

MINISTÈRE DE L'ENSEIGNEMENT SUPÉRIEUR ET DE LA RECHERCHE  
SCIENTIFIQUE



Université El-Hadj Lakhdar - BATNA -  
Faculté des Sciences de l'Ingénieur  
Département de Génie Industriel



# MÉMOIRE DE MAGISTER

PRÉSENTÉ AU

**Laboratoire d'Automatique et Productique**

En vue de l'obtention du

**MAGISTER**

Spécialité : Génie Industriel

Par

**Mohammed BOUZENITA**

Ingénieur en Informatique

**Thème :**

## **CONCEPTION ET IMPLÉMENTATION D'UN SYSTÈME EXPERT HYBRIDE POUR LE DIAGNOSTIC D'UN SYSTÈME INDUSTRIEL**

*Directeur de mémoire : Mme L.H. Mouss*

**JURY**

---

<i>Dr.Dj.Mouss</i>	<i>Président</i>	<i>Université de Batna</i>
<i>Dr.L.H.Mouss</i>	<i>Rapporteur</i>	<i>Université de Batna</i>
<i>Dr.M.Taibi</i>	<i>Examineur</i>	<i>Université de Annaba</i>
<i>Dr.H.Smadi</i>	<i>Examineur</i>	<i>Université de Batna</i>
<i>Dr.H.Kalla</i>	<i>Examineur</i>	<i>Université de Batna</i>

---

Année 2012

---

**CONCEPTION ET  
IMPLÉMENTATION  
D'UN SYSTÈME  
EXPERT HYBRIDE  
POUR LE  
DIAGNOSTIC  
D'UN SYSTÈME  
INDUSTRIEL**

---

**MOHAMMED BOUZENITA**  
Ingénieur en informatique

---

2012

---



## إهداء

إلى الوالدين الحبيبين ثمرة من ثمرات جهدهم تبقى وضاعة في حياتهم  
وبعدهم.

إلى رفيقتي في عمري، زوجتي.

إلى أختي وإخوتي، وكل أفراد أهلي وأقاربي.

إلى كل من عرفني يوماً ما من قريب أو بعيد سلباً أو إيجاباً.

أهدي هذا العمل المتواضع ونفعنا الله به والبشرية جمعاء.

محمد بوزنيته

# REMERCIEMENT

Dieu le tout puissant loué, pour la volonté, la santé et la patience qu'il m'a donné durant ces années d'études afin que je puisse arriver à ce travail.

Le présent travail est réalisé dans le groupe 3S au laboratoire d'automatique et productive sous la direction du Docteur Leila Hayet Mouss au Département Génie Industriel de la Faculté de l'Ingénieur que je tiens à remercier vivement pour ses conseils et encouragements ainsi que sa disponibilité durant toute la réalisation de ce travail.

Un grand remerciement est adressé à monsieur Toufik Bentrchia pour tous ces aides et ces conseils et surtout sa disponibilité durant toute la réalisation de ce mémoire.

Mes remerciements sont aussi adressés aux membres de Jury qui m'ont honorés de leur présence et d'avoir accepté de juger ce modeste travail :

- Docteur Djamel Mouss d'avoir présidé le jury.
- Docteur Mohammed Taibi d'avoir accepté de juger ce travail.
- Docteur Hacem Smaïd d'avoir accepté de juger ce travail.
- Docteur Hammouï Kalla d'avoir accepté de juger ce travail.

Je n'oublie pas de remercier également les responsables et les enseignants du département Génie Industriel, mes amis et toutes les personnes ayant contribué de loin ou de près pour la concrétisation de ce travail.

**Mohammed Bouzenita**

# SOMMAIRE

## INTRODUCTION GENERALE

### CHAPITRE 1 : METHODES DE DIAGNOSTIC HYBRIDES

1.1 Introduction.....	5
1.2 Diagnostic industriel.....	6
1.3 Système de diagnostic des pannes.....	7
1.3.1 Concepts de base du système de diagnostic des pannes .....	7
1.3.2 Caractéristiques désirées pour un système de diagnostic des pannes .....	7
1.4 Méthodes de diagnostic.....	10
1.4.1 Méthodes de diagnostic basées sur un modèle.....	10
1.4.2 Méthodes d'analyse fonctionnelle .....	11
1.4.2.1 Méthode SADT (structure analysis design technique) .....	12
1.4.2.2 Les arbres de défaillances AdD.....	13
1.4.2.2.1 Construction du l'arbre de défaillance .....	15
1.4.2.3 Analyse des modes de défaillances, de leurs effets et de leur criticité (AMDEC).....	16
1.4.2.3.1 Construction des tableaux AMDEC.....	17
1.4.3 Méthodes de diagnostic basées sur un modèle physique .....	19
1.4.4 Méthodes de diagnostic par analyse des signatures externes.....	19
1.4.4.1 Reconnaissance des formes .....	19
1.5 Méthodes de diagnostic hybrides.....	20
1.5.1 Systèmes experts hybrides .....	21
1.5.1.1 Système expert flou .....	22
1.5.1.2 Système expert connexionniste .....	23
1.5.1.3 Systèmes experts neuro-flou .....	24
1.5.2 Réseaux de neurones flous .....	26
1.6. Conclusion .....	27

### CHPITRE 2 : SYSTEMES EXPERTS

2.1. Introduction.....	29
2.2. Historique .....	30
2.3. Intelligence artificielle.....	31
2.4. Systèmes experts.....	32

2.4.1. <i>Importance des systèmes experts:</i> .....	33
2.4.2. <i>Quelques domaines d'applications</i> .....	34
2.5. <i>Architecture d'un système expert</i> .....	34
2.6. <i>Acquisition des connaissances</i> .....	36
2.6.1. <i>Ingénieur de connaissances</i> .....	37
2.6.2. <i>Techniques d'acquisition de connaissances</i> .....	37
2.6.2.1. <i>Techniques d'acquisition manuelles</i> .....	37
2.6.2.2. <i>Acquisition automatisée</i> .....	38
2.7. <i>Systèmes à règles de production</i> .....	40
2.7.1. <i>Règles de production</i> .....	41
2.7.2. <i>Base de faits</i> .....	41
2.7.3. <i>Moteur d'inférence</i> .....	41
2.7.3.1. <i>Cycle de base d'un moteur d'inférence</i> .....	41
2.7.3.2. <i>Stratégies de résolution de conflit</i> .....	42
2.7.3.3. <i>Modes de raisonnement</i> .....	43
2.7.3.4. <i>Ordre d'un système expert à règle de production</i> .....	43
2.8. <i>Développement d'un système expert</i> .....	44
2.8.1. <i>Analyse des connaissances</i> .....	44
2.8.2. <i>Maquettage et prototypage</i> .....	45
2.8.3. <i>Test et validation</i> .....	45
2.8.4. <i>Evolution et maintenance</i> .....	45
2.9. <i>Conclusion</i> .....	46

### CHAPITRE 3 : RESEAUX DE NEURONNES ARTIFICIELS

3.1. <i>Introduction</i> .....	48
3.2. <i>Historique</i> .....	49
3.3. <i>Neurone et réseau de neurones</i> .....	50
3.3.1. <i>Neurone biologique</i> .....	50
3.3.2. <i>Neurone formel</i> .....	51
3.3.3. <i>Réseau de neurones artificiels</i> .....	52
3.4. <i>Types d'architectures</i> .....	52
3.4.1. <i>Réseaux de neurones non bouclés</i> .....	52
3.4.2. <i>Réseaux de neurones bouclés</i> .....	53
3.5. <i>Types d'apprentissage</i> .....	54

3.5.1. Apprentissage supervisé.....	54
3.5.2. Apprentissage non supervisé.....	55
3.6. Quelques réseaux de neurones usuels.....	55
3.6.1. Perceptron.....	56
3.6.2. Perceptron multicouches(PMC)/ Multilayer Perceptron (MLP).....	58
3.6.3. Réseaux RBF (Radial Basis Function).....	59
3.6.4. Adaline et Madaline.....	60
3.6.5. Cartes topologiques de Kohonen.....	61
3.7. Mise en œuvre des réseaux de neurones.....	62
3.7.1. Comment qualifier qu'une application est candidate d'être modélisée par les réseaux de neurones ?.....	62
3.7.2. Conditions d'utilisation des réseaux de neurones.....	63
3.7.3. Mise en œuvre des réseaux de neurones.....	63
3.8. Domaines d'applications des réseaux de neurones.....	64
3.9. Conclusion.....	65
<b>CHAPITRE 4 : APPROCHE DEVELOPPEE APPLIQUEE AU DIAGNOSTIC D'UN PROCESSUS INDUSTRIEL</b>	
4.1. Introduction.....	67
4.2. Présentation de l'approche hybride adoptée.....	68
4.3. Etape 1 : Construction de RNA.....	69
4.4. Approche de réduction statistique.....	70
4.4.1. Construction de vecteur d'état initial.....	71
4.4.2. Sélection des descripteurs.....	71
4.4.3. Représentation graphique.....	72
4.5. Etape 2 : Construction de la base des connaissances.....	73
4.5.1. Définition de système.....	73
4.5.2. Analyse fonctionnelle.....	73
4.5.3. Analyse dysfonctionnelle.....	73
4.5.3.1. Elaboration des tableaux AMDEC.....	73
4.5.3.2. Construction de l'AdD.....	74
4.5.4. Représentation des prédicats.....	75
4.5.5. Insertion des prédicats dans la base des connaissances.....	76
4.6. Conclusion.....	77

## CHAPITRE 5 : APPLICATION AU DIAGNOSTIC D'UN PROCESSUS INDUSTRIEL

<i>5.1.Introduction</i> .....	79
<i>5.2.Exemple d'application</i> .....	80
<i>5.2.1.Description du processus industriel</i> .....	80
<i>5.2.2.Etape 1 : Construction du RNA</i> .....	81
<i>5.2.2.1. Application de l'approche de réduction statistique proposée</i> .....	81
<i>5.2.2.2. Construction de RNA</i> .....	84
<i>5.2.3. Etape 2 : Construction de la base des connaissances</i> .....	87
<i>5.2.3.1. Définition de système</i> .....	87
<i>5.2.3.2. Analyse fonctionnelle</i> .....	87
<i>5.2.3.3. Analyse dysfonctionnelle</i> .....	88
<i>5.2.3.3.1. Elaboration des tableaux AMDEC</i> .....	88
<i>5.2.3.3.2. Construction de l'AdD</i> .....	89
<i>5.2.3.4. Représentation des prédicats</i> .....	90
<i>5.2.4. Codage sous prolog</i> .....	91
<i>5.2.5. Etude comparative</i> .....	95
<i>5.3. Conclusion</i> .....	98

## CONCLUSION GENERALE

## LISTE DES TABLEAUX

### **Chapitre 1**

*Tableau 1.1.a : Operateurs fondamentaux de l'AdD*

*Tableau 1.1.b : Operateurs spéciales de l'AdD*

*Tableau 1.1.c : Evènements de l'AdD*

*Tableau 1.1.d : Triangles de renvoi de l'AdD*

*Tableau 1.2 : Liste de modes de défaillance suivant l'AFNOR*

*Tableau 1.3 : Etude comparative entre quelques systèmes experts hybrides*

### **Chapitre 2**

*Tableau 2.1 : Quelques domaines d'applications des systèmes experts.*

### **Chapitre 5**

*Tableau.5. 1. Modes de fonctionnement de la chaîne de production étudiée.*

*Tableau 5.2 : Résumé des résultats obtenus de l'application de la régression pas à Pas.*

*Tableau 5.3 : Contribution des différentes composantes dans la variabilité des modes de défaillances.*

*Tableau 5.4: Coefficient de performance de chaque phase.*

*Tableau 5.5 : Déroulement de chainage pour la détection des défaillances dans le système étudié.*

*Tableau 5.6 : Etude comparative.*

## LISTE DES FIGURES

### **Chapitre 1**

- Figure 1.2 : Classification des méthodes de diagnostic*
- Figure 1.3 : Description fonctionnelle d'une machine à laver la vaisselle*
- Figure 1.4 : Description matérielle d'une machine à laver la vaisselle*
- Figure 1.5 : Étapes d'analyse fonctionnelle*
- Figure 1.6 : Diagramme type SADT*
- Figure 1.7 : Boîtes SADT : (a) Actigramme, (b) Datagramme*
- Figure 1.8 : Contenu d'un tableau d'AMDEC suivant l'organisme CETIM*
- Figure 1.9 : Étapes de diagnostic par reconnaissance des formes*
- Figure 1.10 : Architecture d'un système expert flou*
- Figure 1.11 : Architecture d'un système expert connexionniste*

### **Chapitre 2**

- Figure 2.1 : Disciplines et domaines d'applications de l'IA*
- Figure 2.2 : pôles principaux de système expert*
- Figure 2.3 : Architecture d'un système expert*
- Figure 2.4 : Acquisition automatique des connaissances*
- Figure 2.5 : Raffinement des paramètres de base de connaissance*
- Figure 2.6 : Architecture d'un système à règles de production*

### **Chapitre 3**

- Figure 3.1 : Neurone biologique*
- Figure 3.2 : Représentation graphique d'un neurone artificiel*
- Figure 3.3 : Représentation graphique d'un réseau de neurones artificiels*
- Figure 3.4 : Réseau de neurones non bouclé*
- Figure 2.5 : Réseau de neurones bouclé*
- Figure 3.6 : Représentation graphique de perceptron*
- Figure 3.7 : Architecteur générale d'un réseau RBF*
- Figure 3.8 : Schémas de principe de l'Adaline*
- Figure 3.9 Schémas de principe de MAdaline (exemple de deux Adalines)*
- Figure 3.10 : l'architecteur générale de la carte topologique*

### **Chapitre 4**

- Figure 4.1 : Construction de la base des connaissances.*
- Figure 4.2 : Architecture de SE hybride proposé.*
- Figure 4.3 : Architecture générale de RNA utilisé pour la détection des défaillances élémentaires.*

*Figure 4.4 : Approche de réduction statistique.*

*Figure 4.5 : Insertion des prédicats dans la base des connaissances.*

### **Chapitre 5**

*Figure 5.1 : Processus industriel avec réacteur et un échangeur de chaleur*

*Figure 5.2 : Différents modes de fonctionnement représentés dans l'espace réduits généré par l'ACP.*

*Figure 5.3 : Architecture optimale de RNA utilisé.*

*Figure 5.4 : Evolution de l'erreur quadratique minimal avec le nombre d'itérations de chaque catégorie.*

*Figure 5.5 : Représentation de la corrélation entre les données réelles et les données de notre RNA pour la phase d'apprentissage.*

*Figure 5.6 : Représentation de la corrélation entre les données réelles et les données de notre RNA pour la phase de validation.*

*Figure 5.7 : Représentation de la corrélation entre les données réelles et les données de notre RNA pour la phase de teste.*

*Figure 5.8 : Diagramme A-0 de processus.*

*Figure 5.9 : Diagramme A0 de processus.*

*Figure 5.10 : Diagramme A1 de processus.*

*Figure 5.11 : Diagramme A2 de processus.*

*Figure 5.12 : Exemple d'un tableau AMDEC de processus.*

*Figure 5.13 : L'AdD de processus.*

*Figure 5.14 : Architecture générale de l'application développée.*

*Figure 5.15 : Une partie de la base des connaissances de notre SE sous SWI-Prolog.*

*Figure 5.16: résultat de diagnostic du SE.*

*Figure 5.17: Propagation des défaillances dans la procédure de diagnostic.*

## **LISTE DES ALGORITHMES**

### **Chapitre 3**

*Algorithme 3.1 : Algorithme d'apprentissage de perceptron.*

*Algorithme 3.2 : Algorithme d'apprentissage non supervisé des réseaux de Kohonen*

### **Chapitre 4**

*Algorithme 4.1 : Etapes d'élaboration de la base des connaissances*

## **LISTE DES ORGANIGRAMMES**

### **Chapitre 1**

*Organigramme 1.1 : Construction d'AMDEC selon IEC 60812.*

## INTRODUCTION GENERALE

Depuis les années soixante, l'influence de l'automatisation sur la conception et la réalisation des processus industriels comme les systèmes de conversion d'énergie, et les systèmes de production, est augmenté progressivement. Cette crise d'automatisation revient à la demande progressive de systèmes plus performants et des produits de grande qualité. L'automatisation des processus industriels augmente de plus en plus à partir de 1975, où les coûts d'automatisation deviennent raisonnables. En plus l'arrivée des microordinateurs performants capables de résoudre plusieurs problèmes d'automatisation et la progression dans le développement des capteurs, des actionneurs, des interfaces homme-machine, simplifient la tâche d'automatisation [1].

A cause de cette automatisation, l'un des problèmes majeur des systèmes industriels, qui est le diagnostic, devient beaucoup plus compliqué, et l'application des méthodes classiques de diagnostic n'a pas donné des solutions optimales. Cette automatisation des procédés industriels implique un besoin d'informatisation des tâches de diagnostic. Pour cela, l'utilisation des techniques et des méthodes issues de l'intelligence artificielle (IA), comme par exemple les systèmes experts (SE) et les réseaux de neurones artificiels (RNA), est considérée comme une solution performante pour le développement des systèmes de diagnostic fiables. En plus, la complexité des systèmes industriels et la prise en considération des aspects non homogènes liés aux domaines mécanique, électronique, informatique,...etc. diminuent les performances des méthodes classiques de diagnostic. Aussi le besoin d'hybridation des méthodes, où une méthode est complémentaire d'une autre, est indispensable pour couvrir les limitations d'utilisation individuelle [2-3].

Les systèmes experts hybrides sont l'une des méthodes hybrides utilisées pour le diagnostic industriel. Un système expert est généralement composé de trois éléments principaux: Une base de connaissances, une base des faits et un moteur d'inférence. Le problème majeur de tel système est l'acquisition des connaissances où l'extraction de ces connaissances de l'expert de domaine est une tâche difficile et nécessite une plus grande patience. L'utilisation des RNA pour l'acquisition des connaissances est l'une des techniques les plus employées. Les RNA ont une grande capacité d'apprentissage, et une robustesse en présence de bruit et même en cas d'insuffisance de données. Donc l'utilisation des RNA pour l'acquisition des

connaissances afin d'alimenter dynamiquement la base des faits de système expert présente une hybridation entre les RNA et les SE, et forme un type des systèmes experts hybrides nommé SE connexionniste.

En effet, l'hybridation des méthodes est un champ actif pour produire des systèmes plus fiables et particulièrement pour le diagnostic industriel, qui reste toujours un problème majeur des systèmes complexes. Alors le but d'un système de diagnostic est non pas seulement la détection des pannes, mais aussi la localisation et l'identification de la cause de la panne (FDII: *Fault Detection, Isolation and Identification*), et pourquoi pas le pronostic des défaillances possibles [4].

L'objectif de ce travail est la proposition d'une approche hybride de diagnostic, ensuite le développement de cette approche sous forme d'un système expert hybride pour le diagnostic des pannes et l'application de l'outil développé pour un processus industriel. Nous avons utilisé le RNA pour la classification des observations, et la détection des pannes élémentaires afin d'alimenter la base des faits du SE. Cette insertion dans la base des faits déclenche l'inférence de SE pour la localisation de la panne et en plus la présentation de toutes les événements indésirables peuvent être générés par ces pannes élémentaires. Les méthodes SADT, AdD et AMDEC sont utilisées pour la construction de la base des règles.

Pour valider cette approche, elle est appliquée pour un processus industriel répandu dans la majorité des systèmes de type agro-alimentaires.

Pour ce faire, le mémoire est organisé en cinq chapitres:

Le **premier chapitre** est dédié à l'étude bibliographique relative aux différentes méthodes utilisées pour le diagnostic industriel et particulièrement les méthodes hybrides. Ce chapitre est organisé en deux parties: La première partie est réservée à la présentation des méthodes classiques de diagnostic industriel. Après une définition de problème de diagnostic, une présentation de quelques aspects des systèmes de diagnostic est donnée. Cette partie est terminée par l'exposition de quelques méthodes usuelles telle que SADT, AdD et AMDEC. Les méthodes hybrides de diagnostic industriel et particulièrement les systèmes experts hybrides (Système expert flou, Système expert connexionniste, système expert neuro-flou) font l'objet de deuxième partie.

Le **deuxième chapitre** est consacré à la présentation des SE: Les grandes lignes historiques, quelques domaines d'applications, leur architecture et particulièrement celle des systèmes à règles de production, le problème d'acquisition des connaissances et les techniques utilisées et enfin les étapes de développement d'un système expert.

L'objet du **troisième chapitre** est la présentation des RNA. Une présentation de quelques repères historiques des RNA et une description des fondements biologiques des neurones est traitée. Ensuite le principe de fonctionnement des RNA, les fameuses architectures connues et l'apprentissage de tel réseau sont exposées. Nous abordons ensuite la présentation de quelques réseaux usuels et particulièrement le perceptron multicouches et les réseaux à fonction de base radiale. Enfin le chapitre termine avec la détermination des étapes à suivre pour la mise en place des RNA et la présentation de quelque domaine d'applications.

Dans le **quatrième chapitre**, on s'intéresse à présenter une approche hybride (SE hybride), pour le diagnostic d'un système industriel en se basant sur les méthodes exposées dans les trois premiers chapitres.

Le **cinquième chapitre** est réservé à l'application de l'approche proposée pour un processus industriel (une partie d'une chaîne de production). Après la présentation de système industriel étudié, nous exposons le déroulement de l'exécution de l'application et les résultats obtenus.

Nous concluons ce mémoire par un bilan final du notre travail et des perspectives futures basées principalement sur les résultats obtenus.

## CHAPITRE 1

# MÉTHODES DE DIAGNOSTIC HYBRIDES

***Résumé :** L'objet de ce chapitre est la présentation des méthodes de diagnostic hybrides. Nous nous intéressons dans une première partie aux méthodes dites classiques de diagnostic industriel. On commence cette partie par la définition du problème de diagnostic industriel et la présentation de quelques aspects des systèmes de diagnostic. Enfin l'illustration de quelques méthodes de diagnostic usuelle : SADT, AdD et AMDEC est traitée. Nous exposons dans la deuxième partie quelques méthodes hybrides de diagnostic industriel. L'importance est donnée aux systèmes experts hybrides (Système expert flou, Système expert connexionniste, système expert neuro-flou) avec une explication de la manière de combinaison.*

## 1.1 Introduction

Le diagnostic industriel est une discipline inspirée du diagnostic médical, dont le but est non pas la détection des maladies chez un patient, mais la détection, la localisation et l'identification des pannes d'un dispositif industriel. Cette discipline vise le développement des outils, des algorithmes et des techniques pour déterminer le type, le degré, et l'emplacement d'une panne possible en plus de leur détection [1]. Plusieurs méthodes ont été développées pour être un outil très efficace d'aide au diagnostic des pannes industrielles. Leur efficacité varie selon le domaine étudié. Généralement ces méthodes sont classées en deux grandes classes: une classe basée sur une modélisation fonctionnelle, matérielle ou physique des défaillances (SADT, AdD, AMDEC, ...), et une deuxième classe s'intéresse à l'analyse des signatures externes, dans la plupart des cas la décision est un fruit d'une expertise ou d'un savoir-faire (Systèmes experts, Réseaux de neurones, reconnaissance des formes).

Les limites des méthodes de diagnostic ouvrent une possibilité de leur combinaison, et le développement des outils hybrides pour augmenter les performances d'une technique en profitant des capacités d'une autre. Cette combinaison peut être embarquée par l'intégration d'une méthode à l'intérieur d'une autre ou bien une méthode est complémentaire d'une autre comme par exemple l'utilisation d'une méthode pour la détection, une deuxième pour la localisation et une troisième pour l'identification des pannes. La combinaison des systèmes experts, des réseaux de neurones et des ensembles flous est un champ actif des méthodes hybride et en particulier pour le diagnostic industriel montré par l'augmentation des articles discutant ce sujet [2-6].

Dans ce chapitre nous commençons par la définition du problème de diagnostic ainsi que la présentation de quelques aspects des systèmes de diagnostic. Nous citerons ensuite quelques méthodes usuelles de diagnostic industriel (SADT, AdD, AMDEC). Nous nous intéressons par la suite à la présentation des méthodes hybrides et en particulier les systèmes experts hybrides (Système expert flou, Système expert connexionniste, système expert neuro-flou), ainsi que l'introduction de la logique floue dans les réseaux de neurones.

## 1.2 Diagnostic industriel

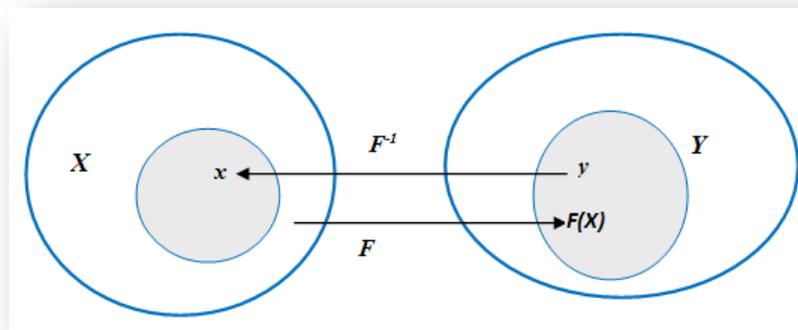
"Le diagnostic est l'identification de la cause probable de la (ou les) défaillance(s) à l'aide d'un raisonnement logique fondé sur un ensemble d'informations provenant d'une inspection, d'un contrôle ou d'un test" AFNOR, CEI

Le diagnostic industriel est une tâche cognitive dans plusieurs domaines : l'économie, la médecine, l'enseignement et l'industrie. Alors le rôle de diagnostic est la détermination des causes d'un dysfonctionnement ou d'une maladie à partir d'observations et des symptômes constatés [7].

La définition donnée par l'AFNOR résume deux tâches essentielles dans le diagnostic: l'observation des symptômes de la défaillance et l'identification de la cause de la défaillance à l'aide d'un raisonnement logique fondé sur des observations [8].

Le modèle mathématique de diagnostic présenté dans la figure 1.1 repose sur la considération d'un espace des causes  $X$ , et un espace  $Y$  des observations (les symptômes), où le problème à étudier est la détermination de  $x$  la solution de  $y$  en se basant sur la fonction  $F$ :  $F(x) = y$ , la solution évidente est donnée par :

$$x = F^{-1}(y) \quad (1.1)$$



**Figure 1.1 : Relation entre l'espace des causes  $X$  et l'espace des symptômes  $Y$**

La résolution du problème inverse repose sur la vérification si le problème est bien posé, cela revient à la réalité physique qui interdit l'utilisation de ce schéma directement [8].

Un problème est bien posé s'il vérifie les trois conditions suivantes :

- Existence de solution pour tout  $y$  appartenant à  $Y$ .
- Unicité de solution de  $x$  dans  $X$ .
- Continuité de la solution  $x(y)$ .

Le problème est considéré mal posé s'il ne remplit pas les conditions précédentes pour les raisons suivantes :

- Données aberrantes ou fausses  $y \notin Y$ .
- Données incompatibles  $y \notin F(x)$  mais  $y \in Y$ .
- Données expérimentales bruitées.
- Erreur de modélisation, approximation grossière de  $F$ .
- Choix inadéquat des topologies dans  $X$  et  $Y$ .

### 1.3 Système de diagnostic des pannes

Un système de diagnostic des pannes est un système de surveillance capable de détecter les pannes d'un système, déterminer leurs emplacements, et estimer les gravités. Donc c'est un système qui peut réaliser les trois tâches : détection, localisation et identification des pannes [9].

Le système de diagnostic des pannes dépendant de leur performances, est appelé FD (Fault Detection) pour la détection des pannes, FDI (Fault Detection and Isolation) pour la détection et la localisation des pannes, FDIA (Fault Detection, Isolation and Analysis) pour la détection, localisation et l'analyse (identification) des pannes, où les sorties correspondent aux signaux d'alarme indiquant l'occurrence d'une panne ou bien les données définissant le type et la grandeur des pannes [10]

#### 1.3.1 Concepts de base du système de diagnostic des pannes

Le diagnostic des pannes est basé généralement sur les trois tâches suivantes [9] [10]:

**Détection de la panne :** C'est la détection de l'occurrence de pannes dans l'unité fonctionnelle de processus, qui mène à la disfonctionnement totale ou partielle du système. Le but est de faire une décision binaire

**Localisation de la panne :** L'isolation et la classification des différentes pannes, c'est-à-dire la détermination de l'endroit de la panne, pour identifier le composant défectueux du système.

**Analyse ou identification de la panne :** Détermination de type, grandeur et la cause de la panne, et l'estimation de sa gravité.

#### 1.3.2 Caractéristiques désirées pour un système de diagnostic des pannes

Un système de diagnostic des pannes doit préférablement vérifier quelques critères. Parmi les paramètres les plus importants on peut situer [9] [11]:

##### 1. Détection et diagnostic rapide

C'est la capacité de diagnostic et de localisation au plus tôt des pannes (par le système de diagnostic. Cette capacité de détection rapide des pannes peut influencer sur les

performances de processus par la génération des fausses alarmes durant leur fonctionnement normal. C'est le problème majeur des systèmes de diagnostic pour la détection rapide des pannes.

## **2. Localisation**

Montre la capacité d'un système de diagnostic de distinguer l'origine de la panne parmi plusieurs sources potentielles ou bien localiser l'élément défectueux parmi les différents composants de systèmes. La localisation des pannes ne dépend pas de l'architecture du système de diagnostic seulement, mais aussi de la méthode de l'influence des pannes sur les sorties du système. En plus, plusieurs sources d'incertitudes comme les erreurs /incertitude de modélisation et les différentes perturbations de système restent le problème majeur pour atteindre une grande capacité de localisation.

## **3. Identification des pannes**

Pour estimer la gravité, le type et la nature des pannes, qui sont utiles pour les tâches de diagnostic, l'identification des pannes est une tâche importante pour le pronostic des pannes ou pour une maintenance éventuelle. Une identification exacte des pannes est généralement difficile à atteindre à cause de présence de bruit, perturbation de processus, modélisation des incertitudes, et en plus le chevauchement entre les sources potentielles des pannes par le système de surveillance [9].

## **4. Robustesse**

Un système de diagnostic efficace doit être robuste face aux bruits et aux incertitudes, car les performances doivent se dégrader au lieu de tomber directement en pannes. L'incertitude est inévitable dans chaque processus industriel, alors la robustesse en présence de bruit, perturbation de système et incertitude est l'un des plus importants paramètres désirés pour un système de diagnostic des pannes destiné pour une implémentation réelle.

## **5. Identification des nouveaux types de pannes**

Un outil industriel d'analyse des défaillances tel que AMDE (Analyse des modes de défaillances et leur effet) et son extension AMDEC (Analyse des modes de défaillances, leur effet et leur criticité) fournissent une masse importante d'informations sur les modes de défaillances, leur effet et impact sur le système ainsi que la probabilité de ces modes, où il ya la possibilité d'occurrence de nouvelles anomalies.

Le système de diagnostic des pannes doit être capable de détecter l'occurrence de nouvelle panne ce qui est traduite par la notion de rejet en distance.

## **6. Identification de pannes multiples**

Cette propriété reflète la capacité d'un système de diagnostic d'identifier et classer correctement des pannes multiples apparues au même temps dans le système. Plutôt cette identification multiple est un besoin difficile à réaliser à cause de non linéarités et interactions existantes entre les états et les sources potentielles des pannes de système dynamique.

## **7. Facilité d'explication**

Le système de diagnostic des pannes doit être capable d'expliquer l'origine de la panne et comment elle est produite dans le système. Cette justification aide l'opérateur pour évaluer et utiliser son expérience. Le système de diagnostic ne doit pas seulement justifier l'utilisation de quelques hypothèses, mais aussi justifier pourquoi il a négligé d'autres.

