

**RÉPUBLIQUE ALGERIENNE DÉMOCRATIQUE ET POPULAIRE**  
MINISTÈRE DE L'ENSEIGNEMENT SUPÉRIEUR ET DE LA RECHERCHE SCIENTIFIQUE

**UNIVERSITÉ M'HAMED BOUGARA - BOUMERDES**  
**FACULTÉ DES SCIENCES DE L'INGÉNIEUR**  
**DÉPARTEMENT MAINTENANCE INDUSTRIELLE**



## **MEMOIRE DE MAGISTER**

Filière : Maintenance Industrielle

Option : Maintenance des Systèmes Mécaniques

Intitulé du Sujet :

*Diagnostic des défaillances dans les systèmes industriels*

Présenté par : **BENAMMAR Samir**

Encadré par : Pr. **D. BENZAZZOUZ**

*Soutenu le : 12 .06 .2010*

*Devant le jury composé de :*

<b>Mr. REFOUFI Larbi</b>	<b>Professeur</b>	<b>UMBB</b>	<b>Président</b>
<b>Mr. BOUKHETALA Djamel</b>	<b>Professeur</b>	<b>ENP</b>	<b>Examineur</b>
<b>Mr. ADJERID Smail</b>	<b>Maître de conférences/A</b>	<b>UMBB</b>	<b>Examineur</b>
<b>Mr. KHELDOUN Aissa</b>	<b>Maître de conférences/B</b>	<b>UMBB</b>	<b>Examineur</b>
<b>Mr. BENZAZZOUZ Djamel</b>	<b>Professeur</b>	<b>UMBB</b>	<b>Rapporteur</b>

**ANNÉE UNIVERSITAIRE 2009-2010**

## REMERCIEMENT

*Ces travaux de mémoire de magistère ont été réalisés au sein du Laboratoire de mécanique des solides et systèmes (LMSS) à la faculté des sciences de l'ingénieur de Boumerdes dirigé par le Professeur A. SERIDI.*

*Je remercie tout d'abord « ALLAH » qui m'a donné la force et le courage pour l'achèvement de ce travail, ainsi que mes parents et toute ma famille qui m'ont apporté tout le soutien nécessaire ainsi que leur réconfort, et qui n'ont pas cessé de m'aider durant toute la période de la réalisation de ce mémoire.*

*Je tiens à remercier plus particulièrement mon directeur de mémoire Monsieur BENZAOUZ Djamel, Professeur à l'Université de M'hamed Bougara de Boumerdes, pour la confiance qu'il m'a témoigné et pour la compréhension et la gentillesse qu'il m'a exprimé pour mener à bien et parachever dans les meilleures conditions ce travail.*

*Je remercie également Monsieur SERIDI Ahcene, Professeur à l'Université de M'Hamed Bougara de Boumerdes et Directeur du Laboratoire de mécanique des solides et systèmes (LMSS) pour les conditions optimales de travail qu'il m'a apporté au sein du laboratoire.*

*Je remercie messieurs les membres du jury pour la caution qu'ils ont bien voulu apporter à ce travail. J'adresse mes remerciements aux :*

*Monsieur RAFOUFI Larbi, Professeur à l'Université de M'Hamed Bougara de Boumerdes et Directeur du Laboratoire du traitement du signal de l'INELEC pour m'avoir fait l'honneur de bien vouloir participer au jury en tant que président, mais également pour tout l'intérêt qu'il a manifesté sur ce travail de recherche,*

*Monsieur ADJRID Smail, Maître de Conférences à l'Université de M'Hamed Bougara de Boumerdes, pour tout l'effort et l'enthousiasme qu'il m'a consacré, la confiance qu'il m'a accordé et tous les conseils qu'il a su me donner, qu'il trouve ici l'expression de mes sincères remerciements,*

*Monsieur BOUKHETALA Djamel, Professeur à l'ENP et Monsieur KHELDOUN Aissa, Maître de Conférences à l'Université de M'Hamed Bougara de Boumerdes, pour m'avoir fait l'honneur de bien vouloir accepter de participer au jury en tant qu'examineurs de mon travail, je les remercie vivement pour l'intérêt qu'ils ont manifesté à mes travaux, pour leurs grandes qualités humaines.*

*Je ne peux omettre de dire merci à mes chers amis avec qui j'ai passé de très agréables moments dont je garde de très beaux souvenirs : Tahar, Moh, Nounou, Hamza, Belkacem, Amine, Toufik, Hicham, Moh, Sofien, Said, Bouhou, Migri, Zaki, Benalya,...*

*Sincèrement*

**RESUME**

Le présent travail a pour but de montrer l'utilisation des réseaux de neurones artificiels de type PMC (perceptron multicouches) dans le domaine de la surveillance des systèmes industriels complexes, au temps réel. Cette étude est décomposée en deux parties principales : dans la première partie, nous avons proposé des configurations neuronales optimales pour la détection et la localisation des défaillances (FDI) dans les turbines à vapeur. Les entrées du réseau sont les valeurs numériques obtenues à partir des capteurs de surveillance placés dans différents points de la turbine à vapeur, les sorties du réseau représentent le mode de fonctionnement et les points défaillants de la turbine dans le cas d'une anomalie. Dans la deuxième partie, nous avons proposé une approche simple, pour le développement d'un système de diagnostic, basée sur la combinaison entre les méthodes déterministes des arbres de défaillances et la méthode des réseaux de neurones artificiels. Dans ce cas, le réseau de neurone possède deux types de données, des données fixes obtenues à partir des arbres de défaillances et des données variables obtenues à partir des capteurs de surveillance pour sélectionner, à la sortie du réseau, la cause probable d'une défaillance.

**MOTS-CLÉS :** Défaillance, Détection, Diagnostic, Localisation, Réseaux de neurones artificiels, Surveillance, Systèmes industriels.

**ABSTRACT**

The purpose of this work is to show the use of the artificial neural networks MLP (multi-layer perceptron) in the supervision field, at real-time, of the complex industrial systems. This study is decomposed into two parts, in the first part, we proposed optimal neuronal configurations for the failures detection and isolation (FDI) on the steam turbines. The network inputs are the numerical values obtained starting from the supervision sensors placed in various points of a steam turbine and the network outputs represent the operating process and the failing points of the turbine in the anomaly case. In the second part, we proposed a simple approach for the development of a diagnosis system based on the combination between the deterministic methods of the failures trees and the artificial neural networks method. In this case, the neural network has two data types; the fixed data obtained starting from the failures trees and the variable data obtained starting from the supervision sensors, to select at the exit of the network the probable failure cause.

**KEY WORDS:** Failure, Detection, Diagnosis, isolation, artificial neural networks, Monitoring, Industrial systems.

**ملخص**

هذا العمل يهدف شرح استخدام الشبكات العصبية الاصطناعية MLP في مجال مراقبة النظم الصناعية في وقت حقيقي . ه قسمين رئيسيين : الجزء الأول ، اقترحنا تكوينات عصبية مثل لكشف وتحديد مكان حدوث الأعطال التربينات البخارية ، باعتبار الشبكة هي القيم العددية التي تم الحصول عليها من أجهزة الاستشعار التي وضعت في أجزاء التربينات البخارية ، الشبكة يمثل طريقة عملها ونقاط التربينات نهج بسيط لتطوير نظام تشخيص قائم على الجمع بين الأساليب القطعية من شجرة الأعطال وأساليب الشبكات العصبية الاصطناعية. في هذه ية نوعين من البيانات، البيانات الثابتة التي تم الحصول عليها من شجرة الأعطال والبيانات المتغيرة التي تم الحصول عليها من أجهزة الاستشعار

: تشخيص، تحديد الشبكات العصبية الاصطناعية، النظم الصناعية.

**Remerciements****Résumé****Abstract****ملخص****Sommaire****Notations et abréviations****INRODUCTION GÉNÉRALE**

1

**CHAPITRE I : METHODES DE DIAGNOSTIC INDUSTRIEL ET DE SURVEILLANCE****INTRODUCTION**

4

**I-1 DEFINITIONS GENERALES**

4

- *Processus industriel* 4
- *Nature des systèmes et des composants* 4
- *Définition du dispositif* 5
- *Décomposition fonctionnelle et matérielle* 5
- *Défaillance* 5
- *Faute (ou défaut)* 5
- *Panne* 5
- *Symptômes* 5
- *Contraintes* 6
- *Erreur* 6
- *Spécifications (cahier des charges)* 6
- *Surveillance* 6

**I-2 LES DIFFÉRENTES ÉTAPES TECHNIQUES DU DIAGNOSTIC**

6

## I-2.1 L'extraction des informations nécessaires

6

## I-2.2 Validation des mesures

6

## I-2.3 Caractérisation du fonctionnement

7

## I-2.4 Détection

7

## I-2.5 Diagnostic

7

## I-2.6 Décision

7

**I-3 CLASSIFICATION DES METHODES DE SURVEILLANCE**

8

## I-3.1 Problématique du diagnostic, État de l'art

8

## I-3.2 Méthodes de surveillance avec modèle

9

## I-3.2.1 Les méthodes de diagnostic par modélisation fonctionnelle et matérielle

9

## I-3.2.2 Méthodes de surveillance par modélisation physique

9

## I-3.3 Méthodes de surveillance sans modèle

10

## I-3.3.1 Surveillance en utilisant des outils statistiques du traitement du signal

10

## I-3.3.2 Surveillance par intelligence artificielle

10

## A) Les méthodes à base de modèles comportementaux

10

- *Les automates d'états finis*

11

- *Les réseaux de Petri*

11

- *Autres formalismes*

11

## B) Méthodes de reconnaissance de formes pour la surveillance

12

- *Systèmes experts*

12

- *Outils statistiques de reconnaissance de formes*

13

- <i>Raisonnement à partir de cas</i>	13
- <i>Reconnaissance de formes par logique floue</i>	14
- <i>Reconnaissance de formes par réseaux neuronaux</i>	14
C) Les méthodes à base de modèles explicatifs	15
- <i>Les graphes causaux</i>	15
- <i>Les graphes contextuels</i>	16
CONCLUSION	18
<b>CHAPITRE II : GENERALITES SUR LES RESEAUX DE NEURONES</b>	
INTRODUCTION	19
II-1 LES RESEAUX DE NEURONES ARTIFICIELS	19
II-1.1 Définition	19
II-1.2 Historique	20
II-1.3 Architectures de réseaux	21
II-1.3.1 Réseaux Non Boucles	22
II-1.3.2 Réseaux Récurrents	23
II-1.4 Apprentissage	24
II-1.4.1 Choix du taux d'apprentissage	26
II-1.4.2 Valeurs initiales des poids	26
II-1.4.3 Test d'arrêt	26
II-1.4.4 Choix de réseau	26
II-1.5 Apprentissage du Perceptron simple	26
II-1.6 Apprentissage des réseaux multicouche	27
II-1.6.1 Algorithme de descente de gradient	28
II-1.6.2 Algorithme de Levenberg-Marquardt	32
II-2 LES DIFFERENTS TYPES DE RESEAUX ET LEUR UTILISATION	35
CONCLUSION	37
<b>CHAPITRE III : ETUDE DE CAS - DETECTION ET LOCALISATION DES DEFAILLANCES D'UNE TURBINE A VAPEUR</b>	
INTRODUCTION	38
III-1 DESCRIPTION DE LA TURBINE	38
III-1.1 Rôle	38
III-1.2 Caractéristiques	39
III-1.3 Fonctionnement de la turbine	40
III-1.4 Positionnement des capteurs de surveillance	40
III-2 ORGANIGRAMME POUR LE CALCUL DE L'ARCHITECTURE OPTIMALE	42
III-3 SURVEILLANCE DE LA VITESSE DE ROTATION DE LA TURBINE	43
III-3.1 Données du constructeur	43
III-3.2 Méthodologie de travail	43
III-3.3 Explication des résultats	45
III-3.3.1 Apprentissage du réseau	45
III-3.3.2 Simulation des sorties de réseau	46
III-3.3.3 Test du réseau	46
III-3.3.4 Performances du réseau	48
III-3.3.5 Généralisation du réseau	48
III-4 SURVEILLANCE DE LA TEMPERATURE D'HUILE DANS LES SORTIES PALIERS	49

III-4.1	Méthodologie du travail	49
III-4.2	Explication des résultats	50
III-4.2.1	Apprentissage du réseau	50
III-4.2.2	Simulation des sorties du réseau	51
III-4.2.3	Détection et localisation des défauts (test du réseau)	51
III-4.2.4	Évaluation des performances	54
III-4.2.5	Généralisation du réseau	54
III-5	RÉSUMÉ DE TOUS LES RÉSULTATS ET COMMENTAIRES	55
	CONCLUSION	58

## **CHAPITRE IV : ETUDE DE CAS - DEVELOPPEMENT D'UN SYSTEME D'IDENTIFICATION DE LA CAUSE PROBABLE DES DEFAILLANCES**

	INTRODUCTION	59
IV-1	ARBRE DE DEFAILLANCES OU ARBRE DE CAUSES	60
IV-1-1	Événement Redouté ER1 : augmentation de vibrations	61
IV-1-2	Événement Redouté ER2 : perturbation du débit d'huile	62
IV-1-3	Événement Redouté ER3 : problème de température d'huile des sorties paliers	62
IV-1-4	Événement Redouté ER4 et ER5: problème de vitesse de rotation et de température de vapeur	63
IV-1-5	Événements Redoutés ER6 : problème de pression vapeur	64
IV-2	CAUSES PROBABLES DE DEFAILLANCE	65
IV-3	DIAGNOSTIC DE LA PREMIERE SERIE DE CAUSES	66
IV-3.1	Méthodologie de travail	66
IV-3.2	Explication des résultats	67
IV-3.2.1	Apprentissage du réseau	67
IV-3.2.2	Simulation des sorties de réseau	67
IV-3.2.3	Évaluation des performances	68
IV-3.2.4	Généralisation du réseau	68
IV-3.2.5	Exemples de test pour l'identification des causes de défaillances	69
IV-4	DIAGNOSTIC DE LA DEUXIEME SERIE DE CAUSES	72
III-4.1	Méthodologie de travail	72
IV-4.2	Explication des résultats	73
IV-4.2.1	Apprentissage du réseau	73
IV-4.2.2	Simulation des sorties du réseau	74
IV-4.2.3	Evaluation des performances	74
IV-4.2.4	Généralisation du réseau	74
IV-4.2.5	Exemples de test pour l'identification des causes de défaillances	75
	CONCLUSION	78

## **CONCLUSION GENERALE ET PERSPECTIVES** 79

### **BIBLIOGRAPHIE**

#### **ANNEXE I SCHEMAS D'UN SYSTEME DE SUPERVISION**

#### **ANNEXE II SCHEMAS DE TURBINE A VAPEUR**

#### **ANNEXE III COMPLEMENT DES RESULTATS OPTIMISES DU CHAPITRE III**

***Notations et abréviations***

AMDEC	Analyse des Modes de Défaillances de leurs Effets et leur Criticité.
ADD	Arbre De Défaillances.
ART	Théorie de Résonance Adaptative (Adaptive Resonance Theory)
BP	Basse Pression
BPN	Behavioral Petri Net.
Dhp1 (dhp1)	Débit d'huile dans le palier 1
Dhp2	Débit d'huile dans le palier 2
Dhp3	Débit d'huile dans le palier 3
Dhp4	Débit d'huile dans le palier 4
FDI	Fault Detection and isolation
HP	Haute Pression
IA	Intelligence Artificielle.
LF	Logique Floue.
LM	Levenberg-Marquardt
MLP	Perceptron Multi-Couches (Multi Layer Perceptron)
MP	Moyenne Pression
MSE	Erreur Quadratique Moyenne
RàPC	Raisonnement à Partir de Cas. (Case Based Reasoning – CBR)
RBF	Fonctions à Base Radiale (Radial Basis Functions)
RdP	Réseau de Petri
R	Coefficient de corrélation du réseau de neurone
RG	Coefficient de corrélation du réseau de neurone généralisé
RN	Réseaux de Neurones
RNF	Réseau Neuro-Floue.
Pehp (pvp1)	Pression de la vapeur à l'entrée corps HP
Pemp	Pression de la vapeur à l'entrée du corps MP

Pshp	Pression de la vapeur à la sortie du corps HP
Psbp	Pression de la vapeur à la sortie du corps BP
Sdeh (spvp1)	Sortie désirée de la pression vapeur à l'entrée corps HP
Sdsh	Sortie désirée de la pression vapeur à la sortie du corps HP
Sdmp	Sortie désirée de la pression vapeur à l'entrée du corps MP
Sdbp	Sortie désirée de la pression vapeur à la sortie du corps BP
Sdh1 (sdhp1)	Sortie désirée du débit d'huile dans le palier 1
Sdh2	Sortie désirée du débit d'huile dans le palier 2
Sdh3	Sortie désirée du débit d'huile dans le palier 3
Sdh4	Sortie désirée du débit d'huile dans le palier 4
SOM	Carte auto-organisatrice de Kohonen (Kohonen Self-Organising Maps)
STHP1 (sthp1)	Sortie désirée de la température d'huile à la sortie du palier 1
STHP2	Sortie désirée de la température d'huile à la sortie du palier 2
STHP3	Sortie désirée de la température d'huile à la sortie du palier 3
STHP4	Sortie désirée de la température d'huile à la sortie du palier 4
STV1	Sortie désirée de la température de vapeur à la sortie basse pression
STV2	Sortie désirée de la température de vapeur à la sortie haute pression
STV3	Sortie désirée de la température de vapeur à l'entrée moyenne pression
STV4 (stvp1)	Sortie désirée de la température de vapeur à l'entrée haute pression.
svitp1	Sortie désirée de la vitesse de rotation de la turbine (palier 1)
Svp1 (svibp1)	Sortie désirée des vibrations relatives de l'arbre du palier 1
Svp2	Sortie désirée des vibrations relatives de l'arbre du palier 2
Svp3	Sortie désirée des vibrations relatives de l'arbre du palier 3
Svp4	Sortie désirée des vibrations relatives de l'arbre du palier 4
Tap	Temps d'apprentissage
Tvsb	Température de vapeur à la sortie basse pression
Tvsh	Température de vapeur à la sortie haute pression

Tvem	Température de vapeur à l'entrée moyenne pression
Tveh (tvp1)	Température de vapeur à l'entrée haute pression.
THP1 (thp1)	Température d'huile à la sortie du palier 1
THP2	Température d'huile à la sortie du palier 2
THP3	Température d'huile à la sortie du palier 3
THP4	Température d'huile à la sortie du palier 4
vitp1	Vitesse de rotation de la turbine (palier 1)
Vrp1 (vibp1)	Vibrations relatives de l'arbre du palier 1
Vrp2	Vibrations relatives de l'arbre du palier 2
Vrp3	Vibrations relatives de l'arbre du palier 3
Vrp4	Vibrations relatives de l'arbre du palier 4

# **INTRODUCTION GÉNÉRALE**

## INTRODUCTION GENERALE

Au début, la maintenance était destinée à remplir essentiellement sa fonction de réparation, c'est-à-dire, remettre en fonctionnement l'équipement défaillant sans pouvoir tenir compte des autres considérations économiques, techniques, sociales et de sécurité. Dans les dernières années, surtout avec l'évolution technologique des systèmes industriels dans plusieurs domaines de l'aéronautique, du nucléaire, de la pétrochimie etc, les industriels ont réfléchi à améliorer la maintenance de leur système de production. Un type plus important de la maintenance, est apparu dans cette période, représenté par la maintenance conditionnelle. Cette dernière est basée essentiellement sur les informations des signatures obtenues à partir des capteurs de surveillance. L'un des points sur lesquels les travaux de recherche sont orientés est la maintenance conditionnelle dans laquelle sont menés des travaux sur la surveillance et le diagnostic industriels.

Le rôle premier de la surveillance industrielle est d'augmenter la disponibilité des installations industrielles afin de réduire les coûts directs et indirects de la maintenance des équipements de production. Les coûts directs de cette maintenance sont ceux relatifs aux diverses pièces de rechange, main d'œuvre, etc. Par contre, les coûts indirects sont essentiellement dus au manque à gagner engendré par un arrêt de production.

La **surveillance industrielle** qui représente la partie principale de la maintenance se décompose en deux fonctions essentielles :

- **la détection** d'une défaillance qui consiste à reconnaître une déviation par rapport au fonctionnement attendu. Elle peut être signalée par une alarme ;
- **le diagnostic** d'une faute qui permet de déterminer la localisation de l'organe défaillant et l'identification des causes de la défaillance. Celui-ci se décompose à son tour en deux parties :

- la localisation d'une faute qui revient à déterminer la région physique du système dans laquelle la défaillance s'est produite ;
- l'identification de la cause qui permet de déterminer précisément l'origine de la défaillance détectée. [DAN 2006]

En **détection** et maintenance, les spécialistes de maintenance sont amenés à connaître et à classer les signatures obtenues à partir des capteurs de plus souvent de façon visuelle. De façon similaire, lors du fonctionnement normal d'un processus complexe tel qu'une centrale thermique de production d'électricité ou centrale nucléaire, les centaines d'informations visuelles délivrées par les indicateurs repartis sur les différents panneaux de la salle de commande constituent de formes visuelles que les opérateurs utilisent pour identifier les modes opératoires. Quand une anomalie du fonctionnement survient, les informations se modifient d'une façon dynamique. La tâche des opérateurs consiste alors à identifier la situation et utiliser les procédures de conduite pour remettre l'installation dans une configuration de repli sûre. Les opérateurs mettent en œuvre d'une façon consciente ou inconsciente de la reconnaissance visuelle de forme.

En **diagnostic**, les spécialistes sont conduits à identifier le mode de fonctionnement du système défaillant. Les premières méthodes de diagnostic furent basées sur la redondance de matériels jugés critiques pour le fonctionnement du système. La redondance matérielle est très répandue dans les domaines où la sûreté de fonctionnement est cruciale pour la sécurité des personnes et de l'environnement, comme dans l'aéronautique ou le nucléaire. Les principaux inconvénients de la redondance matérielle sont liés aux coûts dus à la multiplication des éléments ainsi que l'encombrement et aux poids supplémentaires qu'elle génère. Le développement des calculateurs numériques permet aujourd'hui la mise en œuvre, dans le milieu industriel, des méthodes modernes de l'automatique et de l'intelligence artificielle.

La démarche classique de diagnostic consiste à trouver le (ou les) défaut(s) le(s) plus plausible(s) à partir des comportements observés et des informations suivantes :

- manifestations extérieures des défauts ou « symptômes » (fréquences ou contenu spectral, types d'évolution, localisation...);
- contexte dans lequel apparaît le défaut, (conditions de fonctionnement, manœuvres d'exploitation, corrélations) ;
- connaissance de la machine, technologies favorisant, historique d'incidents.

La démarche se décline en trois étapes complémentaires ; « identification, évocation, confirmation » [JAC 2001]. Si on fait une analyse approfondie concernant cette démarche classique de diagnostic, on peut déduire plusieurs inconvénients. L'un de ces inconvénients est représenté par l'augmentation du temps de localisation de la défaillance et puis une augmentation des coûts indirects de la maintenance. C'est la raison qui nous a poussé à réaliser un système de diagnostic efficace et rapide, fait à programmer toutes ces démarches dès le début de l'anomalie jusqu'au le dernier point (identification de la cause), basant sur les réseaux de neurones artificiels à plusieurs sorties et les méthodes déterministes (arbre de défaillances). Ce système facilite bien aux agents de maintenance d'intervenir directement sans passer par les démarches précédentes.

Les méthodologies de surveillance se basent sur deux concepts : méthodes de surveillance en l'absence de modèles du procédé, et méthodes de surveillance avec modèles. Le modèle d'un système est généralement difficile à obtenir, surtout pour des systèmes complexes soumis à des aléas ou reconfigurables. [NIC 2005]

Les techniques de surveillance sans modèle sont divisées en deux parties. La première partie correspond aux outils statistiques et de traitement du signal qui sont généralement qualifiés d'outils de traitement de bas niveau, parce qu'ils sont en contact direct avec le signal capteur, et ne servent généralement que pour la génération d'alarmes brutes, sans aucune information concernant leur signification. La deuxième partie est celle des techniques dites de haut niveau et qui sont plutôt orientées vers la communication avec l'expert. Celles-ci représentent les techniques de l'Intelligence Artificielle (*IA*) et servent comme outil de base pour l'aide à la décision. [MOH 2004]

A la base de ces travaux, on se propose dans ce mémoire un système de surveillance basé essentiellement sur deux stratégies ; la première est reposée sur l'intelligence artificielle connue sous le nom « les réseaux de neurones artificiels » et la deuxième est descriptive désignée par « les arbres de défaillances ». C'est système hybride de diagnostic, le choix d'un tel système est basé sur deux critères qui sont :

- Capacité et rapidité d'apprentissage des réseaux neuronaux ;
- Simplicité des arbres de défaillances à réaliser, car ils ne nécessitent pas beaucoup de calculs.

L'application du système hybride de diagnostic a été faite sur un cas réel d'une turbine à vapeur de la centrale thermique de Cap-Djinet (SONELGAZ). Cette turbine est munie déjà par un système de surveillance mais, ce dernier n'est pas capable de localiser et d'identifier les causes probables de défaillances. Ce système de surveillance fait la détection des anomalies par un déclenchement d'alarmes obtenues par les capteurs de surveillance placés dans des différents points de la turbine. C'est pour cette raison, que nous avons essayé d'équiper ce système de détection par un autre système de diagnostic pour accomplir la fonction requise de la surveillance.

Ce mémoire est composé d'une introduction générale qui met en exergue l'objectif du travail. Le premier chapitre traite les concepts généraux du diagnostic et des définitions de différents points de la surveillance. L'ensemble enrichi par des schémas explicatifs permettant de citer les différentes méthodes de diagnostic avec des descriptions brèves pour chaque méthode.

Dans le deuxième chapitre nous présentons la stratégie de diagnostic avec laquelle nous avons abordé le problème. Cette approche est basée sur les réseaux de neurones artificiels (RNA). Nous présentons l'ensemble de la théorie des RNA avec les algorithmes d'apprentissage retenues dans l'approche des systèmes hybrides de diagnostic.

Le troisième chapitre représente la détection et la localisation des défauts (FDI) dans les systèmes industriels. Cette approche est appliquée pour les paramètres à surveiller de la turbine pour la détermination des points défaillants.

Le dernier chapitre concerne la localisation des défauts en identifiant les causes probables de ce défaut. L'approche est basée sur la méthode hybride en combinant à la fois les réseaux de neurones artificiels et l'arbre de défaillance. Se basant sur la simulation, on a fait des scénarios de défaillance sur la turbine pour voir la réaction du système développé s'il réagit correctement à la détection et localisation du défaut et donnant les causes probables de ce défaut. Enfin nous finissons notre travail par une conclusion générale dans laquelle nous évaluons les différents résultats obtenus.

**CHAPITRE I : METHODES DE DIAGNOSTIC  
INDUSTRIEL ET DE SURVEILLANCE**

## INTRODUCTION

Dans un grand nombre d'applications industrielles, une demande croissante est apparue en matière de remplacement des politiques de maintenance curative par des stratégies de maintenance préventive. Cette mutation d'une situation où on « subit les pannes » à une situation où on « maîtrise les pannes », nécessite quelques moyens technologiques ainsi que la connaissance de techniques d'analyse appropriées. La fonction surveillance en continu de l'évolution de l'équipement à travers des données quantifiables et qualifiables permet ainsi de prévenir un dysfonctionnement avant qu'il n'arrive et d'écartier les fausses alarmes qui peuvent ralentir la production [BAS 1996]. De nombreux auteurs ont abordé le domaine de la surveillance industrielle mettant ainsi en évidence l'intérêt croissant manifesté par la communauté scientifique et les industriels par rapport à cette problématique. Sans souci d'exhaustivité, nous pouvons citer les travaux suivants : [COM 1991], [DEV 1991], [TOG 1992], [POU 1996], [CUS 1996], [EVS 1998], [WEB 1999], [ZHA 1999], [COB 2000], [LEF 2000].

### I-1 DEFINITIONS GENERALES

#### - Processus industriel

Un processus industriel est une installation complexe assumant un objectif fonctionnel de haut niveau (production de bien ou de service). Pour assurer ces objectifs de haut niveau, le système fait appel à un ensemble de systèmes interconnectés. Chaque système assure une ou plusieurs fonctions bien définies [GIL 1995].

On dit aussi processus industriel, un assemblage fonctionnel de composants technologiques associés les uns aux autres de façon à former une entité unique accomplissant ou pouvant accomplir une activité clairement définie [BEL 2008].

En règle générale et en pratique ce sont sur les composants que l'on effectuera la maintenance et non sur les systèmes, puisque chaque système est décomposé en sous systèmes et chaque sous système est décomposé en composants bien déterminés.

#### - Nature des systèmes et des composants [GIL 1995]

Les méthodes de diagnostic ne possèdent pas de caractère universel. En fonction de la nature de processus, système, sous systèmes, composants, il faudra mettre en œuvre à chaque fois des méthodes spécifiques tenant compte des méthodes déployées.

#### - Technologies déployées

- Systèmes mécaniques dynamiques : moteurs, pompes, turbines, réacteur,....,
- Systèmes mécaniques statiques : tuyauterie, enceinte....,
- Systèmes mécaniques programmés,
- Systèmes thermodynamique : échangeurs, fours, colonne de distillation,....,
- Système électronique et analogique ou logique : capteur, régulateur, automate programmable...

Les fonctions assurées par le système possèdent une hiérarchie en termes d'importance et il convient toujours de s'interroger sur les aspects technico-économiques avant d'envisager la mise en œuvre d'une méthode de diagnostic.

Pour effectuer cette analyse il est nécessaire d'identifier les caractéristiques des systèmes et composants :

- Les fonctions de système en distinguant les missions principales et secondaires et leurs importances relatives,
- La structure du système en analysant les liens entre système et composants,
- Le mode de fonctionnement des systèmes et des caractéristiques des composants,

- Les conditions d'exploitation du système,
- L'environnement du système pour connaître des délimitations et l'influence des facteurs extérieurs,
- L'inventaire des moyens de mesure.

#### - Définition du dispositif

Selon AFNOR X06-501 il s'agit là au produit auquel s'applique l'étude :

- Composant (le plus petit constituant irréparable par l'utilisateur),
- Sous-ensemble,
- Ensemble.

#### - Décomposition fonctionnelle et matérielle

Une décomposition fonctionnelle fait appel à des fonctions principales, secondaires, techniques, de contraintes internes et externes.

Une décomposition matérielle prend en compte les notions de système sous-système, ensemble, sous-ensemble, composants.

#### - Défaillance

Modification suffisante et permanente des caractéristiques physiques d'un composant pour qu'une fonction requise ne puisse plus être assurée dans les conditions fixées. On distingue les défaillances :

- Naissantes,
- Ayant un caractère passager,
- Constantes,
- Évoluant dans le temps,
- Catastrophiques.

#### - Faute (ou défaut)

Déviations d'une variable observée ou d'un paramètre calculé par rapport à sa valeur fixée dans les caractéristiques attendues du processus lui-même, des capteurs, des actionneurs ou de tout autre équipement.

- **Défauts capteur** : Se caractérisent par un écart entre la valeur réelle de la grandeur et sa mesure.
- **Défauts d'actionneurs** : Ils se traduisent par une incohérence entre les commandes et la sortie (la pompe délivre un débit incohérent avec sa caractéristique hydraulique).
- **Défauts du processus physique** : Défaillances dues à des modifications de la structure (fuite, rupture d'un organe,...) ou des paramètres du modèle (encrassement d'un tube d'un four, bouchage d'un tube, ..).
- **Défauts du système (ou de l'algorithme) de commande** : Ils se caractérisent par un écart entre la valeur réelle de la sortie du contrôleur (selon l'algorithme implémenté) et sa mesure

#### - Panne

Interruption permanente de la capacité du système à réaliser sa fonction requise.

#### - Symptômes

Traductions d'un changement d'un comportement d'une variable détectée par comparaison à des valeurs de référence.

**- Contraintes**

Limitations imposées par la nature (lois physiques) ou l'opérateur.

**- Erreur**

Écart entre une valeur mesurée ou estimée d'une variable et la vraie valeur spécifiée par un capteur étalon ou jugée (par un modèle) théoriquement correcte.

**- Spécifications (cahier des charges)**

Objectifs que doit atteindre le système de surveillance.

**- Surveillance**

La surveillance est un dispositif passif, informationnel qui analyse l'état du système et fournit des indicateurs. La surveillance consiste notamment à détecter et classer les défaillances en observant l'évolution du système puis à le diagnostiquer en localisant les éléments défaillants et en identifiant les causes premières. La surveillance se compose donc de deux fonctions principales qui sont la détection et le diagnostic.

- **La détection** : Pour détecter les défaillances du système. Il faut être capable de classer les situations observables comme étant normales ou anormales.
- **Le diagnostic industriel** : « Le diagnostic est l'identification de la cause probable de la (ou des) défaillance(s) à l'aide d'un raisonnement logique fondé sur un ensemble d'informations provenant d'une inspection, d'un contrôle ou d'un test. » AFNOR, CEI

**Deux tâches essentielles en diagnostic :**

- Observer les symptômes de défaillance,
- Identifier la cause de la défaillance à l'aide d'un raisonnement logique fondé sur des observations.

**I-2 LES DIFFÉRENTES ÉTAPES TECHNIQUES DU DIAGNOSTIC**

Les étapes techniques, du diagnostic industriel nécessaires à la conception, au développement et à l'exploitation de systèmes d'aide au diagnostic. Sont définies dans la Figure 1.1 [GIL 1995].

**I-2.1 Extraction des informations nécessaires**

L'extraction des informations nécessaires à la mise en forme des caractéristiques associées aux fonctionnements normaux et anormaux, à partir des moyens de mesures appropriées ou d'observations réalisées hors des rondes par les personnels de surveillance.

Deux façons pour estimer la grandeur physique :

- **mesure directe** : Utilisation de capteurs,
- **mesure indirecte** : Estimateur d'état.

L'élaboration des caractéristiques et signatures associées à des symptômes révélateurs de défaillances et de dégradations en vue de la détection d'un dysfonctionnement.

**I-2.2 Validation des mesures**

C'est l'opération qui consiste à engendrer une information représentant une grandeur physique, qui sera considérée crédible et fiable par les utilisateurs : les opérateurs et les systèmes de diagnostic.

### I-2.3 Caractérisation du fonctionnement

L'élaboration des caractéristiques et signatures associées à des symptômes révélateurs de défaillance et de dégradation en vue de la détection d'un dysfonctionnement,

### I-2.4 Détection

La détection d'un dysfonctionnement par comparaison avec des signatures associées à des états de fonctionnement normaux et la définition des indicateurs de confiance dans la décision,

### I-2.5 Diagnostic

La mise en œuvre d'une méthode de diagnostic de la défaillance ou de la dégradation à partir de l'utilisation des connaissances sur les relations de la cause à l'effet,

### I-2.6 Décision

La prise de décision en fonction des conséquences futures des défaillances et des dégradations.

Cette prise de décision peut conduire à un arrêt de l'installation si les conséquences de la défaillance sont importantes pour la sécurité des personnes et des biens ou à une configuration de fonctionnement du procédé pour éviter une perte de production arrêt de production le plus propice aux opérations de maintenance corrective.

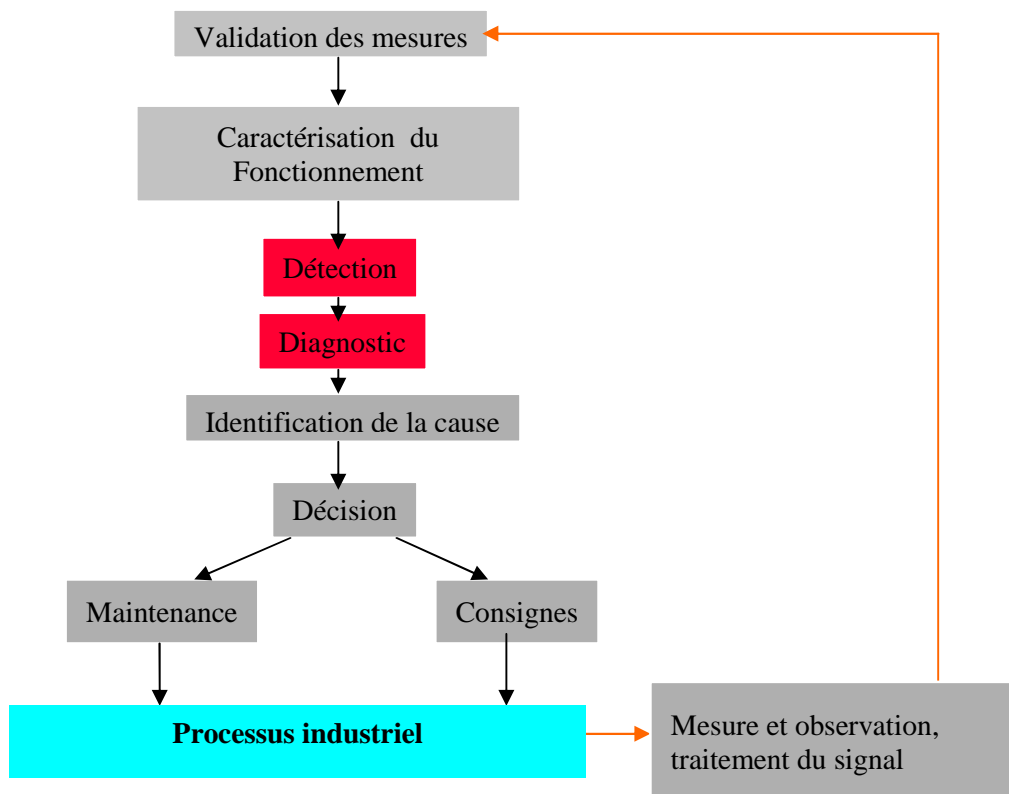


Figure 1.1: Les différentes étapes du diagnostic industriel

### I-3 CLASSIFICATION DES METHODES DE SURVEILLANCE

#### I-3.1 Problématique du diagnostic, État de l'art

Les méthodologies de surveillance sont généralement divisées en deux groupes : méthodologies de surveillance avec modèle et sans modèle [DAS 2000] ;

- Les premières se basent sur l'existence d'un modèle formel de l'équipement et utilisent généralement des techniques de l'Automatique [COM 1991].

- Les techniques de surveillance sans modèle sont divisées en deux parties. La première partie correspond aux outils statistiques de Traitement du Signal qui sont généralement qualifiés d'outils de traitement de bas niveau, parce qu'ils sont en contact direct avec le signal capteur, et ne servent généralement que pour la génération d'alarmes brutes, sans aucune information concernant leur signification.

Les outils statistiques établissent des tests sur les signaux d'acquisition, tests qui ne sont capables d'assurer que la fonction détection de défaillances. La deuxième partie est celle des techniques dites de haut niveau et qui sont plutôt orientées vers la communication avec l'expert [DAN 2006]. Celles-ci représentent les techniques de l'Intelligence Artificielle (IA) et servent comme outil de base pour l'aide à la décision. Leur réponse est donc plus élaborée que celle des techniques de bas niveau et elles sont capables de détecter, interpréter (association à un mode) et diagnostiquer les défaillances.

Notons que les deux approches Automatiques et Intelligence Artificielle peuvent être combinées pour profiter de certains avantages de chacune et avoir ainsi une certaine complémentarité [DUB 2001]. Les méthodes de l'Automatique sont par nature proches du système surveillé, puisqu'elles travaillent directement à partir des données issues des capteurs ; elles sont ainsi principalement utilisées pour la génération d'alarmes. Les méthodes de l'Intelligence Artificielle sont, elles, plus tournées vers la communication avec l'opérateur et se focalisent sur la transformation d'un ensemble d'informations brutes et non reliées entre elles en une information interprétable directement par l'opérateur chargé de la conduite ; elles sont donc utilisées pour l'interprétation des alarmes et l'aide à la décision [BAS 1996]. D'autres réflexions sur la complémentarité entre ces deux domaines peuvent être trouvées dans [DUB 2001].

Dans le cas de l'utilisation des outils de l'Intelligence Artificielle, la fonction surveillance est souvent vue uniquement comme une application de reconnaissance des formes ; les formes représentent le vecteur d'entrée composé par les différentes données de l'équipement (données mesurables et qualifiables), et les classes représentent les différents modes de fonctionnement. Une définition de référence donnée dans [DUB 2001] considère d'ailleurs le diagnostic de la manière suivante :

Un problème de diagnostic peut se définir comme un problème de reconnaissance de forme.

L'ensemble des états est homologué à un ensemble de classes et le vecteur forme est le vecteur des composantes des paramètres observés sur le système.

Dans notre travail, nous abordons le problème de la surveillance avec un aspect complémentaire qui nous paraît fondamental, notamment par rapport au diagnostic qui – comparé à la détection de défauts/défaillances – ne peut pas se résoudre uniquement à une reconnaissance de formes. La recherche des causes de la défaillance devient ainsi une démarche primordiale qui nécessite des techniques différentes.

Étymologiquement, le mot diagnostic vient du grecque et signifie (Dia : par, Gnosis : connaissance). La définition donnée par [PEN 1990], s'en inspire et se formule ainsi :

Étant donné un ensemble de manifestations observées (symptômes, constatations, etc.), il s'agit d'expliquer leur présence, de remonter aux causes, en utilisant un savoir sur le système considéré.

Nous considérerons cette définition pour la suite du travail, elle semble mieux adaptée aux problèmes de diagnostic dans la mesure où elle met en évidence le raisonnement qui doit

être suivi pour réaliser la fonction "comprendre" et l'approche considérée est largement reprise par d'autres auteurs, on peut citer notamment [GIL 1995], [GRO 2001], [DAN 2006]. De plus, la décomposition du diagnostic en deux fonctions : *localisation* et *identification de la cause*, s'applique également à cette définition. Enfin, le diagnostic est réalisé à partir d'observations, ce qui revient bien à considérer la détection comme ne faisant pas partie de la phase de diagnostic. Au regard de ces définitions, le diagnostic – de même que l'ensemble de la surveillance – traite à la fois des données numériques (exploitation des observations s'il s'agit de capteurs par exemple) et des données symboliques (connaissances sur le système considéré). Dans [DUB 1990] ces deux types de données sont considérés comme des connaissances nécessaires à l'opération de diagnostic. D'une part, on trouve la connaissance globale, que l'on peut qualifier de connaissance "a priori" sur le système et reposant sur le passé du système. D'autre part, une connaissance instantanée est nécessaire, correspondant à l'ensemble des éléments dont on dispose à un instant donné pour prendre une décision et l'exploiter. Une autre notion se dégage également de la définition de [PEN 1990]. Elle rend compte de l'exploitation des connaissances causales sur le système. En effet, un "dysfonctionnement" peut être décrit de façon simple par des relations associant ses causes initiales (défaillances de composants, etc.) à ses manifestations observables, les symptômes. Si on dispose d'une théorie modélisant de telles relations, un problème de diagnostic consiste à rechercher à l'aide de cette théorie des explications satisfaisantes aux symptômes observés. L'inférence de base mise en jeu dans ce type de raisonnement qui "remonte des effets aux causes" est appelée abductive. Etant donné le fait "B" et l'association (la relation de causalité) "A → B" ("A" cause "B"), inférer "A possible". Le diagnostic ainsi réalisé sera qualifié de Diagnostic Abductif [DAN 2006].

Un point très important pour le diagnostic est soulevé : dans quelle mesure peut-on considérer que les connaissances sur les systèmes qu'elles soient causales, globales ou instantanées sont suffisantes, et comment faut-il pour les acquérir et les valider ? Compte tenu de la complexité de la tâche, de nombreuses méthodes existent pour réaliser un diagnostic faisant appel à des techniques variées. La partie suivante s'attache à fournir une classification des méthodes de surveillance, en prenant en compte plus particulièrement les critères associés au diagnostic.

