15/94

REPUBLIQUE ALGERIENNE DEMOCRATIQUE ET POPULAIRE

MINISTERE DES UNIVERSITES ET DE LA RECHERCHE SCIENTIFIQUE

ECOLE NATIONALE POLYTECHNIQUE

Département : GENIE ELECTRIQUE

PROJET DE FIN D'ETUDES

Thème

SUR LA LINEARISATION DE CAPTEURS A L'AIDE DES RESEAUX DE NEURONES ARTIFICIELS

Proposé par :
Mr M. ATTARI
Mr F. BOUDJEMA

Etudié par :
Mr MM. HENICHE

Dirigé par :
Mr M. ATTARI
Mr F. BOUDJEMA

PROMOTION
JUILLET 1994

E.N.P. 10, Avenue Hacen BADI El - Harrach. ALGER

REPUBLIQUE ALGERIENNE DEMOCRATIQUE ET POPULAIRE

MINISTERE DES UNIVERSITES ET DE LA RECHERCHE SCIENTIFIQUE

ECOLE NATIONALE POLYTECHNIQUE

Département : GENIE ELECTRIQUE

PROJET DE FIN D'ETUDES

Thème

SUR LA LINEARISATION DE CAPTEURS A L'AIDE DES RESEAUX DE NEURONES ARTIFICIELS

Proposé par :
Mr M. ATTARI
Mr F. BOUDJEMA

Etudié par :
Mr MM. HENICHE

Dirigé par :
Mr M. ATTARI
Mr F. BOUDJEMA

PROMOTION
JUILLET 1994

E.N.P. 10, Avenue Hacen BADI El - Harrach. ALGER

Dédicaces

Je dédie ce modeste travail en signe de reconnaissance

A *ma mère pour son amour*

A *mon père pour ses précieux conseils*

A *mon oncle Slimane et ma tante Fatima pour leur générosité*

A *mes soeurs Khalida et Zahira*

A *mes frères Mostafa , Salim et Djamel*

A *toute ma famille*

A *tous mes amis (ies)*

A *ceux que j' aime et qui sauront se reconnaitre*

M' *hamed.*

Remerciements

Je remercie vivement mes promoteurs Monsieur M. ATTARI et Monsieur F. BOUDJEMA pour leur suivi et pour les conseils précieux qu' ils m' ont prodigué tout au long de l' élaboration du présent travail.

Que les membres du jury trouvent ici l' expréssion de ma reconnaissance pour avoir accepter d' honorer le jugement de ce travail.

Je tiens à remercier particulièrement Monsieur I. ADNANE et son agence ART WARE pour leur contribution dans l' impression de ce mémoire.

A toutes les personnes qui ont contribué de prés ou de loin à l' aboutissement de ce travail, j' exprime ma profonde gratitude.

Enfin, je remercie tous les enseignants qui ont assuré ma formation durant les cinq années d' études.

Table des matières

INTRODUCTION GENERALE

Sous le terme de réseaux de neurones, on regroupe aujourd' hui un certain nombre de modèles dont l' intention est d' imiter certaines fonctions du cerveau humain en reproduisant certaines de ses structures de base.

En résumé, les réseaux de neurones artificiels sont issus de l' approche connexionniste qui prévilègie les avantages suivants.

* L' activité parallèle et en temps réel de nombreux composants.

* La présentation distribuée des connaissances.

* L' apprentissage par médification des connexions.

Actuellement, les applications pratiques des réseaux de neurones commencent à se proliférer [20], [10], [11], [16], et cette discipline va concerner un public de plus en plus large d' étudiants de chercheurs, d' ingénieurs et d' industriels.

Parmi ces applications figure un bon nombre de travaux de recherche dans le domaine d' instrumentation, on citera la contribution de L.F. PAU dans la compréhension des signaux issus des instruments de mesures à l' aide des réseaux de neurones [22], et celle de R. NAIDO pour la détéction des erreurs dûs aux capteurs lors de la commande des systèmes [21].

Dans ce même domaine d' instrumentation, des traveaux ont été menés pour la linéarisation de capteurs non linéaires [17], [19] sans avoir recours au réseaux neuronaux. Pour cela, on va présenter une méthode originale de linéarisation de capteurs à l' aide des réseaux de neurones, tout en s' inspirant des autres méthodes de linéarisation.

Cette étude est basée sur notre propre expérience des difficultés que peut présenter les réseaux de neurones. Elle a pour objectif de tester cette nouvelle méthode et de donner des éléments pouvant être nécessaires à des recherches plus poussées.
Ceci étant, nous avons divisé notre travail en 5 chapitres :

Le premier chapitre est consacré à la présentation des réseaux neuronaux avec certaines définitions et quelques notions sur les classifications des réseaux neuronaux, l' apprentissage et la généralisation.

Le deuxième chapitre donne une idée assez détaillée sur deux algorithmes d' apprentissage rétropropagation (BP) et la méthode d' optimisation

aléatoire (ROM), qu' on utilisera ultérieurement.

Dans le troisième chapitre il nous a semblé intéressant de présenter les autres méthodes de linéarisation leur principe de fonctionnement ainsi que leur avantage et inconvénients.

Le quatrième chapitre quant à lui, contient le début de notre application c'est à dire l' utilisation des réseaux de neurones pour la linéarisation de deux types de fonctions non linéaires celle de type dûr et celle de type mou, cette étape est une préparation à l' application sur les capteurs et a fait l' objet d' un article présenté en annexe.

Enfin, le cinquième chapitre comprend l' utilisation des réseaux de neurones pour la linéarisation des caractéristiques de quatre types de capteurs différents : deux thermistances, un anémomètre, un thermocouple et une éléctrode séléctive d' ions. Ce dernier capteur fait l' objet de linéarisation en présence d' une grandeur de perturbation. Les résultats sont illustrés sous forme de tableaux et de graphes, et donnent une bonne idée au lécteur sur la qualité de la linéarisation dans chaque cas.

En perspective, on mentionnera les atouts qu' apporte cette nouvelle méthode de linéarisation en commande et en instrumentation.

Cette étude aura atteint sont but si les résultats aux quelles elle a abouti : formeront une expérience qui peut être utile dans la suite de tout travail lié aux réseaux de neurones et à la linéarisation en général .

CHAPITRE 1

PRESENTATION DES RESEAUX DE NEURONES ARTIFICIELS

1.1 **Introduction** :

- Les recherches scientifiques menées ces dernières décennies dans le domaine neurobiologique, ont laissé les chercheurs penser à réaliser des systèmes capables d'imiter les capacités extraordinaires du système nerveu humain, cela en exécutant certaines tâches non assurées par le plus puissant des ordinateurs.

- Les premiers travaux furent lancés en 1943 par Mc Culloch et Pitts [1], par la mise au point d'un ensemble de **neurones formels** capables de réaliser certaines fonctions logiques.
Les recherches menées par Rosenblatt sur le **perceptron** et inspirées du système visuel, n'allèrent pas loin vu les limites theoriques qu' elles ont rencontré. Le domaine des réseaux de neurones était donc écarté, et les chercheurs se sont penchés vers l'**intelligence artificielle**, un domaine qui était plus prometteur.

Le formalisme actuel des réseaux de neurones mené par : Hoplield Grossberg, Kohonen pour ne citer que ceux là, semble donner de meilleurs résultats. [10]
Aujourd' hui, la plupart des réseaux de neurones sont réalisés par simulation, en utilisant les algorithmes appropriés aux types d' application envisagées. Dans ce chapitre on présentra,
quelques définitions concernant les réseaux neuronaux, ainsi que leur classification telle qu' elle a été conçue. On parlera brièvement de la notion d' **apprentissage** et de **mémoire associative.**

1.2 Structure et fonctionnement d' un neurone artificiel :

1.2.1 Neurone formel :

C' est le résultat de l' analogie directe, qu' ont faite Mc Culloch et Pitts avec le neurone biologique. Il s' agit d' un corps cellulaire qui execute une somme pondérée des entrées qui lui parviennent. Si cette somme dépasse un certain seuil le neurone est activé, autrement le neurone est dit désactivé ou au niveau bas [1]. Cf figure (1-1)

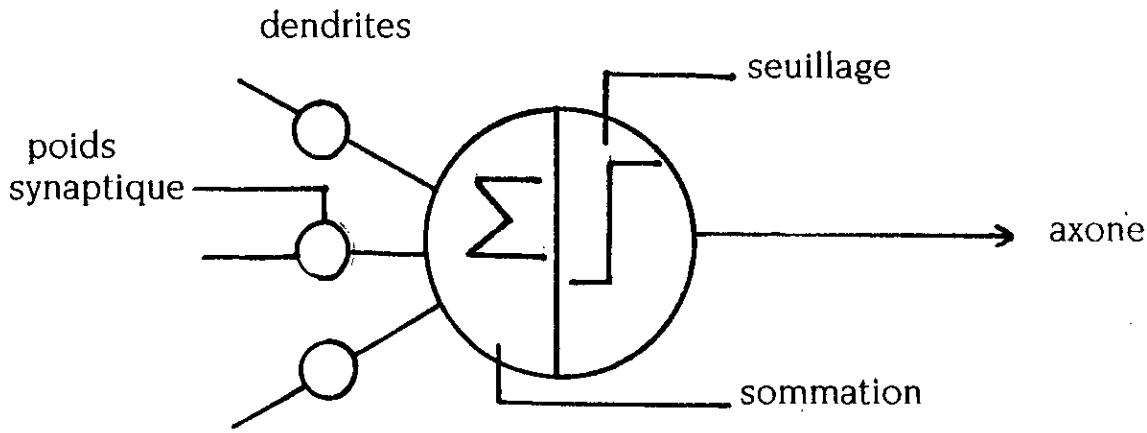


fig 1-1 : Neurone formel

1.2.2 Elément linéaire adaptatif (adaline)

- L' élément linéaire adaptatif, est le bloc de base dans l' architecture d' un réseau de neurone, Cf figure (1-2) [3], il est appelé ainsi car il permet une adaptation de ses poids synaptiques au vu d' un certain comportement.

Cet élément reçoit à l' instant k un vecteur X_k est une réponse désirée dk dont on vera le rôle dans le paragraphe de l' apprentissage, les composantes du vecteur d' entrée X_k sont pondérées à l' aide d' un vecteur poids W_k.

Aprés avoir calculer la somme pondérée, on applique à celle-ci une fonction dite d' **activation.** Le vecteur d' entrée x_k peut contenir des composantes à valeurs continues ou binaires. [4]

Remarques sur la fonction d' activation :

Suivant le type d' application à laquelle on veut méler le réseau, la fonction d' activation peut prendre, plusieurs modèles mathématiques nous citerons :

$$F(x)=\frac{1}{1+\exp(-x)}$$

$$F(x)=\tanh(x)$$

$$F(x)=\mathrm{sgn}(x)$$

Dans tous les cas, la fonction F(x) doit être saturable, pour éviter des valeurs en sortie trop élevées qui peuvent déstabiliser le réseau.
- La remarque importante qu' on peut faire sur la fonction d' activation est son caractère **non linéaire**, ce caractère avantageux donne la possibilité au réseau de simuler des fonctions complèxes, non linéaire [2].

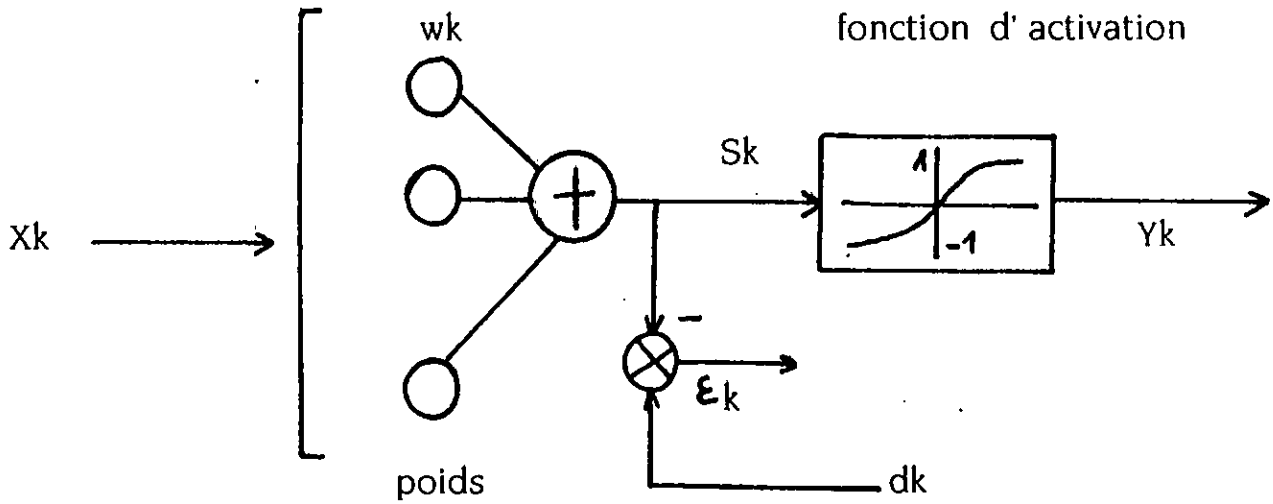


fig (1-2) : Elément linéaire adaptatif.

1.3 Réseau de neurones :

L' assemblage de plusieurs éléments linéaire adaptatifs, où chaque élément est relié à d' autres par le biais de ses entrées et sorties, est appelé réseau de neurones. Ces liaisons peuvent être d' une couche de neurones vers une autre, dans ce cas les éléments ne sont pas tous reliés entre eux, le cas contraire nous donne un réseaux dont les éléments sont entièrement connectés. Tout cela sera détaillé dans ce qui suit.

Un réseau neuronal, n' est autre qu' un système **connexionniste** [5] complexe formé de nombreuses interactions entre les éléments, leur permettant ainsi de travailler ensemble pour résoudre les problèmes. Ces éléments ont un comportement individuel simple, leur influences mutuelles décident du comportement global du réseau.

1.4 Apprentissage des réseaux :

On entend par apprentissage d' un réseau, l' opération qui consiste à modifier les poids des connexions dans le réseau lors de la présentation d' un vecteur d' entrée à celui-ci, la modification des poids se poursuit jusqu' à ce que ces derniers ne varient que d' une façon infime. En effet, dans cette partie du travail, le réseau est entrainé pour le faire adapter à un type d' application. Ce travail d' apprentissage en fait, et réalisé à l' aide, d' algorithmes appropriés qu' on vera par la suite.
L' apprentissage des réseaux de neurones peut être inadaptable à certaines tâches. Alors pourquoi l' utilise - t - on ?
Et-bien parce qu' il peut donner au réseau la possibilité de trouver la relation qui existe entre les différentes entrées qu' on lui fait apprendre et donc il poura aprés réaliser une généralisation, sur les données qu' on ne lui a pas présenté. Cette notion de généralisation est trés utile, car les données réelles du problème sont trés souvent bruitées. [2]

1.4.1 Principe de perturbation minimale :

Tous les algorithmes itératifs utilisés pour l' apprentissage des réseaux travaillent suivant un principe important qui stipule que la minimisation de l' erreur de sortie se fait d' une manière à ce que la perturbation créee sur les exemples déja appris soit minimale. [4] Ce principe intuitif à été à la base de la construction de tous les algorithmes d' apprentissage.

1.4.2 Remarque sur l' apprentissage des réseaux :

Les travaux expérimentaux menés sur les réseaux de neurones ont montré que durant l' apprentissage, "ces derniers peuvent mémoriser des exemples sans pour autant apprendre quel est le lien entre eux". Expliquons-nous. En effet si on dépasse un certain nombre d' exemples d' entrée à mémoriser, le réseau peut perdre la capacité de généralisation et donne ainsi des sorties non satisfaisantes en gardant même des réponses correctes pour les exemples déjà appris. [2] Ce qu' on peut faire dans ce cas, c' est d' effectuer un test de généralisation chaque fois que le réseau aura appris un certain nombre d' exemples, si le test commence à être négatif on stoppe les exemples d' apprentissage, pour éviter ce qu' on appelle

le surapprentisage. Cette explication est réunie dans un schéma graphique de la figure (1-3). [2]

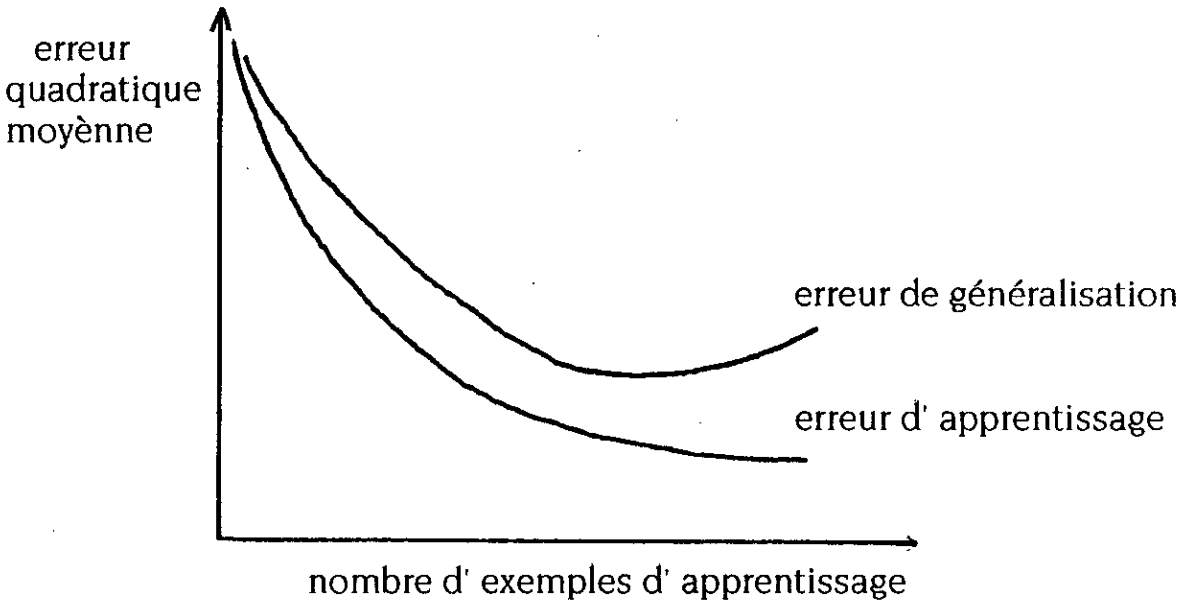


fig 1-3 : Phénomène de surapprentissage

1.5 Classification des réseaux de neurones :

1.5.1 classification selon la structure du réseau :

Dans ce cas les réseaux neuronaux sont classés suivant leur architectures, ce qui nous donne deux types de réseaux :

1.5.1.1 Réseaux multicouches :

Le réseau est constitué de plusieurs couches, comportant chacune un nombre donné d' éléments adaptatifs, ces derniers ne sont pas reliés entre eux. L' élément d' une couche à comme signaux d' entrée, les signaux de sortie de tous les éléments de la couche précédente, sa sortie est alors considérée comme entrée pour tous les éléments de la couche suivante. Cf figure (1-4)

- Le réseau est constitué d' une couche d' entrée liée directement au vecteur d' entrée X, d' une couche de sortie donnant les signaux accessibles du reseau .les couches dont les sorties des éléments ne sont pas

accéssibles sont dites **couches cachées**, on les introduit dans le réseau pour augmenter sa capacité de simuler les fonctions de plus en plus complexes [2] .

Dans ce type de réseaux la propagation des signaux se fait vers l' avant, de la couche d' entrée vers la couche de sortie, on les nomme alors réseaux à propagation avant ou **Feed forword networks** .

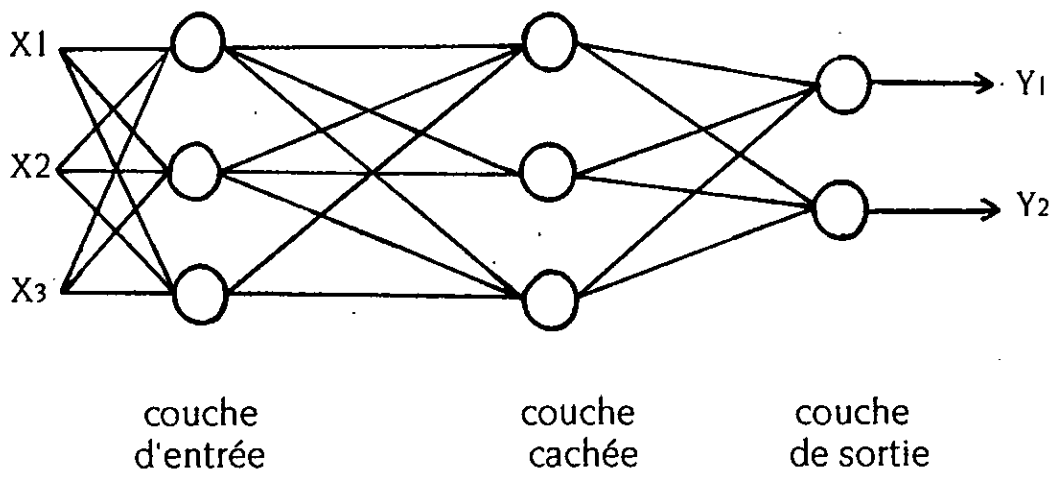


fig 1-4 : Réseaux de neurone multicouches.

1.5.1.2 Réseaux entièrement connectés :

Dans cette structure du réseau, chaque neurone communique avec tous les autres, et même eventuellement avec lui-même. La manière avec laquelle le réseau s' organise était définie par D.O. Hebb elle est basée sur des observations biologiques, et qui dit que l' activation simultanée de deux neurones tend à renforcer leur force de connexion. [6]
La véritable approche de ce type de réseaux a été faite par Hopfield en 1982, et selon lui le réseau doit chercher un état stable parmi plusieurs présentés lors de l' apprentissage. Bien qu' il semble parfois difficile à en-

trainer car il demande des éléments de la physique statistique, ce réseau est le plus proche de la réalité. Cf. fig (1-5)

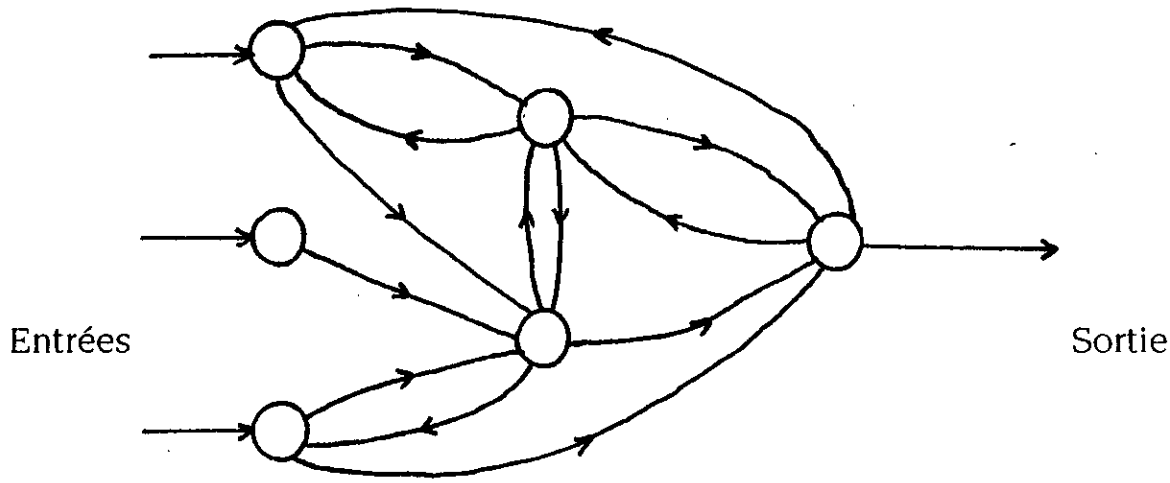


fig 1-5 : Exemple d' un réseau entièrement connecté

1.5.2 classification selon le type d' apprentissage :

1.5.2.1 Apprentissage supervisé :

C' est un apprentissage surveillé. Les poids synatiques se modifient graduellement de manière à ce que le réseau donne une réponse désirée à une sollicitation donnée. Les exemples d' apprentissage seront alors formés de couples (Xk , dk), où Xk est le vecteur d' entrée du réseau et dk est la réponse voulue à la sortie. Parmi les algorithmes utilisés dans ce type d' apprentisage, on citera :

- L' algorithme de **rétropropagation** (**backpropagation**) développé par Rumelhart Hinton et Williams, et qui est basée sur la méthode de la décente du gradient appliquée à la surface d' erreur quadratique, le but étant de minimiser cette erreur.

- L' algorithme Madaline III développé par D. Andes [4] et qui est mathématiquement équivalent à la rétropropagation.

Ce type d' apprentissage est surtout utilisé dans le domaine de reconnaissance des formes, de la modelisation et la simulation des fonctions.

1.5.2.2 Apprentissage non supervisé :

C' est le type d' apprentissage où les exemples d' entrainements sont dé-

pourvus de réponses désirées. Dans ce cas, le réseau se contentera de trouver des relations statistiques qui existent parmi les exemples d' apprentissage. [2] On citera :

- Le réseau d' "**Hopfield**", basée sur la loi de Hebb, est utilisé en tant que " mémoire associative "

- La machine de Boltzmann fait une analogie avec la thermodynamique, les entrées présentées au réseau sont attirées par les exemples appris qui leurs correspondent le mieux. [1]

1.6 Mémoire associative :

La fonction de mémoire associative est étroitement liée aux réseaux neuronaux. Elle consiste en premier lieu à stocker ou mémorise des données sous formes d' entrées / sorties durant la phase d' appentissage, en second lieu d' utilisation ou de rappel, une fois les poids synaptiques stabilisés, on présentera au réseaux une entrée bruitée d' un exemple déjà mémorisé. Ici on distinguera deux cas [7] :

• Si la réponse correspond à celle désirée, déjà apprise on dira qu' il y a une **auto association** .

• Si la réponse n' est pas celle apprise, mais elle est associée à celle-ci, on dira qu' il y a **hétéroassociation** .

Dans la phase d' utilisation, la notion la plus importante est celle de la décision, car le réseau doit décider si la donnée est plus proche à tel exemple appris plûtot qu' à un autre.
Au fait cette capacité est tout simplement tirée du fonctionnement du cerveau humain, car celui-ci peut aprés avoir vu le visage de quelqu' un l' associé automatiquement au nom de celui là.

1.7 Conclusion :

Dans un réseau de neurones tel qu' il soit, le traitement des données ou des connaissances est parallèle et distribué, c' est à dire que toutes les unités du réseau travaillent simultanément.

C' est des systèmes difficiles à construire, qui sont utilisés dans des applications ne permettant pas d' expliquer les résultats fournis [5], mais par contre, ce sont des systèmes adaptatifs leur capacité d' apprentissage leur permet de prendre en compte les nouvelles contraintes qui apparaissent, à l' inverse des ordinateurs qui eux ne sont pas entrainés mais programme [2].
De plus, c' est des systèmes faciles à simuler, par des languages souples, car on a nullement besoin de reprogrammer le fonctionnement du réseau, si on ajoute des neurones, supplémentaire.
Enfin, c' est des systèmes capables de généralisation, une conséquence bénéfique qui résulte d' un bon apprentissage.

CHAPITRE 2

ALGORITHMES D'APPRENTISSAGE DES RESEAUX NEURONAUX

2.1 Introduction:

Comme on vient de le voir dans le chapitre précédent, la phase d' apprentissage est une étape déterminante pour la construction du réseau de neurones. Pour cela, des algorithmes appropriés à cette tâche, ont été élaborés et développés au fil des années pour être appliquer chacun au type de réseau qu' il faut. Ces méthodes sont axées sur des **techniques mathématiques et numériques** connues, qui étaient utilisées dans d' autres domaines où ils ont prouvé leur efficacité.
Dans ce présent chapitre, deux principaux algorithmes d' apprentissage sont présentés :

La méthode de **rétropropagation du gradient** (BP) et celle d' **optimisation aléatoire** (ROM) , qui sont utilisés pour l' apprentissage des réseaux multicouches uniquement.

- L' algorithme de rétropropagation, découvert par Werbos en 1974 et qui a fait l' objet de son sujet de doctorat, a été développé par Rumelhart, Hinton et Williams en 1986 à MIT [4].

- La méthode d' optimisation aléatoire quant à elle, a été découverte par Matyas en 1965, pour être de nouveau améliorée en 1981 par Solis et Wets à l' université de Kentucky [9].

D' autres algorithmes ont été élaboré pour d' autres types d' architectures de réseaux neuronaux et dont on ne vera pas la suite dans ce chapitre, à titre d' exemples on cite la theorie de **résonnance adaptative** (ART) développée par Grossberg à MIT et utilisée pour des réseaux ayant deux couches en interaction [2], [4] pour ne parler que de celle-ci.

2.2 Algorithme de rétropropagation du gradient : (BP)

Cet algorithme est utilisé pour une topologie de réseaux multicouches, l' apprentissage y est supervisé.
Il est basé sur une méthode numérique dite de relaxation qui effectue une descente de gradient sur la surface d' erreur quadratique moyenne, elle utilise donc les techniques de dérivées partielles, ceci étant, on est amené à utiliser une fonction d' activation dérivable appelée **sigmoide**, par exemple

$$F(x) = 1 / (1 + \exp(-x))$$

Le principe de cette méthode est résumé dans la figure (2-1) : il s' agit de présenter au réseau, une entrée ou un vecteur d' entrée et un vecteur de sortie désiré, le réseau s' engagera à ce moment à calculer sa propre sortie par une propagation avant des calculs. Une fois la sortie calculée, celle-ci présente forcément une erreur par rapport à celle désirée, l' algorithme utilise cette erreur pour l' adaptation des pondérations en faisant une propagation arrière (backpropagation) dans le but de minimiser cette erreur.

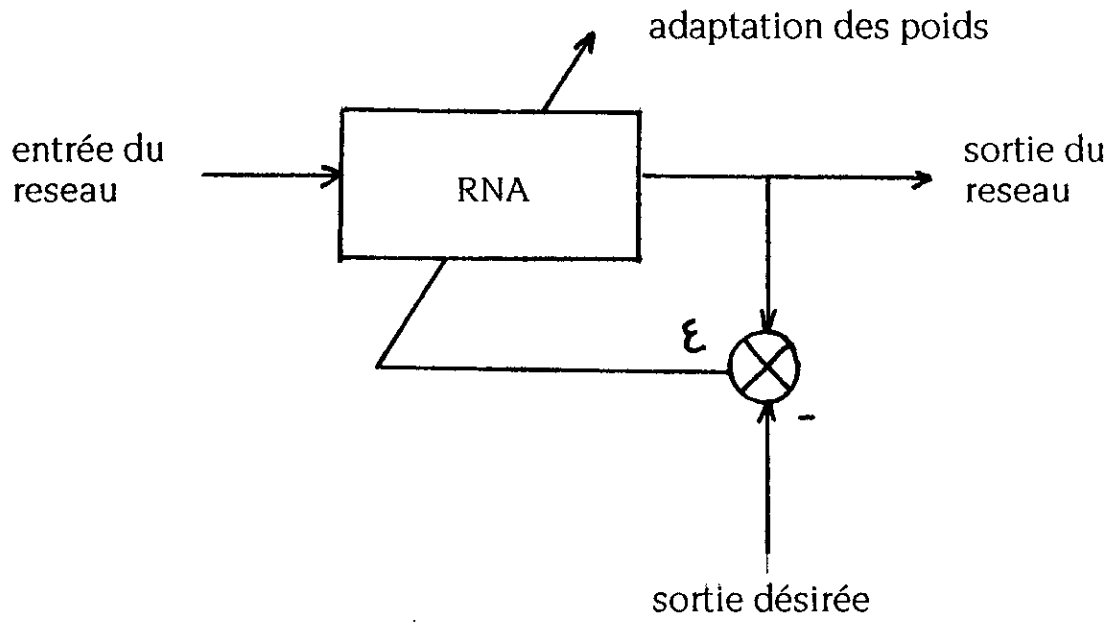


fig 2-1 : Principe de la rétropropagation

- La démonstation mathématique attachée à cet algorithme est la suivante [1] :

On note:

$X = (x_1, x_2, ..., x_n)$ entrées du réseau.
$Y = (y_1, y_2, ..., y_m)$ sorties désirées du réseau.
$S = (s_1, s_2, ..., s_m)$ sorties réelles du réseau.

f : fonction sigmoide de chaque neurone, f' sa dérivée.

