République Algérienne Démocratique et Populaire

MINISTERE DE L'ENSEIGNEMENT SUPERIEUR

ET DE LA RECHERCHE SCIENTIFIQUE

UNIVERSITE DE SKIKDA

FACULTE DES SCIENCES ET SCIENCES DE L'INGENIEUR

INSTITUT D'ELECTROTECHNIQUE

MEMOIRE DE FIN D'ETUDE EN VUE DE L'OBTENTION DU DIPLOME

D'INGENIEUR D'ETAT EN AUTOMATIQUE

Thème

DIAGNOSTIC NEURO-FLOU : APPLICATION A LA MACHINE ASYNCHRONE

Dirigé par :

Dr.M.L BENLOUCIF

Réalisé par :

LARABI Mohand Saïd
ROUIDI Bachir

Promotion Juin 2005

Sommaire

A -INTRODUCTION GÉNÉRALE.....4

B -GÉNÉRALITÉS SUR LE DIAGNOSTIC.....7

I -Introduction.....8

II -Attributs de la sûreté de fonctionnement.....8

III -Entraves à la sûreté de fonctionnement.....8

IV -Détection et isolation de défauts (Système de surveillance).....11

V -Méthodes analytiques pour la détection et l'isolation de défaillances.....15

VI -Conclusion.....16

C -DIAGNOSTIC NEURO-FLOU.....17

I -Introduction.....18

II -Génération des résidus par des réseaux de neurones.....18

III -Analyse qualitative des résidus.....21

IV -Conclusion.....33

D -IDENTIFICATION DES SYSTÈMES PAR RÉSEAUX DE NEURONES.....34

I -Introduction.....35

II -Identification.....36

III -Identification par réseaux de neurones.....42

IV -Conclusion.....50

E -MODÉLISATION DE LA MAS.....51

I -Introduction.....52

II -Description du moteur asynchrone.....52

III -Modélisation du moteur asynchrone.....53

IV -Commande de la machine asynchrone.....62

V -Types de défauts de la machine asynchrone.....65

VI -conclusion.....65

F -ÉTUDE EN SIMULATION.....66

I -Introduction.....67

II -Présentation de la MAS.....67

III -Fonctionnement normal de la MAS (sans défauts).....68

IV -Identification par réseau de neurones.....69

V -Détection et localisation des défauts capteurs.....74

VI -Diagnostique des défauts physiques.....87

VII -Conclusion.....91

G -CONCLUSION.....92

H -ANNEXE - LES RÉSEAUX DE NEURONES.....94

I -Introduction (Historique).....95

II -Fondements biologiques.....96

III -Le neurone artificiel.....97

IV -Les connexions.....101

V -Architectures de réseaux de neurones.....101

VI -Modèles de réseaux de neurones.....102

VII -Apprentissage dans les réseaux de neurones.....104

VIII -Conclusion.....108

I -BIBLIOGRAPHIE.....109

J -DIVERS.....111

A - Introduction générale

De nos jours, le plus grand souci de l'industrie est le rendement (gain en productivité et en qualité de production). Mais quand on dit ''rendement'' on dit sûreté, bon fonctionnement et sécurité des dispositifs. Cette dernière condition est la principale préoccupation des industriels, car elle ne garantit pas seulement la protection des équipements industriels mais aussi, celle des personnes et de l'environnement.

Et c'est ainsi, que plusieurs méthodes de diagnostic, de détection et de localisation de défauts ont été développées pour assurer la surveillance des systèmes et garantir leur tolérance aux défauts (connues sous le label anglais FDIA : Fault Detection Isolation and Accommodation).

On peut classer ces méthodes de diagnostic en deux grandes catégories :

1. <u>Les méthodes quantitatives</u> : qui supposent la connaissance du procédé sous la forme d'un modèle analytique. Elles sont basées sur les techniques d'estimation d'état et d'estimation paramétrique (Observateurs, Filtres de KALMAN, Algorithmes d'identification…).

2. <u>Les méthodes qualitatives</u> : qui sont basées sur des connaissances heuristiques sur le système et son environnement ainsi que sur l'expertise humaine (Système expert, Reconnaissance de forme, Réseaux de neurones et Logique Floue…).

Grâce aux résultats théoriques et pratiques obtenus au cours des dernières années, les réseaux de neurones sont devenus un outil de plus en plus utilisé dans divers domaines (Industrie, Banque, Service…). Ils demeurent un sujet de grand intérêt pour les chercheurs, qui désirent améliorer les performances de ces réseaux et étendre leur champ d'application.

Ainsi, nous avons opté pour un schéma neuro-flou, pour bénéficier des qualités des méthodes neuronales et celles basées sur la logique floue. De ce fait, on tirera avantage des capacités d'apprentissage des premières et la lisibilité et la souplesse des éléments manipulés par les secondes.

Notre travail d'étude porte donc, sur l'utilisation :

1. des réseaux de neurones pour la génération des résidus, en tirant avantage de leur capacité d'excellents approximateurs de fonctions non linéaires (Etape Détection).

2. de la logique floue pour l'analyse de ces résidus car elle permet de traiter des données imprécises et assure une souplesse dans la décision en approchant le raisonnement humain (Etape Evaluation).

Notre choix d'étude a porté sur la machine asynchrone (MAS), sachant qu'elle représente l'entraînement électrique le plus utilisé dans l'industrie, pour sa simplicité et sa souplesse, et qui constitue donc un maillon important dont il faut assurer la fiabilité.

Dans ce mémoire nous nous intéressons à l'application d'un schéma de diagnostic neuro-flou pour la surveillance des défauts capteur et des défauts physiques électriques pouvant affecter une machine asynchrone.

Le présent mémoire comprend cinq chapitres :

Le chapitre I est consacré à la définition et à la classification des défauts ainsi qu'aux généralités sur le diagnostic et la détection des défauts.

Le chapitre II comprend une présentation des techniques de détection et de localisation des défauts basées sur les réseaux de neurones et la logique floue.

Le chapitre III expose les différentes méthodes d'identification des systèmes ainsi que les méthodes d'identification par les réseaux de neurones.

Le chapitre IV est dédié à la description et la modélisation de la MAS.

Le chapitre V présente une étude en simulation de l'approche de diagnostic neuro-floue appliquée à la MAS

B - Généralités sur le diagnostic

I - INTRODUCTION

De nos jours, vu l'importance de la sécurité et de la fiabilité des processus automatisés, le domaine du diagnostic industriel a connu un grand essor. De nombreuses méthodes de diagnostic ont ainsi été développées, parmi lesquelles on distingue les méthodes basées sur l'approche quantitative et l'approche qualitative. Dans ce chapitre on présente les concepts de base de la sûreté de fonctionnement et de la surveillance des systèmes dynamiques, ainsi que la présentation de l'approche analytique de détection et localisation des défauts.

II - ATTRIBUTS DE LA SÛRETÉ DE FONCTIONNEMENT

- **Disponibilité**: La disponibilité est l'aptitude d'un système à être en état d'accomplir la tâche ou la fonction pour laquelle il a été choisi, dans des conditions données et à un instant donné, c'est à dire, la capacité instantanée à rendre le service.
- **Fiabilité**: continuité du service, ou l'aptitude d'un système à accomplir une fonction requise, dans des conditions données, pendant un intervalle de temps déterminé.
- **L'efficacité** : est la capacité d'un système à réaliser correctement la tâche ou la fonction pour laquelle il a été choisi, lorsque les conditions de fonctionnement sont nominales.
- **Securité-innocuité (safety)** : non occurrence de défaillances à caractère catastrophique.
- **Maintenabilité**: facilité de retour à un état sans erreur après défaillance.
- **Securité-confidentialité (security)**: non occurrence de défaillances causées par des fautes intentionnelles.

Ces qualités doivent être établies à l'origine de l'installation, mais aussi, être maintenues dans le temps.

III - ENTRAVES À LA SÛRETÉ DE FONCTIONNEMENT

III.1 - Faute

Action volontaire ou non, dont le résultat est la non prise en compte correcte d'une directive, d'une contrainte exprimée par le cahier des charges, et c'est la cause adjugée ou supposée d'une erreur.

III.1.1 - Origine

- **Cause phénoménologique :**
  - Fautes physiques.
  - Fautes humaines.

- **Frontière du système :**
  - Fautes internes : se sont les parties de l'état structurel susceptibles de produire une erreur.
  - Fautes externes : qui résultent de l'interférence ou des interactions du système avec son environnement.

- **Phase de création :**
  - Fautes de conception : qui résultent d'imperfections commises soit au cours du développement du système, ou en cours de modification ultérieure.
  - Fautes opérationnelles : qui surviennent durant l'exploitation du système.

III.1.2 - La persistance temporelle

- Fautes permanentes : dont la présence n'est pas reliée à des conditions ponctuelles internes (processus de calcul) ou externes (environnement).
- Fautes temporaires : se présentent pour une durée limitée.

III.1.3 - Nature des fautes

- Fautes accidentelles : sans volonté de nuire.
- Fautes intentionnelles : avec volonté de nuire.

FAUTES				
Nature	**Origine**			**Persistance temporelle**
1. Accidentelle	**Cause phénoménologique**	**Frontière du système**	**Phase de création**	1. Temporaire **2.** Permanente
2. Intentionnelle	1. Physique **2.** Humaine	1. Externe **2.** Interne	1. Conception **2.** Opérationnelle	

Figure1 : **caractéristiques des fautes**

III.2 - Erreur

Partie du système ne correspondant pas, ou correspondant incomplètement au cahier des charges. En toute logique, une erreur est la conséquence d'une faute.

III.3 - Défaillance

Perte partielle ou totale des fonctionnalités du système, qui le rend incapable de délivrer le service pour lequel il a été conçu.

III.4 - Panne

Etat d'un système incapable d'assurer le service spécifié à la suite d'une défaillance, et on distingue les classes de pannes suivantes :

- **Panne de capteur :** dégradation d'une caractéristique des blocs de capteurs (biais, dérive,) d'où la transmission des mesures erronées.
- **Panne de processus :** détérioration des paramètres (fuites, rupture des organes).
- **Panne d'actionneur :** fourniture de valeurs de commande erronées, et possibilité d'avoir une rupture ou un blocage d'actionneur.

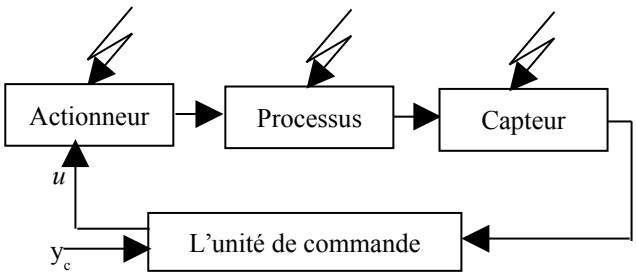


***Figure2*:** Types de pannes

Aussi, une panne peut être :

- **Panne intermittente :** panne d'un dispositif subsistant sur une durée déterminée et limitée. Après cette durée le dispositif est apte à assurer la fonction ou la mission pour lequel il a été conçu, sans avoir fait l'objet d'une action corrective. En pratique ce sont les défauts les plus difficiles à diagnostiquer.
- **Panne fugitive :** panne d'un dispositif qui est intermittente et difficilement observable. Elles sont extrêmement difficiles à diagnostiquer car leur apparition est de nature aléatoire.
- **Panne permanente :** panne d'un dispositif qui subsiste tant qu'une opération de maintenance corrective n'a pas été effectuée.
- **Panne latente ou cachée :** panne d'un dispositif qui existe mais qui n'a pas été détectée.

IV - DÉTECTION ET ISOLATION DE DÉFAUTS (SYSTÈME DE SURVEILLANCE)

De nombreux dysfonctionnements peuvent altérer la sécurité, la fiabilité et la disponibilité d'un système. C'est pourquoi le but de la surveillance en ligne (détection, localisation et diagnostic des défaillances) est de déterminer, à partir des informations prélevées en temps réel, si le système fonctionne correctement ou présente des anomalies, appelées défaillances, susceptibles de compromettre la réalisation de sa mission.

IV.1 - Les fonctions de la surveillance

Un système de surveillance répond généralement à trois fonctions qui sont :

IV.1.1 - La détection

Caractérise le fonctionnement du système de normal ou d'anormal.

IV.1.2 - Le Diagnostic

Etablit un lien de cause à effet, entre un symptôme observé et la défaillance qui est survenue, ses causes et ses conséquences. On distingue classiquement trois étapes :

1. **Localisation** : détermine le sous système fonctionnel à l'origine de l'anomalie et progressivement affine cette détermination pour désigner l'organe ou le dispositif élémentaire défectueux.
2. **Identification** : détermine les causes qui ont engendré la défaillance constatée.
3.
4. **Explication** : justifie les conclusions du diagnostic.

IV.1.3 - La reprise

Englobe la fonction de recouvrement et de décision, celle ci détermine un état accessible pour le retour au nouveau fonctionnement normal et les différentes actions correctives modifiant la configuration du procédé et de la commande.

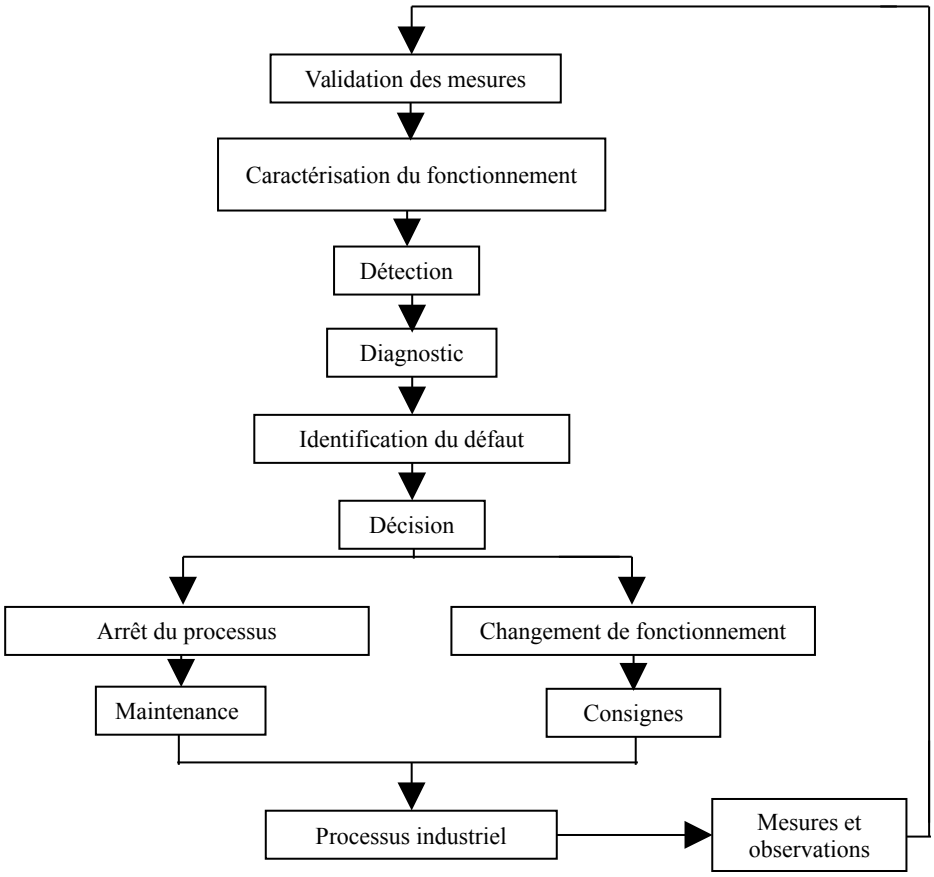


Figure 3 : Les différentes phases du diagnostic

IV.2 - Approches de la surveillance

Pour répondre à la démarche de diagnostic, ils existent de nombreuses approches. Généralement on en distingue quatre principales:

IV.2.1 - Les méthodes par redondances matérielles

Sont celles mises en oeuvre sans avoir une connaissance du modèle du système surveillé (ou en ayant une connaissance partielle). Elles sont relativement simples à mettre en oeuvre et s'appuient plus sur des aspects matériels que sur des aspects analytiques. Les inconvénients de ces méthodes sont qu'elles s'accompagnent d'un accroissement du coût des installations, d'une augmentation de poids ainsi que d'une diminution du temps moyen de bon fonctionnement.

IV.2.2 - Les méthodes par systèmes experts

Semblent particulièrement adaptées lorsque les connaissances sur le système sont de type heuristiques. C'est-à-dire lorsque les informations utilisées (qualitatives ou quantitatives) permettent l'utilisation de règles imbriquées, issues de la connaissance et de l'expérience de l'expert, comme par exemple des liens de cause à effet. Le principe de base d'un système expert s'articule autour de deux éléments principaux, une base de connaissance et un moteur d'inférence. Cette approche est particulièrement séduisante pour le diagnostic, car elle permet de manipuler un grand nombre de données non homogènes et indépendantes du contexte tout en rendant compte du raisonnement suivi.

IV.2.3 - Les méthodes par reconnaissance de formes

Sont des approches intermédiaires entre les techniques fondées sur l'utilisation d'un système expert et les techniques analytiques. Elles reposent sur la détermination d'un certain nombre de classes au moyen d'une procédure d'apprentissage. A chacune de ces classes est associé un mode de fonctionnement (fonctionnement normal, défaillance $n°i$). Chaque donnée prélevée sur le système est ensuite affectée à l'une de ces classes ; ce qui permet de déterminer le mode de fonctionnement du système. Ce type de méthode est utilisé dans le cas des systèmes dont aucun modèle comportemental ne peut être obtenu. Le diagnostic médical en est un exemple. A partir d'une connaissance acquise au fil du temps, une correspondance a été établie entre l'observation d'un certain nombre de caractéristiques du corps humain (les symptômes) et le type de maladie. Cette correspondance définit le modèle en terme de mode de fonctionnement. En fonction des symptômes observés, la maladie peut être diagnostiquée.

IV.2.4 - Les méthodes analytiques (à base de modèles analytiques)

Les conceptions basées sur la redondance analytique sont conçues autour d'un modèle dynamique représentant le système le mieux possible. Elles utilisent principalement les techniques de traitement du signal (filtrage, identification, estimation et statistique).

Les fonctions de détection et de localisation des fautes basées sur l'approche analytique sont généralement réalisées en deux étapes :

a) Génération de résidus

Les entrées et les sorties mesurées du système sont traitées pour générer des signaux, indicateurs de défauts appelés résidus qui restent identiquement nuls lors du fonctionnement normal, et qui réagissent aux fautes auxquelles ils sont sensibles.

b) Logique de décision

La vraisemblance des résidus relativement aux éventuelles fautes est calculée. Souvent un résidu scalaire suffit pour détecter la présence d'une anomalie dans le système. Cependant pour la localiser, un vecteur de résidus sensibles aux différentes fautes est généralement nécessaire.

Les tests effectués pour la décision peuvent se traduire en un simple filtrage passe-bas, suivi d'un contrôle de dépassements de seuils prédéfinis ou bien en l'utilisation de méthodes statistiques plus sophistiquées.

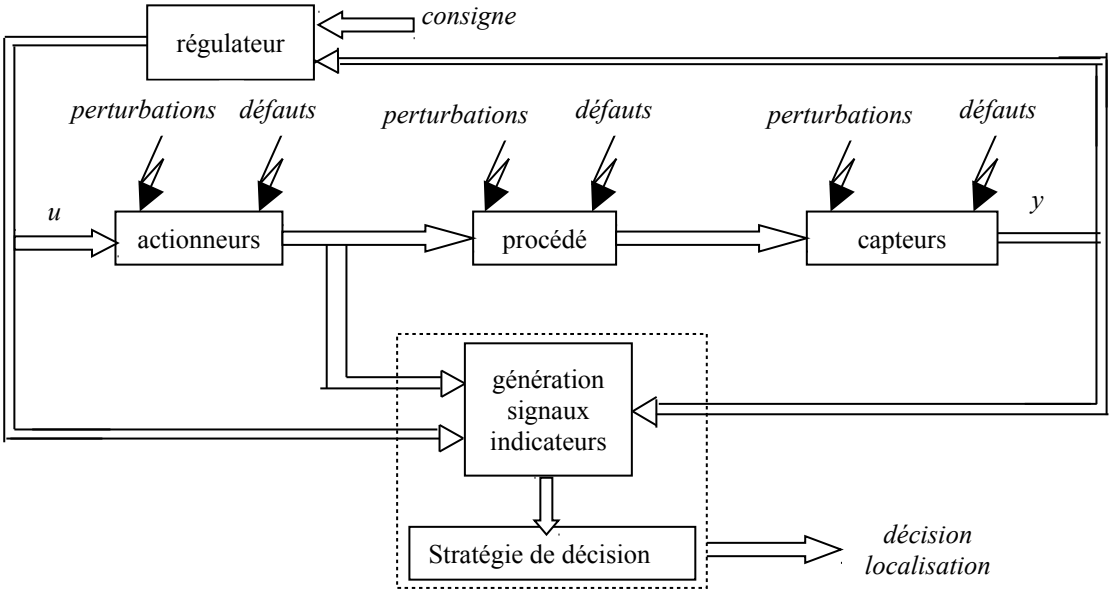


Figure4: Principe général du diagnostic à base de modèles analytiques.

Durant ces dernières années, quatre grandes approches de génération de résidus se sont développées :

- **L'identification paramétrique :** c'est l'estimation d'un vecteur de paramètres dont la variation à l'extérieur d'une plage de référence est significative à l'apparition d'un défaut.
- **L'espace de parité :** l'idée est de générer le vecteur de résidus, en projetant les données mesurées dans un espace, tel que le résidu possède la propriété de nullité en absence de défaillance.
- **L'estimation d'état :** le résidu est défini comme la différence (filtrée) entre les sorties

mesurées et les sorties estimées.

- **La synthèse directe :** qui s'est développée plus récemment avec l'introduction de la modélisation sous forme standard et les outils de synthèse H_∞ .

V - MÉTHODES ANALYTIQUES POUR LA DÉTECTION ET L'ISOLATION DE DÉFAILLANCES

Le but du diagnostic est de résoudre deux tâches de décision importante (la détection et l'isolation). Plusieurs méthodes analytiques ont vu le jour ces dernières années pour la détection et l'isolation de défaillances (connues sous le nom de méthodes FDI). Le principe général de ces méthodes peut être décomposé en deux étapes principales : la détermination des relations de surveillance et la décision.

V.1 - Etape de détermination des relations de surveillance

Elle fait intervenir directement ou indirectement par une étape d'estimation intermédiaire, les grandeurs mesurées qui sont connues (sorties et entrées de commande) du processus. Ces relations doivent donner lieu à des résidus assez petits en fonctionnement normal, et un ou plusieurs d'entre eux doivent être amplifiés en présence d'une anomalie ou défaillance.

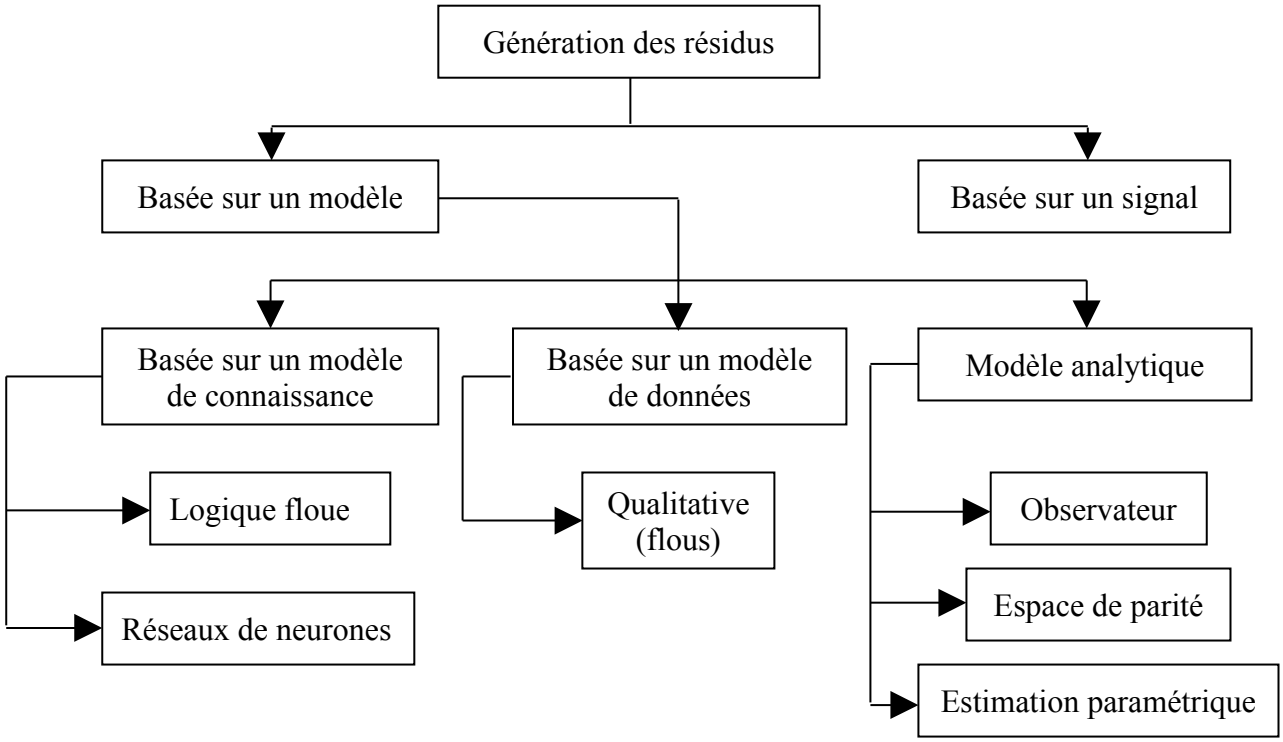


Figure 5 : Méthodes de génération des résidus

Comme le montre la figure (5), plusieurs méthodes existent pour la génération de résidus. Ces méthodes peuvent être subdivisées en deux approches principales :

- **Approche quantitative** : Observateur, filtre de Kalman, espace de parité, estimation paramétrique ...
- **Approche qualitative** : Logique floue, réseaux de neurones, reconnaissance de forme, système expert …

Dans ce mémoire, nous nous sommes intéressés plus particulièrement à la deuxième approche.

V.2 - Etape de décision

Elle consiste à analyser les résidus pour détecter la défaillance et éventuellement isoler l'élément défaillant.

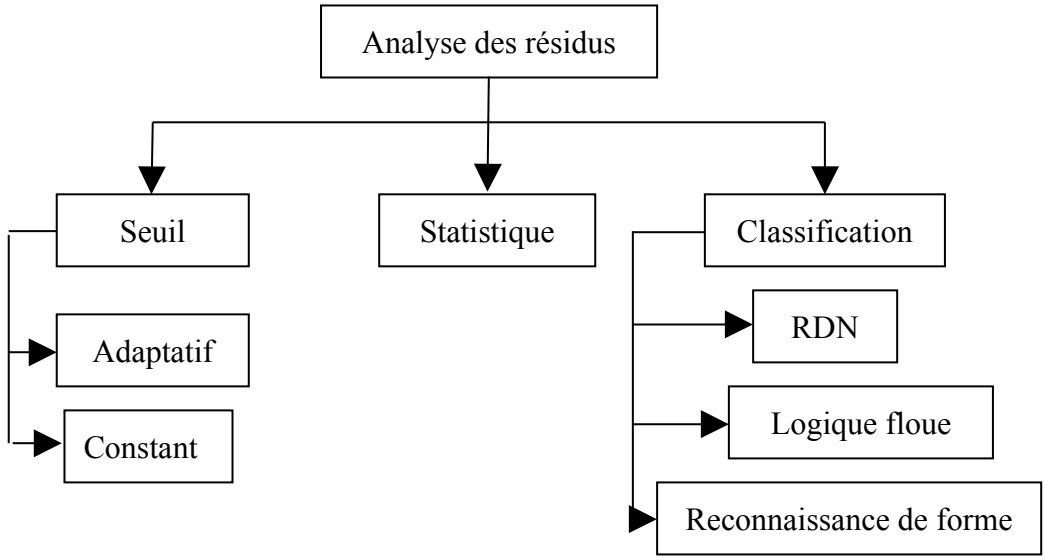


Figure 6 **:** Méthodes d'analyse des résidus

Comme pour la génération de résidus, plusieurs méthodes permettant la satisfaction de cette tâche existent (figure(6)).

VI - CONCLUSION

A l'issue de ce chapitre on peut conclure que le problème du diagnostic consiste à :

- Générer des résidus qui doivent être nuls en fonctionnement normal, et assez sensibles à tout défaut affectant le système à surveiller d'une part ;
- Et d'autre part à analyser ces résidus pour détecter la présence d'un défaut, et localiser l'élément défaillant.

C - Diagnostic Neuro-Flou

I - INTRODUCTION

Dans ce chapitre nous allons passer en revue quelques approches de diagnostic de type non-analytique, et ceci afin de mieux situer les méthodes de diagnostic neuro-flou qui seront présentées dans la suite. Nous allons étudier plus particulièrement les méthodes utilisant les réseaux de neurones et la logique floue, pour l'analyse des résidus et on insistera sur les méthodes FDI basées sur les techniques neuro-floues.

II - GÉNÉRATION DES RÉSIDUS PAR DES RÉSEAUX DE NEURONES

Dans le cas des systèmes non linéaire, l'obtention d'un générateur de résidu par les méthodes quantitatives classiques n'est pas une chose aisée. Il serait alors plus judicieux d'utiliser des réseaux de neurones pour générer les fonctions résidus, en utilisant leur capacité de modéliser des fonctions non linéaires.

La procédure généralement utilisée est représentée par la figure (1)

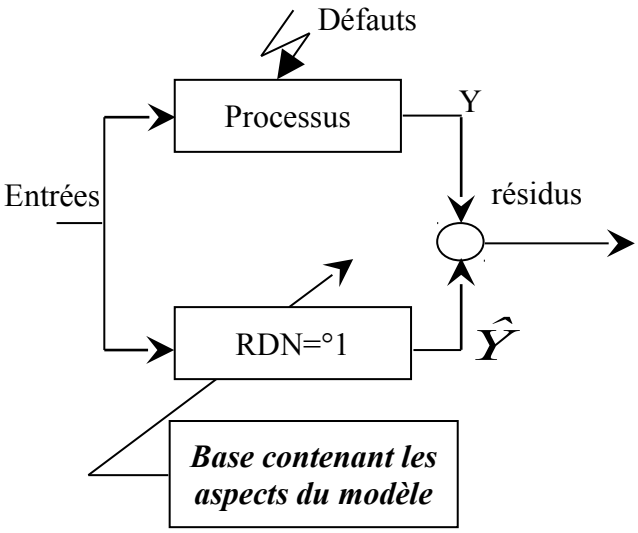


Figure 1 : Schéma de génération de résidus à base de réseau de neurone

Pour la génération de résidus plusieurs étapes sont nécessaires.

II.1 - Création de la base de données

Au préalable, une base de données doit être réalisée en hors ligne par une connaissance experte. Elle doit comporter les principales caractéristiques du processus (point de fonctionnement, stabilité, bruit…).

Une fois cette base réalisée, une structure du réseau de neurone doit être choisie.

II.2 - Choix de la structure du réseau de neurones

Généralement on opte pour la structure NNARX, qui constitue le meilleur choix pour la structure des modèles non linéaires si le système est déterministe ou peu bruité. Ainsi, on évite le problème de stabilité des autres structures comme la NNARMAX, par exemple.

Un réseau de neurones à deux couches (une couche cachée et une couche de sortie) est largement suffisant pour l'identification d'un système dynamique non linéaire multi variable (plusieurs entrées, plusieurs sorties).