### **I-3.2 METHODES DE SURVEILLANCE AVEC MODELE**

La catégorie de méthodes la plus utilisée par les industriels à l'heure actuelle est vraisemblablement celle des méthodes par modélisation fonctionnelle et matérielle. Cette catégorie de méthodes comprend des outils industriels comme l'AMDEC et l'arbre de défaillances (ADD), utilisés essentiellement pour le diagnostic.

#### **I-3.2.1 Les méthodes de diagnostic par modélisation fonctionnelle et matérielle**

Le principe de ces méthodes consiste à établir a priori et de la manière la plus complète possible, les liens entre les causes initiales des défaillances et leurs effets mesurables.

Les méthodes les plus couramment rencontrées sont l'Analyse des Modes de Défaillance, de leurs Effets et de leurs Criticité (AMDEC) et les Arbres De Défaillances (ADD).

Une autre catégorie de méthodes avec modèles est représentée par les méthodes basées sur une modélisation physique des processus surveillés.

#### **I-3.2.2 Méthodes de surveillance par modélisation physique**

Les méthodes de surveillance avec modèle physique ont pour principe de comparer les mesures effectuées sur le système aux informations fournies par le modèle [FRA 1990]. Tout

écart est alors synonyme d'une défaillance. Les outils de la théorie de la décision sont ensuite utilisés pour déterminer si cet écart est dû à des aléas normaux, comme par exemple le bruit de mesure ou s'il traduit une défaillance du système. Les méthodes avec modèle physique sont les méthodes les plus familières aux automaticiens. En règle générale, ces méthodes peuvent être séparées en deux techniques : techniques de redondance physique et analytique, et techniques d'estimation paramétrique [MOH 2004].

### **I-3.3 METHODES DE SURVEILLANCE SANS MODELE**

Nombreuses sont les applications industrielles dont le modèle est difficile, voir impossible à obtenir suite à une complexité accrue ou à de nombreuses reconfiguration intervenants durant le processus de production. Pour ce type d'applications industrielles, les seules méthodes de surveillance opérationnelles sont celles sans modèle. Deux solutions existent dans ce cas : surveillance avec des tests statistiques du traitement du signal et surveillance par intelligence artificielle. [DUB 2001].

#### **I-3.3.1 Surveillance par utilisation des outils statistiques du traitement du signal**

Les outils statistiques de détection de défaillances consistent à supposer que les signaux fournis par les capteurs possèdent certaines propriétés statistiques. On effectue alors quelques tests qui permettent de vérifier si ces propriétés sont présentes dans un échantillon des signaux mesurés. Une grande variété de tests, applicables sur un échantillon de mesures. Parmi les plus importants, nous mentionnons : le test de franchissement de seuils, le test de moyenne et le test de variance [MOH 2004].

#### **I-3.3.2 Surveillance par intelligence artificielle**

Concernant cette catégorie de méthodes, une première approche consiste à considérer le diagnostic par reconnaissance de forme suivant une approche probabiliste, une approche floue ou une approche neuronale [DUB 2001]. Compte tenu du nombre et de la diversité des travaux réalisés en diagnostic en l'absence de modèle du procédé, cette classification des méthodes en l'absence de modèle du procédé est quelque peu réductrice. On trouve dans la littérature des classifications reflétant de manière plus juste les différentes approches sans modèle du procédé. Ces méthodes s'appuient largement sur les techniques de l'Intelligence Artificielle (IA) et font appel à des connaissances symboliques, familières ou au moins partageables par l'opérateur [BAS 1996]. En effet, l'utilisation de l'Intelligence Artificielle permet de pallier la complexité des systèmes à diagnostiquer. De plus, d'une manière générale, l'Intelligence Artificielle - de par ses caractéristiques -, est relativement bien adaptée aux problèmes du diagnostic. En effet, l'IA peut se caractériser par la capacité à traiter :

- une grande quantité d'informations,
- des données non homogènes (numériques/symboliques),
- des données dépendant du contexte,
- des données incomplètes.

Globalement, ces méthodes seront regroupées sous l'expression : "Méthodes par modélisation symbolique". [MON 2004].

#### **A) Les méthodes à base de modèles comportementaux**

Il existe deux approches principales pour la construction de ces modèles : les automates d'états finis et les réseaux de Petri (RdP).

- **Les automates d'états finis**

Ils permettent de modéliser directement le fonctionnement du système, grâce à un automate global obtenu par composition d'automates élémentaires correspondant à des systèmes locaux (composants du système). Cette représentation est donc directement adaptée à la simulation et à la détection. Il existe cependant des systèmes pour lesquels cette représentation est également utilisée pour le diagnostic. Deux étapes pour effectuer le diagnostic. Dans un premier temps, il s'agit de développer le modèle du système à l'aide des automates d'états finis, puis dans un deuxième temps, un "diagnostiqueur" correspondant également à un automate d'états finis est construit à partir du modèle global. Ce dernier effectue un diagnostic en observant en ligne une séquence d'événements. Pour chaque événement consécutif, le diagnostiqueur fournit une estimation de l'état du système et des événements non observés, d'où les occurrences des pannes sont déduites.

Une grande partie des études d'utilisations des automates d'états finis en diagnostic sont dédiées aux réseaux de télécommunications. [AGH 1998]

- **Les réseaux de Petri**

Le réseau de Petri est un outil mathématique et graphique adapté à un grand nombre d'applications où les notions d'événements et d'évolutions simultanées sont importantes. Ils constituent l'un des modèles les plus utilisés lorsqu'il s'agit de systèmes à événements discrets. Ils ont toutefois été enrichis sous plusieurs aspects (RdP temporisés, stochastiques, flous), de manière à mieux rendre compte de la dynamique des systèmes à événements discrets. Utilisés dans un premier temps comme modèles générateurs, ils permettent la réalisation de simulation ainsi que la détection dans une optique d'utilisation en diagnostic de systèmes. Dans ce cadre, les réseaux de Petri peuvent être qualifiés de modèle de bon fonctionnement. Un raisonnement de type chaînage arrière sur les réseaux de Petri est défini. Les réseaux de Petri considérés constituent un modèle de comportement du système à diagnostiquer. Il s'agit d'un modèle du comportement du système construit avec un réseau de Petri comportemental (BPN - Behavioral Petri Net) qui englobe éventuellement des états de pannes. Les BPN sont des réseaux saufs et déterministes. Des règles de tirage arrière sont définies et permettent de rendre compte d'éventuelles incohérences dans le raisonnement de recherche de causes. [ANG 1994]

Dans cette approche, le diagnostic est réalisé par un tirage arrière du réseau, qui permet de remonter au marquage source (cause) à partir du marquage correspondant à l'état (défaillant) observé. Le raisonnement de type chaînage arrière est particulièrement intéressant puisqu'il permet a priori de réaliser une réelle recherche de causes. Cependant, les contraintes liées à une application réelle ne sont pas prises en compte et la validité du diagnostic n'est pas évaluée. De plus, l'approche pour le diagnostic est complètement déterministe. Il serait donc intéressant d'apporter de la finesse au diagnostic réalisé afin d'apporter une aide au diagnostic en prenant en compte les imprécisions et les incertitudes inhérentes à la réalité industrielle.

- **Autres formalismes**

Il existe également d'autres formalismes à rattacher aux méthodes à base de modèles comportementaux tels les modèles de la physique qualitative qui permettent d'obtenir un modèle par abstraction du modèle numérique [BAS 1996], ou les approches en logique classique ou linéaire (utilisées également avec les réseaux de Petri [VAL 1994]).

- **Conclusion sur les méthodes à base de modèles comportementaux**

Les automates d'états finis et les réseaux de Petri constituent donc des outils relativement bien adaptés pour construire des mécanismes de détection lorsque le fonctionnement normal du système est décrit par ces formalismes. En revanche, leurs

utilisations en diagnostic sont encore limitées. Pour les automates, les principales difficultés étant liées à la taille importante de l'espace d'état, ceci conduit donc à des problèmes de mémoire et de vitesse d'exécution du diagnostic. Comme il est souligné dans [VAL 1994], les réseaux de Petri constituent un outil puissant de modélisation et peuvent être considérés comme un outil de description de la connaissance nécessaire au diagnostic.

### B) Méthodes de reconnaissance de formes pour la surveillance

Ces méthodes supposent qu'aucun modèle n'est disponible pour décrire les relations de cause à effet. La seule connaissance repose sur l'expertise humaine confortée par un solide retour d'expérience [GIL 1995]. La plupart de ces méthodes sont basées sur l'Intelligence Artificielle avec en particulier des outils tels que les systèmes experts, les outils statistiques (de reconnaissance de formes), le raisonnement à partir de cas (RàPC), les réseaux neuronaux, la logique floue et les réseaux neuro-flous.

#### - *Systemes experts*

Un système expert est un logiciel qui reproduit le comportement d'un expert humain accomplissant une tâche intellectuelle dans un domaine précis. Il est composé de deux parties indépendantes :

- une base de connaissances, elle-même composée d'une base de règles qui modélise la connaissance du domaine considéré et d'une base de faits qui contiennent les informations concernant le cas que l'on est en train de traiter.
- un moteur d'inférences capable de raisonner à partir des informations contenues dans la base de connaissance, de faire des déductions, etc.

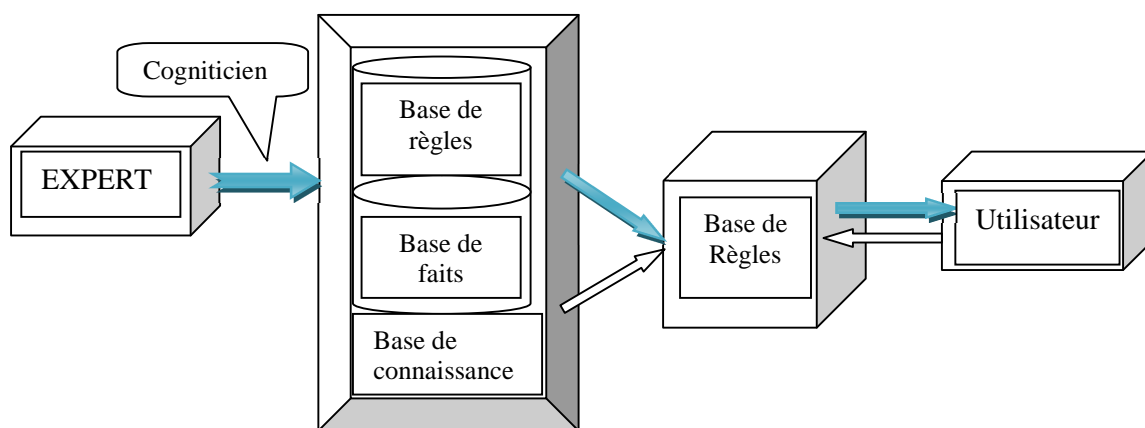


Figure 1.2 : Structure d'un système expert

Le rôle du cogniticien est de soutirer leurs connaissances aux experts du domaine et de traduire ces connaissances dans un formalisme se prêtant à un traitement automatique, c'est-à-dire en règles. Ces deux tâches sont aussi délicates l'une que l'autre. En effet, un expert est la plupart du temps inconscient de la majeure partie de son savoir, et s'il arrive en exprimer une partie, c'est souvent sous une forme difficile à formaliser. Les systèmes experts sont subordonnés à une base de règles logiques représentant l'ensemble des combinaisons importantes d'états, sélectionnées parmi l'explosion combinatoire des états possibles du

système analysé. Les systèmes experts permettent l'exploitation d'une base de données informationnelle considérable. Ils présentent le désavantage d'un comportement rigide, dans un contexte qui s'avère souvent variable et évolutif.

- ***Outils statistiques de reconnaissance de formes***

La première technique présentée est une technique classique de discrimination basée sur les probabilités. Cette technique peut se montrer insuffisante car elle suppose une connaissance a priori de tous les états de fonctionnement et ne prend pas en compte l'évolution du système [DEN 1998].

Le formalisme général de ces techniques de reconnaissance des formes est de devoir décider de l'appartenance de tout vecteur forme d'entrée à un certain nombre de classes (formant un monde fermé – toutes les classes sont supposées connues – ou un monde ouvert – une classe est réservée au rejet en distance, correspondant aux événements inconnus). La classe de rejet en distance est indispensable afin de combler le manque d'information sur le problème. Cette classe représente donc le mélange de toutes les autres classes non identifiées par l'utilisateur. Ce concept présente un intérêt fondamental dans le domaine de la surveillance. En effet, dans ce domaine précis, nous recherchons d'avantage des outils capables de dire « je ne sais pas » que de donner une mauvaise réponse [DAN 2006].

- ***Raisonnement à partir de cas***

Le raisonnement à partir de cas - RàPC (Case Based Reasoning – CBR) est une approche pour résoudre et apprendre des problèmes. Il correspond à la résolution d'un nouveau problème en se rappelant une situation précédente similaire et en réutilisant les informations et les connaissances de cette situation. Il constitue donc une méthodologie pour modéliser le raisonnement et la pensée humaine ou pour développer des systèmes intelligents. Le raisonnement à partir de cas recouvre un ensemble de méthodes de résolution de problèmes à partir d'expériences passées plutôt qu'à partir de connaissances générales comme le raisonnement à partir de règles. Le principe de fonctionnement de la méthode consiste à stocker les expériences précédentes (cas) dans une mémoire afin de résoudre un nouveau problème :

- retrouver l'expérience similaire au nouveau problème dans la mémoire,
- réutiliser cette expérience dans le contexte de la nouvelle situation (complètement, partiellement ou en l'adaptant selon les différences),
- mémoriser la nouvelle expérience dans la mémoire (apprentissage). Le RàPC constitue donc un processus cyclique pour la résolution de problèmes [MAL 1996].

En conclusion, le RàPC constitue une technique pour résoudre des problèmes basés sur l'expérience, et donc relativement bien adapté aux problèmes de diagnostic pour lesquels la notion d'expérience est relativement importante. Pour ce faire, la résolution s'organise en quatre phases pour proposer une solution. Dans le cadre du RàPC, plusieurs techniques sont également disponibles d'une part pour représenter les connaissances, et en particulier pour représenter les cas en fonction du domaine d'utilisation, et d'autre part pour réaliser les différentes phases du cycle. Enfin, le RàPC, présente également les avantages suivants :

- réduit les efforts d'acquisition de connaissances,
- est relativement facile à maintenir,
- l'efficacité de la résolution des problèmes augmente à mesure qu'il est utilisé,
- permet d'utiliser des données existantes comme des bases de données,
- peut s'adapter aux changements de son environnement.

Son utilisation en diagnostic apparaît donc comme relativement aisée, avec comme propriété, une structure de cas adaptée. Cependant, la difficulté tient justement à cette structure de cas et

aux informations qu'elle doit contenir. En effet, l'extraction des connaissances et leurs représentations, sont primordiales dans ce type d'application.

- **Reconnaissance de formes par logique floue**

La logique floue permet de formaliser la représentation et le traitement de connaissances imprécises ou approximatives. Elle offre la possibilité de traiter des systèmes d'une grande complexité dans lesquels sont, par exemple, présents des facteurs humains. Elle intervient dans la manipulation de connaissances imparfaites. Son utilisation dans des domaines tels que l'aide à la décision ou le diagnostic semble donc naturelle dans la mesure où elle fournit un outil puissant pour assister de façon automatique des actions humaines, naturellement empreintes d'imprécisions. La logique floue est ainsi considérée [BOU 1994], [MEU 1995] comme le seul cadre dans lequel on peut traiter des imprévus et des incertitudes, qui autorise également le traitement de certaines incomplétudes, et le seul cadre dans lequel on peut traiter des connaissances numériques et des connaissances exprimées symboliquement par des qualifications du langage naturel.

- **Reconnaissance de formes par réseaux neuronaux**

Les réseaux de neurones sont des outils capables d'effectuer des opérations de perception, classification et prédiction. Leur fonctionnement est basé sur les principes de fonctionnement des neurones biologiques. Leur principal avantage par rapport aux autres outils est leur capacité d'apprentissage et de généralisation de leurs connaissances à des entrées inconnues.

Une des qualités de ce type d'outil est son adéquation pour la mise au point de systèmes de surveillance modernes, capables de s'adapter à un système complexe avec reconfigurations multiples. Les réseaux de neurones peuvent être également implémentés en circuits électroniques, offrant ainsi la possibilité d'un traitement temps réel.

Leur utilisation est principalement guidée par leurs propriétés suivantes:

- capacité d'apprentissage,
- capacité de généralisation,
- parallélisme dans le traitement (rapidité de traitement),
- adaptés aux non-linéarités des systèmes,

Chaque neurone réalise une fonction simple (fonction linéaire, linéaire par morceaux, fonction à seuil, sigmoïde, gaussienne), les propriétés globales de l'outil émergent de sa structure. Toutes les caractéristiques des réseaux de neurones sont exploitées à travers la propriété principale des réseaux de neurone qu'est l'apprentissage. En effet, les mécanismes d'apprentissage sont à l'origine des capacités de résolution de problèmes des réseaux neuronaux. Cet apprentissage permet de configurer les poids synaptiques ainsi que les fonctions d'activation afin d'adopter un comportement désiré. Deux types d'apprentissages sont utilisés : l'apprentissage supervisé et l'apprentissage non supervisé.

**Apprentissage supervisé** : Permet de déterminer les poids synaptiques à partir d'exemples étiquetés auxquels un expert a associé des réponses du réseau. Les paramètres du réseau sont donc modifiés de manière à minimiser l'erreur entre la sortie cible (fournie par l'expert) et la sortie réelle du réseau.

**Apprentissage non supervisé** : Les données fournies en entrée ne contiennent pas d'information sur la sortie désirée. L'apprentissage est réalisé à l'aide de règles qui modifient les paramètres du réseau en fonction des exemples fournis en entrée.

Les architectures les plus utilisées pour ces problèmes de surveillance des systèmes industriels sont le modèle de Hopfield (basé sur la mémoire associative), le réseau de Kohonen (carte topologique auto-adaptative), le Perceptron Multicouches (PMC ou MLP pour Multi layer Perceptron) et le Réseau à Fonctions de base Radiales (RFR ou RBF pour Radial Basis Function) [MOH 2004].

Parmi les réseaux de neurones cités, les réseaux de Kohonen connaissent de nombreuses applications pour la classification de signatures vibratoires de machines. Ils présentent cependant l'inconvénient majeur d'exiger un temps relativement long pendant la phase d'apprentissage [GIL 1995].

Le modèle perceptron multicouche se caractérise donc par une architecture globale (tous les neurones participent à la sortie après la phase d'apprentissage), qui lui confère de mauvaises propriétés en classification et donc le rend moins intéressant pour une utilisation en surveillance. Cependant il existe un nombre non négligeable de travaux en surveillance et diagnostic qui utilisent cette architecture, [MOH 2004].

### **C) Les méthodes à base de modèles explicatifs**

Ces méthodes sont principalement basées sur la représentation des relations entre les différents états de pannes et leurs effets (éventuellement observables). Elles reposent donc sur une analyse profonde du système, de manière à avoir les connaissances suffisantes à l'expression de ses relations de cause à effet. Les modèles ainsi obtenus permettent - pour certains - une approche abductive, qui consiste à remonter aux causes des pannes à partir des observations correspondant aux symptômes. Plusieurs outils de l'Intelligence Artificielle permettent une telle formalisation des connaissances disponibles sur un système. Il s'agit notamment des graphes causaux, des graphes contextuels, techniques auxquelles se joignent également des approches basées sur la logique floue ou les réseaux de Petri. Ces outils, par leurs capacités de modélisation et d'expression, permettent de fournir des modèles explicatifs dans le cadre d'applications au diagnostic.

#### **- Les graphes causaux**

L'exploitation de connaissances causales est assez naturelle pour le diagnostic. En effet, un "dysfonctionnement" peut être assez simplement décrit par les relations associant ses causes à ses manifestations observables. Les graphes causaux constituent un formalisme bien adapté à la représentation de ces liens causaux. Dans une utilisation de diagnostic ils permettent d'exprimer les enchaînements causaux régissant le fonctionnement du système à surveiller en cas de panne. Il s'agit de graphes orientés acycliques. Les nœuds du graphe correspondent aux causes et effets et sont reliés par des arcs orientés.

Cet outil appartient donc aux méthodes à base de modèles explicatifs leur but étant de fournir une explication à des observations d'un fonctionnement anormal d'un système. Ils représentent des connaissances profondes sur le système capables de donner les relations causales entre les différents états de pannes. La construction du modèle repose sur une expertise telle que des catalogues de pannes, des AMDEC ou des arbres de défaillances qui expriment pour une panne l'ensemble des symptômes observables. Leur principale utilisation en diagnostic consiste en un diagnostic abductif. Le graphe causal est utilisé de manière abductive, "à partir des symptômes, on recherche les causes" de manière à relier les symptômes observés à un ensemble de causes possibles [GRO 2000].

- **Les graphes contextuels**

Dans un contexte d'aide à la décision, les outils créés doivent assister un opérateur dans ses activités quotidiennes, sans toutefois prendre de décisions à sa place. Pour ce faire, l'analyse de l'activité des opérateurs permet de mettre en évidence la structure des raisonnements suivis ainsi que l'organisation des sous tâches et des actions. Cette structure peut être modélisée par des graphes. Les graphes ainsi créés donnent les différentes méthodes pour atteindre un but et ces méthodes se distinguent par le contexte dans lequel elles s'appliquent ; on parle alors de graphes contextuels. Il s'agit en effet d'un formalisme bien adapté, car basé sur une expression du raisonnement que suivent les opérateurs en pratique. Les graphes contextuels ont été introduits à partir des arbres de décision pour lesquels on passe d'une représentation basée sur le contexte de l'incident à une représentation basée sur le contexte de résolution de l'incident. Les branches de l'arbre qui conduisent à la même action terminale sont rassemblées dans le graphe et un branchement temporel est introduit dans le graphe pour rendre compte des actions et des décisions qui peuvent être réalisées en parallèle.

Le graphe contextuel est constitué d'actions élémentaires, activités, nœuds contextuels, nœuds de recombinaisons et branchements temporels.

« Un graphe contextuel est un graphe orienté acyclique ayant une unique source et un unique puits. Les sommets du graphe sont des instances des éléments décrits plus hauts. Ils sont reliés par des arcs montrant la précédence temporelle ».

A la lumière de la classification proposée et au vu des précisions concernant l'importance et les spécificités du diagnostic, nous obtenons une vue d'ensemble des méthodes de surveillance, synthétisée par le schéma de la Figure 1.3.

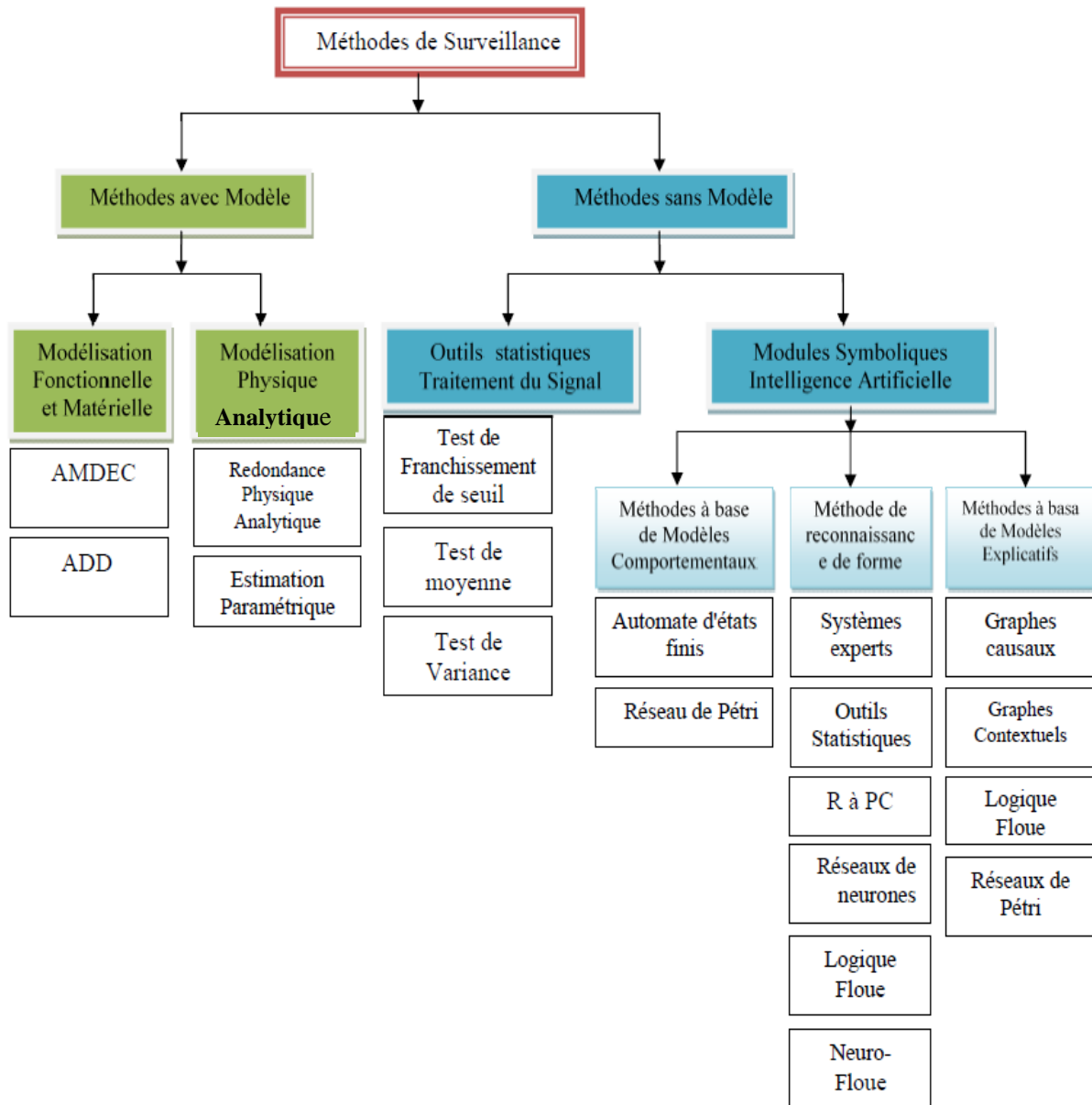


Figure 1.3 : Classification globale des méthodes de surveillance

## CONCLUSION

L'utilisation des méthodes sans modèles pour la surveillance est très nécessaire, car la plupart des systèmes industriels sont des systèmes complexes dont le modèle physique est difficile à trouver ou introuvable. Plusieurs méthodes sont utilisées dans le domaine de surveillance mais jusqu'à maintenant il n'existe pas une méthode universelle qui posséderait tous les avantages et aucun inconvénient :

Pour les méthodes de reconnaissance de formes ; Le diagnostic revient à identifier un mode de fonctionnement du processus qui reflète l'état de panne. En ce sens, la surveillance réalisée ne permet pas d'identifier formellement les causes du dysfonctionnement. Ces outils semblent donc mieux adaptés à la réalisation d'un module de détection « intelligente » dans une architecture complète de surveillance.

Pour les autres méthodes, que ce soit, les méthodes à base de modélisation fonctionnelle et matérielle ou les méthodes à base des modèles explicatifs ; le problème de diagnostic revient essentiellement à un problème de connaissance sur le modèle déterministe entre la cause et l'effet.

**CHAPITRE II : GENERALITES SUR LES**  
**RESEAUX DE NEURONES**

## INTRODUCTION

La physiologie du cerveau montre que celui-ci est constitué de cellules (les neurones) interconnectées. Quelques étapes de cette découverte :

- Van Leuwenhook (1718) : première description fidèle de ce qu'on appellera plus tard les axones,
- Dutrochet (1824) : observation du corps cellulaire des neurones
- Valentin : découverte des dendrites,
- Deiters (1865) : image actuelle de la cellule nerveuse
- Sherrington (1897) : les synapses,
- Les neuro-transmetteurs (première moitié du siècle).

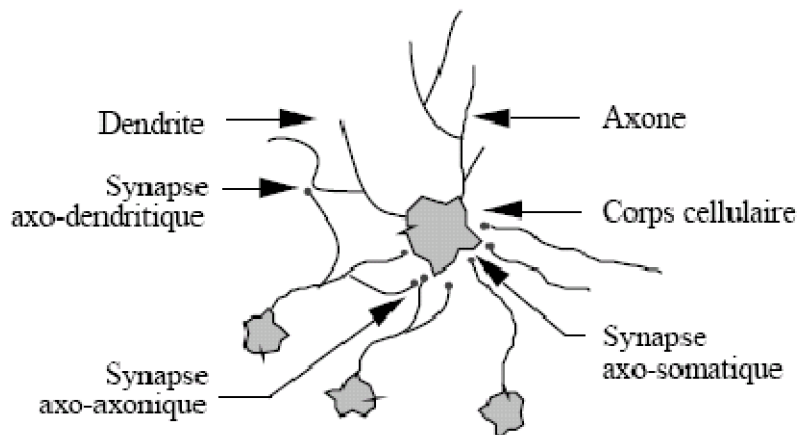


Figure 2.1 : Les neurones schématisés

Les neurones reçoivent les signaux (impulsions électriques) par des extensions très ramifiées de leur corps cellulaire (les dendrites) et envoient l'information par de longs prolongements (les axones). Les impulsions électriques sont régénérées pendant le parcours le long de l'axone. La durée de chaque impulsion est de l'ordre d'1 ms et son amplitude d'environ de 100 mvolts. Les contacts entre deux neurones, de l'axone à une dendrite, se font par l'intermédiaire des synapses. Lorsqu'un potentiel d'action atteint la terminaison d'un axone, des neuromédiateurs sont libérés et se lient à des récepteurs post-synaptiques présents sur les dendrites. L'effet peut être excitateur ou inhibiteur. Chaque neurone intègre en permanence jusqu'à un millier de signaux synaptiques. Ces signaux n'opèrent pas de manière linéaire (effet de seuil).

## II-1 LES RESEAUX DE NEURONES ARTIFICIELS

### II-1.1 Définition

Les réseaux de neurones artificiels sont des réseaux fortement connectés de processeurs élémentaires fonctionnant en parallèle. Chaque processeur élémentaire calcule une sortie unique sur la base des informations qu'il reçoit. Toute structure hiérarchique de réseaux est évidemment un réseau [CLA 1992].

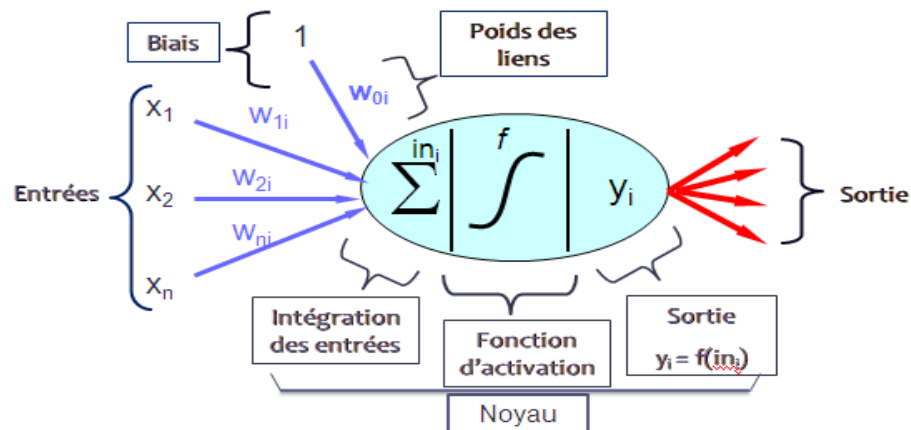


Figure 2.2 : Le neurone artificiel générique

## II-1.2 Historique

- 1890 : W. James, célèbre psychologue américain introduit le concept de mémoire associative, et propose ce qui deviendra une loi de fonctionnement pour l'apprentissage sur les réseaux de neurones connue plus tard sous le nom de loi de Hebb.

- 1943 : J. Mc Culloch et W. Pitts laissent leurs noms à une modélisation du neurone biologique (un neurone au comportement binaire). Ceux sont les premiers à montrer que des réseaux de neurones formels simples peuvent réaliser des fonctions logiques, arithmétiques et symboliques complexes (tout au moins au niveau théorique).

- 1949 : D. Hebb, physiologiste américain explique le conditionnement chez l'animal par les propriétés des neurones eux-mêmes. Ainsi, un conditionnement de type pavlovien tel que, nourrir tous les jours à la même heure un chien, entraîne chez cet animal la sécrétion de salive à cette heure précise même en l'absence de nourriture. La loi de modification des propriétés des connexions entre neurones qu'il propose explique en partie ce type de résultats expérimentaux.

- 1957 : F. Rosenblatt développe le modèle du Perceptron. Il construit le premier neuro-ordinateur basé sur ce modèle et l'applique au domaine de la reconnaissance de formes. Notons qu'à cet époque les moyens à sa disposition sont limités et c'est une prouesse technologique que de réussir à faire fonctionner correctement cette machine plus de quelques minutes.

- 1960 : B. Widrow, un automaticien, développe le modèle Adaline (Adaptative Linear Element). Dans sa structure, le modèle ressemble au Perceptron, cependant la loi d'apprentissage est différente. Celle-ci est à l'origine de l'algorithme de rétropropagation de gradient très utilisé aujourd'hui avec les Perceptrons multicouches. Les réseaux de type Adaline restent utilisés de nos jours pour certaines applications particulières. B. Widrow a créé dès cette époque une des premières firmes proposant neuro-ordinateurs et neuro-composants, la "Memistor Corporation".

- 1969 : M. Minsky et S. Papert publient un ouvrage qui met en exergue les limitations théoriques du perceptron. Limitations alors connues, notamment concernant l'impossibilité de

traiter par ce modèle des problèmes non linéaires. Ils étendent implicitement ces limitations à tous modèles de réseaux de neurones artificiels. Leur objectif est atteint, il y a abandon financier des recherches dans le domaine (surtout aux U.S.A.), les chercheurs se tournent principalement vers l'IA et les systèmes à bases de règles.

- 1967-1982 : Toutes les recherches ne sont, bien sûr, pas interrompues. Elles se poursuivent, mais déguisées, sous le couvert de divers domaines comme : le traitement adaptatif du signal, la reconnaissance de formes, la modélisation en neurobiologie, etc. De grands noms travaillent durant cette période tels : S. Grossberg, T. Kohonen, ....

- 1982 : J. J. Hopfield est un physicien reconnu à qui l'on doit le renouveau d'intérêt pour les réseaux de neurones artificiels. A cela plusieurs raisons :

Au travers d'un article court, clair et bien écrit, il présente une théorie du fonctionnement et des possibilités des réseaux de neurones. Il faut remarquer la présentation anticonformiste de son article. Alors que les auteurs s'acharnent jusqu'alors à proposer une structure et une loi d'apprentissage, puis à étudier les propriétés émergentes ; J. J. Hopfield fixe préalablement le comportement à atteindre pour son modèle et construit à partir de là, la structure et la loi d'apprentissage correspondant au résultat escompté. Ce modèle est aujourd'hui encore très utilisé pour des problèmes d'optimisation. D'autre part, entre les mains de ce physicien distingué, la théorie des réseaux de neurones devient respectable. Elle n'est plus l'apanage d'un certain nombre de psychologues et neurobiologistes hors du coup.

- 1983 : La Machine de Boltzmann est le premier modèle connu apte à traiter de manière satisfaisante les limitations recensées dans le cas du perceptron. Mais l'utilisation pratique s'avère difficile, la convergence de l'algorithme étant extrêmement longue (les temps de calcul sont considérables).

- 1985 : La rétropropagation de gradient apparaît. C'est un algorithme d'apprentissage adapté aux réseaux de neurones multicouches (aussi appelés Perceptrons multicouches). Sa découverte réalisée par trois groupes de chercheurs indépendants indique que "la chose était dans l'air". Dès cette découverte, nous avons la possibilité de réaliser une fonction non linéaire d'entrée/sortie sur un réseau en décomposant cette fonction en une suite d'étapes linéairement séparables. De nos jours, les réseaux multicouches et la rétropropagation de gradient reste le modèle le plus étudié et le plus productif au niveau des applications.

La situation actuelle (1992) : En France, elle est à l'image du congrès Neuro-Nîmes qui a pour thème les réseaux neuromimétiques et leurs applications. Créé en 1988, le chiffre de ses participants croît chaque année et reflète bien l'intérêt que le monde scientifique et industriel (50% des participants) porte au connexionnisme [CLA 1992].

### II-1.3 ARCHITECTURES DES RESEAUX

Les RNA peuvent être vus comme des graphes orientés dans lesquels les neurones formels sont les sommets, et les arcs (pondérés) sont les connexions entre sorties de neurones et entrées de neurones. En fonction du type de connexions (architectures), les RNA sont regroupés en deux catégories (voir Figure 2.3).

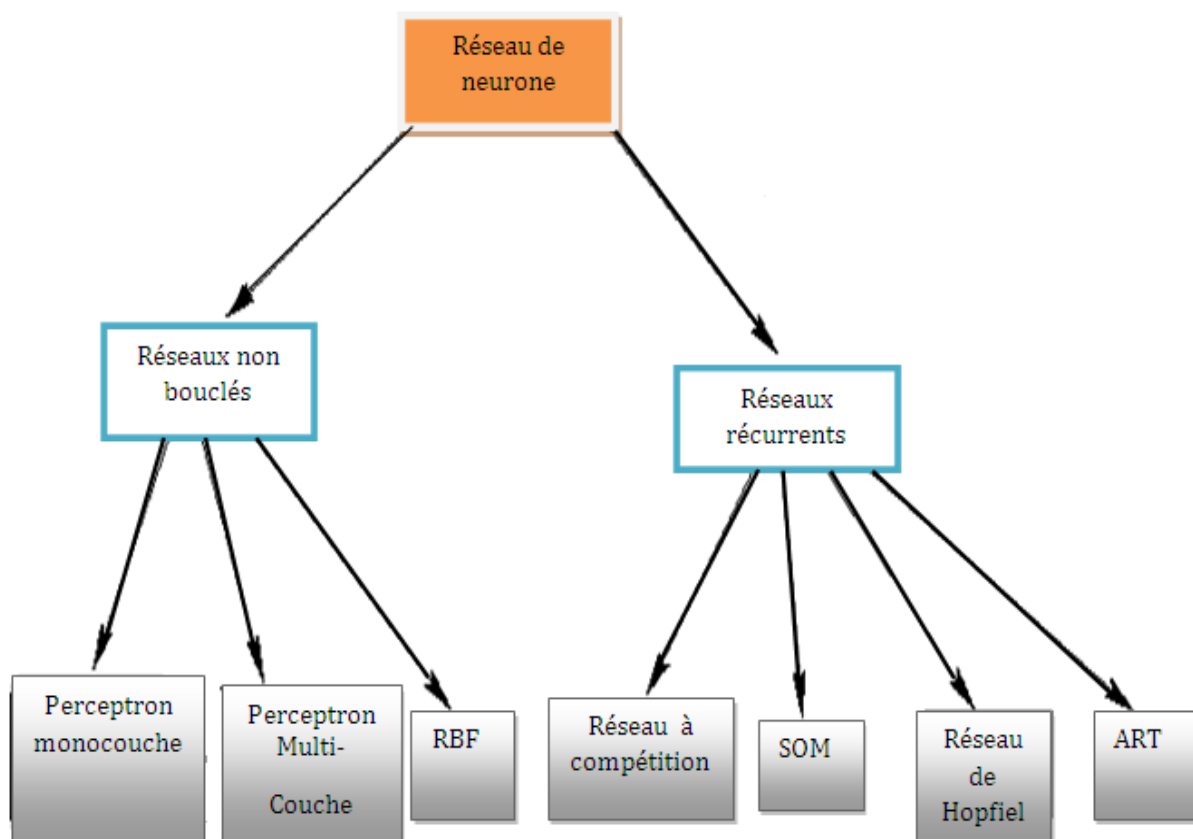


Figure 2.3 : Les différents types des réseaux non-bouclés et récurrents

**II-1.3.1 Réseaux Non Bouclés (Feed-Forward) :** sont des graphes acycliques ; on distingue trois types essentiels :

- *Perceptron simple*

Le *perceptron* (un seul neurone) est un modèle de réseau de neurones avec algorithme d'apprentissage créé par Frank Rosenblatt en 1958. L'apprentissage ayant été réalisé, les poids sont fixes. Le neurone de la figure 2.4 réalise une simple somme pondérée de ses entrées, compare une valeur de seuil, et fourni une réponse binaire en sortie. Par exemple, on peut interpréter sa décision comme classe 1 si la valeur de  $x$  est  $+1$  et classe 2 si la valeur de  $x$  est  $-1$ .

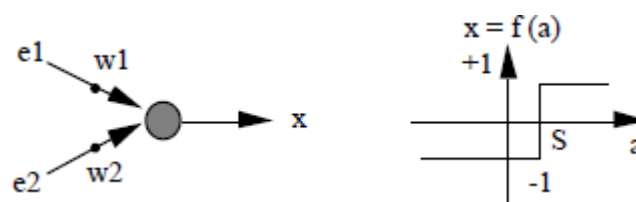


Figure 2.4 : Le perceptron simple

- **Perceptron multicouche (MLP)**

Les neurones sont arrangés par couche, il n'y a pas de connexion entre neurone d'une même couche et la connexion ne se fait qu'avec les neurones des couches en aval. Habituellement, chaque neurone d'une couche est connecté à tous les neurones de la couche suivante et celle-ci seulement, nous permet d'introduire la notion de sens de parcours de l'information (de l'activation) au sein d'un réseau et de définir les concepts de neurone d'entrée et de neurone de sortie. Par extension on appelle couche d'entrée l'ensemble des neurones d'entrées et la couche de sortie l'ensemble des neurones de sorties. Les couches intermédiaires n'ayant aucun contact avec l'extérieur sont appelées couches cachées, tel que représenté sur la Figure 2.5 [NED 2008].

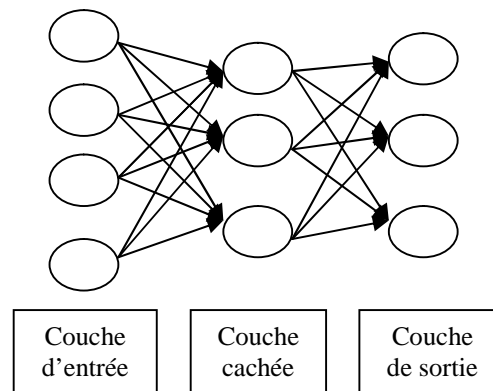


Figure 2.5 : Réseau multicouche

- **Réseau à fonction radiale (RBF)**

Les réseaux à fonction radiale (RBF) qui possèdent deux couches forment une classe particulière de réseaux multi-couches. Chaque cellule de la couche cachée utilise une fonction noyau (kernel function) telle que la Gaussienne en tant que fonction d'activation. Cette fonction est centrée au point spécifié par le vecteur de poids associé à la cellule. La position et la "largeur" de ces courbes sont apprises à partir des patrons. Il y a, en général, beaucoup moins de fonctions noyaux dans un réseau RBF que de patrons d'entrée. Chaque cellule de sortie implémente une combinaison linéaire de ces fonctions, l'idée étant d'approximer une fonction par un ensemble de fonctions. De ce point de vue, les cellules cachées fournissent un ensemble de fonctions qui forment une base représentant les patrons d'entrées dans l'espace "couvert" par les cellules cachées.

**II-1.3.2 Réseaux Récurrents (Feedback) :** sont des graphes avec circuits du fait de la présence d'arcs de rétro-action ; on distingue les types suivants :

**Réseau à compétitions**

Dans le réseau à compétitions ; chaque cellule de sortie est connectée aux cellules d'entrée, ainsi qu'à toutes les cellules voisines de la couche de sortie (connexion inhibitrice) et à elle-même (connexion excitatrice). Le résultat de la compétition est de choisir la cellule  $i_0$  ayant la plus grande (ou la plus petite) entrée, c'est-à-dire que l'on a  $w_{i_0} x > w_i x, \forall i$ .