Oj : sortie du neurone j.
Ii : entrée du neurone i. (Somme pondérée)

μ : pas du gradient.

Cette notion est résumée dans la figure (2-2) :

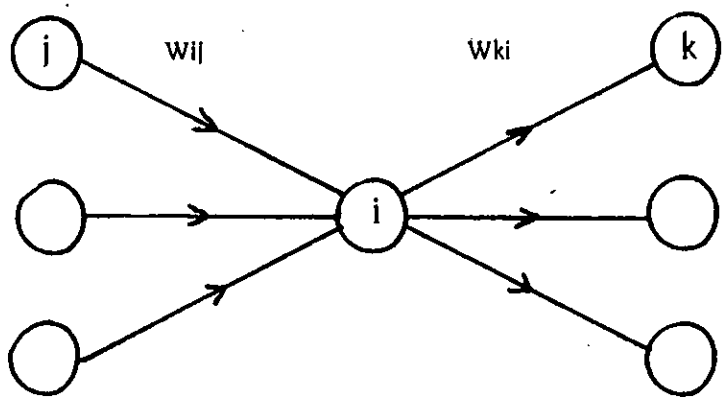


fig 2-2 : liens du neurone i avec les neurones de la couche précédente et suivante.

On définit la fonction coût à minimiser :

$$J(w)=\sum_{i=1}^{m}(si-yi)^2 \qquad (2\text{-}1)$$

$$wij(k+1)=wij(k) - \mu \frac{\partial J}{\partial wij} \qquad (2\text{-}2)$$

on a :

$$\frac{\partial J}{\partial wij} = \frac{\partial J}{\partial Ii} \frac{\partial Ii}{\partial wij} \qquad (2\text{-}3)$$

or :

$$\frac{\partial Ii}{\partial wij} = \frac{\partial(\sum_{p} wip\ Op)}{\partial wij} = Oj \qquad (2\text{-}4)$$

on a donc :

$$\frac{\partial J}{\partial w_{ij}} = d_i \, o_j \qquad (2\text{-}5)$$

avec:

$$d_i = \frac{\partial J}{\partial I_i} \qquad (2\text{-}6)$$

Pour la couche de sortie :

$$d_i = \frac{\partial \sum_j (s_j - y_j)^2}{\partial I_i} = 2 (s_i - y_i) \frac{\partial s_i}{\partial I_i} \qquad (2\text{-}7)$$

$$d_i = 2 (s_i - y_i) \; f' (I_i) \qquad (2\text{-}8)$$

Pour les neurones des couches cachées :

$$d_i = \sum_h \frac{\partial J}{\partial I_h} \frac{\partial I_h}{\partial I_i} = \sum_h d_h \frac{\partial I_h}{\partial I_i} \qquad (2\text{-}9)$$

avec h indicatif des neurones reliées à la sortie du neurone i.

on a donc :

$$d_i = \sum_h d_h \, w_{hi} \, f' (I_i) \qquad (2\text{-}10)$$

la règle de modification des poids sera alors : Cf figure (2-3)

$$w_{ij} (k+1) = w_{ij} (k) - \mu \, d_i \, o_j \qquad (2\text{-}11)$$

avec :

$d_i = 2 (s_i - y_i) f' (I_i)$ pour la couche de sortie .

$d_i = \sum_h d_h w_{ih} f' (I_i)$ pour les couches cachées .

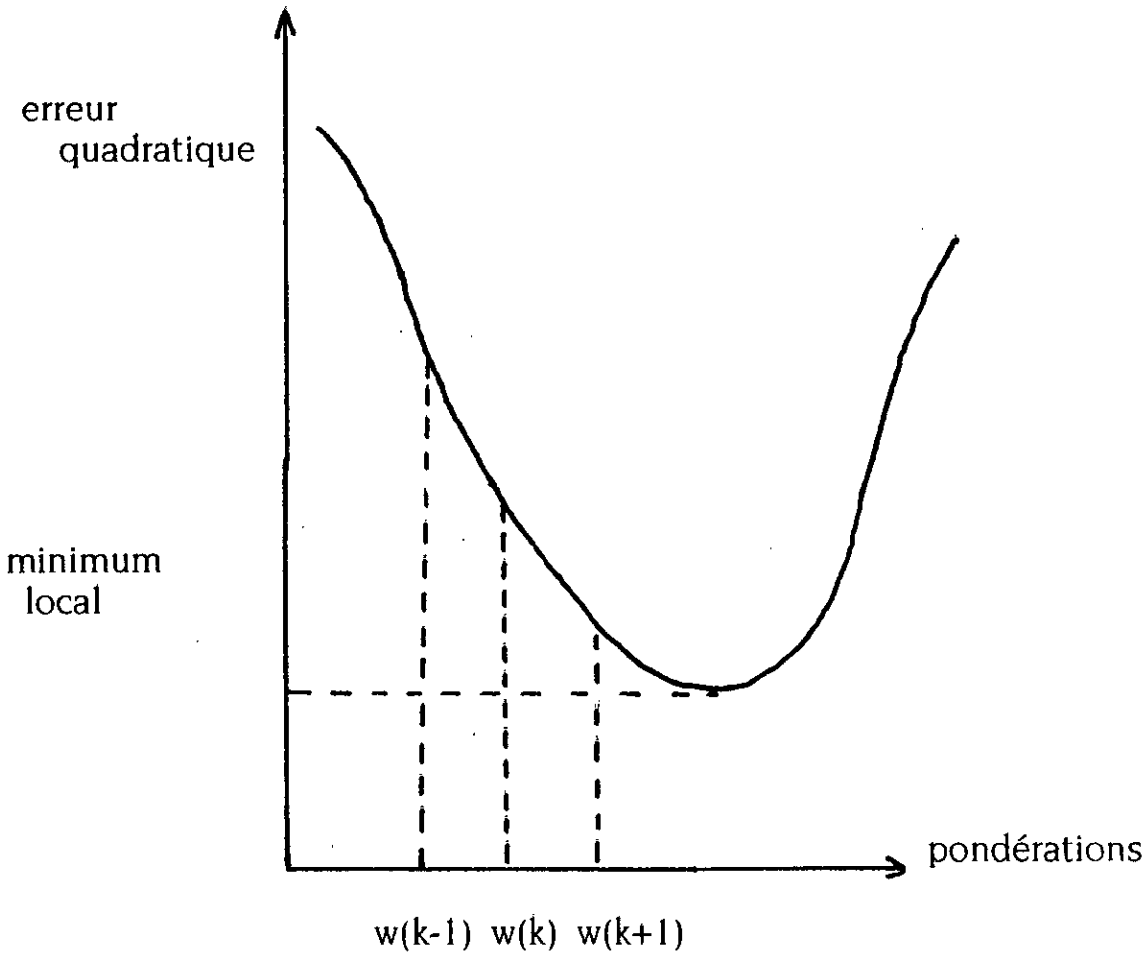


fig 2-3 :descente du gradient, sur la courbe d' erreur.

2.2.1 Résumé de l' algorithme (BP) :

Soit un réseau à N neurones en entrée, M en sortie, Nk le nombre de neurones de la couche cachée numéro k.

1. Initialiser tous les poids W_{ij} des connexions, aléatoirement.

2. Présenter une forme (X_1, ..., X_n) en entrée, et la sortie désirée (Y_1, ..., Y_m).

3. Calculer les sorties de chacun des neurones des couches cachées et de la couche de
sortie par :
pour la première couche :

$$o_j^1 = f\left(\sum_{i=1}^{n} w_{ij}(k)\, x_i \right)$$

pour la deuxième couche :

$$o_j^2 = f\left(\sum_{i=1}^{n1} w_{ij}(k)\, o_i^1 \right)$$

etc ...

4. Modifier les connexions recursivement :

W_{ij} connexions entre le neurone i et le neurone j

O_j représente la sortie du neurone j.

$$W_{ij}(k+1) = W_{ij}(k) - \mu(k)\, d_i\, O_j$$

avec : $d_i = 2 (S_i - Y_i) f'(I_i)$ pour la couche de sortie

$$d_i = \sum_h d_h\, W_{ij}(k)\, f'(I_i)$$ pour la couche cachée

5. Refaire les étapes 2 à 4 jusqu' à stabilisation du réseau.

Remarques concernant l' algorithme (BP) :

L' expérience à montrer qu' il est généralement préférable d' initialiser les poids à des valeurs comprises entre 0 et 1 pour éviter de destabiliser le système des le départ .[4]

Le pas du gradient est choisi entre 0 et 1, et on le diminura au fûr et à mesure pour atteindre une valeur fixe.

- Le choix des exemples avec les quels on doit entrainer le réseau est trés important, un choix judicieux de ces exemples poura faciliter la généralisation sur des exemples non appris. Cela à pousser les chercheurs à faire l' étude statistique des données d' apprentissage pour trier celles qui sont le plus représentatives du phénomène qu' on souhaite avoir. [2]

2.2.2 Difficultés et limites de l' algorithmes (BP) :

Des questions restent posées lors de l' utilisation de cet algorithme: à savoir:

- combien faut-il utiliser de couches ?

- combien doit-il y avoir de neurones dans chaque couche ?

Les réponses sont données que par l' expérience, et il n' existe pas de regle théorique qui puisse fixer le nombre de neurones ou de couches dans le réseau [10].

- Le problème de minimisation n' est pas simple à résoudre, en effet la surface d' erreur peut présenter des caractéristiques peu satisfaisantes telles que:

- des minima locaux qui empèchent la convergence vers le minimum global.

- des plateaux, où les pentes sont trés faibles. [4]

- le pas du gradient peut être difficile à fixer, s' il est faible la convergence risque d' être trés lente; s' il est elevé on risque d' osciller.

- ajoutons à cela que l' algorithme n' a aucune preuve théorique de sa convergence. [10]

2.2.3 Applications de l' algorithmes BP: [1] [10]

Malgrés les difficultés qu' on vient de citer, l' algorithme de rétropropagation (BP) s' est révélé assez performant dans la résolution de plusieurs problèmes tels que:

- La reconnaissance de formes géométriques

- Traitement et analyse des signaux

- Commande des processus

- La classification

- Filtrage du bruit

- Synthèse de la parole

- Diagnostique médical

- Prédiction financière

pour ne citer que ceux là.

C' est au fait, l' algorithme qui a permi aux réseaux neuronaux d' emmerger aprés s' être eclipser pendant un certain temps.

2.3 Méthode d' optimisation aléatoire (ROM) :

Cette méthode est basée sur des techniques aléatoires, elle a été utilisée avec succés dans divers problèmes d' optimisation.
Elle devient performante, notamment dans des cas spécifiques quand les caractéristiques de la fonction sont difficiles à calculer, et surtout quand on veut trouver le minimum global d' une fonction sachant qu' elle présente des minima locaux.
De plus, la nécessité d' utiliser cette méthode survient lorsque la dimension du réseau neuronal devient large, et que le temps de calcul avec l' algorithme de rétropropagation (BP) peut être problématique [8], [11].
La méthode d' optimisation aléatoire, converge vers un minimum global avec une probabilité égale à 1 dans un ensemble compact [8].
L' idée de cette algorithme dont la démonstration est présentée dans l' annexe 2, est d' entacher les poids du réseau d' une séquence de bruit

blanc et calculer la sortie du réseau avec les nouveaux poids, si l' erreur entre celle-ci et la réponse désirée est inférieure à la précédente, on garde ces poids sinon on garde la séquence précédente; on refait l' opération jusqu' à obtention de l' erreur fixée. Cf figure (2-4).

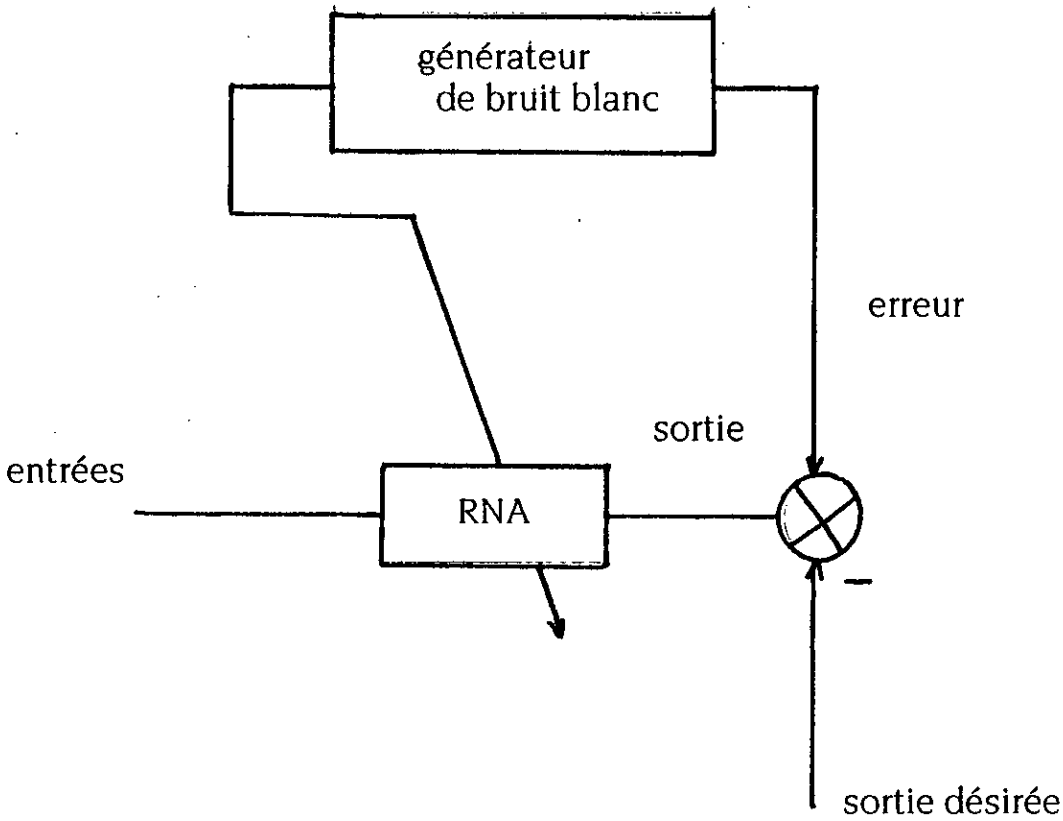


fig 2-4 : Adaptation des poids par ROM.

2.3.1 Résumé de l' algorithme (ROM) : [8]

1. Choisir un point initial x (0), b (0) = 0 pour k = 0

2. Générer un vecteur aléatoire Gaussien (k).

   si (x (k) + g(k)) $\in$ S aller en 3

   sinon aller en 4

3. (i) si f (x(k) + g(k)) < f (x (k))

   poser : x (k+1) = x(k) + g(k)

et $b(k+1) = 0.4\, g(k) + 0.2\, b(k)$

(ii) si $f(x(k) + g(k)) \geq f(x(k))$ et

$f(x(k) - g(k)) < f(x(k))$

poser : $x(k+1) = x(k) - g(k)$

et $b(k+1) = b(k) - 0.4\, g(k)$

sinon poser $x(k+1) = x(k)$

et $b(k+1) = 0.5\, b(k)$

4. Si le minimum est satisfaisant, stopper les calculs. Sinon poser : $k = k + 1$ et aller en 2.

f : étant la fonction à minimiser

g (k) : le vecteur gaussien

b (k) : étant la moyenne du vecteur gaussien g (k).

Remarque sur l' apprentissage :

dans le cas de l' apprentissage du réseau neuronal, la fonction à minimiser sera l' erreur quadratique qui elle est en fonction des poids, soit alors :

$$f(x) = e_k(w) .$$

Remarques sur l' algorithme (ROM)

- Pour les besoins théoriques de l' algorithme, le vecteur poids doit appartenir à un ensemble compact. [8] par exemple, on prendra W comme étant un vecteur dont les composantes Wij sont inferieures à 100, ce n' est que dans ce cas qu' on sera capable de trouver le minimum global de la fonction d' erreur.

- Le choix de la variance est capital pour améliorer la vitesse d' apprendrentissage. Généralement, on choisi la variance entre 0 et 1 pour des raisons purement expérimentales.

2.3.2 Applications de l' algorithme (ROM)

- La principale application pour laquelle est utilisée la méthode d' optimisation aléatoire étant la simulation des fonctions . Récemment, on a montré que n' importe quelle fonction peut être approximée par un réseau à une seule couche .[8]

- Avec les réseaux neuronaux, cette méthode a donné de trés bon résultats dans le domaine de prévision météorologique .

2.4 Conclusions :

Les deux algorithmes d' apprentissage, présentés ici ont fait leurs preuves dans leurs domaines d' application respectifs qu' on a cité.

La méthode du gradient (BP) présente certains désavantages comme celui du minimum local, mais celui-ci peut donner satisfaction pour l' évaluation de l' erreur finale, et c' est souvent le cas. [4]

En plus de ça, ni la rétropropagation du gradient ni la méthode d' optimisation aléatoire, ne peuvent déterminer l' architecture exacte du réseau, et cela est du ressort de l' expérience.

- Les travaux de comparaison des deux algorithmes ont été effectués dans le but de trouver le meilleur outil pour l' entrainement des réseaux multicouches. Les résultats découlants de cette comparaison ont été avantageux à la méthode d' optimisation aléatoire, du point de vue rapidité (nombre d' itérations relativement inferieur) et présentant une erreur finale inférieure à celle obtenue par la rétropropagation . [8], [11]
Toute fois, ces résultats sont spécifiques à certaines applications et la généralisation n' est pas pour bientôt. C' est pour cela qu' on va utiliser dans ce qui suit les deux algorithmes, pour pouvoir comparer leurs résultats dans notre application .
Enfin on notera, que l' algorithme de rétropropagation présente deux manières de faire, soit l' apprentissage point par point c' est à dire rester avec un point un certain nombre d' itérations, le passage au point suivant ne se fait que si l' erreur a atteint son seuil désiré. Soit l' apprentissage par passage multiple sur tous les points, chaque point n' est présenté qu' une seule fois durant lamême itération . Ces deux versions sont discutés au chapitre 4.

CHAPITRE 3

METHODES DE LINEARISATION DES CAPTEURS NON LINEAIRES

3.1. Introduction:

Il n' existe point de domaine scientifique, où l' ingérieur n' utilise pas l' opération de mesure, cette notion joue un rôle capital en faveur du bon fonctionnement des processus en général.
Les activités à caractère industriel ont pû grâce à elle,évoluer rapidement, et le domaine de la Robotique est un véritable témoin sur ce qu' on vient d' affirmer. En régulation automatique," une bonne mesure est liée à la qualité de réglage qu' on veut avoir " ? Et part ceci à un contrôle crédible des grandeurs physiques associés au processus. Les outils prodigués par l' électronique, ont orienté la mesure à ce que toute grandeur physique qu' on veut évaluer soit traduite en grandeur électrique, ce qui est du rôle du capteur. [17]
La nature de la grandeur électrique, définie le type du capteur. Un capteur actif fonctionne en générateur et délivre à sa sortie une tension, un courant ou une charge. Un capteur passif quant à lui, délivre à sa sortie une variation d' impédance qui nous renseigne sur la variation du mesurande.
Comme toute mesure physique, le signal de sortie du capteur (signal de mesure) peut être entaché d' erreurs qui sont systématiques ou aléatoires, ces dernières sont dûes principalement à des grandeurs de perturbations appelés aussi grandeurs d' influence [17], doivent être réduites par différents procédés, comme l' isolement, le blindage etc. Dans le cas contraire, ces grandeurs doivent être prises en considération notamment dans l' étalonnage des capteurs, en les fixant à des valeurs bien connues.
Souvent l' adjonction de circuits annexes aux capteurs, donne de bons résultats en matière de compensation des grandeurs de perturbation.
- Dans le présent chapitre, nous vous présentons certains exemples de capteurs non linéaires, ainsi que les différentes méthodes de linéarisation qui existent et cela pour qu' ils fassent l' objet de comparaison avec la méthode de linéarisation par réseaux de neurones qu' on traitera ultérieurement.

3.2. Non linéarité des capteurs:

3.2.1. Modèle mathématique du capteur:

Le modèle général du capteur est défini comme suit:
Cf figure (3 - 1)

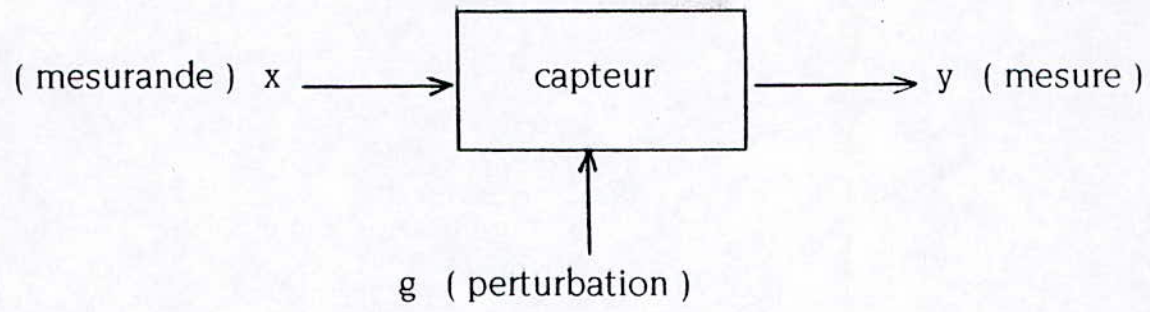


Fig 3-1: Schéma bloc d' un capteur

$$y = F\ (x, g)$$

ici, il sagit d' une seule grandeur de perturbation,il peut y avoir plusieurs.

y étant un signal électrique, et x une grandeur non électrique.

Dans le cas de capteur passif où il y a variation d' impédence, celle-ci est traduite en signal électrique à l' aide d' un conditionneur de capteurs passifs.

La variation de la mesure est donnée par le développement limité autour du point (x , g) à l' ordre 2.

$$\Delta y = \frac{\partial F}{\partial x}\Delta x + \frac{\partial F}{\partial g}\Delta g + \frac{1}{2}\frac{\partial^2 F}{\partial x^2}(\Delta x)^2 + \frac{1}{2}\frac{\partial^2 F}{\partial g^2}(\Delta g)^2 + \frac{1}{2}\frac{\partial}{\partial x}\frac{\partial F}{\partial g}(\Delta x\,\Delta g) + \frac{1}{2}\frac{\partial}{\partial g}\frac{\partial F}{\partial x}(\Delta g\,\Delta x)$$

le terme : $S = \dfrac{\partial F}{\partial x}$ est appelé **sensibilité statique** du capteur

le terme : $Sg = \dfrac{\partial F}{\partial g}$ est appelé **sensibilité aux bruits** .

le terme : $S = \dfrac{\partial^2 F}{\partial^2 x}$ est appelé la **dérive de la sensibilité** en fonction de x.

3.2.2. Linéarité et non linéarité d' un capteur:

- Le capteur est dit linéaire, si :

Sa sensibilité au mesurande, S est indépendante de celui-ci, c' est à dire:

$$S = \frac{\partial F}{\partial x} = \text{cste} \quad \text{ou} \quad \frac{\partial^2 F}{\partial^2 x} = 0$$

il est dit non linéaire, dans le cas contraire : $\frac{\partial^2 F}{\partial x^2} \neq 0$

Le capteur peut être linéaire par morceaux, c' est à dire une linéralité dans une zone où la sensibilité est indépendante du mesurande.

La sensibilité telle qu' elle est fournie par le constructeur, nous permet d' estimer l' ordre de grandeur de la réponse du capteur connaisant les variations du mesurande [17] .

En régime dynamique, on dira que le capteur est linéaire, s' il est régit par une équation différentielle linéaire, et non linéaire dans le cas contraire. Dans ce cas on parlera de sensibilité dynamique qui dépend de la fréquence du mesurande, ou fonction de transfert .

3.2.3. Etalonnage du capteur :

L' étalonnage du capteur est l' opération qui consiste à établir,à partir de données entrées / sorties effectuées sur le capteur, la forme algébrique ou graphique qui lie le mesurande x à la mesure y, en tenant compte des grandeurs de perturbations, et de l' alimentation en amplitude et en fréquence nécessaire au fonctionnement du capteur. [18]

L' étalonnage peut être direct [18], à partir des couples (x_i , y_i) on détermine la forme algébrique $y = F(s)$.

Il peut être par comparaison, cette fois on utilise un capteur étalon, qui nous donnera les mesurandes x_i , les mesures y_i étant fournies par le capteur qu' on veut étalonner.

La linéarité sous sa forme graphique, décrit le degré de concordance entre le diagramme d' étalonnage et une droite référence [18], déterminée souvent par la méthode des moindres carrés. Pour un échantillon de mesures (x_i , y_i), le coefficient de corrélation est calculé par:

$$r = \sqrt{\frac{\sum_{i}(y_i - y_m)}{\sum_{j}(y_j - y_m)}}$$

y_i : l' image de xi prise sur la droite des moindres carrés.
y_j : la mesure de xj

$y_m = \frac{1}{N}\sum y_i$ étant la moyenne des mesures.

Ce coefficient, nous renseigne sur le degré de linéarité du capteur concerné Cf figure (3 - 2)

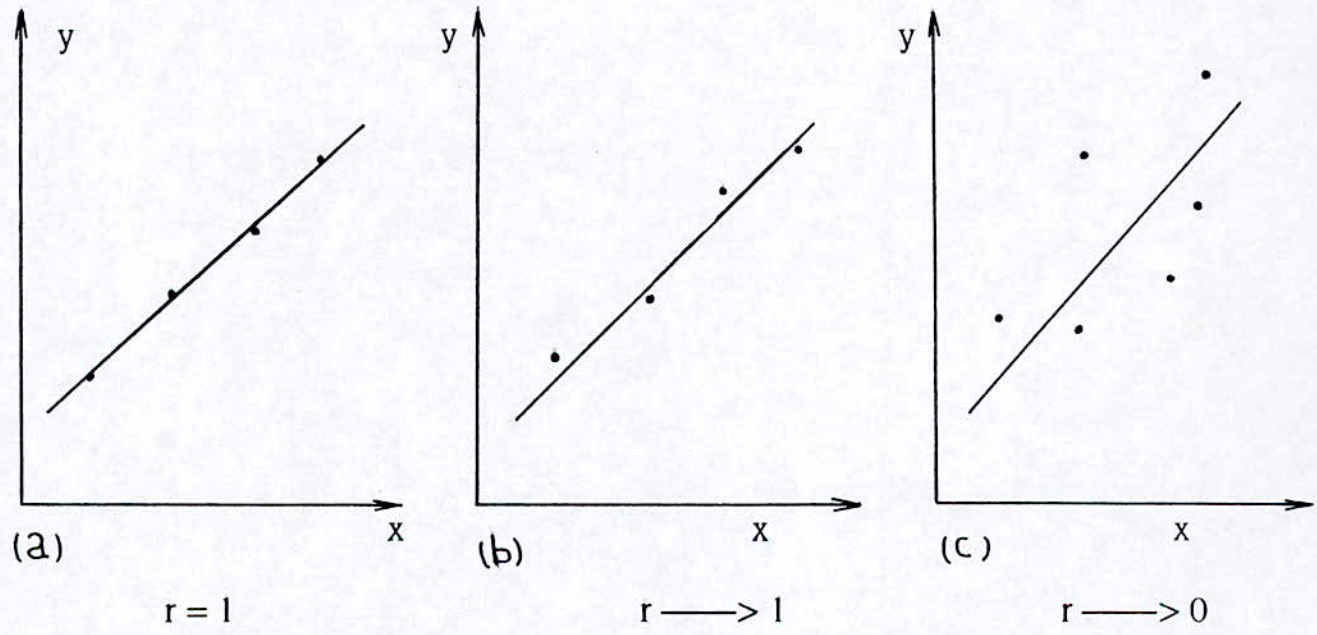


Fig 3 -2 : Linéarité et non Linéarité vues à partir de la forme graphique.

Le capteur de la figure (3 - 2) est :

- Linéaire dans le cas (a)

- Presque linéaire ou trés proche dans le cas (b)

- Non linéaire dans le cas (c)

3.3. Exemples de capteurs non linéaires :

Nous citerons ici trois types de capteurs non linéaires à savoir :

a. Résistances thermiques :

Ils forment une famille typique de capteurs nonlinéaires,ils sont divisés en deux groupes: [17]

*les resistances metalliques:

$$R(T)=R_0(1+aT+bT^2+cT^3)$$

*les thermistances :

$$R(T)=R_0 \exp [B(1/T-1/T_0)]$$

b.capteur de debit d'air :

Un autre exemple de capteur non linéaire, serai celui utilisé pour la mesure de la vitesse d' un débit d' air [19]. Pour cela, on utilise un fil placé dans un écoulement et traversé par un courant constant, celui-ci chauffe à une température supérieur à celle de l' air, on aura donc un échange de chaleur par convection qui dépend de la vitesse du fluide et de l' écart de température.[17]

La puissance joule dissipée par le fil étant :

$$Pj = R(T)\, I^2$$

La puissance échangée par convection du fluide à la température Ta est :

$$Pc = h\, Se\, (T - Ta)$$

h : Coefficient d' échange thermique

Se : Surface latérale du capteur

à l' équilibre on a : $Pj = Pc$

$$R(T).I^2 = h\, Se\, (T - Ta)$$

avec : $h = a + b\sqrt{v}$ (formule de king)

v : Vitesse du fluide

d' où :

$$R(T).I^2 = (a + b\sqrt{v}) \, Se \, (T - Ta)$$

qui est une relation non linéaire, qui lie la résistance R (T) à la racine carrée de la vitesse du fluide V.

c. Capteur de composition gazeuse :

On ajoutera aux deux exemples précédents, un troisième issu de la famille des capteurs chimiques qui sont généralement non linéaires, soit alors un capteur de composition gazeuse, appelé à électrolyte solide [17] . Il est composé de :

- Deux conducteurs électroniques (inertes).

- Un électrolyte, imperméable aux gaz.

- Le gaz qu' on désire analyser, et qui exerce les pressions partielles P et P_0 de part et d' autre de l' électrolyte.

Le capteur donnera alors, une tension e qui suit la loi de Nernst [17] :

$$e = \frac{R\,T}{2n\,F} \log \frac{P}{P_0}$$

avec R : Constante des gaz parfaits

F : Constante de Faraday

n : Le nombre d' électrons échangés entre le gaz et les conducteurs durant la réaction.