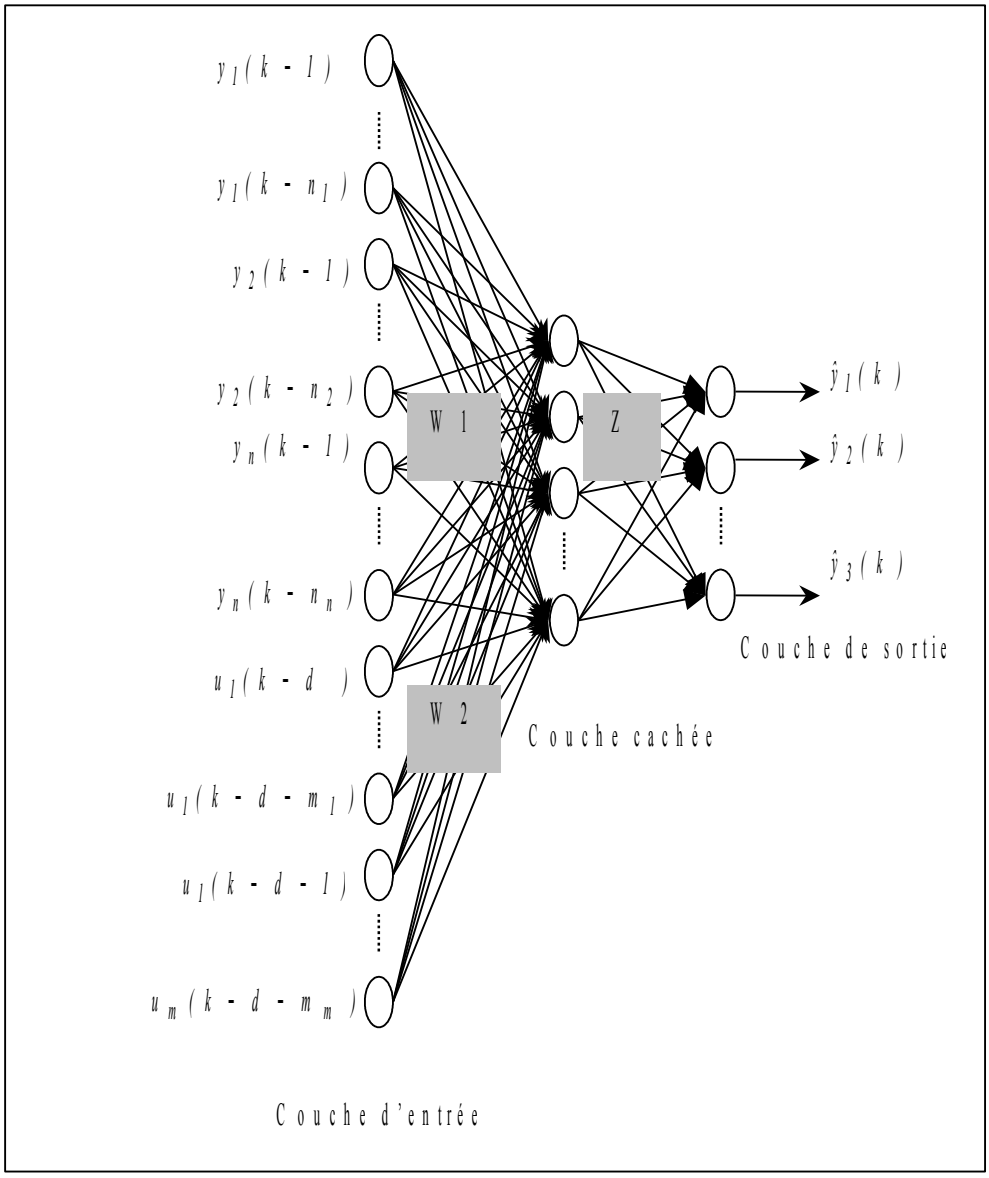


Figure 2 : Architecture d'un réseau à 2 couches

Les entrées du réseau sont :

- les sorties du système décalées $y_1, y_2, .., y_n$ où n est le nombre de sorties et n1...nn détermine l'ordre de chaque sortie,
- et les entrées du système u_1, u_2, …, u_m, .où m est le nombre d'entrées du système, d est le décalage de l'entrée (généralement choisi égal à 1) et m1… mm les ordres de chaque entrée.

La couche cachée est constituée par plusieurs neurones, dont le nombre est choisi de façon expérimentale. Les fonctions généralement choisies pour ces neurones sont de type sigmoïde.
Les sorties du réseau sont les sorties estimées du système. Les fonctions d'activation choisies pour les sorties sont de type linéaire.
Les sorties estimées y peuvent être écrites sous la forme suivante :

$$\hat{y}_1 = \psi_i \cdot (Z \cdot h + b) \qquad i=1,..n \qquad (1)$$

Avec

$$h = \phi\,(W1_{11} y_1(k-1) + \ldots + W1_{1n} y_1(k-n1) + \ldots + W1_{n1} y_1(k-1) + \ldots W1_{nn} y_n(k-nn) + W2_{11} u_1(k-d) + \ldots + W2_{1m} u_m(k-m1-d) + \ldots + W2_{1m} u_1(k-d) + \ldots + W2_{mm} u_m(k-mm-d)$$

Où :

- W1 et W2 sont les poids reliant la couche d'entrée à la couche cachée et Z les poids reliant la couche cachée à la couche de sortie.
- **ψ** est une fonction d'activation de type sigmoïde et **φ** une fonction d'activation de type linéaire.
- b est le biais

II.3 - Apprentissage

Les poids W et Z ainsi que les biais sont initialement choisis de façon aléatoire, puis adaptés par un algorithme d'apprentissage, voir (annexe**)** de façon à minimiser l'erreur quadratique.

$$E = \frac{1}{2}(\|y - \hat{y}\|)^2 \qquad (2)$$

L'algorithme généralement utilisé est l'algorithme de Levenberg- Marquardt.

II.4 - Validation du réseau de neurones

Une fois le réseau entraîné, on obtient les valeurs finales des poids et des biais. Une étape d'évaluation est nécessaire pour voir si le réseau obéit aux exigences fixées. Pour cela, on réalise plusieurs tests sur le réseau. Si, malheureusement le réseau n'est pas satisfaisant, on doit songer soit à modifier la structure du réseau (augmenter les ordres des sorties ou entrées, ajouter des neurones dans la couche cachée), ou augmenter le nombre d'itérations de la phase d'apprentissage si les paramètres du réseau n'ont pas encore convergé suffisamment, ou encore, modifier les valeurs initiales des poids et des biais.

III - ANALYSE QUALITATIVE DES RÉSIDUS

III.1 - Analyse par la logique floue

L'utilisation la plus fréquente de la logique floue dans les méthodes FDI est dans l'évaluation des résidus. Il y a trois principales approches dans le processus de décision : le seuil adaptatif flou, la classification floue et le raisonnement flou.

III.1.1 - Seuil adaptatif flou

Pour contourner les inconvénients d'un placement de seuils statique sur des mesures variables ou des résidus, un système basé sur le flou est utilisé pour obtenir des seuils adaptatifs, qui varient selon les conditions opératoires du processus.

Cette méthode est très utile quand un modèle linéaire est utilisé pour la génération de résidu, alors que le processus réel est non linéaire, elle diminue le taux de fausses alarmes dues aux non¨linéarités. Pour cela on utilise un intervalle (seuil flou) dont la borne supérieure correspond à la plus grande valeur du résidu, et la borne inférieure correspond au bruits (perturbations ou non linéaire), voir figure (3).

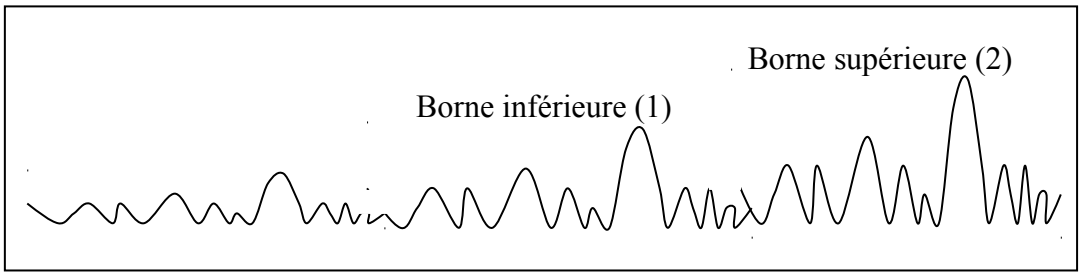


Figure 3 : Seuil flou

La borne (1) représente une perturbation et la borne (2) indique un défaut. Dans une détection classique (seuil fixe), une augmentation du résidu va générer une fausse alarme (la borne (1) dépassera le seuil), mais dans une considération floue (seuil flou ou adaptatif) l'effet d'une augmentation du résidu va se répercuter sur la largeur de l'intervalle et va causer seulement un changement du taux de fausses alarmes.

III.1.2 - Classification floue

L'idée de la classification floue vient directement de l'approche reconnaissance de forme. Un apprentissage utilisant une base de données est effectué en hors ligne, pour déterminer les centres de classes les plus significatifs et ceci pour chaque défaut étudié.

Ensuite en ligne, on détermine le degré d'appartenance de la donnée courante au centre de classe prédéfini (défaut prédéterminé), pour décider de l'occurrence du défaut et son origine. Cette méthode est très utile quand on a plusieurs résidus ou dans le cas ou aucune connaissance experte sur le système n'est disponible.

III.1.3 - Raisonnement flou

Le principal avantage du raisonnement flou, est qu'il peut principalement introduire des informations heuristiques dans le schéma d'analyse. Les schémas des raisonnements flous sont aussi faciles à comprendre, car ils sont très similaires à la manière avec laquelle les humains résolvent des problèmes d'analyse.

L'idée de base de l'utilisation de la déduction floue pour l'analyse des résidus,vient du fait que chaque résidu est zéro, positif ou négatif par rapport à un certain degré. Pour l'élaboration d'un schéma flou, on utilise généralement les étapes suivantes :

- **La fuzzification** : C'est la transformation des valeurs données brutes en des valeurs d'entrée floues. Pour cela, on détermine pour chaque entrée et sortie sa fonction d'appartenance floue.

- **La Déduction (inférence)** : On détermine la base des règles selon le modèle suivant :

  Règle : si $r_i = A$ alors $f_k = B$

  Action : $r_i = A$

  Conclusion : $f_k = B$

Les règles sont formées pour déterminer les conditions sous lesquelles le défaut existe et sous lesquelles le système est non défaillant.

Par exemple **:**

- Si le résidu 1 est >0 et le résidu 2 <0, alors le défaut 1 est présent.
- Si le résidu 1=0 et le résidu 2=0, alors le système est non défaillant.

Si les règles ne reflètent pas l'expérience d'un opérateur, alors elles peuvent être difficilement validées.

- **Defuzzification** : Construction de valeurs de sorties brutes à partir des ensembles de déduction (inférence). La sortie du processus de la décision logique, est une valeur qui donne le degré avec lequel un défaut est présent dans le système, plutôt qu'un constat simple de défaut/non défaut. Le degré peut être une indication aussi bien de la taille du défaut présent, que de la certitude avec laquelle un défaut est présent dans le système. Une telle sortie est donnée pour chaque défaut considéré.

Un des inconvénients majeurs, pour réaliser des schémas FDI utilisant le raisonnement flou, est l'absence de méthodes formelles de conception.

III.2 - Analyse basée sur les réseaux de neurones

III.2.1 - Classification linéaire

Si nous avons des classes qui sont linéairement indépendantes, un Perceptron (annexe) à une seule couche est largement suffisant pour réaliser cette tâche.

- **Utilisation d'un Perceptron** : Dans ce cas, la cellule de décision avec fonction de seuil se comporte comme un adaptateur linéaire, qui réalise la séparation des exemples en deux classes CO_1 et CO_2 à l'aide de la fonction discriminateur linéaire :

$$g(x) = W^T x \quad , \tag{3}$$

Avec :

$W = (W_{1j}, W_{2j}, ..., W_{nj}, W_{0j})^T$, vecteur des poids synaptiques

$x = (x_1, x_2, ..., x_n, 1)^T$

Cette fonction permet la séparation de l'ensemble les vecteurs x en deux classes CO_1 et CO_2 :

$$x \in CO_1 \text{ si } g(x) \geq 0$$

$$x \in CO_2 \text{ si } g(x) < 0$$

La figure (4) montre l'interprétation géométrique du mécanisme de classification.

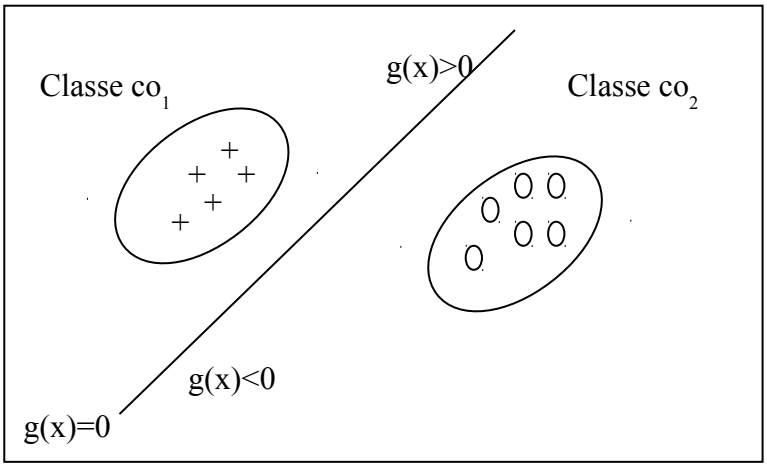


Figure 4 : Séparation de deux classes avec adaptateur linéaire

Un Perceptron linéaire à seuil à n entrées divise l'espace des entrées R en deux sous-espaces délimités par un hyperplan. Réciproquement, tout ensemble linéairement séparable peut être discriminé par un Perceptron.

Il suffit pour s'en convaincre de se rappeler que l'équation d'un hyperplan dans un espace de dimension n est de la forme :

$$\alpha_1 x_1 + \ldots + \alpha_1 x_n = \beta \tag{4}$$

Un Perceptron est donc un discriminant linéaire. On montre facilement qu'un échantillon de R^n est séparable par un hyperplan si, et seulement si l'échantillon de R^{n+1} obtenu en rajoutant une entrée toujours égale à 1, est séparable par un hyperplan passant par l'origine.

- **Utilisation du modèle Adaline (Adatative Linear Neuron)** : il a été mis au point pour la résolution du filtrage adaptatif du signal, il appartient à la famille des classificateurs à deux classes. La structure de l'Adaline diffère du Perceptron par l'utilisation d'une seule cellule d'association et l'utilisation d'une fonction de seuil différente de celle de Heaviside (-1 et +1).

$$\sigma = \sum_{j=0}^{j=n} w_j \cdot x_j \tag{5}$$

Il utilise un algorithme adaptatif pour mesurer l'écart entre la sortie réelle, et la sortie du processeur élémentaire.

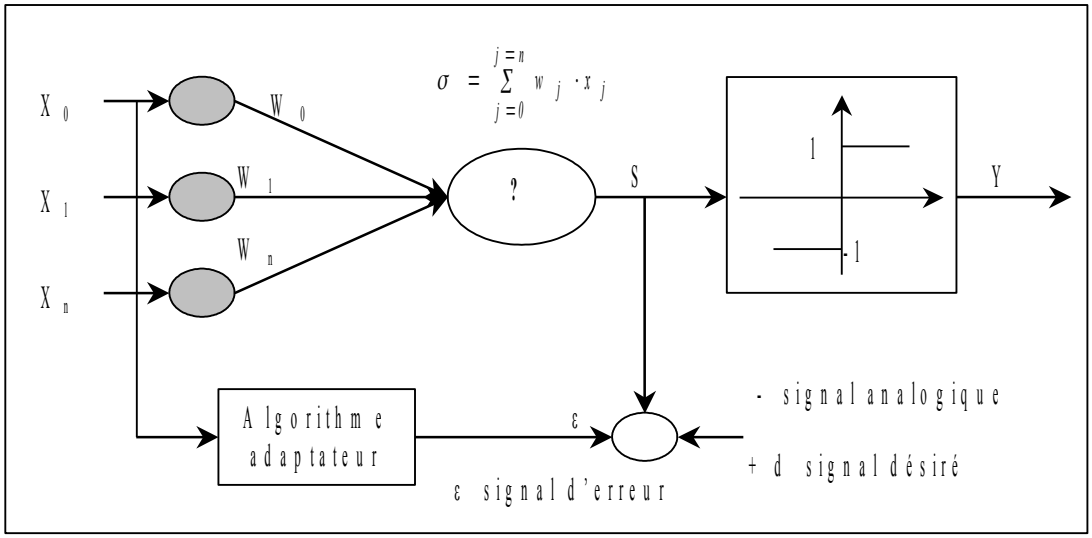


Figure 5 : Schéma de principe de l'Adaline

- **Utilisation du modèle MADALINE (Many Adalines) :** il est constitué de plusieurs Adalines connectés entre eux, suivant le schéma de la figure ci-dessous pour une forme à deux neurones.

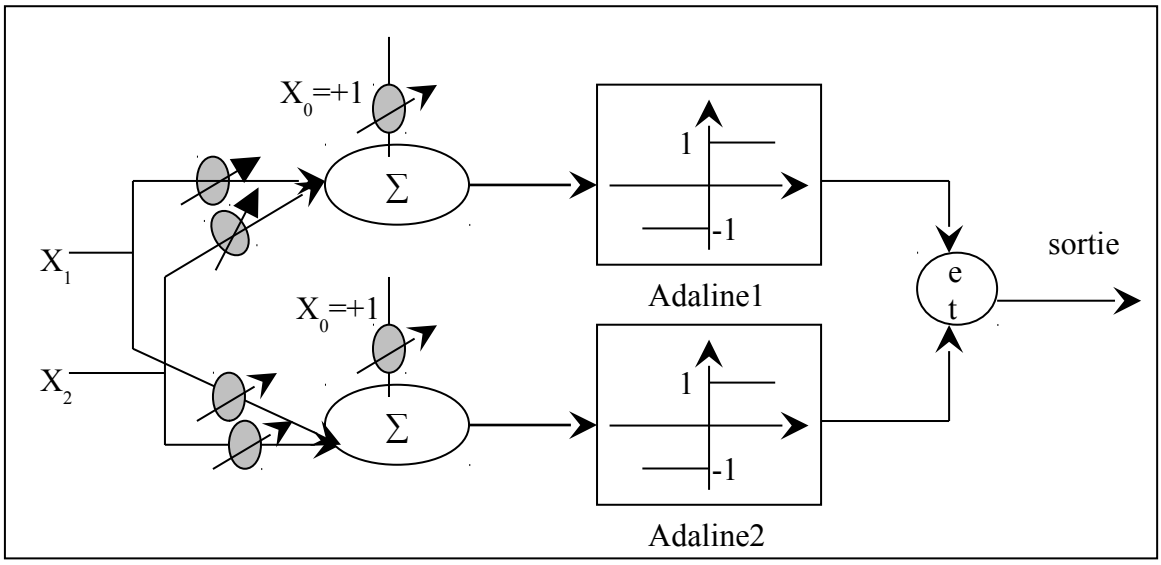


Figure 6 : Schéma de principe du Madaline

L'Adaline et par extension le Madaline utilisent le mode d'apprentissage supervisé avec un « professeur », qui définit l'ensemble des exemples et le jeu des sorties désirées.

La procédure d'apprentissage des poids W_{ij} diffère de la règle du Perceptron, dans la mesure où la modification synaptique des poids, l'incrément du poids synaptique est donnée par :

$$\Delta W_{ij} = \alpha \left(d_j - \sum_{j=1}^{j=n} W_{ij} x_j \right) x_i \qquad (6)$$

Avec : d_j : sorties actuelles.

Cette procédure permet d'obtenir une procédure de la discrimination linéaire, plus performante que celle du Perceptron

III.2.2 - Classification non linéaire

Pour la classification non linéaire, on utilise cette fois ci des réseaux de neurones multi couches. Ils sont aujourd'hui les plus utilisés en diagnostic industriel. L'apprentissage de ces réseaux s'effectue à l'aide de l'algorithme de rétro propagation de gradient dérivé de la règle de Windrow-Hoff, pour le calcul des poids des connexions des neurones des couches internes. Les réseaux à couches sont faciles de mise en œuvre. Généralement, on applique la stratégie suivante

- **Détermination des caractéristiques du réseau :**
  1. Une couche d'entrée qui reçoit l'ensemble des formes à classer.
  2. Une ou plusieurs couches (s) intermédiaire (s) (généralement des cellules sigmoïdes)
  3. Une couche de sortie qui doit restituer l'ensemble des sorties désirées après apprentissage.
  4. La figure suivante représente l'architecture générale d'un réseau à couches :

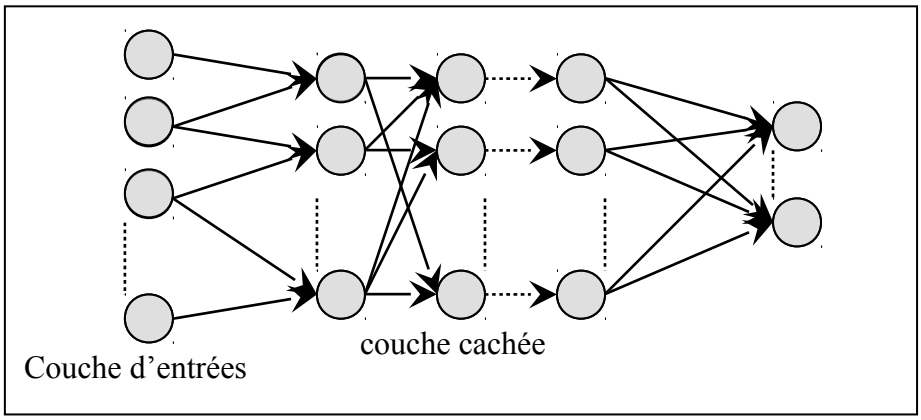


Figure 7 : Architecture d'un réseau à couches

- **Une base d'exemples :**
  1. Entrée x_k , étiquetée $w(x_k)$.
  2. Sortie désirée : 1 pour la cellule associée à la classe de x_k , -1 pour toutes les autres.
  3. Séparation en un ensemble d'apprentissage et un ensemble de test.
- **Apprentissage :**
  1. Algorithme de rétro propagation du gradient.
  2. Si possible, arrêt de l'apprentissage lorsque l'erreur commise pour les formes de test augmente (validation croisée).

L'exemple suivant illustre comment un réseau multi couche permet de séparer deux classes bidimensionnelles par une frontière non linéaire.

Deux classes bidimensionnelles sont représentées dans la figure suivante ainsi que la frontière de décision. Une sortie désirée est imposée [1 ;-1] pour les entrées de la classe ω_1 et à [-1 ; 1] pour ω_2 .

La sortie est obtenue par seuil de la sortie à 0.

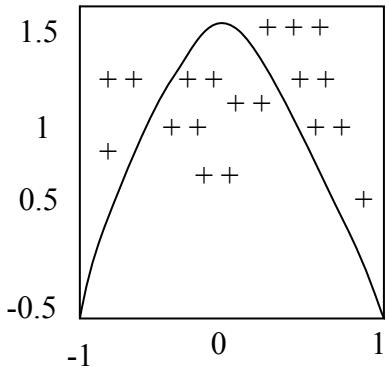


Figure 8 : Frontière de décision obtenue par réseau multicouche

Une fois l'apprentissage du réseau réalisé, la décision d'affectation d'un point à une classe est prise, après avoir comparé la sortie du réseau pour la forme considérée aux sorties types imposées au réseau.

Soient y_i le vecteur cible dédié à la classe ω_i et $y(x)$ le vecteur des sorties réelles du réseau pour la forme d'entrée x , la procédure de décision est la suivante :

Affectation classe ω_i si :

$$\|y(x)-y_i\|^2=\min_{j=1,c}\|y(x)-\dot{y}_j\|^2 \qquad (7)$$

III.3 - Analyse utilisant l'approche neuro-floue

Diverses architectures ont été proposées pour les réseaux neuro-flous, allant d'une fusion des deux types de méthodes à une utilisation séquentielle de l'une ou de l'autre. Nous présenterons celles qui sont les plus utilisées.

III.3.1 - Première architecture

Le premier type d'utilisation, le plus répandu, est représenté par le cas où un système d'inférence flou est mis sous la forme d'un réseau multicouche, dans lequel généralement les poids correspondent aux paramètres du système ; une telle architecture est appelée architecture pré-neuronale.

Par exemple, pour des règles de la forme « si V_1 est A_{i1} et V_2 est A_{i2} alors W est w_i », on fait appel à un réseau de neurones qui admet pour entrées les valeurs x_1 et x_2 prises par les variables V_1 et V_2 et dont les deux couches cachées correspondent respectivement au calcul de la valeur des fonctions d'appartenances A_{i1} pour x_1 et A_{i2} pour x_2 , et à celui de la valeur prise par la conjonction des conditions de chaque règle utilisant un opérateur de conjonction adéquat. Cette situation est schématisée sur la figure (9).

Les fonctions d'appartenance intervenant dans les règles sont considérées comme des paramètres ajustés par l'intermédiaire des poids entrant dans la première couche cachée. Les conclusions w_i des règles sont également des paramètres ajustables par l'intermédiaire des poids associés à la dernière couche.

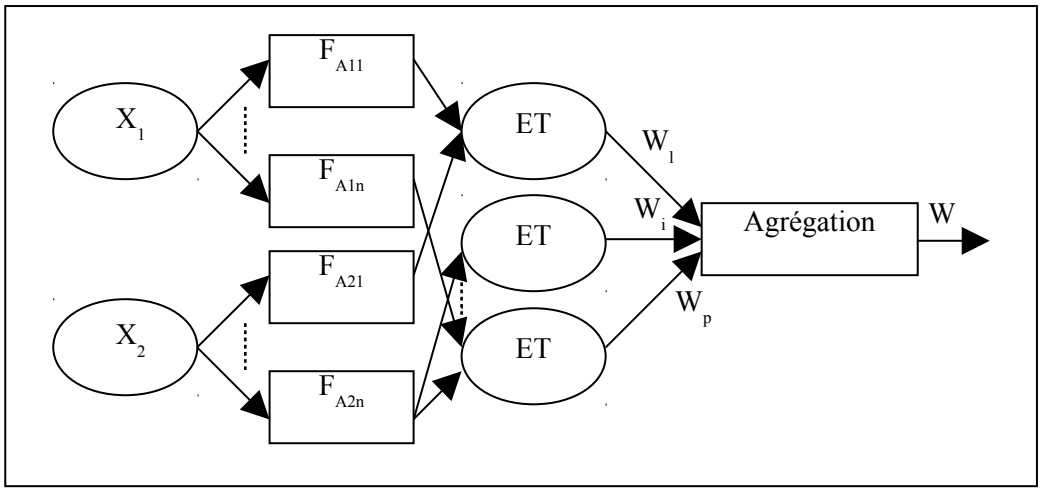


Figure 9: Première architecture neuro-floue

Comme il est indiqué sur la figure (9), les données vont subir trois étapes de calcul :

- **La première couche :** s'occupe du calcul des degrés d'appartenance de chaque entrée. Les paramètres de cette couche vont caractériser la fonction d'appartenance.

Le calcul des degrés d'appartenance s'effectue par des neurones spécialisés, dont la fonction d'activation est la fonction d'appartenance. La fonction la plus utilisée dans ce type d'architecture est la fonction gaussienne ; elle est donnée par :

$$y = \exp - (a \cdot x + b)^2 \qquad (8)$$

Avec :

$$a = \frac{\sqrt{-\log(\varepsilon)}}{dx} \qquad (9)$$

$$b = -a \cdot m$$

Donc les neurones de la première couche ont la forme suivante :

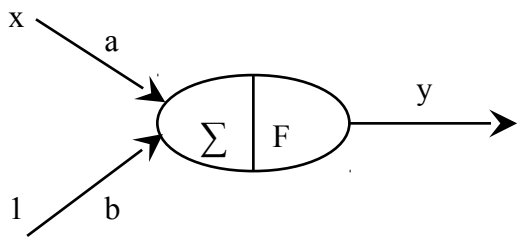


Figure 10 : Forme d'un neurone

- **La deuxième couche cachée** évalue en parallèle les prémisses des règles. Les paramètres de cette couche définissent dans ce cas l'opérateur de conjonction ET.

Plusieurs méthodes ont été proposées pour l'évaluation floue de l'opérateur ET (cette évaluation est appelée aussi calcul de la valeur de vérité). Généralement on utilise celle qui a été proposée par Lukasiewicz, cette dernière présente une certaine facilité d'implémentation neuronale moyennement satisfaisante. Elle est définie par :

$$ET[fA_1(x_1), fA_2(x_2)] = Max(0, fA_1(x_1) + fA_2(x_2) - 1) \qquad (10)$$

Si on pose

$$u = fA_1(x_1) + fA_2(x_2) - 1 \qquad (11)$$

On aura

$$ET[fA_1(x_1), fA_2(x_2)] = Max(0,u) \qquad (12)$$

La fonction $Max(0,u)$ peut être évaluée approximativement par une fonction continue sigmoïde, elle est donnée par

$$f = Max(0,u) = \frac{1}{1+\exp\left(\frac{-u-0.5}{0.227}\right)} \qquad (13)$$

Donc notre opérateur de conjonction ET va prendre la forme suivante :

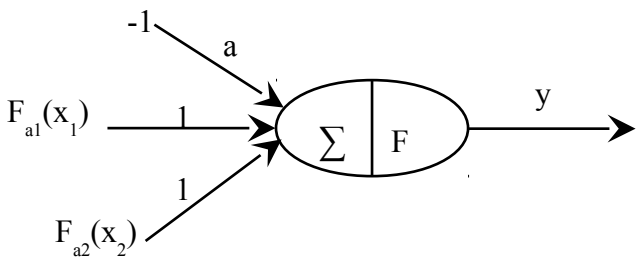


Figure 11 : Forme d'opérateur flou ET

- **La dernière couche :** réalise l'opération d'inférence. Ayant calculé les valeurs de vérités produites par les opérateurs de conjonction, on doit maintenant établir une procédure permettant de déduire une sortie en fonction de ces valeurs de vérités. Cette procédure est réalisée soit en choisissant parmi ces valeurs, celle qui est la plus représentative, c'est à dire celle qui a le plus grand degré de vérité, soit en calculant le barycentre de toutes les valeurs. Une telle approche peut être effectuée par un seul neurone, où les valeurs de vérités sont pondérées par les poids synaptiques de ce neurone. Les poids synaptiques du neurone qui calcule le barycentre (qui fait la défuzzification) sont d'une grande importance, car ce sont eux qui pondèrent les résultats des règles. Une rétro propagation serait indispensable à appliquer sur cette couche (c'est à dire sur les poids synaptiques du neurone chargé de la défuzzification).
- **Algorithme d'apprentissage :** Cet algorithme est basé sur l'algorithme de la rétro propagation du gradient descendant. Il est le même que celui qui a été utilisé pour l'apprentissage des réseaux de neurones type PMC (annexe).La seule différence qui existe, est que les entrées du neurone chargé de la défuzzification seront normalisées. Cette normalisation est nécessaire car elle permet de manipuler des valeurs inférieures ou

égales à l'unité. Une telle approche est réalisée par la division de chaque entrée de ce neurone, sur la somme de toutes ses entrées.

III.3.2 - Deuxième architecture

Le deuxième grand type d'association neuronale et floue correspond à l'utilisation de réseaux de neurones et de systèmes flous organisés en série ou en parallèle. On distingue plusieurs possibilités d'association. On peut tout d'abord construire un réseau de neurones qui fonctionne en amont d'un système flou.

Les variables d'entrées d'un système à commande floue sont, par exemple, déterminées à partir de la sortie d'un réseau de neurones (figure 12), ou bien par un réseau de neurones qui effectue une tâche de classification ou de reconnaissance de forme, et qui est suivi par un système d'aide à la décision floue.

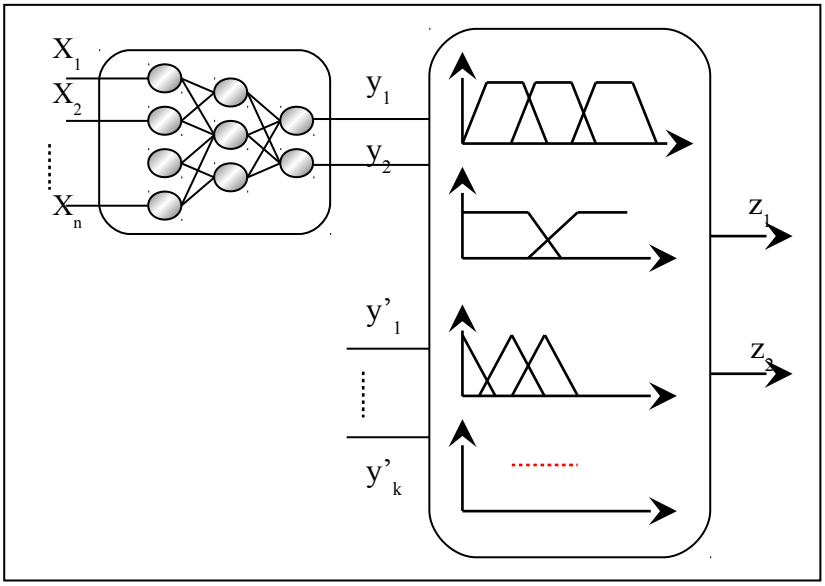


Figure 12 : Deuxième architecture des réseaux Neuro-Flou
Réalisation en série

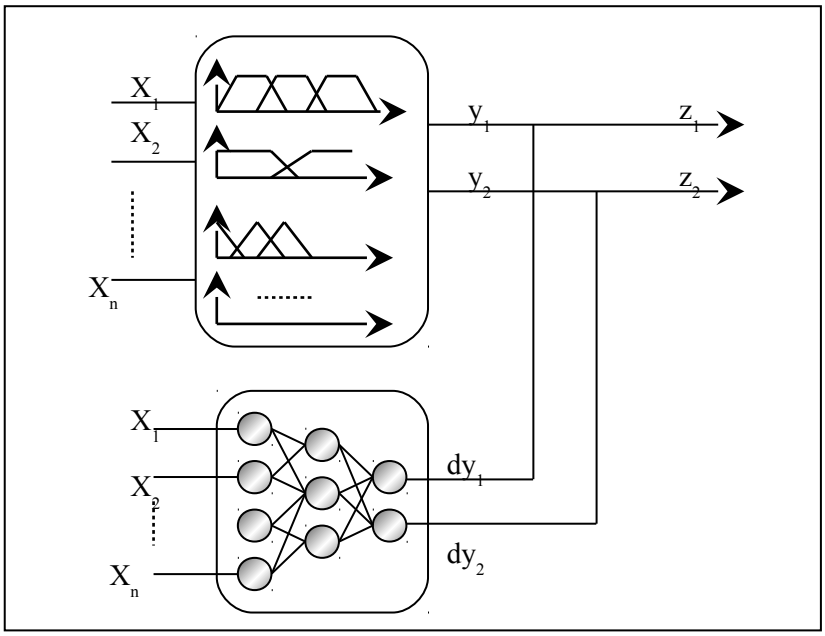


Figure 13: Réseau de neurones fonctionnant en aval d'un système flou

On peut aussi avoir recours au réseau de neurones qui fonctionne en aval d'un système flou. Par exemple, dans le but d'ajuster les sorties d'un système de commande floue aux nouvelles connaissances obtenues. Les variables d'entrée étant l'ensemble de celles du système flou. Les variables de sortie seront celles du système flou plus les corrections résultantes de l'ajustage. Un tel système est représenté sur la figure13.

