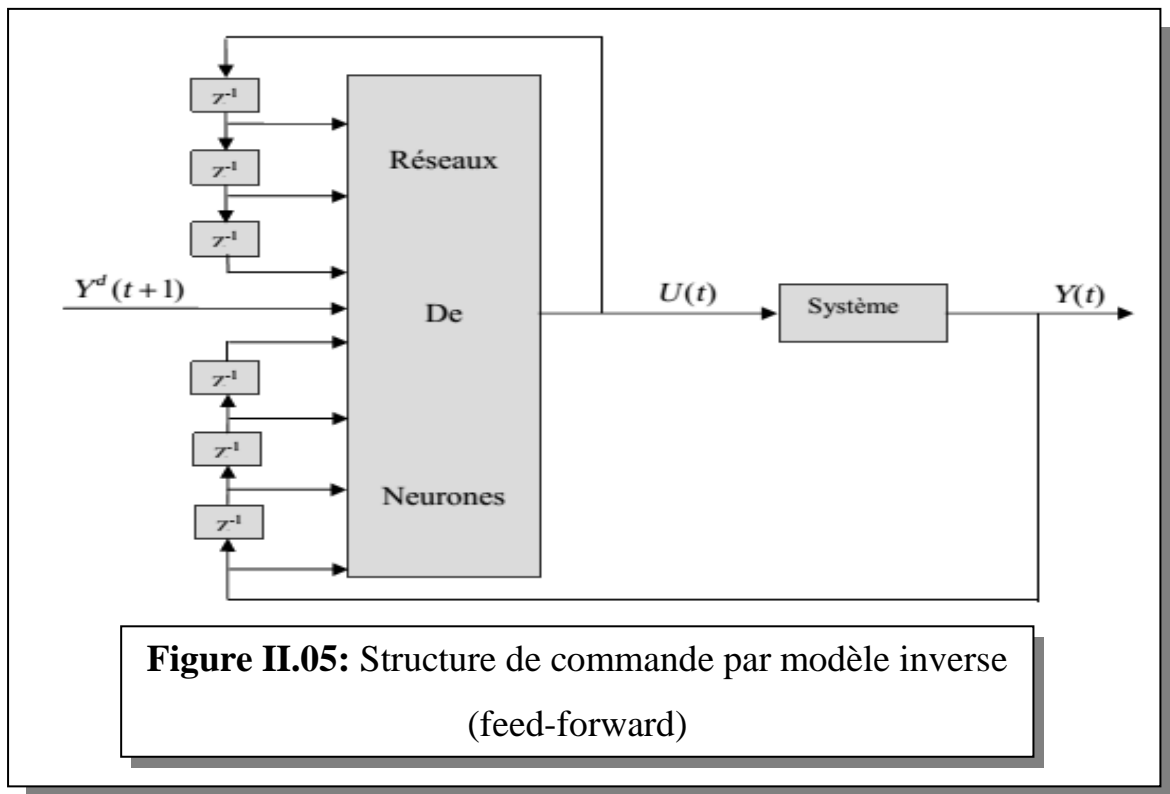
Malgré la diversité des structures de contrôle neuronal, nous allons présenter deux types différentes, telle que l'utilisation du modèle inverse et l'utilisation d'un modèle de référence.

II.3.3.1. Commande par modèle inverse

II.3.3.1.1. Contrôleur Feed-forward

Dans ce type de contrôleur, on utilise un modèle inverse pour le commande direct du système, l'obtention d'un modèle inverse se fait par les techniques étudiées dans la partie de l'identification. Le modèle inverse, comme montre la figure (II.5) est placé en série avec le système à contrôler.

Les deux types d'apprentissage (généralisé et spécialisé) peuvent être utilisé pour l'entraînement du réseau, En principe, le modèle inverse est capable de générer la commande nécessaire pour réaliser la tâche désirée.



Dans cette figure, on a adjoint au réseau de neurones qui constitue le modèle de processus un réseau de neurones qui calcule la loi de commande. Ce réseau est aussi un réseau non bouclé qui a pour entrées sortie du système $y(t)$ (état mesuré à l'instant t ou avec retard), la sortie désirée $y^d(t+1)$ (état au temps suivant) et la commande $u(t)$ (au temps précédent) dans le cas où l'on souhaite que $y^d(t+1)$ soit variable. La sortie du contrôleur neuronal est la commande au temps t , qui lors de l'apprentissage, est appliquée à l'entrée de commande du modèle, et qui, lors de l'utilisation, est appliqué à l'entrée du système. L'ensemble (contrôleur + modèle placés en série) constitue un réseau de neurones non bouclé qui admet pour sortie $y(t+1)$.

L'apprentissage s'effectue en minimisant la différence entre l'état désiré $y^d(t+1)$ et la sortie du système $y(t+1)$. Seuls les paramètres du contrôleur (poids et biais) sont variables et modifiés par le processus d'apprentissage.

En effet, si le système est supposé décrit par l'équation différentielle non linéaire suivante:

$$y(t + 1) = f [y(t), \dots, y(t - n_y + 1); u(t), \dots, u(t - n_u + 1)] \quad (II-18)$$

Pour une sortie désirée $y^d(k + 1)$, la commande $u(k)$ délivrée par la commande est la suivante :

$$y(t + 1) = \hat{f}^{-1} [y(t), \dots, y(t - n_y + 1); u(t - 1), \dots, u(t - n_u + 1); y_d(t + 1)] \quad (II-19)$$

Où \hat{f}^{-1} est une approximation de la fonction inverse de f .

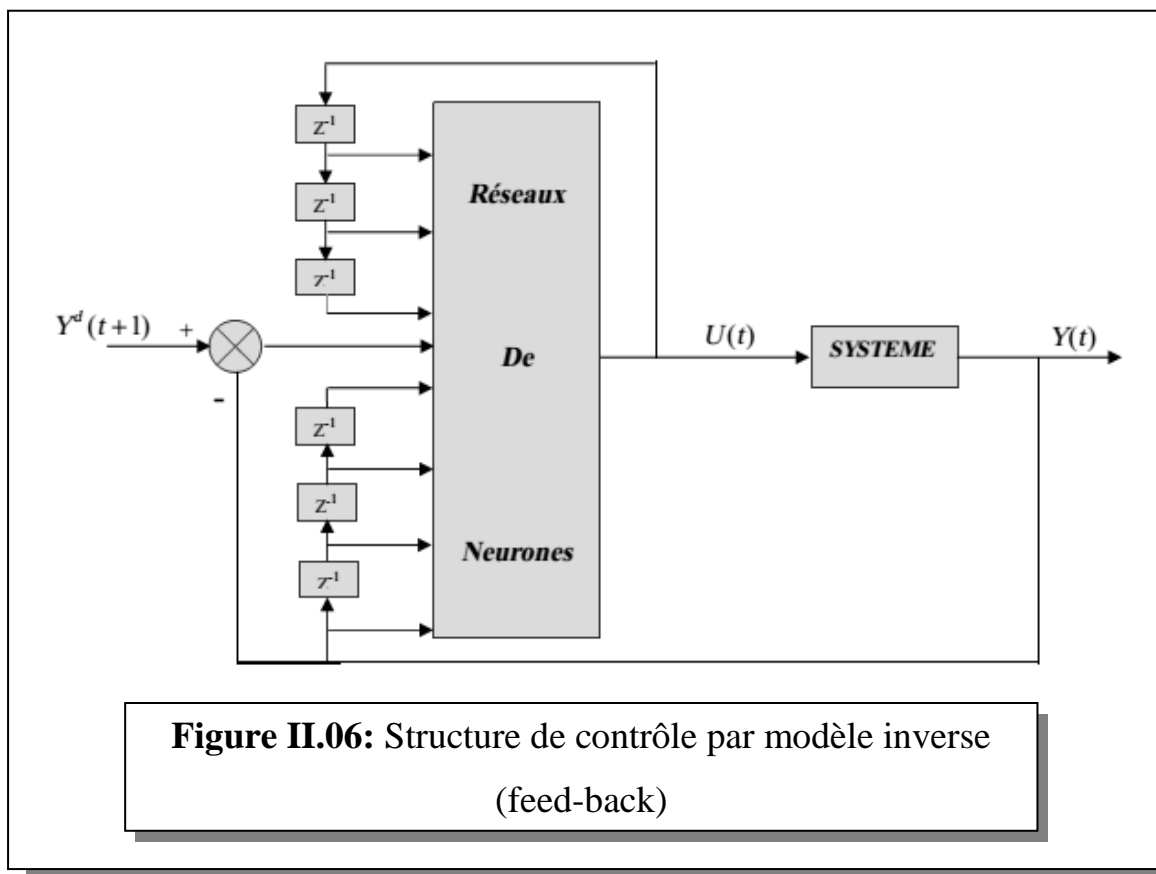
Cette méthode ne donne de bons résultats qu'aux problèmes simples où l'objectif peut s'exprimer instantanément en fonction de l'état $y(t)$, sinon cette méthode ne peut être mise en œuvre.

Il est évident que l'efficacité de cette approche dépend de la qualité, de fidélité et de la capacité de génération du modèle inverse.

L'inconvénient de ce type de contrôle est qu'il souffre de robustesse [13][14].

II. 3.3.1.2. Contrôleur Feed-back:

Ce contrôleur est utilisé surtout comme contrôleur dynamique des systèmes linéaires qui est caractérisé par sa robustesse. Son utilisation est parallèle à celle des régulateurs connus dans la commande des systèmes linéaires. C'est à dire que son entrée est le signal d'erreur (différence entre la consigne et la mesure), ou la sortie d'un régulateur classique PI, alors que son signal de sortie et la commande à appliquer au système, comme la montre la figure (II.6)

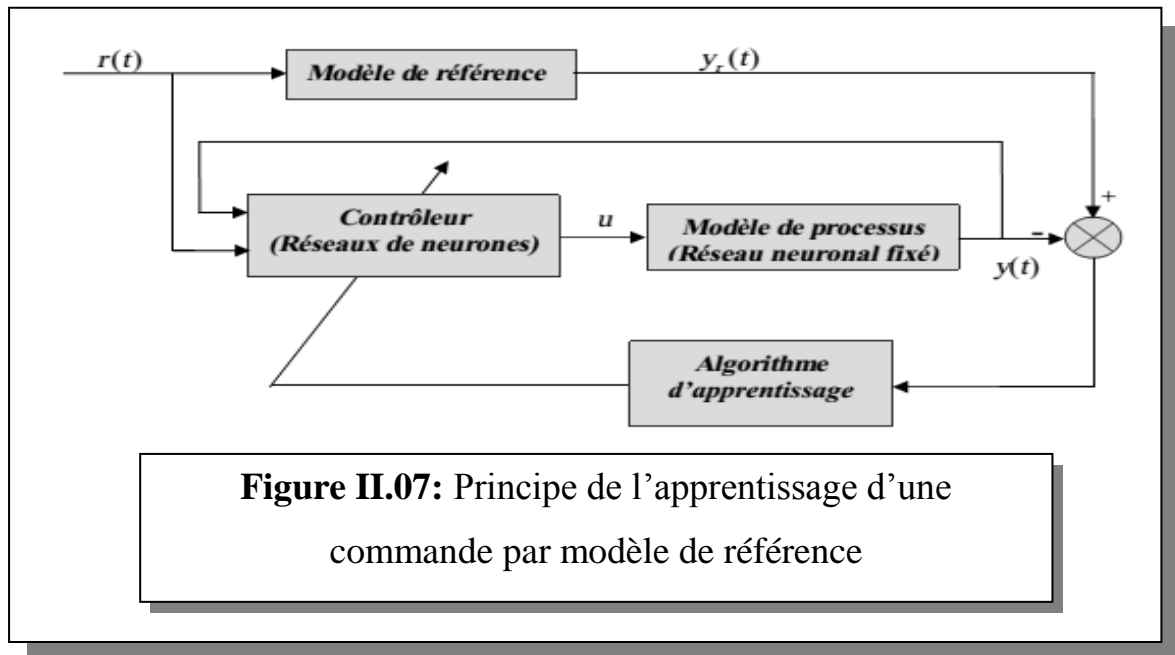


La différence entre les deux structures (Feed-back et Feed-Forward) réside dans l'entraînement des réseaux, le premier utilise l'apprentissage généralisé bien que le second utilise les deux types d'apprentissage (Généralisé et spécialisé) [13][14].

II.3.3.2. Commande par modèle de référence:

Dans ce type de commande, les performances désirées du système en boucle fermée sont spécifiées par un modèle de référence stable, défini par ses couples entrées/sorties ($r(k)$, $y_r(k)$).

Le principe de l'utilisation d'un modèle de référence est donné par la Figure (II.7) [4][17].



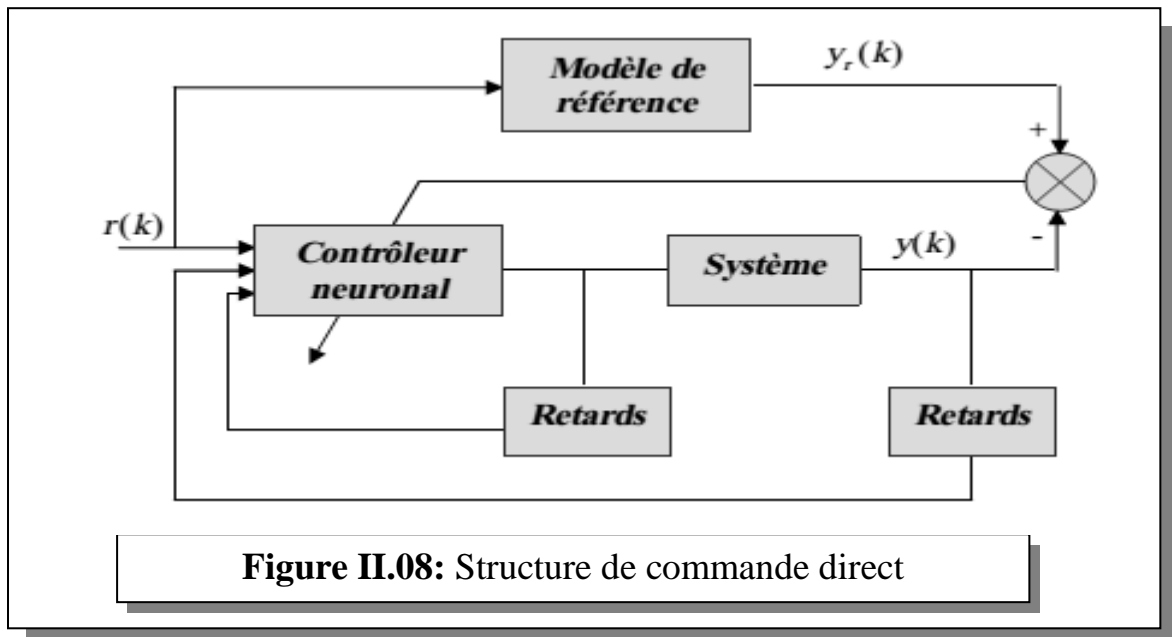
Le rôle du contrôleur est de commander la sortie du système de façon à minimiser l'écart entre la sortie du système et la sortie du modèle de référence.