L'exemple le plus connu d'apprentissage par compétition est le LVQ (Learning Vector Quantization Networks) pour la compression de données, très largement utilisé dans le cadre du traitement de la parole, du stockage d'images, de la transmission et de la modélisation. Il s'agit de représenter un ensemble ou une distribution de vecteurs à l'aide d'un nombre restreint de vecteurs prototypes ou d'un livre de codes. Une fois que le livre de codes a été construit et agréé par le transmetteur et le récepteur, il ne reste alors qu'à transmettre ou stocker l'index du vecteur prototype correspondant au vecteur de données. Étant donné un vecteur de données, son vecteur prototype peut être trouvé en cherchant le vecteur prototype le plus voisin dans le livre des codes.

#### - **CARTES AUTO-ORGANISATRICES DE KOHONEN (SOM)**

Les cartes Auto-Organisatrices (SOM) préservent la topologie, ce qui permet de capturer un aspect important des "cartes de traits" dans le cortex des animaux. Pour les fonctions préservant la topologie, des patrons d'entrée voisins doivent activer des cellules de sortie voisines dans la carte.

#### - **RÉSEAUX DE HOPFIELD**

Hopfield a utilisé une fonction d'énergie associée au réseau comme outil pour définir des réseaux récurrents et pour comprendre leurs dynamiques. La formulation utilisée par Hopfield rend explicite le principe de stockage de l'information en tant qu'attracteurs et popularisa l'utilisation de réseaux récurrents en tant que mémoire associative ainsi que comme outils pour résoudre des problèmes d'optimisation combinatoire (en faisant le parallèle entre fonction d'énergie et fonction objective). Un réseau de Hopfield de  $n$  cellules existe en deux versions : binaire et continue. Soit  $v_i$  l'état de la sortie de la cellule  $i$ . Pour les réseaux binaires,  $v_i$  peut prendre valeur dans  $\{-1,+1\}$ , alors que dans le cas continu,  $v_i$  est à valeur dans  $[0, 1]$ . Soit  $w_{ij}$  le poids de la connexion entre les cellules  $i$  et  $j$ . Dans un réseau de Hopfield, les connexions sont symétriques ( $w_{ij} = w_{ji}$ ) et  $w_{ii} = 0$ .

#### - **LES MODÈLES DE RÉSONNANCE ADAPTATIVE (ART)**

Le dilemme "stabilité-plasticité" est une notion fondamentale dans le cadre des réseaux à compétition. Comment apprenons-nous de nouvelles choses (plasticité) tout en gardant une stabilité garante d'une connaissance ni supprimée ni "abimée". Les modèles développés par Carpenter et Grossberg (ART-1, ART-2, ARTMap) dans le cadre de la théorie de résonance adaptative (ART) essaient pour résoudre ce dilemme. Le réseau possède un réservoir de cellules de sortie qui ne sont utilisées que si nécessaire. Une cellule sera dite *recrutée* (committed) ou *libre* (uncommitted) si elle est utilisée ou non utilisée. L'algorithme d'apprentissage met à jour les vecteurs prototypes stockés uniquement s'ils sont suffisamment proches du patron fourni en entrée au réseau. Un patron et un prototype "résonnent" lorsqu'ils sont voisins.

### II-1.4 Apprentissage

L'apprentissage est une phase du développement d'un réseau de neurone durant laquelle le comportement du réseau est modifié jusqu'à l'obtention du comportement désiré, l'apprentissage neuronal fait appel à des exemples de comportement. Dans le cas des réseaux de neurones artificiels on ajoute souvent à la description un modèle de l'algorithme d'apprentissage. Le modèle sans apprentissage présente en effet peut d'intérêt dans la majorité des algorithmes actuels. Les variables modifiées pendant l'apprentissage sont les poids de connexion.

L'apprentissage est la modification des poids de réseaux dans le but d'accorder la réponse de ce dernier, il est souvent impossible de décider à priori des valeurs des poids des connexions d'un réseau pour une application donnée. A l'issue de l'apprentissage, des poids sont fixés qui est alors la phase d'utilisation de certain modèles des réseaux, dénommés « l'apprentissage permanent ». Dans ce cas il est vrai que l'apprentissage ne s'arrête jamais. Cependant on peut toujours distinguer une phase d'utilisation (en fait de remise à jour du comportement) et une phase d'utilisation, cette technique permet au réseau de conserver un comportement adapté malgré les fluctuations dans les données. Au niveau des algorithmes d'apprentissage, il a été défini deux grandes classes selon que l'apprentissage est dit supervisé ou non supervisé. Cette distinction repose sur la forme des exemples d'apprentissage.

#### - Apprentissage supervisé

Dans le cas d'apprentissage supervisé, les exemples sont des couples (entrée, sortie associée) alors que l'on ne dispose que des valeurs d'entrées. Voir Figure 2.6

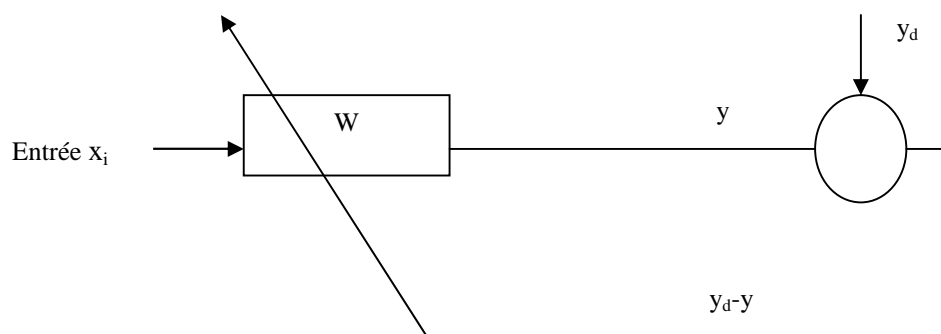


Figure 2.6 : Apprentissage supervisé

#### - Apprentissage non supervisé

L'apprentissage non supervisé nécessite la présence des entrées seulement sans intervention d'un superviseur, cet entraînement se fait sur la base d'information locale existante au niveau des neurones et nous fait découvrir des propriétés collectives qui existent entre les données sur la base desquelles le réseau doit s'organiser. Voir Figure 2.7

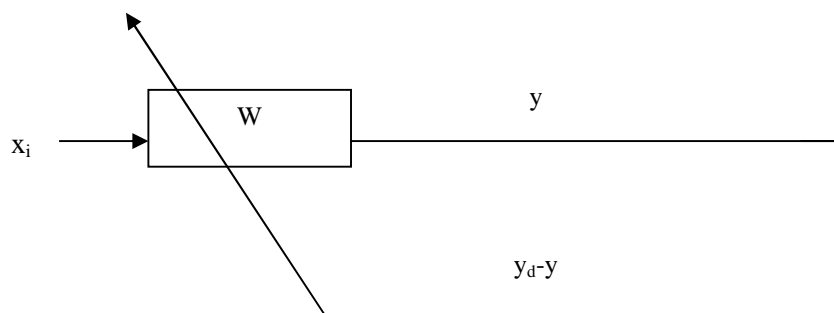


Figure 2.7: Apprentissage non supervisé

### II-1.4.1 Choix du taux d'apprentissage

Plusieurs solutions ont été proposées pour le choix de taux d'apprentissage ou (le pas d'apprentissage), d'après Freeman on choisit un pas d'apprentissage entre (-0.5 ; 0.25) mais le problème qui se pose à chaque fois c'est le rapport vitesse / complexité qui n'est pas toujours convaincant. L'une des solutions est celle de Brunel qui consiste à adapter le pas afin d'avoir une décroissance régulière de l'erreur quadratique moyenne.

### II-1.4.2 Valeurs initiales des poids

Les valeurs des poids doivent être différentes de zéro .Si c'était le cas les inconvénients d'adaptation des poids sont nuls. En pratique on choisit aléatoirement les valeurs initiales des poids.

### II-1.4.3 Test d'arrêt

En pratique il faut calculer un indice de convergence car on ne peut pas assurer la convergence de l'algorithme pour un nombre d'itérations alors qu'un autre cherche à arrêter l'algorithme si l'erreur est minimale.

### II-1.4.4 Choix de réseau

De nombreux travaux théoriques abordent l'approximation des fonctions continues par les réseaux de neurones ce qui a amené à l'application du choix de ne pas pouvoir savoir la taille exacte du réseau pour un problème donné.

## II-1.5 Apprentissage du Perceptron simple

Nous basons ici sur les travaux de Rosenblatt (1958) qui a développé la première version d'un modèle neuronal très connu de nos jours, à savoir le Perceptron. C'est un réseau à deux couches (une couche d'entrée et une couche de sortie) de type feed forward (propagation avant). Les neurones de la couche d'entrée ont pour rôle de fournir au réseau les données externes. Chaque neurone de la couche de sortie effectue une somme pondérée de ses entrées ou ceci est donné par l'équation (II.1) où  $O_i$  représente la sortie. (Figure 2.8)

$$O_i = f(a_i) = f\left(\sum_{k=1}^{k=N} w_{ik} \xi_k\right) \quad (\text{II-1})$$

Où  $w_{ik}$  est le poids de la connexion qui relie l'unité  $i$ ,  $a_i$  est l'activation de l'unité  $i$ ,  $f$  est la fonction d'activation des unités. Cette fonction d'activation est du type fonction à seuil avec l'expression suivante :

$$f(x) = \begin{cases} +1 & \text{si } x \geq 0 \\ -1 & \text{si } x < 0 \end{cases}$$

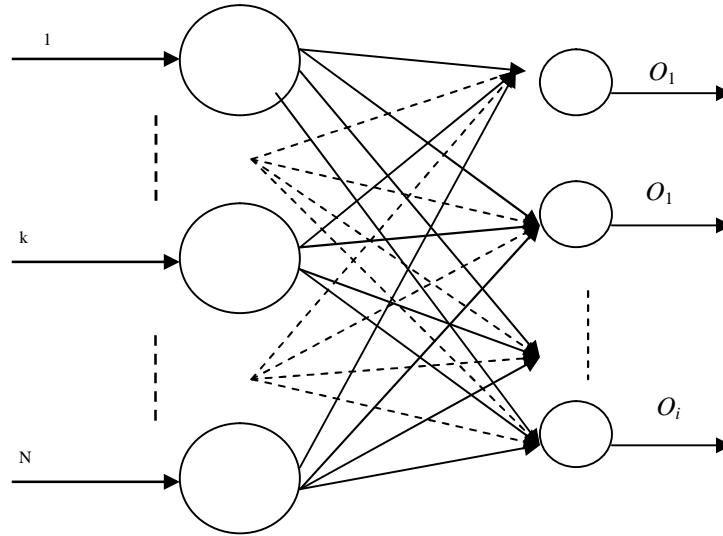


Figure 2.8: Perceptron simple, modèle de Rosenblatt

Rosenblatt s'est inspiré de la règle de Hebb pour la modification des poids. Son principe est de rajouter, dans le cas où la sortie obtenue  $O_i^\xi$  du réseau est différente de la sortie désirée  $\zeta_i^\xi$ , une quantité  $\Delta w_{ik}$  aux poids de chaque connexion. Dans le cas contraire, les connexions demeurent inchangées. On peut exprimer ce principe par :

$$w_{ik}(t+1) = w_{ik}(t) + \Delta w_{ik} \quad (\text{II-2})$$

Où  $\Delta w_{ik}$  est la quantité ajoutée au poids  $w_{ik}(t)$ . Pour chaque exemple  $\wp$  de l'ensemble des exemples d'apprentissage, on peut ainsi écrire :

$$\Delta w_{ik} = \eta (\zeta_i^{\wp} - o_i^{\wp}) \zeta_k^{\wp} \quad (\text{II-3})$$

Le paramètre  $\eta$  est appelé taux d'apprentissage. Il détermine la dynamique suivant laquelle les modifications vont avoir lieu.

Cette procédure d'apprentissage pour le Perceptron simple peut converger vers un état des poids des connexions donnant de bons résultats, à la seule condition que le problème soit linéairement séparable. Malheureusement, un grand nombre de problèmes rencontrés en pratique, ne sont pas linéairement séparables.

### II-1.6 Apprentissage des réseaux multicouche

Les pondérations des différents réseaux utilisées sont ajustées par l'apprentissage supervisé en exploitant l'idée de minimisation itérative d'un critère de l'erreur en sortie qui a pour principe de minimiser une « fonction de coût » de l'erreur calculée entre la sortie du réseau et la sortie souhaitée pour une entrée donnée. Donc le rôle de ce principe est de trouver le plus rapidement possible le minimum d'une fonction d'énergie dont la forme dépend uniquement de l'architecture du réseau ; voir figure 2.9.

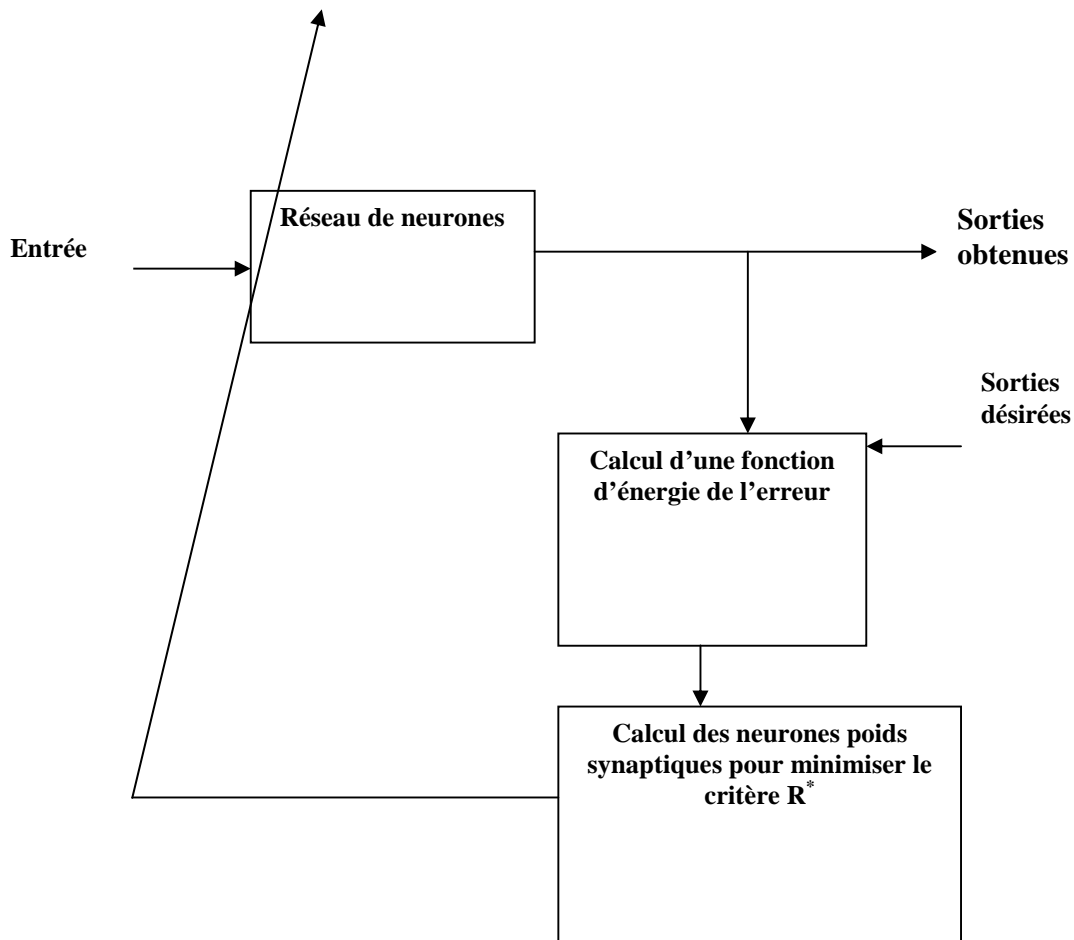


Figure 2.9 : Minimisation itérative d'un critère de l'erreur en sortie

Toutefois, les principes sous-jacents aux règles d'apprentissage les plus connus ne sont rien d'autre que des principes d'optimisation. Le plus utilisé est le principe de descente de gradient, il a été conçu pour dévier les problèmes de non séparabilité linéaire.

### II-1.6.1 Algorithme de descente de gradient

L'algorithme utilise les conventions de notation représentées sur sa description édifiée dans la figure 2.10.

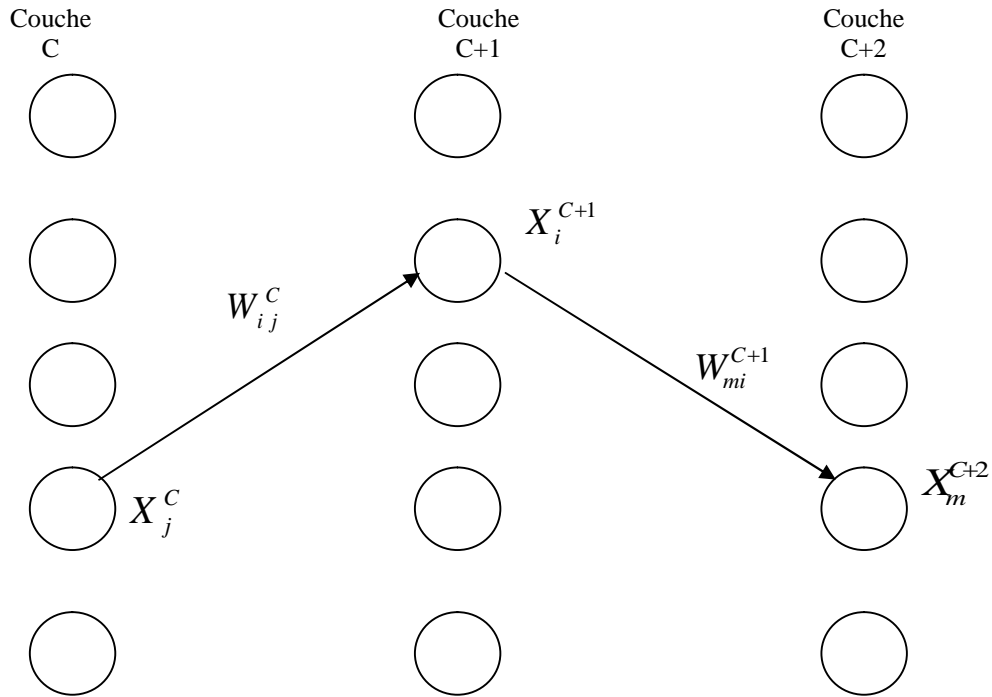


Figure 2.10 : Algorithme de retropropagation de gradient

- soit  $n$  le nombre total des exemples  $d_i$ ,  $i = 1, \dots, n$  de la base d'apprentissage et  $N_n$  le nombre de cellule de la dernière couche.
- La couche de rang  $C$  contient  $N_C$  cellules repérées  $X_i^C$   $i = 1, \dots, N_C$ .
- Pour un réseau contenant  $N$  couches,  $C = 1$  et  $C = N$  correspondent à la couche d'entrée et à la couche de sortie.

Soit  $X_i^{C+1}$  la sortie de la cellule numéro  $i$  de la couche  $C+1$ , son contenu vaut :

$$X_i^{C+1} = f\left(\sum_{j=0}^{N_C} W_{ij}^C X_j^c\right) = f(A_i^{C+1}) \tag{II-4}$$

$f$ : fonction sigmoïde

$$A_i^{C+1} = W_i^{tr} X^C \tag{II-5}$$

- En désignant par  $X_{ik}^N$  la sortie de la cellule de rang  $i$  de la dernière couche associée à l'exemple  $X_k$  présenté sur la couche d'entrée, par  $d_i^k$  la sortie souhaitée, ou  $k$  varie de 1 à  $n$

et  $N_N$  est le nombre de cellules de la dernière couche, la fonction quadratique de l'erreur a pour expression :

$$R = \sum_{k=1}^n \sum_{i=1}^{N_N} \left\| X_{ik}^N - d_i^k \right\|^2 \quad (\text{II-6})$$

Cette fonction de coût fait appel de façon implicite à tous les éléments du réseau et à tous les poids synaptiques. Un élément de la dernière cellule est fourni par :

$$X_i^N = f \left( \sum_{j=0}^{N_N-1} W_{ij}^{N-1} X_j^{N-1} \right) \quad (\text{II-7})$$

Une règle d'apprentissage possible consisterait à minimiser la fonction R avec la technique des moindres carrés, la règle de retropropagation du gradient utilise la règle de Windrow-Hoff en minimisant la fonction de coût évaluée lors de la présentation de l'exemple  $d_i^k$

$$R^* = \sum_{i=0}^{N_N} \left( X_{ik}^N - d_i^k \right)^2 \quad (\text{II-8})$$

La loi d'adaptation du vecteur poids W s'écrit :

$$W_k = W_{k-1} - \alpha_k \nabla_k (W) \quad (\text{II-9})$$

$\alpha_k$  : taux d'apprentissage

La loi de pondération des poids sera noté par :

$$W_{ij}^k = W_{ij}^{k-1} - \alpha_k \frac{\partial R^*}{\partial W_{ij}} \quad (\text{II-10})$$

En considérant la dernière couche, on obtient pour la sortie désirée  $d_i^k$  :

$$\frac{\partial R^*}{\partial W_{ij}^{N-1}} = 2(X_{ik}^N - d_i^k) \frac{\partial X_{ik}^N}{\partial W_{ij}^{N-1}} = 2(X_{ik}^N - d_i^k) \frac{\partial X_{ik}^N}{\partial A_{ik}^{N-1}} \frac{\partial A_{ik}^{N-1}}{\partial W_{ij}^{N-1}} \quad (\text{II-11})$$

Avec :

$$A_{ik}^N = \sum_{j=0}^{j=N-1} W_{ij}^{N-1} X_{jk}^{N-1} \quad \text{et} \quad \frac{\partial A_{ik}^{N-1}}{\partial W_{ij}^{N-1}} = X_{jk}^{N-1} \quad (\text{II-12})$$

Compte tenu de (II-4) :

$$\frac{\partial X_{ik}^N}{\partial A_{ik}^N} = f'(A_{ik}^N) \quad (\text{II-13})$$

D'où :

$$\frac{\partial R^*}{\partial W_{ij}^{N-1}} = 2(X_{ik}^N - d_i^k) f'(A_{ik}^N) X_{jk}^{N-1} \quad (\text{II-14})$$

Le calcul des poids de la dernière couche dépend des éléments des couches précédentes.

- pour la couche C on peut écrire de façon similaire :

$$\frac{\partial R^*}{\partial W_{ij}^C} = \frac{\partial R^*}{\partial A_{ik}^{C+1}} \cdot \frac{\partial A_{ik}^{C+1}}{\partial W_{ij}^C} \quad (\text{II-15})$$

Avec :

$$\frac{\partial A_{ik}^{C+1}}{\partial W_{ij}^C} = X_{ik}^C \quad (\text{II-16})$$

$$\frac{\partial R^*}{\partial A_{ik}^{C+1}} = \sum_m \frac{\partial R^*}{\partial A_{mk}^{C+2}} \frac{\partial A_{mk}^{C+2}}{\partial A_{ik}^{C+1}} \quad (\text{II-17})$$

- l'indice m correspond au nombre de termes qui contribue à la fonction de coût lorsque l'on passe de la couche c+1 à la couche c+2.

$$\frac{\partial A_{mk}^{C+2}}{\partial A_{ik}^{C+1}} = \frac{\partial \left[ \sum_{i=1}^{i=n_{c+2}} W_{mi}^{C+1} f(A_{ik}^{C+1}) \right]}{\partial A_{ik}^{C+1}} = W_{mi}^{C+1} f'(A_{im}^{C+1}) \quad (\text{II-18})$$

$n_{c+2}$  représente le nombre de cellules de la couche (c+2) augmenté d'une unité et finalement on obtient :

$$\frac{\partial R}{\partial w_{ij}^c} = \sum_{m=1}^{m=n_{c+2}} \frac{\partial R}{\partial A_{mk}^{c+2}} w_{mi}^{c+1} f'(A_{ik}^{c+1}) X_{jk}^c \quad (\text{II-19})$$

L'algorithme de rétropropagation de gradient se résume comme suite :

a) Pour la dernière couche, les poids sont adaptés suivant la loi :

$$w_{ij,k}^N = w_{ij,k-1}^{N-1} - \alpha_k (X_{ik}^N - d_i^k) f'(A_{ik}^N) X_{jk}^{N-1} \quad (\text{II-20})$$

b) Pour les couches intermédiaires on a :

$$w_{ij,k}^c = w_{ij,k-1}^{c-1} - \alpha_k \sum_{m=1}^{m=n_{c+2}} \frac{\partial R}{\partial A_{mk}^{c+2}} w_{mi,k}^{c+1} f'(A_{ik}^{c+1}) X_{jk}^c \quad (\text{II-21})$$

### II-1.6.2 Algorithme de Levenberg-Marquardt

L'algorithme de Levenberg-Marquardt (LM) est une technique itérative qui localise le minimum d'une fonction multi-variable qui est exprimée comme somme des carrés des fonctions à valeurs réelles non linéaires. C'est devenu une technique standard pour les problèmes des moindres carrés non linéaires, largement adoptée dans un large éventail des disciplines. Le LM peut être considéré comme combinaison de la descente la plus raide et de la méthode de Gauss-Newton. Quand la solution courante est loin de la correcte, l'algorithme se comporte comme une méthode de la descente la plus raide : ralentir, mais garanti à converger. Quand la solution courante est proche de la solution correcte, ça devient une méthode de Gauss-Newton [MAN 2005].

- **Rappel mathématique sur l'algorithme de Newton et de LM**

Tandis que la rétropropagation avec la technique de descente de gradient est un algorithme de la descente la plus raide, l'algorithme de Levenberg-Marquardt est une approximation à la méthode de Newton. Si une fonction  $V(x)$  doit être réduite au minimum en ce qui concerne le vecteur  $X$  de paramètre, alors la méthode de Newton serait : [OZG 2004]

$$x = -[ \text{ }^2V(x) ]^{-1} \text{ }^1V(x) \quad (\text{II-22})$$

Dont,  $\text{ }^2V(x)$  est la matrice de Hess et  $\text{ }^1V(x)$  est le gradient tel que :

$$V(x) = \sum_{i=1}^N e_i^2(x) \quad (\text{II-23})$$

On peut mettre :

$$V(x) = J^T(x) e(x) \tag{II-24}$$

$${}^2V(x) = J^T(x) J(x) + S(x) \tag{II-25}$$

Dont,  $J(x)$  est la matrice Jacobi et :

$$S(x) = \sum_{i=1}^N e_i \cdot {}^2e_i(x) \tag{II-26}$$

Pour la méthode de Gauss-Newton on pose  $S(x) = 0$ , alors l'équation (II-22) devient :

$$x = -[J^T(x) J(x)]^{-1} J^T(x) e(x) \tag{II-27}$$

Donc l'équation de LM extraite de l'équation de Gauss-Newton est la suivante :

$$x = -[J^T(x) J(x) + \mu I]^{-1} J^T(x) e(x) \tag{II-28}$$

• **L'utilisation de l'algorithme LM dans les réseaux MLP**

La fonction objectif lors de l'apprentissage des réseaux de neurones, c'est la fonction erreur suivante : [DEE 2006]

$$e = \sum_{k=1}^p \frac{1}{2} (t_k - y_k)^2 \tag{II-29}$$

Dont,  $y_k$  c'est la sortie actuelle du réseau de  $k$  (ieme) composante,  $t_k$  c'est la sortie désirée,  $p$  est le nombre total des éléments d'apprentissage,  $z$  représente les poids et les biais du réseau.les différentes étapes pour l'apprentissage du réseau en utilisant l'algorithme LM sont :

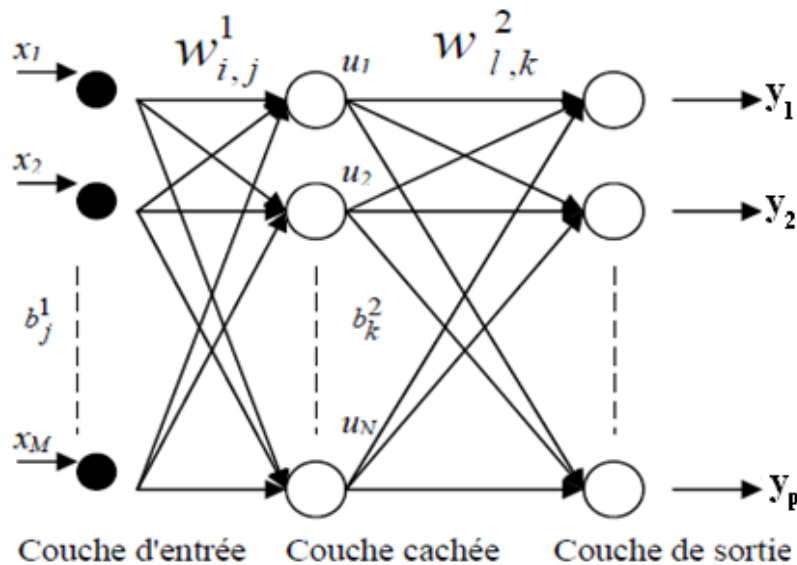


Figure 2.11 : Algorithme LM pour un réseau de trois couches

**Etape 1 :** On présente toutes les données du réseau et on calcule toutes les sorties et les erreurs du réseau en utilisant l'équation (II-29).

- La sortie de chaque neurone d'entrée prend tout simplement les valeurs du vecteur d'entrées (*input patterns*).

$x_i$  : sortie de la couche d'entrée (couche 0),  $i = 1, \dots, M$  ( $M$  : nombre de composantes du vecteur d'entrée)

- L'entrée de la couche cachée est donnée par :

$$A_j^1 = \sum_{i=1}^N w_{i,j}^1 x_i + b_j^1 \quad (\text{II-30})$$

$N$  : nombre de neurones de la couche cachée,  $w$  : matrice poids,  $b$  : vecteur biais

- La sortie de cette couche cachée est donnée par :

$$u_j = f_1(A_j^1) \quad (\text{II-31})$$

En désignant par  $u_j$  la sortie de la cellule de rang  $j$  de la couche cachée,  $j$  varie de 1 à  $N$

$$f_1(A_j^1) = \frac{2}{1 + e^{-2A_j^1}} - 1 \quad (\text{II-32})$$

$f_1$  : C'est la fonction d'activation tangente hyperbolique

- L'entrée de la couche de sortie (dernière couche) est exprimée par :

$$A_k^2 = \sum_{j=1}^P w_{j,k}^2 x_j + b_k^2 \quad (\text{II-33})$$

$P$  : nombre de neurones de la couche de sortie.

- La sortie de cette dernière couche est donnée par :

$$y_k = f_2(A_k^2) \quad (\text{II-34})$$

$$f_2(A_k^2) = A_k^2 \quad (\text{II-35})$$

$f_2$  : C'est la fonction d'activation de type linéaire.

**Etape 2 :** On calcule la matrice de Jacobi  $J(w)$  et  $J(b)$ , dont  $w$  et  $b$  représentent respectivement, les poids et les biais du réseau.

**Etape 3 :** On résoud la nouvelle équation de poids par l'algorithme LM en obtenant  $w$ .

Le vecteur  $W$  de mise à jour de poids est calculé comme suit :

$$.w_{i,k}^2 = [J^T(w_{i,k}^2) J(w_{i,k}^2) + \mu I]^{-1} J^T(w_{i,k}^2) E \quad (\text{II-36})$$

$$.w_{i,j}^! = [J^T(w_{i,j}^!) J(w_{i,j}^!) + \mu I]^{-1} J^T(w_{i,j}^!) E \quad (\text{II-37})$$

$$.b_k^2 = [J^T(b_k^2) J(b_k^2) + \mu I]^{-1} J^T(b_k^2) E \quad (\text{II-38})$$

$$.b_j^! = [J^T(b_j^!) J(b_j^!) + \mu I]^{-1} J^T(b_j^!) E \quad (\text{II-39})$$

Là où  $E$  est un vecteur de longueur  $P$  calculé comme suit :

$$E = [t_1 - y_1 \quad t_2 - y_2 \quad \dots \quad t_p - y_p]^T \quad (\text{II-40})$$

Ici  $J^T(w) J(w)$  est référé comme matrice Hessienne.  $I$  est la matrice d'identité,  $\mu$  est le taux d'apprentissage. Pour  $\mu = 0$  l'algorithme devient la méthode de Gauss-Newton. Pour  $\mu$  très grand l'algorithme de LM devient la méthode de descent de gradient ou l'algorithme de rétropropagation d'erreur. Le paramètre  $\mu$  est automatiquement ajusté à chaque itération afin de fixer la convergence. L'algorithme de LM exige le calcul de la matrice de Jacobi  $J(w)$  et  $J(b)$  à chaque étape d'itération et la matrice transposée  $J^T(w)$ .  $J(w)$  matrice carrée [DEE 2006].

**Etape 4 :** On recalcule l'erreur, en utilisant la nouvelle matrice de poids  $w + w$ . Si cette nouvelle erreur est plus petite que celle calculée dans l'étape 1, alors on doit réduire le taux d'apprentissage  $\mu$  par  $\mu-$ , laisser  $w + w$ , et retourner à étape 1. Si l'erreur n'est pas réduite, alors on augmente  $\mu$  à  $\mu+$  et retournant à l'étape 3. Le  $\mu-$  et les  $\mu+$  sont définis par l'utilisateur.

**Etape 5 :** On assume que l'algorithme converge quand la norme du gradient est inférieure à une certaine valeur prédéterminée, ou quand l'erreur a été réduite à un certain but d'erreur.

## II-2 LES DIFFERENTS TYPES DE RESEAUX ET LEUR UTILISATION

Le choix d'une telle architecture des réseaux de neurones artificiels, dépend généralement au domaine d'utilisation celle-ci (classification, prédiction, catégorisation...). La table II-1 fournit, de manière non exhaustive, différents algorithmes d'apprentissage et la topologie du réseau associé. Les différents paradigmes d'apprentissage (supervisé et non supervisé) emploient des règles d'apprentissage basées sur la correction d'erreurs, la règle de Hebb, l'apprentissage par compétition. Les règles d'apprentissage du type correction d'erreurs peuvent être utilisées pour les réseaux non-bouclés, tandis que la règle de Hebb a été utilisée sur toutes les architectures. Cependant, chaque algorithme d'apprentissage a été défini pour entraîner une architecture particulière : donc, lorsque l'on parle d'un algorithme d'apprentissage, une architecture spécifique est induite dans le discours. Chaque algorithme peut effectuer parfaitement un nombre particulier de tâches. Dans la dernière colonne de la table II-1, on trouvera les tâches spécifiques traitées par les différents algorithmes.

La rangée, représentée en vert dans la table II-1, est le type de réseau qui sera utilisé dans notre étude concernant la surveillance des systèmes industriels ; Voir chapitre III et chapitre IV.

Paradigme	Règle d'apprentissage	Architecture	Algorithme	Tâches
Supervisé	Correction d'erreur	Perceptron Simple ou Multi-Couches	Perceptron, Rétro-Propagation, Adaline, Madaline, LM,	Classification, Approximation de fonctions, Prédiction, Contrôle, Surveillance.
	Bolzmman	Récurrente	Apprentissage de Boltzmann	Classification
	Hebb	Multi-Couches non bouclés	Analyse de discriminants linéaires	Analyse de données, Classification
	par compétition	à compétition	LVQ	Catégorisation au sein d'une classe, Compression de données
		ART	ARTMap	Classification, Catégorisation au sein d'une classe
Non supervisé	Correction d'erreur	Multi-couches non bouclé	Projection de Sammon	Analyse de données
	Hebb	Non bouclé ou à compétition	analyse en composantes principales	Analyse de données, compression de données
	par compétition	à compétition	VQ	Catégorisation, Compression de données
		Cartes de Kohonen	SOM	Catégorisation, Analyse de données
		ART	ART-1, ART-2	Catégorisation
Hybride	Correction d'erreur et par compétition	RBF	RBF	Classification, Approximation de fonctions, Prédiction, Contrôle

Table II-1: Différentes architectures de RN

## CONCLUSION

Il existe une grande variété de projets dans lesquels les réseaux de neurones artificiels ont eu du succès. Les RNA se démarquent de techniques plus conventionnelles notamment grâce à leur capacité d'apprentissage et de généralisation, à leur immunité au bruit ainsi que par leur architecture hautement parallèle, fortement appréciée pour le traitement de l'information en temps réel.

Le choix de la meilleure architecture du réseau dépend de plusieurs paramètres :

- Le nombre de neurones de différentes couches,
- Le nombre de couches,
- Les valeurs des poids et du taux d'apprentissage,
- L'algorithme d'apprentissage.

Il y a plusieurs types d'algorithmes utilisés pour l'apprentissage des réseaux neuronaux. Généralement pour les problèmes d'approximation de fonction des réseaux qui contiennent jusqu'à quelques centaines de poids, l'algorithme de Levenberg-Marquardt aura la convergence la plus rapide. Dans beaucoup de cas, l'algorithme LM peut obtenir des erreurs quadratiques moyennes inférieures que l'un des autres algorithmes examinés.

**CHAPITRE III : ETUDE DE CAS - DETECTION ET  
LOCALISATION DES DEFAILLANCES D'UNE  
TURBINE A VAPEUR**

## INTRODUCTION

La turbine à vapeur est un des matériels stratégiques des installations de production d'énergie électrique. Son indisponibilité entraîne la perte de la production d'énergie, contrairement à d'autres matériels (pompes alimentaires, pompes de circulation assistée, ventilateurs...) dont la redondance permet de poursuivre l'exploitation des installations soit à pleine charge, soit éventuellement à charge partielle. [HEN 2002]

Il est essentiel d'effectuer un suivi de l'évolution de l'ensemble de grandeurs physiques mesurées en exploitation qui sont, pour la plupart, enregistrés en salle de commande des tranches de production. Cette surveillance est réalisée lors du fonctionnement des installations aux conditions nominales mais elle s'avère également utile lors des phases transitoires d'exploitation (démarrage, arrêt, prise de charge...). [HEN 2002]

C'est pour cette raison qu'on a essayé de proposer une méthodologie simple pour le développement d'un système de surveillance de tous grandeurs physiques essentiels ( pression de vapeur, vibrations, température d'huile ...) dans tous les points nécessaires de la turbine à vapeur de la centrale thermique de Cap-Djinet, basée sur la méthode de réseau de neurones artificiels de type perceptron multi couches (MLP), en utilisant l'algorithme de Levenberg-Marquardt ( qui a été présenté dans le chapitre II).

La centrale thermique de production d'électricité de Cap-Dinet (SONELGAZ) est située au bord de la mer, à l'est d'Alger, près de la ville de Cap-Djinet, dans la wilaya de BOUMERDES (ALGERIE). Elle a été construite par un consortium Austro-Allemand : SIEMENS-KWU-SGP.

Après l'apprentissage, les zones de bon et de mauvais fonctionnement, seront illustrées. Nous codons les zones de bon fonctionnement à « 1 », et les zones défectueuses à « -1 ». Enfin, nous évaluerons le taux de reconnaissance des réseaux. La programmation a été intégralement faite par MATLAB 7.5.

Les grandeurs physiques nécessaires à surveiller sont :

- La température de la vapeur,
- La pression de la vapeur,
- Les vibrations relatives,
- La vitesse de rotation de la turbine,
- La température d'huile à la sortie des paliers,
- Le débit d'huile dans les paliers.

### III-1 Description de la turbine

#### III-1.1 Rôle

La turbine transforme l'énergie thermique contenue dans la vapeur provenant de la chaudière en un mouvement de rotation de l'arbre. Le travail mécanique obtenu sert à entraîner l'alternateur. La turbine étudiée dans ce mémoire est constituée de trois (03) corps : HP, MP et BP, elle contient aussi six (06) soutirages de vapeur : alimentant 03 réchauffeurs BP, la bêche alimentaire et 02 réchauffeurs HP [DOC 1984].

### III-1.2 Caractéristiques

- ☞ Longueur : 16,125 m
- ☞ Largeur : 13 m
- ☞ Poids : 500 103 Kg
- ☞ Puissance : 176 MW
- ☞ Pression : 138,2 bars
- ☞ Température vapeur : 535 °C
- ☞ Vitesse de rotation : 3000 T/mn [DOC 1984]



Figure 3.1 : Photo réelle de la turbine étudiée accouplée avec l'alternateur et l'excitatrice

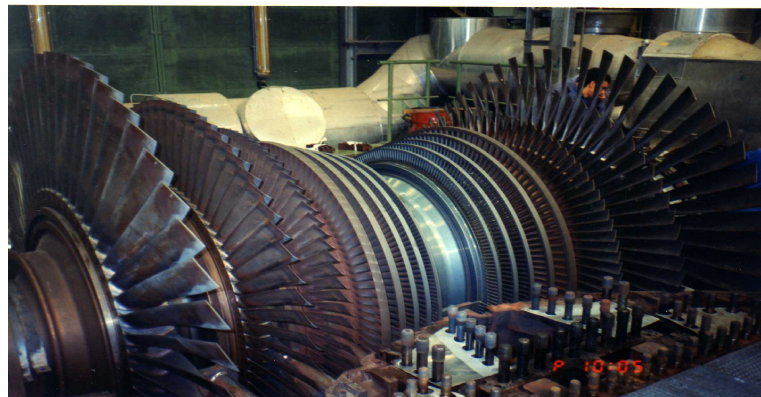


Figure 3.2 : Photo réelle du corps BP de la turbine (double flux)

### III-1.3 Fonctionnement de la turbine

L'eau chimiquement pure circule sous forte pression dans les tubes du générateur de vapeur et se transforme en vapeur à une température de 535 °C.

Cette vapeur surchauffée et à haute pression se détend dans la turbine haute pression. Puis elle retourne dans un réchauffeur, pour aller en suite dans les turbines moyenne et basse pression où elle se détend.

À la sortie des turbines, la vapeur à très basse pression se condense pour se retrouver à l'état liquide dans les condenseurs. Cette eau injectée ensuite dans le réservoir du générateur de vapeur, et le cycle recommence toujours avec la même eau [DOC 1984].

### III-1.4 Positionnement des capteurs de surveillance

Concernant notre étude, nous avons choisi six types essentiels de capteurs de surveillance. Les six types sont placés de la façon suivante :

- Un capteur de vitesse est en avant de l'arbre de la turbine (1);
- Quatre capteurs de vibrations pour les quatre paliers de la turbine (2) (3) (4) (5);
- Quatre capteurs de pression : deux à l'entrée et à la sortie du corps HP (7) (6), un à l'entrée du corps MP (8) et un à la sortie du corps BP (9) ;
- Quatre capteurs de débit d'huile dans les quatre paliers (10) (11) (12) (13) ;
- Quatre capteurs de température d'huile à la sortie de chaque palier (14) (15) (16) (17) ;
- Quatre capteurs de température : deux à l'entrée et à la sortie du corps HP (19) (18), un à l'entrée du corps MP (20) et un à la sortie du corps BP (21) ; Voir (Figure 3.3).