III.3.3 - Troisième architecture

Ce type d'association entre réseau de neurones et systèmes flous, correspond à l'utilisation des réseaux de neurones pour remplacer toutes ou quelques composantes d'un système flou. De tels réseaux servent à l'apprentissage des fonctions d'appartenance, au calcul de l'inférence, à la réalisation de la phase d'agrégation et de défuzzification. Ils peuvent réaliser l'extraction des règles floues, en analysant la corrélation qui existe entre les entrées et les sorties du réseau de neurones. Ces approches ont une grande importance car elles sont capables de résoudre des problèmes importants :

- La détermination et l'apprentissage des fonctions d'appartenances.
- La détermination des règles.
- L'adaptation à l'environnement du système.

Pour le problème de diagnostic un schéma utilisant la troisième architecture pourrait être le suivant :

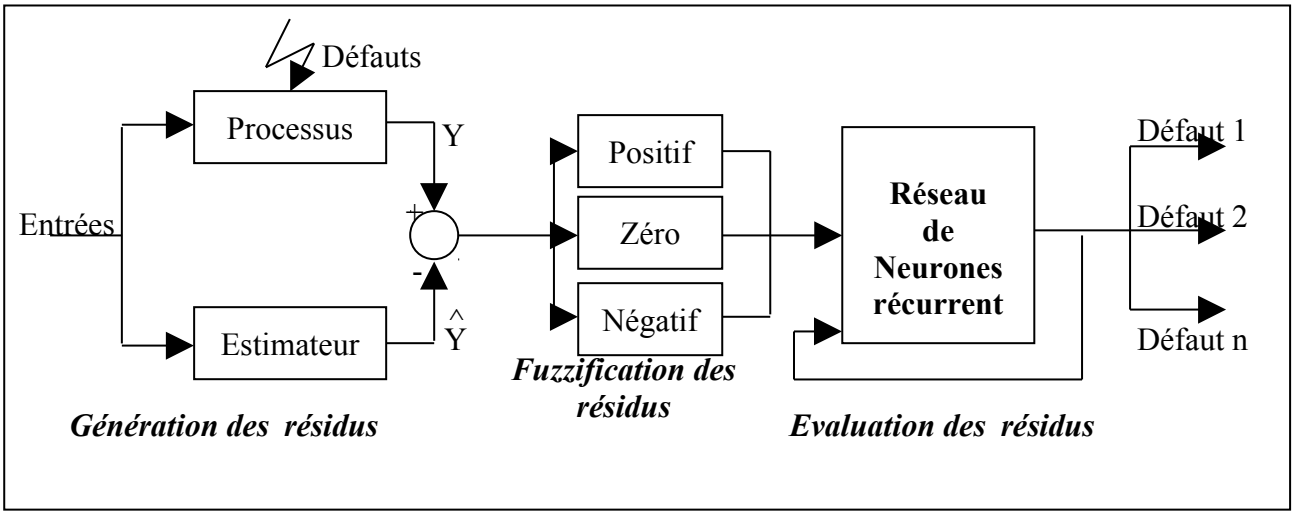


Figure 14 : Schéma de diagnostic Neuro-Flou

Le générateur des résidus peut être issu d'une méthode quantitative ou qualitative ; selon les connaissances sur le modèle on optera pour l'une ou l'autre des deux approches. Dans notre cas, on utilisera un réseau de neurones comme estimateur des sorties de notre système.

L'analyse des résidus est basée, quant à elle, sur l'approche neuro-floue. Plusieurs étapes sont nécessaires pour réaliser cette tâche.

- **Fuzzification des résidus** :A chaque résidu est attribué des fonctions d'appartenance qui vont indiquer avec quel degré il est (ou non) affecté par une défaillance. Généralement on prend comme fonctions d'appartenance des triangles ou trapèzes.

- **Déduction (inférence) :** Pour établir la base de règles, on utilise un réseau de neurones. En entrée du réseau, on a les résidus fuzzifiés (trois fonctions d'appartenance pour chaque résidu) en plus des décisions précédentes (RDN récurrent) et en sorties, on a les décisions, voir la figure(15).

- **Apprentissage :** Une phase d'apprentissage est nécessaire avant d'appliquer en ligne ce réseau. Le but principal de l'apprentissage est l'extraction et la sélection des paramètres qui seront nécessaires pour la modélisation d'une règle « si-alors », qui constituera par la suite une fonction de décision. Dans la phase d'apprentissage, on utilise une base de données où doivent se trouver tous les scénarios possibles de défaillances et les décisions correspondantes. Généralement on utilise l'algorithme de la rétro propagation (voir annexe)

pour l'apprentissage. A titre d'exemple, considérons un système avec deux résidus et trois types de défaillances à détecter.

Le réseau de neurones aura ainsi pour entrée les fonctions d'appartenances { N P Z} de chaque résidu (2x3) en plus de trois décisions précédentes (décisions à l'instant t-1) et trois sorties $D(f_i)$, $i=1\ldots3$.

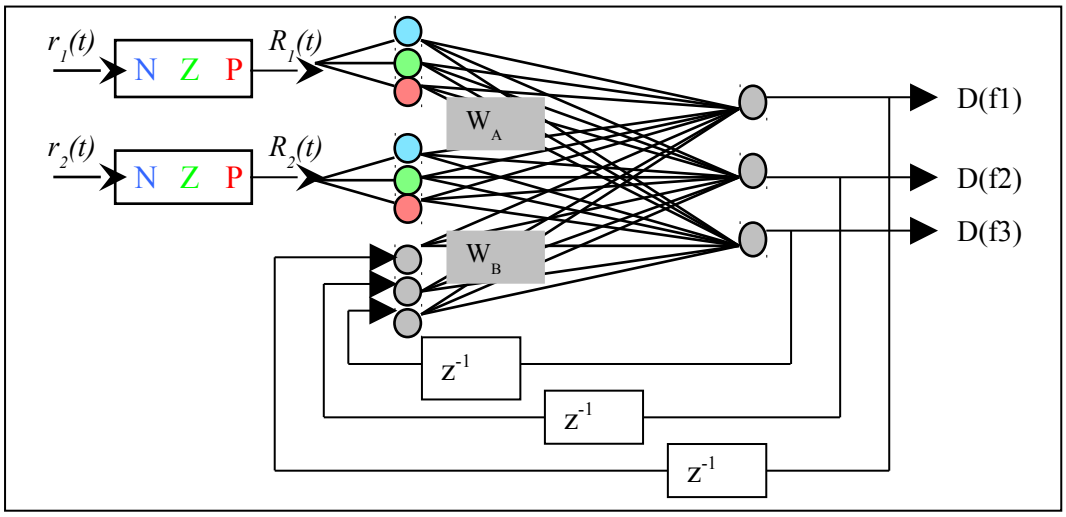


Figure 15 : Analyse des résidus basés sur un système neuro-flou

IV - CONCLUSION

Nous avons opté pour l'utilisation des réseaux de neurones dans l'étape de génération des résidus, en raison de leur capacité à identifier les systèmes dynamiques non-linéaires. De même l'utilisation de l'approche neuro-floue, dans l'étape d'analyse des résidus et de décision est dictée par la capacité d'apprentissage des réseaux de neurones et la souplesse offerte par la logique floue.

D - Identification des Systèmes par réseaux de Neurones

I - INTRODUCTION

L'identification consiste à déterminer modèle reproduisant le mieux possible le procédé.

Deux étapes sont nécessaires : **l'étape qualitative**, où la structure du modèle est fixée, et **l'étape quantitative**, où il s'agit de trouver les valeurs numériques qui permettront au modèle de reproduire le mieux possible le comportement du système. La plupart du temps, l'identification d'un système s'effectue en optimisant un critère d'erreur. Cette erreur caractérise l'écart entre le comportement du procédé et celui de son modèle.

II - IDENTIFICATION

II.1 - Principe de l'identification

Lorsque la structure du modèle est déterminée, il faut évaluer les paramètres du modèle, pour mener à bien cette opération il faut passer par quatre étapes:

1. Choix de l'entrée à appliquer au procédé afin d'obtenir des couples entrées/sorties pertinents.
2. Définir la structure du modèle grâce aux connaissances a priori.
3. Estimer les paramètres du modèle par différentes méthodes.
4. Valider le modèle ainsi obtenu.

Le diagramme ci-dessous résume le cheminement d'une identification classique complète.

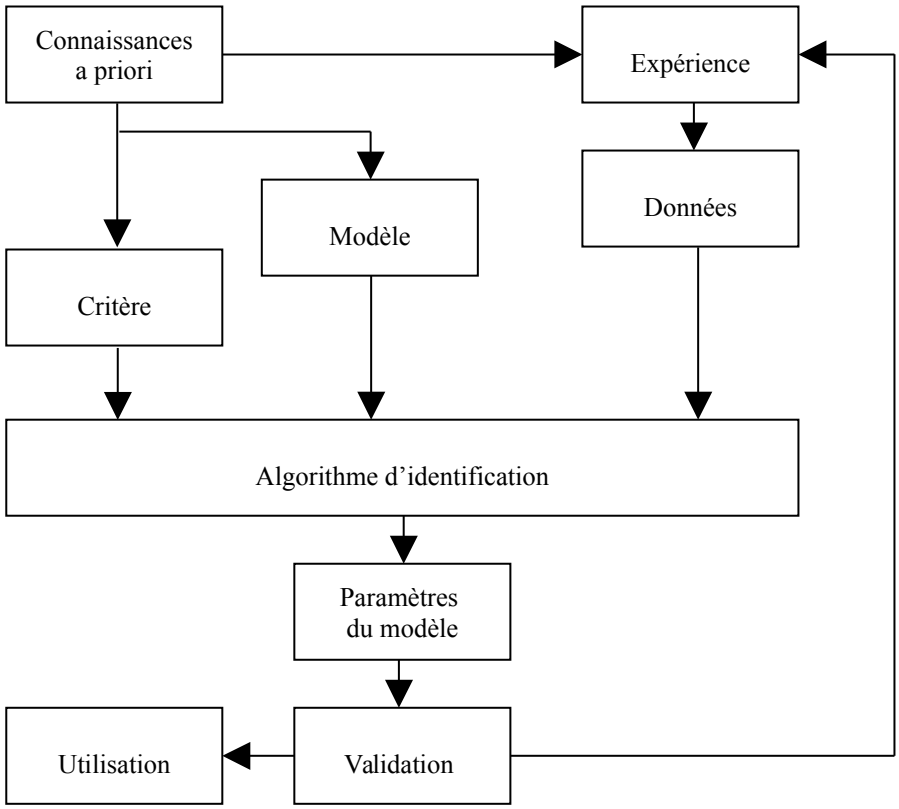


Figure 1 **:** Diagramme de cheminement d'une identification complète

II.2 - Choix des entrées

Le choix des entrées à présenter au système pour son identification n'est pas dû au hasard ; certaines seront inefficaces et d'autres dangereuses. Ainsi une entrée en échelon donnera une bonne estimation du gain statique, mais les paramètres de la dynamique seront complètement erronés. Inversement, une sinusoïde donnera une bonne estimation de paramètres de la dynamique, mais ce genre d'entrée risque de détruire complètement un système industriel. Une entrée idéale serait un

bruit blanc, mais ce signal est physiquement irréalisable.

II.2.1 - Les propriétés intéressantes des entrées

Les principales propriétés intéressantes pour une entrée d'identification sont les suivantes:

- entrée centrée, perturbant peu la sortie.
- spectre riche (pour avoir beaucoup d'information).

Un signal ayant de telles propriétés existe et s'appelle une Séquence Binaire Pseudo Aléatoire : la S.B.P.A.

II.2.2 - La SBPA

Une SBPA est générée par un registre à décalage à N cellules qui s'auto-alimente au travers d'une fonction booléenne (figure 5):

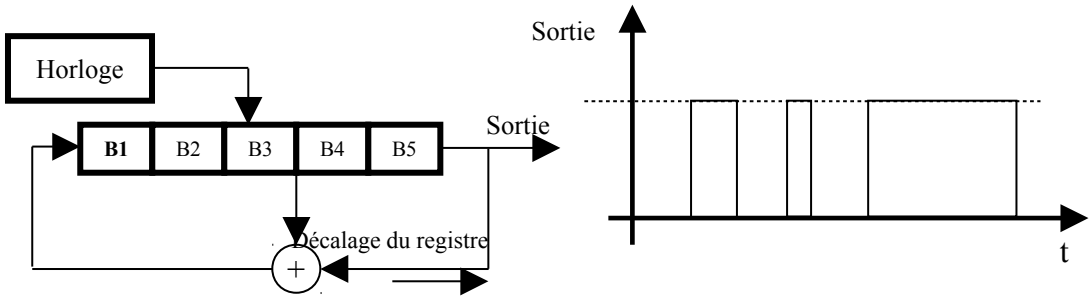


Figure 2 **:** Registre à décalage à **5** cellules

- On ne doit jamais avoir l'état "toutes les cellules à zéro" car la sortie du registre serait constamment nulle.
- La séquence est donc périodique de longueur **$L=2^N-1$.**
- La durée maximale d'une impulsion est définie par $N \cdot Te$ où Te est la période d'échantillonnage du système et N le nombre de cellules. De plus, la durée maximale d'une impulsion doit être supérieure au temps de montée $T \cdot M$, ceci afin de bien identifier le gain statique.
  - Cette dernière inéquation permet de définir le nombre de cellules $N \cdot T_e > M$ du registre. Si N est trop important, il faut alors utiliser un diviseur de fréquence p afin d'éviter des séquences trop longues. On obtient donc:

$$f_{SBPA} = \frac{f_e}{p} \quad \text{avec } p=1,2,3\ldots \qquad (1)$$

Ceci nous donne alors la relation suivante : $p * N * Te > TM$. L'ajout de ce diviseur de

fréquence diminue la qualité de l'identification, mais raccourcit les temps de calcul et d'identification.

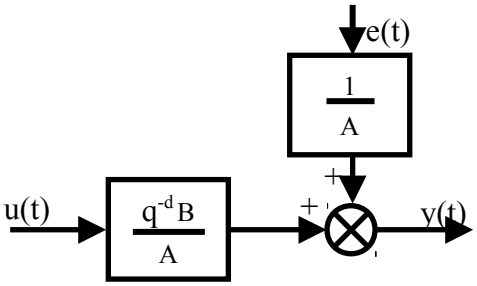
Il faut aussi choisir l'amplitude de la **SBPA**. Les paliers sont choisis a **+U** et **-U** de telle sorte que la moyenne soit nulle, critère intéressant du bruit blanc. Ces valeurs de **U** peuvent être faibles mais doivent toujours être supérieures au bruit.

L'entrée à appliquer au procédé étant dorénavant définie, nous allons nous intéresser au choix de la structure du modèle.

II.3 - Structures du modèle

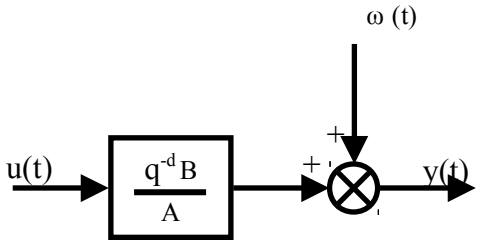
En identification "classique", pour la méthode du modèle, et plus précisément pour les méthodes d'identification récursives, quatre structures de modèles différentes apparaissent. Toutes les structures intègrent le procédé plus une perturbation. Les quatre modèles sont présentés ci-dessous :

Structure S1:



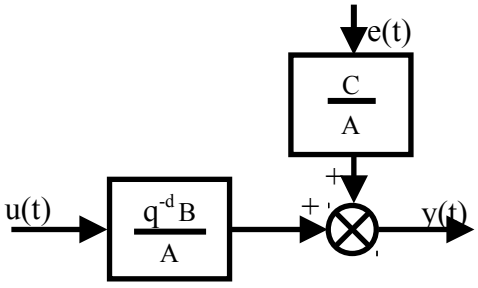
$$A(q^{-1})\, y(t) = q^{-d} B(q^{-1}) u(t) + e(t)$$

Structure S2:



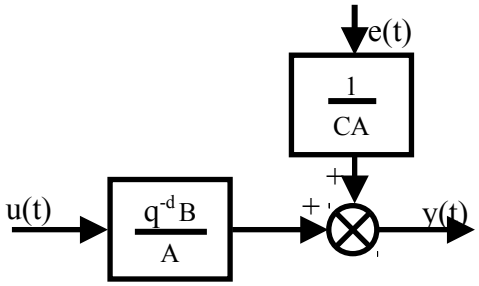
$$A(q^{-1})\, y(t) = q^{-d} B(q^{-1}) u(t) + A(q^{-1}) \omega(t)$$

Structure S3:



$$A(q^{-1})\, y(t) = q^{-d} B(q^{-1}) u(t) + C(q^{-1}) e(t)$$

Structure S4:



$$A(q^{-1})\, y(t) = q^{-d} B(q^{-1}) u(t) + [1/C(q^{-1})] e(t)$$

Figure 3 **:** Représentation des quatre structures d'identification récursive

Ces structures nécessitent une connaissance a priori du système, pour pouvoir être utilisées. Cette connaissance se réduit principalement à l'ordre du procédé et à la nature des perturbations qui s'ajoutent à celui-ci.

La structure du modèle étant définie, il nous faut maintenant passer à l'identification proprement dite.

II.4 - Estimation des paramètres

Il existe beaucoup de méthodes d'identification paramétriques. Elles sont regroupées dans deux types :

- **les méthodes non-récursives** : moindre carré simple, adaptées pour une identification hors-ligne.
- **les méthodes récursives** : moindre carré récursif, moindre carré généralisé, qui permettent une identification en temps réel sur le procédé.

Le diagramme ci-dessous résume le principe de l'estimation paramétrique du modèle échantillonné.

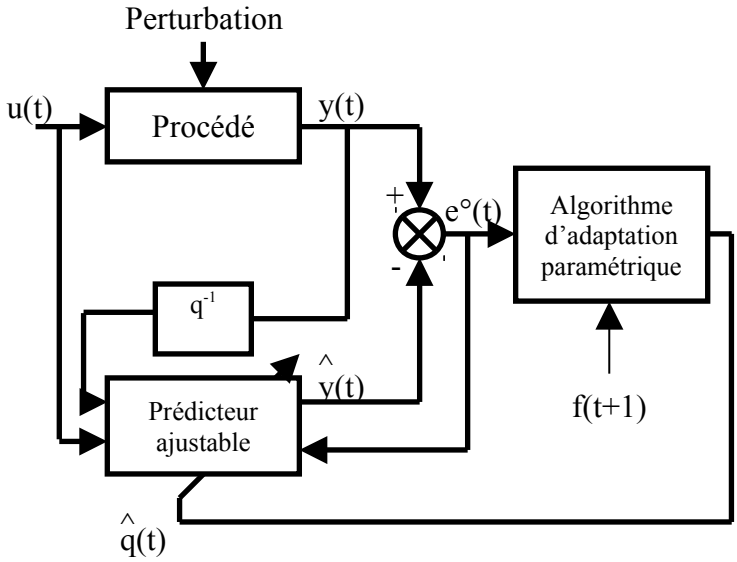


Figure 4 **:** Diagramme de principe des méthodes d'identification récursives

Il s'agit donc à partir des sorties ***y (t)*** réelles et **ŷ(t)**prédites du modèle et d'un critère **ε°(t)** l'erreur de prédiction, de modifier les paramètres ajustables du modèle par le biais d'un algorithme d'adaptation paramétrique. Ils sont assez nombreux et ils fonctionnent en corrélation avec une structure parmi celles énumérées dans le paragraphe précédent.

Ces algorithmes sont regroupés dans deux types :

- ceux basés sur le blanchiment de l'erreur de prédiction.
- Ceux basés sur la décorrélation du vecteur des observations et de l'erreur de prédiction.

Algorithme d'Adaptation Paramétrique	**Structure de modèle associée**
Méthodes basées sur le blanchiment et de l'erreur de prédiction	
Moindre Carré Récursifs	**S1, S2 si $A(q^{-1})\omega(t) = e(t)$**
Moindre Carré Etendu	**S3**
Moindre Carré Généralisé	**S4**
Maximum de vraisemblance	**S4**
Erreur de sortie avec modèle de prédiction étendu	**S3**
Méthodes basées sur la décorrélation du vecteur des observations et de l'erreur de prédiction	
Variable instrumentale à observation retardée	**S1, S2**
Variable instrumentale à modèle auxiliaire	**S1, S2 si $A(q^{-1})\omega(t) = C(q^{-1})e(t)$**
Erreur de sortie à compensateur fixe	**S2**
Erreur de sortie à compensateur ajustable	**S2**

Tableau récapitulatif des AAP et de leurs modèles associés

II.5 - Validation

Nous avons vu qu'il y avait deux types d'AAP donc il y aura deux types de validation.

La première concerne les méthodes de blanchiment de l'erreur de prédiction : c'est le test de "**blancheur**". Il vise à montrer que l'erreur de prédiction modèle-procédé tend à se rapprocher d'un bruit blanc : un signal de moyenne nulle indépendant de l'entrée et de la sortie. Il faut alors vérifier

$$\lim_{t\to\infty} E\left[\varepsilon(t)\cdot\varepsilon(t-1)\right]=0 \quad \text{avec } i = 1,2,3\ldots\ ;-1,-2,-3\ldots \qquad (2)$$

Avec {ε(t)} séquence centrée des erreurs

La deuxième concerne la méthode de décorrélation du vecteur des observations et de l'erreur de prédiction. Cette méthode vise à montrer que les deux variables sont décorréllées en vérifiant que l'espérance mathématique du produit est nulle /LANDAU 93/. Il faut alors vérifier :

$$E\left[\varepsilon(t)\cdot\hat{y}(t-i)\right]\approx\frac{1}{N}\sum_{t=1}^{N}\varepsilon(t)\cdot\hat{y}(t-i)=0 \quad ;\ i=1,2,3\ldots \qquad (3)$$

Avec {ε(t)} séquence centrée des erreurs de sortie.

Avec {ŷ(t)} séquence centrée des sorties du modèle.

II.6 - Avantages et inconvénients des méthodes d'identification récursives

- **Les avantages** :

Ces méthodes conduisent à des calculs simples et rapides par les micro-ordinateurs.

Elles permettent aussi l'identification en ligne du procédé et ainsi un gain de temps et de précision car on travaille sur le système réel.

- **Les inconvénients** :

L'espérance de recherche avec ces méthodes est assez modeste. Un maximum de cinq à sept paramètres à identifier.

On doit avoir une bonne connaissance à priori du système.

Il faut aussi que les fonctions à minimiser, fonction de coût, soient continues et différentiables.

Il se pose aussi le problème des non linéarités. En général une identification paramétrique est utilisée pour un système linéaire : le système possède un comportement unique lorsqu'on le soumet à un couple d'entrée/sorties. Si l'on passe outre, l'identification risque d'être erronée. Ou encore on l'identifie autour d'un point de fonctionnement et on le linéarise.

C'est en ces termes que ces méthodes d'identification trouvent leurs limites.

III - IDENTIFICATION PAR RÉSEAUX DE NEURONES

L'utilisation des réseaux de neurones pour l'identification des systèmes non linéaires découle naturellement des aptitudes de ces derniers à l'approximation et la généralisation.

La détermination du modèle dynamique d'un système comporte en général les étapes suivantes :

- **Acquisition des données d'apprentissage et de test :**

Cette étape fournit les données entrées/sorties susceptibles de permettre l'extraction d'un modèle de procédé significatif.

- **Choix de la structure du modèle *:***

La deuxième étape consiste à choisir la structure du modèle susceptible de représenter la dynamique du système, l'architecture du réseau de neurones et ses entrées. Les réseaux multicouches statiques sont les plus utilisés à cause de la simplicité de leurs algorithmes d'apprentissage et leurs aptitudes à l'approximation et à la généralisation. Il n'existe pas de méthodes générales pour le choix du nombre de neurones sur chaque couche cachée ainsi que le nombre de ces dernières. Cependant, un réseau à une seule couche cachée est dans la majorité des cas suffisant.

- **Estimation des paramètres du modèle *:***

Après avoir choisi la structure du modèle, il faut estimer les paramètres de ce dernier. Ces paramètres sont les poids de connexions entres les neurones qui sont adaptés de telle sorte à minimiser un critère de performance ; ceci est appelé dans la littérature des réseaux de neurones **apprentissage**.

- **Validation du modèle identifié *:***

La dernière étape doit permettre de mettre en évidence si le modèle identifié est représentatif des comportements entrées/sorties du système. Plusieurs méthodes de validation sont données dans la référence.

III.1.1 - Structures de modèles linéaires

Un système est dit linéaire s'il est possible de le décrire par un modèle de la forme suivante :

$$y(t)=G(q^{-1})u(t)+H(q^{-1})e(t) \qquad (4)$$

Où **G** et **H** sont des fonctions de transfert de l'opérateur de retard q^{-1}**.**

L'opérateur de retard q^{-1} agit sur le signal de la manière suivante :

$$q^{-d}x(t) = x(t-d) \tag{5}$$

Où d est un multiple de la période d'échantillonnage.

e(t) : est un bruit blanc qui est indépendant des entrées précédentes et qui peut être caractérisé par une fonction de densité de probabilité. Dans le cas multivariable, u(t), y(t) et e(t) sont des vecteurs et G et H sont des matrices polynomiales.

On suppose que le système réel décrit par :

$$y(t) = G_0(q^{-1})\, u(t) + H_0(q^{-1})\, e_0(t) \tag{6}$$

Et si on pose μ qui représente un ensemble paramétré de modèles candidats.

$$\mu : \{G(q^{-1}, \theta), H(q^{-1}, \theta) \theta \in D_m\}$$

$$y(t) = G(q^{-1}, \theta)\, u(t) + H(q^{-1}, \theta)\, e(t) \tag{7}$$

Où θ représente les p paramètres ajustables et D_m est un sous ensemble de R^p à l'intérieur duquel la recherche du modèle doit être effectuée.

La structure du modèle sous la forme prédictive est donnée par :

$$\hat{y}(t/t-1,\theta) = H^{-1}(q^{-1},\theta)\, G(q^{-1},\theta)\, u(t) + [1 - H^{-1}(q^{-1},\theta)]y(t) \tag{8}$$

La structure du modèle est souvent écrite sous la forme équivalente :

$$\hat{y}(t\backslash\theta) = \varphi^T(t) \tag{9}$$

θ étant le vecteur paramètre et φ le vecteur de régression contenant les entrées et les sorties précédentes ou des signaux obtenus à partir des entrées et des sorties.

La structure générale d'un modèle peut être réécrite comme suit :

$$A(q^{-1})y(t) = q^{-d}(B(q^{-1})/F(q^{-1}))u(t) + (C(q^{-1})/D(q^{-1}))e(t) \tag{10}$$

Où

$$A(q^{-1}) = 1 + a_1 q^{-1} + \ldots\ldots\ldots + a_n q^{-n}$$

$$B(q^{-1}) = b_0 + b_1 q^{-1} + \ldots\ldots\ldots + b_m q^{-m}$$

$$C(q^{-1}) = 1 + c_1 q^{-1} + \ldots\ldots\ldots + c_k q^{-k}$$

$$D(q^{-1}) = 1 + d_1 q^{-1} + \ldots\ldots\ldots + d_l q^{-l} \qquad (11)$$

$$F(q^{-1}) = 1 + f_1 q^{-1} + \ldots\ldots\ldots + f_r q^{-r}$$

III.1.2 - Structure du modèle à réponse impulsionnelle finie (FIR)

La plus simple structure du modèle correspond au choix

$$G(q^{-1},\theta)=q^{-d}B(q) \qquad H(q^{-1},\theta)=1 \qquad (12)$$

Le prédicteur est donc donné par :

$$\hat{y}(t\backslash\theta)=q^{-d}B(q^{-1})u(t) \qquad (13)$$

Sous forme régressive ce dernier peut être écrit comme suit :

$$\hat{y}(t\backslash\theta)=\varphi^T(t) \qquad (14)$$

$$\text{où} \quad \varphi(t)=[u(t-d) \ldots u(t-d-m)]^T \qquad (15)$$

Et le vecteur paramètres $\theta= [\, b_0 \ldots b_m \,]^T$

Un système avec des pôles ne peut être décrit avec exactitude par un modèle FIR. Cependant si le système est stable et la réponse impulsionnelle décroît assez rapidement, le système peut, souvent, être bien approximé par un modèle FIR si $B(q^{-1})$ est choisi comme les premiers m coefficients de la réponse impulsionnelle.

III.1.3 - Modèle autorégressif à entrée exogène (ARX)

La structure du modèle correspond au choix :

$$G(q^{-1},\theta)=q^{-d}B(q^{-1})/A(q^{-1}) \qquad H(q^{-1},\theta)=1/A(q^{-1}) \tag{16}$$

La forme du prédicteur est alors :

$$\hat{y}(t\backslash\theta) = q^{-d}B(q^{-1})\,u(t) + [\,1 - A(q^{-1})\,]\,y(t) \tag{17}$$

$$= \varphi^T(t)\theta$$

Avec $\varphi(t) = [y(t-1)\ldots y(t-n),\ u(t-d),\ \ldots u(t-d-m)]^T$ (18)

$$\theta = [-a_1\ldots\ldots -a_n,\ b_0\ldots b_m\,]^T$$

Même si G a des pôles, il reste simplement une relation algébrique entre la prédiction, les entrées précédentes et les sorties mesurées. Par conséquent le prédicteur sera toujours stable même si le système ne l'est pas.

III.1.4 - Modèle autorégressif à moyenne mobile et entrée exogène (ARMAX)

Ce modèle, plus général que le ARX, a la structure suivante :

$$G(q^{-1},\theta)=q^{-d}B(q^{-1})/A(q^{-1}) \qquad H(q^{-1},\theta)=C(q^{-1})/A(q^{-1}) \tag{19}$$

Et le prédicteur optimal est:

$$\hat{y}(t\backslash\theta)= q^{-d}\,(B\,(q^{-1})/C\,(q^{-1}))\,u\,(t) + (1 - A\,(q^{-1})/C\,(q^{-1}))\,y\,(t) \tag{20}$$

$$= q^{-d}\,B\,(q^{-1})\,u\,(t) + [1 - A\,(q^{-1})]\,y\,(t) + [C\,(q^{-1}) - 1]\,\varepsilon(t,\theta) \tag{21}$$

$$= \varphi^T(t,\theta)\,\theta\,.$$

$\varepsilon(t,\theta)=y(t)-\hat{y}(t\backslash\theta)$ représente l'erreur de prédiction ou résidu.

Les vecteurs régression et paramètres sont définis par :

$$\varphi(t,\theta) = [y(t-1)\ldots y(t-n),\ u(t-d)\ldots u(t-d-m),\ \varepsilon(t,\theta),\ \ldots,\ \varepsilon(t-k,\theta)\,]^T$$

(22)

$$\theta = [-a_1,\ldots -a_n, b_0\ldots b_m, c_1,\ldots c_k]^T$$

A cause de la présence du polynôme C, le prédicteur a cette fois-ci des pôles. Les racines du polynôme C doivent être à l'intérieur du cercle unité pour que le prédicteur soit stable.

Ainsi, la présence des pôles implique que le vecteur de régression dépend des paramètres du

modèle.

III.1.5 - Structure du modèle erreur de sortie (OE)

Le modèle erreur de sortie (ou modèle parallèle) est seulement utilisé quand le bruit affectant le système est un bruit de mesure blanc.

$$y(t)=q^{-d}(B(q^{-1})/F(q^{-1}))u(t)+e(t) \tag{23}$$

Ce qui correspond au choix suivant de G et H:

$$G(q^{-1}, \theta)=q^{-d}B(q^{-1})/F(q^{-1}) \qquad H(q^{-1},\theta)=1 \tag{24}$$

Le prédicteur est donc donné par :

$$\begin{aligned} \hat{y}(t\backslash\theta) &= q^{-d}\ (B(q^{-1})/F(q^{-1}))\ u(t) \\ &= q^{-d}\ B(q^{-1})u(t) + [1 - F(q^{-1})]\ \hat{y}\ (t\backslash\theta) \\ &= \varphi^{T}(t,\theta)\ \theta\ . \end{aligned} \tag{25}$$

Avec $\varphi(t,\theta) = [\hat{y}\ (t-1\backslash\theta), \ldots, \hat{y}\ (t-r\backslash\theta), u(t-d), \ldots u(t-d-m)]$

$\theta = [-f_1, \ldots, -f_r, b_0 \ldots, b_m]^T$

Pour que le prédicteur soit stable, il faut que les racines de F soient à l'intérieur du cercle unité.