3-4. Pourquoi linéariser un capteur :

Pourquoi linéariser ? Telle est la question qu' un ingénieur se pose à chaque fois qu' il doit utiliser un capteur. C' est vrai que tout dépend de l' application à laquelle on veut joindre le capteur, des fois on a pas besoin de linéariser pour des causes qu' on juge valables, et qui servent à un bon fonctionnement de cet instrument. Mais souvent ce n' est pas le cas, et une linéarisation adéquate s' avère indispensable.

Quand la caréctéristique du capteur est linéaire, le signal de sortie peut dépasser le champs de visualisation [19]; mais ce problème peut être réglé par simple mise à l' échelle des grandeurs. Seulement, si le capteur est non linéaire, on se trouve confronter à d' autres genres de problèmes peut être plus difficiles à régler. De plus, la linéarisation à l' avantage d' augmenter la précision des capteurs et leur étendue de mesure [19] .

Généralement, on approxime les parties non - linéaires de la caractéristique du capteur par des segments de droites autour de points de fonctionnement. Cette méthode, a le désavantage d'avoir une étroite étendue de mesure qui diminues, encore plus si le capteur est trés fortement non - linéaire. En régulation automatique, la linéarisation d' un bloc non - linéaire permet de simplifier la synthèse de la loi de commande en utilisant la théorie linéaire de commande.

3-5. Différentes méthodes de linéarisation d' un capteur :

Il existe en général deux principales méthodes de linéarisation des capteurs non linéaires.

* Linéarisation analogique
* Linéarisation numérique

- La première méthode utilise des circuits analogiques non linaires qui sont placés souvent aprés le conditionneur du capteur. Ces circuits ont pour entrée le signal de mesure du capteur et produisent en sortie un signal linéaire en fonction du mesurande. Ils peuvent être conçus à base d' éléments non linéaires, comme les diodes et les transitors [29], dans le but de réaliser la fonction inverse de celle du capteur obtenue par étalonnage.

- La deuxième méthode utilise un calculateur numérique, muni d' une mémoire ROM dans laquelle on implémente une table de mesure prises sur le capteur. L' avantage de cette méthode est la possibilité de linéarisation en prenant en compte les effets des grandeurs de perturbation, comme la

température, la pression, l' humidité etc.

3.5.1. Linéarisation analogique des capteurs :

Cette méthodes est utilisée, surtout quand il s' agit de concevoir des circuits linéarisants à faible côut et à trés grande rapidité de traitement [19] . Néanmoins, elle est limitée par le fait qu' elle est opérationnelle quand le signal de mesure dépend uniquement du signal d' entrée, ou quand les grandeurs de perturbations sont invariables dans le temps ; autrement dit si le capteur a une grande sensibilité aux bruits, il serait difficile de linéariser le capteur par un circuit analogique.

La méthode consiteste à étalonner le capteur,comme déja vu, en maintenant les grandeurs d' influences constantes, on obtient enfin une relation algébrique de la forme :

$$m = a_1 + a_2.e + a_3.e^2 + ... + a_n.e^{n-1} \qquad (3 - 1)$$

m étant le mesurande et e le signal de sortie.

les coefficients a sont déterminés, en résolvant un système de n équations à n inconnus par le biais de n couples d' étalonnage (mi , li). Aprés cela, on déterminera l' expression algébrique du circuit de linéarisation qui relie l' entrée Vin à la sortie V_0 , comme suit :

$$V_0 = b_1 + b_2.Vin + b_3.Vin^2 + ... + b_n.Vin^{n-1} . \qquad (3 - 2)$$

avec : $b_k = K\, a_k$.

Comme exemple de circuit linéarisant, on prendra un du 4 ième ordre [19] qui est constitué de multiplieurs analogiques qui génèrent les termes :

Vin^2 , Vin^3 , ... et d' un amplificateur différentiel qui joue le rôle de sommateur suivant les valeurs de bk.Cf figure (3 - 3)

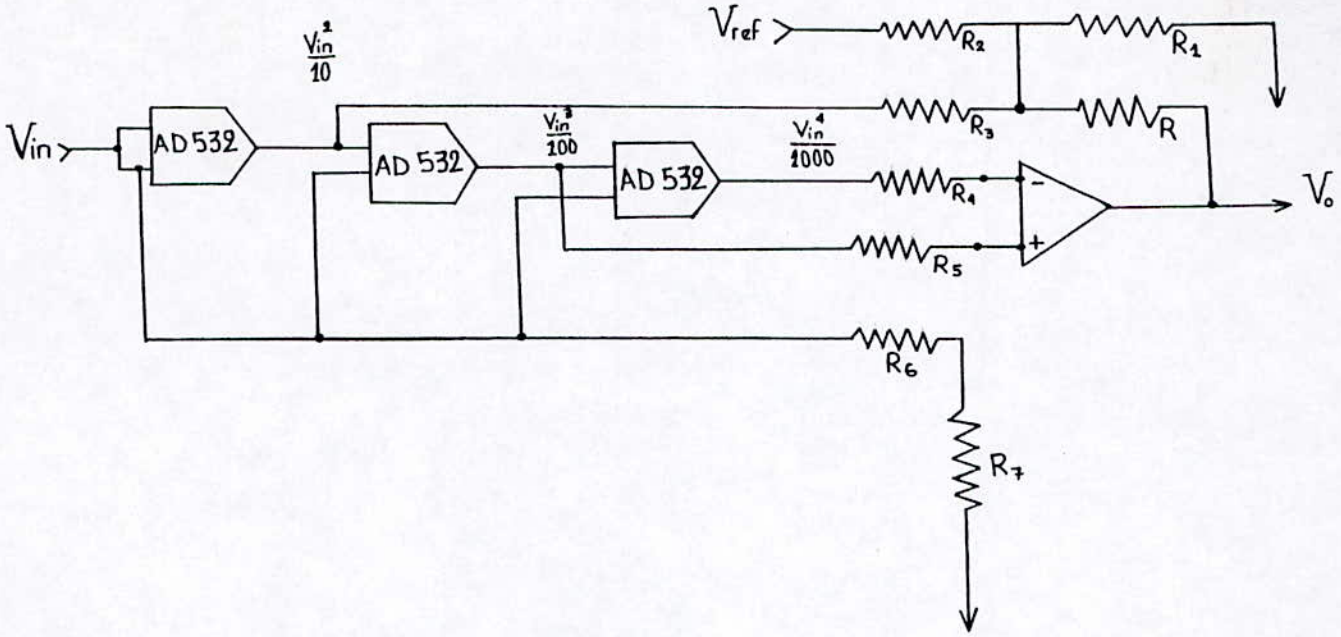


fig 3 - 3 : Exemple de circuit linéarisant, générant un polynome du 4ième ordre.

Si maintenant la fonction qui régit le fonctionnement du capteur est déterminée théoriquement et cela par les différentes lois physiques,le circuit linéarisant sera plus facile à réaliser, et cela par conbinaison d' amplificateurs logarithmiques ou exponentiels pour synthétiser n' importe quelle fonction. La fonction logarithmique et son inverse,peuvent être réalisés par des circuits du type 755 N I C d' analog Devices, car il offre une bonne précision qui peut aller jusqu' à 6 décades [19] .

- Pour l' exemple de la thermistance, on a :

$$R(T) = R_0 \exp[B(1/T - 1/T_0)]$$

La fonction inverse est donc :

$$T = 1 / (1/T_0 + 1/\beta_0 \ln(R/R_0))$$

Il s' agit alors de réaliser la fonction Ln x et la fonction 1/x .

- En ce qui concerne la linéarisation des capteurs passifs, il existe d' autres méthodes de linéarisation analogique. On peut citer, celle ou l' on associe des capteurs résistifs en série pour compenser les non linéarités entre elles [17] . Soit :

$$R_1(x) = R_{01}(1 + a_1 m + b_1 m)$$

$$R_2(x) = R_{02}(1 + a_2 m - b_2 m)$$

L' association en série donne :

$$R = (R_{01} + R_{02})\left(1 + \frac{R_{01} a_1 + R_{02} a_2}{R_{01} + R_{02}} . m\right)$$

avec comme condition:

$$\frac{R_{01}}{R_{02}} = \frac{b_2}{b_1}$$

Le meilleur choix pour les capteurs passifs sera l' utilisation d' un pont, celui-ci , qui effectue le rôle de conditionneur, nous introduira une non linéarité dûe à la tension de déséquilibre du pont, qui peut être évitée, par le dispositif présenté à la figure (3 - 4)

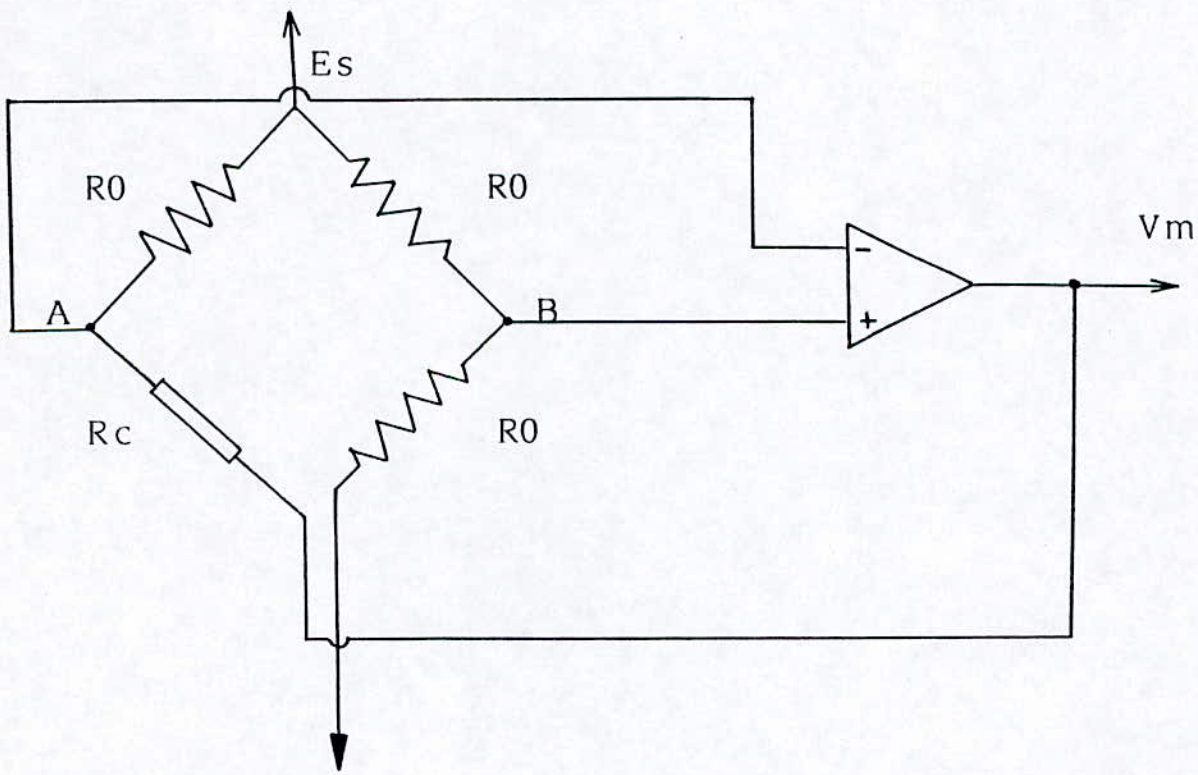


fig (3 - 4) : Linéarisation par réaction sur la tension de désiquilibre du pont .

$$V_B = \frac{Es}{2}$$

$$V_A = \frac{Rc}{R0 + Rc} Es + \frac{R0}{R0 + Rc} Vm$$

$$V_A = V_B$$

donc : $$Vm = - \frac{Es}{2} \cdot \frac{\Delta Rc}{R0}$$

avec : $\Delta Rc = Rc - R0$

3.5.2. Linéarisation numérique des capteurs :

Cette méthode,comme nous l' avons déja mentionné,est basée sur, l' implémentation d' un programme dans une mémoire, qui permet de nous délivrer la valeur du mesurande m à chaque fois qu' on lui donne la valeur de la mesure e . Ceci étant, le signal de sortie du capteur doit être conditionné et digitalisée à travers un conditionneur A/D avant d' attaquer le pont d' entrée du calculateur numérique. L' avantage de cette méthode n' est pas négligeable, car elle permet de linéariser toute en gardant en présence l' influence des perturbations, chose qui n' est pas possible à l' aide du circuit linéarisant analogique. De cette façon là, les mesures de la grandeur physique deviennent sans aucun doute plus précises.

3.5.2.1. Linéarisation sans grandeur de perturbation :

Dans ce cas le capteur en question n' est pas sensible aux grandeurs de perturbations.

- Aprés avoir déterminer les coefficients ak de l' équation (3 - 1) par étalonnage,le programme implémenté doit calculer la valeur de m, par simple évaluation de l' équation (3 - 1), mais cela peut côuter cher en temps de calcul des multiplications mises en jeu. Une méthode plus judicieuse consiste à mettre en mémoire EPROM [19] une table qui relie chaque mesurande mi à sa mesure ei , l' entrée du calculateur sera donc la mesure e et sa sortie le mesurande m, qui ne sera d' autre qu' une interpolation linéaire des deux points qui définissent l' intervale auquel appartient e .

$$m = m_k + (e - e_k) \; F (e_k, e_{k+1})$$

avec :

$$F (e_k, e_{k+1}) = \frac{m_{k+1} - m_k}{e_{k+1} - e_k}$$

Si on désire plus de précision, on utilise une interpolation quadratique qui prend en compte trois points, à savoir (e_{k-1}, m_{k-1}),(e_k, m_k), et (m_{k+1}, e_{k+1}), l' expréssion de m sera :

$$m = m_k + (e - e_{k-1}) F (e_{k-1}, e_k) + (e - e_{k-1}) (e - e_k) F (e_{k-1}, e_k, e_{k+1}) \qquad (3\text{-}3)$$

avec :

$$F (e_{k-1}, e_k, e_{k+1}) = \frac{F (e_k, e_{k+1}) - F (e_{k-1}, e_k)}{e_{k+1} - e_k}$$

3.5.2.2. Linéarisation avec une seule grandeur de perturbation :

Cette fois - ci, l' étalonnage du capteur, s' effectue en faisant varier le mesurande m pour obtenir la mesure e tout en maintenant la grandeur de perturbation fixe. Cette procédure est répétée pour plusieurs paliers de la grandeur de perturbation qui sont à l' intérieur d' une plage bien définie. Dans ce cas les coefficients ak de l' équation (3 -1) deviennent une fonction de la perturbation, qu' on appelera g . Soit alors :

$$m = a_1(g) + a_2(g) e + a_3(g) e_2 + \ldots\ldots\ldots\ldots + a_n(g) e_{n-1} \qquad (3\text{-}4)$$

si on supose : $a_k(g_i) = a_{ki}$

on aura pour différentes valeurs du couple (m,e) :

$$\begin{cases} m_{1i} = a_{1i} + a_{2i} e_{1i} + \ldots\ldots\ldots\ldots + a_{ni} e_{1i}^{n-1} \\ m_{2i} = a_{1i} + a_{2i} e_{2i} + \ldots\ldots\ldots\ldots + a_{ni} e_{2i}^{n-1} \\ \quad . \\ \quad . \\ m_{ni} = a_{1i} + a_{2i} e_{ni} + \ldots\ldots\ldots\ldots + a_{ni} e_{ni}^{n-1} \end{cases} \qquad (3\text{-}5)$$

En resolvant le système (3-5) , on obtient les différents aki corréspondants à g = gi
pour les différentes valeurs de g , soit g_1, g_2, ... ,gm le polynôme qui lie a_k (g) à g sera :

$$a_k(g) = b_{1k} + b_{2k}\, g + \ldots + b_{mk}\, g^{n-1} \qquad (3\text{-}6)$$

connaisant les a_{ki} et les g_i, les m coefficients b_{1k}, b_{2k}..., bmk sont déterminés par le système suivant:

$$\begin{cases} a_{k1} = b_{1k} + b_{2k}\, g_1 + \ldots + b_{mk}\, g_1^{m-1} \\ a_{k2} = b_{1k} + b_{2k}\, g_2 + \ldots + b_{mk}\, g_2^{m-1} \qquad (3\text{-}7) \\ \cdot \\ \cdot \\ \cdot \\ a_{km} = b_{1k} + b_{2k}\, g_k + \ldots + b_{mk}\, g_k^{m-1} \end{cases}$$

Le travail du programme, consiste à lui présenter un couple (e, g), celui-ci calcule les coefficients a_k (g), ainsi que le mesurande m aprés substitution des a_k (g) dans l' équation (3 - 4) . Toujours est-il que cette méthode peut nous valoir baucoups de temps de calcul, un travail plus adéquat, sera de procéder par interpolation aprés avoir établi deux tableaux, le premier contenant les couples (m , e), le deuxième contiendra les couples (g , a_k (g)), soit alors :

$$m(g,e) = m_{k,l} + [\, m_{k,l+1} - m_{k,l} \,](e-l) + [\, m_{k+1,l} - m_{k,l} + (\, m_{k+1,l+1} - m_{k+1,l} - m_{k,l+1} + m_{k,l} \,)(e-l) \,](g-k) \qquad (3\text{-}8)$$

3.6. Conclusion :

A l' issue de ce chapitre, on peut conclure,en ce qui concerne la linéarisation, que dés qu' il s' agit de capteurs peu sensibles aux perturbations, la linéarisation analogique est la plus sollicitée vu son faible côut de réalisation et sa rapidité d' exécution, par contre on a tendance à utiliser la linéarisation numérique dés que l' influence des perturbations sur le capteur devient importante.
Une autre technique de linéarisation par RNA fera l' objet des chapitres 4 et 5.

CHAPITRE 4

Simulation de fonctions non linéaires à l' aide des RNA

4.1 Introduction :

Comme déjà mentionné, l' application à laquelle nous joindrons notre RNA étant la simulation des caractéristiques non-linéaires statiques de capteurs, il parait tout à fait logique d' entamer le travail par des fonctions non linaires théoriques connues, afin de mieux connaitre le comportement du RNA.

Le travail préalable sur des fonctions non linéaires analytiques, nous permet d' acquérir une certaine expérience sur les problèmes posés lors de l' entrainement du RNA à savoir : l' initialisation des pondérations, l' apprentissage et la généralisation, et ceci en vue d' une application à une caractéristique non linéaire réelle d' un capteur. (voir chapitre 5).

Ainsi, on aura un premier apperçu sur le comportement des deux algorithmes d' apprentissage, et leur collaboration éventuelle dans le but d' augmenter les performances du RNA. Le choix de la méthode d' apprentissage se fera dans cette partie.

Dans ce chapitre, l' étude est basée sur deux types de non linéarité en l' occurence, celle de type **dûr** et celle de type **mou** . Des remarques ont été faites aprés chaque étape d' étude et qui constituent le fruit de constructions pratiques lors du travail de simulation .

Enfin, des mesures adéquates ont été prises en compte pour l' amélioration du fonctionnement du RNA avec chaque type de non linéarité et qui sont à leur tour tirées de constatation expérimentales .

4.2 Mise au point :

Ce paragraphe est nécessaire pour la suite du travail. Il concerne une mise au point sur l' utilisation de l' algorithme de rétropropagation (BP), ainsi que le choix de l' architecture du RNA .

En effet, l' apprentissage par BP, consiste comme prévu à apprendre un certain nombre d' exemples, afin de reproduire leur images approchées. Seulement cet algorithme a deux façons de voir le problème :

* La première étant d' apprendre les exemples **un par un**, c' est à dire on présente un seul point d' entainement au RNA, la méthode de BP qui effectue une descente sur la surface d' erreur par la technique du gradient fait un certain nombre d' itérations jusqu' à ce que l' erreur de sortie soit inférieure à un seuil prédéfini, à ce moment là on passe à l' exemple suivant, ainsi de suite.

** La deuxième, étant de présenter au RNA **tous les exemples à la fois**, calculer la sortie de chaque exemple d' apprentissage ainsi que son erreur. Ce passage par tous les points constitue une seule itération, les poids sont corrigés aprés calcul de chaque erreur d' apprentissage. Ce travail est réitéré un certain nombre d' itérations jusqu'à ce que l' erreur soit inférieure à un seuil donné .

Bien que la deuxième méthode est plus prometteuse, car elle permet de revenir sur chaque exemple appris une fois que tous les autres exemples ont été présentés, il est interressant de tester la première façon d' apprentissage pour notre application . Pour cela un réseau de deux couches cachées de trois neurones chacune a été utilisé Cf figure (4-1), avec un nombre de points à apprendre variant entre 50 et 150 points, la fonction à simuler est :

$$f(x) = x^2$$

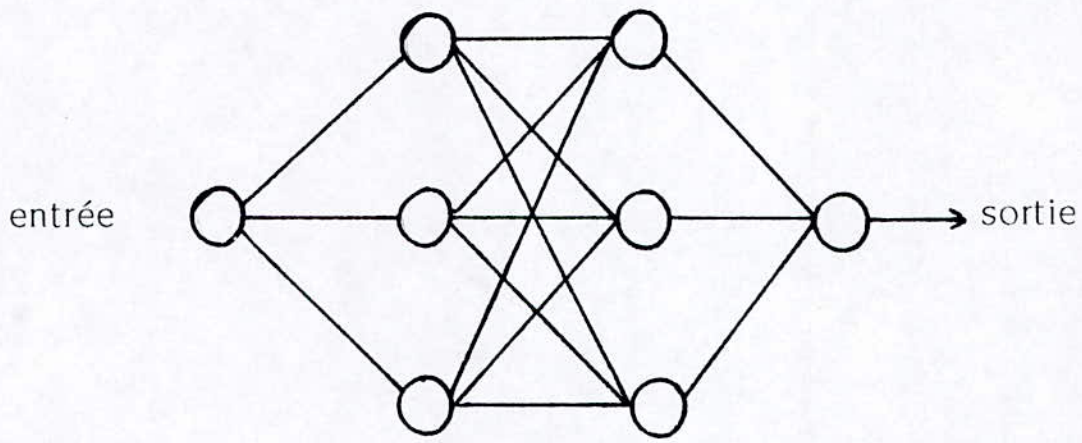


Fig 4-1 : Structure du RNA utilisée

Les résultats d' apprentissage de la première méthode sont présentés dans le tableau (4-1)

nombre de points d' apprentissage	seuil de l' erreur	nombre d' itérations
50	0.001	18000
150	0.001	31091

Tab 4-1 : Résultats d' apprentissage de la première méthode par BP.

Les résultats de cette méthode n' étaient pas satisfaisants, dans la mesure où on arrive pas à maintenir l' erreur commise sur chaque point appris fixe ou trés proche du seuil de l' erreur déjà défini aprés passage d' un point à un autre. Par exemple dans le cas de 50 points à apprendre avec un seuil d' erreur de 10^{-3} , l' erreur commise sur le premier point passe de 10^{-3} à 10^{-2} aprés que le RNA ait appris les 149 point restant, cet accroissement de l' erreur influe surement sur la qualité de la généralisation .

Voici maintenant les résultats d' apprentissage obtenus par la deuxième méthode, Cf tableau (4-2).

nombre de points d' aprentisage	seuil de l' erreur	nombre d' itérations
50	0.001	380
150	0.001	1200

Tab 4-2 : Résultats de la deuxième méthode d' apprentissage par BP.

Par comparaison des deux méthodes, il est clair que la deuxième nécessite beaucoups moins d' itérations, donc moins de temps de calcul de plus celle-ci ne présente pas l' inconvénient de l' accroissement de l' erreur dont fait l' objet la première méthode .

Pour la suite de notre travail, nous adopterons la deuxième technique pour l' entrainement du RNA par BP.

Pour ce qui concerne l' architecture du réseau, des difficultés ont été rencontrées au niveau de l' augmentation du nombre de points à apprendre vu la petite taille du réseau [12]. La décision prise était d' opter pour un réseau à deux couches cachées mais cette fois-ci à 10 neurones chacune Cf figure (4-2), soit l' effectif de 120 poids en tout .
C' est l' architecture adoptée pour la suite du travail .

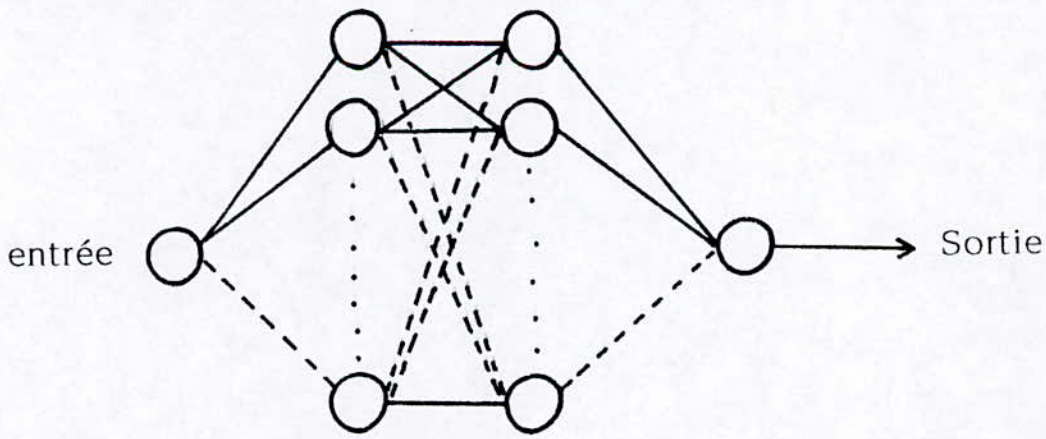


Fig 4-2 : Architecture du RNA adoptée pour la suite du travail.

4-3 Simulation de la non linéarité du type dûr :

Dans ce cas, la non linéarité choisie est de type dûr, c' est à dire qu' elle ne présente pas de saturation .

La fonction qui caractérise le plus ce type de non linéarité, est la fonction :

$$y = x^2$$

Le RNA doit simuler cette fonction, aprés apprentissage effectué par les deux algorithmes : BP et ROM .
La fonction d' activation choisie pour chacun des neurones est :

$$F(x) = \tanh(x)$$

Saturable à -1 et +1, une mise à l' echelle des grandeurs d' entrée et de sortie s' impose [1], on prend alors :

l' entrée x du réseau variant entre 0 et 1, la sortie y est alors bornée dans l' intervalle [0, 1]. Pour ces conditions de travail, on subdivise l' intervalle [0, 1] en 100 points avec un pas constant de 0.01, soit alors 100 exemples à apprendre. L' initialisation des poids est de 0.3 pour les deux algorithmes d' apprentissage.

4.3.1 Apprentissage par BP:

Le travail consiste à faire varier le pas du gradient μ, et de relever le nombre d' itérations et l' erreur finale exprimée comme suit :

$$\xi = 1/m \sum_{i=1}^{m} e_i^2$$

qui est la moyenne de la somme des carrés des erreurs ei commises sur chaque exemple à apprendre.

m étant le nombre des exemples (m = 100).
Les résultats obtenus sont contenus dans le tableau (4-3) :

pas du gradient	nombre d' itérations	valeur finale de $\mathcal{E}$
0.005	10650	0.00513
0.01	10000	0.001
0.1	360	0.0016
0.25	—	—
0.5	—	—

Tab 4-3 : Simulation de $y = x^2$, apprentissage par BP.

Remarques :

- Pour le cas où u = 0.5 et u = 0. 25 le pas est relativement grand on constate que l' erreur converge jusqu' à la 17ième itération ensuite commence à osciller entre deux valeurs fixes non satisfaisantes.

- Pour les autres valeurs de u , la convergence de l' erreur est normale et on a des erreurs ei sur chaque exemple appris, variant entre 10^{-3} et 10^{-4}

- La généralisation de la simulation de la fonction en question, à des points non appris est visualisée dans la figure (4-3) . On remarque que si l' entrée du RNA est superieure à 0.7 , l' erreur entre la sortie du réseau et celle désirée de la fonction fait aspect de divergence. Cela peut être expliqué par le fait que la pente augmente au fûr et à mesure qu' on s' éloigne de l' origine.

Dans ces conditions, il était nécessaire d' augmenter le nombre de points d' entrainement dans la zone] 0.7, 1 [, une discritisation plus fine a été effectuée sur cet intervalle, une légère amélioration de la généralisation a été constatée.

4.3.2 Apprentissage par ROM :

Au début on se met dans les mêmes conditions que précédemment à savoir :

l' initialisation des poids à 0.3 , le nombre de points d' apprentissage est de 100, la fonction d' activation de chaque neurone est :

$$F(x) = \tanh(x)$$

pour plusieurs valeurs de la variance v , on obtient les résultats suivants, Cf tableau (4-4)

valeur de la variance	nombre d' itérations	valeur final de ε
0.001	100	0.0227
0.01	100	0.0219
0.1	100	0.0226
1	100	0.0635
2	100	0.1894

Tab 4-4 : Simulation de $y = x^2$, apprentissage par ROM . (100 points)

en augmentant le nombre de points à apprendre à 200 points , on obtient les résultats suivants cf tableau (4-5)

valeur de la variance	nombre d' itérations	valeur final de ε
0.0005	100	0.005
0.001	100	0.005
0.01	100	0.006
0.1	100	0.005

Tab 4-5 : Simulation de $y = x^2$, apprentissage par ROM (200 points) .

Remarques :

La première constatation à signaler, est le caractère en escalier avec lequel la méthode converge. En effet l' erreur quadratique moyenne peut stagner un nombre d' itérations considérable à une valeur fixe avant de descendre à un niveau plus bas. A titre d' exemple, dans le cas de v = 0.1 , l' erreur stagne à partir de la septième itération, seulement cette attitude diminue chaque fois qu' on diminue la variance, et ce caractère de convergence en escalier est inexistant à partir des valeurs de v inférieures à 0.0005, alors qu' il est paralisant si la variance v est superieure à 0.1 Cf tableau (4-4) .

- La généralisation dans le premier cas n' était pas satisfaisante, cela peut avoir plusieurs causes notamment l' initialisation des poids qui peut être inadéquate, ou tout simplement dûe au nombre de point insuffisant, du moment que l' erreur finale est bien inferieure dans le cas où on a pris 200 points Cf tableau (4-5) . Pour ce dernier cas une généralisation à des points non appris est illustrée dans la figure (4-4)

4.4 Simulation de la non linéarité de type mou :

La non linéarité dans ce cas est saturable, une fonction qui caractérise ce type là est la fonction :

$$y = 1 - \exp(-x)$$

Les exemples d' entrainement sont choisis dans l' intervalle]0, 5] subdivisé en 100 parties, soit alors 100 points à apprendre .

La fonction d' activation reste inchangée, les poids sont initialisés à 0.3.

4.4.1 Apprentissage par BP :

En faisant varier le pas du gradient μ entre 0.005 et 0.5 , on obtient les résultats mentionnés dans le tableau (4-6) .

pas du geadient	nombre d' itérations	valeur final de ε
0.005	3000	$9 . 10^{-6}$
0.01	2000	$2.18 . 10^{-5}$
0.25	135	$9 . 10^{-6}$
0.5	2	$9 . 10^{-6}$

Tab 4-6 : Simulation de la fonction $y = 1 - \exp(-x)$, apprentissage par BP.

Remarques :

- Bien que le nombre d' itérations est relativement élevé quand le pas u est petit Cf tableau (4-6), la qualité de l' erreur est trés satisfaisante par rapport à celle obtenue avec la fonction
$y = x^2$

- La remarque la plus importante à ce stade, c' est la capacité du réseau à

généraliser à des points non appris même à l' exterieur de l' intervalle d' entrainement et cela est d' une importance considérable.
Cf figure (4-5)

4.4.2 Apprentissage par ROM :

Dans les mêmes conditions initiales précédentes, et pour diverses valeurs de la variance v, on obtient les résultats du tableau (4-7)

valeur de la variance	nombre d' itérations	valeur finale de ε
0.001	100	$1.4 . 10^{-4}$
0.01	100	$9.8 . 10^{-4}$
0.1	100	$5.8 . 10^{-5}$
1	—	—
2	—	—

Tab 4-7 : Simulation de la fonction y = 1 - exp (-x) , apprentissage par ROM .