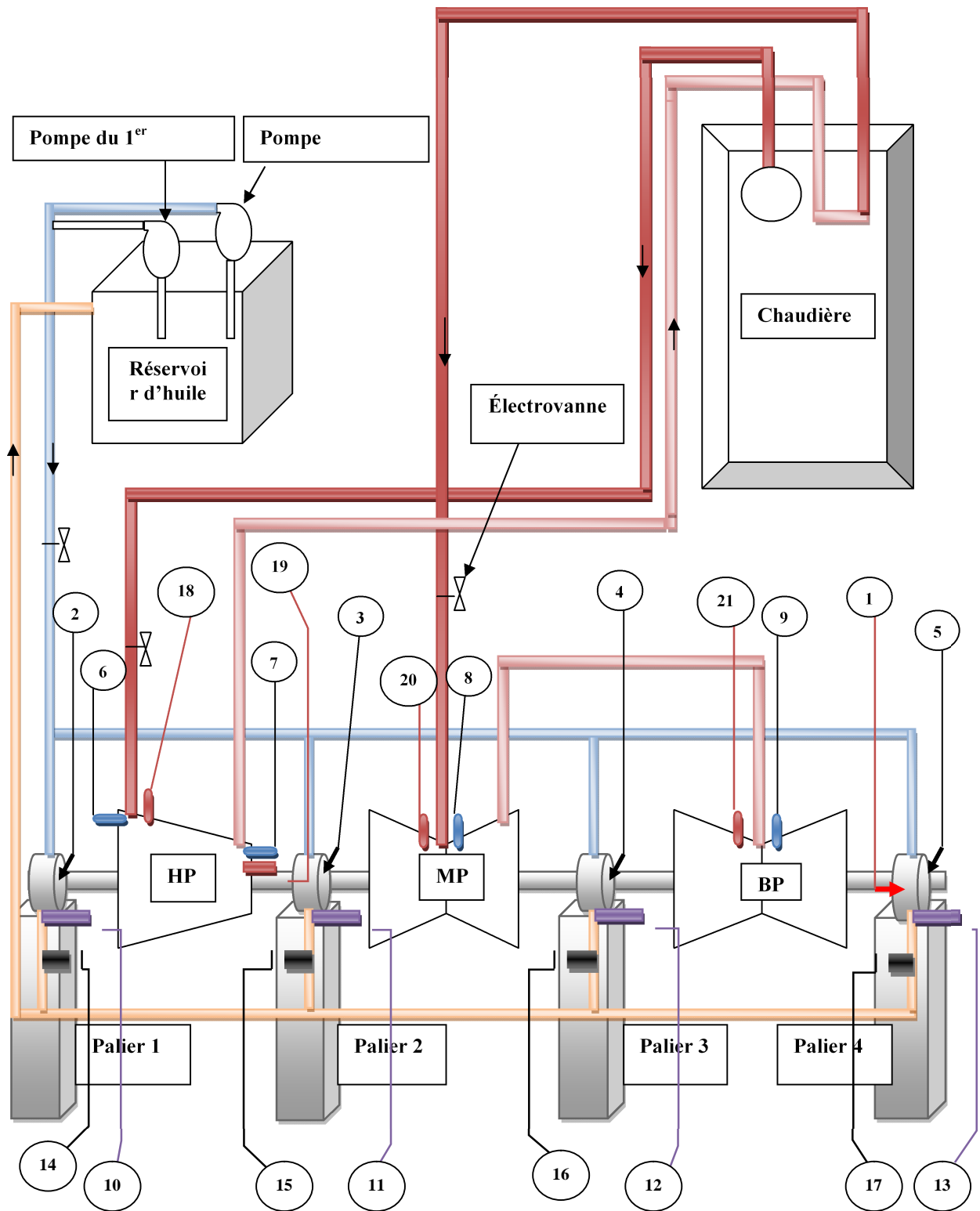


Figure 3.3 : Positionnement des capteurs de surveillance dans la turbine

**III-2 ORGANIGRAMME POUR LE CALCUL DE L'ARCHITECTURE OPTIMALE**

W, b : sont la matrice de poids et le vecteur de biais (l'initialisation des valeurs de poids et de biais se fait aléatoirement, mais toujours entre [-1, +1]),

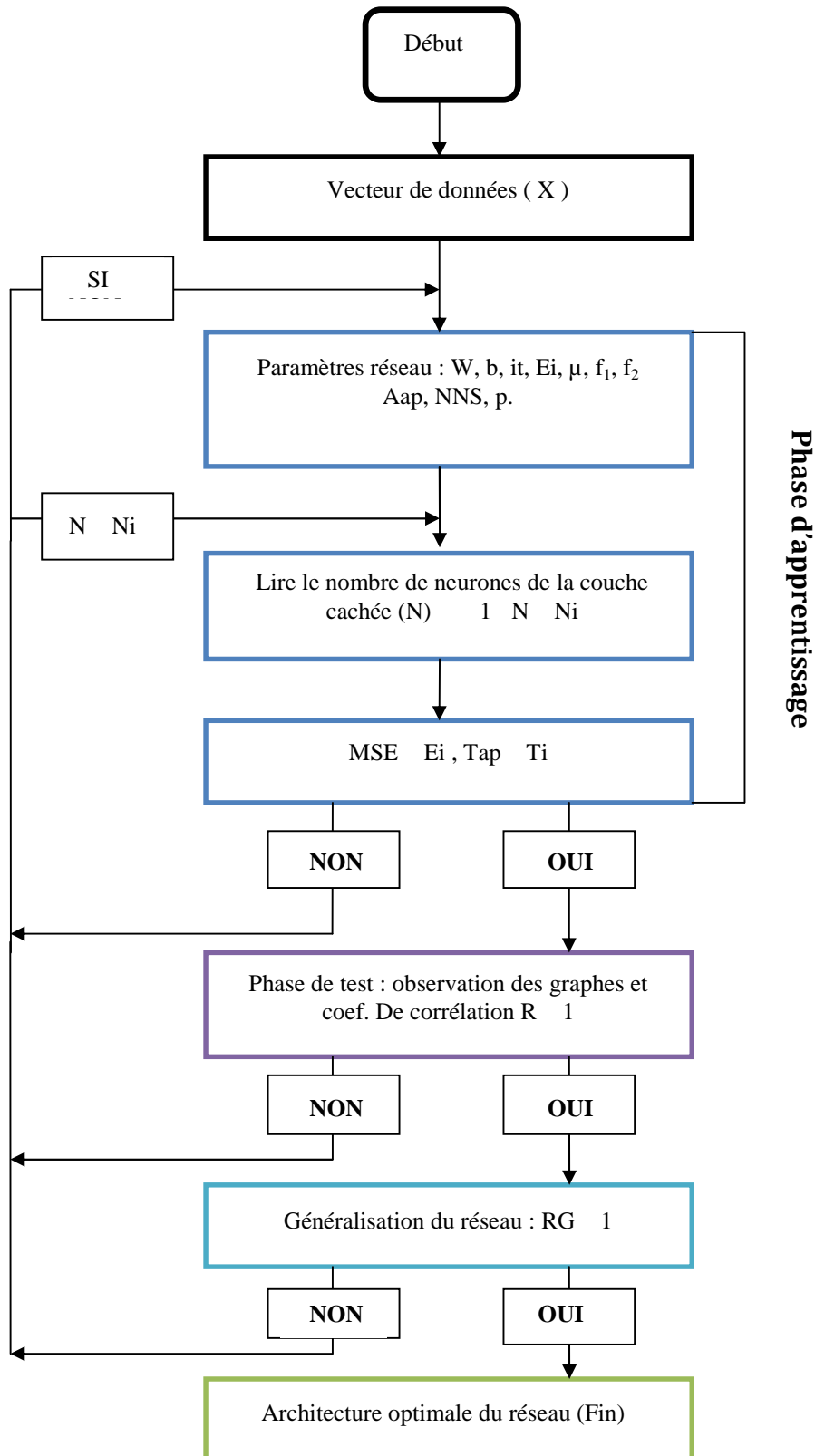


Figure 3.4 : Organigramme générale pour le choix d'une architecture optimale.

it : nombre des itérations,  $E_i$  : erreur imposée,  $\mu$  : taux d'apprentissage (entre 0 et 1)  $f_1$  : fonction d'activation de la couche cachée,  $f_2$  : fonction d'activation de la couche de sortie, Aap : algorithme d'apprentissage, NNS : nombre de neurones dans la couche de sortie, p : le pas,  $N_i$  : nombre imposé, R : coefficient de corrélation, RG : coefficient du réseau généralisé, MSE : erreur quadratique moyenne, Tap : temps d'apprentissage et  $T_i$  : temps imposé pour l'apprentissage.

Cet organigramme est valable pour tous les exemples qu'on va traiter dans ce chapitre du mémoire. Pour obtenir l'architecture optimale nous devons varier premièrement le nombre de neurones de la couche cachée (N) de 1 jusqu'à un nombre imposé ( $N_i$ ) de telle sorte à satisfaire la condition  $MSE \leq E_i$  (erreur moyenne quadratique une erreur donnée), si tout les N qu'on a donné n'ont pas permis d'avoir cette condition, dans ce cas nous devons revenir aux paramètres réseau ; on fait des changements aléatoires pour ces paramètres jusqu'à obtenir la convergence de l'erreur. Après nous allons passer à la deuxième étape c'est l'étape de test, dans ce cas on observe bien la réaction du réseau avec les valeurs de test données. Aussi le coefficient de corrélation doit être proche de « 1 », sinon nous devons refaire toute l'étape d'apprentissage en augmentant par exemple l'erreur imposée ( $E_i$ ). La dernière phase qui est la phase de généralisation, dans ce cas aussi nous devons avoir la coïncidence entre le graphe des sorties désirées et celui des sorties réseau (le coefficient de corrélation du réseau généralisé RG doit être proche de « 1 ») ; Voir figure 3.4.

### III-3 SURVEILLANCE DE LA VITESSE DE ROTATION DE LA TURBINE

#### III-3.1 Données du constructeur

- Vitesse nominale de rotation de la turbine  $V_{nom} = 3000$  tr/min (50 Hz).
- Vitesse minimale de rotation de la turbine  $V_{min} = 2250$  tr/min (47.5 Hz).
- Vitesse maximale de rotation de la turbine  $V_{max} = 3090$  tr/min (51.5 Hz).

#### III-3.2 Méthodologie du travail

On a considéré dans cette étude que la vitesse acceptable pour le bon fonctionnement appartient à l'intervalle [2250 tr/min - 3090 tr/min] et extérieur de cet intervalle c'est la vitesse non admissible. Les zones de bon fonctionnement sont représentées par « 1 » et les zones de mauvais fonctionnement sont représentées par « -1 ».

L'erreur limite donnée au réseau est  $E_i = 10^{-3}$ , le nombre d'itérations imposé  $it = 1000$  et le pas de calcul c'est  $p = 0.001$ , le taux d'apprentissage  $\mu = 0.5$ .

Pour obtenir l'architecture optimale nous devons suivre les étapes représentées dans l'organigramme de la figure 3.3. Le nombre de neurone maximum donné pour la couche cachée est  $N_i = 20$ , donc l'architecture optimale trouvée est : « 5001-06-01 » qui représente la plus petite erreur **0.000487846**, voir tableau 3.1.

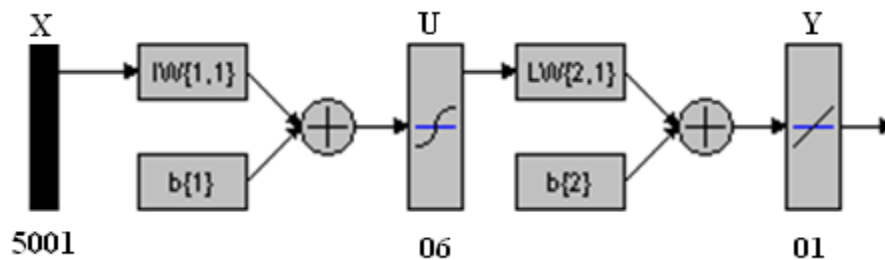


Figure 3.5 : Architecture optimale trouvée

Architecture	Tap(s)	Itérations	R (corrélation)	RG	MSE
5001-01-01	23.12	27	0.2584	0.2555	0.170543
5001-02-01	350.21	1000	0.9348	0.9059	0.0230613
5001-03-01	31.08	46	0.9983	0.8749	0.000612669
5001-04-01	416.78	123	0.9411	0.9108	0.00208892
5001-05-01	497.88	145	0.9498	0.9155	0.00178749
<b>5001-06-01</b>	<b>50.15</b>	<b>95</b>	<b>0.9987</b>	<b>0.9904</b>	<b>0.000487846</b>
5001-07-01	231.82	436	0.9973	0.8748	0.000979888
5001-08-01	221.01	337	0.9973	0.8746	0.000998612
5001-09-01	589.32	1000	0.9941	0.8714	0.00213773
5001-10-01	264.60	470	0.9973	0.8743	0.000995086
5001-11-01	413.49	591	0.9973	0.8751	0.00099572
5001-12-01	33.39	30	0.9980	0.8755	0.000729747
5001-13-01	23.03	19	0.9980	0.8746	0.000759345
5001-14-01	269.70	362	0.9973	0.8732	0.00099916
5001-15-01	807.86	1000	0.9913	0.8757	0.00316105
5001-16-01	522.02	484	0.9974	0.8742	0.00096618
5001-17-01	631.70	740	0.9973	0.8723	0.000995914
5001-18-01	512.71	615	0.9974	0.8726	0.000952272
5001-19-01	846.61	1000	0.9748	0.9132	0.00909053
5001-20-01	514.85	588	0.9973	0.8730	0.00099733

Tableau 3.1 : Variation de l'erreur quadratique moyenne en fonction de neurones de la couche cachée pour la vitesse de rotation de la turbine.

Tap : temps d'apprentissage, R : coefficient de corrélation, RG : coefficient de corrélation du réseau généralisé, MSE : erreur moyenne quadratique.

La figure 3.5 montre l'architecture optimale du réseau étudié, le nombre de neurones de la couche cachée  $N = 6$  tel que le vecteur d'entrée  $X = [x_1, x_2, \dots, x_M]^T$ , le vecteur de la couche cachée  $U = [u_1, u_2, \dots, u_N]$  et le vecteur de sortie  $y = [y_1]$ ,  $M = 5001$  dans notre cas. Les lois générales pour le calcul des sorties des neurones de la couche cachée et de la couche de sortie sont respectivement :

$$u_j = f_1 \left( \sum_{i=1}^M w_{i,j}^1 x_i + b_j^1 \right) \quad (II-1)$$

$$y_k = f_2 \left( \sum_{l=1}^N w_{l,k}^2 u_l + b_k^2 \right) \quad (II-2)$$

M : nombre d'éléments dans le vecteur d'entrée.  
 N : nombre de neurones dans la couche cachée.

$f_1, f_2$  sont respectivement les fonctions d'activation tangente hyperbolique (II-3) et linéaire (II-4) :

$$f_1(v) = \frac{2}{1 + e^{-2v}} - 1 \quad (\text{II-3})$$

$$f_2(v) = v \quad (\text{II-4})$$

**Remarque :** les mêmes fonctions d'activation sont utilisées pour le calcul du reste des exemples.

L'apprentissage de réseau MLP implique de trouver des valeurs des poids de raccordement  $w^1_{ij}$  et  $w^2_{lk}$ , qui réduisent au minimum la fonction de l'erreur moyenne MSE entre les résultats réels produit (calculés) et les valeurs à atteindre (désirées) correspondantes dans l'ensemble d'apprentissage. L'apprentissage du réseau par l'algorithme de Levenberg-Marquardt est fait directement à l'aide de la boîte à outils de réseau de neurones de Matlab.

$$\text{MSE} = \frac{1}{n} \sum_{k=1}^n (d - y_k)^2 \quad (\text{II-5})$$

$d$  : sortie désirée,  $y_k$  : sortie du réseau (couche de sortie),  $n$  : nombre d'exemples.

### III-3.3 Explication des résultats

#### III-3.3.1 Apprentissage du réseau

Le graphe d'apprentissage se converge vers l'erreur fixée d'ordre  $10^{-3}$  après 95 itérations dans une durée de 50.15 s, voir figure 3.6.

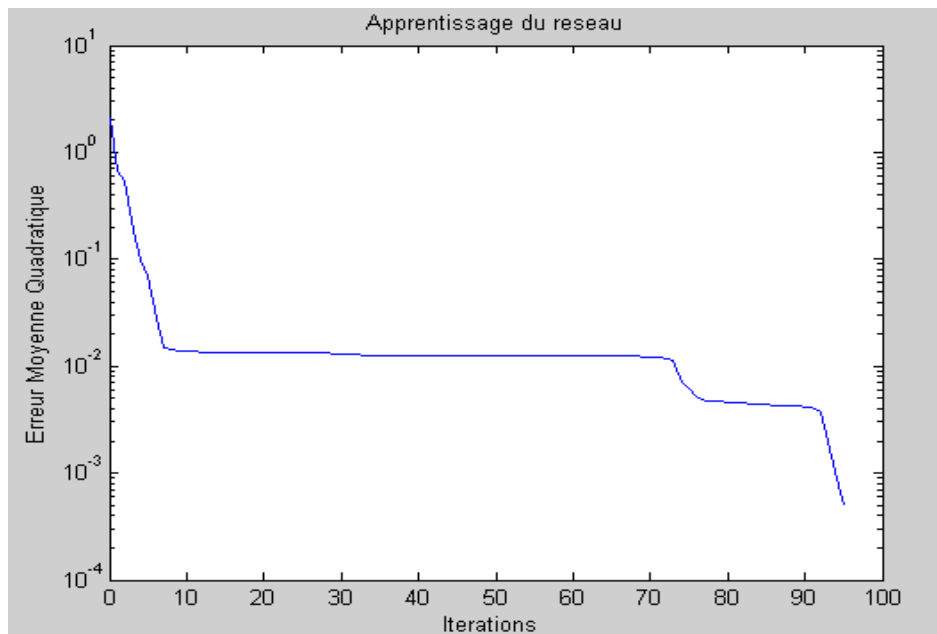


Figure 3.6 : Apprentissage du réseau de neurone « 5001-06-01 ».

### III-3.3.2 Simulation des sorties de réseau

On voit que le graphe vert du réseau est bien coïncidé avec le graphe bleu de la fonction désirée ce qui explique la bonne simulation de l'architecture trouvée, voir figure 3.7.

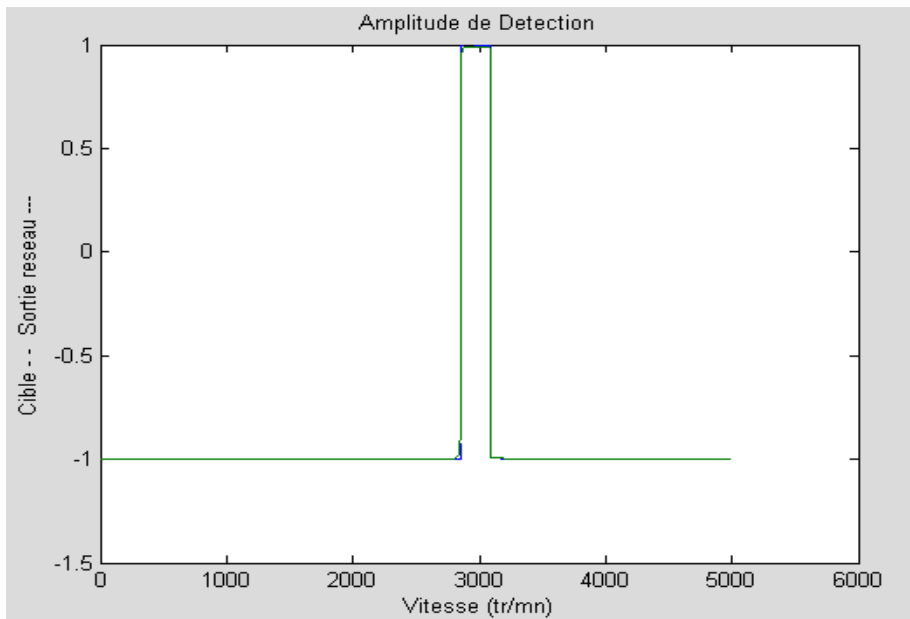


Figure 3.7 : Simulation des sorties du réseau

### III-3.3.3 Test du réseau

Nous avons testé le réseau par une valeur de vitesse acceptable égale à 3000 tr/min, nous remarquons que le réseau n'a détecté aucune alarme, voir la figure 3.8. Par contre dans un deuxième test dont nous avons essayé par une valeur de vitesse qui égale à 4000 tr/min on voit une réaction du réseau qui a déclaré une alarme représentée par un cercle rouge au niveau de « -1 », voir figure 3.9.

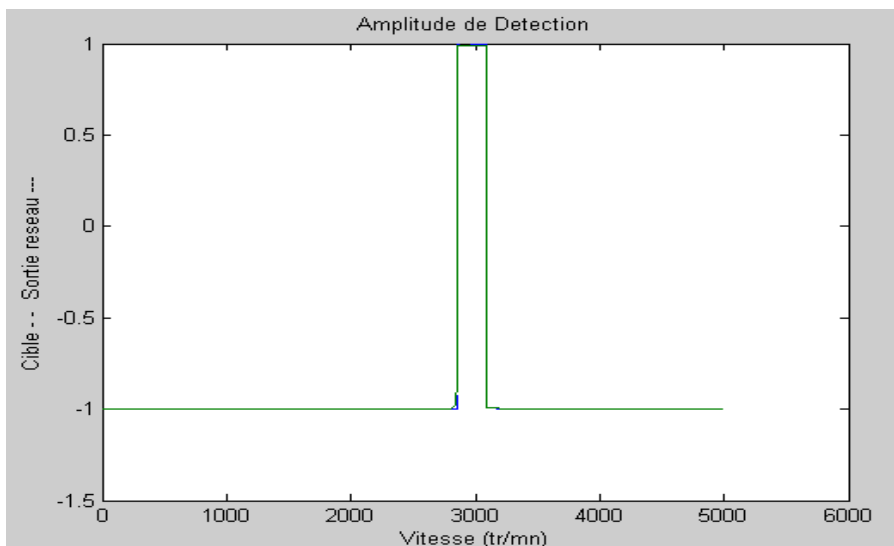


Figure 3.8 : Test avec une valeur acceptable

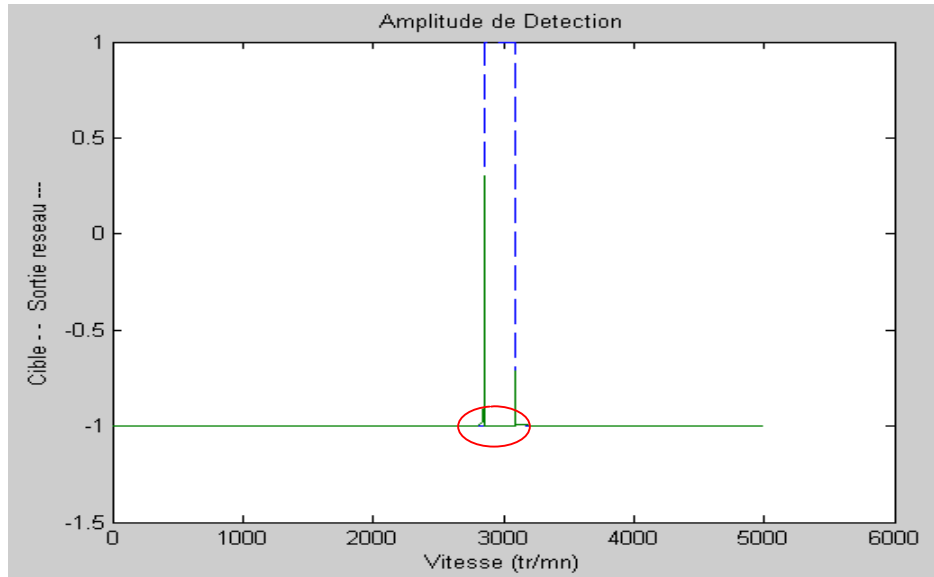


Figure 3.9 : Test avec une valeur non acceptable

Pour la figure 3.10, notre réseau est testé par une base de test appartenant aux valeurs de mauvais fonctionnement, tel que dans cet exemple l'intervalle de test est [2000 tr/mn 2239 tr/mn]. On observe que le réseau a bien détecté le mauvais fonctionnement ; c'est la zone sélectionnée en rouge dans la figure 3.10.

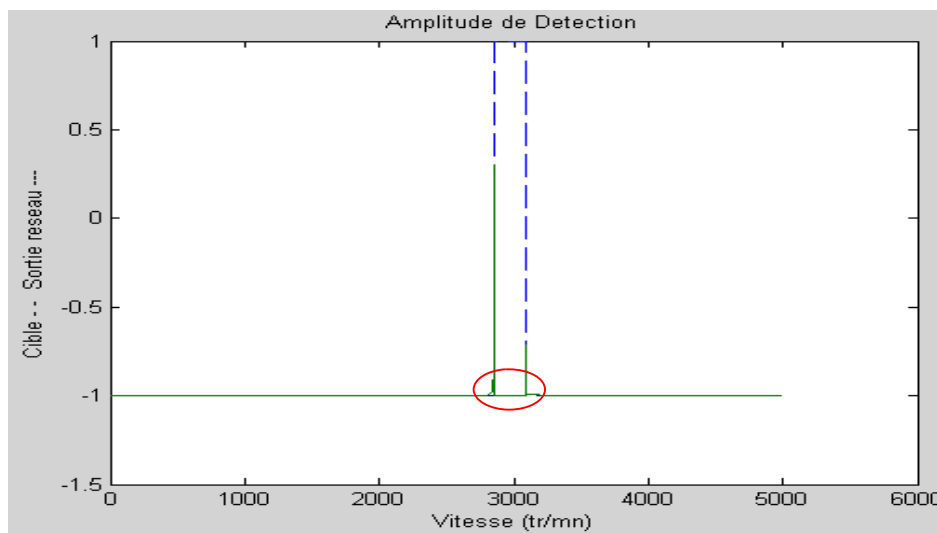


Figure 3.10 : Test avec un ensemble de valeurs de mauvais fonctionnement

### III-3.3.4 Performances du réseau

Dans la Figure 3.11 : on voit que le coefficient de corrélation « R » tend vers « 1 » ce qu'il montre la bonne corrélation, entre les sorties du réseau et les sorties désirées. Ceci est bien expliqué par la superposition entre la ligne continue, qui est le meilleur ajustement linéaire entre les cibles et les sorties du réseau et la ligne discontinue, qui est l'ajustement parfait. Donc nous déduisons que notre réseau est très performant.

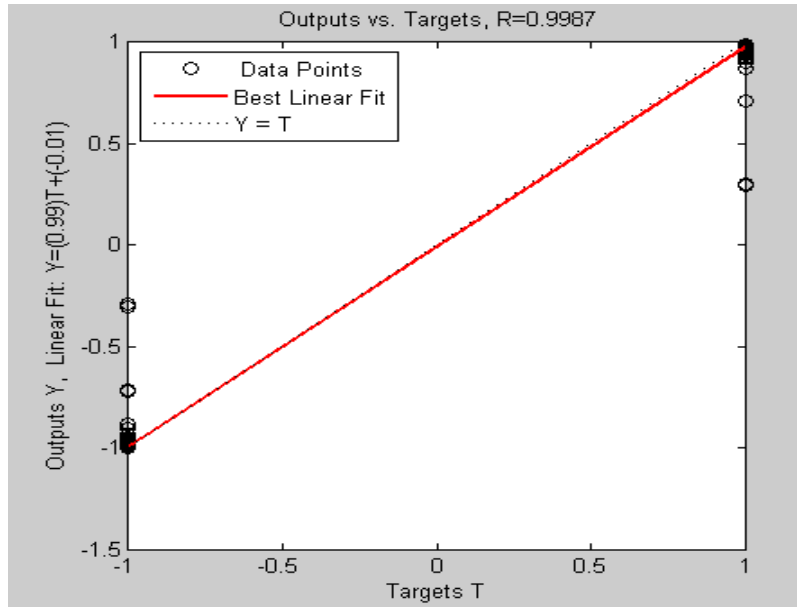


Figure 3.11 : Performances du réseau trouvé

### III-3.3.5 Généralisation du réseau

Pour la généralisation du réseau nous avons augmenté l'amplitude des valeurs d'entrée du réseau, tel que nous avons multiplié le vecteur d'entée par « 1.01 », donc on remarque que le graphe des sorties du réseau en vert est légèrement décalé à celui de la fonction désirée en bleu discontinu (voir la zone sélectionnée en rouge sur la Figure 3.12).

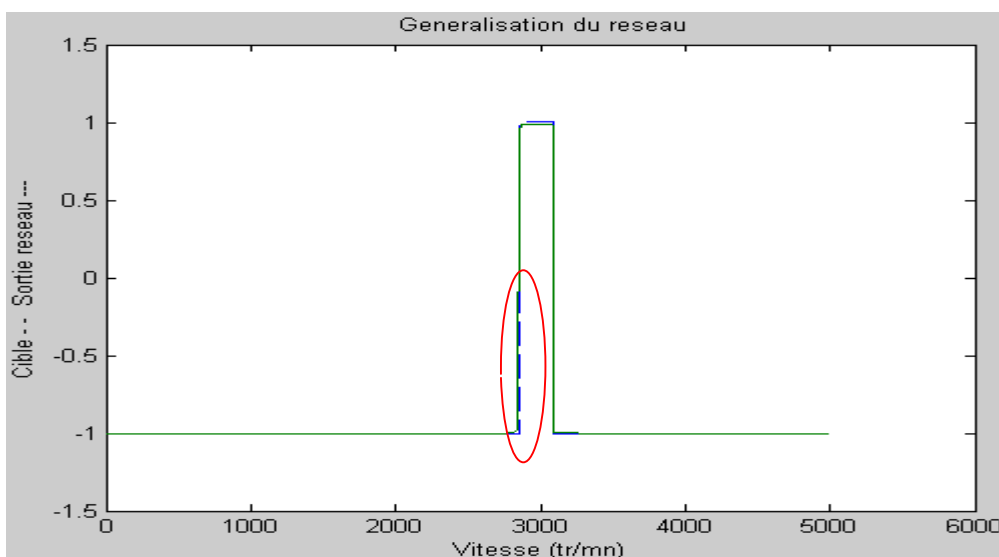


Figure 3.12 : Généralisation du réseau

La figure 3.13 représente les performances du réseau généralisé ; on remarque que la droite ( $Y = m T + b$ ) du meilleur ajustement linéaire est légèrement décalée par rapport à celle de l'ajustement parfait. Cette remarque est bien montrée par les coefficients :  $m = 0.98$ ,  $b = -0.018$  et  $R = 0.99045$ . 'm' est la pente de la droite du meilleur ajustement linéaire, si 'm' égale à '1', nous obtiendrions un ajustement parfait (les deux droites seront coïncidées).

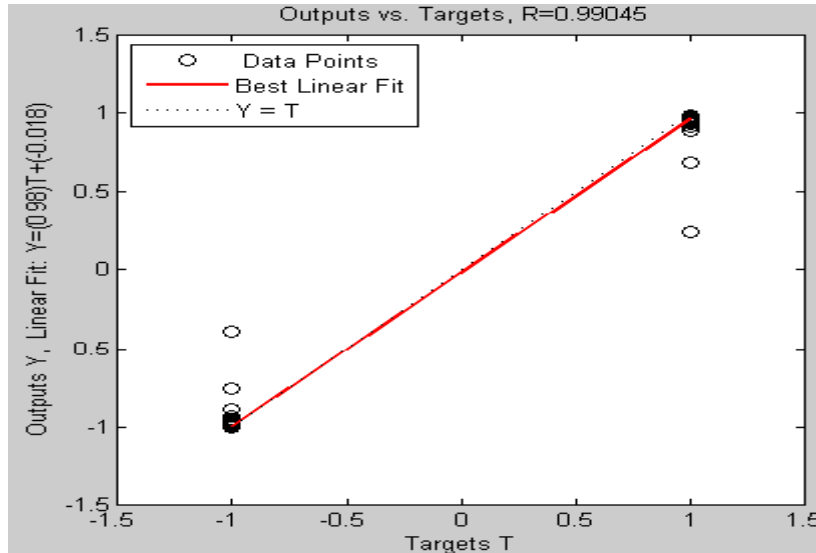


Figure 3.13 : Performances du réseau généralisé

### III-4 SURVEILLANCE DE LA TEMPERATURE D'HUILE DANS LES SORTIES PALIERS

#### III-4.1 Méthodologie du travail

On mesure la température d'huile à la sortie de chaque palier par des sondes de température, tel que la température minimale acceptable est égale à 45 °C et la température maximale est égale à 85 °C, donc l'intervalle de températures admissibles c'est [45 °C 85°C], pour chaque palier.

L'erreur limite donnée au réseau est  $E_i = 10^{-4}$ , le nombre d'itérations imposé  $it = 1000$  et le pas de calcul c'est  $p = 0.001$ , le taux d'apprentissage  $\mu = 0.5$ .

Pour obtenir l'architecture optimale nous devons suivre les étapes représentées dans l'organigramme de la figure 3.3. Le nombre de neurone maximum donné pour la couche cachée est  $N_i = 20$ , donc l'architecture optimale trouvée est : « 401-11-04 » qui représente la plus petite erreur  $8.34153e-005$ , voir tableau 3.2.

La figure 3.14 montre l'architecture optimale du réseau étudié, le nombre de neurones de la couche cachée  $N = 11$  tel que le vecteur d'entrée  $X = [x_1, x_2, \dots, x_M]^T$  et le vecteur de sortie  $y = [y_1, y_2, y_3, y_4]$ ,  $M = 401$  dans notre cas. Les équations (1) et (2) représentent respectivement le calcul des sorties des neurones de la couche cachée et de la couche de sortie.

Architecture	Tap (s)	Itérations	R (corrélation)	RG	MSE
401-01-04	26.19	53	0.3664	0.3697	0.316752
401-02-04	202.48	731	0.4985	0.5021	0.243762
401-03-04	46.33	101	0.3101	0.3144	0.208584
401-04-04	206.00	836	0.9325	0.9334	0.151502
401-05-04	184.85	581	0.9354	0.9351	0.129797
401-06-04	190.14	551	0.9354	0.9350	0.0802512
401-07-04	202.55	496	0.9352	0.9346	0.0816648
401-08-04	218.71	481	0.9359	0.9357	0.0337968
401-09-04	365.46	756	0.9361	0.9359	0.0331011
401-10-04	319.69	581	0.9353	0.9348	0.0428223
<b>401-11-04</b>	<b>153.72</b>	<b>266</b>	<b>0.9999</b>	<b>0.9998</b>	<b>8.34153e-005</b>
401-12-04	160.22	248	0.9999	0.9866	9.4535e-005
401-13-04	463.82	161	0.9369	0.9362	0.010978
401-14-04	575.19	686	0.9915	0.9876	0.0129909
401-15-04	419.14	511	0.9999	0.9867	7.74417e-005
401-16-04	642.55	1000	0.9511	0.9504	0.00864938
401-17-04	796.05	1000	0.9953	0.9861	0.00928352
401-18-04	719.03	1000	0.9991	0.9864	0.00881647
401-19-04	440.30	477	0.9999	0.9864	8.82747e-005
401-20-04	755.05	1000	0.9963	0.9864	0.00901278

Tableau 3.2 : variation de l'erreur quadratique moyenne en fonction de neurones de la couche cachée pour la surveillance de la température d'huile dans les sorties paliers.

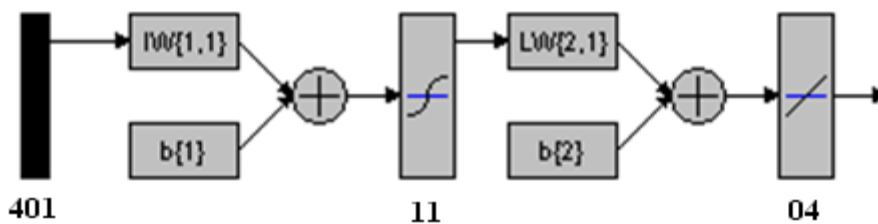


Figure 3.14 : Architecture optimale trouvée

### III-4.2 Explication des résultats

#### III-4.2.1 Apprentissage du réseau

Le graphe d'apprentissage converge vers l'erreur fixée d'ordre  $10^{-4}$  après 266 itérations dans une durée de 153.72 s, voir figure 3.15.

On remarque une convergence lente de 0 à 250 itérations, après ce nombre d'itérations, nous avons eu une convergence rapide du graphe d'apprentissage. Bien sûr, l'adaptation des poids dans le début d'apprentissage sera difficile et nécessite un nombre suffisant d'itérations. Une fois, les poids commencent à s'adapter, il sera facile d'avoir une convergence rapide.

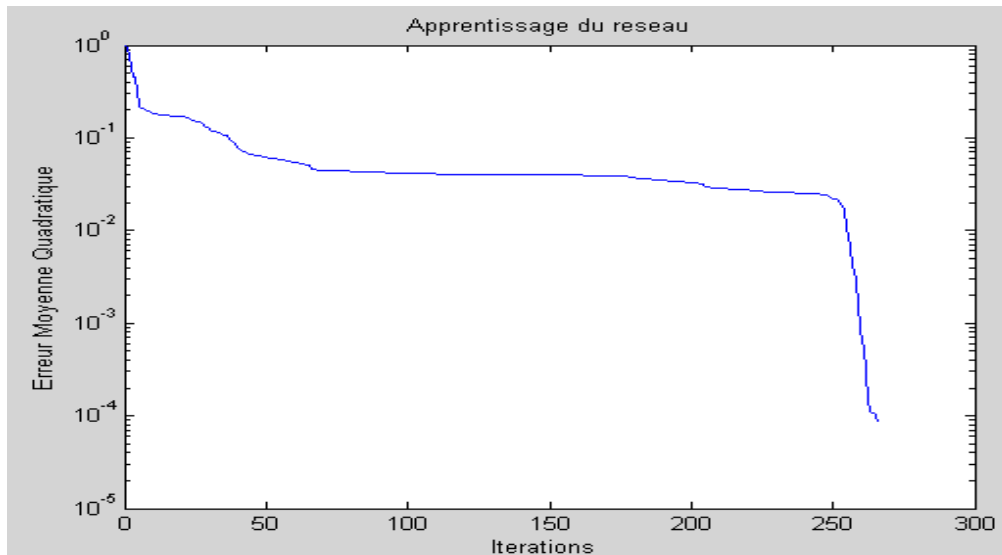


Figure 3.15 : Apprentissage du réseau de neurone « 401-11-04 ».

### III-4.2.2 Simulation des sorties du réseau

On voit que les graphes en ligne continue du réseau sont bien coïncidés avec les graphes en ligne discontinue des fonctions désirées, ce qui explique la bonne simulation de l'architecture trouvée, voir figure 3.16.

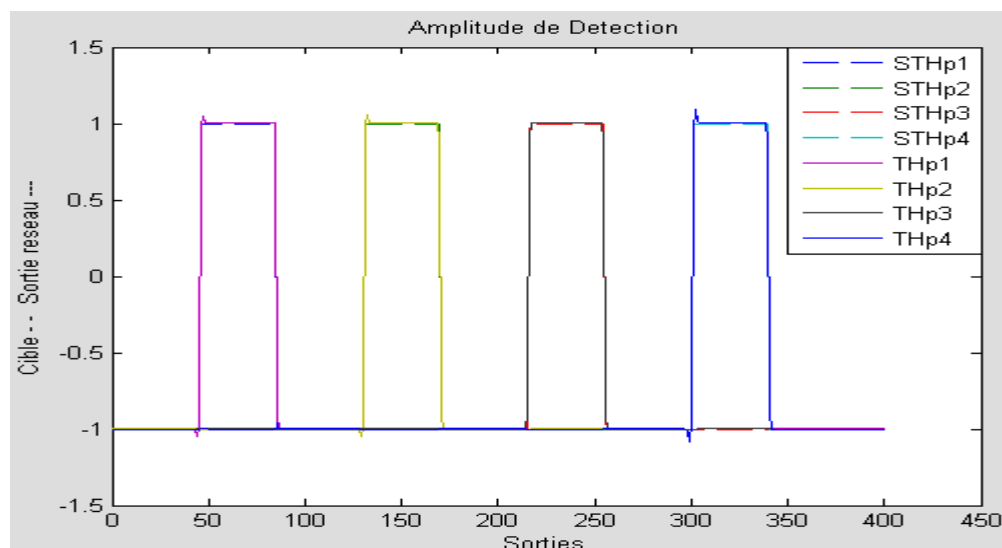


Figure 3.16 : Simulation des sorties du réseau

### III-4.2.3 Détection et localisation des défauts (test du réseau)

Pour bien savoir que notre réseau est fiable et capable de détecter un défaut et de le localiser on a injecté des défauts quelconques (fausses valeurs) dans le réseau et on observe la réaction de ce dernier. C'est un test qui s'est fait sur quatre exemples :

**Exemple 1 :** les quatre valeurs de température sont acceptables, on a donné 50°C pour le palier 1 (THp1), 60°C pour le palier 2 (THp2), 80 °C pour le palier 3 (THp3) et 70°C pour le palier 4 (THp4). Les résultats sont présentés sur la figure 3.17. Le réseau n'a indiqué aucun défaut.

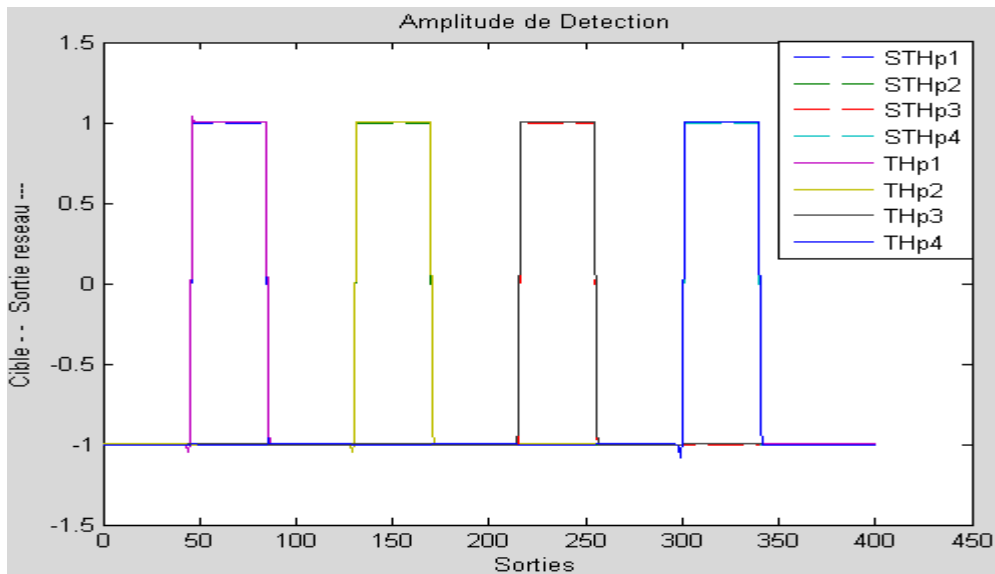


Figure 3.17 : Test avec des valeurs acceptables

**Exemple 2 :** trois valeurs acceptables et une qui n'est pas bonne, pour le palier 1 (THp1) on a donné 60 °C, pour le palier 2 (THp2) c'est 80 °C, pour le palier 3 (THp3) c'est 50 °C et 10 °C pour le palier 4 (THp4).

On observe une réaction du réseau pour la valeur de la température d'huile dans la sortie du palier 4 (THp4). Nous avons utilisé le cercle rouge pour montrer la réponse du réseau pour cette mauvaise valeur de test, voir figure 3.18.