L'erreur $y_r(k) - y(k)$ est utilisée pour ajuster les paramètres du contrôleur neuronal, la procédure d'entraînement force le contrôleur à se comporter comme un modèle inverse réglé dans le sens est défini par le modèle de référence (dans la commande par inversion du modèle, un modèle de référence est utilisé implicitement, il se réduit à un simple retard).

Deux structures de contrôle en utilisant un modèle de référence ont été proposées par Narendra et Parthasarathy ; la structure de contrôle direct et la structure de contrôle indirect.

II.3.3.2.1. Structure de commande direct

La structure de commande direct est illustrée dans la Figure (II.8). Dans ce cas la sortie du contrôleur est au même temps le signal de commande pour le système réel. Les entrées du contrôleur sont des sorties retardées du système réel et des sorties retardées du contrôleur lui-même [13][15].



II.3.3.2.2. Structure de commande indirect

Pour la structure de contrôle indirect, les paramètres du contrôleur sont ajustés de façon à minimiser l'écart entre la sortie du système et celle du modèle de référence. Le signal de commande généré par le contrôleur est utilisé comme une entrée pour le modèle neuronal et le système réel. L'erreur entre la sortie du système et celle du modèle sera utilisée pour ajuster les paramètres du modèle neuronal.

L'objectif du contrôleur est de générer les commandes nécessaires pour deux buts :

- Le premier est d'adapter les paramètres du modèle de façon à avoir une bonne identification, le second est que le système réel suit le comportement du modèle de référence.
- Notons enfin que les performances obtenues par le contrôle indirect sont plus intéressantes que celles obtenues par le contrôle direct.

La Figure (II.9) représente la structure de commande indirect [13][16].

CHAPITRE III :

**Application les techniques
de commande neuronal
sur la parabole de radar
de poursuite**

III.1. INTRODUCTION :

Le radar est d'abord un projecteur d'ondes électromagnétique tout comme un projecteur lumineux, il permet d'envoyer un faisceau d'ondes dans une direction déterminée. Lorsque le faisceau est intercepté par un obstacle (avion ou navire, par exemple), celui-ci diffuse les ondes comme un avion diffuse la lumière dans la nuit lorsqu'il se trouve dans le faisceau d'un projecteur. Il faut ensuite détecter l'onde réfléchi par l'obstacle.

D'une manière semblable, le radar permet de déterminer non seulement dans quelle direction se trouve un obstacle. Cette mesure se fait de la façon suivante :

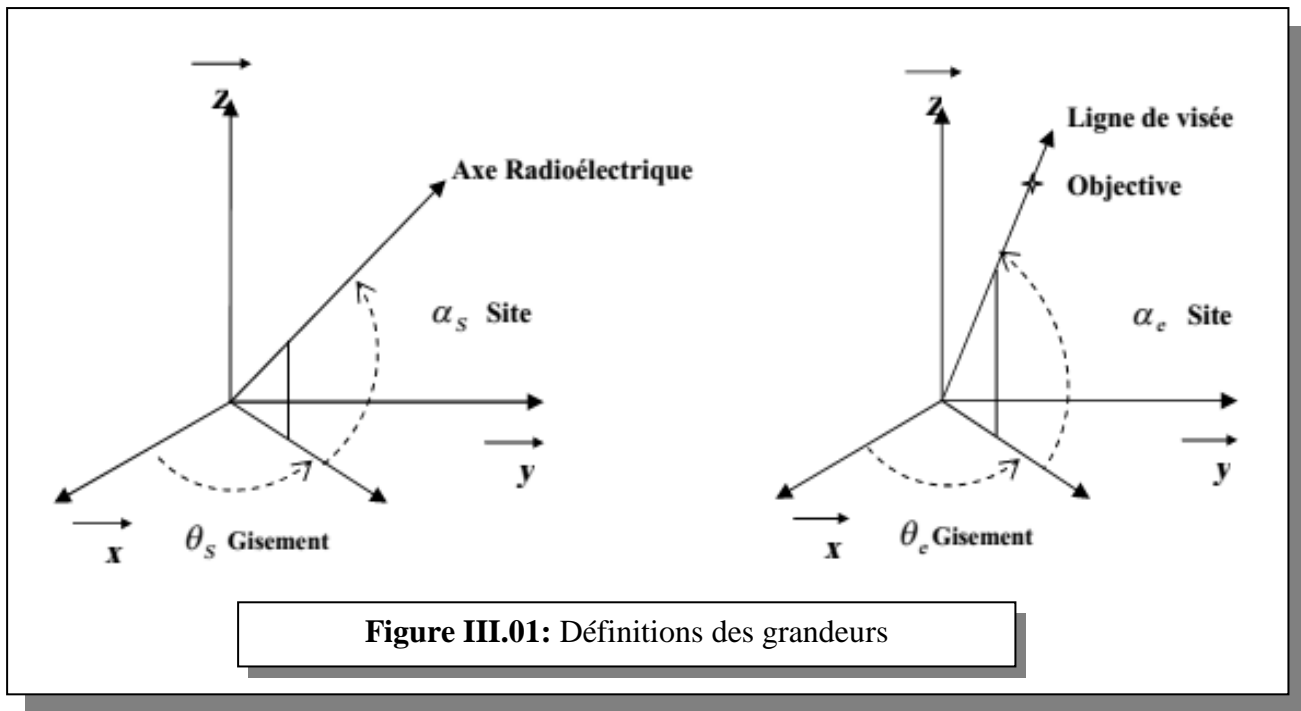
- un émetteur radioélectrique produit une impulsion brève et très puissante qui est ensuite rayonnée dans une direction déterminée par une antenne chaque fois que les ondes correspondantes rencontrent un obstacle, elles sont réfléchies (en partie ou en totalité).
- l'antenne utilisée précédemment pour l'émission, sert ensuite pour capter les ondes réfléchies par les obstacles.
- un récepteur sensible est ensuite employé pour amplifier ce signal et obtenir une image sur un écran [17].

III.2. Fonctionnement du système

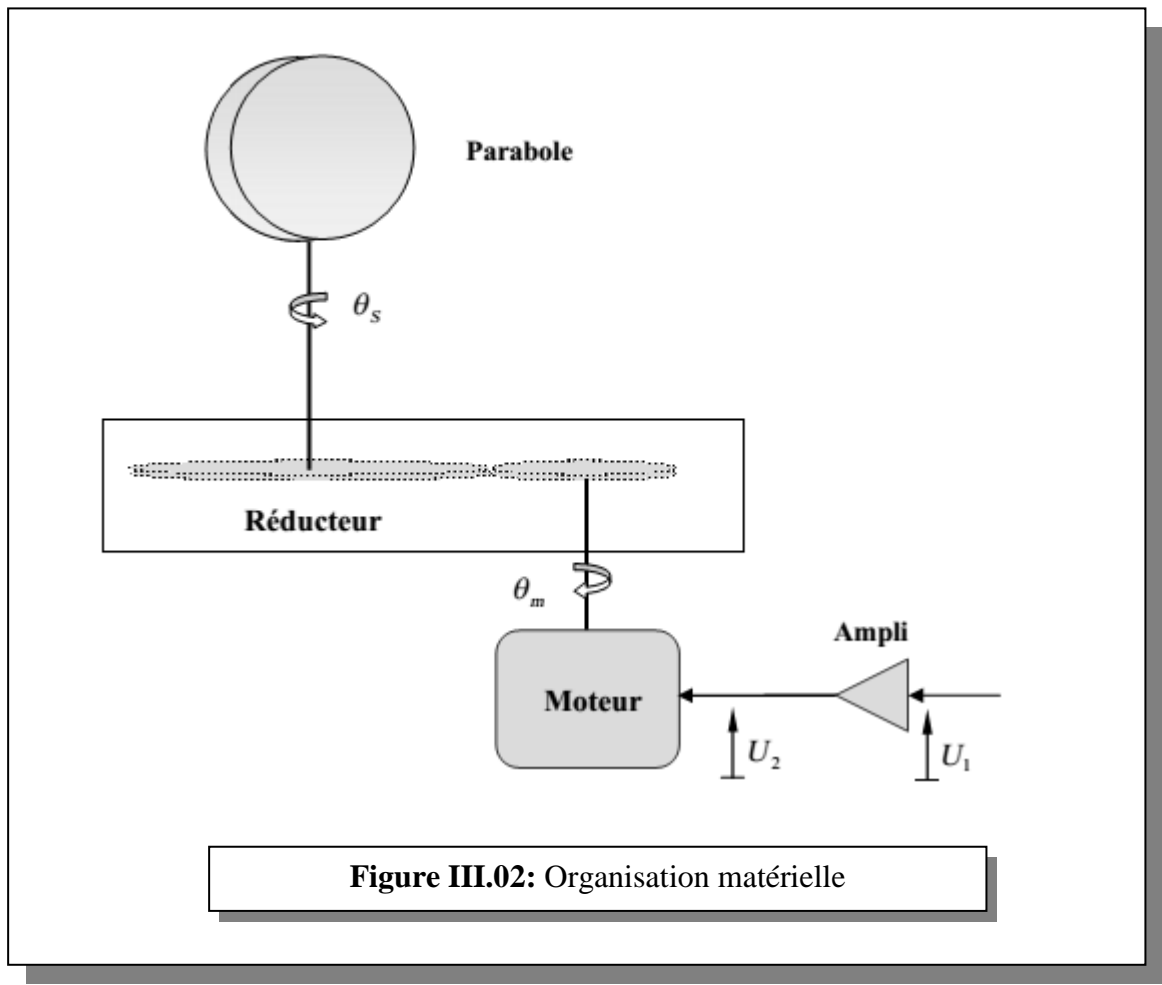
On envisage l'étude de l'asservissement en position angulaire. D'un radar de poursuite destiné à connaître avec précision la position d'un mobile évoluant dans l'espace aérien. Le système comporte une antenne parabolique émettant dans une direction précise appelée axe radioélectrique, cet axe est repéré par les angles de « site » et de « gisement » comme le montre la figure (III.1).

Des capteurs de position permettent d'avoir en permanence une image des angles θ_s et α_s . En présence d'une cible réfléchissant, l'écho reçu par la parabole dépend du « d'épointage angulaire » entre l'axe radioélectrique et la ligne de visée.

Le dispositif radar est capable de délivrer deux tensions proportionnelles aux écarts angulaires $(\theta_e - \theta_s)$ et $(\alpha_e - \alpha_s)$.



On se propose d'étudier l'asservissement en gisement de la tourelle port parabole présenté sur la figure (III.2). L'asservissement en site se fera sur le même principe.



III.2.1. Analyse et performance du système asservi

On admettra les hypothèses suivantes :

Le frottement est négligé.

L'inductance de l'induit du moteur est négligée.

La variation de site ne modifie pas le moment d'inertie autour de l'axe des Z.

Les valeurs numériques :

inertie de l'antenne (+ roue du réducteur) : $J_a = 15000 \text{ Kg m}^2$.

inertie du moteur (+ roue du réducteur) : $J_m = 5.10^{-3} \text{ Kg m}^2$.

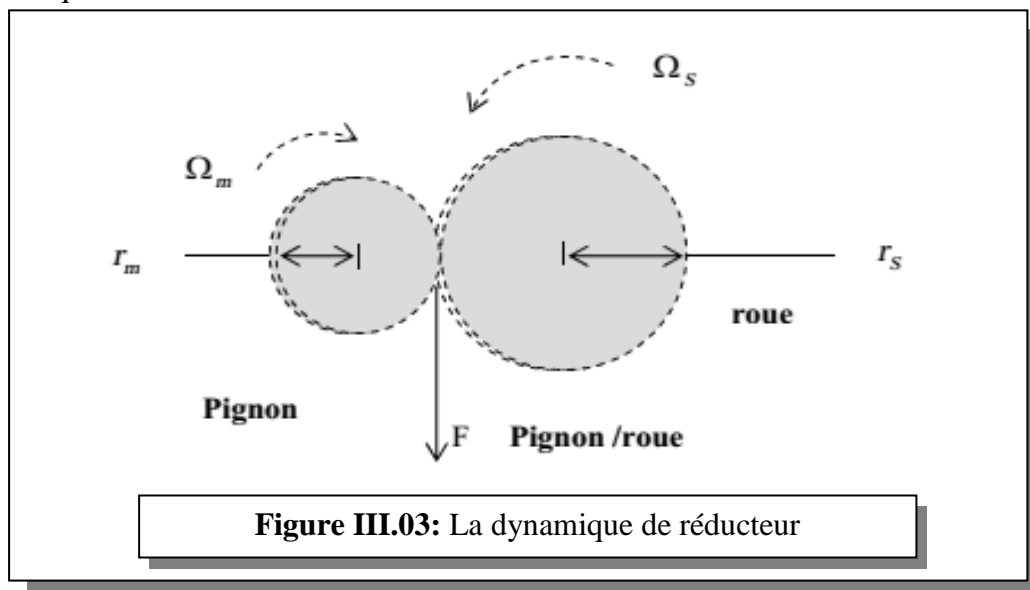
Rapport de réduction : $r = 1000$; $\theta_m/\theta_s = r$

Coefficient de vitesse du moteur : $k_m = 0,5 \text{ V/ (rad/S)}$

Résistance de l'induit du moteur : $R = 0,5 \Omega$

Coefficient d'amplification de puissance : $A = 10$; $U_2/U_1 = A$

Le calcul de l'inertie total ramenée au niveau de l'arbre moteur fait appel aux notions de bases de la dynamique.