Remarques :

- Le problème se répercute sur ce cas, une convergence en escalier est constaté pour les valeurs relativement élevées de la variance v, et la convergence ne s' opère que quand la séquence de bruit adequate soit arrivée.

- Par contre la qualité de l' erreur est satisfaisante, et on a pour avantage

par rapport à BP la réduction du nombre d' itérations.

A l' instar de BP la génération est jugée trés acceptable même en dehors de l' intervalle d' apprentissage voir figure (4-6)

4-5 Modification dynamique de la variance :

Dans le but de se débarasser de la convergence en escalier ou du moins diminuer la durée de la stagnation de l' erreur, qui est fréquente dans notre application, il était interressant de tester une variance dynamique tout au long de l' apprentissage.

Celle-ci doit varier entre 0.1 et 0.01 en décroissance, soit par exemple:

$$v = 0.1 / k$$

k : nombre d' itérations

En appliquant cette formule à l' apprentissage de la fonction $y = x^2$, on obtient pour le même nombre d' itérations et le même nombre d' exemples d' entrainement une erreur meilleure Cf tableau (4-8)

nombre de points	nombre d' itérations	erreur finale ε
100	100	0.0045

Tab 4-8 : Apprentissage par ROM avec une variance variable.

4.6 Autre méthode pour l' initialisation des poids du RNA :

Généralement, les poids du réseau sont initialisés aléatoirement à des valeurs petites variant entre -0.5 et 0.5 , pour éviter au réseau de se saturer trop rapidement [14] , c' est ce qu' on a faitjusqu' à présent . Toute fois si on arrive à trouver une initialisation meilleure cela ne peut qu' améliorer notre rapidité de convergence, ainsi

que la qualité de l' erreur de sortie .

Partant de ce point de vue, une idée pratique a surgi pendant qu' on effectuait le travail précédent .

Cette idée consiste à initialiser les poids par l' une des deux méthodes et à continuer l' apprentissage par l' algorithme BP par exemple , jusqu' à un certain nombre d' itération pas trés grand, les poids aurant à ce moment des valeurs corrigées meilleures que celles prises aléatoirement, et c' est ces poids la qui feront l' objet de poids initiaux pour l' algorithme ROM . Nous avouons à notre surprise, que la qualité de l' erreur finale était meilleure que celle obtenue au paravant Cf tableau (4-9) .

nombre de points	nombre d' itérations	erreur finale ε
100	100	0.0024
200	100	$1.09 . 10^{-4}$

Tab 4-9 : Initialisation des poids par BP et apprentissage par ROM .

L' initialisation des poids été faite par BP et a duré 20 itérations.

La généralisation quand à elle, c' est améliorée aussi, voir figure (4-7).

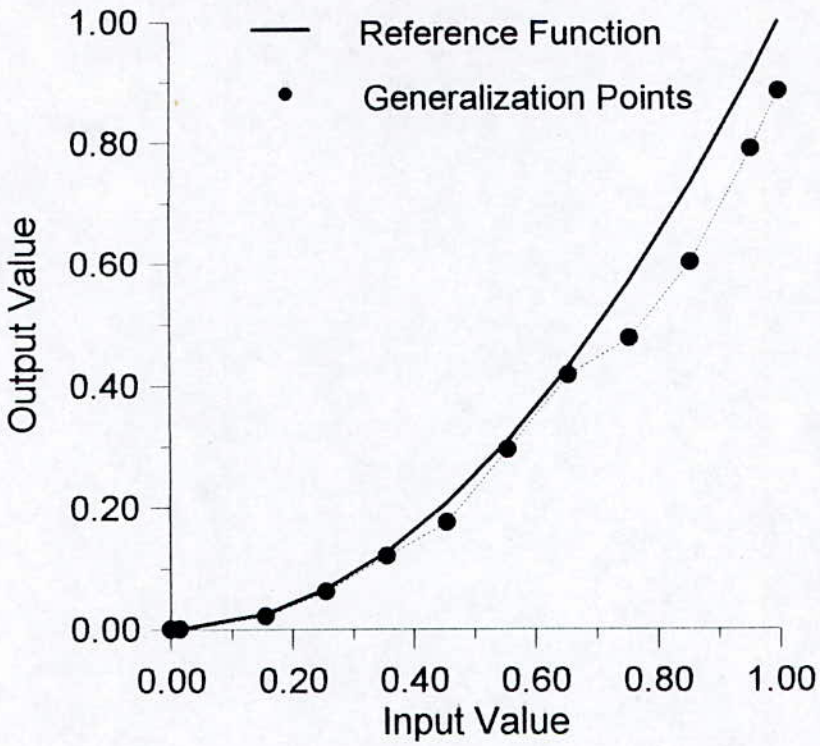


Fig.4.3 Illustration de la courbe de référence et la courbe de généralisation pour PB (fonction dure)

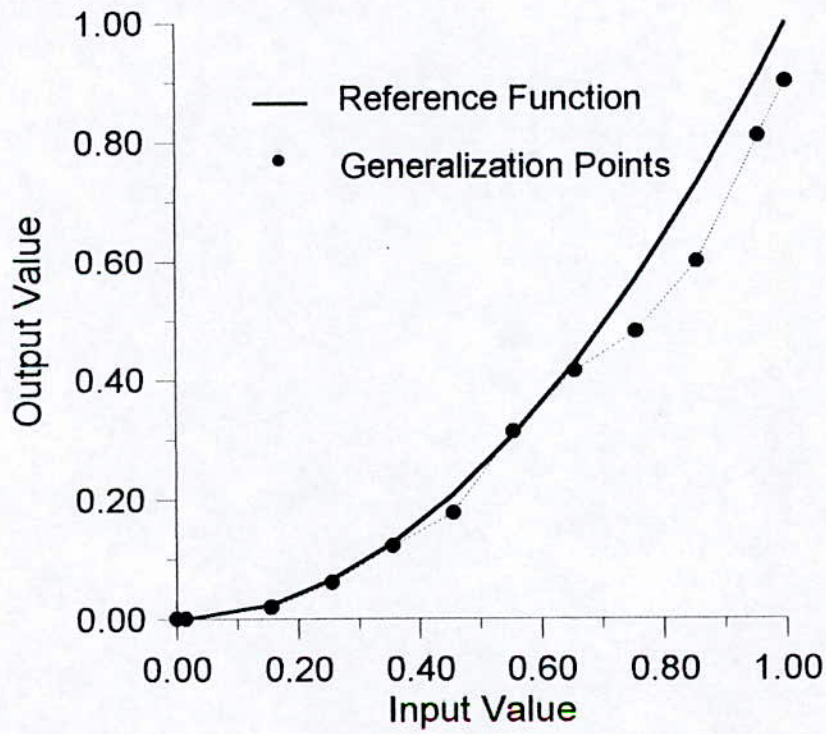


Fig.4.4 Illustration de la courbe de référence et la courbe de généralisation pour ROM (fonction dure)

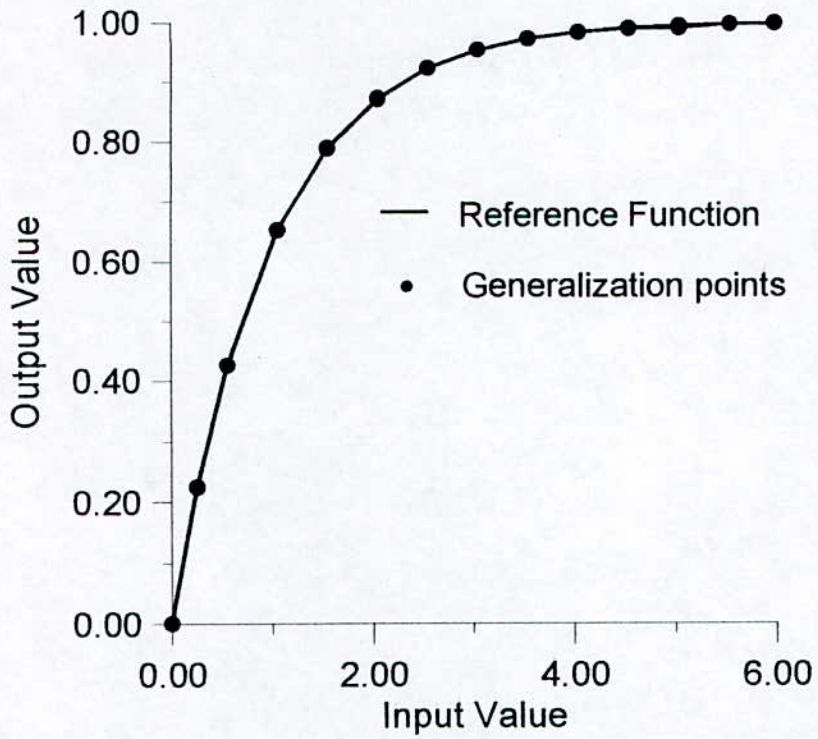


Fig.4.5 Illustration de la courbe de référence et la courbe de généralisation pour PB (fonction douce)

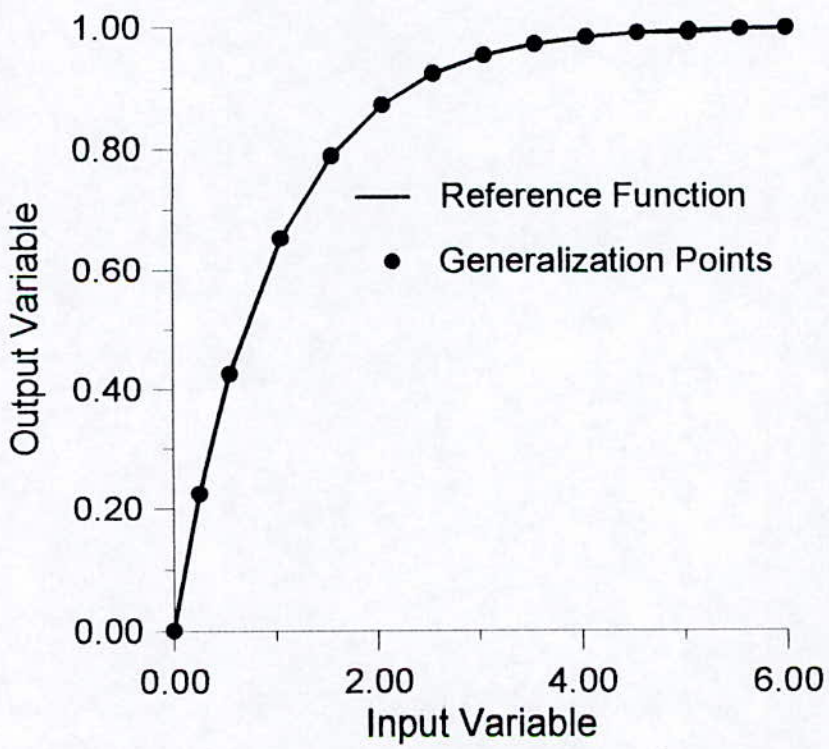


Fig.4.6 Illustration de la courbe de référence et la courbe de généralisation pour ROM (fonction douce)

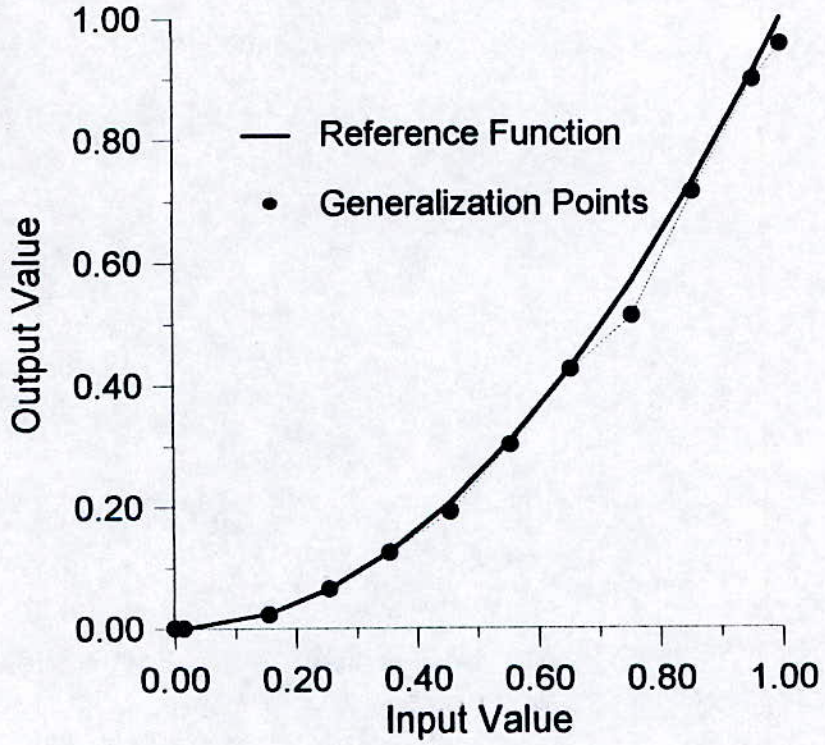


Fig.4.7 Illustration de la courbe de référence et la courbe de généralisation pour PB et ROM (fonction dure)

4.7 Conclusions :

Cette partie de notre travail qui consistait à simuler des fonctions non linéaires ne peut être que bénéfique pour la suite de notre étude, car elle nous a permi de sortir avec des conclusions qui sont les suivantes :

- Le comportement du RNA diffèrt d' un type de non linéarité à un autre . Il arrive à simuler la non linéarité de type mou avec une bonne capacité de généralisation car selon nous ce type ressemble à la non linéarité de la fonction d' activation du neurone qui est elle même saturable. Ce bon comportement commence à se perdre dés qu' on a affaire à des non linéarités de type dûr , vu leur caractère divergent, cela nécessite une discrétisation plus fine du domaine d' apprentissage, voire une discrétisation avec un pas variable.

- L' utilisation de BP comme algorithme d' apprentissage, nous a laissé penser à un choix meilleur du pas du gradient, c' est à dire à le borner entre 0.1 et 0.005 pour le cas des non linéarités de type dûr et entre 0.1 et 0.5 pour ceux de type mou .

- Pour ce qui concerne la méthode ROM, celle-ci peut nous éviter de longues périodes d' apprentissage rencontrées avec BP, seulement un problème de stagnation d' erreur est rencontré, pour cela il faut prévoir une variation de la variance durant l' apprentissage.

- Dans la dernière partie du travail, nous avons combiné les deux techniques d' apprentissage à savoir BP dans un premier temps pour initialiser les poids, et ROM en second lieu pour la convergence vers un minimum global . Nous avons obtenu avec cette combinaison le meilleur résultat, car cette dernière profite des avantages des deux algorithmes; à savoir une convergence de l' erreur quadratique sans stagnation pour BP et une rapidité de convergence relative pour ROM .

CHAPITRE 5

APPLICATION DES RNA POUR LA LINEARISATION DES CARACTERISTIQUES STATISTIQUES NON LINEAIRES DE CAPTEURS

5.1 Introduction :

Dans le troisième chapitre, nous avons présenté deux méthodes utilisées jusqu' à présent pour la linéarisation des caractéristiques non linéaires de capteurs. Durant cette présentation succincte, on a fait paraître quelques avantages et inconvénients qui pouvaient surgir lors de l' utilisation de ces méthodes.

Des performances ont été obtenues, par l' utilisation des RNA et cela dans divers domaines de recherche et notamment en instrumentation [21],[22] ainsi que leur caractère non seulement adaptatif, mais aussi non linéaire. En effet, une idée originale valait la peine d' être testée qui consiste à linéariser les caractéristiques des capteurs par les RNA.

L' objectif de ce travail, est d' évaluer les capacités de cette nouvelle méthode et d' essayer d' obtenir des performances supérieures, sinon égales à celles des méthodes usuelles.

Ce chapitre comprend 4 parties essentielles . Chacune traite de la linéarisation d' un type decapteur.

- Dans la première partie, on effectue la linéarisation de deux thermistances étalonnées pratiquement, avec des discrétisations différentes.

- Dans la deuxième partie, on linéarise la caractéristique d' un anémomètre qui est étalonné à son tour pratiquement .

- La troisième partie, est réservée à la linéarisation de la caractéristique d' un thermocouple, celui-ci est pris en considération vu sa large étendue de mesure, ainsi que sa grande utilité dans le domaine industriel .

- La quatrième partie quant à elle, s' occupe de la linéarisation d' un capteur chimique particulier appelé éléctrode séléctive d' ions, en prenant en compte la grandeur de perturbation qui influe sur le capteur en présence d' ions interférents dans la solution .

A l' entrée de chaque partie, on présentera le capteur en question, ainsi que son principe de fonctionnement .

5.2 Principe de fonctionnement du RNA :

Comme dans le cas de la linéarisation analogique, le RNA doit simuler la caractéristique inverse du capteur qui est non linéaire, l' assemblage en cascade du capteur et du RNA aprés apprentissage,se comportera comme un bloc linéaire Cf figure (5-1) . Le signal numérique à la sortie de l' ADC est linéarisé par le RNA implanté dans une EPROM et le µp affiche un signal proportionnel au mesurande x

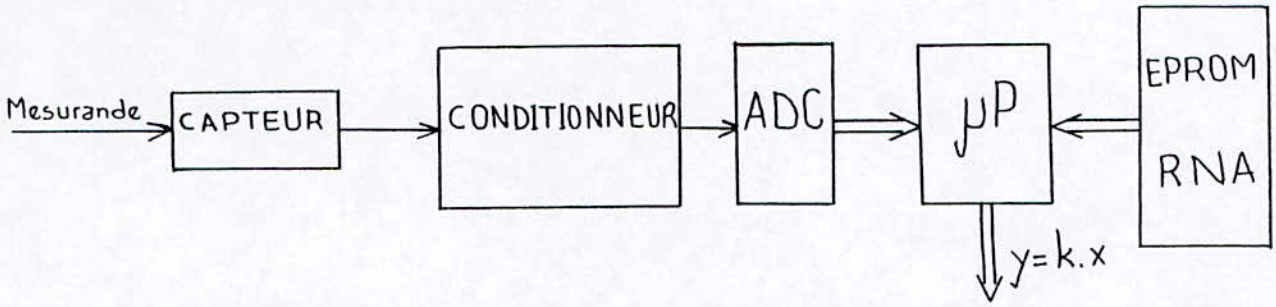


Fig 5-1 : Insertion d' un RNA dans une chaine instrumentale

5.3 Apprentissage du RNA :

La fonction du réseau étant de simuler la caractéristique statique d' un capteur, il doit alors avoir pour exemples d' entrainement des couples (entrées , sorties) pratiques issues de la caractéristique obtenue par lissage lors de l' étalonnage du capteur . Ce choix est validé par le fait que cette caractéristique représente le plus le capteur car c' est bien évident que les mesures obtenues sont bruités.

Les données sont alors calculées à partir du polynome d' étalonnage.

- La phase d' apprentissage du réseau est résumée dans la figure(5-2) :

e : la mesure

x : le mesurande

wij : poids du réseau

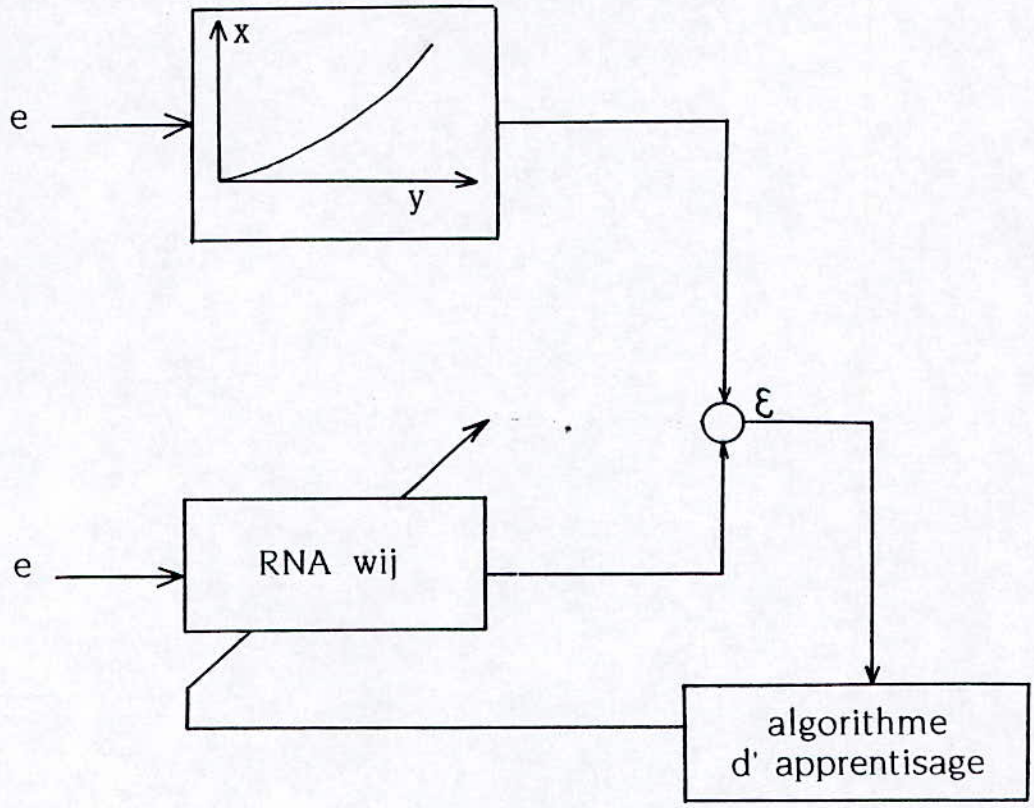


Fig 5-2: Apprentissage du RNA, sans grandeur de perturbation

Dans le cas, où la grandeur de perturbation est prise en compte, celle-ci doit être captée et mesurée à l' aide d' un capteur, elle doit donc être mesurable. La phase d' apprentissage est résumée dans la figure (5-3) .

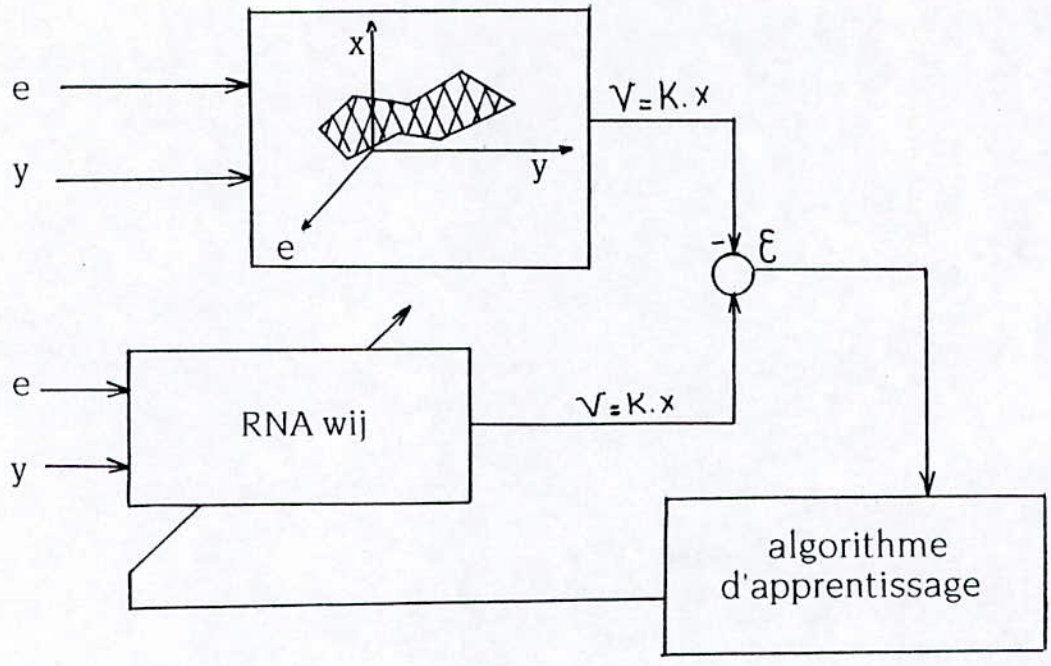


Fig 5-3 Apprentissage du RNA, avec grandeur de perturbation .

Les exemples d' entrainement, sont prise du polynome d' étalonnage :

$$v = f (e, y)$$

y : étant la grandeur de perturbation

5.4 Généralisation :

Il s' agit d' associer en cascade le capteur en question avec le RNA préalablement entrainé pour simuler la caractéristique inverse de ce capteur. Ce dernier doit donner des mesures non appris pour le RNA. Cf figure (5-4) et (5-5) . La sortie du RNA nous permettera de juger de la qualité de la généralisation , étant donné que cette sortie doit être l' image du mesurande. Pour cela, on calculera les erreurs relatives sur chaque exemple non appris .Ces erreurs doivent être inferieures ou égales à 1%, un seuil généralement acceptable pour les appareils de mesure de classe 1.

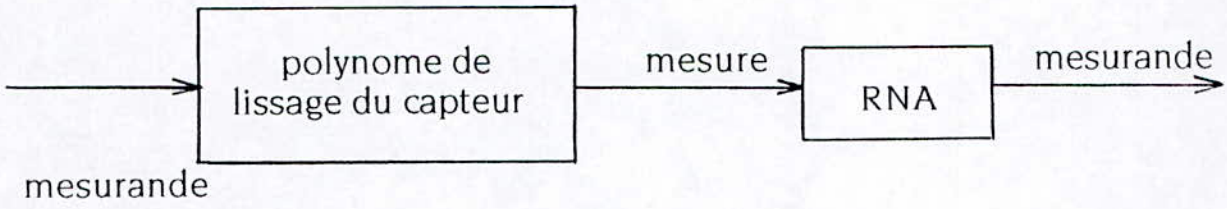


Fig 5-4 : Généralisation sans perturbation

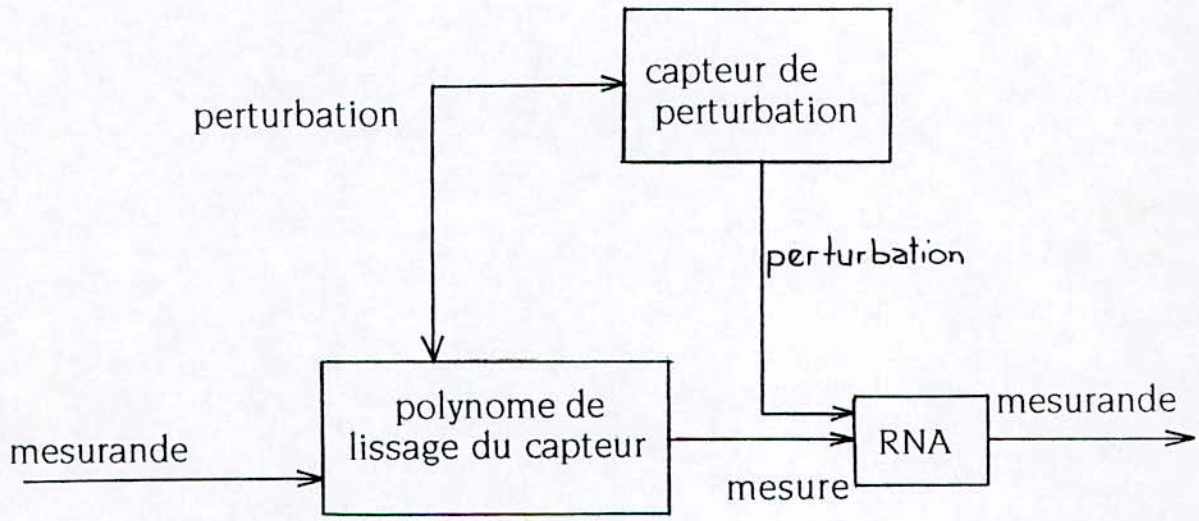


Fig 5-5 : Généralisation avec perturbation .

5.5 Linéarisation de capteurs à l' aide des RNA sans perturbation

5.5.1 Linéarisation de la caractéristique d' une thermistance

5.5.1.1 Présentation de la thermistance :

- La mesure de la température en industrie est très fréquente, car elle permet de toute évidence de déterminer les propriétés de la matière. De ceci découle un contrôle strict de cette grandeur.

Il existe pour cela plusieurs méthodes physiques de mesure de la température telles que [17] .

- Les méthodes optiques découlant de l' élargissement des rais spéctrales par l' effet Doppler, dû à l' agitation thermique.

- Les méthodes mécaniques, basées sur la dilatation d' un solide d' un liquide ou d' un gaz, ou de la vitesse du son par exemple.

- Les méthodes éléctriques, fondées sur la variation de la valeur d' une résistance ou de son bruit de fond, ou sur l'effet seebeck etc.

La thermométrie par résistance, s' effectue par l' utilisation de deux types de résistances :

- Les résistances métalliques dont la loi de variation en fonction de température est [17] :

$$R(T) = R_0 (1 + AT + AT^2) \qquad (5\text{-}1)$$

- Les thermistances, dont la relation résistance - température s' exprime par [17] :

$$R(T) = R_0 \exp (B (1 / T - 1 / T_0)) \qquad (5\text{-}2)$$

L' avantage principal des thermistances par rapport aux résistances métalliques, est leur sensibilité thermique superieure de l' ordre de 10 fois de celle des résistances métalliques .

Elle sont fabriquées à base de mélanges d' oxydes métalliques semi - conducteurs polycristallin tels que [17] :

Mgo , Fe3o4 , Nio etc ...

Leur sensibilité trop élevée leur permet d' acquérir des résistances de valeurs appropriées avec de faibles quantités de matière, et alors des dimensions réduites. Leur stabilité dépend de leur réalisation et des conditions d' emploi. Généralement pour effectuer des mesures de température faible, on utilise des thermistances de faible résistance de l' ordre de 50 à 100 Ω à 25°C, alors qu' on utilise des thermistances ayant des résistances de l' ordre de 100 à 500 Ωk pour mesurer les températures élevées. (max 150°C).

5.5.1.2 Circuit conditionneur de la thermistance .

Le montage de conditionnement utilisé est le suivant : Cf . figure (5-6)

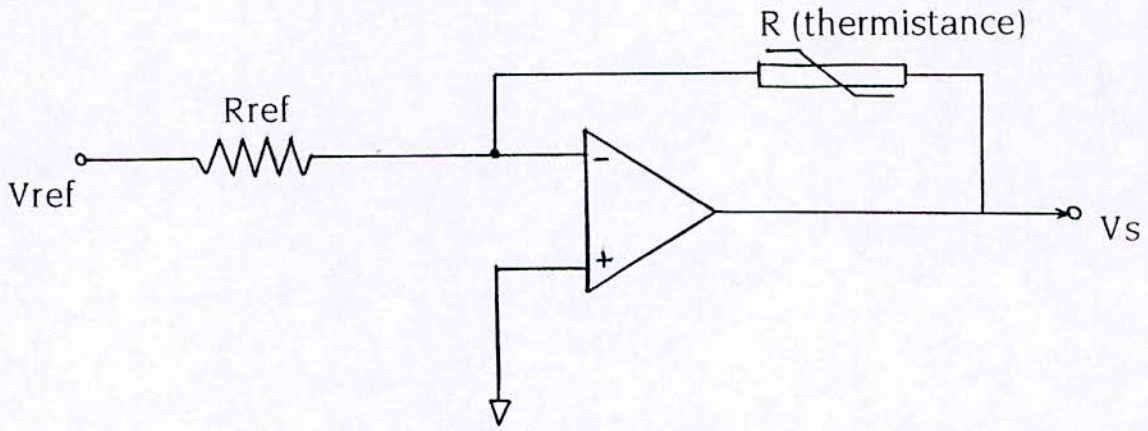


Fig 5-6 : Circuit de conditionnement d' une thermistance .