## **8. Adaptabilité**

Les conditions de fonctionnement de système se change à cause des perturbations externes et les changements de l'environnement. En outre, les performances des composants de système se dégradent par le temps. Alors le système de diagnostic doit s'adapter à ces changements afin de garder leurs performances de diagnostic.

## **9. Capacité de stockage et de calcul**

Les besoins de calcul et de stockage sont deux caractéristiques fondamentales pour n'importe quel algorithme réalisé pour une implémentation en temps réel. Durant la conception de système de diagnostic, il est nécessaire de prendre en considération le besoin en calcul et stockage comme des paramètres spécifique de l'application. En plus, selon l'application, un équilibre entre ces deux paramètres est nécessaire.

## **10. Estimation de l'erreur de classification**

L'estimation de l'erreur de classification est importante pour assurer la fiabilité de système de diagnostic. Cette propriété facilite le classement des pannes pour le système de diagnostic. Donc le calcul de l'erreur reflète le niveau de confiance des décisions de système de diagnostic et montre le degré de fiabilité de système [11].

## **11. Nécessité de modélisation**

La modélisation pour le développement de système de diagnostic est un besoin important. Pour une réalisation simple et rapide de système de diagnostic en temps réel, les efforts de modélisation doit être le minimum possible [11].

### 1.4 Méthodes de diagnostic

Plusieurs classifications des méthodes de diagnostic ont été proposées. Une classification proposée dans [11-13] basée sur le type de l'information introduite initialement dans le système du diagnostic. La figure 1.2 illustre cette classification en trois classes fondamentales .Zwinglestein propose dans son excellent ouvrage " Diagnostic des défaillances" deux classifications, la première est basée sur la modélisation complète des défaillances et des dégradations, c'est-à-dire l'existence d'un modèle liant les causes et leurs effets. La deuxième classification repose sur les connaissances des experts sans modélisation des mécanismes de propagation des dégradations.

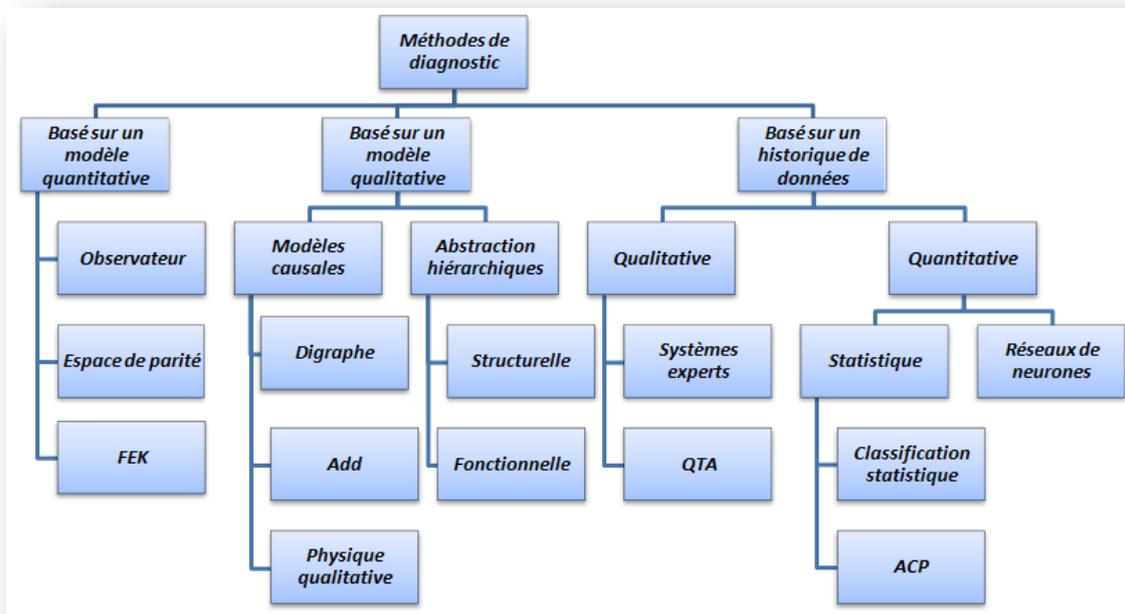


Figure 1.2 : Classification des méthodes de diagnostic

#### 1.4.1 Méthodes de diagnostic basées sur un modèle

Selon la description de processus industriel, on peut distinguer les méthodes de diagnostic par modélisation fonctionnelle reposant sur une description fonctionnelle, et les méthodes de diagnostic par modélisation matérielle basée sur une description matérielle de processus. Le paragraphe suivant présente les deux types de descriptions :

##### *Description fonctionnelle*

Pour décrire fonctionnellement un processus industriel complexe, trois concepts principaux généralement utilisés: les systèmes, les sous-systèmes et les composants. La

description fonctionnelle repose sur une décomposition en arborescence hiérarchisée à plusieurs niveaux. Un exemple explicatif montre une description fonctionnelle d'une machine à laver la vaisselle illustré sur la figure 1.3.

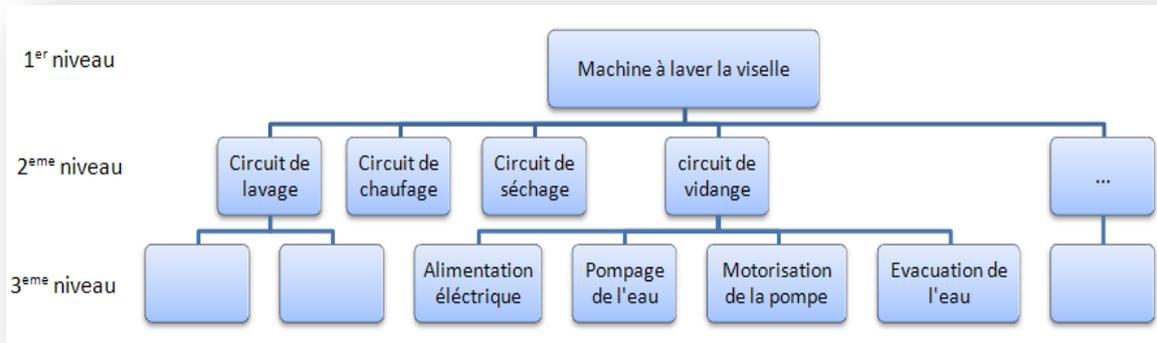


Figure 1.3 : Description fonctionnelle d'une machine à laver la vaisselle

### Description matérielle

Au contraire de la décomposition fonctionnelle, on s'intéresse ici à décrire les éléments constitutifs sans se préoccuper de leurs fonctions. La figure 1.4 représente la décomposition matérielle de la machine à laver la vaisselle. Généralement dans une description matérielle, on utilise les termes: pièces, organes, mécanismes, dispositifs, matériels et installations qui vont du plus petit élément (pièce) au plus important (installation).

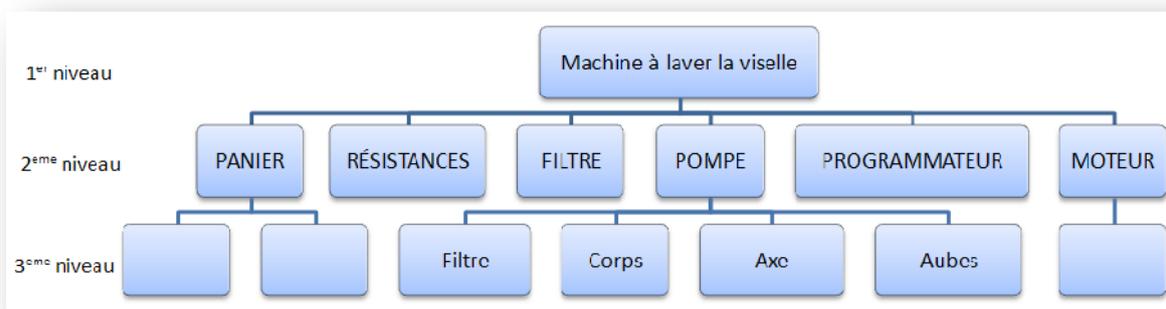


Figure 1.4 : Description matérielle d'une machine à laver la vaisselle

### 1.4.2 Méthodes d'analyse fonctionnelle

L'utilisation de ces méthodes pour décrire le fonctionnement d'un processus industrielle en cours de conception ou en fonctionnement est indispensable dans le monde industriel. Cette décomposition nous permet d'identifier les différents modes de défaillances et leurs conséquences sur les objectifs opérationnels retenus pour l'installation ou l'équipement

concerné. Généralement ces méthodes basées sur les techniques d'analyse de la valeur dédiées principalement aux nouveaux produits, développées par L.D. Miles, en 1947 [14]. Les étapes de cette méthode sont illustrées sur la figure 1.5 [8]. Plusieurs méthodes ont été développées en se basant sur cette technique (méthodes FAST, RELIASEP®, APTE®, SADT®...). On détaille dans ce qui suit quelques méthodes :

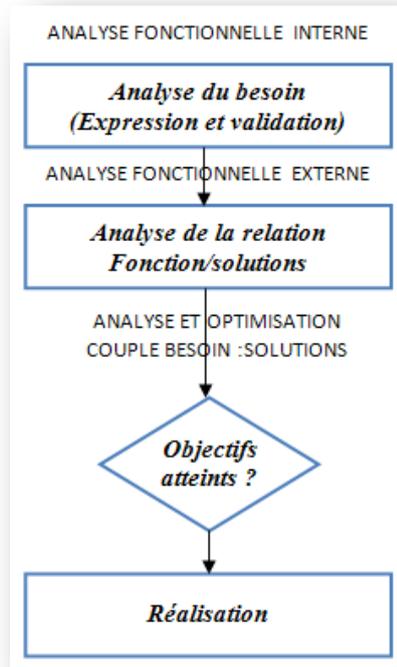


Figure 1.5 : Etapes d'analyse fonctionnelle

#### 1.4.2.1 Méthode SADT (structure analysis design technique)

SADT est une méthode ou bien un langage de modélisation graphique, développé pour être utilisé dans les études de fabrication assistée par ordinateur des forces aériennes américaines en 1970[15]. C'est une méthode de communication entre les spécialistes, elle permet l'analyse et la conception des systèmes importants et complexes. Elle est réalisée pour [14]:

- Conception structurée des systèmes vastes ou complexes ;
- Communication claire et précise des résultats de l'analyse et de la conception;
- Contrôle de l'exactitude, la cohérence et, de façon générale, la qualité de manière continue et systématique par des procédures particulières de critiques et d'approbations.

Le « langage » SADT est composé de diagrammes (actigrammes et datagrammes) (figure 1.6) obtenus par raffinements successifs et organisés en hiérarchie. Plus concrètement, il s'agit de boîtes et de flèches utilisées pour représenter

les activités et les données. *Les actigrammes* pour présenter la transformation des données d'entrées en données de sorties selon certaines contraintes en utilisant certains moyens ou supports (figure 1.7.a). *Les datagrammes* représentent des données générées par des fonctions en entrée et utilisées par des fonctions en sortie, sous la surveillance des activités de contrôle (figure 1.7.b) [14].

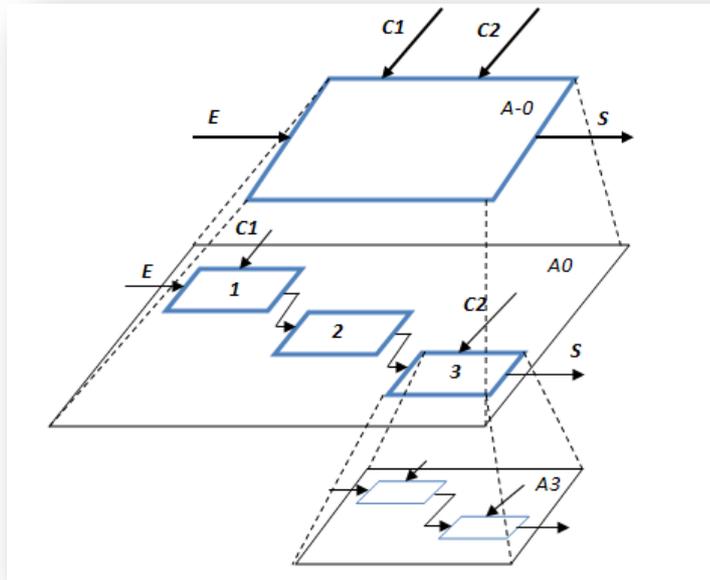
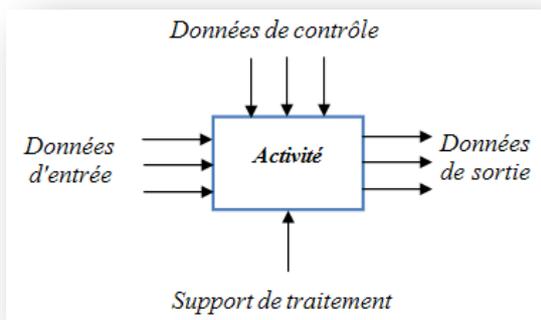
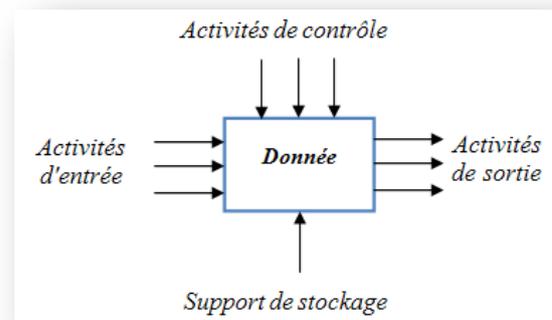


Figure 1.6 : Diagramme type SADT



(a)



(b)

Figure 1.7 : Boîtes SADT : (a) Actigramme, (b) Datagramme

#### 1.4.2.2 Les arbres de défaillances AdD

L'analyse par l'arbre de défaillance est une méthode inductive, qualitative ou quantitative, développé en 1961-1962 à la Société Bell Telephone par A. Watson (fault tree

analysis-FTA) [14]. Cette méthode consiste à identifier toutes les combinaisons d'événements possibles qui sont considérés ensemble comme la cause de la réalisation d'un événement indésirable. Cet événement indésirable est le sommet d'un arbre construit d'une manière descendante, englobant l'ensemble des combinaisons de défaillances des composants mises en jeu. Pour utiliser les arbres de défaillances à des fins de diagnostic, la méthode utilise un symbolisme spécifique et des règles précises [8]. Les Tableaux 1.1.a, 1.1.b, 1.1.c, 1.1.d illustrent l'ensemble de symbolismes utilisé pour la construction des arbres de défaillances [16].

**Tableau 1.1.a : Operateurs fondamentaux.**

<i>Symbole</i>	<i>Nom</i>	<i>signification</i>
	OU	La sortie est générée si l'un des deux entrées existe
	ET	La sortie est générée si toutes les entrées existent

**Tableau 1.1.b : Operateurs spéciales .**

<i>Symbole</i>	<i>Nom</i>	<i>signification</i>
	OU exclusif	La sortie est générée si un et seulement un des entrées existe
	Priorité ou Si séquentielle	La sortie est générée si toutes les entrées existent, avec un ordre d'apparence
	SI	La sortie est générée si l'entrée existe et la condition est vérifiée
	K-de-n	La sortie est générée si K de n entrées existent
	Matrice	La sortie est générée selon certaines combinaisons des entrées
	Délai	La sortie est générée après une durée $\Delta t$ pendant laquelle apparaissent les entrées
	Non	La sortie est générée si l'entrée n'est pas produite

**Tableau 1.1.c : Evènements.**

<i>Symbole</i>	<i>signification</i>
	<i>Rectangle</i> : Evènement de sommet ou intermédiaire
	<i>Cercle</i> : Evènement de base élémentaire
	<i>Losange</i> : Evènement de base non élémentaire
	<i>Losange double</i> : Evènement considéré comme un élément de base dans cette étape et analysé par la suite
	<i>Maison</i> : Evènement normal

**Tableau 1.1.d : Triangles de renvoi.**

<i>Symbole</i>	<i>signification</i>
	<i>Renvoi identique</i> Une partie de l'arbre non indiquée est identique à la partie motionnée par le dernier symbole
	<i>Renvoi similaire</i> Une partie de l'arbre non indiquée est similaire à la partie mentionnée par le dernier symbole
	<i>Indication de renvoi</i> Marque une partie d'arbre identique ou similaire à insérer dans un arbre inachevé

#### 1.4.2.2.1 Construction du l'arbre de défaillance

La construction d'un arbre de défaillance est le fruit d'une collaboration de différents spécialistes, qui interviennent pour la réalisation de système, à partir de concepteur jusqu'à l'opérateur de système. La construction de l'arbre passe par ces trois étapes [16]:

##### 1. Analyse préliminaire

(a) *Décomposition de système* : Cette étape consiste à la décomposition physique de système. Les critères utilisés généralement des critères de technologie, critères de la maintenance et critères de données. Bien sûr on peut avoir d'autres critères.

(b) *Identification des composants* : Cette étape concerne l'identification de tous les composants de base de système.

(c) *Définition des modes de défaillances des composants* : Pour chaque composant, les modes de défaillance possible doivent être décrits.

(d) *Reconstitution de système à partir des différents composants* : Il est nécessaire de reconstituer le système en se basant sur une description fonctionnelle, en inversant l'ordre de décomposition

## 2. Spécifications

(a) *Phases* : On appelle phases les différents modes de fonctionnement de système. Par exemple, pour un avion en vol, nous avons 3 phases (mode de fonctionnement) : décollage, vol en altitude et atterrissage.

(b) *Conditions des limites*: Concernent les interactions de système avec l'environnement.

(c) *Hypothèses spécifiques* : Concernent les relations dans le système lui-même.

(d) *Conditions initiales* : Concernent les hypothèses au début de chaque phase.

## 3. Construction

(a) *Définition des évènements indésirables* : Les évènements indésirables (évènement sommet) sont étudiés et définis Sans aucune ambiguïté en respectant les spécifications précédentes.

(b) *Résolution des évènements* : Elle est semblable à la détermination des évènements indésirables.

(c) *Fin de construction* : La construction de l'arbre est terminée quand toutes les causes des évènements non résolus sont atteints comme modes de défaillance des composants de l'environnement.

### 1.4.2.3 Analyse des modes de défaillances, de leurs effets et de leur criticité (AMDEC)

La méthode AMDEC, traduction de FMECA (Failure Mode, Effects, and Criticality Analysis), est une extension de la méthode AMDE. Ce type d'analyse a été développé pendant les années 60, dans l'aéronautique. Cette méthode considère la probabilité d'occurrence de chaque mode de défaillance ainsi que la gravité des effets associés pour réaliser une classification en basant sur l'échelle en criticité.

Différents types des modes de défaillances critiques peuvent être définis, où chaque société peut définir ses propres classes et catégories. Le tableau 1.2 donne une liste-guide des modes génériques de défaillance (norme AFNOR X 60-510) [14].

Tableau 1.2 : Liste de modes de défaillance suivant l'AFNOR

Modes génériques de défaillance	
1. Défaillance structurelle	17. Écoulement réduit
2. Blocage physique ou coincement	18. Mise en marche erronée
3. Vibrations	19. Ne s'arrête pas
4. Ne reste pas en position	20. Ne démarre pas
5. Ne s'ouvre pas	21. Ne commute pas
6. Ne se ferme pas	22. Fonctionnement prématuré
7. Défaillance en position ouverte	23. Fonctionnement après le délai prévu (retard)
8. Défaillance en position fermée	24. Entrée erronée (augmentation)
9. Fuite interne	25. Entrée erronée (diminution)
10. Fuite externe	26. Sortie erronée (augmentation)
11. Dépasse la limite supérieure tolérée	27. Sortie erronée (diminution)
12. Est au-dessous de la limite inférieure tolérée	28. Perte de l'entrée
13. Fonctionnement intempestif	29. Perte de la sortie
14. Fonctionnement intermittent	30. Court-circuit (électrique)
15. Fonctionnement irrégulier	31. Circuit ouvert (électrique)
16. Indication erronée	32. Fuite (électrique)
33. Autres conditions de défaillance exceptionnelles suivant les caractéristiques du système, les conditions de fonctionnement et les contraintes opérationnelles	

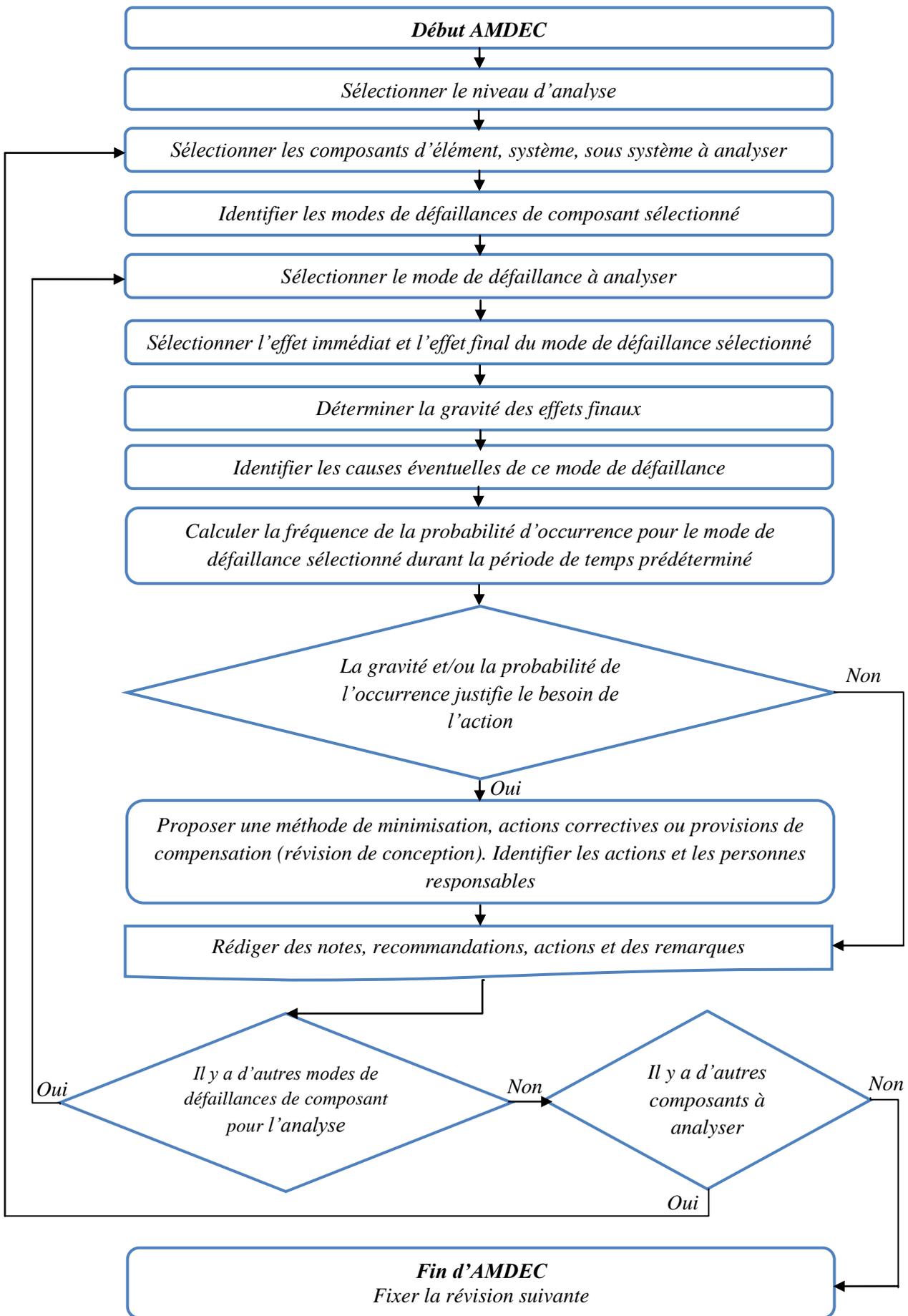
1.4.2.3.1 Construction des tableaux AMDEC

La méthode AMDEC est une méthode dure est lente, besoin d'une participation de tous les membres. Selon le secteur d'application, des tableaux sont définis pour la réalisation de l'AMDEC. La figure 1.8 décrit le contenu d'un tableau d'AMDEC suivant l'organisme CETIM [14].

AMDEC CETIM	ANALYSE DES MODES DE DEFAILLANCE DE LEURS EFFETS ET DE LEUR CRITICITE							AMDEC MACHINE				
	SYSTEME..... SOUS-SYSTEME.....			PHASE DE FONCTIONNEMENT :		DATE DE L4ANALYSE :		.....Page				
Élément	Fonction	Mode de défaillance	Cause	Effet	Détection	TI	Criticité				ACTION CORRECTIVE	
							F	G	N	C		

Figure 1.8 : Contenu d'un tableau d'AMDEC suivant l'organisme CETIM

La construction de ces tableaux passe par plusieurs étapes. L'organigramme 1.1 présente la réalisation d'AMDEC [17].



Organigramme 1.1 : Construction d'AMDEC selon IEC 60812

Le but de l'AMDEC est d'alimenter une banque de données, qui permet systématiquement par la suite une plus grande rapidité d'approches et de résolution, et l'utilisation de cette banque de données ne doit pas constituer un frein à la critique et à la créativité : une vérité n'est pas éternelle [6].

### ***1.4.3 Méthodes de diagnostic basées sur un modèle physique***

Contrairement aux méthodes de l'analyse fonctionnelle qui sont basées sur des analyses de sûreté de fonctionnement, les méthodes de diagnostic par modélisation physique qui proviennent des domaines de l'automatisation des procédés et de la simulation numérique sont élaborées à partir d'un modèle décrivant le processus étudié. L'ensemble des méthodes de diagnostic sont structurées en deux étapes. La première concerne le calcul de résidu, qui est la différence entre l'état actuel et l'état prévu, et la seconde étape s'intéresse au choix d'une règle de décision pour le diagnostic. Pour se faire, un mécanisme de redondance est nécessaire, soit d'une manière physique, on parle alors de redondance physique, ou bien d'une manière analytique, ce type de redondance nommé redondance analytique où le besoin des notions approfondies de mathématique est primordial [18].

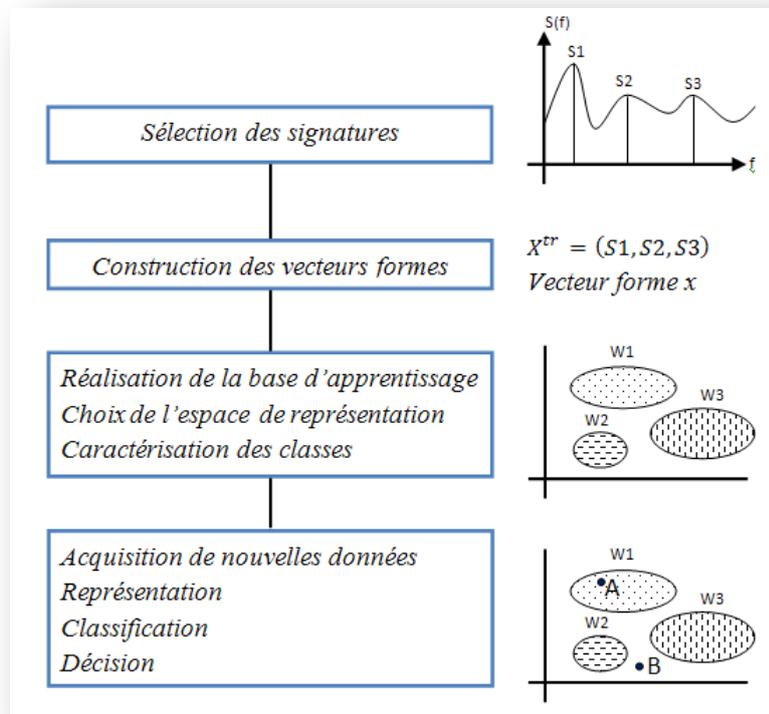
### ***1.4.4 Méthodes de diagnostic par analyse des signatures externes***

Ce type de méthodes est utilisé généralement dans le cas où il n'existe pas un modèle pour représenter les causes de défaillances, ou bien la modélisation ne présente pas d'intérêt économique compte tenu des enjeux recherchés. Généralement on distingue deux classes importantes, une classe basée sur des techniques statistiques appliquées aux données de retour d'expériences, la deuxième classe introduite en absence d'un volume important d'exemplaires, dans ce cas on parle de diagnostic externe basé sur l'expertise humaine [8]. Systèmes expert, réseaux de neurones, et la reconnaissance des formes sont les méthodes les plus utilisées.

#### ***1.4.4.1 Reconnaissance des formes***

La reconnaissance des formes est un outil d'intelligence artificielle, qui a, entre autres, une capacité d'apprentissage. Elle permet l'interprétation de nouvelles observations (formes) à partir d'un ensemble de données ou d'informations regroupées en classes, où la nouvelle observation est comparée pour être identifiée [18]. C'est-à-dire définir à quelle classe est rattachée une forme observée  $M$  parmi plusieurs classes  $W_1, W_2, \dots, W_n$  qui sont définies clairement a priori. Deux grandes familles des techniques de reconnaissance des formes sont connues, une technique qui ne prend en compte que des formes définies par des valeurs numériques, on appelle cette technique *la reconnaissance statistique*, et une seconde technique qui utilise des grammaires pour la représentation des formes, c'est la technique de

*reconnaissance structurelle* [8]. Le diagnostic par reconnaissance de forme met en œuvre quatre étapes principales qui sont résumées sur la figure 1.9.