On note F la force exercée par le pignon sur la roue. Le couple exercé sur la roue est :

$$C_s = F \cdot r_s = J_a \cdot \frac{d\Omega_s}{dt} \quad (\text{III-1})$$

D'où l'expression de la force :

$$F = \frac{J_a}{r_s} \cdot \frac{d\Omega_s}{dt} \quad (\text{III-2})$$

Du côté moteur on peut écrire l'équilibre :

$$C_m = F \cdot r_m = J_m \frac{d\Omega_m}{dt} \quad (\text{III-3})$$

Ce qui donne :

$$C_m = \frac{r_m}{r_s} \cdot J_a \frac{d\Omega_s}{dt} + J_m \frac{d\Omega_m}{dt} \quad (\text{III-4})$$

Les vitesses de rotation sont liées par la relation :

$$V = \Omega_S \cdot r_s = \Omega_m \cdot r_m \quad (\text{III-5})$$

Donc :

$$\Omega_S = \frac{r_m}{r_s} \Omega_m = \frac{1}{r} \cdot \Omega_m \quad (\text{III-6})$$

Et

$$\frac{d\Omega_S}{dT} = \frac{1}{r} \cdot \frac{d\Omega_m}{dt} \quad (\text{III-7})$$

En associant les relations (III-4) et (III-7) on obtient l'inertie J de l'ensemble sur l'arbre moteur.

$$J = J_m + \frac{J_a}{r^2} \quad (\text{III-8})$$

$$J = 0.005 + \frac{1500}{(1000)^2} = 0.02 \text{kgm}^2 \quad (\text{III-9})$$

III.2.2. Fonction de transfert

Dans le domaine temporel on a :

$$U_2(t) = R \cdot i(t) + L \frac{di(t)}{dt} + e(t) \quad (\text{III-10})$$

$$C_m(t) = J \cdot \frac{d\Omega_m(t)}{dt} \quad (\text{III-11})$$

On passe au domaine de la place :

$$U_2(p) = (R + L \cdot p) \cdot I(p) + E(p) \quad (\text{III-12})$$

$$C_m = J \cdot p \Omega_m(p) \quad (\text{III-13})$$

Les équations électromécaniques :

$$C_m(p) = K_m \cdot I(p) \quad (\text{III-14})$$

$$E(p) = K_m \cdot \Omega_m(p) \quad (\text{III-15})$$

De l'équation (III 13) et (III 14) on tire I (p)

$$I(p) = \frac{J \cdot p}{K_m} \Omega_m(p) \quad (\text{III-16})$$

On remplace la valeur de I (p) dans l'équation (III 12) on obtient :

$$U_2 = \left((R + L \cdot p) \frac{J \cdot p}{K_m} \right) \Omega_m(p) \quad (\text{III-17})$$

$$\frac{\Omega_m(p)}{U_2(p)} = \frac{1/K_m}{1 + (R + L \cdot p) \frac{J \cdot p}{K_m^2}} \quad (\text{III-18})$$

Puisque L est négligeable :

$$\frac{\Omega_m(p)}{U_2(p)} = \frac{1/K_m}{1 + \frac{R \cdot J}{K_m^2} \cdot p} \quad (\text{III-19})$$

On peut matérialiser le système sous la forme du schéma bloc donné par la figure (III .4).

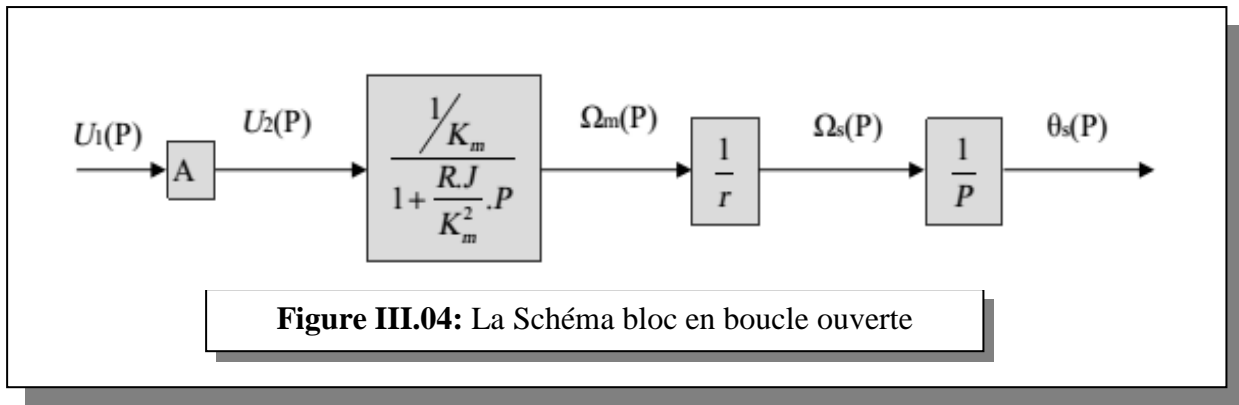


Figure III.04: La Schéma bloc en boucle ouverte

Avec : A=10 ; Km=0.5 ; RJ/Km² =0.04 ; r= 1000

Le système en boucle fermée peut se matérialiser par le schéma bloc donné par la figure (III.5).

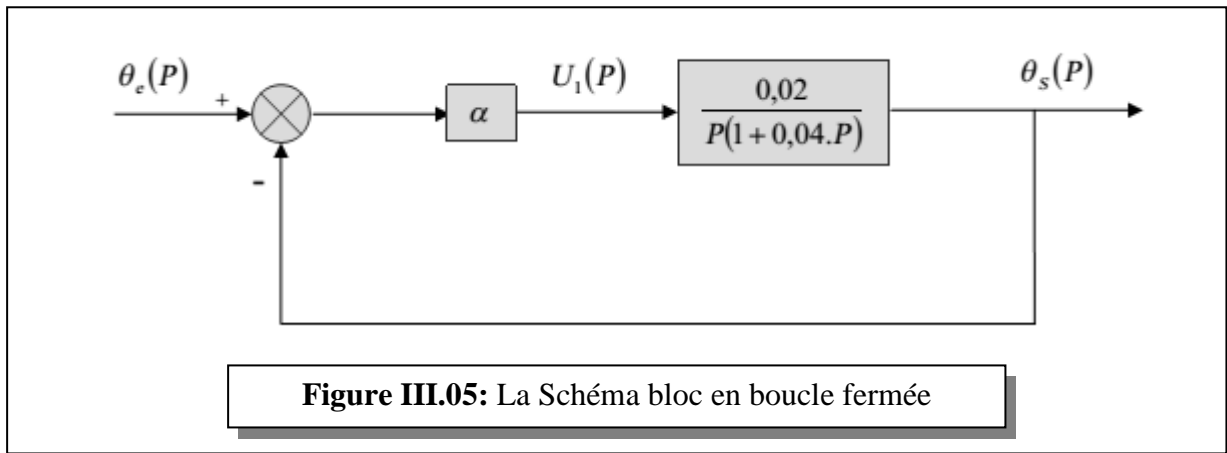


Figure III.05: La Schéma bloc en boucle fermée

On exprime la fonction de transfert :

$$F(p) = \frac{\theta_s(p)}{\theta_c(p)} = \frac{0,02 \cdot \alpha}{0,02 \cdot \alpha + p(1 + 0,04 \cdot p)} = \frac{1}{1 + \left(\frac{1}{0,02 \cdot \alpha}\right)p + \left(\frac{0,04}{0,02 \alpha}\right)p^2} \tag{I-20}$$

$$F(p) = \frac{K_f}{1 + 2Z_f \frac{p}{W_f} + \left(\frac{p}{W_f}\right)^2} \tag{I-21}$$

Avec : W_f : La pulsation propre.
 Z_f : Le facteur d'amortissement
 K_f : Le gain statique.

Par identification on obtient :

$$K_f = 1 ; \quad W_f = \sqrt{\frac{\alpha}{2}} ; \quad Z_f = \frac{1}{\sqrt{0,0032 \alpha}}$$

Pour que le temps de réponse à 5% après une excitation en échelon de position soit le plus faible possible il faut que Z_f = 0,07

Ce qui conduit aux valeurs : α = 625 v/ rad ; Z_f = 17,68 rad / s

Dans ces conditions : tr à 5 % = 3 , W_f = 0,17 s

III.3. Application le RNA sur la parabole de radar de poursuite:

On va appliquer les techniques de contrôle étudiées dans le chapitre II sur notre système (parabole de radar de poursuite).

III.3.1 Identification du modèle direct:

Pour identifier le modèle direct du système, on a utilisé la structure d'identification directe. Le modèle neuronal que nous avons choisi est :

- 01 couche cachée.
- 07 neurones dans la couche d'entrée.
- 05 neurones dans la couche cachée.
- Un seul neurone dans la couche de sortie.
- La fonction d'activation utilisée dans la couche cachée et couche d'entrée est la fonction

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

- La fonction d'activation utilisée dans la couche de sortie est la fonction linéaire $f(x) = x$
- Pas d'apprentissage $\mu = 0.05$.
- Moment $\beta = 0.0001$.

Nous disposons de 200 paires de mesures entrées-sorties (figures (III.06) et figure (III.07)) avec un pas d'échantillonnage de 0.1 seconde.

La sortie du modèle obtenue après 9735 itérations et la sortie du système pour une entrée :

$$u(k) = -180 \cdot \sin(2\pi \cdot k / 7)$$

Et l'erreur entre le modèle et le système sont représentées respectivement sur la figure (III.08) et figure (III.09).

On remarque que l'écart entre les deux sorties est négligeable $9,7 \cdot 10^{-8}$