III.2 - Structures de modèles non linéaires

Le réseau multicouche est capable d'apprendre des relations non linéaires à partir d'un ensemble de données. Il est naturel donc de le choisir pour l'identification des systèmes non linéaires.

Ceci se résume au choix:

- des entrées du réseau ;
- et de l'architecture interne du réseau.

Une approche souvent utilisée est de réemployer les structures d'entrée des modèles linéaires et de prendre comme architecture interne un réseau multicouche.

Cette approche présente plusieurs avantages:

- une extension naturelle des structures bien connues de modèles linéaires ;
- une architecture interne qui peut être graduellement étendue si on veut modéliser des relations non linéaires plus complexes ;
- des décisions sur la structure exigées par l'utilisateur sont réduites à un niveau raisonnable ;
- elle est adéquate pour la conception des systèmes de commande.

Les équivalents non linéaires des structures de modèles linéaires présentées précédemment sont obtenus en mettant:

$$y(t) = g[\varphi(t,\theta),\theta] + e(t) \qquad (26)$$

Ou avec une forme prédictive :

$$\hat{y}(t\backslash\theta) = g[\varphi(t,\theta), \theta] \qquad (27)$$

$\varphi(t,\theta)$ étant le vecteur de régression, θ le vecteur des paramètres ajustables du réseau de neurones à savoir les poids et g la fonction réalisée par le réseau de neurones qu'on suppose en boucle ouverte **.**

Selon le choix du vecteur de régression, différentes structures du modèle non linéaire émergent. Si le vecteur de régression est choisi comme pour les modèles ARX**,** la structure du modèle est appelée NNARX (Neural Network ARX)**.** Il en est de même pour les structures NNFIR, NNARMAX, NNOE.

III.2.1 - NNFIR ET NNARX

Comme pour leurs équivalents linéaires, les prédicteurs sont toujours stables puisqu'ils sont de pures relations algébriques entre la prédiction et les mesures antécédentes des entrées et des sorties.

Ceci est particulièrement important dans le cas non linéaire puisque la recherche de la stabilité est plus complexe ici que dans les systèmes linéaires.

Les structures sont montrées dans la (Figure 5) :

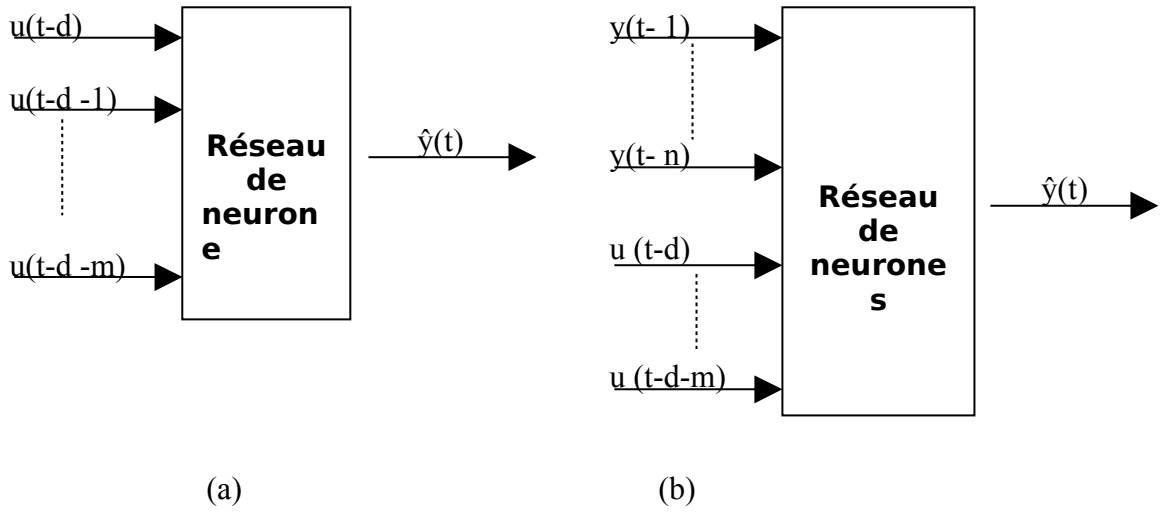


Figure 5 **:** Les structures des modèles NNFIR (a) et NNARX (b)

L'absence de problèmes relatifs à la stabilité dans ces structures de modèles, en particulier la structure NNARX fait d'elle le meilleur choix quand le système est déterministe ou faiblement bruité.

III.2.2 - NNARMAX

Bien que la fonction g dans la figure (5) soit réalisée par un réseau de neurones en boucle ouverte, le prédicteur doit avoir un retour quand les régressions sont sélectionnées comme dans le modèle ARMAX.

$$\varphi(t,\theta)=[y(t)\ldots y(t-n),\ u(t-d)\ \ldots\ u(t-d-m),\ \varepsilon(t,\theta),\ \ldots\varepsilon(t-k,\ \theta)]^T \qquad (28)$$

Les erreurs de prédiction antérieures dépendent des sorties du modèle et par conséquent elles établissent une boucle de retour, voir Figure (6).

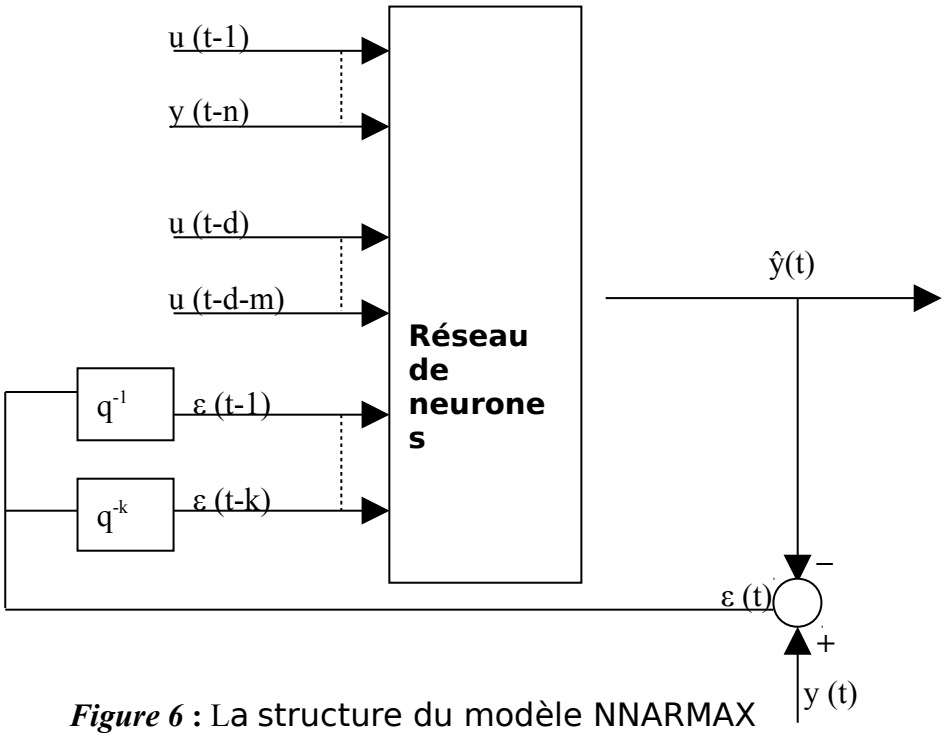


Figure 6 **:** La structure du modèle NNARMAX

Un modèle de réseau avec un bouclage est généralement appelé réseau récurrent. En inspectant les racines du polynôme, il est facile de savoir si le prédicteur d'un modèle linéaire ARMAX est stable.

Pour les modèles NNARMAX il est plus difficile d'analyser les propriétés de stabilité. Il serait toutefois judicieux de considérer la stabilité comme une propriété locale. Il n'en demeure pas moins qu'un modèle NNRMAX soit stable dans certains régimes et instable dans d'autres. Dans l'un ou l'autre cas, le problème en pratique va dépendre du comportement du système.

III.2.3 - NNOE

Quelques régressions dans le modèle NNOE sont des prédictions des sorties antérieures, ce modèle est donc sujet aux mêmes problèmes que la structure NNARMAX. Le modèle NNOE est donné par la figure (7) :

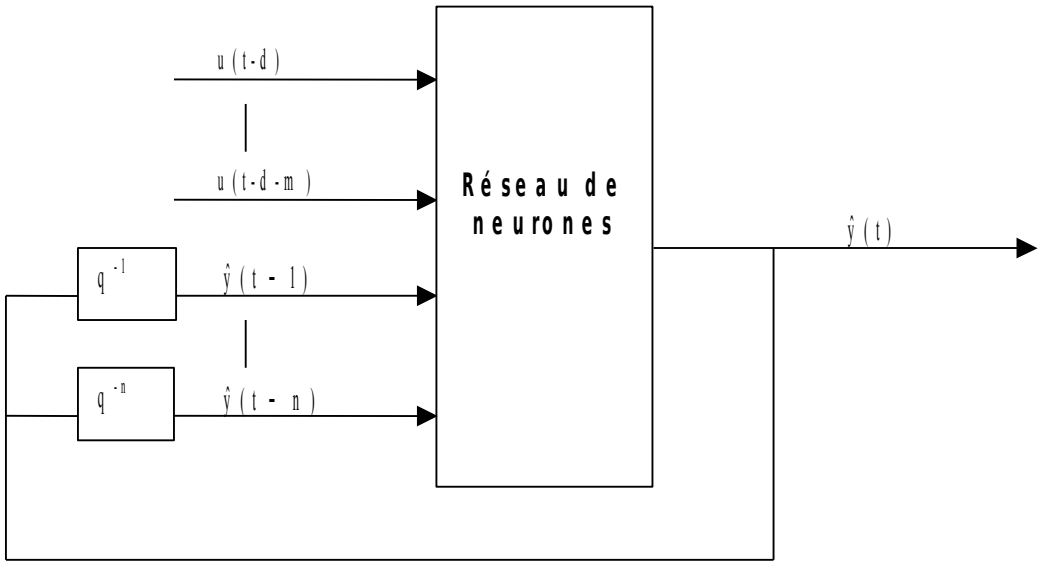


Figure 7 : La structure du modèle NNOE

IV - CONCLUSION

Dans ce chapitre, une présentation de l'identification classique et de l'identification par les réseaux de neurones des systèmes linéaires et non linéaires a été donnée en montrant les différents types structures de modèle, les différentes étapes à suivre, les avantages des méthodes utilisées ainsi que leurs limites.

Dans le chapitre suivant, nous présentons le modèle de la machine asynchrone : on verra que ce modèle est non-linéaire, et on s'intéressera naturellement à son identification par réseau de neurones dans le cadre du schéma de diagnostic qui sera étudié en simulation.

E - Modélisation de la MAS

I - INTRODUCTION

Le moteur asynchrone ou moteur d'induction est actuellement le moteur électrique dont l'usage est le plus répandu dans l'industrie. Son principal avantage réside dans l'absence de contacts électriques glissants, ce qui conduit à une structure simple et robuste facile à construire. Le domaine de puissance va de quelques watts à plusieurs mégawatts. Relié directement au réseau industriel à tension et fréquence constantes, il tourne à vitesse variable peu différente de la vitesse synchrone; c'est lui qui est utilisé pour la réalisation de la quasi-totalité des entraînements à vitesse constante. Le moteur asynchrone permet aussi la réalisation d'entraînements à vitesse variable et la place qu'il prend dans ce domaine ne cesse de croître.

La difficulté de la simulation du moteur asynchrone provient de sa structure fortement non linéaire. L'utilisation de la transformation de Park permet une simplification de la simulation de la MAS. Dans ce chapitre on présente le modèle de Park du moteur asynchrone qui sera utilisé par la suite en simulation et sur lequel on appliquera la méthode de diagnostic neuro-flou retenue dans notre étude.

II - DESCRIPTION DU MOTEUR ASYNCHRONE

Le moteur asynchrone est formé :

- **d'un stator :** anneau de tôles encoché à l'intérieur et portant un enroulement triphasé semblable à celui d'un alternateur. Cet enroulement est presque toujours relié à la source et constitue le primaire.
- **d'un rotor :** anneau de tôles rainuré à l'extérieur, concentrique au premier et séparé de lui par un entrefer étroit d'épaisseur constante. Le rotor porte un enroulement polyphasé mis en court-circuit constituant le secondaire. On distingue principalement deux types de structures de rotors :
  - → **Un rotor bobiné** portant un enroulement du même type que celui du stator permet en reliant les trois bornes (si le bobinage est triphasé) à trois bagues sur lesquelles appuyant trois balais d'avoir accès aux phases rotoriques pour en modifier, par exemple, les caractéristiques.
  - → Chaque fois que c'est possible on utilise un moteur avec **rotor à cage (d'écureuil).** Chaque encoche contient une barre ; le courant qui y passe revient par la barre située à une distance polaire. On a confondu en deux anneaux d'extrémité l'ensemble des connexions reliant deux à deux les barres.

C'est la structure la plus robuste. Elle est utilisée dans les machines de faible puissance (moins de 10 kW), c'est à dire essentiellement destinées aux applications domestiques.

- Dans les deux cas, le circuit est associé à un circuit magnétique qui doit canaliser le flux.

Le moteur asynchrone est donc caractérisé par :

- la présence d'un seul bobinage polyphasé alimenté par une source extérieur, au stator ;
- la présence d'un « bobinage » massif en court-circuit, au rotor.

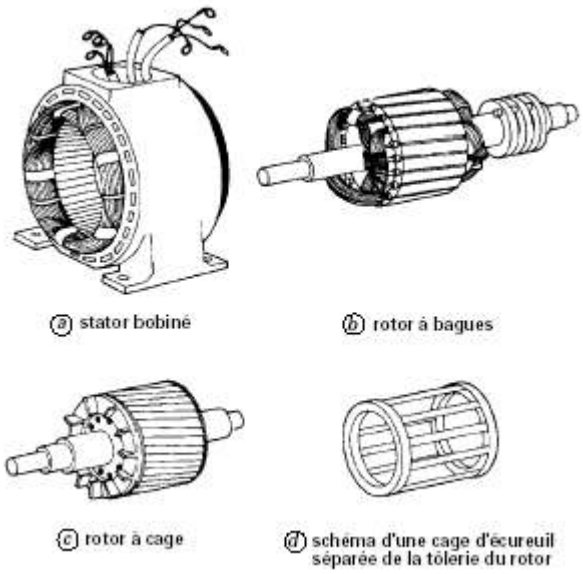


Figure 1 : Représentation d'une façon moins schématique des éléments d'une machine asynchrone.

III - MODÉLISATION DU MOTEUR ASYNCHRONE

Elle est basée sur de nombreuses hypothèses, parmi ces dernières nous citons :

- entrefer parfaitement lisse,
- pertes fer négligeables,
- saturation dans le circuit magnétique négligeable,
- harmoniques d'espace négligeables ; les enroulements du stator et du rotor sont à répartition sinusoïdale de sorte que les inductances mutuelles entre le stator et le rotor sont des fonctions sinusoïdales de la position mécanique du rotor par rapport au stator.

III.1 - Equations électriques et mécaniques

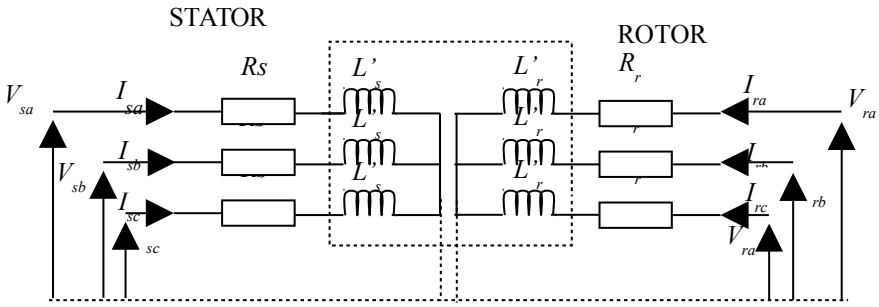


Figure 2 : Schéma électrique d'un moteur asynchrone

Les paramètres du moteur asynchrone sont :

R_S : résistance d'une phase du stator ;

L_S : inductance propre d'une phase du stator ;

R_r : résistance d'une phase du rotor ;

L_r : inductance propre d'une phase du rotor ;

M_s (M_r) : inductance mutuelle entre deux phases du stator (du rotor) ;

M_{rs} : inductance mutuelle (maximale) entre une phase du stator et une phase du rotor ;

P : nombre de paires de pôles ;

J : Inertie ;

fv : Frottement visqueux ;

Ω : la vitesse mécanique du moteur ;

$$U=R.I+\frac{d\varphi}{dt} \quad \text{et } \phi = L.I \tag{1}$$

Pour le stator :

$$U_S=\begin{bmatrix} U_{sa} \\ U_{sb} \\ U_{sc} \end{bmatrix}=\begin{bmatrix} R_S & 0 & 0 \\ 0 & R_S & 0 \\ 0 & 0 & R_S \end{bmatrix}\begin{bmatrix} I_{sa} \\ I_{sb} \\ I_{sc} \end{bmatrix}+\frac{d}{dt}\begin{bmatrix} \Phi_{sa} \\ \Phi_{sb} \\ \Phi_{sc} \end{bmatrix}$$

$$L_S=\begin{bmatrix} L_s & M_s & M_s \\ M_s & L_s & M_s \\ M_s & M_s & L_s \end{bmatrix} \tag{2}$$

Pour le rotor :

$$V_r=\begin{bmatrix} V_{ra} \\ V_{rb} \\ V_{rc} \end{bmatrix}=\begin{bmatrix} R_r & 0 & 0 \\ 0 & R_r & 0 \\ 0 & 0 & R_r \end{bmatrix}\begin{bmatrix} I_{ra} \\ I_{rb} \\ I_{rc} \end{bmatrix}+\frac{d}{dt}\begin{bmatrix} \Phi_{ra} \\ \Phi_{rb} \\ \Phi_{rc} \end{bmatrix}$$

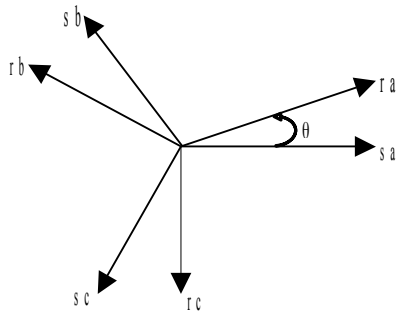
$$L_S=\begin{bmatrix} L_r & M_r & M_r \\ M_r & L_r & M_r \\ M_r & M_r & L_r \end{bmatrix} \tag{3}$$

Couplage Stator/Rotor:

$$M_{sr}(\theta)=M_{rs}(\theta)^T=M_{rs}\,.\begin{bmatrix} \cos\theta & \cos(\theta-\frac{2\pi}{3}) & \cos(\theta+\frac{2\pi}{3}) \\ \cos(\theta+\frac{2\pi}{3}) & \cos\theta & \cos(\theta-\frac{2\pi}{3}) \\ \cos(\theta-\frac{2\pi}{3}) & \cos(\theta+\frac{2\pi}{3}) & \cos\theta \end{bmatrix} \quad (4)$$

$$M_{rs}(\theta)=M_{sr}(\theta)^T=M_{sr}(-\theta)$$

θ désigne l'angle électrique entre une phase du rotor et la phase correspondante du stator:



Bilan de puissance

$$P=I^T.U \quad (5)$$

$$P=I^T.R.I+\frac{d}{dt}\left(\frac{1}{2}.I^T.L.I\right)+\frac{1}{2}I^T.\frac{dL}{dt}.I \quad (III.6)$$

La puissance instantanée se décompose en trios termes:

- puissance dissipée par effet Joule.
- variation totale d'énergie électromagnétique.
- puissance transformée en puissance mécanique ($p_{méca}$).

Puissance mécanique :

$$p_{méca}=\Gamma\Omega=\frac{1}{2}.I^T.\frac{dL}{d\theta}.I.P.\Omega=\frac{d\theta}{dt} \quad (7)$$

Couple moteur :

$$\Gamma = P.\frac{1}{2} I^T \frac{dL}{d\theta}.I \tag{8}$$

Equation mécanique :

$$J.\frac{d\Omega}{dt} + fv.\Omega = \Gamma - C_\Gamma \tag{9}$$

III.2 - Transformation de Concordia

La machine triphasée peut être transformée en une machine biphasée équivalente à l'aide de la transformation de Concordia :

Un vecteur x de grandeur triphasée peut être décrit par les signaux sur chacune des trois phases (abc) : $X_{abc} = [X_a\ X_b\ X_c]^T$. un changement de repère approprié (abc) a un repère (αβh) qui permet de réduire la complexité du système sous certaines hypothèses. Supposant que l 'axe h soit orienté suivant la composante homopolaire du système triphasé (c'est à dire la somme des trois signaux de phase. Lorsque le système triphasé est équilibré, cette composante est nulle, ce qui permet de réduire le système triphasé à un système biphasé (axes α,β).

Sans prise en compte de la composante homopolaire (hypothèse : $X_a + X_b + X_c = 0$) :

$$X_{\alpha\beta} = K_{23}.X_{abc} \quad \text{avec} \quad K_{23} = \sqrt{\frac{2}{3}} \begin{bmatrix} 1 & -\frac{1}{2} & -\frac{1}{2} \\ 0 & \frac{\sqrt{3}}{2} & -\frac{\sqrt{3}}{2} \end{bmatrix} \tag{10}$$

Avec prise en compte de la composante homopolaire :

$$X_{\alpha\beta h} = K_{33}.X_{abc} \quad \text{avec} \quad K_{33} = \sqrt{\frac{2}{3}} \begin{bmatrix} 1 & -\frac{1}{2} & -\frac{1}{2} \\ 0 & \frac{\sqrt{3}}{2} & -\frac{\sqrt{3}}{2} \\ \frac{1}{\sqrt{2}} & \frac{1}{\sqrt{2}} & \frac{1}{\sqrt{2}} \end{bmatrix} \tag{11}$$

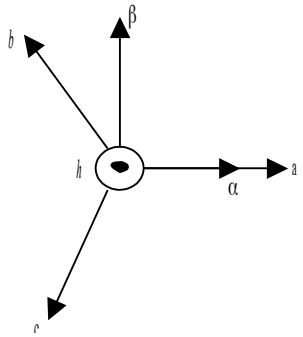


Figure 3 : Transformation de Concordia

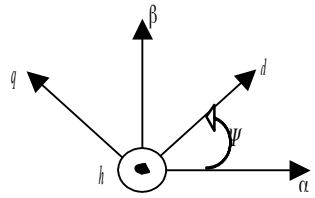


Figure 4 : Rotation

Un repère (dq) quelconque (ou (dqh)) s'obtient par rotation d'un angle Ψ du repère ($\alpha\beta$) (ou ($\alpha\beta h$)). d signifie « direct » et q signifie « quadrature ».

Sans prise en compte de la composante homopolaire :

$$X_{dq} = R(\Psi).X_{\alpha\beta} \qquad \text{avec} \qquad R(\Psi) = \begin{bmatrix} \cos(\Psi) & \sin(\Psi) \\ -\sin(\Psi) & \cos(\Psi) \end{bmatrix} \tag{12}$$

Avec prise en compte de la composante homopolaire :

$$X_{dqh} = \widetilde{R}(\Psi).X_{\alpha\beta h} \quad \text{avec} \quad \widetilde{R}(\Psi) = \begin{bmatrix} \cos(\Psi) & \sin(\Psi) & 0 \\ -\sin(\Psi) & \cos(\Psi) & 0 \\ 0 & 0 & 1 \end{bmatrix} \tag{13}$$

Les changements de repère de ce paragraphe s'appliquent à n'importe quel vecteur de grandeur triphasé. En particulier, ils s'appliquent aussi aux grandeurs triphasées relatives au stator qu'à celle relative au rotor d'une machine asynchrone.

III.3 - Transformation de Park (rotor fictif équivalent fixe)

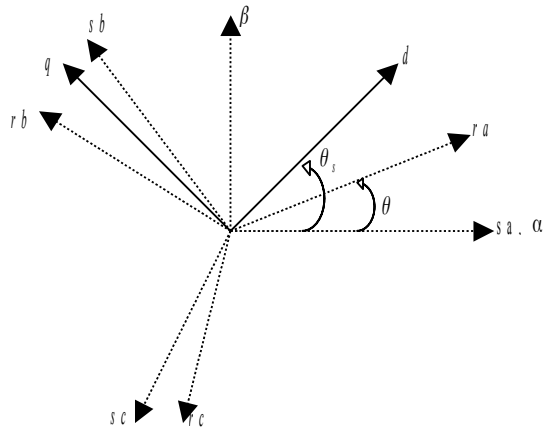


Figure 5 : Repère triphasé fixe par rapport au stator (sa,sb,sc), repère (dq) formant un angle θ_s quelconque par rapport au stator

Pour simplifier d'avantage les expressions électrique et surtout pour que M ne dépende plus de θ, Park a proposé de remplacer le rotor tournant par un rotor équivalent « fixe ». Cela revient à exprimer les grandeurs rotoriques dans le repère (dq) défini par rapport au stator.

- Transformation de Park :

Sans Prise en compte de la composante homopolaire (hypothèse : MAS équilibrée) :

$$X_{dq} = \begin{bmatrix} T(\theta_s) & 0 \\ 0 & T(\theta_s - \theta) \end{bmatrix} \cdot X_{abc} \qquad \text{avec :}$$

$$X_{dq} = \begin{bmatrix} X_{s,dq} \\ X_{r,dq} \end{bmatrix}, \quad X_{abc} = \begin{bmatrix} X_{s,abc} \\ X_{r,abc} \end{bmatrix} \qquad \text{et} \qquad T(\Psi)=R(\Psi).K_{23}$$

(14)

Avec prise en compte de la composante homopolaire :

$$X_{dqh} = \begin{bmatrix} \tilde{T}(\theta_s) & 0 \\ 0 & \tilde{T}(\theta_s - \theta) \end{bmatrix} \cdot X_{abc} \qquad \text{avec :}$$

$$X_{dqh} = \begin{bmatrix} X_{s,dqh} \\ X_{r,dqh} \end{bmatrix}, \quad X_{abc} = \begin{bmatrix} X_{s,abc} \\ X_{r,abc} \end{bmatrix} \qquad et \quad \tilde{T}(\Psi) = \tilde{R}(\Psi).K33 \qquad (15)$$

Remarque 1 :

$\tilde{T}$ est orthonormé ($\tilde{T}$ est une matrice unité d'ordre 3 : $\tilde{T}\ \tilde{T}^T = I_3$) ce qui permet de conserver la puissance instantanée et d'opérer la même transformation sur les tensions que sur les courants. Les axes du repère (dqh) sont orthogonaux.

Remarque 2 :

$$T(\psi) = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \end{bmatrix} \cdot \tilde{T}(\psi)$$

$\omega_S = \dfrac{d\theta_s}{dt}$ Représente la pulsation du repère (dq) par rapport au stator.

III.4 - Obtention d'une représentation d'état dans un repère (dq)

Préliminaires :

$$T(\Psi).L_s = L_s'.T(\Psi) \qquad \text{avec } L_s' = L_s - M_s \tag{16}$$

$$T(\Psi).L_r = L_r'.T(\Psi) \qquad \text{avec } L_r' = L_r - M_r \tag{17}$$

$$T(\Psi).R_{ss} = R_s.T(\Psi) \tag{18}$$

$$T(\Psi).R_{rr} = R_r.T(\Psi) \tag{19}$$

$$T(\Psi).M_{rs}(\theta) = M.T(\Psi - \theta) \qquad \text{avec } M = \frac{3}{2} \cdot M_{rs} \tag{20}$$

$$\tilde{T}(\psi) \cdot M_{rs}(\theta) = \begin{bmatrix} M & 0 & 0 \\ 0 & M & 0 \end{bmatrix} \cdot \tilde{T}(\psi - \theta) \tag{21}$$

$$\frac{dT(\Psi)}{dt} = \frac{d\Psi}{dt} \cdot R\left(\frac{\pi}{2}\right) \cdot T(\Psi) \quad \text{et} \quad R\left(\frac{\pi}{2}\right) = \begin{bmatrix} 0 & 1 \\ -1 & 0 \end{bmatrix} \tag{22}$$

$$\widetilde{T}(\varphi)\cdot\widetilde{T}(\psi)^{-1} = \begin{bmatrix} R(\varphi-\psi) & 0 \\ 0 & 1 \end{bmatrix} \quad \text{(rappel : } \widetilde{T}(\varphi)^{T} = \widetilde{T}(\psi)^{-1} \text{)} \tag{23}$$

Expression des flux statoriques et rotoriques dans un repère (*dq*) :

$$\Phi_{qd} = \begin{bmatrix} T(\theta_S) & 0 \\ 0 & T(\theta_S-\theta) \end{bmatrix}\cdot\Phi_{abc} \quad \text{avec} \quad \Phi_{abc} = \begin{bmatrix} L_s & M_{sr}(\theta) \\ M_{rs}(\theta) & L_r \end{bmatrix}\cdot\begin{bmatrix} I_{s,abc} \\ I_{r,abc} \end{bmatrix} \tag{24}$$

donne :

$$\begin{bmatrix} \Phi_{sd} \\ \Phi_{sq} \\ \Phi_{rd} \\ \Phi_{rq} \end{bmatrix} = \begin{bmatrix} L_s & 0 & M & 0 \\ 0 & L_s & 0 & M \\ M & 0 & L_r & 0 \\ 0 & M & 0 & L_r \end{bmatrix}\cdot\begin{bmatrix} I_{sd} \\ I_{sq} \\ I_{rd} \\ I_{rq} \end{bmatrix} \tag{25}$$

Expression des tensions statoriques et rotoriques dans un repère (*dq*) :

$$\Phi_{qd} = \begin{bmatrix} T(\theta_T) & 0 \\ 0 & T(\theta_s-\theta) \end{bmatrix}\cdot U_{abc} \quad \text{avec} \quad U = (R\cdot I) + \frac{d\Phi}{dt} \tag{26}$$

D'après (26) dans le repère (abc) et en utilisant (22) on obtient :

$$\left\{\begin{array}{l} U_{sd} = R_s\cdot I_{sd} + \dfrac{d\Phi_{sd}}{dt} - \omega_s\cdot\Phi_{sq} \\ U_{sq} = R_s\cdot I_{sq} + \dfrac{d\Phi_{qd}}{dt} - \omega_s\cdot\Phi_{sq} \\ U_{rd} = 0 = R_r\cdot I_{rd} + \dfrac{d\Phi_{rd}}{dt} - (\omega_s - p\cdot\Omega)\cdot\Phi_{rq} \\ U_{rq} = 0 = R_r\cdot I_{rq} + \dfrac{d\Phi_{rq}}{dt} - (\omega_s - p\cdot\Omega)\cdot\Phi_{sq} \end{array}\right\} \tag{27}$$

Dans le cas d'un moteur asynchrone à cage, les tensions rotoriques sont nulles.