**Figure 1.9 : Étapes de diagnostic par reconnaissance des formes** [8]

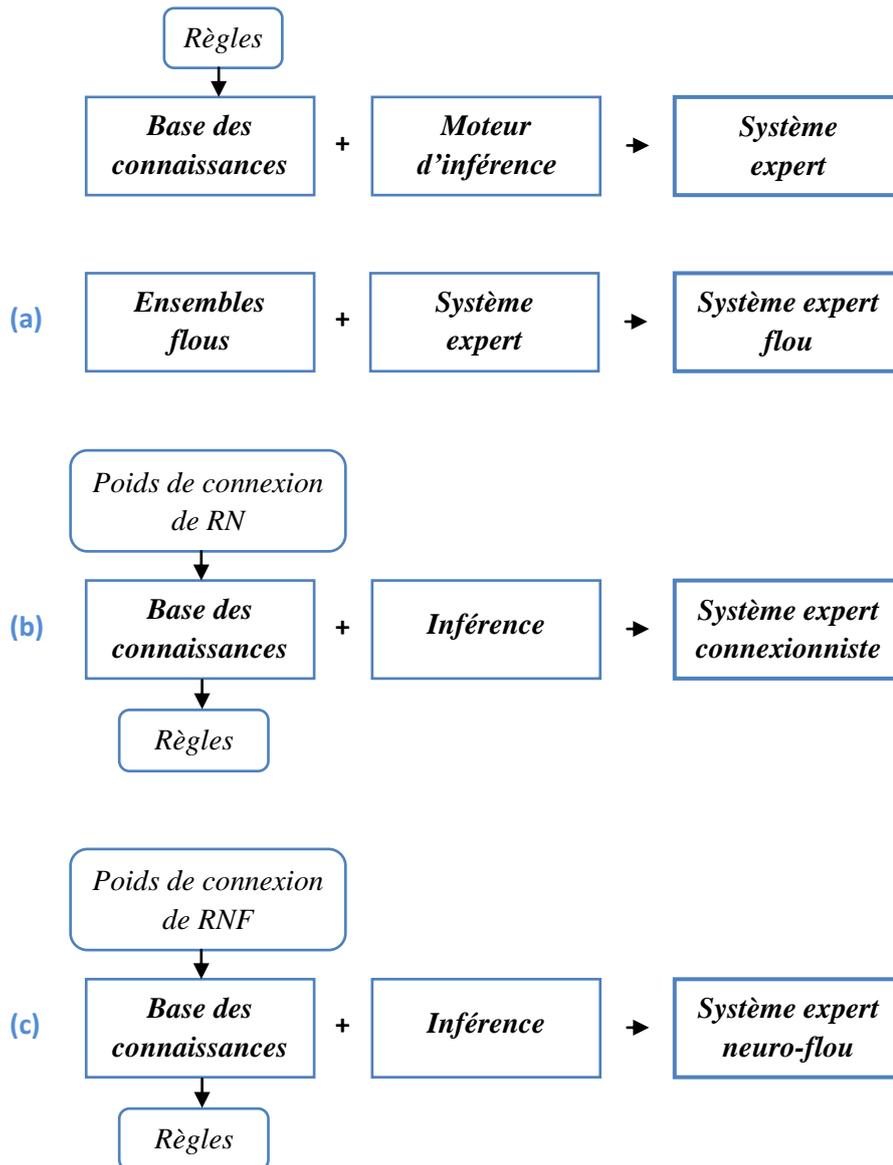
## 1.5 Méthodes de diagnostic hybrides

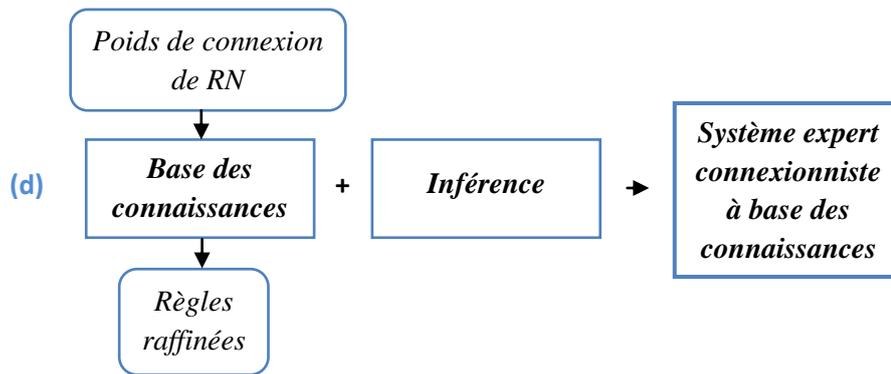
Le problème majeur des méthodes de diagnostic industriel est qu'il n'existe pas d'une seule méthode adéquate pour satisfaire tous les besoins d'un système de diagnostic. Bien que toutes les méthodes soient limitées, elles sont basées sur la qualité d'acquisition des connaissances, car certaines méthodes sont plus adaptées pour un type de données que d'autres. Plusieurs méthodes peuvent être complémentaires dans un système de diagnostic. Cette complémentarité donne la possibilité de combiner ces méthodes pour développer des méthodes hybrides afin de couvrir les limitations de l'utilisation individuelle des méthodes. Cette hybridation permet aux méthodes de coopérer pour la résolution de problème [13].

Plusieurs approches ont été proposées pour la combinaison des méthodes. On peut trouver dans [2-4] [19] des exemples d'intégration afin de profiter des points forts de chaque méthode. Pour bien comprendre l'utilité de cette hybridation, on présente dans ce qui suit quelques méthodes et techniques d'hybridation. On s'intéresse en plus par les systèmes experts hybrides.

### 1.5.1 Systèmes experts hybrides

L'introduction de la logique floue, des réseaux de neurones, du raisonnement probabiliste, des algorithmes génétiques ... etc. dans un système expert, forme ce qu'on appelle les systèmes experts hybrides. La liste suivante montre les différents types des systèmes experts hybrides ainsi que le niveau d'intégration des méthodes [20].





### 1.5.1.1 Système expert flou

La logique floue introduite dans les systèmes experts depuis plusieurs d'années, dans plusieurs domaines qui manipulent des informations imprécises et des données incertaines. Cette intégration est dans le but de résoudre les problèmes de limitation des systèmes experts classiques, en particulier celle concernant le raisonnement incertain [21].

Le système expert ne doit pas seulement trouver la solution, mais en plus décrire son niveau de certitude. Ce niveau de certitude est calculé par l'utilisation de l'imprécision et l'incertitude des faits et des règles appliquées. En plus le système expert doit s'adapter avec des informations incomplètes et non fiables. Plusieurs systèmes experts commerciaux actuels développent des outils basés sur différentes approches comme les facteurs d'incertitudes, bayésiens pour traiter l'incertitude des connaissances et des données [22]. Et d'autres introduisent la logique floue pour manipuler les données floues qui couvrent une partie significative de langage naturel. FuzzyClips est l'un des langages introduisant la logique floue, soit dans ses variables, faits, ou règles. La figure 1.10 illustre l'architecture d'un système expert flou. Pour plus de détail le lecteur est invité à consulter [22] et [23]

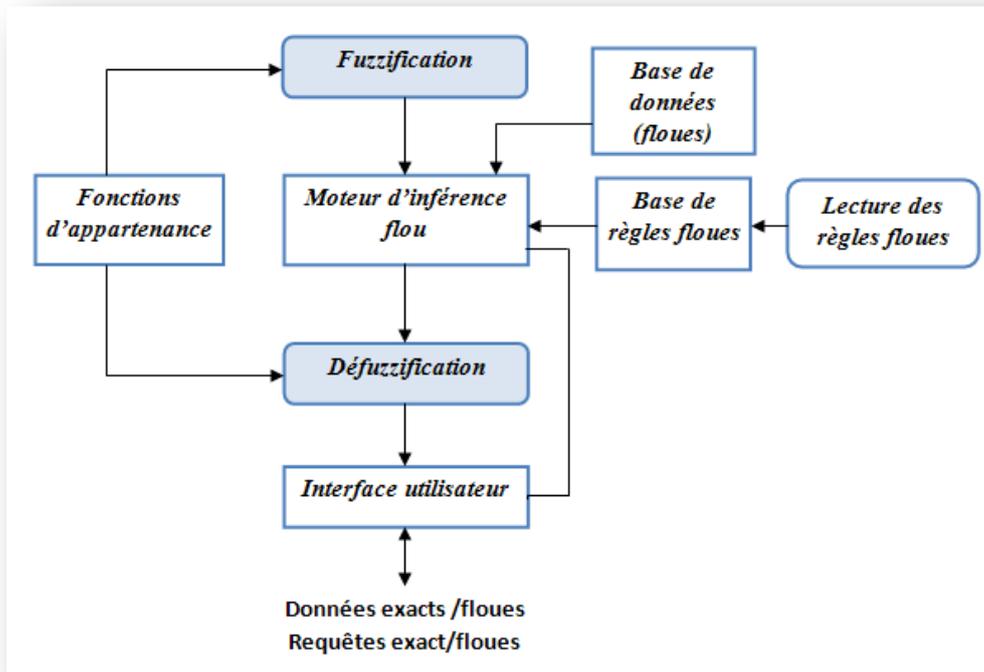


Figure 1.10 : Architecture d'un système expert flou

### 1.5.1.2 Système expert connexionniste

Les réseaux de neurones artificiels sont aussi utilisés dans la conception des systèmes experts. Ce type de systèmes est appelé *systèmes experts connexionnistes*, ils utilisent l'ensemble des poids de connexions de réseau de neurones pour la conception de la base de connaissances. L'utilisation des réseaux de neurones aide une intégration de parallélisme et l'optimisation des problèmes dans la base des connaissances. Ces modèles sont souvent appropriés pour les environnements riches d'informations et utiles pour aider à résoudre le problème de limitation d'acquisition des connaissances des systèmes experts traditionnels. Ils aident la minimisation des interactions humaines durant la phase de formation de la base des connaissances (où un excès de temps est consommé dans le cas des modèles traditionnels), et aussi ils réduisent la possibilité de générer des règles contradictoires. Remarquons aussi que dans les systèmes experts traditionnels la base des connaissances est construite en un ensemble des règles par interaction avec des experts. D'autre part, dans les systèmes experts connexionnistes, les règles sont extraites automatiquement des poids des connexions durant l'apprentissage de la base des connaissances [20].

Une architecture des systèmes experts connexionnistes donnée dans [24] est montrée sur la figure 1.11.

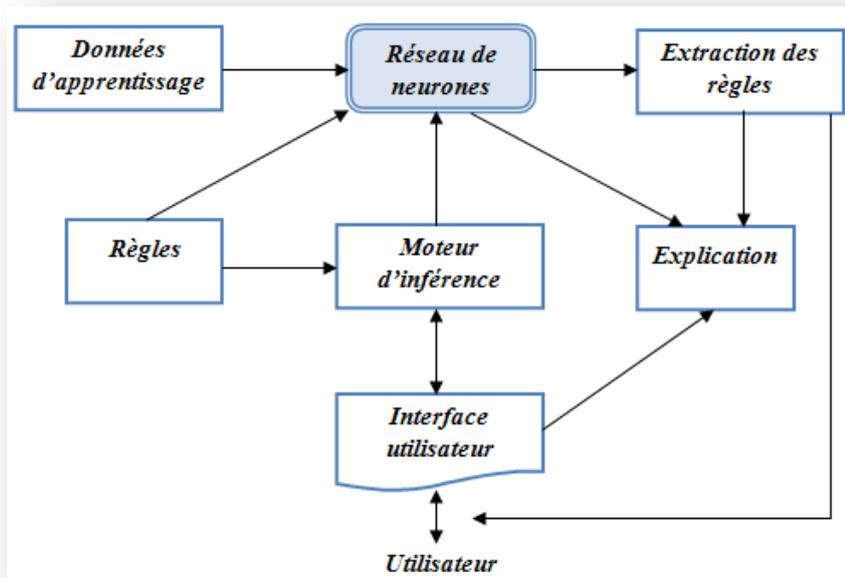


Figure 1.11 : Architecture d'un système expert connexionniste

### 1.5.1.3 Systèmes experts neuro-flou

Les systèmes experts neuro-flous utilisent les poids des connexions des réseaux de neurones flous pour la construction de la base des connaissances, et aussi pour intégrer les avantages de la théorie des ensembles flous dans les systèmes experts. En plus les avantages génériques des réseaux de neurones et les systèmes flous comme le parallélisme, la robustesse, l'adaptabilité, et la manipulation de l'incertitude peut être incorporés. Par exemple, la capacité des réseaux de neurones pour générer des régions de décision linéairement non séparable peut être exploitée. De plus la modélisation de l'incertitude dans les descriptions d'entrées et les décisions de sorties peut être réalisé par le concept des ensembles flous [20].

Une étude comparative entre quelques systèmes experts hybrides présentée dans [20] est donnée dans le tableau 1.3.

Tableau 1.3 : Etude comparative entre quelques systèmes experts hybrides

	Système expert	Système expert connexionniste	Système expert neuro-flou	Système expert à base de connaissances neuro-floue
<b>Base de connaissances</b>	Acquisition des connaissances et représentation sous forme de règles, réseaux sémantiques ou réseaux de croyance	Les poids de connexions de réseau de neurones où ils sont initialisés avec des petites valeurs aléatoires.	Les poids de connexions de réseau de neurones flou où ils sont initialisés avec des petites valeurs aléatoires.	Les poids de connexions de réseau de neurones traditionnels/flou où ils sont initialisés avec des connaissances approximatives dont la règle est formée avec des poids de connexions binaires, information de classe prioritaire et distribution des vecteur formes
<b>Raffinement des connaissances</b>	Addition des nouvelles connaissances. (comme des nouvelles règles)	Addition empirique des liens/nœuds cachés.	Addition empirique des liens/nœuds cachés.	Optimisation de réseau en réduisant les liens/nœuds, en utilisant les données d'apprentissage et des connaissances supplémentaires.
<b>Inférence</b>	Générer des faits à partir de la base des connaissances.	Présentation des entrées exactes, passage avant pour la génération des sorties exactes.	Présentation des entrées floues, passage avant pour la génération des sorties floues.	Présentation des entrées, passage avant et génération des sorties.
<b>Génération des règles</b>	-----	Règles exactes obtenus durant le passage arrière en utilisant des changements dans les niveaux des	Règles floues obtenus durant le passage arrière en utilisant les nœuds d'activations et	Règles obtenus durant le passage arrière ; Des règles négatives sont aussi possibles.

---

unités d'entrées et	les poids de
de sorties, et la	connexions.
grandeur des poids	
de connexions	

---

### 1.5.2 Réseaux de neurones flous

L'état de l'art des différentes techniques de combinaisons des réseaux de neurones et les ensembles flous, implique la synthèse à plusieurs niveaux. On peut distinguer les différentes méthodologies de combinaisons suivantes [20] :

1. *Incorporation de la fuzzification dans la structure de réseau de neurones*: implique la fuzzification des données d'entrées, attribution des notions floues aux échantillons d'apprentissage. En plus la possibilité de la fuzzification des procédures d'apprentissages et le réseau de neurones obtenu.
2. *Construction de réseau de neurones guidée par un formalisme de la logique floue*: Dans ce cas les réseaux de neurones sont construits pour implémenter la logique floue et la réalisation des décisions floues ainsi que la réalisation des fonctions d'appartenances représentant les ensembles flous.
3. *Changement des caractéristiques de base des neurones* : Les neurones sont construits pour réaliser les différentes opérations utilisées dans la théorie des ensembles flous (union floue, intersection, agrégation représenté par ET, OU, et opérateurs hybrides) au lieu des opérations d'addition et de multiplication standards.
4. *Réalisation d'un neurone flou personnel* : Les entrées et les sorties des neurones sont des ensembles flous, L'activité des réseaux en utilisant des neurones flous est aussi un processus flou.
5. *Utiliser les critères flous comme erreur d'instabilité de réseau* : Les mesures floues et incertaines des ensembles flous sont utilisées pour modéliser l'erreur d'instabilité ou la fonction d'activation du système basé sur les réseaux de neurones.

## 1.6 Conclusion

Dans ce chapitre nous avons présenté quelques méthodes de diagnostic industriel les plus utilisées et en particulier les méthodes hybrides, et nous avons donné une importance à l'une des techniques de diagnostic issue de l'intelligence artificiel: les systèmes experts hybrides. Les méthodes hybrides donnent des systèmes de diagnostic plus puissants, et montrent leur capacité de couvrir les limites et les anomalies des stratégies individuelles. Cette hybridation des méthodes reste un champ ouvert à cause de la diversité des systèmes industriels et des technologies utilisées. L'existence d'une méthode de diagnostic universelle est malheureusement impossible, alors la bonne compréhension et modélisation de problème en étude est indispensable pour ne pas tomber dans ce que on appel un problème mal posé, et en plus la participation de tous les membres de projet est importante afin d'arriver à un système d'aide au diagnostic efficace.

Malgré les progrès technologiques et le développement informatique, la décision humaine reste finalement indispensable, pour cela plusieurs techniques s'intéressent à modéliser et imiter le raisonnement humain, citons par exemple les systèmes experts et les réseaux de neurones qui font l'objet des deux chapitres suivants.

## CHAPITRE 2

# SYSTÈMES EXPERTS

**Résumé :** Ce Chapitre est réservé à la présentation de l'une des applications prometteuses de l'intelligence artificielle, les systèmes experts. Après une présentation des grandes lignes historiques de l'intelligence artificielle ainsi que celles des systèmes experts, nous nous intéressons à donner la définition de ces deux notions, l'intelligence artificielle et les systèmes experts, avec un aperçu de quelques domaines d'applications. Nous exposons ensuite l'architecture des systèmes experts et nous détaillons cette architecture dans la représentation des systèmes à règles de production. Avant d'aborder les systèmes à règles de production, nous illustrons le problème d'acquisition des connaissances et les techniques utilisées, que ce soit manuelles ou bien automatisées. Enfin, nous terminons par les étapes de développement d'un système expert.

## 2.1. Introduction

Depuis la découverte des ordinateurs, plusieurs chercheurs ont essayé de donner une notion d'intelligence aux programmes informatiques. Cela mène à l'émergence d'un nouveau concept appelé *intelligence artificielle* (IA). Au début, plusieurs critiques ont été posées concernant l'intelligence artificielle, ensuite et avec le temps, l'IA s'est investie dans nombreux domaines où l'informatique classique n'est pas applicable.

Les SE représentent une des applications prometteuses de l'intelligence artificielle. Ils sont utilisés dans plusieurs domaines d'activité humaines, par exemple: l'industrie [25], la médecine, l'agriculture, et le finance, où ils sont implémentés pour remplacer l'expert humain, pour être constituer un support de travail pour les utilisateurs du domaine et pourquoi pas un bon moyen pour les amateurs de devenir eux-mêmes des experts.

Généralement un système expert est composé de trois éléments principaux, une base des connaissances, un moteur d'inférence, et une base des faits qui est une mémoire de travail, où le développement de ces systèmes est basé sur la méthode d'acquisition et la représentation des connaissances. L'architecture la plus répandue est celle utilisant des règles de production dont la forme générale est proche de raisonnement humain : Si conditions alors conclusion.

Dans ce chapitre, nous commençons par un bref historique sur l'émergence de l'IA et en particulier les SE, ainsi que la définition de ces deux concepts et leurs domaines d'application. Nous citerons ensuite les éléments de base de l'architecture des SE. Nous nous intéressons par la suite au problème d'acquisition des connaissances et la représentation des systèmes à règles de production. Enfin nous terminons avec une définition des différentes étapes à suivre pour la mise en place d'un SE.

## 2.2. Historique

Au début des années 50, Herbert Simon, Allen Newell et Cliff Shaw ont essayé de simuler la pensée des êtres humains par plusieurs programmes. Cette expérience a facilité le lancement du premier programme de démonstration des théorèmes appelé "Logic Theorist". Dans la même période, Shannon et Paper ont proposé des stratégies plus robustes permettant aux ordinateurs de simuler une partie de jeu d'échecs.

Bien que ces travaux aient montré le concept des programmes intelligents, l'année 1956 est considérée comme l'année de départ de la notion de l'IA. Cette apparition du terme est concrétisée pour la première fois dans la conférence de l'IA organisée par John McCarthy, Marvin Minsky, Nathaniel Rochester, et Claude Shannon à l'université de Dartmouth de New Hampshire, où John McCarthy le développeur du langage de programmation LISP, a proposé le terme *Intelligence artificielle* [26].

Le professeur Peter Jackson de l'université d'Edinburgh classe l'histoire de l'intelligence artificielle en trois périodes [27]:

**Période classique:** Débutée en 1950, elle s'intéresse particulièrement à la démonstration des théorèmes et manipulation des jeux. Cette période est limitée par deux événements majeurs: en commençant par la publication de Shannon sur le jeu d'échec et en terminant par celle de Feigenbaum et Feldman. Le test de Turing est considéré comme un outil important pour vérifier l'intelligence de la machine développée dans cette période.

**Période romantique :** Allant du milieu des années 60 jusqu'au milieu des années 70. La plupart des chercheurs de cette période sont intéressés par l'apprentissage automatique du langage naturel. Parmi les prototypes développés dans cette période, on trouve le système SHRDLU, qui peut comprendre quelques mots de l'anglais appartenant à un domaine bien spécifique.

**Période moderne:** c'est la période dans laquelle la totalité des inventions informatiques de nos jours ont vu la lumière. La plupart des applications de cette période s'intéressent à la résolution des problèmes complexes, où les systèmes experts sont des candidats potentiels dans ces applications.

La commercialisation des systèmes experts est le fruit des travaux de développement et de recherche relative au cadre d'intelligence artificielle, dans ce genre d'applications l'idée d'une modélisation globale de l'ensemble de l'intelligence humaine est abandonnée et les approches de la résolution de problèmes très spécifiques sont considérées. Le premier système expert DENDRAL en analyse spectrographique est développé par Feigenbaum à la fin des

années 60, ce programme est mis au point pour simuler le travail d'un chimiste qui reconstitue la formule développée d'un composant organique à partir de sa formule brute et les résultats de sa spectrographie de masse. Ensuite le langage PROLOG de l'université d'Aix Marseille est apparu en 1975, dont la logique de premier ordre marque son départ. Ce langage est adopté comme langage de base par le projet japonais des ordinateurs de la 5<sup>ème</sup> génération [28].

Il existe plusieurs systèmes experts de différents domaines. Dans le domaine du diagnostic médical, le système MYCIN en 1976 pour diagnostiquer certaines maladies infectieuses, et en prospection géologique, le programme PROSPECTOR etc., sont développés en basent sur les concepts définis par Feigenbaum dans DENDRAL [26].

L'utilisation des systèmes experts a connue un essor accéléré à partir des années 80, ce qui est traduit par l'augmentation du nombre des conférences sur le sujet, ainsi que le développement des langages de programmation et outils de réalisation des systèmes experts, et en plus les techniques sont devenues de plus en plus maîtrisées [8].

### 2.3. Intelligence artificielle

*The scientific understanding of the mechanisms underlying thought and intelligent behavior and their embodiment in machines.*

Association américaine de l'intelligence artificielle

L'IA est un ensemble des méthodes, des outils, et des systèmes définis pour résoudre des problèmes dont leur solution nécessite de l'intelligence humaine. Le terme intelligence est défini toujours comme la capacité d'apprendre effectivement, de réagir efficacement, d'établir une bonne décision, de communiquer en langage ou bien en images d'une manière sophistiquée, et de comprendre. Alors l'IA s'intéresse à simuler l'être humain et en particulier le cerveau ainsi que sa manière de raisonnement [24].

Le sujet de l'IA couvre plusieurs domaines, il s'intéresse aux différents types de représentation des connaissances, différentes techniques d'intelligence, méthodes de résolution des problèmes avec des données ou connaissances incertaines, techniques d'automatisations pour l'apprentissage des machines...etc. Parmi les domaines d'application de l'IA, nous avons les systèmes experts, la manipulation des jeux, la démonstration des théorèmes, le traitement de langage naturel et la reconnaissance des formes, ...etc. L'IA est en interaction avec plusieurs sciences, la philosophie, la psychologie, les sciences cognitives, l'informatique, la mathématique et l'ingénierie. La figure 2.1 représente les différents domaines d'applications de l'IA, où les SE dominent la majorité des applications, ainsi que les sciences correspondantes [27].

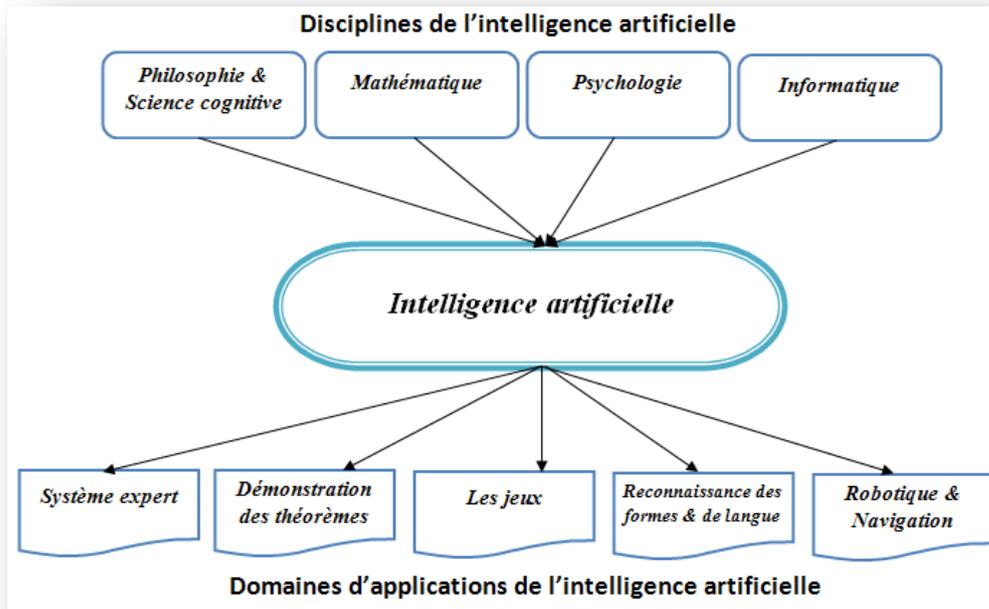


Figure 2.1 : Disciplines et domaines d'applications de l'IA

## 2.4. Systèmes experts

*"Expert system is an intelligent computer program that uses knowledge and inference procedures to solve problems that are difficult enough to require significant human expertise for their solution."* Edward Feigenbaum

Les SE sont des systèmes informatiques résolvant des problèmes pour lesquelles on ne dispose pas de solution algorithmique (dans le cas où les solutions algorithmiques sont à éviter à cause de l'explosion combinatoire).

Le professeur Edward Feigenbaum de l'université de Stanford, et un ancien pionnier des systèmes experts, a défini ce concept comme étant *"Un programme informatique intelligent utilisant des connaissances et des procédures d'inférences pour résoudre des problèmes assez difficiles ayant besoin d'une expertise humaine importante pour leur solution"*. Alors un système expert est un système informatique qu'*imite* la capacité d'un expert humain pour prendre une décision. Le terme *imite* signifie que le système expert tente en tous cas de raisonner comme un être humain. L'imitation est plus générale que la simulation, elle exige de procéder comme l'objet réel imité, bien sûr on respectant quelques aspects [29].

Les systèmes experts sont des systèmes basés sur les connaissances d'un expert humain. Par exemple un système expert de diagnostic des pannes d'une voiture possède une

base de connaissances contenant des règles pour le diagnostic et la détection des pannes de la même manière du raisonnement d'un mécanicien. Donc, le système expert est un programme qui peut fournir une expertise pour la résolution d'un problème défini dans le même domaine de l'expertise originale [24].

D'une manière générale, deux pôles distinguent un système expert, l'expert de domaine et les utilisateurs finaux (figure 2.2). L'expert transfère son expérience au système expert afin que les utilisateurs bénéficient de cette expertise en utilisant le système expert.



*Figure 2.2 : Pôles principaux de système expert*

### **2.4.1. Importance des systèmes experts**

La nécessité des systèmes experts revient aux limitations de processus décisionnel des experts humains, comprises:

1. L'expertise humaine généralement plus rare;
2. L'influence physique et morale sur l'être humain due aux charges du travail;
3. L'oubli des détails critiques du problème par l'être humain,
4. L'inconsistance des décisions humaines
5. L'homme ne peut pas assimiler une masse importante d'information rapidement;
6. L'homme ne peut pas mémoriser une masse importante de données;
7. la récupération de l'information stockée dans la mémoire par l'humain est lente.
8. L'homme est subjectif dans la prise de décision.
9. La responsabilité oblige l'homme à éviter la prise de décision;

En plus des limitations des êtres humains, les limites de la programmation classique et les outils d'aide à la décision traditionnels. Malgré la technologie puissante des ordinateurs, ils ont quelques limitations qui influent sur l'efficacité des programmes classiques imitant le processus décisionnels. Parmi ces limitations on a:

1. Dépendance des faits difficiles à acquérir.
2. incapacité d'appliquer les approches heuristiques utilisées par l'expert humain;
3. difficulté d'adaptabilité au changement de l'environnement de problème;

4. incapacité de donner des explications pratiques pour les solutions fournies.

### 2.4.2. Quelques domaines d'applications

Le développement de plusieurs prototypes de systèmes experts est illustré dans la littérature. La limitation détectée dans la présentation de quelques types seulement des applications des systèmes experts peut être expliquée par le fait que de nombreuses compagnies et organisations militaires n'ont pas présenté leurs systèmes experts à cause des secrets contenus dans les bases de connaissances. En se basant sur les systèmes décrits dans la littérature, on peut extraire quelques classes des systèmes experts qui sont donnés dans le tableau suivant [29]:

*Tableau 2.1 : Quelques domaines d'applications des systèmes experts.*

<b>Classe</b>	<b>Domaine d'application général</b>
<b>Configuration</b>	Assemblage propre des composants d'un système d'une manière correcte.
<b>Diagnostic</b>	Inférence des pannes en se basant sur les symptômes observés.
<b>Enseignement</b>	Enseignement intelligent où l'étudiant peut poser des questions de type Pourquoi?, Comment?, Quand? exactement comme en face d'un enseignant.
<b>Interprétations</b>	Explication de données observées.
<b>Surveillance</b>	Comparer les données observées avec les données désirées pour ajuster les performances.
<b>Planification</b>	Partager les actions pour l'obtention des résultats désirés.
<b>Pronostic</b>	Prédire les résultats d'une situation donnée.
<b>Maintenance</b>	Ordonner un traitement pour un problème spécifique.
<b>Contrôle</b>	Régulation d'un processus: faire une interprétation, diagnostic, surveillance, planification, pronostic et maintenance.

## 2.5. Architecture d'un système expert

Les décisions complexes nécessitent une combinaison compliquée de connaissances factuelles et heuristiques. Pour qu'un ordinateur peut extraire et utilise des connaissances

heuristiques, ces connaissances doivent être organisées d'une manière accessible et séparée entre les données, les connaissances et les structures de contrôles.

Pour ces raisons, les systèmes experts sont organisés en trois niveaux distincts [30]:

1. **Base de connaissances** contient les règles de résolution de problème, les procédures et les méthodes de recherche sur la solution (métarègles), et des données intrinsèques concernant le domaine d'expertise en question. Alors la base de connaissances fait le rôle d'une mémoire d'un expert de domaine. Bien sûr ces connaissances ne sont pas introduites dans la mémoire de l'ordinateur à l'état brut, il est nécessaire de choisir un formalisme susceptible d'être transposé dans un système informatique, c'est un langage de représentation des connaissances [28].

La base de connaissance est le noyau de système expert et elle diffère de la base de données classique. Dans les bases de données, on trouve des relations statiques entre les éléments de la base. La base de connaissance est créée par un cogniticien (ingénieur de connaissances). Ce dernier traduit les connaissances de l'expert humain en règles et stratégies.

2. **Mémoire du travail** contient les faits et les tâches spécifiques de résolution de problème en considération, appelée aussi module de base de données [24]. c'est un espace de travail qui contient momentanément les règles d'inférences utilisées, les faits utilisés et les nouveaux faits inférés ainsi que toutes les données utilisées. Cet espace de travail joue le rôle d'une mémoire RAM dans un ordinateur. En plus cette base de faits ou bien la mémoire de travail est un aide pour le module d'explication défini ci-dessous.
3. **Moteur d'inférence** c'est un mécanisme générique de contrôle, il applique les règles de la base de connaissance sur les données des tâches spécifiques pour arriver à des solutions ou des conclusions. Le moteur d'inférence est un mécanisme de contrôle qui organise les données de problème et cherche dans la base de connaissance les règles applicables pour ces données. A cause de l'augmentation de la popularité des systèmes experts, plusieurs moteurs d'inférences commerciaux sont apparus.