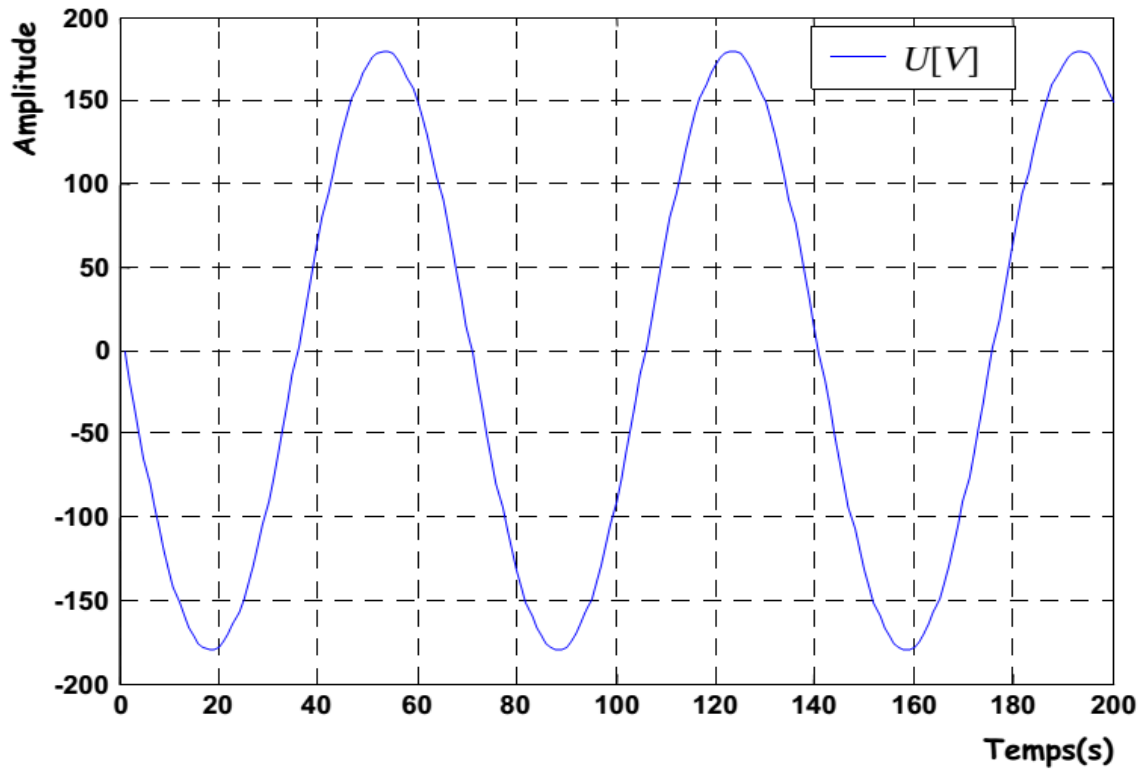


Figure III.06: L'Entrée du système

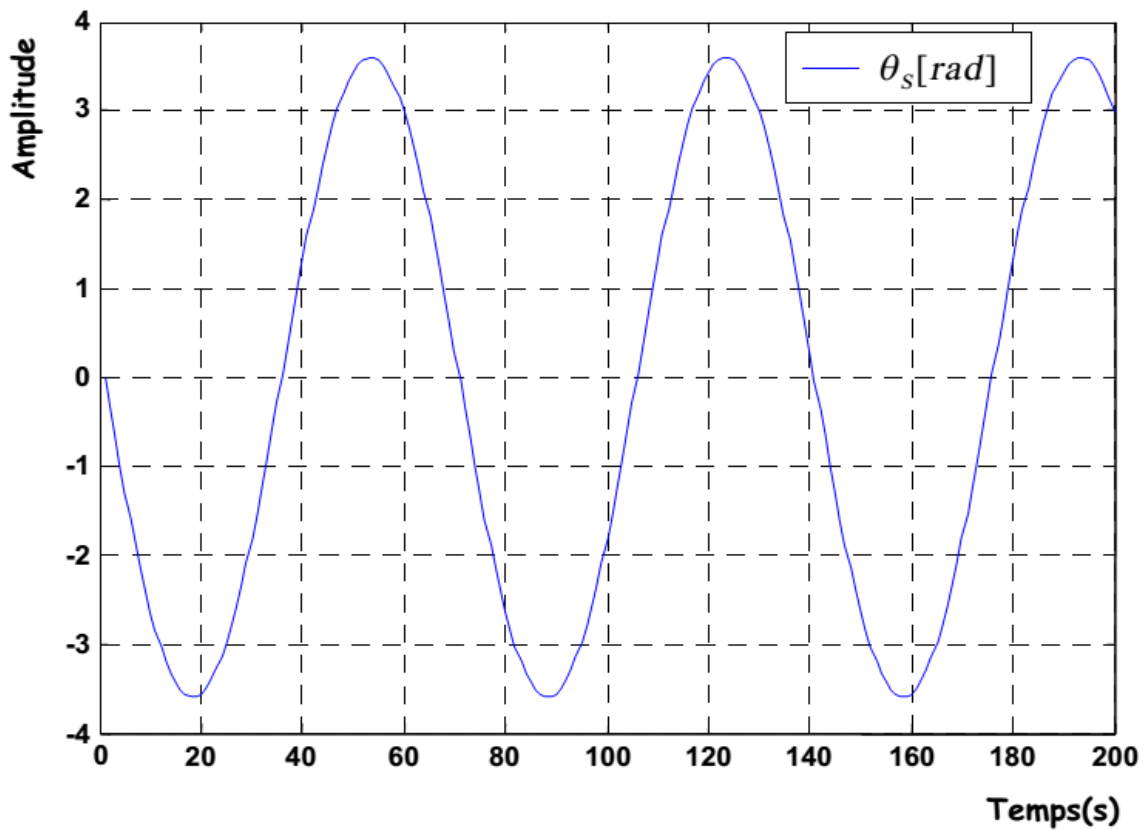
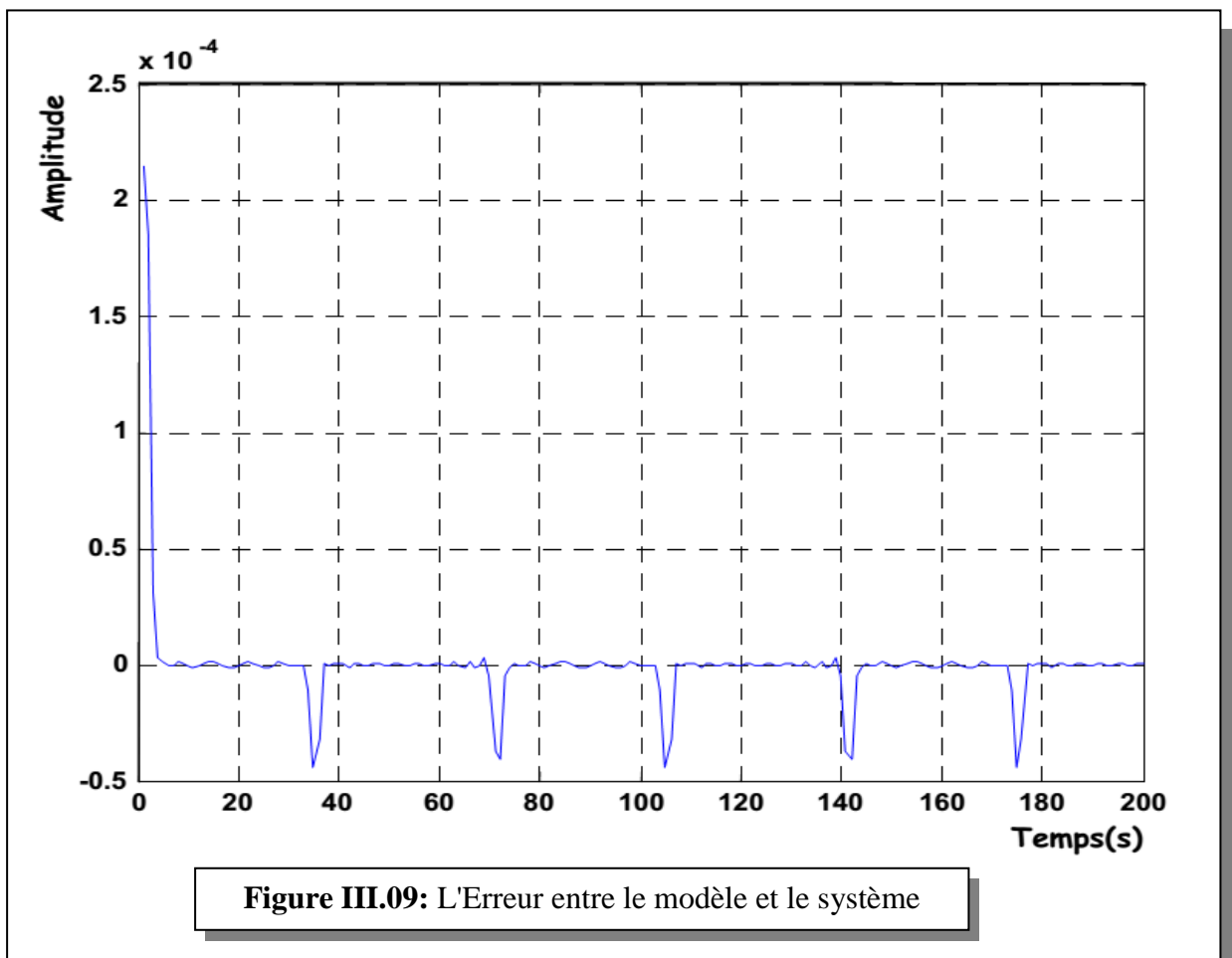
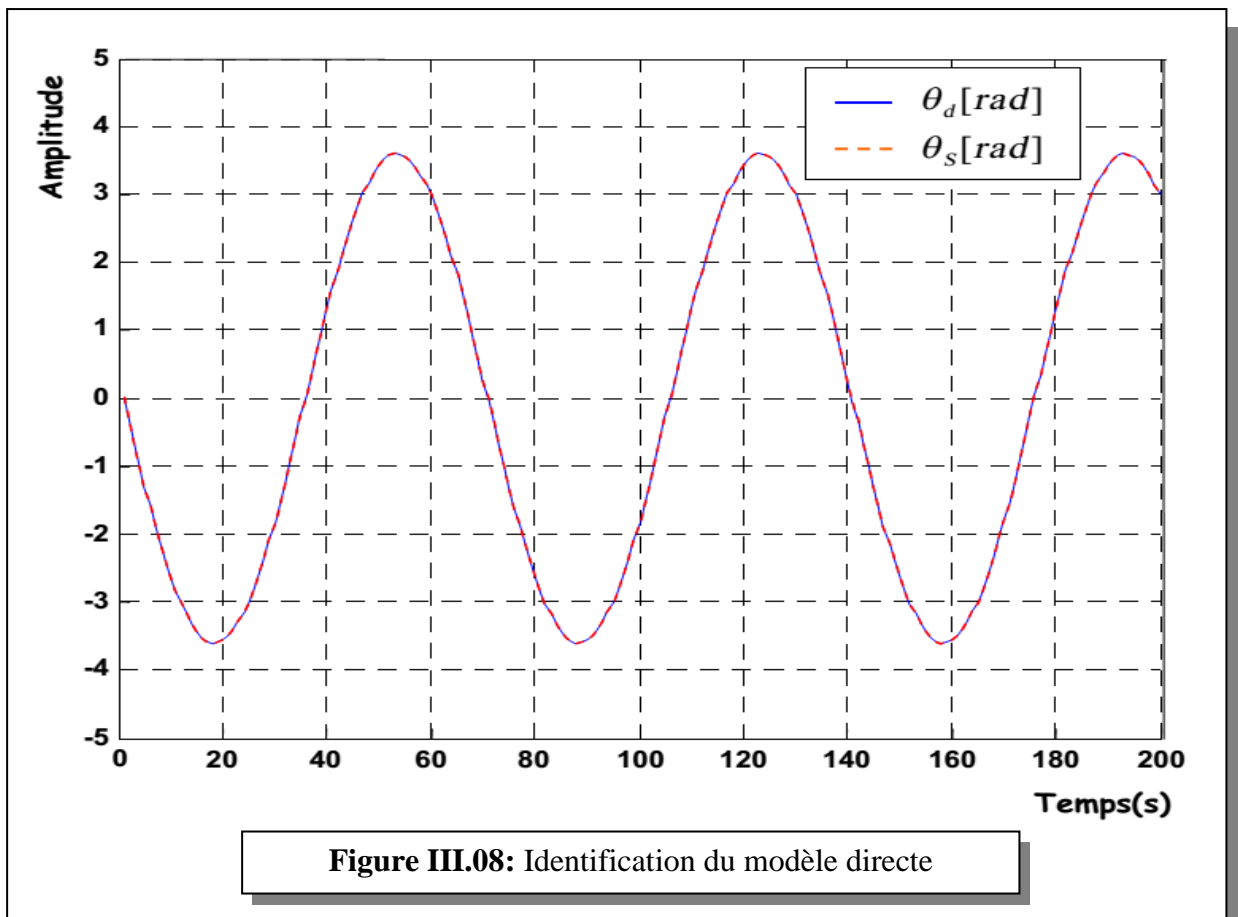


Figure III.07: La Sortie du système



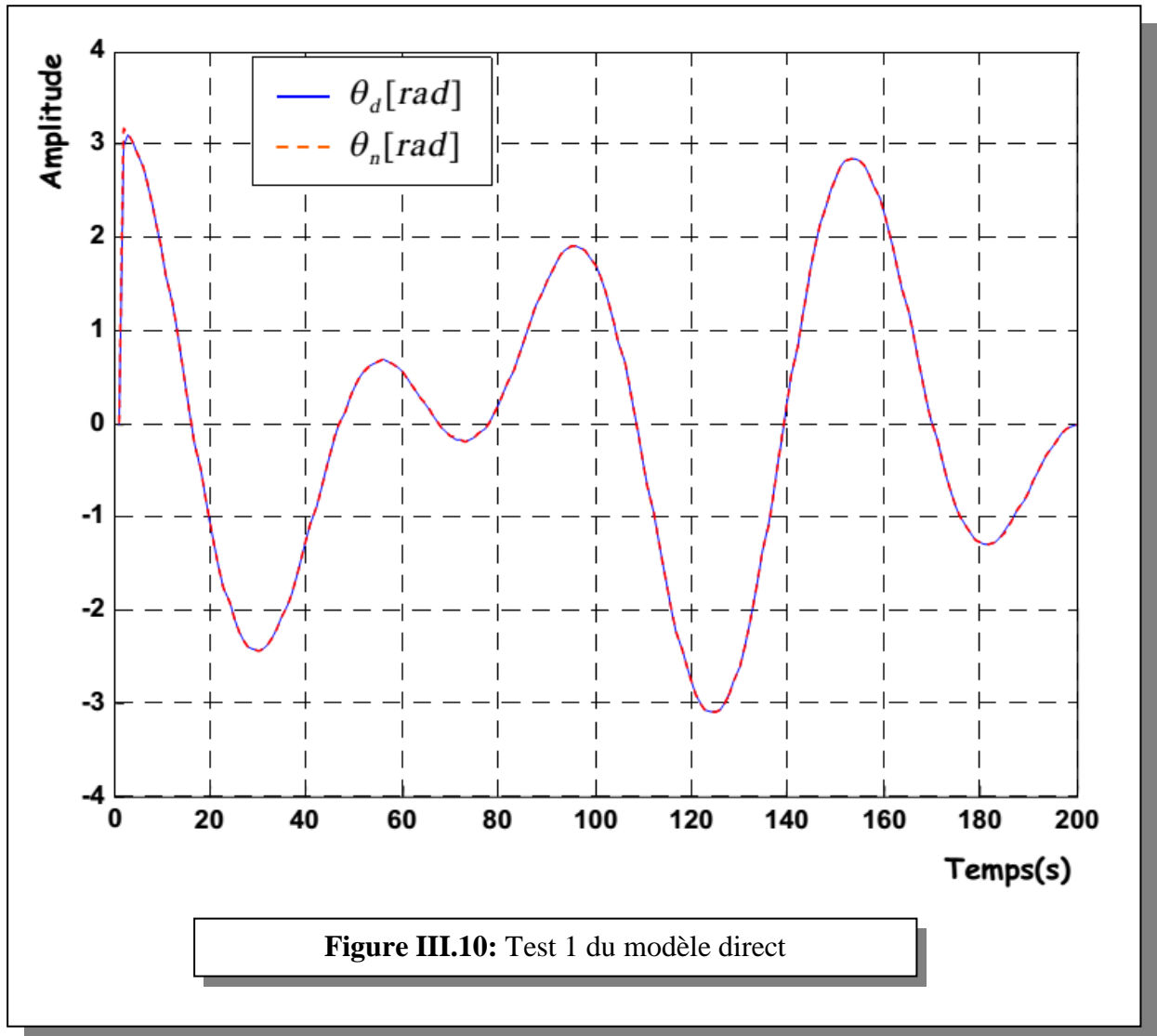
⇒ **test du modèle direct:**

Le modèle direct obtenu par le réseau a été testé par une autre entrée tel que :

$$u_1(k) = 80.(\cos(2.\pi.k / 5) + \cos(2.\pi.k / 8))$$

$$u_2(k) = 100.(\sin(2.\pi.k / 7) + \sin(2.\pi.k / 1.5))$$

- Le résultat du test 1 ($u_1(k)$) et l'erreur sont représentés par la figure (III.10) et figure (III.11)
- Le résultat du test 2 ($u_2(k)$) et l'erreur sont représentés par la figure (III.12) et figure (III.13)