Expression du couple moteur à partir -de grandeurs électriques dans un repère(*dq*) :

$$\Gamma = P.\frac{1}{2}.I_{abc}{}^{T}\cdot\left(\frac{dL}{d\theta}\right)_{abc}.I_{abc} = P.\frac{1}{2}.I_{dqh}{}^{T}\cdot\left(\frac{dL}{d\theta}\right)_{dqh}.I_{dqh} \tag{28}$$

A partir de :

$$\left(\frac{dL}{d\theta}\right)_{dqh} = \begin{bmatrix} \widetilde{T}(\theta_s) & 0 \\ 0 & \widetilde{T}(\theta_s-\theta) \end{bmatrix} \cdot \left(\frac{dL}{d\theta}\right)_{abc} \cdot \begin{bmatrix} \widetilde{T}(\theta_s) & 0 \\ 0 & \widetilde{T}(\theta_s-\theta) \end{bmatrix}^{-1} \tag{29}$$

Et en remarquant que :

$$\frac{dM_{sr}(\theta)}{d\theta} = M_{sr}\left(\theta+\frac{\pi}{2}\right) \tag{30}$$

On montre (en utilisant, entre autres, (23)) :

$$\Gamma = P.M.\left(I_{sq}.I_{rd} - I_{sd}.I_{rq}\right) \tag{31}$$

En utilisant (25), le couple moteur peut être exprimé avec d'autres grandeurs électriques. Par exemple :

$$\Gamma = \frac{P.M}{L_r}.\left(\Phi_{rd}.I_{sq} - \Phi_{rq}I_{sd}\right) \tag{32}$$

$$\Gamma = P.\left(\Phi_{sd}.I_{sq} - \Phi_{sq}I_{sd}\right) \tag{33}$$

Représentations d'état dans un repère (*dq*) :

En combinant (25) avec (27) à (30), plusieurs représentations d'état peuvent être obtenues. (Ces modèles sont mis en ouvre pour analyser les comportements de la machine asynchrone ou pour mettre en place les différentes fonctions des commandes).Nous en donnons deux (En choisissant la transformation de Parck où le système d'axe (*d,q)* est fixe par rapport au stator (repère de Clarck) (θ_s =0)):

Etat constitué des courants statoriques et rotoriques et de la vitesse :

$$x = \begin{bmatrix} I_{sd} & I_{sq} & I_{\mathrm{rd}} & I_{\mathrm{rq}} & \Omega \end{bmatrix}^T \ , \ u = \begin{bmatrix} U_{sd} & U_{sq} \end{bmatrix}^T \tag{34}$$

$$\frac{dx(t)}{dt} = A(x(t)) + B.u(t)$$ avec :

$$A(x) = \begin{pmatrix} \xi<(-L_r R_s \cdot x_1 + M^2 P \cdot x_5 \cdot x_2 + MR_r \cdot x_3 + PL_{rM} \cdot x_5 \cdot x_4) \\ \xi(-M^2 P \cdot x_5 \cdot x_1 - L_r R_s \cdot x_2 - P L_r M \cdot x_5 \cdot x_3 + M R_r \cdot x_4) \\ \xi(M R_s \cdot x_1) - P L_s M \cdot x_5 \cdot x_2 - L_s R_r \cdot x_3 - L_s L_r P \cdot x_5 \cdot x_4 \\ \xi(P L_s \cdot M \cdot x_5 \cdot x_1 + M R_s \cdot x_2 + L_s L_r P \cdot x_5 \cdot x_3 - L_s R_r \cdot x_4) \\ \frac{1}{J}(P M \cdot x_2 \cdot x_3 - P M \cdot x_1 \cdot x_4 - f v \cdot x_5 - C_r) \end{pmatrix} , \quad B = \begin{bmatrix} \xi L_r & 0 \\ 0 & \xi L_r \\ -\xi M & 0 \\ 0 & -\xi M \\ 0 & 0 \end{bmatrix}$$

État constitué des flux statoriques et rotoriques et de la vitesse :

$$x = \begin{bmatrix} \Phi_{sd} & \Phi_{sq} & \Phi_{rd} & \Phi_{rq} & \Omega \end{bmatrix}^T , \; u = \begin{bmatrix} U_{sd} & U_{sq} \end{bmatrix} \tag{35}$$

$$\frac{dx(t)}{dt} = A(x(t)) + B.u(t) \qquad \text{avec :}$$

$$A(x) = \begin{pmatrix} -\xi \cdot R_s(L_r \cdot x_1 - M P \cdot x_3) \\ -\xi \cdot R_s(L_r \cdot x_2 - M \cdot x_4) \\ -\xi \cdot R_r(-M \cdot x_1 + L_s \cdot x_3 - P \cdot x_4 \cdot x_5) \\ -\xi \cdot R_r(-M \cdot x_2 + L_s \cdot x_4) \\ \frac{1}{J}[P M \xi(-x_1 \cdot x_4 + x_2 \cdot x_3) - f v \cdot x_5 - C_r] \end{pmatrix} , \quad B = \begin{bmatrix} \xi L_r & 0 \\ 0 & \xi L_r \\ -\xi M & 0 \\ 0 & -\xi M \\ 0 & 0 \end{bmatrix}$$

$$\zeta = 1/(L_r . L_s - M^2)$$

IV - COMMANDE DE LA MACHINE ASYNCHRONE

Les principaux modes de contrôle de la machine asynchrone sont :

- le contrôle scalaire : qui consiste à imposer à la machine le module de la tension ou du courant statorique ainsi que la fréquence statorique ; préférable pour des performances moyennes et pour des fonctionnements en survitesse.

- le contrôle vectoriel : plus complexe à réaliser que le précédent mais qui permet des performances plus élevées que le contrôle scalaire ; car il est le seul à permettre un fonctionnement satisfaisant à très basses vitesses et à l'arrêt.

- le contrôle direct : basé sur la détermination « directe » de la séquence de commande appliquée aux interrupteurs d'un onduleur de tension.

IV.1 - Robustesse d'un contrôle vectoriel de la machine asynchrone

Les nouvelles applications industrielles nécessitent des variateurs de vitesse ayant des hautes performances dynamiques, une bonne précision en régime permanent, une haute capacité de surcharge sur toute la plage de vitesse et une robustesse aux différentes perturbations.

Ces dernières années plusieurs techniques ont été développées pour permettre au variateur asynchrone d'atteindre ces performances. Le contrôle vectoriel, qui permet un découplage entre les variables de commande, reste le plus utilisé vu les performances dynamiques élevées qu'il offre pour une large gamme d'applications. Pour surmonter la sensibilité de cette stratégie aux variations des paramètres du système et aux perturbations externes, des techniques modernes d'automatique telles que le contrôle robuste, le contrôle adaptatif et le contrôle à structure variable ainsi que les estimateurs à base du filtre de Kalman ont été appliqués.

IV.2 - Principe de la commande vectorielle

Le but de la commande vectorielle est d'arriver à commander la machine asynchrone comme une machine à courant continu à excitation indépendante où il y a un découplage naturel entre la grandeur commandant le flux, le courant d'excitation, et celle liée au couple, le courant d'induit. Ce découplage permet d'obtenir une réponse très rapide du couple.

En parlant d'orientation du flux, c'est plutôt le système d'axe (d, q) que l'on oriente de manière à ce que l'axe d soit en phase avec le flux, c'est-à-dire :

$$\begin{cases} \phi_d = \phi \\ \phi_q = 0 \end{cases} \tag{36}$$

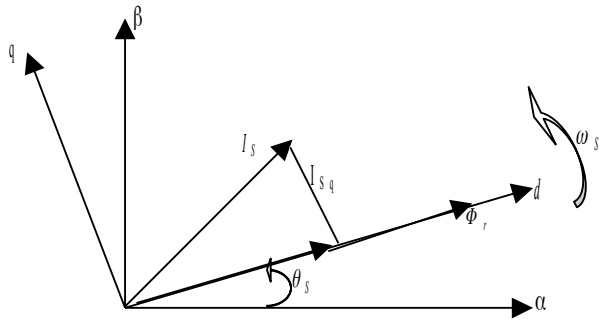


Figure 6 : Principe de la commande vectorielle

Si le flux rotorique est orienté sur l'axe *d* d'un repère lié au champ tournant, alors

ϕ_{rq}= 0. on a alors :

$$\Gamma = \frac{P.M}{L_r}.(\Phi_{rd}.I_{sq}) \tag{37}$$

$$\tau.\frac{d\Phi_{rd}}{dt} + \Phi_{rd} = M.I_{sd} \tag{38}$$

Ainsi, I_{sd} fixe le flux et I_{sq} le couple. On retrouve un découplage entre inducteur et induit similaire à celui d'une machine à courant continu (le courant de l'inducteur fixe le flux, le courant d'induit fixe le couple pour un flux donné).

En tenant compte de ϕ_{rq} =0, les quatre équations (34) de la machine deviennent (39) à (42).

$$U_{sd} = \tilde{\sigma}.L_s.\left(\frac{dI_{sd}}{dt} + \gamma.I_{sd} - \omega_s.I_{sq} - \frac{\kappa}{\tau_r}.\Phi_{rd}\right) \tag{39}$$

$$U_{sq} = \tilde{\sigma}.L_s.\left(\frac{dI_{sq}}{dt} + \omega_s.I_{sd} + \gamma.I_{sq} + \omega.\kappa.\Phi_{rd}\right) \tag{40}$$

$$\tau_r.\frac{d\Phi_{rd}}{dt} + \Phi_{rd} = M.I_{sd} \tag{41}$$

$$\omega_s = \omega + \frac{M}{\tau_r}.\frac{I_{sq}}{\Phi_{rd}} \tag{42}$$

(ω_s : pulsation du repère tournant induite par ϕ_{rq} =0)

Ces équations montrent que U_{sd} influe à la fois sur I_{sd} et I_{sq}, donc sur le flux et le couple. Il en est de même pour U_{sq}. on est alors amené à réaliser un découplage ; deux nouvelles variables U_d et U_q sont définies telles que U_d n'agit que sur I_{sd} (donc que sur le flux) et U_q n'agit que sur I_{sq} (donc sur le couple). Une méthode de découplage possible consiste à compenser les termes de découplage à l'aide de leurs estimées (E_d et Eq) :

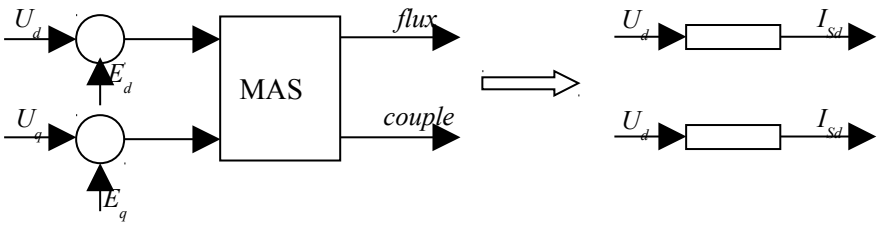


Figure 7 : méthode de découplage

Un régulateur de flux produit l'action U_d et un régulateur de couple l'action U_q. la consigne de flux est souvent une loi fonction de vitesse.

V - TYPES DE DÉFAUTS DE LA MACHINE ASYNCHRONE

Bien que la machine asynchrone ait la réputation d'être robuste, elle peut présenter comme tout autre machine électrique, des défaillances d'ordre électrique ou mécanique, donc, il n'est pas exclu qu'il peut y apparaître des défauts tant au niveau du stator que du rotor.

Les défauts typiques des machines asynchrones, de fortes puissances, sont :

– un défaut au niveau du rotor qui serait une rupture totale ou partielle d'une barre au niveau de l'anneau de la cage d'écureuil, une rupture d'une portion d'anneau,

– un défaut de contact balai - bague dans le cas d'un rotor bobiné,

– un défaut d'alignement prononcé par une irrégularité de l'entrefer qui induirait des frottements, donc des préjudices sur le bobinage du stator,

– un défaut d'isolation électrique au niveau du bobinage du stator suite à un vieillissement prématuré du au milieu de fonctionnement hostile ou non et à la façon dont est alimenté le moteur,

- une dégradation par usure prématurée ou non des roulements à billes.

VI - CONCLUSION

Dans ce chapitre, nous avons présenté un modèle de la machine asynchrone qui permet la réduction du nombre de grandeurs qu'on a besoin de connaître pour pouvoir simuler le fonctionnement de la machine.

Malgré les qualités du moteur asynchrone, ce dernier peut subir des défauts de type mécanique ou électrique. Dans ce mémoire on s'intéressera particulièrement aux défauts électriques.

Dans la suite de ce mémoire, nous exposerons notre étude en simulation pour valider toute l'étude théorique précédente.

F - Étude en Simulation

I - INTRODUCTION

Dans ce chapitre, nous nous intéressons à l'étude en simulation de la méthode de détection et localisation de défauts basée sur l'approche neuro-floue détaillée dans le chapitre 2. L'application de ce schéma neuro-flou concerne le diagnostic des défauts de type capteur et paramétrique pouvant affecter le moteur asynchrone.

II - PRÉSENTATION DE LA MAS

- Le modèle mathématique :

D'après le chapitre précédent, nous avons modélisé la machine asynchrone sous forme d'état, qui est composé de cinq équations non linéaires, où on a choisi le vecteur d'état de la façon suivante: .

$x = \begin{bmatrix} I_{sd} & I_{sq} & I_{rd} & I_{rq} & \Omega \end{bmatrix}^T$.

Le vecteur d'entrée u est donné par $u = \begin{pmatrix} U_{sd} & U_{sq} \end{pmatrix}^T$.

Le vecteur de sortie y est choisit par $y = \begin{pmatrix} I_{sd} I_{sq} & \Omega \end{pmatrix}^T$.

Ce modèle est d'écrit par : $\begin{cases} \dot{x} = A(x(t)) + B.u(t) \\ y(t) = C.x(t) \end{cases}$

Avec : $A(x) = \begin{pmatrix} \xi(-L_r R_r \cdot I_{sd} + M^2 P \cdot \Omega \cdot I_{sq} + M R_r \cdot Ird + P L_r M \cdot \Omega \cdot I_{rq}) \\ \xi(-M^2 P \cdot \Omega \cdot I_{sd} - L_r R_s \cdot I_{sq} - P L_r M \cdot \Omega \cdot I_{rd}) \\ \xi(M R_s \cdot I_{sd} - P L_s M \cdot \Omega \cdot I_{sq} - L_s R_r \cdot I_{rd} - L_s L_r P \cdot \Omega \cdot I_{rd}) \\ \xi(P L_s \cdot \Omega \cdot I_{sd} + M R_s \cdot I_{sq} + L_s L_r P \cdot \Omega \cdot I_{rd} - L_s R_r \cdot I_{rq}) \\ \frac{1}{J}(P M \cdot I_{sq} \cdot I_{rd} - P M \cdot I_{sd} \cdot I_{rq} - f v \cdot \Omega - C_r) \end{pmatrix}$

$$B = \begin{bmatrix} \xi L_r & 0 \\ 0 & \xi L_r \\ -\xi M & 0 \\ 0 & -\xi M \\ 0 & 0 \end{bmatrix}, \quad C = \begin{pmatrix} 1 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 \end{pmatrix}$$

Avec : $\zeta = 1/(L_r . L_s - M^2)$

I_{sd} , I_{sq} , I_{rd} , I_{rq} Sont les courants de Park stator et rotor dans le repère (d, q) lié au stator.

Ω : La vitesse mécanique du moteur ;

Les valeurs nominales des paramètres physiques de la MAS sont :

Résistance statorique Rs = 10 (Ω) ;

Résistance rotorique Rr = 5 (Ω) ;

Inductance statorique Ls = 500 (*mH*) ;

Inductance rotorique Lr = 430 (*mH*) ;

Inductance mutuelle M = 430 (*mH*) ;

Inertie J = 0.08 ($kg.m^2$) ;

Couple résistant Cr = 0 (*N.m*);

Frottement visqueux *fv* = 0 (*N.m.s/rad*) ;

Nombre de paires de pôles p=2

III - FONCTIONNEMENT NORMAL DE LA MAS (SANS DÉFAUTS)

Les entrées sinusoïdales triphasées d'amplitudes 400V et de fréquence 50 Hz, tranformées dans le repère (*d, q)*, sont montrées en figure 1. Le couple électromagnétique Cr et les sorties du système I*sd*, I*sq* et Ω sont montrées en figure 2.

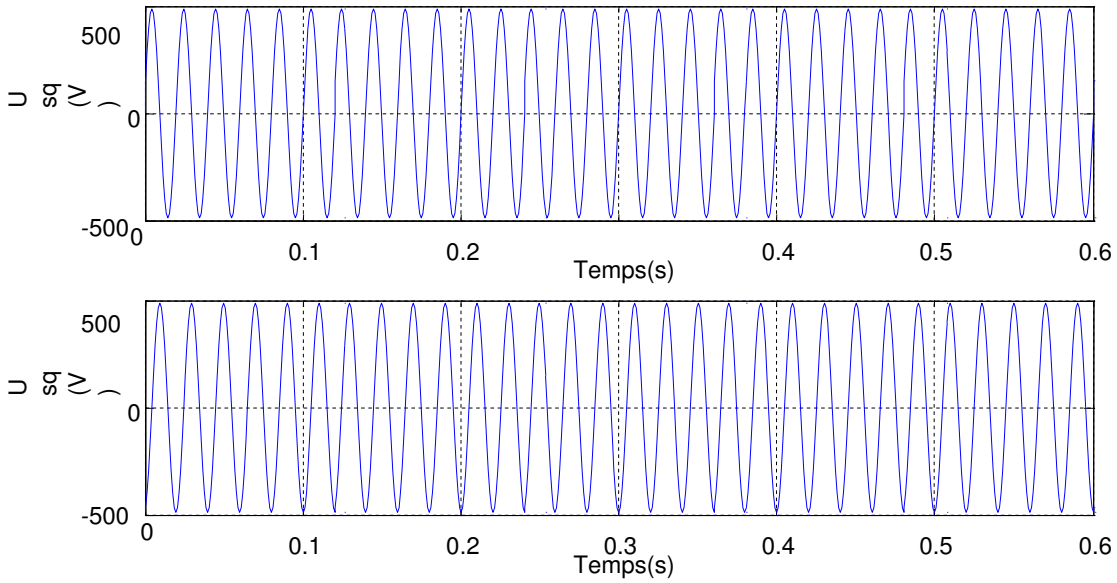


Figure 1 : Les entrées du système

Sachant que la pulsation $\omega = 2\pi 50$ rad/s

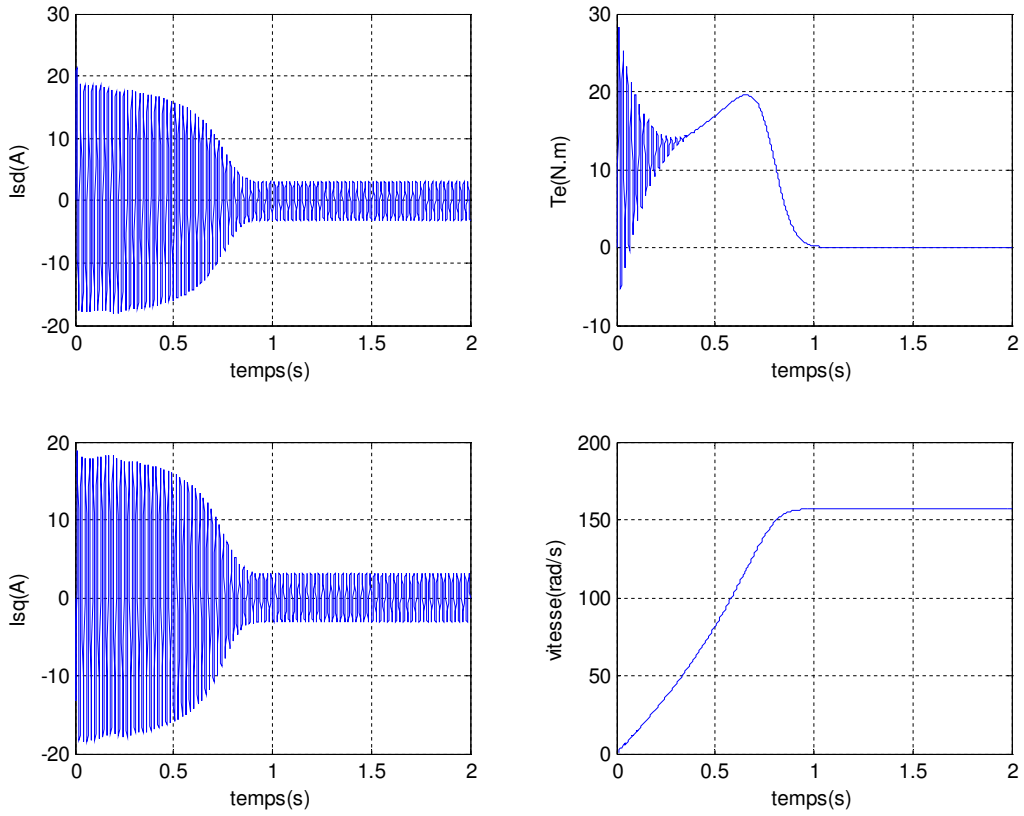


Figure 2 : Les sorties du système et le couple magnétique

N.B: La simulation de la MAS a été réalisée avec un pas de simulation **Te = 1 msec**.

IV - IDENTIFICATION PAR RÉSEAU DE NEURONES

Pour identifier notre modèle non linéaire, on a opté pour un réseau de neurones de type PMC basé sur la structure NNARX, après avoir essayé d'autres structures (voir chapitre III).

IV.1 - Architecture du réseau de neurones

Après plusieurs essais on a choisi l'architecture du réseau de neurones suivante :

- **quatre** neurones dans la couche cachée ;
- **trois** neurones dans la couche de sortie ;
- **cinq** neurones dans la couche d'entrée.

L'architecture de se réseau est illustrée sur la figure 3 :

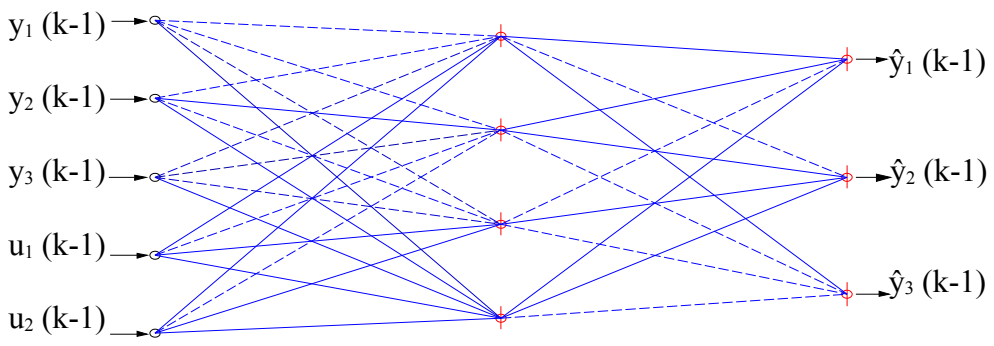


Figure 3 : Architecture du réseau

Les sorties ŷ du réseau sont définies par la relation suivante:

$$\hat{y}_i = \psi_i \cdot (Z_i \cdot h + b_i) \qquad i=1...3 \tag{1}$$

Avec

$$h = \phi\,(w_1 y_1(k-1) + w_2 y_2(k-1) + w_3 y_3(k-1) + w_4 u_1(k-1) + w_5 u_2(k-1)). \tag{2}$$

Où :

- w_i sont les poids reliant la couche cachée à la couche d'entrée.
- Z_i sont les poids reliant la couche cachée à la couche de sortie.
- ψ est une fonction d'activation de type sigmoïde et ϕ une fonction d'activation de type linéaire.
- b est le biais.

IV.2 - Apprentissage

Les poids w_i ,Z_i et les biais b_i sont aléatoirement choisis par l'algorithme de Levenberg Marquardt.

Suite à une série d'essais l'erreur quadratique évolue selon :

- **7.619 10^{-1}** au bout de onze itérations avec une architecture de trois neurones non linéaires de type **tanh** dans la couche cachée et trois neurones linéaires dans la couche de sortie.
- **1.119 10^{-14}** au bout de vingt et une itérations avec une architecture de six neurones hybrides, trois linéaires et trois non-linéaires de type **tanh** dans la couche cachée et trois neurones linéaires dans la couche de sortie.
- **3.215 10^{-15}** au bout de six itérations avec une architecture de quatre neurones linéaires dans

la couche cachée et trois neurones linéaires dans la couche de sortie.

On remarque que la dernière architecture est la meilleure, car la valeur de l'erreur quadratique est la plus petite et a été obtenue au bout de six itérations seulement, ce qui explique que les sorties estimées convergent vers les sorties réelles, et c'est ainsi qu'on a opté pour cette architecture dans la suite du projet.

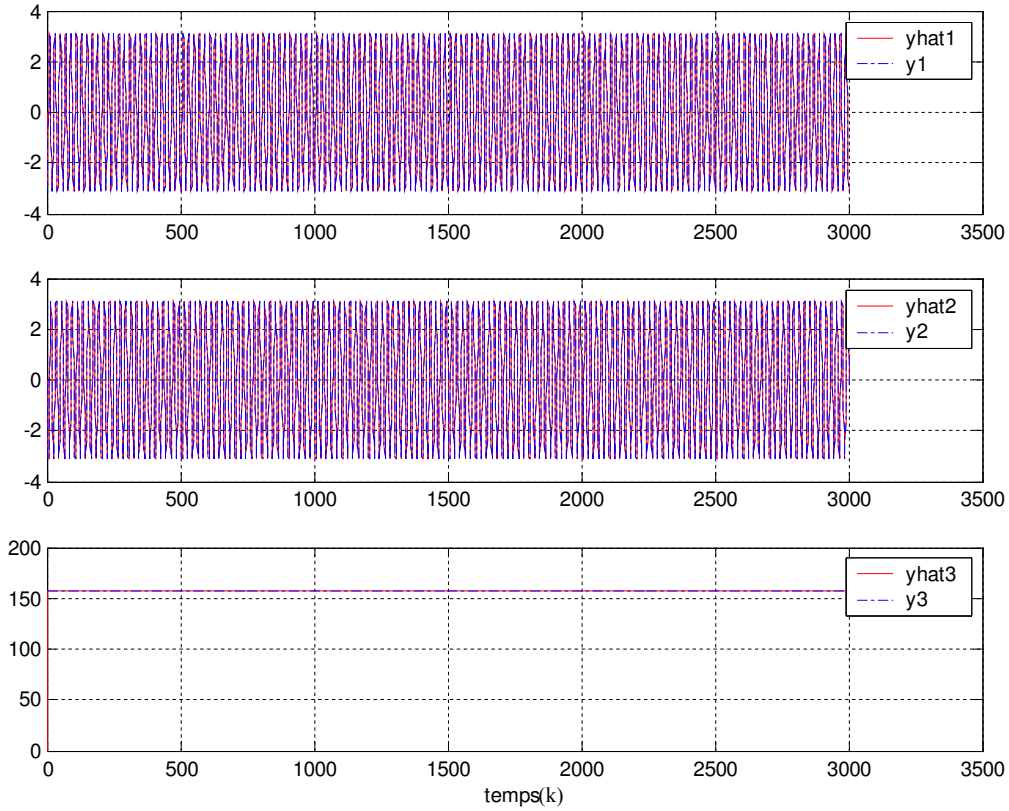


Figure 4 : Les sorties réelles et estimées

IV.3 - Génération des résidus sans défauts

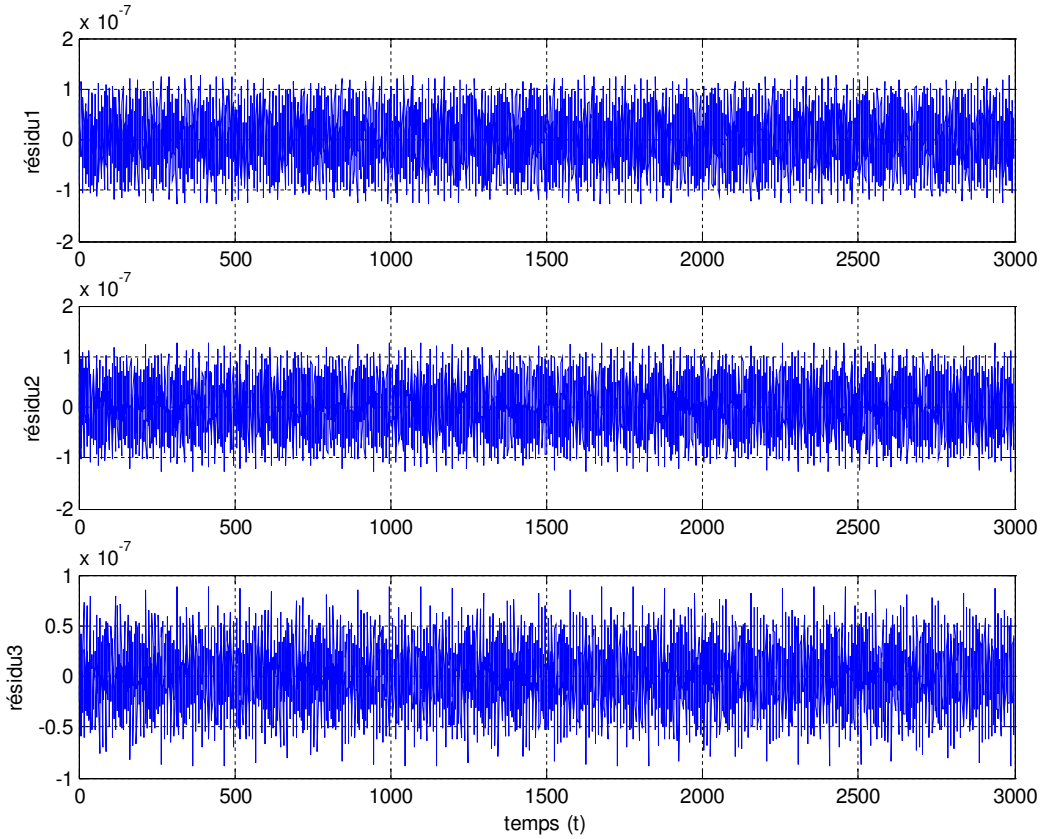


Figure 5 : Les résidus en absence de défauts

D'après la figure 5 les erreurs entre les sorties réelles et les sorties estimées sont de l'ordre de 10^{-7}.

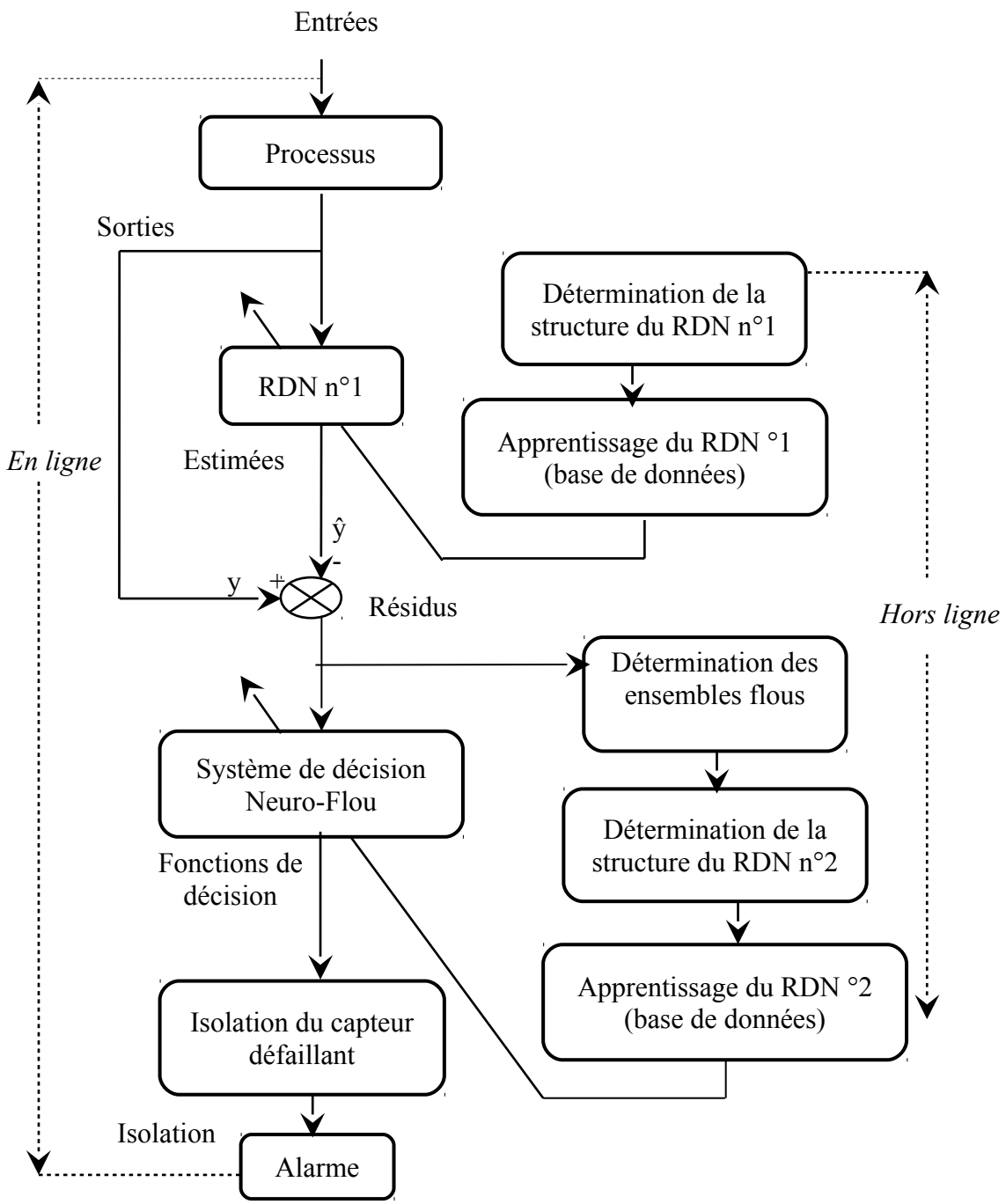


Figure 6 : Procédure générale pour la détection et localisation des capteurs défaillants

N.B: toutes les figures sont en fonction du temps discret **$k = t / T_e$**

V - DÉTECTION ET LOCALISATION DES DÉFAUTS CAPTEURS

V.1 - Analyse des résidus par un schéma neuro-flou

- **F*uzzification* des résidus :** on a choisi pour chaque résidu trois fonctions d'appartenance : deux fonctions de type trapézoïdale et une de type triangulaire, pour le choix de leurs paramètres on a effectué de nombreux essais en présence de plusieurs défauts différents.