La loi régissant la valeur de la thermistance R et la temperature est de la forme :

$$R = K \exp (B / T) \qquad (5\text{-}3)$$

La tension receuillie à la sortie de l' amplificateur qui représente la mesure de la température, s' exprime par :

$$Vs = -(R / R_{ref}) V_{ref} = - (K / R_{ref}) \exp (\beta / T) V_{ref} \qquad (5\text{-}4)$$

- Pour simuler la caractéristique inverse :

$$T = f (Vs)$$

le RNA doit avoir comme entrée la température T, et comme sortie la tension Vs.

Dans notre cas on choisie : $V_{ref} = (5.5 \pm 10^{-4}) V$

$R_{ref} = 45\ \Omega \quad (1\%)$

La thermistance R avec laquelle on travaille, est une thermistance réelle [28], étalonnée dans un laboratoire lors d' un projet de fin d' études, et représentant la température qui est la sortie désirée du réseau, ainsi que la tension d' entrée de celui-ci .

L' architecture du réseau choisie est de deux couches cachées à 10 neurones chacune, plus un neurone en couche de sortie .

Chaque neurone a pour fonction d' activation :

$F(x) = \tanh(x)$

A - Phase d' apprentissage :

Cette phase préliminaire démarre avec l' algorithme BP, celui-ci à pour rôle de diminuer l' erreur jusqu' à un certain seuil, les poids aux quels il arrive sont pris comme poids initiaux pour l' algorithme ROM . Celui-ci termine la tâche d' apprentissage en lui donnant un nombre d' itérations plus grand, dans le but de converger vers un minimum de la somme des erreurs quadratiques satisfaisantes. Les résultats sont les suivants : Cf . Tableau (5-2)

pas du gradient	nombre d' itérations	erreur finale ε
variable	5000	$3.34\ 10^{3}$

Tab 5-1 : Résultat obtenu aprés l' apprentissage par BP

variance	nombre d' itérations	erreur finale ε
variable	10000	$4.1\ 10^{4}$

Tab 5-2 : Résultat obtenu aprés apprentissage par ROM

Les erreurs relatives ΔT/T commises par le réseau sur chaque exemple reportées sur un graphe représentatif . Cf figure (5-7)

Dans cette phase d' apprentissage, on a fait en sorte que toutes les erreurs relatives commises par le RNA, ou une grande partie de celle-ci soient inférieures à 1%, seulement d' aprés l' allure de ces erreurs, on remarque une différence entre une erreur relative et une autre. Dans ce cas précis elle varie entre 2. 4% et 0. 03% la majorité des erreurs relatives par contre sont bornées entre 0% et 1% , des valeurs qui sont satisfaisantes.

Ce comportement reste pour nous inexplicable, et cela est dû au fait que la manière avec laquelle le réseau apprend la tâche a accomplir, et souvent difficile à détailler [2] , [10] .

B. Généralisation :

Les points choisis pour la généralisation se trouvent dans des intervalles limités par les points appris. pour évaluer la qualité de la généralisation,un graphe représentatif a été fait : voir figure (5-7), en superposition avec celui de l' apprentissage. Le résultat était trés satisfaisant vu que le RNA a donné des erreurs relatives nettement inférieures à celles commises lors de l' apprentissage, ces erreurs sont toutes en deça de 1% .

Un autre graphe de figure (5-8) , nous montre le positionnement des points de la généralisation sur la caractéristique inverse de la thermistance et la manière avec laquelle le RNA approche cette caractéristique. Les erreurs comises lors de la généralisation ne sont pas dûes à des erreurs de manipulations ou des erreurs accidentelles comme c' est le cas lors de la mesure avec le capteur, mais sont dûes tout simplement à un comportement intrinsèque au réseau reseau de neurones.

On note au passage que le phénomène de surapprentissage ne s' est pas manifesté lors de la phase de généralisation .

Le graphe de la figure (5-9) a pour rôle de faire apparaitre la qualité de la linéarisation reliant la température réelle qui est l' entrée de la thermistance à la température mesurée donnée à la sortie du RNA, cette température est étalée sur une étendue de 20°C à 90°C .

Dans le but d' améliorer la qualité de la généralisation ainsi que la précision de la mesure, on a pris une deuxième thermistance ayant une étendue de mesure de 0°C à 100°C et on a bien distribué les points d' apprentissage sur la caractéristique non linéaire .

A - Phase d' apprentissage :

Les résultats d' apprentissage obtenus par BP et ROM sont les suivants : Cf . Tableau (5-3) et tableau (5-4) .

pas du gradient	nombre d' itérations	erreur finale ε
variable	2000	$2\ 10^{-4}$

Tab 5-3 : Résultat d' apprentissage par BP .

variance	nombre d' itérations	erreur finale ε
variable	10000	$7.9\ 10^{-5}$

Tab 5-4 : Résultat d' apprentissage par ROM .

B . Généralisation :

De la figure (5-10), il est clair que les erreurs relatives, et de l' apprentissage et de la généralisation se sont améliorées et ont chûté à un seuil inferieur à 1% . Celles de la généralisation sont encore meilleures car il ne dépassent pas les 0.45% .

De plus les points de généralisation coïncident de plus en plus avec la caractéristique de la thermistance Cf figure (5-11) et on arrive plus à les distinguer des points de la caractéristique elle même .
La linéalisation a atteint sa précision désirée voir figure (5-12) est suit l' allure de la droite :

$$y = x$$

y : température à la sortie du réseau .

x : température réelle (entrée du capteur) .

5.5.1.3 Conclusion :

Tout ce qui a été mentionner dans cette première partie est l' influence de la manière de discritisation lors du choix des exemples d' apprentissage. Dans ce cas précis,la caractéristique étant symétrique par rapport à la droite y = x , il fallait donc prendre des points bien distribués de part et d' autre de la droite, le résultat de la deuxième thermistance était donc meilleur du point de vu apprentissage et généralisation .

5.5.2 Linéarisation de la caractéristique d' un anémomètre :

5.5.2.1 Présentation de l' anémomètre :

L' anémomètre, est un transducteur sert à mesurer la vitesse des fluides gazeux . Ce procédé de mesure de la vitesse d' un fluide est basé sur plusieurs phénomènes physiques, on notera comme exemple [17] :

* Précision dynamique exercée par le fluide .

* Effet Doppler subi par un rayonnement Laser ou Ultrasonne .

* Vitesse de rotation d' une hélice ou anémomètre à coupelles .

Dans notre cas on va s' occuper de l' anémomètre à fil ou à filme chaud . Son principe de fonctionnement est basé sur le placement d' un fil résistant, dont la température dûe à l' effet joule est supérieure à celle de l' écoulement , il se produit alors un échange de chaleur par convéction
Cet échange est fonction de l' écart de température entre le fil chauffé et le fluide .

La puissance Pc dissipée par le fluide est :

$$P_c = (a + b \sqrt{v}) (T - T_a) \qquad (5\text{-}5)$$

V : Vitesse du fluide

T : Température du fil

T_a : Température de l' écoulement .

La puissance joule Pj dissipée dans le fil est :

$$P_j = RI^2 \qquad (5\text{-}6)$$

à l' équilibre on obtient la loi d' echange appelée loi de king [17] :

$$RI^2 = (a + b\sqrt{v}) (T - Ta) \qquad (5\text{-}7)$$

Si I est fixe, la mesure de la résistance R nous permet de déterminer la vitesse qui est fonction non linéaire de R .

5.5.2.2 Circuit conditionneur de la résistance de l' anémomètre :

Ce circuit est constitué d' un pont de conditionnement au quel en joint la résistance, plus un étage supplémentaire qui sert à filtré la tension de sortie . voir figure (5-13) .

Le RNA choisi pour simuler la caractéristique inverse de l' anémomètre est un réseau à 2 couches cachées avec 10 neurones chacune; plus un neurone en couche de sortie.

La fonction d' activation de chaque neurone est :

$$F(x) = 1 / (1 + \exp (-2x))$$

cela est dû au fait que les entrées/sorties sont positives.
L' anémomètre sur lequel on effectue la linéarisation, est réalisé et étalonné pratiquement [28]

A . Phase d' apprentissage :

Comme prévu, cette phase se déroule en 2 parties : apprentissage par BP un certain nombre d' itérations, ensuite par ROM sur une période plus longue . Les résultats sont les suivants : Cf . tableau (5-5) , tableau (5-6) :

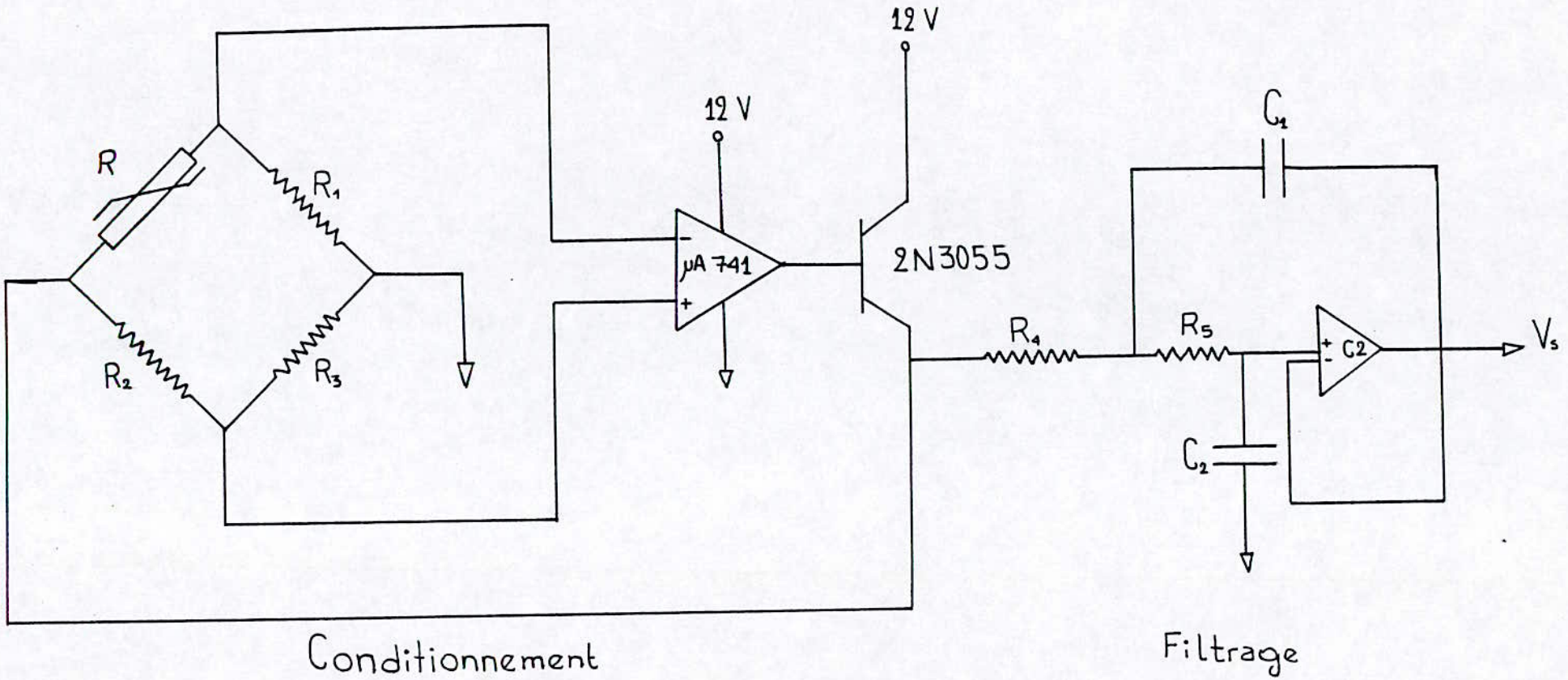


Fig. 5-13 : Circuit de conditionnement de la thermistance de l'anémomètre.

pas du gradient	nombre d' itérations	erreur finale
variable	2000	$6.26\ 10^{4}$

Tab 5-5 : Résultat de l' apprentissage par BP

variance	nombre d' itérations	erreur finale
variable	15000	$9.78\ 10^{6}$

Tab 5-6 : résultat de l' apprentissage par ROM .

Les erreurs individuelles sur chaque exemple d' apprentissage sont notées en annexe .

A l' issu de cet apprentissage,le RNA a réussi à apprendre les points se trouvant sur la partie supérieure du graphe grâce à la collaboration des deux algorithmes . Néanmoins, il a trouvé des difficultés à apprendre quelque points de la partie inférieure, les erreurs relatives pour ces points étant superieures à 1% . Cf . figure (5-14) .

B . Généralisation :

Malgré cette difficulté qu' on vient de citer en apprentissage, la généralisation du RNA était une surprise pour nous vu la qualité de l' erreur obtenue. Pour chaque point non appris , l' erreur relative ne dépassait pas les 0.35% . Cf figure (5-14) . Le réseau neuronal a même réussi à bienapproché la caractéristique inverse de l' anémomètre qui est de nature quadratique . Cf. figure (5-15) . De même, les points de la généralisation ont tendance de se rapprocher de la droite linéaire :

$$y = x$$

avec des petites courbures sur les deux extrémités qui sont dûes au type

dûr de la non linéarité de l' anémomètre . A part ces deux régions, la relation entre la vitesse à la sortie du réseau et la vitesse réelle à mesurer,peut être considérée comme linéaire . Cf . figure (5-16)

5.5.2.3 Conclusion :

La conclusion avec laquelle on sort aprés cette deuxième expérience de linéarisation est que, si la qualité de l' apprentissage est satisfaisante du point de vue erreurs relatives, la qualité de la généralisation est encore meilleure à savoir des erreurs relatives inferieures à celles de l' apprentissage, cette capacité extraordinaire de laquelle fait preuve le RNA ne laisse qu' à espérer à des résultats de plus en plus précis.

5.5.3 Linéarisation de la caractéristique d' un thermocouple :

5.5.3.1 Présentation du thermocouple :

Le thermocouple est constitué de deux conducteurs A et B formant deux jonctions aux températures T1 et T2, il délivre ainsi une F.e.m , qui est fonction des températures T1, T2 et de la nature de conducteurs A et B. L' une des température est fixée et considérée comme référence,l' autre est la température qu' on désire mesurer.

Le thermocouple permet de mesurer des températures ponctuelles, sa capacité calorifique réduite lui confère une vitesse de réponse élevée [17] . Son avantage est qu' il est un capteur actif, délivrant une tension à la sortie, dont la mesure ne nécessite pas un passage de courant dans le capteur, il n' y a donc aucune incertitude dûe à l' échauffement de celui-ci .
La non linéarité du thermocouple apparait de la forme algébrique qui lie la F.e.m du thermocouple à la température, obtenu d' une table de valeurs expérimentales , on aura alors l' expréssion polynomiale suivante :

$$E = \sum_{i=0}^{m} a_i T^i \qquad (5\text{-}8)$$

L' effet seebeck :

Un couple thermoéléctrique est un circuit fermé, constitué de deux conducteurs A et B dont les jonctions sont aux températures T1 et T2 . Cf. figure (5-17) . [17] :

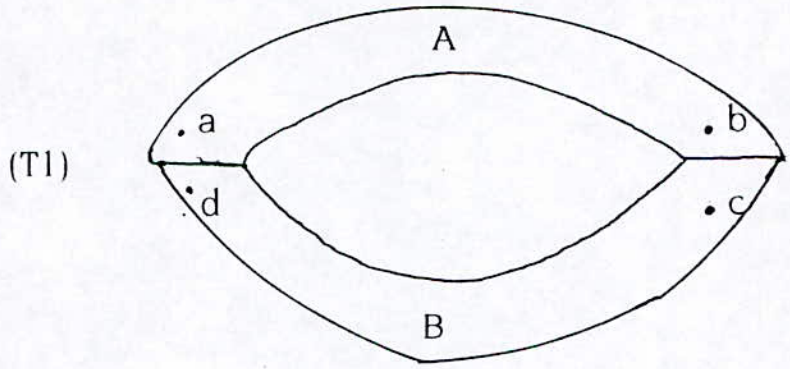


Fig 5.17 : schéma représentant un couple thermoéléctrique .

Ce couple délivre une tension dite de seebeck, calculée suivant la formule [17] :

$$E_{A/B} = P_{A/B}^{T2} - P_{A/B}^{T1} + \int_{T_1}^{T_2} (h_A - h_B) \, dT \qquad (5\text{-}9)$$

$P_{A/B}$: f.e.m qui résulte de l' effet Peltier

h_A , h_B : Coefficients de Thomson des conducteurs A et B, qui sont fonction de la température

Si, on fixe T1 à une valeur de référence, généralement à 0°C la f.e.m du couple en question, ne dépend plus que de T2.

- Pour garantir la stabilité de la f.e.m du thermocouple, la température maximale d' utilisation doit être fixée en tenant compte des conditions d' exploitation à savoir la température de fusion de l' un des conducteur et la nature de l' atmosphère environnante (oxydante, inerte, vide) , et cela pour éviter la dégradation rapide des caractéristiques .

Montage de mesure : [17]

Ce montage est résumé dans la figure suivante :

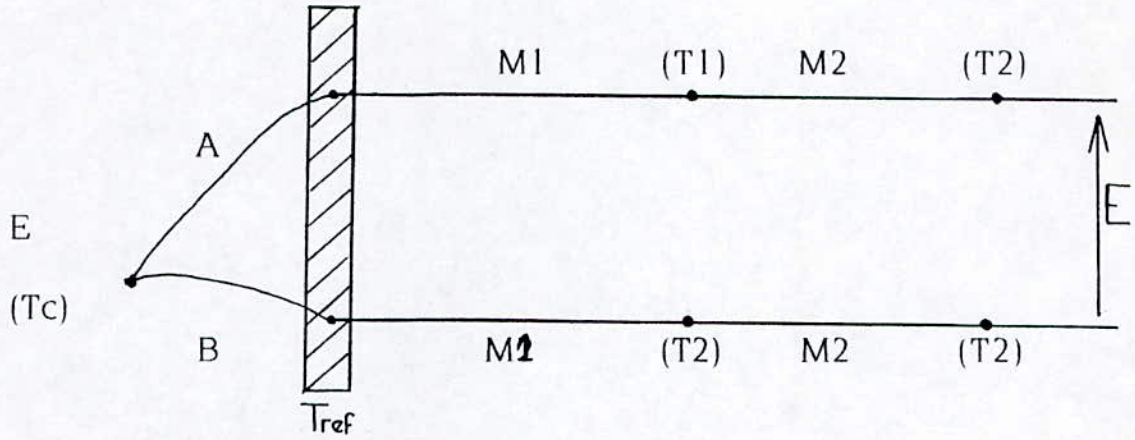


Fig 5-18 : Schéma de principe d' un thermocouple

Pour cette expérience de linéarisation , le choix a été fait sur un thermocouple industriel [27]
qui est du type Tungesten vs Tungesten 26% Rhénium . Les exemples d' apprentissage sont pris sur un intervalle de mesure qui s'étend de 90°C à 2000°C.
L' architècture du réseau est de deux couches cachées à 10 neurones chacune plus un neurone en couche de sortie .

La fonction d' activation de chaque neurone est :

$$y = \tanh(2x)$$

A. Phase d' apprentissage

L' initialisation est faite comme prévu par BP,tandis que l' apprentissage est mené jusqu' au bout par ROM .Les résultats réunis dans les tableaux (5-8),(5-9) :

pas du gradient	nombre d' itérations	erreur finale ε
variable	20000	$1.4\ 10^{-3}$

Tab 5-8 : apprentissage par BP

variance	nombre d' itérations	erreur finale $\mathcal{E}$
variable	100000	$4.12\ 10^{-5}$

Tab 5-9 : apprentissage par ROM

A l' issu de cette phase, les erreurs relatives individuelles sont en majorité inférieures à 2% cf. figure (5-17), et suivent un régime décroissant en moyènne sur toute l' étendue de mesure ,elles sont meme satisfaisantes pour les points audela de 1200°C.
Pour ce qui concèrne l' erreur quadratique , on signalera que celle ci n' a pu être diminuée moins de $4.12\ 10^{-5}$ et semble indiquée l' arrivée à un minimum global ,pour cela on a voulu améliorer l' erreur en utilisant un réseau à 3 couches cachées , mais sans résultat meilleur.

B. Généralisation

les erreurs relatives commises sur chaque exemple de généralisation suivent la même allure que ceux de l' apprentissage cf figure (5-17) . Ces erreurs sont relativement satisfaisantes pour les températures supérieures à 1000°C et moins précises pour celles inférieures à 1000°C .
La qualité de la linéarisation est illustrée dans les figures (5-19),(5-18).

5.5.3.2 conclusion

On signalera dans cette partie, qu' un minimum global peut ne pas être satisfaisant, et qu' un passage à une autre structure du réseau s' avère indispensable, mais sans qu' on puisse évaluer le résultat au préalable, ça sera donc un pur essai.
Toute fois, si les erreurs relatives nous paraissent acceptables, on peut s' arrêter à ce stade; c'est notre cas.Nous ajouterons en fin, et relativement aux autres capteurs qu' on a vu, que le thermocouple possède une étendue de mesure large, et le RNA en conséquence, prend beaucoup de temps d' apprentissage sans arriver à un minimum satisfaisant de l' erreur et une discrétisation encore plus fine des exemples à apprendre peut améliorer le résultat final.

5.6 Linéarisation de capteur à l' aide des RNA avec perturbation

5.6.1 Linéarisation de la caractéristique d' une éléctrode séléctive d' ions :

Le terme éléctrode séléctive d' ions est donné à un type de capteurs chimiques formés de membranes, qui répondent séléctivement à une catégorie d' ions spécifiques en présence d' autres ions . [23] [26]

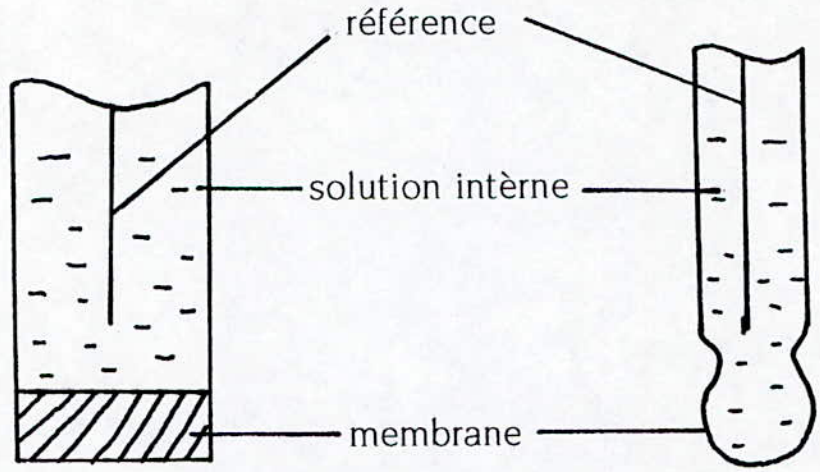
La membrane n' est autre qu' une section formée de matériau conducteur ioniques séparant deux solutions entre les quelles un potentiel se développe .

Ces éléctrodes sont généralement classées suivant le type de matériau avec lequel ils sont fabriqués, soit alors trois sortes d' éléctrodes :

* éléctrodes à base de sels inorganiques insolubles .
* éléctrodes à base de verre et céramique.
* éléctrodes à base de longues chaines d' échange d' ions

Ces matériaux ont pour propriètés communes, leur conductivité éléctrique soit par mouvement d' éléctrons, d' ions ou des deux en même temps [25] , ainsi que leur aptitude à établir un équilibre chimique d' echange d' ions .

La forme physique de l' éléctrode lors de sa construction varie suivant le type de matériau utilisé, la forme en bulbe est la plus souvant utilisée pour les éléctrodes fabriquées à base de verre . Cf . figure (5-22).



(a) membrane solide (b) membrane de verre

Fig . 5.22 : Différentes formes physiques éléctrodes .

Equation de Nernst :

Une éléctrode séléctive d' ion délivre à sa sortie une f.e.m en réponse à une concentration donnée d' un type d' ion spécifique, cette f.e.m est référée à une certaine tension de référence associée à une solution intèrne au capteur . La relation établié par Nernst est la suivante :

$$E = E_0 + S \text{ Log } A \quad (5\text{-}10)$$

E_0 : est le potentiel standard

S : Pente de la réponse de l' éléctrode, elle est généralement déterminée par étalonnage du capteur avec une solution comportant différentes concentrations du même type d' ions .

Théoriquement elle est exprimée par :

$$S = 2.3\ RT / Z\ F \quad (5\text{-}11)$$

R: constante de Boltzmann R = 8.314 J/mol.K

T : température absolue T = 298 K (ambiante)

Z : nombre de change de l' ion à mesurer .

F:constante de Faraday F = 96500 C

Equation de Nicolsky - Eisenmann :

Il n' existe point d' éléctrode séléctive d' ions qui répond exclusivement à l' ion pour lequel elle a été conçue, la présence d' autres ions peut affécter sérieusement les performances de celle-ci .

Le pouvoir de ce capteur à distinguer entre l' ion primaire A et l' ion interférent B dans la même solution est défini par son coefficient de séléctivité potentiométrique K_{AB} . La sortie du capteur suit l' équation de Nicolsky - Eisenmann :

$$E = E_0 + S \, Log (|A| + K_{A,B} |B|^{ZA/ZB}) \qquad (5\text{-}12)$$

Plusieurs méthodes expérimentales ont été établies [23] , [24] ,[26] pour mesurer le coefficient de séléctivité potentiométrique . Dans ce cas l' ion interférent est une grandeur de perturbation pour ce capteur .

Pour ce cas, on éffectue la linéarisation avec la présence d' un ion interférent comme perturbation, soit alors la simulation de la fonction inverse de l' équation de Nicolsky Eisenmann.
Le capteur choisi est à son tour industriel, réaliser par le constructeur Beckman [25]. Il a pour cefficient potentiométrique K = 0.1, il mesure la concentration de l' ion principal qu' est le fluore $[F^-]$ en présence de l' ion OH^-.

Vu la large étendue dans la quelle se trouvent les concentrations des deux types d' ions, et pour ne pas tomber dans le meme problème que le thermocouple, on a pris des exemples d' entrainement entre :

10^{-5} à 10^{-4} pour l' ion interfèrent OH^-

et 10^{-5} à 10^{-3} pour l' ion principal F^-

De plus, on considère qu'on se trouve à une température ambiante.
T = 298°C

La structure du RNA cette fois ci et de deux couches cachées à 8 neurones chacune, plus un neurone en couche de sortie. Ce RNA a bien sûr deux entrées :
La tension issu du capteur;

La concentration de l' ion interférent donnée par un autre capteur.

La fonction d' activation de chaque neurone est :

$$F(x) = 1 / (1 + \exp(-1.5x))$$

car les entrées /sorties du capteur sont superieures à 0.

A. Phase d' apprentissage

Les résultats obtenus par BP et par ROM sont les suivants:

pas du gradient	nombre d' itérations	erreur finale
variable	1000	$5.7\ 10^{-4}$

Tab 5-9 : apprentissage par BP

variance	nombre d' itérations	erreur finale
variable	$2\ 10^{5}$	$2.09\ 10^{-10}$

Tab 5-10 : apprentissage par ROM

On remarque que l' erreur finale ne peut déscendre moins que 2 10^{-10} est reste stagnante pendant plusieurs itérations. Les erreurs relatives quant à eux sont divisés en 3 groupes :

*Pour le cas ou $|OH^-| = 10^{-5}$

Les erreurs individuelles ont une précision accéptables pour une variation de |F-| entre 10^{-4} et 10-3 et moins précises pour une variation de |F-| entre 10^{-5} et 10^{-4}.

*Pour les cas ou $|OH^-| = 5\ 10^{-5}$ et $|OH-|=10^{-4}$

le meme phénomène se reproduit, mais avec des erreurs relatives inférieures à 20% pour

$|OH^-| = 5\ 10^{-5}$ cf figure (5-20);

B.Généralisation

La qualité de la généralisation a diminué relativement aux autres capteurs, car ce cas est plus délicat.
On distingue de la figure (5-21) que les erreurs varient entre 6 10^{-8} et 3 10^{-5} pour des concentrations de F^- allant entre 10^{-5} et 10^{-3}.
La difficulté peut être expliquée par le RNA doit réster indifférent face aux variations de la concentration de l' ion perturbateur.

5.6.2 conclusion

Un compromis doit etre fait pour résoudre ce type de problème entre la structure du réseau et le temps d' apprentissage qu' on veut avoir, car dans ce cas une augmentation du nombre de couches et du nombre d' exemples d' entrainement peut étre sur de l' amélioration du résultat final.
Nous ajoutons à cela qu' une linéarisation sur une très large étendue de mesure peut diminuer la qualité de la linéarisation.