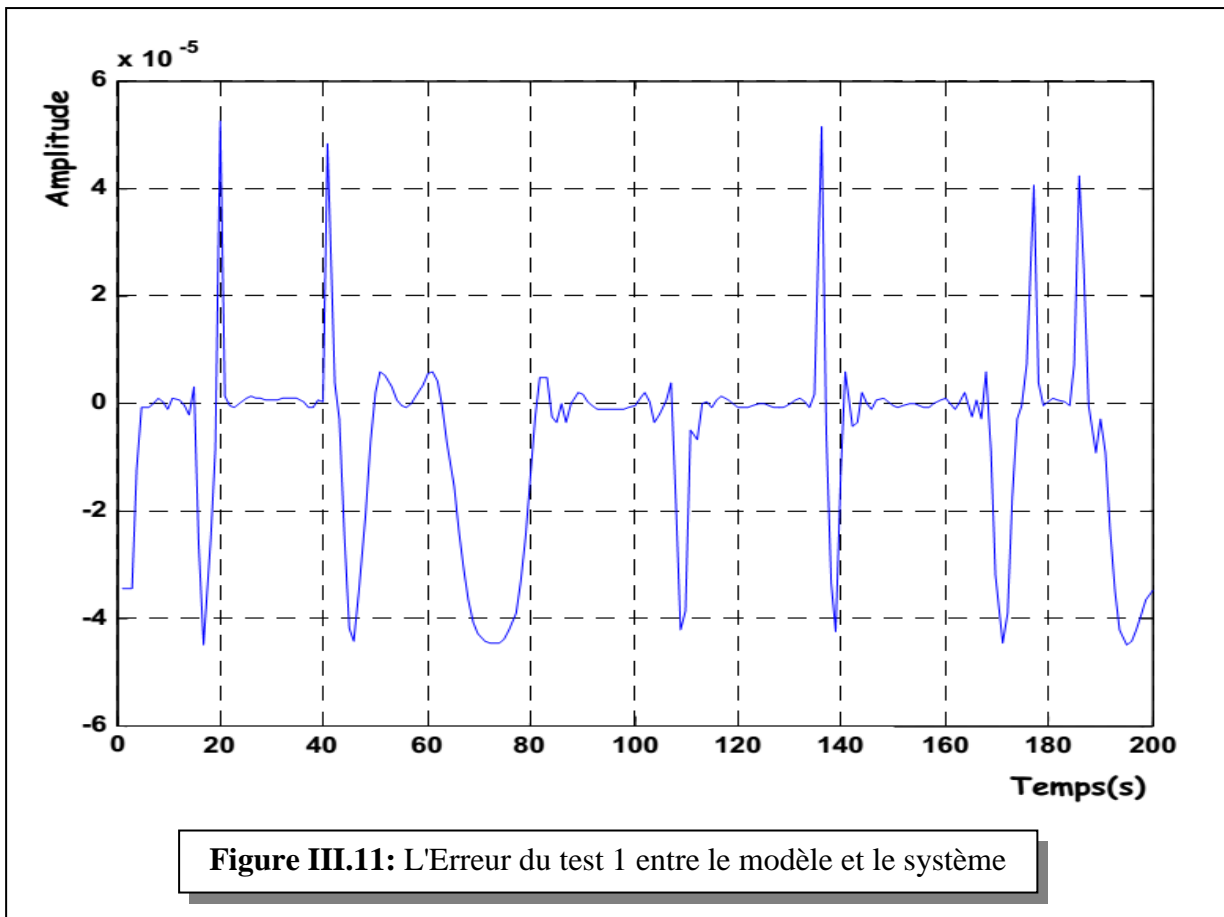


Figure III.11: L'Erreur du test 1 entre le modèle et le système

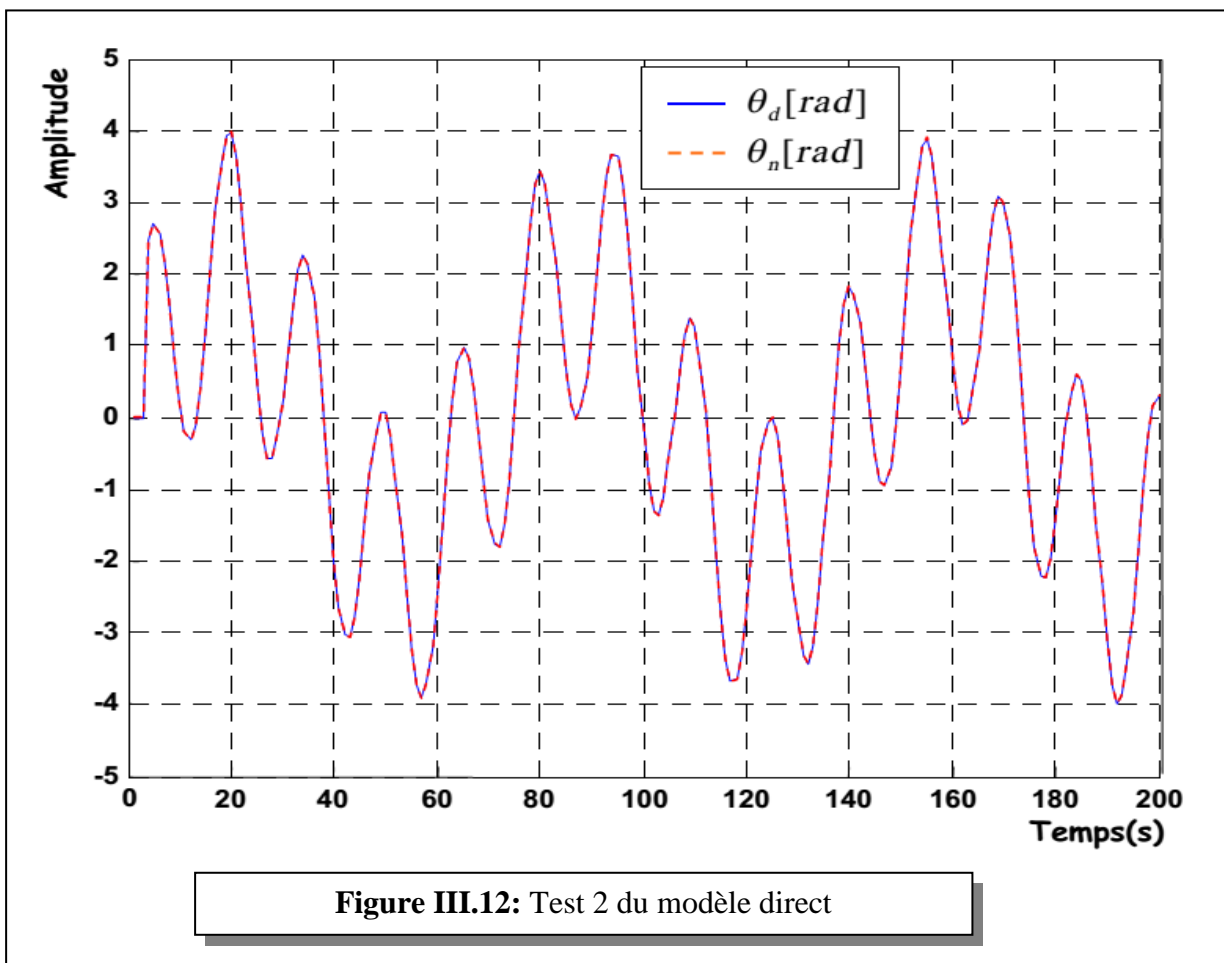
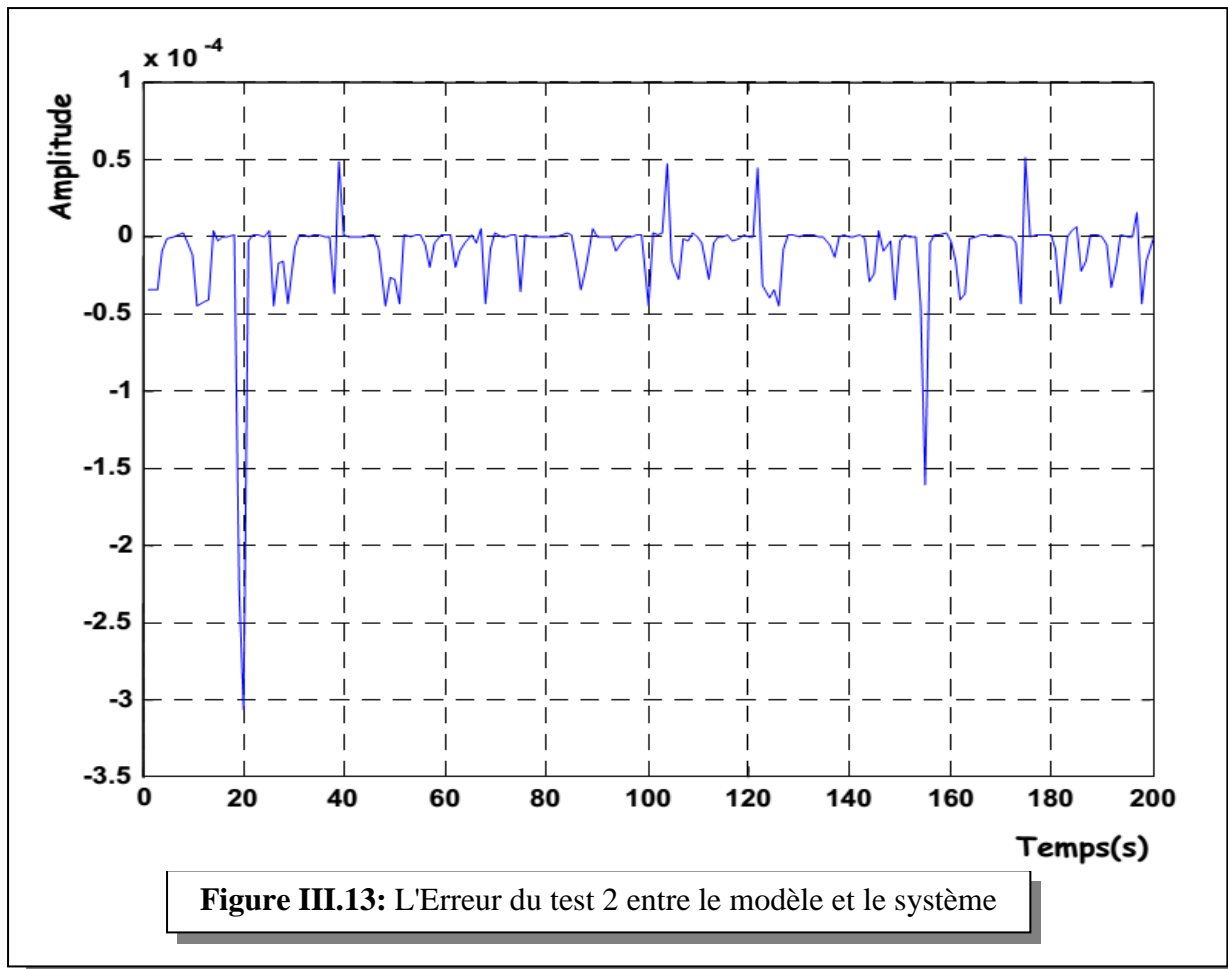


Figure III.12: Test 2 du modèle direct



III.3.2. Identification du modèle inverse

On va entraîner un réseau de neurones à approcher le modèle inverse du système $u_d(k)$, on utilise la structure d'apprentissage généralisé.

$$u_d(k) = -180 \cdot \sin(2 \cdot \pi \cdot k / 7)$$

Le modèle neuronal choisi comporte :

- 02 couches cachées.
- 07 neurones dans la couche d'entrée.
- 05 neurones dans la première couche cachée.
- 03 neurones dans la deuxième couche cachée.
- Un seul neurone dans la couche de sortie.
- La fonction d'activation utilisée dans la couche d'entrée et les deux couches cachées est la

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

- La fonction d'activation utilisée dans la couche de sortie est la fonction linéaire $f(x) = x$
- Pas d'apprentissage $\mu = 0.05$.
- Moment $\beta = 0.00001$.

La sortie du système est appliquée au modèle neuronal, l'erreur entre la commande et la sortie du modèle est utilisée pour l'ajustement des poids.

Après 10000 itérations, l'entrée du système et la sortie de modèle, et l'erreur entre la sortie de modèle et l'entrée de système sont représentés respectivement par la figure (III.14) et figure (III.15)

On remarque que l'écart entre les deux sorties est négligeable $3,19 \cdot 10^{-5}$