  - ***Résidu 1*** :

```
N1= [-1 -1 -2.001e-4 -1.998e-4]
Z1= [-2e-4 0 6e-3]
P1= [5.998e-3 6.001e-3 1 1]
```

  - ***Résidu 2*** :

```
N2= [-1 -1 -2.001e-4 -1.998e-4]
Z2= [-2e-4 0 1e-3]
P2= [0.998e-3 1.001e-3 1 1]
```

  - ***Résidu 3*** :

```
N2= [-2 -2 -5.001e-5 -4.998e-5]
Z2= [-5e-5 0 0.022]
P2= [0.021998 0.022001 1 1]
```

- **Inférence :**

Dans cette étape nous allons analyser les résidus flous obtenus précédemment par un réseau de neurones récurrent de type PMC.

On introduit à l'entrée du système les résidus fuzzifiés, qui contiennent les neuf fonctions d'appartenances et les décisions précédentes [de l'instant (k-1)] et en sortie les décisions, ces dernières sont prises par la boucle de retour pour établir un effet mémoire sur la décision.

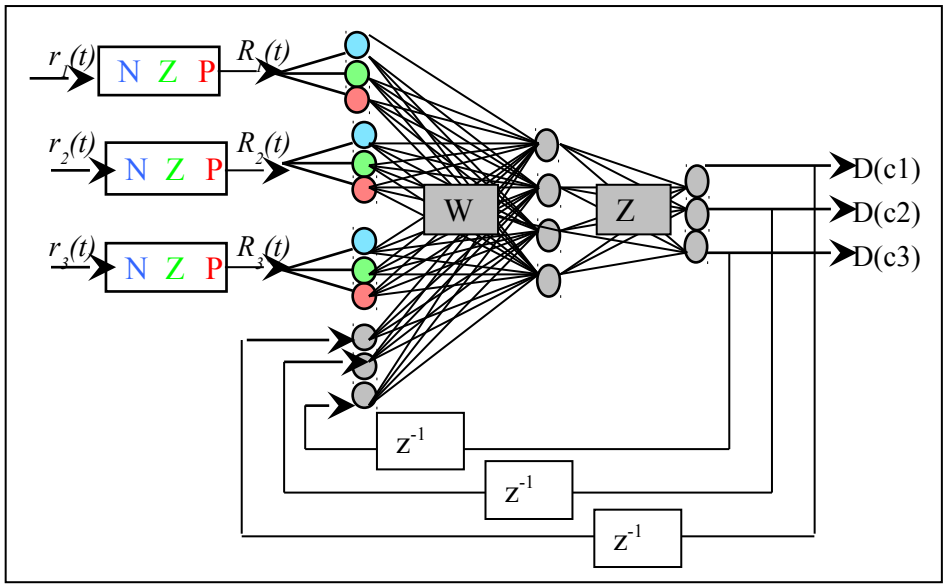


Figure 7 : Architecture du Réseau

- **Structure du réseau** :

Ce réseau est constitué de : douze neurones dans la couche d'entrée représentant les entrées des différents états possibles des résidus à un instant (k) (après fuzzification) et les décisions $D(c_i)$ avec i=1 : 3 à un instant (k-1), quatre neurones dans la couche cachée et trois neurones dans la couche de sortie.

La fonction non linéaire $D(c_i)$ de la sortie du réseau s'obtient par la relation suivante :

$$D(c_i)=\phi_i(Z*h_i+z0) \quad (3)$$

Où

$$h_i=\phi(W*x_i+w0) \quad (4)$$

Avec :

$D(f_i)$: Fonction de décision ;

Z : matrice des poids reliant les neurones de la couche cachée à celles de la couche de sortie ;

$z0$, $w0$: les vecteurs de biais ;

ϕ : représente la fonction sigmoïde qui transforme l'intervalle $[-\infty,+\infty]$ en $[0,1]$;

h_i : représente la sortie de la couche cachée ;

W : matrice des poids reliant les neurones de la couche cachée à celles de la couche d'entrée ;

x_i : représente le stimulus (l'entrée du RDN) qui est les résidus flous et les décisions précédentes.

- **Apprentissage du réseau** :

L'adaptation des poids est choisit par l'algorithme de la rétro propagation (Annexe).

Suite à plusieurs essais l'erreur quadratique est égale à $\mathbf{10^{-2}}$ au bout de 3204 itérations. Voir (Fig. 8)

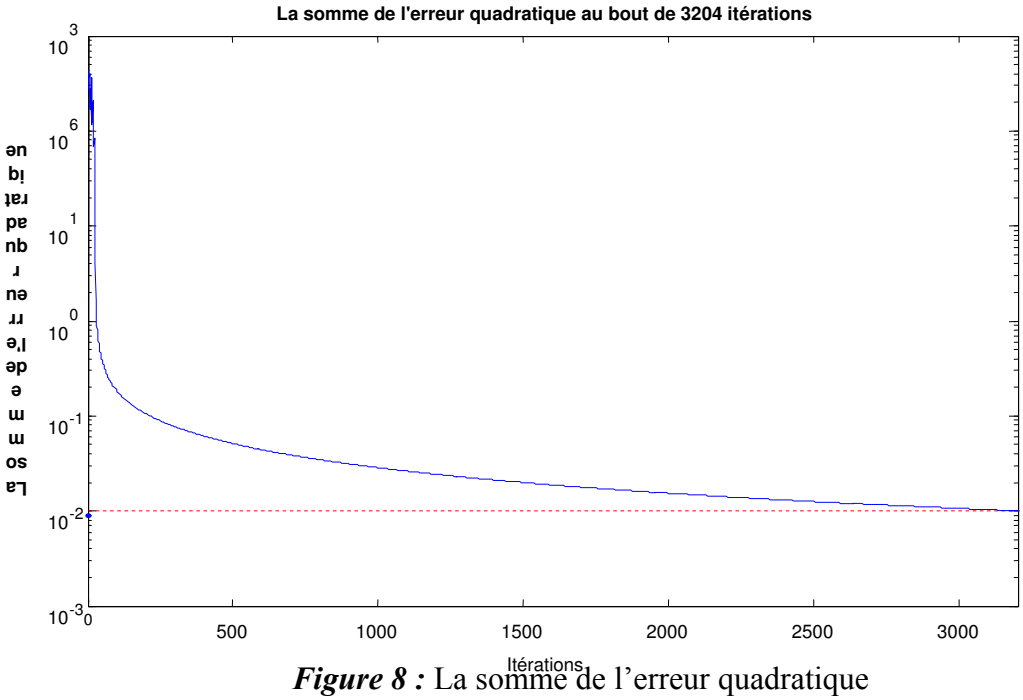


Figure 8 : La somme de l'erreur quadratique

Après plusieurs essais en simulation avec la présence des différents défauts des capteurs, nous avons pu différentier entre ces défauts pour chaque capteur.

Le tableau suivant résume la base d'apprentissage utilisée :

N=°	Résidus 1			Résidus 2			Résidus 3			Décision		
	N1	Z1	P1	N2	Z2	P2	N3	Z3	P3	D1	D2	D3
1	0	1	0	0	1	0	0	1	0	0	0	0
2	0	0	1	0	1	0	0	1	0	1	0	0
3	0	1	0	0	0	1	0	1	0	0	1	0
4	1	0	0	1	0	0	0	0	1	0	0	1
5	0	0	1	0	0	1	0	1	0	1	1	0
6	1	0	0	0	0	1	0	0	1	0	1	1
7	0	0	1	1	0	0	0	0	1	1	0	1
8	0	0	1	0	0	1	0	0	1	1	1	1

Table 1 *:* Base d'apprentissage

<u>NB</u> : Présence de défaut : 1.

Absence de défaut : 0.

V.2 - Résultats de la surveillance

- **Essai 1**: en absence de défauts.

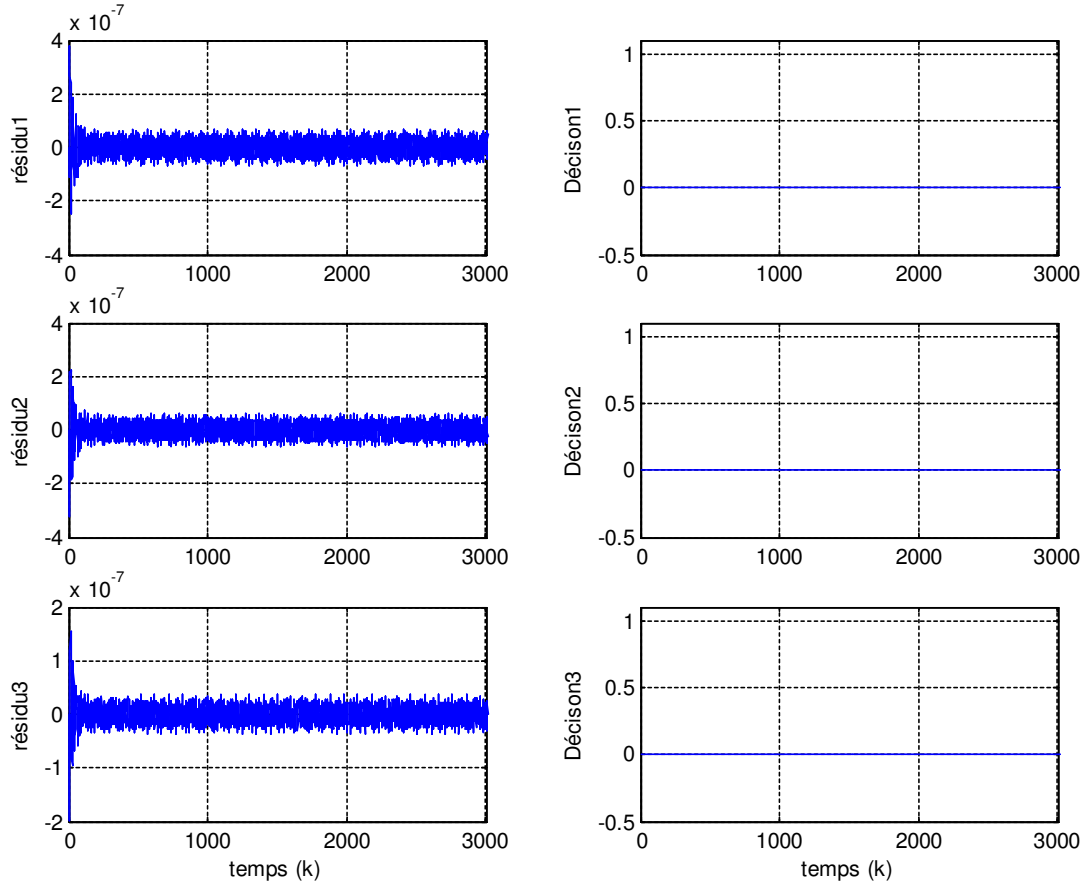


Figure 9 : Résidus et décisions en absence de défauts

- **Essai 2 :** en présence de défauts sur le capteur 1.

Type de défaut biais.

N° du capteur	Instant du défaut	Amplitude du défaut
Capteur N=°1	0.5 s.	0.03

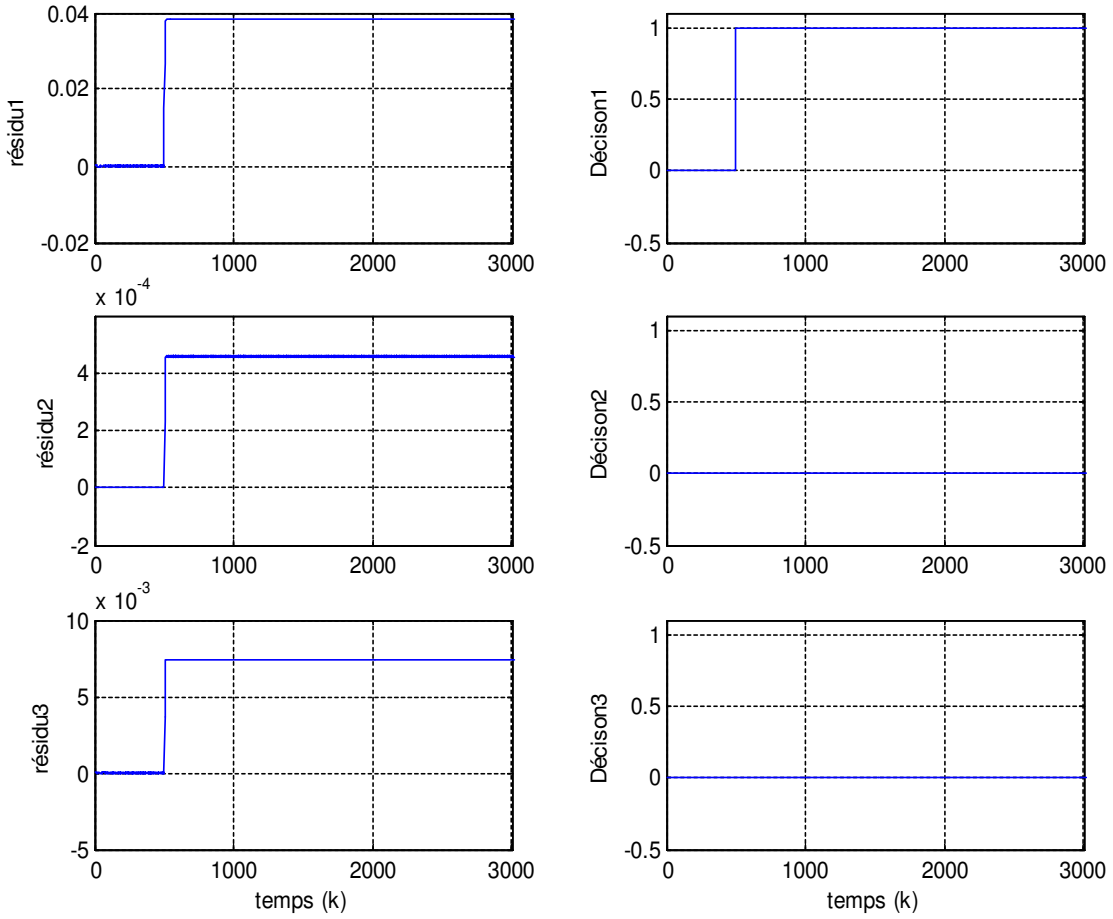


Figure 10 : Résidus et décisions en présence de défauts sur le capteur1

- **Essai 3 :** en présence de défauts sur le capteur 2.

Type de défaut biais.

N° du capteur	Instant du défaut	Amplitude du défaut
Capteur N=°2	0.5s.	0.02

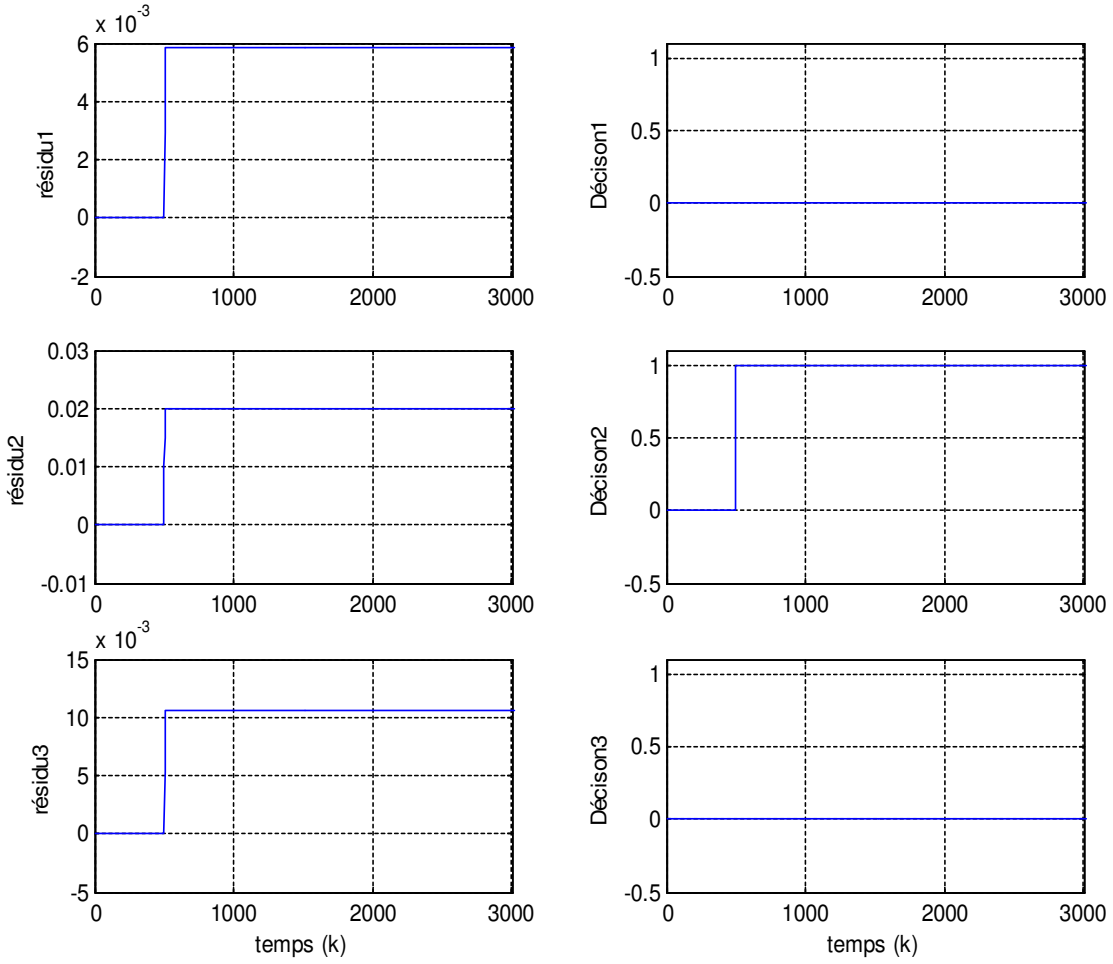


Figure 11 : Résidus et décisions en présence de défauts sur le capteur 2

- **Essai 4 :** en présence de défauts sur le capteur 3.

Type de défaut biais.

N° du capteur	Instant du défaut	Amplitude du défaut
Capteur N=°3	0.5s.	0.09

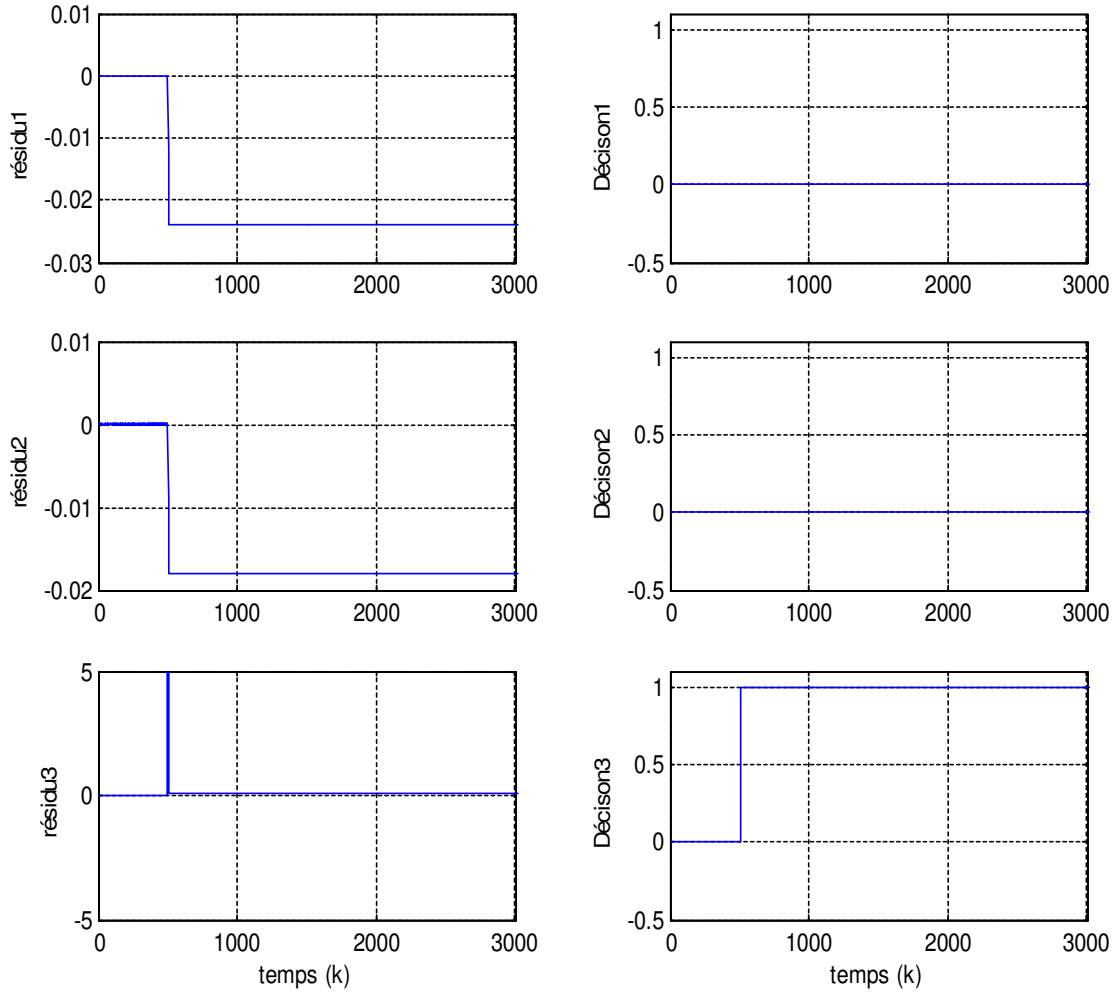


Figure 12 : Résidus et décisions en présence de défauts sur le capteur 3

- **Essai 5 :** en présence de défauts sur le capteur 1 et le capteur 2.

Les amplitudes et les instants des défauts capteurs sont présentés dans le tableau suivant :

N° du capteur	Instant du défaut	Amplitude du défaut
Capteur N=°1	0.2s	0.01
Capteur N=°2	1.2s	0.02

Les défauts et les décisions sont représentés dans la figure 13.

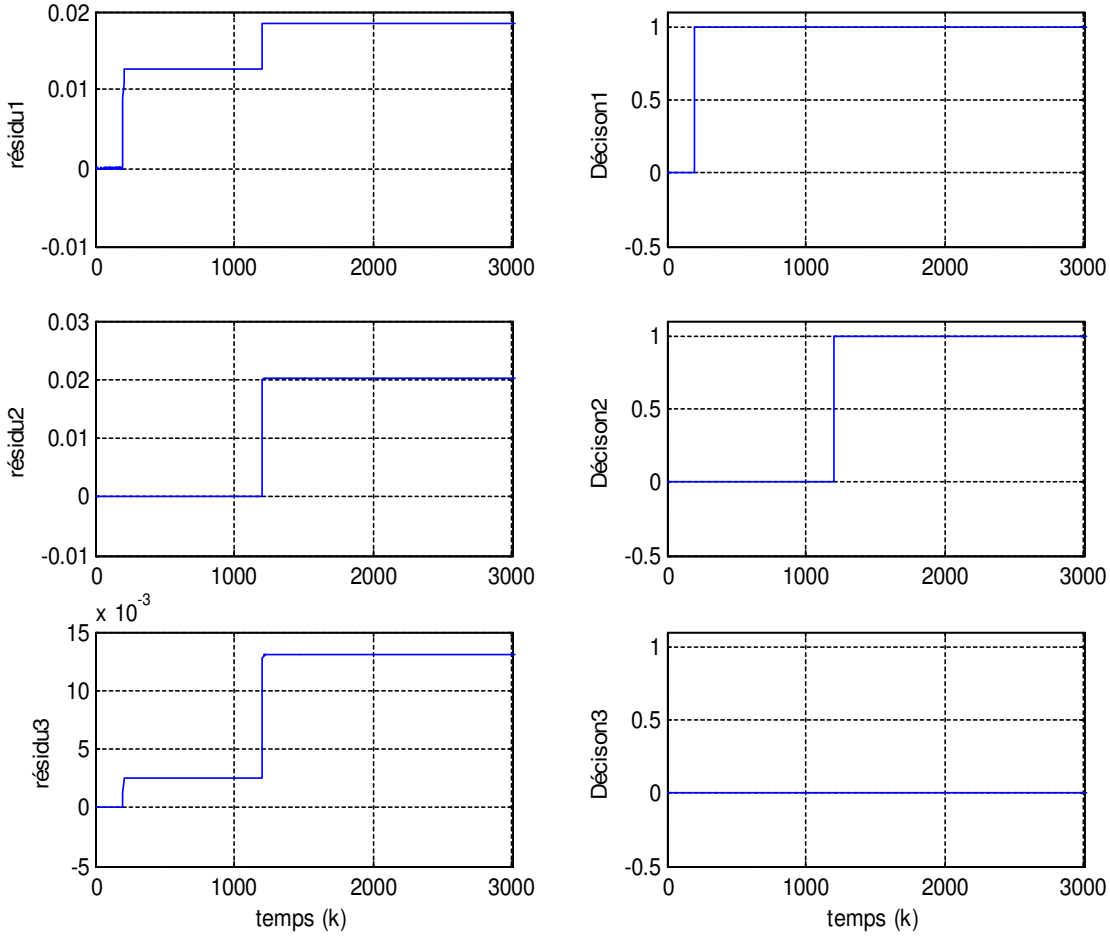


Figure 13 : Résidus et décisions en présence de défauts sur les capteurs 1 et 2

- **Essai 6 :** en présence de défauts sur le capteur 2 et le capteur 3.

Les amplitudes et les instants des défauts capteurs sont présentés dans le tableau suivant :

N° du capteur	Instant du défaut	Amplitude du défaut
Capteur N=°2	0.2s	0.018
Capteur N=°3	1.2s	0.06

Les défauts et les décisions sont représentés dans la figure 14.

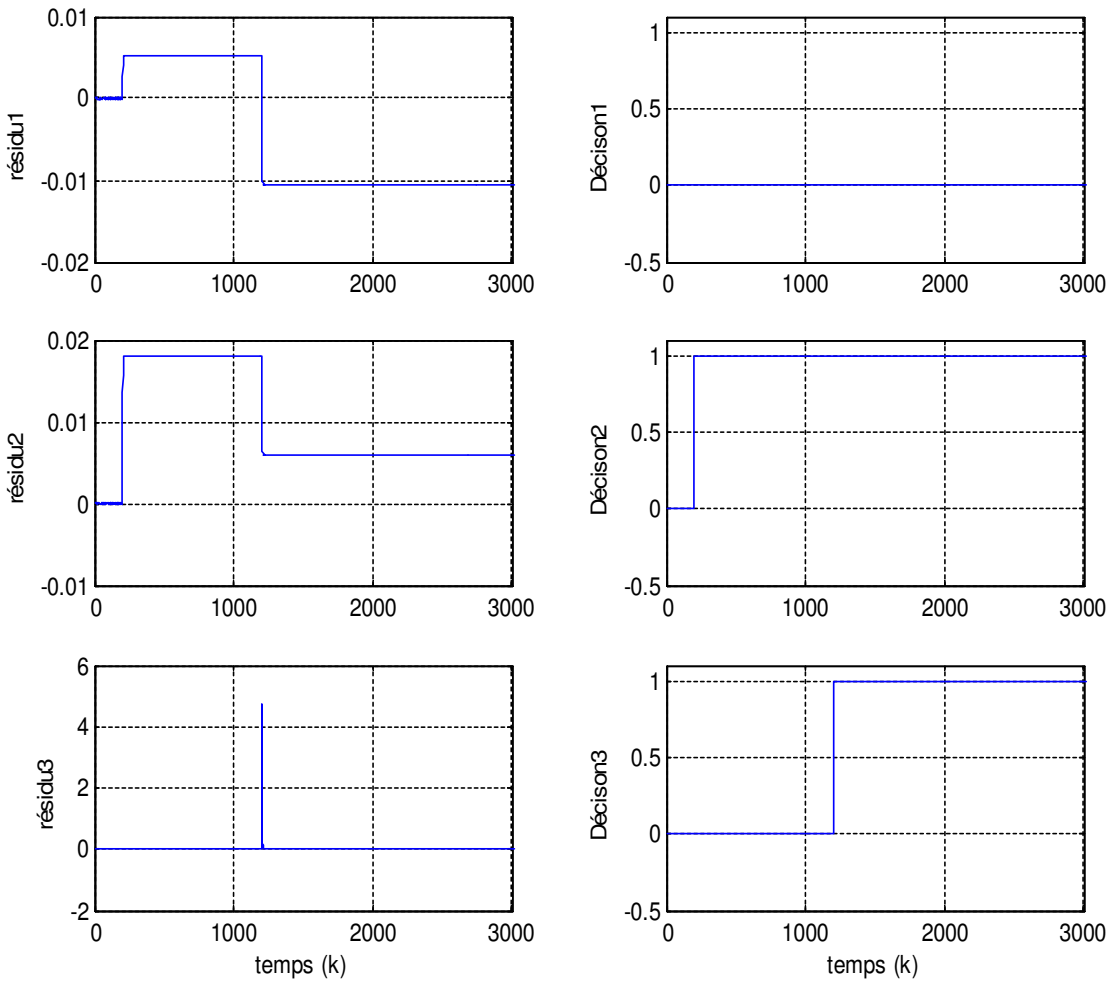


Figure 14 : Résidus et décisions en présence de défauts sur les capteurs 2 et 3

- **Essai 7 :** en présence de défauts sur le capteur 1 et le capteur 3.

Les amplitudes et les instants des défauts capteurs sont présentés dans le tableau suivant :

N° du capteur	Instant du défaut	Amplitude du défaut
Capteur N=°1	0.2s	0.025
Capteur N=°3	1.2s	0.075

Les défauts et les décisions sont représentés dans la figure 15.

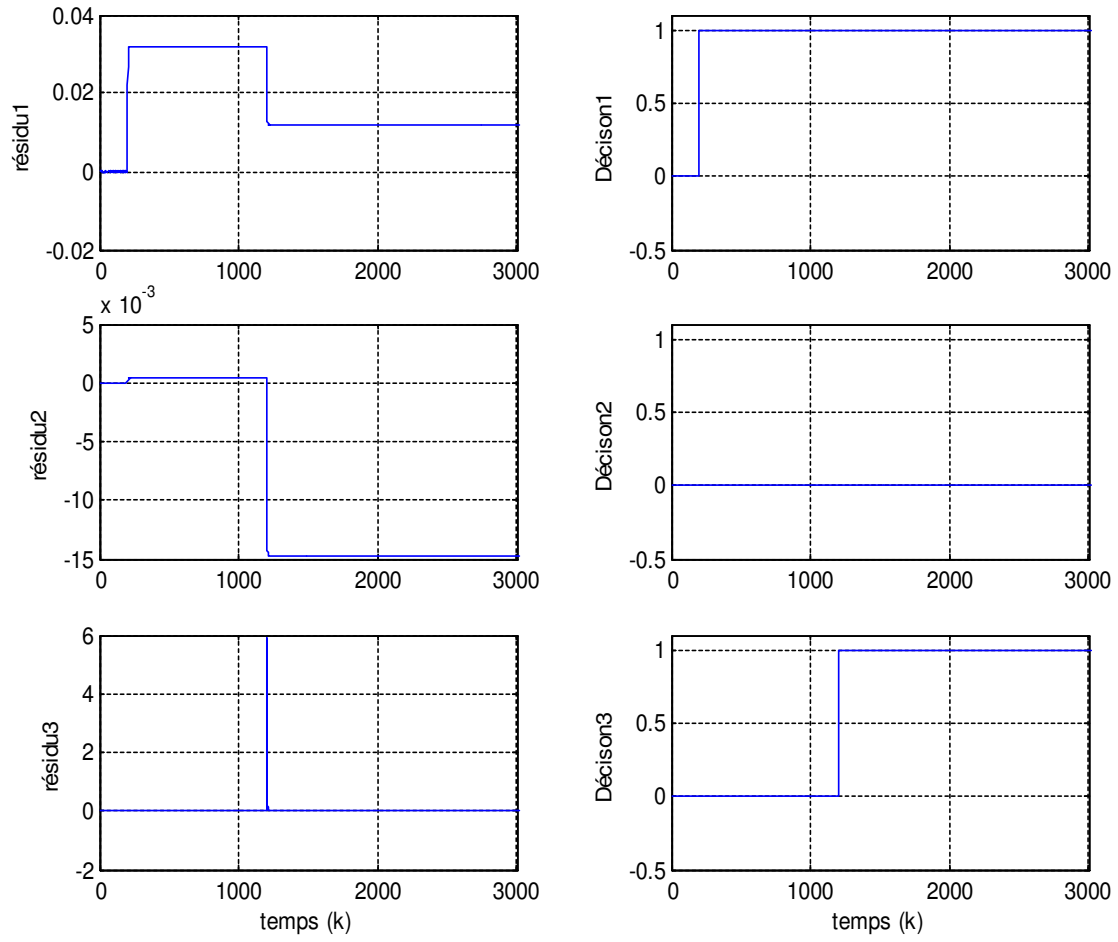


Figure 15 : Résidus et décisions en présence de défauts sur les capteurs 1 et 3

- **Essai 8 :** en présence de défauts sur le capteur1, capteur2 et 3.