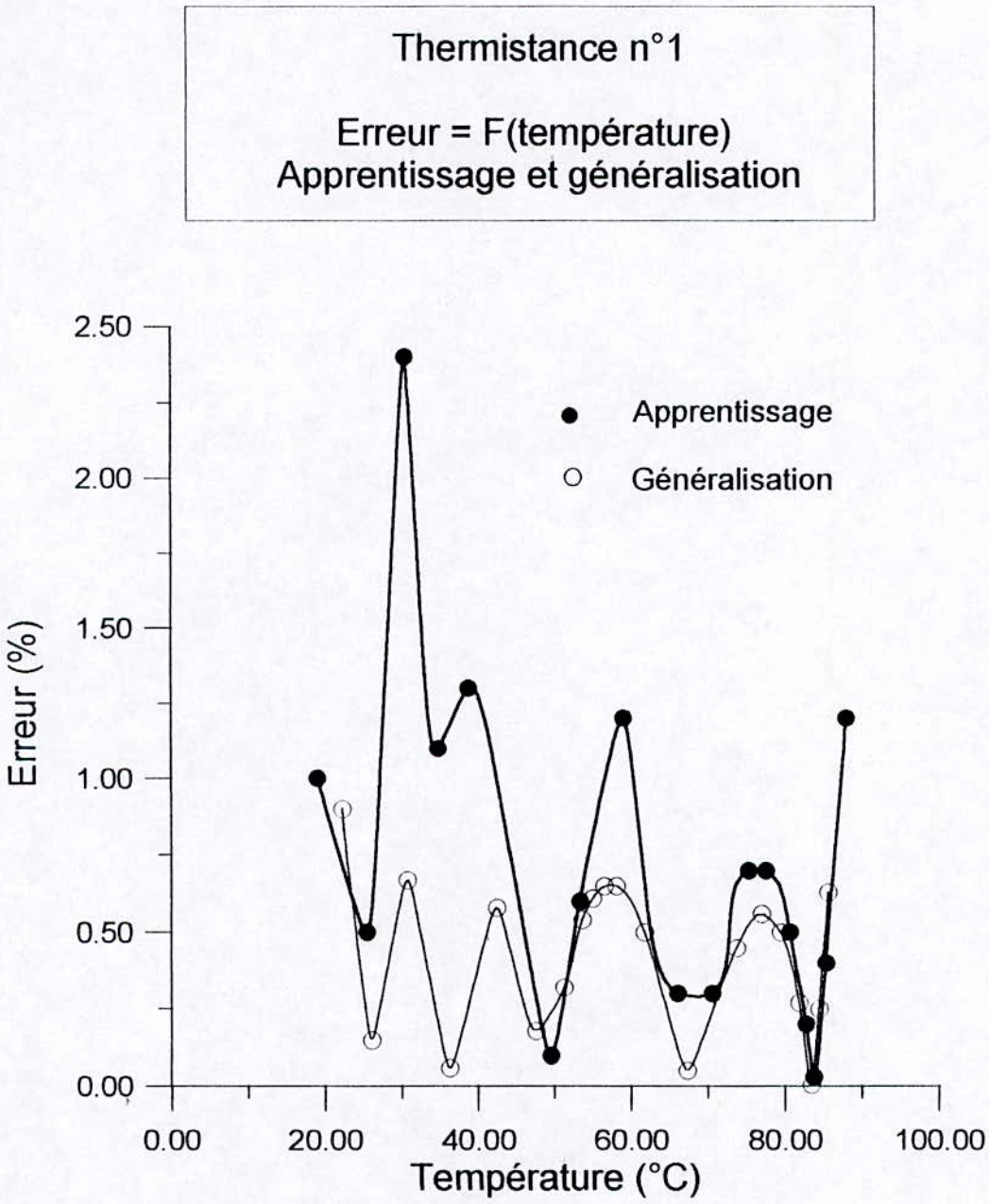


Fig.5.7 Erreurs d'apprentissage et de généralisation

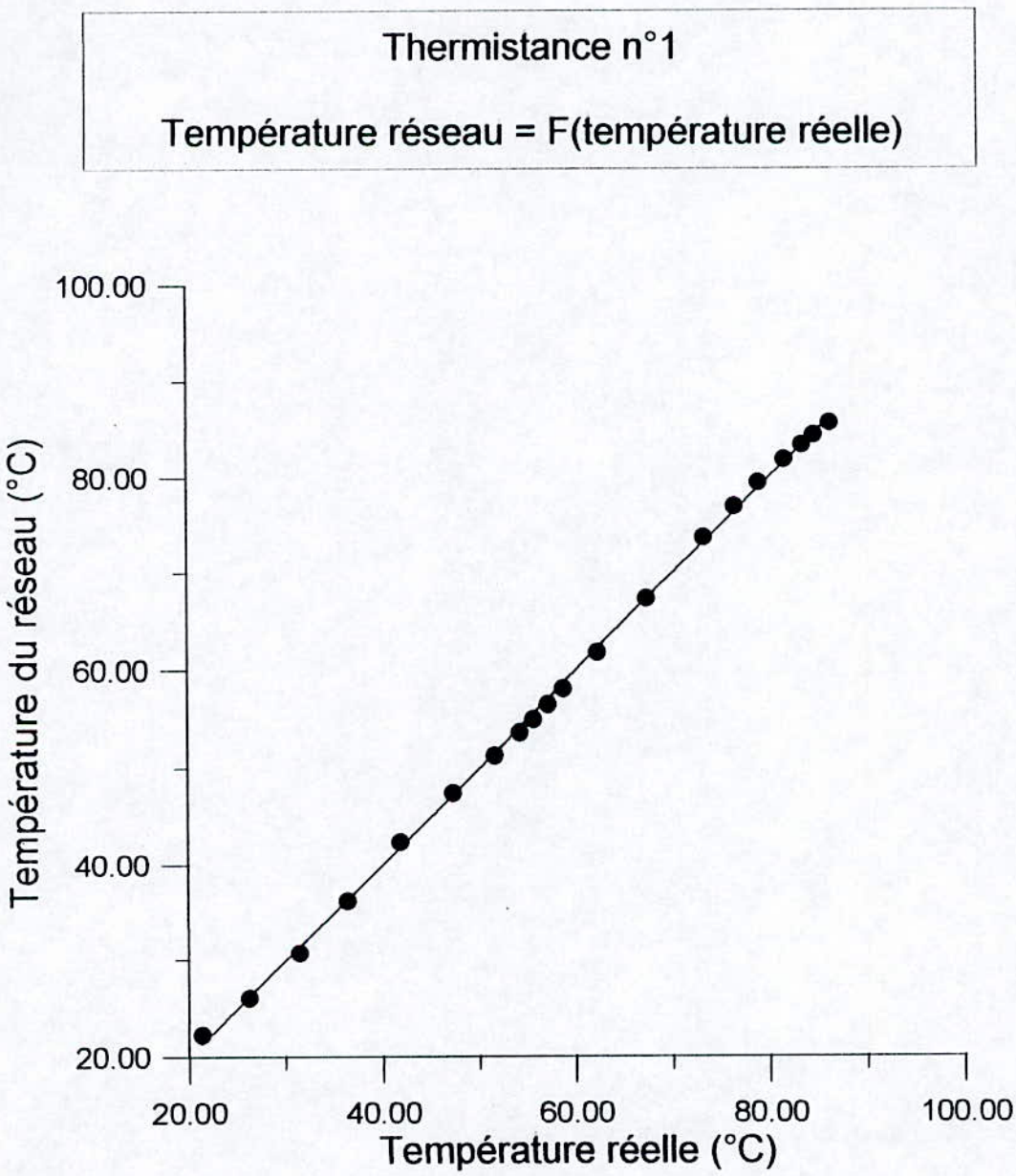


Fig.5.9 Comparaison entre la température à la sortie du réseau et la température réelle

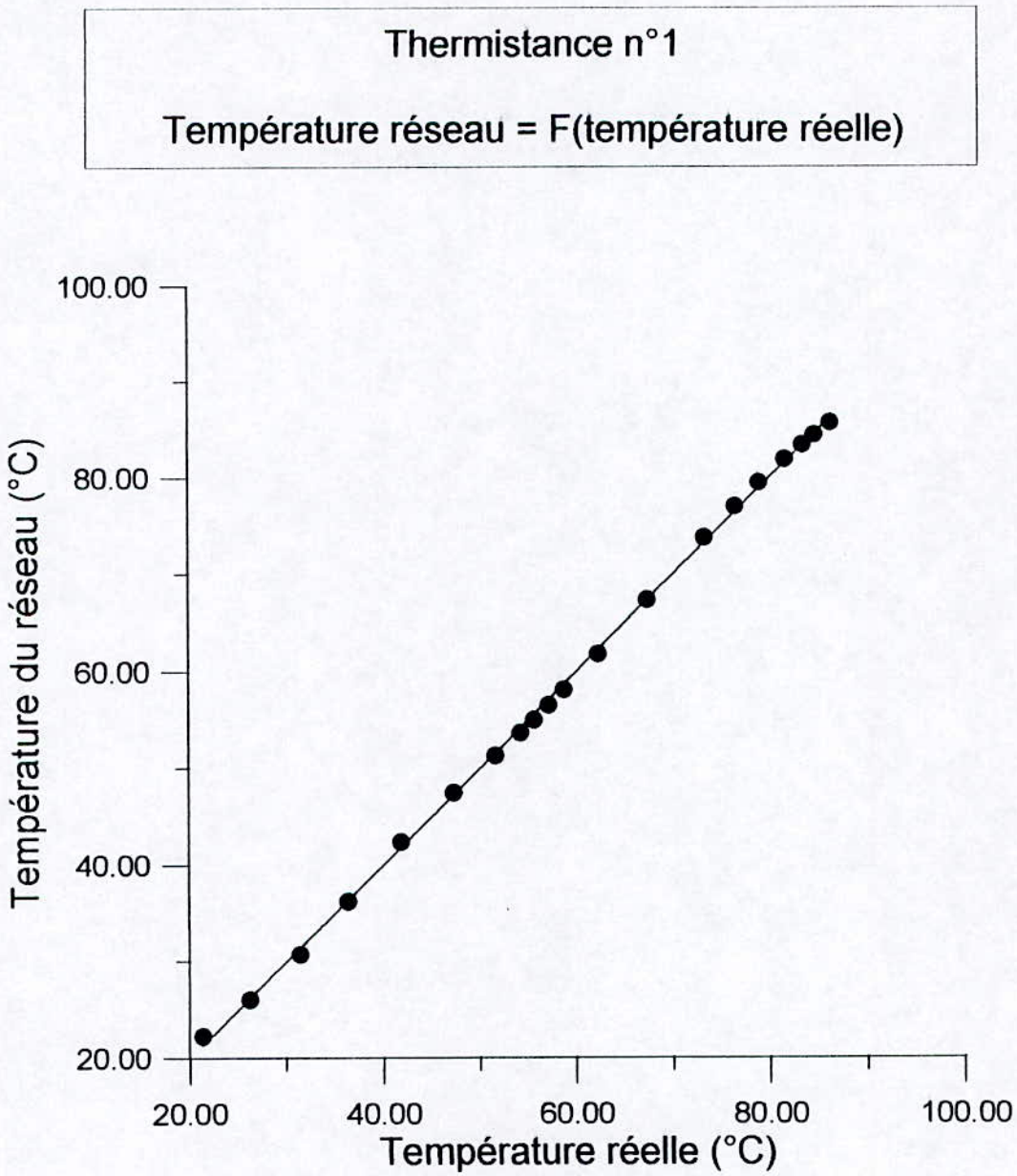


Fig.5.9 Comparaison entre la température à la sortie du réseau et la température réelle

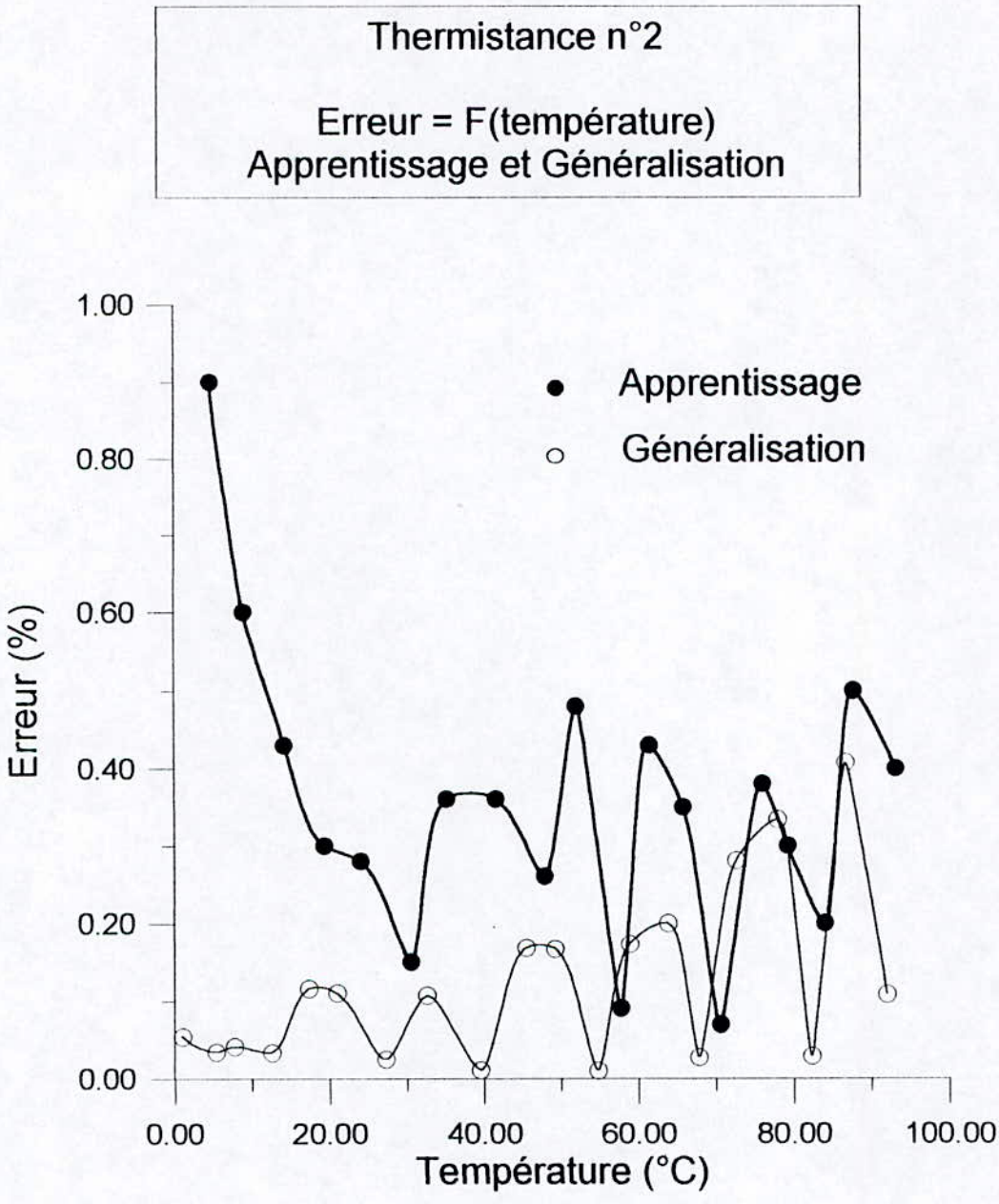


Fig.5.10 Erreurs d'apprentissage et de généralisation

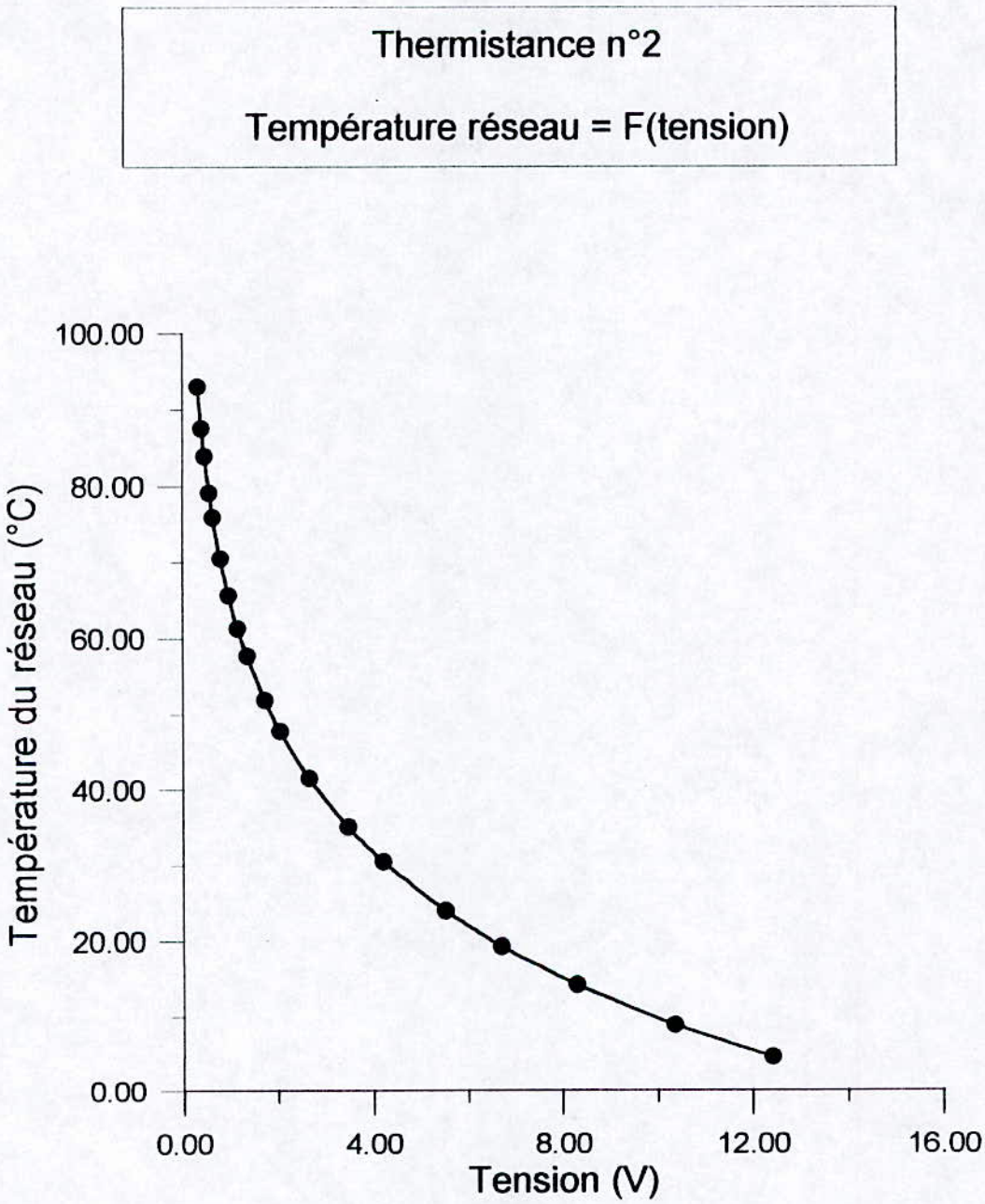


Fig.5.11 Caractéristique inverse simulée par un RNA après apprentissage

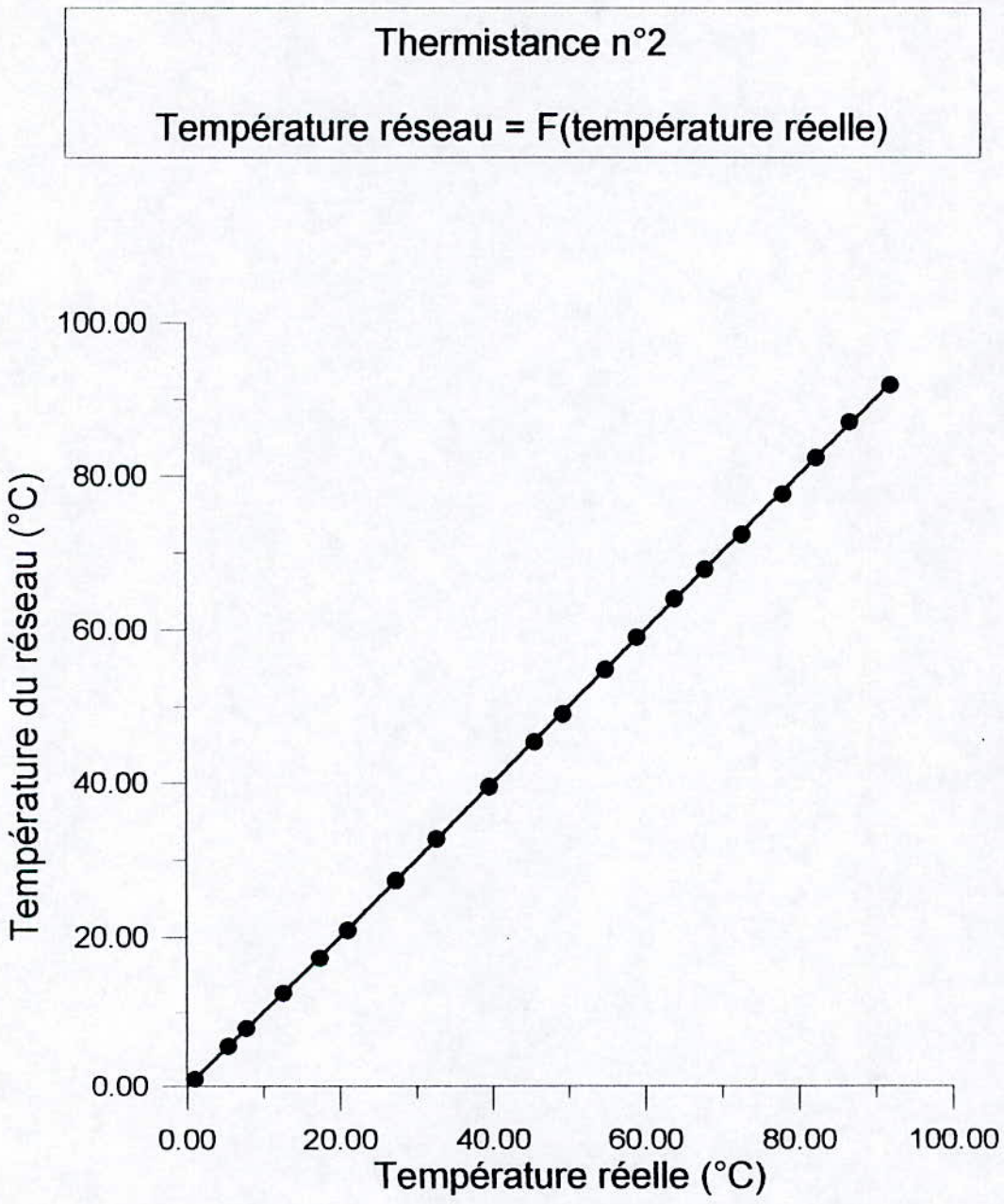


Fig.5.12 Comparaison entre la température à la sortie du réseau et la température réelle