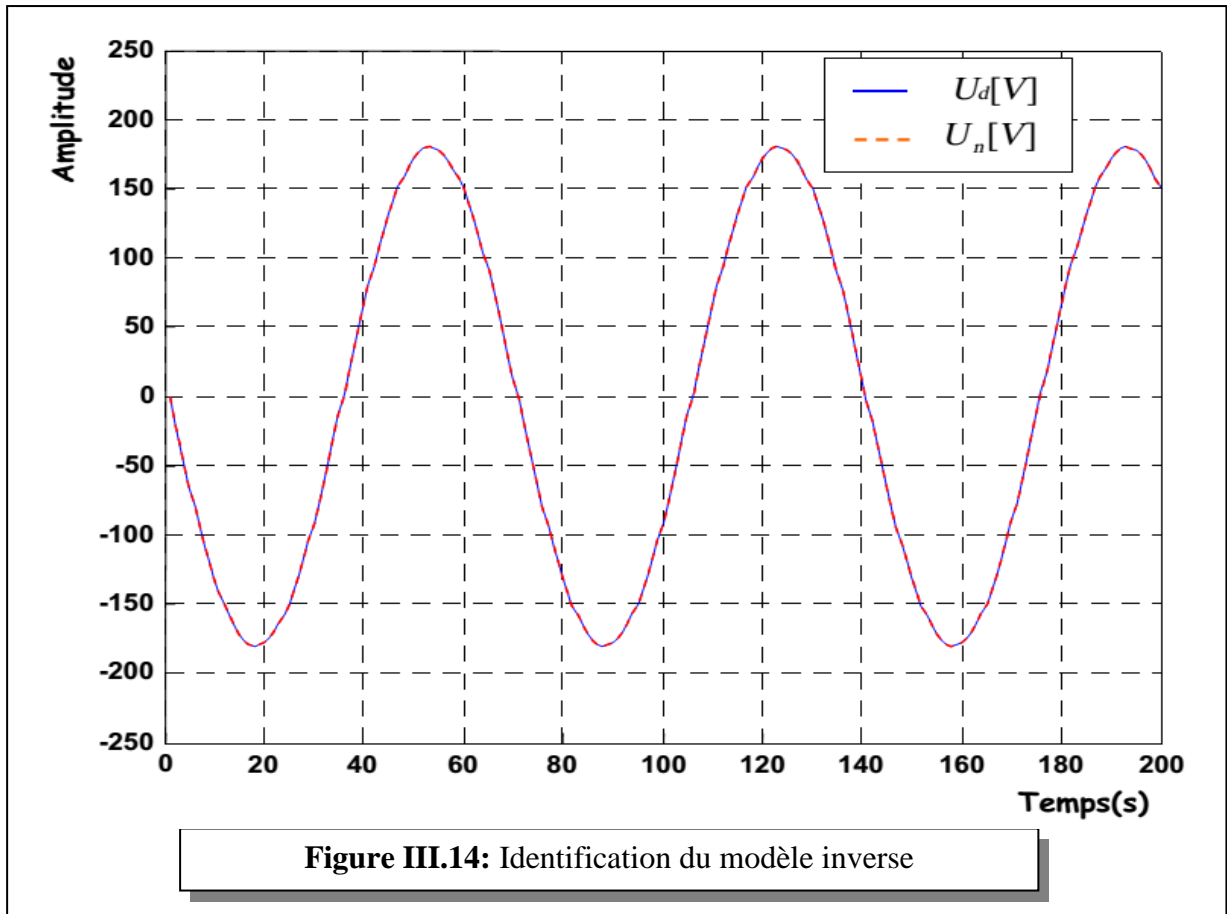


Figure III.14: Identification du modèle inverse

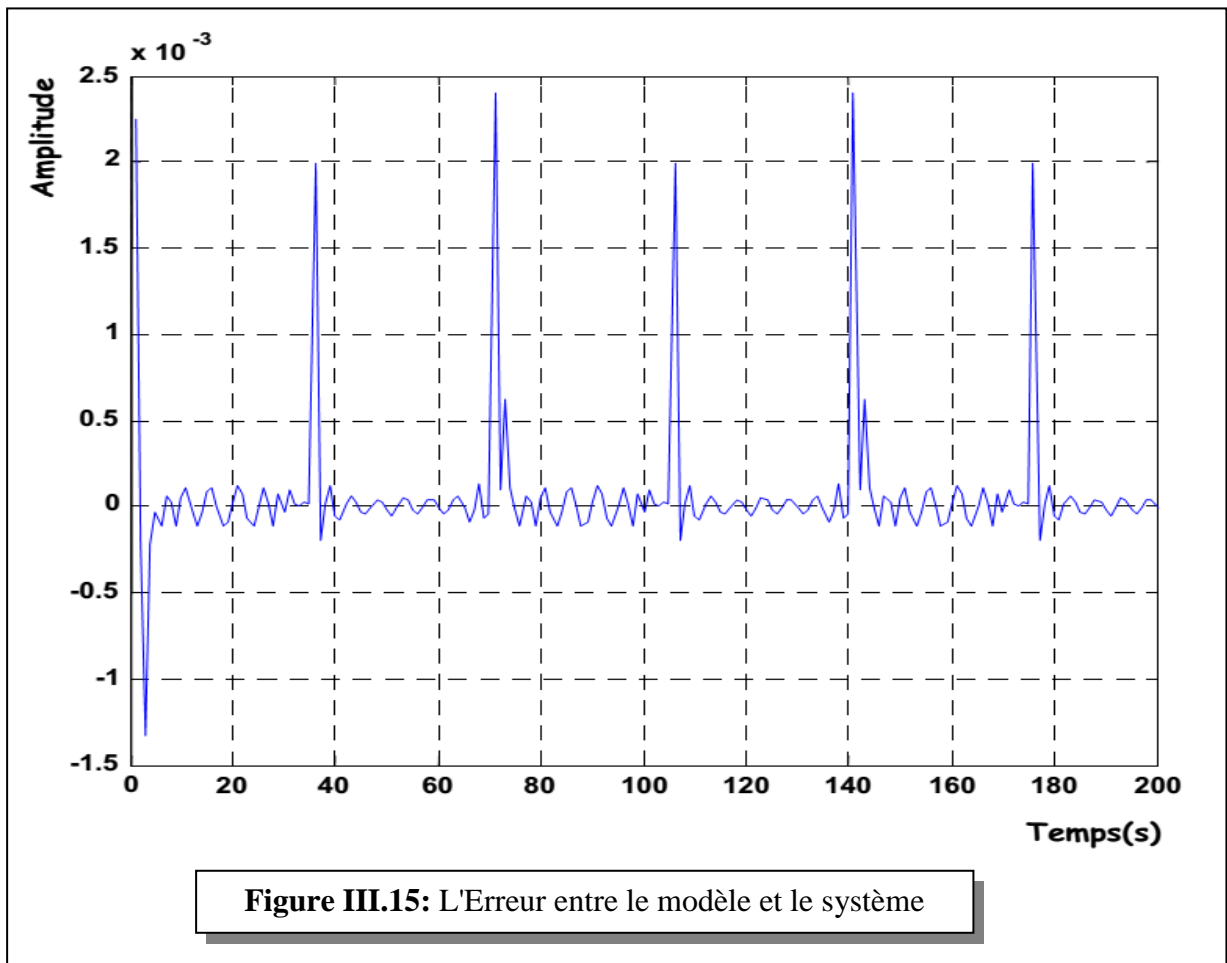


Figure III.15: L'Erreur entre le modèle et le système

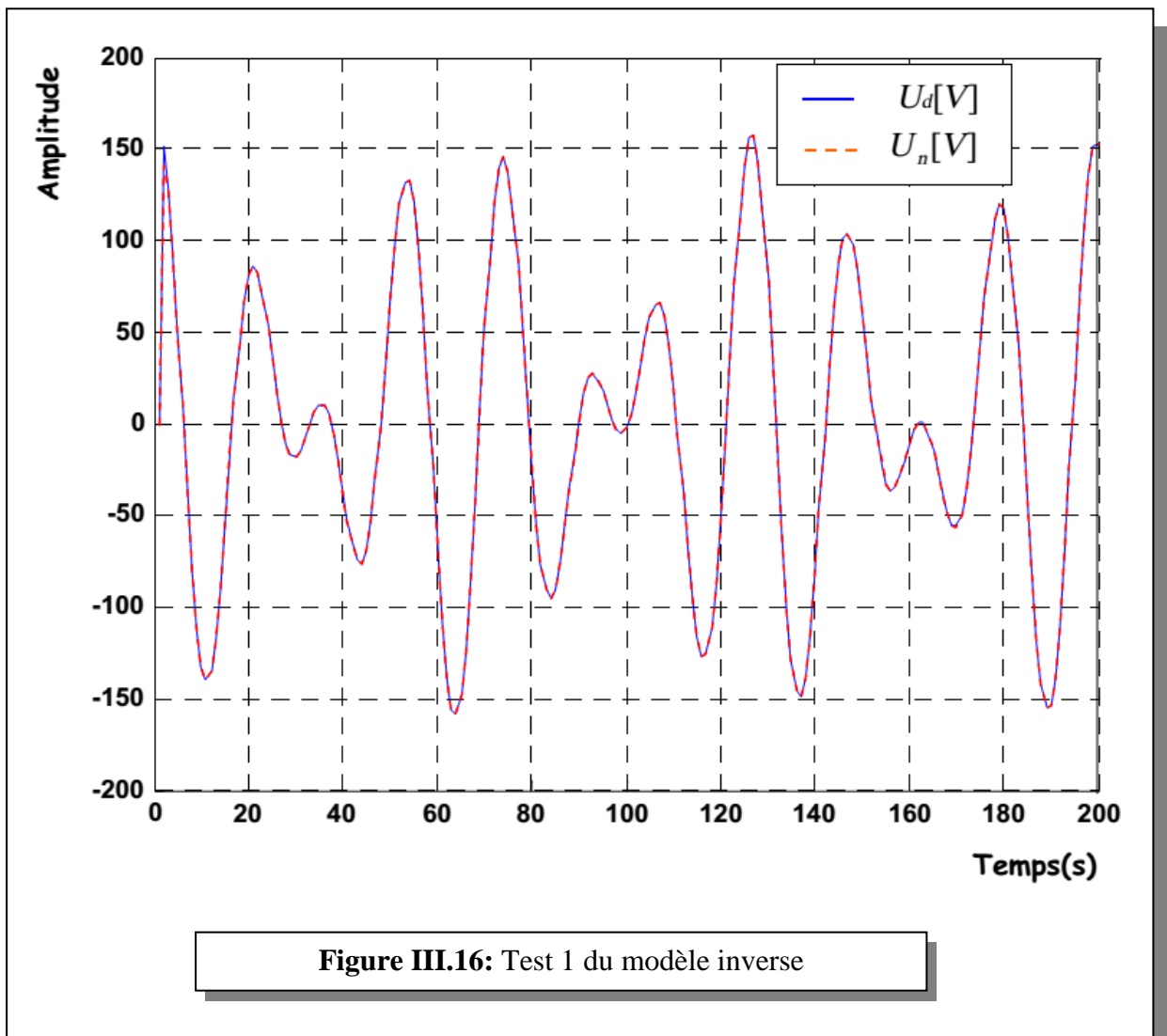
⇒ **test du modèle inverse:**

Le modèle inverse a été testé, en appliquant à l'entrée du modèle la sortie du système tel que la commande appliquée au système $u(k)$:

$$u_1(k) = 80 \cdot (\cos(2\pi k / 5) + \cos(2\pi k / 8))$$

$$u_2(k) = 100 \cdot (\sin(2\pi k / 5) + \sin(2\pi k / 2.5))$$

- Le résultat du test 1 ($u_1(k)$) et l'erreur sont représentés par la figure (III.16) et figure (III.17).
- Le résultat du test 2 ($u_2(k)$) et l'erreur sont représentés par la figure (III.18) et figure (III.19).



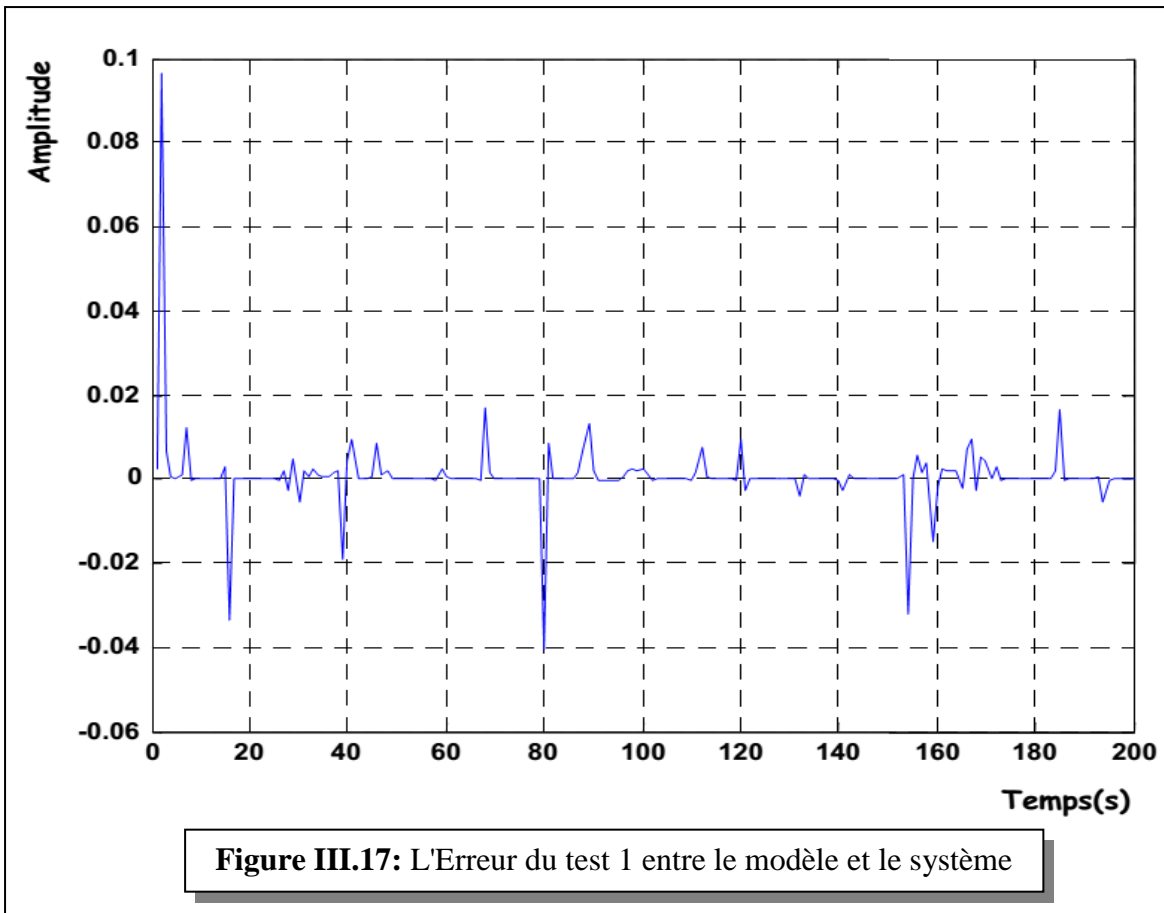


Figure III.17: L'Erreur du test 1 entre le modèle et le système

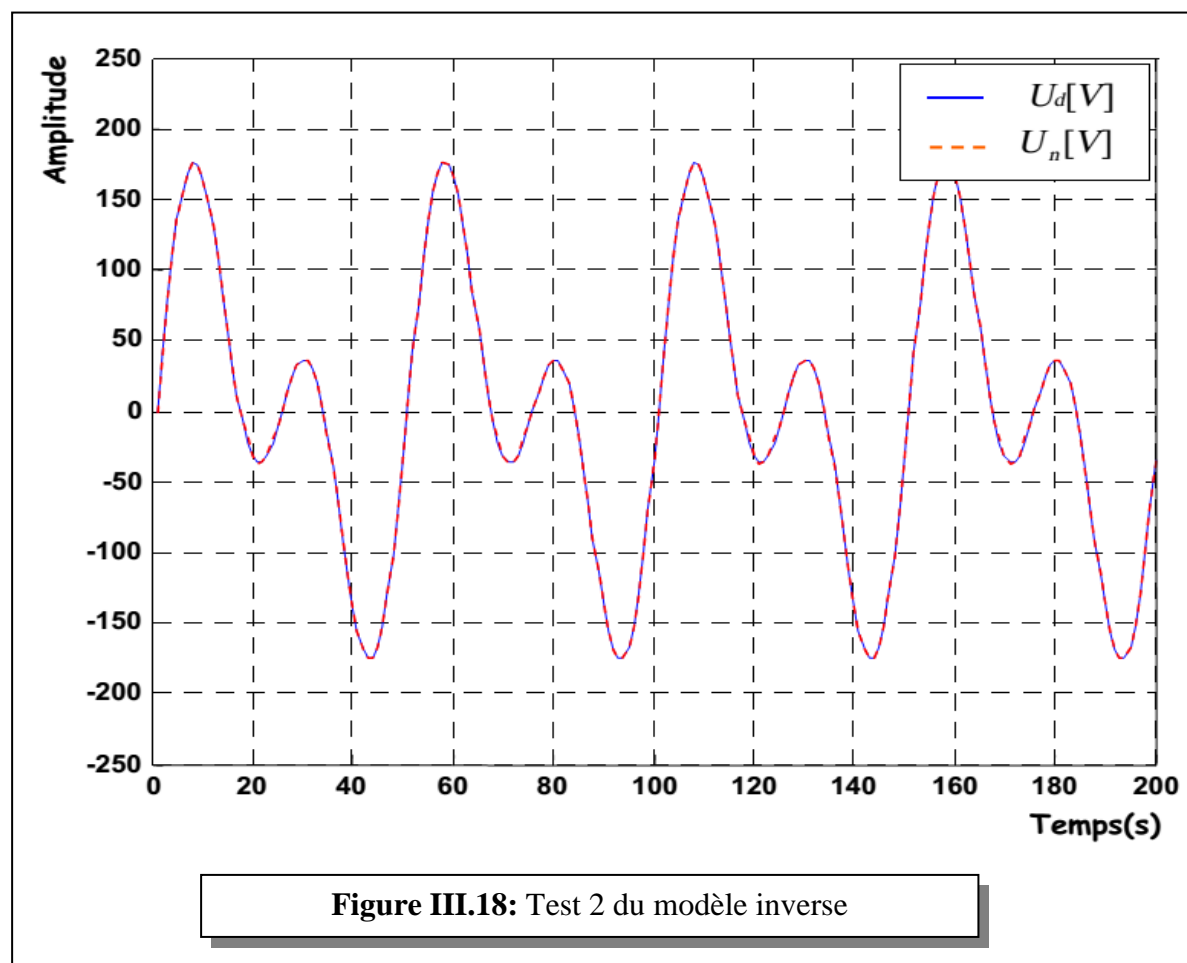
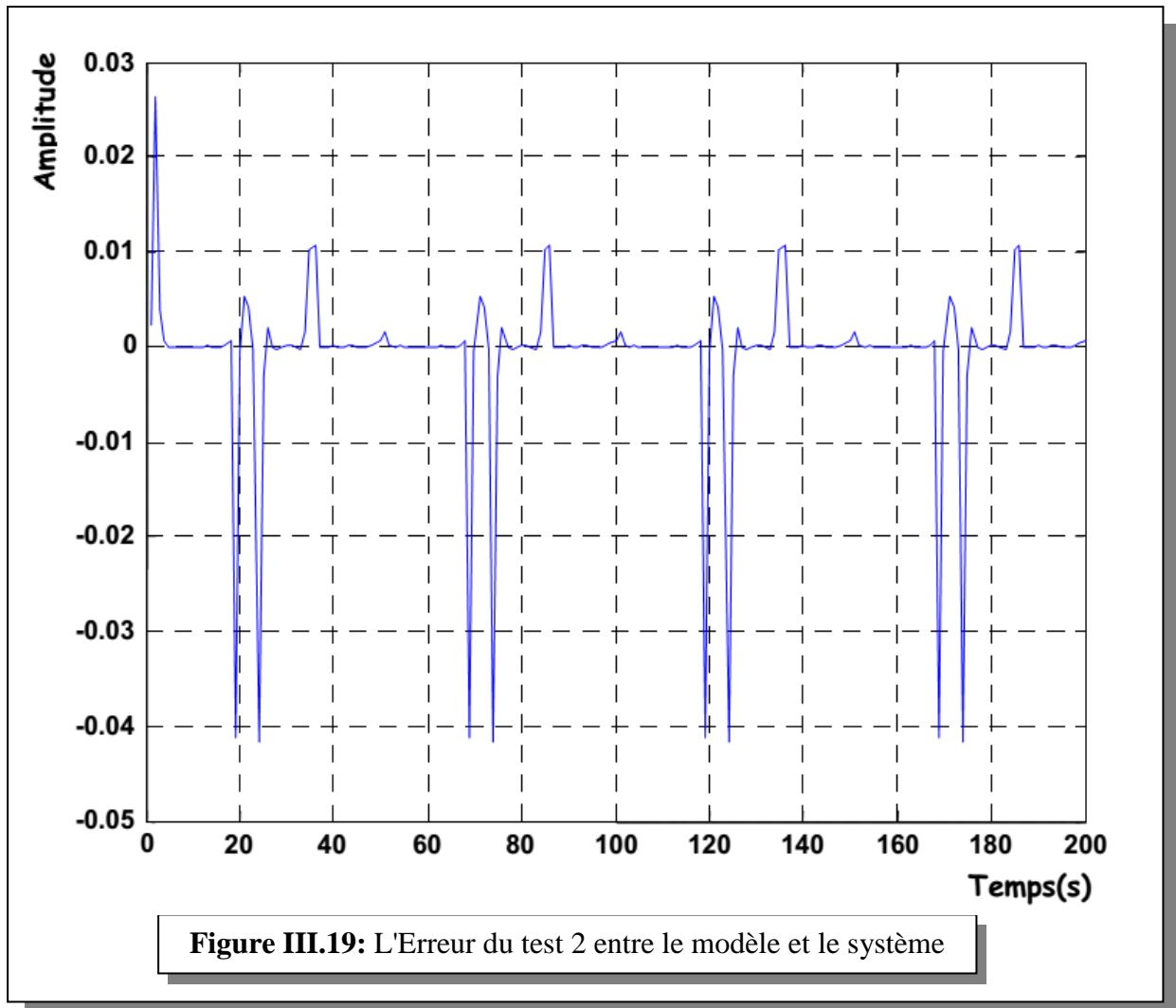