Les amplitudes et les instants des défauts capteurs sont présentés dans le tableau suivant :

N° du capteur	Instant du défaut	Amplitude du défaut
Capteur N=°1	1.5s	0.03
Capteur N=°2	0.5s	0.02
Capteur N=°3	2.3s	0.09

Les défauts et les décisions sont représentés dans la figure 16.

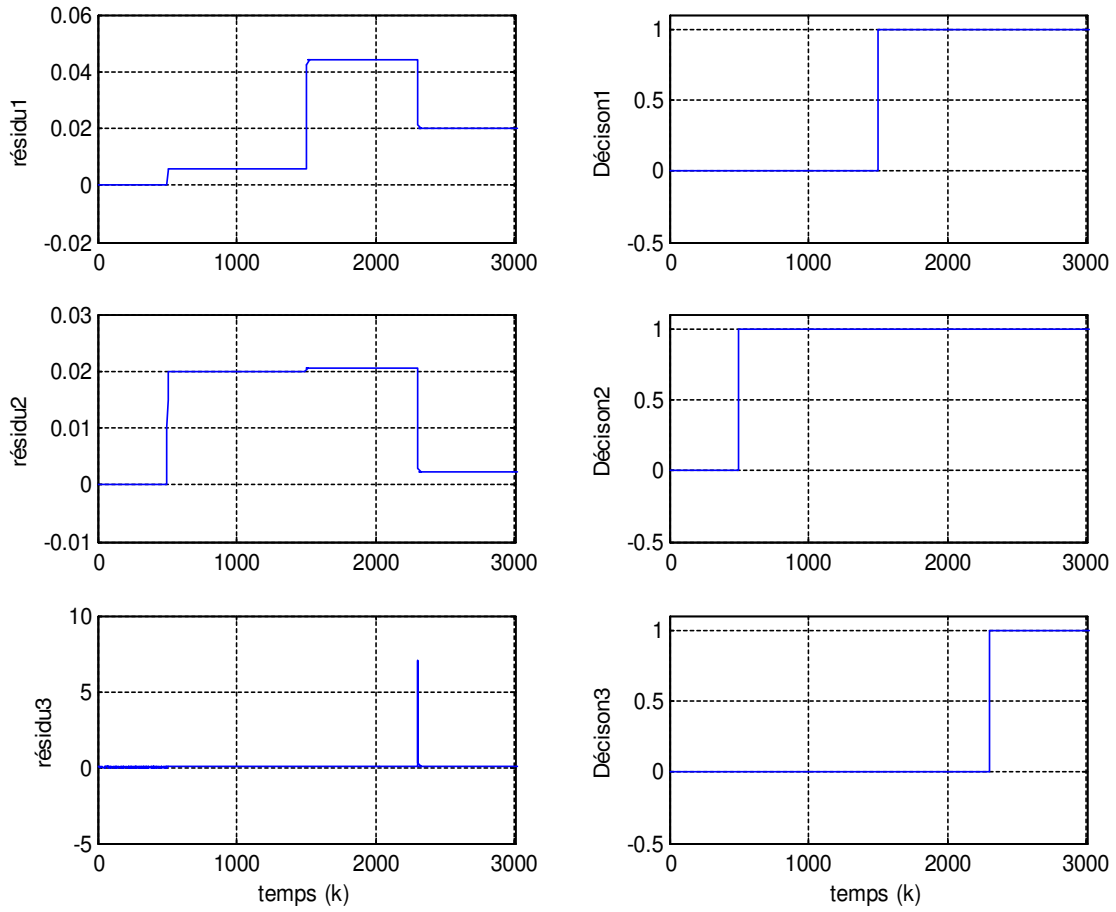


Figure 16 : Résidus et décisions en présence de défauts sur les trois capteurs

- **Essai 9 :** en présence de défauts et bruits de mesure sur le capteur 1.

Les sorties sont corrompues par des bruits blancs de variances (0.001)

N° du capteur	Instant du défaut	Amplitude du défaut de type biais
Capteur N=°1	0.5s.	0.021

Les défauts et les décisions sont représentés dans la figure 17.

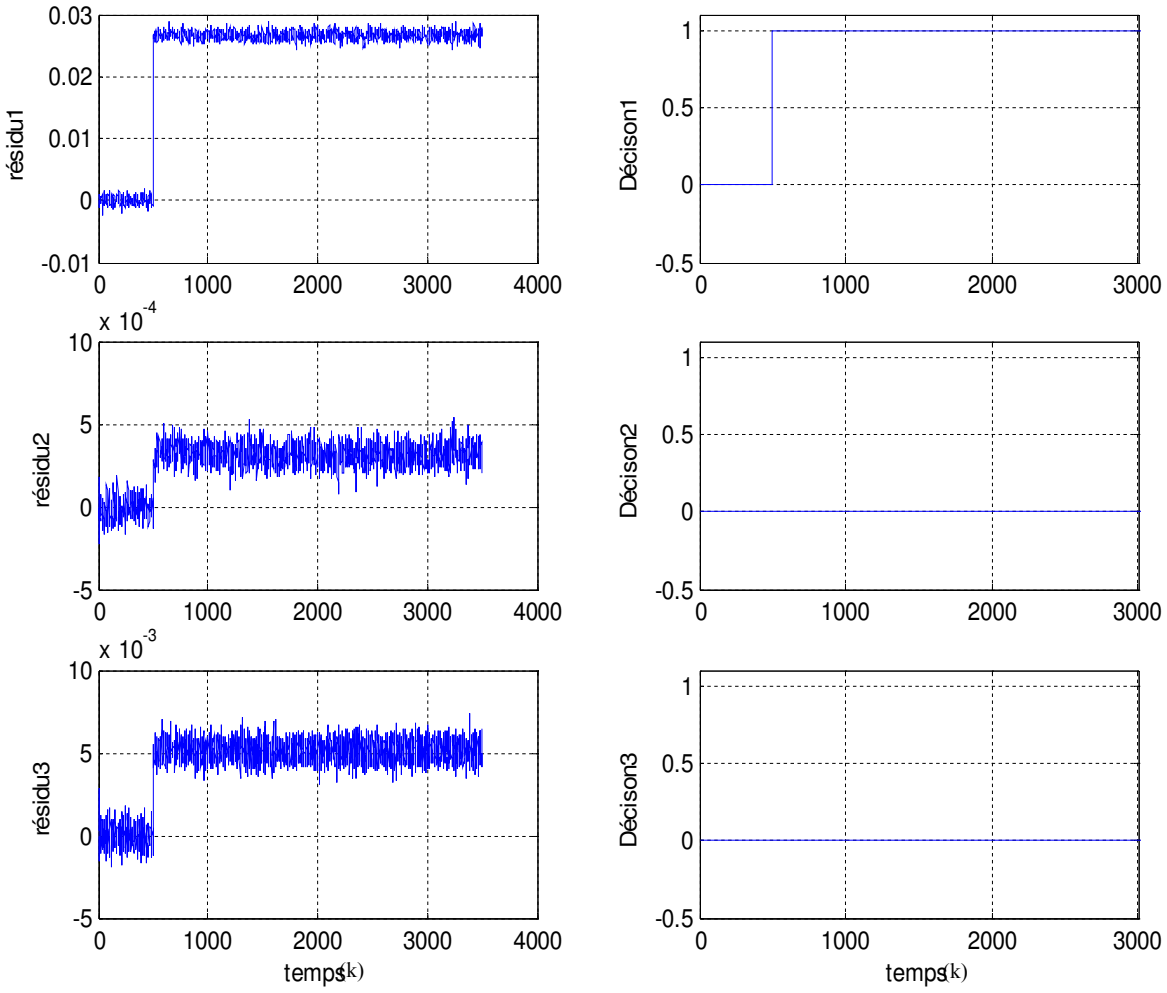


Figure 17 : Résidus et décisions avec le capteur1 en défaut (mesures bruitées)

- **Essai 10 :** en présence de défauts et des bruits de mesures sur le capteur 1, capteur 2 et 3.

Les amplitudes et les instants des défauts capteurs sont présentés dans le tableau suivant :

N° du capteur	Instant du défaut	Amplitude du défaut
Capteur N=°1	0.5s	0.03
Capteur N=°2	1.5s	0.02

Capteur N=°3	0.6s	0.09

Les défauts et les décisions sont représentés dans la figure 18.

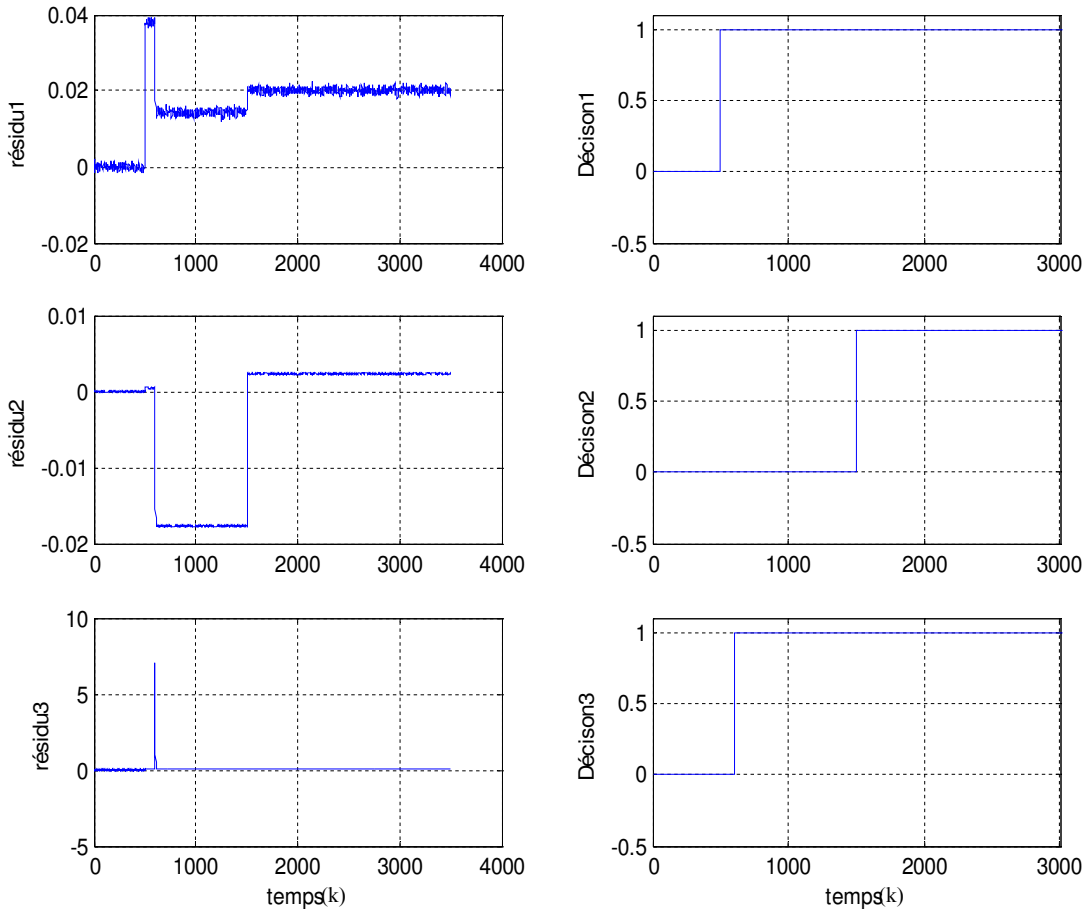


Figure 18 : Résidus et décisions avec présence de défauts sur les trois capteurs

VI - DIAGNOSTIQUE DES DÉFAUTS PHYSIQUES

Pour que les défauts soient bien détecter, on reprend dans cette partie l'apprentissage du réseau avec la valeur du frottement visqueux f_v=0.025.

En appliquant au système les mêmes tensions *Usd et Usq,* à amplitude 400V, on aboutit aux résultats qui sont présentés sur la figure 19.

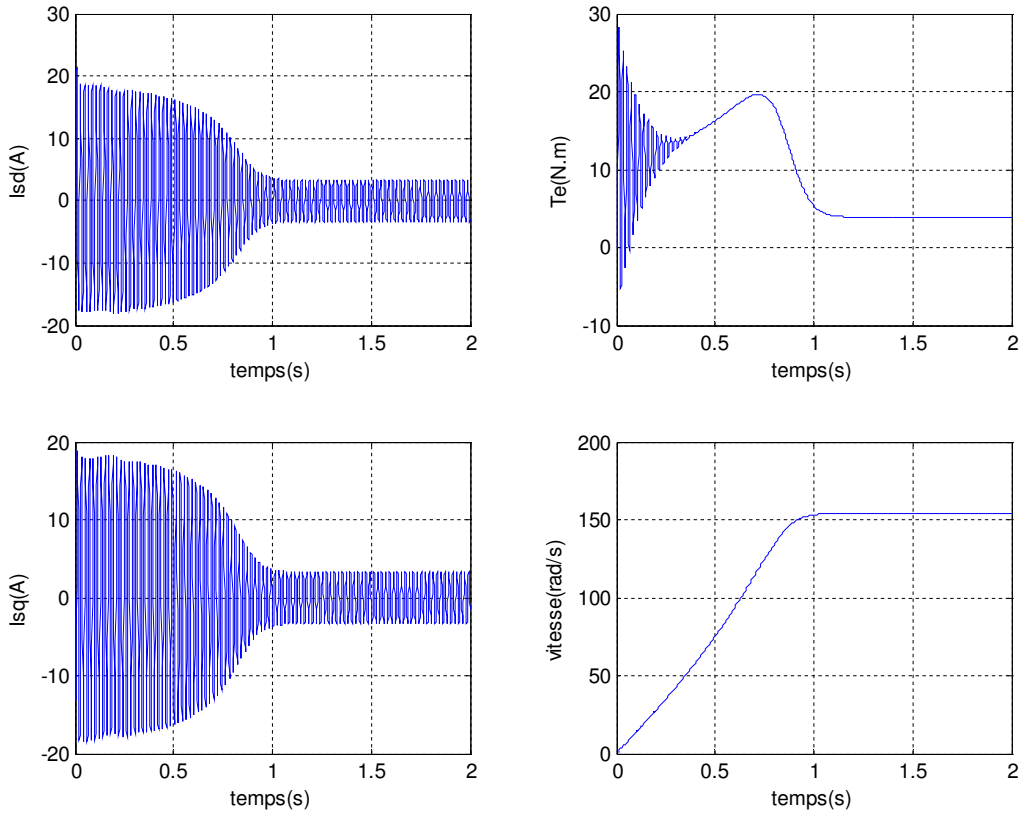


Figure 19 : le comportement du système en présence du f_v

VI.1 - Génération des résidus

VI.1.1 - Structure du réseau de neurones

Après plusieurs essais on a choisi la même structure utilisée dans la première partie de type NNARX pour la génération des résidus, **quatre** neurones dans la couche cachée, **trois** neurones dans la couche de sortie et **cinq** neurones dans la couche d'entrée (Figure 3).

VI.1.2 - Apprentissage

L'évolution de l'erreur quadratique a aboutit à 3.094 10^{-9} au bout de 8 itérations, après avoir effectuer plusieurs essais.

La figure suivante présente les résidus sans défauts :

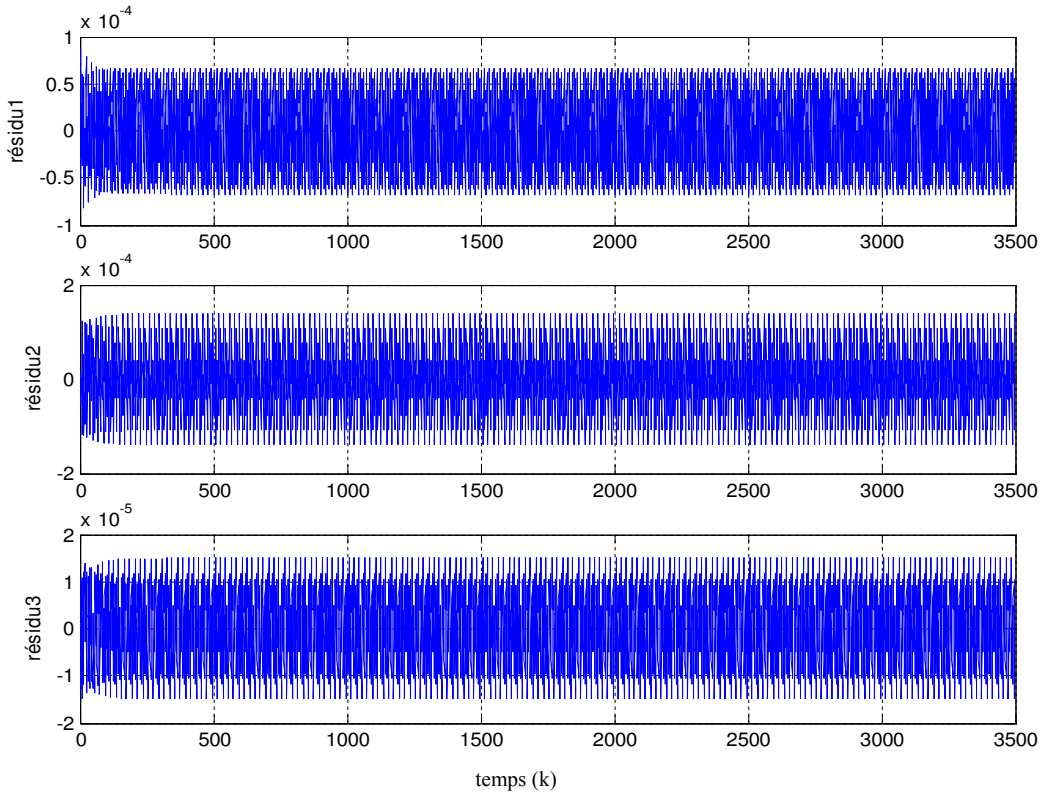


Figure 20 : Résidus sans défauts physiques

Les figures ci-dessous représentent les défauts des paramètres physiques du système (Rs, Rr, Ls, Lr) à 5% :

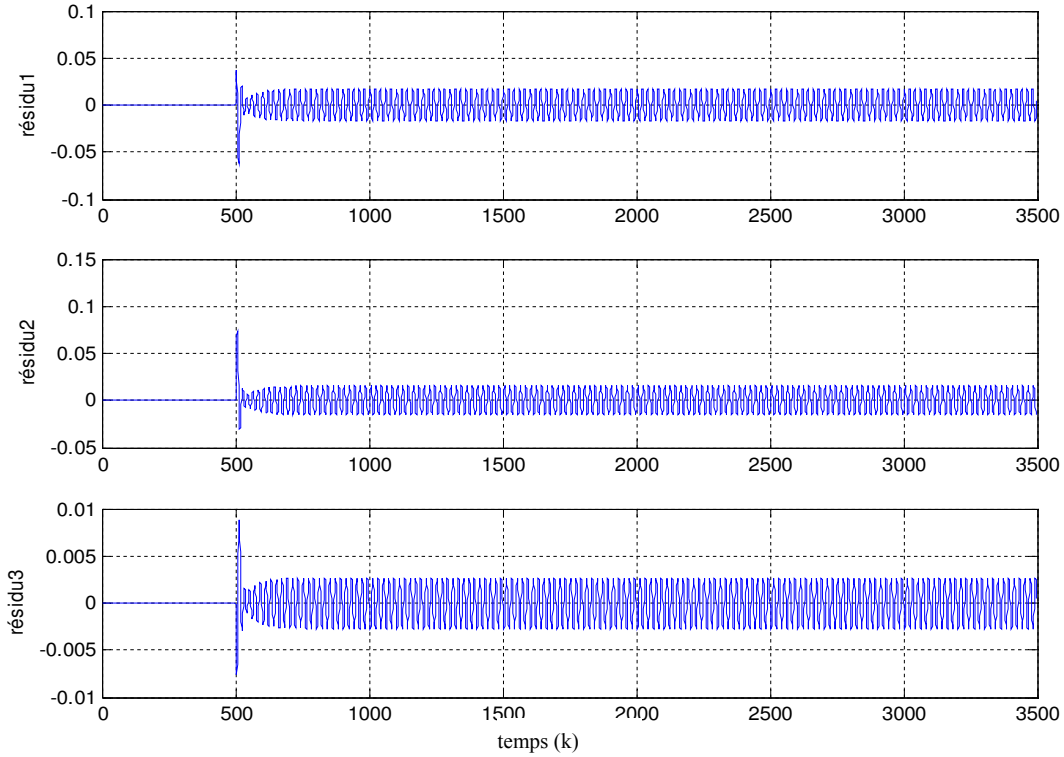


Figure 21: Défauts sur Rs à 5%

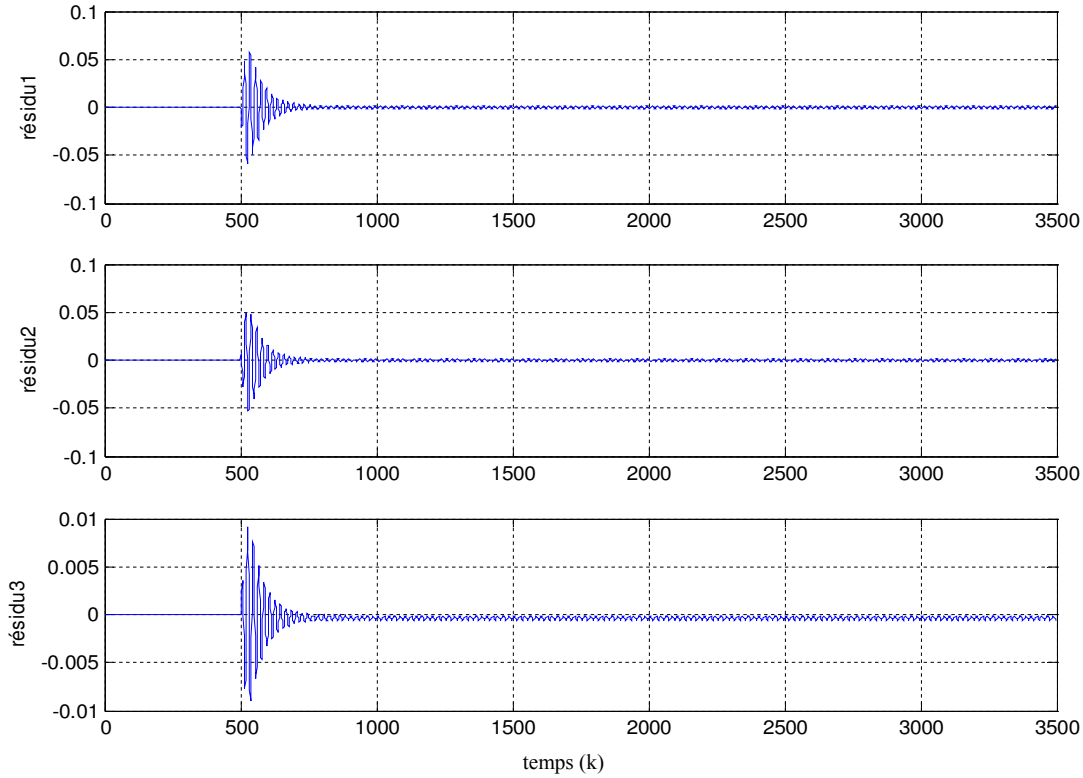


Figure 22 : Défauts sur Rr à 5%

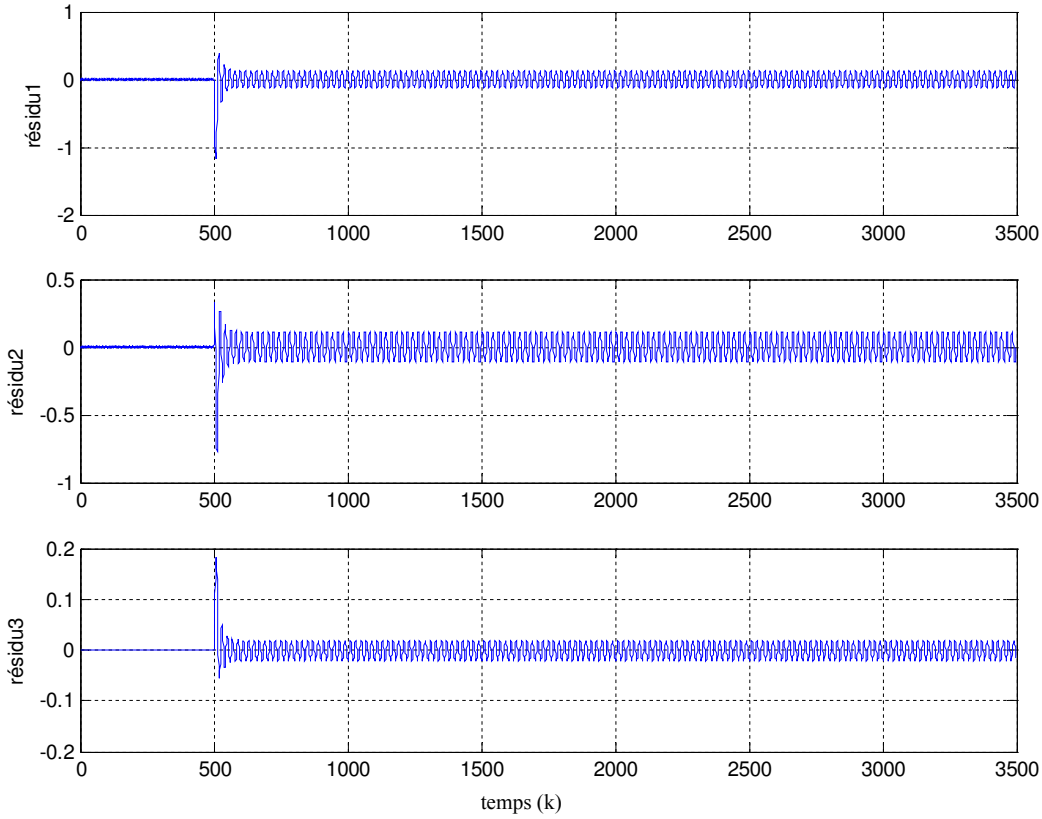


Figure 23 : Défaut sur Ls à 5%

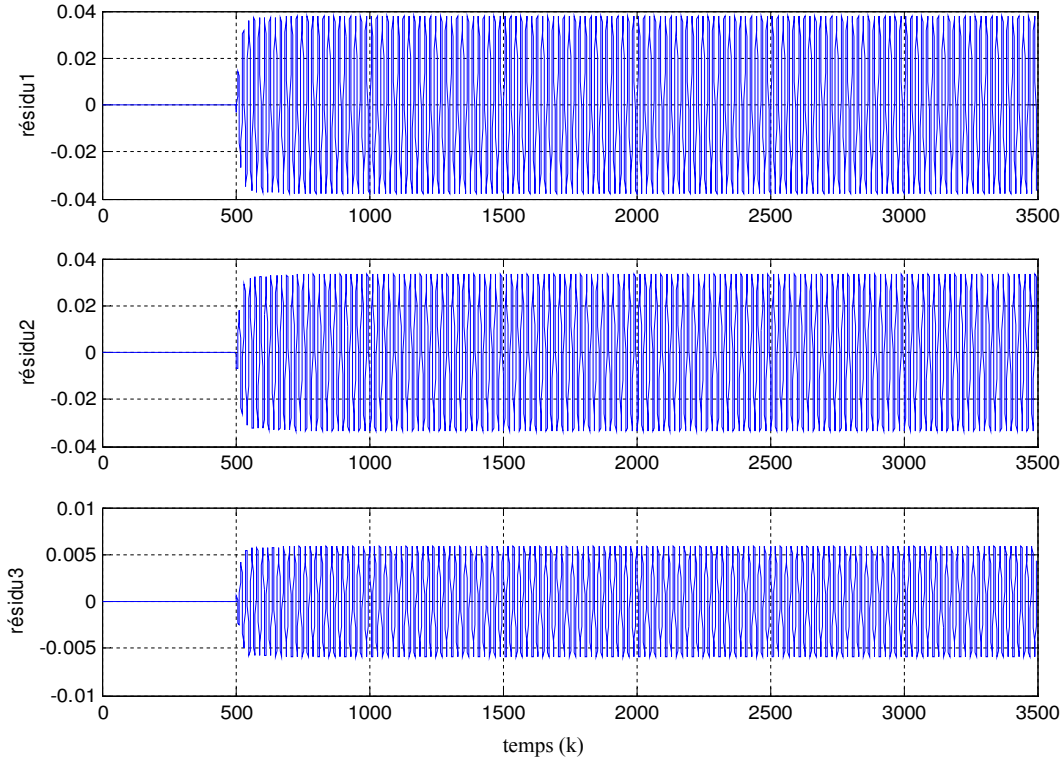


Figure 24 : Défauts sur Lr à 5%

Pour bien représenter les défauts on lès a introduits successivement de cette façon :

	Rs	Rr	Ls	Lr
Défaut	8%	12%	3%	5%
Instant	[2 3] s	[3.5 4.5] s	[5 6]s	[6.5 7.5] s

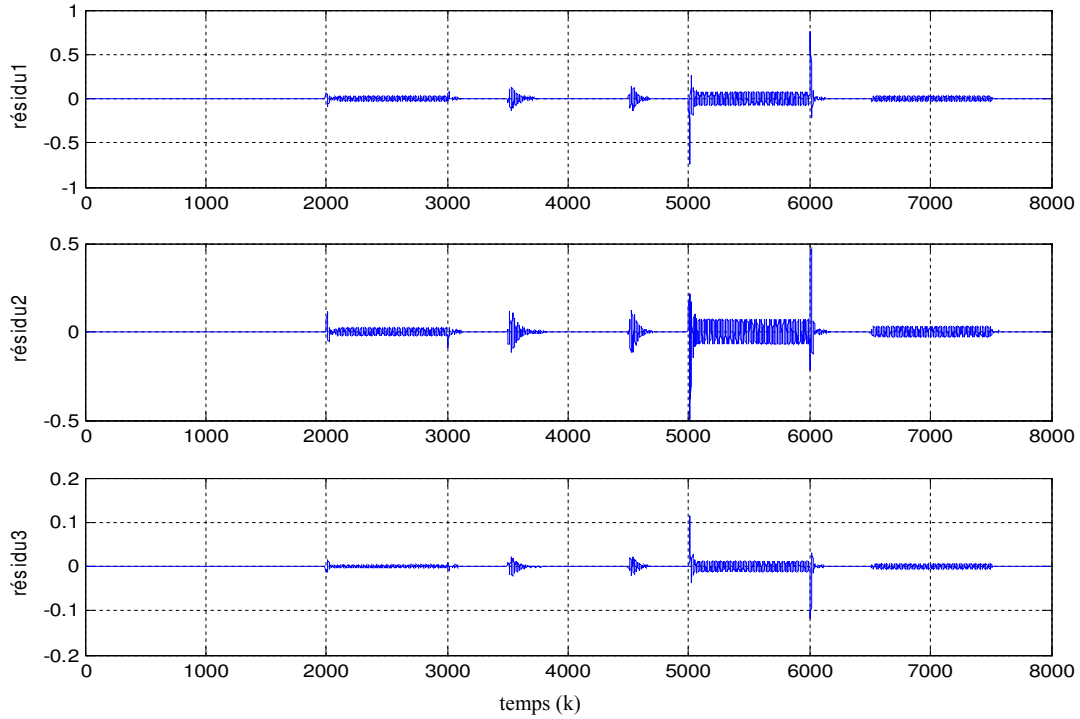


Figure 25 : Défauts successifs sur les paramètres physiques (Rs, Rr, Ls, Lr)

On constate que les défauts paramétriques sont bien détectés, car les résidus sont très sensibles à des petites variations sur (Rs, Rr, Ls, Lr). Toutefois leur localisation n'est pas évidente car leurs signatures sont identiques.

VII - CONCLUSION

Ce chapitre a présenté les résultats obtenus en simulation, qui nous permettent de conclure à l'efficacité de l'approche neuro-floue pour le diagnostic des défauts pouvant affecter la machine asynchrone, particulièrement pour les défauts additifs de type capteur.

G - Conclusion

Le schéma de diagnostic étudié dans ce mémoire comporte deux étapes :

- La génération des résidus de surveillance à l'aide d'un réseau de neurones de type perceptron à deux couches basé sur un modèle NNARX.
- La décision pour la détection et la localisation des défauts à l'aide d'un système neuro-flou.

L'application de ce schéma de diagnostic à la machine asynchrone, qui est un système non-linéaire, a donné des résultats en simulation très concluants, notamment pour les défauts de type capteur.
Les défauts électriques de la MAS sont des défauts physiques internes qui sont en général plus difficiles à diagnostiquer que les défauts de type capteur. Pour les prendre en charge, on suggère l'utilisation d'un schéma de génération de résidus basé sur un banc de réseaux de neurones dont l'apprentissage de chacun est adapté à chaque type de défaut considéré.
En présence de mesures bruitées, il est conseillé d'utiliser la structure NNARMAX pour la génération des résidus.
Ces deux derniers points méritent d'être étudiés dans le futur, pour aboutir à un système de diagnostic plus efficace de la MAS.