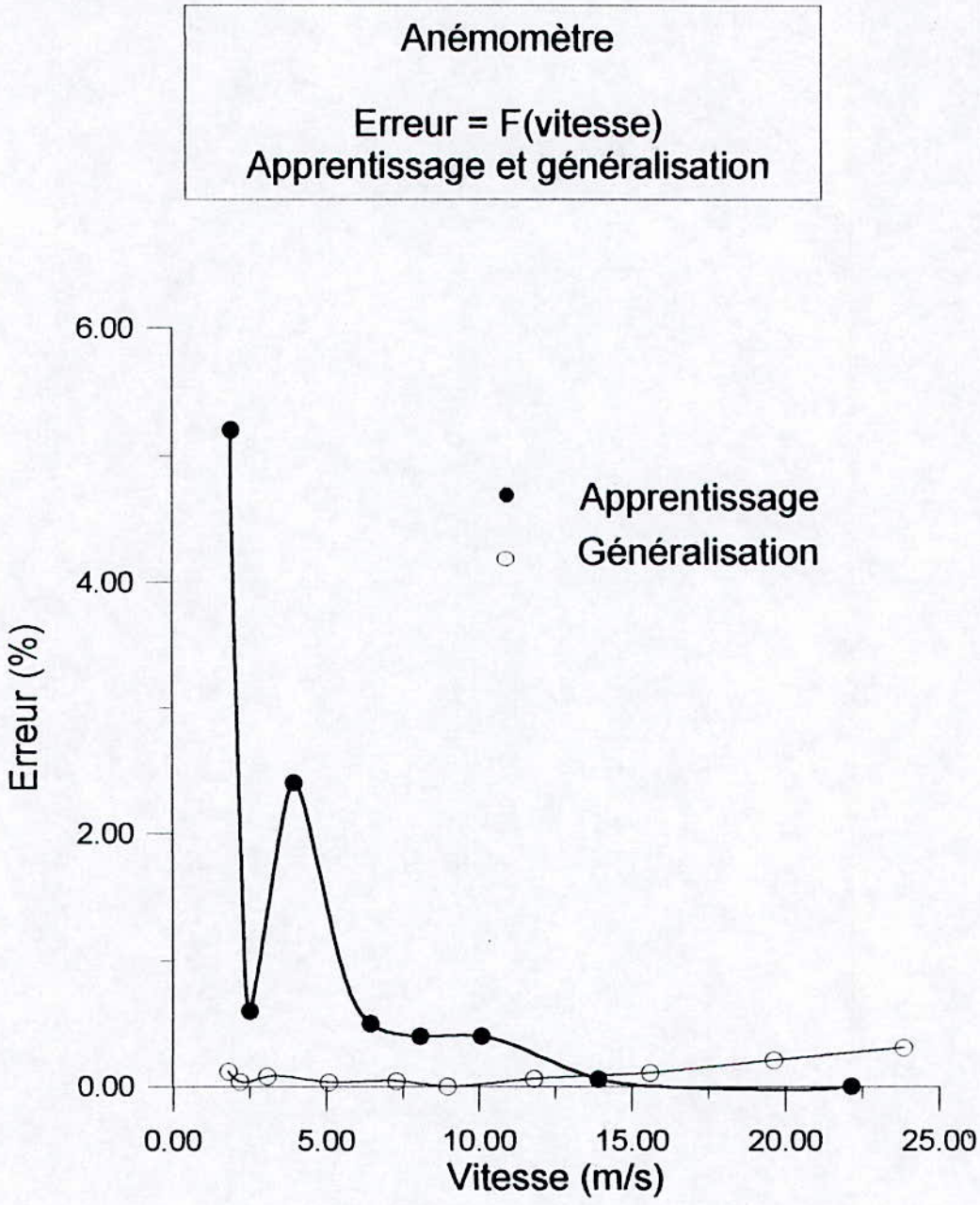


Fig.5.14 Erreurs d'apprentissage et de généralisation

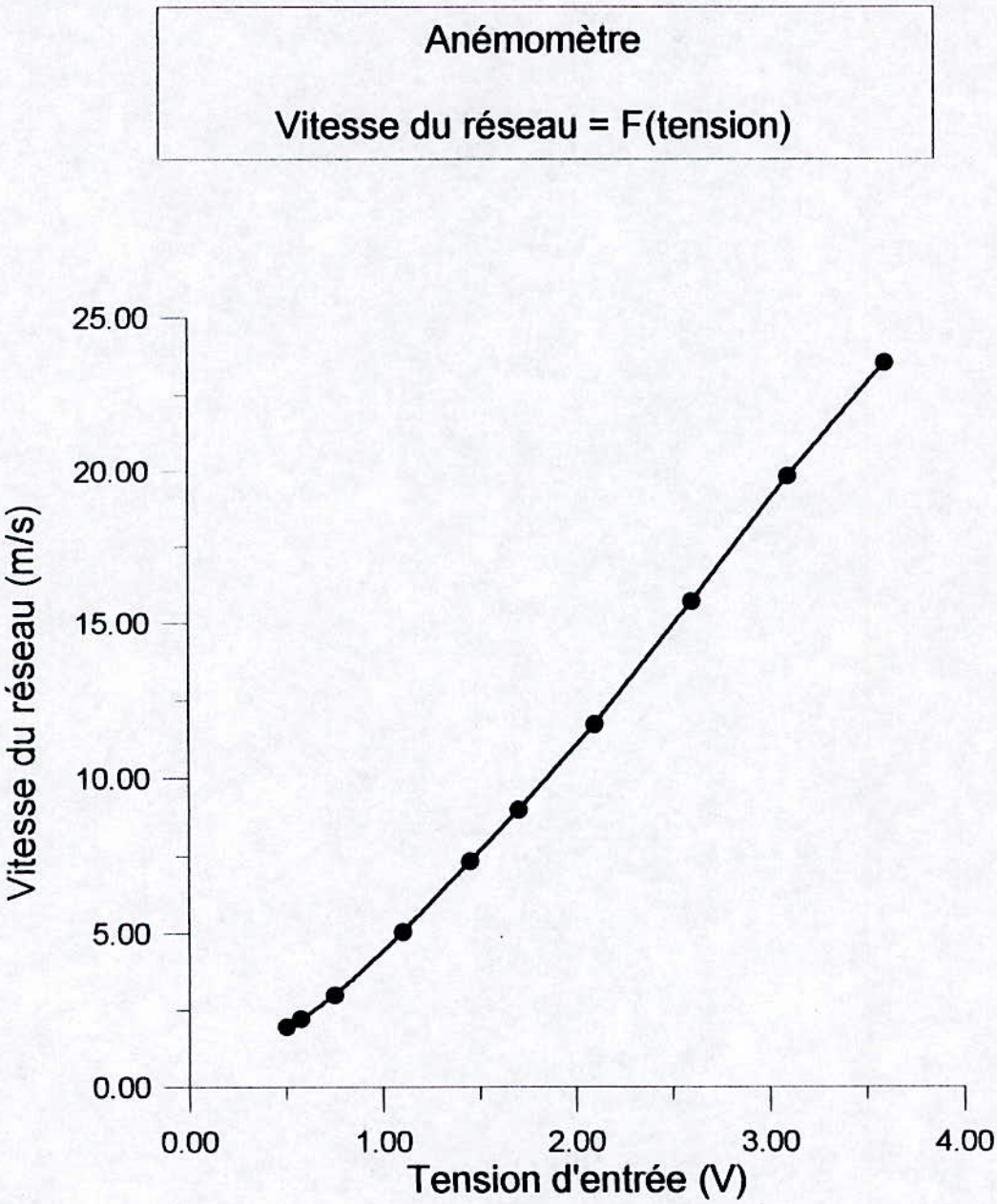


Fig.5.15 Caractéristique inverse simulée par un RNA après apprentissage

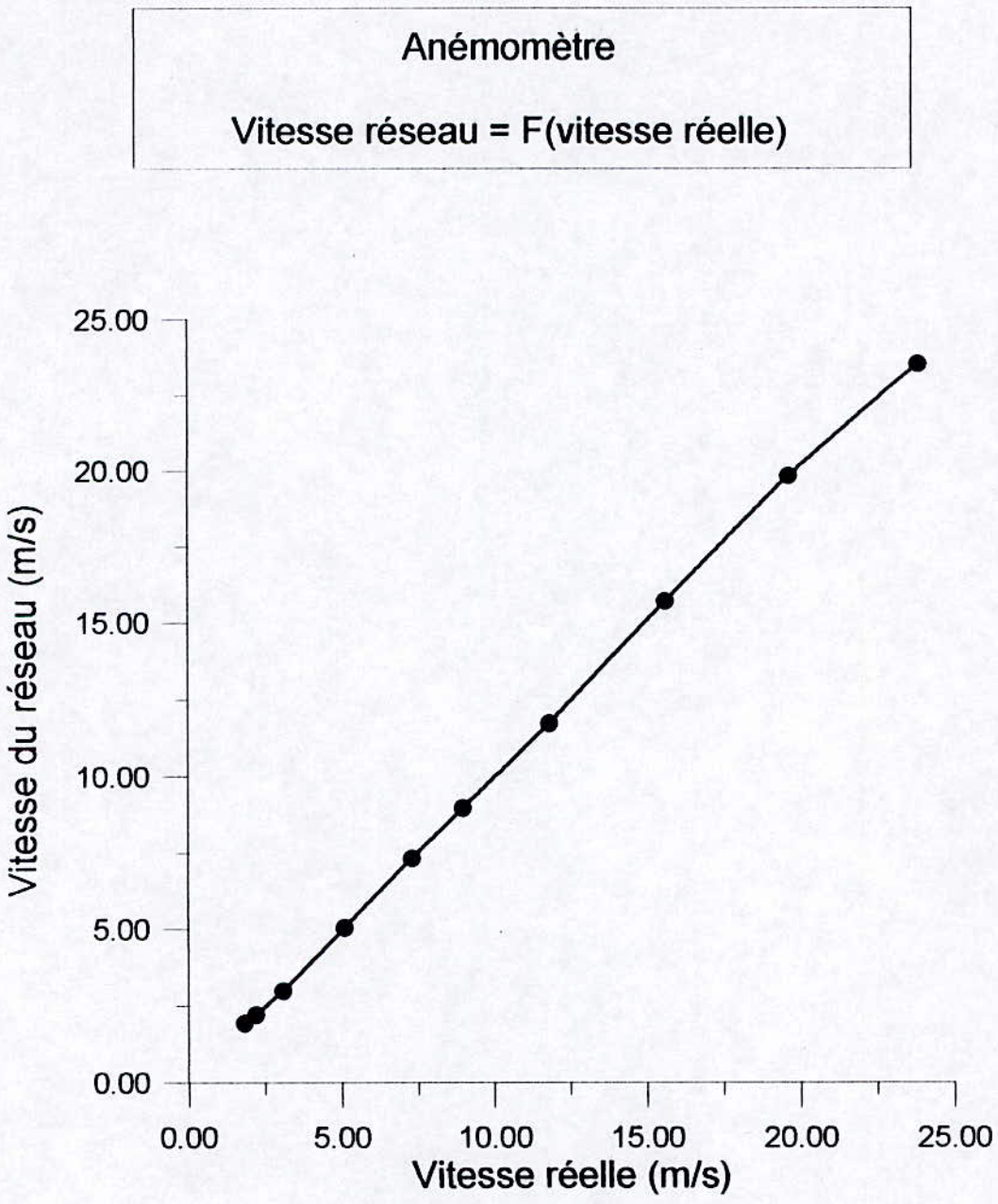


Fig.5.16 Comparaison entre la vitesse à la sortie du réseau et la vitesse réelle

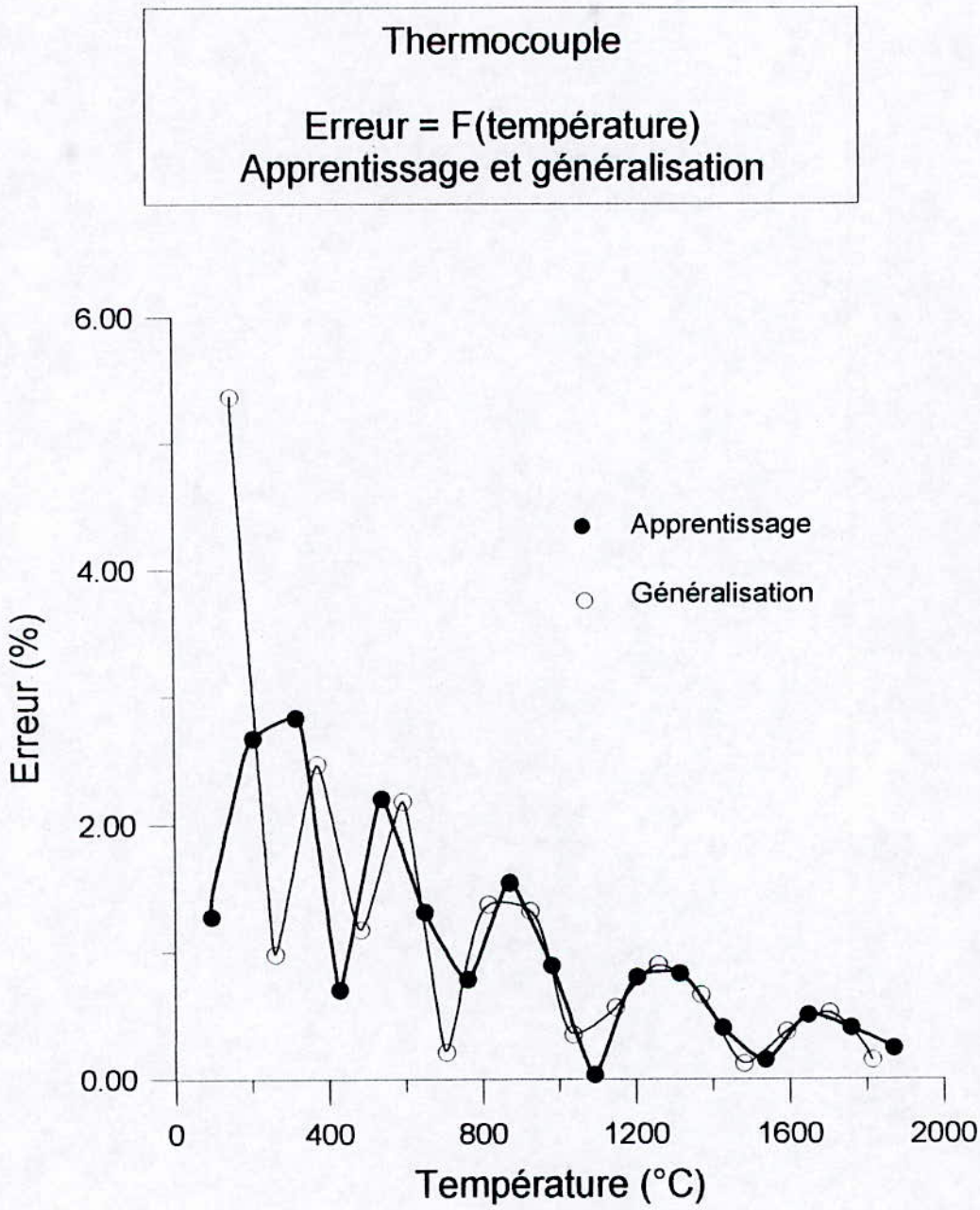


Fig.5.17 Erreurs d'apprentissage et de généralisation

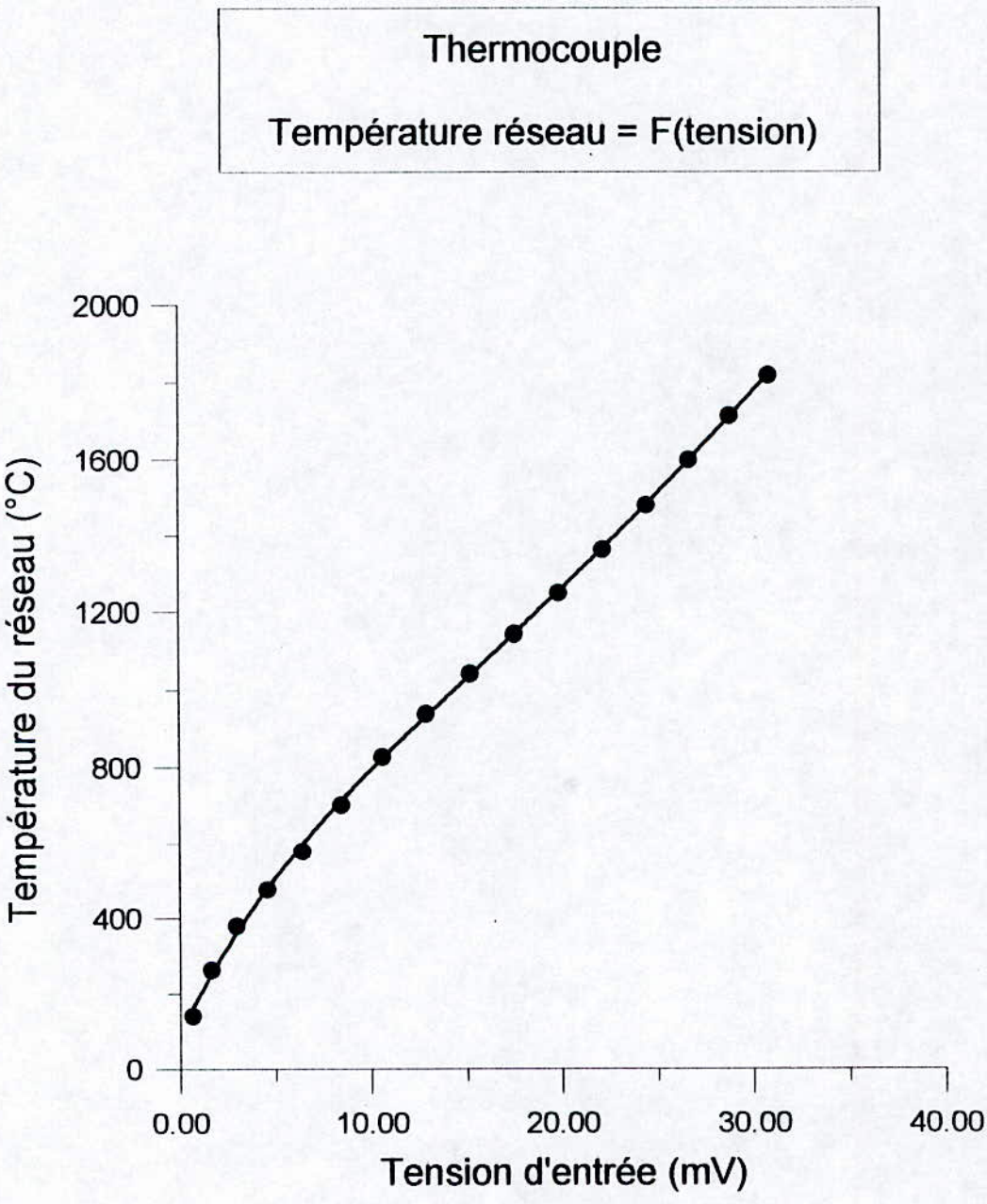


Fig.5.18 Caractéristique inverse simulée par un RNA après apprentissage

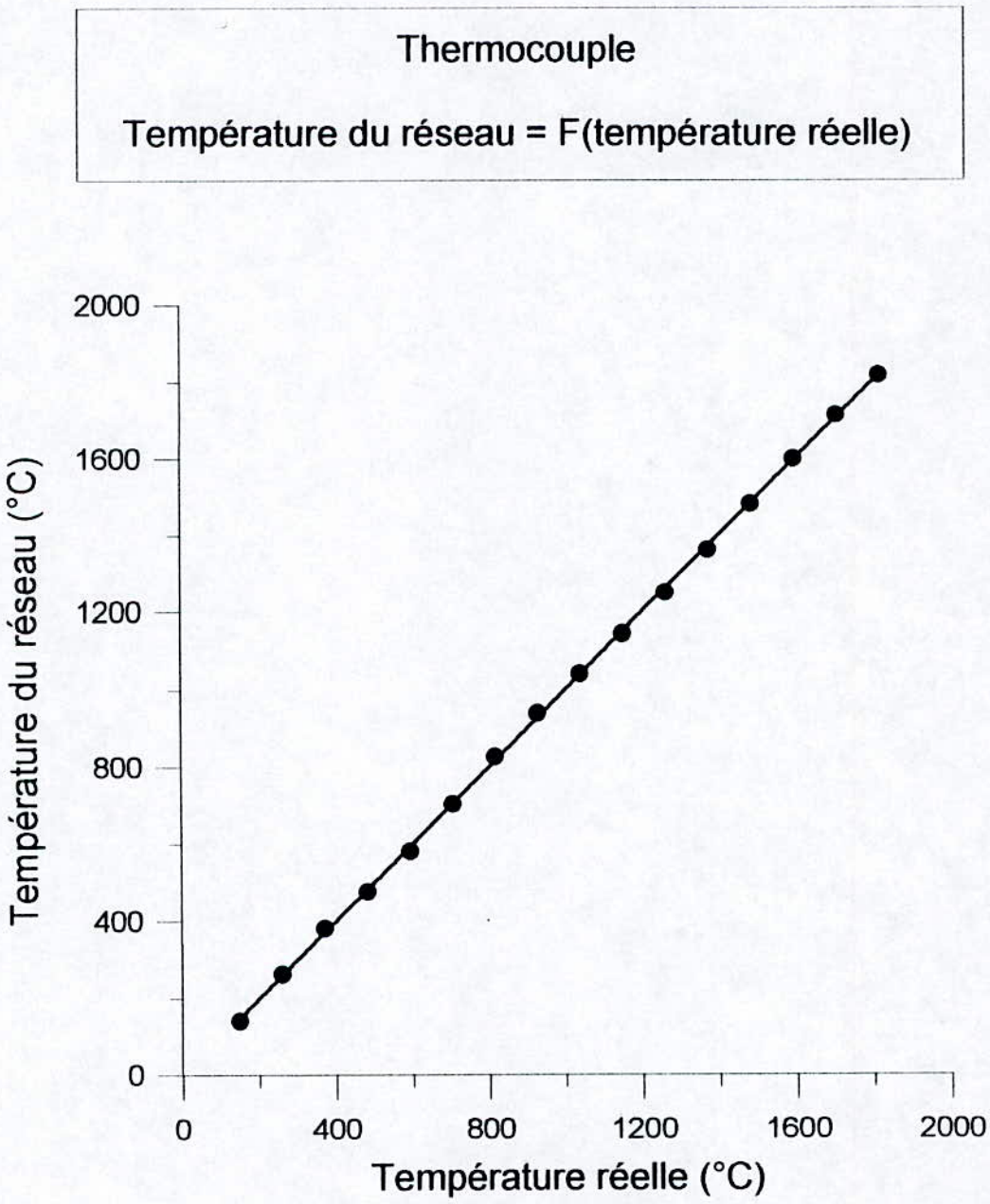


Fig.5.19 Comparaison entre la température à la sortie du réseau et la température réelle

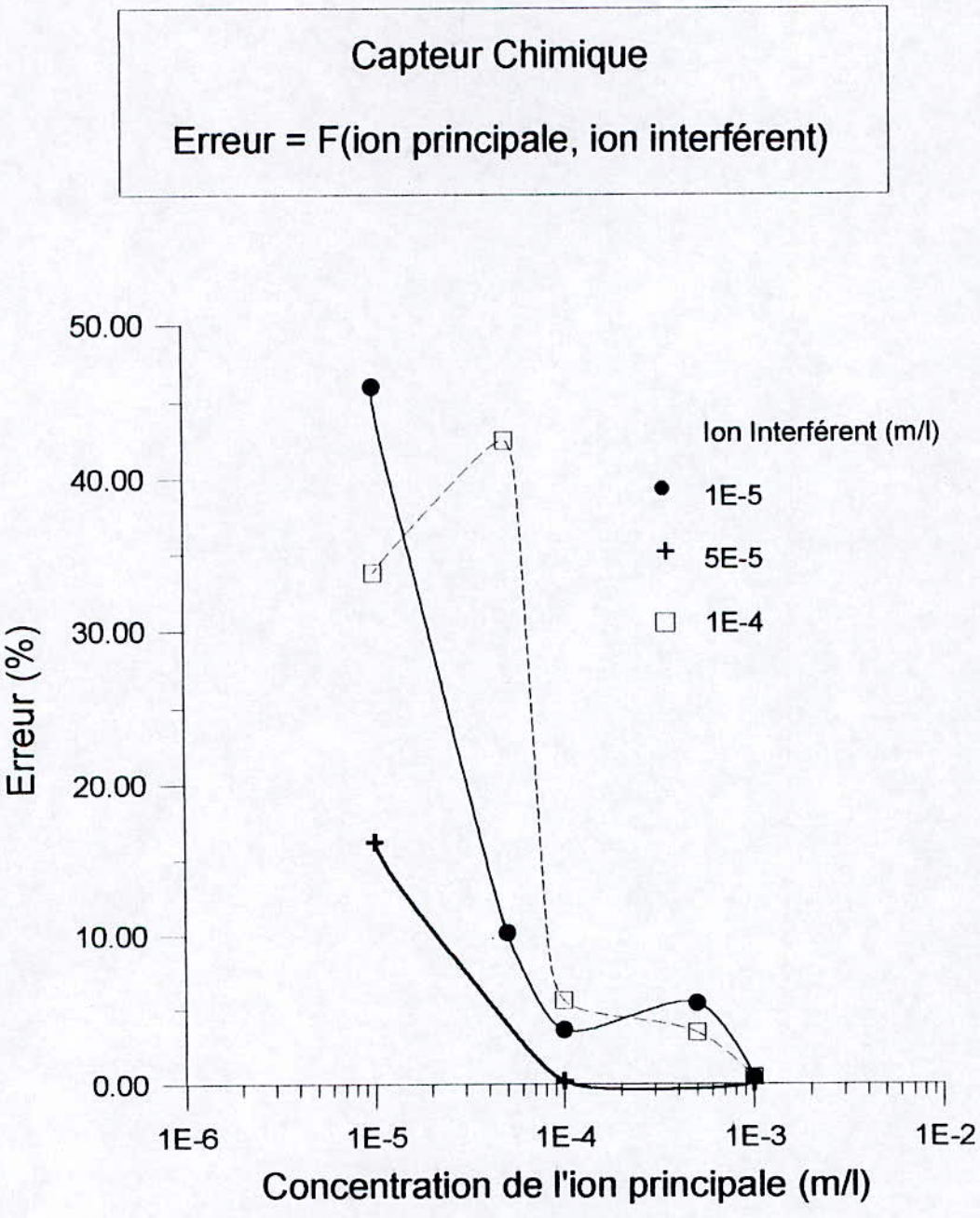


Fig.5.20 Erreurs d'apprentissage
pour 3 concentrations de l'ion interférent

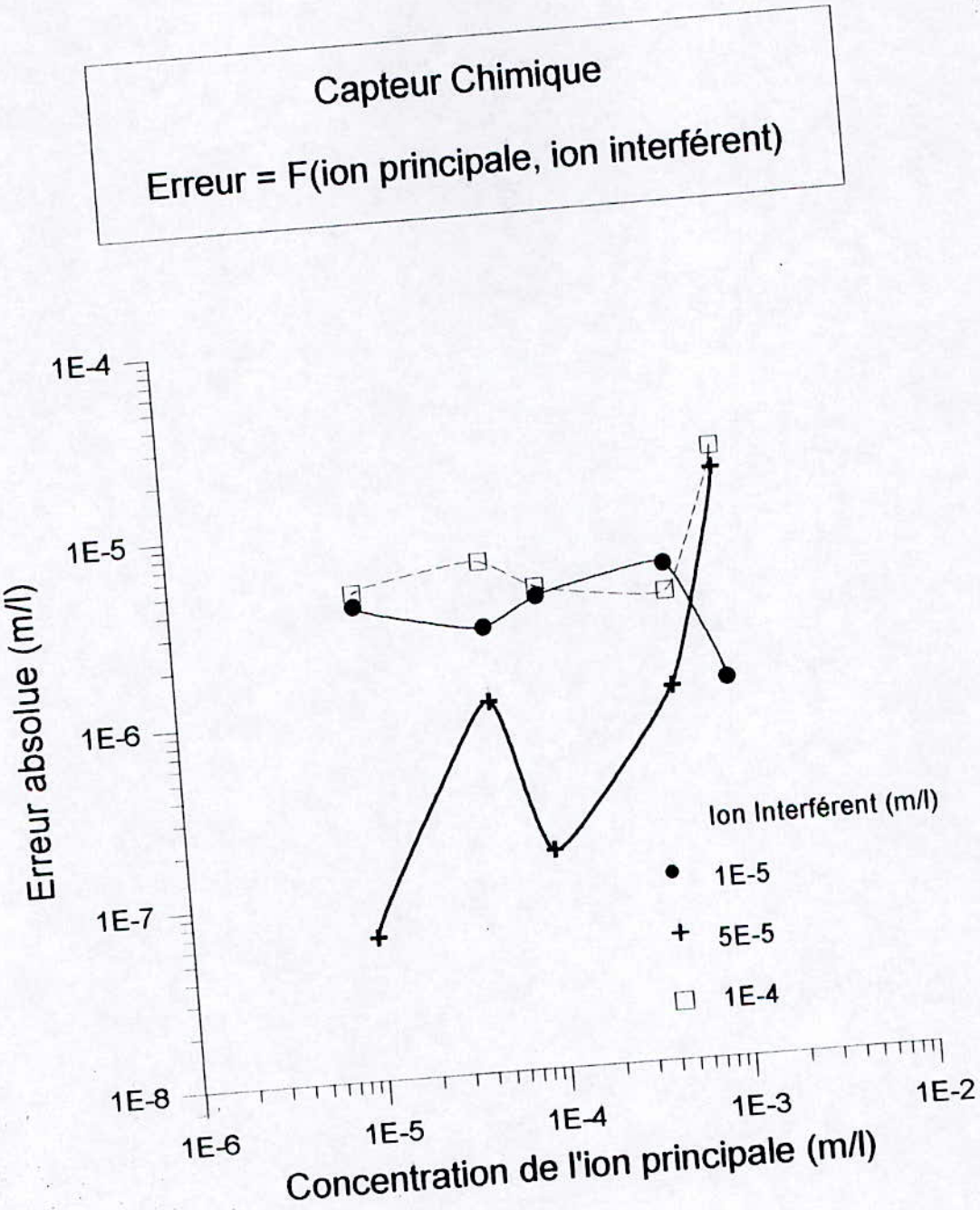


Fig.5.21 Erreurs de généralisation
pour trois concentrations de l'ion interférent

PERSPECTIVES

Cette partie illustre quelques applications aux quelles peut être joint notre travail.

Tout d' abord, on va présenter la chaine d' acquisition des données à partir du capteur jusqu' au RNA . En effet , si cette méthode de linéarisation venait à être réalisée, le réseau neuronal peut
être conçu sous forme de programme installé sur une EPROM et qui a pour rôle,de calculer la sortie du réseau en tenant compte du fichier poids déjà adaptés dans la phase d' apprentissage.

Cette chaine comportera donc, le capteur en question muni de son circuit de conditionnement plus un filtrage de sortie. Nous ajouterons à cela un étage d' amplification, ou un amplificateur d' instrumentation qui servira à amplifier la variation de la mesure en fonction de la variation du mesurande. Un convertisseur analogique digital nécessaire à l' entrée du micro processeur, celui-ci sera connéctée à une RAM et une EPROM dans laquelle se trouve le programme de linéarisation par RNA.
Le système à microprocésseur est reliè à un afficheur alphanumérique afin de visualiser les parametres mesurés.
Cette explication est illustrer dans la figure.

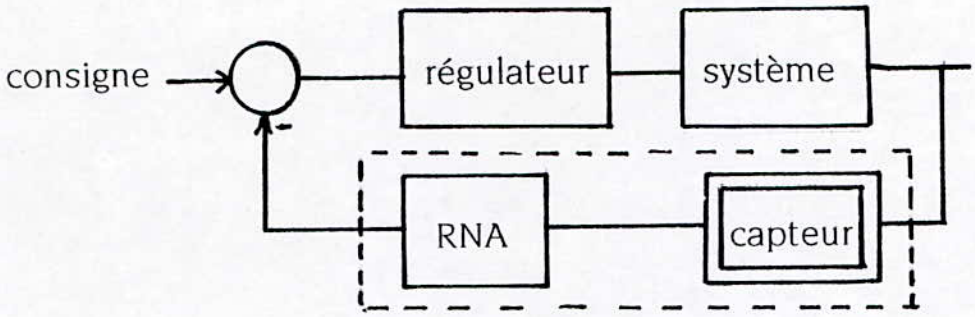
Remarque :

Notre travail d' apprentissage et de généralisation a été effectué en grande partie sur VAX . 128 Digital.

Sur cet ordinateur on a évalué le temps de réponse de notre réseau à l' ordre de quelques micro secondes le programme étant démuni de toute instruction d' affichage ou de commentaires. Ce temps de réponse dépend bien sur du matériel sur lequel on a travaillé . Il peut être donc amélioré si on arrive à travailler avec un micro processeur yant une vitesse d' horloge beaucoups plus rapide, on peut même atteindre l' ordre des microsecondes.

- Dans le domaine de la commande, cette étude est bénéfique dans la mesure ou elle nous évitera d' utiliser les techniques non linéaires pour la synthèse d' un retour d' état ou d' un régulateur classique.

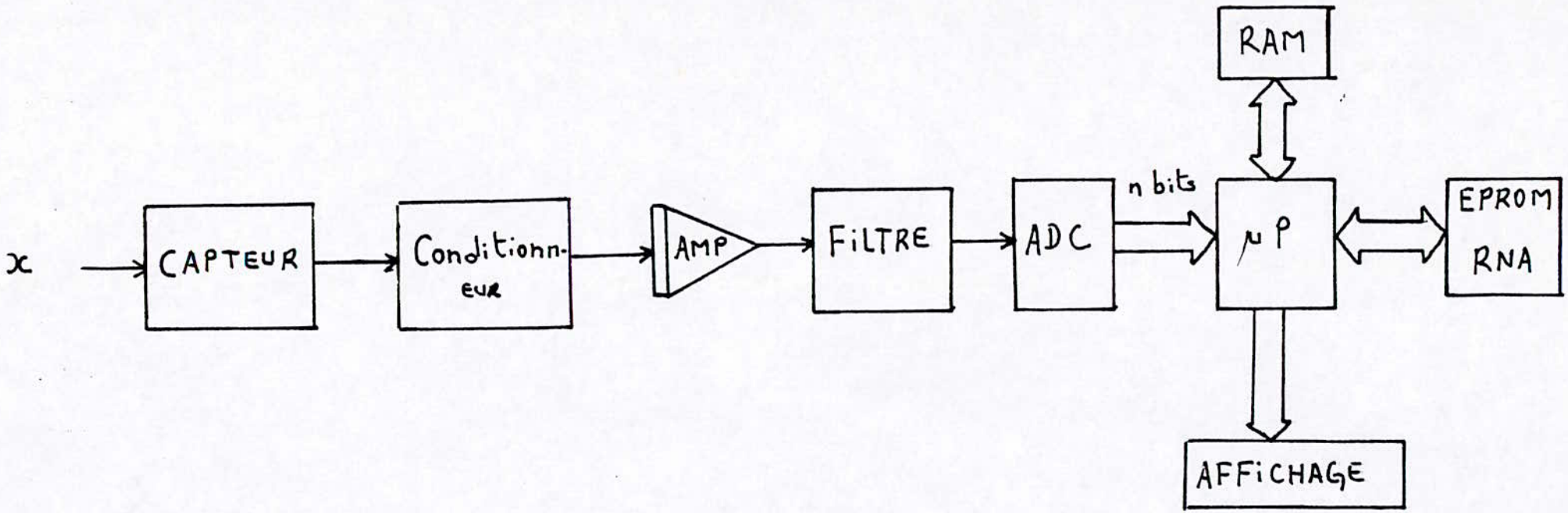
Effectivement, le RNA relié en cascade avec le capteur non linéaire servira a nous reproduire en sortie une image de la grandeur à mesurer et en plus avec un gain de 1: Soit alors une relation linéaire et un retour unitaire de la sortie de notre processus vers le comparateur voir la figure suivante :



Bloc linéaire

Ceci nous pousse à dire que cette manière de linéarisation peut même être exploité pour linéariser le système à commander si celui-ci est non linéaire. C' est ce qui a été fait par une équipe de recherche allemande pour la commande d' un laminoir. [20] Cette technique de linéarisation s' applique aussi bien de la boucle de linéarisation que dans la chaine de mesure.

Enfin, le RNA peut ètre réaliser analogiquement à l'aide de circuits appropriés à base d' amplificateurs opérationnels mais cette fois ci sa realisation sera surement onéreuse.



Chaine d'acquisition des données : capteur plus RNA .

Conclusion générale

Ce présent travail nous a permi de sortir avec une expérience et des conclusions qui sont les suivantes:

Les RNA ont démontré durant ce travail leur capacité de simuler certaines fonctions non linéaires notamment celle de type dur et celle de type mou. Ce dèrnier est beaucoup plus facile àimiter par le RNA car il ressemble au type de la non linéarité de la fonction d' activation.

La qualité de l' apprentissage est souvent du ressort de l' éxpérience celle ci nous a permi de faire collaborer les deux algorithmes d' entrainement à savoir BP et ROM, cette manière de faire à améliorer et de loin la qualité de la généralisation.

ceci étant, les RNA ont eu un bon comportement pour la linéarisation de capteurs sans grandeur de perturbation, c' est ce qu' on a apperçu avec la thermistance, l' anémomètre, et le thermocouple. la qualité de la linéarisation et meme très acceptable relativement aux autres methodes de linéarisation analogique et digitale.

Elle nous permet aussi d' élargir le domaine de linéarisation et donc de se débarasser en quelque sorte de la linéarisation au tour d' un point de fonctionnement.

Dans le cas de la présence d' une grandeur de perturbation l' utilisation des RNA n' est pas à rejeter, seulement il faut la prendre avec précaution et d' essayer de jouer sur les paramètres qui peuvent ameliorer la qualité de l' erreur finale, à savoir l' architecture du RNA, les fonctions d' activations, les exemples d' apprentisssage etc...

Enfin, cette méthode de linéarisation à l' aide des RNA apporte un avantage remarquable en commande des systèmes car elle nous évite d' utiliser des téchniques non linéaires de synthèse surtout dans le cas simple ou la grandeur de perturbation n' est pas prépondérente.

ANNEXES.

ANNEXE 1

Règle de Hebb : [10]

C' est une méthode qui joue sur la dynamique des connexions, c' est à dire à l' évolution des pondérations du réseau neuronal Via les synapses, qui peuvent être escitatrices, si les poids sont important ou inhibitrices dans le cas contraire.

La règle de Hebb a des fondements biologiques, son interprétation est la suivante :

Si deux neurones connectés entre eux, sont activés au même moment, la connexion qui les relies doit être renforcée; autrement, elle n' est pas modifiée. Cf figure (A-1).
Ceci explique bien que si la connexion entre deux cellules est forte, l' excitation de l' une d' elle peut engendrer l' activation de l' autre, il faut donc augmenter le poids de cette connexion.

Si on note : Wij (t) le poids du lien entre le neurone i et le neurone j à l' instant t, et Ai et Aj leurs activations respectives, prenant des valeurs binaires (0, 1) la règle de Hebb sur l' évolution du poids, se traduit par :

$$Wij (t + \Delta t) = Wij (t) + \mu Ai Aj$$

μ : est un paramètre de l' intensité de l' apprentissage.

Remarque :

Au fait, le principe d' apprentissage est basé sur deux issues :

* Les règles biologiques, telle la règle Hebb .

* Les règles mathématiques, basées sur le calcul des paramètres de la fonction de transfert, en imposant une certaine fonctionnalité .

Cette deuxième a abouties à des algorithmes, comme celui du percéptron et de la rétropropagation (BP).

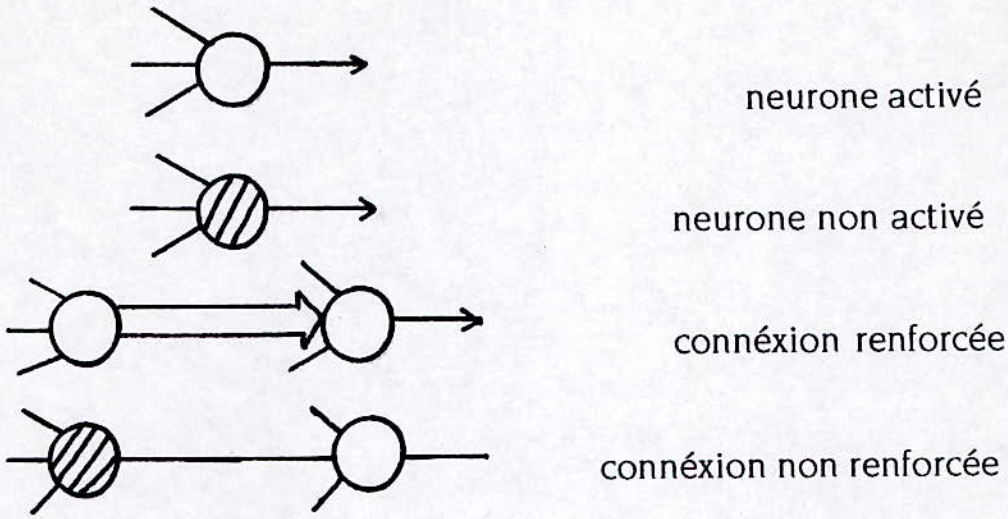


fig (A-1) : Variation des connexions par la règle de Hebb.

ANNEXE 2

Méthode d' optimisation aléatoire : [9]

Soit f une fonction définie de IR vers IR, S un sous ensemble de IR^n .

On veut trouver le point x de S qui minimise f .

* Etape 1 : Choisir x (0) dans S et poser k = 0

* Etape 2 : Générer g (k) dans l' espace (IR^n , B, μ (k))

* Etape 3 : Laisser x (k + 1) = D (x (k), g (k))
Choisir μ (k + 1), et poser k = k + 1.