Figure III.18: Test 2 du modèle inverse



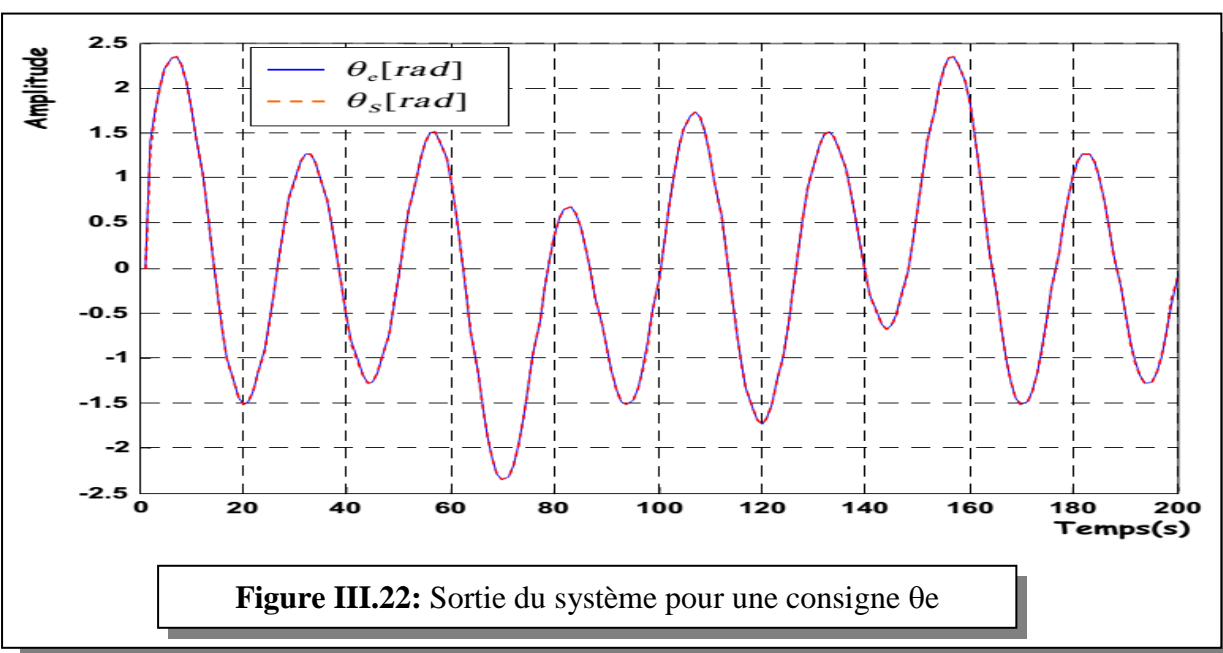
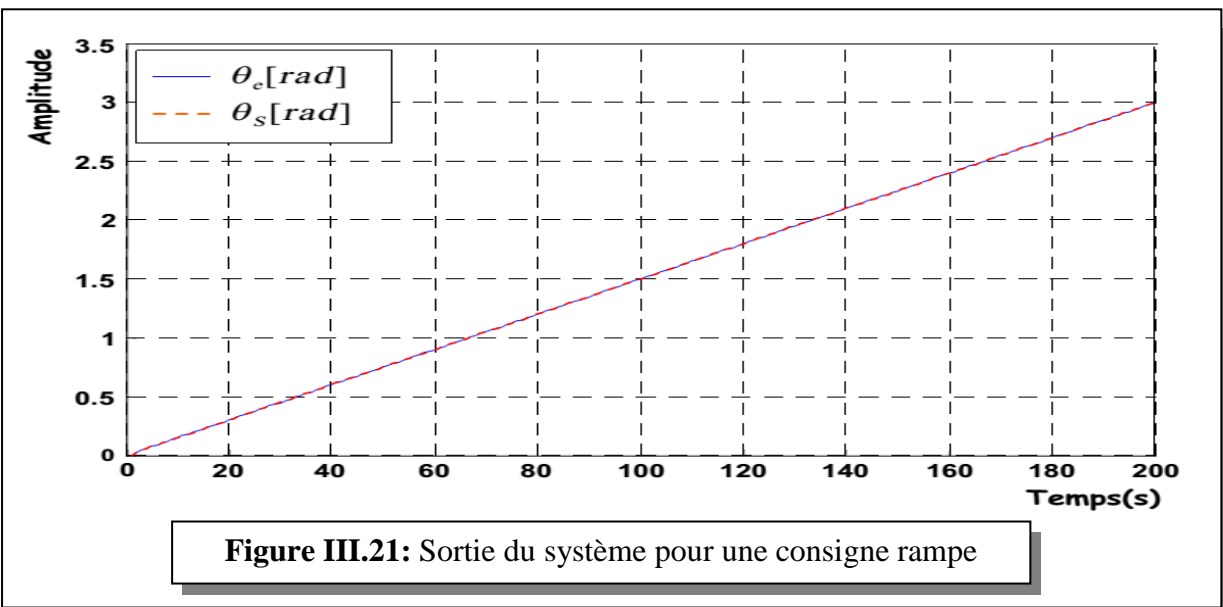
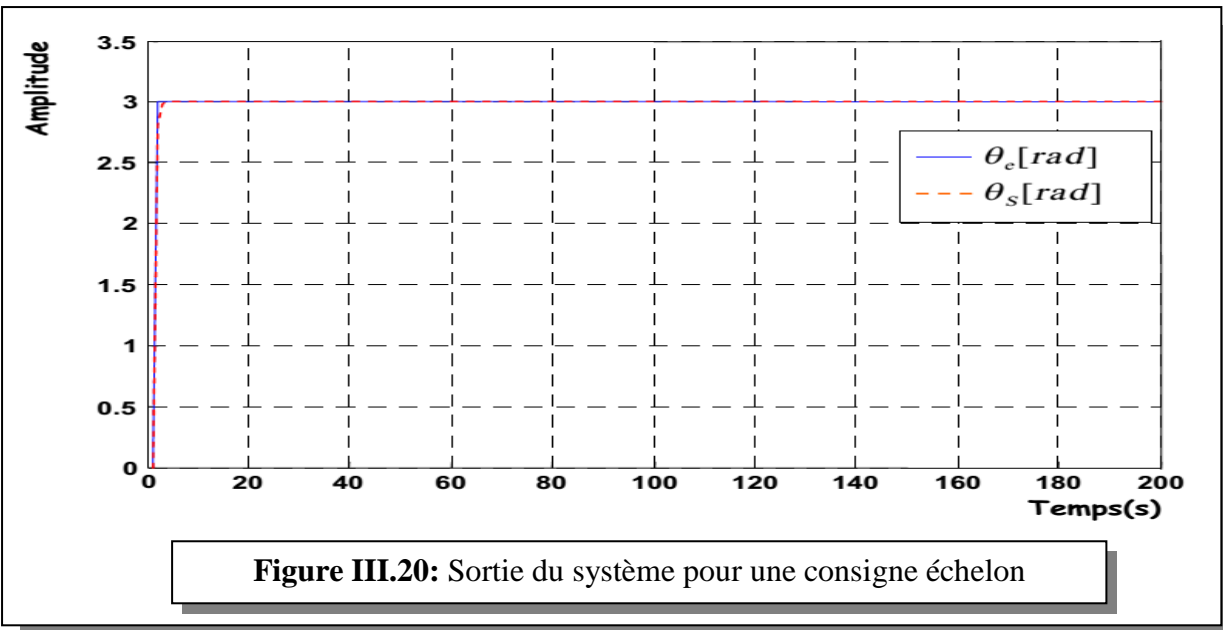
III.3.3. Commande par modèle inverse

Le système peut être commandé avec la structure inverse traitée dans le chapitre II. Le modèle inverse du système obtenu est placé en série avec le système.

La sortie de système pour des consignes échelon, rampe et $\theta_e(k)$ avec :

$$\theta_e(k) = 1.5 \sin(2.\pi.k / 2.5) + 0.5 \cos(2.\pi.k / 15) + 0.5 \cos(2.\pi.k / 5)$$

Sont représentées par les figures, (III.20), (III.21), (III.22) respectivement



III.3.4. Commande par modèle de référence

L'entrée de référence est $\theta_e(k)$.

L'objectif est déterminer une commande $U(k)$ qui minimise l'écart entre la sortie du système et la sortie du modèle de référence.

Le contrôleur neuronal comporte :

- 01 couches cachées.
- 04 neurones dans la couche d'entrée.
- 07 neurones dans la couche cachée.
- Un seul neurone dans la couche de sortie.
- La fonction d'activation utilisée dans la couche d'entrée et les deux couches cachées est la

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

- La fonction d'activation utilisée dans la couche de sortie est la fonction linéaire $f(x) = x$

Pour entraîner ce contrôleur le modèle neuronal du système obtenu est placé en parallèle avec le système. Le rôle du modèle est la rétropropagation de l'erreur de sortie pour l'adaptation des poids du contrôleur neuronale.