H - ANNEXE - Les Réseaux de Neurones

Les neurosciences et les sciences cognitives nous proposent de nombreux modèles, notamment pour représenter les systèmes perceptifs et les systèmes réflexes. Mais, il reste encore beaucoup de travail de recherche en neurosciences et en sciences cognitives pour pouvoir modéliser les systèmes cognitifs, et enfin passer du "réflexe à la réflexion". Si un tel intérêt se porte sur les réseaux de neurones c'est grâce à leurs propriétés : parallélisme, capacité de généralisation, capacité d'adaptation et mémoire distribuée.

I - INTRODUCTION (HISTORIQUE)

Le champ des réseaux neuronaux va démarrer par la présentation en 1943 par W. MC Culloch et W. Pitts du **neurone formel** qui est une abstraction du neurone physiologique. Le retentissement va être énorme. Par cette présentation, ils veulent démontrer que le cerveau est équivalent à une machine de Turing, la pensée devient alors purement des mécanismes matériels et logiques. Il déclara en 1955 "Plus nous apprenons de choses au sujet des organismes, plus nous sommes amenés à conclure qu'ils ne sont pas simplement analogues aux machines, mais qu'ils sont machine." [Mysterium Iniquitatis of Sinful Man Aspiring into the Place of God, repris in Embodiments of mind]. La démonstration de McCulloch et Pitts sera un des facteurs importants de la création de la cybernétique.

En 1949, D. Hebb présente dans son ouvrage "The Organization of Behavior" une règle d'apprentissage. De nombreux modèles de réseaux aujourd'hui s'inspirent encore de **la règle de Hebb**.

En 1958, F. Rosenblatt développe le modèle du Perceptron. C'est un réseau de neurones inspiré du système visuel. Il possède deux couches de neurones : une couche de perception et une couche liée à la prise de décision. C'est le premier système artificiel capable d'apprendre par expérience.

Dans la même période, Le modèle de L'**Adaline** (ADAptive LINar Element) a été présenté par B. Widrow, chercheur américain à Stanford. Ce modèle sera par la suite le modèle de base des réseaux multicouches.

En 1969, M. Minsky et S. Papert publient une critique des propriétés du Perceptron. Cela va avoir une grande incidence sur la recherche dans ce domaine. Elle va fortement diminuer jusqu'en 1972, où T. Kohonen présente ses travaux sur les mémoires associatives. Et propose des applications à la reconnaissance de formes. C'est en 1982 que J. Hopfield présente son étude d'un réseau complètement rebouclé, dont il analyse la dynamique. Aujourd'hui, les réseaux neuronaux sont utilisés

dans de nombreux domaines (entre autres, vie artificielle et intelligence artificielle) à cause de leur propriété en particulier, leur capacité d'apprentissage, et qu'ils soient des systèmes dynamiques.

Depuis, les recherches et les applications ont encore progressé :

1943	Mc Clloch & Pitts	Définition du neurone formel
1949	Hebb	Règles d'apprentissage pour les neurones
1957	Rosenblatt	Modèle de réseau de neurones : Le Perceptron
1969	Minsky & Papert	Optimisations sur les Perceptrons
1982	Hopfield	Modèle en relation avec les systèmes physiques
1989	Kohonen	Auto-organisation et mémoires associatives
1992	Boughanem	Réseaux de neurones appliqués à la recherche d'informations

II - FONDEMENTS BIOLOGIQUES

Les chercheurs en intelligence artificielle sur les réseaux de neurones se sont inspirés des modèles biologiques (cerveau et neurones).

II.1 - Le cerveau

Le cerveau est le centre de contrôle de la perception, de la décision, et de l'action. Le système nerveux traite l'information via des unités interconnectées entre elles : les neurones. Le cerveau en possède 10^{13}et chacun d'entre eux est connecté en moyenne à mille autres.

II.2 - Le neurone

Nous pouvons faire un parallèle entre ce modèle mathématique et certaines informations que nous connaissons (ou que nous croyons connaître) à propos du neurone biologique. Ce dernier possède trois principales composantes : les dendrites, le corps cellulaire et l'axone :

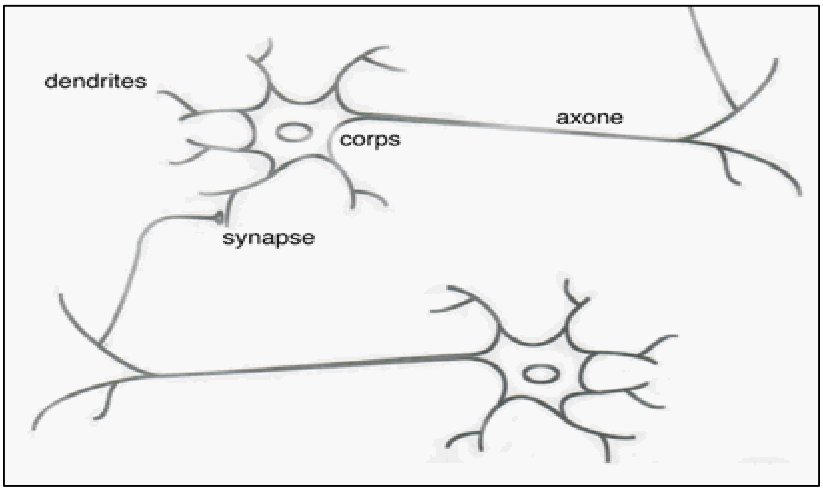


Figure 1

Les dendrites forment un maillage de récepteurs nerveux qui permettent d'acheminer vers le corps du neurone des signaux électriques en provenance d'autres neurones. Celui-ci agit comme une espèce d'intégrateur en accumulant des charges électriques. Lorsque le neurone devient suffisamment excité (lorsque la charge accumulée dépasse un certain seuil), par un processus électrochimique, il engendre un potentiel électrique qui se propage à travers son axone pour éventuellement venir exciter d'autres neurones. Le point de contact entre l'axone d'un neurone et la dendrite d'un autre neurone s'appelle le "synapse". Il semble que c'est l'arrangement spatial des neurones et leur axone, ainsi que la qualité des connexions synaptiques individuelles qui détermine la fonction précise d'un réseau de neurones biologique. C'est en se basant sur ces connaissances que le modèle mathématique décrit ci-dessus a été défini.

III - LE NEURONE ARTIFICIEL

Les structures biologiques et les mécanismes physiologiques des neurones sont extrêmement compliqués. Aussi, pour essayer d'en comprendre le fonctionnement, il a fallu introduire une forme simplifiée du neurone : le neurone formel. Nous allons donc étudier le neurone formel de base, puis un neurone formel plus détaillé et plus général, le neurone formel étendu.

III.1 - Le neurone formel de base

Le neurone formel a été présenté par Mc Culloch & Pitts en 1943, il se présente comme un opérateur de sommation dont les entrées sont pondérées (figure 2).

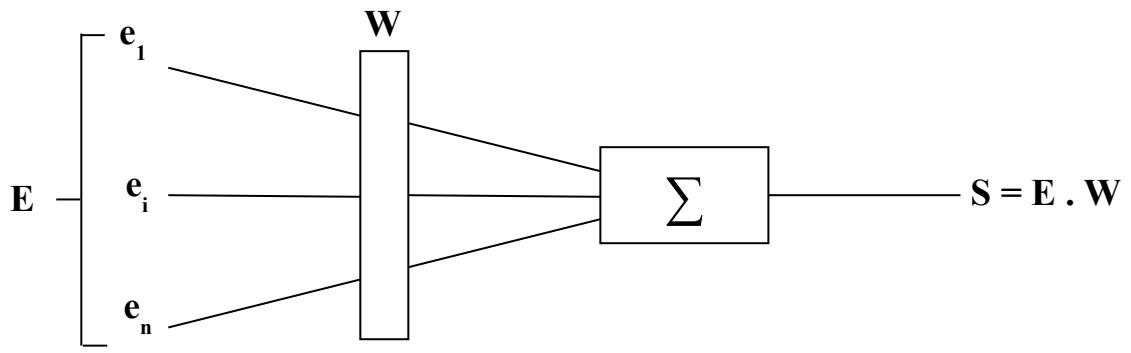


Figure 2

Avec :

E (e1...ei...en) : vecteur d'entrée du neurone

W (w1...wi...wn) : vecteur des poids

S : sommation pondérée des entrées $\mathbf{S} = \sum_{i=1}^{n} e_i w_i$

La sortie **S** est souvent traitée par une fonction d'activation pour produire un signal de sortie.

III.2 - Le neurone formel à fonction d'activation

Le modèle du neurone formel précédent est complété par l'association d'une fonction d'activation à la valeur **S** calculée. Nous pouvons distinguer:

Fonction à seuil : -Heaviside : F(x) = $\begin{Bmatrix} 1 \, si \, x \geq 0 \\ 0 \, si \, x < 0 \end{Bmatrix}$

-Signe : F(x) = $\begin{Bmatrix} +1 \, si \, x \geq 0 \\ -1 \, si \, x < 0 \end{Bmatrix}$

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

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

III.3 - Nature des fonctions d'entrée, d'activation et de sortie

La nature des entrées et des sorties peut être soit binaire, soit réelle. Par extension, le

neurone formel étendu est défini ainsi (figure 3):

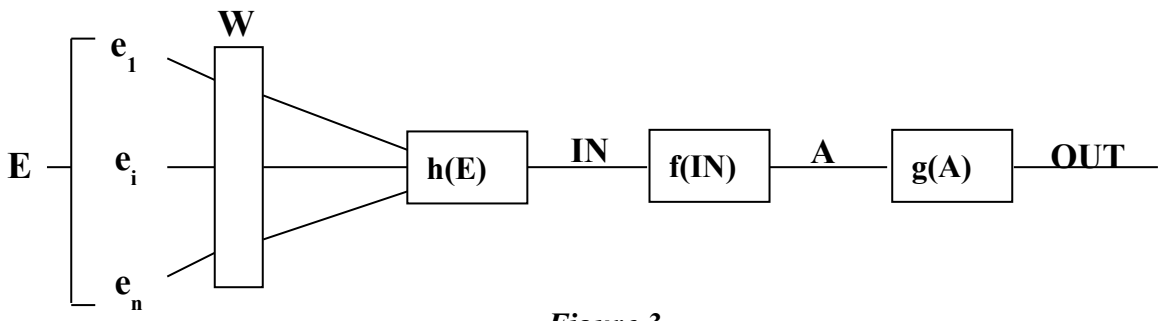


Figure 3

Avec :

E (e1...ei...en) : vecteur d'entrée du neurone

W (w1...wi...wn) : vecteur des poids

IN = h (E) : fonction d'entrée totale (binaire, linéaire, affine ou polynomiale)

A = f (IN) : fonction d'activation (binaire à seuil, linéaire à seuil ou multi seuil, sigmoïde, stochastique, ou toute fonction croissante et impaire).

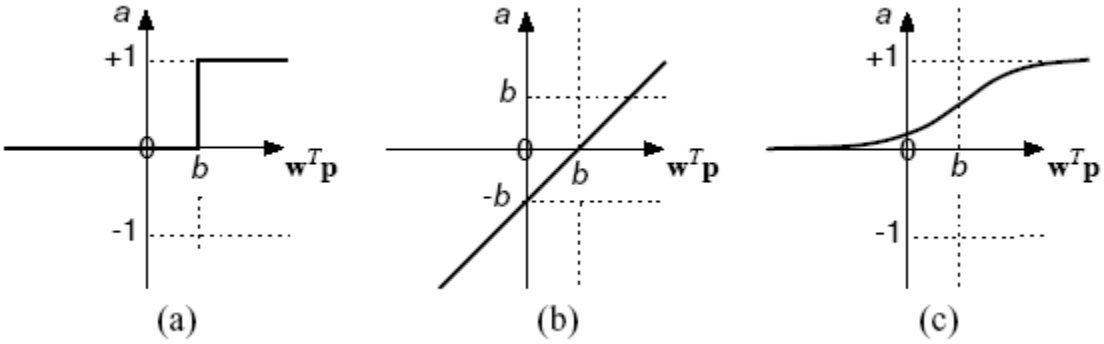
S = OUT = g(A) : fonction de sortie (identité, ou toute fonction croissante et impaire).

Quelques types des fonctions d'activation :

Jusqu'à présent, nous n'avons pas spécifié la nature de la fonction d'activation ***a=f(n)*** de notre modèle. Il se trouve que plusieurs possibilités existent et celles-ci sont quasiment empiriques et à adapter en fonction des situations. Les plus courantes et les plus citées dans la littérature sont énumérées dans la figure ci-dessous :

Nom de la fonction	Relation d'entrée/sortie	Icône
seuil	$a = 0$ si $n < 0$ $a = 1$ si $n \geq 0$	
seuil symétrique	$a = -1$ si $n < 0$ $a = 1$ si $n \geq 0$	
linéaire	$a = n$	
linéaire saturée	$a = 0$ si $n < 0$ $a = n$ si $0 \leq n \leq 0$ $a = 1$ si $n > 0$	
linéaire saturée symétrique	$a = -1$ si $n < -1$ $a = n$ si $-1 \leq n \leq -1$ $a = 1$ si $n > 1$	
linéaire positive	$a = 0$ si $n < 0$ $a = n$ si $n \geq 0$	
sigmoïde	$a = \frac{1}{1+\exp^{-n}}$	
tangente hyperbolique	$a = \frac{e^n - e^{-n}}{e^n + e^{-n}}$	
	$a = 1$ si n maximum $a = 0$ si autrement	C

Les trois les plus utilisées dans le domaine de l'ingénierie sont les fonctions "**seuil**" (a) (en anglais "**hard limit**"), "**linéaire**" (b) et "**sigmoïde**" (c) comme représentées ci-dessous :



Comme son nom l'indique, la fonction seuil applique un seuil sur son entrée. Plus précisément, une entrée négative de passe pas le seuil, la fonction retourne la valeur 0 (faux), alors qu'une entrée positive ou nulle dépasse le seuil, et la fonction retourne 1 (vrai). Il est évidant que ce

genre de fonction permet de prendre des décisions binaires (cette fonction peut aussi être assimilée à la "fonction de Heaviside" pour ceux qui connaissent).

La fonction linéaire est quant à elle très simple, elle affecte directement son entrée à sa sortie selon la relation ***a=f(n)=n*** . Il est évident que la sortie du neurone correspond alors à son niveau d'activation dont le passage à zéro (l'ordonnée à l'origine) se produit lorsque $w \cdot \bar{p} = b$. La fonction de transfert sigmoïde est quant à définir par la relation mathématique :

$$a = \frac{1}{1+e^n}$$

Elle ressemble soit à la fonction seuil, soit à la fonction linéaire, selon que nous sommes loin ou près de *b* respectivement. La fonction seuil est très non linéaire car il y a une discontinuité lorsque $w \cdot \bar{p} = b$. De son côté, la fonction linéaire est tout à fait linéaire. Elle ne comporte aucun changement de pente. La sigmoïde est un compromis intéressant entre les deux précédentes. Notons finalement que la fonction "tangente hyperbolique" est une version symétrique de la sigmoïde.

IV - LES CONNEXIONS

Les connexions vont relier les neurones entre eux. **La structure des connexions** peut aller de la connectivité partielle à la connectivité totale. On associe à une connexion entre deux neurones, un **poids** qui rend compte de l'influence d'un des neurones sur l'autre. **La dynamique des états** correspond à l'évolution des états des différents neurones d'un réseau. Elle dépend à la fois des fonctions d'activation, de la structure et des poids des connexions. **La dynamique des connexions** représente le fait que les poids des connexions peuvent être modifiés par une phase appelée **l'apprentissage**. On peut assimiler cette dynamique à la plasticité synaptique.

V - ARCHITECTURES DE RÉSEAUX DE NEURONES

L'organisation des réseaux de neurones peut se faire à partir de 2 grandes catégorie d'architecture: les réseaux sans couches et les réseaux à couches

V.1 - Réseaux sans couches

Le réseau possède des neurones connectés entre eux, certains sont des neurones d'entrée et d'autres sont des neurones de sortie (figure 5).

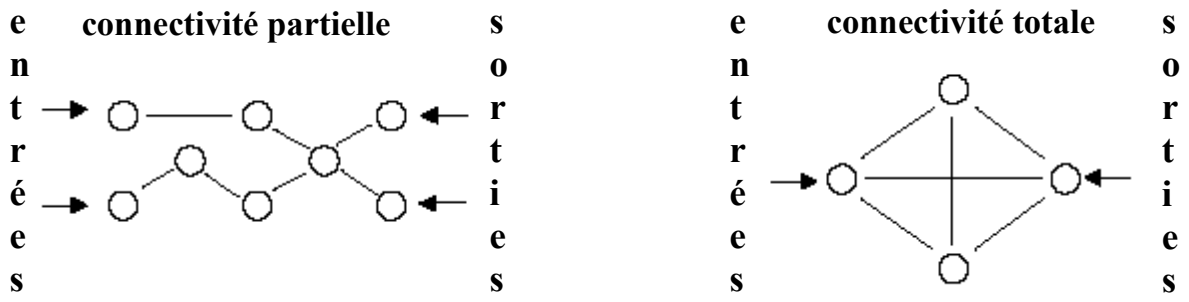


Figure 5

V.2 - Réseaux à couches

La particularité de ce type de réseau est de posséder des couches de neurones indépendantes. Chaque couche a sa propre organisation, mais la connexion entre les couches est particulière. En effet, les neurones d'une couche **Ci** peuvent être connectés avec les neurones de la couche **Ci-1** et avec ceux de la couche **Ci+1**.

On distingue trois types de couches : couche d'entrée, couches cachées et couche de sortie (figure 6).

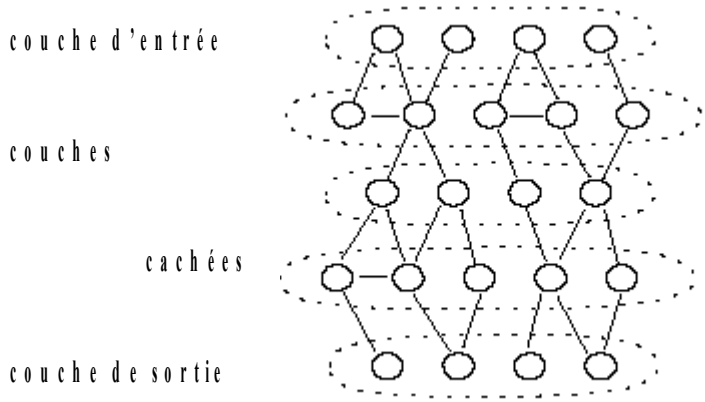


Figure 6

VI - MODÈLES DE RÉSEAUX DE NEURONES

De nombreux chercheurs ont développé des topologies de réseaux de neurones, dont ils ont démontré les propriétés. Par la suite, ces topologies sont devenues des modèles de base dont on s'inspire pour créer de nouvelles architectures de réseau de neurones.

VI.1 - Le modèle du perceptron

Le modèle du Perceptron ou le Perceptron de Rosenblatt est le premier modèle à couches opérationnel, mettant en oeuvre les idées de Hebb et de Mc Culloch & Pitts. Le perceptron est une machine de vision élémentaire composée de trois couches interconnectées (figure 7).

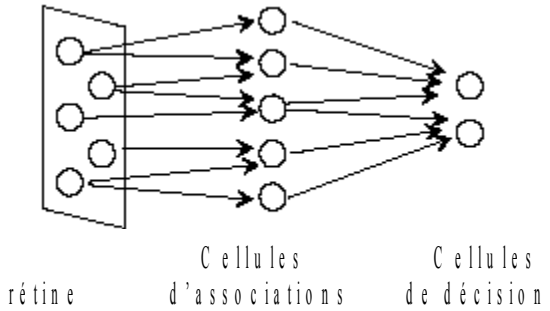


Figure 7

VI.2 - Le modèle de Hopfield

D'après Hopfield, le système nerveux recherche des états stables ou attracteurs dans son espace d'états. Les états voisins tendent à se rapprocher d'un état stable, ce qui autorise la correction d'erreurs et la capacité à compléter les informations manquantes. Un réseau de Hopfield est composé de neurones binaires tous reliés entre eux, ce qui donne une connectivité totale (figure 8).

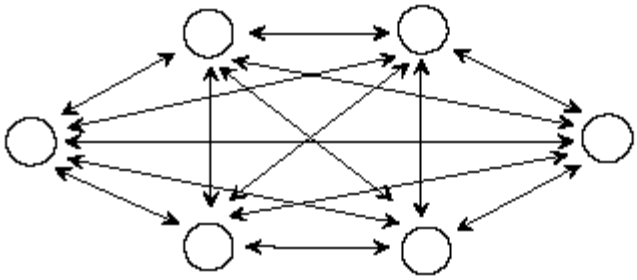


Figure 8

Lorsque la matrice des poids des connexions est symétrique, la stabilité du réseau est déterminée par le minimum d'une fonction d'énergie appelé **Attracteur**.

Si l'on peut démontrer la stabilité du réseau, le rappel des exemples appris n'est pas toujours réalisé. De plus, le nombre d'exemples mémorisables n'est pas très élevé.

VI.3 - Le modèle de Kohonen

Les réseaux du modèle de Kohonen sont utilisés pour réaliser des opérations de classification et de regroupement. Ils sont composés d'une seule couche appelée couche d'auto-organisation (figure 9). L'architecture tient compte des données extérieures des cellules d'entrée et

des connexions internes.

Chaque neurone du réseau est connecté aux **n** composantes du vecteur d'entrée **X**.

Puis, les neurones propagent le signal vers les autres neurones du réseau. Kohonen a montré qu'il existe entre les neurones émetteurs et récepteurs un mécanisme d'interaction latérale qui dépend de la distance entre les neurones concernés. Cette dépendance est caractérisée par la fonction du **"chapeau mexicain"** (figure 10).

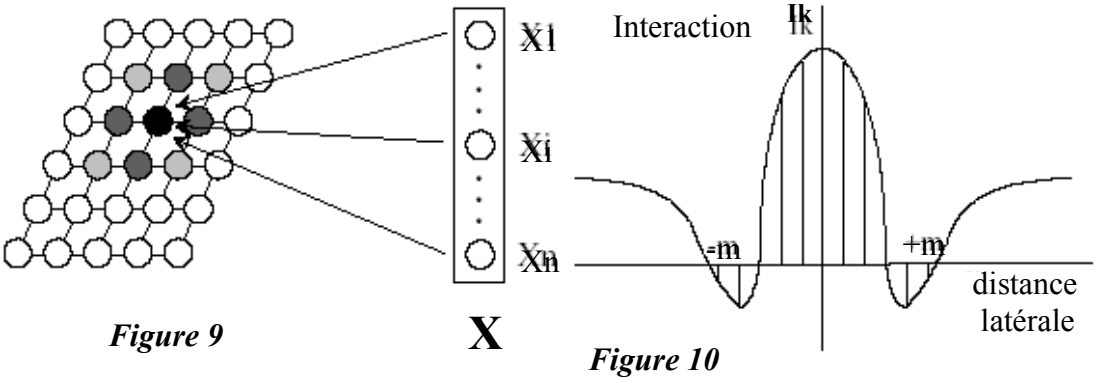


Figure 9

Figure 10

L'interprétation du comportement du réseau est la suivante :

- dans une zone latérale proche d'un neurone, les neurones qui lui sont connectés ont une action excitatrice,
- dans un voisinage plus lointain, l'action est inhibitrice,
- dans un voisinage encore plus lointain, l'action est négligeable.

La discrétisation de la fonction du **"chapeau mexicain"** va permettre de déterminer les poids des connexions internes entre les neurones (figure 10).

Finalement, l'interaction latérale doit provoquer l'amas de neurones excités autour du neurone le plus activé par le stimulus.

VII - APPRENTISSAGE DANS LES RÉSEAUX DE NEURONES

VII.1 - Apprentissage non supervisé

VII.1.1 - Présentation

L'apprentissage modifie les connexions pour produire la sortie la plus consistante, mais il n'y a pas de sorties désirées comme dans l'apprentissage supervisé.

VII.1.2 - Règles d'apprentissage de Hebb

L'apprentissage se fait par les algorithmes suivants :

$$W_{i\ j}^{t+1} = W_{i\ j}^{t} + O_i . O_j \quad \text{ou} \quad W_{i\ j}^{t+1} = W_{i\ j}^{t} + \left(O_i^t - O_i^{t-1}\right).\left(O_j^t - O_j^{t-1}\right)$$

VII.1.3 - Règles d'apprentissage compétitif

L'apprentissage s'effectue par coopération et compétition entre les neurones, le poids des connexions évolue ainsi $W_i^{t+1} = W_i^t + \eta . \left(X_i - W_i^t\right)$.

Avec : $\mathbf{W_i}$: vecteur poids des neurones en sortie du neurone **j**

$\mathbf{X_i}$: vecteur d'entrée du neurone **i**

η : vitesse d'apprentissage

t : étape d'apprentissage

Ce genre d'apprentissage aboutit souvent à la stabilité et la convergence du réseau.

VII.2 - Apprentissage supervisé

VII.2.1 - Présentation

L'apprentissage supervisé consiste à modifier les paramètres du réseau de façon à ce que les réponses de ce dernier soient des réponses attendues. On présente une entrée et on compare la sortie obtenue à la sortie désirée puis on ajuste les poids des connexions pour minimiser la différence.

VII.2.2 - Apprentissage dans le perceptron

L'apprentissage du Perceptron de Rosenblatt est un apprentissage supervisé qui se fait par correction d'erreur. Il consiste à présenter au réseau une série d'exemples à apprendre, puis à minimiser l'erreur entre la sortie désirée **S** et la sortie effectivement obtenue **Y**.

L'algorithme est le suivant :

- Présenter au réseau un exemple **(E,S)**
- Calculer la sortie obtenue : $Y_j = \sum_i e_i . w_{i\ j}$
- Pour chaque neurone, calculer : $w_{i\ j}^{t+1} = w_{i\ j}^{t} + \eta . \left(S_j - Y_j\right)$

Avec : ***wi j*** : poids de la connexion entre le neurone **i** et le neurone **j**

Sj : sortie désirée pour le neurone **j**

Yj : sortie obtenue pour le neurone **j**

η : coefficient représentant la vitesse d'apprentissage

t : temps

VII.2.3 - Apprentissage de Widrow-Hoff

Cet apprentissage est une extension du modèle du Perceptron à sorties binaires à des sortie réelles.

L'algorithme est le suivant :

- Affecter des valeurs aléatoires aux connexions du réseau.
- Présenter un vecteur d'entrée $X(x_1,\ldots,x_n)$.
- Calculer les sorties : $S_j = \sum_i x_i . w_{i\ j}$
- Calculer l'erreur de chaque neurone : $\varepsilon_j = Y_j - S_j$
- Modifier le poids des neurones : $w_{i\ j}^{t+1} = w_{i\ j}^{t} + \eta . x_i . \varepsilon_j$

Avec : ***wi j*** : poids de la connexion entre le neurone **i** et le neurone **j**

Sj : sortie désirée pour le neurone **j**

Yj : sortie obtenue pour le neurone **j**

η : coefficient représentant la vitesse d'apprentissage

VII.2.4 - Apprentissage par rétropropagation

L'apprentissage par rétropropagation est une généralisation de la règle de Widrow-Hoff.

Le principe est de rétropropager l'erreur commise en sortie vers les couches internes du réseau (figure 11).

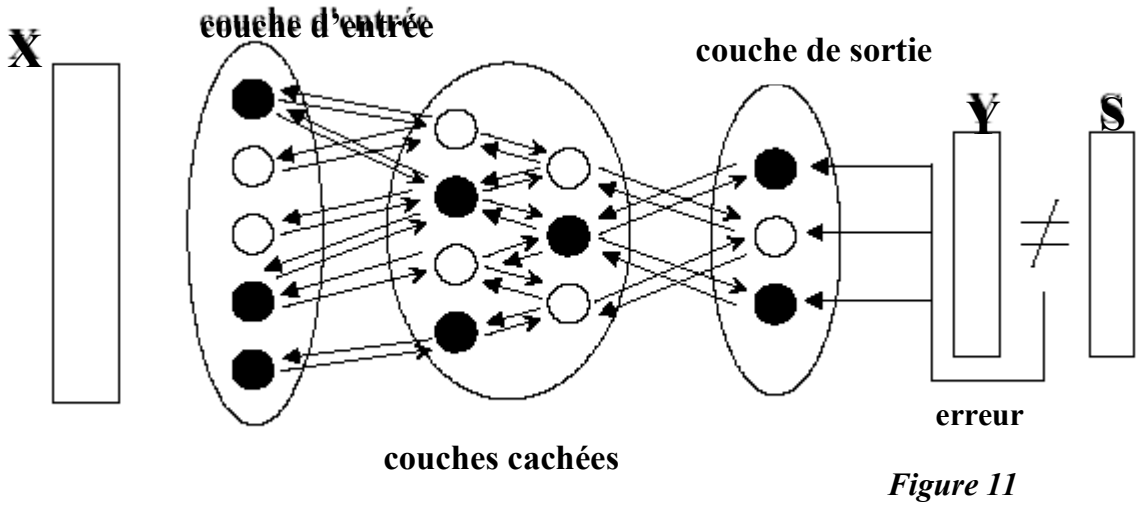


Figure 11

Le calcul de l'erreur commise pour le résultat à un exemple k est donné par

$$E^k(W)=\sum_{i=1}^{m}(Y_i^k-S_i^k)^2$$

Avec : ***Sj*** : sortie désirée pour le neurone **j**

Yj : sortie obtenue pour le neurone **j**

E(W) : erreur quadratique

W : matrice des connexions

On effectue une modification du poids des connexions à chaque exemple **k**

$$w_{i\ \ j}^{k}=w_{i\ \ j}^{k-1}-e(k).d_i.O_j$$

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

Pour les couches cachées : $d_i=(\sum_h d_h.w_{i\ \ h}).f'(I_i)$.

Avec :

h : ensemble des neurones vers lesquels le neurone **i** envoi un signal

f : fonction sigmoïde, **f'** est sa dérivée

Oj : sortie du neurone **j**

Ii : entrée du neurone **i**,

e(k) : pas du gradient à l'étape **k**

VIII - CONCLUSION

Les systèmes à base de réseaux de neurones sont de plus en plus utilisés dans de nombreux secteurs et de nombreux domaines (reconnaissance des formes, de la parole, traitement de l'information, robotique, identification, commande, diagnostic…).

Ils peuvent s'appliquer dans la plupart des cas à tous les systèmes de reconnaissance et d'approximation de données, et en général dans tous les systèmes où l'apprentissage s'avère indispensable. En effet, nous avons vu que grâce à sa propriété d'apprentissage, un réseau de neurones peut changer son comportement d'après les résultats de son expérience passée.

Dans la suite de ce rapport, nous présentons l'utilité des réseaux de neurones pour la détection et l'isolation des défauts physiques de la machine asynchrone.

I - Bibliographie

[1] Adrian Biran, Moshe breiner, « MATLAB pour l'ingénieur Version 6et 7 » Edition PEARSON Education, Août 2004.

[2] Ahmed Rachid « Techniques de l'Ingénieur, Régulation électromécanique ».

[3] D.Maquin, J.Ragot, « Diagnostic des systèmes linéaires », Collection pédagogique d'automatique, Hermes, 2000.

[4] Guy Seguier, Francis Notlet « Electrotechnique Industrielle ».

[5] H.Balaska, S.Karek « Diagnostic de défauts d'une machine asynchrone basé sur les observateurs non linéaires robustes », Mémoire d'ingénieur, U- SKIKDA 2004.

[6] H.Kaddid, T.Trirat, « Utilisation des réseaux de neurones pour la génération des résidus et d'un schéma neuro-flou pour l'isolation des capteurs défaillants des systèmes dynamiques », Mémoire d'Ingénieur, U – SKIKDA 2004.

[7] Hansruedi Buhler, « Réglage par logique floue », Collection électricité, Presses polytechniques et universitaires romandes, Lausanne, 1994.

[8] Lotfi Baghli, « Contribution à la commande de la machine asynchrone, utilisant de la logique flou, des réseaux de neurones et des algorithmes génétiques », Thèse de doctorat,U- Henri Poincaré-Nancy-1, 1999.

[9] M.L.Benloucif, L.Mehenaoui, «Approche Neuro-floue pour le Diagnostic de défauts », 2ND Instrumentation and Measurement in Petroleum Applications Conference IMPAC-2002, Boumerdès, 2002.

[10] M.Mokhtari, M.Marie, « Applications de MATLAB 5 Simulink 2», Edition Springer, 1998.

[11] Magnus Nargaard , O. Ravn, N.K.Poulsen and L.K Hansen « Neural Network for modelling and control of dynamic systems ».

[12] N.M.Bouloudani, L.Memiche, « Détection et isolation de défauts par observateurs pour des systèmes non linéaires », M.I ,CU-Skikda , 2000.

[13] Y. Morere « Identification par réseaux de neurones » Mémoire D.E.A, U- de VALENCIENNES ET DU HAINAUT – CAMBRESIS, 1995-1996.

J - Divers