Avec : μ (k) correspondant à la fonction de distribution définie sur IR^n

B : un sous ensemble de Borel

g (k) : vecteur Gaussien .

De cet algorithme, on va tirer la première hypothèse de convergence vers le minimum :

$H1 = f(\ D(x,\ g)\ ) \leq f(x)$
de plus si $g \leq s$: $f(\ D(x,\ g)\ ) \leq f(g)$

Si le minimum de f est atteint pour un point ou la fonction est singulièrement discontinue, il y a de fortes chances de ne pas trouver ce point, à moins qu' on effectue un test point du domaine S .

C' est pour cela qu' on remplace la recherche du minimum par celui de a , définie par :

$$a = \min \{ t : v \mid x \in S / f(x) < t \mid > 0 \}$$

avec v(A) et le volume de l' ensemble a , appelé généralement la mesure de Lebesgue .

on définit , alors la région d' optimalité par :

$$R(e,M) = \begin{cases} \{ x \in S / f(x) < a + e \} & \text{si a est fini} \\ \{ x \in S / f(x) < M \} & \text{si } a \longrightarrow \infty \end{cases}$$

$e > 0$ et $M < 0$.

pour démontrer la convèrgence de l' algorithme , on a ajouté une autre hypothèse :

H2: pour n' importe quel sous ensemble A de S ou $v(A) > 0$ on a :

$$\prod_{k=0}^{\infty} | 1 - \mu_k(a) | = 0$$

c' est à dire , la probabilité d' omettre l' ensemble A , répétitivement lors de la génération de g (k) doit être nulle.

Theorème de convèrgence : (global search) [9]

On suppose la fonction f mesurable , et les hypothèses H1 et H2 satisfaites.

$$| x(k) |_{k=0}^{\infty}$$

est la suite générée par l' algorithme.

Alors :

$$\lim_{k \longrightarrow \infty} P | x(k) \in R(e,M) | = 1$$

car de l' hypothèse H1 :
si x(k) ou g(k) appartiènnent à R(e,M) , cela implique que :

$$x(k') \in R(e,M) \qquad \text{pour tout } k' \geq k+1$$

ainsi :

$$P | x(k) \in R(e,M) | = 1 - P | x(k) \in S / R(e,M) |$$

$$1 - P[\, x(k) \in S / R(e,M) \,] \geq 1 - \prod_{i=0}^{k} (1 - \mu i\,(\, R(e,M)\,))$$

d'ou :

$$1 \geq \lim_{k \to \infty} P[\, x(k) \in R(e,M) \,] \geq 1 - \lim_{k \to \infty} \prod_{i=0}^{k-1} (\, 1 - \mu i\,(\, R(e,M)\,)\,)$$

donc :

$$\lim_{k \to \infty} P[\, x(k) \in R(e,M) \,] = 1$$

Remarque:

Il existe plusieurs choix de D (x(k) , g(k)) , dans notre cas on prendra :

$$D\,(\, x(k)\,,\, g(k)\,) = x(k) + g(k)$$

ANNEXE3

Génération des signaux numériques [30]

Chaque signal déterministe x (t), peut être numérisé en l' échantillonnant à des instants déchantillonnage déterminés soit alors :

$$x(kT) = x(t)_{t=kT}$$

avec T: période d' échantillonage .

Génération des signaux pseudo-aléatoires [30]

Tout signal pseudo-aléatoire a pour forme de récurrence la relation suivante :

$$x(k+1) = | a\, x(k) | \bmod p \qquad (1)$$

A et p étant des entiers.

p: premier

A:racine primitive de p.

Tout signal généré de cette relation (1) est périodique , et de période p-1 .
Le signal :

$x'(k) = x(k) / p$ a pour densité de probabilité la loi uniforme [0,1].

Génération des signaux possédant d' autres densités de probabilités

A partir d' un signal pseudo-aléatoire (1) à distribution uniforme , on peut générer d' autres signaux pseudo-aléatoires possédant d' autres distributions .

exple

$$y(k) = \sqrt{2\sigma^2 \ln (1 / x(k))}$$

Si x(k) suit la loi uniforme , alors y(k) suit la loi de Reyleigh . de plus :

$$z(k) = y(k) \cos (2\pi\, x(k+1))$$

suit la loi de gauss ou loi normale .

Génération d' un bruit gaussien [30]

La suite : $x1(t) = x(t) / (2^n - 1)$ suit la loi uniforme [0,1]
avec :

$$x(k+1) = A\, x(k) \mid 2^n - 1 \mid$$

$A = 131$

$x(0) = 12375$

$n = 16$

$$R(k) = \sqrt{2\sigma^2 \ln (1 / x1(k))}$$

$$z(k) = R(k) \cos (2\pi\, u1(k+1)) + b$$

z(k) est le signal représentant le bruit gaussien de moyènne b.

ANNEXE 4

International Conference on Signal and Systems, ICSS'94, September 24-26, 1994, Algiers

Simulating Nonlinear Functions With Artificial Neural Networks

M. HENICHE[ϕ], F. BOUDJEMA[ϕ] and M. ATTARI[✦]

[ϕ] *Ecole Nationale Polytechnique*
Département de Génie Electrique
ENP, 10 Rue Hassen Badi, El-Harrach, Algiers

[✦] *Université des Sciences et de la Technologie Houari Boumedienne*
USTHB, Institut d'Electronics/Dpt d'Instrumentation
BP.32, El-Alia, Bab-Ezzour, 16111, Algiers

Abstract **—This paper report on the use of an artificial neural network (ANN) for the purpose of simulating nonlinear functions in order to linearize the calibration curve of devices in instrumentation and control. Two algorithms have been used to determine the weights of the net that is backpropagation (BP) and random optimization method (ROM). Besides, the combination of back propagation and random optimization method (BP and ROM) is proposed and compared to the BP and ROM. Among the functions tested, two analytical functions have been detailed in the paper (a hard and weak nonlinearity functions).**

The results of simulation have shown the best way to train the ANN for these two particular functions.

Keywords — Nonlinear functions, Artificial Neural Networks, Backpropagation, Random Optimization Method.

I. INTRODUCTION

Neural networks are broadly useful in a wide range of applications such as in signal and image processing, pattern recognition [1], intelligent control [2] and recently instrumentation [3]. For example, performing even simple filtering operations on images requires a great deal of computation. The brain accomplish this by utilizing massive parallelism, with millions and even billions of neurons in parts of the brain working together to solve complicated problems. Multilayered neural networks are made out of neuron-like nodes that are arranged in layers and pass data through weighted connections. During the learning phase, the weights of the network are changing, and with suitable weights, such a network can model any computable function. A node in a neural network typically multiplies each input by its weight, sums the products, then passes the sum through a sigmoide function to produce a result (Fig.1), this node is called adaptive linear element or Adaline [4].

Developing a neural network is unlike developing software, because the network is trained, not programmed. For some problems, it can be faster than writing traditional software. Moreover, it can solve some problems that have resisted solution by other methods. Nonetheless, neural networks can be difficult to train and unsuitable for some tasks.

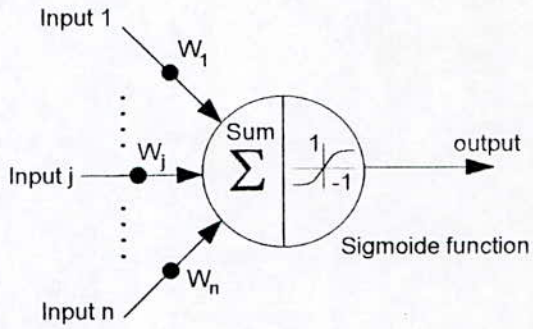


Fig.1 Adaptive linear element of an Artificial Neural Network (ANN)

Furthermore, neural networks offer valuable characteristics unavailable together elsewhere. First, they can infer unknown relationships from data. Second, the networks can generalize, meaning they can respond correctly to patterns that are only broadly similar to the original training patterns. In our case, generalization is useful because real-word-data is distorted and often incomplete. Third, they are nonlinear, that are very useful in solving complex problems more accurately than linear techniques. Among the application of ANN in the field of control systems, Hofer *et al* [5] have shown that the neural network can be used successfully to control the strip thickness of a still rolling mill. In power systems, Aggoune *et al* [2] have proposed an adaptive variable structure voltage regulator via an artificial neural network. This is used to adapt the sliding surface to the power system operating conditions.

In this paper the artificial neural network are used in order to simulate some nonlinear functions. Our goal is to linearize the calibration curves of devices in instrumentation and control. The idea is to choose an appropriate network and a correct training method to learn the selected function. Fig.2 explain this and show the feedback of error via the training method to adjust the weights at the correct values.

For this purpose the backpropagation [4] and random optimization method [6], [7] are chosen to train the neural network. Simulation results of both training algorithms have shown the difference of these two algorithms.

The advantage of the neural network is its capability to learn something different from what its trainer had in mind. This caracteristic is useful in this application because nonlinear function has infinite points. Generalization curves are given to show the performance of the methods used in the training phase. Otherwise, the combination of BP and ROM is tested in order to compare generalization results to those obtained by BP and ROM (independently).

The BP is computed firstly for a finite number of iterations. The weights obtained by this method are considered in the initialization phase of the ROM. For the two analytical functions tested, two hidden layers with ten nodes each are chosen and the neural network is singal input singal output (Fig.2).

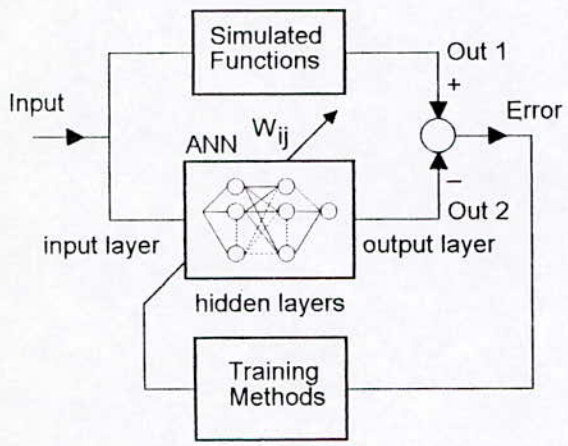


Fig.2 How to learn the nonlinear function according to the training data.

II. TRAINING ALGORITHMS

To train the neural network, Back propagation and Random Optimization Method were tested to adjust the weights of the Net. The first one uses the steepest descent method to calculate the weights of the neural network which satisfy a certain minimum, and the second one is based on random search techniques. The combination of the two algorithms constituts an alternative method to train the ANN.

A. BackPropagation (BP)

This algorithm was found by Werbos in 1974 [8], and then developed by Rumelhart *et al* [9]. BP is a supervised learning method in which an output error signal is fed back through the network, altering connection weights so as to minimize that error (Fig.3).

To simulate a nonlinear function, firstly sampled data were taken and then trained the ANN for the finite points chosen. Each hidden node multiplies every input by its weight and sums the product and then passes the sum through the sigmoide function. The output node of the neural network is compared to the value of the nonlinear function to calculate the error.

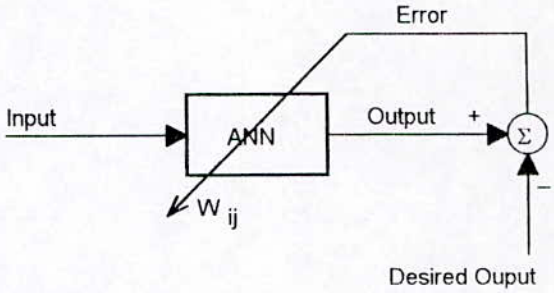


Fig.3 Learning the ANN by BP

Backpropagation is a procedure for finding the weights minimizing the sum of squares error. This minimization has an intuitive geometric meaning [10] that the sets of weights is plotted against the corresponding sum of squares error. The result is an error surface shaped like a bowl, whose bottom marks the set of weights with the smallest sum of squares error. Finding the bottom correspond to the best set of weights and that is the goal of training. The sum of squares error, called also the cost function is described as.

$$\Im(W) = \sum_{i=1}^{m} (s_i - y_i)^2 \qquad (1)$$

and updating the weights of the ANN using the method of steepest descent with the gradient is represented by,

$$W_{k+1} = W_k + \mu(-\nabla_k) \qquad (2)$$

where,

$$\nabla_k = \frac{\partial \Im}{\partial W_k} \qquad (3)$$

The algorithm calculates the instantaneous slope of the error surface with respect to the current weights. It then incrementally changes the weights by a step μ in the direction of the locally steepest path toward the bottom of the bowl.

B. Random Optimization Method (ROM)

This method is based on random search technique [7] and was used with success in several optimization problems. It is more suitable in some cases where the error function has several minima. Furthermore, the ROM gives best results than BP when the neural net is big and time is a crucial parameter in the training phase [6].

Notice that the ROM converge to a global minimum with a probability equal to 1 in a compact set. The idea of this algorithm is to attribute to the weights of the net a set of a white noise sequence and then compute the network's output with these new set of weights. If the error between the calculated output and the desired output is less than the previous error, the new weights are kept, otherwise the previous weights are kept until the desired error is obtained (Fig.4). The algorithm used for this method is given by:

Step 1. Select an initial point $\mathbf{x}^{(0)}$ and let $k = 0$. Let M be the total number of steps.

Step 2. Generate Gaussian random vector $\xi^{(k)}$. If $\mathbf{x}^{(k)}+\xi^{(k)} \in X$, go to Step 3. Otherwise, go to step 4.

Step 3.

(i) If $f\left(\mathbf{x}^{(k)}+\xi^{(k)}\right) < f\left(\mathbf{x}^{(k)}\right)$, let

$\mathbf{x}^{(k+1)} = \mathbf{x}^{(k)} + \xi^{(k)}$

and $\mathbf{b}^{(k+1)} = 0.4\xi^{(k)} + 0.2\mathbf{b}^{(k)}$.

(ii) If $f\left(\mathbf{x}^{(k)}+\xi^{(k)}\right) \geq f\left(\mathbf{x}^{(k)}\right)$

and $f\left(\mathbf{x}^{(k)}-\xi^{(k)}\right) < f\left(\mathbf{x}^{(k)}\right)$, let

$\mathbf{x}^{(k+1)} = \mathbf{x}^{(k)} - \xi^{(k)}$

and $\mathbf{b}^{(k+1)} = \mathbf{b}^{(k)} - 0.4\xi^{(k)}$.

Otherwise, let

$\mathbf{x}^{(k+1)} = \mathbf{x}^{(k)}$ and $\mathbf{b}^{(k+1)} = 0.5\mathbf{b}^{(k)}\left(\mathbf{b}^{(0)} = 0\right)$

Step 4. If $k = M$, stop the total calculation. If $k < M$, let $k = k+1$ and go to Step 2.

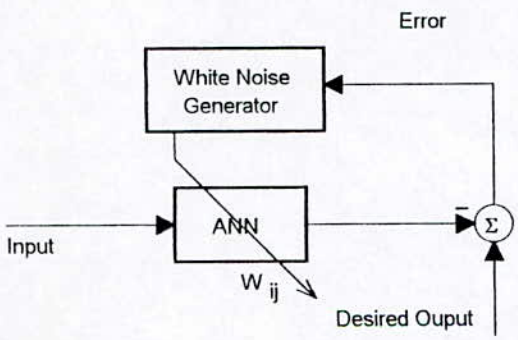


Fig.4 Learning the ANN by ROM

III. SIMULATING HARD NONLINEARITY FUNCTION

A hard function was simulated by an ANN with ten nodes for each layer and training with the two methods described above. Generally, hard nonlinearity function has not saturation such as square function, cubic function, etc. The hard nonlinearity function chosen for this simulation is the square function that is,

$$y = x^2 \tag{4}$$

also the activation function used for every neuron is,

$$F(x) = \tanh(x) \tag{5}$$

A. Learning by BackPropagation

During the training, the gradient step has been varied and the number of iterations and errors were calculated. The error is computed as follow,

$$\varepsilon = \frac{1}{m}\sum_{i=1}^{m} e_i^2 \tag{6}$$

where m is the number of points that equal to 100, e_i is the error on every example. The results obtained are mentioned in Tab.1.

Tab.1 Results obtained by BP

Step of the gradient	NB of Iterations	Error
0.005	10.650	0.00513
0.01	10.000	0.001
0.1	360	0.0016

For the case where the step is close to 1 (0.25 or 0.5) the error decrease until to reach a limit cycle at the 17th iteration. In this case the minimum can not be reached.

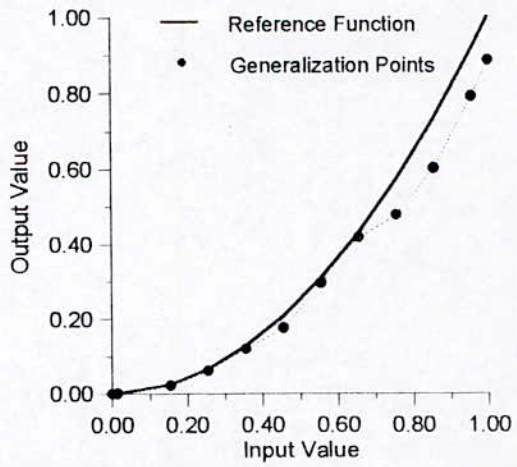


Fig.5 Illustrating the reference curve and the generalization curve for the BP method (*Hard nonlinearity function*)

Also, for high input close to 1, the error began to increase (Fig.5). In fact, when the input is greater than 0.7 the slope of the hard nonlinearity function increase dramatically. In such a case, increasing sampling data is possible to decrease the error.

B. Learning by Random Optimization Method

The same initial conditions are used as in A. In the ROM method the variance υ has been changed and the results below are obtained (Tab.2).

Tab.2 Results obtained by ROM with 100 training points

υ	NB of Iterations	Error $\times 10^2$
0.001	100	2.7
0.01	100	2.19
0.1	100	2.26
1	100	6.35
2	100	18.94

If the number of points is doubled, we obtain the following results (Tab.3).

Tab.3 Results obtained by ROM with 200 training points

υ	NB of Iterations	Error $\times 10^2$
0.001	100	5
0.01	100	5
0.1	100	6
1	100	5

This method (ROM) presents some stagnation during training, the error Vs iterations curve looks likes stairs. Fig.6 illustrate the difference between the reference ($y = x^2$) and the generalization curve of the ANN.

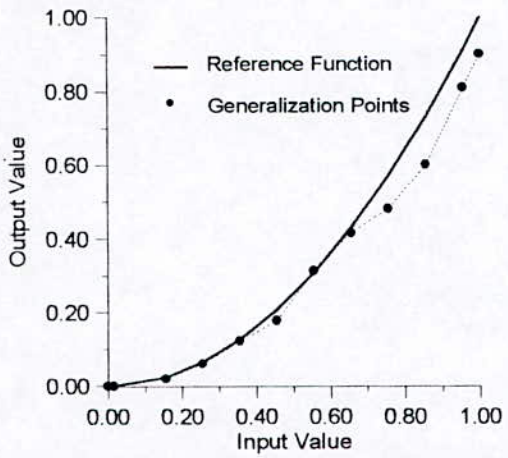


Fig.6 Illustrating the reference curve and the generalization curve for the ROM method (*Hard nonlinearity function*)

IV. Simulating Weak Nonlinearity Function

This type of nonlinear function is weak. The example taken is expressed by,

$$y = 1 - \exp(-x)$$

the working interval is between 0 and 5 that is subdivided into 100 points. All the weights are initialized at 0.3.

A. Learning by BackPropagation

The gradient step is varied between 0.005 and 0.5. The results obtained are mentioned below (Tab.4).

Tab.4 Results obtained by BP

Step of the gradient	NB of Iterations	Error $\times 10^6$
0.005	3000	9
0.01	2000	21.8
0.0.25	135	9
0.5	2	9

With this weak nonlinearity function, the generalization is accurate and shows the capability of the net to generalize points that have not been learned and points outside the learning interval (Fig.7). The generalization points coincide almost with the reference function ($y = 1 - e^{-x}$).

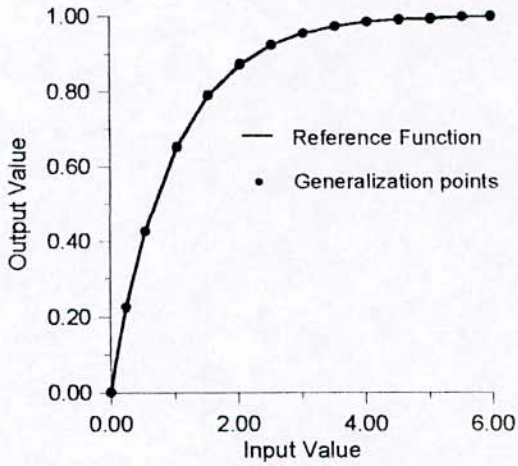


Fig.7 Illustrating the reference curve and the generalization curve for the BP method (*Weak nonlinearity function*)

B. Learning by Random Optimization Method

For the same condition and function as in A, the ROM gives the following results (Tab.5).

Tab.5 Results obtained by ROM with 100 points

υ	NB of Iterations	Error $\times 10^4$
0.001	100	1.4
0.01	100	9.8
0.1	100	0.58
1	–	–
2	–	–

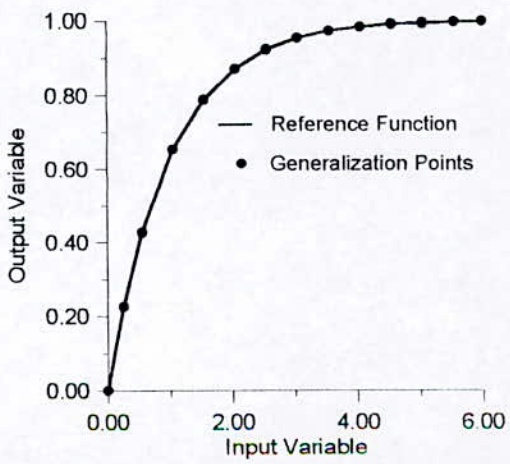


Fig.8 Illustrating the reference curve and the generalization curve for the ROM (*Weak nonlinearity function*)

For These types of nonlinearity functions, ROM is prefered because it provides less iterations than BP for the same final error.

V. SIMULATING HARD NONLINEARITY FUNCTION USING THE COMBINATION OF BP AND ROM

To avoid the problem of initializing the weights in ROM, an idea has risen to our mind to begin the learning by using the BP method for a small number of iteration and then finishing the training with ROM until to reach the final error desired. In this case, convergence speed and accuracy of generalization have increased (Tab.6). Fig.9 shows the difference between the reference and the generalization curve by using firstly the BP (25 iterations) and then learning by ROM with 100 points.

Tab.6 Results obtained by ROM with 100 training points

NB of points trained	NB of iterations	Final error
100	100	0.0045

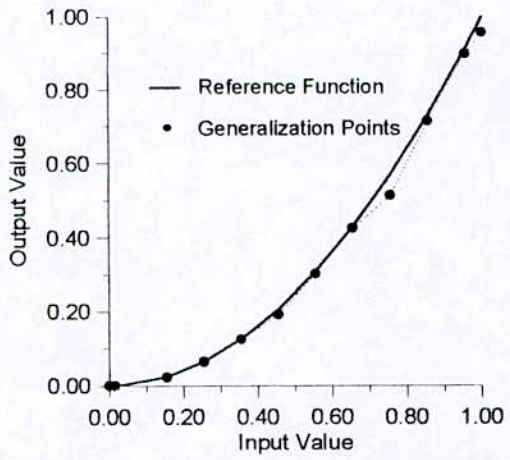


Fig.9 Illustrating the reference curve and the generalization curve for the BP and ROM (*Hard nonlinearity function*)

VI. CONCLUSION

For these types of nonlinearity functions, ROM is prefered because it provides less iterations than BP for the same final error. Acording to the quality of final error and accuracy of generalization, the combination (BP and ROM) constituts the best choice for training the ANN. This one seems to be a powerful tools to simulate such functions and the importance is big in many area of applications. For instance, the look is toward instrumentation and control and the results obtained are dedicated only for this purpose and could not be generalized for all applications and all functions. Often devices used in instrumentation and control are nonlinear such as sensors and actuators. Our group of research is working toward this goal for linearizing transducers [11].

VII. REFERENCES

[1] J.A. Freeman, *Neural Networks: Algorithms, Applications, and Programming Techniques*, Addison-Wesley, Massachusetts, 1992.

[2] M.E. Aggoune, F. Boudjema, A. Bensenouci, A. Hellal, S.V. Vadari, M.R. El Mesai, Design of an adaptive-structure voltage regulator using artificial neural networks, *Proc. of the 2nd IEEE Conference on Control Applications*, Vancouver, Canada, September 13-16, 1993.

[3] L.F. Pau, F.S. Johansen, Neural Network Signal Understanding for Instrumentation, *IEEE Trans on Inst & Meas*, Vol.39, No.4, Aug. 1990, pp.558-564.

[4] B. Widrow, M.A. Lehr, 30 Years of Adaptive Neural Networks: Perceptron, Madaline, and Backpropagation, *Proc of the IEEE*, vol.78, No.9, Sep. 1990, pp.1415-1441.

[5] D. S. Hofer, D. Neumerkel and K. Hunt, Neural Control of a Steel Rolling Mill, *IEEE Control Systems*, June 1993, pp.69-75.

[6] N. Baba, *A new approach for finding the global minimum of error function of neural networks*, Pergamon Press, Vol.2, 1989, pp.367-373.

[7] F.J. Solis, R.J.B. Wets, Minimization by random search techniques, Mathematics of operations research, *The Institute of Management Sciences*, Vol.6, No.1, Feb.1981, pp.19-29.

[8] P. Werbos, *Beyond Regression, New tools for prediction and analysis in the behavioral sciences*, Ph .D thesis, Harvard, Cambridge, MA, Auguste 1974.

[9] D.E. Rumelhart, G.E. Hinton, and R.J. Williams, Learning internal representations by error propagation, in *Parallel Distributed Processing*, Vol.1, ch.8, D.E. Rumelhart and J.L. McClelland, Eds., Cambridge, MA: M.I.T. Press, 1986.

[10] D. Hammerstrom, Working with neural networks, *IEEE Spectrum*, July. 1993, pp.46-53.

[11] M. Attari, "Methods For Linearization of Non Linear Sensors," *Proc.CMMNI-4, Fourth Maghrebin Conference on Numerical Methods of Engineering*, Algiers (Algeria), Vol.1, pp.344-350, Nov.1993.

BIBLIOGRAPHIE

[1] J. A. FREEMAN, M. SKAPURA : NEURAL Networks, algorithms, applications and programming-technics . Addison-Wesley publishing campany 1992 .

[2] DAN HAMMERSTROM : Working with neural networks - IEEE Spectrumjuly 1993 .

[3] D. H. NGUYEN - B. WIDROW : NEURAL networks for self-Learning Control Systems- IEEE control system magazine 1990 PP. 46. 53

[4] B. WIDROW - M. A. LEHR : 30 years of adaptive neural networks perceptron, Madaline, backpropagation proceedings of the IEEE Vol. 78 n°9 sptember 1990 PP. 1415. 1442

[5] M. BINSE : Systèmes connexionnistes. Laboratoire d' informatique de lille. Publication n° I.T. 178 janvier 1990

[6] H. LEMBERG : Les réseaux neuronaux artificiels. Langages et systèmes info. Pc n°71 PP. 20.28

[7] M.L. PERSONNAZ : Etude de réseaux de neurones formels : conception, propriètés et applications. Thèse de doctorat d' état es sciences physiques, université Pierre et Marie Curie - Paris 6. 1986

[8] N. BABA : A new approach for finding the global minimum, of Error function of neural networks . Vol. 2 PP 367 - 373, 1989 Peigamm press plc PP. 367 - 373

[9] F. J. SOLIS. R.J.B. WETS : Minimization by random search techniques. Mathematics of operations research. Vol. 6-n°1. February 1981. The institut of managment sciences. PP 19-29

[10] E. DAVALO, P. NAIM : Des réseaux de neurones. Ed. Eyrolls 1990

[11] BOUDAOUD,BOUKHOBZA : modélisation nonlinéaire et commande neurolinguistique
par régimes glissants de la machine synchrone en monovariable . ENP juin 1993 .Automatique
projet de fin d' études.

[12] E.B. BAUM, D. HASSLER : What size net gives valid généralization NEURAL camputation. PP 151. 160, california institut of technologie. 1989

[13] N. MICHEL, J. FARELL : Associative memories via artificial neural networks. IEEE control systems magazine PP 6.17 (1990)

[14] SHUN - ICH, AMARI : Mathematical foundations of neurocamputing. Proceedings of the iEEE, vol. & N°9, september 1990. PP. 1443-1463

[15] J.P. WERBOS : Back propagation through time : What it does and How to do it. Proceedings of IEEE, vol. 78 n°10 october (1990)

[16] MADANI - OUSSAR : Etude de la commande Neuro - linguistique, application SISO aux systèmes non linéaires et à un système MIMO : bras de Robot. ENP. Juin 1993 (Automatique) . Projet de fin d' études.

[17] G. ASCH :Capteurs en instrumentation industielle.ed.Dunod.

[18] P.A. PARATTE, P. Robert : Systèmes de mesures. Traité d' éléctricité presses polytechniques Romandes.

[19] M. ATTARI : " Méthods For Linearization of non linear Sensors" proc. CMMN1-4, Fourth Maghrebin Conference On numerical méthods of engineering, Algiers (Algeria), Vol.1 PP 344-350, Nov 1993

[20] D. SBARBARO - MOFER : Neural control of a steel Rolling Mill, IEEE, control systems, june 1993, PP 69-75

[21] R. NAIDU, T.J. MCAVOY : Use of neural networks for sensor Failure detection in a control system. IEEE control systems magazine April 1990, PP 94-55.

[22] L.F. PAU : Neurol networks signal understanding for instrumentation IEEE Trans on Inst & Meas, Vol. 39, NO.4, Aug . 1990, PP. 558-564.

[23] C. MACCA : Determination of selectivity coefficient of ion. selective electrodes by means of linearizad multiple standar addition techniques. Elsevier science publishers B.V 1983. PP. 51-60

[24] G. VLASOV, B.L. SLZNEV : Silver ion sensors based on Ag-As-Se-Te glasses. sensors and actuators B,2 (1990) PP 23-31 Elsevier sequoia

[25] A.K.CONVINGTON : Introduction: Basic electrode types, classification and selectivity considérations. Ion. selective methodology Vol I chapitre 1 CRC press 1979. PP. 1. 20.

[26] T. SKOLASKI, A. HULANICKI : The stady of processes influencing apparent selectivity for solid stade ion. selective electrodes. Pergammon

Press, Oxford. 5th symposium on ion selective electrodes Matrafured, 1988. PP. 411-424.

[27] NANMAC Corporation : Temperature measurment handbook, Vol. VII Thermocouples, RTDS and accessories

[28] M. BENCHEIKH, M.O. AIT. ABDELAZIZ : Etude et réalisation d' une station météorologique, projet de fin d' étude . session juin 1993. Département controle, institut d' éléctronique USTHB

[29] M. HASLER : Circuits non linéaires. Press polytechnique Romandes. Lausanne 1985.

[30] M.KUNT : Traitement numérique des signaux .Traité d'Eléctricité,d'Eléctronique et d'eléctrotechnique.ed Dunod 1981.

FAIT PAR COPIE CLAIR DE LUNE. Cité 489 logts Bt 16 local N°10
Bab-Ezzouar (El-Djorf).