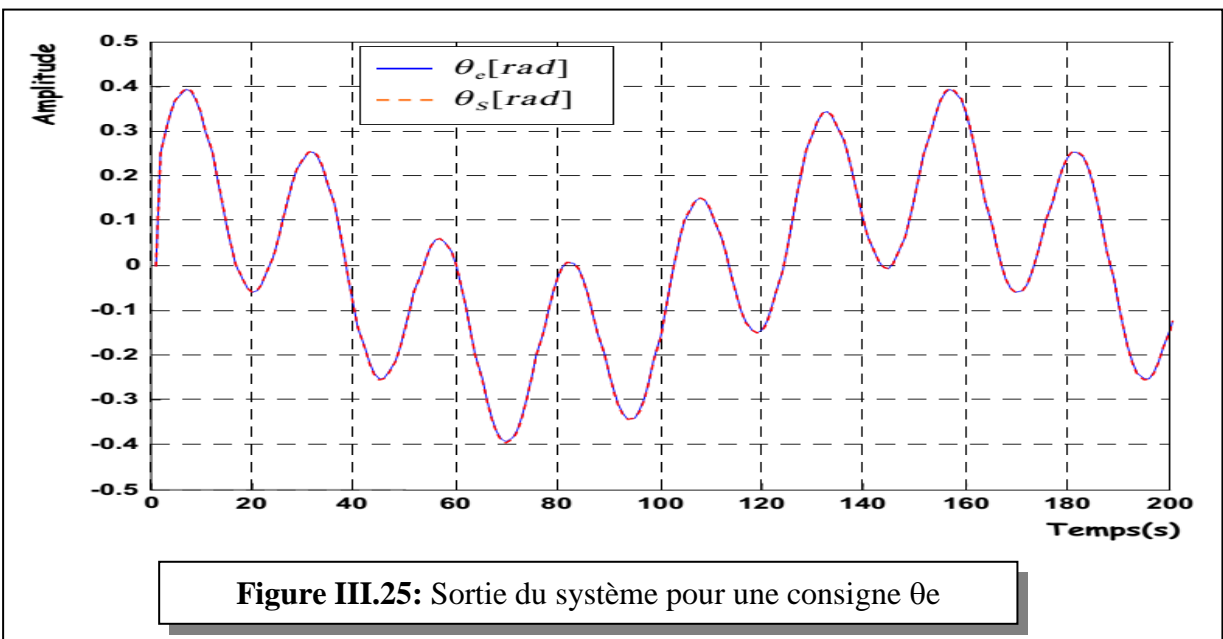
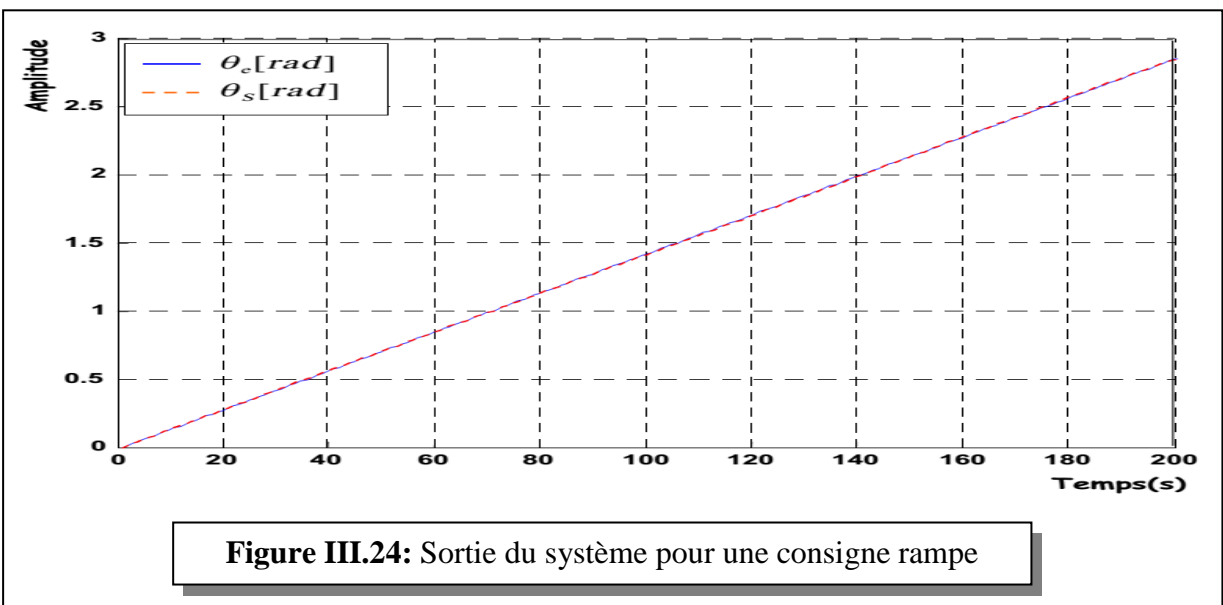
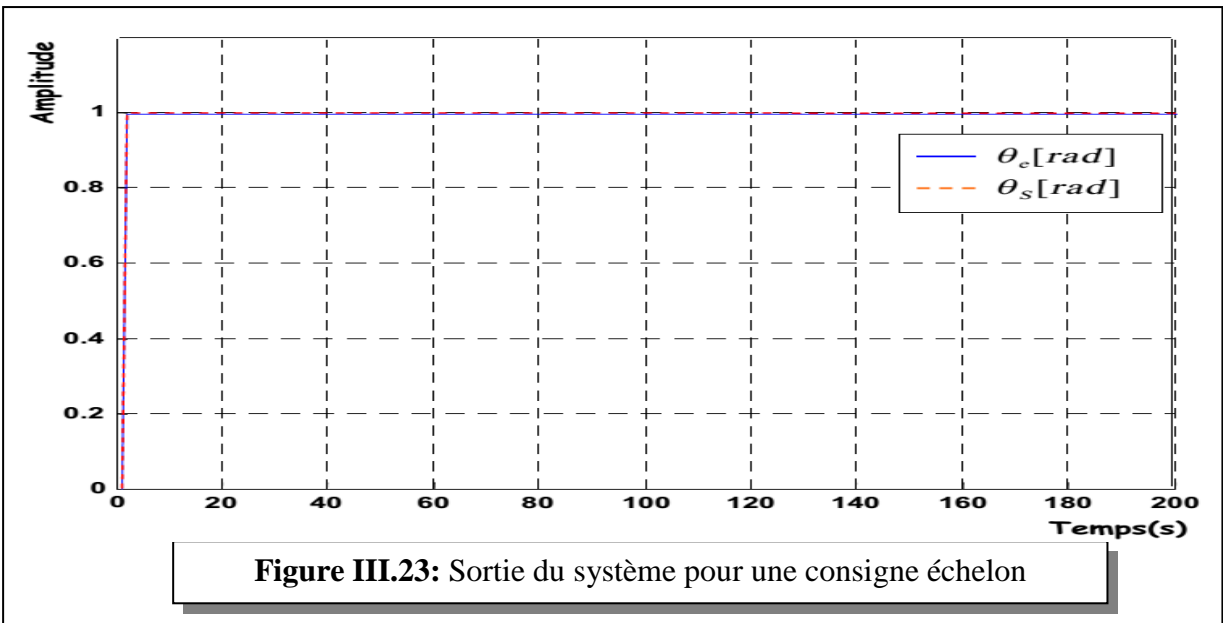
L'adaptation des paramètres est effectuée pendant 14650 périodes.

La sortie du système et le modèle de référence pour des consignes échelon, rampe et $\theta_e(k)$ avec :

$$\theta_e(k) = 0.2(\sin(2.\pi.k / 2.5) + \cos(2.\pi.k / 15))$$

sont représentés par les figures, (III.23), (III.24), (III.25) respectivement.

L'erreur entre la sortie du système et la consigne est de l'ordre de 10^{-4} .



III.4 Conclusion:

Dans ce chapitre on a présenté la parabole de radar de poursuite qui est l'objet de la commande. et, nous allons voir les résultats offerts par quelques structures d'identification et de contrôle neuronal étudiées, à partir de ces résultats nous allons juger les capacités de ces structures neuronales d'approximation et de génération des commandes nécessaires pour la réalisation des tâches désirées.

L'identification du modèle direct et inverse est obtenue après plusieurs essais car le réseau se stabilisait quelques fois à un niveau d'erreur inacceptable à cause du problème du minimum local ce qui nous obligeait d'exécuter le programme une autre fois avec un nouveau pas d'apprentissage.

Les modèles obtenus ont été testés sur un intervalle n'appartenant pas à l'ensemble d'apprentissage, les résultats obtenus confirment la capacité de la généralisation des réseaux de neurones et les structures de contrôle neuronal utilisées ont la possibilité d'engendrer les commandes nécessaires pour obtenir des réponses désirées d'une façon immédiate. Les résultats de simulation montrent que les structures de contrôle neuronal présentent de bonnes performances.



**CONCLUSION
GENERALE**

CONCLUSION GENERALE:

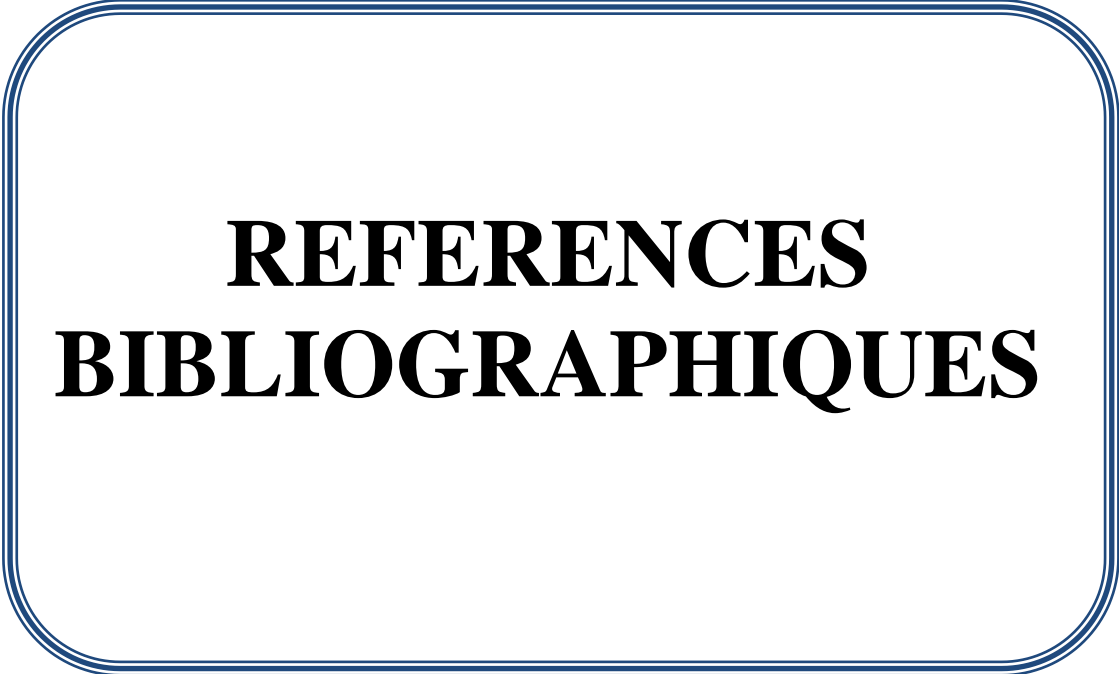
Le présent travail nous a permis de découvrir des méthodes nouvelles, efficaces et puissantes pour l'identification et la commande des systèmes non linéaires.

Nous avons donné un aperçu général sur les principes de base et de fonctionnement des réseaux de neurones, et d'étudier en détail l'apprentissage par la méthode de rétropropagation et son extension pour l'entraînement des réseaux statiques, on a présenté les structures d'identification et de commande à base des réseaux de neurones qui sont appliqués à la commande d'une parabole de radar de poursuite.

Les réseaux de neurones offrent un éventail de structures de contrôle dont nous avons utilisé les structures de contrôle par modèle inverse et par modèle de référence. Les résultats obtenus avec ces structures de commande sont satisfaisants.

Nous avons conclu que la technique de contrôle par inversion du modèle est rarement utilisée dans la pratique parce qu'elle représente une structure de commande en boucle ouverte comme elle n'est évidemment pas robuste aux erreurs de modélisation, c'est ainsi que nous avons établi les modalités de la mise en oeuvre de systèmes de commande neuronaux adaptatifs dans lesquels le contrôleur neuronal est capable de générer des commandes d'une façon immédiate.

On peut donc dire que la technique de commande neuronale est beaucoup plus pratique que la commande classique, et ceci ouvre plusieurs perspectives à d'autres applications dans différents domaines utilisant les réseaux de neurones artificiels.



REFERENCES
BIBLIOGRAPHIQUES

BIBLIOGRAPHIE

- [1] **B.Mimi et B. Elhafid** « *Identification et contrôle de la machine a induction par les réseaux de neurones* » **PFE**. Université Mentori, 2000.
- [2] **M. Michel Weined** « *Les réseaux de neurones* » Technique d'ingénieur 1995.
- [3] **B. Radouane et B. Hafid** « *Contrôle d'une machine à induction par les réseaux neuronaux et neuro-Flous* ». **PFE**. Université Mentori 2002.
- [4] **E. Davalo et P. Naïm** « *Des réseaux de neurones* » Edition Eyrolle, 1989.
- [5] <http://www.ensmp.fr/~moutarde> « *Introduction aux réseaux de neurones* ». **Fabien Moutarde**, Ecole des mines de paris, Décembre 2003.
- [6] **E. Davalo et P. Naïm** « *Des réseaux de neurones* » 2^{ème} édition, Edition Eyrolle 1990.
- [7] **B.Yamina et Z. Fatiha** « *La détection d'anomalie dans les machines synchrone par les réseaux de neurones* » **PFE**. Université Mentori, 2000..
- [8] **C. Touzet** « *les réseaux de neurones artificiels* » 1992.
- [9] **B. Fouad et M .Ilyase** « *diagnostic en temps réel des défaillances d'un ensemble Moteur asynchrone –convertisseur électrique par application des techniques d'intelligence artificielles (réseaux de neurones)* ». **PFE**. Université de M'sila, 2004.
- [10] **G. Zwingelstein**, «*Diagnostic des défaillances, théorie et pratique pour les systèmes industriels* », éd. Hermes, Paris 1995.
- [11] **E. Davalo et P. Naïm** « *Des réseaux de neurones* » Edition Eyrolle 1993.
- [12] **O. Bennia et L. Mohamadi** « *Identification des systèmes non Linéaire par les réseaux de neurones* ».PFE, Université de M'sila.
- [13] **B. Mohamed**, *Identification et contrôle avec réseaux de neurones, thèse de magistère*. Institut d'électronique Sétif, 1993.
- [14] **F. Mossa et Ziadi Aissa**, *commande d'un moteur a courant continu par les réseaux de neurones* Thèse d'ingénieur. Institut d'électronique Sétif, 1999
- [15] **I. Rivale**, *modélisation et commande de processus par réseaux de neurones, application au pilotage d'un véhicule autonome*, thèse de Doctorat de l'université Pierre et Marie, Paris VI, janvier 1995.

- [16] **Bengio S**, *Optimisation d'une règle d'apprentissage pour réseaux de neurones artificiels*, PhD thesis, Département d'Informatique et Recherche Opérationnelle. Université de Montréal, 1993.
- [17] **O. Zoubire et B. k. Salim**, *Identification et commande d'une parabole de radar poursuite par les réseaux de neurones*. Thèse d'ingénieur. Institut d'électrotechnique Sétif, 1997
- [18] **A. Zergaoui** « *Identification et contrôle d'une machine asynchrone par les réseaux de neurones* ». Thèse de magister, Université Mentori, 1997.
- [19] **B.Chetate, DJ.Khodja** « Diagnostic en temps réel des défaillances d'un ensemble Moteur asynchrone–convertisseur électronique en utilisant les réseaux de neurones artificiels », Journal d'Electrotekhnika, Moscou12/2003, pp : 16-20

Résumé : Dans ce travail, nous avons mis en évidence les propriétés et les avantages des réseaux de neurones, et nous avons envisagé leurs exploitations dans l'identification des systèmes non linéaires et la commande des processus. L'algorithme de rétropropagation du gradient est utilisé pour l'entraînement des réseaux multi couche présentés dans toutes les structures de modélisation et de commande que nous avons considérées dans ce mémoire. Nous avons appliqué les structures de commande étudiées sur la parabole de radar de poursuite, pour montrer l'efficacité et la pertinence de ces structures neuronales présentées à la commande.

Mots Cles : Réseaux de neurones artificiels, Apprentissage, Algorithme de rétropropagation, Moteur à courant continu, Identification, Commande.

المخلص : في هذا العمل، قمنا بتسليط الضوء على خصائص وفوائد الشبكات العصبية، ونظرنا في استخداماتها في تحديد الأنظمة غير الخطية والتحكم في العمليات. يتم استخدام خوارزمية التدريب بالانتشار العكسي للمرجعية لتدريب الشبكات المتعددة الطبقات المقدمة في جميع هياكل التصميم والتحكم التي ناقشناها في هذه الرسالة. قمنا بتطبيق الهياكل التحكم المدروسة على منحنى قمره الرادار للتتبع، لنظهر كفاءة وملاءمة هذه الهياكل العصبية المقدمة للتحكم..

الكلمات المفتاحية : شبكات الأعصاب الاصطناعية، التعلم، خوارزمية الانتشار العكسي، محرك التيار المستمر، التحديد، التحكم.

Summary : In this work, we have highlighted the properties and advantages of neural networks, and we have considered their applications in the identification of nonlinear systems and process control. The gradient backpropagation algorithm is used for training the multilayer networks presented in all the modeling and control structures that we have considered in this paper. We applied the studied control structures to the tracking radar parabola to demonstrate the effectiveness and relevance of these neural structures in control..

Keywords: Artificial neural networks, learning, backpropagation algorithm, direct current motor, identification, control.