La figure suivante représente l'architecture d'un système expert :

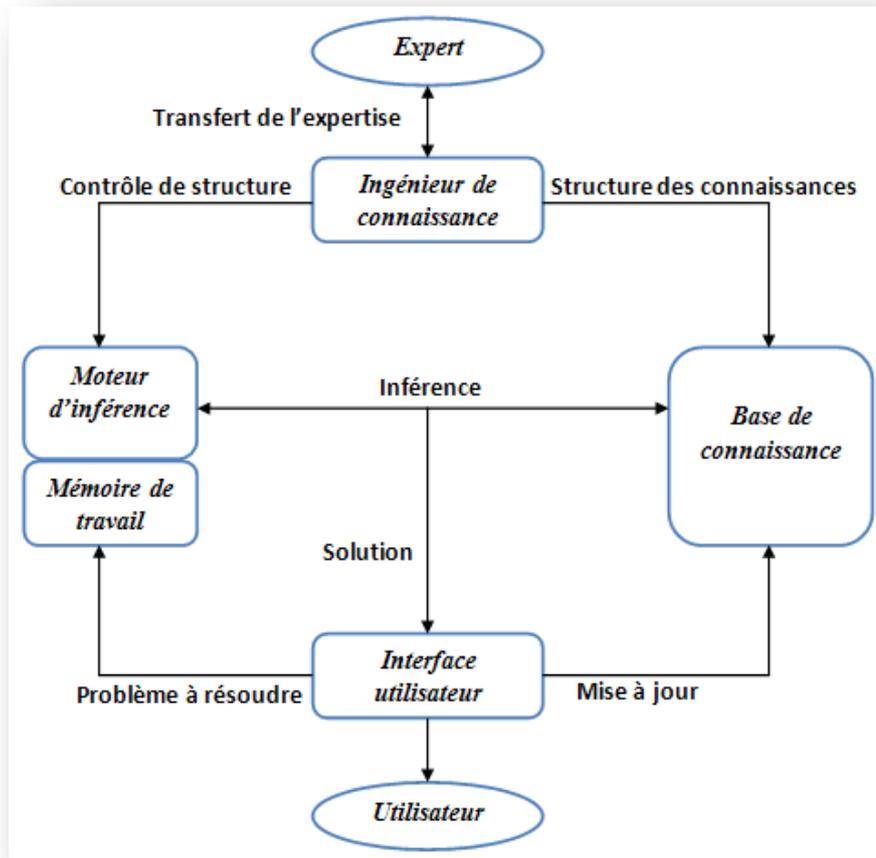


Figure 2.3 : Architecture d'un système expert

En plus de ces trois composants de base on peut trouver les modules suivants : [24] :

- **Module d'acquisition des connaissances** comme son nom l'indique, il est utilisé pour l'acquisition des données afin de remplir la base de connaissance.
- **Module d'interface utilisateur** c'est un module de communication avec l'utilisateur d'une manière facile et flexible. La communication en langage naturel peut être utilisée durant les explications.
- **Module d'explication** c'est un module qui sauvegarde les traces de l'exécution de système expert ainsi que toutes les informations du déroulement de processus de raisonnement afin de fournir ces informations aux utilisateurs de système expert. Alors le système expert doit être capable d'expliquer le raisonnement, par exemple, l'inférence de quelques conclusions durant le processus de déduction

## 2.6. Acquisition des connaissances

L'acquisition des connaissances est l'une des étapes importantes pour le développement d'un système expert. C'est l'étape qui aide le cognicien pour créer la base de

connaissances. La réussite d'un système expert dépend de la qualité des connaissances acquises, ainsi que de leur bonne représentation dans la base [30].

Le processus d'acquisition des connaissances se fait soit manuellement ou bien d'une manière automatique. En mode manuel, Le cogniticien reçoit les connaissances à partir d'un ou plusieurs experts de domaines. En mode automatisé, un système d'apprentissage automatique est utilisé pour l'acquisition et la construction de la base. L'un des grands problèmes d'acquisition des connaissances d'une manière manuelle est que l'expert souvent n'arrive pas à bien transférer son savoir-faire malgré qu'il peut facilement résoudre un grand problème de son domaine [27].

### ***2.6.1. Ingénieur de connaissances***

L'ingénieur de connaissances ou bien le cogniticien est un ingénieur qui prend en charge la modélisation des connaissances et le raisonnement d'un expert humain sous une forme manipulable par le programme informatique. Il est préférable pour un bon cogniticien de satisfaire les caractéristiques désirées suivantes:

- Patience;
- Persévérance (volonté);
- Attention;
- Curiosité;
- Bonne compréhension;
- Crédibilité technique;
- Bonne motivation;
- Bonne organisation;
- Savoir de communication.

### ***2.6.2. Techniques d'acquisition de connaissances***

Plusieurs techniques sont utilisées pour l'acquisition des connaissances parmi lesquelles on peut citer [30]

#### ***2.6.2.1. Techniques d'acquisition manuelles***

##### ***1. Interviews***

L'interview est la méthode la plus utilisée pour l'acquisition des connaissances de l'expert. Avec une méthode non structurée, le cogniticien discute avec l'expert de domaine leur raisonnement face à un problème. L'expert peut expliquer le processus de raisonnement soit verbalement ou bien en combinaison avec l'exécution de la tâche. L'ingénieur de connaissance sauvegarde toutes les informations nécessaires et pose des questions pour

rassembler plus d'informations concernant les approches suivies par l'expert pour résoudre le problème.

L'interview non structurée au début de la phase d'acquisition est important afin d'acquérir une masse importante d'informations. Ensuite une interview structurée est utilisée pour extraire des informations spécifiques concernant les techniques particulières de raisonnement de l'expert.

### ***2. Interprétation de la tâche et de protocoles***

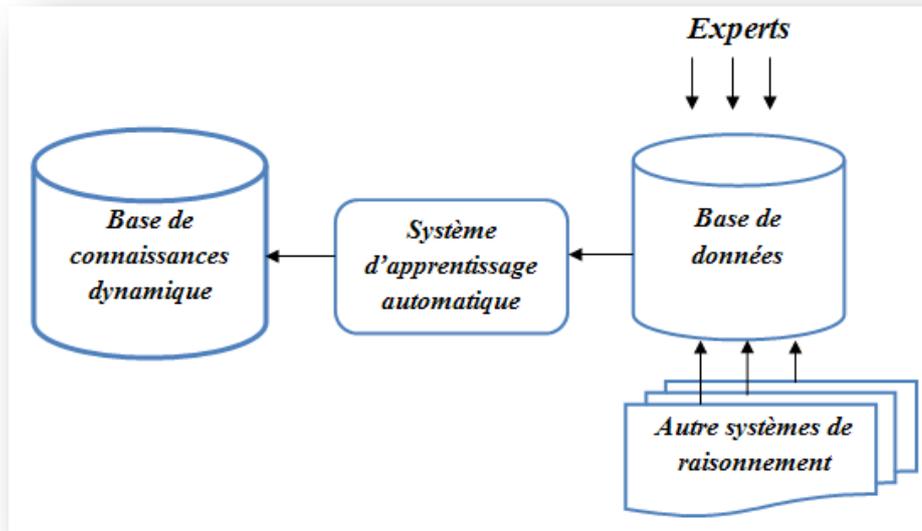
L'observation de l'expert humain dans sa tâche de résolution de problème peut être productive pour acquérir des informations. Les tâches préliminaires observées doivent être habituelles et simples pour que le cognicien prenne une vision globale sur le raisonnement de l'expert. En plus, il faut donner le courage à l'expert pour penser à haute voix face à un problème. Le processus doit être enregistré en audio ou bien en vidéo pour obtenir une vision correcte du raisonnement de l'expert. L'enregistrement peut être analysé ensuite par l'ingénieur de connaissance. L'ingénieur peut demander à l'expert de refaire la tâche pour poser quelques remarques et ajouter des commentaires. Dans cette méthode, l'étude des actions de l'expert appelée aussi analyse des contrôles.

### ***3. Brainstorming***

Brainstorming est une méthode de génération de nouvelles idées. Dans le brainstorming, une équipe des experts se regroupe pour discuter ensemble les alternatives de résolution des problèmes de décision. Les membres de discussions peuvent être de différents services, chacun à son expertise et son raisonnement. Cette combinaison aide à créer un espace de discussion pour générer des nouvelles idées. Un chef d'équipe est nécessaire pour la direction des discussions. En plus, il guide la procédure d'établissement des idées, et demande la participation de tous les membres. Après la présentation des idées, une discussion est ouverte pour accepter, refuser ou ajouter des idées.

#### ***2.6.2.2. Acquisition automatisée***

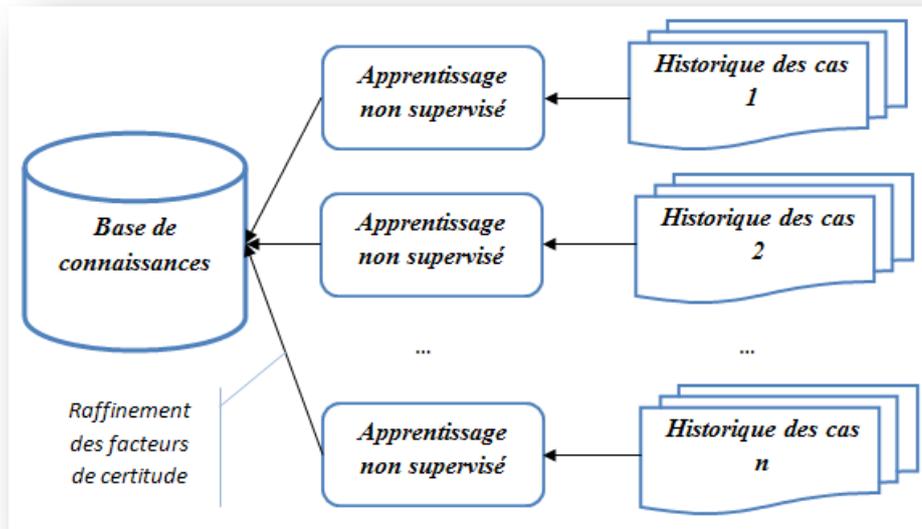
L'acquisition manuelle est difficile à cause des deux raisons. Premièrement, le cognicien doit rester en contact avec l'expert pendant une période considérable, qui peut être hors les heures de travail. Deuxièmement, l'expert lui-même dans plusieurs cas n'arrive pas à bien présenter les connaissances. Cependant, ces deux difficultés pour l'acquisition des connaissances peuvent être la raison pour l'automatisation de l'acquisition des connaissances. La figure 2.4 illustre l'obtention des connaissances d'une manière automatique [27].



**Figure 2.4 : Acquisition automatique des connaissances**

La base de données de la figure 3.4 est construite à l'aide des experts ou d'autres systèmes de raisonnement. L'approche d'apprentissage utilise ces données pour construire des nouvelles connaissances. Les connaissances acquises sont transférées vers la base de connaissance pour une utilisation future et la base est appelée *base de connaissances dynamique*.

Dans quelques systèmes, il est nécessaire de raffiner la base de connaissance pour respecter les paramètres définis pour la base. En plus, les paramètres des règles de la base de connaissances sont redéfinis en prenant en compte les facteurs de certitude estimés de l'historique des cas. Cette étape de raffinement de la base de connaissance est présentée sur la figure 2.5.



**Figure 2.5 : Raffinement des paramètres de base de connaissance**

La figure 2.5 présente une méthode pour l'estimation automatique de quelques paramètres dans un système expert. Le facteur de certitude de connaissance est raffiné selon les valeurs initiales et l'inférence de l'état progressif des  $n$  cas historiques. Dans le cas où les  $n$  problèmes sont similaires, il est préférable que quelques règles de connaissance appartiennent à deux ou plusieurs bases de connaissances. Donc le facteur de certitude d'une règle commune peut avoir des valeurs différentes dépendamment de la base de connaissance choisie.

L'algorithme d'apprentissage non supervisé présenté dans la figure 2.5 diffère d'un système à un autre selon le type de représentation des connaissances utilisée ainsi que la technique d'inférence. Par exemple, si les réseaux bayésiens sont utilisés pour représenter la base de connaissances, le réseau de Hopfield peut être une solution pour l'apprentissage. Si la base de connaissances est réalisée par les réseaux de Petri, l'apprentissage de Hebb peut être adopté pour calculer les facteurs de certitude, présentés par les poids de réseau.

## 2.7. Systèmes à règles de production

Les systèmes à règles de production sont les plus simples et parmi les premières approches de représentation des connaissances. Un système à règle de production est basé sur trois éléments, *les règles de productions* incluses dans la base des connaissances, un ou plusieurs bases de données dynamiques appelés espaces de travail ou bien *bases des faits*, et un mécanisme de contrôle qui joue le rôle d'interpréteur appelé généralement *moteur d'inférence*, il applique les règles de productions de la base de connaissances sur la base des faits pour inférer des nouveaux faits[27].

### 2.7.1. Règles de production

Les règles de production sont étroitement liées au développement de l'informatique, construites d'une simple instruction " if ...then..." employés dans la plupart des langages de programmation allant de BASIC, FORTRAN et C jusqu'aux systèmes complexes de traitement de règles comme Prolog et la version floue associées [23]. Les règles de production sont le mécanisme le plus utilisé pour la représentation de certaines connaissances, une règle de production est une expression de la forme:

**Si** Conditions **Alors** Conclusion

Un exemple typique d'une règle de production:

*Si le moteur ne démarre pas ;  
et les lampes ne s'allument pas ;  
Alors problème de la batterie.*

### 2.7.2. Base de faits

La mémoire de travail ou la base de faits contient des faits qui ont en général une forme de triplet OAV <Objet Attribut Valeur>. Les parties *conditions* et *conclusion* de la règle de production ont la même forme que les faits. Si les conditions d'une règle sont vérifiées alors la règle est exécutée et toutes les conclusions de la règle sont ajoutées comme faits dans la mémoire de travail. Dans d'autres systèmes de production, La partie *Conclusion* indique quels sont les faits à ajouter ou bien à supprimer de la base des faits. Généralement, des nouveaux faits sont ajoutés dans la mémoire de travail et d'autre sont supprimés s'ils ne sont pas utilisés pendant une durée de temps définie afin de minimiser le temps de recherche des conditions vérifiées dans les règles [27].

### 2.7.3. Moteur d'inférence

Le moteur d'inférence ou bien l'interpréteur dans les systèmes à règles de production passe par trois étapes, appelées cycle de base d'un moteur d'inférence [27-28] qui est décrit par la suite.

#### 2.7.3.1. Cycle de base d'un moteur d'inférence

L'exécution d'une instruction passe généralement par trois étapes:

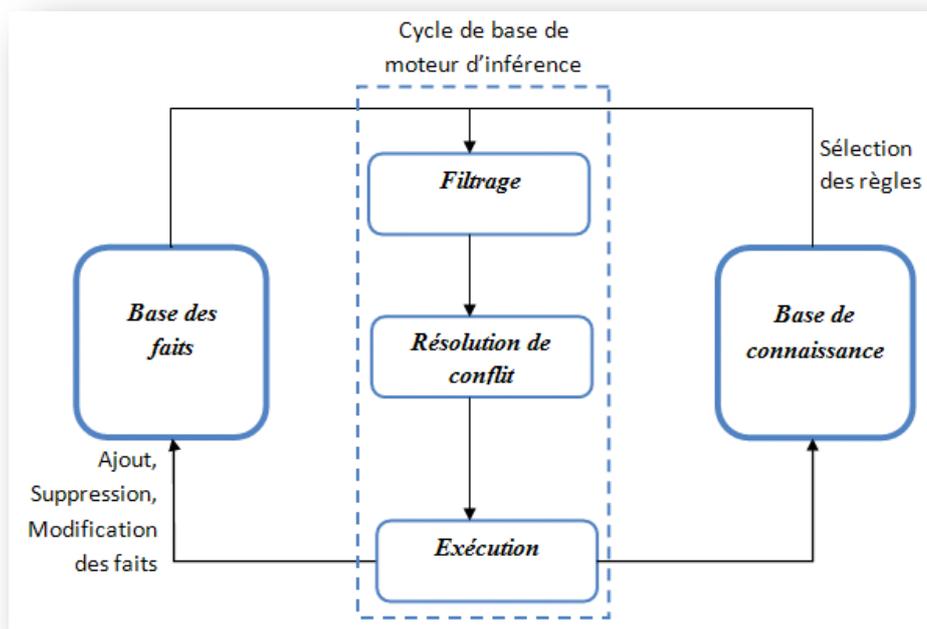
1. **Etape de filtrage** : Dans cette étape on affecte à chaque variable de la partie *Conditions* de la règle les valeurs des faits appropriés enregistrés dans la base des faits, Un

ensemble des règles pouvant être déclenché ou celles qui vérifient un filtre particulier défini dans le système est formé.

2. **Etape de résolution de conflits** S'il y a plus d'une règle admise à déclencher, on va choisir la règle à activer suivant la stratégie de résolution des conflits, Cette stratégie peut être simple (la première règle de la liste, la moins complexe, la moins utilisée ...) ou complexe (la plus prometteuse, la plus fiable, la moins coûteuse...).
3. **Etape d'exécution** : Après le déclenchement de la règle, des nouveaux faits résultants de la règle déclenchée sont insérés dans la base des faits, et les faits non utilisés pour un laps de temps fixe sont annulées de la base et on revient à la première étape.

Généralement, les faits initiaux sont enregistrés dans la base des faits au début d'exécution de programme pour être insérés dans le cycle de base du moteur d'inférence. Le processus d'exécution détermine s'il n ya pas des règles à déclencher ou bien la règle déclenchée contient une commande explicite d'arrêt.

La figure 2.6 illustre l'architecture de système à règles de production ainsi que le cycle de base du moteur d'inférence.



**Figure 2.6 : Architecture d'un système à règles de production**

### 2.7.3.2. Stratégies de résolution de conflit

La stratégie de résolution de conflit diffère d'un système à un autre. Plusieurs systèmes utilisent un, ou bien une combinaison des stratégies suivantes pour résoudre le problème de conflit dans le moteur d'inférence [27]:

### 1. *Réflexion / Refractoriness*

Cette stratégie exige que la même règle ne se déclenche qu'une seule fois pour le même ensemble des faits. Elle est utilisée pour éviter les surcharges dans la base des faits. Une deuxième version de cette stratégie consiste à supprimer de la base les règles utilisées dans le cycle d'exécution précédent. En effet cette stratégie aide le système pour ne pas entrer dans une boucle.

### 2. *Révision / Recency*

Cette stratégie exige que les éléments les plus récents dans la base soient utilisés pour déclencher la règle.

### 3. *Spécificité / Specificity*

Dans cette stratégie la règle qui possède un grand nombre de conditions est déclenchée avant les règles ayant moins des conditions.

#### 2.7.3.3. *Modes de raisonnement*

Généralement deux modes de raisonnement sont utilisés, Le *chaînage avant* et le *chainage arrière*, et un troisième qui combine les deux modes appelé le *chainage mixte*.

- *Chainage avant*

La méthode basée sur la vérification de la partie des conditions de la règle et détermine est ce qu'ils sont vrais ou faux. Si les conditions sont vraies alors les conclusions sont ajoutées dans la base des faits comme vrais. Cette procédure continue jusqu'à trouver une solution ou aucune règle ne peut être appliquée. Le chaînage avant est souvent référé comme un raisonnement basé sur les données.

- *Chainage arrière*

A l'inverse de chainage avant, le chainage arrière est utilisé pour propager des conclusions vers les conditions de la règle. Le chainage arrière est très utile dans le cas où les résultats (but) sont connus et ne sont pas très nombreux. Dans ce cas, le but est spécifié et le moteur d'inférence cherche à trouver les conditions nécessaires pour arriver à ce but [23].

- *Chainage mixte*

Le chainage mixte combine entre l'utilisation des chainages avant et arrière, les deux chainages utilisés ensemble pour la résolution automatique des problèmes [27].

#### 2.7.3.4. *Ordre d'un système expert à règle de production*

L'ordre d'un système revient à la complexité des expressions figurant dans les règles. *Conditions* et *Conclusions* d'une règle sont exprimées par des prédicats. Un Prédicat est une

expression utilisant éventuellement des variables, pouvant prendre des valeurs logiques Vrai ou Faux. On peut distinguer les ordres suivant d'un système expert [8]:

- **Ordre 0** Dans ce type des systèmes à règle de production les expressions Conditions et Conclusions d'une règle sont exprimées en prédicats sans variable c'est-à-dire avec des paramètres constants.
- **Ordre 0+** Dans ce type de systèmes, certains éléments ne sont plus figés mais peuvent varier dans un intervalle précis. Ils intègrent aussi les évaluations symboliques (<, >, =) et les nombres réels.
- **Ordre 1** Les systèmes à règles de production d'ordre 1 sont des systèmes dont leurs règles peuvent contenir des variables.

## 2.8. Développement d'un système expert

Le développement d'un système expert passe par un cycle de vie allant de l'expression des besoins à l'exploitation et l'utilisation d'un produit final. Cependant, une plus grande importance est accordée à la première étape de modélisation et d'analyse. Un maquetage avant la mise en place d'un logiciel cohérent nécessite en préférence un modèle de développement en spirale tel que celui proposé par B.Boehm sous forme de succession d'étapes (expression de besoins, analyse de risques, choix de solutions, maquetage, test de la maquette et extension) au lieu de développement linéaire (expression de besoins, conception générale, conception détaillée, codage, test unitaires, test d'intégration) [8].

### 2.8.1. Analyse des connaissances

Pendant cette étape, le cogniticien joue un rôle indispensable pour planifier le déroulement de l'acquisition des connaissances. Il est préférable aussi que l'expert s'initie à ce que sont les systèmes experts afin de comprendre le type et la forme de connaissances demandées par le cogniticien.

Afin d'élaborer un modèle d'acquisition des connaissances, des interviews sont nécessaires avec l'expert. Au début, le cogniticien cherche à faire une description du domaine en étude et une description de savoir de l'expert ainsi que son savoir-faire, ensuite il s'intéresse à traduire les formules pour la représentation de ces connaissances selon la technologie disponible. Enfin, le modèle final élaboré est utilisé pour l'acquisition des connaissances nécessaires.

### ***2.8.2. Maquettage et prototypage***

Dans cette étape, c'est le rôle de l'équipe informatique pour coder et implémenter le modèle élaboré, bien sûr après une sélection de l'outil logiciel adapté. Lors de cette étape, le rôle général de cognitif est important pour vérifier et valider les différents éléments de modélisation.

Une fois la maquette est élaborée après plusieurs ajustements et sa base de connaissances est complétée, un prototype du système expert est constitué pour être utilisé.

### ***2.8.3. Test et validation***

Comme pour toute application, l'étape de test et validation est nécessaire pour la vérification de prototype du système élaboré. En outre de la validation ordinaire de côté logiciel de système expert, on doit inclure toutes les actions nécessaires pour s'assurer de la qualité de la base de connaissances constituée, qui est la base de la réussite du système expert.

Dans cette étape, on remarque bien le rôle de la modélisation et l'acquisition des connaissances, alors l'étape d'analyse des connaissances influe sur le déroulement de l'étape de test et validation.

### ***2.8.4. Evolution et maintenance***

L'évolution du matériel sur lequel porte le système expert exige l'évolution de ce dernier et impose certaines corrections et adaptations. La maintenance reste une tâche indispensable dans tout le cycle de vie de système expert.

## 2.9. Conclusion

Les systèmes experts sont des outils très importants, qui offrent un bon environnement pour coder, enregistrer et implémenter l'expertise humaine, où dans plusieurs domaines cette expertise est souvent très rare, en bénéficiant du développement progressif de la technologie. Ceci dans le but de remplacer les limites des experts humains, comme la fatigue, l'oubli et le décès qui constituer une grande perte d'expertise.

Généralement les problèmes de natures heuristiques, besoins d'intuition, de jugement, et de raisonnement logique sont les plus candidats pour les systèmes experts. Au contraire, des applications de nature déterministe où les procédures algorithmiques sont les plus adéquates.

Le développement d'un système expert est une procédure longue et dure. La bonne mise en place de tels systèmes basés sur la bonne construction de leur base de connaissances, où l'ingénieur de connaissances joue un rôle indispensable dans l'acquisition de connaissance en plus de la construction de la base, et la qualité des connaissances acquises. Alors, un mauvais choix de représentation des connaissances, même face à un bon moteur d'inférence et une interface utilisateur attirante, influe sur la fiabilité des solutions des systèmes experts.

Le problème qui reste d'actualité est le problème d'acquisition des connaissances soit d'une manière manuelle où le cognaticien souffre pour extraire les connaissances de l'expert, ou bien d'une manière automatique où le besoin des techniques approfondies d'IA. En plus de problème d'acquisition des connaissances, le choix et le filtrage des connaissances significatives sélectionnées effectivement pour la base de connaissances est aussi une tâche difficile et indispensable pour la réalisation d'une base des connaissances.

En ce sens le chapitre suivant sera consacré à la présentation des notions de base de l'un des techniques utilisé pour l'acquisition des connaissances : Réseaux de neurones artificiels.

## CHAPITRE 3

# RÉSEAUX DE NEURONES ARTIFICIELS

***Résumé :** Dans ce chapitre, nous nous intéressons aux réseaux de neurones artificiels(RNA). Après une présentation de quelques repères historiques des RNA et une description des fondements biologiques des neurones qui sont à la base des modèles de neurones artificiels ; nous décrirons le principe de fonctionnement des RNA, leurs différentes architectures, les types d'apprentissage ainsi que la présentation de quelques exemples typiques de ces réseaux. Nous nous intéressons par la suite au quand et au comment de la mise en œuvre de ces réseaux de neurones. Enfin une présentation de leurs grands domaines d'applications est faite.*

### 3.1. Introduction

Le cerveau humain pèse environ 1500 grammes et contient deux types de cellules, les gliales constituent le tissu nourricier et la structure du cerveau et représente environ 99.99% de volume de cerveau, et les neurones dont le nombre estimé entre 10-100 milliards, forment un réseau interconnecté complexe [8].

Les RNA sont à l'origine une tentative d'imitation de ce merveilleux organe. Ils couvrent une voie prometteuse de l'IA, à l'intersection des différents domaines (informatique, électronique, science cognitive, neurobiologie et même philosophie). Ils ont prouvé leur capacité à résoudre des problèmes plus complexes où l'informatique classique n'arrive pas à trouver des solutions.

L'application des réseaux de neurones recouvre plusieurs domaines dont l'implémentation de plusieurs applications a donné de bons résultats, en télécommunications et informatique : analyse du signal, élimination du bruit, reconnaissance de formes (bruits, images, paroles)..., en finance: prévision et modélisation du marché (cours de monnaies...), sélection d'investissements..., et en Industrie : contrôle qualité, diagnostic de pannes...

Le plus grand intérêt des réseaux de neurones réside dans leur capacité d'apprentissage à partir d'exemples, leur parallélisme des traitements, et leur robustesse en présence de bruit et même en cas d'insuffisance de données.

Dans ce chapitre, nous présentons dans une première étape l'historique et quelques aspects des réseaux de neurones formels, nous décrivons ensuite les deux types d'architectures connus : bouclé et non bouclé. Dans un deuxième temps, nous traitons l'apprentissage des réseaux de neurones, soit en mode supervisé ou bien le mode non supervisé. Dans la seconde partie de ce chapitre, nous nous intéressons à présenter quelques réseaux usuels et particulièrement le perceptron multicouches et les réseaux à fonction de base radiale, nous spécifions ensuite les conditions sous lesquelles on peut mettre en œuvre les réseaux de neurones ainsi que les étapes à suivre. Enfin nous terminons avec quelques domaines d'applications de ces architectures des RNA.

### 3.2. Historique

Les premiers travaux sur les neurones artificiels ont débuté au début des années 1940 et ont été menés par McCulloch et Pitts. L'idée principale est de simuler de façon très simplifiée le neurone naturel, ils décrivent les propriétés du système nerveux à partir de neurones idéalisés : ce sont des neurones logiques (0 ou 1) [31]. Le premier mécanisme basé sur des fondements biologiques est établi par le neurophysicien Hebb en 1949, qui met en évidence l'importance des synapses dans le processus d'apprentissage [8]. En 1951, on a constitué le premier modèle réel d'un réseau de neurones. Cet ordinateur (SMART) possédait des connexions entre neurones en grande partie aléatoires.

Les résultats de ses travaux ont été l'origine de modèles connexionnistes de plus en plus élaborés. En 1958, Frank Rosenblatt a développé le concept du perceptron à partir des travaux pour expliquer et modéliser les facultés de reconnaissance de formes de système visuel. L'idée essentielle de ce perceptron est de simuler la rétine de l'œil pour reconnaître des formes visuelles particulières [8]. En 1959, Bernard Widrow et Marc Hoff de Stanford ont développé les deux modèles nommés ADALINE et MADALINE. Ces modèles sont nommés de leur utilisation de Multiple ADaptive LINEar Elements. Le Madaline est le premier réseau de neurone appliqué dans un problème réel. C'est un filtre adaptatif pour éliminer les échoués dans les lignes téléphoniques. Ce type de réseau de neurone est généralement destiné pour une utilisation commerciale [32].

En 1969 Minsky et Papert écrivent l'article définitif, « Perceptrons », ils montrent clairement que les réseaux de neurones de cette période, comme le perceptron, n'étaient même pas capables de comprendre la fonction logique XOR [33]. Puis, durant les années 1970 il y eut une remise en cause de l'intérêt des réseaux car les ordinateurs de neurones apprenaient lentement, coûtaient très cher et leurs performances n'étaient pas si impressionnantes [31]. En 1978, en se basant sur la physique quantique des « verres de spin », le physicien John Hopfield a développé un réseau assimilable à une mémoire associative caractérisé par la propriété que l'on accède à son contenu et non à son adresse comme dans les calculateurs actuels [8].

Cette démarche basée sur la physique quantique et la disponibilité croissante des minis et microordinateurs, vers la fin des années 70 a relancé les travaux sur les applications de neurones artificiels et le développement de réseaux particuliers tels que les réseaux de Boltzman qui mettent en œuvre la technique du recuit simulé. Au début des années 80, un algorithme de rétropropagation a été découvert de façon indépendante par plusieurs

chercheurs (Parker en 1982, Le Cun en 1985, et Rumelhart, Hinton et Williams en 1986) [8]. Cet algorithme de retropropagation est devenu l'algorithme le plus utilisé.

Aujourd'hui, l'utilisation des RNA recouvre plusieurs domaines mais la plupart des réalisations sont souvent plus proche du prototype qu'en exploitation véritable.

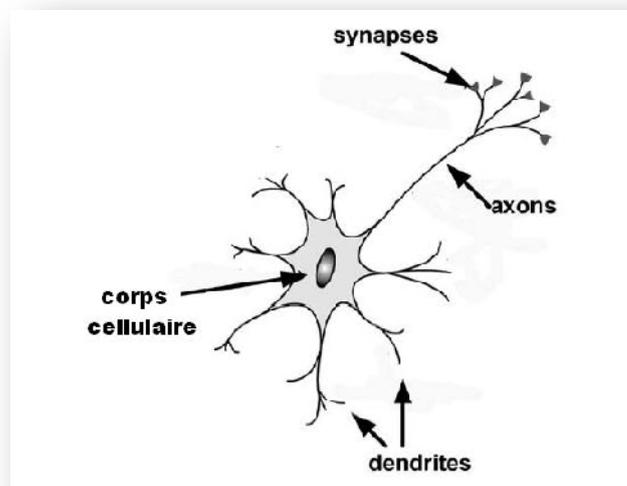
### 3.3. Neurone et réseau de neurones

#### 3.3.1. Neurone biologique

Le cerveau humain pèse environ 1500 grammes et contient trois couches successives:

- le cerveau reptilien, siège des réflexes instinctifs,
- l'hippocampe, siège de l'émotivité et de la sensibilité,
- le cortex, spécifique au mammifère et siège de raisonnement et de la pensée.

Ces couches contiennent deux types de cellules, Les *gliales* représentent environ 99.99% de volume de cerveau et les *neurones* dont le nombre variant entre 10 - 100 milliards organisés en millions de réseaux fonctionnels. Chaque neurone possède un *axone*, un *corps cellulaire*, des *dendrites* qui reçoivent des informations des autres neurones et des terminaisons *axonales* qui émettent des informations vers les autres neurones [8]. La figure 3.1 représente le schéma simplifié d'un neurone biologique.