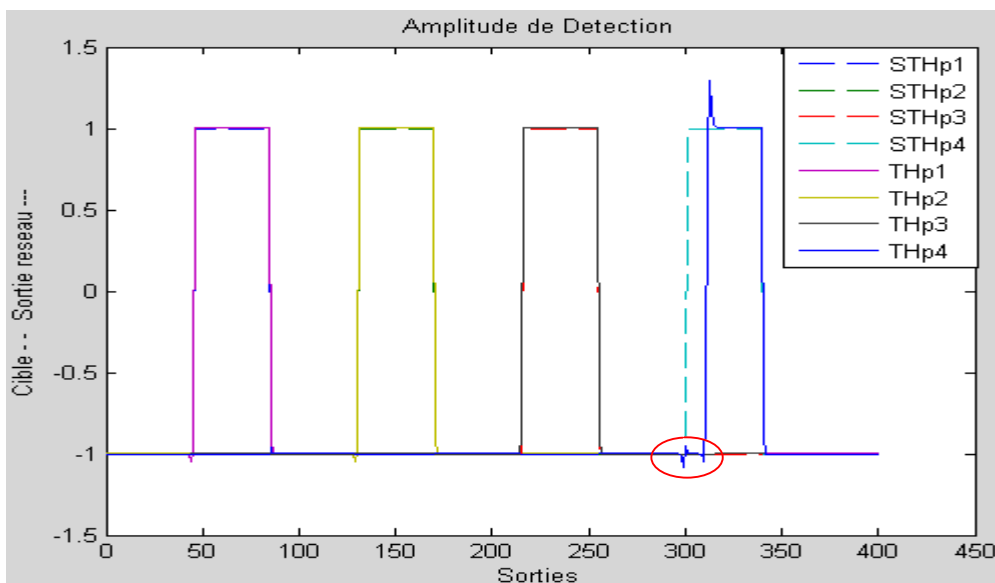


Figure 3.18 : Test avec une valeur inacceptable

**Exemple 3 :** deux valeurs de températures sont bonnes les autres sont mauvaises, donc on a met 90°C pour le palier 1, 50°C pour le palier 2, 20°C pour le palier 3 et 60°C pour le palier 4. C'est la même chose le réseau déclare une alarme pour le palier 1 (THp1) et le palier 3 (THp3), a cause des températures inacceptables reçues par le réseau. Les zones de mauvais fonctionnement sont sélectionnées par les deux cercles en rouge, voir figure 3.19.

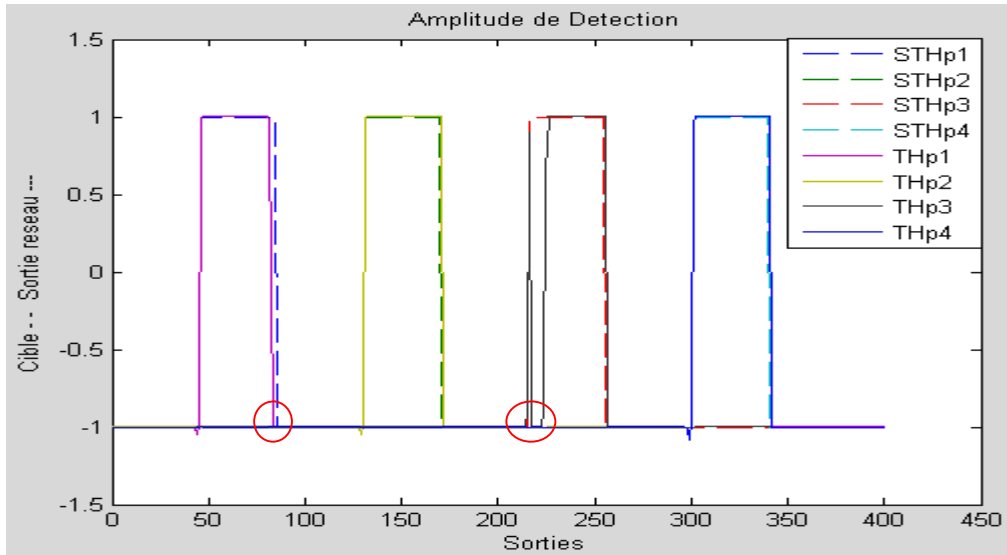


Figure 3.19 : Test avec deux valeurs inacceptables

**Exemple 4 :** On teste maintenant avec des bases de test qui n'appartiennent pas à l'intervalle admissible, on a donné une base de 40 valeurs ; de 90 °C jusqu'à 129°C pour le palier 1 (THp1), de 0°C jusqu'à 39 °C pour le palier 2 (THp2), de 300 °C jusqu'à 339 °C pour le palier 3 (THp3) et de 340°C jusqu'à 379 °C pour le palier 4 (THp4). Le réseau déclare quatre alarmes dans les quartes paliers, à cause des températures inacceptables reçues par le réseau. Voir figure 3.20.

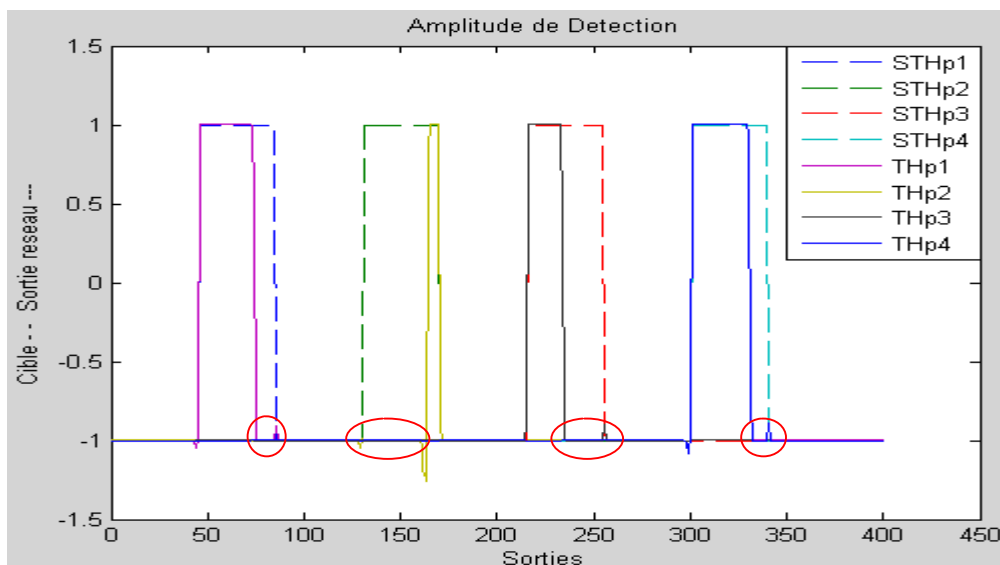


Figure 3.20 : Test avec quatre bases de valeurs inacceptables

### III-4.2.4 Évaluation des performances

Dans la Figure 3.21 : on voit que le coefficient de corrélation « R » tend vers « 1 » ce qui montre la bonne corrélation, entre les sorties du réseau et les sorties désirées. Ceci est bien expliqué par la superposition entre la ligne continue, qui est le meilleur ajustement linéaire entre les cibles et les sorties du réseau et la ligne discontinue, qui est l'ajustement parfait. Donc nous déduisons que notre réseau est très performant.

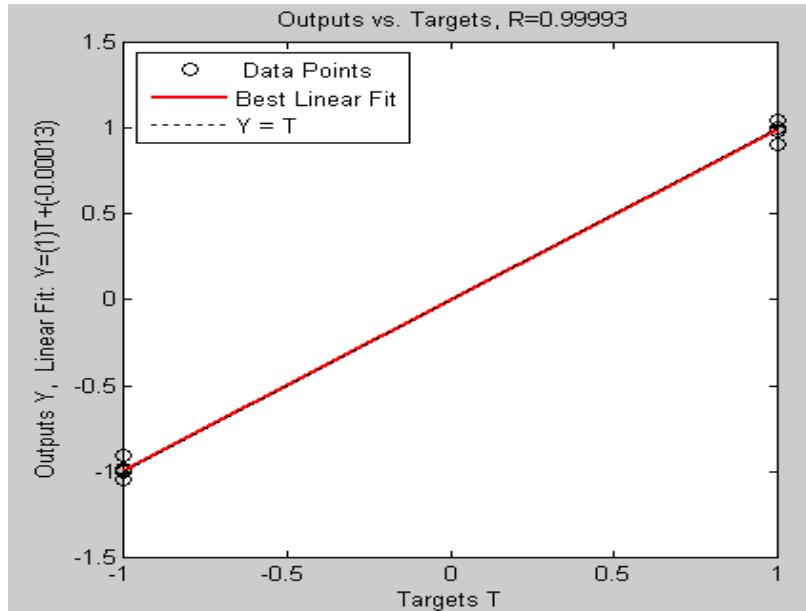


Figure 3.21 : Performances du réseau trouvé

### III-4.2.5 Généralisation du réseau

Pour la généralisation du réseau nous avons augmenté l'amplitude des valeurs d'entrée du réseau, tel que nous avons multiplié le vecteur d'entrée par « 1.002 », donc on remarque que les graphes des sorties du réseau, en lignes continues, sont légèrement décalés (dans quelques points) à ceux des fonctions désirées. Le réseau généralisé représente une bonne corrélation de 0.9998, voir les figures 3.22 et 3.23.

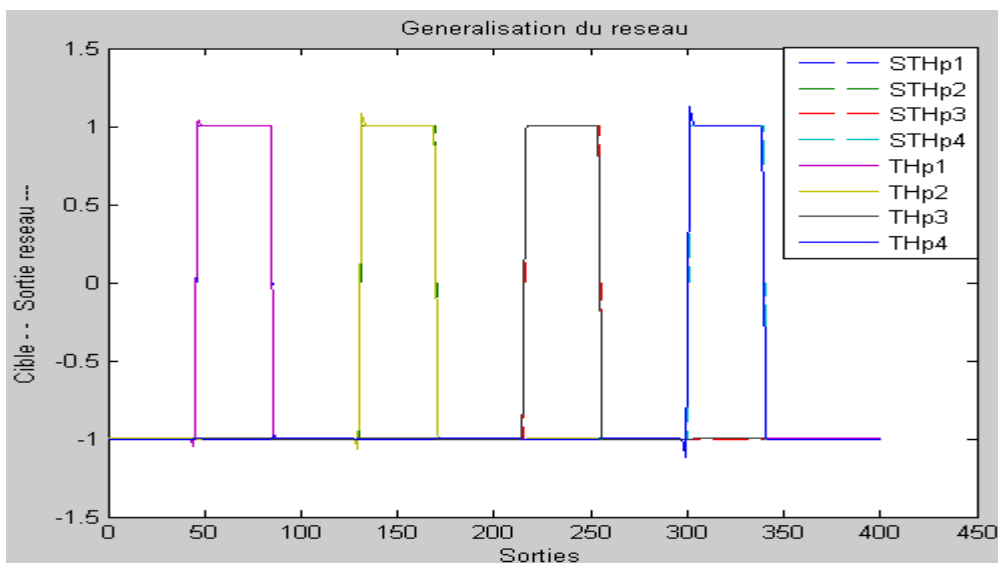


Figure 3.22 : Généralisation du réseau trouvé

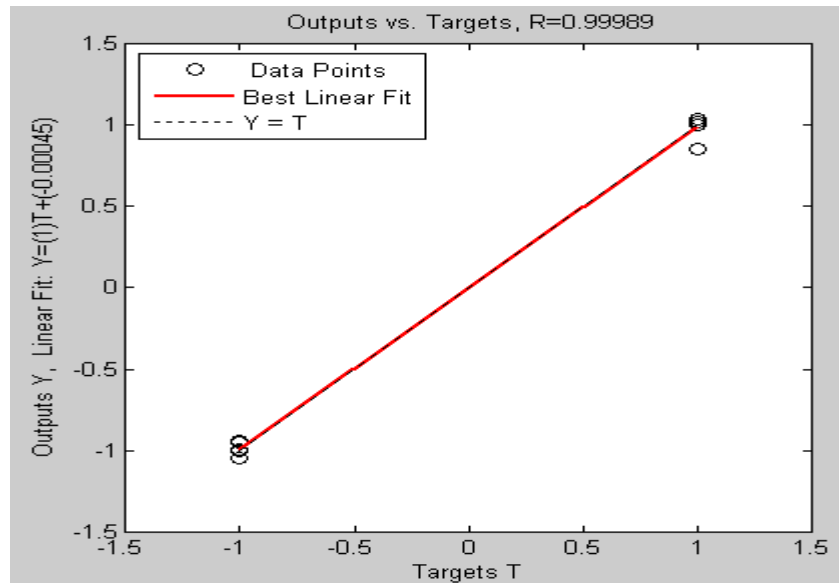


Figure 3.23: Performances du réseau généralisé

### III-5 RÉSUMÉ DE TOUS LES RÉSULTATS ET COMMENTAIRES

Nous suivons les mêmes étapes de la surveillance de température d'huile dans les sorties paliers, on obtient les résultats de la surveillance des autres paramètres (vitesse de rotation, vibrations, pression de vapeur, température de vapeur et débit d'huile). Le tableau 3.3 représente les architectures optimales de chaque paramètre à surveiller, le temps d'apprentissage, le nombre d'itérations, la corrélation du réseau (R), la corrélation du réseau généralisé (RG) et l'erreur moyenne quadratique d'apprentissage (MSE). La méthode de calcul, les graphes de simulation et l'explication des résultats sont bien présentés dans l'annexe III.

Paramètres	Architecture	Tap (s)	Itérations	R (corrélation)	RG	MSE
Vitesse de rotation	5001-06-01	50.15	95	0.9987	0.9904	0.000487846
Température d'huile dans les sorties paliers	401-11-04	153.72	266	0.9999	0.9998	8.34153e-005
Vibrations relatives	951-12-04	113.86	111	1.0000	0.9999	1.90756e-005
Débit d'huile	801-10-04	202.77	249	1.0000	1.0000	1.80361e-006
Température de vapeur	801-08-04	76.82	110	1.0000	0.9919	9.96445e-011
Pression de vapeur	951-08-04	84.70	111	1.0000	0.9999	1.46169e-006

Tableau 3.3 : Résumé de toutes les architectures optimales trouvées

Après une analyse des résultats de ce tableau et des tableaux précédents nous pouvons conclure que :

- L'erreur moyenne quadratique et le temps d'apprentissage ne dépendent pas seulement du vecteur d'entrée du réseau mais aussi du vecteur de sortie (le cas des vibrations par rapport aux autres paramètres).
- Le nombre de neurone de la couche cachée de chaque réseau est entre 06 et 12, malgré que les vecteurs d'entrée soient grands.
- Des architectures de même vecteurs d'entrée et de sortie sont strictement différentes dans le temps d'apprentissage, le nombre d'itération et dans l'erreur moyenne quadratique (cas de vibrations et de pression vapeur).
- On n'aura jamais un bon apprentissage si on diminue le nombre d'itérations ou l'erreur d'apprentissage. La figure 3.24 représente un apprentissage non parfait, de l'architecture « 401-11-04 » de la surveillance de température d'huile, tel que nous avons fixé le nombre d'itérations à **251** et l'erreur trouvée est de **0.0215607**.

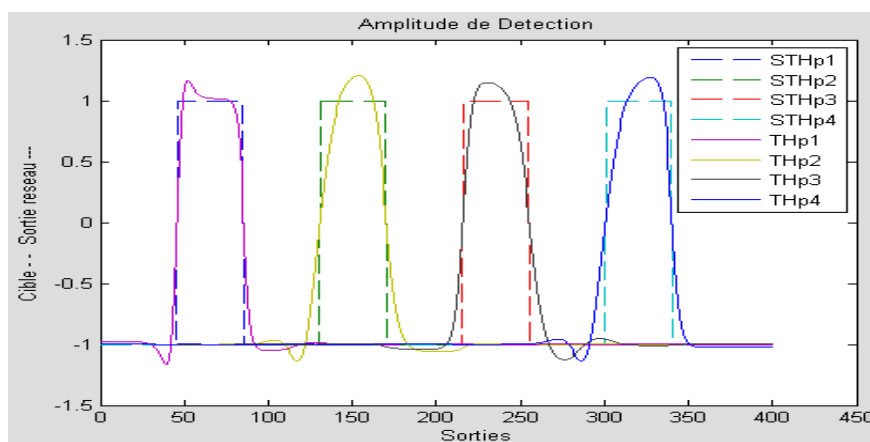


Figure 3.24 : Apprentissage non parfait du réseau « 401-11-04 »

- Le temps d'apprentissage (Tap) augmente, d'une manière alternative, avec l'augmentation du nombre de neurones dans la couche cachée. Par contre l'erreur moyenne quadratique (MSE) diminue jusqu'à un point minimum et commence à se stabiliser par la suite. Voir les figures (3.25, 3.26, 3.27 et 3.28).

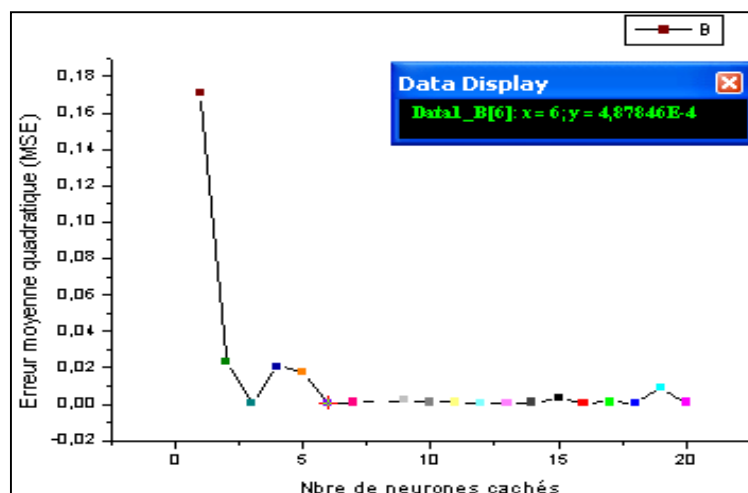


Figure 3.25 : Variation de l'erreur moyenne quadratique en fonction de nombre de neurones cachés pour le réseau « 5001-06-01 » de la surveillance de vitesse de rotation.

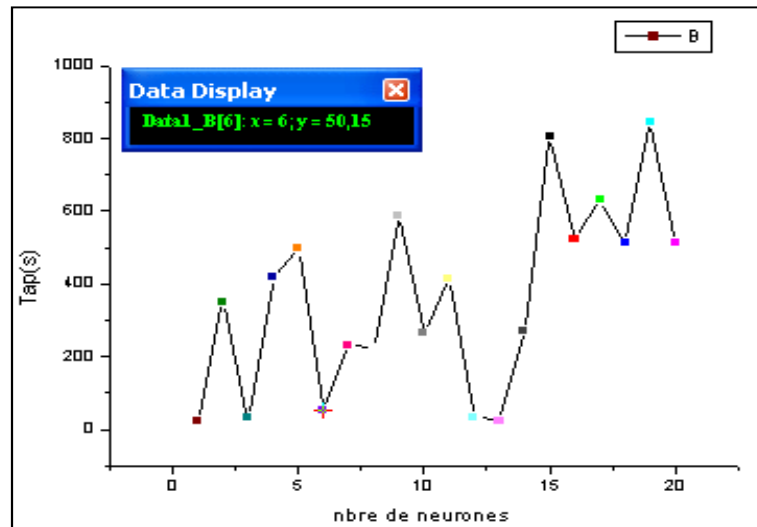


Figure 3.26: Variation du temps d'apprentissage en fonction de nombre de neurones cachés pour le réseau « 5001-06-01 » de la surveillance de vitesse de rotation.

Nous observons sur le graphe de la figure 3.27 que l'erreur minimale est celle reliée au nombre de neurones cachés égale à **15** (MSE=  $7.74417e-005$ ), mais nous l'avons pas choisi comme erreur optimale du réseau ; car elle représente un temps d'apprentissage très élevé (Tap = **419** s). L'erreur optimale est celle reliée au nombre de neurones égale à **11** (MSE =  $8.34153e-005$ ), le temps d'apprentissage de cette erreur est **153.72** s, voir la figure 3.28.

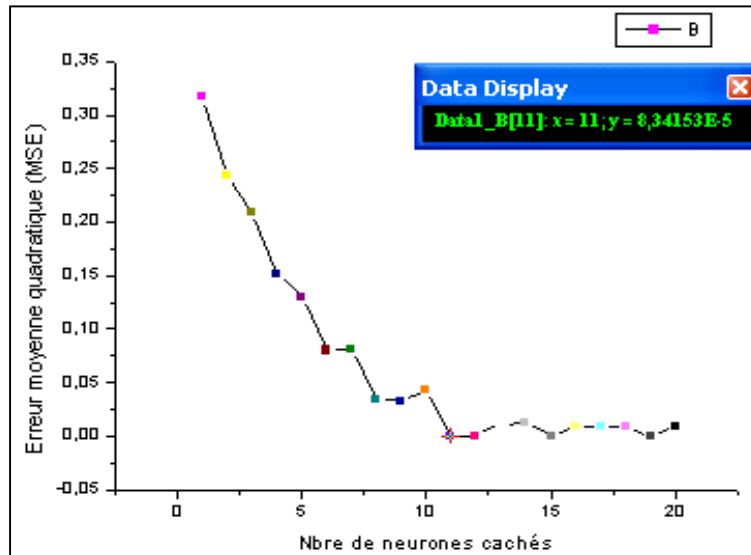


Figure 3.27: Variation de l'erreur moyenne quadratique en fonction de nombre de neurones cachés pour le réseau « 401-11-04 » de la surveillance de température d'huile.

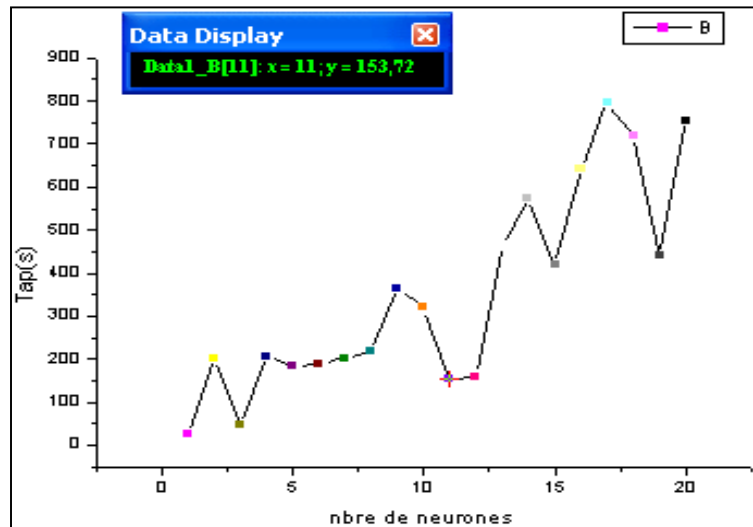


Figure 3.28 : Variation du temps d'apprentissage en fonction de nombre de neurones cachés pour le réseau « 401-11-04 » de la surveillance de température d'huile.

## CONCLUSION

Parmi les avantages essentiels du réseau de neurone artificiel utilisé dans cette partie :

- On peut tout d'abord surveiller plusieurs points au même temps sans utiliser pour chaque point ou paramètre un réseau propre à lui. Car l'utilisation d'un réseau de neurone pour chaque paramètre à surveiller va générer un encombrement des signaux dans la salle de commande ce qui demande beaucoup d'agents de surveillance pour bien maîtriser la situation. Cette situation est bien évitée par l'architecture neuronale proposée.
- La rapidité d'apprentissage due à l'utilisation de l'algorithme de Levenberg-Marquardt telle que cette rapidité permet au réseau de réagir bien avec les données obtenues à partir des capteurs.
- Ce type de réseau de neurone nous permet de réaliser une grande étape du diagnostic c'est la localisation des défauts.

Parmi les inconvénients ou les difficultés rencontrées concernant le calcul des architectures neuronales optimales :

- Une grande difficulté dans le calcul de l'architecture optimale lorsque on augmente le nombre de neurones d'entrée ou de sorties ; l'exemple de la vitesse de rotation par rapport aux autres paramètres.
- Une erreur minimale parmi les erreurs calculées n'est pas forcément implique une architecture optimale ; Parfois on trouve une architecture optimale qui correspond à une erreur qui n'est pas minimale. tel que l'exemple de la surveillance des vibrations ; l'architecture optimale trouvée (951-12-04) ne correspond pas à une erreur moyenne quadratique minimale, ceci revient à la phase de test, si ce dernier n'est pas vérifié, il faut choisir une autre erreur qui représente le bon test ( même si cette erreur est supérieure à la première) ; en suivant l'organigramme représenté dans la figure 3.4.
- Parfois les valeurs à la limite des intervalles dans la phase de test représentent des pics dans les graphes.

**CHAPITRE IV : ETUDE DE CAS -**  
**DEVELOPPEMENT D'UN SYSTEME**  
**D'IDENTIFICATION DE LA CAUSE PROBABLE**  
**DES DEFAILLANCES**

## INTRODUCTION

En réalité la méthode du diagnostic étudiée dans ce chapitre est une combinaison entre deux méthodes, la première appartient aux méthodes descriptives et la deuxième c'est une méthode probabiliste. Nous avons cherché au premier temps toutes les combinaisons des événements élémentaires (causes probables) qui conduisent à un événement redouté par la méthode de l'arbre de défaillance voir figures (4.3, 4.4, 4.5, 4.6, et 4.7). Nous allons étudier par la suite la variation des événements en fonction des événements redoutés, voir Tableau 4.1, tout ça sera traduit en valeurs numériques sous forme des intervalles de bon et de mauvais fonctionnement qui représentent par la suite l'alimentation d'un réseau de neurone artificiel. L'identification de la cause probable est une sélection qui sera faite par le réseau de neurone.

Dans cette étude on suppose que la défaillance est localisée au niveau du palier 1 par la première méthode de détection et de localisation et maintenant on va traiter les causes probables de cette défaillance par la deuxième méthode du diagnostic. Les capteurs de surveillance sont toujours les mêmes, voir Figure 4.1.

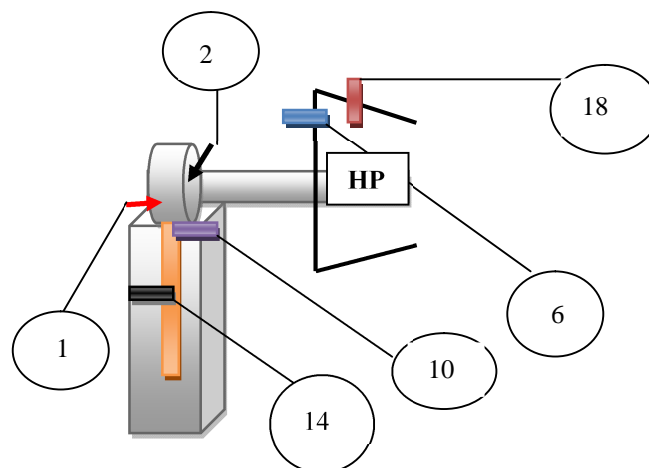


Figure 4.1 : Schéma représentant un quart de la turbine (palier 1 coté corps HP)

- (1) : capteur de vitesse, (2) : capteur de vibrations,
- (6) : capteur de pression vapeur,
- (10) : capteur de débit d'huile,
- (14) : capteur de température d'huile,
- (18) : capteur de température vapeur.

Le schéma de la figure 4.2 représente les principales étapes de l'approche proposée pour l'obtention d'un système de diagnostic ( identification des causes de défaillances). Le système est composé de deux parties en réalité :

- Une partie statique, représente la base de donnée de la deuxieme partie dynamique, cette base de donnée est obtenue à partir des arbres de causes des paramètres à surveiller.
- Une partie dynamique représente le reseau de neurone lui-même, telles que les entrées du reseau sont les sorties capteurs en fonction du temps ( $X_1(t), X_2(t), \dots, X_m(t)$ ) et les sorties du reseau sont les causes probables des défaillances au temps réel ( $y_i(t)$ ).

$X_i ( y_1, y_2, \dots, y_n )$  : paramètres surveillés en fonction des causes probables,  $i = [1, \dots, m]$ , dans notre cas  $m = 6$ .

$Y = F_1(X_1) + F_2(X_2) + \dots + F_m(X_m)$  : vecteur des causes en fonction des paramètres surveillés,  $F_1, F_2, \dots, F_m$  sont des fonctions supposées linéaires.

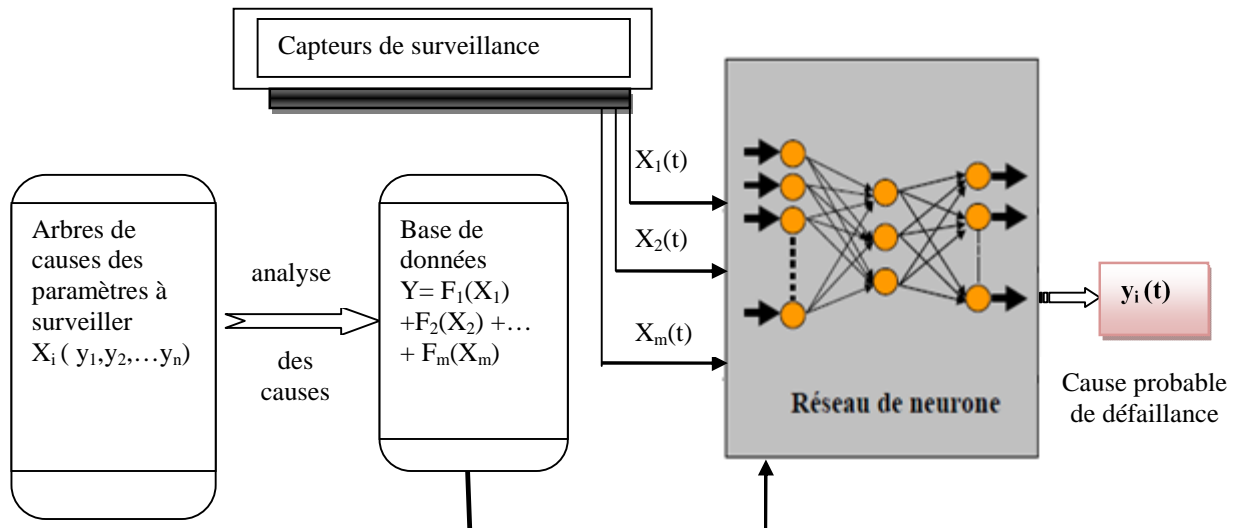


Figure 4.2 : Schéma explicatif de l'approche proposée pour l'identification des causes probables des défaillances.

#### IV-1 ARBRE DE DEFAILLANCES OU ARBRE DE CAUSES

La réalisation d'un arbre de défaillance nécessite les conditions suivantes :

- Connaissance technologique de la machine à étudier,
- Connaissance de l'historique des pannes de la machine,
- Connaissance du fonctionnement de la machine et le déroulement du procès,
- Savoir la relation entre les événements redoutés et leurs influences sur fonctionnement de la machine étudiée. [DOC 1984]

Le déroulement de telle méthode basé sur les quartes étapes suivantes : [HVE' 2007] [GUI 2005]

- 1- Identification des événements redoutés ;
- 2- Décomposition de chaque événement redouté en sous événements reliés par des portes logiques ;
- 3- Décomposition des sous événements jusqu'à l'obtention de l'élément de base ;
- 4- Identification des points faibles du système.

**IV-1-1 Événement Redouté ER1 : augmentation de vibrations**

L'augmentation des vibrations dans les turbines à vapeur sera toujours à cause d'un défaut de balourd, défaut d'alignement, perturbation de la pression de vapeur ou d'un problème de montage (jeu, desserrage, mauvaise fixation). Voir la figure 4.3.

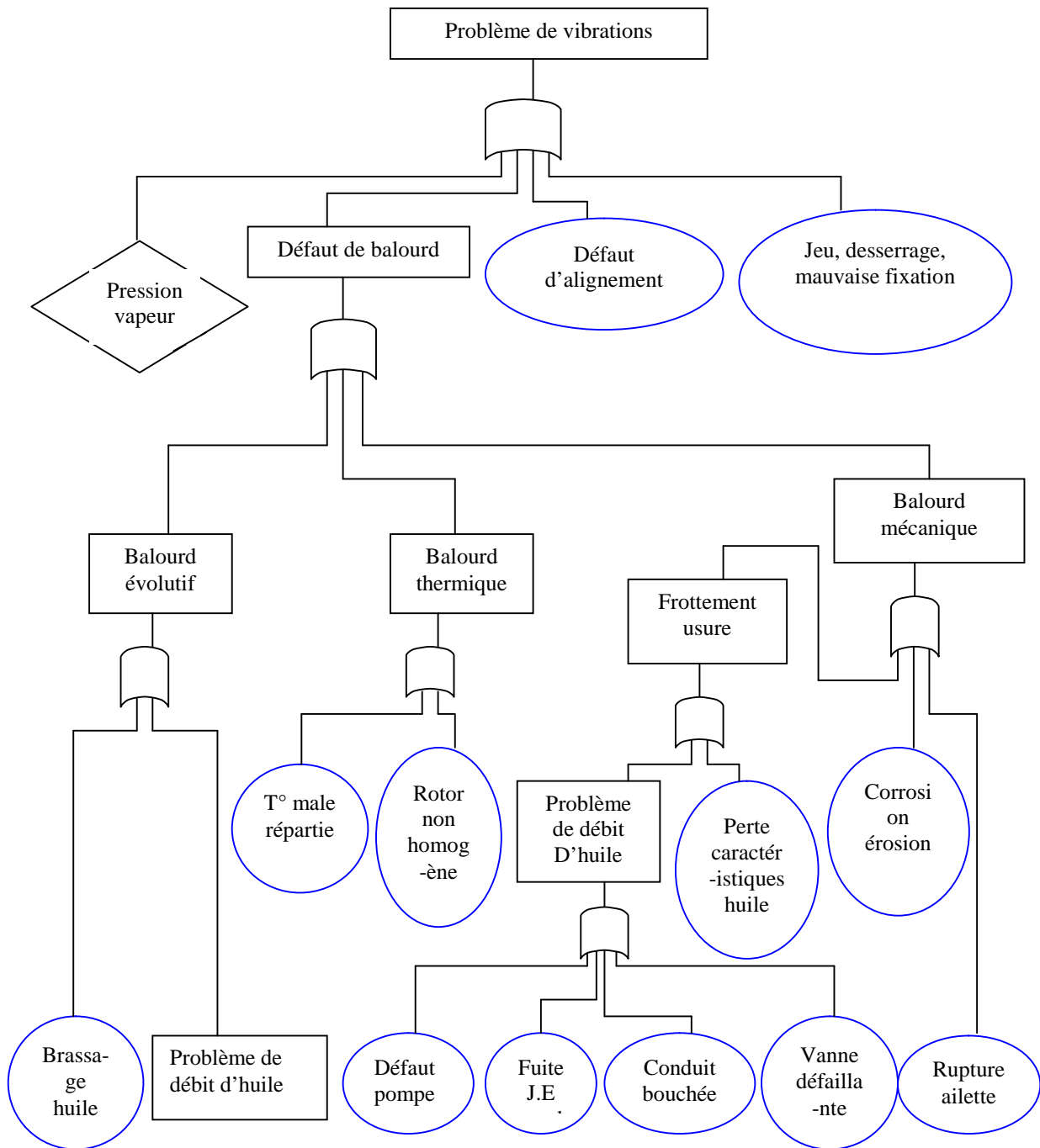


Figure 4.3 : Arbre de défaillances pour les vibrations

Les éléments représentés par des cercles bleus sont les événements de base (les dernières causes) et les éléments représentés par des rectangles sont des sous-événements (sous causes à développer) tel que, le défaut de balourd, la pression de vapeur, l'usure des paliers...etc.

**IV-1-2 Evénement Redouté ER2 : perturbation du débit d'huile**

Quatre sous-causes principales peuvent engendrer la perturbation du débit d'huile (augmentation ou diminution), pompe d'huile principale ou de premier secours est défaillante, fuite au niveau du joint d'étanchéité ou quelque part dans le réservoir, conduite d'huile bouchée ou coupée ou une vanne qui est défaillante ou mal réglée. Voir la figure 4.4.

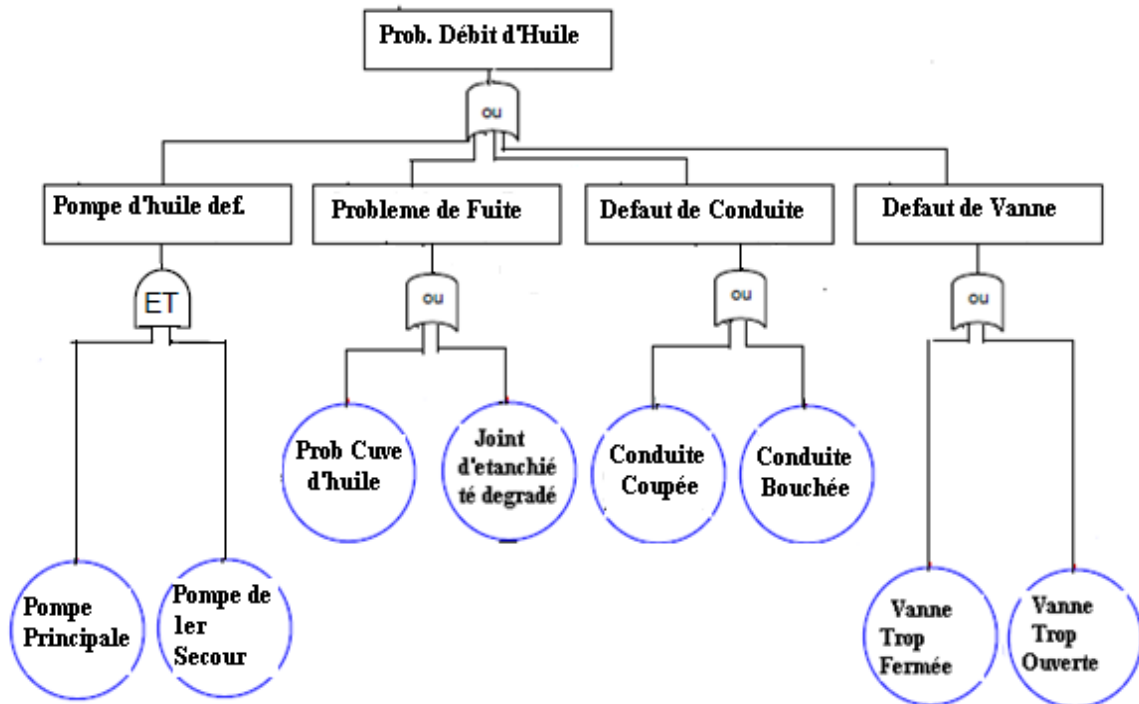


Figure 4.4 : Arbre de dédfaillances pour le débit d'huile

**IV-1-3 Evénement Redouté ER3 : problème de température d'huile des sorties paliers**

L'augmentation de la température d'huile dans les sorties des paliers est généralement à cause des frottements dans les paliers et parfois à cause des conditions d'utilisation d'huile (condition d'environnement, huile hors normes...). Voir la figure 4.5.

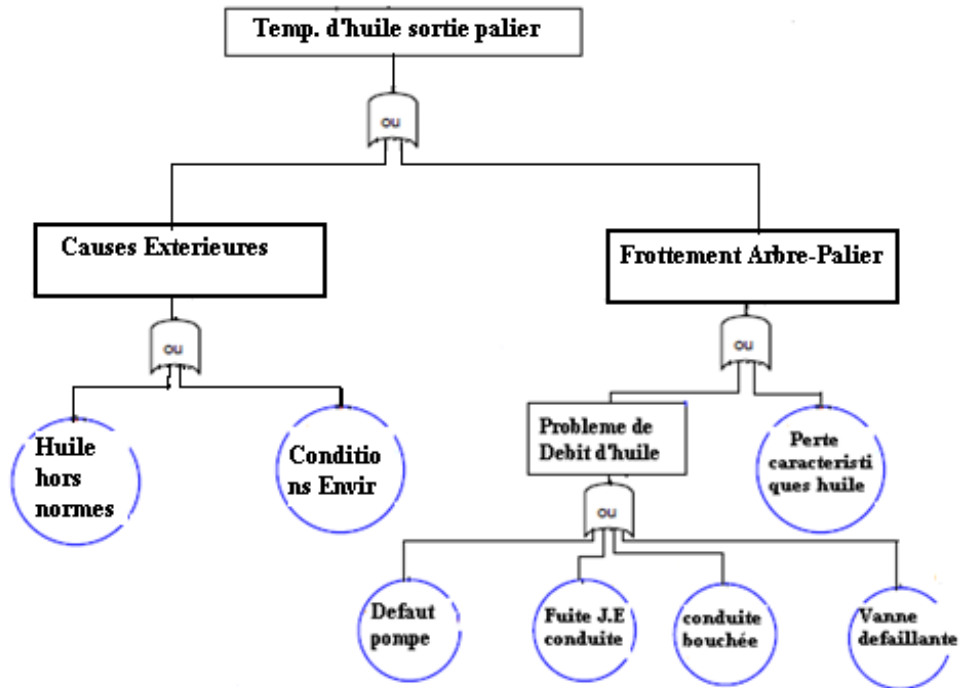


Figure 4.5 : Arbre de défaillances pour la température d'huile

**IV-1-5 Evénements Redoutés ER4 et ER5 : problème de vitesse de rotation et de température de vapeur**

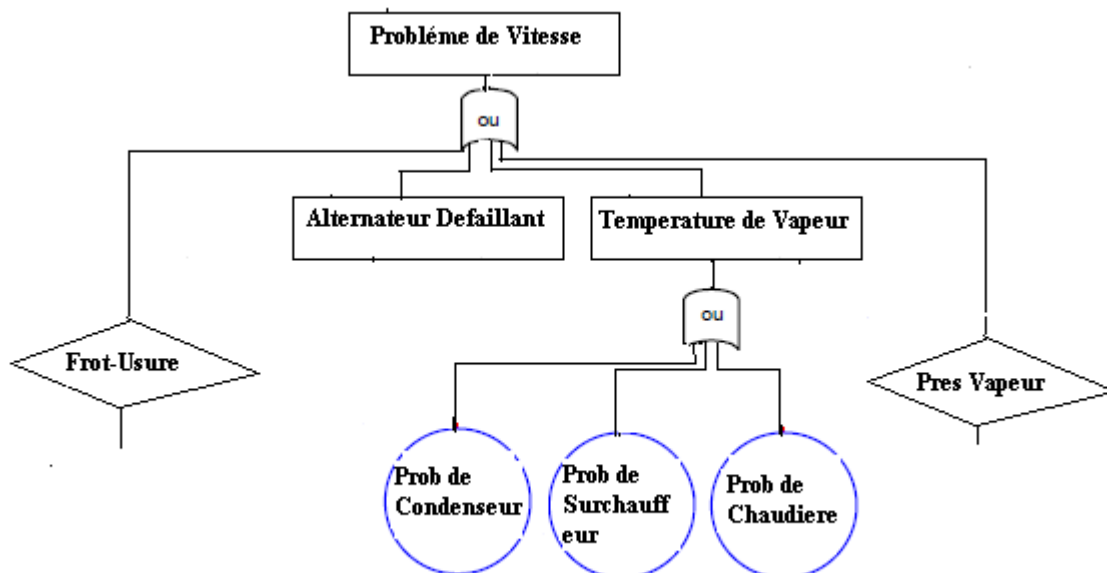


Figure 4.6 : Arbre de défaillances pour la vitesse de rotation et la température de vapeur

Nous remarquons sur la figure 4.6 que les deux paramètres, température et pression de vapeur, sont des éléments essentiels dans la variation de la vitesse de rotation. Ceci, revient aux lois de la thermodynamique et de turbomachines qui relient entre les trois paramètres (température, pression et vitesse). La pression de vapeur et le frottement sont représentés par des losanges car, ils sont développés ailleurs.

**IV-1-4 Événement Redouté ER6 : problème de pression vapeur**

La perturbation dans les paramètres de vapeur (pression et température) revient généralement aux trois éléments : les éléments de génération de vapeur (chaudière, surchauffeur), éléments de conduite de vapeur (conduites, vannes, clapets...) et les éléments de réception de vapeur (aubes, joints d'étanchéité, condenseur). Voir les figures 4.6 et 4.7.