*Figure 3.1 : Neurone biologique*

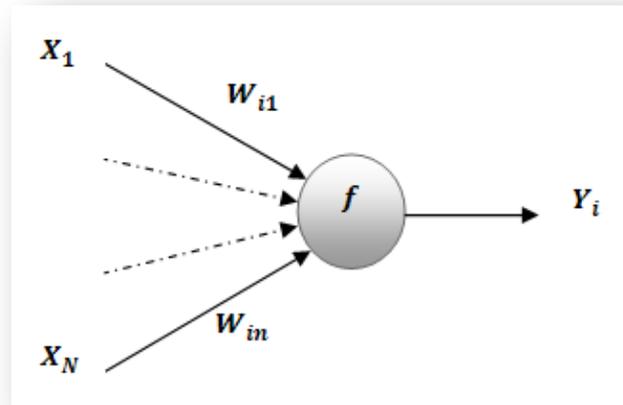
Les neurones sont reliés entre eux par des axones. Ces axones vont eux-mêmes jouer un rôle important dans le comportement logique de l'ensemble. Ils conduisent les signaux électriques de la sortie d'un neurone vers l'entrée (synapse) d'un autre neurone [31]. Ce processus de communication entre neurone se fait par voie électrochimique au niveau des synapses par des neurotransmetteurs [8].

Les RNB sont aussi un modèle pour les RNA que les informaticiens cherchent plus ou moins à imiter, notamment dans le domaine de l'IA.

### 3.3.2. Neurone formel

*Un neurone est une fonction algébrique non linéaire, paramétrée, à valeur bornée [34].*

Il est commode de présenter graphiquement un neurone comme indiqué dans la figure 3.2.



**Figure 3.2 : Représentation graphique d'un neurone artificiel**

Ce modèle non linéaire appelé suivant les auteurs : neurones artificiel, processeur élémentaire, cellule ou automate cellulaire reçoit des informations provenant de  $N$  processeurs  $P_j, j = 1, \dots, N$ , et délivre un signal de sortie  $Y_i$ . Les  $N$  signaux d'activation  $X_j, j = 1, \dots, N$ , provenant des neurones externes multipliée par des coefficients de pondération appelé poids de connexions synaptique  $W_{ij}$ . Ces coefficients représentent le niveau de relation entre les neurones artificiels. Les poids de connexion synaptiques  $W_{ij}$  représentent la mémoire de neurone artificiel.

A l'aide d'une fonction d'entrée totale booléenne, linéaire, ou affine, le neurone artificiel élabore le signal d'entrée total à partir de ces  $N$  signaux d'activation. La sortie  $Y_i$  de neurone artificiel  $P_i$  qui servira ensuite à exciter les autres neurones qui lui sont connectés est calculé à l'aide d'une fonction de transfert  $f_i$ .

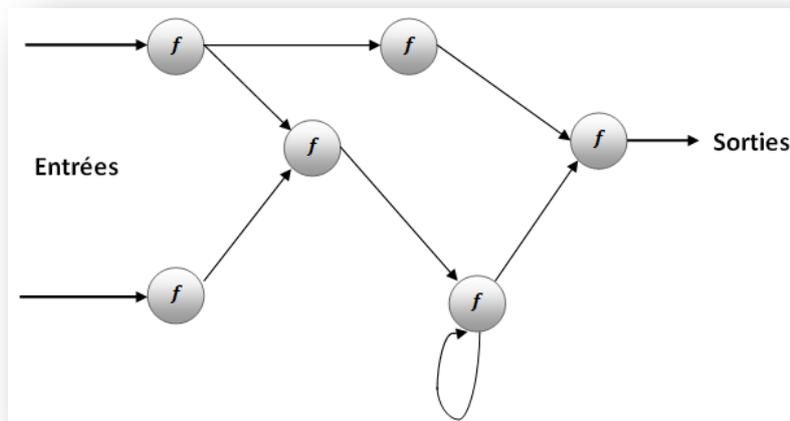
$$Y_i = f_i(S_i) \quad (2.1)$$

$$\text{Où } S_i = \sum_{j=1}^{j=N} W_{ij} X_j - W_{i0} \quad (2.2)$$

$S_i$  : la valeur d'activation dans le cas d'une fonction d'entrée de type affine et  $W_{i0}$  est appelé le signal d'inhibition de neurone et on obtient la fonction linéaire si  $W_{i0} = 0$  [8].

### 3.3.3. Réseau de neurones artificiels

L'interconnexion de plusieurs neurones entre eux forme ce qu'on appelle un réseau de neurone, c'est une combinaison des fonctions algébriques, où la sortie d'un neurone est l'entrée d'un autre. La représentation d'un réseau de neurone est généralement par un graphe orienté où les nœuds sont des neurones et les flèche représentent les connexions entre les neurones, dont une représentation graphique d'un réseau de neurones est illustrée sur la figure 3.3 où Les poids  $W_{ij}$  des arcs du graphe sont appelés "Poids Synaptiques".



**Figure 3.3 : Représentation graphique d'un réseau de neurones artificiels**

Chaque réseau de neurones est représenté par ces neurones d'entrées et un ou plusieurs neurones de sorties. Les neurones qui ne sont ni en entrée ni en sortie sont par définition les neurones "Cachés". Le nombre de neurones et la structure de la matrice des poids déterminent "l'Architecture du réseau".

## 3.4. Types d'architectures

On distingue deux types de RNA : Les réseaux non bouclés et les réseaux bouclés :

### 3.4.1. Réseaux de neurones non bouclés

*Un réseau de neurones non bouclé réalise une (ou plusieurs) fonctions algébriques de ses entrées, par composition des fonctions réalisées par chacun de ses neurones [34].*

Dans un réseau de neurones non bouclée, l'information circulant des entrées vers les sorties sans "retour en arrière" ; si l'on représente le réseau graphiquement (figure 3.4), le graphe d'un réseau non bouclé est acyclique: Si l'on se déplace dans le réseau, à partir d'un neurone quelconque, en suivant les connexions, on ne peut pas revenir au neurone de départ [34].

La majorité des réseaux de neurones non bouclés mise en œuvre pour les tâches de classification automatique sont organisés en plusieurs couches dont certaines sont cachées [8].

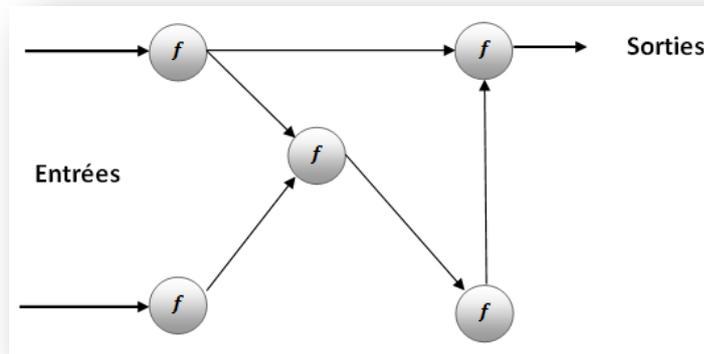


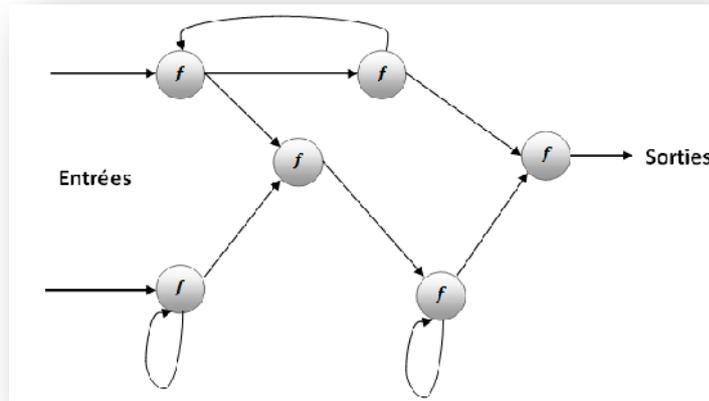
Figure 3.4 : Réseau de neurones non bouclé

### 3.4.2. Réseaux de neurones bouclés

Un réseau de neurones bouclé ou bien à connexions récurrentes signifie qu'une ou plusieurs sorties de neurones d'une couche aval sont connectées aux entrées des neurones de la couche amont ou de la même couche. Ces connexions récurrentes ramènent l'information en arrière par rapport au sens de propagation défini dans un réseau non bouclé [31].

Contrairement aux réseaux de neurones non bouclés, le graphe de connexions des réseaux de neurones bouclés est cyclique : lorsqu'on se déplace dans le réseau, en suivant le sens des connexions, il est possible de trouver au moins un chemin qui revient à son point de départ (un tel chemin est désigné sous le terme de "Cycle"). La figure 3.5 représente cette architecture. Comme pour les réseaux de neurones non bouclés, à chaque connexion d'un réseau de neurones bouclés est attaché un poids, un *retard*, multiple entier de l'unité de temps choisis [34].

*Un réseau de neurones bouclé à temps discret est donc régi par une (ou plusieurs) équations aux différences non linéaires, résultant de la composition des fonctions réalisées par chacun des neurones et des retards associés à chacune des connexions [34].*



**Figure 2.5 : Réseau de neurones bouclé**

Les réseaux de neurones bouclés sont utilisés pour effectuer des tâches de modélisation de systèmes dynamiques, de commande de processus, ou de filtrage [35]...

Tout réseau de neurones bouclé peut être mis sous en forme canonique, comprenant un réseau de neurones non bouclé dont les sorties d'état sont ramenées à ses entrées avec un retard unité. Alors on peut dire que l'élément de base est donc le réseau de neurone non bouclé [34].

### 3.5. Types d'apprentissage

On appelle "apprentissage" des réseaux de neurones la procédure qui consiste à estimer les paramètres des neurones du réseau, afin que celui-ci remplisse au mieux la tâche qui lui est affectée [34].

On distingue deux types d'apprentissages : l'apprentissage supervisé et l'apprentissage non supervisé.

#### 3.5.1. Apprentissage supervisé

Dans un apprentissage supervisé, on connaît les valeurs que doit avoir la sortie de réseau en fonction des données correspondantes, le réseau de neurones utilise ces données d'entrées pour fournir des valeurs en sortie, et compare ces résultats avec la sortie désirée. S'il y a un écart, le réseau ajuste les poids  $W_{ij}$  de telle manière que les sorties du réseau soient aussi proches que possibles des sorties "désirées". Ces exemples utilisés lors de l'apprentissage sont appelés "Ensemble d'apprentissage". Durant l'apprentissage de réseau la même ensemble de données est utilisée plusieurs fois jusqu'à que les poids  $W_{ij}$  génèrent la sortie désirée, qui peuvent être la valeur de la fonction que l'on veut approcher, la classe

d'appartenance de la forme que l'on veut classer ou de la sortie du processus que l'on veut modéliser, ou encore la sortie souhaitée du processus à commander [32] [35].

En général, après la phase d'apprentissage et la convergence des calculs, il n'est plus nécessaire de mémoriser le jeu d'apprentissage. Ce type d'apprentissage est particulièrement dédiés aux problèmes de classification des données bruités ou incomplètes [8].

On peut trouver dans quelques ouvrages un autre type d'apprentissage, qu'est l'apprentissage semi-supervisé. Dans ce type d'apprentissage, Les données d'entrée sont connus, et on ne connaît pas les données de sorties mais on fournit des exemples et des indications sur le résultat (vrai ou faux) [36].

### **3.5.2. Apprentissage non supervisé**

Dans un apprentissage non supervisé, le réseau de neurone est construit en se basant sur les données d'entrées et non plus sur les sorties. Ce type d'apprentissage imite le fonctionnement de cerveau humain qui retrouve les informations par association. On présente à l'entrée du réseau des exemples connus et le réseau s'organise lui-même autour d'attracteurs qui correspondent à des configurations stables du modèle dynamique non linéaire associé au réseau. A l'aide des règles, le système change ou adapte les poids synaptiques en fonction des données présentés à l'entrée, et regroupe ces données en classes. Cette organisation est appelé l'Auto-Organisation ou l'Auto-Adaptation [8] [32].

Si le réseau n'arrive pas à résoudre le problème donné, le concepteur révisé les entrées et les sorties, le nombre des couches, le nombre des nœuds pour chaque couches, les connections entre les couches, les fonctions de transferts, et les fonctions d'apprentissages, et aussi les poids initial  $W_{i0}$ . [32].

La plupart des algorithmes d'apprentissage des réseaux de neurones formels sont des algorithmes d'optimisation : ils cherchent, par des méthodes d'optimisation non linéaire, à minimiser une fonction de coût qui constitue une mesure de l'écart entre les sorties réelles du réseau et ses sorties désirées [35]. Plusieurs algorithmes sont utilisés pour l'ajustement des poids  $w_{ij}$  dans la phase d'apprentissage de réseau. La technique la plus utilisée est l'algorithme de rétro-propagation [32].

### **3.6. Quelques réseaux de neurones usuels**

Dans ce qui suit on présente quelques exemples typiques des réseaux de neurones les plus utilisés.

### 3.6.1. Perceptron

Le perceptron est le premier des réseaux de neurones, conçu en 1958 par Rosenblatt, basé sur le modèle de McCulloch et Pitts (1943), dont le but est de modéliser la vision humaine [24].

Le perceptron généralement comprend trois couches principales, La première couche (d'entrée) représente la rétine. Les neurones de la couche suivante sont les cellules d'association, et la couche finale les cellules de décision. La figure 3.6 illustre cette architecture.

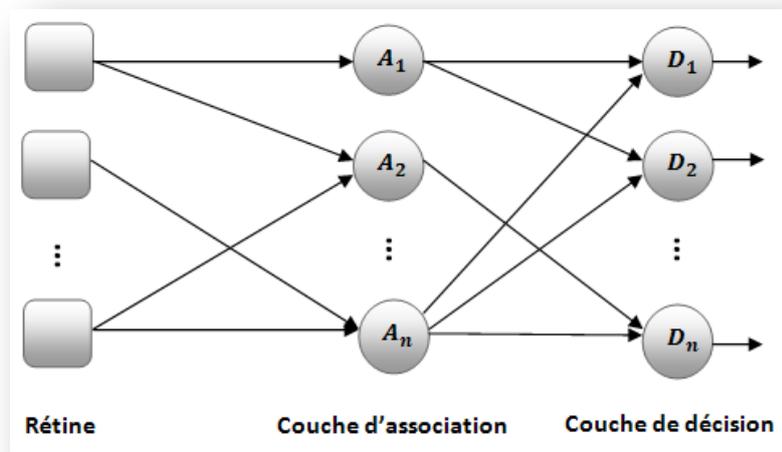


Figure 3.6 : Représentation graphique de perceptron

Les couches de perceptron sont:

- *La rétine :*

Comprend plusieurs cellules qui jouent le rôle de capteurs. Elle reçoit les exemples ou formes à classer. Chaque élément de la rétine peut être considéré comme un pixel prenant des valeurs binaires (0/1). Les cellules de cette couche sont connectées de façon aléatoire ou directement aux cellules de la deuxième couche.

- *La couche d'association :*

Cette deuxième couche est composée de cellules associatives  $A_j$  qui sont dotées de fonctions d'association  $h$  qui peuvent réaliser des fonctions booléennes ou bien utiliser des fonctions linéaires. La cellule de décision  $D_i$  reçoit la sortie  $X_j$  de la cellule d'association  $A_j$  après avoir été pondérée par le coefficient ajustable  $W_{ij}$ . Les fonctions  $h_i$ ,  $i = 1, \dots, n$  sont déterminées à l'avance et restent fixes pendant la phase d'apprentissage.

- *La couche de décision :*

La cellule de décision est un automate à seuil de fonction de transfert  $f_i$  qui délivre la sortie binaire  $S_i$  [8].

### **Apprentissage de perceptron**

Le perceptron apprend seulement s'il y a une erreur de classification dans le vecteur d'entrée d'exemples d'apprentissage. Il ajuste les poids  $W_{ij}$  si la valeur de sortie désiré est 1 et la valeur de sortie générée par le réseau égal à 0, les poids synaptique de neurone de sortie augmente et vice versa. Si la sortie produite  $O_j(k)$  est proche de la valeur de sortie désirée  $Y_j(k)$  de l'exemple d'apprentissage  $(x, y)(k)$ , alors on ne change pas les poids  $W_{ij}$   $i = 0, \dots, n$ . L'algorithme d'apprentissage de perceptron est donné dans le tableau suivant [24] :

---

<i>P1</i>	<i>Soit <math>(n+1)</math> entrées, <math>m</math> sorties de perceptron. Associer des petites valeurs aléatoires pour les poids de réseau <math>W_{ij}, i = 0, \dots, n; j = 1, \dots, m</math>.</i>
<i>P2</i>	<i>Choisir un vecteur forme d'entrée <math>X</math> et calculer le signal d'entrée <math>u_j</math> pour chaque sortie de neurone de perceptron <math>j</math> en utilisant la formule standard :</i> $u_j = \sum(X_i \cdot W_{ij}), \text{ pour } i = 0, \dots, n; j = 1, \dots, m; \text{ où } X_0 = 1.$
<i>P3</i>	<i>Appliquer une fonction d'activation de seuil élevé pour les signaux d'entrée comme suit : <math>O_j = 1</math> si <math>u_j &gt; \text{seuil}</math>, sinon <math>O_j = 0</math></i> <i>(Il est possible d'appliquer une fonction linéaire)</i>
<i>P4</i>	<i>Calculer l'erreur pour chaque neurone en se basant sur la différence entre la sortie actuelle et la sortie désirée : <math>Err_j = Y_j - O_j</math></i>
<i>P5</i>	<i>Modifier chaque poids <math>W_{ij}</math> par le calcul de leur valeur suivante <math>W_{ij}(t + 1)</math> :</i> $W_{ij}(t + 1) = W_{ij}(t) + \alpha X_i \cdot Err_j$ <i>Où <math>\alpha</math> est le coefficient d'apprentissage compris entre 0 et 1</i>
<i>P6</i>	<i>Répéter les étapes P2-P5 jusqu'à l'obtention de vecteur d'erreur faible, ce qui signifie la convergence.</i>

---

### **Algorithme 3.1 : Algorithme d'apprentissage de perceptron.**

En 1969, Papert et Minsky démontrent les limites du perceptron classique, incapable, par exemple de simuler la fonction OU exclusif (XOR) [33], les réseaux en couches peuvent être considérés comme une solution alternative.

### 3.6.2. *Perceptron multicouches(PMC)/ Multilayer Perceptron (MLP)*

Les PMCs sont des réseaux de neurones à couches, ils comprennent :

- Une couche d'entrée qui reçoit l'ensemble des formes à classer,
- Une ou plusieurs couches cachées intermédiaires,
- Une couche de sortie qui doit restituer l'ensemble des sorties désirées après apprentissage.

Aujourd'hui les réseaux de neurones multicouches sont les plus utilisés en diagnostic industriel en raison de leur capacité à résoudre les problèmes de classification non linéaire par l'utilisation de l'algorithme de retropropagation du gradient, qui est une généralisation de la règle de Widrow-Hoff pour le calcul des poids de connections  $W_{ij}$ [8].

#### *Apprentissage basé sur l'algorithme de rétropropagation*

L'algorithme le plus utilisé dans l'apprentissage des réseaux de neurones est l'algorithme de rétro-propagation. Cet algorithme est développé en 1969 par Bryson et Ho, mais il était ignoré complètement ou partiellement au milieu des années 80. Ce qui peut être expliqué par la complexité de calcul lorsque l'algorithme est appliqué pour des problèmes sophistiqués. [37].

La méthode de retropropagation est similaire à celle utilisée par Widrow Hoff, Elle est proposée par plusieurs auteurs dont le but est de procéder à l'apprentissage des réseaux de neurones multicouches. On attribue à la couche finale une fonction d'erreur et on ajuste chaque poids  $W_{ij}$  de réseau d'une manière que l'erreur soit minimal. Si on prend  $d_i$  la réponse désirée de  $i^{me}$  vecteur d'entrée,  $X_i$ , dans l'ensemble d'apprentissage, E, on peut calculer l'erreur d'apprentissage de la manière suivante :

$$\varepsilon = \sum_{X_i \in E} (d_i - f_i)^2 \quad (2.3)$$

Où  $f_i$  est la réponse actuelle de réseau pour l'entrée  $X_i$ . Pour faire minimiser cette erreur quadratique, on ajuste chaque  $W_{ij}$  par une quantité qui est proportionnelle à la dérivé partielle de l'erreur par rapport à ce poids. [38].

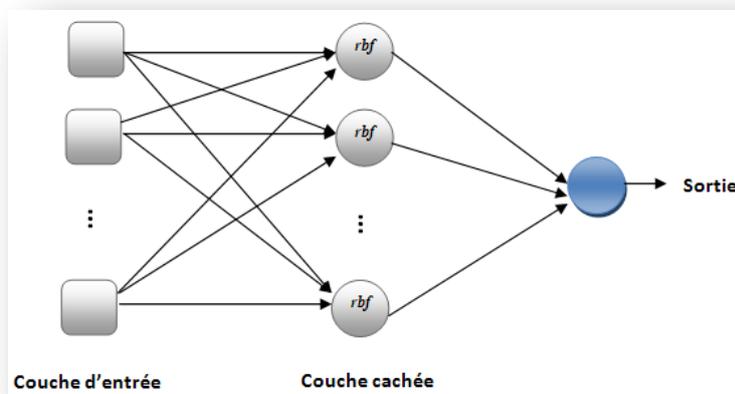
Les études pratiques de l'algorithme de retropropagation ont montré que son application fait appel aux remarques suivantes :

- l'algorithme peut converger vers des minima locaux si la fonction de coût à minimiser n'est pas convexe, alors la valeur initiale des poids influe sur la solution finale.
- le temps d'apprentissage est relativement long, de quelques minutes à quelques heures, suivant la complexité et le nombre d'exemples d'apprentissage.

- il n'existe que des méthodes empiriques pour obtenir le réseau adapté à la résolution d'un problème particulier et la structure de réseau est figée. Pour faire converger l'algorithme plus vite l'utilisation de fonctions de types sigmoïde avec des seuils égaux à -1 et +1 en particulier est utile.
- l'erreur global décroît rapidement au début de l'apprentissage puis se met à augmenter. Alors la mémorisation de la configuration des poids qui reflète l'erreur minimale est importante. Pratiquement il est préférable de partitionner l'ensemble des exemples en trois sous-ensembles : le premier pour l'apprentissage, le deuxième pour la généralisation et le dernier pour le test [8].

### 3.6.3. Réseaux RBF (Radial Basis Function)

Les Réseaux à fonction de base radiale (RBF) proposés et utilisés par plusieurs auteurs (Moody et Darken en 1989; Renals et Rohwer en 1989;...) . Leur architecture en couches ressemble fortement à celle du PMC, mais avec quelques différences. L'architecture générale d'un réseau RBF comporte trois couches: couche d'entrée, couche cachée et une couche de sortie. La première couche en N entrées, elle est complètement connectée à la deuxième couche. Les neurones cachés ont une fonction RBF comme une fonction d'activation. La figure 3.7 représente une architecture générale d'un réseau RBF [24].



**Figure 3.7 : Architecture générale d'un réseau RBF**

D'une manière générale, pour un problème particulier, on peut utiliser soit un réseau RBF ou bien on peut également utiliser un PMC. L'avantage d'un réseau RBF réside dans leur facilité à construire et leur vitesse d'apprentissage par rapport au PMC. Mais face à des problèmes difficiles (peu d'exemples, données bruitées, beaucoup de variables), les réseaux RBF ne peuvent pas lutter avec le PMC en termes de qualité des résultats.

### 3.6.4. Adaline et Madaline

Les deux modèles Adaline et Madaline sont développés par Widrow et M.Hoff de Stanford pour la résolution du filtrage adaptatif du signal, en 1959. Ces modèles sont nommés de leur utilisation de Multiple ADAPtive LINear Elements[32].

L'architecture générale de l'Adaline est présentée sur la figure 3.8 L'Adaline est similaire au perceptron avec une seule cellule d'association et utilise une fonction de seuil différente de celle de Heaviside (-1 et +1). De plus, Pour mesurer l'écart entre la sortie réelle et la sortie désirée l'Adaline utilise un algorithme adaptatif.

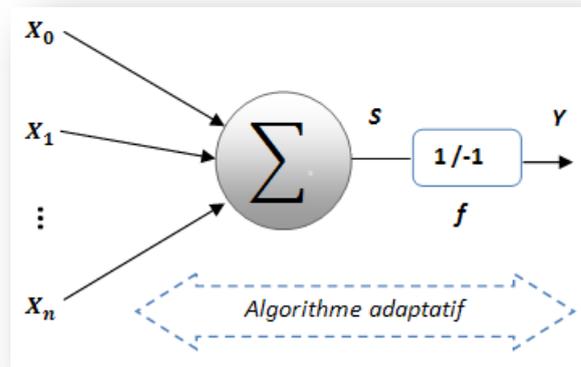


Figure 3.8 : Schémas de principe de l'Adaline

Le réseau de neurones Adaline est le seul réseau utilisé massivement dans l'industrie, il est utilisé pour améliorer le signal sur bruit dans les télécommunications en prenant en compte la variation de l'impédance des différentes lignes téléphoniques [8].

La connexion de plusieurs Adalines entre eux suivant le schéma 3.9 forme le MAdaline (Many Adalines).

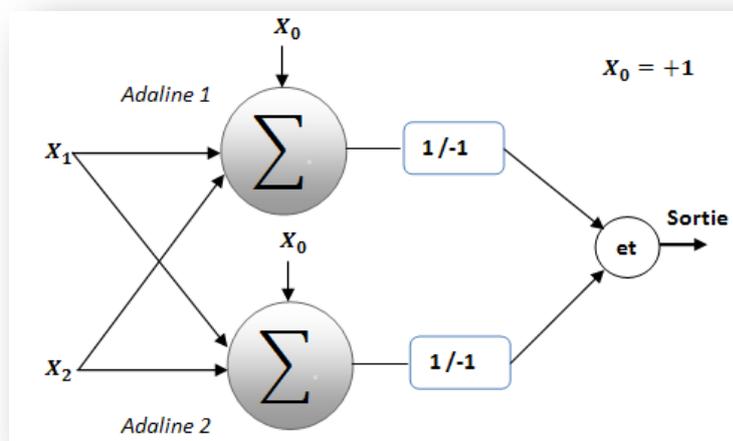
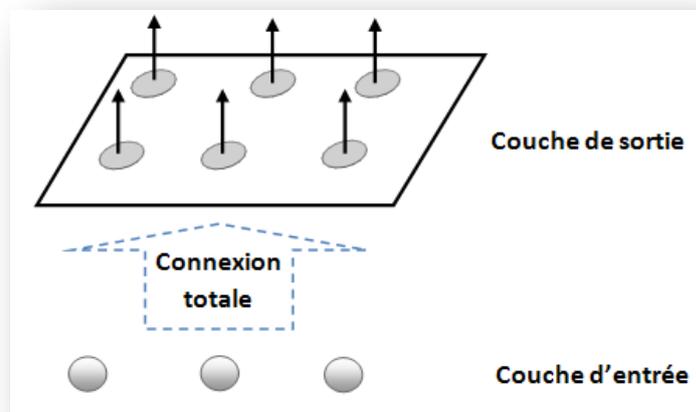


Figure 3.9 Schémas de principe de MAdaline (exemple de deux Adalines)

### 3.6.5. Cartes topologiques de Kohonen

Les cartes topologiques développées par Teuvo Kohonen au début des années 80. En se basant sur des études des systèmes de perception visuelle et auditive des mammifères, Kohonen a construit un réseau où des cellules regroupées autour d'une cellule concentreraient l'information liée à une forme particulière [8].

La carte auto-organisation comprend deux couches, une couche d'entrée et une couche de sortie, appelée la carte topologique, elle représente les vecteurs de sorties de l'espace des sorties. Les nœuds d'entrée sont totalement connectés aux nœuds de sorties. Les valeurs d'entrées sont continues ou bien discrètes, mais les valeurs de sorties sont binaires. La figure 3.10 représente l'architecture générale de la carte topologique [24].



*Figure 3.10 : l'architecture générale de la carte topologique*

La première utilisation de la carte de Kohonen est la visualisation des structures hiérarchiques et topologiques dans un espace d'entrée en deux dimensions. Le réseau de Kohonen est utilisé pour créer des courbes à surface pleine dans un espace bidimensionnel à l'aide de la couche de Kohonen. On peut aussi utiliser cette couche de Kohonen pour les problèmes d'optimisations par l'ajustement des poids de connexions jusqu'à l'obtention de la solution minimale.

La différence majeure entre ce réseau et d'autres types de réseaux réside dans leur apprentissage non supervisé de la carte auto-organisation. En plus, si les réseaux de Kohonen sont combinés avec d'autres couches de neurones, l'apprentissage de l'ensemble au départ est en mode non supervisé et passe ensuite à l'apprentissage supervisé [32].

L'apprentissage non supervisé des réseaux de Kohonen est représenté dans l'algorithme suivant [24] :

---

*K0* Attribuer des petites valeurs aléatoires au vecteur des poids initiaux  $W_j(t = 0)$  pour chaque neurone  $j$  de la carte de sortie

*K1* Choisir un vecteur d'entrée  $X$  à l'instant  $t$  suivante.

*K2* Calculer la distance  $d_j$  dans un espace à  $n$  dimensions entre  $X$  et le vecteur des poids  $W_j(t)$  pour chaque neurone  $j$ . Dans un espace euclidien la distance est donnée par :

$$d_j = \text{sqrt}\left(\sum ((X_i - W_{ij})^2)\right)$$

*K3* Le neurone  $k$  le plus proche à  $X$  est sélectionné comme centre de la région de voisinage  $N_t$ .

*K4* Changer tous les vecteurs de poids au sein de la région de voisinage :

$$W_j(t + 1) = W_j(t) + \alpha \cdot (X - W_j(t)), \text{ si } j \in N_t,$$

$$W_j(t + 1) = W_j(t), \text{ sinon.}$$

Les étapes K1-K4 sont répétées pour la totalité des instances de l'ensemble d'apprentissage.  $N_t$  et  $\alpha$  diminuent avec le temps. La même procédure d'apprentissage est répétée avec les mêmes instances de l'ensemble d'apprentissage jusqu'à la convergence.

---

### **Algorithme 3.2 : Algorithme d'apprentissage non supervisé des réseaux de Kohonen**

Plusieurs applications utilisent les réseaux de Kohonen comme préprocesseur de traitement de réseaux à couches en utilisant la rétropropagation de gradient. Ce type de réseaux est utilisé massivement pour la discrimination de classes en robotique et pour la classification de signatures vibratoires de machines. Le temps d'apprentissage élevé des réseaux de Kohonen est leur inconvénient major [8].

## **3.7. Mise en œuvre des réseaux de neurones**

Avant de présenter comment mettre en œuvre ces réseaux de neurones, il est important de déterminer si le problème est qualifié d'être modélisé par les réseaux de neurones. Ensuite on aborde sous quelles conditions peut-on utiliser cette approche.

### **3.7.1. Comment qualifier qu'une application est candidate d'être modélisée par les réseaux de neurones ?**

Généralement pour quelqu'un qui veut utiliser les réseaux de neurones, il doit d'abord vérifier les deux critères suivants [32] :

- Le remplacement de la technologie existante par des réseaux de neurones a un impact économique important. Comme exemples d'application qui vérifient ce critère on peut citer : les demandes de location et la prédiction dans les marchés financiers.
- Ou bien la technologie existante montre qu'elle n'a pas donné de bons résultats. Les applications suivantes sont un bon exemple pour ce critère : Reconnaissance de la parole, Reconnaissance du texte et l'analyse des buts.