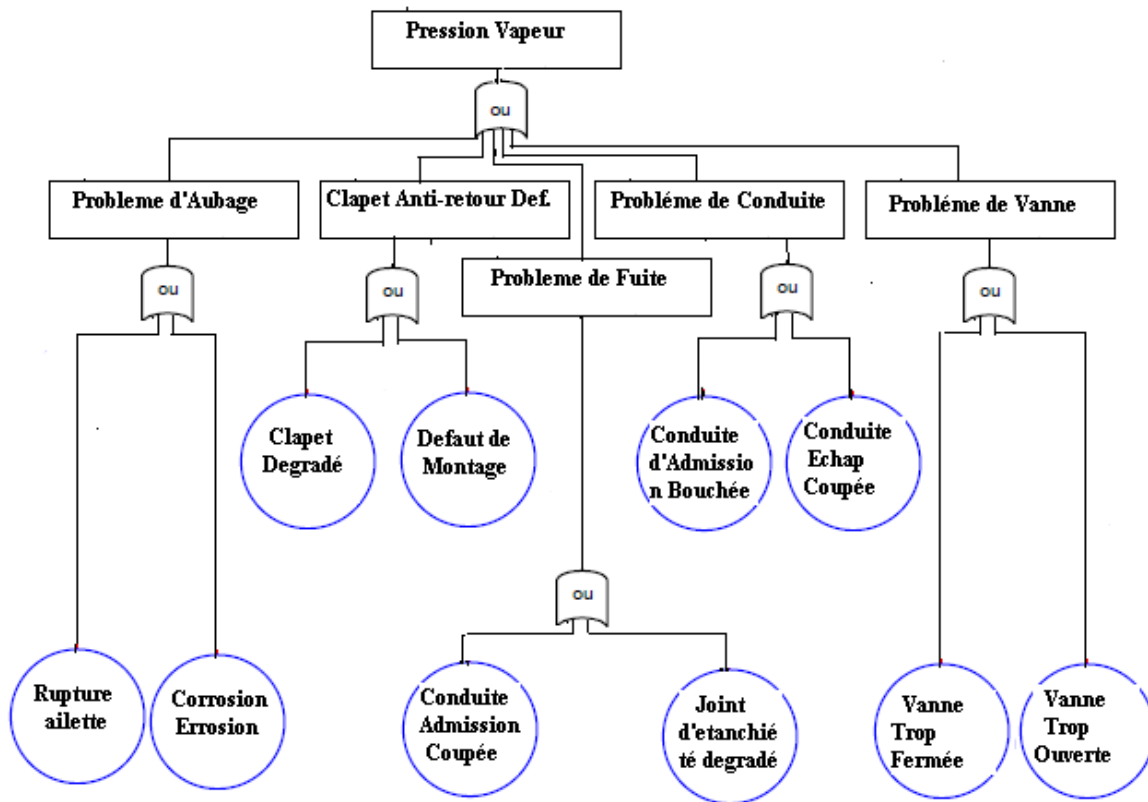


Figure 4.7 : Arbre de défaillances pour la pression vapeur

## IV-2 CAUSES PROBABLES DE DEFAILLANCE

Après avoir analysé les arbres de défaillances précédents, on peut extraire le tableau suivant :

Effets Causes Probables	Vibrations	Vitesse de rotation	Pression Vapeur	Débit d'Huile	Température d'Huile	Température de Vapeur
1- Perte ou Rupture ailette	A	D	D			
2- Corrosion-Érosion d'aubes	A	D	A			
3- Joint d'étanchéité vapeur dégradé		D	D			
4- Conduite d'admission vapeur bouchée ou coupée		DR	DR			
5- Vanne de vapeur trop ouverte		A	A			
6- Vanne de vapeur trop fermée		DR	DR			
7- Jeu, Mauvais serrage	A					
<b>8- Vanne d'huile trop ouverte</b>	A			A	D	
<b>9- Vanne d'huile trop fermée</b>	AR			D	A	
<b>10- Conduite bouchée</b>	A			D	A	
<b>11- Conduite coupée</b>	A			DR	A	
<b>12- Pompe d'huile défaillante</b>	AR			DR	AR	
<b>13- Perte de caractéristiques d'huile</b>	A				A	
<b>14- Délignage</b>	A					
<b>15- Rotor non homogène</b>	AR				A	
<b>16- Chaudière – Surchauffeur défaillants</b>	A					D
<b>17- Condenseur défaillant</b>						A

Tableau 4.1 : L'influence des causes probables de défaillances de la turbine sur les paramètres de fonctionnement de cette dernière

A : augmentation (codée par « 1 »), AR : augmentation rapide (codée par « +1 »),  
D : diminution (codée par « -1 »), DR : diminution rapide (codée par « - -1 »),

Pour les paramètres qui ne changent pas, sont codés dans le programme par « 0 ».

Le tableau 4.1 représente l'influence des causes probables de défaillances de la turbine sur les grandeurs physiques de fonctionnement de cette dernière. Par exemple pour la première cause on dit une augmentation de vibrations, diminution de vitesse et diminution de pression c'est probablement à cause d'une rupture ou une perte d'ailette de la turbine et ainsi de suite.

### VI-3 DIAGNOSTIC DE LA PREMIERE SERIE DE CAUSES

#### VI-3.1 Méthodologie de travail

Nous avons sélectionné les causes de 8 à 17 (bande verte) dans le tableau 4.1 et nous l'avons traité dans un seul réseau de neurone. Cette sélection est basée aussi sur la sélection des variables qui se répètent toujours en fonction de ces causes telles que le débit d'huile, les vibrations, la température d'huile et la température de vapeur. On laisse les autres causes qui ne dépendent pas en générale de ces variables pour un deuxième réseau de neurone. Le type du réseau de neurone utilisé est le MLP à quatre neurones dans la dernière couche. La minimisation de l'erreur d'apprentissage est faite par l'algorithme de Levenberg-Marquardt.

L'erreur limite donnée au réseau est de  $10^{-7}$ , le nombre d'itérations limite est **1000** et le pas de calcul c'est **0.001**.

Architecture	Tap(s)	Itérations	R	RG	MSE
750-01-04	15.23	28	0.8980	0.8751	0.298088
750-02-04	111.04	320	0.8985	0.8951	0.231267
750-03-04	326.58	1000	0.8580	0.8052	0.169039
750-04-04	60.75	125	0.9580	0.9050	0.137119
750-05-04	395.55	1000	0.9880	0.9550	0.0888211
750-06-04	62.71	105	0.9950	0.9850	0.0529986
750-07-04	540.28	780	0.9980	0.9850	0.0382671
750-08-04	570.70	820	0.9995	0.9949	0.0121545
<b>751-09-04</b>	<b>191.06</b>	<b>274</b>	<b>1.0000</b>	<b>0.9950</b>	<b>1.98662e-008</b>
750-10-04	680.01	826	0.9990	0.9945	0.0135518
750-11-04	762.58	1000	0.9995	0.9949	0.0121542
750-12-04	127	148	1.0000	0.9949	4.27102e-008
750-13-04	302.58	308	1.0000	0.9950	3.06071e-008
750-14-04	1159.84	1000	0.9998	0.9944	0.00997069
750-15-04	285.44	276	1.0000	0.9950	2.20399e-008

Tableau 4.2 : Variation de l'erreur quadratique moyenne en fonction de neurones de la couche cachée pour le réseau « **751-09-04** ».

Tap : temps d'apprentissage du réseau, R : coefficient de corrélation du réseau, RG : coefficient de corrélation du réseau généralisé, MSE : erreur moyenne quadratique.

Pour obtenir l'architecture optimale, nous avons varié le nombre de neurone de la couche cachée de **1** jusqu'à **15**, donc l'architecture optimale trouvée est : « **751-09-04** » qui représente la plus petite erreur **1.98662e-008**, voir Tableau 4.2.

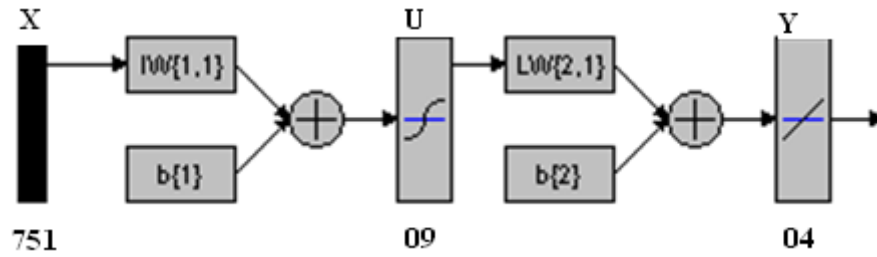


Figure 4.8 : Architecture optimale trouvée

### IV-3.2 Explication des résultats

#### IV-3.2.1 Apprentissage du réseau

Le graphe d'apprentissage converge vers l'erreur fixée d'ordre  $10^{-7}$  après 274 itérations dans une durée de 191.06s, voir figure 4.9.

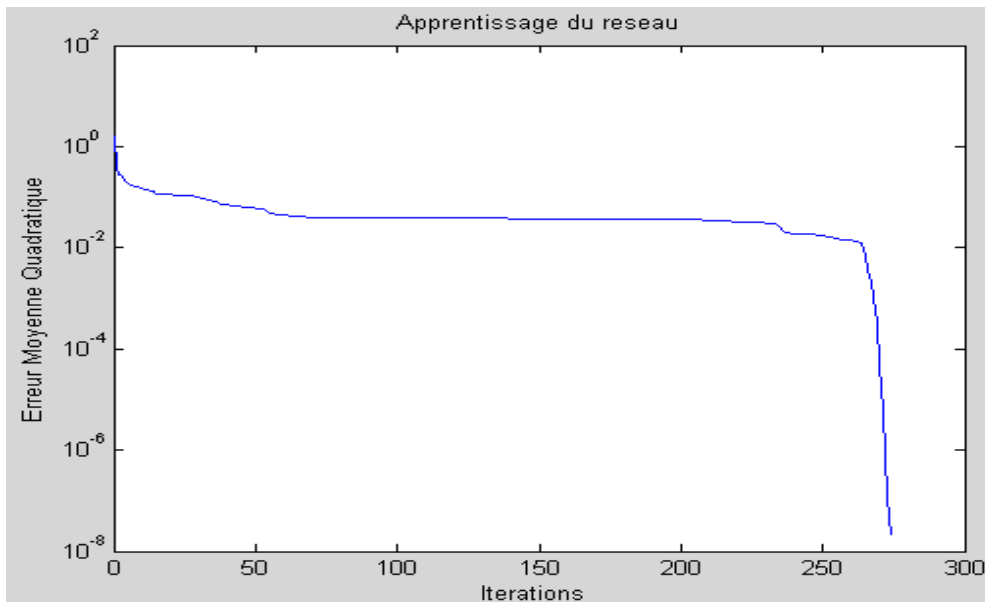


Figure 4.9 : Apprentissage du réseau de neurone « 751-09-04 ».

#### IV-3.2.2 Simulation des sorties de réseau

On voit que les graphes en ligne continue du réseau sont bien coïncidés avec les graphes en ligne discontinue des fonctions désirées, ce qui explique la bonne simulation de l'architecture trouvée, voir figure 4.10.

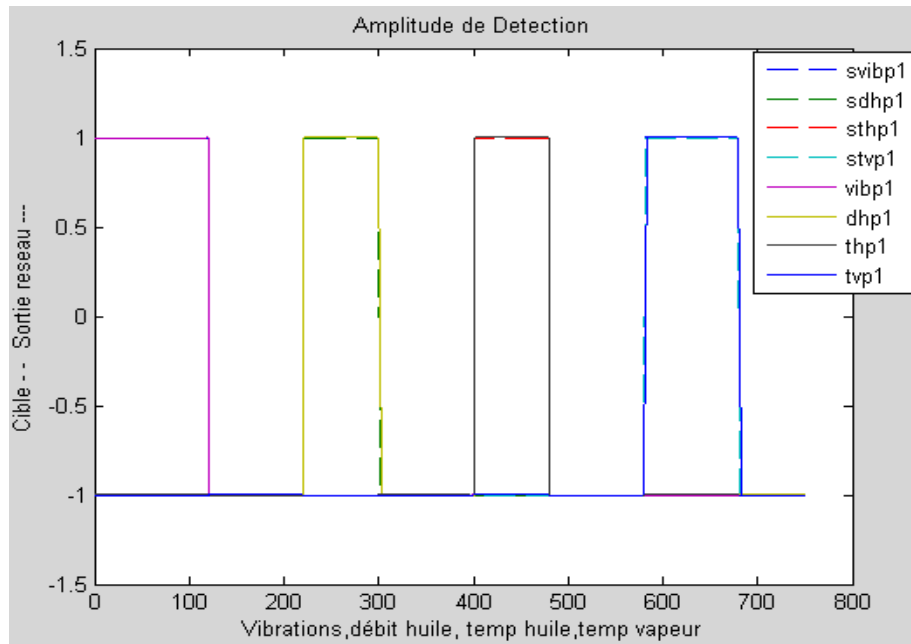


Figure 4.10 : Simulation des sorties du réseau

### IV-3.2.3 Évaluation des performances

Dans la figure 4.11: on voit que le coefficient de corrélation « R » tend vers « 1 » ce qui montre la performance du réseau. Voir figure 4.11.

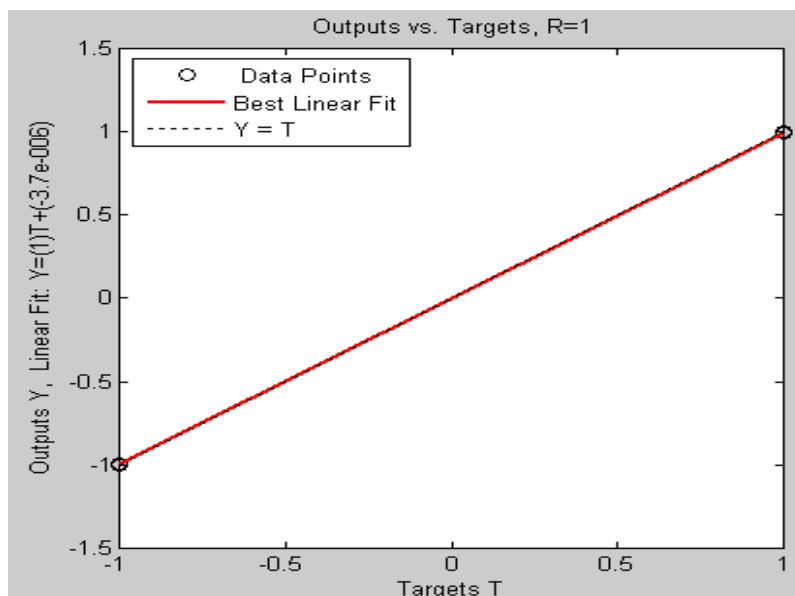


Figure 4.11 : Performances du réseau trouvé

### IV-3.2.4 Généralisation du réseau

Pour la généralisation du réseau nous avons augmenté l'amplitude des valeurs d'entrée du réseau, tel que nous avons multiplié le vecteur d'entrée par « 1.001 », donc nous remarquons, presque, une coïncidence totale entre les graphes des sorties du réseau et ceux des sorties désirées. Les figures (4.12 et 4.13) représentent respectivement, la généralisation et les performances du réseau trouvé.

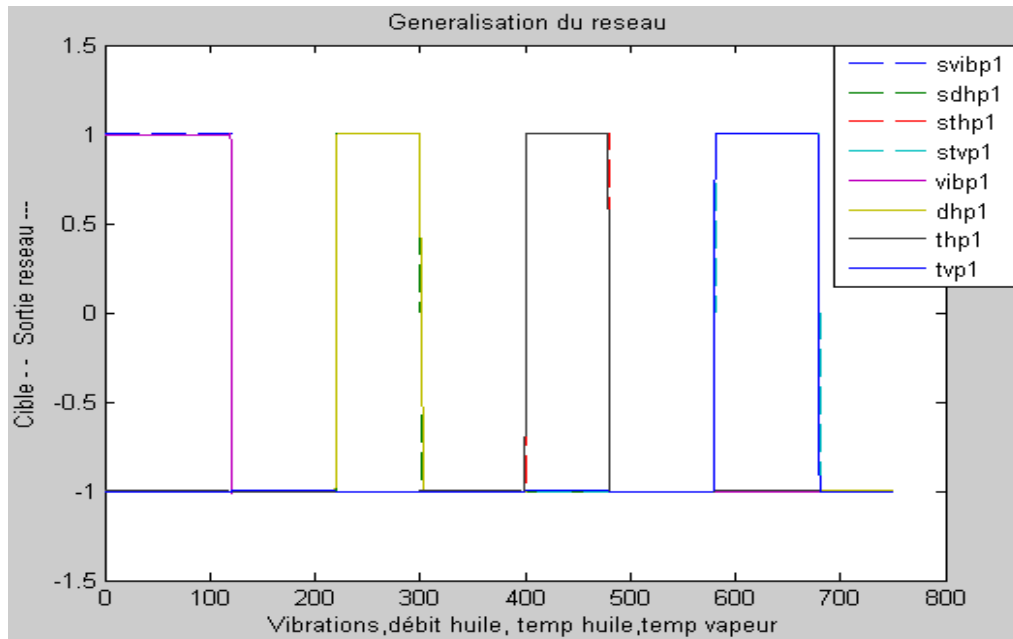


Figure 4.12 : Généralisation du réseau trouvé

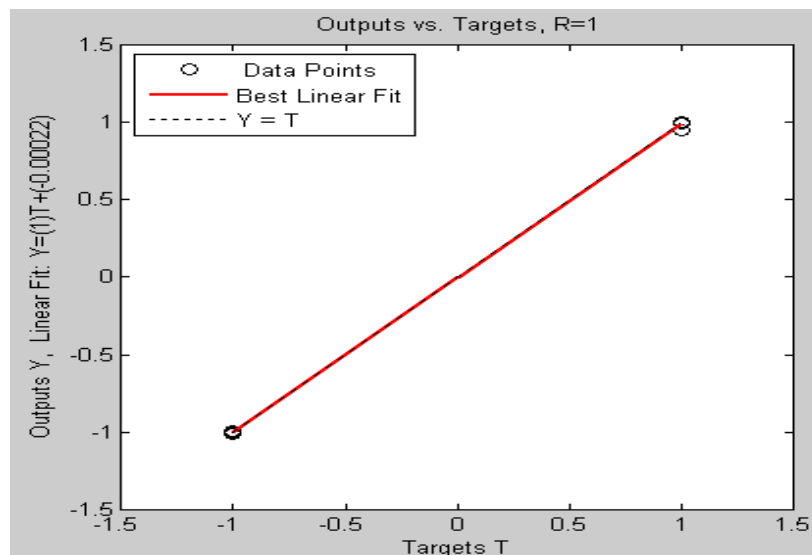


Figure 4.13 : Performances du réseau généralisé

#### IV-3.2.5 Exemples de test pour l'identification des causes de défaillances.

Dans tous les exemples qu'on va citer, la défaillance majeure c'est le fonctionnement anormal de la machine ou l'arrêt total de cette dernière.

**Exemple 1 :** on donne des valeurs élevées de vibrations et de température d'huile de lubrification palier 1 ( $vibp1=200 \mu m$ ,  $thp1=95 C^\circ$ ) et des valeurs normales de débit d'huile et de température vapeur ( $dhp1= 65.10^{-5} m^3/s$ ,  $tvp1=550 C^\circ$ ). Le réseau indique la cause probable de cette défaillance c'est la perte de caractéristiques d'huile de lubrification qui porte le code « **1010** », voir Figure 4.14. Le décodage de ce code « **1010** » est comme suite :

Tout d'abord Les paramètres surveillés sont ordonnés de la gauche vers la droite de cette manière : vibrations relatives, débit d'huile, température d'huile et température de vapeur. Alors, les vibrations portent le code « 1 » qui veut dire augmentation, le débit d'huile porte le code « 0 » qui veut dire rien ne se passe, la température d'huile porte le code « 1 » qui veut dire augmentation, la température de vapeur porte le code « 0 » qui veut dire rien ne se passe.

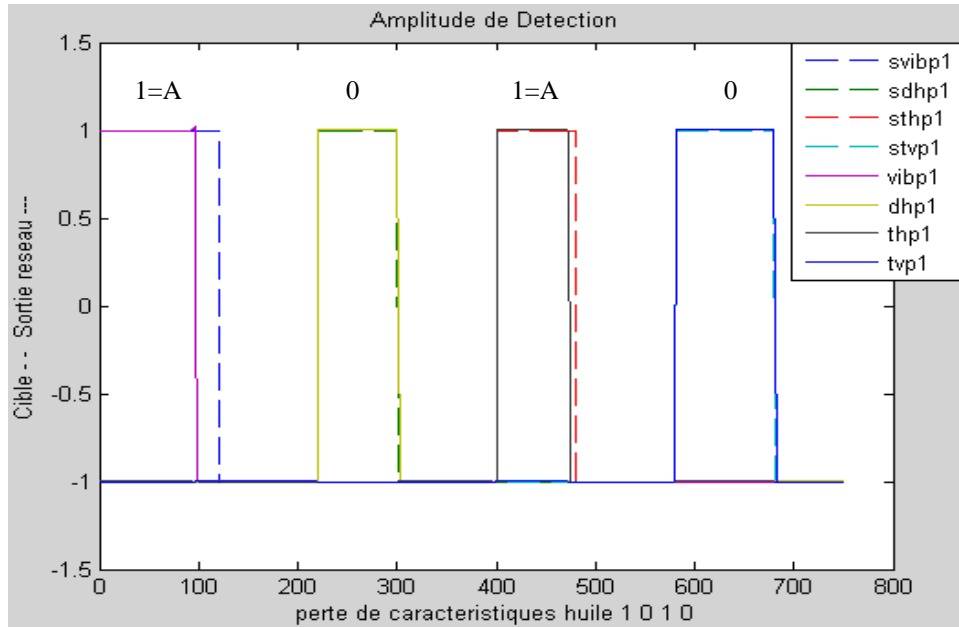


Figure 4.14 : Perte de caractéristiques d'huile

**Exemple 2 :** on injecte des valeurs élevées de vibrations et de température d'huile ( $vibp1=170 \mu m$ ,  $thp1= 100 C^\circ$ ), une valeur basse de débit ( $dhp1= 30 \cdot 10^{-5} m^3$ ) et une valeur normale de température de vapeur ( $thp1= 580 C^\circ$ ).

Le réseau signale la cause de la défaillance c'est que la vanne d'huile est trop fermée représentée par le code « +1 -1 1 0 », voir Figure 4.15.

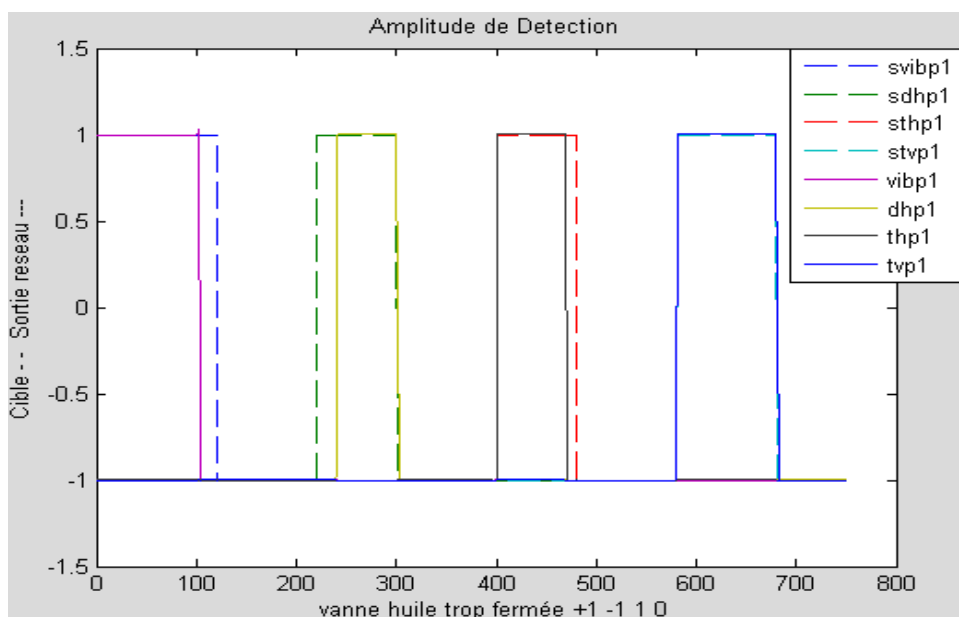


Figure 4.15 : Vanne trop fermée

**Exemple 3 :** nous avons donné des valeurs élevées de vibrations et de débit d'huile ( $vibp1=140 \mu m$ ,  $dhp1= 150 C^\circ$ ), une valeur basse de température d'huile ( $thp1= 30 C^\circ$ ) et une valeur normale de température de vapeur ( $thp1= 550 C^\circ$ ).

Le réseau signale la cause de la défaillance c'est que la vanne d'huile est trop ouverte représentée par le code « **1 1 -1 0** », voir Figure 4.16.

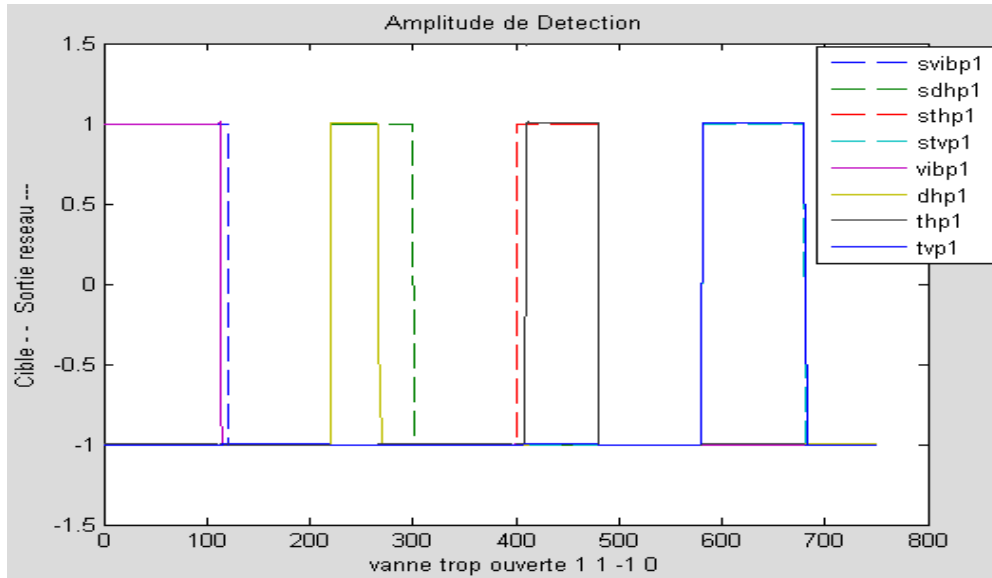


Figure 4.16: Vanne trop ouverte

**Exemple 4 :** nous avons testé cette fois-ci par des valeurs élevées de vibrations et de température d'huile ( $vibp1=190 \mu m$ ,  $thp1= 110 C^\circ$ ), une valeur nulle de débit ( $dhp1= 0.00 m^3$ ) et une valeur normale de température de vapeur ( $thp1= 530 C^\circ$ ).

Le réseau signale la cause de la défaillance c'est que la pompe d'huile est défaillante représentée par le code « **+1 --1 +1 0** », voir Figure 4.17.

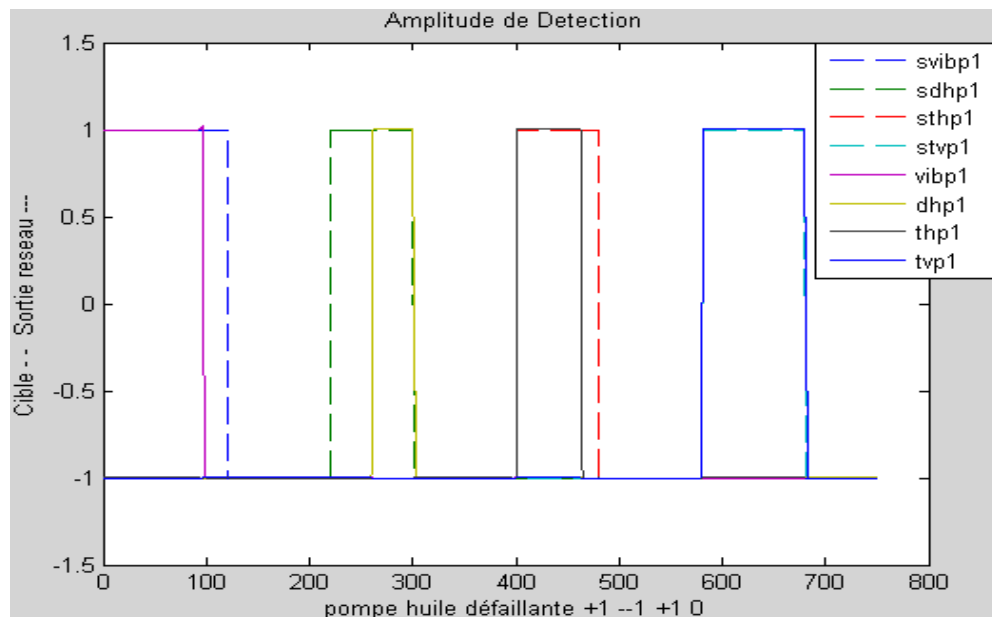


Figure 4.17 : Pompe d'huile défaillante

**Exemple 5 :** On donne des valeurs élevées de vibrations ( $vibp1=250 \mu m$ ), des valeurs normales de débit d'huile et de température d'huile ( $dhp1= 70.10^{-5} m^3/s$ ,  $thp1=60 C^\circ$ ) et des valeurs basses pour la température de vapeur ( $tvvp1=400 C^\circ$ ). Le réseau indique la cause probable de cette défaillance « **1 0 0 -1** » se trouve exactement au niveau de la chaudière ou dans le surchauffeur, voir Figure 4.18.

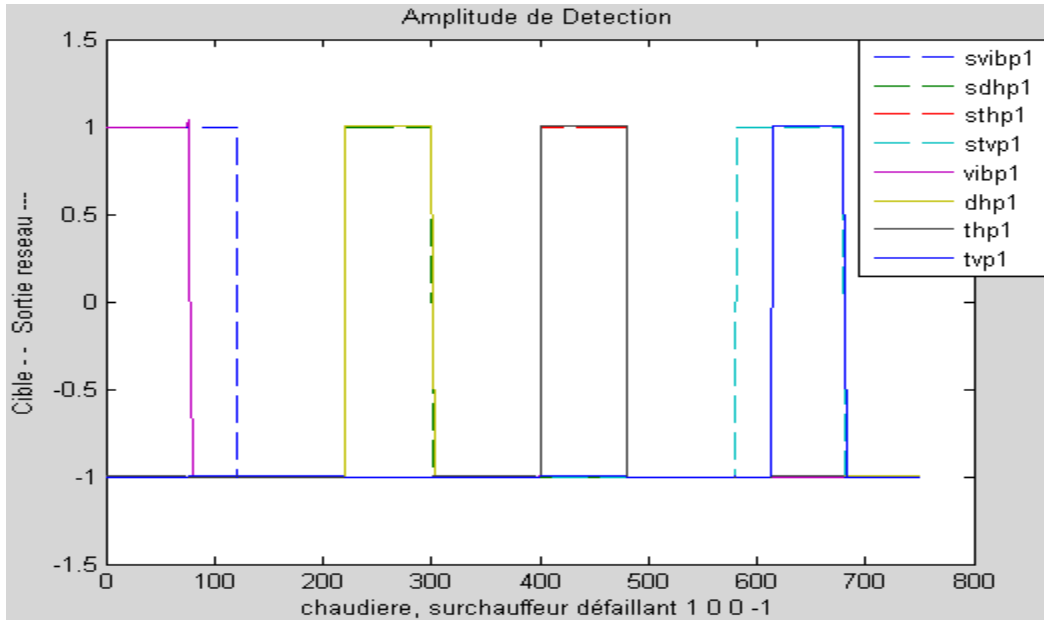


Figure 4.18 : Chaudière ou surchauffeur défaillant

#### IV-4 DIAGNOSTIC DE LA DEUXIEME SERIE DE CAUSES

##### IV-4.1 Méthodologie du travail

Le reste des causes, de 1 à 7 (bande jaune dans le tableau 4.1) correspondantes aux vibrations, à la vitesse de rotation et à la pression de vapeur, sera identifié par un deuxième réseau de neurone à trois sorties.

L'erreur limite donnée au réseau est de  $10^{-9}$ , le nombre d'itérations limite est **1000** et le pas de calcul c'est **0.001**.

Pour obtenir l'architecture optimale nous avons varié le nombre de neurone de la couche cachée de **1** jusqu'à **15**, donc l'architecture optimale trouvée est : « **601-06-03** » qui représente la plus petite erreur **5.49629e-010**, voir le tableau 4.3.

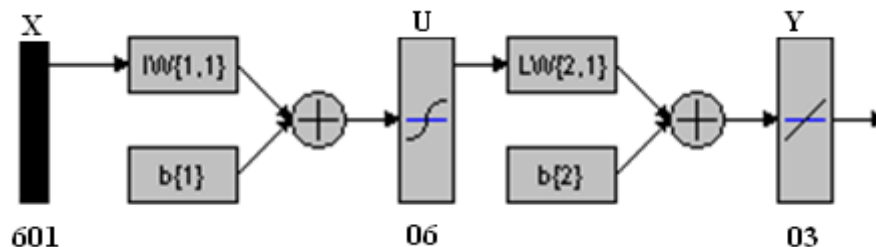


Figure 4.19 : Architecture optimale trouvée

Architecture	Tap(s)	Itérations	R	RG	MSE
601-01-03	31.59	41	0.9988	0.9850	0.399581
601-02-03	103.96	285	0.8629	0.8656	0.244137
601-03-03	298.87	1000	0.9995	0.9940	0.161528
601-04-03	31.19	66	1.0000	0.9948	0.0677959
601-05-03	50.83	79	1.0000	0.9948	9.8108e-010
<b>601-06-03</b>	<b>46.83</b>	<b>74</b>	<b>1.0000</b>	<b>0.9948</b>	<b>5.49629e-010</b>
601-07-03	90.11	187	1.0000	0.9948	9.19899e-010
601-08-03	33.40	54	1.0000	0.9948	8.71338e-010
601-09-03	90.45	166	1.0000	0.9948	6.57351e-010
601-10-03	685.60	1000	1.0000	0.9948	4.04554e-008
601-11-03	106.22	141	1.0000	0.9948	9.49359e-010
601-12-03	223.66	293	1.0000	0.9947	7.70922e-010
601-13-03	770.10	1000	0.9997	0.9943	0.000584656
601-14-03	3170.19	401	1.0000	0.9946	6.35073e-010
601-15-03	753.46	928	1.0000	0.9947	8.90926e-010

Tableau 4.3 : variation de l'erreur quadratique moyenne en fonction de neurones de la couche cachée pour le réseau « **601-06-03** ».

#### IV-4.2 Explication des résultats

##### IV-4.2.1 Apprentissage du réseau

Le graphe d'apprentissage converge vers l'erreur fixée d'ordre  $10^{-9}$  après **74** itérations dans une durée de **46.83 s**, voir Figure 4.20.

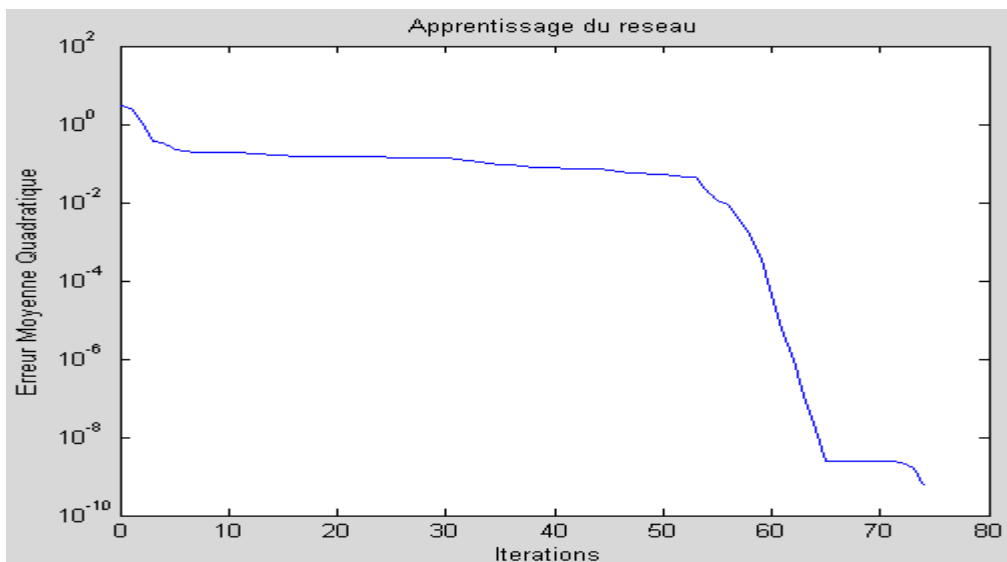


Figure 4.20: Apprentissage du réseau de neurone « **601-06-03** ».

#### IV-4.2.2 Simulation des sorties du réseau

On voit que les graphes en ligne continue du réseau sont bien coincés avec les graphes en ligne discontinue des fonctions désirées, ce qui explique la bonne simulation de l'architecture trouvée, voir Figure 4.21.

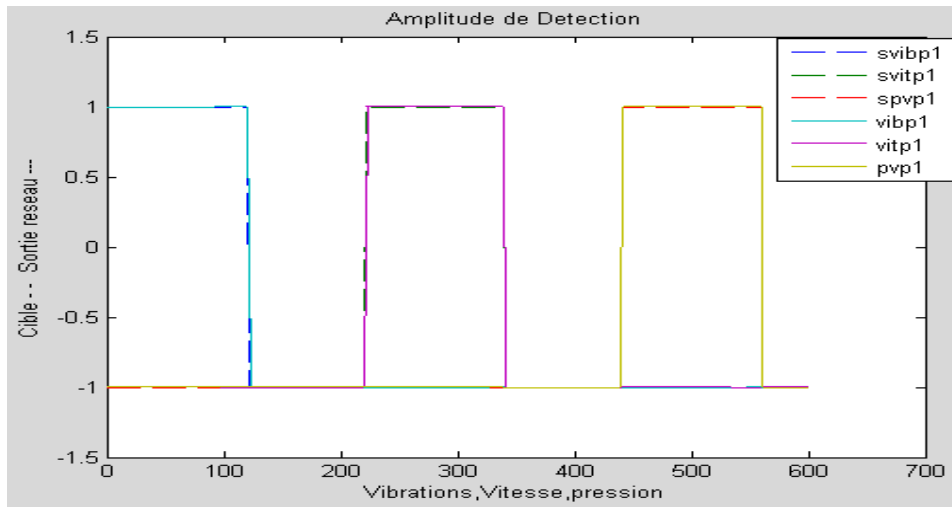


Figure 4.21: Simulation des sorties du réseau

#### IV-4.2.3 Évaluation des performances

Dans la figure 4.22 : on voit que le coefficient de corrélation « R » tend vers « 1 » ce qu'il montre la performance du réseau. Voir figure 4.22.

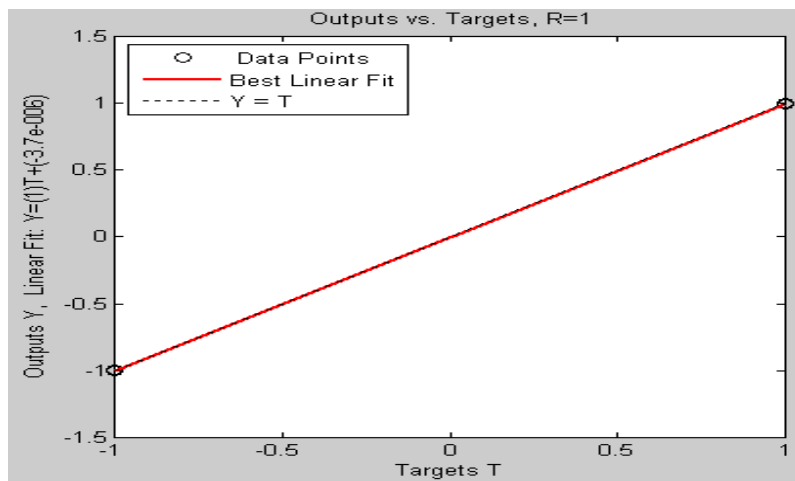


Figure 4.22 : Performances du réseau trouvé

#### IV-4.2.4 Généralisation du réseau

Pour la généralisation du réseau nous avons augmenté l'amplitude des valeurs d'entrée du réseau, tel que nous avons multiplié le vecteur d'entrée par « 1.001 », donc nous remarquons, presque, une coïncidence totale entre les graphes des sorties du réseau et ceux des sorties désirées. Les figures (4.23 et 4.24) représentent respectivement, la généralisation et les performances du réseau trouvé.

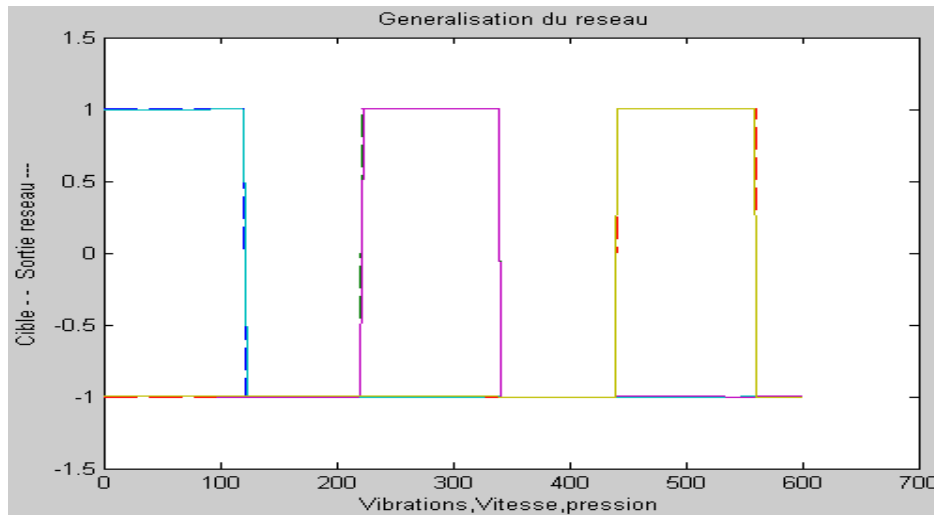


Figure 4.23 : Généralisation du réseau trouvé

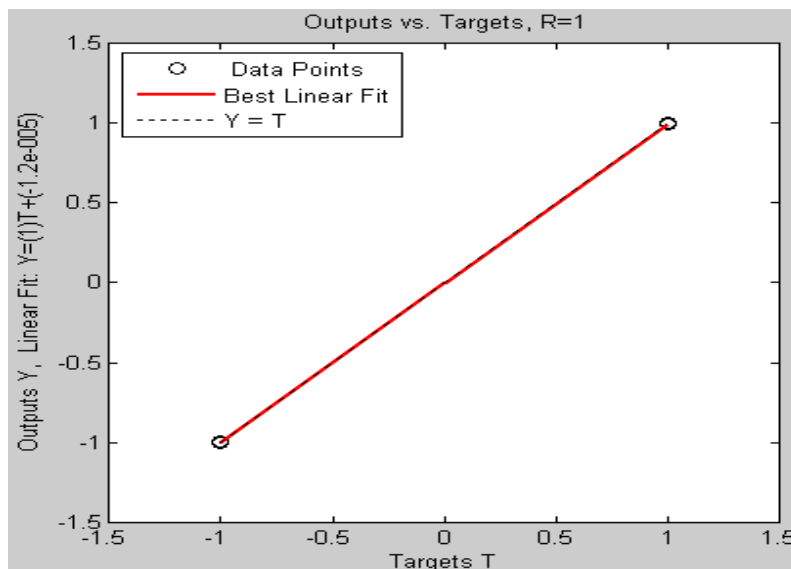


Figure 4.24 : Performances du réseau généralisé

#### IV-4.2.5 Exemples de test pour l'identification des causes de défaillances

Cette fois ci en combinant entre les trois grandeurs suivantes : les vibrations, la vitesse de rotation et la pression.

**Exemple 1 :** on injecte une valeur élevée de vibration ( $vibp1=180 \mu m$ ) et deux valeurs basses de vitesse et de pression vapeur ( $vitp1=2500 \text{ tr/min}$ ,  $pvp1=80 \text{ bar}$ ). Le réseau signale que la cause de ce problème c'est la rupture d'une ailette du corps HP de la turbine codée par « **1 -1 -1** ». Le décodage de ce code « **1 -1 -1** » est comme suite :

Tout d'abord, les paramètres surveillés sont ordonnés (figure 4.25) de la gauche vers la droite de cette manière : vibrations relatives, vitesse de rotation et pression de vapeur. Alors, les vibrations portent le code « 1 » qui veut dire augmentation, la vitesse de rotation porte le code « -1 » qui veut dire diminution, la pression de vapeur porte le code « -1 » qui veut dire diminution. Voir la figure 4.25.

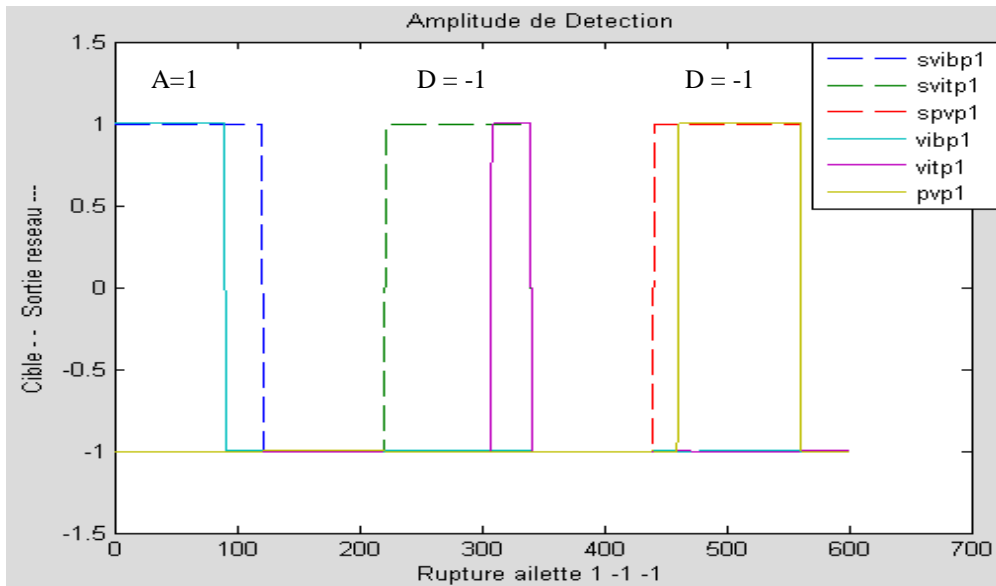


Figure 4.25 : Rupture d'ailette

**Exemple 2:** On injecte des valeurs élevées de vibration et de pression vapeur (vibp1=160  $\mu\text{m}$ , pvp1=180 bar) et une valeur basse de vitesse de rotation (vitp1=2700 tr/min).

Le réseau signale que la cause de ce problème c'est la corrosion des ailettes du corps HP de la turbine dans ce cas. Voir la figure 4.26.

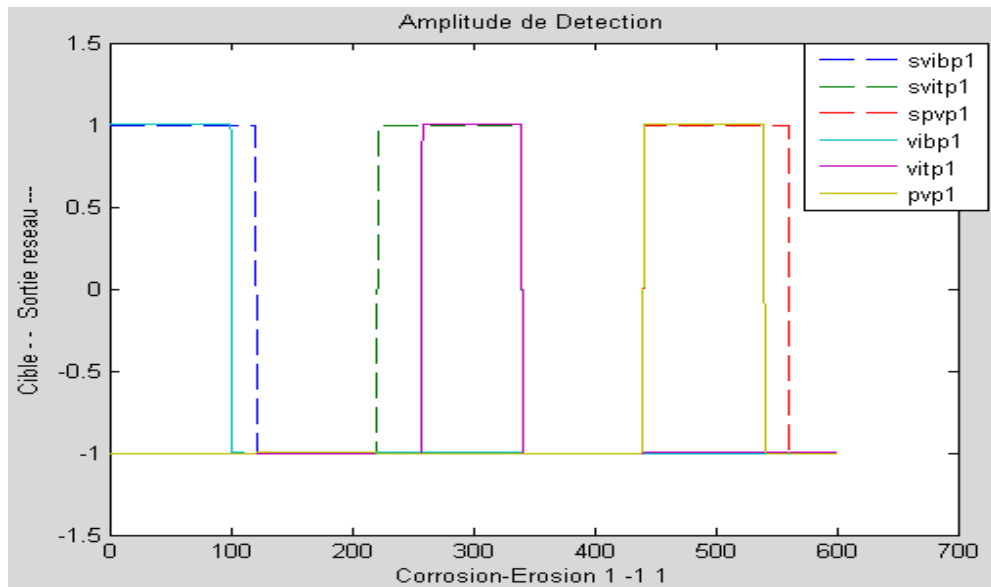


Figure 4.26 : Corrosion et érosion d'ailette

**Exemple 3:** On reçoit d'après les capteurs de surveillance des valeurs acceptables de vibrations (vibp1=80  $\mu\text{m}$ ) et des valeurs inférieures de vitesse et de pression (vitp1=2500 tr/min, pvp1=70 bar).ces données seront injectées dans le réseau, ce dernier nous a déclaré la cause probable qui génère ce type de défaillance.

La cause signalée c'est que la conduite d'admission de vapeur est coupée ou bouchée, codée par « 0 --1 --1 », Voir la figure 4.27.

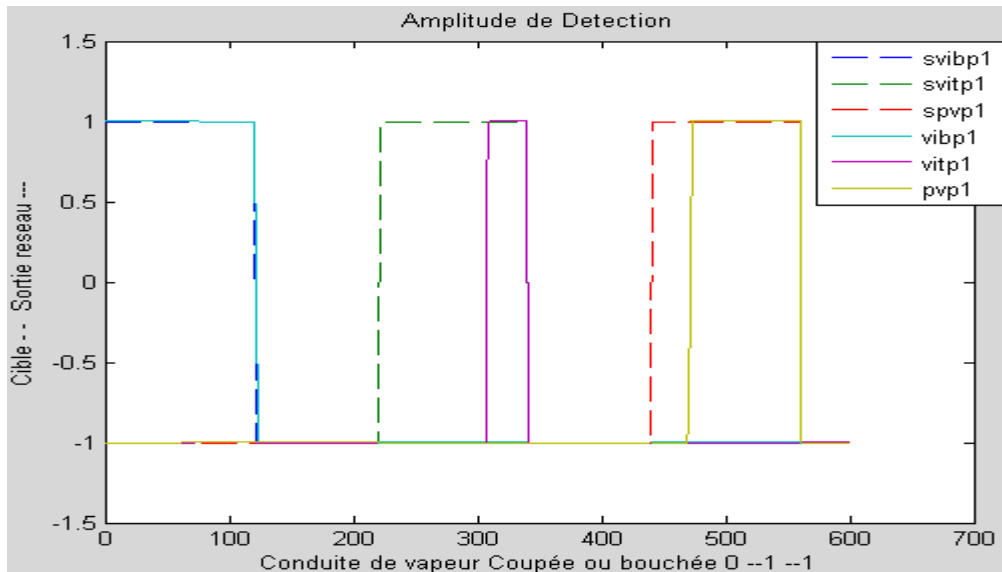


Figure 4.27 : Conduite coupée

**Exemple 4:** dans cet exemple on va tester notre réseau par des valeurs acceptables de vibrations ( $vibp1=80 \mu m$ ) et des valeurs élevées de vitesse et de pression ( $vitp1=3200 \text{ tr/min}$ ,  $pvp1=190 \text{ bar}$ ). Le réseau indique que la vanne de vapeur est trop ouverte c'est la cause « **0 1 1** » pour laquelle la turbine male fonctionner (survitesse). Voir la figure 4.28.

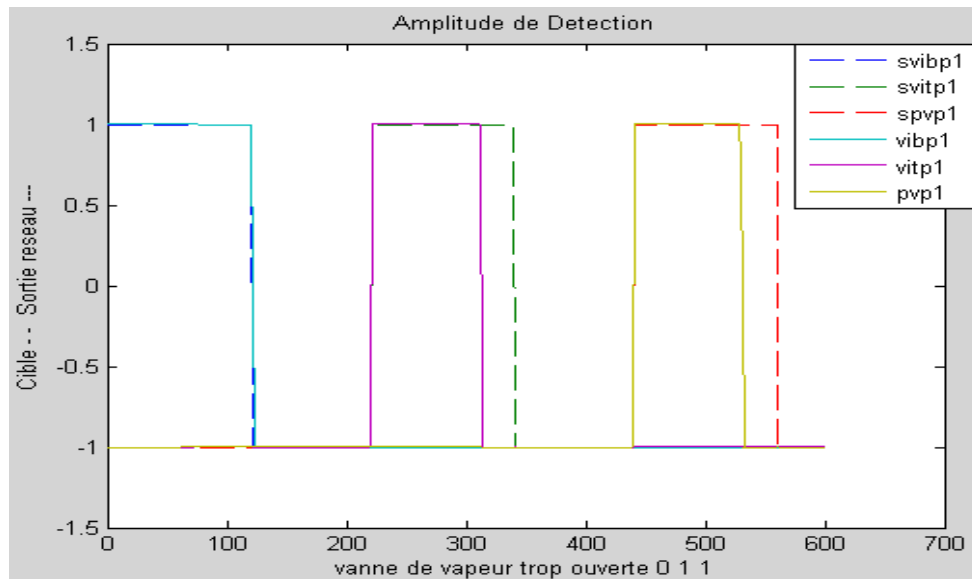


Figure 4.28 : Vanne de vapeur trop ouverte

**Exemple 5:** Dans cet exemple on va tester notre réseau par des valeurs élevées de vibrations ( $vibp1=160 \mu m$ ) et des valeurs acceptables de vitesse et de pression ( $vitp1=3000 \text{ tr/min}$ ,  $pvp1=140 \text{ bar}$ ). Le réseau indique qu'il ya un jeu, desserrage ou mauvaise fixation « **1 0 0** » c'est la cause pour laquelle la turbine male fonctionner (vibrations, bruit anormal). Voir la figure 4.29.

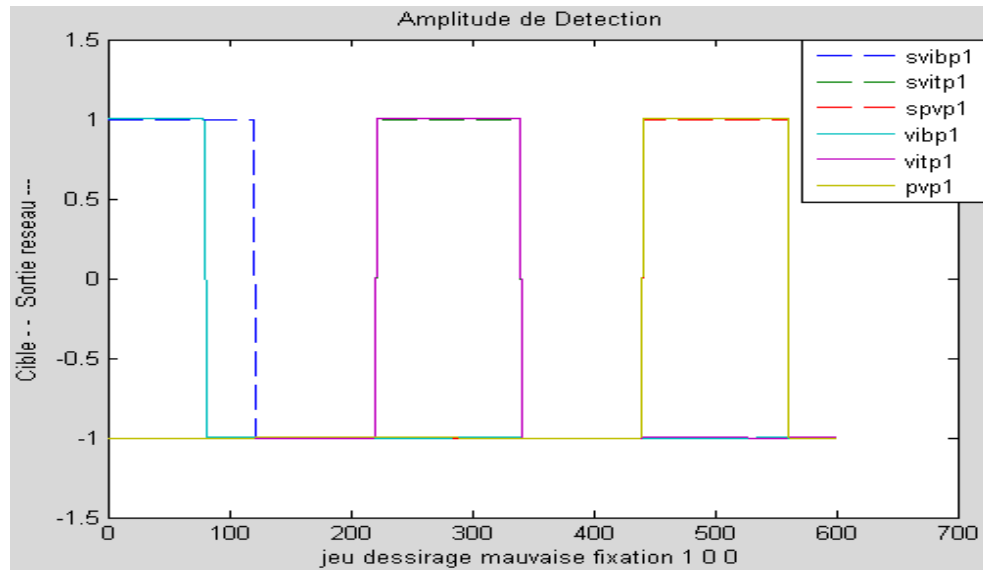


Figure 4.29 : Jeu, desserrage ou mauvaise fixation

## CONCLUSION

Un tel système neuronal de diagnostic, basé sur les données obtenues d'après les méthodes descriptives telles que les arbres de défaillances et les méthodes experts, permet de construire une méthode hybride efficace et rapide, a la possibilité d'automatiser la tâche de diagnostic. Parmi les caractéristiques essentielles de cette méthode hybride nous désignons :

- La rapidité d'identification des causes probables de la défaillance à cause de l'apprentissage fiable et rapide des réseaux de neurones, cette propriété (rapidité d'identification des causes) est un grand avantage qui facilite aux spécialistes de maintenance de trouver l'origine de la défaillance et de réparer cette dernière le plus rapidement possible, tout ça s'ajoute dans le but de minimisation des coûts de maintenance d'un coté et d'augmenter le taux de production d'un autre coté.
- L'exploitation des données de capteurs de surveillance de paramètres de production (pression vapeur, température vapeur, débit d'huile ...) pour le diagnostic de défaillances est un avantage participant pour éviter la redondance matériel et l'utilisation très large des capteurs.
- Une méthode qui dépend toujours de l'exactitude des données reçues par le réseau de neurone soit les données capteurs soit les données des méthodes descriptives. Une telle donnée fautive va automatiquement influencer sur la fiabilité du système proposé.
- La réalisation d'un système fiable de diagnostic basé sur la méthode combinée exige toujours des gens qualifiés dans tous les domaines techniques pour rassembler le maximum d'informations concernant l'équipement à diagnostiquer.
- Parfois une seule combinaison mène à plusieurs causes, dans ce cas on doit ajouter d'autres paramètres tel que le temps, prenons comme exemple le cas d'une (conduite bouchée) et le cas du (joint d'étanchéité dégradé) ces deux causes appartiennent à la même combinaison « 0 -1 -1 », dans ce cas nous devons entrer le paramètre du temps par exemple. La combinaison se change et elle va devenir « 0 -1 -1 » pour la dégradation du joint d'étanchéité et « 0 --1 --1 » pour la fuite au niveau de la conduite, le double signe « -- » veut dire diminué rapidement ; puisque en générale la fuite à cause d'une coupure se fait rapidement par contre le joint d'étanchéité prend beaucoup de temps pour qu'il soit dégradé.

## **CONCLUSION GENERALE ET PERSPECTIVES**

## CONCLUSION GENERALE ET PERSPECTIVES

Le travail de recherche présenté dans ce mémoire porte sur l'étude des réseaux de neurones artificiels pour la surveillance des systèmes de production industriels. Dans le cadre de la surveillance, notre étude concerne la détection de défaillances et le diagnostic industriel.

Les deux architectures neuronales les plus utilisées en surveillance sont le Perceptron Multi Couches (MLP) avec sa représentation globale de son espace de données et les Réseaux à Fonctions de base Radiales (RBF) avec une représentation plutôt locale de cet espace. Le choix de L'architecture MLP pour tous les systèmes de surveillance traités revient à sa grande utilisation dans le domaine de surveillance (détection) d'un coté, et d'un autre coté, à sa rapidité et sa capacité d'apprentissage par l'utilisation de l'algorithme de Levenberg-Marquardt. Le premier inconvénient des réseaux MLP c'est qu'ils ne peuvent pas traiter la deuxième partie du diagnostic (identification des causes des défaillances). Donc, il est nécessaire d'enrichir la méthode MLP par une autre méthode, de type différent, représentée par les arbres de défaillance (ADD) qui sont à base de modélisation fonctionnelle et structurelle.

Les principales contributions de ce mémoire sont regroupées en deux parties. Une première partie regroupe l'état de l'art autour de deux points d'intérêts en étroite corrélation :

- Différentes méthodologies de surveillance des systèmes de production (Chapitre I).
- Application des réseaux de neurones à la surveillance industrielle (Chapitre II).

La deuxième partie de notre travail, articulée autour des deux chapitres suivants, synthétise l'essentiel de notre contribution scientifique. Dans le troisième chapitre de ce mémoire, les réseaux MLP ont été utilisés pour détecter et localiser des défaillances sur une turbine à vapeur. Dans le dernier chapitre du mémoire (Chapitre IV), une procédure pour construire un modèle de diagnostic pour des systèmes industriels complexes (turbine à vapeur) a été proposée. Cette procédure repose sur le couplage de deux outils : les ADDs et les MLPs.

Les premiers (ADDs) sont utilisés pour modéliser les causes probables des différentes défaillances que peut subir le système. L'utilisation de cet outil a permis de modéliser la connaissance fournie par un expert pour faire du diagnostic. Les seconds (MLPs) sont utilisés pour la classification rapide et fiable, au temps réel, des causes probables de toutes les défaillances survenant sur le système.

Les principaux résultats obtenus et les différentes remarques proposées, lors de la simulation, sont résumés dans les points suivants :

- La difficulté d'apprentissage du réseau (temps d'apprentissage plus grand) lors de l'augmentation du vecteur de données (nombre de neurones de la première couche). Par contre le nombre de neurones de la couche cachée n'a pas une règle fixe. Le nombre de neurones de la dernière couche a été choisi selon le besoin ; soit dans la détection, soit dans le diagnostic. Et chaque fois on augmente le nombre de neurones de cette dernière couche, la difficulté d'apprentissage sera d'autant plus grande. Parfois la phase d'apprentissage se passe bien et se converge vers l'erreur fixée mais ça sera l'inverse pour la phase de test, dans ce cas ça pose un problème (le cas de la température de vapeur et la température d'huile de sorties paliers).
- Le choix des poids d'initialisation et des fonctions d'activation ne doit pas être négligé. Les poids initiaux jouent un rôle très important dans le raffinement du réseau, la sélection de ceux-ci dépend entièrement du concepteur. Par contre, l'utilisation des

fonctions d'activation de type tangente hyperbolique dans les couches cachées et celles de type linéaire dans la couche de sortie est fortement souhaitée dans notre cas.

- Le choix de l'architecture optimale du réseau est basé généralement sur la valeur de l'erreur quadratique moyenne (MSE), donc une faible erreur d'apprentissage implique une bonne architecture. Parfois on reçoit des mauvais résultats dans la phase de test, ce qui nous oblige de choisir une erreur supérieure pour avoir l'architecture optimale. Il faut tenir en compte aussi le nombre d'itérations et le temps d'apprentissage pour le choix d'une telle architecture, car un nombre important d'itérations peut bloquer le calculateur et un temps d'apprentissage plus grand est déconseillé dans le domaine de la surveillance ; surtout pour les industries qui nécessitent une grande sécurité (nucléaire, chimie-pharmaceutique...).
- La réalisation de l'arbre de défaillance de chaque événement traité nécessite un bon bagage théorique, des idées d'experts et une documentation technique très riche. Dans la réalité, le nombre de causes des défaillances dans les turbines est strictement grand, mais nous avons limité ce nombre, dans notre étude, basant sur les causes principales. Le but est de montrer la méthodologie et le système de diagnostic proposé. Parmi les spécifications principales de ce système, c'est que ce dernier peut réagir directement avec les valeurs numériques obtenues à partir des capteurs de surveillance (surveillance au temps réel).
- Le nouveau système de diagnostic permet aux opérateurs relativement non qualifiés de prendre des décisions fiables sans connaître le mécanisme du système et analyser les données. Toutes les opérations de diagnostic se passent à l'intérieur du programme, sous forme d'une boîte noire, et l'opérateur à la salle de commande n'a aucune grande tâche à faire, il doit seulement lire les instructions sur l'écran.
- La méthode hybride proposée permet d'automatiser la tâche de diagnostic et de remplacer l'expert humain dans sa mission, pour rassembler toutes les informations lui permettant d'apprendre sa décision.

En perspective, il est intéressant :

- De faire une étude de comparaison entre le Perceptron Multi Couches (PMC) et les Réseaux à Fonctions de base Radiales (*RFR*), l'erreur d'apprentissage, le temps d'apprentissage et de test, l'efficacité de chacun pour la surveillance en ligne des systèmes de production.
- De construire un réseau qui peut traiter tous les événements (variables) à la fois, tenant en compte la rapidité de la localisation de défaillances et de l'identification des causes.
- D'étendre l'étude pour d'autres types et architectures de réseaux de neurones, entre autres les réseaux feed-forward avec rétroaction (feed-back) et les mémoires auto-associatives,
- D'étendre l'étude pour d'autres outils de surveillance, tels que les réseaux de Pétri, les réseaux baysiens, réseaux Pétri-Floues.

## Références Bibliographiques

- [AGH 1998] **Aghasaryan, A.**, (1998). « *Formalisme HMM pour les réseaux de Pétri partiellement stochastiques : Application au diagnostic de pannes dans les systèmes répartis* », Thèse de doctorat université de Rennes1.
- [ANG 1994] **Anglano, C., L. Portinale**, (1994). « *B-W Analysis: a Backward Reachability Analysis for Diagnostic Problem Solving Suitable to Parallele Implementation* », Proceeding of the 15th International Conference on Application and Theory of Petri Nets, Zaragoza, Espagne.
- [BAS 1996], **Basseville, M. et M-O. CORDIER** (1996). « *Surveillance et diagnostic de systèmes dynamiques: approche complémentaire du traitement de signal et de l'intelligence artificielle* », *Rapport INRIA*, N°2861.
- [BEL 2008], **Belkacem OULD BOUAMAMA**, (2008) « *Surveillance des systèmes de production* », cours de Master AGII Laboratoire d'Automatique, Génie Informatique et Signal (LAGIS - UMR CNRS 8021), Ecole Polytechnique de Lille (poltech-lille.fr).
- [BOU 1994], **Bouchon-Meunier B.**, (1994). « *La logique floue. Collection Que sais-je* », n.2702, Ed. Presses Universitaires de France, Paris.
- [BOU 1995], **Bouchon-Meunier B.**, (1995). « *La Logique Floue et ses applications* », Edition Addison- Wesley.
- [CLA 1992], **Claude TOUZET** (juillet 1992), « *Les réseaux de neurones artificiels introduction au Connexionnisme cours, exercices et travaux pratiques* ».
- [COM 1991], **Combacau M.**, (1991). « *Commande et Surveillance des Systèmes à Évènements Discrets Complexes : Application aux Ateliers Flexibles* », Thèse de doctorat, Université Paul Sabatier, Toulouse.
- [COB 2000], **Combastel C.**, (2000). « *Méthodes d'aide à la décision pour la détection et la localisation de défauts dans les entraînements électriques* », Thèse de Doctorat INPG, Grenoble.
- [CUS 1996], **Cussenot C.**, (1996). « *Surveillance et diagnostic de la chaîne de dépollution d'une automobile* », Thèse de doctorat, Université de Rennes 1.
- [DAN 2006] **Daniel RACOCEANU**, (2006) « *Contribution à la surveillance des Systèmes de Production en utilisant les Techniques de l'Intelligence Artificielle* » Synthèse des activités de recherche et d'encadrement. Université de FRANCHE-COMTÉ de Besançon, France.
- [DAS 2000], **Dash S. et V. Venkatasubramanian** (2000). « *Challenges in the industrial applications of fault diagnostic systems* », Proceedings of the Conference on Process Systems Engineering Computing & Chemical Engineering, Keystone, Colorado, pp. 785-791.

[DEE 2006], **Deepak MISHRA, Abhishek YADAV, Sudipta RAY, and Prem K. KALRA** (July 2006), « *Levenberg-Marquardt Learning Algorithm for Integrate-and-Fire Neuron Model* », Department of Electrical Engineering Indian Institute of Technology, Kanpur, India. Neural Information Processing - Letters and Reviews Vol.9, No.2.

[DEN 1998], **Denoeux T., M. Masson et B. Dubuisson** (1998). « *Advanced pattern recognition techniques for system monitoring and diagnosis: a survey* », Journal Européen des Systèmes Automatisés (RAIRO-APII-JESA), 31(9-10), pp.1509-1539.

[DEV 1991], **Devauchelle G.**, (1991). « *Diagnostic mécanique des fatigues sur les structures soumises à des vibrations en ambiance de travail* », Thèse de Doctorat, Université de Paris IX Dauphine.

[DOC 1984], Documentation du constructeur de la centrale thermique de Cap-Djinet, **KraftWerk Union AG**.

[DUB 2001], **Dubuisson B., E. Boutleux, P. Dague, T. Denoeux, E. Didelet, Y. Gandvalet et M. Masson** (2001). « *Diagnostic, Intelligence Artificielle et reconnaissance de formes* », Ed. Hermes.

[DUB 1990], **Dubuisson B.**, (1990). « *Diagnostic et reconnaissance des formes* », Paris, Edition Hermès.

[EVS 1998], **Evsukoff A.**, (1998). « *Raisonnement approché pour la surveillance des procédés* », Thèse de Doctorat, INPG, Grenoble.

[FRA 1990], **Frank P.M.**, (1990). « *Fault Diagnosis in Dynamic Systems Using Analytical and Knowledge Based Redundancy – A survey and New Results* », Automatica, Vol. 26, pp. 459-474.

[GIL 1995] **Gilles ZWINGELSTEIN**, (1995) « *diagnostic des défaillances, théorie et pratique pour les systèmes industriels* » Editions Hermès 14 rue Lantiez 75017 Paris.

[GUI 2005], **Guillaume COMTE** (juin 2005) « *L'arbre des défaillances* » Institut National de l'Environnement Industriel et des Risques, France.

[GRO 2000], **Grosclaude I. et R. QUINIOU** (2000). « *Dealing with Interacting Faults in Temporal Abductive Diagnosis* », Proc. of the 8th International Workshop on Principles of Diagnosis.

[GRO 2001], **Grosclaude I.**, (2001). « *Diagnostic Abductif Temporel : Scenarios de Pannes, modèles causaux et traitement de l'interaction* », Thèse de Doctorat, Université de Rennes 1.

[HEN 2002], **Henri-Pierre RAMELLA** (Janv.2002), « *Maintenance des turbines à vapeur* », réf. BM4186 – Techniques de l'ingénieur.

[HIC 2007], **Hicham CHATTOU, Vathana SITTHISACK, Erwan CUINET** (2007) « *Analyse préliminaire des risques; les arbres de défaillances* » ISTIA, université d'ANGERS.

[HOW 2002], **Howard DEMUTH, Mark BEALE**, (July 2002) « *Neural Network Toolbox; For Use with MATLAB* », The Math Works, Inc. 3 Apple Hill Drive Natick, MA 01760-2098.

[JAC 2001], **Jacques MOREL** (Mars 2001), « *Surveillance vibratoire et maintenance prédictive* » École nationale supérieure de mécanique (ENSM), R 6100– Techniques de l'ingénieur.

[KAM 2008] **Kamal MEDJAHHER, Amine MECHRAOUI, Noureddine ZERHOUNI** Article (2008), « *Diagnostic et pronostic de défaillances par réseaux bayésiens* ». Institut FEMTO-ST, UMR CNRS 6174 - UFC / ENSMM / UTBM, GIPSA-lab., CNRS UMR 5216, ENSIEG, France.

[KAR 2007] **Jawad KARIM**, (2007) « *Surveillance, Diagnostic et Pronostic en Temps Réel de Systèmes Hybrides* » : Application à des Bancs d'Essais CERTIA. LAAS-CNRS Groupe DISCO 7, avenue du Colonel Roche 31077 Toulouse, France Article 2007.

[LEF 2000], **Lefebvre D.**, (2000). « *Contribution à la modélisation des systèmes dynamiques à événements discrets pour la commande et la surveillance* », Habilitation à Diriger des Recherches, Université de Franche Comté/ IUT Belfort - Montbéliard.

[MAL 1996], **Malek M.** (1996). « *Un Modèle Hybride de Mémoire pour le Raisonnement à Partir de Cas* », Thèse de doctorat de l'Université Joseph Fourier Grenoble1.

[MAN 2005], **Manolis I. A. LOURAKIS** (February, 2005), « *A Brief Description of the Levenberg-Marquardt Algorithm* », Implemened Institute of Computer Science Foundation for Research and Technology - Hellas (FORTH) Vassilika Vouton, P.O. Box 1385, GR 711 10 Heraklion, Crete, GREECE.

[MOH 2004] **Mohamed Ryad ZEMOURI**, (2004) « *Contribution à la surveillance des systèmes de production à l'aide des réseaux de neurones dynamiques : Application à l'e-maintenance* ». Thèse de doctorat 2004. L'UFR des Sciences et Techniques de l'Université de Franche-Comté.

[MON 2004] **Monnin M.**, (2004). « *Diagnostic et pronostic industriel en utilisant les techniques de l'intelligence artificielle* ». Application a la plate-forme de e-maintenance du LAB, DEA IAP de l'Université de Franche-Comté, Besançon.

[NED 2008] **Nedjma SALHI, Djamel BENAZZOUZ**, (2008) « *Surveillance et Diagnostic d'une chaîne de production par les réseaux de neurone artificiels* ». Laboratoire LM2S - Université M'hamed Bougara Boumerdes, 35000, Algérie. Mémoire de magister.

[NIC 2005] **Nicolas PALLUAT, Daniel RACOCEANU, Noureddine ZERHOUNI** « *Utilisation des réseaux de neurones temporels pour le pronostic et la surveillance dynamique* ». Etude comparative de trois réseaux de neurones récurrents. RSTI - RIA. Volume 19 – n° 6/2005, pages 911 à 948. France Article 2005.

[OUD 2001], **Ouldyahia N. S.**, (2001). « *Diagnostic des systèmes de production en utilisant les réseaux de neurones* ». Introduction au neuro-flou, Mémoire DEA, Université de Franche-Comté, Besançon.

[OZG 2004], **Özgür KI I**, (2004) “*Multi-layer perceptrons with Levenberg-Marquardt training algorithm for suspended sediment concentration prediction and estimation*”, Hydrological Sciences–Journal–des Sciences Hydrologiques, 49(6) December 2004. Civil Engineering Department, Hydraulics Division, Engineering Faculty, Erciyes University, 38039 Kayseri, Turkey.

[PEN 1990], **Peng Y. et J.A. REGGIA** (1990). « *Abductive Inference Models For Diagnostic Problem-Solving* », Springer-Verlag, New York.

[POU 1996], **Poulard H.**, (1996). « *Statistiques et réseaux de neurones pour un système de diagnostic* », *Application au diagnostic de pannes automobiles*, Thèse de Doctorat, LAAS, Toulouse, France.

[TOG 1992], **Toguyeni A.**, (1992). « *Surveillance et diagnostic en ligne dans les ateliers flexibles de l'industrie manufacturière* », Thèse de doctorat, Université de Lille.

[VAL 1994], **Valette R., L. A. KÜNZLE** (1994). « *Réseaux de Petri pour la détection et le diagnostic* », Journées Nationales : Sûreté, Surveillance, Supervision, GdR Automatique, Paris, France.

[VIL 1988], **Villemeur A.**, (1988). « *Sûreté de fonctionnement des systèmes industriels* », Edition Eyrolles, Collection DER-EDF, Volume 67.

[WEB 1999], **Weber P.**, (1999). « *Diagnostic de procédés par l'analyse des estimations paramétriques de modèles de représentation à temps discret* », Thèse de Doctorat, INPG, Grenoble.

[ZHA 1999], **Zhang Q.**, (1999). « *Identification et Surveillance de systèmes Dynamiques* », Habilitation à diriger les recherches, Université de Rennes1, Institut de Formation Supérieure en Informatique et en Communication.

## **ANNEXES**

ANNEXE I

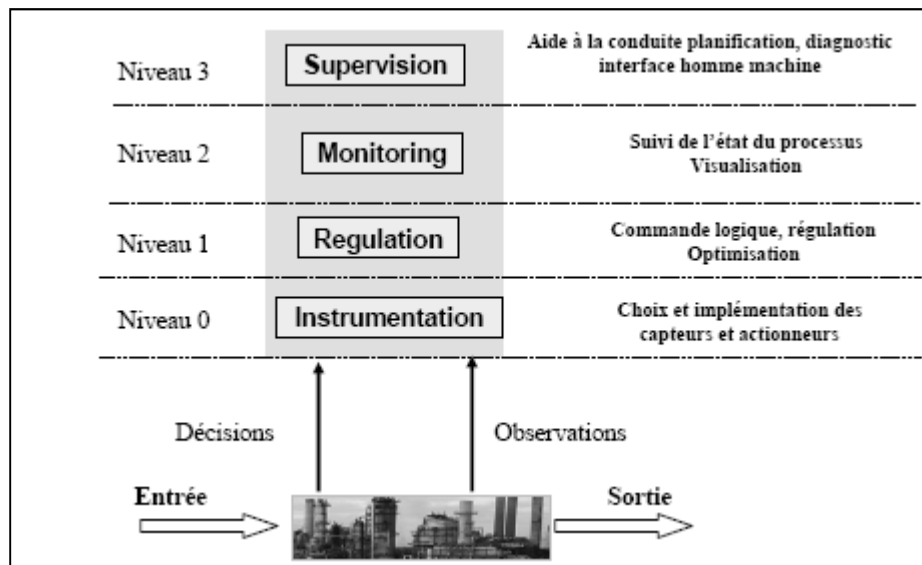


Figure I-1 : Niveaux de la supervision

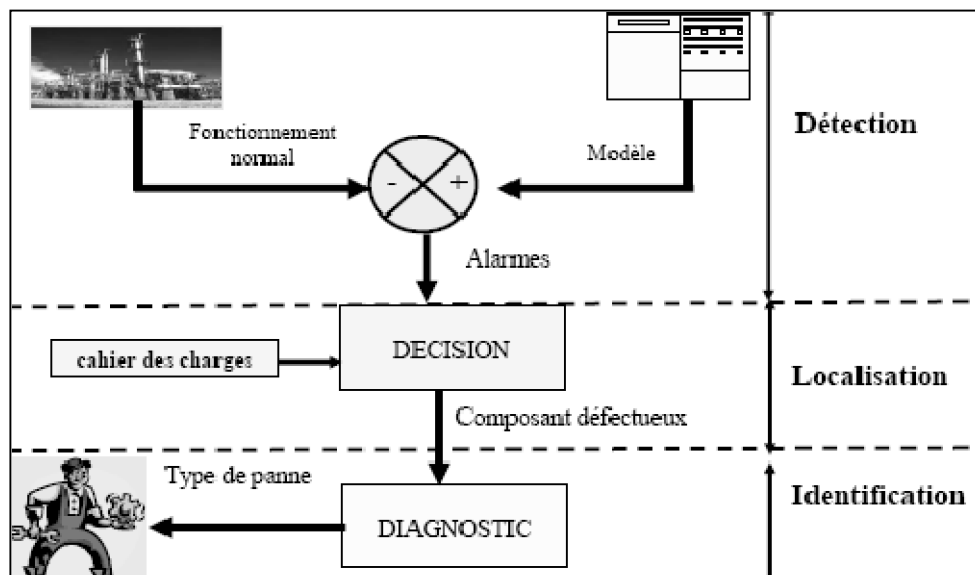


Figure I-2 : Taches d'un système de surveillance (FDI)

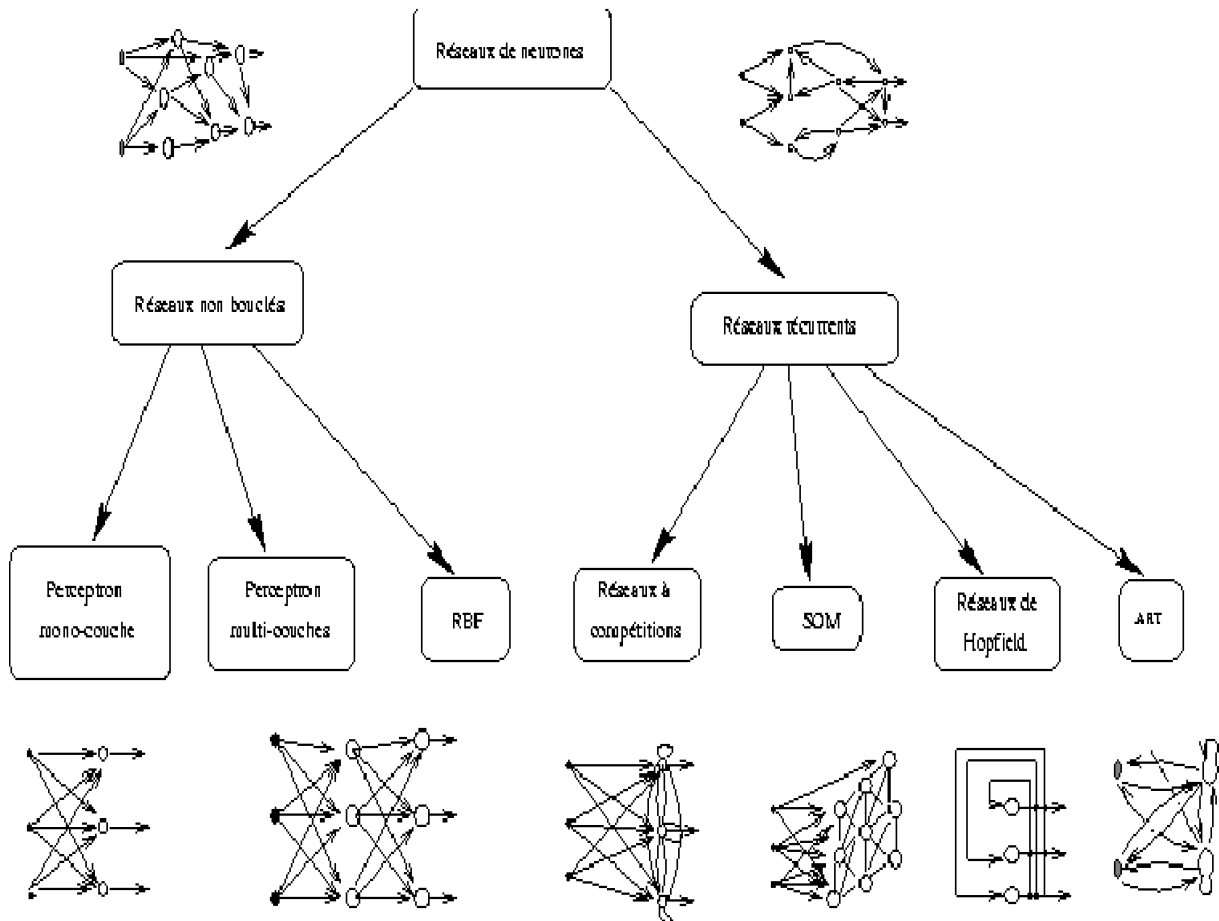


Figure I-3: Taxonomie des réseaux non-bouclés et récurrents

1. réseaux "feed-forward", qui sont des graphes acycliques ;
2. réseaux récurrents ou "feedback", qui sont des graphes avec circuits du fait de la présence d'arcs de rétro-action.

ANNEXE II

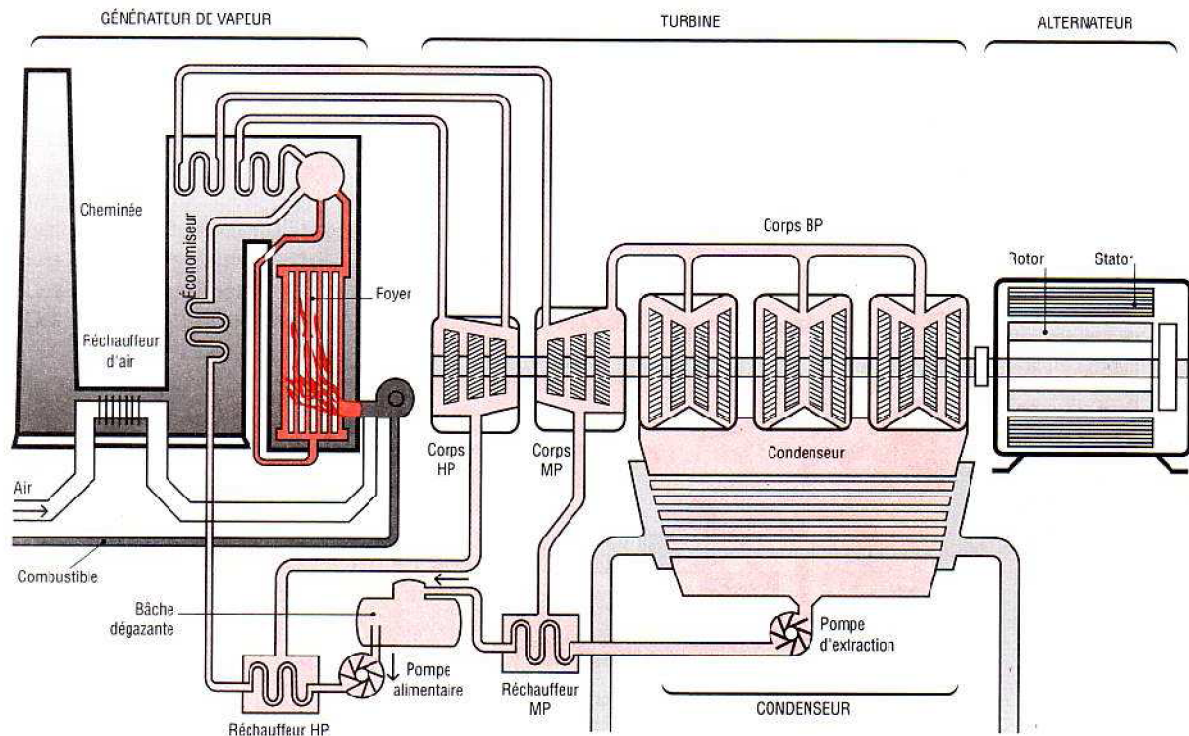


Figure II-1 : Implantation des différents éléments de la centrale thermique

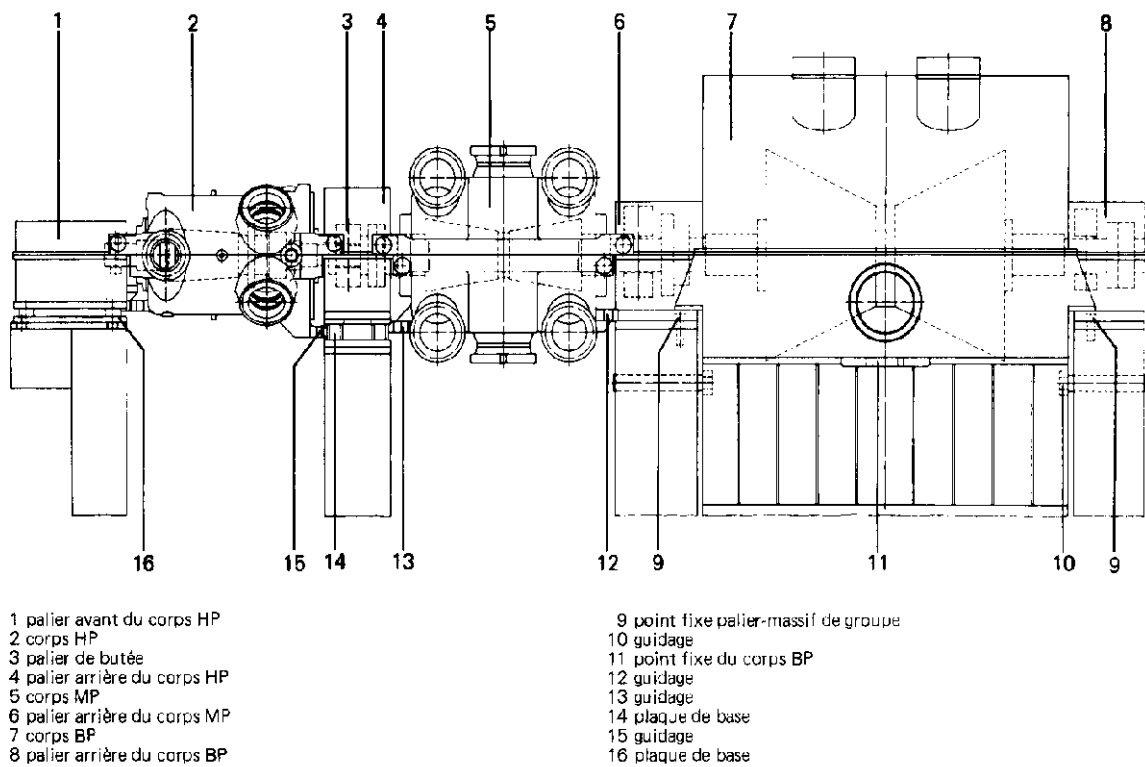


Figure II-2 : Schéma de la turbine à vapeur de la centrale thermique de Cap-Djinet

## ANNEXE III

### III-1 SURVEILLANCE DE LA TEMPERATURE DE VAPEUR

#### III-1-1 Données réelles

Il existe quatre sondes de mesure de température sous forme de thermocouples, deux sont placées à l'entrée et à la sortie du corps HP, une à l'entrée du corps MP et l'autre à la sortie du corps BP.