### 3.7.2. Conditions d'utilisation des réseaux de neurones

Avant de procéder à l'application des réseaux de neurones, on doit mettre en exergue les conditions suivantes : [34]

- Il faut disposer d'échantillons de données de taille suffisamment grande, et bien représentatifs. Cette condition est nécessaire mais pas suffisante.
- Il faut s'assurer de l'intérêt réel d'utiliser un modèle non linéaire pour l'application considérée, car l'utilisation d'un modèle linéaire est toujours plus simple qu'un modèle non linéaire, et moins coûteuse en temps de calcul.
- Si les deux premières conditions sont vérifiées, il faut s'interroger sur l'opportunité d'utiliser un réseau de neurone de préférence à une autre famille de fonctions non linéaire.

En générale, pour toute application qui mette en jeu l'estimation d'une fonction de régression non linéaire et possédant au moins trois variables, il est avantageux d'utiliser les réseaux de neurones. Dans ce cas l'utilisation des RNA à fonction d'activation sigmoïde est utile. Si le nombre de variables est inférieur à trois, l'utilisation des réseaux de neurones RBF à centres et écarts-types fixés, ou des ondelettes à centres et dilatations fixés, ou encore des polynômes, peuvent être précises, et plus simple à mettre en œuvre.

### 3.7.3. Mise en œuvre des réseaux de neurones

La conception d'un RNA passe par trois étapes [7] [35] :

- *la construction du réseau de neurones* : il faut tout d'abord choisir l'architecture du réseau (nombre de couches, le nombre de neurones cachés, et l'agencement des neurones entre eux) ainsi que les fonctions à associer aux neurones, de telle manière que le réseau soit en mesure de reproduire ce qui est déterministe dans les données. Le problème de la détermination de l'architecture optimale reste pendant longtemps un problème ouvert, mais il existe actuellement diverses méthodes, mettant notamment en jeu des tests

statistiques, qui permettent de déterminer cette architecture pour une vaste classe de réseaux.

- *l'ajustement des différents paramètres* : Cette étape consiste à calculer les poids du réseau, en minimisant l'erreur d'approximation sur les points de l'ensemble d'apprentissage, de telle manière que le réseau réalise la tâche désirée, c'est la phase d'apprentissage.
- *Le teste et l'exploitation du réseau de neurones* : il faut enfin estimer la qualité du réseau obtenu en lui présentant des exemples qui ne font pas partie de l'ensemble d'apprentissage. et ceci dans le but d'évaluer l'erreur de classification que ce soit pour le test ou bien l'exploitation.

### 3.8. Domaines d'applications des réseaux de neurones

Les RNA recouvrent des domaines d'application très variés. Il est difficile de faire une liste complète de toutes les applications des réseaux de neurones. Les grands domaines d'application découlent naturellement de leur propriété fondamentale [35].

#### 1. *La régression non linéaire, ou modélisation de données statiques*

Les RNA sont considérés comme une bonne solution pour modéliser les relations déterministe entre des causes et des effets des phénomènes statiques à partir d'observations expérimentales (suffisamment nombreuses et représentatives).

#### 2. *La modélisation de processus dynamiques non linéaires*

La modélisation d'un processus consiste à trouver une solution mathématique, c'est-à-dire trouver des équations mathématiques qui peut utiliser les données d'entrées de processus pour évaluer les données de sorties. Si le phénomène que l'on désire modéliser est non-linéaire, les RNA sont typiquement de bons candidats pour une telle modélisation. La prédiction de séries chronologiques (prédictions financières [39], prédiction de consommation, etc.) fait partie de ce domaine d'application.

#### 3. *La commande de processus*

La commande de processus peut donc être considérée comme un système qui réalise une fonction (non linéaire) qu'un réseau de neurones peut approcher.

#### 4. *La classification*

Les réseaux de neurones sont des bons candidats pour réaliser des approximations qui constituent des estimations de la probabilité d'appartenance d'une forme inconnue à une classe défini au priori. Cette propriété remarquable (que les réseaux de neurones partagent avec d'autres classifieurs) n'est malheureusement pas mise à profit dans la plupart des applications.

### 3.9. Conclusion

Actuellement, les RNA arrivent à un degré de maturité permettant d'obtenir de bons résultats dans domaines d'applications divers, et en plus l'étude de nouvelles structures et architectures neuro-mimétiques, les progrès théoriques, et l'expérience croissante ouvrent les portes sur la résolution des problèmes de grande complexité.

Le PMC et les RBF sont deux bonnes architectures neuronales utilisées massivement dans le domaine de diagnostic industriel, leur avantage réside dans leur capacité d'apprentissage à partir d'exemples représentatifs et de reconnaissance des formes, ainsi que la capacité de résoudre les problèmes de classification non linéaire par l'utilisation de l'algorithme de retropropagation de gradient.

Le choix de l'architecture de réseau de neurones (les entrées, les neurones cachées, les poids des connexions,...) ainsi que le choix des fonctions d'activations sont parmi les facteurs fondamentaux de la réussite d'une application basée sur les réseaux de neurones. Un mauvais choix de ces paramètres (un trop grand nombre de poids par exemple) influe sur les performances de réseau, alors l'application de cette technique de manière irréfléchie ou inadaptée ne peut conduire qu'à des déceptions. Alors, il est important de modifier les conditions initiales de réseau (nombre de neurones, nombre de couches, les poids des connexions,...) afin d'arriver à des performances fiables de réseau.

Malgré que l'inconvénient majeur des RNA est que leur comportement est difficile à comprendre par rapport aux autres techniques, une bonne compréhension et une bonne utilisation de ces réseaux peut donner des bons résultats pour des applications très variées.

Dans le chapitre suivant nous faisons une hybridation entre les SE et les RNA pour réaliser un prototype d'un système de diagnostic industriel, en profitant des capacités des RNA pour l'acquisition automatique des connaissances.

## CHAPITRE 4

# APPROCHE DEVELOPPEE APPLIQUEE AU DIAGNOSTIC D'UN PROCESSUS INDUSTRIEL

***Résumé :** Ce chapitre porte sur l'application des méthodes présentées dans les chapitres précédents pour l'élaboration d'une approche de diagnostic sous forme d'un SE hybride pour le diagnostic d'un processus industriel. Cette approche comporte deux étapes principales. La première étape s'intéresse à discuter le rôle de RNA et l'architecture à adopter, ensuite une présentation d'une approche statistique pour la réduction de nombre de variables d'entrée de RNA. La deuxième étape est consacrée à la présentation des phases à suivre pour la construction de la base des connaissances, à partir de la définition du système jusqu'à l'insertion des prédicats dans la base des connaissances.*

## 4.1.Introduction

Ce chapitre est réservé à la présentation d'une méthode hybride (SE hybride) pour le diagnostic d'un processus industriel. L'hybridation réalisée est basée sur deux méthodes prometteuses de l'IA qui sont les SE et les RNA.

Nous avons proposé un RNA pour l'acquisition automatique des connaissances plus exactement pour alimenter la base des faits du SE, dite base des faits dynamique. Le rôle de RNA est la détection des défaillances élémentaires qui sont la cause primaire de génération des événements indésirables et dégradation du fonctionnement normal du système. Aussi le rôle de notre système expert dans ce cas est la localisation des composants défaillants par l'inférence des faits (pannes élémentaires) et en plus la présentation de tous les événements indésirables pouvant être générés par ces pannes élémentaires.

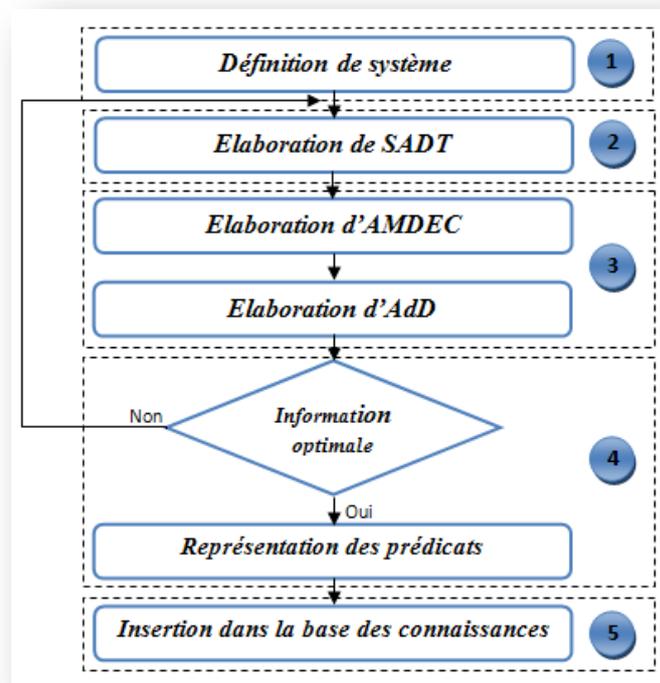
Pour l'alimentation de la base des règles de notre SE, nous avons appliqué l'approche proposée dans [41], basée sur l'utilisation des méthodes SADT, AMDEC et AdD. On utilise aussi l'AMDEC pour générer l'ordre d'exécution des règles, ainsi que les tâches de maintenance à entreprendre.

## 4.2. Présentation de l'approche hybride adoptée

Le but essentiel de cette approche est concrétisé par deux objectifs principaux :

*Le premier objectif* est l'utilisation d'un RNA pour l'acquisition automatique des connaissances en se basant sur les données fournies par les différents capteurs, afin d'alimenter la base des faits du SE. Alors dans ce cas le rôle de ce RNA est la détection des pannes élémentaires.

*Le deuxième objectif* est la construction de la base des connaissances en se basant sur l'approche proposée dans [41], comme il est illustré dans la figure 4.1.



*Figure 4.1: construction de la base des connaissances*

Le contenu de cette architecture est détaillé dans 4.5.

La figure suivante (4.2) montre l'architecture générale de cette approche:

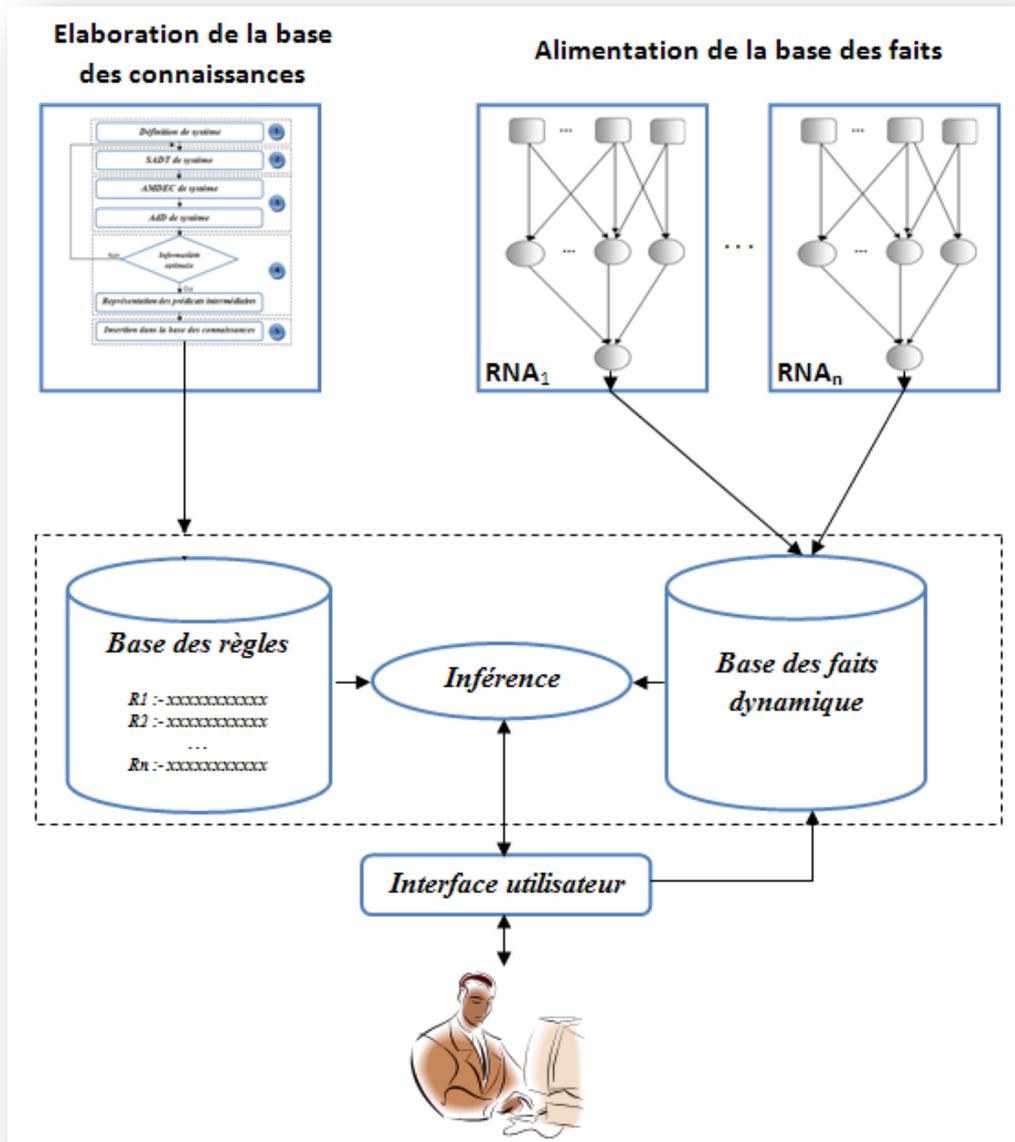
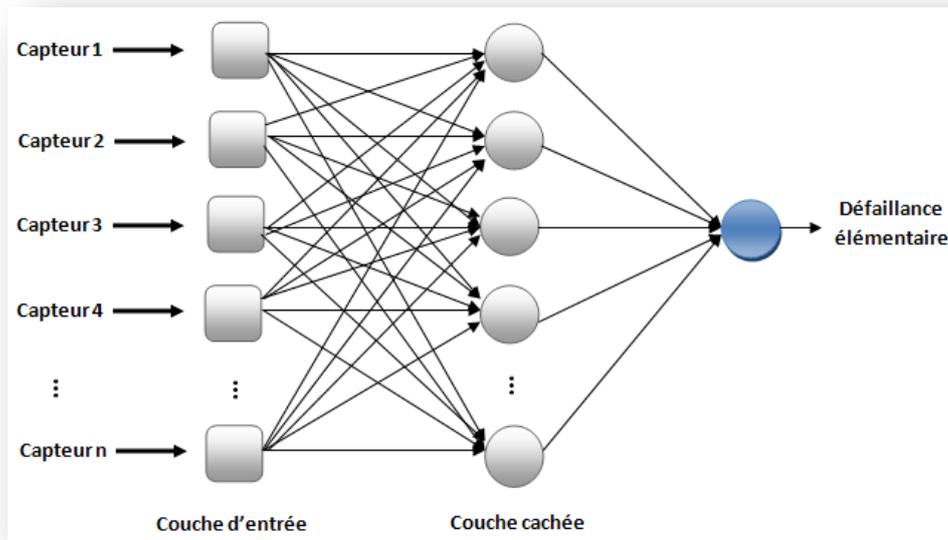


Figure 4.2 : Architecture de SE hybride proposée

### 4.3. Etape 1 : Construction de RNA

Le but de cette étape est la construction d'un RNA pour l'acquisition automatique des connaissances et en particulier les faits indiquant une défaillance dans les composants élémentaires. L'entrée de ce RNA est un vecteur des valeurs fournies par les différents capteurs du système. La sortie de ce réseau représente l'un des modes de défaillance définis dans l'AMDEC du système étudié. L'occurrence d'une défaillance est interprétée par la divergence entre la valeur de sortie de réseau à l'instant  $t$  et la valeur à l'instant  $t+1$ . Cette occurrence exprimée par une assertion logique est insérée automatiquement dans la base des faits. La figure suivante (4.3) montre l'architecture de ce RNA.



*Figure 4.3 : Architecture générale de RNA utilisé pour la détection des défaillances élémentaires*

Afin de réduire la taille du vecteur d'entrée (vecteur d'état du processus industriel) de RNA ce qui revient à réduire le nombre de capteurs utilisés nous avons proposé dans [42] une approche basée sur l'utilisation de quelques méthodes statistiques.

#### **4.4. Approche de réduction statistique**

Le schéma général de notre approche est illustré sur la figure 4.4. Les étapes 2 et 3 sont répétées itérativement jusqu'à l'obtention d'un bon indice de séparation. A ce niveau la procédure est arrêtée, et le nouveau vecteur d'état obtenu est utilisé comme entrée pour le RNA.

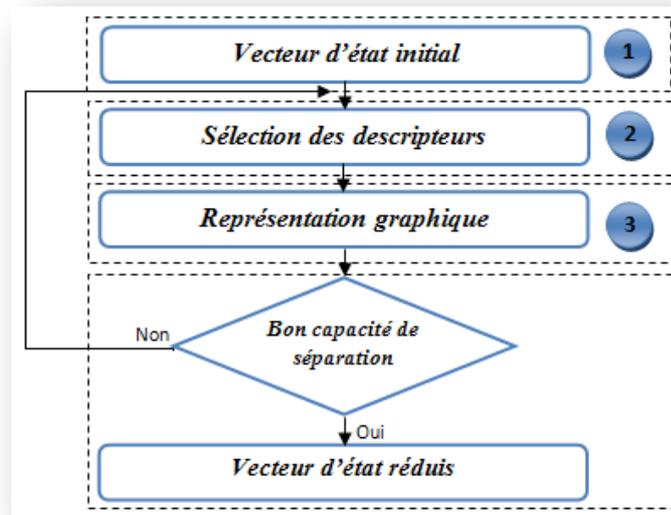


Figure 4.4 : Approche de réduction statistique

#### 4.4.1. Construction de vecteur d'état initial

Cette étape consiste à la transformation des signaux mesurés, décrivant le fonctionnement de processus, en valeurs quantifiables. Cette procédure est largement facilitée par l'insertion de plusieurs capteurs au niveau de processus industriel. Cependant, le problème commun rencontré est la multi-colinéarité des variables d'entrées. Les paramètres mesurés peuvent être corrélés avec tous les autres. Dans ce cas, la présence d'une variable d'entrée dans le vecteur d'état masque l'effet des autres variables. Cette anomalie est réduite dans l'étape suivante.

#### 4.4.2. Sélection des descripteurs

Dans le but d'obtenir un ensemble réduit des descripteurs représentatifs, l'approche de régression pas à pas, qui est une technique pour choisir les variables à inclure dans le modèle de régression multiple, est adoptée pour obtenir les paramètres les plus significatifs expliquant la variabilité dans les classes de défaillances. Au début, dans la régression pas à pas on commence sans termes dans le modèle initial. Durant chaque étape on ajoute le terme le plus significatif statistiquement (le terme avec la plus grande valeur du test de Fisher ou la moindre valeur de p-test) jusqu'à qu'aucun terme n'est resté. Il est possible aussi de démarrer avec le sous-ensemble de tous les termes et ensuite éliminer les termes non significatifs.

Le but de la régression linéaire multiple est d'élaborer la relation quantitative entre un groupe de variables de prédiction  $X$  et une réponse  $y$ . Cette relation est utile par exemple pour

comprendre le descripteur avec le plus grand effet. Le modèle linéaire général est exprimé par:

$$y = X\beta + \varepsilon \quad (4.1)$$

La solution de problème est un vecteur,  $b$ , qui estime le vecteur inconnu des paramètres,  $\beta$ . La solution basée sur la méthode des moindres carrés est définis par :

$$b = \hat{\beta} = (X^T X)^{-1} X^T y \quad (4.2)$$

Le résidu est la différence entre les valeurs de  $y$  observée et le  $y$  prévue.

$$r = y - \hat{y} = (I - X(X^T X)^{-1} X^T) y \quad (4.3)$$

Les résidus peuvent être utilisés pour la détection des défaillances dans le modèle élaboré, à partir de l'erreur correspondante  $\varepsilon$ .

#### 4.4.3. Représentation graphique

Cette phase permet la visualisation en 3D des classes de défaillances multidimensionnelle modélisées. On peut simplifier notre problème par le remplacement d'un groupe de variables par une seule nouvelle variable. L'analyse à composantes principale (ACP) est une méthode quantitative rigoureuse pour réaliser cette simplification. La méthode génère un nouvel ensemble des variables nommé composantes principales, qui sont orthogonales, avec les autres informations redondantes éliminées dans l'espace de données résultant. Cette base orthogonale est exprimée par les vecteurs propres  $v_j$  de la matrice variance-covariance  $S$  satisfaisant à la relation :

$$S \times v_j = \lambda_j \times v_j \quad (4.4)$$

Par la considération seulement des premiers vecteurs propres dans un ordre décroissant des valeurs propres associées, une représentation optimale de la matrice de données initiale peut être réalisée dans un espace réduit. L'analyse par composantes principales assure que la déformation des différentes classes est minimale en termes de critère  $J_p$ , en basant sur la distance quadratique minimale entre les données originales et les données projetées dans la nouvelle base orthogonale formée:

$$J_p = \frac{1}{n} \sum_{k=1}^N d^2(x_k, x_k^p) \quad (4.5)$$

La qualité de représentation dans l'espace réduits de  $d'$  dimension peut être quantifiée par le calcul de pourcentage cumulé comme suit :

$$\Gamma_i = \frac{\sum_{i=1}^d \lambda_i}{\sum_{i=1}^d \lambda_i} \quad (4.6)$$

Dans le but d'obtenir une représentation correcte reflétant le comportement réel, ce pourcentage doit être proche de 1. En pratique, généralement, il suffit de fixer le seuil dans l'intervalle [80% - 90%]. Alors, la procédure consiste à la sélection du vecteur propre possédant la plus grande valeur propre jusqu'à qu'on obtient le seuil fixé. A ce niveau, la base réduite est composée des vecteurs propres retenus.

## 4.5. Etape 2 : Construction de la base des connaissances

La construction de notre base des connaissances passe par les étapes montrées dans la figure 4.1. On détaille dans ce qui suit ces étapes [41] :

### 4.5.1. Définition de système

La définition de système nous permet de déterminer les frontières du système étudié et son environnement. Cette étape est importante pour éviter l'introduction des éléments externes dans l'étude, qui peuvent influencer sur la modélisation de système.

### 4.5.2. Analyse fonctionnelle

Le rôle de l'analyse fonctionnelle est de comprendre le fonctionnement de système en définissant les différentes activités de système par le développement des abstractions fonctionnelles hiérarchiques. Elle permet l'établissement des relations fonctionnelles de système d'une manière rationnelle et compréhensible. Pour atteindre ce but, nous avons utilisé la méthode SADT pour la décomposition fonctionnelle de système.

### 4.5.3. Analyse dysfonctionnelle

#### 4.5.3.1. Elaboration des tableaux AMDEC

Le but de L'AMDEC est de bien situer les modes de défaillance, leurs effets, leur criticité et les causes probables de ces défaillances, ainsi que les tâches de maintenance proposées. En plus, la valeur de la criticité des défaillances servi dans notre approche à définir l'ordre d'exécution des règles dans le SE.

### 4.5.3.2. Construction de l'AdD

Il n'existe pas une méthode formelle pour vérifier l'exactitude de l'arbre développé, nous avons utilisé l'algorithme proposé dans [41], pour déduire directement l'AdD à partir des modèles SADT et AMDEC élaborées précédemment.

#### ALGORITHME

---

<b>ETAPE 1</b>	<i>Démarrer la racine de l'AdD avec la négation de l'activité initial (A0) donnée par le model SADT réalisé (niveau k= 1).</i>
<b>ETAPE 2</b>	<i>Placer la négation des activités résultantes de la décomposition de l'activité initiale dans le niveau suivant (niveau k=2).</i> <ul style="list-style-type: none"> <li>• <i>Si nous avons une redondance parallèle des activités, alors l'opérateur entre la racine et les nœuds est un ET.</i></li> <li>• <i>Si nous avons des activités dans un ordre séquentiel, alors l'opérateur entre la racine et les nœuds est un OU.</i></li> </ul>
<b>ETAPE 3</b>	<i>Les nœuds obtenus dans le niveau k+1 sont connectés au niveau k en appliquant le même raisonnement de l'ETAPE 1 et l'ETAPE 2.</i>
<b>ETAPE 4</b>	<i>S'il n y a pas des nœuds à extraire à partir de modèle SADT élaboré dans le niveau k+1 alors :</i> <i>Les derniers nœuds obtenus reflètent le mauvais fonctionnement des éléments élémentaires dans le modèle SADT qui peut être causé par la défaillance des mécanismes ou un mauvais contrôle</i>
<b>ETAPE 5</b>	<i>Les nœuds de niveau k+2 sont obtenus à partir de modèle AMDEC élaboré, par la connexion de chaque nœud de niveau k+1 avec les nœuds représentant les défaillances des composants des mécanismes ou bien des procédures de régulation.</i>
<b>ETAPE 6</b>	<i>Obtenir le reste de l'AdD à partir de modèle AMDEC par la connexion de chaque nœud de niveau k+2 avec les causes de défaillances donné dans l'AMDEC de système.</i>

---

**Algorithme 4.1 : Etapes d'élaboration de la base des connaissances**

#### 4.5.4. Représentation des prédicats

Plusieurs méthodes existent pour la représentation des prédicats, mais dans notre approche on va exploiter les résultats obtenus dans la forme de diagramme de l'AdD.

L'AdD construite décrit les relations causales d'une manière graphique, sa traduction sous forme des prédicats est une tâche simple à réaliser. On peut définir deux types d'expressions de prédicats dans la modélisation de l'AdD :

- Expressions des prédicats définissent les relations entre les nœuds de l'arbre sous forme des opérateurs logiques (ET, OU). Ces prédicats reflètent l'aspect statique de processus.
- Expressions des faits définissent les défaillances présentées dans le dernier niveau de l'arbre. Ces faits reflètent l'aspect dynamique de processus. L'existence et le nombre de ces faits varient avec le temps selon l'occurrence des défaillances des composants élémentaires.

La forme de chaque type de prédicat est définie comme suit :

- L'existence d'un opérateur OU entre les nœuds A et B est exprimée par :

$$\text{lienOU}(A, B)$$

- L'existence d'un opérateur ET entre les nœuds A et B est exprimée par :

$$\text{lienET}(A, B).$$

La propagation des défaillances des nœuds fils vers le nœud père est donnée par :

##### **Pour l'opérateur OU**

$$[\text{lienOU}(A, x1) \text{ et } \dots \text{ et } \text{lienOU}(A, xn)] \text{ et } [\text{défaillance}(x1) \text{ ou } \dots \text{ ou } \text{défaillance}(xn)] \\ \Rightarrow \text{défaillance}(A).$$

##### **Pour l'opérateur ET**

$$[\text{lienET}(A, x1) \text{ et } \dots \text{ et } \text{lienET}(A, xn)] \text{ et } [\text{défaillance}(x1) \text{ et } \dots \text{ et } \text{défaillance}(xn)] \\ \Rightarrow \text{défaillance}(A).$$

$$\text{nbrFilsDéfaillants}(A) = \text{nbrFils}(A).$$

Le fait qui reflète l'occurrence d'une défaillance est présenté comme suit :

$$\text{défaillance}(x)$$

#### 4.5.5. Insertion des prédicats dans la base des connaissances

Le parcours de l'AdD nous permet d'extraire tous les prédicats nécessaires afin de les insérer dans la base des connaissances. Un exemple explicatif de cette extraction et insertion des prédicats est illustré sur la figure 4.5.

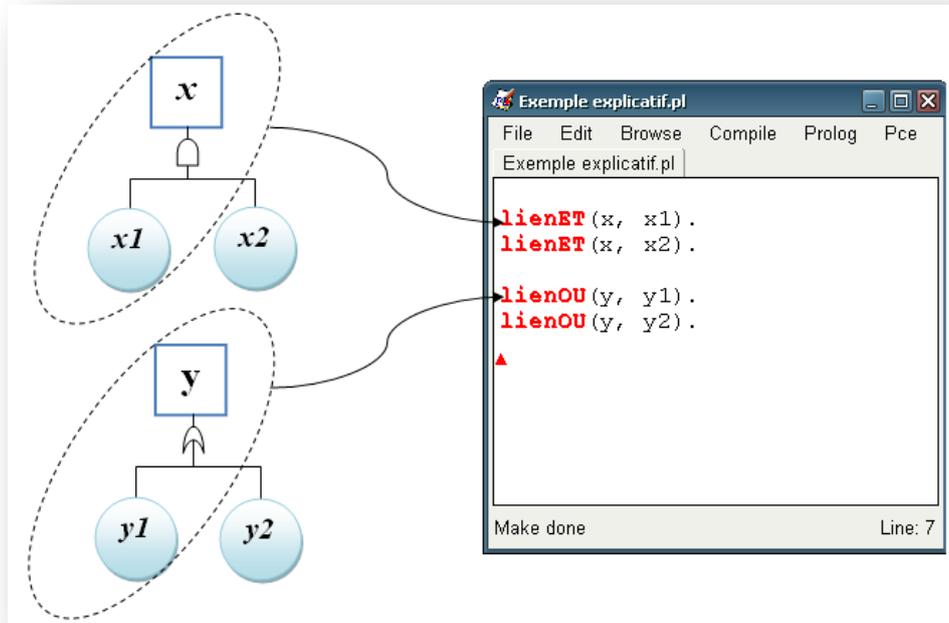


Figure 4.5 : Insertion des prédicats dans la base des connaissances

## 4.6. Conclusion

Dans ce chapitre nous avons présenté une méthodologie d'hybridation entre deux méthodes de l'IA, les SE et les RNA, dont le but est de réaliser un système de diagnostic des pannes. Ces deux méthodes sont utilisées pour achever une fonction bien définie et bien déterminée.

Le rôle générale de RNA est la détection des pannes élémentaires afin d'alimenter la base des faits (base des faits dynamique) de SE. Dans le but de réduire le nombre des variables d'entrées de RNA et implicitement une réduction des capteurs, nous avons proposé une approche statistique basée sur l'application de la régression pas à pas et l'analyse par composantes principales.

D'un autre côté, le rôle de SE consiste en la localisation et la prévision des défaillances probables et en plus la proposition des tâches de maintenance possibles en exploitant l'AMDEC de système étudié. Nous avons utilisé une approche bien structurée pour construire la base des connaissances du SE, établie par une analyse fonctionnelle (SADT) et une analyse dysfonctionnelle (AMDEC, AdD) du processus étudié, et nous avons présenté une technique simple d'insertion des prédicats dans la base des connaissances à partir de l'AdD.

Afin de valider l'approche proposée, le but de chapitre suivant est la discussion des résultats obtenus de l'application de cette approche pour un processus existant dans les industries du type agroalimentaire.

## CHAPITRE 5

# APPLICATION AU DIAGNOSTIC D'UN PROCESSUS INDUSTRIEL

***Résumé :** Ce chapitre est réservé à l'application de l'approche hybride proposée pour le diagnostic d'un processus industriel, dans le but de tester les performances de cette approche. En premier, on présente le processus industriel utilisé, ensuite, les résultats obtenus pour chaque phase sont donnés. On termine par une étude comparative et une conclusion concernant la validité des résultats et les limites de l'approche.*

## 5.1.Introduction

Dans ce chapitre, on s'intéresse à l'application de l'approche proposée pour le diagnostic des pannes d'un processus industriel. On a choisi comme prototype une partie d'une chaîne de production existante dans la plupart des systèmes industriels et plus précisément les systèmes agroalimentaires. L'application est réalisée en trois étapes.

Dans la première étape, on élabore un RNA dans le but d'alimenter la base des faits du SE. Au début on applique l'approche de réduction statistique proposée afin de réduire la taille de vecteur d'état. L'architecture du RNA (nombre de nœuds de la couche cachée) est testée d'une façon itérative jusqu'à l'obtention d'une structure adéquate au processus étudié.