Position du capteur	Température nominale (°C)	Température minimale (°C)	Température maximale (°C)
entrée corps HP	535	500	650
sortie corps HP	356	300	480
entrée corps MP	535	500	600
sortie corps BP	33	25	90

#### III-1-2 Méthodologie du travail

Dans cette étude on a quatre intervalles de température à surveiller selon les capteurs existés, donc automatiquement on a quatre sorties désirées du réseau de neurone. Chaque intervalle est décomposée en deux parties, une partie représente les valeurs de température de bon fonctionnement est affectée à « 1 » et une autre partie qui représente les valeurs de température de mauvais fonctionnement est affectée à « -1 ». Entre chaque valeur minimale et maximale de température c'est l'intervalle de bon fonctionnement et extérieur c'est l'intervalle de mauvais fonctionnement comme suit :

Position du capteur	Intervalles affectées à « 1 »	Intervalles affectées à « -1 »
entrée corps HP	[500 650]	[0 500[ et ]650 800]
sortie corps HP	[300 480]	[0 300[ et ]480 800]
entrée corps MP	[500 600]	[0 500[ et ]600 800]
sortie corps BP	[25 90]	[0 25[ et ]90 800]

L'erreur limite ( $E_i$ ) donnée au réseau est de  $10^{-10}$ , le nombre d'itérations limite ( $it$ ) est **1000**, le pas de calcul c'est **0.001** et le taux d'apprentissage  $\mu = 0.5$ .

Pour obtenir l'architecture optimale nous devons suivre les étapes représentées dans l'organigramme de la figure 3.3 (chapitre III). Le nombre de neurone maximum donné pour la couche cachée est  $N_i = 20$ , donc l'architecture optimale trouvée est : « **801-08-04** » qui représente la plus petite erreur **9.96445e-011**.

#### III-1.2.1 Apprentissage du réseau

L'apprentissage de réseau MLP implique de trouver des valeurs des poids de raccordement  $w^1_{i,j}$  et  $w^2_{l,k}$ , qui réduisent au minimum la fonction de l'erreur moyenne MSE entre les résultats réels produits (calculés) et les valeurs à atteindre (désirés) correspondantes dans l'ensemble d'apprentissage. L'apprentissage du réseau par l'algorithme de Levenberg-Marquardt est fait directement à l'aide de la boîte à outils de réseau de neurones de Matlab.

Le graphe d'apprentissage se converge vers l'erreur fixée d'ordre  $10^{-10}$  après **110** itérations dans une durée de **76.82s**, voir figure III-1.

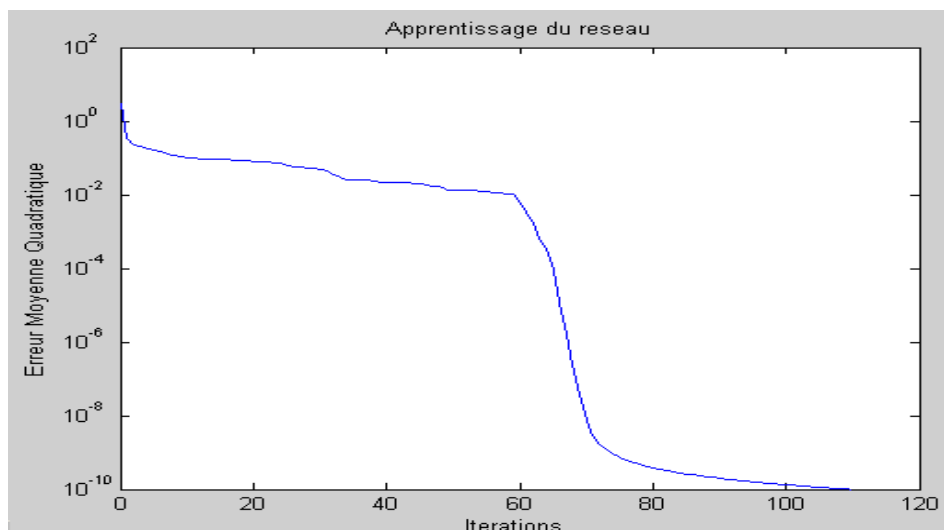


Figure III-1 : Apprentissage du réseau de neurone « 801-08-04 ».

### III-1.2.2 Simulation des sorties du réseau

On voit que le graphe vert du réseau est bien coïncidé avec le graphe bleu de la fonction désirée ce qu'il explique la bonne simulation de l'architecture trouvée, voir figure III-2.

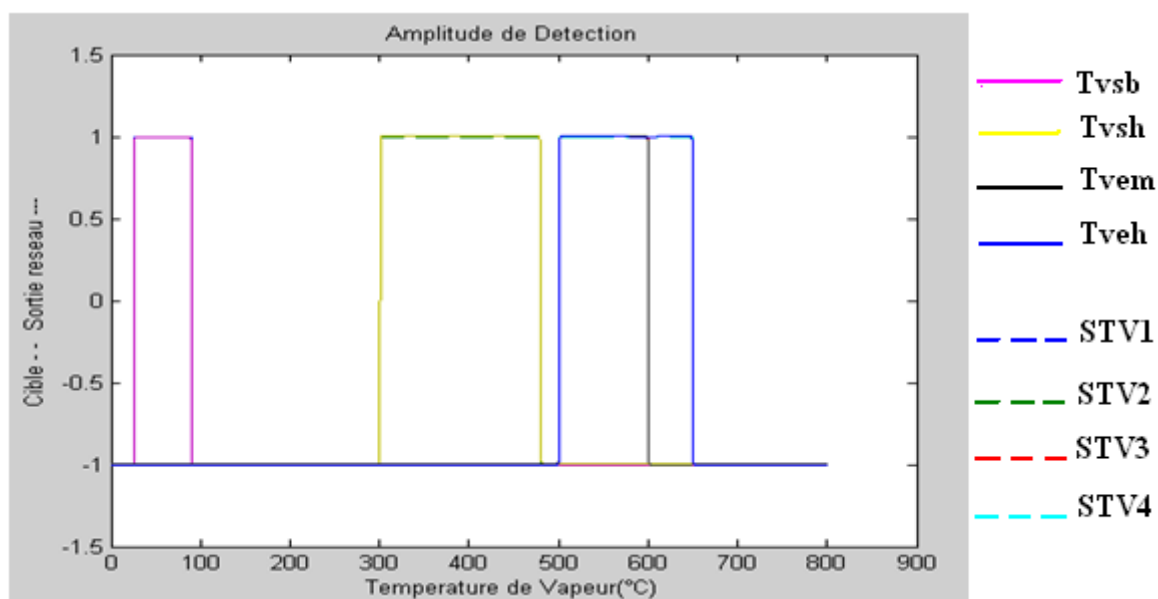


Figure III-2 : Simulation des sorties du réseau

### III-1.2.3 Détection et localisation des défauts (test du réseau)

Pour savoir bien que notre réseau est fiable et capable de détecter un défaut et de localiser ce dernier, on a injecté des défauts quelconques dans le réseau et on suit la réaction de ce dernier.

**Exemple :** On va tester maintenant avec des bases de test tel que toutes les températures n'appartiennent pas à l'intervalle admissible, on a testé par l'intervalle [100°C 164°C] pour Tehp, [500°C 679°C] pour Tshp, [600°C 699°C] pour Temp et par l'intervalle [100°C 249°C] pour Tsbp. Le réseau déclare quatre alarmes dans les quartes paliers, a cause des températures inacceptables reçues par le réseau. Voir figure III-3.

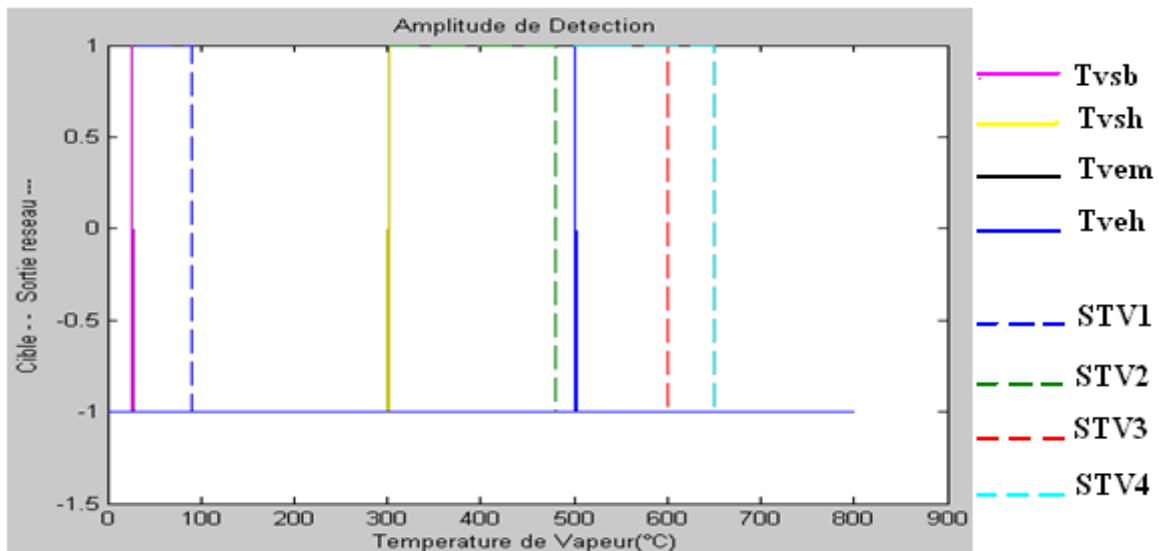


Figure III-3 : Test avec des intervalles de valeurs aléatoires

### III-1.2.4 Évaluation des performances

Nous remarquons sur la figure suivante la superposition entre la ligne continue, qui est le meilleur ajustement linéaire entre les cibles et les sorties du réseau et la ligne discontinue, qui est l'ajustement parfait. Ce qu'il montre la bonne corrélation entre les sorties du réseau et les sorties désirées tel que le coefficient de corrélation « R » égale à « 1 ». Donc nous déduisons que notre réseau est très performant ; voir figure III-4.

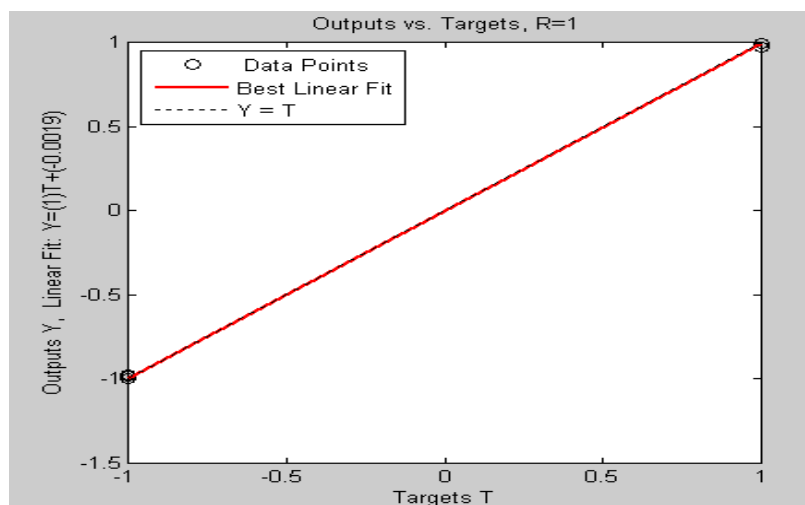


Figure III-4 : Performances du réseau trouvé

## III-2 SURVEILLANCE DES VIBRATIONS

### III-2-1 Méthodologie de travail

Il existe quatre points de mesure vibratoires sur la ligne d'arbre au niveau de chaque palier, tel que la signalisation d'alarme commence à 120  $\mu\text{m}$ , donc chaque valeur supérieur à cette valeur est une valeur inacceptable.

L'erreur limite donnée au réseau est de  $10^{-4}$ , le nombre d'itérations limite est **1000**, le pas de calcul c'est **0.001** et le taux d'apprentissage  $\mu = 0.5$ .

Pour obtenir l'architecture optimale, nous devons suivre les mêmes étapes de l'exemple précédent (III-1) on trouve l'architecture : « **951-12-04** » qui représente la plus petite erreur **1.90756e-005**.

### III-2-2 Explication des résultats

#### III-2.2.1 Apprentissage du réseau

Le graphe d'apprentissage se converge vers l'erreur fixée d'ordre  $10^{-4}$  après **111** itérations dans une durée de **113.86 s**, voir figure III-5.

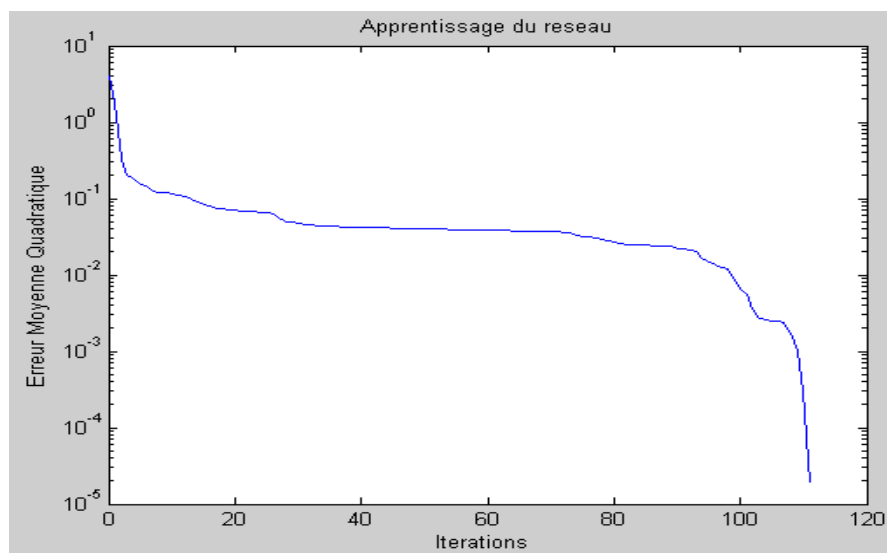


Figure III-5 : Apprentissage du réseau de neurone « **951-12-04** ».

#### III-2.2.2 Simulation des sorties de réseau

On voit que les graphes en ligne continue du réseau sont bien confondus avec les graphes en ligne discontinue des fonctions désirées, ce qu'il explique la bonne simulation de l'architecture trouvée, voir figure III-6.

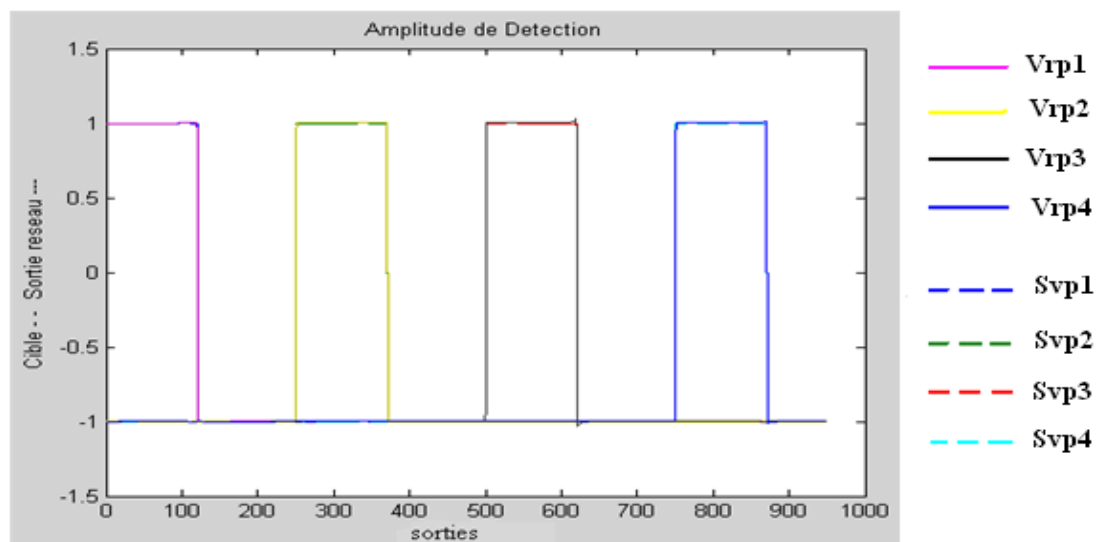


Figure III-6 : Simulation des sorties du réseau

### III-2.2.3 Détection et localisation des défauts (test du réseau)

Pour savoir bien que notre réseau est fiable et capable de détecter un défaut et de le localiser on injecte des défauts quelconques dans le réseau et on suit la réaction de ce dernier.

**Exemple** : dans cet exemple on va généraliser notre test à donner des intervalles aléatoires qui n'appartiennent pas à l'intervalle admissible, on a testé par l'intervalle  $[120 \mu\text{m} \ 239 \mu\text{m}]$  pour  $V_{rp1}$ ,  $[300 \mu\text{m} \ 419 \mu\text{m}]$  pour  $V_{rp2}$ ,  $[120 \mu\text{m} \ 239 \mu\text{m}]$  pour  $V_{rp3}$  et par  $[400 \mu\text{m} \ 519 \mu\text{m}]$  pour  $V_{rp4}$ .

Le réseau déclare quatre alarmes dans les quatre points de mesure, à cause des vibrations inacceptables reçues par le réseau. Voir figure III-7.

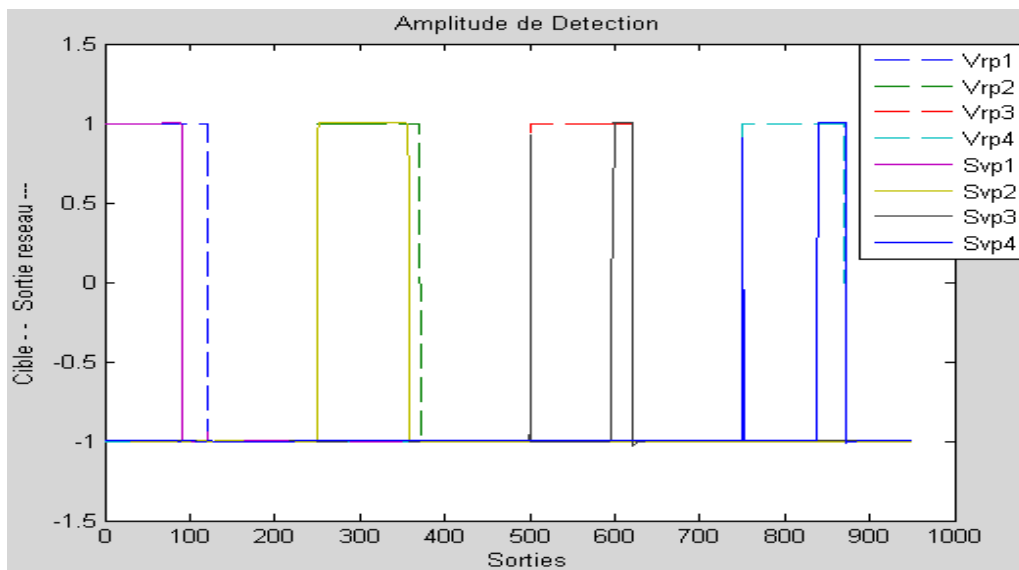


Figure III-7 : Test avec des intervalles de valeurs aléatoires

### III-2.2.4 Évaluation des performances

Dans la figure III-8 : on voit que le coefficient de corrélation « R » tend vers « 1 » ce qu'il montre la performance du réseau. Voir figure III-27.

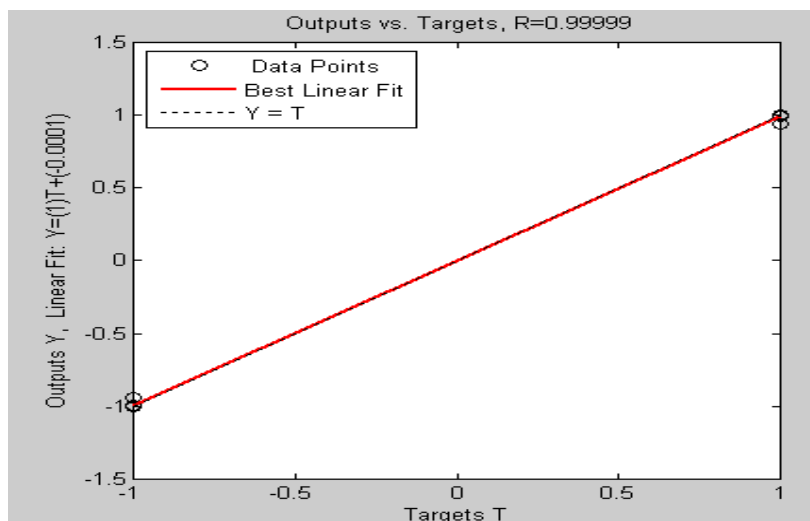


Figure III-8 : Performances du réseau trouvé

### III-3 SURVEILLANCE DU DEBIT D'HUILE DANS LES PALIERS

#### III-3.1 Valeurs minimales et maximales de débit d'huile dans chaque palier

N° du palier	Débit minimal ( $.10^{-4} \text{ m}^3/\text{s}$ )	Débit maximal ( $.10^{-4} \text{ m}^3/\text{s}$ )
Palier 1	6.000	10.000
Palier 2	80.000	100.000
Palier 3	35.000	50.000
Palier 3	30.000	45.000

#### III-3.2 Méthodologie de travail

Dans cette étude on a quatre intervalles de débit d'huile à surveiller selon les capteurs existés, donc automatiquement on a quatre sorties désirées du réseau de neurone. Chaque sortie réagit avec les données d'un de ces capteurs .on fait décomposer les valeurs de débit obtenues à partir des capteurs en deux intervalles .une intervalle porte les valeurs acceptables de bon fonctionnement et l'autre porte les valeurs non acceptables de mauvais fonctionnement.

N° du palier	Intervalles affectées à « 1 »	Intervalles affectées à « -1 »
Palier 1	[6.000 10.000]	[0 6.000 [ et ]10.000 120.0]
Palier 2	[80.000 100.000]	[0 80.00 [ et ]100.00 120.0]
Palier 3	[35.000 50.000]	[0 35.00 [ et ]50.00 120.0]
Palier 4	[30.000 45.000]	[0 30.00 [ et ]45.00 120.0]

L'erreur limite ( $E_i$ ) donnée au réseau est de  $10^{-5}$ , le nombre d'itérations limite  $it = 500$ , le pas de calcul c'est  $0.001$  et le taux d'apprentissage  $\mu = 0.5$ .

On suit les mêmes procédures des exemples précédents, nous obtenons l'architecture optimale suivante : « 04-10-04 » qui représente la plus petite erreur  $1.80361e-006$ .

#### III-3.3 Explication des résultats

##### III-3.3.1 Apprentissage du réseau

Le graphe d'apprentissage se converge vers l'erreur fixée d'ordre  $10^{-5}$  après 249 itérations dans une durée de 202.77 s, voir figure III-9.

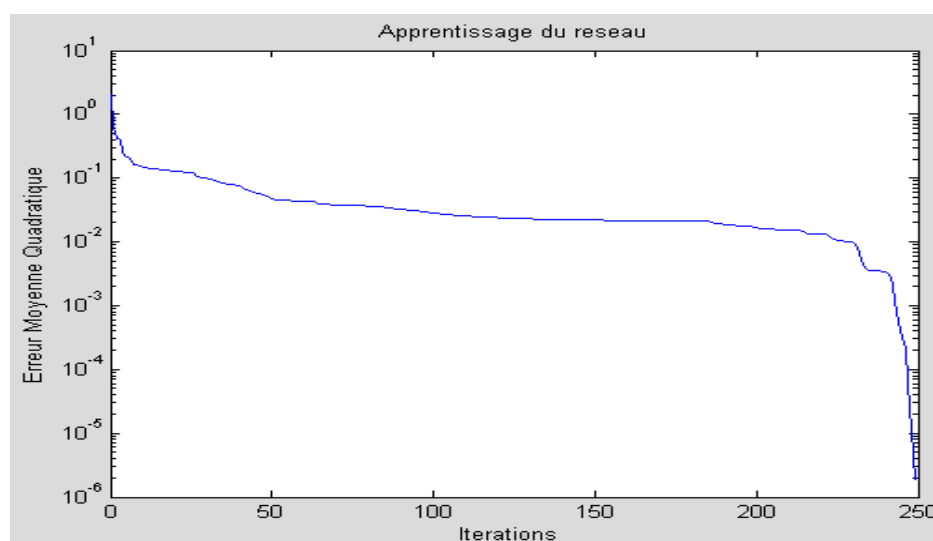


Figure III-9 : Apprentissage du réseau de neurone « 801-10-04 ».

### III-3.3.2 Simulation des sorties du réseau

On voit que les graphes en ligne continue du réseau sont bien coïncidés avec les graphes en ligne discontinue des fonctions désirées, ce qu'il explique la bonne simulation de l'architecture trouvée, voir figure III-10.

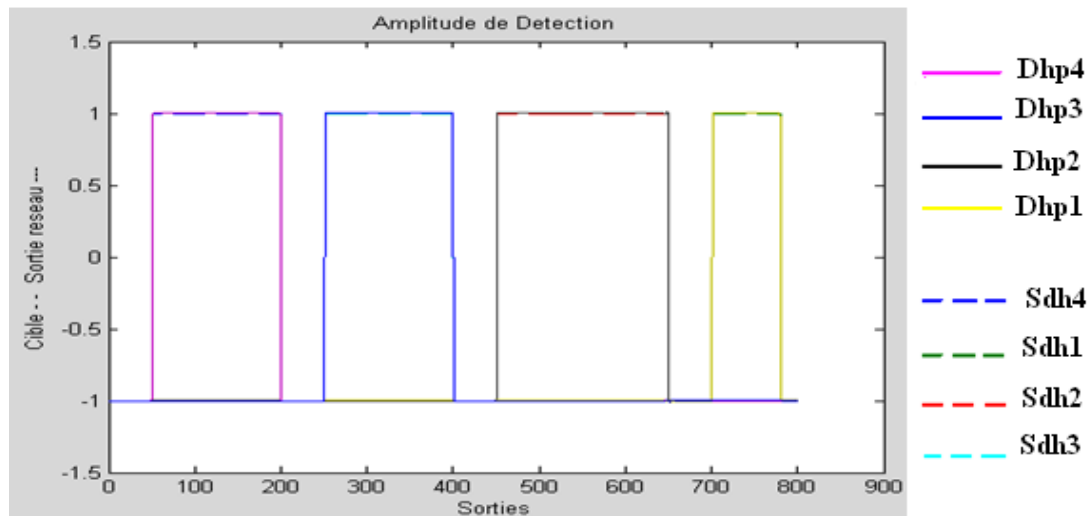


Figure III-10 : Simulation des sorties du réseau

### III-3.3.3 Détection et localisation des défauts (test du réseau)

Pour savoir bien que notre réseau est fiable et capable de détecter un défaut et de le localiser on injecte des défauts quelconques dans le réseau et on suit la réaction de ce dernier. C'est un test qui s'est fait sur quatre exemples

**Exemple** : toutes les températures n'appartiennent pas à l'intervalle admissible, on a donné  $3.10^{-4} \text{ m}^3/\text{s}$  pour le palier 1,  $50.10^{-4} \text{ m}^3/\text{s}$  pour le palier 2,  $30.10^{-4} \text{ m}^3/\text{s}$  pour le palier 3 et  $50.10^{-4} \text{ m}^3/\text{s}$  pour le palier 4. Le réseau déclare quatre alarmes dans les quartes paliers, à cause des températures inacceptables reçues par le réseau. Voir figure III-11.

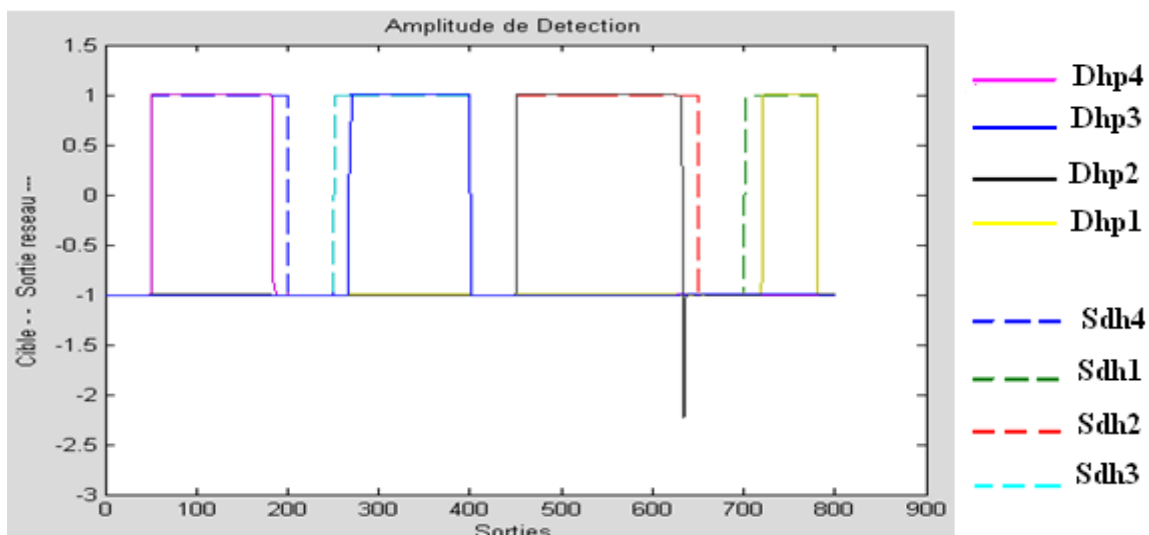


Figure III-11 : Test avec quatre valeurs inacceptables

**Remarque** : nous observons un pic vers le bas dans le graphe noir du palier 2, l'origine de ce pic reste toujours inconnue chez nous.

### III-3.3.4 Évaluation des performances

Dans la figure III-51 : on voit que le coefficient de corrélation « R » tend vers « 1 » ce qu'il montre la performance du réseau. Voir figure III-12.

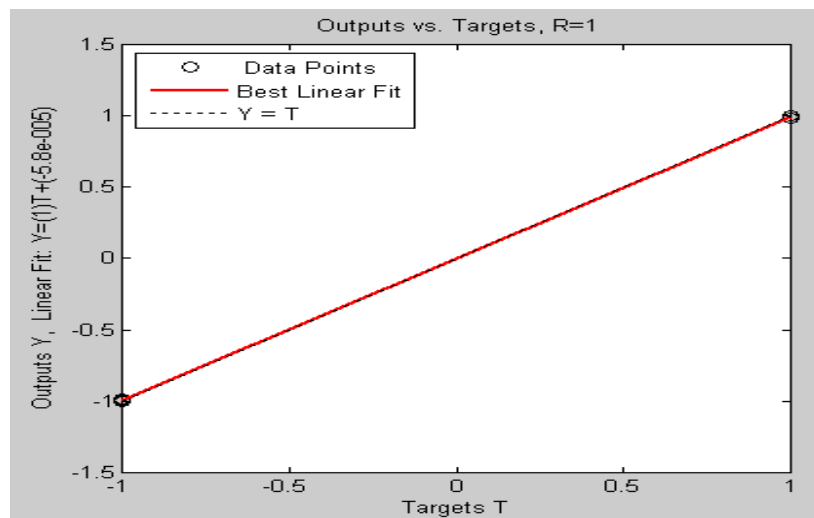


Figure III-12 : Performances du réseau trouvé

## III-4 SURVEILLANCE DE LA PRESSION DE LA VAPEUR

### III-4.1 Données réelles

Il existe quatre capteurs de mesure de pression de la vapeur sèche, deux sont placés à l'entrée et à la sortie du corps HP, un à l'entrée du corps MP et un à la sortie du corps BP.

Position du capteur	pression nominale ( bar )	pression minimale ( bar )	pression maximale ( bar )
entrée corps HP (Pehp)	109.8	100.00	130.00
sortie corps HP (Pshp)	40.00	35.000	50.000
entrée corps MP (Pemp)	35.00	30.000	45.000
sortie corps BP (Psbp)	1.500	0.5000	03.000

### III-4.2 Méthodologie du travail

On mesure la pression de la vapeur à quatre points nécessaires de la turbine, entrée corps HP, sortie corps HP, entrée corps MP, sortie corps BP. Les intervalles de pressions admissibles sont présentés dans le tableau suivant :

	Intervalles affectées à « 1 »	Intervalles affectées à « -1 »
entrée corps HP	[100 130]	[0 100[ et ]130 950]
sortie corps HP	[35 50]	[0 35[ et ]50 950]
entrée corps MP	[30 45]	[0 30[ et ]45 950]
sortie corps BP	[0.5 3.0]	[0 0.5[ et ]3.0 950]

L'erreur limite donnée au réseau est  $E_i = 10^{-5}$ , le nombre d'itérations limite est  $it = 500$ , le pas de calcul c'est  $0.001$  et le taux d'apprentissage  $\mu = 0.5$ .

Pour obtenir l'architecture optimale, nous devons suivre les étapes de l'organigramme représenté dans le début de ce chapitre, donc l'architecture optimale trouvée est : « 951-08-04 » qui représente la plus petite erreur  $1.46169e-006$ .

### III-4.3 Explication des résultats

#### III-4.3.1 Apprentissage du réseau

Le graphe d'apprentissage se converge vers l'erreur fixée d'ordre  $10^{-5}$  après 111 itérations dans une durée de **84.70 s**, voir figure III-13.

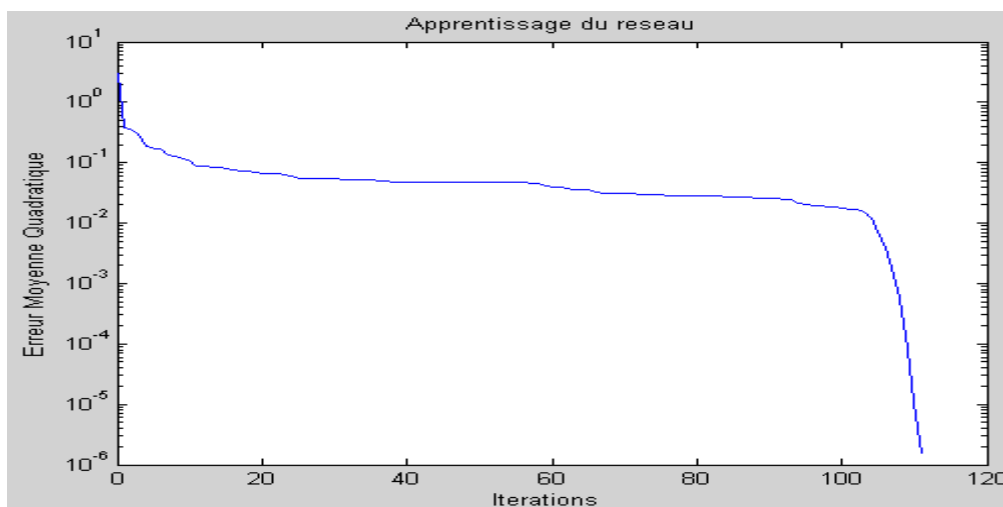


Figure III-13 : Apprentissage du réseau de neurone « 951-08-04 ».

#### III-4.3.2 Simulation des sorties du réseau

On voit que les graphes en ligne continue du réseau sont bien coincés avec les graphes en ligne discontinue des fonctions désirées, ce qu'il explique la bonne simulation de l'architecture trouvée, voir figure III-14.

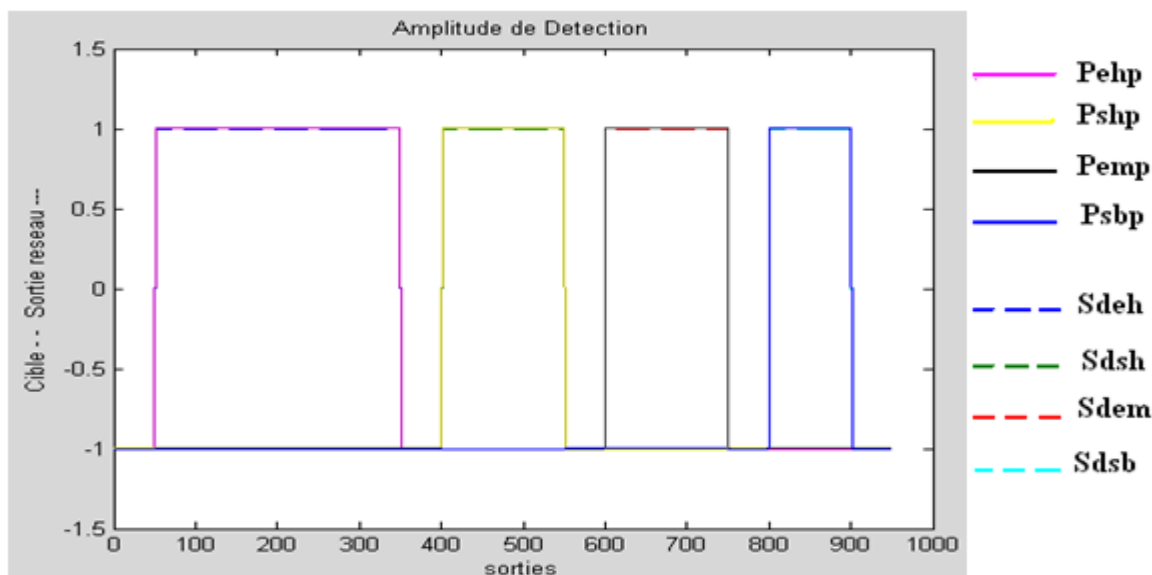


Figure III-14 : Simulation des sorties du réseau

#### III-4.3.3 Détection et localisation des défauts (test du réseau)

Pour savoir bien que notre réseau est fiable et capable de détecter un défaut et de le localiser on injecte des défauts quelconques dans le réseau et on suit la réaction de ce dernier.

**Exemple** : toutes les valeurs de pression n'appartiennent pas à l'intervalle admissible, on a donné 140 bar pour  $P_{EHP}$ , 60 bar pour  $P_{SHP}$ , 25 bar pour  $P_{EMP}$  et 4.00 bar pour  $P_{SBP}$ . Le réseau déclare quatre alarmes dans les quatre points de mesure, à cause des pressions inacceptables reçues par le réseau. Voir figure III-15.

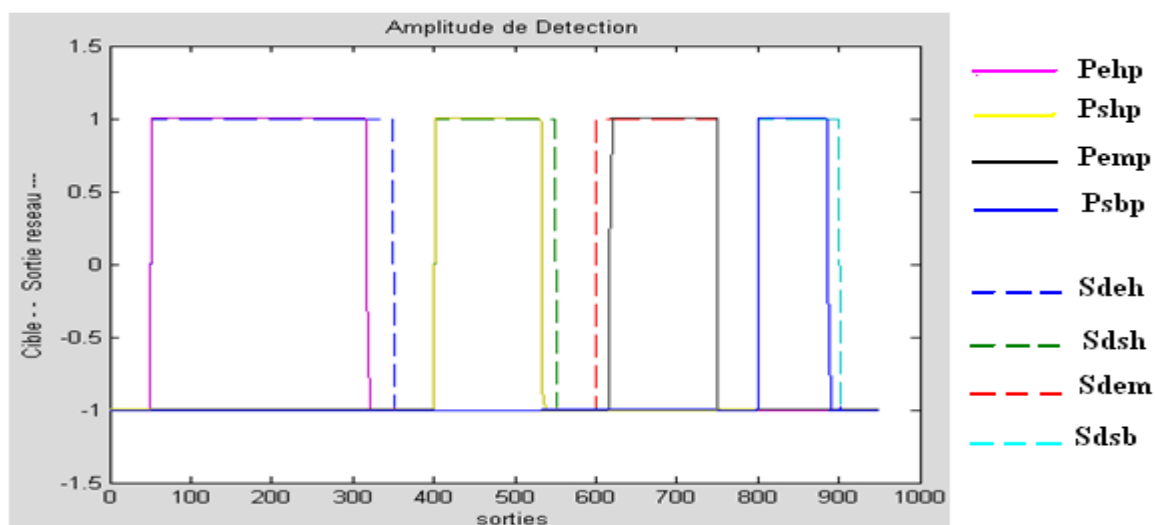


Figure III-15 : Test avec quatre valeurs inacceptables

#### III-4.3.4 Évaluation des performances

Dans la figure III-16 : on voit que le coefficient de corrélation « R » tend vers « 1 » ce qu'il montre la performance du réseau. Voir figure III-16.

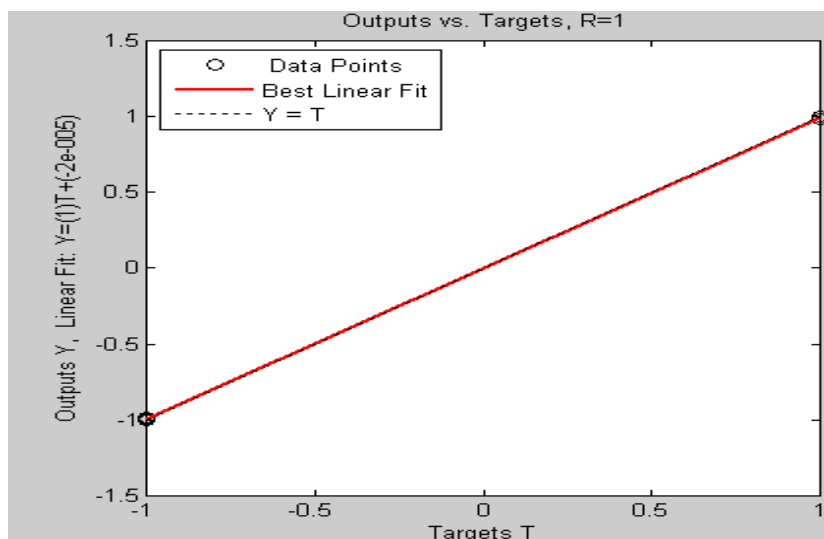


Figure III-16 : Performances du réseau trouvé

Nom du document : SAMIR  
Répertoire : H:  
Modèle : C:\Documents and Settings\ben samir\Application  
Data\Microsoft\Templates\Normal.dotm  
Titre : MEMORY.SAMIR 2010  
Sujet :  
Auteur : Client  
Mots clés : SAMI 1983  
Commentaires :  
Date de création : 17/01/2003 12:33:00  
N° de révision : 96  
Dernier enregistr. le : 24/06/2010 11:17:00  
Dernier enregistrement par : samir  
Temps total d'édition : 564 Minutes  
Dernière impression sur : 24/06/2010 11:18:00  
Tel qu'à la dernière impression  
Nombre de pages : 114  
Nombre de mots : 27 552 (approx.)  
Nombre de caractères : 151 537 (approx.)