Le rôle de la deuxième étape est la construction de la base des connaissances, par l'extraction de tous les prédicats possibles à partir de l'AdD du processus étudié. Cet arbre est construit à partir des modèles SADT et AMDEC du système élaborés.

La troisième étape est consacrée au codage sous prolog des prédicats générés, et la présentation d'un exemple de diagnostic du SE hybride réalisé.

## 5.2.Exemple d'application

### 5.2.1.Description du processus industriel

Notre application est réalisée pour un processus industriel existant dans les systèmes de production agroalimentaire présenté dans [18]

Le processus étudié est composé d'un réacteur et un échangeur de chaleur.

Une réaction exothermique  $A \rightarrow B$  aura lieu dans le réacteur. La température dans le réacteur est fixée à une valeur constante, par le biais d'un liquide préchauffé dans l'échangeur de chaleur où un autre liquide refroidissant diminue la température du liquide sortant du réacteur. Le débit de ce liquide est contrôlé à l'aide d'un contrôleur PI et une valve. En plus un autre contrôleur est utilisé pour stabiliser le niveau du liquide dans le réacteur. La figure 5.1 illustre le schéma de ce processus.

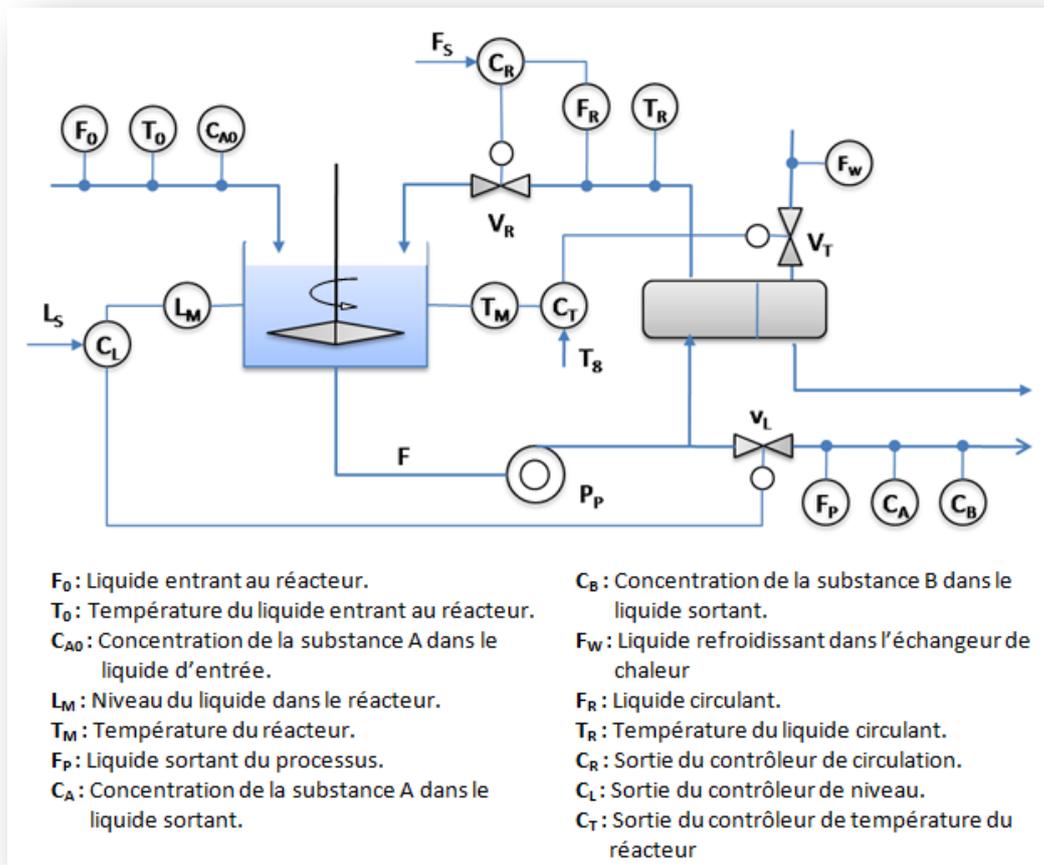


Figure 5.1 : Processus industriel avec réacteur et un échangeur de chaleur

L'historique des données concernant le fonctionnement du processus sont représentées sous forme d'une matrice avec 15 colonnes et 300 lignes. Chaque ligne indique une observation (la valeur des différents capteurs à une instante donnée). Les colonnes nous

donnent les critères (variables) mesurés par les capteurs et l'état de fonctionnement associé pour chaque observation. Les variables mesurés sont :  $\{F_0, T_0, C_{A0}, L_M, T_M, F_P, C_A, C_B, F_W, F_R, T_R, C_R, C_L, C_T\}$ .

Les modes de fonctionnement ayant servis à valider la procédure de diagnostic sont repris dans le tableau (5.1).

**Tableau. 5.1. Modes de fonctionnement de la chaîne de production étudiée.**

<i>Numéro</i>	<i>État de fonctionnement</i>
0	Normal
1	Blocage dans le liquide d'entrée
2	Blocage dans le liquide circulant
3	Haut concentration de la substance A
4	Haut seuil de circulation
5	Contamination de l'échangeur de chaleur
6	Blocage de la Valve pour le control de la température dans la position ouverte
7	Fuite dans le réacteur
8	Saturation du liquide circulant mesuré
9	Défaillance de la pompe

## **5.2.2. Etape 1 : Construction du RNA**

### **5.2.2.1. Application de l'approche de réduction statistique proposée**

Le vecteur d'état initial de notre processus contient les 14 paramètres définis dans le processus (Valeurs des capteurs).

$$Ve = [ F_0, T_0, C_{A0}, L_M, T_M, F_P, C_A, C_B, F_W, F_R, T_R, C_R, C_L, C_T ]$$

Les modes de défaillance sont déjà cités sur le tableau 4.2.

Par l'application de la régression pas à pas avant, le nombre initial de descripteurs peuvent êtres réduisis considérablement, alors pour les tâche de diagnostic, c'est une simplification commode pour garder seulement les termes significatifs statistiquement dans le modèle. On réitère le processus jusqu'à l'obtention d'une valeur plus de 90 indiquant que plus de 90% de variabilité des observations est conservée dans l'ensemble retenu comportant ainsi la plupart des paramètres pertinents.

La valeur du test F (F-statistic) est égale à 649.352 et la valeur p correspondante (p-value) égale à  $5.4 \times 10^{-166}$  indique qu'il n'est probable que tous les coefficients de régression

sont nuls. L'évaluation de la racine de l'erreur quadratique moyenne (RMSE) et le coefficient de corrélation pour chaque variable ajoutée est représenté dans le tableau 5.2 suivant :

**Tableau 5.2 : Résumé des résultats obtenus de l'application de la régression pas à pas.**

L'ensemble des descripteurs	RMSE	R <sup>2</sup>
{ }	2.877	0
{ F <sub>P</sub> }	2.836	0.031
{ F <sub>P</sub> , F <sub>W</sub> }	2.485	0.285
{ F <sub>P</sub> , F <sub>W</sub> , F <sub>R</sub> }	2.454	0.279
{ F <sub>P</sub> , F <sub>W</sub> , F <sub>R</sub> , C <sub>R</sub> }	2.315	0.361
{ F <sub>P</sub> , F <sub>W</sub> , F <sub>R</sub> , C <sub>R</sub> , C <sub>L</sub> }	0.986	0.884
{ F <sub>P</sub> , F <sub>W</sub> , F <sub>R</sub> , C <sub>R</sub> , C <sub>L</sub> , C <sub>T</sub> }	0.768	0.93

Selon la dimension de vecteur d'état obtenu (6 dimensions), une difficulté statistique inhérente concernant la visualisation multidimensionnelle est rencontrée, alors une représentation en 3D dans ce cas est utile. L'application de l'ACP a trois avantages: Elle rend les composantes orthogonales pour éliminer la corrélation; Elle classe les composantes orthogonales obtenues (composante principale) pour avoir une grande variation au début; et elle élimine les composantes qui ont une moindre variation dans l'ensemble d'observations. Les différents critères illustrant la variabilité conservée par chaque composante sont indiqués dans le tableau 5.3 suivant :

**Tableau 5.3 : Contribution des différentes composantes dans la variabilité des modes de défaillances**

Composante	Variance	Pourcentage	Pourcentage cumulé
<i>p1</i>	2.777	46.28%	46.28%
<i>p2</i>	1.800	30%	76.28%
<i>p3</i>	0.905	15%	91.28
<i>p4</i>	0.414	6.9%	98.18
<i>p5</i>	0.065	1.07%	99.25
<i>p6</i>	0.039	0.75%	100%

Il est clair que les trois premières composantes préservent plus de 90% de la variabilité totale fournie par le vecteur d'état réduit {F<sub>P</sub>, F<sub>W</sub>, F<sub>R</sub>, C<sub>R</sub>, C<sub>L</sub>, C<sub>T</sub>}. Alors le

nouveau système de coordonnées est représenté seulement par les trois composantes premières lié aux paramètres de vecteur d'état standards avec l'expression :

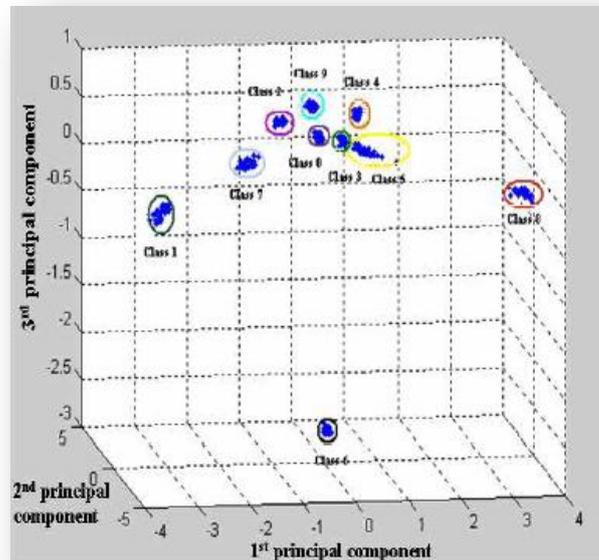
$$\begin{bmatrix} p1 \\ p2 \\ p3 \\ p4 \\ p5 \\ p6 \end{bmatrix} = \Pi \begin{bmatrix} F_P \\ F_W \\ F_R \\ C_R \\ C_L \\ C_T \end{bmatrix} \quad (5.1)$$

Où  $\Pi$  est la matrice des coefficients de modèle obtenu en utilisant la méthode de l'erreur quadratique minimal et elle est donné par :

$$\Pi = \begin{bmatrix} 0.461 & 0.463 & 0.392 & -0.395 & 0.405 & 0.354 \\ 0.460 & -0.033 & -0.414 & 0.559 & 0.532 & -0.137 \\ -0.005 & -0.607 & 0.101 & 0.161 & 0.074 & 0.767 \\ 0.092 & -0.344 & 0.777 & 0.243 & 0.135 & -0.438 \\ -0.321 & 0.545 & 0.247 & 0.671 & -0.134 & 0.269 \\ -0.681 & -0.008 & -0.025 & -0.149 & 0.715 & -0.045 \end{bmatrix} \quad (5.2)$$

En se basant sur l'espace obtenu par cette transformation, la visualisation graphique des différents modes de défaillance est possible comme schématisé sur la figure 5.2.

*Figure 5.2 : Différents modes de fonctionnement représentés dans l'espace réduis généré par l'ACP*

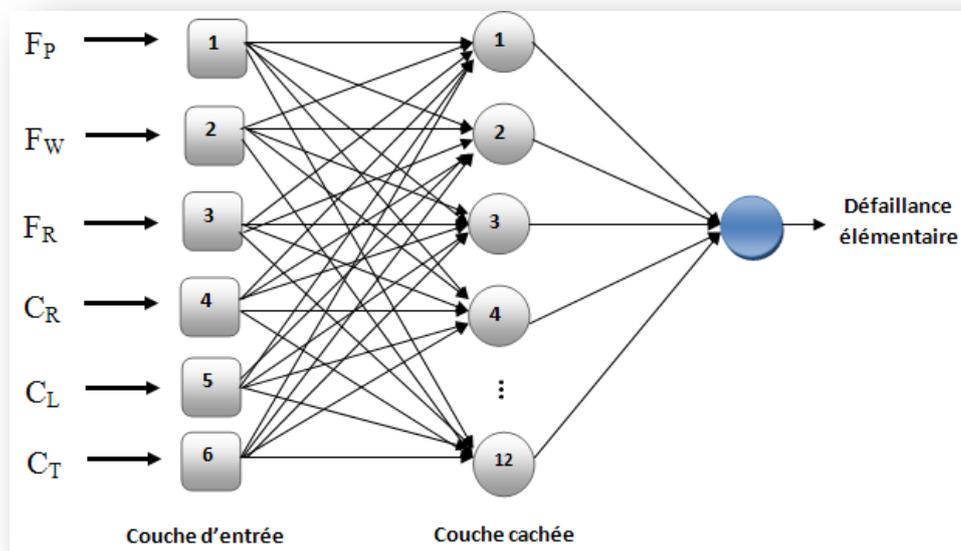


L'ensemble des paramètres préservés présente une grande capacité de séparation, comme il y a une bonne distinction des différents modes de défaillances de système en étude, l'application de l'approche est arrêtée.

### 5.2.2.2. Construction de RNA

La base de données de notre RNA contient 300 observations divisées en trois catégories : Apprentissage (225 observations), validation (45 observations) et test (30 observations). La première est utilisée pour ajuster les poids des connexions de RNA et la catégorie test est utilisée pour tester la configuration des poids. Les catégories des échantillons doivent être différentes et elles sont sélectionnées d'une manière aléatoire à partir de la base de données. Les étapes d'apprentissage et de teste sont effectuées pour arriver à une structure de RNA avec des valeurs optimales des poids. Le nombre de neurones est modifier de 1 à 15 et on prend en considération la valeur optimale du nombre de poids dans la couche cachée. La structure optimale de RNA est sélectionnée selon le minimum des erreurs résiduelles minimales obtenues pour l'ensemble des structures optimales. Le nombre optimal des neurones dans la couche cachée est fixé à une valeur de 12. Dans l'apprentissage de RNA, les poids de connexions sont ajustés par l'algorithme de retropropagation.

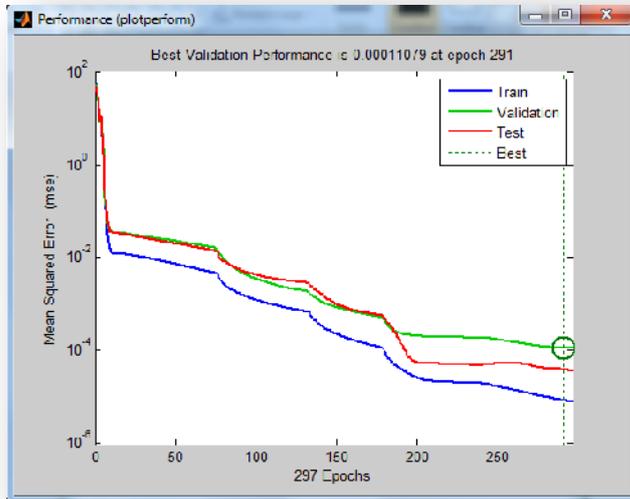
L'architecture optimale de notre RNA obtenu est illustrée sur la figure 5.3 :



**Figure 5.3 : Architecture optimale de RNA utilisé**

L'évolution de l'erreur quadratique minimale des différentes catégories est montrée sur la figure 5.4. La meilleure performance de validation est de 0.00011079 à l'itération 291. Il est possible de l'obtenir à partir des différentes courbes, où l'erreur quadratique minimale dans le

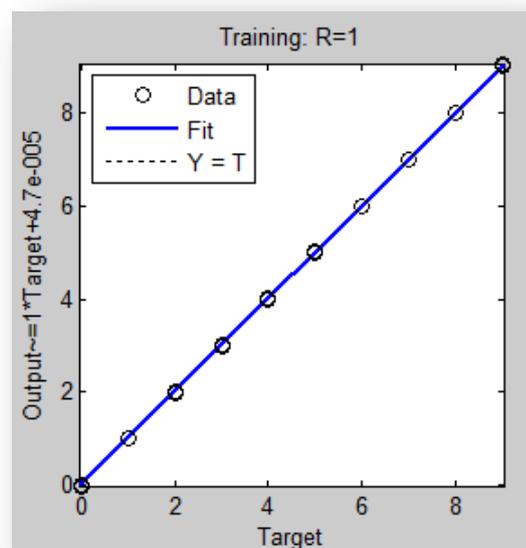
cas d'apprentissage est inférieure par rapport aux autres catégories. Cette différence peut être expliquée par l'existence de plus d'échantillons dans l'ensemble d'apprentissage par rapport aux autres, donc offrir plus de précision dans l'optimisation des résultats.

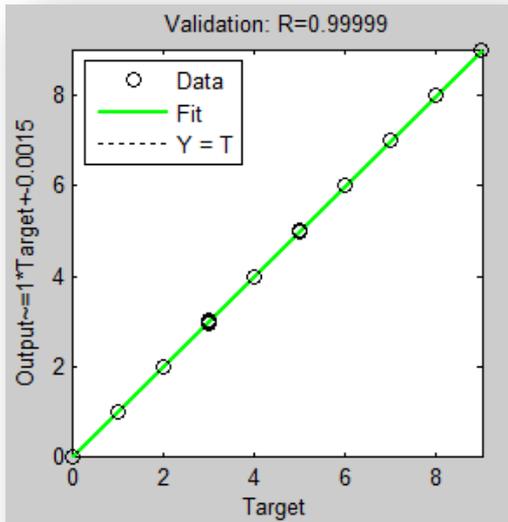


*Figure 5.4 : Evolution de l'erreur quadratique minimale avec le nombre d'itérations de chaque catégorie*

Dans le but de valider la prédiction de la structure optimale de notre réseau, l'ensemble d'apprentissage, l'ensemble de validation et l'ensemble de test sont comparées aux réponses du réseau. L'ensemble des courbes de régression est donné dans les figures 5.5, 5.6 et 5.7, où il est montré clairement la bonne concordance entre les résultats prédits et les résultats pratiques.

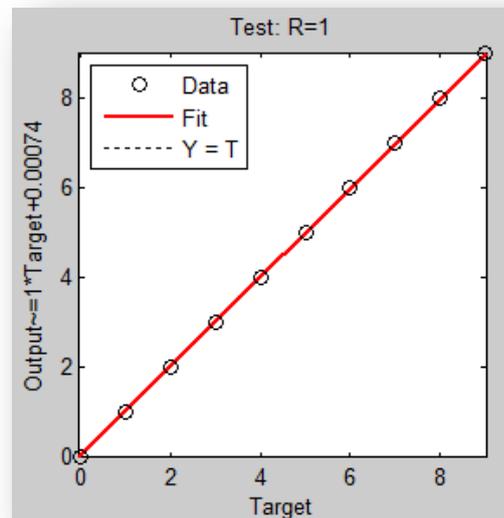
*Figure 5.5 : Représentation de la corrélation entre les données réelles et les données de notre RNA pour la phase d'apprentissage*





*Figure 5.6 : Représentation de la corrélation entre les données réelles et les données de notre RNA pour la phase de validation*

*Figure 5.7 : Représentation de la corrélation entre les données réelles et les données de notre RNA pour la phase de teste*



Comme résultat, la structure optimisée peut être utilisée pour prédire le mode correcte associé avec les combinaisons futures de variables d'entrées.

Le résumé des résultats obtenus des différentes catégories est présenté dans le tableau 5.4.

**Tableau 5.4 : Coefficient de performance de chaque phase.**

Catégorie	Erreur quadratique minimale	Coefficient de corrélation
<i>Apprentissage</i>	$8.22272 \times 10^{-6}$	0.999999
<i>Validation</i>	$1.10788 \times 10^{-4}$	0.999992
<i>Test</i>	$3.79833 \times 10^{-5}$	0.999997

Le coefficient de corrélation est très proche de 1 dans les différents cas, indiquant que l'ensemble des variables utilisées pour caractériser les modes de fonctionnement reflète d'une manière fidèle le comportement réel de processus.

Cette propriété montre bien la possibilité d'élaborer une séparation linéaire entre les différentes classes de fonctionnement, qui confirme le bon choix des paramètres mesurés utilisés pour la construction de vecteur d'état initial de processus étudié.

### 5.2.3. Etape 2 : Construction de la base des connaissances

#### 5.2.3.1. Définition de système

La définition de système est donnée dans la section 5.2.

#### 5.2.3.2. Analyse fonctionnelle

Les figures 5.8, 5.9, 5.10 et 5.11 décrivent le modèle SADT élaboré pour notre processus.

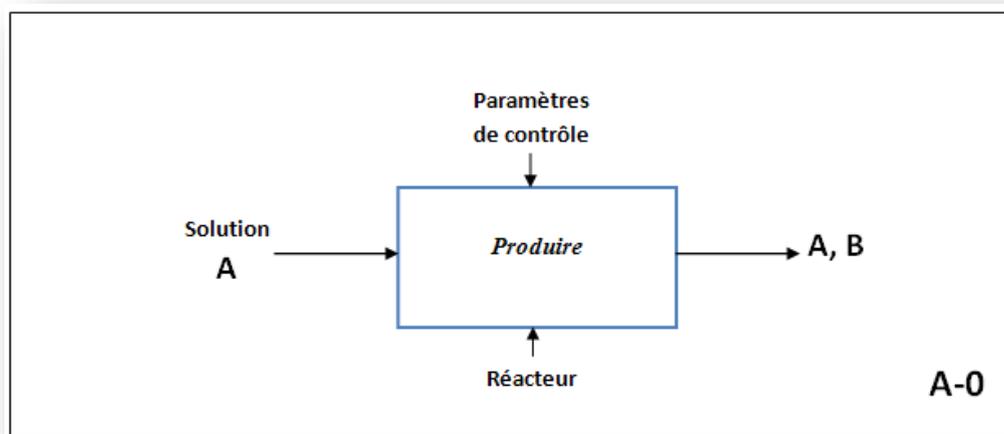


Figure 5.8 : Diagramme A-0 de processus

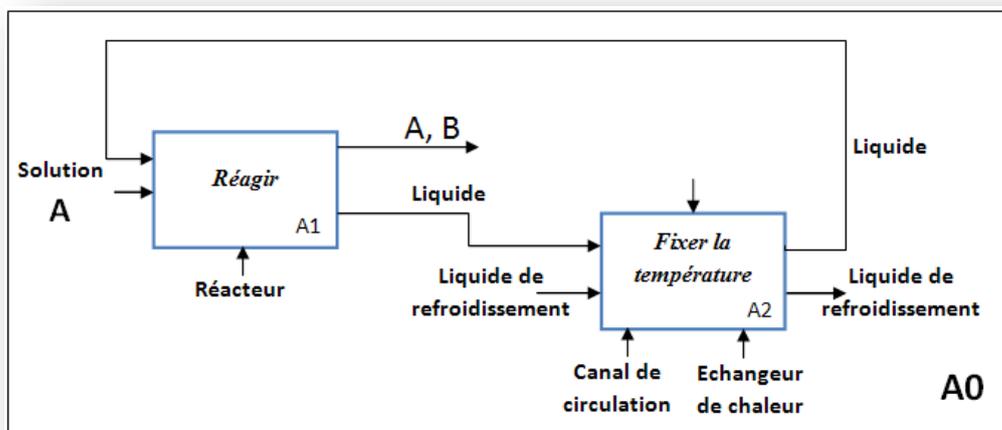


Figure 5.9 : Diagramme A0 de processus

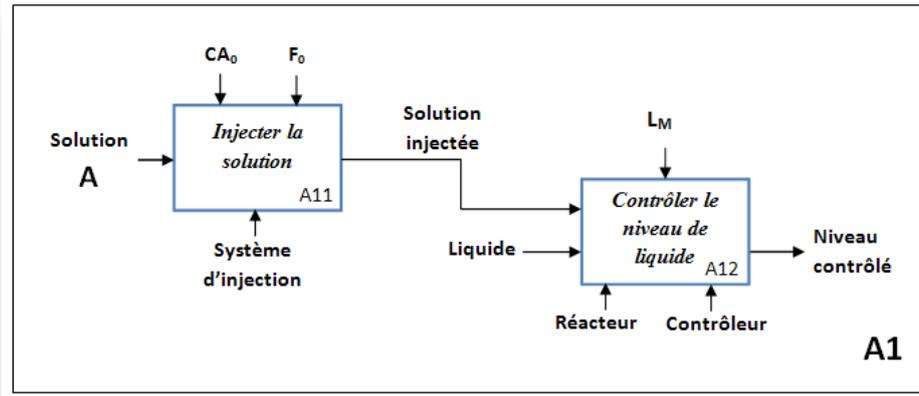


Figure 5.10 : Diagramme A1 de processus

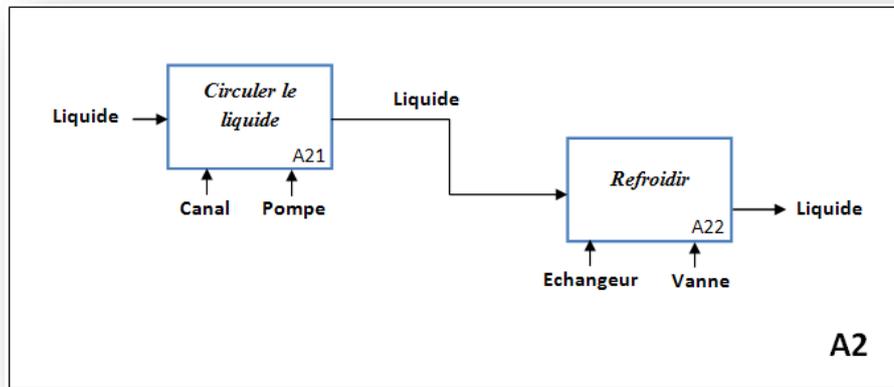


Figure 5.11 : Diagramme A2 de processus

5.2.3.3. Analyse dysfonctionnelle

5.2.3.3.1. Elaboration des tableaux AMDEC

La figure suivante (5.12) montre une partie d'un tableau AMDEC de processus.

AMDEC	ANALYSE DES MODES DE DEFAILLANCE DE LEURS EFFETS ET DE LEUR CRITICITE										AMDEC
	SYSTEME.....			PHASE DE FONCTIONNEMENT	DATE DE L4ANALYSE :	Criticité :				.....Page	
	SOUS-SYSTEME.....					TI	F	G	N		
Elément	Fonction	Mode de défaillance	Cause	Effet	Détection					ACTION CORRECTIVE	
Echangeur	Changer la chaleur	Problème de refroidissement	.Contamination .Fuite	Augmentation de la température			1	2	1	2	.Nettoyage .Soudage
...	...	...	...	...	...						...

Figure 5.12 : Exemple d'un tableau AMDEC de processus

5.2.3.3.2. Construction de l'AdD

La figure 5.13 illustre l'AdD construit.

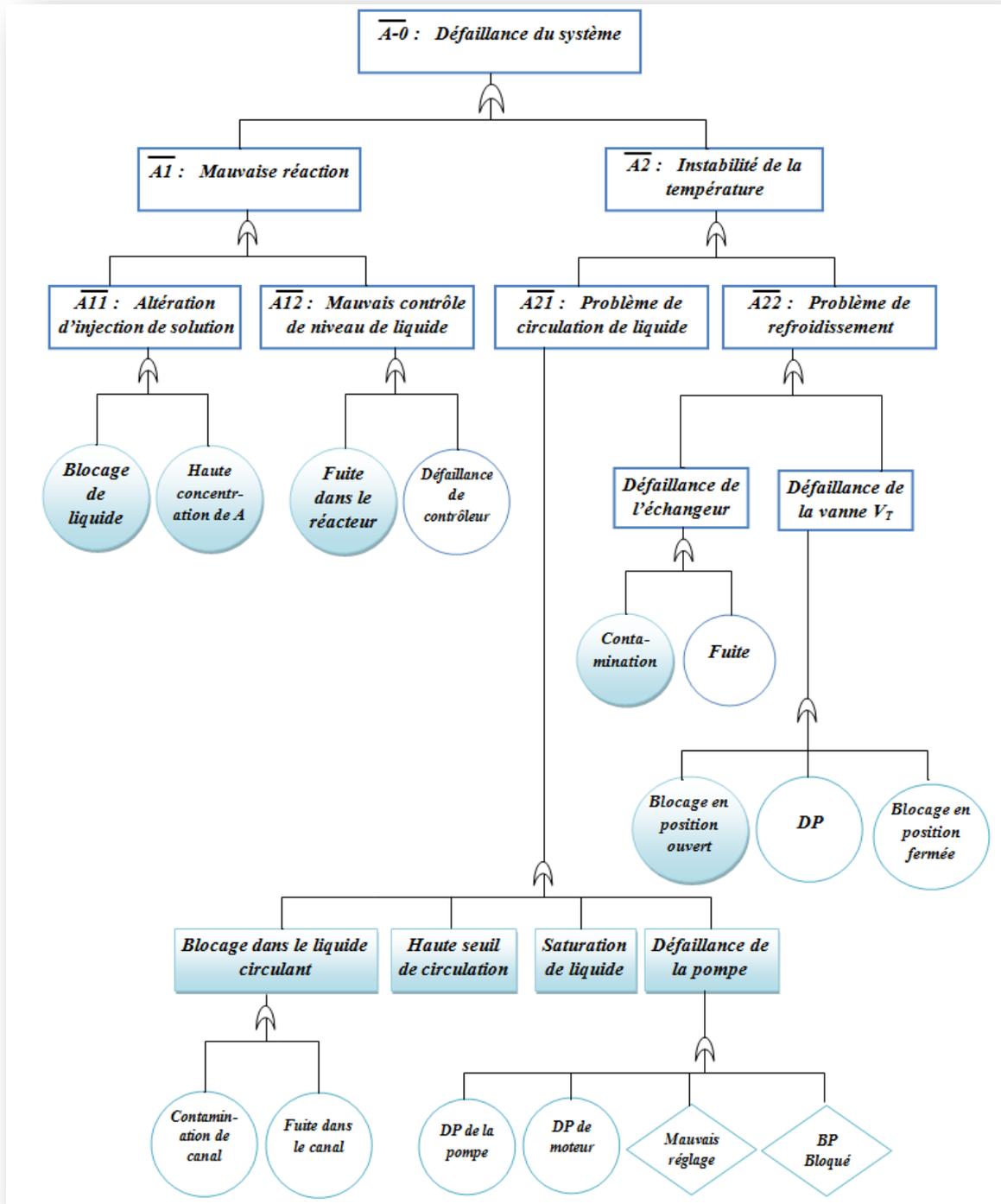


Figure 5.13 : L'AdD de processus

L'arbre construis peut être détaillé plus, mais nous n'avons visualisé que les défaillances détectés par les capteurs de notre système (les éléments colorés), dans le but d'illustrer l'application de l'approche.

Toutes les opérateurs existants dans l'arbre élaboré sont de type OU à cause de l'absence de redondances physique dans le système.

#### **5.2.3.4. Représentation des prédicats**

Les expressions des prédicats associées à l'AdD précédent:

##### ***Prédicats***

*lienOU (nonA0, nonA1).*

*lienOU (nonA0, nonA2).*

*lienOU (nonA1, nonA11).*

*lienOU (nonA1, nonA12).*

*lienOU (nonA2, nonA21).*

*lienOU (nonA2, nonA22).*

*lienOU (nonA11, ble).*

*lienOU (nonA11, hca).*

*lienOU (nonA12, fr).*

*lienOU (nonA12, dc).*

*lienOU (nonA21, blc).*

*lienOU (nonA21, hsc).*

*lienOU (nonA21, sl).*

*lienOU (nonA21, dp).*

*lienOU (blc, cc).*

*lienOU (blc, fc).*

*lienOU (dp, dpp).*

*lienOU (dp, dpmp).*

*lienOU (dp, mrp).*

*lienOU (dp, bpb).*

*lienOU (nonA22, de).*

*lienOU (nonA22, dvVT).*

*lienOU (de, ce).*

*lienOU (de, fe).*

*lienOU (dvVT, bpoVT).*

*lienOU (dvVT, bpfVT).*

*lienOU (dvVT, dpVT).*

### Indication des abréviations

*nonA0* : Défaillance du système

*nonA1* : Mauvaise réaction

*nonA2* : Instabilité de la température

*nonA11* : Altération d'injection de solution

*nonA12* : Mauvais contrôle de niveau de liquide

*nonA21* : Problème de circulation de liquide

*nonA22* : Problème de circulation de liquide

*ble* : Blocage de liquide d'entrée

*hca* : Haute concentration de A

*fr* : Fuite dans le réacteur

*dc* : Défaillance de contrôleur

*blc* : Blocage dans le liquide circulant

*hsc* : Haute seuil de circulation

*sl* : Saturation de liquide

*dp* : Défaillance de la pompe

*cc* : Contamination de canal

*fc* : Fuite dans le canal

*dpp* : Défaillance première de la pompe

*dpmp* : DP de moteur de la pompe

*mrp* : Mauvais réglage de la pompe

*bpb* : Bouton poussoir bloqué

*de* : Défaillance de l'échangeur

*dvVT* : Défaillance de la vanne VT

*ce* : Contamination de l'échangeur

*fe* : Fuite dans l'échangeur

*bpoVT* : Blocage en position ouvert de la vanne VT

*dpVT* : Défaillance première de la vanne VT

*bpfVT* : Blocage en position fermée de la vanne VT

On propose la règle suivante pour modéliser la propagation des défaillances dans l'AdD

*lienOU(Père, Fils) et défaillance(Fils) => défaillance(Père).*

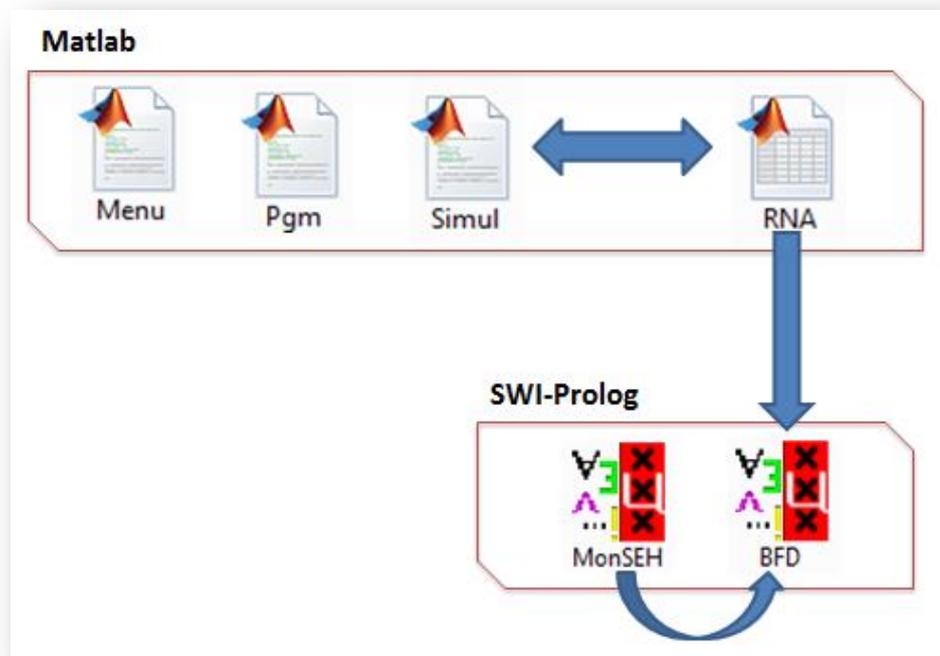
### 5.2.4. Codage sous prolog

Pour implémenter les prédicats obtenus et réaliser notre SE hybride proposé, nous avons utilisé le langage prolog.

Il existe différents outils qui nous permettent de programmer en Prolog, citons par exemple : SWI Prolog (*Possède un débogueur graphique ainsi que plusieurs solveurs de contraintes*) [43], GNU Prolog (*Propose un solveur de contraintes sur domaine fini*) [44], Sicstus Prolog (payant) (*Possède des extensions en plus, dont plusieurs solveurs de contraintes*) [45].

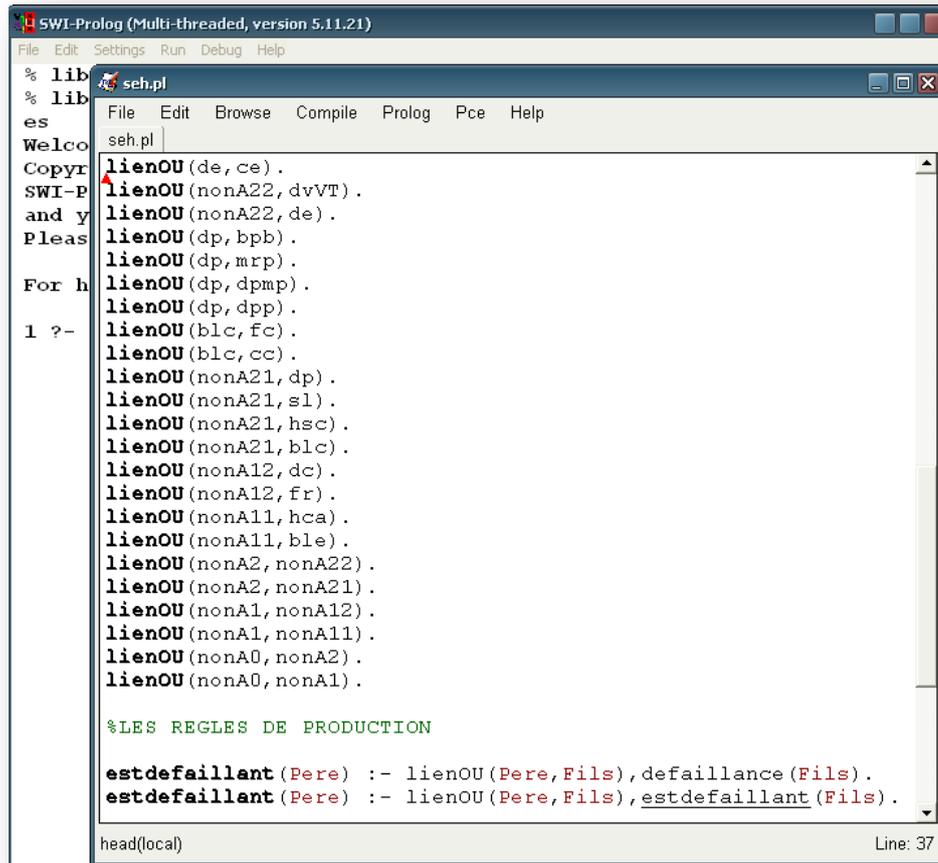
Pour notre cas nous avons adoptés le SWI Prolog.

Nous avons également utilisé le Matlab pour développer le RNA, le programme de simulation et en plus réaliser une interface simple pour la consultation. Le langage Prolog est utilisé pour développer le système expert où la base des faits dynamique de système est séparée de programme principal. Le RNA insert l'évènement détecté dans la base des faits dynamique de système sous forme d'un prédicat, et le système expert consulte la base et lance le chainage pour la détection de la propagation des défaillances. La figure suivante 5.14 montre les différents niveaux de notre programme ainsi que le lien commun.



*Figure 5.14 : Architecture générale de l'application développée.*

La figure (5.15) illustre une partie de la base des connaissances de notre SE :



```

SWI-Prolog (Multi-threaded, version 5.11.21)
File Edit Settings Run Debug Help

% lib
% lib
es
Welco
Copyr
SWI-P
and y
Pleas
For h
1 ?-

seh.pl
File Edit Browse Compile Prolog Pce Help

lienOU(de, ce).
lienOU(nonA22, dvVT).
lienOU(nonA22, de).
lienOU(dp, bpb).
lienOU(dp, mrp).
lienOU(dp, dpmp).
lienOU(dp, dpp).
lienOU(blC, fc).
lienOU(blC, cc).
lienOU(nonA21, dp).
lienOU(nonA21, s1).
lienOU(nonA21, hsc).
lienOU(nonA21, blC).
lienOU(nonA12, dc).
lienOU(nonA12, fr).
lienOU(nonA11, hca).
lienOU(nonA11, ble).
lienOU(nonA2, nonA22).
lienOU(nonA2, nonA21).
lienOU(nonA1, nonA12).
lienOU(nonA1, nonA11).
lienOU(nonA0, nonA2).
lienOU(nonA0, nonA1).

%LES REGLES DE PRODUCTION

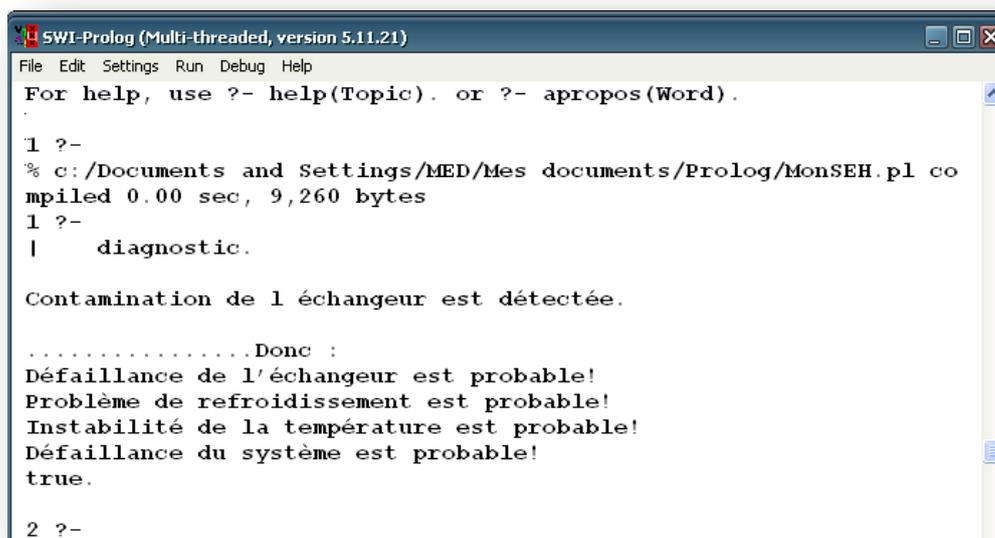
estdefaillant(Pere) :- lienOU(Pere, Fils), defaillance(Fils).
estdefaillant(Pere) :- lienOU(Pere, Fils), estdefaillant(Fils).

head(local)
Line: 37

```

Figure 5.15 : Une partie de la base des connaissances de notre SE sous SWI-Prolog

Le résultat de diagnostic du SE est montré sur la figure 5.16 suivante :



```

SWI-Prolog (Multi-threaded, version 5.11.21)
File Edit Settings Run Debug Help

For help, use ?- help(Topic). or ?- apropos(Word).

1 ?-
% c:/Documents and Settings/MED/Mes documents/Prolog/MonSEH.pl co
mpiled 0.00 sec, 9,260 bytes
1 ?-
| diagnostic.

Contamination de l échangeur est détectée.

.....Donc :
Défaillance de l'échangeur est probable!
Problème de refroidissement est probable!
Instabilité de la température est probable!
Défaillance du système est probable!
true.

2 ?-

```

Figure 5.16 : résultat de diagnostic du SE

La propagation des de défaillances dans l'arbre associé aux résultats précédents est indiquée sur la figure 5.17.

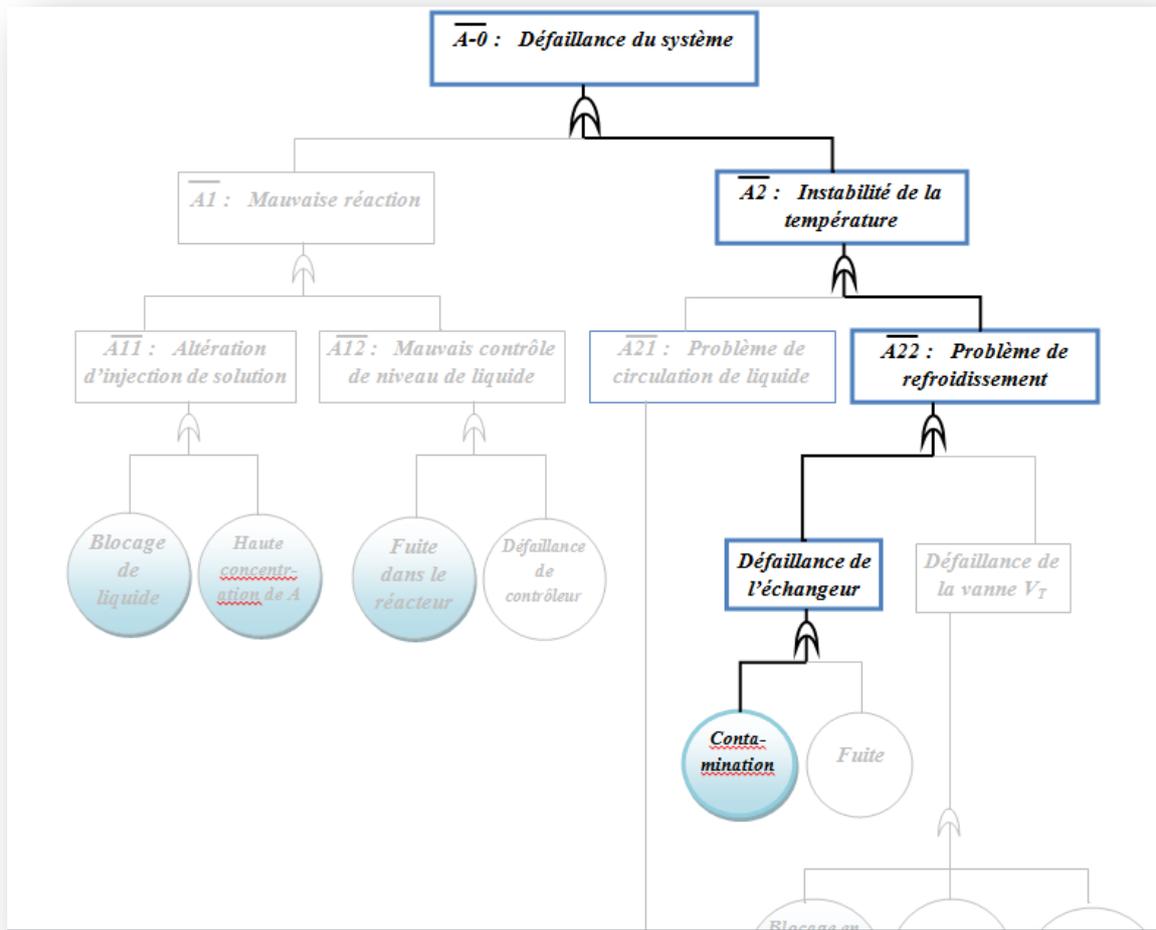


Figure 5.17: Propagation des défaillances dans la procédure de diagnostic.

Dans ce qui suit le chaînage de l'exécution des règles ainsi que le contenu de la base des faits sont illustrés dans le tableau 5.5. Le processus commence avec l'insertion de prédicat correspondant la panne élémentaire détecté jusqu'à la fin de l'exécution.

**Tableau 5.5 : Déroulement de chainage pour la détection des défaillances dans le système étudié.**

Règle	Base des faits	Description
0	$\emptyset$	Aucune défaillance détectée.
1	<i>defaillance(ce)</i>	Détection d'une panne élémentaire.
2	<i>lienOU(de, ce) et defaillance(ce) =&gt; defaillance(de).</i>	Génération de défaillance de deuxième niveau.
3	<i>lienOU(nonA22, de) et defaillance(de) =&gt; defaillance(nonA22).</i>	Génération de défaillance de troisième niveau.
4	<i>lienOU(nonA2, nonA22) et defaillance(nonA22) =&gt; defaillance(nonA2).</i>	Génération de défaillance de quatrième niveau.
5	<i>lienOU(nonA0, nonA2) et defaillance(nonA2) =&gt; defaillance(nonA0).</i>	Défaillance de système globale.

### 5.2.5. Etude comparative

Afin de bien situer les avantages et les inconvénients de notre approche, une étude comparative avec une approche de diagnostic présentée dans [18] pour le même système, est illustrée sur le tableau 5.6.

L'architecture proposée dans [18] joue le rôle de notre RNA utilisé pour la détection des défaillances et elle ne prend pas en considération la propagation des défaillances et les tâches de maintenances à entreprendre.

Tableau 5.6 : Etude comparative.

	Notre approche	Approche [18]
<i>Mesures de maintenance</i>	Prise en considération des effets, la criticité de la défaillance et les tâches de maintenance à entreprendre en suivant les consignes de l'AMDEC.  Facilité de bien assimiler la structure et les relations de causalité existantes dans le système en parcourant l'AdD.	Non supportées
<i>Prise en compte de la propagation des défaillances</i>	La propagation de la défaillance élémentaire à travers le système est schématisée à l'aide de l'AdD où le SE déduit toutes les propagations possibles de la défaillance initialement détectée par le RNA, en outre il est possible de présenter les causes probables de cette défaillance	Non supportée
<i>Critère de décision</i>	Le critère de décision utilisé est basé sur le calcul de la racine de l'erreur quadratique moyenne considéré comme un critère global car il prend en considération la totalité de l'ensemble des observations donc il est plus significatif.	La distance par rapport au k plus proche observation donc il est considéré comme une mesure locale et la sélection des observations dépendent de la métrique choisie.
<i>Erreur de classification dans la phase de l'apprentissage</i>	L'erreur de classification est calculée en utilisant l'algorithme de retropropagation qui est égale à $8.2227 \times 10^{-6}$	L'erreur de classification est basée sur une règle floue qui nous donne une valeur de $8.2304 \times 10^{-3}$
<i>Réduction de la</i>	La réduction est achevée en	La réduction est basée sur

<i>taille de vecteur d'état</i>	utilisant un modèle analytique: l'optimisation d'une fonction modèle linéaire multiple objective (rapport de Fischer). contenant les variables les plus significatives.
<i>Rejets d'ambiguïté et de distance</i>	Non supportés Ces rejets sont pris en considération par l'introduction des seuils au niveau des fonctions d'appartenance.

### 5.3. Conclusion

Dans ce chapitre nous avons appliqué l'approche proposée (SE hybride) pour le diagnostic d'une chaîne de production.

Pour atteindre ce but, nous avons utilisé MATLAB afin de développer un RNA efficace et exactement un réseau basé sur l'algorithme de retropropagation. Pour assurer une précision acceptable de classification de ce RNA, nous avons utilisé plusieurs catégories de données (Apprentissage, test et validation), avec l'évolution de l'erreur quadratique minimale des différentes catégories et les courbes de régression. L'entrée de RNA est un vecteur contenant les valeurs des capteurs utilisés qui sont le résultat de la réduction de la taille du vecteur d'état par l'application d'une approche de réduction statistique proposée.

Nous avons utilisé également le prolog, outil de programmation logique, pour coder et implémenter ce SE.

Selon les résultats obtenus, on peut dire que l'implémentation de ce SE hybride proposé sur un processus industriel réel est rentable, dans le sens de la réduction des coûts, mais nécessite un groupe des développeurs et une participation de tous les membres concernés par le domaine de l'industrie ciblé.

## CONCLUSION GENERALE

Le développement d'un pays est une exigence indispensable. L'un des critères importants de l'évaluation de cette exigence est le développement industriel. Alors l'importance donnée à ce secteur actif est rationnel, concrétisée par l'augmentation des projets de recherches industriels et le développement des nouvelles technologies pour automatiser le processus industriel. Cette automatisation des tâches d'exécution de processus oblige une automatisation dans l'une des tâches importantes pour tout système industriel, qui est le diagnostic des défaillances afin d'augmenter la productivité de système.

Plusieurs méthodes et technique ont été développées pour les tâches de diagnostic afin de bien surveiller les systèmes de production. Un état de l'art de ces méthodes et particulièrement les méthodes hybrides font l'objet de premier chapitre de mémoire.

Le diagnostic industriel est un problème de taille : Si des symptômes sont parvenus alors l'occurrence de telles défaillances est possible. Les SE sont l'un des méthodes de diagnostic basés sur ce type de règles, (*Si symptômes alors défaillance*). Ils constituent un bon moyen pour réaliser des systèmes de diagnostic. Le deuxième chapitre du mémoire est dédiée à la présentation des SE.

Les SE et particulièrement les systèmes à règles de production comporte deux éléments principaux : un moteur d'inférence et une base des connaissances. L'acquisition des connaissances est le problème majeur des SE, pour cela nous avons proposé un RNA basé sur l'algorithme de retropropagation pour détecter les pannes élémentaires par la classification des données fournies par les différents capteurs implantés au niveau du système industriel. La présentation des RNA fait l'objet du troisième chapitre.

Une première réalisation de ce RNA est affectée par la taille de l'ensemble des variables d'entrées un peu élevée, reflétant ainsi sur le nombre des capteurs utilisé. Un grand nombre de capteurs a une influence sur l'utilité d'adopter cette technique. Aussi, pour résoudre ce problème nous avons proposé une approche statistique permettant de réduire la taille de vecteur d'entrée de RNA en se basant sur la méthode de régression pas à pas et la méthode ACP.

Le choix de RNA pour l'acquisition des connaissances revient à leur capacité d'apprentissage pour la classification même en présence des données insuffisantes ou bruitées.

D'un autre côté, pour la construction de la base des connaissances nous avons adopté une approche basée sur l'analyse fonctionnelle et dysfonctionnelle du système étudié. La base des connaissances doit comporter toutes les informations relatives au dysfonctionnement de processus industriel. Pour cela, nous avons utilisé des AdD pour visualiser toutes les relations de causalité existantes dans le processus. Cette étape est enrichie par l'utilisation des tableaux AMDEC dans le but de bien présenter les causes, les effets et la criticité des défaillances et en plus les tâches de maintenance à entreprendre. Pour bien définir les modes de dysfonctionnement et remplir notre AdD il faut bien cerner le comportement fonctionnel du système. La méthode SADT, qui est une méthode graphique et structurelle, est employée pour bien présenter et comprendre le fonctionnement de système étudié. L'approche adoptée et les résultats obtenus sont détaillés dans le quatrième et cinquième chapitre de ce mémoire.

Dans ce mémoire nous avons réalisé une hybridation non seulement entre les SE et les RNA mais en plus avec d'autres méthodes (SADT, AMDEC...etc.) pour développer un système de diagnostic, où nous avons allégé les inconvénients d'une méthode par les avantages d'une autre.

Pour exploiter les objectifs visés nous avons développé et codé sous Prolog un SE hybride pour le diagnostic d'un processus industriel, et nous avons utilisé également le MATLAB pour développer le RNA. Les résultats obtenus sont acceptables pour valoriser cette approche à travers son implémentation sur un système réel. Comme il est également possible de prendre en considération dans nos futures perspectives le problème de l'apparition des défaillances multiples, ceci à travers des méthodes d'optimisation telles que les algorithmes génétiques ou bien le développement d'un RNA approprié pour chaque partie du système.

Enfin, on peut conclure que les systèmes experts hybrides sont de bons outils d'aide au diagnostic dans l'industrie, en particulier dans les environnements où l'expertise humaine est inexistante ou rare, et pourquoi pas constitué une solution pour la migration de la technologie (expertise) vers des domaines non encore maîtrisés.

## BIBLIOGRAPHIE

1. **Rolf Isermann.** *Fault-Diagnosis Applications - Model-Based Condition Monitoring : Actuators, Drives, Machinery, Plants, Sensors, and Fault-tolerant Systems.* Berlin : Springer, 2011.
2. **Shen-Tsu Wang, Wen-Tsann Lin.** *Research on integrating different methods of neural networks with case-based reasoning and rule-based system to infer causes of notebook computer breakdown.* Elsevier, 2009.
3. **Jian-Da Wu, Peng-Hsin Chiang, Yo-Wei Chang, Yao-jung Shiao.** *An expert system for fault diagnosis in internal combustion engines using probability neural network.* Elsevier, 2008. pp 2704–2713.
4. **George Vachtsevanos, Frank Lewis, Michael Roemer, Andrew Hess, Biqing Wu.** *Intelligent fault diagnosis and prognosis for engineering systems.* United States of America : JOHN WILEY & SONS., 2006.
5. **I. Hatzilygeroudis et J. Prentzas.** *Integrating (rules, neural networks) and cases for knowledge representation and reasoning in expert systems.* 2004, Elsevier, pp. 63–75.
6. **Robert Chapeaucou.** *Techniques d'amélioration continue en production.* Paris : DUNOD , 2000.
7. **Sylvain Piechowiak.** *Intelligence artificielle et diagnostic.* s.l. : Techniques de l'ingénieur.
8. **Gilles Zwingelstein.** *Diagnostic des défaillances.* Paris : Hermes, 1995.
9. **Ehsan Sobhani-Tehrani et Khashayar Khorasani.** *Fault Diagnosis of Nonlinear Systems Using a Hybrid Approach ,Lecture Notes in Control and Information Sciences.* New York : Springer Dordrecht Heidelberg, 2009.
10. **Steven X. Ding.** *Model-based Fault Diagnosis Techniques Design Schemes, Algorithms, and Tool.* Berlin : Springer, 2008.
11. **Venkat Venkatasubramanian, Raghunathan Rengaswamy, Surya N. Kavuri.** *A review of process fault detection and diagnosis Part I: Quantitative model-based methods.* 2003, Computers and Chemical Engineering, Elsevier Vol 27, pp. 293-311.
12. **Venkat Venkatasubramanian, Raghunathan Rengaswamy, Surya N. Kavuri.** *A review of process fault detection and diagnosis Part II: Qualitative models and search strategies.* 2003, Computers and Chemical Engineering, Elsevier, pp. 313-326.
13. **Venkat Venkatasubramanian, Raghunathan Rengaswamy, Surya N. Kavuri.** *A review of process fault detection and diagnosis Part III: Process history based methods.* 2003, Computers and Chemical Engineering, Elsevier ,pp. 327-346.
14. **Gilles Zwingelstein.** *Sûreté de fonctionnement des systèmes industriels complexes.* Techniques d'ingénieur.

15. **Matthias Kreimeyer et Udo Lindemann.** *Complexity Metrics in Engineering Design Managing the Structure of Design Processes.* Berlin : Springer, 2011.
16. **Limnios Nikolaos.** *Fault Trees.* USA : ISTE Ltd, 2007.
17. **Massimo Lazzaroni, Loredana Cristaldi, Lorenzo Peretto, Paola Rinaldi, et Marcantonio Catelani.** *Reliability Engineering, Basic Concepts and Applications in ICT.* Berlin : Springer, 2011.
18. **Toufik Bentrchia.** *Mémoire de magister : diagnostic et heuristique des systèmes industriels complexes.* Département de Génie Industriel Université El-Hadj Lakhdar – Batna – : Laboratoire d'Automatique et Productique, 2005.
19. **Vasile Palade, Cosmin Danut Bocaniala et Lakhmi Jain.** *Computational Intelligence in Fault Diagnosis Advanced Information and Knowledge Processing.* London : Springer, 2006.
20. **Cornelius T. Leondes.** *Fuzzy Logic and Expert Systems Applications Neural Network Systems Techniques and Applications.* United states of america : Academic press, 1998.
21. **Raouf Ketata et Yosra Najar.** *Générateur de Systèmes Experts Flous pour la Maintenance Industrielle.* s.l. : Institut National des Sciences Appliquées et de Technologie, Tunis Cedex, Tunisie.
22. **Orchard R. A.** *FuzzyCLIPS Version 6.04A User's Guide.* Canada : Institute for Information Technology, 1998.
23. **William.Siler et J.Buckley.** *Fuzzy expert systems and fuzzy reasoning.* New Jersey : John Wiley & Sons, 2005.
24. **K.Kasabov.** *Foundations of Neural Networks, Fuzzy Systems, and Knowledge Engineering.* London, England : The MIT Press, 1998.
25. **د محمد زكي، أ محمد خضر.**  
المؤتمر التقني الرابع لنظم الإتصالات والمعلومات في الدول الإسلامية . الأنظمة الخبيرة في التطبيقات الصناعية
26. **C.S.Krishnamoorthy, S.Rajeev.** *Artificial Intelligence and Expert Systems for Engineers.* s.l. : CRC Press, 1996.
27. **Konar Amit.** *Artificial Intelligence and Soft Computing Behavioral and Cognitive Modeling of the Human Brain.* Calcutta, India : CRC Press, 2000.
28. **Djouidi Mahieddine.** *Cours de systèmes experts.* s.l. : Université de Poitiers Département d'Informatique. Support du cours.
29. **J.Giareatano et G.Riley.** *Expert\_Systems\_Principles\_and\_Programming.* 2002.
30. **B.Badiru et Y.Cheung.** *Fuzzy engineering expert systems with neural network applications.* New York : JOHN WILEY & SONS, INC., 2002.

31. **Fabrice SORIN, Lionel Broussard, Pierre Roblin.** *Régulation d'un processus industriel par réseaux de neurones.* s.l. : Techniques d'ingénieur.
32. **Dave Anderson, George McNeill.** *Artificial neural networks technology.* utica, new york : Griffiss AFB, 1992.
33. **Eyfe Colin.** *Artificial Neural Networks .* Room : Department of Computing and Information Systems, The University of Paisley , 1996.
34. **G.Dreyfus, M.Martinez, M.Samielides, M.B.Gordon, F.Badran, S.Thiria, L.Hérolt.** *Réseaux de neurones - Méthodologie et applications.* s.l. : Eyrolles.
35. **G.Dreyfus.** *Les réseaux de neurones.* Paris : École Supérieure de Physique et de Chimie Industrielles de la Ville de Paris (ESPCI), 1998.
36. **Philippe Leray** *Quelques types de réseaux de neurones. Support de cours.*
37. **J.Russell, P.Norvig.** *Artificial intelligence A modern approach .* New Jersey : s.n., Vol. Prentice-Hall. 1995.
38. **J.Nilsson.** *Introduction to machine learning - an early draft of a proposed textbook .* Stanford : s.n., 1996 .
39. **T.Bentrcia,**  
 دراسة حالة مؤسسة صناعة قارورات الغاز : استخدام نموذج الشبكة العصبية الاصطناعية في التنبؤ بمستوى المبيعات  
 مذكرة مقدمة لنيل شهادة ليسانس في علوم التسيير 2009
40. **Shen-Tsu Wanga et Wen-Tsann Lin.** *Research on integrating different methods of neural networks with case-based reasoning and rule-based system to infer causes of notebook computer breakdown.,* Elsevier 2009.
41. **T.Bentrcia, L.H.Mouss et D.Mouss .** *A Reliability Based Approach for the Design of Hybrid Expert Systems in Industrial Diagnosis Field.,* in Proc, 2004 ISIICT'04, Information and Communication Technology Symp., pp. 81-91.
42. **T.Bentrcia, M.Bouzenita et L.H.Mouss .** *Enhancement of the Process State Vector for Industrial Diagnostic Applications Using a Statistical Reduction Approach,* CIAM'2011 International Conference on Automation and Mechatronics Oran, 2011. pp 386-390.
43. [www.swi-prolog.org](http://www.swi-prolog.org)
44. <http://gnu-prolog.inria.fr>
45. [www.sics.se/sicstus](http://www.sics.se/sicstus)